

제 1 장 임베디드 리눅스 개발 환경

[1] 호스트 PC 환경 설정

- 호스트 PC에는 VMware에 ubuntu 를 설치한다
- 타겟 보드와의 인터페이스를 위한 호스트 PC 의 RS-232C 포트와 이더넷 포트가 준비 되어야 한다
- 호스트 PC에는 TFTP와 NFS 서버를 설치하여야 한다
- 호스트 PC에는 타겟보드용 크로스 컴파일러가 설치 되어 있어야 한다

[2] 크로스 컴파일러 설치

- 하드디스크의 root 디렉토리에 “el6410-cd” 라는 디렉토리를 만들고 제공된 CD를 복사한다
- CD에서 제공하는 linux용 크로스 컴파일러는 4.3.2-eabi-arm.tar.gz 이다.
이 크로스 컴파일러는 Linux kernel 2.6 버전을 컴파일 하기 위한 것이다.
- mkdir 명령을 이용하여 크로스 컴파일 툴 체인 파일의 압축을 풀 디렉토리 /cross_compile 을 생성한다
- 생성한 디렉토리로 이동 후 tar 명령을 이용하여 압축을 푼다

```
root@elayer-desktop:~# mkdir /cross_compile  
  
root@elayer-desktop:~# cd /cross_compile  
  
root@elayer-desktop:/cross_compile# tar -zxvf /root/el6410-cd/linux/toolchain/  
4.3.2-eabi-arm.tar.gz
```

- 설치가 완료되면 bin 디렉토리 아래에 arm 크로스 컴파일을 위한 바이너리 파일들을 확인 할 수 있다

```
root@elayer-desktop:/cross_compile# ls -al 4.3.2-eabi-arm/bin
```

- 환경변수 파일을 수정한다

```
root@elayer-desktop:~# vi /etc/environment (혹은 gedit /etc/environment)
```

- 아래와 같이 PATH 를 추가한다 (한줄로 작성한다)

```
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/cross_compile/4.3.2-eabi-arm/bin"
```

- source 명령을 실행하여 변경된 환경 변수를 시스템에 적용하고 PATH를 확인해본다

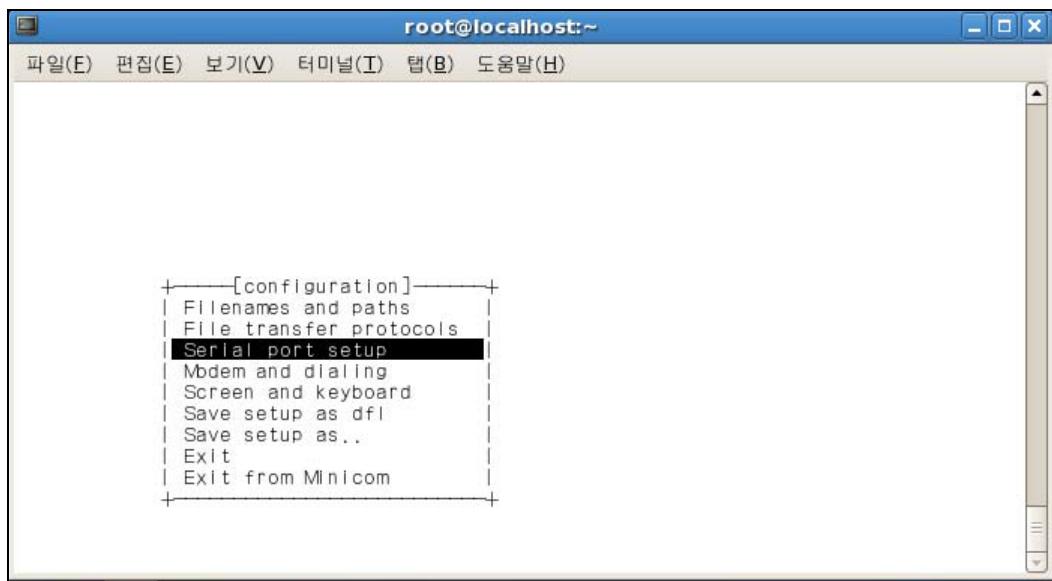
```
root@elayer-desktop:~#
root@elayer-desktop:~# source /etc/environment
root@elayer-desktop:~#
root@elayer-desktop:~# echo $PATH
```

[3] minicom 설정

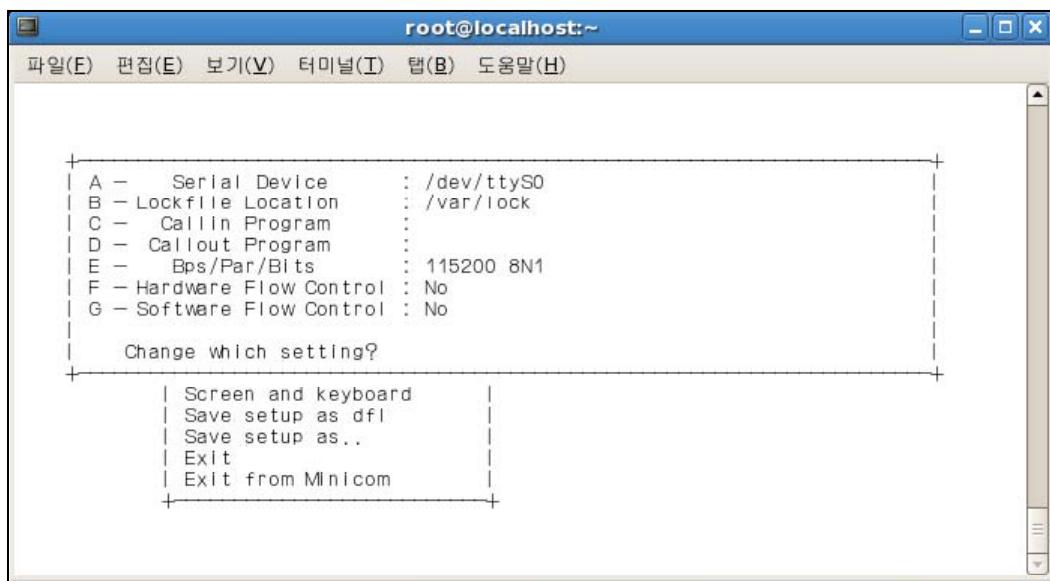
리눅스의 Minicom은 원도우의 하이퍼터미널이나 새롬 데이터 맨과 같은 시리얼 통신 애플레이터로서 eL6410의 모니터 역할을 하는 유ти리티이다.

- Minicom 설정을 위한 명령으로 다음의 명령을 수행한다.

```
root@elayer-desktop:~# apt-get install minicom  
root@elayer-desktop:~# minicom -s
```



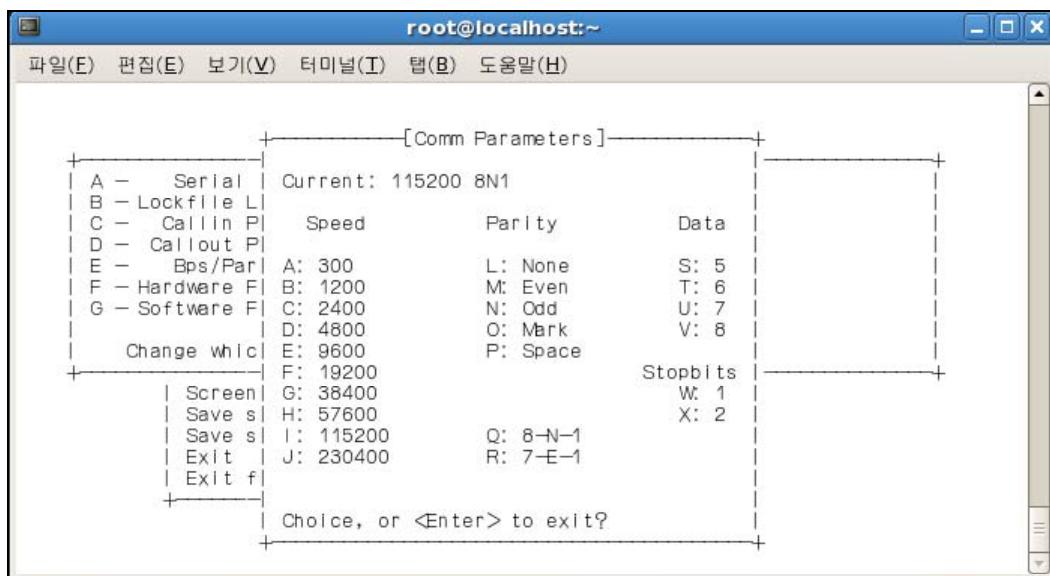
Serial port setup을 선택한다. PC의 COM1 일 경우에는 /dev/ttyS0, COM2 일 경우에는 /dev/ttyS1 이다. USB to Serial 인 경우에는 /dev/ttyUSB0 이다



Parameters를 설정한다.

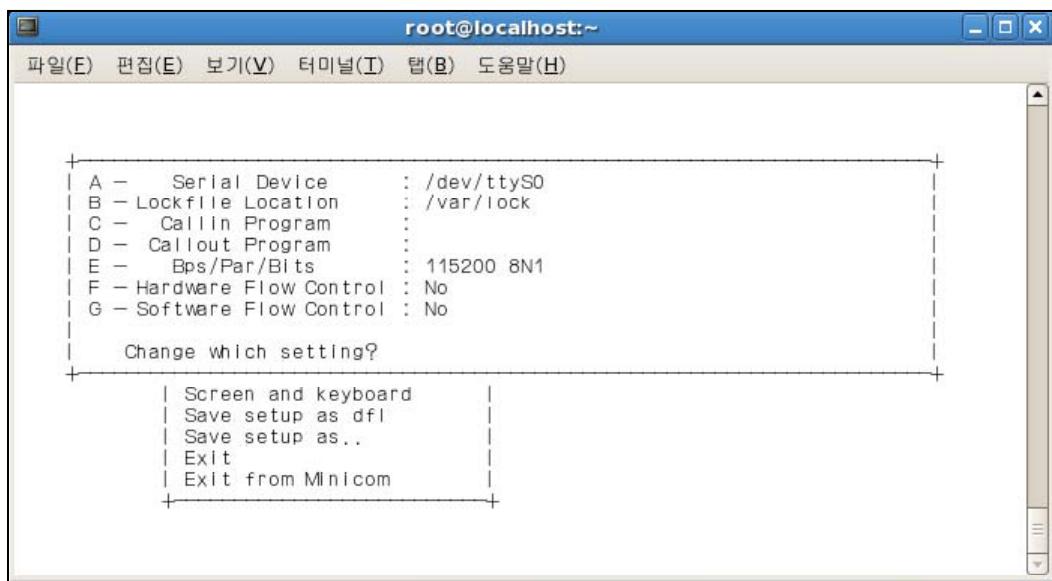
설정해야 할 값들 : [Baudrate(speed) : 115200 Parity : none Data bits : 8 Stop bits : 1]

E를 누르면 아래와 같은 모습을 볼 수 있다.



I (115200 선택)와 Q (8-N-1 선택)를 누르고 엔터를 친다.

다음으로 F와 G를 눌러 Hardware Flow Control과 Software Flow Control을 No로 설정한다



설정을 저장한다. Save setup as df1 을 선택하고 enter를 친다. 저장이 완료되면 Exit from Minicom을 선택하고 enter를 친 후 빠져 나온다.

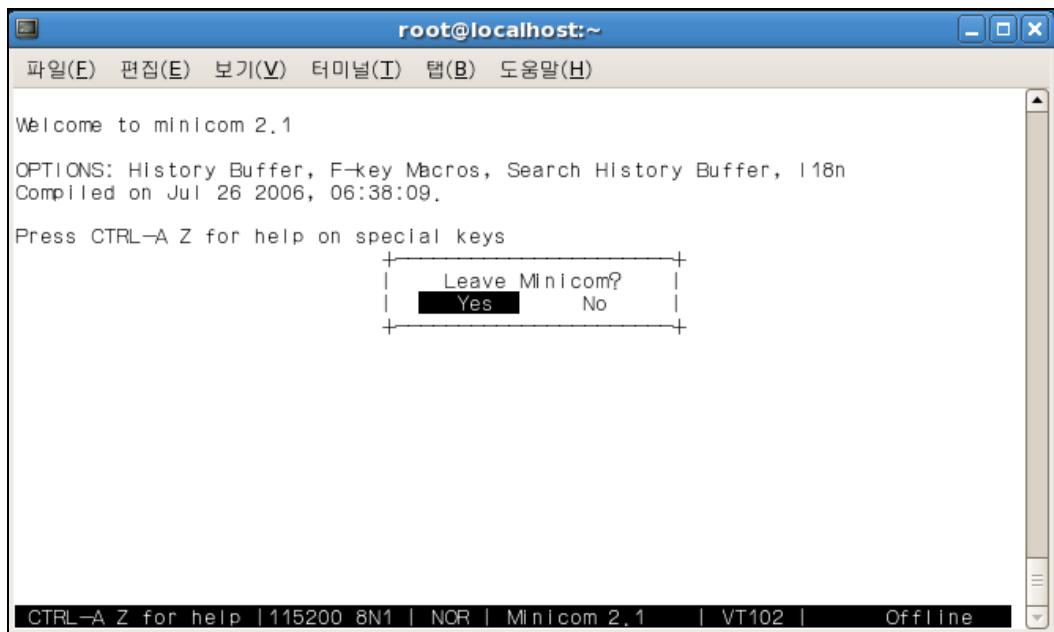
이제부터는 minicom 이라는 명령만 입력하면 설정된 값대로 minicom이 동작하게 된다.
Minicom을 다시 실행해 보자.

```
root@elayer-desktop:~# minicom
```



이 상황에서 Ctrl+A를 누른 후 Z키를 누르면 다음과 같은 minicom 메뉴가 나온다.
각 메뉴에 대한 설명은 하지 않는다. 본인이 직접 한번 해보기 바란다.

Minicom이 동작하는 상황에서 Ctrl+A를 누른 후 X를 누르면 minicom을 종료하게 된다.



- 타겟 보드와 시리얼 포트를 연결하고 전원을 켜서 아래 화면이 출력되는지를 확인한다

```
U-Boot 1.1.6 (Nov 11 2009 - 23:12:28) for SMDK6410
```

```
CPU:      S3C6410@532MHz
          Fclk = 532MHz, Hclk = 133MHz, Pclk = 66MHz, Serial = CLKUART (SYNC Mod)
Board:    SMDK6410
DRAM:    128 MB
Flash:   0 kB
NAND:    256 MB
In:      serial
Out:     serial
Err:     serial
Hit any key to stop autoboot:  0

NAND read: device 0 offset 0x40000, size 0x200000
2097152 bytes read: OK
Boot with zImage

Starting kernel ...

Uncompressing Linux.....
Linux version 2.6.29-00245-g03f14d7-dirty (root@elayer-desktop) (gcc version 4.0
CPU: ARMv6-compatible processor [410fb766] revision 6 (ARMv7), cr=00c5387f
CPU: VIPT nonaliasing data cache, VIPT nonaliasing instruction cache
Machine: SMDK6410
Memory policy: ECC disabled, Data cache writeback
CPU S3C6410 (id 0x36410101)
...
(중략)
```

[4] TFTP 서버 설치

- TFTP란 FTP 프로토콜의 하나로서 파일을 송수신 하기 위한 UDP 기반의 프로토콜이다. 임베디드 시스템을 개발하는 과정에서 eL6410 u-boot에서 리눅스의 커널이나 파일 시스템, WinCE 이미지같이 큰 데이터를 PC로부터 받아 와야 할 경우가 빈번하게 발생하게 되는데 이 경우 아주 유용하게 사용할 수 있는 네트워크 프로토콜이다.
이 프로토콜을 기반으로 서버로 동작하는 TFTP 서버를 설치한다.

- TFTP,NFS server, xinetd 설치

```
root@elayer-desktop:~# apt-get install nfs-kernel-server tftpd tftp xinetd
```

☞ 뒤에 커널 컴파일시 make menuconfig의 정상실행을 위해 아래를 함께 실행해놓자
root@elayer-desktop:~# apt-get install libncurses5-dev

- xinetd 설정파일 편집

```
$ vi /etc/xinetd.d/tftpd (혹은 gedit /etc/xinetd.d/tftpd )
```

```
service tftp
{
    protocol      = udp
    port          = 69
    socket_type   = dgram
    wait          = yes
    user          = nobody
    server        = /usr/sbin/in.tftpd
    server_args   = /tftpboot
    disable       = no
}
```

- TFTP 서비스를 사용 가능하도록 xinetd 데몬을 재시작 한다.

```
mkdir /tftpboot
chmod 777 /tftpboot
/etc/init.d/xinetd restart
```

- 개발 보드 부팅

호스트 PC 의 /tftpboot 디렉토리에 ramdisk.gz 와 zImage 파일을 복사해두고 minicom 을 실행하고 u-boot로 부팅후 아래 TFTP 명령을 실행하면 리눅스로 부팅할 수 있다
(VMware를 의 네트워크를 NAT에서 Bridged로 설정 변경하여 재부팅한다)

```
# setenv bootargs "root=/dev/ram0 rw ramdisk=16384 initrd=0x51000000,16M console=ttySAC1,115200"
# tftp 50008000 zImage
# tftp 51000000 rootfs_ramdisk.gz
# bootm 50008000
```

```

mmc_bus: source is mout_epll (0), rate is 24000000
mmc_bus: source is mout_epll (0), rate is 24000000
mmc_bus: source is mout_epll (0), rate is 24000000
usb-bus-host: source is mout_epll (0), rate is 24000000
uclk1: source is dout_mppll (1), rate is 66500000
spi-bus: source is mout_epll (0), rate is 24000000
spi-bus: source is mout_epll (0), rate is 24000000
audio-bus: source is mout_epll (0), rate is 24000000
audio-bus: source is mout_epll (0), rate is 24000000
irda-bus: source is mout_epll (0), rate is 24000000
camif: source is hclk (0), rate is 133000000
camera: source is hclk2 (0), rate is 266000000
Built 1 zonelists in Zone order, mobility grouping on. Total pages: 32512
Kernel command line: rootfstype=jffs2 root=/dev/mtdblock2 init=/init
console=ttySAC1,115200
PID hash table entries: 512 (order: 9, 2048 bytes)
Console: colour dummy device 80x30
console [ttySAC1] enabled
Dentry cache hash table entries: 16384 (order: 4, 65536 bytes)
Inode-cache hash table entries: 8192 (order: 3, 32768 bytes)
Memory: 128MB = 128MB total
Memory: 125424KB available (3384K code, 780K data, 144K init)
SLUB: Genslabs=10, HWalign=32, Order=0-3, MinObjects=0, CPUs=1, Nodes=1
...
Empty flash at 0x01dca72c ends at 0x01dca800
Empty flash at 0x01dd1e60 ends at 0x01dd2000
VFS: Mounted root (jffs2 filesystem) on device 31:2.
Freeing init memory: 144K
Warning: unable to open an initial console.
init: cannot open '/initlogo.rle'
yaffs: dev is 32505859 name is "mtdblock3"
yaffs: passed flags ""
yaffs: Attempting MTD mount on 31.3, "mtdblock3"
s3c-nand: ECC uncorrectable error detected
s3c-nand: ECC uncorrectable error detected
block 8 is bad
block 524 is bad
yaffs_read_super: isCheckpointed 0
init: cannot find '/system/bin/playmp3', disabling 'bootsound'
sh: can't access tty; job control turned off
# warning: `rild` uses 32-bit capabilities (legacy support in use)
smdk6410_hifi_hw_params: clockrate 24000000
s3c2410_dma_devconfig: channel 10, source 1, dev 7f002010, chan c039feec
smdk6410_hifi_hw_params: clockrate 24000000
s3c2410_dma_devconfig: channel 10, source 1, dev 7f002010, chan c039feec
JFFS2 notice: (1670) check_node_data: wrong data CRC in data node at 0x01dd13e8:
read 0xfa059749, calculated 0xeb14e3d7.
...

```

[5] NFS 설정

- NFS란 파일 시스템 중의 하나로서 RPC를 이용하여 원격 호스트의 디렉토리를 마치 유저 자신의 로컬디렉토리인 것처럼 사용할 수 있게 해주는 파일 시스템이다.

이 파일 시스템은 원격 호스트 상에서 NFS를 지원해 주어야 할 뿐만 아니라, 유저가 사용하는 시스템에서도 지원해 주어야 한다.

- /etc/exports 파일을 수정한다.

```
# vi /etc/exports (혹은 gedit /etc/exports)
```

- 아래와 같이 마지막 라인을 추가한다. 공백문자 없도록 작성한다

```
# /etc/exports: the access control list for filesystems which may be exported
#           to NFS clients.  See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes      hostname1(rw,sync) hostname2(ro,sync)
#
# Example for NFSv4:
# /srv/nfs4      gss/krb5i(rw,async,fsid=0,crossmnt)
# /srv/nfs4/homes  gss/krb5i(rw,async)
#
/tftpboot      *(rw,no_root_squash,no_all_squash,async)
```

/tftpboot : nfs mount시 마운트 될 디렉토리 지정(다른 디렉토리도 상관 없음)

* : 모든 클라이언트가 접근 가능하도록 설정

rw : 읽기,쓰기 권한 부여

no_root_squash,no_all_squash : 클라이언트들이 I/O 요청을 할 때 서로 충돌하지 않도록 해주기 위한 설정

- 환경 설정이 끝나고 나면 설정을 확인한다

```
# exportfs -r
```

- nfs 데몬을 다시 시작한다.

```
# /etc/init.d/nfs-kernel-server restart
```

- 타겟 보드에서 NFS를 마운트하여 본다

```
# ifconfig eth0 192.168.100.3 up
# mount -t nfs -o nolock,rsize=1024,wsize=1024 192.168.100.2:/tftpboot /mnt/nfs
# df
[root@pjlinux ~]$ df
Filesystem      1k-blocks   Used   Available Use% Mounted on
/dev/root        15863     10688     4356    71% /
192.168.100.2:/tftpboot 29640796 14644372 13490736 52% /mnt/nfs
[root@pjlinux ~]$ cd /mnt/nfs
[root@pjlinux nfs]$ ls
```

- 마운트 실패시 Host-PC에서 NFS가 제대로 동작하는지 테스트 해보자.

```
# mkdir /mnt/nfs
# mount -t nfs 192.168.100.2:/tftpboot /mnt/nfs
# df
```

mkdir 명령을 이용하여 /mnt/nfs 디렉토리를 하나 만든다

192.168.100.2 : nfs 서버가 동작하고 있는 호스트 PC 의 IP 주소

/tftpboot : nfs 서버가 동작하고 있는 호스트 PC 에서 export시킨 디렉토리

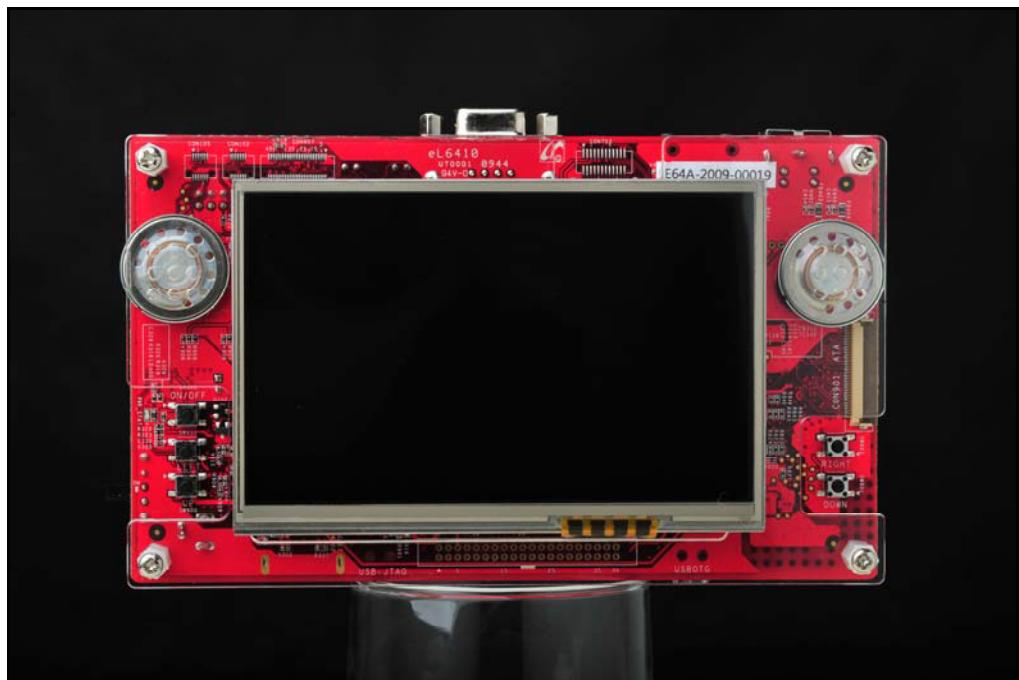
/mnt/nfs : 마운트 시키고자 하는 클라이언트의 로컬 디렉토리

[6] Target Board 소개

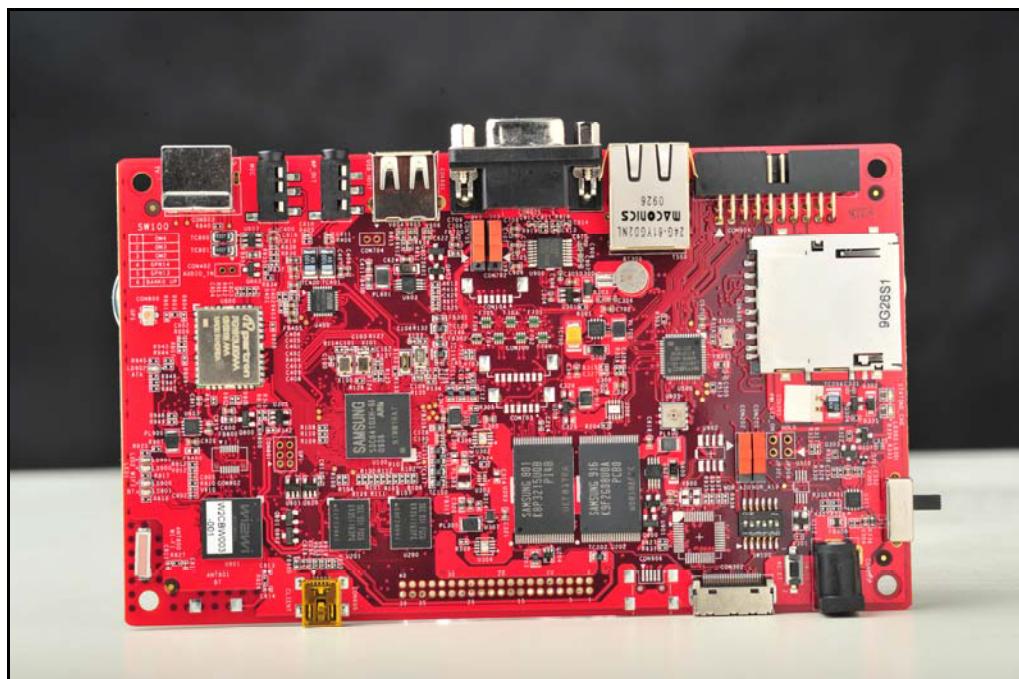
S3C6410를 기반으로 한 eL6410 보드는 S3C6410와 여러 가지 디바이스들을 하드웨어로 연결하고 이를 이용하여 여러 가지 solution들을 개발할 수 있음을 보여 주기 위한 reference 보드이다. 이레이어 시스템(www.elayer.co.kr) 홈페이지 참조

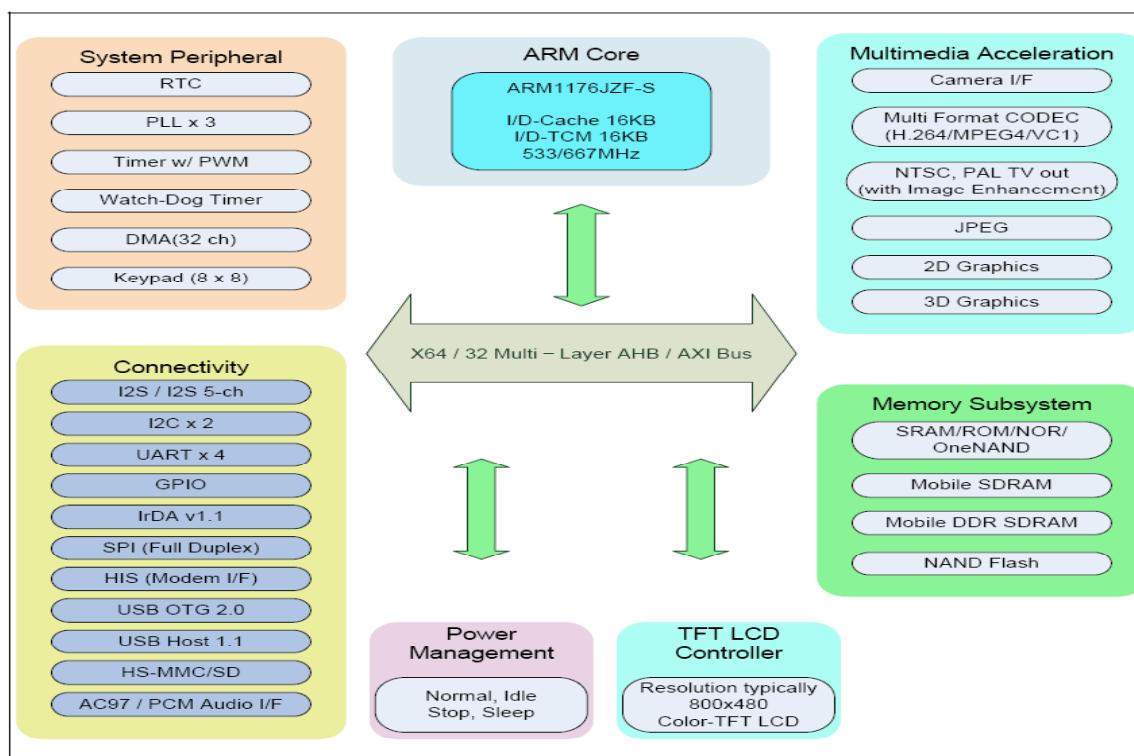
- CPU : Samsung **S3C6410X** ARM1176JZF-S 667MHz Application Processor
- Memory : **128Mbytes Mobile DDR**
 - 256Mbytes SLC NAND Flash
 - 4Mbytes NOR Flash, sectored into 4 separate banks of 1Mbytes memory space can be accessed by S3C6410
- Display : **5" WVGA(800x480) Color TFT** with Touch Screen Interface
- Audio : Wolfson **WM8960** Audio Codec with 1W Stereo Speaker Amplifier
- Ethernet : **SMSC LAN9220 10/100Mbps** Ethernet Controller
- USB : **USB 1.1 Host / USB 2.0 OTG**
- SD : SD/MMC Port 0 : WiFi(Wi2Wi) or Expansion Connector
 - SD/MMC Port 1 : Standard SD Connector
 - SD/MMC Port 2 : Expansion Connector
- SPI :
 - SPI Port 0 : Expansion Connector
 - SPI Port 1 : Expansion Connector shared with SD/MMC Port 2
- UART : UART Port 0 : GPS or Expansion Connector
 - UART Port 1 : Standard DSUB9
 - UART Port 2 : Expansion Connector
 - UART Port 3 : Bluetooth or Expansion Connector
- JTAG : Standard ARM JTAG Interface
- TV-out
- WiFi/Bluetooth Combo : Wi2Wi WiFi/BT combo module, not mounted on alpha board
- GPS
- Power : 5V/2A DC-JACK
- Camera Expansion Port
- Support Interlaced and Non-Interlaced CCIR601/656 Standard Camera Interface
- ATA HDD Connector
- Sensors : 3-Axis Acceleration Sensor
- Keys : **4 General Purpose Keys**, 1 Reset Keys, 1 Tactile Power Keys
- Indicators : **2 Indicator LED** controllable from GPIO ports
- Expansion Connectors : IrDA, Camera, I2S, SPI0/SPI1, SDIO0/2, EBI, UART, LCD

- eL6410 보드 앞면

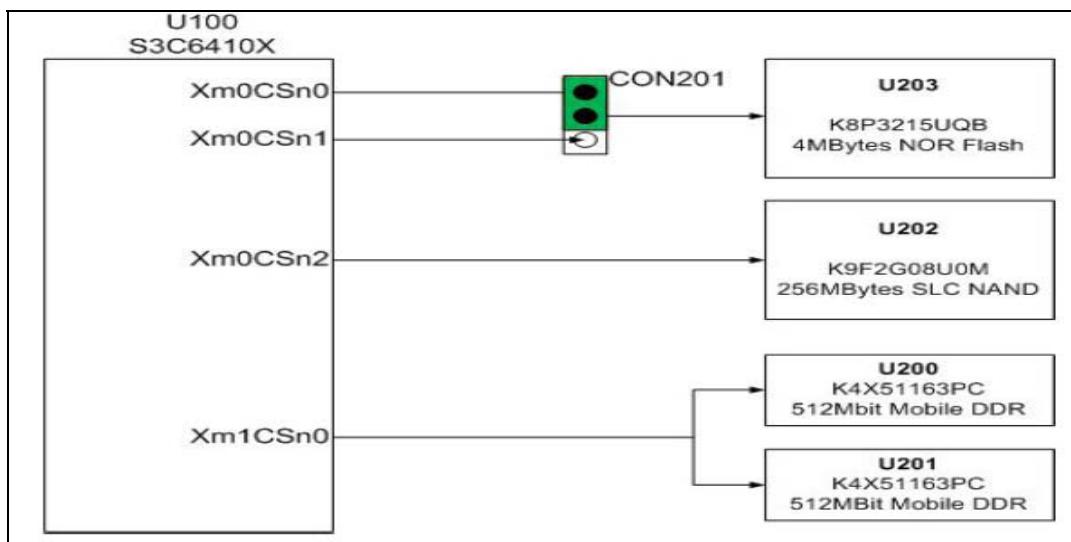


- eL6410 보드 뒷면





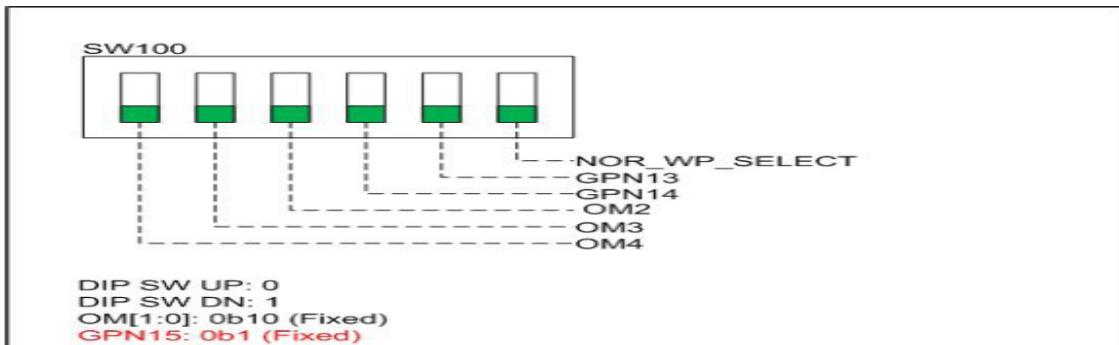
[S3C6410X Block Diagram]



[eL6410 메모리 시스템 구조]

- SW100 스위치의 설정

No.	Signal Name	Description
1	OM4	OM[1:0] = 0b10
2	OM3	
3	OM2	
4	GPN14	GPN15 = 0b1
5	GPN13	
6	NOR_WP_SELECT	NOR Flash Write Protection Mode

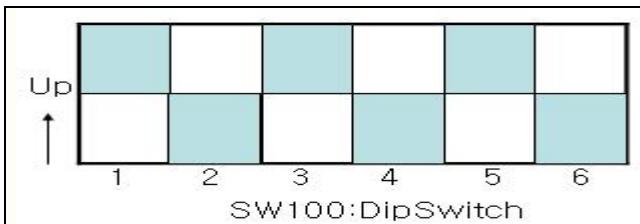


위로 올리면 : 0 (Low)

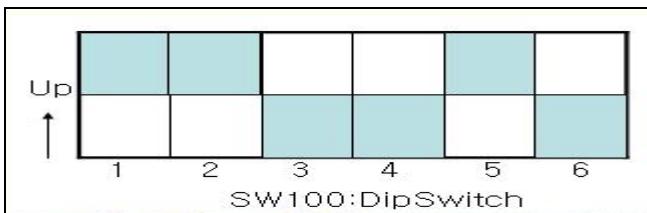
아래로 내리면 : 1 (High)

XSELNAND	OM[4:0]	GPN[15:13]	Boot Device	Function	Clock Source
1	0000X	XXX	RESERVED	RESERVED	NAND Stepping-Stone Boot
1	0001X			RESERVED	
1	0010X			RESERVED	
1	0011X			RESERVED	
X	0100X	SROM(8bit)		-	XXTIpll if OM[0] is 0. XEXTCLK if OM[0] is 1.
X	0101X	SROM(16bit)		-	
0	0110X	OneNAND ¹⁾	OneNAND	Don't use NAND Device	
X	0111X		MODEM	Don't use Xm0CSn2 for SROMC	
X		000	IROM ²⁾	SD/MMC(CH0)	
0		001		OneNAND	
1		010		NAND(512Byte, 3-Cycle)	
1		011		NAND(512Byte, 4-Cycle)	
1	1111X	100		NAND(2048Byte, 4-Cycle)	
1		101		NAND(2048Byte, 5-Cycle)	
1		110		NAND(4096Byte, 5-Cycle)	
X		111		SD/MMC(CH1)	

- NOR Flash Booting mode 선택 : 0101X



- NAND Flash Booting mode 선택 : 0011X



Xm0 Address	Xm1Data 공유	GPQ 공유	비고
Xm0Addr[15:0]	Xm1Data[26:16]		Xm1Data[31:16]은 DRAM에서 사용하므로, GPQ로 공유
Xm0Addr[19:16]		GPQ[1,0,7,8]	
Xm0Addr[26:20]			

S3C6410X에서 NOR Flash 및 기타 IO에 접근하는 메모리 포트의 주소 공간은, mobile SDRAM 또는 mobile DDR SDRAM의 데이터 버스를 32 비트로 사용할 경우, 20bit로 제한되어, 1MByte의 메모리 공간만 접근 가능하다. 이 보드에 장착된 NOR Flash는 4MByte의 용량이므로, 이 Flash를 최대한 활용하기 위해, 4 개의 BANK로 나누어 사용할 수 있도록 되어있다.

