



## Contents

- [1 Introduction](#)
- [2 Getting live accelerometer sensor data](#)
- [3 Using Platform Services to Manipulate Sensor Data](#)
  - ◆ [3.1 How to calculate acceleration](#)
  - ◆ [3.2 How to calculate the angle of acceleration](#)
  - ◆ [3.3 How to calculate the orientation using accelerometer data](#)
  - ◆ [3.4 How to reduce accelerometer sensor noise](#)
- [4 Plotting graphics with Accelerometer Data](#)
- [5 Simulating](#)
- [6 Download](#)
- [7 References](#)
- [8 Authors](#)

## Introduction

In this article we are going to show tips and tricks to use accelerometer sensor data to better create user interaction and improve game and user interfaces. Enclosed in this article you will also find two full implemented swf and the respective source code to download.

## Getting live accelerometer sensor data

While developing a Flash Lite game at [i2tecnologia](#) we found a small issue, to get real-time Accelerometer sensor data you need to set a high frame rate, setting the frame rate to 100 frames per second gets live accelerometer sensor data.

## Using Platform Services to Manipulate Sensor Data

The S60 platform allows Flash Lite applications installed on S60 mobile devices to [access data from the physical sensors of a device](#) using the [Sensor Service API](#).

Flash Lite applications use the S60 Platform Services through Service APIs. The Service APIs are supported through a Nokia-proprietary ActionScript 2.0 library. Before you can create Flash Lite applications that use platform services, you must install the library for your Flash IDE:

- Download the library package (ZIP) to your computer by clicking the following link:

[S60 ActionScript API library 1.0.zip](#)

- Extract the package to the class path folder of your IDE. For example, if you are using Adobe Flash CS3 on Microsoft Windows XP, extract the package to:

```
%USERPROFILE%\Local Settings\Application Data\Adobe\Flash CS3\en\Configuration\Classes\
```

## How\_to\_use\_S60\_Platform\_Services\_to\_create\_accelerometer\_based\_user\_interaction\_with\_Adobe\_Flash\_Lite

Make sure that the extracted ActionScript class files (.as) are located in:

```
<classpath>\com\nokia\lib\
```

or MAC:

```
/Users/%USER%/Library/Application Support/Adobe Flash CS4/en/Configuration/Classes/
```

You can now create Flash Lite applications that use platform services. Programming a Flash Lite application to access the Sensor Service API through its Service API involves three steps:

- Import the Service object from the ActionScript library for S60 Platform Services:

```
import com.nokia.lib.Service;
```

- Create a new Service object for calling the Service API.

```
var sensors = new Service("Service.Sensor", "ISensor");
```

- Use the service object to register the user to receive data from one sensor channel asynchronously

```
sensors.RegisterForNotification(inParams, callBack);
```

The following code shows all steps required to register for listen sensor data asynchronously.

```
fscommand2("DisableKeypadCompatibilityMode");
fscommand2("FullScreen", "true");

Stage.scaleMode = "noScale";

import com.nokia.lib.Service;
import com.nokia.lib.Device;

var deviceObject:Object = new Device();
deviceObject.DisableAutoRotation(true);

// Create a new Service object which has Sensor data
var sensors = new Service("Service.Sensor", "ISensor");

// Define input parameters for picking Accelerometer sensor
var inParam = {SearchCriterion:"AccelerometerAxis"};
// Define result value
var outParams = sensors.FindSensorChannel(inParam);
// Define channel info
var channelInfo = outParams.ReturnValue;

// Define valid values for channel info
var channelId = channelInfo[0].ChannelId;
var contextType = channelInfo[0].ContextType;
var quantity = channelInfo[0].Quantity;
var channelType = channelInfo[0].ChannelType;
var location = channelInfo[0].Location;
var vendorId = channelInfo[0].VendorId;
var dataItemSize = channelInfo[0].DataItemSize;
var channelDataTypeId = channelInfo[0].ChannelDataTypeId;
var channelInfo = {
    ChannelId:channelId, ContextType:contextType, Quantity:quantity,
    ChannelType:channelType, Location:location, VendorId:vendorId,
```

## How\_to\_use\_S60\_Platform\_Services\_to\_create\_accelerometer\_based\_user\_interaction\_with\_Adobe\_Flash\_Lite

```
        DataItemSize:dataItemSize, ChannelDataTypeId:channelDataTypeId
    };

    // Define input parameters for listening rotation channel data
    var inParams = {ListeningType:"ChannelData", ChannelInfoMap:channelInfo};

    // The RegisterForNotification method registers the user to receive data from
    // one sensor channel asynchronously
    sensors.RegisterForNotification(inParams, callBack);
```

### How to calculate acceleration

You measure the distance between two points by the following equation.

```
/**
    getAcceleration

    @arguments (aXAxis:Number, aYAxis:Number)
    @description Get acceleration from two axis
*/
function getAcceleration(aXAxis:Number, aYAxis:Number):Void
{
    return Math.sqrt((aXAxis * aXAxis) + (aYAxis * aYAxis));
}
```

This is an asynchronous method, we define the callback function get all channel data, calculate the axis acceleration, the angle, set rotation and set display orientation.

```
function callBack(transactionID:String, eventID:String, outParam:Object)
{
    // check for errors
    if (outParam.ErrorCode == 0)
    {
        // get all channel data
        var channelData = outParam.ReturnValue;

        xAxis = channelData.XAxisData;
        yAxis = channelData.YAxisData;
        zAxis = channelData.ZAxisData;

        // calculate the axis acceleration
        acc = getAcceleration(xAxis, yAxis);
        acc_txt.text = acc;

        // calculate the max acceleration
        if(maxAcceleration < acc) {
            maxAcc_txt.text = "" + acc;
            maxAcceleration = acc;
        }

        // calculate the angle
        angle = Math.atan2(yAxis, -xAxis);

        // set rotation
        setRotation(-angle);

        // set display orientation
        setDisplayOrientation(-aAngle)
```

## How\_to\_use\_S60\_Platform\_Services\_to\_create\_accelerometer\_based\_user\_interaction\_with\_Adobe\_Flash\_Lite

```
        angle_txt.text = " " + (Math.round(angle * 180/Math.PI) + 90);
        text_txt.text = "xAxis: " + xAxis + "\ryAxis: " + yAxis + "\rzAxis: " + zAxis;
    } else {
        var errorId = outParam.ErrorCode;
        text_txt.text = "Error: " + errorId;
    }
};
```

## How to calculate the angle of acceleration

The follow code make angle corrections and set movieclip rotation.

```
function setRotation(aAngle:Number):Void
{
    mcRotation._rotation = Math.round(aAngle * 180/Math.PI) + 90;
}
```

## How to calculate the orientation using accelerometer data

This part of code is just a proof-of-concept, not a full replacement for Rotation sensor API.

```
/**
    setDisplayOrientation

    @arguments (aAngle:Number)
    @description Set display orientation
*/
function setDisplayOrientation(aAngle):Void
{
    if(aAngle >= -2.25 && aAngle <= -0.25)
    {
        displayOrientation = "Portrait Up";
    }
    else if(aAngle >= -1.75 && aAngle <= 0.75)
    {
        displayOrientation = "Landscape Right";
    }
    else if(aAngle >= 0.75 && aAngle <= 2.25)
    {
        displayOrientation = "Portrait Down";
    }
    else if(aAngle <= -2.25 || aAngle >= 2.25)
    {
        displayOrientation = "Landscape Left";
    }

    orientation_txt.text = displayOrientation;
}
```

## How to reduce accelerometer sensor noise

The follow code is based on the wiki article [Reducing accelerometer sensor noise in Symbian C++](#).

```
/*
```

How to calculate the angle of acceleration

## How\_to\_use\_S60\_Platform\_Services\_to\_create\_accelerometer\_based\_user\_interaction\_with\_Adobe\_Flash\_Lite

```
    calcAverageValue

    @argument aAcc:Number
    @description calculate the average value
*/
private function calcAverageValue(aAcc:Number):Number
{
    var len:Number = averageList.length;
    var errors:Number = 0;
    var sum:Number = 0;

    if(len == 16) {
        averageList;
    }

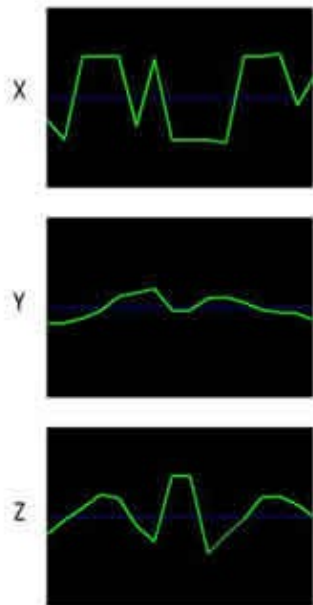
    averageList.push(aAcc);

    // sum the value
    for(var k:Number = 0; k < averageList.length; k++) {
    if(averageList[k] != undefined &&
        averageList[k] != NaN &&
        averageList[k] != null) {

        sum = sum + averageList[k];
    } else {
        errors++;
    }
    }

    return (sum/(averageList.length - errors));
}
```

## Plotting graphics with Accelerometer Data



flash lite user interface

How to reduce accelerometer sensor noise

## How\_to\_use\_S60\_Platform\_Services\_to\_create\_accelerometer\_based\_user\_interaction\_with\_Adobe\_Flash\_Lite

```
function drawChart(chartMc:MovieClip, dataArray:Array):Void{
    chartMc.clear();
    chartMc.lineStyle(2, 0x00ff00);

    for(var i:Number=0; i<dataArray.length; i++){
        var x:Number = (i/(dataArray.length-1)) * chartMc._width;
        var y:Number = (dataArray[i] / maximum) * (chartMc._height/2)

        if(i==0)
            lineMc.moveTo(x, chartMc._height/2);
        else
            lineMc.lineTo(x, chartMc._height/2);
    }
}
```

## Simulating

Unfortunately the accelerometer can't be simulated in the Adobe Device Central. To get it running you will have to compile and run it on your S60 5th Edition device. We made tests on the Nokia 5800 XpressMusic device.

## Download

[Media:Charts acc.zip](#)

## References

[Forum Nokia Library](#)

## Authors

[Felipe Andrade](#)

[Mario Barbosa](#)