

# PhidgetServo 1-Motor

Phidgets are the most user-friendly system available for controlling and sensing the environment from your computer. People with absolutely no hardware knowledge or experience can include servo motors into their projects. It is just a matter of plugging your device into the board, which in turn is plugged into the USB port on your computer. After that, you can use the simple to program Phidgets software libraries to access this device.



The PhidgetServo boards are our most popular Phidgets; we sell lots of these. They can control most Remote Control (RC) servo motors, with an accuracy of 0.1 degrees, directly from your PC.

The PhidgetServo 1-Motor is a USB based controller that can control one RC servo motor; the servo motor is powered from the USB port. The PhidgetServo 1-Motor board can be controlled from Windows, Linux, and Mac OS X. High-level programming interfaces are available for Visual Basic, C, C++, Flash, .NET, Java, LabVIEW, etc.

## What Can the PhidgetServo 1-Motor Do?

RC servo motors have the useful property that they can be positioned absolutely, and they have a lot of torque. Generally, they are used to actuate / push / pull, not like a traditional motor that just spins. What can you do with a servo motor? What would you like to do? PhidgetServo is our most popular product, and nearly every project uses servo motors.

# Getting Started on Windows 2000 / XP

The PhidgetServo 1-Motor board can be controlled from a variety of Software Development Environments. In Visual Basic the PhidgetServo software component provide a high-level programmer interface. Here is how to get started under Windows 2000 or Windows XP.

## Step 1. What you need to have ready

- Your PhidgetServo 1-Motor.
- A standard Remote Control (RC) servo motor.
- A USB cable.
- A computer running Windows 2000 or Windows XP.
- An Internet connection and a Web browser.

## Step 2. Assemble your hardware

- Attach the connector from a servo motor onto the PhidgetServo board. The board is labeled with B R W (Black Red White) to match the wire colors from servo motors. If you connect it backwards, it will not work! Some servo motors have a yellow wire instead of a white wire.
- Plug the square end of the USB cable into the USB input connector on the Interface Kit.
- Plug the rectangular end of the USB cable into the USB connector on your computer (you can do this at any time).

## Step 3. Install the software

Phidgets use a library installed on your computer. This only has to be installed once no matter how many different types of Phidgets you have.

- Have you previously installed the Phidget library? If so, you can skip this step.
- To install the library go to [www.phidgets.com](http://www.phidgets.com) >> Downloads >> Release.
- Select the PHIDGET.msi file.
- A dialog box will appear asking if you would like to open the file or save it to your computer. You can do either, but if you are unsure just select "Open" and follow the instructions.
- Do you want to update a previously installed Phidget library? If so, you must remove the old library when prompted to do so.

## Step 4. Download the sample programs

- Go to [www.phidgets.com](http://www.phidgets.com) >> Downloads >> Release.
- Select the Examples.zip file.
- Save the zip file to a place of your choosing, and then uncompress it.
- You will find many executables in the Visual Basic folder, and their source in the sub-folders.
- PhidgetMonitor.exe will show you what Phidgets are plugged in. If you have followed Step 2 and Step 3, a PhidgetServo should be displayed in the list, along with its serial number. The source for this .exe is in the PhidgetManagerExamples folder.
- All PhidgetServo samples have a "servo" prefix. Try them! All source code can be found in the PhidgetServoExamples folder.

## Step 5. Try Programming a Phidget

- Go to [www.phidgets.com](http://www.phidgets.com) >> Documentation.
- Read the documentation for the PhidgetManager, the IPhidget, and read the documentation under the PhidgetServo heading.
- Based on this documentation, we recommend you examine the source of the sample programs mentioned in Step 4.
- Now try modifying the samples. Or code your own from scratch.

## Step 6. Learning more ...

- Explore [www.phidgets.com](http://www.phidgets.com). We recommend you visit Projects and Examples to see what other people have done. For new applications or other programming languages visit the Forums.

## Step 7. Read the license agreement

Go to [www.phidgets.com](http://www.phidgets.com) >> Documentation and select the License link. Note that the term Phidget™ is trademarked, and that the name PhidgetServo is synonymous with Phidget™.

## Angles and Contiguous Rotation

The MotorPosition software component in the programming interface will accept a range of values from -23 to 231, corresponding to the time width of the control pulse. MotorPosition will return an error if the value is not between -23 and 231. The angle that the servo motor moves to depends on the characteristic of the individual motor. As an example, the Hitec HS-322HD operates between 20 and 220, corresponding to a rotation of 200 degrees. When a value of 0 or 225 is sent, the motor cannot move beyond its physical limit and this may cause "chattering" and excessive wear. If you use a different servo motor then these numbers will be different.

Another important point is that motor position requests are not cumulative. That is, let us say the current actual motor position is 0 and you do something like the following:

- `Servo.MotorPosition(1) = 180`
- `Servo.MotorPostion(1) = 0`

The motor will not rotate all the way to 180 before returning back to 0. Rather, it will likely just begin rotating based on the first call, but will immediately return to 0 from its current position when it gets the second call. All you would see (perhaps) is a very slight twitch. If you want a cascading effect, you will have to use a timer to guarantee that the previous rotation is completed.

## Pulse Width Modulation (PWM)

The position of a servo motors is controlled by varying the length of a + 5 V pulse on the signal line (white or yellow wire). This is referred to Pulse Width Modulation (PWM), or sometimes Pulse Code Modulation (PCM). The width of the pulse translates to a position in the rotation of the motor. For example, a pulse of 1.5 milliseconds is generally the center position. The other extremes of the timing allowed depend on the manufacturer and model of the servo motor. Refer to the manufacturer's datasheet for a particular motor to determine what angle corresponds to a specific pulse width. The timing of the pulse from the PhidgetServo controller can be calculated as follows:

$$\text{millisecond Pulse} = (\text{Value} * 10.6) + 243.8$$

where Value is used in the MotorPosition software component in the programming interface.

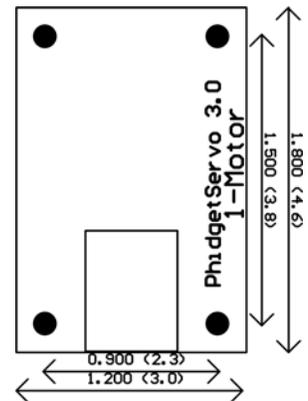
# Hardware Description

The PhidgetServo 1-Motor controls one standard hobby servo motors. It is designed to work with various types of servo motor. The ones we use are relatively inexpensive hobbyist RC servo motors that are commonly available. These include:

- Futaba FP-S-148 Servo (available from HVWTEch and Tower Hobbies).
- Tower Hobbies TS-53 (available from Tower Hobbies).

Other hobbyist servo motors are likely to work as long as they take 5 V and use a Pulse Width Modulation (PWM) scheme, but care must be taken that the motor does not operate past its boundaries and overheat.

The labeling on the PhidgetServo board is B – Ground, R – Power (+ 5 V), and W – Signal. The signal wire from the servo motor can be white or yellow depending on the manufacturer.



## Device Specification

Position Update Rate	Approx. 30 Hz
USB Current Consumption	Max. 500 mA