

리눅스 기반 공개 블루투스 프로토콜 스택의 분석 및

구현에 관한 연구*

문천풍⁰ 박찬일 문승진
수원대학교 정보공학대학 컴퓨터학과
(mcpbe⁰, pci96, sjmoon)⁰@mail.suwon.ac.kr

A Study on Analysis and Implementation of Linux-Based Open Bluetooth Protocol Stack

Chun-Poong Moon⁰ Chan-Il Park Seung-Jin Moon
Dept. of Computer Science, The University of Suwon

요 약

블루투스는 저 전력과 휴대성이 높은 특징을 갖는 단거리 무선 통신 기술이며 다양한 응용 범위를 지니고 있어 차세대 통신기술로 각광받고 있다. 이에 따라 다양한 블루투스 서비스들의 요구가 증대되고 있고 이들의 개발에 필요한 프로토콜 스택의 중요성 또한 커지고 있다. 본 논문에서는 오픈소스 프로토콜 스택인 OpenBT를 블루투스 모듈과 호스트 사이에서 중요한 역할을 하는 HCI를 중심으로 분석하였으며 이를 기반으로 한 응용 프로그램의 구현 및 테스트에 관하여 논하였다

1. 서 론

블루투스는 이동전화, 헤드셋, 그리고 휴대용 컴퓨터와 같은 장치를 연결하고 있는 케이블을 제거하기 위해 개발된 단거리 무선 통신 기술이다. 블루투스는 현재 블루투스 기술의 표준을 확정하기 위한 Bluetooth SIG(Special Interest Group)라는 컨소시엄을 중심으로 전세계 2000여 개의 회원사가 참여하고 있으며 그 관심이 점점 커지고 있다. [1]

이러한 블루투스는 그 응용 범위도 매우 넓어 각종 휴대장치 뿐만 아니라 산업 현장에도 적용이 가능하며 차세대 무선통신 기술로 떠오르고 있다. 이에 따라 다양한 응용 애플리케이션의 개발이 필요해지고 있으며 그 기반이 되는 블루투스 프로토콜 스택 개발의 중요성도 커지고 있다. [2]

이에 본 논문에서는 블루투스 프로토콜 스택을 개발하기 위하여 공개되어 있는 블루투스 프로토콜 스택을 분석하였다. 특히 하드웨어와 소프트웨어 사이에서 인터페이스 역할을 담당하는 HCI(Host Controller Interface)를 면밀히 분석하였고 간단한 응용 프로그램을 구현하여 이를 확인하였다. 본 논문의 구성은 2장의 관련연구에서 블루투스 프로토콜 스택과 OpenBT에 대해 설명하고 3장에서 HCI의 정의와 구조, 주요 함수들을 살펴본다. 그리고 4장에서 스택을 이용한 구현을 보이며 마지막으로 결론 및 향후 과제를 기술하였다.

2. 관련 연구

* 본 연구는 (주) M2I 와 공동연구한 중소기업 기술 혁신과제 연구결과파의 일부를 포함함

2.1 블루투스 프로토콜 스택

블루투스 프로토콜 스택은 일련의 계층으로 이루어진 블루투스 통신 규격의 핵심이며 블루투스 기반 장비 및 응용 애플리케이션 제작에 필수적으로 필요한 기술이다. 블루투스 프로토콜 스택의 기본 구조는 다음과 같다.

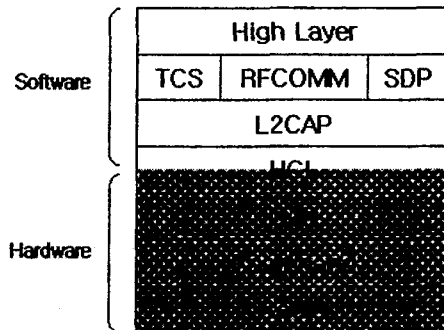


그림 1 블루투스 프로토콜 스택의 구조

블루투스 프로토콜 스택의 구조는 HCI를 기준으로 L2CAP과 그 이상을 구현하는 호스트, LMP와 그 이하를 구현하는 모듈로 나누어진다. 블루투스 모듈은 하드웨어를 직접 제어하는 펌웨어 레벨을 포함하여 링크의 설정과 에러 검출, 신뢰성 있는 데이터 전송 등을 담당한다. 블루투스 호스트에 속하는 L2CAP(Logical Link Control and Adaptation Protol)은 서로 다른 엔티티 간의 다중화, 패킷의 분할과 재조립 등의 역할을 수행하여 상위 계층 프로토콜들이 통신을 수행하는 데에 필요한

편의를 제공한다. HCI는 모듈과 호스트 사이에서 균일한 인터페이스를 제공하는 역할을 한다. [3][4]

2.2 OpenBT

OpenBT는 Axis사에서 개발한 공개 프로토콜 스택으로 스택 개발 시 참조가 될 수 있도록 HCI를 비롯한 상위 계층의 프로토콜에 대한 소스를 제공한다. 이 스택은 기본적으로 리눅스에서 운용되도록 디자인 되어 있지만 이를 이용하여 Windows 등의 다른 OS 환경으로 포팅 하는 것도 가능하다. 많은 스택 개발 업체들도 실제로 스택 개발 시 참조하는 기본 요소이기 때문에 스택을 개발하는데 적절한 연구 모델이다.

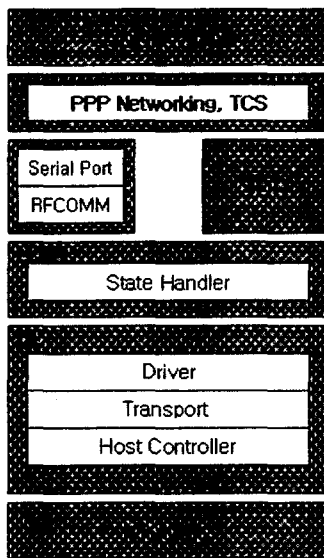


그림 2 OpenBT 스택의 구성

OpenBT 스택은 SIG에서 제정한 표준 블루투스 규격에 의거하여 C언어로 구현되어 있다. OpenBT 스택은 블루투스 모듈에 해당하는 Baseband, LMP, 그리고 Radio 계층의 상위에 위치한다. 이 스택은 블루투스 코어 부분인 HCI, L2CAP, SDP 계층과 상위계층인 RFCOMM, PPP, TCP/IP를 구현하고 있다. 각 계층은 최 상위 응용계층과 상호 유기적인 관계를 유지하고 있으며 각종 정보를 주고 받고 있다. [5]

이들 중 HCI는 블루투스 모듈과 호스트를 모두 포함하는 계층으로 이들 사이에서 서로의 데이터를 연결해주는 역할을 수행하여 상호 호환성을 보장해 준다. 블루투스에서는 이러한 HCI를 표준으로 정하여 하드웨어와 소프트웨어를 분리시킴으로 다양한 모듈에 대한 동일한 인터페이스를 제공해준다.[6] 따라서 HCI의 구현은 상위 프로토콜을 개발하기 위해 필수적이다. 이에 본 논문에서는 HCI를 중심으로 OpenBT를 분석하였다.

3. HCI

3.1 HCI 정의와 구조

HCI는 블루투스 하드웨어 모듈과 서로 주고받는 명령 구조와 패킷 및 절차를 정의한다. 즉, 블루투스 모듈이 이해할 수 있는 표준 포맷으로 데이터를 만들어 보내고 블루투스 모듈은 그 결과를 표준 패킷으로 만들어 호스트로 보내는 방법을 정의하고 있다. [7]

| 패킷 | 정의 |
|-------------|---|
| Commands 패킷 | 호스트에서 블루투스 모듈에 전송 블루투스 모듈의 상태를 점검하고 제어하기 위해 사용 |
| Event 패킷 | 모듈에서 호스트로 전송되는 이벤트 Command Packet의 요청에 대한 상태나 결과 설정값 등에 대한 정보를 포함하는 패킷 |
| Data 패킷 | ACL 패킷 : 데이터 전송에 기본적으로 사용되는 패킷 SCO 패킷 : Piconet 연결이나 음성 데이터에 사용되는 패킷 |

표 1 HCI 주요 패킷

HCI 계층은 HCI Driver, HCI Transport, Host Controller 3개의 모듈로 이루어져 있다.

HCI Driver는 HCI 계층의 상위 모듈로 명령어, 이벤트, 데이터 교환등의 작업을 블루투스 하드웨어에 요구하는 상위 계층을 위한 논리적인 인터페이스를 제공한다. HCI Transport 모듈은 논리적인 인터페이스와 물리적인 인터페이스로 나뉘어져 제공된다. 이 계층은 호스트에서 HCI 패킷을 받아 블루투스 모듈로 보내는데 쓰이며 USB, RS-232, UART 3개의 계층을 정의한다. Controller는 스택의 하드웨어 계층에 접근하기 위해 하드웨어 모듈위에 펌웨어로 존재하여 여러 하드웨어의 제어와 접근을 제어한다.[8]

3.2 HCI 주요 함수

OpenBT에 구현된 HCI의 주요 함수는 그림 3과 같다.

Local Function 부분은 이벤트나 ACL, SCO 패킷에 대한 처리 과정으로 계층간에 패킷이 원활하게 전송되도록 버퍼를 생성하고, 명령어를 초기화 하는 작업을 하며 핸들러와 관련된 일들을 수행한다.

Link Control Commands 부분은 명시한 블루투스 기기의 접속을 형성하는 LM을 수행시키는 create_connection, 접속 종료시 사용되는 disconnect, 새로 들어오는 접속 요구를 수용하는데 사용되는 accept_connection_request, 새로운 연결 요구에 대한 거절을 수행하는 reject_connection_request, 패킷 타입을 변경하는데 사용하는 change_connection_packet_type, 다른 블루투스 기기의 사용자 이름을 얻는데 필요한 remote_name_request 함수들로 구성되어 있다.

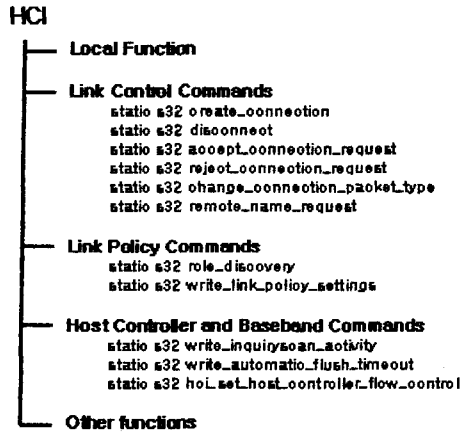


그림 3 HCI의 주요 함수

]

Link Policy Commands 부분은 특정 Connection Handle을 가지고 수행되고 있는 기기의 역할을 결정하는 role_discovery, 명시된 Connection 핸들을 위한 Link policy, 설정하는 write_link_policy_settings으로 구성된다.

Host Controller and Baseband Commands 부분은 inquiry_scan_activity configuration 파라미터에 대한 값을 쓸 수 있는 write_inquiryscan_activity, 특정한 접속 핸들에 대한 값을 저장하는 write_automatic_flush_timeout, 호스트 컨트롤러에서 호스트의 흐름제어를 제어할 때에 사용되는 hci_set_host_controller_flow_control 함수들로 이루어져 있다.

Others 부분은 호스트에서 호스트 컨트롤러로 보낸 ACL과 SCO데이터 패킷의 최대 크기를 읽거나 HCI 연결에 관한 설정과 연결된 이름을 얻는데 사용되는 함수들로 구성되어 있다.

4. 구현

OpenBT를 이용하여 gcc로 간단한 파일 전송 프로그램인 BlueFT를 구현하였다. Linux Kernel 2.2.7 버전을 기반으로 하였고 하드웨어 장치로 CSR의 BlueCore TM01 모듈을 사용하였다. 모듈과 호스트의 통신은 시리얼을 통해 이루어졌다.

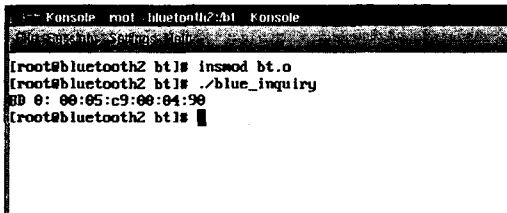


그림 4 스택의 적재와 장치 검색

그림 4는 OpenBT 스택을 커널에 적재하는 모습과 다른 장치를 검색하여 찾아낸 주소를 보여주고 있다.

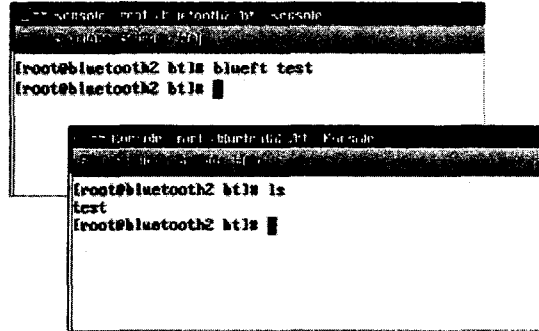


그림 5 파일 전송

그림 5는 BlueFT 프로그램을 이용한 파일 전송과 성공적으로 전송되었음을 확인하고 있다. BlueFT 프로그램은 검색된 장치 주소를 바탕으로 그 장치에 연결을 설정하고 인자로 받아들이는 파일을 전송한다.

5. 결론 및 향후 연구 과제

블루투스 그 적용분야가 점차 늘어나고 있고 그에 맞는 다양한 응용 제품의 개발에 반드시 필요한 프로토콜 스택 개발이 필요하게 되었다.

본 논문에서는 리눅스를 기반으로 한 공개된 블루투스 프로토콜 스택인 OpenBT를 상위 계층을 구현하기 위해 필수적인 HCI를 중심으로 분석하였다. 그리고 이를 시스템에 적재하여 간단한 파일 전송 프로그램을 구현하였다. 향후 이를 스트롱 암을 이용한 임베디드 시스템에 적용시키고자 하며 더 나아가 원천기술 확보를 위해 블루투스 프로토콜 스택을 구현하고자 한다.

6. 참고 문헌

- [1] <http://www.seecode.com/>
- [2] Bluetooth Demystified
Author, Nathan J. Muller
- [3] Specification v1.1 core of the Bluetooth System
<http://www.bluetooth.com/>
- [4] Bluetooth Connect Without Cables
Author, Jennifer Bray, Charles F Sturman
- [5] Axis OpenBT
<http://www.openbt.org>
- [6] Bluetooth Revealed, PH PTR
Author, Brent A. Miller, Chatschik Bisdikian
- [7] <http://edu.lut.fi/lutpub/>
- [8] Bluetooth Tutorial
<http://www.palowireless.com/bluetooth/>