1、Android中支持的Snesor种类

|  |  |
| --- | --- |
| Sensor | 说明 |
| Sensor.TYPE\_ACCELEROMETER | 加速度感应检测 |
| Sensor.TYPE\_MAGNETIC\_FIELD | 磁场感应检测 |
| Sensor.TYPE\_ORIENTATION | 方位感应检测 |
| Sensor.TYPE\_GYROSCOPE | 回转仪感应检测 |
| Sensor.TYPE\_LIGHT | 亮度感应检测 |
| Sensor.TYPE\_PRESSURE | 压力感应检测 |
| Sensor.TYPE\_TEMPERATURE | 温度感应检测 |
| Sensor.TYPE\_PROXIMITY | 接近感应检测 |

2、Android感应检测管理----SensorManager

感应检测功能：

1、取得SensorManager

   使用感应检测Sensor首要先获取感应设备的检测信号，你可以调用Context.getSysteService(SENSER\_SERVICE)方法来取得感应检测的服务

2、实现取得感应检测Sensor状态的监听功能

   实现以下两个SensorEventListener方法来监听，并取得感应检测Sensor状态：

**[java]** [view plaincopyprint?](http://blog.csdn.net/huangbiao86/article/details/6745933)

1. //在感应检测到Sensor的精密度有变化时被调用到。
2. **public** **void** onAccuracyChanged(Senso sensor,**int** accuracy);
3. //在感应检测到Sensor的值有变化时会被调用到。
4. **public** **void** onSensorChanged(SensorEvent event);

3、实现取得感应检测Sensor目标各类的值

   实现下列getSensorList()方法来取得感应检测Sensor的值；

**[java]** [view plaincopyprint?](http://blog.csdn.net/huangbiao86/article/details/6745933)

1. List<Sensor> sensors = sm.getSensorList(Sensor.TYPE\_TEMPERATURE);

4、注册SensorListener

**[java]** [view plaincopyprint?](http://blog.csdn.net/huangbiao86/article/details/6745933)

1. sm.regesterListener(SensorEventListener listener, Sensor sensor, **int** rate);

第一个参数：监听Sensor事件，第二个参数是Sensor目标种类的值，第三个参数是延迟时间的精度密度。延迟时间的精密度参数如下：

|  |  |
| --- | --- |
| 参数 | 延迟时间 |
| SensorManager.SENSOR\_DELAY\_FASTEST | 0ms |
| SensorManager.SENSOR\_DELAY\_GAME | 20ms |
| SensorManager.SENSOR\_DELAY\_UI | 60ms |
| SensorManager.SENSOR\_DELAY\_NORMAL | 200ms |

因为感应检测Sensor的服务是否频繁和快慢都与电池参量的消耗有关，同时也会影响处理的效率，所以兼顾到消耗电池和处理效率的平衡，设置感应检测Sensor的延迟时间是一门重要的学问，需要根据应用系统的需求来做适当的设置。

感应检测Sensor的硬件检测组件收不同的厂商提供。你可以采用Sensor的getVendor(),Sensor()的getName()和Sensor的getVeesrion()方法来取得 厂商的名称、产品和产品版本。

5、取消注册

**[java]** [view plaincopyprint?](http://blog.csdn.net/huangbiao86/article/details/6745933)

1. sm.unregisterListener(SensorEventListener listener)

**加速度感应检测——Accelerometer**

Accelerometer Sensor测量的是所有施加在设备上的力所产生的加速度的负值（包括重力加速度）。加速度所使用的单位是m/sec^2，数值是加速度的负值。

SensorEvent.values[0]：加速度在X轴的负值

SensorEvent.values[1]：加速度在Y轴的负值

SensorEvent.values[2]：加速度在Z轴的负值

例如：

当手机Z轴朝上平放在桌面上，并且从左到右推动手机，此时X轴上的加速度是正数。

当手机Z轴朝上静止放在桌面上，此时Z轴的加速度是+9.81m/sec^2。

当手机从空中自由落体，此时加速度是0

当手机向上以Am/sec^2的加速度向空中抛出，此时加速度是A+9.81m/sec^2

**重力加速度感应检测——Gravity**

重力加速度，其单位是m/sec^2，其坐标系与Accelerometer使用的一致。当手机静止时，gravity的值和Accelerometer的值是一致的。

**线性加速度感应检测——Linear-Acceleration**

Accelerometer、Gravity和Linear-Acceleration三者的关系如下公式：

accelerometer = gravity + linear-acceleration

**地磁场感应检测——Magnetic-field**

地磁场的单位是micro-Tesla(uT)，检测的是X、Y、Z轴上的绝对地磁场。

**陀螺仪感应检测——Gyroscope**

陀螺仪的单位是弧度/秒，测量的是物体分别围绕X，Y，Z轴旋转的角速度。它的坐标系与加速度传感器的坐标系相同。逆时针方向旋转的角度正的。也就是说，如果设备逆时针旋转，观察者向X，Y，Z轴的正方向看去，就报告设备是正转的。请注意，这是标准的正旋转的数学定义。

**光线感应检测——Light**

values[0]:表示环境光照的水平，单位是SI lux。

**位置逼近感应检测——Proximity**

values[0]:逼近的距离，单位是厘米(cm)。有一些传感器只能支持近和远两种状态，这种情况下，传感器必须报告它在远状态下的maximum\_range值和在近状态下的小值。

**旋转矢量感应检测——Rotation Vector**

旋转向量是用来表示设备的方向，它是由角度和轴组成，就是设备围绕x,y,z轴之一旋转θ角度。旋转向量的三个要素是，这样旋转向量的大小等于sin(θ/2)，旋转向量的方向等于旋转轴的方向。

values[0]: x\*sin(θ/2)   
values[1]: y\*sin(θ/2)   
values[2]: z\*sin(θ/2)   
values[3]: cos(θ/2) (optional: only if value.length = 4)

**方向感应检测——Orientation**

其单位是角度

values[0]: Azimuth(方位)，地磁北方向与y轴的角度，围绕z轴旋转(0到359)。0=North, 90=East, 180=South, 270=West   
values[1]: Pitch(俯仰),围绕X轴旋转(-180 to 180), 当Z轴向Y轴运动时是正值  
values[2]: Roll(滚)，围绕Y轴旋转(-90 to 90)，当X轴向Z轴运动时是正值

注意：这里的定义与航空中定义的yaw、pitch和roll不同，航空中的X轴是沿着飞机的最长边(从头到尾)。

注意：这个传感器类型存在遗留问题，请使用与getRotationMatrix()和remapCoordinateSystem()以及getOrientation()配合使用，来计算值代替得到的值。

重要说明：由于历史的原因，以顺时针旋转的滚动角是正的（从数学上讲，它应该是逆时针方向）。