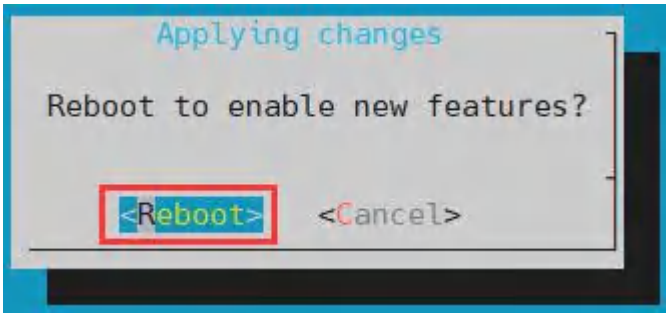


The above settings will eventually add the configuration of overlays=edp to /boot/orangepiEnv.txt. After setting, you can check it first. If this configuration does not exist, then there is a problem with the settings.

If you find it troublesome to use orangepi-config, you can also use the vim editor to open /boot/orangepiEnv.txt, and then add the configuration of overlays=edp.

orangepi@orangepi:~$ **cat /boot/orangepiEnv.txt | grep "edp"**
overlays=edp #sample configuration

3) After startup, you can see the display of the eDP screen as follows:

3. 31. 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:~$ 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:~$ 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. 32. How to use the ZFS file system

3. 32. 1. How to install ZFS

Note that linux6.6 is not supported yet.

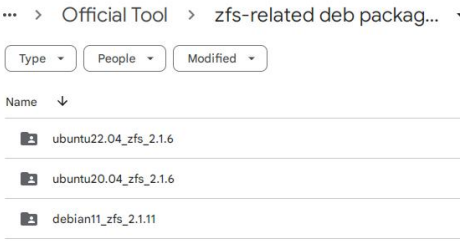
Before installing zfs, please make sure that the Linux image used is the latest version. In addition, if zfs is already installed in the system, it needs to be installed again.

Before installing zfs, you need to install the kernel header file first. For the method of installing the kernel header file, please refer to the instructions in the **section on the method of installing the kernel header file**.

In Ubuntu20.04, Ubuntu22.04 and Debian11 systems, zfs cannot be installed directly through apt, because the default apt source zfs version is lower than 2.1.6, and there is a problem of incompatibility with rk Linux5.10 kernel. This problem is fixed in zfs version 2.1.6 and later.

To solve this problem, we provide a zfs deb package that can be installed normally, which can be downloaded from the **official tool** of the development board. Open the **official tool**, and enter the **zfs-related deb package folders used by Ubuntu and**

Debian systems. You can see three types of deb packages: Ubuntu20.04, Ubuntu22.04 and Debian11. Please download the required version.



After downloading the zfs deb packages of the corresponding version, please upload them to the Linux system of the development board. For the upload method, please refer to **the description in the section of the method of uploading files to the Linux system of the development board.**

After the upload is complete, use the **cd** command in the command line of the development board Linux system to enter the directory of the deb package, and then use the following command to install the deb package of zfs.

```
orangepi@orangepi:~$ sudo apt install ./*.deb
```

After the installation is complete, use the following command to see the zfs-related kernel modules:

```
orangepi@orangepi:~$ ls /lib/modules/5.10.160-rockchip-rk356x/updates/dkms/
icp.ko  spl.ko  zavl.ko  zcommon.ko  zfs.ko  zlua.ko  znvpair.ko  zunicode.ko  zzstd.ko
```

Then restart the Linux system to see that the zfs kernel module will be automatically loaded:

```
orangepi@orangepi:~$ lsmod | grep "zfs"
zfs                  2801664  0
zunicode              327680  1 zfs
zzstd                 471040  1 zfs
zlua                  139264  1 zfs
zcommon                69632  1 zfs
znvpair                61440  2 zfs,zcommon
zavl                   16384  1 zfs
icp                   221184  1 zfs
spl                    77824  6 zfs,icp,zzstd,znvpair,zcommon,zavl
```

In Debian12, the default version of zfs is 2.1.11, so we can install zfs directly through the following command. Again, please make sure that the system has installed the deb package of the kernel header file before installation.

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

3. 32. 2. Methods of creating ZFS pools

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

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

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

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

2) Then enter the following command to create a ZFS pool, including two storage devices, NVMe SSD and U disk

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

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

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

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

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

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

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

6) Then we can test copying a file to the ZFS pool

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

3. 32. 3. Test the data deduplication function of ZFS

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

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

2) Then do a simple test, first enter pool1, and then execute the following command to generate a random file with a size of 1G

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

3) Then use the following command to copy 1000 random files of size 1G

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

4) Then use **du -lh** to see that there are currently 1002G of data in the pool, but in fact the size of the ZFS pool is only **504GB** (the total capacity of SSD+U disk), which cannot hold such a large amount of data

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

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

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

3. 32. 4. Test the data compression function of ZFS

1) Because the stored data is different, the disk space saved by compression will also be different, so we choose to compress relatively large plain text files for compression testing, and execute the following commands to pack the **/var/log/** and **/etc/** directories into a tarball

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

2) Then the file size that can be seen through the **ls -lh** command and the space occupied in the ZFS pool are both **27M**

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

3) Then we enable compression in the ZFS pool pool1

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

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

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

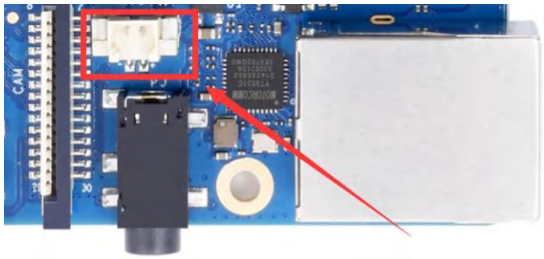
5) At this time, you can see that the size of the **text.tar** file is still 27M, but it only

occupies 9.47M in the ZFS pool, indicating that the file is compressed

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

3. 33. How to use RTC

1) An RTC battery interface is reserved on the development board. The location is as follows：：



2) The RTC battery that needs to be purchased is as follows, the interface is 2pin, 1.25mm pitch

3) The RTC chip used on the development board is HYM8563TS. The characteristics of this chip are:

a. Wide operating voltage range: 1.0~5.5v

b. Low sleep current: typical value is 0.25 μ A (VDD =3.0V, TA =25° C)

4) After connecting the RTC battery to the development board, use the following method to test whether the RTC is working normally:

a. First boot into the system, and then record the current time of the system.

b. Then use the poweroff command to shut down the system gracefully

c. Then unplug the power, make sure the development board is not connected to network cables and wireless WIFI, and wait a few more minutes.

d. Then start the system again. If you see that the time has moved forward a few minutes after entering the system, it means

The RTC module and battery are working normally

5) The command to view RTC information through the procfs interface of the Linux system is

```
orangepi@orangepi:~$ cat /proc/driver/rtc
rtc_time        : 10:10:36
rtc_date        : 2023-10-19
alrm_time       : 00:00:00
alrm_date       : 1999-12-16
alarm_IRQ       : no
alrm_pending    : no
update IRQ enabled      : no
periodic IRQ enabled    : no
periodic IRQ frequency  : 1
max user IRQ frequency  : 64
24hr            : yes
```

3. 34. Testing method of GPU in Linux6.6 system

Note that the desktop wallpaper may display a black screen after the GPU is turned on, so the GUP is turned off by default.

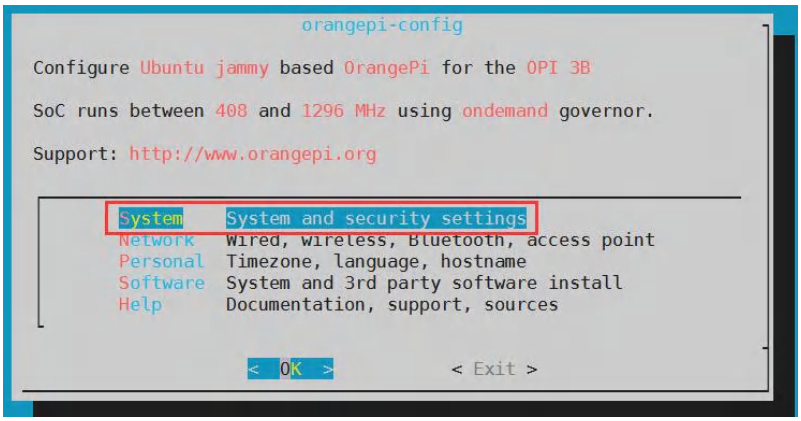
GPU is only available in Ubuntu22.04 and Debian12.

1) In Linux 6.6 system, the GPU is turned off by default and needs to be turned on manually before it can be used. Detailed steps are as follows:

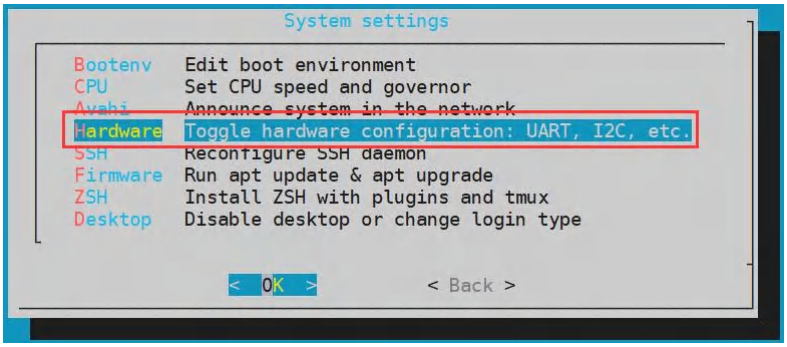
a. First run **orangepi-config**. Ordinary users remember to add **sudo** permissions

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

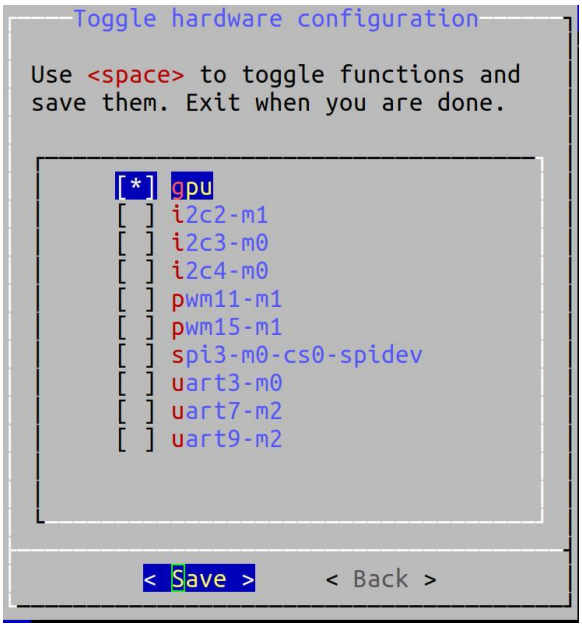
b. Then select System



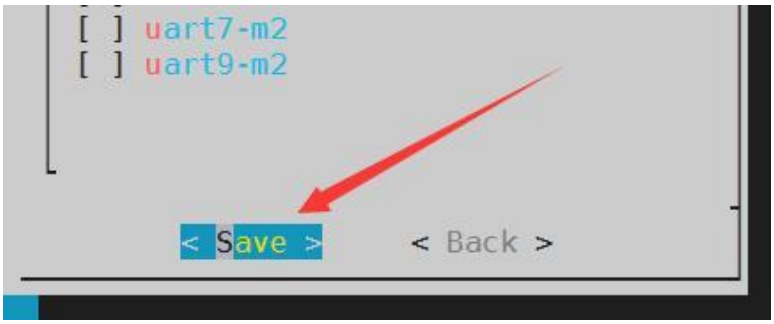
c. Then select Hardware



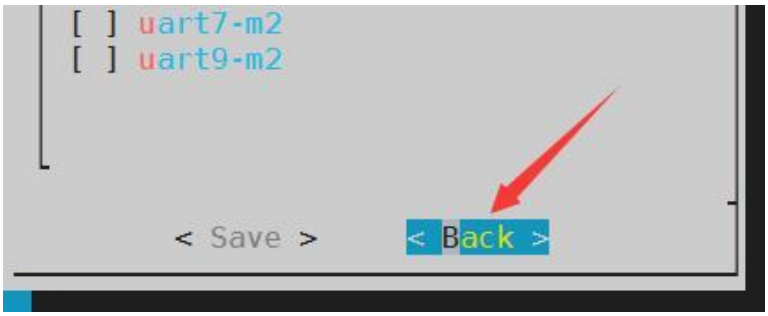
d. Then use the keyboard’s arrow keys to locate the location shown in the picture below, and then use the space to select the gpu option



e. Then select <Save> to save



f. Then select <Back>



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

2) Open a terminal on the desktop and enter the glmark2 command. If you can see GL_VENDOR followed by **Panfrost**, it means a GPU is used.

```
orangepi@orangepi:~$ glmark2
=======================================================
    glmark2 2023.01
=======================================================
    OpenGL Information
    GL_VENDOR:          Panfrost
    GL_RENDERER:        Mali-G52 r1 (Panfrost)
    GL_VERSION:         3.1 Mesa 22.3.6
    Surface Config: buf=32 r=8 g=8 b=8 a=8 depth=24 stencil=0 samples=0
    Surface Size:       800x600 windowed
=======================================================
```

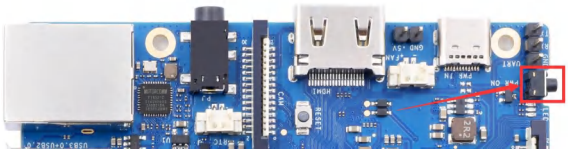
3. 35. How to shut down and restart the development board

1) During the running of the Linux system, if you directly unplug the Type-C power supply and cut off the power, the file system may lose some data or be damaged.

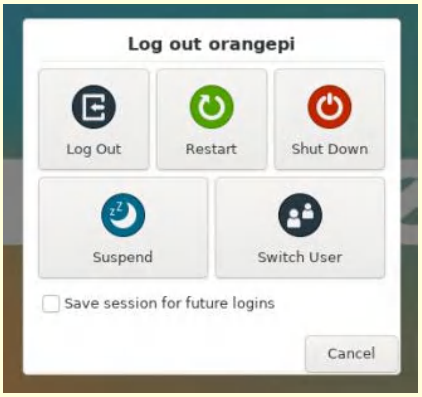
Therefore, please use the **poweroff** command to shut down the Linux system of the development board before cutting off the power. Then unplug the power supply.

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

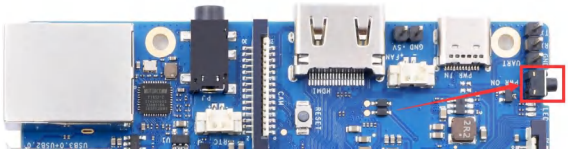
2) In addition, the development board is equipped with a power on/off button, and you can also **short press** the power on/off button on the development board to shut down.



Note that after pressing the power button on the Linux desktop version, the confirmation box shown in the figure below will pop up. You need to click the Shut Down option before Shut Down.



3) After shutting down, short press the power button on the development board to turn it on.



4) The command to restart the Linux system is

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

4. Linux SDK——orangepi-build instructions

4. 1. Compilation system requirements

We can cross-compile the Linux image of the development board on the x64 computer, or compile the Linux image of the development board on the Ubuntu22.04 system of the development board, please choose one according to your preference.

If you use orangepi-build to compile the Linux image in the Ubuntu22.04 system of the development board, please do a good job of cooling (especially when the SSD starts). If the heat dissipation is not done well, it is prone to the error of file system runaway.

4. 1. 1. Compile with the Ubuntu22.04 system of the development board

1) The Linux SDK, namely **orangepi-build**, supports running on the **Ubuntu 22.04** of the development board (other systems have not been tested), so before downloading orangepi-build, please first ensure that the Ubuntu version installed on the development board is Ubuntu 22.04. The command to check the Ubuntu version installed on the development board is as follows. If the Release field does not display **22.04**, it means that the current Ubuntu version does not meet the requirements. Please replace the system before performing the following operations.

```
orangepi@orangepi:~$ lsb_release -a
No LSB modules are available.
Distributor ID:	Ubuntu
Description:	Ubuntu 22.04.1 LTS
Release:	22.04
Codename:	jammy
```

2) Since the source codes such as the kernel and U-boot are stored on GitHub, it is very important to ensure that the development board can download codes from GitHub normally when compiling the image.

3) 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
or
https://repo.huaweicloud.com/ubuntu-releases/22.04/ubuntu-22.04.1-desktop-amd64.iso

4.1.2. Compile with x64 Ubuntu22.04 computer

1) The Linux SDK, **orangepi-build**, supports running on 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
```

2) If the computer is installed with 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 be careful not to 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.

3) 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
Or
https://repo.huaweicloud.com/ubuntu-releases/22.04/ubuntu-22.04.1-desktop-amd64.iso

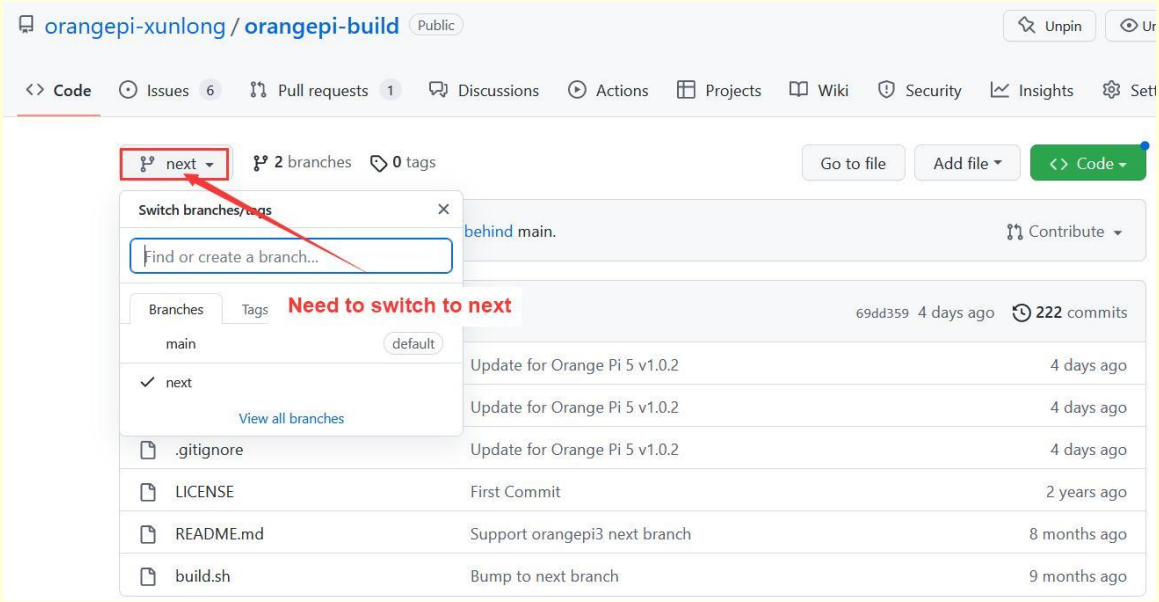
4. 2. Get the source code of Linux sdk

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

1) The Linux sdk actually 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. First download the code of orangepi-build, the command is as follows:

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

Note that the Orange Pi 3B development board 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.

2) The u-boot and Linux kernel versions currently used by the development board are as follows

branch	u-boot version	Linux Kernel version
legacy	u-boot 2017.09	Linux5.10
current	u-boot 2017.09	Linux6.6

The branch mentioned here is not the same thing as the branch of orangepi-build source code, please don't get confused. This branch is mainly used to distinguish different kernel source code versions.

We define the Linux5.10 bsp kernel currently provided by RK as the legacy branch. We define the latest Linux6.6 kernel as the current branch.

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

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

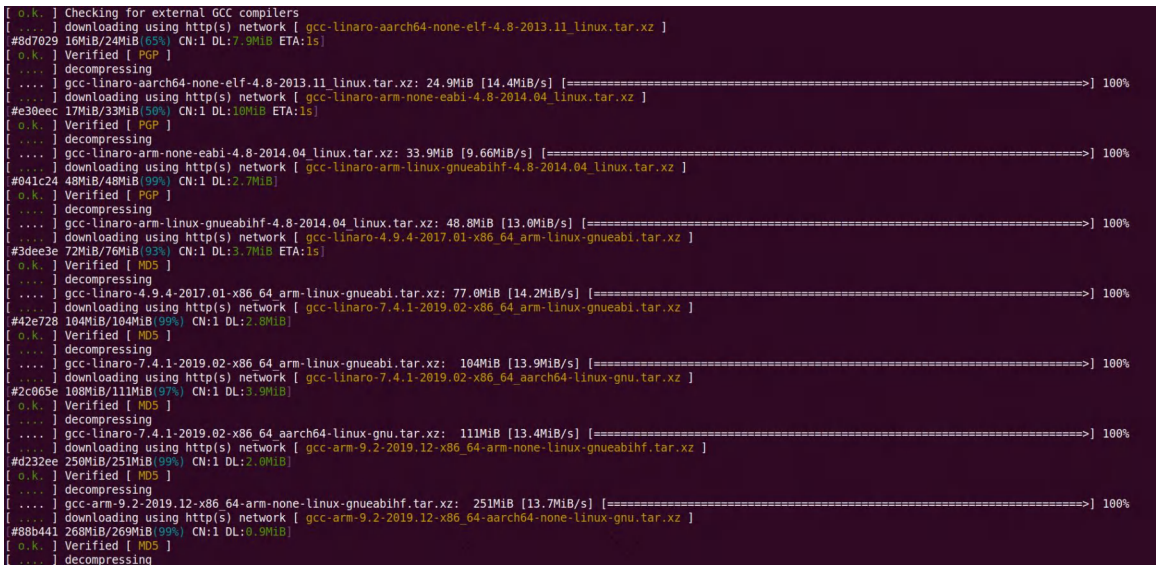
```
test@test:~/orangepi-build$ ls
build.sh  external  LICENSE  README.md  scripts
```

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

The cross-compilation toolchain will only be downloaded when the orangepi-build compilation image is used on an x64 computer. Compiling the Linux image of the development board in the Ubuntu22.04 of the development board will not download the cross-compilation toolchain. At this time, orangepi-build/toolchains will be an empty folder.

1) 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 after running the build.sh script of orangepi-build, it will check whether the cross-compilation toolchain in **toolchains** exists , if it does not exist, the download will be restarted, if it exists, it will be used directly, and the download will not be repeated.



2) The image URL of the cross-compilation toolchain in China is the open source software image site of Tsinghua University

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

3) After **toolchains** is downloaded, it will contain multiple versions of cross-compilation toolchains, and the development board will only use two of them

```
test@test:~/orangepi-build$ ls toolchains/
gcc-arm-11.2-2022.02-x86_64-aarch64-none-linux-gnu
gcc-arm-11.2-2022.02-x86_64-arm-none-linux-gnueabihf
gcc-arm-9.2-2019.12-x86_64-aarch64-none-linux-gnu
```

gcc-arm-9.2-2019.12-x86_64-arm-none-linux-gnueabihf gcc-linaro-4.9.4-2017.01-x86_64_arm-linux-gnueabi gcc-linaro-5.5.0-2017.10-x86_64_arm-linux-gnueabihf gcc-linaro-7.4.1-2019.02-x86_64_aarch64-linux-gnu gcc-linaro-7.4.1-2019.02-x86_64_arm-linux-gnueabi gcc-linaro-aarch64-none-elf-4.8-2013.11_linux gcc-linaro-arm-linux-gnueabihf-4.8-2014.04_linux gcc-linaro-arm-none-eabi-4.8-2014.04_linux

4) The cross-compilation toolchain used to compile the Linux kernel source code is

a. Linux5.10

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

5) The cross-compilation tool chain used to compile the u-boot source code is

a. v2017.09

gcc-linaro-7.4.1-2019.02-x86_64_aarch64-linux-gnu

4. 2. 3. orangepi-build complete directory structure description

1) After downloading, the orangepi-build warehouse does not contain the source code of the Linux kernel, u-boot and cross-compilation tool chain. The source code of the Linux kernel and u-boot is stored in an independent git warehouse.

a. The git warehouse where the Linux kernel source code is stored is as follows. Please note that the branch of the linux-orangepi warehouse is switched to

a) Linux5.10

https://github.com/orangepi-xunlong/linux-orangepi/tree/orange-pi-5.10-rk35xx

b) Linux6.6

https://github.com/orangepi-xunlong/linux-orangepi/tree/orange-pi-6.6-rk35xx

b. The git warehouse where the b.u-boot source code is stored is as follows:

https://github.com/orangepi-xunlong/u-boot-orangepi/tree/v2017.09-rk3588

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

a. **build.sh**: compile startup script

b. **external**: Contains the configuration files needed to compile the image, scripts with specific functions, and the source code of some programs. The rootfs

compressed package cached during the image compilation process is also stored in external

c. **kernel**: Stores the source code of the Linux kernel. The folder named orange-pi-5.10-rk35xx stores the kernel source code of the legacy branch of the RK3588/RK3588S/RK3566 series development boards. The folder named orange-pi-6.6-rk35xx stores the source code. It is the kernel source code of the current branch of the RK3566 series development board. Please do not modify the name of the folder of the kernel source code manually. If it is modified, the kernel source code will be re-downloaded when the compilation system is running.d. **LICENSE**: GPL 2 license file

e. **README.md**: orangepi-build documentation

f. **output**: Store compiled deb packages such as u-boot and Linux, compilation logs, and compiled images and other files

g. **scripts**: general scripts for compiling Linux images

h. **toolchains**: store cross-compilation toolchain

i. **u-boot**: stores the source code of u-boot, the folder named **v2017.09-rk3588** stores the u-boot source code of the legacy branch of the RK3588/RK3588S/RK3566 series development boards, the name of the folder of the u-boot source code Please do not modify it manually, if it is modified, the u-boot source code will be re-downloaded when the compiling system is running

j. **userpatches**: Store configuration files needed to compile scripts

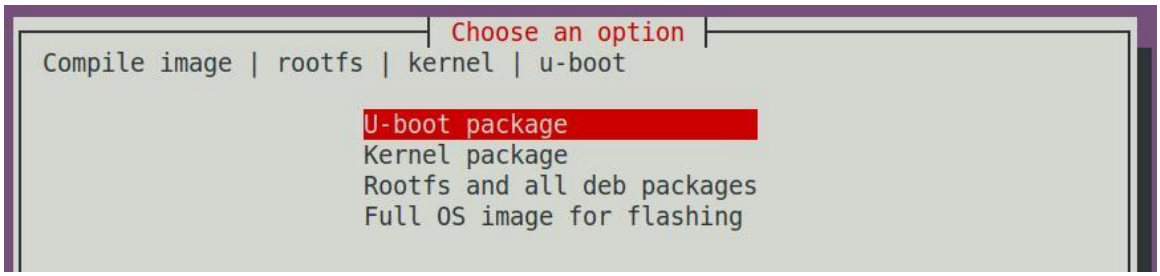
```
test@test:~/orangepi-build$ ls
build.sh  external  kernel  LICENSE  output  README.md  scripts  toolchains  u-boot  userpatches
```

4. 3. Compile u-boot

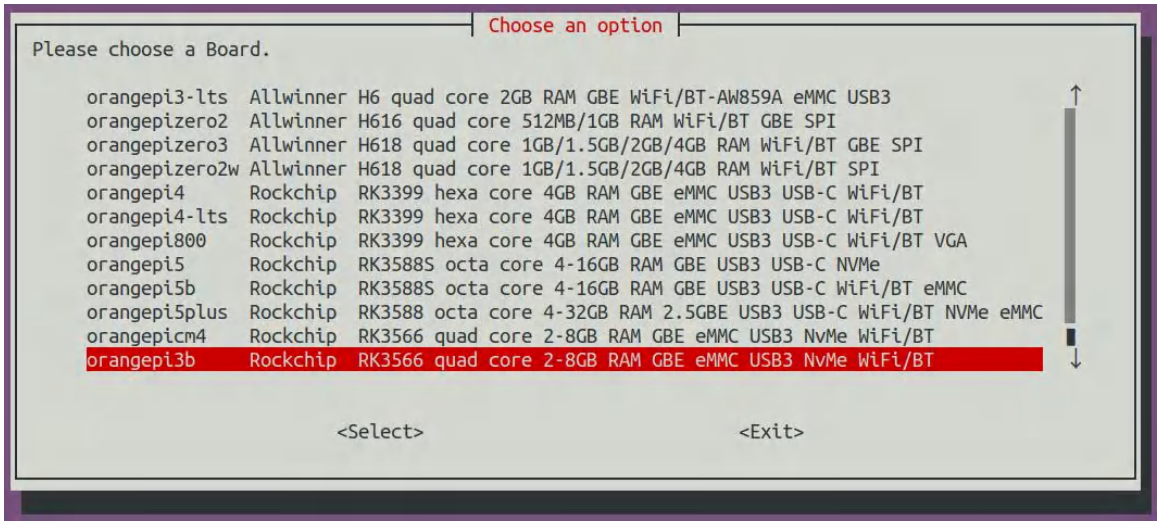
1) Run the build.sh script, remember to add sudo permission

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

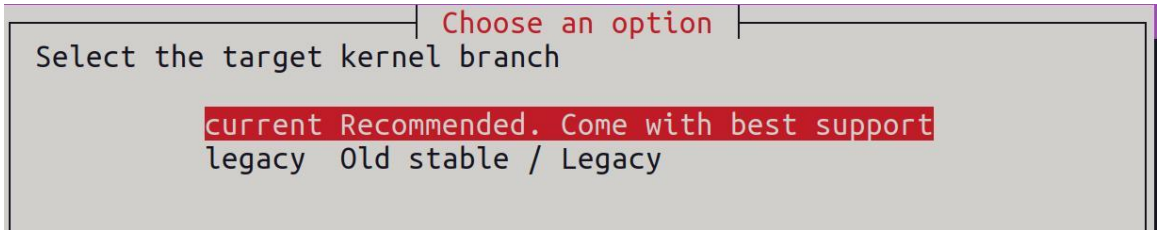
2) Select **U-boot package**，then enter



3) Then select the model of the development board



4) Then select u-boot to select the branch type. Both the current branch and the legacy branch will compile the code of the u-boot v2021.07 version that needs to be used.



5) Then it will start to compile u-boot. Some of the information prompted during compilation is as follows:

a. u-boot source code version

[o.k.] Compiling u-boot [**v2017.09**]

b. The version of the cross-compilation toolchain

[o.k.] Compiler version [**aarch64-linux-gnu-gcc 7.4.1**]

c. 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-legacy-orangepi3b_1.0.0_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=orangepi3b BRANCH=legacy BUILD_OPT=u-boot KERNEL_CONFIGURE=no ]
```

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

```
test@test:~/orangepi-build$ ls output/debs/u-boot/
linux-u-boot-legacy-orangepi3b_1.0.0_arm64.deb
```

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

a. Use the following command to decompress the deb package

```
test@test:~/orangepi-build$ cd output/debs/u-boot
test@test:~/orangepi_build/output/debs/u-boot$ $ dpkg -x \
linux-u-boot-legacy-orangepi3b_1.0.0_arm64.deb .      (Note that there is a "." at the end of the command)
test@test:~/orangepi_build/output/debs/u-boot$ ls
linux-u-boot-legacy-orangepi3b_1.0.0_arm64.deb    usr
```

b. The decompressed file is as follows

```
test@test:~/orangepi-build/output/debs/u-boot$ tree usr
usr
└── lib
    ├── linux-u-boot-legacy-orangepi3b_1.0.0_arm64
    │   ├── idbloader.img
    │   ├── rkspi_loader.img
    │   └── u-boot.itb
    └── u-boot
        ├── LICENSE
        ├── orangepi-3b-rk3566_defconfig
        └── platform_install.sh
```

```
3 directories, 6 files
```

8) 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. If the source code package downloaded from Baidu cloud disk, there is no such problem, because the source code of u-boot is all cached)**, otherwise the changes made will be reverted, the method is as follows:

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

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

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

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

```
test@test:~/orangepi-build$ cd output/debs/u-boot
test@test:~/orangepi_build/output/debs/u-boot$ scp \
linux-u-boot-legacy-orangepi3b_1.0.0_arm64.deb root@192.168.1.xxx:/root
```

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

```
root@orangepi:~# apt purge -y linux-u-boot-orangepi3b-legacy
```

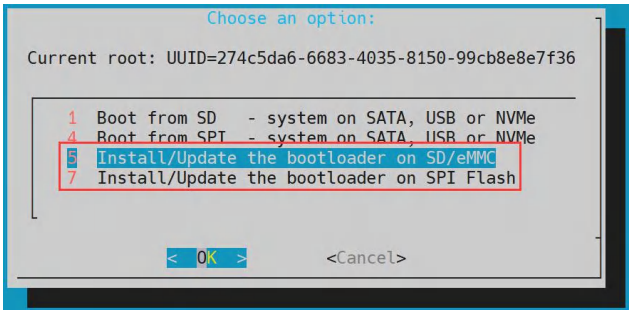
c. Install the new u-boot deb package just uploaded

```
root@orangepi:~# dpkg -i linux-u-boot-legacy-orangepi3b_1.0.0_arm64.deb
```

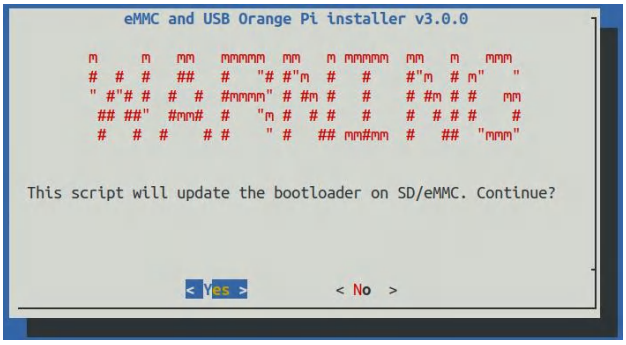
d. Then run the nand-sata-install script

```
root@orangepi:~# nand-sata-install
```

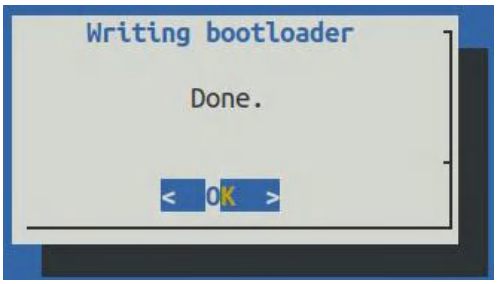
e. Then select **5 Install/Update the bootloader on SD/eMMC** to update the u-boot in the TF card or **7 Install/Update the bootloader on SPI Flash** to update the u-boot in the SPI Flash



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



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



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

10) Other useful information

a. In the u-boot 2017.09 source code, the defconfig configuration file used by the development board is

orangepi-build/u-boot/v2017.09-rk3588/configs/orangepi-3b-rk3566_defconfig

b. In the u-boot 2017.09 source code, the dts file used by the development board is

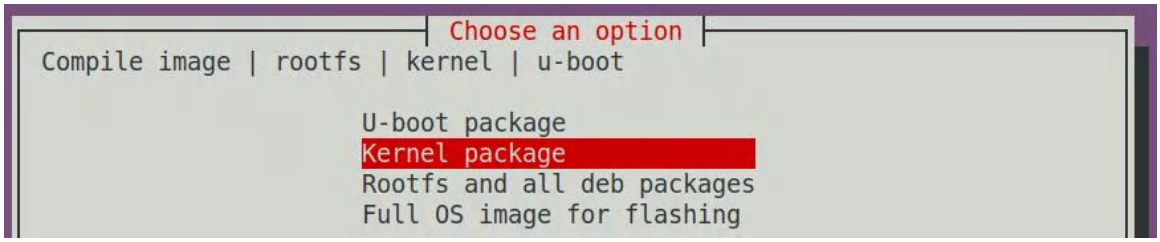
orangepi-build/u-boot/v2017.09-rk3588/arch/arm/dts/rk3566-orangepi-3b.dts

4. 4. Compile the Linux kernel

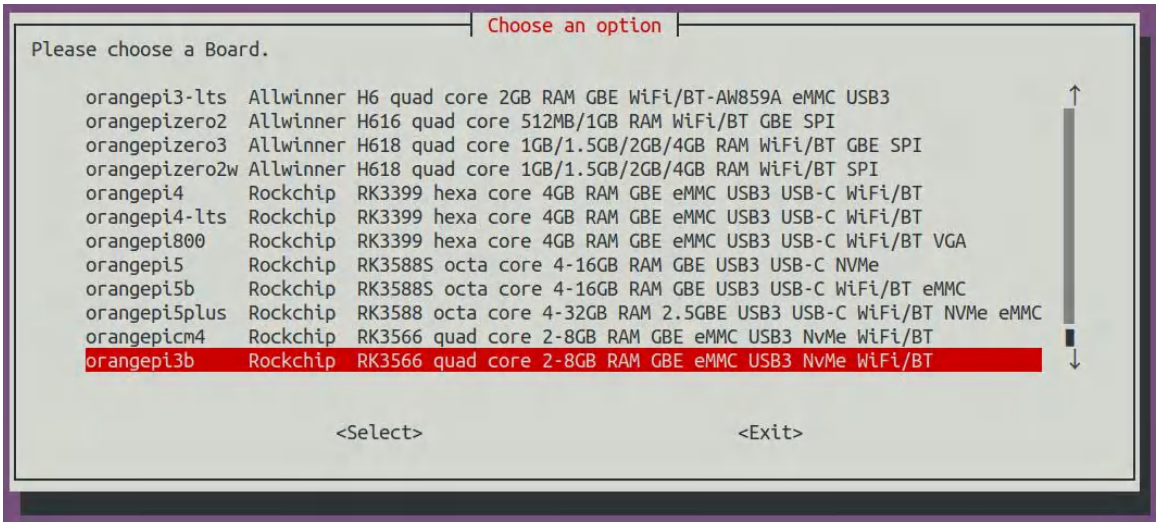
1) Run the build.sh script, remember to add sudo permission

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

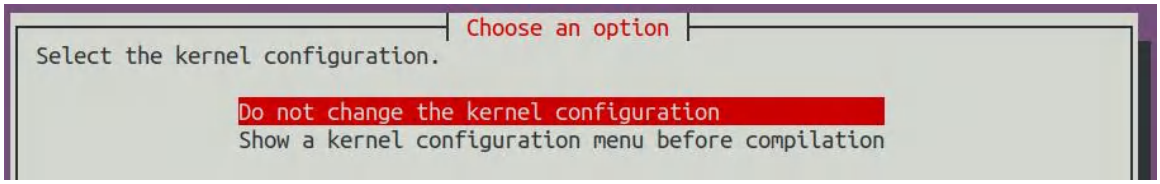
2) Select **Kernel package**，then enter



3) Then select the model of the development board



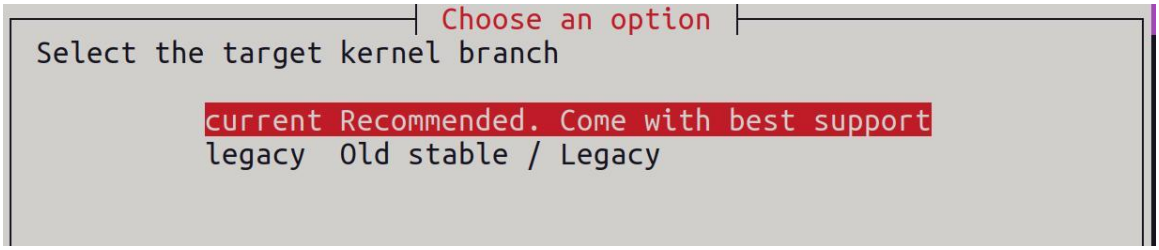
4) Then it will prompt whether to display the kernel configuration interface. If you do not need to modify the kernel configuration, select the first one. If you need to modify the kernel configuration, select the second one.



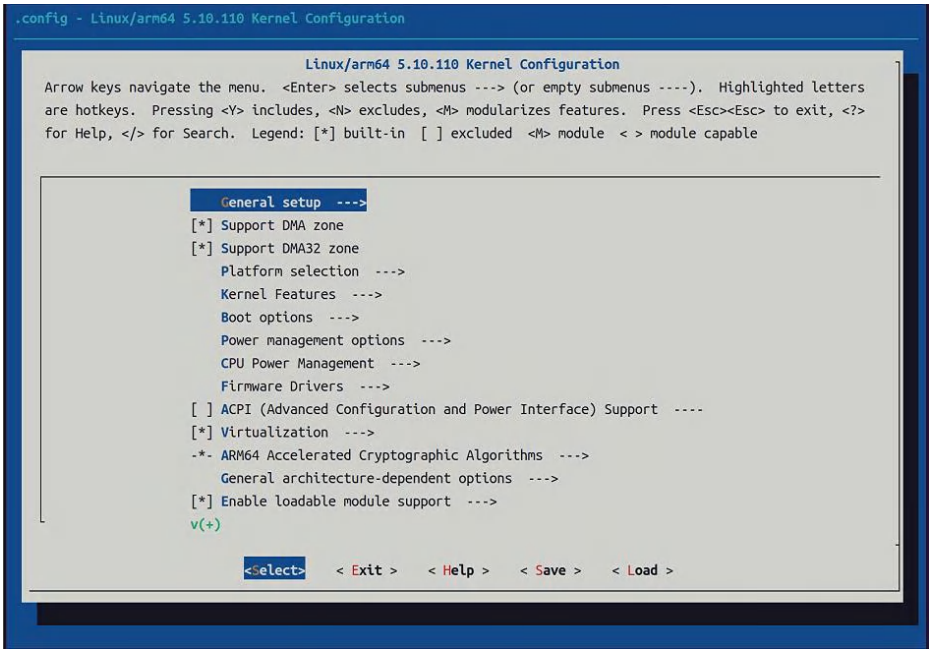
5) Then select the branch type of the kernel source code

a. The legacy branch will compile the linux5.10 kernel source code

b. The current branch will compile the linux6.6 kernel source code



6) If you choose to display the kernel configuration menu (the second option) in step 4, the kernel configuration interface opened by **make menuconfig** will pop up. At this time, you can directly modify the kernel configuration, save and exit after modification. Yes, after exiting, the kernel source code will be compiled



a. If you do not need to modify the configuration options of the kernel, when running the build.sh script, pass in **KERNEL_CONFIGURE=no** to temporarily 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 legacy branch kernel source code is explained as follows:

a. The version of the Linux kernel source code

[o.k.] Compiling current kernel [**5.10.160**]

b. The version of the cross-compilation toolchain used

[o.k.] Compiler version [**aarch64-none-linux-gnu-gcc 11.2.1**]

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

[o.k.] Using kernel config file [**config/kernel/linux-rockchip-rk356x-legacy.config**]

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

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

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

[o.k.] File name [**linux-image-legacy-rockchip-rk356x_1.0.0_arm64.deb**]

f. The time used for compilation

[o.k.] Runtime [**5 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=orangepi3b BRANCH=legacy BUILD_OPT=kernel KERNEL_CONFIGURE=no**]

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

a. **linux-dtb-legacy-rockchip-rk356x_1.0.0_arm64.deb** Contains dtb files used by the kernel

b. **linux-headers-legacy-rockchip-rk356x_1.0.0_arm64.deb** Include kernel header files

c. **linux-image-legacy-rockchip-rk356x_1.0.0_arm64.deb** Contains kernel images and kernel modules

```
test@test:~/orangepi-build$ ls output/debs/linux-*
output/debs/linux-dtb-legacy-rockchip-rk356x_1.0.0_arm64.deb
output/debs/linux-image-legacy-rockchip-rk356x_1.0.0_arm64.deb
output/debs/linux-headers-legacy-rockchip-rk356x_1.0.0_arm64.deb
```

9) The files contained in the generated Linux-image deb package are as follows

a. Use the following command to decompress the deb package

```
test@test:~/orangepi-build$ cd output/debs
test@test:~/orangepi_build/output/debs$ mkdir test
test@test:~/orangepi_build/output/debs$ cp \
linux-image-legacy-rockchip-rk356x_1.0.0_arm64.deb test/
test@test:~/orangepi_build/output/debs$ cd test
test@test:~/orangepi_build/output/debs/test$ dpkg -x \
linux-image-legacy-rockchip-rk356x_1.0.0_arm64.deb .
test@test:~/orangepi_build/output/debs/test$ ls
boot  etc  lib  linux-image-legacy-rockchip-rk356x_1.0.0_arm64.deb  usr
```

b. The decompressed file is as follows

```
test@test:~/orangepi-build/output/debs/test$ tree -L 2
.
├── boot
│   ├── config-5.10.160-rockchip-rk356x
│   ├── System.map-5.10.160-rockchip-rk356x
│   └── vmlinuz-5.10.160-rockchip-rk356x
├── etc
│   └── kernel
├── lib
│   └── modules
├── linux-image-legacy-rockchip-rk356x_1.0.0_arm64.deb
└── usr
    ├── lib
    └── share
```

10) The orangepi-bulid compilation system will first synchronize the Linux kernel

source code with the Linux kernel source code of the github server when compiling the Linux kernel source code, so if you want to modify the Linux kernel source code, you first need to turn off the update function of the source code **(You need to fully compile the Linux kernel source code before turning off this function. Otherwise, you will be prompted that the source code of the Linux kernel cannot be found. If you download the source code package from Baidu cloud disk, there is no such problem, because the source code of Linux has been cached.)**, otherwise the The changes made will be reverted 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"
```

11) 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-legacy-rockchip-rk356x_1.0.0_arm64.deb root@192.168.1.xxx:/root
```

b. Then log in to the development board and uninstall the deb package of the installed Linux kernel

```
root@orangepi:~# apt purge -y linux-image-legacy-rockchip-rk356x
```

c. Install the deb package of the new Linux kernel just uploaded

```
root@orangepi:~# dpkg -i linux-image-legacy-rockchip-rk356x_1.0.0_arm64.deb
```

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

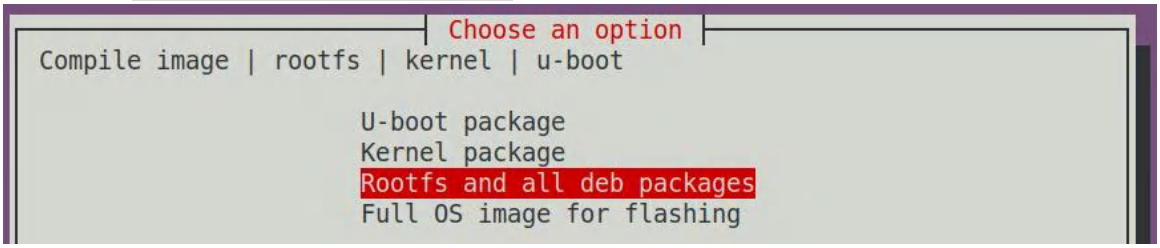
```
root@orangepi:~# reboot
```

4. 5. Compile rootfs

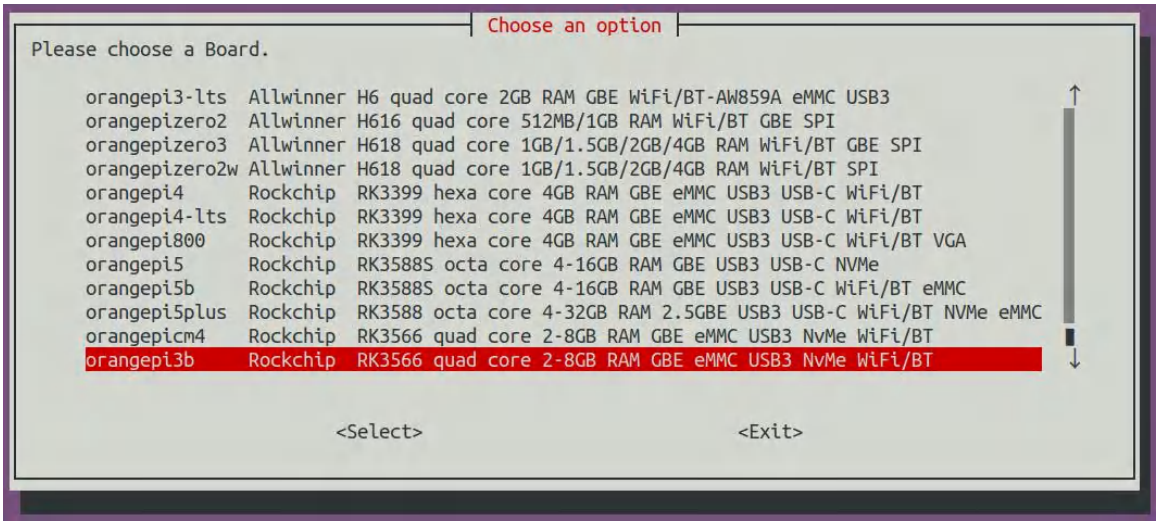
1) Run the build.sh script, remember to add sudo permission

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

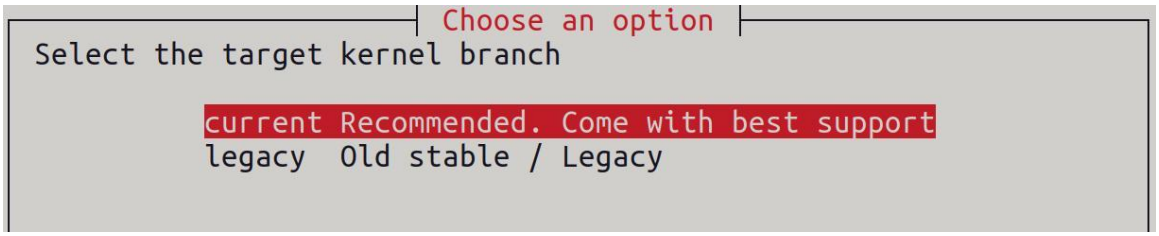
2) Select **Rootfs and all deb packages**，then enter



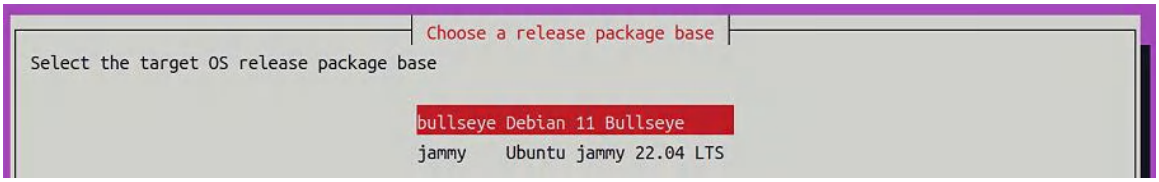
3) Then select the model of the development board



4) Then select the branch type of the kernel source code. Currently, the rootfs maintained by the kernel source code uses the same set.



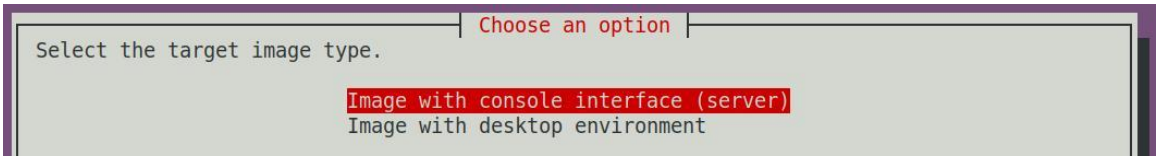
5) Then select the type of rootfs



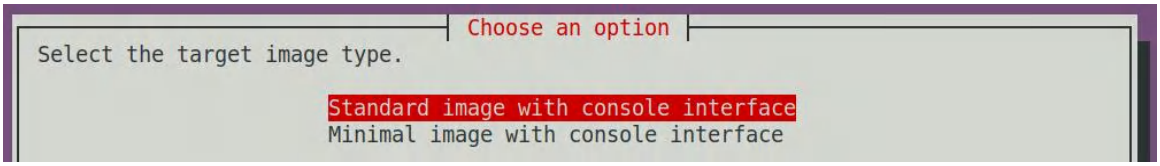
6) Then select the type of image

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

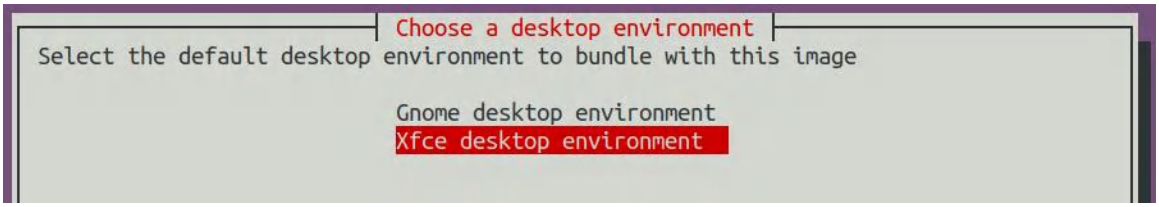
b. **Image with desktop environment** Indicates a image with a desktop, which is relatively large

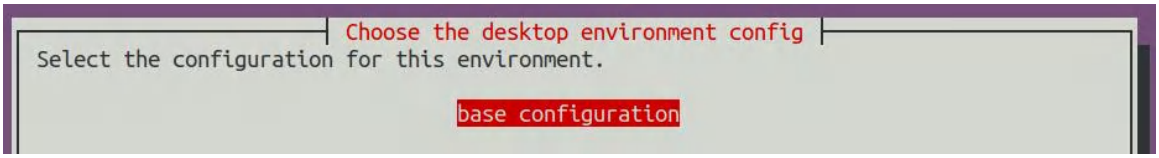


7) If you are compiling the image of the server version, you can also choose to compile the Standard version or the Minimal version. The pre-installed software of the Minimal version will be much less than that of the Standard version **(please do not choose the Minimal version if there is no special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

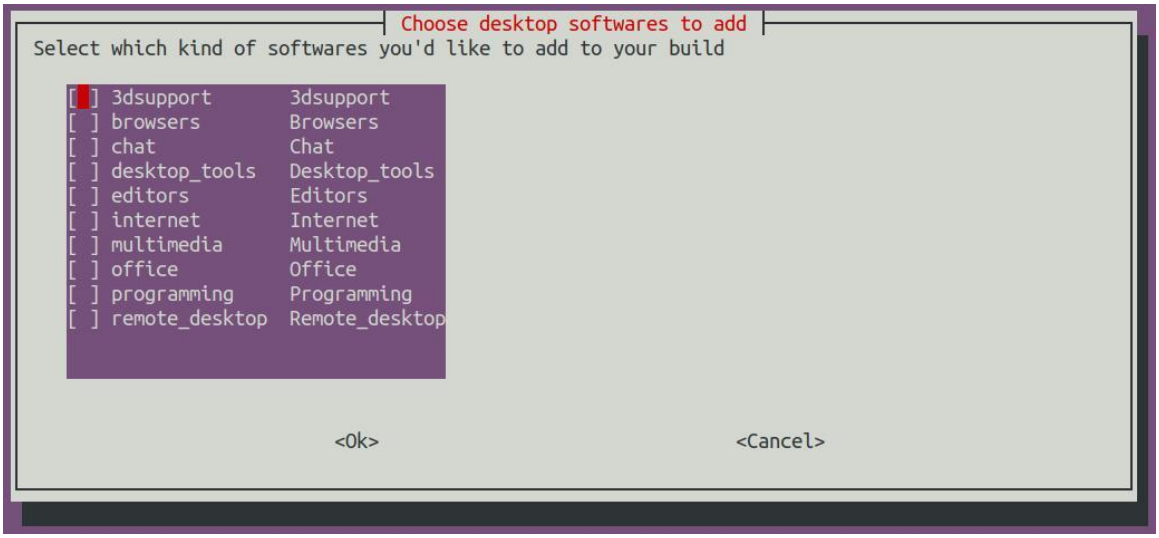


8) If you are compiling the image of the desktop version, you also need to select the type of desktop environment. Currently, Ubuntu Jammy mainly maintains XFCE and Gnome desktops, Ubuntu Focal only maintains XFCE desktops, and Debian Bullseye mainly maintains XFCE and KDE desktops





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 of the information prompted during compilation are as follows

a. The type of rootfs

[o.k.] local not found [Creating new rootfs cache for **jammy**]

b. The storage path of the compiled rootfs compressed package

[o.k.] Target directory [**external/cache/rootfs**]

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

[o.k.] File name [**jammy-xfce-arm64.f930ff6ebbac1a72108a2e100762b18f.tar.lz4**]

d. The time used for compilation

[o.k.] Runtime [**13 min**]

10) View the rootfs compressed package generated by compilation

a. **jammy-xfce-arm64.f930ff6ebbac1a72108a2e100762b18f.tar.lz4** is the rootfs compressed package, the meaning of each field of the name is

a) **jammy** 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) **f930ff6ebbac1a72108a2e100762b18f** 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. **jammy-xfce-arm64.f930ff6ebbac1a72108a2e100762b18f.tar.lz4.list** lists the package names of all packages installed by rootfs

```
test@test:~/orangepi-build$ ls external/cache/rootfs/
jammy-xfce-arm64.f930ff6ebbac1a72108a2e100762b18f.tar.lz4
jammy-xfce-arm64.f930ff6ebbac1a72108a2e100762b18f.tar.lz4.current
jammy-xfce-arm64.f930ff6ebbac1a72108a2e100762b18f.tar.lz4.list
```

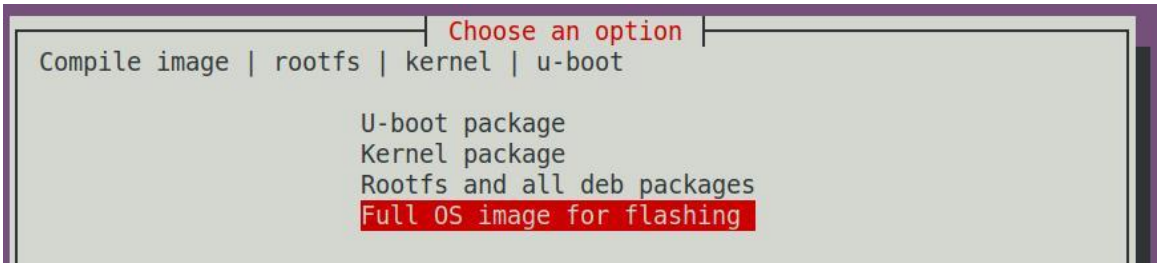
11) If the required rootfs already exists under **external/cache/rootfs**, then compiling rootfs again will directly skip the compilation process and will not restart the compilation. When compiling the image, it will also go to **external/cache/rootfs** to find out whether it has If there is rootfs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

4. 6. Compile Linux image

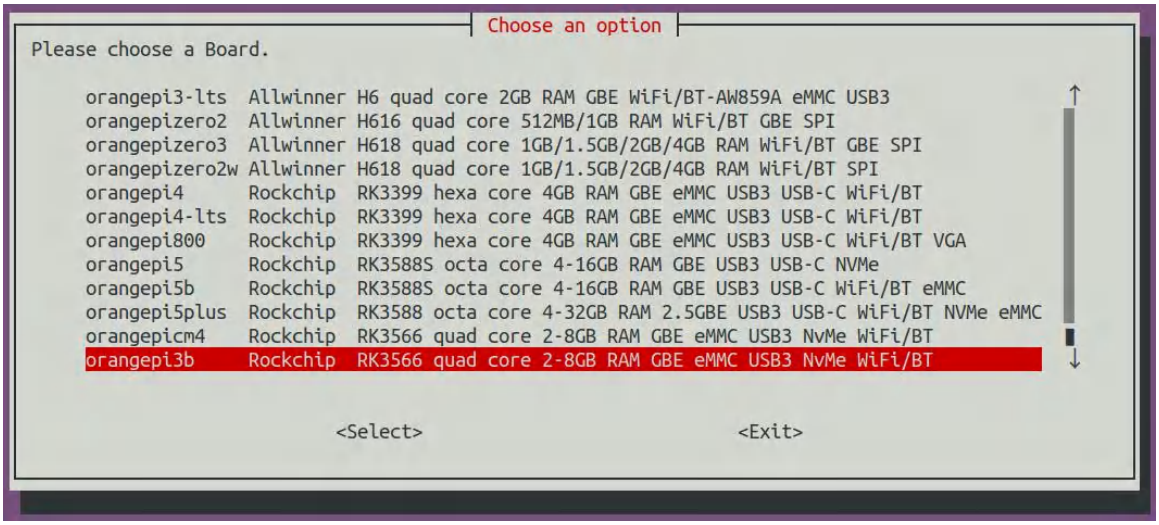
1) Run the build.sh script, remember to add sudo permission

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

2) Select **Full OS image for flashing**，then enter



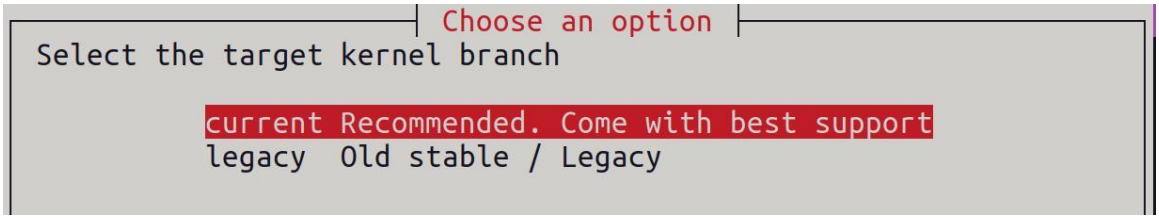
3) Then select the model of the development board



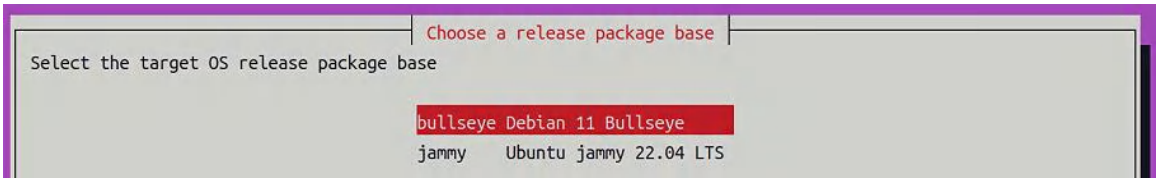
4) Then select the branch type of the kernel source code

a. The legacy branch will compile the linux5.10 kernel source code

b. The current branch will compile the linux6.6 kernel source code



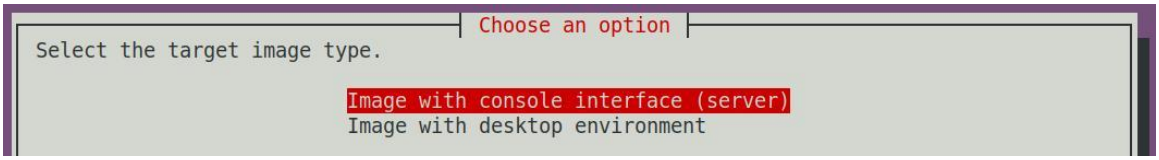
5) Then select the type of rootfs



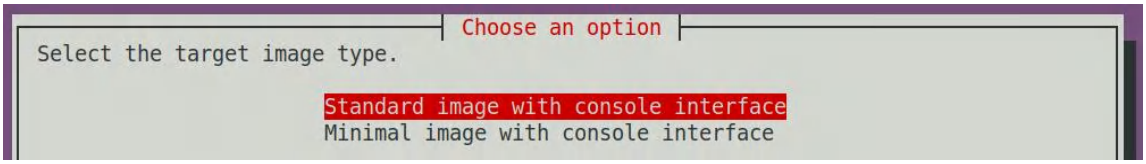
6) Then select the type of image

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

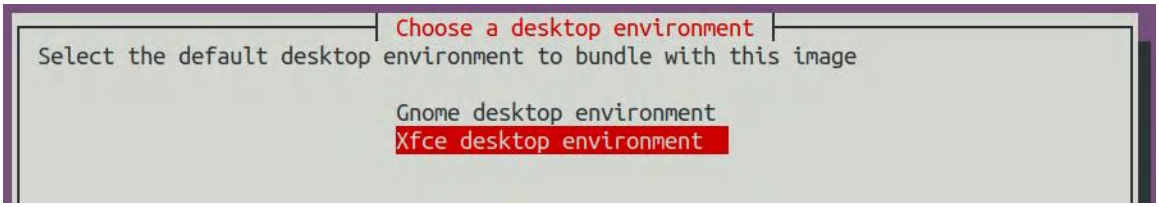
b. **Image with desktop environment** Indicates a image with a desktop, which is relatively large

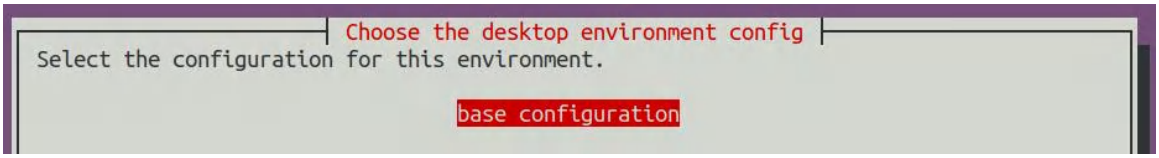


7) If you are compiling the image of the server version, you can also choose to compile the Standard version or the Minimal version. The pre-installed software of the Minimal version will be much less than that of the Standard version **(please do not choose the Minimal version if there is no special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

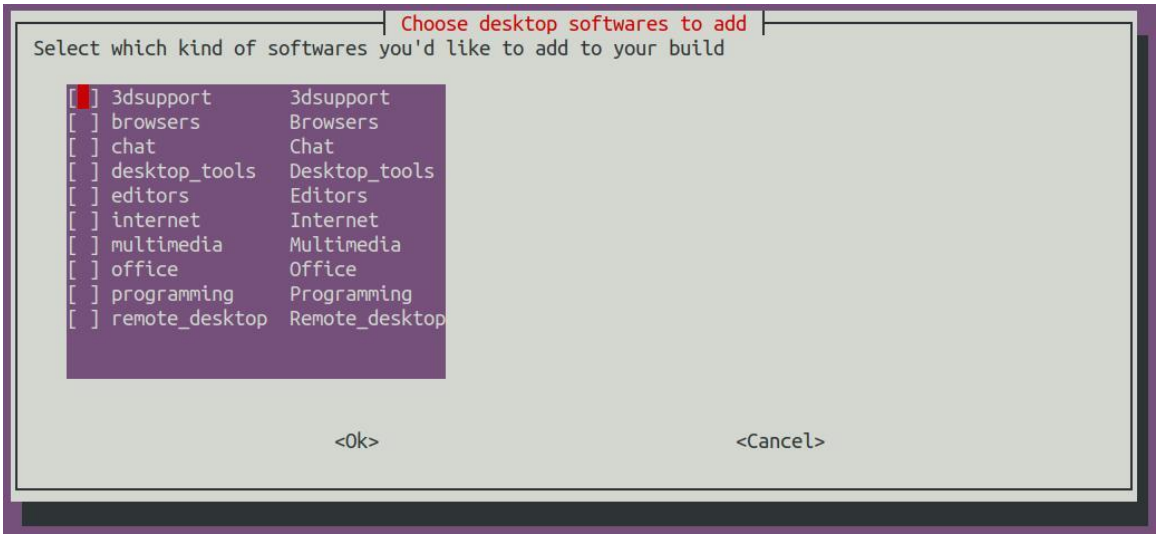


8) If you are compiling the image of the desktop version, you also need to select the type of desktop environment. Currently, Ubuntu Jammy mainly maintains XFCE and Gnome desktops, Ubuntu Focal only maintains XFCE desktops, and Debian Bullseye mainly maintains XFCE and KDE desktops





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 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

10) After compiling the image, the following information will be prompted

a. The storage path of the compiled image

[o.k.] Done building [**output/images/Orangepi3b_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160/Orangepi3b_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160.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=orangepi3b BRANCH=legacy BUILD_OPT=image RELEASE=bullseye BUILD_MINIMAL=no BUILD_DESKTOP=no KERNEL_CONFIGURE=yes**]

5. Instructions for using the Orange Pi OS Arch system

5.1. Orange Pi OS Arch system function adaptation

Function	OPi OS Arch
USB2.0x3	OK
USB3.0x1	OK
SPIFlash+M.2 NVMe SSD Boot	OK
WIFI	OK
Bluetooth	OK
GPIO (40pin)	OK
UART (40pin)	OK
SPI (40pin)	OK

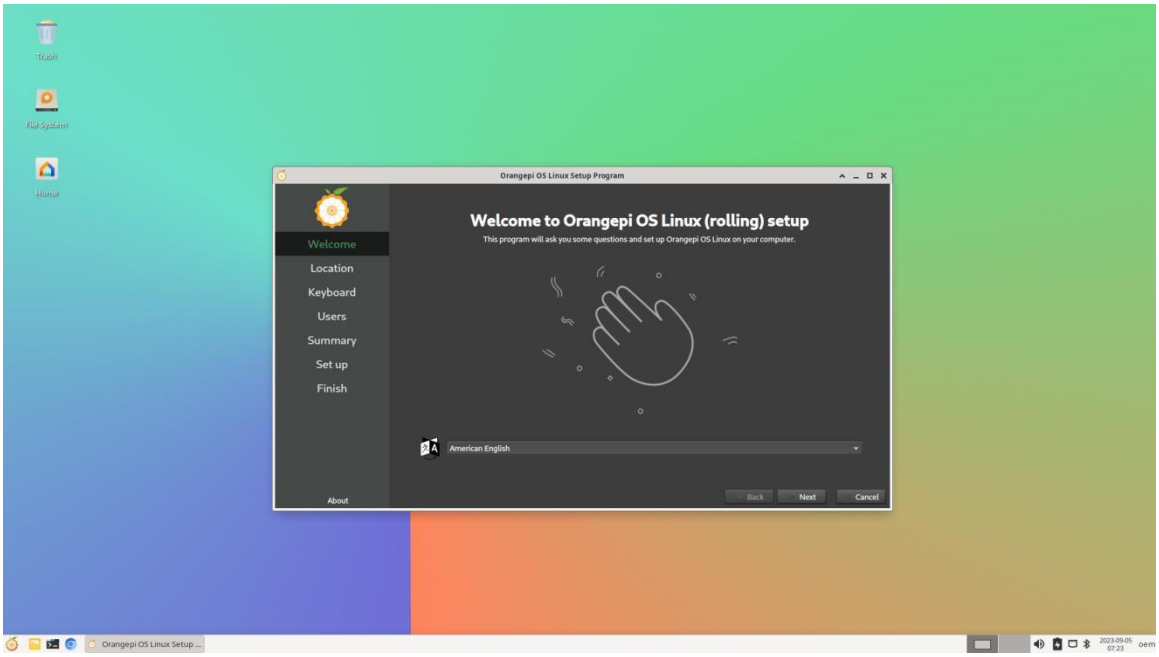
I2C (40pin)	OK
PWM (40pin)	OK
3pin debugging serial port	OK
eMMC start	OK
TF card start	OK
HDMI video	OK
HDMI audio	OK
Raspberry Pi 5 inch screen display	OK
Raspberry Pi 5-inch screen touch function	OK
eDP display	OK
OV5647 camera	The kernel driver is OK, 3A is not adjusted
Gigabit Ethernet port	OK
Network port status light	OK
headphone playback	OK
headphone recording	OK
LED lights	OK
GPU	NO
NPU	NO
VPU	NO

5.2. Orange Pi OS Arch System User Guide Instructions

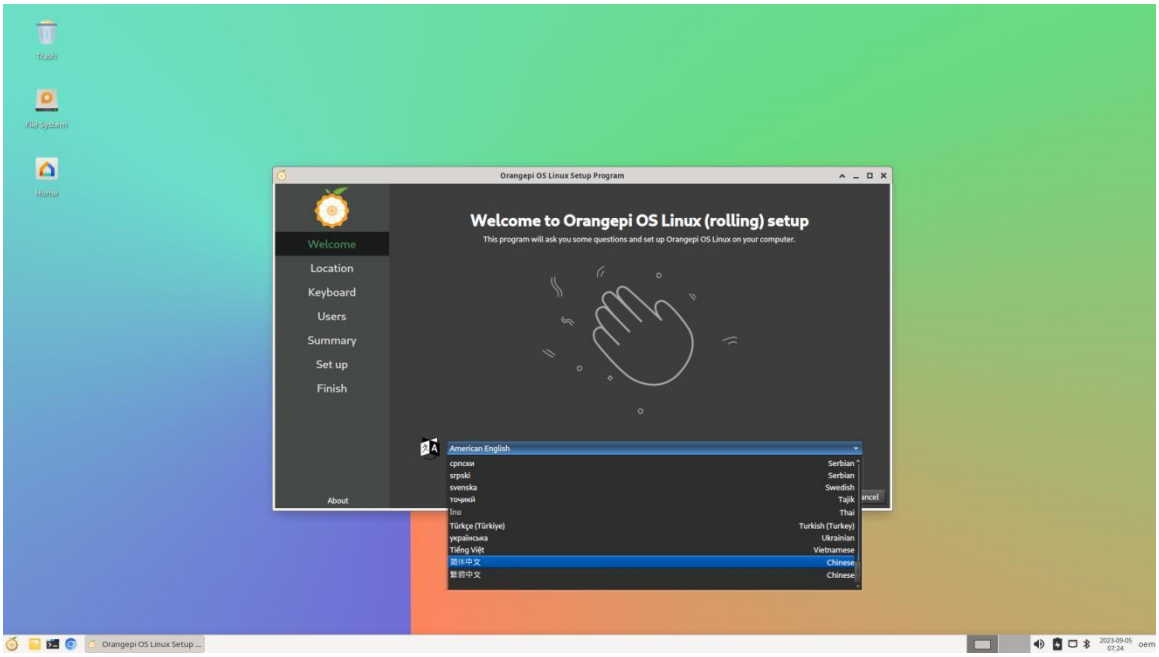
First of all, please note that the OPi OS Arch system does not have a default orangepi user and password, so after burning and starting the system, it is impossible to log in remotely through the serial port and ssh directly (not even the root user). This is different from Ubuntu and Debian systems.

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:

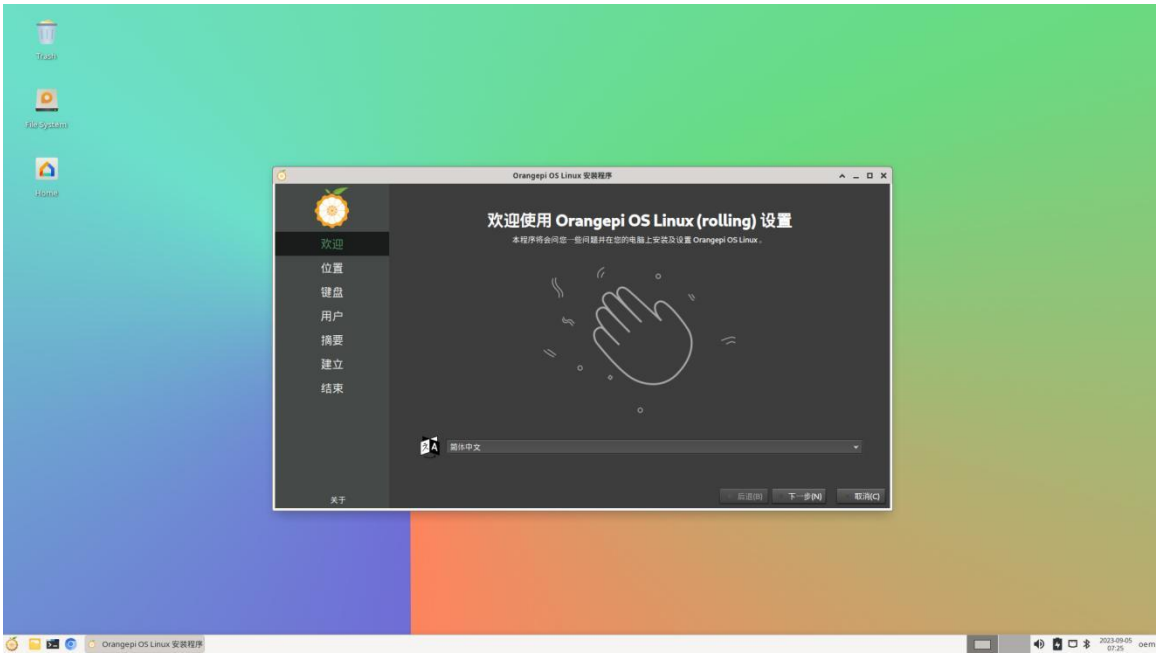
a) After burning the system for the first time and enter the desktop, you will see the user wizard program shown in the figure below



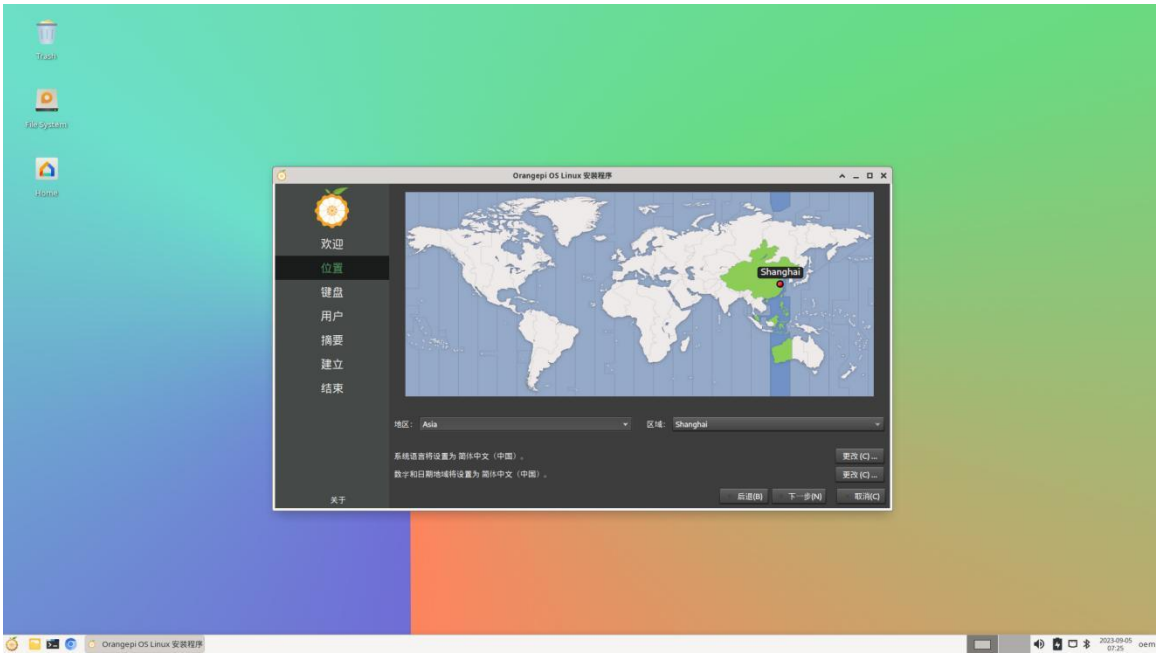
b) First you need to choose the desired language



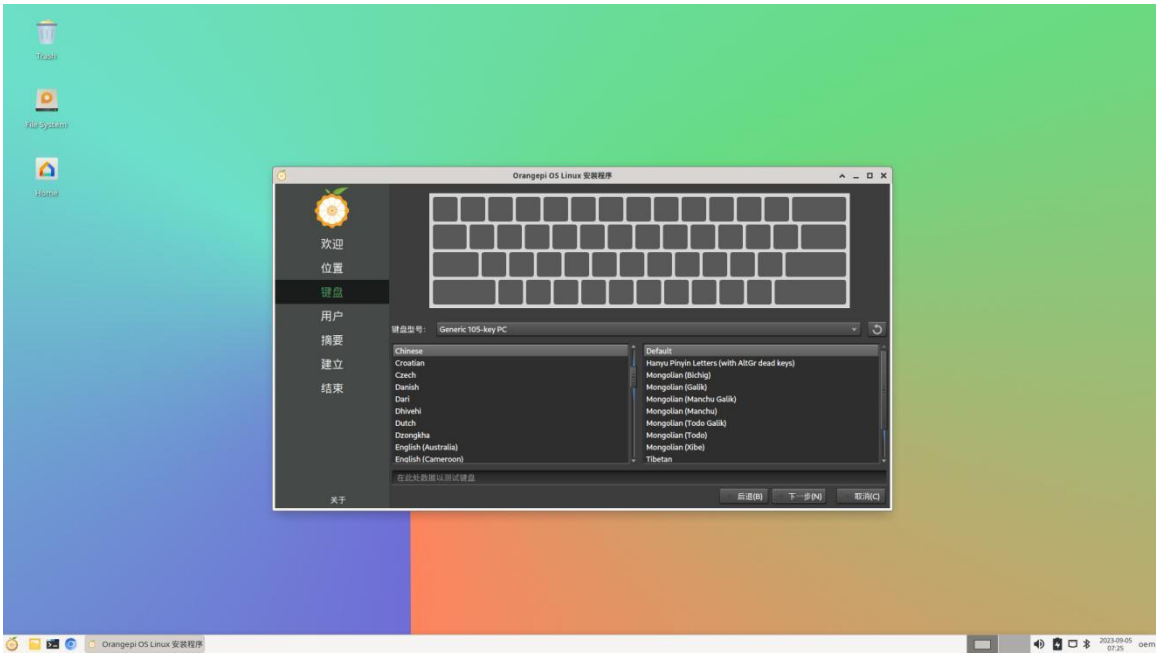
c) After selecting the language, the user guide will immediately switch to the corresponding language interface, such as the Chinese display as shown below



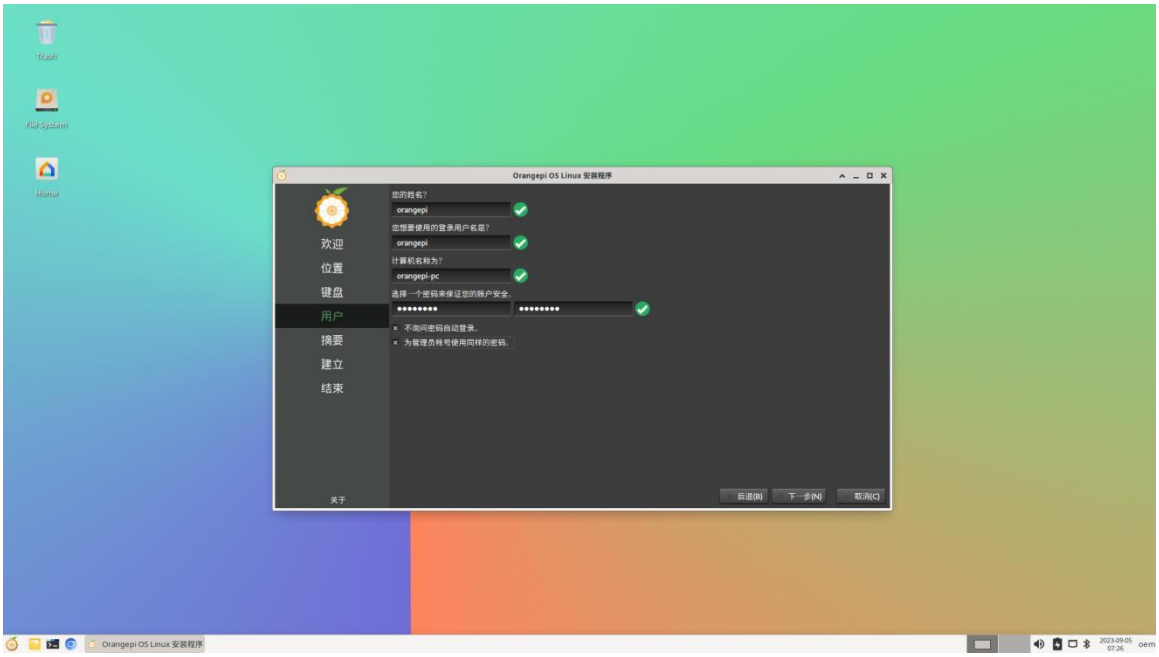
d) Then select the area



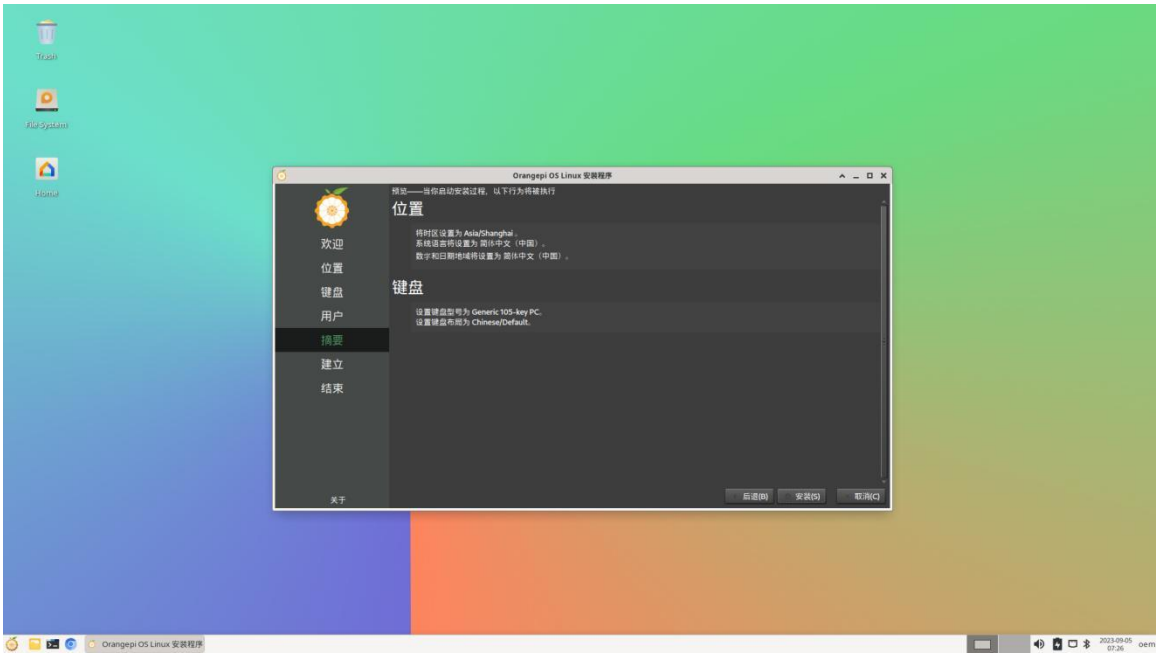
e) Then select the keyboard model



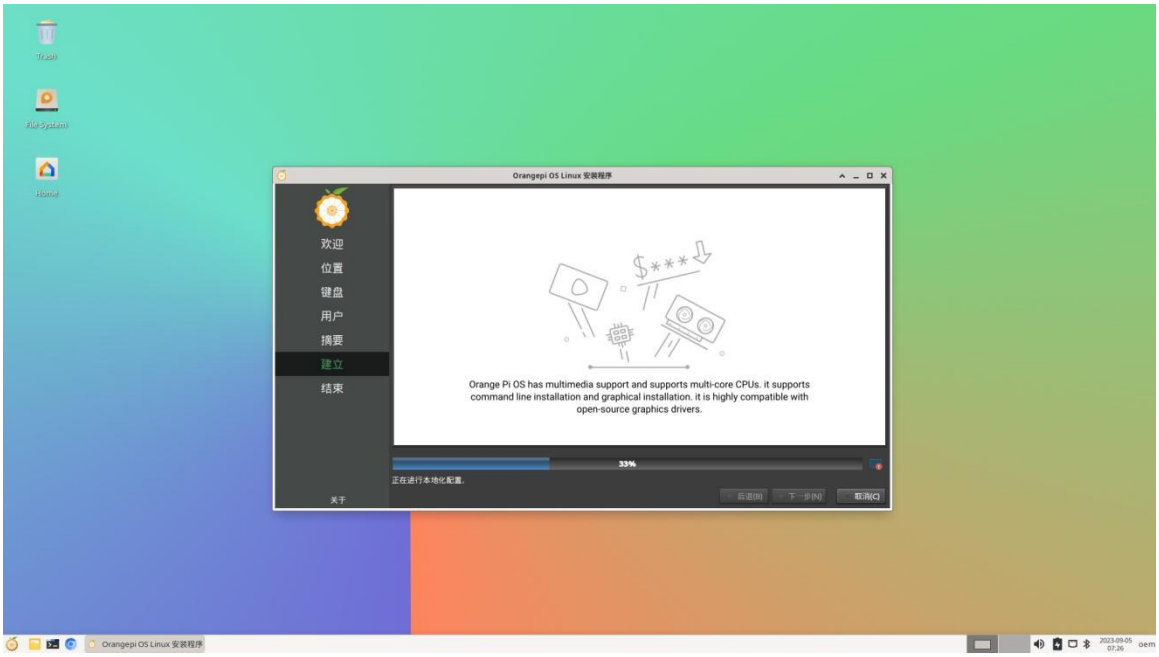
f) Then create a new user name and set a password



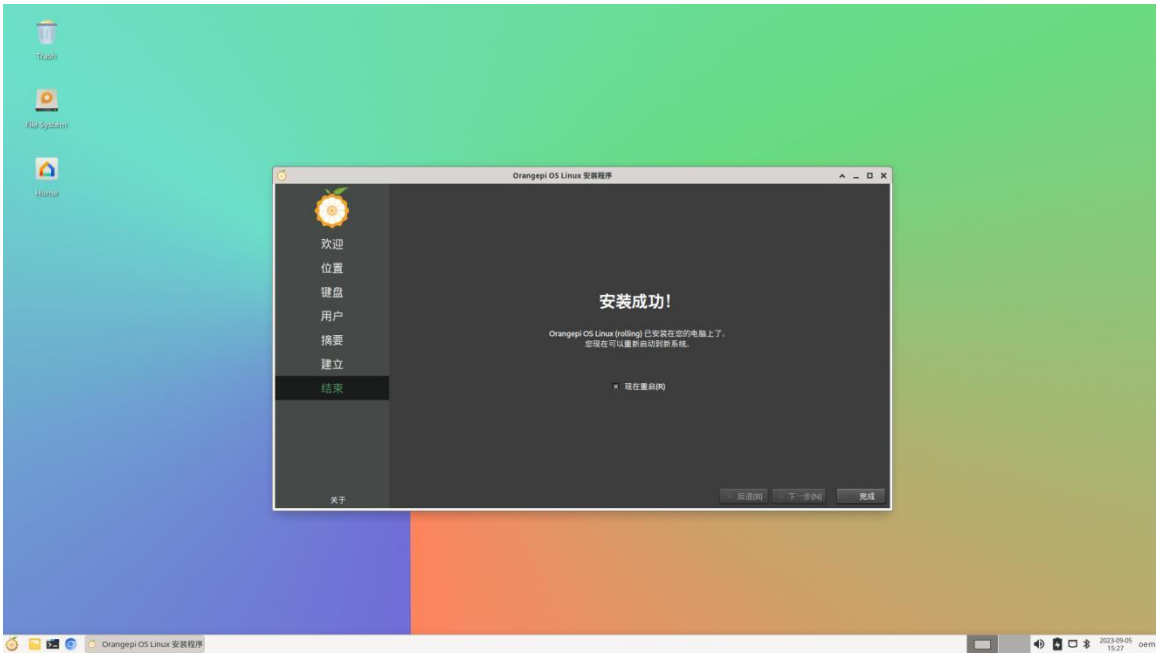
g) Then make sure that there is no problem with the selection, and then click the **install** button



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 Orange Pi 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 Orange Pi 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

LCD MIPI screen, eDP screen, and multiplexing functions such as I2C/SPI/UART/PWM in 40pin are disabled by default in the dts of the kernel, and the corresponding DT overlays need to be manually enabled 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/rockchip/overlay/xxx.dtbo** in **/boot/extlinux/extlinux.conf**

Note that xxx.dtbo in FDTOVERLAYS /dtbs/rockchip/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 OPIOS ARM
LINUX /Image
FDT /dtbs/rockchip/rk3566-orangepi-3b.dtb
FDTOVERLAYS /dtbs/rockchip/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/rockchip/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
EDP screen	rk356x-edp.dtbo
Raspberry Pi 5 inch screen	rk356x-raspi-7inch-touchscreen.dtbo
40pin expansion interface - I2C2	rk356x-i2c2-m1.dtbo

40pin expansion interface - I2C3	**rk356x-i2c3-m0.dtbo**
40pin expansion interface - I2C4	**rk356x-i2c4-m0.dtbo**
40pin extension interface - PWM11	**rk356x-pwm11-m1.dtbo**
40pin extension interface - PWM15	**rk356x-pwm15-m1.dtbo**
40pin expansion interface - UART3	**rk356x-uart3-m0.dtbo**
40pin expansion interface - UART7	**rk356x-uart7-m2.dtbo**
40pin expansion interface - UART9	**rk356x-uart9-m2.dtbo**
40pin expansion interface - SPI3	**rk356x-spi3-m0-cs0-spidev.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 i2c2 and pwm11 at the same time are as follows

```
[orangepi@orangepi-pc ~]$ sudo vim /boot/extlinux/extlinux.conf
LABEL OPIOS ARM
LINUX /Image
FDT /dtbs/rockchip/rk3566-orangepi-3b.dtb
FDTOVERLAYS /dtbs/rockchip/overlay/rk356x-i2c2-m1.dtbo  /dtbs/rockchip/overlay/rk356x-pwm11-m1.dtbo
```

6) After setting, you need to restart the system to make the configuration take effect

```
[orangepi@orangepi-pc ~]$ sudo reboot
```

5. 4. Use of Raspberry Pi 5-inch screen

5. 4. 1. How to assemble the Raspberry Pi 5-inch screen

Please refer to **the assembly method of the Raspberry Pi 5-inch screen** (**click the text in the blue part to jump to the corresponding position**).

5. 4. 2. How to open Raspberry Pi 5-inch screen configuration

By default, OPi OS Arch mirroring does not enable the configuration of the Raspberry Pi 5-inch screen. If you need to use the Raspberry Pi 5-inch screen, you need to manually open it. The method to open the configuration is as follows:

a. First add the following configuration in **/boot/extlinux/extlinux.conf**

```
[orangepi@orangepi-pc ~]$ sudo vim /boot/extlinux/extlinux.conf
LABEL OPIOS ARM
LINUX /Image
FDT /dtbs/rockchip/rk3566-orangepi-3b.dtb
FDTOVERLAYS /dtbs/rockchip/overlay/rk356x-raspi-7inch-touchscreen.dtbo #Configuration that needs to be added
```

b. Then restart the system

```
[orangepi@orangepi-pc ~]$ sudo reboot
```

After restarting, you can see the display on the LCD screen as follows:

5. 5. How to use the eDP screen

5. 5. 1. Assembly method of eDP screen

Please refer to how to use the eDP screen (**click the text in the blue part to jump to**

the corresponding position).

5.5.2. How to open eDP screen configuration

The OPi OS Arch image does not enable the eDP screen configuration by default. If you want to use the eDP screen, you need to manually open it. The method to open the configuration is as follows:

a. First add the following configuration in **/boot/extlinux/extlinux.conf**

```
[orangepi@orangepi-pc ~]$ sudo vim /boot/extlinux/extlinux.conf
LABEL OPIOS ARM
LINUX /Image
FDT /dtbs/rockchip/rk3566-orangepi-3b.dtb
FDTOVERLAYS /dtbs/rockchip/overlay/rk356x-edp.dtbo #Configuration that needs to be added
```

b. Then restart the system

```
[orangepi@orangepi-pc ~]$ sudo reboot
```

After restarting, you can see that the display of the eDP screen is as follows:

5.6. 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. Orange Pi OS OH system usage instructions

6.1. Orange Pi OS OH system function adaptation status

功能	OPi OS OH
USB2.0x3	OK
USB3.0x1	OK
SPIFlash+M.2 NVMe SSD Boot	NO
WIFI	OK
Bluetooth	NO
3pin Debug serial port	OK
eMMC start	OK
TF card start	OK
HDMI video	OK
HDMI audio	NO
Raspberry Pi 5 inch screen display	NO
Raspberry Pi 5-inch screen touch function	NO
eDP display	NO
OV5647 camera	NO
Gigabit Ethernet port	OK
Network port status light	OK
Network port status light	OK
headphone recording	NO
LED lights	OK
GPU	OK
NPU	NO
VPU	NO

Currently, the 8GB memory version of the development board can only use 4GB

of memory in the OPi OS OH system. Please pay special attention to this.

6.2. How to use the Gigabit Ethernet port

1) First use a network cable to connect the development board and router

2) Then use the ifconfig command in the debugging serial port to see the IP address assigned by the router to the development board network port.

```
# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 4e:fc:9d:f3:67:26  Driver rk_gmac-dwmac
          inet addr:192.168.1.189  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::4cfc:9dff:fef3:6726/64 Scope: Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:378 errors:0 dropped:0 overruns:0 frame:0
          TX packets:24 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:30663 TX bytes:2904
          Interrupt:45

#
```

3) Then use the ping command to test whether the network can be used normally

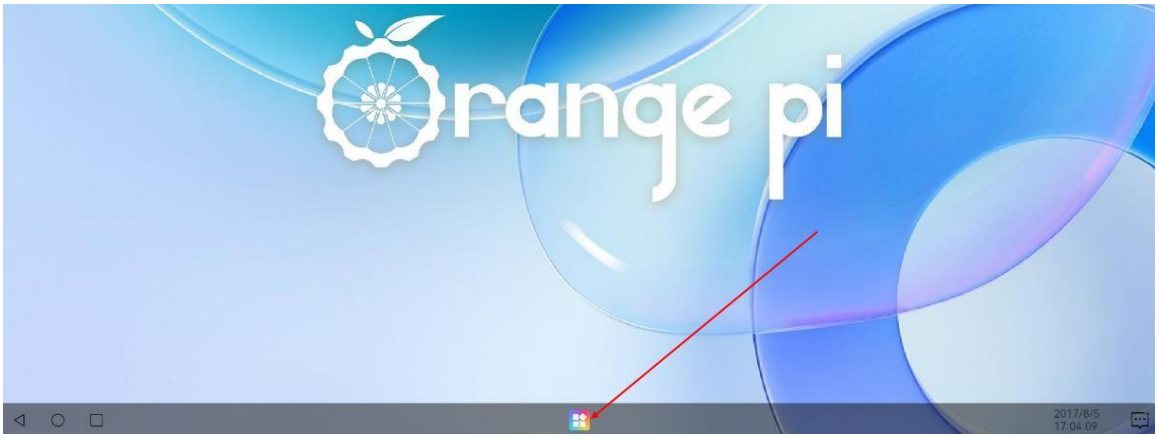
```
# ping www.orangepi.cn
Ping www.orangepi.cn (123.57.147.237): 56(84) bytes.
64 bytes from 123.57.147.237: icmp_seq=1 ttl=0 time=42 ms
64 bytes from 123.57.147.237: icmp_seq=2 ttl=0 time=43 ms
64 bytes from 123.57.147.237: icmp_seq=3 ttl=0 time=44 ms

--- 123.57.147.237 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss
round-trip min/avg/max = 0/0/43 ms
#
```

4) In addition to checking the IP address of the network port in the command line, you can also check the IP address of the network port in the OH settings. The method is

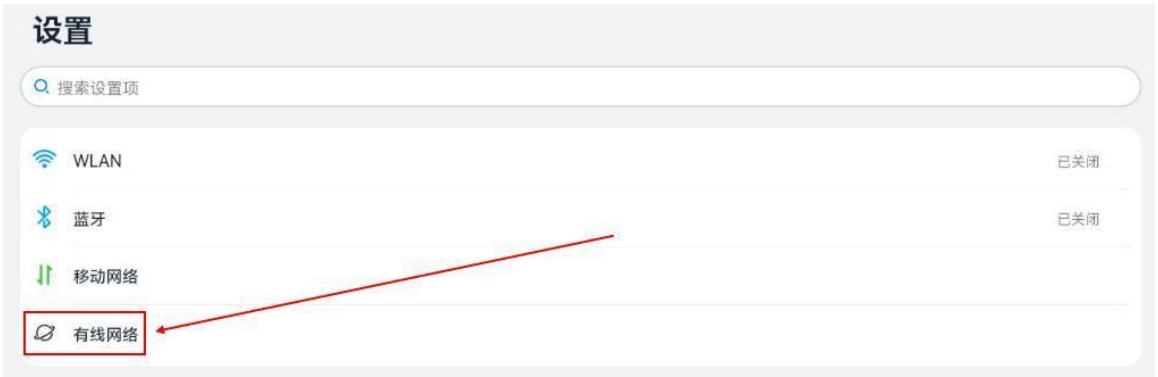
as follows

a. First click on the **application list**



b. Then open **settings**

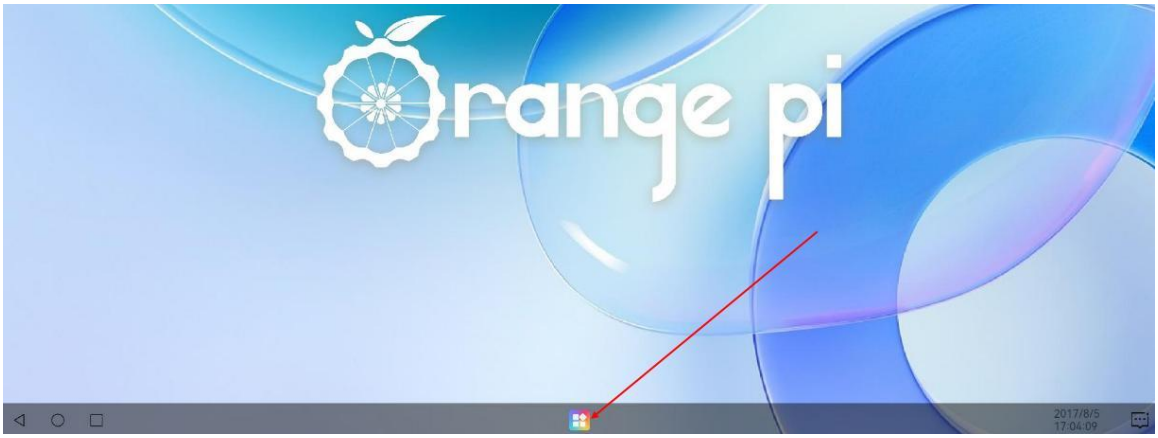
c. Then select **wired network**



d. Then you can see the IP address of the network port and other information.

How to use WIFI

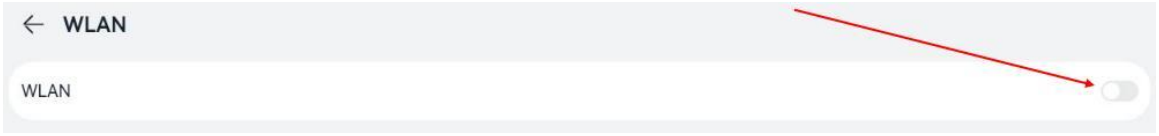
1) First click on the **application list**



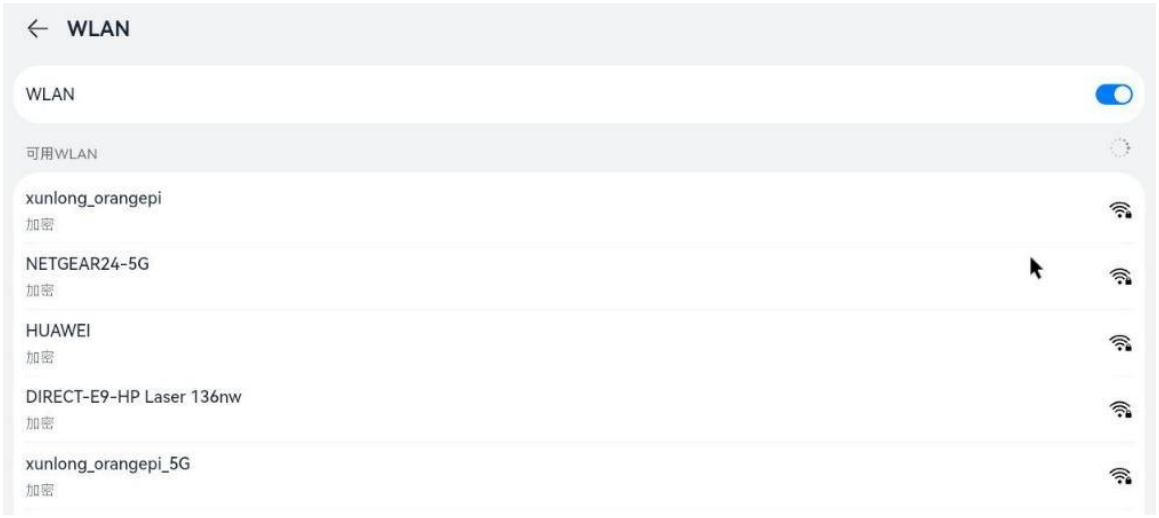
2) Then open **settings**

3) Then select **WLAN**

4) Then turn on **WLAN**



5) Then connect to the searched WIFI hotspot



7. Android 11 operating system instructions

7.1. Supported Android versions

Android version	Kernel version
Android 11	Linux4.19

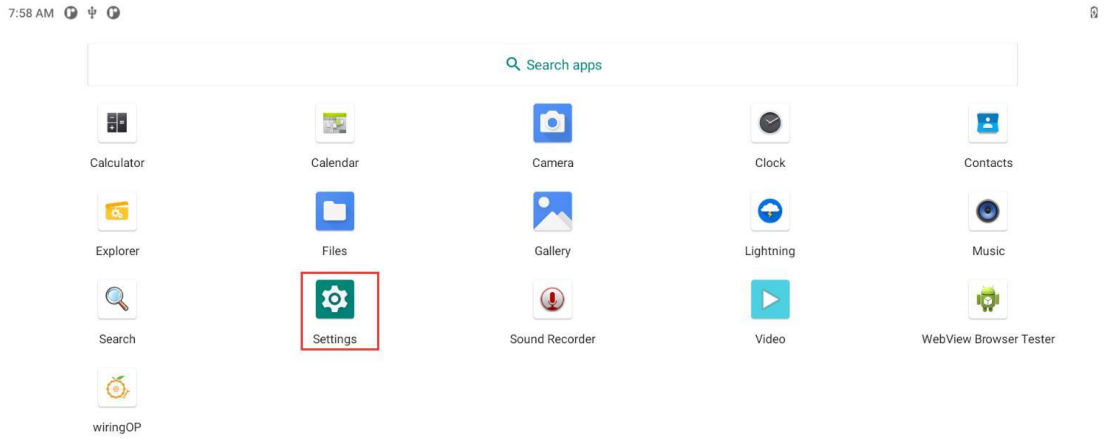
7.2. Android Function Adaptation

Functions	Android 11
USB2.0x3	OK
USB3.0x1	OK
M.2 NVMe SSD boot	OK
WIFI	OK
Bluetooth	OK
GPIO（40pin）	OK
UART（40pin）	OK
SPI（40pin）	OK
I2C（40pin）	OK
PWM（40pin）	OK
PWM fan interface	OK
3pin Debugging serial port	OK
EMMC	OK
TF card boot	OK
HDMI video	OK
HDMI Audio	OK
LCD	OK
eDP display	OK
OV5647 Camera	The kernel driver is OK, 3A is not adjusted
Gigabit network port	OK
Network port status indicator	OK

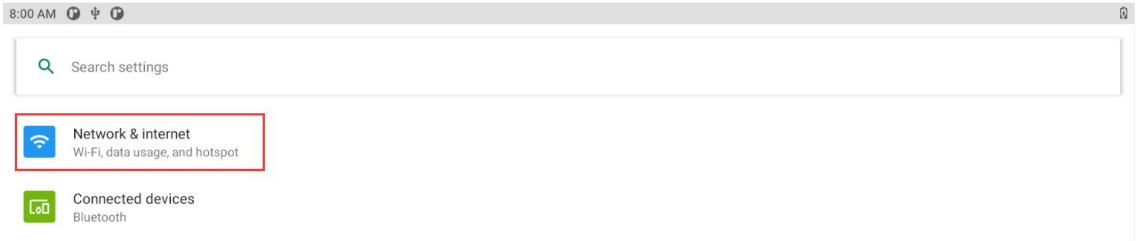
Headphone playback	OK
Headphone recording	OK
LED Light	OK
GPU	OK
NPU	OK
VPU	OK
RTC	OK

7. 3. WIFI connection test method

1) First click enter **Setting**

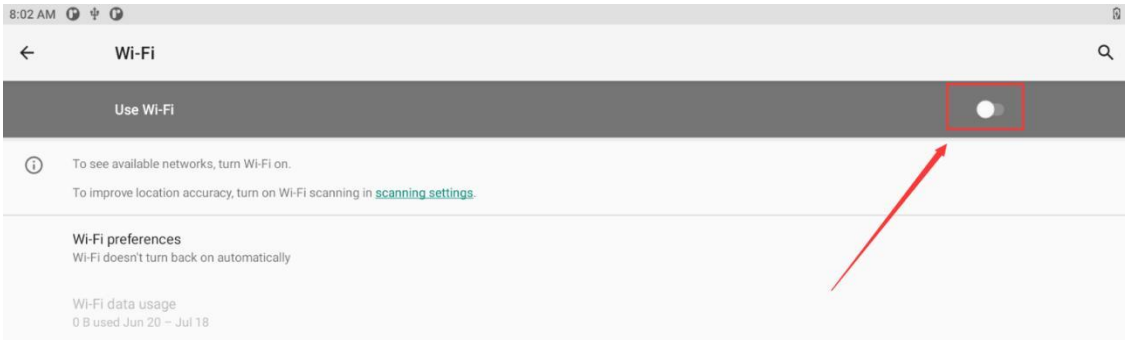


2) Then select **Network & internet**



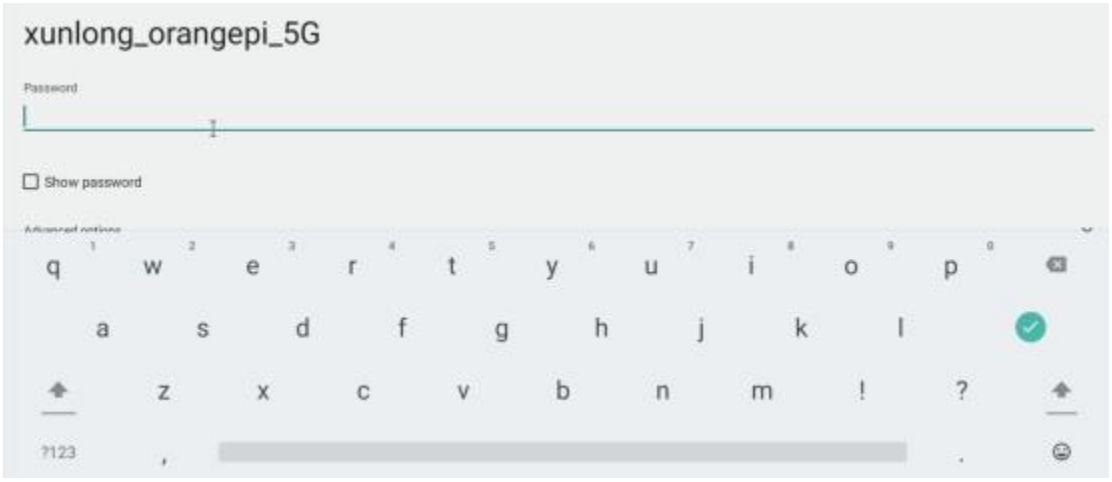
3) Then select **Wi-Fi**

4) Then turn on the **Wi-Fi** switch

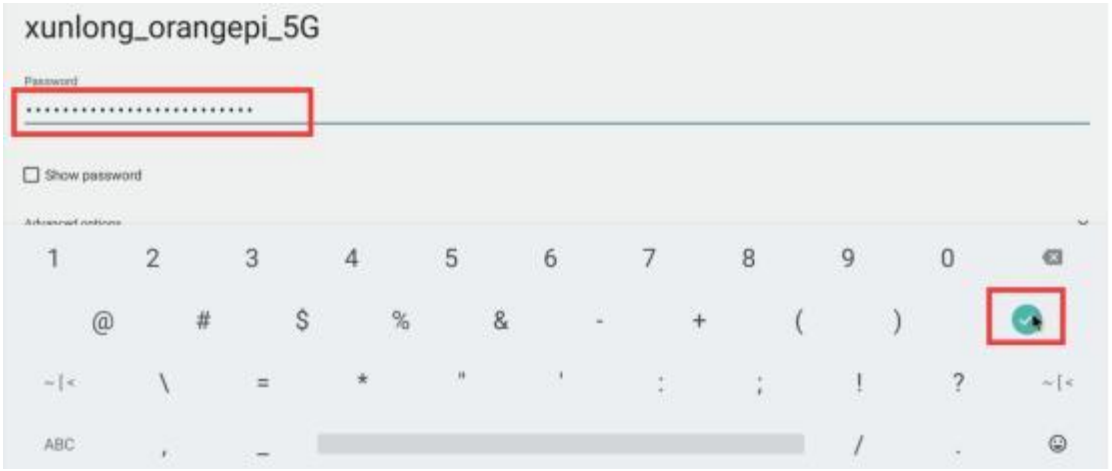


5) After turning on **Wi-Fi**, if everything is normal, you can scan for nearby Wi-Fi hotspots

6) Then select the Wi-Fi you want to connect to, and the password input interface shown in the figure below will pop up



7) Then use the keyboard to enter the password corresponding to Wi-Fi, and then use the mouse to click the Enter button in the virtual keyboard to start connecting to Wi-Fi

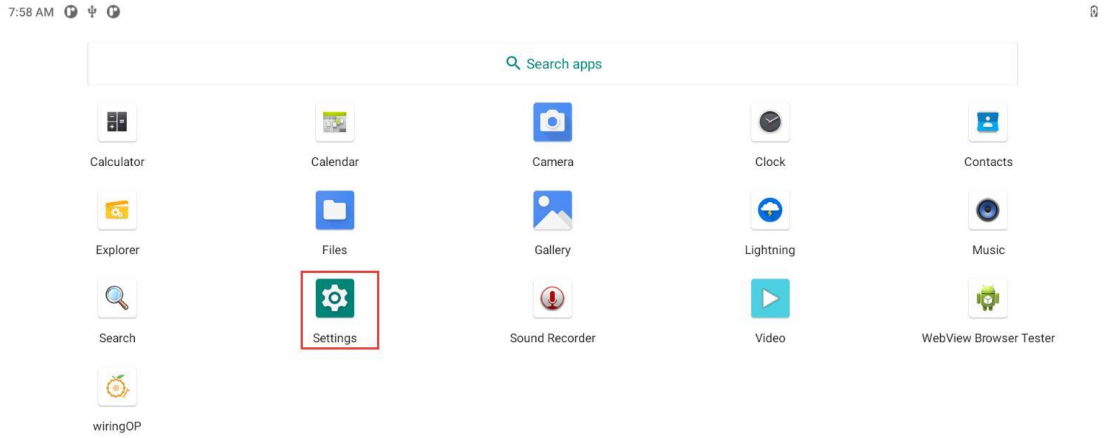


8) After the Wi-Fi connection is successful, the display is as shown in the figure below:

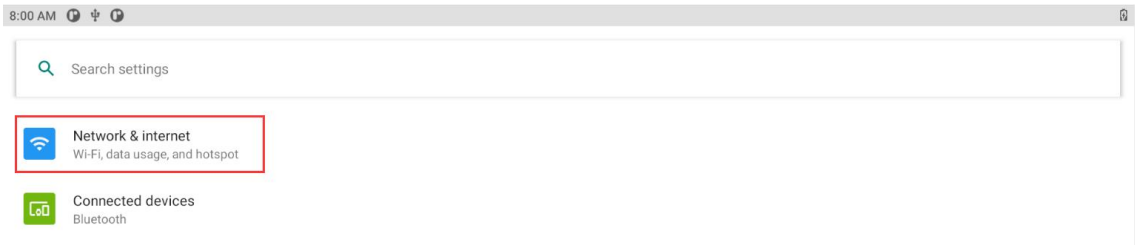
7. 4. How to use Wi-Fi hotspot

1) First, please make sure that the Ethernet port is connected to the network cable and can access the Internet normally

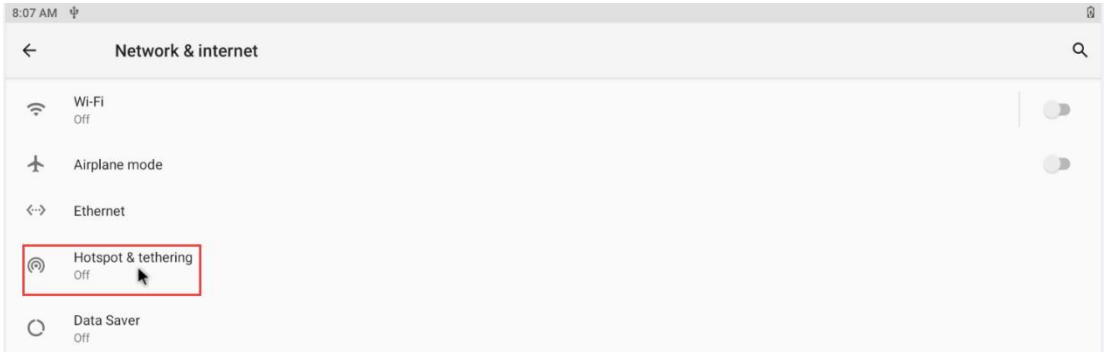
2) Then select **Settings**



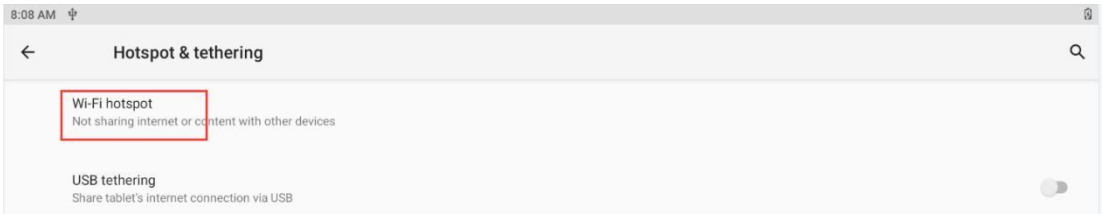
3) Then select **Network & internet**



4) Then select **Hotspot & tethering**



5) Then select **Wi-Fi hotspot**



6) Then turn on the **Wi-Fi hotspot**, 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 the **Wi-Fi hotspot** first, and then you can modify it)

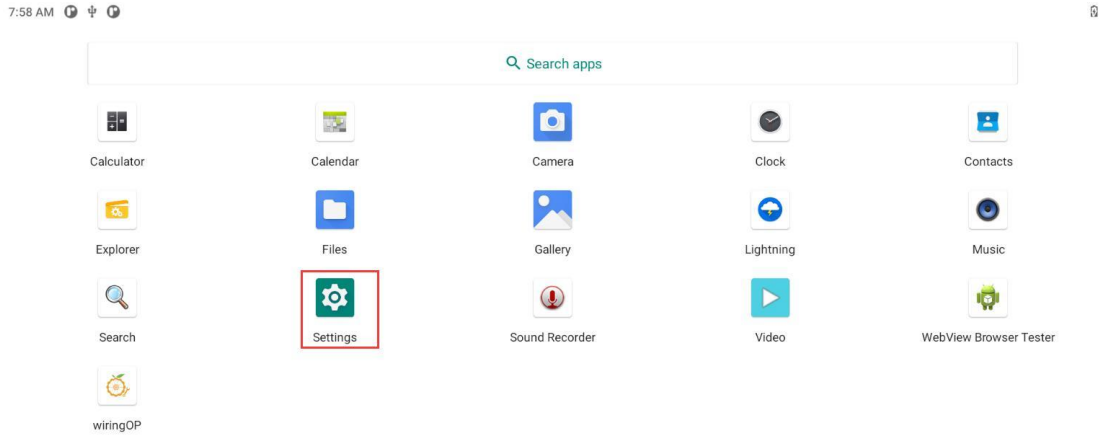
7) At this time, you can take out your mobile phone. If everything is normal, you can find the WIFI hotspot with the same name (**here AndroidAP_6953**) displayed under the **Hotspot name** in the above picture in the WI-FI list searched by the mobile phone. Then you can click **AndroidAP_6953** to connect to the hotspot, and the password can be seen under the **Hotspot password** in the above picture

8) 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

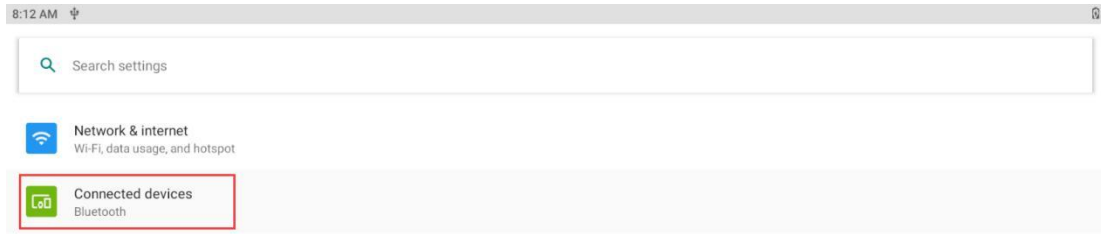
to the display of your mobile phone). At this point, you can open a webpage on your mobile phone to see if you can access the Internet. If you can open the webpage normally, it means that the **WI-FI Hotspot** of the development board can be used normally.

7. 5. Bluetooth test method

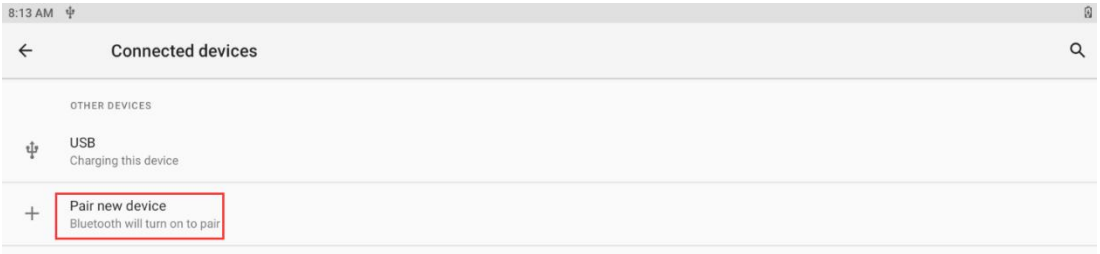
1) First click enter **Setting**



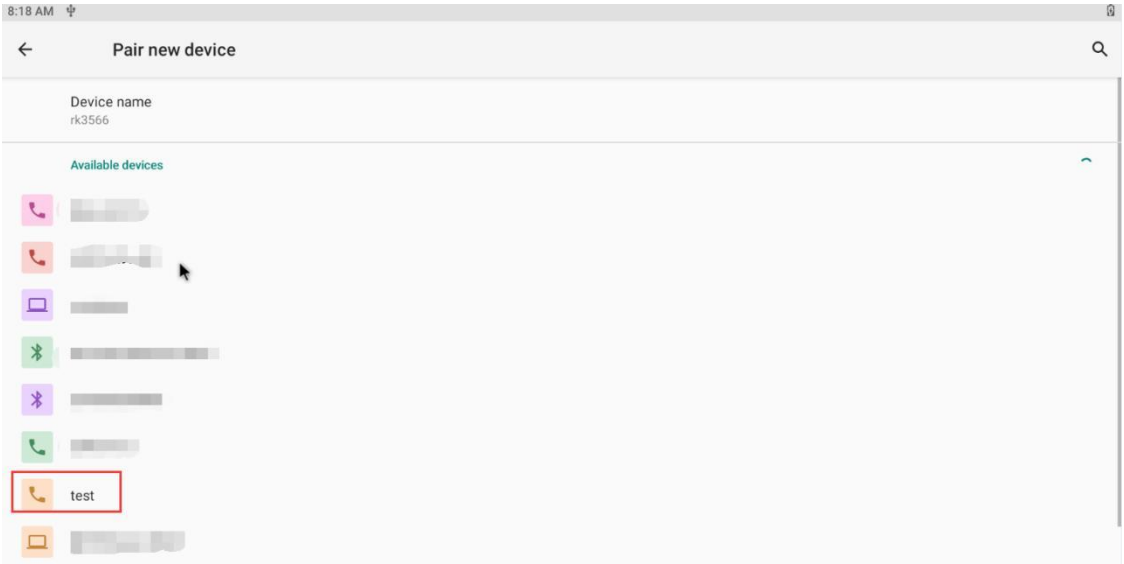
2) Then select **Connected devices**



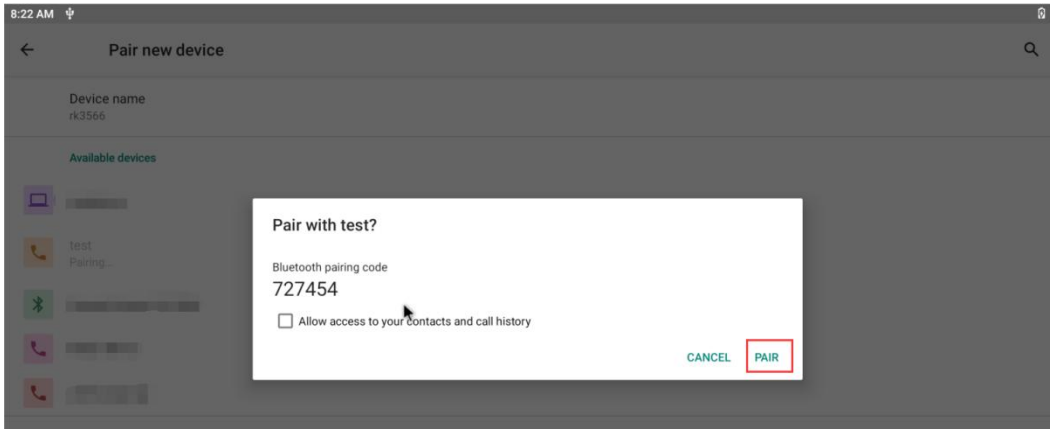
3) Then click **Pair new device** to turn on Bluetooth and start scanning the surrounding Bluetooth devices



4) The searched Bluetooth devices will be displayed under **Available devices**



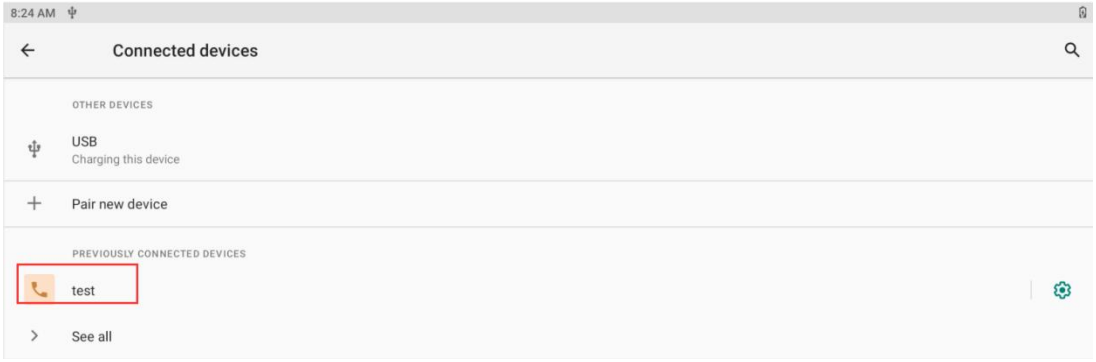
5) Then click the Bluetooth device you want to connect to start pairing. When the following interface pops up, please use the mouse to select the **Pair** option



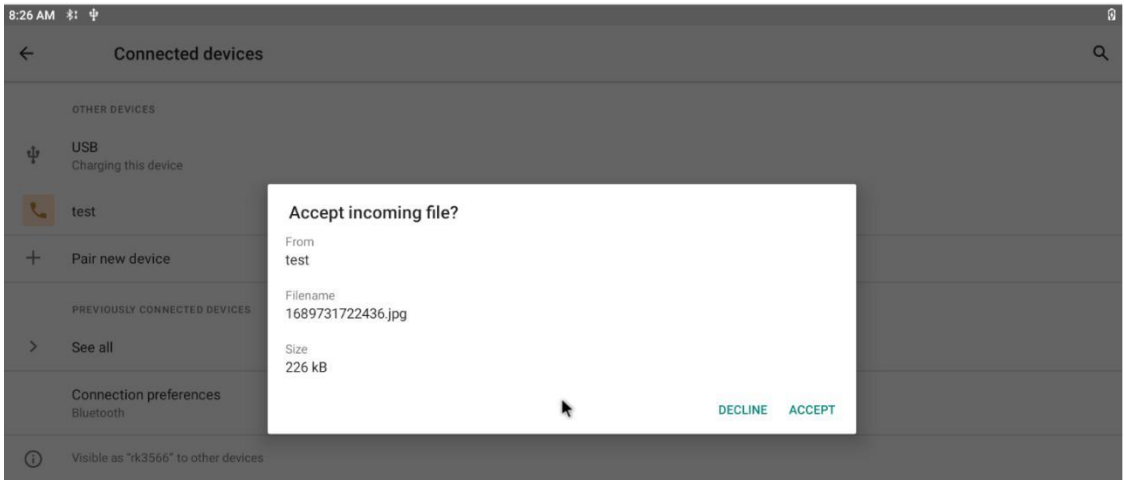
6) The test here is the configuration process of the development board and the Bluetooth of the Android mobile phone. At this time, the following confirmation interface will pop up on the mobile phone. After clicking the pairing button on the mobile phone, the

pairing process will start

7) After the pairing is completed, you can see the paired Bluetooth device as shown in the figure below



8) 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.



9) You can open the **Download** directory in the file manager to view the pictures received by the Android system Bluetooth of the development board

7. 6. How to use Raspberry Pi 5-inch screen

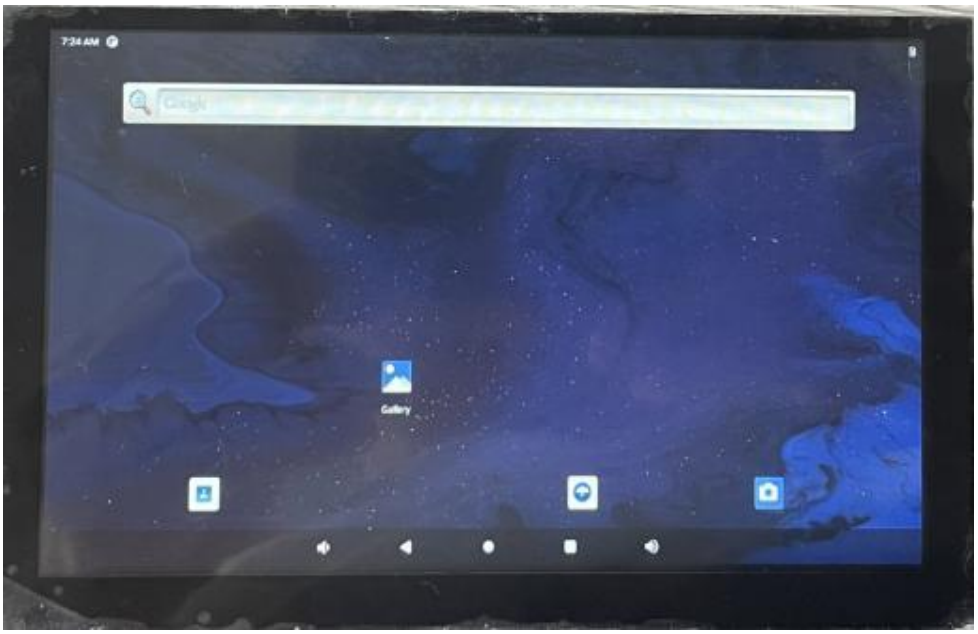
Please make sure that the image used is the following two versions of the image:
OrangePi3B_RK3566_Android11_lcd_v1.x.x.img
OrangePi3B_RK3566_Android11_spi-nvme_lcd_v1.x.x.img

1) The screen needs to be assembled first, please refer to **the assembly method of the Raspberry Pi 5-inch screen**

2) Connect the Type-C power supply to the board and power it on. After the system starts, you can see the screen display as shown in the figure below

Both the display and touch of the Raspberry Pi 5-inch screen can be used. If you have problems with the screen test, please make sure that the screen you purchased is exactly the same as the screen that the Orange Pi is compatible with.

The Orange Pi compatible screen is described in the assembly method of the Raspberry Pi 5-inch screen.

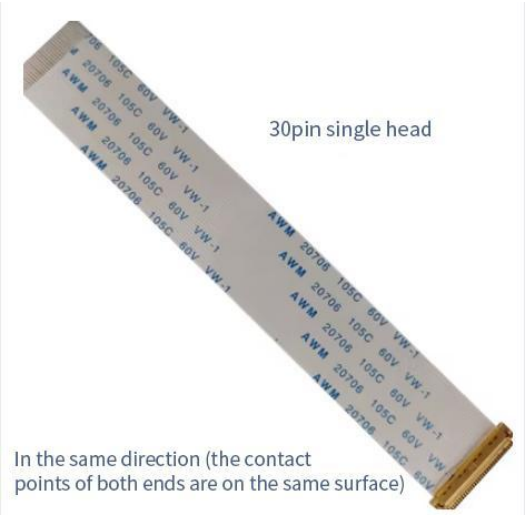


7. 7. How to use the eDP screen

Please make sure that the image used is the following two versions of the image:
OrangePi3B_RK3566_Android11_lcd_v1.x.x.img
OrangePi3B_RK3566_Android11_spi-nvme_lcd_v1.x.x.img

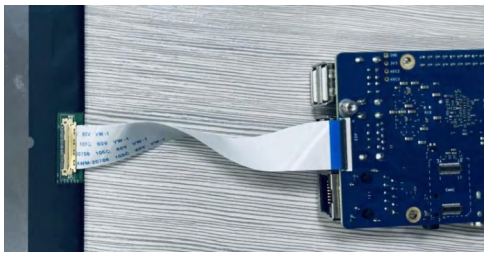
The eDP screen has no touch function.

1) Currently only one eDP screen is compatible, including the following accessories:
   a. 0.5 pitch 30pin single-head cable in the same direction



b. 15.6-inch eDP display with a resolution of 1920x1080

2) Connect the FPC end of the 30pin single-head codirectional cable to the eDP interface of the development board, and connect the other end to the eDP interface of the screen

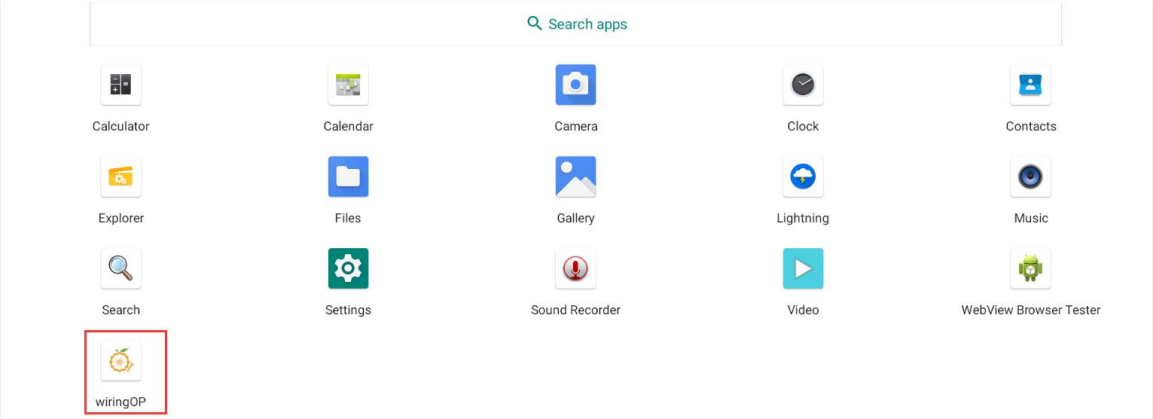


3) Then connect the Type-C power supply to the board and power it on. After the system starts, you can see the screen display as shown in the figure below

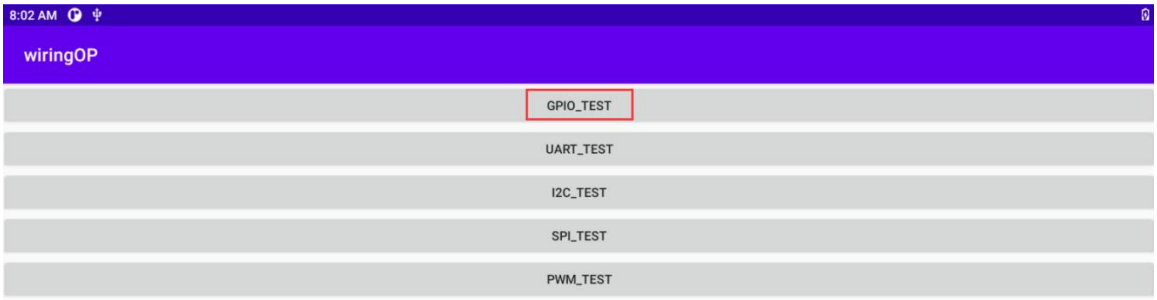
7. 8. 40pin interface GPIO, UART, SPI and PWM test

7. 8. 1. 40pin GPIO port test

1) First click on the wiringOP icon to open the wiringOP APP

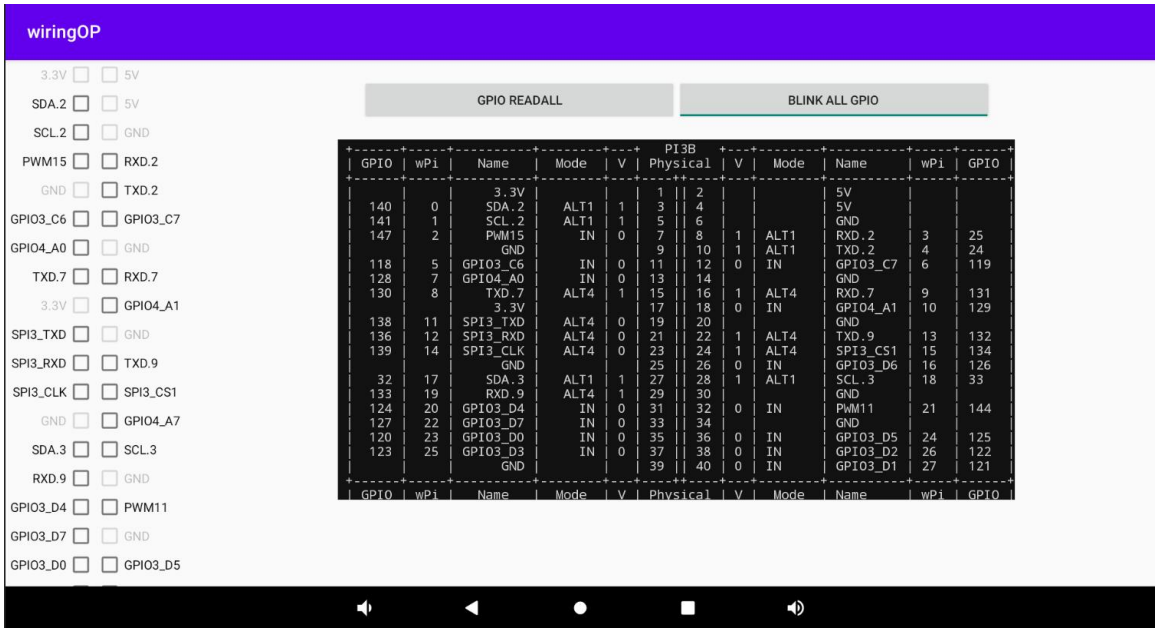


2) The main interface of wiringOP APP is displayed as shown in the figure below, and then click the **GPIO_TEST** button to open the GPIO test interface



3) The GPIO test interface is shown in the figure below. The two rows of **CheckBox** buttons on the left are in one-to-one correspondence with the 40pin 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 wPi number, GPIO mode, and pin level can be obtained; when the **BLINK ALL GPIO** button is clicked, the program will control the 28 GPIO ports to continuously switch between high and low levels

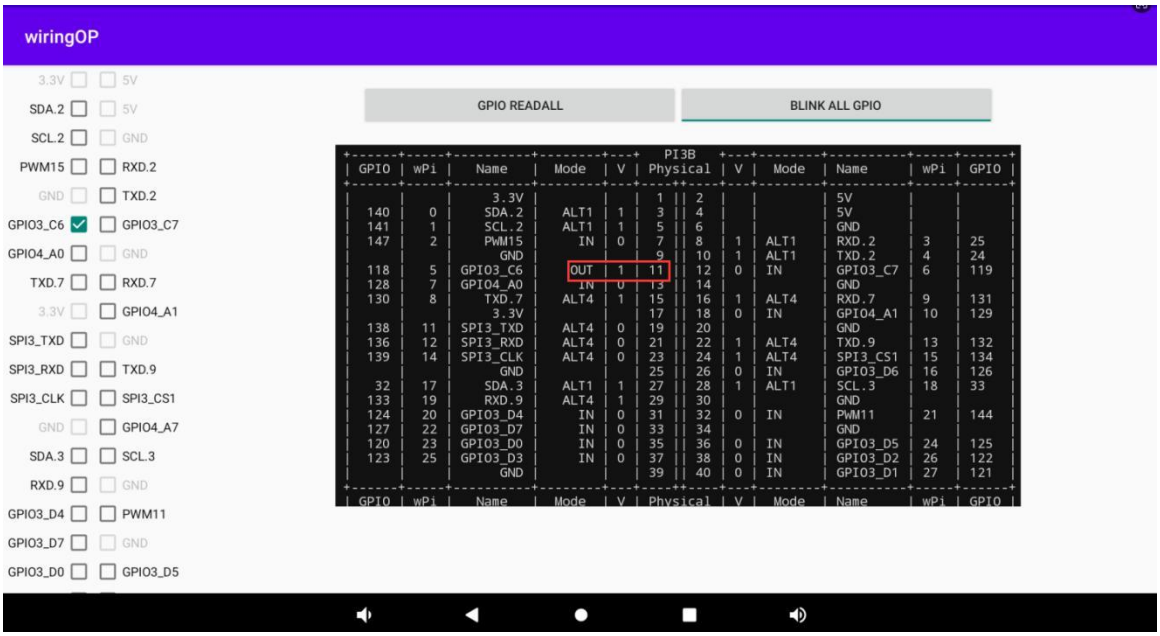
4) Then click the **GPIO READALL** button, the output information is as shown in the figure below:



5) There are a total of 28 GPIO ports in the 40pins of the development board that can be used. The following uses pin 7 — the corresponding GPIO is GPIO4_A4 — the corresponding wPi 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

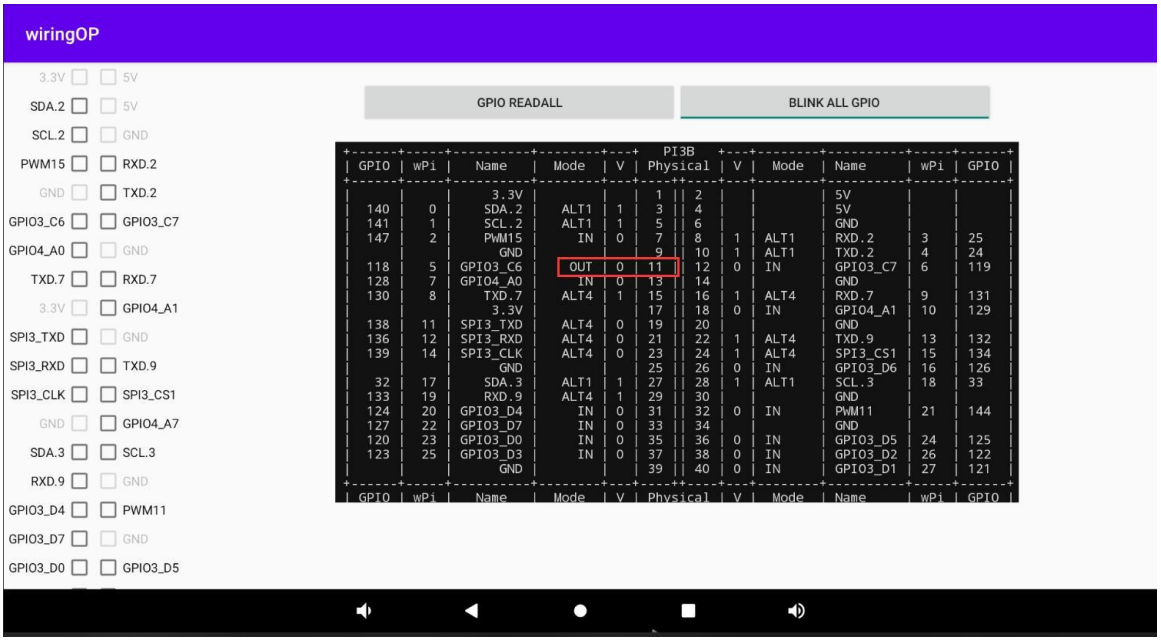
OUT, and the pin level is high



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.

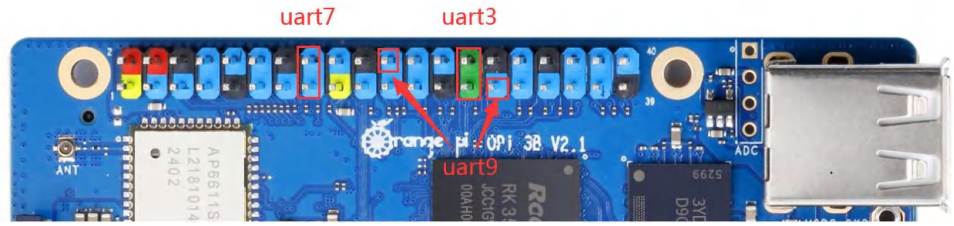
8) Then click the **GPIO READALL** button, you can see that the current pin 7 mode is

OUT, and the pin level is low

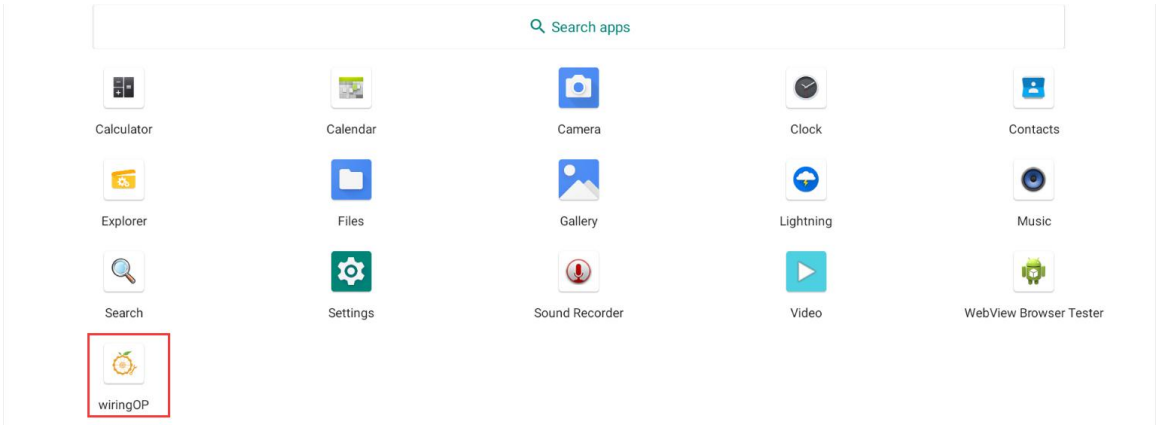


7. 8. 2. 40pin UART test

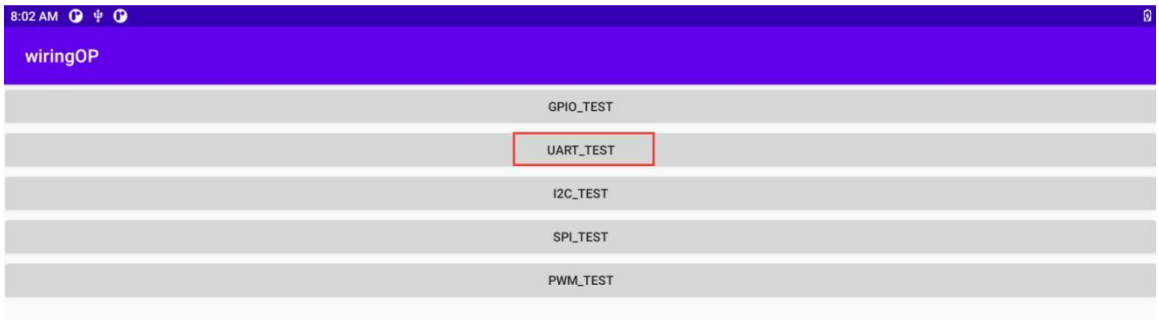
1) UART7 and UART9 are enabled by default in Android. The position of the 40pin is shown in the figure below, and the corresponding device nodes are **/dev/ttyS7** and **/dev/ttyS9** respectively



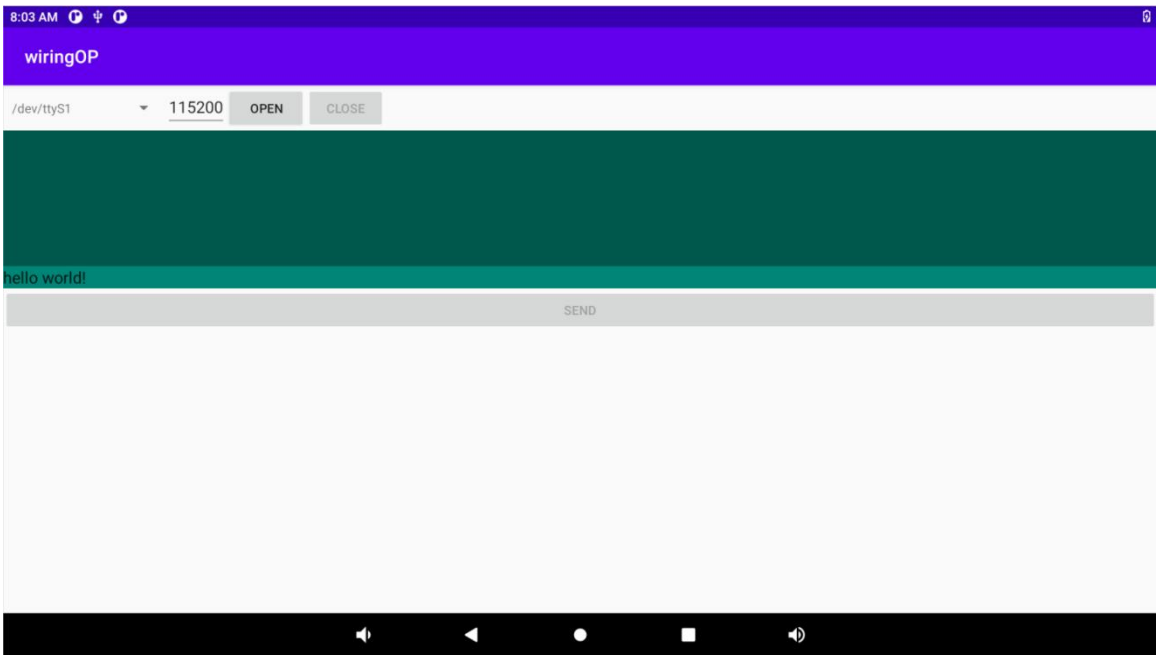
2) First click on the wiringOP icon to open the wiringOP APP



3) The main interface of wiringOP APP is displayed as shown in the figure below, and then click the **UART_TEST** button to open the UART test interface



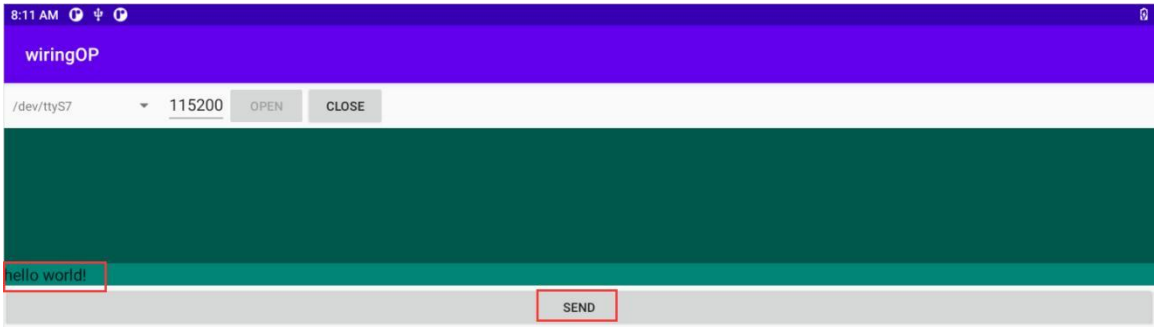
4) The serial port test interface of the APP is shown in the figure below



5) Take the test of **UART7** as an example below, select the **/dev/ttyS7** node in the selection box, enter the baud rate you want to set in the edit box, and then click the **OPEN** button to open the /dev/ttyS7 node. After the opening is successful, the **OPEN** button becomes unselectable, and the **CLOSE** button and **SEND** button become selectable

6) Then use Dupont wire to short the RXD and TXD pins of uart7

7) Then you can enter a character in the send edit box below, and click the **SEND** button to start sending



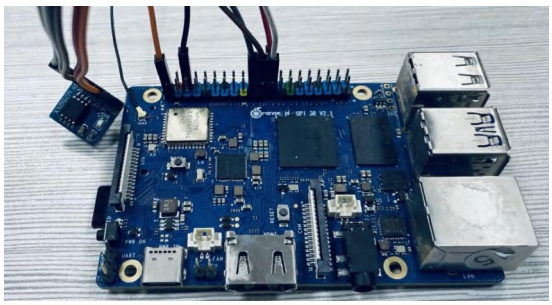
8) If everything is normal, the received string will be displayed in the receiving box

7. 8. 3. 40pin SPI test

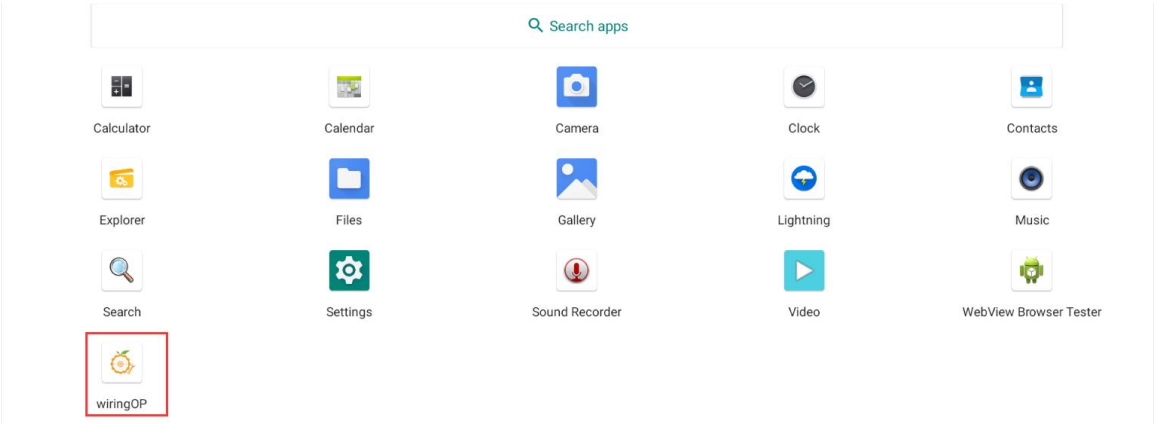
1) According to the table below, the spi available for Orange Pi 3B is spi3

复用功能	复用功能	GPIO	GPIO序号	引脚序号	引脚序号	GPIO序号	GPIO	复用功能	复用功能
		3.3V		1	2		5V		
	I2C2_SDA_M1	GPIO4_B4	140	3	4		5V		
	I2C2_SCL_M1	GPIO4_B5	141	5	6		GND		
	PWM15_IR_M1(fe700030)	GPIO4_C3	147	7	8	25	GPIO0_D1	UART2_TX_M0	
		GND		9	10	24	GPIO0_D0	UART2_RX_M0	
		GPIO3_C6	118	11	12	119	GPIO3_C7		
		GPIO4_A0	128	13	14		GND		
	UART7_TX_M2	GPIO4_A2	130	15	16	131	GPIO4_A3	UART7_RX_M2	
		3.3V		17	18	129	GPIO4_A1		
I2C4_SDA_M0	SPI3_MOSI_M0	GPIO4_B2	138	19	20		GND		
	SPI3_MISO_M0	GPIO4_B0	136	21	22	132	GPIO4_A4	UART9_TX_M2	
I2C4_SCL_M0	SPI3_CLK_M0	GPIO4_B3	139	23	24	134	GPIO4_A6	SPI3_CS0_M0	
		GND		25	26	126	GPIO3_D6		
UART3_RX_M0	I2C3_SDA_M0	GPIO1_A0	32	27	28	33	GPIO1_A1	I2C3_SCL_M0	UART3_TX_M0
	UART9_RX_M2	GPIO4_A5	133	29	30		GND		
		GPIO3_D4	124	31	32	144	GPIO4_C0	PWM11_IR_M1(fe6f0030)	
		GPIO3_D7	127	33	34		GND		
		GPIO3_D0	120	35	36	125	GPIO3_D5		
		GPIO3_D3	123	37	38	122	GPIO3_D2		
		GND		39	40	121	GPIO3_D1		

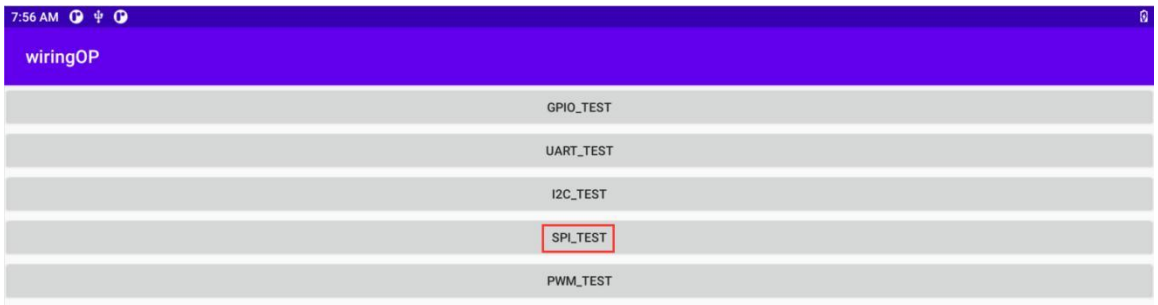
2) Here, the SPI interface is tested through the w25q64 module. First, the w25q64 device is connected to the SPI3 interface



3) Then click the wiringOP icon to open the wiringOP APP

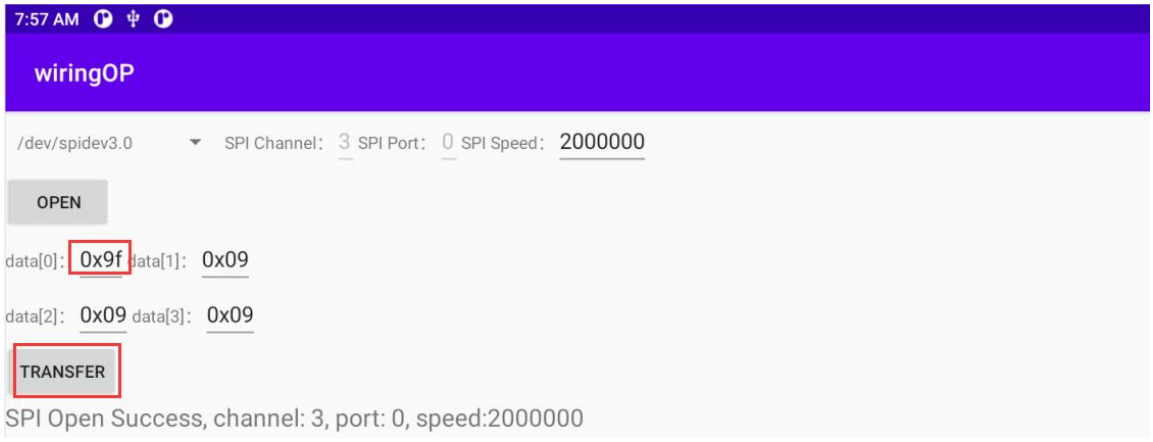


4) The main interface of wiringOP APP is displayed as shown in the figure below, click the SPI_TEST button to open the SPI test interface

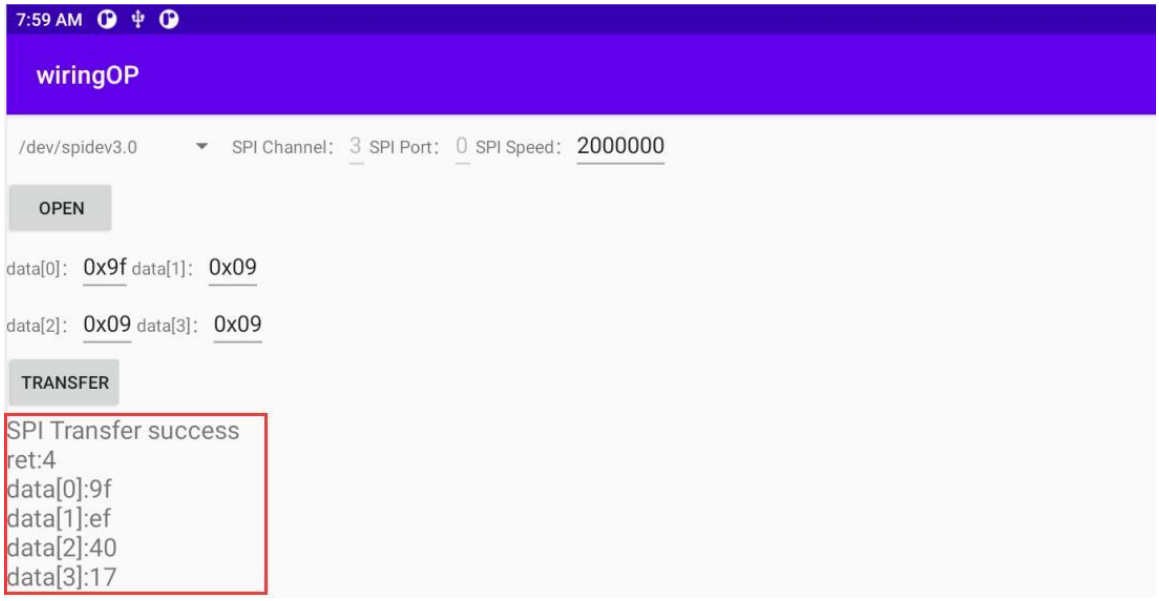


5) Then click the **OPEN** button to initialize the SPI

6) Then fill in the bytes that need to be sent, such as reading the ID information of w25q64, fill in the address 0x9f in data[0], and then click the **TRANSFER** button



7) Finally, the APP will display the read ID information



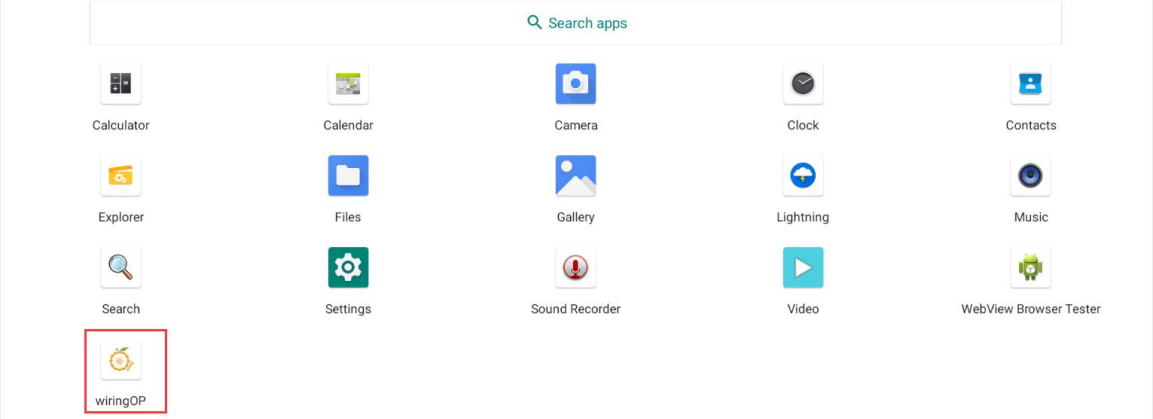
8) The MANUFACTURER ID of the w25q64 module is EFh, and the Device ID is 4017h, corresponding to the value read above (h stands for hexadecimal)

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

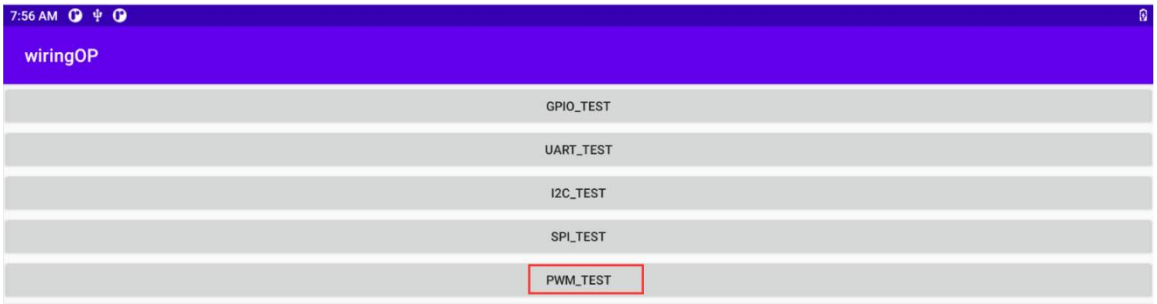
7. 8. 4. 40pin PWM test

1) Android enables **PWM11** by default, and the corresponding pin is located at 40pin as shown in the figure below

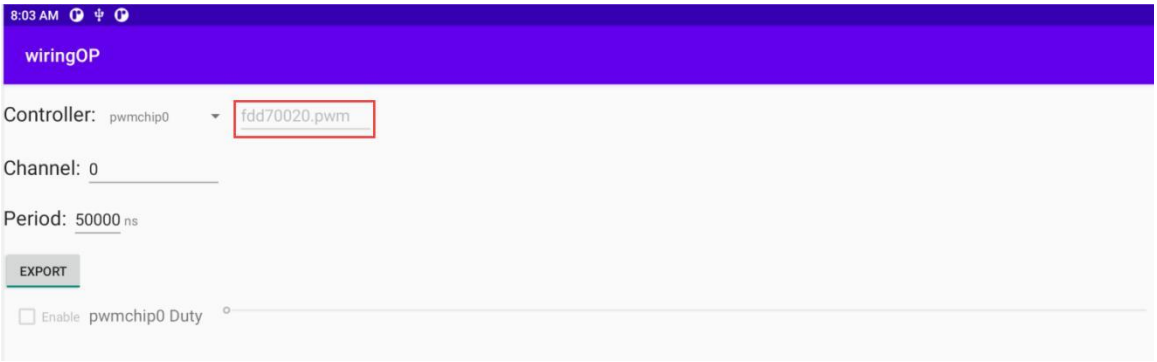
2) First click on the wiringOP icon to open the wiringOP APP



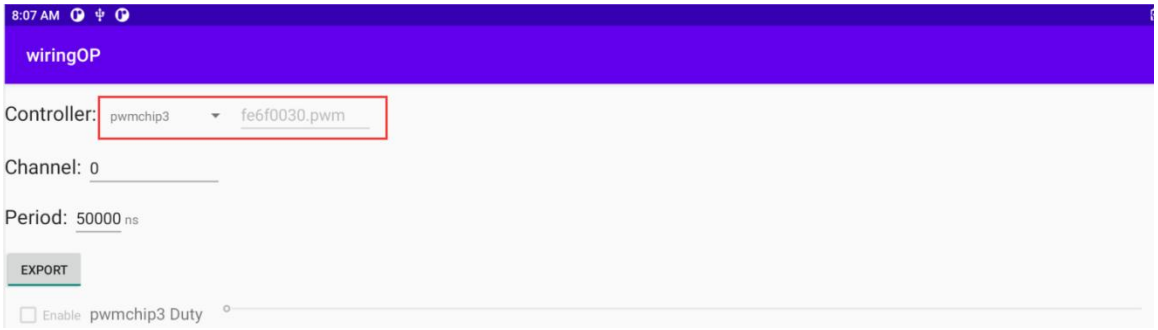
3) Then click the **PWM_TEST** button on the main interface of wiringOP to enter the PWM test interface



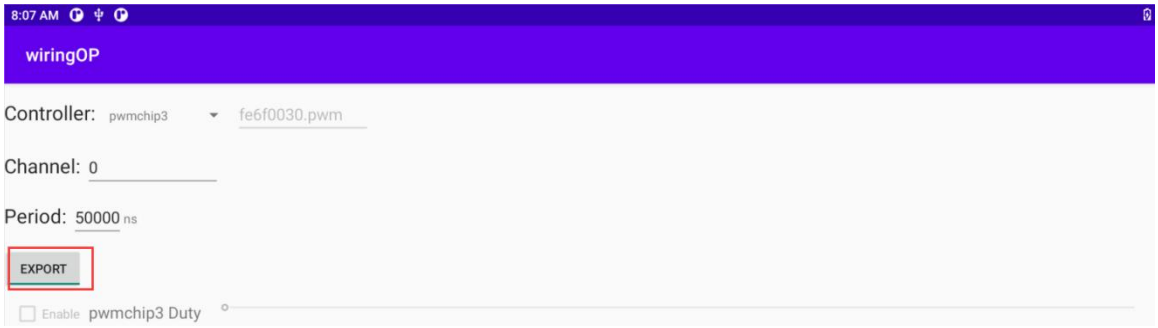
4) The base address corresponding to PWM11 is **fe6f0030**, here pwmchip0 shows **fdd70020.pwm** on the right, then you need to click the drop-down option to select other pwmchips until **fe6f0030.pwm** is displayed on the right



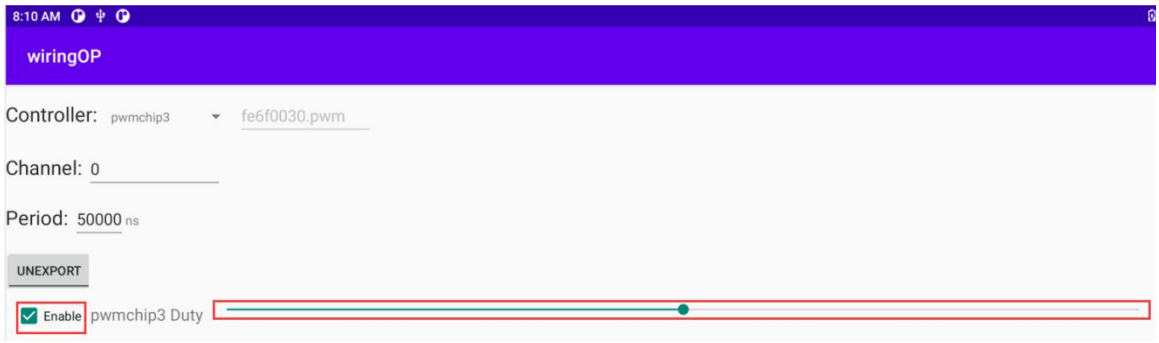
5) When the drop-down option selects **pwmchip3**, the corresponding base address of PWM11 is **fe6f0030** on the right



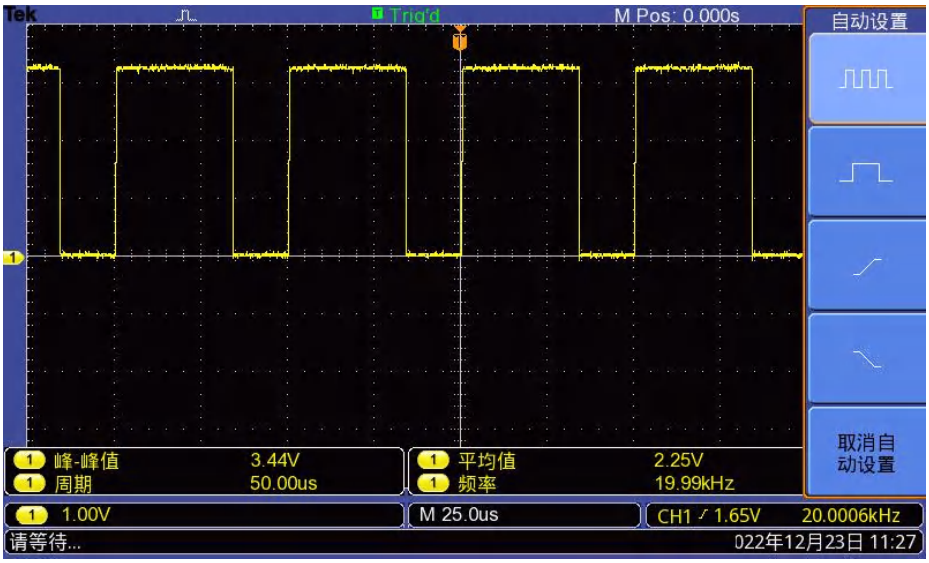
6) Then confirm the PWM channel, the default is channel 0, and confirm the PWM cycle, the default configuration is **50000ns**, converted to PWM frequency is **20KHz**, you can modify it yourself, click the **EXPORT** button to export **PWM11**



7) Then drag the drag bar below to change the PWM duty cycle, and then check Enable to output the PWM waveform



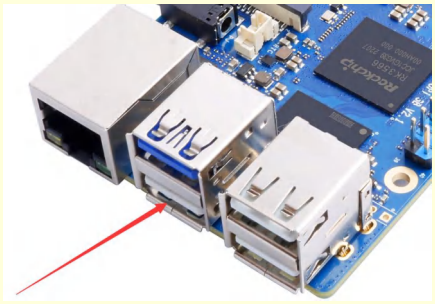
8) Then use an oscilloscope to measure the No. 32 pin in the 40pin of the development board, and you can see the following waveform



7. 9. How to use ADB

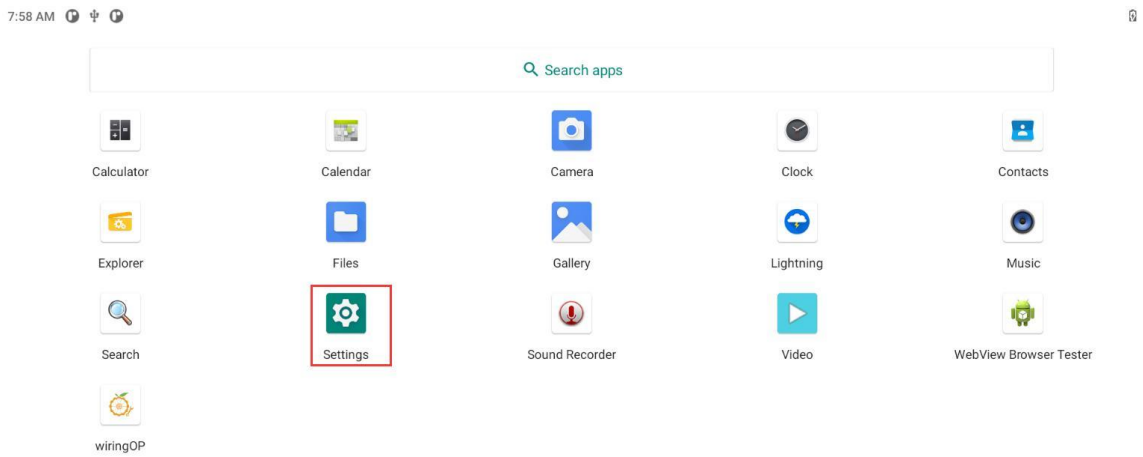
7. 9. 1. The method of USB OTG mode switching

The development board has 4 USB interfaces, among which the USB interface marked in red box in the figure below can support both Host mode and Device mode, and the other 3 USB interfaces only support Host mode.

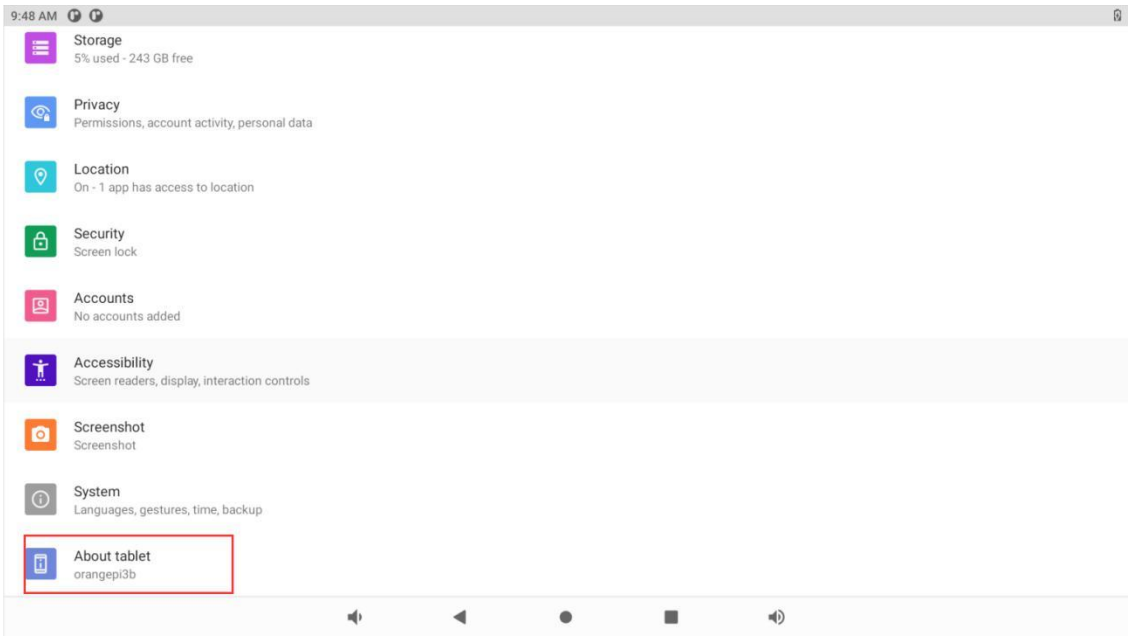


The USB OTG interface defaults to Host mode, which can be used to connect USB devices such as mouse and keyboard. If you want to use ADB, you need to manually switch to Device mode.

1) First open Settings

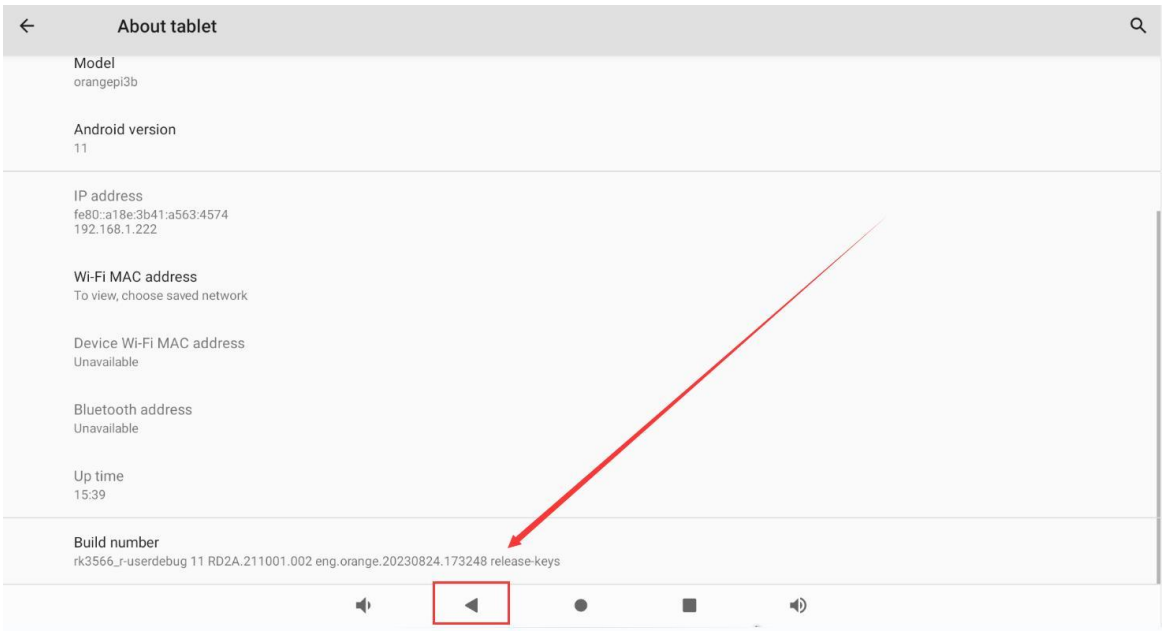


2) Then select **About tablet**

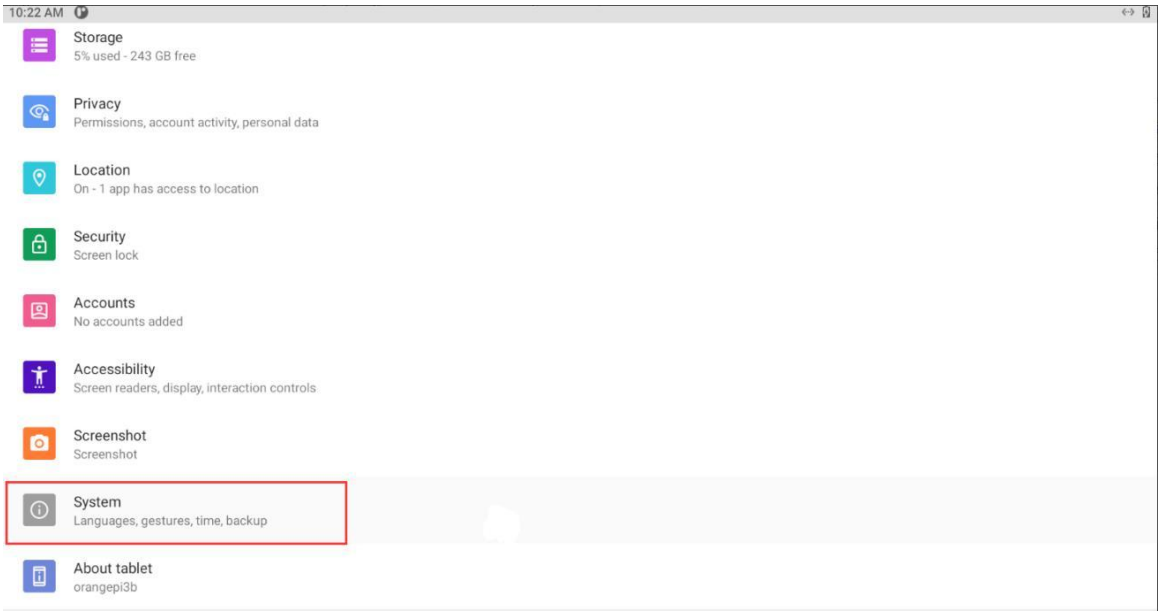


3) Then click the **Build number** menu bar several times with the mouse until the prompt **You are now a developer!** appears

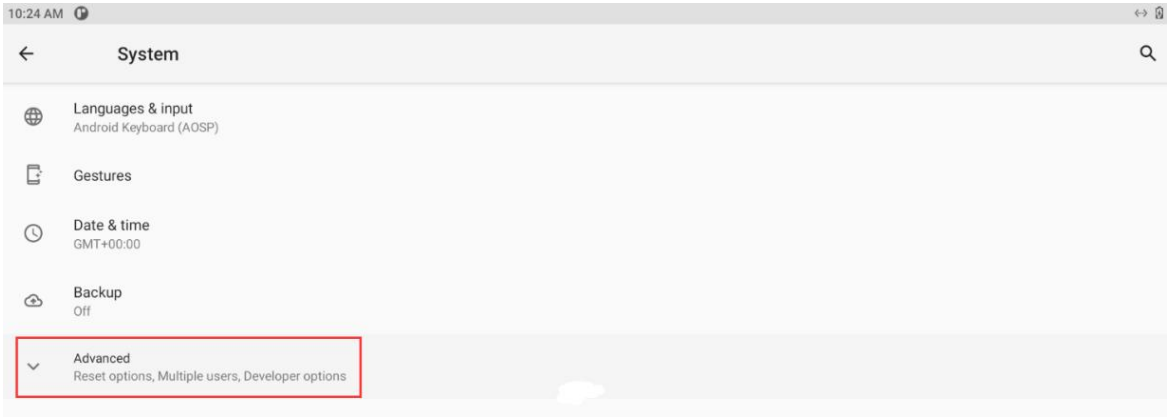
4) Then click to return to the previous menu



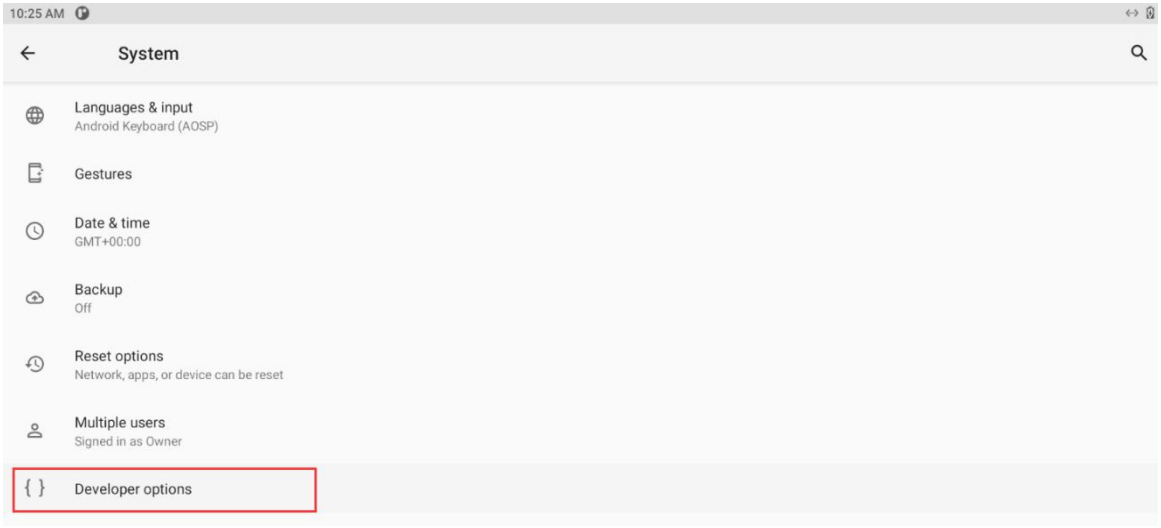
5) Then select **System**



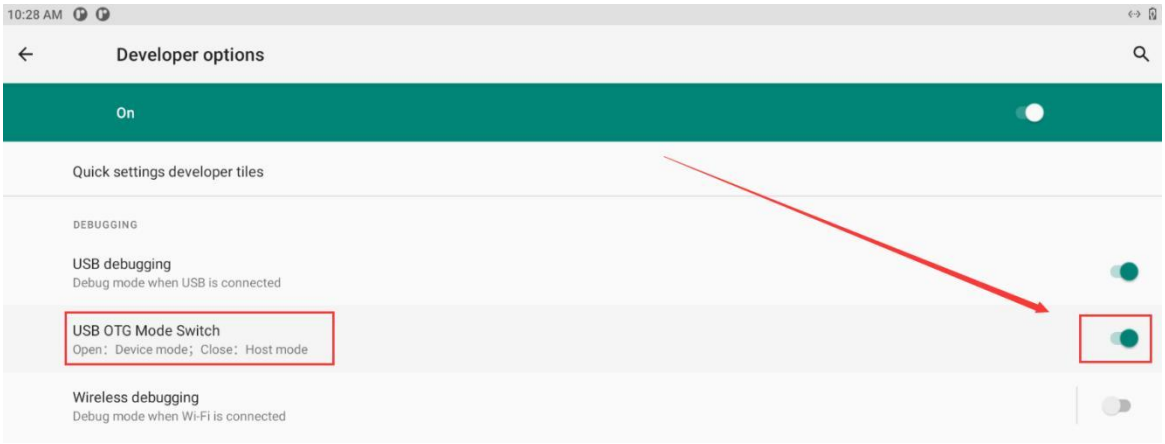
6) Then select **Advanced**



7) Then select **Developer options** in the expanded column



8) Finally find the **USB OTG Mode Switch** switch, **turn on the switch to switch to Device mode, turn off the switch to switch to Host mode**

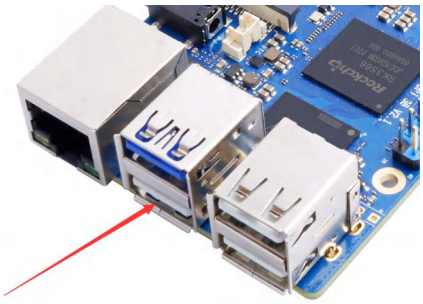


7. 9. 2. Use the data cable to connect to adb debugging

1) First prepare a good quality USB2.0 male-to-male data cable

2) Then refer to **the method of USB OTG mode switching** to switch USB OTG to Device mode

3) Then connect the development board to the Ubuntu PC through the USB2.0 male-to-male data cable. The position of the USB OTG interface on the development board is shown in the figure below:



4) Then install the adb tool on the Ubuntu PC

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

5) You can view the identified ADB devices through the following command

```
test@test:~$ adb devices
List of devices attached
S63QCF54CJ      device
test@test:~$ lsusb
Bus 003 Device 006: ID 2207:0006
```

6) Then you can log in to the android system through the adb shell on the Ubuntu PC

```
test@test:~$ adb shell
console:/ $
```

7) Execute the following command to remount the Android system

```
test@test:~$ adb root
test@test:~$ adb remount
```

8) Then you can transfer files to the Android system

```
test@test:~$ adb push example.txt /system/
```

7. 9. 3. Use network connection adb debugging

Using the network adb does not require a 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 is connected, and then obtain the IP address of the development board, which will be used later.

1) Make sure that the **service.adb.tcp.port** of the Android system is set to port number 5555

```
console:/ # getprop | grep "adb.tcp"
[service.adb.tcp.port]: [5555]
```

2) If **service.adb.tcp.port** is not set, you can use the following command to set the port number of network adb

```
console:/ # setprop service.adb.tcp.port 5555
console:/ # stop adbd
console:/ # start adbd
```

3) Install adb tool on Ubuntu PC

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

4) Then connect to the network adb on the Ubuntu PC

```
test@test:~$ adb connect 192.168.1.xxx     (IP 地址需要修改为开发板的 IP 地址)
* 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
console:/ #
```

8. Instructions for using the OpenWRT system

8. 1. OpenWRT Version

OpenWRT Version	Kernel version
snapshot	Linux6.1

8. 2. OpenWRT adaptation situation

Function	OpenWRT
USB2.0x3	OK
USB3.0x1	OK
3pin debugging serial port	OK
TF card startup	OK
UWE5622 WIFI	NO
M.2 NVMe SSD boot	OK
Network port	OK
Network port status light	OK
RTL8811 USB network card	OK
USB to wired network card	OK
LED light	OK
FAN interface	OK
eMMC expansion interface	OK

8. 3. Start expanding rootfs for the first time

1) When the OpenWRT system is started for the first time, the resize-rootfs.sh script will be executed to expand the rootfs, and it will automatically restart after the expansion is completed.

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

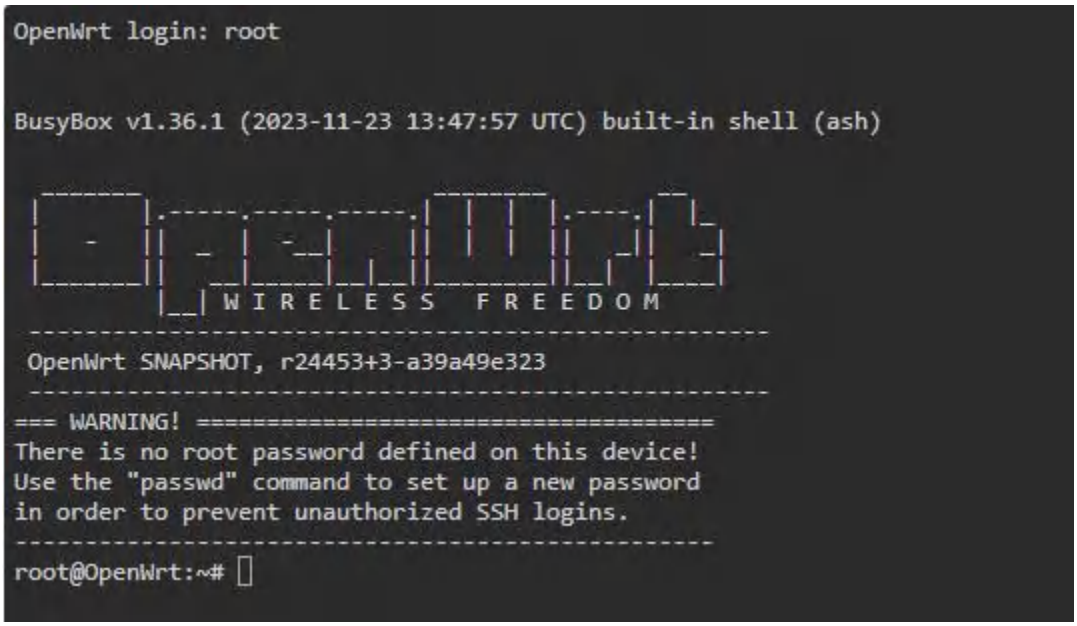
```
root@OpenWrt:~# df -h
Filesystem                Size      Used Available Use% Mounted on
/dev/root                14.8G     14.7G     91.6M  99% /
tmpfs                   495.5M      6.1M    489.4M   1% /tmp
tmpfs                   512.0K         0    512.0K   0% /dev
/dev/root                14.8G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

8. 4. How to log in to the system

8. 4. 1. Log in through serial port

1) First, for the use of the debugging serial port, please refer to the chapter on **how to use the debugging serial port**.

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



8. 4. 2. Log in to the system via SSH

Please note that in the OpenWrt system of Orange Pi 3B, the onboard network port is configured as a LAN port function by default.

1) First use a network cable to connect the LAN port of the board to the network port of the computer so that the computer's network port can obtain the IP address through DHCP.

2) The default LAN port IP of the board is set to **192.168.2.1**, so at this time the computer can obtain an IP address starting with **192.168.2**

3) If the Ubuntu system is installed on the computer, you can execute the following command to log in to the system through SSH. By default, you can log in directly without a password

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

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

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

8. 4. 3. Log in to the LuCI management interface

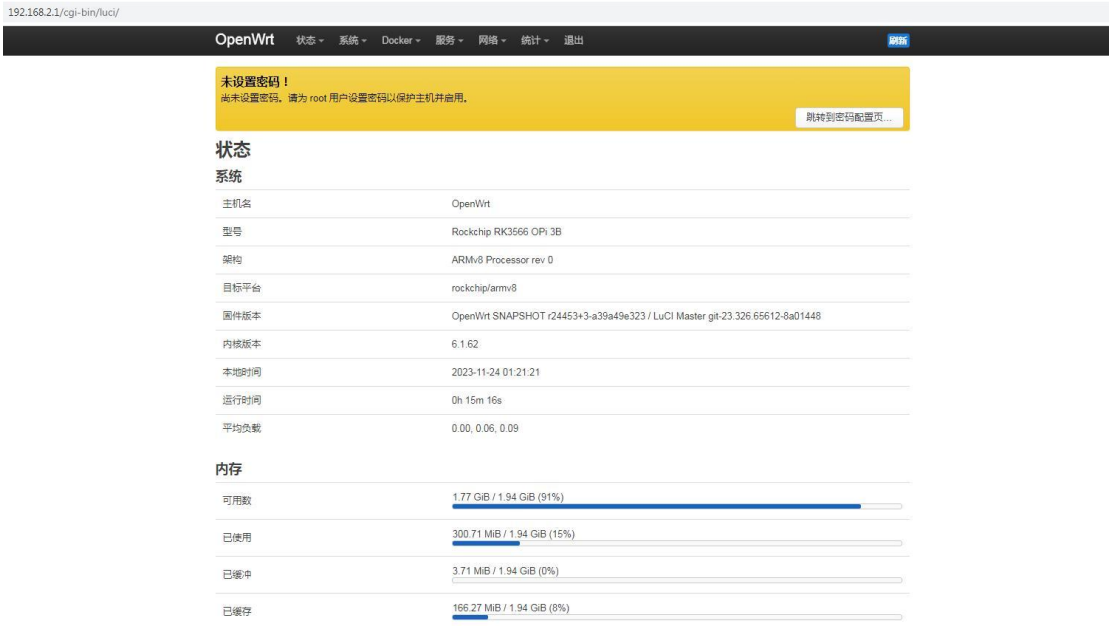
Please note that in the OpenWrt system of Orange Pi 3B, the onboard network port is configured as a LAN port function by default.

1) First use a network cable to connect the LAN port of the board to the network port of the computer so that the computer's network port can obtain the IP address through DHCP.

2) The default LAN port IP of the board is set to **192.168.2.1**, so at this time the computer can obtain an IP address starting with **192.168.2**

3) Enter the IP address **192.168.2.1** in the browser on your computer to log in to the LuCI interface

4) The OpenWrt system does not set a password by default, so just click the login button. After successful login, the interface will be displayed as shown below.



8. 4. 4. Log in to the terminal through the LuCI management interface

Please note that in the OpenWrt system of Orange Pi 3B, the onboard network port is configured as a LAN port function by default.

1) First use a network cable to connect the LAN port of the board to the network port of the computer so that the computer's network port can obtain the IP address through

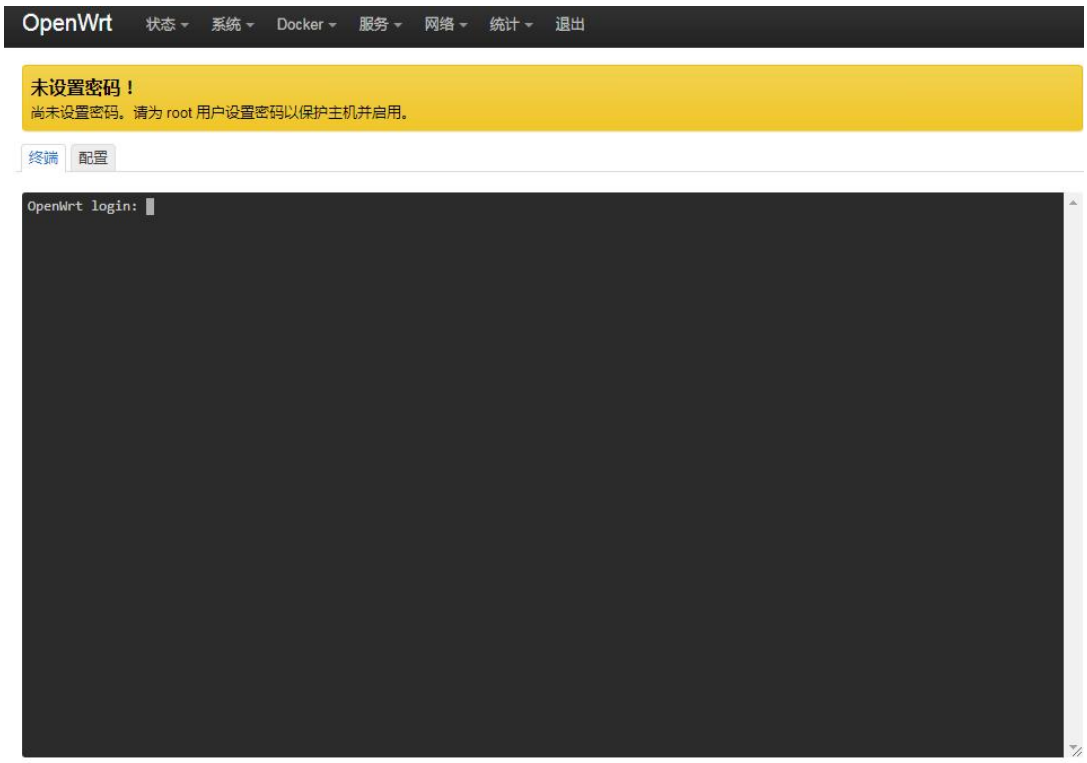
DHCP.

2) The default LAN port IP of the board is set to **192.168.2.1**, so at this time the computer can obtain an IP address starting with **192.168.2**

3) Enter the IP address **192.168.2.1** in the browser on your computer to log in to the LuCI interface

4) Select **"Terminal"** in the **"Service"** column of the navigation bar and click to enter

5) At this time, the terminal display interface is as shown below



6) Enter the username **root** to log in

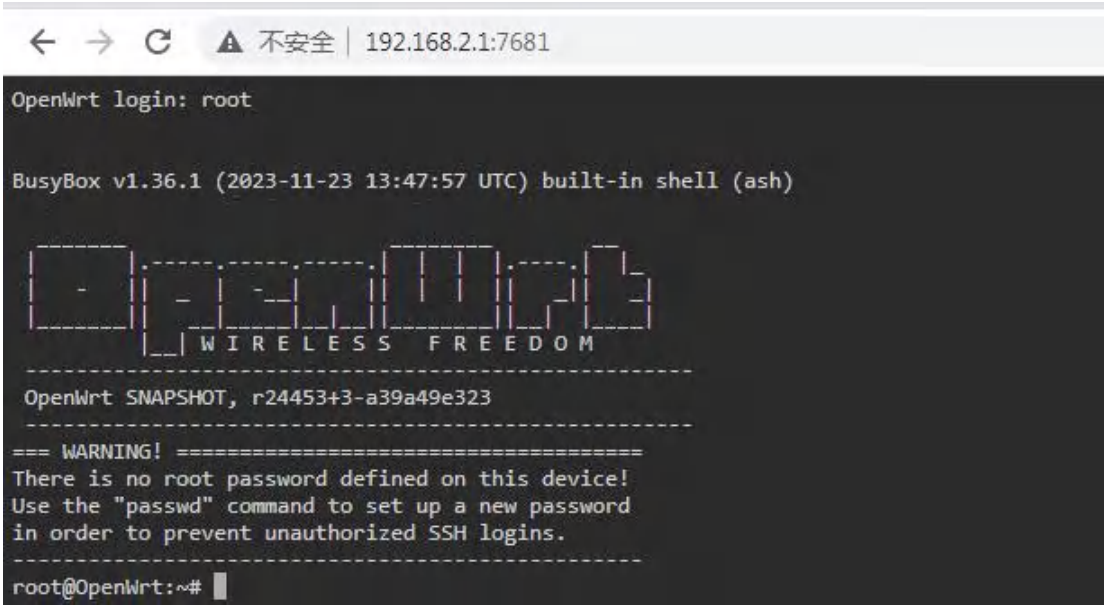
8. 4. 5. Log in to the terminal using IP address + port number.

Please note that in the OpenWrt system of Orange Pi 3B, the onboard network port is configured as a LAN port function by default.

1) First use a network cable to connect the LAN port of the board to the network port of the computer so that the computer's network port can obtain the IP address through DHCP.

2) The default LAN port IP of the board is set to **192.168.2.1**, so at this time the computer can obtain an IP address starting with **192.168.2**

3) Then enter **192.168.2.1:7681** in the browser to log in to the OpenWRT terminal



8.5. How to modify the LAN port IP address through the command line

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

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

```
root@OpenWrt:~# uci show network
...
```

```
network.lan=interface
network.lan.device='br-lan'
network.lan.proto='static'
network.lan.ipaddr='192.168.2.1'
network.lan.netmask='255.255.255.0'
network.lan.ip6assign='60'
....
```

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

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

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

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

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

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

5) Restart the network through ubus. For instructions on using ubus, please refer to the **official documentation**.

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

6) At this time, enter the command and you can see that the IP of the LAN port is already **192.168.100.1**

```
root@OpenWrt:~# ifconfig br-lan
br-lan      Link encap:Ethernet  HWaddr FE:55:13:A3:EF:E7
```

```
inet addr:192.168.100.1  Bcast:192.168.100.255  Mask:255.255.255.0
inet6 addr: fd60:c4cd:1033::1/60 Scope:Global
UP BROADCAST MULTICAST  MTU:1500  Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:3 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:0 (0.0 B)  TX bytes:370 (370.0 B)
```

8. 6. How to change the root password

8. 6. 1. Modification through command line

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

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

2) You will then be prompted to re-enter your password. At this time, enter your password again to confirm and press Enter.

```
Retype password:
```

3) The successful modification is displayed as follows

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

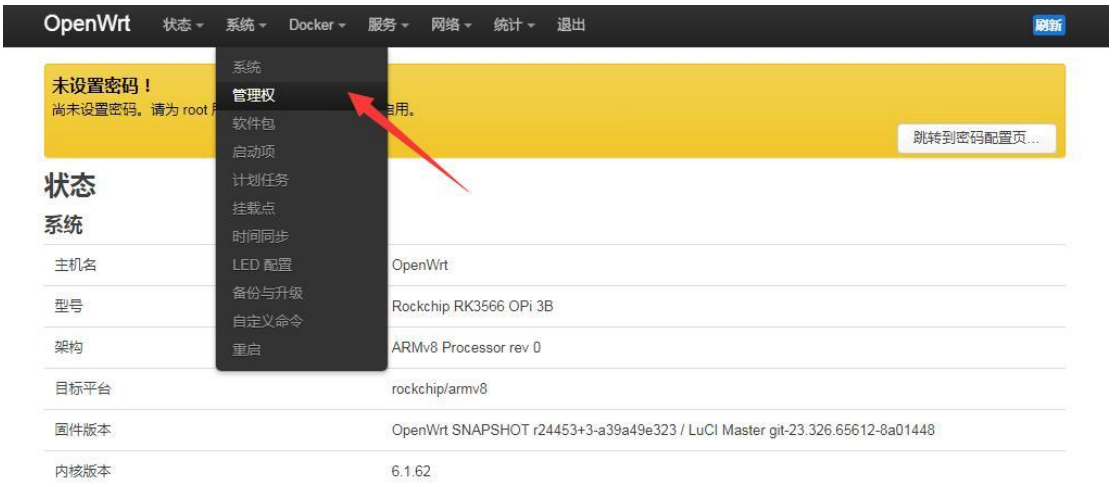
8. 6. 2. Modify through LuCI management interface

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

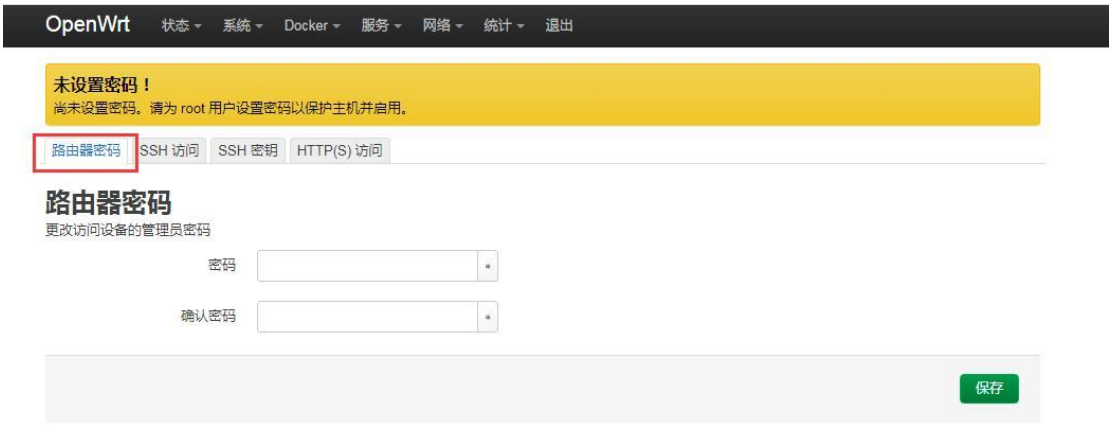
2) Then follow the steps below to change the password

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

b. In the vertical column options below the system, select "**Management Rights**" and click



c. Select the "**Router Password**" option on the Tab page



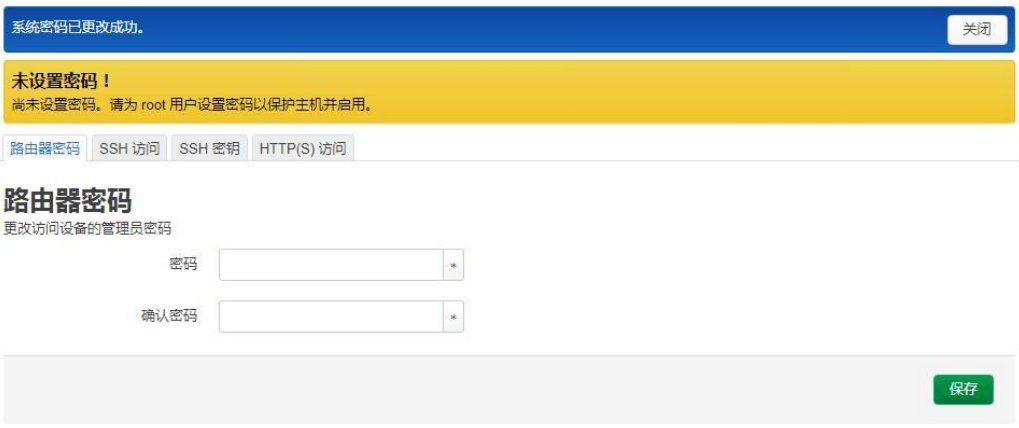
3) Modify and save the router password

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

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

Note: In the "Password" and "Confirm Password" dialog boxes, the passwords entered twice must be consistent.

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



8. 7. USB interface test

8. 7. 1. Mount USB storage device from command line

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

2) Execute the following command. If you can see the output of sdX, it means the USB disk is successfully recognized.

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

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

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

```
test.txt
```

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

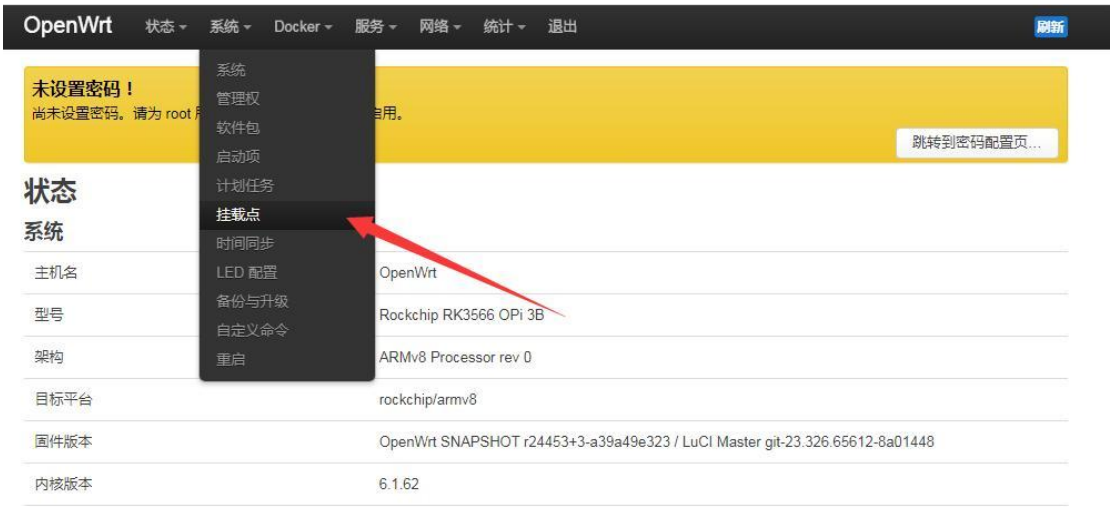
```
root@OpenWrt:~# df -h | grep "sd"
/dev/sda                 14.4G       187.2M        14.2G   1% /mnt
```

8. 7. 2. Mount USB storage device in LuCI management interface

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

2) Then log in to **the LuCI management interface** to enter the LuCI management interface.

3) Then in the LuCI management interface, click "System->Mount Point" to enter the mount point configuration interface



4) Then follow the steps below to add a mount point

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

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

c. The following pop-up window interface will pop up.

d. Then you can start mounting the storage device

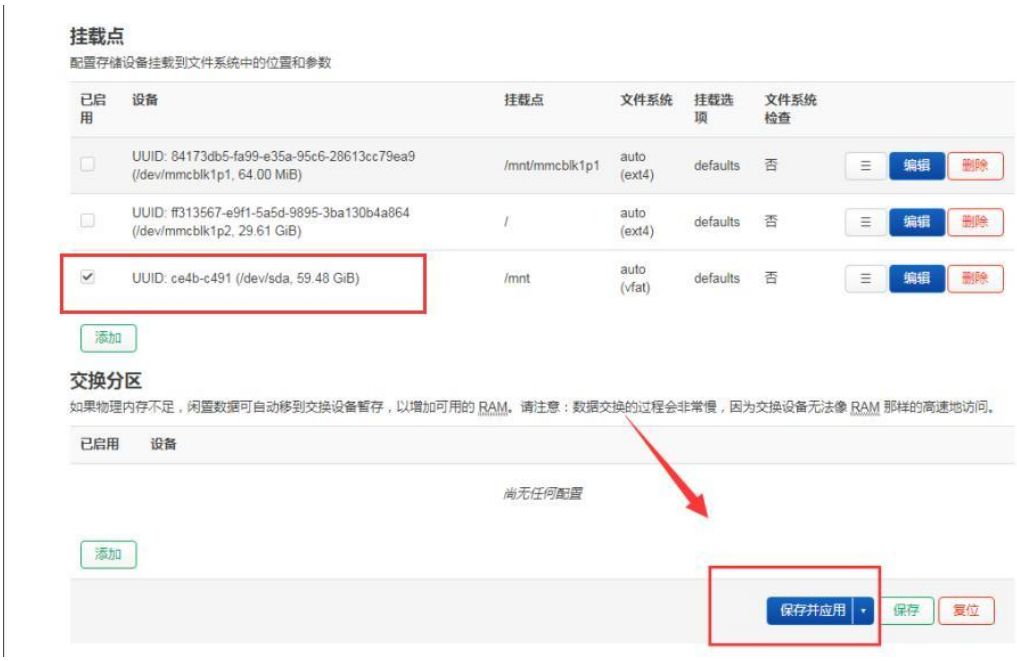
a) Check "**Enabled**"

b) Select the actual connected device /dev/sda in the General Settings UUID column (select according to your own device)

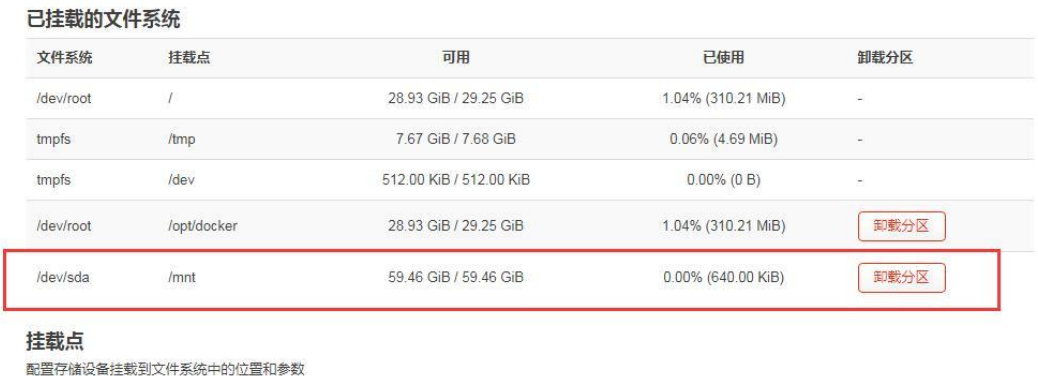
c) Select "**Custom**" in the mount point column and fill in the target directory to be mounted. Here, the **/mnt** directory is used as an example. After filling in, press **Enter** to confirm.

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

5) Then you will return to the mount point global settings page. Click "Save and Apply" in the lower left corner of the page to make the mount point effective.



6) After saving, you can see that the storage device has been mounted successfully in "**Mounted File Systems**"



8.8. USB wireless network card test

The usable USB wireless network cards that **have been tested** so far are as follows. Please test other models of USB wireless network cards by yourself. If it cannot be used, you need to transplant the corresponding USB wireless network card driver.

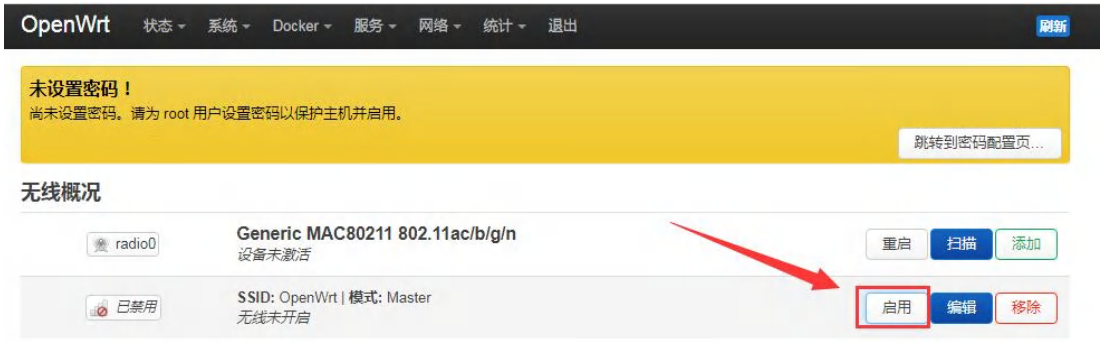
serial number	model	
1	RTL8811 Support 2.4G +5G WIFI	

8.8.1. Method to create WIFI hotspot using USB wireless network card

1）Insert the USB wireless network card into the USB port of the development board, and then connect the power supply to power up the development board.

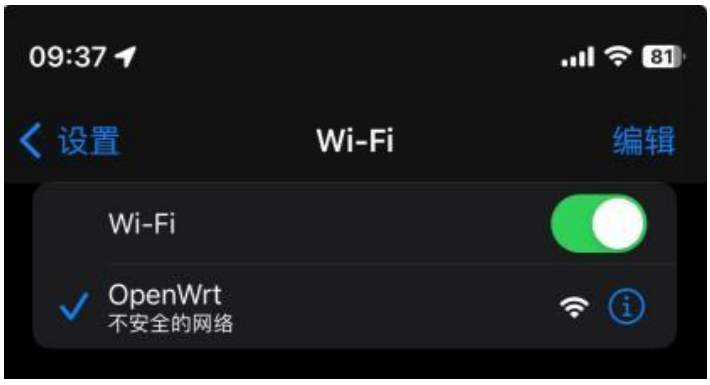
2）After the system starts, click **Network -> Wireless** to enter the wireless WiFi configuration interface. If there is no **wireless** option, it means that the USB wireless network card model is not supported by the system.

3）The default wireless configuration of the OpenWRT system is **Master** mode. We can directly click to **enable** to create an open WIFI hotspot.



4）The display interface of successfully creating a WIFI hotspot is as shown below

5）At this point, you can connect to the WIFI hotspot named **OpenWrt** through your mobile phone

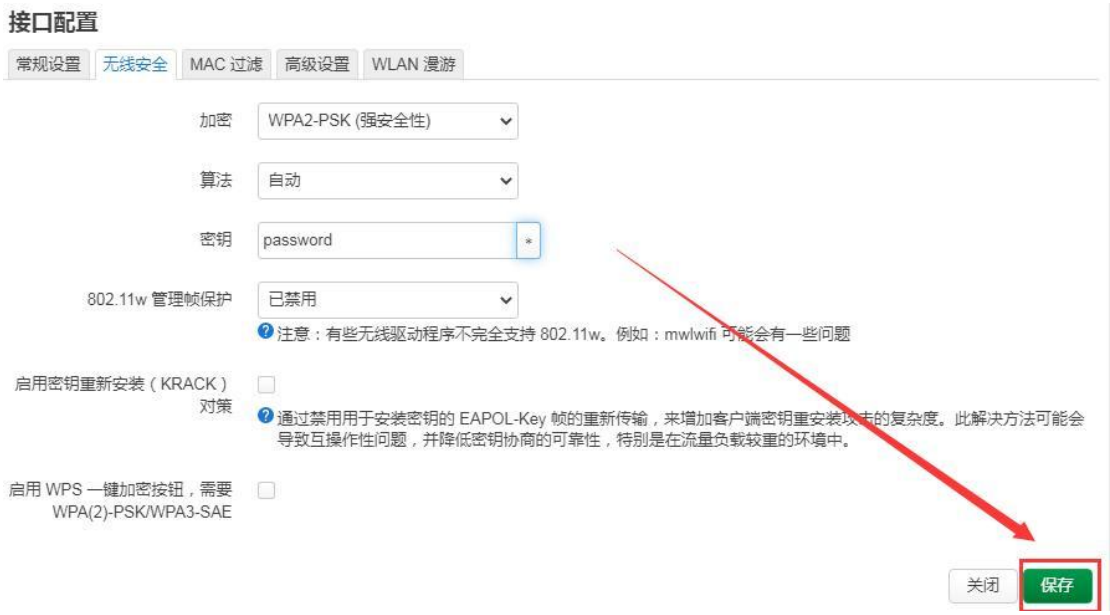


6）If you want to create a WIFI hotspot with a password, click the **Edit** option

7）In the pop-up tab, we click on the **Wireless Security** column.

8）Then in **Interface Configuration -> Wireless Security**, select **WPA2-PSK** as the encryption algorithm; set the key (wireless password) to **password**

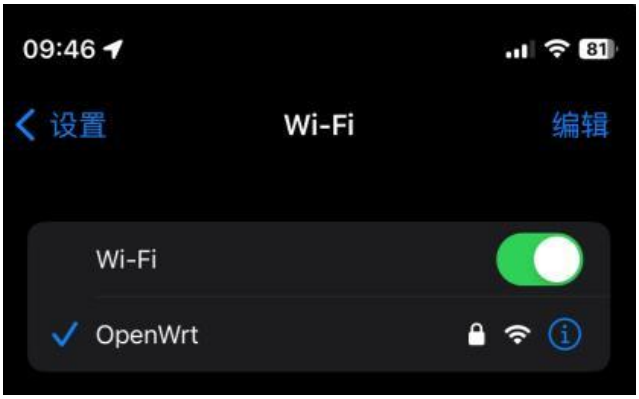
9）After the above settings are completed, click **Save** in the lower right corner of the page, and then exit the tab page



10）Then click **Save and Apply** in the lower right corner of the page and wait for the configuration to be applied.

11）The display interface of successfully creating a hotspot is as shown below. At this time, you can see that the WIFI hotspot has been encrypted.

12）Then use your mobile phone or computer to search for the WiFi corresponding to the SSID and connect. After the connection is successful, as shown in the figure below



8.8.2. How to use USB wireless network card to connect to WIFI hotspot

1）Insert the USB wireless network card into the USB port of the development board, and then connect the power supply to power up the development board.

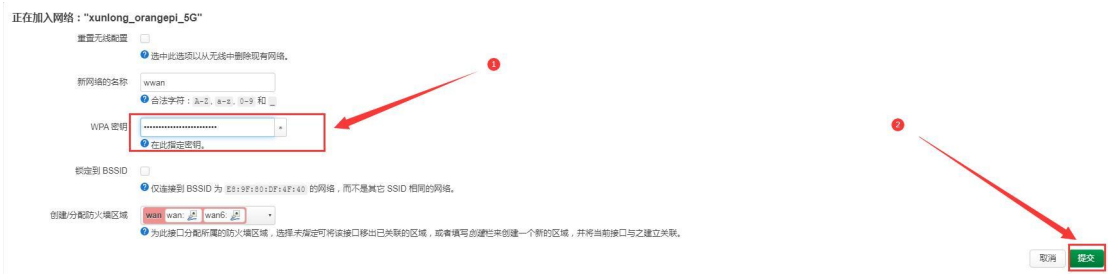
2）After the system startup is completed, click **"Network > Wireless"** to enter the wireless WiFi configuration interface.

3）First, you need to remove the default wireless configuration, click the **"Remove"** button as shown below

4）Then click the **"Scan"** button to scan the surrounding WiFi hotspots

5）Then the following window will pop up to display the available WiFi hotspots. Click the **"Join Network"** button to the right of the WiFi hotspot you want to connect to.

6）Then enter the password as shown in the picture below, and then click **"Submit"**



7）Then the following interface will pop up, click Save

8）Finally, you will return to the main interface of wireless configuration, click **"Save and Apply"**

9）After successfully connecting to the WiFi hotspot, the interface displays as shown below

8. 9. Installing packages via the command line

8. 9. 1. Install through opkg in the terminal

1) Update the list of available software packages

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

2) Get the software list

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

3) Install the specified software package

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

4) View installed software

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

5) Uninstall the software

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

8. 10. OpenWRT management interface installation package

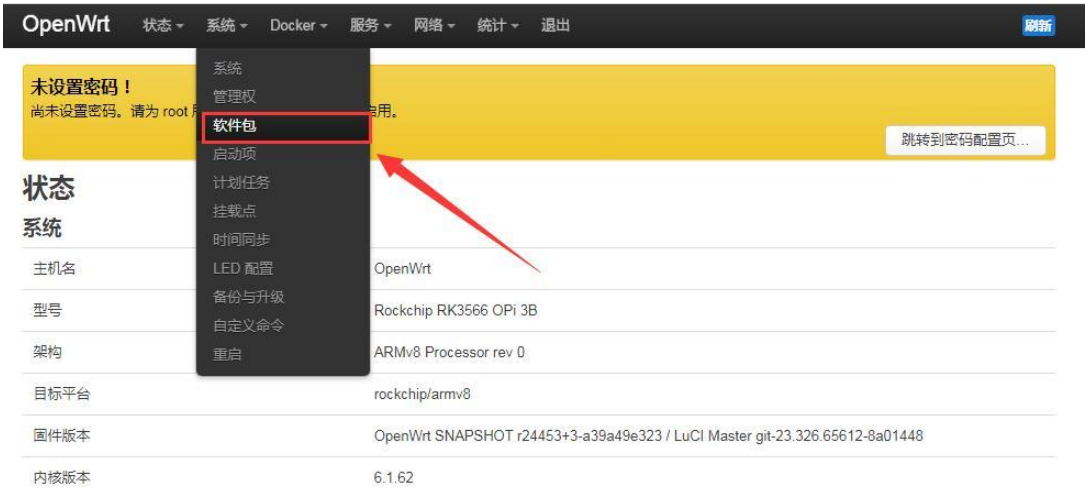
If you need to add a new software package, you can install it through the OpenWRT management interface.

8. 10. 1. View the list of available software packages on the system 系

1) First enter the software package management page

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

b. In the vertical column options below the system, select "**Software Package**" and click to enter



2) Then the main page of the software package will appear, as shown in the figure below, to obtain the list of available software

a. In the "**Action**" option of the software package, click "**Update List**" to obtain the list of available software packages.

b. In the Tab page, click "**Available**" to view the currently available software packages.

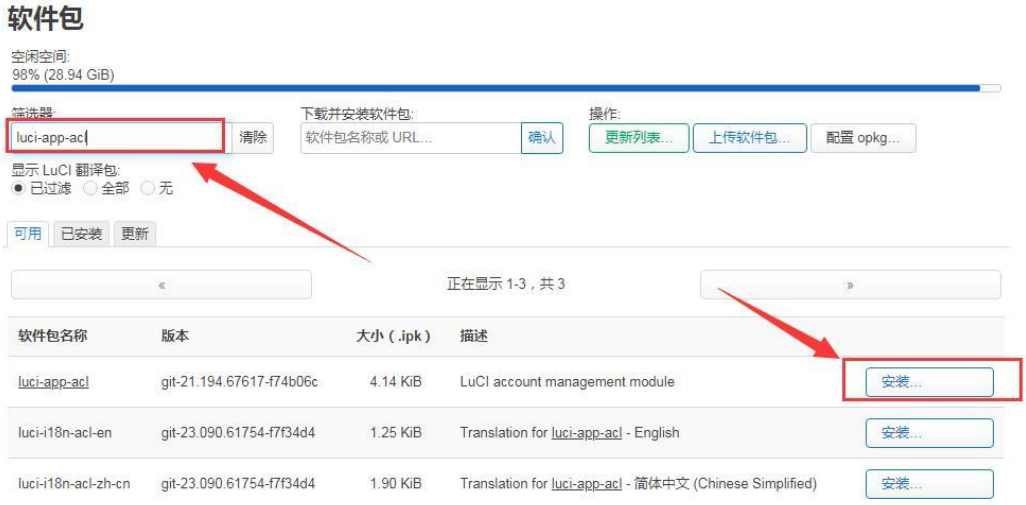
c. View the number of currently available software packages

8. 10. 2. Installation package example

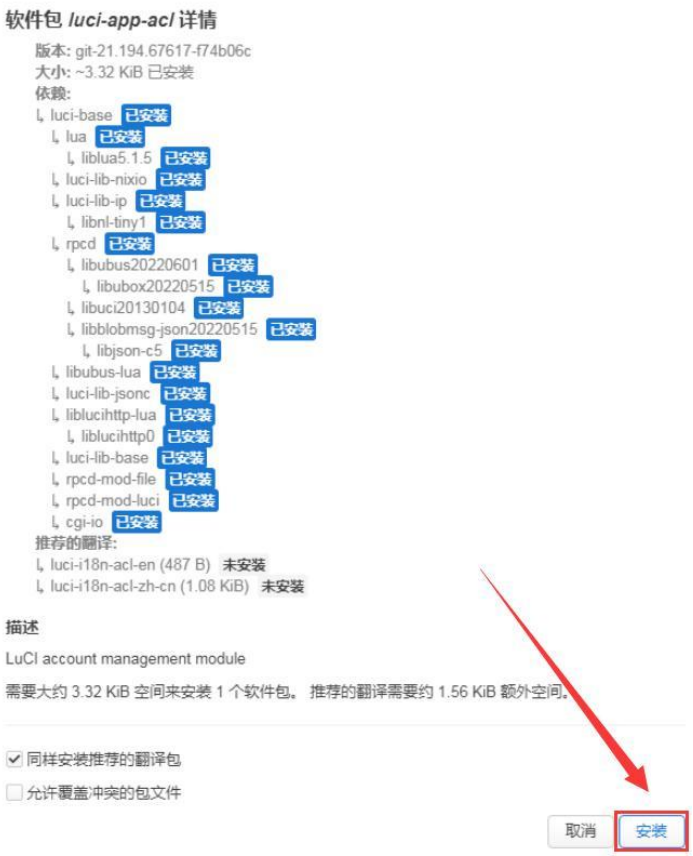
1) Take the installation of the software package "**luci-app-acl**" as an example

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

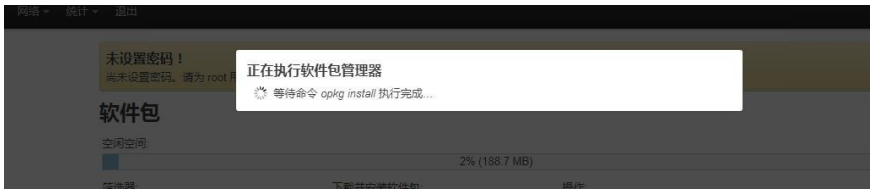
   b. In the list of software packages, you can see the version, package size and description information of the "**luci-app-acl**" software package, and then click the "**Insta**ll" button



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



d. Then wait for the installation to complete



e. The installation completion display is as follows

2) Check whether the software package is installed successfully
   a. In the OpenWRT package management interface, click the filter dialog box and enter "**luci-app-acl**"
   b. Select and click "**Available**" on the Tab page
   c. The "**luci-app-acl**" software package will be displayed in the software package list, and the status will be updated to "**Installed**"”

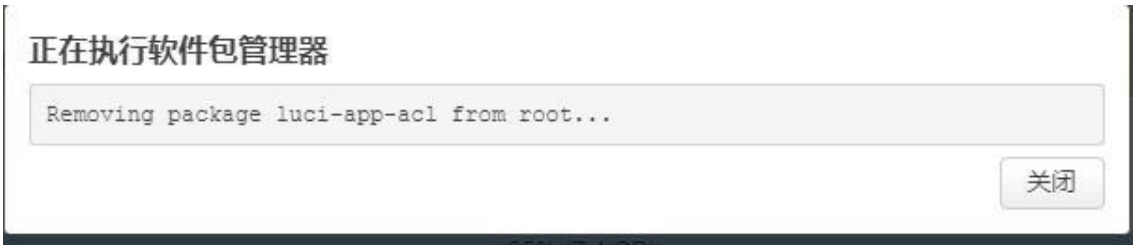
8. 10. 3. Example of removing software packages

1) Take removing the software package "**luci-app-acl**" as an example
   a. In the OpenWRT package management interface, click the filter dialog box and enter "**luci-app-acl**"
   b. Select "**Installed**" on the Tab page to display the list of installed software packages.

c. Click "**Remove**" on the right to remove the corresponding software package

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

b. 移除成功后，显示界面如下



2) Check whether the software package was successfully removed

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

b. Select and click "**Installed**" on the Tab page

c. The "**luci-app-acl**" software package will not be displayed in the software package list. At this time, the "**luci-app-acl**" software package has been successfully removed.

8. 11. Using Samba network sharing

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

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

2) Select the interface that the Samba service needs to monitor
   a. Select "**General Settings**" in the navigation bar of network sharing and click to enter

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

3) Set up a shared directory for network sharing

a. Click "**Add**" shared directory address in "**Shared Directory**" of "**General Settings**" of network sharing.

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

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

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

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

4) Windows 10 starts network discovery and sharing

Note: When accessing Samba and sharing under Windows 10 system, you need

to first confirm whether Windows 10 has enabled network discovery and sharing. If not, perform the following settings first.

a. Enable Samba v1/v2 access

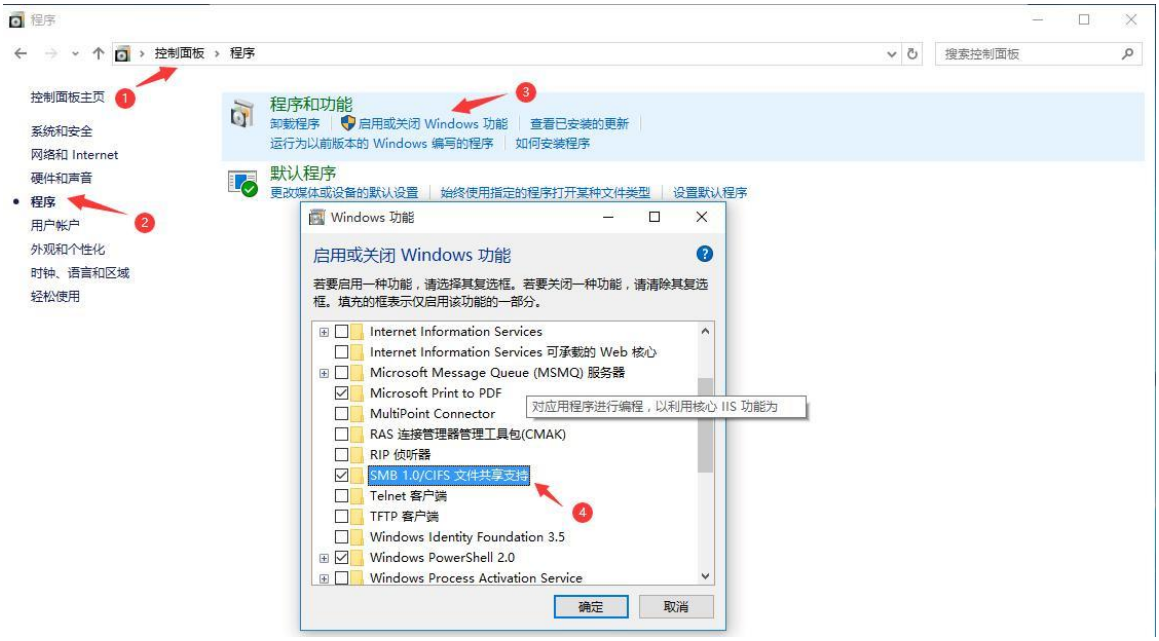
a) Enter the "Control Panel" of Windows 10

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

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

d) Check "SMB 1.0/CIFS file sharing support" in the pop-up box to enable or disable Windows features.

e) Click "OK" to configure the application



b. Turn on network discovery in Windows 10

a) Enter the "Control Panel" of Windows 10

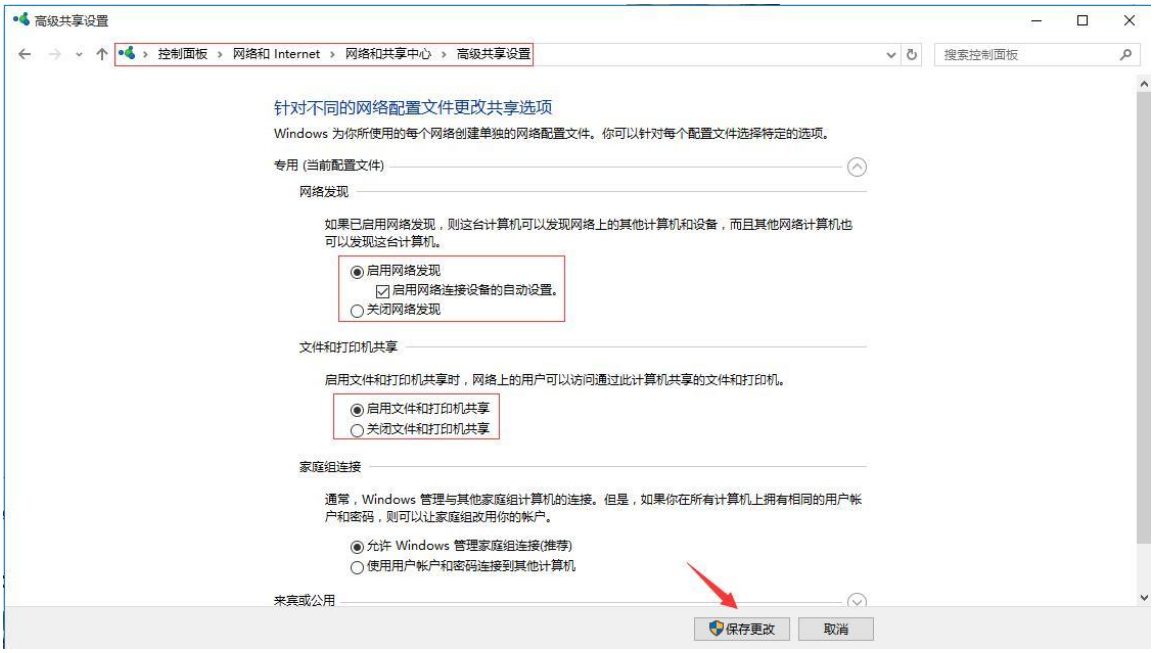
b) Select "Network and Internet" in the Control Panel

c) Then open "Network and Sharing Center"

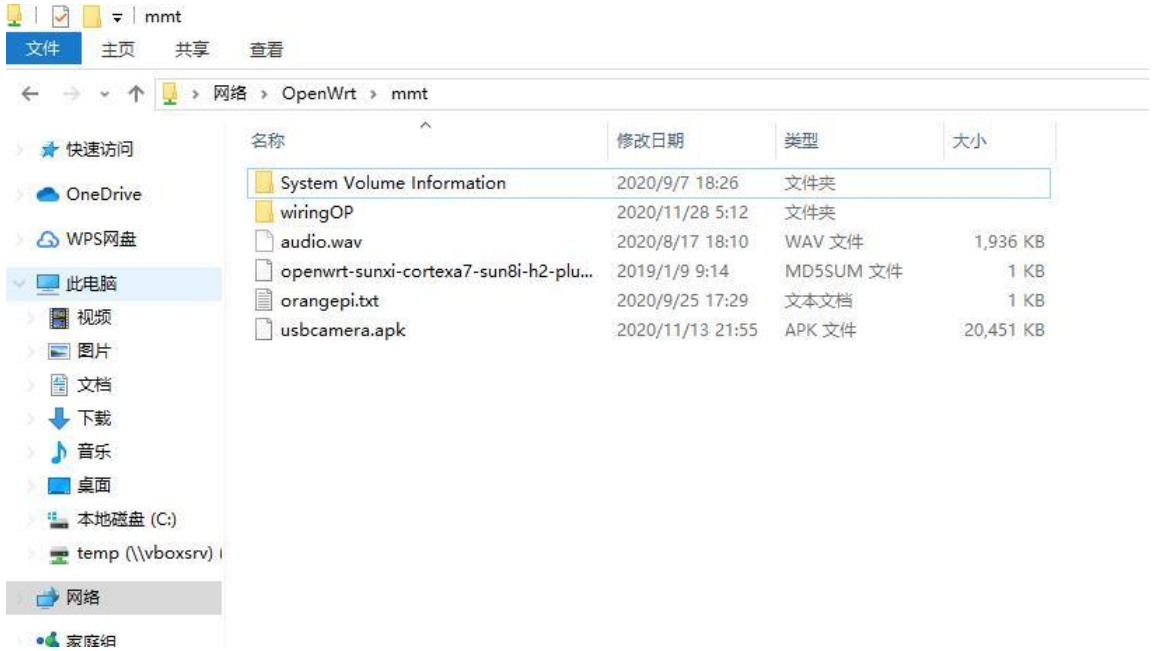
d) Click | "Advanced Sharing Settings"

e) Turn on "**Enable network discovery**" and "**Enable file and printer sharing**"

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



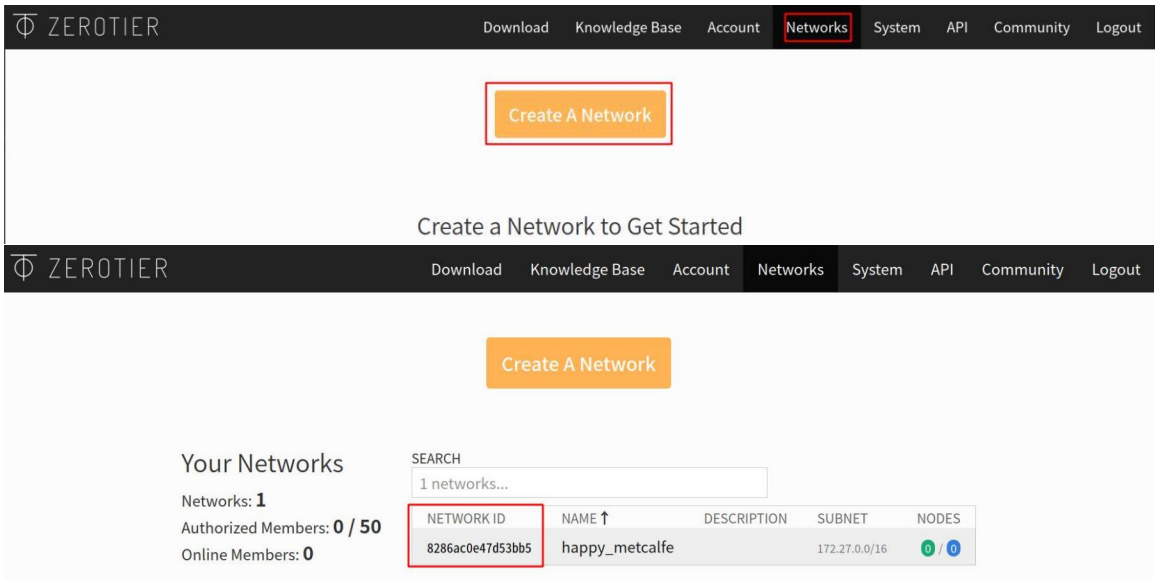
5) After the setting is completed, enter \\OpenWrt in the address bar of the resource manager to access the shared directory. The user name is root, and the password is the password set by the development board host.



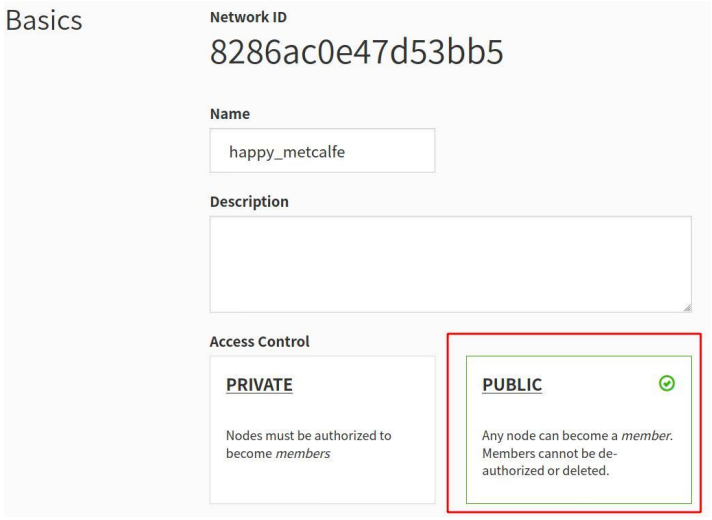
8. 12. Zerotier usage instructions

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

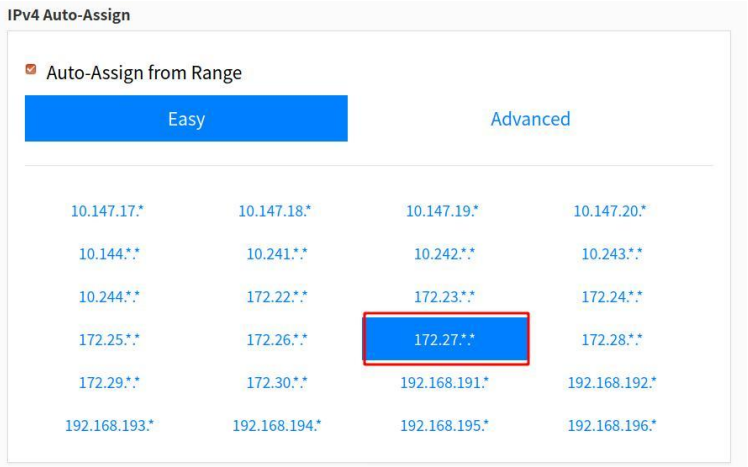
1) Log in to zerotier official website **https://my.zerotier.com/network**, register and click Network->Create A Network to create a virtual LAN



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



3) Next, you can choose the network segment for automatically assigning the address. The selected network segment here is 172.27.*.*



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

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

5) Enter ifconfig in the terminal and you can see that there is a new **ztks54inm2** device with an IP address of **172.27.214.213**

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

6) Install the zerotier client on another device (Ubuntu 18.04 is used as an example here), execute the following command to install. After the installation is completed, you need to restart the computer.

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

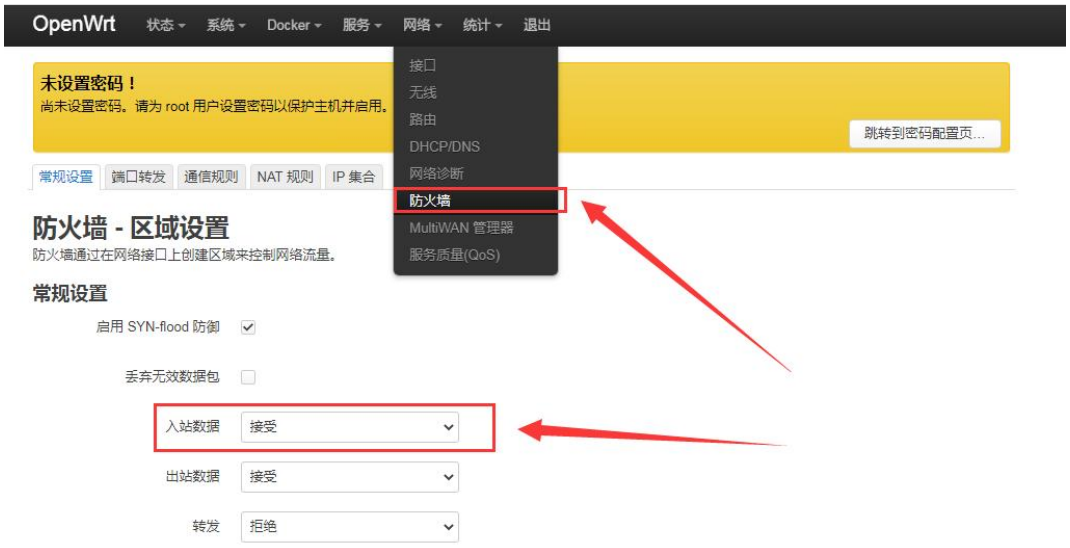
7) After restarting, join the virtual LAN according to the Network ID. You can also see that the IP address assigned by zerotier has been obtained. At this time, the Ubuntu PC and the development board are in the same LAN, and the two can communicate freely.

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

8) Test whether the two terminals can communicate

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

9) If the computer cannot ping the development board, please log in to the LuCI interface and change the inbound data in the firewall configuration to accept



10) Other common commands of zerotier

```
root@OpenWrt:/# zerotier-one -d                    #Start zerotier client
root@OpenWrt:/# zerotier-cli status                #Get address and service status
root@OpenWrt:/# zerotier-cli join # Network ID     #Join the network
root@OpenWrt:/# zerotier-cli leave # Network ID    #Leave the network
root@OpenWrt:/# zerotier-cli listnetworks          #list networks
OPENWRT_DEVICE_REVISION="v0"
OPENWRT_RELEASE="OpenWrt 22.03.4 r20123-38ccc47687"
```

9. How to compile Android11 source code

9. 1. Download the source code of Android 11

1) First download the Android 11 source code sub-volume compressed package from the

Google network disk

a. Google Drive

名称	所有者	上次修改日期	文件大小
RK356X_Android11.tar.gz06	OrangePi	19:55 OrangePi	962.1 MB
RK356X_Android11.tar.gz05	OrangePi	19:31 OrangePi	4 GB
RK356X_Android11.tar.gz04	OrangePi	19:31 OrangePi	4 GB
RK356X_Android11.tar.gz03	OrangePi	19:32 OrangePi	4 GB
RK356X_Android11.tar.gz02	OrangePi	16:37 OrangePi	4 GB
RK356X_Android11.tar.gz01	OrangePi	16:37 OrangePi	4 GB
RK356X_Android11.tar.gz00	OrangePi	16:37 OrangePi	4 GB
RK356X_Android11.tar.gz.md5sum	OrangePi	16:37 OrangePi	420 个字节

2) After downloading the sub-volume compression package of the Android 11 source code, please check whether the MD5 checksum is correct, if not, please download the source code again

```
test@test:~$ md5sum -c RK356X_Android11.tar.gz.md5sum
RK356X_Android11.tar.gz00: OK
RK356X_Android11.tar.gz01: OK
RK356X_Android11.tar.gz02: OK
RK356X_Android11.tar.gz03: OK
RK356X_Android11.tar.gz04: OK
RK356X_Android11.tar.gz05: OK
RK356X_Android11.tar.gz06: OK
```

3) Then you need to merge multiple compressed files for decompression

```
test@test:~$ cat RK356X_Android11.tar.gz0* | tar -xvzf -
```

9. 2. Compile the source code of Android 11

1) First install the software packages required to compile the Android11 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 libncurses5 \
lib32ncurses5-dev x11proto-core-dev libx11-dev lib32z1-dev ccache \
libgl1-mesa-dev libxml2-utils xsltproc unzip liblz4-tool
```

2) There is a build.sh compilation script in the source code, and the compilation parameters are as follows

a. **-U**：Compile uboot
b. **-K**：Compile kernel
c. **-A**：compile android
d. **-u**：Package and generate update.img and update_spi_nvme.img
e. **-o**：Compile OTA package
f. **-d**：Specify kernel dts

3) Compile uboot, kernel, android and package them into update.img

a. The command to compile and support HDMI 4K display mirroring (LCD is turned off by default) is as follows:

```
test@test:~$ cd RK356X_Android11
test@test:~/ RK356X_Android11$ export BOARD=orangepi3b
test@test:~/ RK356X_Android11$ source build/envsetup.sh
test@test:~/ RK356X_Android11$ lunch rk3566_r-userdebug
test@test:~/ RK356X_Android11$ ./build.sh -AUKu
```

b. The command to compile and support LCD display mirroring (HDMI is disabled by default) is as follows:

```
test@test:~$ cd RK356X_Android11
test@test:~/ RK356X_Android11$ export BOARD=orangepi3b
test@test:~/ RK356X_Android11$ export DUAL_LCD=true
test@test:~/ RK356X_Android11$ source build/envsetup.sh
test@test:~/ RK356X_Android11$ lunch rk3566_r-userdebug
test@test:~/ RK356X_Android11$ ./build.sh -AUKu
```

4) After the compilation is complete, the following information will be printed

```
********rkImageMaker ver 2.1********
Generating new image, please wait...
storage is spinor
Writing head info...
Writing boot file...
Writing firmware...
```

```
Generating MD5 data...
MD5 data generated successfully!
New image generated successfully!
********rkImageMaker ver 2.1********
Merging storage firmware, please wait...
storage count = 2
adding spinor_update.img...ok
adding pcie_update.img...ok
Merging firmware success.
Making update_spi_nvme.img OK.
Make update image ok!
/wspace3/RK3566/RK356X_Android11
```

5) The final image file will be placed in the **rockdev/Image-rk3566_r/** directory. Among them, **update.img** is the boot image that supports TF card and eMMC, and **update_spi_nvme.img** is the boot image of NVME SSD

```
test@test:~/RK356X_Android11$ cd rockdev/Image-rk3566_r
test@test:~/RK356X_Android11/rockdev/Image-rk3566_r $ ls update*
update.img update_spi_nvme.img
```

10. Compilation method of OpenWRT source code

10.1. Download OpenWRT source code

1) First execute the following command to download the source code

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

2) After the OpenWRT code is downloaded, it will contain the following files and folders

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

10. 2. Compile OpenWRT source code

1) First install the following dependent software (currently only tested on Ubuntu22.04, you need to install the following software. If you compile on other versions of the system, please install the dependent software yourself according to the error message)

```
test@test:~/openwrt$ sudo apt update
test@test:~/openwrt$ sudo apt install -y ack antlr3 asciidoc autoconf    \
automake autopoint binutils bison build-essential    \
bzip2 ccache cmake cpio curl device-tree-compiler fastjar \
flex gawk gettext gcc-multilib g++-multilib git gperf haveged \
help2man intltool libc6-dev-i386 libelf-dev libglib2.0-dev \
libgmp3-dev libltdl-dev libmpc-dev libmpfr-dev \
libncurses5-dev \libncursesw5-dev libreadline-dev libssl-dev \
libtool lrzsz mkisofs msmtp nano ninja-build p7zip p7zip-full \
patch pkgconf python2.7 python3 python3-pyelftools \
libpython3-dev qemu-utils rsync scons squashfs-tools \
subversion swig texinfo uglifyjs upx-ucl unzip \
vim wget xmlto xxd zlib1g-dev
```

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

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

3) Then choose to use the configuration file of OrangePi 3B

```
test@test:~/openwrt$ cp configs/orangepi-3b-rk3566_defconfig .config
```

4) Then execute the following command to make the configuration take effect

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

5) Execute the following command to start compiling the openwrt source code

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

6) After compilation is completed, the path where the image is generated is:

```
test@test:~/openwrt$ tree -L 1 bin/targets/rockchip/armv8/
```

```
bin/targets/rockchip/armv8/
├──── config.buildinfo
├──── feeds.buildinfo
├──── openwrt-rockchip-armv8-xunlong_orangepi-3b-ext4-sysupgrade.img.gz
├──── openwrt-rockchip-armv8-xunlong_orangepi-3b.manifest
├──── openwrt-rockchip-armv8-xunlong_orangepi-3b-squashfs-sysupgrade.img.gz
├──── packages
├──── profiles.json
├──── sha256sums
└──── version.buildinfo

1 directory, 9 files
```

11. Appendix

11. 1. User Manual Update History

Version	Date	Update Notes
v1.0	2023-08-17	initial version
v1.1	2023-08-24	1. Compilation method of Android11 source code 2. Android11：The method of USB OTG mode switching 3. Android11：The method of using the data cable to connect to adb debugging
v1.2	2023-08-25	1. Added instructions for purchasing PCIe NVMe SSDs
v1.3	2023-09-05	1. Instructions for using the Orange Pi OS Arch system
v1.4	2023-09-21	1. Linux：How to create a WIFI hotspot through create_ap
v1.5	2023-11-08	1. How to burn Orange Pi OS (OH) image to TF card 2. How to burn Orange Pi OS (OH) image into eMMC 3. Orange Pi OS OH system usage instructions

		4. Added instructions for Linux 6.6 system
v1.6	2023-11-24	1. How to use wiringOP hardware PWM 2. Instructions for using the OpenWRT system 3. How to compile OpenWRT source code
v1.7	2024-09-24	1. v2.1 Hardware Version Manual Image Update

11.2. Image Update History

Date	Update Notes
2023-08-17	Orangepi3b_1.0.0_ubuntu_focal_server_linux5.10.160.7z Orangepi3b_1.0.0_ubuntu_jammy_server_linux5.10.160.7z Orangepi3b_1.0.0_debian_bullseye_server_linux5.10.160.7z Orangepi3b_1.0.0_debian_bookworm_server_linux5.10.160.7z Orangepi3b_1.0.0_ubuntu_focal_desktop_xfce_linux5.10.160.7z Orangepi3b_1.0.0_ubuntu_jammy_desktop_xfce_linux5.10.160.7z Orangepi3b_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160.7z Orangepi3b_1.0.0_debian_bookworm_desktop_xfce_linux5.10.160.7z OrangePi3B_RK3566_Android11_v1.0.0.tar.gz OrangePi3B_RK3566_Android11_lcd_v1.0.0.tar.gz OrangePi3B_RK3566_Android11_spi-nvme_v1.0.0.tar.gz OrangePi3B_RK3566_Android11_lcd_spi-nvme_v1.0.0.tar.gz * initial version
2023-08-23	Opios-arch-aarch64-xfce-opi3b-23.08-linux5.10.160.img.xz * initial version
2023-08-24	OrangePi3B_RK3566_Android11_v1.0.1.tar.gz OrangePi3B_RK3566_Android11_lcd_v1.0.1.tar.gz OrangePi3B_RK3566_Android11_spi-nvme_v1.0.1.tar.gz OrangePi3B_RK3566_Android11_lcd_spi-nvme_v1.0.1.tar.gz * Support USB OTG mode switching function
2023-08-25	Opios-arch-aarch64-xfce-opi3b-23.08.1-linux5.10.160.img.xz

	* Solve the problem that the app store cannot be used
2023-09-21	Orangepi3b_1.0.2_ubuntu_focal_server_linux5.10.160.7z Orangepi3b_1.0.2_ubuntu_jammy_server_linux5.10.160.7z Orangepi3b_1.0.2_debian_bullseye_server_linux5.10.160.7z Orangepi3b_1.0.2_debian_bookworm_server_linux5.10.160.7z Orangepi3b_1.0.2_ubuntu_focal_desktop_xfce_linux5.10.160.7z Orangepi3b_1.0.2_ubuntu_jammy_desktop_xfce_linux5.10.160.7z Orangepi3b_1.0.2_debian_bullseye_desktop_xfce_linux5.10.160.7z Orangepi3b_1.0.2_debian_bookworm_desktop_xfce_linux5.10.160.7z * Solve the problem of CPU frequency being limited to 1.2GHz * Add rk356x-uart2-m0.dtbo
2023-11-06	Opios-openharmony-4.0-beta1-aarch64-opi3b-23.11-linux5.10.img.tar.gz * initial version
2023-11-08	Orangepi3b_1.0.0_ubuntu_jammy_desktop_xfce_linux6.6.0-rc50.7z Orangepi3b_1.0.0_debian_bullseye_desktop_xfce_linux6.6.0-rc5.7z Orangepi3b_1.0.0_debian_bookworm_desktop_xfce_linux6.6.0-rc5.7z * initial version
2023-11-08	Opios-oh-4.0-beta1-aarch64-opi3b-23.11.1-linux5.10.img.tar.gz * Support TF card boot
2023-11-24	openwrt-aarch64-opi3b-23.05-linux6.1.62-ext4.img.gz * initial version

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.