



군산대학교
KUNSAN NATIONAL UNIVERSITY

DDMS와 센서

모바일 시스템 소프트웨어
남 광 우

DDMS - 에뮬레이터에서 GPS와 통화/SMS 사용하기

❖ 안드로이드 DDMS

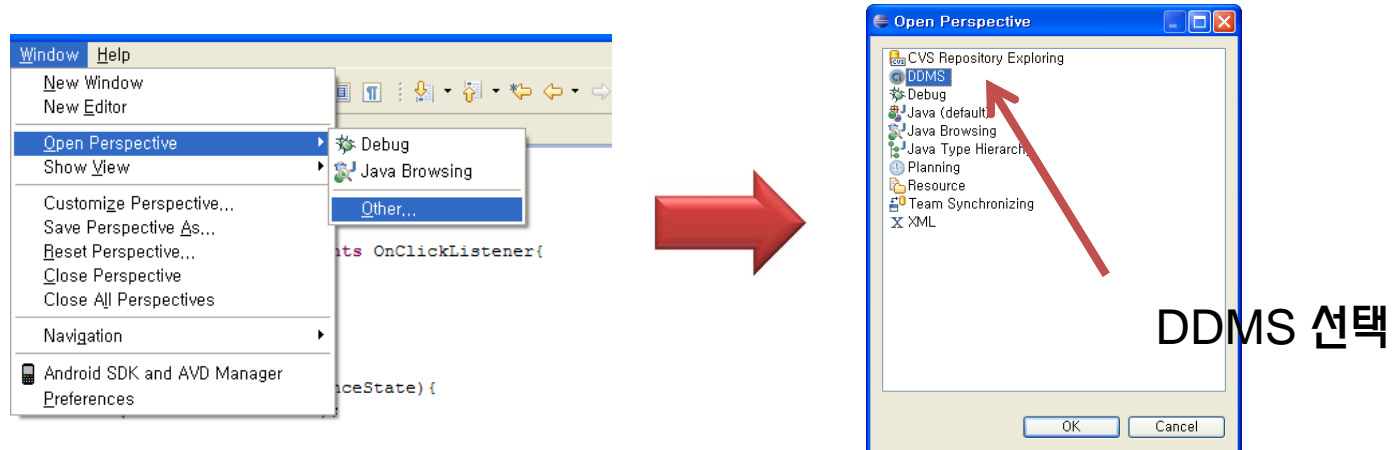
- ✓ DDMS = Dalvik Debug Monitor Server
- ✓ 에뮬레이터/단말기 내의 로그나 디버깅, 실행중인 프로세스 확인이나 화면 캡처 등의 작업을 수행할 수 있도록 해주기 위한 Eclipse용 툴

❖ DDMS의 지원 기능

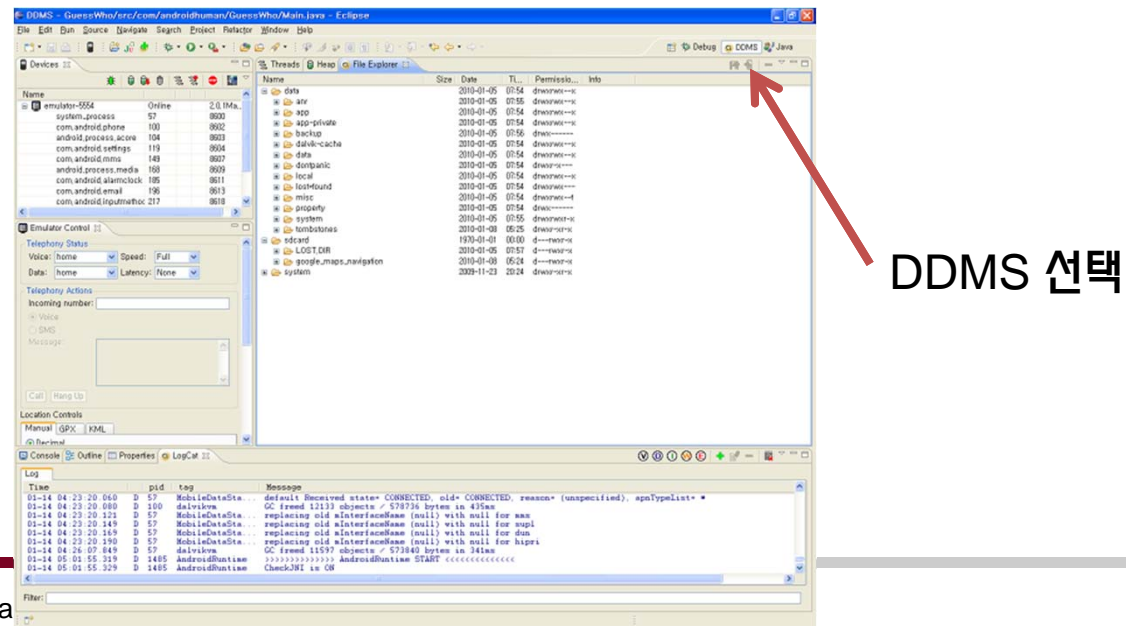
- ✓ Device의 모니터링
- ✓ Device Thread 모니터링
- ✓ Application Tracker
- ✓ File Explorer
- ✓ 가상 Telephony/SMS 전송
- ✓ 가상 GPS 가상 전송
- ✓ 스크린 캡처

DDMS - 에뮬레이터에서 GPS와 통화/SMS 사용하기

❖ Eclipse에서 DDMS Perspective 실행하기



✓ 또는,



DDMS - 에뮬레이터에서 GPS와 통화/SMS 사용하기

❖ DDMS 화면을 통한 기능 개요

File Explorer 및 Thread등

Device들
상태보기

The screenshot shows the DDMS interface with the following components:

- Devices List:** A table listing virtual devices. The first device is 'emulator-555'.
- File Explorer:** A tree view showing the file system of the selected device. It includes directories like 'cache', 'config', 'data', 'dev', 'etc', 'init', 'mnt', 'proc', 'root', 'sbin', 'sdcard', 'sys', and 'system'.
- LogCat:** A window at the bottom showing system logs. It includes a search bar and a table of log messages with columns for Time, PID, TID, Application, Tag, and Text.

Name	State	Version
emulator-555	Online	4.1 [4.1...]
system_p	147	8600
com.andri	213	8601
com.andri	227	8602
com.andri	245	8603
com.andri	254	8604
com.andri	277	8605
android.pr	306	8606
com.andri	326	8607
com.andri	345	8608
com.andri	373	8609
com.andri	387	8610
com.andri	413	8611
android.pr	427	8612
com.andri	458	8613
com.andri	476	8614
com.andri	533	8615
com.svox	575	8616
com.andri	588	8617
com.andri	607	8618
com.andri	621	8619
com.goog	650	8620

Time	PID	TID	Application	Tag	Text
10-04 01:54:38.059	621	720	com.android.vending	vending	NDROID
10-04 01:54:38.069	621	621	com.android.vending	vending	[89] MarketWidgetProvider\$updateTask.doInBackground: .api.FatalCommunicationException: Sent a bad r
10-04 01:55:08.169	621	721	com.android.vending	vending	[1] MarketWidgetProvider\$updateTask.onPostExecute [90] VendingApplication.getAuthTokenBlocking() NDR

Emulator
Control :
GPS/SMS
/Telephone

DDMS - 전화걸기

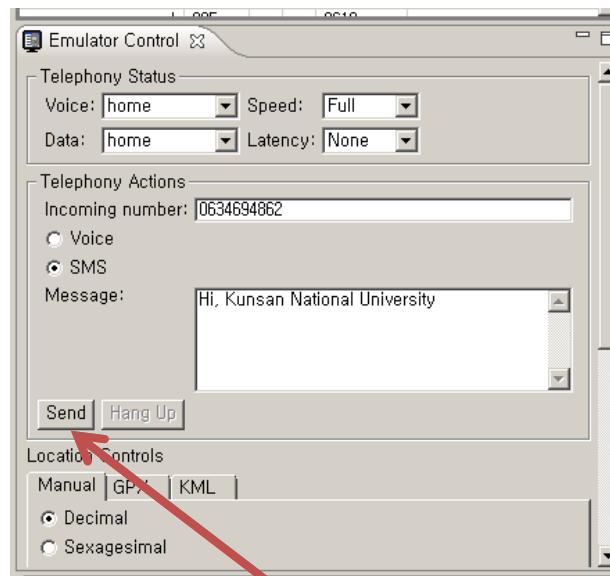
❖ DDMS Emulator Control을 통한 전화걸기

- ✓ 에뮬레이터의 통신 상태 조절 및 가상의 위치 정보를 설정해주는 것이 가능
- ✓ 에뮬레이터에 가상으로 전화를 걸거나 SMS를 전송 가능
 - 실제 기기를 선택한 상태에서는 이 메뉴를 사용할 수 없음
 - Contact에 전화번호가 있으면 이름이 보임

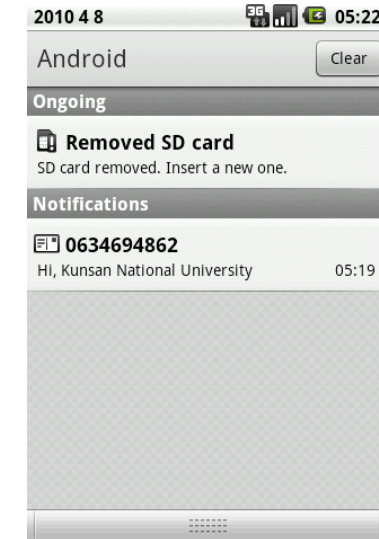
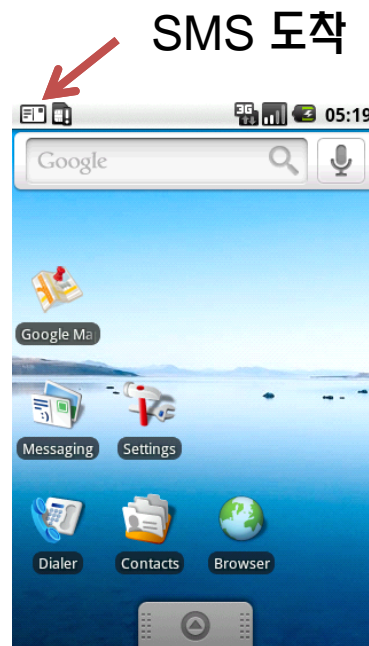


DDMS – SMS 보내기

- ❖ DDMS Emulator Control을 통한 SMS 보내기
 - ✓ 한글 SMS는 화면에 깨져서 보이므로 영문으로 전송



SMS 보내기



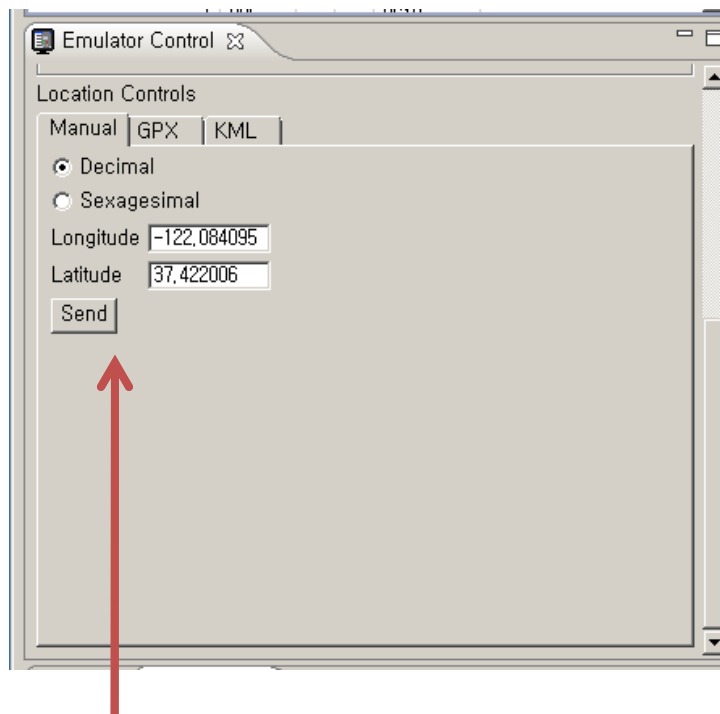
0634694862: Hi, Kunsan National University
Sent: 05:18

Type to compose Send

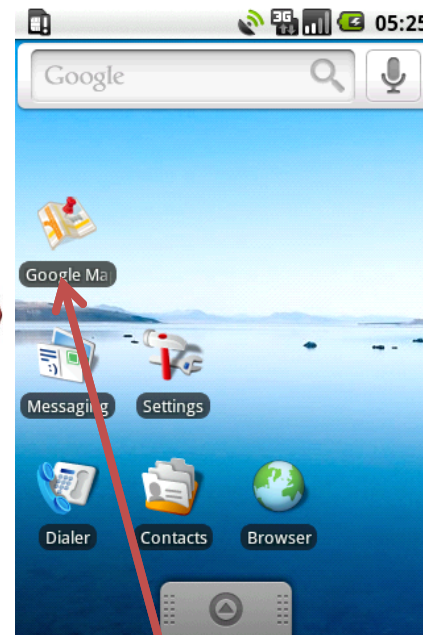
DDMS – GPS 값 보내기

❖ DDMS Emulator Control에서 GPS값 보내기

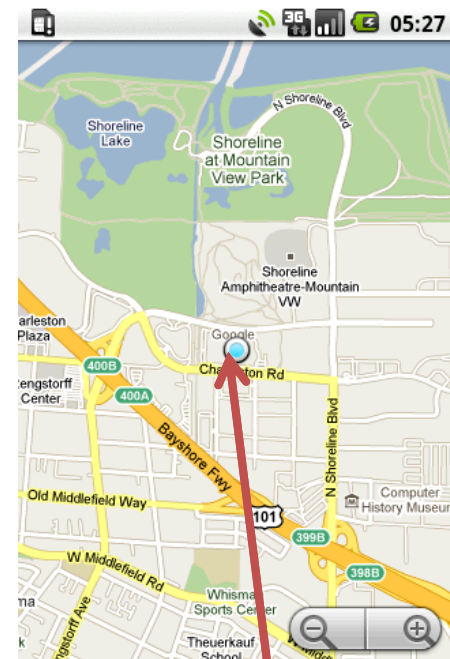
✓ 단일 GPS값 보내기



위경도 좌표설정 후 Send



안드로이드 마켓에서
구글맵설치후 실행

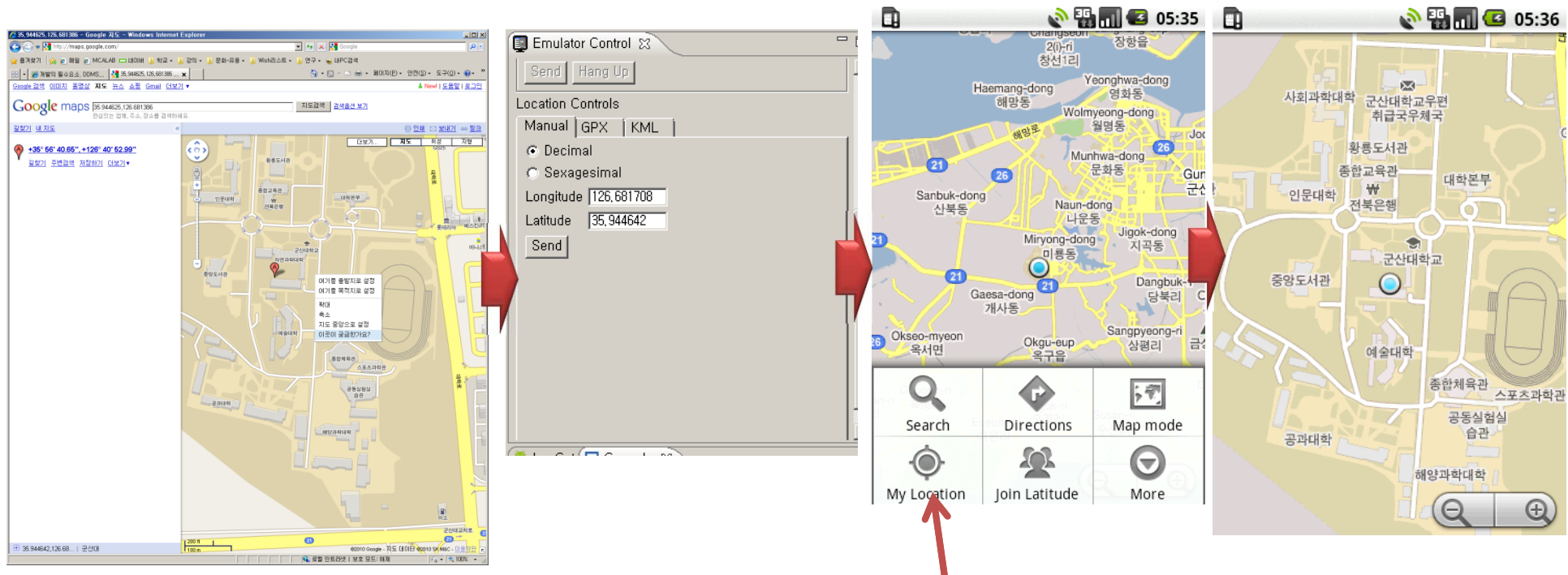


현재위치

DDMS – GPS 값 보내기

❖ DDMS Emulator Control에서 GPS값 보내기

✓ 위치 아는 방법: maps.google.com => 마우스 오른쪽버튼



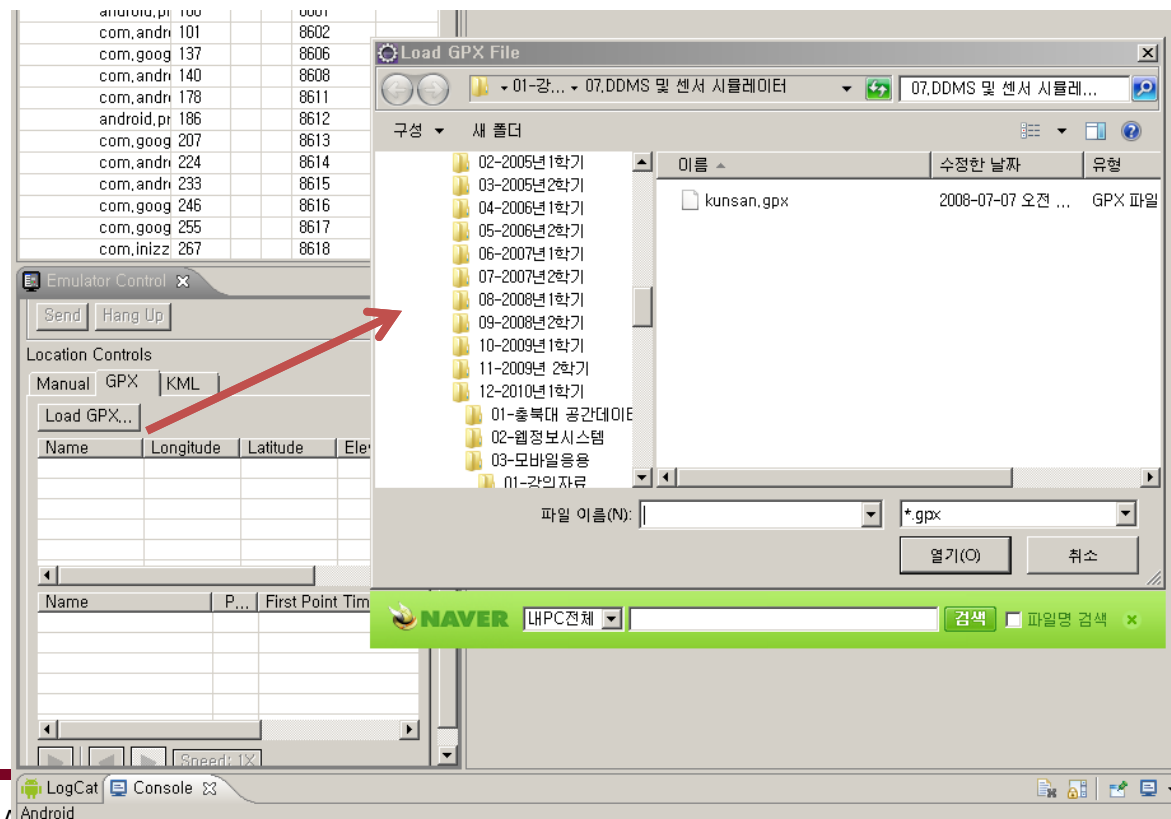
메뉴 버튼후 MyLocation

DDMS – GPS Tracking 정보 보내기

❖ DDMS Emulator Control에서 GPS값 보내기

✓ GPS Tracking 정보 보내기(GPX 데이터)

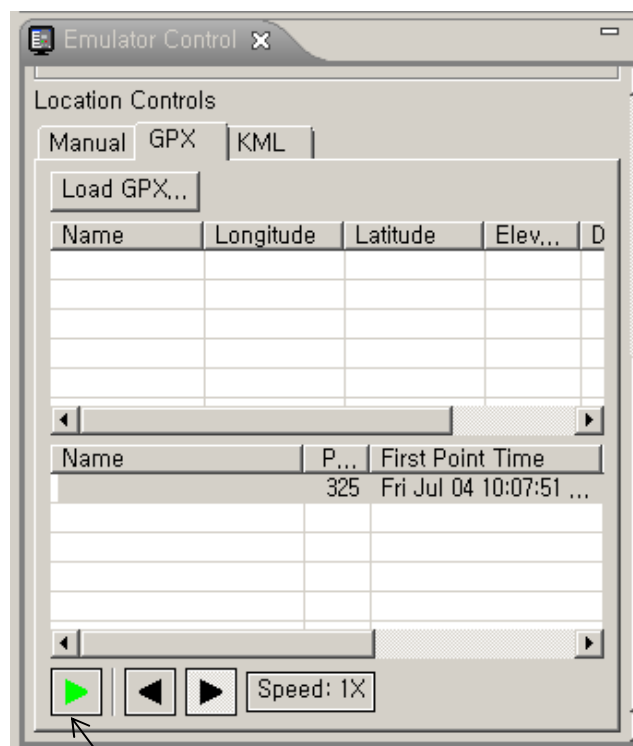
- mcalab.kunsn.ac.kr => 강의게시판=> 강의자료05번 첨부파일
- GPX 1.1 데이터 => 인터넷 참고
 - KML 사용가능



DDMS – GPS Tracking 정보 보내기

❖ DDMS Emulator Control에서 GPS값 보내기

- ✓ GPS Tracking 정보 보내기(GPX 데이터)



Play 누르기



GPS 값 자동 전송됨

Sensor의 종류

- 모바일 디바이스에서 주로 사용되는 센서는 아래와 같고 대부분은 안드로이드에서는 대부분 지원한다.

가속도 : 기울기 정보

지자기 : 자기 세기 정보

Orientation Sensor : 가속도와 지자기를 같이 감지할 수 있다. autocalibration 기능을 가지고 있다.

조도 ALS(Ambient Light Sensor) : 주변 밝기 정보 : lux 단위

근접 PS(Proximate Sensor) : 조도 센서와 같은 칩으로 사용되는 것이 대부분이며 물체와의 거리 정보를 알 수 있다. cm단위

자이로센서 : 좌우로 반복적 이동시 중간 데이터 처리 없이? 줄여 반응 속도를 개선한 것임

Android Sensor Class Type

- TYPE_ACCELEROMETER : 3차원 중력 가속도를 측정하는 가속도 센서
- TYPE_ALL : 모든 센서
- TYPE_GYROSCOPE : 회전운동 센서
- TYPE_LIGHT : 빛의 밝기를 측정하는 조도 센서
- TYPE_MAGNETIC_FIELD : 3차원 자기장 세기를 측정하는 지자기 센서(나침반)
- TYPE_ORIENTATION : 다른 장치들과 결합해 기기의 현재 방향결정
- TYPE_PRESSURE : 압력 센서
- TYPE_PROXIMITY : 어떤 물체와의 거리를 측정하는 근접 센서
- TYPE_TEMPERATURE : 온도를 측정하는 온도 센서
-

[출처] [Sensor 처리 과정](#) | 작성자 [즐겁게](#)

Sensor

❖ 좌표 시스템

✓ 좌표 축 정의

X축 : 화면에 수평축(portrait 모드에서 짧은 에지를 landscape 모드에서 긴 에지)을 나타내며 오른쪽을 가리킨다.

Y축 : 화면에 수직축을 나타내며 화면 위쪽을 가리킨다.(원점은 왼쪽 밑 코너)

Z축 : 단말이 화면을 위로 해서 테이블에 올려져 있다고 생각했을 때 하늘을 가리킨다.



Sensor - Orientation

❖ Orientation

✓ SENSOR_ORIENTATION (방향 센서)

각 배열의 값은 각도를 나타냅니다.

단말기 화면이 하늘을 향한 상태로 테이블 위에 수평으로 놓여있는 상태를 기준으로 각 축을 중심으로 회전시키는 것을 생각하면 됩니다.

values[0] : Z 축을 중심으로 회전 ($0 \leq \text{azimuth} < 360$)

0 = 북, 90 = 동, 180 = 남, 270 = 서

values[1] : X축을 중심으로 회전 ($-180 \leq \text{pitch} \leq 180$)

Z축이 Y축 방향으로 향하면 0보다 큰값

화면이 하늘을 향하고 테이블위에 수평으로 놓여있는 상태 0, 화면이 아래를 향하면 -180 or 180,

똑바로 세우면 -90, 거꾸로 세우면 +90

values[2] : Y축을 중심으로 회전 ($-90 \leq \text{roll} \leq 90$)

Z축이 X축 방향으로 향하면 0보다 큰값

Sensor - Accelerometer

❖ 가속도계

✓ Accelerometer

SENSOR_ACCELOROMETER (가속도 센서)

각 배열의 값은 (m/s^2) 단위로 되어있으며, 접촉힘(Contact Force)을 측정합니다.

values[0] : X축에 적용되는 힘

values[1] : Y축에 적용되는 힘

values[2] : Z축에 적용되는 힘

❖ 지자기 센서

✓ Magnetic field

SENSOR_MAGNETIC_FIELD (자기장 센서)

모든 값은 micro-Tesla (μT) 단위로 되어있으며, X, Y, -Z 축 주변 자기장을 측정합니다.

* Z축의 값이 바뀐 것 주의

Sensor Programming

```
public class SensorActivity extends Activity, implements SensorEventListener {
    private final SensorManager mSensorManager;
    private final Sensor mAccelerometer;

    public SensorActivity() {
        mSensorManager = (SensorManager) getSystemService(SENSOR_SERVICE);
        mAccelerometer = mSensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
    }

    protected void onResume() {
        super.onResume();
        mSensorManager.registerListener(this, mAccelerometer, SensorManager.SENSOR_DELAY_NORMAL);
    }

    protected void onPause() {
        super.onPause();
        mSensorManager.unregisterListener(this);
    }

    public void onAccuracyChanged(Sensor sensor, int accuracy) {
    }

    public void onSensorChanged(SensorEvent event) {
    }
}
```

Sensor Programming

❖ SensorEventListener(2.2)

✓ 예제 : 핸드폰 흔들림 감지 <http://pulsebeat.tistory.com/44>

```
public class ShakeActivity extends Activity implements SensorEventListener {  
  
    ....  
  
    private SensorManager sensorManager;  
    private Sensor accelerometerSensor;  
  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        sensorManager = (SensorManager) getSystemService(SENSOR_SERVICE);  
        accelerometerSensor = sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);  
    }  
  
    public void onStart() {  
        super.onStart();  
        if (accelerometerSensor != null)  
            sensorManager.registerListener(this, accelerometerSensor, SensorManager.SENSOR_DELAY_GAME);  
    }  
  
    public void onStop() {  
        super.onStop();  
        if (sensorManager != null)  
            sensorManager.unregisterListener(this);  
    }  
}
```

Sensor Programming

❖ onAccuracyChanged(), onSensorChanged()

```
public void onAccuracyChanged(Sensor sensor, int accuracy) {  
}  
  
@Override  
public void onSensorChanged(SensorEvent event) {  
    if (event.sensor.getType() == Sensor.TYPE_ACCELEROMETER) {  
        long currentTime = System.currentTimeMillis();  
        long gabOfTime = (currentTime - lastTime);  
        if (gabOfTime > 100) {  
            lastTime = currentTime;  
            x = event.values[SensorManager.DATA_X];  
            y = event.values[SensorManager.DATA_Y];  
            z = event.values[SensorManager.DATA_Z];  
            speed = Math.abs(x + y + z - lastX - lastY - lastZ) /  
                gabOfTime * 10000;  
            if (speed > SHAKE_THRESHOLD) {  
                // 이벤트 발생!!  
            }  
            lastX = event.values[DATA_X];  
            lastY = event.values[DATA_Y];  
            lastZ = event.values[DATA_Z];  
        }  
    }  
}
```

Sensor Programming

❖ 예 : Bouncing Ball



<http://xiangchen.wordpress.com/2011/12/17/an-android-accelerometer-example/>