Lab 9 Android Pedometer

1. Purpose

(1) By programming an Android pedometer program using mobile platform,we learn to familiarize the calling methods of mobile platform sensors.

(2) Simple mobile apps' design and implementation.

2. Contents

(1) Acceleration sensor's application

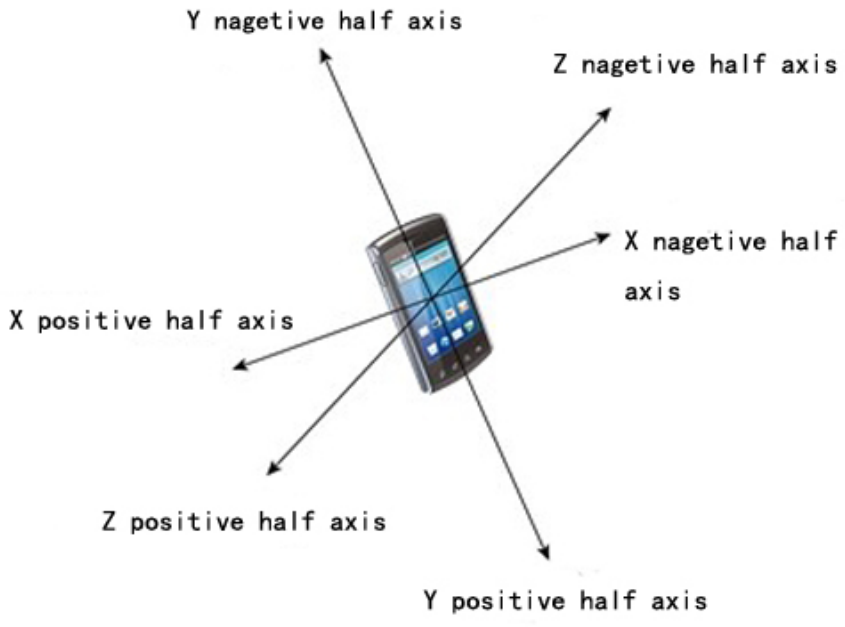
By using a compiler like Eclipse, write an Android program which could realize the function listed below.

1st Use the acceleration sensor to measure the mobile's acceleration in the X,Y and Z axis directions and the scalar value of the total acceleration.

2nd Display the results on the screen.

3rd Record the results in the specified file.

The relative position of the mobile's X,Y and Z axis derections is showed in the picture below



Reference Code:

```
package com.practice.cos;

import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
```

```
import java.io.RandomAccessFile;
import java.text.DecimalFormat;

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.os.Environment;
import android.util.Log;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.TextView;

public class PracticeActivity extends Activity implements SensorEventListener,
OnClickListener {
    /** Called when the activity is first created. */

  //Create a LOG label
  private Button mWriteButton, mStopButton;
  private boolean doWrite = false;
  private SensorManager sm;
  private float lowX = 0, lowY = 0, lowZ = 0;
  private final float FILTERING_VALAUE = 0.1f;
  private TextView AT,ACT;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
```

```
            AT = (TextView)findViewById(R.id.AT);
            ACT = (TextView)findViewById(R.id.onAccuracyChanged);

            //Create a SensorManager to get the system's sensor service
            sm =
(SensorManager)getSystemService(Context.SENSOR_SERVICE);
            /*
             *Using the most common method to register an event
             * Parameter1 ： SensorEventListener      detectophone
             * Parameter2 ： Sensor      one service could have several Sensor
realizations.Here,We use getDefaultSensor to get the defaulted Sensor
             * Parameter3 ： Mode    We can choose the refresh frequency of the
data change
             * */
            //    Register the acceleration sensor
            sm.registerListener(this,
                    sm.getDefaultSensor(Sensor.TYPE_ACCELEROMETER),
                    SensorManager.SENSOR_DELAY_FASTEST);//High
sampling rate；.SENSOR_DELAY_NORMAL means a lower sampling rate
            try {
             FileOutputStream fout = openFileOutput("acc.txt",
Context.MODE_PRIVATE);
             fout.close();
            } catch (IOException e) {
             e.printStackTrace();
            }
            mWriteButton = (Button) findViewById(R.id.Button_Write);
            mWriteButton.setOnClickListener(this);
            mStopButton = (Button) findViewById(R.id.Button_Stop);
            mStopButton.setOnClickListener(this);
        }

        public void onPause(){
          super.onPause();
```

```
    }


  public void onClick(View v) {
      if (v.getId() == R.id.Button_Write) {
          doWrite = true;
      }
      if (v.getId() == R.id.Button_Stop) {
          doWrite = false;
      }
  }

  public void onAccuracyChanged(Sensor sensor, int accuracy) {
      ACT.setText("onAccuracyChanged is detonated");
  }

  public void onSensorChanged(SensorEvent event) {
      String message = new String();
      if(event.sensor.getType() == Sensor.TYPE_ACCELEROMETER) {

          float X = event.values[0];
          float Y = event.values[1];
          float Z = event.values[2];

          //Low-Pass Filter
           lowX = X * FILTERING_VALAUE + lowX * (1.0f -
FILTERING_VALAUE);
           lowY = Y * FILTERING_VALAUE + lowY * (1.0f -
FILTERING_VALAUE);
           lowZ = Z * FILTERING_VALAUE + lowZ * (1.0f -
FILTERING_VALAUE);

          //High-pass filter
```

```
                float highX    = X -    lowX;
                float highY    = Y -    lowY;
                float highZ    = Z -    lowZ;
            double highA = Math.sqrt(highX * highX + highY * highY + highZ
* highZ);

            DecimalFormat df = new DecimalFormat("#,##0.000");

            message = df.format(highX) + "    ";
            message += df.format(highY) + "    ";
            message += df.format(highZ) + "    ";
            message += df.format(highA) + "\n";

            AT.setText(message + "\n");
            if (doWrite) {
                write2file(message);
            }
        }
    }

    private void write2file(String a){

        try {

                File file = new File("/sdcard/acc.txt");//write the result
into/sdcard/acc.txt
                if (!file.exists()){
                    file.createNewFile();}

            // Open a random access file stream for reading and writing
            RandomAccessFile randomFile = new
RandomAccessFile("/sdcard/acc.txt", "rw");
            // The length of the file (the number of bytes)
            long fileLength = randomFile.length();
```

```
                // Move the file pointer to the end of the file
                randomFile.seek(fileLength);
                 randomFile.writeBytes(a);
                  randomFile.close();
                } catch (IOException e) {
                      // TODO Auto-generated catch block
                      e.printStackTrace();
                }

   }

}
```

Layout files

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
      android:layout_width="fill_parent"
      android:layout_height="fill_parent"
      android:orientation="vertical" >

      <CheckedTextView
            android:id="@+id/AT"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/AT"
            android:textSize="20dp" />

      <CheckedTextView
            android:id="@+id/onAccuracyChanged"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/onAccuracyChanged"
            android:textSize="18dp" />
```

```
    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content" >

        <Button
            android:id="@+id/Button_Write"
            android:layout_width="90dp"
            android:layout_height="wrap_content"
            android:text="@string/Button_Write"
android:textSize="20dp"/>

        <Button
            android:id="@+id/Button_Stop"
            android:layout_width="90dp"
            android:layout_height="wrap_content"
            android:text="@string/Button_Stop" android:textSize="20dp"/>

    </LinearLayout>

</LinearLayout>
```

String

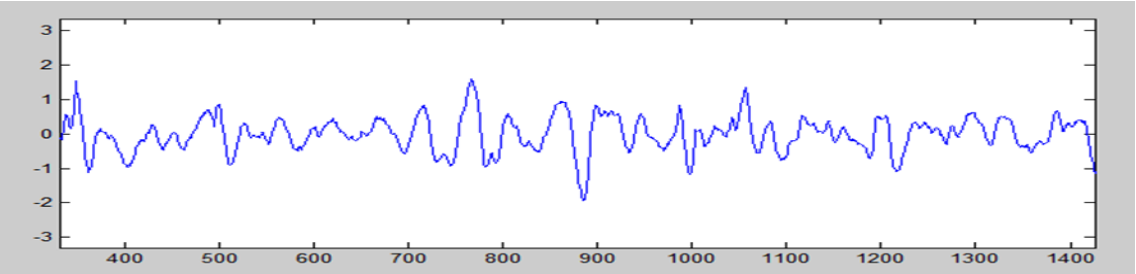
```
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <string name="hello">Hello World, PracticeActivity!</string>
    <string name="app_name">Practice</string>
    <string name="onAccuracyChanged">onAccuracyChanged doesn’t
detonate</string>
    <string name="AT">0</string>
    <string name="Button_Write">Write</string>
    <string name="Button_Stop">Stop</string>

</resources>
```

(2) Step counting method

There are many mobile step-counting methods but the best method is not found now. A typical acceleration scalar graph is showed below.



Self-designed methods are encouraged in this step and the display of the results on the screen in real time is required.

Hint

We can use the threshold filtering method. When the acceleration is higher or lower than the threshold, we can treat it as a step's beginning or ending. We also need to consider the waveform burr and the acceleration of walking is not a simple single-peak waveform.

3. Questions

(1) Why we need low pass filtering when we measure the acceleration ?
What are the differences made by the filtering? Make a contrast.

(2) Can we estimate the stride length using the acceleration oscillogram?

(3) What's the meaning of using super.onPause ?

4. Extended problems

(1) Suppose a driver uses an Android mobile.Use the GPS data and acceleration data to judge the switch of the moving and rest states and upload the status information to the server.

(2) Suppose there are a great number of drivers using the app mentioned in the 1st problem.How could we judge the traffic light's color using these information?Design and build a system to realize this function

Hint

You can refer the literature

Yiran Zhao, Yang Zhang, Tuo Yu, Tianyuan Liu, Xinbing Wang, Xiaohua Tian, Xue Liu, "CityDrive: A map-generating and speed-optimizing driving system," in proceedings of IEEE International Conference on Computer Communications, pp. 1986-1994, 2014.