

# Getting started with Phidgets in MATLAB

## Environment and Libraries

Phidgets supports development in MATLAB for all types of Phidgets using calls to the C library. First, we need to set up the proper environment and get the necessary files off the Phidgets website. Visit the drivers and programming section at [www.phidgets.com](http://www.phidgets.com) and get the latest:

- Phidget Framework
- MATLAB examples

You will need the `phidget21Matlab` library from the MATLAB examples to program with Phidgets, and the Phidget Framework to use them. We also recommend that you download the following reference materials from the programming section:

- C API Manual
- Programming Manual
- The Product Manual for your device

The C API manual contains calls and events for every type of Phidget and can be used as a reference. You can find a high level discussion about programming with Phidgets in general in the Programming Manual. The Product manual for your device also contains an API section that describes limitations, defaults, and implementation details specific to your Phidget. You may want to have these manuals open while working through these instructions.

## Setting up a Phidgets Project

The Phidget examples were written using MATLAB `m`-files and this tutorial assumes its use. Your project will need to be able to find the `phidget21Matlab.h` from the MATLAB examples.

## Coding For Your Phidget

Programming with Phidgets will make extensive use of the `calllib()` function. Before you can use the Phidget, you must include a reference to the library in the code in the main body of code. Copy `phidget21Matlab.h` to your project directory and then in MATLAB:

```
function phidgettest(n)
    loadlibrary phidget21 phidget21Matlab.h;
    % More code goes here
end
```

Afterwards, the Phidget object will need to be declared and then initialized. For example, we can declare a `PhidgetInterfaceKit` with:

```
ikptr = libpointer('int32Ptr',0);
calllib('phidget21', 'CPhidgetInterfaceKit_create', ikptr);
ikhandle = get(ikptr, 'Value');
```

The ikptr is converted to ikhandle as a handle of the PhidgetInterfaceKit, and is used for all the C function calls where CPhidgetHandle phid is used.

The object name for any type of Phidget is listed in the API manual. Every type of Phidget also inherits functionality from the Phidget base class.

## Connecting to the Phidget

The program can try to connect to the Phidget through an open call. Open will continuously try to connect to a Phidget, based on the parameters given, even trying to reconnect if it gets disconnected. This means that simply calling open does not guarantee you can use the Phidget immediately. We can handle this by calling waitForAttachment. WaitForAttachment will block indefinitely until a connection is made to the Phidget, or an optional timeout is exceeded.

```
calllib('phidget21', 'CPhidget_open', ikhandle, -1);
if calllib('phidget21', 'CPhidget_waitForAttachment', ikhandle, 2500) == 0
    % Insert your code here
end
```

The different types of open can be used with parameters to try and get the first device it can find, open based on its serial number, or even open across the network. The API manual lists all of the available modes that open provides. One important thing to remember is that when working with Phidgets, a local connection will reserve the device until closed. This prevents any other instances from retrieving data from the Phidget, including other programs. The one connection per device limit does not apply when exclusively using the Phidget Webservice.

At the end of your program, don't forget to call close to free any locks on the Phidget.

```
calllib('phidget21', 'CPhidget_close', ikhandle);
calllib('phidget21', 'CPhidget_delete', ikhandle);
```

## Working directly with the Phidget

MATLAB does not support event handling, so all data must be read and sent directly. Simply use the C API functions such as CPhidgetInterfaceKit\_getSensorValue() or CPhidgetInterfaceKit\_setOutState() for InterfaceKits.

```
while n<10
    dataptr = libpointer('int32Ptr',0);
    calllib('phidget21', 'CPhidgetInterfaceKit_getSensorValue', ikhandle, 0, dataptr)
    disp(get(dataptr, 'Value'));
    n=n+1;
end
```

Here a pointer is created to mark a value from the InterfaceKit inside a polling loop, and then its value is displayed to screen.

## **Working with multiple Phidgets**

Multiple Phidgets of the same type can easily be run inside the same program. In our case, it requires another pointer and handle initialized for a PhidgetInterfaceKit. The new instance can then be set up, opened and used in the same fashion as the previous one.

If the application needs to distinguish between the devices, open can be called with the serial number of a specific Phidget.

## **Other Phidgets**

The design given in this document can also be followed for almost all Phidgets. For example, if you were using a PhidgetRFID instead of an Interfacekit, you would call CPhidgetRFID\_create instead of CPhidgetInterfaceKit\_create. The functions and events available would change but they can be accessed in a similar manner.