感知技术与应用

实验指导

——光线传感器

目录

[实验：光线传感器 3](#_Toc37058095)

[实验目的 3](#_Toc37058096)

[实验要求 3](#_Toc37058097)

[基础知识 3](#_Toc37058098)

[实验内容 7](#_Toc37058099)

[一、获取设备中光线传感器的值 7](#_Toc37058100)

[作业： 12](#_Toc37058101)

[注意事项 13](#_Toc37058102)

# 实验：光线传感器

## 实验目的

本次实验的目的是让大家了解Android中光线传感器的基本知识，掌握Android中光线传感器的使用方法。

## 实验要求

1. 了解Android中光线传感器基本知识
2. 掌握Android中光线传感器使用方法

## 基础知识

**光线传感器基础**

光线传感器的好处是可以根据手机所处环境的光线来调节手机屏幕的亮度和键盘灯。例如在光线充足的地方屏幕会很亮，键盘灯就会关闭。相反如果在暗处，键盘灯就会亮，屏幕较暗（与屏幕亮度的设置也有关系），这样既保护了眼睛又节省了电量。光线传感器在进入睡眠模式时候会发出蓝色周期性闪动的光，非常美观。在本节的内容中，将详细讲解Android系统光线传感器的基本知识。

**光线传感器介绍**

在Android设备中，光线传感器通常位于前摄像头旁边的一个小点，如果在光线充足的情况下（室外或者是灯光充足的室内），大约在2～3秒之后键盘灯会自动熄灭，即使再操作机器键盘灯也不会亮，除非到了光线比较暗的地方才会自动地亮起来。如果在光线充足的情况下用手将光线感应器遮上，在2～3秒后键盘灯会自动亮起来，在此过程中光线感应器起到了一个节电的功能。

要想在Android设备中监听光线传感器，需要掌握如下所示的监听方法。

（1）registerListenr(SensorListenerlistenr,int sensors,int rate)：已过时。

（2）registerListenr(SensorListenerlistenr,int sensors)：已过时。

（3）registerListenr(SensorEventListenerlistenr,Sensor sensors,int rate)。

（4）registerListenr(SensorEventListenerlistenr,Sensor sensors,int rate,Handlerhandler)：因为SensorListener已经过时，所以相应的注册方法也过时了。

在上述方法中，各个参数的具体说明如下所示。

Listener：相应监听器的引用。

Sensor：相应的感应器引用。

Rate：感应器的反应速度，这个必须是系统提供的4个常量之一。

SENSOR\_DELAY\_NORMAL：匹配屏幕方向的变化。

SENSOR\_DELAY\_UI：匹配用户接口。

SENSOR\_DELAY\_GAME：匹配游戏。

SENSOR\_DELAY\_FASTEST.：匹配所能达到的最快。

开发光传感器应用时需要监测SENSOR\_LIGHT，例如下面的代码。

　private SensorListener mySensorListener = new SensorListener(){

　@Override

　 public void onAccuracyChanged(int sensor, int accuracy) {}//重写onAccuracyChanged方法

　 @Override

　 public void onSensorChanged(int sensor, float[] values) { //重写onSensorChanged方法

　　　if(sensor == SensorManager.SENSOR\_LIGHT){　　　　 　　　　//只检查光强度的变化

　　　　 myTextView1.setText("光的强度为："+values[0]);　 　　//将光的强度显示到TextView

　　　　　　　 }

　　　　　 }

　};

　@Override

　protected void onResume() {　　　　　　　　　　　　 　　//重写的onResume方法

　　　　　mySensorManager.registerListener(　　　　　　　//注册监听

　　　　　　　　　mySensorListener,　　　　　　　　　　　 　　//监听器SensorListener对象

　　　　　　　　　SensorManager.SENSOR\_LIGHT,　　　　　　　　//传感器的类型为光的强度

　　　　　　　　　SensorManager.SENSOR\_DELAY\_UI　　　　 　　//频率

　　　　　　　　　);

　　　　　super.onResume();

　}

在上述代码中，通过if语句判断是否为光的强度改变事件。在代码中只对光强度改变事件进行处理，将得到的光强度显示在屏幕中。光传感器只得到一个数据，而并不像其他传感器那样得到的是X 、Y 、\_Z \_3个方向上的分量。

在注册监听时，通过传入“SensorManager.SENSOR\_LIGHT”来通知系统只注册光传感器。

**在Android中使用光线传感器的方法**

在Android设备中，使用光线传感器的基本流程如下所示。

（1）通过一个SensorManager来管理各种感应器，要想获得这个管理器的引用，必须通过如下所示的代码来实现。

(SensorManager)getSystemService(Context.SENSOR\_SERVICE);

（2）在Android系统中，所有的感应器都属于Sensor类的一个实例，并没有继续细分下去，所以Android对于感应器的处理几乎是一模一样的。既然都是Sensor类，那么怎么获得相应的感应器呢？这时就需要通过SensorManager来获得，可以通过如下所示的代码来确定要获得的感应器类型。

sensorManager.getDefaultSensor(Sensor.TYPE\_LIGHT);

通过上述代码获得了光线感应器的引用。

（3）在获得相应的传感器的引用后可以用来感应光线强度的变化，此时需要通过监听传感器的方式来获得变化，监听功能通过前面介绍的监听方法实现。Android提供了两个监听方式，一个是SensorEventListener，另一个是SensorListener，后者已经在Android API上显示过时了。

（4）在Android中注册传感器后，此时就说明启用了传感器。使用感应器是相当耗电的，这也是为什么传感器没有被广泛应用的主要原因，所以必须在不需要它的时候及时关掉。在Android中通过如下所示的注销方法来关闭。

unregisterListener(SensorEventListenerlistener)；

unregisterListener(SensorEventListenerlistener,Sensor sensor)。

（5）使用SensorEventListener来具体实现，在Android设备中有下面两种方式实现这个监听器的方法。

onAccuracyChanged(Sensor sensor, int accuracy)：是反应速度变化的方法，也就是rate变化时的方法；

onSensorChanged(SensorEvent event)：是传感器值变化的相应的方法。

读者需要注意的是，上述两个方法会同时响应。也就是说，当感应器发生变化时，这两个方法会一起被调用。上述方法中的accuracy的值是4个常量，对应的整数如下所示。

SENSOR\_DELAY\_NORMAL：3。

SENSOR\_DELAY\_UI：2。

SENSOR\_DELAY\_GAME：1。

SENSOR\_DELAY\_FASTEST：0。

而类SensorEvent有4个成员变量，具体说明如下所示。

Accuracy：精确值。

Sensor：发生变化的感应器。

Timestamp：发生的时间，单位是纳秒。

Values：发生变化后的值，这个是一个长度为3的数组。

光线传感器只需要values[0]的值，其他两个都为0。而values[0]就是开发光线传感器所需要的，单位是lux照度单位。

## 实验内容

### 一、获取设备中光线传感器的值

#### 1编写布局文件activity\_main.xml，具体实现代码如下所示。

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

　　xmlns:tools="http://schemas.android.com/tools"

　　android:layout\_width="match\_parent"

　　android:layout\_height="match\_parent"

　　android:paddingBottom="@dimen/activity\_vertical\_margin"

　　android:paddingLeft="@dimen/activity\_horizontal\_margin"

　　android:paddingRight="@dimen/activity\_horizontal\_margin"

　　android:paddingTop="@dimen/activity\_vertical\_margin"

　　tools:context=".MainActivity" >

　　<TextView

　　　　android:layout\_width="wrap\_content"

　　　　android:layout\_height="wrap\_content"

　　　　android:text="@string/hello\_world" />

</RelativeLayout>

#### 2编写程序文件MainActivity.java，具体实现代码如下所示。

package com.example.sensor;

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.renderscript.Sampler.Value;

import android.app.Activity;

import android.view.Menu;

import android.widget.TextView;

public class MainActivity extends Activity implements SensorEventListener　{

　　private SensorManager sensor;

　　private TextView text;

　　@Override

　　protected void onCreate(Bundle savedInstanceState) {

　　　　super.onCreate(savedInstanceState);

　　　　setContentView(R.layout.activity\_main);

　　　　sensor = (SensorManager)getSystemService(SENSOR\_SERVICE);

　　　　text = (TextView)findViewById(R.id.textView1);

　　}

　　@Override

　　public boolean onCreateOptionsMenu(Menu menu) {

　　　　// Inflate the menu; this adds items to the action bar if it is present

　　　　getMenuInflater().inflate(R.menu.activity\_main, menu);

　　　　return true;

　　}

　　@Override

　　protected void onPause() {

　　　　// TODO Auto-generated method stub

　　　　sensor.unregisterListener(this);

　　　　super.onPause();

　　}

　　@Override

　　protected void onResume() {

　　　　// TODO Auto-generated method stub

　　　　sensor.registerListener(this,sensor.getDefaultSensor(Sensor.TYPE\_LIGHT),

　　　　SensorManager.SENSOR\_DELAY\_GAME);

　　　　super.onResume();

　　}

　　@Override

　　protected void onStop() {

　　　　// TODO Auto-generated method stub

　　　　sensor.unregisterListener(this);

　　　　super.onStop();

　　}

　　@Override

　　public void onAccuracyChanged(Sensor sensor, int accuracy) {

　　　　// TODO Auto-generated method stub

　　}

　　@Override

　　public void onSensorChanged(SensorEvent event) {

　　　　// TODO Auto-generated method stub

　　　　float[] values = event.values;

　　　　int sensorType = event.sensor.TYPE\_LIGHT;

　　　　if(sensorType==Sensor.TYPE\_LIGHT)

　　　　{

　　　　　　text.setText(String.valueOf(values[0]));

　　　　}

　　}

}

# 作业：

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。

完成实验内容后，设计一个光线传感器的app，至少包含如下功能：

* + 显示当前光线传感器的光照强度；
  + 显示当前光线传感器的名称，耗电量，以及最大测量范围；
  + 能自动调整屏幕的亮度。

获取上述参数方法如下：

myS = mySm.getDefaultSensor(Sensor.TYPE\_LIGHT);

名称： myS.getName() ;

耗电量（mA）：myS.getPower();

最大测量范围：myS.getMaximumRange() ;

# 注意事项

1. 独立自主完成实验任务，**切勿抄袭！**

如实验报告或代码有雷同现象则视为抄袭，**被抄袭者**与**抄袭者**实验**总成绩**均按**0分**处理。

1. 本次实验需提交如下内容：
2. 完整可运行的程序源代码。
3. 实验报告电子版。
4. 提交时间：

在规定之间内提交

1. 评分方法：
   * 达到基本作业要求，85分，根据实验报告质量和源码质量上下浮动5分
   * 作业的每项要求，少完成一项，分数减少3-5分，如控件数量不足、类别不足、没有消息相应处理等，最低分为0分
   * 超过基本作业要求，酌情加分，最高分100
   * 发现抄袭，0分
   * 作业未按时提交，0分