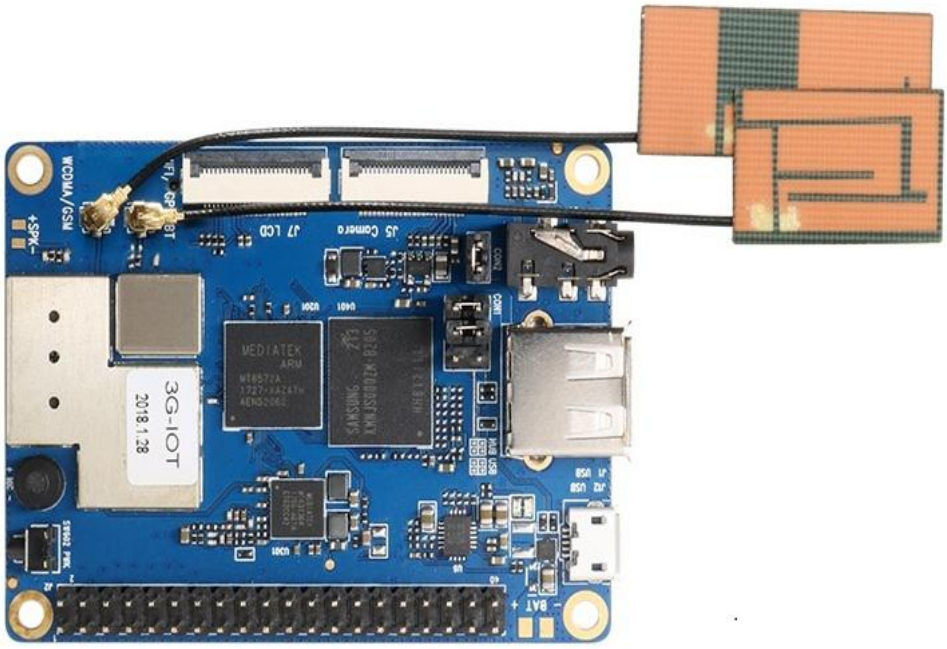
Orange Pi 3G-IoT
User Manual



History

Ver	Data	Author	Brief	Publish	Memo
1.1	2018-05-25	Engineer Pan	Create Documentation	2018-6-14	
1.2	2018-10-18	Engineer Pan	Usage of GPIO	2018-10-19	

Contents

I. Orange Pi Introduction

1. What is Orange Pi 3G-IOT?

It's an open-source single-board computer. It can run Android4.4、Linux. It uses the MTK serial MT6572 CPU.

2. What can I do with Orange Pi 3G-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 3G-IOT is open source.

3. Who is it for?

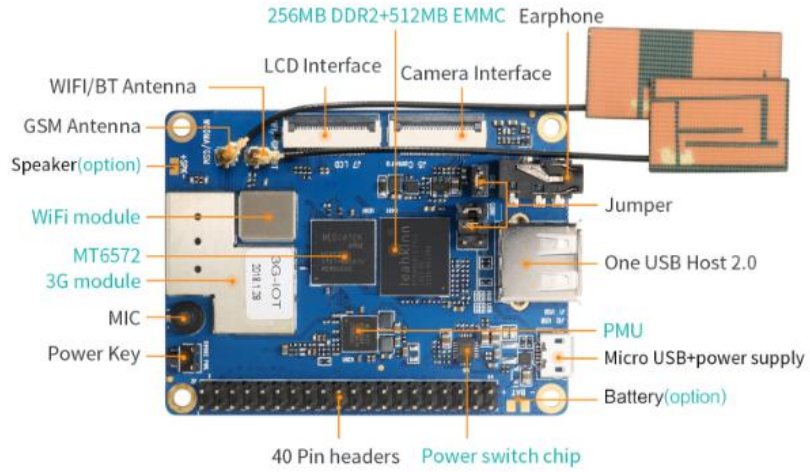
Orange Pi 3G-IOT is for anyone who wants to start creating with technology – not just consuming it. It's a simple, fun, useful tool that you can use to start taking control of the world around you.

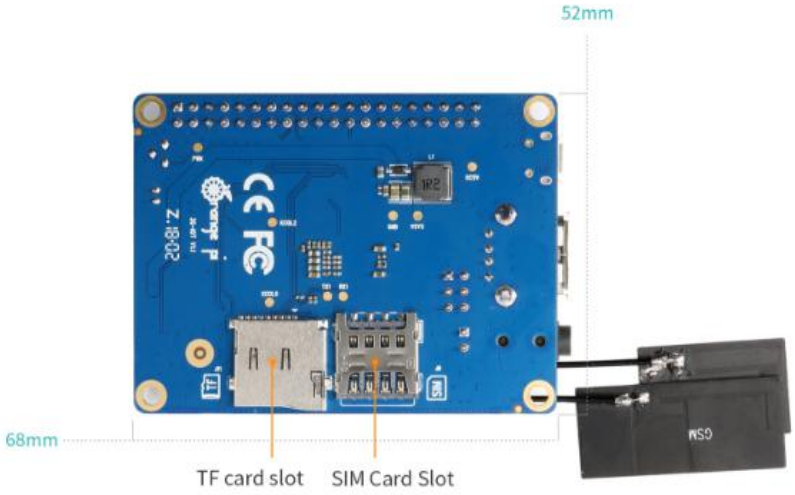
4. Orange Pi 3G-IOT Hardware Specification

OrangePi 3G-IoT-256MB Hardware Specification	
Processor	MT6572

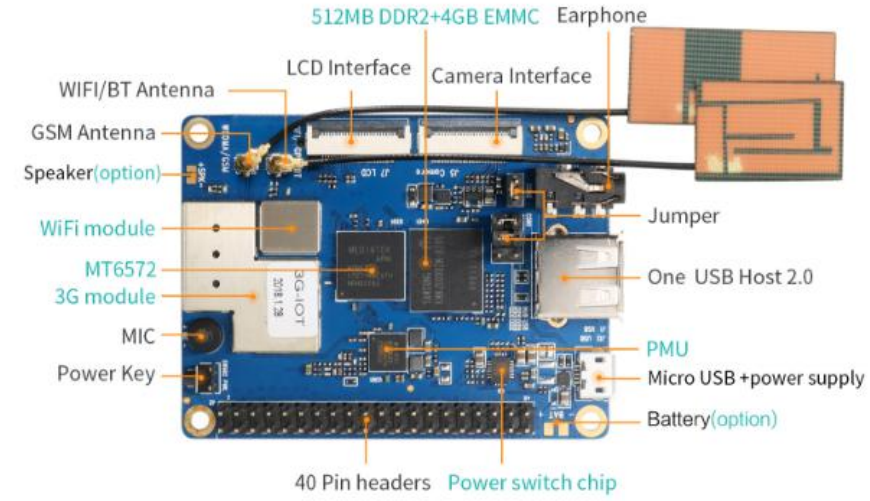
CPU		Quad core ARM® Cortex-A7, Main frequency up to 1.25GHz
GPU		ARM Mali-400MP1 GPU
Memory		Version A: 256MB DDR2+512MB EMMC Flash Version B: 512MB DDR2+4GB EMMC Flash
Wireless		WIFI / BT / FM / GPS Four in one
Radio frequency	GSM	850/900/1800/1900
	WCDMA	B1/B2/B5/B8
	TD-CDMA	/
	CDMA2000	/
Display		FWVGA(FPC zif Connector)
Capacitance touch		Support (FPC zif Connector)
Camera		MIPI Connector
SIM Card		mini Single SIM Card
TF Card		Support hot-plugging
Audio	Earphone	For audio input / output
	Mic	For audio input
USB Port		One USB Host, one Micro USB
LED		Red for power indicator, Green for status indicator
Key		Power button(SW602)
Low-level peripherals		40pin Expansion Headers: GPIO 1.8V, SPI × 2 , I2C × 3, UART×2
Power Supply		Micro USB(5V/2A)，Battery(Optional)
Software		
OS		Android 4.4
Programming support		C, C++, Kotlin, Java, Shell, Python, etc

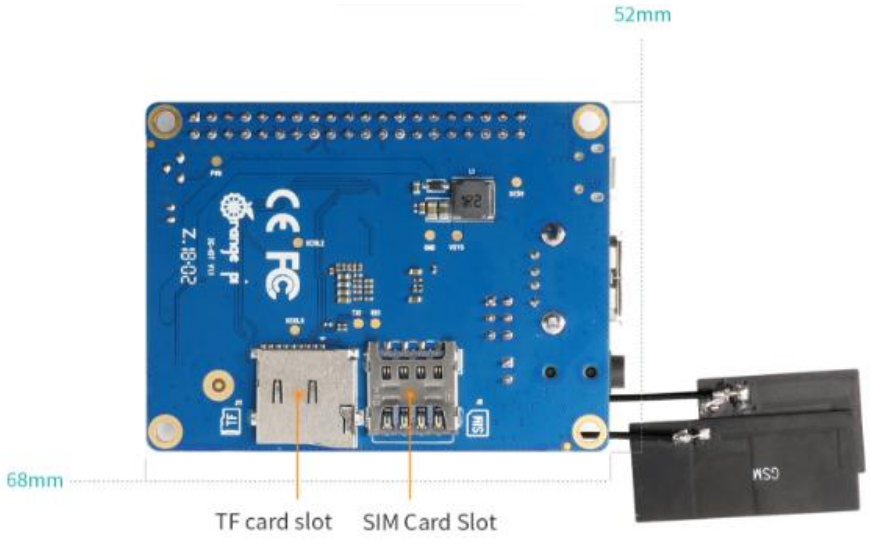
256MB DDR2 + 512MB EMMC Version Specs:





512MB DDR2 + 4GB EMMC Version Specs:





5. GPIO Specs

PIN1	VIO28 PMU		**PIN21**	GPIO24
PIN2	DC5V		**PIN22**	GPIO109
PIN3	SDA_1		**PIN23**	GPIO25
PIN4	DC5V		**PIN24**	GPIO139
PIN5	SCL_1		**PIN25**	GND

PIN6	GND		**PIN26**	EINT0/GPIO30/PWM B
PIN7	EINT4/GPIO26		**PIN27**	SDA_0
PIN8	UTXD2		**PIN28**	SCL_0
PIN9	GND		**PIN29**	SPI_CS
PIN10	URXD2		**PIN30**	GND
PIN11	URXD1		**PIN31**	SPI_CK
PIN12	EINT10/GPIO56		**PIN32**	EINT15/GPIO90
PIN13	UTXD1		**PIN33**	SPI_MO
PIN14	GND		**PIN34**	GND
PIN15	EINT2/GPIO128/PW A		**PIN35**	SPI_MI
PIN16	EINT12/GPIO58		**PIN36**	GPIO145/PWMBL
PIN17	VIO28 PMU		**PIN37**	EINT3/GPIO27
PIN18	EINT14/GPIO89		**PIN38**	EINT5/GPIO144/PWM B
PIN19	GPIO140		**PIN39**	GND
PIN20	GND		**PIN40**	GPIO141

II. Using Method

1. Prepare the Hardware and Software

Hardware Requirement:

- Orange Pi 3G-IoT Development Board
- A PC for compilation with following specs:

  64bit CPU

  Up to 16GB RAM

  UP to 40GB spare disk space

  Operation system should up to Ubuntu12.04, it would be better if it is Ubuntu16.04

You could refer to Google file for more details: https://source.android.com/source/building

Software Requirement:

- Orange Pi 3G-IoT SDK
- Orange Pi 3G-IoT Firmware
- Android-image-flash-tool

2. Power Methods

There are two methods for power supply:

- Micro USB（5V 2A）in for power:
- Battery in for power:

Usually use 4.2V battery to solder on the back side of the development board.

3. Before Usage

After receiving the product, please put the antennas of the product from the position of Pic 1 to the position of Pic 2 (or to the outside of the board), which can not be attached to the board so as not to affect the signal.


range pi
Regulatory Compli
Safety Informatio
range pi
A new generation of minipc
Getting Started

图 1


range pi
Regulatory Compli
Safety Informatio
range pi
A new generation of minipc
Getting Started

图 2

III. Android Compilation Environment Construction

1. Download SDK compression package

Take OrangePi_3G-IoT_Android4.4_V1.0_2018.tar.gz as an example, after get the original compression package:

```
mkdir OrangePi_3G-IoT
tar zxvf OrangePi_3G-IoT_Android4.4_V1.0_2018.tar.gz -C OrangePi_3G-IoT
cd OrangePi_3G-IoT
```

2. Construct Compilation Environment

It could also refer to Google file: http://source.android.com/source/initializing.html

- **Install JDK**

Compilation of Android6.0 is base on JAVA6, it needs to first install OpenJDK, gcc4.4 before compilation.

Command for installing:

```
sudo apt-get install  openjdk-6-jdk
sudo apt-get install gcc-4.4
```

If the installation path is /usr/lib/jvm/jdk1.6.0_31 when configure environment variable of JAVA, then execute the the following command on terminal to configure the data.

```
export JAVA_HOME=/usr/lib/jvm/jdk1.6.0_31
export PATH=$JAVA_HOME/bin:$PATH
export CLASSPATH=.:$JAVA_HOME/lib:$JAVA_HOME/lib/tools.jar
```

- **Install Software Package**

For Ubuntu12.04:

```
sudo apt-get update
sudo apt-get install git-core gnupg flex bison ccache gperf libsdl1.2-dev
libesd0-dev libwxgtk2.6-dev build-essential zip curl libncurses5-dev
zlib1g-dev valgrind libc6-dev lib32ncurses5-dev x11proto-core-dev
libx11-dev lib32readline-gplv2-dev lib32z1-dev libgl1-mesa-dev gcc-4.4
```

```
g++-4.4 g++-4.4-multilib g++-multilib mingw32 tofrodos python-markdown
libxml2-utils xsltproc wine
```

For Ubuntu14.04:

```
sudo apt-get update
sudo apt-get install git-core gnupg flex bison ccache gperf libsdl1.2-dev
libesd0-dev libwxgtk2.8-dev build-essential zip curl libncurses5-dev
zlib1g-dev valgrind libc6-dev lib32ncurses5-dev x11proto-core-dev
libx11-dev lib32readline-gplv2-dev lib32z1-dev libgl1-mesa-dev
g++-multilib g++-4.8-multilib mingw32 tofrodos python-markdown
libxml2-utils xsltproc libc6-dev-i386 lib32z1 lib32ncurses5 lib32bz2-1.0
lib32readline-gplv2-dev wine
```

We could process to SDK compilation after finished the above.

3. Compilation of SDK Source Code

Full compilation

There are many compilation shell scripts for development.

Directory is: SDK/code/orangepi/scripts

```
$ cd code/orangepi/scripts
$ ls
anr_kk.sh  auto.sh  clean.sh  init_project.sh  tar_img.sh
```

auto.sh is automatically compilation script

clean.sh is automatically scavenging the compiled result script

tar_img.sh is packing script

One the directory of code/orangepi/scripts, we could execute automatically compilation script:

```
If the board is: 32g4g
$ ./auto.sh IoT03_mt6572_emmc_b1258_32g4g_ry_smt v00 eng

If the board is: 4g2g
$ ./auto.sh IoT03L_mt6572_lca_b1258_wg_4g2g_ry_smt v00 eng
```

The meaning of the parameter is:

#$1 project_info [eg: IoT_bd6737m_35g_b_m0_op_smt_hd720_pcb_v2]

#$2 version_info [eg: v00 v01 ...]

#$3 compile_mode [eng:user userdebug eng]

Module compilation

Usually use the following command to change Kernel file:

./mk -o=TARGET_BUILD_VARIANT=[user/userdebug/eng] projName n

K && ./mk -o=TARGET_BUILD_VARIANT=[user/userdebug/eng] projName r bootimage

For example, if project is: hexing72_cwet_kk

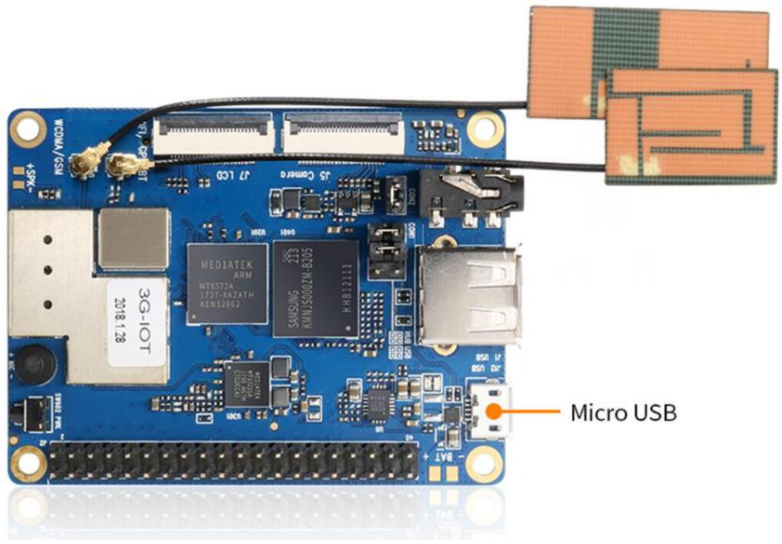
./mk -o=TARGET_BUILD_VARIANT=eng hexing72_cwet_kk n k &&

./mk -o=TARGET_BUILD_VARIANT=eng hexing72_cwet_kk r bootimage

You could refer to MT6592_Driver_All_In_One_Part.pdf (after unzip SDK you will have this file) for modification of driver compilation.

IV. Android Firmware Flashing

Relevant keys and connectors for firmware flashing of 3G-IOT:



After compilation, all the firmware will generate on the directory of: code/IoT03_b1258_32g4g_ry_smt or code/IoT03L_b1258_wg_4g2g_ry_smt, and packed into compression file as the name of IoT03_b1258_32g4g_ry_smt_20180403182516_v00_eng.zip.

```
IoT03L_b1258_wg_4g2g_ry_smt
├──── images
│       ├──── boot.img
│       ├──── lk.bin
│       ├──── logo.bin
│       ├──── MT6572_Android_scatter.txt
│       ├──── preloader_hexing72_cwet_lca.bin
│       ├──── ramdisk.img
│       ├──── ramdisk-recovery.img
│       ├──── recovery.img
│       ├──── secro.img
│       ├──── system.img
│       └──── userdata.img
└──── modem
    ├──── APDB_MT6572_S01_MAIN2.1_W10.24
    ├──── _APDB_MT6572_S01_MAIN2.1_W10.24.check
    ├──── APDB_MT6572_S01_MAIN2.1_W10.24_ENUM
```

```
    ├──── BPLGUInfoCustomAppSrcP_MT6572_S00_MOLY_WR8_W1315_MD_WG_MP_V47_1_wg_n
    ├──── catcher_filter_1_wg_n.bin
    ├──── DbgInfo_WR8.W1315.MD.WG.MP_HEXING72_CWET_KK_HSPA_MOLY_WR8_W1315_MD_WG_MP_V47_2017_10_26_22_35_1_wg_n
    ├──── modem_1_wg_n.img
    └──── modem_1_wg_n.mak
```

After the above steps, you could refer to the previous section to flash image.

Or you could use the image we have compiled and pack: http://www.orangepi.org/downloadresources/

Unzip Android6.0 image via the following command:

```
$ tar zxvf IoT_op_smt_hd720_pcb_v2_v00_eng_20180126140300.tar.gz
```

You could get the list of firmware after ran the above command, or you could also compile by yourself with reference of previous section.

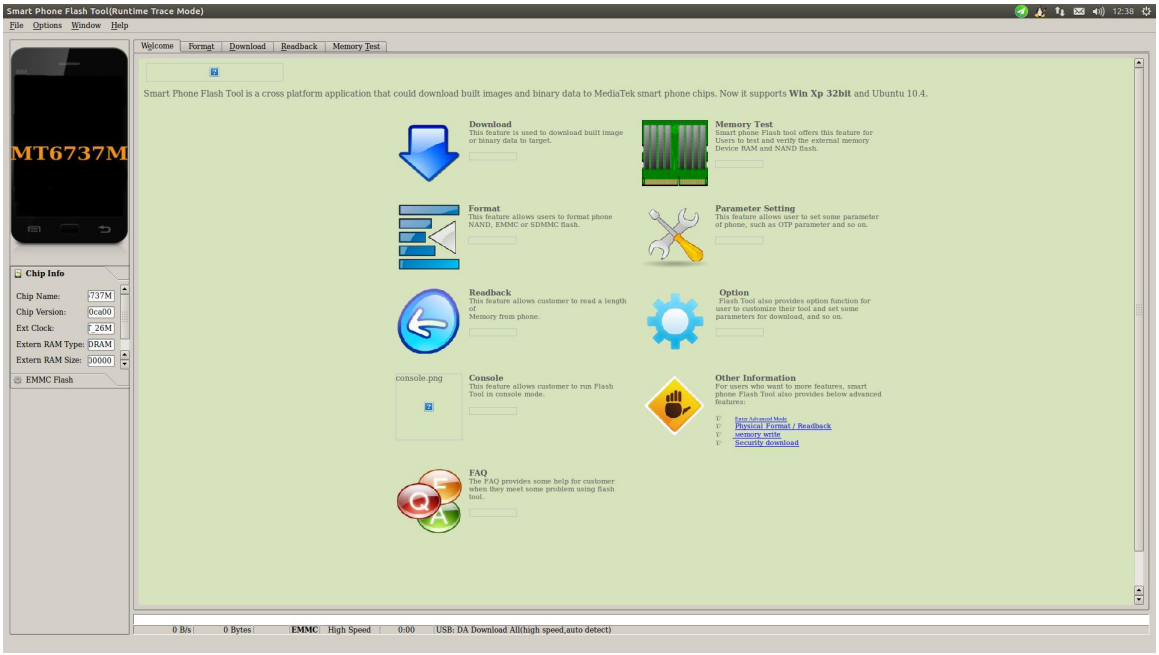
Supporting OS of PC:

- Windows 10
- Windows 7 (32/64bit)
- Windows 8 (32/64bit)
- Ubuntu10.04 / 12.04 / 14.04 (32/64bit)

1. Flash Tool Introduction

You could download the **Smart Phone Flash Tool** on the download page of Orange Pi 3G-IoT section. There are tools for Windows and Linux version, you could select a suitable version according to your PC environment.

Interface like the following:



Using method for both Windows and Linux versions are same, here will illustrate with Linux version.

2. Method for Image Flashing

```
MTK suggesting Ubuntu12.04 verstion, if it is not this version and cannot connect to
PC, then need to:
$sudo apt-get remove modemmanager
$sudo /etc/init.d/udev retstart
Reboot your PC

Download corresponding MTK driver according to your OS when running on Windows 解压
Driver_Auto_Installer_EXE_v5.1453.03.rar
Install
```

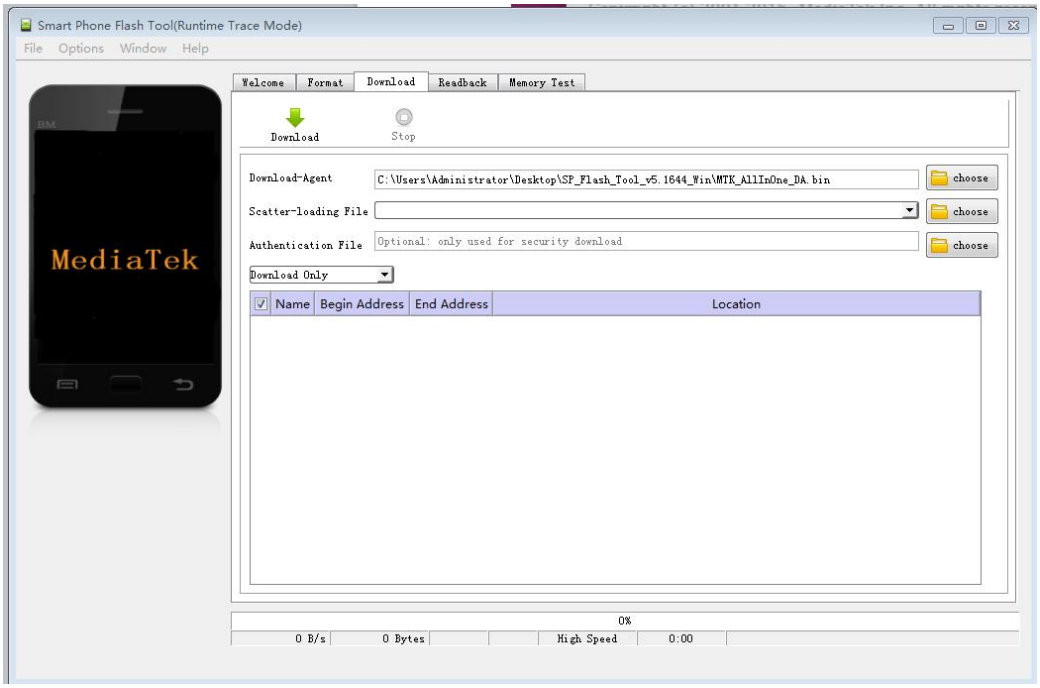
- **Unzip and open flash tool**

```
$ unzip SP_Flash_Tool_v5.1644_Linux.zip
$ cd SP_Flash_Tool_v5.1644_Linux
$ sudo ./flash_tool.sh
```

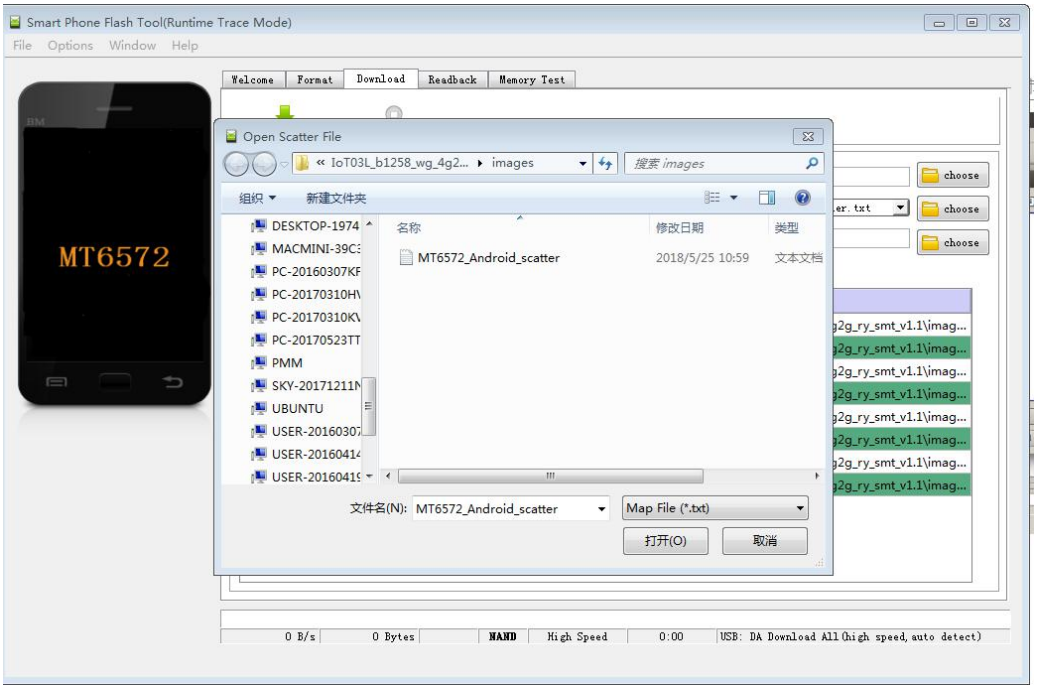
If it is the first time you use this software, you might receive the warn like the following. It is normal to receive this, you could click OK enter into the software. In the future you could manually specified the path of Scatter File.

- **Enter into flash mode**

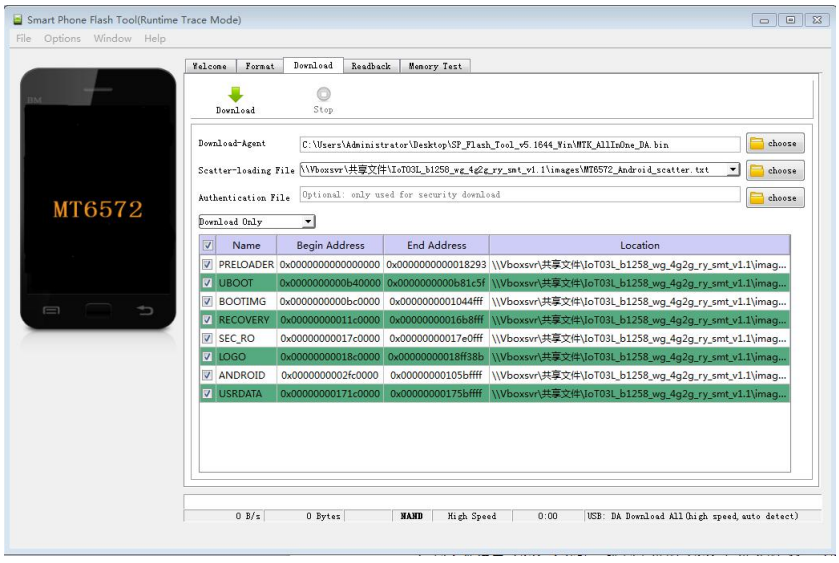
a. Switch into Download page like the following:



b. Click choose on the right side of Scatter-loading File and select the path of Scatter File like the following:



c. After double click the selection, the **partition information display section** will automatically fill the path of each partition file and the absolute starting address to which they are to be flashed.



d. In the top left corner of **partition information display section**, there would be a drop-down menu. Three of this options:

Format All + Download // Format all information on the partitions and re-download the selected partition

Firmware Upgrade // Update the difference on the selected partition

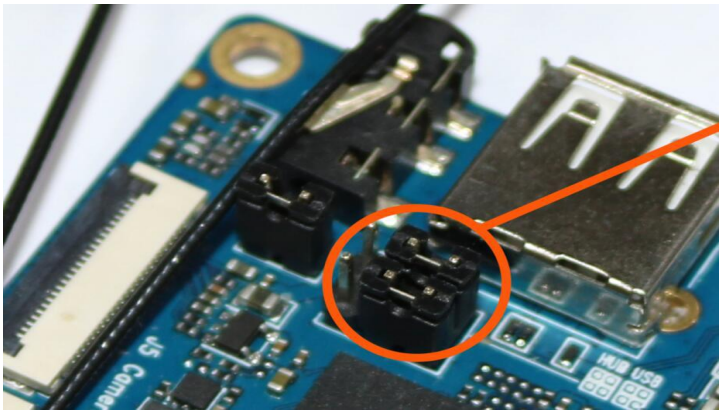
Download Only // Re-download no matter there is difference or not

Please note it: Usually update firmware you only need to select **Firmware Upgrade**, please **do not**

select **Format All + Download**

It you select Format All, you will lose the calibration information which we worked before sending out products. If this situation is inadvertent, please contact the Orange Pi service and obtain the calibration parameters through the machine code, and re-flash the calibration parameters.

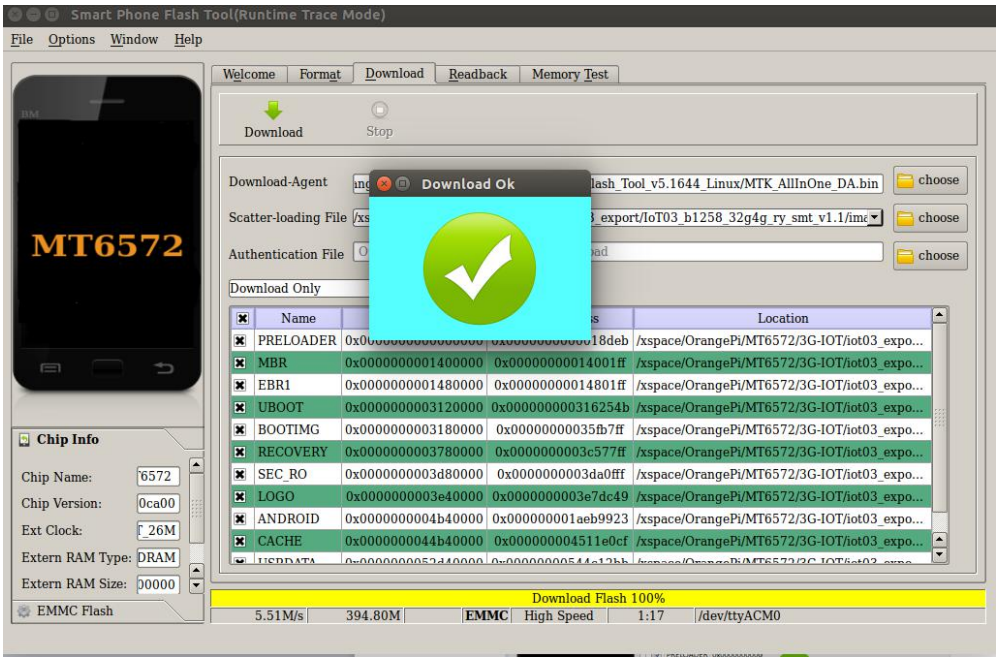
e. Insert Jumper cap：



f. Then connect with USB on PC and Mirco USB on Orange Pi via USB cable, the red LED will light up.

g. Click Download button 

h. The interface would show like the following after downloaded:



i. Take of the USB cable and insert Micro USB power supply

Wait around 5 seconds, it will display the charging interface of shutdown

When the Power button is loosened after 5 seconds, the system will start to enter the system
When the updated partition is more, the first boot will take a long time (the full partition update needs 8min), please be patient.

Windows OS would require to download SP_Flash_Tool_v5.1644_Win.zip, unzip and install it. Then could refer to the manual of Linux to flash image.

V. Usage of GPIO

Modify and Display of the GPIO Status Under ADB Mode

Get the Open/Close Status of GPIO : cat /sys/devices/virtual/misc/mtgpio/pin

Enter the command under ADB mode: cat /sys/devices/virtual/misc/mtgpio/pin , then it will show：

pin: [mode] [pull_sel] [din] [dout] [pull en] [dir] [ies] [smt]

0:11101010

1:01101010

、 、 、 、 、 、

22:1-100-10-1-1

、 、 、 、 、 、 、

42：00000110

Corresponding meaning per row：

IO Number：mode，pull select，input value，output value，pull enable，direction，ies

Modify the Status of GPIO：

You could get the 40pin GPIO specifications from this manual or from schematic which have been uploaded to our official website: http://www.orangepi.org/downloadresources/. For example the 37th pin on 40pins is GPIO123,

echo -wdout123 1 > pin -This is Set the GPIO to output high level

echo -wdout123 0> pin -This is Set the GPIO to output high level