**Read Sensor Information Android (Gyroscope, Magnetic, Orientation, Accelerometer)**  
<https://www.vogella.com/tutorials/AndroidSensor/article.html>

**Android Sensor - Tutorial**

TABLE OF CONTENTS

[1. Android Sensors](https://www.vogella.com/tutorials/AndroidSensor/article.html" \l "android-sensors)

[2. Tutorial: Accelerometer](https://www.vogella.com/tutorials/AndroidSensor/article.html" \l "tutorial-accelerometer)

[3. Tutorial: Building a compass](https://www.vogella.com/tutorials/AndroidSensor/article.html" \l "tutorial-building-a-compass)

[4. Links and Literature](https://www.vogella.com/tutorials/AndroidSensor/article.html" \l "links-and-literature)

[5. vogella training and consulting support](https://www.vogella.com/tutorials/AndroidSensor/article.html" \l "vogella-training-and-consulting-support)

[Appendix A: Copyright, License and Source code](https://www.vogella.com/tutorials/AndroidSensor/article.html" \l "copyright-license-and-source-code)

Android Sensor. This tutorial describes how to use the Android Sensor manager. The tutorial is based on Eclipse 3.6, Java 1.6 and Android 2.3.3 (Gingerbread).

**[1. Android Sensors](https://www.vogella.com/tutorials/AndroidSensor/article.html" \l "android-sensors)**

**[1.1. SensorManager](https://www.vogella.com/tutorials/AndroidSensor/article.html" \l "sensormanager)**

Android Sensors supports several sensors via the SensorManager, for example the accelerometer. Unfortunately, you cannot test the accelerometer on the Android emulator.

You can access a SensorManager via getSystemService(SENSOR\_SERVICE). The Sensor class defines several constants for accessing the different sensors.

Sensor.TYPE\_GYROSCOPE

Sensor.TYPE\_MAGNETIC\_FIELD

Sensor.TYPE\_ORIENTATION

Sensor.TYPE\_ACCELEROMETER

You can access the sensor via the sensorManager.getDefaultSensor() method, which takes the sensor type and the delay defined as constants on SensorManager as parameters.

**[1.2. Sensor listener](https://www.vogella.com/tutorials/AndroidSensor/article.html" \l "sensor-listener)**

Once you acquired a sensor, you can register a SensorEventListener object on it. This listener will get informed, if the sensor data changes.

To avoid the unnecessary usage of battery power, you can register your listener in the onResume() method and de-register it in the onPause() method.

**[2. Tutorial: Accelerometer](https://www.vogella.com/tutorials/AndroidSensor/article.html" \l "tutorial-accelerometer)**

We will build an application which will change its background color if it is shuffled. Create a new Android project called de.vogella.android.sensor with an activity called SensorTestActivity.

Change your layout file to the following code.

***<?xml version="1.0" encoding="utf-8"?>*<LinearLayout** xmlns:android=**"http://schemas.android.com/apk/res/android"**

android:layout\_width=**"match\_parent"**

android:layout\_height=**"match\_parent"**

android:orientation=**"vertical"** **>**

**<TextView**

android:id=**"@+id/textView"**

android:layout\_width=**"match\_parent"**

android:layout\_height=**"match\_parent"**

android:text=**"Shake to get a toast and to switch color"** **/>**

**</LinearLayout>**

Change your activity class to the following code.

**package** de**.**vogella**.**android**.**sensor**;**

**import** **android.app.Activity;import** **android.graphics.Color;import** **android.hardware.Sensor;import** **android.hardware.SensorEvent;import** **android.hardware.SensorEventListener;import** **android.hardware.SensorManager;import** **android.os.Bundle;import** **android.view.View;import** **android.view.Window;import** **android.view.WindowManager;import** **android.widget.Toast;**

**public** **class** **SensorTestActivity** **extends** Activity **implements** SensorEventListener **{**

**private** SensorManager sensorManager**;**

**private** **boolean** color **=** **false;**

**private** View view**;**

**private** **long** lastUpdate**;**

*/\*\* Called when the activity is first created. \*/*

**@Override**

**public** **void** onCreate**(**Bundle savedInstanceState**)** **{**

requestWindowFeature**(**Window**.**FEATURE\_NO\_TITLE**);**

getWindow**().**setFlags**(**WindowManager**.**LayoutParams**.**FLAG\_FULLSCREEN**,**

WindowManager**.**LayoutParams**.**FLAG\_FULLSCREEN**);**

**super.**onCreate**(**savedInstanceState**);**

setContentView**(**R**.**layout**.**main**);**

view **=** findViewById**(**R**.**id**.**textView**);**

view**.**setBackgroundColor**(**Color**.**GREEN**);**

sensorManager **=** **(**SensorManager**)** getSystemService**(**SENSOR\_SERVICE**);**

lastUpdate **=** System**.**currentTimeMillis**();**

**}**

**@Override**

**public** **void** onSensorChanged**(**SensorEvent event**)** **{**

**if** **(**event**.**sensor**.**getType**()** **==** Sensor**.**TYPE\_ACCELEROMETER**)** **{**

getAccelerometer**(**event**);**

**}**

**}**

**private** **void** getAccelerometer**(**SensorEvent event**)** **{**

**float[]** values **=** event**.**values**;**

*// Movement*

**float** x **=** values**[0];**

**float** y **=** values**[1];**

**float** z **=** values**[2];**

**float** accelationSquareRoot **=** **(**x **\*** x **+** y **\*** y **+** z **\*** z**)**

**/** **(**SensorManager**.**GRAVITY\_EARTH **\*** SensorManager**.**GRAVITY\_EARTH**);**

**long** actualTime **=** event**.**timestamp**;**

**if** **(**accelationSquareRoot **>=** **2)** *//*

**{**

**if** **(**actualTime **-** lastUpdate **<** **200)** **{**

**return;**

**}**

lastUpdate **=** actualTime**;**

Toast**.**makeText**(this,** **"Device was shuffed",** Toast**.**LENGTH\_SHORT**)**

**.**show**();**

**if** **(**color**)** **{**

view**.**setBackgroundColor**(**Color**.**GREEN**);**

**}** **else** **{**

view**.**setBackgroundColor**(**Color**.**RED**);**

**}**

color **=** **!**color**;**

**}**

**}**

**@Override**

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

**}**

**@Override**

**protected** **void** onResume**()** **{**

**super.**onResume**();**

*// register this class as a listener for the orientation and*

*// accelerometer sensors*

sensorManager**.**registerListener**(this,**

sensorManager**.**getDefaultSensor**(**Sensor**.**TYPE\_ACCELEROMETER**),**

SensorManager**.**SENSOR\_DELAY\_NORMAL**);**

**}**

**@Override**

**protected** **void** onPause**()** **{**

*// unregister listener*

**super.**onPause**();**

sensorManager**.**unregisterListener**(this);**

**}}**

**[3. Tutorial: Building a compass](https://www.vogella.com/tutorials/AndroidSensor/article.html" \l "tutorial-building-a-compass)**

Create a new Android project called de.vogella.android.sensor.compass with an activity called MainActivity.

Create the following custom View class.

**package** de**.**vogella**.**android**.**sensor**.**compass**;**

**import** **android.content.Context;import** **android.graphics.Canvas;import** **android.graphics.Color;import** **android.graphics.Paint;import** **android.view.View;**

**public** **class** **MyCompassView** **extends** View **{**

**private** Paint paint**;**

**private** **float** position **=** **0;**

**public** MyCompassView**(**Context context**)** **{**

**super(**context**);**

init**();**

**}**

**private** **void** init**()** **{**

paint **=** **new** Paint**();**

paint**.**setAntiAlias**(true);**

paint**.**setStrokeWidth**(2);**

paint**.**setTextSize**(25);**

paint**.**setStyle**(**Paint**.**Style**.**STROKE**);**

paint**.**setColor**(**Color**.**WHITE**);**

**}**

**@Override**

**protected** **void** onDraw**(**Canvas canvas**)** **{**

**int** xPoint **=** getMeasuredWidth**()** **/** **2;**

**int** yPoint **=** getMeasuredHeight**()** **/** **2;**

**float** radius **=** **(float)** **(**Math**.**max**(**xPoint**,** yPoint**)** **\*** **0.6);**

canvas**.**drawCircle**(**xPoint**,** yPoint**,** radius**,** paint**);**

canvas**.**drawRect**(0,** **0,** getMeasuredWidth**(),** getMeasuredHeight**(),** paint**);**

*// 3.143 is a good approximation for the circle*

canvas**.**drawLine**(**

xPoint**,**

yPoint**,**

**(float)** **(**xPoint **+** radius

**\*** Math**.**sin**((double)** **(-**position**)** **/** **180** **\*** **3.143)),**

**(float)** **(**yPoint **-** radius

**\*** Math**.**cos**((double)** **(-**position**)** **/** **180** **\*** **3.143)),** paint**);**

canvas**.**drawText**(**String**.**valueOf**(**position**),** xPoint**,** yPoint**,** paint**);**

**}**

**public** **void** updateData**(float** position**)** **{**

**this.**position **=** position**;**

invalidate**();**

**}**

**}**

Change the coding of your activity.

**package** de**.**vogella**.**android**.**sensor**.**compass**;**

**import** **android.app.Activity;import** **android.content.Context;import** **android.hardware.Sensor;import** **android.hardware.SensorEvent;import** **android.hardware.SensorEventListener;import** **android.hardware.SensorManager;import** **android.os.Bundle;import** **android.util.Log;import** **android.widget.Toast;**

**public** **class** **MainActivity** **extends** Activity **{**

**private** **static** SensorManager sensorService**;**

**private** MyCompassView compassView**;**

**private** Sensor sensor**;**

*/\*\* Called when the activity is first created. \*/*

**@Override**

**public** **void** onCreate**(**Bundle savedInstanceState**)** **{**

**super.**onCreate**(**savedInstanceState**);**

compassView **=** **new** MyCompassView**(this);**

setContentView**(**compassView**);**

sensorService **=** **(**SensorManager**)** getSystemService**(**Context**.**SENSOR\_SERVICE**);**

sensor **=** sensorService**.**getDefaultSensor**(**Sensor**.**TYPE\_ORIENTATION**);**

**if** **(**sensor **!=** **null)** **{**

sensorService**.**registerListener**(**mySensorEventListener**,** sensor**,**

SensorManager**.**SENSOR\_DELAY\_NORMAL**);**

Log**.**i**("Compass MainActivity",** **"Registerered for ORIENTATION Sensor");**

**}** **else** **{**

Log**.**e**("Compass MainActivity",** **"Registerered for ORIENTATION Sensor");**

Toast**.**makeText**(this,** **"ORIENTATION Sensor not found",**

Toast**.**LENGTH\_LONG**).**show**();**

finish**();**

**}**

**}**

**private** SensorEventListener mySensorEventListener **=** **new** SensorEventListener**()** **{**

**@Override**

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

**}**

**@Override**

**public** **void** onSensorChanged**(**SensorEvent event**)** **{**

*// angle between the magnetic north direction*

*// 0=North, 90=East, 180=South, 270=West*

**float** azimuth **=** event**.**values**[0];**

compassView**.**updateData**(**azimuth**);**

**}**

**};**

**@Override**

**protected** **void** onDestroy**()** **{**

**super.**onDestroy**();**

**if** **(**sensor **!=** **null)** **{**

sensorService**.**unregisterListener**(**mySensorEventListener**);**

**}**

**}**

**}**