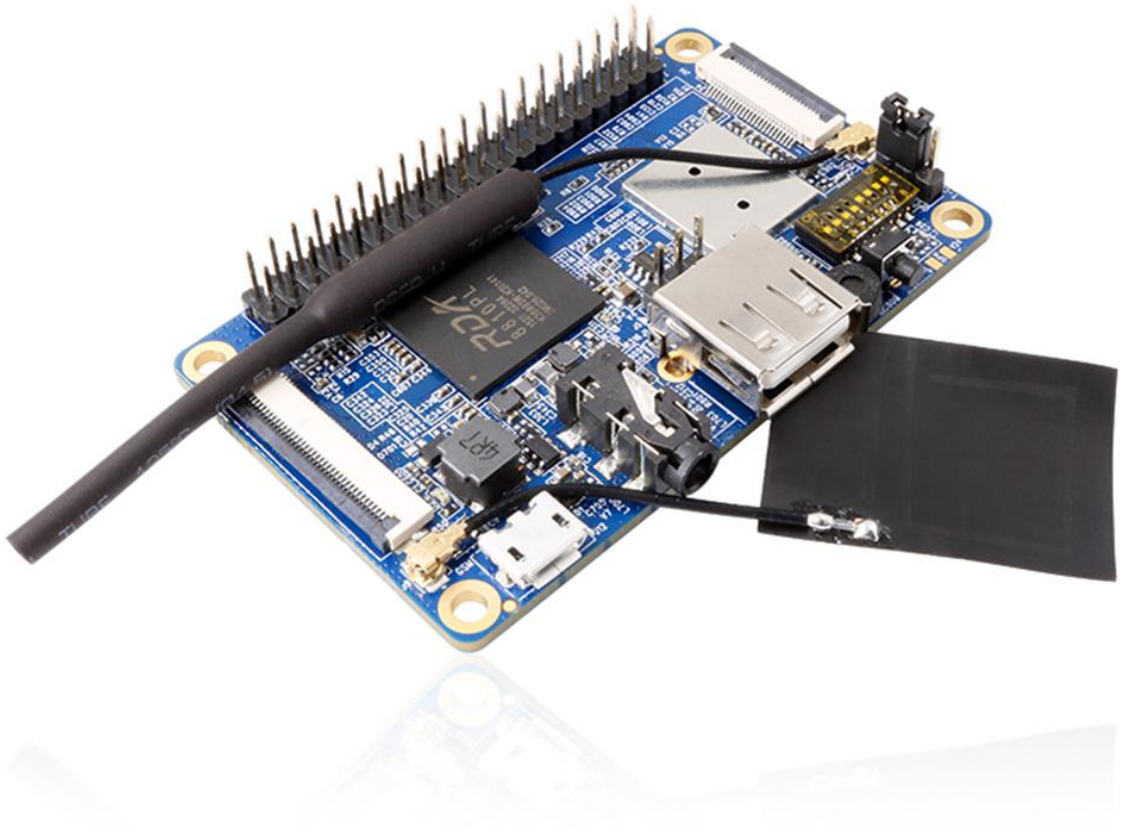
Orange Pi 2G-IOT

User Manual



Contents

I. Orange Pi Introduction....3
1. What is Orange Pi 2G-IOT？....3
2. What can I do with Orange Pi 2G-IOT?....3
3. Whom is it for?....3
4. Hardware specification....3
5. GPIO Specifications....6
II. Using Method....7
1. Step 1: Prepare Accessories Needed....8
2. Step 2: Prepare a TF Card....8
1) Introduction of Writing image into a SD card....8
2)Wifi connecting and sound output setting....11
3. Step 3: Start your Orange Pi....11
4. Step 4: Turn off your Orange Pi correctly....18
5. Enter the system via vnc and ssh....16
6. Universal software configuration....15
1) Change default account....18
2) System source configurationi....16
3) Remote desktop installation....16
4) Modify the size of ext4 file system....17
III. Source Code Compilation of Android and Linux....19
1. Install JDK....19
2. Install Platform Supported Software....20
3. Download the Source Package and Unzip it....20
4. Android source code compiler....20
5. Compile Linux source Code....21
IV. Orange Pi Driver development....23
1. Device driver and application programming....23
2. Compile device driver....25
3. Compiling method of application....26
4. Running driver and application....27
V. Using Debug tools on OrangePi....29
Operations on Windowst....29
1. Install USB driver on Windows....34
2. Install putty on WindowS....35
3. Connect style....361
4.Equipment information acquisition....36
5. Putty configuration....33
6. Start debug....34
Operations on Linux....35
1. Install Kermit....35
2. Connect method for debug....36
3. Equipment information acquisition....36
4. Start debug....36

I. Orange Pi Introduction

1. What is Orange Pi 2G-IOT?

It's an open-source single-board computer. It can run Android 4.4, Ubuntu, Debian, Raspberry Pi image. It uses the RDA8810 Soc, and has 256MB LPDDR2 SDRAM.

2. What can I do with Orange Pi 2G-IOT?

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 2G-IOT is open source.

3. Whom is it for?

Orange Pi 2G-IOT 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

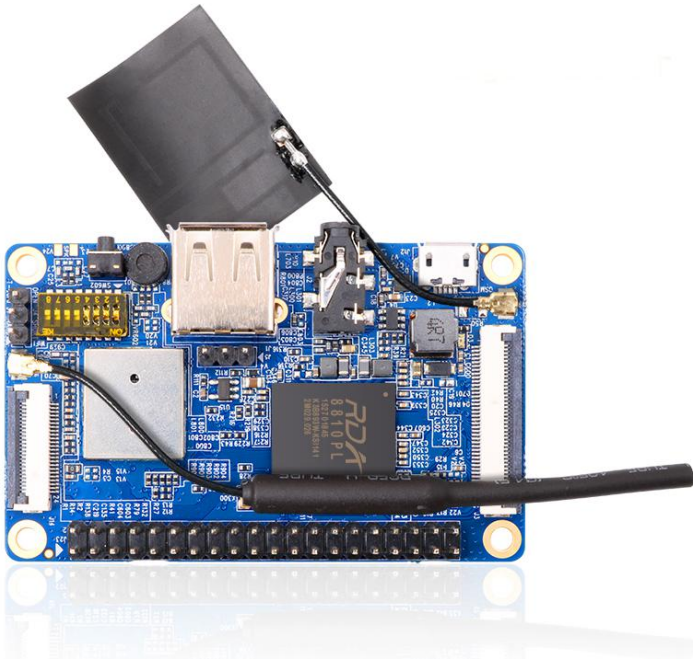
Hardware specification	
CPU	ARM Cortex-A5 32bit
GPU	Separate graphic processor, Vivante's GC860 support OpenGLES1.1/2.0 support OpenVG1.4 support DirectFB support GDI/DirecShow 30M Triangle/s, 250M Pixel/s
Memory (SDRAM)	Integrated 256MB LPDDR2 SDRAM

Onboard Storage	TF card / Integrated 500MB 8Bit 1.8V 4K SLC Nand Flash
Onboard WIFI+BT	RDA5991, WIFI+BT
2G model	The four frequency single card GSM/GPRS Dedicated accelerators SIM card
Video Input	A CSI input connector Camera： Supports 8-bit YUV422 CMOS sensor interface Supports CCIR656 protocol for NTSC and PAL Supports SM pixel camera sensor Supports video capture solution up to 1080p@30fps
Audio Input	MIC, 3.5 mm Jack
Video Outputs	LCD
Audio Output	3.5 mm Jack、 FM、SPEAK（Optional）
Power Source	USB OTG input can supply power Battery input can supply power（Optional）
USB 2.0 Ports	One USB 2.0 HOST, One USB 2.0 OTG
Buttons	Power Button(SW602)
Low-level peripherals	40 Pins Header, compatible with Raspberry Pi B+
GPIO(1x3) pin	UART, ground.
LED	Power led
Supported OS	Android, Ubuntu, Debian, Rasbian

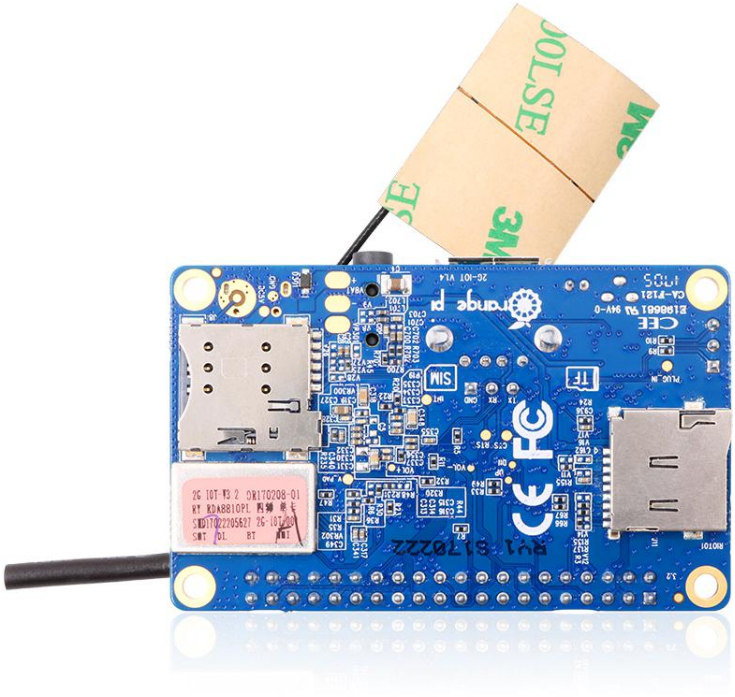
Interface definition

Product size	67mm × 42mm
Weight	35g

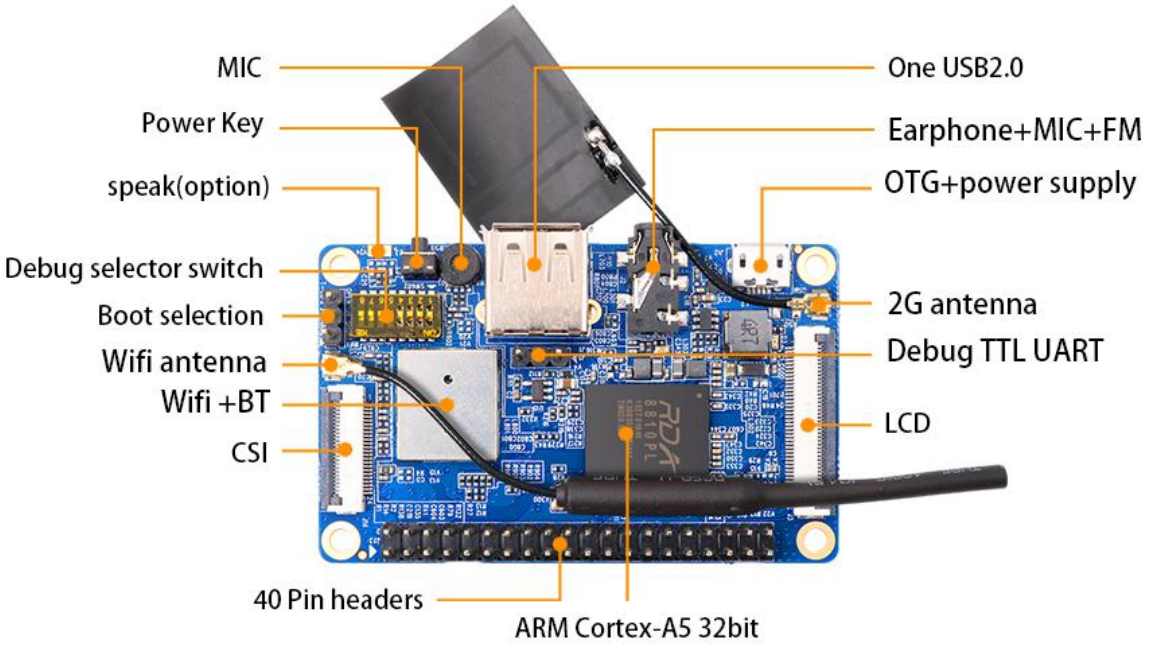
Top view

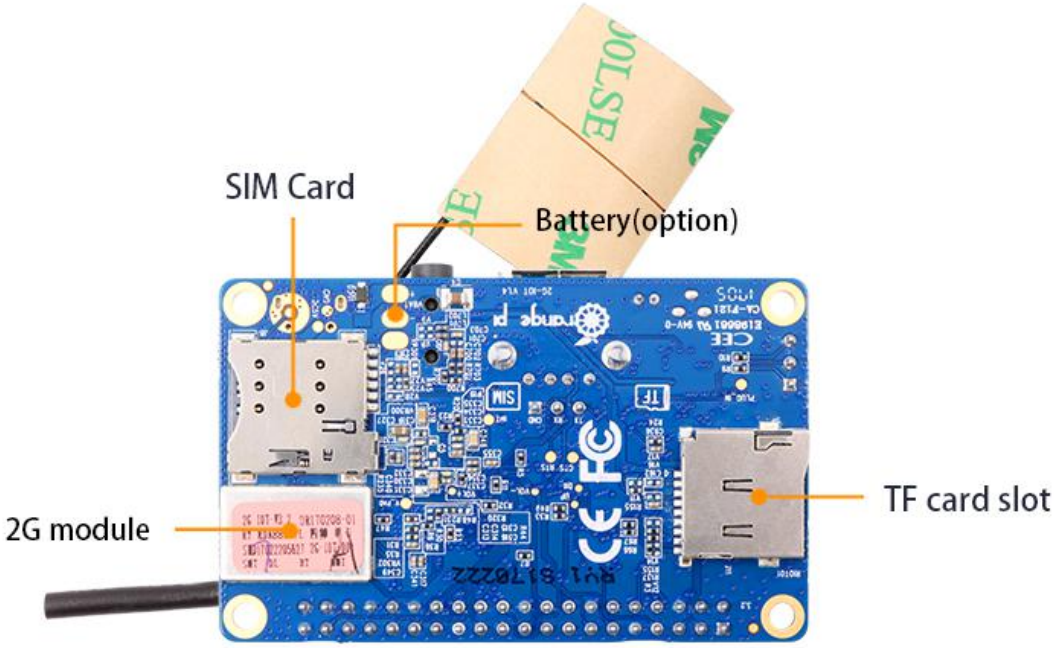


Bottom view



Interface instructions:

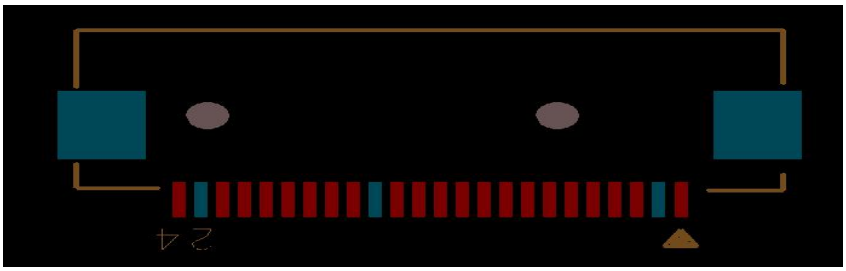




5. GPIO Specifications

The CSI Camera Connector is a 24-pin FPC connector which can connect external camera module with proper signal pin mappings. The pin of CIS connector can be defined as follows. The connector marked with "CON 1" on the Orange Pi 2G-IOT is

camera connector.



OrangePi 2G-IOT-CSI

CON1-P01	NC	
CON1-P02	GND	
CON1-P03	TWI2-SDA	PE13
CON1-P04	VCC-CSI	
CON1-P05	TWI2-SCK	PE12
CON1-P06	CSI-RESET#	PE15
CON1-P07	CSI-VSYNC	PE3
CON1-P08	CSI-STBY-EN	PE15
CON1-P09	CSI-HSYNC	PE2
CON1-P10	VDD1V8-CSI	
CON1-P11	VCC-CSI	
CON1-P12	CSI-D7	PE11
CON1-P13	CSI-MCLK	PE1
CON1-P14	CSI-D6	PE10
CON1-P15	GND	
CON1-P16	CSI-D5	PE9
CON1-P17	CSI-PCLK	PE0
CON1-P18	CSI-D4	PE8
CON1-P19	CSI-D0	PE4
CON1-P20	CSI-D3	PE7
CON1-P21	CSI-D1	PE5
CON1-P22	CSI-D2	PE6
CON1-P23	GND	
CON1-P24	AFVCC-CSI	

II. Using Method

You can configure your Orange Pi in a very short period of time and use it according

to the following steps. You need to fulfill the several steps before booting your Orange Pi.

1. Step 1: Prepare Accessories Needed

The first time you use the Orange Pi, you need at least some parts for the following:

No.	Items	Requirements and Instructions
1	TF card	8GB min.; class 10 (the class indicates how fast the card is). Branded TF cards which are much more reliable are the good choice
2	Power adapter	At lease 5V/2A high quality power adapter, OTG could use as power supply.
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	TTL to USB cable	Support debug log in.
5	Audio cable (Optional)	You can select an audio cable with 3.5mm jack to feel stereo audio.
6	SIM Card (Optional)	Support 2G SIM card

TF card

DC power adapter

2. Step 2: Prepare a TF Card

In order to be able to us Orange Pi normally, you must install the operating system in the TF card first. The following instructions will teach you how to write the operating system image file to the Windows and Linux environments.

1) Introduction of Writing image into a SD card

Windows:

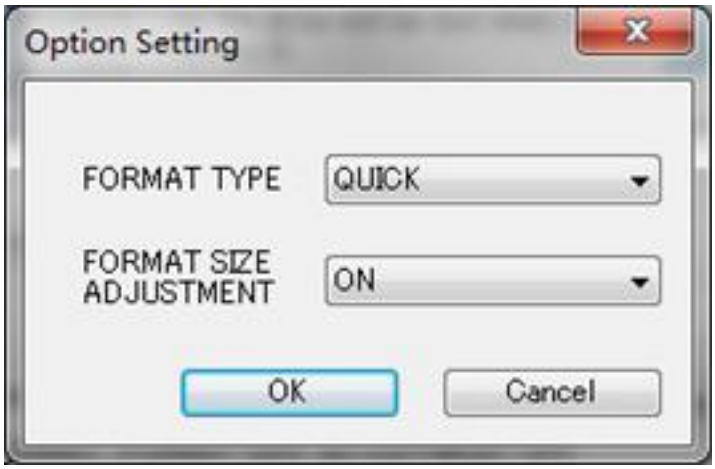
a. Inserting the TF card into the computer, the capacity of the card must be larger than the operating system image, usually requires 8GB or bigger capacity.

b. Formatting the TF card.

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

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

iii. In the *options settings* option set the format type option to quick formatting. *Logical size adjustment* option to open "(ON)"



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

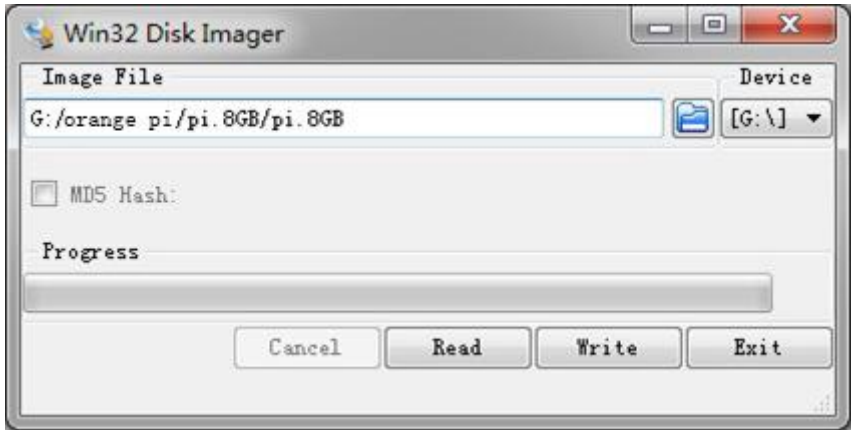
v. Click the "Format"button.

c. Download the operating system image file from the download page, the page address is as follows: http://www.orangepi.cn/downloadresourcescn/

d. model to 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 the downloaded file, select "Unzip file" to write image to TF card.

i. Download tools to write image, such as **Win32 Diskimager**, http://sourceforge.net/projects/win32diskimager/files/Archive/.
ii. Select the path of image file that has been unzipped.



iii. Click the "Write" button and wait for the image writing.
iv. After the image is written, click the "Exit" button.

Linux:

a. Inserting the TF card into the computer, the capacity of the card must be larger than the operating system image, usually requires 8GB or bigger capacity.
b. Formatting the TF card.
i. Run ***fdisk –l*** command to make sure TF disk.
ii. Run ***umount /dev/sdxx*** to uninstall all partitions of TF Card.
iii. Run ***sudo fdisk /dev/sdx*** command. Use director to delete all partitions of TF Card, and then us ***n*** command 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 image OS from download page:
http://www.orangepi.cn/`downloadresourcescn/

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

e. Write image into TF card

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

ii. Make sure the image file ***hash key*** is the same as download page offered(optional) :

sha1sum [path]/[imagename]

Here will be output some number which should be same as the image page line of *"SHA-1"*

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

iv. Run the command of *sudo dd bs=4M if=[path]/[imagename] of=/dev/sdx* to write image file and wait for it finished. You can run *sudo pkill –USR1 –n –x dd* command to check the procedure.

2) Wifi connecting and sound output setting

Linux OS:

a. Connecting the wireless network: you could operate the following command to connect in Ubuntu OS:

```
sudo Connect_New_Wifi.sh
```

b. 3.5mm sound output setting: you could operate the following command to connect in Ubuntu OS:

```
sudo Audio_Configure.sh
```

Android OS:

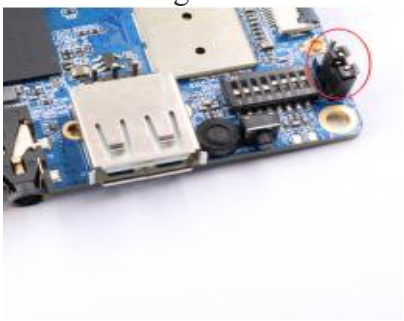
a. Connecting the wireless network：Connecting Wifi in setting directory of Android OS.

b. 3.5mm sound output setting：Setting sound is in Setting directory of Android OS.

3. Step 3: Start your Orange Pi

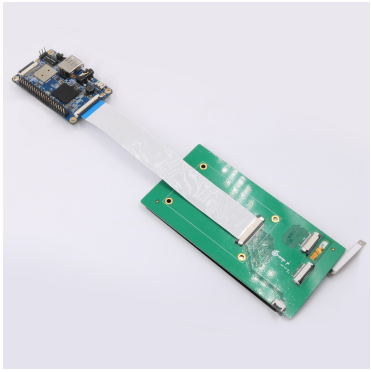
- Insert the TF card with written image into the TF card slot

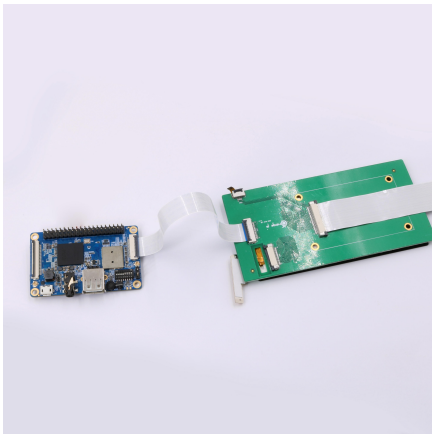
- Make sure the toggle switch is showing like the following, booting from SD card.



- Insert the keyboard or mouse into the USB port.
- Connect wifi antenna and base-band antenna

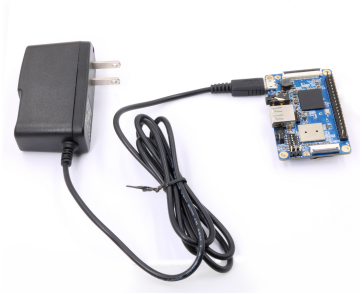
- Connecting LCD and Camera





- Connecting TTL cable, you could refer to the Debug method in this instruction.

- It is the power input interface on the right side for connecting a 5V and at least 2A or bigger than 2A power adapter. Avoid using smaller power GSM mobile phone charger, it is not able to output 2A even if it marked "5V/2A".



If the above steps are successful, the OrangePi will start in a few minutes. The monitor Graphical interface of display system. 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 command 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. If all the above steps run, then your Orange Pi could shut down.

5. Enter the system via vnc and ssh

You could enter the system via vnc or ssh remote enter as long as your system has already connected wifi network. The default port number is 1, pass word is orangepi, ip address is your actual address.

6. Universal software configuration

1) Change default account

The default log-in account and password is orangepi/orangepi or root/orangepi. It is recommended to modify the default orangepi account to your own account for secure sake. Take changing into Zhangsan as a sample. Steps are as follows:

a. Use root account to login Orange Pi

b. $ usermod -l zhangsan orangepi

Change account of orangepi 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 into orangepi user's home directory

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

f. $ cat /etc/passwd

It should be shown as following:

```
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 steps, you could use the new account Zhangsan to log in.

2) System source configuration

This instruction will take Ubuntu as an example:

a. Open the source file

$ sudo vi /etc/apt/sources.list

```
root@curry:/home/curry# vim /etc/apt/sources.list
root@curry:/home/curry#
```

b. Edit source file

Replace the source file with your favourite 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.

3) Remote desktop installation

There is a lot of software, such as VNG, XRDP、X2GO, etc. As for X2GO, there are more functions and the desktop color restore is very good which does not need too many configuration. And xrdp is much more safety than vnc.

a. $sudo apt-get install tightvncserver

Install VNC

```
apt-get install tightvncserver
```

b. vncpassw

Do not execute this command for setting password, only need to executing vncserver which would indicate you enter the password. It will prompt you to enter the password twice, when prompts whether can be read only, select 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 the desktops via vncserver or vncserver:1(vncserver:2). You can also transfer more parameters through full command as below:
vncserver :1 -geometry 1024x768 -depth 16 -pixelformat rgb565

(Note: If it prompts 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.)

4) Modify the size of ext4 file system

It could promote system performance via expanding the rootfs partitions of file system after writing image, which could avoid the problems caused by insufficient space.
Expanding rootfs partitions on TF card of PC:
Using GParted to adjust the size:
Select the specified letter, right-click the corresponding letter, select "change the size" to adjust into the desired size, click "adjust the size", close the dialog box and click "apply to all operations", select the "apply" to complete the expansion operation.

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).
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.
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.

vi. 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 bootable 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 offial Lubuntu and Raspbian could not use it.

III. Source Code Compilation of Android and Linux

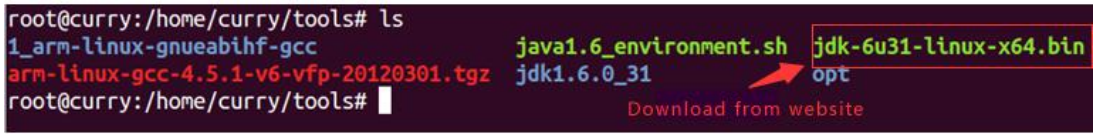
Hardware: Orange Pi development board*1, Card reader*1, TF card*1, power supply*1

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

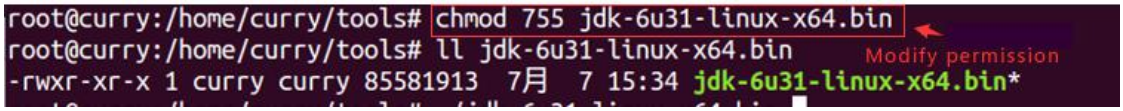
Linux host computer needs:
Version 2.7.3 of Python;
Version 3.81-3.82 of GNU Make;
JDK 6;
Version 1.7 or higher version of Git.

1. Install JDK

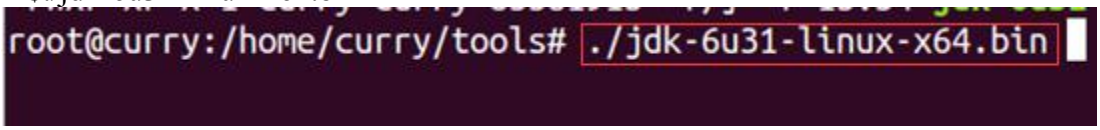
- Download and install JDK, you will get jdk-6u31-linux-x64.bin after downloaded:



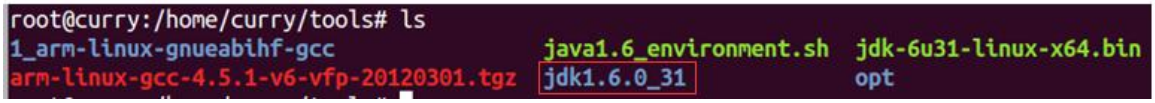
- Modify the permission of jdk-6u31-linux-x64.bin, which has no executable authority



- $./jdk-6u31-linux-x64.bin

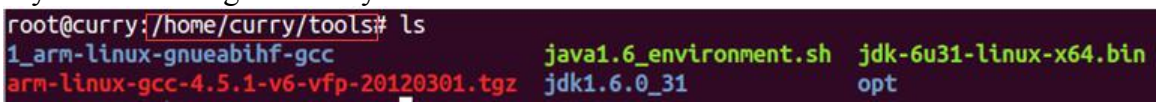


- It will generate a folder

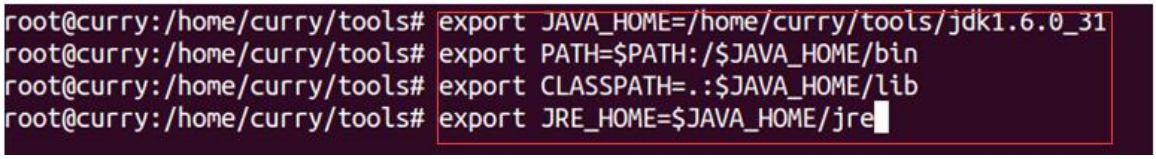


- Enter at terminal

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 whether it installed successfully and check whether the java is version 1.6.

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 the Source Package and Unzip it

Download website: http://www.orangepi.org/downloadresources/

Downloaded source package

```
$ tar –xf RDA8810_trunk.tar.gz
```

Unzip the file you will get the trunk directory, enter it via the terminal.

4. Android source code compiler

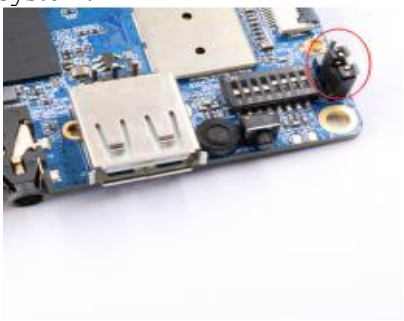
Enter in the command line:

```
$ cd /trunk
$ source build/envsetup.sh
```

$ lunch
Select slt-userdebug
Hardware targets select NollecA9V2VV8810P_ext4
$ make –j the value after # is the process of simultaneous compilation, host dependent configuration
$ cd /trunk/out/target/product/slt-NollecA9V2VV8810P_ext4

Insert U disk into computer and check whether the TF card has mounted(make sure the TF card has been formatted before this).
Use mount tool to check the U disk mount node number, suppose it is /dev/sdc

Execute the following command to run Android SD card image:
sudo dd if=bootloader.img of=/dev/sdc bs=512 seek=256 count=4096 && sync
sudo dd if=modem.img of=/dev/sdc bs=512 seek=12544 count=8192 && sync
sudo dd if=boot.img of=/dev/sdc bs=512 seek=20736 count=16384 && sync
sudo dd if=recovery.img of=/dev/sdc bs=512 seek=37120 count=20480 && sync
sudo dd if=system.ext4.img of=/dev/sdc bs=512 seek=57600 count=512000 && sync
sudo dd if=vendor.ext4.img of=/dev/sdc bs=512 seek=569600 count=512000 && sync
Insert the written image into Orange Pi and connect the Jumper Cap like the following, you could boot into Android system.



5. Compile Linux source Code

Input the following in the command line, enter to the RDA8810 trunk directory
$ cd /trunk
$ source build/envsetup.sh
$ lunch
Select slt-userdebug
Hardware targets select NollecA9V2VV8810P_ext4

compile uboot
$ make bootloader -j the value after # is the process of simultaneous compilation, host dependent configuration

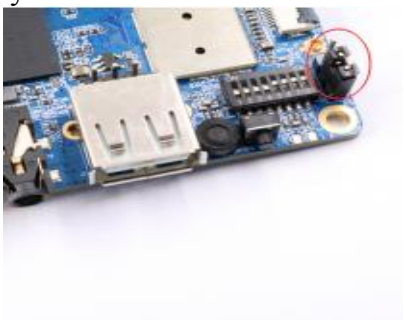
Compile kernel
$ make bootimage -j

Update the image, please write down original image into SD card, the execute the following:

$ cd /trunk/out/target/product/slt-NollecA9V2VV8810P_ext4

Insert U disk into computer and check whether the TF card has mounted(make sure the TF card has been formatted before this).
Use mount tool to check the U disk mount node number, suppose it is /dev/sdc

Execute the following command to run Linux SD card image
sudo dd if=bootloader.img of=/dev/sdc bs=512 seek=256 count=4096 && sync
sudo dd if=modem.img of=/dev/sdc bs=512 seek=12544 count=8192 && sync
sudo dd if=boot.img of=/dev/sdc bs=512 seek=20736 count=16384 && sync
Insert the written image into Orange Pi and connect the Jumper Cap like the following, you could boot into Android system.



IV. Orange Pi Driver development

In order to help developers more familiar with Orange Pi, this instruction will make a brief illustration on device driver module and application program.

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 */trunk/kernel/driver/misc:

Enter to */trunk/kernel/driver/misc

Modify Makefile on currently file, shown as following:

```
43 obj-$(CONFIG_SPEAR13XX_PCIE_GADGET) += spear13xx_pcie_gadget.o
44 obj-$(CONFIG_VMWARE_BALLOON)    += vmw_balloon.o
45 obj-$(CONFIG_ARM_CHARLCD)   += arm-charlcd.o
46 obj-$(CONFIG_PCH_PHUB)      += pch_phub.o
47 obj-y               += ti-st/
48 obj-$(CONFIG_AB8500_PWM)    += ab8500-pwm.o
49 obj-y               += lis3lv02d/
50 obj-y               += carma/
51 obj-$(CONFIG_USB_SWITCH_FSA9480) += fsa9480.o
52 obj-$(CONFIG_ALTERA_STAPL)  +=altera-stapl/
53 obj-$(CONFIG_MAX8997_MUIC)  += max8997-muic.o
54 obj-$(CONFIG_WL127X_RFKILL) += wl127x-rfkill.o
55 obj-$(CONFIG_SENSORS_AK8975)    += akm8975.o
56 obj-$(CONFIG_SUNXI_VIBRATOR)    += sunxi-vibrator.o
57 obj-$(CONFIG_SUNXI_BROM_READ)   += sunxi_brom_read.o
58 obj-$(CONFIG_NET)     += rf_pm/
59 obj-$(CONFIG_ORANGEPI_MISC) += OrangePi_misc.o
```

Re-modify Makefile

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.

```
config SUNXI_BROM_READ
    tristate "Read the BROM infomation"
    depends on ARCH_SUN8I
    default n
    ---help---
     This option can allow program access brom space by the file node.

config ORANGEPI_MISC
    tristate
    default n
```

Modify Kconfig

Back to the source code directory /trunk:

$ make bootimage

Make sure it have already finished make-engineer-configuration before execute this command, if not, please refer to last section about Linux source code compilation.

Update the new generated module file into Linux system.

It will show on *cd /trunk/out/target/product/slt-NollecA9V2VV8810P_ext4/obj/KERNEL/out/target/product/slt-NollecA9V2VV8810P_ext4/obj/KERNEL/modules/lib/modules/3.10.62-rel5.0.2/ generated corresponding .ko file, it is the module that generated after OrangePi_misc.c compilation.

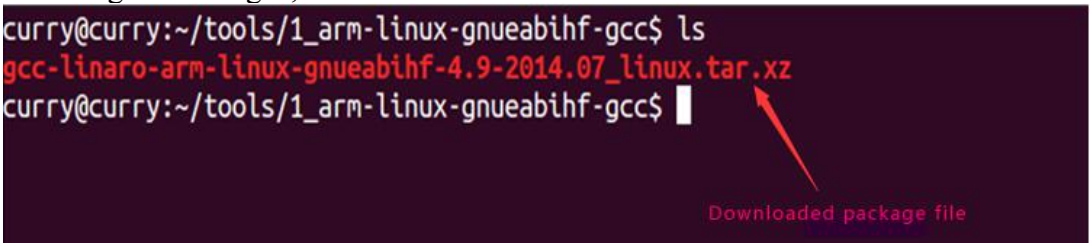
Insert U disk (please note the SD card should have written image) if the SD card is mounted to the directory system of /dev/sdc, then SD card will mount to rootfs, which is /dev/sdc7, and mounted to rootfs partition automatic.

Copy the directory file:
/trunk/out/target/product/slt-NollecA9V2VV8810P_ext4/obj/KERNEL/out/target/product/slt-NollecA9V2VV8810P_ext4/obj/KERNEL/modules/lib/modules/3.10.62-rel5.0.2/
into:
/media/*/lib/modules/

3. Compiling method of application

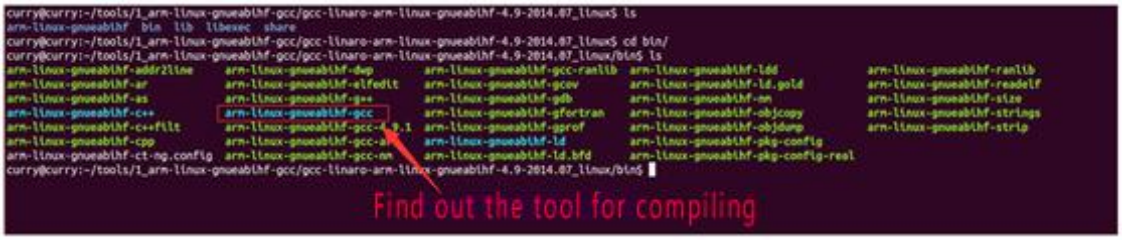
Check whether there is the cross compiler, if not, then download and install it.
$ arm-linux-gnueabihf-gcc -v

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



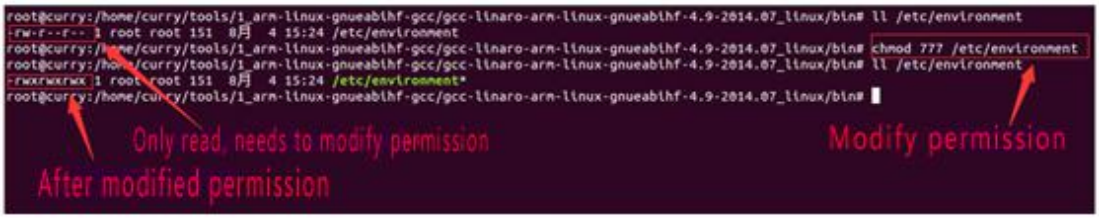
Unzip the downloaded file and enter the the directory

Check the information after entering bin directory

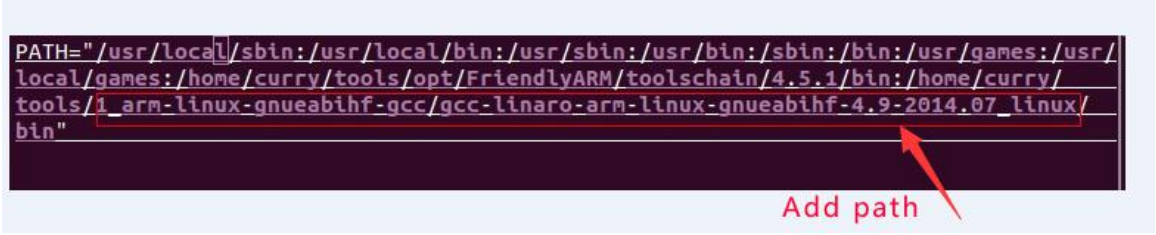


pwd hows the path and export it into the whole project

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



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/

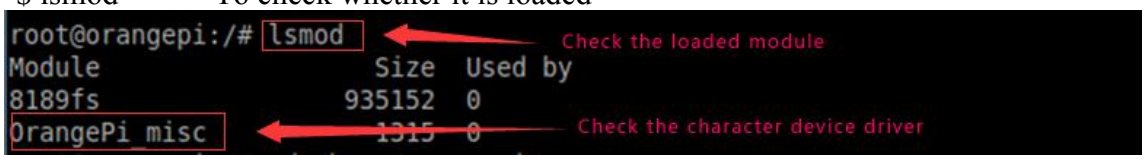
4. Running driver and application

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

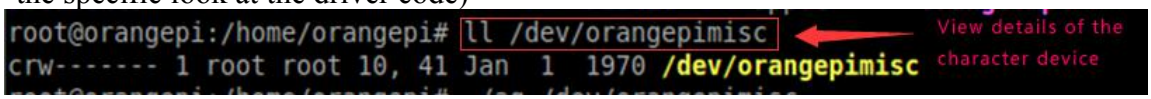
You need to switch to root users and load module driver module to the development board first.

$ 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, check the code for specify)

$ ./aq /dev/orangepimisc

V. 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

TTL to USB cable

Operations 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.

Android baud rate set as 961200

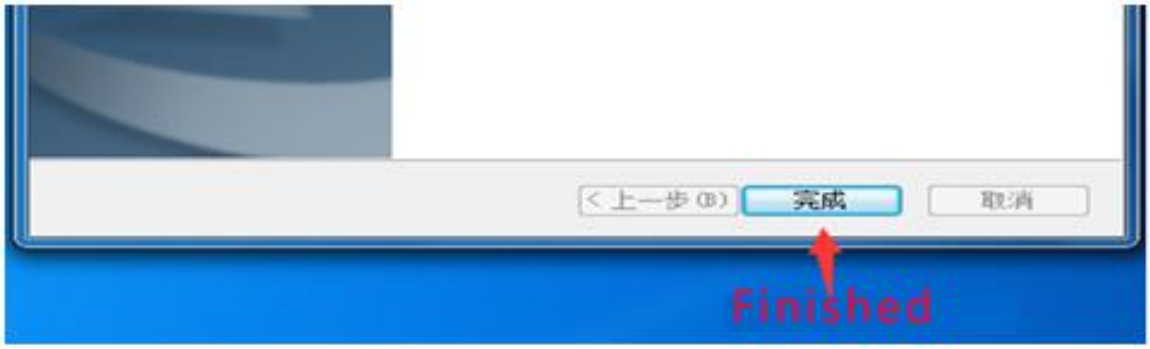
Linux baud rate set as 115200

1. Install USB driver on Windows

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

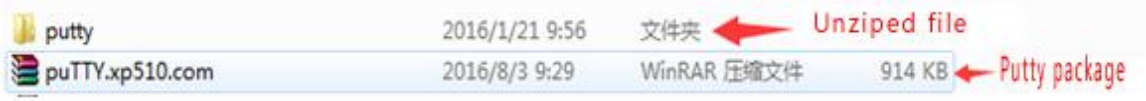
- Choose application installation as Administrator

- Wait for installation completing



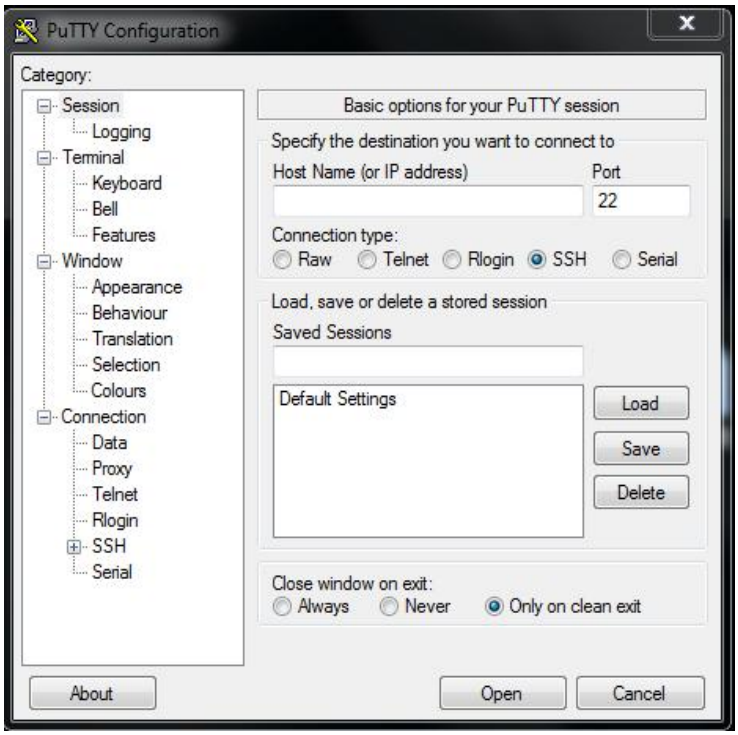
2. Install putty on Windows

- Download putty installation package



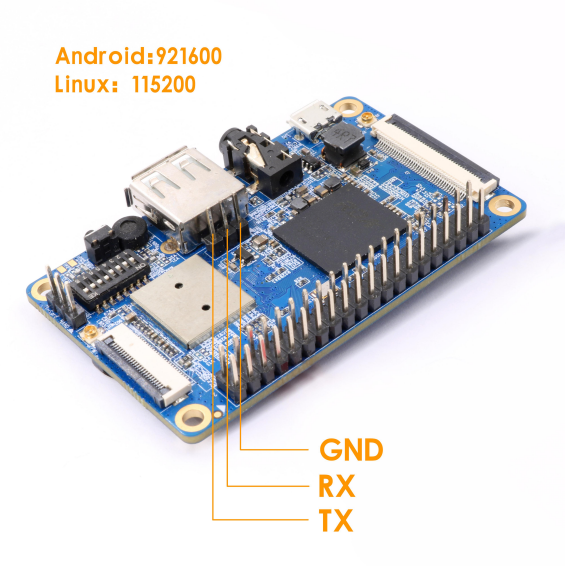
- Unzip and install it

- Open it after installed, shown as below:



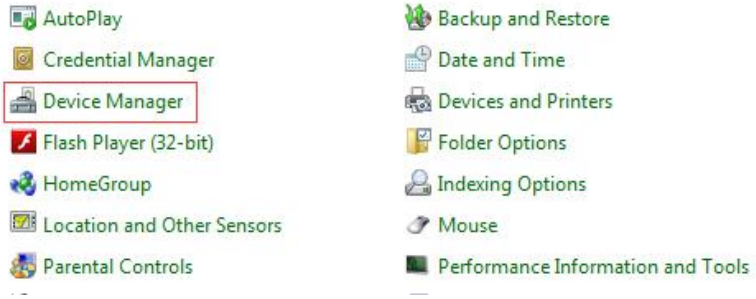
3. Connect style

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

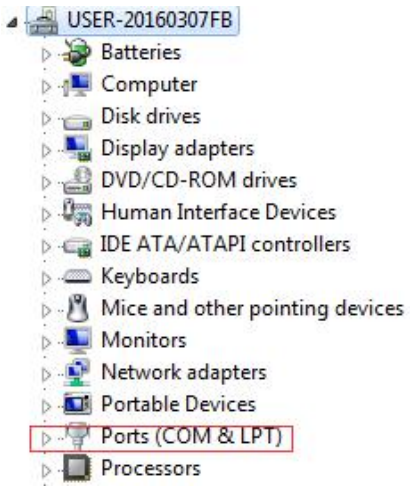


4. Equipment information acquisition

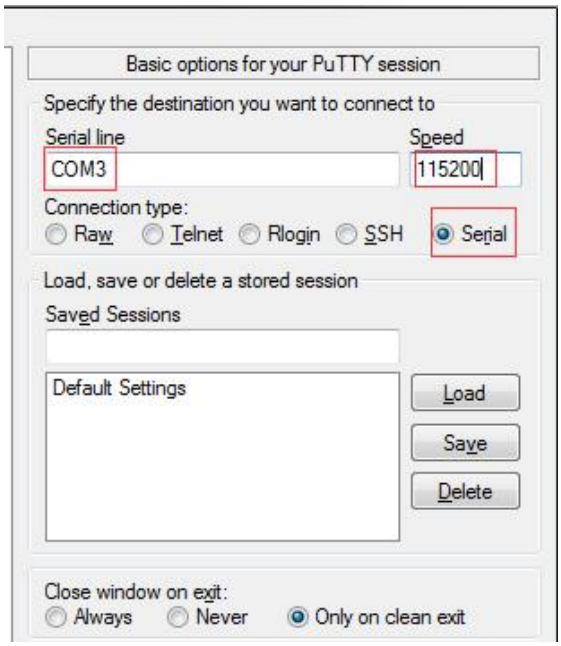
- Select *control panel on Start* menu



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



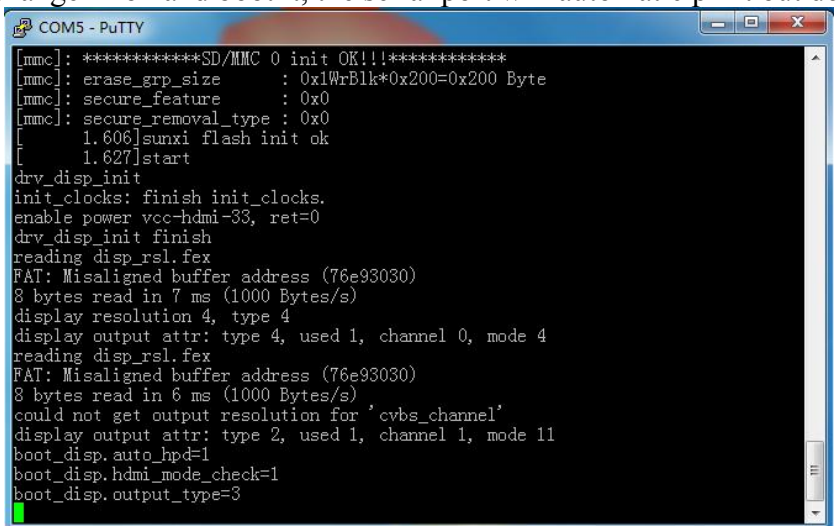
5. Putty configuration



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

6. Start debug

- Power Orange Pi on and boot it, the serial port will automatic print out debug log.



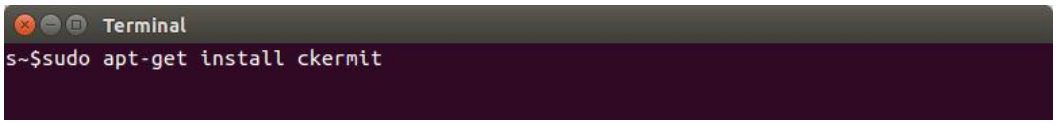
Operations on Linux

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

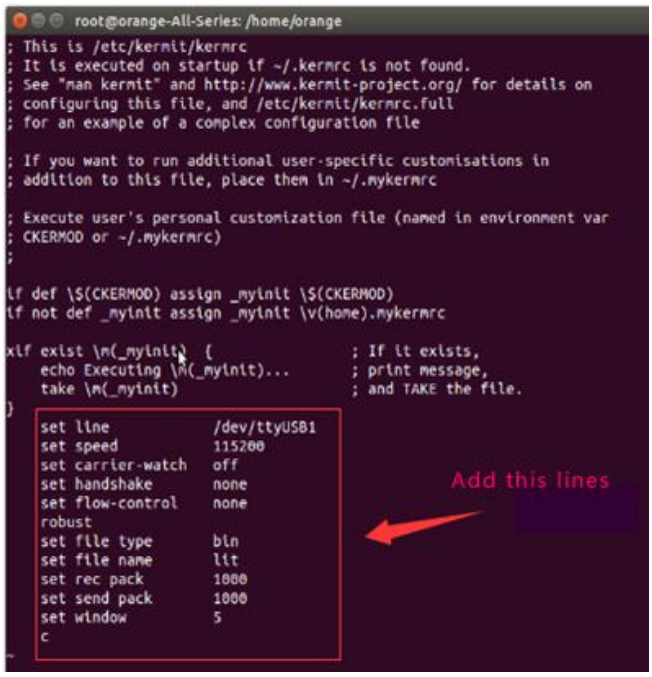


- Configurate Kermit

  $ sudo vi /etc/kermit/kermrc

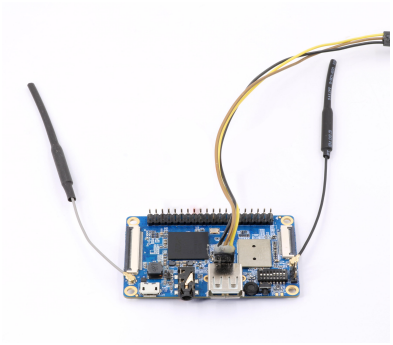
- Add lines：

```
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
```



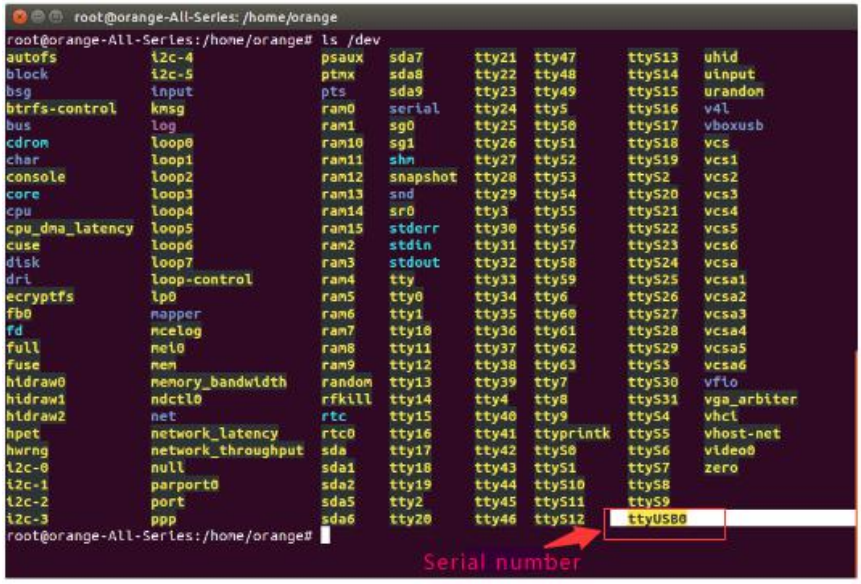
2. Connect method for debug

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

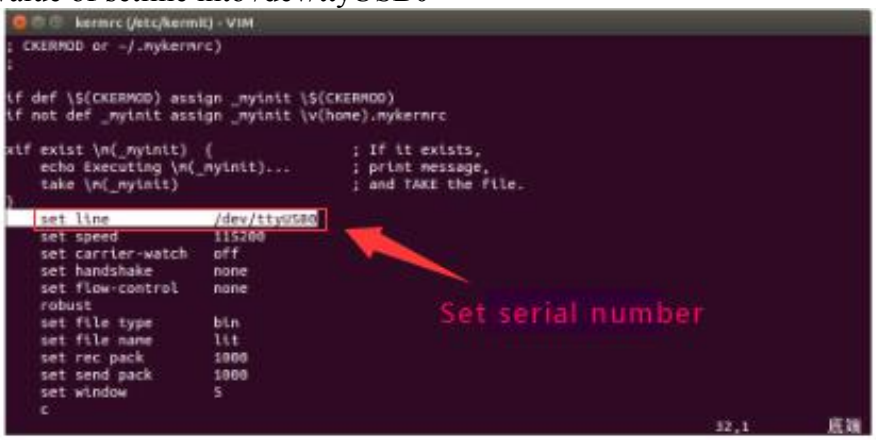


3. Equipment information acquisition

$ ls /dev/ (Input command in the PC terminal to check the device number of TTL to the serial cable)

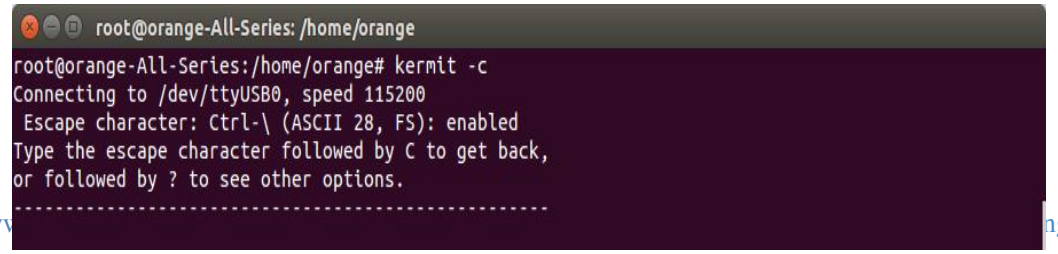


- It can be seen from the figure that TTL to the serial port cable is identified as ttyUSB0, configure the /ect/kermit/kermitc file, update the serial port information.
  $ sudo vi /etc/kermit/kermitc
- Set the value of setline into /dev/ttyUSB0



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, the account and password ard root/orangepi and orangepi/orangepi