

---

# RFID기반의 자가 혈당측정 시스템의 구현

박태진\* · 이종희\*\*

Implementation of RFID based on the self-blood sugar measurement system

Tae-jin Park\* · Jong-hee Lee\*\*

## 요 약

최근 헬스케어 응용 시스템 중의 하나인 자가 진단 시스템 기술의 연구가 활발하게 이루어져 왔다. 이와 함께 본 연구논문은 PC, RFID, 생체신호측정기와 연동된 자가 혈당측정 시스템 및 사용자가 이해하기 쉬운 인터페이스를 구현하는데 있다. 따라서 PC, RFID, 그리고 생체신호측정 센서와 연동되어 손쉽게 자가진단 데이터를 확보할 수 있도록 디바이스와 PC사이의 통신을 위한 손쉬운 인터페이스 구성과 RFID 태그 및 리더기 통신 상태를 분석, 테스트를 수행했다. 즉, 자가 혈당측정 데이터에 대한 통신 프로토콜 수행(프로토콜 정의에 따른 리퀘스트 및 리스폰스 데이터 분석)으로 연월일, 체온 및 혈당수치, 그리고 혈당체크 시점 등 신뢰성 높은 결과가 출력되었음을 나타내었다.

## ABSTRACT

Recently, technology of self diagnosis system which one of health-care has been research very active . With this, In this paper, we have to implement the self-blood sugar measurement system that is interlock with PC, RFID, and Bio-Signal glucose tester and accessible user interface. Therefore the equipments can performs accessible interface composition for communication between devices and PC, analysis and test of communication status of the RFID tag and reader that it easy to obtain the data of self-diagnosis linkage with PC, RFID, and Bio-Signal glucose tester. In other words, as result of communication protocol experiment with self-blood sugar measurement data, it has been showed to display a high-reliability result through year, month, and day, body temperature, and levels and check point of blood glucose.

## 키워드

혈당측정 시스템, 혈당측정 프로토콜, RFID기반 생체신호측정기 연동

## Key word

System of self-blood sugar measurement Protocol of blood sugar measurement,  
Interlocking of RFID based on Bio-Signal glucose tester

---

\* 정회원 : 신라대학교 컴퓨터공학과 외래교수  
\*\* 종신회원 : 신라대학교 컴퓨터공학과 (교신저자, jhlee@silla.ac.kr)

접수일자 : 2011. 05. 31  
심사완료일자 : 2011. 06. 20

## I. 서 론

한국 산업사회에서의 국민건강 증진과 관련해서 한국산업안전공단 및 노동부에서는 고혈압, 당뇨병, 고지혈증, 간질화, 비만 등을 6대 건강생활 실천으로 규정한다.

IT 기반 융합기술의 발전과 u-헬스케어 서비스를 위한 인프라 구축은 만성질환자의 증가 및 소득 수준의 향상과 함께 새로운 의료서비스에 대한 욕구가 늘어나고 있다[1]. 또한, 최근에는 차세대 이동통신의 개발과 개인 휴대용 단말기를 통한 쌍방향 및 IP기반의 고속 무선통신은 다양한 서비스 지원과 함께 누구나 쉽게 질적 자가 진단 의료서비스를 누릴 수도 있다. 본 연구논문에서는 RFID 기반인 자가 혈당측정 시스템 구현을 위해서 활용이 손쉬운 인터페이스 구현 및 통신 프로토콜을 정의하고 동작 상태를 분석하게 되며, 이와 같이 수집된 자료는 건강진단 의료서비스 플랫폼을 구축해 나가는데 중요한 역할을 하게 된다. 즉, PC, RFID, 그리고 혈당측정 센서와 연동되어 손쉽게 자가진단 데이터를 확보할 수 있으며, RFID 태그 및 리더기 통신 상태 분석과 동시에 데이터 전송 단위 및 혈당계 프로토콜 테스트 수행으로 자가진단 목적의 실시간 의료지원시스템에 대한 신뢰성 있는 통신 프로토콜과 인터페이스 구현에 있다.

## II. 관련 연구 분석 및 기술

관련 연구 분석에는 헬스케어 서비스와 관련된 기존 연구들을 분석하고 RFID를 이용한 헬스케어 애플리케이션 요구사항을 살펴보고자 한다[2,3,4,5,6,7].

### 2.1. 기존 연구 분석

모바일 헬스케어와 관련된 기존 방식의 취약성을 분석하면, 기존의 방식들은 모바일 상에서의 헬스케어를 위한 모델만을 제시했다는 한계점과 특정한 인프라만 구성했다는 점에서 새로운 모델에의 적용에 있어서 어려운 한계점을 가지고 있기도 한다[8]. 또한, RFID를 이용한 의료정보시스템은 주로 대형 병원에서 사용될 수 있으며, 다양한 비즈니스 모델 구현과 식별이 요구되는 데이터 관리를 위해서 구축되기도 한다. 아직까지 서류

에 의존해서 데이터를 보관하고 있는 경우가 대부분인 현실에서 또 다른 자가 진단을 위한 기존 의료장비들은 환자에 의한 단순 의료상식과 지식에 의해서 잘못된 판단을 할 수 있다는 점이다.

### 2.2. RFID를 이용한 헬스케어 시스템

장기간 입원해 있는 환자 중심으로 프로토콜을 구성하는 것으로 모든 객체, 즉 환자, 의사, 간호사 등 RFID 태그가 내장된 객체를 소유한다. 여기에 병원의 통합관리 시스템은 유선망으로 구성된 시스템이다. 이와 같이 제안된 시스템은 환자 주변에 RFID를 감지하기 위한 센서 및 리더기가 설치되고, 환자의 위치검색 및 다양한 물건을 검색, 간호사에게 필요에 따른 메시지를 전송한다. 또한, 간호사가 소유하는 PDA는 간호사의 고유 식별 정보가 내장된 RFID 태그가 포함된다. PDA는 애플리케이션이 탑재될 수 있는 환경으로써, 환자의 태그를 통해 읽어 들인 데이터를 확인하고, 그 이외의 절차들을 수행하기 위해서 유저 인터페이스형 애플리케이션을 탑재, 간호사가 가지고 있는 PDA 및 애플리케이션은 센서를 통해 RFID 태그를 읽어 들일 수 있는 반경으로 하나의 셀 안에 구성된다. 추가해서 의사를 통한 진료와 진료를 받은 후의 사후 처방에 대한 모든 정보를 저장하고 관리하는 것으로 한다[8]. 그러나 이 모든 시스템은 병원 구조적 문제 및 경제적 문제를 고려할 수 있어야 하며, 병원 통합시스템에 대한 가상된 시나리오 중심이어서 실용화되고 응용화 되기에는 유용하지 못하다.

## III. 자가 혈당측정 시스템

### 3.1. 혈당측정 시스템의 구성과 동작절차

본 시스템의 구성과 동작절차를 살펴보면, 시리얼 통신 모듈이 내장된 리더기와 혈당측정기를 USB포트에 사용하여 PC(호스트)와 연결하고 RFID 태그의 고유번호인 UID(User Identification)를 인식함으로써 시스템에 로그인되며, PC기반에서의 시리얼 통신을 통해서 혈당계로 데이터 전송 요청, 동시에 명령어를 전송하는 것으로 혈당계는 명령어(COMMAND) 및 레코드번호를 확인하고 기기에 저장된 데이터 값을 PC로 전송하게 된다. 그림 1은 디바이스와 PC사이의 통신을 위한 데이터 흐름을 나타낸 것이다.

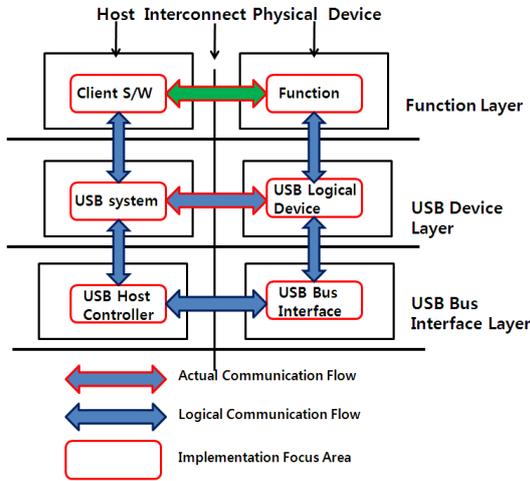


그림 1. USB 시스템의 데이터 흐름  
Fig. 1 Data flow of USB system

PC에서의 실제 데이터 흐름은 클라이언트 소프트웨어에서 USB 시스템 소프트웨어(USB 드라이버, USB 호스트 컨트롤러 드라이버)를 거친 다음 USB 호스트 컨트롤러에서 전기적 신호가 USB 버스 인터페이스를 통해 디바이스 함수 계층까지 전달된다.

3.2. 통신 프로토콜 정의

혈당계는 RS-232의 직렬전송 프로토콜 방식을 사용하며, 데이터 구조는 표 1과 같이 STX, SIZE, ~SIZE, COMMAND, DATA, Checksum(CKL, CKH)으로 정의했다. 통신환경은 보오율(Baud Rate)이 초당 9600비트, 패리티 비트는 'None', 정지 비트(Stop Bit)는 1비트, 데이터 길이는 8비트이다.

표 1. 혈당계의 데이터 구조  
Table. 1 Data structure of glycemc measurement

| NAME    | STX   | SIZE     | ~SIZE  | COMMAND | DATA       | Checksum (Low) | Checksum (High) |
|---------|-------|----------|--------|---------|------------|----------------|-----------------|
| Byte(s) | 1     | 1        | 1      | 1       | N          | 1              | 1               |
| VALUE   | 0x80  | N+1      | ~(N+1) |         | DATA Field | *              | **              |
| REMARK  | START | CMD+DATA |        |         | Data       |                |                 |

표 1의 STX는 전송 시작을 나타내며, SIZE는 명령어와 데이터의 크기, DATA는 데이터 필드, COMMAND는 명령어로서 상세한 내용은 표 2와 같으며, CKL(Checksum Low)는 ~(STX ^ ~SIZE ^ DATA[1] ^ DATA[3] ...)로 표시되고, CKH(Checksum High)는 ~(SIZE ^ COMMAND ^ DATA[2] ...)로 표시한다.

표 2. 혈당계의 명령어  
Table. 2 Command of glycemc measurement

| No | COMMAND | REQUEST COMMAND                 | RE MARK |
|----|---------|---------------------------------|---------|
| 1  | 0x00    | Total number of records request | -       |
| 2  | 0x01    | One record request              | -       |
| 3  | 0x02    | Code setting                    | -       |
| 4  | 0x03    | Data setting                    | -       |
| 5  | 0x04    | Unit setting                    | -       |
| 6  | 0x05    | Average day setting             | -       |
| 7  | 0x06    | Alarm setting                   | -       |
| 8  | 0x07    | Device ID setting               | -       |
| 9  | 0x08    | Device ID request               | -       |
| 10 | 0x09    | Software version request        | -       |
| 11 | 0x0A    | Power off request               | -       |

표 2에서 명령어 0x00은 혈당계에 저장된 측정데이터의 총 개수를 읽어오기 위함이며, 0x01은 혈당계에 저장된 데이터 중 레코드 번호(가장 최근에 측정된 데이터 순)를 선택해서 해당 데이터를 읽어오기 위한 구조이다. 본 연구논문에서 구현된 시스템은 가장 최근에 측정된 데이터만 PC로 전송하기 때문에 이에 해당하는 명령어인 0x01(ONE RECORD REQUEST)만 사용하도록 했다.

3.2.1. 리퀘스트/리스폰스 데이터 필드

리퀘스트/리스폰스(Request/Response) 데이터 필드는 명령어(0x01)를 전송할 때의 형식으로 그림 2와 같다.

리퀘스트는 데이터 전송을 위해서 PC에서 혈당계로 요청할 때의 형식이며, 데이터 전송 명령어는 0x01값으로 한다. 또한, 데이터에는 혈당계에 저장된 목록의 레코드 번호를 설정해서 전송되도록 구현했다. 혈당계에 저장된 목록의 번호는 가장 최근에 측정된 데이터 순으로 0x00(0)부터 0xF9(249)까지 기록 할 수 있다. 개발 시스

템에서는 가장 최근에 측정된 데이터를 전송해야 하므로 데이터에는 0x00으로 설정했다.

| NAME    | STX   | SIZE      | ~SIZE  | CMD | DATA       | CKL | CKH |
|---------|-------|-----------|--------|-----|------------|-----|-----|
| Byte(s) | 1     | 1         | 1      | 1   | N          | 1   | 1   |
| VALUE   | 0x80  | N+1       | ~(N+1) |     | DATA Field | *   | **  |
| REMARK  | START | CMD+ DATA |        |     | Data       |     |     |

| NAME    | COMMAND | DATA          |
|---------|---------|---------------|
| Byte(s) | 1       | 1             |
| VALUE   | 0x01    | Refer to **   |
| REMARK  | Command | Record number |

| NAME    | COMMAND | DATA          | VAL                   |
|---------|---------|---------------|-----------------------|
| Byte(s) | 1       | 1             | 6                     |
| VALUE   | 0x01    | Refer to **   | Refer to Record Field |
| REMARK  | Command | Record number | Value of Record       |

| Bit     | 15   | 14   | 13   | 12     | 11   | 10   | 9   | 8 | 7   | 6   | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|------|------|------|--------|------|------|-----|---|-----|-----|---|---|---|---|---|---|
| 1,2byte | YEAR |      |      |        |      |      | MON |   |     | DAY |   |   |   |   |   |   |
| 3,4byte | TEMP |      |      | RESULT |      |      |     |   |     |     |   |   |   |   |   |   |
| 5,6byte | PLAS | HEAL | EXER | STRE   | ATTN | HOOR |     |   | MIN |     |   |   |   |   |   |   |

그림 2. 리퀘스트/리스폰스 데이터 필드 형식  
Fig. 2 Format of data field for request/response

리스폰스는 데이터 전송요청 명령어를 받은 혈당계가 PC로 데이터 전송할 때의 형식을 가지며, 명령어에는 리퀘스트와 동일하게 0x01값으로 설정한다. 데이터 필드는 DATA와 VAL값을 합친 것으로 DATA에는 리퀘스트와 동일하게 0x00값을, VAL에는 최근에 측정된 데이터를 실어 전송하도록 구현했다.

| STX    | 데이터 길이 | n문자 데이터부                                | ETX | BCC |
|--------|--------|---|-----|-----|
| 데이터    | 문자수    | 내용                                      |     |     |
| STX    | 1      | 통신 데이터의 선두 제어 코드(02h)                   |     |     |
| 데이터 길이 | 1      | 데이터부만이 가지는 문자수의 합계 (바이너리)               |     |     |
| 데이터부   | n      | 각 명령어의 파라미터                             |     |     |
| ETX    | 1      | 통신 데이터의 말미 제어 코드(03h)                   |     |     |
| BCC    | 1      | 데이터길이, 데이터부 끝부분의 값을 각 논리합(XOR)을 처리된 데이터 |     |     |

그림 3. RFID 리더기의 데이터 구조  
Fig. 3. Data structure of RFID reader

그림 3의 데이터부는 RFID 리더에서 RFID 태그로 명령을 보내는 명령어와 RFID 태그에서 RFID 리더로 응답을 하는 리스폰스 상태로 구분한다. 데이터 전송을 위한 단계는 총 4가지(1.명령어 송신, 2.ACK 또는 NAK 응답, 3.리스폰스 명령어 송신, 4..데이터 수신에 대한 ACK 또는 NAK 응답)로 구성된다. RFID 리더기(Host)와 RFID 태그(R/W)간의 정상적인 통신 순서는 그림 4와 같다.

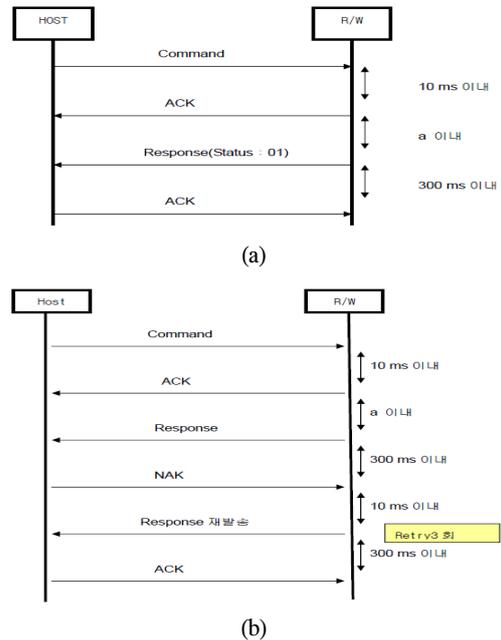


그림 4. RFID 리더기와 RFID 태그 간 통신 순서  
(a) 정상상태 절차 (b) 이상상태 절차  
Fig 4. Communication order between RFID reader and RFID tag (a) Normality processing (b) Abnormality processing

RFID 리더기와 RFID 태그 간 통신함에 있어서 ‘오류’가 발생했다면, RFID 태그는 RFID 리더기로부터 명령어를 수신하고 10ms 이내에 ACK(06h) 또는 NAK(15h)를 돌려준다. RFID 태그는 ACK를 송신하고 명령어에 의해서 처리를 한 후, 리스폰스를 RFID 리더기에 송신한다. RFID 태그는 NAK를 송신 후, 리스폰스를 ‘오류’라 표시하고 RFID 리더기에 송신한다. RFID 태그는 리스폰스를 송신하고 RFID 리더기로부터의 리스폰스를 기다린다. RFID 리더기로부터 NAK의 리스폰스를 수신하

면, 리스폰스를 재발송한다. 리스폰스의 재발송 처리를 최대 3회까지 허용하고, 그 이상이면 명령어 대기 상태가 된다.

3.2.2. RFID 리더기의 명령어

RFID 리더기의 Command목록 중 주로 사용되는 명령어를 다음과 같이 설정했다.

| No | Command | 내용   |
|----|---------|------|
| 1  | 읽기      | 'RB' |
| 2  | 기록      | 'WB' |
| 3  | UID 읽기  | 'IV' |

추가해서 사용자가 시스템에 접근하기 위해 RFID 태그를 인식하는 과정에 필요한 RFID 리더기의 명령어(IV)에 대해서 전송된 데이터 형식을 그림 5와 같이 설정했다.

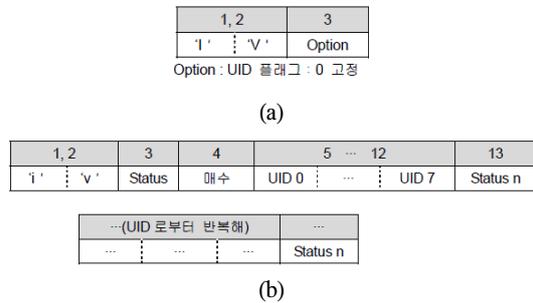


그림 5. RFID 태그에서 RFID 리더기로 전송되는 데이터 형식 (a) 개괄적 데이터 프레임 (b) 상세 데이터 프레임  
Fig. 5 Form of data transmitted from RFID tag to RFID reader (a) General data frame (b) Detailed data frame

그림5에서의 상태(Status)에서 바이너리 값으로 0x01h일 때는 종료 명령어를 나타내며, 그 외의 값은 '오류'가 있다는 의미로 설정했다. 매수는 인식된 RFID 태그의 매수를, UID는 읽어낸 RFID 태그의 고유번호, 상태 n(Status n)은 RFID 태그와의 통신 상태 값으로 설정했다.

IV. 프로토콜 분석 및 테스트

혈당계는 측정이 완료된 후, 장치의 화면에 'PC' 글자가 나타나면 데이터를 전송하도록 구현했다. 단, 글자가 나타났을 때 데이터 전송을 요청하는 명령어(One Record Request)를 보내주어야만 PC로 전송하게 되어있다. 따라서 혈당계를 사용해서 혈당을 측정하는 동안 PC에서는 'One Record Request' 명령어를 실시간으로 보내주고, 측정이 완료되어 기기에서 'PC' 글자가 나타나면 혈당계는 전송 요청 명령어를 확인하고 즉시 PC로 측정된 데이터를 전송하도록 구현했다. 표 3은 측정된 데이터(VAL)에 대한 내용을 상세히 나타낸 것으로, 옵션(Optional)은 항상 00h, 03h, 07h만을 지정해야 하고 그 이외는 설정 에러 21h를 반환, 명령어에 의해 설정값이 달라지도록 한다.

표 3. Command(Host로부터 R/W에 대해서 Command 송신)  
Table. 3 Command(Command sending of R/W from HOST)

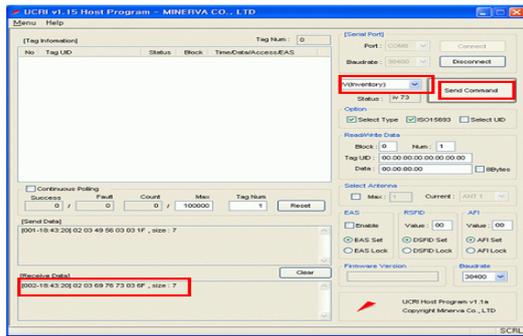
| Command | Option | Data 1   | ..... | Data n |
|---------|--------|--|-------|--------|
| Data    | 문자수    | 내용   |       |        |
| Command | 2      | RW에 대해서 요구하는 명령어(ASCII)  |       |        |
| Option  | 1      | Command Option(바이너리)<br>비트0: Tag Type Select flag<br>(0:Tag지정없음, 1:Tag지정있음)<br>비트1: Tag flag(0:I-CODE1,1:I-CODE2)<br>비트2: UID flag(0:UID지정없음,<br>1:UID지정있음)<br>비트3~비트7: 예비 |       |        |
| Data    | n      | 각 명령어(Command)의 파라미터   |       |        |

4.1. 데이터 전송 단위 테스트

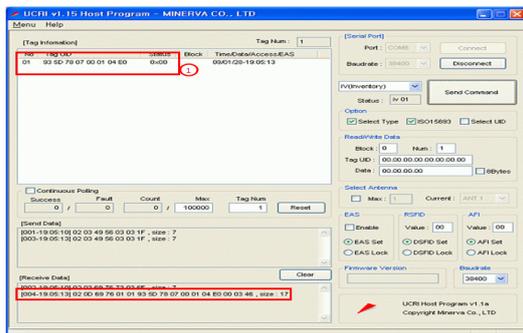
RFID 리더기를 PC에 연결하면 장치관리자에서 COM포트가 인식됨을 확인 할 수 있는데, 여기서 RFID 리더기의 포트 번호는 COM8로 하고 보오율은 38,400으로 설정한다. 본 연구논문에서는 RFID 리더기와 RFID태그 간에 전송되는 데이터 확인 및 프로토콜을 분석하기 위해서 PORTMON 프로그램을 사용했다. PORTMON은 시리얼통신 감시프로그램으로 COM포트를 통해서 주고받는 메시지를 확인 할 수 있다.

### 4.2. 혈당계 프로토콜 실험 및 테스트

RFID 리더기에 대한 프로토콜 분석 및 테스트 진행을 위해서 미네라바의 UCRI 프로그램을 사용했으며, RFID 태그를 RFID 리더기에 대지 않고 명령어를 전송한 경우와 RFID 태그를 RFID 리더기에 접촉 후, 명령어 전송한 경우를 통해서 태그의 고유번호(UID) 출력상태를 분석하고 테스트 한다.



(a)



(b)

그림 6. 태그의 정보탭에 표시되는 데이터  
 (a) 태그의 정보탭에 보이지 않은 데이터  
 (b) 태그의 정보탭에 보인 데이터

Fig. 6. Data that shows in information tab of tag  
 (a) Not data shows information tab of tag  
 (b) Data shows information tab of tag

그림 6(a)의 경우를 살펴보면, 태그의 정보탭에 아무런 정보도 나타나지 않았음을 알 수 있고, 그림6(b)는 ①의 탭 정보와 같이 RFID 태그 고유번호(UID)를 확인 할 수 있다. 이때의 수신데이터는 RFID 프로토콜의 프레임 포맷에 따라서 RFID 리더기에서 RFID 태그로 전송하는 데이터의 형식과 RFID 리더기에서 RFID 태그를 정상적

으로 인식했을 때의 전송된 데이터를 표현했다. 시스템의 로그인 과정에서는 RFID 태그의 고유번호만 인식하면 되므로 고유번호의 전송 요청 명령어인 IV명령어만 필요하다. 따라서 RFID 태그가 인식되지 않은 상태에서 'IV' 명령어를 전송했을 경우, 데이터부의 마지막 바이트에는 RFID 태그 부재오류에 대한 상태 값인 0x73을 설정, 전송하도록 설정했다.

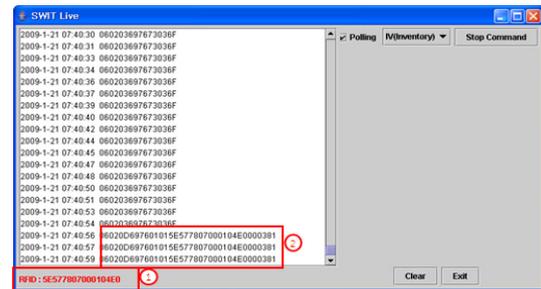


그림 7. RFID 태그와 RFID 리더기 간 전송된 데이터의 출력

Fig. 7 Output of transmitted data between RFID tag and RFID reader

그림 7은 JAVA로 구현한 리더기의 데이터 분석 프로그램인 GUI이다. 이 프로그램은 RFID 태그와 RFID 리더기 간 전송된 데이터를 화면상으로 출력해주며, RFID 태그가 인식되면 ②와 같이 RFID 리더기로 전송된 데이터를 출력하고 프로토콜에 따라 데이터를 분석해서 ①과 같이 출력해준다. ①에 출력된 값은 RFID 태그의 UID로 이 값을 임시로 저장한 후, 시스템에 등록된 사용자와 일치하는 사용자의 ID값을 찾아 로그인 할 수 있도록 한다.

|         |   |       |         |      |     |     |
|---------|---|-------|---------|------|-----|-----|
| 80      | 02                                      | FD    | 01      | 00   | 82  | FC  |
| STX     | SIZE                                    | ~SIZE | COMMAND | DATA | CKL | CKH |
| STX     | 0x80                                    |       |         |      |     |     |
| SIZE    | COMMAND 1byte+ DATA 1byte= 2byte = 0x02 |       |         |      |     |     |
| ~SIZE   | ~0x02 = FD                              |       |         |      |     |     |
| COMMAND | 0x01(One record request 명령어)            |       |         |      |     |     |
| DATA    | 0x00(Record number)                     |       |         |      |     |     |
| CKL     | ~( STX ^ ~SIZE ^ DATA[!]) = 0x82        |       |         |      |     |     |
| CKH     | ~( SIZE ^ COMMAND) = 0xFC               |       |         |      |     |     |

그림 8. 리퀘스트 데이터의 출력

Fig. 8 Output of request data

그림 8의 리퀘스트 데이터에서 크기는 명령어와 데이터 크기를 합친 값이며, 명령어(COMMAND)에는 데이터 전송 요청 명령어인 ‘One Record Request’ 명령어 값 0x01로 설정되어 있다. 리퀘스트에서의 데이터(DATA)는 가장 최근에 측정된 데이터의 레코드번호(Record number)인 0x00으로 설정한다. CKL과 CKH는 체크섬으로 CKL은  $\sim(\text{STX} \wedge \sim \text{SIZE} \wedge \text{DATA}[1] \wedge \text{DATA}[3] \dots)$ 의 형식으로 구성되고 CKH는  $\sim(\text{SIZE} \wedge \text{COMMAND} \wedge \text{DATA}[2] \dots)$ 의 형식으로 계산되고 설정된다.

|     |    |    |      |       |         |      |    |    |    |    |     |     |
|-----|----|----|------|-------|---------|------|----|----|----|----|-----|-----|
| 80  | 08 | F7 | 01   | 00    | 12      | 42   | 58 | 34 | 03 | 95 | 6B  | BF  |
| STX |    |    | SIZE | ~SIZE | COMMAND | DATA |    |    |    |    | CKL | CKH |

|         |   |
|---------|---|
| STX     | 0x80  |
| SIZE    | COMMAND 1byte+ DATA 7byte = 8byte = 0x08  |
| ~SIZE   | ~0x08 = 0xF7  |
| COMMAND | 0x01(One record request 명령어)  |
| DATA    | 0x00(record number)<br>나머지 값은 VAL(측정된 데이터)  |
| CKL     | $\sim(\text{STX} \wedge \sim \text{SIZE} \wedge \text{DATA}[1] \wedge \text{DATA}[3] \wedge \text{DATA}[5] \wedge \text{DATA}[7]) = 0x6B$ |
| CKH     | $\sim(\text{SIZE} \wedge \text{COMMAND} \wedge \text{DATA}[2] \wedge \text{DATA}[4] \wedge \text{DATA}[6]) = 0xBF$                        |

그림 9. 리스폰스 데이터의 출력  
Fig. 9 Output of response data

그림 9의 리스폰스의 데이터 크기 또한, 명령어(COMMAND)와 데이터(DATA)의 크기를 합친 값이며, 리퀘스트 명령어와 같이 0x01로 설정되어 있다. 리스폰스의 데이터에서 첫 번째 바이트는 리퀘스트 데이터와 동일하게 가장 최근에 측정된 데이터의 번호인 0x00으로 설정된다. 0x00을 제외한 나머지, 즉, 그림 10의 붉은 색으로 표시된 6바이트 값은 측정된 데이터(VAL)를 나타낸다. 혈당계의 프로토콜에 따라서 측정된 데이터는 그림 10과 같다.

전송된 16진수 값 12, 42, 58, 34, 03, 95를 2바이트로 나누어 세 그룹으로 분류하고 16진수를 이진수로 변환, 나열한다. 그러면 16비트씩 나누어지게 된다. 즉, 12, 42는 00010010 01000010으로 변환하여 7bit에 해당하는 값인 0001001(9)을 YEAR로, 4bit에 해당하는 0010(2)을 MON로, 5bit에 해당하는 00010(2)을 DAY로 구분한다. TEMP(6bit), RESULT(10bit), EVENT VALUE(5bit), HOUR(5bit), MIN(6bit)도 동일한 방식으로 계산하여 측

정된 데이터로 변환한다. 이벤트 값은 사용자의 현재 측정상태를 나타내는 값으로 식후, 운동 후, 스트레스, 주의를 나타내지만 현재 혈당계에는 식전, 식후의 상태만 선택 할 수 있도록 했다. 만약 식전이라면 00000값을, 식후라면 00001의 값으로 전송된다.

|    |    |    |    |    |    |
|----|----|----|----|----|----|
| 12 | 42 | 58 | 34 | 03 | 95 |
|----|----|----|----|----|----|

| 6byte(HEX)                | Binary     | Bit | Decimal | Field     | 측정결과    |
|---------------------------|------------|-----|---------|-----------|---------|
| (12 42) 00010010 01000010 | 0001001    | 7   | 9       | YEAR      | 2009년   |
|                           | 0010       | 4   | 2       | MON       | 2월      |
|                           | 00010      | 5   | 2       | DAY       | 2일      |
| (58 34) 01011000 00110100 | 010110     | 6   | 22      | TEMP      | 22°C    |
|                           | 0000110100 | 10  | 52      | RESULT    | 52mg/dL |
|                           | 0000       | 5   | 0       | EVENT_VAL | 식전      |
| (03 95) 00000011 10010101 | 01110      | 5   | 14      | HOUR      | 오후2시    |
|                           | 010101     | 6   | 21      | MIN       | 21분     |

그림 10. 레코드 필드에서의 실제 측정된 데이터  
Fig. 10 VAL data of record fields

## V. 결론

본 연구논문에서 PC, RFID, 혈당측정 센서와 연동되어 손쉽게 자가진단 데이터를 확보하는데, 먼저, 프로토콜 정의와 구현된 절차에 따라서 RFID태그 및 리더기 통신 상태를 분석했으며, 다음은 생체신호측정기로부터 전송된 데이터 평가를 위해서 세 그룹으로 분류 후, 실험 및 측정결과를 보이도록 했다. 즉 프로토콜 정의에 따른 리퀘스트 및 리스폰스 데이터를 분석, 그 결과가 연월일, 체온 및 혈당, 그리고 혈당체크 시점 등 세 그룹에 대해서 신뢰성 있는 데이터가 출력되었음을 나타내는데 있다. 결과적으로는 자가진단을 목적으로 하는 통신 프로토콜 정의와 구현, 그리고 활용이 손쉬운 인터페이스 구현 및 동작 상태를 분석함으로써 건강진단 의료서비스 플랫폼 구축에 효과적임을 보였다. 이와 같이 응용된 자가진단 의료서비스 활동은 앱서비스를 통해서 개별적 질병예측과 예방활동을 적극적으로 수행할 수 있게 되며, 구축된 의료 종합정보가 가지는 데이터의 보안성을 확보하고 인증된 의료서비스 제공자에게는 환자 진단, 치료에 대한 지능적 의사결정 기준 제공과 생체정보 측정 결과에 대한 통계정보를 제공할 수 있다. 따라서 해당 질병 예측 및 치료 효율성 등 향상된 의료서비스가 가능하게 된다.

참고문헌

[1] 김순창, “의료기기 국제규격의 현황조사 및 분석 연구,” 식품의약품안전청, 2007

[2] Dimitri Konstantas and Val Jones, "MobiHealth-innovative 2.5/3G mobile services and applications for healthcare," NEC Research, 2002.

[3] Healthmate: <http://www.Healthmate-Project.org>.

[4] I. Widya and A. van Halteren, "Telematic Requirements for a Mobile and Wireless Healthcare System deviced from Enterprise Models," EC Programme 1ST, 2002.

[5] MobiHealth,, "MobiHealth project IST-2001-3 6006," EC programme IST, 2002: <http://www.mobihealth.org>.

[6] Ramon Marti and Jaime Delgado, "Security in a Wireless Mobile Health Care System," NEC Research, 2000.

[7] Salla Kalaja, "Security in Mobile Health Care Work," Tik-110.501 Seminar on Network mSecurity, 2000.

[8] 백장미, 홍인식, "RFID를 이용한 효율적인 환자관리 애플리케이션 시스템 개발에 관한 연구," 한국멀티미디어학회 논문지, 제8권, 8호, 1142-1151쪽, 2005년 8월

이종희(Jong-hee Lee)



1984년 경북대학교 전자공학과  
전산공학(공학석사)  
1990년 경북대학교 전자공학과  
전산공학전공(공학박사)

1979년 ~1987년 경남정보대학 전자계산과  
1988년 ~현재 신라대학교 컴퓨터공학과 교수  
※관심분야: 신경회로망 설계, 영상처리, 컴퓨터교육

저자소개



박태진(Tae-Jin Park)

1983 동의대학교 물리학과 이학사  
1995 부경대학교 전산정보학과  
이학석사  
2008 부경대학교 전자계산학과  
이학박사

2000 ~ 2003 거제대학 전자계산학과 초빙전임강사  
2005 ~ 2007 마산대학 조선메카트로닉스과 강의전담  
2000 ~ 현재 신라대학교 컴퓨터정보학부 외래교수  
2008 ~ 현재 한국폴리텍VII대학 메카트로닉스학과  
외래교수  
2010 ~ 현재 (주)4D Zone, 사외기술이사  
※관심분야 : 영상처리, 신호처리, 멀티미디어정보처  
리, 임베디드시스템