

IDEA6410 Linux 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 do not 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 lightning 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 made 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.

IDEA6410 Linux User Manual	1
1. Wok Environment.....	3
1.1. Install Cross-compile	3
1.2. Check the complies installation status	3
2. Compile u-boot	4
2.1. Compile u-boot-movi.bin	4
2.2. Compile u-boot-nand.bin	4
3. Compile Kernel.....	4
4. Burn Image	5
4.1. Burn u-boot-movi.bin to SD Card.....	5
4.2. Download u-boot-nand.bin into Nand Flash	7
4.2. Set “bootargs” and “bootcmd”	9
4.3. Burn zImage into Nand Flash 0x100000.....	10
4.4. Burn “rootfs_qt.cramfs” into Nand Flash	11
5. Interface Drivers test	12
5.1. SD Card test	12
5.2. Audio Test	14
5.3. Video test.....	15
5.4. Ethernet Interface test	16
5.5. 7inch TFT LCD test	18
5.6. VGA test.....	18
5.7. Camera test.....	19
5.8. Serial interface test.....	20

1. Wok Environment

- . Version: Idea6410_Linux2.6.24_V0.19
- . Linux Working Environment: Ubuntu-9.04

1.1. Install Cross-compile

. Open Linux-2.6.24_v0.19\cross_compile\ to copy files:
arm-none-linux-gnueabi-arm-2008q3-72-for-linux.tar.bz2, cross-4.2.2-eabi.tar.bz2,
cross-3.3.2.tar.bz2, cross-3.4.1.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 command is: fusq@fusq-urbetter: ~ /test\$ **tar jxvf arm-none-linux-gnueabi-arm-2008q3-72-for-linux.tar.bz2**

Note: The default path is /usr/local/arm/, do not need to assign again.

. Under the directory /usr/local/arm, install cross-4.2.2-eabi.tar.bz2,

The command is: fusq@fusq-urbetter: ~ /test\$ **tar jxvf cross-4.2.2-eabi.tar.bz2 -C /usr/local/arm/**

. Under the directory /usr/local/arm, install cross-3.3.2.tar.bz2

The command is: fusq@fusq-urbetter: ~ /test\$ **tar jxvf cross-3.3.2.tar.bz2 -C /usr/local/arm**

. Under the directory /usr/local/arm, install cross-3.4.1.tar.bz2

The command is: fusq@fusq-urbetter: ~ /test\$ **tar jxvf cross-3.4.1.tar.bz2 -C /usr/local/arm**

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, the compile 3.3.2, 3.4.1, 4.2.2 – eabi, arm-none-linux-gnueabi were 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 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 "linux-2.6.24_v0.18\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 zxvf u-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

Please copy the file "linux-2.6.24_v0.19\uurbetter_linux-2.6.24.tar.gz" to /home/fusq/test, and decompress "urbetter_linux-2.6.24.tar.gz" to the current directory, then execute below commands:

```
fusq@fusq-urbetter:~/test$ tar zxvf urbetter_linux-2.6.24.tar.gz
fusq@fusq-urbetter:~/test$ cd linux-2.6.24/
fusq@fusq-urbetter:~/test$ make clean
fusq@fusq-urbetter:~/test$ make menuconfig
fusq@fusq-urbetter:~/test$ make
```

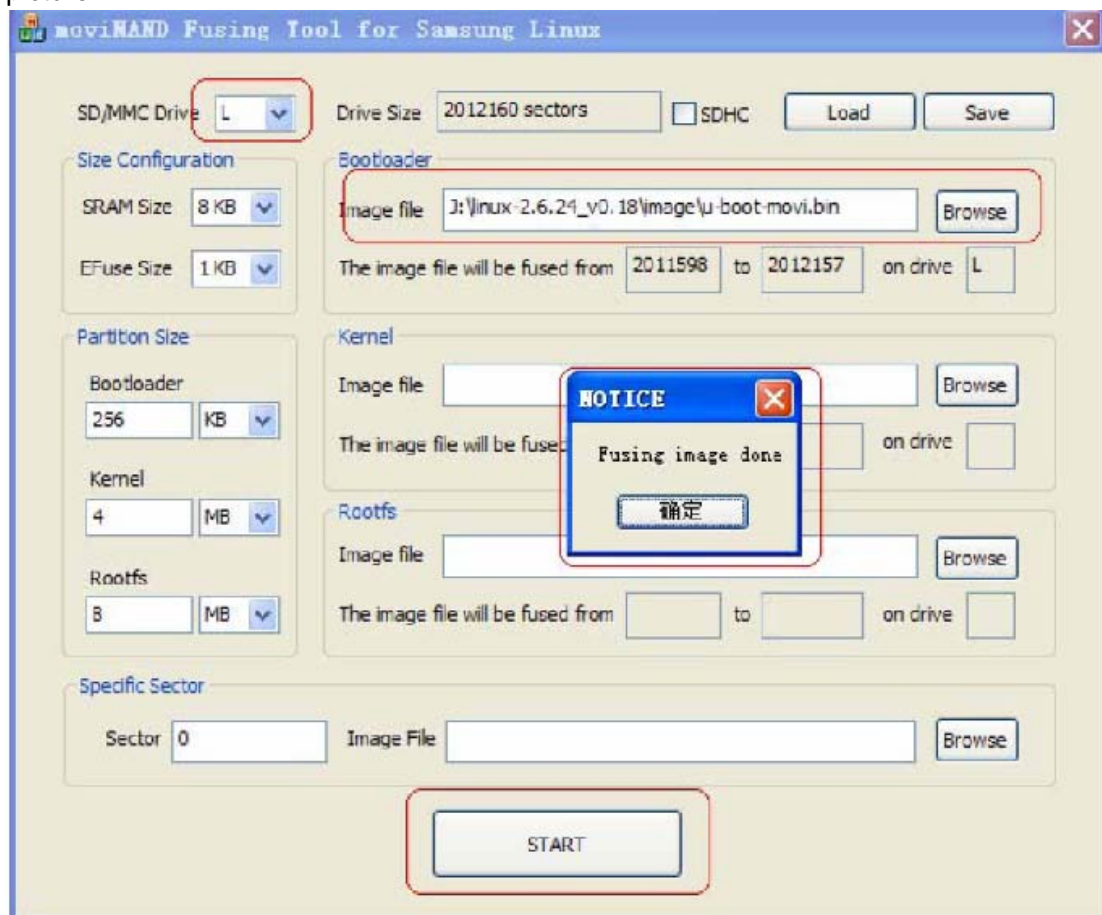
The zImage will be made under the current directory \arch\arm\boot\

4. Burn Image

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

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

- . Under Windows XP working environment,
open Linux-2.6.24_v0.19\image\moviNAND_Fusing_Tool.exe.
- . At the directory of “SD/MMC Driver”, Please select the SD Card’s mapped disc path under Windows XP, please see below picture-2.
- . At the place of “Image file”, click “Browse” to add “u-boot-movi.bin”, please see below picture-2



Picture-2

The next step is:

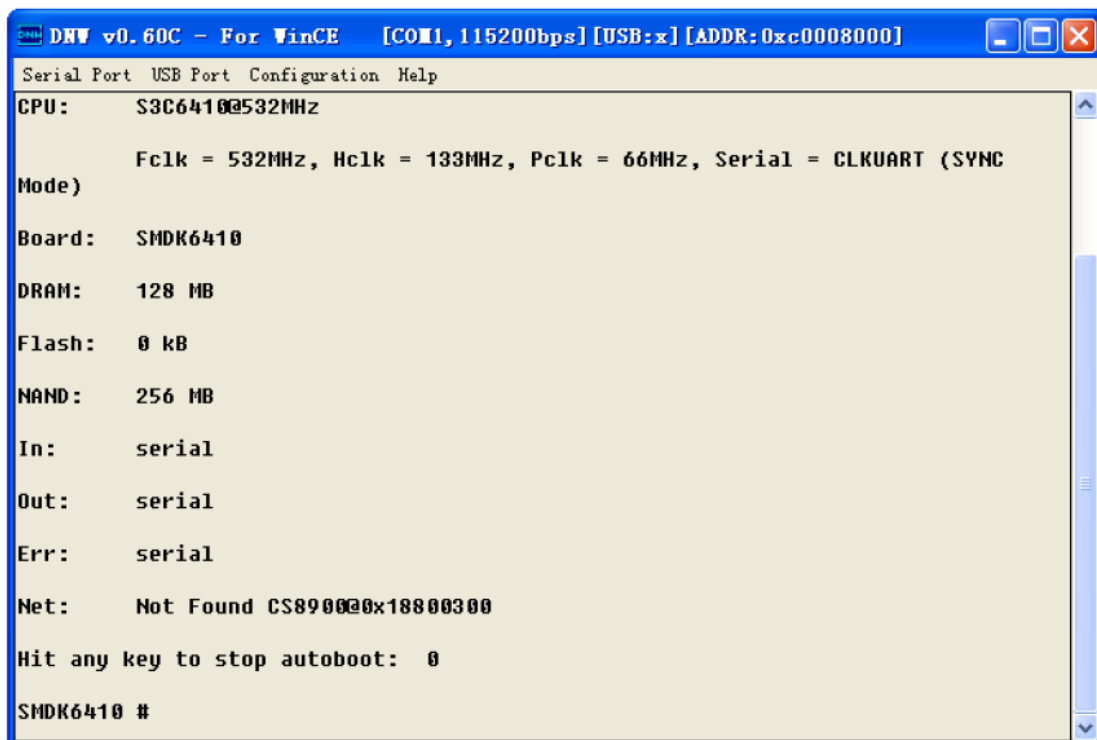
- . Click “START”, it will pop-up “Fusing image done” when burn successfully.
- . Set the button of “SW” of Idea6410 as SD boot mode.
The value of “1234” is “1111” (the SW1 pushing at “on” is 1)
- . Insert SD Card into SD Card connector on the board
- . Connecting serial cable, USB line, and Power supply
- . Open “Linux-2.6.24_v0.19\tools\dnw.exe

. Click "Serial Port" → "Connect", please see below picture-3



Picture-3

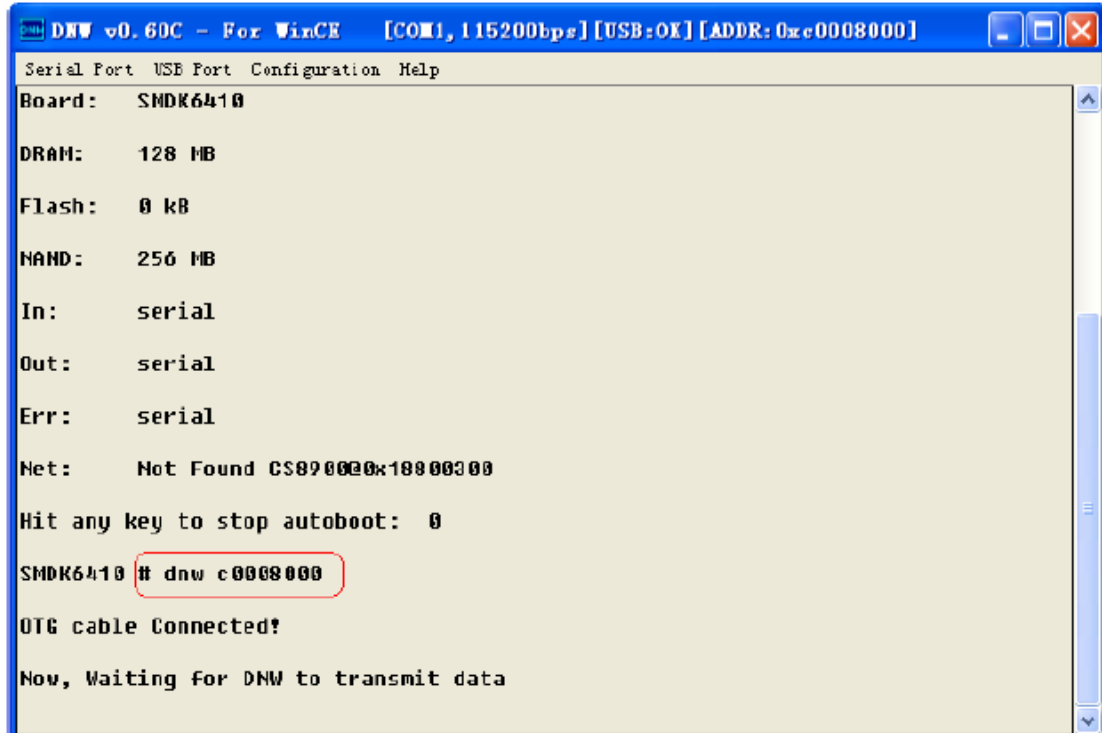
. Switch power on Idea6410, much printing information is coming soon, then click the key "SPACE" to enter u-boot. Please see below picture-4.



Picture-4

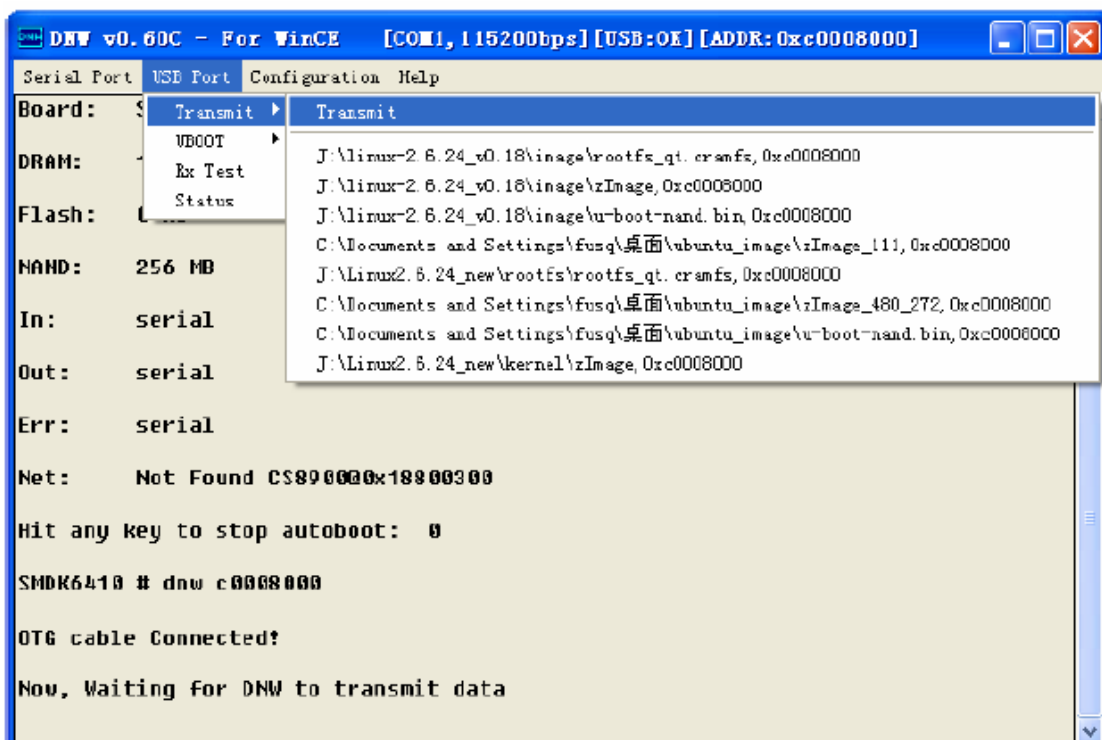
4.2. Download u-boot-nand.bin into Nand Flash

. Input command- `dnw c0008000` under DNW.exe then enter. Please see below picture-5



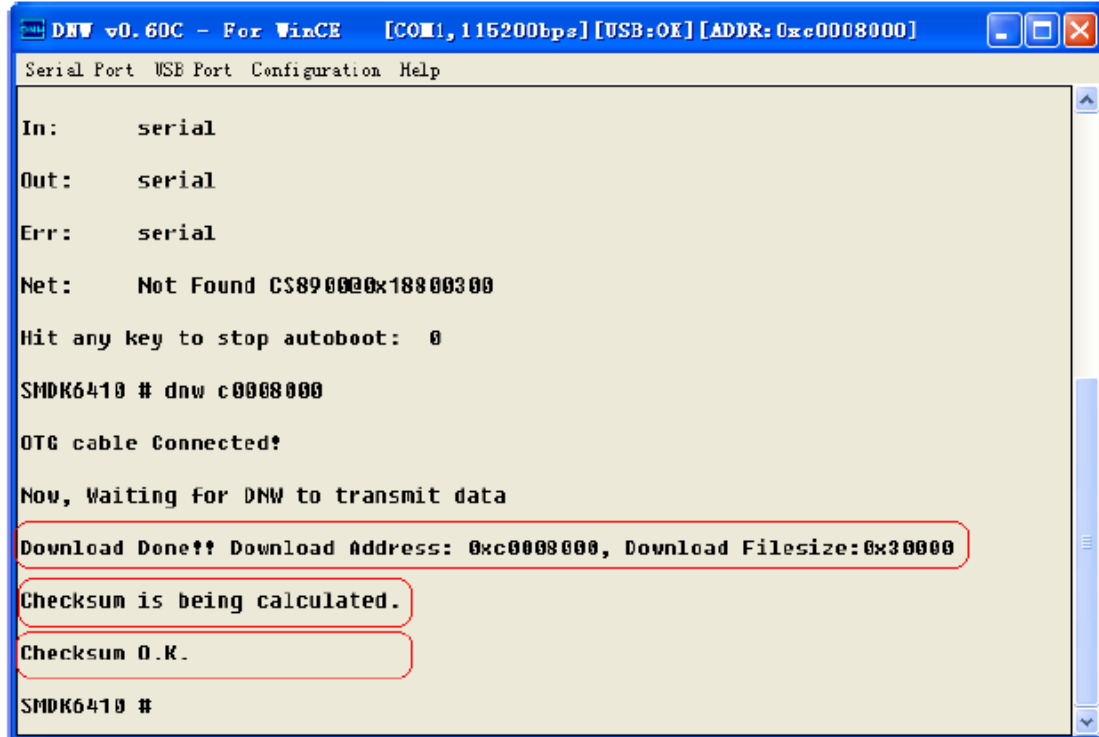
Picture-5

. Then click "USB Port → Transmit → Transmit", please see below picture-6



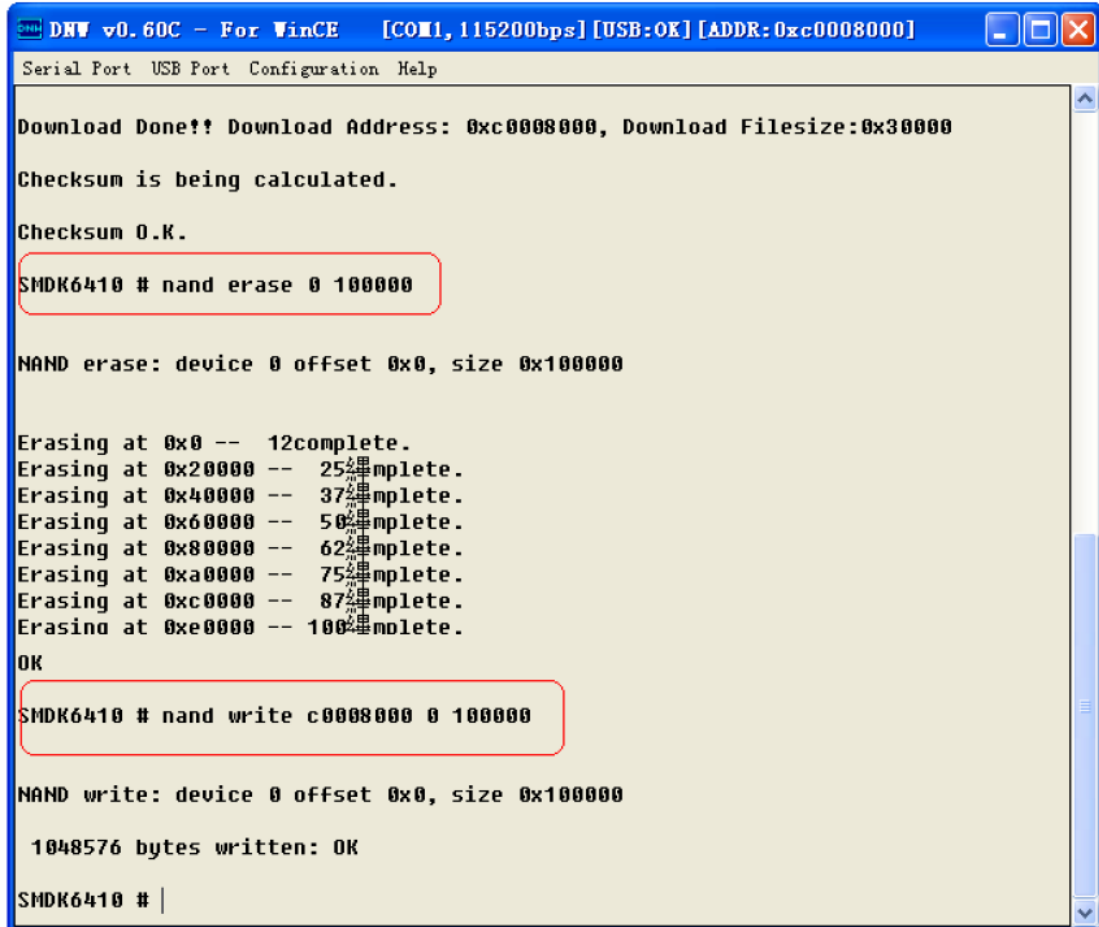
Picture-6

. Make “u-boot-nand.bin” read into address “ram 0xc0008000, please see below picture-7



Picture-7

. Make “u-boot-nand.bin” write into the address “nand flash 0x0”, please see below picture-8



```

DNW v0.60C - For WinCE [COM1, 115200bps] [USB:OK] [ADDR:0xc0008000]
Serial Port USB Port Configuration Help

Download Done!! Download Address: 0xc0008000, Download Filesize:0x30000
Checksum is being calculated.
Checksum O.K.
SMDK6410 # nand erase 0 100000

NAND erase: device 0 offset 0x0, size 0x100000

Erasing at 0x0 -- 12complete.
Erasing at 0x20000 -- 25complete.
Erasing at 0x40000 -- 37complete.
Erasing at 0x60000 -- 50complete.
Erasing at 0x80000 -- 62complete.
Erasing at 0xa0000 -- 75complete.
Erasing at 0xc0000 -- 87complete.
Erasing at 0xe0000 -- 100complete.
OK
SMDK6410 # nand write c0008000 0 100000

NAND write: device 0 offset 0x0, size 0x100000

1048576 bytes written: OK
SMDK6410 # |
  
```

Picture-8

By now u-boot-nand.bin is successfully downloaded into Nand Flash, shut off power switch, and set "SW1" into Nand Flash boot mode: the corresponding 1234 value is 1100 (the switch pushing at on is 1). Restart the board then boot up from Nand Flash.

4.2. Set "bootargs" and "bootcmd"

Please see below picture-9



```
DNU v0.60C - For WinCE [COM1, 115200bps] [USB:x] [ADDR: 0xc0008000]
Serial Port USB Port Configuration Help
ethaddr=00:40:5c:26:0a:5b
ipaddr=192.168.0.20
serverip=192.168.0.10
gatewayip=192.168.0.1
netmask=255.255.255.0
stdin=serial
stdout=serial
stderr=serial

Environment size: 332/16380 bytes
SMDK6410 # setenv bootargs "root=/dev/mtdblock2 rootfstype=cramfs console=tty0
console=ttySAC0,115200"
SMDK6410 # setenv bootcmd nand read 0xc0008000 0x100000 0x200000\;bootm 0xc0008000
SMDK6410 # saveenv
Saving Environment to NAND...
Erasing Nand...Writing to Nand... done
SMDK6410 # pri
```

Picture-9

After save the setting, then Reset the board into u-boot

4.3. Burn zImage into Nand Flash 0x100000

. Input the command: dnu c0008000, please see below picture-10

```
Net:      Not Found CS8900@0x18800300
Hit any key to stop autoboot: 0
SMDK6410 # dnu c0008000
DTG cable Connected!
Now, Waiting for DNU to transmit data
```

Picture-10

. Make zImage read the address ram 0xc0008000, please see below picture-11

```
Download Done!! Download Address: 0xc0008000, Download Filesize:0x1baa4c
Checksum is being calculated..
Checksum O.K.
```

Picture-11

. Erase Nand Flash, the start address is 0x100000, the size is 0x200000 bytes, please see below picture-12

```
Checksum O.K.
SMDK6410 # nand erase 100000 200000

NAND erase: device 0 offset 0x100000, size 0x200000
```

Picture-12

Make zImage in the raw be written into Nand Flash, please see below picture-13

```
SMDK6410 # nand write c0008000 100000 200000

NAND write: device 0 offset 0x100000, size 0x200000
2097152 bytes written: OK
```

Picture-13

Zimage is successfully downloaded into Nand Flash, Reset the board and enter to u-boot.

4.4. Burn “rootfs_qt.cramfs” into Nand Flash

. Input the command: dnw c0008000, please see below picture-14

```
Net:      Not Found CS8900@0x18800300
Hit any key to stop autoboot: 0
SMDK6410 # dnw c0008000
OTG cable Connected!
Now, Waiting for DNW to transmit data
```

Picture-14

. Make “rootfs_qt.cramfs” be read into the address “ ram 0xc0008000”, please see below picture-15

```
Now, Waiting for DNW to transmit data
Download Done!! Download Address: 0xc0008000, Download Filesize:0x17b9000
Checksum is being calculated.....
Checksum O.K.
```

Picture-15

. Erase Nand Flash, the start address is 0x300000, the size is 0x3000000 bytes,
Please see below picture-16

```
SMDK6410 # nand erase 300000 3000000  
  
NAND erase: device 0 offset 0x300000, size 0x3000000
```

Picture-16

. Make "rootfs_qt.cramfs" in the ram be written into Nand Flash, please see below picture-17

```
SMDK6410 # nand write c0008000 300000 3000000  
  
NAND write: device 0 offset 0x300000, size 0x3000000  
  
50331648 bytes written: OK
```

Picture-17

By now, u-boot-nand.bin, zImage, rootfs_qt.cramfs have been downloaded into Nand Flash. Please reset to re-enter into the system.

Note: NAND FLASH partition:

- . 0x00000000-0x00100000: "Bootloader"
- . 0x00100000-0x00300000: "Kernel"
- . 0x00300000-0x03300000: "Root-Cramfs"
- . 0x03300000-0x10000000: "File System"

5. Interface Drivers test

5.1. SD Card test

. Please insert SD Card into SD interface, the system will identify the card, and print below information showed picture-18

```
s3c-hsmmc: card inserted.  
mmc0: host does not support reading read-only switch. assuming write-enable.  
mmc0: new high speed SD card at address 0379  
mmcblk0: mmc0:0379 SD01G 1006080KiB  
mmcblk0: p1
```

Picture-18

Mount SD Card to the directory /tmp/sd, the command is : mount -t vfat /dev/mmcblk0p1 /tmp/sd, below picture-19 shows the command.

```
[root@none] /]#  
[root@none] /]# mount -t vfat /dev/mmcblk0p1 /tmp/sd/  
[root@none] /]#
```

Picture-19

Then check the mount whether successful, the command is : Ls /tmp/sd, below picture-20 shows the command.

```
[root@none] /]#  
[root@none] /]# ls /tmp/sd/  
005.rmwb                photo  
20071019218_447032.wmv  rec_test.wav  
22.txt                  rec_test1.wav  
333.txt                 rmwb1.rmwb  
3d_test                 rmwb2.rmwb  
4.bmp                  tcpmp_0.73b  
ChangHong-2.avi        test1.mp3  
Crysis_DX10_trailer_291012007.avi test2.mp3  
_mg_1228.jpg           test3.mp3  
demo1-264-dl.mp4       wavplay  
demo2_divx_dl.avi      wavrec  
ov9650_test  
[root@none] /]#
```

Picture-20

The test of write/read SD Card, below picture-21 shows the test

```
[root@(none) /]#
[root@(none) /]# cp /tmp/sd/test1.mp3 /tmp/
[root@(none) /]# ls /tmp/
Applications      qcop-msg-clock      sd
Documents         qcop-msg-qpe        test1.mp3
Settings          qcop-msg-quicklauncher  udisk
photo             qtembedded-unknown

[root@(none) /]#
[root@(none) /]#
[root@(none) /]# mv /tmp/test1.mp3 /tmp/sd/sdtest.mp3
[root@(none) /]# ls /tmp/sd/
005.rmvb          photo
20071019218_447032.wmv  rec_test.wav
22.txt            rec_test1.wav
333.txt           rmvb1.rmvb
3d_test           rmvb2.rmvb
4.bmp             sdtest.mp3
ChangHong-2.avi   tcpmp_0.73b
Crysis_DX10_trailer_291012007.avi test1.mp3
_gg_1228.jpg      test2.mp3
demo1-264-dl.mp4  test3.mp3
demo2_divx_dl.avi wavplay
ov9650_test       wavrec

[root@(none) /]#
[root@(none) /]#
```

Picture-21

. Demount SD Card, the command is: `umount /tmp/sd`, below picture-22 shows the command. Note: When Demount the SD Card, please make sure that the SD Card is not at the directory of `/tmp/sd/`.

```
[root@(none) /]#
[root@(none) /]# umount /tmp/sd/
[root@(none) /]# s3c-hsmmc: card removed.
mmc0: card 0379 removed
```

Picture-22

5.2. Audio Test

. Copy Audio files like test1.MP3 to SD Card, then mount SD Card to the directory of `/tmp/sd/`.

Then play the Audio file test1.mp3 in the SD Card via MPEGPLAYER. It can stop the

play by pressing "Ctrl+C". Below picture-23 shows the command.

```
[root@none] /]#  
[root@none] /]# mount -t vfat /dev/mmcblk0p1 /tmp/sd/  
[root@none] /]# mpegplayer /tmp/sd/test1.mp3  
Create pluginlibman in libqpe  
Use QPEApplication's PluginLibraryManager  
QuickLauncher invoked as: mpegplayer  
Thread fixed version  
querying: ffmpegplugin  
loading: ffmpegplugin  
querying: wavplugin  
loading: wavplugin  
querying: wavrecord  
2 decoders found  
running as root, can set realtime priority  
checking if file is supported /tmp/sd/test1.mp3  
opening file /tmp/sd/test1.mp3  
opened file /tmp/sd/test1.mp3  
initing file /tmp/sd/test1.mp3  
setting audio stream with id: 0  
finished opening /tmp/sd/test1.mp3
```

Picture-23

5.3. Video test

. Play the Video file in the SD Card via the player MPEGPLAYER. Below picture-24 shows the test.

```
[root@(none) /]# mpegplayer /tmp/sd/Crysis_DX10_trailer_291012007.avi
Create pluginlibman in libqpe
Use QPEApplication's PluginLibraryManager
QuickLauncher invoked as: mpegplayer
Thread fixed version
querying: ffmpegplugin
loading: ffmpegplugin
querying: wavplugin
loading: wavplugin
querying: wavrecord
2 decoders found
running as root, can set realtime priority
checking if file is supported /tmp/sd/Crysis_DX10_trailer_291012007.avi
opening file /tmp/sd/Crysis_DX10_trailer_291012007.avi
opened file /tmp/sd/Crysis_DX10_trailer_291012007.avi
initing file /tmp/sd/Crysis_DX10_trailer_291012007.avi
setting video stream with id: 0
setting audio stream with id: 1
finished opening /tmp/sd/Crysis_DX10_trailer_291012007.avi
Using the LibFFmpegPlugin decoder
set video skin
```

Picture-24

5.4. Ethernet Interface test

. When the system started up, using the Ethernet line provided connects Exchanger with the board's Ethernet interface, then it will print below information showed picture-25.

```
[root@(none) /]# eth0: link up, 100Mbps, full-duplex, lpa 0x45E1
[root@(none) /]#
```

Picture-25

Use the command ifconfig to check IP Setting, if the IP address, Bcast and Mask do not meet your working place's configuration, please reset the IP address, Bcast and Mask. Below picture-26 shows the command.


```
[root@(none) /]#
[root@(none) /]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:E0:4A:BC:15:E7
          inet addr:192.168.2.12  Bcast:192.168.2.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:71 Base address:0x300

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

Picture-26

For example reset IP address into 192.168.1.20, execute the command: `ifconfig eth0 192.168.1.20`. Then execute the command `ifconfig` to check whether the set is ok. Below picture-27 shows the command.

```
[root@(none) /]# ifconfig eth0 192.168.1.20
[root@(none) /]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:E0:4A:BC:15:E7
          inet addr:192.168.1.20  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:6 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:448 (448.0 B)  TX bytes:0 (0.0 B)
          Interrupt:71 Base address:0x300

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

Picture-27

Ping host IP 192.168.1.106, please see below picture-28 shows the command.

```
[root@(none) /]# ping 192.168.1.106
PING 192.168.1.106 (192.168.1.106): 56 data bytes
64 bytes from 192.168.1.106: seq=0 ttl=128 time=5.150 ms
64 bytes from 192.168.1.106: seq=1 ttl=128 time=0.346 ms
64 bytes from 192.168.1.106: seq=2 ttl=128 time=0.354 ms
64 bytes from 192.168.1.106: seq=3 ttl=128 time=0.357 ms
64 bytes from 192.168.1.106: seq=4 ttl=128 time=0.348 ms
64 bytes from 192.168.1.106: seq=5 ttl=128 time=0.354 ms
```

Picture-28

5.5. 7inch TFT LCD test

For Linux OS, the Zimage provided contains 4.3inch TFT LCD driver, if the user want to change into 7inch TFT LCD driver, please reconfigure and recompile the Kernel, below are steps.

- . Enter into linux directory, and execute command: make menuconfig to configure Device Driver →

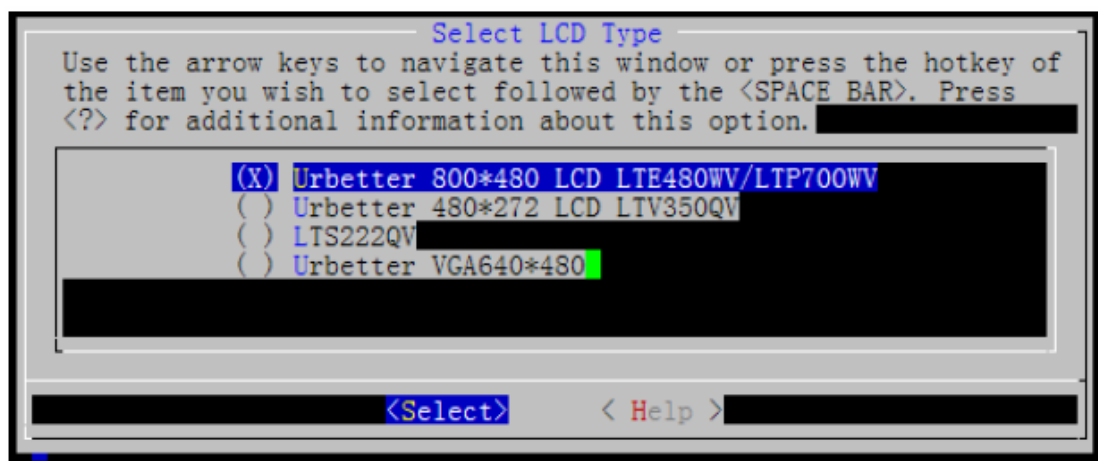
Graphics support ->

<*>Support for frame buffer device →

Select LCD Type (Urbetter 480*272 LCD LTV350QV) →

It will enter into below windows after selected followed above (Press “space key” to select), below picture-29 shows the operation.

Urbetter 800*480 LCD LTE480WV/LTP700WV



Picture-29

- . Exit the windows, and save the configuration.
- . Execute command: Make, the final zImage contains the Kernel supporting 7 inch TFT LCD driver.

5.6. VGA test

- . Enter into linux directory, and execute command: make menuconfig to configure Device Driver →

Graphics support ->

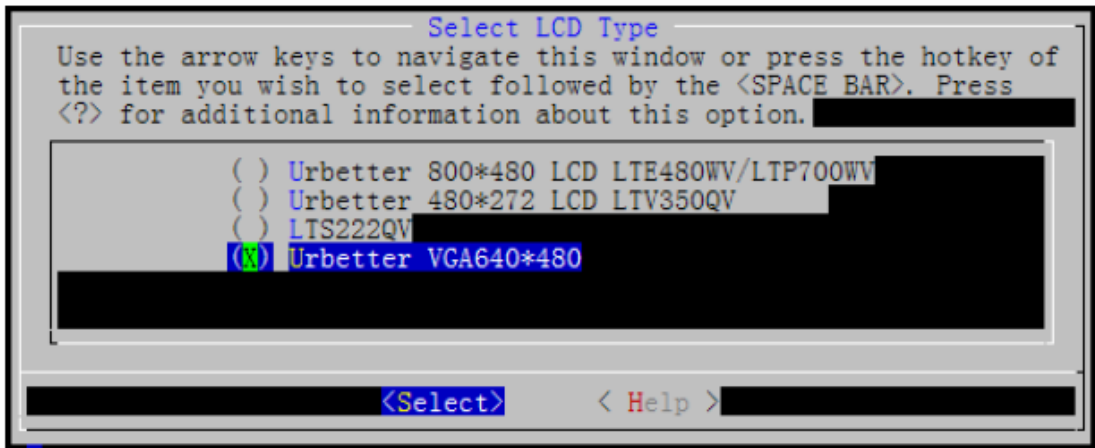
<*>Support for frame buffer device →

Select LCD Type (Urbetter 480*272 LCD LTV350QV) →

It will enter into below windows after selected followed above (Press “space key” to select), below picture-29 shows the operation.

Urbetter VGA640*480

Below picture-30 shows the test

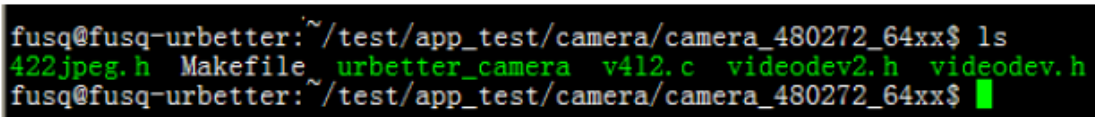


Picture-30

- . Exit the windows, and save the configuration.
- . Execute command: Make, the final zImage contains the Kernel supporting VGA(640*480). At this time, you can connect the VGA with the monitor to see the result.

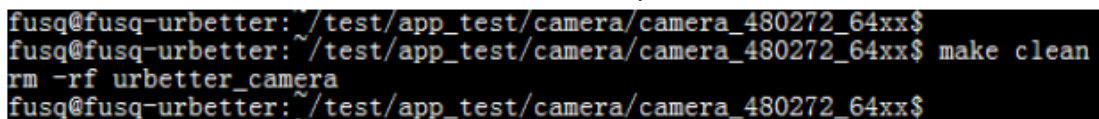
5.7. Camera test

- . Copy the file app_test into the working directory of host Linux PC.
- . Enter into the application directory: app_test/camera/camera_480272_64xx (Note the application is contained in the provided CD-ROM). Below picture-31 shows the command.



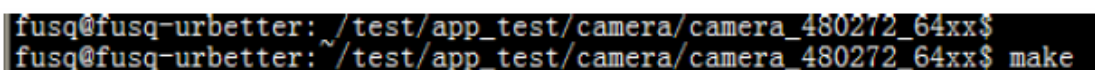
Picture-31

- . Then execute the command: make clean. Below picture-32 shows the command.



Picture-32

- . Then execute the command: make. Below picture-33 shows the command.



Picture-33

- . Then execute the command: ls to make executable file. Below picture-34 shows the command.

```
fusq@fusq-urbetter:~/test/app_test/camera/camera_480272_64xx$ ls
422jpeg.h Makefile urbetter_camera v4l2.c videodev2.h videodev.h
fusq@fusq-urbetter:~/test/app_test/camera/camera_480272_64xx$
fusq@fusq-urbetter:~/test/app_test/camera/camera_480272_64xx$
```

Picture-34

- . Copy the executable file urbetter_camera into SD Card
- After system started up, mount SD Card to /tmp/sd
- . Execute command: `chmod 777 urbetter_camera`, below picture-35 shows the command.

```
[root@(none) /]#
[root@(none) /]# chmod 777 /tmp/sd/urbetter_camera
[root@(none) /]# █
```

Picture-35

- . Then execute the command: `/tmp/sd/urbetter_camera`, below picture-36 shows the command.

```
[root@(none) /]# tmp/sd/urbetter_camera
Start Main pixmt is 16
Codec memory required: 0x00098000 bytes

already open the devicise codec
Before openning FB
start to get pic 1

[root@(none) /]#
```

Picture-35

At this time, you will see the picture that Camera filed taking. You may press the key "Ctrl+C" to stop the work of camera.

Note when use Camera Module:

- a. Make sure that installed 3.3.3 cross-compile(the compile must be installed under the directory of /usr/local/arm)
- b. Please equip with the Camera module before the system startup, as the system has to identify and configure the module.

5.8. Serial interface test

- . Enter into the application directory: `app_test/camera/camera_480272_64xx`, below picture-36 shows the step.

```
fusq@fusq-urbetter:~/test/app_test/uart$
fusq@fusq-urbetter:~/test/app_test/uart$ ls
ext_uart  ext_uart.c  Makefile  sscom32.exe
fusq@fusq-urbetter:~/test/app_test/uart$
```

Picture-36

. The next step, execute command: make clean, below picture-37 shows the step.

```
fusq@fusq-urbetter:~/test/app_test/uart$
fusq@fusq-urbetter:~/test/app_test/uart$ make clean
rm -rf ext_uart
fusq@fusq-urbetter:~/test/app_test/uart$
```

Picture-37

. The next step, execute command: make, below picture-38 shows the step.

```
fusq@fusq-urbetter:~/test/app_test/uart$
fusq@fusq-urbetter:~/test/app_test/uart$ make
/usr/local/arm/3.4.1/bin/arm-linux-gcc ext_uart.c -o ext_uart
fusq@fusq-urbetter:~/test/app_test/uart$
```

Picture-38

. The next step, execute command: ls, then check the executable file, below picture-39 shows the step.

```
fusq@fusq-urbetter:~/test/app_test/uart$
fusq@fusq-urbetter:~/test/app_test/uart$ ls
ext_uart  ext_uart.c  Makefile  sscom32.exe
fusq@fusq-urbetter:~/test/app_test/uart$
```

Picture-39

The next step:

- . Copy the executable file ext_uart to SD Card
- . After system started up, mount SD Card to /tmp/sd
- . Execute command: chmod 777 urbetter_uart, below picture-40 shows the command.

```
[root@none] /]#
[root@none] /]# chmod 777 /tmp/sd/ext_uart
[root@none] /]#
```

Picture-40

The next step, execute command: /tmp/sd/ext_uart, then input "1", below picture-41 shows the step.

```
[root@none] /]#
[root@none] /]# /tmp/sd/ext_uart
0:ttySAC0,1:ttySAC1,2:ttySAC2,3:ttySAC3
please input the open serial:1
```

Picture-41

- . Hardware connection

TtySAC0 is corresponding to COM0, ttySAC1 is corresponding to COM1, using the serial cable provided connects COM1 with host PC. Then open the serial interface debugging tool sscom32.ext(provided in the CD-ROM), below picture-42 shoes the operation.



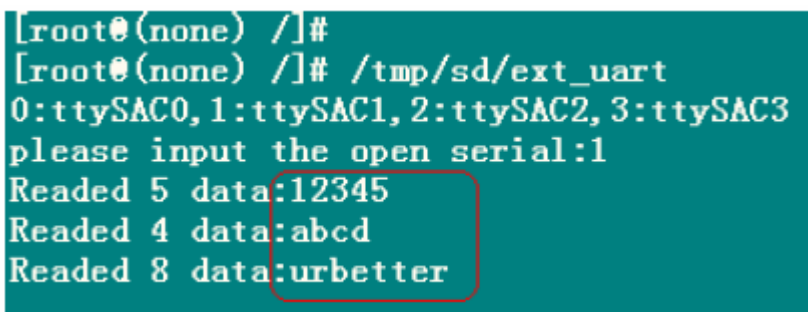
Picture-42

. The next step

Send three group data to test

1. 12345
2. abcd
3. urbetter

. There has output on the ttySC0, please see below picture-43



Picture-43

Above test shows the serial interface working.