

안드로이드에서 블루투스 HDP를 이용한 표준건강정보의 송수신

박영진[°], 조희섭^{*}, 손종욱^{**}

Transmitting/Receiving of Standard Health Data Using Bluetooth HDP on the Android Platform

Young-jin Park[°], Hui-sup Cho^{*}, Jong-wuk Son^{**}

요 약

현재 헬스케어에 대한 관심이 높아지면서 건강정보측정기기에 필수적으로 적용되어야 할 표준에 대한 연구가 활발히 진행되고 있다. 이러한 표준화 및 시험/인증은 IEEE 11073 PHD, Health Level Seven 그리고 Continua Health Alliance를 통해 진행되고 있으며, 앞으로 더 많은 연구와 실험이 계속될 것이라고 생각된다. 본 논문에서는 원격 건강 정보의 교환을 위한 표준인 IEEE 11073 PHD와 Bluetooth의 건강정보측정기기를 위한 프로파일인 Bluetooth Health Device Profile(HDP)을 소개하고, HDP가 적용된 스마트 디바이스와 건강정보측정기기 간 개인의 건강 정보를 교환하는 실험을 진행하였다. 실험을 통해 HDP를 이용하여 IEEE 11073 PHD 표준 프로토콜의 송수신으로 스마트 디바이스와 건강정보측정기기 간에 상호연결 및 운용이 가능함을 확인할 수 있었고, 향후 이를 지원하는 기기를 통해 다양한 원격 의료 서비스가 구현될 수 있을 것이라 기대한다.

Key Words : Bluetooth, HDP, IEEE 11073, Android, Health Care

ABSTRACT

These days, with increasing interest about the health care, research about standard for transmitting/receiving data of health device has been actively conducted. Standardization and testing / certification is underway through the IEEE 11073 PHD, Health Level Seven and the Continua Health Alliance, it'll continue to be more research and experiments in the future. In this paper, we introduce the IEEE 11073 PHD standards for exchange of remote health information and the Bluetooth Health Device Profile(HDP) for health device profile of the Bluetooth technology, also we conducted experiments about exchange health information between health device and smart device which are adapted the IEEE 11073 PHD and Bluetooth HDP. Transmitting/receiving of the IEEE 11073 PHD standard protocol using the Bluetooth HDP profile between health device and smart device can be known by this experiment, and it expect to implement variety of remote medical services through health device supporting these standards in future.

I. 서 론

의료정보교환은 의료 서비스를 구현하는 것으로
의료정보의 교환과 관련된 표준화의 구현이 중요하

※ 본 연구는 미래창조과학부에서 지원하는 대구경북과학기술원 기관고유사업(13-IT-04)에 의해 수행됨.

◆ 주저자겸 교신저자 : 대구경북과학기술원 IT융합연구부, yjpark@dgist.ac.kr, 정희원

* 대구경북과학기술원 IT융합연구부, mozart73@dgist.ac.kr, 정희원

** 대구경북과학기술원 IT융합연구부, jwson@dgist.ac.kr, 정희원

논문번호 : KICS2012-03-129, 접수일자 : 2013년 3월 8일, 최종논문접수일자 : 2013년 5월 13일

다. HL7과 IEEE 11073 PHD 등 의료정보 표준을 통해 의료정보를 가공 및 처리하여 사용자에게 서비스하려는 노력이 계속되고 있다^[1]. 이렇듯 IT 기술을 이용하여 의료정보를 표준화하고 교환하는 방식을 구현한 것을 의료 정보화라 한다. 의료정보화가 실현되면 환자와 의료기관 그리고 건강정보추정기기 간 표준화된 정보를 교환하고, 교환된 정보의 활용은 의료체계의 효율성을 재고하는 가장 강력한 수단으로 인식되고 있다^[2].

현재 건강정보의 표준화 및 단체는 HL7, ISO/TC215 그리고 IEEE 11073 PHD 등을 통해 진행되고 있으며, 그 중에서도 IEEE 11073 PHD는 건강정보추정기기 (Agent)와 관리기기 (Manager) 간의 상호운용성 보장을 통한 개인건강기기의 보급 활성화를 위해 최적화된 응용 프로토콜 (OXP, Optimized Exchange Protocol)인 IEEE 11073-20601을 중심으로 각 개인건강기기별 표준 제정 작업을 지속적으로 진행하고 있다^[3]. 또한 헬스케어 서비스 산업을 위한 산업체 협력 조직으로 Continua Health Alliance (CHA)가 있으며, CHA에서는 현재 IEEE 11073 PHD 표준을 통해 건강정보 추정기기과 PC 및 스마트 디바이스 사이의 상호운용성을 보장하도록 시험 및 인증 제도를 운영 중이다. 그리고 기기 간 통신 방식은 새롭게 정의되는 것이 아니라 기존의 Bluetooth, USB, Zigbee 등을 채택하고 있다^[8]. 우리나라의 경우에도 의료정보의 표준화에 대한 인식이 점차 증가되고 있으나 아직 학계 및 산업계에서의 연구 및 제품화는 미약한 수준이며, IEEE 11073 PHD 표준화 및 CHA 인증 시험은 한국정보통신기술협회에서 지원하고 있다.

또한 Bluetooth HDP(Health Device Profile)는 건강기와 스마트 디바이스 등의 데이터 수집 기기 간의 블루투스 연결 및 통신을 정의하고 구현한 프로파일로 안드로이드 플랫폼일 경우 아이스크림 샌드위치 (ICS, IceCream Sandwich) 버전부터 탑재가 되어 있어 아직 연구가 많이 부족한 상황이다.

본 논문에서는 HDP와 IEEE 11073 PHD를 소개하고, 두 표준 기술을 이용하여 원격으로 개인의 건강 정보를 교환하는 방법을 실험하였다. 실험은 스마트 디바이스에서 HDP를 통해 혈압계와 체중계에서 송신하는 건강 정보인 IEEE 11073 PHD 프로토콜을 수신하고, 수신된 프로토콜을 분석 및 처리하여 사용자에게 디스플레이해 주는 어플리케이션을 구현하여 실험하였다. 참고로 본 논문에서 사용되는 용어로 스마트 디바이스와 같은 데이터 수집 장치

는 HDP의 경우에는 싱크 (Sink), IEEE 11073 PHD에서는 매니저 (Manager)로 정의되며, 건강정보추정기기의 경우에는 HDP에서는 소스 (Source) 그리고 IEEE 11073 PHD에서는 에이전트 (Agent)로 용어 정의가 되어 있다. 본 논문에서는 IEEE 11073 표준 설명 절을 제외한 공통적인 장과 절에서는 HDP에 정의된 용어를 사용하였다.

논문의 구성은 2장에서는 표준화 동향과 HDP 그리고 IEEE 11073 PHD 표준화 관련 연구에 관해 기술하고, 3장에서 실험 내용 및 결과를 보이고, 마지막으로 4장에서 결론을 맺는다.

II. 관련 연구

2.1. 의료정보 표준 동향

IEEE 11073 PHD는 IEEE 11073-20601을 바탕으로 건강정보추정기기별로 표준을 제정하고 있다. IEEE 11073-20601은 Optimized Exchange Protocol로서 건강정보추정기기 와 관리기기 간 데이터 형식과 최적화된 교환 프로토콜을 정의하여 데이터 송수신 관련 상호운용성을 보장하고 있다. IEEE 11073 PHD 표준은 의료 기기와 외부에 있는 컴퓨터 시스템 간의 통신을 가능하게 하며, 통신에 대한 규범적인 정의를 내린다. 표준의 구성은 크게 전송 계층, 최적화된 교환 프로토콜 그리고 기기별 표준으로 구분되며, 물리적 전송 방법에 대해서는 정의하지 않고 현재까지 구현 가능한 블루투스, HDP, USB, Zigbee 등을 모두 사용할 수 있도록 가정하고 있다. 현재 IEEE 11073 PHD 표준화는 개념이 추상적인 부분이 있어 에이전트의 특성에 따라 IEEE 11073-104zz에 기기별 표준을 채택하고 있다.

Health Level Seven (HL7)은 보건의료정보시스템 간 데이터를 교환하기 위해 개발된 표준 프로토콜로 현재 북아메리카 지역에서는 의료 정보의 전자적 교환을 위한 표준으로 채택되어 있다. HL7을 이용하여 개인의 건강 정보를 교환하기 위해서는 메시지를 이용한다. 메시지는 데이터 전송의 가장 작은 단위이며, 서로 다른 의료정보시스템 간의 정보 호환성 보장을 위한 규칙들의 집합이며, 그 내용은 여러 자료형으로 정의되며, 스트링 형식으로 전송된다^[2].

HL7이 IEEE 11073 PHD는 의료정보의 표준화라는 점에서 비슷하며, 2절에서 설명할 DIM 등의 모델링 방식 역시 비슷한 구조로 표준화되어 있다. 두 표준은 거의 비슷한 양식의 표준화 작업을 진행

하고 있는데 현재까지 사용되는 형태를 보면, HL7 이 병원에서 사용되는 의료기기들 간의 통신 표준에 가깝다면 IEEE 11073 PHD는 개인이 사용하는 의료기기들 간의 통신 표준에 가깝고 할 수 있을 것이다.

2.2. IEEE 11073 PHD 표준

2.2.1. IEEE 11073-20601 모델

IEEE 11073-20601 OXP의 각 프로토콜 데이터는 데이터와 상호운용성에 관한 추상적 모델링에 사용되는 표준 언어인 ASN.1 방식을 따르며, 기기별 호환성을 위해 IEEE 11073-10101에서 용어와 관련한 이진 코드 매크로 (Macro)로 정의된 명명법인 Nomenclature를 사용하며, 도메인 정보 모델 (DIM, Domain Information Model), 서비스 모델 (Service Model) 그리고 통신 모델 (Communication Model)로 구성된다. 먼저, 도메인 정보 모델은 기기와 생리학적인 데이터에 관해 기술하는 객체지향 모델로 건강정보측정기기를 하나의 객체 집합으로 표현하여 객체를 선택 및 구성할 수 있으며, 객체 속성 및 이용 가능한 메서드 및 이벤트 그리고 서비스 등을 정의한다. 두 번째로, 서비스 모델은 기기 간 상호 동작과 기기가 데이터에 접근하는 것에 관한 모델로 에이전트와 매니저 간의 데이터 접근 방법을 나타낸다. 서비스 모델의 명령으로는 GET, SET, ACTION 그리고 Event Report 등이 있으며 실제 측정된 값의 전달은 Event Report을 통해 이루어진다. Event Report는 가변형 (Variable format), 고정형 (fixed format) 그리고 그룹형 (Grouped format)으로 해당 기기에 맞게 구성할 수 있다. 마지막으로, 통신 모델은 통신 상태에 관련된 8개의 State machine (Connected, Disconnected, Disassociating, Unassociated, Associating, Associated, Configuring, Operating)을 정의하고 있다. 매니저가 에이전트의 구성정보를 알고 있는 경우 에이전트와 매니저가 물리적/논리적 연결 후 Operating 단계로 진행되며, 에이전트의 구성 정보를 모르고 있는 경우 Configuring 단계를 통해 구성 확인 과정 및 등록 후 Operating 단계로 넘어 간다⁴⁾.

2.2.2. Medical Device System(MDS)

MDS는 DIM의 최상위 클래스로 에이전트의 데이터 형태에 따라 몇 가지 하위 클래스로 나누어 구성된다. 수치 데이터만 다룰 경우 수치 클래스

(Numeric Class)를 사용하고, 측정값에 영향을 줄 수 있는 여러 요인들에 대한 추가 정보를 전달하거나 기기 상태 등을 전송하고자 하는 경우에는 열거형 클래스 (Enumeration Class)를 구성에 추가한다. 또한 실시간 데이터 처리를 위해 실시간 샘플 배열 (RT-SA, Real Time-Sample Array)가 사용되며, 연결이 끊어진 경우에도 지속적으로 측정 및 일정량의 데이터를 저장할 수 있도록 하는 PM-store 클래스가 있다.

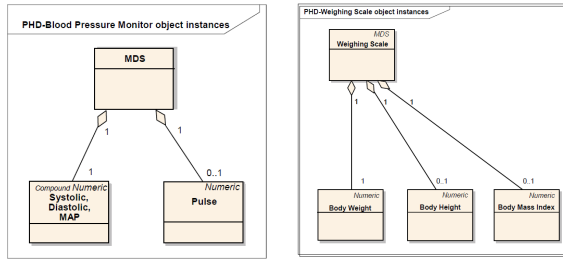
표 1. 체중계의 MDS 객체 이벤트
Table 1. MDS object events of weighing scale

Service	Subservice type name	subservice type (event-type)	Parameters (event-info)
Event Report	MDS-Configuration-Event	MDC_NOTI_CONFIG	ConfigReport
	MDS-Dynamic-Data-Var	MDC_NOTI_SCAN_REPORT_VAR	ScanReportInfoVar
	MDS-Dynamic-Data-Update-Fixed	MDC_NOTI_SCAN_REPORT_FIXED	ScanReportInfoFixed
	MDS-Dynamic-Data-MP-Var	MDC_NOTI_SCAN_REPORT_MP_VAR	ScanReportInfoMPVar
	MDS-Dynamic-Data-Update-MP-Fixed	MDC_NOTI_SCAN_REPORT_MP_FIXED	ScanReportInfoMPFixed

그리고 스캐너 클래스 (Scanner Class)는 통신 효율을 위해 불필요한 값들을 제외하고, 측정값만을 전송하거나 그룹 단위로 전송할 수 있도록 한다. 스캐너 클래스는 표 1에 정의된 event report 서비스를 통해 에이전트가 매니저에 요청 및 응답하거나 측정된 값을 송신하기 위해 사용되는 서비스로 가변형, 고정형 그리고 그룹형으로 구성하여 제공한다. 본 논문의 실험에서 사용된 혈압계는 고정형 event report를 사용하고, 체중계는 가변형 event report를 사용하는 건강정보측정기기이다.

2.2.3. IEEE 11073-104zz

1절에서 언급한 대로 IEEE 11073 PHD는 의료기기 별로 IEEE 11073-104zz에 표준을 각각 정의하고 있다. 본 논문에서는 실험을 위해 사용한 혈압계 (IEEE 11073-10407)와 체중계 (IEEE 11073-10415)에 대해서만 다룬다.



(a) 혈압계 (a) Blood pressure meter (b) 체중계 (b) Weighing scale
 그림 1. DIM 객체 인스턴스 다이어그램
 Fig. 1. DIM object instance diagram

그림 1-(a)^[6]는 혈압계 DIM의 객체의 인스턴스를 나타낸 그림으로, 두 개의 수치 클래스 (Numeric class)가 존재하는 것을 볼 수 있다. 첫 번째 클래스는 수축 (Systolic) 혈압, 이완 (Diastolic) 혈압 그리고 평균 동맥압 (MAP, Mean Arterial Prassue) 측정 값을 Medical Device System (MDS)로 보내는 객체들을 포함하는 표준 구성으로 정의되는 클래스이고, 두 번째 클래스는 심박수 (Heart Rate) Pulse 측정 값을 MDS로 보내는 객체로 추가 구성을 위한 클래스이다. 그림 1-(b)^[7]는 체중계 DIM의 객체 인스턴스를 나타낸 그림으로, 세 개의 수치 클래스가 존재한다. 체중을 나타내는 객체를 포함하는 표준 구성을 위한 클래스 하나 외에 추가 구성을 위한 두 개의 클래스로 구성되어 있다. 참고로 클래스를 표시하는 박스의 캡션을 보면 표준 구성은 1로 표시되며 추가 구성은 0.1로 표시하고 있는 것을 확인할 수 있다.

2.3. Bluetooth Health Device Profile

HDP는 Bluetooth Special Interest Group(SIG)의 Medical Device Working Group (MED WG)가 건강정보추정기기 (source)와 데이터 수집 디바이스 (Sink) 사이의 상호운용성을 위해 개발한 프로파일로 2008년 채택 (adapted)되었으며, Bluetooth Serial Port Profile (SPP) 기반으로 구현되었다. 또한 어플리케이션 단에서는 IEEE 11073-20601 PHD 프로토콜과 IEEE 11073-104zz를 제공하여 상호운용성을 확보하였다.

HDP는 제어 채널 (Control channel)과 데이터 채널 (Data channel) 등 두 가지 형태의 채널을 제공한다. 제어 채널은 데이터 채널 파라미터들을 negotiate하기 위해 사용되며, 데이터 채널(들)을 설정하는 기능을 한다. 데이터 채널을 실제 IEEE 11073-104zz 데이터 전송을 위해 사용되는데, 데이

터 채널은 비연속적인 데이터를 위한 Reliable 데이터 채널과 연속적인 데이터를 위한 Streaming 데이터 채널로 다시 나눌 수 있다. 먼저 제어 채널로 연결이 확립된 후 하나 또는 그 이상의 데이터 채널의 연결을 확립한다. 두 채널의 연결이 확립이 되면 데이터의 특성에 맞게 Reliable 데이터 채널 또는 Streaming 데이터 채널을 통해 데이터를 송수신하게 된다.

현재 CHA에서 HDP와 IEEE 11073 PHD와 관련한 시험/인증을 하고, 실제 HDP를 탑재한 의료기기들이 해외 업체에서 생산 및 판매되고 있다. 하지만 국내 건강정보추정기기 또는 의료기기 제조사에서는 아직 상용화된 제품이 없는 실정이며, 국내에서 이러한 표준을 적용한 예는 삼성전자의 갤럭시 S3에 HDP를 이용하여 건강 정보를 측정할 수 있는 ‘S헬스’ 앱이 유일한 것으로 파악된다. 안드로이드에서 HDP의 지원은 2012년 발표 및 적용된 ICS 버전부터 가능하기 때문에 향후 국내외 시장 확보를 위해 스마트 디바이스 제조사의 기본 어플리케이션 및 특정 의료기기의 연결이 아닌 학계 또는 산업계의 활발한 연구 및 제품화가 필요할 것으로 생각된다.

III. 실험

3.1. 실험 환경

본 논문의 실험을 위한 소스로 사용한 기기는 HDP가 탑재된 A&D 사의 혈압계와 체중계를 사용하였고, 싱크로 사용한 기기는 안드로이드 젤리빈 버전이 탑재된 LG전자의 옵티머스 LTE II를 사용하였다. 그리고 안드로이드 앱 개발을 위해 이클립스 Juno 버전을 사용하였으며, 안드로이드 ICS 버전이 탑재된 삼성전자의 갤럭시탭 10.1에서도 실험을 동시에 진행하였다.

3.2. 실험 내용

본 실험의 목적은 현재 서서히 많은 관심을 가지고 연구되고 있는 의료정보의 표준화와 관련하여 Bluetooth HDP와 IEEE 11073 PHD를 통한 건강 정보를 교환하는 방법에 관한 것이다. 따라서 실험을 위해 스마트 디바이스에서 건강정보추정기기와 Bluetooth HDP를 이용한 연결 및 데이터 송수신 결과를 UI에 디스플레이할 필요가 있으므로 안드로이드에서 어플리케이션(앱)을 구현하여 진행하였다.

실제 사용자가 안드로이드 플랫폼이 탑재된 스마

트 디바이스를 이용하여 혈압계를 연결하고, 안드로이드 앱인 DGIST_HDP와 데이터 송수신 및 결과를 확인하는 과정의 예를 아래 표 2의 Use-case에서 단계(1~15) 별로 나타내었다.

표 2. 유즈케이스 시나리오
Table 2. Use-Case scenario

1. Turn the power of blood pressure monitor on and then make sure that the power is applied.
2. Run DGIST_HDP App on smart device.
3. Make sure that the main screen appears.
4. Select the blood pressure monitor in the health device selection screen on smart device.
5. Push the Registration button
 - A. Make sure that pop-up about “completion of registration” occurs.
 - B. Push the Registration button again, if pop-up occurs about “error of registration”
6. Push the Connection button.
7. Make sure that pop-up about connectable device list occurs.
 - A. Select the blood pressure monitor that want to connect with smart device.
 - B. Push the Connection button
 - C. Make sure that pop-up about completion of connection occurs.
 - D. Retry above process, if pop-up is appeared by the App about connection error
8. Wind the cuff on left(or right) upper arm.
9. Push the START button on blood pressure monitor.
10. Make sure that measurement is running and the air into the cuff is injected.
11. Make sure that you can see a systolic(SYS./) diastolic(DIA.) blood pressure and pulse(PUL.) on LCD of blood pressure monitor.
 - A. If an error occurs, turn the power off and then turn it on again.
12. Make sure that you can see a systolic(SYS./) diastolic(DIA.) blood pressure and pulse(PUL.) on the App.
 - A. Make sure that result value on the App is same with the displayed values on LCD of the blood pressure monitor.
 - B. if measured value does not display, retry every process from the beginning.
13. Unwind the cuff on arm.
14. Turn the power of blood pressure monitor off.
16. Push the Exit button on smart device and then make sure that the App does not running on screen.

안드로이드에서 Bluetooth HDP 앱을 구현하기 위해서는 BluetoothHealth 클래스, BluetoothHealthCallback 클래스 그리고 BluetoothHealthAppConfiguration 클래스의 API들을 적절히 이용하여 Bluetooth 연결 및

데이터 송수신 관련 기능을 구현할 수 있다. 위 클래스의 자세한 내용은 구글 개발자 사이트^[9]를 참조하면 클래스 및 API 설명 그리고 샘플 코드를 볼 수 있다. 위에 언급된 클래스들을 이용하여 구현 후 직접 실험을 해 보면, 혈압계의 측정 데이터 로그(Log)를 나타내는 그림 2 및 3과 측정계의 데이터 수신 결과 로그를 나타내는 그림 4를 통해 두 기기 모두 정상적으로 연결 및 데이터 수신이 되는 것을 볼 수 있다. 데이터는 IEEE 11073 PHD와 IEEE 11073-10407 및 IEEE 11073-10415 표준을 준수하는 바이트 형태의 프로토콜로 수신되며, 안드로이드 앱에서 IEEE 11073 프로토콜 데이터를 파싱하여 UI에 디스플레이 하였다. 그림 2는 혈압 측정 후 수축 혈압과 이완 혈압을 안드로이드 앱에서 수신한 결과이다. 바이트 배열의 38~39번 인덱스의 값은 수축 혈압 값이고, 40~41번 인덱스의 값은 이완 혈압 값이다. 정수형으로 보면 수축 혈압은 0x00 0x82로 정수로는 130이 되고, 이완 혈압은 0x00 0x5a로 정수로는 90이 된다. 혈압계는 수축 및 이완 혈압과 함께 심장박동수를 다른 패킷으로 구성하여 함께 보내 준다.

```
data[ 0 ] : 0xe7 data[ 1 ] : 0x00 data[ 2 ] : 0x00 data[ 3 ] : 0x30 data[ 4 ] : 0x00
data[ 5 ] : 0x2e data[ 6 ] : 0x10 data[ 7 ] : 0x02 data[ 8 ] : 0x01 data[ 9 ] : 0x01
data[ 10 ] : 0x00 data[ 11 ] : 0x28 data[ 12 ] : 0x00 data[ 13 ] : 0x00 data[ 14 ] : 0xff
data[ 15 ] : 0xff data[ 16 ] : 0xff data[ 17 ] : 0xff data[ 18 ] : 0x0d data[ 19 ] : 0x1d
data[ 20 ] : 0x00 data[ 21 ] : 0x1e data[ 22 ] : 0xf0 data[ 23 ] : 0x00 data[ 24 ] : 0x00
data[ 25 ] : 0x02 data[ 26 ] : 0x00 data[ 27 ] : 0x01 data[ 28 ] : 0x00 data[ 29 ] : 0x16
data[ 30 ] : 0x00 data[ 31 ] : 0x01 data[ 32 ] : 0x00 data[ 33 ] : 0x12 data[ 34 ] : 0x00
data[ 35 ] : 0x03 data[ 36 ] : 0x00 data[ 37 ] : 0x06 data[ 38 ] : 0x00 data[ 39 ] : 0x82
data[ 40 ] : 0x00 data[ 41 ] : 0x5a data[ 42 ] : 0x00 data[ 43 ] : 0x64 data[ 44 ] : 0x20
data[ 45 ] : 0x13 data[ 46 ] : 0x03 data[ 47 ] : 0x07
```

그림 2. 수신된 이완 혈압과 수축 혈압 데이터
Fig. 2. A data set included systolic and diastolic value which is received from the blood pressure meter

그림 3은 심장박동수를 수신한 결과로 혈압 측정 값과 마찬가지로 방식으로 프로토콜을 분석해 보면, 바이트 배열의 34~35번 인덱스의 값이 심장박동수 값으로 0x00 0x45 즉, 정수로 표현하면 69가 된다. 결과적으로 수신된 수축 및 이완 혈압값과 심장박동수는 각각 130, 90 그리고 69가 되고 실제 안드로이드 앱과 혈압계에서 정상적으로 표현되는 것을 3절의 실험결과에서 확인할 수 있다.

```
data[ 0 ] : 0xe7 data[ 1 ] : 0x00 data[ 2 ] : 0x00 data[ 3 ] : 0x28 data[ 4 ] : 0x00
data[ 5 ] : 0x26 data[ 6 ] : 0x10 data[ 7 ] : 0x03 data[ 8 ] : 0x01 data[ 9 ] : 0x01
data[ 10 ] : 0x00 data[ 11 ] : 0x20 data[ 12 ] : 0x00 data[ 13 ] : 0x00 data[ 14 ] : 0xff
data[ 15 ] : 0xff data[ 16 ] : 0xff data[ 17 ] : 0xff data[ 18 ] : 0x0d data[ 19 ] : 0x1d
data[ 20 ] : 0x00 data[ 21 ] : 0x16 data[ 22 ] : 0xf0 data[ 23 ] : 0x00 data[ 24 ] : 0x00
data[ 25 ] : 0x03 data[ 26 ] : 0x00 data[ 27 ] : 0x01 data[ 28 ] : 0x00 data[ 29 ] : 0x0e
data[ 30 ] : 0x00 data[ 31 ] : 0x02 data[ 32 ] : 0x00 data[ 33 ] : 0x0a data[ 34 ] : 0x00
data[ 35 ] : 0x45 data[ 36 ] : 0x20 data[ 37 ] : 0x13 data[ 38 ] : 0x03 data[ 39 ] : 0x07
```

그림 3. 수신된 심장박동수 데이터
Fig. 3. A data set included pulse value which is received

from the blood pressure meter

그림 4는 체중 측정 후 안드로이드 앱에서 수신한 결과를 나타낸 것으로 혈압 측정 방법과 마찬가지로 분석해 보면, 바이트 배열의 42~43번 인덱스의 값이 0x20 0x8a로 정수로 표현하면 8330, 킬로그램(kg)으로 표현하면 83.3(kg)이 된다.

```
data[ 0 ] : 0xe7 data[ 1 ] : 0x00 data[ 2 ] : 0x00 data[ 3 ] : 0x3a data[ 4 ] : 0x00
data[ 5 ] : 0x38 data[ 6 ] : 0x10 data[ 7 ] : 0x00 data[ 8 ] : 0x01 data[ 9 ] : 0x01
data[ 10 ] : 0x00 data[ 11 ] : 0x32 data[ 12 ] : 0x00 data[ 13 ] : 0x00 data[ 14 ] : 0xff
data[ 15 ] : 0xff data[ 16 ] : 0xff data[ 17 ] : 0xff data[ 18 ] : 0x0d data[ 19 ] : 0x1e
data[ 20 ] : 0x00 data[ 21 ] : 0x28 data[ 22 ] : 0xf0 data[ 23 ] : 0x00 data[ 24 ] : 0x00
data[ 25 ] : 0x00 data[ 26 ] : 0x00 data[ 27 ] : 0x01 data[ 28 ] : 0x00 data[ 29 ] : 0x20
data[ 30 ] : 0x00 data[ 31 ] : 0x01 data[ 32 ] : 0x00 data[ 33 ] : 0x03 data[ 34 ] : 0x00
data[ 35 ] : 0x1a data[ 36 ] : 0x0a data[ 37 ] : 0x56 data[ 38 ] : 0x00 data[ 39 ] : 0x04
data[ 40 ] : 0xfe data[ 41 ] : 0x00 data[ 42 ] : 0x20 data[ 43 ] : 0x8a data[ 44 ] : 0x09
data[ 45 ] : 0x90 data[ 46 ] : 0x00 data[ 47 ] : 0x08 data[ 48 ] : 0x20 data[ 49 ] : 0x13
data[ 50 ] : 0x03 data[ 51 ] : 0x07 data[ 52 ] : 0x02 data[ 53 ] : 0x50 data[ 54 ] : 0x40
data[ 55 ] : 0x00 data[ 56 ] : 0x09 data[ 57 ] : 0x86
```

그림 4. 체중 측정 후 수신 데이터
Fig. 4. The data which is received from source after measure with weighing scale device

3.3. 실험 결과

실험을 통해 안드로이드와 같은 스마트 디바이스에서 Bluetooth HDP를 통해 블루투스 무선 송수신과 IEEE 11073 PHD 데이터의 활용 방법을 알 수 있다.

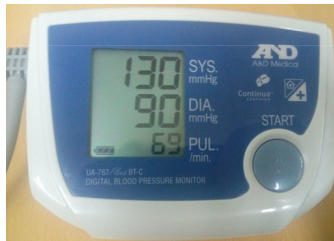


그림 5. 혈압계 상의 결과
Fig. 5. The result of the Blood pressure meter

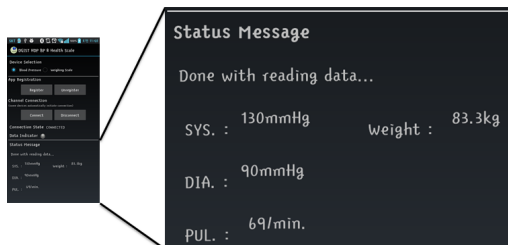


그림 6. 스마트 디바이스 상의 결과
Fig. 6. The result on the smart device

그림 5는 혈압계상에 표시된 결과 값으로 그림 6의 스마트 디바이스로 정확히 전송된 것을 알 수 있다. 그림 6은 본 논문의 실험을 위해 구현된 안

드로이드 앱에서 혈압계를 이용하여 측정 후, 앱 화면에 사용자가 읽을 수 있는 형식으로 디스플레이한 것을 캡처한 그림이다.

실험에서 사용된 혈압계와 체중계는 데이터 송수신을 위해 사용되는 스캐너 클래스의 event report 서비스가 달라 프로그램 상에서 IEEE 11073 PHD 프로토콜 데이터의 분석을 각각 구현 하였다. 실제로 IEEE 11073 관련 표준 문서에는 스캐너 클래스의 고정형 event report 서비스에 대한 프로토콜 송수신 예만 나타나 있다. 실제로 대부분의 의료기기 제조사에서는 데이터 시트를 제공해주지 않는 경우가 많기 때문에 수신된 프로토콜을 직접 분석하여 사용하려는 건강정보측정기기가 어떤 event report 서비스를 이용하고 있는지 확인 후 구현해야 한다.

마지막으로 실험에서 스마트 디바이스와 연동하여 측정에 소요된 시간은 실험 조건에 따라 다르기 때문에 특별한 의미를 가지는 것은 아니다. 다만 참고로 대략적인 시간을 본다면 체중계의 경우 사람이 직접 올라가 측정하였을 경우 약 7초 가량(무게: 84kg) 소요되었고, 가벼운 물건(무게: 20~30kg)을 올려두었을 경우 약 4초 정도 소요되었다. 혈압계의 경우도 커프의 위치 및 기타 측정 환경 및 상황에 따라 다르지만 대략 30초 이상 소요되었다.

IV. 결 론

IEEE 11073 PHD 표준은 헬스케어 서비스 환경에서 개인용 건강기기의 상호운용성을 보장하여 관련 기기의 잠재적인 시장을 성장 시키고, 사람들이 건강관리에 있어서 좀 더 의식 있는 서비스 이용자가 되도록 하여 혼란을 최소화하고 개인건강기기의 신뢰성을 확보하여 헬스케어 서비스 시장을 자연스럽게 활성화시켜 나갈 것이다. Bluetooth HDP 역시 SPP를 기반으로 동작이 이루어지기 때문에 기존 Bluetooth의 연결 및 데이터 송수신 방식과 차이가 없다. 따라서 Bluetooth HDP와 IEEE 11073 PHD는 HL7과 함께 앞으로 개발될 건강정보측정기기와 스마트 디바이스의 어플리케이션은 헬스케어 서비스 구현 시 필수 요소로 자리 잡을 것이다.

앞으로 이미 대중화된 스마트 기기들과 어떻게 호환해야하고, 나아가 더 나은 환경에서 개인 의료 서비스를 제공하는 방법을 찾아나가는 것이 중요한 관건이라고 생각한다. 본 논문에서는 쉽게 사용이 가능하고 어디서든 접할 수 있는 체중계와 혈압계 등을 이용하여 기기별 실험을 진행하였다. 이와 같

은 실험을 통해 의료 데이터의 표준을 조금이나마 이해하고, 스마트 디바이스와의 표준 연동 방법을 파악하여 향후 표준 적용을 확대한다면 국내 건강정보측정기기 및 의료기기 산업의 국제 경쟁력을 높이는데 도움이 될 것으로 기대한다.

향후 과제로는 기기들 간 PnP(Plug and Play) 방식으로 다중 연결 방식을 구현하여 사용자의 편의성을 높이고, 나아가 기기 간 송수신되는 모든 프로토콜을 분석하여 제공할 수 있는 분석 틀을 구현할 예정이다.

References

- [1] Y. J. Park and H. S. Cho, "Exchange method of the HL7 message using open source on OSGi environment," *J. KIISE*, vol. 18, no. 11, pp. 775-779, Nov. 2012.
- [2] D. Y. Hwang, "Implementation of HL7 interface engine for medical information exchange," *J. Korea Soc. Comput. Inform.*, vol. 15, no. 8, pp. 89-98, Aug. 2010.
- [3] S. H. Kim, "Trend of personal health device standardization for health care service," *J. KIISE*, vol.29, no. 1, pp. 31-37, Jan. 2011.
- [4] H. N. Park, S. H. Kim and D. S. Yoo, "Present status and analysis for IEEE 11073 personal health device specializations," *J. KICS*, vol. 37C, no. 6, pp. 469-475, June 2012.
- [5] ISO/IEEE, "*IEEE 11073-20601 specification - Part 20601: Optimized exchange protocol, 1st Ed.*," ISO/IEEE 11073-20601:2010, 2010.
- [6] ISO/IEEE, "*IEEE 11073-20601 specification - Part 10407: Blood Pressure, 1st Ed.*," ISO/IEEE 11073-20601:2010, 2010.
- [7] ISO/IEEE, "*IEEE 11073-10415 specification - Part 10415: Weighing Scale, 1st Ed.*," ISO/IEEE 11073-10415:2010, 2010.
- [8] Continua, "*The next generation of personal telehealth*", Continua Health Alliance, 2009.
- [9] Google, *Android developers-Bluetooth*, Retrieved Feb., 22, 2013, from <http://developer.android.com/guide/topics/connectivity/bluetooth>.

박 영 진 (Young-jin Park)



2004년 8월 경북대학교 컴퓨터공학과 석사
2011년 8월~현재 대구경북과학기술원 IT융합연구부 <관심분야> 스마트 헬스케어, 임베디드 소프트웨어

조 희 섭 (Hui-sup Cho)



2001년 2월 경북대학교 전자공학과 석사
2005년 5월~현재 대구경북과학기술원 IT융합연구부 <관심분야> 전자공학, 의용생체전자공학

손 종 욱 (Jong-wuk Son)



2006년 2월 한국과학기술원 전자공학과 석사
2006년 1월~현재 대구경북과학기술원 IT융합연구부 <관심분야> 무선통신, 디바이스 관리, 의료 IT융합 서비스