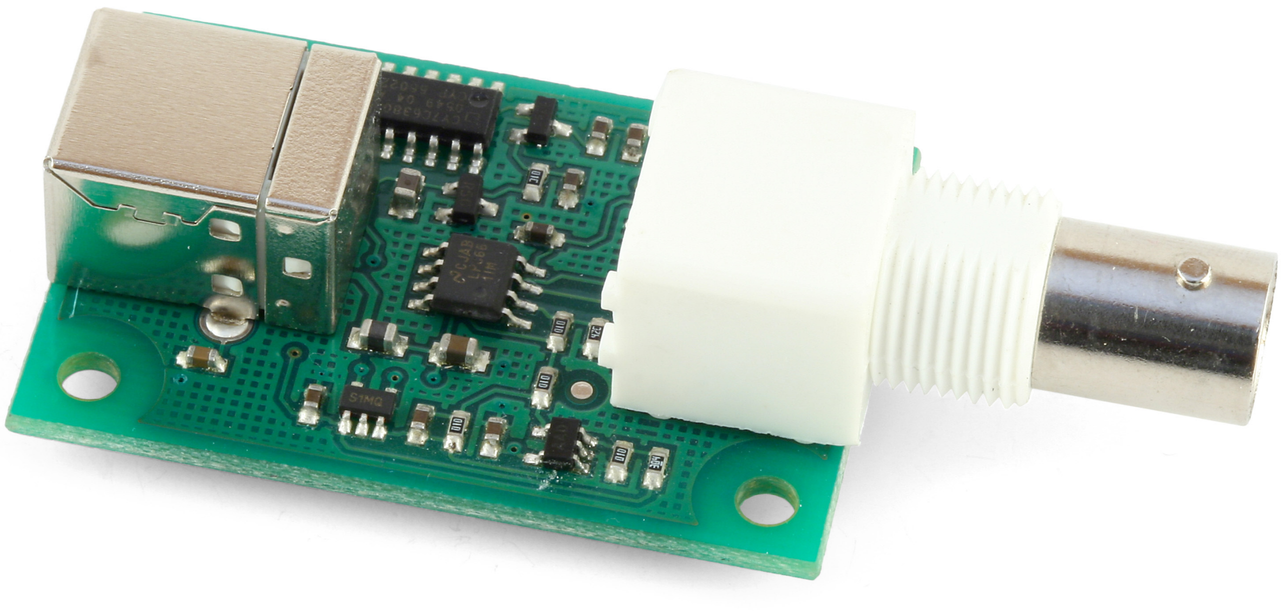
1058 - PhidgetPHSensor

For Board Revision 0



Product Features

- Connects to any typical pH or ORP Electrode through a common BNC connection
- Measures the pH level of a sample.
- Measures the Oxygen Reduction Potential of a sample.
- Connects directly to a computer's USB port

Programming Environment

Operating Systems: Windows 2000/XP/Vista, 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.

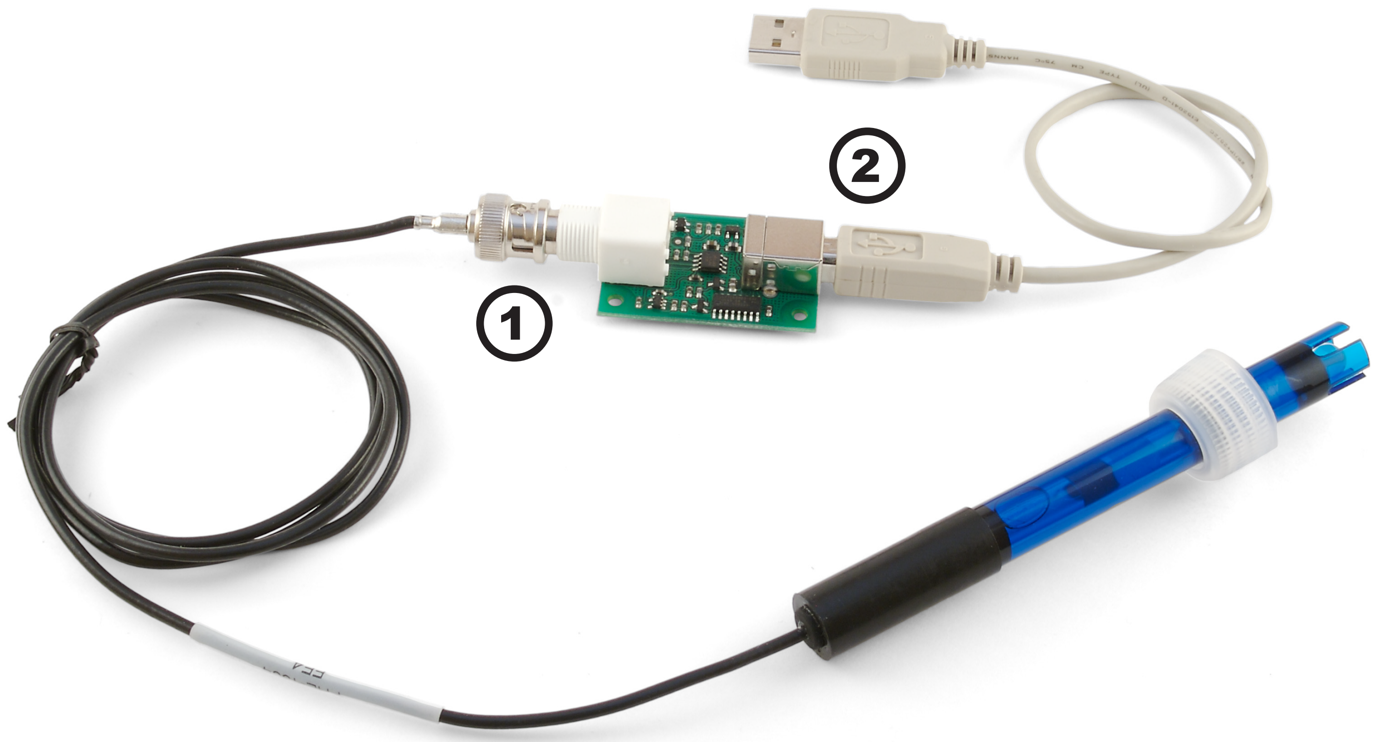
Installing the hardware

The kit contains:

- A PHidgetPHSensor
- A USB cable

You will also need:

- A general purpose pH Electrode or ORP Electrode



1. Connect your electrode to the BNC connector on the PHSensor board.
2. Connect the PhidgetPHSensor to your computer using the USB cable.

Downloading and Installing the software

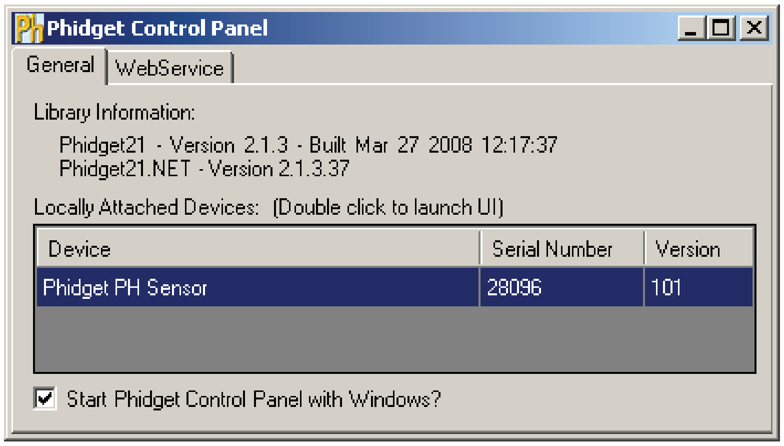
If you are using Windows 2000/XP/Vista

Go to www.phidgets.com >> Drivers

Download and run Phidget21.MSI

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

Testing the PhidgetPH Sensor Functionality



Phidget Control Panel

Library Information:

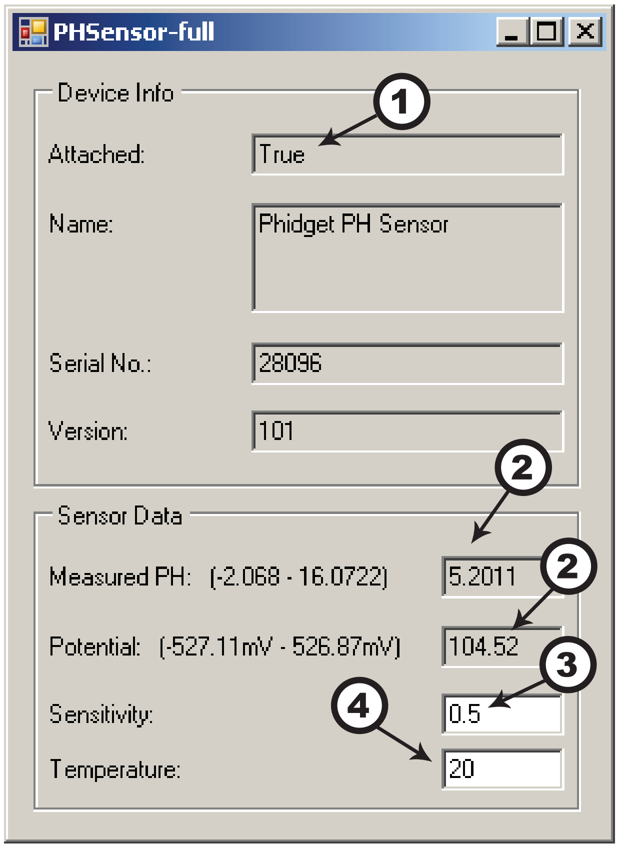
Phidget21 - Version 2.1.3 - Built Mar 27 2008 12:17:37
Phidget21.NET - Version 2.1.3.37

Locally Attached Devices: (Double click to launch UI)

Device	Serial Number	Version
Phidget PH Sensor	28096	101

☑ Start Phidget Control Panel with Windows?

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



PHSensor-full

Device Info

Attached:	True
Name:	Phidget PH Sensor
Serial No.:	28096
Version:	101

Sensor Data

Measured PH: (-2.068 - 16.0722)	5.2011
Potential: (-527.11mV - 526.87mV)	104.52
Sensitivity:	0.5
Temperature:	20

1. Double Click on **Phidget PH Sensor** in the Phidget Control Panel to bring up PHSensor-full and check that the box labelled Attached contains the word True.
2. Check your probe by following the manufacturer's instructions. The results are displayed as measured pH and Potential in millivolts.
3. You can adjust the sensitivity by typing a decimal number between 0 and 14.
4. You can type in the temperature in degrees Celsius of the solution you are measuring.

If you are using Mac OS X

Go to www.phidgets.com >> Drivers

- Download Mac OS X Framework
- Click on System Preferences >> Phidgets (under Other) to activate the Preference Pane.
- Make sure that your Phidget is properly attached.
- Double click on the attached Phidget to launch the Example.

If you are using 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 or ARMV4I, 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.
- The Phidget APIs are designed to be used in an event-driven architecture. While it is possible to poll them, we don't recommend it. Please familiarize yourself with event programming.

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.

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 under Programming and are listed under the appropriate language.

API documentation

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 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 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.

Support

- Call the support desk at 1.403.282.7335 8:00 AM to 5:00 PM Mountain Time (US & Canada) - GMT-07:00
- E-mail support@phidgets.com

Technical Section

Electrode Fundamentals

The study of the complex operating principles of pH, ORP, and ion-selective electrodes is inappropriate for a brief technical manual. Many sources of information are available on the internet:

For information on pH: http://en.wikipedia.org/wiki/pH

For information on Glass pH Electrodes: http://en.wikipedia.org/wiki/Glass_electrode

For information on Oxidation/Reduction: http://en.wikipedia.org/wiki/Redox

For information on IS electrodes: http://en.wikipedia.org/wiki/Ion_selective_electrode

Choosing Electrodes

Review the data sheet for the electrode you have selected for your application carefully to ensure that it complies with the device specifications of the PhidgetPHSensor. Many electrodes will work but it is important to verify compliance before connecting an electrode to the Phidget. We have reviewed the following electrodes, and found that they can be used with the PhidgetPHSensor. This is by no means a comprehensive list, but can be used as a comparison with other electrodes if necessary.

Manufacturer	Web Page	Part Number
Omega	www.omega.com	PHE13XX, PHE14XX, ORE1311, ORE1411
Cole - Parmer	www.coleparmer.com	EW-59001, EW-27003
Mettler - Toledo	www.mt.com	InLab (BNC) Series

Temperature Compensation

The PhidgetPHSensor cannot measure the temperature of the solution. If the temperature of the solution is not 25 Degrees Celsius, the actual pH will not be exactly as measured. On the other hand, if the temperature is known, our APIs do provide a method to set the Temperature, allowing temperature compensation of pH.

Words of Caution

PhidgetPHSensor should be used to measure solutions that are 'electrically quiet'.

Measuring PH in electrically noisy environments such as tanks with mixing pumps, and even other measuring devices, is not recommended.

Discussion of measuring PH in a solution with Electromagnetic Interference

This is typically done by measuring the pH Voltage in a 'symmetric configuration' - that is, with a differential amplifier. In order for this to be effective, you need a pH Electrode with a 'solution ground'. On the meter side, the solution ground is attached to a Potential Matching Pin, sometimes called just a matching pin. This allows the amplifier grounds to stay close to the voltage of the solution. In the case of a fully isolated meter, the ground on the meter will track the potential of the solution. If the meter is not isolated, the solution ground connection will hopefully keep the solution and the meter at roughly the same ground. Since the amplifier is differential, the grounds do not have to be identical.

It may be possible to approximate a solution ground with just a piece of wire, instead of using a PH electrode with a built in solution ground. The piece of wire would likely not be as effective, especially if the conductivity of the solution is low.

Phidgets can be used in an electrically noisy environment by inserting a meter that can output an analog voltage. An example is the mV 600 or pH 500 models from Hanna Instruments. The output from these instruments can be configured to 0-5V. This output voltage would be wired into an Analog Input on a PhidgetInterfaceKit (set to non-Ratiometric).

API (Software Technical)

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, please refer to the device specifications.

Functions

double PH () [get]

Returns the measured pH. This value can range from between PHMin and PHMax, but some of this range is likely outside of the valid range of most pH sensors. For example, when there is no pH sensor attached, the board will often report an (invalid) pH of 15, which while technically within a valid pH range, is unlikely to be seen.

double PHMax () [get] : Constant

Returns the maximum ph that will be returned by the pH sensor input.

double PHMin () [get] : Constant

Returns the minimum pH that will be returned by the ph sensor input.

double PHChangeTrigger () [get,set]

An event that is issued when the pH varies by more than the PHChangeTrigger property.

double Potential () [get] : Millivolts

Returns the Potential. This returns the voltage potential measured by the A/D. This value will always be between getPotentialMin and getPotentialMax. This is the value that is internally used to calculate pH in the library.

double PotentialMax () [get] : Constant

Returns the maximum potential that will be returned by the pH sensor input.

double PotentialMin () [get] : Constant

Returns the minimum potential that will be returned by the pH sensor input.

double Temperature () [set] : Celsius

Sets the probe temperature in degrees Celsius. This value is used while calculating the pH. The default value in the library is 20 degrees Celsius. If the temperature of the liquid being measured is not 20 degrees, then it should be measured and set for maximum accuracy.

Note: All that this does is set a value in the library that is used for calculating pH. This does not set anything in the hardware itself.

Events

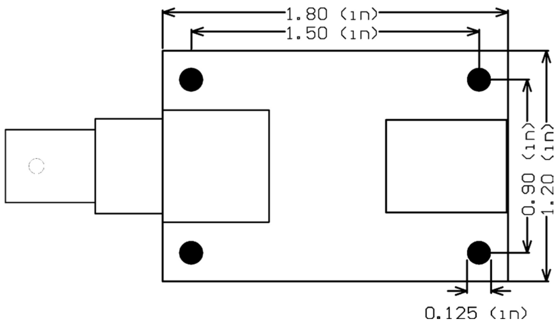
OnPHChange(double PH) [event]

An event issued when the pH changes by more than the PHChangeTrigger.

Device Specifications	
Input Range	+400mV to –400mV
Input Resolution	0.03mV
Input Impedance	>1 Tera Ohm
Update Rate	60 updates/second
USB Power Current Specification	500mA max
Device Current Consumption	25mA

Mechanical Drawing

1:1 scale



Product History

Date	Product Revision	Comment
March 2006	100	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