
남서울대학교 정보통신공학과 졸업작품 상세설명서

모바일 헬스 케어

Mobile Health Care



Department of
Information and Communication Engineering

Namseoul University

Revision History

Revision	Description
1.00	조장: 장지용, 조원: 이진영, 김묘진

목 차

1. 개발환경 구축	1
1.1 하드웨어 개발환경	1
1.2 소프트웨어 개발환경	2
1.3 오픈 솔루션(회로도, 소스코드 등) 분석	3
2. 제안 시스템 및 핵심기술	4
2.1 서비스 모델	4
2.2 시스템 모델	5
2.3 핵심 솔루션	6
3. 구현 및 실험 결과	7
3.1 테스트결과	7
3.2 구현결과	8
4. 결 론	9
5. 참고문헌	10

1. 개발환경 구축

1.1 하드웨어 개발환경

□ 아두이노 메가: 싱글보드 컴퓨터 (모델명 : Arduino MEGA)

○ 출처:

<http://vctec.co.kr/product/%EC%95%84%EB%91%90%EC%9D%B4%EB%85%B8-%EB%A9%94%EA%B0%802560-r3-arduino-mega-2560-r3/771/>



<그림 1. Arduino MEGA>

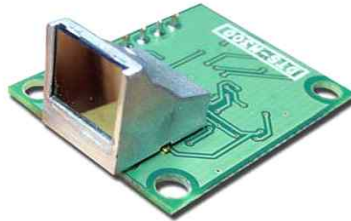
□ 초음파 센서: 키 측정용 센서 (모델명: HC-SR04)

○ 출처: <http://deneb21.tistory.com/215>



<그림 2. HC-SR04>

-
- 적외선 센서: 체온 측정용 센서 (모델명: DTS-M300)
 - 출처: <http://diwellshop.com/product/dts-m300/86/>



<그림 3. DTS-M300>

- 심박센서: 맥박 측정용 센서 (모델명: XD-58C)
 - 출처: http://iotcommunitymk.org/en_US/%D0%9F%D1%83%D0%BB%D1%81-%D0%BD%D0%B0-%D1%81%D1%80%D1%86%D0%B5-%D1%81%D0%BE-xd58c-%D1%81%D0%B5%D0%BD%D0%B7%D0%BE%D1%80/

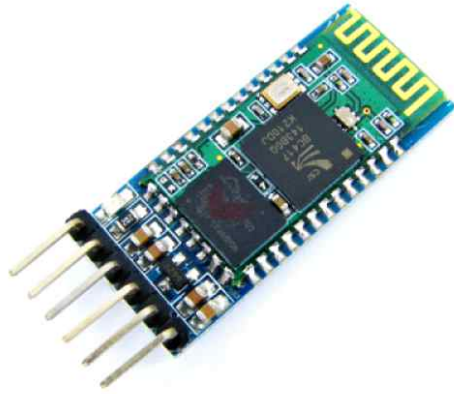


<그림 4. XD-58C>

□ 블루투스센서: 블루투스 통신 (모델명: HC-06)

○ 출처:

<https://smartstore.naver.com/ic11401/products/558894557?NaPm=ct%3Djm7nssk8%7Cci%3D8577a2a411e917812e70829221c833fda428d600%7Ctr%3Dsls%7Csn%3D434525%7Chk%3D7de68dbb39be116f6b45bdb5208d47722171154b>



<그림 5. HC-06>

□ Android Smart Phone: 삼성 갤럭시 S6 (모델명: SM-G920S)

○ OS 버전: Android 7.1.1 (Nougat)

○ 출처: <http://developer.android.com/index.html>



<그림 6. Android Smart Phone>

1.2 소프트웨어 개발환경

- 애플리케이션 개발
 - 개발 도구 : MIT app Inventor
 - <http://appinventor.mit.edu/explore/>
 - 개발 언어 : C/C+ , JAVA
 - 아두이노의 기본 개발 언어, 앱 개발 언어

1.3 오픈 솔루션(회로도, 소스코드 등) 분석

□ 출처: <http://deneb21.tistory.com/215>

□ 소스코드 분석 : 주석처리

○ 초음파 센서

<pre>//출력핀(trig)과 입력핀(echo) 연결 설정, 다른 핀을 연결해도 됨 int trigPin = 9; int echoPin = 8; //시리얼 속도설정, trigPin을 출력, echoPin을 입력으로 설정 void setup(){ Serial.begin(9600); pinMode(trigPin, OUTPUT); pinMode(echoPin, INPUT); } //초음파를 보낸다. 다 보내면 echo가 HIGH (신호받기) 상태로 대기 void loop(){ float duration, distance; digitalWrite(trigPin, HIGH); delay(10); digitalWrite(trigPin, LOW); // echoPin 이 HIGH를 유지한 시간을 저장 한다.</pre>	<pre>duration = pulseIn(echoPin, HIGH); // HIGH 였을 때 시간(초음파가 보냈다가 다시 들어온 시간)을 가지고 거리를 계산 한다. // 340은 초당 초음파(소리)의 속도, 10000은 밀리세컨드를 세컨드로, 왕복거리이므로 2로 나뉜다. distance = ((float)(340 * duration) / 10000) / 2; //시리얼모니터에 Echo가 HIGH인 시간 및 거리를 표시해준다. Serial.print("Duration:"); Serial.print(duration); Serial.print("\nDistance:"); Serial.print(distance); Serial.println("cm\n"); delay(500); }</pre>
---	---

- 초음파의 속도는 340m/s 이다.
- 초음파가 발생되어 대상물체에 부딪히게 되면 반사되어 돌아오게 된다. 이의 시간차를 계산하면 거리를 계산할 수 있다.
- Trig가 초음파를 발생하는 신호이며, Echo가 반사되어 오는 초음파를 받는 역할을 한다.

1.3 오픈 솔루션(회로도, 소스코드 등) 분석

□ 출처: <http://diwellshop.com/product/dts-m300/86/>

□ 소스코드 분석 : 주석처리

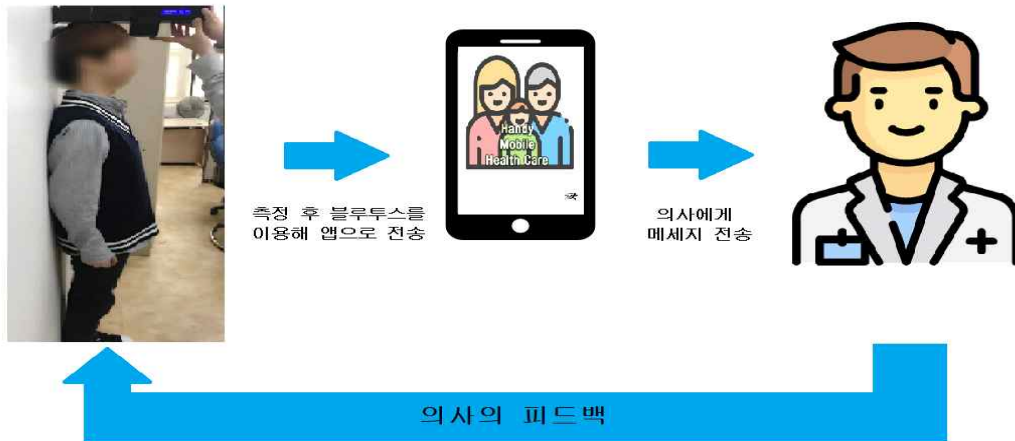
○ 적외선 온도센서

<pre> #include<SPI.h> #define TARGET_CMD 0xA0 // 대상 온도 커맨드 #define SENSOR_CMD 0xA1 // 센서 온도 커맨드 const int chipSelectPin = 10; unsigned char T_high_byte; unsigned char T_low_byte; int iTARGET, iSENSOR; // 부호 2byte 온도 저장 변수 void setup() { /* Initialize PORT */ pinMode(MISO, INPUT); pinMode(chipSelectPin, OUTPUT); pinMode(MOSI, OUTPUT); pinMode(SCK, OUTPUT); Serial.begin(9600); /* Setting CS & SPI */ digitalWrite(chipSelectPin, HIGH); // CS High Level SPI.setDataMode(SPI_MODE3); // Setting SPI Mode SPI.setClockDivider(SPI_CLOCK_DIV16); // 16MHz/16 = 1MHz SPI.setBitOrder(MSBFIRST); // MSB First SPI.begin(); // Initialize SPI delay(500); // wating for DTS setup time } </pre>	<pre> int SEND_COMMAND(unsigned char cCMD) { digitalWrite(chipSelectPin, LOW); // CS Low Level delayMicroseconds(10); // delay(10us) SPI.transfer(cCMD); // Send 1st Byte delay(10); // delay(10ms) T_low_byte = SPI.transfer(0x22); // Send 2nd Byte delay(10); // delay(10ms) T_high_byte = SPI.transfer(0x22); // Send 3rd Byte digitalWrite(chipSelectPin, HIGH); // CS High Level return (T_high_byte<<8 T_low_byte); // 상위, 하위 바이트 연산 } void loop() { // put your main code here, to run repeatedly: while(1) { iTARGET = SEND_COMMAND(TARGET_CMD); // 대상 온도 Read delay(50); // 50ms : 이 라인을 지우지 마세요 iSENSOR = SEND_COMMAND(SENSOR_CMD); // 센서 온도 Read delay(500); // 500ms : 이 라인을 지우지 마세요. Serial.print("Target Temp : "); Serial.print(float(iTARGET)/100); Serial.print(" Ambient Temp : "); Serial.println(float(iSENSOR)/100); } } </pre>
---	---

- 온도계산 프로세서를 내장하고 있어 정확한 온도 값 출력한다.
- 디지털 통신으로 온도 값 출력한다.
- 센서 온도와 대상 온도를 동시에 측정한다.

2. 제안 시스템 및 핵심기술

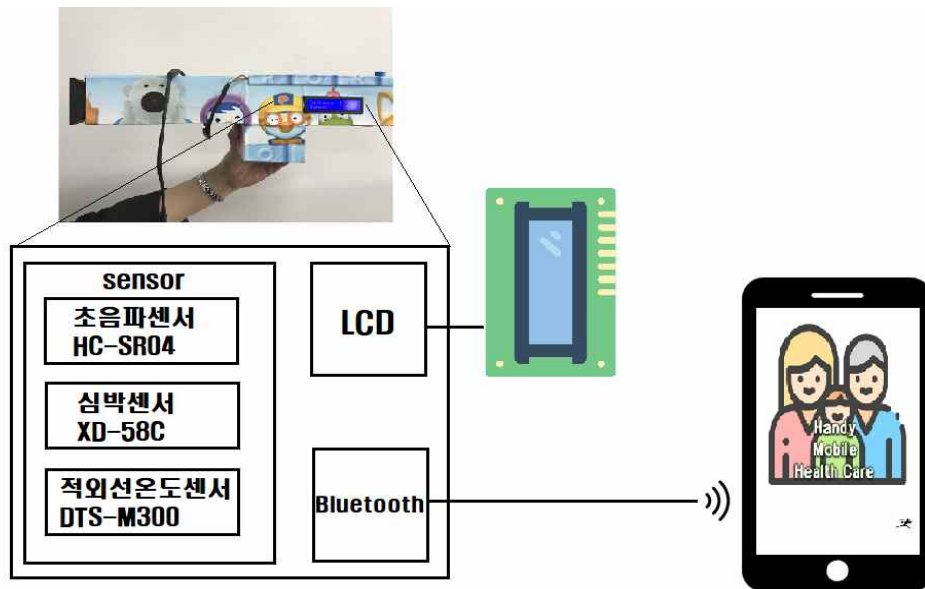
2.1 서비스 모델



< 그림 7. 서비스 모델 >

- 그림 7은 제품을 사용하여, 환자나 노약자, 어린이 등 대상으로 병원이 아니더라도, 간단히 집에서 중요 신체 상태를 기록하여 병원에 전달해, 원격으로 간단한 진단 및 진찰을 제공하기 위함 모델임
- 전국 병원 통합 관리 시스템 APP(Application)을 통해 내원이 아니더라도, 의사와 소통이 가능하며, 간단한 증상에 대한 예방과 처방 방법을 볼 수 있게 구현함
 - 기본적으로 체중, 신장, 체온, 혈압, 맥박을 측정한 기록 LCD모니터로 확인을 하고 블루투스 통신을 통해 측정한 기록을 APP(Application)으로 전송 후 DB(Data Base)에 저장하여, 병원의 프로그램으로 확인이 가능하고, 의사가 APP(Application)통해 확인을 하고, 그에 대한 처방을 할 수 있게 함
 - 체중, 신장, 체온, 혈압, 맥박 등은, 신장을 측정할 수 있게, 아두이노를 활용하여, 초음파센서, 비접촉 체온센서, 맥박센서, 무게센서, 혈압센서를 하나로 통합하여 간편하게 측정할 수 있음
 - APP(Application)의 소통공간을 통해 이용자 끼리 같은 병이나, 이미 완치된 병에 대한 지식을 공유할 수 있게 게시판을 사용함. 실시간으로 파악할 수 있음

2.2 시스템 모델

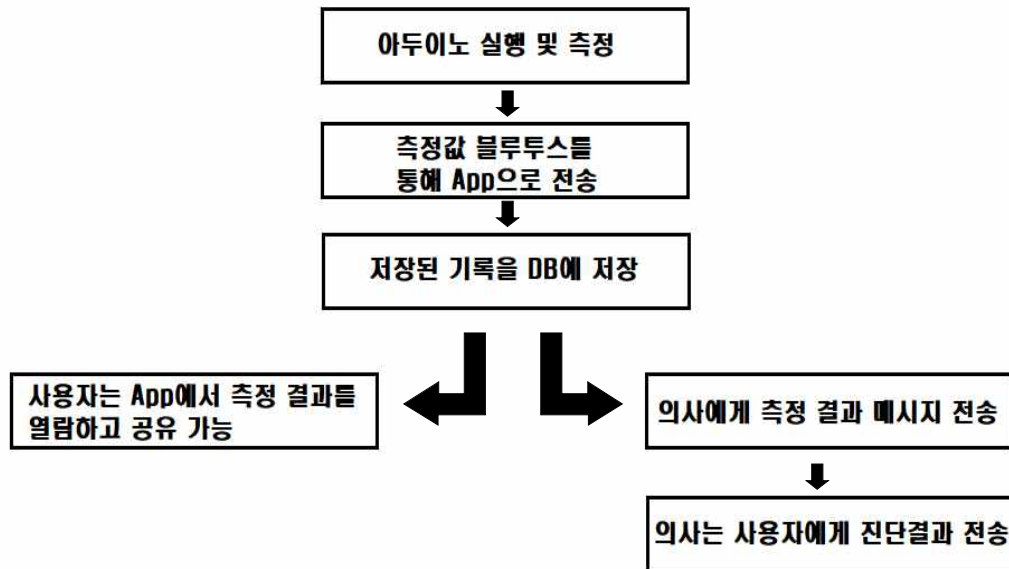


< 그림 8. 시스템 모델 >

□ 시스템 구성

- Arduino Mega에 각종 센서 와 Bluetooth3.0 센서를 장착해 각 센서에서 측정 한 값을 Bluetooth 전송기능을 통하여 App으로 보냄
- 초음파센서 (Ultrasonic Sensor)는 센서로 부터 초음파를 내보내 물체에 닿는 순간 다시 돌아오는 초음파를 재는 방식으로, 이번 작품에서는 이 점을 이용해 신장을 재는 용도로 사용하였음
- 적외선센서 (Infrared Sensor)는 접촉을 하지 않고 원하는 물체 표면에 온도를 500ms 이 내에 정확하게 측정할 수 있는데 이번 작품에서는 이 점을 이용해 체온을 재는 용도로 사용하였음
- 맥박센서 (Pulse Sensor)는 심장 박동에 따라 일어나는 동맥의 주기적인 움직임을 측정하는 센서로, 이번 작품에서는 이 점을 이용해 1분 동안의 측정값을 저장하여 맥박을 재는 용도로 사용하였음
- 블루투스센서 (Bluetooth Sensor)는 전파를 이용해서 짧은 거리의 데이터를 통신하는 방식으로 문자 정보 및 음성 정보를 비교적 낮은 속도로 디지털 정보를 무선통신을 통해 주고 받을 수 있는 센서로, 현재 5.0버전까지 개발이 되어있지만 이번 작품에선 3.0버전을 이용해 각 센서 에서 측정한 값을 App으로 전송하는데 사용하였음

2.3 핵심 솔루션



< 그림 9. 플로우 차트 >

□ 개발한 모바일 헬스 케어 시스템의 상세설명은 다음과 같음

- 이 시스템은 건전지를 삽입, On/Off 버튼을 활용해 전원을 관리할 수 있고, 아두이노가 실행됨과 동시에 측정을 시작하게 됨
- 측정을 완료 시 측정값을 블루투스를 통해 APP에 전송
- 사용자 APP(Application)에 저장된 기록을 DB(Data Base)에 저장
- 저장된 기록을 의사 APP(Application)에서 확인 후 간단한 기록을 남김
- 사용자 APP(Application)에서 남긴 기록을 확인

2.3 핵심 솔루션

```
#include <Wire.h>
#include <SPI.h>
#include <NewPing.h>
#include <SoftwareSerial.h>
#include <LiquidCrystal_I2C.h> //각 센서들의 헤더파일

LiquidCrystal_I2C lcd(0x3F, 16, 2); //LCD 위치

#define TARGET_CMD 0xA0
#define USE_ARDUINO_INTERRUPTS true

#include <PulseSensorPlayground.h> //맥박센서 헤더파일

const int chipSelectPin = 10;
const int PulseWire = 0;
const int LED13 = 13; //pin번호
unsigned char T_high_byte;
unsigned char T_low_byte;
int iTARGET;
int trigPin = 8;
int echoPin = 9; //pin번호
int Threshold = 550;

NewPing sensor(trigPin, echoPin);
SoftwareSerial mySerial(2,3);
PulseSensorPlayground pulseSensor;

void setup()
{
  lcd.begin();
  lcd.backlight();

  /* Initalize PORT */
  pinMode(MISO, INPUT);
  pinMode(chipSelectPin, OUTPUT);
  pinMode(MOSI, OUTPUT);
  pinMode(SCK, OUTPUT);
  pinMode(trigPin, OUTPUT); // 센서 Trig 핀
  pinMode(echoPin, INPUT); // 센서 Echo 핀

  Serial.begin(9600);
  mySerial.begin(9600);
  pulseSensor.begin();

  pulseSensor.analogInput(PulseWire);
  pulseSensor.blinkOnPulse(LED13);
  pulseSensor.setThreshold(Threshold);

  digitalWrite(chipSelectPin, HIGH);
  SPI.setDataMode(SPI_MODE3);
  SPI.setClockDivider(SPI_CLOCK_DIV16);
  SPI.setBitOrder(MSBFIRST);
  SPI.begin();
  delay(500); //결과값 올라오는 속도
}

int SEND_COMMAND(unsigned char cCMD)
```

```

{
    digitalWrite(chipSelectPin , LOW);
    delayMicroseconds(10); // delay(10us)
    SPI.transfer(cCMD); // Send 1st Byte
    delay(10); // delay(10ms)
    T_low_byte = SPI.transfer(0x22);
    delay(10); //delay(10ms)
    T_high_byte = SPI.transfer(0x22);
    digitalWrite(chipSelectPin , HIGH);

    return (T_high_byte<<8 | T_low_byte);
}

void loop()
{
    int myBPM = 0;
    myBPM = pulseSensor.getBeatsPerMinute();

    digitalWrite(trigPin, HIGH); // 센서에 Trig 신호 입력
    delayMicroseconds(500); // 10us 정도 유지
    digitalWrite(trigPin, LOW); // Trig 신호 off

    long duration = pulseIn(echoPin, HIGH); // Echo pin: HIGH->Low 간격을 측정
    long distance = (duration / 29 / 2)+12 ; // 거리(cm)로 변환

    char buf[20];
    snprintf(buf, sizeof(buf), "Distance %4d cm", distance); //신장 출력 부분
    lcd.setCursor(0, 0);
    lcd.println(buf);
    mySerial.print("신장 : ");
    mySerial.print(distance);
    mySerial.println(" cm");

    long tempers = SEND_COMMAND(TARGET_CMD); // 대상 온도 Read
    long temper = float(tempers/100);
    snprintf(buf, sizeof(buf), "Temper %4d C ", temper); //체온 출력 부분
    lcd.setCursor(0, 1);
    lcd.println(buf);
    mySerial.print("체온 : ");
    mySerial.print(temper);
    mySerial.println(" C");

    if (pulseSensor.sawStartOfBeat()) { //심박 체온 부분
        mySerial.print("심박 : ");
        mySerial.println(myBPM/2);
    }
    delay(500);
}

```

□ 개발한 핵심 소스

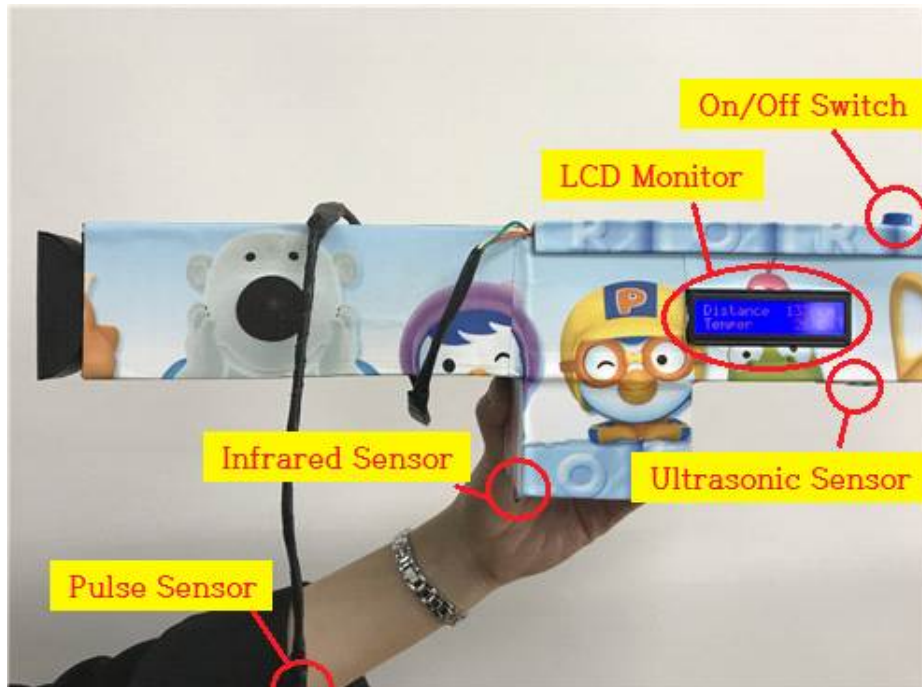
- 초음파센서, 적외선센서, 맥박센서 결합
 - 각각 센서를 구동하는 오픈소스들을 한 소스로 결합했다.
 - 소스들을 한 번의 실행으로 세 센서들을 구동시킬 수 있다.
- LCD 출력
 - 센서들로 측정한 값을 LCD보드에 출력할 수 있다.
- 블루투스 연결
 - 블루투스 모듈을 이용해 스마트폰과 연결해서 정보를 공유할 수 있다.

□ 활용 방안

- 다른 센서들도 결합가능하다.
 - 예를 들면, 혈압센서, 무게센서 등을 이용해 더 다양한 건강 측정을 할 수 있다.
- 와이파이 연결
 - 와이파이 연결로 개선하면 기계 한 개로 여러 명에서 측정할 수 있다.

3. 구현 및 실험 결과

3.1 구현결과

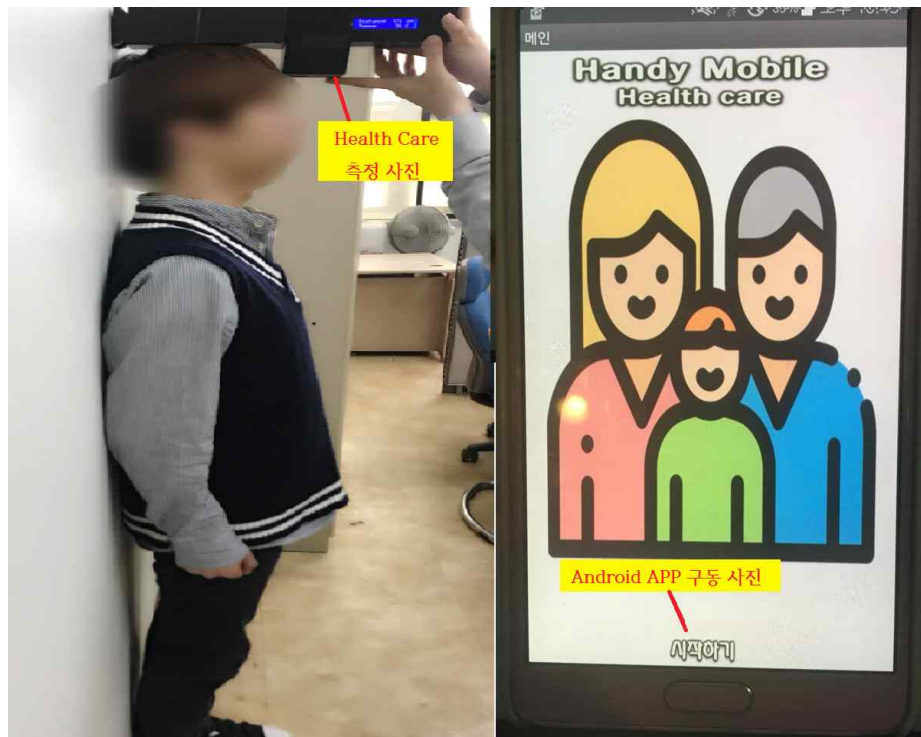


< 그림 10. 테스트 환경 >

□ 시스템 테스트 환경

- 그림 10과 같이 On/Off Switch를 통해 전원을 제어 할 수 있음
- 그림10은 제안한 서비스를 구현하기 위하여 Health Care 장비에 초음파센서(Ultrasonic Sensor), 적외선센서(Infrared Sensor), 맥박센서(Pulse Sensor)를 장착 함
- 그림 10과 같이 측정값을 확인 할 수 있게 LCD Monitor를 장착 함

3.2 구현결과



< 그림 11. 실험결과 >

□ 구현 결과

- 그림 11과 같이 Health Care 장비를 통해 키/체온을 측정
- 그림 11과 같이 측정한 값을 블루투스(Bluetooth) 통해 실시간으로 Android App으로 전송되는 것을 확인함
- 이후 측정값을 사용자 APP(Application)에 저장된 기록을 DB(Data Base)에 저장, 저장된 기록을 다른 Android 기기의 APP(Application)에서 확인함

4. 결론

- 인류가 살면서 의식주를 비롯한 건강 문제는 기술이 나날이 발전하면서 여러 개의 통합 형태의 발전 모습을 보이면서 다양한 분야에 활용되고 있음
 - 한 번의 검사로, 사용자의 여러 건강 기능에 대해 측정할 수 있음
- 사용자의 신체 건강을 측정 후 블루투스를 통해 APP(Application)으로 전송, DB(Data Base)에 저장된 사용자의 건강 정보를 의사가 확인할 수 있고, 확인한 정보를 토대로 간단한 조치를 취할 수 있음
- APP(Application)내의 의사들이 공동으로 적어서 수정 보완할 수 있는 간단한 질병에 대한 예방 방법과, 처방 방법을 기록해 사용자들이 의사가 적은 글을 확인할 수 있음
- 아두이노를 사용한 측정기기로 굳이 진단 목적이 아니더라도, 간단하게 건강 체크 및 신장 체크를 할 수 있고, APP(Application)를 활용해 기록으로 저장이 가능함

5. 참고문헌

- [1] 앱인벤터+아두이노 스마트폰 앱 프로젝트 (블루투스 와이파이 통신을 이용한) / 우지윤 저
- [2] 앱 인벤터 (쉽고 재미있게 만드는 앱 프로그래밍) / 이종원 저
- [3] 모두의 아두이노 (누구나 쉽게 배우는 전자 회로 공작과 프로그래밍) / 다카모토 다카요리 저
- [4] 아두이노, 상상을 현실로 만드는 프로젝트 입문편 / 최재규, 이준혁 저