비글보드 포팅
안드로이드

1. 비글보드 케이블

2. minicom 시리얼 통신 연결

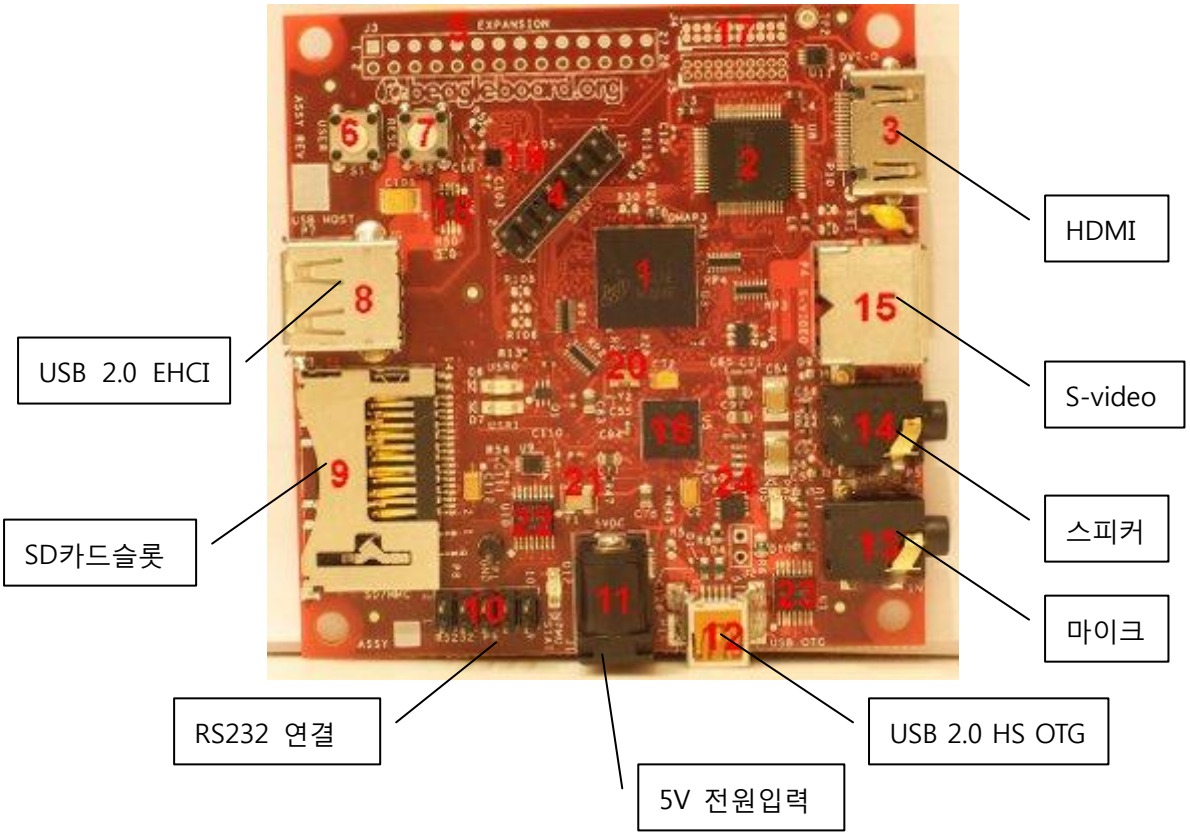
3. 안드로이드 포팅 (1) embinux

4. 안드로이드 포팅 (2) 0xdroid

5. WiFi 드라이버 설치하기

이봉준(bbongcol@gmail.com 010-2062-6008)
http://beagleandroid.springnote.com/

1. 비글보드 케이블



(1) USB Mini OTG 케이블

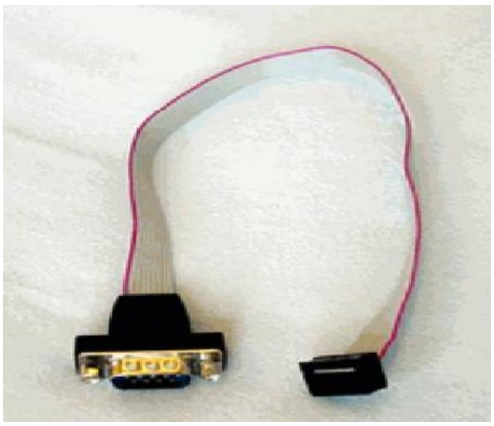
전원을 공급하거나 다른 UBS 기기(키보드, 마우스)를 연결하는데 사용된다. 디카나 핸드폰, MP3 용 케이블에서 많이 사용된다. 케이블을 바로 컴퓨터에 꼽으면 보드로 전원이 공급되고, USB 허브에 뽑으면 다른 UBS 기기를 꼽아 사용할 수 있다. 본 **프로젝트에서는 비글보드의 전원으로 활용하였다.**

이봉준(bbongcol@gmail.com 010-2062-6008)
http://beagleandroid.springnote.com/

(2) DVI-D 케이블

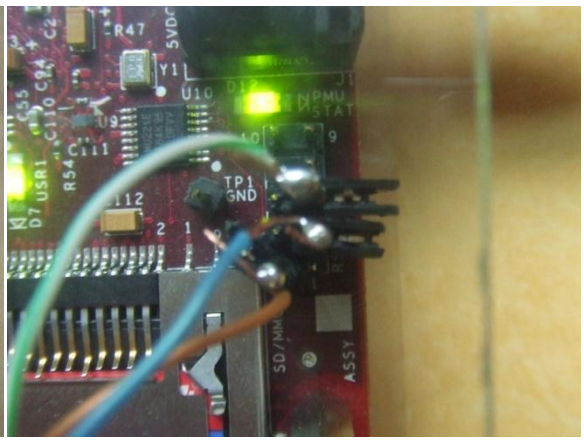
화면 출력을 위한 케이블이다. 비글보드가 HDMI출력이 가능하기 때문에 화면출력으로 이 케이블을 사용하게 된다. 요즘 모니터 뒤에 보면 대부분 이 HDMI 단자가 있기 때문에 쉽게 연결할 수 있다. 비글보드는 이것 말고도 화면 출력으로 S-단자도 있는데 많이 사용하지 않기 때문에 이 DVI-D케이블을 사용하여 화면 출력을 구성하였다.

(3) RS232 케이블



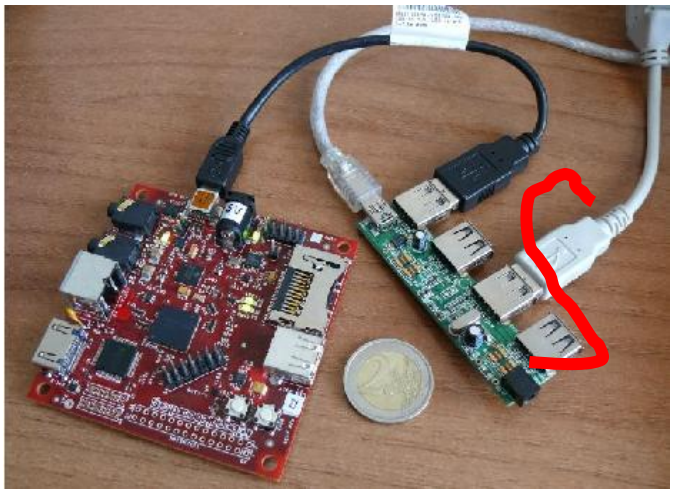
터미널프로그램(minicom)을 이용해 비글보드와 시리얼 통신을 할 때 사용되는 케이블이다. 사실 위 두 케이블만으로 보드가 동작하고 출력을 확인할 수 있지만 RS232 케이블이 연결되어 시리얼 통신으로 부팅 관련 설정을 해줘야 하기 때문에 처음에 꼭 필요한 케이블이다. 만약 케이블이 없으면 보드의 RS232 단자에서 RX, TX, GND 단자만 컴퓨터와 연결할 수도 있다. 비글보드의 핀 정보 http://elinux.org/BeagleBoard 여기에서 확인하능하다.

본 프로젝트에서는 RS232 케이블을 직접 만들어 사용하였다. 연결 방법은 아래 그림과 같이 한다. 이때 보드의 방향을 잘 고려하고 케이블 색을 잘 맞춰 연결하고 쇼트를 주의한다. 잘못 연결 시 비글보드가 고장 날수 있으니 주의한다.



이봉준(bbongcol@gmail.com 010-2062-6008)
http://beagleandroid.springnote.com/

(4) USB 2.0 허브



비글보드에서 키보드, 마우스등의 USB 장치를 사용하기 위해서는 USB 2.0 허브를 사용해야 한다. 위 사진은 USB 2.0 허브를 OTG 부분에 연결한 모습이지만 실제 이렇게 연결하면 안드로이드에서 USB 2.0 허브를 인식 못하는 현상이 나타났다.

안드로이드에서 USB 2.0 허브를 인식하기 위해서는 비글보드의 USB 단자에 USB2.0 허브를 연결해야 한다. 이때 USB Mini OTG 케이블을 이용해 연결하면 된다. 비글보드의 USB 단자는 rev C 부터 장착되어있다.

안드로이드에서 USB 2.0 허브를 인식하기 위해 또 한가지 중요한 점은 **USB 허브 자체 전원**을 줘야 한다는 점이다. 비글보드가 워낙 높은 클럭으로 동작하다 보니 USB쪽으로는 전기를 거의 흘려보내지 못한다. 그래서 USB 허브 자체 전원이 없다면 허브를 인식하지 못하는 문제가 발생한다.

전원 연결 순서는 먼저 USB 허브 자체 전원을 연결 후 비글보드의 전원을 넣으면 비글보드가 USB 허브들 인식하게 된다. 허브 전원을 나중에 주게 되면 비글보드가 부팅이 안되는 문제가 발생할 수 있기 때문에 전원을 주는 순서도 주의해서 연결한다.

※ 참고 웹 사이트

http://elinux.org/BeagleBoard

비글보드에 관련된 모든 자료가 설명되어있다. 비글보드를 제대로 활용하기 위해서는 반드시 한번 읽어봐야 한다. 여기에 설명된 모든 자료는 위 웹사이트를 참고하였다.

이봉준(bbongcol@gmail.com 010-2062-6008)
http://beagleandroid.springnote.com/

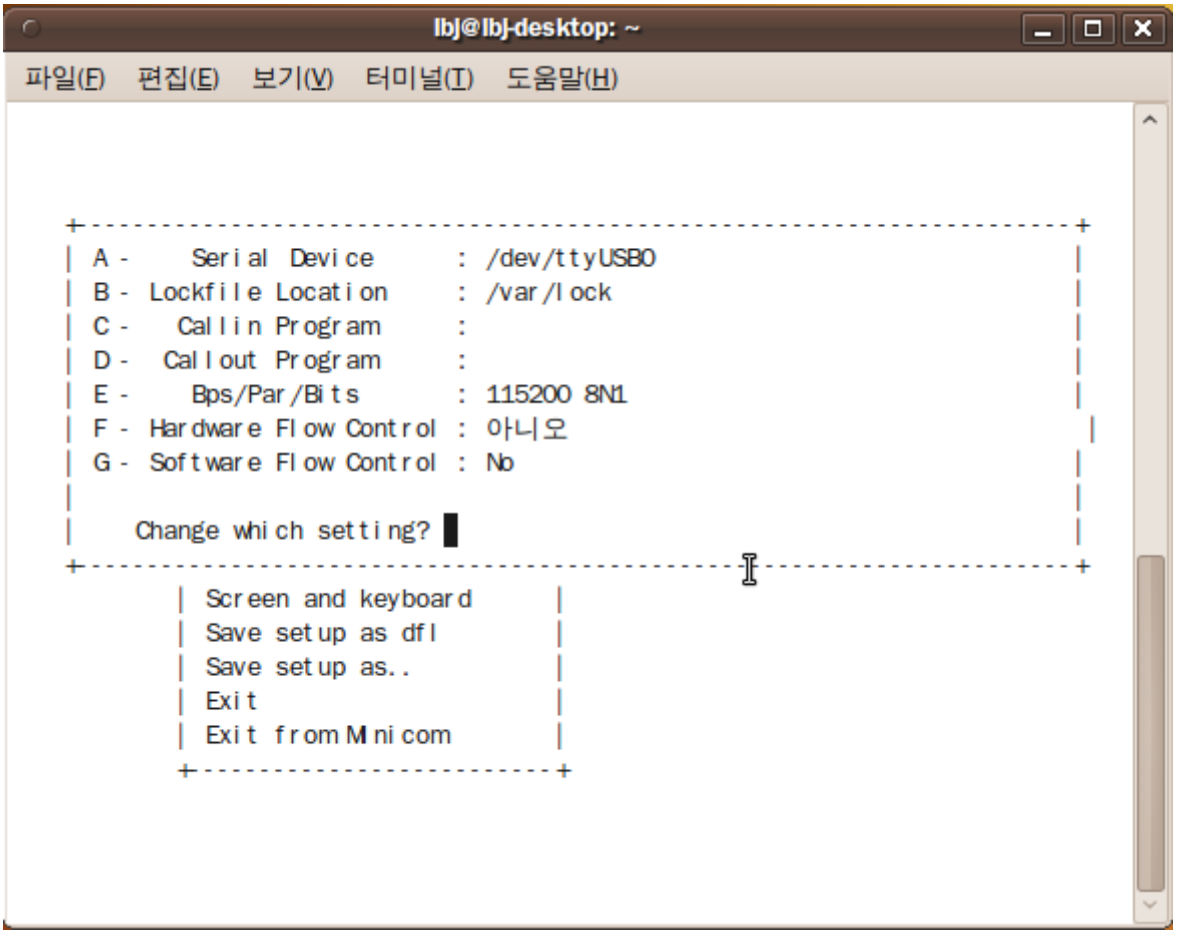
2. minicom 시리얼 통신 연결

(1) minicom 설치하기

```
sudo apt-get install minicom
```

(2) minicom 설정하기

```
minicom - s
```



minicom이 실행되면 **Serial port setup**에서 위와 같이 입력한다. ttyUSB0은 본 프로젝트에서 RS232 케이블을 USB 젠더를 이용해 USB로 연결했기 때문에 바꿔주는것이다.

만약 키보드의 입력이 제대로 이뤄지지 않는다면 우분투 기본 패키지중 하나인 시각 장애를 가진 사람들을 위한 brltty이라는 패키지 때문에 그런것이다. 아래 명령어로 지워 준다.

```
sudo apt-get remove brltty
```

(3) 케이블 연결 후 설정

RS232 케이블을 처음 연결 후 터미널에서 다음과 같이 입력한다. 케이블을 마운트 하는 과정이다.

```
sudo su  (비번입력 후 루트계정으로 들어음)
mknod /dev/ttyUSB0 c 188 0
```

리눅스를 vmware와 같은 가상 머신에서 돌리는 경우 USB가 인식이 되었다 안되었다 하는 경우가 있기 때문에 위 명령어가 에러가 날 경우 USb를 빳다가 다시 꼽은 후 위 명령어를 다시 쳐본다.

이후 USB를 꼽으면 **시스템 – 관리 – 로그파일보기 - dmesg**에서 USB 장치가 인식 되었는지 안되었는지 확인할 수 있다.

3. 안드로이드 포팅 (1) embinux

비글보드 공식적으로 진행되는 안드로이드 포팅 프로젝트이다. SD 카드에 안드로이드 소스파일을 넣고 비글보드에 장착 후 부팅하게 된다. 기본적으로 키보드입력을 지원한다. 안드로이드 소스파일을 그대로 SD카드에 올려서 사용하기 때문에 소스파일 수정이 쉽다.

(1) Install needed software packages

터미널 창에서 다음을 그대로 실행한다. 관리자 계정은 포팅과정에서 필수로 필요하므로 **sudo su** 명령어로 미리 관리자 계정으로 들어가 있는다.

```
apt-get update
apt-get dist-upgrade
apt-get install git-core bison sun-java5-jdk flex g++ zlib1g-dev
apt-get install   libx11-dev libncurses5-dev gperf uboot-mkimage
```

주의할 점은 Java6 을 지원하지 않는다는 것이다. 우분투 9.10에서는 Java6이 기본적으로 설치 되어있어 Java5로 되돌아 갈 수 없기 때문에 우분투 9.04나 8.10버전을 사용할 것을 추천한다.

다음은 repo를 다운로드 하고 권한설정을 한다. repo란 어던 프로젝트 파일을 한번에 뭉쳐서 받기 위한 스크립트이다. 이것을 이용해 안드로이드 소스를 다운로드 한다.

```
mkdir -p ~/bin
cd ~/bin
wget http://android.git.kernel.org/repo
chmod +x repo
```

크로스 컴파일러를 다운로드 한다.

```
cd
wget  http://www.codesourcery.com/sgpp/lite/arm/portal/package1787/public/arm-none-linux-gnueabi/arm-2007q3-51-arm-none-linux-gnueabi-i686-pc-linux-gnu.tar.bz2
cd /opt
sudo tar jxf arm-2007q3-51-arm-none-linux-gnueabi-i686-pc-linux-gnu.tar.bz2
```

크로스 컴파일러란 원래 컴파일은 그 소스가 올라가는 기기(x86 등듣)에서 하는게 맞는데 임베디드 같은 장치에서는 해당 기기에서 컴파일 할 수 없기 때문에 컴퓨터 상에서 임베디드 환경을 구성해 컴파일 시키는 것을 말한다. 별로 어려운 것은 아니고 gcc 같은 소스를 임베디드용으로 다운로드해 설정해 주는거다.

(2) Download android sources

안드로이드를 다운로드 한다. repo 스크립트를 이용해 다운로드 한다. 자동으로 최신 안드로이드 빌드가 다운로드 된다.

```
mkdir ~/beagledroid
cd ~/beagledroid
repo init -u git://labs.embinux.org/repo/android/platform/beaglemanifest.git/
repo sync
```

(3) Building Android

다운로드 받은 소스를 빌드한다. 4~6시간 정도 걸린다.

```
make  (듀얼코어에서는 make -j 4)
```

(4) Building Android

안드로이드는 리눅스 커널위에 올라간다. 따라서 리눅스 커널을 다운로드해 빌드해야 한다.

```
export CC_PATH=/opt/arm-2007q3/bin/arm-none-linux-gnueabi-
cd ~/beagledroid/kernel
../vendor/embinux/support-tools/beagle_build_kernel.sh
```

여기에서 CC_PATH는 위에서 다운로드 한 크로스 컴파일러 위치를 나타낸다. 만약 크로스 컴파일러를 다른 폴더에 다운로드 받았다면 여기를 수정해 줘야한다. 지금까지 명령어 그대로 따라 왔다면 수정이 필요 없을 것이다.

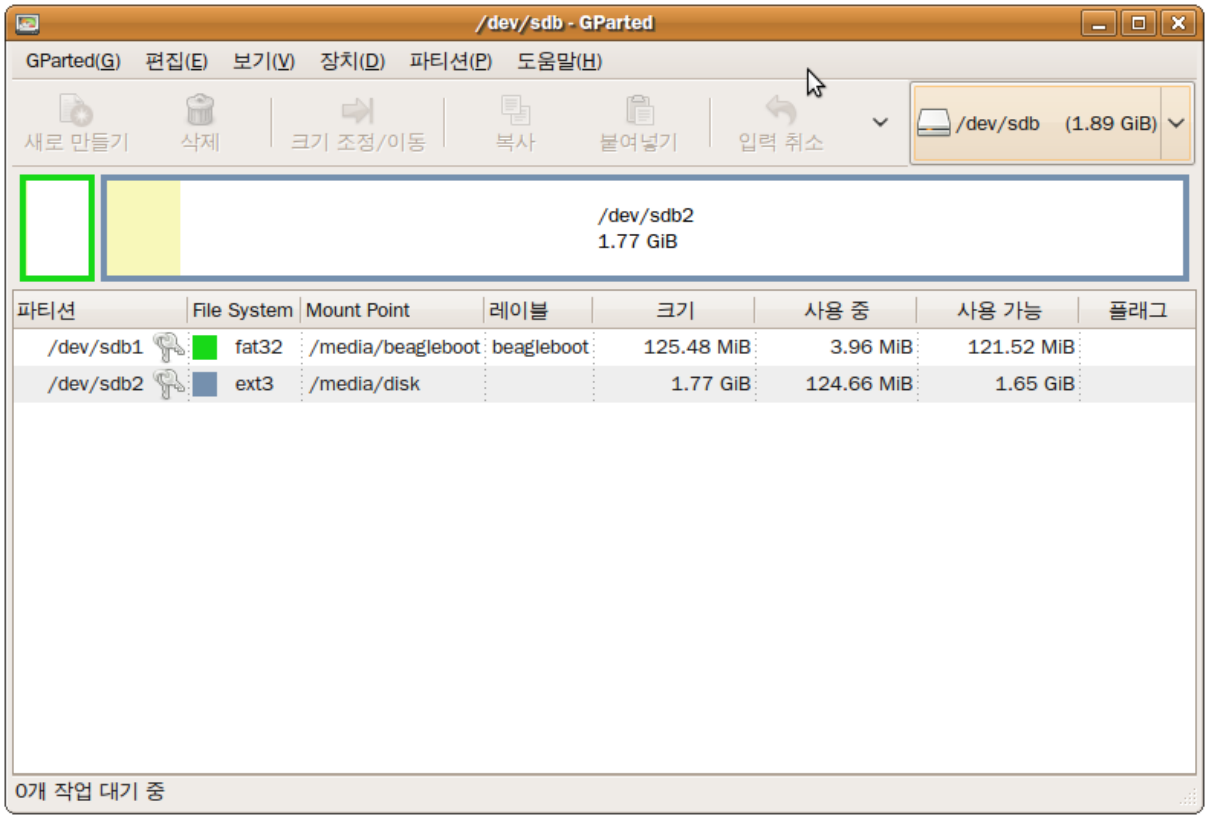
(5) Copying the Android root filesystem

지금까지 만들어진 안드로이드 파일과 리눅스 커널 부팅이미지를 아래 폴더에 집어 넣는다.
~/beagledroid/out/target/product/generic

```
cd ~/beagledroid/out/target/product/generic
mkdir ~/beagledroid/rootfs
cp -a root/* ~/beagledroid/rootfs/
cp -a system/* ~/beagledroid/rootfs/system/
cd ~/beagledroid/rootfs
sudo chown -R root.root .
sudo chmod -R a+rwX data system
```

(6) Formatting an SD card

이제 SD카드를 포맷을 해야 한다. SD를 2G짜리를 추천한다. 리눅스 상에서 포맷은 gparted라는 프로그램을 이용해 간단히 할 수 있다. 우분투에서 **프로그램 – 우분투 소프트웨어 센터**에서 gparted라고 검색후 프로그램을 설치한다.



이제 SD카드를 꼽고 gparted를 실행해 위와 같이 포맷한다. 포맷과정에 대한 상세한 설명은 아래 동영상을 참고 한다.

http://www.youtube.com/watch?v=zymOmduNWyI&feature=related

(7) Copying data to the MMC/SD card

파일을 sd카드에 복사한다. 먼저 커널 부팅이미지 uimage와 기타 부팅관련 파일들을 fat32 파티션에 복사한다.

```
cd /media/beagleboot
wget http://free-electrons.com/pub/demos/beagleboard/android/MLO
http://free-electrons.com/pub/demos/beagleboard/android/u-boot.bin
cp ~/beagledroid/kernel/arch/arm/boot/uImage .
```

이봉준(bbongcol@gmail.com 010-2062-6008)
http://beagleandroid.springnote.com/

다운은 안드로이드 파일을 ext3 파티션에 복사한다.

```
sudo rsync -a ~/beagledroid/rootfs/ /media/disk/
```

이 과정이 조금 익숙해 지면 윈도우 상에서 ~/beagledroid/out/target/product/generic 에 있는 파일들을 그냥 복사해서 sd카드로 붙여 넣어도 된다. 이때 권한이 없어 붙여 넣기가 안될때가 있는데 그때는 터미널에서 **gksudo nautilus** 명령어를 실행해 열린 윈도우 창에서 붙여 넣기 하면된다. 유용한 명령어니 기억해 두도록 한다.

이제 마운트를 해제 한다.

```
sudo umount /media/beagleboot
sudo umount /media/disk
```

(7) 부팅 셋업

이제 sd카드를 비글보드에 장착하고 RS232 케이블을 보드에 연결 후 비글보드에 전원을 넣는다. 다음 minicom을 실행하면 비글보드와 시리얼 통신이 시작된다. 비글보드는 시작과 동시에 부팅이 진행된다. minicom이 실행되면 제일먼저 **Ctrl + a**을 누르고 다음 **w** 를 누른다. 프롬프트에서 다음 줄로 넘어 갈 수 있도록 해주는 설정이다. 꼭 기억하자

Hit any key to stop autoboot 에서 엔터를 처서 **OMAP3 beagleboard.org #** 프롬프트로 넘어간다. 그럼 아래 명령어들을 입력한다.

```
setenv bootcmd 'mmc init;fatload mmc 0 80000000 uImage;bootm 80000000'
saveenv
```

부팅 이미지 위치를 설정하는 부분이다. 이때 mmc init가 안먹 힌다면 mmcinit로 시도 해본다. 또 80000000이 안 먹힌다면 80200000을 시도해 본다.

```
setenv bootargs console=ttyS2,115200n8 noinitrd  root=/dev/mmcblk0p2
omapfb.video_mode=dvi:1280x720MR-24@60 init=/init rootfstype=ext3 rw rootdelay=1
nohz=off androidboot.console=ttyS2
```

위에 명령어는 한 줄짜리 명령어다 전부 복사해서 붙여 넣기 하면 된다. 1280x720는 출력화면비율인데 적용이 잘 안되는 것 같다. 출력은 DVI단자로 나가게 된다.

만약 S-video로 출력을 하고 싶으면 아래와 같이 입력한다. 하지만 화면이 잘리고 떨려서 실효성이 거의 없다.

```
setenv bootargs console=ttyS2,115200n8 noinitrd  root=/dev/mmcblk0p2
omapdss.def_disp=tv omapfb.video_mode=tv:800x600MR-24@60 init=/init rootfstype=ext3 rw
rootdelay=1 nohz=off androidboot.console=ttyS2
```

omapdss.def_disp=lcd라고 하면 lcd로 출력한다. 비글보드 위키에 보면 LCD 포트라는게 있느데 보드상에 구멍만 뚫려 있고 LCD를 연결하기 위해서는 따로 연결 단자를 만들어야 한다.

※레졸루션 정보
http://groups.google.com/group/beagleboard/msg/4c64b2c614622053?pli=1

```
boot
```

boot 명령어를 입력하면 부팅이 시작된다.

```
reading uImage

1905104 bytes read
## Booting kernel from Legacy Image at 80300000 ...
   Image Name:   Linux-2.6.29-omap1-07177-g5cda98
   Image Type:   ARM Linux Kernel Image (uncompressed)
   Data Size:    1905040 Bytes =  1.8 MB
   Load Address: 80008000
   Entry Point:  80008000
   Verifying Checksum ... OK
   Loading Kernel Image ... OK
OK

Starting kernel ...

Uncompressing Linux.....................................................................
```

※참고 웹사이트

http://labs.embinux.org/index.php/Android_Porting_Guide_to_Beagle_Board
http://elinux.org/Android_on_OMAP
http://code.google.com/p/beagleboard/wiki/BeagleboardRevCValidation
http://elinux.org/Android_Portal
http://free-electrons.com/blog/android-beagle/

이봉준(bbongcol@gmail.com 010-2062-6008)
http://beagleandroid.springnote.com/

4. 안드로이드 포팅 (2) 0xdroid

0xdroid는 대만의 0xlab이라는 곳에서 진행중인 프로젝트로 매일 빌드한 파일을 업로드 하기 때문에 안드로이드 포팅 과정 없이 비글보드에 안드로이드를 올릴 수 있어서 편리하다. 또한 sd카드에 파일을 올리고 부팅하면 안드로이드 이미지를 비글보드 내부의 낸드에 복사하는 과정이 진행되어 다음 부팅시에는 sd 카드 없이 비글보드 만으로 안드로이드를 부팅할 수 있다. 낸드에서 부팅하기 때문에 속도가 sd카드를 이용하는 것 보다 빠르다. 또한 키보드와 마우스 입력을 완벽히 지원한다.

http://code.google.com/p/0xdroid/

위 웹사이트를 통해 0xdroid 프로젝트에 대한 모든 정보를 확인해 볼 수 있다.

http://downloads.0xlab.org/dailybuild/

위 웹사이트를 방문하게 되면 간단히 도든 과정에 대해서 설명되어 있다. 여기서는 간단히 이 방법을 소개하고 이후 소스를 다운로드해 포팅하는 과정을 설명하겠다.

(1) dailybuild를 이용해 간단히 비글보드에 안드로이드 올리기

- Gparted를 이용해 sd카드 전체를 Fat32로 포맷한다.
- http://downloads.0xlab.org/installer/ 여기에서 uImage.bin를 다운로드 후 sd카드에 복사
- **0xkernel-beagle.bin**를 다운로드 후 sd 카드에 넣는다.
- **android-beagle.ubi**를 다운로드 후 sd카드에 넣는다
- sd 카드를 꼽고 비글보드에 전원을 넣는다.

```
mmcinit      or mmc init (depends on your u-boot)
run loaduimage
nand erase 280000 400000
nand write ${loadaddr} 280000 400000
boot
```

이제 안드로이드 이미지를 비글보드 낸드에 복사하는 과정이 인쟁된다. 복사가 끝났다는 메시지가 나오게 되면 sd카드를 분리하고 비글보드를 리셋한다.

여기에서 중요한 점은 비글보드가 키보드와 마우스를 인식하는가 하는 문제인데 비글보드 케이블 단락에서 설명했듯이 USB2.0에 자체 전원을 먼저 입력하고 비글보드 전원을 입력해야 하는 것을 유의해야 한다.

이봉준(bbongcol@gmail.com 010-2062-6008)
http://beagleandroid.springnote.com/

(2) 0xdroid 프로젝트 안드로이드 소스 빌드

먼저 관리자 권한으로 들어가는 것을 잊지 말자. 아래 명령어들을 모두 입력해 관련 패키지를 다운로드 한다.

```
sudo apt-get install build-essential
sudo apt-get install make
sudo apt-get install gcc
sudo apt-get install g++
sudo apt-get install libc6-dev
sudo apt-get install patch
sudo apt-get install texinfo
sudo apt-get install libncurses-dev
sudo apt-get install git-core gnupg
sudo apt-get install flex bison gperf libsdl-dev libesd0-dev libwxgtk2.6-dev build-essential zip curl
sudo apt-get install ncurses-dev
sudo apt-get install zlib1g-dev
sudo apt-get install valgrind
sudo apt-get install python2.6
sudo apt-get install gawk
sudo apt-get install sun-java5-jdk   sun-java5-jre
```

위에서도 언급했듯이 안드로이드 빌드에는 java5를 지원하는데 우분우 9.10에서는 java6이 이미 설치 되어있어 java5가 설치되지 않는다 따라서 우분투 9.04나 8.10사용 할것을 권장한다.

```
mkdir beagle-cupcake
cd beagle-cupcake
repo init -u git://gitorious.org/0xdroid/manifest.git -b beagle-cupcake
echo  "TARGET_PRODUCT := beagleboard"  > buildspec.mk
echo "INSTALL_PREBUILT_DEMO_APKS := true" >> buildspec.mk
repo sync
make   # if you have 4 cpus you can use "make -j4"
```

repo sync를 통해 다운로드 시 중간에 계속 끊긴다면 해외 플록시 서버를 경유해서 받는 것을 추천한다. 소스 다운로드와 커널 빌드에 상당히 많은 시간이 소요된다. 크로스 컴파일러 문제가 생길 수 있는데 embnux에서의 설명을 참고 해 해결한다. 크로스 컴파일러는 http://downloads.0xlab.org/toolchain/armv7/ 여기에서 다운로드 할 수 있다. Make파일의 CROSS_COMPILE을 직접 수정해 문제를 해결 할 수도 있다.

(3) 0xdroid 프로젝트 커널 소스 빌드

커널 소스를 다운도르 하고 빌드 한다.

```
git clone git://gitorious.org/0xlab-kernel/kernel.git
git branch -r
git checkout -b omap3 origin/omap3

cp arch/arm/configs/omap3_beagle_defconfig .config
make menuconfig
make
```

make menuconfig는 커널 빌드 전 커널환경을 설정하는 곳으로 각종 드라이버를 설치 하기 위해서는 여기서 설정해주어야 한다.

(4) 안드로이드 이미지 파일 만들기

0xdroid 프로젝트의 경우 빌드된 안드로이드를 루트파일들을 그대로 사용하지 않고 이미지를 만들어야 한다. dailybuild에 올라온 0xkernel-beagle.bin 파일이 바로 안드로이드 파일 시스템이다. 우리도 이 파일을 생성해야 한다.

```
cd out/target/product/beagleboard
wget http://downloads.0xlab.org/release/beagle-cupcake-0x1/ubi_scripts.tgz
tar xvf (where you put ubi scripts)/ubi_scripts.tgz
mv ubi_scripts/* .
./generateubifs.sh

mkfs.ubifs에서 에러가 난다면 아래 명령어를 친다.
sudo apt-get install liblzo2-dev uuid-dev
```

잠시 후 간은 폴더에 0xkernel-beagle.bin 파일이 생성된 것을 확인 할 수 있다.

※참고 웹사이트

http://gitorious.org/0xdroid/pages/Build-from-Scratch
http://code.google.com/p/0xdroid/
http://downloads.0xlab.org/dailybuild/
http://code.google.com/p/0xdroid/wiki/boot_nand
http://downloads.0xlab.org/installer/

이봉준(bbongcol@gmail.com 010-2062-6008)
http://beagleandroid.springnote.com/

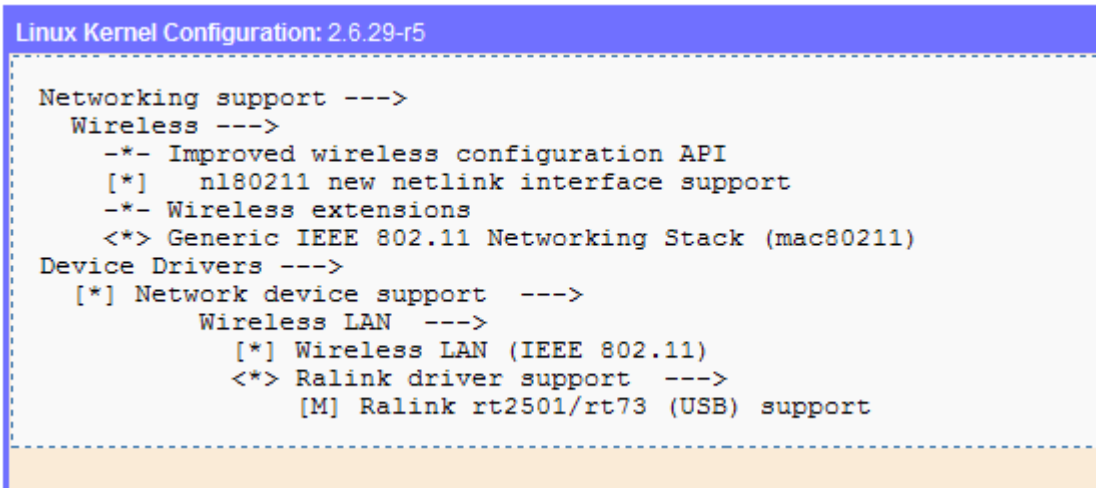
5. WiFi 드라이버 설치하기

비글보드 포팅 안드로이드에서 USB WiFi를 사용하기 위해서는 커널 빌드시 WiFi 드라이버를 설치해야 한다. 안드로이드의 설정-무선제어-WiFi에서 바로 인식이 안되는 이유는 구글에서 안드로이드를 배포시 안드로이드 협력 제조업체 칩셋만을 자동으로 인식하도록 해놓았기 때문이다.

현재 대부분의 USB Wifi는 Ralink사의 칩셋을 사용하고 있다 본 프로젝트에 사용한 USB WiFi 역시 Ralink사의 rt73칩셋을 사용한 제품이다. 이것을 인식하기 위해서는 리눅스 커널 빌드시 이 드라이버를 포함해 주어야 한다.

(1) 커널 설정

본 프로젝트에서는 0xdroid 안드로이드에서만 WiFi 인식에 성공했다. 따라서 아래 설명은 모두 0xdroid커널에서 진행되는 과정이다. 바로 위에서 커널 빌드시 make menuconfig를 입력하게 되면 커널 환경설정을 하는 부분이 나오는데 아래와 같이 Ralink driver support 부분을 찾아 모두 선택해 준다.



위 챕처 화면은 모듈로 설치 하는것인데 그냥 스페이스를 눌러 * 가 나오도록해 커널에 설치하도록한다. 리눅스 커널 빌드에 대해 많은 지식이 있다면 이 부분을 원하는 대로 설정한다.

(2) 드라이버 파일 생성, 안드로이드 이미지 만들기

이제 비글보드에 맞게 포팅해 관련 드라이버 파일을 생성해야 하는데 이 부분에 있어서는 직접적으로 성공하지 못했고 이미 만들어진 파일을 그대로 사용했다. 파일은 http://cid-877835fb9c4f0926.skydrive.live.com/self.aspx/%eb%ac%b8%ec%84%9c/util.zip 에서 구할 수 있다. 드리이버 포팅에 관해서는 http://www.aesop.or.kr 에서 wifi로 검색하면 관련 설명을 구할 수 있다.

이봉준(bbongcol@gmail.com 010-2062-6008)
http://beagleandroid.springnote.com/

파일을 다운로드해 모두 빌드가 끝난 안드로이드 루트 파일 시스템의 system/lib 폴더에 집어 넣는다. PATH 설정이 되어 있어 어디서든 실행가능하게 된다. 이제 ./generateubifs.sh 를 통해 이미지 파일을 만들고 이 파일로 부팅을 시작한다.

USB2.0 허브에 WiFi가 꼽혀 있다면 부팅과정에서 자동으로 칩셋을 인식하게 된다.

(3) 드라이버 파일 생성, 안드로이드 이미지 만들기

이제 이 wifi를 이용해 무선인터넷에 접속하기 위해서는 몇가지 설정을 해주어야 한다. minicom에서 아래 명령어를 실행한다.

```
ifconfig wlan2 inet 192.168.123.10 up
iwconfig wlan2 essid "bbongcol"
route add default gw 192.168.123.1 dev wlan2
setprop net.dns1 165.246.10.2
```

wlan2: 안드로이드에 잡힌 무선랜 이름, minicom에서 netcfg를 치면 나온다. 리눅스 커널 설정에 따라 wlan0이나 wlan1이 나올수도 있음

192.168.1423.10: 임의 부여한 안드로이드 IP주소, 192.168.123은 WIFI 호스트 주소에 따라 달라진다. 10은 임의 부여한 주소.

bbongcol: 무선인터넷의 essid 이름

192.168.123.1: 무선인터넷을 뿌리고 있는 WiFi의 주소이다. 컴퓨터에 WiFi를 연결하고 Ad-hock 모드로 실행하면 무선인터넷을 뿌리게 되는데 이때 자신의 IP 주소이다. 무선인터넷을 사용하는 안드로이드 입장에서는 이 주소가 기본 게이트웨이 주소가 되는것이다. 이 게이트웨이 주소에 따라 안드로이드 ip 주소를 임의 결정하면 된다. (끝자리면 바꿔서)

165.246.10.2: DNS 서버의 주소이다. 학교에서는 사용할 경우 학교 DNS 주소를 입력한다.

※위 명령어를 다 입력 후 **iwconfig wlan2**를 입력하면 현재 접속된 무선 WiFi의 정보가 나온다. WiFi의 맥 주소가 뜨지 않는다면 접속되지 않은 것이다.

※윈도우 비스타에 USB WiFi 만들어 무선인터넷을 뿌리면 안드로이드에서 WiFi를 잡아도 무선인터넷이 안 되는 현상이 있는듯하다. XP에서는 정상 실행된다. 윈7에서는 테스트 못해봄

※안드로이드의 설정-무선제어-WiFi에서 바로 인식이 안되는 이유는 구글에서 안드로이드를 배포시 안드로이드 협력 제조업체 칩셋만을 자동으로 인식하도록 해놓았기 때문이다.

(4) 안드로이드에서 WiFi 명령어를 실행 할 수 있는 환경 만들기

위와 같은 방법으로 WIFI를 이용해 인터넷에 연결하게 되면 minicom 시리얼 통신이 항상 필요하게 되어 상당히 불편하다. 안드로이드 자체에서 위 명령어를 실행할 수 있도록한다면 시리얼 통신이 필요 없게 되어 사용하기 편리하다.

안드로이드의 Terminal Emulator에서 각종 리눅스 명령어를 쓸수 있지만 기본적으로 root권한을 얻을 수 얻기 때문에 설정은 불가능하다. 따라서 안드로이드 상에서 root 권한을 얻기 위해 다음 절차를 따라야 한다.

http://www.magicandroidapps.com/su.zip

먼저 위에서 다운로드 받은 SU 파일을 안드로이드 루프 파일시스템의 system/xbin 폴더에 집어 넣는다. 그리고 init.rc 파일에 다음을 추가 해 준다.

chmod 6755 /system/xbin/su

중간에 chmod 설정 부분이 많이 있는데 그 중간에 추가해 주면 된다. 새롭게 안드로이드 이미지 파일을 만들고 안드로이드를 부팅한후 **더보기 – Dev Tools – Terminal Emulator** 에서 아래 명령어를 입력한다.

```
su
ifconfig wlan2 inet 192.168.123.10 up
iwconfig wlan2 essid "bbongcol"
route add default gw 192.168.123.1 dev wlan2
setprop net.dns1 165.246.10.2
```

여기에서 su는 root 권을을 얻기위한 명령어이다. 이제 안드로이드상에서 간단하게 wifi 연결 설정을 할 수 있다.

WiFi의 경우 데이터를 많이 다운로드 할 수 경우 USB 2.0 허브의 전력이 딸려 안드로이드에서 인식이 갑자기 안될 경우가 발생할수 있다. 따라서 비글보드와 USB 허브에 충분한 전원을 공급해 주어야 한다.

※참고 웹사이트

http://en.gentoo-wiki.com/wiki/Ralink_RT73

http://www.aesop.or.kr

http://cid-877835fb9c4f0926.skydrive.live.com/self.aspx/%eb%ac%b8%ec%84%9c/util.zip

이봉준(bbongcol@gmail.com 010-2062-6008)

http://beagleandroid.springnote.com/