Curtin University

Computer Technology Project 2
Final Report

"Health management Via Bicycle Sensor"

Name: HyoJin Cha
ID: 18611458

Synopsis

This report describes the progress of project entitled “Health management via Bicycle sensor system”. The project is consist of hardware and software. The hardware part is sensor circuit design and embedded that main duties are reading the RPM of the wheel, gaining the heart beat rate and communicating to device via Bluetooth module. The software part is coding the Arduino using IDE (Integrated Development Environment) and android programming to be coded in Android Studio that using JAVA and XML. The database system also has been implemented which using SQLite system that provided by Android Studio. The database records all of the user’s activity information and shared via internet also someone has been travel the searching route then database information is going to be used to show the previous travel route, checking the calories burn and time to be destination.

Table of Contents

Background & Introduction

Nowadays, the Internet of Things (IoT) is growing fast and deeply entrenched in wide variety of fields because it can be used anywhere if there has internet connection and any size of device can apply the IoT concept. Therefore, the IoT concept has permeated our life and people look for IoT system for having better lifestyle. Through the IoT system, people can share their experience and data and other people to be able to manage their life in a better direction. Health and wellness is one of the big issue in society, more than 2.1 billion overweight people worldwide. For those people exercise is necessary activity and riding bicycle is one of effective exercise, in the past, people just riding without any database, however, if IoT concept is applied in this field the ton of data can be shared via internet and can manage the weight systematically. Therefore, in this project will cover the health management with bicycle through the sensor module.

The project aims to make small bicycle sensor system that measure values such as heart beat rate and RPM of the wheel then transmit to the device and the device will show the analyzed data on the screen. In this project uses the Arduino, some of sensor modules and Android studio software for testing and coding the system.

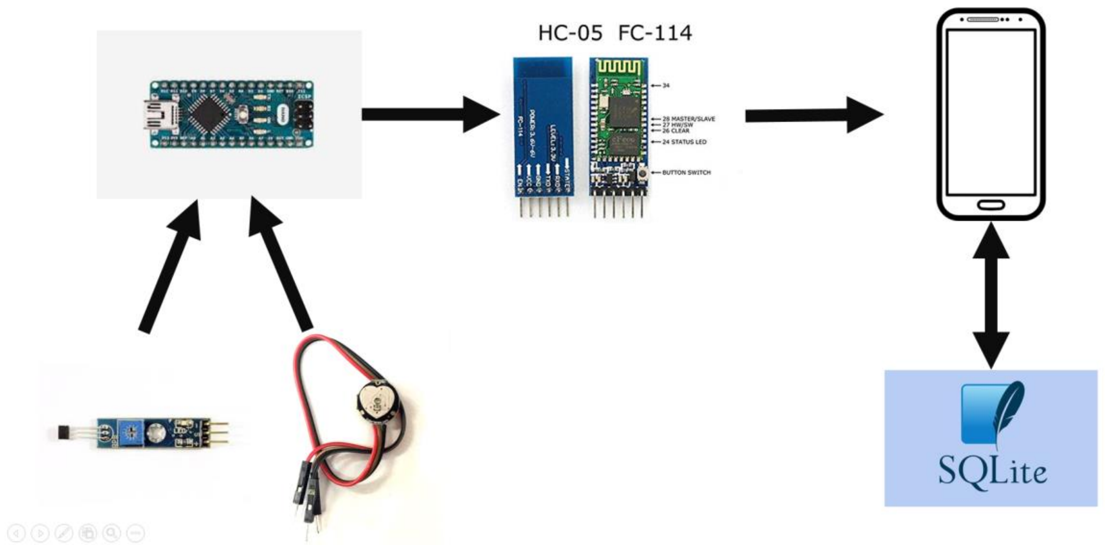
Over the project, there has some issues, first, the circuit embedded design has been changed for wire issue because it used to plan to measure the pedal force to gain more information for energy usage, however, the pedal part required wireless condition that not twisting the wire between pedal and mainboard also pressure sensor is not accurate measurement value because of too much noise value in there, so this part is modified in the future when it can be solved through research. The other issue is that mobile application part is unfamiliar field because B.Tech curriculum not involve that so this part has not done perfectly, such as Bluetooth and map API part is required higher knowledge of mobile application development, these problem are explained in later part in this report.

Requirements

The Requirements for this project are shown below

- Arduino Nano
- Sensor modules
- Android Studio

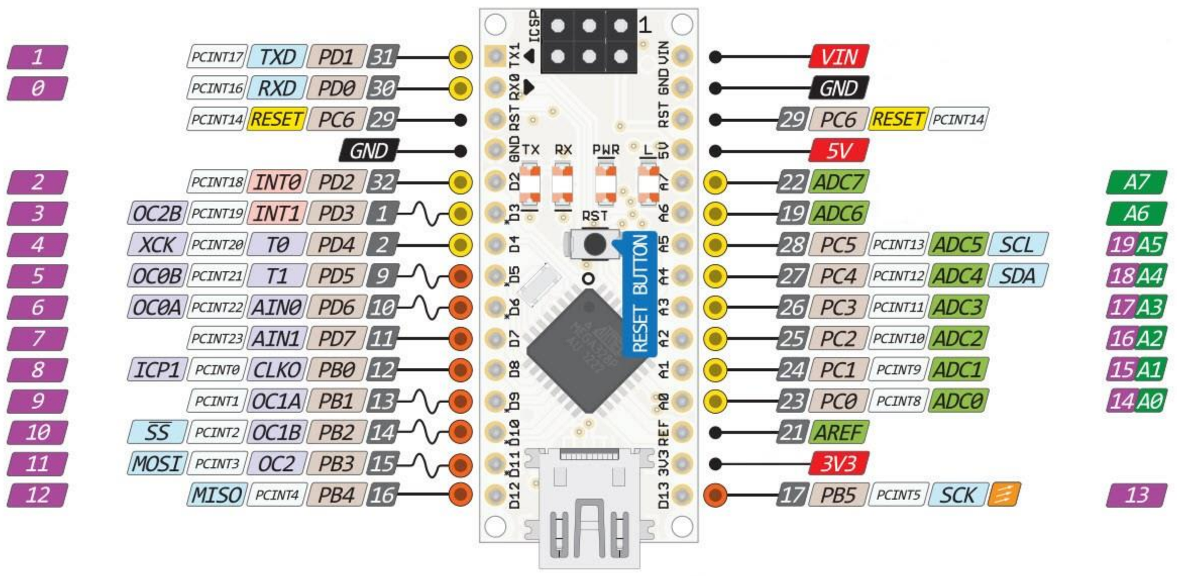
The overview of this project diagram is show below.



The overview diagram is only shown what has done this stage. It can be extended at sensor module part and Arduino Nano can be replaced to another level of Arduino such as Arduino Due because Arduino Nano only provide 1 serial (TX/RX) port so that cannot communicate with multiple sensors, for example, pedal pressure sensor required wireless communication between main microcontroller and pedal part microcontroller that used to plan to be embedded in this project. In the nutshell, there is possibility for sensor module part and Arduino type can be replaced and extended.

Arduino Nano

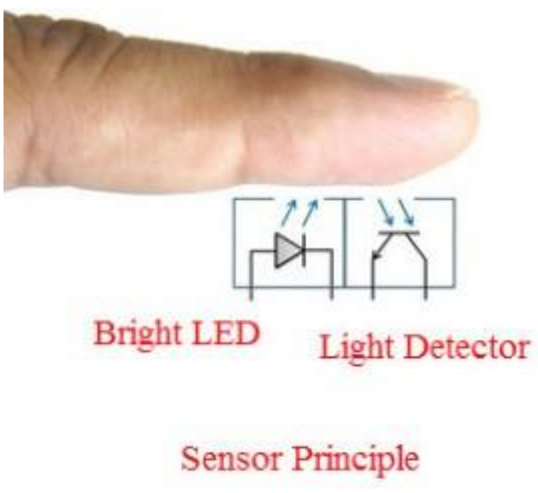
The Arduino Nano is a one of most popular microcontroller board that working on 5V and using ATmega328P Memory that has 2KB SRAM. The power is able to be supplied via computer USB connection or external power generator (regulate 5V or unregulated 6-20V). The programming of Arduino Nano can be processed on IDE software that using the C (coding language) and can be monitored on serial monitor function in Arduino IDE software. The detail pin information is displayed below



Sensor Modules

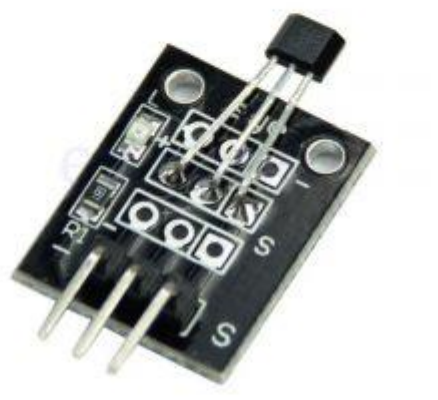
Pulse Rate Sensor (#Hbm0395)

The pulse Rate sensor is using to measure the heart beat rate while riding bicycle. The working condition is on 5V and 40mA. The procedure of this sensor is the LED on the sensor board pass the brightness to the light detector on the board. So when user touching the LED and detector together, the user’s finger is slightly expanded because of blood circulation so the light amount is reduced to detector and the sensor counts this. The Arduino code is provided by sensor module manufacturer.



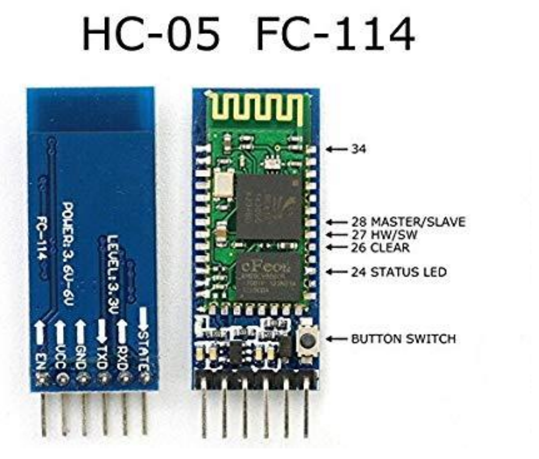
Hall Effect Sensor (KY-003)

The Hall Effect sensor is using to measure the RPM (revolution per minute) of the wheel while riding the bicycle. The working condition is on 5V. The procedure of Hall Effect sensor is the magnetic of the wheel pass the Hall Effect Sensor then the sensor count this number.



Bluetooth Module (HC-05)

The Bluetooth module is using to communicate between micro-controller and device. The working condition is on 4~6V, however, this module specification is not clear on the web because all of provider mention different value so above value is come from purchased website. The procedure of Bluetooth module is simple that decide to master (transmit) or slave (receive) and connect to the target.



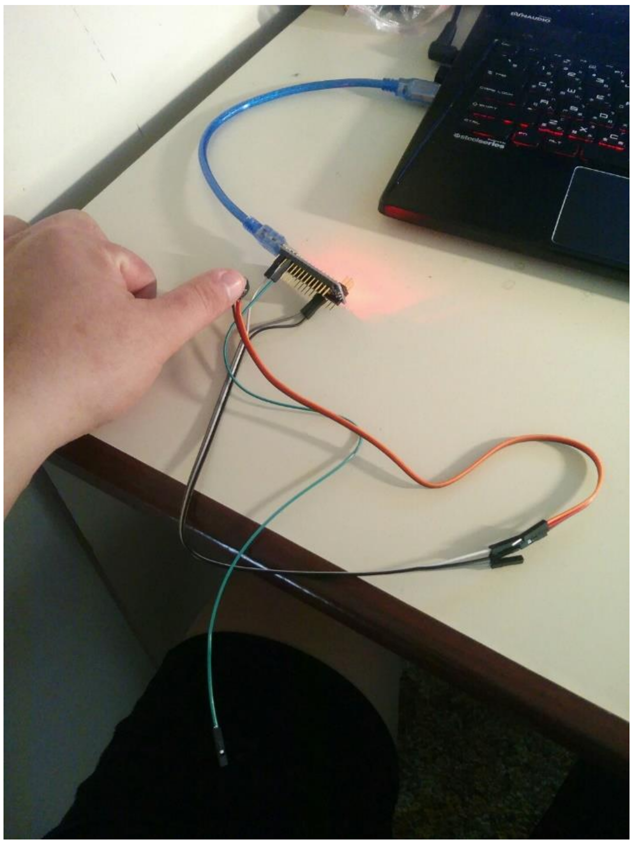
Android Studio

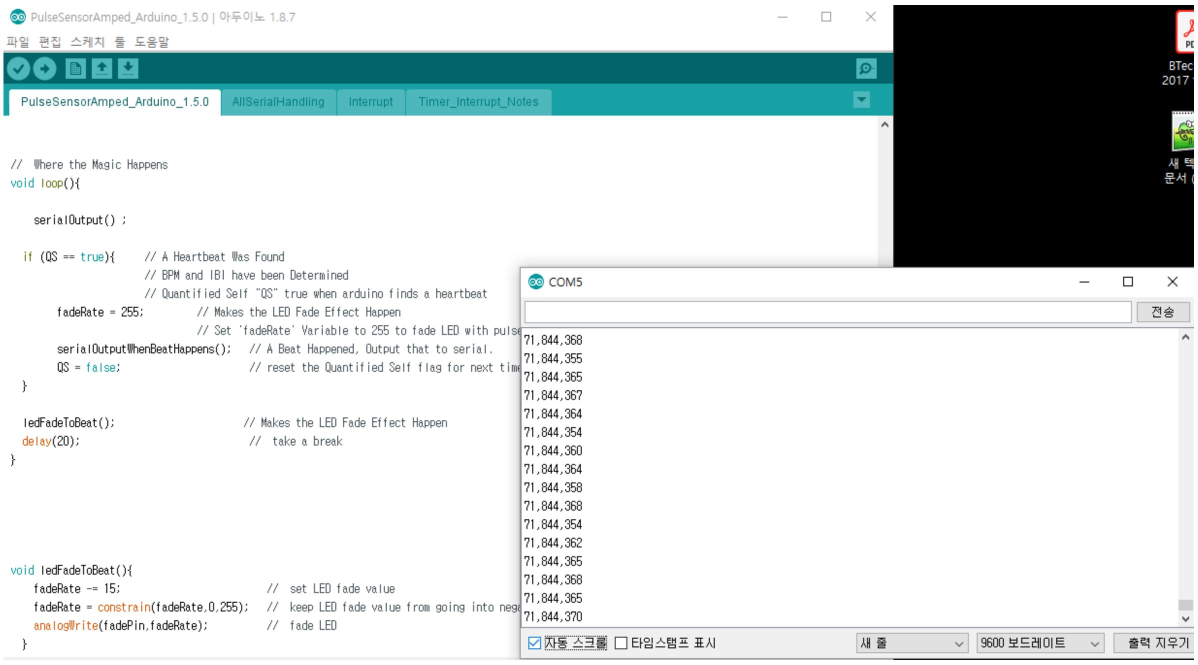
The Android studio is the official IDE for Android operating system provided by Google. The Android studio supporting few languages JAVA, KOTLIN and C++ but mainly using JAVA and XML for layout format. The feature of the studio is Gradle-based build support which is one of powerful build automation and supporting Android emulator that provide to user testing the application. The system requirements is a bit high, minimum 3GB RAM + 1GB RAM for Android Emulator, 6GB disk space for running the studio and JDK8 or above version, however, if running the studio smoothly then required at least 8GB RAM + 1GB RAM.

Progress

The project can be process step by step, however, it is mainly separated two main categories, the hardware embedded with coding and application programming. As shown above overview diagram, sensor modules are the lowest level of hardware part so the project begin from the sensor module. To process the sensor modules embedded, Arduino IDE must be installed in PC and required specification sheet for manage the voltage control. First, all of sensor modules are tested by single connection because of IDE coding test, after the test, all of IDE code is combined in main IDE code file and implemented on bread-board.

To start with pulse rate sensor which can measure the heart beat rate using the brightness and light detector. It generate noise as well so sometimes the value is fluctuated. The floater and serial monitor shown 3 values that is BPM, IBI and signal. The calculation is provided from pulse sensor manufacturer that can be checked on web-site (https://pulsesensor.com/pages/getting-advanced).

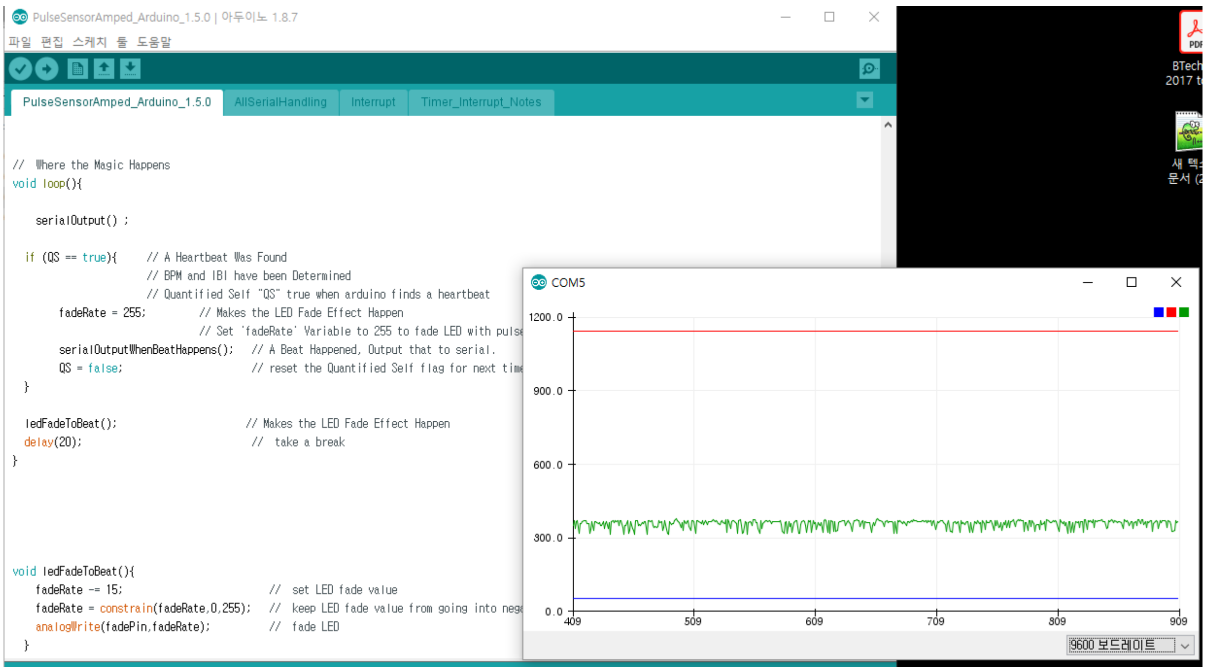




```
//  Where the Magic Happens
void loop(){

    serialOutput() ;

  if (QS == true){     // A Heartbeat Was Found
                       // BPM and IBI have been Determined
                       // Quantified Self "QS" true when arduino finds a heartbeat
        fadeRate = 255;         // Makes the LED Fade Effect Happen
                                // Set 'fadeRate' Variable to 255 to fade LED with pulse
        serialOutputWhenBeatHappens();   // A Beat Happened, Output that to serial.
        QS = false;                      // reset the Quantified Self flag for next time
  }

  ledFadeToBeat();                      // Makes the LED Fade Effect Happen
  delay(20);                             //  take a break
}



void ledFadeToBeat(){
    fadeRate -= 15;                         //  set LED fade value
    fadeRate = constrain(fadeRate,0,255);   //  keep LED fade value from going into nega
    analogWrite(fadePin,fadeRate);          //  fade LED
  }
```

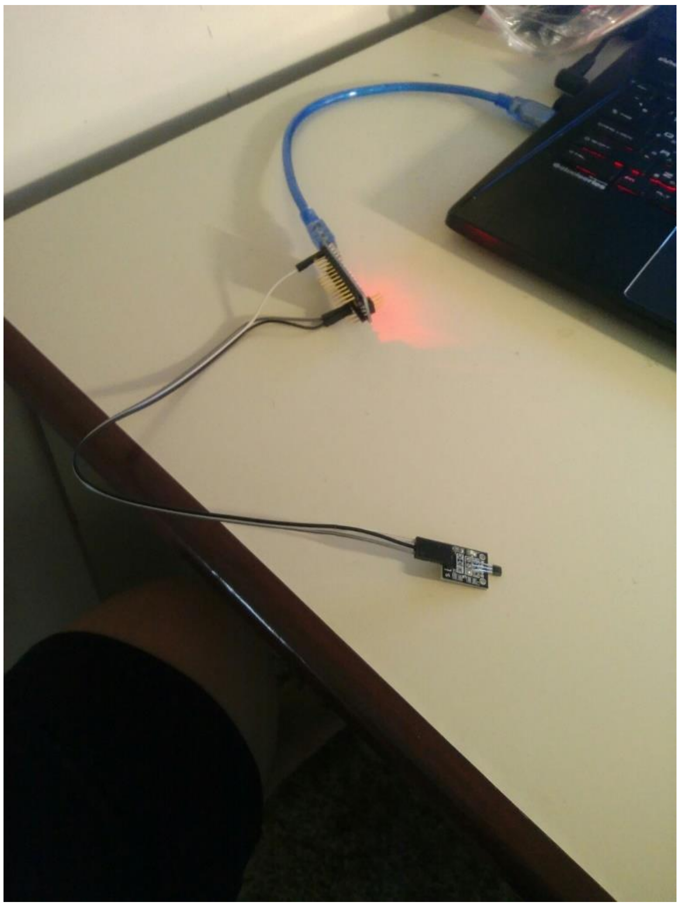
```
71,844,368
71,844,355
71,844,365
71,844,367
71,844,364
71,844,354
71,844,360
71,844,364
71,844,358
71,844,368
71,844,354
71,844,362
71,844,365
71,844,368
71,844,365
71,844,370
```

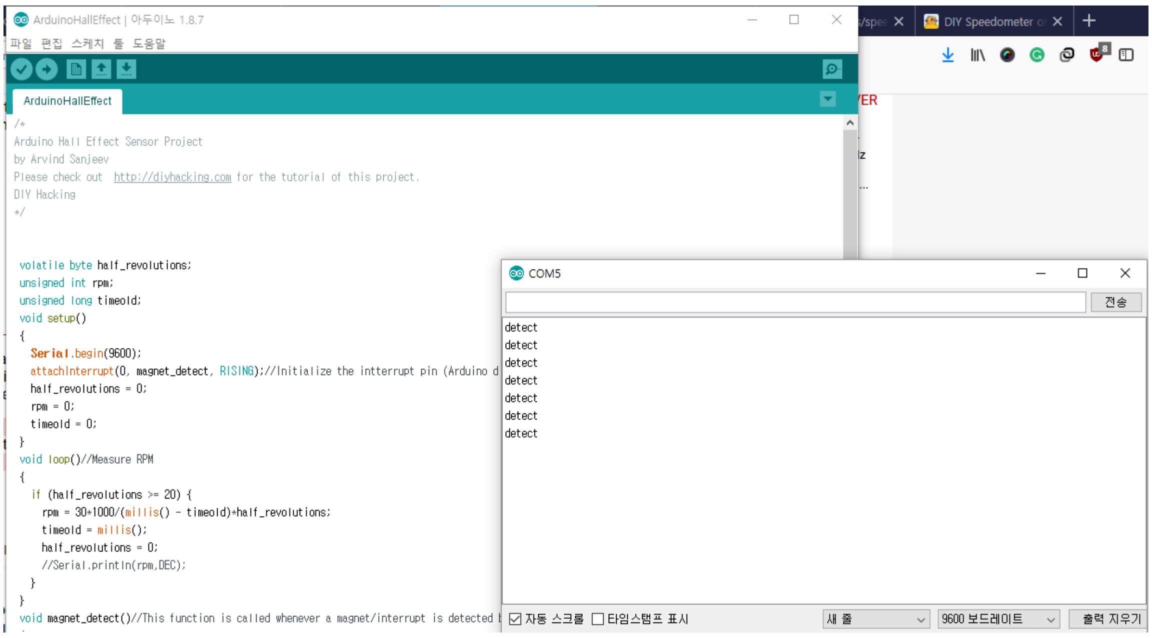
Settings: [✓ 자동 스크롤] 타임스탬프 표시 | 새 줄 | 9600 보드레이트



```
//  Where the Magic Happens
void loop(){

    serialOutput() ;

  if (QS == true){     // A Heartbeat Was Found
                       // BPM and IBI have been Determined
                       // Quantified Self "QS" true when arduino finds a heartbeat
        fadeRate = 255;         // Makes the LED Fade Effect Happen
                                // Set 'fadeRate' Variable to 255 to fade LED with pulse
        serialOutputWhenBeatHappens();   // A Beat Happened, Output that to serial.
        QS = false;                      // reset the Quantified Self flag for next time
  }

  ledFadeToBeat();                      // Makes the LED Fade Effect Happen
  delay(20);                             //  take a break
}



void ledFadeToBeat(){
    fadeRate -= 15;                         //  set LED fade value
    fadeRate = constrain(fadeRate,0,255);   //  keep LED fade value from going into nega
    analogWrite(fadePin,fadeRate);          //  fade LED
  }
```

The next step is Hall Effect sensor that can measure the speed of the wheel. This sensor is simply tested that magnetic can generate the signal then display “detected” on serial monitor so testing code is different to speed measure code that required little bit modified from test code.





ArduinoHallEffect | 아두이노 1.8.7

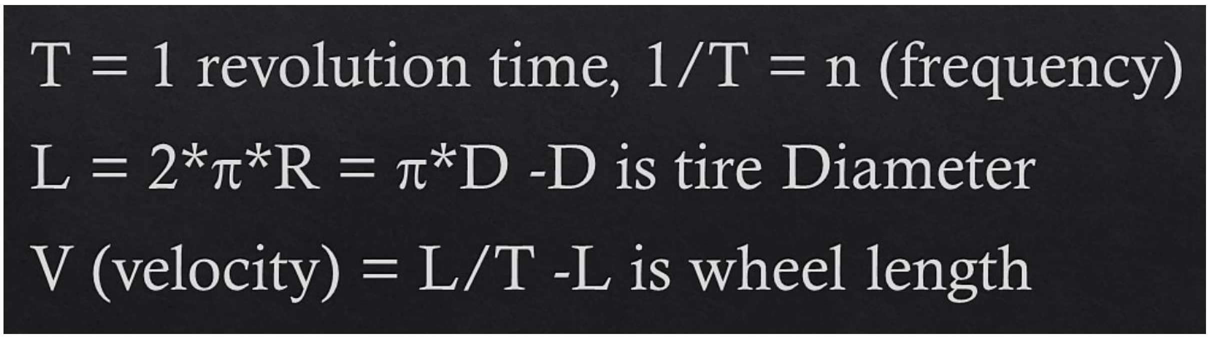
```
/*
Arduino Hall Effect Sensor Project
by Arvind Sanjeev
Please check out  http://diyhacking.com for the tutorial of this project.
DIY Hacking
*/


 volatile byte half_revolutions;
 unsigned int rpm;
 unsigned long timeold;
 void setup()
 {
   Serial.begin(9600);
   attachInterrupt(0, magnet_detect, RISING);//Initialize the intterrupt pin (Arduino d
   half_revolutions = 0;
   rpm = 0;
   timeold = 0;
 }
 void loop()//Measure RPM
 {
   if (half_revolutions >= 20) {
     rpm = 30*1000/(millis() - timeold)*half_revolutions;
     timeold = millis();
     half_revolutions = 0;
     //Serial.println(rpm,DEC);
   }
 }
 void magnet_detect()//This function is called whenever a magnet/interrupt is detected b
```

COM5

```
detect
detect
detect
detect
detect
detect
detect
```

[✓ 자동 스크롤] 타임스탬프 표시 새 줄 9600 보드레이트

The final code is required display speed on the screen instead "detected" that display formula is same as above.

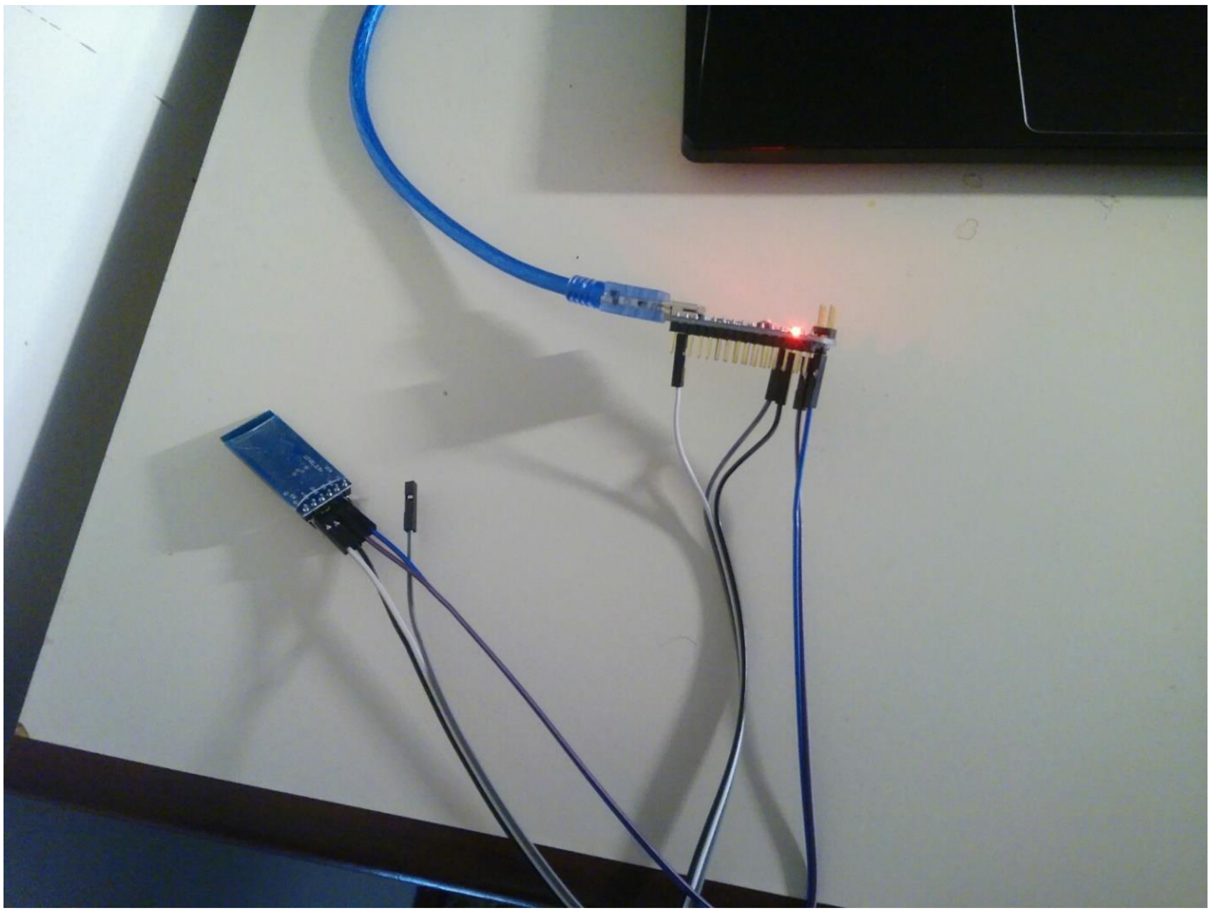


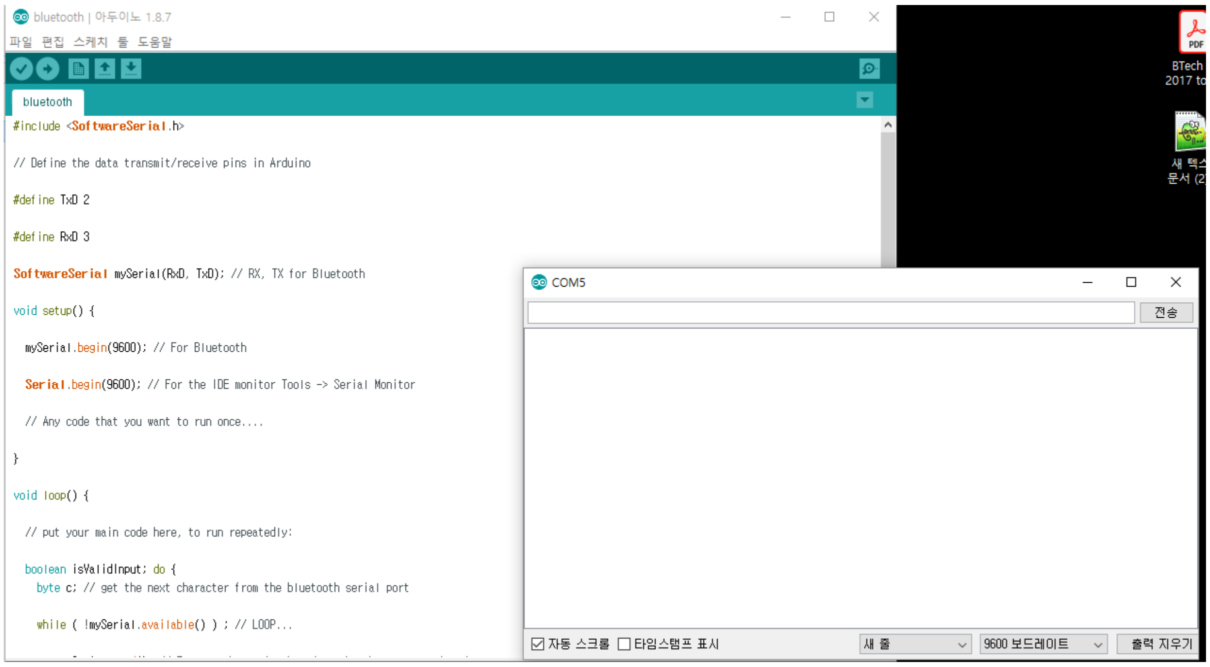
T = 1 revolution time, 1/T = n (frequency)

L = 2*π*R = π*D -D is tire Diameter

V (velocity) = L/T -L is wheel length

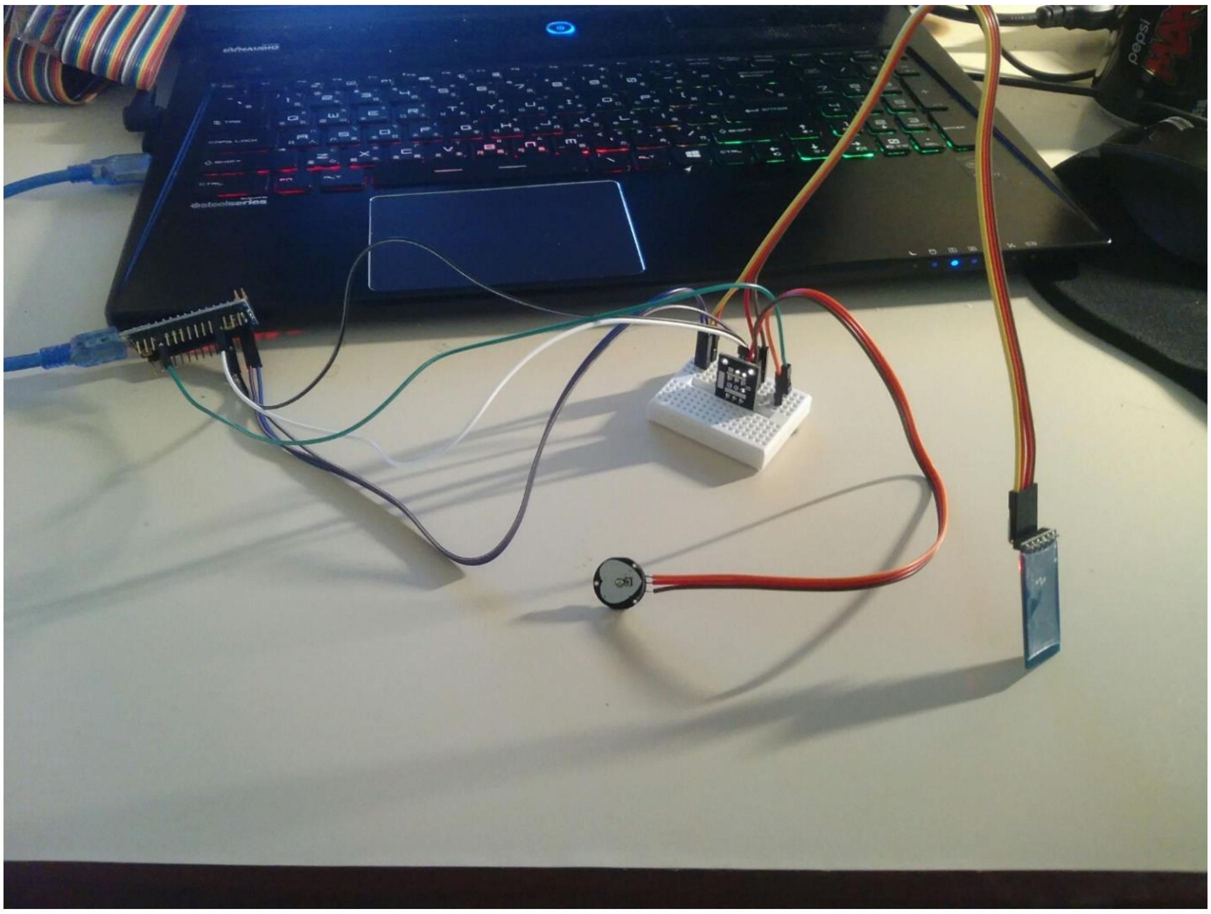
The last part is Bluetooth module that makes device and sensor modules can communicate and show the information on the mobile screen. This part may have problem because device can connect to the Bluetooth module, however, no respond from device also serial monitor.



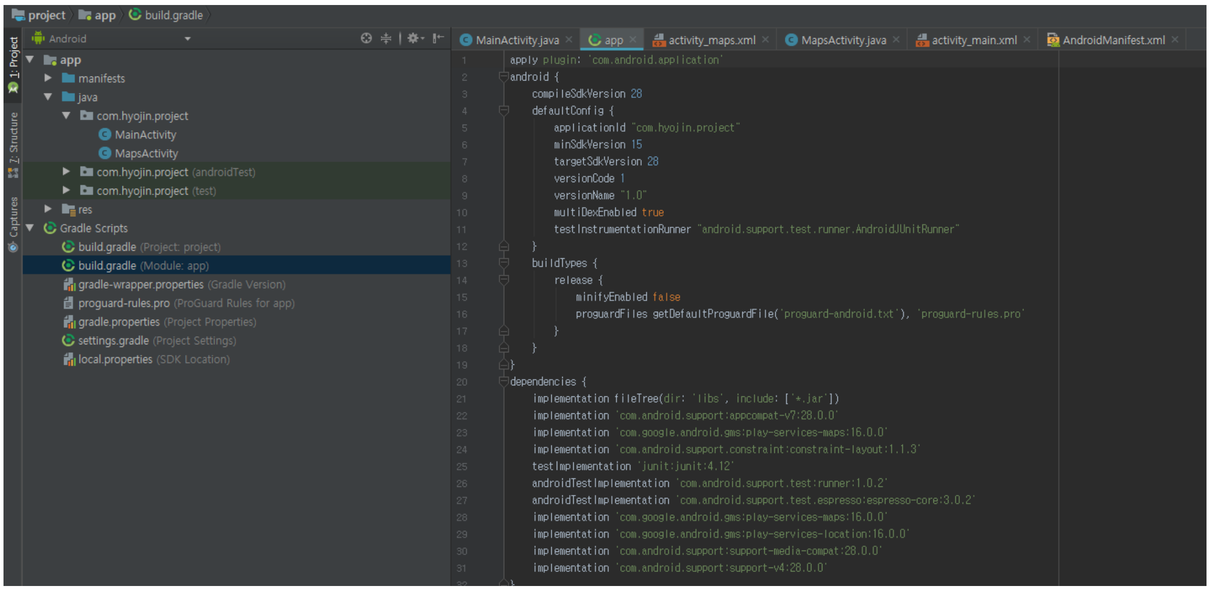

bluetooth | 아두이노 1.8.7
파일 편집 스케치 툴 도움말
bluetooth
COM5
전송
☑ 자동 스크롤 ☐ 타임스탬프 표시
새 줄
9600 보드레이트
출력 지우기
BTech 2017 to
새 텍스 문서 (2

```
#include <SoftwareSerial.h>

// Define the data transmit/receive pins in Arduino

#define TxD 2

#define RxD 3

SoftwareSerial mySerial(RxD, TxD); // RX, TX for Bluetooth

void setup() {

  mySerial.begin(9600); // For Bluetooth

  Serial.begin(9600); // For the IDE monitor Tools -> Serial Monitor

  // Any code that you want to run once....

}

void loop() {

  // put your main code here, to run repeatedly:

  boolean isValidInput; do {
    byte c; // get the next character from the bluetooth serial port

    while ( !mySerial.available() ) ; // LOOP...
```

The below photo is to combine the all of module to be embedded on one board.



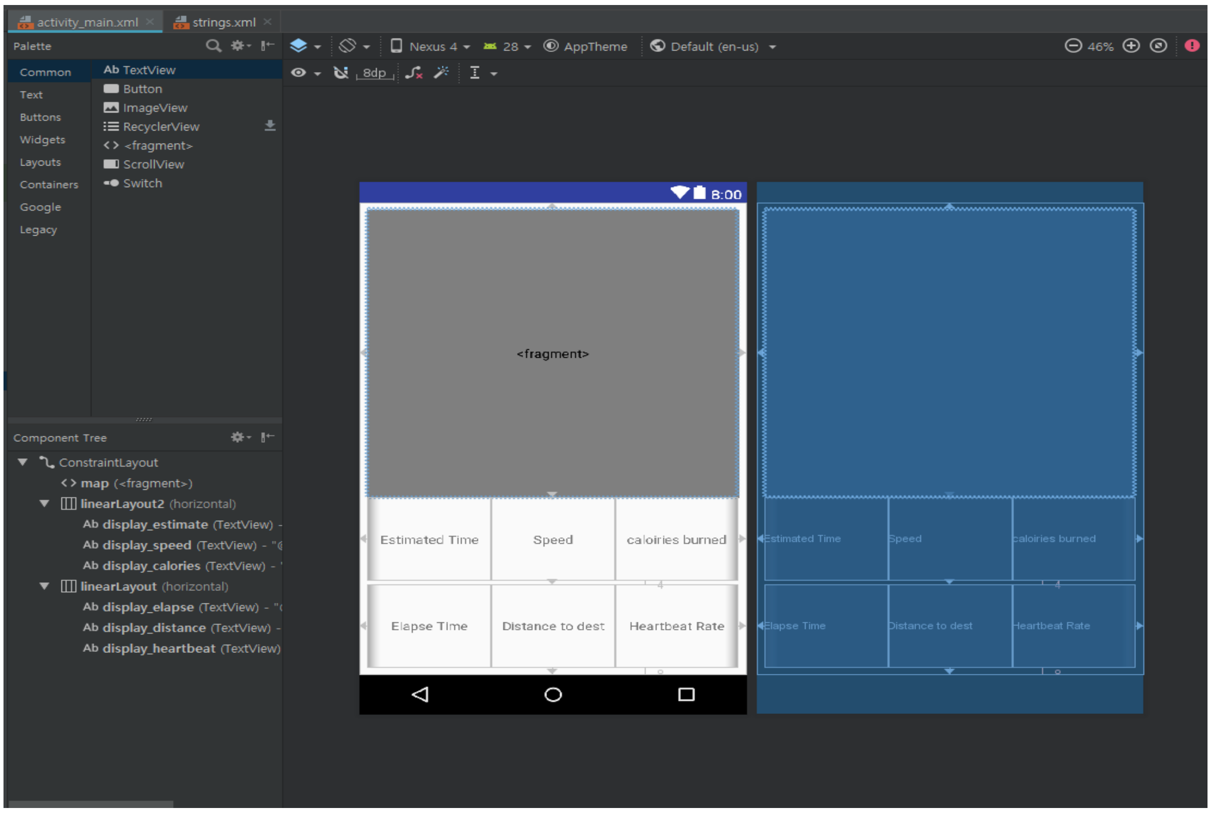
Before starting the coding, android studio is require to add the dependency and gradle build source modification work because google map and Bluetooth is not general option for build the application so additional dependency must be added.



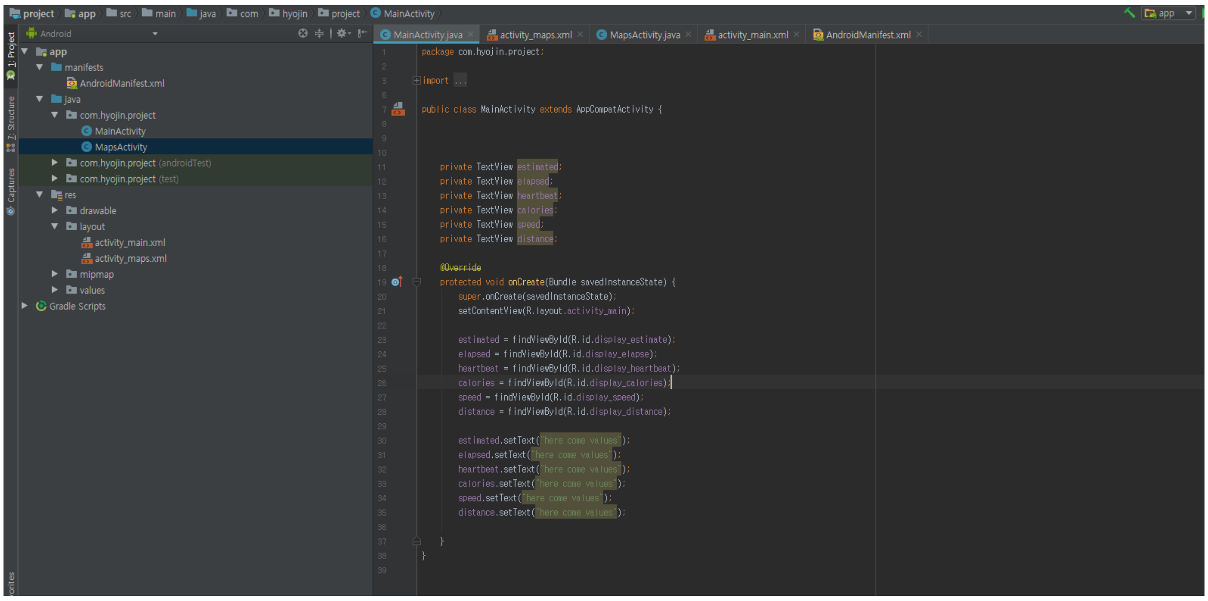
```
apply plugin: 'com.android.application'
android {
    compileSdkVersion 28
    defaultConfig {
        applicationId "com.hyojin.project"
        minSdkVersion 15
        targetSdkVersion 28
        versionCode 1
        versionName "1.0"
        multiDexEnabled true
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:appcompat-v7:28.0.0'
    implementation 'com.google.android.gms:play-services-maps:16.0.0'
    implementation 'com.android.support.constraint:constraint-layout:1.1.3'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:runner:1.0.2'
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
    implementation 'com.google.android.gms:play-services-maps:16.0.0'
    implementation 'com.google.android.gms:play-services-location:16.0.0'
    implementation 'com.android.support:support-media-compat:28.0.0'
    implementation 'com.android.support:support-v4:28.0.0'
}
```

project
app
build.gradle
Android
app
manifests
java
com.hyojin.project
MainActivity
MapsActivity
com.hyojin.project (androidTest)
com.hyojin.project (test)
res
Gradle Scripts
build.gradle (Project: project)
build.gradle (Module: app)
gradle-wrapper.properties (Gradle Version)
proguard-rules.pro (ProGuard Rules for app)
gradle.properties (Project Properties)
settings.gradle (Project Settings)
local.properties (SDK Location)
MainActivity.java
app
activity_maps.xml
MapsActivity.java
activity_main.xml
AndroidManifest.xml

The application programming is next step that using the android studio. First step is to make the layout for the application, in this project using the blank activity for flexible design because Android studio provide the Google map activity but that not included layout format so the layout for this project required to be made.

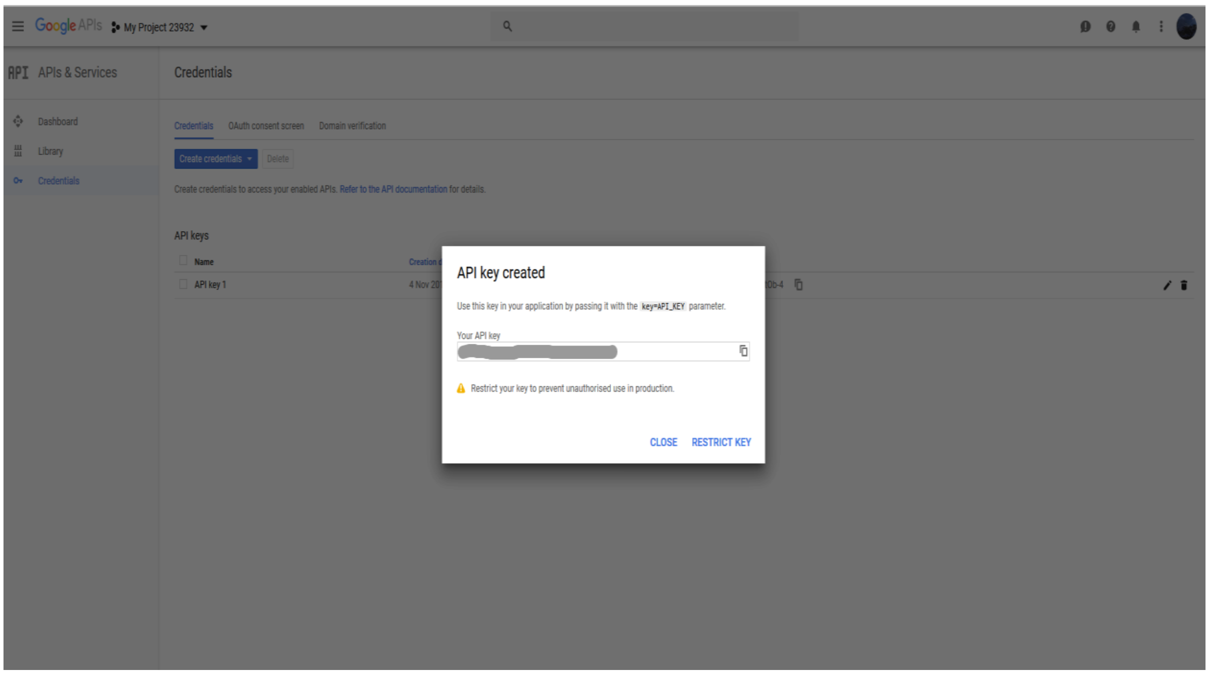


After made the layout, the all of Textview (display box) required to bind proper value but there has some issue so in this stage put String value that ‘here comes value’ when the issue is resolved then it will be replaced to proper variable, if the variable is not string then the variable can be converted using the method Valueof().



```java
package com.hyojin.project;

import ...

public class MainActivity extends AppCompatActivity {



    private TextView estimated;
    private TextView elapsed;
    private TextView heartbeat;
    private TextView calories;
    private TextView speed;
    private TextView distance;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        estimated = findViewById(R.id.display_estimate);
        elapsed = findViewById(R.id.display_elapse);
        heartbeat = findViewById(R.id.display_heartbeat);
        calories = findViewById(R.id.display_calories);
        speed = findViewById(R.id.display_speed);
        distance = findViewById(R.id.display_distance);

        estimated.setText("here come values");
        elapsed.setText("here come values");
        heartbeat.setText("here come values");
        calories.setText("here come values");
        speed.setText("here come values");
        distance.setText("here come values");

    }
}
```

The Google map is implemented as a fragment so main activity call the fragment manager after call the google map on the main activity. Using the google map API, to agree the google map API agreement on google API website (https://console.cloud.google.com) then will get the private key that is needed to enter in the XML file in Android studio. However, the map fragment is working partially, it only load the map information not specific marking point and additional function so that could not process from this stage.



Create credentials to access your enabled APIs. Refer to the API documentation for details.

API key created

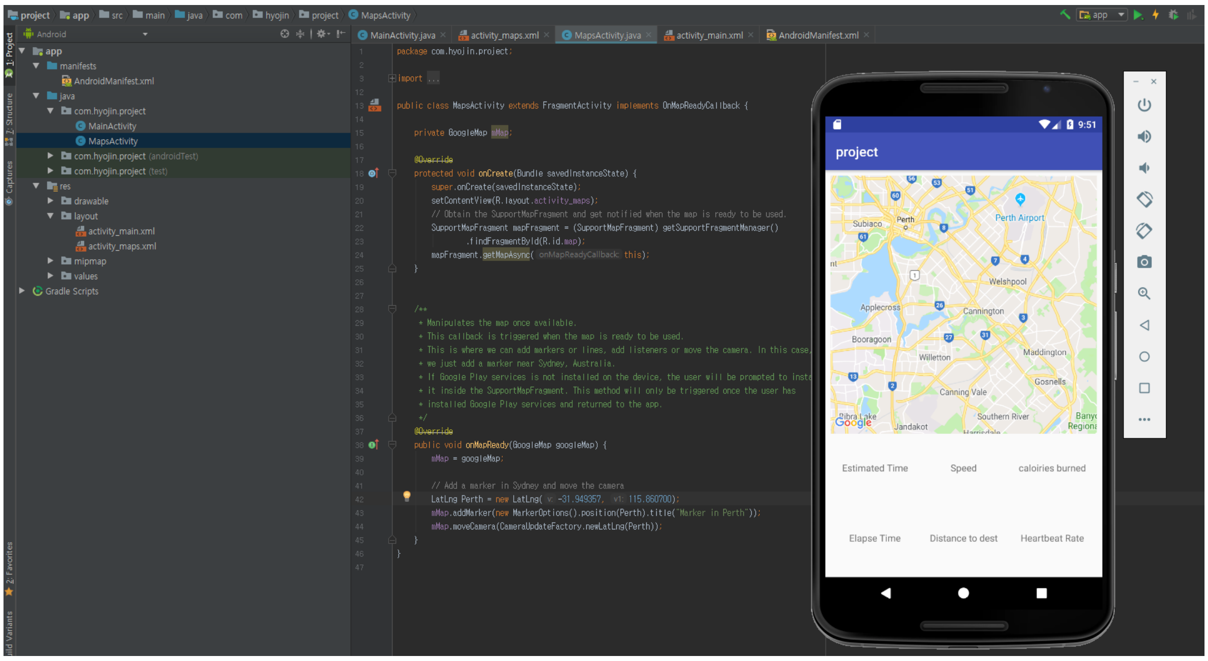
Use this key in your application by passing it with the key=API_KEY parameter.

Your API key

[redacted]

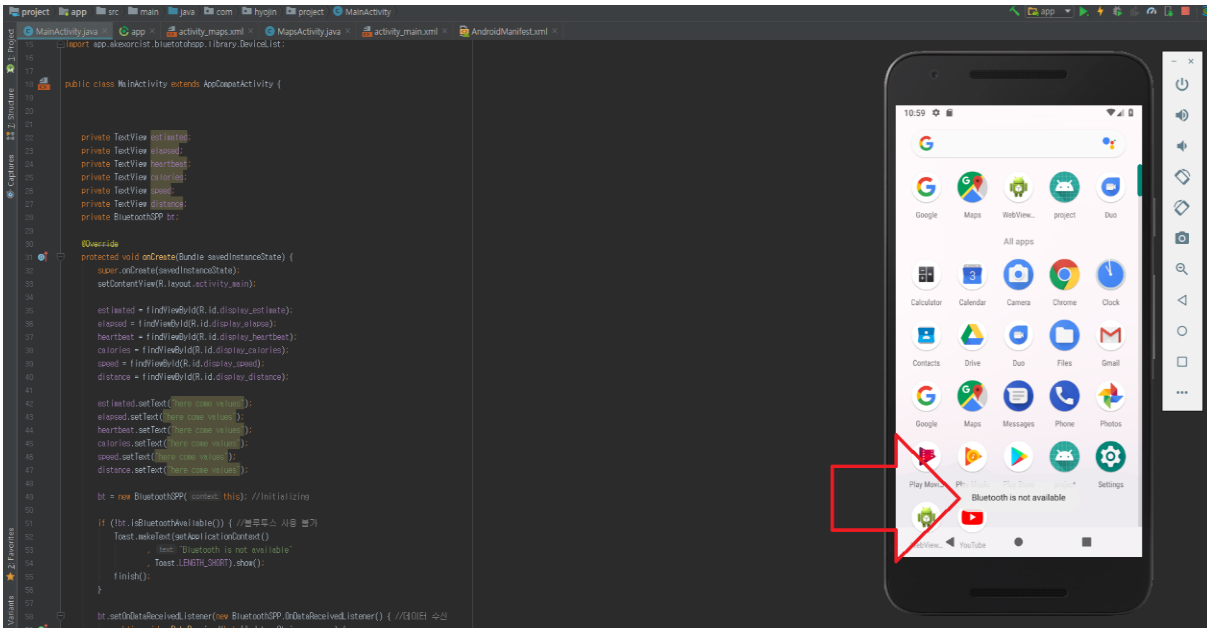
Restrict your key to prevent unauthorised use in production.

CLOSE RESTRICT KEY

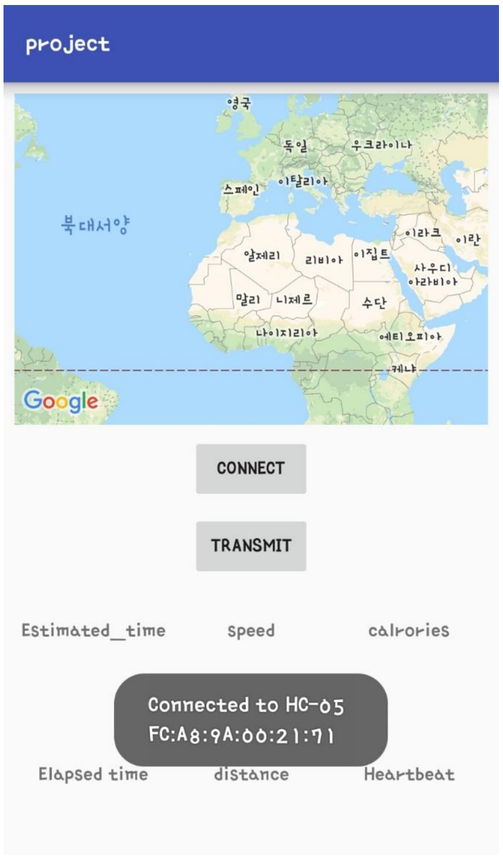


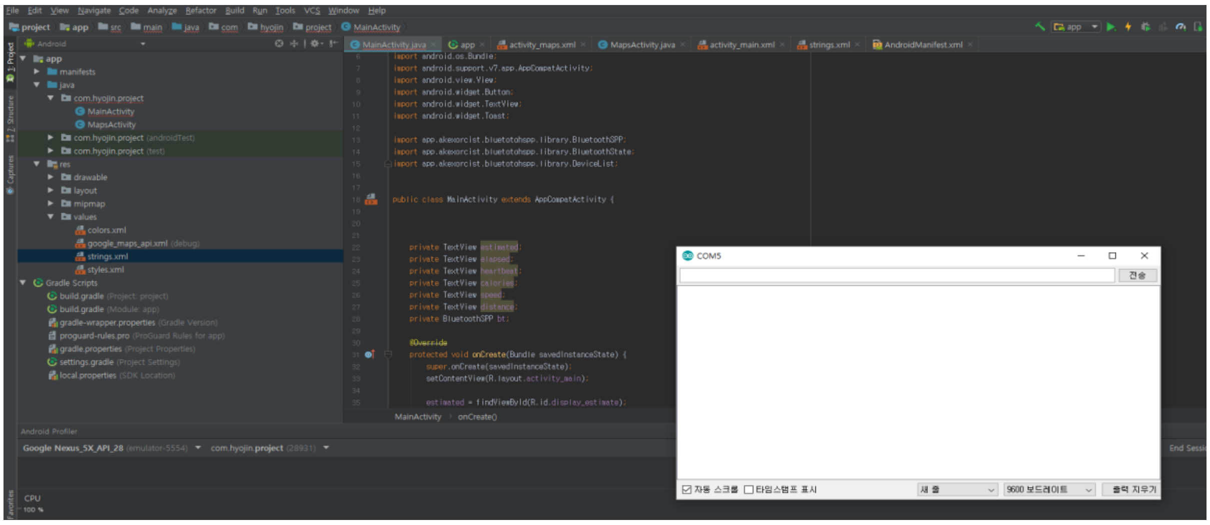
```java
package com.hyojin.project;

import ...

public class MapsActivity extends FragmentActivity implements OnMapReadyCallback {

    private GoogleMap mMap;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_maps);
        // Obtain the SupportMapFragment and get notified when the map is ready to be used.
        SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()
                .findFragmentById(R.id.map);
        mapFragment.getMapAsync( onMapReadyCallback: this);
    }


    /**
     * Manipulates the map once available.
     * This callback is triggered when the map is ready to be used.
     * This is where we can add markers or lines, add listeners or move the camera. In this case,
     * we just add a marker near Sydney, Australia.
     * If Google Play services is not installed on the device, the user will be prompted to insta
     * it inside the SupportMapFragment. This method will only be triggered once the user has
     * installed Google Play services and returned to the app.
     */
    @Override
    public void onMapReady(GoogleMap googleMap) {
        mMap = googleMap;

        // Add a marker in Sydney and move the camera
        LatLng Perth = new LatLng( v: -31.949357,  v1: 115.860700);
        mMap.addMarker(new MarkerOptions().position(Perth).title("Marker in Perth"));
        mMap.moveCamera(CameraUpdateFactory.newLatLng(Perth));
    }
}
```

Following step is the Bluetooth module set up in the studio, this is quiet complicated but coding is completed by using the reference web-site so it is working on the android emulator, however, emulator not provide the Bluetooth function so modified layout a bit and make as .apk file for installing on device. In the device, application is working and connect to Arduino Bluetooth module is successful, however, transmit the data to Arduino Bluetooth module is not success.



```java
import app.akexorcist.bluetotohspp.library.DeviceList;


public class MainActivity extends AppCompatActivity {



    private TextView estimated;
    private TextView elapsed;
    private TextView heartbeat;
    private TextView calories;
    private TextView speed;
    private TextView distance;
    private BluetoothSPP bt;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        estimated = findViewById(R.id.display_estimate);
        elapsed = findViewById(R.id.display_elapse);
        heartbeat = findViewById(R.id.display_heartbeat);
        calories = findViewById(R.id.display_calories);
        speed = findViewById(R.id.display_speed);
        distance = findViewById(R.id.display_distance);

        estimated.setText("here come values");
        elapsed.setText("here come values");
        heartbeat.setText("here come values");
        calories.setText("here come values");
        speed.setText("here come values");
        distance.setText("here come values");

        bt = new BluetoothSPP( context: this); //Initializing

        if (!bt.isBluetoothAvailable()) { //블루투스 사용 불가
            Toast.makeText(getApplicationContext()
                    , text: "Bluetooth is not available"
                    , Toast.LENGTH_SHORT).show();
            finish();
        }

        bt.setOnDataReceivedListener(new BluetoothSPP.OnDataReceivedListener() { //데이터 수신
```





```java
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import app.akexorcist.bluetotohspp.library.BluetoothSPP;
import app.akexorcist.bluetotohspp.library.BluetoothState;
import app.akexorcist.bluetotohspp.library.DeviceList;


public class MainActivity extends AppCompatActivity {



    private TextView estimated;
    private TextView elapsed;
    private TextView heartbeat;
    private TextView calories;
    private TextView speed;
    private TextView distance;
    private BluetoothSPP bt;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        estimated = findViewById(R.id.display_estimate);
```

The last step is database implementation the database coding has been done that create the schema and updating, adding and remove the data function is implemented, however, data is not receiving from map function and Arduino module so database part is not going to join the main activity in this stage because data add part required the specific variable but as mention above the specific variable is not exist in this stage. The code is attached last part of this report Appendix B.

Future Improvement

Currently, this project has two big problem that Bluetooth transmit issue and google map not working properly. Therefore, urgent improvement required at those section. For Bluetooth module may need the voltage divider that another supplier’s data sheet informed recommendation working voltage is 3.3V. additionally, pedal pressure sensor is used to plan implemented in beginning of this project, however, the pedal pressure sensor is discarded because of technical and component issue so it can be improved by replaced the Arduino Nano to Arduino Due then without wiring (such as Bluetooth) can communicate between pedal pressure sensor and micro-controller because Arduino Due provide 4 Serial port that means can connect maximum 4 TX/RX connection between modules. Also health management algorithm can be updated because in this project only use the one type of calories burned formula but for more accurate value of calories burned can be used vary type of formula.

Conclusion

The project is purposed to provide health management system while riding bicycle. Similar product is released in the market but the commercial product is mainly focus on entertaining purpose such as speed and angle of slope also price is quite expensive. So beginning of the project, targeting the inexpensive cost and focus on more health management.

In a nutshell, the project has been failed because the main reason is Bluetooth communication channel is not working so the value can gain from the sensors, however, those value not transmit to device so pointless for gaining the data. Another fatal issue is google map is not working properly, it show the map but supplementary function is important for this project, therefore, the value display part in application is not showing anything because 3section from the sensors modules and rest of values are got from google map information.

Nonetheless, the project is precious experience for final year of university. Through the project can be revision of the whole curriculum and research the new knowledge without any guideline so that make chance to improve the skill and knowledge widely.

Reflection

In this project, I can learn many new knowledge such android programming, embedded system and understanding how to manage the project planning. Because first planning was not possible to process in limited time so beginning of project2 I have to revise the plan that can be finished in semester2. Unfortunately, the revised plan was not successful as well because the component delivery time took too long time also component output is different to theory. Furthermore, android programming is very sensitive programming because even make little mistake in the code, it has not shown any information why the application is not working. Through all of those situation, it not only makes me to learn academic knowledge that makes me learn self-management.

In conclusion, I cannot say the project is successful and doing well, however, the project make a chance to learn the new programming, android programming and XML layout editing, and thinking about how to manage the project schedule. Also processing the project alone make me feel having confidence for I can do any project if I had enough time for it. Therefore, this experience would be useful for in my future projects also career.

References

n.d. *Arduino AND Bluetooth HC-05 Connecting Easily.* https://www.instructables.com/id/Arduino-AND-Bluetooth-HC-05-Connecting-easily/.

n.d. *Arduino Bike Speedometer.* https://www.instructables.com/id/Arduino-Bike-Speedometer/.

n.d. *Arduino-Nano.* https://components101.com/microcontrollers/arduino-nano.

be?, What should my heart rate. n.d. https://www.medicalnewstoday.com/articles/235710.php.

n.d. *CONNECTED HEALTH & WELLNESS.* https://evrythng.com/industries/health-and-wellness/.

n.d. *Cycling - health benefits .* https://www.betterhealth.vic.gov.au/health/healthyliving/cycling-health-benefits.

n.d. *DIY Speedometer on Arduino.* https://www.instructables.com/id/DIY-Speedometer-on-Arduino/.

n.d. *Fomular Behand Calorie Calculators.* http://www.calories-calculator.net/Calculator_Formulars.html#burned_by_hr.

n.d. *HC-05 - Bluetooth Module.* https://components101.com/wireless/hc-05-bluetooth-module.

n.d. *Pulse Sensor and Arduino – Interfacing.* http://www.circuitstoday.com/pulse-sensor-arduino.

n.d. *Pulse Sensor With Arduino Tutorial.* https://www.instructables.com/id/Pulse-Sensor-With-Arduino-Tutorial/.

Tutorial, How to Make Arduino Based Digital Tachometer Simple DIY. n.d. https://www.instructables.com/id/Ho-to-Make-Arduino-Based-Digital-Tachometer-Simple/.

n.d. *Why do you have to use a voltage divider with HC-05 bluetooth module? (Arduino).* https://electronics.stackexchange.com/questions/280500/why-do-you-have-to-use-a-voltage-divider-with-hc-05-bluetooth-module-arduino.

Appendices

Appendix A

Pulse sensor Arduino code

Please download the zip file for processing the pulse sensor module:

https://github.com/WorldFamousElectronics/PulseSensor_Amped_Arduino

Bluetooth module Arduino code

```
#include <SoftwareSerial.h>
// Define the data transmit/receive pins in Arduino
#define TxD 31
#define RxD 30
SoftwareSerial mySerial(RxD, TxD); // RX, TX for Bluetooth
void setup() {
   mySerial.begin(9600); // For Bluetooth
   Serial.begin(9600); // For the IDE monitor Tools -> Serial Monitor
}
void loop() {
      byte c;
      //c = mySerial.read(); // here come to heart beat rate and speed value
      Serial.print(c); // Print the character received to the IDE serial monitor
}
```

Hall Effect Sensor Module

```
 unsigned long lastturn;

 float length = 2.070; // 26 inch wheel

 volatile float Dist=0;

 volatile float speed;

 void setup()

 {

    Serial.begin(9600);

    attachInterrupt(0, magnet_detect, RISING);//Initialize the intterrupt pin (Arduino
digital pin 2)

    lastturn = 0;

 }

 void loop()//Measure RPM

 {

    if (millis() - lastturn >80) {

       lastturn = millis();

       speed = length    / ((millis() - lastturn) / 1000) *3.6;

       Dist = Dist + length / 1000;

       Serial.println(speed,DEC);

    }

 }

 void magnet_detect()//This function is called whenever a magnet/interrupt is
detected by the arduino

 {

    half_revolutions++;

    Serial.println("detect");

 }
```

Appendix B

Layout Code

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">
    <Button
        android:id="@+id/btnConnect"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="connect"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/btnSend"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:text="transmit"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/btnConnect" />
    <fragment
        android:id="@+id/map"
        class="com.google.android.gms.maps.MapFragment"
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginLeft="8dp"
        android:layout_marginRight="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        app:layout_constraintBottom_toTopOf="@+id/btnConnect"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />


    <LinearLayout
        android:id="@+id/linearLayout2"
        android:layout_width="match_parent"
        android:layout_height="100dp"
        android:layout_marginBottom="4dp"
        android:layout_marginEnd="8dp"
        android:layout_marginLeft="8dp"
        android:layout_marginRight="8dp"
        android:layout_marginStart="8dp"
        android:orientation="horizontal"
        app:layout_constraintBottom_toTopOf="@+id/linearLayout"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent">

        <TextView
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:text="@string/estimated_time"
            android:gravity="center_horizontal|center_vertical"
            android:id="@+id/display_estimate"/>

        <TextView
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:text="@string/speed"
            android:gravity="center_horizontal|center_vertical"
            android:id="@+id/display_speed"/>

        <TextView
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:text="@string/calories"
            android:gravity="center_horizontal|center_vertical"
            android:id="@+id/display_calories"/>
    </LinearLayout>

    <LinearLayout
        android:id="@+id/linearLayout"
        android:layout_width="match_parent"
        android:layout_height="100dp"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginLeft="8dp"
        android:layout_marginRight="8dp"
        android:layout_marginStart="8dp"
        android:orientation="horizontal"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent">

        <TextView
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:text="@string/elapse_time"
            android:gravity="center_horizontal|center_vertical"
            android:id="@+id/display_elapse"/>
        <TextView
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:text="@string/distance"
            android:gravity="center_horizontal|center_vertical"
            android:id="@+id/display_distance"/>

        <TextView
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:text="@string/heartbeat"
            android:gravity="center_horizontal|center_vertical"
            android:id="@+id/display_heartbeat"/>
    </LinearLayout>

</android.support.constraint.ConstraintLayout>
```

Main Activity JAVA code

```
package com.hyojin.project;

import android.app.Activity;
import android.bluetooth.BluetoothAdapter;
import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import app.akexorcist.bluetotohspp.library.BluetoothSPP;
import app.akexorcist.bluetotohspp.library.BluetoothState;
import app.akexorcist.bluetotohspp.library.DeviceList;


public class MainActivity extends AppCompatActivity {



    private TextView estimated;
    private TextView elapsed;
    private TextView heartbeat;
    private TextView calories;
    private TextView speed;
    private TextView distance;
    private BluetoothSPP bt;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        estimated = findViewById(R.id.display_estimate);
        elapsed = findViewById(R.id.display_elapse);
        heartbeat = findViewById(R.id.display_heartbeat);
        calories = findViewById(R.id.display_calories);
        speed = findViewById(R.id.display_speed);
        distance = findViewById(R.id.display_distance);

        estimated.setText("Estimated_time");
        elapsed.setText("Elapsed time");
        heartbeat.setText("Heartbeat");
        calories.setText("calrories");
        speed.setText("speed");
        distance.setText("distance");

        bt = new BluetoothSPP(this); //Initializing

        if (!bt.isBluetoothAvailable()) { //블루투스 사용 불가
            Toast.makeText(getApplicationContext()
                    , "Bluetooth is not available"
                    , Toast.LENGTH_SHORT).show();
            finish();
        }

        bt.setOnDataReceivedListener(new BluetoothSPP.OnDataReceivedListener() { //데이터 수신
            public void onDataReceived(byte[] data, String message) {
                Toast.makeText(MainActivity.this, message, Toast.LENGTH_SHORT).show();
            }
        });

        bt.setBluetoothConnectionListener(new BluetoothSPP.BluetoothConnectionListener()
{ //연결됐을 때
            public void onDeviceConnected(String name, String address) {
                Toast.makeText(getApplicationContext()
                        , "Connected to " + name + "₩n" + address
                        , Toast.LENGTH_SHORT).show();
            }

            public void onDeviceDisconnected() { //연결해제
                Toast.makeText(getApplicationContext()
                        , "Connection lost", Toast.LENGTH_SHORT).show();
            }

            public void onDeviceConnectionFailed() { //연결실패
                Toast.makeText(getApplicationContext()
                        , "Unable to connect", Toast.LENGTH_SHORT).show();
            }
        });

        Button btnConnect = findViewById(R.id.btnConnect); //연결시도
        btnConnect.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                if (bt.getServiceState() == BluetoothState.STATE_CONNECTED) {
                    bt.disconnect();
                } else {
                    Intent intent = new Intent(getApplicationContext(), DeviceList.class);
                    startActivityForResult(intent, BluetoothState.REQUEST_CONNECT_DEVICE);
                }
            }
        });

    }
    public void onDestroy() {
        super.onDestroy();
        bt.stopService(); //블루투스 중지
    }

    public void onStart() {
        super.onStart();
        if (!bt.isBluetoothEnabled()) { //
            Intent intent = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
            startActivityForResult(intent, BluetoothState.REQUEST_ENABLE_BT);
        } else {
            if (!bt.isServiceAvailable()) {
                bt.setupService();
                bt.startService(BluetoothState.DEVICE_OTHER); //DEVICE_ANDROID는 안드로이드 기기
끼리
                setup();
            }
        }
    }

    public void setup() {
        Button btnSend = findViewById(R.id.btnSend); //데이터 전송
        btnSend.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                bt.send("Text", true);
            }
        });
    }

    public void onActivityResult(int requestCode, int resultCode, Intent data) {
        if (requestCode == BluetoothState.REQUEST_CONNECT_DEVICE) {
            if (resultCode == Activity.RESULT_OK)
                bt.connect(data);
        } else if (requestCode == BluetoothState.REQUEST_ENABLE_BT) {
            if (resultCode == Activity.RESULT_OK) {
                bt.setupService();
                bt.startService(BluetoothState.DEVICE_OTHER);
                setup();
            } else {
                Toast.makeText(getApplicationContext()
                        , "Bluetooth was not enabled."
                        , Toast.LENGTH_SHORT).show();
                finish();
            }
        }
    }
```

```
}
```

Map source XML

```
<?xml version="1.0" encoding="utf-8"?>
<fragment xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:map="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/map"
    android:name="com.google.android.gms.maps.SupportMapFragment"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MapsActivity" />
```

Map Fragment JAVA code

```
package com.hyojin.project;

import android.support.v4.app.FragmentActivity;
import android.os.Bundle;

import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;

public class MapsActivity extends FragmentActivity implements OnMapReadyCallback {

    private GoogleMap mMap;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_maps);
        // Obtain the SupportMapFragment and get notified when the map is ready to be used.
        SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()
                .findFragmentById(R.id.map);
        mapFragment.getMapAsync(this);
    }
    @Override
    public void onMapReady(GoogleMap googleMap) {
        mMap = googleMap;
        // Add a marker in Sydney and move the camera
        LatLng Perth = new LatLng(-31.949357, 115.860700);
        mMap.addMarker(new MarkerOptions().position(Perth).title("Marker in Perth"));
        mMap.moveCamera(CameraUpdateFactory.newLatLng(Perth));
    }
}
```

Data.java

```java
package com.hyojin.project;

public class Data {
    private double mDistance;
    private int mCalroie;
    private int mTime;
    private byte[] maMap;


    Data(byte[] map, int time, double distance, int calroie)
    {
        this.maMap = map;
        this.mTime = time;
        this.mDistance = distance;
        this.mCalroie = calroie;
    }

    public double getDistance() {
        return mDistance;
    }

    public void setDistance(double distance) {
        mDistance = distance;
    }

    public int getCalroie() {
        return mCalroie;
    }

    public void setCalroie(int calroie) {
        mCalroie = calroie;
    }

    public int getTime() {
        return mTime;
    }

    public void setTime(int time) {
        mTime = time;
    }

    public byte[] getMaMap() {
        return maMap;
    }

    public void setMaMap(byte[] maMap) {
        this.maMap = maMap;
    }
}
```

dataSchema.java

```
package com.hyojin.project;

public class dataSchema {
    public static class dataTable
    {
        public static final String NAME = "bicycleData";
        public static class cols
        {
            public static final String travelTime = "travel_time";
            public static final String caloriesBurned= "calories_wasted";
            public static final String Disatnce= "distance";
            public static final String mapInfo ="MapInfo";
        }
    }
}
```

dataStore.java

```
package com.hyojin.project;

import android.content.ContentValues;
import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import com.hyojin.project.dataSchema.dataTable;

public class dataStore {
    private SQLiteDatabase db;
    public dataStore(Context context)
    {
        this.db = new database(context.getApplicationContext()).getWritableDatabase();
    }

    public void addData(Data data)
    {
        ContentValues cv = new ContentValues();
        cv.put(dataTable.cols.mapInfo, data.getMaMap());
        cv.put(dataTable.cols.travelTime, data.getTime());
        cv.put(dataTable.cols.travelTime, data.getDistance());
        cv.put(dataTable.cols.caloriesBurned, data.getCalroie());
        db.insert(dataTable.NAME, null, cv);
    }

    public void updateData(Data data)
    {
        ContentValues cv = new ContentValues();

        String[] whereValue = { String.valueOf(data.getMaMap())};
        db.update(dataTable.NAME, cv, dataTable.cols.mapInfo + " = ?", whereValue);
    }

    public void removeData(Data data)
    {
        String[] whereValue = { String.valueOf(data.getMaMap())};
        db.delete(dataTable.NAME, dataTable.cols.mapInfo + " = ?", whereValue);
    }
}
```

database.java

```
package com.hyojin.project;

import com.hyojin.project.dataSchema.dataTable;
import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

public class database extends SQLiteOpenHelper{


    private static final int VERSION = 1;
    private static final String dataBase_Name = "bicycleData.db";

    public database (Context context)
    {
        super(context, dataBase_Name, null, VERSION);
    }



    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL("create table "+ dataTable.NAME+ "(" +
                "_id integer primary key autoincrement, "+
                dataTable.cols.mapInfo +", "+
                dataTable.cols.travelTime+", "+
                dataTable.cols.Disatnce+", "+
                dataTable.cols.caloriesBurned + ")");
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int a, int b) {
        // to be added additional function
    }
}
```