Phidgets

LiveCode API Manual
Contents

Introduction
4 Overview
4 How to use Phidgets
4 Class Hierarchy

Phidget
5 Functions
8 Events

PhidgetInterfaceKit
10 Functions
16 Events
Introduction

Overview
At the moment, Phidgets supports development in LiveCode 4.5 for InterfaceKit 8/8/8 Devices on local connections in Windows XP SP3 and Mac OS X 10.6.4 environments. This manual describes the Application Programming Interface (API) for the InterfaceKit 8/8/8, as exposed by the LiveCode library.

How to use Phidgets
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 library is written and available for the InterfaceKit 8/8/8 in Windows XP and Mac OS 10.6.4 environments only.

Refer to the InterfaceKit product manual and the Programming Manual for more detailed, language unspecific API documentation. Also, there are an InterfaceKit example available for download.

Class Hierarchy
• Phidget
  • PhidgetInterfaceKit
Phidget

Class documentation for Phidget. This is the base class from which all other device classes inherit. These calls are common to all Phidgets. See the programming manual for more specific API details, supported functionality, units, etc.

Functions

phidgets_open

Opens a phidget.
get phidgets_open(phid, serialNumber)

Parameters:

phid
A Phidget handle.

SerialNumber
Serial number of the Phidget to open. Put -1 to open any.

phidgets_close

Closes a Phidget.
get phidgets_close(phid)

Parameters:

phid
A opened Phidget handle.

phidgets_delete

Frees a Phidget handle.
get phidgets_delete(phid)

Parameters:

phid
A closed Phidget handle.
**phidgets_getDeviceName**

Gets the specific name of a Phidget.

get phidgets_getDeviceName(phid, "deviceName")

**Parameters:**

- **phid**
  - A Phidget handle.
- **deviceName**
  - The name of the variable to store the device name (in parentheses).

**phidgets_getSerialNumber**

Gets the serial number of a Phidget.

get phidgets_getSerialNumber(phid, "serialNumber")

**Parameters:**

- **phid**
  - A Phidget handle.
- **SerialNumber**
  - The name of the variable to store the serial number (in parentheses).

**phidgets_getDeviceVersion**

Gets the firmware version of a Phidget.

get phidgets_getDeviceVersion(phid, "deviceVersion")

**Parameters:**

- **phid**
  - A Phidget handle.
- **deviceVersion**
  - The name of the variable to store the device version (in parentheses).

**phidgets_getDeviceStatus**

Gets the attached status of a Phidget.

get phidgets_getDeviceStatus(phid, "deviceStatus")

**Parameters:**

- **phid**
  - A Phidget handle.
- **deviceStatus**
  - The name of the variable to store the attached status of a Phidget (in parentheses).
**phidgets_getErrorDescription**

Gets the description for an error code.

```plaintext
get phidgets_getErrorDescription(errorCode, “errorString”)
```

**Parameters:**

- **errorCode**
  - The error code to get the description of.

- **errorString**
  - The name of the variable to store the error description (in parentheses).

**phidgets_waitForAttachment**

Waits for attachment to happen. This can be called right after calling phidgets_open, as an alternative to using the attach handler.

```plaintext
get phidgets_waitForAttachment(phid, milliseconds)
```

**Parameters:**

- **phid**
  - An opened Phidget handle.

- **milliseconds**
  - Time to wait for the attachment. Specify 0 to wait forever.
Events

Note that these events are actually members of each Phidget device class rather then the base class. However, since they are common to the InterfaceKit 8/8/8, they are documented here.

**phidgets_set_OnDetach_Handler**

Sets a detach handler callback command. This is called when this Phidget is unplugged from the system.

```
get phidgets_set_OnDetach_Handler(phid, "commandName")
```

**Parameters:**

*phid*  
A Phidget handle.

*commandName*  
The name of the command to be called(in parentheses). The command must be in the script of the card that called phidgets_set_OnDetach_Handler or its stack. The form of the command is:

```
command commandName phid
...
end commandName
```

**phidgets_set_OnAttach_Handler**

Sets an attach handler callback command. This is called when this Phidget is plugged into the system, and is ready for use.

```
get phidgets_set_OnAttach_Handler(phid, "commandName")
```

**Parameters:**

*phid*  
A Phidget handle.

*commandName*  
The name of the command to be called(in parentheses). The command must be in the script of the card that called phidgets_set_OnAttach_Handler or its stack. The form of the command is:

```
command commandName phid
...
end commandName
```
**phidgets_set_OnError_Handler**

Sets an error handler callback command. This is called when an asynchronous error occurs.

```plaintext
get phidgets_set_OnError_Handler(phid, "commandName")
```

**Parameters:**

- **phid**
  A Phidget handle.

- **commandName**
  The name of the command to be called (in parentheses). The command must be in the script of the card that called `phidgets_set_OnError_Handler` or its stack. The form of the command is:

```plaintext
command commandName phid errorCode errorString

end commandName
```
PhidgetInterfaceKit

Class documentation for PhidgetInterfaceKit. This class contains all calls specific to the Phidget Interface Kit. See the product manual for more specific API details, supported functionality, units, etc.

Functions

phidgetsInterfaceKit_create

Creates a Phidget Interface Kit handle.

get phidgetsInterfaceKit_create("phid")

Parameters:

phid

The name of the variable to store the Phidget Interface Kit handle (in parentheses).

phidgetsInterfaceKit_getInputCount

Gets the number of digital inputs supported by this board.

get phidgetsInterfaceKit_getInputCount(phid, "count")

Parameters:

phid

An attached Phidget Interface Kit handle.

count

The name of the variable to store the digital input count (in parentheses).

phidgetsInterfaceKit_getInputState

Gets the state of a digital input.

get phidgetsInterfaceKit_getInputState(phid, index, "inputState")

Parameters:

phid

An attached Phidget Interface Kit handle.

index

The input index.

inputState

The name of the variable to store the input state (in parentheses). Possible values are 1(TRUE) or 0(FALSE).
**phidgetsInterfaceKit_getOutputCount**

gets the number of digital outputs supported by this board.

```plaintext
get phidgetsInterfaceKit_getOutputCount(phid, "count")
```

**Parameters:**

- **phid**
  - An attached Phidget Interface Kit handle.

- **count**
  - The name of the variable to store the output count (in parentheses).

**phidgetsInterfaceKit_getOutputState**

gets the state of a digital output.

```plaintext
get phidgetsInterfaceKit_getOutputState(phid, index, "outputState")
```

**Parameters:**

- **phid**
  - An attached Phidget Interface Kit handle.

- **index**
  - The output index.

- **outputState**
  - The name of the variable to store the output state (in parentheses). Possible values are 1(TRUE) or 0(FALSE).

**phidgetsInterfaceKit_setOutputState**

sets the state of a digital output.

```plaintext
get phidgetsInterfaceKit_setOutputState(phid, index, outputState)
```

**Parameters:**

- **phid**
  - An attached Phidget Interface Kit handle.

- **index**
  - The output index.

- **outputState**
  - The output state. Possible values are 1(TRUE) or 0(FALSE).
**phidgetsInterfaceKit_getSensorCount**

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

```plaintext
get phidgetsInterfaceKit_getSensorCount(phid, "count")
```

**Parameters:**

- **phid**
  - An attached Phidget Interface Kit handle.
- **count**
  - The name of the variable to store the sensor count (in parentheses).

**phidgetsInterfaceKit_getSensorValue**

Gets a sensor value (0-1000).

```plaintext
get phidgetsInterfaceKit_getSensorValue(phid, index, "sensorValue")
```

**Parameters:**

- **phid**
  - An attached Phidget Interface Kit handle.
- **index**
  - The sensor index.
- **sensorValue**
  - The name of the variable to store the sensor value (in parentheses).

**phidgetsInterfaceKit_getSensorRawValue**

Gets a sensor raw value (12-bit).

```plaintext
get phidgetsInterfaceKit_getSensorRawValue(phid, index, "sensorRawValue")
```

**Parameters:**

- **phid**
  - An attached Phidget Interface Kit handle.
- **index**
  - The sensor index.
- **sensorRawValue**
  - The name of the variable to store the sensor raw value (in parentheses).
**phidgetsInterfaceKit_getSensorChangeTrigger**

Gets a sensor change trigger.

```
get phidgetsInterfaceKit_getSensorChangeTrigger(phid, index, "trigger")
```

**Parameters:**

- **phid**
  - An attached Phidget Interface Kit handle.
- **index**
  - The sensor index.
- **trigger**
  - The name of the variable to store the change trigger (in parentheses).

**phidgetsInterfaceKit_setSensorChangeTrigger**

Sets a sensor change trigger.

```
get phidgetsInterfaceKit_setSensorChangeTrigger(phid, index, trigger)
```

**Parameters:**

- **phid**
  - An attached Phidget Interface Kit handle.
- **index**
  - The sensor index.
- **trigger**
  - The change trigger.

**phidgetsInterfaceKit_getRatiometric**

Gets the ratiometric state for this board.

```
get phidgetsInterfaceKit_getRatiometric(phid, "ratiometric")
```

**Parameters:**

- **phid**
  - An attached Phidget Interface Kit handle.
- **ratiometric**
  - The name of the variable to store the ratiometric state (in parentheses). Possible values are 1(TRUE) or 0(FALSE).
**phidgetsInterfaceKit_setRatiometric**

Sets the ratiometric state for this board.

```
get phidgetsInterfaceKit_setRatiometric(phid, ratiometric)
```

**Parameters:**

- **phid**
  - An attached Phidget Interface Kit handle.

- **ratiometric**
  - The ratiometric state. Possible values are 1(TRUE) or 0(FALSE).

**phidgetsInterfaceKit_getDataRate**

Gets the Data Rate for an analog input.

```
get phidgetsInterfaceKit_getDataRate(phid, index, “milliseconds”)
```

**Parameters:**

- **phid**
  - An attached Phidget Interface Kit handle.

- **index**
  - The sensor index.

- **milliseconds**
  - The name of the variable to store the data rate in ms (in parentheses).

**phidgetsInterfaceKit_setDataRate**

Sets the data rate for an analog input.

```
get phidgetsInterfaceKit_setDataRate(phid, index, milliseconds)
```

**Parameters:**

- **phid**
  - An attached Phidget Interface Kit handle.

- **index**
  - The sensor index.

- **milliseconds**
  - Data rate in ms.
**phidgetsInterfaceKit_getDataRateMax**

Gets the maximum supported data rate for an analog input.

get phidgetsInterfaceKit_getDataRateMax(phid, index, “max”)

**Parameters:**

*phid*
- An attached Phidget Interface Kit handle.

*index*
- The sensor index.

*max*
- The name of the variable to store the data rate in ms (in parentheses).

**phidgetsInterfaceKit_getDataRateMin**

Gets the minimum supported data rate for an analog input.

get phidgetsInterfaceKit_getDataRateMin(phid, index, “min”)

**Parameters:**

*phid*
- An attached Phidget Interface Kit handle.

*index*
- The sensor index.

*min*
- The name of the variable to store the data rate in ms (in parentheses).
Events

**phidgetsInterfaceKit_set_OnInputChange_Handler**

Sets a digital input change handler. This is called when a digital input changes.

`get  phidgetsInterfaceKit_set_OnInputChange_Handler(phid, "commandName")`

**Parameters:**

**phid**

An attached Phidget Interface Kit handle.

**commandName**

The name of the command to be called (in parentheses). The command must be in the script of the card that called phidgetsInterfaceKit_set_OnInputChange_Handler or its stack. The form of the command is:

```
  command commandName phid pIndex pState
      ...
  end commandName
```

**phidgetsInterfaceKit_set_OnOutputChange_Handler**

Sets a digital input change handler. This is called when a digital input changes.

`get  phidgetsInterfaceKit_set_OnOutputChange_Handler(phid, "commandName")`

**Parameters:**

**phid**

An attached Phidget Interface Kit handle.

**commandName**

The name of the command to be called (in parentheses). The command must be in the script of the card that called phidgetsInterfaceKit_set_OnOutputChange_Handler or its stack. The form of the command is:

```
  command commandName phid pIndex pState
      ...
  end commandName
```

**phidgetsInterfaceKit_set_OnSensorChange_Handler**

Sets a digital input change handler. This is called when a digital input changes.

`get  phidgetsInterfaceKit_set_OnSensorChange_Handler(phid, "commandName")`

**Parameters:**

**phid**

An attached Phidget Interface Kit handle.

**commandName**

The name of the command to be called (in parentheses). The command must be in the script of the card that called phidgetsInterfaceKit_set_OnSensorChange_Handler or its stack. The form of the command is:

```
  command commandName phid pIndex pValue
      ...
  end commandName
```