

Product Manual

1052 - PhidgetEncoder



Phidgets 1052 - Product Manual
For Board Revision 0
© Phidgets Inc. 2009

Contents

5 Product Features

- 5 Programming Environment
- 5 Connection

6 Getting Started

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

9 Programming a Phidget

- 9 Architecture
- 9 Libraries
- 9 Programming Hints
- 9 Networking Phidgets
- 10 Documentation
 - 10 Programming Manual
 - 10 Getting Started Guides
 - 10 API Guides
- 10 Code Samples
- 10 API for the PRODUCTNAME
 - 10 Functions
 - 11 Events

11 Technical Section

- 11 Quadrature Encoder Fundamentals
- 11 Using Other Encoders
- 12 Device Specifications

12 Product History

12 Support

Product Features

- Uses a two-bit mechanical encoder with a built-in momentary-action push-button switch.
- Returns 80 counts for 360 degrees of rotation.
- Detects changes in incremental and absolute position.
- Easily track changes with respect to time.
- Intended to be used as a human interface, not as a device to measure shaft speed.
- Connects directly to a computer's USB port

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

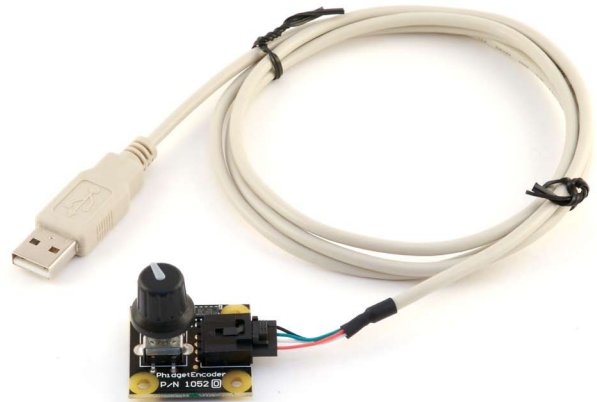
Checking the Contents

You should have received:

- A PhidgeEncoder board
- A custom USB cable

Connecting all the pieces

Connect the PhidgetEncoder board to the computer using the 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)

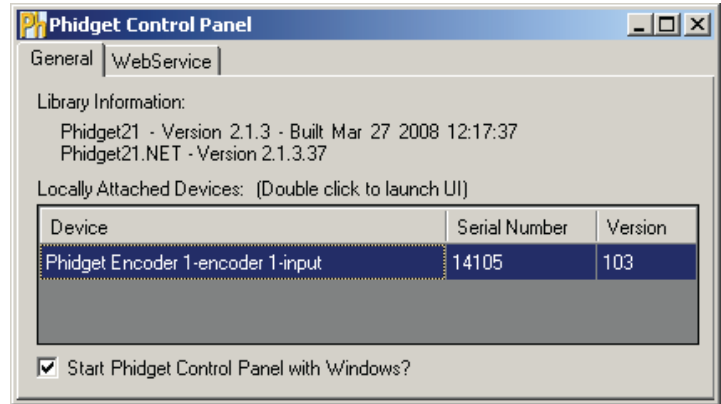
You should see the  icon on the right hand corner of the Task Bar.

Running Phidgets Sample Program

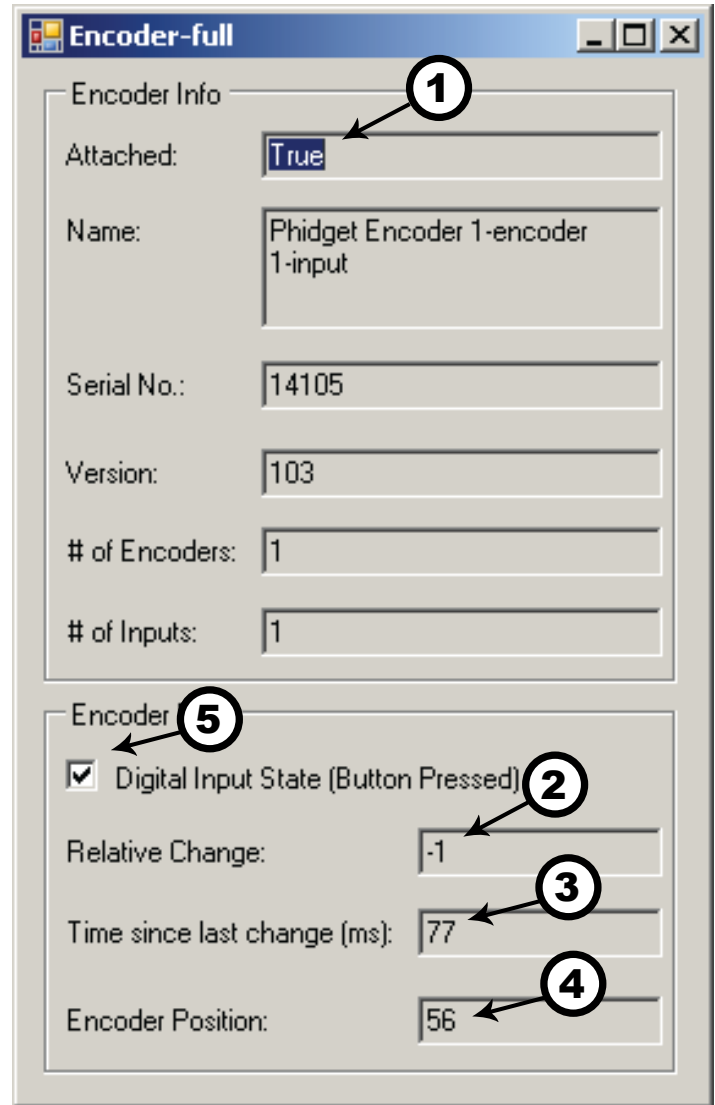
Double clicking on the  icon loads the Phidget Control Panel; we will use this program to make sure that your new Phidget works properly.

The source code for the Encoder-full sample program can be found under C# by clicking on www.phidgets.com >> Programming.

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



1. Double Click on **Phidget Encoder** in the Phidget Control Panel to bring up Encoder-full and check that the box labelled Attached contains the word True.
2. As you turn the knob the relative change will be positive when turning counterclockwise and negative when turning clockwise.
3. Displays the time in milliseconds since the last time you turned the knob.
4. Returns 80 counts for each knob revolution. The number will decrease when turning the knob clockwise and increase when turning counterclockwise.
5. Tick mark appears when the knob is depressed.



Testing Using Mac OS X

- Click on System Preferences >> Phidgets (under Other) to activate the Preference Pane
- Make sure that the PRODUCTNAME is properly attached.
- Double Click on Phidget PRODUCTNAME in the Phidget Preference Pane to bring up the PRODUCTNAME-full Sample program. This program 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

There is no Control Panel written for Linux, but there are C/C++ and Java code samples available for all Phidgets which will compile and run on Linux without modification.

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 PRODUCTNAME

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 InputCount() [get] : Constant

Returns the number of digital inputs supported by this PhidgetEncoder. On the 1052, there is one digital input - the push button on the shaft of the encoder.

bool InputState (int EncoderIndex) [get]

Returns the state of a particular digital input.

int EncoderCount() [get] : Constant

Returns the number of encoders supported by this PhidgetEncoder. On the 1052, there is one encoder - the unit mounted on the board.

int Position(int EncoderIndex) [get,set]

Returns/sets the position of an encoder. This is an absolute position as calculated since the encoder was plugged in. Dividing position by the number of increments per revolution (80) will give the number of rotations the encoder has travelled.

Position can be set, typically used when an encoder has reached an identifiable (through external means, such as a limit switch) home position. This call does not send information to the device, as an absolute position is maintained only in the library. After this call, position changes from the encoder will use the new value to calculate absolute position

Events

OnDigitalInputChange(int InputIndex, bool State) [event]

An event that is issued whenever the state of a digital input changes.

OnPositionChange(int EncoderIndex, int Time, int PositionChange) [event]

An event that is issued whenever a change in encoder position occurs. This event returns the length of time that the change took (in milliseconds), and the amount of change (positive/negative encoder increments)

Technical Section

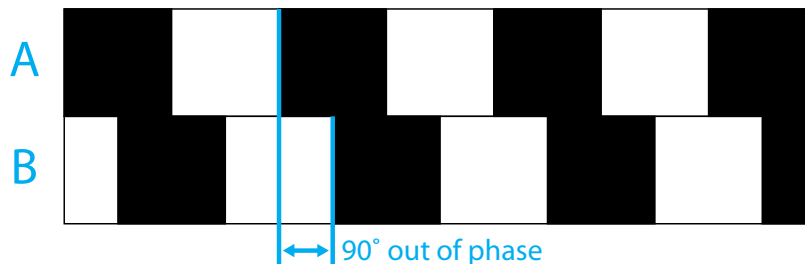
The PhidgetEncoder uses a two-bit quadrature mechanical encoder with a built-in momentary pushbutton SPST switch. It returns 80 counts for 360 degrees of rotation. With It you can:

- Detect changes in incremental and absolute position
- Easily track these changes with respect to time

The encoder that comes with the PhidgetEncoder is manufactured by CTS, part number 290UAA5F201B1.

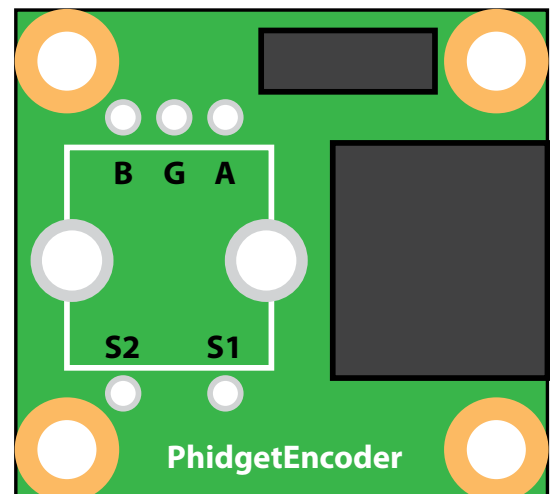
Quadrature Encoder Fundamentals

Quadrature encoders are the most popular type of encoder, using two output channels to dictate both angular displacement and direction of rotation. The term quadrature relates to a system whose components exist at an angle of 90° to each other. In an encoder system, two parallel mechanical disks or optical gratings are set 90° out of phase. In this way, as the two disks spin in unison, the output can signify both the number of pulses that have occurred (the angular displacement) as well as which output channel is leading the other (direction of rotation). In the diagram below, the black segments represent a closed switch and the white segments represent an open switch.



Using Other Encoders

If you decide that the mounted encoder does not meet your requirements it is possible to use other two-bit mechanical encoders. Optical encoders with large counts per revolution should not be expected to work with this device. To change the encoder the user will have to carefully desolder and remove the existing encoder. However, if you attempt to remove the encoder the warranty will be void and we can not offer technical support. For assistance on desoldering techniques we recommend using any search engine on the Internet. Since your new encoder may not have the exact same pin configuration, you should use wire to connect your new encoder to the appropriate pins on the PhidgetEncoder board. We have tested the following encoders, and found that they can be used as replacements:



Manuf.	Part Number	Description
CTS	290UAA5F201B2	2-Bit 9mm 20CPR Encoder Detent w/MO SPST
CTS	290UAA5F201B1	2-Bit 9mm 20CPR Encoder Non-Detent w/MO SPST
CTS	288V232R161B2	2-Bit 16mm 16CPR Encoder Detent w/MO SPST

Note: Most of the above components can be bought at www.digikey.com

Device Specifications

Characteristic	Value
Maximum Velocity	80 counts per second
Device Current Consumption	20mA
Operating Temperature	0 - 70°C

Product History

Date	Board Revision	Device Version	Comment
January 2003	0	100	Product Release
July 2004	0	110	Protocol modified to allow multiple encoder input expansion

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