

COMP4336/9336 Lab 4

Sensors 1

Lab Objectives

- Learn how to determine at runtime which sensors are available on a device.
- Learn how to work with accelerometer of mobile phones.
- Learn how the orientation/position of mobile phone can be detected via accelerometer data.

Preparation

- **Background:**

Most Android-based devices have built-in sensors that measure motion, orientation, and various environmental conditions such as pressure, temperature and humidity. These sensors are capable of providing raw data with high precision and accuracy. As accelerometer, gyroscope and magnetometer sensors have been covered in the lecture, we focus on these sensors. Android developer has different classes and methods to deal with these sensors. Please read these links before come to the lab:

http://developer.android.com/guide/topics/sensors/sensors_overview.html

http://developer.android.com/guide/topics/sensors/sensors_motion.html

- **Some useful Android classes and methods to working with sensors:**

- *Class SensorManager*: This class lets you to access the device's sensors. You can get an instance of this class by calling `Context.getSystemService()` with the argument `SENSOR_SERVICE`.

Note: you can use *getSensorList* method from this class to get the list of available sensors of the mobile phone. Make multiple calls to get sensors of different types or use `Sensor.TYPE_ALL` to get all the sensors. It would be useful for task1.

More details:

<http://developer.android.com/reference/android/hardware/SensorManager.html>

- *Class SensorEvent*: This class represents a Sensor event and holds information such as the sensor's type, the time-stamp, accuracy and of course the sensor's data. It has 4 properties including: `int accuracy`, `Sensor sensor`, `long timestamp` and final `float[] values`.

Note : You can use the *time-stamp* to do integration overtime.

Note: The raw data of measurement saves in the *values* filed.

More details:

<http://developer.android.com/reference/android/hardware/SensorEvent.html>

- *Class Sensor*: Class representing a sensor which could be: `TYPE_ACCELEROMETER`, `TYPE_PROXIMITY`, `TYPE_AMBIENT_TEMPERATURE`, `TYPE_GAME_ROTATION_VECTOR`, `TYPE_GRAVITY`, `TYPE_GYROSCOPE`,

The method *getMinDelay()* from this class can give you the minimum possible delay between two events in microsecond or zero if this sensor only returns a value when the data it's measuring changes. It has some other useful methods such as *getResolution()*, *getPower()*.

More details:

<http://developer.android.com/reference/android/hardware/Sensor.html>

- ***Some useful notes and codes:***

- To get access to the specific sensor, you can use this code:

```
private SensorManager mSensorManager;  
private Sensor mSensor;  
...  
mSensorManager = (SensorManager) getSystemService(  
Context.SENSOR_SERVICE);  
  
mSensor =  
mSensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
```

- Android support various range of motion sensors, of course if the physical sensor would be available on the device. In the above code, the type of sensor could be one of the following parameter: TYPE_ACCELEROMETER, TYPE_GRAVITY, TYPE_GYROSCOPE, TYPE_LINEAR_ACCELERATION, TYPE_ROTATION_VECTOR. To get all available sensors you can use this line of code :

```
List< sensor> sensorList = mSensorManager.getSensorList(Sensor.TYPE_ALL);
```

- You can access to the raw data of measured parameter via sensors by reading *values* property of *SensorEvent* class. For example if you *override* *onSensorChanged* method, such as below code, you can get the raw values of accelerometer continuously:

```
@Override  
public void onSensorChanged(SensorEvent event)  
{  
    double x = event.values[0];  
    double y = event.values[1];  
    double z = event.values[2];  
}
```

Note: In order to override this method, your *MainActivity* class has to *implements* the *SensorEventListener* class.

Lab Tasks

Task1: Check the availability of sensors in the mobile devices (25 Marks)

Develop a simple android application to give you all available sensors on the mobile phone. It has to display this information for each sensor: Name, Vendor, Version, MaximumRange and MinDelay like Figure1 (task1).

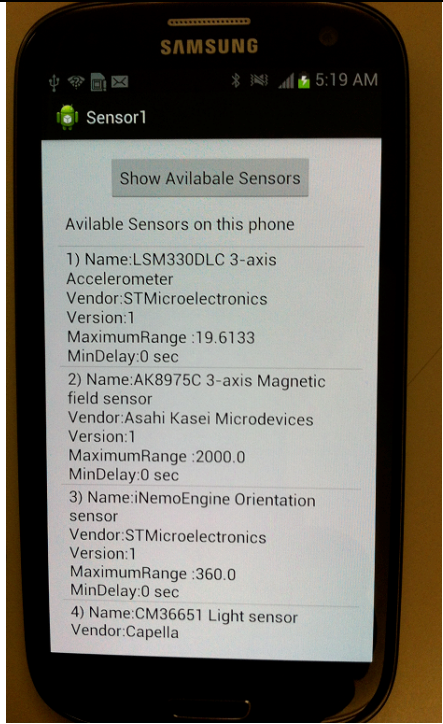
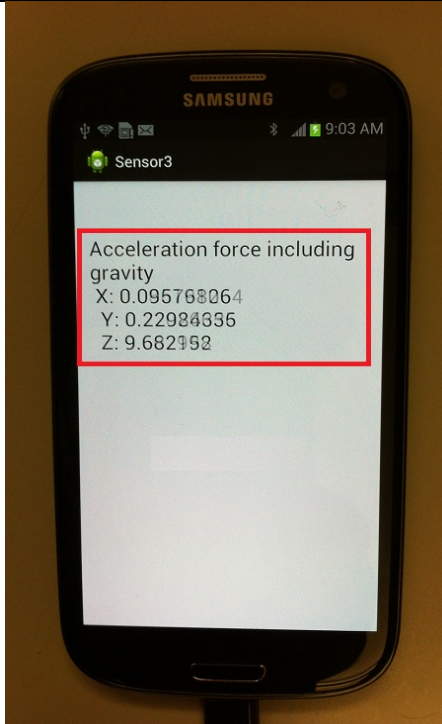
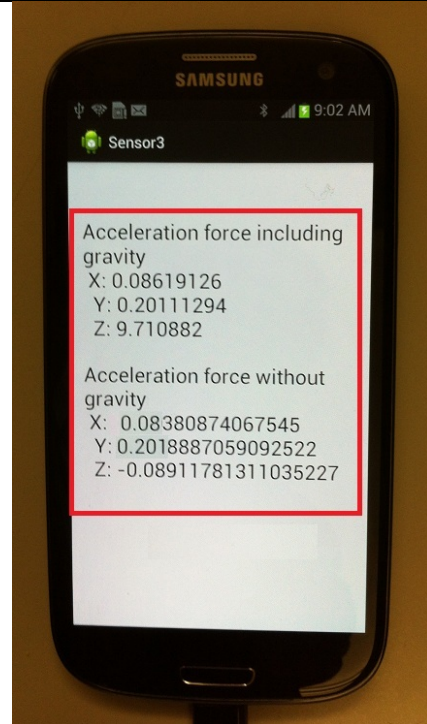
Note: You can use different methods of *Sensor* class to get wanted information for each type of sensor.

Task2: Working with accelerometer 1

Read the accelerometer data continuously. (25 Marks)

Develop an Android application which would be able to give you the acceleration force along the x, y and z axis and display the values on the screen. In the next step, filter out the gravity and show the result in another text view such as you can see in the Figure. 2

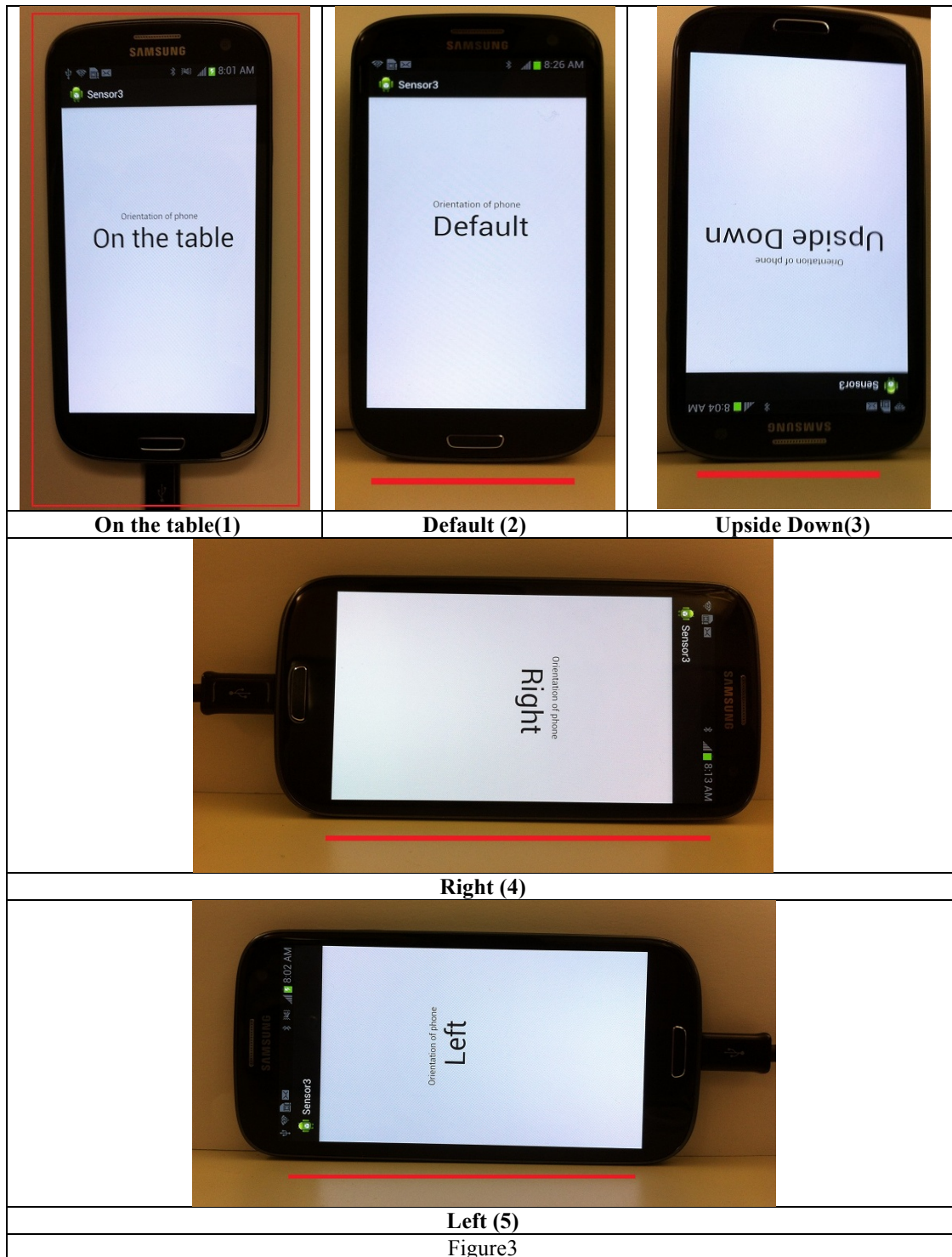
Note: You have to filtering out gravity when after reading raw data and before integrating. To filter out the gravity just refer to the lecture slides.

		
Task1	Task2 (1)	Task2 (2)
Figure1		

Task3: Working with accelerometer 2:

Detect the orientation/position of mobile phone via accelerometer. (50 Marks)

Develop an Android application that would be able to detect the orientation/position of the mobile screen and show a text constantly reporting the current orientation of mobile phone. It should be able to detect these five positions including: On the table, default, upside down, right and left such as it can be seen in Figure3.



Note: The red line indicates the surface of table.