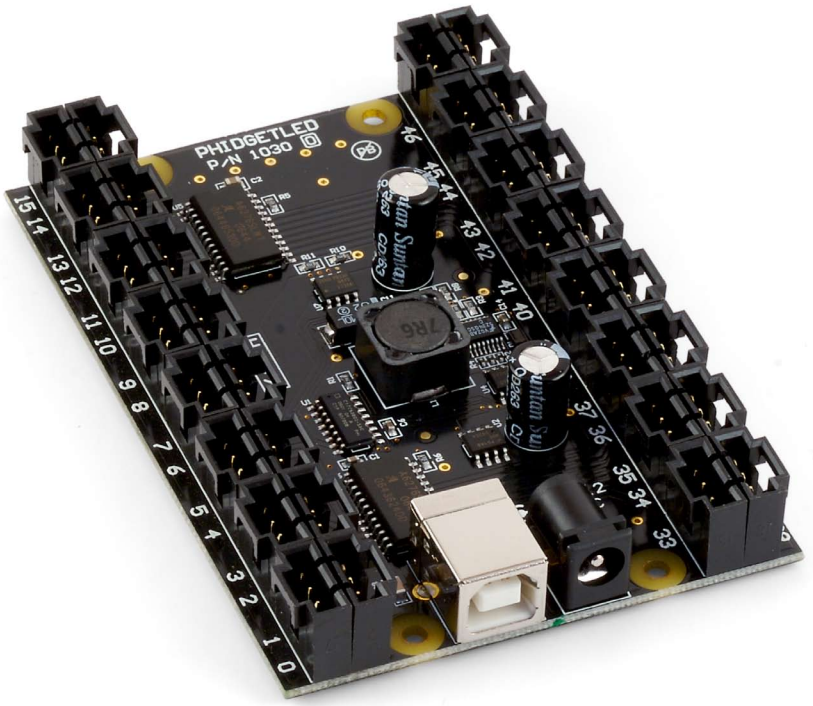
1030 - PhidgetLED 64



Product Features

- The PhidgetLED 64 allows you to independently control 64 Light Emitting Diodes.
- Each LED can be turned on and off, and its brightness controlled.
- Output Update Rate: 30 updates/second
- Output Anode Voltage: 3.3VDC
- Output Current (max): 30mA
- LED Forward Voltage (max): 3.0VDC
- Requires an external 6 to 12VDC power supply
- Connects directly to a computer's USB port

Programming Environment

Operating Systems: Windows 2000/XP/Vista, Windows CE, Linux, and Mac OS X

Programming Languages (APIs): VB6, VB.NET, C#.NET, C++, Flash 9, Flex, Java, LabVIEW, Python, Max/MSP, and Cocoa.

Examples: Many example applications for all the operating systems and development environments above are available for download at www.phidgets.com.

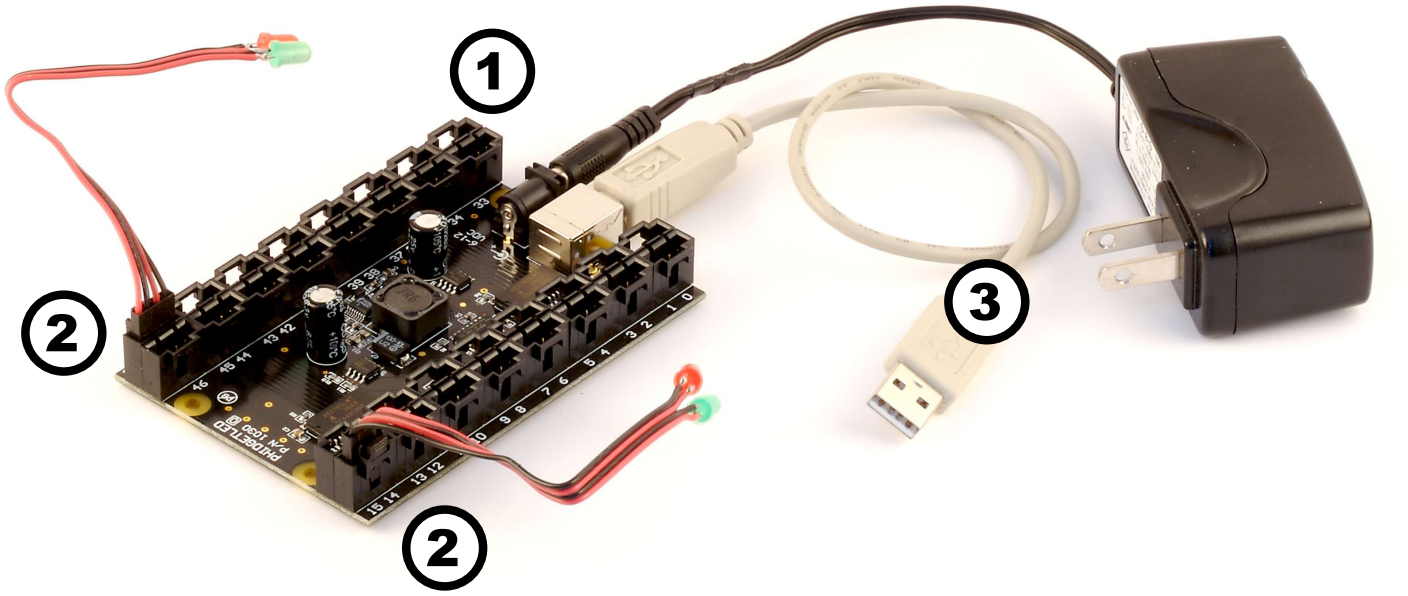
Installing the hardware

The kit contains:

- A PhidgetLED 64 board
- 16 Double ended 4-wire cables (for LEDs)
- A USB Cable

You will also need:

- Some LEDs
- A 6 to 12VDC power supply



1. Connect the power supply to the PhidgetLED using the barrel connector.
2. Cut one of the 4-wire cables to an appropriate length. Strip the wire ends and solder 2 LEDs to each cable. The longer lead of the LED (the anode) is connected to the red wire, and the shorter lead of the LED (the cathode, also marked with a notch in the base of the plastic case) is connected to the black wire. Insert the LED cable into the board connector.
3. Connect the PhidgetLED to your computer using the USB cable.

Downloading and Installing the software

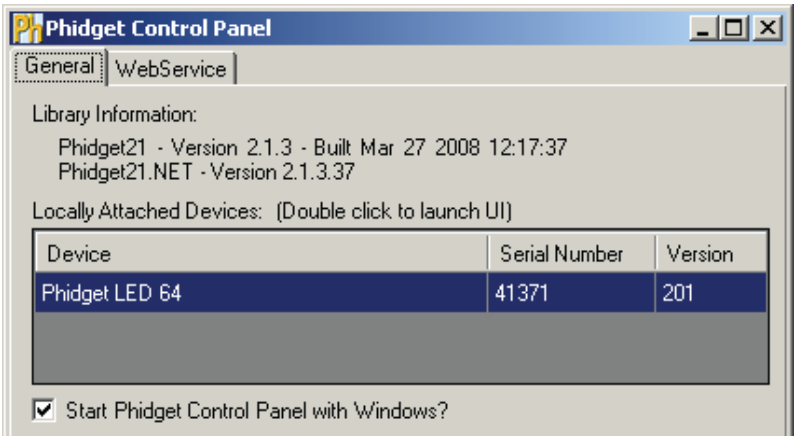
If you are using Windows 2000/XP/Vista

Go to www.phidgets.com >> Drivers

Download and run Phidget21.MSI

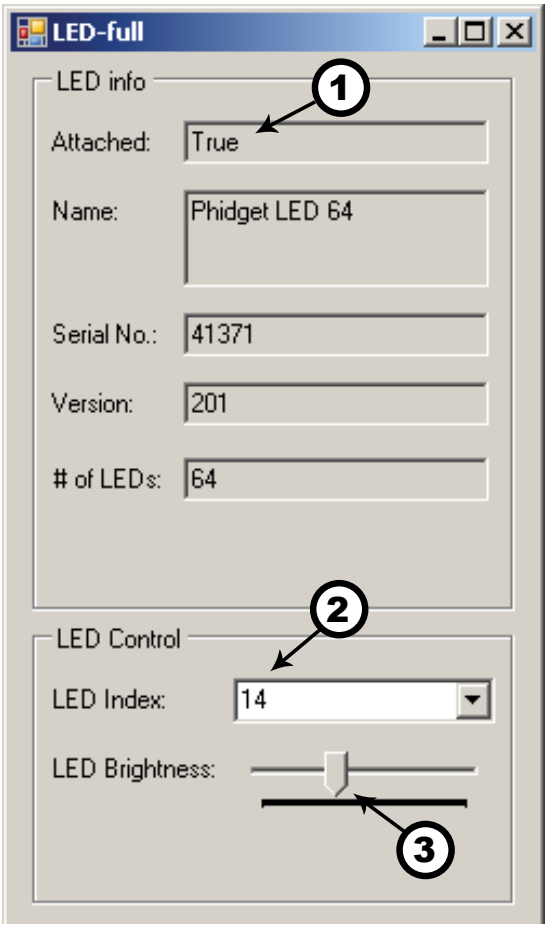
You should see the icon on the right hand corner of the Task Bar.

Testing the PhidgetLED 64 Functionality



Double Click on the icon to activate the Phidget Control Panel and make sure that the ***PhidgetLED 64*** is properly attached to your PC.

1. Double Click on ***PhidgetLED 64*** in the Phidget Control Panel to bring up LED-full and check that the box labelled Attached contains the word True.
2. Select the LED Index to turn on the LED. If you connected your LEDs at the same position as the ones in the picture on the previous page, select 14, 15, 48 or 49.
3. Use the LED Brightness slider to increase or decrease the LED brightness.



If you are using Mac OS X

Go to www.phidgets.com >> Drivers

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

If you are using Linux

Go to www.phidgets.com >> Drivers

Download Linux Source

- Have a look at the readme file
- Build Phidget21

The most popular programming languages in Linux are C/C++ and Java.

Notes:

Many Linux systems are now built with unsupported third party drivers. It may be necessary to uninstall these drivers for our libraries to work properly.

Phidget21 for Linux is a user-space library. Applications typically have to be run as root, or udev/hotplug must be configured to give permissions when the Phidget is plugged in.

If you are using Windows Mobile/CE 5.0 or 6.0

Go to www.phidgets.com >> Drivers

Download x86 or ARMV4I, depending on the platform you are using. Mini-itx and ICOP systems will be x86, and most mobile devices, including XScale based systems will run the ARMV4I.

The CE libraries are distributed in .CAB format. Windows Mobile/CE is able to directly install .CAB files.

The most popular languages are C/C++, .NET Compact Framework (VB.NET and C#). A desktop version of Visual Studio can usually be configured to target your Windows Mobile Platform, whether you are compiling to machine code or the .NET Compact Framework.

Programming a Phidget

Phidgets' philosophy is that you do not have to be an electrical engineer in order to do projects that use devices like sensors, motors, motor controllers, and interface boards. All you need to know is how to program. We have developed a complete set of Application Programming Interfaces (API) that are supported for Windows, Mac OS X, and Linux. When it comes to languages, we support VB6, VB.NET, C#.NET, C, C++, Flash 9, Flex, Java, LabVIEW, Python, Max/MSP, and Cocoa.

Architecture

We have designed our libraries to give you the maximum amount of freedom. We do not impose our own programming model on you.

To achieve this goal we have implemented the libraries as a series of layers with the C API at the core surrounded by other language wrappers.

Libraries

The lowest level library is the C API. The C API can be programmed against on Windows, CE, OS X and Linux. With the C API, C/C++, you can write cross-platform code. For systems with minimal resources (small computers), the C API may be the only choice.

The Java API is built into the C API Library. Java, by default is cross-platform - but your particular platform may not support it (CE).

The .NET API also relies on the C API. Our default .NET API is for .NET 2.0 Framework, but we also have .NET libraries for .NET 1.1 and .NET Compact Framework (CE).

The COM API relies on the C API. The COM API is programmed against when coding in VB6, VBScript, Excel (VBA), Delphi and Labview.

The ActionScript 3.0 Library relies on a communication link with a PhidgetWebService (see below). ActionScript 3.0 is used in Flex and Flash 9.

Programming Hints

- Every Phidget has a unique serial number - this allows you to sort out which device is which at runtime. Unlike USB devices which model themselves as a COM port, you don't have to worry about where in the USB bus you plug your Phidget in. If you have more than one Phidget, even of the same type, their serial numbers enable you to sort them out at runtime.
- Each Phidget you have plugged in is controlled from your application using an object/handle specific to that phidget. This link between the Phidget and the software object is created when you call the .OPEN group of commands. This association will stay, even if the Phidget is disconnected/reattached, until .CLOSE is called.
- The Phidget APIs are designed to be used in an event-driven architecture. While it is possible to poll them, we don't recommend it. Please familiarize yourself with event programming.

Networking Phidgets

The PhidgetWebService is an application written by Phidgets Inc. which acts as a network proxy on a computer. The PhidgetWebService will allow other computers on the network to communicate with the Phidgets connected to that computer. ALL of our APIs have the

capability to communicate with Phidgets on another computer that has the PhidgetWebService running.

The PhidgetWebService also makes it possible to communicate with other applications that you wrote and that are connected to the PhidgetWebService, through the PhidgetDictionary object.

Documentation

Programming Manual

The Phidget Programming Manual documents the Phidgets software programming model in a language and device unspecific way, providing a general overview of the Phidgets API as a whole.

Getting Started Guides

We have written Getting Started Guides for most of the languages that we support. If the manual exists for the language you want to use, this is the first manual you want to read. The Guides can be found under Programming and are listed under the appropriate language.

API documentation

We maintain API references for COM (Windows), C (Windows/Mac OSX/Linux), Action Script, .Net and Java. These references document the API calls that are common to all Phidgets. These API References can be found under Programming and are listed under the appropriate language. To look at the API calls for a specific Phidget, check its Product Manual.

Code Samples

We have written sample programs to illustrate how the APIs are used.

Due to the large number of languages and devices we support, we cannot provide examples in every language for every Phidget. Some of the examples are very minimal, and other examples will have a full-featured GUI allowing all the functionality of the device to be explored. Most developers start by modifying existing examples until they have an understanding of the architecture.

Go to Programming to see if there are code samples written for your device. Find the language you want to use and click on the magnifying glass besides “Code Sample”. You will get a list of all the devices for which we wrote code samples in that language.

Support

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

Technical Section

LEDs

Like normal diodes, Light Emitting Diodes (LEDs) are semiconductor devices designed to conduct current in one direction only. What makes LEDs unique is their internal material makeup: when atoms in an LED release energy due to the flow of forward current, it is released in the form of photons (light). Different construction materials and various phosphor coatings are used to produce numerous colors of light.

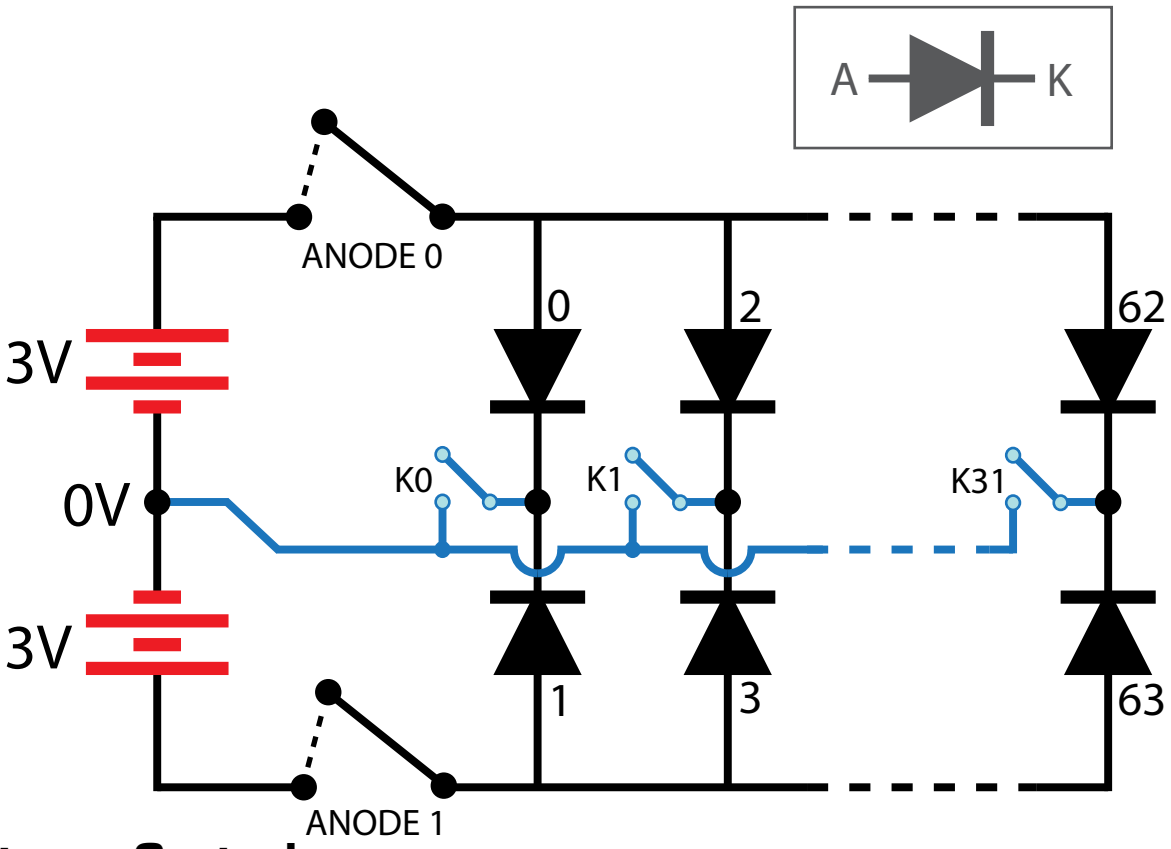
Forward Voltage

The materials used within LEDs that cause them to emit different colors of light affect a property called its forward voltage. The forward voltage is the voltage at which current will flow through the device and allow the LED to convert electrical energy into light. If the voltage applied to the LED is below the forward voltage of the LED, very little current (or none) may flow, and therefore very little light will be emitted.

Most standard LEDs with colors such as red, amber, orange, yellow, and green have forward voltages below 3.0 Volts, and can be used with the PhidgetLED by simply soldering them to a connector-wire and inserting the wire into any PhidgetLED board connector. Some green, blue, and white colored LEDs, as well as some ultra-bright LEDs, may have a forward voltage significantly higher than 3.0 Volts. These will work with the PhidgetLED, but may appear dimmer as less current will be flowing through the LED. The maximum current available below 3.0V is 30mA.

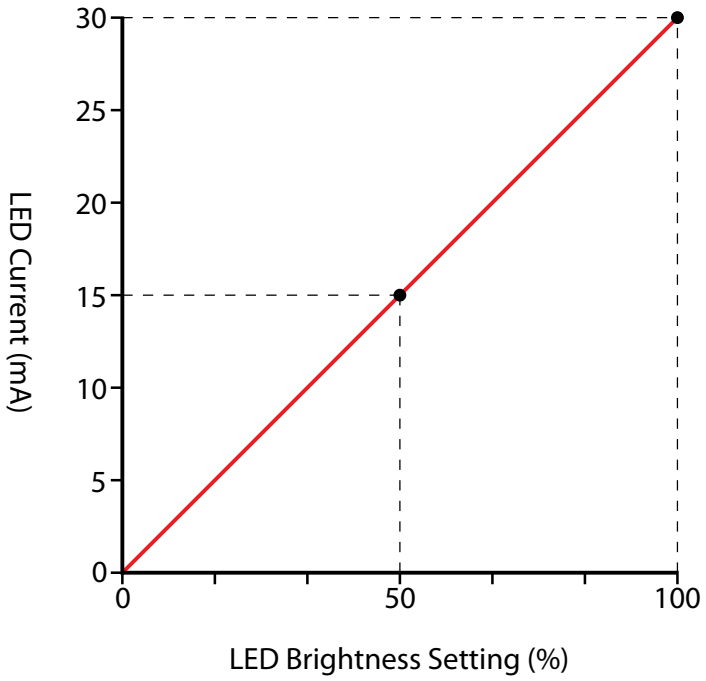
Multiplexed LEDs

The PhidgetLED 64 operates as a 2 x 32 multiplexer with 2 anodes and 32 cathodes. The method in which LEDs are connected to the PhidgetLED is illustrated above. The PhidgetLED operates as a constant-current controller, providing only as much voltage as is necessary for the LED to draw 30 mA. Adding a series resistor to protect LEDs from overcurrent conditions is not necessary. LEDs with built-in series resistors will work, but will appear dimmer as power is dropped across the resistor. If the two anodes are joined together at any point, the PhidgetLED is reduced to a 32 LED controller providing 60mA of current per LED.



Brightness Control

The brightness of each LED is controlled by Pulse Width Modulating the current to each LED. The source current is adjusted linearly between 0 and 100% up to 30mA. This however does not imply a precise method for controlling the visibility of emitted light, as this is affected by the construction and quality of the LED as well as the eyes of the viewer. If both anodes of the PhidgetLED are connected together to create a 60mA controller, the brightness property of both connector LED ports must be set in order to control the current flow (for example, if an LED is plugged into port 0, both ports 0 and 1 must be set in software).



Indicator Types

Different types of indicators will have to be connected to the PhidgetLED in different ways. Tri-color LEDs, seven-segment displays, bar LEDs, and other array-style arrangements of LEDs should be purchased as common-anode devices. As all the Anodes on the 1031 are wired together, effort can be saved by only connecting to one Anode. As there is no LED multiplexing on the 1031, it is impossible to use common-cathode LED devices.

Indicators

The PhidgetLED 64 can be used with several different styles of indicators, including standard LEDs, multi-colored LEDs, seven-segment displays, and bar/array style LEDs. Several examples are shown below.

Manufacturer	Part Number	Description
LiteON	LTL-307E	5mm Round High-Efficiency Red LED
LiteON	LTL-307G	5mm Round Green LED
Lumex	SSL-LX5099SI	5mm Round Tri-Color RGB LED
Lumex	LTS-4801JR	0.4" Red Common-Anode 7-Segment Display

NOTE: most of the indicators listed, and many others, are available from www.digikey.com

Cable and Connector Components for LED Outputs

Manufacturer	Part Number	Description
Molex	50-57-9404	4 Position Cable Connector
Molex	16-02-0102	Wire Crimp Insert for Cable Connector
Molex	70543-0003	4 Position Vertical PCB Connector
Molex	70553-0003	4 Position Right-Angle PCB Connector (Gold)
Molex	70553-0038	4 Position Right-Angle PCB Connector (Tin)
Molex	15-91-2045	4 Position Right Angle PCB Connector - Surface Mount

Note: Most of the above components can be purchased at www.digikey.com

Expected Uses

The PhidgetLED 64 can only be used with Light Emitting Diodes. It can not be used to control Motors or Transistors and the outputs can not be used as standard digital I/O. The LED brightness is controlled using Pulse Width Modulation and the anode and cathode are Multiplexed (see page 8). The outputs are not a standard positive voltage and ground pin. Some opto-couplers can be used with the outputs, but your design must use the PWM and multiplexing correctly.

API (Software Technical)

We document API Calls specific to this product in this section. Functions common to all Phidgets and functions not applicable to this device are not covered here. This section is deliberately generic. For calling conventions under a specific language, refer to the associated API manual. For exact values, please refer to the device specifications.

Functions

int LEDCount() [get] : Constant

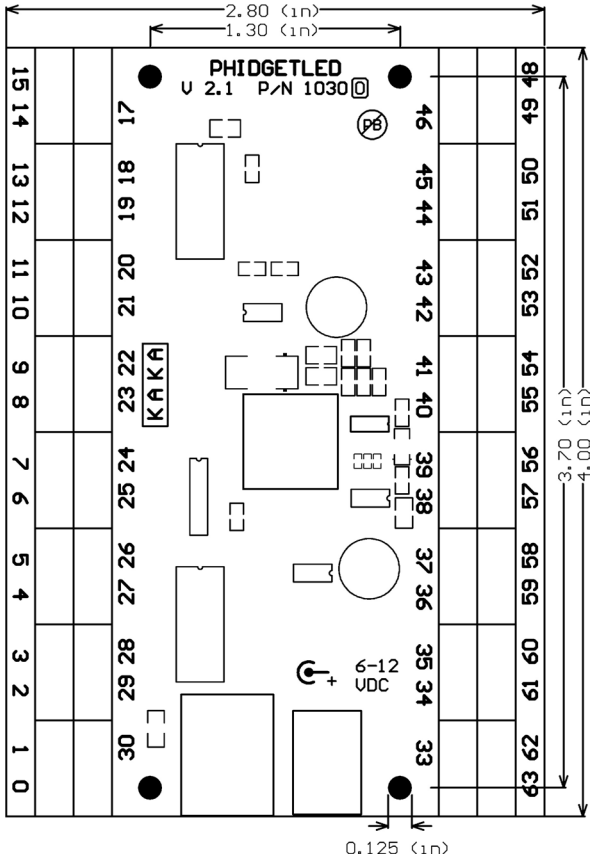
Returns the number of LEDs that this board can drive. This may not correspond to the actual number of LEDs attached.

int DiscreteLED(int LEDIndex) [get,set] : Constant

Sets the brightness of an LED. Valid values are 0-100, with 0 being off and 100 being the brightest. This 0-100 value is converted internally to a 6-bit value (0-63) so only 64 levels of brightness are actually possible.

Mechanical Drawing

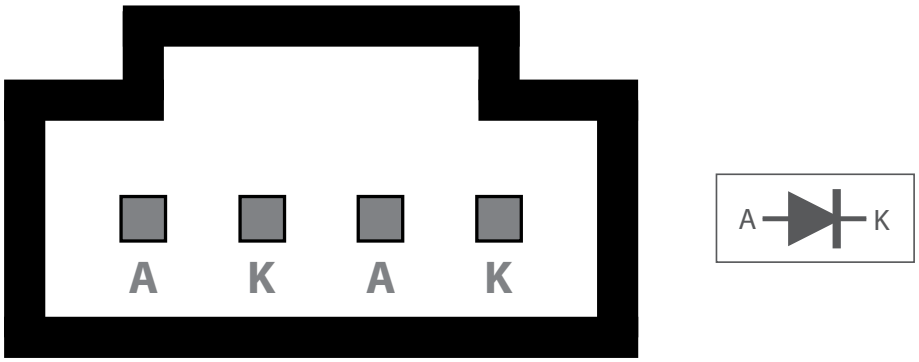
1:1 scale



Note: When printing the mechanical drawing, "**Page Scaling**" in the Print panel must be set to "**None**" to avoid re-sizing the image.

Board Connector Drawing

Not to Scale



Device Specifications

Output Update Rate	30 updates / second
Output Anode Voltage	3.3VDC
Output Current (max) (V_F <= 3.0V)	30mA
LED Forward Voltage (max)	3.0VDC
USB Power Current Specification	500mA max
Device Current Consumption	31mA
External Power Supply Voltage (min)	6VDC
External Power Supply Voltage (max)	12VDC
Externally Supplied Current (max)	1.5A

Product History

Date	Product Revision	Comment
January 2003	n/a	Product Release, solder to hole mounting
January 2005	n/a	Migrated to 4-pin dual LED connector