Waveshare Electronics
Address: Rm 813, Dynamic World Building, Zhenhua Rd, Futian District, Shenzhen, China
TEL: +86-755-83040832 Web: http://www.wvshare.com/
FAX: 86-755-83042572 E-mail: sales@wvshare.com

E9 Mini PC User Manual

July 10th, 2014 (V2.0.0)

Waveshare Electronics
Website: http://www.wvshare.com/

Waveshare Electronics
Address: Rm 813, Dynamic World Building, Zhenhua Rd, Futian District, Shenzhen, China
TEL: +86-755-83040832 Web: http://www.wvshare.com/
FAX: 86-755-83042572 E-mail: sales@wvshare.com

Copyright Statement

Preface

E9 mini PC is a kind of Card-PC based on i.MX 6 quad processor, using 40nm process, quad core and ARM Cortex-A9 architecture. The maximum frequency for a single core is up to 1.2 GHz, and it is with 1MB L2 Cache, supports 1080P@60fps decoding and 1080P@30fps encoding. Meanwhile, it supports 1080P HD video. Besides, it supports HDMI 1.4, USB 2.0, SATA-II and Gigabit Ethernet card.

As time is limited, it is hard to avoid flaws and mistakes in the manual. Sincerely welcome some valuable advice for our omissions and deficiencies parts; please contact us by email (service@wvshare.com). Let's make progresses together.

Catalogue

E9 Mini PC User Manual 0
July 10th, 2014 (V2.0.0) 0
Copyright Statement 1
Preface 2
Catalogue 3
Chapter One Introduction of E9 6
1.1 E9 Introduction 6
1.2 E9 diagram 7
Chapter Two E9 Usage Instructions 9
2.1 E9 Data Introductions 9
2.1.1 E9 Software Tools Instructions 9
2.1.2 E9 Image Instruction 9
2.1.3 E9 Ubuntu Source Code Instruction 9
2.1.4 E9 Android Source Code Instruction 10
2.1.5 E9 linux Source Code Instruction 10
2.1.6 E9 Other Data Instruction 10
2.2 E9 Software Introductions 10
2.2.1 u-boot Introduction 10
2.2.2 Kernel Introductions 10
Chapter Three Development Environment Set Up 13
3.1 Ubuntu Platform Environment Set Up 13
3.1.1 Ubuntu Compile Environment Set Up 14
3.1.2 Android Development Environment Set Up 16
3.2 Ubuntu Tools Installation and configuration (minicom) 19
3.2.1 Install minicom 19
3.2.2 Config minicom 19
3.3 Windows Platform Environment Set Up 22
Chapter Four Uboot Use 22
4.1 Interpretation of the Boot Jumper Cap 23
4.2 Boot the E9 mini PC 24
4.3 Resolution Settings 24
4.4 The Default Boot System Settings 26
Ubuntu 27
Introduction 28
Chapter Five Ubuntu Source Code & Compile 29
5.1 Ubuntu Source Code Drivers 29
5.2 Compile Ubuntu Source Code 30
5.2.1 Compile Bootloader 30
5.2.2 Compile Ubuntu Kernel 33
5.2.3 Ubuntu File System 36
Chapter Six Ubuntu System Burning 37
6.1 Preparations 37

6.1.1 Boot Modes Selection 37
6.1.2 Burning the Image 38
6.2 Windows Platform Burning 38
6.2.1 Burning Ubuntu12.04 System 39
6.2.2 Hardware Connection 40
6.2.3 Begins to burn 41
6.3 Linux Platform Burning 43
6.3.1 Create SD Card 44
6.3.2 Prepare to burn image 44
6.3.3 Burning ubuntu image 44
Chapter Seven Ubuntu Functional Test 47
7.1 Display Test 47
7.2 USB Test 47
7.3 Network Test 48
7.4 TF Card Test 49
7.5 Music Test 49
7.6 Video Test 50
7.7 USB Camera Test 51
7.8 Serial Port Test 51
7.9 Hard Disk Test 51
7.10 Others 51
Android 53
Introduction 53
Chapter VIII Android Source Code & Compile 54
8.1 Android Source Code: 54
8.2 Andorid Compiling 55
Chapter Nine Android System Burning 57
9.1 Preparations 57
9.1.1 Boot Mode Selection 57
9.1.2 Burning the Image 58
9.2 Windows Platform Burning 58
9.2.1 Burning Android System 58
9.2.2 Hardware Connection 60
9.2.3 Begins to Burn 61
9.3 Linux Platform Burning 63
9.3.1 Create SD Card 64
9.3.2 Prepare to Burn Image 64
9.3.3 Burning System 65
Chapter Ten Android Testing 67
10.1 TF Card Testing 67
10.2 U Disk Testing 69
10.3 SATA Testing 70
10.4 Network Card Testing 71
10.5 Camera Testing 71

10.6 Audio Testing 72
10.7 Video Testing 73
10.8 HDMI Testing 74
Linux 76
Introduction 76
11.1 Linux driver Source Code 77
11.2 Linux Source Code Compile 78
11.2.1 Compile Bootloader 78
11.2.2 Linux kernel compile 78
11.2.3 Linux File System 81
Chapter Twelve Linux System Burning 82
12.1 Preparations 82
12.1.1 Boot Modes Selection 82
12.1.2 Burning the Image 83
12.2 Windows Platform Burning 83
12.2.1 Burning Linux System 84
12.2.2 Hardware Connection 86
12.2.3 Begin to Burn 87
12.3 Linux Platform Burning 89
12.3.1 Create SD Card 90
12.3.2 Ready to Burn Image 90
12.3.3 Burning Linux Image 90
FAQ 93

Chapter One Introduction of E9

1.1 E9 Introduction

E9 is a new quad-core mini PC. Its biggest advantage is that its size is slightly bigger than common used IC card, but it owns interfaces are as rich as computer peripherals. Namely, having an E9, you will have a hand-held mini PC.

E9 is the 2nd generation mini PC using Freescale Cortex-A9 I.MX6Q platform. It is mainly used in various kinds of terminal display devices. Its size is 100 x 72 mm, and supports rich interfaces such as FPC, HDMI, VGA, LVDS, UART, USB, OTG, SATA, Micro SD CARD, Camera, 3G, CAN Bus, G-Sensor, Wifi, BlueTooth, Ir, RTC and supports other functional modules for peripheral extension.

Launched by Freescale, i.MX6Q is a new generation application processor for consumer electronics, industrial, automobile audio entertainment system and many other fields. It uses ARM Cortex-A9 kernel, 40nm process technology, the maximum operation frequency up to 1.2GHz with ARMv7TM, Neon, VFPV3 and Trustzone support. The processor is 64/32bit bus structure,32/32KB Internal Cache,1M External Cache, and can realize high performance computing capabilities of 12000DMIPS (1.2 billion instructions set per second). It is with built-in 3D graphics acceleration engine, 2D graphics acceleration, supports 4096*4096 pixels resolution maximally. Video encoding supports MPEG-4/H.263/H.264 achieving 1080p @ 30fps; video decoding MPEG2/VC1/Xvid achieves 1080p @ 30fps. And it also supports HD HDMI TV output.

I.MX6Q is a kind of high-performance, low-power processor. It's applied to making Handheld Electronic Equipment, Communication Equipment, Medical Application Equipment, Learning Machine, Notebook, Video Surveillance Equipment and a variety of Man-machine interface, etc., such as High-definition Games, Wireless GPS Navigation, Mobile Video Playback, Intelligent Control, Instrumentation, Navigation Devices, PDA Devices, Remote Monitoring, Game Development and so on.

E9 mini PC is adopted the powerful I.MX6Q chip and lead out most of I.MX6Q functional interfaces, there are various kinds of application interfaces on the board to meet the basic requirements of learning, developing. Just connect the corresponding modules to realize the function you want. With the E9 mini PC, you can create a variety of DIY products as long as having new fresh ideas.

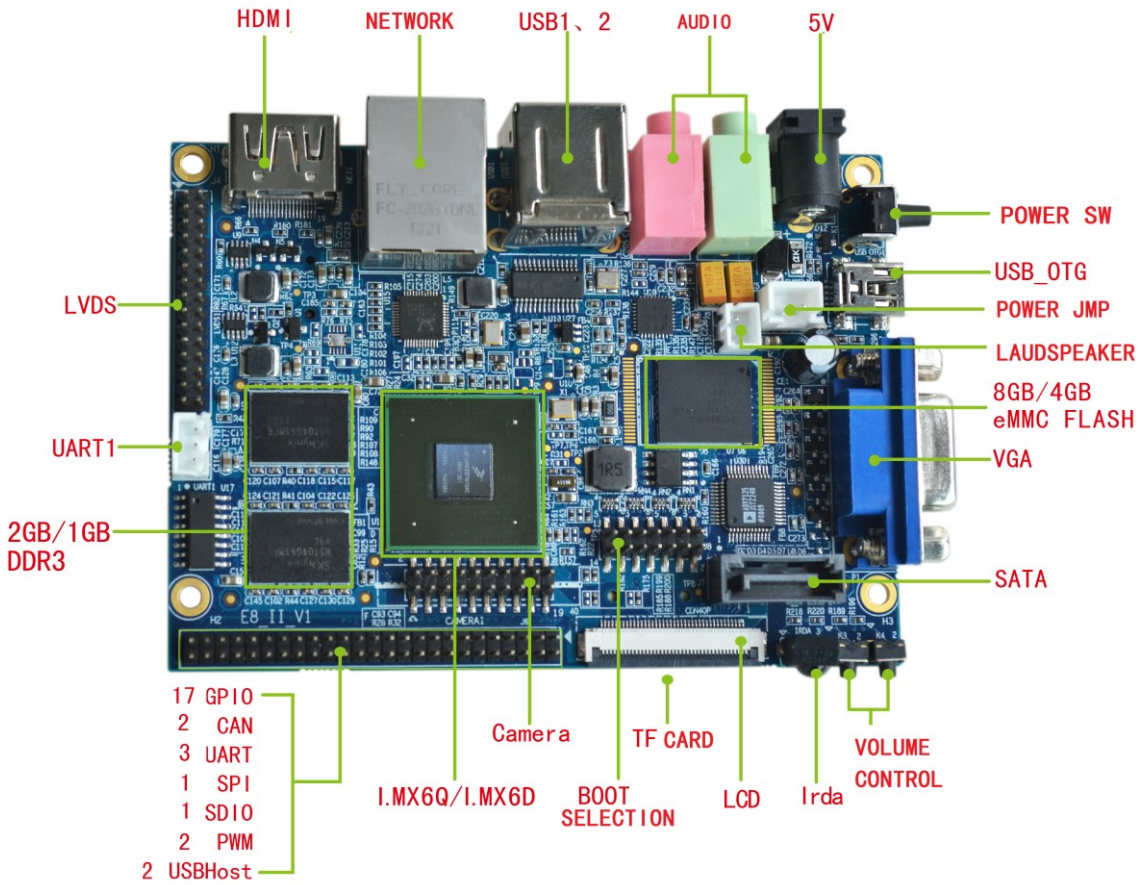
E9 mini PC's best advantages on software is that it supports multi-systems switching when start up, it's similar to PC boot that you can choose different system to boot. Currently, we provide Ubuntu12.04 and Android 4.2.

Using this board with provided software library and finished product, you can realize the functions and meet your needs easily. As long as you have E9, you have a Quad hand-held mini-PC. Through it, you can realize:

- Simple All-in-one Computer
- Auto PC
- Smart Set-top Box
- Cloud Terminal Equipment

- Advertising Multimedia Terminal Equipment
- ……..

1.2 E9 diagram



The board offers a variety of peripherals function as follows:

- **5V** (Note: To prevent burning the board, please confirm whether it's 5V power supply before connection.)
- **POWER SW**
- **POWER JMP** (Skip the power management circuit and directly to control the board's power on or off.)
- **BOOT SELECTION** (For boot mode selection)
- **UART1** (Debug serial port interface)
- **USB OTG** (For image download)
- **USB1/2** (For connecting U disk, USB mouse & keyboard, USB camera and other USB equipment.)
- **SATA** (Hard disk interface, external hard disk power supply is required)
- **TF CARD** (TF card interface)
- **NETWORK** (100/1000M Ethernet interface)
- **HDMI** (HDMI video interface)
- **LCD** (40Pin 0.5mm pitch FPC interface)

- **AUDIO INPUT/OUTPUT** (Audio Input Output interface)
- **CAMERA** (CMOS camera interface)
- **VGA** (1-channel)
- **LVDS** (Single/Dual-channel LVDS, External backlight power supply is required.)
- **LRDA** (infrared remote control interface)
- **RTC** (on the backside of the board)
- **Expansion interface**

Chapter Two E9 Usage Instructions

2.1 E9 Data Introductions

2.1.1 E9 Software Tools Instructions

The driver or software need to be installed in the PC when using E9 board includes:

- Serial interactive software: Double click to open the putty.exe software, for more information in using, please refer to Chapter 3.4.
- USB Programming Software: Uncompress "Mfgtools-Rel-1.1.0_121218_MX6Q_UPDATER.zip", and follow the tips in Chapter 6.2 and 9.2 to burn the system.

2.1.2 E9 Image Instruction

All the path and Image in the following form can be founded in the "Image" directory.

Android image

Image	Instructions
Android u-boot	u-boot_E9_for_android_v2.0.bin
Android kernel	boot_E9_for_android_V2.0.img
Android Recovery	recovery_E9_for_android_v2.0.img
Android File System	system.img_E9_for_android_v2.0.img

Ubuntu image

Image	Instructions
u-boot	u-boot_E9_for-ubuntu_v2.0.bin
Ubuntu Kernel	uImage_E9_for_ubuntu_v2.0
Ubuntu Rootfs	oneiric_E9_for_ubuntu_v1.0

Linux image

Image	Instructions
u-boot	u-boot_E9_for_linux_v2.0.bin
Linux Kernel	uImage_E9_for_linux_v2.0
Linux File System	rootfs_E9_for_linux_v2.0.tar.gz

2.1.3 E9 Ubuntu Source Code Instruction

The path and file name in the following form can be founded in the "Source Code\Ubuntu Source Code" directory.

File Name	Instructions
uboot	uboot_E9_1.3.4_for_ubuntu_v2.0.tar.bz2
kernel	linux_E9_3.0.35_for_ubuntu_v2.0.tar.bz2
Ubuntu ubuntu file system	oneiric_E9_12_04_for_ubuntu_v2.0.tgz

2.1.4 E9 Android Source Code Instruction

The path and file name in the following form can be founded in “Source Code\Android Source Code" directory.

File Name	Instructions
Android4.2	TQIMX6_Android4.2.2_v2.0.tar.bz2

Note: The Android4.2 includes Source Code for kernel, uboot, and Android File system.

2.1.5 E9 linux Source Code Instruction

The path and file name in the following form can be founded in “Source Code/Linux Source Code” directory

File Name	Instructions
uboot	uboot_E9_1.3.4_for_linux_v2.0.tar.bz2
Linux Kernel	linux_E9_3.0.35_for_linux_v2.0.tar.bz2
Linux QT File System	rootfs_E9_12_04_for_linux_v2.0.tgz

2.1.6 E9 Other Data Instruction

File Name	Instructions
Cross Compiler	\Software\Linux tools\gcc-4.6.2-glibc-2.13-linaro.tar.bz2
Schematics	Schematics
Linux tools	ubuntu_env_install.tar.bz2
Chip_PDF	

2.2 E9 Software Introductions

2.2.1 u-boot Introduction

The uboot_2009 supports screen and launch parameters settings.

Function	特点
SD update the firmware automatically	No need to connect with PC, more quickly for updating
Supports LCD switch manually	No need to change u-boot
Boot System	Supports kernel dynamically adjust the LCD

Note:

1. For the setting of the DIP switch, please refer to Chapter 4

2. For the detail u-boot usage instructions, please refer to relevant chapters about E9 firmware update.

2.2.2 Kernel Introductions

The Linux Kernel provided includes:

1. Display:

- TFT interface: 7inch/5inch (800*480@60) RGB24, Backlight adjustable;
- LVDS interface

Single: 2 channels, Supports 1366*768@60, RGB666, it can display only one single way, and also two-way simultaneous display (two shows for the same content)

Dual: Supports 1920*1080@60, RGB666, Backlight adjustable;

- VGA interface: Supports 1024*768@70, 1280*768@60, 1360*768@60, RGB24。
- HDMI: Supports 1920*1080P, Audio and video output at the same time, support hot-pluggable devices

2. Bus interface:

- Network: Cable Network：100/1000M self adaption, data loss rate is less than 0.1% on the condition of stable network and it supports most of the network protocol. Users can configure the kernel if it is needed.
- USB interface:

Hub: Can connect the U disk (16G at the maximum), USB Mouse Keyboard, IR touch and so on. USB transmission rate reach about 9.5M/s, and write at 7.5M/s.

OTG: At present it can be used as device only. The actual speed is about 1280KB/s, testing by adb tool.

- I2C: At present the transmission speed is about 100K/S.
- Serial port：Baud rate commonly can be set as 9600, 38400, 115200, 230400.

3. Multi-media:

- Video:

Support various decoding/encoding format in hardware, reaching 1080P 30fps in decoding (H.264, VC1, RV10, DivX and so on), and reaching 108030fps in encoding H.264.

Graphics Processing Unit (GPU3Dv4), Graphics processing in accordance with the following points

• OpenGL ES 1.1 and 2.0 including extensions.

• OpenVG 1.1

• Windows Direct3D

• OpenCL EP

• Graphics Processing

- Audio:

Frequency: 8 K - 192 KHz, Supported formats: MP3, wav, ogg, arm, aac, flac and so on, 2 channels to output (HP and SPK), which can switch automatically.

- Camera interface:

4. Storage：

- U disk: 16G at the maximum, according to the test, it can read at the speed 9 M/S and write at 7M/Ss.
- Micro TF Card: 32G at the maximum, according to the test, it can read at the speed 7M/S, write at 4M/S
- SATA: 1T at the maximum, according to the test, it can read at the speed 10.5M/S and write at 8M/S.

It is suggested to use the ntfs format; though it also supports fat and ext4, the fat and ext4 can only be used for terminal access (Android Platform and ubuntu system are not limited by the format.)

5. Others:

- IrDA: It supports the Infrared Remote Controller directly. If remote controller needs to be changed, then it needs to revise the remote controller driver.
- Keypad: It supports volume and other functions under the Android.
- RTC: To save the real-time clock
- Expansion port (can, spi, uart and so on): It can use the CAN bus, SPI bus or UART bus to connect the corresponding peripheral interface when it is needed.

Chapter Three Development Environment Set Up

3.1 Ubuntu Platform Environment Set Up

Use the libraries for Script and gcc compiler provided, and execute the command in the PC/opt directory: #mkdir Embedsky

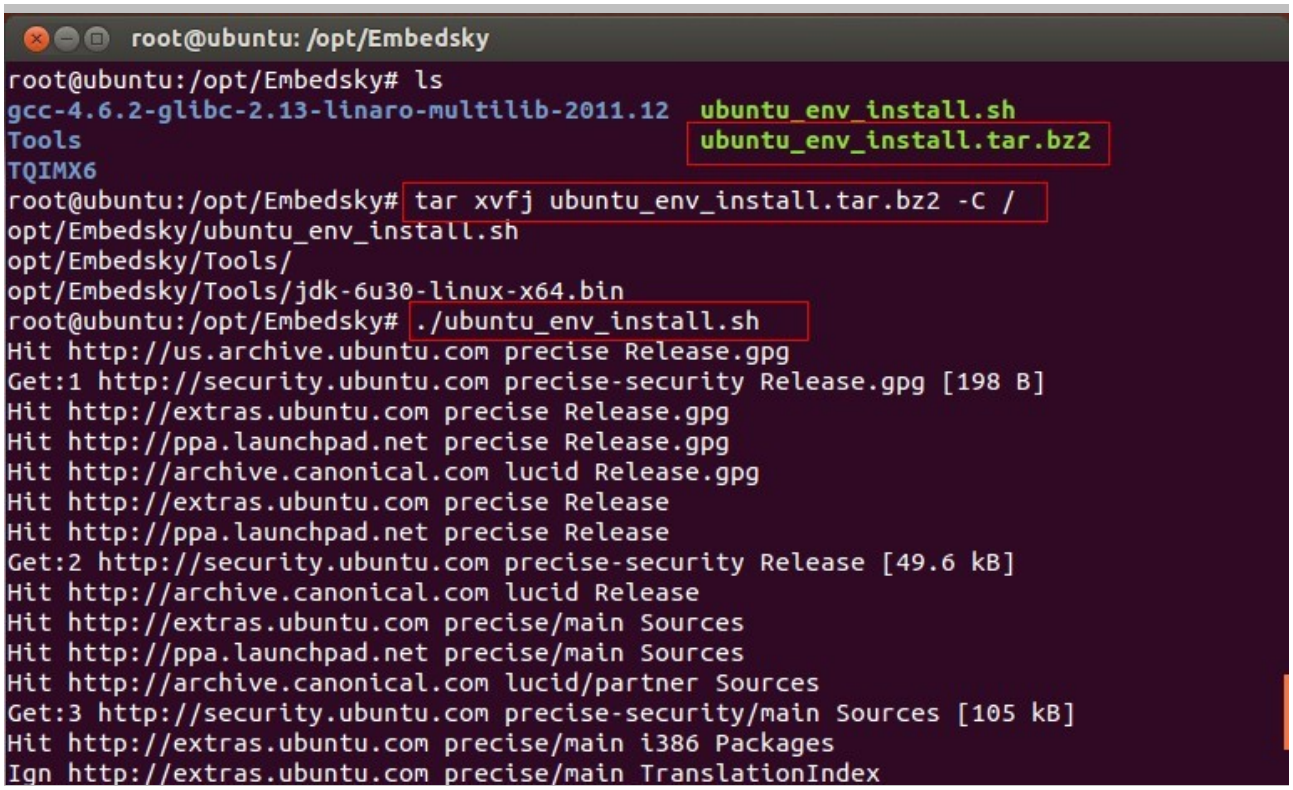
Create a new directory, and copy the Software\Linux tools\ubuntu_env_install.tar.bz2" to Ubuntu /opt/Embedsky directory, then use the following command in the terminal:

#cd /opt/Embedsky

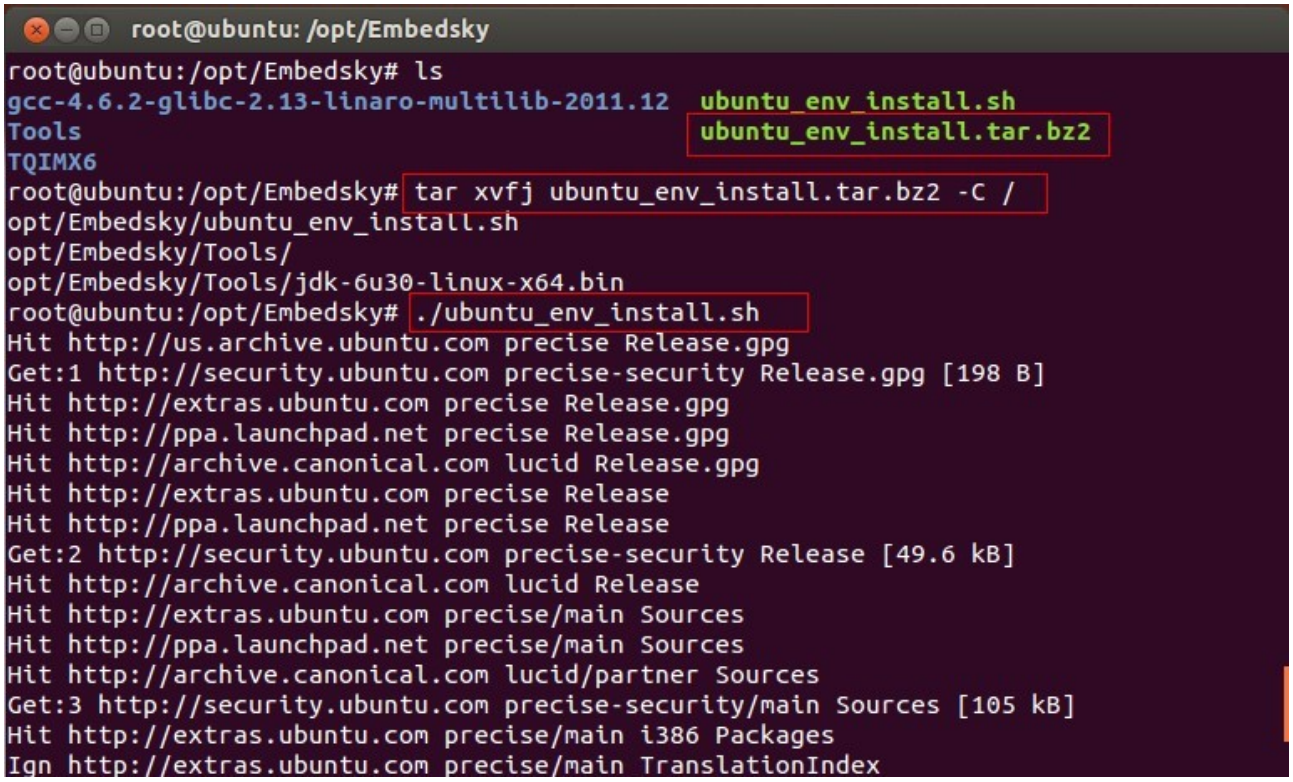
#tar xvfj ubuntu_env_install.tar.bz2 -C /

Extract the zip file to a local directory opt/Embedsky, and named: ubuntu_env_install.sh, then use the following command in the terminal: #cd /opt/Embedsky; ./ubuntu_env_install.sh

Then it can automatically finish for installation for libraries and gcc

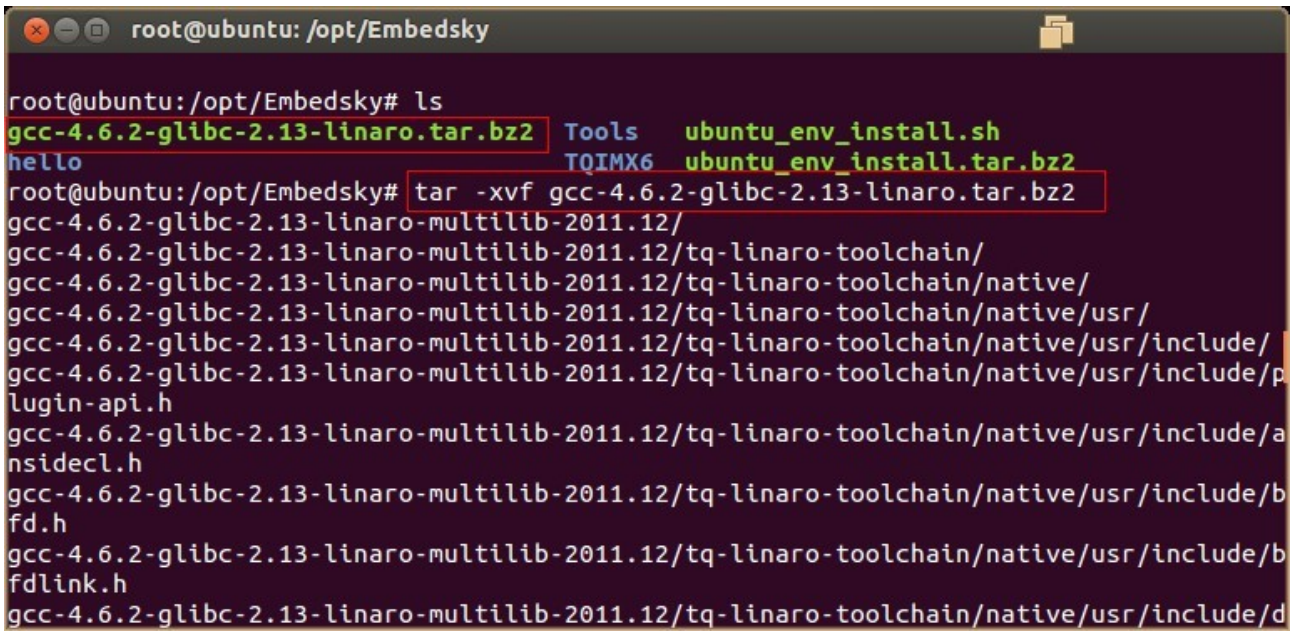


After installation, input the command in the terminal: gcc –v , then you can see the version of gcc.



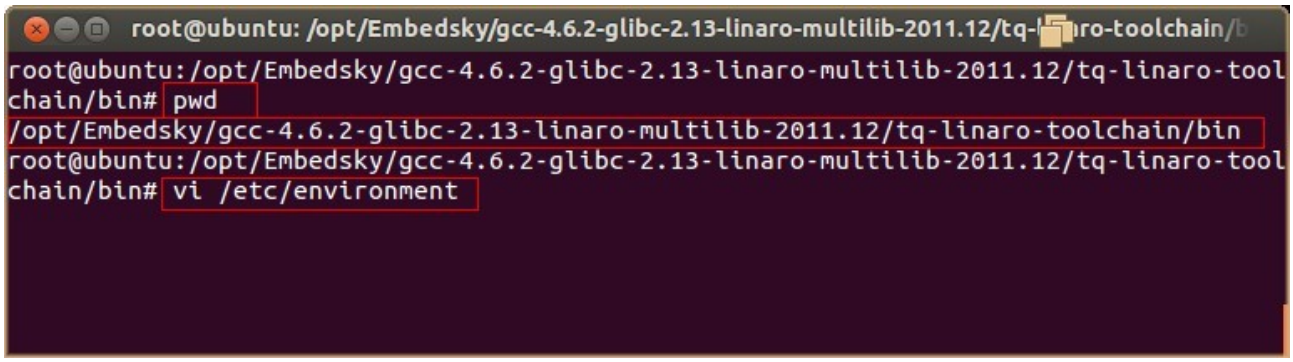
3.1.1 Ubuntu Compile Environment Set Up

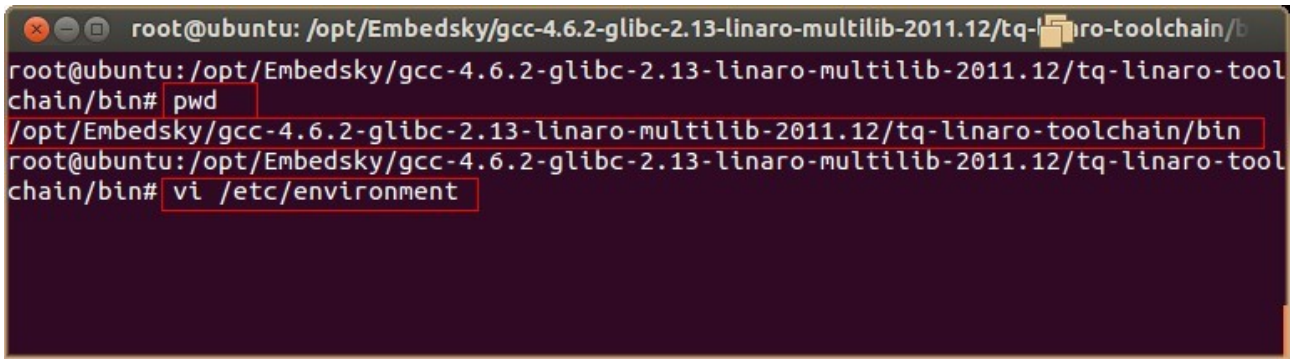
The cross-compiler provided is gcc-4.6.2 version; it is a kind of specifically cross compiler for Linaro. Copy "gcc-4.6.2-glibc-2.13-linaro.tar.bz2" compressed file to /opt/Embedsky root directory in the PC, then extract it in the terminal:



Modify the environment variable after decompression, and add the cross compiler path, then use the following command: #vi /etc/environment

After the modification, it is like the follows:

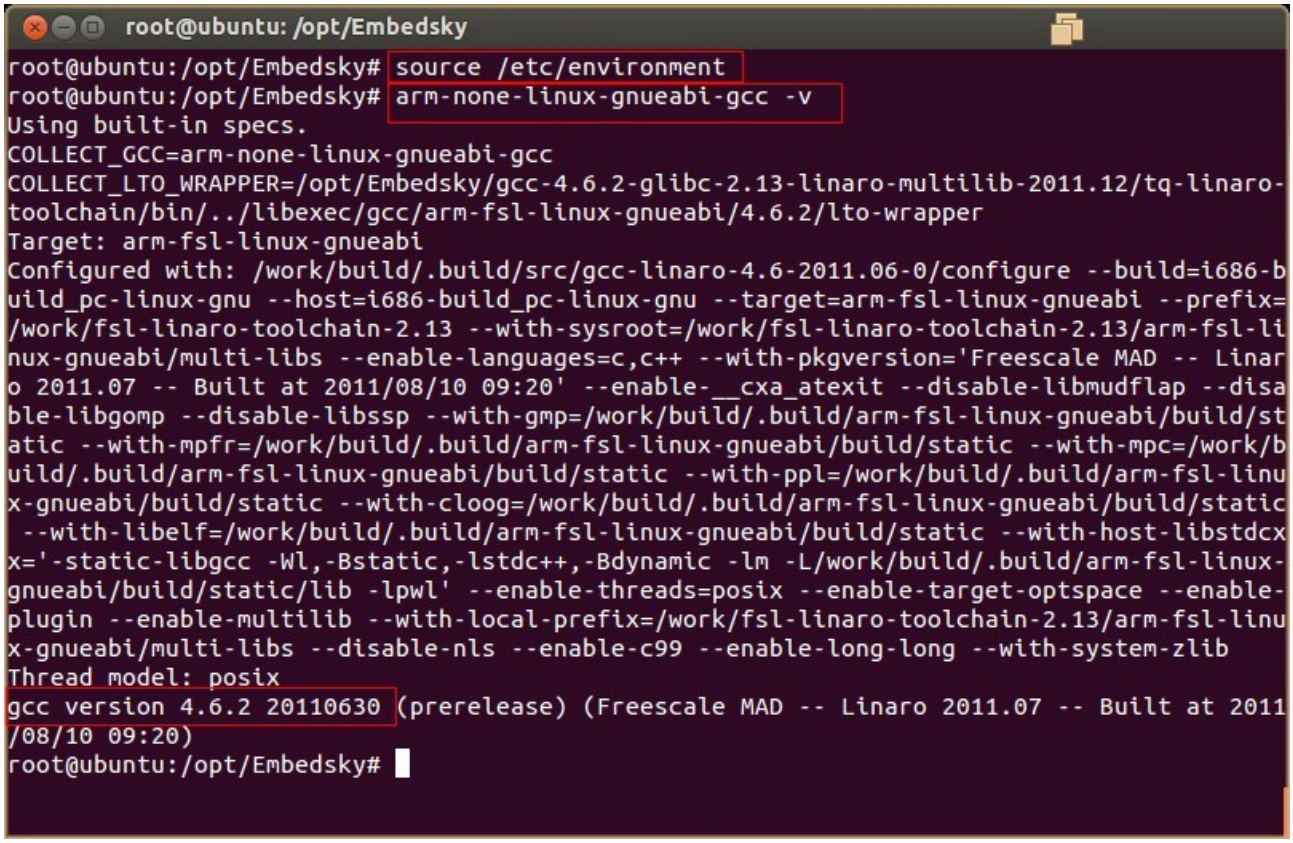




Execute the command: #source /etc/environment

And execute: #arm-none-linux-gnueabi-gcc -v

Then you can see the installed cross compiler:



3.1.2 Android Development Environment Set Up

Extract the java installation pack (jdk-6u30-linux-x64.bin) to /opt/Embedsky/Tools directory. For the convenience, use the command in terminal: mv jdk-6u30-linux-x64.bin /opt/Tools

Put the jdk-6u30-linux-x64.bin to /opt/Tools directory, and then customers can run it in terminal directly:

cd /opt/Tools/

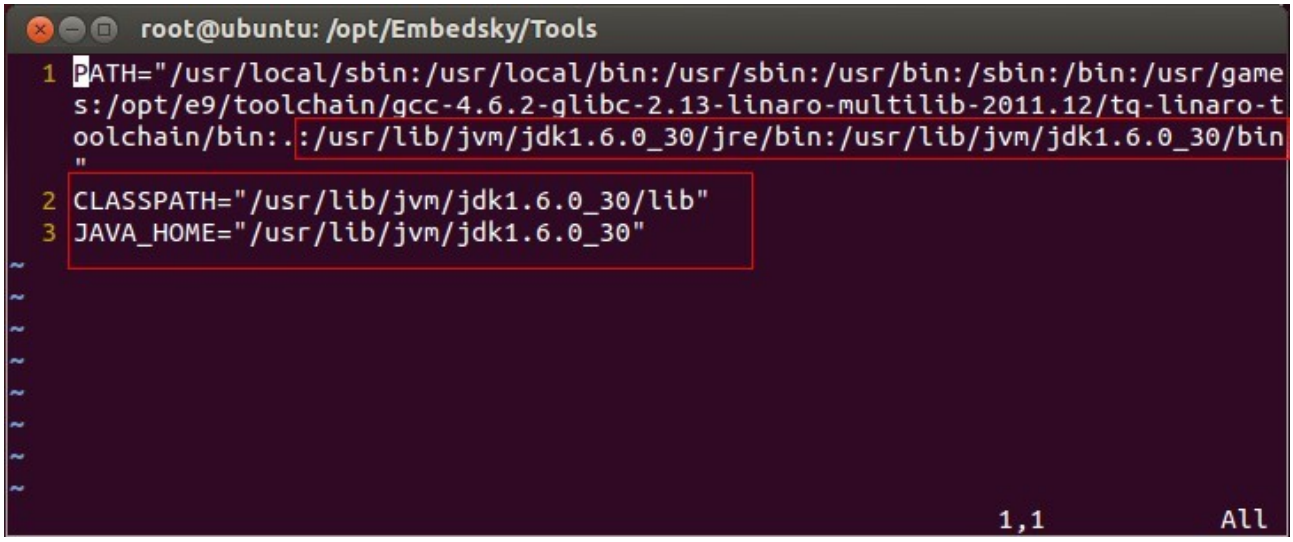
./jdk-6u30-linux-x64.bin, follow the tips to finish the installation.

The new directory will also be founded in the same directory (that is /opt/Tools/) after installation, and move it to the directory "/usr/lib/jvm/", then set the development environment variable:

```
root@ubuntu: /opt/Embedsky/Tools
root@ubuntu:/opt/Embedsky/Tools# ls
jdk-6u30-linux-x64.bin  mkimage
root@ubuntu:/opt/Embedsky/Tools# ./jdk-6u30-linux-x64.bin
Unpacking...
Checksumming...
Extracting...
UnZipSFX 5.50 of 17 February 2002, by Info-ZIP (Zip-Bugs@lists.wku.edu).
   creating: jdk1.6.0_30/
   creating: jdk1.6.0_30/db/
   creating: jdk1.6.0_30/db/bin/
  inflating: jdk1.6.0_30/db/bin/ij
  inflating: jdk1.6.0_30/db/bin/NetworkServerControl
  inflating: jdk1.6.0_30/db/bin/setNetworkClientCP.bat
  inflating: jdk1.6.0_30/db/bin/setEmbeddedCP.bat
  inflating: jdk1.6.0_30/db/bin/stopNetworkServer.bat
  inflating: jdk1.6.0_30/db/bin/NetworkServerControl.bat
  inflating: jdk1.6.0_30/db/bin/setNetworkServerCP
  inflating: jdk1.6.0_30/db/bin/sysinfo.bat
  inflating: jdk1.6.0_30/db/bin/derby_common.bat
```

```
root@ubuntu: /opt/Embedsky/Tools
* Access to early releases and documentation

Product and system data will be collected. If your configuration
supports a browser, the JDK Product Registration form will
be presented. If you do not register, none of this information
will be saved. You may also register your JDK later by
opening the register.html file (located in the JDK installation
directory) in a browser.

For more information on what data Registration collects and
how it is managed and used, see:
http://java.sun.com/javase/registration/JDKRegistrationPrivacy.html

Press Enter to continue.....



Done.
root@ubuntu:/opt/Embedsky/Tools#
root@ubuntu:/opt/Embedsky/Tools#
root@ubuntu:/opt/Embedsky/Tools#
```

```
root@ubuntu: /opt/Embedsky/Tools
root@ubuntu:/opt/Embedsky/Tools# ls
jdk1.6.0_30  jdk-6u30-linux-x64.bin  mkimage
root@ubuntu:/opt/Embedsky/Tools# mv -f jdk1.6.0_30/ /usr/lib/jvm/
root@ubuntu:/opt/Embedsky/Tools# vi /etc/environment
```

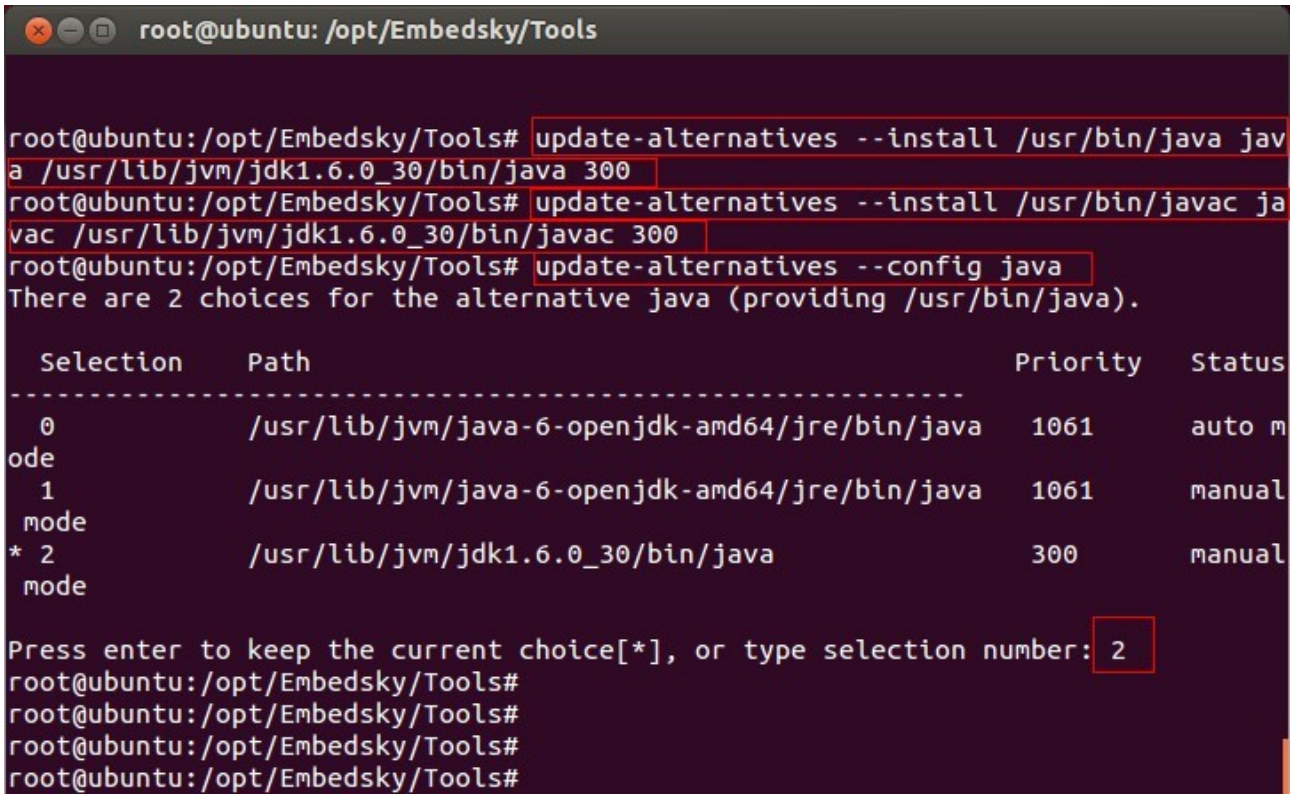


Input the command in terminal, and then configure the JDK:

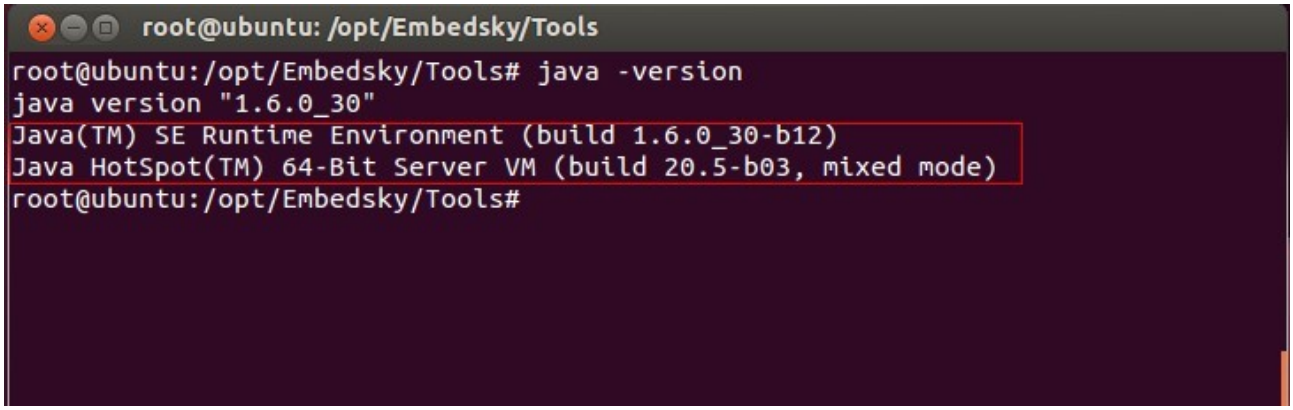
update-alternatives --install /usr/bin/java java /usr/lib/jvm/jdk1.6.0_30/bin/java 300

update-alternatives --install /usr/bin/javac javac /usr/lib/jvm/jdk1.6.0_30/bin/javac 300

update-alternatives --config java



Input the command in terminal java -version, then the JDK version.



As all the steps are finished, then Android development environment has been installed.

3.2 Ubuntu Tools Installation and configuration (minicom)

3.2.1 Install minicom

Usrs can install the minicom via network or using the Source Code contained in the CD directly.

- Network Installation: It needs to connect the internet, easy and suggested.

  Directly input the command in the terminal: #apt-get install minicom

- Source Code Installation:

  Minicom installation source code is in the “Software\Linux tools” directory.

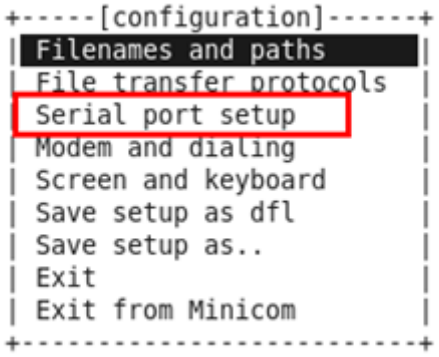
  Copy the minicom-2.5.tar.gz to PC Ubuntu system /opt/EmbedSky/ directory, and then extract it to PC Ubuntu, carry out the command #cd /opt/EmbedSky/, and then use the unpack command: #tar zxf minicom-2.5.tar.gz -C /opt/EmbedSky to /opt/EmbedSky directory, carry out the command #cd /opt/EmbedSky/minicom-2.5 to the source code directly unpacked just now.

  Enter the following commands in sequence to start the installation:

#./configure	configure source code
#make	compile source code
#make install	install minicom

3.2.2 Config minicom

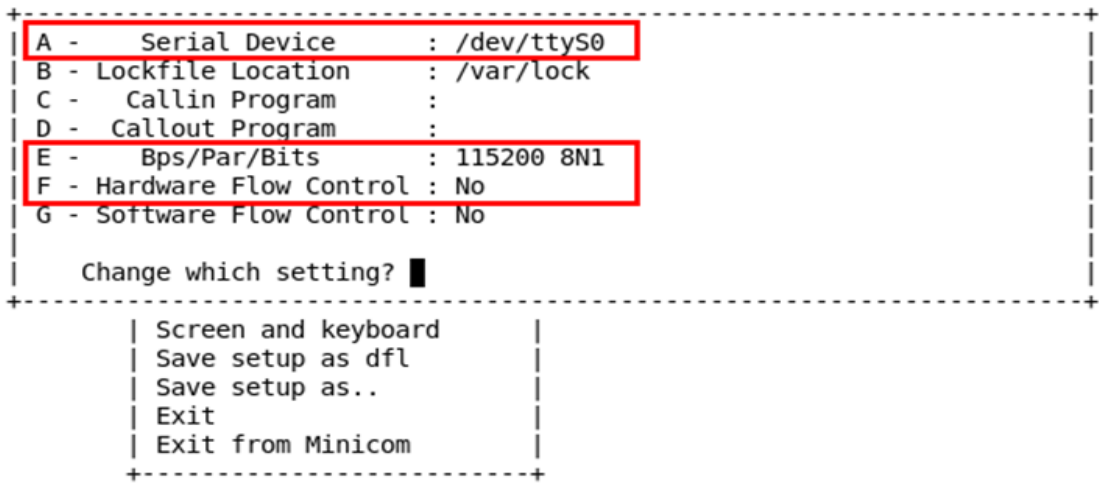
After install the minicom, input the command #minicom –s to start, then there will be a configuration list: Using the up and down key on the PC keyboard to choose serial port setup, then press Enter to enter a serial port configuration:



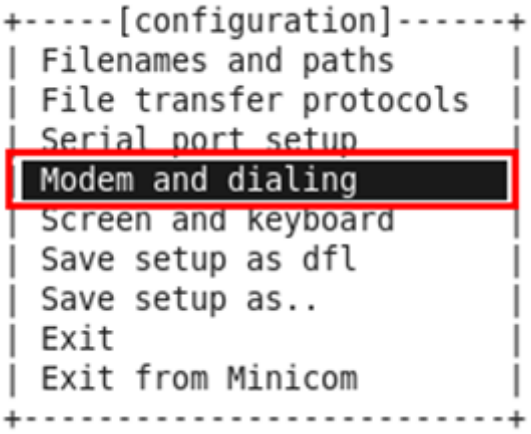
In the configuration page, input /dev/ttySn in the A Serial Device (Please according to the actual serial interface to enter the right number. n is for numbers, generally, it is 0), then press Enter; If it is the usb serial port equipment, then it should name is as /dev/ttyUSBn (n is for numbers, generally it is 0, but it needs to base on the actually device port);

In the E, Set the configuration speed as 115200 8N1

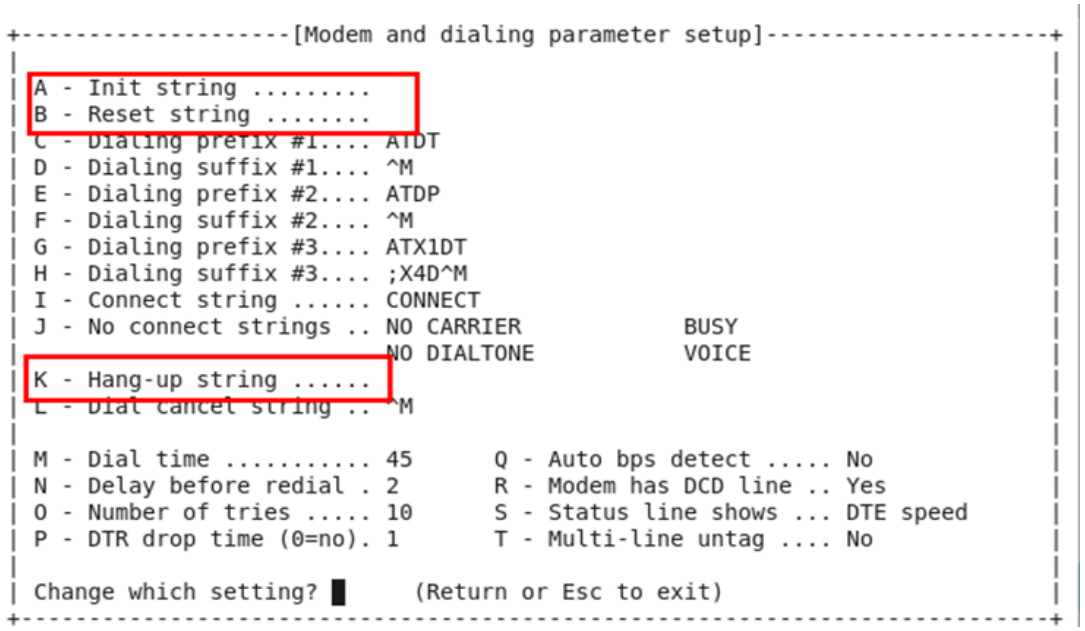
In the F, Set the Hardware Flow Control as NO, and then press Enter



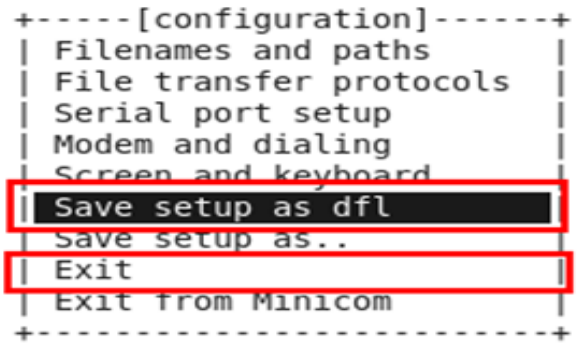
As the minicom will be used as super terminal control devices such as router, rather than controlling the modem, so it needs to modify it as modem and dialing as follows:



Set the Init string, Reset string, Hang-up string in the Modem and dialing configuring list as NULL:



Step6. After setting, please choose “save setup as dfl” in configuring list to save the current settings as the default settings. (It must do this step, otherwise it needs to reconfigure next time you start minicom). Choose Exit to get out of the configuration page and enter minicom.



When you want to start the minicom next time, input minicon in the terminal and then press Enter, then you 后回 can enter it directly. The command minicom is for enter the serial port super terminal page, while minicom –s is the command for config minicom. Minicom is based on the window page. If you want the function pop-up window, then you can press ctrl-a, then press the functional key (a-z or A-Z), first press ctrl-a, and then z, then there will be a help window, provides all of the command description.

Note 1: If close minicom improperly, there will be several LCK* files in the /var/lock, these files may preven minicom running, you can delete them then the minicom will return to work again.

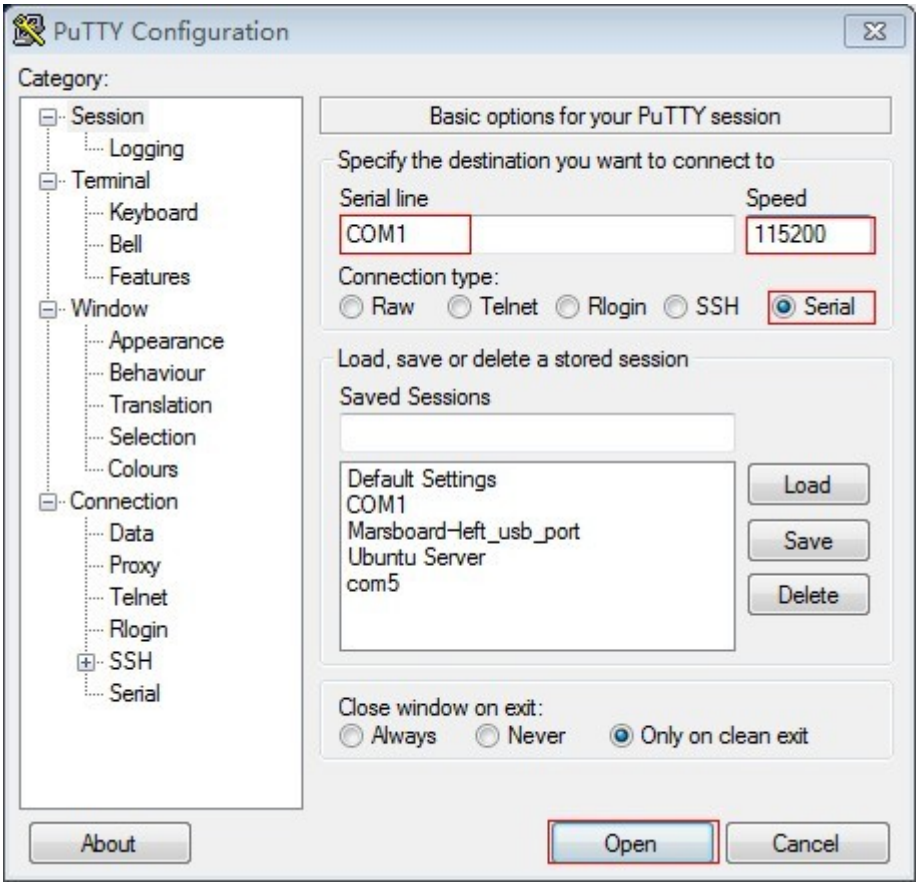
Note 2: If you use minicom in a virtual device, you need to add the serial port equipment to the virtual device before start the Ubuntu in the virtual device, and then start the Ubuntu. And also one point need to note that when Windows uses in the serial port devices (such as SecureCRT software, and has already used COM3 serial port), and if the virtual device is also COM3, then after started Ubuntu in the virtual device, the serial port can't be used in the minicom. Only close the SecureCRT software, release the COM3, then you can use the serial port in Ubuntu.

Note 3: Input sudo minicom -D /dev/ttyUSB0, then it can specify USB serial port device directly.

3.3 Windows Platform Environment Set Up

As the Windows platform is used for testing only, it is not used for developing, so the setup process is simple, just need to install a terminal serial port.

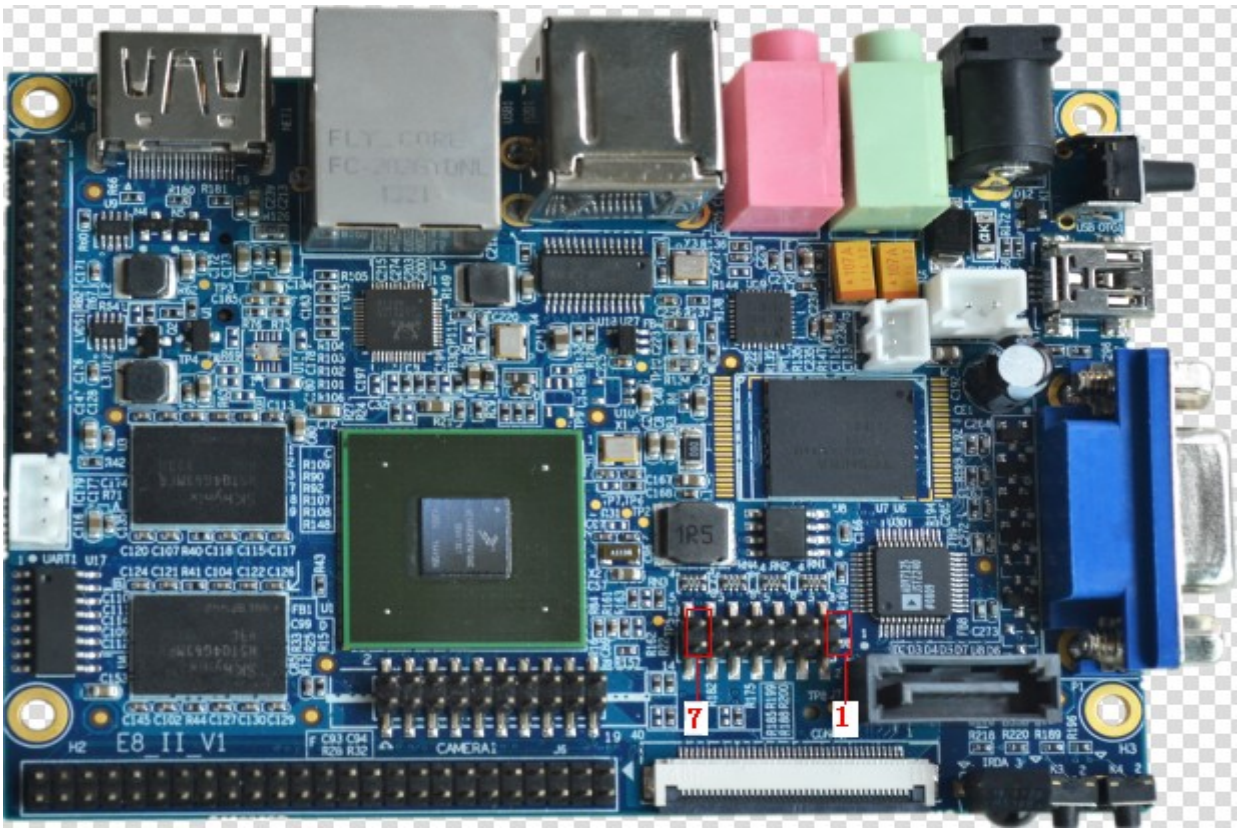
Double-click to open the software PuTTY, choose Serial in the Connection type, and set the Serial line & Speed, then click “Open”:



Chapter Four Uboot Use

4.1 Interpretation of the Boot Jumper Cap

J8 is the interface of choice of system boot mode for E9 mini PC. Short different jumper cap will have different boot modes. It is 2.00mm pitch and 10Pin interface and it needs to work with 2.00mm pitch jumper cap.



E9 Mini PC keeps all the boot modes, and it uses the following 3 boot modes temporarily. (In the above picture, the pins, from right to left, are 1, 2, 3, 4, 5, 6, 7)

Boot select tablet							
Boot device	**7**	**6**	**5**	**4**	**3**	**2**	**1**
	BT_CFG2_3	BT_CFG2_4	BT_CFG2_5	BT_CFG2_6	BT_CFG1_4	BT_CFG1_5	BT_CFG1_6
Emmc	1	1	1	X	X	1	1
SD	1	0	0	X	X	0	1
Dwonload	0	0	0	0	1	1	0

Note:

1. If the Ethernet upwards, then the J8 interface, from left to right, the pins are 7, 6, 5, 4, 3, 2, 1, So corresponding to the above table, short the Jumper Cap is marked as “1”;” X” means it can connect and it can also disconnect.
2. Please connect the development board and PC via serial cable before switch on the device, and then open the corresponding serial port terminal in the PC. (It suggests using the minicom in linux, and sourceCRT in

windows), for more details, please check the chapter 3.3, 3.4

4.2 Boot the E9 mini PC

According to different jumper cap connections, it supports to boot the E9 mini PC from tf card or from emmc

Connect the 5V power supply and the serial interface, then long press the power switch (the trigger power switch next to the 5V jack), then release. The default system is android, and LCD for display is 800*480. Users can press any keys in the PC to interrupt the uboot automatically and enter into the Uboot settings menu:

```
#####      Boot for E9 Main Menu          #####
[1]boot from emmc
[2]boot from sdcard
[3]boot from tftp
[5]download from sdcard
[0]setting boot args
[q]exit to command mode
Enter your selection:
```

[1] Boot from emmc, it can boot the corresponding system according to the setting, and the default system is android.

[2] Boot from sd card

[3] Boot from tftp. It needs to use tftp and nfs, and only the android system can be booted from tftp

[5] Burning system from sd card

[0] Setting boot args

[q] Exit to uboot command mode

4.3 Resolution Settings

The exact operation as follows, enter uboot menu, then:

0: Enter the parameter settings menu

3: Enter the LCD settings menu

X: According to the actual situation, 0 is for display android and ubuntu independently; 1 is for android HDMI+LVDS or HDMI+FPC display, and 3 is for ubuntu LVDS+LVDS+HDMI display. For more details, please refer to the testing chapter.

Y: Choose the resolution according to the actual situation. FPC->VGA indicates FPC interface to VGA output, so the actual output interface is VGA, LVDS->VGA is the same.

For example: LVDS: 1366*768 display

```
#####      Boot for IMX6 Main Menu          #####
[1]boot from emmc
```

```
[2]boot from sdcard
[3]boot from tftp
[5]download from sdcard
[0]setting boot args                    // Enter uboot launch parameters settings menu
[q]exit to command mode
Enter your selection: 0

##### param setting #####
[1]setting nfs args
[2]setting tftp args
[3]setting display args                 // Setting the related parameters for display
[4]setting default boot
[s]save setting
[q]quit
Enter your selection: 3

##### display select Menu #####
[1] 1 screen for android/ubuntu/linux.          //Single display for android or linux or ubuntu
[2] 2 screen for android                        //android double-screen display, HDMI and FPC or LVDS
[3] 3 screen for ubuntu.                        //ubuntu 3 screen display, HDMI and FPC and LVDS
[s] save setting
[q] quit
Enter your selection: 1

##### display select device #####
[1] lcd or VGA.
[2] lvds or VGA.                                //LVSD or LVDS to VGA
[3] hdmi.
[4] quit
Enter your selection: 2

##### display select LVDS #####
[1] 800*600 for LVDS->VGA
[2] 1024*768 for LVDS->VGA
[3] 1280*768 for LVDS->VGA
[4] 1366*768 for LVDS                           //LVDS at 1366*768 display
[5] 1920*1080 for LVDS (2 lvds is needed,LVDS only)     //LVDS at 1920*1080 needs 2 channels LVDS output
   at the same time.
[q] quit
Enter your selection: 4

##### display select Menu #####
[1] 1 screen for android/ubuntu/linux.
[2] 2 screen for android.
[3] 3 screen for ubuntu.
[s] save setting
[q] quit
Enter your selection: s
Saving Environment to MMC...
Writing to MMC(3)... done
```

```
##### display select Menu #####
[1] 1 screen for android/ubuntu/linux.
[2] 2 screen for android.
....
```

4.4 The Default Boot System Settings

The exact operation as follows, enter uboot menu, then:

0: Enter the parameter Settings menu

4: Enter the operated Settings menu

X: It is set according to the actual system needed, 0 indicates the default boot mode is ubuntu; 1 indicates the default boot mode is android; 2 indicates the default boot mode is using the tftp and nfs to boot android;

S: Save the setting for the future boot.

System will use the default mode here next time when booting.

```
#####       Boot for E9 Main Menu            #####
[1]boot from emmc
[2]boot from sdcard
[3]boot from tftp
[5]download from sdcard
[0]setting boot args            // Enter uboot launch parameters Settings menu
[q]exit to command mode
Enter your selection: 0

##### param setting #####
[1]setting nfs args
[2]setting tftp args
[3]setting display args
[4]setting default boot         // Set the default boot mode
[s]save setting
[q]quit
Enter your selection: 4

##### param setting #####
[1]ubuntu used emmc             // Set booting the Android from EMMC by default
[2]android used emmc            // Set booting the Ubuntu from EMMC by default
[3]linux used emmc              // Set booting the Linux from EMMC by default
[4]android used tftp & nfs      //Set booting the Android from TFTP and NFS by default
[s]save setting
[q]quit
Enter your selection: 1

##### param setting #####
[1]ubuntu used emmc
[2]android used emmc
[3]android used tftp & nfs
[s]save setting                 //The parameters are not saved by default, these users who want to save the parameters, please choose “save settings”
[q]quit
Enter your selection: s
```

```
Saving Environment to MMC...
Writing to MMC(3)... done

##### param setting #####
[1]ubuntu
[2]android
.....
```

Ubuntu

Introduction

Note: The "#" in front of following commands is the user permissions for using the root. It is recommended to use root user when login Linux in PC; if it is not login root user, then please use the sudo commands to get the root user permission, otherwise it may fail to compile.

Login Ubuntu:

ID: **linaro**

Password: **linaro**

Chapter Five Ubuntu Source Code & Compile

5.1 Ubuntu Source Code Drivers

Ubuntu uses 3.0.35 version kernel. And all the paths below are relative to the root directory of the kernel. Due to the driver code is involved, it is difficult to introduce one by one, so the following mentioned driver code are for the main drivers or those changing frequently only.

drivers	paths	File name
emmc/sdcard driver	drivers/mmc/host: sdhci-esdhc-imx.c sdhci.c sdhci-platform.c sdhci-pltfm.c	/dev/mmcblk*
LCD & LVDS driver	drivers/video/mxc/: ldb.c mxc_dispdrv.c mxc_dvi.c mxc_ipuv3_fb.c mxc_lcdif.c	/dev/fb*
Capacitive touch screen driver	drivers/input/touchscreen： gt811.c	dev/input/event*
Resistive touch screen driver	drivers/input/touchscreen： tsc2007.c	dev/input/event*
USB OTG driver	drivers/usb/otg： fsl_otg.c	
Cable network driver	drivers/net： fec.c	eth0
Serial Driver	drivers/tty/serial： imx.c mxc_uart_early.c serial_core.c	/dev/ttySAC*
WM8960 Audio Driver	sound/soc/imx/*	/dev/snd/*
I2C driver	drivers/i2c/busses: i2c-imx.c	
Keypad driver	drivers/input/keyboard： gpio_keys.c	dev/input/event*

Irda Driver	drivers/input/keyboard： tq_hs0038.c	dev/input/event*
Backlight driver	drivers/video/backlight/ ： pwm_bl.c backlight.c	
pwm driver	arch/arm/plat-mxc/ ： pwm.c	
RTC driver		/dev/rtc0
watchdog driver	drivers/watchdog： imx2_wdt.c	/dev/watchdog
Multimedia video camera driver	drivers/media/video/mxc/capture/*	/dev/video*
sata driver	drivers/ata： ahci_platform.c libahci.c	/dev/sdx

5.2 Compile Ubuntu Source Code

5.2.1 Compile Bootloader

➢ **Extract U-boot Source Code**

Extract u-boot source code (“Source Code\Ubuntu Source Code\uboot”, named uboot_E9_1.3.4_for_ubuntu.tar.bz2) and copy it to ubuntu system root directory. (Here the writer of this manual choose root directory to operate and take it a screenshots, actually, any directory will be OK), then use the command: #tar xvfj uboot_E9_1.3.4_for_Ubuntu.tar.bz2 -C /

Extract the source code, as shown in the figure below:

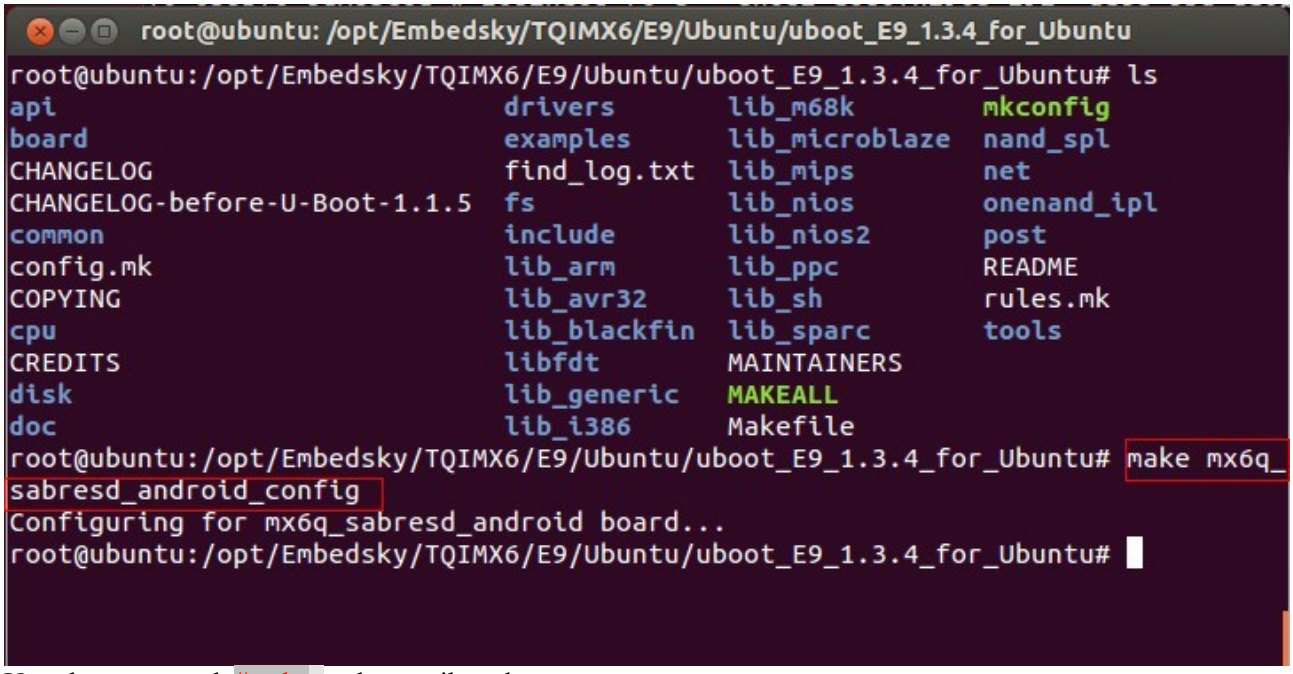
```
root@ubuntu: /opt/Embedsky
root@ubuntu:/opt/Embedsky# ls
Tools    uboot_E9_1.3.4_for_Ubuntu_v2.0.tar.bz2  ubuntu_env_install.tar.bz2
TQIMX6   ubuntu_env_install.sh
root@ubuntu:/opt/Embedsky# tar xvfj uboot_E9_1.3.4_for_Ubuntu_v2.0.tar.bz2 -C /
opt/Embedsky/TQIMX6/E9/Ubuntu/uboot_E9_1.3.4_for_Ubuntu/
opt/Embedsky/TQIMX6/E9/Ubuntu/uboot_E9_1.3.4_for_Ubuntu/fs/
opt/Embedsky/TQIMX6/E9/Ubuntu/uboot_E9_1.3.4_for_Ubuntu/fs/reiserfs/
opt/Embedsky/TQIMX6/E9/Ubuntu/uboot_E9_1.3.4_for_Ubuntu/fs/reiserfs/reiserfs.c
opt/Embedsky/TQIMX6/E9/Ubuntu/uboot_E9_1.3.4_for_Ubuntu/fs/reiserfs/Makefile
opt/Embedsky/TQIMX6/E9/Ubuntu/uboot_E9_1.3.4_for_Ubuntu/fs/reiserfs/mode_string.
c
opt/Embedsky/TQIMX6/E9/Ubuntu/uboot_E9_1.3.4_for_Ubuntu/fs/reiserfs/reiserfs_pri
vate.h
opt/Embedsky/TQIMX6/E9/Ubuntu/uboot_E9_1.3.4_for_Ubuntu/fs/reiserfs/dev.c
opt/Embedsky/TQIMX6/E9/Ubuntu/uboot_E9_1.3.4_for_Ubuntu/fs/cramfs/
opt/Embedsky/TQIMX6/E9/Ubuntu/uboot_E9_1.3.4_for_Ubuntu/fs/cramfs/cramfs.c
opt/Embedsky/TQIMX6/E9/Ubuntu/uboot_E9_1.3.4_for_Ubuntu/fs/cramfs/Makefile
opt/Embedsky/TQIMX6/E9/Ubuntu/uboot_E9_1.3.4_for_Ubuntu/fs/cramfs/uncompress.c
opt/Embedsky/TQIMX6/E9/Ubuntu/uboot_E9_1.3.4_for_Ubuntu/fs/fat/
```

```
root@ubuntu: /opt/Embedsky/TQIMX6/E9/Ubuntu/uboot_E9_1.3.4_for_Ubuntu
root@ubuntu:/opt/Embedsky# ls
Tools    uboot_E9_1.3.4_for_Ubuntu_v2.0.tar.bz2  ubuntu_env_install.tar.bz2
TQIMX6   ubuntu_env_install.sh
root@ubuntu:/opt/Embedsky# cd TQIMX6/E9/Ubuntu/uboot_E9_1.3.4_for_Ubuntu/
root@ubuntu:/opt/Embedsky/TQIMX6/E9/Ubuntu/uboot_E9_1.3.4_for_Ubuntu# ls
api                          drivers       lib_m68k        mkconfig
board                        examples      lib_microblaze  nand_spl
CHANGELOG                    find_log.txt  lib_mips        net
CHANGELOG-before-U-Boot-1.1.5 fs           lib_nios        onenand_ipl
common                       include       lib_nios2       post
config.mk                    lib_arm       lib_ppc         README
COPYING                      lib_avr32     lib_sh          rules.mk
cpu                          lib_blackfin  lib_sparc       tools
CREDITS                      libfdt        MAINTAINERS
disk                         lib_generic   MAKEALL
doc                          lib_i386      Makefile
root@ubuntu:/opt/Embedsky/TQIMX6/E9/Ubuntu/uboot_E9_1.3.4_for_Ubuntu#
```

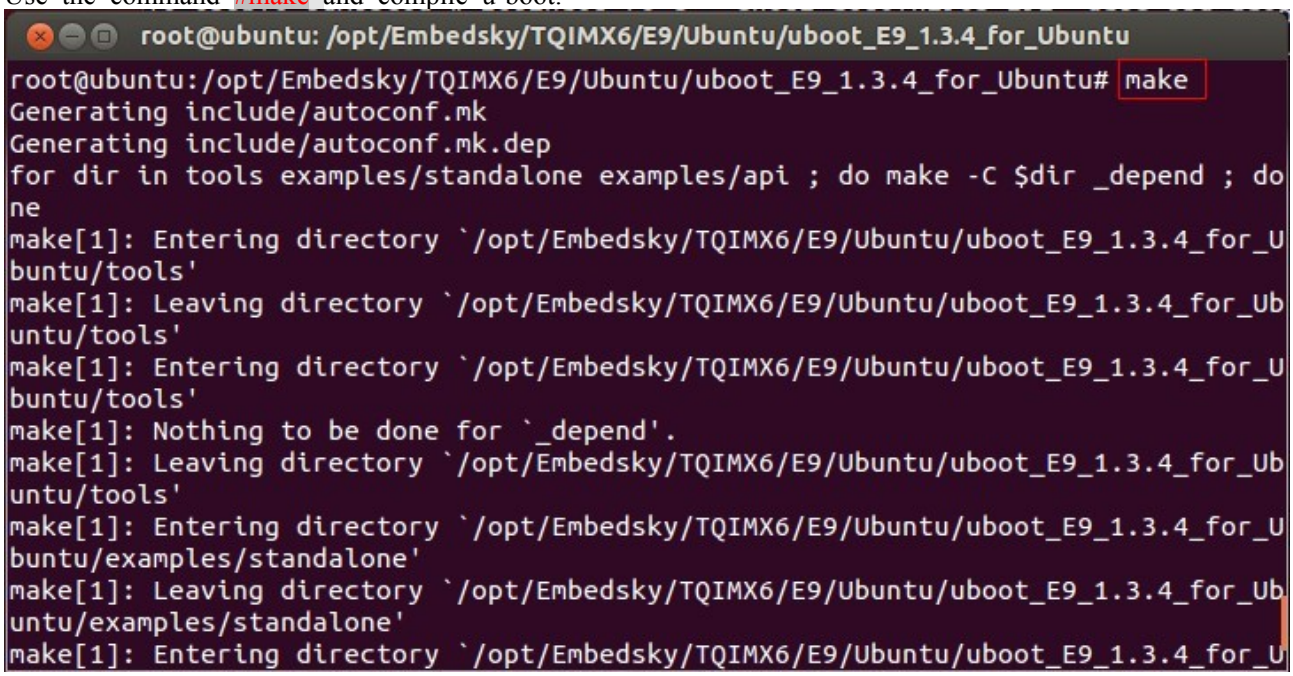
After extract, the source code can be founded in directory “/opt/Embedsky/TQIMX6/E9/Ubuntu”.

➢ U-boot Source Code Compile

After extract the source code, please use the command #make mx6q_sabresd_android_config and configure u-boot as shown below:



Use the command #make and compile u-boot.



```
root@ubuntu: /opt/Embedsky/TQIMX6/E9/Ubuntu/uboot_E9_1.3.4_for_Ubuntu
libubifs.a net/libnet.a disk/libdisk.a drivers/bios_emulator/libatibiosemu.a dri
vers/block/libblock.a drivers/dma/libdma.a drivers/fpga/libfpga.a drivers/gpio/l
ibgpio.a drivers/hwmon/libhwmon.a drivers/i2c/libi2c.a drivers/input/libinput.a
drivers/misc/libmisc.a drivers/mmc/libmmc.a drivers/mtd/libmtd.a drivers/mtd/nan
d/libnand.a drivers/mtd/onenand/libonenand.a drivers/mtd/ubi/libubi.a drivers/mt
d/spi/libspi_flash.a drivers/net/libnet.a drivers/net/phy/libphy.a drivers/net/s
k98lin/libsk98lin.a drivers/pci/libpci.a drivers/pcmcia/libpcmcia.a drivers/powe
r/libpower.a drivers/spi/libspi.a drivers/fastboot/libfastboot.a drivers/rtc/lib
rtc.a drivers/serial/libserial.a drivers/twserial/libtws.a drivers/usb/gadget/li
busb_gadget.a drivers/usb/host/libusb_host.a drivers/usb/musb/libusb_musb.a driv
ers/video/libvideo.a drivers/watchdog/libwatchdog.a common/libcommon.a libfdt/li
bfdt.a api/libapi.a post/libpost.a board/freescale/mx6q_sabresd/libmx6q_sabresd.
a --end-group /opt/Embedsky/TQIMX6/E9/Ubuntu/uboot_E9_1.3.4_for_Ubuntu/lib_arm/e
abi_compat.o -L /opt/e9/toolchain/gcc-4.6.2-glibc-2.13-linaro-multilib-2011.12/w
aveshare-linaro-toolchain/bin/../lib/gcc/arm-fsl-linux-gnueabi/4.6.2/default -lg
cc -Map u-boot.map -o u-boot
arm-linux-objcopy -O srec u-boot u-boot.srec
arm-linux-objcopy --gap-fill=0xff -O binary u-boot u-boot.bin
root@ubuntu:/opt/Embedsky/TQIMX6/E9/Ubuntu/uboot_E9_1.3.4_for_Ubuntu#
```

After compiled, you will find the image named “u-boot.bin” in the directory of /opt/Embedsky/TQIMX6/E9/Ubuntu/uboot_E9_1.3.4_for_Ubuntu, as the following two figures show:

```
root@ubuntu: /opt/Embedsky/TQIMX6/E9/Ubuntu/uboot_E9_1.3.4_for_Ubuntu
abi_compat.o -L /opt/e9/toolchain/gcc-4.6.2-glibc-2.13-linaro-multilib-2011.12/w
aveshare-linaro-toolchain/bin/../lib/gcc/arm-fsl-linux-gnueabi/4.6.2/default -lg
cc -Map u-boot.map -o u-boot
arm-linux-objcopy -O srec u-boot u-boot.srec
arm-linux-objcopy --gap-fill=0xff -O binary u-boot u-boot.bin
root@ubuntu:/opt/Embedsky/TQIMX6/E9/Ubuntu/uboot_E9_1.3.4_for_Ubuntu# ls
api                          examples        lib_mips     onenand_ipl
board                        find_log.txt    lib_nios     post
CHANGELOG                    fs              lib_nios2    README
CHANGELOG-before-U-Boot-1.1.5 include        lib_ppc      rules.mk
common                       lib_arm         lib_sh       System.map
config.mk                    lib_avr32       lib_sparc    tools
COPYING                      lib_blackfin    MAINTAINERS  u-boot
cpu                          libfdt          MAKEALL      u-boot.bin
CREDITS                      lib_generic     Makefile     u-boot.lds
disk                         lib_i386        mkconfig     u-boot.map
doc                          lib_m68k        nand_spl     u-boot.srec
drivers                      lib_microblaze  net
root@ubuntu:/opt/Embedsky/TQIMX6/E9/Ubuntu/uboot_E9_1.3.4_for_Ubuntu#
```

Follow the chapter 5.2, 5.3, and burn it in the E9 mini PC then you can do some developments to your needs.

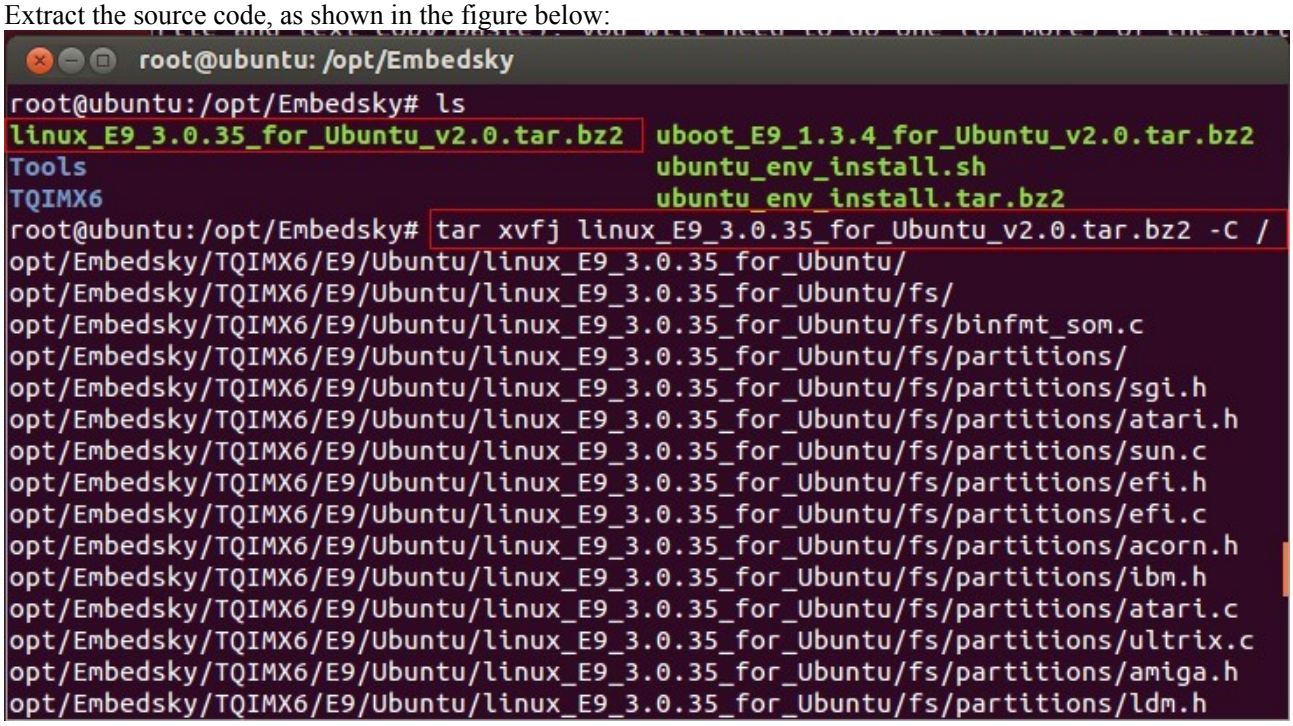
5.2.2 Compile Ubuntu Kernel

- **Extract Kernel Source Code**

  Copy the Ubuntu Kernel Source Code (\e9_cd\Source Code\Ubuntu Source Code\kernel, named linux_E9_3.0.35_for_Ubuntu_v2.0.tar) to the PC ubuntu system root directory (Here the writer of this manual

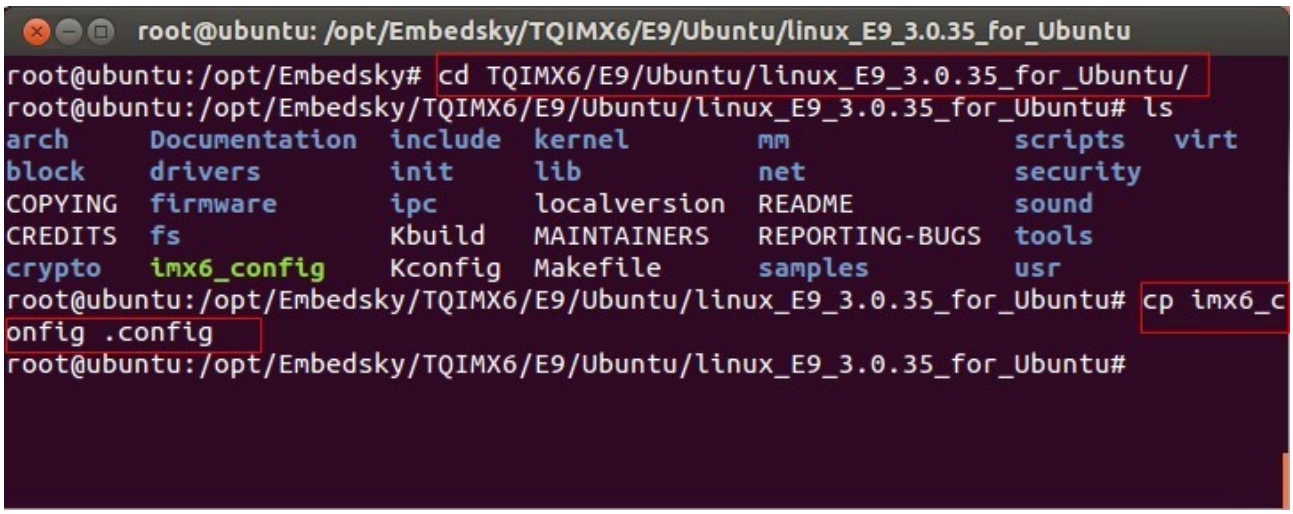
choose root directory to operate and take it a screenshots, actually, any directory will be OK), then use the command: #tar xvfj linux_E9_3.0.35_for_Ubuntu.tar.bz2 -C /

Extract the source code, as shown in the figure below:

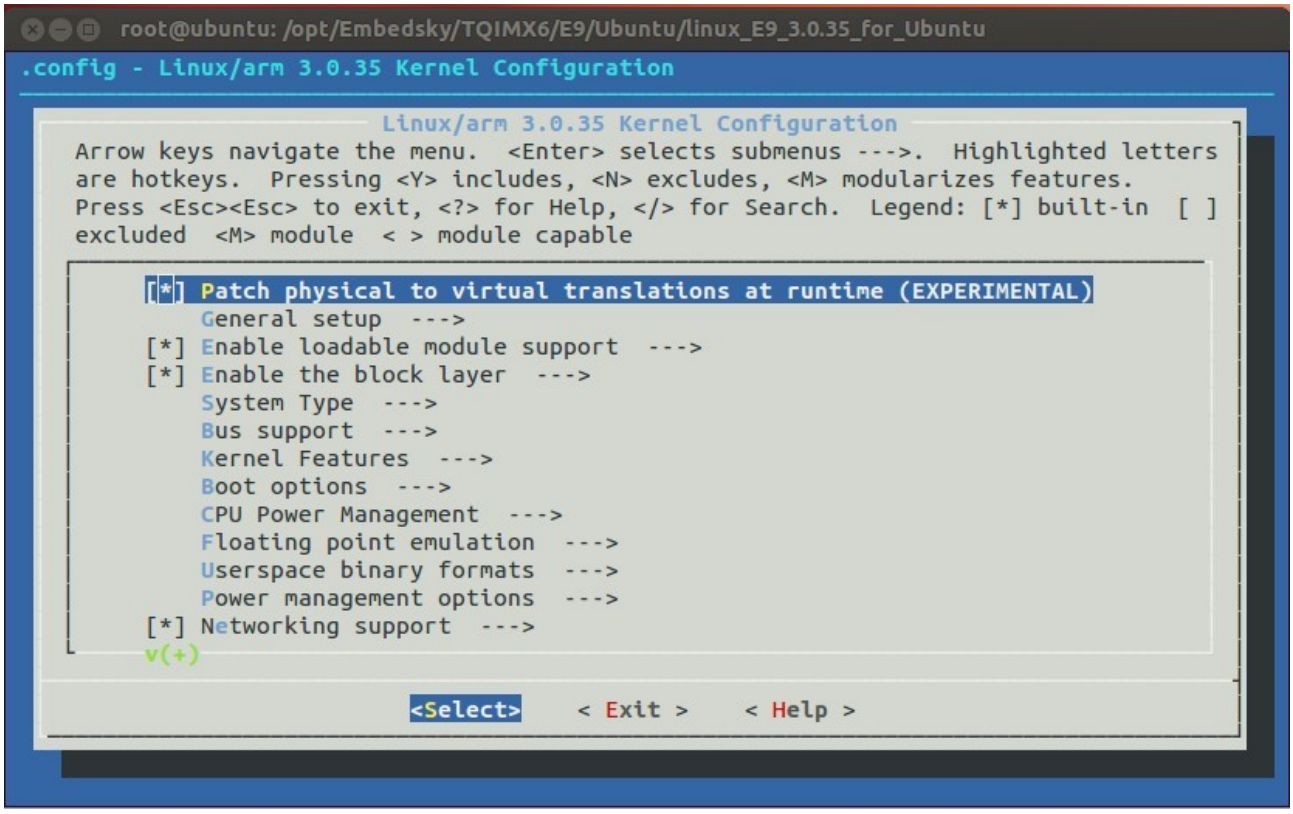


- **Kernel Source Code Compile**

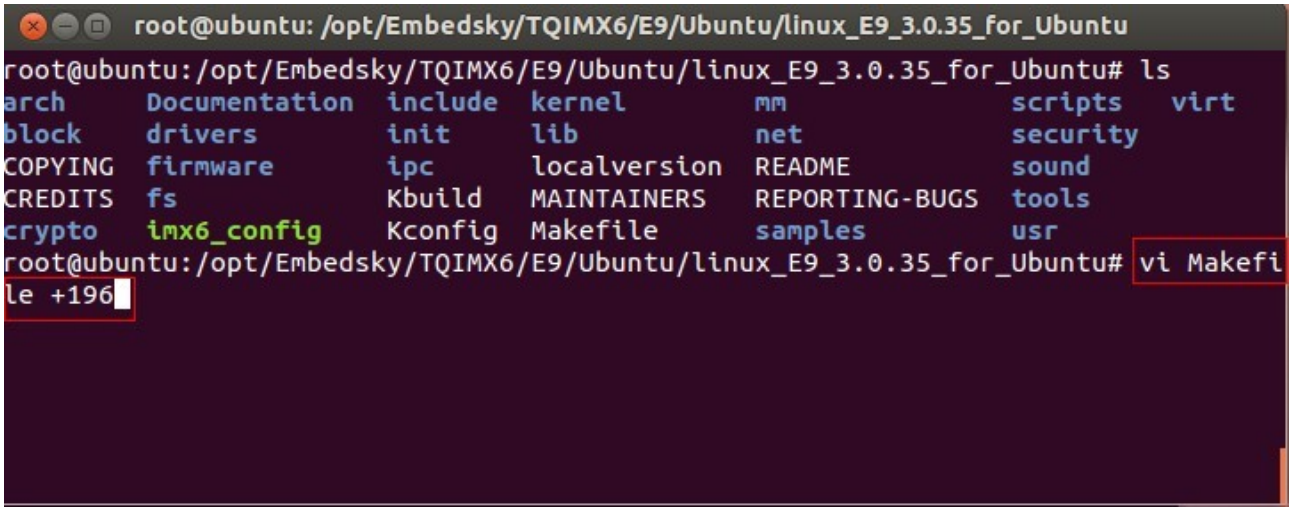
After extract the source code, please use the command: #cp imx6_defconfig .config, then copy the config list and set it as the default config list, just as the following picture shows:



Then use the command #make menuconfig to enter the config page, If you want to make any changes on the default config list, then you can amend the config list according to the needs, then save the config information:



Use the command #vi Makefile to appoint the compile tools:



Modify the CROSS_COMPILE as: /opt/Embedsky/gcc-4.6.2-glibc-2.13-linaro-mult ilib-2011.12/tq-linaro -toolchain/bin/arm-none-linux-gnueabi-

And then use the command #make uImage to compile the kernel.

```
root@ubuntu: /opt/Embedsky/TQIMX6/E9/Ubuntu/linux_E9_3.0.35_for_Ubuntu
root@ubuntu:/opt/Embedsky/TQIMX6/E9/Ubuntu/linux_E9_3.0.35_for_Ubuntu#
root@ubuntu:/opt/Embedsky/TQIMX6/E9/Ubuntu/linux_E9_3.0.35_for_Ubuntu# make uImage
  CHK     include/linux/version.h
  CHK     include/generated/utsrelease.h
make[1]: `include/generated/mach-types.h' is up to date.
  CALL    scripts/checksyscalls.sh
  CHK     include/generated/compile.h
  CC      init/do_mounts.o
  LD      init/mounts.o
  CC      init/noinitramfs.o
  CC      init/calibrate.o
  LD      init/built-in.o
  LD      usr/built-in.o
  CC      arch/arm/vfp/vfpmodule.o
  AS      arch/arm/vfp/entry.o
  AS      arch/arm/vfp/vfphw.o
  CC      arch/arm/vfp/vfpsingle.o
  CC      arch/arm/vfp/vfpdouble.o
  LD      arch/arm/vfp/vfp.o
  LD      arch/arm/vfp/built-in.o
```

After compiled, you will find the uImage in the directory: /opt/Embedsky/TQIMX6/E9/Ubuntu/linux_E9_3.0.35_for_Ubuntuarch/arm/boot, then it can be burned on the development board to test. Just as the following picture shows:

```
  Kernel: arch/arm/boot/zImage is ready
  UIMAGE  arch/arm/boot/uImage
Image Name:   Linux-3.0.35-2508-g54750ff
Created:      Sat Oct 12 11:22:33 2013
Image Type:   ARM Linux Kernel Image (uncompressed)
Data Size:    3955908 Bytes = 3863.19 kB = 3.77 MB
Load Address: 10008000
Entry Point:  10008000
  Image arch/arm/boot/uImage is ready
root@gavin:/opt/Embedsky/TQIMX6/E9/Ubuntu/linux_E9_3.0.35_for_Ubuntu# ls arch/arm/boot/
bootp  compressed  Image  install.sh  Makefile  uImage  zImage
root@gavin:/opt/Embedsky/TQIMX6/E9/Ubuntu/linux_E9_3.0.35_for_Ubuntu#
```

5.2.3 Ubuntu File System

E9 ubuntu file system is the complete system which can be founded in “Image\Ubuntu\Rootfs” directory.

Chapter Six Ubuntu System Burning

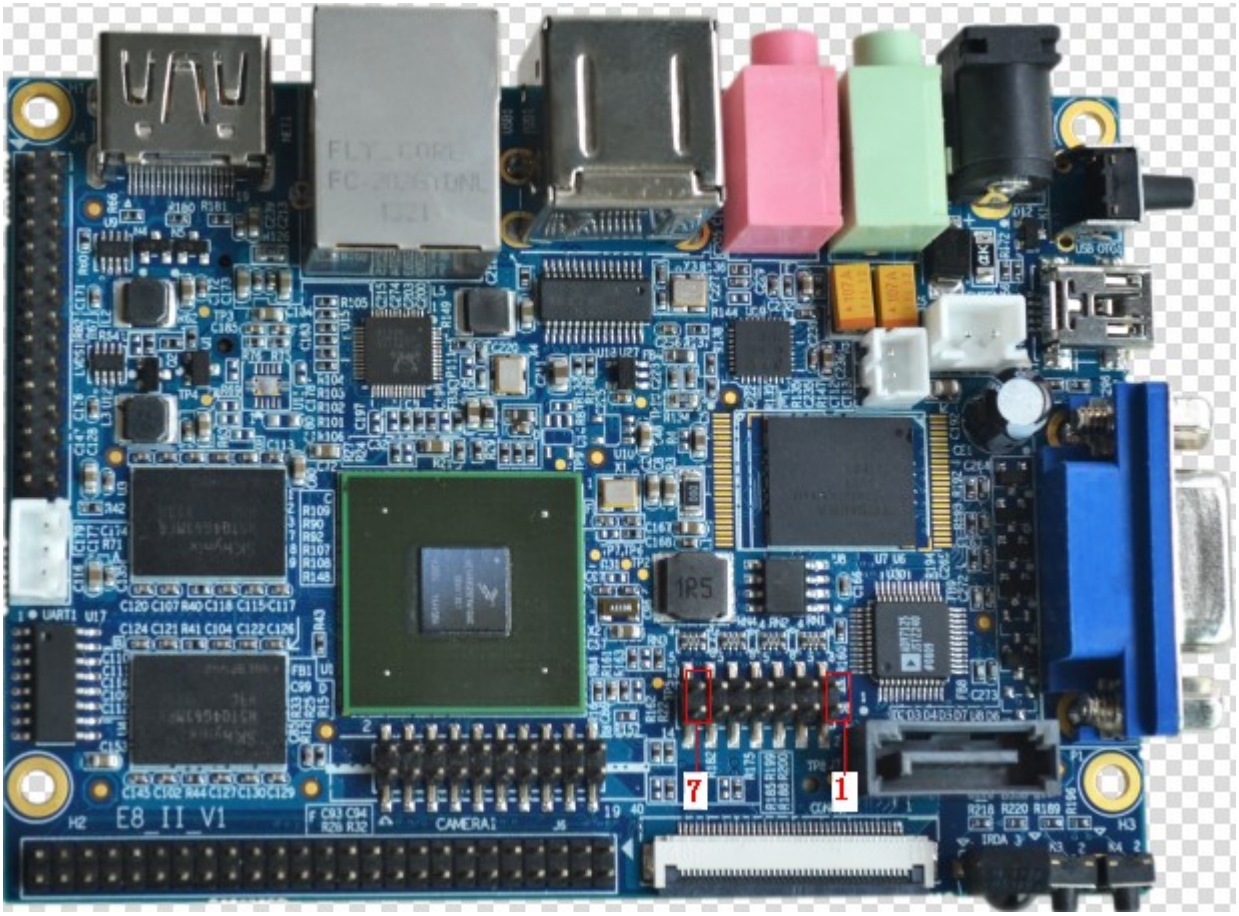
6.1 Preparations

E9 provides two boot modes: boot from the SD card and emmc respectively. Boot mode selection (eMMC, SD can be chose from the onboard SW6 DIP switch. Here we will introduce the DIP switch status and boot modes.

6.1.1 Boot Modes Selection

J8 is the E9 mini PC BOOT mode selection interface; Short different jumper cap will have different boot modes. It is 2.00mm pitch and 10Pin interface and it needs to work with 2.00mm pitch jumper cap.

Here we will use download mode, please short the pin 2 and 3 first.



E9 Mini PC keeps all the boot modes, and it uses the following 3 boot modes temporarily. (In the above picture, the pins, from right to left, are 1, 2, 3, 4, 5, 6, 7)

Boot select table							
Boot device	7	6	5	4	3	2	1

	Bt-cfg2_3	Bt-cfg2_4	Bt-cfg2_5	Bt-cfg2_6	Bt-cfg1_4	Bt-cfg1_5	Bt-cfg1_6
emmc	1	1	1	X	X	1	1
sd	1	0	0	X	x	0	1
Download	0	0	0	0	1	1	0

Note: If the Ethernet upwards, then the J8 interface, from left to right, the pins are 7, 6, 5, 4, 3, 2, 1, So corresponding to the above table, short the Jumper Cap is marked as “1”;” X” means it can connect and it can also be open.

6.1.2 Burning the Image

There are two ways for update the E9 firmware: burning via SD card or burning by MFG-tools.

SD card burning mainly provides a convenient and efficient firmware updated way for customers in the ubuntu system when developing. And it also applies to burn in a one-to-many way in the mass production to increase production capacity and save labor costs.

MFG-tools is a kind of burning tool for E9 under PC environment, which supports ubuntu 12.04 burning, simple, convenient, and easy to carry out. It is suitable for users to update the firmware in Windows environment. Please follow the tips below to finish image update.

- **Tools and Images**

Burning tool Mfgtool	Run in the Windows, and save it in “Software\Windows tools”
Virtual terminal software	Such as Windows's own super terminal
Kernel Image	uImage
u-boot Image	u-boot.bin
File system	oneiric.tgz
E9 mini PC	1pcs
USB OTG cable	1pcs
J5 jumper cape	1pcs
5V-2A DC Power Supply	1pcs

6.2 Windows Platform Burning

- **Burning Environment Setup**

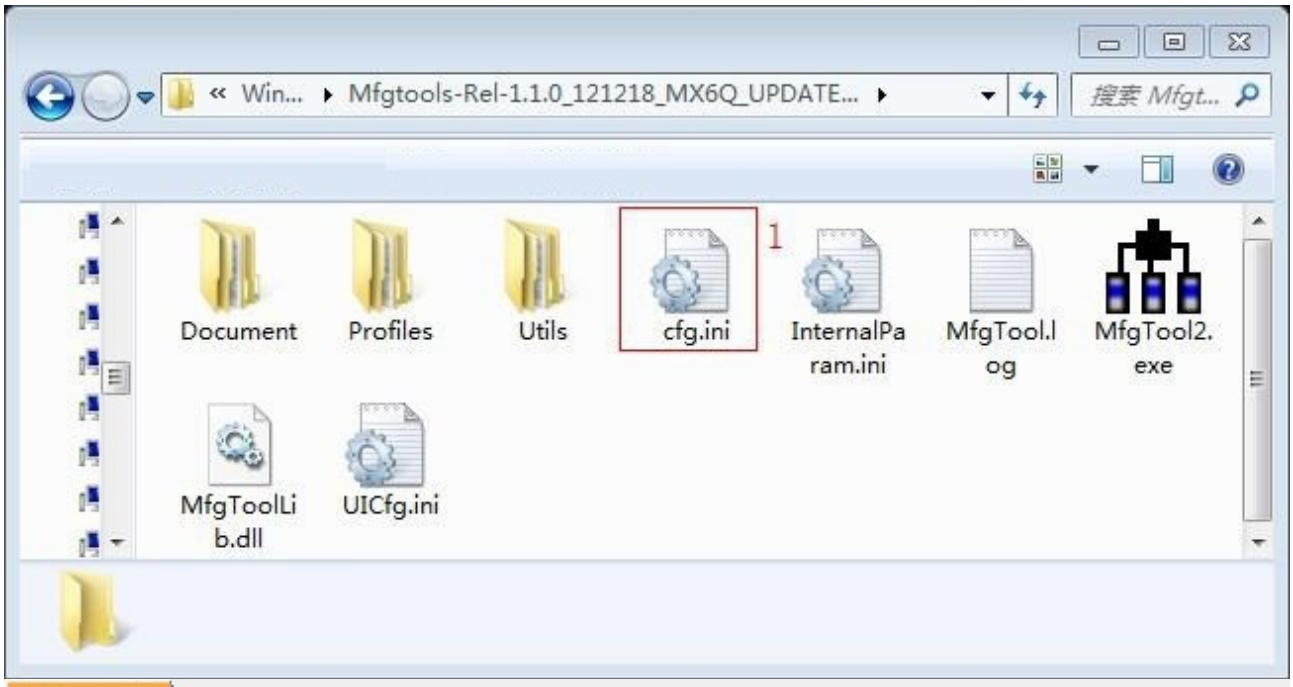
Copy Mfgtools-Rel-1.1.0_121218_MX6Q_UPDATER.zip to your computer and extract it to Mfgtools-Rel-1.1.0_121218_MX6Q_UPDATER.

6.2.1 Burning Ubuntu12.04 System

- **Modify the config file**

1) Open the cfg file which is marked in red “1”;

2) Change the “2” marked in red box to “Ubuntu” (The first letter "U" must be capitalized, as the system supports case-sensitive).



cfg.ini

```
1  [profiles]
2  chip = MX6Q Linux Update
3
4  [platform]
5  board = SabreSD
6
7  [LIST]
8  name = Ubuntu-SabreSD-eMMC
9
```

2

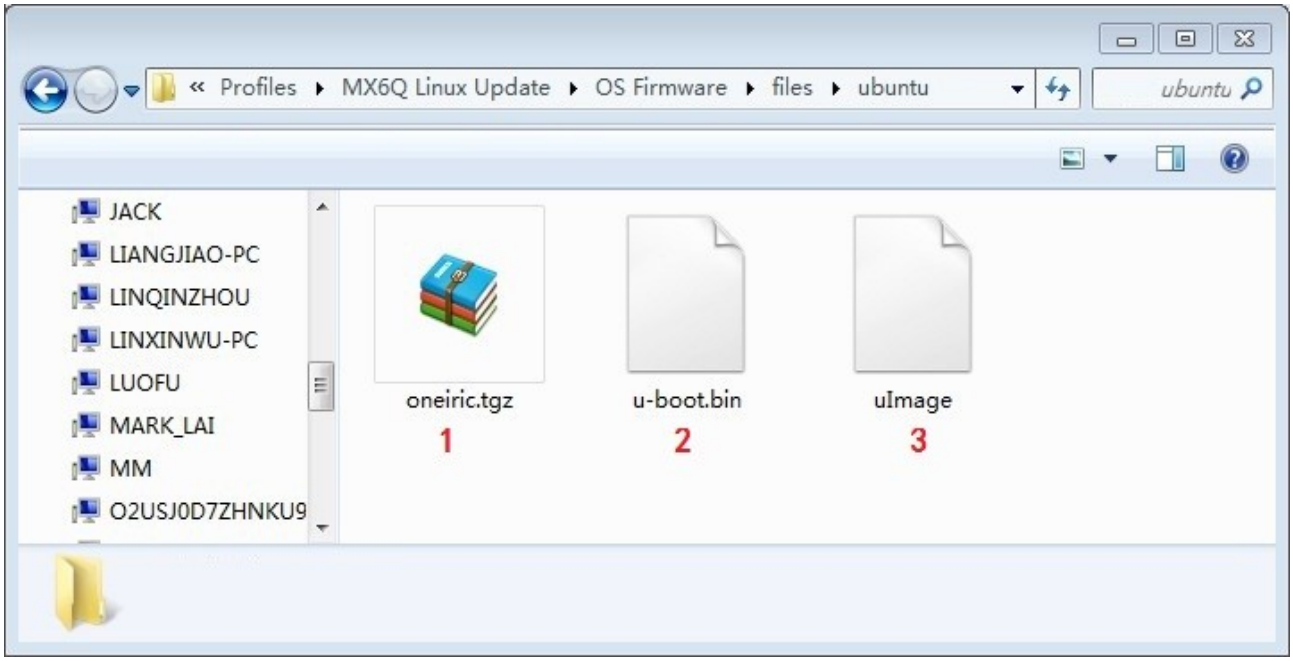
- **Copy the file**

Uboot Image	Copy u-boot.bin to Mfgtools-Rel-1.1.0_121218_MX6Q_UPDATER\Profiles\MX6Q Linux Update\OS Firmware\files\ubuntu (Or replace an existing file)

Kernel Image	Copy uImage to Mfgtools-Rel-1.1.0_121218_MX6Q_UPDATER\ Profiles\MX6Q Linux Update\OS Firmware\files\ubuntu (Or replace an existing file)
File System Image	Copy oneiric.tgz to Mfgtools-Rel-1.1.0_121218_MX6Q_UPDATER\Profiles\MX6Q Linux Update\OS Firmware\files\ubuntu (Or replace an existing file)

Note: After copy the files; please change the "u-boot_E9_for-ubuntu_v2.0.bin" to "u-boot.bin", delete "_E9_for-ubuntu_v2.0", and so ad the "uImage" and "oneiric.tgz".

After done, the file structure under the directory "Mfgtools-Rel-1.1.0_121218_MX6Q_UPDATER\Profiles\MX6Q Linux Update\OS Firmware\files\ubuntu" is as showed below:



Check and confirm if all the above three files are copied successfully, all of them are indispensable.

6.2.2 Hardware Connection

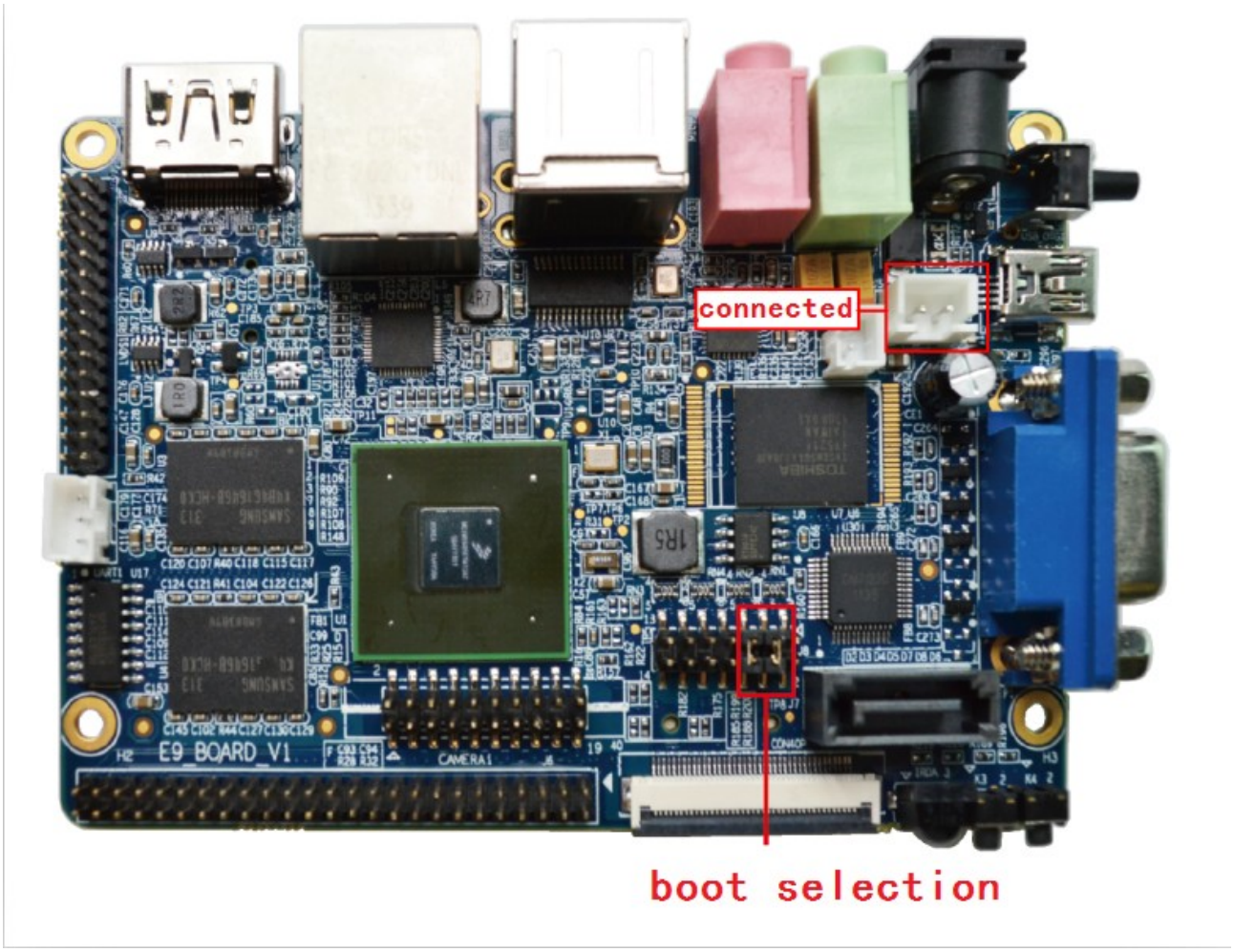
(1) Connect the PC and E9 via USB OTG cable.

(2) Connect the 5V DC power supply to E9 5V jack

(3) Set the USB download mode as the picture shows: short the pin2 and pin3 (from right to left) with jumper cap, and then download it.

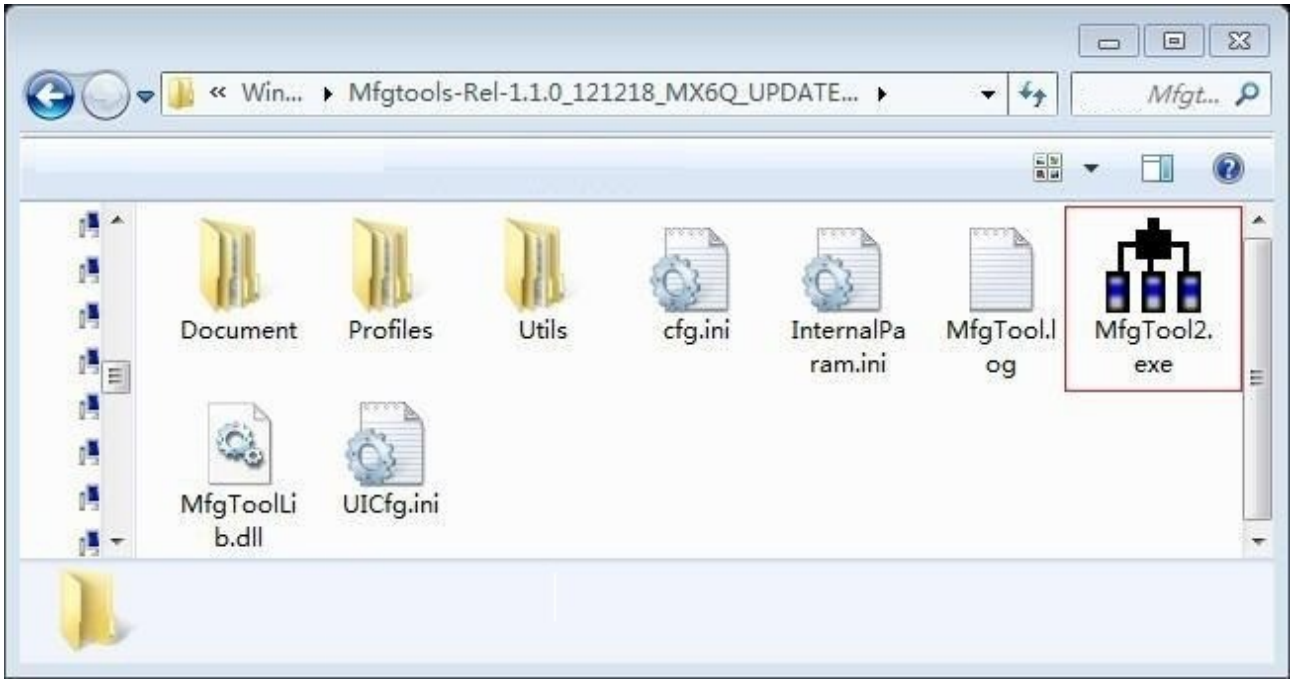
(4) Short / Connect the J5 spare external power supply interface (as marked below)

Note: J5 must be connected in the burning process.

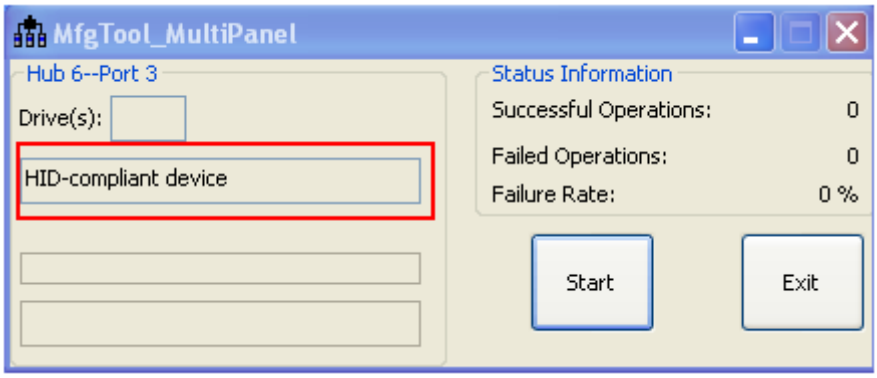


6.2.3 Begins to burn

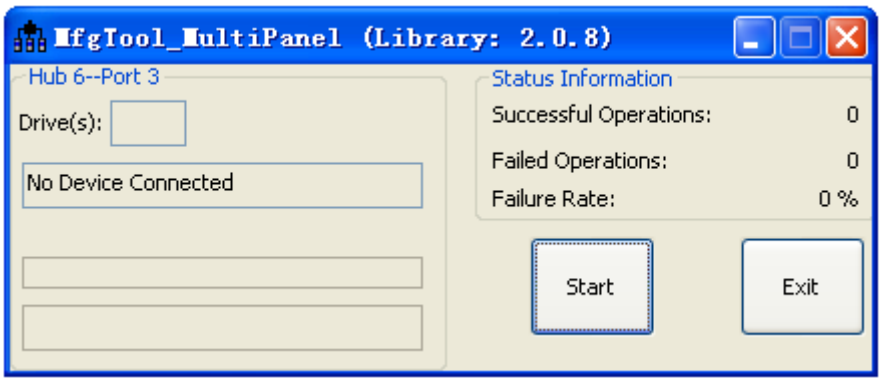
(1) Open the file “MfgTool2.exe” in the directory “Mfgtools-Rel-1.1.0_121218_MX6Q_UPDATER”:



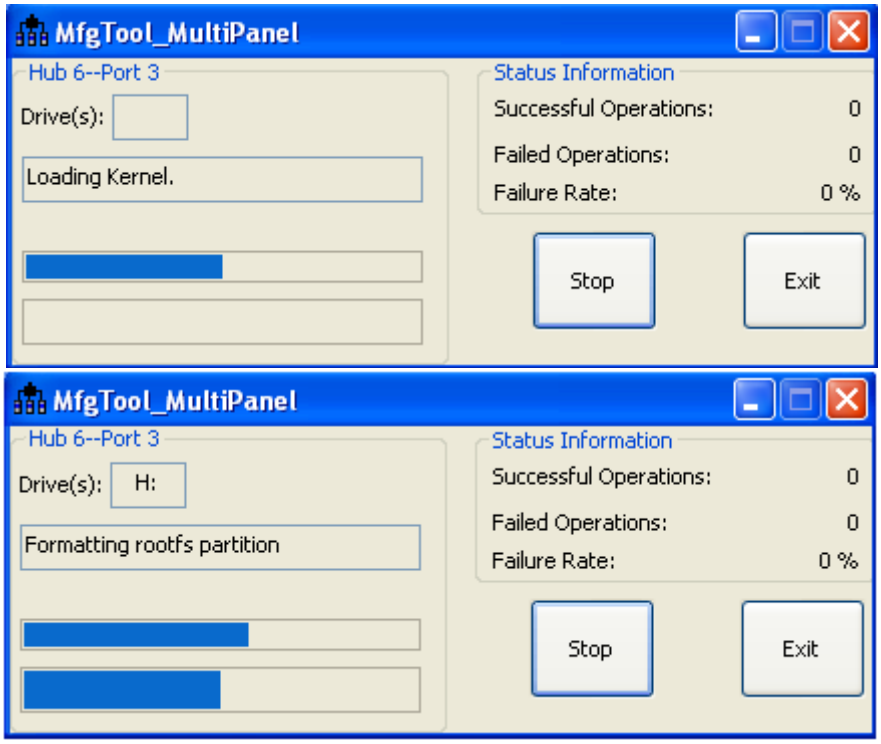
After "MfgTool2.exe" is opened, it can detect HID-compliant device as the following picture shows:



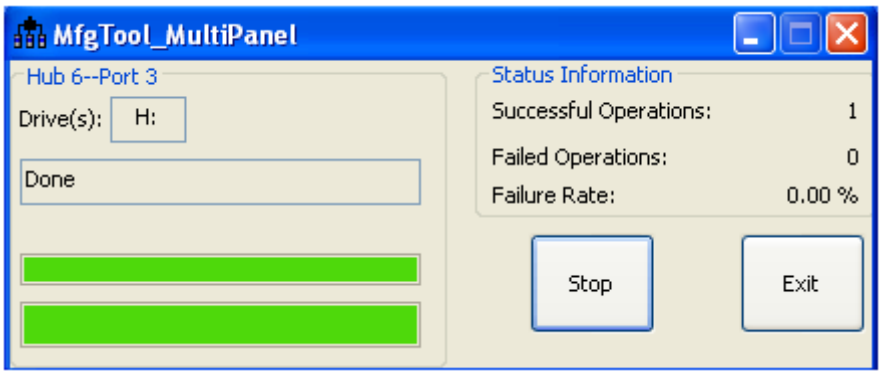
Note: If the USB device can not be detected, please recheck the OTG connection.



(2) Click the "start" button in the software MfgTool2 to begin burning (The burning process may take about 20~30 minutes due to different computer configuration, please be patient!)



(3) After done as the following picture shows, click the “stop” button, turn off the power supply and Unplug the USB OTG cable. Then follow the tips in Chapter 6.1.1, set it as emmc boot mode to run the system.



Note: After burning is done, users need to reset the default boot mode as “ubuntu used emmc” in the uboot config, then it can boot the system normally.

6.3 Linux Platform Burning

In the linux platform, it will use SD card burning automatically. Please prepare the following things before burning:

Burning tools	Software\Linux tools\downloader.tar.bz2
Kernel image	uImage
u-boot image	u-boot.bin
File System	oneiric.tgz
E9 mini PC	1pcs
Tf card	1pcs

TF card reader	1pcs
5V-2A DC Power Supply	1pcs

6.3.1 Create SD Card

Extract the prepared related tools and images to any directory #tar jxvf downloader.tar.bz2 -C / and related directory #./create_sdcard.sh -f /dev/sdc

Note: Here the example based on the writer's computer, SD card disk is the /dev/sdc, and your own computer will based on the disk after the " #fdisk –l", just avoid burning the program to your computer hard drive, damage to your data, and cause needless loss.

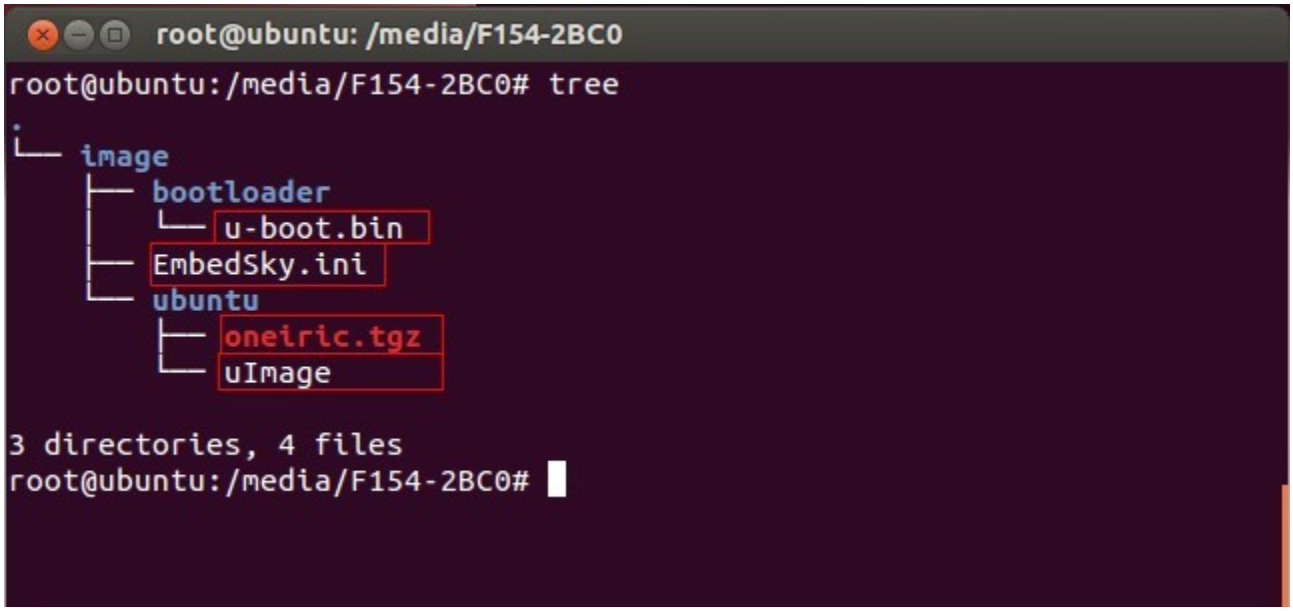
6.3.2 Prepare to burn image

SD card burning automatically is judged from the EmbedSky.ini contents on the images directory in the SD card root directory to choose whether to burn and which images need to be burned. The file "EmbedSky.ini" must be put in the SD card images directory, if there is no such directory in the SD card, and then please create a new directory like that.

6.3.3 Burning ubuntu image

- **Copy the images:**

Copy the ubuntu images "u-boot_E9_for-ubuntu_v2.0.bin", "uImage_E9_for-ubuntu_v2.0" and "oneiric_E9_for-ubuntu_v2.0.tgz" to SD card directory "images\bootloader\" and "images\ubuntu" separately, and remove the "_E9_for-ubuntu_v2.0" in the image name. After that the burning file in the SD card should be like this:



- **Modify EmbedSky.ini file**

Note 1: The file format can't change optionally, and there is no space behind "=", and it cannot add or delete a line either.

Note 2: The following "//XXX" is for comments only, and when using the file, please don't add these comments, or else there will be mistakes.

Note 3: When users burning the ubuntu, it needs to pay attention to the comments, and modify it according to actual situation. There is no need to change others.

Note 4: For those images which don't need to update, it can be set as null, such as "Kernel =".

```
#### This is E9 SD config files ####
operating condition=install
OS Type=ubuntu                    // operating system instructions
LED indicator=off
Beep=off
extern_indication=partition      // external indication, partition means it needs to partition, it is better to add this.

#### Others ####
bootloader=images\bootloader\u-boot.bin          //u-boot image storage directory
LOGO=
#### Others end ####

#### Linux ####
kernel=images\
rootfs=images\
#### Linux end ####

#### Windows CE ####
NK=images\wince\NK.bin
#### Windows CE end ####

#### Android ####
boot_img=images/android/boot.img
system_img=images/android/system.img
recovery_img=images/android/recovery.img
#### Android end ####

#### Ubuntu ####
kernel=images/ubuntu/uImage              //ubuntu uImage storage directory
rootfs=images/ubuntu/oneiric.tgz         //ubuntu rootfs storage directory
#### Ubuntu end ####

#### uCOS-II ####
uCOS-II Image=images\uCOS-II\TQ6410_UCOS-II.bin
#### uCOS-II end ####
```

```
#### User Application ####
Application Image=images\UserAPPS\TQ6410_Test.bin
#### User Application end #####
```

- **Burning System**

Put the TF card to the TF card slot, long press the power button to boot the system, then the system will complete burning automatically according to the EmbedSky.ini content. (The burning process may take about 20~30 minutes due to different computer configuration, please be patient!)

After done, the power light will be turned off automatically.

Note: After burning is done, users need to reset the default boot mode as “ubuntu used emmc” in the uboot config, and then it can boot the system normally. For the setting instructions, please check Chapter 4.4.

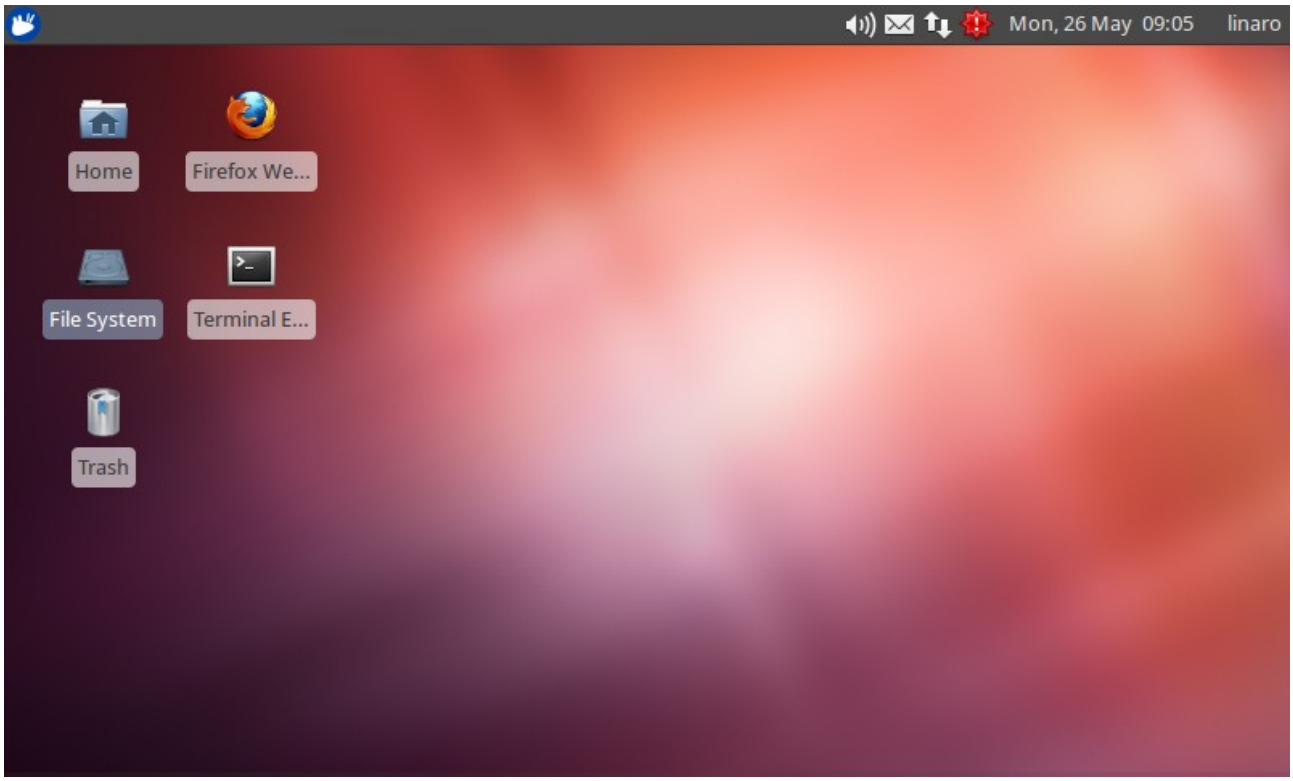
Chapter Seven Ubuntu Functional Test

This chapter is mainly for beginners to learn how to use the system ubuntu 12.04 and test the board function. Maybe some test interface is different from the actual interface when using, but as long as test it according to the following tips, then board function is no problem at all.

Set the boot mode of E9 mini PC as EMMC, power on and long press the power button. The red power light near the power button is on (If it is not long press the power button, the power light is not on at all). By this, it means that the board uboot has run up. Users can enter and set uboot; or wait for the system boots automatically.

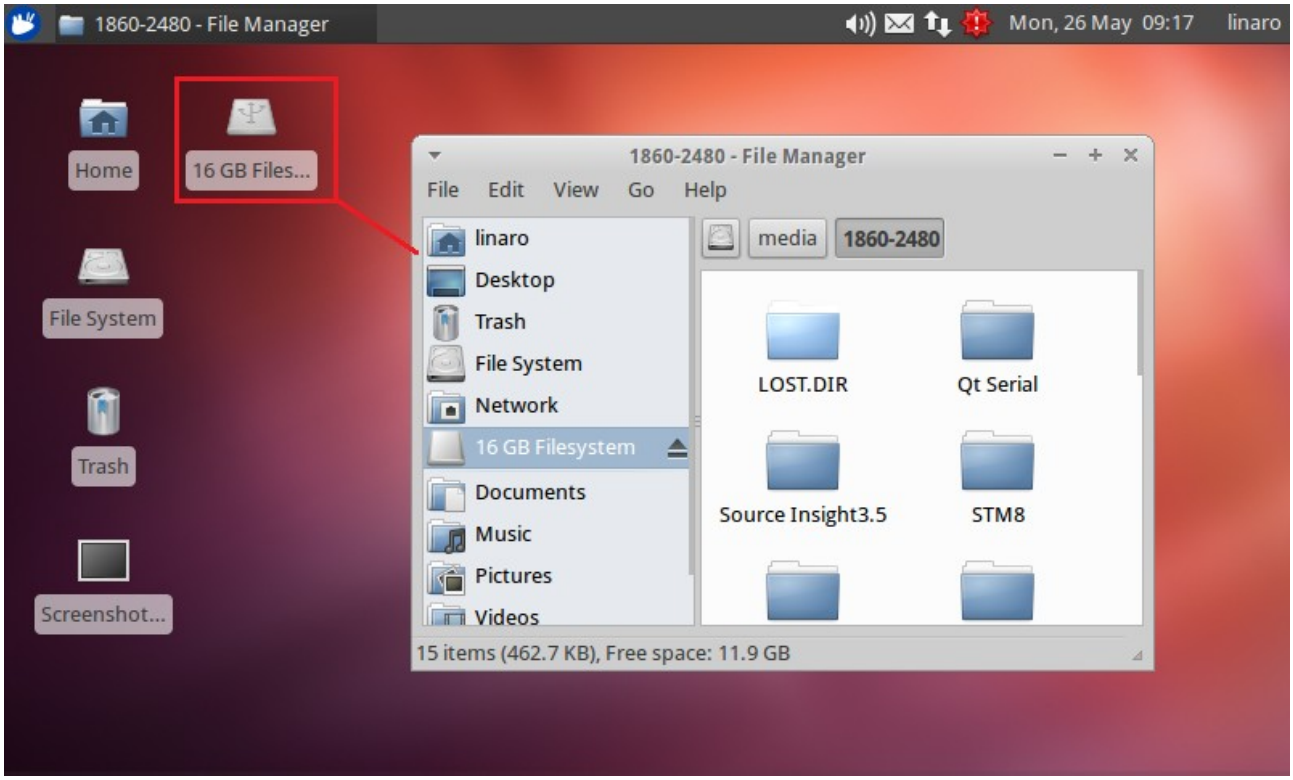
7.1 Display Test

The E9 ubuntu supports 7inch/5inch LCD display, VGA, HDMI display. Setting Resolution according to the tips in Chapter8.1, and choose LCD, or VGA or HDMI display according to your own display in hand. The display shows like this if it's normally boot:



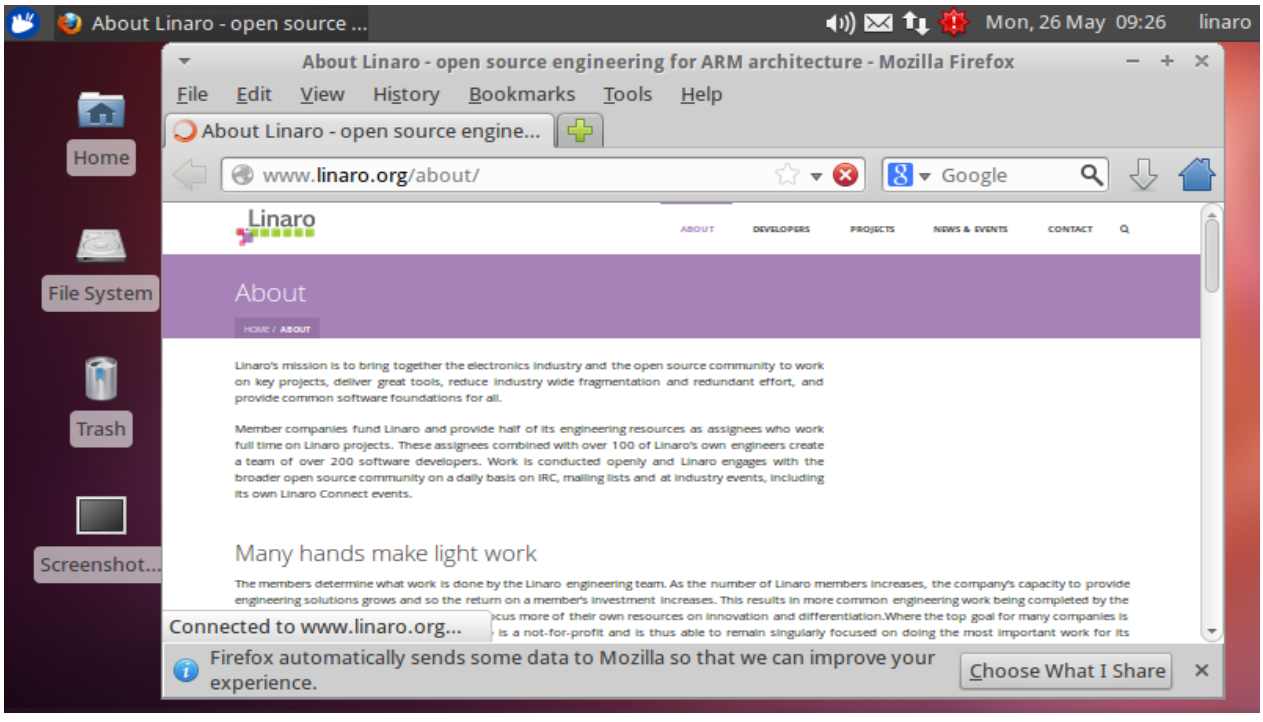
7.2 USB Test

Connect USB keyboard and mouse. Move the mouse, then you can see the Mouse icon moves on the display. And it can also connect U disk, it will pop up the file folder automatically, then read and write the data. If it is not pop up, then you can use the command # fdisk -l in the terminal to check whether there is a new dish.



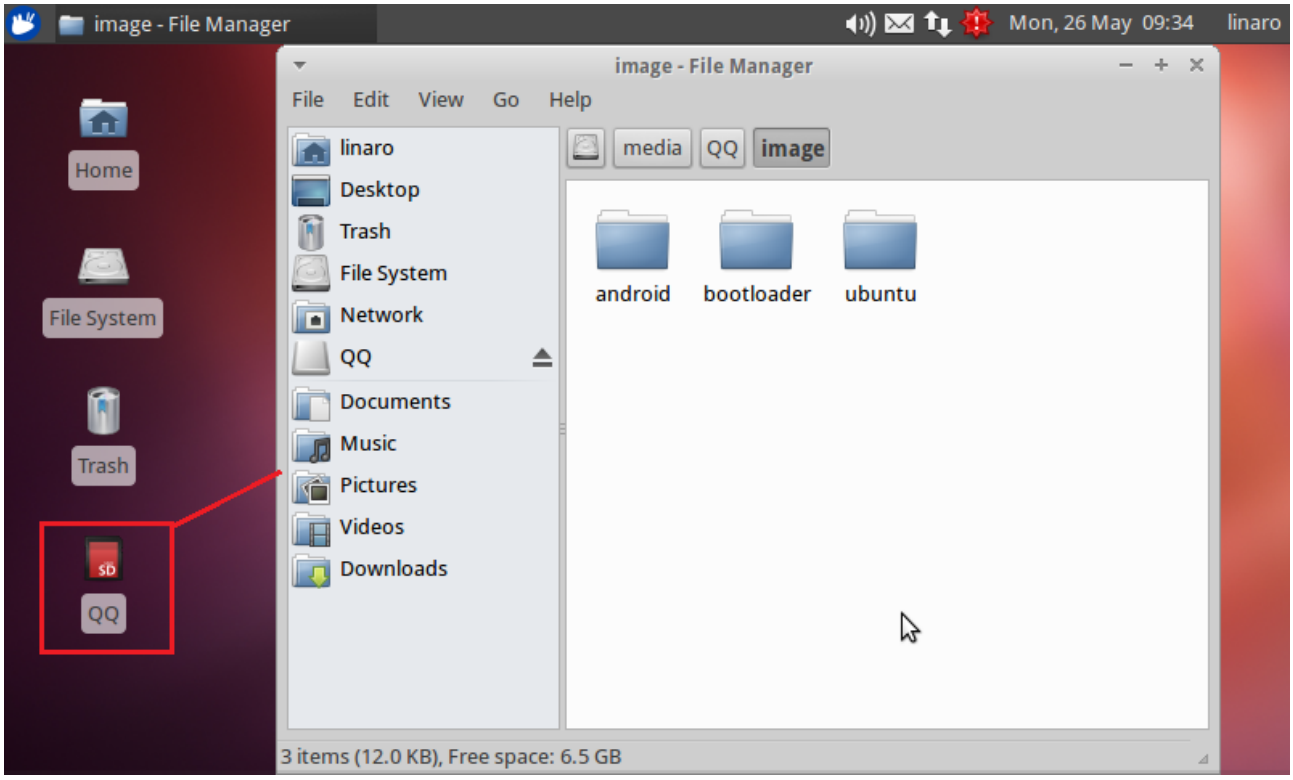
7.3 Network Test

Check the upper right corner to see whether the network state is normal or not (If there is something wrong with the network connection, then the icon will be gray). The default system is dynamic IP. To set a static, it can be set according to its own network parameters. After checking the network, you can open the Firefox browser on the desktop and surf the Internet or watch online video (flash plugin is needed to install manually).



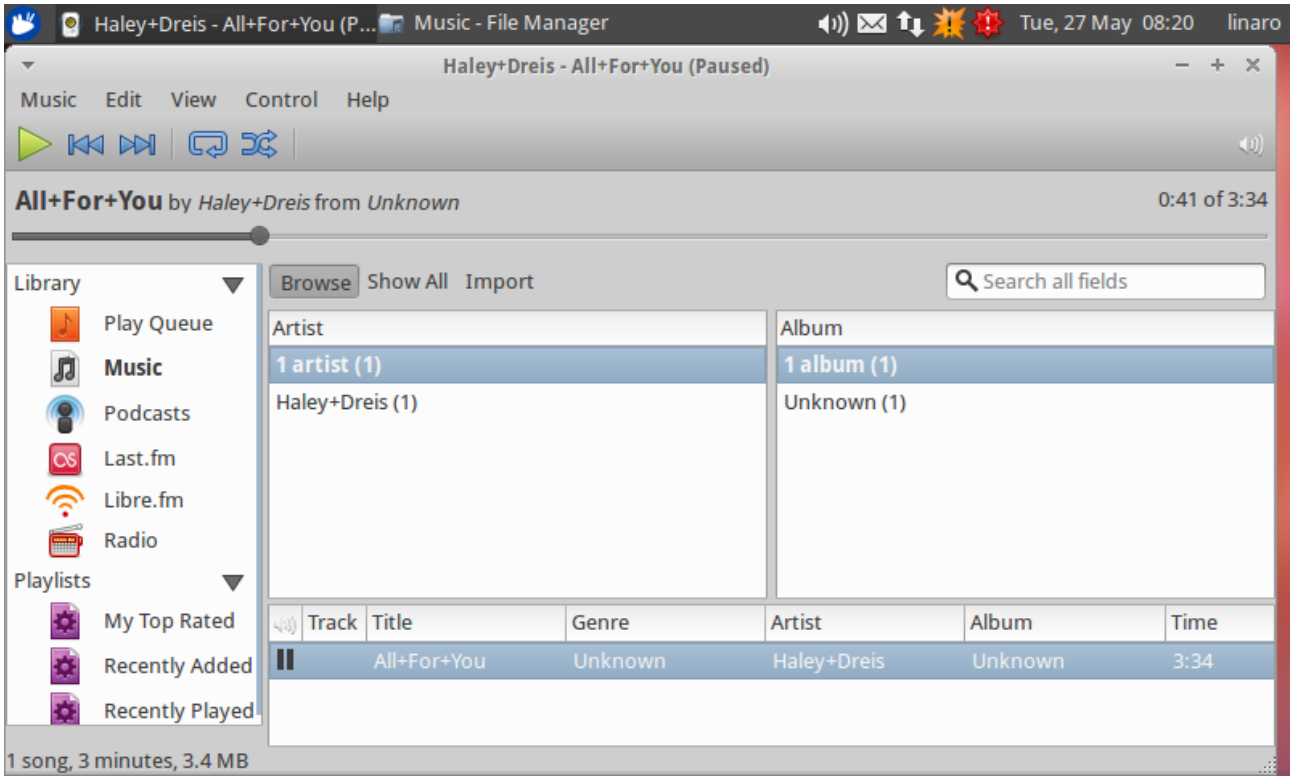
7.4 TF Card Test

Insert a TF card, and it will pop up a TF card folder automatically. The TF card and E9 will copy data each other, the read and write speed is very fast, reaching 10M/s or more. And currently, it supports 32G TF card maximumly, and also supports hot-swappable.



7.5 Music Test

Use music player Rhythmbox to play music, and output via speaker by default. When plugged in headphone, it will switch to the headphone output automatically. Unplug the headphone, then it will restore to speaker output. Volume can be adjusted by the upper right corner volume controller. Besides, the Rhythmbox also supports podcasts, radio and other functions. User can use them according to their own habits. They can also download the favorite music player in the software center.

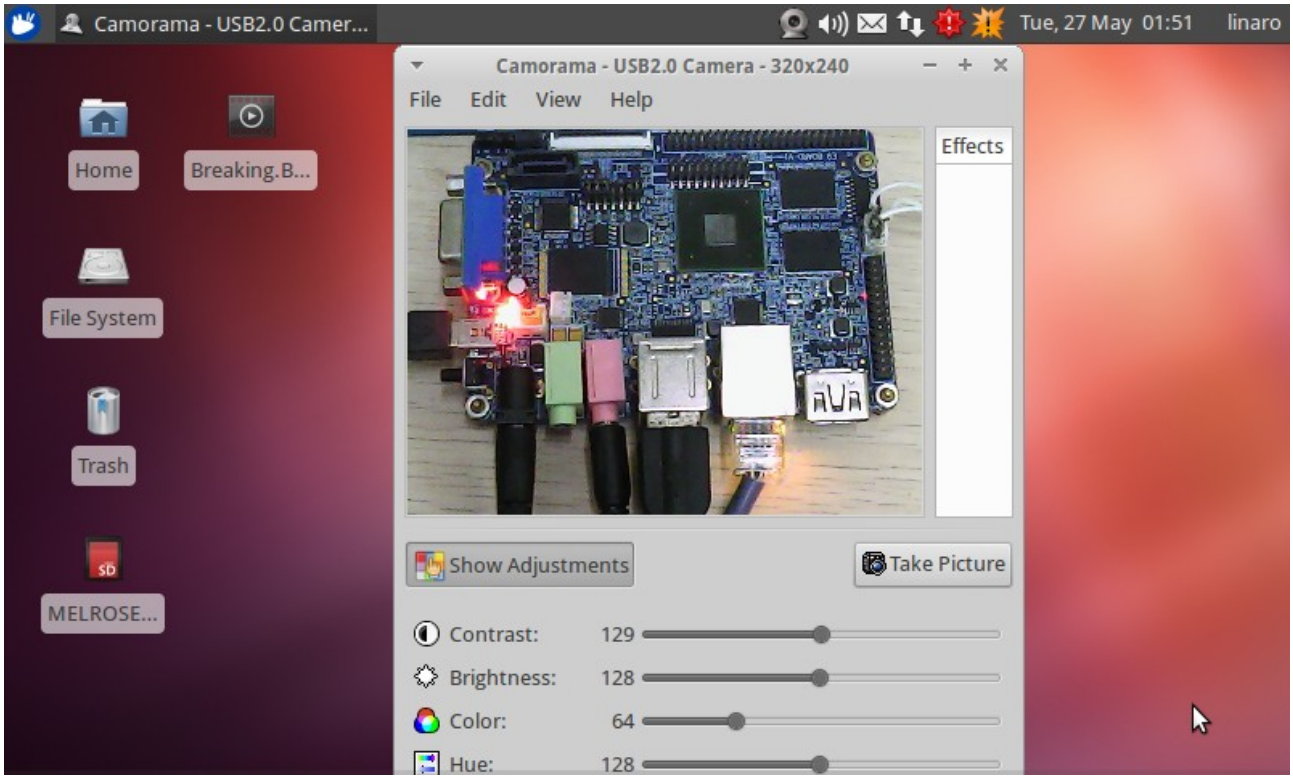


7.6 Video Test

It supports to play local video (720P, 1080P HD video) or network video (need to install flash plug-in). And it can use the movie player that comes with system to play the video. You can also download the favorite video player in the software center.

7.7 USB Camera Test

Connect the USB camera, then open the camera viewer, you will see the image. You can set the photo size and storage paths and other parameters. If it is not normal, then please check whether there is a node /dev/video0 generated, and then find reasons.



7.8 Serial Port Test

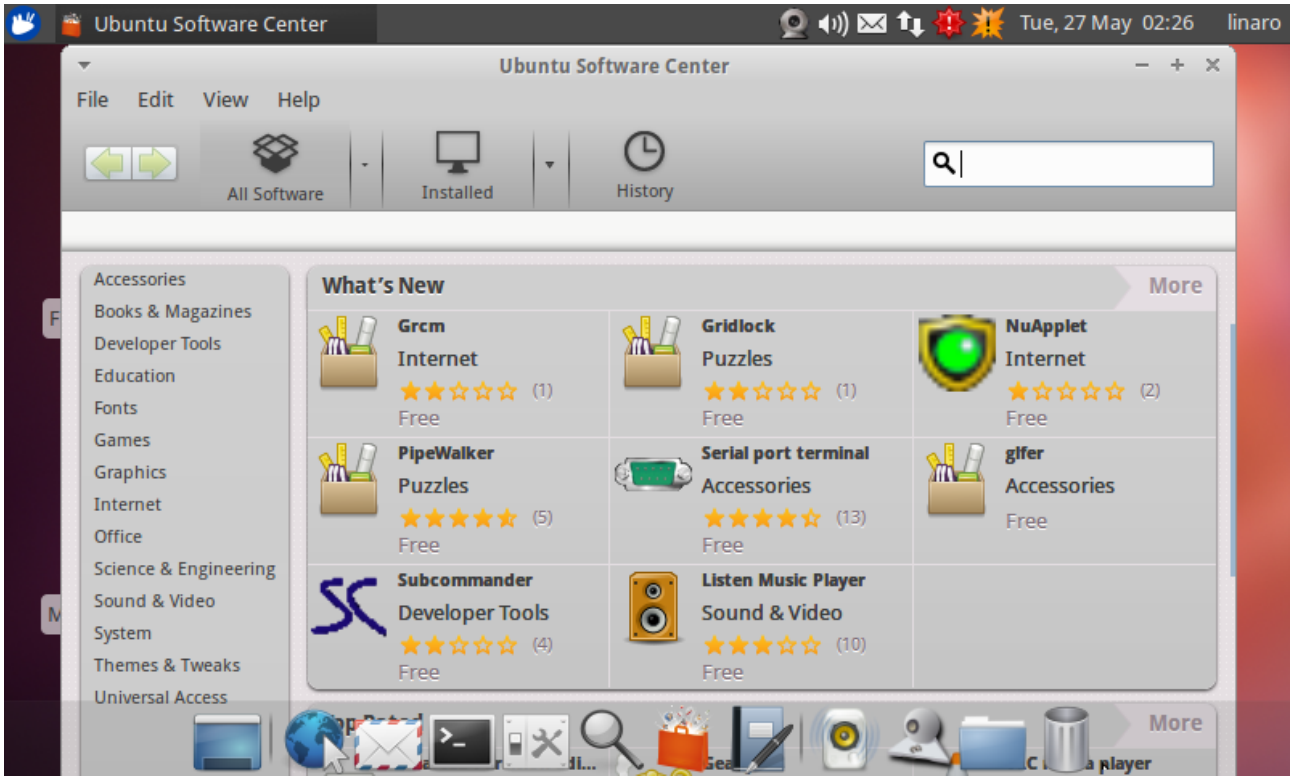
E9 has one debug serial port only, so as long as you can see the debugging information in using, and it can communicate with E9 in the serial terminal, then it means that the serial port is OK.

7.9 Hard Disk Test

Connect the hard disk (external power supply is required) and then boot it, open the file browser, then you can see the hard disk will be mounted automatically. It can identify the EXT3, FAT, NTFS format, etc. Here it is mounted temporarily. If you want to mount it directly to a directory, then you can mount it in accordance with the method of computer ubuntu.

7.10 Others

In the Ubuntu Software Center, you can download a variety of software needed, as shown below:



In the above "application menu", there are applications and services that are already installed; for the familiar use, you can go and check one by one. The navigation bar is at the bottom; you can increase or delete software shortcuts.

Android

Introduction

Note: In the following operation, the "#" before the commands indicates a root user authority to use. It is recommended to use the root user to login Linux PC, if not, please uses the sudo command to gain root user privileges, or may fail to compile.

Chapter Eight Android Source Code & Compile

8.1 Android Source Code:

Android Source Code includes kernel source code, uboot source code, and the file system source code

Uboot: bootable/bootloader/uboot-imx/

Kernel: kernel_imx/

Others are for android file system code.

Android kernel version is kernel_3.0.35, the following paths are relative to the kernel root directory. As the driver code involves a lot, and then it is difficult to introduce them one by one, so the driver source code mentioned below is only for the important ones and those which are used in high frequency.

* represents number 0-9, and when it is after "/", it represents all of the directory.

Driver	Location	Name
emmc/sdcard driver	drivers/mmc/host: sdhci-esdhc-imx.c sdhci.c sdhci-platform.c sdhci-pltfm.c	/dev/mmcblk*
LCD & LVDS driver	drivers/video/mxc/: ldb.c mxc_dispdrv.c mxc_dvi.c mxc_ipuv3_fb.c mxc_lcdif.c	/dev/graphics/fb*
capacitive touch screen driver	drivers/input/touchscreen： gt811.c	dev/input/event*
resistive touch screen driver	drivers/input/touchscreen： tsc2007.c	dev/input/event*
USB OTG driver	drivers/usb/otg： fsl_otg.c	
cable network driver	drivers/net： fec.c	eth0
serial port driver	drivers/tty/serial： imx.c mxc_uart_early.c serial_core.c	/dev/ttySAC*
WM8960 audio driver	sound/soc/imx/*	/dev/snd/*
I2C driver	drivers/i2c/busses: i2c-imx.c	
keypad driver	drivers/input/keyboard： gpio_keys.c	/dev/input/event*
Irda Driver	drivers/input/keyboard：	/dev/input/event*

	tq_hs0038.c	
backlight driver	drivers/video/backlight/ : pwm_bl.c backlight.c	
pwm driver	arch/arm/plat-mxc/ : pwm.c	
RTC driver		/dev/rtc*
Watch dog driver	drivers/watchdog : imx2_wdt.c	/dev/watchdog
Multimedia video camera driver	drivers/media/video/mxc/capture/*	/dev/video*
Sata driver	drivers/ata : ahci_platform.c libahci.c	/dev/sd*

8.2 Andorid Compiling

Please establish android compilation environment before compiling the source code.

Please copy the Android Kernel source code (the TQIMX6_Android4.2.2_v2.0.tar.bz2 in the Android4.2 part1) to the root directory of the Linux system of PC. (Here the writer of this manual choose root directory to operate and take it a screenshots, actually, any directory will be OK), then use the command:

```
#tar  xvfj  TQIMX6_Android4.2.2_v2.0.tar.bz2  -C  /
```

Extract the source code, as shown in the figure below:

```
root@z:/opt# ls
EmbedSky  lost+found  qingleqq  TQIMX6_Android4.2.2_v1.0.tar.bz2
root@z:/opt# tar jxvf TQIMX6_Android4.2.2_v1.0.tar.bz2 -C /
opt/EmbedSky/TQIMX6/E9/TQIMX6_android-4.2.2/abi/
opt/EmbedSky/TQIMX6/E9/TQIMX6_android-4.2.2/abi/cpp/
opt/EmbedSky/TQIMX6/E9/TQIMX6_android-4.2.2/abi/cpp/Android.mk
opt/EmbedSky/TQIMX6/E9/TQIMX6_android-4.2.2/abi/cpp/src/
opt/EmbedSky/TQIMX6/E9/TQIMX6_android-4.2.2/abi/cpp/src/class_type_info.cc
opt/EmbedSky/TQIMX6/E9/TQIMX6_android-4.2.2/abi/cpp/src/function_type_info.cc
opt/EmbedSky/TQIMX6/E9/TQIMX6_android-4.2.2/abi/cpp/src/new.cc
opt/EmbedSky/TQIMX6/E9/TQIMX6_android-4.2.2/abi/cpp/src/pointer_to_member_type_info.cc
opt/EmbedSky/TQIMX6/E9/TQIMX6_android-4.2.2/abi/cpp/src/vmi_class_type_info.cc
opt/EmbedSky/TQIMX6/E9/TQIMX6_android-4.2.2/abi/cpp/src/pbase_type_info.cc
```

Note:

1. The File is large; it is about 7G after extracting and about 30G after compiled. It is suggested to extract it in a place more than 35G free space.

2. The following screenshot may be different from the actual version, so the screenshots are for reference only. Please subject to in the chapter text and the actual case.

3. Compiling the script directly, it will well compile the kernel, android file system intact. And please use the root user to compile. If using the ordinary user to compile, the compiled file system may be failed to boot.

```
#./build_sh
```

```
root@z:/opt/EmbedSky/TQIMX6/E9/TQIMX6_android-4.2.2# ls
abi         build_sh     device      gdk         libnativehelper  pdk
bionic      cts          docs        hardware    Makefile         prebuilts
bootable    dalvik       external    kernel_imx  ndk              sdk
build       development  frameworks  libcore     packages         system
root@z:/opt/EmbedSky/TQIMX6/E9/TQIMX6_android-4.2.2# ./build_sh
including device/asus/grouper/vendorsetup.sh
including device/asus/tilapia/vendorsetup.sh
including device/fsl/imx5x/vendorsetup.sh
including device/fsl/imx6/vendorsetup.sh
including device/generic/armv7-a-neon/vendorsetup.sh
including device/generic/armv7-a/vendorsetup.sh
including device/generic/mips/vendorsetup.sh
including device/generic/x86/vendorsetup.sh
including device/lge/mako/vendorsetup.sh
including device/samsung/maguro/vendorsetup.sh
including device/samsung/manta/vendorsetup.sh
including device/samsung/toroplus/vendorsetup.sh
including device/samsung/toro/vendorsetup.sh
including device/ti/panda/vendorsetup.sh
including sdk/bash_completion/adb.bash

============================================
PLATFORM_VERSION_CODENAME=REL
PLATFORM_VERSION=4.2.2
TARGET_PRODUCT=sabresd_6dq
TARGET_BUILD_VARIANT=user
TARGET_BUILD_TYPE=release
TARGET_BUILD_APPS=
TARGET_ARCH=arm
TARGET_ARCH_VARIANT=armv7-a-neon
HOST_ARCH=x86
HOST_OS=linux
HOST_OS_EXTRA=Linux-3.5.0-23-generic-x86_64-with-Ubuntu-12.04-precise
HOST_BUILD_TYPE=release
BUILD_ID=1.0.0-rc3
OUT_DIR=out
```

Compiling takes very a long time which is different due to the difference of computer configurations. And the author's computer probably takes four hours. And it is a normal phenomenon that the computer may be very slow.

After compiled, it will generate relevant image in out/target/product/sabresd_6dq. When burning, it may need to use the images u-boot.bin, boot.img, system.img, recovery.img;

```
Creating filesystem with parameters:
    Size: 293601280
    Block size: 4096
    Blocks per group: 32768
    Inodes per group: 5984
    Inode size: 256
    Journal blocks: 1120
    Label:
    Blocks: 71680
    Block groups: 3
    Reserved block group size: 23
Created filesystem with 1147/17952 inodes and 55675/71680 blocks
+ '[' 0 -ne 0 ']'
Install system fs image: out/target/product/sabresd_6dq/system.img
out/target/product/sabresd_6dq/system.img+out/target/product/sabresd_6dq/obj/PACKAGING/recovery_patch_interm
12 blocksize=4224 total=294095346 reserve=3028608
root@gavin:/opt/work/mx6/android/TQIMX6_android-4.2.2# ls out/target/product/sabresd_6dq/
android-info.txt  dex_bootjars         previous_build_config.mk  recovery.img  system.img      u-boot.bin
boot.img          installed-files.txt  ramdisk.img               root          test            uImage
clean_steps.mk    kernel               ramdisk-recovery.img      symbols       u-boot-6dl.bin  userdata.img
data              obj                  recovery                  system        u-boot-6q.bin
root@gavin:/opt/work/mx6/android/TQIMX6_android-4.2.2#
```

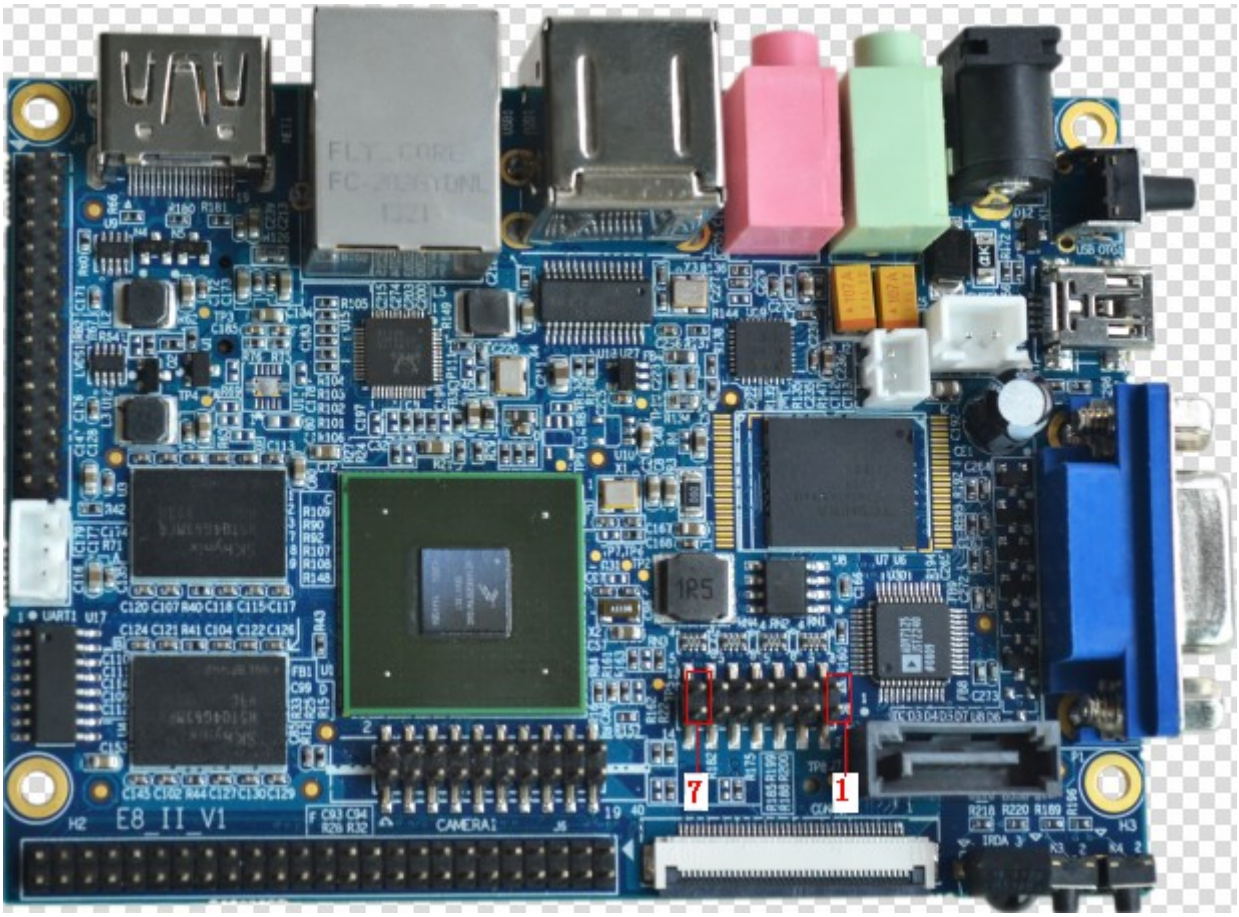
Chapter Nine Android System Burning

9.1 Preparations

E9 provides two boot modes, respectively boot from the SD card and boot from the EMMC. The boot mode is depended on DIP switch SW6 onboard. Here we will introduce the DIP switch status and boot mode.

9.1.1 Boot Mode Selection

J8 is the E9 mini PC BOOT mode selection interface. Shorting different jumper cap will have different boot modes. It is 2.00mm pitch and 10Pin interface and it needs to work with 2.00mm pitch jumper cap.
Here we will use download mode, please short the pin 2 and 3 first.



E9 Mini PC keeps all the boot modes, and it uses the following 3 boot modes temporarily. (In the above picture, the pins, from right to left, are 1, 2, 3, 4, 5, 6, 7)

Boot select table							
Boot device	7	6	5	4	3	2	1

	Bt-cfg2_3	Bt-cfg2_4	Bt-cfg2_5	Bt-cfg2_6	Bt-cfg1_4	Bt-cfg1_5	Bt-cfg1_6
emmc	1	1	1	X	X	1	1
sd	1	0	0	X	x	0	1
Download	0	0	0	0	1	1	0

Note: If the Ethernet upwards, then the J8 interface, from left to right, the pins are 7, 6, 5, 4, 3, 2, 1, So corresponding to the above table, short the Jumper Cap is marked as "1";" X" means it can connect and it can also be open.

9.1.2 Burning the Image

There are two ways for update the E9 firmware: burning via SD card or burning by MFG-tools.

SD card burning mainly provides a convenient and efficient firmware updated way for customers in the ubuntu system when developing. And it also applies to burn in a one-to-many way in the mass production to increase production capacity and save labor costs.

MFG-tools is a kind of burning tool for E9 under PC environment, which supports ubuntu 12.04 burning, simple, convenient, and easy to carry out. It is suitable for users to update the firmware in Windows environment. Please follow the tips below to finish image update.

9.2 Windows Platform Burning

- Tools and Images

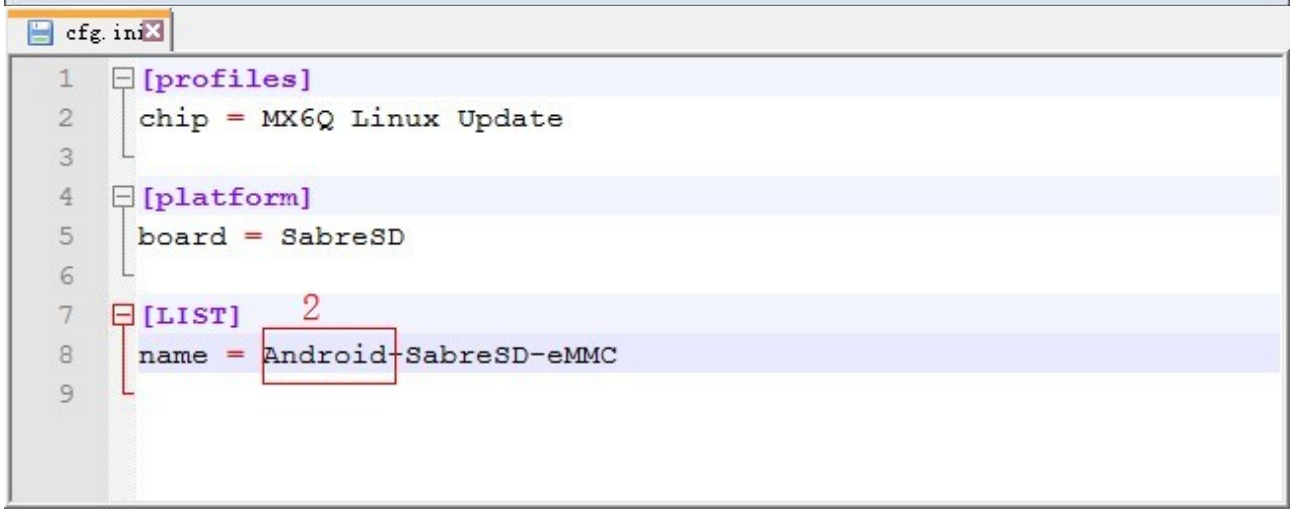
Burning tool Mfgtool	Run in the Windows, and save it in "Software\Windows tools"
Virtual terminal software	Such as Windows's own super terminal
Kernel Image	boot.img
u-boot Image	u-boot.bin
Recovery Image	recovery.img
File system	system.img
E9 mini PC	1pcs
USB OTG cable	1pcs
J5 jumper cape	1pcs
5V-2A DC Power Supply	1pcs

9.2.1 Burning Android System

- **Modify the config file**

1) Open the cfg file which is marked in red "1", Mfgtools-Rel-1.1.0_121218_MX6Q_UPDATER/cfg, use text to open it.

2) Change the “2” marked in red box to “Android” (The first letter "A" must be capitalized, as the system supports case-sensitive).



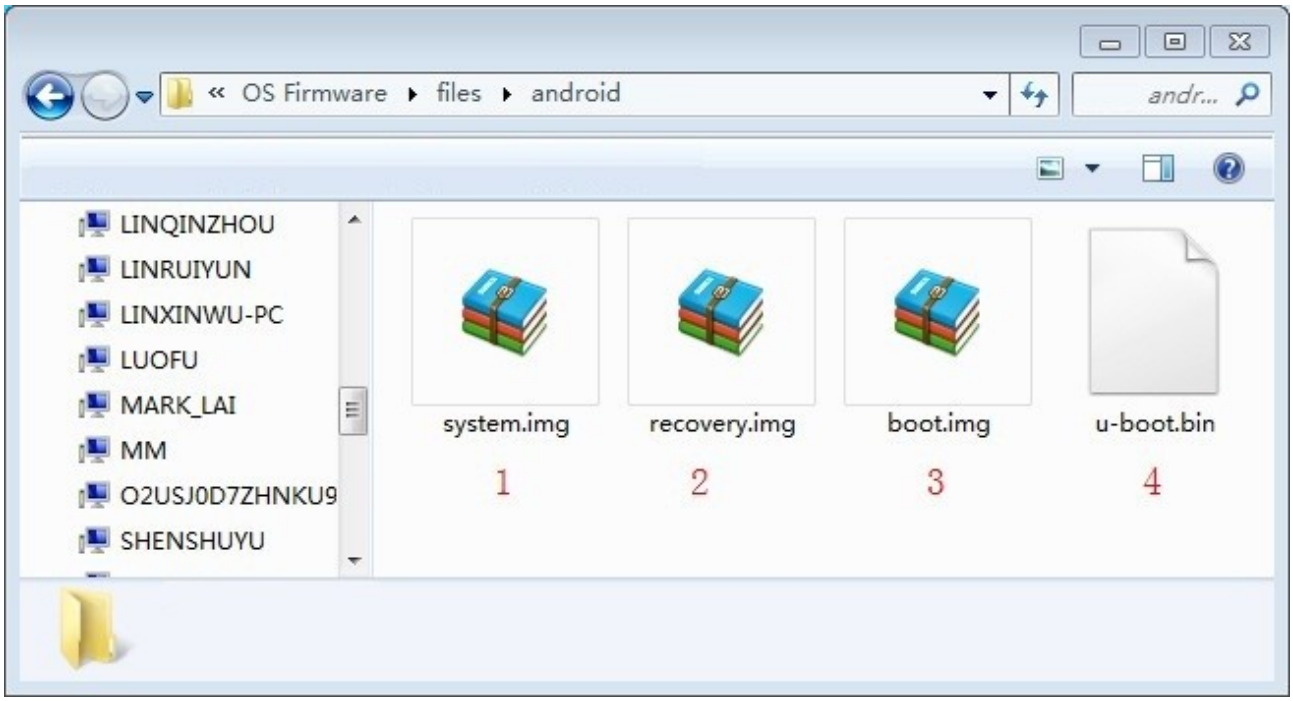
- **Copy the file**

Copy the Uboot image, bootimg image, system image and recovery image to Mfgtools-Rel-1.1.0_121218_MX6Q_UPDATER\Profiles\MX6QLinuxUpdate\OSFirmware\files\android directory (Or replace an existing file).

Note: The image above should be named as u-boot.bin (uboot image), boot.img (bootimg image), recovery.img (recovery image), system.img (system image), they cannot name as others, and so after the images are copied to MFG android directory, the “_E9_for_android_v2.0” should be removed.

If it is generated after compiled, then after the completion of the execution /build_sh, it will be out/target/product/sabresd_6dq/

After done, the file structure under the directory, Mfgtools-Rel-1.1.0_121218_MX6Q_UPDATER\Profiles\MX6Q Linux Update\OS Firmware\files\android is as showed below:



Check and confirm if all the above three files are copied successfully, all of them are indispensable.

9.2.2 Hardware Connection

(1) Connect the PC and E9 via USB OTG cable.

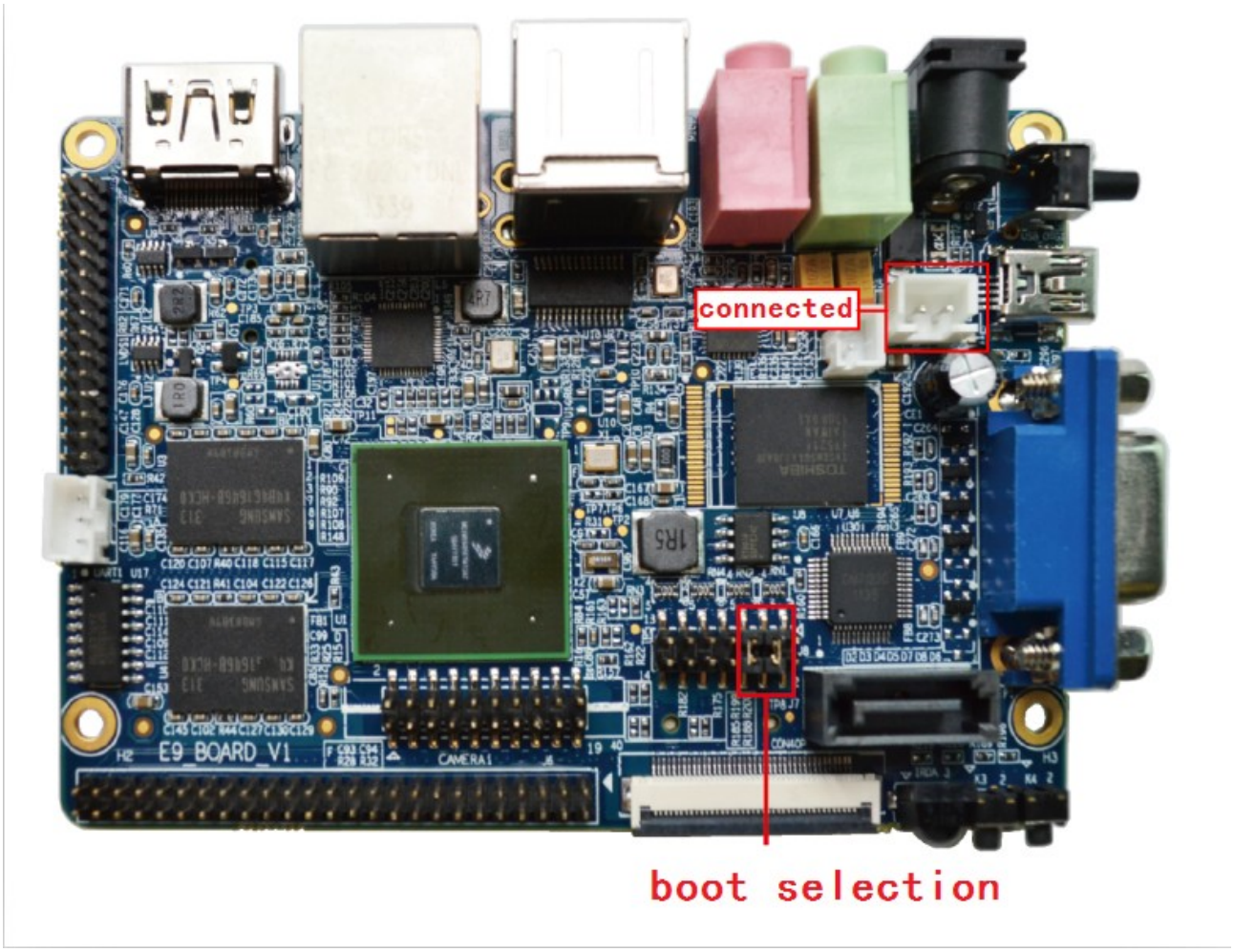
(2) Connect the 5V DC power supply to E9 5V jack

(3) Set the USB download mode as the picture shows: short the pin2 and pin3 (from right to left) with jumper cap, and then download it.

(4) Short / Connect the J5 spare external power supply interface (as marked below)

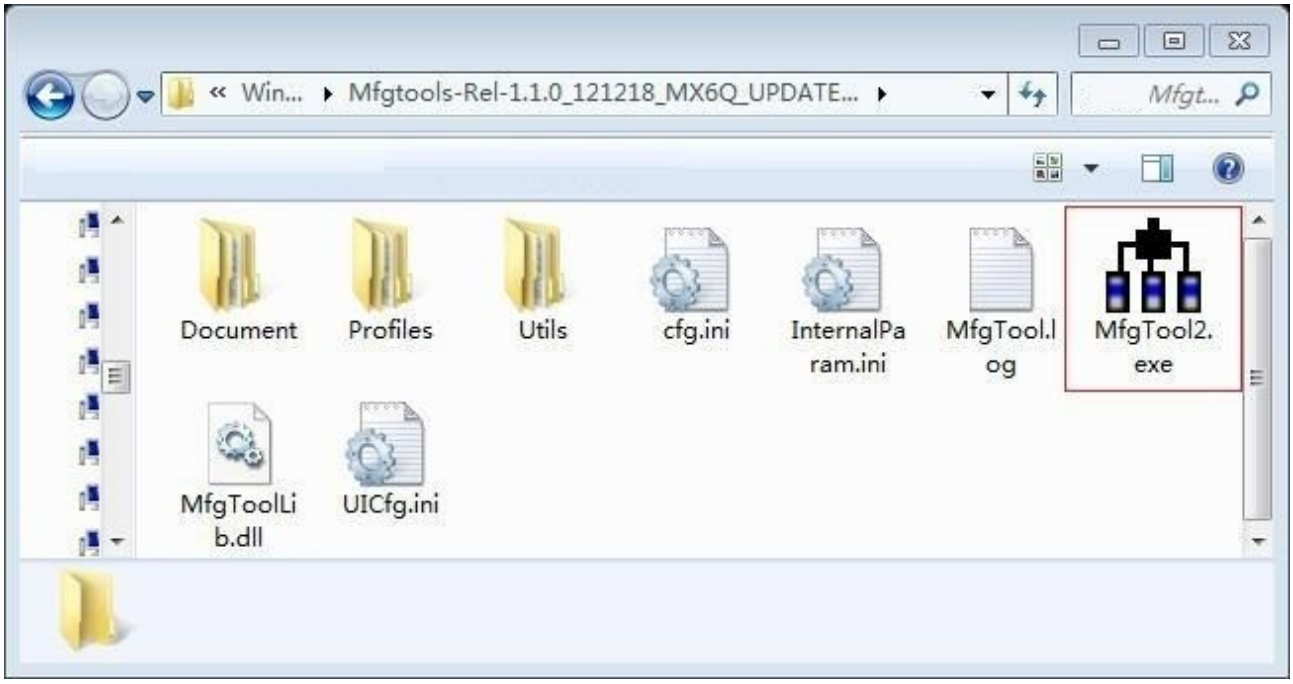
Note: J5 must be connected during burning. If it is not convenient to connect it, then you can keep pressing the power button in the whole process, from power on to the end of burning.

After the above three steps are done, E9 connections as shown below:

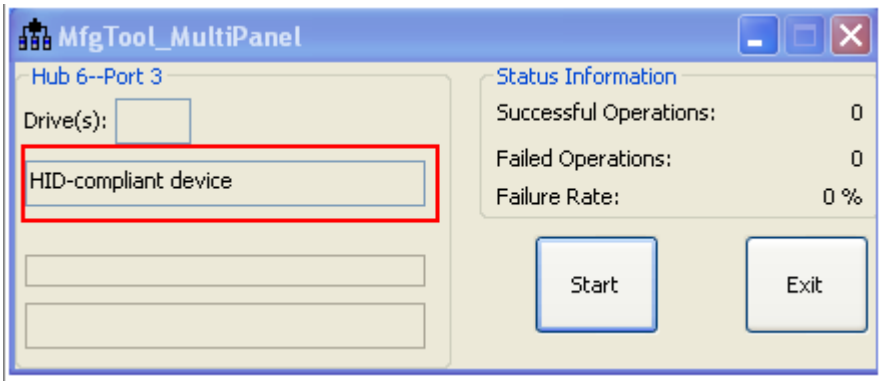


9.2.3 Begins to Burn

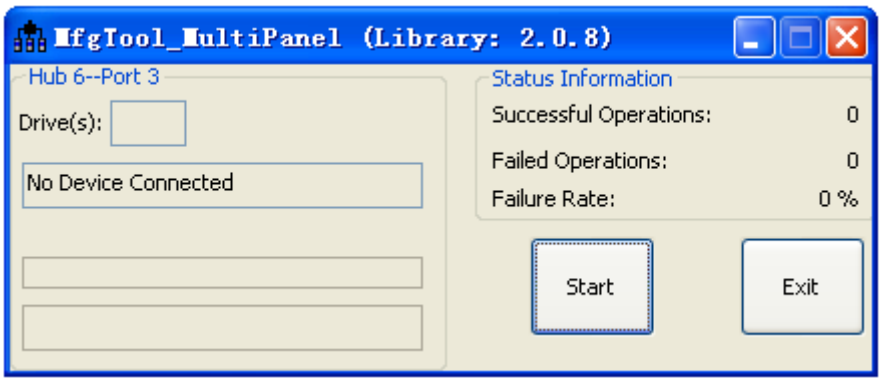
(1) Open the file “MfgTool2.exe” in the directory “Mfgtools-Rel-1.1.0_121218_MX6Q_UPDATER”:



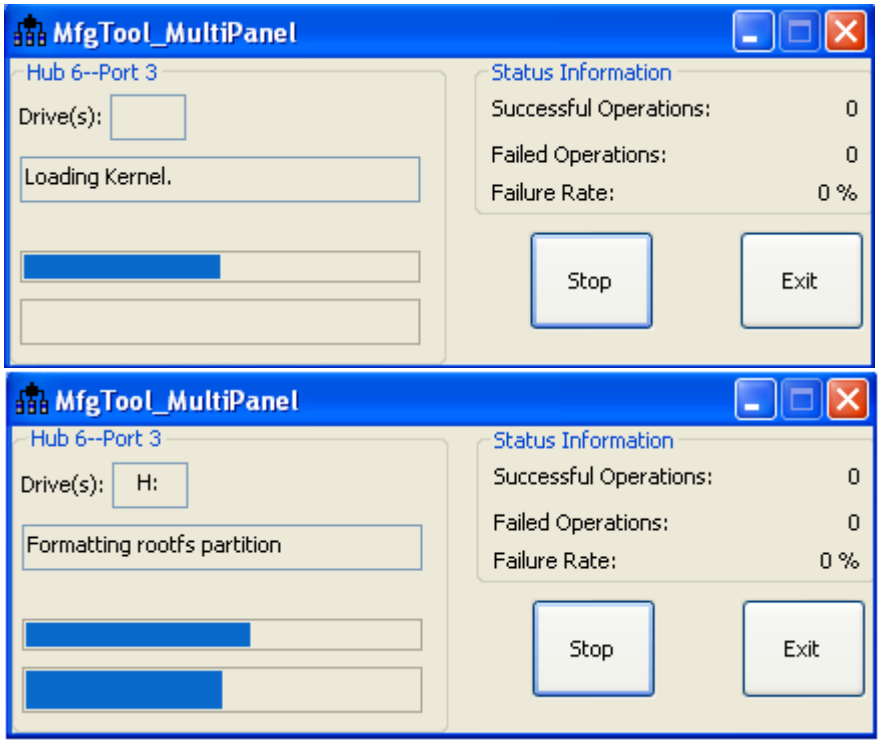
After open the MfgTool2.exe, it will detect the HID-compliant device, As shown below:



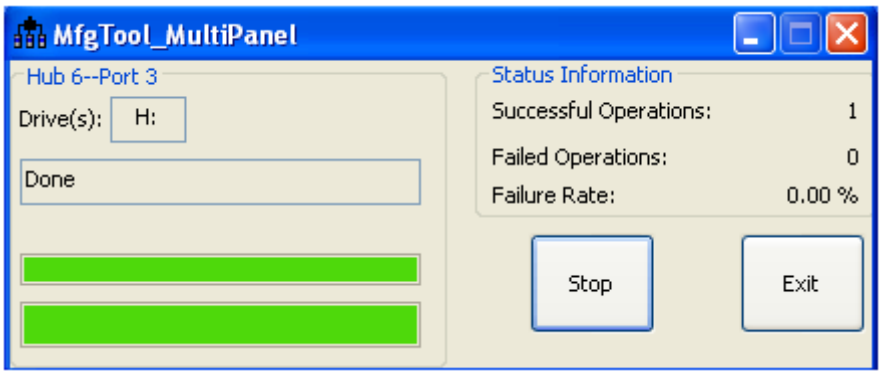
Note: If No Device Connected, please check the OTG connection, the USB connection is not normal.



(2) Click the “start” button in the software MfgTool2 to begin burning (The burning process may take about 20~30 minutes due to different computer configuration, please be patient!)



(3) After done as the following picture shows, click the “stop” button, turn off the power supply and Unplug the USB OTG cable. Then follow the tips in the following chapter and set it as emmc boot mode to run the system.



Note: After done, users need to reset the default boot mode (The default system is android) in the uboot config (see the tips in the chapter 4.4), then it can boot the system normally.

9.3 Linux Platform Burning

In the linux platform, it will be burned automatically. Please prepare the following things before burning:

Burning tools	Software\Linux tools\downloader.tar.bz2
Kernel image	boot.img
u-boot image	u-boot.bin
Recovery image	recovery.img
File System	system.img
E9 mini PC	1pcs
Tf card	1pcs

TF card reader	1pcs
5V-2A DC Power Supply	1pcs

9.3.1 Create SD Card

Extract the prepared related tools and images to any directory: #tar jxvf downloader.tar.bz2 -C

And related directory, and carry out the command: #./create_sdcard.sh -f /dev/sdc

Note: Here the example based on the writer's computer, SD card disk is the /dev/sdc, and your own computer will based on the disk after the " #fdisk –l", just avoid burning the program to your computer hard drive, damage to your data, and cause needless loss.

9.3.2 Prepare to Burn Image

SD card burning automatically is judged from the EmbedSky.ini contents in the images directory in the SD card root directory, and choose whether to burn or not and which images need to be burned. The file "EmbedSky.ini" must be put in the SD card images directory, if there is no such directory in the SD card, then please create a new directory (The directory name is images not image)

Note 1: The file format cannot be changed casually, and there is no space behind =, nor can increase or delete a row. It is strongly recommended to use EmbedSky.ini, and modify on it.

Note 2: The "/ / XXX" below is for interpreting only and when using the file, please don't add these comments, or else it will result in an error.

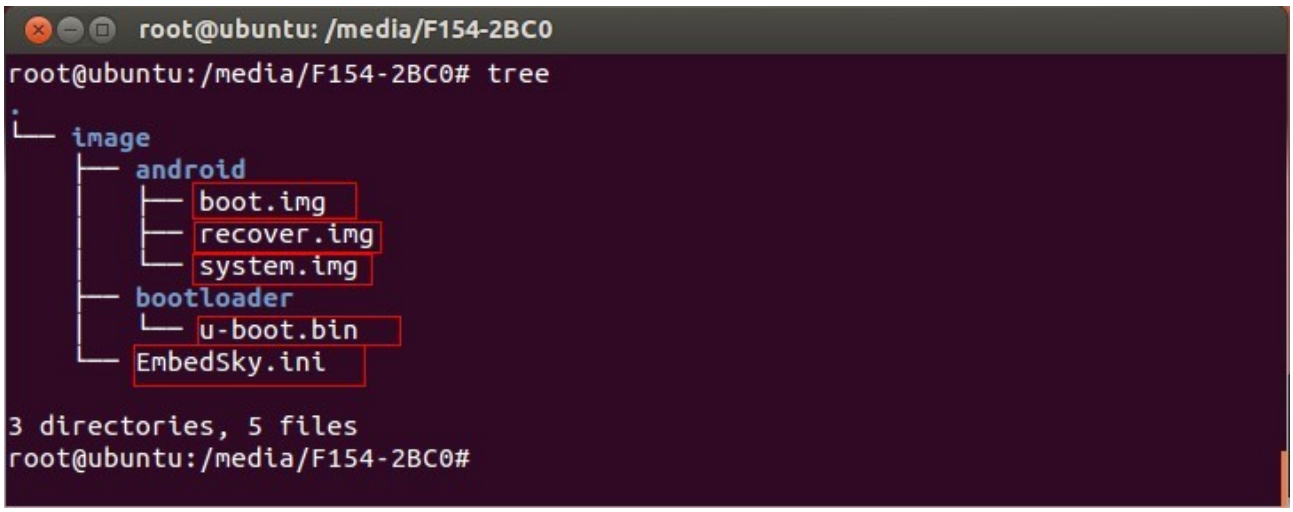
Note 3: When users burning the android, it needs to pay attention to the comments, and modify it according to actual situation, no need to change others.

Note 4: For those images which don't need to update, it can be set as null, such as "boot_img="

```
#### This is E9 SD config files ####
operating condition=install
OS Type=android                    // operating system type
LED indicator=off
Beep=off
extern_indication=partition        // external operation tips
#### Others ####
bootloader=images\bootloader\u-boot.bin          //u-boot     image storage path
LOGO=
#### Others end ####

#### Linux ####
kernel=images\
rootfs=images\
#### Linux end ####
```

```
#### Windows CE ####
NK=images\wince\NK.bin
#### Windows CE end ####

#### Android ####
boot_img=images/android/boot.img                    //android boot.img storage path
system_img=images/android/system.img                //android system.img storage path
recovery_img=                                       //android recovery.img storage path
#### Android end ####

#### Ubuntu ####
kernel=images/ubuntu/uImage
rootfs=images/ubuntu/oneiric.tgz
#### Ubuntu end ####

#### uCOS-II ####
uCOS-II Image=
#### uCOS-II end ####

#### User Application ####
Application Image=
#### User Application end #####
```

Corresponding burning file should be:



9.3.3 Burning System

J8 is the E9 mini PC BOOT mode selection interface. Shorting different jumper cap will have different boot modes. It is 2.00mm pitch and 10Pin interface and it needs to work with 2.00mm pitch jumper cap.

As it is used SD card to burn, then it needs to boot via SD card. Please short the pin 1 and pin 7 with jumper cap and put the tf card to the TF slot, long press the power button, then the system boots. Then the system burning will automatically finished according to the EmbedSky.ini contents.

(The whole process may take about 5 minutes, which will be different from computer to computer, please be patient!)

When the burning is complete, the power indicator will be off automatically.

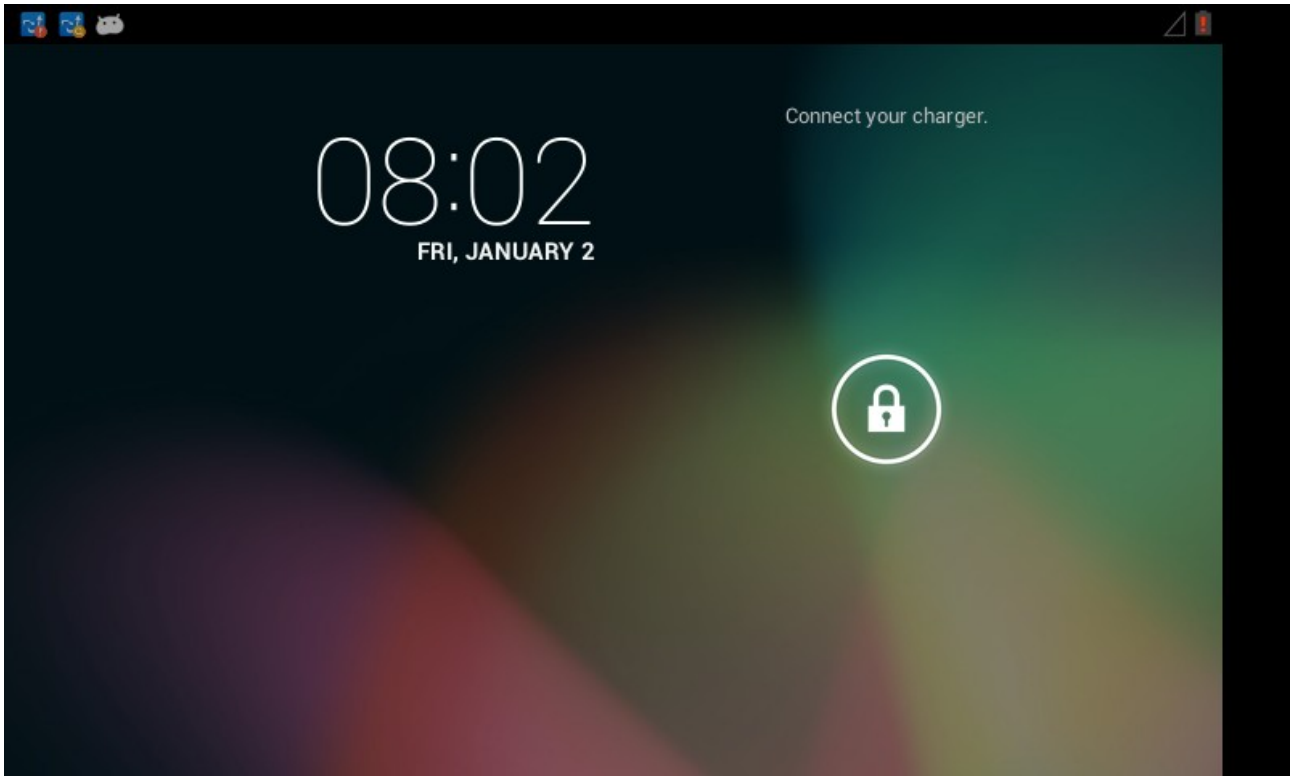
Note: After done, users need to reset the default boot mode (The default system is android) in the uboot config (see the tips in the chapter 4.4), then it can boot the system normally.

Chapter Ten Android Testing

The following test is used the 1280*768 resolution output by FPC->VGA, if users use other resolution (such as 800*400 output by FPC), then there may be slightly different, but it does not affect the use at all.

Set the E9 boot mode as EMMC by jumper cap, then power up and long press the power button. The red power light near the power will always be bright (if press it only but not long press, the power is not bright at all), then it means that the board uboot is already run up. Users can enter uboot and make related settings to wait for the system boot up automatically.

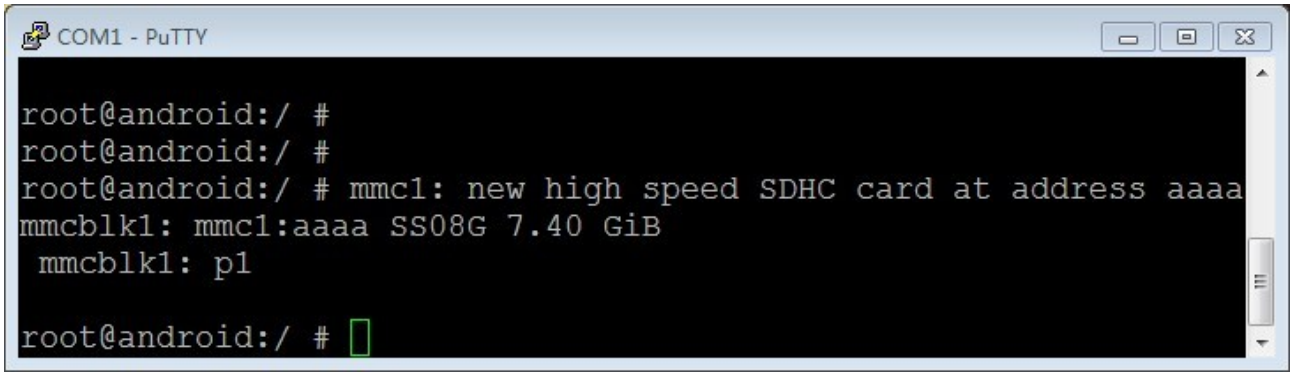
After the system has been booted, then you will see it as the following picture shows, slide to unlock and enter to the system. As the testing is almost the same as the Android phone/tablet, here we will introduce briefly.



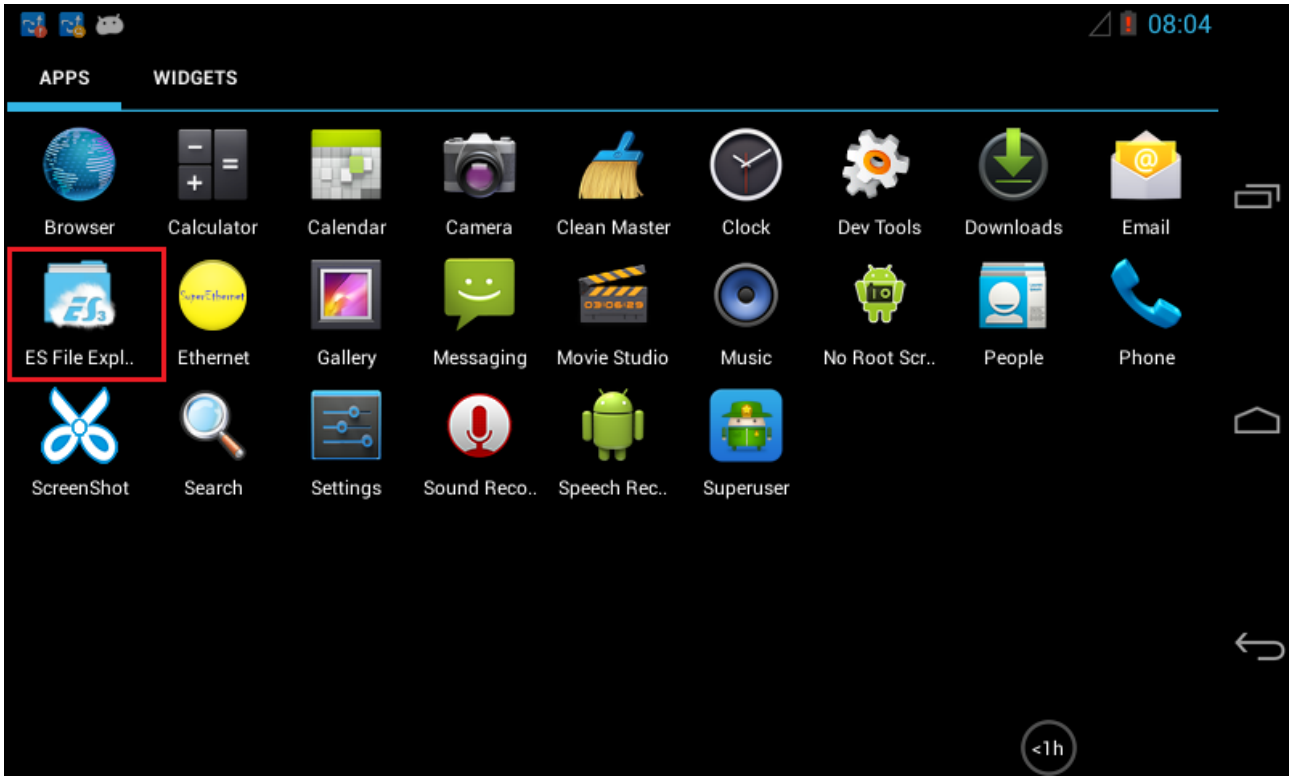
10.1 TF Card Testing

TF card supports hot plug, so you can insert TF card before or after the system boot, the following test is taken when the TF card is inserted after the system is booted. After inserted, the system detects TF card, then it will mount automatically. The following is the serial printing information when mounting:

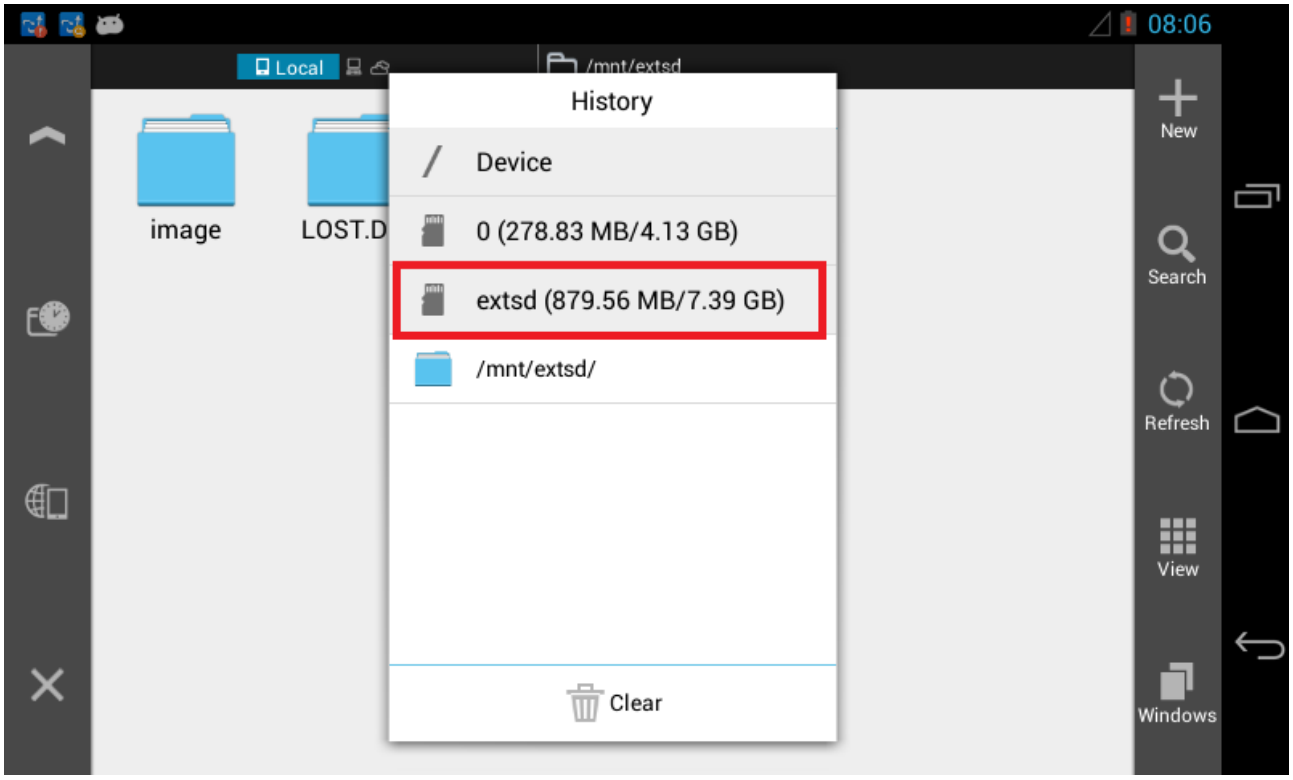
Note: The serial print information will be different due to the different TF card; here it is for reference only.



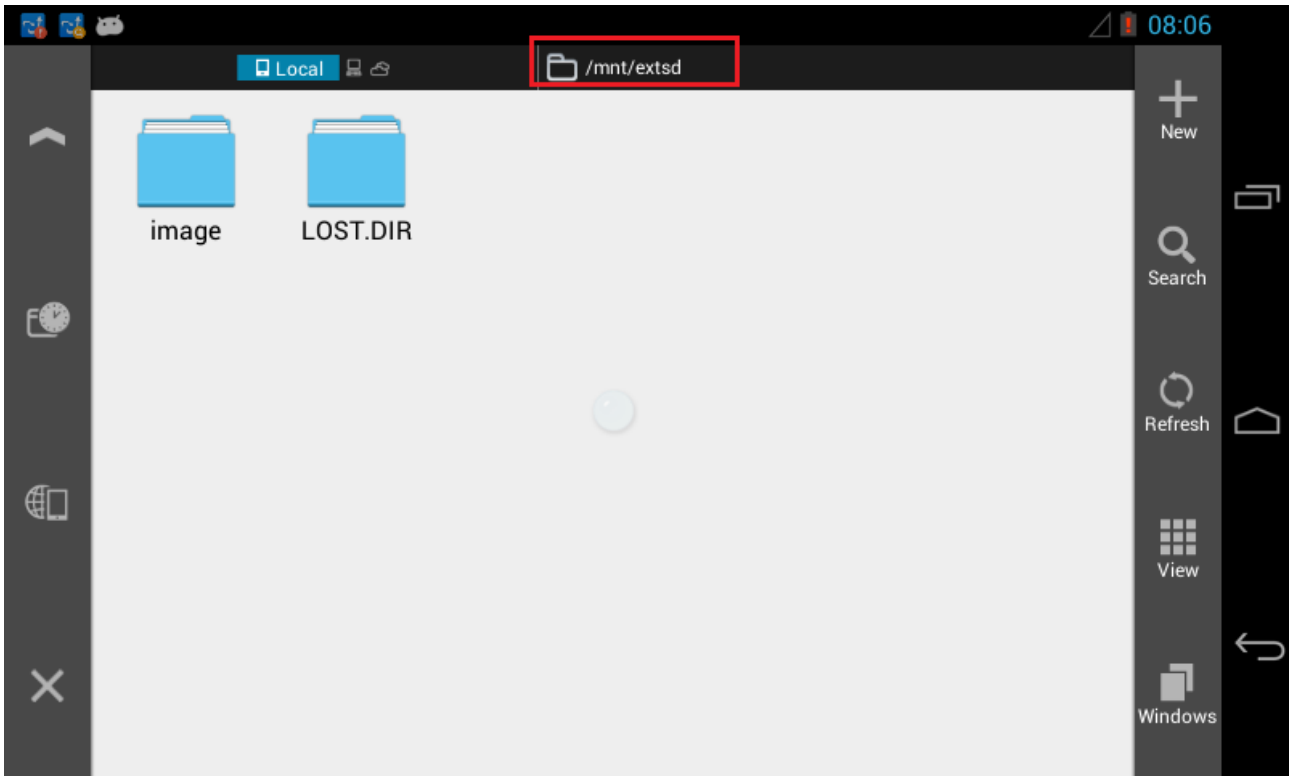
The mounted directory TF uses is /mnt/ext_sd, please use ES File Explorer to see the contents inside. Open the ES file manager:



Select the collection, then choose /mnt/extsd directory

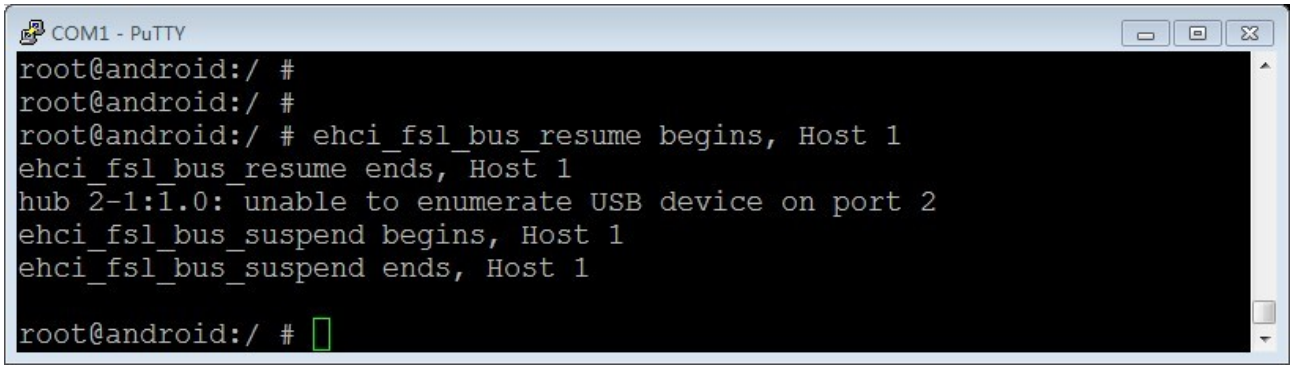


You can see the contents in the TF card.



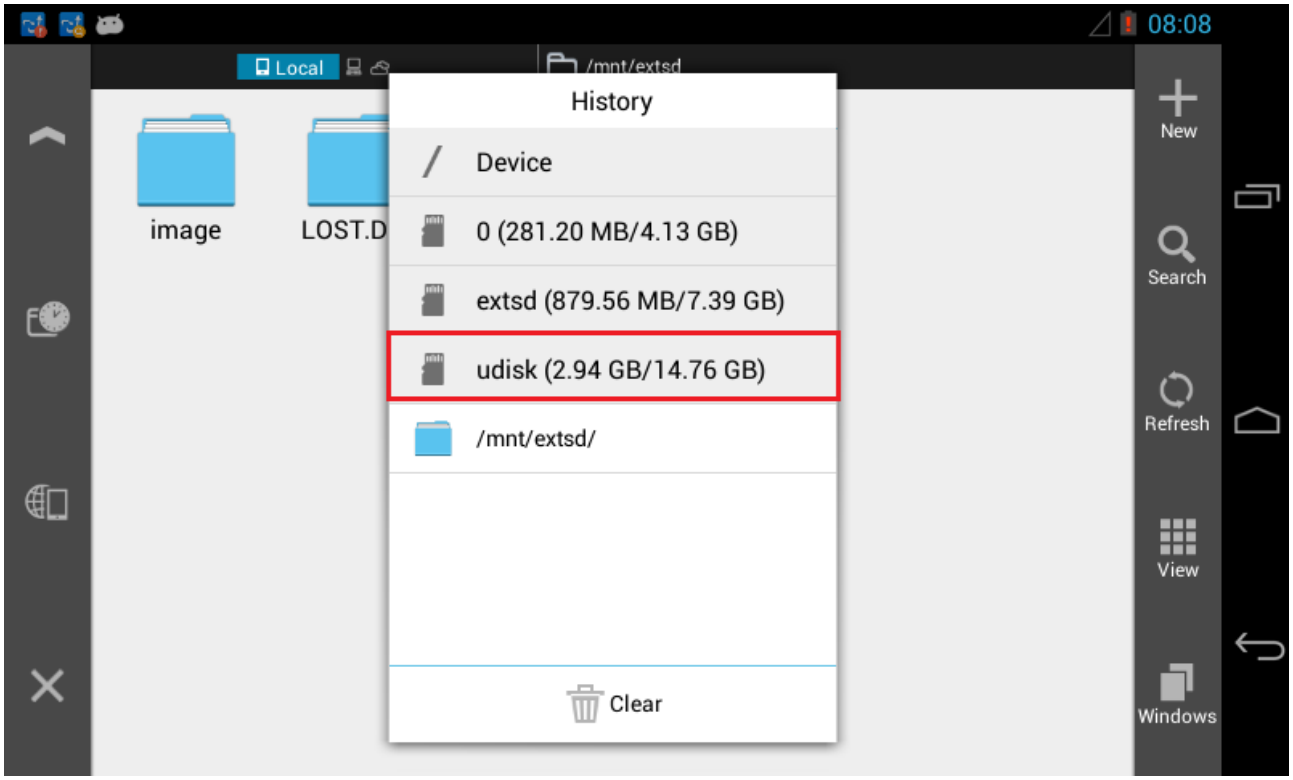
10.2 U Disk Testing

The U disk use is almost the same as TF card use except the mount points. Inserted the U disk into the system, then Android will mount automatically. The following is mounted serial printing information:



Note: The serial print information will be different due to the different TF card; here it is for reference only.

Use the file manager to see the content inside. Different from SD card, here we use the mnt/udisk mounted directory.



Open it then you can see the U disk contents.

10.3 SATA Testing

SATA does not support hot plug. The SATA hard disk must be inserted before the system is booted. And it supports NTFS format. After the system is booted, please use the tool ntfs-3g to mount. It uses the directory /mmt/shm to mount in the test.

#ntfs-3g /dev/block/sda1 /mnt/shm/

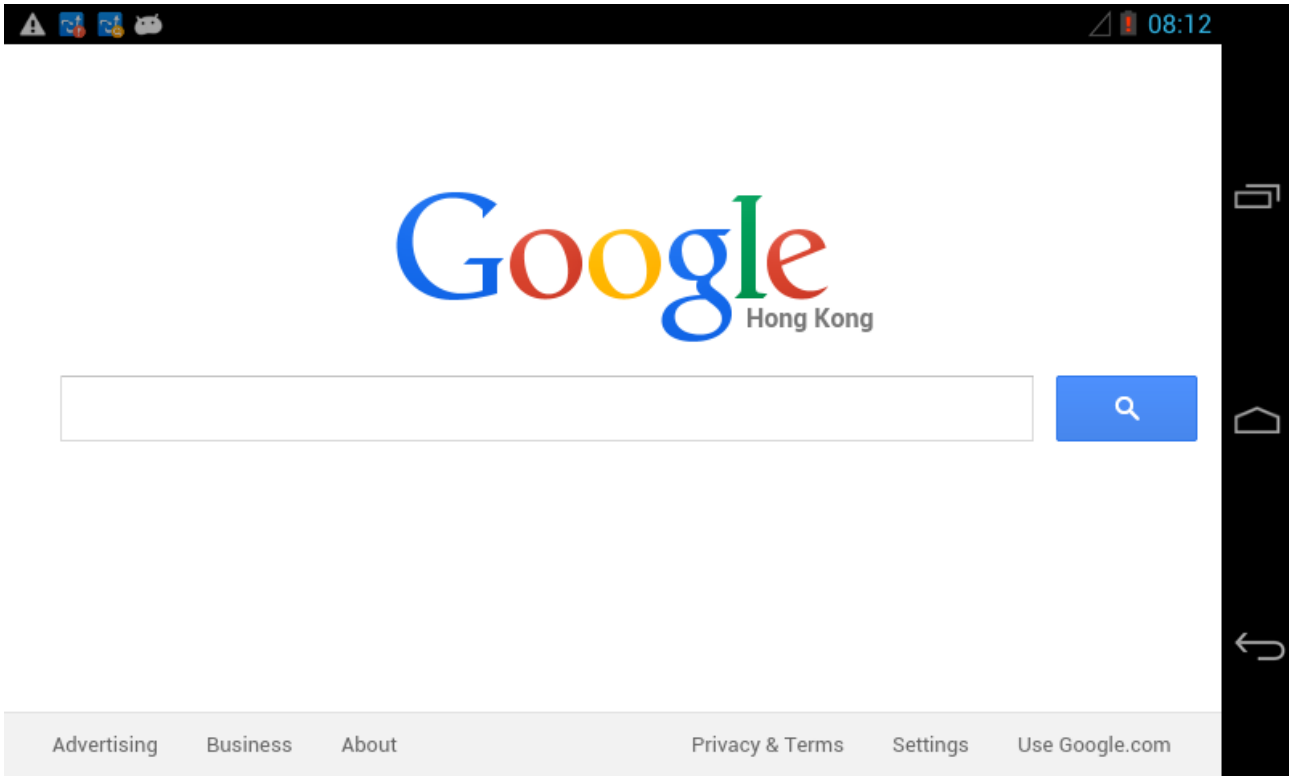
Open the directory to see the files in SATA hardware.

10.4 Network Card Testing

The Network uses the DHCP to assign IP automatically. After the system is booted, then connect the Router then it can surf the Internet directly. Users can also modify IP by entering the commands in serial port:

```
ifconfig eth0 down
ifconfig 1926.128.xxx.xxx up
```

Open the browser, you can access to the Internet via wired network.

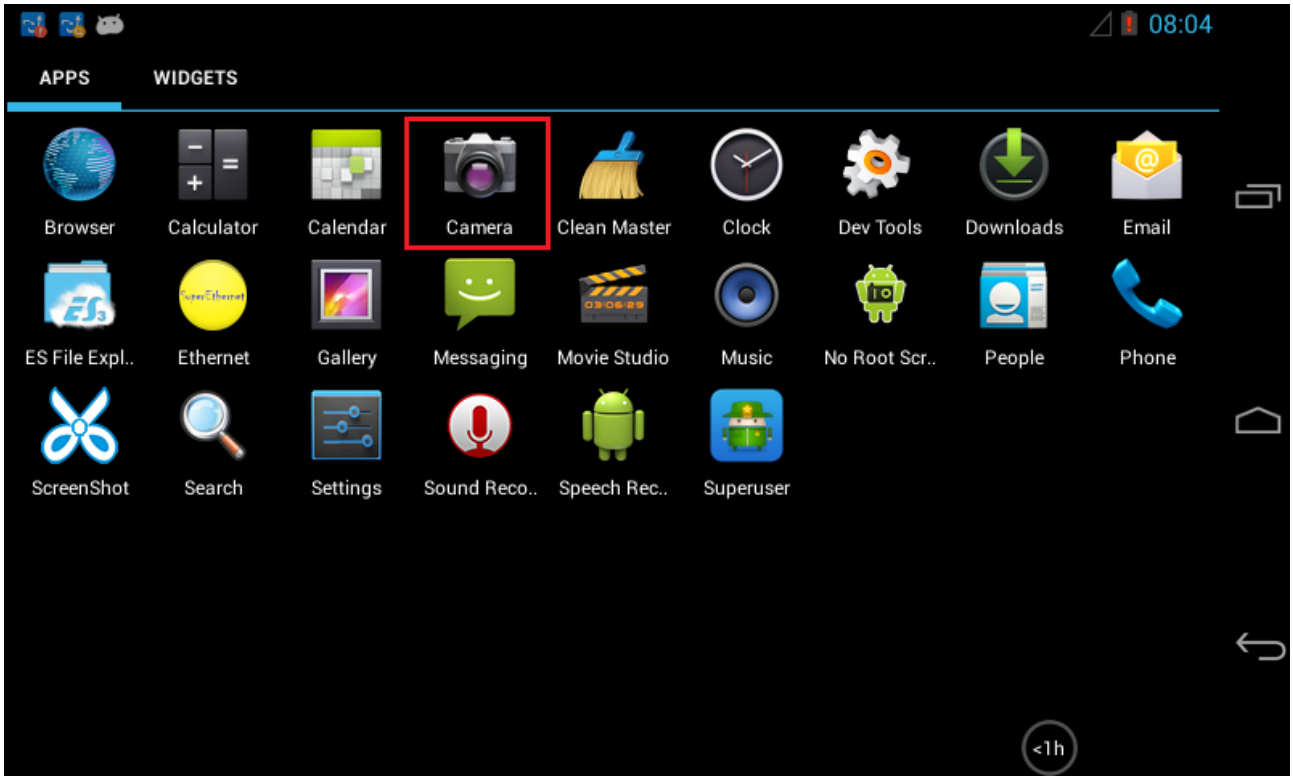


10.5 Camera Testing

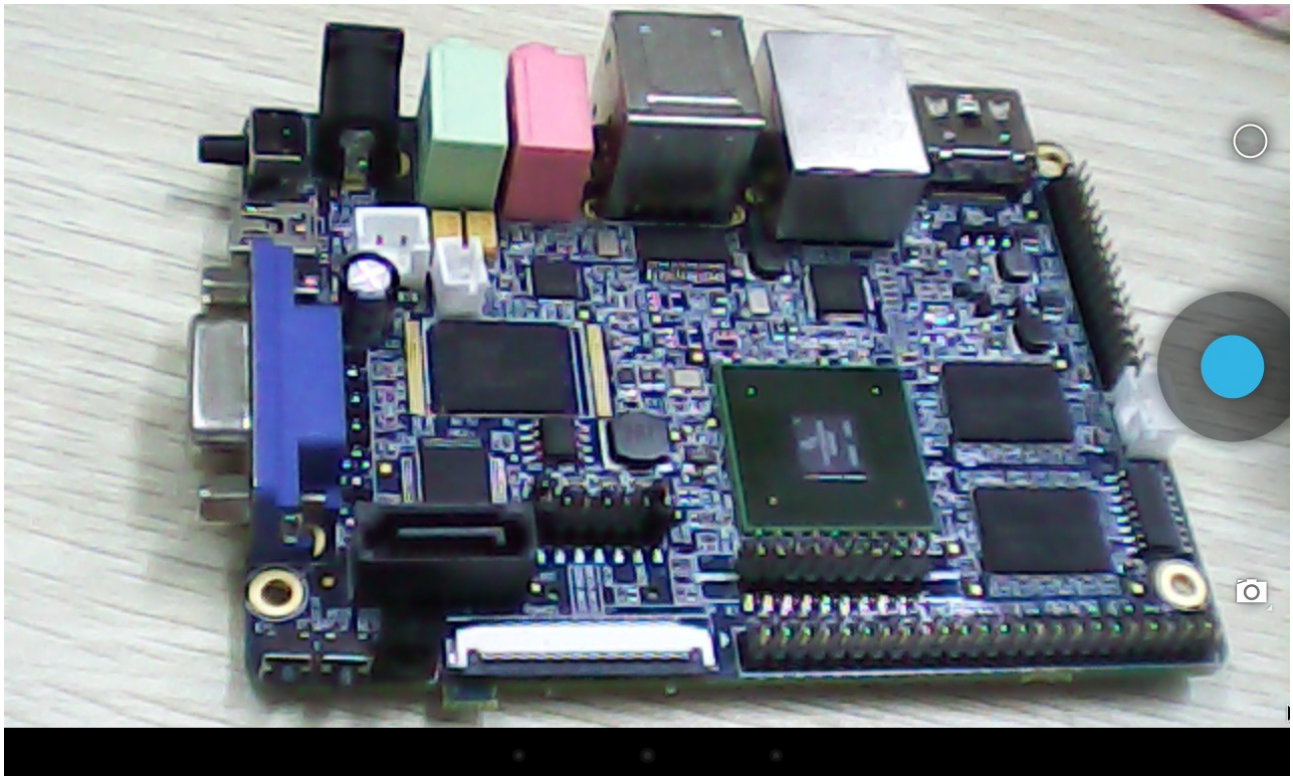
The SD card mounted point that E9 uses is emmc inner space. So it will not be prompted that SD card is not inserted. E9 supports USB camera and CMOS camera, and they can switch to front and rear camera each other.

Please plug one camera at least before testing, USB camera supports hot plug, but CMOS camera does not. The CMOS camera may cause to be burned if hot plugged. Currently, the E9 board supports CMOS camera OV3640. And it will support more cameras as the system upgrade.

Open the camera test program:



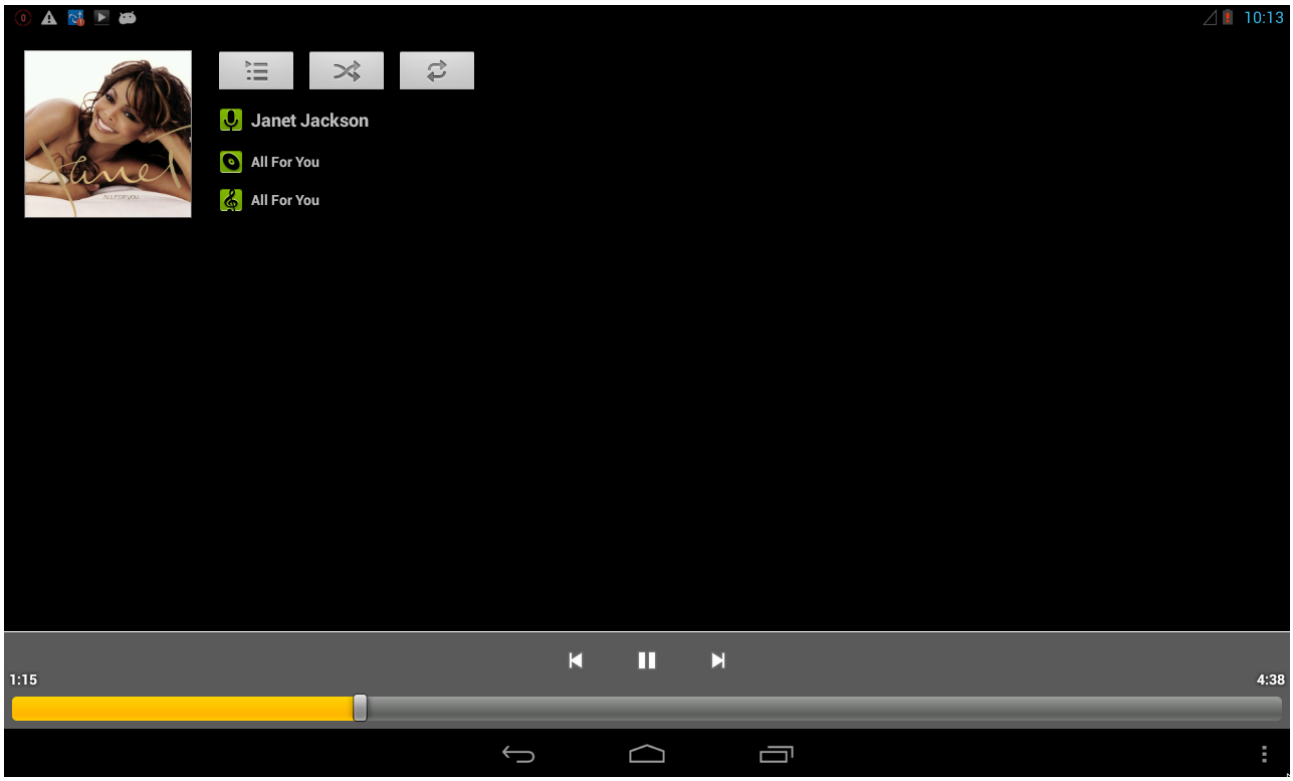
You can see the captured image.



10.6 Audio Testing

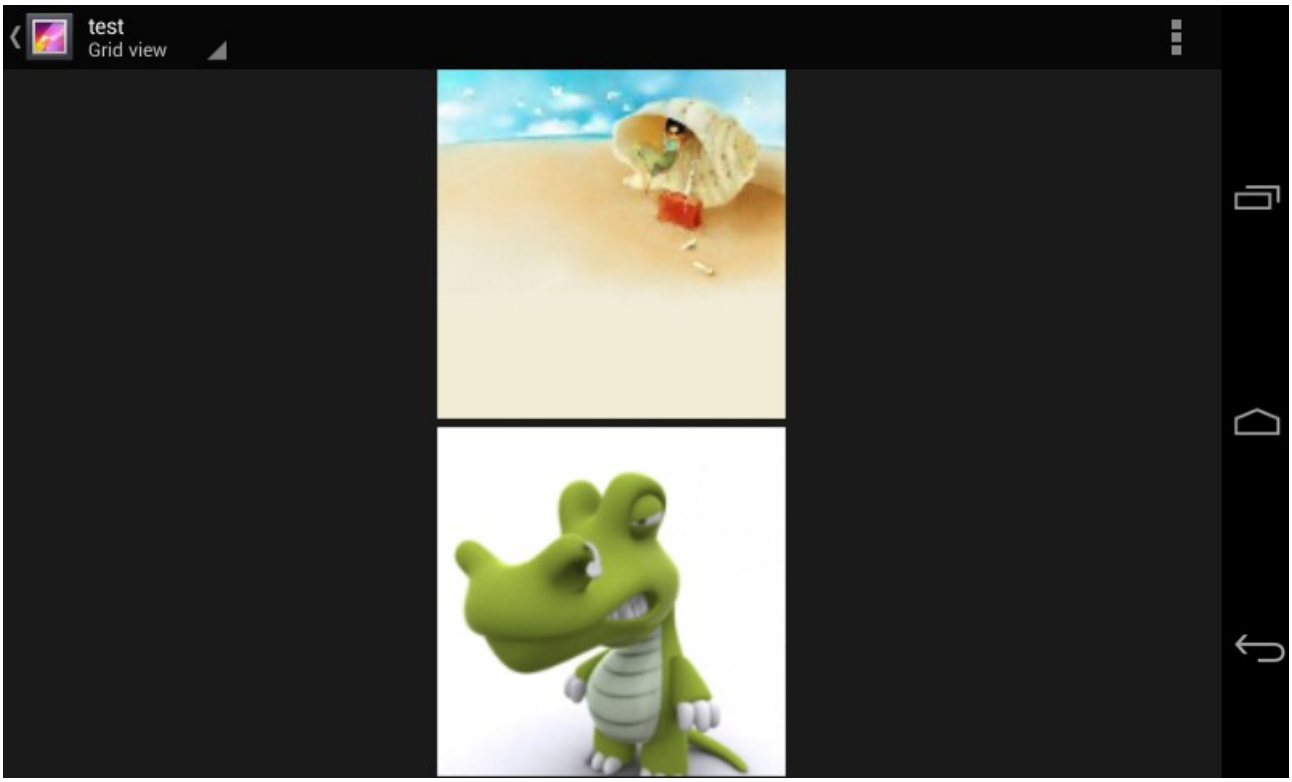
The E9 Audio Card does not support recording currently. It will be updates in the future. So here we will play sound only.

Open an audio file with the file manager, the default player will be OK, and then you can hear it.



10.7 Video Testing

Just as the Audio Testing, open a video in the video player, then it can finish the test. Currently, it supports mp4, h264 and other hardware solution high-definition video.



10.8 HDMI Testing

HDMI uses 1080P resolution by default. If users want to use HDMI, then they need to do some settings in uboot before it can be used normally. HDMI supports hot plug, users can insert into the HDMI before or after power on.

Note: HDMI uses its own audio output, no audio output in the board headphone and speaker when using HDMI.

Linux

Introduction

Note: In the following operation, the "#" before the commands indicates a root user authority to use. It is recommended to use the root user to login Linux PC, if not, please uses the sudo command to gain root user privileges, or it may fail to compile.

Chapter Eleven Linux Source Code & Compile

11.1 Linux driver Source Code

Linux kernel version is kernel_3.0.35, the following paths are relative to the kernel root directory. As the driver code involves a lot, and then it is difficult to introduce them one by one, so the driver source code mentioned below is only for the important ones and those which are used in high frequency.

driver	Location	Device name
emmc/sdcard driver	drivers/mmc/host: sdhci-esdhc-imx.c sdhci.c sdhci-platform.c sdhci-pltfm.c	/dev/mmcblk*
LCD&LVDS driver	drivers/video/mxc/: ldb.c mxc_dispdrv.c mxc_dvi.c mxc_ipuv3_fb.c mxc_lcdif.c	/dev/fb*
capacitive touch screen driver	drivers/input/touchscreen： gt811.c	dev/input/event*
resistive touch screen driver	drivers/input/touchscreen： tsc2007.c	dev/input/event*
USB OTG driver	drivers/usb/otg： fsl_otg.c	
cable network driver	drivers/net： fec.c	eth0
serial port driver	drivers/tty/serial： imx.c mxc_uart_early.c serial_core.c	/dev/ttySAC*
WM8960 audio driver	sound/soc/imx/*	/dev/snd/*
I2C driver	drivers/i2c/busses: i2c-imx.c	
keypad driver	drivers/input/keyboard： gpio_keys.c	dev/input/event*

Irda driver	drivers/input/keyboard： tq_hs0038.c	dev/input/event*
backlight driver	drivers/video/backlight/： pwm_bl.c backlight.c	
pwm driver	arch/arm/plat-mxc/： pwm.c	
RTC driver		/dev/rtc0
Watch dog driver	drivers/watchdog： imx2_wdt.c	/dev/watchdog
Multimedia video camera driver	drivers/media/video/mxc/capture/*	/dev/video*
sata driver	drivers/ata： ahci_platform.c libahci.c	/dev/sdx

11.2 Linux Source Code Compile

11.2.1 Compile Bootloader

- **u-boot source code extract**

Linux+QT and ubuntu system use the same uboot, and uboot source code named uboot_E9_1.3.4_for_ubuntu.tar.bz2, please cope it and follow the chapter 5.2.1 to compile it.

Note: If the computer configuration is relatively high, then it can add a few threads when compiling, such as make-j4 or-j8 to speed up compilation.

11.2.2 Linux kernel compile

- **Extract Kernel Source Code**

Extract linux Kernel Source Code (\Source Code\Linux Source Code\linux Kernel, with the name linux_E9_3.0.35_for_linux_v2.0.tar) to PC linux system root directory (Here the writer of this manual choose root directory to operate and take it a screenshots, actually, any directory will be OK), then use the command:

```
#tar  xvfj  linux_E9_3.0.35_for_Linux_v2.0.tar.bz2  -C  /
```

Extract the source code, as shown in the figure below:

```
root@ubuntu: /opt/Embedsky
root@ubuntu:/opt/Embedsky# ls
linux_E9_3.0.35_for_linux_v2.0.tar.bz2   uboot_E9_1.3.4_for_Ubuntu_v2.0.tar.bz2
linux_E9_3.0.35_for_Ubuntu_v2.0.tar.bz2  ubuntu_env_install.sh
Tools                                    ubuntu_env_install.tar.bz2
TQIMX6
root@ubuntu:/opt/Embedsky# tar xvfj linux_E9_3.0.35_for_linux_v2.0.tar.bz2 -C /
opt/Embedsky/TQIMX6/E9/Linux/linux_E9_3.0.35_for_Linux/
opt/Embedsky/TQIMX6/E9/Linux/linux_E9_3.0.35_for_Linux/fs/
opt/Embedsky/TQIMX6/E9/Linux/linux_E9_3.0.35_for_Linux/fs/binfmt_som.c
opt/Embedsky/TQIMX6/E9/Linux/linux_E9_3.0.35_for_Linux/fs/partitions/
opt/Embedsky/TQIMX6/E9/Linux/linux_E9_3.0.35_for_Linux/fs/partitions/sgi.h
opt/Embedsky/TQIMX6/E9/Linux/linux_E9_3.0.35_for_Linux/fs/partitions/atari.h
opt/Embedsky/TQIMX6/E9/Linux/linux_E9_3.0.35_for_Linux/fs/partitions/sun.c
opt/Embedsky/TQIMX6/E9/Linux/linux_E9_3.0.35_for_Linux/fs/partitions/efi.h
opt/Embedsky/TQIMX6/E9/Linux/linux_E9_3.0.35_for_Linux/fs/partitions/efi.c
```

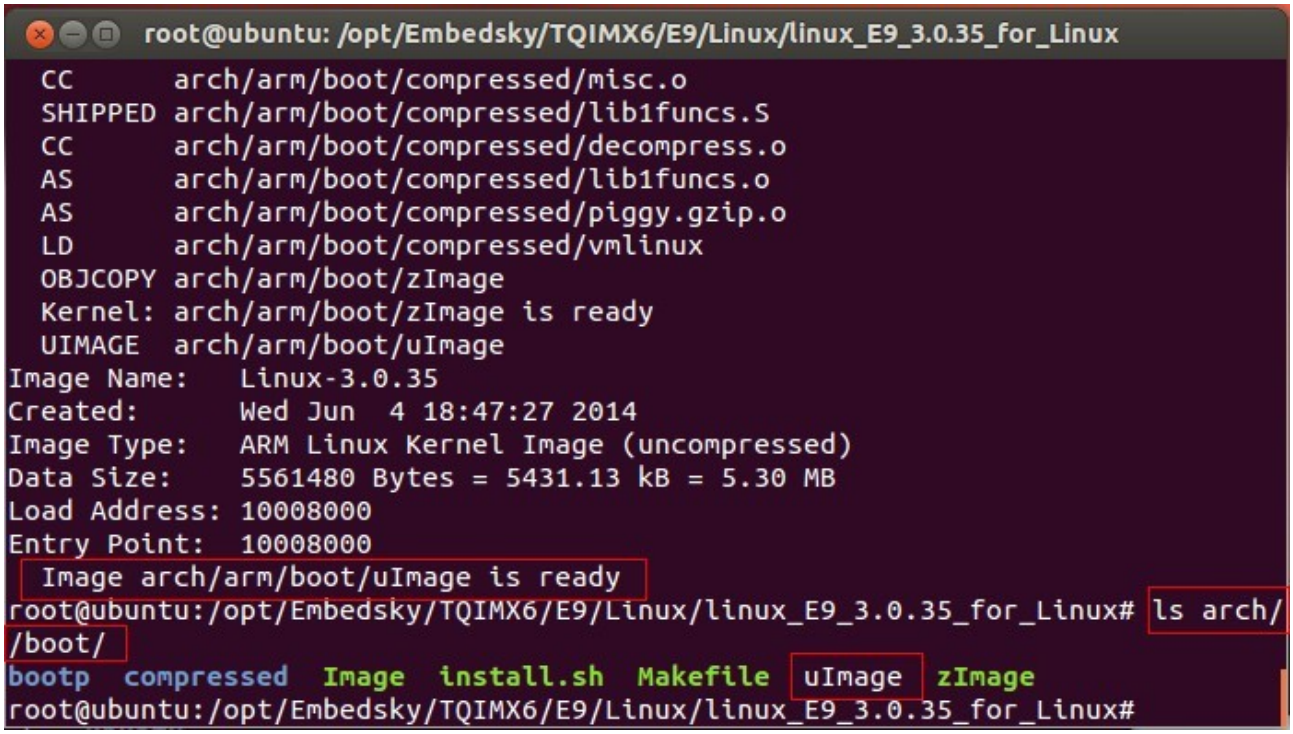
- **Kernel Source Code Compile**

After extract the source code, please use the command: #. build.sh

And compile it as follows:

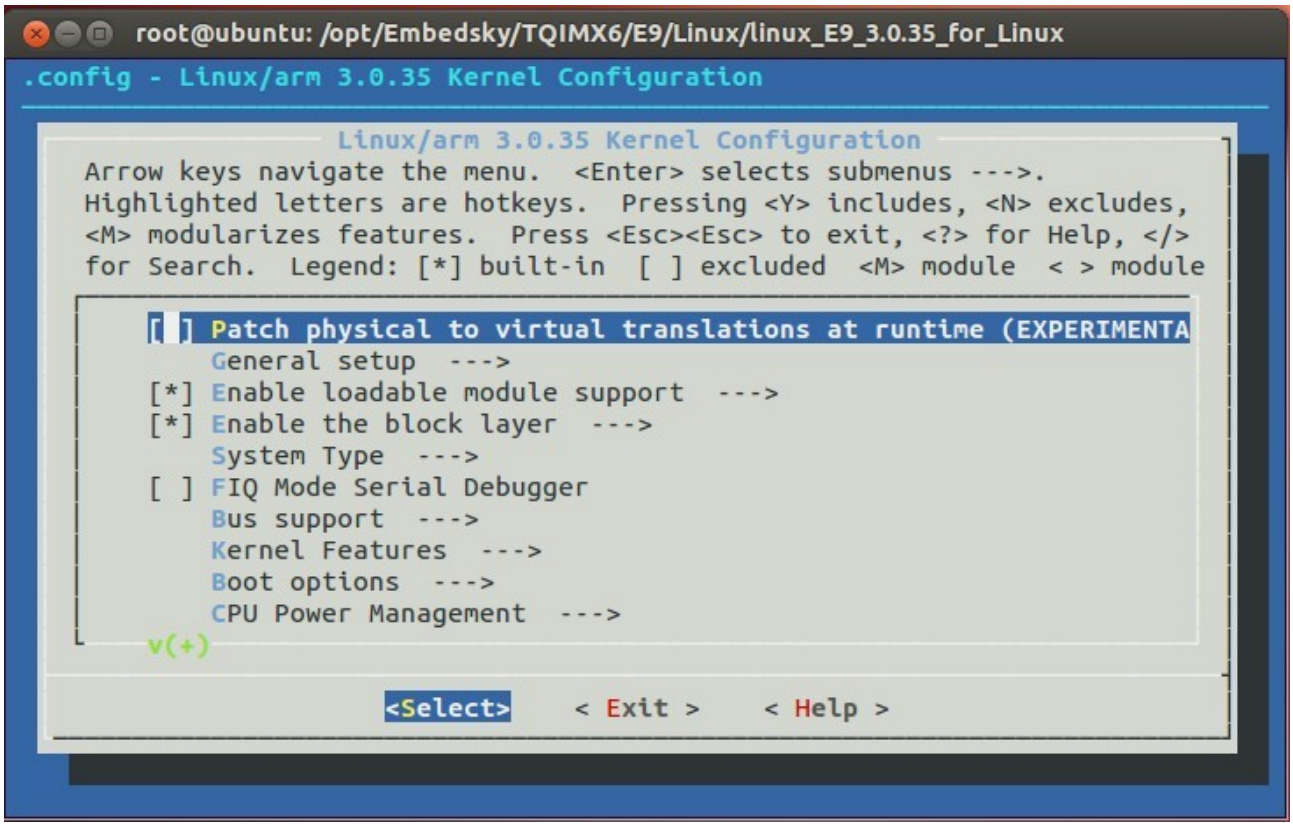
```
root@ubuntu: /opt/Embedsky/TQIMX6/E9/Linux/linux_E9_3.0.35_for_Linux

root@ubuntu:/opt/Embedsky/TQIMX6/E9/Linux/linux_E9_3.0.35_for_Linux# ls
arch             COPYING        firmware  Kbuild       Makefile  scripts   virt
block            CREDITS        fs        Kconfig      mm        security
build.sh         crypto         include   kernel       net       sound
build.sh~        Documentation  init      lib          README    tools
config_e9_linux  drivers        ipc       MAINTAINERS  samples   usr
root@ubuntu:/opt/Embedsky/TQIMX6/E9/Linux/linux_E9_3.0.35_for_Linux# . build.sh
CPU number is 12
scripts/kconfig/conf --silentoldconfig Kconfig
  CHK     include/linux/version.h
  CHK     include/generated/utsrelease.h
make[1]: `include/generated/mach-types.h' is up to date.
  HOSTCC  scripts/recordmcount
  CC      kernel/bounds.s
  CC      scripts/mod/empty.o
  HOSTCC  scripts/genksyms/lex.o
  MKELF   scripts/mod/elfconfig.h
  HOSTCC  scripts/mod/file2alias.o
  HOSTCC  scripts/mod/modpost.o
  HOSTCC  scripts/mod/sumversion.o
  HOSTLD  scripts/genksyms/genksyms
  GEN     include/generated/bounds.h
  CC      arch/arm/kernel/asm-offsets.s
```

After compiled, you will find the image named “uImage” in the directory \e9_cd\Image\Linux\Kernel, as the following two figures show:



Then you can test in the board, as the following shows:

If you want to modify the default configuration, please use the command #make menuconfig to enter the configuration interface, then you can modify the configuration according to the actual needs, and save the changes after done. Here are the steps:



After the configuration, please run #make uImage to recompile it.

11.2.3 Linux File System

E9 linux file system is the complete system which can be founded in “Image\Linux\File System” directory.

Chapter Twelve Linux System Burning

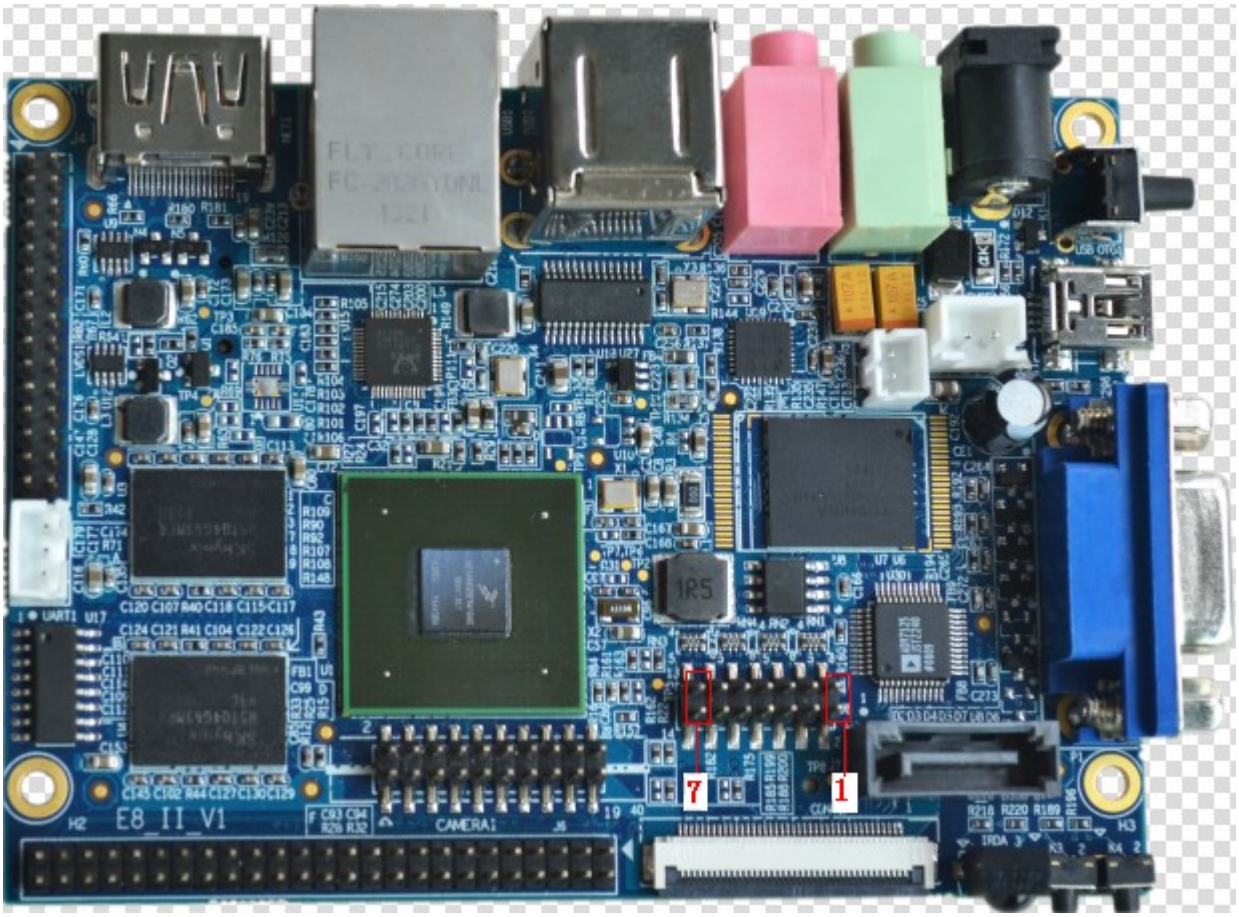
12.1 Preparations

E9 provides two boot modes: boot from the SD card and emmc respectively. Boot mode selection (eMMC, SD can be chose from the onboard SW6 DIP switch. Here we will introduce the DIP switch status and boot modes.

12.1.1 Boot Modes Selection

J8 is the E9 mini PC BOOT mode selection interface; Short different jumper cap will have different boot modes. It is 2.00mm pitch and 10Pin interface and it needs to work with 2.00mm pitch jumper cap.

Here we will use download mode, please short the pin 2 and 3 first.



E9 Mini PC keeps all the boot modes, and it uses the following 3 boot modes temporarily. (In the above picture, the pins, from right to left, are 1, 2, 3, 4, 5, 6, 7)

Boot select table							
Boot device	7	6	5	4	3	2	1

	Bt-cfg2_3	Bt-cfg2_4	Bt-cfg2_5	Bt-cfg2_6	Bt-cfg1_4	Bt-cfg1_5	Bt-cfg1_6
emmc	1	1	1	X	X	1	1
sd	1	0	0	X	x	0	1
Download	0	0	0	0	1	1	0

Note: If the Ethernet upwards, then the J8 interface, from left to right, the pins are 7, 6, 5, 4, 3, 2, 1, So corresponding to the above table, short the Jumper Cap is marked as “1”;” X” means it can connect and it can also be open.

12.1.2 Burning the Image

There are two ways for update the E9 firmware: burning via SD card or burning by MFG-tools.

SD card burning mainly provides a convenient and efficient firmware updated way for customers in the ubuntu system when developing. And it also applies to burn in a one-to-many way in the mass production to increase production capacity and save labor costs.

MFG-tools is a kind of burning tool for E9 under PC environment, which supports ubuntu 12.04 burning, simple, convenient, and easy to carry out. It is suitable for users to update the firmware in Windows environment. Please follow the tips below to finish image update

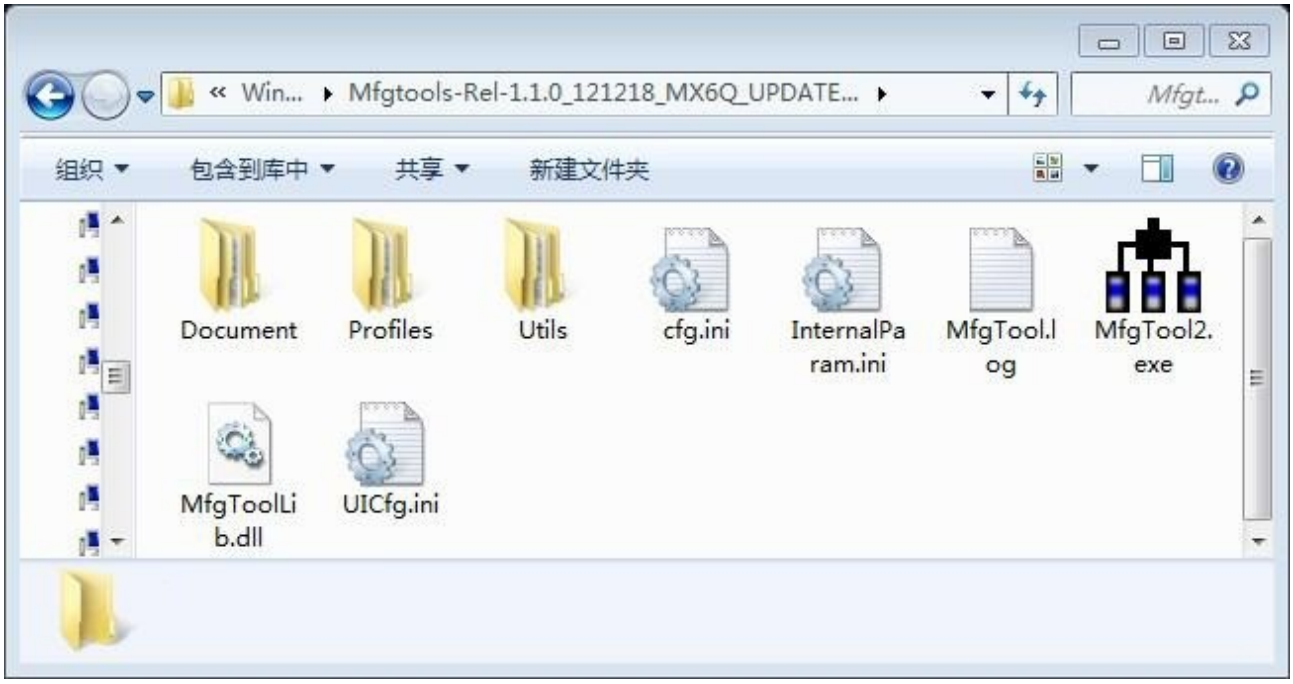
- **Tools and image**

Burning tool Mfgtool	Run in the Windows, and save it in “Software\Windows tools”
Virtual terminal software	Such as Windows's own super terminal
Kernel Image	uImage
u-boot Image	u-boot.bin
File system	rootfs.tgz
E9 mini PC	1pcs
USB OTG cable	1pcs
J5 jumper cape	1pcs
5V-2A DC Power Supply	1pcs

12.2 Windows Platform Burning

- **Burning Environment Setup**

Copy Mfgtools-Rel-1.1.0_121218_MX6Q_UPDATER.zip to your computer and extract it to Mfgtools-Rel-1.1.0_121218_MX6Q_UPDATER, then enter the directory as following shows:

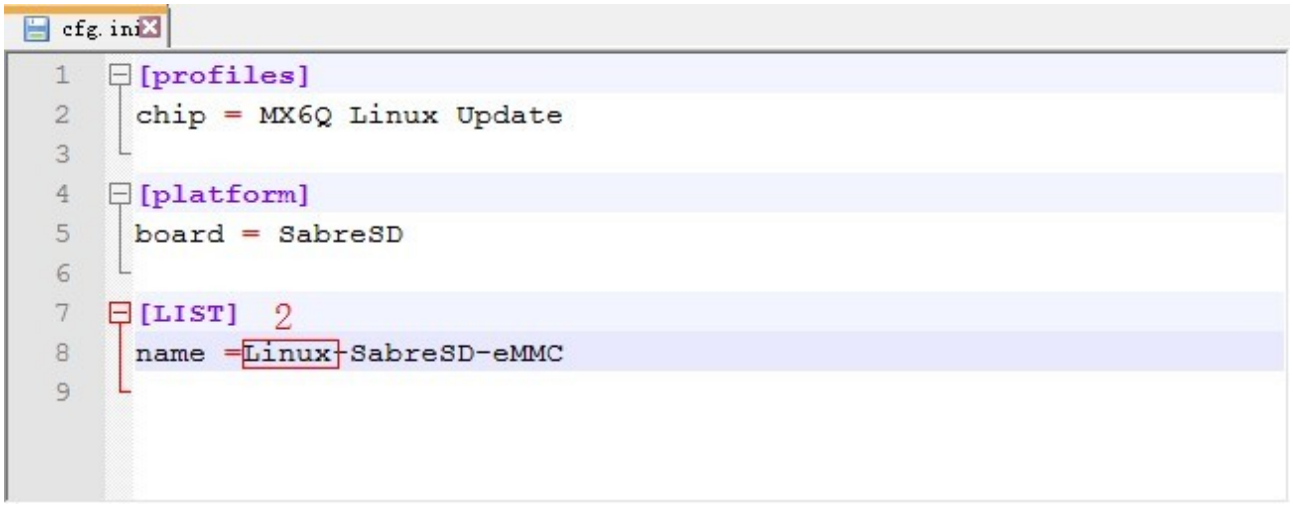


12.2.1 Burning Linux System

- **Modify the config file**

  1) Open the cfg file which is marked in red “1”;

2) Change the “2” marked in red box to “Linux” (The first letter "L" must be capitalized, as the system supports case-sensitive).

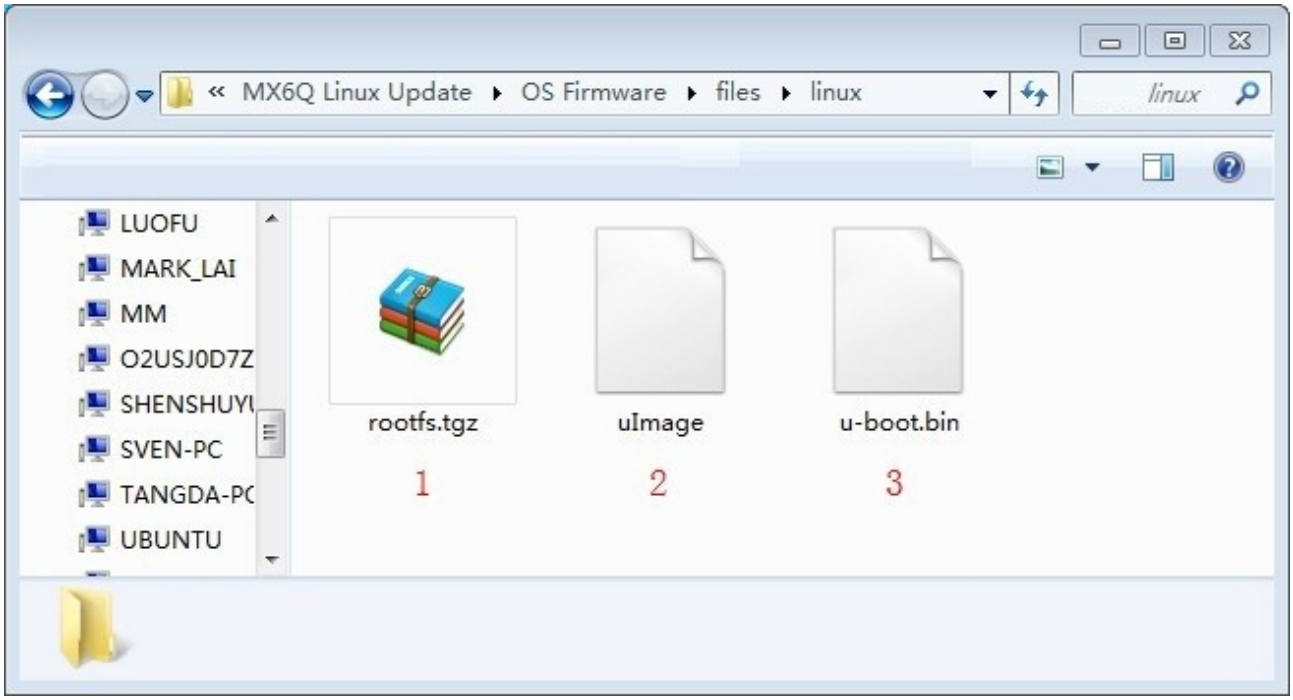


➢ **Copy the file**

Uboot image	Copy u-boot.bin to Mfgtools-Rel-1.1.0_121218_MX6Q_UPDATER\Profiles\MX6Q Linux Update\OS Firmware\files\linux (Or replace an existing file)
Kernel image	Copy uImage to Mfgtools-Rel-1.1.0_121218_MX6Q_UPDATER\ Profiles\MX6Q Linux Update\OS Firmware\files\linux (Or replace an existing file)
File System Image	Copy oneiric.tgz to Mfgtools-Rel-1.1.0_121218_MX6Q_UPDATER\Profiles\MX6Q Linux Update\OS Firmware\files\linux (Or replace an existing file)

Note: After copy the files; please change the "u-boot_E9_for_linux_v2.0.bin" to "u-boot.bin", delete "_E9_for_linux_v2.0", and so as the "uImage" and "oneiric.tgz".

After done, the file structure under the directory "Mfgtools-Rel-1.1.0_121218_MX6Q_UPDATER\Profiles\MX6Q Linux Update\OS Firmware\files\linux" is as showed below:



Check and confirm if all the above three files are copied successfully, all of them are indispensable.

12.2.2 Hardware Connection

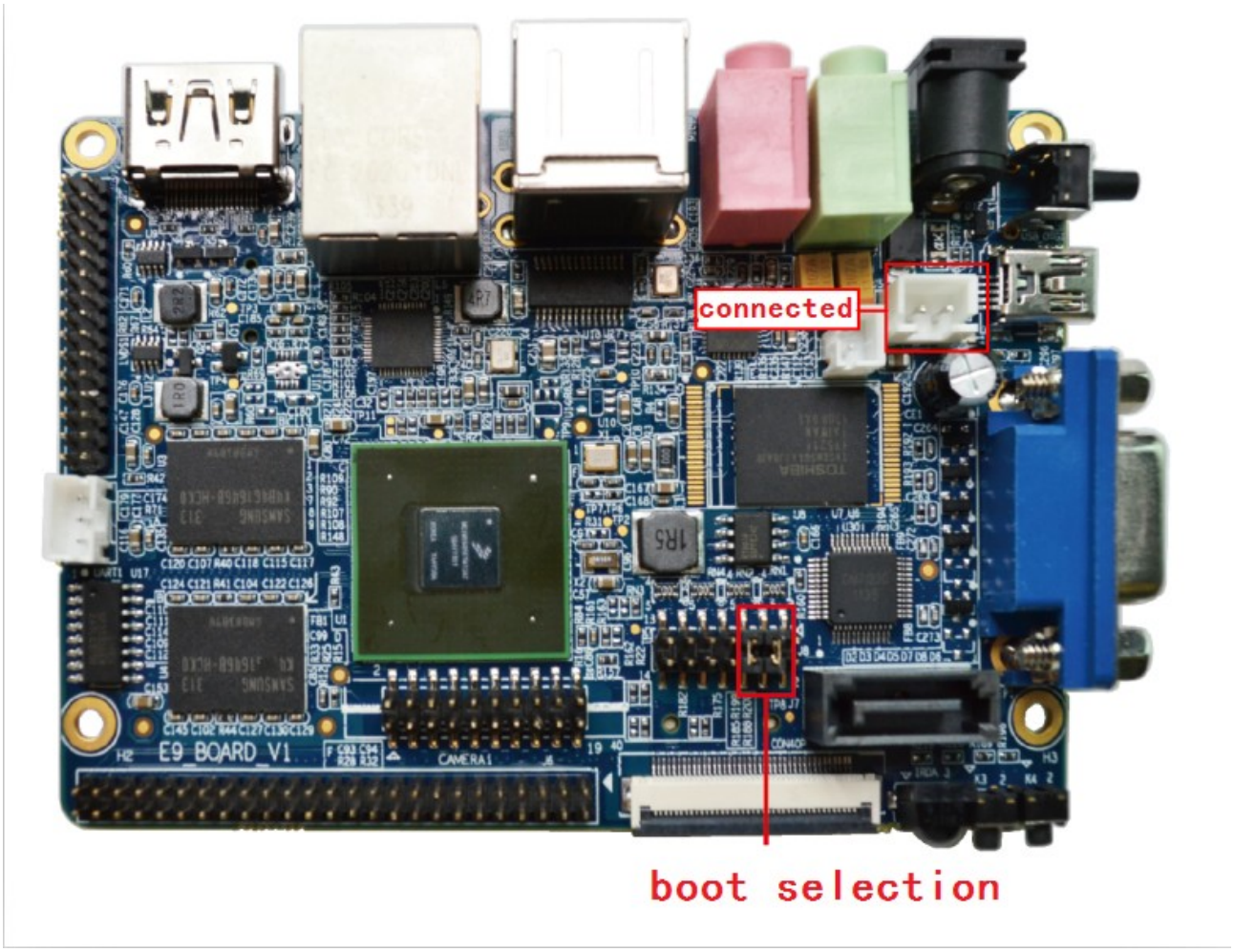
(1) Connect the PC and E9 via USB OTG cable.

(2) Connect the 5V DC power supply to E9 5V jack

(3) Set the USB download mode as the picture shows: short the pin2 and pin3 (from right to left) with jumper cap, and then download it.

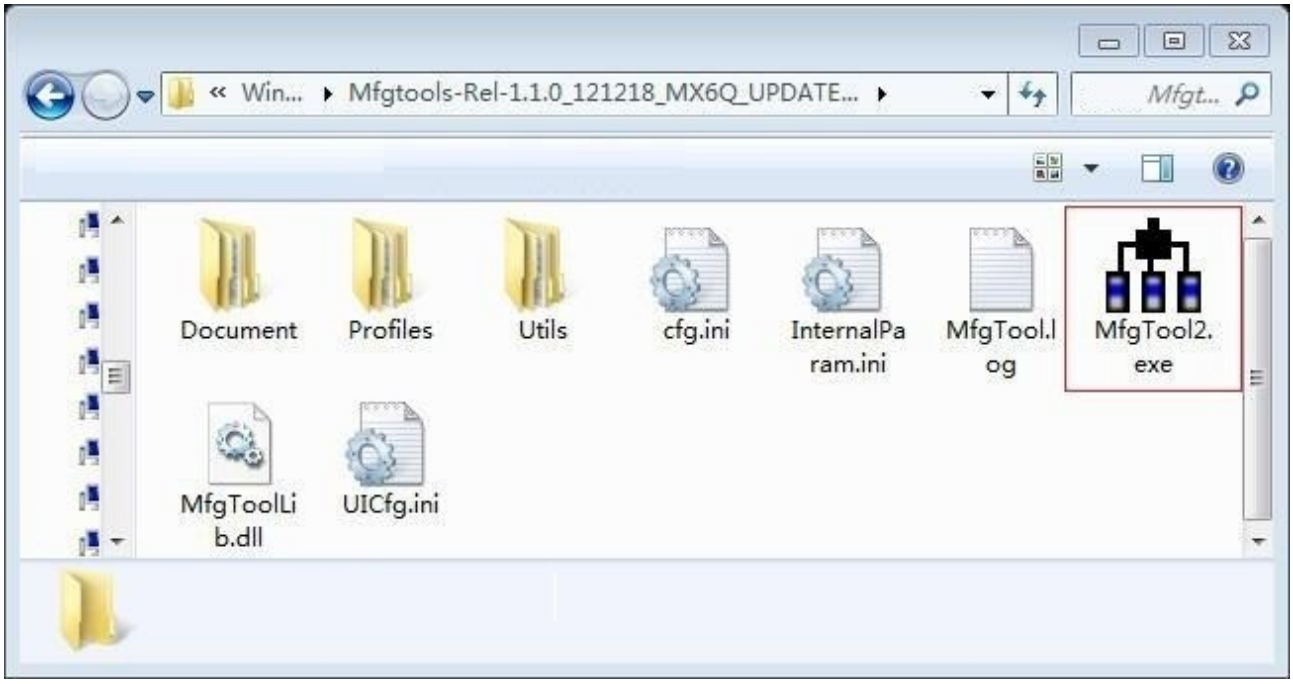
(4) Short / Connect the J5 spare external power supply interface (as marked below)

Note: J5 must be connected in the burning process.

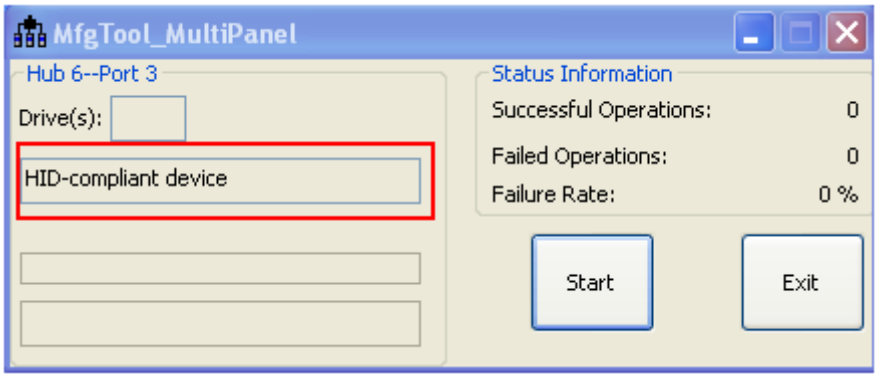


12.2.3 Begin to Burn

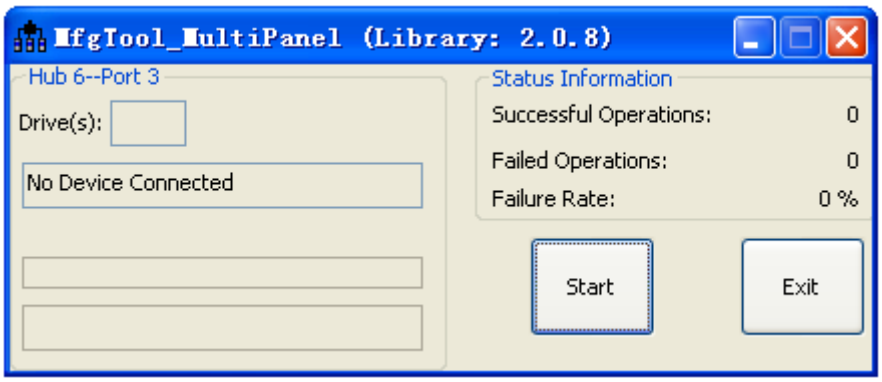
(1) Open the file “MfgTool2.exe” under the directory “Mfgtools-Rel-1.1.0_121218_MX6Q_UPDATER”.



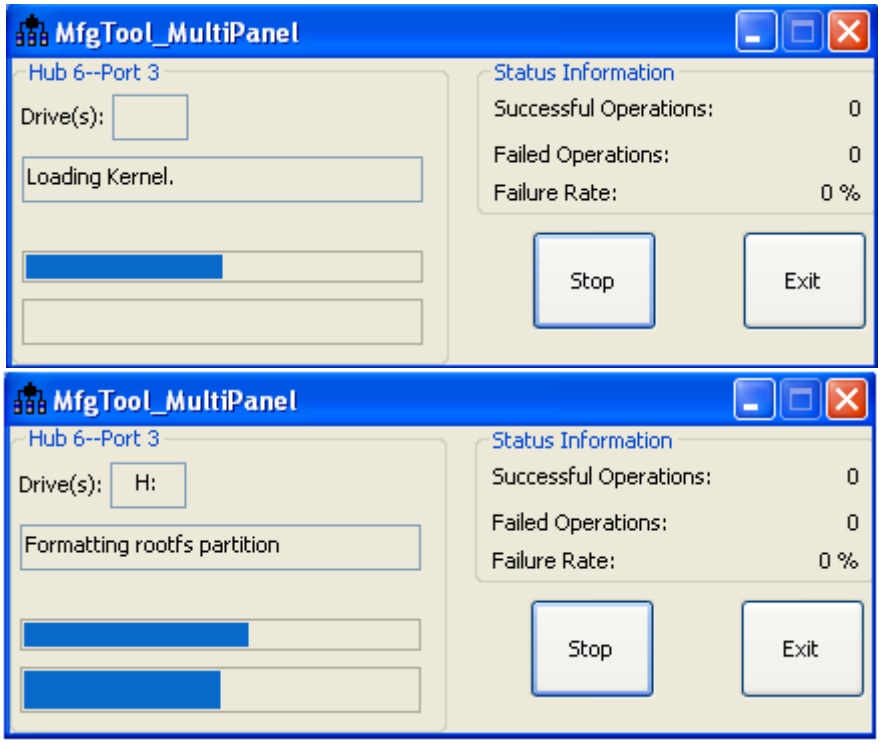
After "MfgTool2.exe" is opened, it can detect HID-compliant device as the following picture shows:



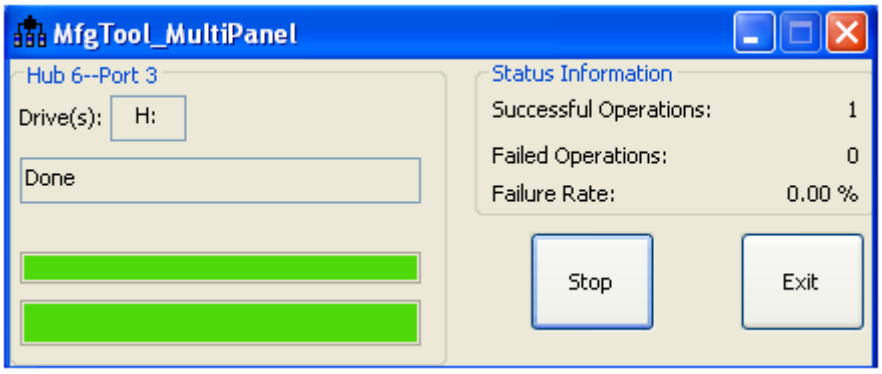
Note: If the USB device can not be detected, please recheck the OTG connection.



(2) Click the "start" button in the software MfgTool2 to begin burning (The burning process may take about 20~30 minutes due to different computer configuration, please be patient!)



(3) After done as the following picture shows, click the “stop” button, turn off the power supply and Unplug the USB OTG cable. Then follow the tips in Chapter 10.1.1, set it as emmc boot mode to run the system.



Note: After burning is done, users need to reset the default boot mode as “linux used emmc” in the uboot config, and then it can boot the system normally.

12.3 Linux Platform Burning

In the linux platform, it will use SD card burning automatically. Please prepare the following things before burning:

Burning tools	Software\Linux tools\downloader.tar.bz2
Kernel image	uImage
u-boot image	u-boot.bin
File System	rootfs.tgz
E9 mini PC	1pcs
TF card	1pcs

TF card reader	1pcs
5V-2A DC Power Supply	1pcs

12.3.1 Create SD Card

Extract the prepared related tools and images to any directory:

```
#tar  jxvf  downloader.tar.bz2  -C /
```

to the related directory:

```
#./create_sdcard.sh  –f  /dev/sdc
```

Note: Here the example based on the writer's computer, SD card disk is the /dev/sdc, and your own computer will based on the disk after the " #fdisk –l", just avoid burning the program to your computer hard drive, damage to your data, and cause needless loss.

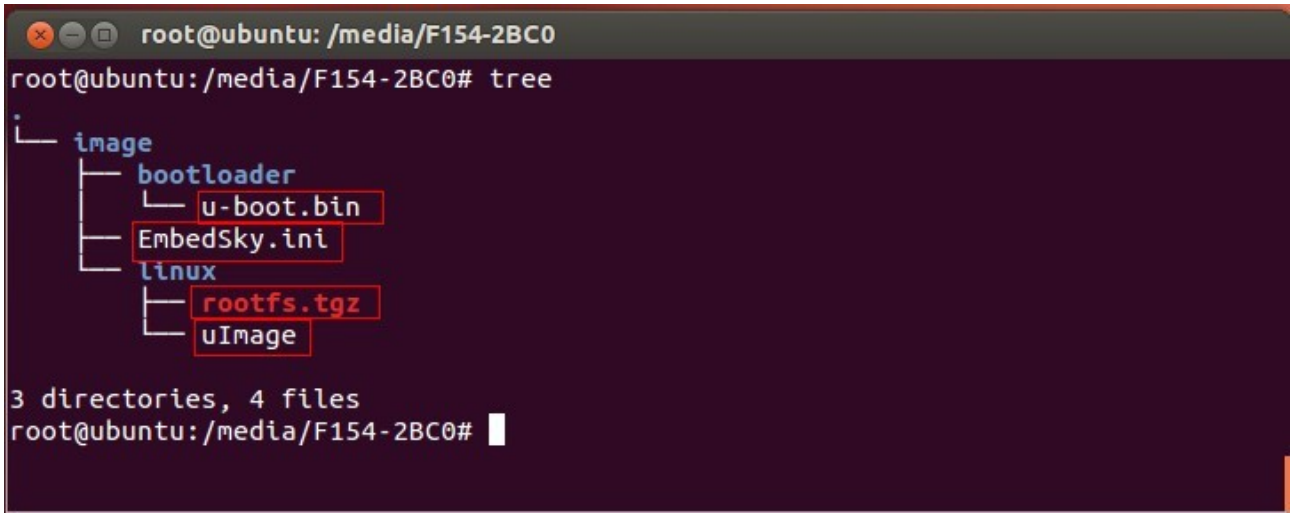
12.3.2 Ready to Burn Image

SD card burning automatically is judged from the EmbedSky.ini contents on the images directory in the SD card root directory to choose whether to burn and which images need to be burned. The file "EmbedSky.ini" must be put in the SD card images directory, if there is no such directory in the SD card, then please create a new directory like that.

12.3.3 Burning Linux Image

➢ **Copy image:**

Copy the linux image: u-boot_E9_for_linux_v2.0.bin, uImage_E9_for_linux_v2.0, rootfs_E9_for_linux_v2.0.tgz to the SD images\bootloader\ directory and images\linux directory separately, remove the "_E9_for_linux_v2.0" in the image name, then after that the burning file in the SD card should be like this:



➢ **Modify EmbedSky.ini file**

Note 1: The file format can't change optionally, and there is no space behind "=", and it cannot add or delete

a line either.

Note 2: The following "//XXX" is for comments only, and please don't add these comments when using the file, or else there will be mistakes.

Note 3: When users burning the ubuntu, it needs to pay attention to the comments, and modify it according to actual situation. There is no need to change others.

Note 4: For those images which don't need to update, it can be set as null, such as "Kernel ="

```
#### This is E9 SD config files ####
operating condition=install
OS Type=linux                    // operating system instructions
LED indicator=off
Beep=off
extern_indication=partition      // external indication, partition means it needs to partition, it is better to add this

#### Others ####
bootloader=images\bootloader\u-boot.bin        //u-boot image storage directory
LOGO=
#### Others end ####

#### Linux ####
kernel=images/linux/uImage             //linux uImage storage directory
rootfs=images/ linux/rootfs.tgz              //linux rootfs storage directory
#### Linux end ####

#### Windows CE ####
NK=images\wince\NK.bin
#### Windows CE end ####

#### Android ####
boot_img=images/android/boot.img
system_img=images/android/system.img
recovery_img=images/android/recovery.img
#### Android end ####

#### Ubuntu ####
kernel=images\
rootfs=images\
#### Ubuntu end ####

#### uCOS-II ####
uCOS-II Image=images\uCOS-II\TQ6410_UCOS-II.bin
#### uCOS-II end ####

#### User Application ####
Application Image=images\UserAPPS\TQ6410_Test.bin
```

```
#### User Application end #####
```

- **Burning System**

Put the TF card to the TF card slot, long press the power button to boot the system, then the system will complete burning automatically according to the EmbedSky.ini content. (The burning process may take about 20~30 minutes due to different computer configuration, please be patient!)

After done, the power light will be turned off automatically.

Note: After burning is done, users need to reset the default boot mode as “linux used emmc” in the uboot config, and then it can boot the system normally. For the setting instructions, please check Chapter 4.4

FAQ

How to power E9 on the conditions of without uboot?

The E9 uboot can supply power to E9, so on the conditions of without uboot, users can short the J5 connector, and can also keep pressing the power button.

What is the Ubuntu's username and password?

Username: linaro

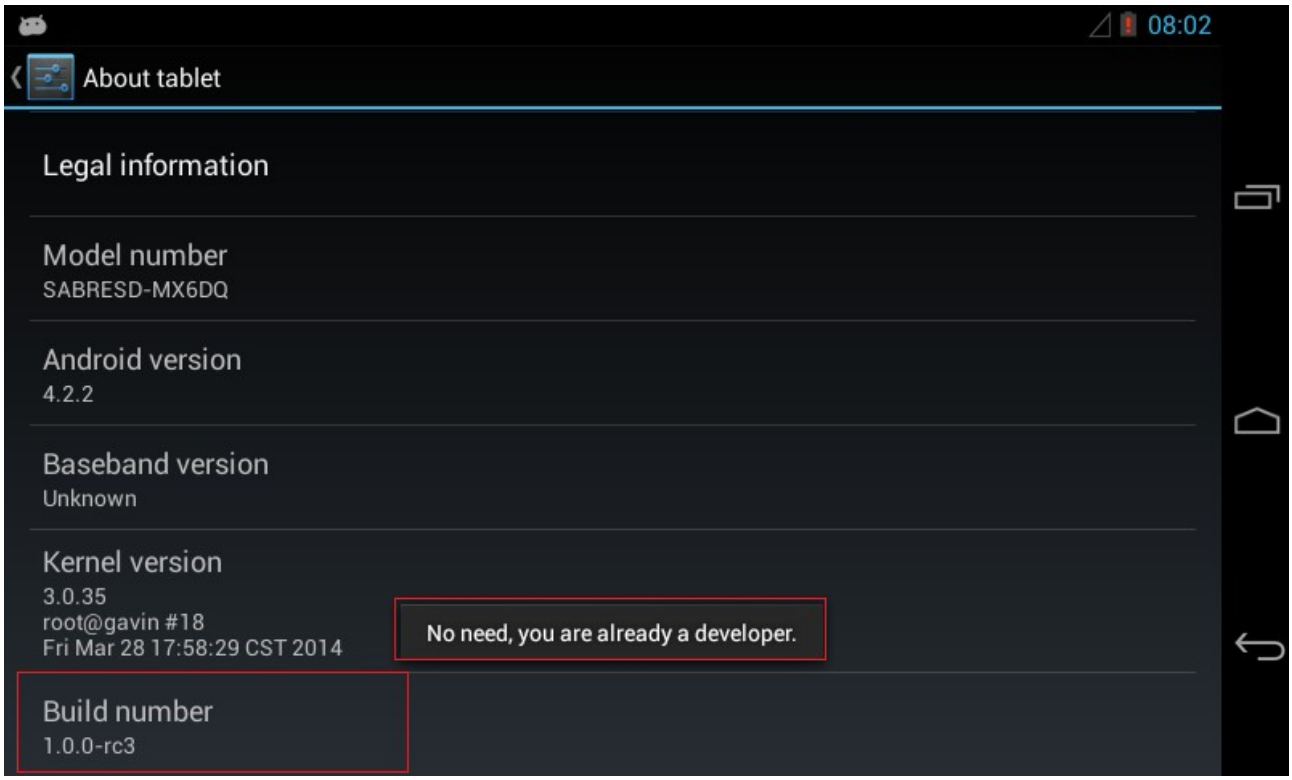
Password: linaro

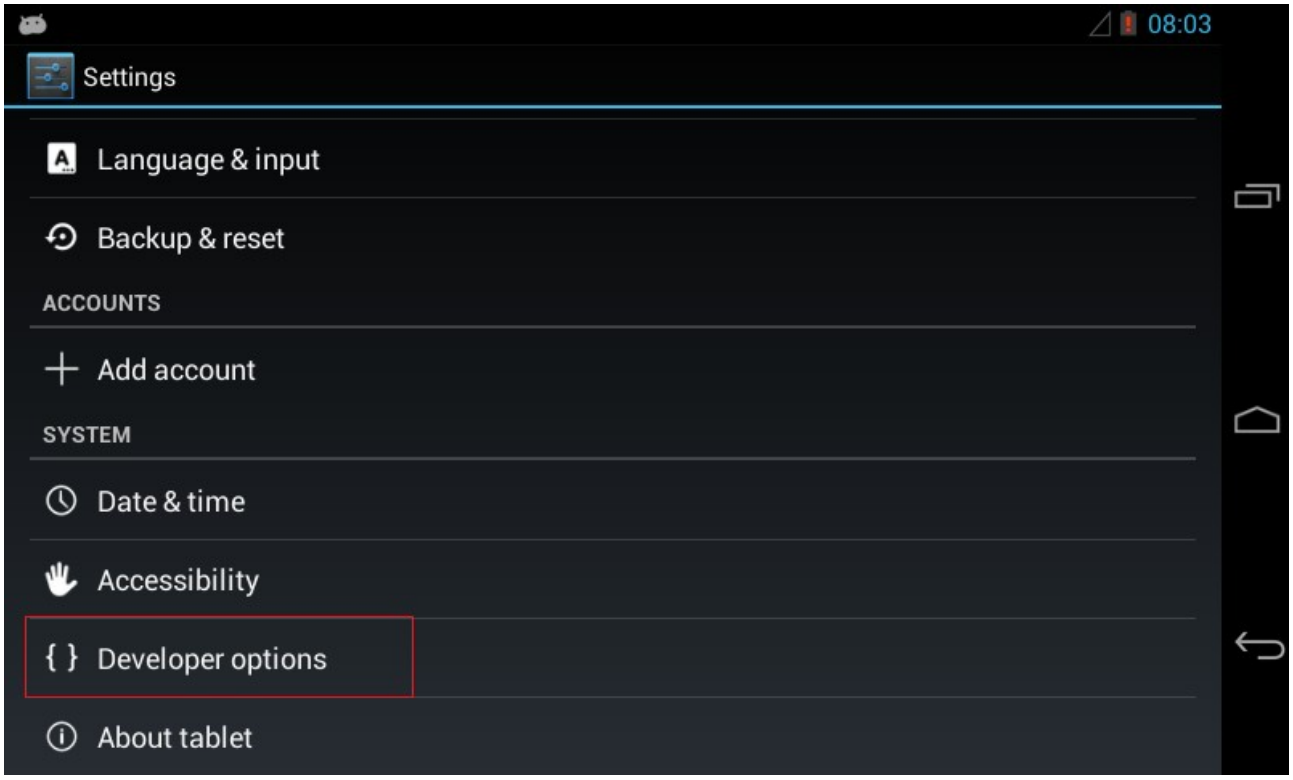
Can the Ubuntu username and password be modified? Or can create a new user?

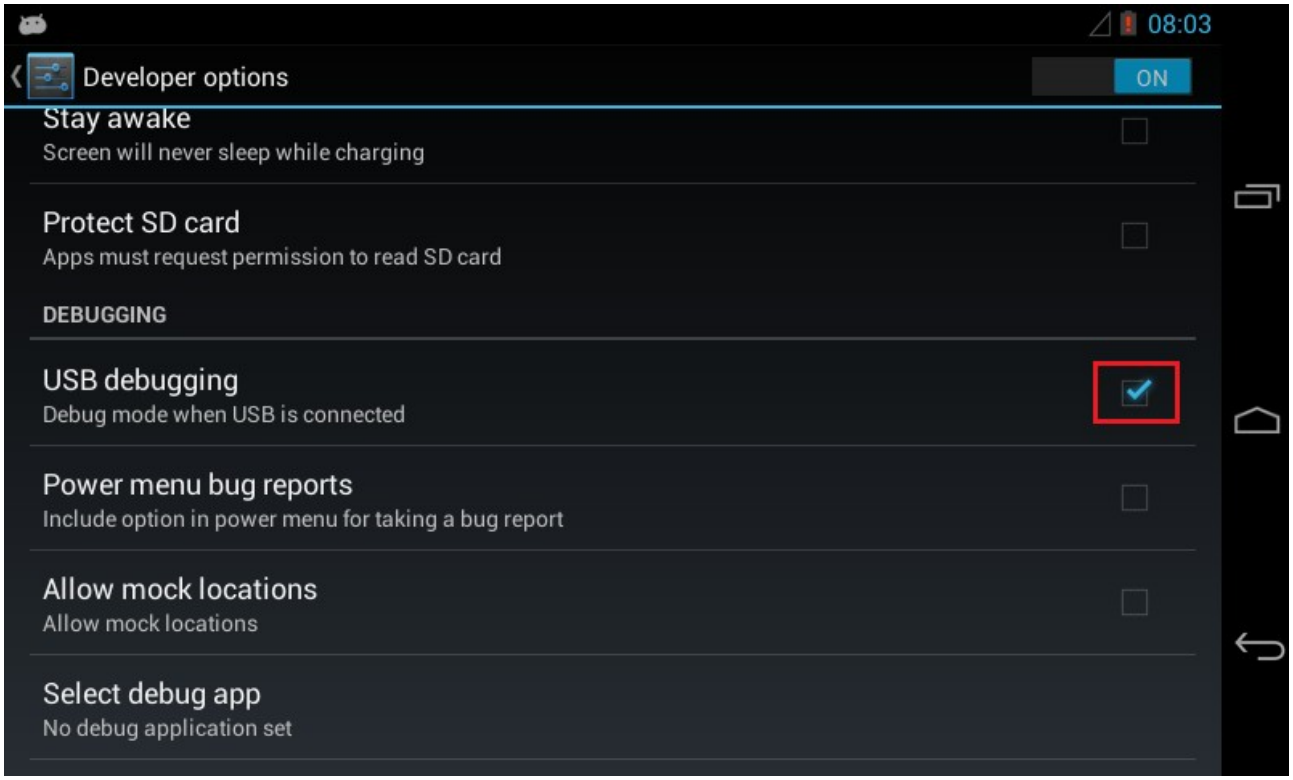
Currently we cannot modify the user name nor cannot create a new user.

How to open usb debugging function in Android?

Click on the red box repeatedly until you see the tips "No need, you are already a developer"







Ubuntu supports touch function or not?

Ubuntu system itself does not support touch function.

E9 sata hard disk uses?

It requires cable and power supply which need to prepared by users.