Ingenic®

Newton Android Development Guide

Date: Aug. 2014

北京君正集成电路股份有限公司
Ingenic Semiconductor Co., Ltd.

Ingenic®

Newton Android Development Guide

Release history

Date	Revision	Revision History
Aug 5, 2014	1.03	- Modified the binaries for flashing - Modified the offset address of flashing - Modified the UI of flashing tool - Modified the method to make a boot logo
Jun 12, 2014	1.02	- Modified commands to get Android source - Modified commands to compile u-boot
Apr. 30, 2014	1.01	- First released

Disclaimer

Ingenic Semiconductor Co., Ltd.

Ingenic Headquarters, East Bldg. 14, Courtyard #10,
Xibeiwang East Road, Haidian Dist., Beijing 100193, China
Tel: 86-10-56345000
Fax: 86-10-56345001
Http: //www.ingenic.cn

Content

1 Overview

This document provides information on how to set up the environment for Android developing for Newton platform. It also describes how to do the customization for your own product. This document is written for the system software engineers.

This document includes following information:

- How to setup environment for Android developing
- How to get the source code of Android
- How to compile
- How to setup the ADB
- How to do the customization

Before reading this document, you should:

- Familiar with the Ubuntu, Linux environment variables and shell commands
- Familiar with the Windows OS, and know how to install a device driver on it

2 Prepare the Environment

This section introduces how to setup the development environment for Newton platform.

2.1 Prepare Linux Development Host

The host PC requirements:

a. Hard disk capacity is not less than 128 GB.
b. RAM is not less than 4 GB.
c. Install the Ubuntu 12.04 64-bit system(It is required by compiling of Android).

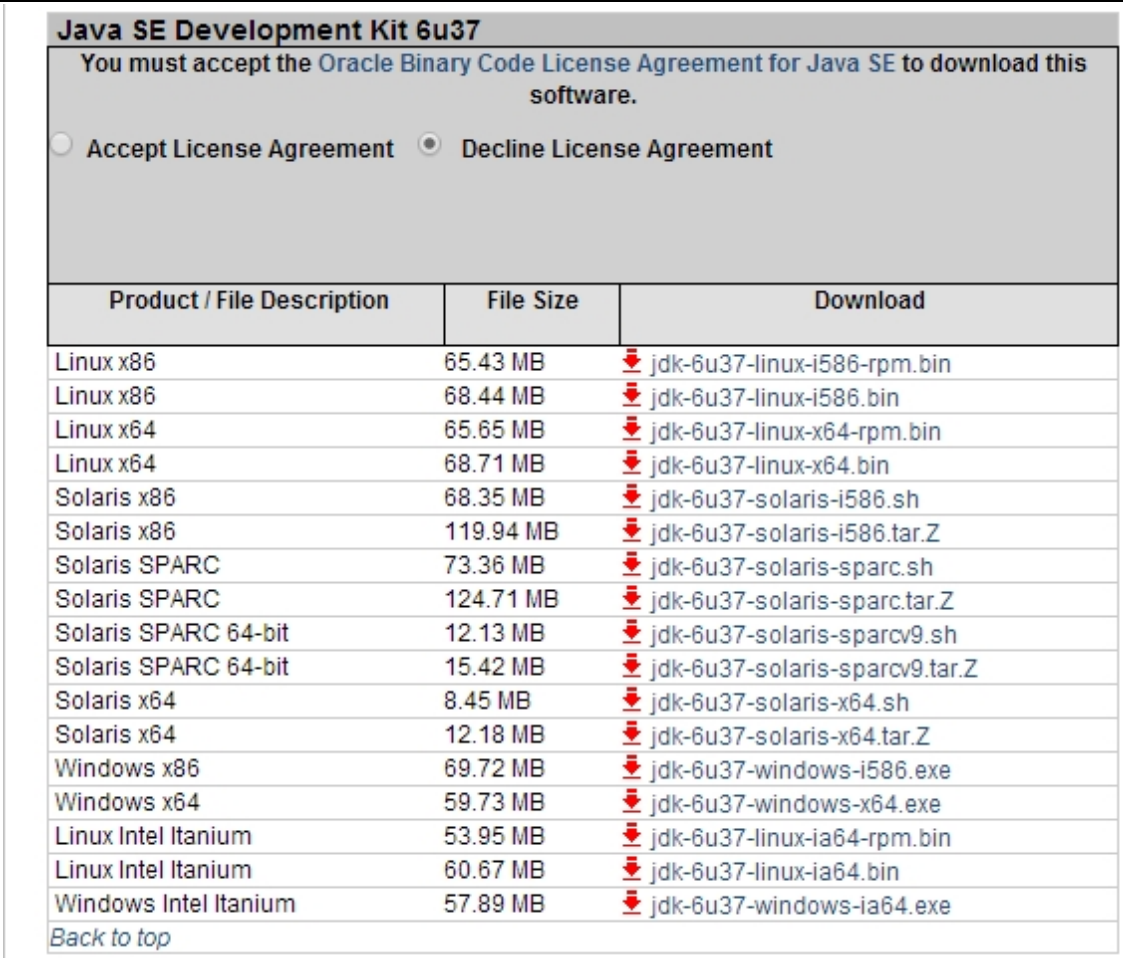
Steps to install:

a. Install 64-bit Ubuntu-12.04.
b. Install the following required packages:

   $ sudo apt-get install cpp-4.6 g++-4.6 gcc-4.6 gcc-4.6-multilib gcc g++ cpp gcc-multilib g++-4.6-multilib git-core git gnupg flex bison gperf build-essential zip curl libc6-dev libncurses5-dev:i386 x11proto-core-dev libx11-dev:i386 libreadline6-dev:i386 libgl1-mesa-glx:i386 libgl1-mesa-dev g++-multilib mingw32 tofrodos python-markdown libxml2-utils xsltproc zlib1g-dev:i386 ia32-libs gawk qt4-dev-tools libgl1-mesa-dri:i386 libglapi-mesa:i386 libncurses5-dev libqt3-mt-dev u-boot-tools

   The development environment of the Newton Android SDK is fully compatible with the Google Android. If you want to get more information about setting up the development environment, please visit http://source.android.com/source/initializing.html
c. Install JDK

   The recommended JDK version is jdk-6u37-linux-x64. It is available at http://www.oracle.com/technetwork/java/javase/downloads/java-archive-downloads-javase6-419409.html



Java SE Development Kit 6u37

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

Accept License Agreement Decline License Agreement

Product / File Description	File Size	Download
Linux x86	65.43 MB	jdk-6u37-linux-i586-rpm.bin
Linux x86	68.44 MB	jdk-6u37-linux-i586.bin
Linux x64	65.65 MB	jdk-6u37-linux-x64-rpm.bin
Linux x64	68.71 MB	jdk-6u37-linux-x64.bin
Solaris x86	68.35 MB	jdk-6u37-solaris-i586.sh
Solaris x86	119.94 MB	jdk-6u37-solaris-i586.tar.Z
Solaris SPARC	73.36 MB	jdk-6u37-solaris-sparc.sh
Solaris SPARC	124.71 MB	jdk-6u37-solaris-sparc.tar.Z
Solaris SPARC 64-bit	12.13 MB	jdk-6u37-solaris-sparcv9.sh
Solaris SPARC 64-bit	15.42 MB	jdk-6u37-solaris-sparcv9.tar.Z
Solaris x64	8.45 MB	jdk-6u37-solaris-x64.sh
Solaris x64	12.18 MB	jdk-6u37-solaris-x64.tar.Z
Windows x86	69.72 MB	jdk-6u37-windows-i586.exe
Windows x64	59.73 MB	jdk-6u37-windows-x64.exe
Linux Intel Itanium	53.95 MB	jdk-6u37-linux-ia64-rpm.bin
Linux Intel Itanium	60.67 MB	jdk-6u37-linux-ia64.bin
Windows Intel Itanium	57.89 MB	jdk-6u37-windows-ia64.exe

Back to top

Figure 2-1 Java JDK Download Web page

Select to download jdk-6u37-linux-x64.bin.

Steps to install the JDK:

i Copy jdk-6u37-linux-x64.bin to /usr/java/.

ii Add executing mode.

```
$ sudo chmod u+x /usr/java/jdk-6u37-linux-x64.bin
```

iii Install JDK by executing the following command:

```
$ sudo /usr/java/jdk-6u37-linux-x64.bin
```

iv When finishing the installation, you should setup the system environment variables for the JDK.Here provides two methods:

Method 1: Add following lines to /etc/environment

```
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/java/jdk1.6.0_37/bin"
JAVA_HOME=/usr/java/jdk1.6.0_37
JRE_HOME=/usr/java/jdk1.6.0_37/jre
CLASSPATH=$CLASSPATH:$JAVA_HOME/lib:$JRE_HOME/lib
```

Login again and the environment variables will take effect automatically.

Method 2: Add following lines to /etc/profile

```
export JAVA_HOME=/usr/java/jdk1.6.0_37
export JRE_HOME=/usr/java/jdk1.6.0_37/jre
```

export CLASSPATH=$CLASSPATH:$JAVA_HOME/lib:$JRE_HOME/lib

export PATH=$JAVA_HOME/bin:$JAVA_HOME/jre/bin:$PATH

Login again and the environment variables will take effect automatically.

d. Check the JDK tool

Type following command to check whether the JDK was install correctly.

```
root@Ubuntu-1204:~# java -version
java version "1.6.0_37"
Java(TM) SE Runtime Environment (build 1.6.0_37-b06)
Java HotSpot(TM) 64-Bit Server VM (build 20.12-b01, mixed mode)
root@Ubuntu-1204:~#
```

Figure 2-2 Java Version

The above message shows that the JDK was installed correctly.

2.2 Prepare Host for Flashing

The USB Burning Tool can only be run under Windows XP or 7. Refer to another document “How to burn Newton demo” for details.

3 Get Newton Android Source Code

3.1 Download repo

$ mkdir newton-android

$ cd newton-android

$ wget http://git.ingenic.cn:8082/bj/repo

$ chmod +x repo

3.2 Download the Android source

$ ./repo init -u http://git.ingenic.cn:8082/gerrit/AOSP/platform/manifest -b android-4.3-newton-4775

$ ./repo sync

3.3 Sync to the latest tag

$ ./repo forall -c "git reset --hard ingenic-android4.3.0-kernel3.0.8-newton-vx.x-yyyymmdd"

Method to find out the latest tag:

$ cd kernel

$ git tag | grep newton

4 Compiling Android

4.1 Android Source Tree

Android source tree will be looked like this:

```
root@Ubuntu-1204:~/newton-android# ls
abi         dalvik       docs        kernel           ndk         system
bionic      developers   external    libcore          packages    vendor
bootable    development  frameworks  libnativehelper  pdk
build       device       hardware    Makefile         prebuilts
root@Ubuntu-1204:~/newton-android#
```

Figure 4-1 Android Home

The source code of bootloader is located at bootable/bootloader/uboot/.

The source code kernel is located at kernel/.

The device configuration of Newton is located at device/ingenic/newton/.

4.2 Building the Whole Project

4.2.1 Build bootloader

```
$ source build/envsetup.sh
$ lunch full_newton-userdebug
$ cd bootable/bootloader/uboot/
$ make distclean
$ make newton_android_msc0
```

That will generate u-boot-with-spl-mbr-gpt.bin.

4.2.2 Build kernel

```
$ cd ../../../kernel
$ make newton_android_msc_defconfig
$ make zImage
$ cp arch/mips/boot/compressed/zImage ../device/ingenic/newton/kernel
```

4.2.3 Build Android

```
$ cd ..
$ make
```

That will generate boot.img, system.img and recovery.img under out/target/product/newton/.

4.3 Build boot.img alone

Execute following commands under the Android home directory:

```
$ source build/envsetup.sh
$ lunch full_newton-userdebug
$ cd kernel/
$ make newton_android_msc_defconfig
$ make zImage
$ cp arch/mips/boot/compressed/zImage ../device/ingenic/newton/kernel
$ cd ..
$ make bootimage
```

That will generate boot.img under out/target/product/newton/.

4.4 Build system.img alone

Execute following commands under the Android home directory:

```
$ source build/envsetup.sh
$ lunch full_newton-userdebug
$ make systemimage
```

That will generate system.img under out/target/product/newton/.

4.5 Build Android modules

```
$ source build/envsetup.sh
$ lunch full_newton-userdebug
```

Change to the directory of the compiled module, for an example:

```
$ cd packages/apps/Calendar
$ mm
```

4.6 Burning the Android Images

Following binaries should be flashed into Newton. Please visit website of Ingenic to get the latest Flashing tool.

- mbr-xboot-gpt.bin
- boot.img
- system.img
- recovery.img

The partition table of the Newton is:

Newton Partition table

Board	Hardware	File	Offset(B)	Option	Configuration
Newton	Storage: EMMC 4G, 512B/Sector Mem: Mobile DDR	u-boot-with-spl-mbr-gpt.bin	0	MMC0	Newton_mmc_lpddr.cfg
		boot.img	0x300000	MMC0	
		recovery.img	0xb00000	MMC0	
		system.img	0x3800000	MMC0	

To configure this parameters, click the “Configure” button:

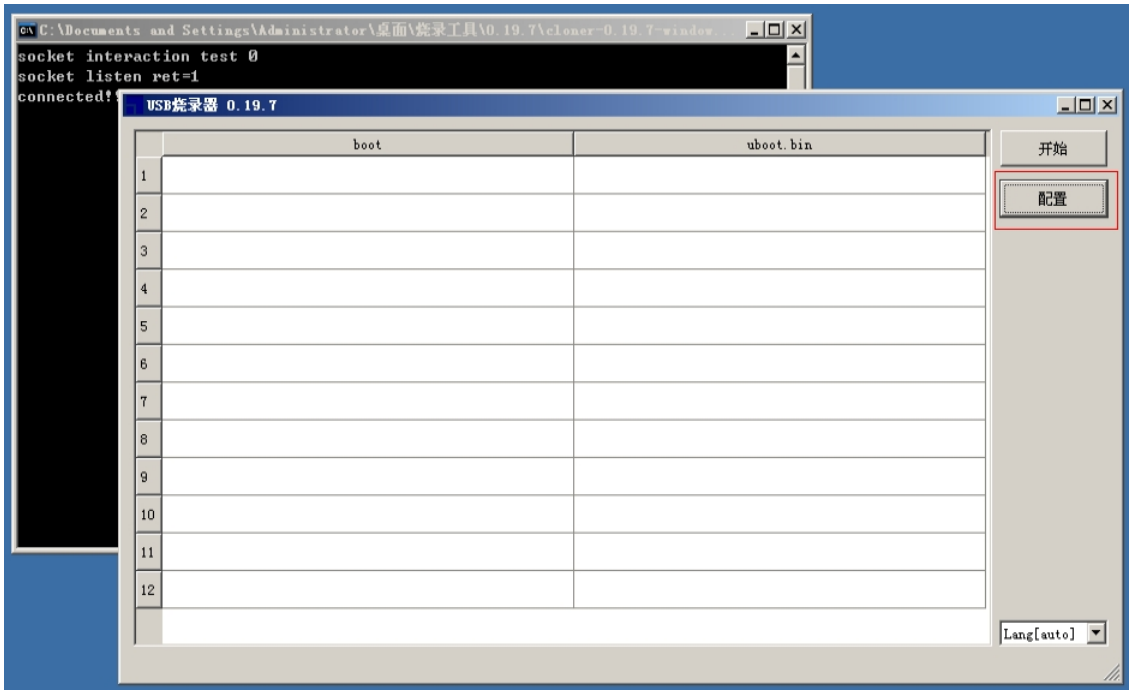


Figure 4-2 Configure Button

Make sure the configure file is newton_mmc_lpddr.cfg.

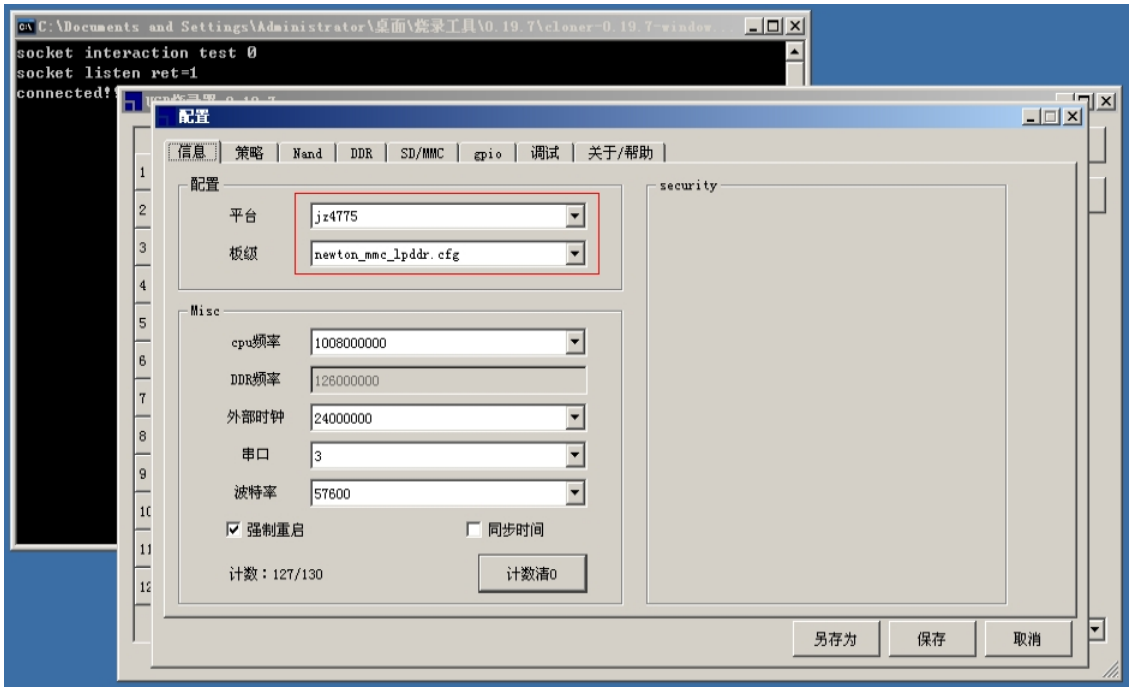


Figure 4-3 Configure File

Check the files being flashed and their offset. You can select/unselect certain files according to your situation.

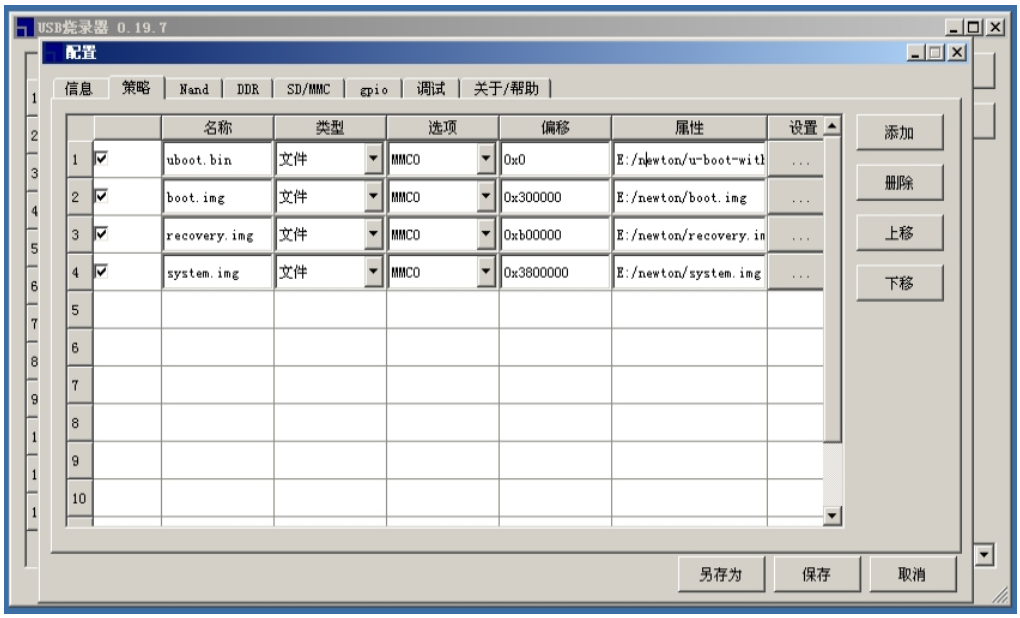


Figure 4-4 Select All Files

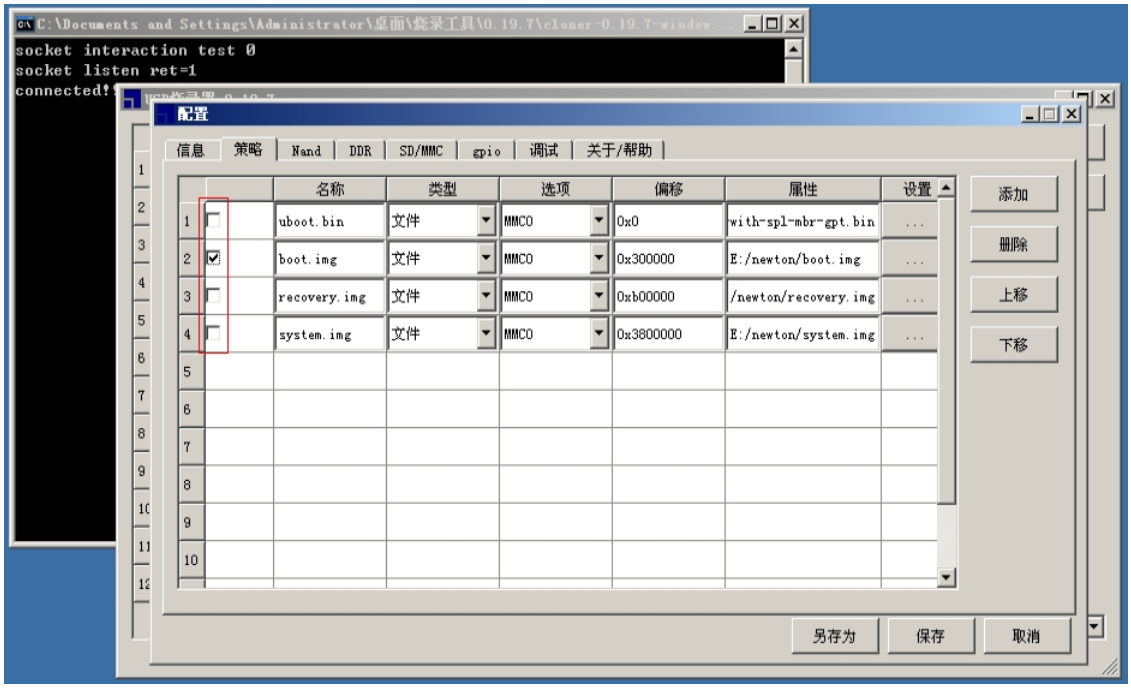


Figure 4-5 Only Select boot.img

'Force erase ' and 'Erase all' should be checked if you want to update all images in Newton.

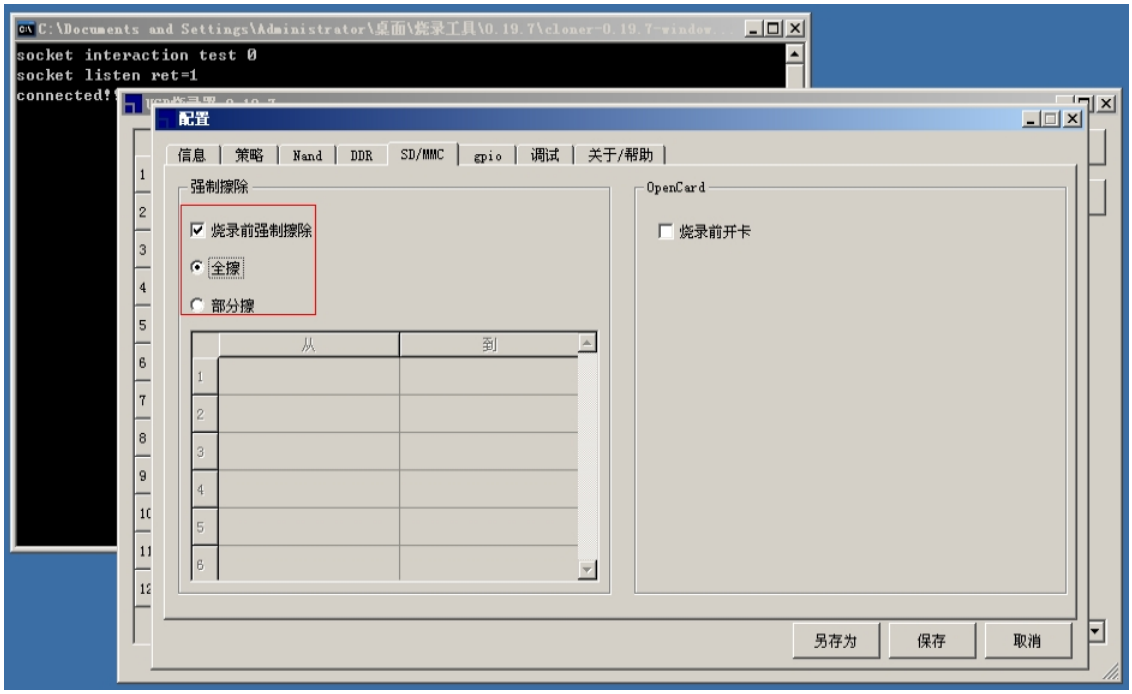


Figure 4-6 Erase All

If not all images need to be updated, you should not check ‘Force erase’ and ‘Erase all’.

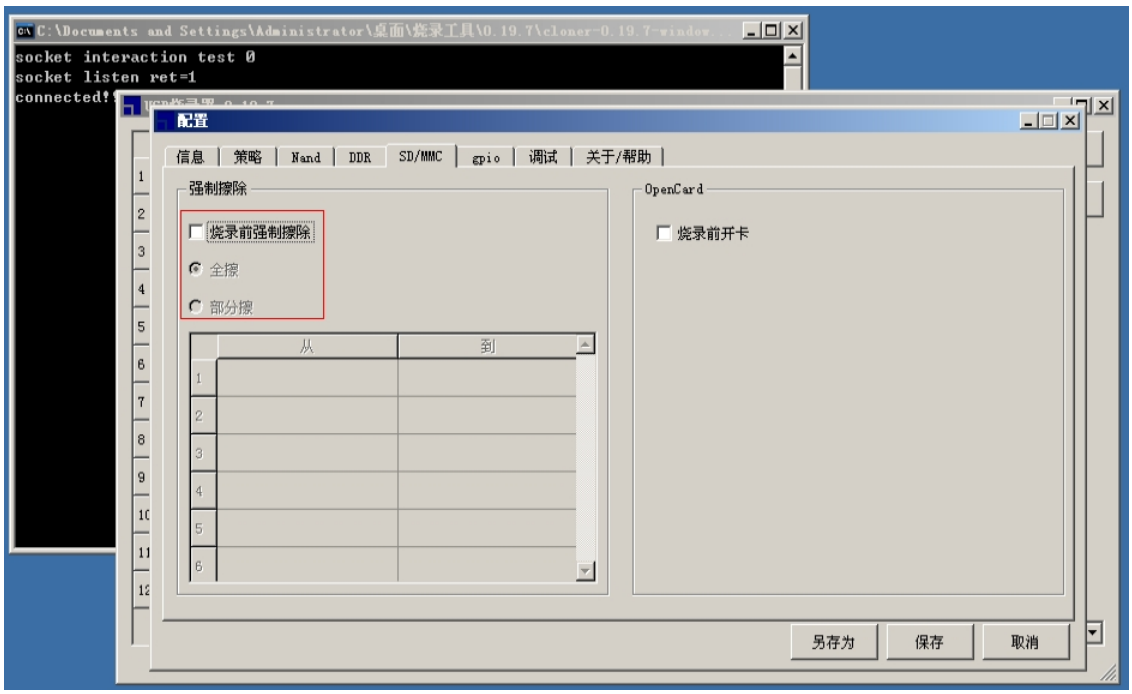


Figure 4-7 Do not Erase All

5 Install ADB Tool

5.1 Install ADB Tool on Linux Host

5.1.1 Add Configure File

Create a new file named /etc/udev/rules.d/51-android.rules, add the following line:

```
SUBSYSTEM=="usb|usb_device", ATTR{idVendor}=="18d1", MODE="0666", GROUP="plugdev"
```

5.1.2 Download the Android SDK

You can find it at: http://developer.android.com/sdk/index.html

ADT Bundle

Platform	Package	Size	MD5 Checksum
Windows 32-bit	adt-bundle-windows-x86-20140321.zip	535085536 bytes	b61495a6bf591cc374c31bce4fc46ec0
Windows 64-bit	adt-bundle-windows-x86_64-20140321.zip	535287324 bytes	a6f4699bbdc5a29b371ed60610535651
Mac OS X 64-bit	adt-bundle-mac-x86_64-20140321.zip	501955296 bytes	4a08649cea9b098cdf7349f452294014
Linux 32-bit	adt-bundle-linux-x86-20140321.zip	527971926 bytes	943ae4d28fe7c79108c8bf2aafd5e6d2
Linux 64-bit	adt-bundle-linux-x86_64-20140321.zip	528187678 bytes	f2a2153b5c7dbaeb86b550bf4f770c36

Figure 5-1 List of Android Linux SDK

Choose the certain version of SDK. For example, choose 'adt-bundle-linux-x86_64-20140321.zip' for 64-bit version of Ubuntu. Then:

a. Extract the zip file to the <ANDROID_SDK_INSTALL_PATH> directory.
b. Modify ~/.bash_profile, add:
   export PATH=${PATH}:<ANDROID_SDK_INSTALL_PATH>/platform-tools
c. Run "source ~/.bash_profile", or re login.

5.2 Install ADB Tool on Windows Host

If you are debugging on Windows, you need to install ADB driver.

5.2.1 Download SDK

You can find it at: http://developer.android.com/sdk/index.html

ADT Bundle

Platform	Package	Size	MD5 Checksum
Windows 32-bit	adt-bundle-windows-x86-20140321.zip	535085536 bytes	b61495a6bf591cc374c31bce4fc46ec0
Windows 64-bit	adt-bundle-windows-x86_64-20140321.zip	535287324 bytes	a6f4699bbdc5a29b371ed60610535651
Mac OS X 64-bit	adt-bundle-mac-x86_64-20140321.zip	501955296 bytes	4a08649cea9b098cdf7349f452294014
Linux 32-bit	adt-bundle-linux-x86-20140321.zip	527971926 bytes	943ae4d28fe7c79108c8bf2aafd5e6d2
Linux 64-bit	adt-bundle-linux-x86_64-20140321.zip	528187678 bytes	f2a2153b5c7dbaeb86b550bf4f770c36

Figure 5-2 List of Android Windows SDK

Choose the certain version of SDK. For example, choose 'adt-bundle-windows-x86-20140321.zip' for 32-bit version of Windows. Then:

a. Right-click on 'My Computer' and choose 'Properties'.
b. Click 'Advanced system settings', open 'Environment Variables'.
c. Find and edit 'Path' in 'System variables'.
d. Add the path of adb at the tail.

If the former step was correct, you will get the version of ADB by running 'adb version' in cmd line.

```
C:\Users\Administrator>adb version
Android Debug Bridge version 1.0.31
```

Figure 5-3 ADB version

5.2.2 Install ADB driver

a. Download Google USB driver

http://developer.android.com/sdk/win-usb.html

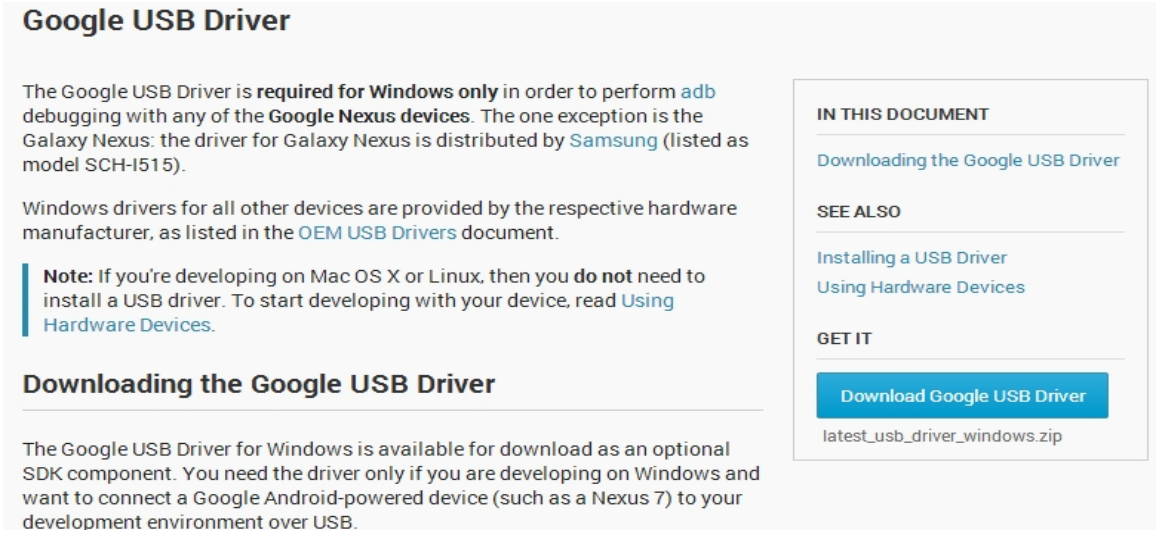


Figure 5-4 Google USB driver

b. Install the driver

Please refer the detail at Android website

http://developer.android.com/tools/extras/oem-usb.html#InstallingDriver

6 Customization

6.1 Customize Bootloader Logo

6.1.1 Prepare an original picture of logo

Recommended to use a picture which has the same resolution or smaller than LCD screen and rename it to logo.jpg.

For example, if the screen resolution of 288x192, please create a 288x192 jpg picture and saved as logo.jpg.

If have problem while displaying this picture, please change the width and height into even numbers.

6.1.2 Copy logo.jpg to certain directory

Copy logo.jpg into bootable/bootloader/uboot/tools/logos/

6.1.3 Compile u-boot

a. Make some modification

```
$ cd bootable/bootloder/uboot
$ vim tools/ingenic-tools/logo.mk
```

Change

```
BOOT_LOGO_JPG ?= $(TOPDIR)/tools/logos/ingenic.jpg
```

into

```
BOOT_LOGO_JPG ?= $(TOPDIR)/tools/logos/logo.jpg
```

b. Compile u-boot

```
$ make clean
$ make newton_android_msc0
```

6.1.4 Flash the New u-boot-with-spl-mbr-gpt.bin into Newton

Boot logo will be changed.

6.2 Customize Boot Animation

6.2.1 Prepare Pictures for Bootanimation

Prepare the pictures of each frame for bootanimation according the screen resolution. Name these pictures according to the playback order, such as all_0001.jpg, all_0002.jpg... and so on.

6.2.2 Prepare the directory for Compression

```
$ mkdir bootanimation
$ cd bootanimation
$ mkdir part1 part2
$ cp ../*.jpg part1
$ cp ../*.jpe part2
```

```
$ touch desc.txt
```

part1 is the part only played once, part2 is the part played until booting up complete.

Edit desc.txt as below, save and quit.

```
288 192 30
p 1 0 part1
p 0 0 part2
```

The first line is the resolution(width, height) and f/s.

The second line: ‘p’ is a signal, ‘1’ means looping only once, ‘0’ means interval is 0, ‘part1’ is the name of directory.

The third line: ‘p’ is a signal, ‘0’ means looping endlessly, ‘0’ means interval is 0, ‘part2’ is the name of directory.

Make sure there no other files except of the above files above. Directory tree:

```
├──── desc.txt
├──── part1
│     ├──── all_0001.jpg
│     ├──── all_0002.jpg
│     └──── all_0003.jpg
└──── part2
      ├──── all_0001.jpg
      ├──── all_0002.jpg
      └──── all_0003.jpg
```

6.2.3 Compress the Directory

```
$ zip -0 bootanimation.zip part1/*jpg part2/*jpg desc.txt
```

6.2.4 Replace boodanimation.zip

If bootanimation.zip is ready, copy it into /system/media/bootanimation.zip or /data/local/bootanimation.zip.