

사물인터넷(IoT) 자동화 시스템

개발자 과정



4조

신경아, 문우정, 최희지, 정혜윤

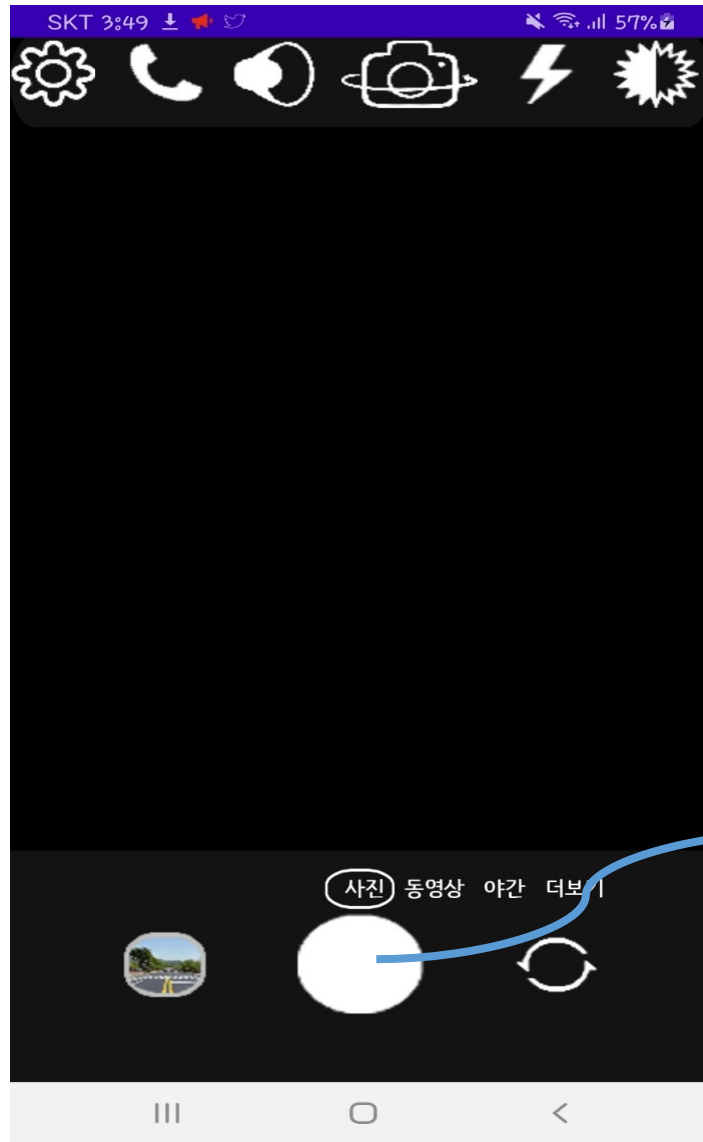


팀 프로젝트

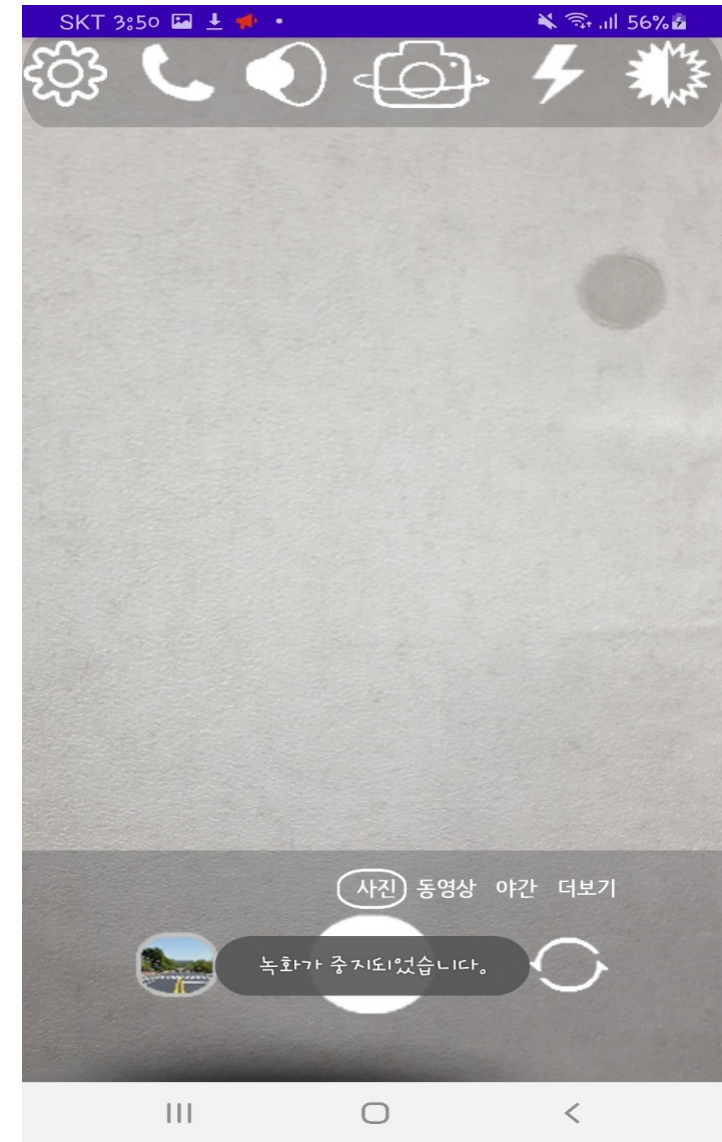
블랙박스 앱 개발

- 화면
- 주행모드
- 촬영 & 재생
- GPS & 현재위치 확인
- 충격 감지
- 긴급 전화
- 설정
- 외부, 내부 디렉토리

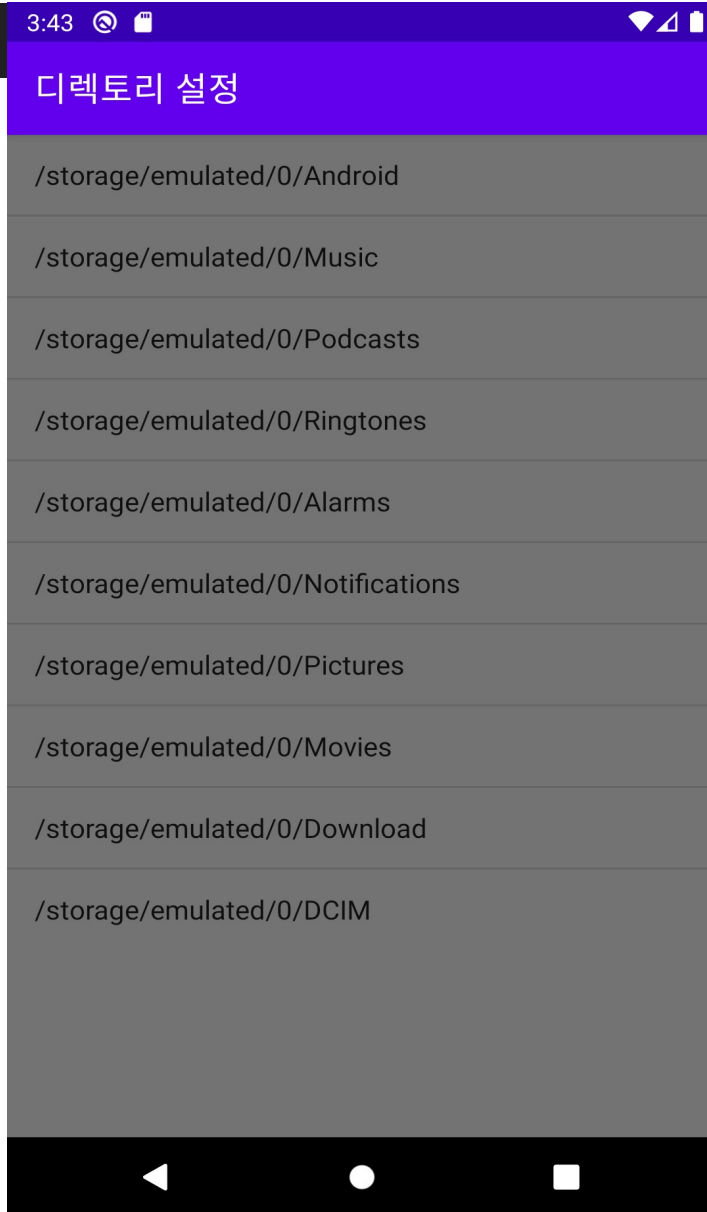




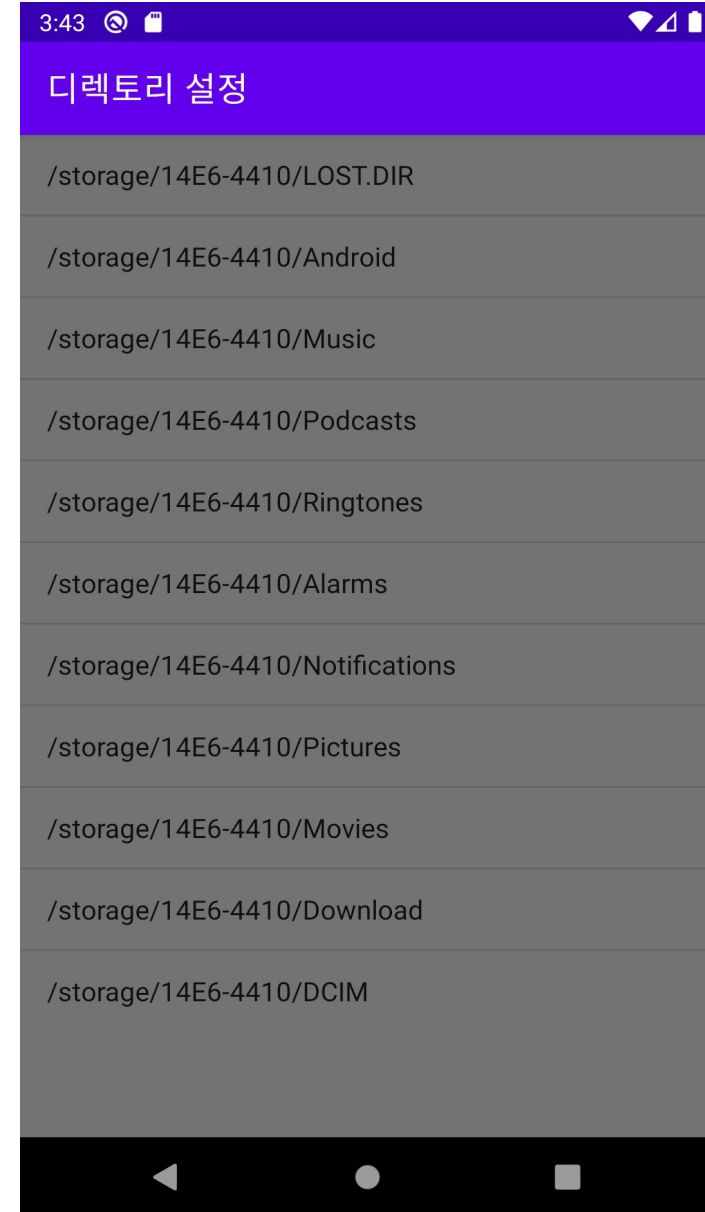
촬영 버튼



내부 저장소

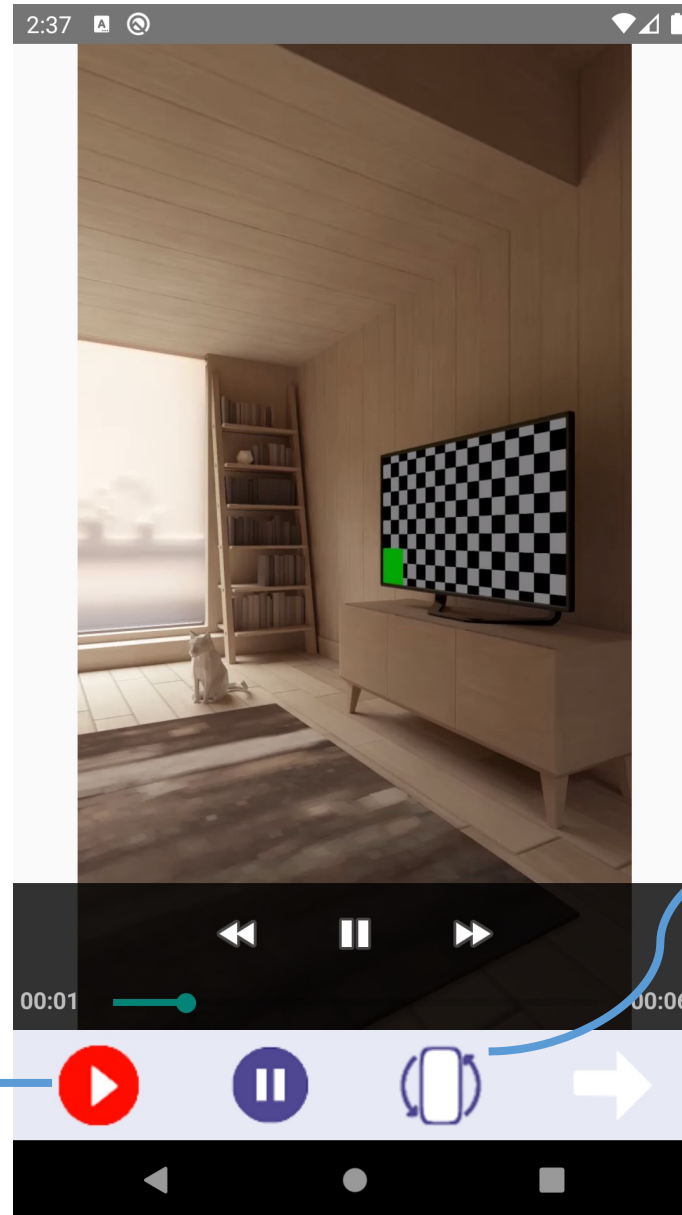


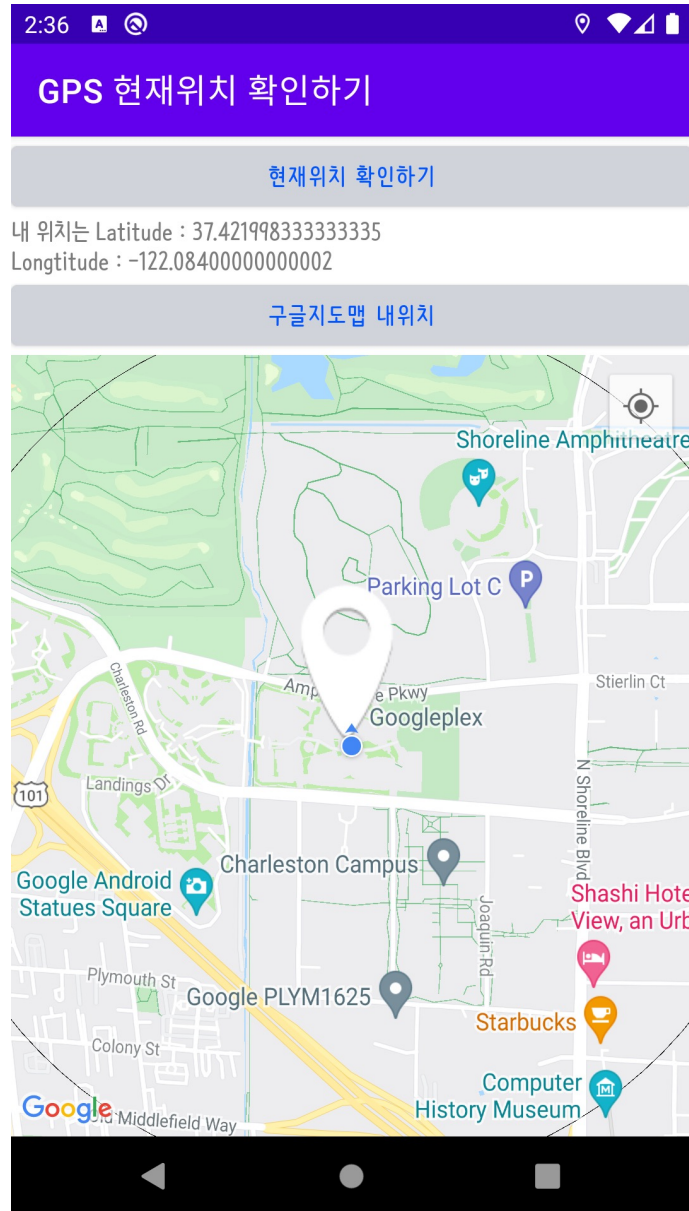
외부 저장소

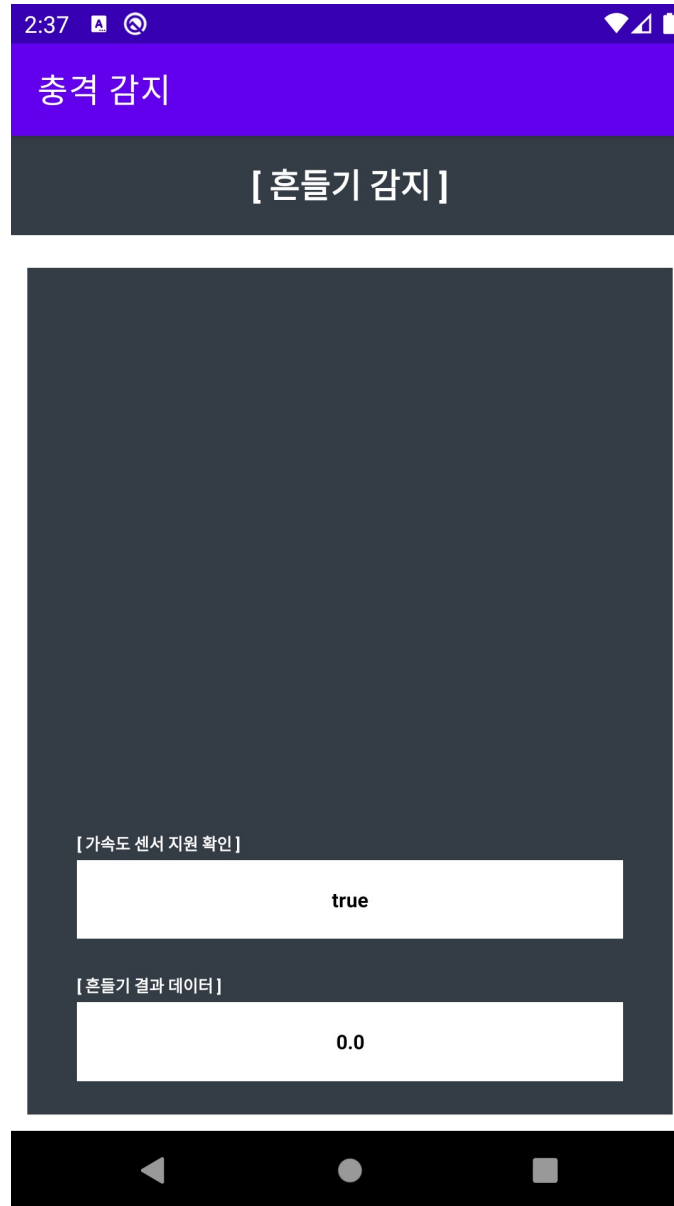


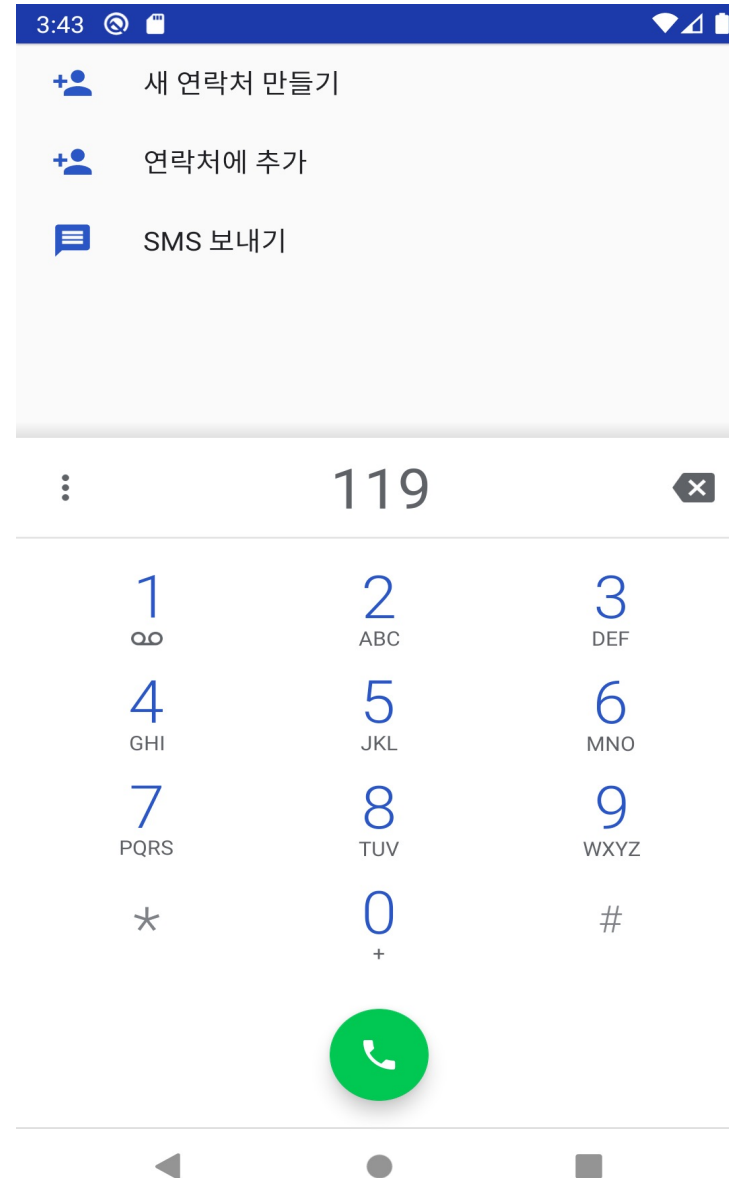
재생 &
일시정지

화면 회전











```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.FOREGROUND_SERVICE" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_BACKGROUND_LOCATION" /><!--gps 권한 -->
```

```
<meta-data
    android:name="com.google.android.maps.v2.API_KEY"
    android:value="AIzaSyBHvnmcr6AXOS7keoEXtyNnhIU5T8_mdXY" />
```

```
public void onLocationChanged(Location location) {
    double latitude = location.getLatitude();
    double longitude = location.getLongitude();
    String message = "내 위치는 Latitude : " + latitude + "\nLongitude : " + longitude;
    textView1.setText(message);

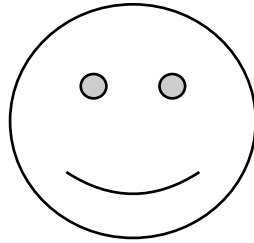
    showCurrentLocation(latitude,longitude);
    Log.i( tag: "MyLocTest", msg: "onLocationChanged() 호출되었습니다.");
}
```

```
public class SampleVideoRecorder extends Activity implements SurfaceHolder.Callback {
```

```
    public void run() {  
        Toast.makeText(context: SampleVideoRecorder.this, text: "녹화가 시작되었습니다.", Toast.LENGTH_SHORT).show();  
        try {  
            mediaRecorder = new MediaRecorder();  
            camera.unlock();  
            mediaRecorder.setCamera(camera);  
            mediaRecorder.setAudioSource(MediaRecorder.AudioSource.CAMCORDER);  
            mediaRecorder.setVideoSource(MediaRecorder.VideoSource.CAMERA);  
            mediaRecorder.setProfile(CamcorderProfile.get(CamcorderProfile.QUALITY_720P));  
            mediaRecorder.setOrientationHint(90);  
            mediaRecorder.setOutputFile("/sdcard/Pictures/TEST2.mp4");  
            mediaRecorder.setPreviewDisplay(surfaceHolder.getSurface());  
            mediaRecorder.prepare();  
            mediaRecorder.start();  
            recording = true;  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

```
    public void onPermissionGranted() {  
        Toast.makeText(context: SampleVideoRecorder.this, text: "권한 허가", Toast.LENGTH_SHORT).show();  
  
        camera = Camera.open();  
        camera.setDisplayOrientation(90);  
        surfaceView = (SurfaceView)findViewById(R.id.surfaceView);  
        surfaceHolder = surfaceView.getHolder();  
        surfaceHolder.addCallback(SampleVideoRecorder.this);  
        surfaceHolder.setType(SurfaceHolder.SURFACE_TYPE_PUSH_BUFFERS);  
    }  
}
```

```
public void onSensorChanged(SensorEvent event) {  
    if (event.sensor.getType() == Sensor.TYPE_ACCELEROMETER) {  
        Log.d( tag: "---", msg: "---");  
        Log.d( tag: "//=====//", msg: "=====");  
        Log.d( tag: "", msg: "\n"+ "[A_Shake > onSensorChanged() 메소드 : 가속도 센서 값 변경 상태 확인]");  
        Log.d( tag: "//=====//", msg: "=====");  
        Log.d( tag: "---", msg: "---");  
  
        //TODO [흔들기 감지를 위한 변수값 선언]  
        //final float SHAKE_GRAVITY = 2.7F;  
        final float SHAKE_GRAVITY = 0.5F;  
  
        //TODO [이벤트로 들어온 X, Y, Z 값 확인]  
        float axisX = event.values[0];  
        float axisY = event.values[1];  
        float axisZ = event.values[2];  
  
        //TODO [중력 가속도 X, Y, Z 값 구하기]  
        float gravityX = (axisX / SensorManager.GRAVITY_EARTH);  
        float gravityY = (axisY / SensorManager.GRAVITY_EARTH);  
        float gravityZ = (axisZ / SensorManager.GRAVITY_EARTH);
```



감사합니다.