
Linux 2.6.28 Guide

For MVC100



MicroVision Co., Ltd.

Document Information

Version	1.0
File Name	MVC100_Linux_Guide.doc
Date	2010. 1. 5.
Satus	Working

Revision History

Date	Version	Update Descriptions	Editor
2010. 1. 5.	V1.0	First Edition	Jongill Wee

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>

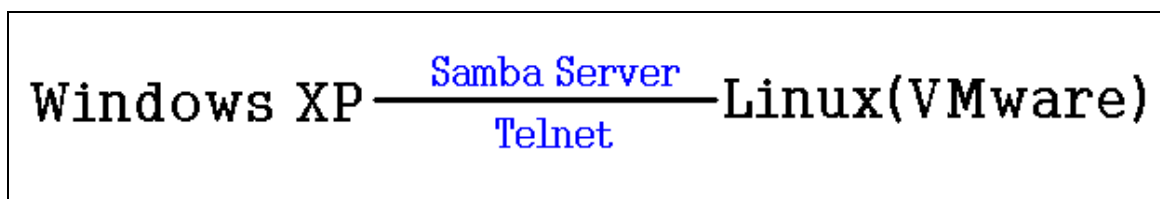
Room #610, Hanshin IT Tower 235, Guro3-dong, Guro-gu, Seoul, Korea.

Contents

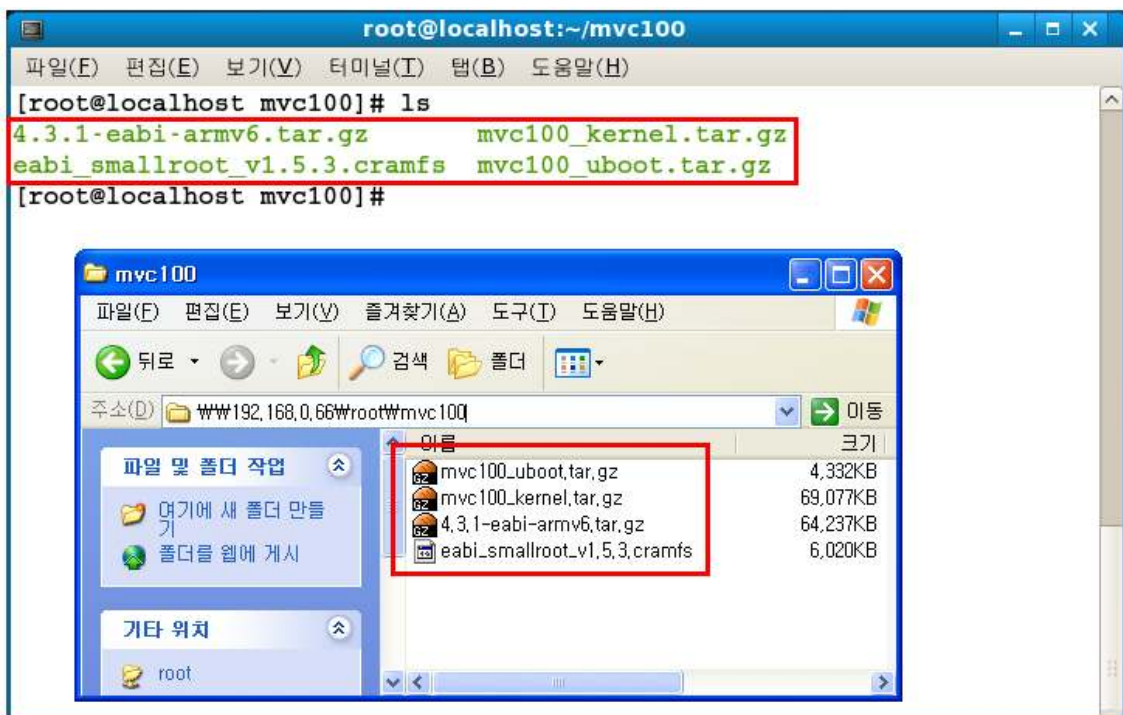
1. Environment 4/41
2. To setup GCC 5/41
3. To setup Bootloader 8/41
 - 3.1. To compile 8/41
4. To setup Kernel 10/41
 - 4.1. To compile 10/41
5. The way to modify File System 13/41
6. Download 15/41
7. To mount 27/41
 - 7.1. USB OTG mount 27/41
 - 7.2. USB Stick mount 31/41
 - 7.3. SD CARD mount 31/41
8. To setup USB Gadget 32/41

1. Environment

File	Name	Version
mvcl00_uboot.tar.gz	Bootloader	1.3.4
mvcl00_kernel.tar.gz	Kernel	2.6.28
eabi_smallroot_v1.5.3.cramfs	File System (cramfs)	1.5.3
4.3.1-eabi-armv6.tar.gz	GCC Compiler	4.3.1



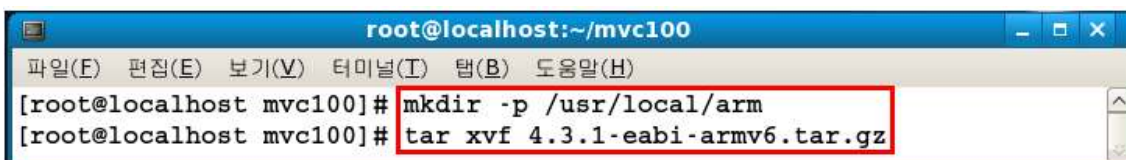
Copy Linux package to Linux server through the Samba Server.



2. To setup GCC

Unzip

```
# mkdir -p /usr/local/arm
# tar jxvf 4.3.1-eabi-armv6.tar.bz2
```



```
root@localhost:~/mvc100
파일(F) 편집(E) 보기(V) 터미널(T) 탭(B) 도움말(H)
[root@localhost mvc100]# mkdir -p /usr/local/arm
[root@localhost mvc100]# tar xvf 4.3.1-eabi-armv6.tar.gz
```

Move “4.3.1-eabi-armv6” to in /usr/local/arm

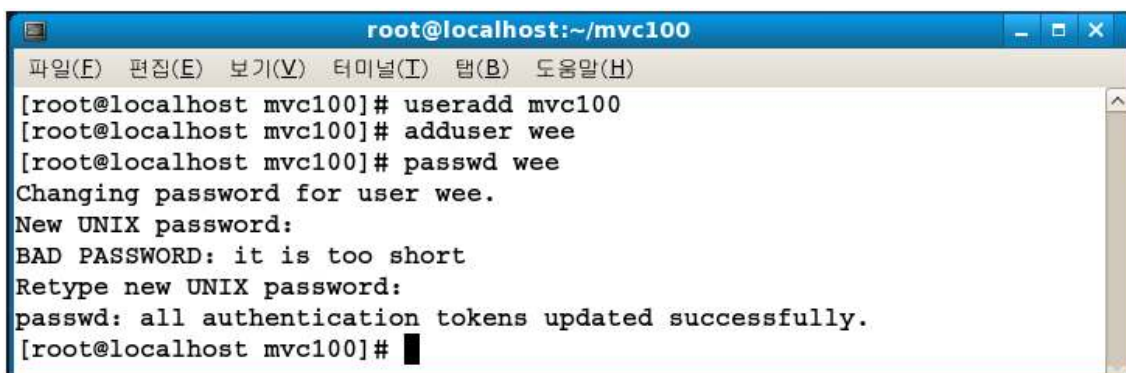
```
# mv 4.3.1-eabi-armv6/usr/local/arm/
```



```
root@localhost:~/mvc100
파일(F) 편집(E) 보기(V) 터미널(T) 탭(B) 도움말(H)
4.3.1-eabi-armv6/sbin/sln
4.3.1-eabi-armv6/sbin/ldconfig
[root@localhost mvc100]# mv 4.3.1-eabi-armv6 /usr/local/arm/
```


Make new user account not to be mistake GCC PATH in Root.

```
# useradd mvc100
# adduser wee
# passwd wee
```



```
root@localhost:~/mvc100
파일(F) 편집(E) 보기(V) 터미널(T) 탭(B) 도움말(H)
[root@localhost mvc100]# useradd mvc100
[root@localhost mvc100]# adduser wee
[root@localhost mvc100]# passwd wee
Changing password for user wee.
New UNIX password:
BAD PASSWORD: it is too short
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
[root@localhost mvc100]#
```

After made new account, copy package to your user account and then Login.



```

wee@localhost:~
파일(F) 편집(E) 보기(V) 터미널(T) 탭(B) 도움말(H)
[root@localhost mvc100]# ls
4.3.1-eabi-armv6.tar.gz          mvc100_kernel.tar.gz
eabi_smallroot_v1.5.3.cramfs    mvc100_uboot.tar.gz
[root@localhost mvc100]# cp mvc100_kernel.tar.gz /home/wee/
[root@localhost mvc100]# cp mvc100_uboot.tar.gz /home/wee/
[root@localhost mvc100]# cp eabi_smallroot_v1.5.3.cramfs /home/wee/
[root@localhost mvc100]# su -l wee
[wee@localhost ~]$ ls
eabi_smallroot_v1.5.3.cramfs  mvc100_uboot.tar.gz
mvc100_kernel.tar.gz
[wee@localhost ~]$

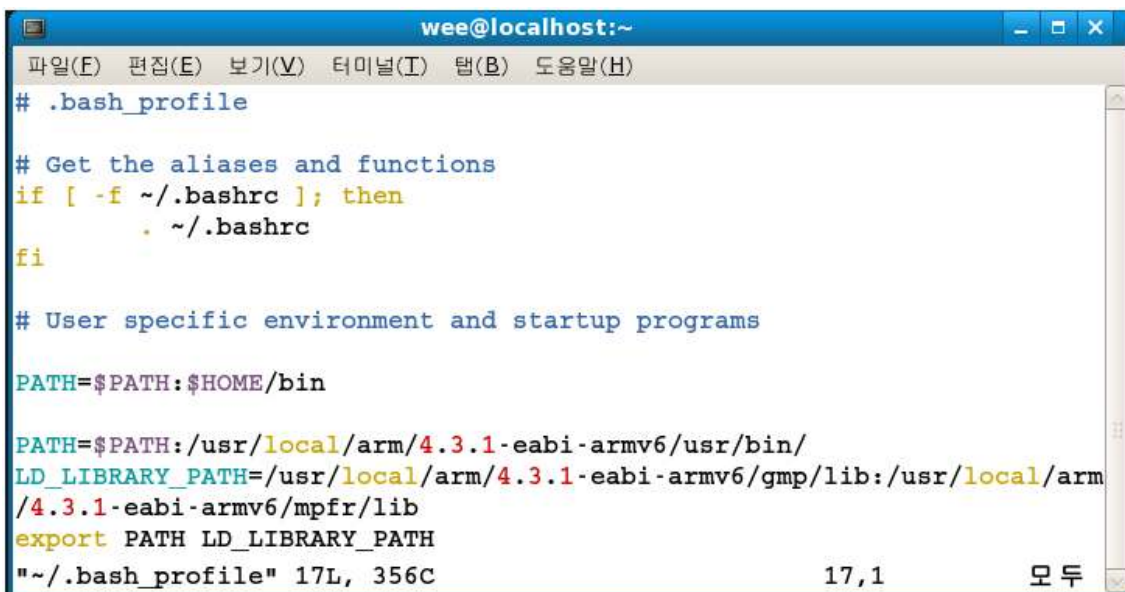
```

\$ vi ~/.bash_profile (Setup GCC PATH.)

PATH=\$PATH:\$HOME/bin:/usr/local/arm/4.3.1-eabi-armv6/usr/bin/

LD_LIBRARY_PATH=/usr/local/arm/4.3.1-eabi-armv6/gmp/lib:/usr/local/arm/4.3.1-eabi-armv6/mpfr/lib

export PATH LD_LIBRARY_PATH



```

wee@localhost:~
파일(F) 편집(E) 보기(V) 터미널(T) 탭(B) 도움말(H)
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
    . ~/.bashrc
fi

# User specific environment and startup programs

PATH=$PATH:$HOME/bin

PATH=$PATH:/usr/local/arm/4.3.1-eabi-armv6/usr/bin/
LD_LIBRARY_PATH=/usr/local/arm/4.3.1-eabi-armv6/gmp/lib:/usr/local/arm/
/4.3.1-eabi-armv6/mpfr/lib
export PATH LD_LIBRARY_PATH
"~/ .bash_profile" 17L, 356C          17,1          모두

```

After saved and then apply environment.

\$ source ~/.bash_profile



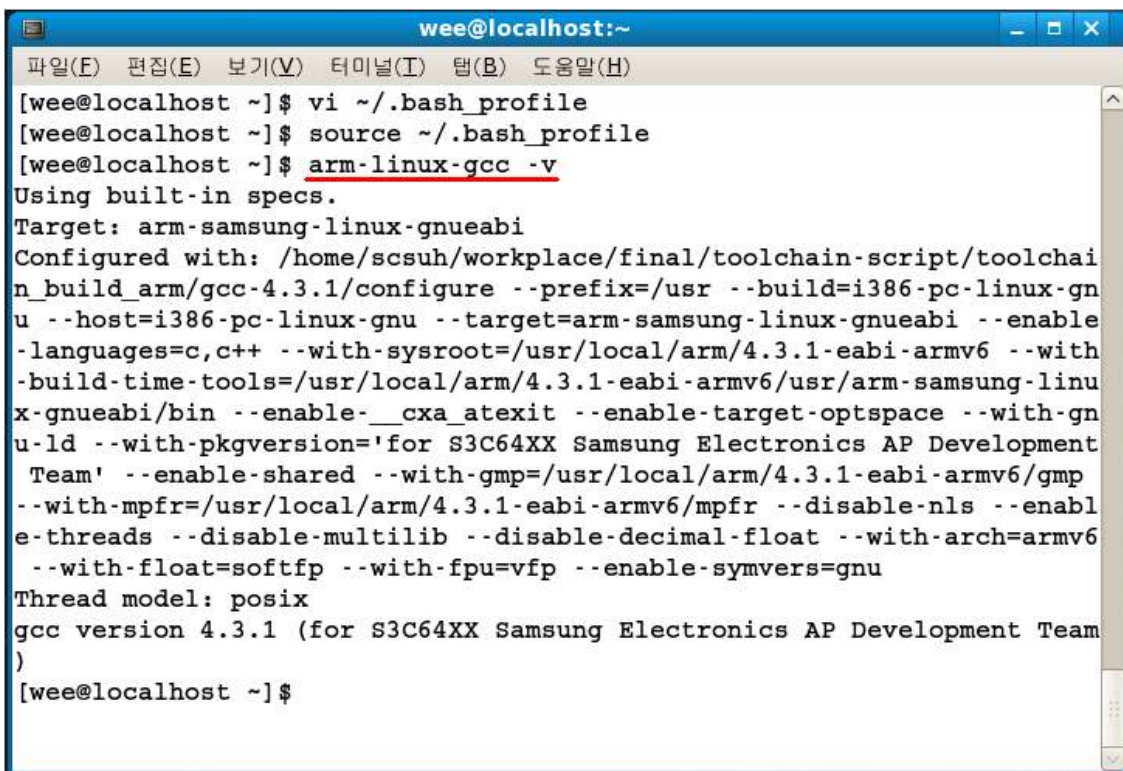
```

wee@localhost:~
파일(F) 편집(E) 보기(V) 터미널(T) 탭(B) 도움말(H)
[wee@localhost ~]$ vi ~/.bash_profile
[wee@localhost ~]$ source ~/.bash_profile
[wee@localhost ~]$

```

You can see GCC version as a command.

```
$ arm-linux-gcc -v
```



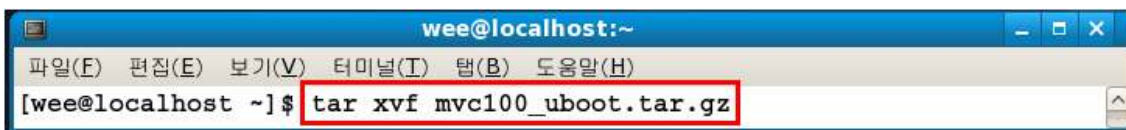
```
wee@localhost:~
파일(F) 편집(E) 보기(V) 터미널(T) 탭(B) 도움말(H)
[wee@localhost ~]$ vi ~/.bash_profile
[wee@localhost ~]$ source ~/.bash_profile
[wee@localhost ~]$ arm-linux-gcc -v
Using built-in specs.
Target: arm-samsung-linux-gnueabi
Configured with: /home/scsuh/workplace/final/toolchain-script/toolchain_build_arm/gcc-4.3.1/configure --prefix=/usr --build=i386-pc-linux-gnu --host=i386-pc-linux-gnu --target=arm-samsung-linux-gnueabi --enable-languages=c,c++ --with-sysroot=/usr/local/arm/4.3.1-eabi-armv6 --with-build-time-tools=/usr/local/arm/4.3.1-eabi-armv6/usr/arm-samsung-linux-gnueabi/bin --enable-__cxa_atexit --enable-target-optspace --with-gnu-ld --with-pkgversion='for S3C64XX Samsung Electronics AP Development Team' --enable-shared --with-gmp=/usr/local/arm/4.3.1-eabi-armv6/gmp --with-mpfr=/usr/local/arm/4.3.1-eabi-armv6/mpfr --disable-nls --enable-threads --disable-multilib --disable-decimal-float --with-arch=armv6 --with-float=softfp --with-fpu=vfp --enable-symvers=gnu
Thread model: posix
gcc version 4.3.1 (for S3C64XX Samsung Electronics AP Development Team)
[wee@localhost ~]$
```

3. To setup Bootloader

3.1. To compile

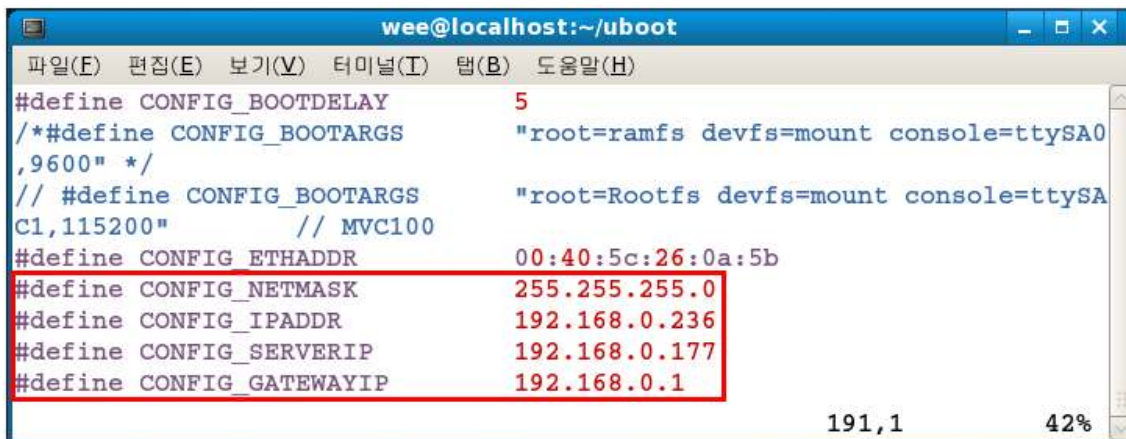
unzip

\$ tar xvf mvc100_uboot.tar.gz



To setup IP address

\$ vi include/configs/smdkc100.h



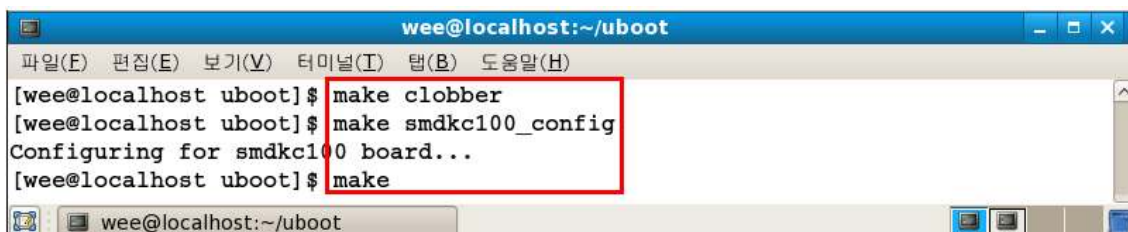
#define CONFIG_ETHADDR MAC ADDRESS
#define CONFIG_NETMASK SUBNETMASK
#define CONFIG_IPADDR Target Board
#define CONFIG_SERVERIP TFTP Server PC

The compile is order.

```
$ make clobber
```

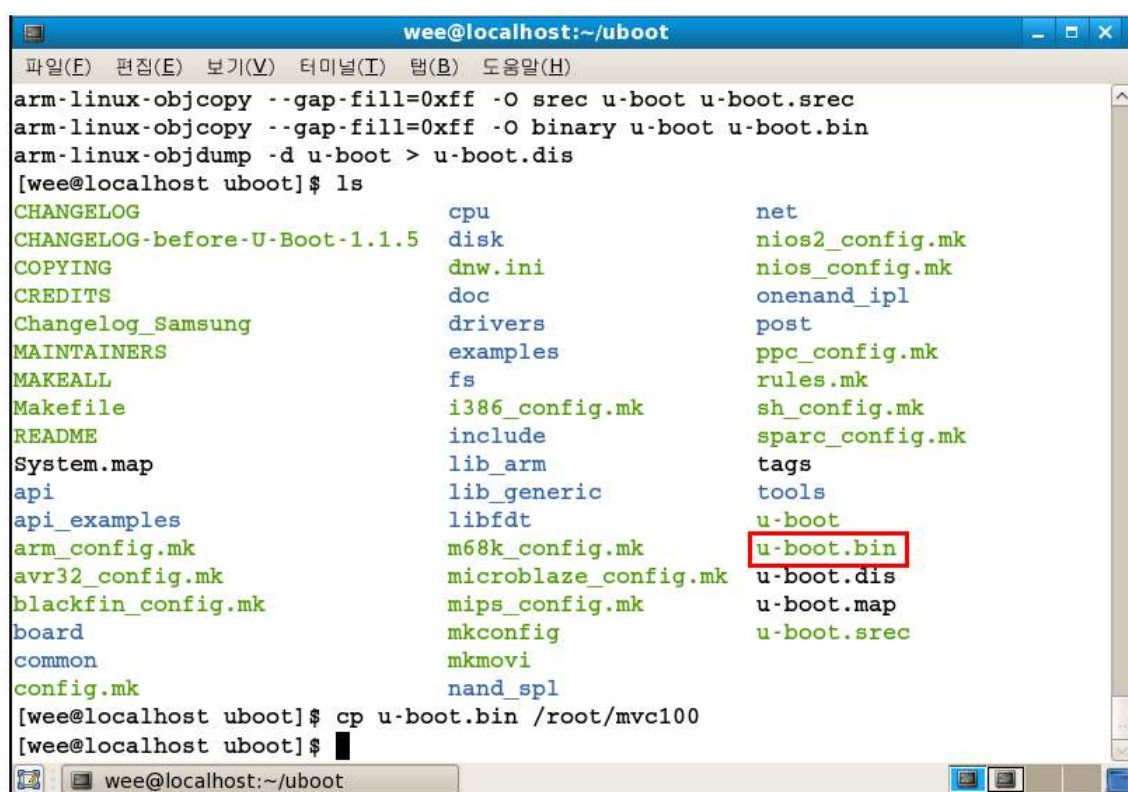
```
$ make smdkc100_config
```

```
$ make
```



```
wee@localhost:~/uboot
[wee@localhost uboot]$ make clobber
[wee@localhost uboot]$ make smdkc100_config
Configuring for smdkc100 board...
[wee@localhost uboot]$ make
```

You can see “u-boot.bin” in /uboot.



```
wee@localhost:~/uboot
arm-linux-objcopy --gap-fill=0xff -O srec u-boot u-boot.srec
arm-linux-objcopy --gap-fill=0xff -O binary u-boot u-boot.bin
arm-linux-objdump -d u-boot > u-boot.dis
[wee@localhost uboot]$ ls
CHANGELOG          cpu                net
CHANGELOG-before-U-Boot-1.1.5  disk              nios2_config.mk
COPYING            dnw.ini           nios_config.mk
CREDITS            doc               onenand_ipl
Changelog_Samsung drivers           post
MAINTAINERS        examples          ppc_config.mk
MAKEALL            fs                rules.mk
Makefile           i386_config.mk   sh_config.mk
README             include           sparc_config.mk
System.map         lib_arm           tags
api                lib_generic       tools
api_examples       libfdt            u-boot
arm_config.mk      m68k_config.mk   u-boot.bin
avr32_config.mk    microblaze_config.mk u-boot.dis
blackfin_config.mk mips_config.mk   u-boot.map
board              mkconfig          u-boot.srec
common             mkmovi
config.mk          nand_spl
```

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

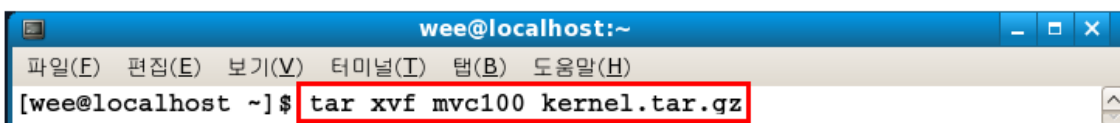
```
$ cp u-boot /tftpboot
```

4. To setup Kernel

4.1. To compile

Unzip.

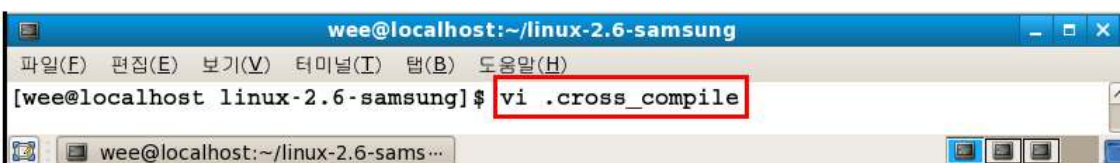
```
$ tar jxvf mvcl00_kernel.tar.gz
```



```
wee@localhost:~  
파일(F) 편집(E) 보기(V) 터미널(T) 탭(B) 도움말(H)  
[wee@localhost ~]$ tar xvf mvcl00 kernel.tar.gz
```

As under picture, Run vi editor “.cross_compile” and then modify GCC PATH.

```
$ vi .cross_compile
```



```
wee@localhost:~/linux-2.6-samsung  
파일(F) 편집(E) 보기(V) 터미널(T) 탭(B) 도움말(H)  
[wee@localhost linux-2.6-samsung]$ vi .cross_compile
```

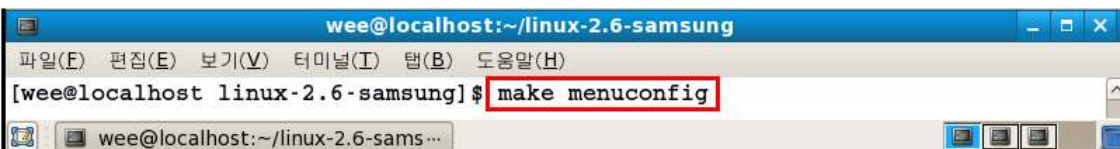
Save after make sure.



```
wee@localhost:~/linux-2.6-samsung  
파일(F) 편집(E) 보기(V) 터미널(T) 탭(B) 도움말(H)  
usr/local/arm/4.3.1-eabi-armv6/usr/bin/arm-linux-  
".cross_compile" 1L, 51C 1,1 모두
```

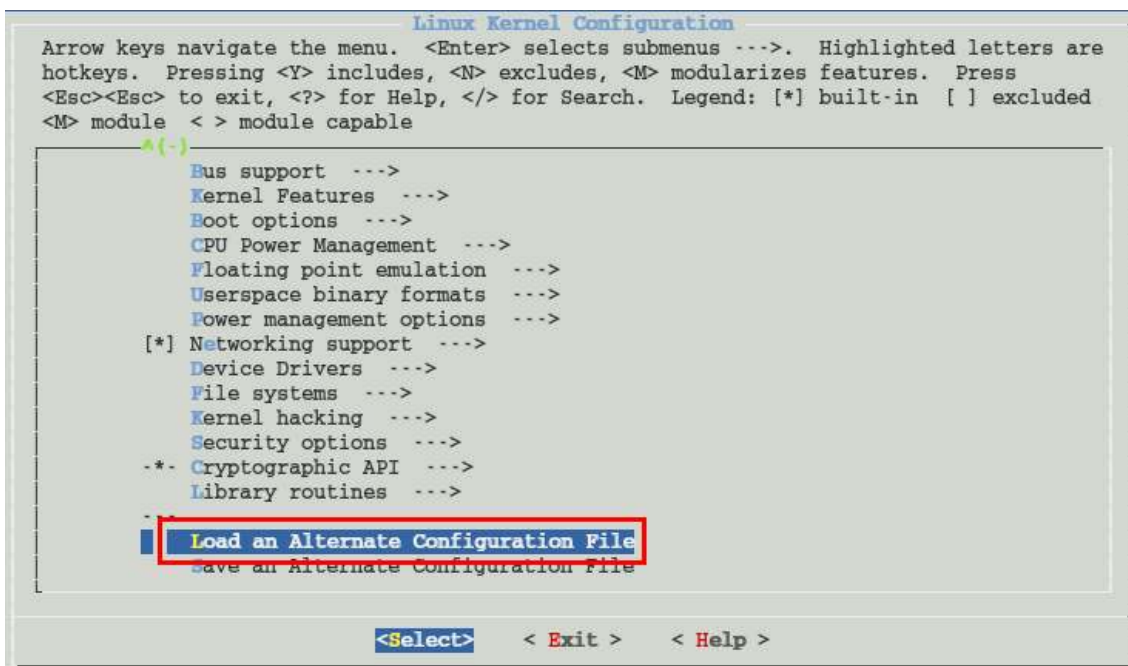
Run a command for compiling.

```
$ make menuconfig
```

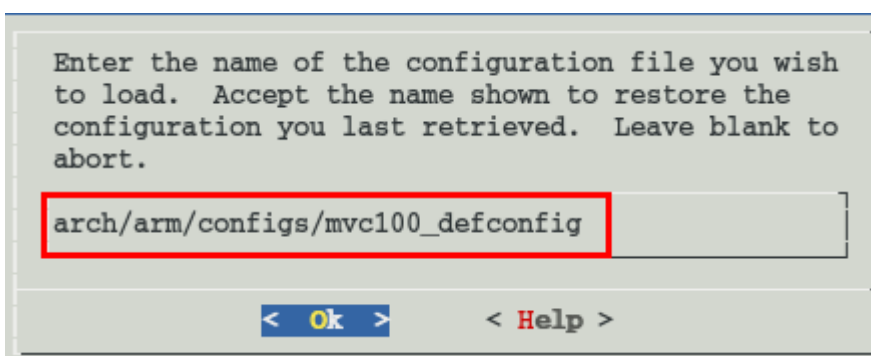


```
wee@localhost:~/linux-2.6-samsung  
파일(F) 편집(E) 보기(V) 터미널(T) 탭(B) 도움말(H)  
[wee@localhost linux-2.6-samsung]$ make menuconfig
```

Select "Load an Alternate Configuration File".

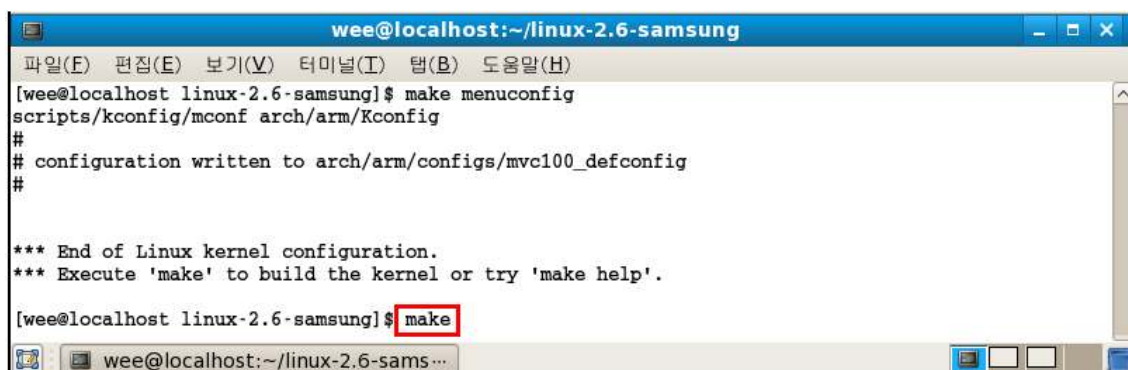


Load "arch/arm/configs/mvc100_defconfig".

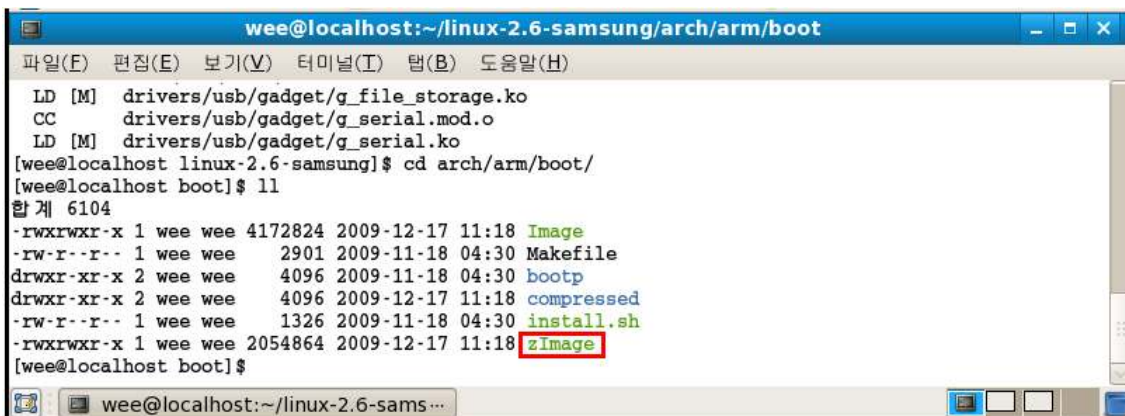


After make sure and then exit.

\$ make



You can see “zImage” in kernal/arch/arm/boot.
After compiled, Copy “zImage” to “/tftpboot”



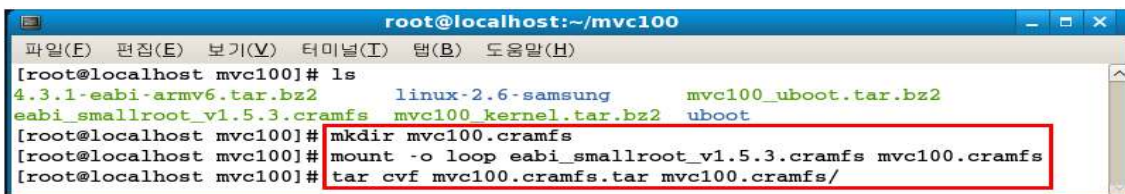
\$ cp zImage /tftpboot

5. The way to modify File System

After make something folder, Mount File System and then Zip that folder was Mounted by File System of image and then unmount.

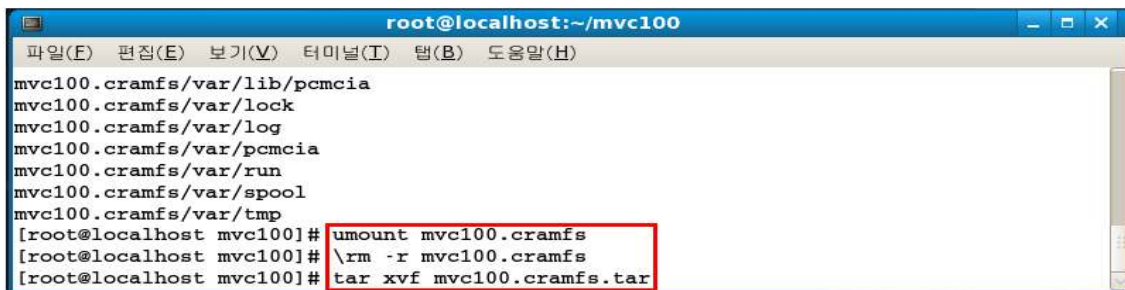
As that order.

```
# mkdir mvc100.cramfs
# mount -o loop eabi_smallroot_v1.5.3.cramfs mvc100.cramfs
# tar cvf mvc100.cramfs.tar mvc100.cramfs
```



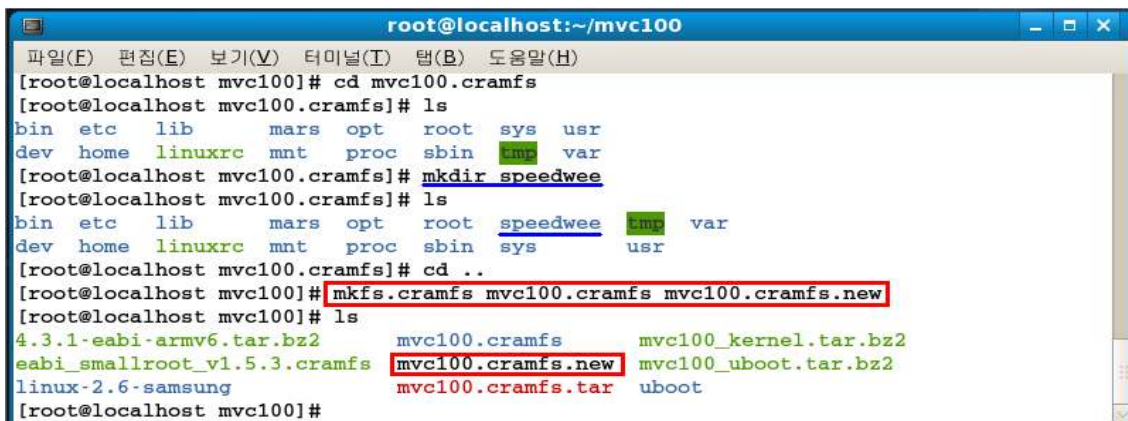
```
root@localhost:~/mvc100
파일(E) 편집(E) 보기(V) 터미널(T) 탭(B) 도움말(H)
[root@localhost mvc100]# ls
4.3.1-eabi-armv6.tar.bz2      linux-2.6-samsung      mvc100_uboot.tar.bz2
eabi_smallroot_v1.5.3.cramfs  mvc100_kernel.tar.bz2  uboot
[root@localhost mvc100]# mkdir mvc100.cramfs
[root@localhost mvc100]# mount -o loop eabi_smallroot_v1.5.3.cramfs mvc100.cramfs
[root@localhost mvc100]# tar cvf mvc100.cramfs.tar mvc100.cramfs/
```

```
# umount mvc100.cramfs
# Wrm -r mvc100.cramfs
# tar xvf mvc100.cramfs.tar
```



```
root@localhost:~/mvc100
파일(E) 편집(E) 보기(V) 터미널(T) 탭(B) 도움말(H)
mvc100.cramfs/var/lib/pcmcia
mvc100.cramfs/var/lock
mvc100.cramfs/var/log
mvc100.cramfs/var/pcmcia
mvc100.cramfs/var/run
mvc100.cramfs/var/spool
mvc100.cramfs/var/tmp
[root@localhost mvc100]# umount mvc100.cramfs
[root@localhost mvc100]# \rm -r mvc100.cramfs
[root@localhost mvc100]# tar xvf mvc100.cramfs.tar
```

```
# mkfs.cramfs mvc100.cramfs mvc100.cramfs.new
```

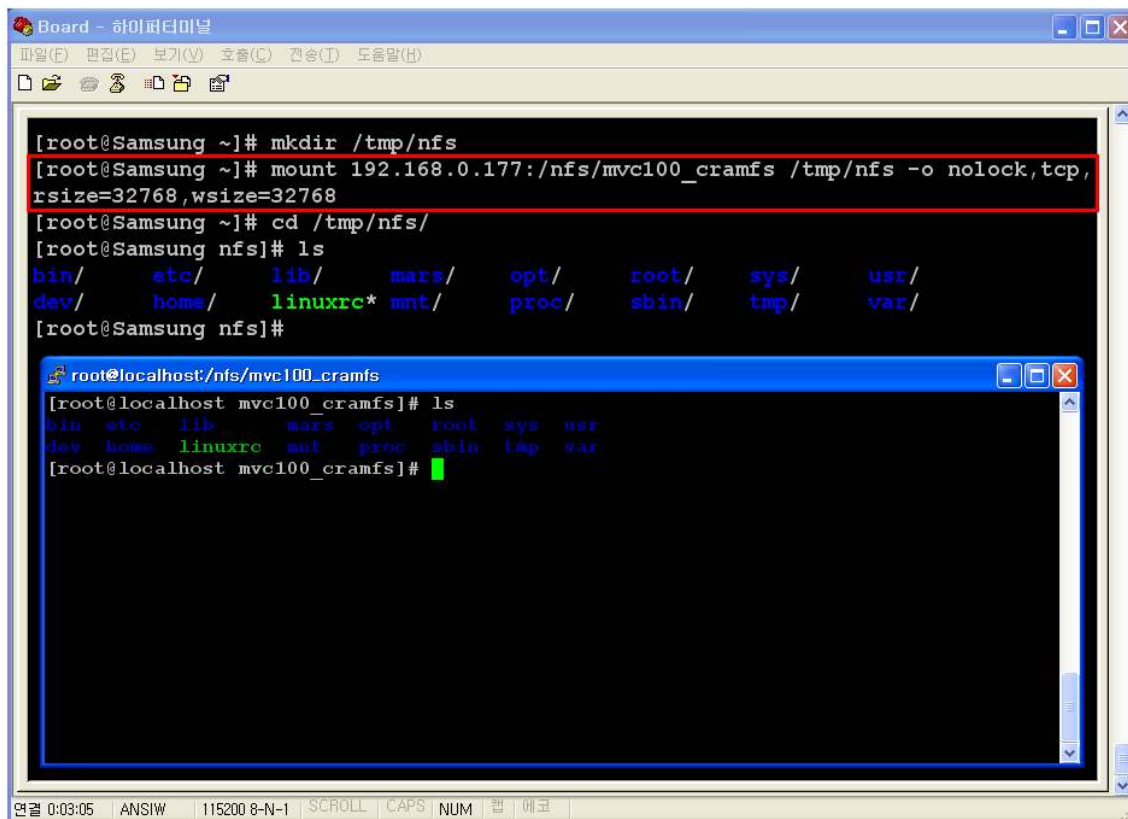


CF: mkfs.cramfs is fedora of command.

CF: The way to mount NFS form linux PC to Board.

mkdir /tmp/nfs

mount 192.168.0.177:/nfs/mvc100_cramfs /tmp/nfs -o nolock,tcp,rsize=32768,wsiz=32768

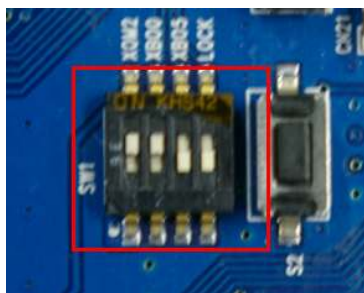


-It is mount-

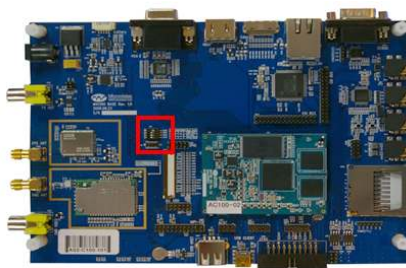
6. Download

As I introduced in the document MVC100_Operating.pdf, S5PC100 has in the CPU 32KB ROM and 96KB SRAM so that can program without JTAG emulator by itself.

Please set up as under picture Boot mode. This is Boot mode which can download to mDDR through USB Cable

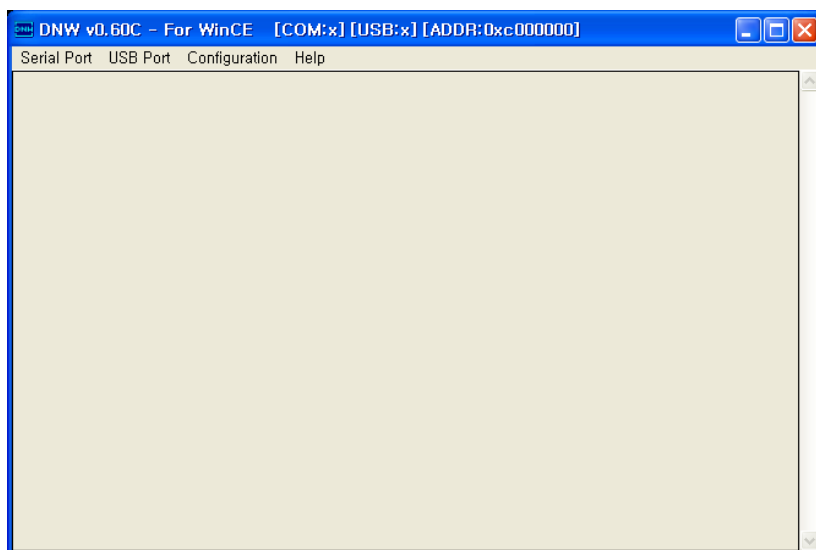


USB Boot mode



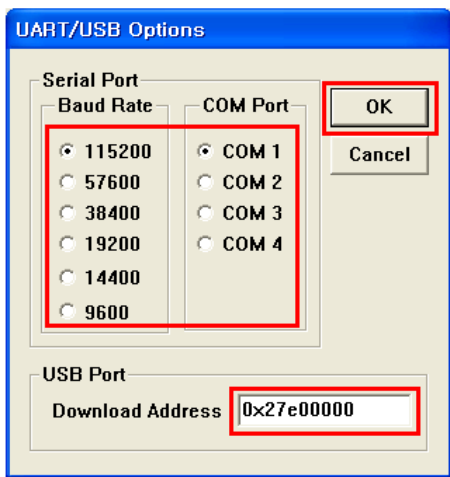
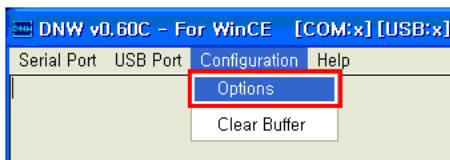
Back Switch position

Run DNW.exe in WTool\DNW v0.60C

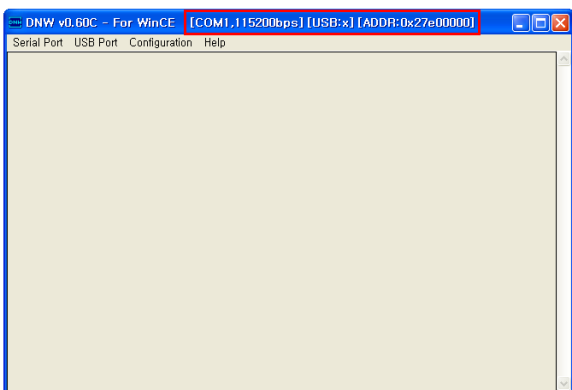
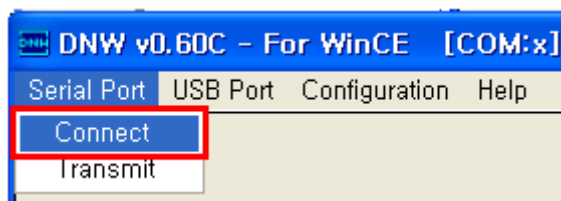


DNW

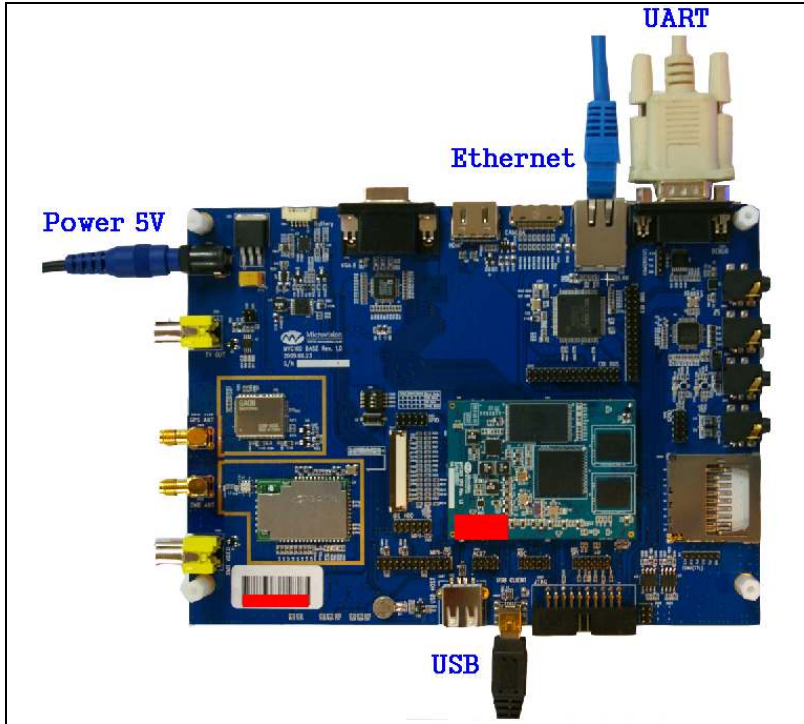
On the Configuration menu, click Options to set the UART/USB options. The following window appears on your screen. Select Baud Rate and COM Port as shown in figure "UART/USB options", enter the download address as 0x27e00000 and then click OK button.



On the Serial Port menu, click Connect. Switch ON the reference board and then press any key and then install the USB driver in WSRWCWTool\DNW v0.60C driver directory.



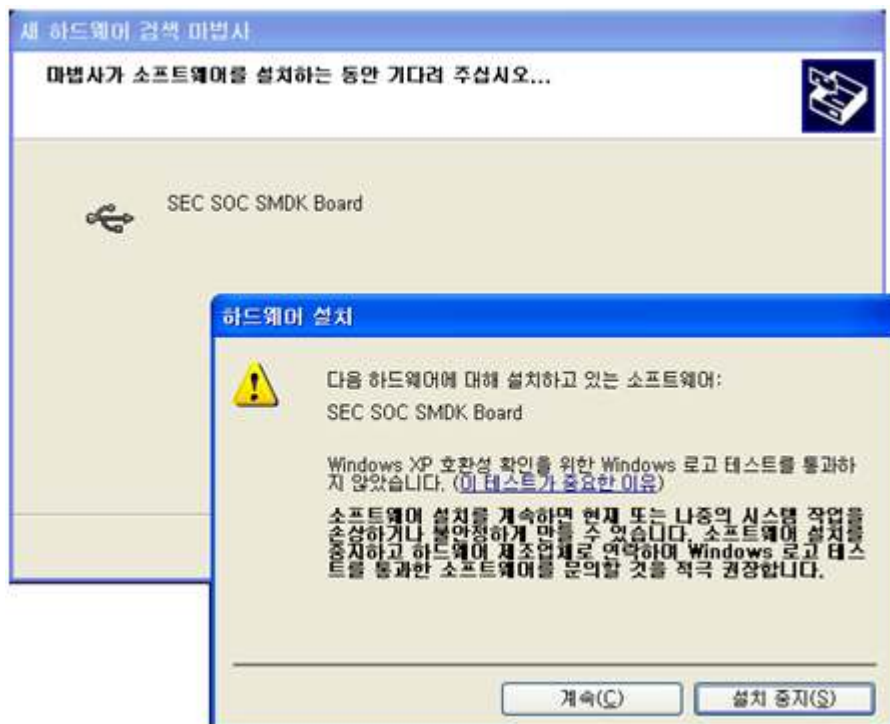
Connect Power 5V with Board and then USB 2.0 OTG Device with your Host PC to download through USB and also RS232 for monitoring.



As soon as connect power, PC is aware of USB Driver



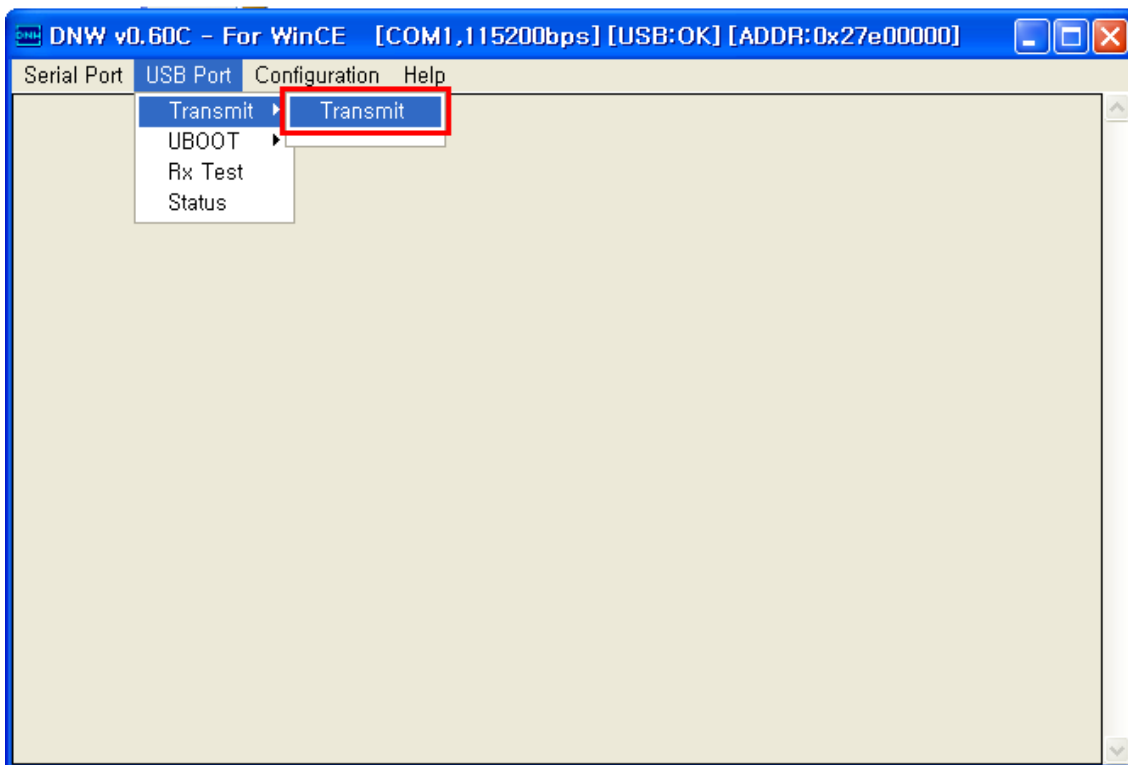
Installing in CD W\TOOL\WSMDKC100 USB Driver USB Driver



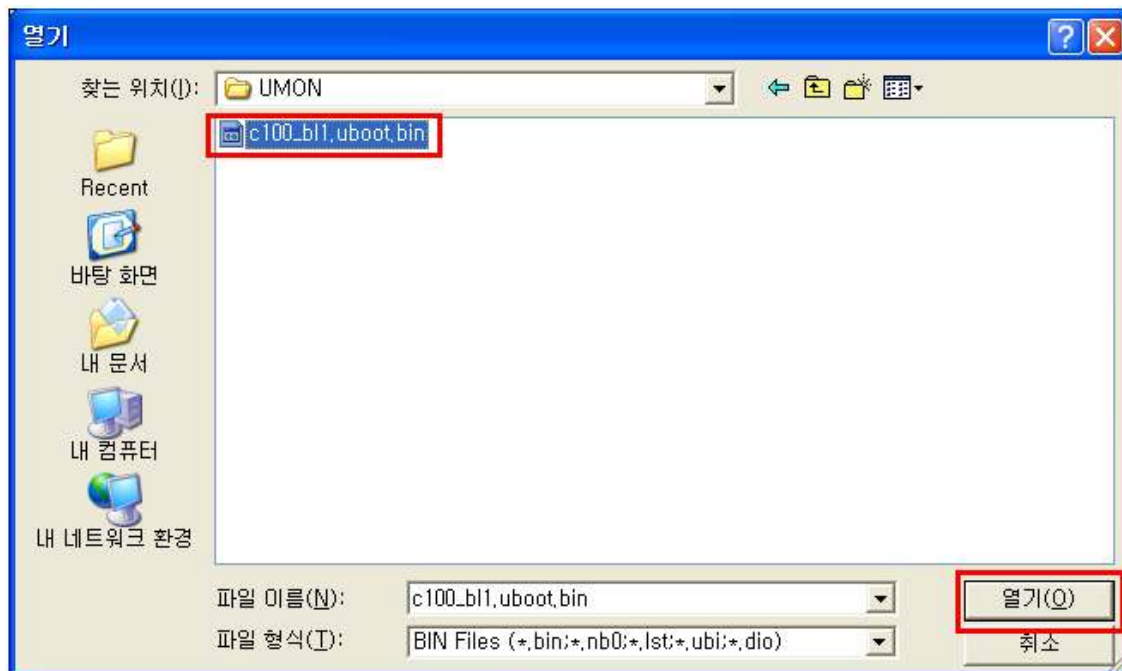
Success installed output message "USB OK"



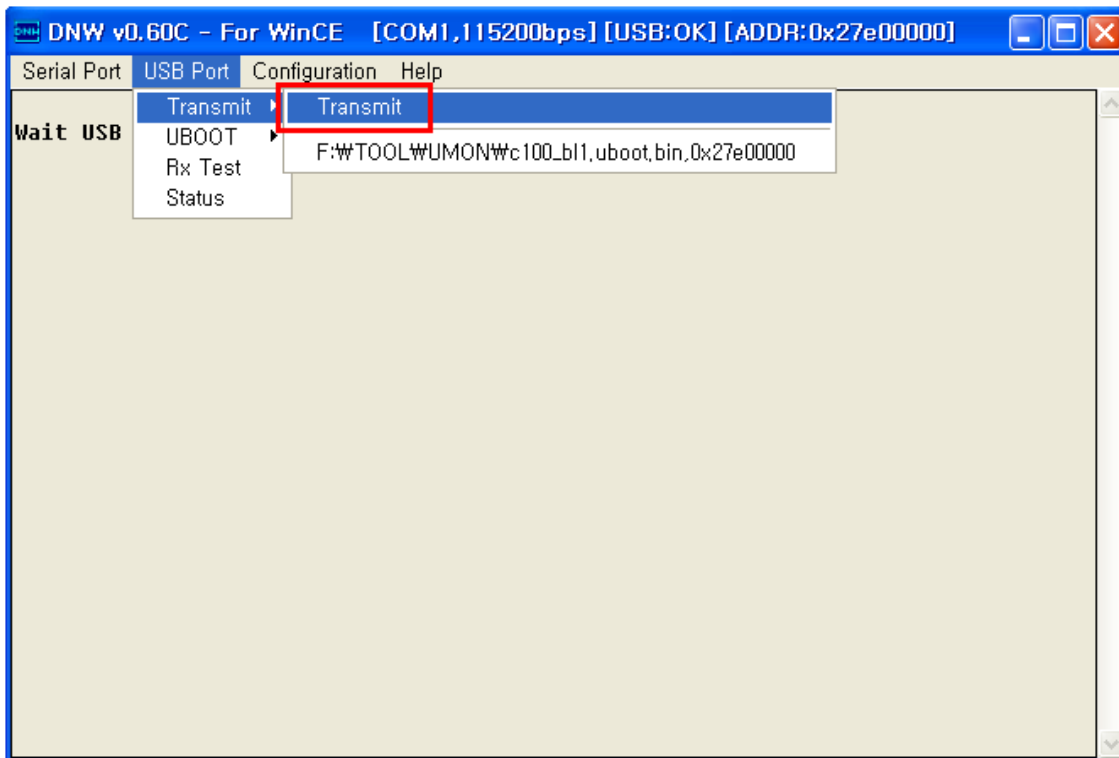
Select USB Port -> Transmit -> Transmit for loading "c100_b11.uboot.bin".



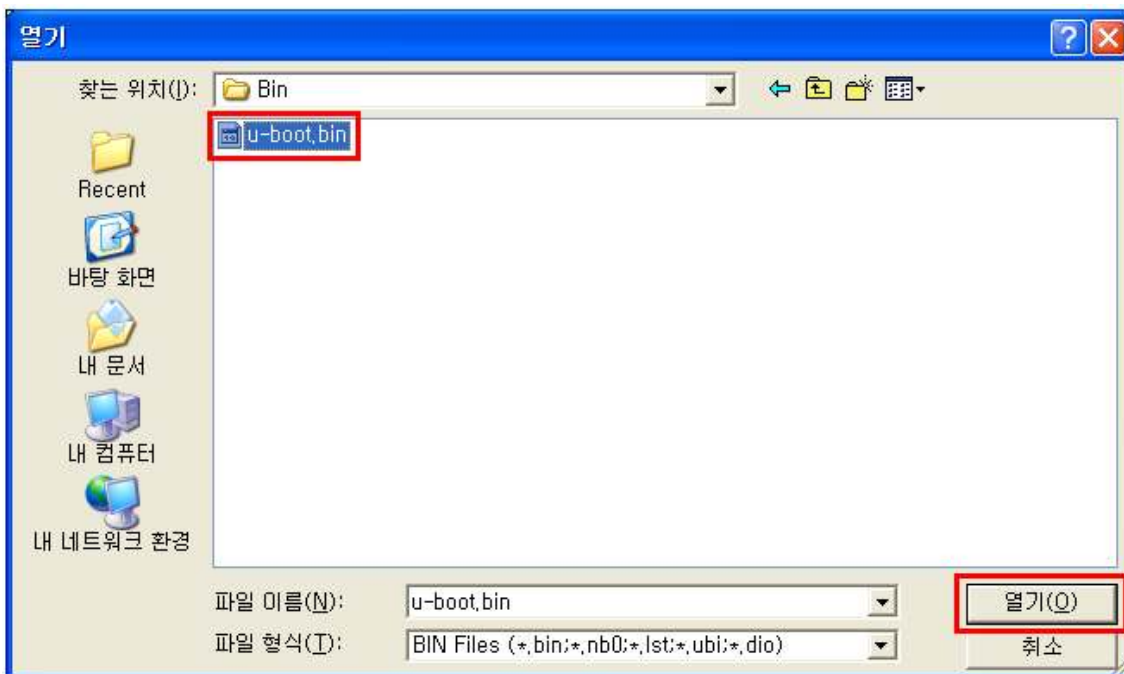
Open in CD /TOOL/UMOIN "c100_b11.uboot.bin".



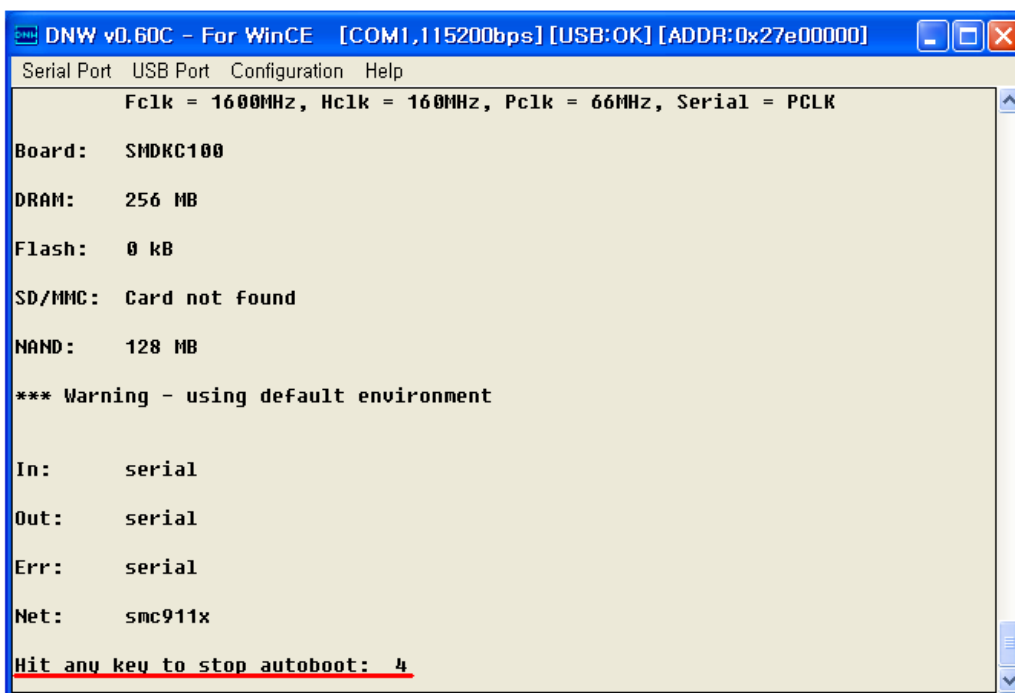
Select USB Port -> Transmit -> Transmit for running "u-boot.bin"



Open in CD SRCWBin "u-boot.bin".



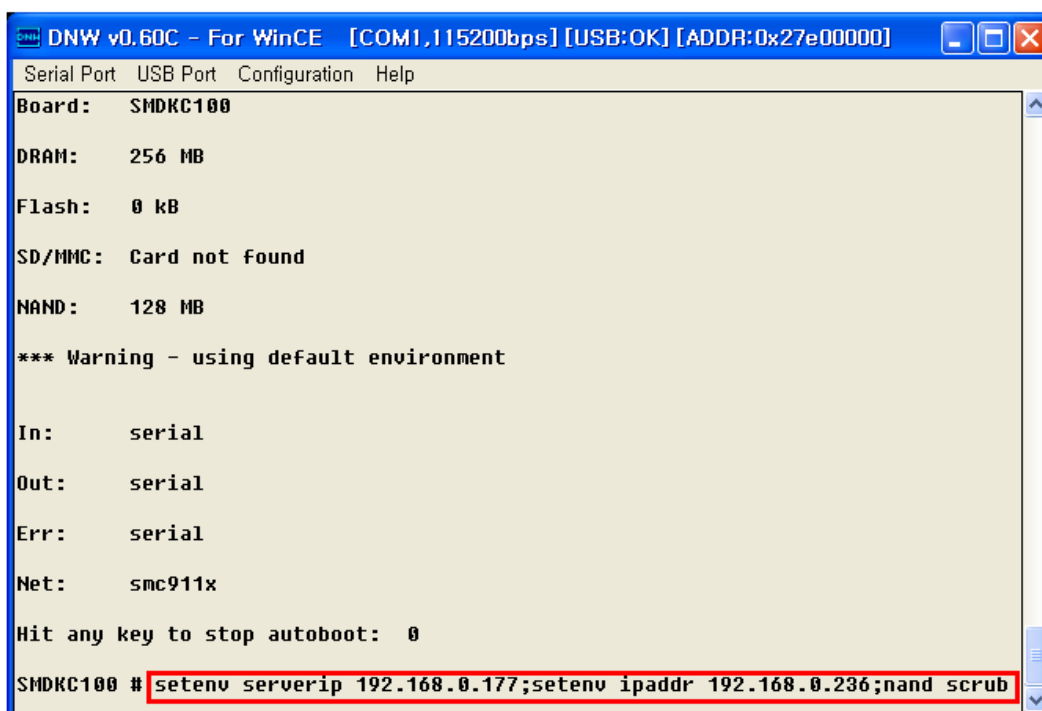
As soon as U-boot.bin download is over, the following messages appear in the DNW window. Please hit the SPACE BAR key to view.



```
DNW v0.60C - For WinCE [COM1,115200bps] [USB:OK] [ADDR:0x27e00000]
Serial Port  USB Port  Configuration  Help
      Fc1k = 1600MHz, Hc1k = 160MHz, Pc1k = 66MHz, Serial = PCLK
Board:  SMDKC100
DRAM:   256 MB
Flash:  0 kB
SD/MMC: Card not found
NAND:   128 MB
*** Warning - using default environment
In:     serial
Out:    serial
Err:    serial
Net:    smc911x
Hit any key to stop autoboot:  4
```

Fill in IP address for TFTP.

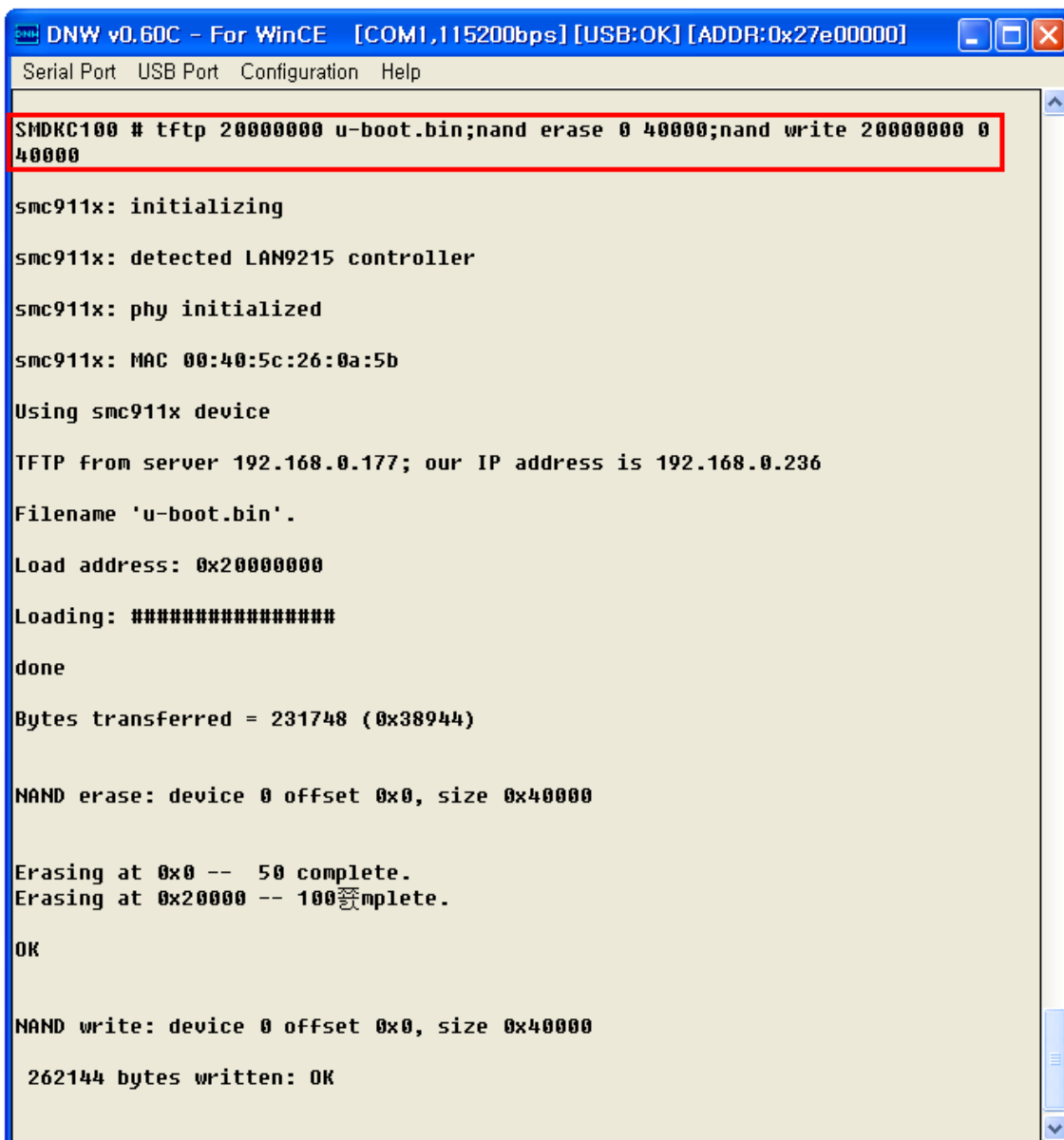
```
# setenv serverip 192.168.0.177;setenv ipaddr 192.168.0.236;nand scrub
-> enter "y"
```



```
DNW v0.60C - For WinCE [COM1,115200bps] [USB:OK] [ADDR:0x27e00000]
Serial Port  USB Port  Configuration  Help
Board:  SMDKC100
DRAM:   256 MB
Flash:  0 kB
SD/MMC: Card not found
NAND:   128 MB
*** Warning - using default environment
In:     serial
Out:    serial
Err:    serial
Net:    smc911x
Hit any key to stop autoboot:  0
SMDKC100 # setenv serverip 192.168.0.177;setenv ipaddr 192.168.0.236;nand scrub
```

Fill in commands for programming "u-boot.bin".

```
# tftp 20008000 u-boot.bin;nand erase 0 40000;nand write 20008000 0 40000
```



```
DNW v0.60C - For WinCE [COM1,115200bps] [USB:OK] [ADDR:0x27e00000]
Serial Port USB Port Configuration Help
SMDKC100 # tftp 20008000 u-boot.bin;nand erase 0 40000;nand write 20008000 0
40000

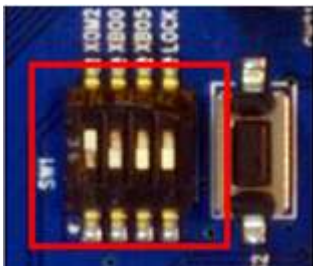
smc911x: initializing
smc911x: detected LAN9215 controller
smc911x: phy initialized
smc911x: MAC 00:40:5c:26:0a:5b
Using smc911x device
TFTP from server 192.168.0.177; our IP address is 192.168.0.236
Filename 'u-boot.bin'.
Load address: 0x20008000
Loading: #####
done
Bytes transferred = 231748 (0x38944)

NAND erase: device 0 offset 0x0, size 0x40000

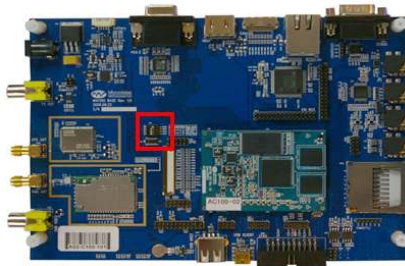
Erasing at 0x0 -- 50 complete.
Erasing at 0x20000 -- 100% complete.
OK

NAND write: device 0 offset 0x0, size 0x40000
262144 bytes written: OK
```

NAND was programmed 0x0 u-boot. Now change NAND Flash boot mode for programming kernel, file system.

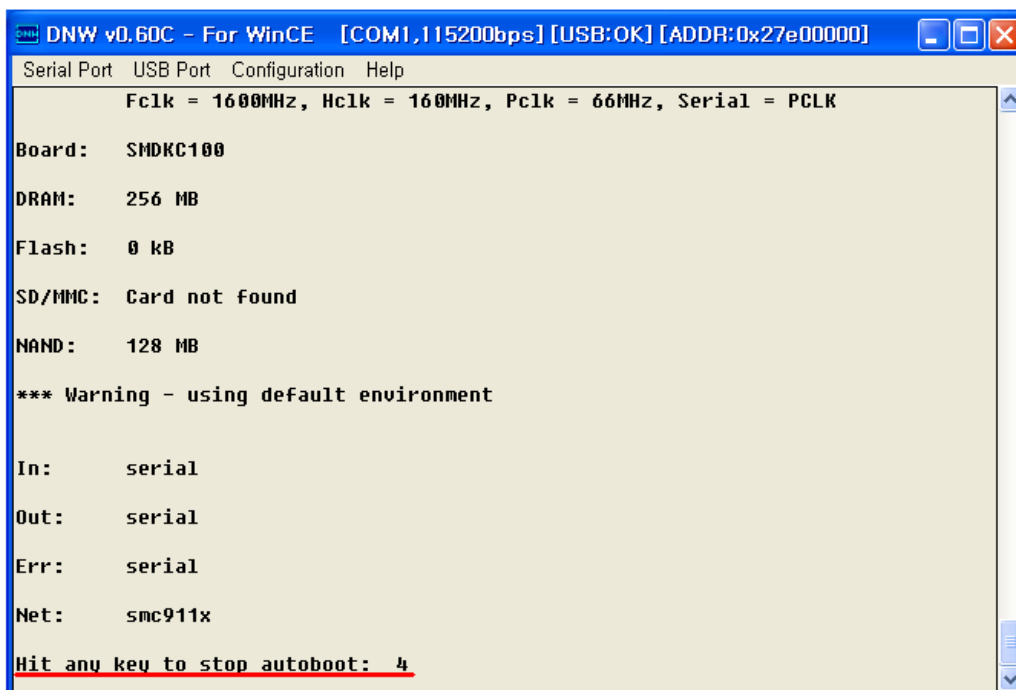


NAND boot mode

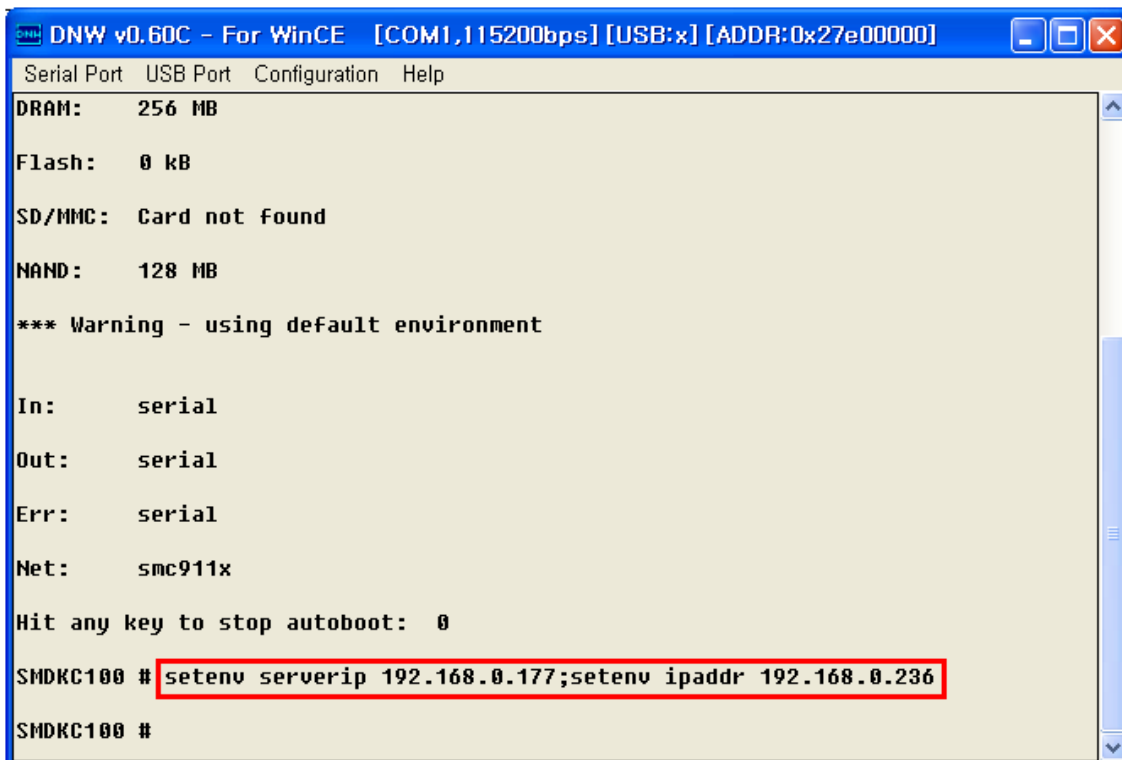


Back Reset Switch position

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.



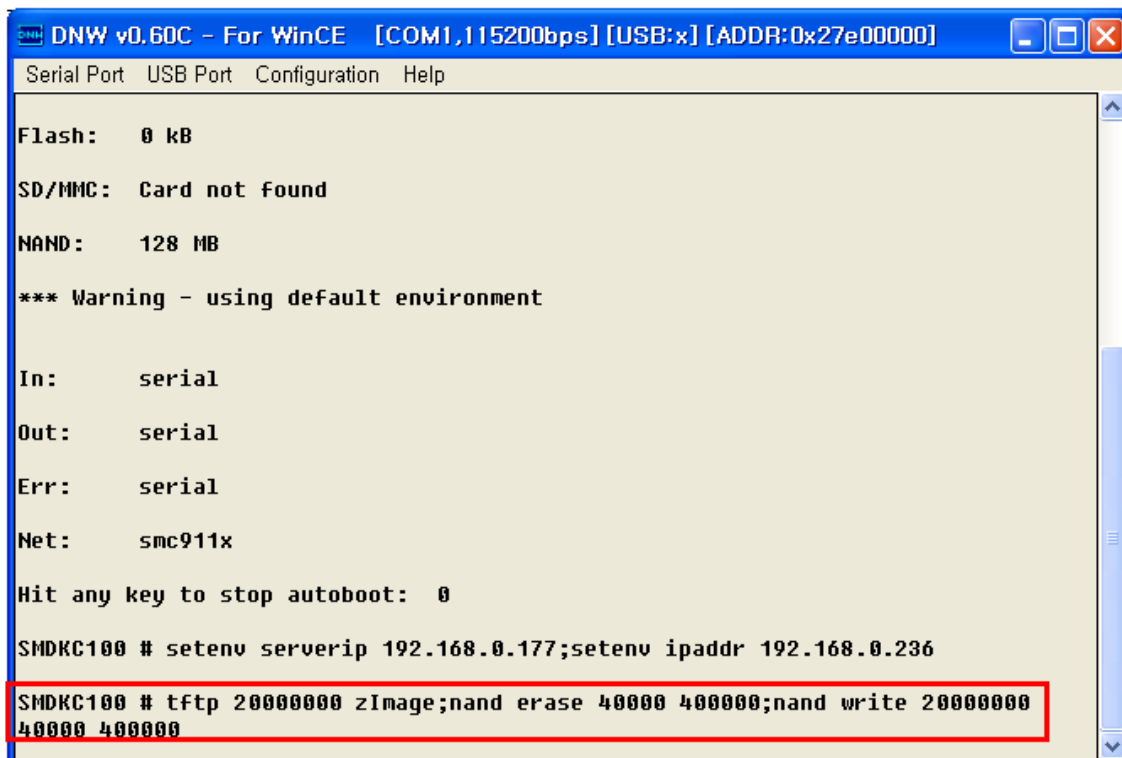
```
# setenv serverip 192.168.0.177;setenv ipaddr 192.168.0.236
```



The screenshot shows a terminal window titled "DNW v0.60C - For WinCE [COM1,115200bps] [USB:x] [ADDR:0x27e00000]". The window contains the following text:

```
Serial Port  USB Port  Configuration  Help
DRAM:      256 MB
Flash:     0 kB
SD/MMC:    Card not found
NAND:      128 MB
*** Warning - using default environment
In:        serial
Out:       serial
Err:       serial
Net:       smc911x
Hit any key to stop autoboot: 0
SMDKC100 # setenv serverip 192.168.0.177;setenv ipaddr 192.168.0.236
SMDKC100 #
```

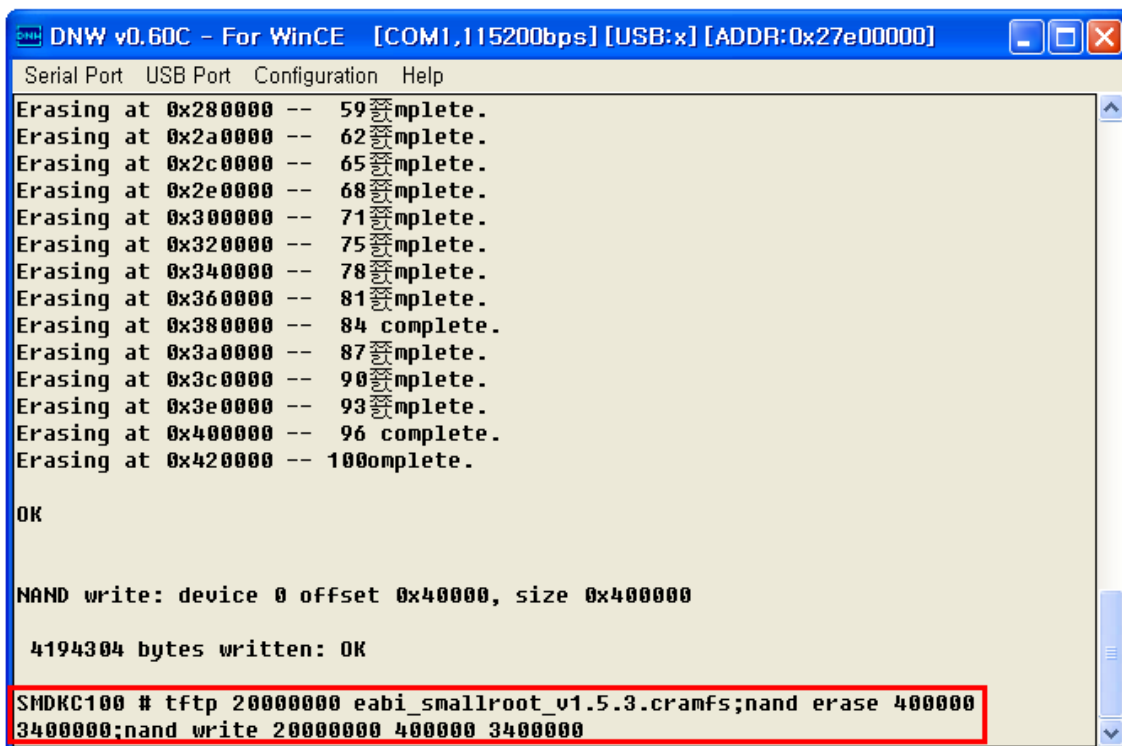
```
# tftp 20000000 zImage;nand erase 40000 400000;nand write 20000000 40000
400000
```



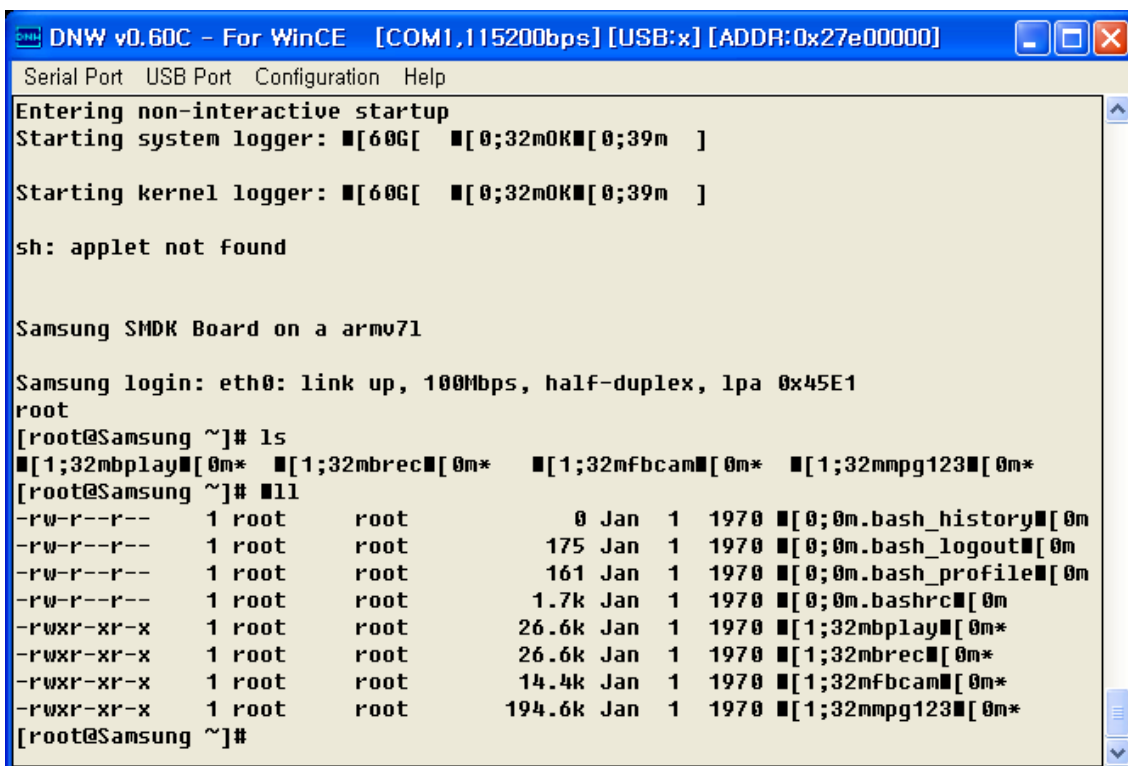
The screenshot shows the same terminal window as above, but with the following text added:

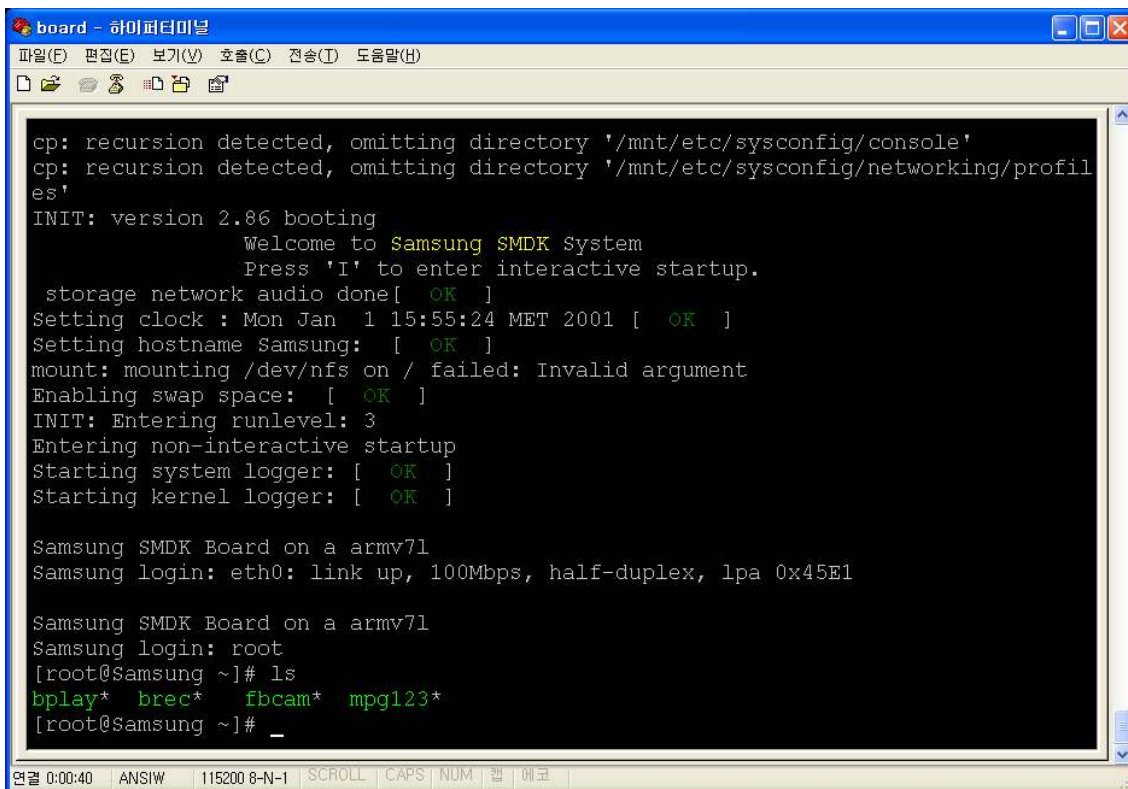
```
SMDKC100 # setenv serverip 192.168.0.177;setenv ipaddr 192.168.0.236
SMDKC100 # tftp 20000000 zImage;nand erase 40000 400000;nand write 20000000
400000 400000
```

```
tftp 2000000 eabi_smallroot_v1.5.3.cramfs;nand erase 40000 340000;nand
write 2000000 40000 340000
```



DNW can't copy text so I recommended "HyperTerminal"



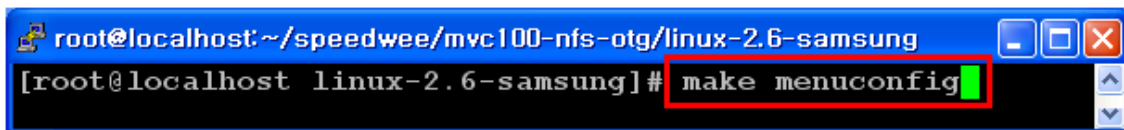


This is done Linux Boot.

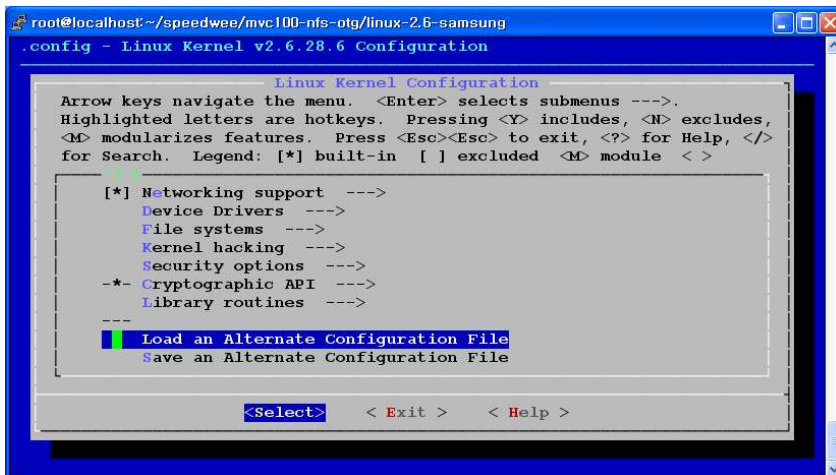
7. To mount

7.1. USB OTG mount

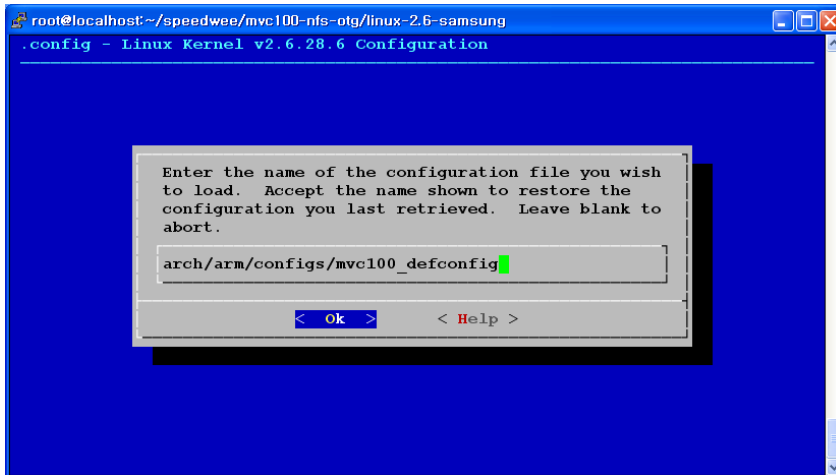
Run “make menuconfig” in Kernel for using USB OTG.



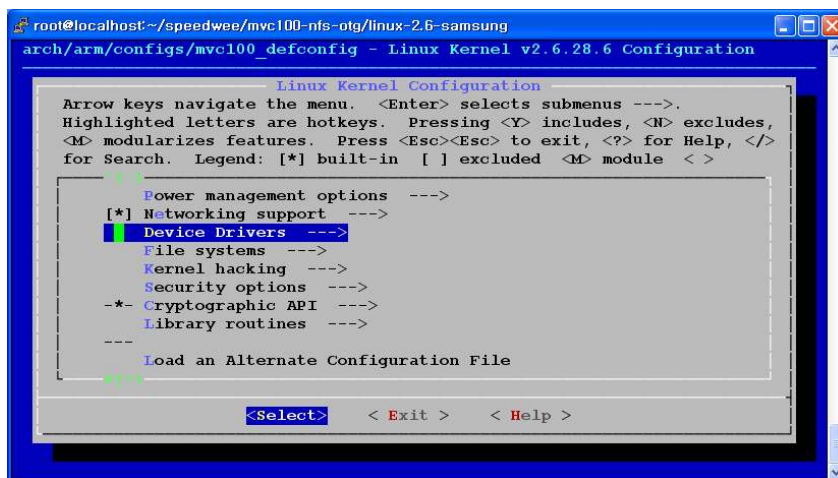
Select “Load an Alternate Configuration File”.



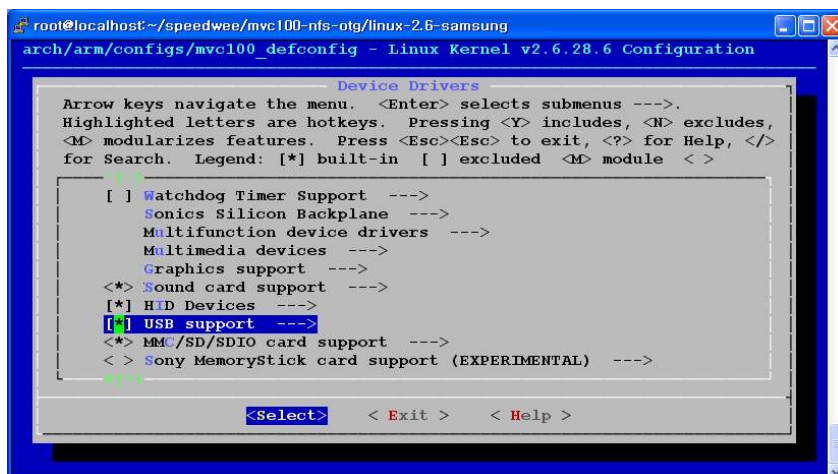
Load “arch/arm/configs/mvc100_defconfig”.



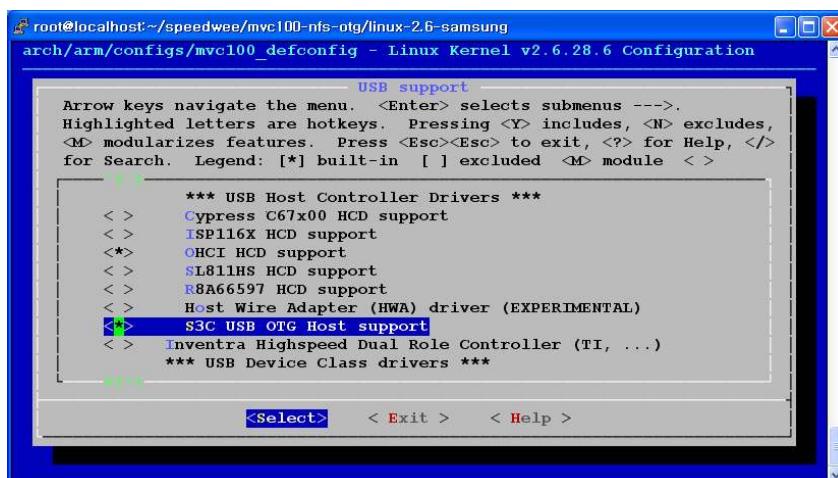
Select Device Drivers ---->



Select [*] USB support ---->



Check "*" "<*> S3C USB OTG Host support" and then EXIT.



Run "make" for compiling Kernel and then program new zImage in "/linux-2.6-samsung/arch/arm/boot".

```
Board - 하이퍼터미널
파일(F) 편집(E) 보기(V) 호출(C) 전송(T) 도움말(H)
SMDKC100 # setenv serverip 192.168.0.177;setenv ipaddr 192.168.0.236;
SMDKC100 # tftp 20000000 zImage;nand erase 40000 400000;nand write 20000000 4000
0 400000
smc911x: initializing
smc911x: detected LAN9215 controller
smc911x: phy initialized
smc911x: MAC 00:40:5c:26:0a:5b
Using smc911x device
TFTP from server 192.168.0.177; our IP address is 192.168.0.236
Filename 'zImage'.
Load address: 0x20000000
Loading: #####
#####
#####
done
Bytes transferred = 2054860 (0x1f5acc)
NAND erase: device 0 offset 0x40000, size 0x400000
Erasing at 0x420000 -- 100% complete.
OK
NAND write: device 0 offset 0x40000, size 0x400000
4194304 bytes written: OK
SMDKC100 #
```

Connect USB OTG with USB Stick.



After connect USB Stick, it appear "sda1" and then fill in "mount /dev/sda1 /mnt". You can mount to in /mnt.

```
usb 2-1: configuration #1 chosen from 1 choice
scsi0 : SCSI emulation for USB Mass Storage devices
usb-storage: device found at 2
usb-storage: waiting for device to settle before scanning
scsi 0:0:0:0: Direct-Access    DW           Jewelry           1.20 PQ: 0 ANSI: 2
sd 0:0:0:0: [sda] 4030463 512-byte hardware sectors: (2.06 GB/1.92 GiB)
sd 0:0:0:0: [sda] Write Protect is off
sd 0:0:0:0: [sda] Mode Sense: 00 00 00 00
sd 0:0:0:0: [sda] Assuming drive cache: write through
sd 0:0:0:0: [sda] 4030463 512-byte hardware sectors: (2.06 GB/1.92 GiB)
sd 0:0:0:0: [sda] Write Protect is off
sd 0:0:0:0: [sda] Mode Sense: 00 00 00 00
sd 0:0:0:0: [sda] Assuming drive cache: write through
sda: sda1
sd 0:0:0:0: [sda] Attached SCSI removable disk
sd 0:0:0:0: Attached scsi generic sg0 type 0
usb-storage: device scan complete

[root@Samsung ~]# mount /dev/sda1 /mnt
[root@Samsung ~]# cd /mnt
[root@Samsung mnt]# ls
12.03_256MB/      ???c100???/    android/
??? ???/        WinCE.zip*     c100_b11.091026/
[root@Samsung mnt]#
```

7.2. USB Stick mount

After connect USB Stick, it appear "sda1" and then fill in "mount /dev/sda1 /mnt". You can mount to in /mnt.

```
sd 0:0:0:0: [sda] Write Protect is off
sd 0:0:0:0: [sda] Mode Sense: 00 00 00 00
sd 0:0:0:0: [sda] Assuming drive cache: write through
sda: sda1
sd 0:0:0:0: [sda] Attached SCSI removable disk
sd 0:0:0:0: Attached scsi generic sg0 type 0
usb-storage: device scan complete

[root@Samsung ~]# mount /dev/sda1 /mnt
[root@Samsung ~]# cd /mnt
[root@Samsung mnt]# ls
12.03_256MB/      c100_b11.091026/  android/
??? ???/        WinCE.zip*
```

7.3. SD CARD mount

After connect SD Socket, it appear "p1" and then fill in "mount /dev/mmcblk0p1 /mnt". You can mount to in /mnt.

```
[root@Samsung ~]# mmc0: new SD card at address e624
mmcblk0: mmc0:e624 SD02G 1.89 GiB
mmcblk0: p1

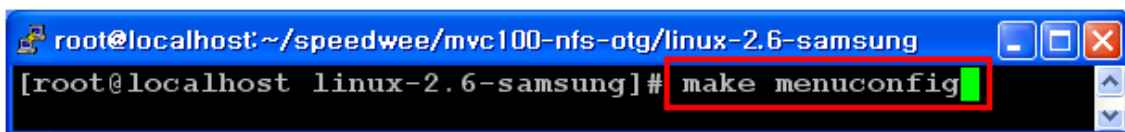
[root@Samsung ~]# mount /dev/mmcblk0p1 /mnt
[root@Samsung ~]# cd /mnt
[root@Samsung mnt]# ls
CD-EWAVR-520-2.zip*      ebook/
Eclipse/                mp3/
G31MX-K_Lan_Xp-1-1.zip* others/
Intel_6.14.10.4837.zip* photo/
NDP1.0sp3-KB867461-X86-Kor.exe* pre.txt*
NDP1.1sp1-KB867460-X86.exe* vmemo/
[root@Samsung mnt]#
```

8. To setup USB Gadget

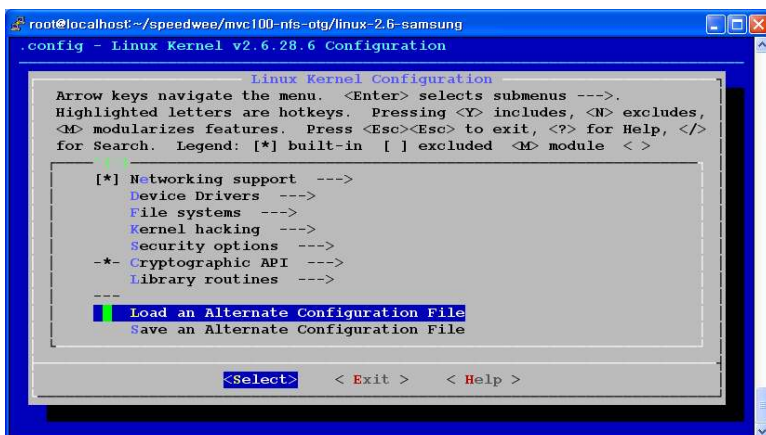
You made an image “Ext2 File system” which is USB Gadget can share file system Windows from Target of Board.

For USB Gadget, most check “*” “<> S3C USB OTG Host support”.

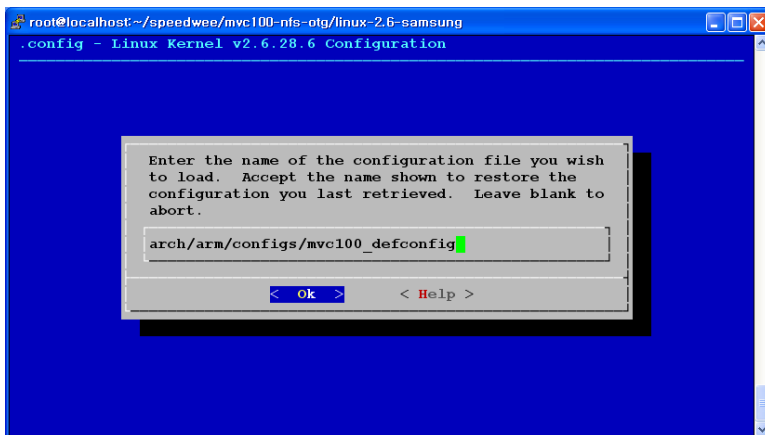
Run “make menuconfig” in Kernel for USB OTG.



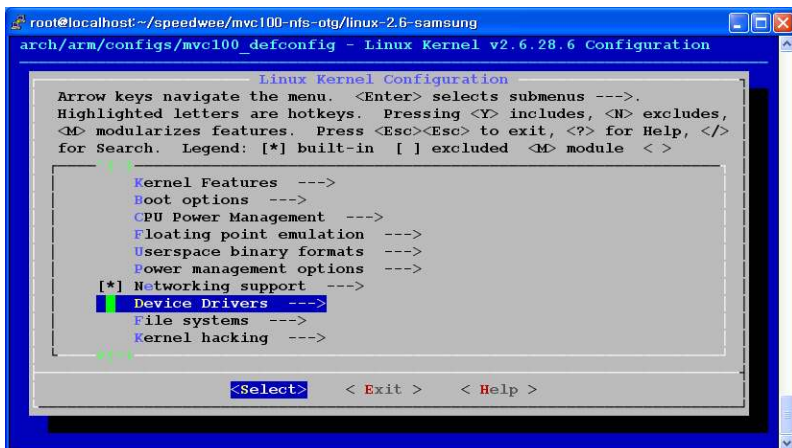
Select “Load an Alternate Configuration File”.



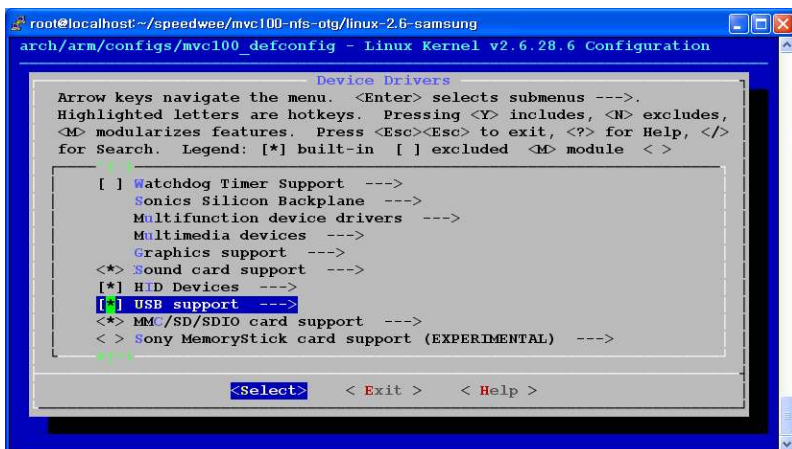
Load "arch/arm/configs/mvc100_defconfig".



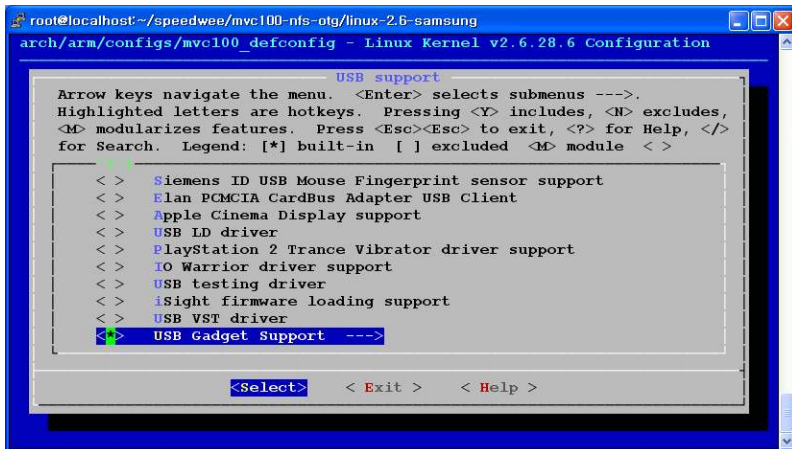
Select "Device Drivers --->".



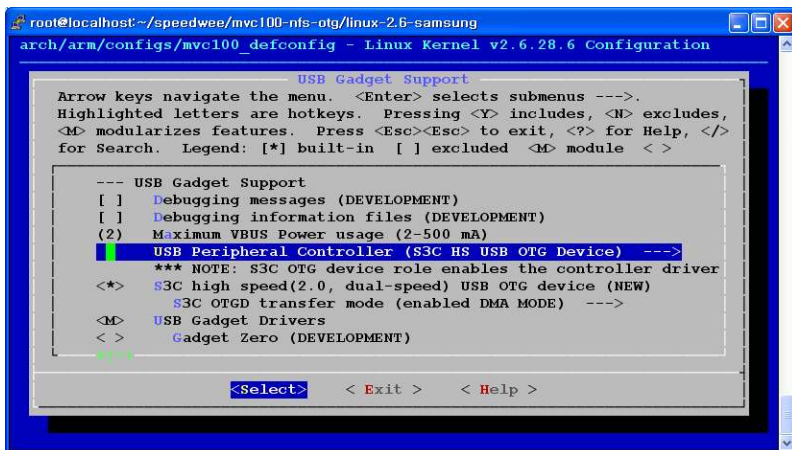
Select "[*] USB support --->".



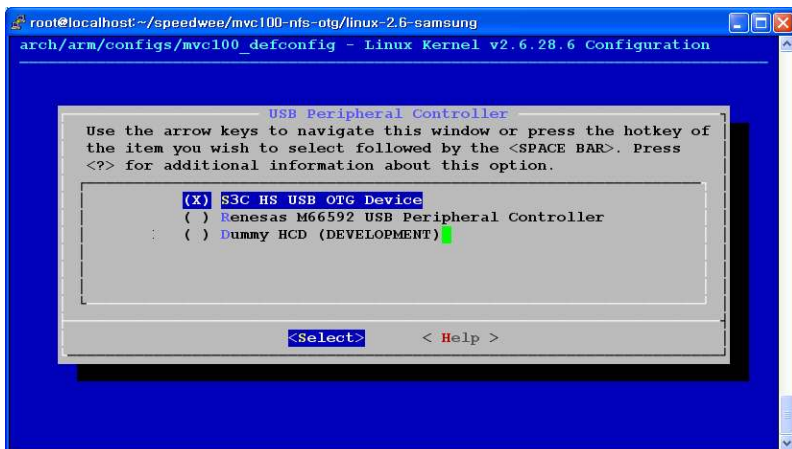
Select “<*> USB Gadget Support --->”.



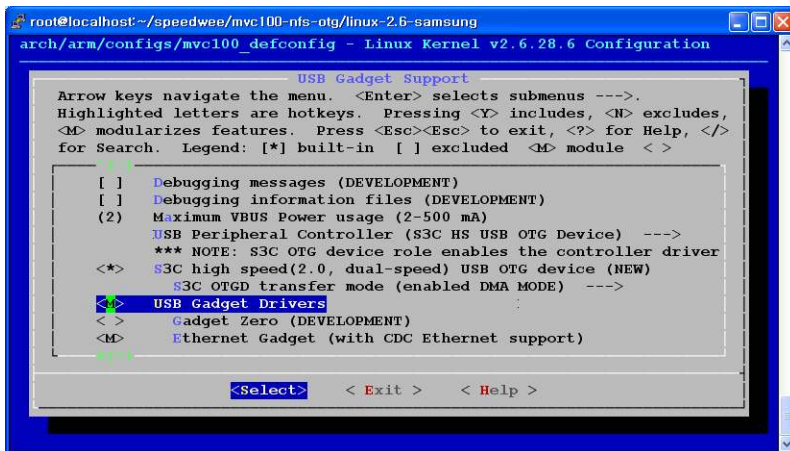
Select “USB Peripheral Controller (S3C HS USB OTG Device) --->”.



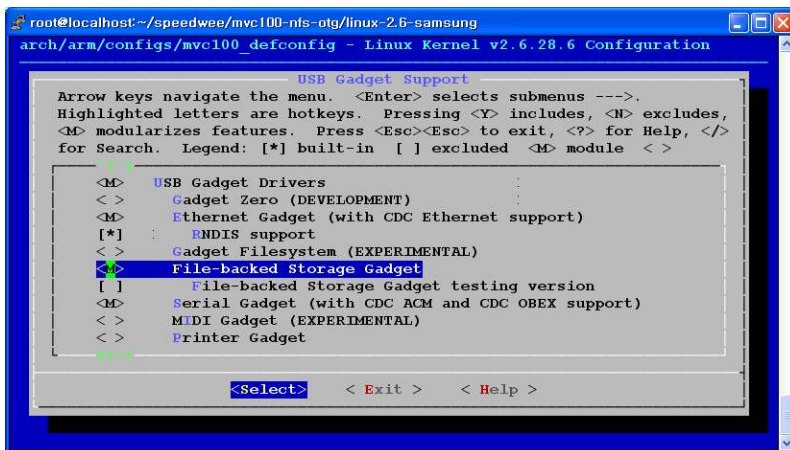
Select “(X) S3C HS USB OTG Device”.



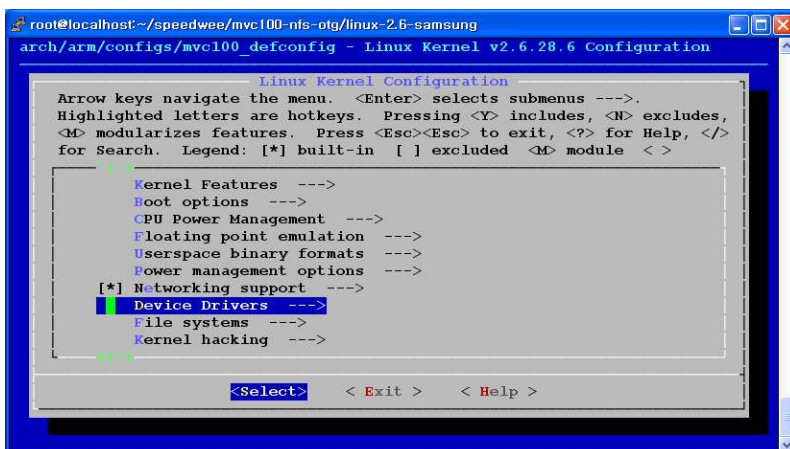
Make sure “<M> USB Gadget Drivers”.



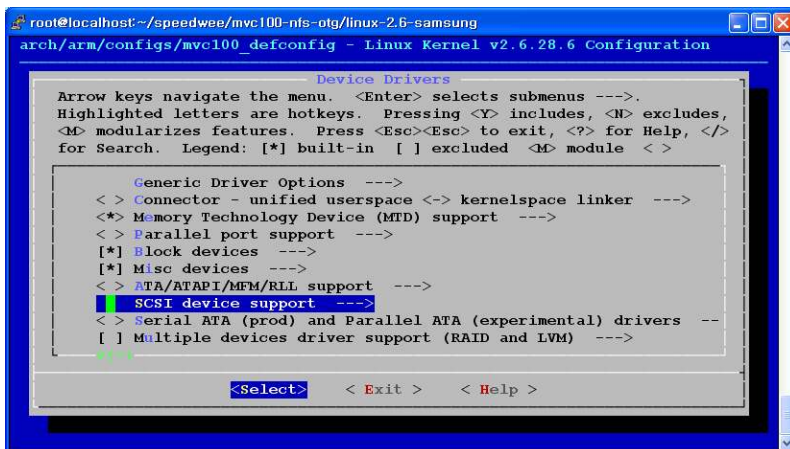
Make sure “<M> File-backed Storage Gadget”.



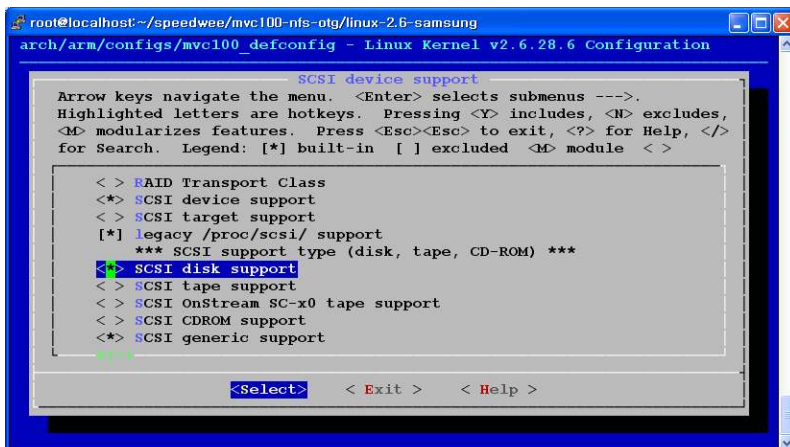
Select “Device Drivers --->”.



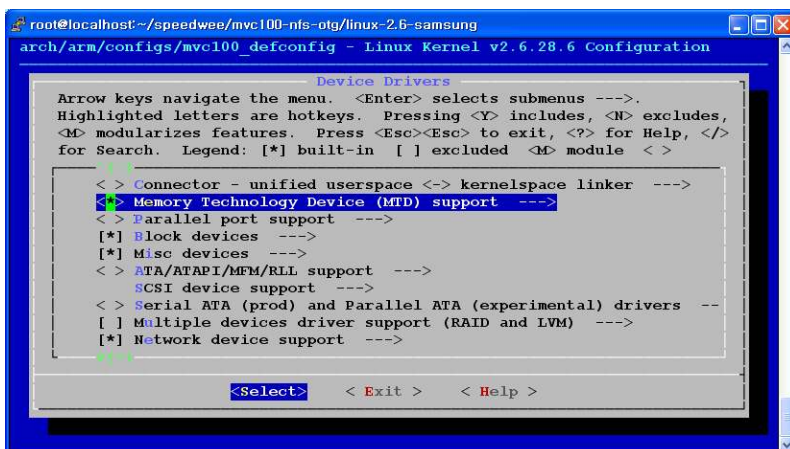
Select “SCSI device support --->”.



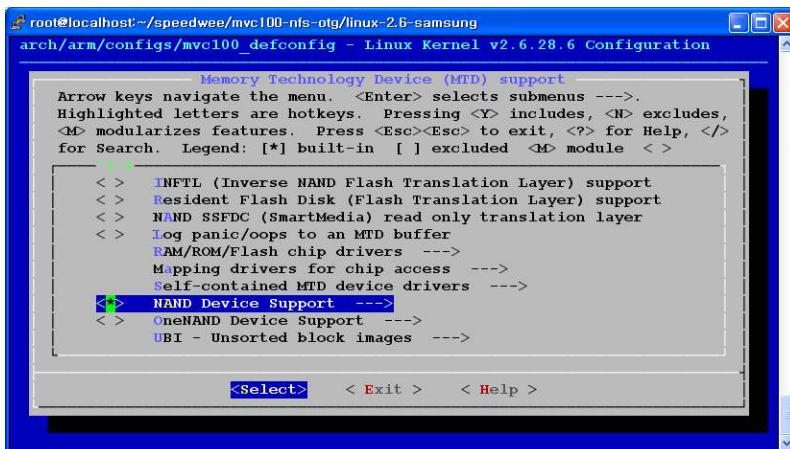
Check “*” “<*> SCSI disk support”.



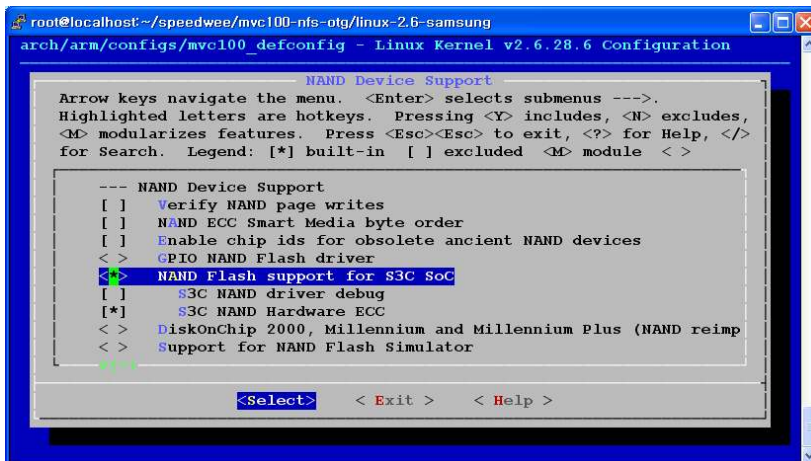
Select “<*> Memory Technology Device (MTD) support --->”.



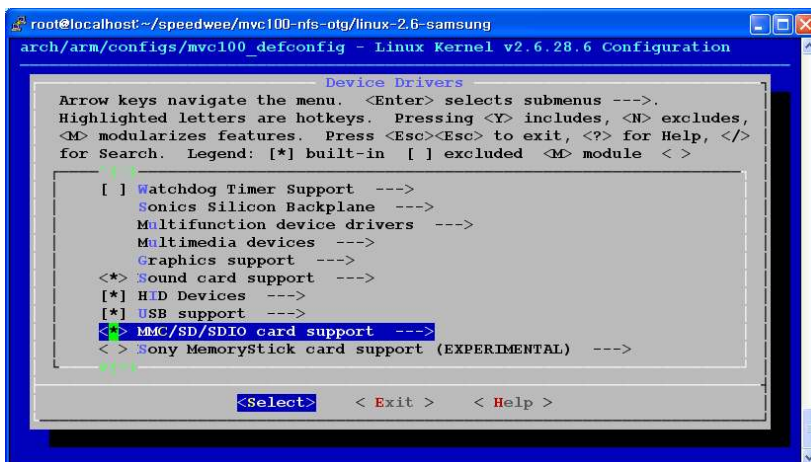
Select “<*> NAND Device Support --->”.



Check “*” “<*> NAND Flash support for S3C SoC”.



Select “<*> MMC/SD/SDIO card support --->”.



Check "*" "<*>" "SDHCI support on Samsung S3C SoC" and then run "Make"

```

root@localhost:~/speedwee/mvc100-nfs-otg/linux-2.6-samsung
arch/arm/configs/mvc100_defconfig - Linux Kernel v2.6.28.6 Configuration

MMC/SD/SDIO card support
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [ ] excluded <M> module <>

[ ] : Allow unsafe resume (DANGEROUS)
*** MMC/SD/SDIO Card Drivers ***
<*> MMC block device driver
[*] Use bounce buffer for simple hosts
<> SDIO UART/GPS class support
<> MMC host test driver
*** MMC/SD/SDIO Host Controller Drivers ***
<*> Secure Digital Host Controller Interface support
<*> SDHCI support on Samsung S3C SoC
<> MMC/SD/SDIO over SPI

<Select> <Exit > <Help >

```

After compiled, Run 3 commands for making "Ext2 File System".

```

# dd if=/dev/zero of=gadgetdisk bs=512 count=4096
# mkdosfs -n "GadgetDisk" -S 512 gadgetdisk 2048
# mkfs -t ext2 -m 0 gadgetdisk 2048

```

```

root@localhost:~/speedwee/gadget

[root@localhost speedwee]# mkdir gadget
[root@localhost speedwee]# cd gadget
[root@localhost gadget]# dd if=/dev/zero of=gadgetdisk bs=512 count=4096
4096+0 records in
4096+0 records out
2097152 bytes (2.1 MB) copied, 0.056229 seconds, 37.3 MB/s
[root@localhost gadget]# mkdosfs -n "GadgetDisk" -S 512 gadgetdisk 2048
mkdosfs 2.11 (12 Mar 2005)
[root@localhost gadget]# mkfs -t ext2 -m 0 gadgetdisk 2048
mke2fs 1.39 (29-May-2006)
gadgetdisk is not a block special device.
Proceed anyway? (y,n) y
Filesystem label=
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
256 inodes, 2048 blocks
0 blocks (0.00%) reserved for the super user
First data block=1
Maximum filesystem blocks=2097152
1 block group
8192 blocks per group, 8192 fragments per group
256 inodes per group

Writing inode tables: done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 20 mounts or
180 days, whichever comes first. Use tune2fs -c or -i to override.
[root@localhost gadget]# ls
gadgetdisk
[root@localhost gadget]#

```

As order picture,

It was made “g_file_storage.ko” in /linux-2.6-samsung/drivers/usb/gadget.

```

root@localhost:~/speedwee/mvc100_20100105/linux-2.6-samsung/drivers/usb/gadget
atmel_usba_udc.c  g_ether.o          pxa25x_udc.h
atmel_usba_udc.h  g_file_storage.ko  pxa27x_udc.c
built-in.o        g_file_storage.mod.c  pxa27x_udc.h
cdc2.c           g_file_storage.mod.o  rndis.c
composite.c      g_file_storage.o     rndis.h
config.c         g_serial.ko          s3c2410_udc.c
dummy_hcd.c     g_serial.mod.c       s3c2410_udc.h
epautoconf.c    g_serial.mod.o       s3c_udc.h
ether.c          g_serial.o           s3c_udc_otg.c
ether.o          g_zero.h             s3c_udc_otg.o
f_acm.c          gadget_chips.h       s3c_udc_otg_xfer_dma.c
f_ecm.c          gmidi.c              s3c_udc_otg_xfer_slave.c
f_loopback.c    goku_udc.c           serial.c
f_obex.c         goku_udc.h           serial.o
f_rndis.c        inode.c              u_ether.c
f_serial.c       lh7a40x_udc.c        u_ether.h
f_sourcesink.c   lh7a40x_udc.h        u_serial.c
f_subset.c       m66592-udc.c         u_serial.h
file_storage.c   m66592-udc.h         usbstring.c
file_storage.o   modules.order        zero.c
fsl_qe_udc.c     ndis.h

[root@localhost gadget]# pwd
/root/speedwee/mvc100_20100105/linux-2.6-samsung/drivers/usb/gadget
[root@localhost gadget]#

```

Copy “g_file_storage.ko and “gadgetdisk” to SD Card and then mount on the board. make new folder /test in /tmp and then copy “g_file_storage.ko and “gadgetdisk” /tmp/test.

```

[root@Samsung ~]# mount /dev/mmcblk0p1 /mnt
[root@Samsung ~]# cd /
[root@Samsung /]# cd /tmp/
[root@Samsung tmp]# mkdir test
[root@Samsung tmp]# cd /mnt
[root@Samsung mnt]# ls
Recycled/          demo/              g_file_storage.ko* gadgetdisk*
[root@Samsung mnt]# cp g_file_storage.ko gadgetdisk /tmp/test/
[root@Samsung mnt]# cd /tmp/test/
[root@Samsung test]# ls
g_file_storage.ko* gadgetdisk*
[root@Samsung test]# _

```

Run for insmod “g_file_storage.ko”

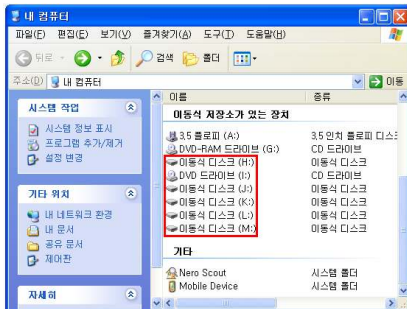
```
# insmod g_file_storage.ko file=gadgetdisk removable=1
```

```

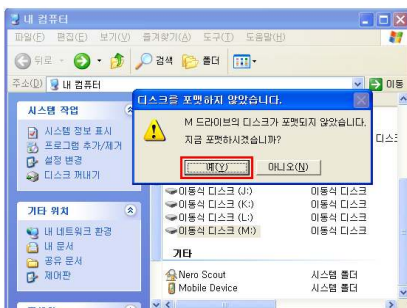
[root@Samsung test]# ls
g_file_storage.ko* gadgetdisk*
[root@Samsung test]# insmod g_file_storage.ko file=gadgetdisk removable=1
g_file_storage gadget: File-backed Storage Gadget, version: 7 August 2007
g_file_storage gadget: Number of LUNs=1
g_file_storage gadget-lun0: ro=0, file: /tmp/test/gadgetdisk
Registered gadget driver 'g_file_storage'
[root@Samsung test]# hub 1-0:1.0: unable to enumerate USB device on port 2
g_file_storage gadget: high speed config #1

```

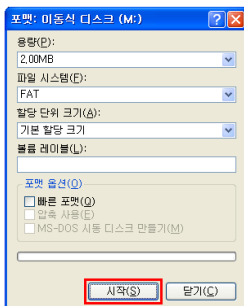
USB Cable connects on Windows PC so that is aware of “like USB”. Click that appear a message “format”.



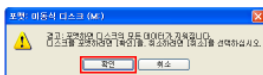
Click “Yes”.



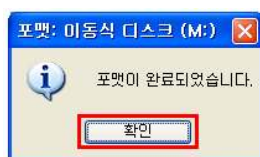
Click “Start”.



Click “OK”.



Click “OK”.



You can copy from windows PC to Board in the “gadgetdisk” File System image.

You'd like to see, you would mount “gadgetdisk”.

```
# umount /mnt
```

```
# mount -o loop /tmp/test/ gadgetdisk /mnt
```

