

# 在 Java ME 中如何获取加速度传感器值

本文上次修订时间：2009 年 6 月 3 日 13:03

本文翻译自《How to: Get Accelerator Sensor Values in Java ME》

## 目录

[隐藏]

- 1 概述
- 2 代码范例
- 3 范例应用
- 4 参考资料

## 概述

S60 5th Edition 终端支持 Mobile Sensor API (JSR-256)。诺基亚 5800 XM 和诺基亚 N97 是首批内置该 API 的终端。S60 5th Edition（以及大多数的 S60 3rd Edition FP1 和 S60 3rd Edition FP2）终端都配有加速度计，会给出基于终端位置和运动状态的一些值。比如，当终端屏幕被切换到横向位置时，屏幕内容也被更新为横向显示。

正如《[在 Java ME 中如何获取传感器信息](#)》一文所述，诺基亚 N97 Mobile Sensor API 能识别两个不同的加速度传感器，一个提供整数型值，另一个则提供双精度型值。本文讲述如何在 MIDlet 中获取来自这两个加速度计的值。

加速度计的使用步骤如下：

- § 在终端中查找类型为“acceleration”的传感器
- § 选择提供整数值或双精度值的传感器
- § 获取传感器的 URL
- § 打开连接到传感器的 `SensorConnection`
- § 实现 `dataReceived()` 方法，以便使用传感器数据

## 代码范例

下面列出的代码范例展示了如何搜索“acceleration”类型的传感器，及如何选择传感器的值类型。该方法也能获取传感器 URL，并返回针对该 URL 的正确 `SensorConnection`。

```

/**
 * Searches sensors of "accelerator" quantity and if found returns a
 * SensorConnection opened to it. Based on the boolean value of type_int
 * either TYPE_DOUBLE or TYPE_INT sensor is used.
 * @return SensorConnection, which has been opened to a sensor matching the criteria
 */
private SensorConnection openSensor() {
    infos = SensorManager.findSensors("acceleration", null);
    if (infos.length==0) return null;
    int datatypes[] = new int[infos.length];
    int i = 0;
    String sensor_url = "";
    if (!type_int) {
        System.out.println("Searching TYPE_DOUBLE sensor...");
        while (!sensor_found) {
            datatypes[i] = infos[i].getChannelInfos()[0].getDataType();
            if (datatypes[i] == 1) { //ChannelType.TYPE_DOUBLE = 1
                sensor_url = infos[i].getUrl();
                System.out.println("Sensor: " + sensor_url + ": TYPE_DOUBLE found.");
                sensor_found = true;
            }
            else i++;
        }
    }
    else if (type_int) {
        System.out.println("Searching TYPE_INT sensor...");
        while (!sensor_found) {
            datatypes[i] = infos[i].getChannelInfos()[0].getDataType();
            if (datatypes[i] == 2) { //ChannelType.TYPE_INT = 2
                sensor_url = infos[i].getUrl();
                System.out.println("Sensor: " + sensor_url + ": TYPE_INT found.");
                sensor_found = true;
            }
            else i++;
        }
    }
    System.out.println("Sensor: " + sensor_url);
    try {
        return (SensorConnection)Connector.open(sensor_url);
    }catch (IOException ioe) {
        ioe.printStackTrace();
        return null;
    }
}

```

```
}
```

下面的方法展示了如何设置 `DataListener` 用以侦听传感器值：

```
/**
 * Initializes (opens) the sensor and sets the DataListener
 */
private synchronized void initSensor() {
    sensor = openSensor();
    if (sensor == null) return;
    try {
        sensor.setDataListener(this, BUFFER_SIZE);
        while(!isStopped){
            try{
                wait();
            }catch(InterruptedException ie){}
        }
        sensor.removeDataListener();
    } catch (IllegalMonitorStateException imse) {
        imse.printStackTrace();
    } catch (IllegalArgumentException iae) {
        iae.printStackTrace();
    }
    try {
        sensor.close();
    } catch(IOException ioe){
        ioe.printStackTrace();
    }
    if (isStopped) {
        sensor = null;
    }
}
```

最后实现 `dataReceived()` 方法，以获取传感器值。同时也列出了 `getIntegerDirections()` 方法，该方法计算被缓存的传感器值的平均值。

```
/**
 * Notification of the received sensor data.
 * @param sensor - SensorConnection, the origin of the received data
 * @param data - the received sensor data
```

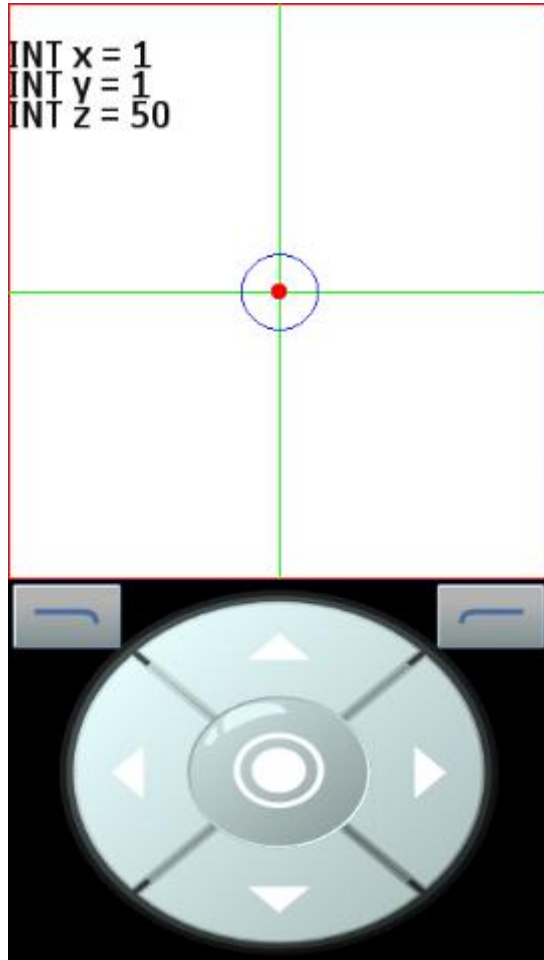
```

    * @param isDataLost - true if some data has been lost
    */
    public void dataReceived(SensorConnection sensor, Data[] data, boolean isDataLost) {
        if (!type_int) {
            double[] directions = getDirections(data);
            x = directions[0];
            y = directions[1];
            z = directions[2];
        }
        else {
            int[] directions = getIntegerDirections(data);
            x_int = directions[0];
            y_int = directions[1];
            z_int = directions[2];
        }
        repaint();
    }

    /**
     * Gets the directions (axis_x, axis_y and axis_z values) from the accelerator
     * sensor data. An average value is calculated from the buffer values.
     * @param data The sensor data
     * @return directions The array containing the axis_x, axis_y and axis_z values
     */
    private static int[] getIntegerDirections(Data[] data) {
        int [][] intValues = new int[3][BUFFER_SIZE];
        int[] directions = new int[3];
        for (int i=0; i<3; i++){
            intValues[i] = data[i].getIntValues();
            int temp = 0;
            for (int j = 0; j<BUFFER_SIZE; j++) {
                temp = temp + intValues[i][j];
            }
            directions[i] = temp/BUFFER_SIZE;
        }
        return directions;
    }
}

```

本文也提供了 `SensorTestMIDlet.jad` 和 `SensorTestMIDlet.jar` 文件。这个 MIDlet 显示了传感器的值，并根据这些值绘出一个示意图，请见下图。



图示为运行于诺基亚 N97 SDK 0.5 上的 SensorTest MIDlet

## 范例应用

§ [SensorTestMIDlet.zip](#) 含有 SensorTestMIDlet.jad、SensorTestMIDlet.jar 和一些源数据

## 参考资料

§ [针对诺基亚 5800 XpressMusic 的 Sensor API \(JSR-256\) beta 插件](#)

§ [诺基亚 N97 SDK 0.5](#)

§ [Mobile Sensor API \(JSR-256\) javadoc 文档及参考实现二进制文件](#)

§ [2009 年摩纳哥诺基亚开发伙伴峰会上展示的以 Mobile Sensor API 控制 RC 小汽车的视频录像](#)

§ [在 Java ME 中如何获取传感器信息](#)