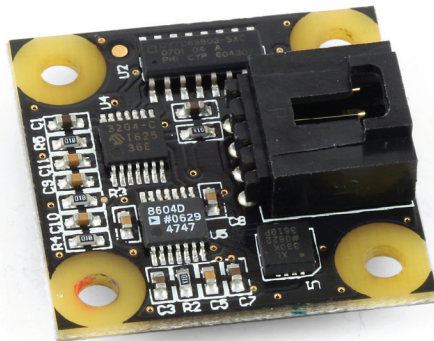


Product Manual

1059 - PhidgetAccelerometer 3-Axis



Phidgets 9999 - Product Manual
For Board Revision 0
© Phidgets Inc. 2009

Contents

4 Product Features

- 4 Programming Environment
- 4 Connection

5 Getting Started

- 5 Checking the Contents
- 5 Connecting all the pieces
- 5 Testing Using Windows 2000/XP/Vista
 - 5 Downloading the Phidgets drivers
 - 5 Running Phidgets Sample Program
- 6 Testing Using Mac OS X
- 7 If you are using Linux
- 7 If you are using Windows Mobile/CE 5.0 or 6.0

8 Programming a Phidget

- 8 Architecture
- 8 Libraries
- 8 Programming Hints
- 8 Networking Phidgets
- 9 Documentation
 - 9 Programming Manual
 - 9 Getting Started Guides
 - 9 API Guides
- 9 Code Samples
- 9 API for the PhidgetAccelerometer 3-Axis
 - 9 Functions
 - 9 Events

10 Technical Section

- 11 Calculating Tilt Angle from Acceleration
- 11 Device Specifications

12 Product History

12 Support

Product Features

- Three axis accelerometer
- Measures ± 3 gravities ($\pm 29.4\text{m/s}^2$) change per axis
- Measure both dynamic acceleration (vibration) and static acceleration (gravity or tilt)
- Internally calibrated at the factory
- Can be used for:
 - Sensing tilt in remotely operated vehicles.
 - Capturing motion input for gaming or for Human Interface Research.
 - Capturing long term vibration data to help with the analysis of machinery performance.
 - Detecting movement or presence by measuring structural vibration.

Programming Environment

Operating Systems: Windows 2000/XP/Vista/7, Windows CE, Linux, and Mac OS X

Programming Languages (APIs): VB6, VB.NET, C#.NET, C++, Flash 9, Flex, Java, LabVIEW, Python, Max/MSP, and Cocoa.

Examples: Many example applications for all the operating systems and development environments above are available for download at www.phidgets.com >> Programming.

Connection

The board connects directly to a computer's USB port.

Getting Started

Checking the Contents

You should have received:

- A PhidgetAccelerometer 3-Axis
- A custom USB cable

Connecting all the pieces

Connect the PhidgetAccelerometer to your PC using the custom USB cable.




Testing Using Windows 2000/XP/Vista/7

Downloading the Phidgets drivers

Make sure that you have the current version of the Phidget library installed on your PC. If you don't, do the following:

Go to www.phidgets.com >> Drivers


Download and run Phidget21 Installer (32-bit, or 64-bit, depending on your PC)

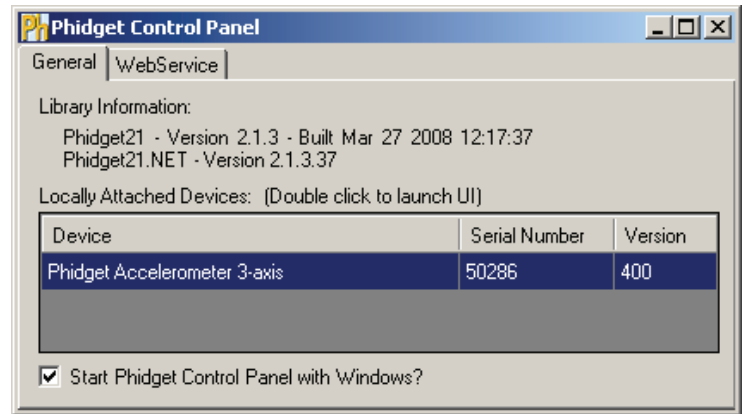
You should see the  icon on the right hand corner of the Task Bar.

Running Phidgets Sample Program

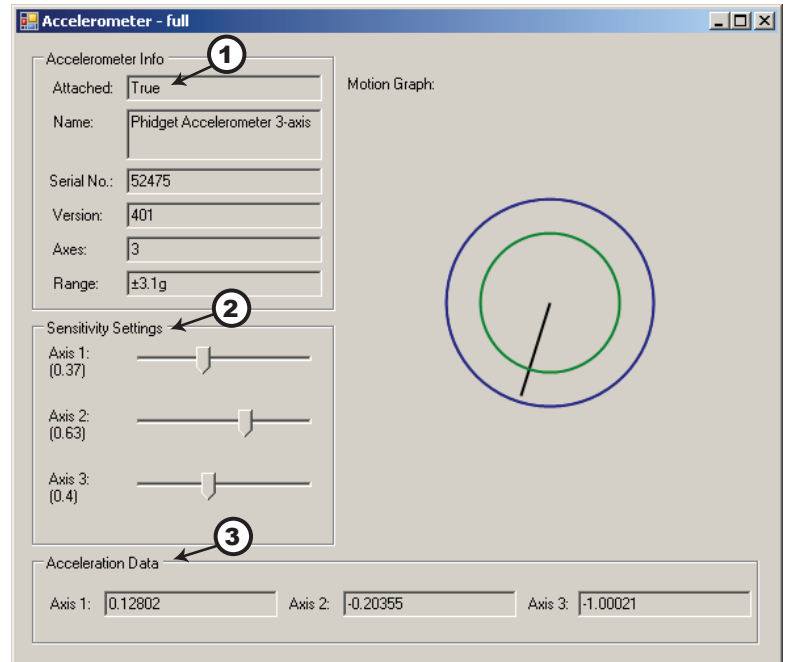
Double clicking on the  icon loads the Phidget Control Panel; we will use this program to make sure that your new Phidget works properly.

The source code for the Accelerometer-Full sample program can be found under C# by clicking on Phidget.com > Programming.

Double Click on the  icon to activate the Phidget Control Panel and make sure that the **Phidget Accelerometer 3-axis** is properly attached to your PC.



1. Double Click on **Phidget Accelerometer 3-Axis** in the Phidget Control Panel to bring up Accelerometer-full and check that the box labelled Attached contains the word True.
2. You can adjust the sensitivity for each axis by moving the appropriate slider.
3. Move the Accelerometer board and you should see the data change to reflect the change of position along the 3 axes.



Testing Using Mac OS X

- Click on System Preferences >> Phidgets (under Other) to activate the Preference Pane
- Make sure that the **Phidget Accelerometer 3-axis** is properly attached.
- Double Click on **Phidget Accelerometer 3-axis** in the Phidget Preference Pane to bring up the Accelerometer-full example. This example will function in a similar way as the Windows version.

If you are using Linux

There are no sample programs written for Linux.

Go to www.phidgets.com >> Drivers

Download Linux Source

- Have a look at the readme file
- Build Phidget21

The most popular programming languages in Linux are C/C++ and Java.

Notes:

Many Linux systems are now built with unsupported third party drivers. It may be necessary to uninstall these drivers for our libraries to work properly.

Phidget21 for Linux is a user-space library. Applications typically have to be run as root, or udev/hotplug must be configured to give permissions when the Phidget is plugged in.

If you are using Windows Mobile/CE 5.0 or 6.0

Go to www.phidgets.com >> Drivers

Download x86, ARMV4I or MIPSII, depending on the platform you are using. Mini-itx and ICOP systems will be x86, and most mobile devices, including XScale based systems will run the ARMV4I.

The CE libraries are distributed in .CAB format. Windows Mobile/CE is able to directly install .CAB files.

The most popular languages are C/C++, .NET Compact Framework (VB.NET and C#). A desktop version of Visual Studio can usually be configured to target your Windows Mobile Platform, whether you are compiling to machine code or the .NET Compact Framework.

Programming a Phidget

Phidgets' philosophy is that you do not have to be an electrical engineer in order to do projects that use devices like sensors, motors, motor controllers, and interface boards. All you need to know is how to program. We have developed a complete set of Application Programming Interfaces (API) that are supported for Windows, Mac OS X, and Linux. When it comes to languages, we support VB6, VB.NET, C#.NET, C, C++, Flash 9, Flex, Java, LabVIEW, Python, Max/MSP, and Cocoa.

Architecture

We have designed our libraries to give you the maximum amount of freedom. We do not impose our own programming model on you.

To achieve this goal we have implemented the libraries as a series of layers with the C API at the core surrounded by other language wrappers.

Libraries

The lowest level library is the C API. The C API can be programmed against on Windows, CE, OS X and Linux. With the C API, C/C++, you can write cross-platform code. For systems with minimal resources (small computers), the C API may be the only choice.

The Java API is built into the C API Library. Java, by default is cross-platform - but your particular platform may not support it (CE).

The .NET API also relies on the C API. Our default .NET API is for .NET 2.0 Framework, but we also have .NET libraries for .NET 1.1 and .NET Compact Framework (CE).

The COM API relies on the C API. The COM API is programmed against when coding in VB6, VBScript, Excel (VBA), Delphi and Labview.

The ActionScript 3.0 Library relies on a communication link with a PhidgetWebService (see below). ActionScript 3.0 is used in Flex and Flash 9.

Programming Hints

- Every Phidget has a unique serial number - this allows you to sort out which device is which at runtime. Unlike USB devices which model themselves as a COM port, you don't have to worry about where in the USB bus you plug your Phidget in. If you have more than one Phidget, even of the same type, their serial numbers enable you to sort them out at runtime.
- Each Phidget you have plugged in is controlled from your application using an object/handle specific to that phidget. This link between the Phidget and the software object is created when you call the .OPEN group of commands. This association will stay, even if the Phidget is disconnected/reattached, until .CLOSE is called.
- For full performance, the Phidget APIs are designed to be used in an event driven architecture. Applications that require receiving all the data streaming from the device will have to use event handlers, instead of polling.

Networking Phidgets

The PhidgetWebService is an application written by Phidgets Inc. which acts as a network proxy on a computer. The PhidgetWebService will allow other computers on the network to communicate with the Phidgets connected to that computer. ALL of our APIs have the capability to communicate with Phidgets on another computer that has the PhidgetWebService running.

The PhidgetWebService also makes it possible to communicate with other applications that you wrote and that are connected to the PhidgetWebService, through the PhidgetDictionary object.

Documentation

Programming Manual

The Phidget Programming Manual documents the Phidgets software programming model in a language and device unspecific way, providing a general overview of the Phidgets API as a whole. You can find the manual at www.phidgets.com >> Programming.

Getting Started Guides

We have written Getting Started Guides for most of the languages that we support. If the manual exists for the language you want to use, this is the first manual you want to read. The Guides can be found at www.phidgets.com >> Programming, and are listed under the appropriate language.

API Guides

We maintain API references for COM (Windows), C (Windows/Mac OSX/Linux), Action Script, .Net and Java. These references document the API calls that are common to all Phidgets. These API References can be found under www.phidgets.com >> Programming and are listed under the appropriate language. To look at the API calls for a specific Phidget, check its Product Manual.

Code Samples

We have written sample programs to illustrate how the APIs are used.

Due to the large number of languages and devices we support, we cannot provide examples in every language for every Phidget. Some of the examples are very minimal, and other examples will have a full-featured GUI allowing all the functionality of the device to be explored. Most developers start by modifying existing examples until they have an understanding of the architecture.

Go to www.phidgets.com >> Programming to see if there are code samples written for your device. Find the language you want to use and click on the magnifying glass besides "Code Sample". You will get a list of all the devices for which we wrote code samples in that language.

API for the PhidgetAccelerometer 3-Axis

We document API Calls specific to this product in this section. Functions common to all Phidgets and functions not applicable to this device are not covered here. This section is deliberately generic. For calling conventions under a specific language, refer to the associated API manual. For exact values, refer to the device specifications.

Functions

int AxisCount() [get] : Constant

Returns the number of axis this PhidgetAccelerometer can measure acceleration on.

double Acceleration (int AxisIndex) [get] : Units = G (9.8m/s^2)

Returns the acceleration of a particular axis. At a standstill each axis will measure between -1.0 and 1.0 g's depending on orientation.

This value will always be between getAccelerationMin and getAccelerationMax.

Index 0 is the x-axis, 1 is the y-axis, and 2 is the z-axis.

double AccelerationMax (int AxisIndex) [get] : Constant

Returns the maximum acceleration value that this axis will report.

double AccelerationMin (int AxisIndex) [get] : Constant

Returns the minimum acceleration value that this axis will report.

double AccelerationChangeTrigger (int AxisIndex) [get,set]

An event that is issued when the Acceleration varies by more than the AccelerationChangeTrigger property.

Events

OnAccelerationChange (int AxisIndex, double Acceleration) [event]

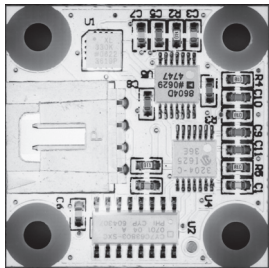
An event issued when a change in acceleration occurs that is greater than the AccelerationChangeTrigger.

Technical Section

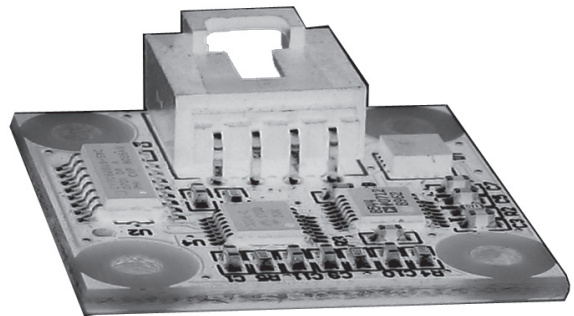
The PhidgetAccelerometer is a 3-Axis accelerometer that can measure ± 3 gravities ($\pm 29.4\text{m/s}^2$) change per axis. It will measure both dynamic acceleration (change in velocity) and static acceleration (gravity or tilt). The PhidgetAccelerometer is internally calibrated. The internal sensor used is an Analog Devices ADXL330.

The theoretical static response of the PhidgetAccelerometer to gravitational acceleration (1g) in space is diagrammed below. This diagram is intended for use in sensor orientation.

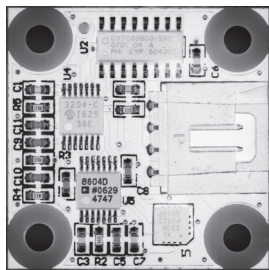
GRAVITY



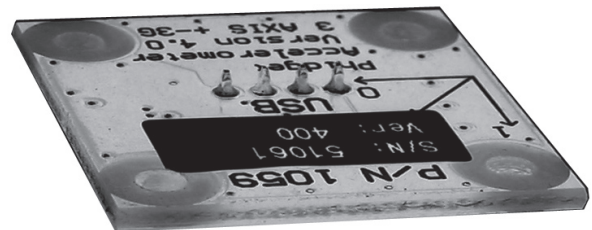
Axis 0 $\approx -1\text{G}$
 Axis 1 $\approx 0\text{G}$
 Axis 2 $\approx 0\text{G}$



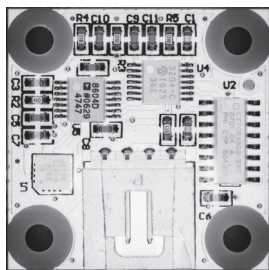
Axis 0 $\approx 0\text{G}$
 Axis 1 $\approx 0\text{G}$
 Axis 2 $\approx +1\text{G}$



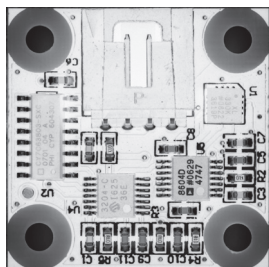
Axis 0 $\approx +1\text{G}$
 Axis 1 $\approx 0\text{G}$
 Axis 2 $\approx 0\text{G}$



Axis 0 $\approx 0\text{G}$
 Axis 1 $\approx 0\text{G}$
 Axis 2 $\approx -1\text{G}$



Axis 0 $\approx 0\text{G}$
 Axis 1 $\approx -1\text{G}$
 Axis 2 $\approx 0\text{G}$



Axis 0 $\approx 0\text{G}$
 Axis 1 $\approx +1\text{G}$
 Axis 2 $\approx 0\text{G}$

Calculating Tilt Angle from Acceleration

Tilt angle can only be measured when the 1056 is stationary. When in motion, the acceleration experienced by the accelerometer is a combination of the device's physical orientation, and the device's movement, and the tilt can therefore not be determined.

The tilt is calculated as follows:

$$\text{Tilt} = (\arcsin(g_x))^{\circ} \mathbf{a}_x + (\arcsin(g_y))^{\circ} \mathbf{a}_y$$

Where g_x = acceleration in the x-axis

g_y = acceleration in the y-axis

\mathbf{a} = indicates which axis the tilt is in

Example of tilt angle calculation

Axis	Measure Acceleration (g)
x	0.25
y	-0.73

$$\text{Tilt} = (\arcsin(0.25))^{\circ} \mathbf{a}_x + (\arcsin(0.73))^{\circ} \mathbf{a}_y$$

$$\text{Tilt} = 14.48^{\circ} \mathbf{a}_x - 46.89^{\circ} \mathbf{a}_y$$

Device Specifications

Characteristic	Value
Update Rate	60 samples/second
Measurement Range (XYZ Axis)	±3G (29.4 m/s ²)
Bandwidth (XYZ Axis)	30Hz
Axis 0 Noise Level (X Axis)	1.9mG standard deviation (σ)
Axis 1 Noise Level (Y Axis)	1.9mG standard deviation (σ)
Axis 2 Noise Level (Z Axis)	2.9mG standard deviation (σ)
Device Quiescent Current Consumption	20mA
Device Active Current Consumption	20mA max
Operating Temperature	0 - 70°C

Product History

Date	Board Revision	Device Version	Comment
January 2007	0	120	Product Release

Support

Call the support desk at 1.403.282.7335 9:00 AM to 5:00 PM Mountain Time (US & Canada) - GMT-07:00

or

E-mail us at: support@phidgets.com