Orange Pi Zero Plus
User Manual

V1.1

Contents

I. Orange Pi Zero Plus Introduction

1. What is Orange Pi Zero Plus?

It's an open-source single-board computer. It can run Android 5.1, Ubuntu, Debian, Armbian Image. It uses the AllWinner H5 SoC, and has 512MB DDR3 SDRAM.

2. What can I do with Orange Pi Zero Plus?

You can use it to build…

- A computer
- A wireless server
- Games
- Music and sounds
- HD video
- A speaker
- Android
- Scratch
- .....

Pretty much anything else, because Orange Pi Zero Plus is open source

3. Whom is it for?

Orange Pi Zero Plus is for anyone who wants to create with technology– not just consuming. It's a simple, fun, useful tool and you can use it to take control of the world around you.

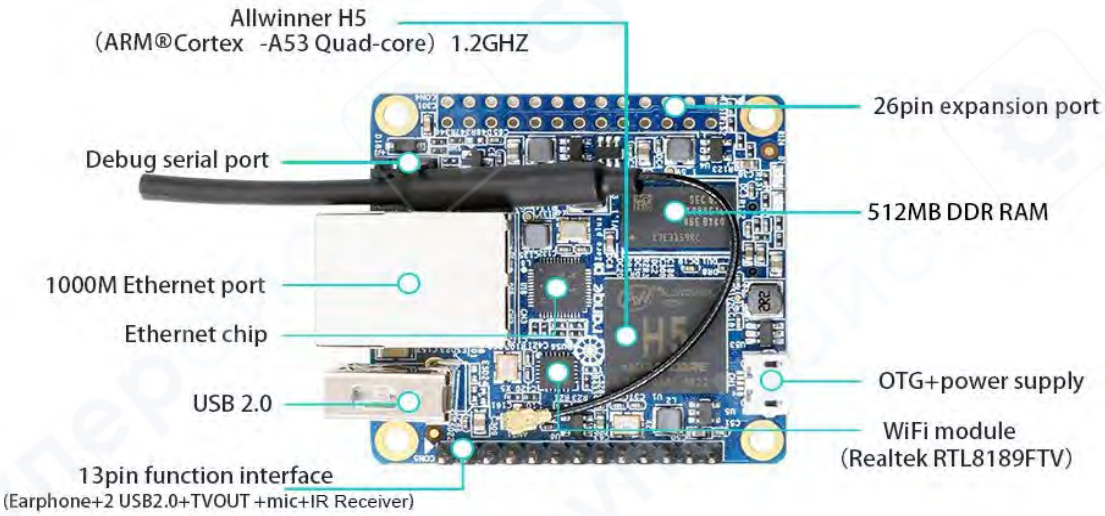
4. Hardware specification of Orange Pi Zero Plus

Hardware specification	
CPU	H5 Quad-core 64-bit Cortex-A53
GPU	Mali450MP2 GPU @600MHz, Supports OpenGL ES 2.0
Memory (SDRAM)	512MB DDR3 (shared with GPU)
Onboard Storage	TF card (Max. 32GB) /Spi flash(2MB)
Wifi Antenna	Yes

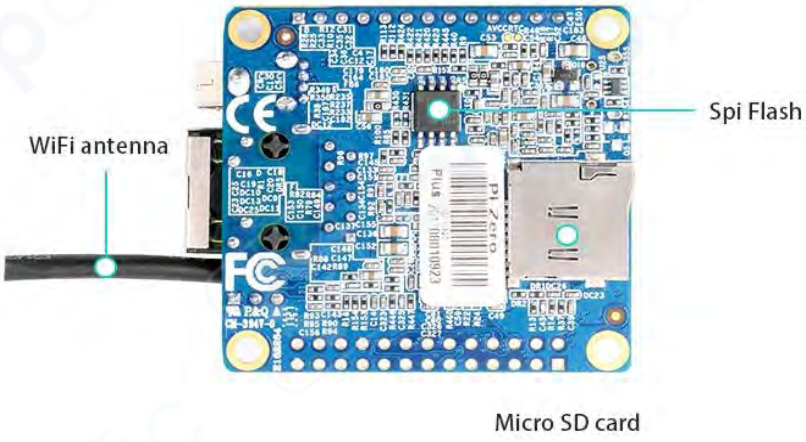
Onboard Network	1000M/100M/10M Ethernet RJ45
Onboard WIFI	Realtek RTL8189FTV,IEEE 802.11 b/g/n
Video Outputs	Support output via 13pin
Power Source	USB OTG can supply power
Buttons	Power button
Low-level peripherals	26 Pins Header, compatible with Raspberry Pi B+ 13 Pins Header, with 2x USB, IR pin, AUDIO
GPIO(1x3) pin	UART, ground.
LED	Power led & Status led
Supported OS	Android5.1, Lubuntu, Debian, Armbian Image
Interface definition	
Product size	45mm × 48mm
Weight	48g
Orange Pi™ is a trademark of the Shenzhen Xunlong Software CO., Limited	

Interface instructions

Top view

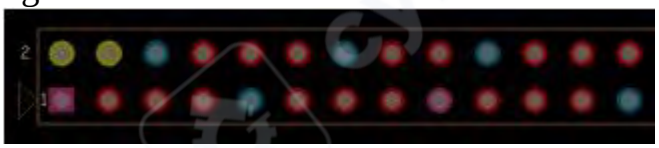


Bottom view



5. GPIO Specifications

A 26-pin GPIO interface on the Orange Pi Zero Plus is the same as Model A and Model B of Raspberry Pi. The picture below is GPIO pin definition of Orange Pi Zero Plus.



OrangePi_Zero Plus(H2)		
CON3-P01	VCC-3V3	
CON3-P02	VCC-5V	
CON3-P03	TWI0-SDA	PA12
CON3-P04	VCC-5V	
CON3-P05	TWI0-SCK	PA11
CON3-P06	GND	
CON3-P07	PWM1	PA6
CON3-P08	UART1_TX	PG6
CON3-P09	GND	
CON3-P10	UART1_RX	PG7
CON3-P11	UART2_RX	PA1
CON3-P12	PA7	PA7
CON3-P13	UART2_TX	PA0
CON3-P14	GND	
CON3-P15	UART2_CTS	PA3
CON3-P16	TWI1-SDA	PA19
CON3-P17	VCC3V3-EXT	
CON3-P18	TWI1-SCK	PA18
CON3-P19	SPI1_MOSI	PA15
CON3-P20	GND	
CON3-P21	SPI1_MISO	PA16
CON3-P22	UART2_RTS	PA2
CON3-P23	SPI1_CLK	PA14
CON3-P24	SPI1_CS	PA13
CON3-P25	GND	
CON3-P26	PA10	PA10

II. Using Method Introduction

Follow these steps, you can configure and run your Orange Pi in a very short period of time. Boot your Orange Pi need to complete the following steps.

1. Step 1: Prepare Accessories Needed

You need at least some accessories like the following if it is your first time to use the Orange Pi (we would suggest you using the Expansion boardat the same time).

No.	Items	Requirements and Instructions
1	TF card	8GB min.; class 10. Branded TF cards would be reference which are much more reliable.
2	AV video cable	A standard AV video cable can be used to connect stimulated monitor if a HDMI monitor is unavailable.
3	Keyboard and mouse	Any keyboard and mouse with USB port is applicable; Keyboard and mouse are high-power, so a USB concentrator is required.
4	Ethernet cable(Optional)	Network is optional, It makes more convenient to mount and upgrade software in your Orange Pi PC.
5	DC power adapter	5V/2V min. high qualified power adapter, OTG can used a power supply.
6	Audio cable (Optional)	You can select an audio cable with 3.5mm jack to feel stereo audio.

AV video cable

TF card

DC power adapter

Expansion Board

2. Step 2: Prepare a TF Card

In order to use Orange Pi normally, you must install the operating system into TF card first.

1) Write Linux into TF Card Based on Windows Platform

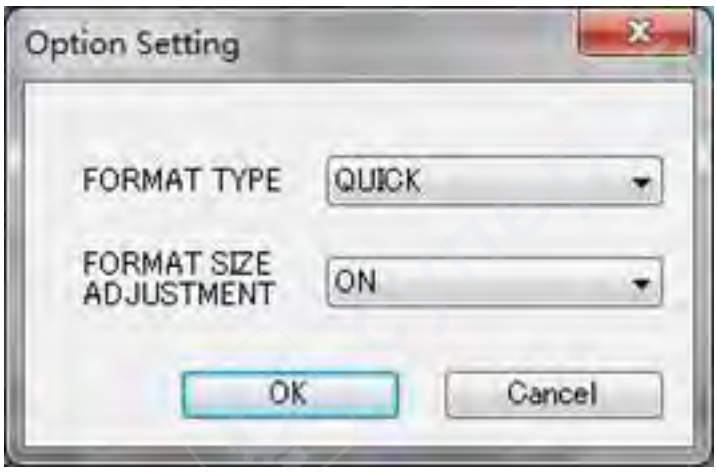
a. Inserting the TF card into the computer, the capacity of the card must be bigger than the operating system, usually requires 8GB or bigger.

b. Formatting the TF card.

i Download tools for formatting TF card, such as TF Formatter, it could be downloaded from:
https://www.sdcard.org/downloads/formatter_4/eula_windows/

ii Unzip the downloaded files, and run *setup.exe*

iii In the *options settings* select the "*form*at" button for quick formatting. "*Format size adjustment*" select "(ON)"



iv Make sure the inserted TF card disk are in accordance with the chosen disk.

v Click the "*Format*" button.

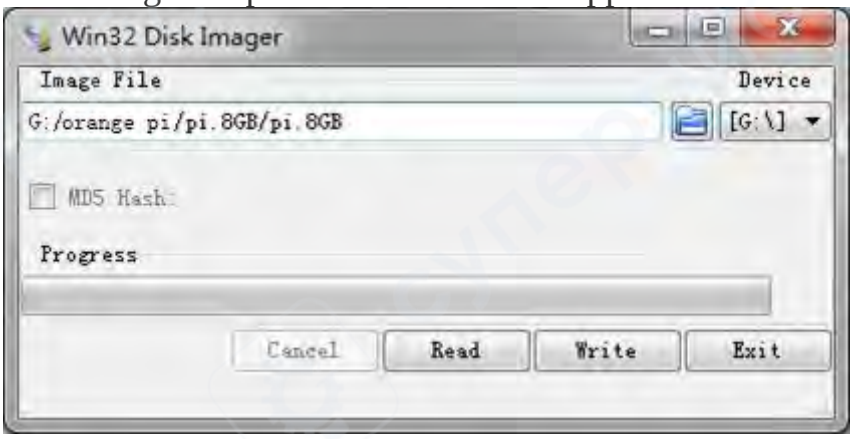
c. Download the operating system image file from the download page, the page address:

d. Unzip the downloaded file (in addition to the Android system, this method can be used to burn to write, the Android system need another burn, the following will introduce)

e. Right click to download the file, select "*Unzip file*" to write image to TF card

i Download tools to write image,such as *Win32 Diskimager*, here is the download page:
http://sourceforge.net/projects/win32diskimager/files/Archive/

ii Select the image file path that has been unzipped.



iii Click "*Write*" button and wait for the image to write.

iv After the image is written, click "*Exit*" button.

2) Write Linux into TF card based on Linux platform?

a. Inserting the TF card into the computer, the capacity of the card must be larger than the operating system image, usually requires 4GB or greater capacity.

b. Formatting the TF card.

i Run ***fdisk –l*** order to make sure TF disk.

ii Run ***umount /dev/sdxx*** to uninstall all partitions of TF Card.

iii Run ***sudo fdisk /dev/sdx*** order. Use ***o*** command to delete all partitions of TF Card, and then us ***n*** order to add a new partition, finally use ***w*** command to save and exit.

iv Run ***sudo mkfs.vfat /dev/sdx1*** command to format the TF card partition set up last step to FAT32 form(according to your TF card disk to replace***x***). Or you could skip this step since command in Linux will format TF card automatic.

c. Download the OS image from download page

d. Unzip and right click the downloaded file, select " *Unzip file*"

e. Write image to TF card

i Run **sudo fdisk –l** order to make sure the TF card disk

ii make sure the image file **hash key** is the same as download page mention(optional). It will output ***sha1sum [path]/[imagename]***, which should be same as the image paye *"SHA-1"*

iii Run ***umount /dev/sdxx*** order to uninstall all partitions in TF Card

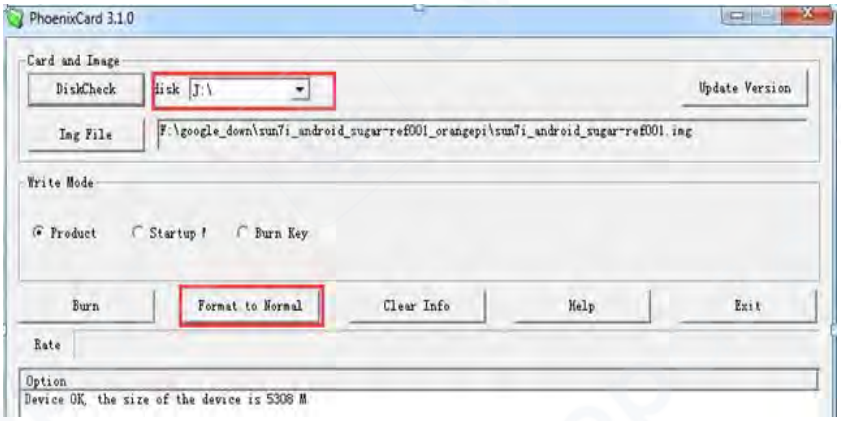
iv Run ***sudo dd bs=4M if=[path]/[imagename] of=/dev/sdx*** to write down image file. Wait for the image to write. If it cannot work at 4M, then replace a 1M which takes more time. You can run ***sudo pkill –USZero Plus –n –x dd*** order to monitoring procedure.

3) Use PhoenixCard tool to write Android image into TF card

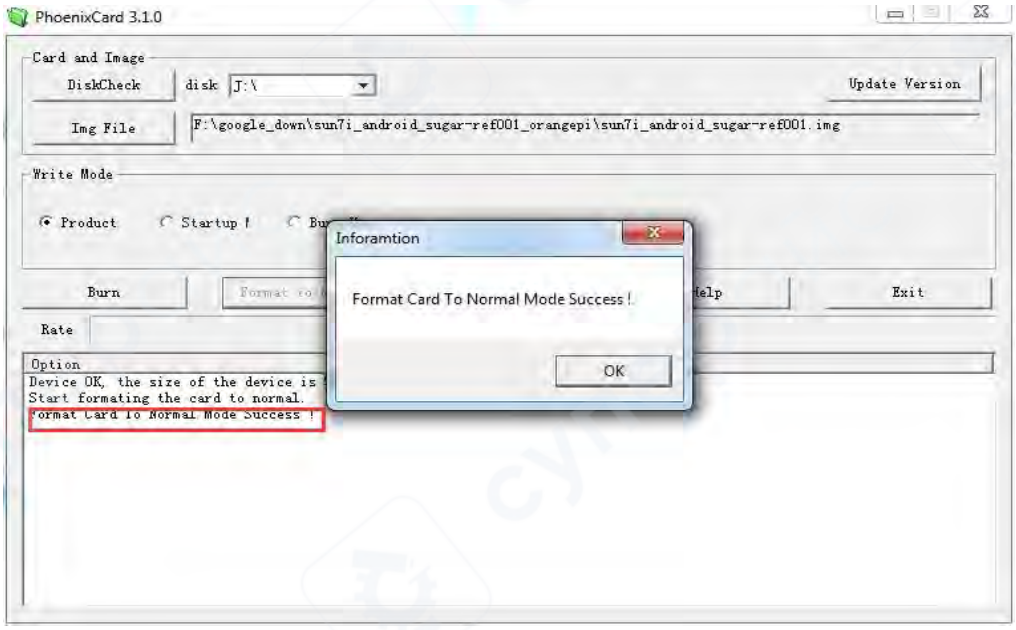
It is impossible for Android image to be written into TF card by using *dd* command under Linux or using *Win32 Diskimager* under Windows. Here

PhoenixCard tool is applicable for Android image writing.

a. Download the Android OS image and **PhoenixCard** tool.
   Download **PhoenixCard.**
   Download Android OS image.
b. Format the TF card

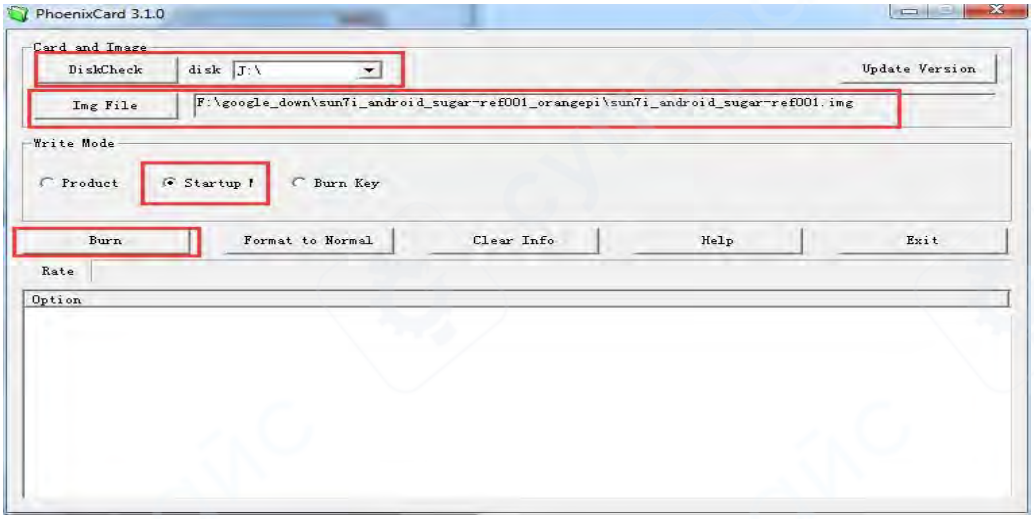


c. Please make sure the inserted TF card is in accordance with the chosen TF card, click "format to normal" button for TF card formatting.

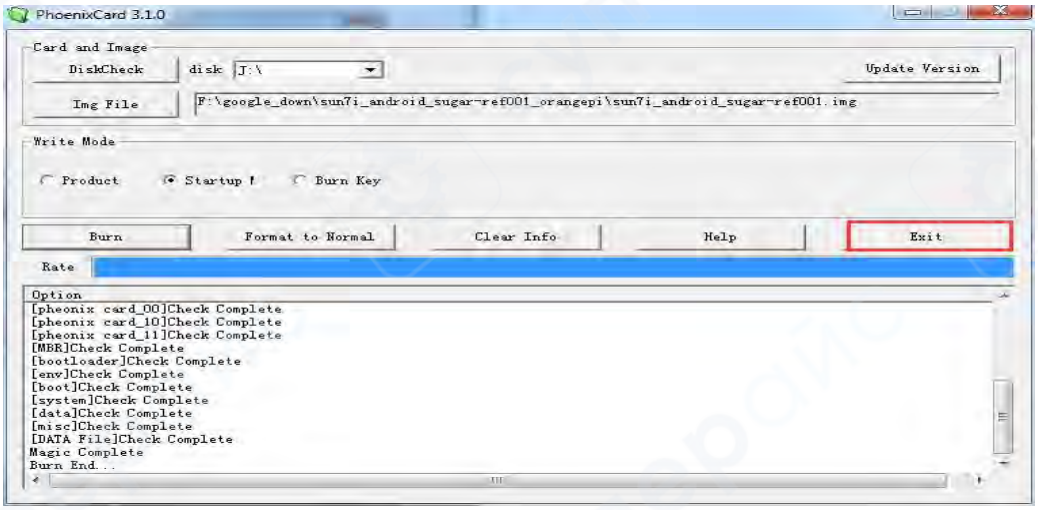


d. Click "*OK*" button after successfully formatted the TF card to normal.

e. Burn the Android OS image into your TF card. Please pay attention to the following with red marks.



f. Click "*Burn*" button for writing to TF card and wait for it finish



g. Click "*Exit*" button after burn Android image to TF card successfully.

4) Write Armbian Image into TF Card

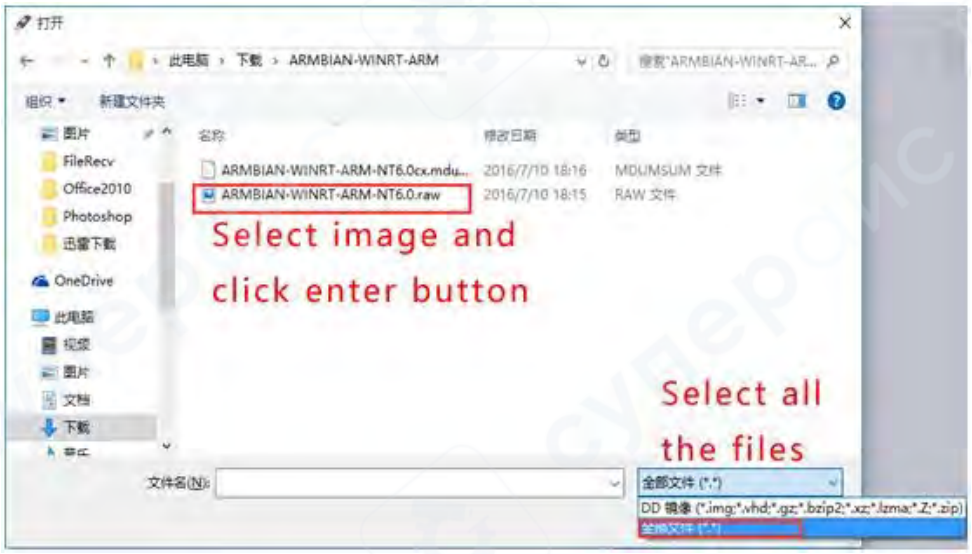
a. Insert TF card into computer, please note that the TF card capacity must bigger than the operating system image, usually need to be 8GB or bigger.

b. Download the OS image file from the download page:
http://www.armbian.com/download/

c. Write the image into TF card.

i Download image writing tool such as *Rufus*, the download page: https://rufus.akeo.ie/

ii Select the image file path that has been unzipped



iii Click "*start*" button and wait for the image to write.

iv After the image is written, click "*close*" button

3. Step 3: Boot your Orange Pi

1) Hardware Connection

Orange Pi Zero Plus runs on Android 5.1 system

Orange Pi Zero Plus runs on Debian system

Orange Pi Zero Plus runs on Ubuntu system

2) Details of Booting Steps

a. Insert the TF card with written image in to the TF card slot.
b. You could use HDMI cable to connect your Orange Pi to HDMI TV or monitor.

You could also use AV interface and audio interface to connect output

video and audio to analog TV or display.

c. There is 13pin on board which you could connect to expansion board. For expansion board, 2USB ports, mic and IR receiver are available.

d. It is the network module on board, which you can access Orange Pi to the wired network.

e. You could connect to a power adapter on mic USB OTG with a power adapter up to or bigger than 5V/2A. Avoid using smaller power GSM mobile phone charger, it is not able to output 2A even if it marked "5V/2A ".

The Orange Pi will boot in a few minutes If the above steps are successful. There will be graphical interface in he monitor. It may take a long time to start the first time, please wait patiently. The next time will boot very fast.

4. Step 4: Turn off your Orange Pi Correctly

- You can use the shutdown button on the interface to safety close the Orange Pi.
- You can also close the system by entering commands in the shell:

sudo halt
or
sudo shutdown –h

It will be safety to turn off the Orange Pi. If directly use the power button to shut down the system may damage the file system on TF Card. After the system is closed, the power can be cut off by more than 5 seconds' press.

5. Other configuration

1) Connect to the wired network

- Method 1:

a. Enter the following in the command line:
$ ifconfig
To check whether there is (wlan*)
b. If no, load the corresponding module corresponding to the wlan model

$ insmod 8189fs.ko

For example: For RTL8189 FTV is corresponding to 8189fs.ko

c. Enter command ifconfig, you should see wlan0(hypothesis it is wlan0)

d. Configure wireless network, first you need to know ssid and psk(account and password), enter corresponding wlan*, ssid, psk

```
$ sudo nano /etc/network/interfaces      (add the following contents)
auto wlan0
iface wlan0 inet dhcp
wpa-ssid xxxx
wpa-psk xxxx
```

e. Reboot the computer and the wired network will work.

```
$ sudo reboot
```

- Method 2:

a. Build wifi hotspot configuration file of wpa_supplication.conf for on /etc/network/ directory and add the following:

```
network={
        ssid="wifi hot spot name"
        psk="wifi hot spot password"
        priority=1
}
```

b. Connect wifi, here is the command:

```
ifconfig wlan0 up
sudo wpa_supplicant -i wlan0 -c /etc/network/wpa_supplication.conf &
dhcpcd wlan0 &
```

c. Test the condition of wifi connection

Use iwconfig command, you will find the related information of wlan0, use ping command to test.

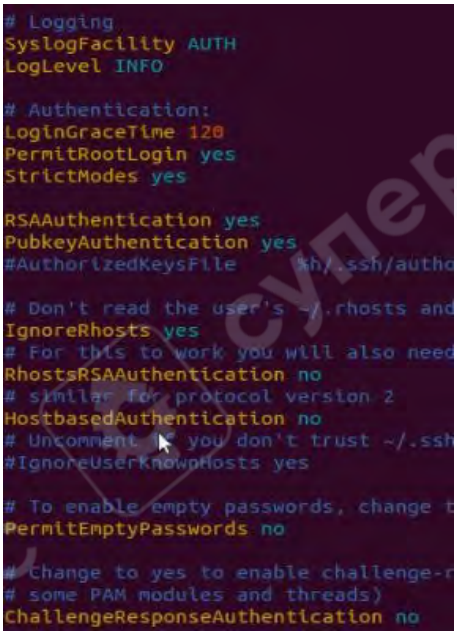
2) Login via vnc and ssh

If there is no condition for connecting HDMI, you could enter the system via vnc or ssh remote login.

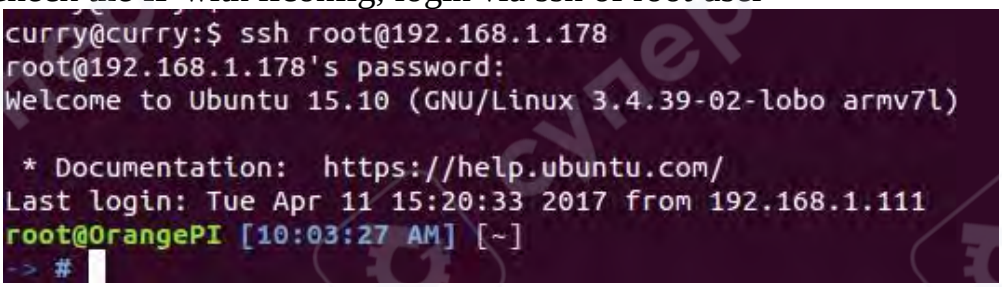
- Login via serial port and install ssh

```
apt-get install ssh
```

- Modify ssh configuration file /etc/ssh/sshd_config



- Check the IP with ifconfig, login via ssh of root user



3) HDMI or 3.5mm Sound Output(3.5mm sound output would require using an expansion board)

a. The sound was default to output via HDMI on image, it could check and change via alsamixer.

ls /etc/asound.conf

card indicates card number, device indicates device number.

aplay -l it could check the system to load the sound card number and details

cat /proc/asound/cards it also could check the sound card and details

It could be used after use alsamixer to change the sound card.

alsactl store -f /var/lib/alsa/asound.state used for saving modified parameters

b. It needs to modify configuration on file system for output on 3.5mm of /etc/asound.conf, modify card1 into card0, or use amixer to modify. The default one is configured, or you could use player on graphical interface to switch via sound channel selection.

c. How to use mic sound recording
arecord -d 5 -f cd -t wav 123.wav
After recording, use the following to play
aplay 123.wav

6. Universal Software Configuration

1) Default Account Changing

The default log in account is orangepi. In order to secure, it is recommended to modify the default orangepi accounts to your own account, for example Zhangsan. Steps are as follows:

a. Use root account to login Orange Pi(please note that do not login with the account of orangepi)
b. $ usermod -l zhangsan orangepi Change orangepi account into Zhangsan

```
@orangepi:~$ usermod -l zhangsan orangepi
```

c. $ groupmod -n zhangsan orangepi Change group

```
@orangepi:~$ groupmod -n zhangsan orangepi
```

d. $ mv /home/ornagepi /home/zhangsan Change directory of original orangepi

```
@orangepi:~$ mv /home/orangepi /home/zhangsan
```

e. $ usermod -d /home/orangepi orangepi Set this directory to orangepi user's home directory

```
@orangepi:~$ usermod -d /home/zhangsan zhangsan
```

f. $ cat /etc/passwd It should be shown as below:

```
pulse:x:112:121:PulseAudio daemon,,,:/var/run/pulse:/bin/false
zhangsan:x:1001:1001:orangepi,,,,:/home/zhangsan:/bin/bash
```

After the modification of the above iterms, it can be used the new account Zhangsan to land.

2) U Disk Automatic Mounted Configuration

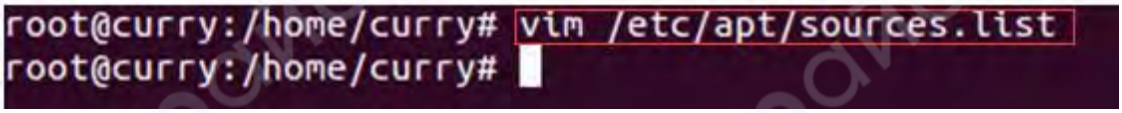
a. sudo apt-get install usbmount
b. sudo vim /etc/udev/rules.d/automount.rules
ACTION=="add",KERNEL=="sdb*", RUN+="/usr/bin/pmount --sync --umask 000 %k"
ACTION=="remove", KERNEL=="sdb*", RUN+="/usr/bin/pumount %k"
ACTION=="add",KERNEL=="sdc*", RUN+="/usr/bin/pmount --sync

--umask 000 %k"
ACTION=="remove", KERNEL=="sdc*", RUN+="/usr/bin/pumount %k"

c. udevadm control –reload-rules
It could refer to this:
http://unix.stackexchange.com/questions/134797/how-to-automatically-mount-an-usb-device-on-plugin-time-on-an-already-running-sy

3) System Source Configuration

Take Ubuntu as an example:
a. Open the source file
$ sudo vi /etc/apt/sources.list



b. Edit source file

Replace the source file with your favorite source. Take an example of Ubuntu 16.04 on Zhonkeda source:

deb http://mirrors.ustc.edu.cn/ubuntu-ports/ xenial main multiverse restricted universe

deb http://mirrors.ustc.edu.cn/ubuntu-ports/ xenial-backports main multiverse restricted universe

deb http://mirrors.ustc.edu.cn/ubuntu-ports/ xenial-proposed main multiverse restricted universe

deb http://mirrors.ustc.edu.cn/ubuntu-ports/ xenial-security main multiverse restricted universe

deb http://mirrors.ustc.edu.cn/ubuntu-ports/ xenial-updates main multiverse restricted universe

deb-src http://mirrors.ustc.edu.cn/ubuntu-ports/ xenial main multiverse restricted universe

deb-src http://mirrors.ustc.edu.cn/ubuntu-ports/ xenial-backports main multiverse restricted universe

deb-src http://mirrors.ustc.edu.cn/ubuntu-ports/ xenial-proposed main multiverse restricted universe

deb-src http://mirrors.ustc.edu.cn/ubuntu-ports/ xenial-security main multiverse restricted universe

deb-src http://mirrors.ustc.edu.cn/ubuntu-ports/ xenial-updates main multiverse restricted universe

Note: xenial is the version of the code name in this source, if the other version of Ubuntu needs to replace the corresponding version code which can be found on the internet.

4) Remote desktop installation

There are a lot of software, such as VNG, XRDP, X2GO, etc. For X2GO, it has more functions, and desktop color restore is very good which does not need too much configuration. And XRDP is much more safety than VNC.

a. $sudo apt-get install tightvncserver Install VNC

```
apt-get install tightvncserver
```

b. vncpassw Set the password: do not execute this command but executing vncserver directly. It will prompt you to enter the password twice, when prompted whether can be read only to select the *N*.

```
root@curry:/home/curry/tools/minidlna/minidlna-1.1.0# vncpasswd
Using password file /root/.vnc/passwd
VNC directory /root/.vnc does not exist, creating.
Password:
Verify:
```

c. Open one or more of desktops by vncserver or vncserver:1(vncserver:2)… you can also transfer more parameters through the full command as below:
vncserver :1 -geometry 1024x768 -depth 16 -pixelformat rgb565

(*Note:* If it prompted you that cannot find the file or other error when installing, please run sudo apt-get update to update the software source and try installing again.)

5) NAS and DLAN Configuration

a. NAS:
There are many files could be reference from Internet.

b. DLNA:

Mainly through the minidlna software to achieve the sharing of media resources within the LAN, such as sharing video, music, etc.. The installation steps are as follows:

i sudo apt-get minidlna

ii Execute the following command to modify the configuration file:
sudo nano /etc/minidlna.conf

Note: you can also use other text editor to modify.

iii Add the following:

media_dir=A,/nas, path: /DLNA/Music
media_dir=V,/nas, path: /DLNA/Video
media_dir=P,/nas, path: DLNA/Picture
db_dir=/nas, path: /DLNA/log
db_dir=/nas, path: /DLNA/db

ctrl +o and enter, ctrl +x to save and exit.

iv Established above folders respectively, noted that path consistency and assigned to read and write permissions. In order for convenient, it could be Chmod 755, such as sudo Chmod 755 /nas path /DLNA/Music

v Re-start minidlna to take effect the configuration: /etc/init.d/minidlna restart.

Transmit the corresponding file on the computer to the corresponding folder through samba.

Note: It is recommended to download MoliPlayer on the mobile device. The effect is good and no blue light pressure on both Android and IOS.

6) Thunder remote download

a. Go to the Thunder routing forum to download the required installation package first.

Download Xware1.0.31_cubieboard zip file.

Note: If you want to try the latest version, you can download the latest test version.

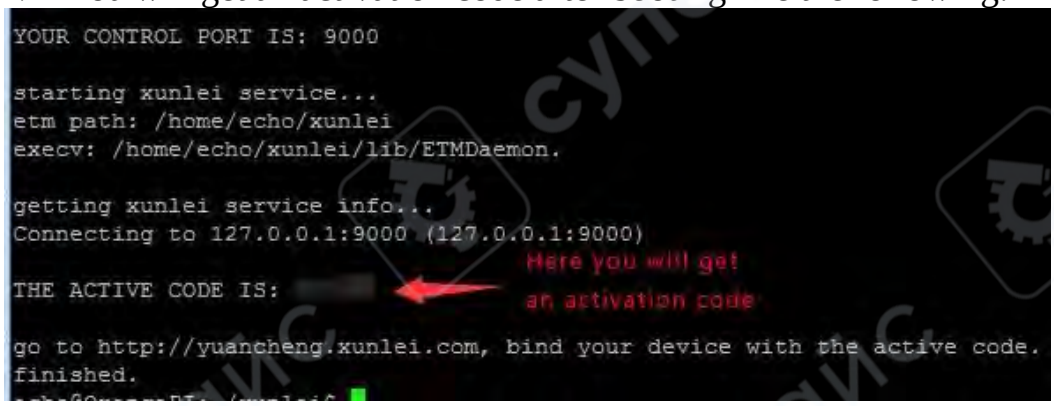
b. Enter the directory after uploaded the unzip file to OrangePi. It is recommended to rename the file to xunlei

c. Installation method of version 1.0.31:

i $ cd /xxx/xunlei The xxx is the directory of installation xunlei file
ii $ chmod 755 portal
iii $ ./portal

```
root@curry:/home/curry/Downloads/xunlei# ls
EmbedThunderManager  ETMDaemon  portal  vod_httpserver
root@curry:/home/curry/Downloads/xunlei# chmod 755 portal
root@curry:/home/curry/Downloads/xunlei#
```

iv You will get an activation code after booting like the following:



v Copy this activation code to *http://yuancheng.xunlei.com* (Which required to log in with account of Thunder). Then click the tab on the top right corner to add, fill in the activation code to complete the binding according to the following figure.

vi Setting start up

$ sudo nano /etc/rc.loacl

add the following contents before exit 0

cd /xx/xunlei

./portal &

ctrl +o and enter, ctrl +x to save and exit.

d. Installation of version 3.0.32.253:

i $ cd /xxx/xunlei The xxx is the directory of installation file of xunlei

ii $ sudo nano thunder_mounts.cfg Modify the download path

iii chmod +x etm_monitor

iv Run ./etm_monitor, there will be an activation code page like version 1.0.32. And then binding on the Thunder remote page (above steps 4, 5). There might be one or two errors while running, ignore it (selection type of shell and generation of INI file).

v Setting start up

sudo nano /etc/rc.loacl add the following contents before exit 0

cd /xx/xunlei

./etm_monitor &

ctrl +o and enter, ctrl +x to save and exit.

It could be remote downloading on computer, mobile phone or tablet by login yuancheng.xunlei.com

7) Modify the size of ext4 file system

After made the written image into SD card for booting, enter into rootfs partition's expansion of file system. It could enhance the performance of SD card to avoid limited storage cause problem.

- Method 1

Extend rootfs file partition of TF card on PC:

Select the specified disk, right click and select the corresponding disk, select "change size" and adjust it into your desired size, click "re-size", close the dialog box and click "apply all operations", select the application to complete the expansion operation

- Method 2

Enter into the system and extend via shell

Before partition

```
root@Orangepi:~# df -lh
Filesystem      Size  Used Avail Use% Mounted on
/dev/mmcblk0p2  2.0G  565M  1.4G  30% /
devtmpfs        482M     0  482M   0% /dev
tmpfs           490M     0  490M   0% /dev/shm
tmpfs           490M   13M  478M   3% /run
tmpfs           5.0M  4.0K  5.0M   1% /run/lock
tmpfs           490M     0  490M   0% /sys/fs/cgroup
/dev/mmcblk0p1   50M   13M   38M  26% /boot
```

Enter into system and expend via resize_rootfs.sh

```
root@Orangepi:/usr/local/sbin# resize_rootfs.sh
+ DEVICE=/dev/mmcblk0
+ PART=2
+ resize
+ fdisk -l /dev/mmcblk0
+ grep /dev/mmcblk0p2
+ awk {print $2}
+ start=143360
+ echo 143360
143360
+ set +e
+ fdisk /dev/mmcblk0

Welcome to fdisk (util-linux 2.27.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.
```

Enter resize_rootfs.sh on command line, the system will expending automatically,

Reboot the system and use df -lh to check whether expending is successful

```
+ set -e
+ partx -u /dev/mmcblk0
+ resize2fs /dev/mmcblk0p2
resize2fs 1.42.13 (17-May-2015)
Filesystem at /dev/mmcblk0p2 is mounted on /; on-line resizing required
old_desc_blocks = 1, new_desc_blocks = 1
The filesystem on /dev/mmcblk0p2 is now 3871616 (4k) blocks long.

+ echo Done!
Done!
root@Orangepi:/usr/local/sbin# df -lh
Filesystem      Size  Used Avail Use% Mounted on
/dev/mmcblk0p2   15G  566M   14G   4% /
devtmpfs        482M     0  482M   0% /dev
tmpfs           490M     0  490M   0% /dev/shm
tmpfs           490M   13M  478M   3% /run
tmpfs           5.0M  4.0K  5.0M   1% /run/lock
tmpfs           490M     0  490M   0% /sys/fs/cgroup
/dev/mmcblk0p1   50M   13M   38M  26% /boot
```

a. Expand file system

i Boot to Linux, umount /dev/sdb1 and /dev/sdb2, if it prompts disk busy, then use fuser to clean the using disk(we will recommend using another Linux booting disk to lead the system).

ii Use fdisk /dev/sdb to adjust the partition size, after into it, enter p, and keep in mind about the initial position of needed extending size partition.

iii Enter d to delete the partition need to change the size(my file system is /dev/sdb2, which is the 2 partition).

iv Enter n to build a new partition, make sure the initial position is the

same as you deleted, and enter the number as you desire.

v Enter w to save the partition data.

vi Use the following command to check the file system(make sure it is a right file system)

e2fsck -f /dev/sdb2

vii Adjust the partition size

resize2fs /dev/sdb2

viii It could mount a disk partition, you could check whether it has changed.

b. Shrink file system

i Boot to Linux, umount /dev/sdb1 and /dev/sdb2, if it prompts disk busy, then use fuser to clean the using disk(we will recommend using another Linux booting disk to lead the system).

ii Use the following command to check the file system(make sure it is a right file system)

e2fsck -f /dev/sdb2

iii Modify the size of file system(Use resize2fs)

resize2fs /dev/sdb2 900M

The "s"after the number represents specifying the size of file system via the sectors(every sector calculated by 512 bite). You could also specify it into K(KB), M(MB), G(GB), etc.

iv Use fdisk /dev/sdb to adjust the partition size, after into it, enter p, and keep in mind about the initial position of needed extending size partition. You need to first delete the partition then build a new one because the fdisk could not modify the size dynamic(you need to calculate the size, it have to enough to contain the file system adjusted in last step).

v Enter d to delete the partition need to change the size(my file system is /dev/sdb2, which is the 2 partition).

vi Enter n to build a new partition, make sure the initial position is the same as you deleted, and enter the number as you desire. Besides, if it is boot-able partition you want to change, note that need to keep the bootable mark in case cannot boot.

The above illustration is using fdisk and resize2fs to modify partition and file system, you could also use gparted. Gparted has graphical interface and it could help you to re-size file system at the same time of re-sizing partition. Goarted is much easier to use and reduce the change to make mistake. For now our official Lubuntu and Raspbian could not use it.

8) eth0 and wlan0 static mac address setting

a. If the system do not use systemd, you could modify rc.local directory and add the following:

```
$ vim /etc/rc.local
MAC=00:e0:4c:a1:2b:d4
ifconfig wlan0 down
ifconfig wlan0 hw ether $MAC
ifconfig wlan0 up
dhclient &
```

After rebooting, you could use ifconfig to check whether mac address has changed.

b. If the system used systemd, you also need to add the following besides the above steps:

```
$ cd /etc/systemd/system/
```

$ vim change_mac_address.service (You could name the server, format just like the following)

```
[unit]
Description=Change OrangePi Wifi mac address

[Service]
ExecStart=/etc/rc.local
RemainAfterExit=yes

[Install]
sWantedBy=multi-user.target
```

```
$ systemctl enable change_mac_address.service
```

Modify mac address of eth0 is same as modifying wlan0's, just need to replace wlan0 into eth0.

9) Orange Pi Android root

There is defaulted with root permission on Android pre-installed, but lacking authorization management software. The following is how to add authorization management software.

You need to have UsbModeSwitch.apk and UPDATE-SuperSU-v2.46.zip, install kingroot and make sure OTG on Orange Pi could connect to PC.

a. Open adb debug mode

Use U disk or card reader to install UsbModeSwitch.apk into Orange Pi OS and open it, tick "enable usb device mode" and use debug cable to

connect OTG port and PC (make sure it is micro usb-cable in case other cables could not be recognized). Normally PC would search and install adb driver software automatically. If PC failed to install, you could install PC version's Peasecod to install the driver software.

b. After connected Orange PI and PC, open command mode of PC, enter related command of adb(you need to install adb debug command, which Peasecod has adb command). Here is the command:

adb remount
adb shell

windows(win+r) command line enter into command mode, then enter into kingroot directory and execute the following steps:

adb shell
root@rabbit-p1:/ # mkdir /tmp
root@rabbit-p1:/ # cd /system/bin
root@rabbit-p1:/ # mount -o remount, rw /system
root@rabbit-p1:/system/bin # ln -s busybox-smp unzip
Logout adb shell Mode
root@rabbit-p1:/exit (Or Ctrl + C)
Unzip UPDATE-SuperSU-v2.46.zip
You will obtain META-INF/com/google/android/update-binary and put it into specific catalog.
adb push /path/UPDATE-SuperSU-v2.46.zip /data/local/tmp path is file's path
adb push /path/ update-binary /data/local/tmp
adb shell
root@rabbit-p1:/ #cd /data/local/tmp
root@rabbit-p1:/ #sh update-binary 0 1
/data/local/tmp/UPDATE-SuperSU-v2.46.zip

.....
.....

After executed scripts, enter reboot command and reboot it, you could use the device authorization management software normally.

After rebooted, there might be no super administrator icon, you need to delete the desk configuration file and reboot the board.

III. Linux Kernel Source Code Compilation

In order to support the rapid development of the project, we are writing this sections for project configuration options to the binary file. When the system is running, it can get the information of the system running by reading the binary file, which can greatly simplify the time of project development.

This manual describes how to use the binary file to speed up the development of the project.

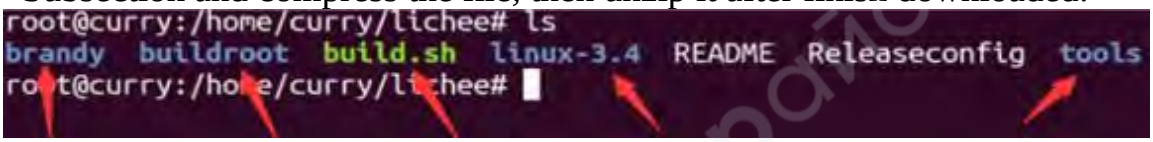
Hardware: Orange Pi development board*1, Card reader*1, TF card*1, power supply*1

Note: In the following sections, * indicates wild-cards, you need to fill in the actual values according to their file storage path.

1. Download Linux Source Code

You could download the source code from the official website:

Subsection and compress the file, then unzip it after finish downloaded:



buildroot: Project compilation script
brandy: gcc-linaro, boot and uboot source code and open source cross compiler tool
linux-3.10: Kernel source code
tools: Tools of project compilation
build.sh: compilation script

2. Compile Project Source Code

You need to compile the entire project while it is your first time to use the source code. You can use the following commands in the /lichee directory to complete the project:

- Enter into content of lichee, command

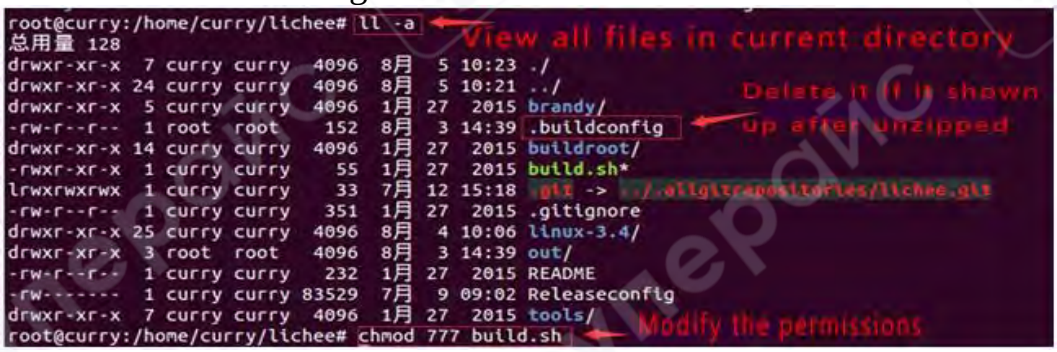
  $ ll -a

Check if there is an executable permission on build.sh, if not, modify the permissions

  $ chmod 755 build.sh

- If there is .buildconfig after commanded ll –a, delete it

  $ rm –rf .buildconfig



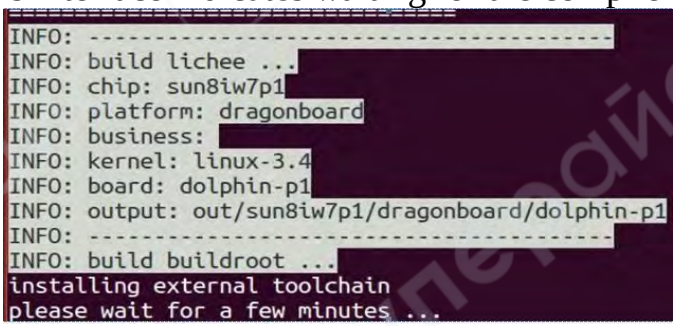
- Use the following command to compile the entire project

  $ ./build.sh config

At this point the system will prompt the choice of the chip, for OrangePi, select sun50iw2p1

At this point, the system will be prompted the choice of the board, for the OrangePi, select dragonboard，dolphin and dolphin-p2

Appear this interface indicates waiting for the compiler.



Wait fifteen minutes or so, compile complete.

3. Update the Kernel Image File and Replace Library

- After compilation is finished, the following files will be generated in the directory：

libs： lichee/out/sun50iw2p1/android/common/lib/modules/3.10.65

Download image from official website：

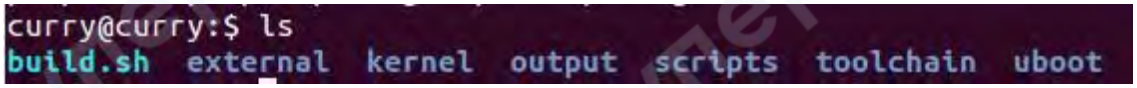
- Write the image：
  $ sudo dd bs=4M if=*.img of=/dev/sdb

- Pull out the card reader, and then insert it again.

Copy the kernel image file generated by the compiler to the first partition (boot partition)

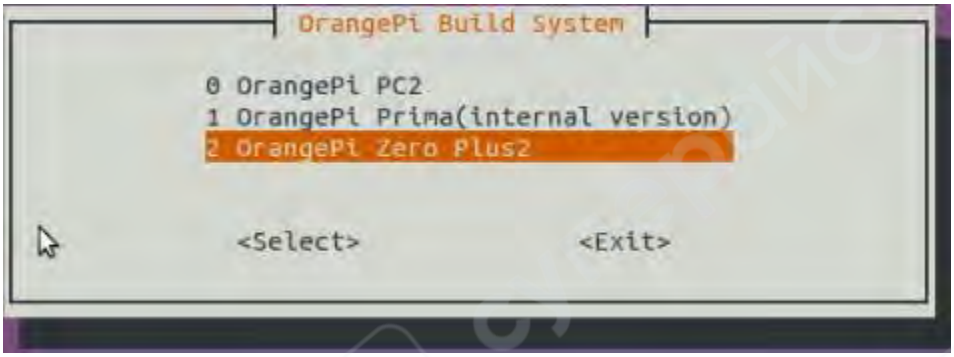
Copy the lib library which generated after compilation to the second partition (rootfs partition)

We would suggest using compilation system on github of official website.

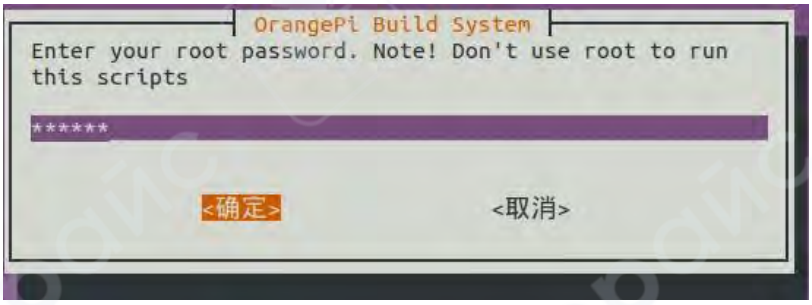


build.sh Execute script into the graphical interface of compilation
extenal Inside are patch and some configuration kernel file
output File generated
script Script compiled
toolchain Cross compiler location
uboot uboot source code

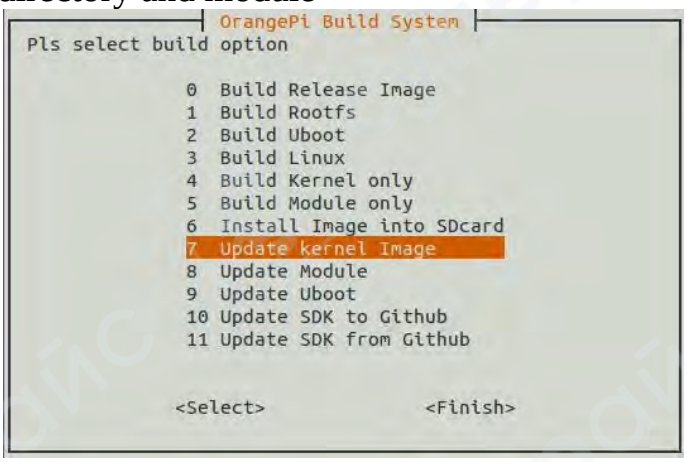
Execute./build.sh enter into graphical interface and select Zero plus2, the interface for Zero Plus is same as Zero plus2



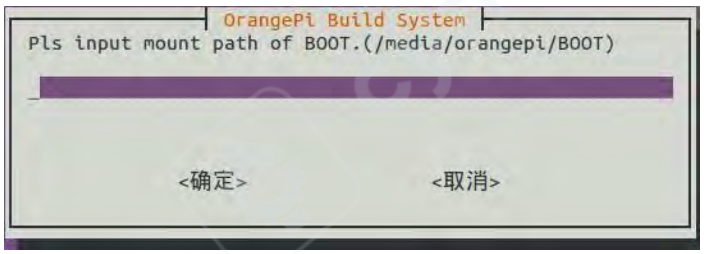
Enter password of root



Update Kernel directory and module



Select corresponding file directory and update uImage and modules



IV. Android Kernel Source Code Compilation

Hardware: Orange Pi development board*1, Card reader*1, TF card*1, power supply*1

Software

Linux host computer, which hard disk space at least 50G (to meet a fully compiled need)

Linux host computer needs:

Version 2.7.3 of Python;

Version 3.81-3.82 of GNU Make;

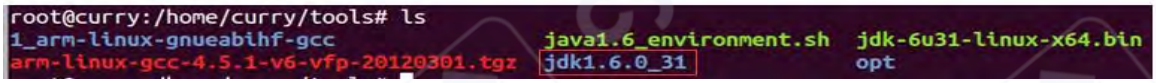
Version 1.7 or higher version of Git.

1. Install JDK

The following will illustrate jdk1.6 installation, it would be same for jdk1.7 installation.

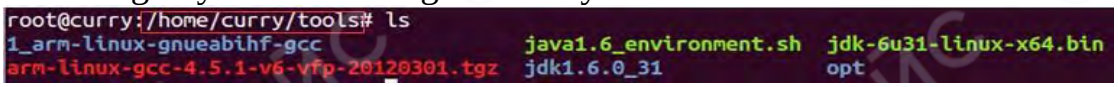
- Download and install JDK, you will obtain jdk-6u31-linux-x64.bin
- Modify the permission of jdk-6u31-linux-x64.bin, which has no prior permission
- $./jdk-6u31-linux-x64.bin

It will generate a folder:



- Input at terminal

Note that JAVA_HOME is the name of the current directory, you need to fill in according to your own storage directory.



$ export JAVA_HOME=*/jdk1.6.0_31
$ export PATH=$PATH:/$JAVA_HOME/bin
$ export CLASSPATH=.:$JAVA_HOME/lib
$ export JRE_HOME=$JAVA_HOME/jre

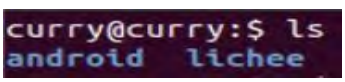
- Command line input Jav and press tab to see whether it can auto completion (Java), which indicates it can successfully installed version1.6.

2. Install Platform Supported Software

$ sudo apt-get install git gnupg flex bison gperf build-essential \
zip curl libc6-dev libncurses5-dev:i386 x11proto-core-dev \
libx11-dev:i386 libreadline6-dev:i386 libgl1-mesa-glx:i386 \
libgl1-mesa-dev g++-multilib mingw32 tofrodos \
python-markdown libxml2-utils xsltproc zlib1g-dev:i386
$ sudo ln -s /usr/lib/i386-linux-gnu/mesa/libGL.so.1
/usr/lib/i386-linux-gnu/libGL.so

3. Download Android Source Package

Download website.Then you will obtain the following directories:



4. Install Compiler Tool Chain

The compiler tool chain has been integrated in Android SDK. Tool chain is on: lichee/brandy/gcc-linaro/ of Android SDK(already exist)

```
brandy  buildroot  build.sh  linux-3.4  README  tools
root@curry:/home/curry/OrangePi/android/lichee# cd brandy/gcc-linaro/bin/
root@curry:/home/curry/OrangePi/android/lichee/brandy/gcc-linaro/bin# ls
arm-linux-gnueabi-addr2line     arm-linux-gnueabi-gprof
arm-linux-gnueabi-ar            arm-linux-gnueabi-ld
arm-linux-gnueabi-as            arm-linux-gnueabi-ld.bfd
arm-linux-gnueabi-c++           arm-linux-gnueabi-ldd
arm-linux-gnueabi-c++filt       arm-linux-gnueabi-ld.gold
arm-linux-gnueabi-cpp           arm-linux-gnueabi-nm
```

5. Compile Lichee Source Code

There are Android and Lichee after unzipped the package, enter the directory of Lichee:

$ cd lichee

$ ./build.sh lunch

Select sun50iw2p1

Print information of successful compilation

```
INFO: build kernel OK.
INFO: build rootfs ...
INFO: skip make rootfs for android
INFO: build rootfs OK.
----------------------------------------
build sun50iw2p1 android  lichee OK
----------------------------------------
```

6. Compile Command of Android Code

Input the command:

$ cd android

$ source ./build/envsetup.sh

```
root@curry:/home/curry/OrangePi/android/android# source ./build/envsetup.sh
including device/generic/armv7-a-neon/vendorsetup.sh
including device/generic/x86/vendorsetup.sh
including device/generic/mips/vendorsetup.sh
including device/asus/tilapia/vendorsetup.sh
including device/asus/grouper/vendorsetup.sh
including device/asus/deb/vendorsetup.sh
including device/asus/flo/vendorsetup.sh
```

$ lunch dolphin_fvd_p1-eng # Select the scheme number

```
curry@curry:$ lunch

You're building on Linux

Lunch menu... pick a combo:
     1. rabbit_cmccwasu_p1-eng
     2. rabbit_gms_p1-eng
     3. rabbit_fvd_p1-eng
     4. rabbit_aosp_p1-eng
     5. rabbit_aosp_p1-user
     6. rabbit_fvd_p1-user
     7. rabbit_fvd_p1-userdebug
     8. rabbit_aosp_perf-eng
     9. jaws_optimus-eng
     10. cheetah_fvd_p1-eng
     11. cheetah_fvd_p1-user
     12. cheetah_cts_p1-eng
     13. cheetah_cts_p1-user
     14. cheetah_cmcc_p1-eng
     15. cheetah_cmcc_p1-user
     16. molly-eng
     17. jaws_tvd_p1-eng
     18. rabbit_32bit_fvd_p1-eng
     19. cheetah_perf-eng
     20. eagle_fvd_p1_normal-eng
     21. eagle_fvd_p1_secure-eng
     22. dolphin_fvd_p1-eng
     23. dolphin_fvd_p1-user

Which would you like? [aosp_arm-eng] 10
```

$ extract-bsp # Copy the kernel and the drive module

```
curry@curry:$ extract-bsp
/expansion/xspace/OrangePi/Android/H5/zeroplus/android/device/*/cheetah-p1/bImage copied!
/expansion/xspace/OrangePi/Android/H5/zeroplus/android/device/*/cheetah-p1/modules copied!
curry@curry:$ make
============================================
PLATFORM_VERSION_CODENAME=REL
PLATFORM_VERSION=5.1
TARGET_PRODUCT=cheetah_fvd_p1
TARGET_BUILD_VARIANT=eng
TARGET_BUILD_TYPE=release
```

$ make The rear values of # is for the simultaneous compilation process, dependent on the host configuration

```
Creating filesystem with parameters:
    Size: 805306368
    Block size: 4096
    Blocks per group: 32768
    Inodes per group: 8192
    Inode size: 256
    Journal blocks: 3072
    Label:
    Blocks: 196608
    Block groups: 6
    Reserved block group size: 47
Created filesystem with 1393/49152 inodes and 79017/196608 blocks
+ '[' 0 -ne 0 ']'
Install system fs image: out/target/product/dolphin-fvd-p1/system.img
out/target/product/dolphin-fvd-p1/system.img+out/target/product/dolphin-fvd-p1/obj/PACKAGING
/recovery_patch_intermediates/recovery_from_boot.p maxsize=822163584 blocksize=4224 total=31
3479604 reserve=8308608
```

$ pack #Packaged into firmware

```
Dragon execute image.cfg SUCCESS !
----------image is at----------

/home/curry/OrangePi/android/lichee/tools/pack/sun8iw7p1_android_dolphin-p1_uart0.img

pack finish
```

$ cd */lichee/tools/pack/

```
root@curry:/home/curry# cd /home/curry/OrangePi/android/lichee/tools/pack/
root@curry:/home/curry/OrangePi/android/lichee/tools/pack# ls
chips  common  createkeys  out  pack  parser.sh  pctools  sun8iw7p1_android_dolphin-p1_uart0.img
root@curry:/home/curry/OrangePi/android/lichee/tools/pack#
```

V. Use Project Configuration Files

1. sys_config.fex Introduction

Configure hardware: sys_config.fe

The sys_config.fex is a binary configuration file that used by the SOC kernel driver or LiveSuit for a particular target board, including how to set up a variety of peripherals, ports, and I/O which based on the target version.

For OrangePi, the location of the project configuration document is：
lichee/tools/pack/chips/sun50iw2p1/configs/dolphin-p1/sys_config.fex

Copy the file to the directory of /lichee, use command:

```
$ cd ./lichee
$ cp ./tools/pack/chips/sun50iw2p1/configs/dolphin-p1/sys_config.fex ./
```

2. Examples

1) Modify the output mode into tv

- tv-out out, the output type of tv0 is invalid, you need to set the output type of tv1 into pal.

Modify defaulted enable display output configuration into tv

```
[tv0]
used                            = 1
tv_dac_used                     = 1
dac_src0                        = 0
dac_type0= 0
interface= 1
[tvout_para]
tvout_used= 1
tvout_channel_num= 1

[disp]
disp_init_enable= 1
disp_mode= 1
screen0_output_type= 2
screen0_output_mode= 11
screen1_output_type= 2
screen1_output_mode= 11
dev0_output_type = 4
dev0_output_mode = 4
```

dev0_screen_id = 0
dev0_do_hpd = 1
dev1_output_type = 2
dev1_output_mode = 11

Modify sys_confi and replace it when it generated scruot.bin. It would be faster if use compilation system on github. About compilation you could refer to the charter of Linux Compilation.

2) Loading tv.ko module automatically after booted

Enter /lib/ directory, enter command:
depmod -a
Add one more line on /etc/modules
tv
It would be tv out after booted

- Capacitance touch panel (capacitor tp)

Configuration Item	Configuration Meaning
ctp_used=xx	Whether turn on capacitance touch panel, if so set the value as 1, and vice verso 0.
ctp_name =xx	Indicates the control scheme used in the specified scheme, for now there are: "ft5x_ts" or "Goodix-TS".
ctp_twi_id=xx	Used for selecting i2c adapter, there are 0 and 2.
ctp_twi_addr =xx	Indicates the device address of i2c, it is related to the specific hardware.
ctp_screen_max_x＝xx	Maximum coordinates of the X axis of the touch panel
ctp_screen_max_y=xx	Maximum coordinates of the Y axis o the touch panel
ctp_revert_x_flag=xx	Whether needed to flip the X coordinates, if so then set 1, and vice verso 0.
ctp_revert_y_flag=xx	Whether needed to flip the Y coordinates, if so then set 1, and vice verso 0.
ctp_int_port=xx	GPIO configuration of the interrupt signal of capacitive touch panel
ctp_wakeup=xx	GPIO configuration of the wake-up signal of capacitive touch panel
ctp_io_port=xx	Capacitive screen IO signal, currently share with interrupt signal common pin

Configuration samples:

```
ctp_used        = 1
ctp_name        = "ft5x_ts"
ctp_twi_id      = 2
ctp_twi_addr    = 0x70
ctp_screen_max_x    = 800
ctp_screen_max_y    = 480
ctp_revert_x_flag   = 0
ctp_revert_y_flag   = 0
ctp_int_port        = port:PH21<6><default>
ctp_wakeup          = port:PB13<1><default><default><1>
ctp_io_port         = port:PH21<0><default>
```

Note: If you want to support the new capacitive touch IC, you need to combine the configuration of the BSP A10 layer, which should be based on the original capacitive touch IC code, to make the appropriate changes. Specifically, 1) ctp_twi_id should be consistent with the hardware connection in sys_config; 2) In the drive part of the code: the use of twi from the device name + address should be consistent with the ctp_name and ctp_twi_addr in sys_config configuration. At the same time, the other sub configuration in sysconfig should also be properly configured, these configurations should be corresponding processing in the program.

VI. OrangePi Driver development

In order to help developers become more familiar with OrangePi, this manual describes how to use simple device driver modules and applications on the development board.

Hardware: Orange Pi development board*1, Card reader*1, TF card*1, power supply*1

1. Device Driver and Application Programming

1) Application Program (app.c)

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <string.h>

int main(int argc, char *argv[])
{
	int cnt, fd;
	char buf[32] = {0};
	if(argc != 2)
	{
		printf("Usage : %s </dev/xxx>\r\n", argv[0]);
		return -1;
	}

	fd = open(argv[1], O_RDWR);
	if(fd < 0)
	{
		printf("APP Error : open device is Failed!\r\n");
		return -1;
	}
	read(fd, buf, sizeof(buf));
	printf("buf = %s\r\n", buf);
	close(fd);
	return 0;
}
```

2) Driver Program (OrangePi_misc.c)

```
#include <linux/kernel.h>
#include <linux/module.h>
#include <linux/fs.h>
#include <linux/miscdevice.h>
#include <linux/init.h>
#include <asm-generic/uaccess.h>

static int orangepi_open(struct inode *inodp, struct file *filp)
{
	return 0;
}

static ssize_t orangepi_read(struct file *filp, char __user *buf, size_t
count, loff_t *offset)
{
	char str[] = "Hello World";
	copy_to_user(buf, str, count);
	return 0;
}

static struct file_operations tOrangePiFops = {
	.owner = THIS_MODULE,
	.open  = orangepi_open,
	.read  = orangepi_read,
};

static struct miscdevice OrangePi_Misc = {
	.minor = 255,
	.name  = "orangepimisc",
	.fops  = &tOrangePiFops,
};
```

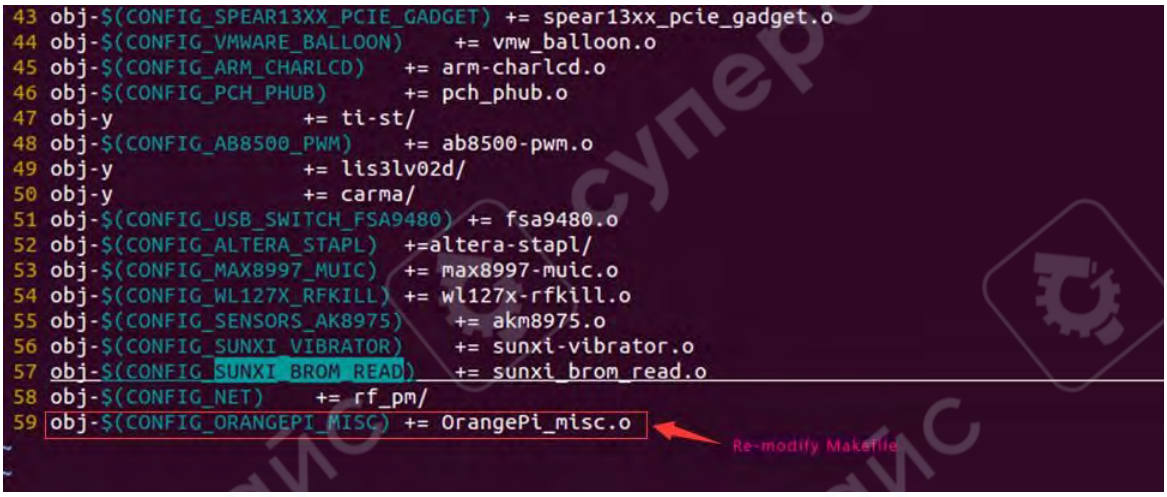
```
static int __init OrangePi_misc_init(void)
{
	int ret;
	printk("func : %s, line : %d\r\n", __func__, __LINE__);

	ret = misc_register(&OrangePi_Misc);
	if(ret < 0){
		printk("Driver Error : misc_register is Failed!\r\n");
		return -1;
	}
	return 0;
}

static void __exit OrangePi_misc_exit(void)
{
	int ret;
	printk("func : %s, line : %d\r\n", __func__, __LINE__);
	ret = misc_deregister(&OrangePi_Misc);
	if(ret < 0){

		printk("Driver Error : misc_register is Failed\r\n");
	}
}

module_init(OrangePi_misc_init);
module_exit(OrangePi_misc_exit);
```

2. Compile device driver

Copy the OrangePi_misc.c to the */lichee/linux-3.10/driver/misc directory:

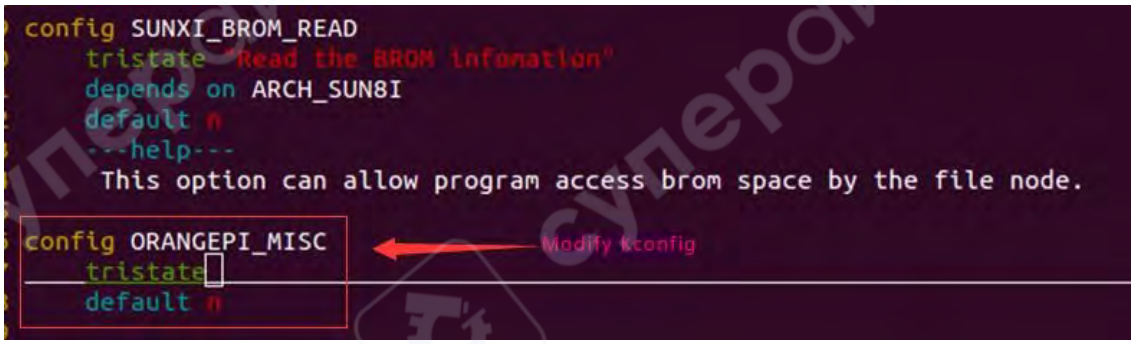
Enter to */lichee/linux-3.10/drivers/misc/, and modify makefile

Modify Makefile on currently file, shown as following:



There is Kconfig on the same sibling folders with Makefile. Each Kconfig respectively describes the the source directory file related kernel configuration menu. In the kernel configuration making menuconfig, it read from the Kconfig config menu and the user configuration saved to the config. In the kernel compile, the main Makefile by calling this.Config could know the user's configuration of the kernel.

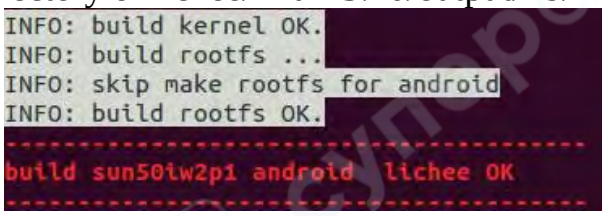
Kconfig is corresponding to the kernel configuration menu. Add a new driver to the kernel source code, you can modify the Kconfig to increase the configuration menu for your drive, so you can choose whether the menuconfig driver was compiled or not.



Back to the source code directory:

$ ./build.sh

After compiled the kernel, there will be an orangepi_misc.ko file generated on the directory of lichee/linux-3.10/output/lib/modules/3.10.65



There is a .ko module which generated after compiled of OrangePi_misc.c on */lichee/linux-3.10/output/lib/modules/3.10.65/

Insert U disk (please note the SD card should have been written image) if the SD card system is mounted to the directory / dev / SDB, SD card will have two sub mount points, respectively are / dev / sdb1 and /dev/sdb2. Two partition of SD card will automatically mount to the PC /media/ directory, the first partition is the boot partition and the second partition is the rootfs partition.

Copy the OrangePi_misc.ko file to /media/*/lib/modules/3.10.65.

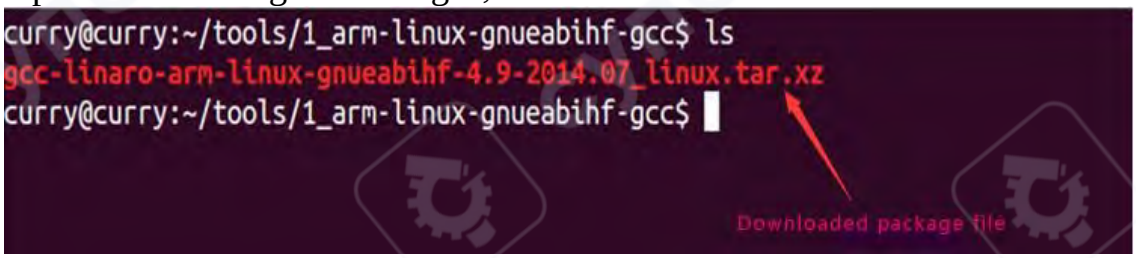
$ cp OrangePi_misc.ko /media/*/lib/modules/3.10.65

3. Cross compiler Application Program

Here will take arm-linux-gnueabihf-gcc as an example. Check whether there is the cross compiler, if not, then download and install it.

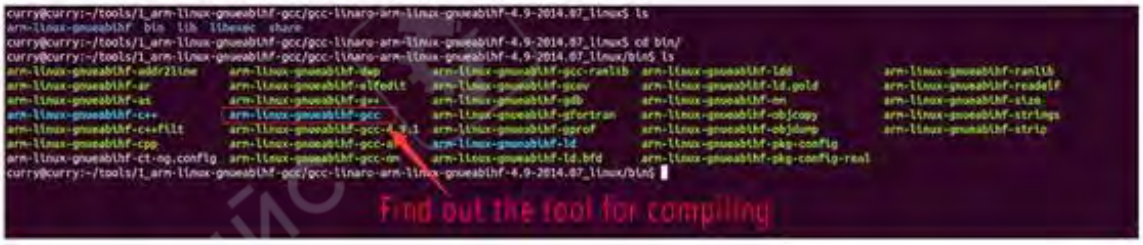
$ arm-linux-gnueabihf-gcc -v

While compiling the application, you will fill that you need the cross compiler arm-linux-gnueabihf-gcc, download and install it.

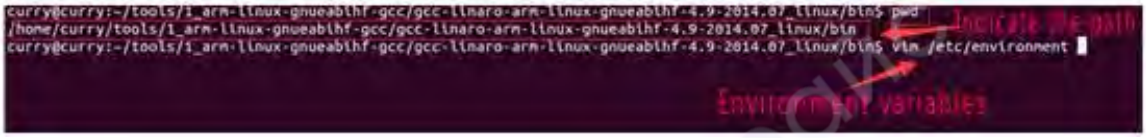


Unzip the downloaded file and enter the the directory

Check the information after entering bin directory

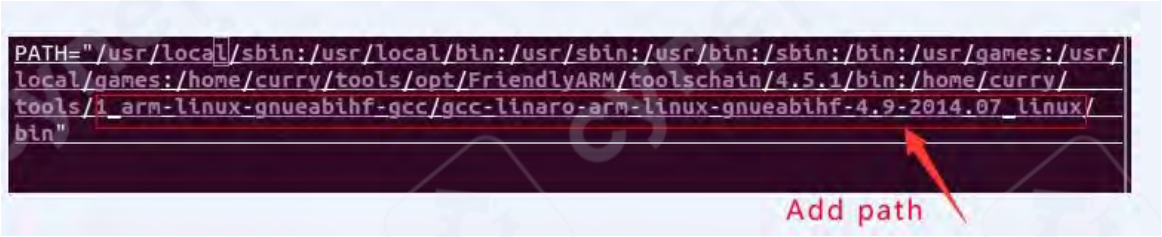


pwd hows the path and export it into the whole project



$ ll /etc/environment shows that the file can only read, need to modify permissions
$ chmod 755 /etc/environment
Modify permission

Add the path to the whole environment variable



Compile the application with cross compiler

$ arm-linux-gnueabihf-gcc app.c –o aq
There will be an ap application generated in the directory, copy it to the development board file system(on the rootfs directory of /home/orangepi/)
$ cp aq /media/*/home/orangepi/

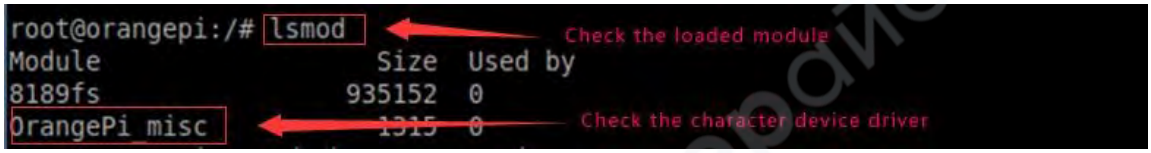
4. Running Driver and Application

Removed the SD card and inserted it into the development board and power on.

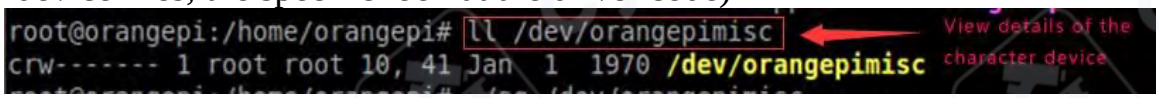
You need to switch to root users and load module driver module to the development board first.

$ insmod /lib/modules/orangepi.ko

$ lsmod To check whether it is loaded



$ ll /dev/orangepimisc(Miscellaneous equipment automatically generated device files, the specific look at the driver code)



Executive application (note the use of the application, the specific check at the code)

$ ./aq /dev/orangepimisc

VII. Using Debug tools on OrangePi

Hardware: Orange Pi development board*1, Card reader*1, TF card*1, power supply*1

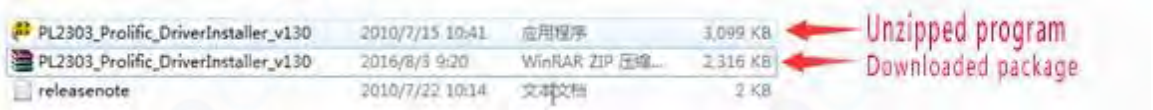
TTL to USB cable

1. Operation Steps on Windows

In order to get more debugging information in the project development process of using OrangePi, OrangePi default support for serial information debugging. For developers, you can simply get the serial port debugging information with the materials mentioned above. The host computer using different serial debugging tools are similar, basically can reference with the following manual for deployment. There are a lot of debugging tools for Windows platform, the most commonly used tool is putty. This section takes putty as an example to explain the deployment.

1) Install USB driver on Windows

- Download and unzip the latest version of driver PL2303_Prolific_DriverInstaller_v130.zip

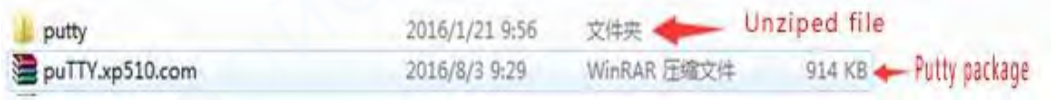


- Choose application installation as Administrator

- Wait for completing installation

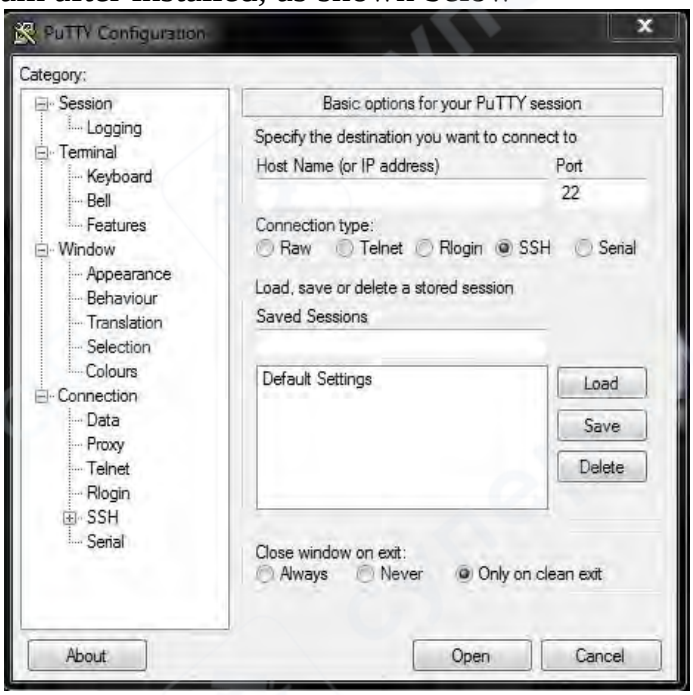
2) Install putty on Windows

- Download putty installation package



- Unzip and install

- Open program after installed, as shown below

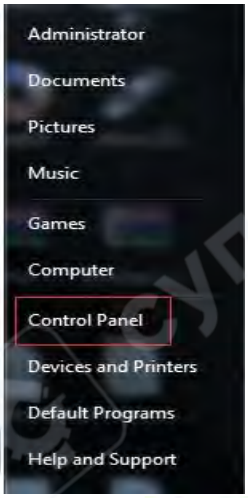


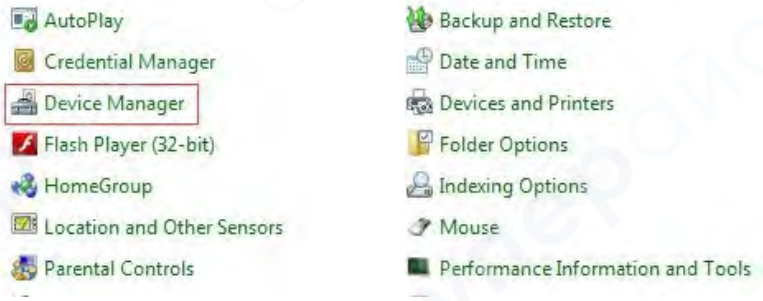
3) Connecting method

Use the TTL to the serial port cable, one end connected to OrangePi, the other end connected to PC

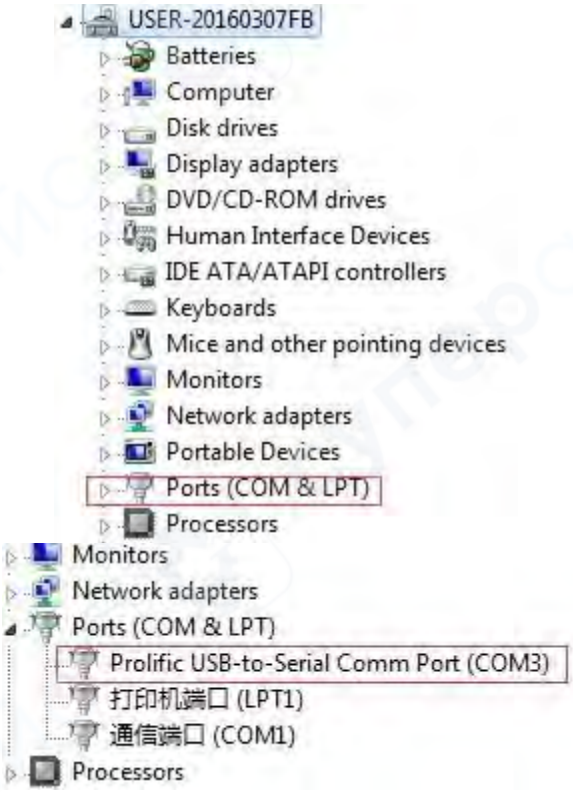
4) Equipment information acquisition

- *Start* menu select *control panel*

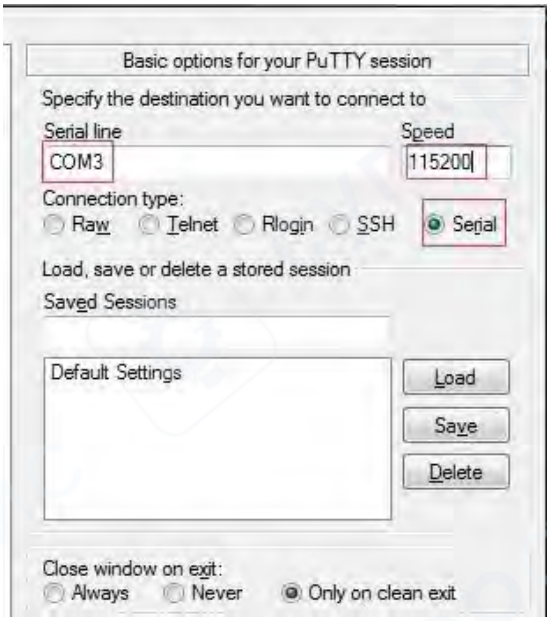




- Click on the *device manager* to check the *port numbe*r



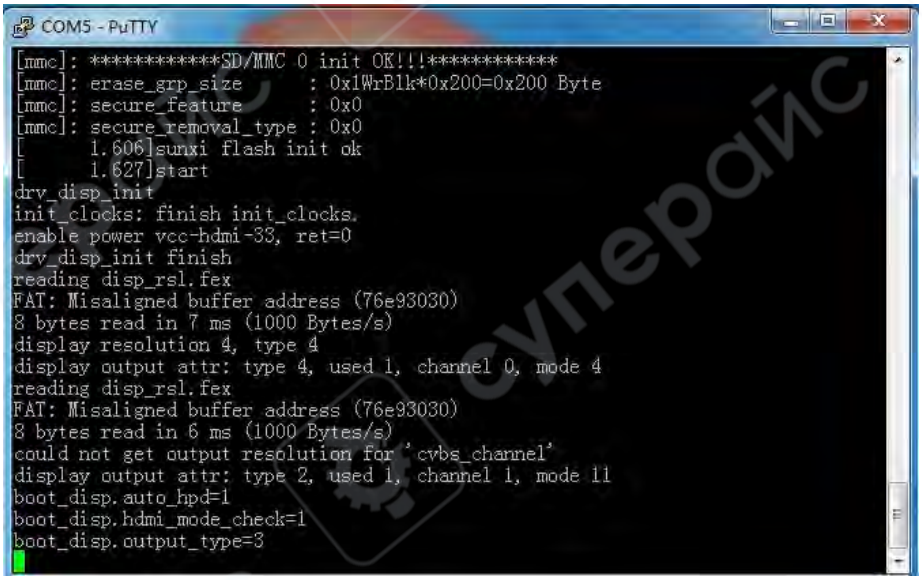
5) Putty Configuration



Serial port should set to the corresponding port number (COM5), the speed should set to 115200

6) Serial Debug Port

Power on and boot OrangePi, the serial port will automatic print debug log

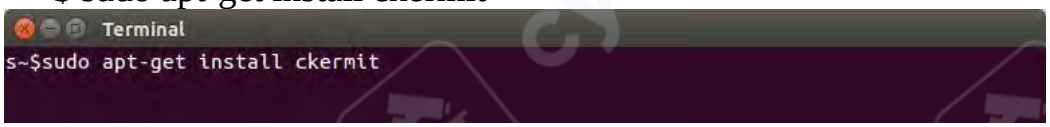


2. Operation Steps on Linux

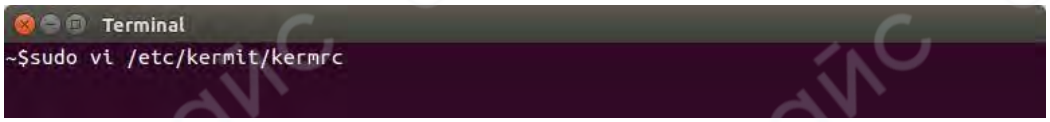
There are Minicom and Kermit serial debugging tools for Linux, this section will take Kermit as an example to have an illustrate.

1) Install Kermit

- Install the Kermit by execute command:
  $ sudo apt-get install ckermit

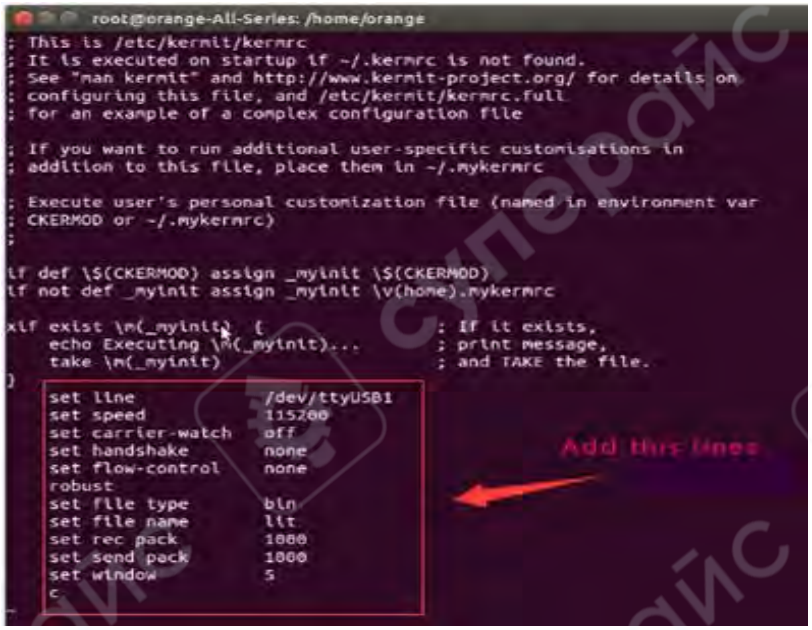


- Configurate Kermit
  $ sudo vi /etc/kermit/kermrc



- Add lines：

```
set line          /dev/ttyUSB1
set speed         115200
set carrier-watch    off
set handshake  none
set flow-control     none
robust
set file type     bin
set file name        lit
set rec pack         1000
set send pack       1000
set window           5
```



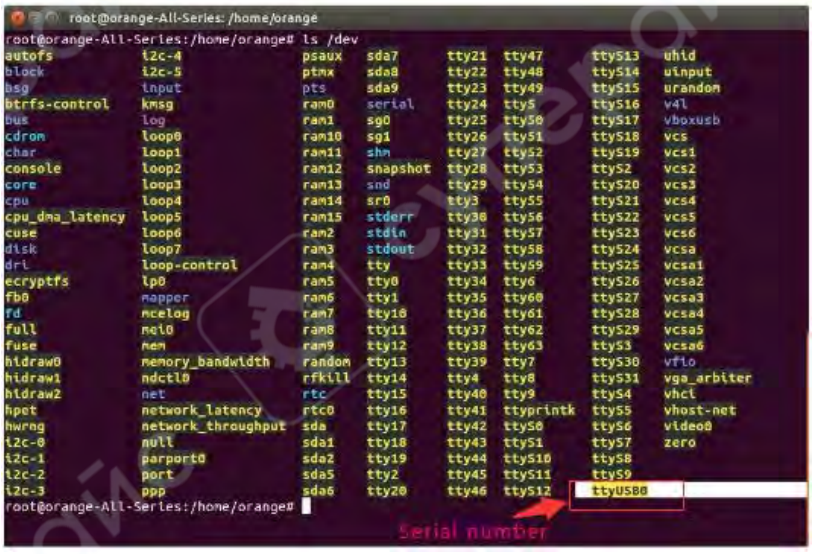
2) Connecting method

Use the TTL to the serial port cable, one end connected to OrangePi, the other end connected to PC

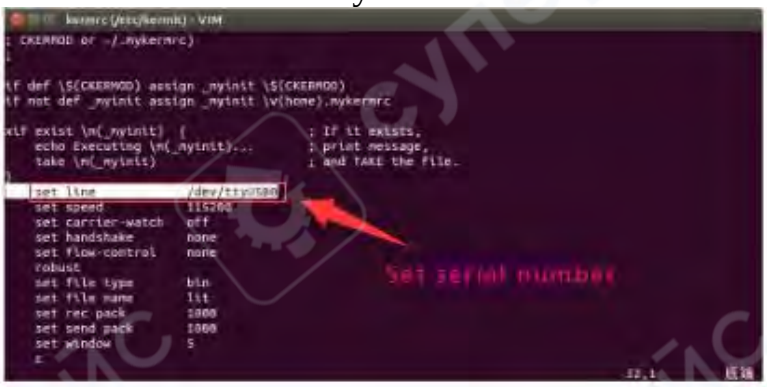
3) Equipment information acquisitio

Input command in the PC terminal to check the device number of TTL to the serial cable

$ ls /dev/

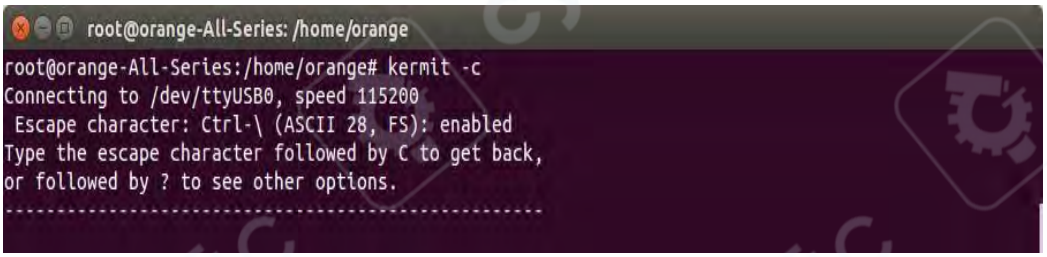


- It can be seen from the figure that TTL to the serial port cable is identified as ttyUSB0, configure the /ect/kermit/kermitc file, update the serial port information.
  $ sudo vi /etc/kermit/kermitc
- Set the value of setline into /dev/ttyUSB0



4) Start debug

- Input command in the host computer terminal, enter the Kermit mode:
  $ sudo kermit –c



- Power on and boot OrangePi, the serial port will automatic print debug log