



EZ-X5 Board 를 위한 블루투스 포팅 안내 v1.0

(Bluetooth Porting Guide for EZ-X5 Board v1.0)



저자 : 박계현 (Kye Hyun Park)
홈페이지 : <http://kyehyun.com>
전자메일 : kyehyun@gmail.com
MSN : kyehyun@gmail.com
ICQ# : 115-079-903

1. 포팅 환경

- 1) Board : EZ-X5 and EZ-PCMCIA (Xscale PXA255)
- 2) OS : Linux 2.4.19 or Higher (patched)
- 3) Bluetooth Card : Ambicom BT2000E CF Card
(Model : BT2000-CF)
- 4) Bluetooth Stack : BlueZ (<http://www.bluez.org>)
- 5) Bluetooth Driver : BT2000E_CS Linux Driver 0.31



2. BlueZ - Kernel Patch

- 1) <http://www.holtmann.org/linux/kernel> 에서 Linux Kernel Version에 맞는 Bluetooth Patch를 구합니다. 저의 Kernel Version은 2.4.19 이므로, patch-2.4.18-mh18.gz 을 받았습니다.

<http://www.holtmann.org/linux/kernel/patch-2.4.19-mh18.gz>

- 2) Board Level 포팅이 적용 된 커널 소스가 들어있는 디렉토리로 이동하여, 다음과 같이 패치 작업을 시작합니다. 작업 종료 후 커널 컴파일 설정에 들어가서 Bluetooth 항목에 들어가서 HCI, L2CAP, RFCOMM, ... builtin 으로 선택 후 커널 컴파일을 시작합니다.

```
# gunzip patch-2.4.18-mh18.gz  
# patch -p0 < patch-2.4.18-mh18  
# cd linux-2.4.19
```

```
# make menuconfig
# make dep && make clean && make zImage
```

3) 생성된 커널 이미지를 타겟으로 푸징을 합니다.

3. BlueZ – Libs, Utils

1) <http://www.bluez.org/download.html> 에서 Libs와 Utils Sources를 구합니다.

```
http://bluez.sf.net/download/bluez-libs-2.13.tar.gz
http://bluez.sf.net/download/bluez-utils-2.13.tar.gz
```

2) BlueZ Libs 는 BlueZ Utils은 물론 향후 Bluetooth 어플리케이션 개발을 하기 위하여 ARM Core로 컴파일되어야 합니다.

```
# tar xvfz bluez-libs-2.13.tar.gz
# cd bluez-libs-2.13
# ./configure --prefix=/usr/armv5l-linux --build=i686-linux --host=armv5l-linux W
--libdir=/usr/armv5l-linux/lib --includedir=/usr/armv5l-linux/include W
CC=armv5l-linux-gcc AR=armv5l-linux-ar RANLIB=armv5l-linux-ranlib
# make
# make install
```

3) BlueZ Libs 컴파일 후 생성된 라이브러리 파일을 타겟의 /lib 으로 복사합니다.

```
libbluetooth.so.1.0.11
libbluetooth.so.1 -> libbluetooth.so.1.0.11
libbluetooth.a
```

4) BlueZ Utils 는 Bluetooth 장치의 HCI, L2CAP, RFCOMM 레이어를 위한 도구들이 제공됩니다.

```
# tar xvfz bluez-utils-2.13.tar.gz
# cd bluez-utils-2.13
# ./configure --prefix=$PWD/build --build=i686-linux --host=armv5l-linux W
--libdir=/usr/armv5l-linux/lib --includedir=/usr/armv5l-linux/include W
CC=armv5l-linux-gcc AR=armv5l-linux-ar RANLIB=armv5l-linux-ranlib
# make
# make install
# cd build
```

5) BlueZ Utils 작업 후 생성된 build 디렉토리의 etc/bluetooth 디렉토리를 타겟의 /etc 으로 복사합니다. 그리고, bin/* sbin/* 디렉토리 내 파일을 타겟의 /bin /sbin 으로 복사합니다.

4. Bluetooth Driver

1) <http://www.iral.com/~albertr/linux/bt2000e/> 에서 BT2000E_CS Linux Driver 0.31 을 구합니다.

```
http://www.iral.com/~albertr/linux/bt2000e/bt2000e\_cs-0.31.tar.gz
```

2) BT2000E_CS Linux Driver 0.31 압축을 풀고, 컴파일을 합니다.

```
# tar xvfz bt2000e_cs-0.31.tar.gz
# cd bt2000e_cs-0.31
# armv5l-linux-gcc -c bt2000e_cs.c -O2 -Wall -D__KERNEL__ -DMODULE W
-l/root/work/toolchain/linux-2.4.19/include
```

- 3) 생성된 bt2000e_cs.o 드라이버 파일을 타겟의 /lib/modules/`uname -r`/ 에 복사합니다.
- 4) 타겟의 /etc/pcmcia 디렉토리에 bt2000e_cs.conf 파일을 생성합니다.

```
device "bt2000e_cs"
    class "bluetooth" module "bt2000e_cs"

card "AmbiCom BT2000E Bluetooth Card"
    version "AmbiCom,Inc", "BT2000E"
    bind "bt2000e_cs"

card "Pretec BT2000E Bluetooth Card"
    version "Pretec", "BT2000E"
    bind "bt2000e_cs"
```

5. EZ-PCMCIA 보드에 Ambicom BT2000E CF Card를 장착합니다.

- 1) 타겟을 reset 을 한 후 부팅 중에 Device를 검색 메시지를 나타냅니다.

```
bt2000e_cs 0.31 (Albert Rybalkin <albertr@iral.com>)
bt2000e_cs: requested baud rate = 460800 bps
bt2000e_cs: hci0: index 0x01: Vcc 3.3, irq 36, io 0xf60002f8-0xf60002f
```

- 2) hciconfig 명령어로 hci0 장치를 열어주고 초기화 해줍니다.

```
TARGET # hciconfig hci0 up

bt2000e_cs: hci_reset
bt2000e_cs: new uart clock divisor = 1
bt2000e_cs: unhandled RLSI interrupt, LSR=0xF9
bt2000e_cs: received bogus packet: type=0x00 size=1
```

- 3) hciconfig 명령어로 hci0 장치의 정보를 얻을 수 있습니다.

```
TARGET # hciconfig

hci0: Type: PCCARD
BD Address: 00:10:7A:4C:41:53 ACL MTU: 672:8 SCO MTU: 64:0
UP RUNNING PSCAN ISCAN
RX bytes:78 acl:0 sco:0 events:8 errors:1
TX bytes:43 acl:0 sco:0 commands:8 errors:
```

- 4) hcitool을 이용하여 piconet에 있는 Bluetooth Device를 Inquiry 를 해봅니다.

```
TARGET # hcitool inq

Inquiring ...
00:02:78:00:DD:DF      clock offset: 0x7cc7      class: 0x120104
00:02:78:00:DD:E3      clock offset: 0x5f7b      class: 0x120104
00:02:78:00:CA:8E      clock offset: 0x5b9a      class: 0x12010
```

- 5) l2ping을 이용하여 장치간의 ACL 링크 연결이 이루어지는지 확인이 가능합니다.

```
TARGET # l2ping 00:02:78:00:DD:E3
```

```
Ping: 00:02:78:00:DD:E3 from 00:10:7A:4C:41:53 (data size 20) ...
20 bytes from 00:02:78:00:DD:E3 id 0 time 91.81ms
20 bytes from 00:02:78:00:DD:E3 id 1 time 24.72ms
20 bytes from 00:02:78:00:DD:E3 id 2 time 27.19ms
20 bytes from 00:02:78:00:DD:E3 id 3 time 29.68ms
20 bytes from 00:02:78:00:DD:E3 id 4 time 22.09m
```

6. 이제 부터는 /lib 디렉토리에 있는 libbluetooth를 이용하여 Bluetooth Applicaiton 을 작성하여 타겟에서 올려보도록 합니다.