

Ingenic[®]
JZ4780 Android 4.1 Development
Guide

Date: Sep. 2013



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

Ingenic®

JZ4780 Android 4.1 Development Guide

Copyright © Ingenic Semiconductor Co. Ltd 2005 - 2013. All rights reserved.

Release history

Date	Revision	Change
Apr. 2013	1.01	First release
Sep.2013	1.02	Update git download site and corporation address

Disclaimer

This documentation is provided for use with Ingenic products. No license to Ingenic property rights is granted. Ingenic assumes no liability, provides no warranty either expressed or implied relating to the usage, or intellectual property right infringement except as provided for by Ingenic Terms and Conditions of Sale.

Ingenic products are not designed for and should not be used in any medical or life sustaining or supporting equipment.

All information in this document should be treated as preliminary. Ingenic may make changes to this document without notice. Anyone relying on this documentation should contact Ingenic for the current documentation and errata.

Ingenic Semiconductor Co., Ltd.

**Ingenic Headquarters, Zhongguancun Software Park
Dongbeiwang West Road, Haidian District, Beijing, China, 100193**

Tel: 86-10-56345000

Fax: 86-10-56345001

Http: //www.ingenic.cn

Content

1	Overview	1
2	Building Development Environment.....	3
2.1	Installing required packages	3
2.2	Installing the JDK	3
2.3	Extend info	3
2.4	Get Android source of Ingenic	3
3	Compiling and Prepare Burn file	5
3.1	Source code	5
3.2	Compiling kernel and system file	5
3.3	Compiling bootloader	5
3.4	Prepare Burn Files	6
3.4.1.1	Prepare the binary files for burning:	6
4	Flashing with FastBoot	7
4.1	Enter Fastboot Mode	7
4.2	Erase and Flash Images	7
4.3	Update Fastboot	7
5	Burn with SD Card boot	9
5.1	Prepare before operation	9
5.2	Make the Firmware for SD Burn	9
5.3	9
5.4	Make the Burning Tool on Sdcard.....	9
5.4.1	Prepare a 2G SD Card	9
5.4.2	Format SD Card	9
5.4.3	Modify Configuration.txt.....	9
5.5	4 Open the SD CARD burning tool	9
5.6	5 Make the MMC/SD Partition is the same with your SD card on your Windows XP.	10
5.7	6 Start Burning	11
5.8	7 Burning succeed	14
5.9	8 Take Card Reader	14
5.10	9 Quick format your SD Card.....	14
5.11	10 Configure your SD Card.....	14
6	Board configuration	15
6.1	Board configuration	15
6.2	Kernel Board configuration	15
6.3	Product configuration in system.....	16

1 Overview

This paper describes how to build Android4.1 system development environment for development board based on Ingenic JZ4780 (32-bit RISC) processor.

The document focuses on the following aspects:

- 1) Prepare basic environment based on a 64-bit PC,
- 2) Building the compiler environment;
- 3) Download the source code;
- 4) How to compile;
- 5) The configuration for product;

2 Building Development Environment

Before building the environment, a PC with 64-bit LinuxOS is required. 64-bit ubuntu 12.04 is recommended.

2.1 Installing required packages

```
user@ubuntu:sudo apt-get install cpp-4.6 g++-4.6 gcc-4.6 gcc-4.6-multilib gcc g++ cppgcc-multilib
g++-4.6-multilib git-core 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 la32-libs
```

2.2 Installing the JDK

The Sun JDK is no longer in Ubuntu's main package repository while openjdk is preloaded. Please uninstall openjdk completely first, and then install the latest JDK package corresponding to 64-bit machine from java.sun.com, jdk-6u37-linux-x64 is recommended;

In order to download it, you need to add the appropriate repository and indicate to the system which JDK should be used.

2.3 Extend info

The development environment for JZ4780 is compatible with official android. To get more detailed compilation environment to establish methods, please refer to the android official website: <http://source.android.com/source/initializing.html>

2.4 Get Android source of Ingenic

Ingenic can provide integrated source code about Android. The following will show you how to use repo to get Android source of Ingenic.

1. Download repo script

```
$ mkdir android-ingenic
$ cd android-ingenic
$ wget http://git.ingenic.cn:8082/bj/repo
$ chmod +x repo
```

2. Download Android source

```
$ ./repo init -u http://git.ingenic.cn/android/platform/manifest -b dev-ing-jb-local-4780 -m
outside.xml
$ ./repo sync
$ ./repo forall -c "git reset --hard ingenic-COMMONV1.0-20130403"
```

Note: "ingenic-COMMONV1.0-20130403" is a release TAG of Android source. Please pay attention to the latest update info in Ingenic official website <http://www.ingenic.cn>.

3 Compiling and Prepare Burn file

3.1 Source code

The AndroidJellyBeanBase on JZ4780 include the following the directories:

```
user@ubuntu:~/android-41$ ls
```

abi	development	hardware	ndk	system
bionic	device	kernel	packages	update
bootable	docs	libcore	pdk	vendor
build	external	libnativehelper	prebuilt	
cts	frameworks	Makefile	prebuilts	
dalvik	gdk	nda	sdk	

- 1) Bootloader:bootable/bootloader/xboot
- 2) AndroidLinux kernel: kernel
- 3) Others are Android Source

3.2 Compilingkernel and system file

Change dir to root directory of Android, run command:

For JZ4780 GRUSdevboard:

```
user@ubuntu:~/android-41$ ./build/scripts/mbuild_nandallimggrusgrus_nand
```

For JZ4780 Warrior PAD:

```
user@ubuntu:~/android-41$ ./build/scripts/mbuild_nandallimg warrior  
warrior_release_nand
```

The whole compiling may need several hours.

When compiling is finished, please get the binary files in out/target/product/grus or out/target/product/warrior.

system.img.bin (For USBBurn)

system.img (For SDBurn)

boot.img

3.3 Compilingbootloader

Change directory bootable/bootloader/xboot/ and run command

For GRUS devboard:

```
user@ubuntu:~/android-41$ make clean; make grus_nand_config; make
```

For Warrior PAD:

```
user@ubuntu:~/android-41$ make clean; make warrior_nand_config; make
```

The whole compiling may need several minutes.

When compile is finish, get the binary file in current directory:

x-boot-nand.bin

Now we can burn the x-boot-nand.bin,system.img.bin or System.img or system.imgand

boot.img into development board.

3.4 Prepare Burn Files

3.4.1.1 Prepare the binary files for burning:

- x-boot-nand.bin
 - boot.img
 - system.img (or system.img.bin)
- system.img.bin: burn with USBBurnTool.
system.img: burn with SDBurnTool.

Please reference the USBBurnTool/SDBurnTool User Manual or Quick Start Manual of the board.

4 Flashing with FastBoot

4.1 Enter Fastboot Mode

Press 'esc' and do a reset;

NOTICE: Keeping 'esc' pressed until seeing the fastboot logo

(A green Android robot and a few words).

4.2 Erase and Flash Images

Once the device is in fastboot mode, use **super user account** run

```
$ cp out/host/linux-x86/bin/fastboot out/target/product/warrior/  
$ cd out/target/product/warrior/  
$ ./fastboot erase userdata  
$ ./fastboot erase cache  
$ ./fastboot erase boot  
$ ./fastboot erase system  
$ ./fastboot flash boot boot.img  
$ ./fastboot flash:raw system system.img.bin  
$ ./fastboot reboot
```

4.3 Update Fastboot

If you met some problem with fastboot, you can try updating x-boot.

5 Burn with SD Card boot

5.1 Prepare before operation

i) An SD card for contain the Firm

5.2 Make the Firmware for SD Burn

5.3

5.4 Make the Burning Tool on Sdcard

5.4.1 Prepare a 2G SD Card.

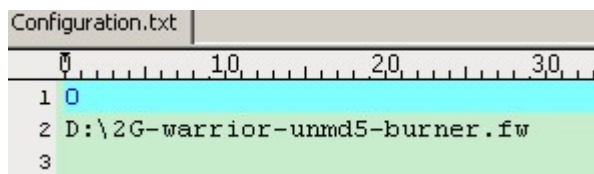
Put the SD card in a Card Reader and plug in the USB slot.

5.4.2 Format SDCard

Format the sdcard with FAT32.

5.4.3 Modify Configuration.txt

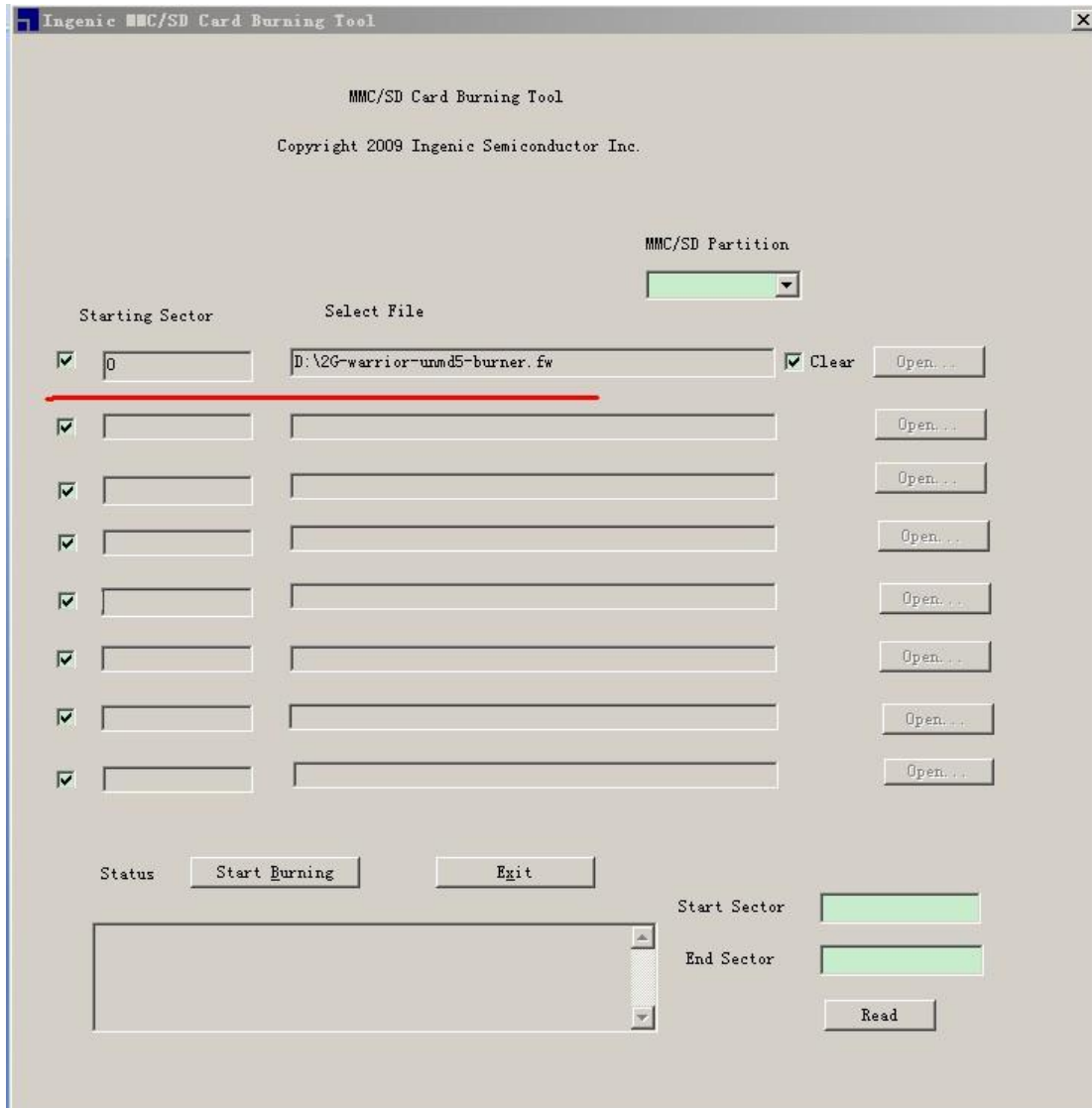
Modify the configure file, set the path direct your burner fw file. And make the address is "0".
For example, I put the burner file to the root directory on disk of D.



```
Configuration.txt |
0 10 20 30
1 0
2 D:\2G-warrior-unmd5-burner.fw
3
```

5.5 4 Open the SD CARD burning tool

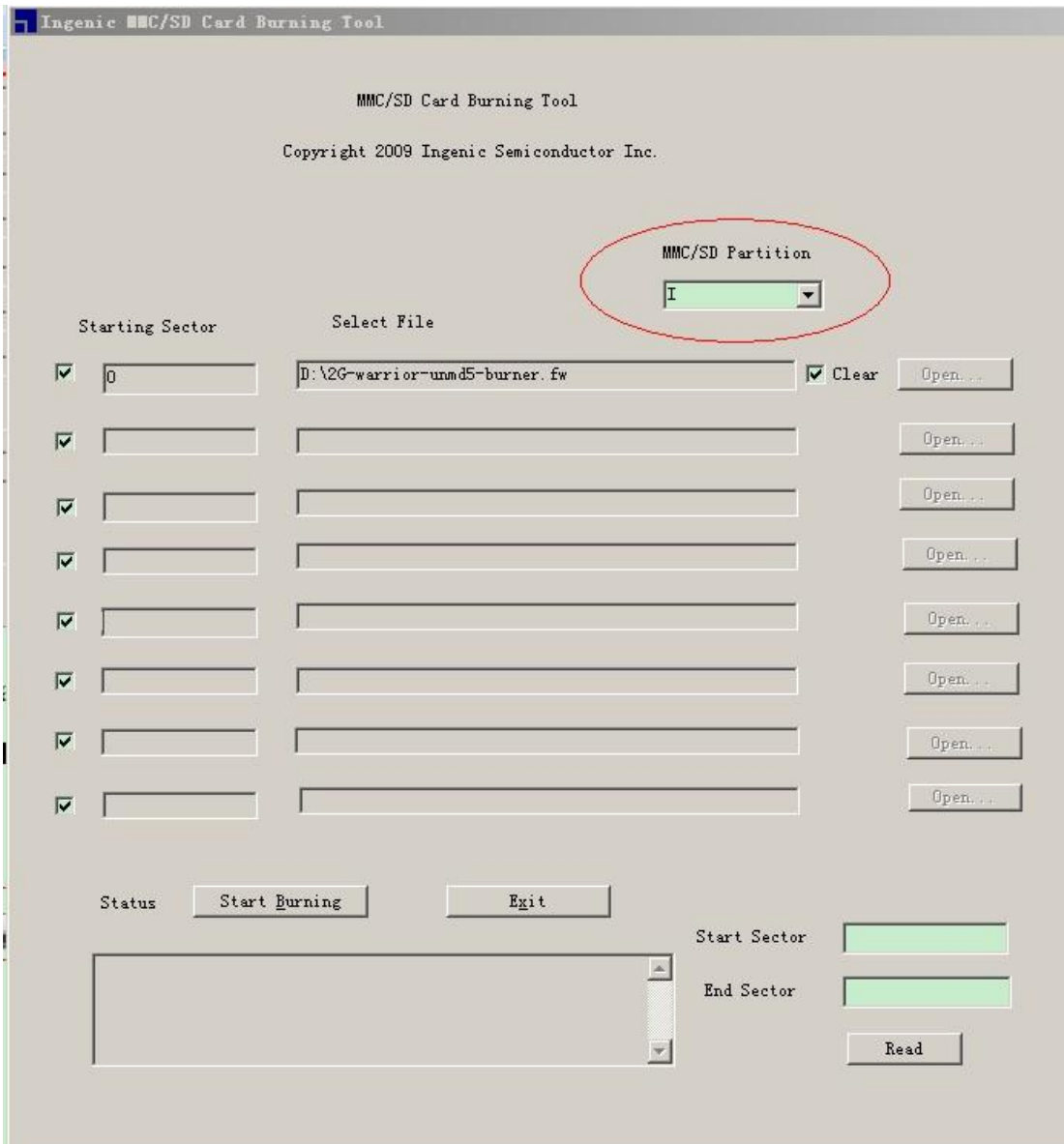
Open the **SDBoot.exe**



In Windows Vista and Windows 7 you will need to right click on this launcher and choose 'Run as Administrator'. Then open the tool and according to do.

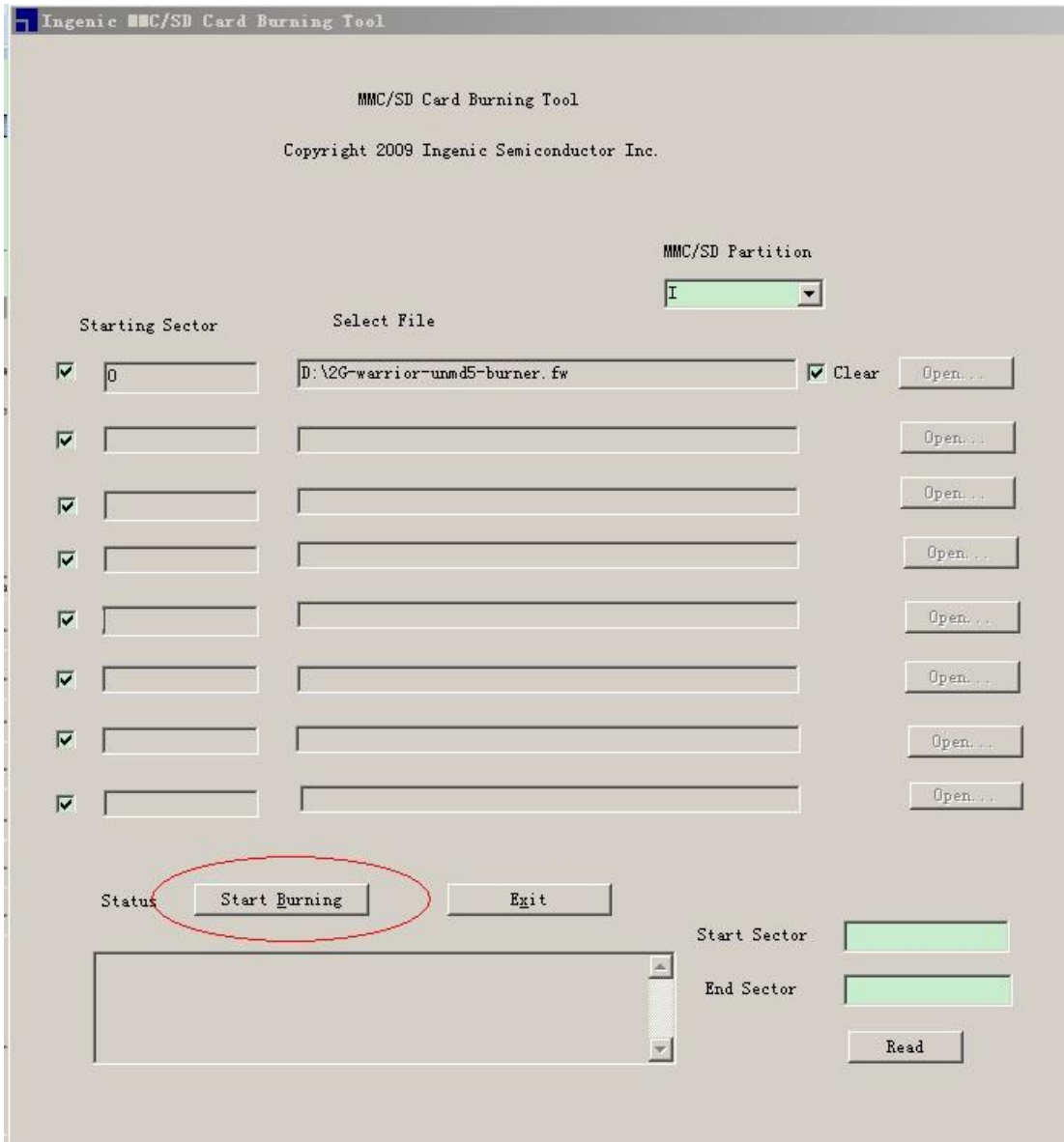
5.6 5 Make the MMC/SD Partition is the same with your SD card on your Windows XP.

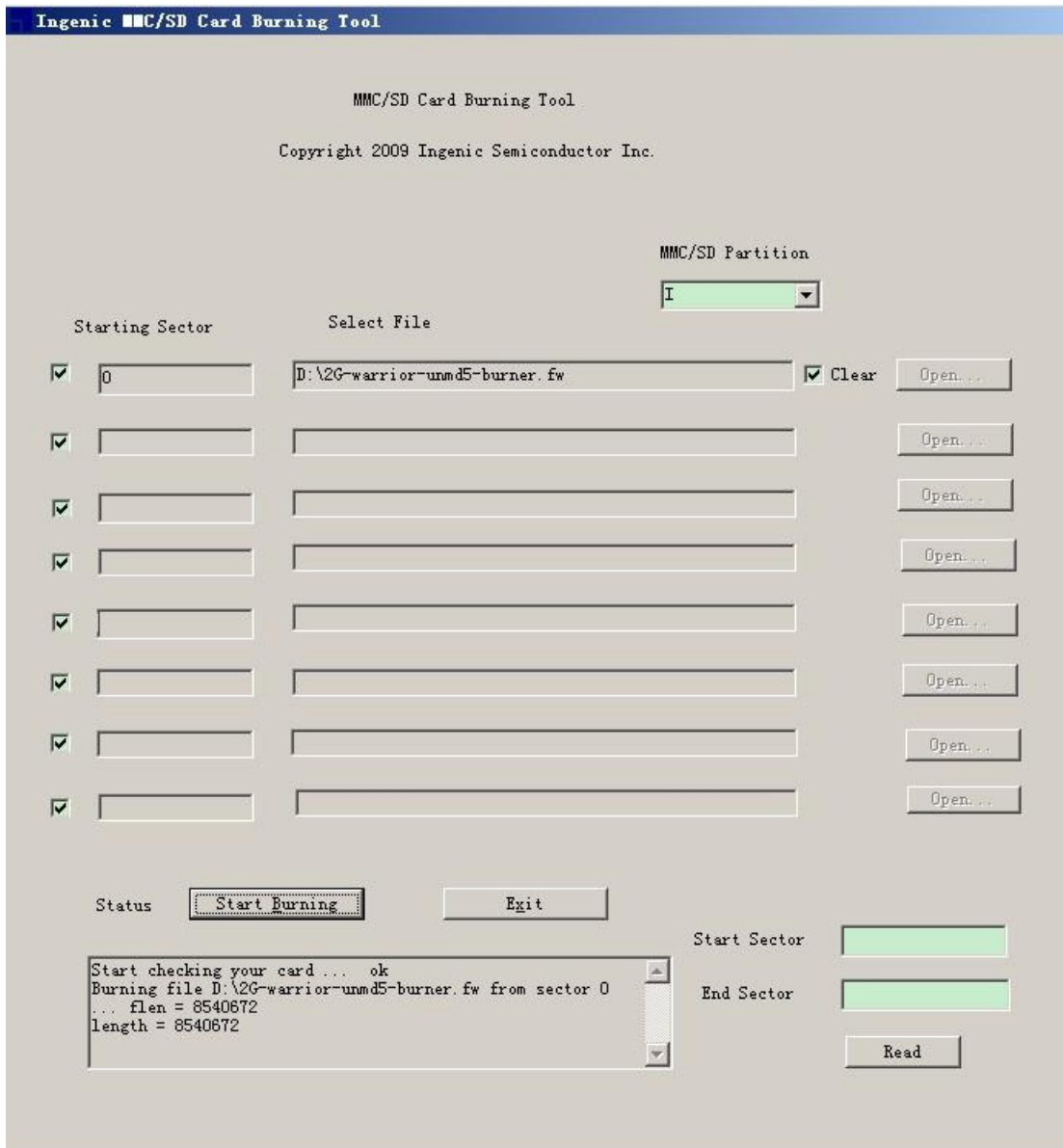




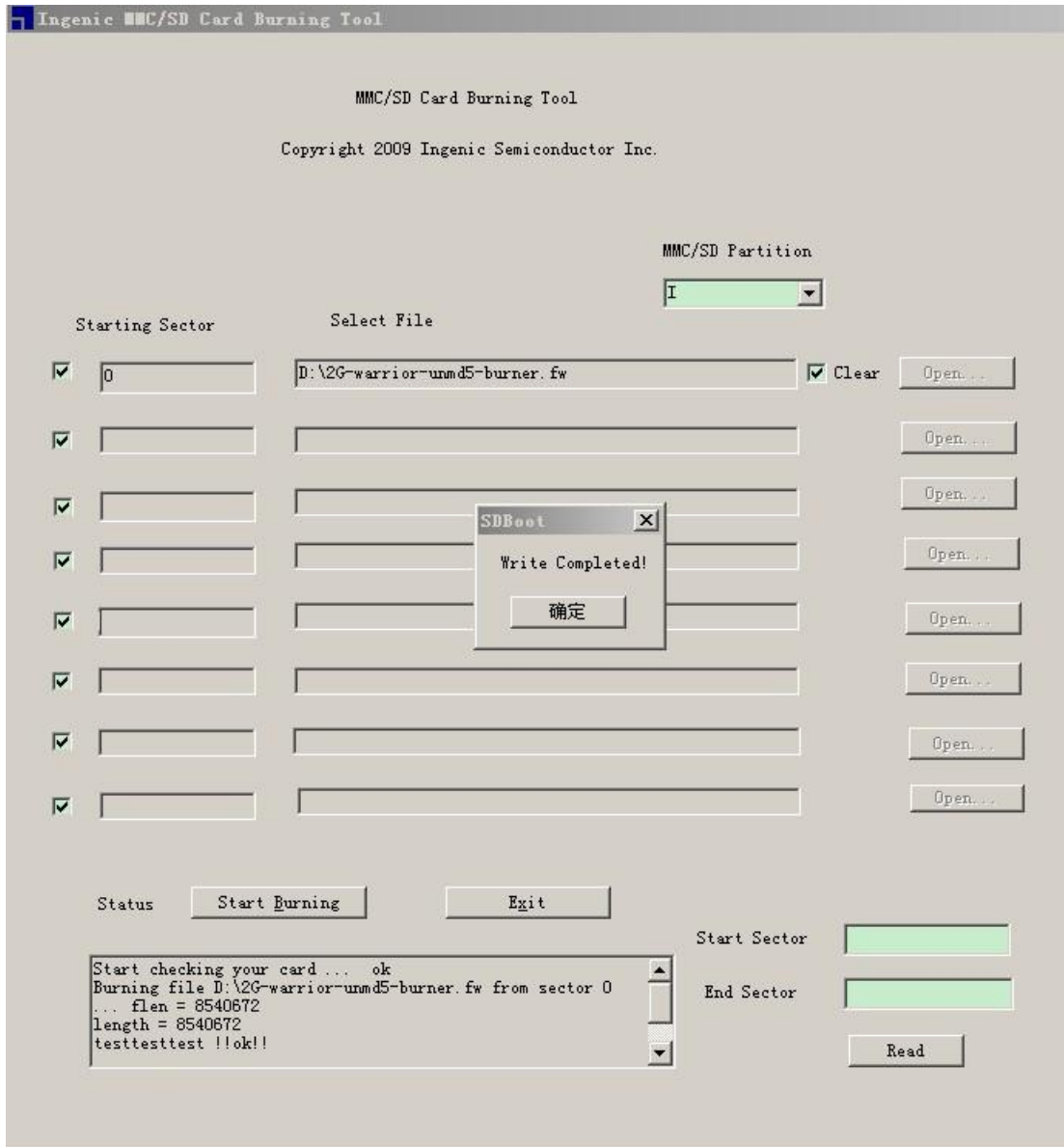
5.7 6 Start Burning

Click on the button of “Start Burning” on the SD Card Burning Tool.





5.8 7 Burning succeed



5.9 8 Take Card Reader

Take down the Card Reader from your Windows PC.

5.10 9 Quick format your SD Card

5.11 10 Configure your SD Card

Following the Part 2.

6 Board configuration

user@ubuntu:~/android-41\$ ls

```

abi                development      hardware         ndk              system
bionic            device          kernel           packages        update
bootable         docs            libcore         pdk             vendor
build            external        libnativehelper prebuilt
cts              frameworks      Makefile        prebuilts
dalvik          gdk             nda             sdk
  
```

There are three parts of board configuration in Android system.

- i) Bootloader
- ii) Kernel
- iii) Android system.

We will describe the configure files of JZ4780 development board. This will help you to set up your product platform environment.

6.1 Board configuration

Bootloader source locate at android-41/bootable/bootloader/xboot

```

./Makefile                /*Make info and config info of product: for example
                          *grus_nand_config
                          */
./include/configs/grus.h; /* Board info: memory, nandflash, CPU frequency,
                          * UART, binary load address and GPIO set.
                          */
./boot/board/Makefile     /* Makefile of board */
./boot/board/grus/jz4780_grus_special.c /* the special board code, such as power
                          * controlling of LCD
                          */
./boot/board/grus/mbr.h   /*MBR info, only foremmc-nand or inand: partition
info */
  
```

6.2 Kernel Board configuration

Android linux kernel source locate at android-41/kernel

```

./arch/mips/configs/grus_release_defconfig /* Defconfig file for grus board*/
./arch/mips/xburst/soc-4780/board/grus /* Board level configuration directory*/
./arch/mips/xburst/soc-4780/board/grus/grus-nand.c
./arch/mips/xburst/soc-4780/board/grus/grus-regulator.c
./arch/mips/xburst/soc-4780/board/grus/grus-pm.c
./arch/mips/xburst/soc-4780/board/grus/grus-i2c.c
./arch/mips/xburst/soc-4780/board/grus/grus-lcd.c
  
```

```
./arch/mips/xburst/soc-4780/board/grus/grus-sound.c
./arch/mips/xburst/soc-4780/board/grus/grus-mmc.c
./arch/mips/xburst/soc-4780/board/grus/grus.h
./arch/mips/xburst/soc-4780/board/grus/grus-misc.c
```

6.3 Product configuration in system

Others directory are Android System Source

```
./build/target/product/security/grus.pk8 /* Product key for update */
./build/target/product/security/grus.pem
./build/target/product/security/grus.x509.pem
./build/target/board/grus /* Main system configuration of board*/
```