Wireless Music Veto Button for your Office

Created by Paul Rickards

https://learn.adafruit.com/wireless-bluetooth-music-veto-button

Last updated on 2022-12-01 02:09:13 PM EST

Table of Contents

Overview and Parts List 3

- Parts List

Prepare the Box 4

- Cut the openings

Wiring and Assembly 9

- EZ-Key Headers
- Jumper Wires
- Charge Controller Jumpers
- Arcade Microswitch Jumpers
- Dry Test
- Assemble
- Charge the Battery

Configure the EZ-Key 14

- Pairing EZ-Key
- Output Key Remapping

Configure the Hot Key 14

- AppleScript
- Quicksilver
- What about Windows?

Use it! 18

- Improvements and Enhancements

Overview and Parts List

How many times have you suffered at the hand of your co-worker that plays Spotify radio in the office? Wouldn't it be great if you could simply veto their song that's playing and go on to the next?

This project creates a wireless veto button for music. The button is self-powered and can be recharged using a standard USB cable. Simply pair the Bluefruit EZ-Key to the computer playing the music, configure the software and you're all set to "cast your vote" and skip to the next song.

This project is specific for the Mac (because of AppleScript's ability to control apps) but could be translated to work with Windows if your music app can listen to media-control keypresses (like 'fast forward')

Parts List

For this project, you'll need

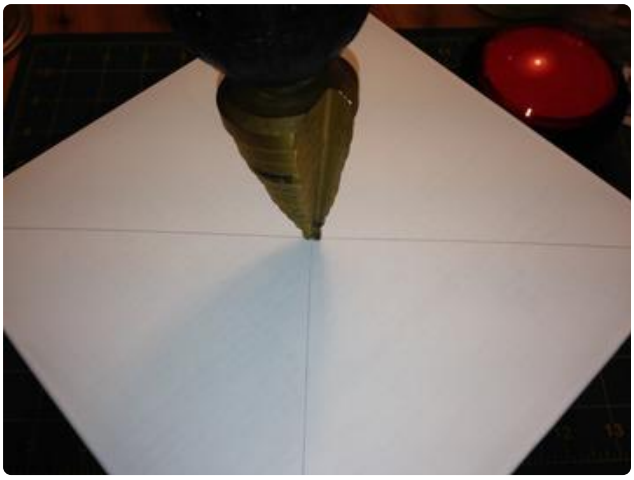
- Massive Arcade Button (http://adafru.it/1185)
- Adafruit Bluefruit EZ-Key (http://adafru.it/1535)
- Break-away right-angle male headers ()
- Jumper Wires ()
- USB Lithium Ion charger ()
- Lithium Ion Battery ()
- Panel mount USB cable (B female to Mini-B male) ()
- USB cable (standard A-B) (http://adafru.it/62)
- Suitable box or case (I used a sturdy white cardboard box)
- Double sided foam tape

If you want to skip the re-chargeable battery, you can simply use the 3xAAA battery case () (omit the battery, charger and USB cables).

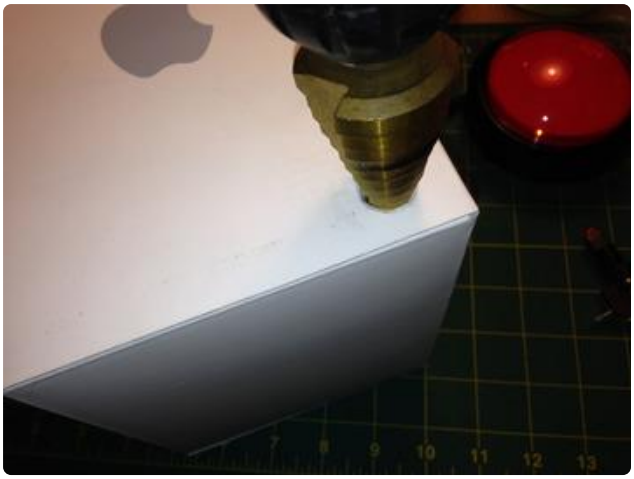
Prepare the Box

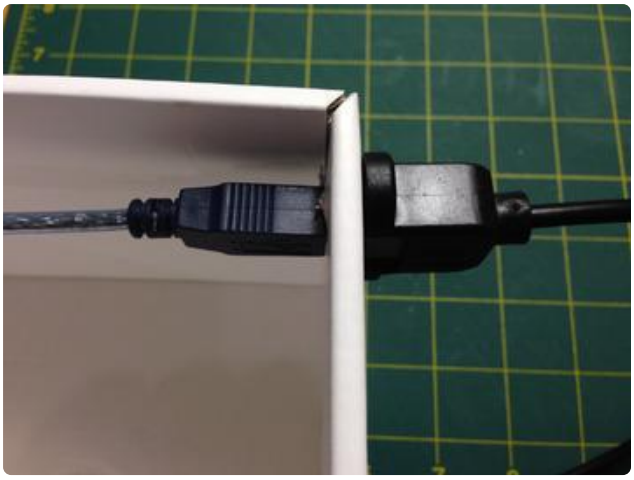
Cut the openings

You'll need to make a couple of openings in your box to accommodate the button and the USB charging port.



The massive arcade button needs a 1" circular hole cut. Measure to find the center line of your box. Using a step drill bit, drill down until you reach 1".

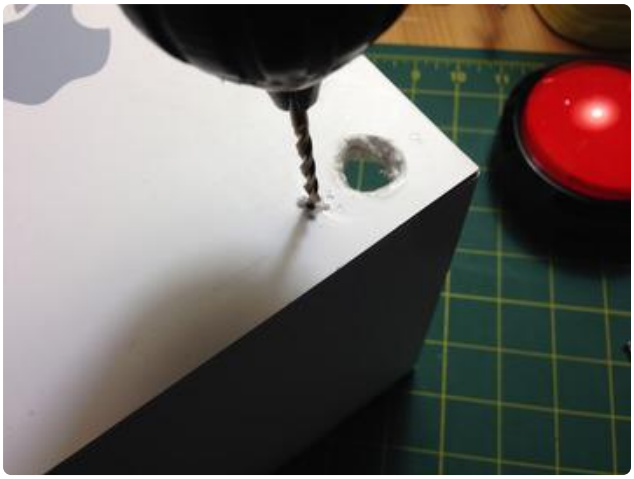


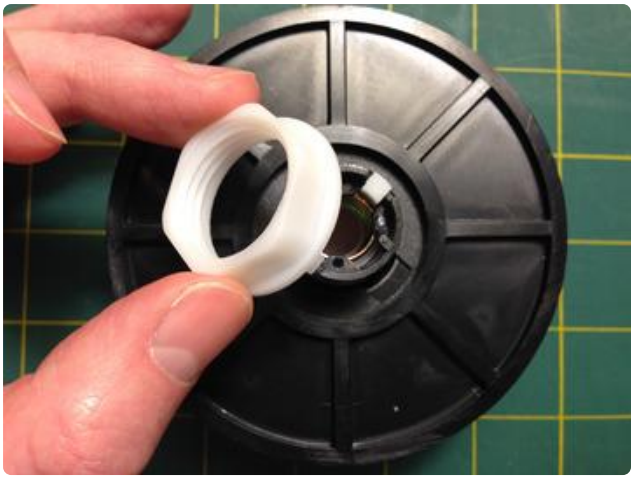


Now drill the hole for the USB port, locating it on the back, lower right of the box using a 1/2" circular hole. Next, drill the two holes for the mounting ears that hold the USB port in place. The mounting ear holes are ~28.5mm apart.

The easiest way is to place the port on the outside and plug in a USB cable from the inside to hold it in place. Use a pencil to mark the centers of the two screw holes. Drill both holes using a 1/8" drill bit.

Attach the USB port to the box using the two screws.

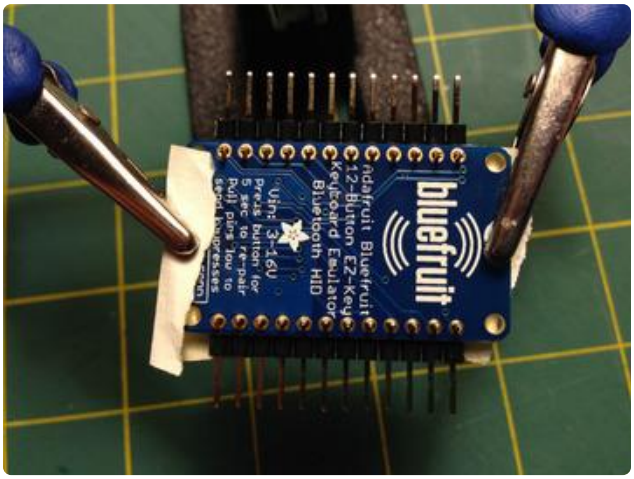


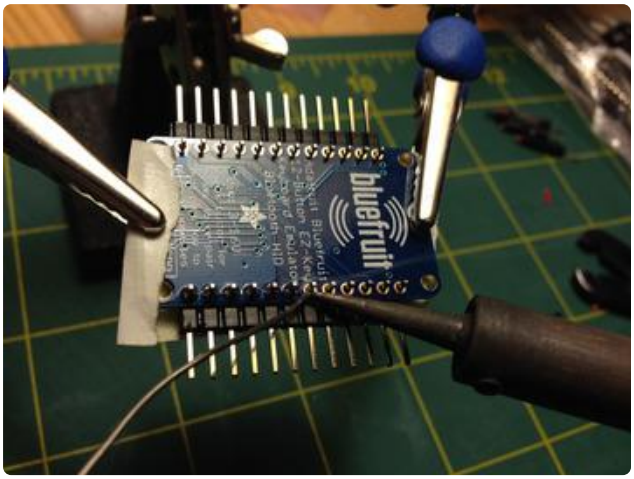


Next, install the massive arcade button on the top of the box. Remove the white plastic locking nut on the bottom of the button. Place the button through the hole and tighten the locking nut from below.

For now, leave the switch and LED assembly out to make it easier to work on.

Wiring and Assembly

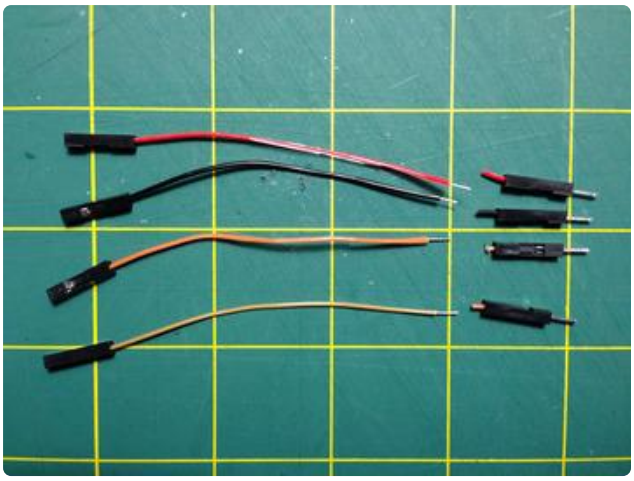




EZ-Key Headers

Count two sets of 12 right angle headers and break them away. Insert the headers into the EZ-Key from the top as shown and secure them with a piece of masking tape. Secure the board into a holder and begin soldering all 24 connections.

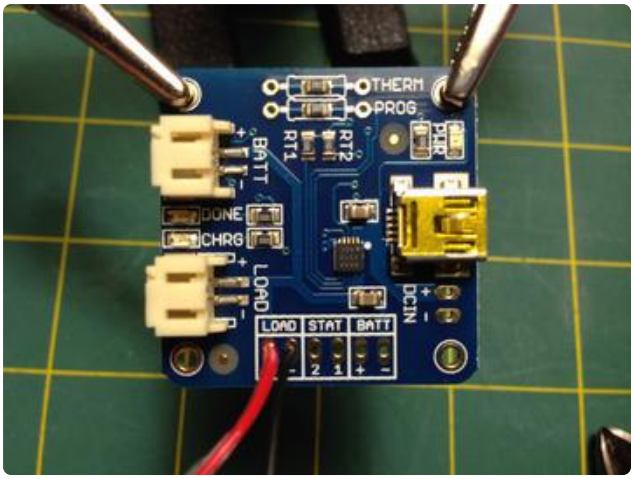
I prefer to add headers to the EZ-Key because it makes it very easy later to disassemble the unit should I want to make changes or use it in another project.

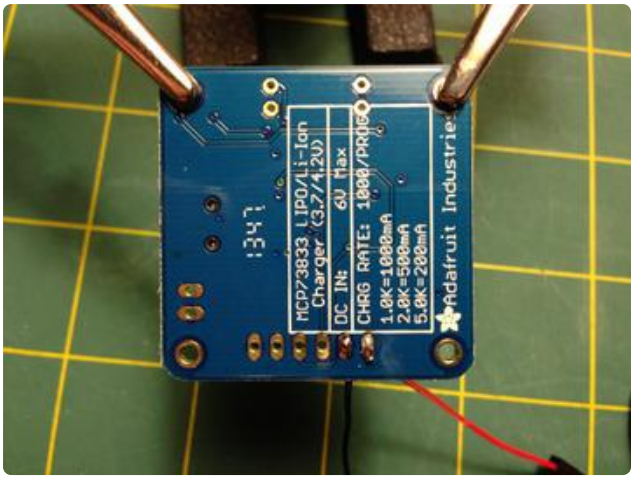


Jumper Wires

Pull four jumpers away from the bunch. The colors don't matter as long as you make the right connections. Snip the male leads off of all four jumpers (we won't use them). Strip about 1/8" of insulation away from the cut end and twist the bare ends.

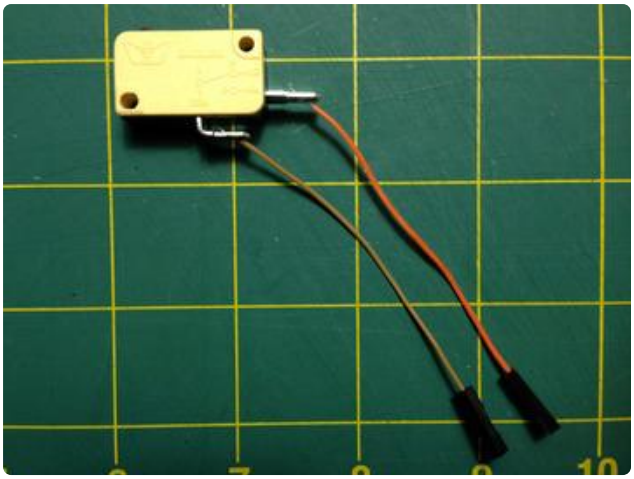
Two leads will connect the EZ-Key to the charge controller and two will connect the button to the EZ-Key.

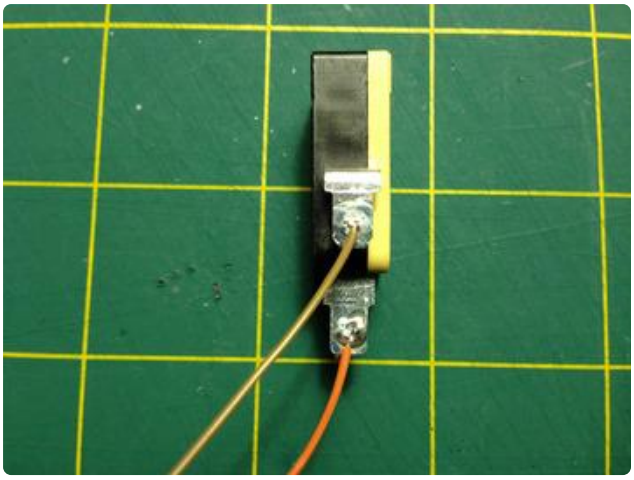


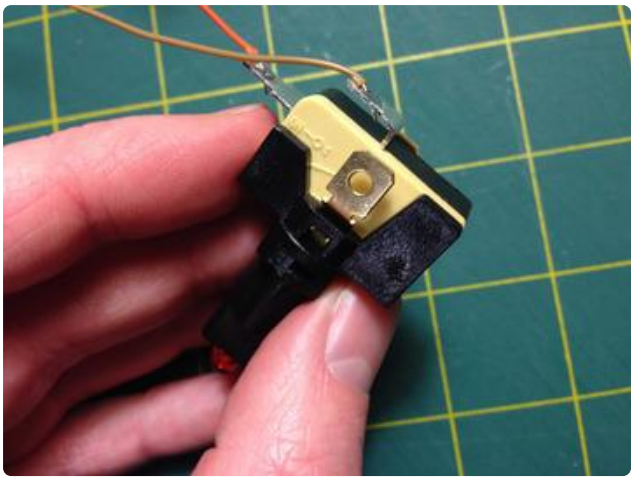


Charge Controller Jumpers

The EZ-Key will be powered from the charge controller. Take two of the jumper wires and solder them to the two holes labeled "LOAD" as pictured.







Arcade Microswitch Jumpers

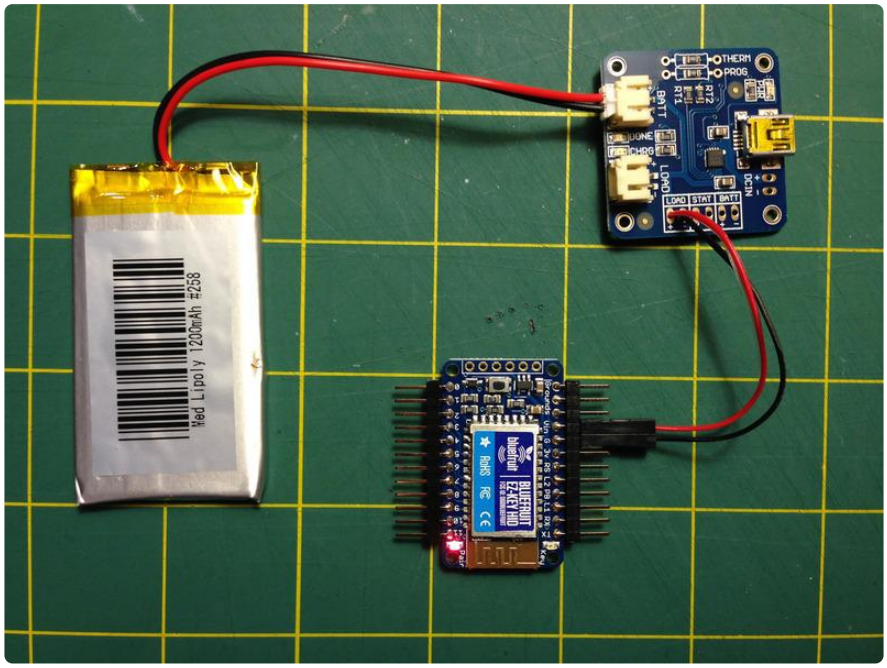
The massive arcade button includes a standard microswitch. The switch type is a "NO" which means "normally open." Pushing the button will cause the two jumpers to become connected. This is what triggers the EZ-Key to send a key press.

Take other two jumper wires and solder one to each of the terminals. It doesn't matter which one goes where.

Snap the microswitch into the button assembly along with the included LED. We won't be connecting the LED in this project, but install it for safe keeping.

Insert the button assembly into the massive red button from inside the box. Insert it and twist to lock. Once it's in, you should hear a distinctive click when you press on the red dome.

Dry Test



Let's test to make sure we're getting power from the battery through the charge controller to the EZ-Key.

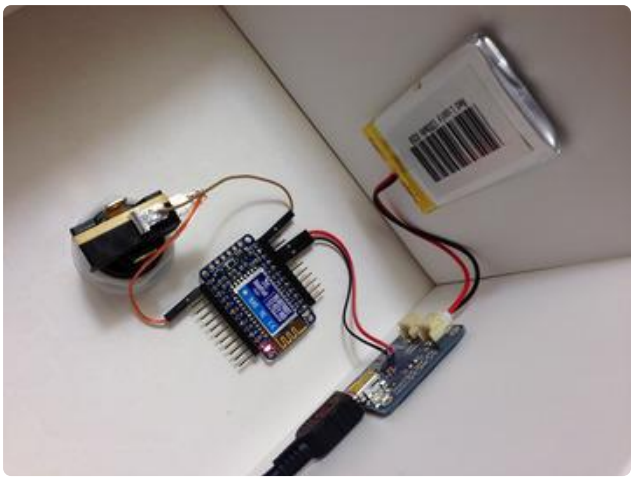
Connect the jumpers from the charge controller labeled "LOAD" to the EZ-Key. Connect plus (+) to "VIN" on the EZ-Key and the minus (-) to "G" on the EZ-Key.

EZ-Key	Charge Controller
VIN	LOAD Plus (+)
G	LOAD Minus (-)

Make sure to observe the correct polarity when connecting the EZ-Key to the charge controller!

Plug in the JST connector from the battery to the charge controller labeled "BATT."

The red light on the EZ-Key should now begin blinking. If so, congratulations! If it doesn't light, the battery may not have enough power. See the step below to charge the battery first before assembling.



Assemble

Cut short pieces of double-sided foam tape and stick it to the back of the battery, charge controller and EZ-Key. Remove the backing and stick the three inside the box in close proximity. Connect the battery and EZ-Key to the charge controller as in the dry test above.

Attach the button jumpers to the EZ-Key. One lead will go to pin 0 and the other will go to any of the leads marked "Grounds."

Charge the Battery

Charge the battery by connecting it to USB via the port on the rear. The yellow "CHRG" LED will light up. When the charge is complete, the green "DONE" LED will light up. You can unplug the veto button and move it anywhere in your office (within it's range).

Configure the EZ-Key

Pairing EZ-Key

Before we can configure the Bluefruit EZ-Key, we must first pair it with your computer. Follow the instructions for pairing in the Bluefruit EZ-Key guide ().

Bluefruit EZ-Key Paring Guide

Output Key Remapping

The Bluefruit EZ-Key comes with 12 key presses pre-configured.

- #0 - Up Arrow
- #1 - Down Arrow
- #2 - Left Arrow
- #3 - Right Arrow
- etc..

None of these will work for us since they're common keys that we can't assign to a global hot key. Therefore, we need to reconfigure the mapping. Follow the instructions for changing the stored key presses in the EZ-Key in the Bluefruit EZ-Key Guide (). Reconfigure output #0 since that's what we've wired to the button. In my case, I chose F3 as my global hot key.

Bluefruit EZ-Key Wireless Key Remapping

Remember the hot key you choose, you'll need it in the next step.

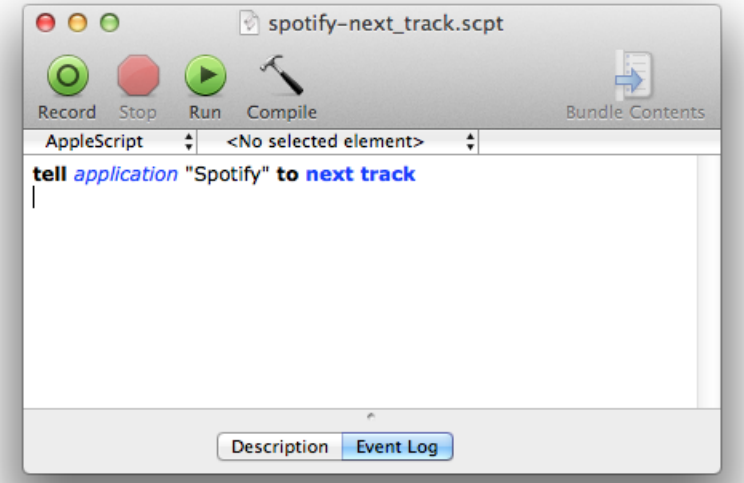
Configure the Hot Key

AppleScript

Now we need to get the key press that will be coming from the Bluefruit EZ-Key over Bluetooth connected to an action on your computer. We'll do this through a combination of AppleScript and a program called QuickSilver to bind that script to a key press that works regardless of which program is in focus.

Open AppleScript Editor and paste in the following line into a new document.

```
tell application "Spotify" to next track
```



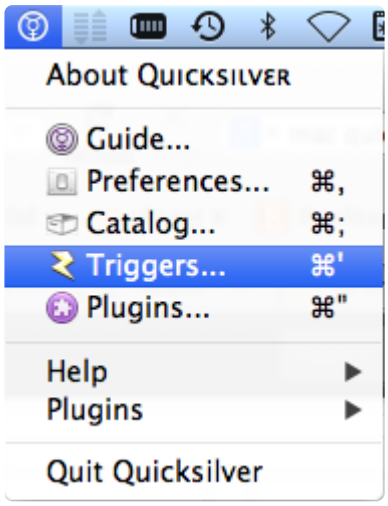
You can test this script by clicking the "Play" icon at the top. If Spotify is already open and playing a song, it should skip to the next song. If it's open but not playing, it will begin playing the next track. If Spotify isn't open, it will open the application.

Save this AppleScript to a location you can find again (your Documents folder is a good location). The file format should be "Script."

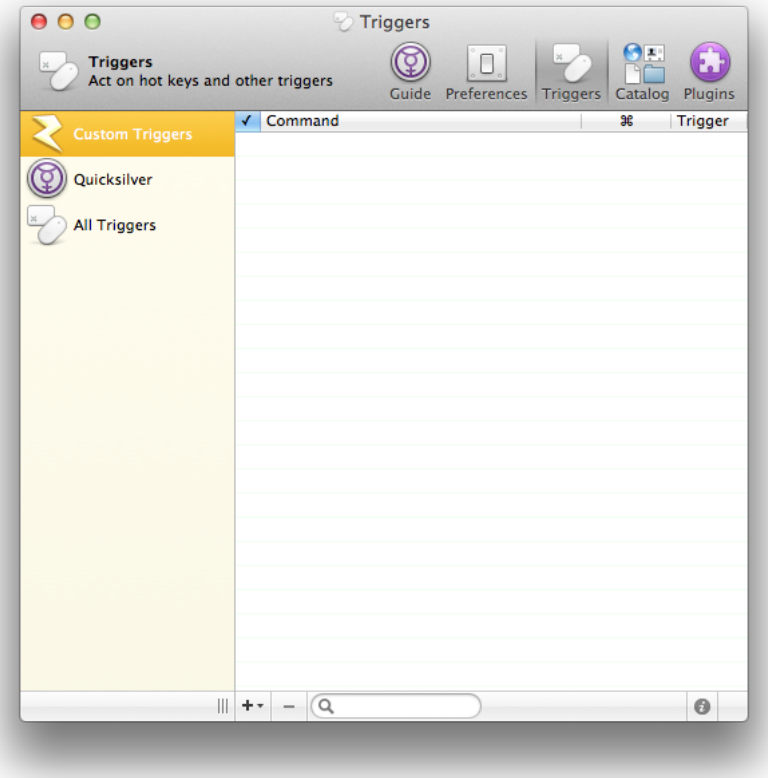
Quicksilver

Next, we go to QuickSilver. It's main use is a hyper fast application launcher that puts Spotlight to shame. We'll be using it in this case for it's "Triggers" feature.

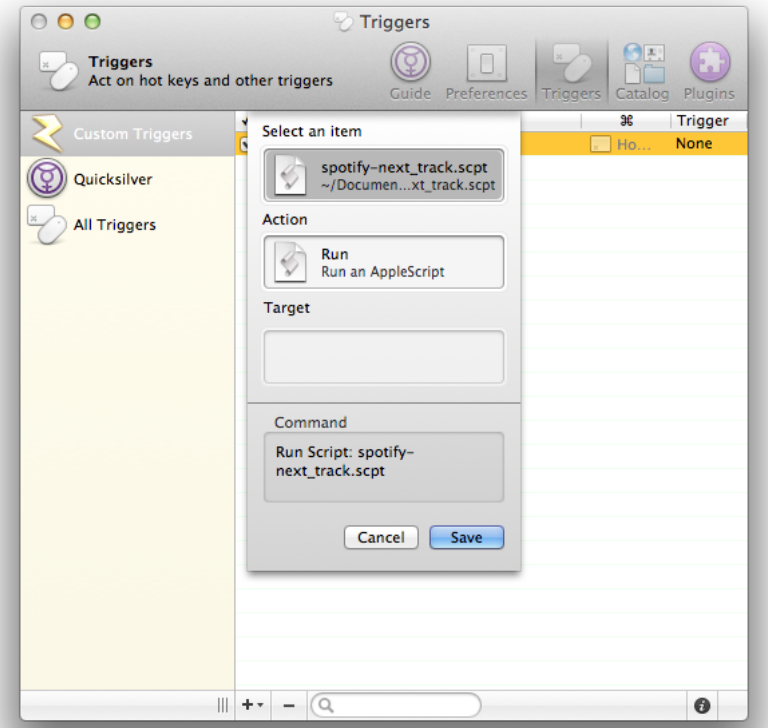
If you haven't already, install the latest QuickSilver () on your Mac. When ready, you should have a new menu bar item at the top near the right. Click it and choose "Triggers.."



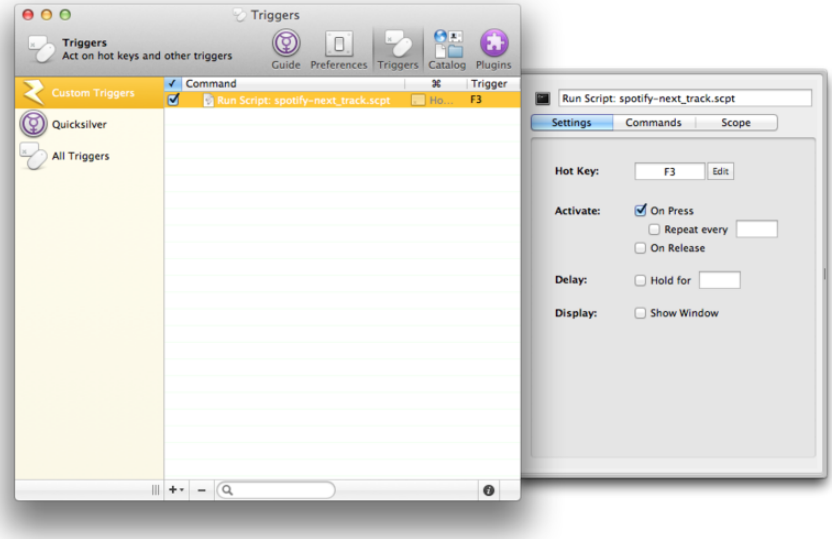
Click the "+" at the bottom of the window and choose "HotKey".



In the box under "Select an item," drag and drop the new AppleScript you wrote into the space. The action will change automatically to "Run an AppleScript." Click "Save."



Now to assign the hot key. Click "None" under Trigger to reveal the Settings pane. Click inside the box next to "Hot Key." The words "Set Keys" should now be pulsing. Select your keystroke you programmed into the EZ-Key. In my case, it was F3.



Close the triggers window. You should now be able to test the operation from your keyboard. Press your chosen hot key. If Spotify is already open and playing a song, it should skip to the next song. If it's open but not playing, it will begin playing the next track. If Spotify isn't open, it will open the application.

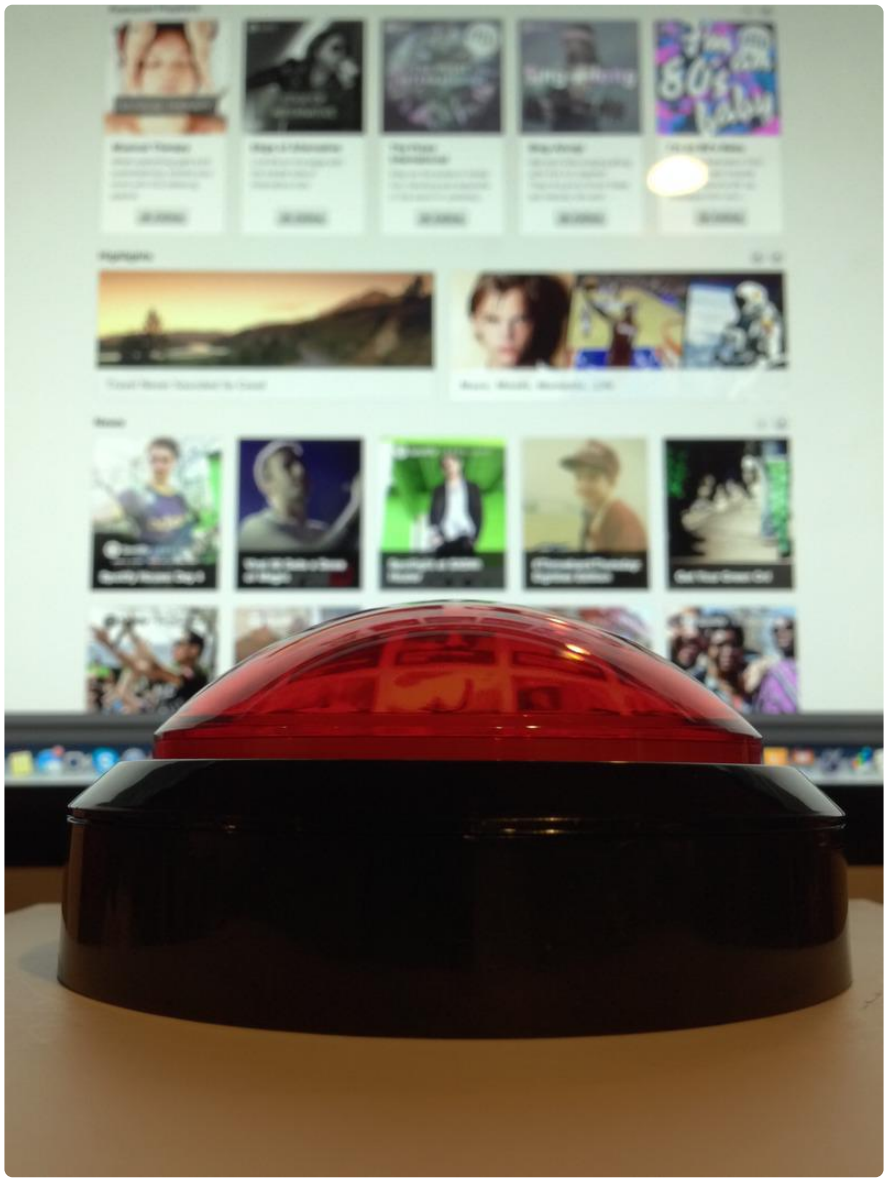
If it doesn't work, first check the hot key settings inside QuickSilver's Triggers window.

I have discovered that the hot key will not work if you are controlling another computer via screen sharing. This makes sense since the hot key is being sent to the remote computer. Keep this in mind if the button is being mashed mercifully.

What about Windows?

The project could be translated to Windows given the appropriate software to map a global hot key to an event.

Use it!



Get ready to cast your veto!

If music is playing and you press the button, it should skip to the next track.

If no music is playing and you press the button, it should begin playing.

If Spotify isn't open and you press the button, it should launch the app. Press the button again and it will begin playing.

Charge the battery every few days (it should last about 3 days on a full charge)

Improvements and Enhancements

There are a couple of ways you can build on this project. Since there are 11 unused output pins on the EZ-Key you can add a few more smaller buttons that perform other tasks like volume, muting, pause, etc. You can also add LED charge indicator lights to the outside by connecting them to the charge controller "STAT" area.

Want to control iTunes instead? It's as simple as changing your AppleScript as follows.

```
tell application "iTunes" to next track
```

Program the AppleScript to perform any number of actions on your computer like opening a new web browser, turning the page of an e-book, or even shutdown your computer!