

# 자전거 도난 방지를 위한 장치

어드벤처디자인\_01

6조

컴퓨터공학과 이우제(팀장)  
컴퓨터공학과 남상원  
컴퓨터공학과 이승지(발표자)  
수학과 변찬현

# 목차

1. 주제선정

2. 프로젝트 목표

3. 기능 구현방법

4. 데모영상

# 주제 선정

- 아두이노를 사용하여 적절한 구현이 되어야 함
- 일어나고 있는 문제점을 개선하고 싶음

⇒ 분실 자전거의 수가 늘어나고 있는 상황에 대한  
적절한 방법 필요

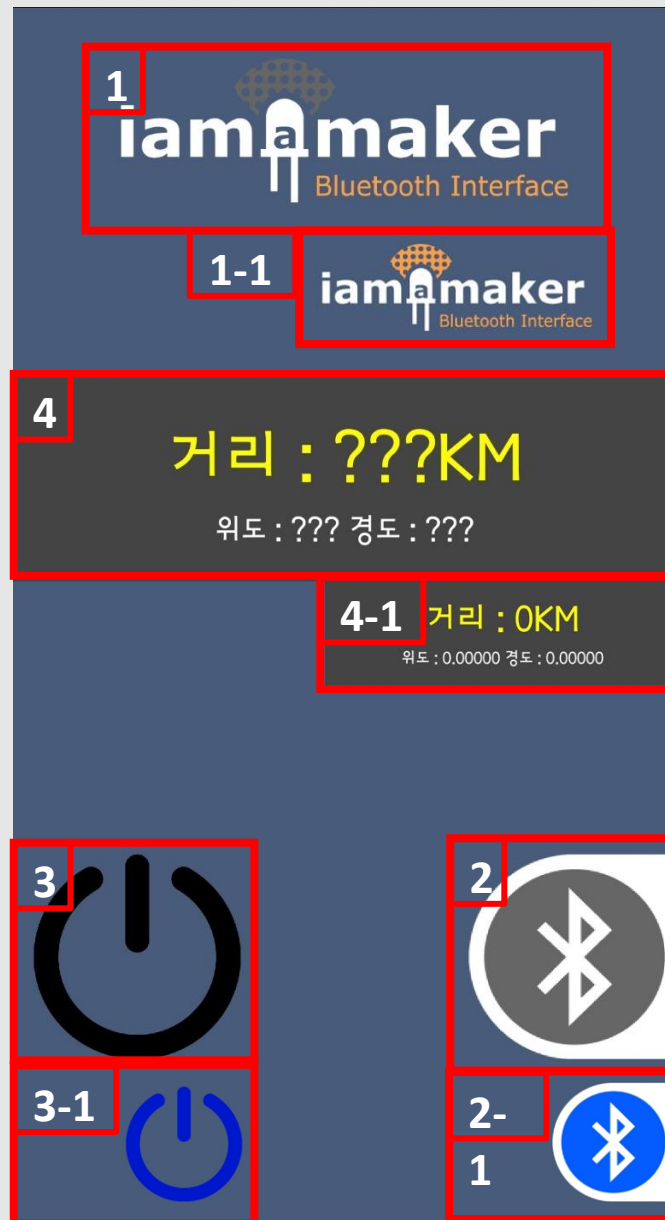
# 프로젝트 목표

- 선행조사 결과  
=> 자물쇠를 끊고 도난하는 경우가 있어 실질적인 문제 해결 불가능
- 위도, 경도, 거리 값을 통해 물건의 거리를 대략 알 수 있음
- 장치에 신호를 보내어 위치를 찾기 쉽게 함
- 움직임이 감지되거나 장치와 사용자 간의 거리가 멀어지면  
사용자에게 알림을 보냄

# 기능 구현 방법

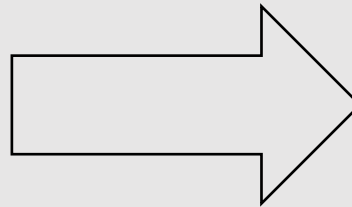
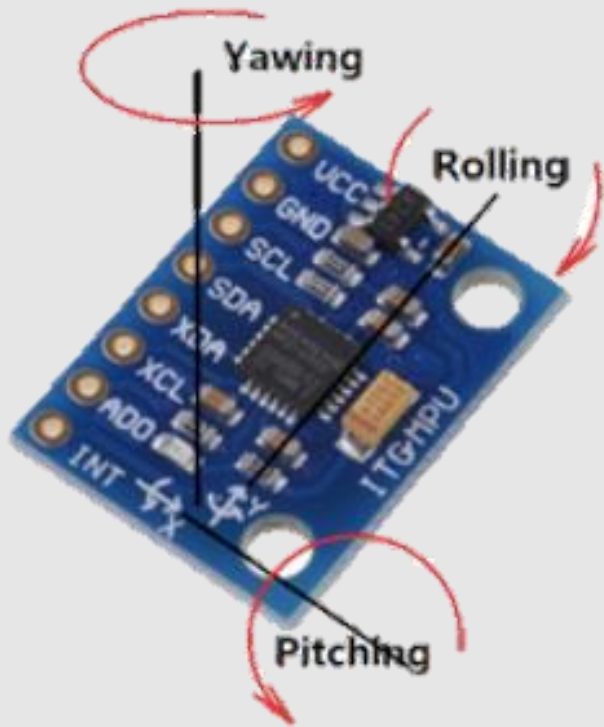
## 1. 휴대폰에 있는 앱을 통한 휴대폰과 물건의 블루투스 연결



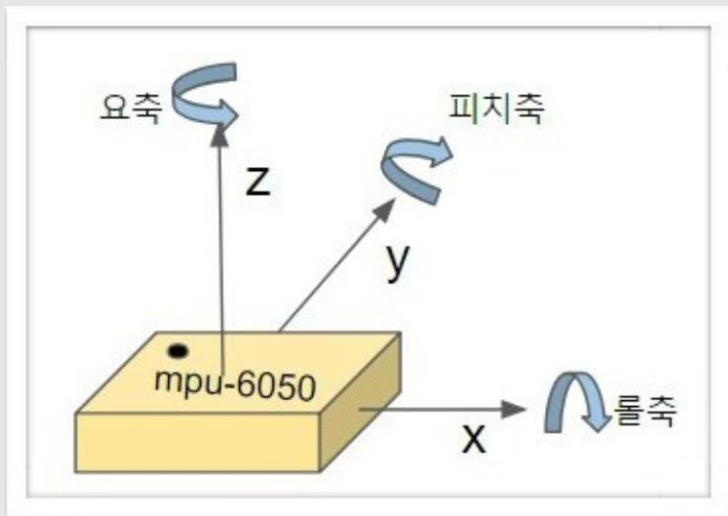


# 기능 구현 방법

2. 가속도와 각속도를 통해 값을 구해 기울기 값을 구하여 기울임을 감지



# MPU6050 구동원리



- 가속도(x,y,z), 자이로(각속도)(x,y,z)와 온도 값을 측정할 수 있는 센서
- 상보필터를 통해 비율을 설정해 가속도를 통한 각도와 자이로(각속도)를 통한 각도의 장점을 취함
- 이 코드에 적용된 상보 필터 식 :  
$$\text{angle} = 0.98 * (\text{angle} + \text{자이로값} * dt) + 0.02 * (\text{가속도값})$$
  
(angle:출력각, dt:적분할 시간)

$$\begin{aligned}\text{피치 } (\rho) &= \text{atan}\left(\frac{A_X}{\sqrt{A_Y^2 + A_Z^2}}\right) \\ \text{롤 } (\phi) &= \text{atan}\left(\frac{A_Y}{\sqrt{A_X^2 + A_Z^2}}\right) \\ \text{요 } (\theta) &= \text{atan}\left(\frac{\sqrt{A_X^2 + A_Y^2}}{A_Z}\right)\end{aligned}$$

- 자이로 센서: '각속도'를 측정하는 센서
- 특정 축을 기준으로 회전하는 속도를 나타냄
- 자이로 값을 적분하면 **각도**를 얻을 수 있음
- 초기값을 0이라고 했을 경우 각 방향으로 얼마나 움직였는지를 알면 초기값과 비교하여 현재 각도를 알 수 있음



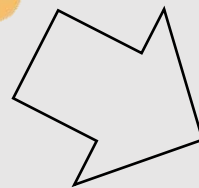
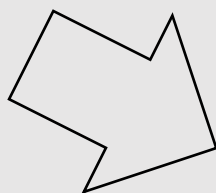
# 기능 구현 방법

3.

장치가 기울어지면  
사용자에게 신호를 보냄

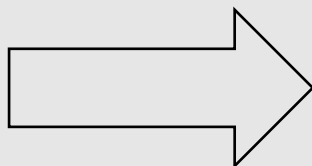
GPS를 통해 위도값과 경도값을 구해  
사용자와 장치 간의 거리를 출력해줌

장치가(100m)멀어지면  
핸드폰에 알람을 울림



# 기능 구현 방법

4. 자전거의 위치를 쉽게 찾기 위해 휴대폰 버튼을 통해 물건에 부착된 부저를 울릴 수 있음



# 1. 위치 찾는 코드

- 아두이노

```
#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
```

```
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
```

# 1. 기울임 값과 위도, 경도 데이터 출력

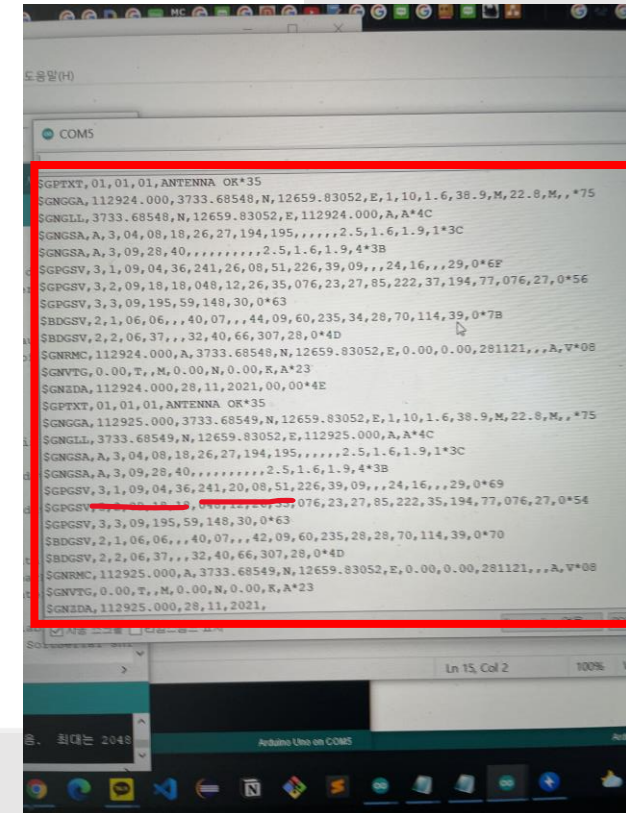
```
// 각 각도가 -45도나 45도를 넘어가면 움직임 값 조정
if ( angle_x > 45 || angle_x < -45 || angle_y > 45 || angle_y < -45 || angle_z > 45 || angle_z < -45 )
    output[0] = 45;
else
    output[0] = 0;

// 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)+"+"Stop";

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



<출력할 데이터의 형태>

“Start,(기울기 값),(장치 위도 값),(장치 경도 값),Stop”

# 1. 데이터 받는 코드

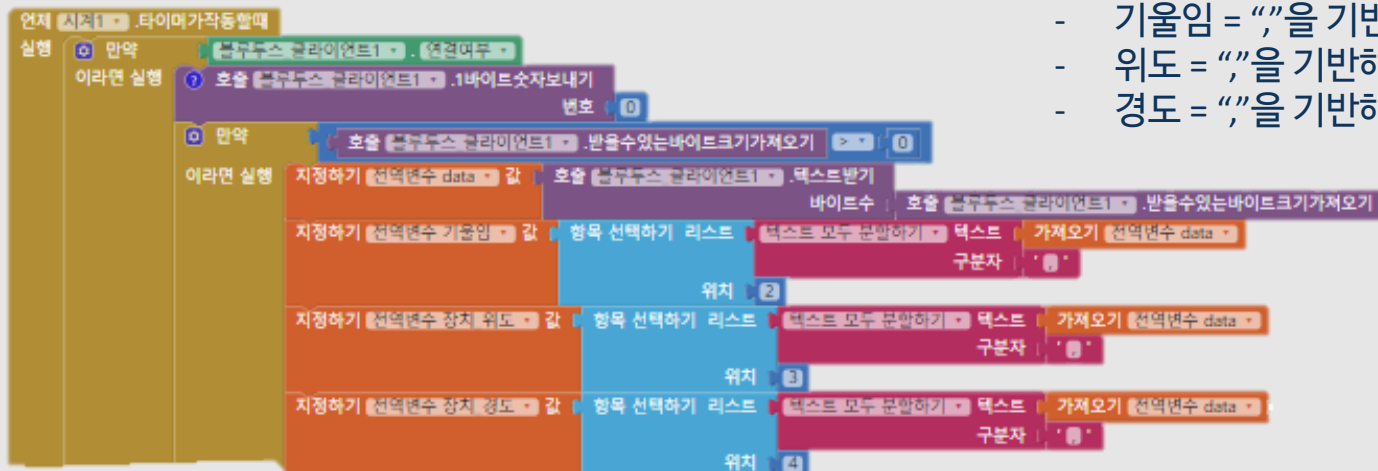
## • 앱

장치에서 받아온 값들과  
핸드폰의 위치 값을 저장하기  
위한 전역 변수 생성



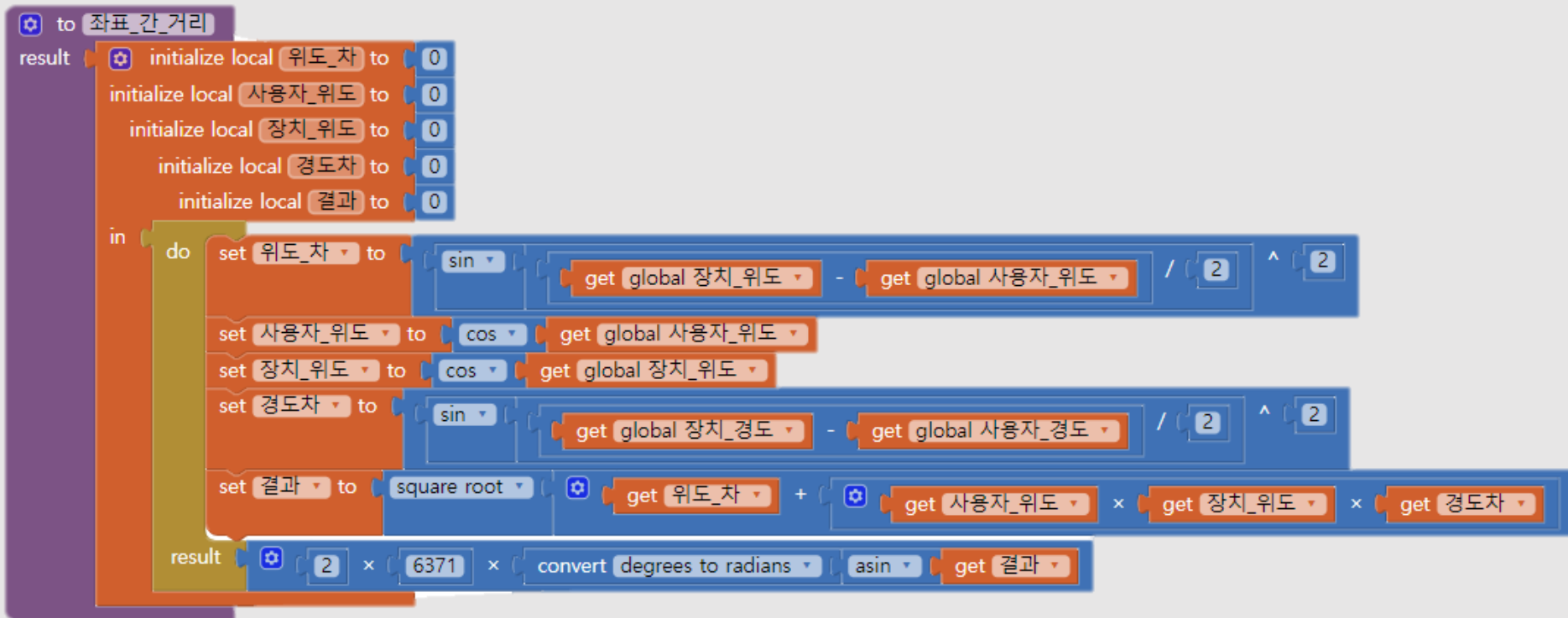
<입력 값 및 출력 값 갱신>

- 블루투스가 연결되어 있을 때
- 작동 확인을 위해 장치에 0의 신호를 보냄
- 블루투스가 장치와 통신하고 있다면
  - 장치에서 데이터를 받아옴
  - 기울임 = “”을 기반으로 받아온 데이터의 2번째 값
  - 위도 = “”을 기반으로 받아온 데이터의 3번째 값
  - 경도 = “”을 기반으로 받아온 데이터의 4번째 값



# 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/>

# 하버사인 공식 (Haversine Formula)

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

하버 사인

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

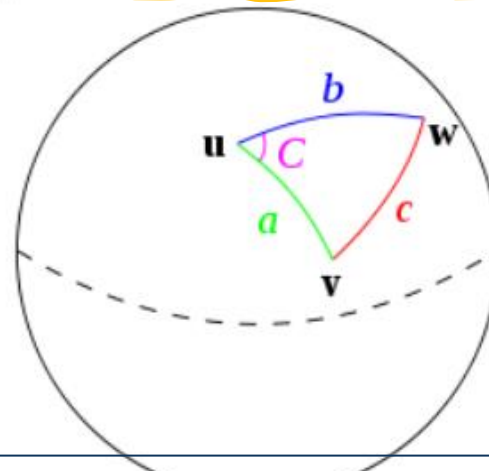
$$d = r \text{ archav}(h) = 2r \arcsin(\sqrt{h})$$

$\varphi_1, \varphi_2$  (각 위치의 위도)

$\lambda_1, \lambda_2$  (각 위치의 경도)

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

$$\begin{aligned} d &= 2r \arcsin\left(\sqrt{\text{hav}(\varphi_2 - \varphi_1) + \cos(\varphi_1) \cos(\varphi_2) \text{hav}(\lambda_2 - \lambda_1)}\right) \\ &= 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\varphi_2 - \varphi_1}{2}\right) + \cos(\varphi_1) \cos(\varphi_2) \sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right) \end{aligned}$$



## 2. 멀어지거나 기울어지면 부저 울리는 코드

- 아두이노

```
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
    }
}
```



## 2. 멀어지거나 기울어지면 부저 울리는 코드

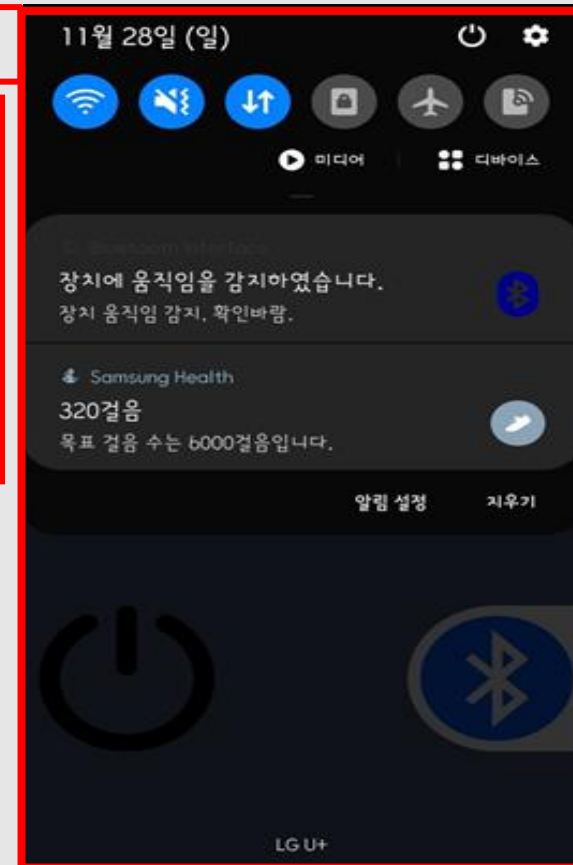
### • 앱

5

만약 가져오기 전역변수 기울임 = 45  
이러면 실행 호출 NotificationStyle1 .LargeIconNotification  
제목 '장치에 움직임을 감지하였습니다.'  
subtitle '장치 움직임 감지. 확인바람.'  
bigText 참  
largeIcon bluetooth on.png  
startValue  
id 3  
호출 블루투스 클라이언트1 .텍스트보내기  
텍스트 '1'

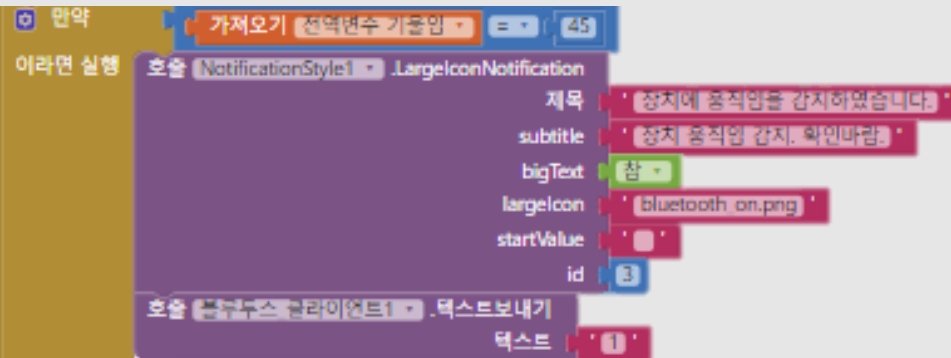
지정하기 전역변수 사용자 위도 값 LocationSensor1 . 위도  
지정하기 전역변수 사용자 경도 값 LocationSensor1 . 경도  
지정하기 거리 숫자 . 텍스트 값 합치기 거리  
call 차프 간 거리 \* 1000  
m  
지정하기 장치 위도 값 . 텍스트 값 합치기 위도  
가져오기 전역변수 장치 위도  
지정하기 장치 경도 값 . 텍스트 값 합치기 경도  
가져오기 전역변수 장치 경도

만약 call 차프 간 거리 \* 1000 > 100  
이러면 실행 호출 블루투스 클라이언트1 .텍스트보내기  
텍스트 '1'



## 2. 멀어지거나 기울어지면 부저 울리는 코드

### • 앱



- 사용자의 위도, 경도 값을 갱신함
- 4-1번 그림과 같이 받아온 거리 값과 위도, 경도를 표시함

# 데모영상