Labview Manual

Labview Manual

© 2011 Phidgets Inc.

Table of Contents

Part 5   Specific Modules   58

Index 364

1   Welcome to Phidgets

LabVIEW Help

November 2010, Version 0.1-021110

This help system includes information about LabVIEW programming for each Phidget device. It contains programming concepts, step-by-step instructions, and reference information about VIs, functions and palettes.

To navigate this help system, use the **Contents**, **Index**, and **Search** tabs to the left of this windows.

Phidgets are an easy to use set of building blocks for low cost sensing and control from your PC. Using the Universal Serial Bus (USB) as the basis for all Phidgets, the complexity is managed behind this easy to use and robust Application Program Interface (API) library.

This help system may link to Portable Document Format (PDF) versions of documents. You must have Adobe Reader installed to view or search the PDF versions of these manuals.

> **Note: (Mac OS X)** Phidgets recommends that you use Safari 1.3.2 or later or Firefox 1.0.2 or later to view the *Help*. **(Linux)** Phidgets recommends that you use Mozilla 1.2 or later or Firefox 1.0.2 or later to view the *Help*.

For more information about this help, refer to the following topics:

Introduction

Programming Concept

Phidgets Common

Specific Modules

Phidgets Constant

2   Introduction

Phidgets are an easy to use set of building blocks for low cost sensing and control from your PC. Using the Universal Serial Bus (USB) as the basis for all Phidgets, the complexity is managed behind this easy to use and robust Application Program Interface (API) library.

This manual documents the Phidgets software programming model in National Instruments Labview language. The **Programming Concept** should be the first section to be read for someone beginning to use Phidgets. After the concepts described are understood, users can read **Phidgets Common** and **Specific Modules** for function reference and device documentation in general. Note that these sections are light on function documentation - generally only containing specific reference information and basic function information.

For a more detailed introduction, please refer to **Understanding Phidgets** and **Platform Support**.

2.1   Understanding Phidgets

Hardware Model

All Phidgets are connected to the computer using USB. Most computers support up to 127 USB devices (or more), so it is easy to connect as many Phidgets as are required for almost any project. Phidgets can be connected either directly to a computer or through Hubs, but there are some limitations.

The maximum cable length for USB is 15 feet. This is a maximum distance between device and computer, even if there are one or more Hubs in between. There are cable extenders available on the market, but these can be unreliable and are not endorsed by Phidgets Inc. Users should never try to run USB over anything other then a certified USB cable, and should never try to run it longer than the spec.

Phidgets run as USB 1.1 low speed or full speed devices, and are supported by both USB 1.1 and USB 2.0 hosts.

Software Model

The Phidgets Labview library is written under the C library - phidget21, which implements the low-level protocols necessary to communicate with the Phidgets, and exports a unified interface to the software programmer. This also makes the Phidgets Labview library cross-platform.

The Phidget Labview library contains only glue logic for interfacing with the C library, thus making maintenance much easier. It should be noted the library employs threading and events extensively. (See Programming Concept for more information.)

2.2 Platform Support

Operating System Support

Windows
Microsoft Windows 2000 and later are supported, including 64-bit editions. The Windows libraries are installed using an MSI installer that can be found on the Phidgets web site. This installs the C library, the .NET library, the COM library, the Java library, the Phidget Web Service and the Phidget Control Panel.

The Phidget Control Panel is represented by a "Ph" icon that runs in the system tray (usually on the right end of the Windows task bar). This program can be used to list and control any Phidgets attached to the system, and to control the Web Service.

Mac OS X
Mac OS X 10.3.9 and newer on Intel and PPC are supported. The Mac libraries are distributed in a .dmg and are installed using a standard Mac package installer. This installs the C library, the Kernel driver, the Java library, the Phidget Web Service and the Phidget Preference Pane.

The Phidget Preference Pane is a preference pane which resides in System Preferences. This program can be used to list and control any Phidgets attached to the system, and to control the Web Service.

Linux
Linux version 2.4 is supported, including 64-bit editions, but 2.6.7 or newer is recommended. The Linux libraries are distributed as source. The source for the C library, with optional JNI (Java support) extensions and the source for the Phidget Web Service are available as a .tar.gz. The included Makefile makes it easy to build and install the libraries on most Linux distribution.

Other
Other Operating System support is not currently available.

Labview Version Support

The Phidgets Labview library supports 32-bit Labview version 7.1.1 or higher. It also supports 64-bit Labview version 2009 or higher.

Note: If you use Labview 64-bit, you need to install the Phidgets Labview 64-bit library.

3   Programming Concept

This manual is designed such that both novice and expert users can quickly reference the various Phidget LabVIEW functions.

The manual is subdivided into 6 sections: **Getting Started**, **Event Handler**, **Multiple Devices**, **Phidgets Common, Specific Modules** and **Phidgets Constants**.

Each section is defined as follow:

Getting Started: tells users how to communicate with phidgets and perform some basic functions. Use the **Getting Started** manual as a tutorial to familiarize yourself with Phidget LabVIEW functions and basic features you use to build data acquisition and instrument control applications.

Event Handler: needs only be used in applications that need to receive events. Use the **Event Handler** manual as a tutorial to learn how to construct the event and use the handler.

Multiple Devices: needs only be used in applications that involve multiple phidgets. Use the **Multiple Devices** manual as a tutorial to configure and control many phidgets in one VI.

Phidgets Common: contains the common functions for all phidgets.

Specific Modules: contains all the functions for specific phidgets.

Phidgets Constants: explains all Phidgets pre-defined constants.

Note: It is *important* for user to upgrade the Phidgets21 library to the most recent version. Click **here** to check and download the latest version library.

3.1   Getting Started

Phidgets are an easy to use set of building blocks for low cost sensing and control from your PC. Using the Universal Serial Bus (USB) as the basis for all Phidgets, the complexity is managed behind this easy to use and robust Application Program Interface (API) library.

As such, the Phidgets Labview VI features a very simple and easy-to-use set of VIs. At the simplest level, all you have to do to control a Phidget is explained in this section.

For illustration purposes, Phidget Accelerometer will be used.

Phidgets Accelerometer Example
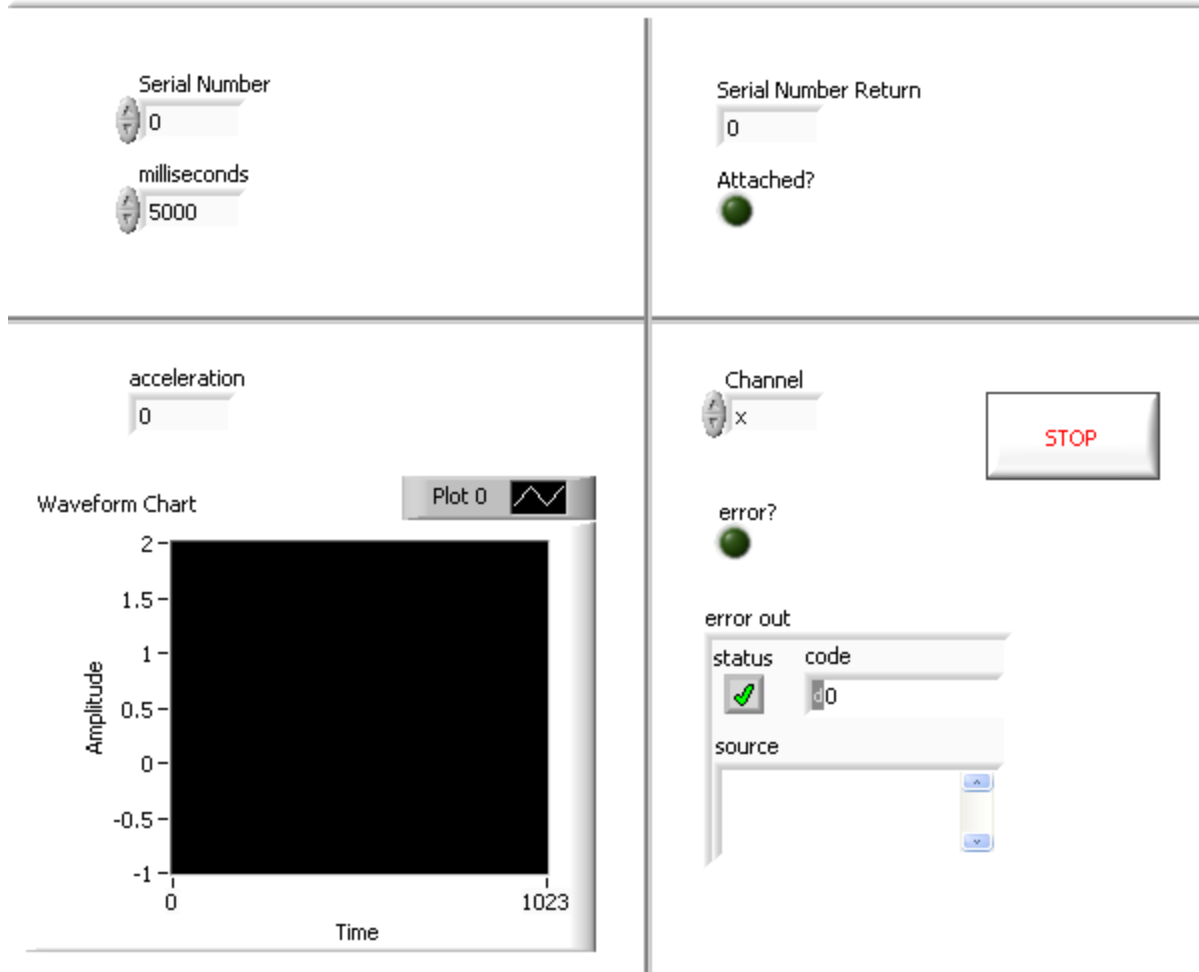
Serial Number
0

milliseconds
5000

Serial Number Return
0

Attached?

acceleration
0

Channel
x

STOP

error?

error out

status   code
0

source

Waveform Chart   Plot 0

Amplitude
2 -
1.5 -
1 -
0.5 -
0 -
-0.5 -
-1 -
0   1023
Time

> **Note:** For a more specific illustration on different phidgets, users can refer to different examples accordingly.

Open the diagram of the "Single control example.vi" under the "Accelerometer" folder.
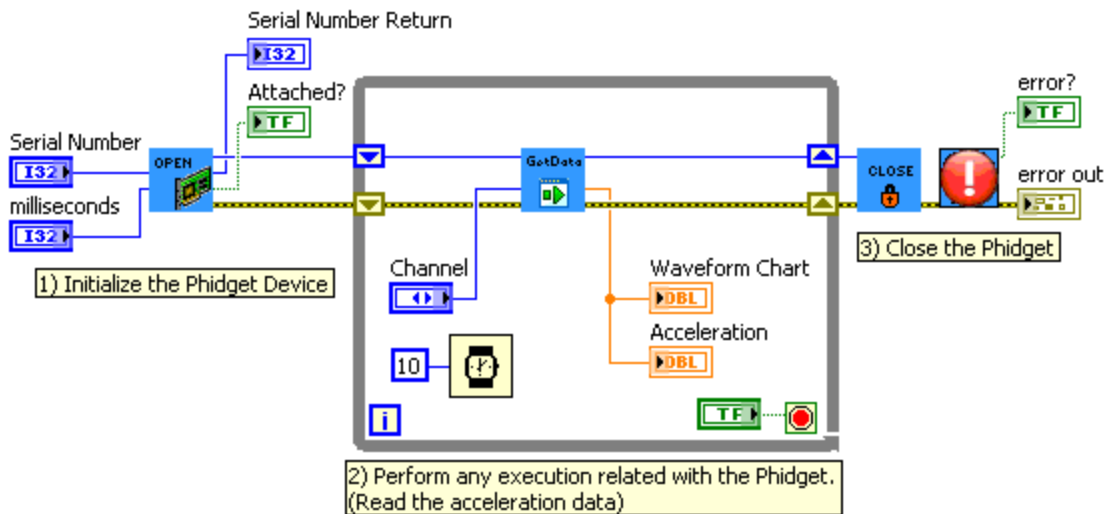
Phidgets can be programmed into 3 steps:

<Step1> Initialize the Phidget. This includes opening a Phidget hardware, creating a Phidget handler or setting up parameters of a Phidget.
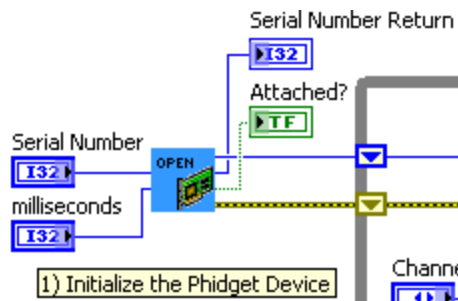
Users can also acquire other information in this step. For more details, please refer to the example called "Remote Example.vi" under "TemperatureSensor" folder.

If users want to open Phidgets over the network, a PhidgetOpenRemote and AcceCreate functions will be called instead of AcceOpen.

<Step2> Perform any execution related with the Phidget. This includes data acquisition, device control, event execution etc.

<Step3> Close the Phidget. This may include closing the device, releasing all the resources or freeing a Phidget handle and an error handler.

3.2   Event Handler

This demonstrates how to call a Phidgets event and how to use them.

For illustration purposes, Phidget Spatial will be used.
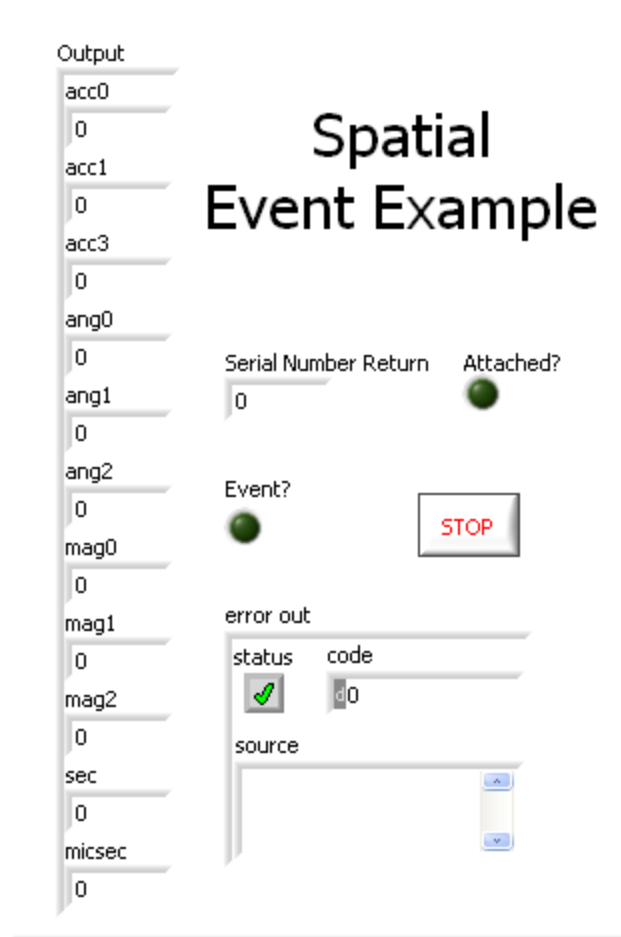
> **Note:** For a more specific illustration on different phidgets, users can refer to different examples accordingly.

Open the diagram of the "Spatial event example.vi" under the "Spatial" folder.
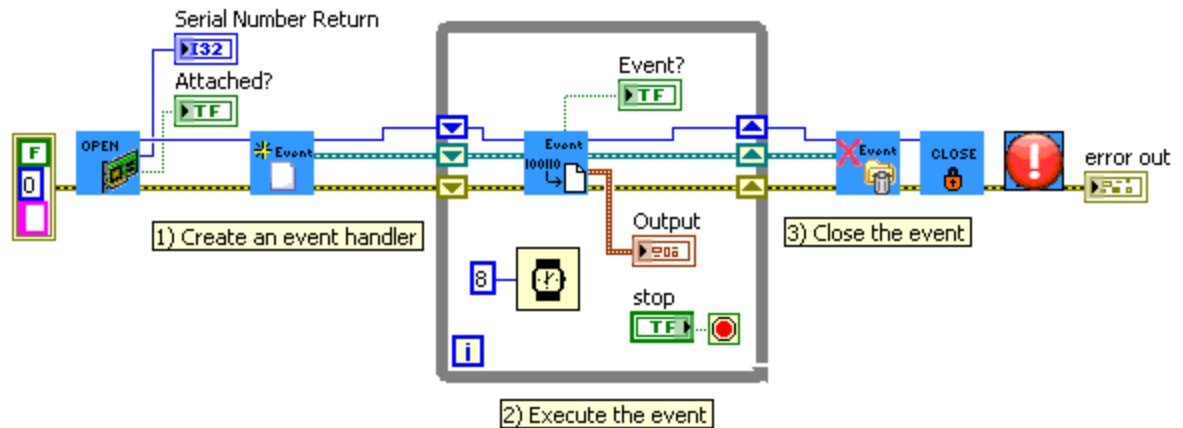
Phidgets Event can be programmed into 3 steps:

<Step1> Create an event handler.

<Step2> Perform any event execution related with the Phidget.

<Step3> Close the related event.

3) Close the event

> **Note:** When execute an event, please make sure "create", "execute" and "close" the same event. For example, user wants to run an Event called A. He has to place "CreateEventA.vi", "ExeEventA.vi" and "CloseEventA.vi" on the block diagram. He cannot place "CloseEventB.vi" instead of "CloseEventA.vi". However, for some Phidgets event, they share same VIs. (For more details, please refer to specific Phidgets.)

Phidget Error Event:

Another useful example will be "Temp On Error.vi" under "TemperatureSensor" folder.

In this example, the error event is actually located in "Common Functions" folder. It will detect asynchronous errors from Phidgets.

> **Note:** Phidgets can also support multiple events. For an example, please refer to "Event handler example.vi" under the "InterfaceKit" folder.

3.3   Multiple Devices

Phidgets Labview can control multiple phidgets. As long as the handlers are different, different phidgets can run in parallel.

For a more detailed illustration, please refer to the example of "Multiple control example (Parallel).vi" and "Multiple control example (Advanced).vi" under the "Accelerometer" folder.
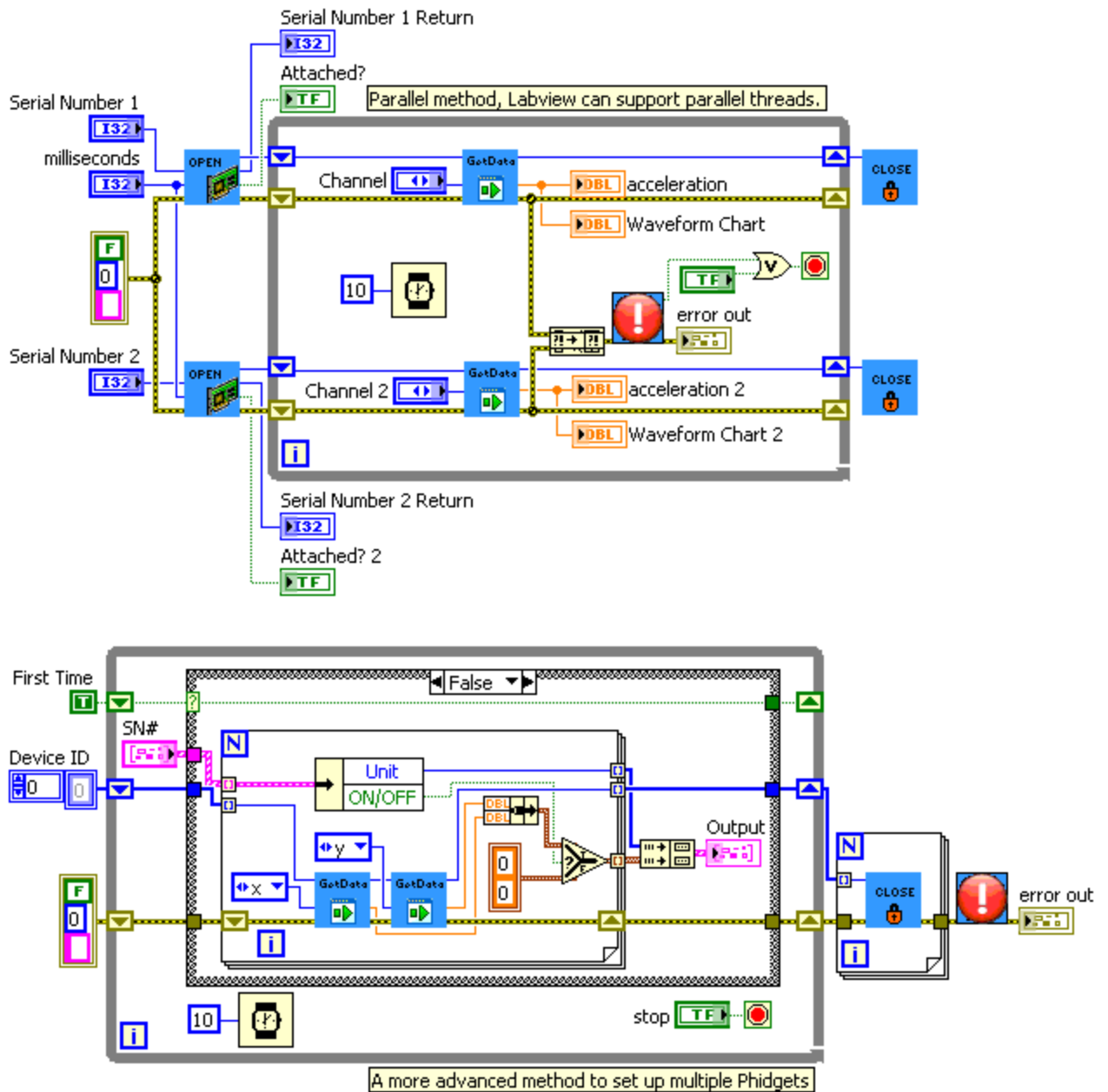
Note: Another example will be "TemperatureDisplay.vi" under "TextLCD" folder. This example shows how to combined use different phidgets.

4   Phidgets Common

This section describes the VI functions used by all Phidgets. The SubVI folder contains advanced
LabVIEW VIs

ErrorHandler

PhidgetClose

PhidgetDelete

PhidgetEventCloseOnError

PhidgetEventCreateOnError

PhidgetEventExeOnError

PhidgetGetDeviceClass

PhidgetGetDeviceID

PhidgetGetDeviceLabel

PhidgetGetDeviceName

PhidgetGetDeviceType

PhidgetGetDeviceVersion

PhidgetGetServerStatus

PhidgetLibraryVersion

PhidgetOpen

PhidgetOpenRemote

PhidgetOpenRemoteIP

PhidgetServerAddress

PhidgetServiceID

PhidgetSetDeviceLabel

SubVIs (Folder)

4.1 ErrorHandler

Description:

Create a Phidget Error handle.

Connector Pane:

Controls and Indicators:

Input

error in (no error)

Describes error conditions that occur before this node runs.

Output

error?

ON if error occurs.

error out

Contains error information.

4.2   PhidgetClose

Description:

Close a Phidget handle.
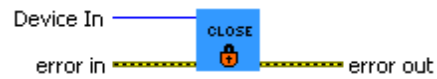
Connector Pane:

Device In
error in   error out

Controls and Indicators:

Input

Device In
[I32]   Device # identification.

error in (no error)
                          Describes error conditions that occur before this node runs.

Output

error out
                          Contains error information.

4.3   PhidgetDelete

Description:

Delete a Phidget handle.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output

error out

Contains error information.

4.4   PhidgetEventCloseOnError

Description:

Close the the error handle.

Connector Pane:

Device In ——————————— Device Out
Event Registration Refnum In
error in (no error) ——————————— error out

Controls and Indicators:

Input

Device In
[I32]   Device # identification.

Event Registration Refnum In
[D]   Event # identification.

error in (no error)
[error]   Describes error conditions that occur before this node runs.

Output

Device Out
[I32]   Same as the Device In.

error out
[error]   Contains error information.

4.5   PhidgetEventCreateOnError

Description:

Set up an error event handle.

Connector Pane:

Device In ———————— Device Out
                    Event Registration Refnum Out
error in (no error) ———————— error out

Controls and Indicators:

Input

Device In

Device # identification.

error in (no error)

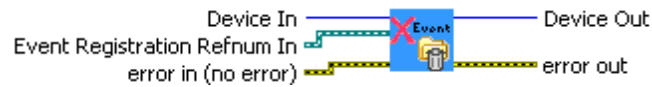Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

Event Registration Refnum Out

Event # identification.

error out

Contains error information.

4.6 PhidgetEventExeOnError

Description:

This is called when an asynchronous error occurs.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

Event Registration Refnum In

Event # identification.

error in (no error)

Describes error conditions that occur before this node runs.
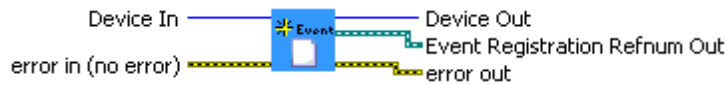
Output

Device Out

Same as the Device In.

Event Registration Refnum Out

Same as the Event Registration Refnum In.
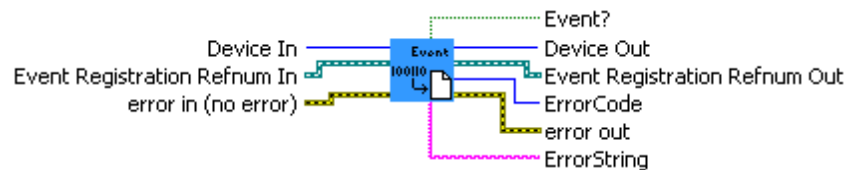
Event?

Returns the event status. (Executed is T; Not executed is F)

ErrorCode

The error code to get the description of.

ErrorString

Contain the error description string.

error out

Contains error information.

4.7   PhidgetGetDeviceClass

Description:

Get the class of a Phidget.

Connector Pane:

Controls and Indicators:

Input

Device In | Device # identification. This function will create a new device identification if it's 0 or invalid.

error in (no error) | Describes error conditions that occur before this node runs.

Output

Device Out | Same as the Device In.

DeviceClass | Returns the device class constant.

error out | Contains error information.

4.8 PhidgetGetDeviceID

Description:

Get the device ID of a Phidget.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification. This function will create a new device identification if it's 0 or invalid.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

DeviceID

Returns the device ID constant.

error out

Contains error information.

4.9   PhidgetGetDeviceLabel

Description:

Get the label of a Phidget.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification. This function will create a new device identification if it's 0 or invalid.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

DeviceLabel

Returns the device label.

error out

Contains error information.

4.10   PhidgetGetDeviceName

Description:

Get the specific name of a Phidget.

Connector Pane:

Device In ———————
Device Out
DeviceName
error in
error out

Controls and Indicators:

Input

Device In

Device # identification. This function will create a new device identification if it's 0 or invalid.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

DeviceName
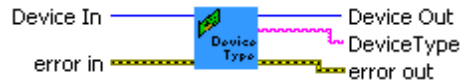
Returns the device name.

error out

Contains error information.

4.11   PhidgetGetDeviceType

Description:

Get the type (class) of a Phidget.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification. This function will create a new device identification if it's 0 or invalid.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

DeviceType

Returns the device type.

error out

Contains error information.

4.12   PhidgetGetDeviceVersion

Description:

Get the firmware version of a Phidget.

Connector Pane:

Controls and Indicators:

Input

Device In

I32

Device # identification. This function will create a new device identification if it's 0 or invalid.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

I32

Same as the Device In.

Version

I32

Returns the device version.

error out

Contains error information.

4.13   PhidgetGetServerStatus

Description:

Get the connected to server status of a remotely opened Phidget. This will fail if the Phidget was opened locally.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification. This function will create a new device identification if it's 0 or invalid.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

ServerStatus

Returns the server status. Possible values are 0 for unattached, 1 for attached and others for undefined.

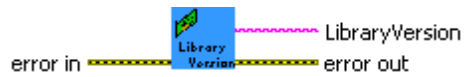Attached?

The server status.

error out

Contains error information.

4.14   PhidgetLibraryVersion

Description:

Get the library version. This contains a version number and a build date.

Connector Pane:

Controls and Indicators:

Input

error in (no error)

Describes error conditions that occur before this node runs.

Output

LibraryVersion

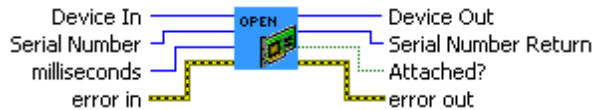Returns the library version.

error out

Contains error information.

4.15   PhidgetOpen

Description:

Open a Phidget locally.

Connector Pane:

Controls and Indicators:

Input

Device In
I32

Device # identification. This function will create a new device identification if it's 0 or invalid.

Serial Number
I32

Serial number. Specify -1 to open any.

milliseconds
I32

Time to wait for the attachment. Specify 0 to wait forever. (Default is 5000)

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out
I32

Same as the Device In.

Serial Number Return
I32

Returns the serial number.

Attached?
TF

Returns the device status. (Attached is T; Not attached is F)

error out

Contains error information.

4.16 PhidgetOpenRemote

Description:

Open a Phidget remotely by ServerID. Note that this requires Bonjour (mDNS) to be running on both the host and the server.

Connector Pane:

serverID
Device In   Device Out
Serial Number   Serial Number Return
milliseconds   Attached?
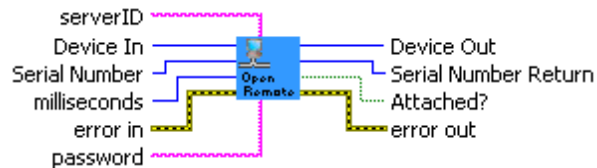error in   error out
password

Controls and Indicators:

Input

Device In
`I32`
Device # identification. This function will create a new device identification if it's 0 or invalid.

Serial Number
`I32`
Serial number. Specify -1 to open any.

milliseconds
`I32`
Time to wait for the attachment. Specify 0 to wait forever. (Default is 5000)

serverID
`abc`
Server ID. Specify NULL to open any.

password
`abc`
Password. Can be NULL if the server is running without password.

error in (no error)
Describes error conditions that occur before this node runs.

Output

Device Out
`I32`
Same as the Device In.

Serial Number Return
`I32`
Returns the serial number.

Attached?
`TF`
Returns the device status. (Attached is T; Not attached is F)

error out
Contains error information.

4.17   PhidgetOpenRemoteIP

Description:

Open a Phidget remotely by address and port.

Connector Pane:

Controls and Indicators:

Input

Device In
`[I32]`
Device # identification. This function will create a new device identification if it's 0 or invalid.

Serial Number
`[I32]`
Serial number. Specify -1 to open any.

milliseconds
`[I32]`
Time to wait for the attachment. Specify 0 to wait forever. (Default is 5000)

address
`[abc]`
Address. This can be a hostname or IP address.

password
`[abc]`
Password. Can be NULL if the server is running without password.

port
`[I32]`
Port number. Default is 5001.

error in (no error)
`[...]`
Describes error conditions that occur before this node runs.

Output

Device Out
`[I32]`
Same as the Device In.

Serial Number Return
`[I32]`
Returns the serial number.

Attached?
`[TF]`
Returns the device status. (Attached is T; Not attached is F)

error out
`[...]`
Contains error information.

4.18   PhidgetServerAddress

Description:

Get the address and port of a remotely opened Phidget. This will fail if the Phidget was opened locally.
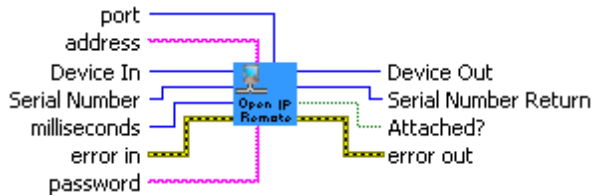
Connector Pane:
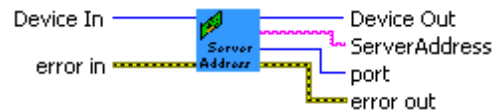
Controls and Indicators:

Input

Device In

Device # identification. This function will create a new device identification if it's 0 or invalid.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

ServerAddress

Returns the address.

port

Returns the port number.

error out

Contains error information.

4.19   PhidgetServiceID

Description:

Get the server ID of a remotely opened Phidget. This will fail if the Phidget was opened locally.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification. This function will create a new device identification if it's 0 or invalid.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

ServiceID

Returns the server ID.

error out

Contains error information.

4.20 PhidgetSetDeviceLabel

Description:

Set the label of a Phidget. Note that this is not supported on very old Phidgets, and not yet supported in Windows.

Connector Pane:
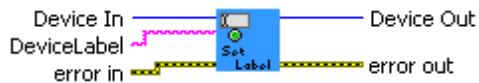
Controls and Indicators:

Input

Device In
I32

Device # identification. This function will create a new device identification if it's 0 or invalid.

DeviceLabel
abc

The label to be set.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out
I32

Same as the Device In.

error out

Contains error information.

4.21 SubVIs

This contains a set of advanced LabVIEW VIs that provide a firmware access to all phidgets. This should not be direct accessed by users.

_AttachCHK

_ChkError

_Close

_Delete

_Open

_OpenRemote

_OpenRemoteIP

_SerialReturn

_WaitAttach

EventCloseIntDouble

EventCloseIntInt

EventExeIntDouble

EventExeIntInt

4.21.1 _AttachCHK

Description:

Get the attach state of a Phidget.

Connector Pane:
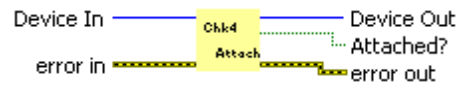
Controls and Indicators:

Input

Device In

Device # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

Attached?

The attach state (Boolean type).

error out

Contains error information.

4.21.2 _ChkError

Description:

Check the error of a Phidget.

Connector Pane:
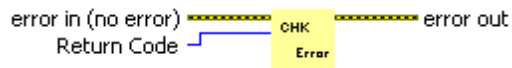
error in (no error) ▭▭▭▭▭ CHK ▭▭▭▭▭ error out
        Return Code ⌐   Error

Controls and Indicators:

Input

error in (no error)

Describes error conditions that occur before this node runs.

Return Code

The error code to get the description of.

Output

Device Out

Same as the Device In.

error out

Contains error information.

4.21.3 _Close

Description:

Close a Phidget device.

Connector Pane:

Controls and Indicators:

Input

Device In

I32   Device # identification.

error in (no error)

            Describes error conditions that occur before this node runs.

Output

Device Out

I32   Same as the Device In.

error out

            Contains error information.

4.21.4 _Delete

Description:

Delete a Phidget handler.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output

error out

Contains error information.

4.21.5 _Open

Description:

Open a Phidget device.

Connector Pane:

Controls and Indicators:

Input

Device In

I32   Device # identification.

Serial Number

I32   Serial number. Specify -1 to open any. (Default: -1)

error in (no error)

    Describes error conditions that occur before this node runs.

Output

Device Out

I32   Same as the Device In.

error out

    Contains error information.

4.21.6 _OpenRemote

Description:

Open a Phidget remotely by ServerID. Note that this requires Bonjour (mDNS) to be running on both the host and the server.

Connector Pane:
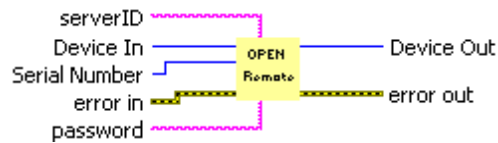
Controls and Indicators:

Input

Device In

Device # identification. This function will create a new device identification if it's 0 or invalid.

Serial Number

Serial number. Specify -1 to open any.

serverID

Server ID. Specify NULL to open any.

password

Password. Can be NULL if the server is running without password.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

error out

Contains error information.

4.21.7 _OpenRemoteIP

Description:

Open a Phidget remotely by address and port.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification. This function will create a new device identification if it's 0 or invalid.

Serial Number

Serial number. Specify -1 to open any.

address

Address. This can be a hostname or IP address.

password

Password. Can be NULL if the server is running without password.

port

Port number. Default is 5001.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

error out

Contains error information.

4.21.8 _SerialReturn

Description:

Return the serial number of a Phidget.

Connector Pane:
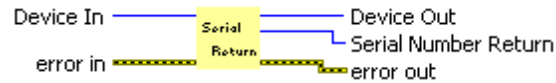
Controls and Indicators:

Input

Device In

[I32]   Device # identification.

error in (no error)

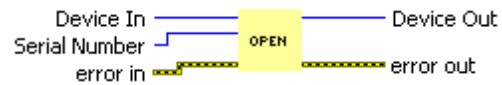[error]   Describes error conditions that occur before this node runs.

Output

Device Out

[I32]   Same as the Device In.

Serial Number Return

[I32]   Returns the serial number.

error out

[error]   Contains error information.

4.21.9 _WaitAttach

Description:

Wait until a Phidget attached.

Connector Pane:
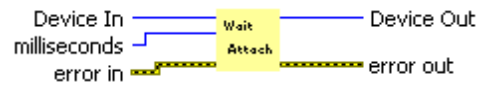
Controls and Indicators:

Input

Device In

[I32]   Device # identification.

milliseconds

[I32]   Time to wait for the attachment. Specify 0 to wait forever. (Default is 5000)

error in (no error)

[ ]   Describes error conditions that occur before this node runs.

Output

Device Out

[I32]   Same as the Device In.

error out

[ ]   Contains error information.

4.21.10 EventCloseIntDouble

Description:

Close the event handler which contains an integer and a double event variables. This should not be directly accessed by users.

Connector Pane:

Controls and Indicators:

Input

Device In

I32   Device # identification.

Event Registration Refnum In

        Event # identification.

error in (no error)

        Describes error conditions that occur before this node runs.

Output

Device Out

I32   Same as the Device In.

error out

        Contains error information.

4.21.11 EventCloseIntInt

Description:

Close the event handler which contains an integer and an integer event variables. This should not be directly accessed by users.

Connector Pane:

Controls and Indicators:

Input

Device In

I32   Device # identification.

Event Registration Refnum In

           Event # identification.

error in (no error)

           Describes error conditions that occur before this node runs.

Output

Device Out

I32   Same as the Device In.

error out

           Contains error information.

4.21.12 EventExeIntDouble

Description:

Call the event handler which contains an integer and a double event variables. This should not be directly accessed by users.

Connector Pane:
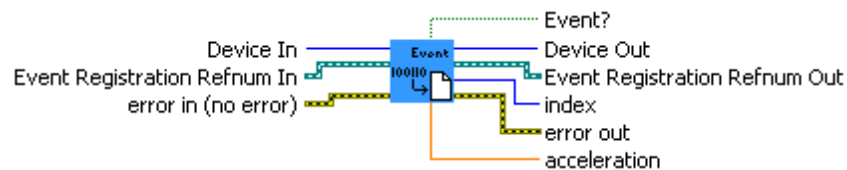
Controls and Indicators:

Input

Device In
Device # identification.

Event Registration Refnum In
Event # identification.

error in (no error)
Describes error conditions that occur before this node runs.

Output

Device Out
Same as the Device In.

Event Registration Refnum Out
Same as the Event Registration Refnum In.

Event?
Returns the event status. (Executed is T; Not executed is F)

index
The integer.

acceleration
The double.

error out
Contains error information.

4.21.13 EventExeIntInt

Description:

Call the event handler which contains an integer and an integer event variables. This should not be directly accessed by users.

Connector Pane:
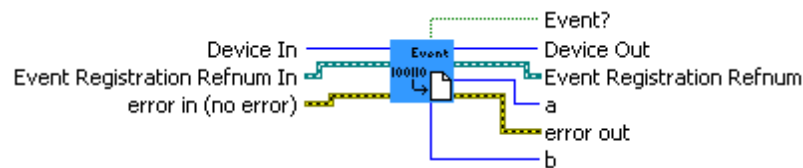
Controls and Indicators:

Input

Device In
Device # identification.

Event Registration Refnum In
Event # identification.

error in (no error)
Describes error conditions that occur before this node runs.

Output

Device Out
Same as the Device In.

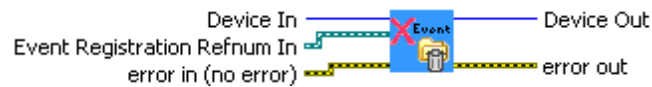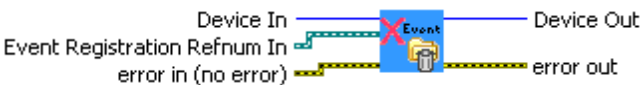Event Registration Refnum Out
Same as the Event Registration Refnum In.

Event?
Returns the event status. (Executed is T; Not executed is F)

a
The integer.

b
The integer.

error out
Contains error information.

5   Specific Modules

This section describes each of the VI function used by different Phidgets. All the VI functions are located in its dll folder correspondingly.

Note: Refer to the Product manual for your Phidget and the C Programming Manual for more detailed, language unspecific API documentation.

Please select a specified module accordingly.

Phidget Accelerometer

Phidget Advanced Servo

Phidget Encoder

Phidget InterfaceKit

Phidget IR

Phidget LED

Phidget Motor Control

Phidget PH Sensor

Phidget RFID

Phidget Servo

Phidget Spatial

Phidget Stepper

Phidget Temperature Sensor

Phidget TextLCD

Phidget TextLED (Discontinued)

Phidget Weight Sensor (Discontinued)

5.1   Phidget Accelerometer

This contains VI functions for Phidget Accelerometer. See the product manual for more specific API details, supported functionality, units, etc.

AcceAxisCount

AcceCreate

AcceEventClose

AcceEventCreate

AcceEventExe

AcceGetData

AcceGetMax

AcceGetMin

AcceGetTrigger

AcceOpen

AcceSetTrigger

5.1.1   AcceAxisCount

Description:

Get the number of acceleration axes supported by this accelerometer.

Connector Pane:
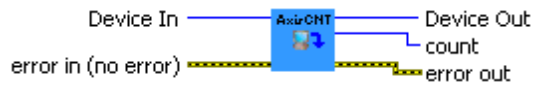
Controls and Indicators:

Input

Device In

Device # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

count

The axis count.

error out

Contains error information.

5.1.2   AcceCreate

Description:

Create a Phidget Accelerometer handle.

Connector Pane:
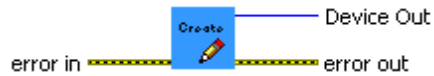
Controls and Indicators:

Input

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Device # identification.

error out

Contains error information.

5.1.3   AcceEventClose

Description:

Close the acceleration change event handle.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

Event Registration Refnum In

Event # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

error out

Contains error information.

5.1.4 AcceEventCreate

Description:

Set up an acceleration change event handle.

Connector Pane:

Device In ——————— Device Out
⁜ Event
error in (no error) ╌╌╌╌╌╌╌ Event Registration Refnum Out
error out

Controls and Indicators:

Input

Device In
[I32] Device # identification.

error in (no error)
Describes error conditions that occur before this node runs.

Output

Device Out
[I32] Same as the Device In.

Event Registration Refnum Out
Event # identification.

error out
Contains error information.

5.1.5 AcceEventExe

Description:

This is called when the acceleration changes by more then the change trigger.

Connector Pane:

Controls and Indicators:

Input

Device In
$\boxed{\text{I32}}$ Device # identification.

Event Registration Refnum In
$\boxed{\text{□}}$ Event # identification.

error in (no error)
$\boxed{\text{ }}$ Describes error conditions that occur before this node runs.

Output

Device Out
$\boxed{\text{I32}}$ Same as the Device In.

Event Registration Refnum Out
$\boxed{\text{□}}$ Same as the Event Registration Refnum In.

Event?
$\boxed{\text{TF}}$ Returns the event status. (Executed is T; Not executed is F)

index
$\boxed{\text{I32}}$ The acceleration index.

acceleration
$\boxed{\text{DBL}}$ The acceleration.

error out
$\boxed{\text{ }}$ Contains error information.

5.1.6   AcceGetData

Description:

Get the current acceleration data of an axis.

Connector Pane:

Controls and Indicators:

Input

Device In

[I32]   Device # identification.

Channel

[◄►]   The acceleration index. (x, y, z)

error in (no error)

[≡≡]   Describes error conditions that occur before this node runs.

Output

Device Out

[I32]   Same as the Device In.

acceleration

[DBL]   The acceleration.

error out

[≡≡]   Contains error information.

5.1.7   AcceGetMax

Description:

Get the maximum acceleration supported by an axis.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

Channel

The acceleration index. (x, y, z)

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

acce_max

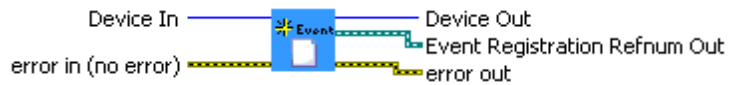The maximum acceleration.

error out

Contains error information.

5.1.8   AcceGetMin

Description:

Get the minimum acceleration supported by an axis.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

Channel

The acceleration index. (x, y, z)

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

acce_min

The minimum acceleration.

error out

Contains error information.

5.1.9   AcceGetTrigger

Description:

Get the change trigger for an axis.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

Channel

The acceleration index. (x, y, z)

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

Tigger Out

The change trigger.

error out

Contains error information.

5.1.10 AcceOpen

Description:

Open a Phidget Accelerometer.

Connector Pane:

Controls and Indicators:

Input

Device In

[I32]

Device # identification. This function will create a new device identification if it's 0 or invalid.

Serial Number

[I32]

Serial number. Specify -1 to open any.

milliseconds

[I32]

Time to wait for the attachment. Specify 0 to wait forever. (Default is 5000)

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

[I32]

Same as the Device In.

Serial Number Return

[I32]

Returns the serial number.

Attached?

[TF]

Returns the device status. (Attached is T; Not attached is F)

error out

Contains error information.

5.1.11 AcceSetTrigger

Description:

Set the change trigger for an axis.

Connector Pane:
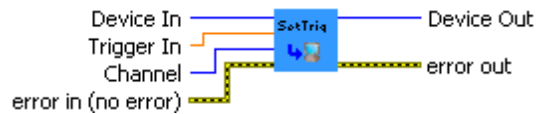
Controls and Indicators:

Input

Device In

I32   Device # identification.

Trigger In

DBL   The change trigger.

Channel

The acceleration index. (x, y, z)

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out
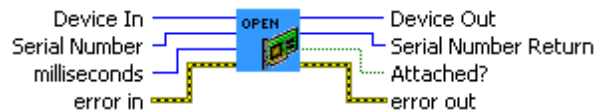
I32   Same as the Device In.

error out

Contains error information.

5.2   Phidget Advanced Servo

This contains VI functions for Phidget Advanced Servo. See the product manual for more specific API details, supported functionality, units, etc.

AdvServoCount

AdvServoCreate

AdvServoEventClose

AdvServoEventCreateCrtChange

AdvServoEventCreatePosChange

AdvServoEventCreateVelChange

AdvServoEventExe

AdvServoGetAcce

AdvServoGetAcceMax

AdvServoGetAcceMin

AdvServoGetCurrent

AdvServoGetEngaged

AdvServoGetPos

AdvServoGetPosMax

AdvServoGetPosMin

AdvServoGetRampingState

AdvServoGetServoType

AdvServoGetVel

AdvServoGetVellmt

AdvServoGetVelmax

AdvServoGetVelmin

AdvServoOpen

AdvServoSetAcce

AdvServoSetEngaged

AdvServoSetPos

AdvServoSetPosMax

AdvServoSetPosMin

AdvServoSetRampingState

AdvServoSetServoParameters

AdvServoSetServoType

AdvServoSetVellmt

AdvServoStoppedState

5.2.1   **AdvServoCount**

Description:

Gets the number of motors supported by this controller.

Connector Pane:
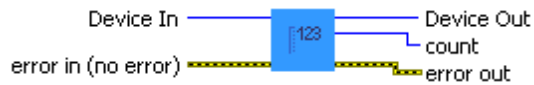
Controls and Indicators:

Input

Device In
I32
Device # identification.

error in (no error)
Describes error conditions that occur before this node runs.

Output

Device Out
I32
Same as the Device In.

count
I32
The motor count.

error out
Contains error information.

5.2.2   AdvServoCreate

Description:

Create a Phidget Advanced Servo handle.

Connector Pane:

Controls and Indicators:

Input

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Device # identification.

error out

Contains error information.

5.2.3   AdvServoEventClose

Description:

Close the Phidget Advanced Servo event handle.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

Event Registration Refnum In

Event # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

error out

Contains error information.

5.2.4   AdvServoEventCreateCrtChange

Description:

Set up a current change event handle.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output
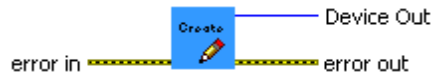
Device Out

Same as the Device In.

Event Registration Refnum Out

Event # identification.

error out

Contains error information.

5.2.5   AdvServoEventCreatePosChange

Description:

Set up a postion change event handle.

Connector Pane:

Controls and Indicators:

Input

Device In

[I32]   Device # identification.

error in (no error)

[error]   Describes error conditions that occur before this node runs.

Output

Device Out

[I32]   Same as the Device In.

Event Registration Refnum Out

[▶□]   Event # identification.

error out

[error]   Contains error information.

5.2.6   AdvServoEventCreateVelChange

Description:

Set up a velocity change event handle.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

error in (no error)

Describes error conditions that occur before this node runs.
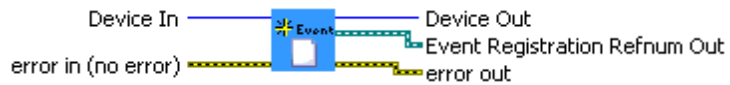
Output

Device Out

Same as the Device In.

Event Registration Refnum Out
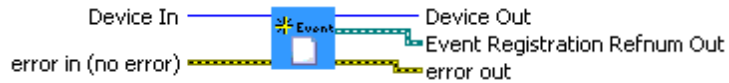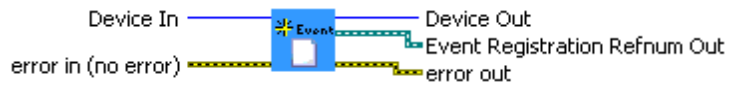
Event # identification.

error out

Contains error information.

5.2.7   AdvServoEventExe

Description:

This is called when the Phidget Advanced Servo event changes.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

Event Registration Refnum In

Event # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

Event Registration Refnum Out

Same as the Event Registration Refnum In.

index

The motor index.

Value
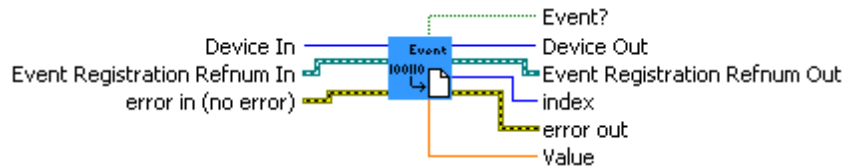
The return value of related event. (E.g.: For a position change event, this value is position.)

Event?

Returns the event status. (Executed is T; Not executed is F)

error out

Contains error information.

5.2.8   AdvServoGetAcce

Description:

Get the last set acceleration of a motor.

Connector Pane:

Controls and Indicators:

Input

Device In

[I32] Device # identification.

index

[I32] The motor index.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

[I32] Same as the Device In.

acce_out

[DBL] The acceleration.

error out

Contains error information.

5.2.9   AdvServoGetAcceMax

Description:

Get the maximum acceleration supported by a motor.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

index

The motor index.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

accemax_out

The maximum acceleration.

error out

Contains error information.

5.2.10 AdvServoGetAcceMin

Description:

Get the minimum acceleration supported by a motor.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

index

The motor index.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

accemin_out
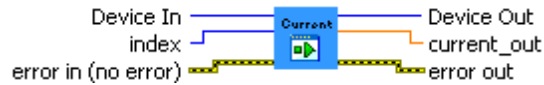
The minimum acceleration.

error out

Contains error information.

5.2.11 AdvServoGetCurrent

Description:

Get the current current draw for a motor.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

index

The motor index.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.
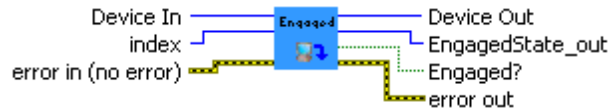
current_out

The current.

error out

Contains error information.

5.2.12 AdvServoGetEngaged

Description:

Get the engaged state of a motor. This is whether the motor is powered or not.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

index

The motor index.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

EngagedState_out

The engaged state. Possible values are 0 for False, 1 for True and others for undefined.

Engaged?

The engaged state. Possible values are True for Engaged and False for Not Engaged.

error out

Contains error information.

5.2.13 AdvServoGetPos

Description:

Get the current position of a motor.

Connector Pane:

Controls and Indicators:

Input

Device In

[I32]   Device # identification.

index

[I32]   The motor index.

error in (no error)

[error]   Describes error conditions that occur before this node runs.

Output

Device Out

[I32]   Same as the Device In.

pos_out

[DBL]   The position.

error out

[error]   Contains error information.

5.2.14 AdvServoGetPosMax

Description:

Get the maximum position that a motor can go to.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

index

The motor index.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

posmax_out
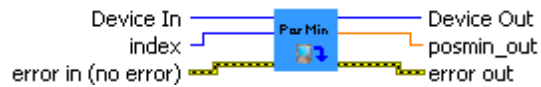
The maximum current.

error out

Contains error information.

5.2.15 AdvServoGetPosMin

Description:

Get the minimum position that a motor can go to.

Connector Pane:

Controls and Indicators:

Input

Device In
I32   Device # identification.

index
I32   The motor index.

error in (no error)   Describes error conditions that occur before this node runs.

Output

Device Out
I32   Same as the Device In.

posmin_out
DBL   The minimum position.

error out   Contains error information.

5.2.16 AdvServoGetRampingState

Description:

Get the speed ramping state for a motor. This is whether or not velocity and acceleration are used.

Connector Pane:
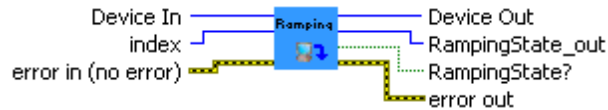
Controls and Indicators:

Input

Device In

Device # identification.

index

The motor index.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

RampingState_out The speed ramping state. Possible values are 0 for False, 1 for True and others for undefined.

RampingState?

The speed ramping state (Boolean type).

error out

Contains error information.

5.2.17 AdvServoGetServoType

Description:

Get the servo type of a motor.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

index

The motor index.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

GetServoType

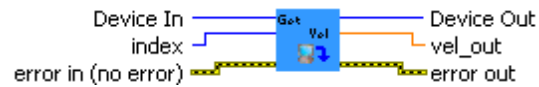Returns the servo type. This is an enum. Please refer to **Phigets Constant** -> **ServoType**

error out

Contains error information.

5.2.18 AdvServoGetVel

Description:

Get the current velocity of a motor.

Connector Pane:

Controls and Indicators:

Input

Device In

[I32]   Device # identification.

index

[I32]   The motor index.

error in (no error)

[error]   Describes error conditions that occur before this node runs.

Output

Device Out

[I32]   Same as the Device In.

vel_out

[DBL]   The current velocity.

error out

[error]   Contains error information.

5.2.19 AdvServoGetVellmt

Description:

Get the last set velocity limit of a motor.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

index

The motor index.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.
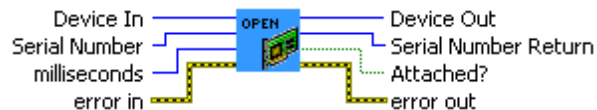
vellim_out

The velocity limit.

error out

Contains error information.

5.2.20 AdvServoGetVelmax

Description:

Get the maximum velocity that can be set for a motor.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

index

The motor index.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

velmax_out

The maximum velocity.

error out

Contains error information.

5.2.21 AdvServoGetVelmin

Description:

Get the minimum velocity that can be set for a motor.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

index

The motor index.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

velmin_out

The minimum velocity.

error out

Contains error information.

5.2.22 AdvServoOpen

Description:

Open a PhidgetAdvancedServo.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification. This function will create a new device identification if it's 0 or invalid.

Serial Number

Serial number. Specify -1 to open any.

milliseconds

Time to wait for the attachment. Specify 0 to wait forever. (Default is 5000)

error in (no error)

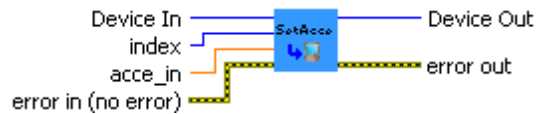Describes error conditions that occur before this node runs.
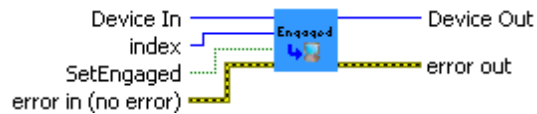
Output

Device Out

Same as the Device In.

Serial Number Return

Returns the serial number.

Attached?

Returns the device status. (Attached is T; Not attached is F)

error out

Contains error information.

5.2.23 AdvServoSetAcce

Description:

Set the acceleration for a motor.

Connector Pane:

Controls and Indicators:

Input

Device In

[I32]   Device # identification.

index

[I32]   The motor index.

acce_in

[DBL]   The acceleration.

error in (no error)

[===]   Describes error conditions that occur before this node runs.

Output

Device Out

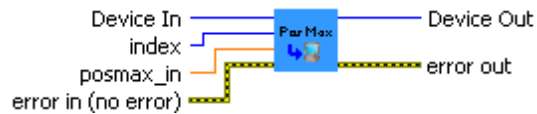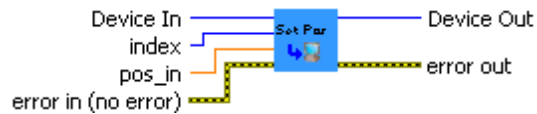[I32]   Same as the Device In.

error out

[===]   Contains error information.

5.2.24 AdvServoSetEngaged

Description:

Set the engaged state of a motor. This is whether the motor is powered or not.

Connector Pane:

Controls and Indicators:

Input

Device In
$I32$ Device # identification.

index
$I32$ The motor index.

SetEngaged
TF   Set the engage state.

error in (no error)
              Describes error conditions that occur before this node runs.

Output

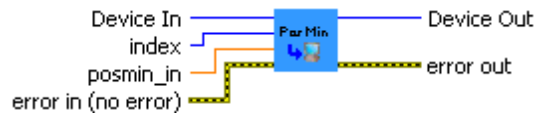Device Out
$I32$ Same as the Device In.

error out
              Contains error information.

5.2.25 AdvServoSetPos

Description:

Set the position of a motor.

Connector Pane:

Controls and Indicators:

Input

Device In

I32   Device # identification.

index

I32   The motor index.

pos_in

DBL   The position.

error in (no error)

            Describes error conditions that occur before this node runs.

Output

Device Out
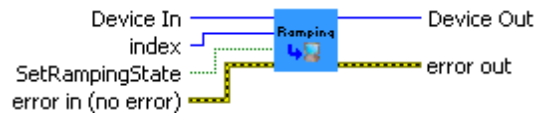
I32   Same as the Device In.

error out

            Contains error information.

5.2.26 AdvServoSetPosMax

Description:

Set the maximum position that a motor can go to.

Connector Pane:

Controls and Indicators:

Input

Device In
| I32 |　　　　Device # identification.

index
| I32 |　　　　The motor index.

posmax_in
| DBL |　　　　The maximum position.

error in (no error)
　　　　　　　　Describes error conditions that occur before this node runs.

Output
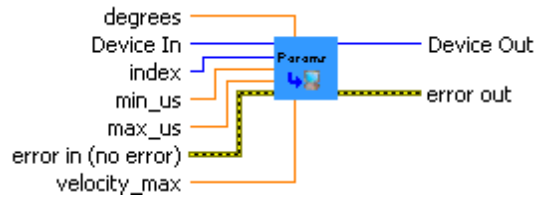
Device Out
| I32 |　　　　Same as the Device In.

error out
　　　　　　　　Contains error information.

5.2.27 AdvServoSetPosMin

Description:

Set the minimum position that a motor can go to.

Connector Pane:

Controls and Indicators:

Input

Device In

I32   Device # identification.

index

I32   The motor index.

posmin_in

DBL   The minimum position.

error in (no error)

Describes error conditions that occur before this node runs.

**Output**

Device Out

I32   Same as the Device In.

error out

Contains error information.

5.2.28 AdvServoSetRampingState

Description:

Set the speed ramping state for a motor. This is whether or not velocity and acceleration are used.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

index

The motor index.

SetRampingState

The speed ramping state. (0 = False 1 = True)

error in (no error)

Describes error conditions that occur before this node runs.
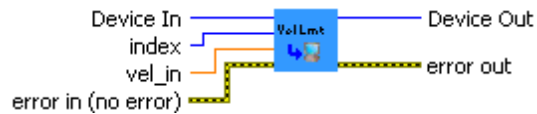
Output

Device Out

Same as the Device In.

error out

Contains error information.

5.2.29 AdvServoSetServoParameters

Description:

Set the servo parameters of a motor.

Connector Pane:

Controls and Indicators:

Input

Device In
[I32]   Device # identification.

index
[I32]   The motor index.

min_us
[DBL]   The minimum supported PCM in microseconds.

max_us
[DBL]   The maximum supported PCM in microseconds.

degrees
[DBL]   The degrees of rotation defined by the given PCM range.

velocity_max
[DBL]   The maximum velocity in degrees/second.

error in (no error)
[   ]   Describes error conditions that occur before this node runs.

Output

Device Out
[I32]   Same as the Device In.

error out
[   ]   Contains error information.

5.2.30 AdvServoSetServoType
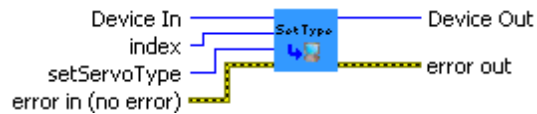
Description:

Set the servo type of a motor.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

index

The motor index.

setServoType

The servo type. This is an enum. Please refer to **Phigets Constant** -> **ServoType**

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

error out

Contains error information.

5.2.31 AdvServoSetVelImt

Description:

Set the velocity limit for a motor.

Connector Pane:

Controls and Indicators:

Input

Device In

I32   Device # identification.

index

I32   The motor index.

vel_in

DBL   The velocity limit.

error in (no error)

      Describes error conditions that occur before this node runs.

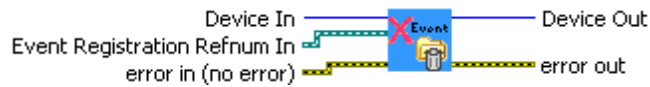**Output**

Device Out

I32   Same as the Device In.

error out

      Contains error information.

5.2.32 AdvServoStoppedState

Description:

Get the stopped state of a motor. This is true when the motor is not moving and there are no outstanding commands.

Connector Pane:

Controls and Indicators:

Input

Device In

I32   Device # identification.

index

I32   The motor index.

error in (no error)

             Describes error conditions that occur before this node runs.

Output

Device Out

I32   Same as the Device In.

StoppedState

I32   The stopped state. Possible values are 0 for False, 1 for True and others for undefined.

Stopped?

TF   The stopped state (Boolean type). Possible values are True for Stopped and False for Not Stopped.

error out

             Contains error information.

5.3   Phidget Encoder

This contains VI functions for Phidget Encoder. See the product manual for more specific API details, supported functionality, units, etc.

EncoderCreate

EncoderEventCloseInput

EncoderEventClosePosition

EncoderEventCreateInput

EncoderEventCreatePosition

EncoderEventExeInput

EncoderEventExePosition

EncoderGetCount

EncoderGetEnabledState

EncoderGetIndexPosition

EncoderGetInputCount

EncoderGetInputState

EncoderGetPosition

EncoderOpen

EncoderSetEnabled

EncoderSetPosition

5.3.1   EncoderCreate

Description:

Create a Phidget Encoder handle.

Connector Pane:

Controls and Indicators:

Input

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Device # identification.

error out

Contains error information.

5.3.2 EncoderEventCloseInput

Description:

Close the input change event handle.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

Event Registration Refnum In

Event # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

error out

Contains error information.

5.3.3   EncoderEventClosePosition

Description:

Close the encoder position change event handle.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

Event Registration Refnum In

Event # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output

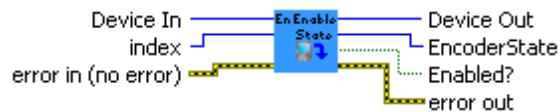Device Out

Same as the Device In.

error out

Contains error information.

5.3.4 EncoderEventCreateInput

Description:

Set up an input change event handle.

Connector Pane:

Device In ———————— Device Out
                                    Event Registration Refnum Out
error in (no error) ════════════ error out
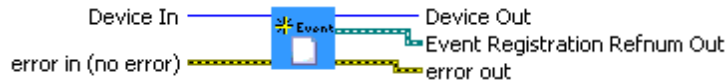
Controls and Indicators:

Input

Device In
[I32]   Device # identification.

error in (no error)
[ ]   Describes error conditions that occur before this node runs.

Output

Device Out
[I32]   Same as the Device In.

Event Registration Refnum Out
[ ]   Event # identification.

error out
[ ]   Contains error information.

5.3.5   EncoderEventCreatePosition

Description:

Set up an encoder position change event handle.
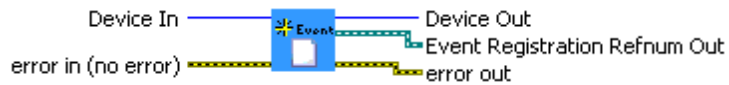
Connector Pane:

Device In ──────── Device Out
                   Event Registration Refnum Out
error in (no error) ══════ error out

Controls and Indicators:

Input

Device In
[I32]   Device # identification.

error in (no error)
[Error]   Describes error conditions that occur before this node runs.

Output

Device Out
[I32]   Same as the Device In.

Event Registration Refnum Out
[ ]   Event # identification.

error out
[Error]   Contains error information.

5.3.6   EncoderEventExeInput

Description:

This is called when the acceleration changes by more then the change trigger.

Connector Pane:

Controls and Indicators:

Input

Device In

I32   Device # identification.

Event Registration Refnum In

                            Event # identification.

error in (no error)

                            Describes error conditions that occur before this node runs.

Output

Device Out

I32   Same as the Device In.
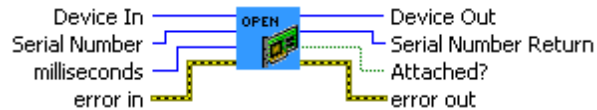
Event Registration Refnum Out

                            Same as the Event Registration Refnum In.

Event?

TF   Returns the event status. (Executed is T; Not executed is F)

index

I32   The input index.

Value

I32   The input state value. Possible values are 0 for False, 1 for True and
                            others for undefined.

error out

                            Contains error information.

5.3.7   EncoderEventExePosition

Description:

This is called when an encoder position changes.

Connector Pane:
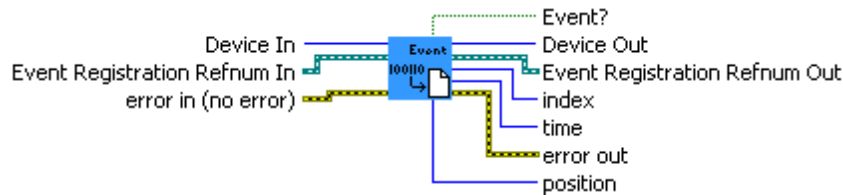
Controls and Indicators:

Input

Device In

Device # identification.

Event Registration Refnum In

Event # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

Event Registration Refnum Out

Same as the Event Registration Refnum In.

Event?

Returns the event status. (Executed is T; Not executed is F)

index

The encoder index.

time

The time in ms since the last position change event.

position

The current position of the encoder. (This is a relative not absolute position.)

error out

Contains error information.

5.3.8   EncoderGetCount

Description:

Get the number of encoders supported by this board.

Connector Pane:

Device In ──────── Device Out
                   └ EncoderCount
error in (no error) ═══════ error out

Controls and Indicators:

Input

Device In

Device # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

EncoderCount

The encoder input count.

error out

Contains error information.

5.3.9   EncoderGetEnabledState

Description:

Get the enabled state of an encoder. This is whether the encoder is powered or not. Please note that 1057 doesn't support this function.

Connector Pane:

Controls and Indicators:

Input

Device In
[I32]   Device # identification.

index
[I32]   The encoder index.

error in (no error)
[...]   Describes error conditions that occur before this node runs.

Output

Device Out
[I32]   Same as the Device In.

EncoderState
[I32]   The enabled state. Possible values are 0 for False, 1 for True and others for undefined.

Enabled?
[TF]   The enabled state (Boolean type). Possible values are True for Enabled and False for Not Enabled (Disabled).

error out
[...]   Contains error information.

5.3.10 EncoderGetIndexPosition

Description:

Get the position of the last index pulse, as referenced to **Phidget Encoder** -> **EncoderGetPosition**. The function will return an error (EPHIDGET_UNKNOWN) if there hasn't been an index event, or if the encoder doesn't support index.

Connector Pane:

Controls and Indicators:

Input

Device In
Device # identification.

index
The encoder index.

error in (no error)
Describes error conditions that occur before this node runs.

Output

Device Out
Same as the Device In.

IndexPosition
The index position.

error out
Contains error information.

5.3.11 EncoderGetInputCount

Description:

Get the number of digital inputs supported by this board.

Connector Pane:

Controls and Indicators:

Input

Device In

[I32]   Device # identification.

error in (no error)

[error]   Describes error conditions that occur before this node runs.
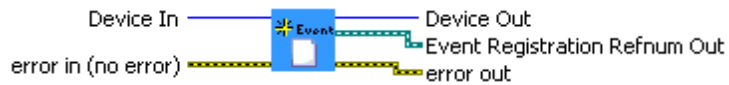
Output

Device Out

[I32]   Same as the Device In.

InputCount

[I32]   The input count.

error out

[error]   Contains error information.

5.3.12 EncoderGetInputState

Description:
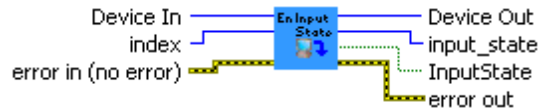
Get the state of a digital input.

Connector Pane:

Controls and Indicators:

Input

Device In

[I32]   Device # identification.

index

[I32]   The input index.

error in (no error)

[≡≡≡]   Describes error conditions that occur before this node runs.

Output

Device Out

[I32]   Same as the Device In.

input_state

[I32]   The input state. Possible values are 0 for False, 1 for True and others for undefined.

InputState

[TF]   The input state (Boolean type). Possible values are True for Engaged and False for Not Engaged.

error out

[≡≡≡]   Contains error information.

5.3.13 EncoderGetPosition

Description:

Get the current position of an encoder.
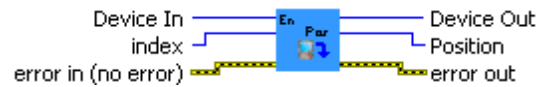
Connector Pane:

Controls and Indicators:

Input

Device In

I32   Device # identification.

index

I32   The encoder index.

error in (no error)

        Describes error conditions that occur before this node runs.

Output

Device Out

I32   Same as the Device In.

Position

I32   The current position.

error out

        Contains error information.

5.3.14 EncoderOpen

Description:

Open a PhidgetEncoder.

Connector Pane:

Controls and Indicators:

Input

Device In

`I32`

Device # identification. This function will create a new device identification if it's 0 or invalid.

Serial Number

`I32`

Serial number. Specify -1 to open any.

milliseconds

`I32`

Time to wait for the attachment. Specify 0 to wait forever. (Default is 5000)

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

`I32`

Same as the Device In.

Serial Number Return

`I32`

Returns the serial number.

Attached?

`TF`

Returns the device status. (Attached is T; Not attached is F)

error out

Contains error information.

5.3.15 EncoderSetEnabled

Description:

Set the enabled state of an encoder. This is whether the encoder is powered or not. Please note that 1057 doesn't support this function.

Connector Pane:
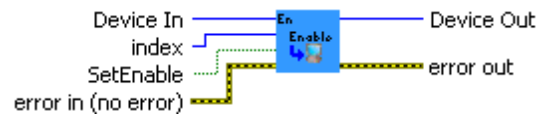
Controls and Indicators:

Input

Device In

Device # identification.

index

The encoder index.

SetEnable

The encoder state. (0 = False 1 = True)

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

error out

Contains error information.

5.3.16 EncoderSetPosition

Description:

Set the position of an encoder.
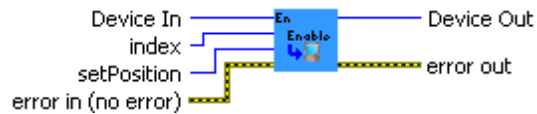
Connector Pane:

Controls and Indicators:

Input

Device In

I32   Device # identification.

index

I32   The encoder index.

setPosition

I32   The new position.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

I32   Same as the Device In.

error out

Contains error information.

5.4   Phidget InterfaceKit

This contains VI functions for Phidget InterfaceKit. See the product manual for more specific API details, supported functionality, units, etc.

IFCreate

IFEventClose

IFEventCreateInput

IFEventCreateOutput

IFEventCreateSensor

IFEventExe

IFGetDataRate

IFGetDataRateMax

IFGetDataRateMin

IFGetInputCount

IFGetInputState

IFGetOutputCount

IFGetOutputState

IFGetRatio

IFGetSensorCount

IFGetSensorValue

IFGetSensorValueRaw

IFGetTrig

IFOpen

IFSetDataRate

IFSetOutputState

IFSetRatio
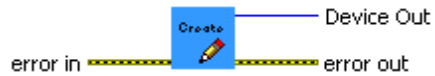
IFSetTrig

5.4.1   IFCreate

Description:

Create a Phidget InterfaceKit handle.

Connector Pane:

Controls and Indicators:

Input

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Device # identification.

error out
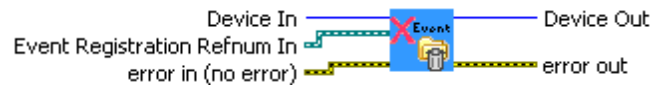
Contains error information.

5.4.2   IFEventClose

Description:

Close the Phidget InterfaceKit event handle.

Connector Pane:

Device In ──────────────── Device Out
Event Registration Refnum In ─┐ ┌─ Event
error in (no error) ──────────┘ └─ error out

Controls and Indicators:

Input

Device In
[I32]   Device # identification.

Event Registration Refnum In
[ ]   Event # identification.

error in (no error)
[ ]   Describes error conditions that occur before this node runs.

Output

Device Out
[I32]   Same as the Device In.

error out
[ ]   Contains error information.

5.4.3   IFEventCreateInput

Description:

Set up a digital input change event handle.

Connector Pane:

Controls and Indicators:

Input

Device In

[I32]   Device # identification.

error in (no error)

[===]   Describes error conditions that occur before this node runs.

Output

Device Out

[I32]   Same as the Device In.

Event Registration Refnum Out

[===]   Event # identification.

error out

[===]   Contains error information.

5.4.4   IFEventCreateOutput

Description:

Set up a digital output change event handle.
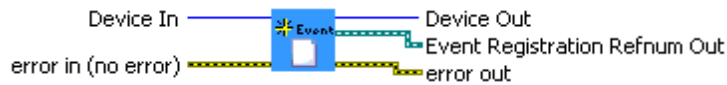
Connector Pane:

Device In ──────────── Device Out
                    ⊹ Event ⌐── Event Registration Refnum Out
error in (no error) ═══════════ error out

Controls and Indicators:

Input

Device In
[I32]   Device # identification.

error in (no error)
[error]   Describes error conditions that occur before this node runs.

Output

Device Out
[I32]   Same as the Device In.
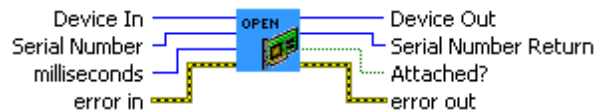
Event Registration Refnum Out
[▶□]   Event # identification.
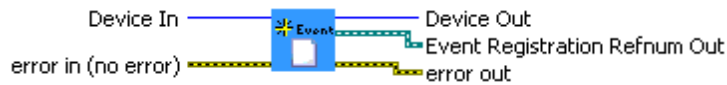
error out
[error]   Contains error information.

5.4.5   IFEventCreateSensor

Description:

Set up a sensor change event handle.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

Event Registration Refnum Out

Event # identification.

error out

Contains error information.

5.4.6   IFEventExe

Description:

This is called when the Phidget Advanced Servo event changes.

Connector Pane:
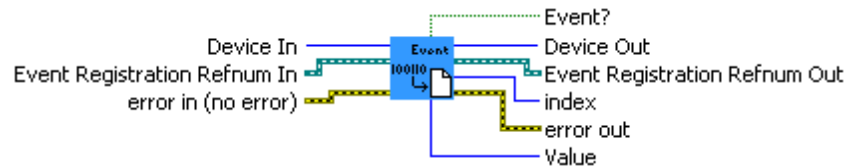
Controls and Indicators:

Input

Device In
I32   Device # identification.

Event Registration Refnum In
                           Event # identification.

error in (no error)
                           Describes error conditions that occur before this node runs.

Output

Device Out
I32   Same as the Device In.

Event Registration Refnum Out
                           Same as the Event Registration Refnum In.

index
I32   The input, output or sensor index.

Value   The return value of related event. (E.g. for sensor change event, this value
I32   is sensor reading.)

Event?
TF   Returns the event status. (Executed is T; Not executed is F)

error out
                           Contains error information.

5.4.7   IFGetDataRate

Description:

Get the data rate for an analog input.

Connector Pane:

Controls and Indicators:

Input

Device In

[I32] Device # identification.

index

[I32] The sensor index.

error in (no error)

[   ] Describes error conditions that occur before this node runs.
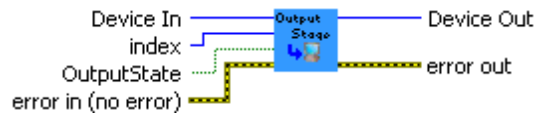
Output

Device Out

[I32] Same as the Device In.

DataRate

[I32] The data rate for an analog input in ms.

error out

[   ] Contains error information.

5.4.8   IFGetDataRateMax

Description:

Get the maximum supported data rate for an analog input.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

index

The sensor index.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

DataRateMax

The maximum data rate for an analog input in ms.

error out

Contains error information.

5.4.9   IFGetDataRateMin

Description:

Get the minimum supported data rate for an analog input.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

index

The sensor index.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

DataRateMin

The minimum data rate for an analog input in ms.

error out

Contains error information.

5.4.10 IFGetInputCount

Description:

Get the number of digital inputs supported by this board.

Connector Pane:

Device In ────────── Device Out
                      └ Input Count
error in (no error) ═══════════ error out
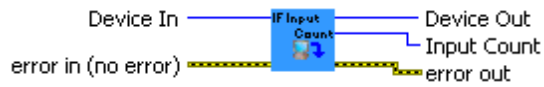
Controls and Indicators:

Input

Device In
[I32]   Device # identification.

error in (no error)
[   ]   Describes error conditions that occur before this node runs.

Output

Device Out
[I32]   Same as the Device In.

Input Count
[I32]   The ditial input count.

error out
[   ]   Contains error information.

5.4.11 IFGetInputState

Description:

Get the state of a digital input.

Connector Pane:
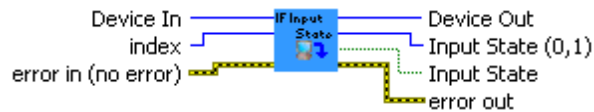
Controls and Indicators:

Input

Device In

Device # identification.

index

The input index.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

Input State (0,1)

The input state. Possible values are 0 for False, 1 for True and others for undefined.

Input State

The input state (Boolean type).

error out

Contains error information.

5.4.12 IFGetOutputCount

Description:

Get the number of digital outputs supported by this board.

Connector Pane:

Device In ─────────────── Device Out
                          └ Output count
error in (no error) ════════ error out

Controls and Indicators:

Input

Device In
I32   Device # identification.

error in (no error)
             Describes error conditions that occur before this node runs.

Output
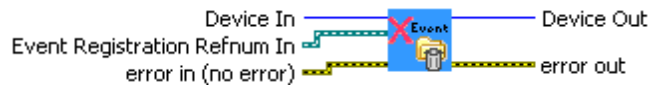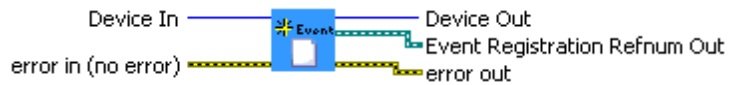
Device Out
I32   Same as the Device In.

Output count
I32   The ditial output count.

error out
             Contains error information.

5.4.13 IFGetOutputState

Description:

Get the state of a digital output.

Connector Pane:
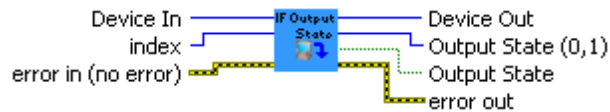
Controls and Indicators:

Input

Device In

Device # identification.

index

The output index.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

Output State (0,1)

The output state. Possible values are 0 for False, 1 for True and others for undefined.

Output State

The output state (Boolean type).

error out

Contains error information.

5.4.14 IFGetRatio

Description:

Get the ratio metric state for this board.
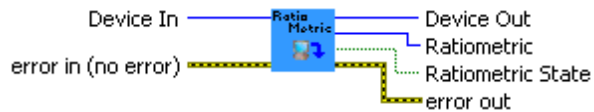
Connector Pane:

Controls and Indicators:

Input

Device In

[I32]   Device # identification.

error in (no error)

[error]   Describes error conditions that occur before this node runs.

Output

Device Out

[I32]   Same as the Device In.

Ratiometric   The ratio metric state. Possible values are 0 for False, 1 for True and others for

[I32]   undefined.

Ratiometric State

[TF]   The ratio metric state (Boolean type).

error out

[error]   Contains error information.

5.4.15 IFGetSensorCount

Description:

Get the number of sensor (analog) inputs supported by this board.

Connector Pane:

Device In ——————————— Device Out
                                    Sensor Count
error in (no error) ———————— error out

Controls and Indicators:

Input

Device In

[I32]   Device # identification.

error in (no error)

[   ]   Describes error conditions that occur before this node runs.

Output

Device Out

[I32]   Same as the Device In.

Sensor Count

[I32]   The sensor input count.

error out

[   ]   Contains error information.

5.4.16 IFGetSensorValue

Description:

Get a sensor value (0-1000).

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

index

The sensor index.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

Sensor Value

The sensor value.

error out

Contains error information.

5.4.17 IFGetSensorValueRaw

Description:

Get a sensor raw value (12-bit).

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

index

The sensor index.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

Sensor Value (Raw)

The sensor value.

error out

Contains error information.

5.4.18 IFGetTrig

Description:

Get a sensor change trigger.
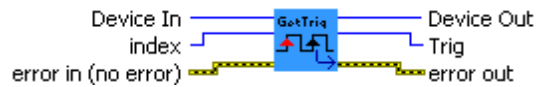
Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

index

The sensor index.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

Trig

The change trigger.

error out

Contains error information.

5.4.19 IFOpen

Description:

Open a PhidgetInterfaceKit.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification. This function will create a new device identification if it's 0 or invalid.

Serial Number

Serial number. Specify -1 to open any.

milliseconds

Time to wait for the attachment. Specify 0 to wait forever. (Default is 5000)

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

Serial Number Return

Returns the serial number.

Attached?

Returns the device status. (Attached is T; Not attached is F)

error out

Contains error information.

5.4.20 IFSetDataRate

Description:

Set the data rate for an analog input.

Connector Pane:
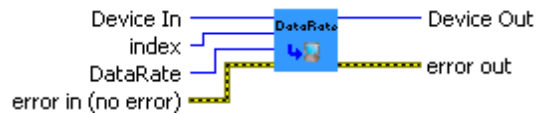
Controls and Indicators:

Input

Device In

Device # identification.

index

The sensor index.

DataRate

Data rate in ms.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

error out

Contains error information.

5.4.21 IFSetOutputState

Description:

Set the state of a digital output.

Connector Pane:

Controls and Indicators:

Input

Device In
I32   Device # identification.

index
I32   The output index.

OutputState
TF   The output state. (0 = False 1 = True)

error in (no error)
    Describes error conditions that occur before this node runs.

Output

Device Out
I32   Same as the Device In.

error out
    Contains error information.

5.4.22 IFSetRatio

Description:

Set the ratio metric state for this board.

Connector Pane:

Device In ─────────┌─────────┐───────── Device Out
Ratiometric ─ ─ ─ ─│ Ratio   │
                   │ Metric │
error in (no error) ═══════│   │═══════ error out
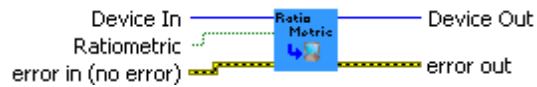
Controls and Indicators:

Input

Device In
[I32]   Device # identification.

Ratiometric
[TF]   The ratio metric state. (0 = False 1 = True)

error in (no error)
[...]   Describes error conditions that occur before this node runs.

Output

Device Out
[I32]   Same as the Device In.

error out
[...]   Contains error information.

5.4.23 IFSetTrig

Description:

Set a sensor change trigger.

Connector Pane:
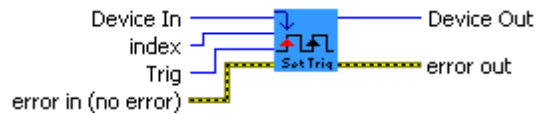
Controls and Indicators:

Input

Device In

[I32]   Device # identification.

index

[I32]   The sensor index.

Trig

[I32]   The change trigger.

error in (no error)

[error]   Describes error conditions that occur before this node runs.

Output

Device Out

[I32]   Same as the Device In.

error out

[error]   Contains error information.

5.5   Phidget IR

This contains VI functions for Phidget IR. See the product manual for more specific API details, supported functionality, units, etc.

IRCreate

IREventCloseOnCode

IREventCloseOnLearn

IREventCloseOnRawData

IREventCreateOnCode

IREventCreateOnLearn

IREventCreateOnRawData

IREventExeOnCode

IREventExeOnLearn

IREventExeOnRawData

IRGetLastCode

IRGetLastLearnedCode

IRGetRawData

IROpen

IRTransmit

IRTransmitRaw

IRTransmitRepeat

5.5.1   IRCreate

Description:

Create a Phidget IR handle.

Connector Pane:

Controls and Indicators:

Input

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Device # identification.

error out

Contains error information.

5.5.2   IREventCloseOnCode

Description:

Close the Phidget IR OnCode event handle.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

Event Registration Refnum In

Event # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

error out

Contains error information.

5.5.3   IREventCloseOnLearn

Description:

Close the Phidget IR OnLearn event handle.
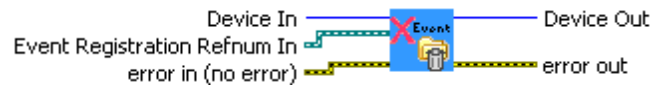
Connector Pane:

Device In ——————— Device Out
Event Registration Refnum In
error in (no error) ——————— error out

Controls and Indicators:

Input

Device In
[I32]   Device # identification.

Event Registration Refnum In
[□]   Event # identification.

error in (no error)
[⊞]   Describes error conditions that occur before this node runs.

Output

Device Out
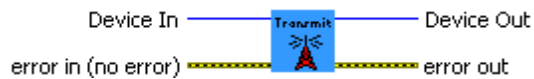[I32]   Same as the Device In.

error out
[⊞]   Contains error information.

5.5.4   IREventCloseOnRawData

Description:

Close the Phidget IR OnRawData event handle.

Connector Pane:

Device In ——————————— Device Out
Event Registration Refnum In
error in (no error) ———————— error out

Controls and Indicators:

Input

Device In

Device # identification.

Event Registration Refnum In

Event # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

error out

Contains error information.

5.5.5   IREventCreateOnCode

Description:

Set up a Phidget IR OnCode event handle.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

Event Registration Refnum Out

Event # identification.

error out

Contains error information.

5.5.6   IREventCreateOnLearn

Description:

Set up a Phidget IR OnLearn event handle.
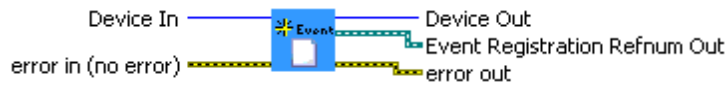
Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

Event Registration Refnum Out
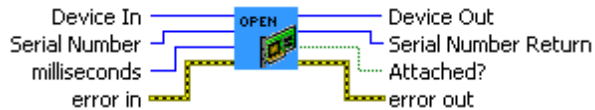
Event # identification.

error out

Contains error information.

5.5.7   IREventCreateOnRawData

Description:

Set up a Phidget IR OnRawData event handle.
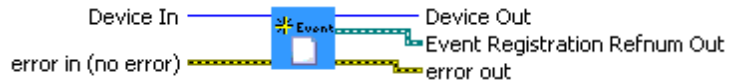
Connector Pane:

Device In ————————————— Device Out
                        Event Registration Refnum Out
error in (no error) ————————— error out

Controls and Indicators:

Input

Device In

[I32]   Device # identification.

error in (no error)

[   ]   Describes error conditions that occur before this node runs.

Output

Device Out

[I32]   Same as the Device In.

Event Registration Refnum Out

[ ]   Event # identification.

error out

[   ]   Contains error information.

5.5.8   IREventExeOnCode

Description:

This is called when a code has been received that could be automatically decoded. Data is return as an array with MSB in index 0. Bit count and a repeat flag are also returned. Repeats are detected as either the same code repeated in < 100ms or as a special repeat code.
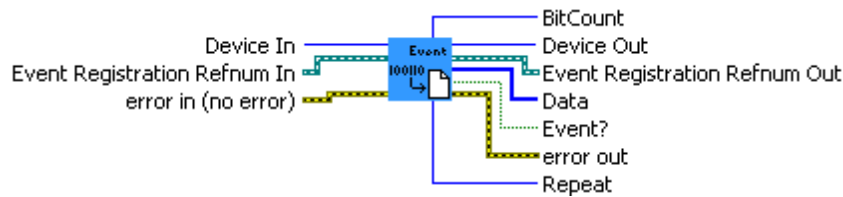
Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

Event Registration Refnum In

Event # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

Event Registration Refnum Out

Same as the Event Registration Refnum In.

Data

A user array to store the code data in.

BitCount

The bit count of the code.

Repeat

Returns the repeats.

Event?

Returns the event status. (Executed is T; Not executed is F)

error out

Contains error information.

5.5.9   IREventExeOnLearn

Description:

This is called when a code has been received for long enough to be learned. The returned CodeInfo structure can be used to retransmit the same code.

Connector Pane:
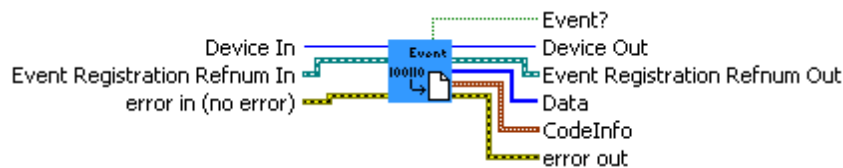
Controls and Indicators:

Input

Device In

[I32]   Device # identification.

Event Registration Refnum In

[☐]   Event # identification.

error in (no error)

[error]   Describes error conditions that occur before this node runs.
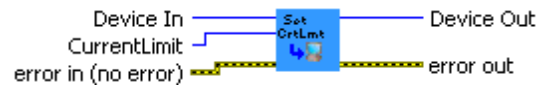
Output

Device Out

[I32]   Same as the Device In.

Event Registration Refnum Out

[☐]   Same as the Event Registration Refnum In.

Data

[U8]   A user array to store the code data in.

CodeInfo   This contains all information needed to transmit a code.

[905]   Please refer to **Phigets Constant** -> **CodeInfo**

Event?

[TF]   Returns the event status. (Executed is T; Not executed is F)

error out

[error]   Contains error information.

5.5.10 IREventExeOnRawData

Description:

This is called when raw data has been read from the device. Raw data always starts with a space and ends with a pulse.
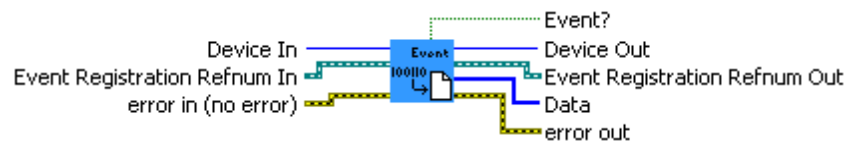
Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

Event Registration Refnum In

Event # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

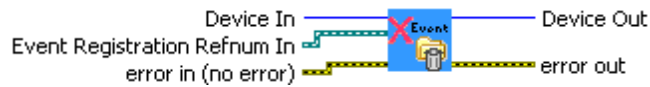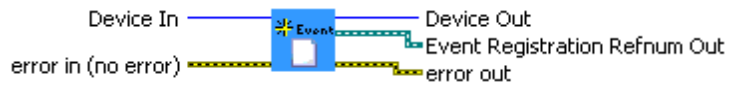Event Registration Refnum Out

Same as the Event Registration Refnum In.

Data

A user array for raw data to be written into.

Event?

Returns the event status. (Executed is T; Not executed is F)

error out

Contains error information.

5.5.11 IRGetLastCode

Description:

Get the last code that was received.

Connector Pane:

Controls and Indicators:

Input

Device In

[I32]   Device # identification.

DataLength In (16) Length of the user array - should be at least 16. This is set to the amount of data
[I32]   actually written to the array.

error in (no error)

[ERR]   Describes error conditions that occur before this node runs.

Output

Device Out

[I32]   Same as the Device In.

Data

[U8]   A user array to store the code data in.

BitCount

[I32]   Set to the bit count of the code.

dataLength

[I32]   Length of the user array.

error out

[ERR]   Contains error information.

5.5.12 IRGetLastLearnedCode

Description:

Get the last code that was learned.

Connector Pane:
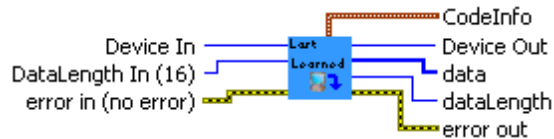
Controls and Indicators:

Input

Device In

Device # identification.

DataLength In (16)   Length of the user array - should be at least 16. This is set to the amount of data actually written to the array.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

Data

A user array to store the code data in.

CodeInfo   The CodeInfo structure for the learned code. Please refer to **Phigets Constant** -> **CodeInfo**

dataLength

Length of the user array.

error out

Contains error information.

5.5.13 IRGetRawData

Description:

Read any available raw data. This should be polled continuously (every 20ms) to avoid missing data. Read data always starts with a space and ends with a pulse.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

DataLength In

The maximum amount of data to read. This is set to the actual amount of data read.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

Data

A user array for raw data to be written into.

dataLength

The amount of data to read.

error out

Contains error information.

5.5.14 IROpen

Description:

Open a PhidgetIR.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification. This function will create a new device identification if it's 0 or invalid.

Serial Number

Serial number. Specify -1 to open any.

milliseconds

Time to wait for the attachment. Specify 0 to wait forever. (Default is 5000)

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

Serial Number Return

Returns the serial number.

Attached?

Returns the device status. (Attached is T; Not attached is F)

error out

Contains error information.

5.5.15 IRTransmit

Description:

Transmit a code according to the settings in a CodeInfo structure.

Connector Pane:
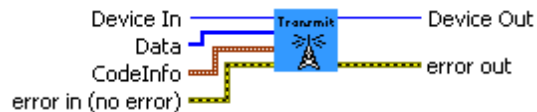
Controls and Indicators:

Input

Device In
I32

Device # identification.

Data
U8

The code to send. Data is transmitted MSBit first. MSByte is in array index 0. LSBit is right justified, so MSBit may be in bit positions 0-7 in array index 0 depending on the bit count.

CodeInfo

The CodeInfo structure specifying to to send the code. Anything left as null to select default is filled in for the user. Please refer to **Phigets Constant** -> **CodeInfo**

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out
I32

Same as the Device In.

error out

Contains error information.

5.5.16 IRTransmitRaw

Description:

Transmit RAW data as a series of pulses and spaces.

Connector Pane:
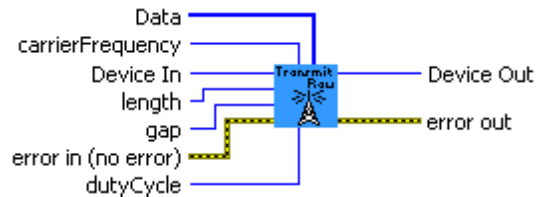
Controls and Indicators:

Input

Device In

[I32]   Device # identification.

Data

[U8]   The data to send. The array must start and end with a pulse and each element is a
               positive time in us.

length

[I32]   The length of the data array. Maximum length is 1024, but streams should be kept much
               shorter, ie. < 100ms between gaps.

carrierFrequency

[I32]   The Carrier Frequency in Hz. leave as 0 for default.

dutyCycle

[I32]   The Duty Cycle (10-50). Leave as 0 for default.

gap

[I32]   The gap time in us. This guarantees a gap time (no transmitting) after the data is sent,
               but can be set to 0.

error in (no error)

[   ]   Describes error conditions that occur before this node runs.

Output

Device Out

[I32]   Same as the Device In.

error out

[   ]   Contains error information.

5.5.17 IRTransmitRepeat

Description:

Transmits a repeat of the last transmitted code. Depending of the CodeInfo structure, this may be a retransmission of the code itself, or there may be a special repeat code.

Connector Pane:

Device In ———— Transmit ———— Device Out
error in (no error) ======   ====== error out

Controls and Indicators:

Input

Device In

[I32]   Device # identification.

error in (no error)

[   ]   Describes error conditions that occur before this node runs.

Output

Device Out

[I32]   Same as the Device In.

error out

[   ]   Contains error information.

5.6   Phidget LED

This contains VI functions for Phidget LED. See the product manual for more specific API details, supported functionality, units, etc.

LEDCount

LEDCreate

LEDGetBrightness

LEDGetCurrentLimit

LEDGetVoltage

LEDOpen

LEDSetBrightness

LEDSetCurrentLimit

LEDSetVoltage

5.6.1   LEDCount

Description:

Get the number of LEDs supported by this board.

Connector Pane:

Controls and Indicators:

Input

Device In

[I32]   Device # identification.

error in (no error)

[error]   Describes error conditions that occur before this node runs.

Output

Device Out

[I32]   Same as the Device In.

count

[I32]   The LED count.

error out

[error]   Contains error information.

5.6.2   LEDCreate

Description:

Create a Phidget LED handle.

Connector Pane:

Controls and Indicators:

Input

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Device # identification.

error out

Contains error information.

5.6.3   LEDGetBrightness

Description:

Get the brightness of an LED.

Connector Pane:

Controls and Indicators:

Input

Device In

[I32]   Device # identification.

index

[I32]   The LED index.

error in (no error)

[errorin]   Describes error conditions that occur before this node runs.

Output

Device Out

[I32]   Same as the Device In.

brightness

[I32]   The LED brightness (0-100).

error out

[errorout]   Contains error information.

5.6.4   LEDGetCurrentLimit

Description:

Get the current limit. This is for all outputs.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

error in (no error)

Describes error conditions that occur before this node runs.
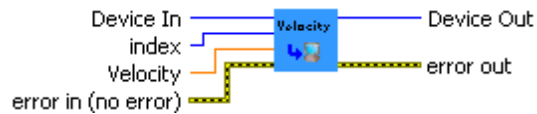
Output

Device Out

Same as the Device In.

CurrentLimit

The current limit.

error out

Contains error information.

5.6.5   LEDGetVoltage

Description:

Get the output voltage. This is for all outputs.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

Voltage

The output voltage.

error out

Contains error information.

5.6.6   LEDOpen

Description:

Open a Phidget LED.

Connector Pane:

Controls and Indicators:

Input

Device In
`I32`
Device # identification. This function will create a new device identification if it's 0 or invalid.

Serial Number
`I32`
Serial number. Specify -1 to open any.

milliseconds
`I32`
Time to wait for the attachment. Specify 0 to wait forever. (Default is 5000)

error in (no error)
Describes error conditions that occur before this node runs.
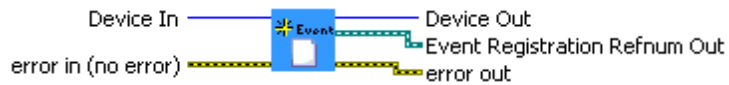
Output

Device Out
`I32`
Same as the Device In.

Serial Number Return
`I32`
Returns the serial number.

Attached?
`TF`
Returns the device status. (Attached is T; Not attached is F)

error out
Contains error information.

5.6.7   LEDSetBrightness

Description:

Set the brightness of an LED.

Connector Pane:
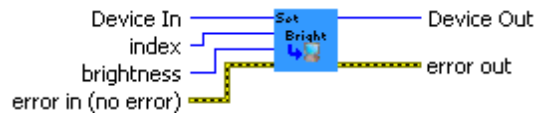
Controls and Indicators:

Input

Device In

$\boxed{\texttt{I32}}$ Device # identification.

index

$\boxed{\texttt{I32}}$ The LED index.

brightness

$\boxed{\texttt{I32}}$ The LED brightness (0-100).

error in (no error)

$\boxed{\texttt{...}}$ Describes error conditions that occur before this node runs.

Output

Device Out

$\boxed{\texttt{I32}}$ Same as the Device In.

error out

$\boxed{\texttt{...}}$ Contains error information.

5.6.8   LEDSetCurrentLimit

Description:

Set the current limit. This is for all outputs.

Connector Pane:

Device In ——————— Device Out
CurrentLimit ——| Set CrtLmt |
error in (no error) ——————— error out

Controls and Indicators:

Input

Device In
[I32]   Device # identification.

CurrentLimit
[I32]   The current limit.

error in (no error)
[ ]   Describes error conditions that occur before this node runs.

Output

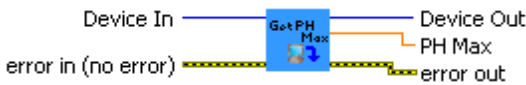Device Out
[I32]   Same as the Device In.

error out
[ ]   Contains error information.

5.6.9   LEDSetVoltage

Description:

Set the output voltage. This is for all outputs.

Connector Pane:

Controls and Indicators:

Input

Device In

[I32]   Device # identification.

Voltage

[I32]   The output voltage.

error in (no error)

[   ]   Describes error conditions that occur before this node runs.

Output

Device Out

[I32]   Same as the Device In.

error out

[   ]   Contains error information.

5.7 Phidget Motor Control

This contains VI functions for Phidget Motor Control. See the product manual for more specific API details, supported functionality, units, etc.

MCCreate

MCEventCloseInput

MCEventCloseVelocityCurrent

MCEventCreateCurrent

MCEventCreateInput

MCEventCreateVelocity

MCEventExeInput

MCEventExeVelocityCurrent

MCGetAcceleration

MCGetAccelerationMax

MCGetAccelerationMin

MCGetCurrent

MCGetInputCount

MCGetInputState

MCGetMotorCount

MCGetVelocity

MCOpen

MCSetAcceleration
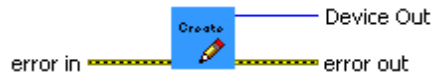
MCSetVelocity

5.7.1   MCCreate

Description:

Create a Phidget Motor Control handle.

Connector Pane:

Controls and Indicators:

Input

error in (no error)

Describes error conditions that occur before this node runs.

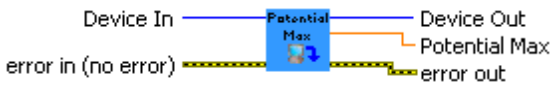**Output**

Device Out

Device # identification.

error out

Contains error information.

5.7.2   MCEventCloseInput

Description:

Close the Phidget Motor Control input change event handle.
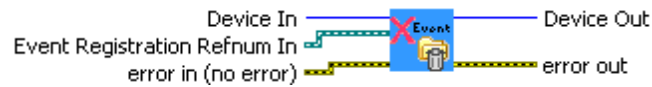
Connector Pane:

Controls and Indicators:

Input

Device In

[I32]   Device # identification.

Event Registration Refnum In

[□]   Event # identification.

error in (no error)

[⚡]   Describes error conditions that occur before this node runs.

Output

Device Out

[I32]   Same as the Device In.

error out

[⚡]   Contains error information.

5.7.3   MCEventCloseVelocityCurrent

Description:

Close the Phidget Motor Control velocity/current change event handle.

Connector Pane:

Controls and Indicators:

Input

Device In

I32   Device # identification.

Event Registration Refnum In

Event # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

I32   Same as the Device In.

error out

Contains error information.

5.7.4   MCEventCreateCurrent

Description:

Set up a current change event handle.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

Event Registration Refnum Out

Event # identification.

error out

Contains error information.

5.7.5   MCEventCreateInput
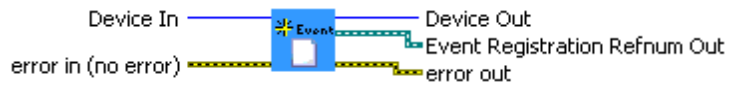
Description:

Set up an input change event handle.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

Event Registration Refnum Out

Event # identification.

error out

Contains error information.

5.7.6   MCEventCreateVelocity

Description:

Set up a velocity change event handle.

Connector Pane:

Controls and Indicators:

Input

Device In

I32   Device # identification.

error in (no error)

            Describes error conditions that occur before this node runs.

Output

Device Out

I32   Same as the Device In.

Event Registration Refnum Out

            Event # identification.

error out

            Contains error information.

5.7.7   MCEventExeInput

Description:

This is called when a digital input changes.

Connector Pane:
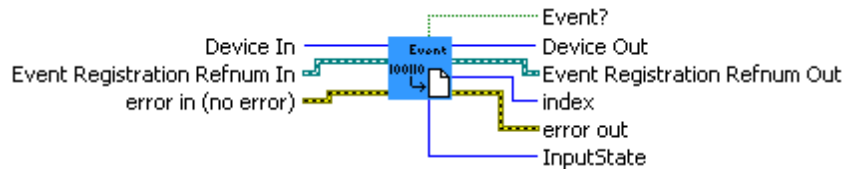
Controls and Indicators:

Input

Device In

[I32]   Device # identification.

Event Registration Refnum In

[D]   Event # identification.

error in (no error)

[═══]   Describes error conditions that occur before this node runs.

Output

Device Out

[I32]   Same as the Device In.

Event Registration Refnum Out

[D]   Same as the Event Registration Refnum In.

index

[I32]   The input index.

InputState

[I32]   The return value of input state.

Event?

[TF]   Returns the event status. (Executed is T; Not executed is F)

error out

[═══]   Contains error information.

5.7.8   MCEventExeVelocityCurrent

Description:

This is called when the velocity or current changes.

Connector Pane:
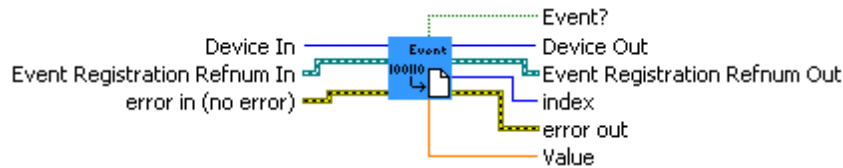
Controls and Indicators:

Input

Device In
[I32]   Device # identification.

Event Registration Refnum In
[□]   Event # identification.

error in (no error)
[⚏]   Describes error conditions that occur before this node runs.
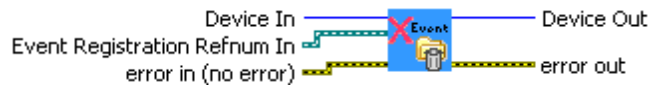
Output

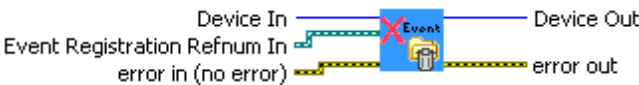Device Out
[I32]   Same as the Device In.

Event Registration Refnum Out
[□]   Same as the Event Registration Refnum In.

index
[I32]   The motor index.

Value   The return value of related event. (E.g. for velocity change event, this value
[DBL]   is velocity reading.)

Event?
[TF]   Returns the event status. (Executed is T; Not executed is F)

error out
[⚏]   Contains error information.

5.7.9   MCGetAcceleration
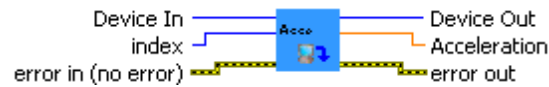
Description:

Get the last set acceleration of a motor.

Connector Pane:

Controls and Indicators:

Input

Device In

[I32]   Device # identification.

index

[I32]   The motor index.

error in (no error)

[error]   Describes error conditions that occur before this node runs.

Output

Device Out

[I32]   Same as the Device In.

Acceleration

[DBL]   The acceleration.

error out

[error]   Contains error information.

5.7.10 MCGetAccelerationMax

Description:

Get the maximum acceleration supported by a motor.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

index

The motor index.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

Acceleration Max

The maximum acceleration.

error out

Contains error information.

5.7.11 MCGetAccelerationMin

Description:

Get the minimum acceleration supported by a motor.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

index

The motor index.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

Acceleration Min

The minimum acceleration.

error out

Contains error information.

5.7.12 MCGetCurrent

Description:

Get the current current draw for a motor.

Connector Pane:

Controls and Indicators:

Input

Device In

[I32]   Device # identification.

index

[I32]   The motor index.

error in (no error)

[error]   Describes error conditions that occur before this node runs.

Output

Device Out

[I32]   Same as the Device In.

Current

[DBL]   The current.

error out

[error]   Contains error information.

5.7.13 MCGetInputCount

Description:

Get the number of digital inputs supported by this board.

Connector Pane:

Controls and Indicators:

Input

Device In

[I32] Device # identification.

error in (no error)

[error] Describes error conditions that occur before this node runs.

Output

Device Out

[I32] Same as the Device In.

count

[I32] The digital input count.

error out

[error] Contains error information.

5.7.14 MCGetInputState

Description:

Get the state of a digital input.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

index

The motor index.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

InputState

The input state. Possible values are 0 for False, 1 for True and others for undefined.

error out

Contains error information.

5.7.15 MCGetMotorCount

Description:

Get the number of motors supported by this controller.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

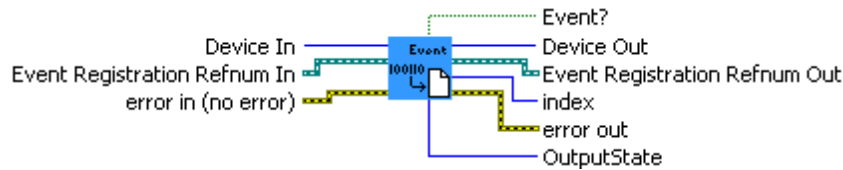Same as the Device In.

count

The motor count.

error out

Contains error information.

5.7.16 MCGetVelocity

Description:

Get the current velocity of a motor.

Connector Pane:

Device In ──────────── Device Out
index ────[Velocity]──── Velocity
error in (no error) ────── error out

Controls and Indicators:

Input

Device In
[I32]   Device # identification.

index
[I32]   The motor index.

error in (no error)
     Describes error conditions that occur before this node runs.

Output

Device Out
[I32]   Same as the Device In.

Velocity
[DBL]   The current velocity.
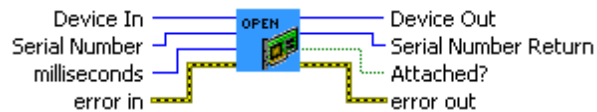
error out
     Contains error information.

5.7.17 MCOpen

Description:

Open a Phidget Motor Control.

Connector Pane:

Controls and Indicators:

Input

Device In

I32

Device # identification. This function will create a new device identification if it's 0 or invalid.

Serial Number

I32

Serial number. Specify -1 to open any.

milliseconds

I32

Time to wait for the attachment. Specify 0 to wait forever. (Default is 5000)

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

I32

Same as the Device In.

Serial Number Return

I32

Returns the serial number.

Attached?

TF

Returns the device status. (Attached is T; Not attached is F)

error out

Contains error information.

5.7.18 MCSetAcceleration

Description:

Set the last set acceleration of a motor.
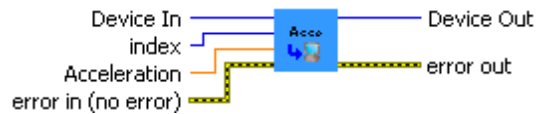
Connector Pane:

Controls and Indicators:

Input

Device In

[I32]   Device # identification.

index

[I32]   The motor index.

Acceleration

[DBL]   The acceleration.

error in (no error)

[error]   Describes error conditions that occur before this node runs.

Output

Device Out

[I32]   Same as the Device In.

error out

[error]   Contains error information.

5.7.19 MCSetVelocity

Description:

Set the velocity of a motor.

Connector Pane:

Controls and Indicators:

Input

Device In

I32   Device # identification.

index

I32   The motor index.

Velocity

DBL   The velocity.

error in (no error)

             Describes error conditions that occur before this node runs.

Output

Device Out

I32   Same as the Device In.

error out

             Contains error information.

5.8   Phidget PH Sensor

This contains VI functions for Phidget PH Sensor. See the product manual for more specific API details, supported functionality, units, etc.

PHCreate

PHEventClose

PHEventCreate

PHEventExe

PHGetPH

PHGetPHMax

PHGetPHMin

PHGetPHTrigger

PHGetPotential

PHGetPotentialMax

PHGetPotentialMin

PHOpen

PHSetTemperature

PHSetTrig

5.8.1   PHCreate

Description:

Create a Phidget PH Sensor handle.

Connector Pane:

Controls and Indicators:

Input

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Device # identification.

error out

Contains error information.

5.8.2   PHEventClose

Description:

Close the Phidget PH Sensor change event handle.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

Event Registration Refnum In

Event # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

error out

Contains error information.

5.8.3 PHEventCreate

Description:

Set up a PH change event handle.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

Event Registration Refnum Out

Event # identification.

error out

Contains error information.

5.8.4   PHEventExe

Description:

This is called when the PH changes by more then the change trigger.
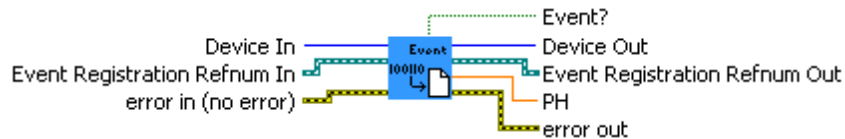
Connector Pane:

Controls and Indicators:

Input

Device In

I32   Device # identification.

Event Registration Refnum In

Event # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

I32   Same as the Device In.

Event Registration Refnum Out

Same as the Event Registration Refnum In.

PH

DBL   The PH.

Event?

TF   Returns the event status. (Executed is T; Not executed is F)

error out

Contains error information.

5.8.5 PHGetPH

Description:

Get the sensed PH.

Connector Pane:

Controls and Indicators:

Input

Device In

$\boxed{I32}$ Device # identification.

error in (no error)

        Describes error conditions that occur before this node runs.

Output

Device Out
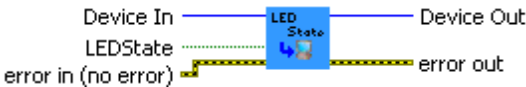
$\boxed{I32}$ Same as the Device In.

PH

$\boxed{DBL}$ The PH.

error out

        Contains error information.

5.8.6   PHGetPHMax

Description:

Get the maximum PH that the sensor could report.

Connector Pane:

Device In ─────────┌─────────┐───── Device Out
                   │ Get PH │──── PH Max
                   │ Max   │
error in (no error) ┈┈┈┈┈┈┈┈│   │┈┈┈┈ error out

Controls and Indicators:

Input

Device In
[I32]   Device # identification.

error in (no error)
[≡≡≡]   Describes error conditions that occur before this node runs.

Output

Device Out
[I32]   Same as the Device In.

PH Max
[DBL]   The maximum PH.

error out
[≡≡≡]   Contains error information.

5.8.7 PHGetPHMin

Description:

Get the minimum PH that the sensor could report.

Connector Pane:

Controls and Indicators:

Input

Device In

I32   Device # identification.

error in (no error)

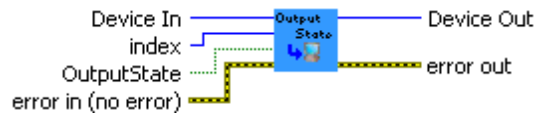   Describes error conditions that occur before this node runs.

Output

Device Out

I32   Same as the Device In.

PH Min

DBL   The minimum PH.

error out

    Contains error information.

5.8.8   PHGetPHTrigger

Description:

Get the PH change trigger.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

PH Trigger

The change trigger.

error out

Contains error information.

5.8.9   PHGetPotential

Description:

Get the sensed potential.

Connector Pane:
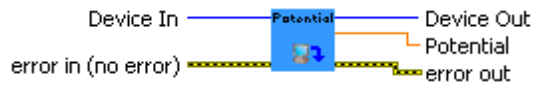
Controls and Indicators:

Input

Device In

[I32]   Device # identification.

error in (no error)

[ ]   Describes error conditions that occur before this node runs.

Output

Device Out

[I32]   Same as the Device In.

Potential

[DBL]   The potential.

error out

[ ]   Contains error information.

5.8.10 PHGetPotentialMax

Description:

Get the maximum potential that can be sensed.

Connector Pane:

Controls and Indicators:

Input

Device In

[I32]   Device # identification.

error in (no error)

[≡≡]   Describes error conditions that occur before this node runs.

Output

Device Out

[I32]   Same as the Device In.

Potential Max

[DBL]   The maximum potential.

error out

[≡≡]   Contains error information.

5.8.11 PHGetPotentialMin

Description:

Get the minimum potential that can be sensed.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

error in (no error)

Describes error conditions that occur before this node runs.
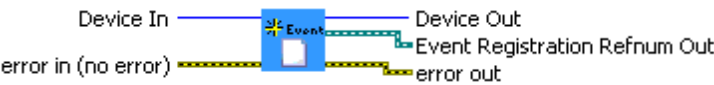
Output

Device Out

Same as the Device In.

Potential Min

The minimum potential.

error out

Contains error information.

5.8.12 PHOpen

Description:

Open a Phidget PH Sensor.

Connector Pane:
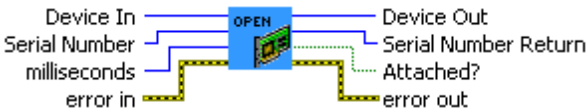
Controls and Indicators:

Input

Device In

Device # identification. This function will create a new device identification if it's 0 or invalid.

Serial Number

Serial number. Specify -1 to open any.

milliseconds

Time to wait for the attachment. Specify 0 to wait forever. (Default is 5000)

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

Serial Number Return

Returns the serial number.

Attached?

Returns the device status. (Attached is T; Not attached is F)

error out

Contains error information.

5.8.13 PHSetTemperature

Description:

Set the temperature to be used for PH calculations.

Connector Pane:

Device In ──────────────┐ Set Temp ┌────────── Device Out
Temperature ────────────┤   │
error in (no error) ────┴──────────┴────────── error out

Controls and Indicators:

Input

Device In
`I32`   Device # identification.

Temperature
`DBL`   The temperature (degrees celcius). By default this is 20.

error in (no error)
`error`   Describes error conditions that occur before this node runs.

Output

Device Out
`I32`   Same as the Device In.

error out
`error`   Contains error information.

5.8.14 PHSetTrig

Description:

Set the PH change trigger.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

Trigger

The change trigger.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

error out

Contains error information.

5.9   Phidget RFID

This contains VI functions for Phidget RFID. See the product manual for more specific API details, supported functionality, units, etc.

RFIDCreate

RFIDEventClose

RFIDEventCloseOutput

RFIDEventCreateOutput

RFIDEventCreateTag

RFIDEventCreateTagLost

RFIDEventExe

RFIDEventExeOutput

RFIDGetAntennaState

RFIDGetLastTag

RFIDGetLEDState

RFIDGetOutputCount

RFIDGetOutputState

RFIDGetTagState

RFIDOpen

RFIDSetAntennaState

RFIDSetLEDState

RFIDSetOutputState

5.9.1   RFIDCreate

Description:

Create a Phidget RFID handle.

Connector Pane:
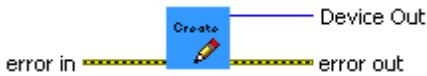
Controls and Indicators:

Input

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Device # identification.

error out

Contains error information.

5.9.2   RFIDEventClose

Description:

Close the Phidget RFID Tag or TagLost event handle.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

Event Registration Refnum In

Event # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

error out

Contains error information.

5.9.3   RFIDEventCloseOutput

Description:

Close the Phidget RFID output change event handle.

Connector Pane:

Controls and Indicators:

Input

Device In

[I32]   Device # identification.

Event Registration Refnum In

[▢]   Event # identification.

error in (no error)

[⚡]   Describes error conditions that occur before this node runs.
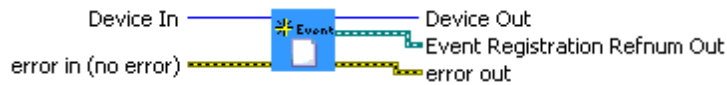
Output

Device Out

[I32]   Same as the Device In.

error out

[⚡]   Contains error information.

5.9.4   RFIDEventCreateOutput

Description:

Set up an output change event handle.

Connector Pane:

Controls and Indicators:

Input

Device In

[I32]   Device # identification.

error in (no error)

[----]   Describes error conditions that occur before this node runs.

Output

Device Out

[I32]   Same as the Device In.

Event Registration Refnum Out

[▸□]   Event # identification.

error out

[----]   Contains error information.

5.9.5   RFIDEventCreateTag

Description:

Set up a tag change event handle.
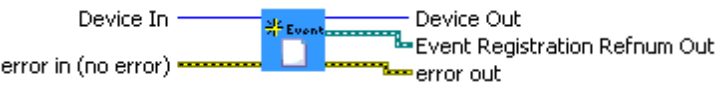
Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

Event Registration Refnum Out

Event # identification.

error out

Contains error information.

5.9.6   RFIDEventCreateTagLost

Description:

Set up a tag lost change event handle.

Connector Pane:

Device In ─────────┐┌─────── Device Out
                   │Event│─── Event Registration Refnum Out
error in (no error) └────┘─── error out

Controls and Indicators:

Input

Device In

`I32`   Device # identification.

error in (no error)

                               Describes error conditions that occur before this node runs.

Output

Device Out

`I32`   Same as the Device In.

Event Registration Refnum Out

                               Event # identification.

error out

                               Contains error information.

5.9.7   RFIDEventExe

Description:

This is called when the Phidget RFID Tag or TagLost event changes. Tag Event is called when a tag is first detected by the reader. TagLost Event is called when a tag is no longer detected by the reader.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

Event Registration Refnum In

Event # identification.

error in (no error)

Describes error conditions that occur before this node runs.
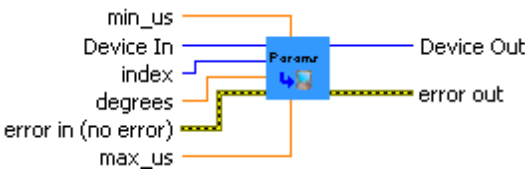
Output

Device Out

Same as the Device In.

Event Registration Refnum Out

Same as the Event Registration Refnum In.

Tag

The tag.

Event?

Returns the event status. (Executed is T; Not executed is F)

error out

Contains error information.

5.9.8   RFIDEventExeOutput

Description:

This is called when an output changes.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

Event Registration Refnum In

Event # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

Event Registration Refnum Out

Same as the Event Registration Refnum In.

index

The output index.

OutputState

The output state. Possible values are 0 for False, 1 for True and others for undefined.

Event?

Returns the event status. (Executed is T; Not executed is F)

error out

Contains error information.

5.9.9   RFIDGetAntennaState

Description:

Get the state of the antenna.

Connector Pane:

Controls and Indicators:

Input

Device In
I32   Device # identification.

error in (no error)
    Describes error conditions that occur before this node runs.

Output

Device Out
I32   Same as the Device In.

Antenna State
I32   The antenna state. Possible values are 0 for False, 1 for True and others for undefined.

Antenna On?
TF   The antenna state.

error out
    Contains error information.

5.9.10 RFIDGetLastTag

Description:

Get the last tag read by the reader. This tag may or may not still be on the reader.

Connector Pane:
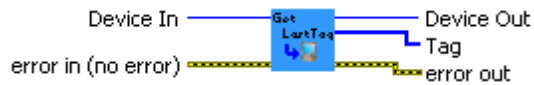
Controls and Indicators:

Input

Device In

Device # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

Tag

The tag. This must be an unsigned char array of size 5.

error out

Contains error information.

5.9.11 RFIDGetLEDState

Description:
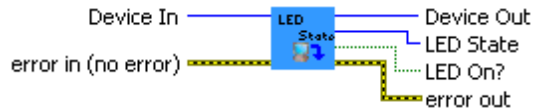
Get the state of the onboard LED.

Connector Pane:

Controls and Indicators:

Input

Device In

[I32]   Device # identification.

error in (no error)

[error]   Describes error conditions that occur before this node runs.

Output

Device Out

[I32]   Same as the Device In.

LED State

[I32]   The LED state. Possible values are 0 for False, 1 for True and others for undefined.

LED On?

[TF]   The LED state.

error out

[error]   Contains error information.

5.9.12 RFIDGetOutputCount

Description:

Get the number of outputs supported by this board.
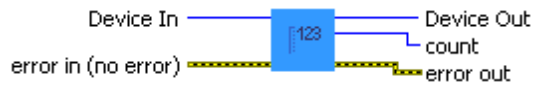
Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

count

The output count.

error out

Contains error information.

5.9.13 RFIDGetOutputState

Description:

Get the state of an output.
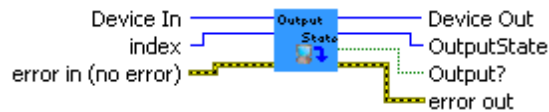
Connector Pane:

Controls and Indicators:

Input

Device In

[I32]   Device # identification.

index

[I32]   The output index.

error in (no error)

        Describes error conditions that occur before this node runs.

Output

Device Out

[I32]   Same as the Device In.

OutputState

[I32]   The output state. Possible values are 0 for False, 1 for True and others for undefined.

Output?

[TF]   The output state.

error out

        Contains error information.

5.9.14 RFIDGetTagState

Description:

Get the tag present status. This is whether or not a tag is being read by the reader.

Connector Pane:
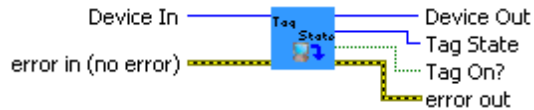
Controls and Indicators:

Input

Device In

Device # identification.

error in (no error)

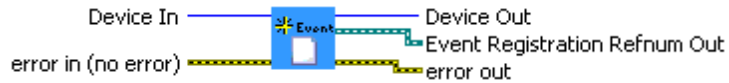Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

Tag State

The tag state. Possible values are 0 for False, 1 for True and others for undefined.

Tag On?

The tag state.

error out
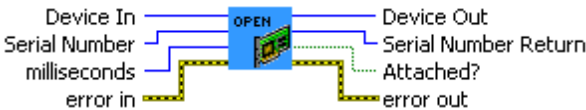
Contains error information.

5.9.15 RFIDOpen

Description:

Open a Phidget RFID.

Connector Pane:

Controls and Indicators:

Input

Device In
[I32]
Device # identification. This function will create a new device identification if it's 0 or invalid.

Serial Number
[I32]
Serial number. Specify -1 to open any.

milliseconds
[I32]
Time to wait for the attachment. Specify 0 to wait forever. (Default is 5000)

error in (no error)
Describes error conditions that occur before this node runs.

Output

Device Out
[I32]
Same as the Device In.

Serial Number Return
[I32]
Returns the serial number.

Attached?
[TF]
Returns the device status. (Attached is T; Not attached is F)

error out
Contains error information.

5.9.16 RFIDSetAntennaState

Description:

Set the state of the antenna. Note that the antenna must be enabled before tags will be read.

Connector Pane:
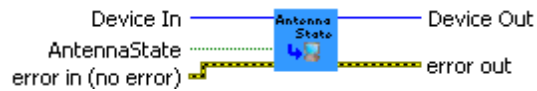
Controls and Indicators:

Input

Device In

Device # identification.

AntennaState

Set the antenna state.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

error out

Contains error information.

5.9.17 RFIDSetLEDState

Description:

Set the state of the onboard LED.

Connector Pane:

Controls and Indicators:

Input

Device In

[I32]   Device # identification.

LEDState

[TF]   Set the LED state.

error in (no error)

[error]   Describes error conditions that occur before this node runs.

Output

Device Out

[I32]   Same as the Device In.

error out

[error]   Contains error information.

5.9.18 RFIDSetOutputState

Description:

Set the state of an output.

Connector Pane:

Controls and Indicators:

Input

Device In
|I32| Device # identification.

index
|I32| The output index.

OutputState
|TF| Set the output state.

error in (no error)
| | Describes error conditions that occur before this node runs.

Output

Device Out
|I32| Same as the Device In.

error out
| | Contains error information.

5.10   Phidget Servo

This contains VI functions for Phidget Servo. See the product manual for more specific API details, supported functionality, units, etc.

ServoCount

ServoCreate

ServoEventClose

ServoEventCreate

ServoEventExe

ServoGetEngaged

ServoGetPos

ServoGetPosMax

ServoGetPosMin

ServoGetServoType

ServoOpen

ServoSetEngaged

ServoSetPos

ServoSetServoParameters

ServoSetServoType

5.10.1 ServoCount

Description:

Gets the number of motors supported by this controller.

Connector Pane:

Controls and Indicators:

Input

Device In

[I32]   Device # identification.

error in (no error)

[error]   Describes error conditions that occur before this node runs.

Output

Device Out

[I32]   Same as the Device In.

count

[I32]   The motor count.

error out

[error]   Contains error information.

5.10.2 ServoCreate

Description:

Create a Phidget Servo handle.

Connector Pane:
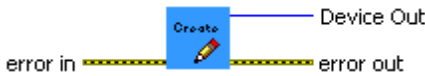
Controls and Indicators:

Input

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Device # identification.

error out

Contains error information.

5.10.3 ServoEventClose

Description:

Close the Phidget Servo event handle.
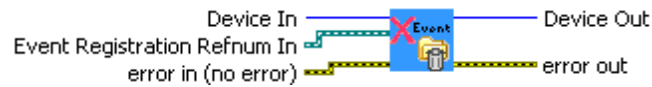
Connector Pane:

Controls and Indicators:

Input

Device In

I32   Device # identification.

Event Registration Refnum In

Event # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

I32   Same as the Device In.

error out

Contains error information.

5.10.4 ServoEventCreate

Description:

Set up a postion change event handle.

Connector Pane:

Device In ─────────────────┐ Event ┌──── Device Out
                           │   ├── Event Registration Refnum Out
error in (no error) ═══════┘   └── error out

Controls and Indicators:

Input

Device In

I32   Device # identification.

error in (no error)

             Describes error conditions that occur before this node runs.

Output

Device Out

I32   Same as the Device In.

Event Registration Refnum Out

             Event # identification.

error out

             Contains error information.

5.10.5 ServoEventExe

Description:

This is called when the Phidget Advanced Servo event changes.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

Event Registration Refnum In

Event # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

Event Registration Refnum Out

Same as the Event Registration Refnum In.

index

The motor index.

Value
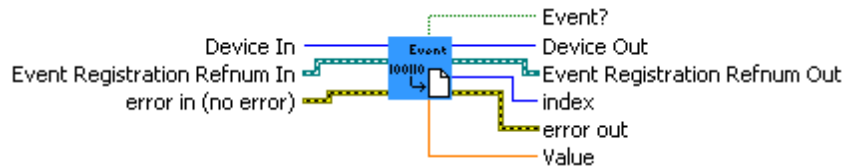
The return value of the position.

Event?

Returns the event status. (Executed is T; Not executed is F)

error out

Contains error information.

5.10.6 ServoGetEngaged

Description:

Get the engaged state of a motor. This is whether the motor is powered or not.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

index

The motor index.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

EngagedState_out

The engaged state. Possible values are 0 for False, 1 for True and others for undefined.

Engaged?

The engaged state (Boolean type). Possible values are True for Engaged and False for Not Engaged.

error out

Contains error information.

5.10.7 ServoGetPos

Description:

Get the current position of a motor.

Connector Pane:

Controls and Indicators:

Input

Device In

[I32] Device # identification.

index

[I32] The motor index.

error in (no error)

[error] Describes error conditions that occur before this node runs.

Output

Device Out

[I32] Same as the Device In.

pos_out

[DBL] The position.

error out

[error] Contains error information.

5.10.8 ServoGetPosMax

Description:

Get the maximum position that a motor can go to.

Connector Pane:

Controls and Indicators:

Input

Device In

[I32]   Device # identification.

index

[I32]   The motor index.

error in (no error)

[STR]   Describes error conditions that occur before this node runs.

Output

Device Out

[I32]   Same as the Device In.

posmax_out

[DBL]   The maximum current.

error out
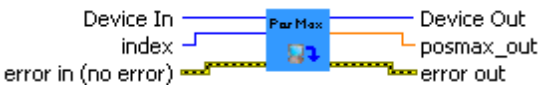
[STR]   Contains error information.

5.10.9 ServoGetPosMin

Description:

Get the minimum position that a motor can go to.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

index

The motor index.

error in (no error)

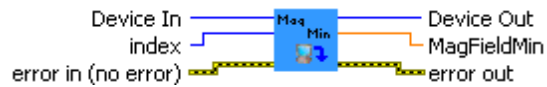Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

posmin_out

The minimum position.

error out

Contains error information.

5.10.10 ServoGetServoType

Description:

Get the servo type of a motor.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

index

The motor index.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

GetServoType

Returns the servo type. This is an enum. Please refer to **Phigets Constant** -> **ServoType**

error out

Contains error information.

5.10.11 ServoOpen

Description:

Open a PhidgetServo.

Connector Pane:

Controls and Indicators:

Input

Device In

I32

Device # identification. This function will create a new device identification if it's 0 or invalid.

Serial Number

I32

Serial number. Specify -1 to open any.

milliseconds

I32

Time to wait for the attachment. Specify 0 to wait forever. (Default is 5000)

error in (no error)

Describes error conditions that occur before this node runs.
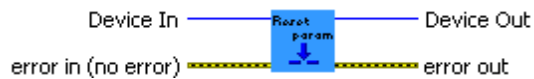
Output

Device Out

I32

Same as the Device In.

Serial Number Return

I32

Returns the serial number.

Attached?

TF

Returns the device status. (Attached is T; Not attached is F)

error out

Contains error information.

5.10.12 ServoSetEngaged

Description:

Set the engaged state of a motor. This is whether the motor is powered or not.

Connector Pane:

Controls and Indicators:

Input

Device In

I32   Device # identification.

index

I32   The motor index.

SetEngaged

TF   Set the engage state.

error in (no error)

           Describes error conditions that occur before this node runs.

Output

Device Out

I32   Same as the Device In.

error out
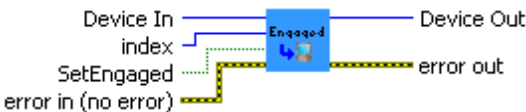
           Contains error information.

5.10.13 ServoSetPos

Description:

Set the position of a motor.

Connector Pane:

Controls and Indicators:

Input

Device In

I32   Device # identification.

index

I32   The motor index.

pos_in

DBL   The position.

error in (no error)

Describes error conditions that occur before this node runs.

**Output**

Device Out

I32   Same as the Device In.

error out

Contains error information.

5.10.14 ServoSetServoParameters

Description:

Set the servo parameters of a motor.

Connector Pane:

```
         min_us
       Device In ─────────┐  ┌───────── Device Out
           index ─────────┤Params├
         degrees ─────────┤  4  ├───────── error out
error in (no error) ──────┘  └──
          max_us
```

Controls and Indicators:

Input

Device In
[I32]   Device # identification.

index
[I32]   The motor index.

min_us
[DBL]   The minimum supported PCM in microseconds.

max_us
[DBL]   The maximum supported PCM in microseconds.

degrees
[DBL]   The degrees of rotation defined by the given PCM range.

error in (no error)
[≡≡≡]   Describes error conditions that occur before this node runs.

Output

Device Out
[I32]   Same as the Device In.

error out
[≡≡≡]   Contains error information.

5.10.15 ServoSetServoType
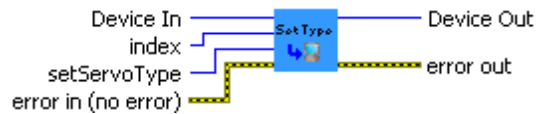
Description:

Set the servo type of a motor.

Connector Pane:

Controls and Indicators:

Input

Device In

I32   Device # identification.

index

I32   The motor index.

setServoType

I32   The servo type. This is an enum. Please refer to **Phigets Constant** -> **ServoType**

error in (no error)

Describes error conditions that occur before this node runs.

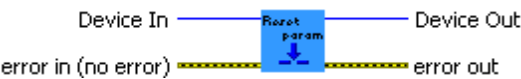**Output**

Device Out

I32   Same as the Device In.

error out

Contains error information.

5.11   Phidget Spatial

This contains VI functions for Phidget Spatial. See the product manual for more specific API details, supported functionality, units, etc.

SpatialCreate

SpatialEventClose

SpatialEventCreate

SpatialEventExe

SpatialGetAcce

SpatialGetAcceAxisCount

SpatialGetAcceMax

SpatialGetAcceMin

SpatialGetAngRate

SpatialGetAngRateMax

SpatialGetAngRateMin

SpatialGetCompassAxisCount

SpatialGetDataRate

SpatialGetDataRateMax

SpatialGetDataRateMin

SpatialGetGyroAxisCount

SpatialGetMagField

SpatialGetMagFieldMax

SpatialGetMagFieldMin

SpatialOpen

SpatialResetCompassCorrectionParameters

SpatialSetCompassCorrectionParameters

SpatialSetDataRate

SpatialZeroGyro

5.11.1 SpatialCreate

Description:

Create a Phidget Spatial handle.

Connector Pane:
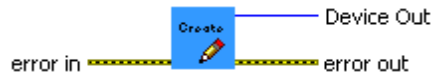
Controls and Indicators:

Input

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out
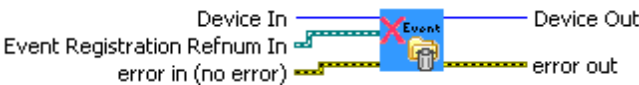
Device # identification.

error out

Contains error information.

5.11.2 SpatialEventClose

Description:

Close the Phidget Spatial event handle.

Connector Pane:
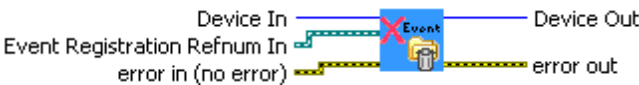
Device In ──────────────── Device Out
Event Registration Refnum In
error in (no error) ──────────── error out

Controls and Indicators:

Input

Device In
[I32]   Device # identification.

Event Registration Refnum In
[□]   Event # identification.

error in (no error)
[⚏]   Describes error conditions that occur before this node runs.

Output

Device Out
[I32]   Same as the Device In.

error out
[⚏]   Contains error information.

5.11.3   SpatialEventCreate

Description:

Set up a data change event handle.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

Event Registration Refnum Out

Event # identification.

error out

Contains error information.

5.11.4 SpatialEventExe

Description:

This is called when data come. It is called at **SpatialGetDataRate**, up to 8ms. For the rate faster than 8ms, multiple sets of data are supplied in a single event.

Connector Pane:
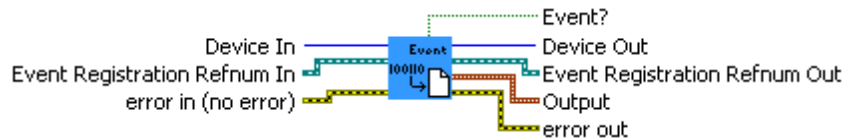
Controls and Indicators:

Input

Device In

Device # identification.

Event Registration Refnum In

Event # identification.

error in (no error)

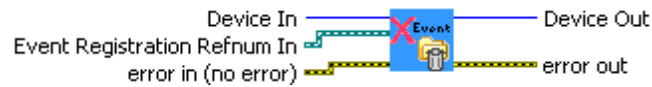Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

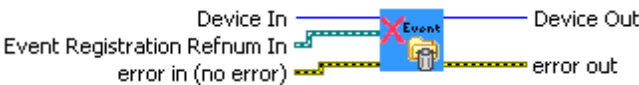Event Registration Refnum Out

Same as the Event Registration Refnum In.

Event?

Returns the event status. (Executed is T; Not executed is F)

Output

The Output Data.

DBL	acc0	Acceleration X.
DBL	acc1	Acceleration Y.
DBL	acc2	Acceleration Z.
DBL	ang0	Angular rate X.
DBL	ang1	Angular rate Y.
DBL	ang2	Angular rate Z.
DBL	mag0	Magnetic field X.
DBL	mag1	Magnetic field Y.
DBL	mag2	Magnetic field Z.
I32	sec	Timestamp in s.
I32	micsec	Timestamp in ms.

error out

Contains error information.

5.11.5 SpatialGetAcce

Description:

Get the current acceleration data of an axis.

Connector Pane:

Controls and Indicators:

Input

Device In

[I32]   Device # identification.

index

[I32]   The acceleration index. (x, y, z)

error in (no error)

[===]   Describes error conditions that occur before this node runs.

Output

Device Out

[I32]   Same as the Device In.

Acce

[DBL]   The acceleration in gs.

error out

[===]   Contains error information.

5.11.6 SpatialGetAcceAxisCount

Description:

Get the number of acceleration axes supplied by this board.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

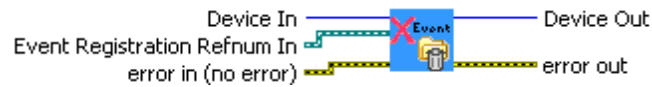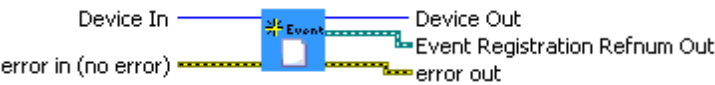Same as the Device In.

AcceAxisCount

The axis count.

error out

Contains error information.

5.11.7 SpatialGetAcceMax

Description:

Get the maximum acceleration supported by an axis.

Connector Pane:

Controls and Indicators:

Input

Device In

I32   Device # identification.

index

I32   The acceleration index. (x, y, z)

error in (no error)

    Describes error conditions that occur before this node runs.

Output

Device Out

I32   Same as the Device In.

AcceMax

DBL   The maximum acceleration.

error out

    Contains error information.

5.11.8 SpatialGetAcceMin

Description:

Get the minimum acceleration supported by an axis.

Connector Pane:

Controls and Indicators:

Input

Device In

[I32]   Device # identification.

index

[I32]   The acceleration index. (x, y, z)

error in (no error)

[≡≡]   Describes error conditions that occur before this node runs.

Output

Device Out

[I32]   Same as the Device In.

AcceMin

[DBL]   The minimum acceleration.

error out

[≡≡]   Contains error information.

5.11.9 SpatialGetAngRate

Description:

Get the current angular rate of an axis.

Connector Pane:
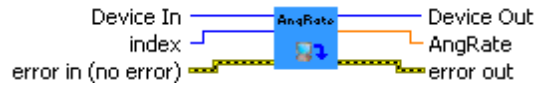
Controls and Indicators:

Input

Device In

Device # identification.

index

The angular rate index. (x, y, z)

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

AngRate

The angular rate in degrees/second.

error out

Contains error information.

5.11.10 SpatialGetAngRateMax

Description:

Get the maximum angular rate supported by an axis.

Connector Pane:

Controls and Indicators:

Input

Device In

| I32 | Device # identification.

index

| I32 | The angular rate index. (x, y, z)

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

| I32 | Same as the Device In.

AngRateMax

| DBL | The maximum angular rate.

error out

Contains error information.

5.11.11 SpatialGetAngRateMin

Description:

Get the minimum angular rate supported by an axis.
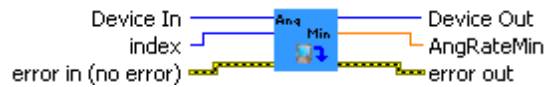
Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

index

The angular rate index. (x, y, z)

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

AngRateMin

The minimum angular rate.

error out

Contains error information.

5.11.12 SpatialGetCompassAxisCount

Description:

Get the number of compass axes supplied by this board.

Connector Pane:

Controls and Indicators:

Input

Device In

[I32]   Device # identification.

error in (no error)

[error]   Describes error conditions that occur before this node runs.

Output

Device Out

[I32]   Same as the Device In.

CompassAxisCount

[I32]   The number of compass axes.

error out

[error]   Contains error information.

5.11.13 SpatialGetDataRate

Description:

Get the data rate.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

DataRate

The data rate in milliseconds.

error out

Contains error information.

5.11.14 SpatialGetDataRateMax

Description:

Get the maximum supported data rate.

Connector Pane:

Controls and Indicators:

Input

Device In

[I32]   Device # identification.

error in (no error)

[ ]   Describes error conditions that occur before this node runs.

Output

Device Out

[I32]   Same as the Device In.

DataRateMax

[I32]   The data rate in milliseconds.

error out

[ ]   Contains error information.

5.11.15 SpatialGetDataRateMin

Description:

Get the minimum supported data rate.

Connector Pane:

Controls and Indicators:

Input

Device In

[I32]   Device # identification.

error in (no error)

[err]   Describes error conditions that occur before this node runs.

Output

Device Out

[I32]   Same as the Device In.

DataRateMin

[I32]   The data rate in milliseconds.

error out

[err]   Contains error information.

5.11.16 SpatialGetGyroAxisCount

Description:

Get the number of gyroscope axes supplied by this board.

Connector Pane:

Device In ———————— Device Out
                    Sp Gyro   GyroAxisCount
error in (no error) Count   error out

Controls and Indicators:

Input

Device In

I32   Device # identification.

error in (no error)

       Describes error conditions that occur before this node runs.

Output

Device Out
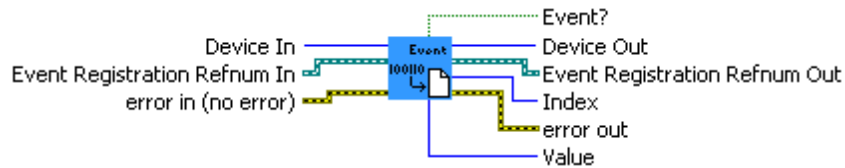
I32   Same as the Device In.

GyroAxisCount

I32   The number of gyro axes.

error out

       Contains error information.

5.11.17 SpatialGetMagField

Description:

Get the current magnetic field strength of an axis.

Connector Pane:

Controls and Indicators:

Input

Device In
[I32]   Device # identification.

index
[I32]   The magnetic field index. (x, y, z)

error in (no error)
[ ]   Describes error conditions that occur before this node runs.

Output

Device Out
[I32]   Same as the Device In.

MagField
[DBL]   The magnetic field strength in Gauss.

error out
[ ]   Contains error information.

5.11.18 SpatialGetMagFieldMax

Description:

Get the maximum magnetic field strength supported by an axis.

Connector Pane:

Controls and Indicators:

Input

Device In

[I32]   Device # identification.

index

[I32]   The magnetic field index. (x, y, z)

error in (no error)

[error]   Describes error conditions that occur before this node runs.

Output

Device Out

[I32]   Same as the Device In.

MagFieldMax

[DBL]   The maximum magnetic field strength in Gauss.

error out

[error]   Contains error information.

5.11.19 SpatialGetMagFieldMin

Description:

Get the minimum magnetic field strength supported by an axis.

Connector Pane:
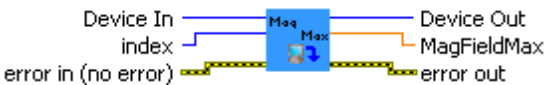
Controls and Indicators:

Input

Device In

Device # identification.

index

The magnetic field index. (x, y, z)

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

MagFieldMin

The minimum magnetic field strength in Gauss.

error out

Contains error information.

5.11.20 SpatialOpen

Description:

Open a PhidgetSpatial.

Connector Pane:

Controls and Indicators:

Input

Device In
I32
Device # identification. This function will create a new device identification if it's 0 or invalid.

Serial Number
I32
Serial number. Specify -1 to open any.

milliseconds
I32
Time to wait for the attachment. Specify 0 to wait forever. (Default is 5000)

error in (no error)
Describes error conditions that occur before this node runs.

Output

Device Out
I32
Same as the Device In.

Serial Number Return
I32
Returns the serial number.

Attached?
TF
Returns the device status. (Attached is T; Not attached is F)

error out
Contains error information.

5.11.21 SpatialResetCompassCorrectionParameters

Description:

Reset the compass correction factors. Magnetic field data will be presented directly as reported by the sensor.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

error out

Contains error information.

## 5.11.22 SpatialSetCompassCorrectionParameters

Description:

Set the compass correction factors. This can be used to correcting any sensor errors, including hard and soft iron offsets and sensor error factors.

Connector Pane:

Device In ─────────┐ Params ┌───────── Device Out
Settings ─────────┘   └───────── error out
error in (no error) ═══════════════════
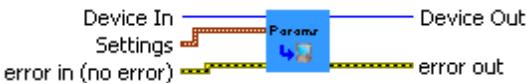
Controls and Indicators:

Input

Device In
[I32]   Device # identification.

Input Settings:

[DBL]	magField	Local magnetic field strength.
[DBL]	offset0	Axis 0 offset correction.
[DBL]	offset1	Axis 1 offset correction.
[DBL]	offset2	Axis 2 offset correction.
[DBL]	gain0	Axis 0 gain correction.
[DBL]	gain1	Axis 1 gain correction.
[DBL]	gain2	Axis 2 gain correction.
[DBL]	T0	Non-orthogonality correction factor 0.
[DBL]	T1	Non-orthogonality correction factor 1.
[DBL]	T2	Non-orthogonality correction factor 2.
[DBL]	T3	Non-orthogonality correction factor 3.
[DBL]	T4	Non-orthogonality correction factor 4.
[DBL]	T5	Non-orthogonality correction factor 5.

Settings
[sosb]

error in (no error)
[error]   Describes error conditions that occur before this node runs.

Output

Device Out
[I32]   Same as the Device In.
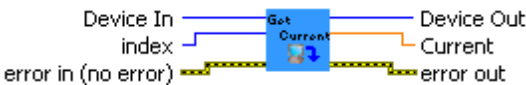
error out
[error]   Contains error information.

5.11.23 SpatialSetDataRate

Description:

Set the data rate. Note that data at rates faster then 8ms will be delivered to events as an array of data.

Connector Pane:
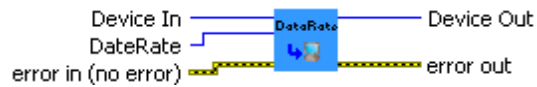
Controls and Indicators:

Input

Device In
I32   Device # identification.

DateRate
I32   The data rate in milliseconds.

error in (no error)
    Describes error conditions that occur before this node runs.

Output

Device Out
I32   Same as the Device In.

error out
    Contains error information.

5.11.24 SpatialZeroGyro

Description:

Zero the gyroscope. This takes about two seconds and the gyro axes will report 0 during the process. This should only be called when the board is not moving.

Connector Pane:

Device In ———— Device Out

error in (no error) ========= error out

Controls and Indicators:

Input

Device In

Device # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

error out

Contains error information.

5.12 Phidget Stepper

This contains VI functions for Phidget Stepper. Since Labview version 7.1 doesn't support 64bit integers, the Phidget Labview library contains special functions for Labview 7.1 marked by 71. See the product manual for more specific API details, supported functionality, units, etc.

StepperCreate

StepperEventCloseCurrentVelocity

StepperEventCloseInput

StepperEventClosePosition

StepperEventClosePosition71

StepperEventCreateCurrent

StepperEventCreateInput

StepperEventCreatePosition

StepperEventCreatePosition71

StepperEventCreateVelocity

StepperEventExeCurrentVelocity

StepperEventExeInput

StepperEventExePosition

StepperEventExePosition71

StepperGetAcceleration

StepperGetAccelerationMax

StepperGetAccelerationMin

StepperGetCurrent

StepperGetCurrentLimit

StepperGetCurrentMax

StepperGetCurrentMin

StepperGetCurrentPosition

StepperGetCurrentPosition71

StepperGetEngaged

StepperGetPositionMax

StepperGetPositionMax71

StepperGetPositionMin

StepperGetPositionMin71

StepperGetTargetPosition

StepperGetTargetPosition71

StepperGetVelocity

StepperGetVelocityLimit

StepperGetVelocityMax

StepperGetVelocityMin

StepperInputCount

StepperInputState

StepperMotorCount

StepperOpen

StepperSetAcceleration

StepperSetCurrentLimit

StepperSetCurrentPosition

StepperSetCurrentPosition71

StepperSetEngaged

StepperSetTargetPosition

StepperSetTargetPosition71

StepperSetVelocityLimit

StepperStoppedState

5.12.1 StepperCreate

Description:

Create a Phidget Stepper handle.

Connector Pane:
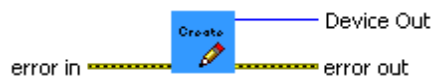
Controls and Indicators:

Input

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Device # identification.

error out

Contains error information.

5.12.2 StepperEventCloseCurrentVelocity

Description:

Close the Phidget Stepper current change event handle.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

Event Registration Refnum In

Event # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

error out

Contains error information.

5.12.3 StepperEventCloseInput

Description:

Close the Phidget Stepper input change event handle.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

Event Registration Refnum In

Event # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

error out

Contains error information.

5.12.4 StepperEventClosePosition

Description:

Close the Phidget Stepper position change event handle.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

Event Registration Refnum In

Event # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

error out

Contains error information.

5.12.5 StepperEventClosePosition71

Description:

Close the Phidget Stepper position change event handle for Labview version 7.1 only.

Connector Pane:

Controls and Indicators:

Input

Device In

I32   Device # identification.

Event Registration Refnum In

Event # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

I32   Same as the Device In.

error out

Contains error information.

5.12.6 StepperEventCreateCurrent

Description:

Set up a current change event handle.

Connector Pane:

Device In ──────────────┐   ┌────── Device Out
                        │   Event   ├── Event Registration Refnum Out
error in (no error) ════╧══════════════╧══ error out

Controls and Indicators:

Input

Device In

[I32]   Device # identification.

error in (no error)

[error]   Describes error conditions that occur before this node runs.

Output

Device Out

[I32]   Same as the Device In.

Event Registration Refnum Out

[▶▫]   Event # identification.

error out

[error]   Contains error information.

5.12.7 StepperEventCreateInput

Description:

Set up an input change event handle.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

Event Registration Refnum Out

Event # identification.

error out

Contains error information.

5.12.8 StepperEventCreatePosition

Description:

Set up a position change event handle.

Connector Pane:

Controls and Indicators:

Input

Device In

I32   Device # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

I32   Same as the Device In.

Event Registration Refnum Out

Event # identification.

error out
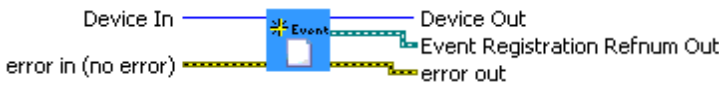
Contains error information.

5.12.9   StepperEventCreatePosition71

Description:

Set up a position change event handle for Labview version 7.1 only.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

Event Registration Refnum Out

Event # identification.

error out

Contains error information.

5.12.10 StepperEventCreateVelocity

Description:

Set up a velocity change event handle.

Connector Pane:

Controls and Indicators:

Input

Device In

I32   Device # identification.

error in (no error)

        Describes error conditions that occur before this node runs.

Output

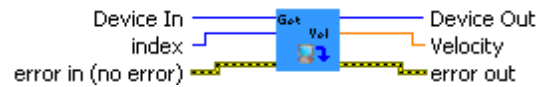Device Out

I32   Same as the Device In.

Event Registration Refnum Out

        Event # identification.

error out

        Contains error information.

5.12.11 StepperEventExeCurrentVelocity

Description:

This is called when the Phidget Stepper Current/Velocity event changes.

Connector Pane:
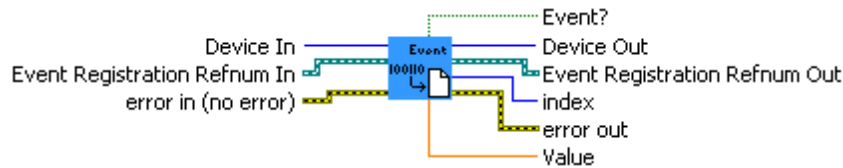
Controls and Indicators:

Input

Device In

Device # identification.

Event Registration Refnum In

Event # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

Event Registration Refnum Out

Same as the Event Registration Refnum In.

index

The motor index.

Value

The return value of related event. (E.g. for velocity change event, this value is velocity.)

Event?

Returns the event status. (Executed is T; Not executed is F)

error out

Contains error information.

5.12.12 StepperEventExeInput

Description:

This is called when the Phidget Stepper Input event changes.

Connector Pane:

Controls and Indicators:

Input

Device In
I32

Device # identification.

Event Registration Refnum In

Event # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out
I32

Same as the Device In.

Event Registration Refnum Out

Same as the Event Registration Refnum In.

index
I32

The motor index.

Value
I32

The input.

Event?
TF

Returns the event status. (Executed is T; Not executed is F)

error out

Contains error information.

5.12.13 StepperEventExePosition

Description:

This is called when the Phidget Stepper Position event changes.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

Event Registration Refnum In

Event # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

Event Registration Refnum Out

Same as the Event Registration Refnum In.

index

The motor index.

Value

The position.

Event?

Returns the event status. (Executed is T; Not executed is F)

error out

Contains error information.

5.12.14 StepperEventExePosition71

Description:

This is called when the Phidget Stepper Position event changes. This function is for Labview version 7.1 only.

Connector Pane:

Event?
Device In ─── Device Out
Event Registration Refnum In ─── Event Registration Refnum Out
error in (no error) ─── Index
error out
HighBit
LowBit

Controls and Indicators:

Input

Device In
[I32]   Device # identification.

Event Registration Refnum In
[▫]   Event # identification.

error in (no error)
[ ]   Describes error conditions that occur before this node runs.

Output

Device Out
[I32]   Same as the Device In.

Event Registration Refnum Out
[▫]   Same as the Event Registration Refnum In.

index
[I32]   The motor index.

HighBit
[I32]   The high 8 bits of the position.

LowBit   The low 32 bits of the position. Note that if the position is negative, the
[I32]   sign bit must be extended into the HighBit manually.

Event?
[TF]   Returns the event status. (Executed is T; Not executed is F)

error out
[ ]   Contains error information.

5.12.15 StepperGetAcceleration

Description:

Get the last set acceleration for a motor.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

index

The motor index.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out
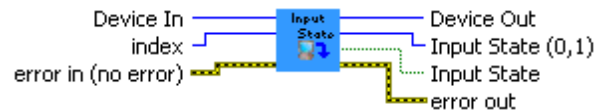
Same as the Device In.
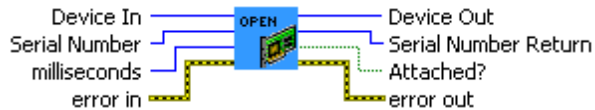
Acce

The acceleration.

error out

Contains error information.

5.12.16 StepperGetAccelerationMax

Description:

Get the maximum acceleration supported by a motor.

Connector Pane:

Controls and Indicators:

Input

Device In

[I32]   Device # identification.

index

[I32]   The motor index.

error in (no error)

[≡≡]   Describes error conditions that occur before this node runs.

Output

Device Out

[I32]   Same as the Device In.

AcceMax

[DBL]   The maximum acceleration.

error out

[≡≡]   Contains error information.

5.12.17 StepperGetAccelerationMin

Description:

Get the minimum acceleration supported by a motor.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

index

The motor index.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

AcceMin

The minimum acceleration.

error out
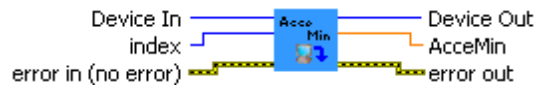
Contains error information.

5.12.18 StepperGetCurrent

Description:

Get the current current draw for a motor.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

index

The motor index.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

Current

The current.

error out

Contains error information.

5.12.19 StepperGetCurrentLimit

Description:

Get the current limit for a motor.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

index

The motor index.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

Current Limit

The current limit.

error out

Contains error information.

5.12.20 StepperGetCurrentMax

Description:

Get the maximum current limit.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

index

The motor index.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

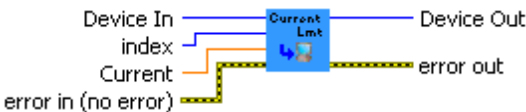CurrentMax

The maximum current limit.

error out

Contains error information.

5.12.21 StepperGetCurrentMin

Description:

Get the minimum current limit.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

index

The motor index.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

CurrentMin

The minimum current limit.

error out

Contains error information.

5.12.22 StepperGetCurrentPosition

Description:

Get the current position of a motor.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

index

The motor index.

error in (no error)

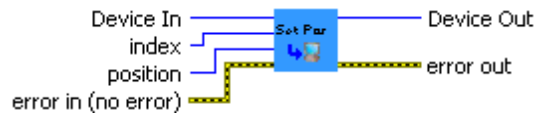Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

Position

The position.

error out

Contains error information.

5.12.23 StepperGetCurrentPosition71

Description:

Get the current position of a motor. This function is for Labview version 7.1 only.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

index

The motor index.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

Position

The position.This includes hight bit and low bit. Please refer to **StepperEventExePosition71**.

error out
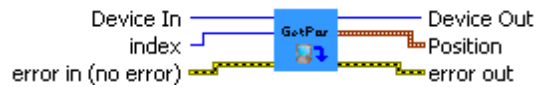
Contains error information.

5.12.24 StepperGetEngaged

Description:

Get the engaged state of a motor. This is whether the motor is powered or not.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

index

The motor index.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

EngagedState_out

The engaged state. Possible values are 0 for False, 1 for True and others for undefined.

Engaged?

The engaged state (Boolean type). Possible values are True for Engaged and False for Not Engaged.

error out

Contains error information.

5.12.25 StepperGetPositionMax

Description:
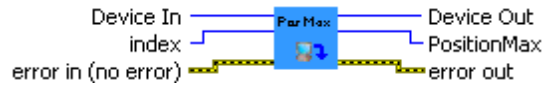
Get the maximum position that a motor can go to.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

index

The motor index.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

PositionMax

The maximum position.

error out

Contains error information.

5.12.26 StepperGetPositionMax71

Description:

Get the maximum position that a motor can go to. This function is for Labview version 7.1 only.

Connector Pane:

Device In ─────────────── Device Out
index ─┐   ┌─ Position
error in (no error) ───────── error out

Controls and Indicators:

Input

Device In
[I32]   Device # identification.

index
[I32]   The motor index.

error in (no error)
[   ]   Describes error conditions that occur before this node runs.

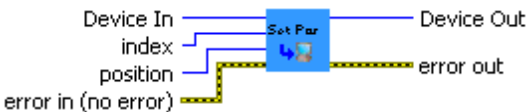**Output**

Device Out
[I32]   Same as the Device In.

Position   The position.This includes hight bit and low bit. Please refer
[906]   to **StepperEventExePosition71**.

error out
[   ]   Contains error information.

5.12.27 StepperGetPositionMin

Description:

Get the minimum position that a motor can go to.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

index

The motor index.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

PositionMin

The minimum position.

error out

Contains error information.

5.12.28 StepperGetPositionMin71

Description:

Get the minimum position that a motor can go to. This function is for Labview version 7.1 only.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

index

The motor index.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

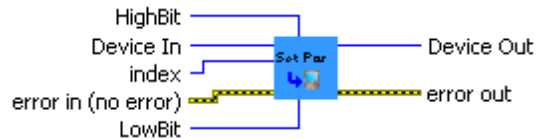Same as the Device In.

Position

The position.This includes hight bit and low bit. Please refer to **StepperEventExePosition71**.

error out
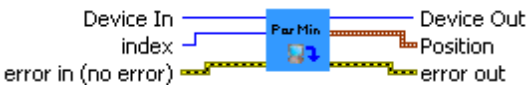
Contains error information.

5.12.29 StepperGetTargetPosition

Description:

Get the last set target position of a motor.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

index

The motor index.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

Position

The position.
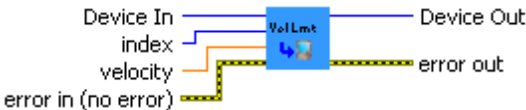
error out

Contains error information.

5.12.30 StepperGetTargetPosition71

Description:

Get the last set target position of a motor. This function is for Labview version 7.1 only.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

index

The motor index.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

Position

The position.This includes hight bit and low bit. Please refer to **StepperEventExePosition71**.

error out

Contains error information.

5.12.31 StepperGetVelocity

Description:

Get the current velocity of a motor.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

index

The motor index.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

Velocity

The velocity.

error out

Contains error information.

5.12.32 StepperGetVelocityLimit

Description:

Get the last set velocity limit for a motor.
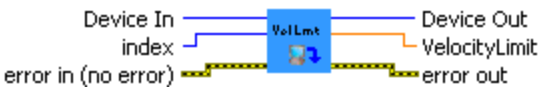
Connector Pane:

Controls and Indicators:

Input

Device In

[I32]   Device # identification.

index

[I32]   The motor index.

error in (no error)

[ERR]   Describes error conditions that occur before this node runs.

Output

Device Out

[I32]   Same as the Device In.

VelocityLimit

[DBL]   The velocity limit.

error out

[ERR]   Contains error information.

5.12.33 StepperGetVelocityMax

Description:

Get the maximum velocity that can be set for a motor.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

index

The motor index.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

VelocityMax

The maximum velocity.

error out

Contains error information.

5.12.34 StepperGetVelocityMin

Description:

Get the minimum velocity that can be set for a motor.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

index

The motor index.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

VelocityMin

The minimum velocity.

error out

Contains error information.

5.12.35 StepperInputCount

Description:

Get the number of digital inputs supported by this board.

Connector Pane:

Controls and Indicators:

Input

Device In
I32   Device # identification.

error in (no error)   Describes error conditions that occur before this node runs.

Output

Device Out
I32   Same as the Device In.

count
I32   The digital input count.

error out   Contains error information.

5.12.36 StepperInputState

Description:

Get the state of a digital input.

Connector Pane:

Controls and Indicators:

Input

Device In

[I32]   Device # identification.

index

[I32]   The motor index.

error in (no error)

[error]   Describes error conditions that occur before this node runs.

Output

Device Out

[I32]   Same as the Device In.

Input State (0,1)

[I32]   The input state. Possible values are 0 for False, 1 for True and others for undefined.

Input State

[TF]   The input state (Boolean type).

error out

[error]   Contains error information.

5.12.37 StepperMotorCount

Description:

Get the number of motors supported by this controller.

Connector Pane:
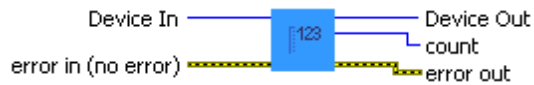
Controls and Indicators:

Input

Device In

Device # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

count

The motor count.

error out

Contains error information.

5.12.38 StepperOpen

Description:

Open a PhidgetStepper.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification. This function will create a new device identification if it's 0 or invalid.

Serial Number

Serial number. Specify -1 to open any.

milliseconds

Time to wait for the attachment. Specify 0 to wait forever. (Default is 5000)

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

Serial Number Return

Returns the serial number.

Attached?

Returns the device status. (Attached is T; Not attached is F)

error out

Contains error information.

5.12.39 StepperSetAcceleration
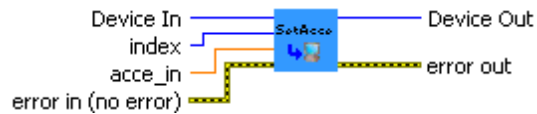
Description:

Set the acceleration for a motor.

Connector Pane:

Controls and Indicators:

Input

Device In
I32   Device # identification.

index
I32   The motor index.

acce_in
DBL   The acceleration.

error in (no error)
             Describes error conditions that occur before this node runs.

Output

Device Out
I32   Same as the Device In.

error out
             Contains error information.

5.12.40 StepperSetCurrentLimit

Description:

Set the current limit for a motor.

Connector Pane:

Controls and Indicators:

Input

Device In
[I32]   Device # identification.

index
[I32]   The motor index.

Current
[DBL]   The current limit.

error in (no error)
[≡]   Describes error conditions that occur before this node runs.

Output

Device Out
[I32]   Same as the Device In.

error out
[≡]   Contains error information.

5.12.41 StepperSetCurrentPosition

Description:

Set the current position of a motor. It will not move the motor, just update the position value.

Connector Pane:

Controls and Indicators:

Input

Device In
I32   Device # identification.

index
I32   The motor index.

position
I64   The position.

error in (no error)   Describes error conditions that occur before this node runs.

Output

Device Out
I32   Same as the Device In.

error out   Contains error information.

5.12.42 StepperSetCurrentPosition71

Description:

Set the current position of a motor. It will not move the motor, just update the position value. This function is for Labview version 7.1 only.
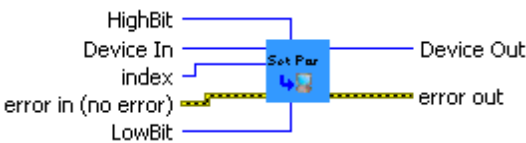
Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

index

The motor index.

HighBit

The high 8 bits of the position.

LowBit

The low 32 bits of the position. Note that if the position is negative, the sign bit must be extended into the HighBit manually.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

error out

Contains error information.

5.12.43 StepperSetEngaged

Description:

Set the engaged state of a motor. This is whether the motor is powered or not.

Connector Pane:

Controls and Indicators:

Input

Device In

I32   Device # identification.

index

I32   The motor index.

SetEngaged

TF   Set the engage state.

error in (no error)

               Describes error conditions that occur before this node runs.

Output

Device Out

I32   Same as the Device In.

error out
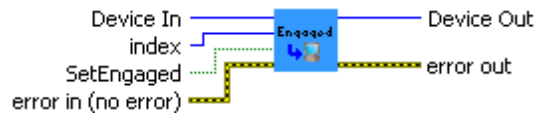
               Contains error information.

5.12.44 StepperSetTargetPosition

Description:

Set the target position of a motor.

Connector Pane:

Controls and Indicators:

Input

Device In
[I32]   Device # identification.

index
[I32]   The motor index.

position
[I64]   The position.

error in (no error)
[≡≡≡]   Describes error conditions that occur before this node runs.

Output

Device Out
[I32]   Same as the Device In.

error out
[≡≡≡]   Contains error information.

5.12.45 StepperSetTargetPosition71

Description:

Set the target position of a motor. This function is for Labview version 7.1 only.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

index

The motor index.

HighBit

The high 8 bits of the position.

LowBit

The low 32 bits of the position. Note that if the position is negative, the sign bit must be extended into the HighBit manually.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

error out

Contains error information.

5.12.46 StepperSetVelocityLimit

Description:

Set the velocity limit for a motor.

Connector Pane:

Controls and Indicators:

Input

Device In

I32   Device # identification.

index

I32   The motor index.

velocity

DBL   The velocity limit.

error in (no error)

             Describes error conditions that occur before this node runs.

Output
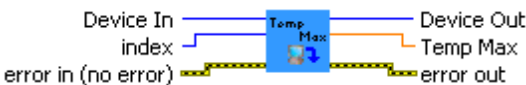
Device Out

I32   Same as the Device In.

error out

             Contains error information.

5.12.47 StepperStoppedState

Description:

Get the stopped state of a motor. This is true when the motor is not moving and there are no outstanding commands.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

index

The motor index.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

StoppedState

The stopped state. Possible values are 0 for False, 1 for True and others for undefined.

Stopped?

The stopped state (Boolean type). Possible values are True for Stopped and False for Not Stopped.

error out

Contains error information.

5.13 Phidget Temperature Sensor

This contains VI functions for Phidget Temperature Sensor. See the product manual for more specific API details, supported functionality, units, etc.

TempCreate

TempEventClose

TempEventCreate

TempEventExe

TempGetAmbient

TempGetAmbientMax

TempGetAmbientMin

TempGetPotential

TempGetPotentialMax

TempGetPotentialMin

TempGetTemperature

TempGetTemperatureMax

TempGetTemperatureMin

TempGetThermocoupleType

TempGetTrigger

TempInputCount

TempOpen

TempSetThermocoupleType

TempSetTrigger

5.13.1 TempCreate

Description:

Create a Phidget Temperature Sensor handle.

Connector Pane:

Controls and Indicators:

Input

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Device # identification.

error out

Contains error information.

5.13.2 TempEventClose

Description:

Close the Phidget Temperature Sensor event handle.

Connector Pane:

Controls and Indicators:

Input

Device In

I32   Device # identification.

Event Registration Refnum In

     Event # identification.

error in (no error)

     Describes error conditions that occur before this node runs.

Output
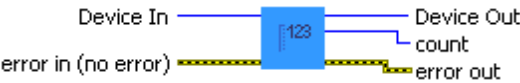
Device Out

I32   Same as the Device In.

error out

     Contains error information.

5.13.3 TempEventCreate

Description:

Set up a temperature change event handle.
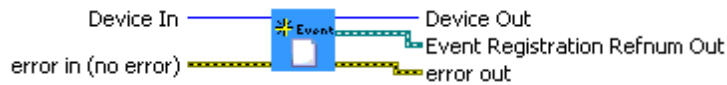
Connector Pane:

Device In ——————— Device Out
error in (no error) ——————— Event Registration Refnum Out
error out

Controls and Indicators:

Input

Device In

I32   Device # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

I32   Same as the Device In.

Event Registration Refnum Out

Event # identification.

error out

Contains error information.

5.13.4 TempEventExe

Description:

This is called when the temperature changes by more then the change trigger.

Connector Pane:

Controls and Indicators:

Input

Device In
Device # identification.

Event Registration Refnum In
Event # identification.

error in (no error)
Describes error conditions that occur before this node runs.

Output

Device Out
Same as the Device In.

Event Registration Refnum Out
Same as the Event Registration Refnum In.

index
The thermocouple index.

Value
The temperature.

Event?
Returns the event status. (Executed is T; Not executed is F)

error out
Contains error information.

5.13.5 TempGetAmbient

Description:

Get the ambient (board) temperature.

Connector Pane:

Controls and Indicators:

Input

Device In

I32 | Device # identification.

error in (no error)

| Describes error conditions that occur before this node runs.

Output

Device Out

I32 | Same as the Device In.

Ambient

DBL | The ambient (board) temperature.

error out

| Contains error information.

5.13.6 TempGetAmbientMax

Description:

Get the maximum temperature that the ambient onboard temperature sensor can measure.

Connector Pane:

Controls and Indicators:

Input

Device In

[I32]   Device # identification.

error in (no error)

[===]   Describes error conditions that occur before this node runs.

Output

Device Out

[I32]   Same as the Device In.

Ambient Max

[DBL]   The maximum temperature.

error out

[===]   Contains error information.

5.13.7 TempGetAmbientMin

Description:

Get the minimum temperature that the ambient onboard temperature sensor can measure.

Connector Pane:
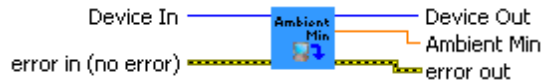
Controls and Indicators:

Input

Device In

I32   Device # identification.

error in (no error)

               Describes error conditions that occur before this node runs.

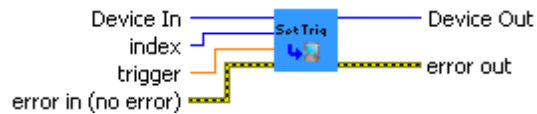**Output**

Device Out

I32   Same as the Device In.

Ambient Min

DBL   The minimum temperature.

error out

               Contains error information.

5.13.8 TempGetPotential

Description:

Get the currently sensed potential for a thermocouple input.
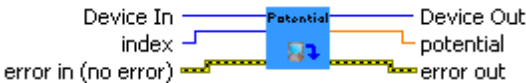
Connector Pane:

Controls and Indicators:

Input

Device In
I32   Device # identification.

index
I32   The thermocouple index.

error in (no error)
       Describes error conditions that occur before this node runs.

Output

Device Out
I32   Same as the Device In.

potential
DBL   The potential.

error out
       Contains error information.

5.13.9 TempGetPotentialMax

Description:

Get the maximum potential that a thermocouple input can measure.

Connector Pane:

Device In ─────── Device Out
index ─────── PotentialMax
error in (no error) ─────── error out

Controls and Indicators:

Input

Device In

[I32] Device # identification.

index

[I32] The thermocouple index.

error in (no error)

[error] Describes error conditions that occur before this node runs.

Output

Device Out

[I32] Same as the Device In.

PotentialMax

[DBL] The maximum potential.

error out
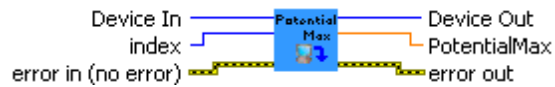
[error] Contains error information.

5.13.10 TempGetPotentialMin

Description:

Get the minimum potential that a thermocouple input can measure.

Connector Pane:

Controls and Indicators:

Input

Device In

[I32]   Device # identification.

index

[I32]   The thermocouple index.

error in (no error)

[5u]   Describes error conditions that occur before this node runs.

Output

Device Out

[I32]   Same as the Device In.

PotentialMin

[DBL]   The minimum potential.

error out
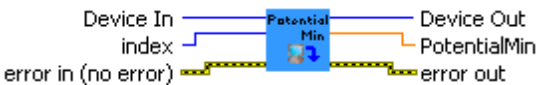
[5u]   Contains error information.

5.13.11 TempGetTemperature

Description:

Get the temperature measured by a thermocouple input.
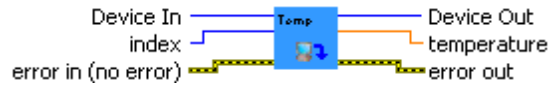
Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

index

The thermocouple index.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

temperature

The temperature.

error out

Contains error information.

5.13.12 TempGetTemperatureMax

Description:

Get the maximum temperature that can be measured by a thermocouple input. This depends on the type of thermocouple attached, as well as the ambient temperature.

Connector Pane:

Controls and Indicators:

Input

Device In

[I32]   Device # identification.

index

[I32]   The thermocouple index.

error in (no error)

[error]   Describes error conditions that occur before this node runs.

Output

Device Out

[I32]   Same as the Device In.

temperature

[DBL]   The maximum temperature.

error out

[error]   Contains error information.

5.13.13 TempGetTemperatureMin

Description:

Get the minimum temperature that can be measured by a thermocouple input. This depends on the type of thermocouple attached, as well as the ambient temperature.

Connector Pane:
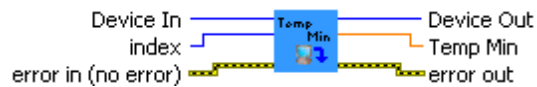
Controls and Indicators:

Input

Device In

Device # identification.

index

The thermocouple index.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

Temp Min

The minimum temperature.

error out

Contains error information.

5.13.14 TempGetThermocoupleType

Description:

Get the type of thermocouple set to be at a thermocouple input. By default this is K-Type (1).

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

index

The thermocouple index.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

GetServoType

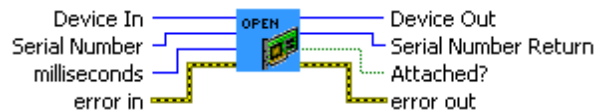The thermocouple type. This is an enum. Please refer to **Phigets Constant** -> **ThermocoupleType**

error out

Contains error information.

5.13.15 TempGetTrigger

Description:
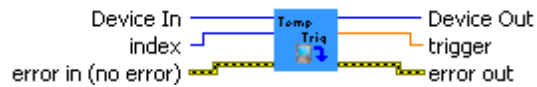
Get the change trigger for a thermocouple input.

Connector Pane:

Controls and Indicators:

Input

Device In
I32   Device # identification.

index
I32   The thermocouple index.

error in (no error)
    Describes error conditions that occur before this node runs.

Output

Device Out
I32   Same as the Device In.

trigger
DBL   The change trigger.

error out
    Contains error information.

5.13.16 TempInputCount

Description:

Get the number of thermocouple inputs supported by this board.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

count

The thermocouple input count.

error out

Contains error information.

5.13.17 TempOpen

Description:

Open a Phidget Temperature Sensor.

Connector Pane:

Controls and Indicators:

Input

Device In
`I32`
Device # identification. This function will create a new device identification if it's 0 or invalid.

Serial Number
`I32`
Serial number. Specify -1 to open any.

milliseconds
`I32`
Time to wait for the attachment. Specify 0 to wait forever. (Default is 5000)

error in (no error)
Describes error conditions that occur before this node runs.

Output

Device Out
`I32`
Same as the Device In.

Serial Number Return
`I32`
Returns the serial number.

Attached?
`TF`
Returns the device status. (Attached is T; Not attached is F)

error out
Contains error information.

5.13.18 TempSetThermocoupleType

Description:

Set the type of thermocouple plugged into a thermocouple input. By default this is K-Type.

Connector Pane:
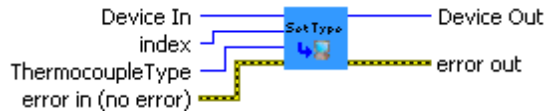
Controls and Indicators:

Input

Device In

Device # identification.

index

The thermocouple index.

setServoType

The thermocouple type. This is an enum. Please refer to **Phigets Constant** -> **ThermocoupleType**

error in (no error)

Describes error conditions that occur before this node runs.
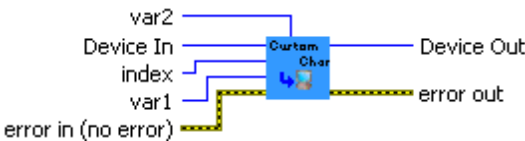
Output

Device Out

Same as the Device In.

error out

Contains error information.

5.13.19 TempSetTrigger

Description:

Set the change trigger for a thermocouple input.

Connector Pane:

Controls and Indicators:

Input

Device In

I32   Device # identification.

index

I32   The thermocouple index.

trigger

DBL   The change trigger.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

I32   Same as the Device In.

error out

Contains error information.

5.14   Phidget TextLCD

This contains VI functions for Phidget TextLCD. See the product manual for more specific API details, supported functionality, units, etc.

TextCreate

TextGetBacklightState

TextGetBrightness

TextGetColumnCount

TextGetContrast

TextGetCursorBlinkState

TextGetCursorState

TextGetRowCount

TextOpen

TextSetBacklightState

TextSetBrightness

TextSetCharacter

TextSetContrast

TextSetCursorBlinkState

TextSetCursorState

TextSetDisplayChar

TextSetDisplayString

5.14.1 TextCreate

Description:

Create a Phidget TextLCD handle.

Connector Pane:

Controls and Indicators:

Input

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Device # identification.

error out

Contains error information.

5.14.2 TextGetBacklightState
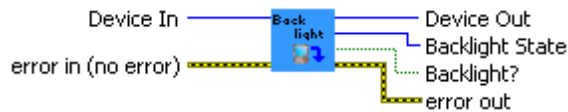
Description:

Get the state of the backlight.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

Backlight State

The backlight state. Possible values are 0 for False, 1 for True and others for undefined.

Backlight?

The backlight state (Boolean type).

error out

Contains error information.

5.14.3 TextGetBrightness

Description:

Get the brightness of the backlight. Not supported on all TextLCDs.

Connector Pane:

Device In —————   ——— Device Out
                  Bright   └ Brightness
error in (no error) ═══════   ┈┈ error out

Controls and Indicators:

Input

Device In
[I32]   Device # identification.

error in (no error)
[≡≡≡]   Describes error conditions that occur before this node runs.

Output

Device Out
[I32]   Same as the Device In.

Brightness
[I32]   The backlight brightness (0-255).

error out
[≡≡≡]   Contains error information.

5.14.4 TextGetColumnCount

Description:

Get the number of columns per supported by this display.
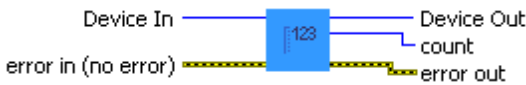
Connector Pane:

Controls and Indicators:

Input

Device In

[I32]   Device # identification.

error in (no error)

[≡≡]   Describes error conditions that occur before this node runs.

Output
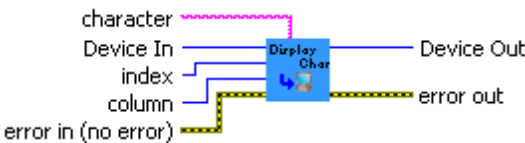
Device Out

[I32]   Same as the Device In.

count

[I32]   The column count.

error out

[≡≡]   Contains error information.

5.14.5 TextGetContrast

Description:

Get the last set contrast value.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

Contrast

The contrast (0-255).

error out

Contains error information.

5.14.6 TextGetCursorBlinkState

Description:

Get the cursor blink state.
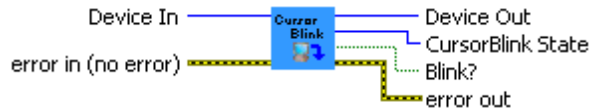
Connector Pane:

Controls and Indicators:

Input

Device In

I32   Device # identification.

error in (no error)

        Describes error conditions that occur before this node runs.

Output

Device Out

I32   Same as the Device In.

CursorBlink State

I32   The cursor blink state. Possible values are 0 for False, 1 for True and others for undefined.

Blink?

TF   The cursor blink state (Boolean type).

error out

        Contains error information.

5.14.7 TextGetCursorState

Description:

Get the cursor visible state.
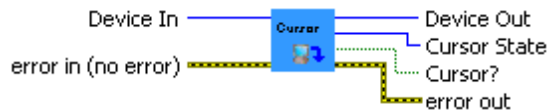
Connector Pane:

Controls and Indicators:

Input

Device In

[I32]   Device # identification.

error in (no error)

[ ]   Describes error conditions that occur before this node runs.

Output

Device Out

[I32]   Same as the Device In.

Cursor State   The state of the cursor. Possible values are 0 for False, 1 for True and others for

[I32]   undefined.

Cursor?

[TF]   The state of the cursor (Boolean type).

error out

[ ]   Contains error information.

5.14.8 TextGetRowCount

Description:

Get the number of rows supported by this display.
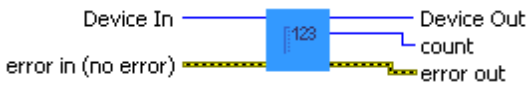
Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

count

The row count.

error out

Contains error information.

5.14.9 TextOpen

Description:

Open a Phidget TextLCD.

Connector Pane:

Controls and Indicators:

Input

Device In
I32

Device # identification. This function will create a new device identification if it's 0 or invalid.

Serial Number
I32

Serial number. Specify -1 to open any.

milliseconds
I32

Time to wait for the attachment. Specify 0 to wait forever. (Default is 5000)

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out
I32

Same as the Device In.

Serial Number Return
I32

Returns the serial number.

Attached?
TF

Returns the device status. (Attached is T; Not attached is F)

error out

Contains error information.

5.14.10 TextSetBacklightState

Description:

Set the state of the backlight.

Connector Pane:
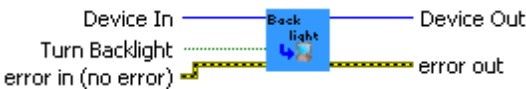
Controls and Indicators:

Input

Device In

Device # identification.

Turn Backlight

Set the backlight state.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

error out

Contains error information.

5.14.11 TextSetBrightness

Description:

Set the brightness of the backlight. Not supported on all TextLCDs

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

Brightness

The backlight brightness (0-255).

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

error out

Contains error information.

5.14.12 TextSetCharacter

Description:

Set a custom character. See the product manual for more information.

Connector Pane:

Controls and Indicators:

Input

Device In

[I32]   Device # identification.

index

[I32]   The custom character index (8-15).

var1

[I32]   The first part of the custom character.

var2

[I32]   The second part of the custom character.

error in (no error)

[图]   Describes error conditions that occur before this node runs.

Output

Device Out

[I32]   Same as the Device In.

error out

[图]   Contains error information.

5.14.13 TextSetContrast

Description:

Set the last set contrast value.

Connector Pane:

Controls and Indicators:

Input

Device In

[I32]   Device # identification.

Contrast

[I32]   The contrast (0-255).

error in (no error)

[error]   Describes error conditions that occur before this node runs.

Output

Device Out

[I32]   Same as the Device In.

error out

[error]   Contains error information.

5.14.14 TextSetCursorBlinkState

Description:

Set the cursor blink state.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

Cursor Blink

Set the cursor blink state.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

error out

Contains error information.

5.14.15 TextSetCursorState

Description:

Set the cursor visible state.

Connector Pane:
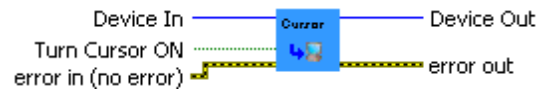
Controls and Indicators:

Input

Device In
I32   Device # identification.

Turn Cursor ON
TF   Set the state of the cursor.

error in (no error)   Describes error conditions that occur before this node runs.

Output

Device Out
I32   Same as the Device In.

error out   Contains error information.

5.14.16 TextSetDisplayChar

Description:

Set a single character on the display.

Connector Pane:

Controls and Indicators:

Input

Device In

Device # identification.

index

The row index.

column

The column index.

character

The character to display.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

error out

Contains error information.

5.14.17 TextSetDisplayString

Description:

Set a row on the display.

Connector Pane:
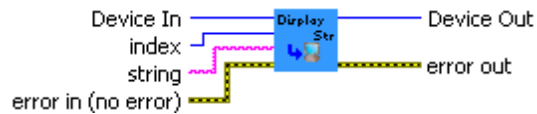
Controls and Indicators:

Input

Device In

Device # identification.

index

The row index.

string

The string to display. Make sure this is not longer then **TextGetColumnCount**.

error in (no error)

Describes error conditions that occur before this node runs.

Output

Device Out

Same as the Device In.

error out

Contains error information.

5.15   Phidget TextLED

Discontinued.

5.16 Phidget Weight Sensor

Discontinued.

6   Phidgets Constants

This section describes each of the Phidgets constant used by different Phidgets.

ServoType

ThermocoupleType

CodeInfo

IREncoding

IRLength

6.1   ServoType

An enum value with the following definition:

Value	Comments
PHIDGET_SERVO_DEFAULT = 1,	Default - This is what the servo API been historically used, originally based on the Futaba FP-S148
PHIDGET_SERVO_RAW_us_MODE = 2,	Raw us mode - all position, velocity, acceleration functions are specified in microseconds rather then degrees
PHIDGET_SERVO_HITEC_HS322HD = 3,	HiTec HS-322HD Standard Servo
PHIDGET_SERVO_HITEC_HS5245MG = 4,	HiTec HS-5245MG Digital Mini Servo
PHIDGET_SERVO_HITEC_805BB = 5,	HiTec HS-805BB Mega Quarter Scale Servo
PHIDGET_SERVO_HITEC_HS422 = 6,	HiTec HS-422 Standard Servo
PHIDGET_SERVO_TOWERPRO_MG90 = 7,	Tower Pro MG90 Micro Servo
PHIDGET_SERVO_HITEC_HSR1425CR = 8,	HiTec HSR-1425CR Continuous Rotation Servo
PHIDGET_SERVO_HITEC_HS785HB = 9,	HiTec HS-785HB Sail Winch Servo
PHIDGET_SERVO_HITEC_HS485HB = 10,	HiTec HS-485HB Deluxe Servo
PHIDGET_SERVO_HITEC_HS645MG = 11,	HiTec HS-645MG Ultra Torque Servo
PHIDGET_SERVO_HITEC_815BB = 12,	HiTec HS-815BB Mega Sail Servo
PHIDGET_SERVO_FIRGELLI_L12_30_50_06_R = 13,	Firgelli L12 Linear Actuator 30mm 50:1
PHIDGET_SERVO_FIRGELLI_L12_50_100_06_R = 14,	Firgelli L12 Linear Actuator 50mm 100:1
PHIDGET_SERVO_FIRGELLI_L12_50_210_06_R = 15,	Firgelli L12 Linear Actuator 50mm 210:1
PHIDGET_SERVO_FIRGELLI_L12_100_50_06_R = 16,	Firgelli L12 Linear Actuator 100mm 50:1
PHIDGET_SERVO_FIRGELLI_L12_100_100_06_R = 17,	Firgelli L12 Linear Actuator 100mm 100:1
PHIDGET_SERVO_USER_DEFINED = others	Undefined

6.2   ThermocoupleType

An enum value with the following definition:

Value	Comments
PHIDGET_TEMPERATURE_SENSOR_K_TYPE = 1,	K-Type thermocouple
PHIDGET_TEMPERATURE_SENSOR_J_TYPE = 2,	J-Type thermocouple
PHIDGET_TEMPERATURE_SENSOR_E_TYPE = 3,	E-Type thermocouple
PHIDGET_TEMPERATURE_SENSOR_T_TYPE = 4,	T-Type thermocouple
PHIDGET_SERVO_USER_DEFINED = others	Undefined

6.3   CodeInfo

The PhidgetIR CodeInfo structure contains all information needed to transmit a code, apart from the actual code data. Some values can be set to null to select defaults. See the product manual for more information.

Data Type	Item	Comments
I32	**bitCount**	Number of bits in the code.
I32	**encoding**	Encoding used to encode the data. This is an enum. Please refer to **Phigets Constant** -> **IREncoding**
I32	**length**	Constant or Variable length encoding. This is an enum. Please refer to **Phigets Constant** -> **IRLength**
I32	**gap**	Gap time in us.
I32	**trail**	Trail time in us - can be 0 for none.
I32	**header** [2]	Header pulse and space - can be 0 for none.
I32	**one** [2]	Pulse and Space times to represent a '1' bit, in us.
I32	**zero** [2]	Pulse and Space times to represent a '0' bit, in us.
I32	**repeat** [26]	A series or pulse and space times to represent the repeat code. Start and end with pulses and null terminate. Set to 0 for none.
I32	**min_repeat**	Minimum number of times to repeat a code on transmit.
U8	**toggle_mask** [16]	Bit toggles, which are applied to the code after each transmit.
I32	**carrierFrequency**	Carrier frequency in Hz - defaults to 38kHz.
I32	**dutyCycle**	Duty Cycle in percent (10-50). Defaults to 33.

6.4   IREncoding

The PhidgetIR supports these data encodings:

Value	Comments
PHIDGET_IR_ENCODING_UNKNOWN = 1,	Unknown - the default value
PHIDGET_IR_ENCODING_SPACE = 2,	Space encoding, or Pulse Distance Modulation
PHIDGET_IR_ENCODING_PULSE = 3,	Pulse encoding, or Pulse Width Modulation
PHIDGET_IR_ENCODING_BIPHASE = 4,	Bi-Phase, or Manchester encoding
PHIDGET_IR_ENCODING_RC5 = 5,	RC5 - a type of Bi-Phase encoding
PHIDGET_IR_ENCODING_RC6 = 6,	RC6 - a type of Bi-Phase encoding
Others	Undefined

6.5   IRLength

The PhidgetIR supports these encoding lengths:

Value	Comments
PHIDGET_IR_LENGTH_UNKNOWN = 1,	Unknown - the default value
PHIDGET_IR_LENGTH_CONSTANT = 2,	Constant - the bitstream + gap length is constant
PHIDGET_IR_LENGTH_VARIABLE = 3,	Variable - the bitstream has a variable length with a constant gap
Others	Undefined

Index

- L -

- M -

- P -

Phidgets Inc.

Unit 1, 6115 4th Street S.E.
Calgary, AB
Canada T2H 2H9
Telephone: 1-403-282-7335
Facsimile: 1-403-282-7332
E-mail: support@phidgets.com
Web site: www.phidgets.com