

# 리눅스 기반의 유연한 블루투스 프로토콜 스택 개발

신기수, 조철수, 박장식, 김현태  
프리눅스(주), 울산기능대학, 동의공업대학, 동의대학교

## Development of Flexible Bluetooth Protocol Stack Based on Linux

Gi Soo Shin, Cheol Soo Cho, Jang Sik Park, Hyun Tae Kim  
Freenux Corp, Dept. of Information and Communication, Ulsan Polytechnic College  
Dept. of Visual Technologies, Dong-Eui Institute of Technology  
Dept. of Multimedia Eng., Dong-Eui University  
E-mail : think@freenux.net

### 요 약

본 논문에서는 리눅스상에서 블루투스 프로토콜 스택을 구현하는 방법에 대하여 제안한다. 리눅스는 Unix와 동일한 파일 시스템 구조와 시스템 접근 방법을 사용한다. 리눅스에서는 여러 가지 장치에 대해 Device class라는 구조를 가지고 접근하며, Unix시스템과 같이 각각의 디바이스들은 Block 또는 Character device file이라는 형태로 구현되고, 네트워크 인터페이스와 기타 장치들은 특정한 프로토콜 등으로 구현 및 접근을 하게 된다. 리눅스 시스템에서 기존 및 특정한 응용프로그램들이 블루투스 장치를 사용하도록 하려면 블루투스 프로토콜 스택을 디바이스 드라이버 형태로 구현하여야 하며, 리눅스에서는 이러한 디바이스 드라이버들을 커널 내에 적재, 혹은 비적재에 따라 커널 내에서 구현하거나 모듈의 형태로 구현하여야 한다. 본 논문에서는 리눅스에서의 디바이스 드라이버를 구현방법을 제안한다. 커널과 모듈을 혼합한 블루투스 프로토콜 스택의 세부적인 구현 방법과 특수한 목적에 쉽게 적용이 가능한 유연한 블루투스 프로토콜 스택을 제안한다.

### 1. 서론

블루투스 기술은 작고, 저렴한 가격, 100mW이하의 저전력 소모로 근거리 송수신기를 PDA, 노트북, 휴대폰 등 이동통신 단말기에 직접 또는 PC카드 등과 같은 어댑터를 통해 탑재되어 간접적으로 고속의 무선 환경을 제공해 주는 하나의 기술적인 규격 사양이다. 무선 환경은 세계적으로 이용 가능한 전역 주파수대역(ISM Band : 비 상업적 주파수대역)인 2.45 GHz 밴드를 이용하고 721 kbps의 데이터 전송 속도와 3개의 음성 채널을 지원한다. 또한, 블루투스는 전력 소모량이 30  $\mu$ A 인 대기모드에서부터 3 ~ 30 mA 범위의 전송량이 많은 장치에 이르기까지 다양한 제품들 대상으로 하고 있다[1, 2].

도달 거리 측면에서 블루투스는 사무실, 회의실 및 가정과 사용자의 주변공간 내에서 지원하도록 개발되었다. 블루투스 장치는 사용요구에 따라 주변장치들과 10m 반경 내에서 정보 교환 능력을 갖는다. 그리고 Data Access Point로 +20dB 정도의 전송단과 -90dB 정도의 수신단을 사용한다면 개방된 공간에서 100m 까지 도달할 수 있다[2].

블루투스 기술은 초기에 제안되었던 바와 달리 저 소비전력과 저가격의 실현성 및 기타 특성으로 인해 실제의 적용에 있어서는 아직 많은 과제를 가지고 있다. 이러한 이유로 아직은 많은 부분에 적용이 되어지고 있지 않다. 현재는 주로 PC 및 PDA 등 휴대 통신 기기를 연결하는 근거리 무선네트워크를 제공하는 LAP(LAN Access Profile)와 휴대폰과 주변기기간의 근거리 무선연결을 지원하기 위한 CTP(Codeless

Telephony Profile) 등이 적용되어지는 대부분이다[3].

휴대폰과 주변기기를 연결하는 기술은 많이 개발되어 있으나, 휴대통신기와 주변기기의 근거리 무선연결을 지원하는 기술의 개발은 미약하다. 가까운 미래에는 휴대폰과 주변기기의 근거리 무선네트워크 보다는 휴대통신기와 주변기기의 근거리 무선 네트워크로의 연결이 더욱 중대될 것이다. 리눅스는 커널 소스로부터 자신의 환경에 가장 최적화된 운영체제로 구성할 수 있어 안정된 시스템을 구성할 수 있다. 그리고 다양한 파일 시스템과 인터넷 기본 프로토콜인 TCP/IP 외에 SLIP/PPP, NNTP, SMTP, IPX 등 현존하는 거의 대부분의 네트워크 프로토콜을 지원한다[4].

본 논문에서는 휴대통신기와 주변기기의 근거리 무선네트워크 구현을 위하여 리눅스기반의 블루투스 프로토콜 스택인 BlueFree를 제안한다.

BlueFree는 크게 하드웨어 의존적인 부분과 소프트웨어 부분으로 구성되어 있으며, 전체는 4가지로 나뉘어 있다. RF부는 하드웨어 부분으로 실제 물리적인 무선장치이며 장착 기기와의 접속부위를 제공한다. 소프트웨어 부분의 HCI Core는 크게 Upper 부분과 Lower 부분으로 나눌 수 있다. BlueFree의 Upper Protocol Stack은 SDP(Service Discovery Application Profile), LAP, FTP 및 SPP(Serial Port Profile)와 같은 프로파일을 지원한다. BlueFree가 특수한 목적에 쉽게 적용이 가능한 유연한 블루투스 프로토콜 스택임을 확인한다.

## 2. 리눅스의 디바이스 드라이브 구현 방법

리눅스에서 각각의 장치들은 디바이스 드라이브를 통해서 동작된다. 리눅스에서의 디바이스 드라이브들은 커널 내에서 디바이스 드라이브로 구현되기도 하고, 커널 밖에서 모듈의 형태로 구현되기도 한다. 또한, 각각의 디바이스들은 그 장치의 특성에 따라 보통 Block device와 Character device로 구별하고, 특성이 다른 네트워크 인터페이스나 일부의 장치들은 특정한 프로토콜의 형태로 구현된다. 그리고 시스템 내에서 각각의 장치는 Device file이라는 File system의 일부로 구현된다[4].

이러한 방법으로 리눅스상에서 블루투스 프로토콜 스택을 구현하려면 커널에는 스택의 Core를 블루투스의 Daemon형태로 구현하고 커널에 적재 가능한 모듈

에는 각각의 기능적인 부분들이 구현되어야 한다. 그렇게 하여 응용프로그램에서 블루투스 장치를 호출하게 되면 초기의 Daemon에서 호출을 받고 사용 가능한 기능 모듈들을 커널에 적재하여 동작되도록 준비하게 된다. 블루투스 프로토콜 스택의 구조는 그림 1과 같이 물리층을 규정하는 RF, Hopping pattern 등을 규정하는 Baseband, 패킷의 구성 등을 규정하는 LMP(Link Manager Protocol), L2CAP(Logical Link Control and Adaptation Protocol)와 상위의 호스트 시스템간의 인터페이스를 규정하는 HID와 RFCOMM의 부분으로 나누어진다.

Baseband의 상위에서 프레임을 구성하거나, Error correction, Authentication, Encryption 등을 정의하고 있다. 또한, TCP/IP의 프로토콜 스택 등은 L2CAP의 상위에 실장하게 된다. 호스트와의 인터페이스로서는 USB, EIA-232가 탑재되어 있다.

블루투스 프로토콜 스택의 구현에 있어 핵심은 Core에 해당하는 HCI(Host Controller Interface)에 있다. 여기서 HCI는 Baseband와 LM을 통한 하드웨어 상태에 대한 Monitoring 및 Register들에 대한 Command interface 역할을 한다. HCI의 동작 형태를 보면 Command의 전송과 그에 대한 결과인 Event를 수신하는 형태가 된다. 이는 하나의 Command에 대한 개 혹은 다수 개의 Event를 돌려 받을 수 있다. 이는 블루투스에 적용된 네트워크 방식인 Single slave, Pico net 및 Scatter net에 기인한다.

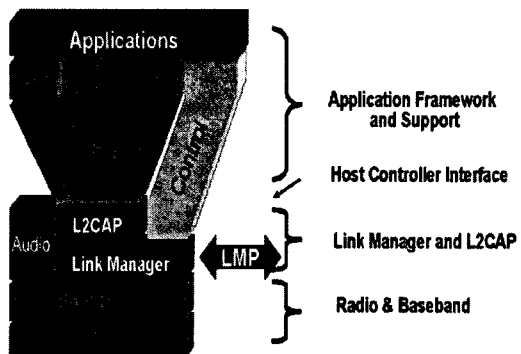


그림 1. 블루투스 프로토콜 스택의 구조

## 3. 리눅스 블루투스 프로토콜 스택 구현

본 논문에 기술하는 리눅스에 기반한 유연한 프로토콜 스택인 BlueFree의 구조는 그림 2와 같다. BlueFree는 크게 하드웨어 의존적인 부분과 응용 프로그램에 연결된 부분으로 나눌 수 있으며, 전체는 4가지로 나뉘어 있다. 여기서 RF부는 하드웨어 부분으로 실제 물리적인 무선장치이며 장착기기와의 접속부위를 제공한다.

소프트웨어적인 관점에서 볼 때 블루투스 프로토콜의 가장 핵심적인 부분은 HCI Core이다. BlueFree의 HCI Core는 크게 Upper 부분과 Lower 부분으로 나눌 수 있다. 하드웨어에 기반한 LMP가 Lower HCI Core내의 Host controller와 같은 계층에 위치하고, Upper HCI Core에는 물리적인 연결제어를 위한 ACL (Asynchronous Connection Less) Link 계층과 SCO(Synchronous Connection Oriented) Link 계층이 위치한다.

일반적으로 LMP는 시큐리티의 역할을 담당하며 링크나 암호키의 생성, 교환 체크와 베이스밴드의 패킷사이즈 제어를 한다. 그리고 파워모드, 블루투스 무선 디바이스의 사이클, 피코넷의 접속상태를 제어하는 등의 역할을 한다.

ACL과 SCO계층들은 블루투스에 있어 장치간 접속 형태가 정의된 부분이다. ACL 링크는 비동기 링크로써 데이터 전송시의 에러 보정 등의 기능이 있어 데이터 통신을 위한 링크로써 이용이 된다.

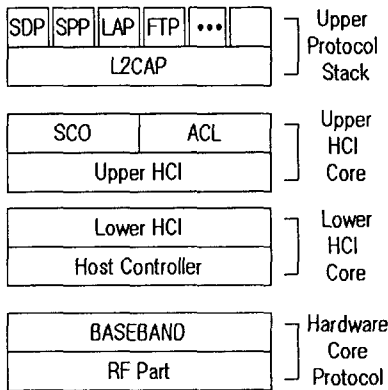


그림 2. BlueFree의 구조

SCO 링크는 동기링크이며 통신속도의 보장이라는 장점이 있는 반면 데이터 보정과 같은 기능은 없다. 3가지 형태의 음성 통신 혹은 비동기 데이터와 동기의 음성 통신이 가능하다.

ACL과 SCO계층의 상위계층에 L2CAP가 위치하여

Multiplexing과 Packet exchange 등의 기능을 수행하게 된다. SDP, RFCOMM, TCP/IP 등의 계층들은 특정한 목적을 이루기 위한 기능 계층들로써 임의 목적에 맞게 정의된 프로파일에 따라 추가할 수 있다. L2CAP은 베이스 밴드의 상위에 위치하며 상위 프로토콜 스택에 대하여 다중통신 기능, 데이터의 분할과 조합, 그룹의 추상화 역할을 담당하는 서비스를 제공한다. L2CAP에 의해 상위의 프로토콜 및 어플리케이션은 64 kByte까지의 길이를 가지는 L2CAP 패킷 데이터를 송수신하는 것이 가능하다.

SDP는 블루투스의 이용용도를 미리 결정하는 서비스를 제공하고 중요한 역할을 담당한다. SDP를 이용함으로써 각각의 기기가 가진 정보, 서비스의 내용을 파악하여 기기간의 접속을 할 수 있게 된다.

BlueFree에서는 SDP(Service Discovery Application Profile), LAP, FTP 및 SPP(Serial Port Profile)와 같은 프로파일을 지원한다.

현재 1.0b규격에서는 13 가지의 프로파일이 규정되어 있고, 그것을 응용 프로그램 적용의 관점에서 구분하면 표 1과 같다. BlueFree에서 적용하고 있는 유연한 구조의 프로토콜 스택이라는 부분은 이러한 프로파일을 기존보다 손쉽게 적용이 가능한 부분을 의미하는 것이다. 이는 블루투스 본래의 구성 방법과 같다고 할 수 있다. 핵심인 HCI Core는 항상 동작 중이어야 하며 이는 리눅스의 커널에 대응되고, 개개의 프로파일에 해당하는 것은 모듈화된 디바이스 드라이브에 해당한다.

표 1. 1.0b규격에서 13 가지 프로파일의 분류

Profile Type	Profile
공통기능 Profile	- Generic Access Service Discovery Application
전화기능 Profile	- Cordless Telephony - Intercom
Serial 연결기능 Profile	- Serial Port - Dial-Up Network - FAX - Headset - LAN Access
OBEX 기능 Profile	- Serial Port
	- Generic Object Exchange
	- File Transfer
	- Object Push
	- Synchronization

```

lab# hciconfig
hci0: Type: UART
BD Address: 00:00:07:83:48:F0 BCL MTU: 672:10 SCO: MTU 0:0
UP RUNNING NORMAL PSCHAN ISCAN
RX bytes:62 acl:0 sco:0 events:7 errors:0
TX bytes:36 acl:0 sco:0 commands:7 errors:0

hci1: Type: UART
BD Address: 00:00:07:83:48:F0 BCL MTU: 600:10 SCO: MTU 0:0
UP RUNNING NORMAL PSCHAN ISCAN
RX bytes:62 acl:0 sco:0 events:7 errors:0
TX bytes:36 acl:0 sco:0 commands:7 errors:0

lab# lsmod
Module                Size  Used by
hci_uart              4056  2 (autoclean)
hci                   2256  3 (autoclean) [hci_uart]

lab# /12cap ping 00:00:07:83:48:F0
Ping: 00:00:07:83:48:F0 from 00:00:07:83:48:F0 (data size 20) ...
20 bytes from 00:00:07:83:48:F0 id 200 time 40.91ms
20 bytes from 00:00:07:83:48:F0 id 201 time 50.02ms
2 sent, 2 received, 0% loss

lab# /usr/src/bluefree/tools5 ./12cap_test -r to 10 -l 2300 &
[1] 22761
12cap_test[22761]: Waiting for connection on psm 10 ...

lab# /usr/src/bluefree/tools5 ./12cap_test -s 00:00:07:83:48:F0
12cap_test[22762]: Connected [00:00:07:83:48:F0, Psm: 0 55525]
12cap_test[22762]: Sending ...
12cap_test[22762]: Connect from 00:00:07:83:48:F0 [intu 2000, ortu 6]
12cap_test[22762]: Receiving ...
12cap_test[22762]: 10240 bytes in 0.01s speed 11.12 kb
12cap_test[22762]: 10240 bytes in 0.01s speed 11.12 kb
12cap_test[22762]: 10240 bytes in 0.02s speed 11.00 kb
12cap_test[22762]: 10240 bytes in 0.01s speed 11.12 kb
12cap_test[22762]: 10240 bytes in 0.01s speed 11.12 kb
12cap_test[22762]: 10240 bytes in 0.01s speed 11.12 kb
12cap_test[22762]: Read Failed. Connection reset by peer(104)
12cap_test[22762]: Disconnect
    
```

그림 3. BlueFree의 Local 및 Remote 테스트 결과

시스템의 커널은 항상 동작중이고 임의의 기능을 사용자 프로그램이 요구할 때, 그 기능에 해당하는 Device module을 커널에 적재, 실행하는 것과 동작의 구조가 같음을 알 수 있다.

그림 3은 리눅스에서 GNU gcc를 이용하여 개발한 결과이다. Local 테스트 및 Lab/Remote 테스트를 통하여 BlueFree 프로토콜이 제대로 동작하고 있는 상태를 보여준다. hciconfig는 호스트 설정 및 모니터를 위한 기능으로 미리 설정되어진 주소를 볼 수 있으며 통신방식을 볼 수 있다. lsmod 는 현재 커널 내에서 로드되어 있는 모듈과 상태를 보여준다. 12cap\_ping 은 가장 기본적인 패킷 전송 테스트의 하나인 Ping 테스트를 할 수 있다. 그리고 12cap\_test 는 원격지 시스템과 패킷 데이터 전송 테스트 기능을 갖고 있다.

#### 4. 결론

블루투스 기술은 작고, 저렴한 가격, 저전력 소모로 근거리 송수신기를 이동통신 단말기에 직접 또는 PC 카드 등과 같은 어댑터를 통해 탑재되어 고속의 무선 환경을 제공해 주는 하나의 기술적인 규격 사양이다. 리눅스는 커널 소스로부터 자신의 환경에 가장 최적화된 운영체제로 구성할 수 있어 안정된 시스템을 구성할 수 있고, 다양한 파일 시스템과 인터넷 기본 프로토콜인 TCP/IP외에 SLIP/PPP, NNTP, SMTP, IPX 등 현존하는 거의 대부분의 네트워크 프로토콜을 지원한다.

본 논문에서는 휴대통신기와 주변기기의 근거리 무선네트워크 구현을 위하여 리눅스기반의 블루투스 프로토콜 스택인 BlueFree를 제안하였고, 본사와 대학이 공동 개발하였다. 또한, BlueFree가 특수한 목적에 쉽게 적용이 가능한 유연한 블루투스 프로토콜 스택임을 보여주었다. 향후 연구 및 개발이 필요한 부분은 데이터량이 많은 영상 신호 등을 전송하기 위하여 FDTP(Fast Data Transfer Profile)이며, 휴대전화, PDA 등의 휴대단말기와 PC 등이 데이터를 주고 받는 등의 사용 모델을 제공하는 어플리케이션에서 사용되는 프로토콜인 SP(Synchronization Profile)를 개발하고자 한다. FDTP 및 SP는 개발 예정인 정보 시각장치의 기능 구현을 위한 부분이며 차기 블루투스 2.0 버전에서 고려되고 있는 프로파일 중의 하나이다.

#### 참고 문헌

- [1] Jennifer Bray and Charles F Sturman, *Bluetooth: Connect Without Cables*, Prentice Hall, 2001.
- [2] *Specification of the Bluetooth System: Core Version 1.1*, Bluetooth SIG, 2001.
- [3] *Specification of the Bluetooth System: Profiles Version 1.1*, Bluetooth SIG, 2001.
- [4] 이민호, 유창완, 전정훈, 최재형, 이동철, "리눅스 기반 다기능 보안 게이트웨이와 원격 관리 시스템의 설계 및 구현", NCS 2001, pp. 233-237, 2001년 12월.