

# 설계과제 최종보고서

설계과제명 : 자전거 도난 방지 장치

교과목명	어드벤처 디자인	
담당교수	정진우 교수님	
팀 원	학 번	이 름
	2019111985	남상원
	2018110399	변찬현
	2020111990	이승지
	2020112138	이우제

# 설계과제 요약서

과 제 요약서	
설계과제명	물건 분실 방지와 잃어버린 물건 찾기
주요기술용어 (2~7개 단어)	블루투스, GPS기술
<ul style="list-style-type: none"><li>과제 목표 자전거의 분실 위험성을 줄이고 물건과 휴대폰의 송수신을 하여 자전거의 움직임을 감지해 자전거 도난을 방지하는 것이 목표입니다.</li><li>수행 내용 MPU6050 센서를 이용하여 기울임을 감지해 휴대폰과 물건의 블루투스로 송수신을 통해 사용자에게 알림을 올린다. GPS센서를 통하여 위도, 경도값과 거리를 측정하고 앱에 있는 버튼을 누를 경우 부저를 울리도록 하였다.</li><li>수행 결과 실행결과 기울임을 감지후에 GPS센서를 통해 위도,경도와 거리에 대한 값이 잘 나왔다. 분실했다고 판단하기 위해 기울기의 값에 범위(45도)를 제한하였다.</li><li>결과 분석 실내에서는 0 의 값이 뜨는 경우가 발생하였지만 실외에서는 휴대폰과 물건을 블루투스로 연결하였을 때 페어링이 잘 되었고, GPS를 통한 결과값이 잘 도출됐다.</li></ul>	

## 제 1 장 설계과제 문제 정의

### 1.1 문제 정의

해가 갈수록 자전거 도난 분실이 크게 늘고 있다. 고가의 자전거를 분실하여도 분실한지도 모른 채 찾을 수 없는 사건도 있다. 살면서 물건은 언제든지 잃어버릴 수 있다. 특히 집에 들여놓지 못하는 자전거 같은 경우는 더욱더 분실 위험이 크다. 자전거를 나아가서, 어떤 물품의 분실 방지를 위하여 휴대폰을 통하여 물건의 움직임을 감지하여 위치를 찾으면 분실의 위험성을 낮출 수 있다. 이때 물건과 휴대폰의 통신은 블루투스를 이용하고, 물건의 위치파악은 GPS기술을 이용한다.

## 제 2 장 설계과제 요구사항 분석

### 2.1 기능 요구 조건

- 절에서 기술한 문제를 해결하기 위해 필요한 요구기능들을 나열한다.

블루투스 앱

- 휴대폰과 물건의 위치의 송수신 기능
- 물건의 움직임을 감지하여 알림하는 기능
- 물건과의 통신을 위한 1차적인 기능

GPS센서, 기울기 센서

- 물건과의 통신을 위한 2차적인 기능
- 물건의 기울기의 범위를 정해 움직임을 감지하는 기능
- 블루투스가 끊기더라도 물건의 위치를 파악하는 기능
- 먼 거리에서도 물건의 위치를 정확히 파악하는 기능
- 분실을 하였더라도 물건의 위치를 알 수 있어야 함

- 성능 규격 조건

- 어느정도 기울기를 때 물건이 분실했는지 기울기의 범위
- 블루투스의 거리 한계값(m)와 GPS가 움직여야하는 거리의 범위(m)
- GPS기능이 정상적으로 작동이 가능한건지 여부

### 제 3 장 설계과제 제한요소 분석

#### • 동작환경

- 자전거와 같은 물건들의 기울임이 45도의 범위를 넘어서면 분실가능성이 있다고 판단한다.
- 기울임을 감지하거나(OR) GPS 센서를 통해 도출된 휴대폰 사이의 간격이 5m 이상이 되면 휴대폰에 블루투스를 이용하여 알림을 준다. 여기서 기준 거리의 기본값은 5m이며 사용자 요구에 따라 앱인터페이스를 통해 기준 거리를 조절할 수 있다.
- .휴대폰 앱(Bluetooth Interface)을 통해 사용자와 물건(아두이노 회로)의 위도, 경도 값을 보여 주며 도난 위험이 있을시 버튼을 통해 부저와 LED를 작동시킬 수 있다.
- 물건 분실 방지 목적 외에 다른 기능으로 이용하면 안된다.
- 아두이노 회로는 상시 켜진 상태여야 한다.
- 회로가 물건에 부착이 되어하고, 부착될 회로의 크기가 작을수록 좋으므로 회로를 상자같은 곳에 모아 넣도록 한다.

#### • 개발환경

- 물건과 휴대폰간 블루투스 통신은 원활해야한다.
- 사용하는 센서외에 다른 센서의 사용은 허용하지 않는다.
- 버튼을 눌러 알리는 부저/LED 기능은 사용자가 선택하여 사용한다.
- 사용자의 휴대폰 설정에서 위치정보 접근을 허용으로 설정해야 한다.
- GPS-Air530 모듈을 사용하였는데, 실내보다 실외에서 작동이 원활하다.
- 회로가 물건에 부착이 되어하고, 부착될 회로의 크기가 작을수록 좋으므로 회로를 상자같은 곳에 모아 넣도록 한다.

#### • 개발비용

- 연결에 혼선을 줄 수 있는 다른 센서는 사용할 수 없다.
- 추가부품을 구매할 경우 만원 내로 가능하게 하고, 추가 부품의 추가시 공급 전력이 부족한 경우 9V 배터리 및 배터리 소켓을 추가할 수 있다.

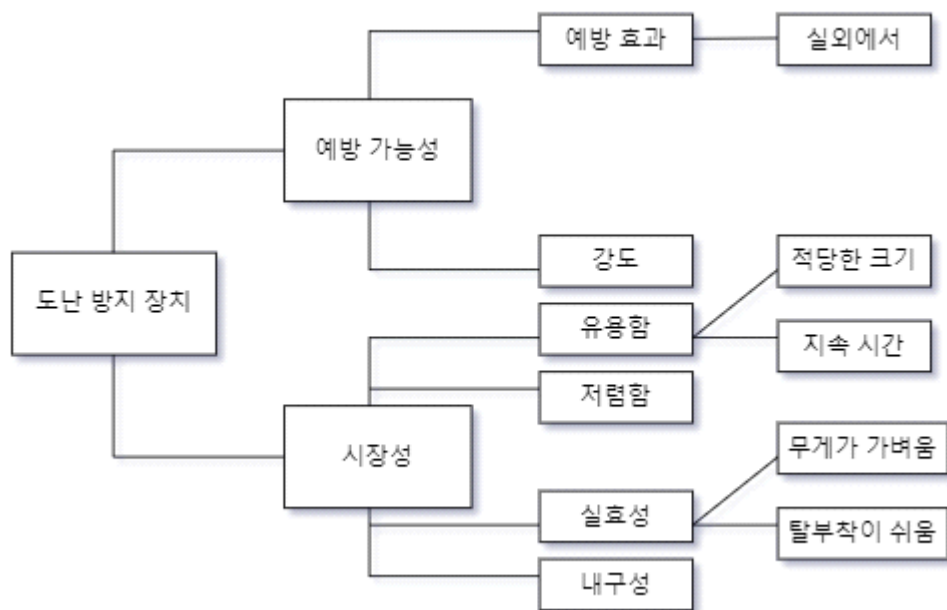
#### • 사회성

- 물건 외의 사람한테 붙이는 것은 허용하지 않는다.

- 어떤 사람을 쫓아갈 목적으로 다른 사람의 물건에 장치를 설치하는 것은 금한다.

## 제 4 장 설계안 도출

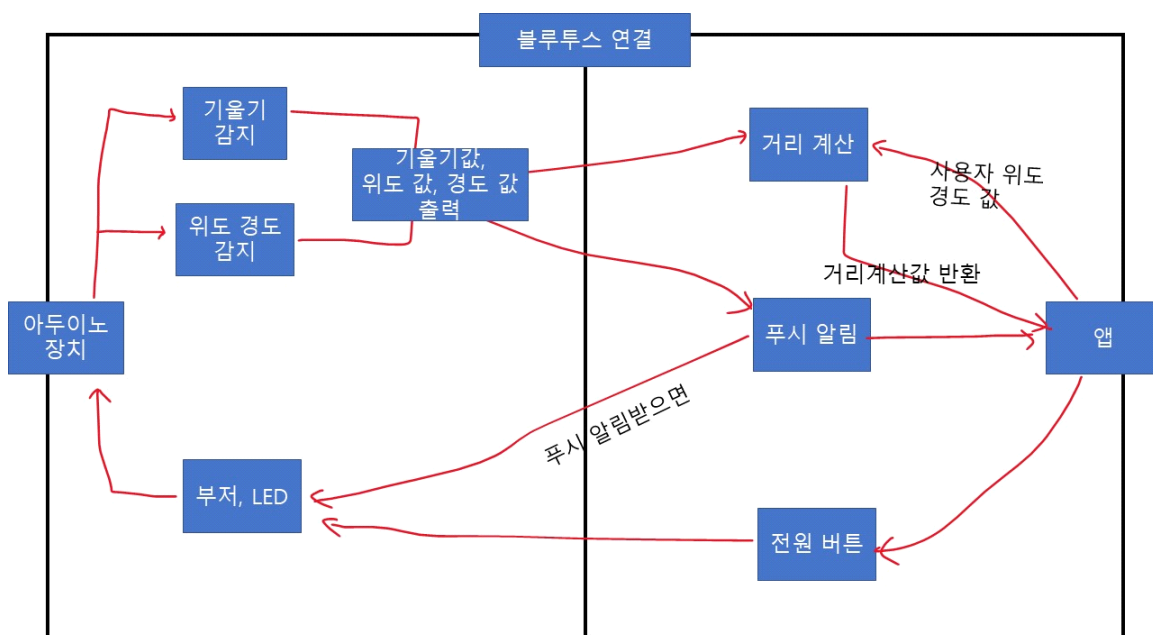
- 설계목표

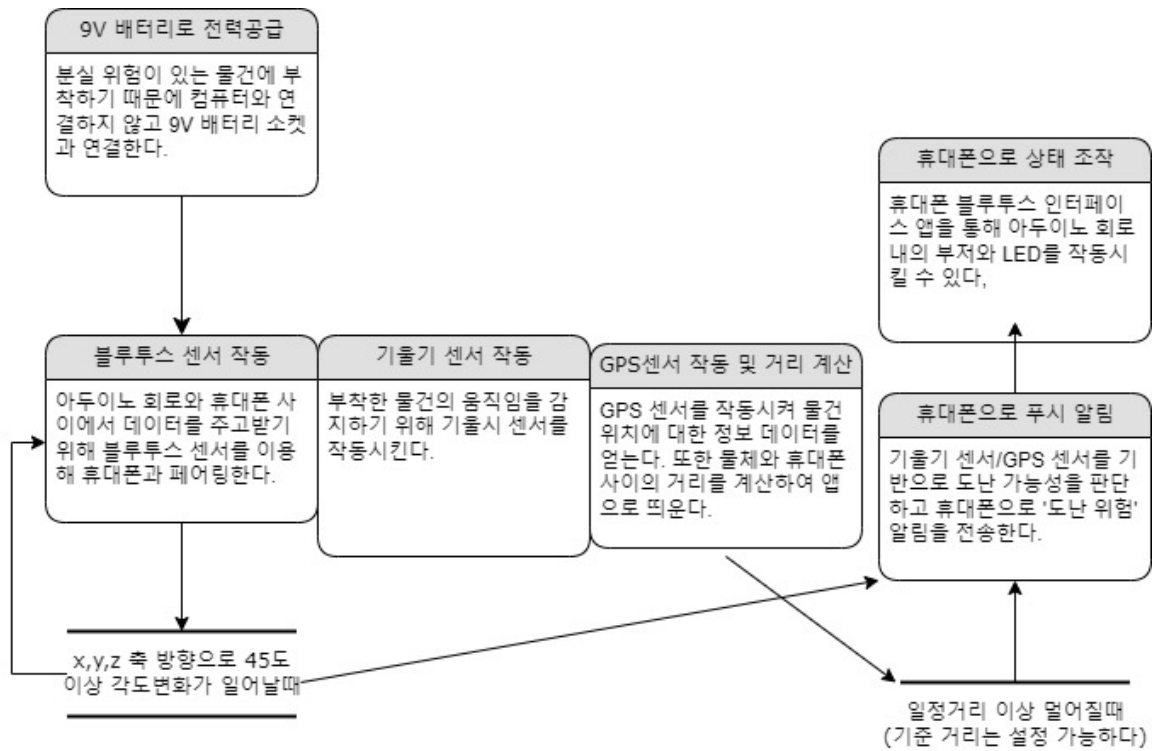


- 설계안

아두이노 회로에 전력이 공급되고, 휴대폰에서 hc-10 블루투스 모듈이 등록된 상태에서, 블루투스 모듈과 휴대폰 간 페어링이 된 상태에 기능을 수행한다. 우선 물건의 도난이 발생한 상황에 대해 설명하자면 범인이 물건에 손을 댈 경우 물건이 움직일 가능성이 있다. 또한 범인이 물건을 갖고 제자리에 있지 않고 이동할 가능성이 크다. 따라서 도난이 발생한 상황을 '물건의 움직임' 그리고 '물건과 사용자(물건의 주인)' 이렇게 두 가지 기준으로 판단하기로 하였다. 이때 범인이 물건을 들고 움직일때는 이동하는 물건의 가속도가 발생하지 않을 수도 있으므로 '물건에 움직임이 감지되면' 또는(OR) '물건과 주인 사이의 거리가 멀어지면' 도난 위험이 있다고 정의하였다. 이때 물건의 움직임을 감지하기 위해 가속도의 변화를 감지하는 '기울기 센서'와 센서의 위치 정보를 위도 및 경도 값으로 측정하는 GPS-Air530 모듈이 사용되었다. 기울기 센서를 통해 x,y,z 방향으로 45도 이상의 움직임이 감지되면 블루투스 모듈을 통해 무선으로 사용자의 휴대폰에 푸시알림("물체의 움직임이 감지되었습니다.")이 뜨도록 한다. 이를 본 사용자는 앱에 있는 버튼을 누를 수 있고 버튼을 누를시 신호를 다시 회로로 보내어 회로의 부저와 LED가 작동한다. 마찬가지로 물건

과 주인간의 거리가 멀어져도 푸시알림이 뜬다. GPS 모듈은 상시 작동하여 실시간으로 위도 및 경도 값을 측정하고 이를 블루투스 모듈을 통해 무선으로 휴대폰에 위치 정보를 전송한다. 이때 휴대폰 앱은 휴대폰의 위치 정보와 수신한 회로의 위치 정보를 비교하여 휴대폰과 회로 사이의 거리를 측정한다. 측정된 휴대폰과 회로의 위도 및 경도 그리고 둘 사이의 거리에 대한 정보는 휴대폰 화면으로 출력한다. 둘 사이의 거리값이 5m 이상일 경우(앱인벤터를 통해 5 대신 다른 숫자로 설정 가능합니다.) 블루투스 모듈을 통해 신호를 보내게 되고 휴대폰에서 푸시 알림이 뜬다. 이후론 기울기 센서 경우와 동일하게 사용자는 버튼을 통해 회로의 부저와 LED를 조종할 수 있다. 추가로 덧붙이자면 앱 제작툴 '앱인벤터' 통해 물건의 정보를 확인하고 회로를 조종할 수 있는 'Bluetooth Interface'라는 앱을 제작하였다. 앱인벤터는 코드 블록을 이용해서 앱의 기능을 구현할 수 있게 도와주며 코드에 대한 자세한 설명은 '제 5장 구현' 부분에서 다루었다.





## • 평가 항목과 평가 방법

기울기 센서 및 GPS 모듈이 잘못 측정할 경우 도난의 상황을 잘못 판단할 수 있기 때문에 두 센서 측정값과 실제 값의 오차를 평가항목으로 삼았다. 특히 GPS 모듈의 경우 아두이노에 사용된 GPS이기 때문에 오차 발생의 위험이 더 컸다.

평가항목	평가기준	근거 (보통 기준)	평가점수				
			아주미흡 (1)	미흡 (2)	보통 (3)	우수 (4)	아주우수 (5)
기울기 센서를 통한 각도 측정시 발생하는 오차 ('45-측정값'의 절대값)	각도	0	4	3	2	1	0
GPS 모듈을 통한 위치 정보 측정시 발생하는 오차 ('실제 거리-측정거리'의 절대값)	m	0	8	6	4	2	0

다음은 각 센서의 사용여부와 개발 및 동작 환경에 개입하는 평가항목들에 대한 평가 기준표이다.

평가항목	평가방법	평가기준
------	------	------

블루투스 모듈과 휴대폰 간 페어링이 된다.	5번 두 장치 사이의 페어링을 시도한다.	5번 모두 성공
휴대폰 앱에 정상적으로 위치 정보와 거리 정보가 출력된다.	두 장치간 페어링이 된 상태에서 화면에 제대로 값들이 출력되는지 10번 확인한다.	시도 횟수 10번 중 8번 성공(문제가 발생한 경우 아두이노 코드를 재업로드하면 대체로 해결됨)

평가항목	평가방법	평가기준
기울기 변화가 감지되면 휴대폰으로 푸시 알람이 전송된다.	기울기 센서에 변화를 주고 알람이 정상적으로 뜨는지 10번 시도한다.	10번 모두 성공
앱에서 버튼을 누를시 부저와 LED가 정상적으로 작동한다.	버튼을 누르면 부저와 LED가 정상적으로 작동하는지 10번 시도한다.	10번 모두 성공

## • 대안 분석

(예시) 의사결정 행렬표

	가 중 치	대안 1	대안 2	대안 3
개요		GPS-530을 통한 위치 정보 측정	HM-10 블루투스 모듈을 통한 위치 정보 측정	HC-06 블루투스 모듈을 통한 위치 정보 측정
회로의 위치 정보 측정에 대한 용이성 (목적)	25%	타 블루투스 모듈과 GPS 모듈 모두를 사용해본 결과, 거리 값 측정 오차는 적었다. <b>(점수: 50)</b>	블루투스 모듈의 경우 GPS 모듈보다는 낮은 오차를 출력하였다. 다만 실시간으로 거리값을 출력할 경우 출력되는 거리값의 기록이 심하였다. <b>(점수: 100)</b>	HM-10과 센서명이 블루투스 모듈로 같지만 부품 자체가 거리 측정 기능이 현저히 떨어졌다. <b>(점수: 80)</b>
센서 작동 환경에 대한 유연성 (목적)	5%	타 블루투스 모듈과 달리 실내보다는 실외에서 작동이 원활하여 실내의 경우 작동에 어려움이 있을 수 있다. <b>(점수: 60)</b>	GPS 모듈과 비교해서, 두 모듈 모두 거리 측정이 가능하지만 실외뿐만 아니라 실내에서도 원활하게 작동되는 점에서 차별점을 갖는다. <b>(점수: 100)</b>	센서 작동 환경에 대한 유연성 항목은 대안2와 동일하다. <b>(점수: 90)</b>



얼마나 세부적인 정보를 측정할 수 있는가 (목적)	10%	블루투스 모듈과 달리 위도와 경도 값을 측정할 수 있으며 거리계산식 알고리즘을 이용하여 거리값까지 알 수 있다.(점수: 60)	휴대폰과 센서간 거리 정보만 측정이 가능하다. (점수: 100)	대안 2와 동일하다. (점수: 70)
회로내 다른 센서/보드들과의 호환성 (목적)	5%	GPS 모듈의 경우 소프트웨어 시리얼을 사용하는데, 블루투스 모듈 역시 소프트웨어 시리얼을 사용하여 추후에 하드웨어 시리얼과 소프트웨어 시리얼을 분리해서 회로를 구성해야하는 어려움이 있었다. (점수: 100)	블루투스 모듈 역시 거리값 측정과 휴대폰간 송수신을 동시에 하기 위해서 다중 시리얼 문제를 해결해야했다. (점수: 70)	대안 2와 동일하다. (점수: 90)

설치 편의성(동작환경 제한요소)	5%	GPS 모듈 특성상 센서와 안테나가 선으로 연결되어 있어서 완성품을 만들때 테이프 등 접착도구를 사용해야할 수 있다. 그러나 센서 크기 자체는 블루투스 모듈과 유사하다. (점수: 100)	설치가 쉽다.(점수: 70)	대안 2와 동일 (점수: 90)
개발편의성 (개발환경 제한요소)	15%	실내에서는 사용이 어려울 수 있다. (점수: 100)	실내. 실외 모두 이용가능하다.(점수: 70)	대안 2와 동일(점수: 90)
User Interface 개발 편의성 (개발환경 제한요소)	5%	위도, 경도 프로토콜을 알고 파싱 코드를 알아야하여 과정이 복잡하다.. (점수: 90)	ibeacon 기술을 이용하면 쉽게 해결된다. (점수: 80)	대안 2와 동일 (점수: 80)
개발후 유지보수 (개발환경 제한요소)	5%	전력소모가 적지 않다. (점수: 80)	전력소모가 적다. (점수: 100)	전력소모가 적다. (점수: 100)

소)				
경제성 (제한요소)	25%	모듈 자체의 값이 비싼 편이며 해외배송으로 얻을 수 있다. <b>(점수: 100)</b>	접근이 쉽다. <b>(점수: 70)</b>	대안 2와 동일 <b>(점수: 100)</b>
일정 (제한요소)	15%	1주일 <b>(점수: 100)</b>	하루 <b>(점수: 80)</b>	하루 <b>(점수: 90)</b>
총합	100%	<b>88</b>	<b>72</b>	<b>40</b>

위 표에 의하여 최종 대안은 대안 1으로 결정하였다.

## 제 5 장 구현

### • 구현 방법

```
#include <MPU6050_tockn.h> // MPU-6050라이브러리 사용함 -> 이를 통해 상보필터 값을 계산해 줌
#include <Wire.h> // I2C 통신을 위한 라이브러리
#include <SoftwareSerial.h> // 블루투스 통신을 위한 라이브러리
#include <TinyGPSPlus.h> // GPS 위도, 경도 값을 읽어오기 위한 라이브러리

// 블루투스 모듈은 하드웨어 시리얼로 통신 (0, 1)번 핀 사용
SoftwareSerial gpsSerial(11,12); // GPS를 소프트웨어 시리얼로 통신 (11, 12)번 핀 사용
MPU6050 mpu6050(Wire); //I2C 통신을 통해 MPU6050값 입력 받기

TinyGPSPlus gps; // TinyGPSPlus = gps로 함

int speakerPin = 6; // 스피커 핀 = 6번 핀
int numTones = 4; // 음 출력 횟수 = 4번
int ledPin = 8; // LED 핀 8번 핀
float output[3]; // output[0] = 움직임, output[1] = 위도, output[2] = 경도 값 저장용
char character[10] = {""}; // \n 인지 구분 및 str에 저장.
String str = ""; // \n 전까지 character[0] 값을 저장.
String targetStr = "GPGGA"; // str의 값이 NMEA의 GPGGA 값인지 타겟

int tones[] = {392, 415, 392, 415}; // { 4옥솔, 4옥솔#, 4옥솔, 4옥솔# }
```

```

void setup()
{
    // 시리얼 통신을 사용하도록 만드는 함수
    // 9600의 속도로 통신 (통상적으로 사용하는 속도)
    // 보드 레이아웃이 달라지면 시리얼 모니터의 포트 깨짐
    Serial.begin(9600); // = 하드웨어 시리얼로 연결한 블루투스 모듈
    gpsSerial.begin(9600);

    // 각 핀을 출력 모드로
    pinMode(speakerPin, OUTPUT); // 스피커 핀
    pinMode(ledPin, OUTPUT); // LED 핀

    Wire.begin(); // I2C 통신 활성화
    mpu6050.begin(); // 각도 변환 활성화
    mpu6050.calcGyroOffsets(true); // 위치 값 계산
}

void loop()
{
    // mpu6050칩에 가속도센서, 자이로센서가 내장되어 두 개의 방법으로 각도를 계산
    // 이를 상보필터를 통해 비율을 설정해 가속도를 통한 각도와 자이로(각속도)를 통한 각도의 장점을 취함
    // 가속도 - 진동에 약하지만 긴시간에 정확한 값
    // 자이로 - 진동에 강해서 짧은 시간에 정확한 값을 얻지만 오차가 쌓여서 긴시간에 취약
    // 상보 필터 : filtered_angle = (0.02 * accel) + (0.98 * gyro) (이 코드에 적용된 식)
    mpu6050.update(); // 각도 변화 갱신
    float angle_x = mpu6050.getAngleX(); //x축 각도 변화
    float angle_y = mpu6050.getAngleY(); //y축 각도 변화
    float angle_z = mpu6050.getAngleZ(); //z축 각도 변화

```

```

if (Serial.available())
{
    char cmd = char(Serial.read()); // 핸드폰에서 읽어온 데이터

    // 전원 버튼을 통해 보낸 신호 값 '1'을 받으면
    if (cmd == '1')
    {
        for(int i = 0; i < numTones; i++)
        {
            tone(speakerPin, tones[i]); // i번째 tone[]의 음을 8번핀을 통해 출력
            delay(300); // 0.3초 딜레이
            digitalWrite(speakerPin, LOW); // 8번핀의 입력을 받아 LED 끄기
            delay(100); // 0.1초 딜레이
            digitalWrite(ledPin, HIGH); // 8번핀의 입력을 받아 LED 켜기
            delay(300); // 0.3초 딜레이
            digitalWrite(ledPin, LOW); // 8번핀의 입력을 받아 LED 끄기
            delay(100); // 0.1초 딜레이

            // 실행 도중 전원 버튼을 통해 보낸 신호 값 '1'을 받으면 부저 및 LED 종료
            char cmd = char(Serial.read());
            if (cmd == '2')
                break;
        }
        noTone(speakerPin); // 스피커 핀인 6번 핀의 출력을 멈춤

        delay(200); // 0.2초 딜레이
    }
}

if ( angle_x > 45 || angle_x < -45 || angle_y > 45 || angle_y < -45 || angle_z > 45 || angle_z < -45 )
    output[0] = 45;
else
    output[0] = 0;

// gps의 값을 읽어오기
while(gpsSerial.available())
{
    gps.encode(gpsSerial.read());
}
// gps의 위치가 업데이트되면
if(gps.location.isUpdated())
{
    output[1] = gps.location.lat(); // 위도 값
    output[2] = gps.location.lng(); // 경도 값
    // Start, (45 or 0), (위도 값(소수점 5자리까지)), (경도 값(소수점 5자리까지)), Stop 형태로 출력
    String data = String("Start")+","+String(output[0])+","+String(output[1],5)+
        "+String(output[2],5)+"+"+String("Stop");

    Serial.println(data); // data를 블루투스 모듈에 출력
    delay(100); // 딜레이 0.1
}

```

```

to 블루투스 연결시 초기화
do
  set 이미지1 - Picture - to IAMLogoOn.png
  set 블루투스 버튼 - Image - to BTButtonOn.png

```

- <블루투스 연결시> <함수>
- 1번 그림이 1-1번 그림으로 변경
  - 2번 그림이 2-1번 그림으로 변경

```

to 블루투스 연결 끊기
do
  call 블루투스 플라이언트1 - Disconnect
  set 이미지1 - Picture - to IAMLogoOff.png
  set 블루투스 버튼 - Image - to BTButtonOff.png
  set 신호 버튼 - Image - to power-on.png
  set 거리 숫자 - Text - to 0 join "KM"
  set 장치 위치 값 - Text - to 0 join "위도"
  set 장치 경도 값 - Text - to 0 join "경도"

```

- <블루투스 연결을 끊을 시> <함수>
- 블루투스 연결을 끊음
  - 1번 그림이 1-1번 그림으로 변경
  - 2번 그림이 2-1번 그림으로 변경
  - 4-1번 그림과 같이 출력되던 것이 4번 그림과 같이 변경

```

when Screen1 - Initialize
do
  set Screen1 - BackgroundColor - to make color 73 91 122
  call 블루투스 연결 끊기
  set 시작1 - TimerEnabled - to true
  set 시작1 - TimerInterval - to 1000

```

- <화면 초기화>
- 배경 색 설정
  - 블루투스 연결을 끊은 상태로
  - 1초마다 화면 갱신

```

when 블루투스 목록 - AfterPicking
do
  if call 블루투스 플라이언트1 - Connect address 블루투스 목록 - Selection
  then call 블루투스 연결시 초기화

```

- <블루투스 목록에서 고른 후>
- 블루투스가 연결되어 있다
  - <블루투스 연결> 함수 실행

<전역 변수>  
버튼 = off

<신호 보내기 버튼 클릭 시>(3번 그림)

```

when 신호 버튼 - Click
do
  if 블루투스 플라이언트1 - isConnected
  then
    if get global 버튼 = on
    then
      set 신호 버튼 - Image - to power-on-blue.png
      call 블루투스 플라이언트1 - SendText text 1
      set global 버튼 - to off
    else if get global 버튼 = off
    then
      set 신호 버튼 - Image - to power-on.png
      call 블루투스 플라이언트1 - SendText text 2
      set global 버튼 - to on
    end if
  end if

```

- 블루투스가 연결되어 있다면
- 버튼이 꺼져 있을 때
- 3번 그림 -> 3-1번 그림으로 변경
- 핸드폰에 '1'을 보냄
- <전역 변수> 버튼을 on으로 변경
- 버튼이 켜져 있을 때
- 3-1번 그림 -> 3번 그림으로 변경
- 핸드폰에 '2'을 보냄
- <전역 변수> 버튼을 off로 변경

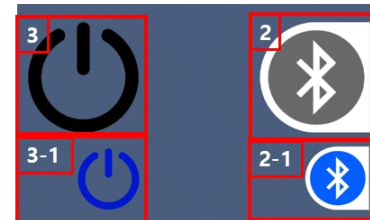
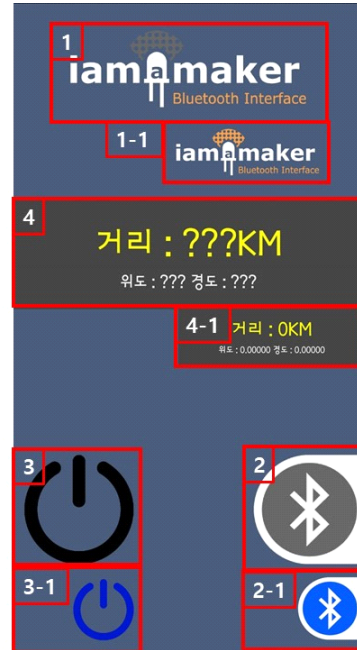
```

when 블루투스 버튼 - Click
do
  if 블루투스 플라이언트1 - Enabled
  then
    if 블루투스 플라이언트1 - isConnected
    then
      call 블루투스 ShowChooseDialog
      message 블루투스 연결을 끊으시겠습니까?
      title 블루투스 연결 끊기
      button1Text 0
      button2Text 1
      cancelable false
    else
      set 블루투스 버튼 - Elements - to 블루투스 플라이언트1 - AddressesAndNames
      call 블루투스 목록 - Open
    end if
  else
    call 블루투스 ShowChooseDialog
    message 설정에서 블루투스를 활성화 시켜주세요!
    title 블루투스가 꺼져 있습니다
    button1Text 1
    button2Text 2
    cancelable false
  end if

```

블루투스 연결 끊기  
블루투스 연결을 끊으시겠습니까?  
예 아니요

블루투스가 꺼져 있습니다  
설정에서 블루투스를 활성화 시켜주세요!  
설정 취소



<블루투스 버튼 클릭 시>

- 휴대폰 블루투스가 활성화 되어있을 때
- 블루투스 연결되어 있을 때
- 블루투스 연결 끊기 알림

- 블루투스 연결되어 있지 않을 때
- 블루투스 연결할 목록 보이기

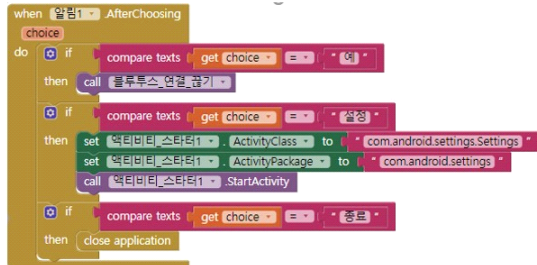
Bluetooth Interface  
98:DA:b0:01:97:E4 HC-06  
E8:7F:b8:48:E9:49 Galaxy Buds Pro (E949)

- 휴대폰 블루투스가 활성화 되어있지 않을 때
- 블루투스 활성화 요청 알림



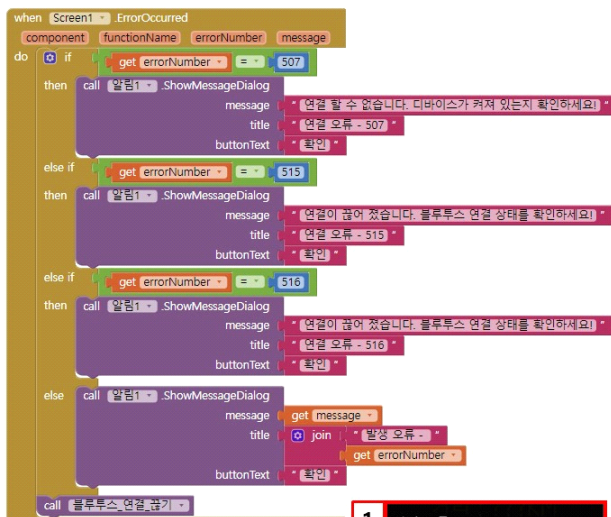
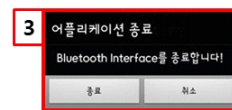
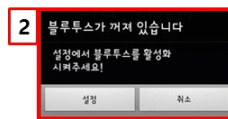
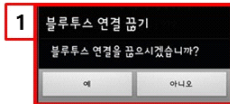
<화면에서 뒤로 가기 클릭 시>

- 앱 종료 알림



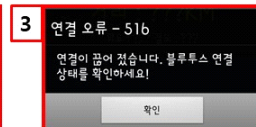
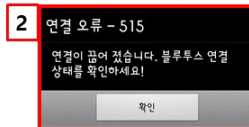
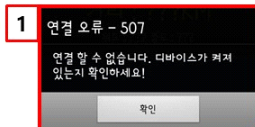
<알림 후 선택지 중 선택 시>

- 1번 그림에서 예를 클릭 시
  - 블루투스 연결 끊기
- 2번 그림에서 설정 클릭 시
  - 휴대폰 설정 열기
- 3번 그림에서 종료 클릭시
  - 앱 종료



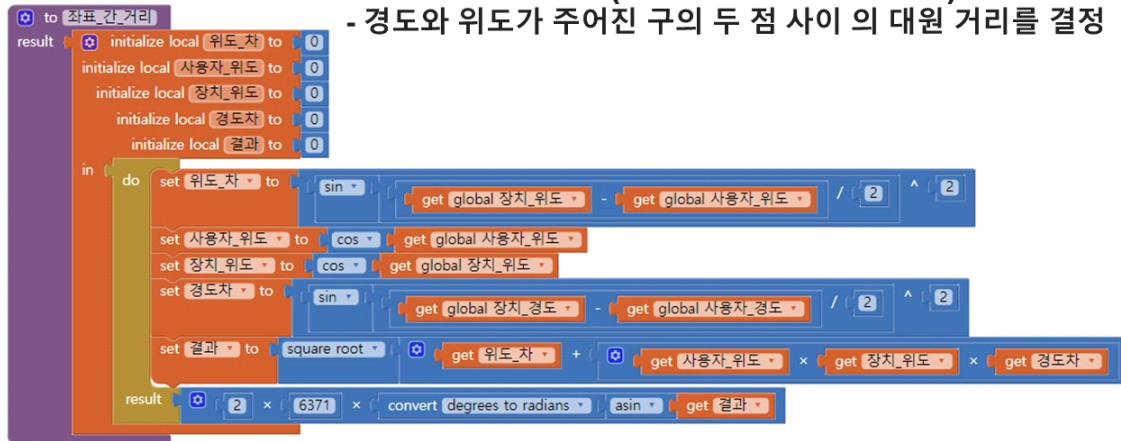
<오류 발생 시>

- 오류 번호 507번이면
  - 1번 그림과 같이 출력
- 오류 번호 515번이면
  - 2번 그림과 같이 출력
- 오류 번호 516번이면
  - 3번 그림과 같이 출력
- 설정하지 않은 오류 번호이면
  - 밑의 그림과는 같은 형식이지만 메시지가 다르게 출력
- 블루투스 연결 끊기 함수 실행



## 하버사인 공식 (Haversine Formula)

- 경도와 위도가 주어진 구의 두 점 사이의 대원 거리를 결정



<https://aia.bizadmin.co.kr/gps%EC%A2%8C%ED%91%9C%EA%B0%84%EC%9D%98-%EA%B1%B0%EB%A6%AC%EA%B5%AC%ED%95%98%EA%B8%B0/>

$$\text{hav}(\theta) = \underbrace{\text{hav}(\varphi_2 - \varphi_1)}_{\text{위도 차}} + \underbrace{\cos(\varphi_1)}_{\text{사용자 위도}} \underbrace{\cos(\varphi_2)}_{\text{장치 위도}} \underbrace{\text{hav}(\lambda_2 - \lambda_1)}_{\text{경도 차}}$$

하버 사인

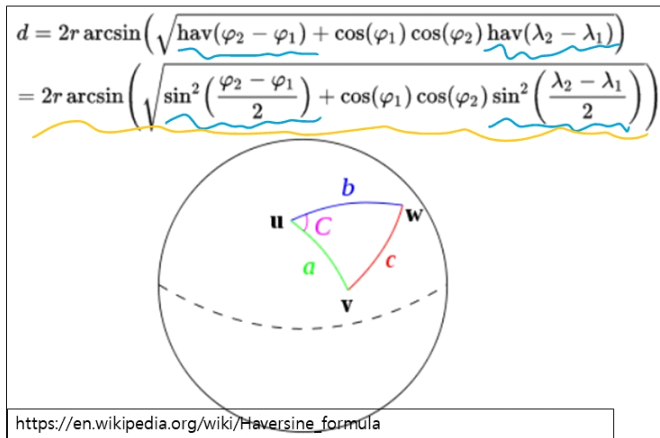
$$\text{hav}(\theta) = \sin^2\left(\frac{\theta}{2}\right) = \frac{1 - \cos(\theta)}{2}$$

$$d = r \text{ archav}(h) = 2r \arcsin(\sqrt{h})$$

$\varphi_1, \varphi_2$  (각 위치의 위도)

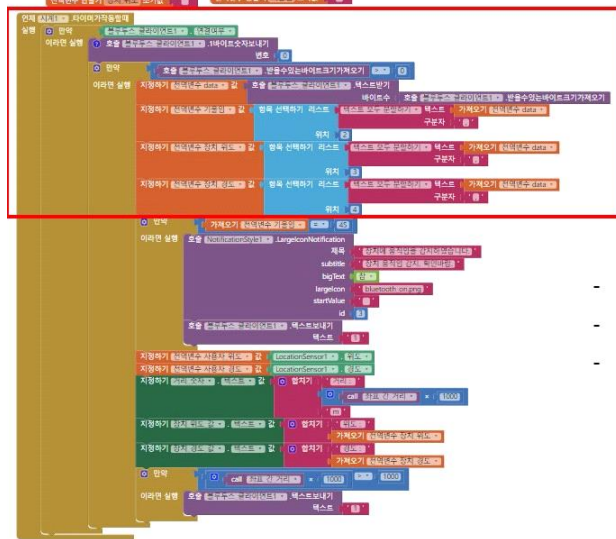
$\lambda_1, \lambda_2$  (각 위치의 경도)

$$\theta = \frac{d}{r} \quad (\text{중심각})$$





장치에서 받아온 값들과 핸드폰의 위치 값을 저장하기 위한 전역 변수 생성

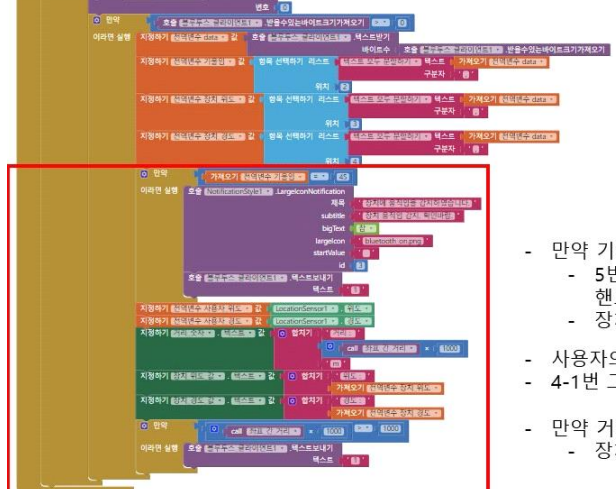


<입력 값 및 출력 값 갱신>

- 블루투스가 연결되어 있을 때
- 작동 확인을 위해 장치에 0의 신호를 보냄
- 블루투스가 장치와 통신하고 있다면
  - 장치에서 데이터를 받아 올
  - 기움임 = " "을 기반해서 받아온 데이터의 2번째 값
  - 위도 = " "을 기반해서 받아온 데이터의 3번째 값
  - 경도 = " "을 기반해서 받아온 데이터의 4번째 값

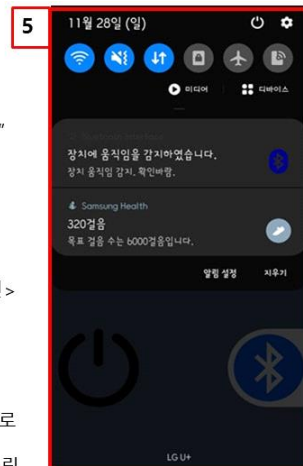
장치에서 받아온 값들과 핸드폰의 위치 값을 저장하기 위한 전역 변수 생성

<데이터 값>  
"Start,(기움임 값),(장치 위도 값),(장치 경도 값),Stop"



<입력 값 및 출력 값 갱신>

- 만약 기움임의 값 45라면
  - 5번 그림과 같이푸시 알림으로 핸드폰에 알림을 보냄
  - 장치에 "1"을 보내 부저를 울림
- 사용자의 위도, 경도 값을 갱신함
- 4-1번 그림과 같이 거리 값과 위도, 경도를 표시함
- 만약 거리 차이가 5m가 넘어가면
  - 장치에 "1"을 보내 부저를 울림



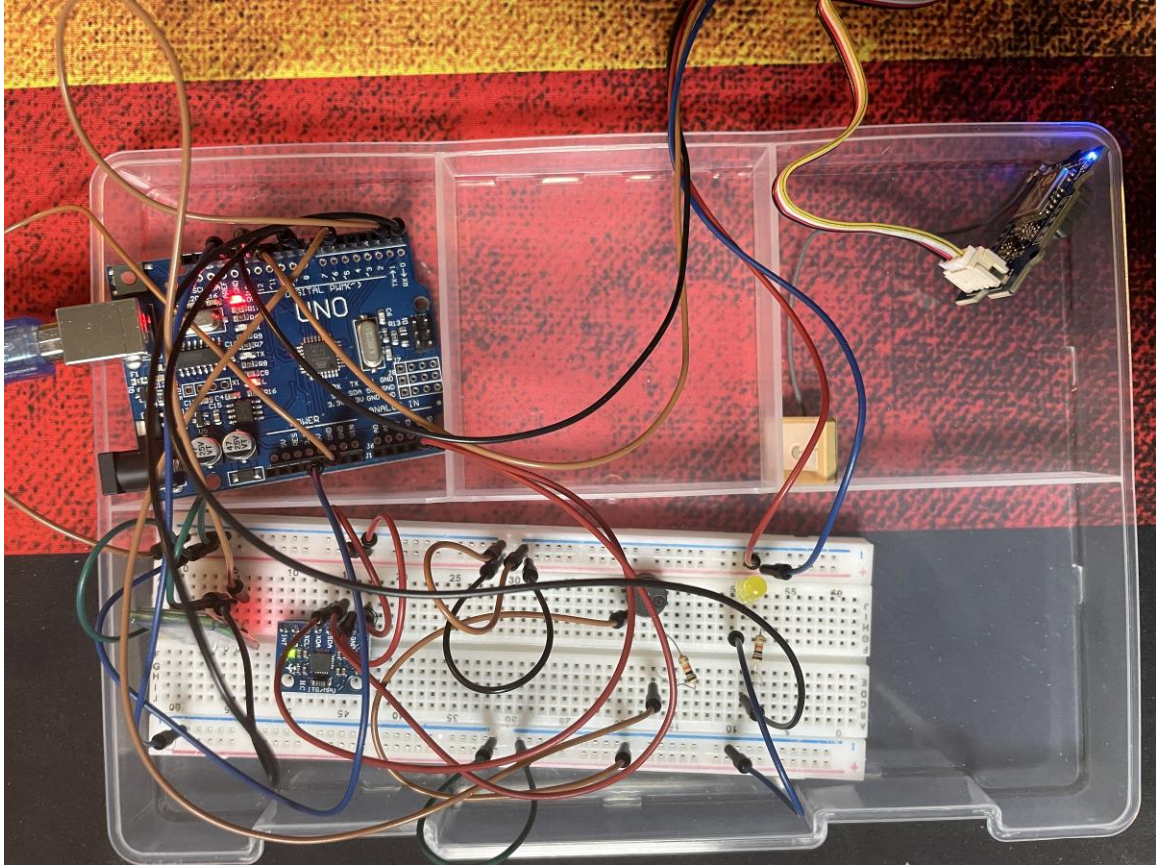
## 구현 도구

아두이노 회로에서 측정한 정보를 휴대폰에서 확인할 수 있고, 거리계산 알고리즘 및 회로 원격 조작을 구현하기 위해 앱 제작툴인 앱인벤터를 사용하였다. 이외에 아두이노 회로 코딩을 위해 아두이노 사이트에서 제공하는 아두이노 ide를 사용하였다.

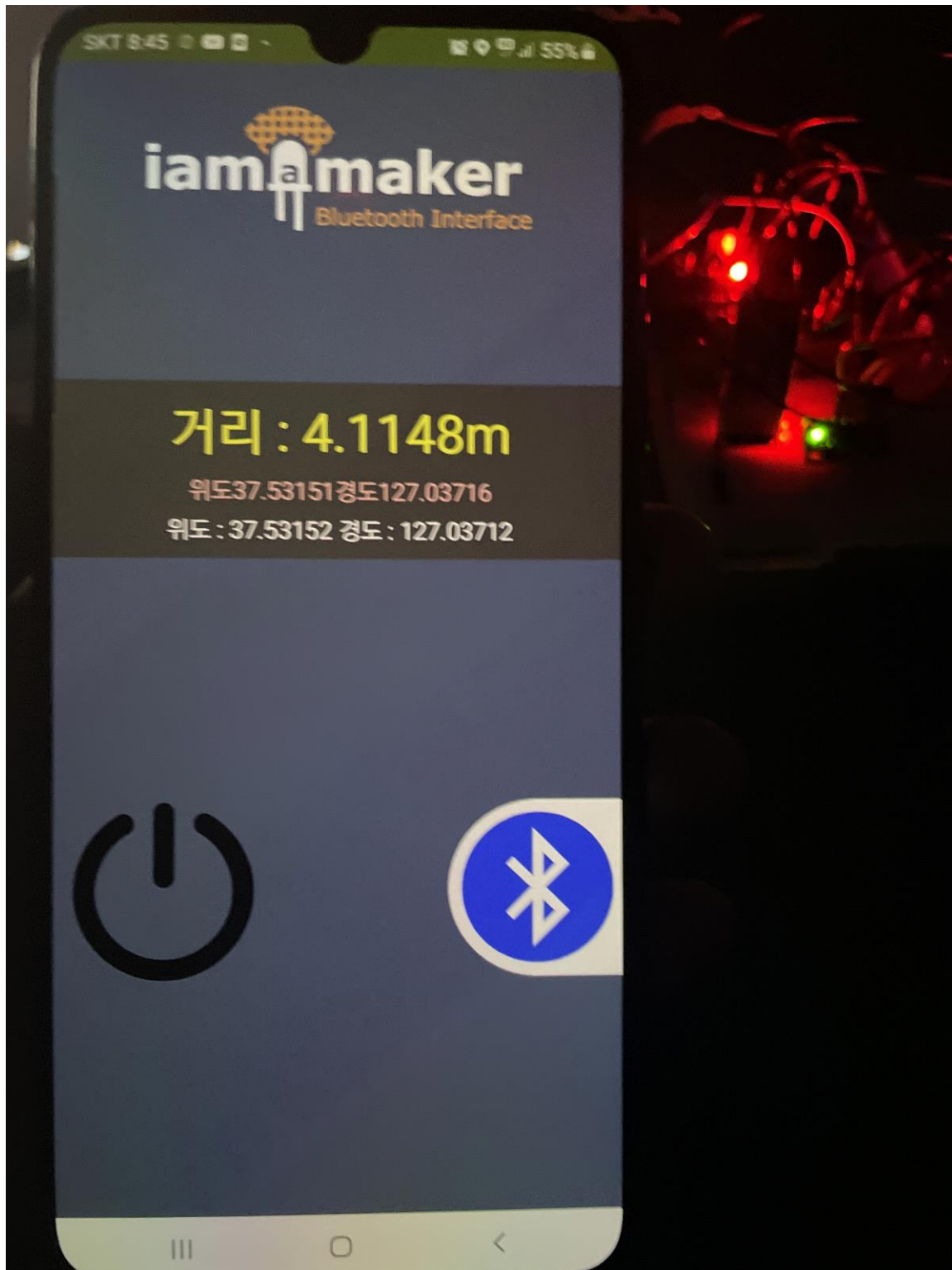
## 제 6 장 결과



- 결과물 설명



아두이노 보드 하나, 세 센서들, 부저, LED로 구성된 회로입니다.



앱을 열면 사용자 및 회로의 위도/경도 정보가 출력됩니다. 이를 바탕으로 앱에서 두 사이의 거리를 계산하여 거리 값을 출력한다. 버튼을 누르면 회로의 부저와 LED를 조작할 수 있다. 블루투스 색깔이 파란색이면 페어링 되었다는 신호이다.

데모

영상

링크:

<https://drive.google.com/file/d/1IBZAYiH9ThCrvPADYjYuodratZgUoY4l/view?usp=sharing>

데모영상을 통해 구현된 결과물을 쉽게 보실 수 있습니다.

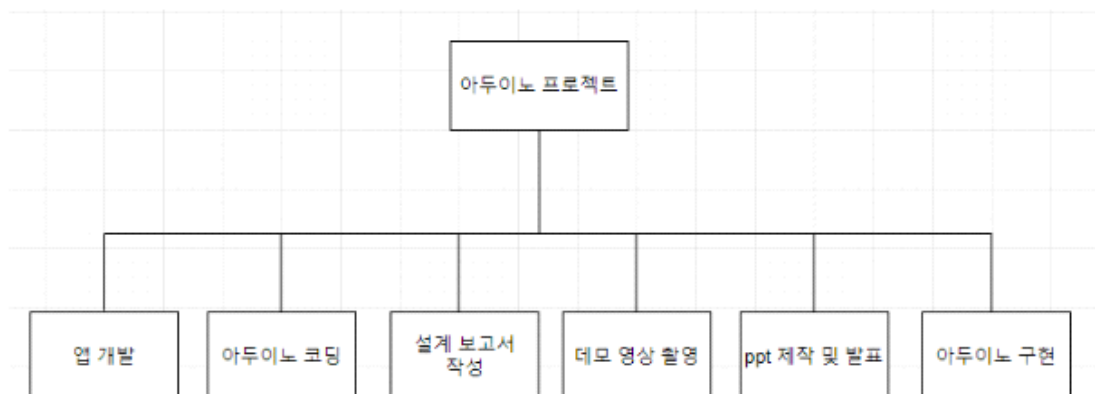
- **결과물 분석**

결과물을 4.3 절에서 제시한 평가 항목과 평가 방법에 맞추어 평가하고 평가 결과를 분석한다.

## 제 7 장 작업진행 방법

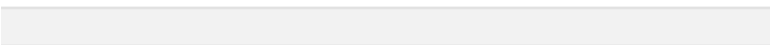
- **작업 분담 구조**

전체적인 업무는 '어플리케이션 개발', '아두이노 코딩', '설계 보고서 작성', '데모 영상 촬영', 'ppt제작 및 발표', '아두이노 구현' 으로 나누었다.



- **설계 일정 및 역할 분담**

No	업무이름	시작	완료	기간
1	아두이노 코딩	11.07	11.17	10d



2	앱 개발	11.17	11.29	12d
3	아두이노 구현	11.26	11.29	4d
4	데모 영상 촬영	11.26	11.29	4d
5	ppt 제작 및 발표	11.26	11.29	4d
6	설계 보고서 작성	11.26	11.29	4d

#### 역할분담

남상원 : 아두이노 코딩, 어플리케이션 개발, 아두이노 구현  
이승지 : 설계 보고서 작성, ppt 제작 및 발표, 아두이노 구현  
이우제 : 설계 보고서 작성, 데모 영상 촬영, 아두이노 구현  
변찬현 : 설계 보고서 작성, 아두이노 구현

#### • 예산집행내역

Grove - GPS (Air530) [109020022] 22,000원

<https://www.devicemart.co.kr/goods/view?no=12546919>

5V 능동 DC 부저 [FQ-014] 500원

<https://www.devicemart.co.kr/goods/view?no=1361184>

SW-18010P 진동센서모듈 [SZH-EK023] 1,200원

<https://www.devicemart.co.kr/goods/view?no=1278061>

9V건전지 배터리홀더 전선형[1개입] 500원

<https://www.devicemart.co.kr/goods/view?no=3116>

MN1604-BP(9V 6LF22) 3,500원

<https://www.devicemart.co.kr/goods/view?no=2756>

HM-10 블루투스 4.0 BLE 모듈 [SZH-EK108] 12,000원

<https://www.devicemart.co.kr/goods/view?no=1384572>

## 제 8 장 결론

### • 결론

바깥에 자전거, 키보드와 같은 물건을 두었을 때 도난당한 경험이 있을 것이다. 해당 아두이노 프로젝트는 이러한 물건들의 도난 방지를 위하여 설계된 프로젝트이다. 어떻게 물건의 도난을 방지하면 좋을지 생각하던 중 물건을 훔치는 과정에서 물건에 기울기가 발생할 것이기 때문에 기울기 센서를 이용하기로 하였고 단순히 기울어짐만으로는 도난 당했다는 것을 확신할 수 없기 때문에 GPS센서를 추가하여 물건이 일정거리이상 멀어지면 도난 당했다고 판단을 하는 아두이노 프로젝트를 구상하게 된 것이다. 이 과정에서 GPS센서의 데이터를 밖에 있는 물건과 송수신 하기 위해 블루투스 센서도 활용하게 되었다. 그리고 도난이 발생했다고 판단되면 LED와 부저가 울리며 스마트폰의 어플리케이션으로 이를 제어할 수 있었다. 아두이노 우노의 소프트웨어 시리얼은 센서 1개만이 사용가능하기 때문에 블루투스 센서와 GPS센서 둘 다 송수신 하기 위해서 업로드 센서인 하드웨어 시리얼을 블루투스 센서의 시리얼로 사용하였고 이 구현 과정에서 오류가 많이 발생하여 수정 작업을 거쳤다. 하지만 하드웨어 시리얼과 소프트웨어 시리얼의 사용법을 배우고 활용하여 구현에 성공하였다.

## -기대효과

해당 아두이노 프로젝트로 바깥에 둘 수있는 자전거와 키펠드에 설치한다면 도난율이 현저히 적어질 것이다. LED와 부저로 경고를 할 수 있고 물건이 도난당했다는 사실을 빨리 알아차려 신고하면 범인을 잡는 것이 더 원활하게 진행될 것이다. 하지만 블루투스 센서와 스마트폰의 연동과정에서 배터리의 문제가 존재하며 어떻게 이것을 보완해야 하는가가 이 프로젝트의 가장 큰 문제점인 것 같다. 그래도 위 프로젝트로 사회적으로 물건의 도난율이 적어질 것이고 바깥에 물건을 두어도 안전한 세상이 오는 효과를 기대할 수 있겠다.

## 제 9 장 참고문헌

<https://steemit.com/kr-arduino/@codingman/mpu-6050>

<https://codeda.tistory.com/116>

<https://m.blog.naver.com/PostView.naver?isHttpsRedirect=true&blogId=roboholic84&logNo=220401407348>

- mpu-6050 참고 웹 사이트

<http://www.iamamaker.kr/ko/projects/%EC%95%B1-%EC%9D%B8%EB%B2%A4%ED%84%B0-%EB%B8%94%EB%A3%A8%ED%88%AC%EC%8A%A4%EB%A5%BC-%EC%9E%90%EB%8F%99%EC%9C%BC%EB%A1%9C-%EC%97%B0%EA%B2%B0%ED%95%98%EC%9E%90/>

- 애플벤더 블루투스 연결 오픈소스 사용 및 참고

<https://kyoungin90.tistory.com/280>

- I2C통신 참고 Wire.h라이브러리 참고

<https://aia.bizadmin.co.kr/gps-%EC%A2%8C%ED%91%9C%EA%B0%84%EC%9D%98-%EA%B1%B0%EB%A6%AC-%EA%B5%AC%ED%95%98%EA%B8%B0/>

[https://en.wikipedia.org/wiki/Haversine\\_formula](https://en.wikipedia.org/wiki/Haversine_formula)

- 두 좌표간의 gps 거리 값 구하기

<https://lamlic36.tistory.com/3>

- 좌표 값 받아오기

<https://feel0804.tistory.com/3>

- TinyGPSPlus라이브러리 참고