IDEA6410 Android User Manual

1. Introduction

1.1. About this Manual

This manual is intended to provide the user with an overview of the board and benefits, complete features specifications, and set up procedures. It contains important safety information as well.

1.2. Feedback and Update to this Manual

To help our customers make the most of our products, we are continually making additional and updated resources available on the Boardcon website (www.armdesigner.com).

These include manuals, application notes, programming examples, and updated software and hardware. Check in periodically to see what's new!

When we are prioritizing work on these updated resources, feedback from customers is the number one influence, If you have questions, comments, or concerns about your product or project, please no hesitate to contact us at support@armdesigner.com.

1.3. Limited Warranty

Boardcon warrants this product to be free of defects in material and workmanship for a period of one year from date of buy. During this warranty period Boardcon will repair or replace the defective unit in accordance with the following process:

A copy of the original invoice must be included when returning the defective unit to Boardcon. This limited warranty does not cover damages resulting from lighting or other power surges, misuse, abuse, abnormal conditions of operation, or attempts to alter or modify the function of the product.

This warranty is limited to the repair or replacement of the defective unit .In no event shall Boardcon be liable or responsible for any loss or damages, including but not limited to any lost profits, incidental or consequential damages, loss of business, or anticipatory profits arising from the use or inability to use this products.

Repairs make after the expiration of the warranty period are subject to a repair charge and the cost of return shipping. Please contact Boardcon to arrange for any repair service and to obtain repair charge information.

1. Wok Environment

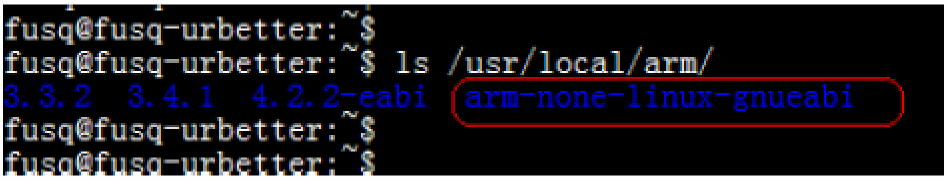
Version: Android_V0.19
Linux Working Environment: Ubuntu-9.04

1.1. Install Cross-compile

. Open Android_v0.19\cross_compile\ to copy file:
arm-none-linux-gnueabi-arm-2008q3-72-for-linux.tar.bz2 to the place of running Linux PC.
Note that “/home/fusq/test” is working directory (fusq is user name of linux PC).
. Under the directory /usr/local/arm,
install arm-none-linux-gnueabi-arm-2008q3-72-for-linux.tar.bz2.
The commend is: fusq@fusq-urbetter:～/test$
tar jxvf arm-none-linux-gnueabi-arm-2008q3-72-for-linux.tar.bz2 –C /
Note: The default path is /usr/local/arm/, do not need to assign again.

1.2. Check the complies installation status

Please see below picture-1



```
fusq@fusq-urbetter:~$
fusq@fusq-urbetter:~$ ls /usr/local/arm/
3.3.2  3.4.1  4.2.2-eabi  arm-none-linux-gnueabi
fusq@fusq-urbetter:~$
fusq@fusq-urbetter:~$
```

Picture-1

From above picture-1 arm-none-linux-gnueabi was successfully installed under the directory /usr/local/arm/.

2. Compile u-boot

There are two kinds of u-boot, one is u-boot-movi.bin that is ported in SD card, another one is u-boot-nand.bin that is ported in Nandflash for Nand Flash booting use.

2.1. Compile u-boot-movi.bin

The file u-boot-movi.bin is at the directory /image/, only providing bin file, no source code.

2.2. Compile u-boot-nand.bin

Please copy the file “android_v0.19\u-boot\u-boot-1.1.6-ut-s3c6410-nand.tar.gz” to /home/fusq/test, and decompress “u-boot-1.1.6-ut-s3c6410-nand.tar.gz” to the current directory, then execute below commands:

```
fusq@fusq-urbetter:～/test$ tar zxvfu-boot-1.1.6-ut-s3c6410-nand.tar.gz
fusq@fusq-urbetter:～/test$ cd u-boot-1.1.6-ut-s3c6410-nand/
fusq@fusq-urbetter:～/test$ make clean
fusq@fusq-urbetter:～/test$ make smdk6410_config
fusq@fusq-urbetter:～/test$ make
fusq@fusq-urbetter:～/test$ ./maknand
```

The u-boot-nand.bin will be made under the current directory u-boot-nand.bin

3. Compile Kernel

There are two Kernel, One is zImage-fix, another one is zImage-fix-nand.

. zImage-fix was compiled from Linux-2.6.29.1-for-burn-android.tar.gz. The Kernel is used to burn image from SD Card.

. zIamge-fix-nand was compiled from linux-2.6.29-android.tar.gz. The kernel is used for Nand Flash startup.

3.1. zImage-fix

Please copy the file “android-0.9_v0.19\kernel\linux-2.6.29.1-for-burn-android.tar.gz” to /home/fusq/test, and decompress “linux-2.6.29.1-for-burn-android.tar.gz” to the current directory, then execute below commands:

```
fusq@fusq-urbetter:～/test$ tar zxvf linux-2.6.29.1-for-burn-android.tar.gz
fusq@fusq-urbetter:～/test$ cd linux-2.6.29.1-for-burn-android
fusq@fusq-urbetter:～/test$ make clean
fusq@fusq-urbetter:～/test$ make menuconfig
fusq@fusq-urbetter:～/test$ make
fusq@fusq-urbetter:～/test$ ./fix-image
```

The zImage-fix will be made under the current directory \arch\arm\boot\

3.2. zImage-fix-nand

Please copy the file “android-0.9_v0.19\kernel\linux-2.6.29.1-android.tar.gz” to /home/fusq/test, and decompress “linux-2.6.29.1-android.tar.gz” to the current directory, then execute below commands:

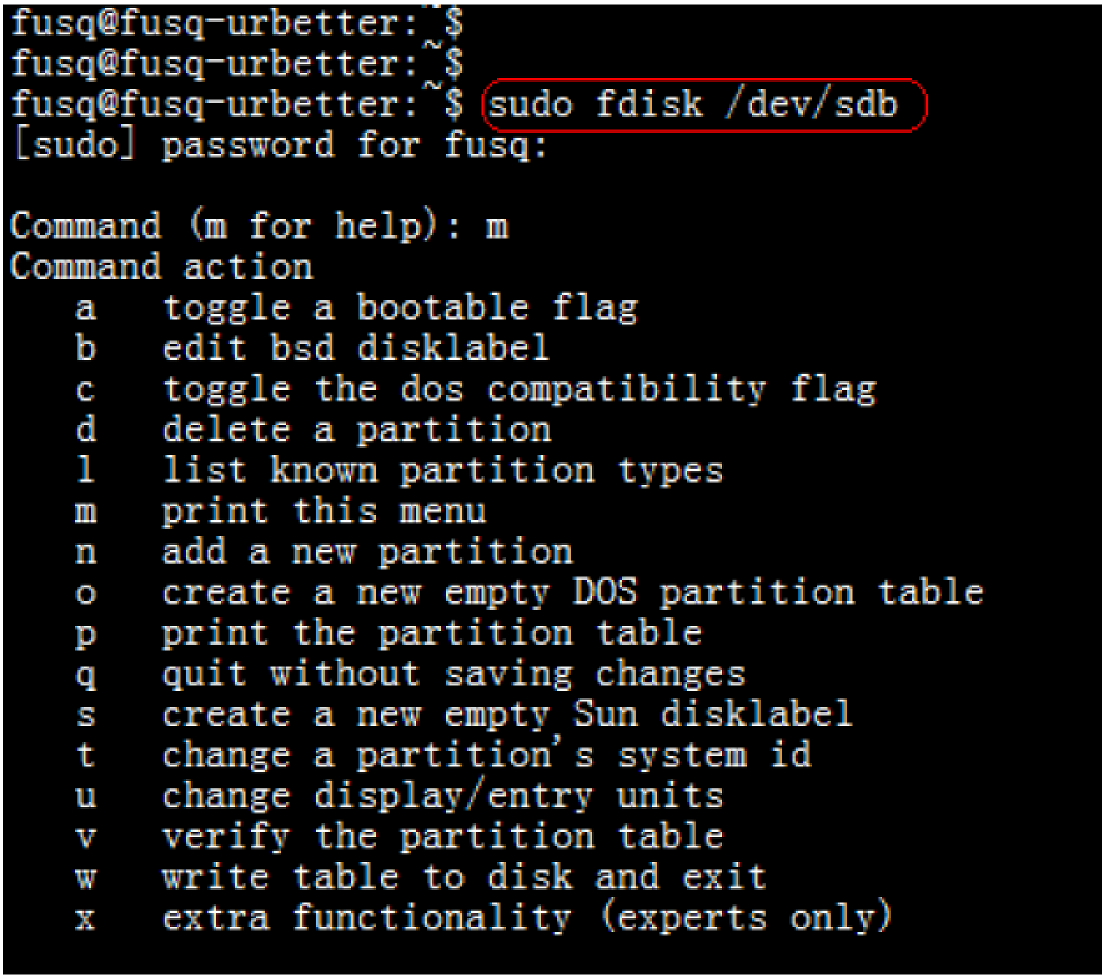
fusq@fusq-urbetter:～/test$ tar zxvf linux-2.6.29.1-android.tar.gz
fusq@fusq-urbetter:～/test$ cd linux-2.6.29.1-android
fusq@fusq-urbetter:～/test$ make clean
fusq@fusq-urbetter:～/test$ make menuconfig
fusq@fusq-urbetter:～/test$ make
fusq@fusq-urbetter:～/test$ ./fix-image
fusq@fusq-urbetter:～/test$ cd arch/arm/boot
fusq@fusq-urbetter:～/test$ mv zImage-fix zImage-fix-nand

The zImage-fix-nand will be made under the current directory \arch\arm\boot\

4. Burn Image

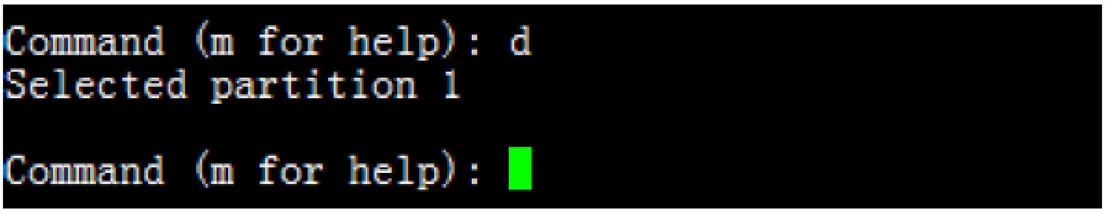
4.1. Burn u-boot-movi.bin to SD Card

Prepare one piece of SD Card above 1GB, and divide SD Card into two parts under Linux Environment, the first part around 100MB is FAT format, the second part around 800MB is EXT3 format. Below picture-2 and picture-3 show the operation.



```
fusq@fusq-urbetter:~$
fusq@fusq-urbetter:~$
fusq@fusq-urbetter:~$ sudo fdisk /dev/sdb
[sudo] password for fusq:

Command (m for help): m
Command action
   a   toggle a bootable flag
   b   edit bsd disklabel
   c   toggle the dos compatibility flag
   d   delete a partition
   l   list known partition types
   m   print this menu
   n   add a new partition
   o   create a new empty DOS partition table
   p   print the partition table
   q   quit without saving changes
   s   create a new empty Sun disklabel
   t   change a partition's system id
   u   change display/entry units
   v   verify the partition table
   w   write table to disk and exit
   x   extra functionality (experts only)
```

Picture-2



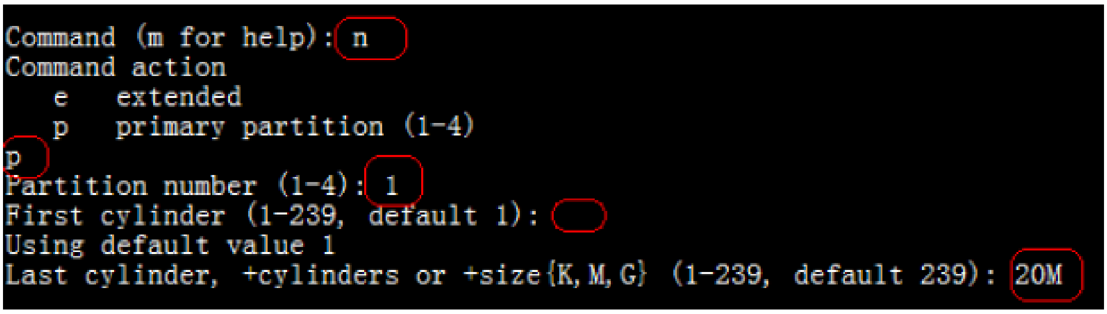
```
Command (m for help): d
Selected partition 1

Command (m for help):
```

Picture-3

The next step is to create the first SD Card part.

. Input “n”, Enter → input “p’, Enter → input “1”, Enter “20M” ,Enter

Below picture-4 shows the steps.



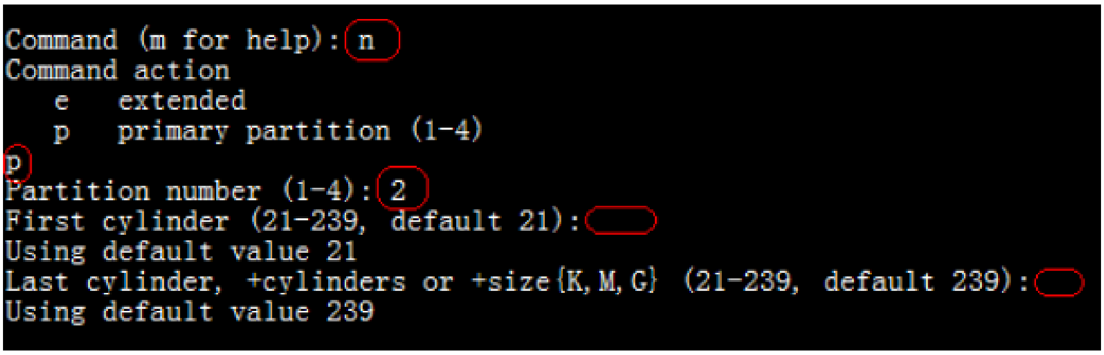
```
Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-239, default 1):
Using default value 1
Last cylinder, +cylinders or +size{K,M,G} (1-239, default 239): 20M
```

Picture-4

The next step is to create the second SD Card part

. Input “n”, Enter → input “p’, Enter → input “2”, Enter , Enter

Below picture-6 shows the steps.



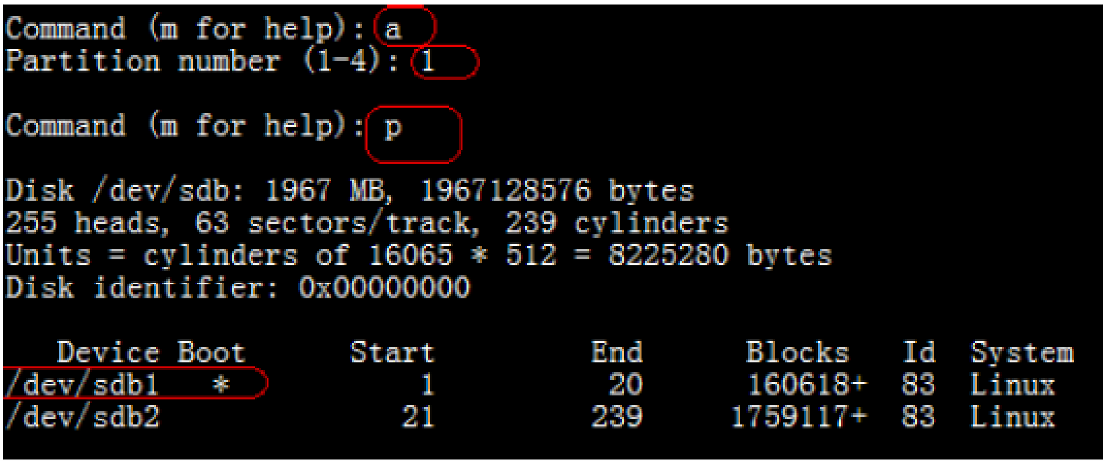
```
Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 2
First cylinder (21-239, default 21):
Using default value 21
Last cylinder, +cylinders or +size{K,M,G} (21-239, default 239):
Using default value 239
```

Picture-5

The next step is to mark the first SD Card part.

. Input “a”, Enter → input “1’, Enter → input “p”, Enter

Below picture-5 shows the steps.

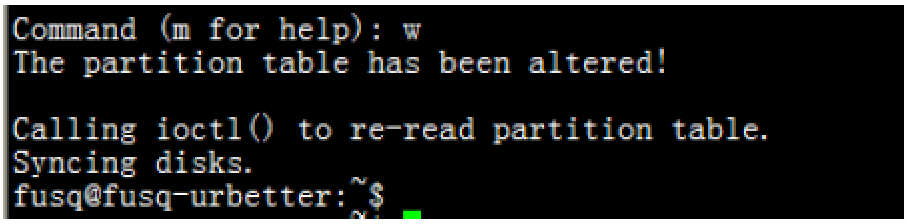


```
Command (m for help): a
Partition number (1-4): 1

Command (m for help): p

Disk /dev/sdb: 1967 MB, 1967128576 bytes
255 heads, 63 sectors/track, 239 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0x00000000

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1   *           1          20      160618+  83  Linux
/dev/sdb2              21         239     1759117+  83  Linux
```

Picture-6

The next step is to write partition table

. Input “w”, Enter. Below picture-7 shows the step.



```
Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
fusq@fusq-urbetter:~$
```

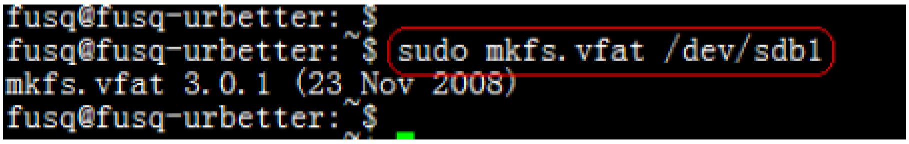
Picture-7

By now, the two parts of SD Card were divided.

Note: After part division, the SD Card normally will automatically mount. Before format the SD Card, please make sure that the SD Card is on demounted status.

The next step is to format the two parts.

. Format the first part into .vfat format, execute command: sudo mkfs.vfat /dev/sdb1, Below picture-8 shows the step.

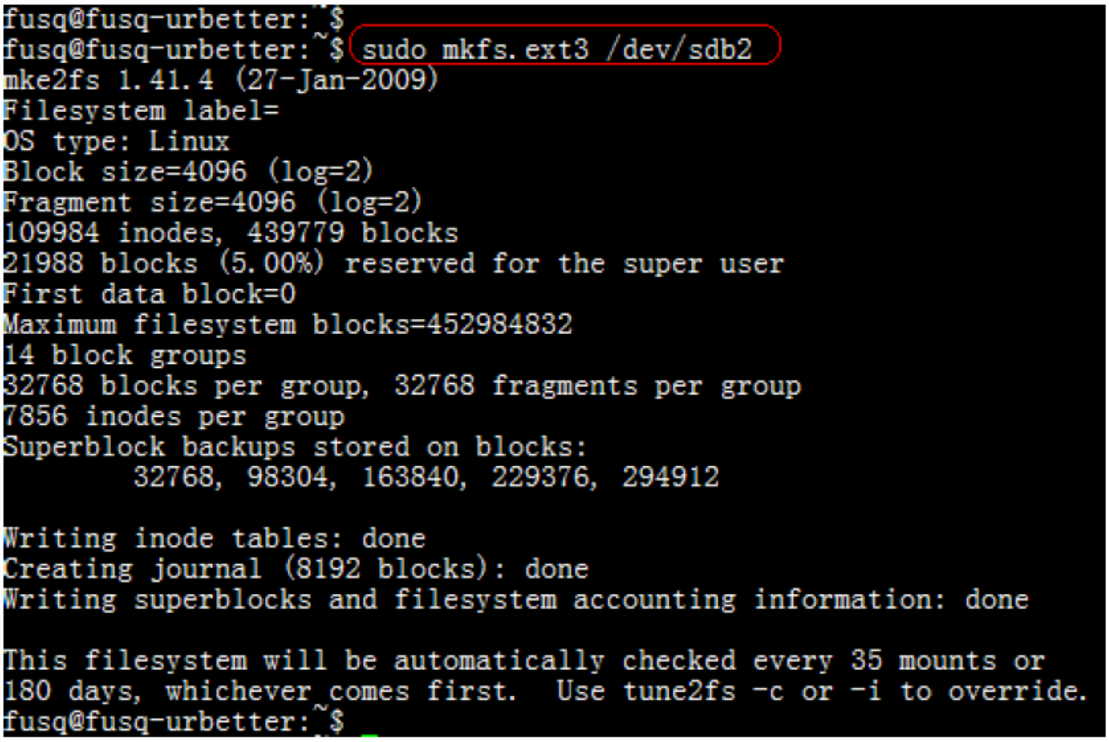


```
fusq@fusq-urbetter:~$
fusq@fusq-urbetter:~$ sudo mkfs.vfat /dev/sdb1
mkfs.vfat 3.0.1 (23 Nov 2008)
fusq@fusq-urbetter:~$
```

Picture-8

The next step is to format the first part into .ext3 format, execute command:

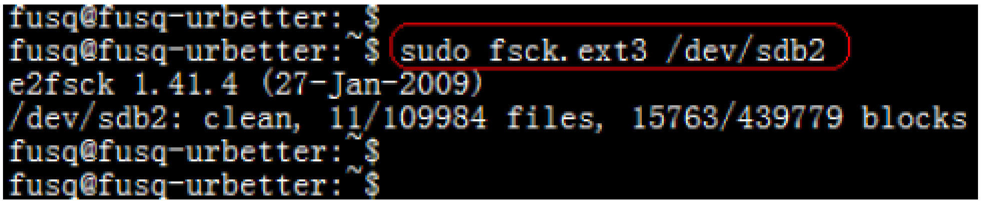
Sudo mkfs.ext3 /dev/sdb2. Below picture-9 shows the step.



```
fusq@fusq-urbetter:~$
fusq@fusq-urbetter:~$ sudo mkfs.ext3 /dev/sdb2
mke2fs 1.41.4 (27-Jan-2009)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
109984 inodes, 439779 blocks
21988 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=452984832
14 block groups
32768 blocks per group, 32768 fragments per group
7856 inodes per group
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912

Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 35 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
fusq@fusq-urbetter:~$ _
```

Picture-9

The next step is to check the file system of the second part, execute command: Sudo fsck.ext3 /dev/sdb2, below picture-10 shows the step.



```
fusq@fusq-urbetter:~$
fusq@fusq-urbetter:~$ sudo fsck.ext3 /dev/sdb2
e2fsck 1.41.4 (27-Jan-2009)
/dev/sdb2: clean, 11/109984 files, 15763/439779 blocks
fusq@fusq-urbetter:~$
fusq@fusq-urbetter:~$
```

Picture-10

4.2. Burn “u-boot-movi.bin” and “zImage-fix” into SD Card

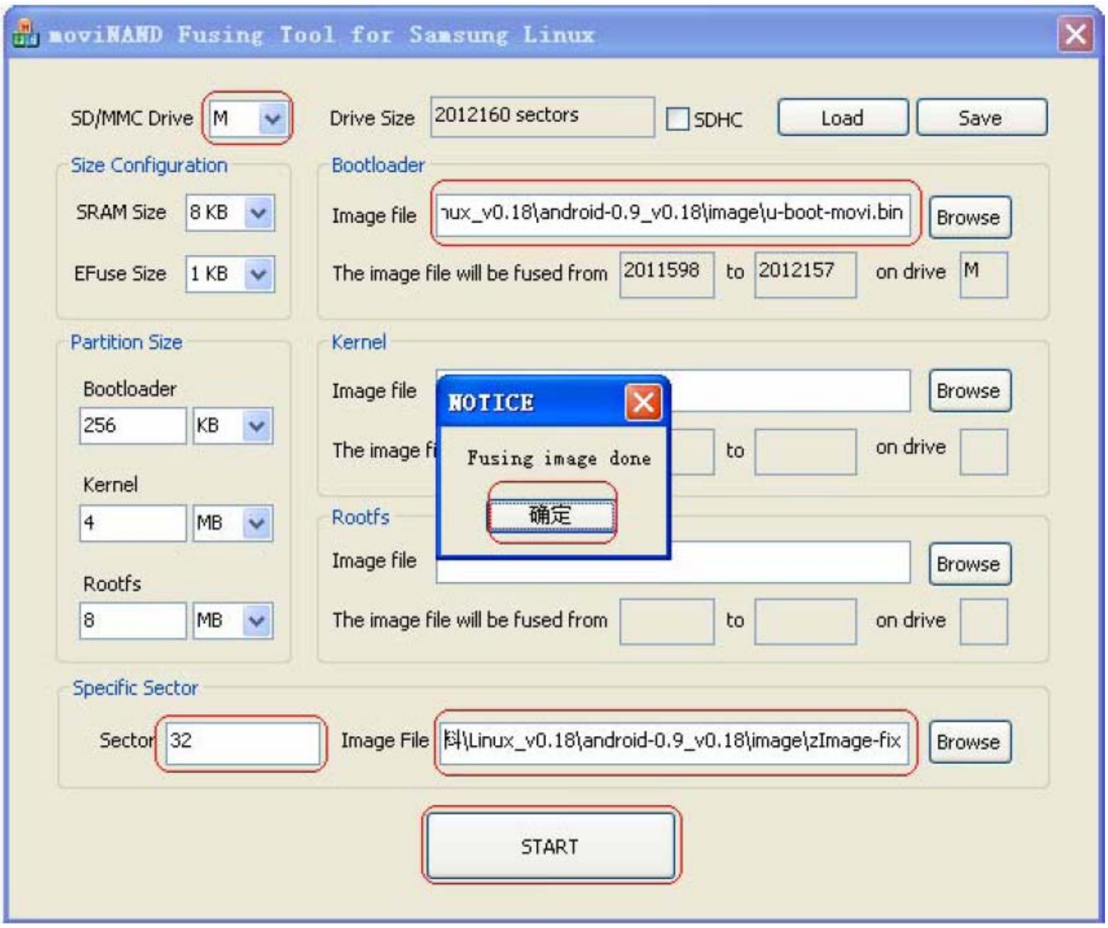
. Please insert SD Card to the SD Card Reader, then connect PC.

. Under Windows XP working environment, open “android-0.9_v0.19\image\moviNAND_Fusing_Tool.ext”

. At the place of “SD/MMC Driver”, Please select the SD Card’s mapped disc path under Windows XP, please see below picture-11.

. Click “ Browse” to add “u-boot-move.bin” at the place of Bootloader\Image file, please see below picture-11.

. Input “32” at place of “Specific Sector\sector” , then Click ‘Browse” to add “zImage-fix”. Please see below picture-11.



SD/MMC Drive: M
Drive Size: 2012160 sectors
□ SDHC

Size Configuration
SRAM Size: 8 KB
EFuse Size: 1 KB

Bootloader
Image file: ıux_v0.18\android-0.9_v0.18\image\u-boot-movi.bin
The image file will be fused from 2011598 to 2012157 on drive M

Partition Size
Bootloader: 256 KB
Kernel: 4 MB
Rootfs: 8 MB

Kernel
Image file:
The image fi… to … on drive

NOTICE
Fusing image done
确定

Rootfs
Image file:
The image file will be fused from … to … on drive

Specific Sector
Sector: 32
Image File: 科\Linux_v0.18\android-0.9_v0.18\image\zImage-fix

Picture-11

Finished above steps, then click the key " START", it will pop up "Fusing image done" if burn successfully.

Above steps was to burn "u-boot-movi.bin" and "zImage-fix" into SD Card. The next step is to burn file system into SD Card.

4.3. Copy files into the part of ext3 of the SD Card

The first step is to insert the SD Card into the Linux host PC.

Then copy the files " android-0.9_v0.19\filesystem\androidfs-sdk_m5-rc15-fix.tar" , "android-0.9_v0.19\image\zImage-fix-nand" and "u-boot-nand.bin" total three files into the part of ext3 of the SD Card.

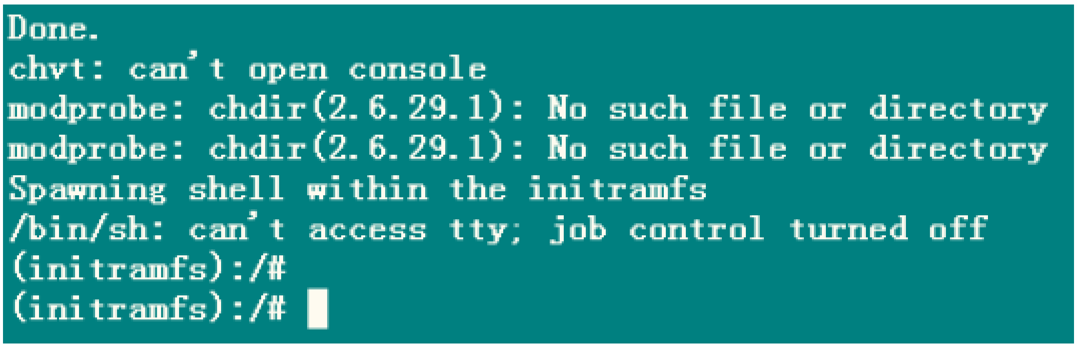
5. Set SD Card as startup mode

Set the button "SW1" on the IDEA6410 board as SD boot mode
"1234" corresponding value is : "1111"

Insert SD Card into SD Card interface of the IDEA6410 board, then start the board.

6. Burn “image’” into Nand Flash

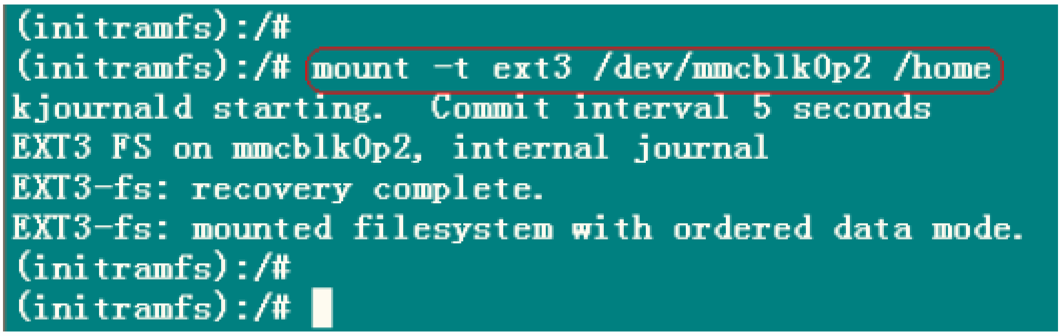
After the system started up, please wait the OS into command status. Please see below picture-12 about the step.



```
Done.
chvt: can't open console
modprobe: chdir(2.6.29.1): No such file or directory
modprobe: chdir(2.6.29.1): No such file or directory
Spawning shell within the initramfs
/bin/sh: can't access tty; job control turned off
(initramfs):/#
(initramfs):/#
```

Picture-12

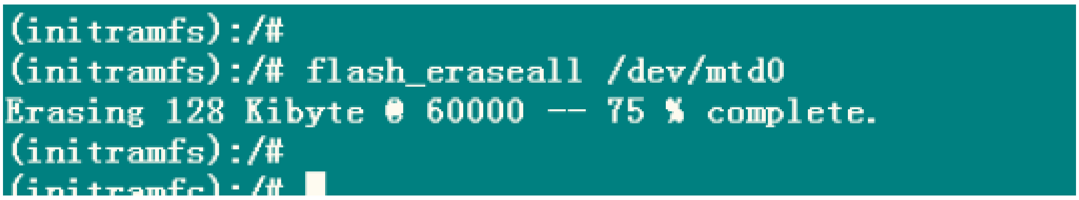
Make the part of ext3 mount under directory of /home, the command;
Mount –t ext3/dev/mmcblk0p2 /home. Below picture-13 shows the step.



```
(initramfs):/#
(initramfs):/# mount -t ext3 /dev/mmcblk0p2 /home
kjournald starting.  Commit interval 5 seconds
EXT3 FS on mmcblk0p2, internal journal
EXT3-fs: recovery complete.
EXT3-fs: mounted filesystem with ordered data mode.
(initramfs):/#
(initramfs):/#
```

Picture-13

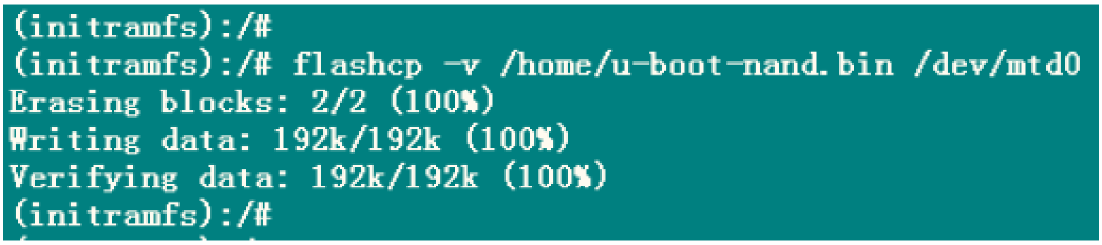
Erase the part 0 of Nand Flash, the command is: flash_eraseall /dev/mtd0. Below picture-14 shows the step.



```
(initramfs):/#
(initramfs):/# flash_eraseall /dev/mtd0
Erasing 128 Kibyte @ 60000 -- 75 % complete.
(initramfs):/#
(initramfs):/#
```

Picture-14

Burn the file “ u-boot-nand.bin” into the part 0 of the Nand Flash, the command is:
Flashcp –v /home/u-boot-nand.bin /dev/mtd0. Below picure-15 shows the step.

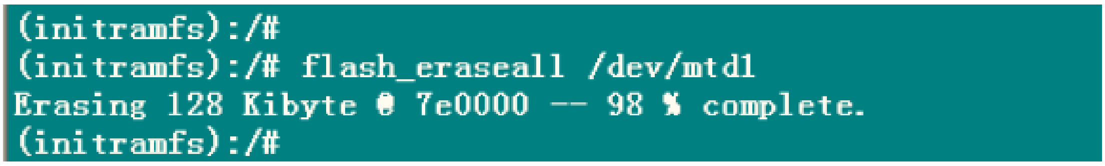


```
(initramfs):/#
(initramfs):/# flashcp -v /home/u-boot-nand.bin /dev/mtd0
Erasing blocks: 2/2 (100%)
Writing data: 192k/192k (100%)
Verifying data: 192k/192k (100%)
(initramfs):/#
```

Picture-15

Erase the part 1 of the Nand Flash, the command is: flash_eraseall /dev/mtd1

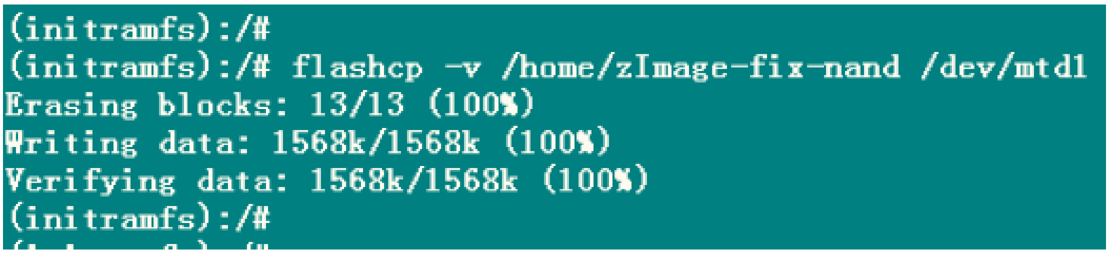
Below picture-16 shows the step.



```
(initramfs):/#
(initramfs):/# flash_eraseall /dev/mtd1
Erasing 128 Kibyte @ 7e0000 -- 98 % complete.
(initramfs):/#
```

Picture-16

Burn the file “zImage-fix-nand” into the part 1 of the Nand Flash, the command is: Flashcp –v /home/zImage-fix-nand/dev/mtd1. Below picure-17 shows the step.

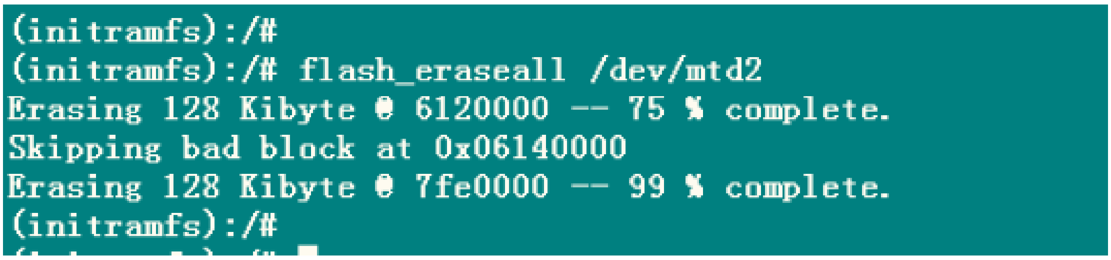


```
(initramfs):/#
(initramfs):/# flashcp -v /home/zImage-fix-nand /dev/mtd1
Erasing blocks: 13/13 (100%)
Writing data: 1568k/1568k (100%)
Verifying data: 1568k/1568k (100%)
(initramfs):/#
```

Picture-17

Erase the part 2 of the Nand Flash, the command is: flash_eraseall /dev/mtd2

Below picture-18 shows the step.

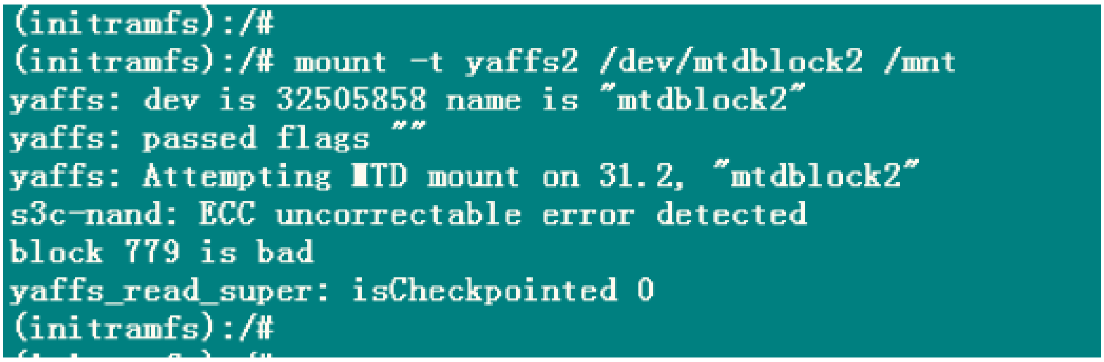


```
(initramfs):/#
(initramfs):/# flash_eraseall /dev/mtd2
Erasing 128 Kibyte @ 6120000 -- 75 % complete.
Skipping bad block at 0x06140000
Erasing 128 Kibyte @ 7fe0000 -- 99 % complete.
(initramfs):/#
```

Picture-18

Mount file system

The command is: mount –t yaffs2 /dev/mtdblock2 /mnt. Below picture-19 shows the step.



```
(initramfs):/#
(initramfs):/# mount -t yaffs2 /dev/mtdblock2 /mnt
yaffs: dev is 32505858 name is "mtdblock2"
yaffs: passed flags ""
yaffs: Attempting MTD mount on 31.2, "mtdblock2"
s3c-nand: ECC uncorrectable error detected
block 779 is bad
yaffs_read_super: isCheckpointed 0
(initramfs):/#
```

Picture-19

Decompress the file system into the part2 of the Nand Flash, the command is:
Tar xvf /home/androidfs-sdk_m5-rc15-fix.tar –C /mmt

7. Set startup mode as Nand Flash boot

Set the button “SW1” on the IDEA6410 board as Nand Flash boot mode.
1234 corresponding value is : 1100.