심전도(ECG) 무선 모니터링 시스템

작성자 : 이동우

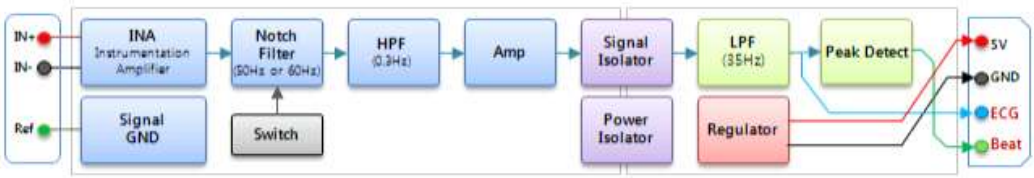
1. 프로젝트 개요

본 프로젝트는 심전도(ECG) 센서로부터 데이터를 수집하고, ADC를 통해 디지털 신호로 변환하여 ZigBee 기반의 무선 데이터 교환을 통해 LCD에 실시간으로 모니터링하는 시스템을 구축하는 것을 목표로 합니다. 이는 환자의 심장 건강 상태를 지속적으로 모니터링하면서 이동성을 제약하지 않는 방향성을 가진 의료 서비스를 제공하기 위한 시스템 제작에 중점을 두고 있다.

2. 구성요소 및 기능

Arduino Uno (R3) - 2개
XBee ZigBee TH(S2C) - XB24CZ7WIT-004 - 2개
아두이노 (Arduino) 호환 XBee 쉴드 XBee Shield 3.0 - 2개
TFT LCD 쉴드 arduino 2.4inch Color Touch TFT LCD Shield - 2개
PSL-iECG2(소형 2채널 심전도 및 심박 측정 모듈) - 2개

3. 시스템 작동 원리

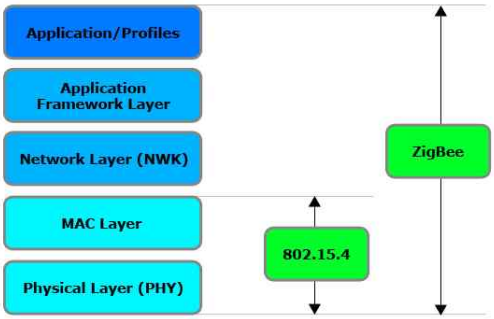


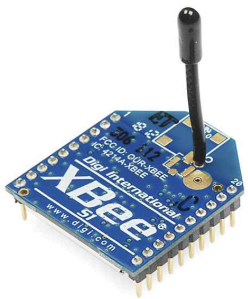
PSL-ECG2와 Arduino를 연결하여 ecg 전압 데이터(0 ~ 3.3V)를 획득, 2채널 심전도에서 ch.1은 ecg signal, ch.2는 r-peak detection 정보를 담고 있어, 이를 활용하여 Serial 통신에서 송신부에 해당하는 파트에는 ecg 센서 및 chipset이 직접 연결되어 있어야 한다.

```python
import serial
import struct

# COM 포트 설정
port_tx = 'COM6'
port_rx = 'COM7'

# 시리얼 연결 설정
ser_tx = serial.Serial(port_tx, 1000000, timeout=0)  # 송신용 시리얼 포트
ser_rx = serial.Serial(port_rx, 1000000, timeout=0)  # 수신용 시리얼 포트

try:
    while True:
        if ser_tx.in_waiting:
            # 시리얼 포트에서 데이터 한 줄을 읽고, cp1252로 디코딩한 뒤 공백 문자 제거
            data = ser_tx.readline().decode('cp1252').rstrip()
            # 콤마로 데이터를 나눔
            split_data = data.split(',')

            # 첫 번째 데이터 항목을 바이트로 인코딩하여 전송
            try:
                # 데이터가 숫자인 경우를 가정하고 float 변환을 시도
                float_value = float(split_data[0])
                print(float_value)
                # float 값을 바이트로 변환하여 전송
                ser_rx.write(struct.pack('<f', float_value))
            except ValueError:
                # 변환에 실패한 경우 무시하고 계속 진행
                pass

except KeyboardInterrupt:
    # 종료 전에 시리얼 포트 닫기
    ser_tx.close()
    ser_rx.close()
    print("프로그램 종료")
```

python으로 작성된 Serial 통신을 위한 중계 스크립트를 작성하였다. 이를 기반으로 별도의 시리얼 통신을 통해 ecg 데이터의 채널 정보를 문자열 자료형으로 변환한 데이터를 주고받는 것을 구현하였다.

유선 통신 코드를 발전시켜, Zigbee(블루투스) 모듈을 활용하여 통신하기 위해 추가적인 코드 작업을 수행하였다. Zigbee는 IEEE 802.15.4-2003을 기반으로 한 저전력의 디지털 라디오를 활용한 high-level 통신 프로토콜이다. 낮은 데이터률과 적은 배터리 소모, 네트워크의 안전성을 요구하는 RF 어플리케이션에 주로 사용된다는 특징이 있다. 즉, 휴대성을 목적으로 한 기기에 활용하는 것에 유용하다고 판단하여 채용하였다. Zigbee 통신 모듈 중에서 Digi 사의 Xbee 모듈이 대중적임을 알게 되어, 채용하기로 하였다.





<sender.ino>

```ino
#include <TimerOne.h>

int analogPin0=0;
int analogPin1=1;

void setup() {
  Serial.begin(9600);
  Timer1.initialize(2000); //uS단위로 설정
  Timer1.attachInterrupt(AnalogAD);
  Timer1.start();
}

void AnalogAD()
{
  int reading0=analogRead(analogPin0);
  int reading1=analogRead(analogPin1);

  float Voltage0=(float)reading0/1023*5;
  float Voltage1=(float)reading1/1023*5;

  if (Voltage1>1.65) digitalWrite(13,HIGH);
  else digitalWrite(13,LOW);

  String str=String(Voltage0,3) +","+ String(Voltage1,3);
  Serial.println(str);
}


void loop() {
}
```

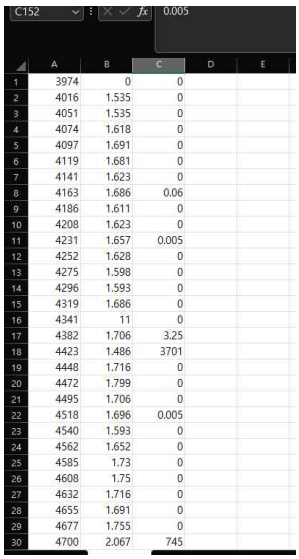
Zigbee 모듈 통신 속도에 맞춰 9600으로 설정하여 준다. (수신부도 동일) analog signal에서 sampling된 discret signal을 10bit (1024개)의 값에 mapping 시켜 quantization을 수행한다. 기본적으로 시리얼 통신에 주고 받는 데이터의 형식은 “{ch.1 데이터},{ch.2 데이터}”를 따른다. 또한, Xbee 기본 모듈의 전송속도는 9600이므로, Zigbee 모듈 통신 속도 9600에 맞춰 송신부와 수신부의 통신 속도를 설정한다.

<receiver.ino - parser 부분>

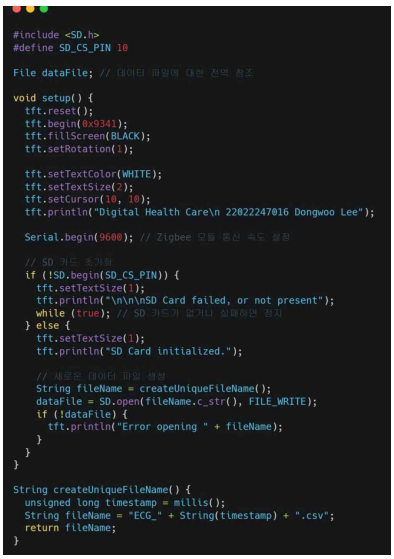
```
void loop() {
  if (Serial.available() > 0) {
    String receivedData = Serial.readStringUntil('\n');

    receivedData.trim();

    int commaIndex = receivedData.indexOf(',');
    if (commaIndex != -1) {
      String volt0String = receivedData.substring(0, commaIndex);
      String volt1String = receivedData.substring(commaIndex + 1);

      float VOLT0 = volt0String.toFloat();
      float VOLT1 = volt1String.toFloat();
      drawECG(VOLT0, VOLT1);
  }
}
}
```

그렇게 수신부에서 string type의 데이터를 float형으로 parsing 해야 한다. 그러기 위해서, python split function과 같은 기능을 하는 코드를 작성하여 준다. 이를 통해 결정된 두 채널의 데이터를 drawECG 함수의 인자로 받는다.

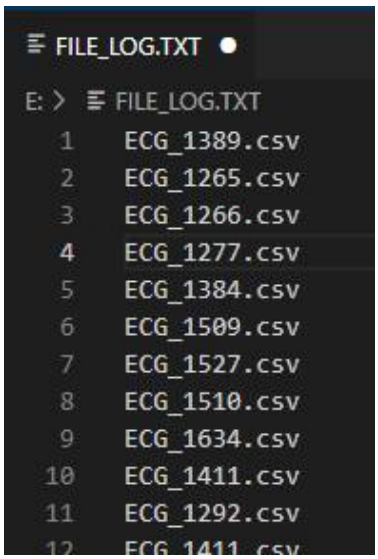
TFT LCD shleid 후면부에 있는 SD card socket을 활용하여, SD 카드가 있는 경우에만 ECG 데이터 측정을 시작하도록 하고, 저장은 CSV 형식으로 두 채널의 ECG 데이터를 소숫점 4자리까지 저장하는 코드를 구현했다.



	A	B	C	D	E
1	3974	0	0		
2	4016	1.535	0		
3	4051	1.535	0		
4	4074	1.618	0		
5	4097	1.691	0		
6	4119	1.681	0		
7	4141	1.623	0		
8	4163	1.686	0.06		
9	4186	1.611	0		
10	4208	1.623	0		
11	4231	1.657	0.005		
12	4252	1.628	0		
13	4275	1.598	0		
14	4296	1.593	0		
15	4319	1.686	0		
16	4341	11	0		
17	4382	1.706	3.25		
18	4423	1.486	3701		
19	4448	1.716	0		
20	4472	1.799	0		
21	4495	1.706	0		
22	4518	1.696	0.005		
23	4540	1.593	0		
24	4562	1.652	0		
25	4585	1.73	0		
26	4608	1.75	0		
27	4632	1.716	0		
28	4655	1.691	0		
29	4677	1.755	0		
30	4700	2.067	745		



```
#include <SD.h>
#define SD_CS_PIN 10

File dataFile; // 데이터 파일에 대한 전역 참조

void setup() {
  tft.reset();
  tft.begin(0x9341);
  tft.fillScreen(BLACK);
  tft.setRotation(1);

  tft.setTextColor(WHITE);
  tft.setTextSize(2);
  tft.setCursor(10, 10);
  tft.println("Digital Health Care\n 22022247016 Dongwoo Lee");

  Serial.begin(9600); // Zigbee 모듈 통신 속도 설정

  // SD 카드 초기화
  if (!SD.begin(SD_CS_PIN)) {
    tft.setTextSize(1);
    tft.println("\n\n\nSD Card failed, or not present");
    while (true); // SD 카드가 없거나 실패하면 정지
  } else {
    tft.setTextSize(1);
    tft.println("SD Card initialized.");

    // 새로운 데이터 파일 생성
    String fileName = createUniqueFileName();
    dataFile = SD.open(fileName.c_str(), FILE_WRITE);
    if (!dataFile) {
      tft.println("Error opening " + fileName);
    }
  }
}

String createUniqueFileName() {
  unsigned long timestamp = millis();
  String fileName = "ECG_" + String(timestamp) + ".csv";
  return fileName;
}
```



header 정보로는 timestamp, ch.1, ch.2 순으로 저장되며 위의 그림과 같이 저장된다. csv 파일로 저장된 2 channel ecg data는 연구, 혹은 정적인 visualization과 pan-tompkins algorithm과 같은 ecg 데이터의 전형적인 필터링 기법을 r-r interval 구간으로 split하여 분석하는 데 있어서 유용하게 사용할 수 있다. file_log.txt를 csv 데이터를 저장할 때 txt 파일에 row마다 저장하는식으로 하여 해당 데이터를 한꺼번에 읽어오는 식으로의 응용이 가능하다.

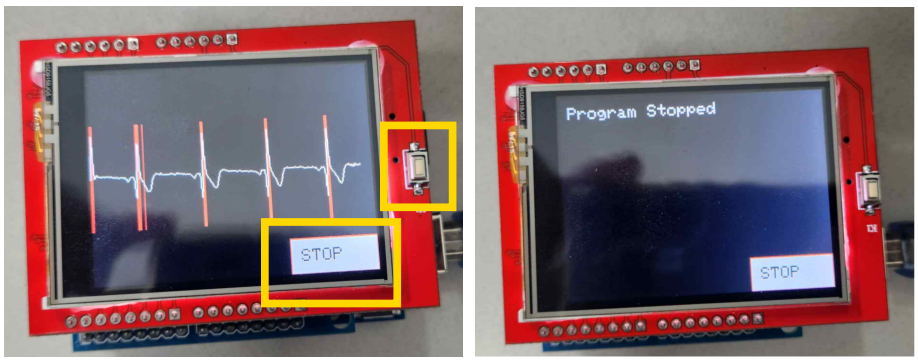
특히, python method를 활용하여 txt 파일에 있는 csv 데이터 file name을 list indent로 저장하여 ecg 데이터 load 함수를 제작하면 모든 SD 카드 내부에 있는 ecg 데이터를 불러와

서 특정 작업을 수행하기도 편리하다. 한 예시로, Pan_tompkins_algorithm을 구현한 코드를 응용하여 불러온 데이터를 전처리하여 시각화하는 것을 넘어 Deep Learning 모델에 필요한 데이터로ㅆㅓ의 활용 또한 가능할 것으로 보고 있다.
https://github.com/Pramod07Ch/Pan-Tompkins-algorithm-python/blob/main/Pan_tompkins_algorithm.py

4. 부가적인 기능 (UI)
1) SETUP Button (디지털 핀 레벨에서 구현): TFT LCD 스크린 상부에 버튼을 누르면 새로운 측정을 수행하며, 기존까지 측정되었던 데이터는 csv 파일 형태로 저장된다. txt파일 안에 csv file명이 담기며 (f“ecg_{timestamp}_.csv\n“)이 txt를 읽어와 모든 csv 파일 리스트를 확보할 수 있다.

2) STOP Button (터치스크린 레벨에서 구현) : LCD 터치스크린 패널 위의 우측 하단부의 STOP 버튼을 누를 시, 일시정지(Pause)되며, 다시 누를 경우 재생(RESUME)되고, 본래의 용도는 STOP 버튼을 통해 시리얼 통신을 멈춘 상태에서 1)에서의 SETUP 버튼을 눌려 데이터를 저장하는 데서 안정성을 확보하기 위한 장치로써 제작했다.

3) ecg signal visuatlization : 우선 0 ~ 3.3V의 전압 신호를 TFT 스크린 좌우 너비에 맞게 Normalization을 수행하여, 시각화하였다. 1채널인 ecg data(흰색) 2채널인 r-peak(빨간색 수직선)을 통시에 시각화하여 segment 정보를 추출가능하게 시각화하여 보았다.



5. 프로젝트 고찰 및 지속 가능성
이 시스템은 환자의 심장 건강 상태를 모니터링하는 데 사용될 수 있다. 또한 무선 통신을 이용함으로써 환자의 이동성의 제약을 줄이고, 연속적인 건강 모니터링을 제공할 수 있다. 또한, 이것은 스마트 병원에서의 비콘(beacon)을 활용하여 환자의 위치를 추적하는 프로토콜과 결합할 수 있을 것이라고 생각한다. 그러나, 의료 정보 교환 표준(HL7)을 사용하지 않고 무선 통신을 구현했기 때문에, 보안상의 문제가 있고 이를 의료 기기라 정의할 수는 없는 문제 등으로 인해, 개선의 여지가 있다. 아래 내용을 참고하여, ECG Device Mertric으로써 ECG 데이터 교환 예제를 참고하면, 가능은 할 것으로 보고 있다.

Example DeviceMetric: ECG Sample Array metric example
https://build.fhir.org/ig/HL7/uv-pocd/DeviceMetric-ECGSampleArrayDeviceMetric.html