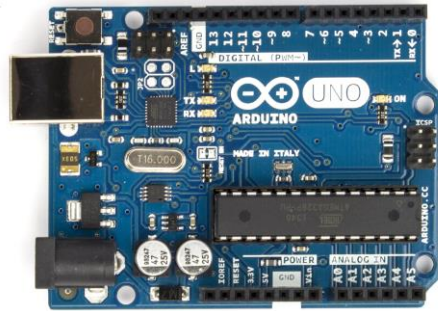


넘어짐 감지 신발

작품 개요 설계 및 구동방법

개념 설계 내용



자이로센서
부저
압력센서
블루투스 측정 및 작동

넘어짐 여부를 휴대폰에 전송



블루투스를 통해 넘어짐 여부를 받아옴.



현재 위치를 지도에 출력
넘어진 상태면 경고 알림 출력

작품 외관

USB 케이블
전원 공급 및
시리얼 값(회전 각도) 확인

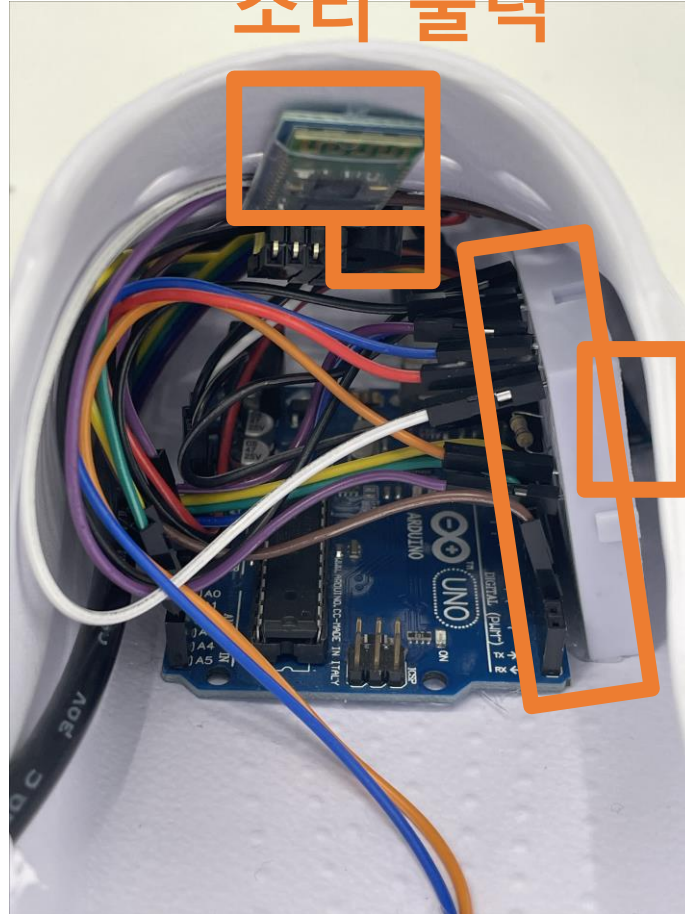
아두이노
여러 센서 감지

FSR(압력)
감지 센서
신발 착용여부



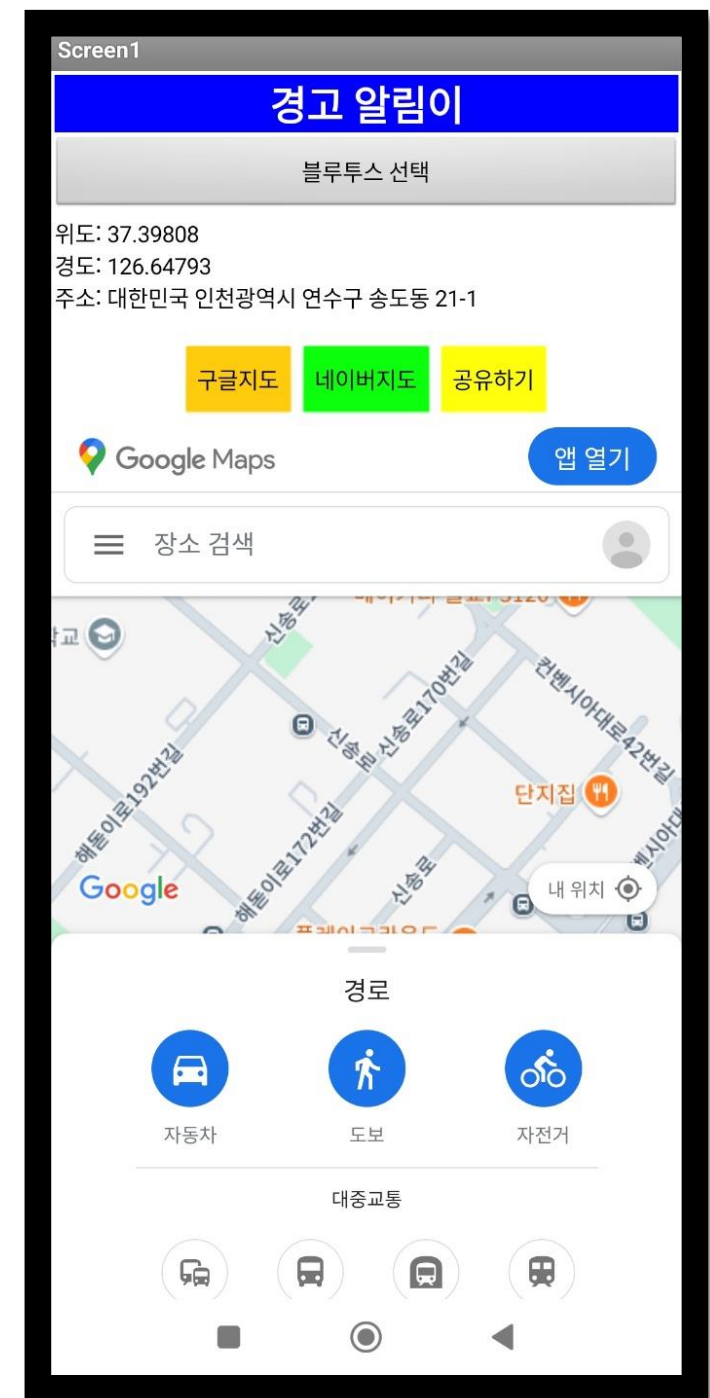
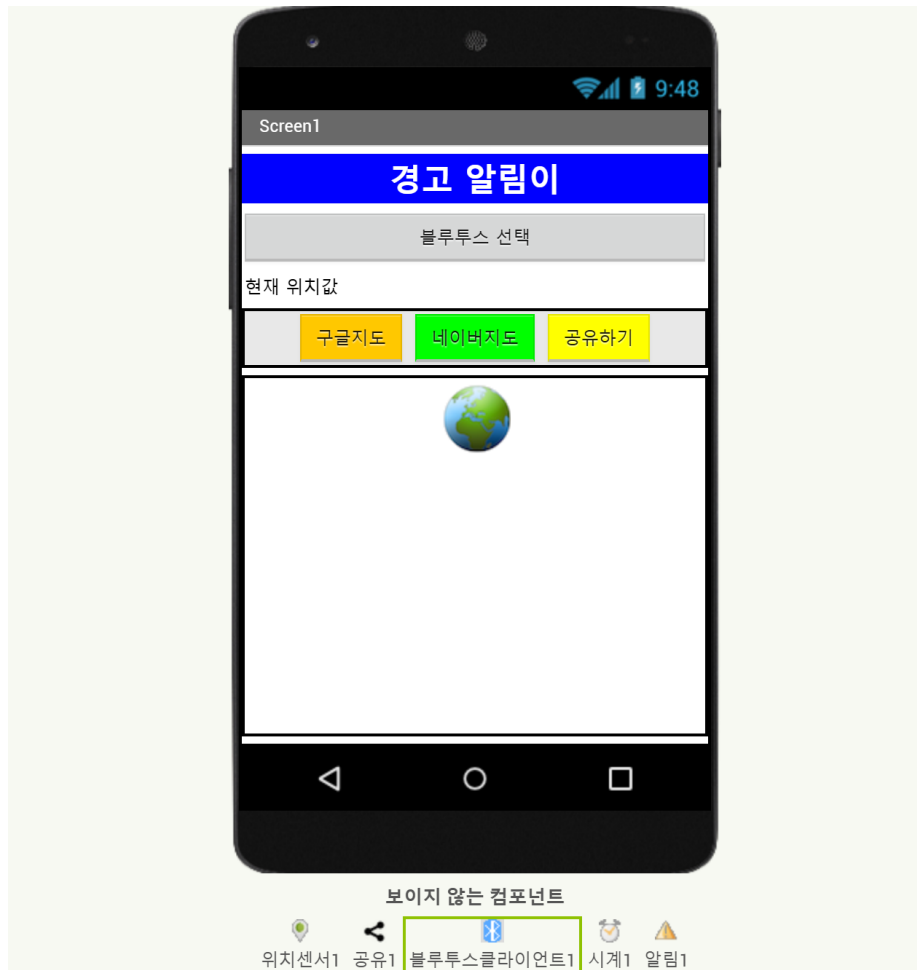
작품 내부

블루투스&부저
휴대폰 정보 전달
소리 출력

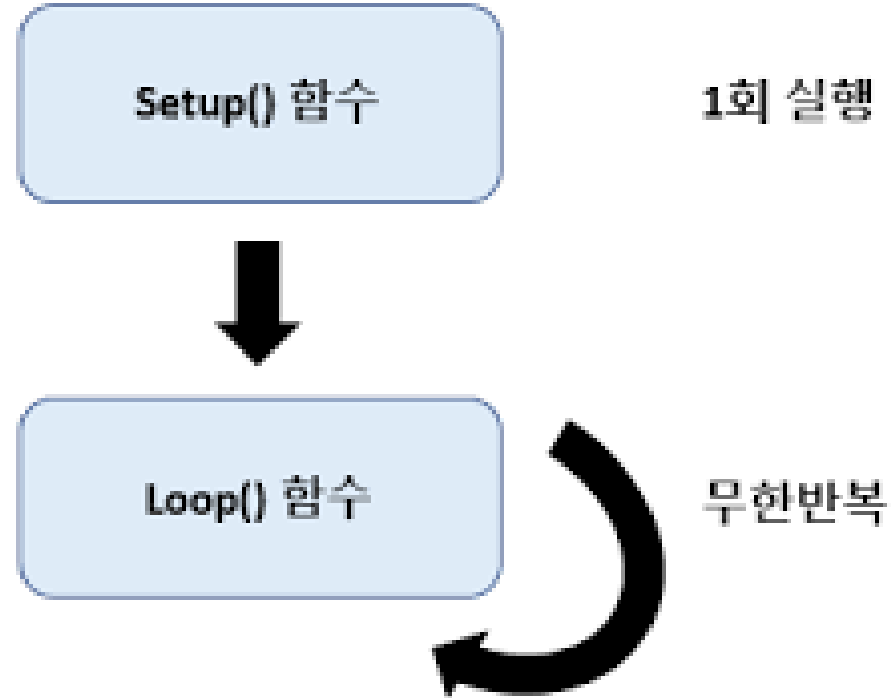


브레드보드&자이로 센서
센서 연결
각도 측정

어플화면 디자인



아두이노 작동 순서 과정



코드 전체 코드 아두이노

```
#include <SoftwareSerial.h>
#include <Wire.h>
#include <MPU6050.h>

#define BT_RXD 3
#define BT_TXD 4
SoftwareSerial hc06(BT_RXD, BT_TXD);

MPU6050 mpu;

const int buzzerPin = 8; // 부저 핀
long lastTime = 0; // 시간 체크를 위한 변수
bool buzzerState = false; // 부저 상태

int FSRsensor = A0; // 센서값을 아날로그 A0핀 설정
int value = 0;

void setup() {
  // 시리얼 모니터 시작
  Serial.begin(9600);
  pinMode(buzzerPin, OUTPUT);

  // hc06
  hc06.begin(9600);

  // MPU6050 초기화
  Wire.begin();
  mpu.initialize();

  // 센서 캘리브레이션
  Serial.println("MPU6050 캘리브레이션 중...");
  mpu.CalibrateGyro(6); // 자이로 캘리브레이션
  mpu.CalibrateAccel(6); // 가속도 캘리브레이션
  Serial.println("캘리브레이션 완료.");
  tone(buzzerPin, 523, 300);
}
```

```
void loop() {
  value = analogRead(FSRsensor); // 아날로그 센서값을 value 변수에 저장
  value = map(value, 0, 1023, 0, 255); // value에 아날로그 센서값(0~1023)을 PWM 조절을위해 값 0~255로 변환
  Serial.println(value);
  // MPU6050에서 가속도 데이터를 읽음

  int16_t ax, ay, az;
  mpu.getAcceleration(&ax, &ay, &az);

  // 가속도 값을 각도로 변환 (tilt angle 계산)
  float angle = atan2(ay, az) * 180.0 / PI;

  // 각도 출력 (디버깅용)
  Serial.print("Angle: ");
  Serial.println(angle);

  // 자이로 센서에서 구해진 각도가 15도 이상이거나, -15도 이하일 때
  if (angle >= 15 || angle <= -15) {
    if (millis() - lastTime >= 3000 && value >= 128) { // 3초간 유지되었을 때
      if (!buzzerState) {
        tone(buzzerPin, 700, 1000);
        delay(500);
        // noTone(buzzerState);
        buzzerState = true;
        Serial.println("부저 울림");
        hc06.write("1"); // 넘어짐 감지되어 상황을 휴대전화로 전송
        lastTime = millis(); // 3초 유지 여부 파악
      }
    } else {
      digitalWrite(buzzerPin, LOW); // 부저 끄기
      buzzerState = false;
      lastTime = millis(); // 90도 상태를 벗어나면 시간 초기화
    }
  }

  // 블루투스기 전송된 값이 정상적으로 받아왔을 때
  if (hc06.available())
    Serial.write(hc06.read()); // 시리얼 모니터에 전송 받은 값 출력
  // 시리얼 모니터에 입력한 값이 전송된 경우
  if (Serial.available())
    hc06.write(Serial.read()); // 블루투스에게 입력한 값을 전송함.

  delay(100); // 루프 속도 제어
}
```

아두이노 소스코드 준비코드

```
#include <SoftwareSerial.h> // 블루투스 HC06 작동하기 위한 라이브러리
#include <Wire.h> // MPU6050 통신 위함
#include <MPU6050.h> // 자이로센서 MPU6050 작동에 필요한 코드(각도 측정)

// 블루투스 통신 핀 설정, 블루투스 초기화
#define BT_RXD 3
#define BT_TXD 4
SoftwareSerial hc06(BT_RXD, BT_TXD);

// 자이로센서 객체 생성
MPU6050 mpu;

const int buzzerPin = 8; // 부저 핀
long lastTime = 0; // 시간 체크를 위한 변수
bool buzzerState = false; // 부저활성여부

int FSRsensor = A0; // 압력센서 핀
int value = 0; // 압력센서 압력 수치 저장 변수
```


아두이노 소스코드 SETUP 함수

```
void setup() {  
  // 시리얼 모니터 데이터 통신 허용 및 속도 지정  
  Serial.begin(9600);  
  // 부저 디지털 8핀 출력용 선언  
  pinMode(buzzerPin, OUTPUT);  
  
  // 블루투스 통신허용 및 초기화  
  hc06.begin(9600);  
  
  // MPU6050 초기화  
  Wire.begin();  
  mpu.initialize();  
  
  // 자이로센서 캘리브레이션  
  Serial.println("MPU6050 캘리브레이션  
중");  
  mpu.CalibrateGyro(6); // 자이로 캘리브레이션  
  mpu.CalibrateAccel(6); // 가속도 캘리브레이션  
  Serial.println("캘리브레이션 완료.");  
  tone(buzzerPin, 523, 300);  
  // 센터 준비 모두 다하면 처음에 부저 소리출력  
}
```

아두이노 소스코드 LOOP 함수

```
void loop() {
    value = analogRead(FSRsensor);    // 아날로그 센서값을 value 변수에 저장
    value = map(value, 0, 1023, 0, 255);
    // value에 아날로그 센서값(0~1023)을 PWM 조절을 위해 값 0~255로 변환

    Serial.println(value);
    // MPU6050에서 가속도 데이터를 읽음

    // 가속도 저장 및 가속도 계산
    int16_t ax, ay, az;
    mpu.getAcceleration(&ax, &ay, &az);

    // 가속도 값을 각도로 변환 (tilt angle 계산)
    float angle = atan2(ay, az) * 180.0 / PI;

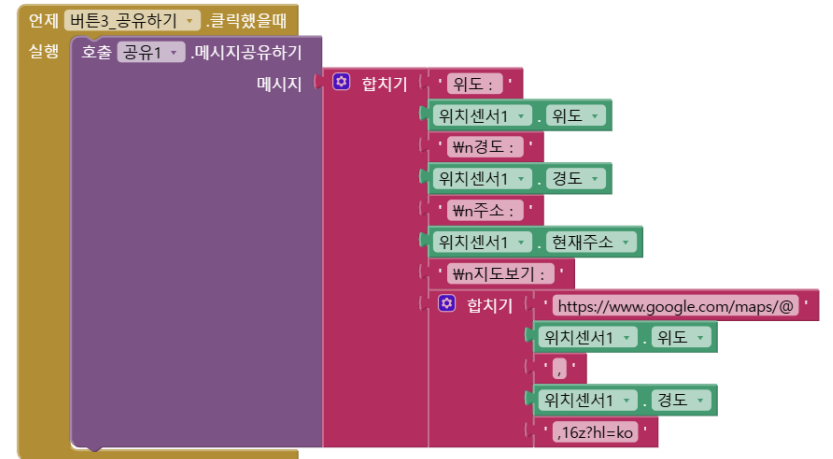
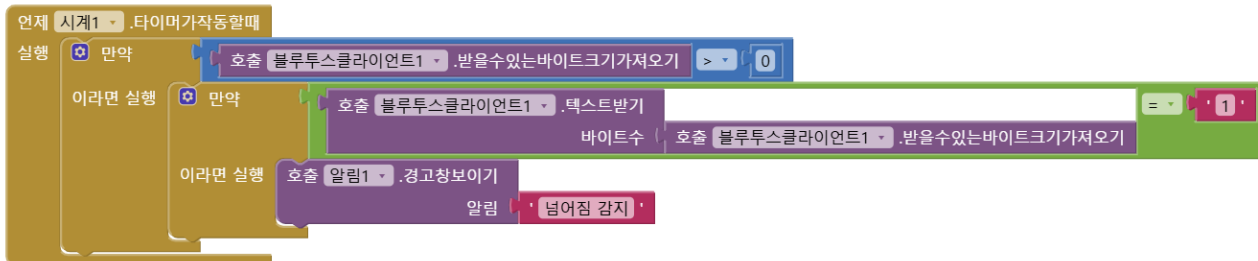
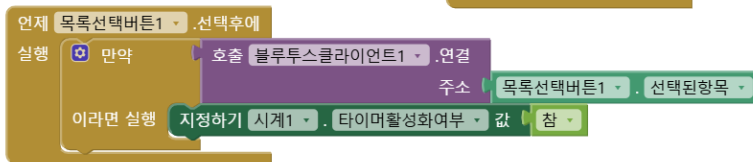
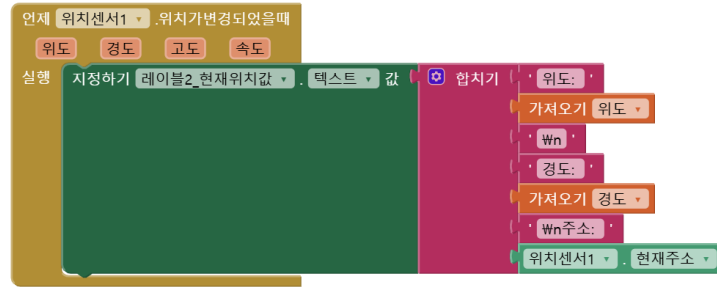
    // 각도 출력 (디버깅용)
    Serial.print("Angle: ");
    Serial.println(angle);

    // 자이로 센서에서 구해진 각도가 15도 이상이거나, -15도 이하일 때
    if (angle >= 15 || angle <= -15) {
        if (millis() - lastTime >= 3000 && value >= 128) {
            // 넘어짐 걸린시간이 3초간 유지되었을 때
            if (!buzzerState) {
                // 부저가 소리 안내했을 때, 1초간 소리 출력
                tone(buzzerPin, 700, 1000);
                delay(500); // 0.5초 딜레이
                buzzerState = true;
                Serial.println("부저 울림");
                hc06.write("1"); // 넘어짐 감지되어 상황을 휴대폰으로 전송
                lastTime = millis(); // 넘어진 상태 마지막 시간 저장
            }
        }
    } else { // 넘어진 경우가 아니라면(신발이 정상적으로 바닥에 안착될 때)
        digitalWrite(buzzerPin, LOW); // 부저 끄기
        buzzerState = false;
        lastTime = millis(); // 정상시 상태 마지막 시간 저장
    }

    // 블루투스가 전송된 값이 정상적으로 받아왔을 때
    if (hc06.available())
        Serial.write(hc06.read()); // 시리얼 모니터에 전송 받은 값 출력
    // 시리얼 모니터에 입력한 값이 전송된 경우
    if (Serial.available()) {
        hc06.write(Serial.read()); // 블루투스에게 입력한 값을 전송함.

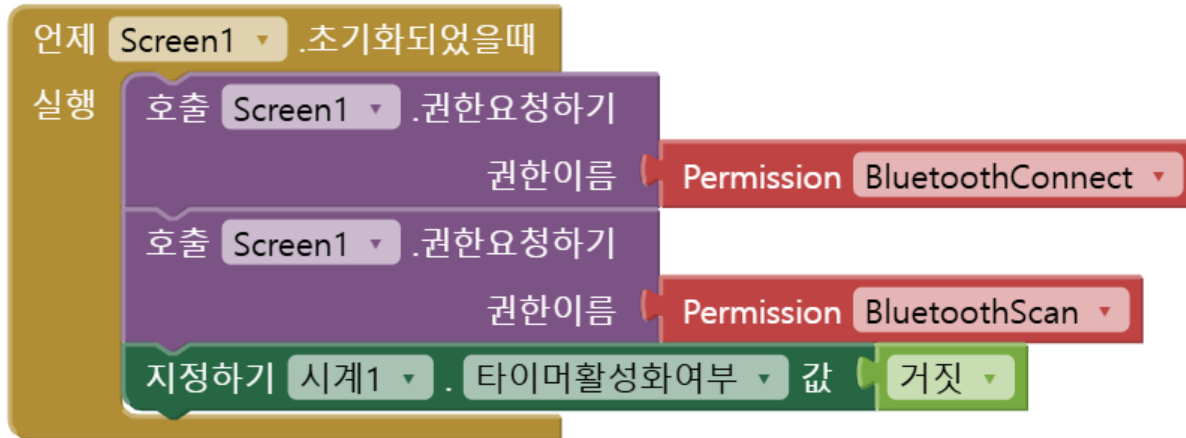
        delay(100); // 루프 속도 제어
    }
}
```

코드 전체 코드 앱인벤터



앱 인벤터 어플 실행될 때 처리

- 블루투스 연결, 블루투스 목록 출력 권한 요구
- 시계 비활성화 함.

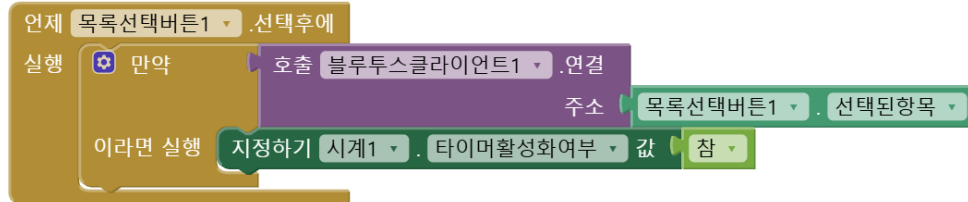


앱 인벤터 블루투스 선택 전, 후

- 블루투스 선택 전
 - 블루투스 연결할 장치 목록 출력

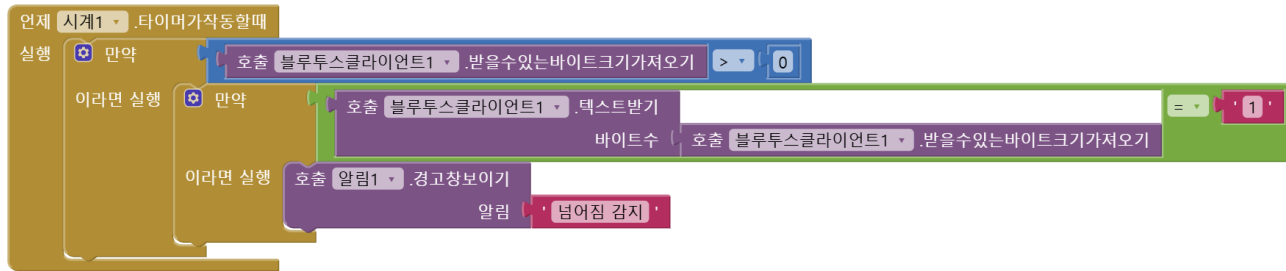


- 블루투스 선택 후
 - 선택한 블루투스 장치 연결
 - 연결한 후, 타이머활성화 여부 킴.
 - 타이머 활성화 키면 타이머가 작동할 때 이벤트가 실행됨.



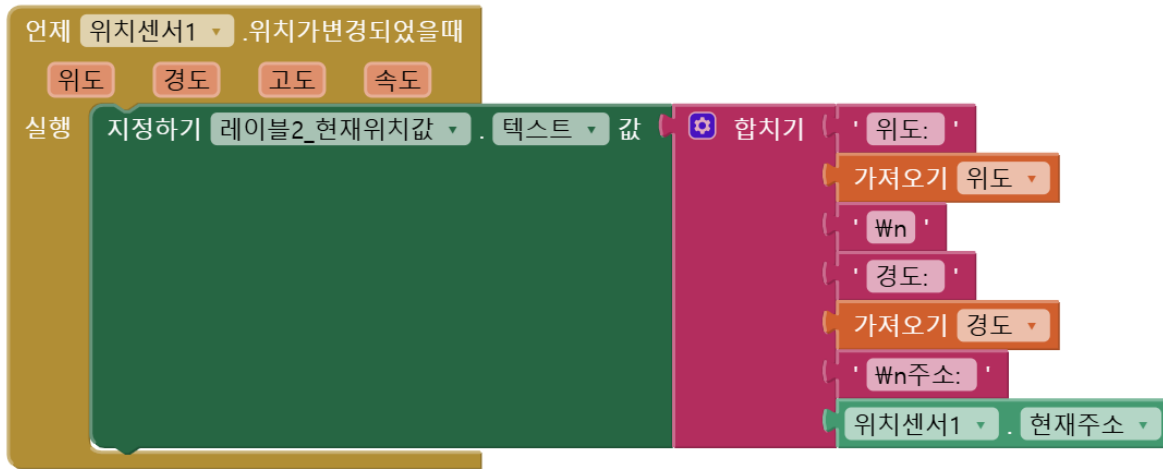
앱 인벤터 타이머 작동할 때

- 블루투스에서 받은 데이터 길이가 0보다 크면(받은 값이 있으면)



- 데이터 받은 길이 만큼 데이터 받아옴
- 만약 받아온 데이터가 "1"이면
 - 넘어진 상태로 알림 출력

앱 인벤터 GPS 정보 출력



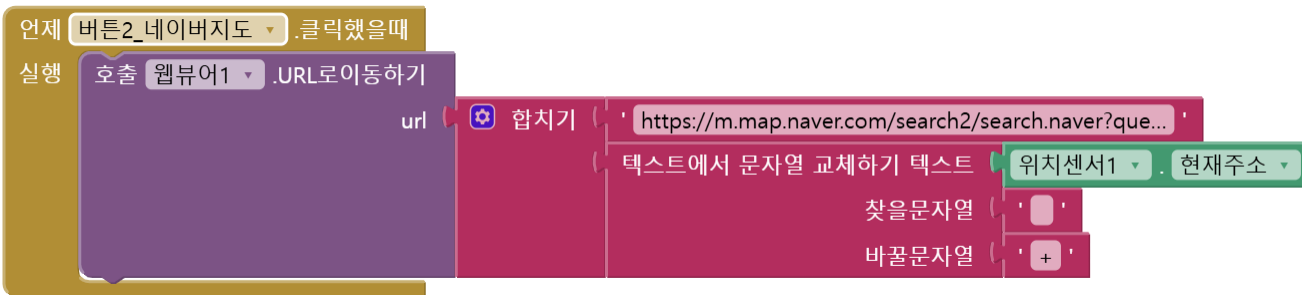
- GPS 상태 값이 달라지면
 - 현재 위도와 경도를 텍스트 출력

앱 인벤터 GPS 지도 출력



- 구글 지도 링크 구조에 맞추어 접속해야함.

- 구글 지도는 위도 경도를 주소 안에 넣어야 휴대폰 위치에 맞는 지도 출력함.



네이버 지도 또한 링크 구조에 맞추어 링크 접속해야함.

- 위치센서에 있는 현재 주소를 가져온 값(한국주소)를 기반으로 검색하여 결과 출력

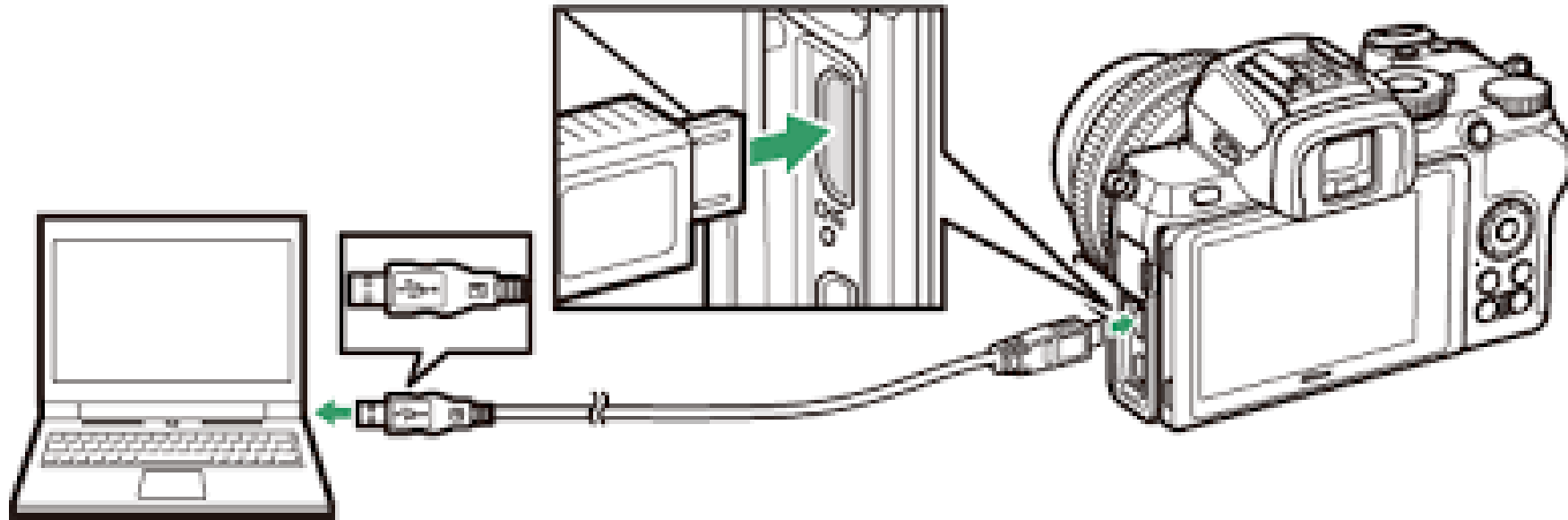
앱 인벤터 공유 텍스트 출력



- 위도 경도 주소를 하나로 텍스트 제작하고, 구글 지도 링크로 현 위치 지도 텍스트로 출력

작동방법

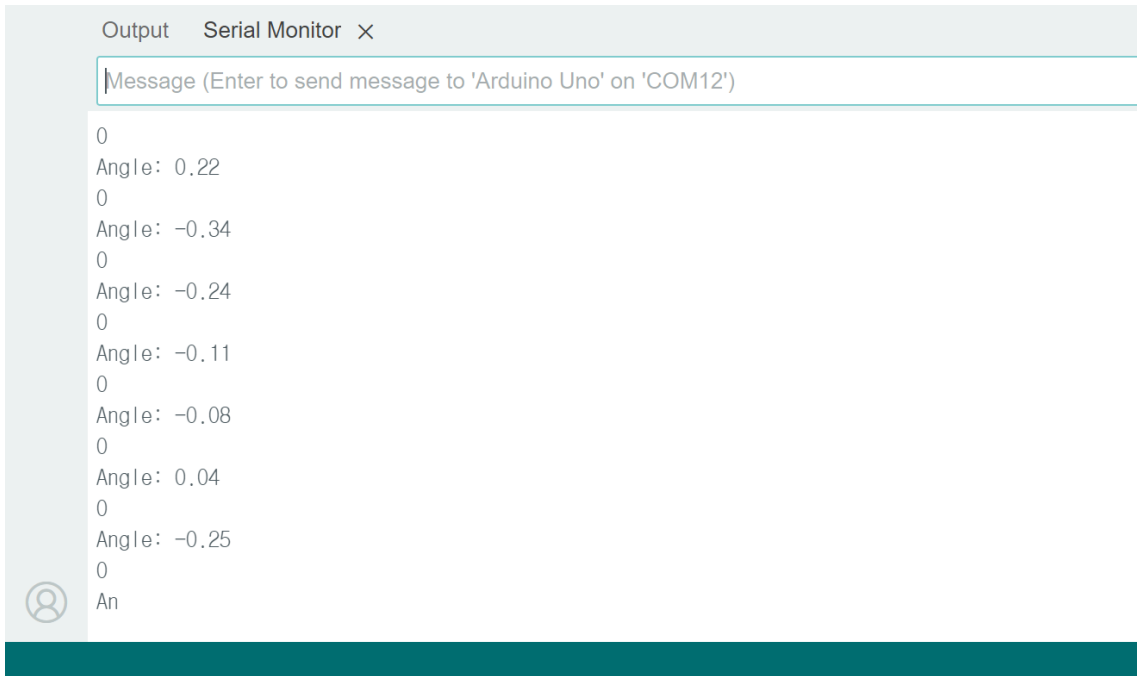
1. USB 연결하기(아두이노, 웹캠)



예시 이미지임.

2. 아두이노 센서 값 확인

아두이노 실행 시 보여지는 시리얼 값



```
Output Serial Monitor X
Message (Enter to send message to 'Arduino Uno' on 'COM12')
0
Angle: 0.22
0
Angle: -0.34
0
Angle: -0.24
0
Angle: -0.11
0
Angle: -0.08
0
Angle: 0.04
0
Angle: -0.25
0
An
```

- **USB 연결(전원 켤때)**

- **SETUP 함수 실행**

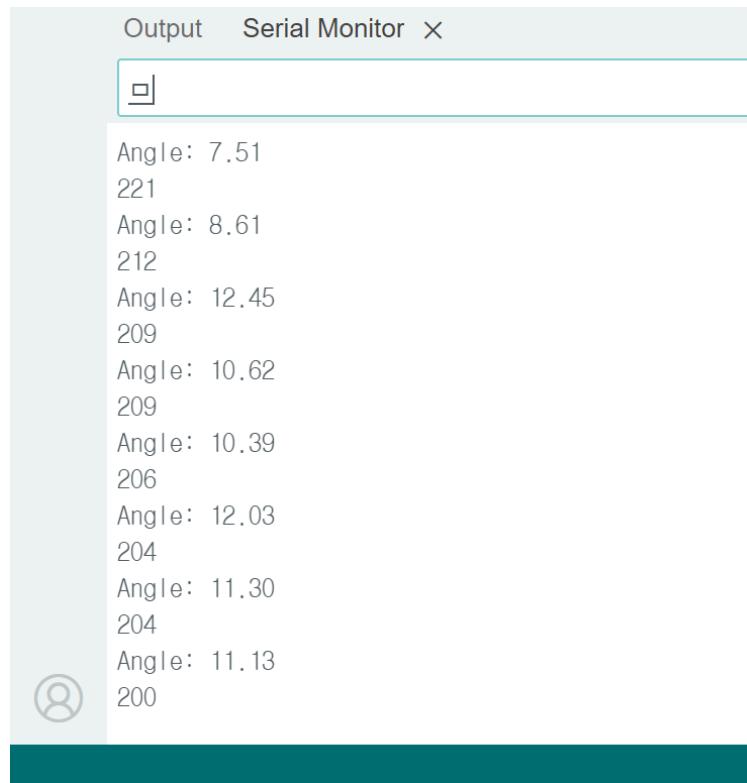
- SETUP 함수에는 자이로센서 캘리브레이션(쉽게 말해 영점조절)하므로 처음에 신발이 바닥에 두어야함.
 - 캘리브레이션 끝나면 부저로 소리 출력함.

- **! 즉 ! 부저 소리나기전에 신발은 움직이면 안됨!**

- **신발을 바닥에 두어 둘때에**

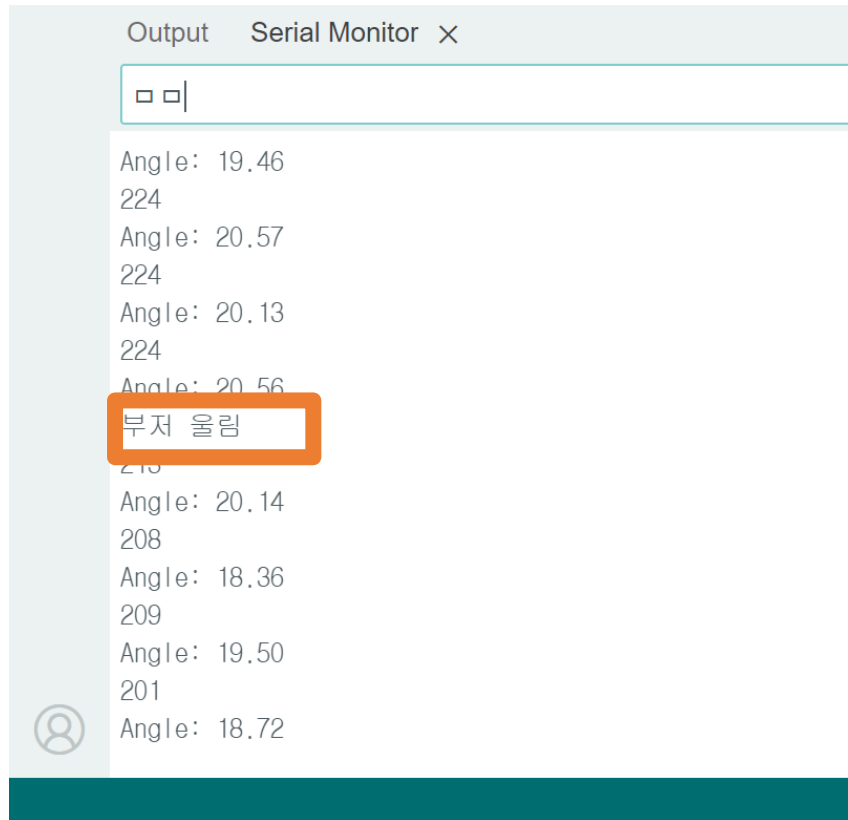
- **Angle은 자이로센서 값, 다른 숫자는 압력센서 측적값으로 0에 근접하게 나와야함.**

아두이노 실행, 신발 회전과 압력센서 누르면



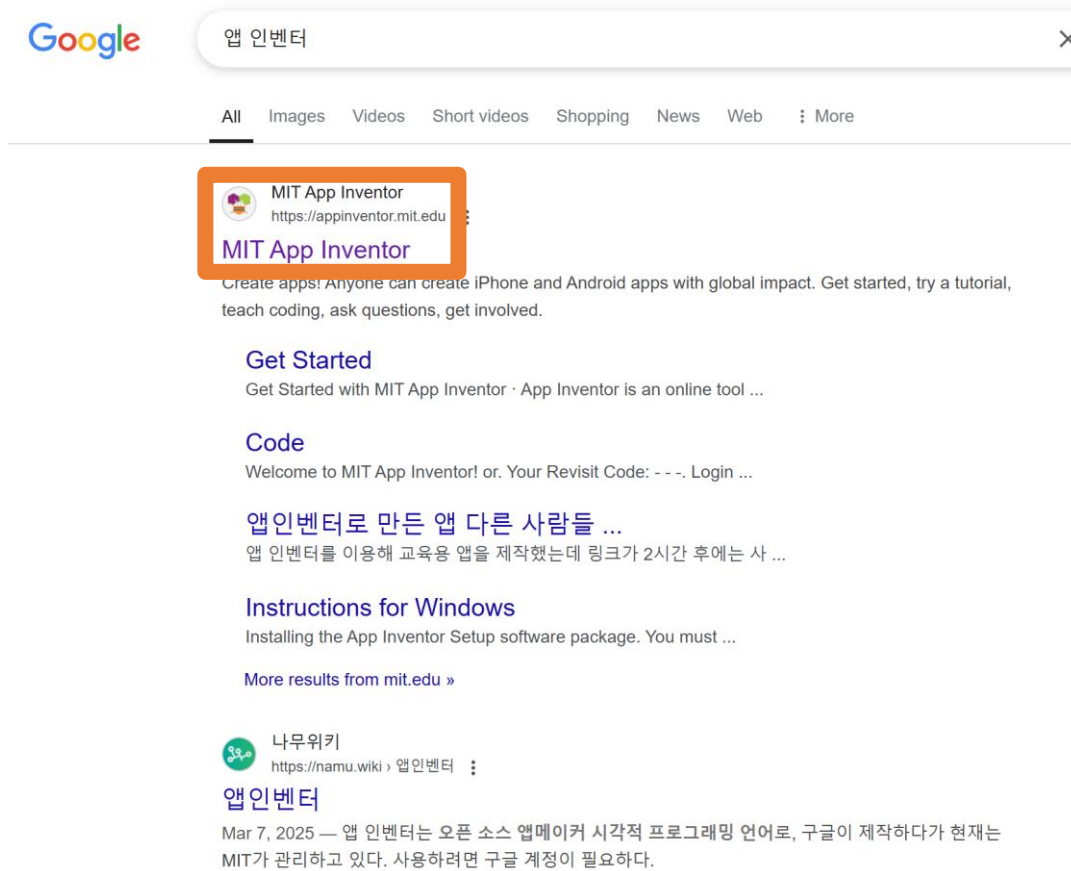
- Angle 값이 0과 멀어져야함.
 - 양수든 음수로 떠야함.
 - 넘어짐 감지는 15도 이상, -15 이하여야함.
- 압력서가 128이상 출력되게 움직여서 시연해야함.
- 그리고 3초 이상 위 조건에 맞추어야함. 그래야 넘어짐 감지로 출력

아두이노 넘어짐 조건으로 실행되면?



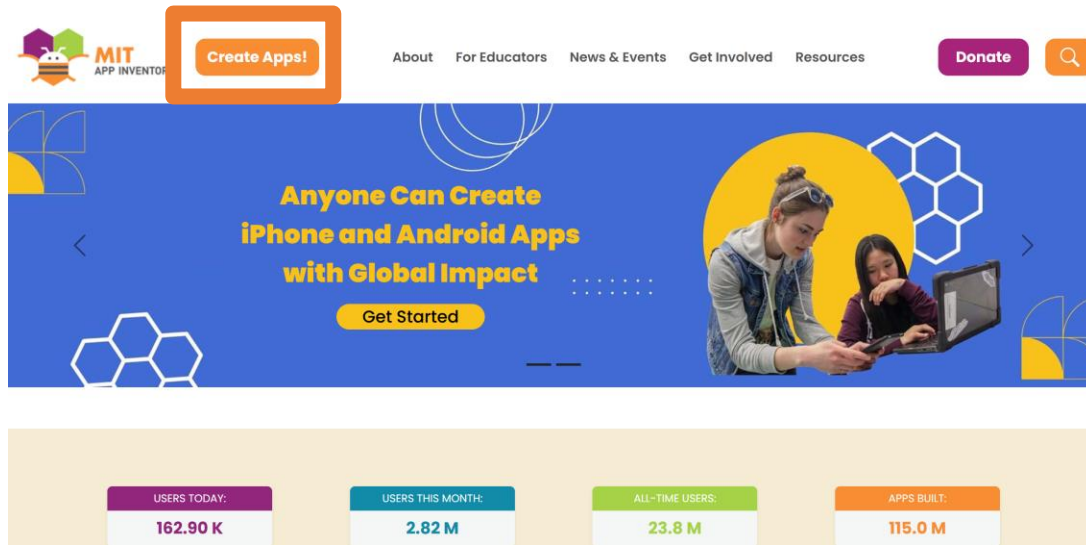
- 부저 울림 출력되면서 소리가 출력 됨.
- 어플 제대로 세팅했으면, 넘어짐 감지 알림이 가야함.
- 또 넘어짐 시연하려면, 각도가 0도가깝게 하거나, 압력센서가 128미만으로 나오게 하고 시연해야함.

3. 앱 인벤터 접속



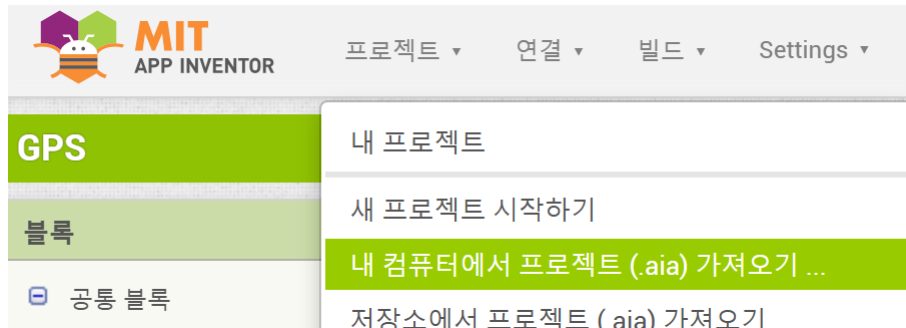
- 강 구글에 앱인벤터 실행하고 네모 박스에 맞추어서 접속

4. 앱인벤터 로그인

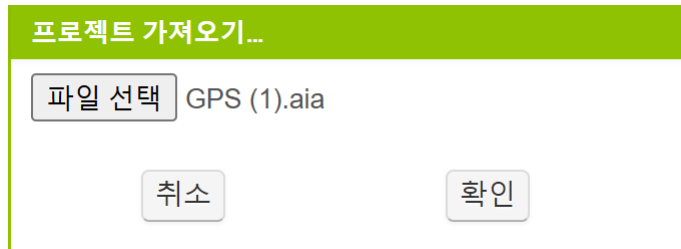


- Create Apps 눌러서 앱인벤터 로그인한다.
- 구글로 간편하게 하세요 구글 로그인도 못하면 시연도 못하니깐 구글 로그인 되는지 보고,
- 부모님 계정 보호로 인해서 로그인 안되면 학생 부주의임.

5. 프로젝트 파일 불러오기



- GPS (1).aia 파일이 있어야 한다!
- 그 파일 선택하고 확인 버튼 누른다.



6. 휴대폰에 앱 인벤터 설치(안드로이드)

MIT App Inventor

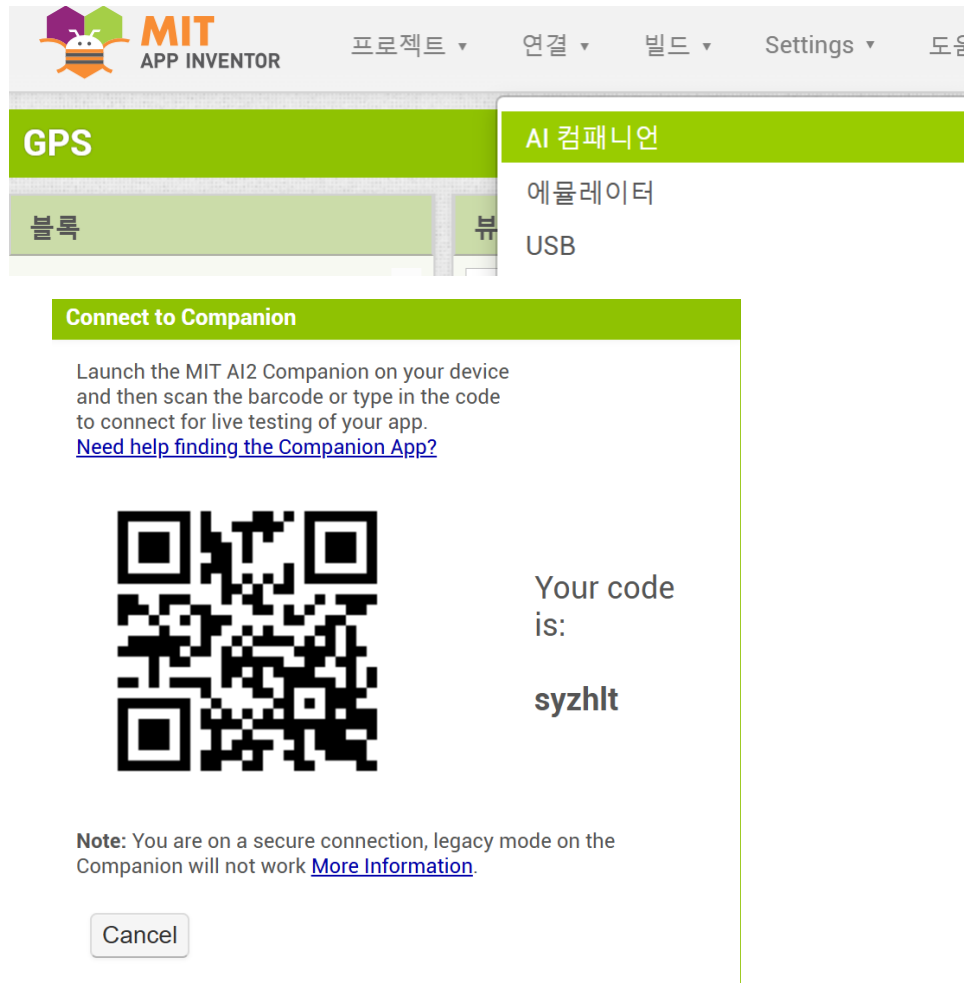


MIT AI2 Companion
MIT App Inventor
3.1 ★

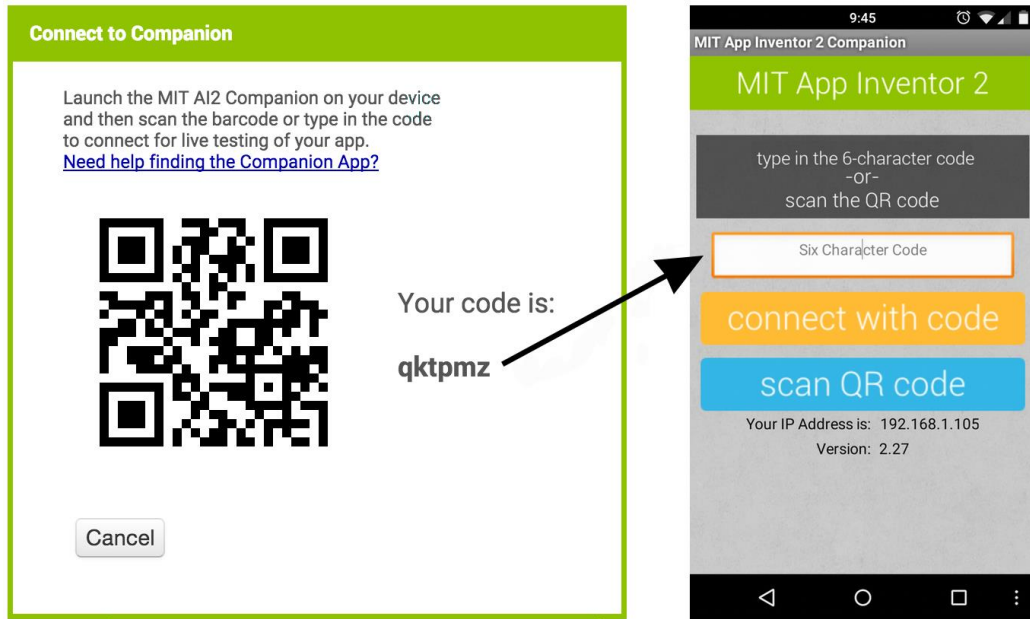
- 옆에 구글플레이 스토어에 옆에 이름 검색해서 다운로드 받으셈.

7. 컴퓨터에 앱 연결하기

- 연결-ai 컴패니언 들어가서 나오는 QR 띄우도록 한다.

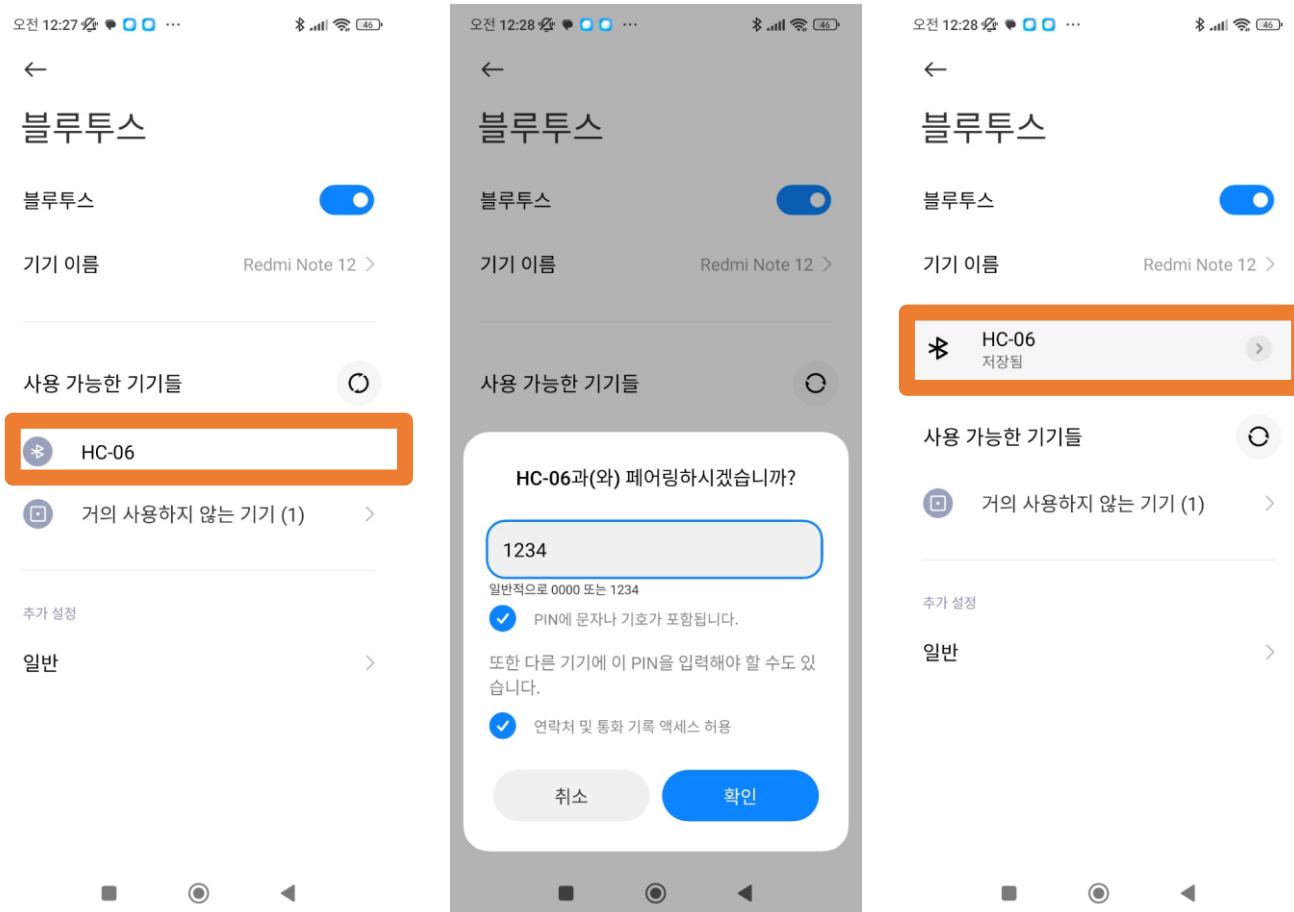


8. 앱 인벤터에서 QR 스캔



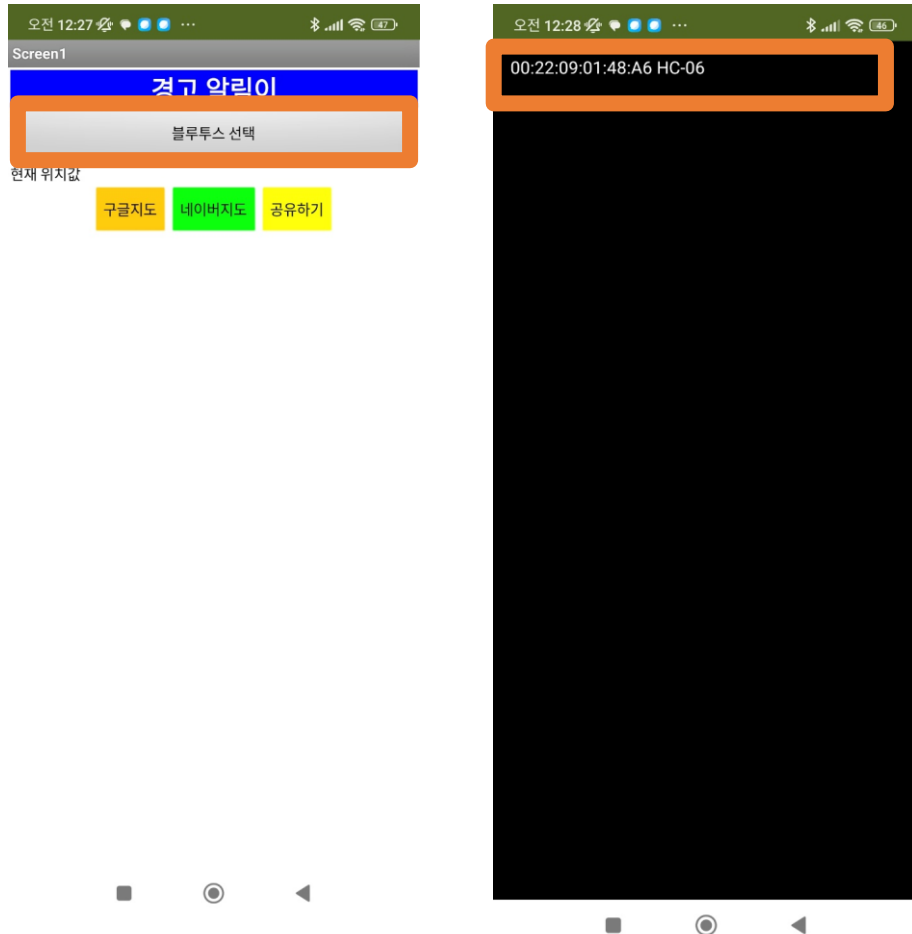
- Scan QR code 눌러서 본인 앱 인벤터 화면에 보이는 QR에 스캔해서 프로젝트 불러오기.

9. hc06 블루투스 연결하기



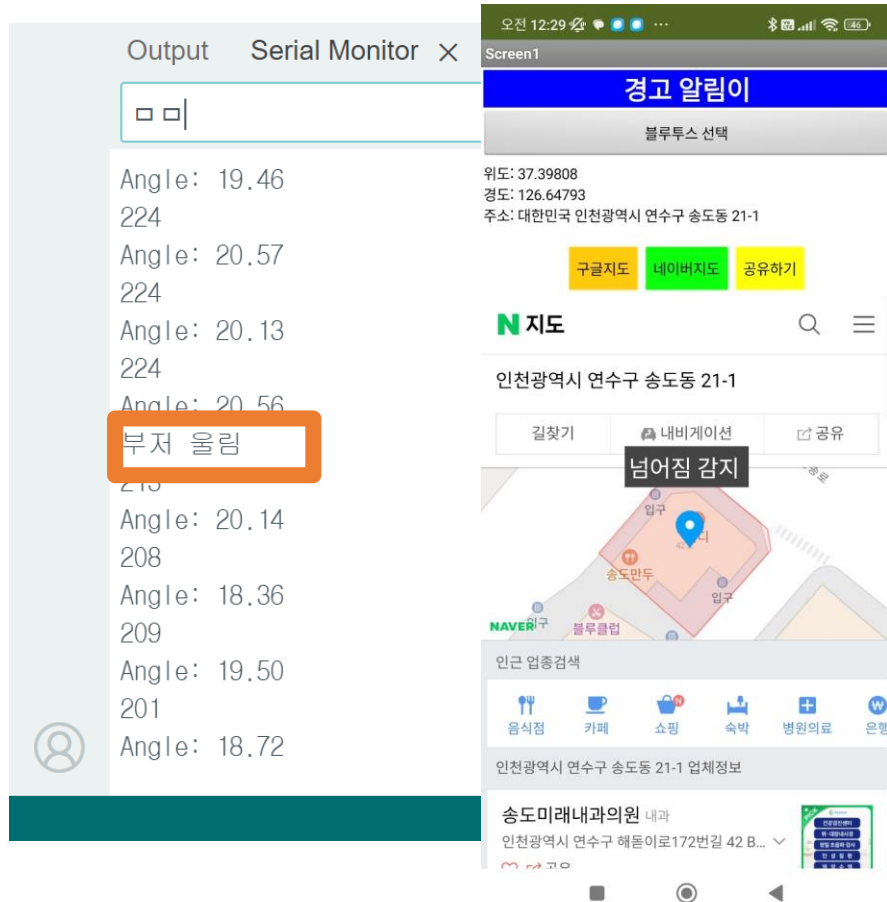
- 블루투스 활성화 하고 HC-06 연결하기
- 핀 번호는 1234 입력 후, 엑세스 허용 체크하고 연결.
- HC-06 저장됨으로 뜨게끔 하면 됨.

10. 앱 인벤터에서 블루투스 연결하기



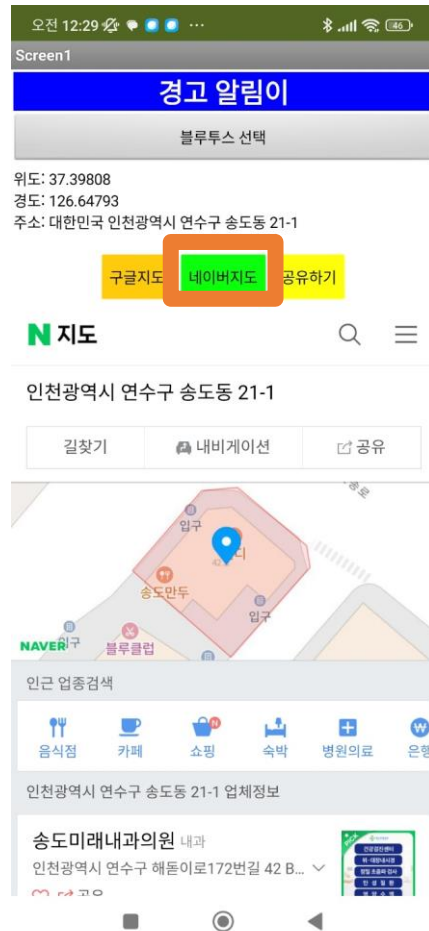
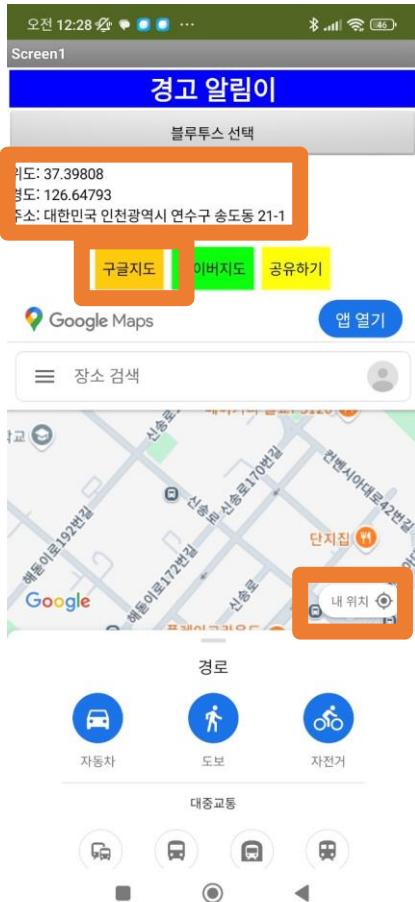
- 블루투스 선택 버튼 누르면
- HC-06 뜨는지 확인하고.
- HC-06 눌르면 블루투스 연결 될거임.

11. 앱 인벤터 넘어짐 감지 하기



- 아두이노에서 부저 울리겠금 센서를 조절한다.
- 부저 울림 텍스트가 출력되면, 어플에서도 넘어짐 감지 알림 감.

12. 지도 시연 해주기



- 웬만해서 따로 해야할 것은 없음.
- 위도 경도 출력되는지 보고, 주소도 자동으로 보여질것이다.
- 구글, 네이버 지도 버튼 누르면
서 시연하면 됨.
 - 구글은 내 위치 버튼 눌러서 해당
위치 출력하겠금 하셈.