

Application Note: Using AutoHotkey with Pololu USB Products



Automation. Hotkeys. Scripting.

1. Overview	2
2. Products with Command Line Programs	3
3. Installation	4
4. Using Autohotkey	5
5. Example AutoHotkey Scripts	6
5.a. Maestro Scripts	6
5.a.1. Toggle an LED	6
5.a.2. Start a Sequence	6
5.a.3. Move a Servo	6
5.b. Simple Motor Controller Scripts	6
5.b.1. Control a Motor	6
5.c. Jrk Scripts	6
5.c.1. Control a motor	7
6. Customer Projects	8

1. Overview

Did you ever want to control a motor or servo from your computer simply by pressing a key? You can do this with a few easy lines of code by using **AutoHotkey** [<http://www.autohotkey.com/>] to run the command-line utilities that come with some of our USB products. AutoHotkey is a flexible, free, open-source utility for Windows that runs in the background. It can be set up to detect when a keyboard-combination is pressed and respond by running a custom script. Since these custom scripts can run Windows command-line programs, they can run the command-line utilities included with some of Pololu's USB products. There are many possible applications of using AutoHotkey with our USB products, including home automation, computer case modifications, and computer-controlled art/displays. See **Section 6** for examples of customer projects using AutoHotkey and Pololu products.

2. Products with Command Line Programs

The following Pololu USB products have command-line utilities:

- the **Maestro Servo Controllers** [<http://www.pololu.com/maestro>] have *usccmd*,
- the **Simple Motor Controllers** [<http://www.pololu.com/smc>] have *smccmd*, and
- the **Jrk Motor Controllers with Feedback** [<http://www.pololu.com/catalog/category/95>] have *jrkcmm*.

3. Installation

1. **Download** [<http://www.autohotkey.com/download/>] and install AutoHotkey_L.
2. Install the latest version of your Pololu USB product's command-line utility. The command-line utility is usually packaged with the product's software and drivers and can be found on the Resources tab of the product page.



The *jrk's jrkcnd* is only available by compiling it with the **Pololu USB Software Development Kit** [<http://www.pololu.com/docs/0J41>].

4. Using Autohotkey

AutoHotkey has a brief **tutorial and overview** [<http://www.autohotkey.com/docs/Tutorial.htm>], which explains the basics of using the program.

When using AutoHotkey with these command-line utilities there are a couple of issues to look out for:

- The command-line utility will not run while the product software is connected to the USB device (e.g. *usccmd* cannot run while the Maestro Control Center is connected to a servo controller).
- Only one instance of the command-line utility can be running at a time. If you try to run two commands in a row without using AutoHotkey's RunWait, the command-line utility will not work.
- Commas(,) in command-line utility commands need to be escaped with backtics (\).

5. Example AutoHotkey Scripts

5.a. Maestro Scripts

5.a.1. Toggle an LED

In this example, an LED is connected to channel 0. See **Section 7.b** of the Maestro User's Guide to see how to connect an LED. Ctrl+Alt+C will turn the LED on, and Ctrl+Alt+B will turn the LED off.

```
^!c:: Run usccmd --servo 0`,8000,,Hide
^!b:: Run usccmd --servo 0`,0,,Hide
```

5.a.2. Start a Sequence

After making a sequence as described in **Section 4.c** of the Maestro User's Guide, use the *Copy all Sequences to Script* button to save the sequences on the Maestro. You then can call the sequence by figuring out which subroutine on the Script tab corresponds to your sequence. *UscCmd* needs to be given the number (NUM) of the subroutine you want to call. To figure out the NUM of your subroutine, go to the *Script* tab and count the number of subroutines that appear before the subroutine you want to call, the number you count is the subroutine you want to call's number (i.e. the first subroutine's NUM is 0, the second is 1 and so on). Typing Ctrl+Alt+U runs the subroutine whose NUM is 0.

```
^!u:: Run usccmd --sub 0,,Hide
```

5.a.3. Move a Servo

In the script below, Ctrl+Alt+C sets the servo's target to near the end of its range. Ctrl+Alt+B sets the servo's target to near the other end of its range.

```
^!c::Run usccmd --servo 0`,7500,,Hide
^!b:: Run usccmd --servo 0`,4500,,Hide
```

In the script below, Ctrl+Alt+C will set the speed and acceleration of the servo on channel 0, then set its target to near the end of its range. Ctrl+Alt+B will remove the speed and acceleration limits, then set the servo's target to near the other end of its range.

```
^!c::
RunWait usccmd --accel 0`,12,,Hide
RunWait usccmd --speed 0`,12,,Hide
Run usccmd --servo 0`,7500,,Hide
return

^!b::
RunWait usccmd --accel 0`,0,,Hide
RunWait usccmd --speed 0`,0,,Hide
Run usccmd --servo 0`,4500,,Hide
return
```

5.b. Simple Motor Controller Scripts

5.b.1. Control a Motor

Ctrl+Alt+C stops the motor, Ctrl+Alt+V sets the motor to go in the forward direction, Ctrl+Alt+B sets the motor to go in the reverse direction,

```
^!c:: Run smccmd --brake 32,,Hide
^!v:: Run smccmd --resume --speed 1500,,Hide
^!b:: Run smccmd --resume --speed -1500,,Hide
```

5.c. Jrk Scripts

5.c.1. Control a motor

Ctrl+Alt+C stops the motor, Ctrl+Alt+V sets the motor to go in the forward direction, Ctrl+Alt+B sets the current target position to 1500,

```
^!c:: Run jrkcmd --stop,,Hide
^!v:: Run jrkcmd --run,,Hide
^!b:: Run jrkcmd --target 1500,,Hide
```

6. Customer Projects

- A **custom, media-center computer case** [<http://forum.pololu.com/viewtopic.php?f=16&t=3567>] built inside a **Musikschrank 7025** [http://www.radiomuseum.org/r/grundig_musikschrank_7025.html] radio.

If you make a project using AutoHotkey and a Pololu product, please **share it with us** [<http://www.pololu.com/contact>]!