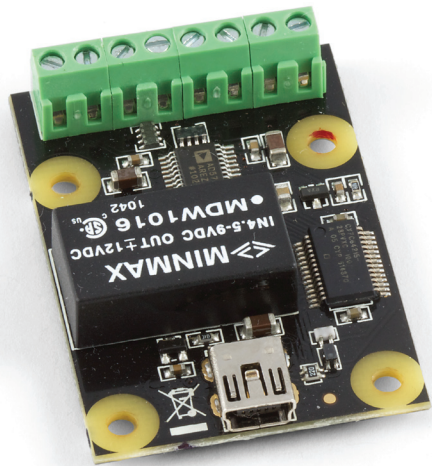


Product Manual

1002 - PhidgetAnalog 4-Output



Phidgets 1002 - Product Manual
For Board Revision 0
© Phidgets Inc. 2011

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 PhidgetAnalog 4-Output
 - 10 Functions
 - 10 Properties
 - 11 Events
- 12 Technical Section
 - 12 General
 - 12 Isolation
 - 12 Current
- 13 Device Specifications

13 Product History

13 Support

Product Features

- Produces a set voltage from -10V to +10V
- You can connect up to 4 devices
- Current limited to $\pm 20\text{mA}$ per channel
- Can be used to control devices such as Incandescent and LED dimmers, valve actuators, motor controls, etc.

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:

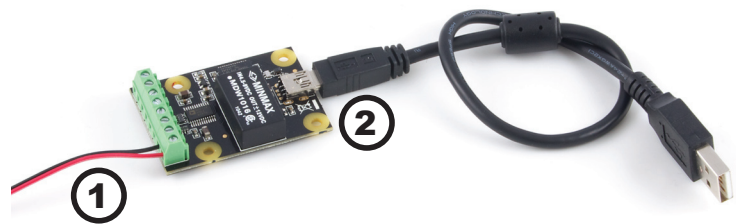
- A PhidgetAnalog 4-Output
- A Mini-USB cable
- A Hardware Mounting Kit

In order to test your new Phidget you will also need:

- A device requiring a 0 to 10V signal
- A multimeter could also be used to monitor the output voltage

Connecting all the pieces

1. Connect your device to the PhidgetAnalog board using one of the terminal blocks.
2. Connect the PhidgetAnalog 4-Output to your computer using the Mini-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)

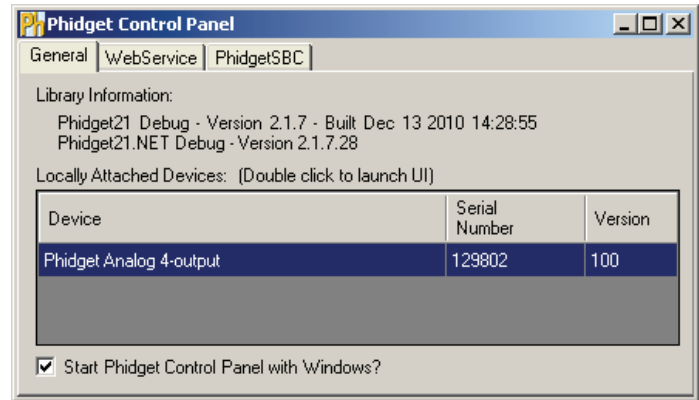
You should see the  icon on the right hand corner of the Task Bar.

Running Phidgets Sample Program

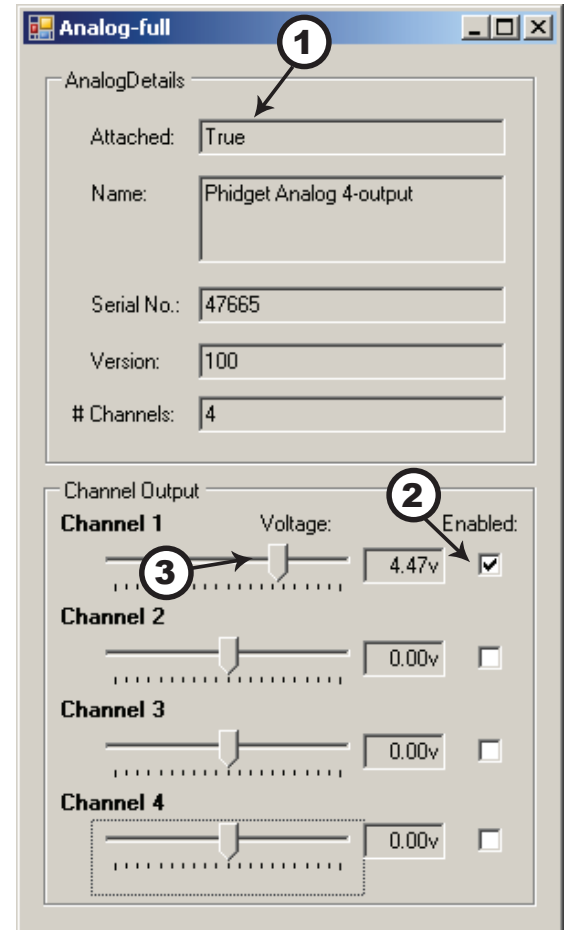
Double clicking on the  icon loads the Phidget Control Panel; we will use this program to make sure that your new Phidget works properly.

The source code for the Analog-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 Analog 4-output** is properly attached to your PC.



1. Double Click on **Phidget Analog 4-output** in the Phidget Control Panel to bring up Analog-full and check that the box labelled Attached contains the word True.
2. Enable the Channel to which your device is connected.
3. Use the slider to set the voltage. Your device should react according to the set voltage. If you are using a multimeter, it should display the voltage as set by the slider.



Testing Using Mac OS X

- Click on System Preferences >> Phidgets (under Other) to activate the Preference Pane
- Make sure that the **PhidgetAnalog 4-Output** is properly attached.
- Double Click on Phidget PhidgetAnalog 4-Output in the Phidget Preference Pane to bring up the Analog-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

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 PhidgetAnalog 4-Output

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

Properties

int OutputCount() [get]

Gets the number of analog outputs supported by the PhidgetAnalog.

double Voltage(int OutputIndex) [get, set]

Gets or sets the current voltage output setting for an analog output, in Volts. The range is `VoltageMin-VoltageMax`. Voltage is not actually applied until `Enabled` is set to true.

double VoltageMin(int OutputIndex) [get]

Gets the minimum supported output voltage, for an output, in Volts.

double VoltageMax(int OutputIndex) [get]

Gets the maximum supported output voltage, for an output, in Volts.

bool Enabled(int OutputIndex) [get, set]

Gets the enabled state for an output. When enabled, the output drives at the set Voltage, up to 20mA. When disabled, the output is tied to ground via a 4K pull-down.

Events

OnError(int ErrorCode, String ErrorDescription)

The PhidgetAnalog will throw error events under certain circumstances:

ErrorCode = EEPHIDGET_OVERCURRENT - An overcurrent condition has occurred on an output. Under this condition, the output is clamped to 20mA.

ErrorCode = EEPHIDGET_OVERTEMP - A Thermal Shutdown state has occurred. The outputs will be shut down under this condition.

When overcurrent or overtemperature state have ended, there will be an error event with the EEPHIDGET_OK code.

See the ErrorDescription string for specific error details.

Technical Section

General

Produces a voltage over -10V to +10V. This voltage will be asserted up to +-20mA (short-circuit). An error will be reported when this current is exceeded, and the voltage the customer is getting is no longer accurate. The voltage is produced with 12 bit resolution (4.8mV).

Isolation

The 1002 is not isolated. All 4 channels use the same ground.

Current

The PhidgetAnalog board will limit the available current per channel at approximately 20mA. For reliable results, don't approach this limit, as it will vary from channel to channel. For maximum accuracy, limit the current to 5mA. If more than +- 20mA of current is drawn, the device goes into a constant current supply mode. When this happens, the current will be held constant at it's maximum value, and the output voltage will depend on the value of the load that the device is trying to drive.

If your application requires a larger current source / sink, the analog output can be used as an input to a larger external amplifier circuit.

All the power is supplied by the USB bus.

Device Specifications

Characteristic	Value
Output Current Limit per channel	20 mA
Output Voltage Resolution	4.8 mV (12bit)
Output Impedance (channel enabled)	0.5 Ω
Output Impedance (channel not enabled)	4 k Ω
Expected Load	2 k Ω
USB Voltage Range	4.5 - 5.25 VDC
USB Max Current	300 mA
USB Quiescent Current	55 mA
Capacitive Load Stability	4000 pF
Output Voltage Range	± 10 V
Terminal Block min/max wire gauge	16-26 AWG
Isolation	Not Isolated: all channels share common ground
Operating Temperature	0 - 70°C

Product History

Date	Board Revision	Device Version	Comment
March 2011	0	100	Product Release, requires phidget21 2.1.8+

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