

感知技术与应用

实 验 报 告

学 院 网安学院
年 级 大二
班 级 物联网工程
学 号 2212005
姓 名 秦泽斌

2024 年 4 月 12 日

目录

一、实验目标.....	1
二、实验内容.....	1
三、实验步骤.....	1
四、实验遇到的问题及其解决方法.....	10
五、实验结论.....	11

一、实验目的

本次实验的目的是让大家了解 Android 中加速度传感器的基本知识，掌握 Android 中加速度传感器的使用方法。

二、实验内容

(一)、获取设备中加速度传感器的值

1、编写布局文件 activity_main.xml。

2、编写程序文件 MainActivity.java。

(二)、按照实验的内容进行操作，掌握 Android 中光线传感器的使用方法等内容。

(三)、完成实验内容后，设计一个测试手机是否处于静止状态的 app，至少包含如下功能：

- 显示当前手机三个方向的加速度值；
- 通过加速度值判断手机当前是否处于静止状态。
- 记录用户步行的步数。

三、实验步骤及实验结果

(一) 编写布局文件 activity_main.xml。

```
</> activity_main.xml × MainActivity.java
1  <?xml version="1.0" encoding="utf-8"?>
2  <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:app="http://schemas.android.com/apk/res-auto"
4      xmlns:tools="http://schemas.android.com/tools"
5      android:layout_width="match_parent"
6      android:layout_height="match_parent"
7      tools:context=".MainActivity">
8      <LinearLayout
9          android:layout_width="wrap_content"
10         android:layout_height="wrap_content"
11         android:gravity="top"
12         android:layout_centerHorizontal="true"
13         android:orientation="vertical">
```

具体代码如下：

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="top"
        android:layout_centerHorizontal="true"
        android:orientation="vertical">
        <TextView
            android:id="@+id/acceleration_x"
            android:textSize="20sp"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="acceleration_x"/>
        <TextView
            android:id="@+id/acceleration_y"
            android:textSize="20sp"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="acceleration_y"/>
        <TextView
            android:id="@+id/acceleration_z"
            android:textSize="20sp"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="acceleration_z"/>

        <TextView
            android:id="@+id/ifmove"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:text="ifmove"
            android:textSize="30sp" />
    </LinearLayout>
</RelativeLayout>
```

```

        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:layout_centerHorizontal="true"
        android:orientation="vertical">
        <TextView
            android:id="@+id/acceleration_total"
            android:textSize="30sp"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="acceleration_total"/>

        <TextView
            android:id="@+id/stepCountTextView"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="当前步数: 0"
            android:textSize="18sp"
            android:layout_marginTop="16dp"
            android:layout_gravity="center_horizontal"/>

    </LinearLayout>

</RelativeLayout>

```

(二) 编写程序文件 MainActivity.java。

```

</> activity_main.xml    © MainActivity.java  ×
1      package com.example.accelerationsensors;
2      import androidx.appcompat.app.AppCompatActivity;
3      import android.hardware.Sensor;
4      import android.hardware.SensorEvent;
5      import android.hardware.SensorEventListener;
6      import android.hardware.SensorListener;
7      import android.hardware.SensorManager;
8      import android.os.Bundle;
9      import android.widget.TextView;
10
11
12  </> public class MainActivity extends AppCompatActivity {
13      TextView acceleration_x;//x方向的加速度
14      TextView acceleration_y;//y方向的加速度

```

具体代码如下：

```
package com.example.accelerationsensors;

import androidx.appcompat.app.AppCompatActivity;
import android.hardware.Sensor;
import android.hardware.SensorEvent;
import android.hardware.SensorEventListener;
import android.hardware.SensorListener;
import android.hardware.SensorManager;
import android.os.Bundle;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {
    TextView acceleration_x;//x 方向的加速度
    TextView acceleration_y;//y 方向的加速度
    TextView acceleration_z;//z 方向的加速度
    TextView acceleration_total;//显示总加速度
    TextView stepCountTextView; // 声明用于显示步数的 TextView 对象
    //显示运动情况
    TextView ifmove;

    SensorManager mySensorManager;//SensorManager 对象引用
    //SensorManagerSimulator mySensorManager;//声明
    SensorManagerSimulator 对象, 调试时用

    private int step;
    private double original_value;
    private double last_value;
    private double current_value;
    private boolean motionState=true; //是否处于运动状态
    private boolean processState=true; //是否已经开始计步

    @Override
    public void onCreate(Bundle savedInstanceState) { //重写 onCreate 方法
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);//设置当前的用户界面
        acceleration_x = (TextView) findViewById(R.id.acceleration_x);//
        得到 acceleration_x 的引用
        acceleration_y = (TextView) findViewById(R.id.acceleration_y);//
        得到 acceleration_y 的引用
        acceleration_z = (TextView) findViewById(R.id.acceleration_z);//
        得到 acceleration_z 的引用
        acceleration_total = (TextView)
        findViewById(R.id.acceleration_total);//得到 acceleration_total 的引用
```

```

        stepCountTextView =(TextView)
findViewById(R.id.stepCountTextView); //计步器的引用

        step=0;
        original_value=0;
        last_value =0;
        current_value =0;

        //设置一个用于判断是否运动的控件
        ifmove = (TextView) findViewById(R.id.ifmove); //得到 ifmove 的引用
        mySensorManager =
(SensorManager) getSystemService (SENSOR_SERVICE); //获得 SensorManager
    }
    private SensorEventListener mySensorListener = new
SensorEventListener() {
        @Override
        public void onSensorChanged(SensorEvent sensorEvent) {
            float total_acceleration;
            if (sensorEvent.sensor.getType() ==
Sensor.TYPE_ACCELEROMETER) {
                float[] values = sensorEvent.values;
                //通过开平方和得到总加速度
                total_acceleration = (float)
Math.sqrt(sensorEvent.values[0] * sensorEvent.values[0]
                    + sensorEvent.values[1] * sensorEvent.values[1]
                    + sensorEvent.values[2] * sensorEvent.values[2]);

                //设置加速度的显示情况
                acceleration_x.setText("X 方向上的加速度: " +
sensorEvent.values[0]);
                acceleration_y.setText("Y 方向上的加速度: " +
sensorEvent.values[1]);
                acceleration_z.setText("Z 方向上的加速度: " +
sensorEvent.values[2]);
                acceleration_total.setText("和加速度: " +
total_acceleration);

                //通过与本地 9.8 左右的加速度进行比较从而判断手机是否运动
                //因为实际本地加速度会在 9.8-9.9 之间浮动,通过物理知识可知小于 9.8
是在上升, 大于 9.9 是在下降
                if (total_acceleration < 9.9 && total_acceleration > 9.8)
{
                    ifmove.setText("静止中");
                } else if (total_acceleration >= 9.9) {

```

```

        ifmove.setText("正在下降");
    } else if (total_acceleration <= 9.8) {
        ifmove.setText("正在上升");
    }
}

if (sensorEvent.sensor.getType() ==
Sensor.TYPE_ACCELEROMETER) {

    double range = 5; //设置一个精度范围
    float[] value = sensorEvent.values;
    float current_value = (float)
Math.sqrt(sensorEvent.values[0] * sensorEvent.values[0]
        + sensorEvent.values[1] * sensorEvent.values[1]
        + sensorEvent.values[2] * sensorEvent.values[2]);
    //获取当前的总加速度

    //向上加速的状态
    if (motionState == true) {
        if (current_value >= last_value)
            last_value = current_value;
        else {
            //检测到一次峰值
            if (Math.abs(current_value - last_value) > range) {
                original_value = current_value;
                motionState = false;
            }
        }
    }

    //向下加速的状态
    if (motionState == false) {
        if (current_value <= last_value)
            last_value = current_value;
        else {
            //检测到一次峰值
            if (Math.abs(current_value - last_value) > range) {
                original_value = current_value;
                if (processState == true) {
                    step++; //检测到开始记录，步数加 1
                    if (processState == true) {
                        updateStepCount(); //更新读数
                    }
                }
            }
        }
        motionState = true;
    }
}

```



```

        }
    }
}

// 更新步数显示
private void updateStepCount() {
    // 可以将步数显示在一个 TextView 上
    // 这里假设你有一个名为 stepCountTextView 的 TextView 来显示步数
    stepCountTextView.setText("当前步数: " + step);
}

@Override
public void onAccuracyChanged(Sensor sensor, int i) {
}

};

@Override
protected void onResume() { // 重写的 onResume 方法
    mySensorManager.registerListener( // 注册监听
        mySensorListener, // 监听器 SensorListener 对象

mySensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER), // 传感器的
类型为加速度

        SensorManager.SENSOR_DELAY_UI // 传感器事件传递的频度
    );
    super.onResume();
}

@Override
protected void onPause() { // 重写 onPause 方法
    mySensorManager.unregisterListener(mySensorListener); // 取消注册
监听器
    super.onPause();
}
}
}

```

（三）运行程序并检查结果

程序运行结果如下：



经过检查，实验结果符合要求。

四、实验遇到的问题及其解决方法

在本次实验中，本人遇到的困难主要集中在计步器的设计上：

问题一：在选择合适的计步器算法上遇到了困难。

解决办法：因为对计步器算法的知识薄弱，本人一开始仅设计了一个简单的计步器，即检测加速度的大幅改变来检测步数，代码如下：

```
@Override
public void onSensorChanged(SensorEvent sensorEvent) {
    if (sensorEvent.sensor.getType() == Sensor.TYPE_ACCELEROMETER) {
        float yAcceleration = sensorEvent.values[1]; // 获取竖直方向上的加速度

        // 通过加速度变化判断是否迈出了一步
        if (Math.abs(yAcceleration) > STEP_THRESHOLD) {
            stepCount++; // 步数加 1
            updateStepCount(); // 更新步数显示
        }
    }
}
```

但是，经过安装到手机进行检测后，发现该算法的效果非常不理想，阈值太小会导致轻轻一挥手便加了十几步，阈值过大则会使几步困难，需要大力甩动手机。随后经过调查，本人得知计步器的核心算法应该是检测上下峰值，故对计步器算法进行改进，通过 `motionState` 变量来记录上升和下降的过程，并设置一个阈值来检测是否存在峰值，若连续出现上升峰值和下降峰值，则步数加一。

问题二：对于计步器算法的阈值设定存在困难

解决办法：即使在经过改进后的代码中，阈值的设置仍然十分重要，阈值过小则会使计步器太过灵敏，阈值过大则会使其反应迟钝。但是由于虚拟机上没有加速度传感器，本人只能一点一点改变阈值的大小，打包将 `apk` 发送给手机，利用智能手机去进行测试，经过十几个版本的调整，目前计步器的灵敏度已做到该简易算法下的最好。

五、实验结论

通过本次实验，本人学习到了很多，首先了解了 Android 中加速度传感器的基本知识，掌握了 Android 中加速度传感器的使用方法，其次我也熟悉了解了 Android Studio 的基本使用方法，并掌握了一部分的语法及使用技巧。另外，我也锻炼了我的个人排错能力，做到了独立地发现并解决问题，最终完成实验。