1048 User Guide

Jump to: navigation, search

Contents [hide]

Go to this device's product page

Getting Started

## Checking the Contents

You should have received:

- A PhidgetTemperatureSensor 4 input
- A Mini-USB Cable

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

- Thermocouples with stripped end

Connecting the Pieces

1. Connect the thermocouples to the terminal blocks on the 1048. Make sure that the thermocouple ground wire goes into the ground terminal. For example, if you are using a 3107 - Phidget K-type Thermocouple, the white wire is the ground wire.
2. Connect the 1048 to your PC using the Mini-USB cable.
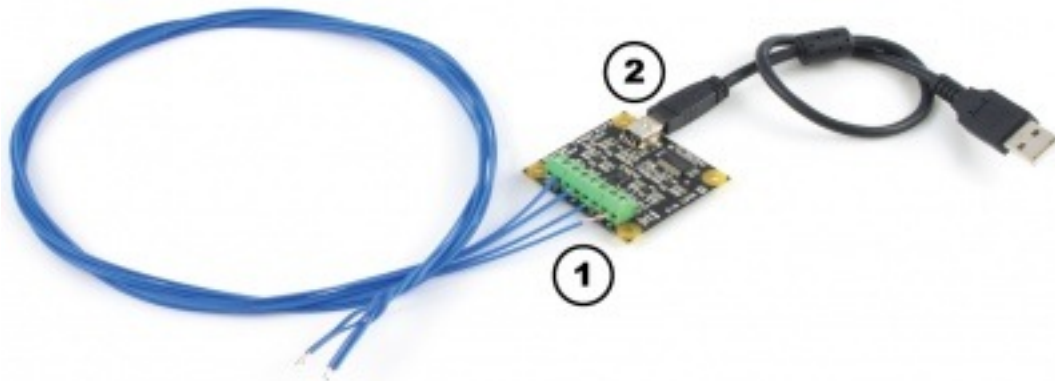
Testing Using Windows 2000 / XP / Vista / 7

Make sure you have the current version of the Phidget library installed on your PC. If you don't, follow these steps:

1. Go to the Quick Downloads section on the Windows page

2. Download and run the Phidget21 Installer (32-bit, or 64-bit, depending on your system)
3. You should see the Ph icon on the right hand corner of the Task Bar.

Running Phidgets Sample Program

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

The source code for the **TemperatureSensor-full** sample program can be found in the quick downloads section on the C# Language Page. If you'd like to see examples in other languages, you can visit our Languages page.

Updating Device Firmware

If an entry in this list is red, it means the firmware for that device is out of date. Double click on the entry to be given the option of updating the firmware. If you choose not to update the firmware, you can still run the example for that device after refusing.

Double Click on the Ph icon to activate the Phidget Control Panel and make sure that the Phidget Temperature Sensor 4-Input is properly attached to your PC.
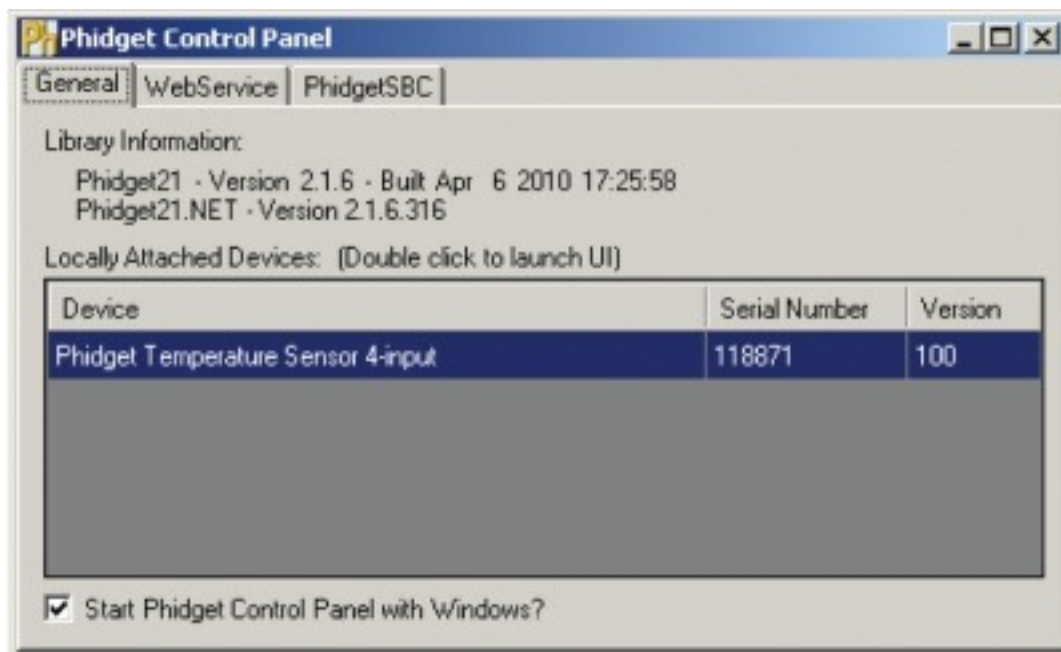
1. Double Click on Phidget Temperature Sensor 4-input in the Phidget Control Panel to bring up TemperatureSensor-full and check that the box labelled Attached contains the word True.
2. Select input 0

3. Select the thermocouple type (we're using K-type in this example). If you don't choose the correct type, your temperature results will be incorrect.
4. Touch a source of heat (light bulb, candle or lighter flame) with the thermocouple probe and watch the Thermocouple temperature increase.
5. The potential value will also increase as the thermocouple temperature increases.
6. The ambient temperature gives you the board temperature.
7. You can adjust the data sensitivity by moving the slider pointer.

TemperatureSensor-full

Device Info

Attached: **True** ①

Name: Phidget Temperature Sensor 4-input

Serial No.: 118871

Version: 100

Thermocouples: 4

Sensor Data

②

Choose Thermocouple: 0

Thermocouple Temperature:
(-200°C - 1250°C)   48.0211 ④

③

Thermocouple Type: K-Type

⑤

Potential: (-10.0094mV - 77.41038mV)   0.90599

Sensitivity: 0.1   ⑦

⑥

Ambient Temperature: (-55°C - 150°C)   25.875

Testing Using Mac OS X

1. Go to the Quick Downloads section on the Mac OS X page
2. Download and run the Phidget OS X Installer
3. Click on System Preferences >> Phidgets (under Other) to activate the Preference Pane
4. Make sure that the Phidget Temperature Sensor 4-Input is properly attached.
5. Double Click on Phidget Temperature Sensor 4-Input in the Phidget Preference Pane to bring up the TemperatureSensor-full Sample program. This program will function in a similar way as the Windows version.

Using Linux

For a step-by-step guide on getting Phidgets running on Linux, check the Linux page.

Using Windows Mobile / CE 5.0 / CE 6.0

For a step-by-step guide on getting Phidgets running on Windows CE, check the Windows CE page.

Technical Details

Cold Junction Compensation and Self-heating

Thermocouples consist of two junctions, one where the thermocouple meets the Phidget and one where the two wires are welded together at the sensing end of the device. In order to get accurate readings from the thermocouple one of these temperatures must be a known quantity. It doesn't make much sense to know the temperature at the end you are trying to sense at! Therefore it stands to reason that we need to know the temperature at the end where the thermocouple connects to the Phidget. This is why there is an ambient temperature sensor built onto the board.

An important thing to note though is that the ambient temp sensor measures the temperature of the air around the board and not specifically the junction. Generally you can assume they are nearly one and the same value, however as the electronics get hot due to being powered up there can be some small error introduced. This is exacerbated by having the board in an enclosed space where normal airflow is restricted thereby increasing the effect of self-heating. As a result we recommend that the board be left in as open and well ventilated/cooled a place as possible to minimize this error source.

For more information on thermocouples, check out the Thermocouple Primer.

API

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 in the Quick Downloads section for that language. For exact values, refer to the device specifications.

Functions

int TemperatureInputCount () [get] : Constant

Returns the number of thermocouples that can be read by the PhidgetTemperatureSensor.

double Temperature (int ProbeIndex) [get] : Celsius

Returns the temperature of a thermocouple. This value is returned in degrees Celsius but can easily be converted into other units. This value will always be between TemperatureMin and TemperatureMax. Please refer to the device specifications for noise and accuracy details.

double TemperatureMax (int ProbeIndex) [get] : Constant, Celsius

Returns the maximum temperature that can be returned by a thermocouple input. This value depends on the thermocouple type.

double TemperatureMin (int ProbeIndex) [get] : Constant, Celsius

Returns the minimum temperature that can be returned by a thermocouple input. This value depends on the thermocouple type.

double TemperatureChangeTrigger(int ProbeIndex) [get,set] : Celsius

Sets the change trigger for an input. This is the amount by which the sensed temperature must change between TemperatureChangeEvents. By default this is set to 0.5. Setting TemperatureChangeTrigger to 0 will cause all temperature updates to fire events. This is helpful for applications that are implementing their own filtering.

double Potential (int ProbeIndex) [get] : Millivolts

Returns the potential of a thermocouple input. This value is returned in millivolts, and will always be between getPotentialMin and getPotentialMax. This is the value that is internally used to calculate temperature in the library.

double PotentialMax (int ProbeIndex) [get] : Constant, Millivolts

Returns the maximum voltage that can be measured by the 1048.

double PotentialMin (int ProbeIndex) [get] : Constant, Millivolts

Returns the minimum voltage that can be measured by the 1048.

double AmbientTemperature () [get] : Celsius

Returns the temperature of the 1048 board, measured near the thermocouple connector. This temperature is used as a reference for the thermocouple voltage. This value will always be between getAmbientTemperatureMin and getAmbientTemperatureMax.

double AmbientTemperatureMax () [get] : Constant, Celsius

Returns the maximum temperature that can be returned by the ambient sensor.

double AmbientTemperatureMin () [get] : Constant, Celsius

Returns the minimum temperature that can be returned by the ambient sensor.

int ThermocoupleType(int ProbeIndex) [get,set]

Returns/sets the thermocouple type for an input. The possible values are J, K, E, and T, corresponding to K, E, J and T-Type Thermocouples. Support for other thermocouple types, and voltage sources other than thermocouples in the valid range (between getPotentialMin and getPotentialMax) can be achieved using getPotential.
Note that this function will vary widely between APIs. Please refer to the Manual for the API you are programming against for exact calling conventions.

Events

OnTemperatureChange(int ProbeIndex, double Temperature) [event]

Event that fires whenever the temperature changes by more than the TemperatureChangeTrigger.

OnError(int code, String description) [event]

PhidgetTemperatureSensor returns EEPHIDGET_OUTOFRANGE when an input goes out of the range specified by PotentialMin-PotentialMax or TemperatureMin-TemperatureMax. The event is reported once per input per out-of-range type, until the value has gone back into range. The description specifies the input, the type of out-of-range error and the out of range value that was read. This should only happen for inputs which don't have a thermocouple attached.

Product History

Date	Board Revision	Device Version	Comment
April 2010	0	100	Product Release
May 2010	0	101	fixed setlabel
May 2011	0	102	getLabelString fixed for labels longer than 7 characters