Installation Guide for MV2443 (Linux 2.6)

MicroVision.,Cd.Ltd

Document Information

Version	1.0
File Name	MV2443 Windows CE 5.0 Guide.doc
Date	2009. 1. 19
Satus	Working

Revision History

Date	Version	Update Descriptions	Editor
2009.01.19.	V1.0	First Edition	Jongill Wee

MV2443-LCD WinCE BSP Guide

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

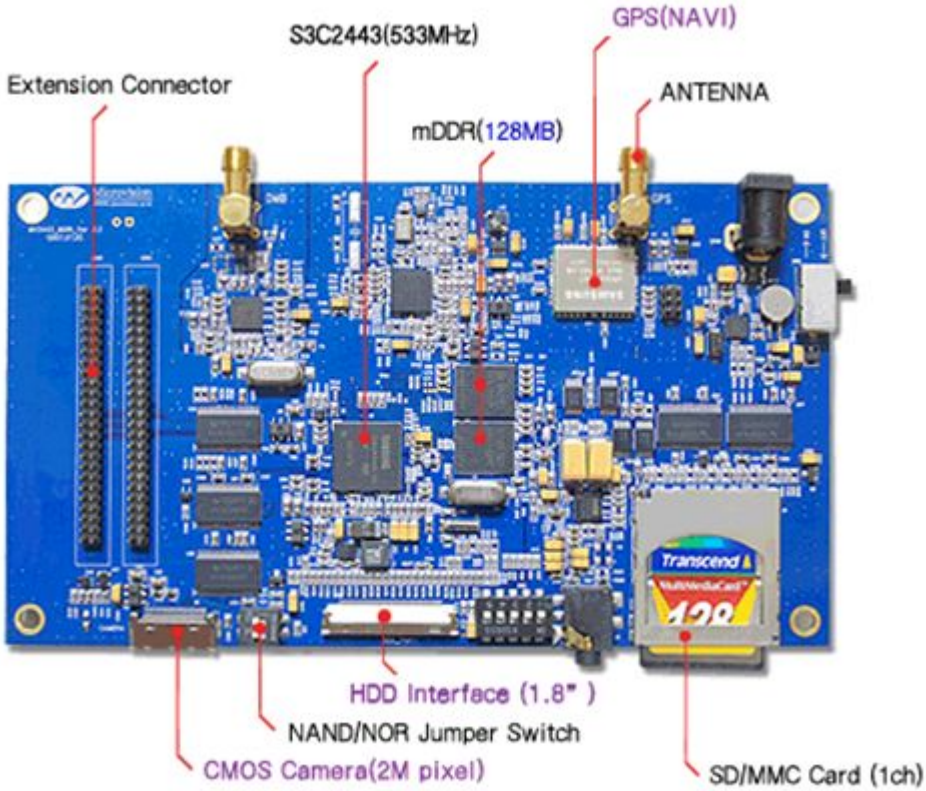
The system controller consists of three parts; reset control, system clock control, and system power-management control. The system clock control logic in MV2443 can generate the required system clock signals which are the inputs of ARM920T, several AHB blocks, and APB blocks. There are two PLLs in MV2443 to generate internal clocks. One is for general functional blocks, which include ARM, AHB, and APB. The other is for the special functional clocks which are the USB, I2S and camera interface clock. Software program control the operating frequency of the PLLs, internal clock sources and enabled or disabled the clocks to reduce the power consumption.

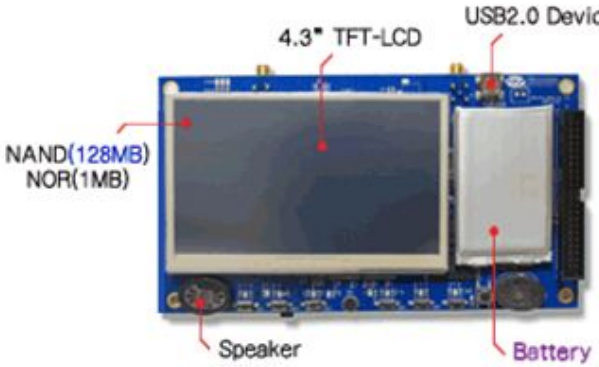
MV2443 has various power-down modes to keep optimal power consumption for a given task. The power-down modes consists of four modes; NORMAL mode, IDLE mode, STOP mode, and SLEEP mode. In NORMAL mode, the input clock of each block is enabled or disabled according to the software to eliminate the power consumption of unused blocks for a certain application. For example, if an UART is not needed, the software can disable the input clock independently. The major power dissipation of MV2443 is due to ARM core, since the operating speed is relative higher than that of the other blocks. Typically, the operating frequency of the ARM core is 533MHz, while the AHB blocks and the APB blocks operate on 133MHz and 66MHz, respectively. Thus, the power control of the ARM core is major issue to reduce the overall power dissipation in MV2443, and IDLE mode is supported for this purpose. In IDLE mode, the ARM core is not operated until the external interrupts or internal interrupts. The STOP mode freezes all clocks to all peripherals as well as the ARM core by disabling PLLs. The power consumption is only due to the leakage current and the minimized alive block in MV2443. SLEEP mode is intended to disconnect the internal power. So, the power consumption due to the ARM core and the internal logic except the wake-up logic will be nearly zero in the SLEEP mode. In order to use the SLEEP mode two indenpendent power sources are required. One of the two power soruces supplies the power for the wake-up logic. The other one supplies the normal functional blocks including the ARM core. It should be controlled in order to turn ON/OFF with a special pin in MV2443. The detailed description of the power-saving modes such as the entering sequence to the specific power-down mode or the wake-up sequence from a power-down mode is given in the following Power Management section.

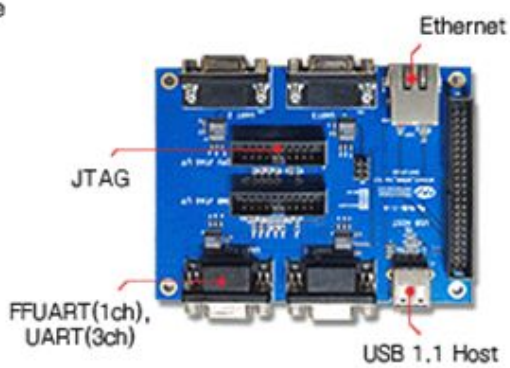
2. Packages

Item	Description	
Board	Base Board	S3C2443(534MHz), DDR2(128MB), NAND Flash(128MB), Camera(2.0M), GPS, SD/MMC(2), HDD I/F(1) USB 1.1 Host/Device, USB 2.0 Device, Ethernet(1), Audio(SPK/MIC), Keypad, I/O Extension.
	Debug Board	UART(4), LAN(1) Jtag
	LCD	4.3" Wide TFT-LCD (480 * 272, 64K Color, Touch)
Power	AC Power Adaptor (Output: DC 5V, 3A) 1 EA	
Cable	Serial Cable 1EA, USB Cable 1EA	
CD	BSP Images & Sources, Schematic, Data Sheet, Document	
Antenna	GPS Antenna 1EA	
Camera	2.0Mega Pixel (CMOS type)	

3. An Arrangement Plan



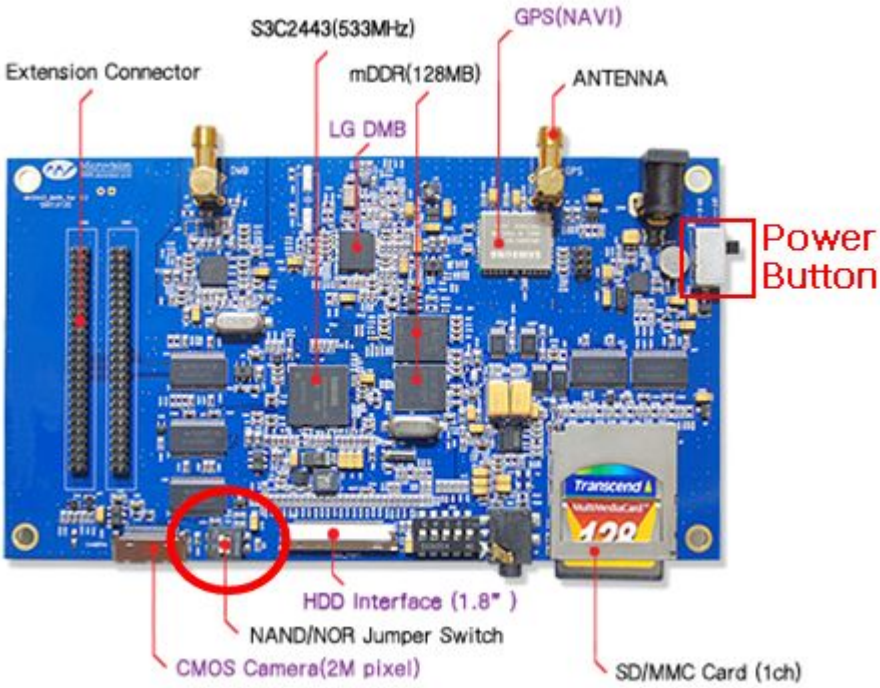


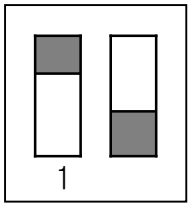


4. H/W Specifications

Item	Description
CPU(MCU)	Samsung ARM 920T CORE 400/533MHz
DDR2(SDRAM)	Samsung 64MB*2 (m)DDR 16bit access, clock speed 266MHz
NAND FLASH	Samsung NAND LARGE BLOCK 1Gb(128MB), 8 bit bus
NOR FLASH	AMD NORFLASH 8Mb(1MB), 16Bit
Ethernet	CS8900. 10 BASE-T, LINK LED
Display	4.3" Wide TFT-LCD(WVGA, 480X272, 16.7M Color), Touch
GPS(Navigation)	Samsung GPD14B001007(GPS Module)
Camera	2.0Mega Pixel CIS Module, CMOS type
UART	UART 4 Ports(Debug 1, UART 3)
USB	USB 1.1 Host, USB 2.0 Device
HDD Interface	1.8" HDD Interface
Audio	WM8960. STEREO 1000mW , MIC IN, HEAD SET
SD/MMC	STANDARD SD
I/O Extension	GPIO, Address, Data Bus Pin (26*2)
JTAG	10 Pin JTAG Emulator Port 1 EA
Power	DC 5V Jack, Battery Connector

5. Setting Boot of mode



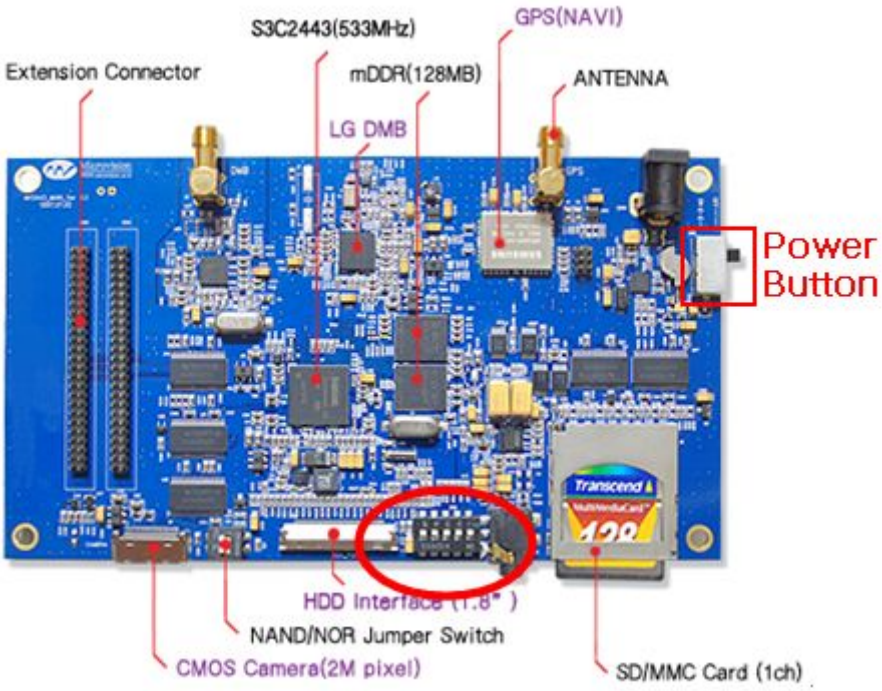


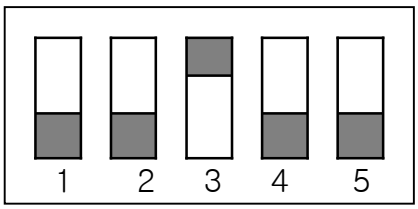
<NOR>

It connected both nCS0 and nCS1.

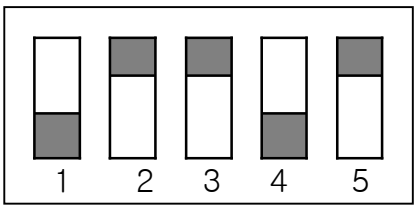
You want to work NOR Flash which by nCS0, Please Setup Number 1 “ON”.

6. Setting SW2 your wants Boot of mode



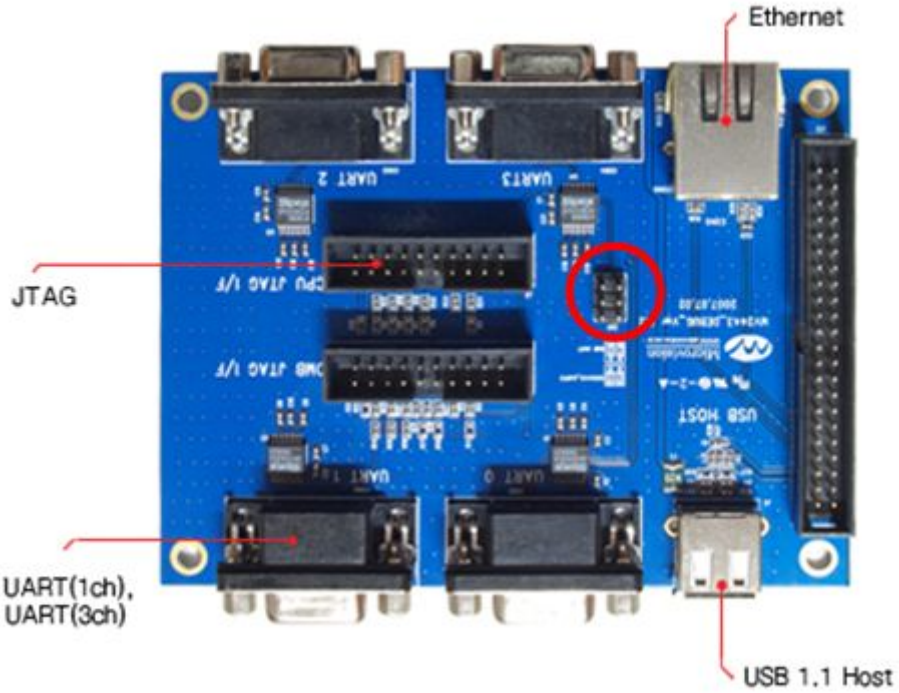


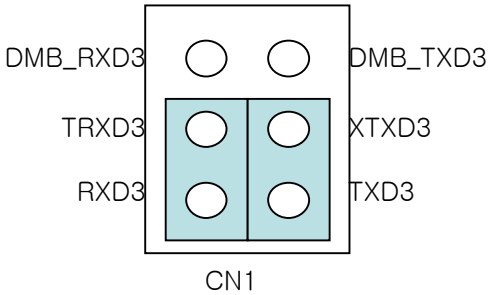
<NAND BOOT>



<NOR BOOT>

7. To use the CN1 which is connected TX3 on the Debug of Board





<TX3>

8. Getting Started

Filename	Description
cross-3.3.4.tar.bz2	Toolchain for Compilation
s3c-linux-2.6.16_v0.91.tar.bz2	Kernel
s3c-uboot-1.1.4_v1.0.tar.bz2	Bootloader
rootfs_qtopia.cramfs	Qtopia window Root file System

After made folder mv2443 through the command of mkdir, Copy it to the working directory /mv2443

[root@localhost mv2443]# ls
cross-3.3.4.tar.bz2
s3c-linux-2.6.16_v0.91.tar.bz2
s3c-uboot-1.1.4_v1.0.tar.bz2
rootfs_qtopia.cramfs
[root@localhost mv2443]#

DDR of Area.

BANK	Physical Address	Virtual Address	Size
0	0x30000000 ~ 0x33FFFFFF	0xC0000000 ~ 0xC3FFFFFF	64 MB
1	0x38000000 ~ 0x3BFFFFFF	0xC8000000 ~ 0xCBFFFFFF	64 MB

NAND of Area

NAND block No.	NAND block offset	Size	Contents
0x0 (0x0 ~ 0xF)	0x0 (0x0 ~ 0x3FFFF)	256Kb	U-Boot Bootloader
0x10 (0x10 ~ 0x7F)	0x40000 (0x40000 ~ 0x1FFFFF)	1.75MB	Linux Kernel
0x80 (0x80 ~ 0xC7F)	0x200000 (0x200000 ~ 0x31FFFFF)	48MB	Root file system
0xC80 (0xC80 ~ end)	0x3200000 (0x3200000 ~ end)	(NAND 50 MB)	Extra area

9. Installing Toolchain

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-3.3.4.tar.bz2 to the working directory /mv2443

The above commands will generate the "3.3.4" folder under the /mv2443/ directory. Move this folder under "/usr/local/arm/" directory.

Please follow the commands

```
[root@localhost mv2443]# madir –p /usr/local/arm
[root@localhost mv2443]# tar xjvf cross-3.3.4.tar.bz2
[root@localhost mv2443]# mv 3.3.4 /usr/local/arm
[root@localhost mv2443]# exportPATH=$PATH:/usr/local/arm/3.3.4/bin
```

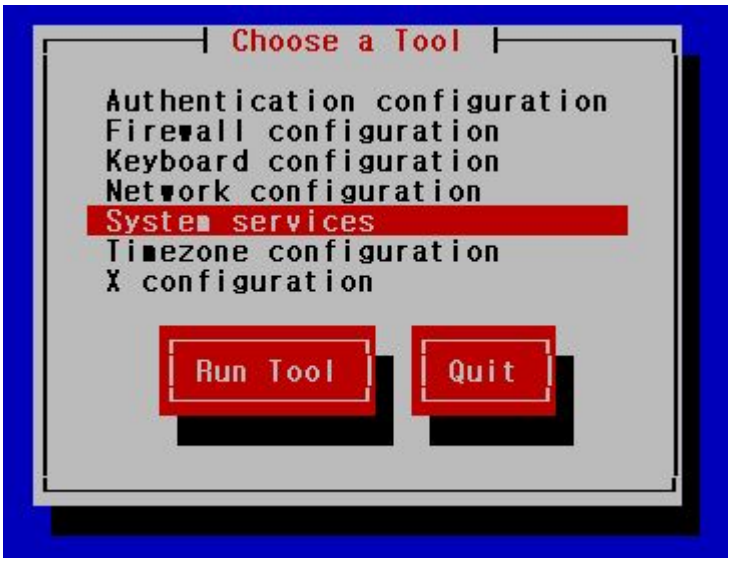
The toolchain object files such as arm compilers, loaders etc. will be available in the '/usr/local/arm/3.3.4/bin' directory.

10. Setting Up TFTP Server

Follow the command

[root@localhost mv2443]# setup

Choose one from "System services"



Choose one from "tftp"

Click "OK". Finally "quit" setup utility and execute Follow the command

[root@localhost mv2443]# service xinetd restart

11. Building U-boot

u-boot source file is compressed with tarball 's3c-uboot-1.1.4_v1.0.tar.bz2'. Extract it by executing following command.

```
[root@localhost mv2443]# tar jxvf s3c-uboot-1.1.4_v1.0.tar.bz2
[root@localhost mv2443]# cd s3c-uboot-1.1.4
[root@localhost s3c-uboot-1.1.4]# vim Makefile
```

Go to 's3c-uboot-1.1.4' directory created after extracting the tarball and then execute the 'vi Makefile' command to check the 'CROSS_COMPILE' setting for uboot bootloader

ifeq ($(ARCH),arm)
CROSS_COMPILE = /usr/local/arm/3.3.4/bin/arm-linux-
endif

```
ifeq ($(ARCH),ppc)
CROSS_COMPILE = powerpc-linux-
endif
ifeq ($(ARCH),arm)
CROSS_COMPILE = /usr/local/arm/3.3.4/bin/arm-linux-
endif
ifeq ($(ARCH),i386)
ifeq ($(HOSTARCH),i386)
CROSS_COMPILE =
else
CROSS_COMPILE = i386-linux-
```

Now execute the 'make smdk2443_config' command.

```
[root@localhost s3c-uboot-1.1.4]# make smdk2443_config

Configuring for smdk2443 board...
[root@localhost s3c-uboot-1.1.4]#

Finally compile u-boot by executing 'make' command.
[root@localhost s3c-uboot-1.1.4]# make
```

If the compilation of u-boot progresses well, u-boot binary image file will be created under '/tftpboot' and 's3c-uboot-1.1.4' directory.
In Next chapter we will port u-boot (bootloader), kernel image, and root file system to target board. To do this work more conveniently, it is good to collect all the compiled images to '/tftpboot' directory.
To port binary image file to target board, run tftp server service on your computer.
u-boot.bin image will automatically copy in /tftpboot directory. If it is not copied, copy it manually.

[root@localhost s3c-uboot-1.1.4]# cp u-boot.bin /tftpboot/

Setting U-Boot for IP address

[root@localhost s3c-uboot-1.1.4]# vi include/configs/smdk2443.h

```
#define CONFIG_ETHADDR      00:40:5c:26:0a:5b
#define CONFIG_NETMASK      255.255.255.0
#define CONFIG_IPADDR       192.168.0.177
#define CONFIG_SERVERIP     192.168.0.232
```

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

12. Building the embedded Linux Kernel

Kernel source is compressed by the name of 's3c-linux-2.6.16.tar.bz2'. Extract this bz2 file by executing the following command. After extracting the kernel tarball file 's3c-linux-2.6.16' directory will be generated. Go to 's3c-linux-2.6.16.tar' directory and check 'Makefile' and '.cross_compile'.

```
[root@localhost mv2443]# tar xjvf s3c-linux-2.6.16_v0.91.tar.bz2

[root@localhost mv2443]# cd s3c-linux-2.6.16
[root@localhost s3c-linux-2.6.16]#
[root@localhost s3c-linux-2.6.16]# vi .cross_compile
```

Please make sure GCC path

```
/usr/local/arm/3.3.4/bin/arm-linux-
~
~
~
~
~
~
```

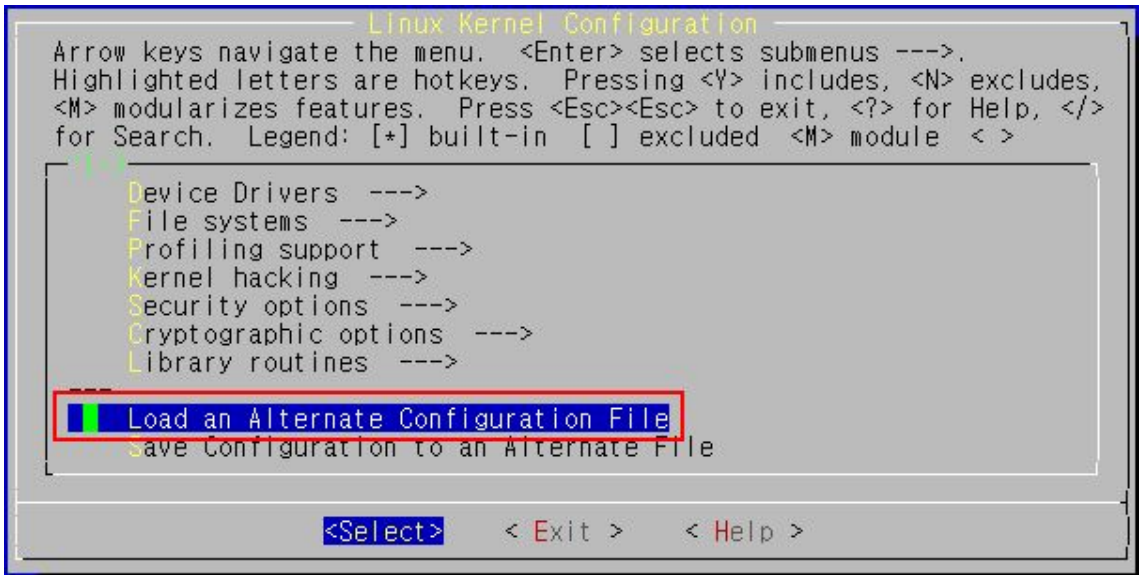
Set the values by executing 'make mv2443_defconfig' command. You can load default-configuration-file that is composed with values optimized to target board. In the case of kernel, default-configuration-files are located in 's3c-linux-2.6.16/arch/arm/configs/' directory.

After executing above commands, the Kernel image will be created in '/tftpboot' and 's3c-linux-2.6.16/arch/arm/boot' directory by the name of 'zImage'.

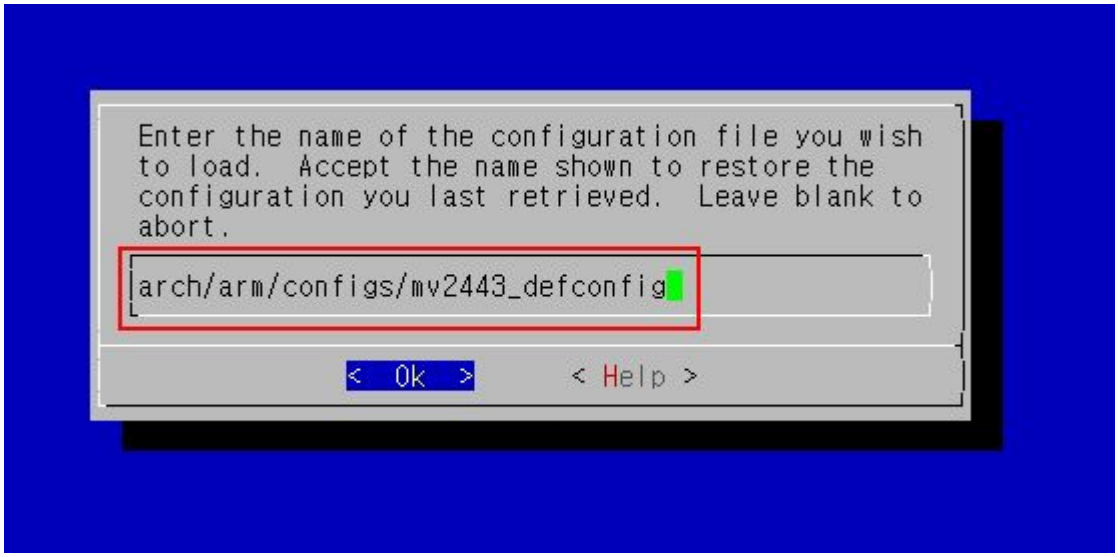
Please following command

[root@localhost s3c-linux-2.6.16]# make menuconfig

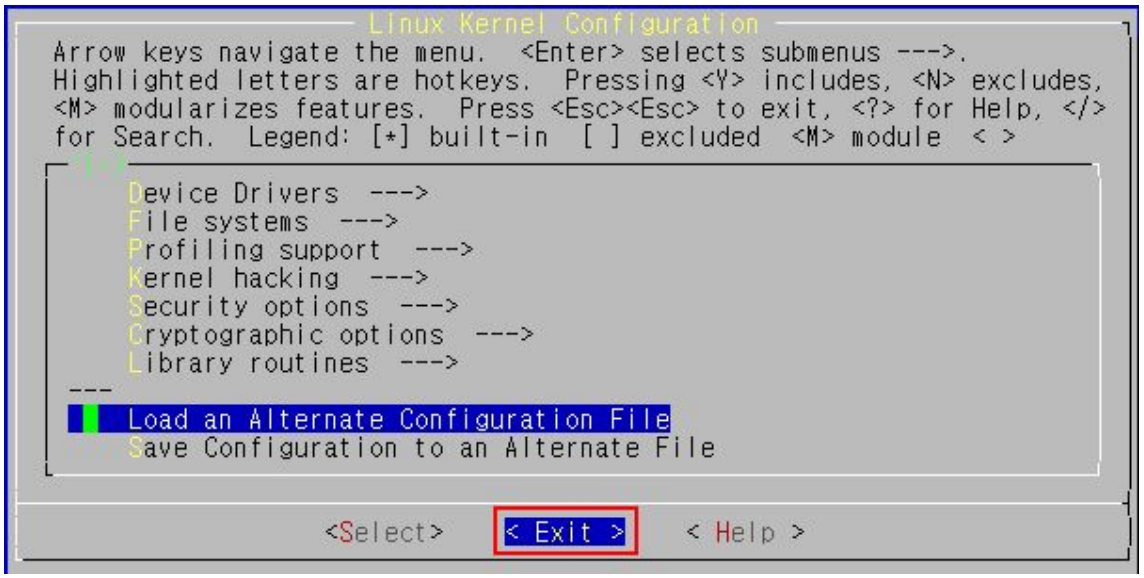
Select "Load an Alternate Configuration File"



Fill in "arch/arm/configs/mv2443_defconfig"



Select “EXIT”



After exit. Make

Please following command

[root@localhost s3c-linux-2.6.16]# make zImage

[root@localhost s3c-linux-2.6.16]# cd arch/arm/boot/

[root@localhost boot]# cp zImage /tftpboot/

Without “make menuconfig” others the Way

[root@localhost s3c-linux-2.6.16]# make mv2443_defconfig

[root@localhost s3c-linux-2.6.16]# make zImage

13. Root file System

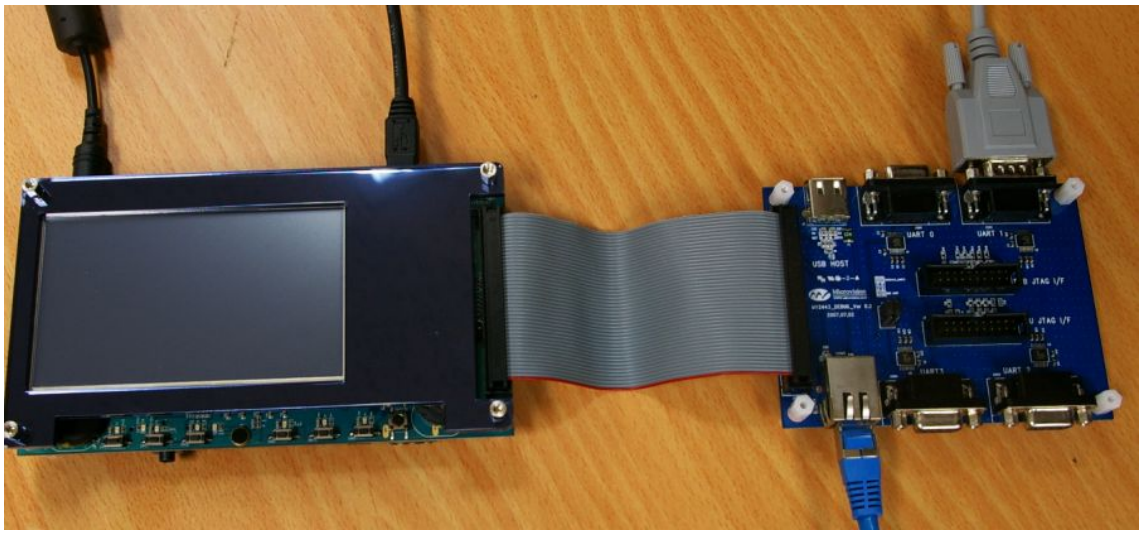
Root filesystem is composed of Cramfs (Compressed ROM file system).
To port the Root File System onto the target board easily, copy the root file system to '/tftpboot/' directory.

[root@localhost mv2443]# cp rootfs_qtopia.cramfs /tftpboot/

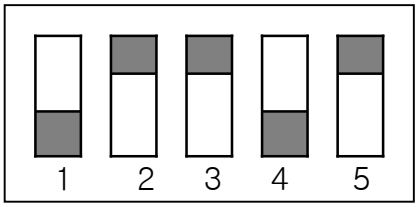
14. Setting for downloading Host PC to Board

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.

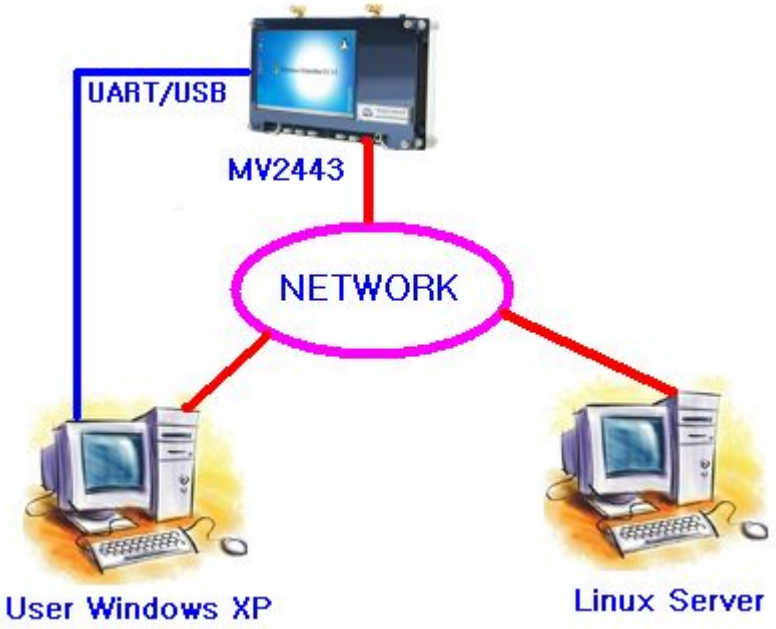


To do configuration NOR Flash through the Dipswitch.



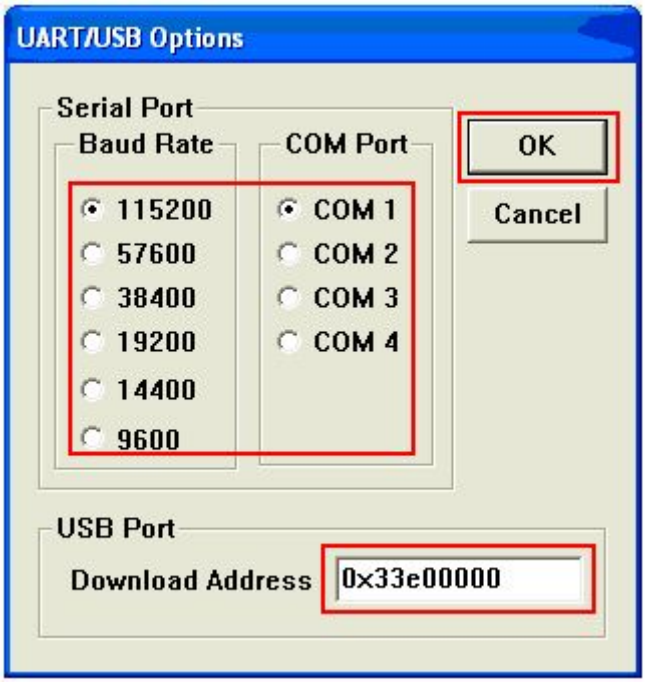
<NOR BOOT>

The ways are between DNW and Minicom through TFTP form Linux of Server. I tell you about DNW through TFTP.



Run DNW.exe in ₩SRC₩DNW₩

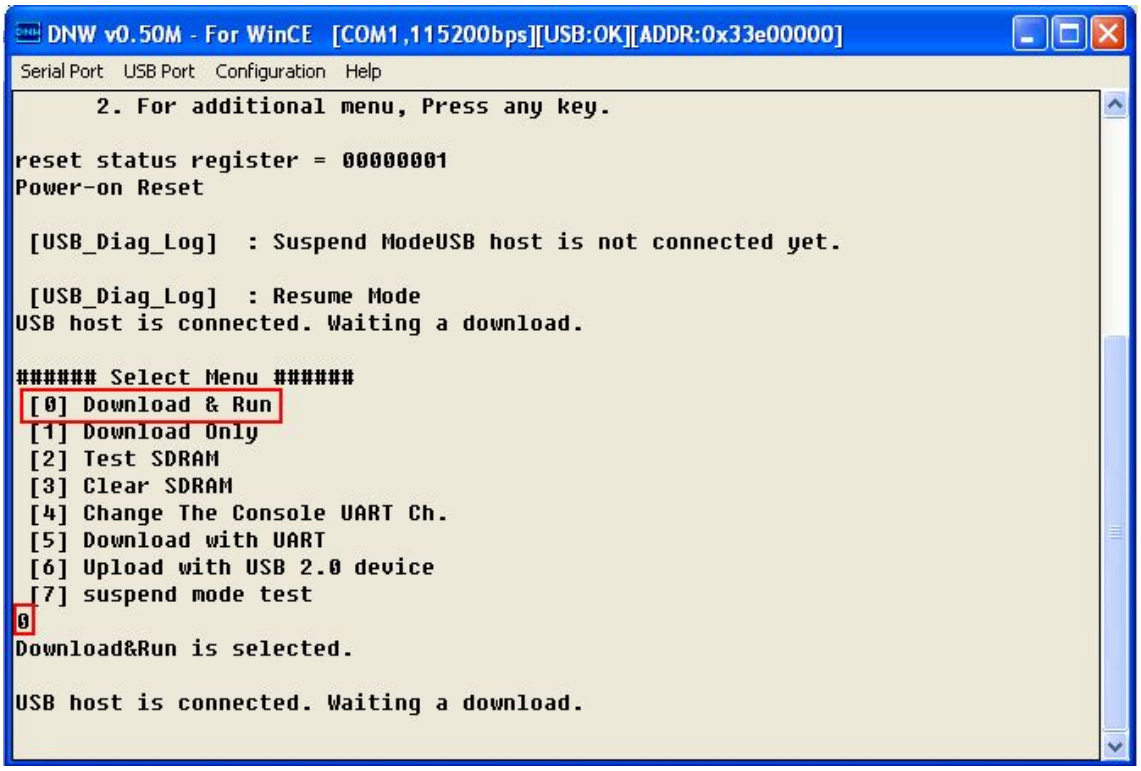
On the Configuration menu, click Options to set the UART/USB options. The following window appears on your screen. Select Baud Rate and COM Port as shown in figure "UART/USB options", enter the download address as 0x33e00000 and then click OK button.

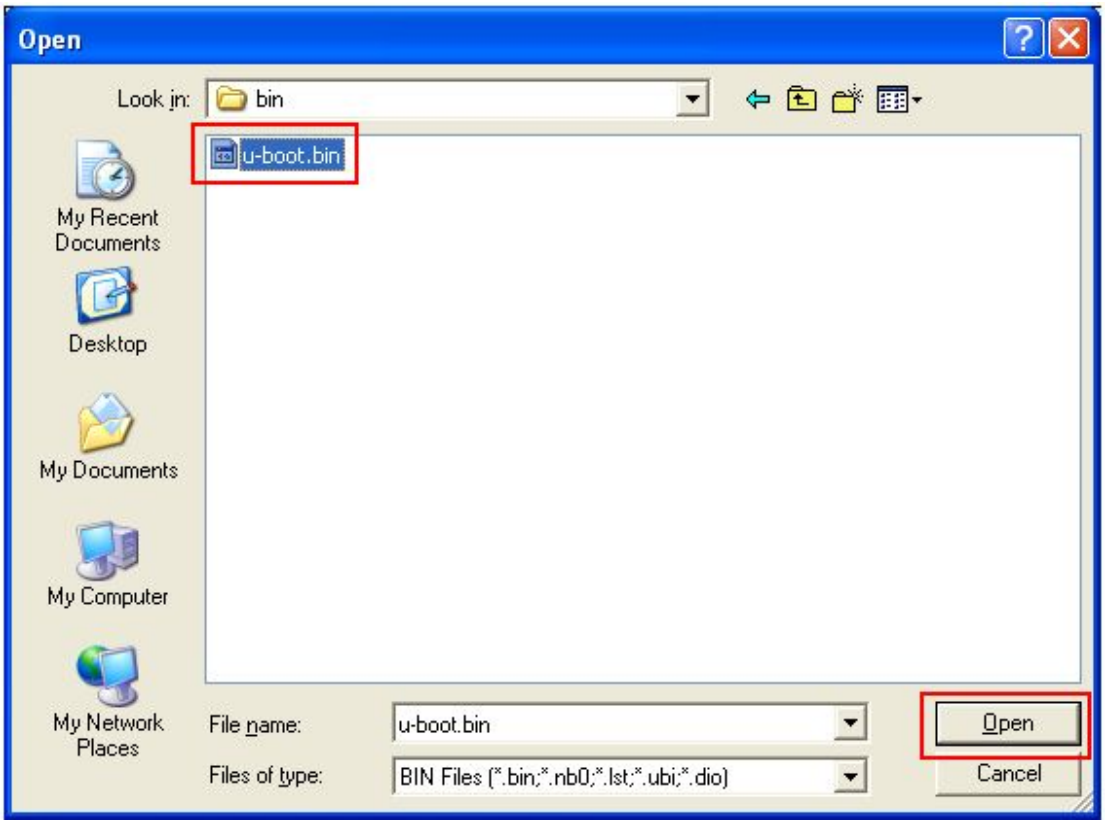


<UART/USB options>

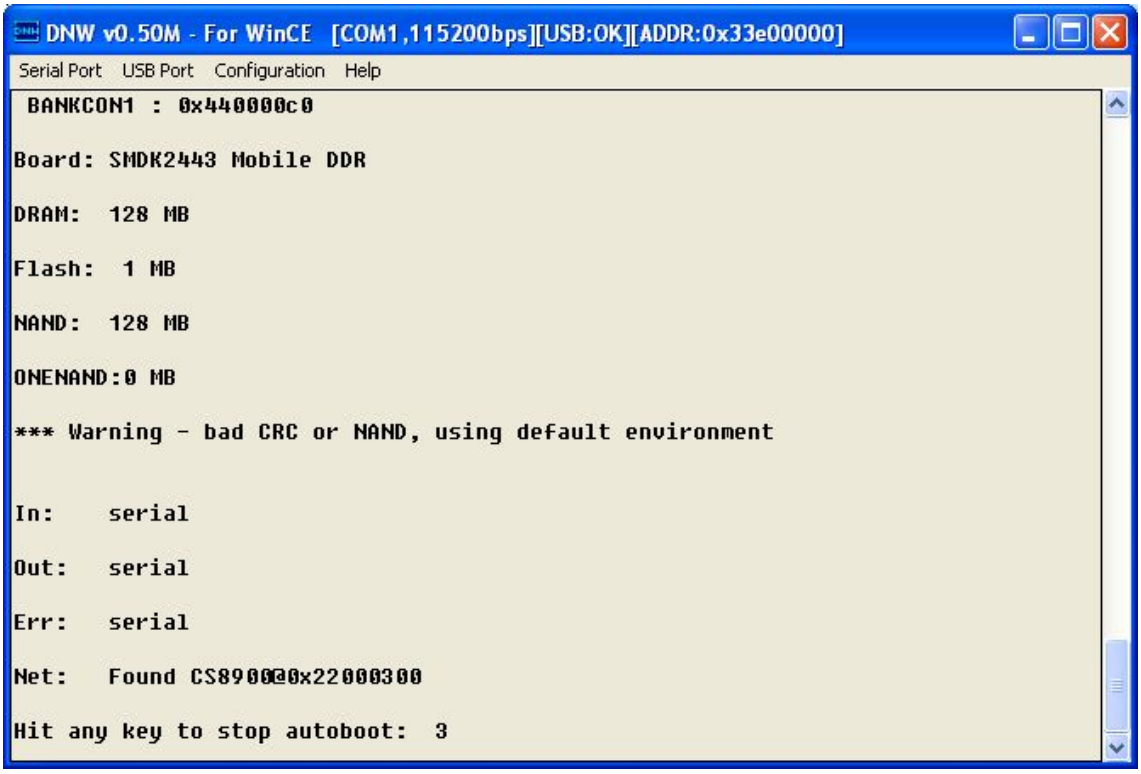
On the Serial Port menu, click Connect. Switch ON the reference board and then press any key and then install the USB driver in ₩SRC₩USB driver directory.

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.



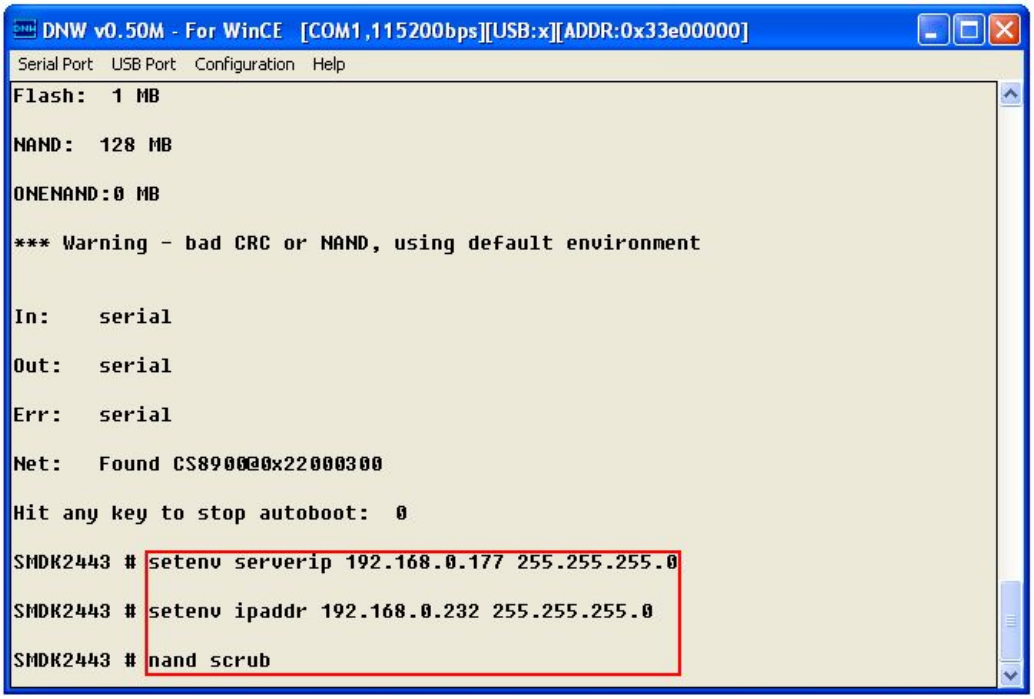
Setting IP address

Linux Server PC for taking in tftpboot.

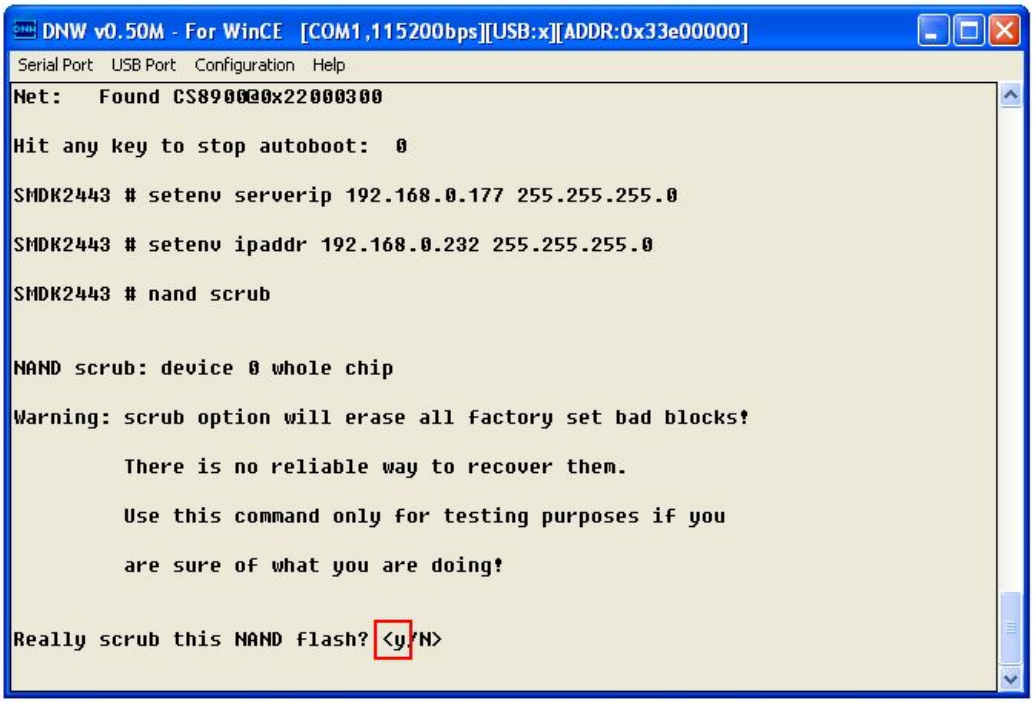
TFTP Server IP : setenv serverip 192.168.0.177 255.255.255.0

Device PC IP : setenv ipaddr 192.168.0.232 255.255.255.0

For erasing Bad blocks in NAND : nand scrub

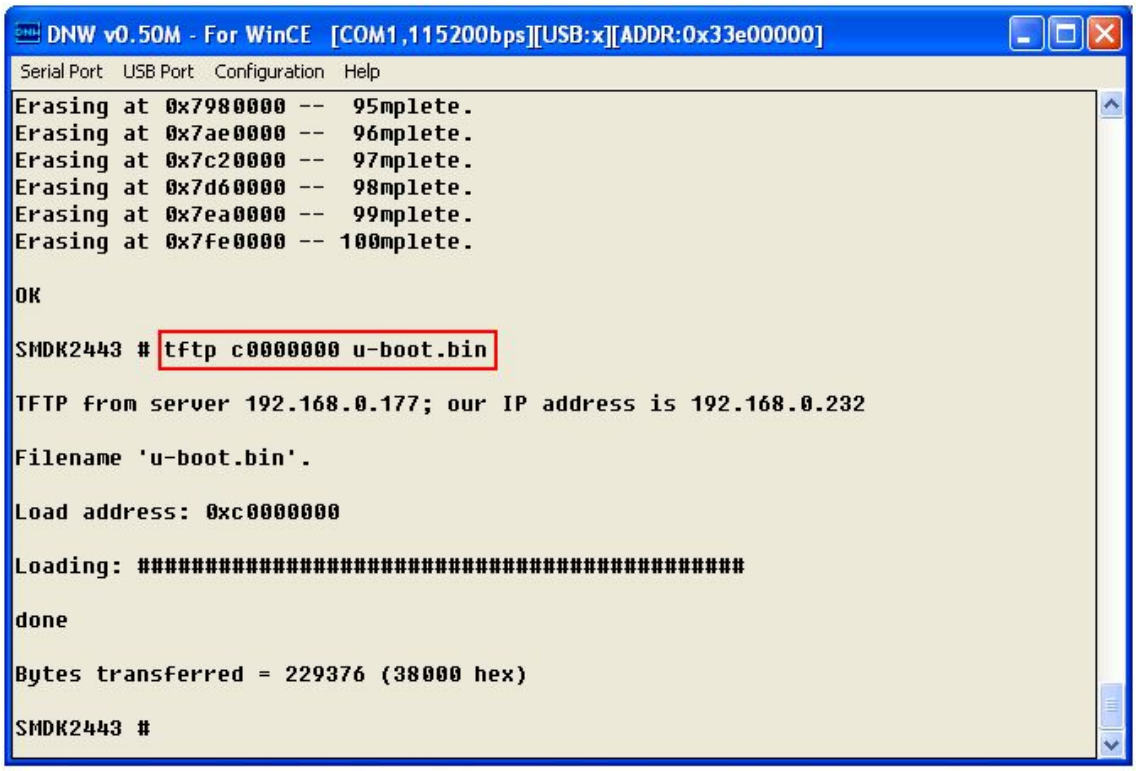


Y -> Enter



Transfer the u-boot.bin image by using following command

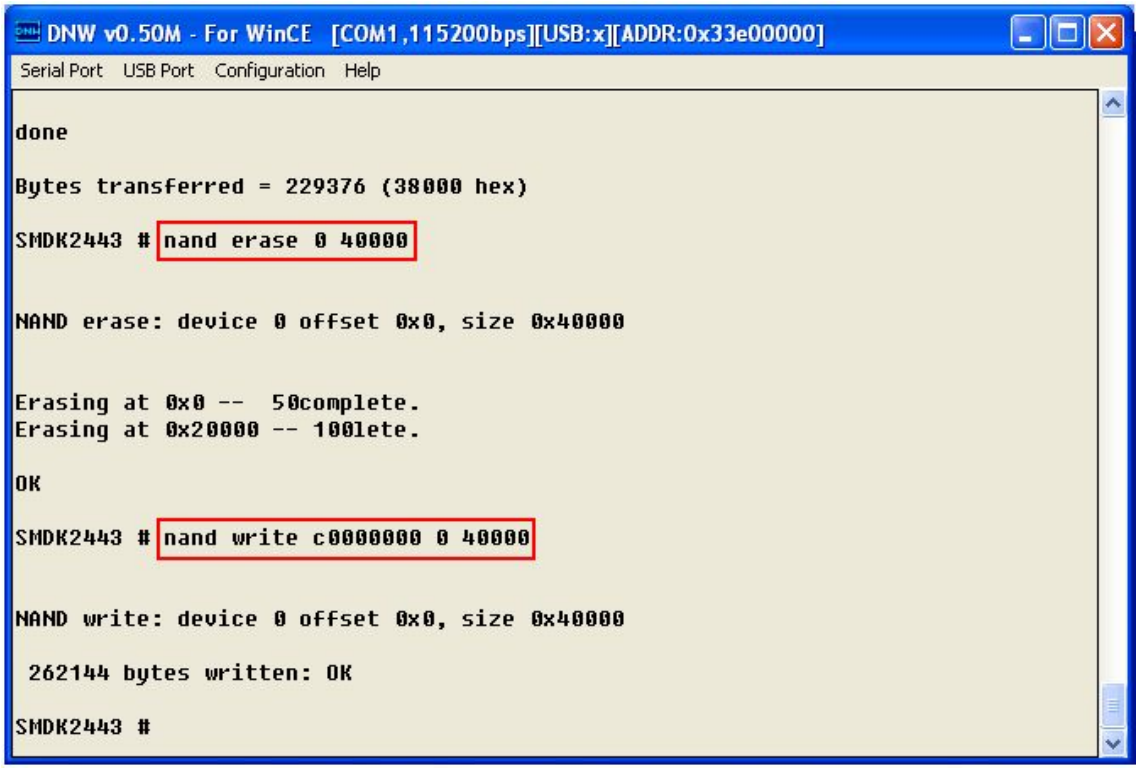
tftp c0000000 u-boot.bin



Temporary address is base address of SDRAM, c0000000 Start offset of bootloader is 0x0, it is located in NAND flash's first block number. 0xf block of NAND flash is environment space. In this case, we have written the u-boot.bin file size as 0x40000. Image size of bootloader will be below 240K, hence bootloader will occupy space within blocks (0x0~0xf, 0x40000)

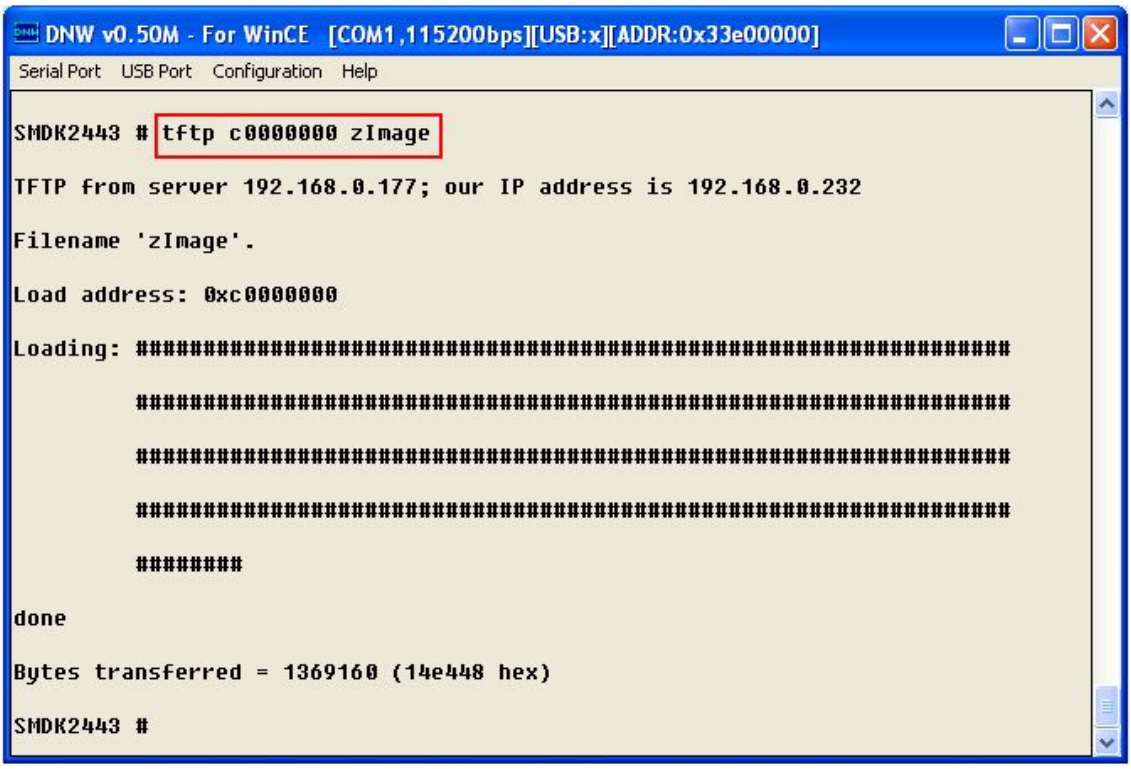
Write the u-boot.bin image to the NAND flash by using following command.

nand erase 0 40000



Transfer the zImage by using following command

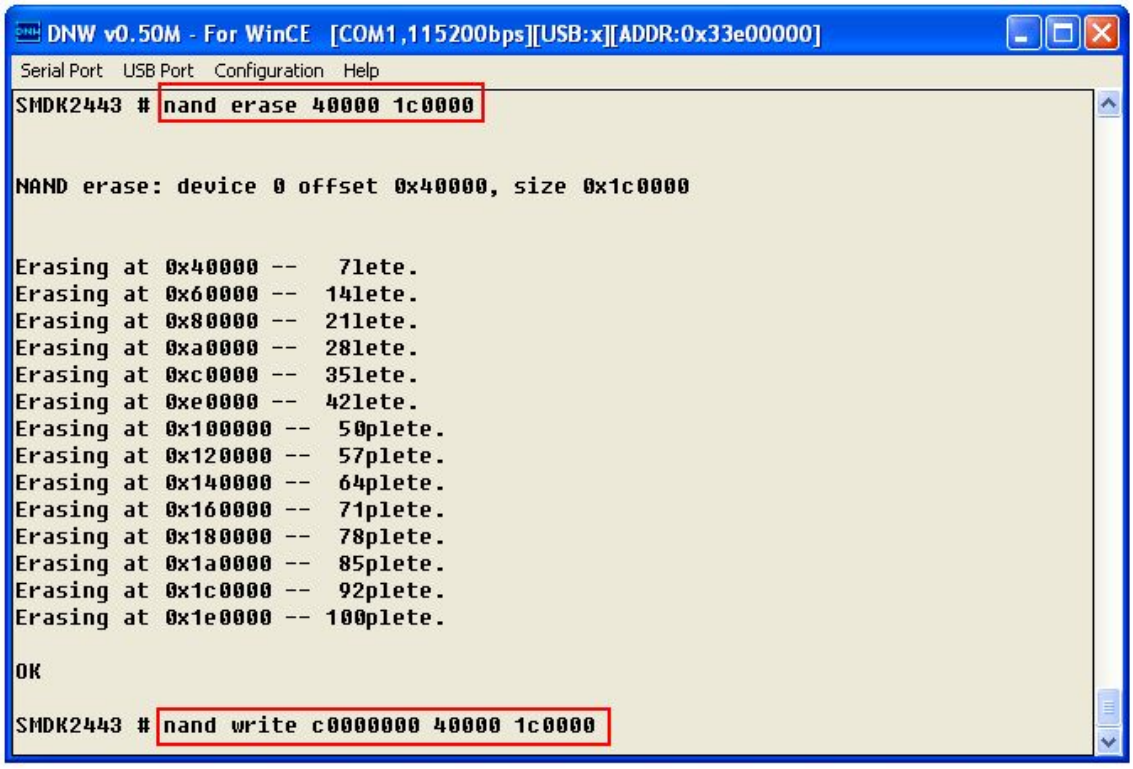
tftp c0000000 zImage



Temporary address is base address of SDRAM, 0xc0000000 Start block number of kernel is 0x10 or 0x40000 in offset. In this case, we have written the kernel image size as 1c0000, it varies depending on the menuconfig options and module selection. Kernel image can occupy up to 1.75M (0x10~0x7F blocks), so image size will be well below 1.75MB.

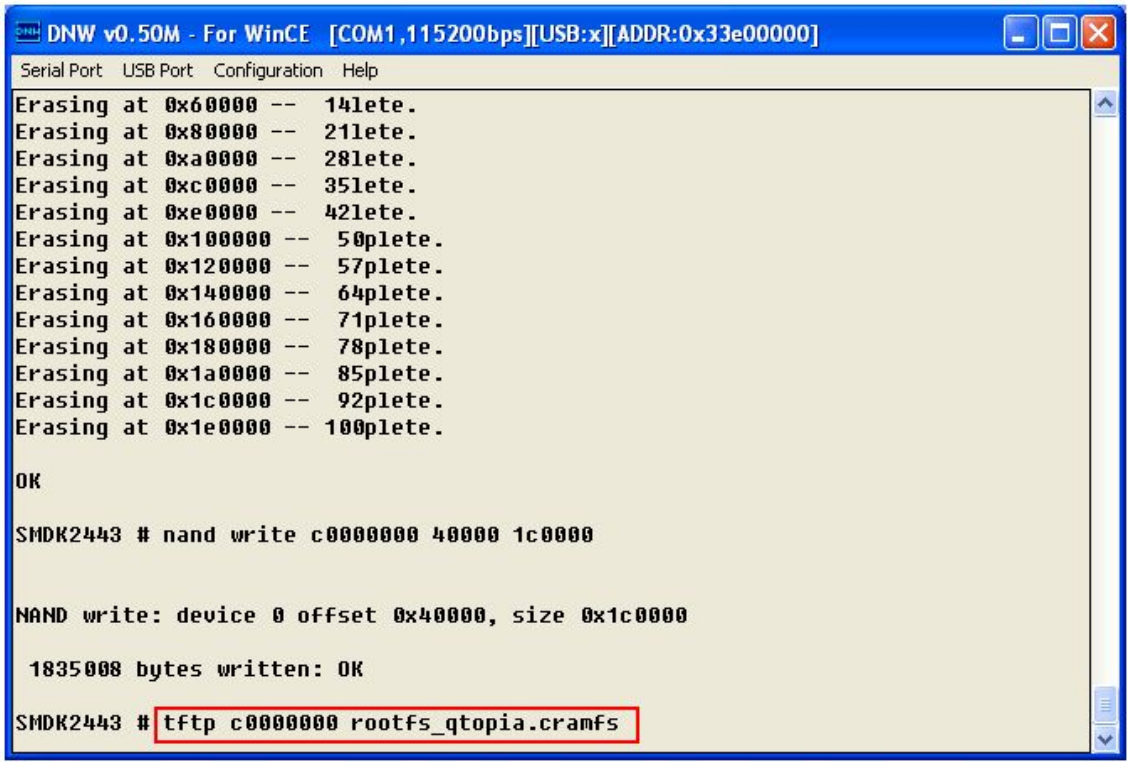
Write the zImage image to the NAND flash by using following command.

nand erase 40000 1c0000
nand write c0000000 40000 1c0000



Transfer the rootfs_qtopia.cramfs image by using following command

tftp c0000000 rootfs_qtopia.cramfs



Write the rootfs_qtopia.cramfs image to the NAND flash by using following command.

nand erase 200000 3000000

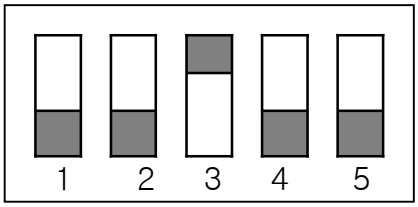
```
DNW v0.50M - For WinCE  [COM1,115200bps][USB:x][ADDR:0x33e00000]
Serial Port  USB Port  Configuration  Help

         ##################################################################
         ##################################################################
         ##################################################################
         ##################################################################
         ##################################################################
         ##################################################################
         ##################################################################
         ##################################################################
         ############################
done
Bytes transferred = 36679680 (22fb000 hex)
SMDK2443 # nand erase 200000 3000000
```

nand write c0000000 200000 3000000

```
DNW v0.50M - For WinCE  [COM1,115200bps][USB:x][ADDR:0x33e00000]
Serial Port  USB Port  Configuration  Help
Erasing at 0x2c20000 --  88mplete.
Erasing at 0x2ca0000 --  89mplete.
Erasing at 0x2d20000 --  90mplete.
Erasing at 0x2da0000 --  91mplete.
Erasing at 0x2e20000 --  92mplete.
Erasing at 0x2ea0000 --  93mplete.
Erasing at 0x2f00000 --  94mplete.
Erasing at 0x2f80000 --  95mplete.
Erasing at 0x3000000 --  96mplete.
Erasing at 0x3080000 --  97mplete.
Erasing at 0x3100000 --  98mplete.
Erasing at 0x3180000 --  99mplete.
Erasing at 0x31e0000 -- 100mplete.

OK

SMDK2443 # nand write c0000000 200000 3000000


NAND write: device 0 offset 0x200000, size 0x3000000

 50331648 bytes written: OK

SMDK2443 #
```

Temporary address is base address of SDRAM, 0xc0000000 Start block number of kernel is 80.
Image size of rootfs is about 50Mbyte

Please reboot After Setting NAND of Boot



<NAND BOOT>

15. Doing Camera Application

Please connect Camera with Board as a picture.

Following command

[root@(none) /]# cd usr/bin

[root@(none) /]# rgb_4p 320 240 16 0 0