

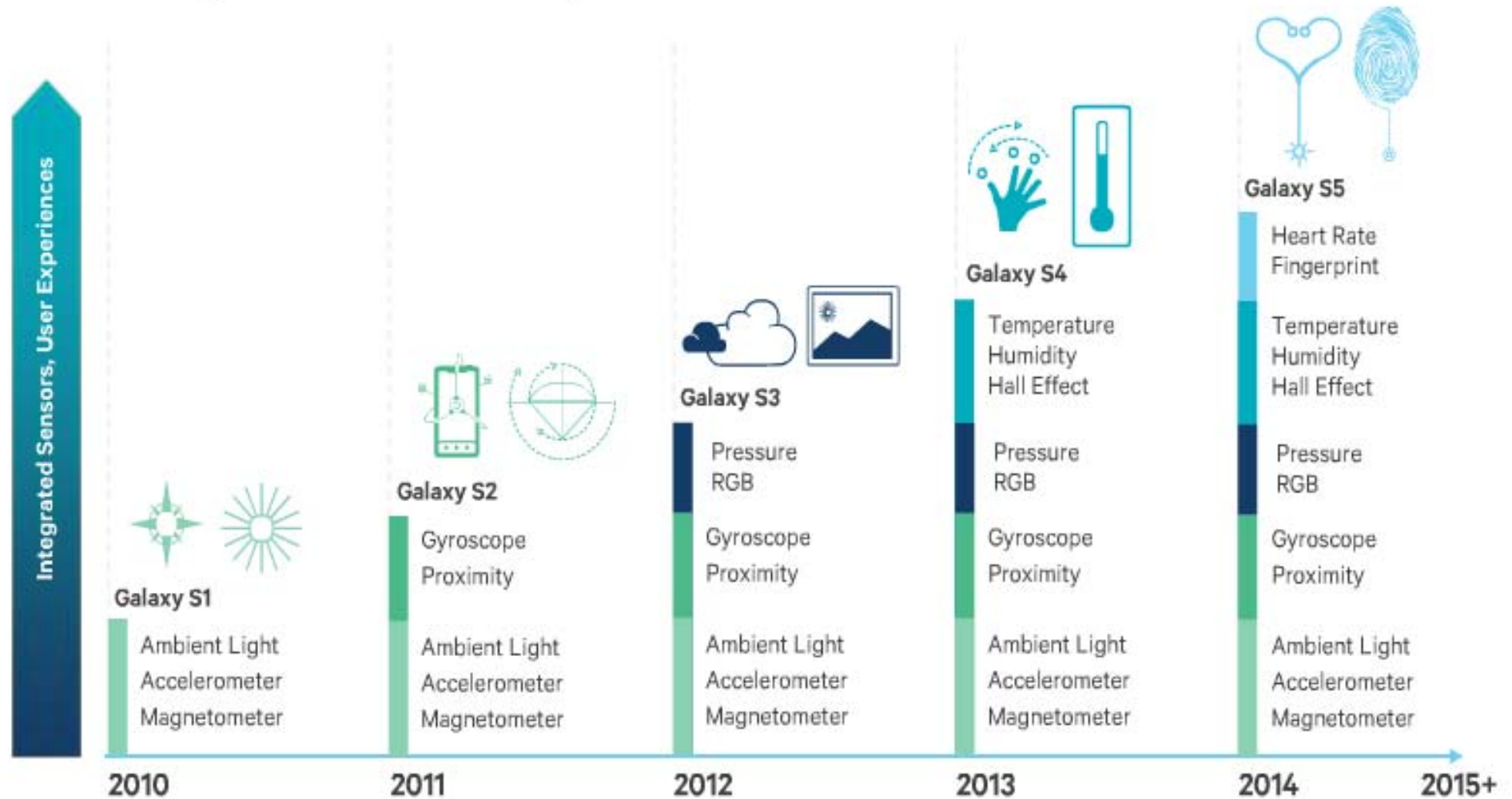
# 모바일 시스템 프로그래밍

## 03 Sensor Manager

2017 1학기

강승우

# Sensor growth in smartphones

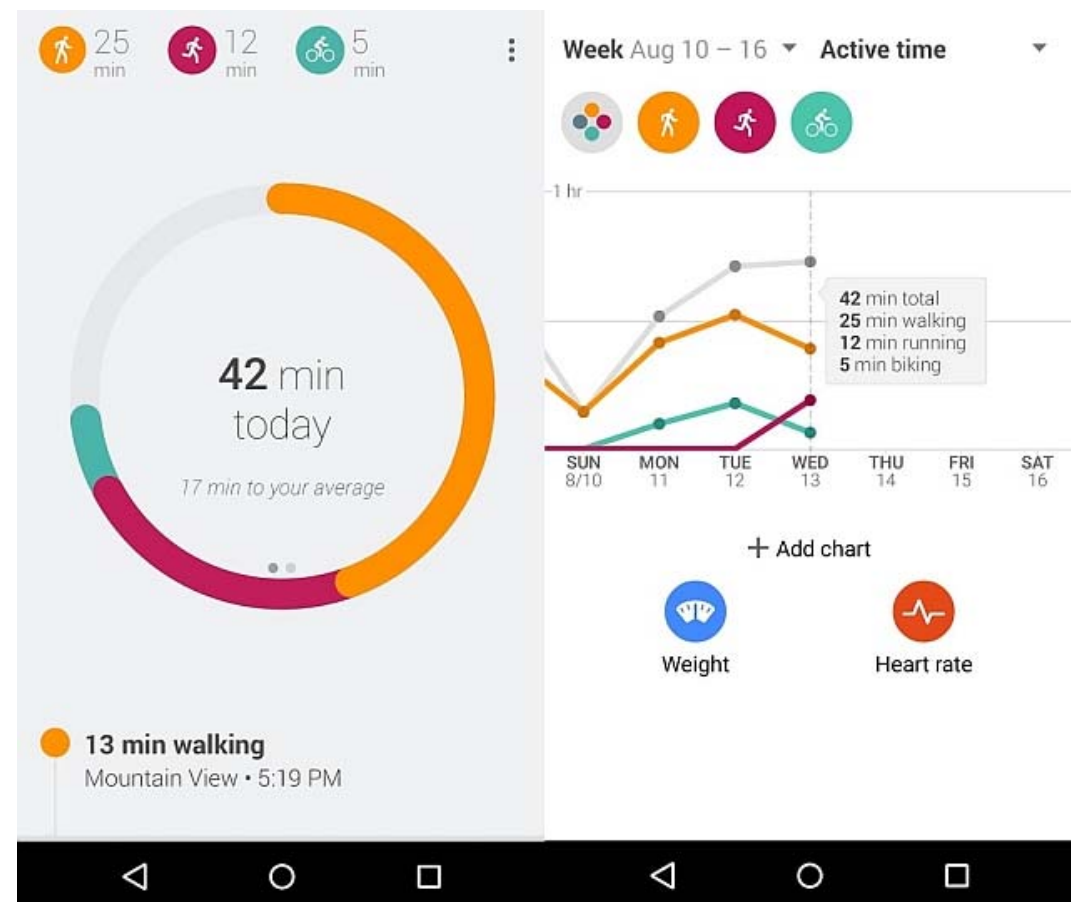


# 센서 사용 예 - Activity tracking

Moves



Google Fit



# 안드로이드 센서 사용 및 관리

- 관련 주요 class 및 interface
  - SensorManager
  - Sensor
  - SensorEvent
  - SensorEventListener
- 센서 관련 API의 이용
  - 사용 가능한 센서 식별, 센서의 특성/기능 파악
  - 센서 이벤트 모니터링
    - Sensor name, timestamp (이벤트 발생 시간), accuracy, raw sensor data

[https://developer.android.com/guide/topics/sensors/sensors\\_overview.html?hl=ko](https://developer.android.com/guide/topics/sensors/sensors_overview.html?hl=ko)

# SensorManager 클래스

- 안드로이드 플랫폼에서 센서를 관리하고 이용할 수 있도록 제공되는 시스템 서비스
  - 센서에 대한 정보를 제공하고, 특정 타입의 센서를 액세스 할 수 있게 해줌
  - 센서 이벤트 리스너 등록을 통해 센서 데이터를 받을 수 있음

사용 예:

```
SensorManager sm =  
(SensorManager)getSystemService(Context.SENSOR_SERVICE);
```

<https://developer.android.com/reference/android/hardware/SensorManager.html?hl=ko>

# Sensor 클래스

- SensorManager 클래스의 메소드를 이용하여 사용하고자 하는 센서 객체를 얻을 수 있음
    - getDefaultSensor(int type) 메소드 이용
- 사용 예: `Sensor accel = sm.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);`

- Sensor 클래스 메소드

`float getMaximumRange()`

센서 값의 최대 범위

`int getMinDelay()`

두 개의 센서 이벤트 사이의 최소 딜레이 (ms 단위) 혹은 0 (측정 데이터에 변화가 있을 때만 값을 주는 경우)

`String getName()`

`float getPower()`

사용 중에 소모한 파워 (mA 단위)

`String getStringType()`

`int getType()`

`String getVendor()`

`int getVersion()`

<https://developer.android.com/reference/android/hardware/Sensor.html?hl=ko>

# 센서 타입

- **Sensor** 클래스에 정의된 상수

(하드웨어 센서 / 소프트웨어 센서)

상수	설명	단위
<a href="#">TYPE_ALL</a>	모든 센서	
<a href="#">TYPE_ACCELEROMETER</a>	가속도 센서	$m/sec^2$
<a href="#">TYPE_GRAVITY</a>	중력 센서	$m/sec^2$
<a href="#">TYPE_GYROSCOPE</a>	자이로스코프 센서	
<a href="#">TYPE_ORIENTATION</a>	방향 센서	도(degree)
<a href="#">TYPE_LIGHT</a>	조도 센서	lux
<a href="#">TYPE_LINEAR_ACCELERATION</a>	선형 가속 센서	
<a href="#">TYPE_MAGNETIC_FIELD</a>	자기장 센서	마이크로테슬라(uT)
<a href="#">TYPE_PRESSURE</a>	압력 센서	
<a href="#">TYPE_PROXIMITY</a>	근접 센서	미터
<a href="#">TYPE_ROTATION_VECTOR</a>	회전 벡터 센서	
<a href="#">TYPE_TEMPERATURE</a>	온도 센서	

[https://developer.android.com/guide/topics/sensors/sensors\\_motion.html?hl=ko](https://developer.android.com/guide/topics/sensors/sensors_motion.html?hl=ko)

# 예제: 센서 리스트 보기

- 모든 타입의 센서 정보를 화면에 표시
- 예제 프로젝트 이름: **MSP01\_SensorList**

전체 센서 수: 27

0 name: MPU6515 Accelerometer

power: 0.4

resolution: 5.950928E-4

range: 19.613297

vendor: InvenSense

min delay: 5000

1 name: AK8963 Magnetometer

power: 5.0

resolution: 0.14953613

range: 4911.9995

vendor: AKM

min delay: 16666

2 name: AK8963 Magnetometer Uncalibrated

power: 5.0

resolution: 0.14953613

range: 4911.9995

vendor: AKM

min delay: 16666

3 name: MPU6515 Gyroscope

power: 3.2

resolution: 0.0010681152

range: 34.906586

vendor: InvenSense

min delay: 5000



@Override

```
protected void onCreate(Bundle savedInstanceState) {
```

```
    super.onCreate(savedInstanceState);
```

```
    setContentView(R.layout.activity_main);
```

```
    String list = "";
```

```
    // SensorManager 객체를 getSystemService 메소드를 통해 얻음
```

```
    SensorManager sm = (SensorManager)getSystemService(Context.SENSOR_SERVICE);
```

```
    // 모든 타입의 센서 목록을 얻음
```

```
    List<Sensor> sensors = sm.getSensorList(Sensor.TYPE_ALL);
```

```
    list += "전체 센서 수: " + sensors.size() + "\n";
```

```
    int i = 0;
```

```
    for(Sensor s: sensors) {
```

```
        list += "" + i++ + " name: " + s.getName() + "\n" + "power: " + s.getPower() + "\n"
```

```
            + "resolution: " + s.getResolution() + "\n" + "range: " + s.getMaximumRange() + "\n"
```

```
            + "vendor: " + s.getVendor() + "\n" + "min delay: " + s.getMinDelay() + "\n\n";
```

```
    }
```

```
    TextView text = (TextView)findViewById(R.id.text);
```

```
    // TextView에 텍스트 내용이 화면 크기를 넘어서 들어갈 때 스크롤 가능하게 만들기 위한 메소드 호출
```

```
    text.setMovementMethod(new ScrollingMovementMethod());
```

```
    text.setText(list);
```

```
}
```

# SensorEventListener 인터페이스

- 센서 값이 변경됐을 때 SensorManager로부터 이벤트 형태로 전달 받을 수 있도록 해줌
- SensorEventListener 등록
  - SensorManager 클래스에 정의된 메소드 이용  
`boolean registerListener(SensorEventListener listener, Sensor sensor, int samplingPeriodUs)`
    - listener: SensorEventListener 객체
    - sensor: 데이터를 받고 싶은 센서 객체 (앞에서 언급한 것처럼 SensorManager의 `getDefaultSensor` 메소드를 통해 얻음)
    - samplingPeriodUs: 데이터를 전달 받고자 하는 속도. SensorManager 클래스에 다음 상수가 정의되어 있음. 일정한 속도를 보장하지는 않음. 기기마다 다를 수 있음
      - SENSOR\_DELAY\_NORMAL
      - SENSOR\_DELAY\_UI
      - SENSOR\_DELAY\_GAME
      - SENSOR\_DELAY\_FASTEST

# SensorEventListener 인터페이스

- SensorEventListener 해제

  - `public void unregisterListener(SensorEventListener listener, Sensor sensor)`

  - `public void unregisterListener(SensorEventListener listener)`

    - 센서 데이터 업데이트가 더 이상 필요하지 않을 때는 반드시 해제를 해주어야 함

- SensorEventListener 인터페이스 구현

  - 아래 두 개의 메소드를 구현해야 함

  - `public void onAccuracyChanged(Sensor sensor, int accuracy)`

    - ✓ 등록된 센서의 정확도가 변경됐을 때 호출

  - `public void onSensorChanged(SensorEvent event)`

    - ✓ 센서 값이 변했을 때 호출. SensorEvent 객체로 데이터를 전달 받게 됨

<https://developer.android.com/reference/android/hardware/SensorEventListener.html?hl=ko>

# SensorEvent 클래스

- 센서로부터 센서 데이터가 업데이트 되었다는 이벤트를 나타냄
- 센서 타입, 시간(timestamp), 정확도, 센서 데이터와 같은 정보를 담고 있음

public int accuracy

✓ 이 이벤트의 정확도

public Sensor sensor

✓ 이 이벤트를 발생한 센서

public long timestamp

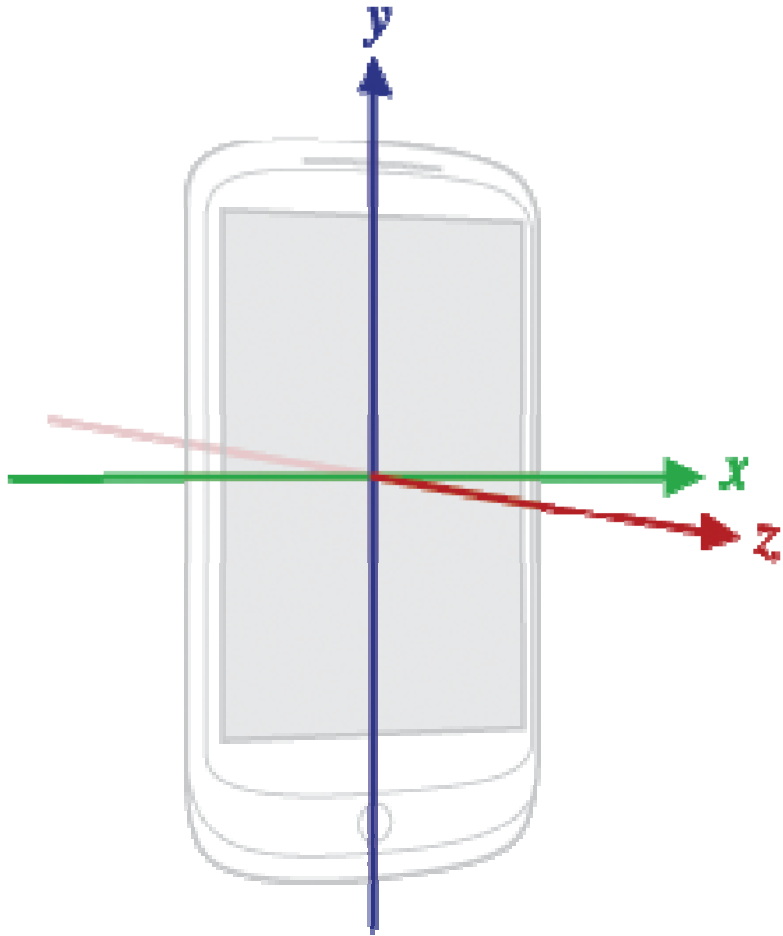
✓ 이벤트가 발생한 시각 (나노초 단위)

public final float[] values

✓ 센서 데이터를 담고 있는 배열. 센서 타입에 따라 길이나 그 내용은 달라짐

<https://developer.android.com/reference/android/hardware/SensorEvent.html?hl=ko>

# SensorEvent 좌표계



- SensorEvent API에서 사용하는 좌표계
  - X축: 수평 방향, 오른쪽이 양의 방향
  - Y축: 수직 방향, 위쪽이 양의 방향
  - Z축: 화면의 전방 수직 방향이 양의 방향

# 센서 데이터 (타입에 따른 values 배열 값)

- Sensor.TYPE\_LIGHT:
  - values[0]: 조도 (lux)
- Sensor.TYPE\_PRESSURE:
  - values[0]: 대기압 (hPa)
- Sensor.TYPE\_PROXIMITY:
  - values[0]: 근접도 (cm)
- Sensor.TYPE\_ORIENTATION:
  - values[0]: 방위각(Azimuth) 북쪽과 y축 사이 각도 (0 to 359). z축 중심 회전각 0=North, 90=East, 180=South, 270=West
  - values[1]: 피치(Pitch) x축 중심 회전각 (-180 to 180)
  - values[2]: 롤(Roll) y축 중심 회전각 (-90 to 90).

# 예제: 방향 센서 데이터 얻기

방향 센서값

방위각: 332.24887

피치: -29.534147

롤: 7.6935744

- 예제 프로젝트 이름: **MSP02OrientationSensor**

# 예제: 방향 센서 데이터 얻기

```
public class MainActivity extends AppCompatActivity implements SensorEventListener {  
    private SensorManager mSensorManger;  
    private Sensor mOrientation;  
    TextView mText;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        mSensorManger = (SensorManager) getSystemService(Context.SENSOR_SERVICE);  
        mOrientation = mSensorManger.getDefaultSensor(Sensor.TYPE_ORIENTATION);  
        mText = (TextView) findViewById(R.id.text);  
    }  
}
```



# 예제: 방향 센서 데이터 얻기

- SensorEventListener 등록 및 해제

@Override

```
protected void onResume() {
```

```
    super.onResume();
```

```
    // SensorEventListener 등록
```

```
    mSensorManger.registerListener(this, mOrientation, SensorManager.SENSOR_DELAY_UI);
```

```
}
```

@Override

```
protected void onPause() {
```

```
    super.onPause();
```

```
    // SensorEventListener 해제
```

```
    mSensorManger.unregisterListener(this);
```

```
}
```

# 예제: 방향 센서 데이터 얻기

- SensorEventListener 구현 - 2개 메소드  
onSensorChanged 메소드에서 입력으로 받는 SensorEvent 객체를 통해서 센서 데이터를 얻음

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

```
public void onSensorChanged(SensorEvent event) {  
    if(event.sensor.getType() == Sensor.TYPE_ORIENTATION) {  
        mText.setText("방향 센서값\n\n방위각: " + event.values[0]  
            + "\n피치: " + event.values[1] + "\n롤: " + event.values[2]);  
    }  
}
```



# 실습: 가속도 데이터 얻기

- 가속도
  - 속도의 변화율
  - 기기의 방향 혹은 움직임을 알아내는데 사용
  - 3차원(x, y, z축)으로 표현 – x축 가속도, y축 가속도, z축 가속도
  - 중력 가속도
- 센서 타입
  - `Sensor.TYPE_ACCELEROMETER`
  - `Sensor.TYPE_LINEAR_ACCELERATION`
    - 중력 가속도를 제외한 기기에 가해지는 힘에 의한 가속도 값

# 실습: 가속도 데이터 얻기

- 두 가지 타입의 가속도 데이터를 비교해보자
  - `Sensor.TYPE_ACCELEROMETER`
  - `Sensor.TYPE_LINEAR_ACCELERATION`

Accel: x: 0.178 y: -0.271 z: 9.643

Linear: x: -0.016 y: 0.000 z: 0.012