

# 블루투스 프로토콜 스택을 위한 HCI의 구현

류수형<sup>0</sup> 김 식 현무용 이상윤  
세명대학교 대학원 전산정보학과, 세명대학교 정보통신학과  
대원과학대학 컴퓨터정보통신과, 대원과학대학 컴퓨터정보처리과  
{shliu,shikm}@telcom.semyung.ac.kr, {myhyun,sylee}@daewon.ac.kr

## Host Controller Interface Implementation for Bluetooth Protocol Stack

Soohyung Liu<sup>0</sup> Shik Kim Muyoung Hyun Sangyun Lee  
Dept. of Computer Information Science, Semyung University  
Dept. of Information Communication, Semyung University  
Dept. of Computer Information Communication, Daewon College  
Dept. of Computer Information System, Daewon College

### 요 약

블루투스는 차세대 근거리 무선통신 기술로 각광받고 있으며, 프로토콜 스택의 개발을 통한 다양한 서비스제공을 필요로 한다. 프로토콜 스택의 상위 계층과 각종 프로파일은 유일한 것이 아니고, 응용 목적에 따라 다양한 형태로 개발될 수 있으며, 이러한 프로토콜 스택에 대한 개발은 향후 지속적으로 개발될 블루투스 어플리케이션에 대한 원천기술을 확보하는 의미를 가진다. 이 논문에서는 블루투스 프로토콜 스택을 개발하기 위한 첫 단계로 블루투스 모듈과 호스트사이의 인터페이스인 HCI를 구현했고, 일련의 실험을 통해 블루투스 장치검색과 연결 설정 및 데이터 송수신 동작들이 성공적으로 수행되는 것을 확인하였다.

### 1. 서 론

지난 수년동안 세계적으로 휴대용 장치가 급속히 보급됨에 따라 노트북PC등과 연결한 모바일 컴퓨팅(mobile computing)이 활발하게 진행되어 왔다. 블루투스는 이러한 각종 전자장치 간의 통신에 물리적인 케이블 없이 무선주파수를 이용하여, 고속으로 데이터를 주고받을 수 있는 무선통신 기술이며, 블루투스 프로토콜 스택의 표준규격을 책정하기 위한 목적으로 1998년 에릭슨, 노키아, IBM, 도시바, 인텔 5개 사가 중심 멤버가 되어 Bluetooth SIG(Special Interest Group)라는 컨소시엄을 구성했고, 현재는 회원사가 전세계적으로 2000여 개에 이를 만큼 블루투스 기술에 대한 업계의 관심이 높아지고 있다[1].

이처럼 차세대 근거리 무선통신 기술로 많은 호응을 불러일으키고 있는 블루투스 기술은 그 응용 범위가 매우 넓기 때문에 블루투스 프로토콜 스택을 응용한 각종 어플리케이션 개발의 필요성은 점점 확대되고 있다. 현재 블루투스 프로토콜은 1.1버전의 규격이 나와있고 이것은 물리적인 부분으로 펌웨어를 기술한 코어(Core)사양과 상호 기기 간의 호환성을 위한 프로파일로 나뉘어져 있다. 프로파일은 블루투스를 최상위 어플리케이션에서 어떻게 사용할지를 정해놓은 규격으로 유일한 것은 아니며, 응용목적에 따라 다양한 형태로 개발해서 사용하게 된다[1][2]. 이러한 프로파일개발에 적합한 상위계층 프로토콜을 개발하기 위해서는 하드웨어와 소프트웨어의 중간 계층으로, 블루투스 장치와 호스트 사이의 인터페이스 역할을 하는 프로토콜인 HCI 계층의 구현이 필수적이다. 따라서 이 논문에서는 블루투스 프로토콜 스택을 개발하기 위한 첫 단계 연구과제로서 HCI(Host Controller Interface)를 구현하였으며, 일련의 실험을 통해 본 연구에서 구현된 HCI가 정상적으로 동작함을 확인하였다.

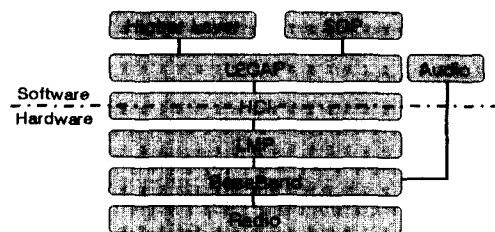
<sup>0</sup>본 연구는 (주)진두네트웍스와 의 연구과제로 수행되었음.

### 2. 블루투스 프로토콜 스택

이 장에서는 먼저 블루투스 프로토콜 스택에 대한 기본구조를 통해서 HCI 계층이 상·하 위 프로토콜들과 어떤 관계에 있는지와 각 프로토콜 계층의 역할을 살펴보고, HCI의 구현에 필요한 구조와 동작에 대해서 논의하도록 한다.

#### 2.1 기본 구조

블루투스 프로토콜 스택의 구조는 [그림 1]과 같다.



[그림 1] 블루투스 프로토콜 스택

최하위 계층인 Radio부터 Baseband, LMP(Link Manager Protocol)계층과 HCI 계층의 일부는 블루투스 시스템의 하드웨어 및 펌웨어를 구성하는 모듈로서 장치의 특성을 정의하고, 물리적 무선 연결과 생성을 관리하며, 불필요한 신호들에 대한 필터링기능 등을 제공한다. 블루투스 프로토콜 스택에서 하드웨어 계층과 소프트웨어 계층으로 양분되어 있는 HCI 계층은 블루투스 장치와 호스트 사이의 인터페이스 역할을 하는 계층으로 전송되는 패킷들의 타입을 정의하고 있다. L2CAP(Logical Link Control Adaptation Layer)은 프로토콜 종류를 구분하기 위한 멀티플렉싱과 송수신 데이터에 대한 패킷분할 및 재조합 그리고 QoS(Quality Of Service)정보의 전달을 목적으로 한다. SDP(Service Discovery Protocol)는 블루투스

장치들 상호간에 제공하는 서비스와 속성을 알아내는 역할을 하며, Audio는 BaseBand 계층을 통해 직접 음성데이터를 송수신 한다. 그 밖의 상위 프로토콜 계층들은 블루투스 장치의 응용과 관련되는 프로토콜들이다[5][16].

2.2 HCI 계층의 구조와 동작

HCI 계층은 블루투스 하드웨어 부분과 호스트의 소프트웨어 부분으로 나뉘어져 있다. 따라서, 블루투스 하드웨어는 물리적인 버스(USB, RS232C, UART 등)를 통하여 호스트 시스템과 패킷교환을 수행하게 된다. 교환되는 패킷은 호스트 시스템에서 블루투스 하드웨어로 보내어지는 Command 패킷, 송신 데이터 패킷 그리고 반대 방향으로 전달되는 Event 패킷, 수신 데이터 패킷의 네 가지이다. 이 때 호스트 시스템으로 전달되는 Event와 수신 데이터 패킷은 비동기 적으로 전달되며, 특히 데이터 전달은 표준규약에 따른 일련의 흐름제어 메커니즘을 필요로 한다[3][5]. 다음은 흐름제어 및 각 패킷의 구성과 구현을 위한 고려 사항이다.

- 1) Command 패킷은 총 6종류 95가지로 구성 되어있으며, 구현을 위해서는 모든 Command를 HCI의 표준규약에 정해진 형태로 구성하여 버스 드라이버로 전달할 수 있는 기능이 필요하다.
- 2) Event 패킷은 총 32가지이며, 구현을 위해서는 버스 드라이버로부터 비동기 적으로 전달되는 모든 Event 패킷을 받은 후, 프로토콜 스택의 상위계층으로 전달하는 기능이 필요하다. 이때 Event 패킷은 반드시 해석(Decode)된 것이어야 한다.
- 3) 송·수신 데이터 패킷에 대한 구현에는 전송할 데이터를 HCI 데이터 패킷의 표준규약 형태로 구성한 후, 버스 드라이버로 전달 할 수 있어야 하고, 버스 드라이버로부터 비동기 적으로 전달되는 모든 HCI 데이터 패킷을 받은 후, 프로토콜 스택의 상위계층으로 전달하는 기능이 필요하다. 이때, 전달받은 HCI 데이터 패킷들은 상위 계층의 요구에 맞게 재구성되어야 한다.
- 4) 표준규약에 따른 흐름제어는 처음 데이터를 전송할 때 호스트 측에서 최초로 Read\_Buffer\_Size Command를 이용하여 전송버퍼의 크기를 구한 후, 버퍼크기의 범위 내에서 데이터 패킷을 전송하게 된다. 이후 전달된 패킷에 의해서 Number\_Of\_Completed\_Packet Event가 발생하고, 이 Event정보를 통해 추가로 전달할 수 있는 패킷의 개수를 갱신하게 된다. 따라서 흐름제어의 구현을 위해서는 데이터 패킷을 보내는 스레드와 이에 상응하는 Event를 받는 스레드를 독립적으로 구현해야 하며, 이들 사이의 정보전달을 위한 대책이 마련되어야 한다.

3. 구현

HCI의 구현은 객체지향설계가 갖는 장점을 갖추기 위해 C++언어로 구현하였다. 구현된 HCI의 자료구조는 HCL\_Command 클래스, HCL\_Event 클래스, Pipe 클래스, HCL\_Stack 클래스로 구성하였으며, 각 자료구조는 블루투스 프로토콜 스택 1.1 규정에 따라 설계하였다. 다음은 구현된 각 클래스에 대한 정의와 Command전송에 필요한 동작 메커니즘 그리고 데이터 송수신을 위한 흐름제어에 대해 논의하도록 한다.

3.1 구현을 위한 자료구조

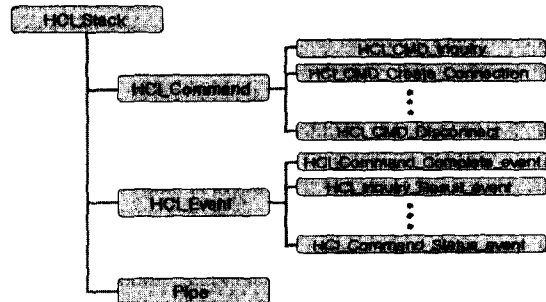
HCI 자료구조의 클래스 계층도는 [그림 2]와 같으며, 각 클래스에 대한 정의는 다음과 같다.

**HCL\_Command 클래스 :** Command의 일반적인 형식을 정의한 클래스이며, 하위 클래스로 HCI에 규정된 95개의 Command클래스 정의와 멤버함수로 AssembleCommand함수를 등록하여 각 Command패킷들의 인스턴스를 구현하였다.

**HCL\_Event 클래스 :** 블루투스 장치로부터 발생하는 Event코드의 일반적인 형식을 정의한 클래스이다. 하위 클래스로 HCI에 규정된 32개의 Event클래스 정의와 멤버함수로 MakeEvent함수를 등록하여 각 Event패킷들의 인스턴스를 구현하였다.

**Pipe 클래스 :** 호스트와 블루투스 장치간의 Command 전송과 데이터 송수신을 위한 스레드의 유기적인 동작과 흐름제어를 위해 사용되는 IPC(Inter-Process Communication) 메커니즘 중의 하나인 파이프(Pipe)를 구현한 클래스이다.

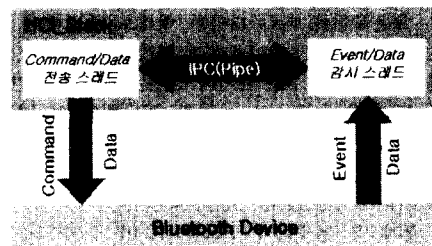
**HCL\_Stack 클래스 :** 위 클래스들을 멤버함수로 사용하였으며, Command실행에 따라 발생한 다양한 Event에 대해서 콜백함수를 등록하여 호스트가 Event응답을 받을 수 있게 했고 흐름제어를 통해서 데이터 패킷을 송·수신하도록 구현하였다.



[그림 2] HCI의 클래스 계층도

3.2 Event기반의 스레드 동작 메커니즘

블루투스 프로토콜 표준규약에 따르면 Command의 실행에 대한 결과로 다양한 Event가 비동기 적으로 발생한다는 것을 알 수 있다. 따라서, 발생하는 Event들에 대한 비동기 적인 처리메커니즘이 필수적이다. 이 논문에서는 비동기 적으로 발생하는 다양한 Event들을 처리하기 위해 [그림 3]과 같은 Event기반의 스레드 동작 메커니즘을 제안하였다[3][4].



[그림 3] Event기반의 스레드 동작 메커니즘

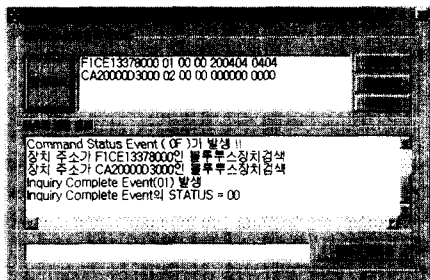
\*본 연구는 (주)진두네트웍사의 연구과제로 수행되었음.

이 메커니즘은 다중 스레드 환경에서 발생할 수 있는 스레드 동기화에 관련된 문제를 해결하기 위한 것으로, Event를 감시하는 스레드는 블루투스 장치로부터 Event가 발생했을 경우에 대한 정보를 파이프 객체를 이용해서 Command를 전송하는 스레드에게 다음 Command를 실행할 수 있도록 통지하는 역할을 한다. 또한 호스트에서 블루투스 장치로 전달하는 데이터 패킷은 블루투스 장치의 데이터 버퍼가 넘치지 않도록 보내야 한다. 이러한 일련의 과정을 흐름제어라 하며, 이 논문의 HCI 흐름제어 부분은 다음과 같이 구현하였다.

[그림 3]과 같이 데이터 패킷을 블루투스 장치로 전달하는 스레드와 Number\_Of\_Completed\_Packet Event를 전달받는 스레드를 각각 독립적으로 동작하도록 구현하였으며, 이들 간의 정보전달을 위하여 파이프 객체를 사용하였다.

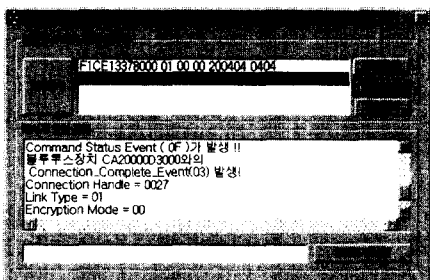
4. 실험 및 고찰

HCI의 구현은 Windows2000을 기반으로 하였으며, 실험을 위해 사용된 장치는 MMC Technology의 Bluetooth USB Adaptor를 사용하였고, USB드라이버와 통신하기 위해 CSR (Cambridge Silicon Radio)에서 제공되는 BlueCore01 USBStack 드라이버 모듈을 사용했다. 다음에 나오는 그림들은 HCI의 동작을 검증하기 위해 HCI\_Test 프로그램을 구현하여 실험한 일련의 결과들이다.



[그림 4] 블루투스 장치 검색

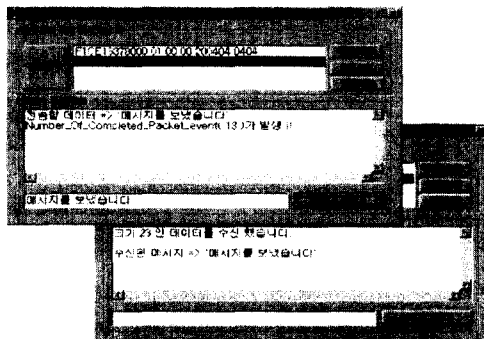
HCI\_Test 프로그램을 실행하여 블루투스 장치를 검색한 화면은 [그림 4]와 같다. 장치검색 버튼을 누르면 상단 리스트 박스에 검색된 장치들의 주소와 세부적인 속성들이 순서대로 나오고, 중간 메시지 박스에는 검색된 블루투스 장치들의 주소와 발생한 Event코드 정보가 성공적으로 표시되는 것을 확인할 수 있다.



[그림 5] 블루투스 장치 연결

\*본 연구는 (주)진두네트웍과의 연구과제로 수행되었음.

[그림 5]는 검색된 블루투스 장치리스트 중 원하는 블루투스 장치와 연결을 설정한 화면이다. 리스트 박스에서 장치를 선택하고 연결설정 버튼을 누르면 선택된 블루투스 장치와의 연결이 이루어지고 연결된 장치의 주소와 발생한 Event코드 정보가 메시지 박스에 성공적으로 표시된다.



[그림 6] 블루투스 장치 연결을 통한 데이터 전송

[그림 6]은 연결이 설정된 장치들 사이에서 데이터를 전송한 화면이다. 연결 설정 후 메시지를 입력하고 메시지 보내기 버튼을 누르면, 연결하고 있는 다른 블루투스 장치로 데이터가 전송된다. 수신 측 호스트는 보내온 데이터 패킷의 순서와 크기에 맞게 패킷을 받게 되고, 전송 측 호스트는 전송이 확인된 데이터 패킷마다 성공적인 Event가 발생하였다. 이것으로 두 장치간의 데이터통신에 흐름제어가 이루어지는 것을 확인할 수 있었다.

5. 결론

블루투스는 차세대 근거리 무선통신 기술로 각광받고 있으며, 이것은 프로토콜 스택의 개발을 통한 다양한 서비스제공을 필요로 하게 된다. 따라서 이에 대해 수행한 첫 단계 연구과제로 상위계층 프로토콜 스택 개발을 위한 HCI를 구현하였고, 표준 규약에 따르는 HCI의 동작 과정을 검증하기 위하여 HCI\_Test 프로그램을 통한 일련의 실험들을 해보았다. 실험결과로 블루투스 장치 검색과 연결설정 및 데이터 전송이 모두 성공적으로 수행되는 것을 확인할 수 있었다.

6. 참고 문헌

- [1] <http://www.zenocom.co.kr/data/>
- [2] <http://www.3com.co.kr/mobile/wireless/>
- [3] The Bluetooth specification v1.1  
<http://www.bluetooth.com>
- [4] CSR BlueCore01 USBStack document  
<http://www.csr.com>
- [5] Jennifer Bray and Charles F Sturman,  
Bluetooth Connect Without Cables,  
Prentice Hall, 2001
- [6] James Y. Wilson and Jason A. Kronz,  
Inside Bluetooth Part I, Dr. Dobb's Journal, 2000