

# **Installation Guide for MV2450 (Linux 2.6.24)**



**MicroVision Co.,Ltd.**

## Document Information

Version	1.1
File Name	MV2450 Linux Guide.doc
Date	2009. 9. 29
Satus	Working

## Revision History

Date	Version	Update Descriptions	Editor
2009. 9. 29.	V1.1	First Edition	J.I W

## MV2450-LCD WinCE BSP Guide

Copyright © 2007 MicroVision Co.,Ltd. All rights reserved.

Published by MicroVision Co.,Ltd.

(☎) +82-2-3283-0101, (✉) sale@microvision.co.kr

<http://www.microvision.co.kr>, <http://www.mvtool.co.kr>

Room #610, Hanshin IT Tower 235, Guro3-dong, Guro-gu, Seoul, Korea.

# Contents.....

- 1. OVERVIEW 4/39**
  - 1.1 H/W 5/39**
  - 1.2 S/W 6/39**
  - 1.3 An Arrangement Plan 7/39**
  
- 2. Setting Boot of mode 8/39**
  
- 3. Power switch 10/39**
  
- 4. To Set up Linux 12/39**
  
- 5. Getting Started 19/39**
  - 5.1 Installing Tool chain 19/39**
  - 5.2 Building u-boot 20/39**
  - 5.3 Building Kernel 22/39**
  - 5.3 Building File System 25/39**
  
- 6. (Boot, Kernel, F/S) 26/39**
  - 6.1 Setting for downloading Host PC to Board 27/39**
  
- 7. Doing Camera Application 39/39**

## 1. OVERVIEW

MV2450 is designed to provide hand-held devices and general applications with low-power, and high-performance microcontroller solution in small die size. To reduce total system cost, the MV2450 includes the following components. The MV2450 is developed with ARM926EJ core, 65nm CMOS standard cells and a memory complier. Its lowpower, simple, elegant and fully static design is particularly suitable for cost- and power-sensitive applications. It adopts a new bus architecture known as Advanced Micro controller Bus Architecture (AMBA).



The MV2450 offers outstanding features with its CPU core, a 16/32-bit ARM926EJ RISC processor designed by Advanced RISC Machines, Ltd. The ARM926EJ implements MMU, AMBA BUS, and Harvard cache architecture with separate 16KB instruction and 16KB data caches, each with an 8-word line length.

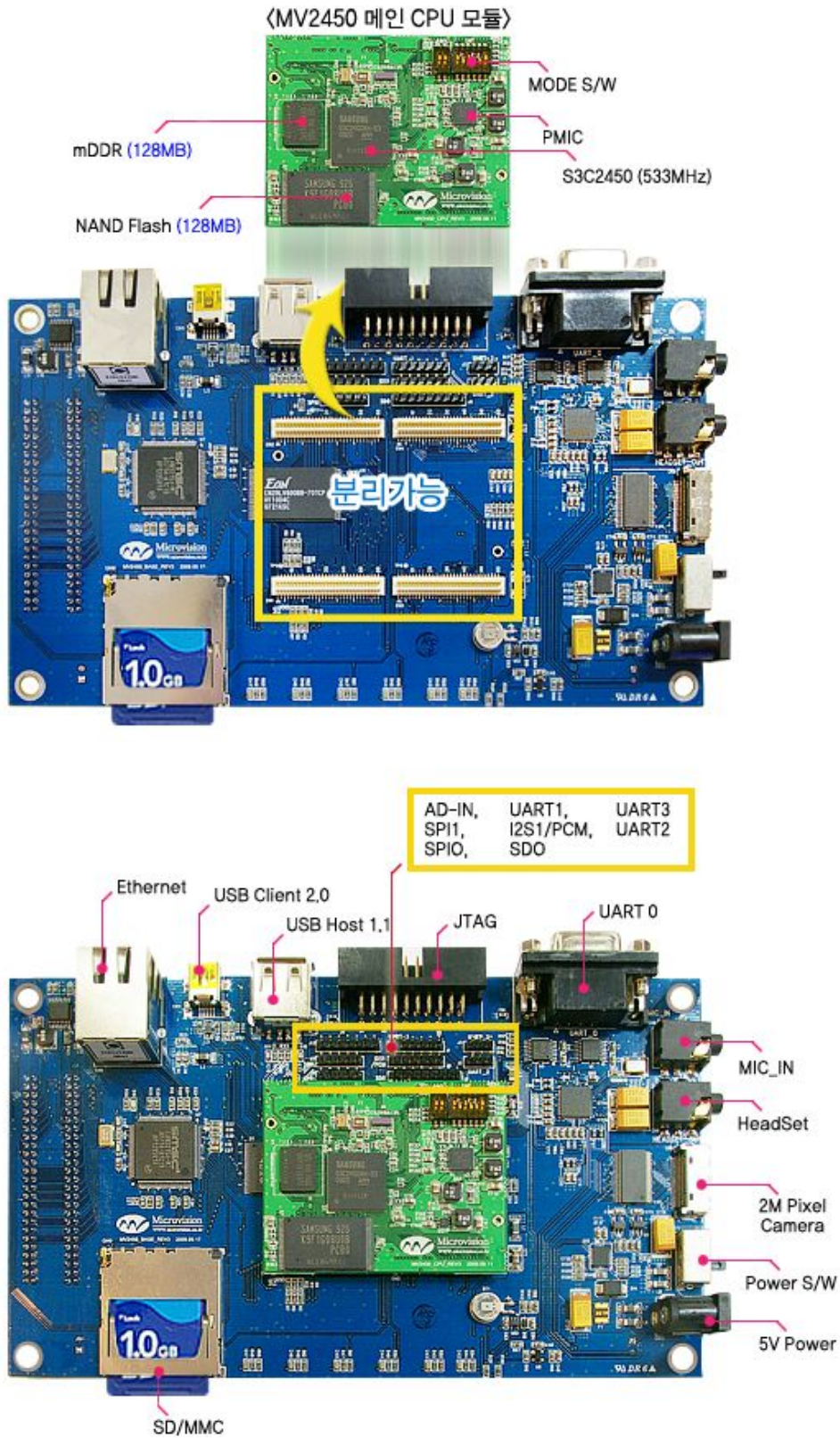
**1.1 H/W**

Items	SPEC & PARTS	Descriptions
CPU	S3C2450X	ARM 926EJ-S CORE 400/533MHz
DDR2(SDRAM)	mDDR	128MB 32bit access, clock speed 266MHz
NAND FLASH	NAND Flash	LARGE BLOCK 1Gb(128MB)
Audio Codec	AC97 I/F	Stereo output capacity 1000mW
Graphic	2D	Graphic Controller
USB	2Ports	USB 1.1 Host Full Speed USB 2.0 Device High Speed
UART	4Ports	3ports, 1Port Debugger
Extension Connector		SPI, GPIO
Camera	Aptina 2M Pixel	Auto Focus, Preview \$ Snap Shot
Ethernet		10/100 Base-T,
SD/MMC	1Port	SD Host Controller v2.0
LCD	4.2" TFT	480X272XRGB Dot, DE, SYSNC Mode, 16.7M, Touch
Battery	1530mA @3.7v	Full Operation 1 Hour
Charger	Micro Semi LX2205	AC & Battery Switching, Battery Protection AC Present, CC-CV

**1.2 S/W**

<b>OS</b>	Windows CE 6.0 Embedded Linux 2.6.24
<b>Bootloader</b>	H/W UART 115200
<b>Image Download</b>	USB 2.0 Client / 10/100M Ethernet
<b>Flash</b>	NAND Flash File System
<b>Driver</b>	TFT LCD 4.3" USB1.1HOST USB Client 2.0 Active Sync (Version 4.x) <ul style="list-style-type: none"><li>- Debugger UART</li><li>- Serial communication</li></ul> MMC/SD Card CMOS Camera Ethernet Device Driver
<b>Application</b>	Camera APP

### 1.3 An Arrangement Plan



## 2. Setting Boot of mode

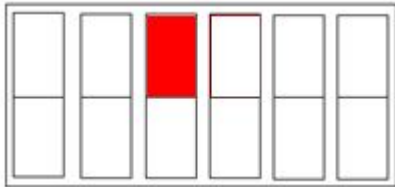
MV2450 has a NOR Flash that has programmed USB of firmware which is Boot of firmware in order to program to NAND Flash using of USB Client 2.0 so that you can program Eboot or u-boot to the NAND Flash using of USB firmware on the NOR Flash.

When you are deleted bootloader on the NAND flash but you can program without JTAG emulator. Because of NOR Flash

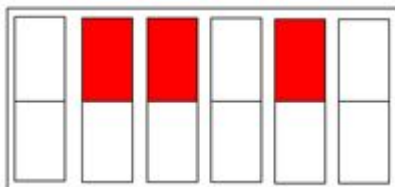


## Setting Boot of mode

NAND Flash BOOT



NOR Flash BOOT



### 3. Power switch

5V Power.



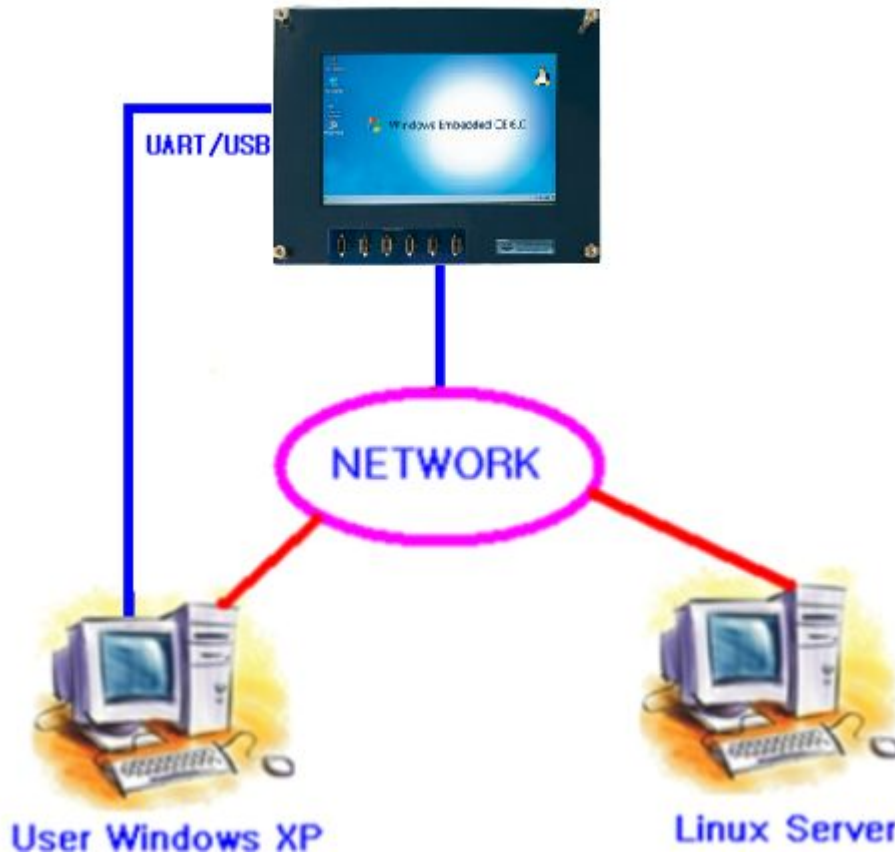
<Connected 5V power>

This is a reset button



<a reset button >

## 4. To Set up Linux



You have to connect the Ethernet with Linux Server, Windows XP, MV2450 and UART to see messages.

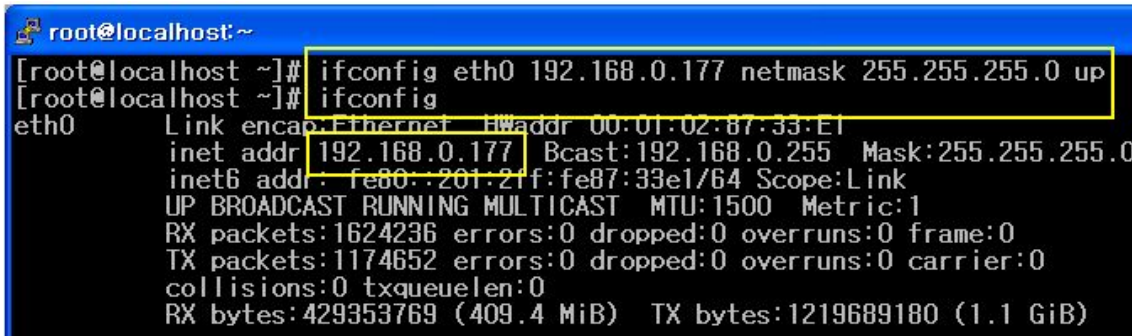
Lists

- ▶ To set up IP
- ▶ To set up minicom
- ▶ To set up tftp Server

### 1) To set up IP

You can setup IP, Please follow the commands

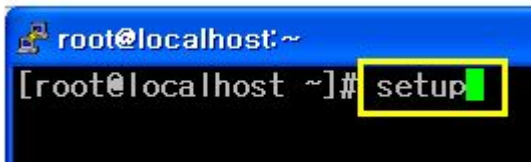
**# ifconfig eth0 192.168.0.177 netmask 255.255.255.0 up**



```
root@localhost:~  
[root@localhost ~]# ifconfig eth0 192.168.0.177 netmask 255.255.255.0 up  
[root@localhost ~]# ifconfig  
eth0      Link encap:Ethernet  HWaddr 00:01:02:87:33:E1  
          inet addr:192.168.0.177  Bcast:192.168.0.255  Mask:255.255.255.0  
          inet6 addr: fe80::201:21f:fe87:33e1/64  Scope:Link  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
          RX packets:1624236  errors:0  dropped:0  overruns:0  frame:0  
          TX packets:1174652  errors:0  dropped:0  overruns:0  carrier:0  
          collisions:0 txqueuelen:0  
          RX bytes:429353769 (409.4 MiB)  TX bytes:1219689180 (1.1 GiB)
```

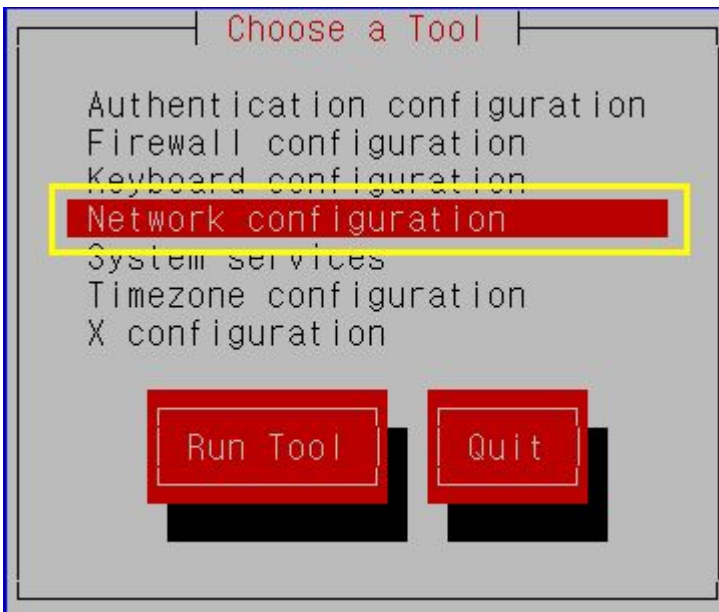
And also can do using of “**setup**”.

**# setup**

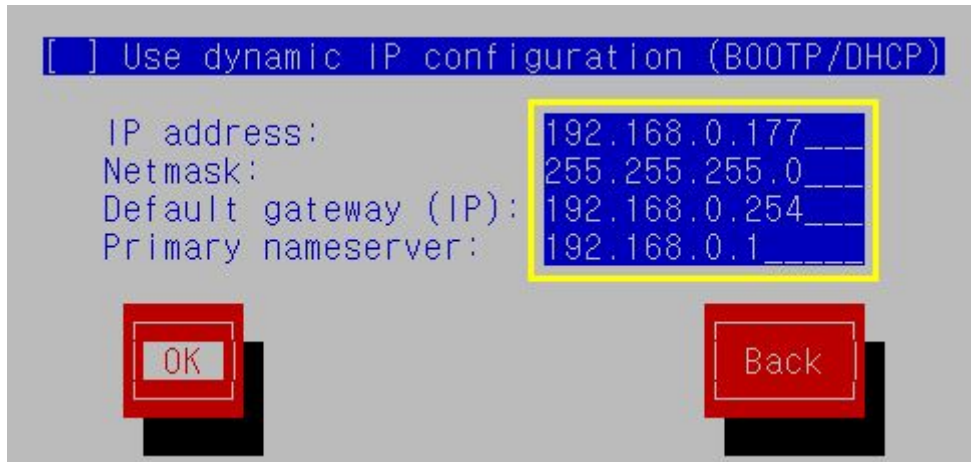


```
root@localhost:~  
[root@localhost ~]# setup
```

Select “**Network configuration**”



Select **“OK”**

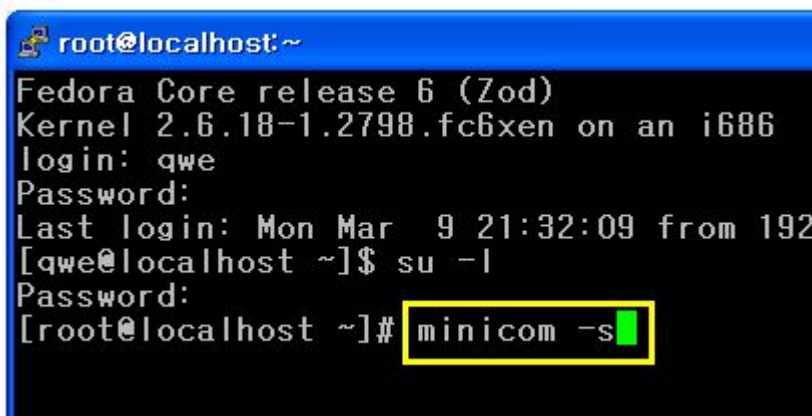


## 2) To set up minicom

The minicom is program to communication with MV2450 in Linux.

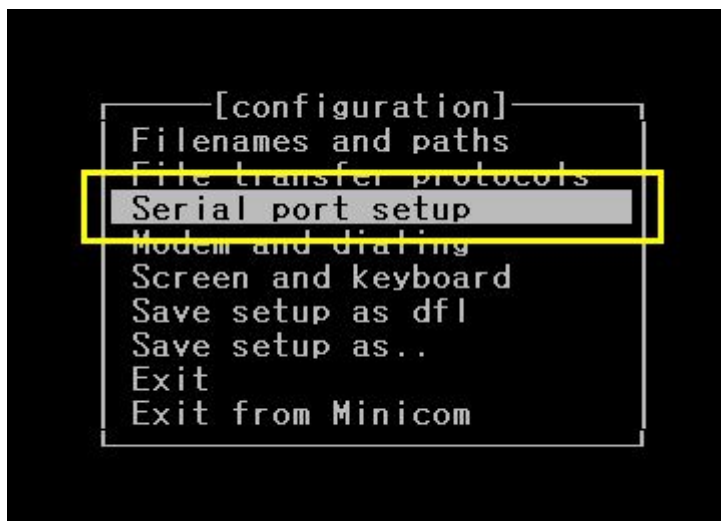
Please follow the commands. **“minicom -s”**

**# minicom -s**



```
root@localhost:~  
Fedora Core release 6 (Zod)  
Kernel 2.6.18-1.2798.fc6xen on an i686  
login: qwe  
Password:  
Last login: Mon Mar  9 21:32:09 from 192  
[qwe@localhost ~]$ su -l  
Password:  
[root@localhost ~]# minicom -s
```

Select **“Serial port setup”**



```
[configuration]  
Filenames and paths  
File transfer protocols  
Serial port setup  
Modem and dialing  
Screen and keyboard  
Save setup as df1  
Save setup as..  
Exit  
Exit from Minicom
```

If you want to setup "Serial Device," Select "a" and then press button "Enter" on the keyboard.

```
A - Serial Device      : /dev/ttyS0
B - Lockfile Location  : /var/lock
C - Callin Program    :
D - Callout Program   :
E - Bps/Par/Bits      : 115200 8N1
F - Hardware Flow Control : No
G - Software Flow Control : No

Change which setting? █

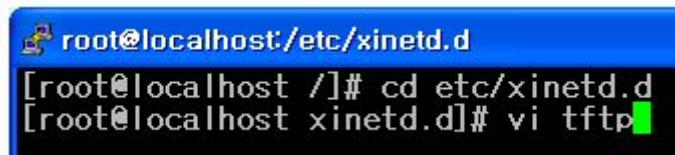
Screen and keyboard
Save setup as dfl
Save setup as..
Exit
Exit from Minicom
```

### 3) To set up tftp Server

TFTP put up so that can download u-boot, zImage, File System to the MV2450.

```
# cd etc/xinetd.d
```

```
# vi tftp
```



```
root@localhost:/etc/xinetd.d
[root@localhost /]# cd etc/xinetd.d
[root@localhost xinetd.d]# vi tftp
```

#### Before

```
service tftp
{
    disable = yes
    socket_type      = dgram
    protocol         = udp
    wait            = yes
    user            = root
    server          = /usr/sbin/in.tftpd
    server_args     = -s /tftpboot
    per_source      = 11
    cps             = 100 2
    flags           = IPv4
}
```

#### After

```
service tftp
{
    disable = no
    socket_type      = dgram
    protocol         = udp
    wait            = yes
    user            = root
    server          = /usr/sbin/in.tftpd
    server_args     = -s /tftpboot
    per_source      = 11
    cps             = 100 2
    flags           = IPv4
}
```

You have to modify “**disable = no**” so that can work.

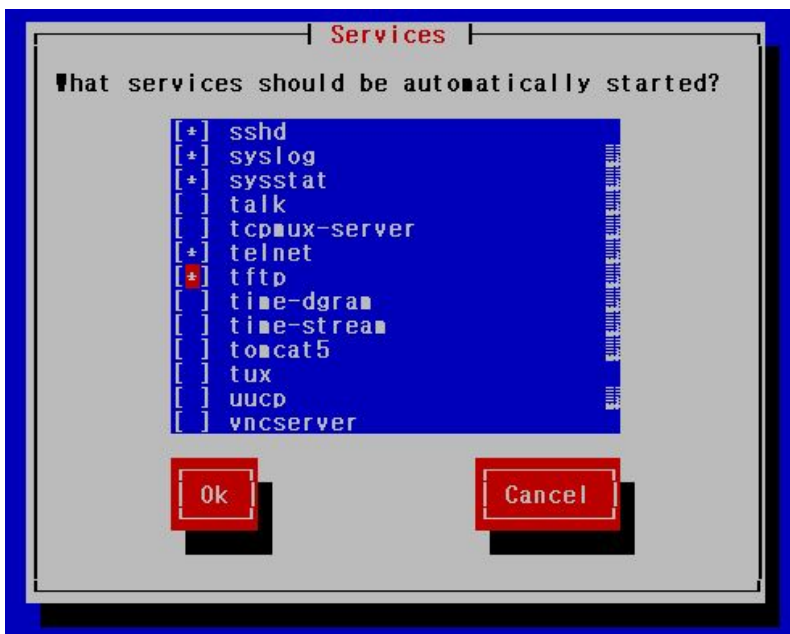
You should add “tftp” through “**setup**”

**# setup**

Select “**System services**”.



Select “**tftp**”



Run “**xinetd**” After end up “**OK**”.

**# service xinetd restart**

## 5. Getting Started

### 5.1 Installing Tool chain

Building the tool chain is not a trivial exercise and for most common situations pre-built tool chains already exists. Unless you need to build your own, or you want to do it anyway to gain a deeper understanding, then simply installing and using a suitable ready-made tool chain is strongly recommended.

Copy “cross-4.2.2-eabi.tar.bz2” in CD /usr/local/arm to Linux PC /usr/local/arm.

Please follow the commands

```
# mkdir -p /usr/local/arm
# tar xvf cross-4.2.2-eabi.tar.bz2
# mv 4.2.2-eabi /usr/local/arm/
# export PATH=$PATH:/usr/local/arm/4.2.2-eabi/usr/bin/arm-linux-
```

## 5.2 Building u-boot

### ► Unzip

u-boot-1.3.4-s3c2450.tar.gz

Please follow the commands

```
[root@localhost mv2450]# tar xvf u-boot-1.3.4-s3c2450.tar.gz
[root@localhost mv2450]# cd u-boot-1.3.4-samsung
[root@localhost u-boot-1.3.4-samsung]#
```

### ► To set up the GCC PATH to compile

Please follow the commands

[root@localhost u-boot-1.3.4-samsung]# vi Makefile

```
endif
ifeq ($(ARCH), arm)
CROSS_COMPILE = /usr/local/arm/4.2.2-eabi/usr/bin/arm-linux-
endif
ifeq ($(ARCH), i386)
CROSS_COMPILE = i386-linux-
```

CROSS\_COMPILE = /usr/local/arm/4.2.2-eabi/usr/bin/arm-linux-

### ► To set up IP u-boot IP

Please follow the commands

[root@localhost u-boot-1.3.4-samsung]# vi include/configs/smdk2450.h

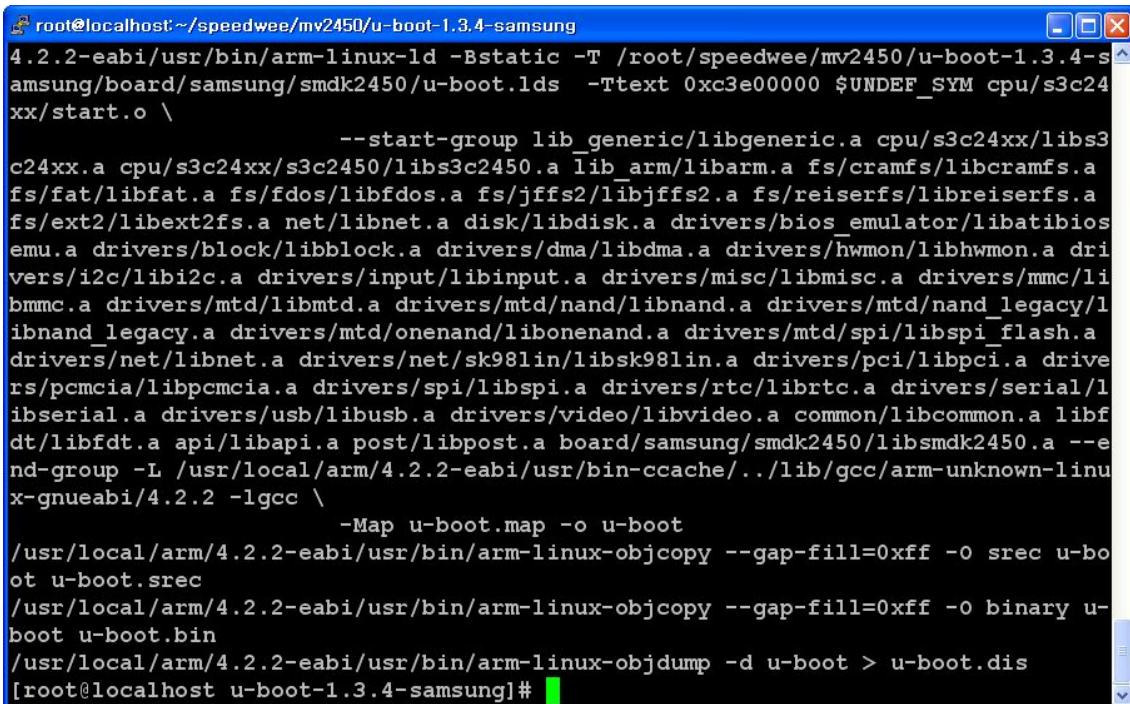
```
#define CONFIG_ETHADDR      00:40:5c:26:0a:5b
#define CONFIG_NETMASK      255.255.255.0
#define CONFIG_IPADDR       192.168.0.230
#define CONFIG_SERVERIP     192.168.0.228
#define CONFIG_GATEWAYIP    192.168.0.1
```

```
#define CONFIG_ETHADDR      MAC ADDRESS
#define CONFIG_NETMASK      SUBNETMASK
#define CONFIG_IPADDR       Target Board PC
#define CONFIG_SERVERIP     HOST PC
```

► **To compile u-boot**

Please follow the commands

```
[root@localhost u-boot-1.3.4-samsung]# make clobber
[root@localhost u-boot-1.3.4-samsung]# make
smdk2450_config
Configuring for smdk2450 board...
[root@localhost u-boot-1.3.4-samsung]# make
```



```
root@localhost:~/speedwee/mv2450/u-boot-1.3.4-samsung
4.2.2-eabi/usr/bin/arm-linux-ld -Bstatic -T /root/speedwee/mv2450/u-boot-1.3.4-samsung/board/samsung/smdk2450/u-boot.lds -Ttext 0xc3e00000 $UNDEF_SYM cpu/s3c24xx/start.o \
--start-group lib_generic/libgeneric.a cpu/s3c24xx/libs3c24xx.a cpu/s3c24xx/s3c2450/libs3c2450.a lib_arm/libarm.a fs/cramfs/libcramfs.a fs/fat/libfat.a fs/fdos/libfdos.a fs/jffs2/libjffs2.a fs/reiserfs/libreiserfs.a fs/ext2/libext2fs.a net/libnet.a disk/libdisk.a drivers/bios_emulator/libatibios_emu.a drivers/block/libblock.a drivers/dma/libdma.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/nand/libnand.a drivers/mtd/nand_legacy/libnand_legacy.a drivers/mtd/onenand/libonenand.a drivers/mtd/spi/libspi_flash.a drivers/net/libnet.a drivers/net/sk98lin/libsk98lin.a drivers/pci/libpci.a drivers/pcmcia/libpcmcia.a drivers/spi/libspi.a drivers/rtc/librtc.a drivers/serial/libserial.a drivers/usb/libusb.a drivers/video/libvideo.a common/libcommon.a libfdt/libfdt.a api/libapi.a post/libpost.a board/samsung/smdk2450/libsmdk2450.a --end-group -L /usr/local/arm/4.2.2-eabi/usr/bin-ccache/./lib/gcc/arm-unknown-linux-gnueabi/4.2.2 -lgcc \
-Map u-boot.map -o u-boot
/usr/local/arm/4.2.2-eabi/usr/bin/arm-linux-objcopy --gap-fill=0xff -O srec u-boot u-boot.srec
/usr/local/arm/4.2.2-eabi/usr/bin/arm-linux-objcopy --gap-fill=0xff -O binary u-boot u-boot.bin
/usr/local/arm/4.2.2-eabi/usr/bin/arm-linux-objdump -d u-boot > u-boot.dis
[root@localhost u-boot-1.3.4-samsung]#
```

After compile, u-boot is in the “**u-boot-1.3.4-samsung**”

After compile, Copy u-boot.bin to. “/tftpboot”

Please follow the commands

```
[root@localhost u-boot-1.3.4-samsung]# cp u-boot /tftpboot
```

## 5.3 Building Kernel

### ► Unzip

s3c2450-linux-2.6.24.tar.gz

Please follow the commands

```
[root@localhost mv2450]# tar xvf s3c2450-linux-2.6.24.tar.gz
```

### ► To set up the GCC PATH to compile

Please follow the commands

```
[root@localhost mv2450]# cd s3c-linux-2.6.24  
[root@localhost s3c-linux-2.6.24]#  
vi .cross_compile
```

```
/usr/local/arm/4.2.2-eabi/usr/bin/arm-linux-
```

```
/usr/local/arm/4.2.2-eabi/usr/bin/arm-linux-
```

- ▶ To compile

Please follow the commands

**[root@localhost s3c-linux-2.6.24]# make mv2450\_defconfig**

```

root@localhost:~/speedwee/mv2450/s3c-linux-2.6.24
S3C UART to use for low-level debug (DEBUG_S3C_UART) [0] 0
*
* Security options
*
Enable access key retention support (KEYS) [N/y/?] n
Enable different security models (SECURITY) [N/y/?] n
File POSIX Capabilities (EXPERIMENTAL) (SECURITY_FILE_CAPABILITIES) [N/y/?] n
*
* Cryptographic API
*
Cryptographic API (CRYPTO) [N/y/?] n
*
* Library routines
*
CRC-CCITT functions (CRC_CCITT) [N/m/y/?] n
CRC16 functions (CRC16) [N/m/y/?] n
CRC ITU-T V.41 functions (CRC_ITU_T) [N/m/y/?] n
CRC32 functions (CRC32) [Y/?] y
CRC7 functions (CRC7) [N/m/y/?] n
CRC32c (Castagnoli, et al) Cyclic Redundancy-Check (LIBCRC32C) [N/m/y/?] n
#
# configuration written to .config
#
[root@localhost s3c-linux-2.6.24]#

```

Please follow the commands

**[root@localhost s3c-linux-2.6.24]# make**

```

root@localhost:~/speedwee/mv2450/s3c-linux-2.6.24
#
[root@localhost s3c-linux-2.6.24]# make
scripts/kconfig/conf -s arch/arm/Kconfig
sound/soc/s3c64xx/Kconfig:40:warning: choice values currently only support a single prompt
sound/soc/s3c64xx/Kconfig:46:warning: choice values currently only support a single prompt
sound/soc/s3c24xx/Kconfig:47:warning: choice values currently only support a single prompt
sound/soc/s3c24xx/Kconfig:53:warning: choice values currently only support a single prompt
drivers/usb/core/Kconfig:118:warning: defaults for choice values not supported
CHK include/linux/version.h
SYMLINK include/asm-arm/arch -> include/asm-arm/arch-s3c2410
make[1]: `include/asm-arm/mach-types.h' is up to date.
CHK include/linux/utsrelease.h
CALL scripts/checksyscalls.sh
<stdin>:1097:2: warning: #warning syscall fadvise64 not implemented
<stdin>:1265:2: warning: #warning syscall migrate_pages not implemented
<stdin>:1321:2: warning: #warning syscall pselect6 not implemented
<stdin>:1325:2: warning: #warning syscall ppoll not implemented
<stdin>:1365:2: warning: #warning syscall epoll_pwait not implemented
CHK include/linux/compile.h

```

After compile so that made “zimage” in arch/arm/boot/.  
Copy “zimage” to “/tftpboot”.

```
[root@localhost boot]# cp zimage /tftpboot
```

## 5.4 Building File System

rootfs\_mv2450.tar.gz

### ► To make a Cramfs

Please follow the commands

```
[root@localhost mv2450]# tar xvf rootfs_mv2450.tar.gz
```

```
[root@localhost mv2450]# mkfs.cramfs rootfs_mv2450 fs.mv2450.cramfs
```

```
[root@localhost mv2450]# ll
total 250200
drwxr-xr-x 18 root root      4096 Jul 30 16:58 rootfs_mv2450
-rwxr-w-r-- 1 root root 13068220 Jul 31 14:48 rootfs_mv2450.tar.gz
drwxrwxr-x 21 1000 1000     4096 Aug 19 20:56 s3c-linux-2.6.24
-rwxr-w-r-- 1 root root 229045378 Jul 31 14:48 s3c2450-linux-2.6.24.tar.gz
-rwxr-w-r-- 1 root root 13800067 Jul 31 14:47 u-boot-1.3.4-s3c2450.tar.gz
drwxrwxr-x 32 jong jong     4096 Aug 19 21:06 u-boot-1.3.4-samsung
[root@localhost mv2450]# mkfs.cramfs rootfs_mv2450 fs.mv2450.cramfs
[root@localhost mv2450]# ll
total 263860
-rw-r--r-- 1 root root 13967360 Aug 19 21:06 fs.mv2450.cramfs
drwxr-xr-x 18 root root      4096 Jul 30 16:58 rootfs_mv2450
-rwxr-w-r-- 1 root root 13068220 Jul 31 14:48 rootfs_mv2450.tar.gz
drwxrwxr-x 21 1000 1000     4096 Aug 19 20:56 s3c-linux-2.6.24
-rwxr-w-r-- 1 root root 229045378 Jul 31 14:48 s3c2450-linux-2.6.24.tar.gz
-rwxr-w-r-- 1 root root 13800067 Jul 31 14:47 u-boot-1.3.4-s3c2450.tar.gz
drwxrwxr-x 32 jong jong     4096 Aug 19 21:06 u-boot-1.3.4-samsung
[root@localhost mv2450]#
```

This is mkfs.cramfs which in fedora 6 of the commanding.

Copy File System to “/tftpboot”

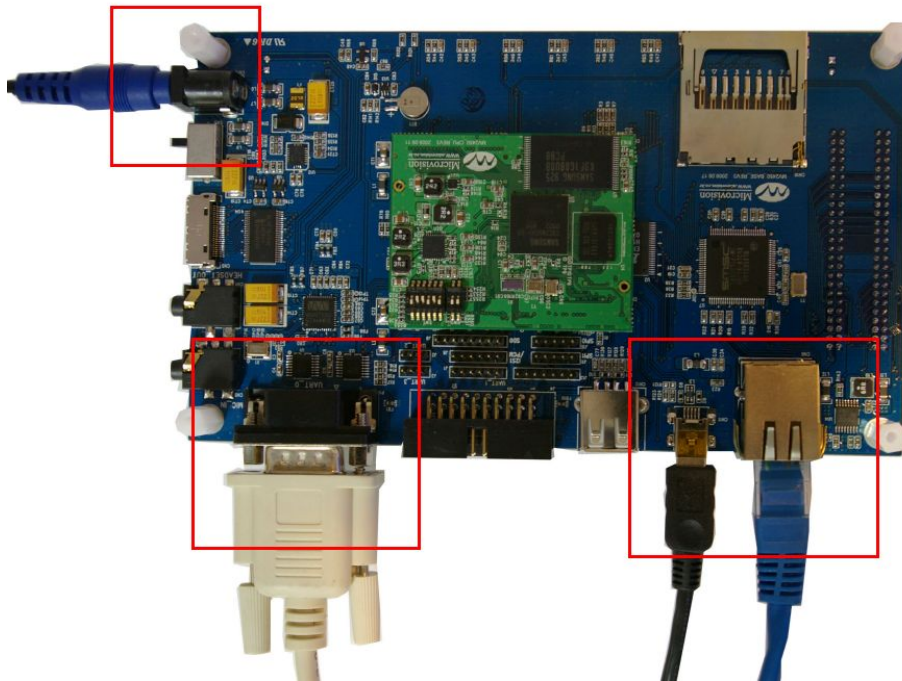
Please follow the commands

```
[root@localhost mv2450]# cp fs.mv2450.cramfs /tftpboot/
```

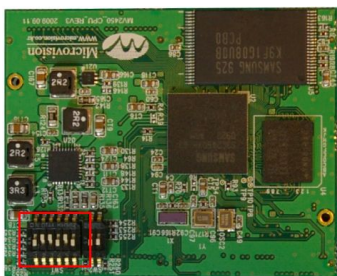
### 6. (Boot, Kernel, F/S)

In this chapter, you can understand how to download u-boot.bin zImage and file system Please the following window appears on your screen.

First, you have to set up environment such as Board with Host PC and then Connect USB 2.0 Device with your Host PC to download through USB and also UART for monitoring.

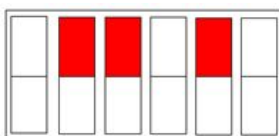


Set up NOR Flash of mode. As I first introduced in Nor Flash number 2. NOR Flash has programmed a “2450MON\_NOR.bin “that can download. Set up a switch in order to boot and then run “DNW” in TOOL\DNW.



**To set up switch on Main Board.**

NOR Flash Mode

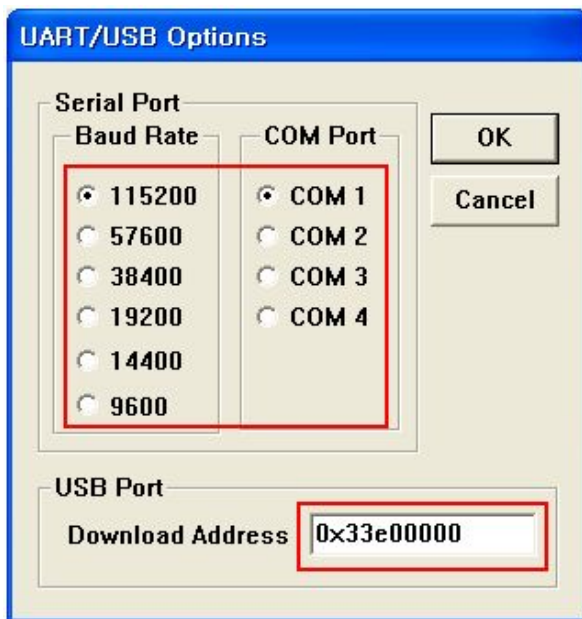
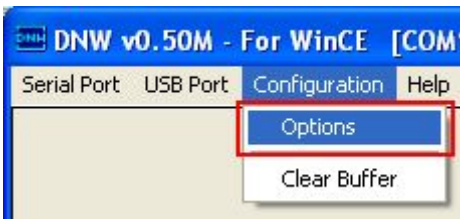


## 6.1 Setting for downloading Host PC to Board

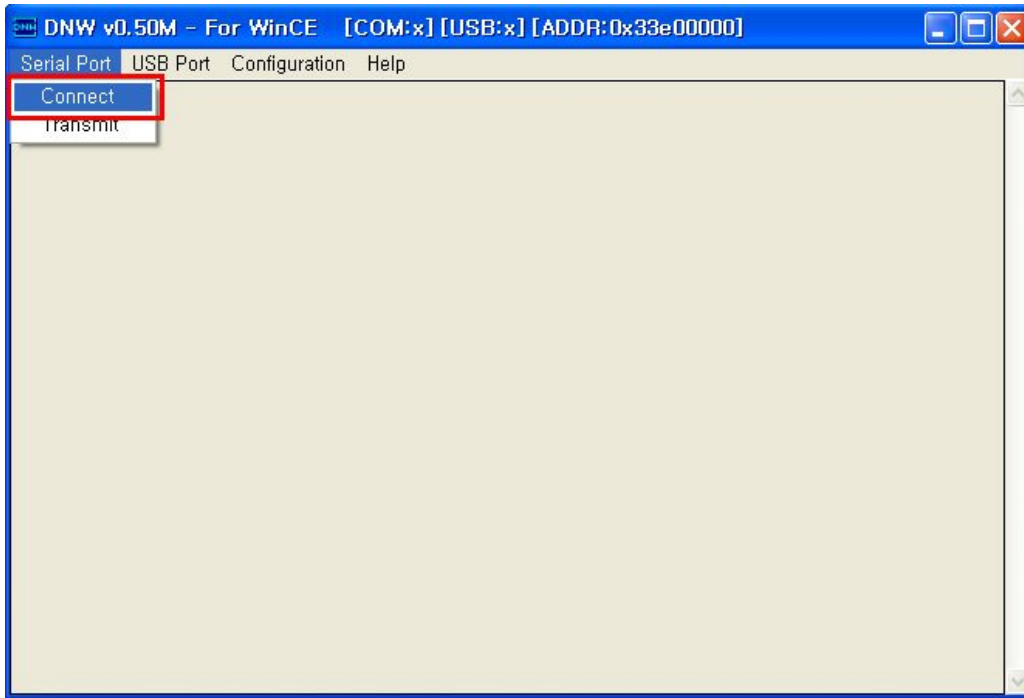
Run DNW.exe in \Tool



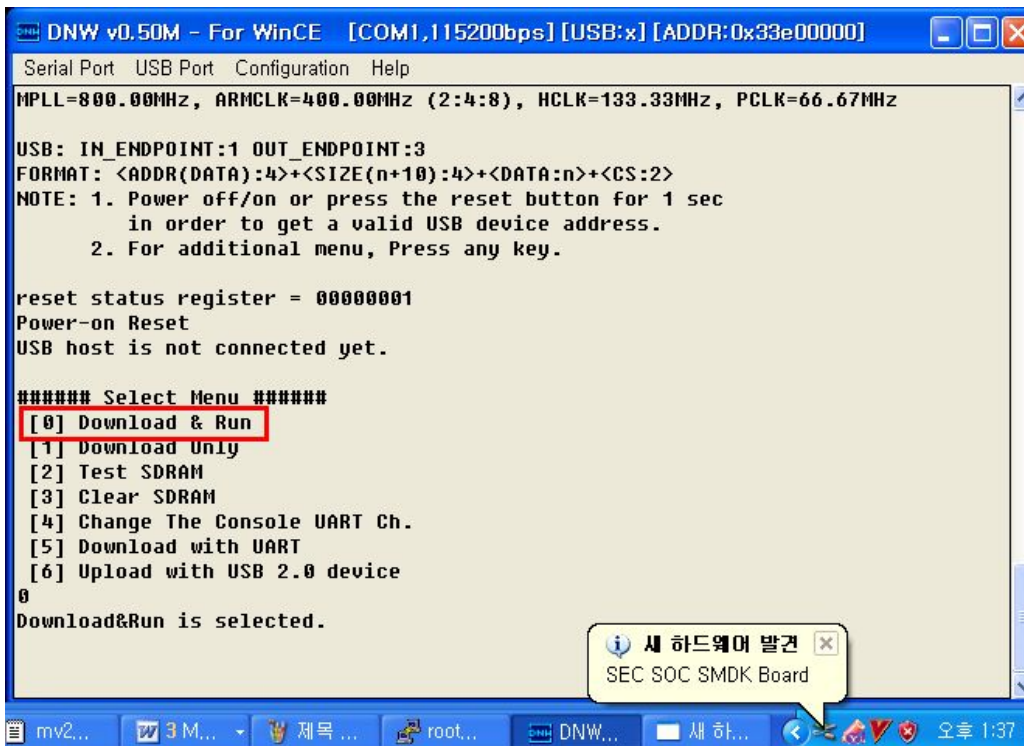
On the Configuration menu, click Options to set the UART/USB options. The following window appears on your screen. Select Baud Rate and COM Port as shown in figure "UART/USB options", enter the download address as 0x33e00000 and then click OK button.



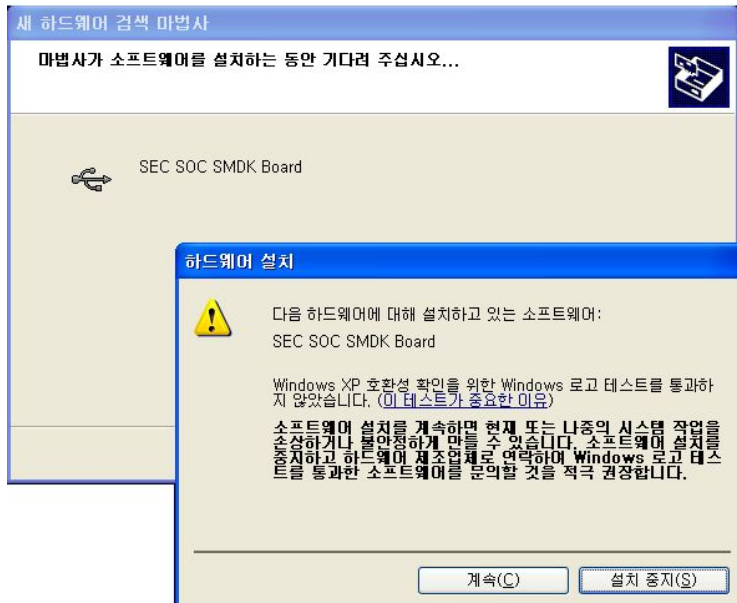
On the Serial Port menu, click Connect. Switch ON the reference board and then press.



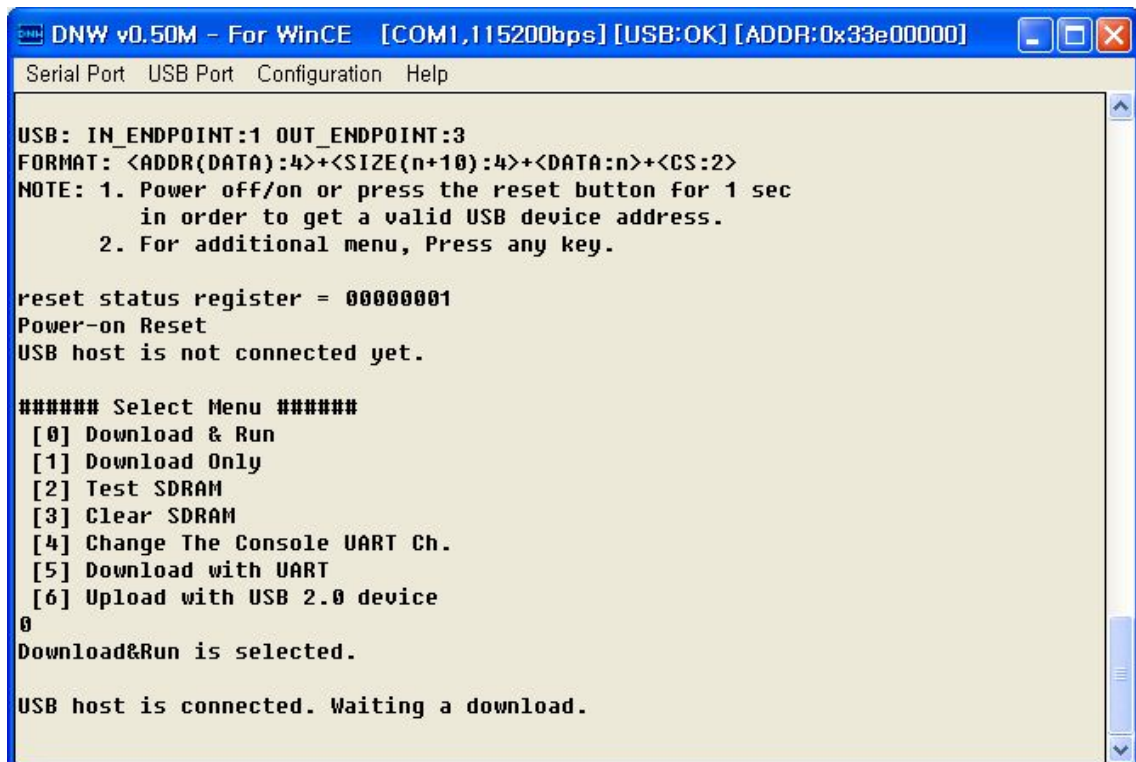
Enter "0"



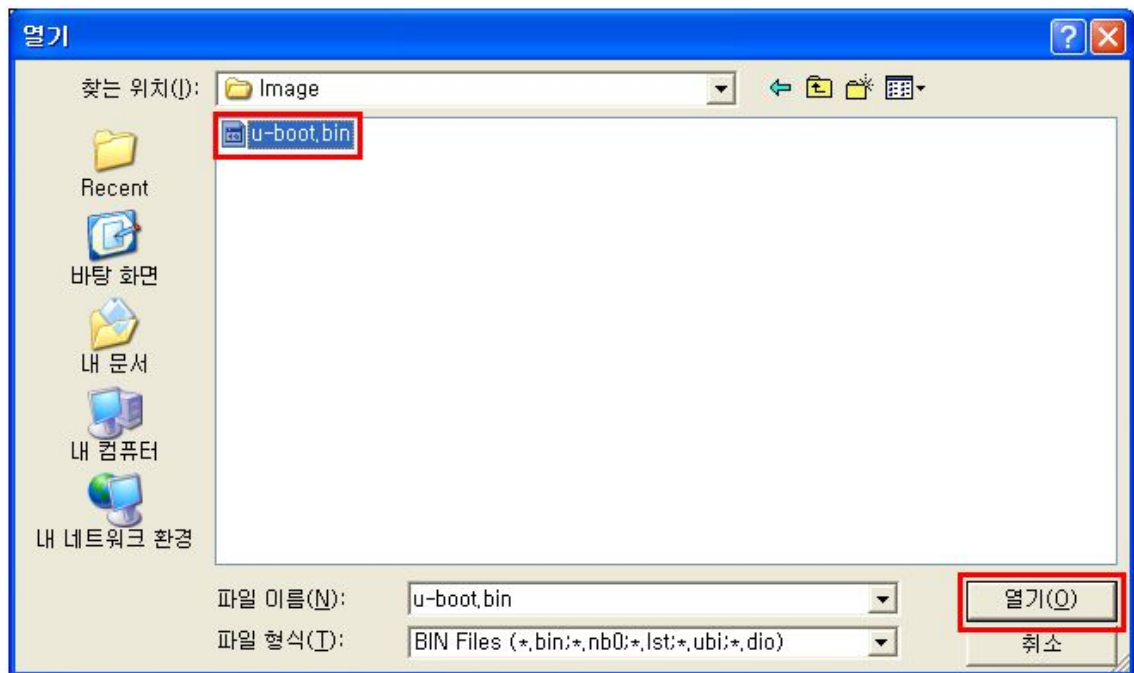
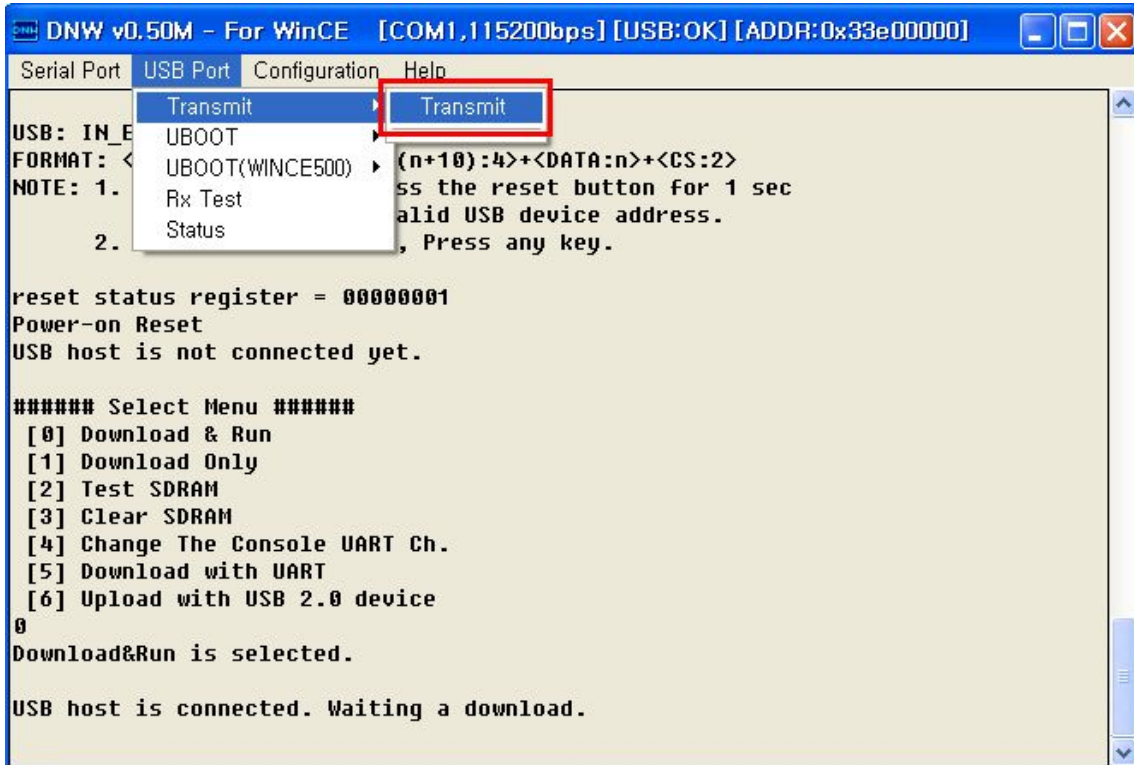
USB device driver is in CD TOOL\DNW.



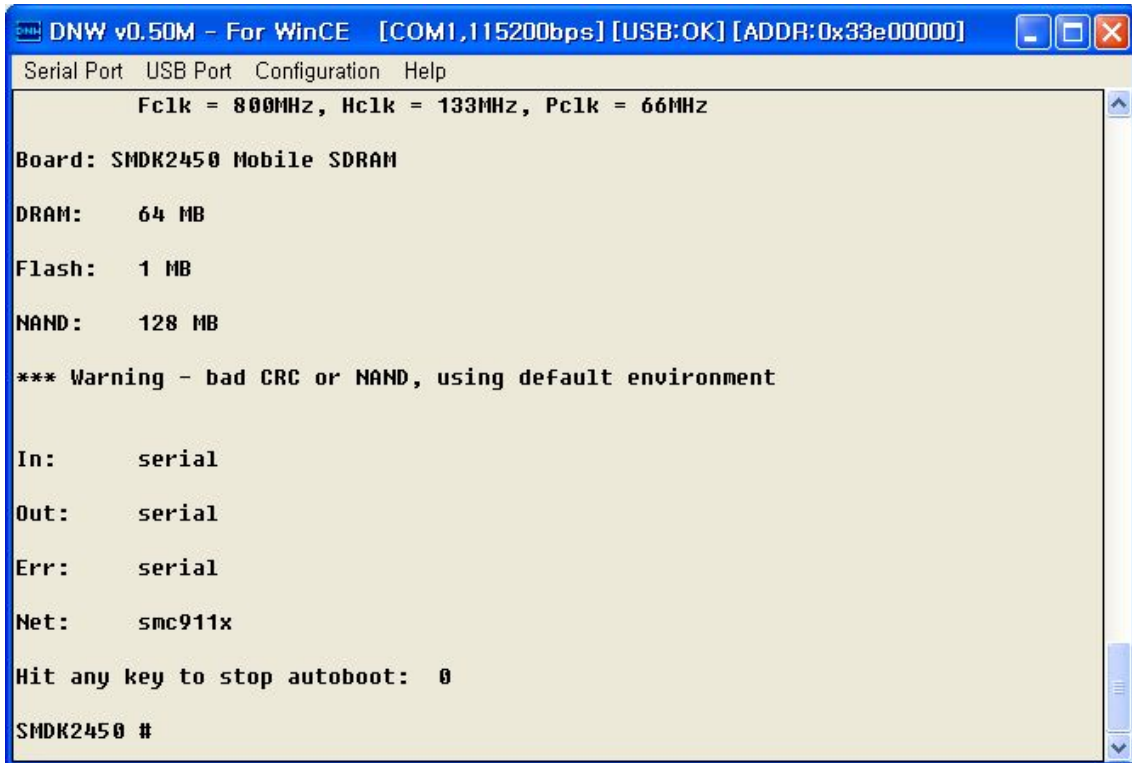
It is done



Enter "0", On the USB Port menu, click Transmit and the following window appears on your screen. Select u-boot.bin file from X\SRC\Linux\bin directory and then click Open button.



As soon as u-boot.bin download is over, the following messages appear in the DNW window. Please hit the SPACE BAR key to view the current Ethernet Boot Loader Configuration. Configure the Ethernet Boot loader as follows by entering the respective options..



```
DNW v0.50M - For WinCE [COM1,115200bps] [USB:OK] [ADDR:0x33e00000]
Serial Port  USB Port  Configuration  Help
      Fc1k = 800MHz, Hc1k = 133MHz, Pc1k = 66MHz

Board: SMDK2450 Mobile SDRAM

DRAM:      64 MB
Flash:     1 MB
NAND:     128 MB

*** Warning - bad CRC or NAND, using default environment

In:      serial
Out:     serial
Err:     serial
Net:     smc911x

Hit any key to stop autoboot:  0

SMDK2450 #
```

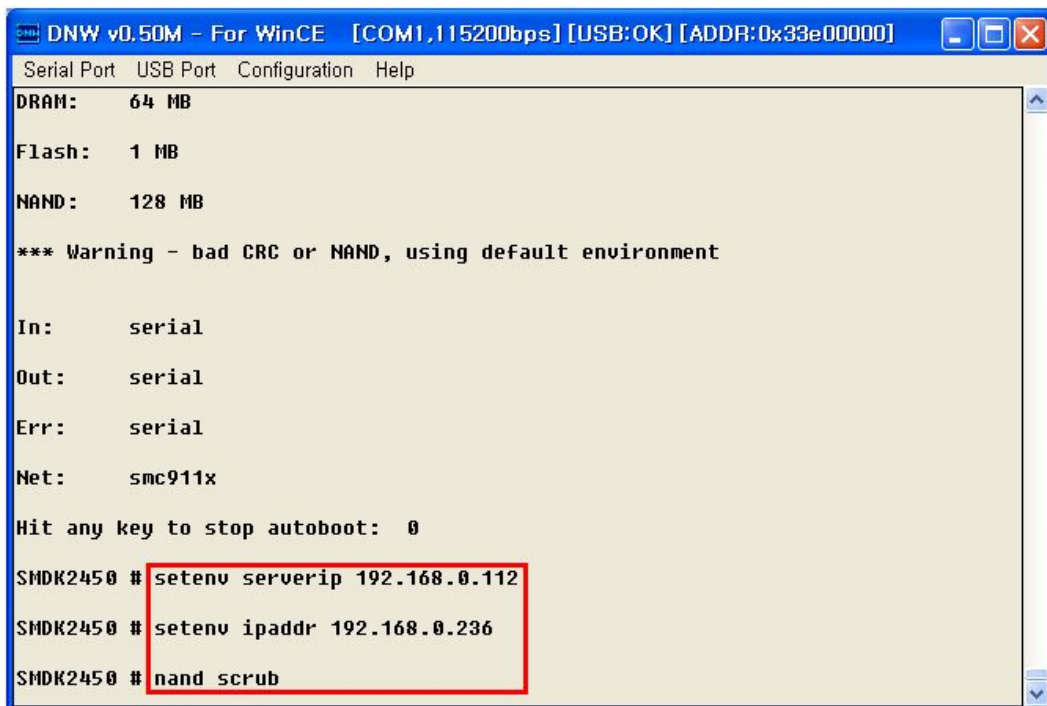
## Setup IP

Linux Server PC for taking in tftpboot.

TFTP Server IP : **setenv serverip 192.168.0.122**

Device PC IP : **setenv ipaddr 192.168.0.236**

For erasing Bad blocks in NAND : **nand scrub**



The screenshot shows a terminal window titled "DNW v0.50M - For WinCE [COM1,115200bps] [USB:OK] [ADDR:0x33e00000]". The window contains the following text:

```
Serial Port  USB Port  Configuration  Help
DRAM:      64 MB
Flash:     1 MB
NAND:     128 MB

*** Warning - bad CRC or NAND, using default environment

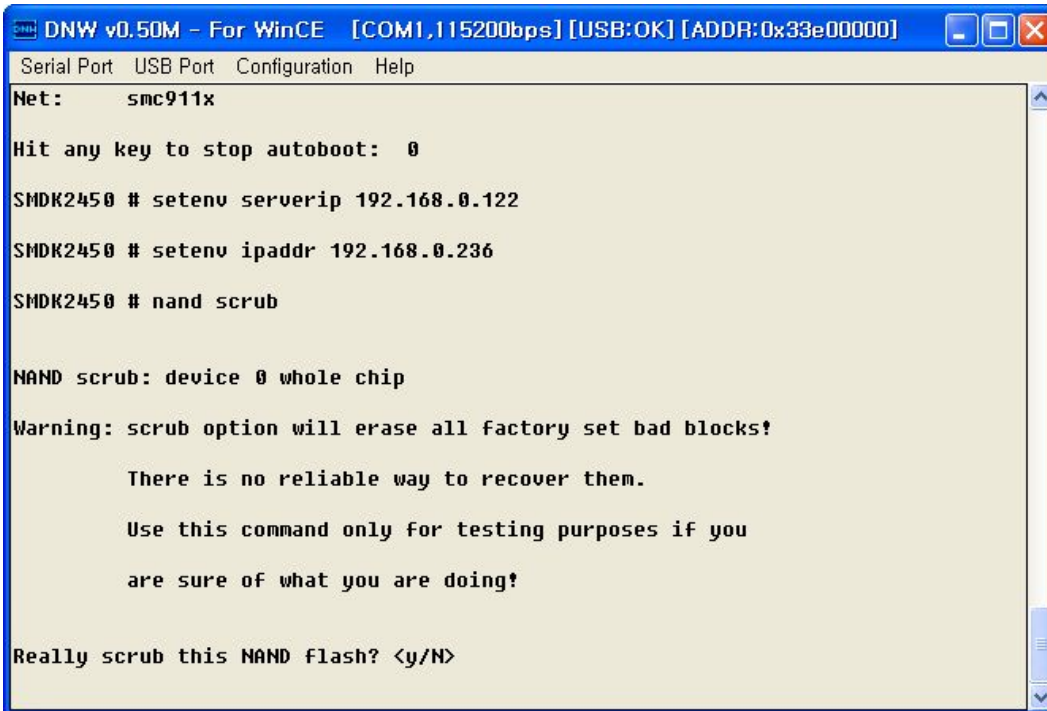
In:      serial
Out:     serial
Err:     serial
Net:     smc911x

Hit any key to stop autoboot:  0

SMDK2450 # setenv serverip 192.168.0.112
SMDK2450 # setenv ipaddr 192.168.0.236
SMDK2450 # nand scrub
```

The last three lines of the terminal output are enclosed in a red rectangular box.

Y -> Enter



```

DNW v0.50M - For WinCE [COM1,115200bps] [USB:OK] [ADDR:0x33e00000]
Serial Port  USB Port  Configuration  Help
Net:         smc911x

Hit any key to stop autoboot:  0

SMDK2450 # setenv serverip 192.168.0.122

SMDK2450 # setenv ipaddr 192.168.0.236

SMDK2450 # nand scrub

NAND scrub: device 0 whole chip

Warning: scrub option will erase all factory set bad blocks!

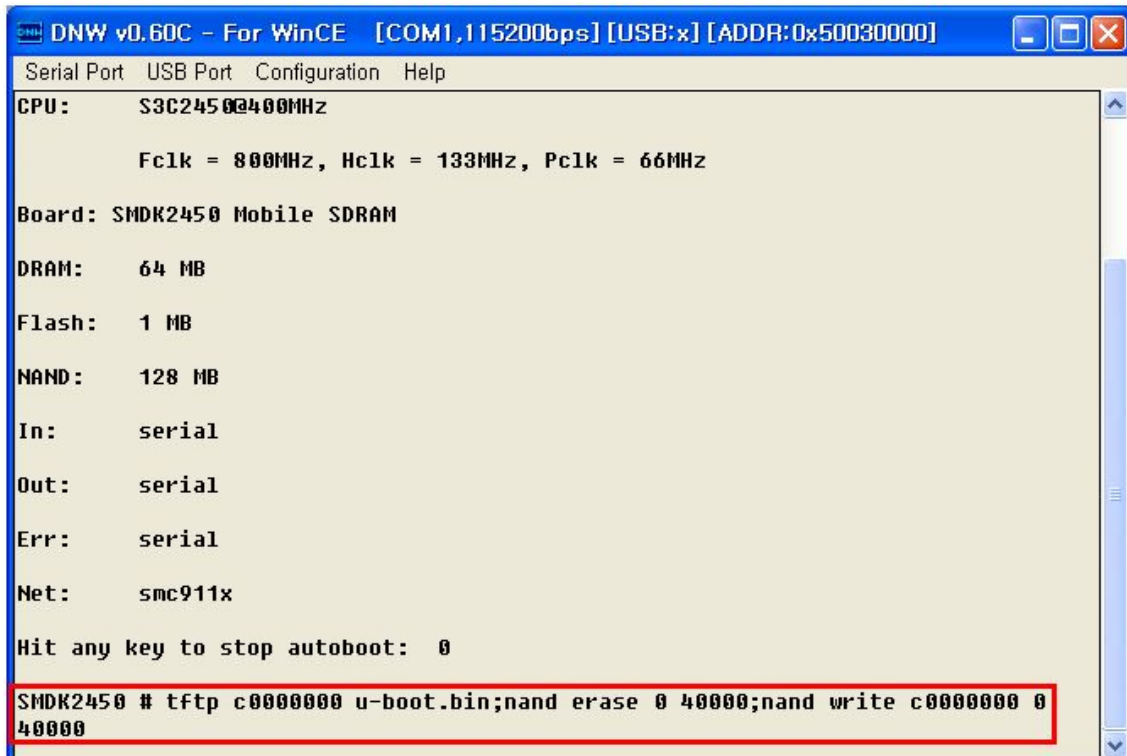
        There is no reliable way to recover them.

        Use this command only for testing purposes if you
        are sure of what you are doing!

Really scrub this NAND flash? <y/N>

```

**tftp c0000000 u-boot.bin;nand erase 0 40000;nand write c0000000 0 40000**



```

DNW v0.60C - For WinCE [COM1,115200bps] [USB:x] [ADDR:0x50030000]
Serial Port  USB Port  Configuration  Help
CPU:        S3C2450@400MHz

          Fc1k = 800MHz, Hc1k = 133MHz, Pc1k = 66MHz

Board: SMDK2450 Mobile SDRAM

DRAM:      64 MB

Flash:     1 MB

NAND:      128 MB

In:        serial

Out:       serial

Err:       serial

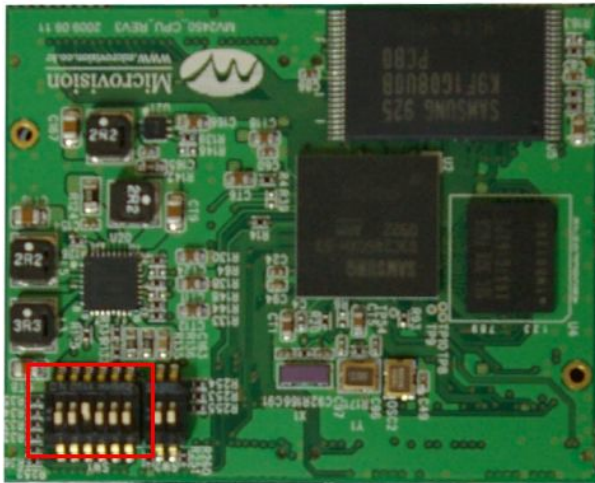
Net:       smc911x

Hit any key to stop autoboot:  0

SMDK2450 # tftp c0000000 u-boot.bin;nand erase 0 40000;nand write c0000000 0
40000

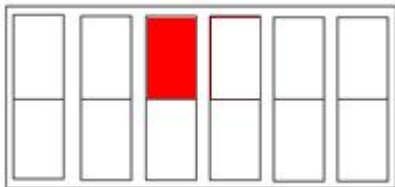
```

Now, You have to download zImage, File System.  
First, Set up NAND boot mod after Turn off.

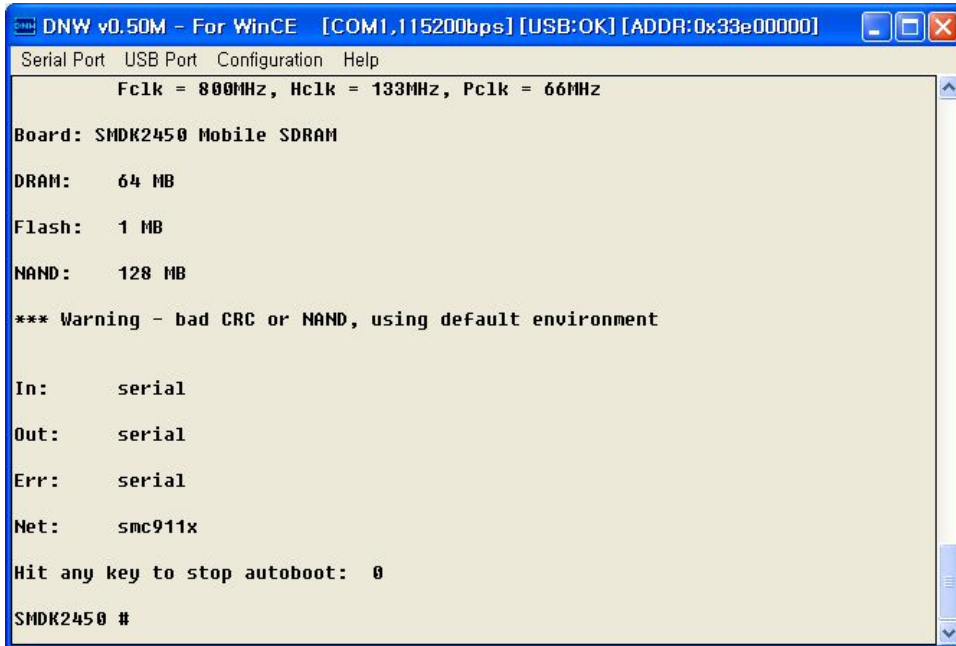


**Setting switch Main of board**

NAND Flash mode



Turn on, The following window appears on your screen.



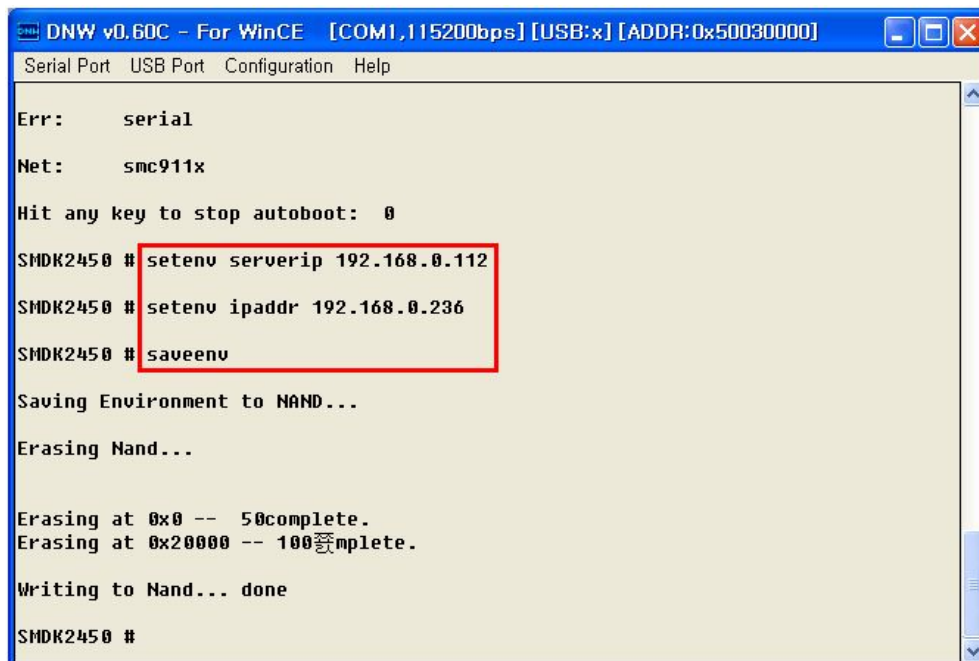
Set up IP for downloading.

IP 설정 방법

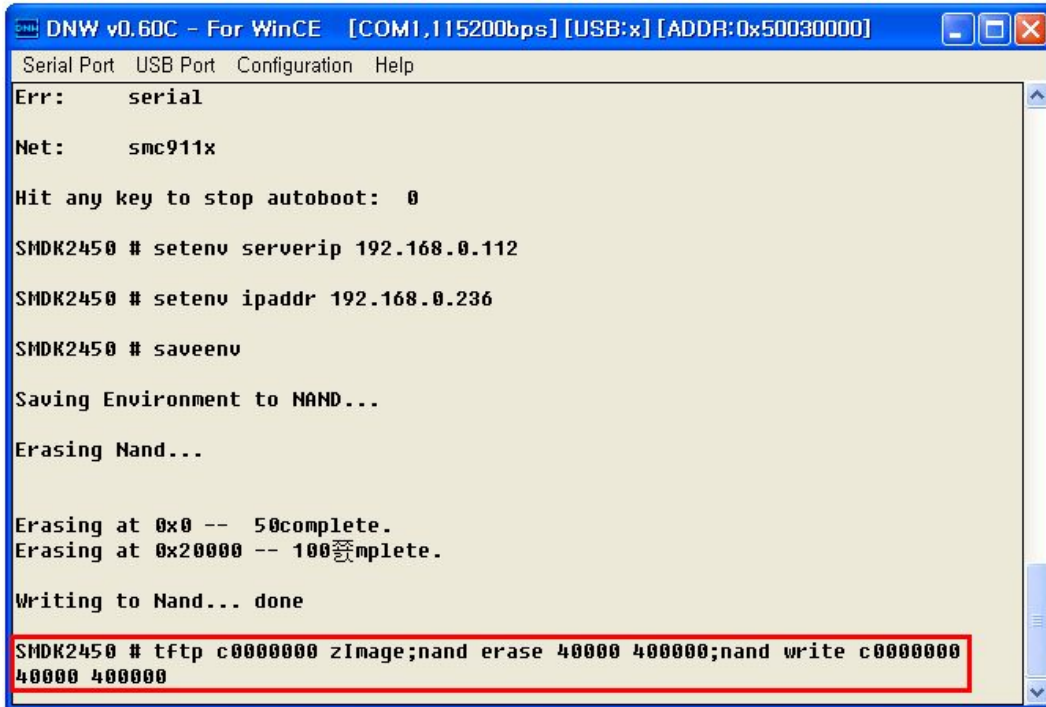
TFTP Server IP : **setenv serverip 192.168.0.122**

Device PC IP : **setenv ipaddr 192.168.0.236**

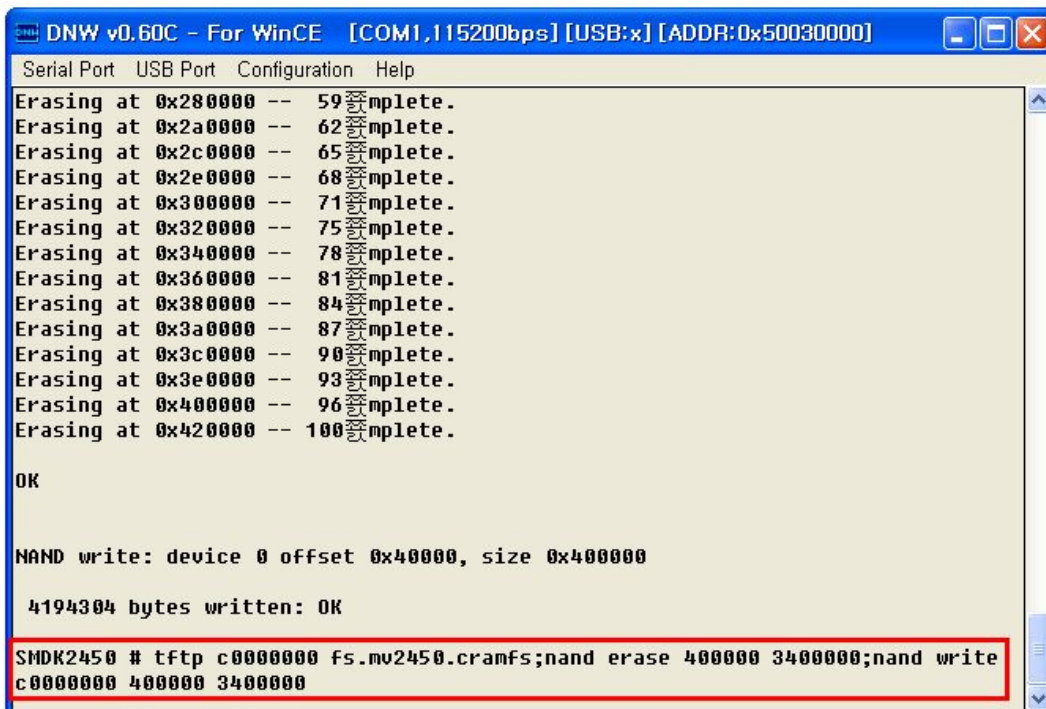
Save : **saveenv**

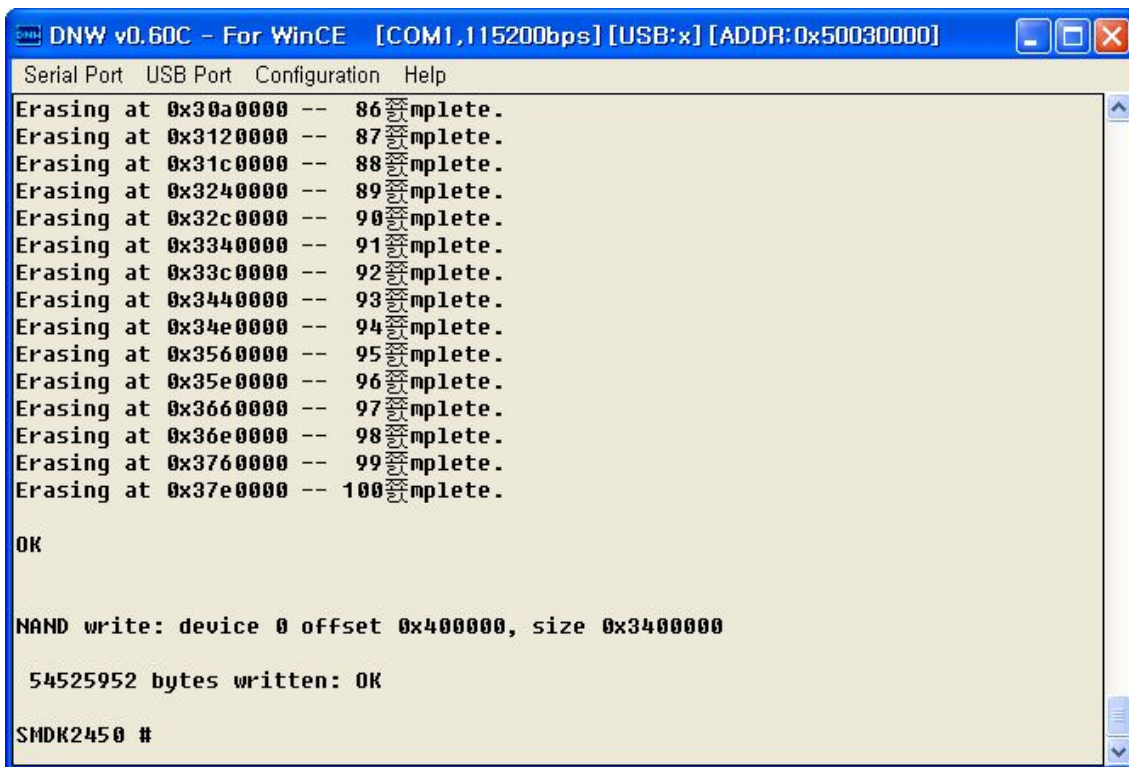


tftp c0000000 zImage;nand erase 40000 400000;nand write c0000000 40000 400000

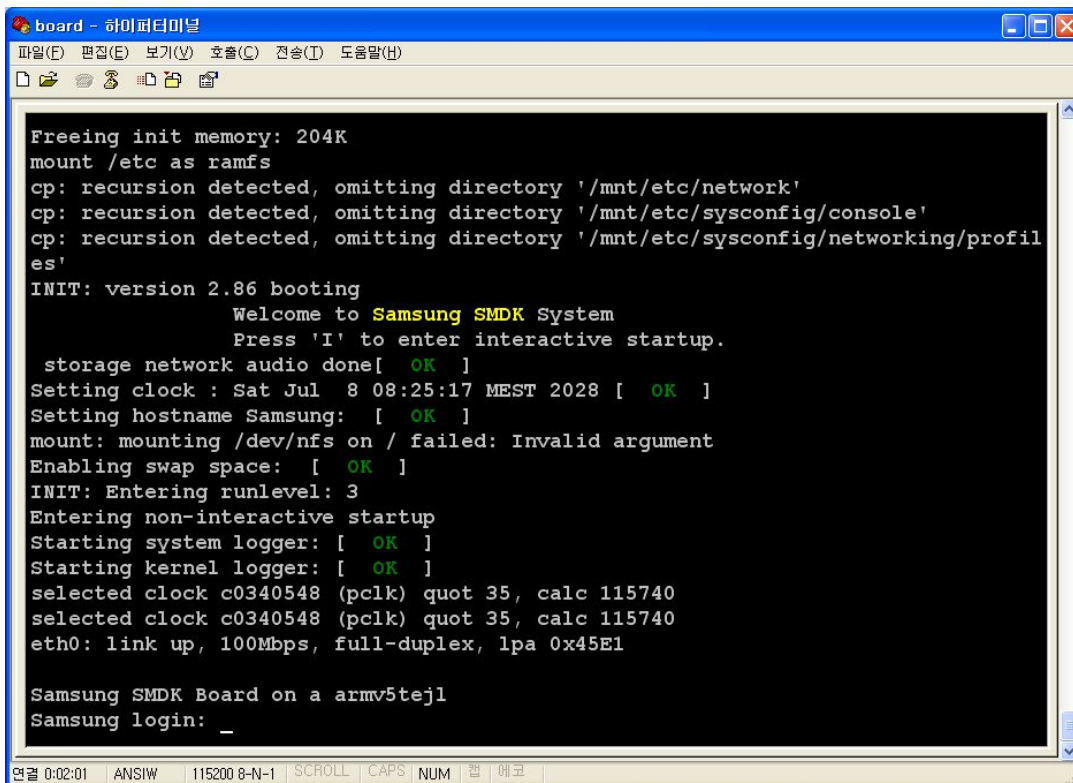


tftp c0000000 fs.mv2450.cramfs;nand erase 400000 3400000;nand write c0000000 400000 3400000





I have been using hyper terminal.  
Because hyper terminal is good for monitoring better DNW.



## 7. Doing Camera Application

Please connect Camera with Board as a picture.

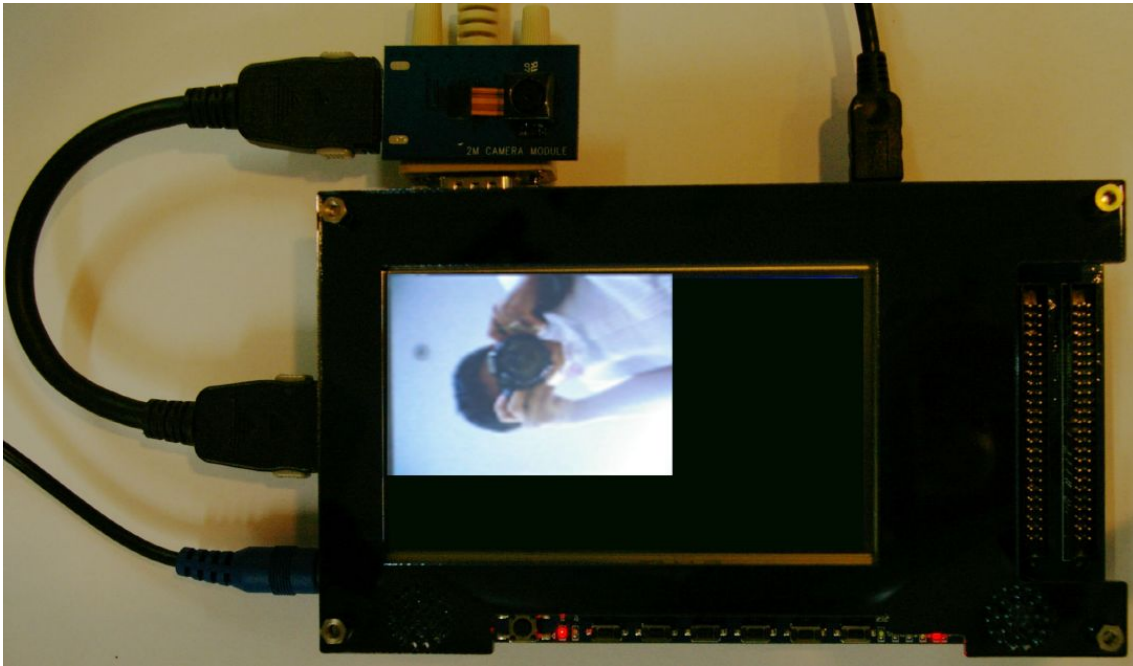


Following command

```
cp: recursion detected, omitting directory '/mnt/etc/sysconfig/networking/profiles'
INIT: version 2.86 booting
      Welcome to Samsung SMDK System
      Press 'I' to enter interactive startup.
  storage network audio done[ OK ]
Setting clock : Sat Jul  8 08:31:34 MEST 2028 [ OK ]
Setting hostname Samsung: [ OK ]
mount: mounting /dev/nfs on / failed: Invalid argument
Enabling swap space: [ OK ]
INIT: Entering runlevel: 3
Entering non-interactive startup
Starting system logger: [ OK ]
Starting kernel logger: [ OK ]
selected clock c0340548 (pclk) quot 35, calc 115740
selected clock c0340548 (pclk) quot 35, calc 115740
eth0: link up, 100Mbps, full-duplex, lpa 0x45E1

Samsung SMDK Board on a armv5tej1
Samsung login: rootselected clock c0340548 (pclk) quot 35, calc 115740

[root@Samsung ~]# ls
091.mp3*      fbcam 320x240*  madplay*
[root@Samsung ~]# ./fbcam_320x240
```



<It is working>