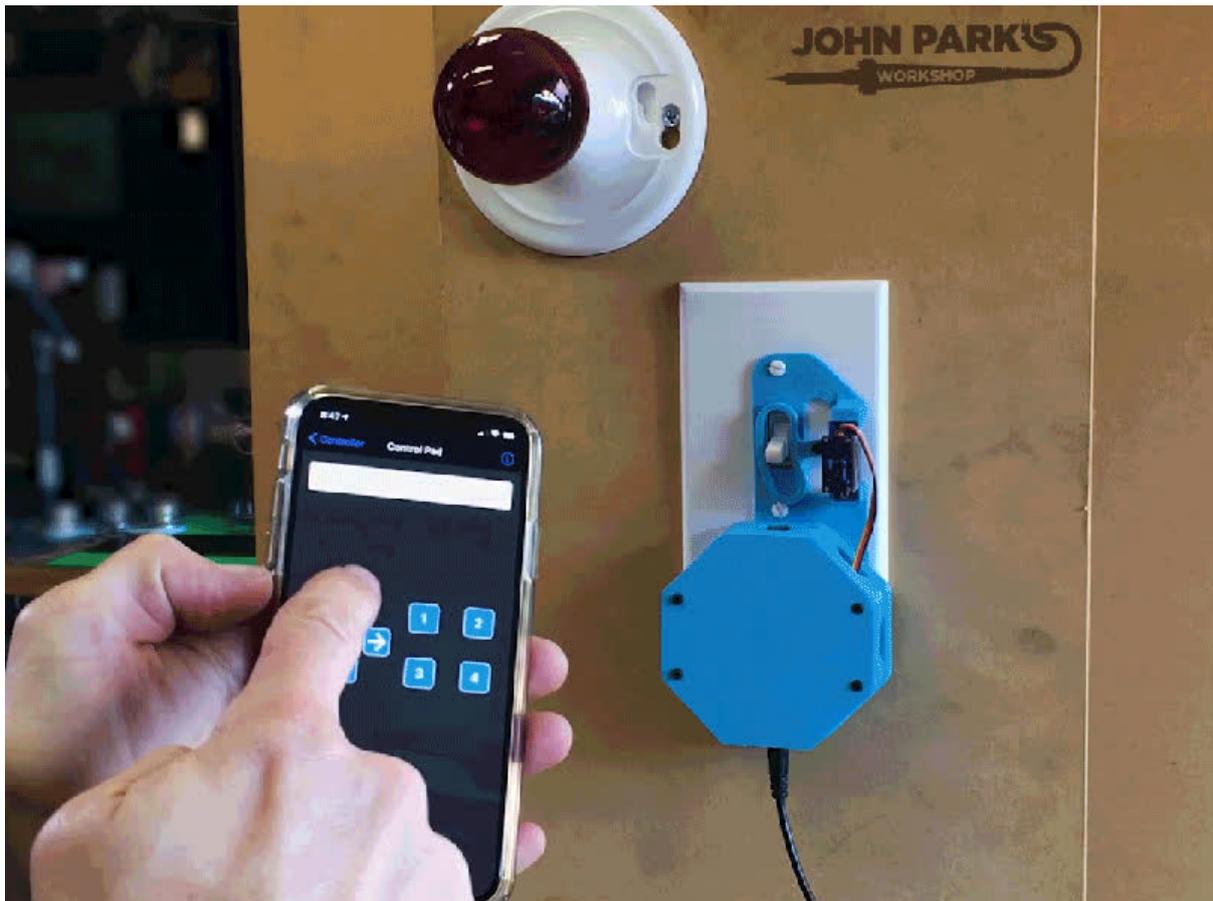




BLE Light Switch with Feather nRF52840 and Crickit

Created by John Park



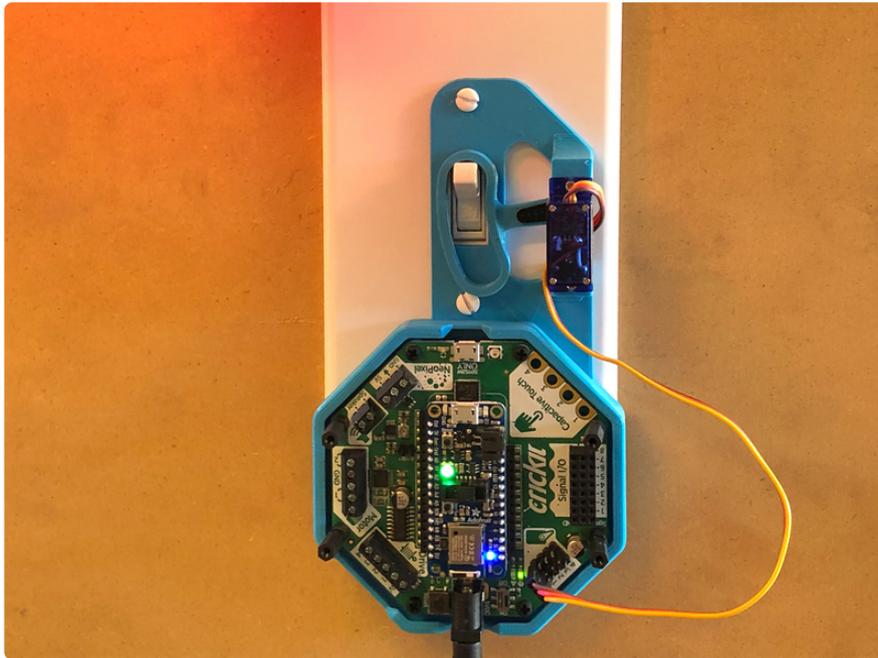
<https://learn.adafruit.com/bluetooth-light-switch-with-crickit-and-nrf52840>

Last updated on 2024-03-08 03:19:02 PM EST

Table of Contents

Overview	3
<ul style="list-style-type: none">• Parts• Materials and Tools	
Build the Switch Bracket	5
<ul style="list-style-type: none">• Download STL Model Files• Slicing in CURA• Prep the Feather and Crickit• Threaded Inserts• Mount the Crickit into the Bracket• Servo Arm• Servo Connection	
Code with CircuitPython	15
<ul style="list-style-type: none">• Feather and Crickit Prep• Libraries• CircuitPython Code• How the Program Works• Control the Light Switch with the Bluefruit LE Connect App• Connect to Feather• Troubleshooting• Device Menu• Controller Menu• Move the Servo Arm	
Mount and Use the Remote Switch	23
<ul style="list-style-type: none">• Mounting• Cover	

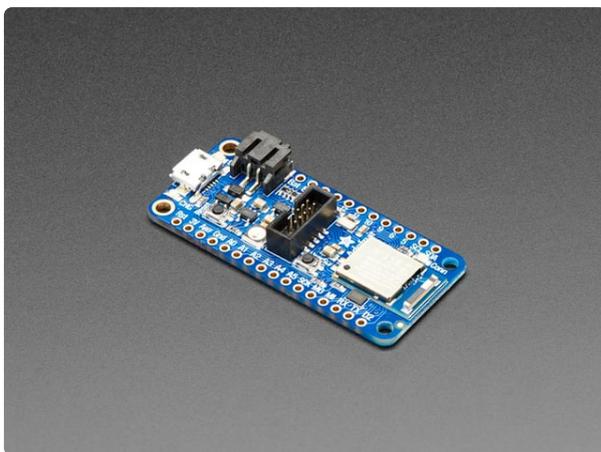
Overview



Build your own Bluetooth, remote-controlled robotic light switch to turn on and off the lights! Using CircuitPython running on the Feather nRF52840 Express and Crickit FeatherWing to flip the switch with a servo motor, you can avoid all of the complexities and dangers of high voltage mains wiring!

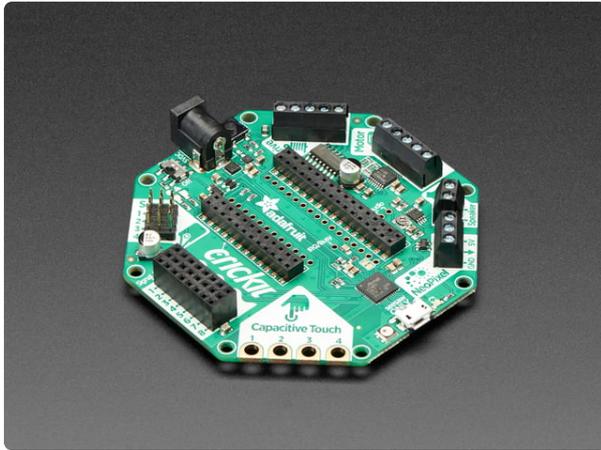
The 3D printed actuator, mount, and case makes it reliable and tidy. And the Adafruit Bluefruit app running on your iOS or Android device makes it fun and easy!

Parts



[Adafruit Feather nRF52840 Express](https://www.adafruit.com/product/4062)

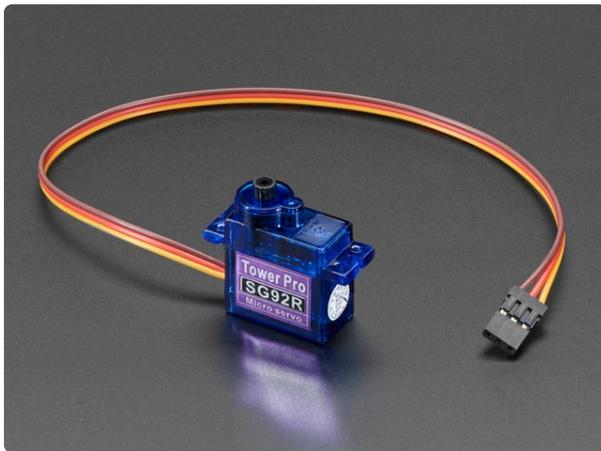
The Adafruit Feather nRF52840 Express is the new Feather family member with Bluetooth Low Energy and native USB support featuring the nRF52840! It's... <https://www.adafruit.com/product/4062>



Adafruit CRICKIT FeatherWing for any Feather

Sometimes we wonder if robotics engineers ever watch movies. If they did, they'd know that making robots into servants always ends up in a robot rebellion. Why even go down that...

<https://www.adafruit.com/product/3343>



Micro servo

Tiny little servo can rotate approximately 180 degrees (90 in each direction) and works just like the standard kinds you're used to but smaller. You can use any servo...

<https://www.adafruit.com/product/169>



5V 2A (2000mA) switching power supply - UL Listed

This is an FCC/CE certified and UL listed power supply. Need a lot of 5V power? This switching supply gives a clean regulated 5V output at up to 2000mA. 110 or 240 input, so it works...

<https://www.adafruit.com/product/276>



Black Nylon Machine Screw and Stand-off Set - M2.5 Thread

Totalling 380 pieces, this M2.5 Screw Set is a must-have for your workstation. You'll have enough screws, nuts, and hex standoffs to fuel your maker...

<https://www.adafruit.com/product/3299>

Materials and Tools

In addition to the parts above you'll also need:

- 8 ea. [Heat-set, M2.5 x 0.45mm threaded inserts](https://adafru.it/DR9) (<https://adafru.it/DR9>)
- Soldering iron
- Small Philips screwdriver
- Slotted screwdriver
- 3D printer and PLA filament



Build the Switch Bracket

The 3D printed parts are a mashup of a remix! I started with this [Smart light switch with a servo](https://adafru.it/DRa) (<https://adafru.it/DRa>) by Thingiverse user [pulsar123](https://adafru.it/DRb) (<https://adafru.it/DRb>) (which is a remix of [Servo Switch Plate Mount](https://adafru.it/DRc) (<https://adafru.it/DRc>) by [carjo3000](https://adafru.it/DRd) (<https://adafru.it/DRd>)).

I combined it with the excellent [Mount for Crickit](https://adafru.it/BXU) (<https://adafru.it/BXU>) by the Ruiz Brothers.

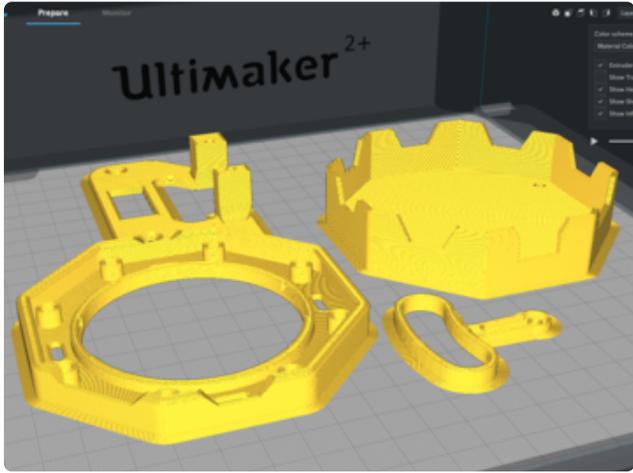
I also modeled a cover for the Crickit that can be screwed in place once everything is set up.

Download STL Model Files

You'll need to download these three STL files to 3D print the mount, servo arm, and cover. Click the button below to download the files.

[Download STL files from Thingiverse](https://adafru.it/DRf)

<https://adafru.it/DRf>



Slicing in CURA

These settings were used to print the three pieces (not all at once as shown here!):

0.4mm Nozzle

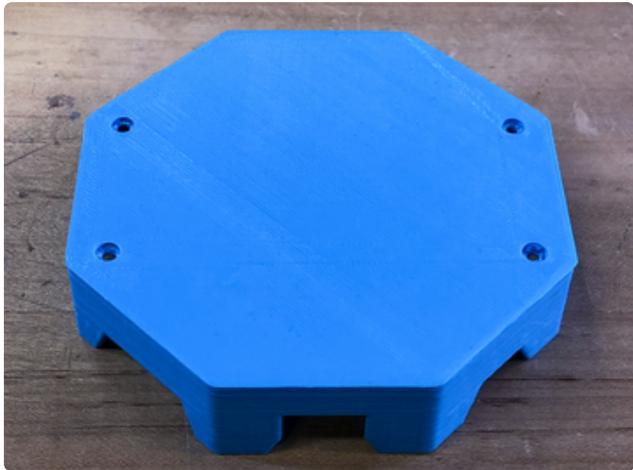
0.2mm Layer Height

0.38 Line Width

2 Wall line count

70mm/s printing speed

10% infill

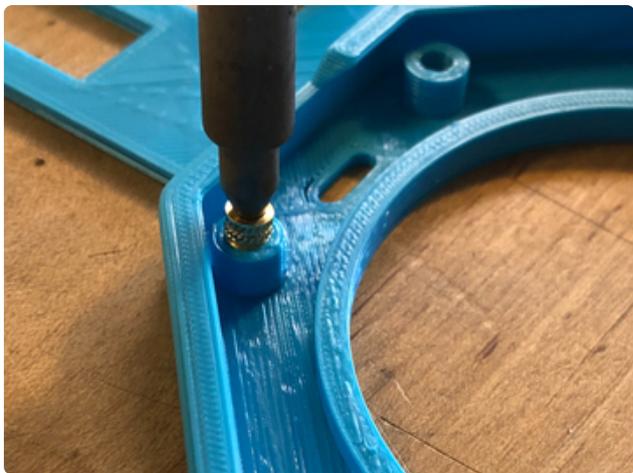


Print the three parts and then get ready to add the electronics!

Prep the Feather and Crickit

Next, solder the plain male header pins into place on the Feather so that it can be mounted onto the Crickit. You can [follow this guide for instructions \(https://adafru.it/DRI\)](https://adafru.it/DRI).

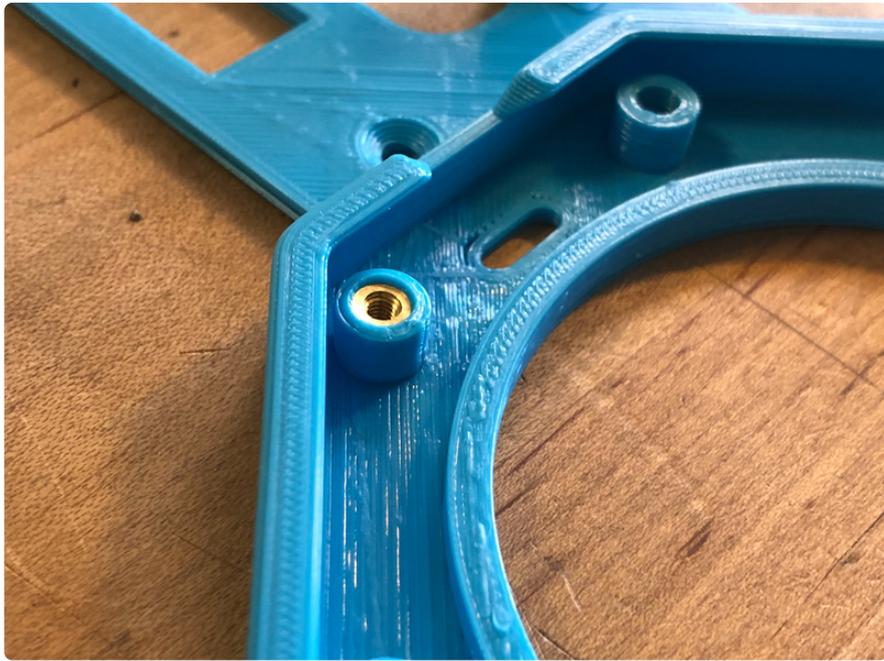
Then, mount the Feather onto the Crickit's header sockets.

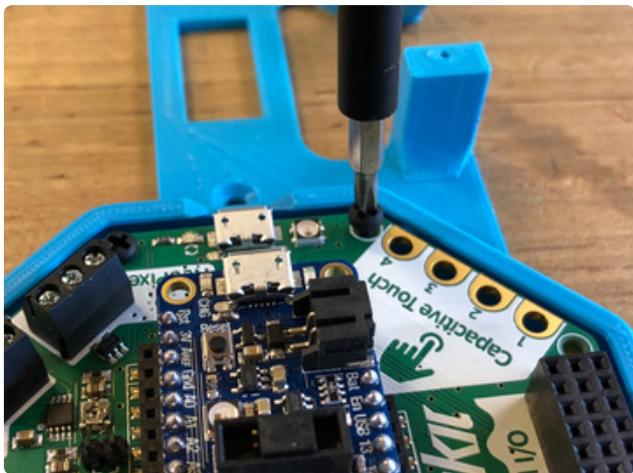
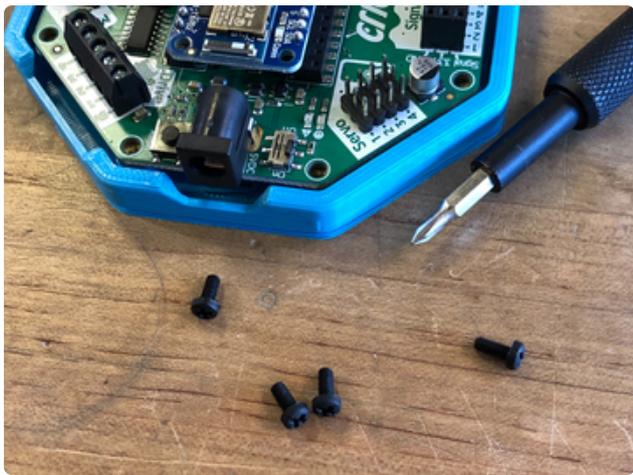
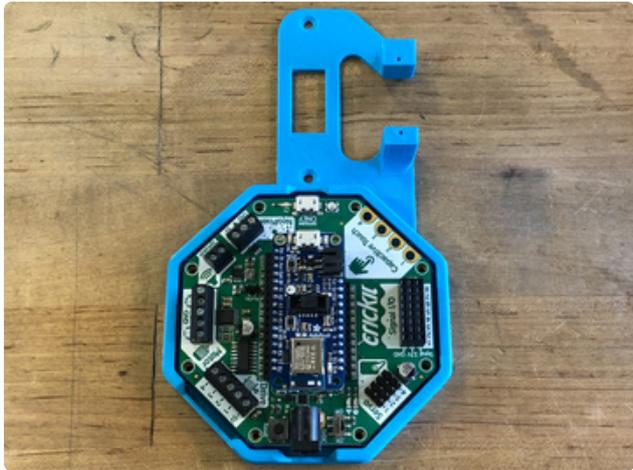


Threaded Inserts

In order to screw the Crickit in place, as well as the standoffs for the cover, we'll add heat-set threaded inserts to the bracket.

Place the tapered smaller end of an insert into an opening and then use the soldering iron to heat and press the brass insert into place.

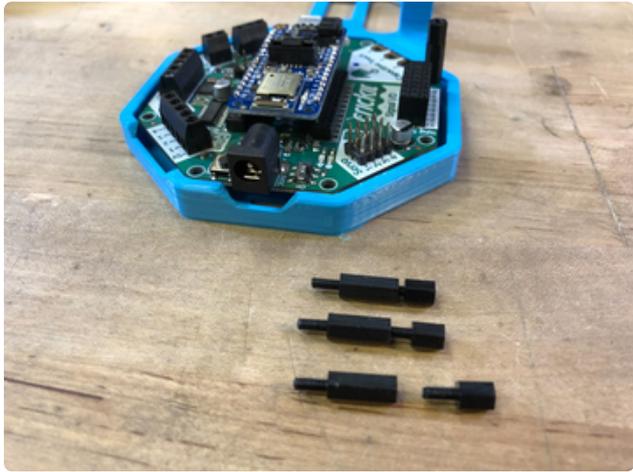




Mount the Crickit into the Bracket

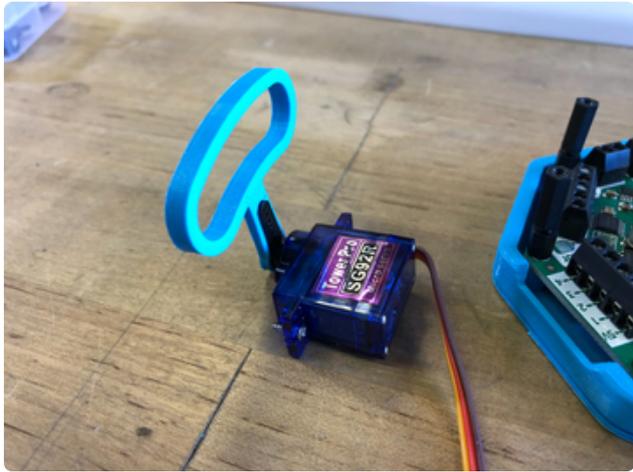
Use four of the M2.5 x 6mm screws in to fasten the Crickit to the top two and bottom two threaded inserts.

Then, screw in four stacks of paired M-F standoffs. Each pair will consist of a 6mm and 12mm standoff to get the proper clearance needed for the cover.

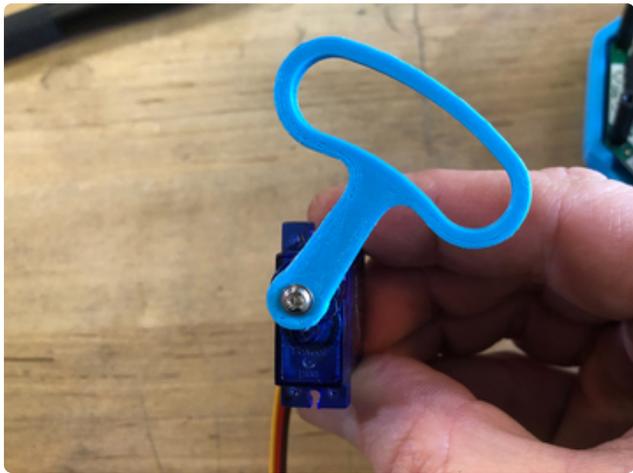


Servo Arm

The servo arm would have a hard time turning the switch on it's own, so we'll add the 3D printed extension. It has a curved guide with stops that allow it to flip the traditional light switch easily.



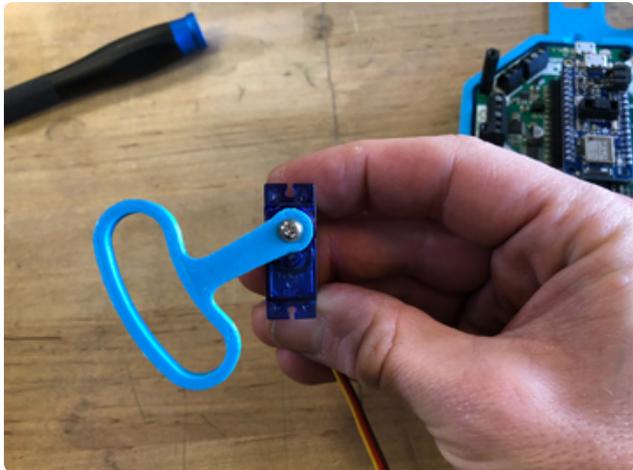
Fit the 3D printed arm over the small servo horn supplied with the micro servo.



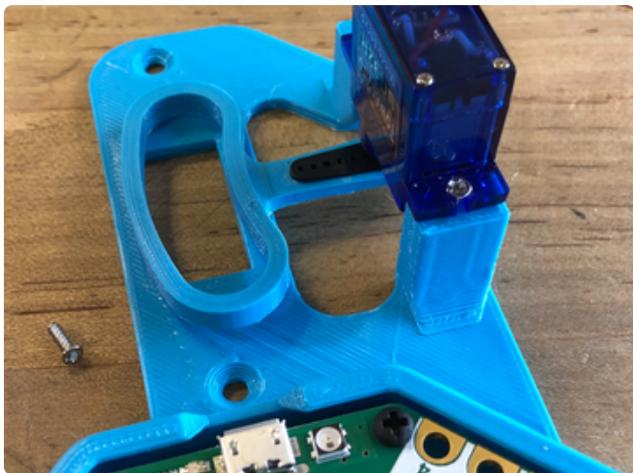
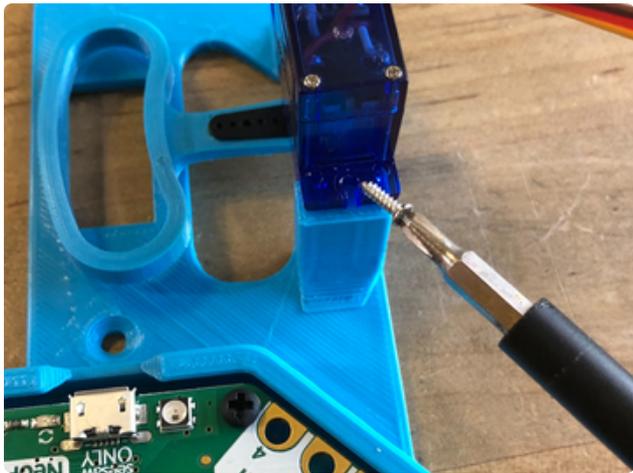
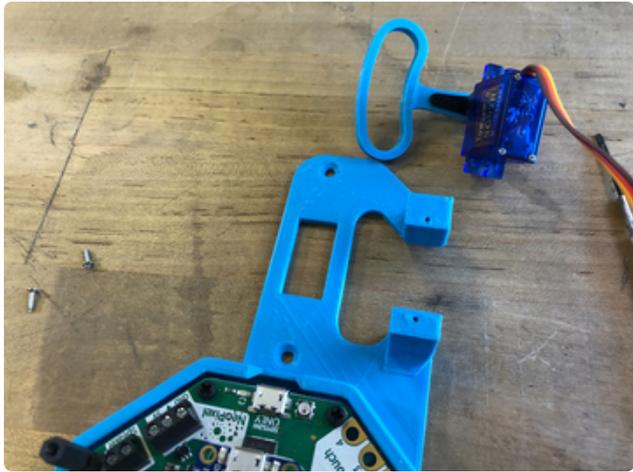
Place the horn onto the spline shaft of the servo, then turn it to find the range of motion.

Turn the servo to roughly its middle position and then remove and re-seat the arm so it is oriented as shown so that it will be able to flip the switch when mounted.

Use the supplied screw to secure the arm and horn to the shaft.

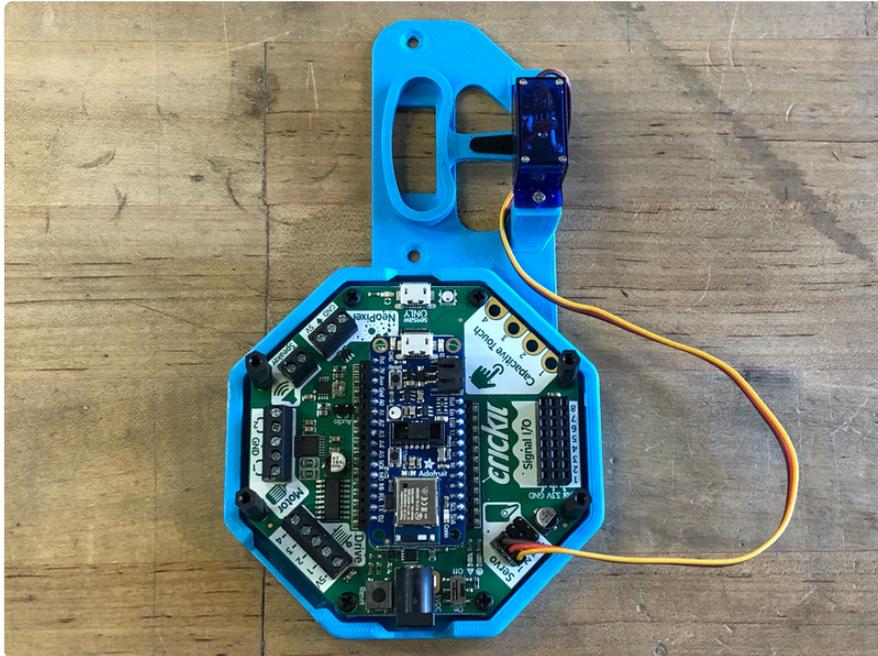


Secure the servo to the mount with the two small screws.



Servo Connection

The servo motor uses three pins -- power, ground, and signal. Plug the servo header into the Crickit Servo 1 port with the yellow signal wire facing out as shown.



Next, we'll program the Feather using CircuitPython and test it before mounting it to the switch plate.

Code with CircuitPython

Feather and Crickit Prep

We'll be using CircuitPython for this project. Are you new to using CircuitPython? No worries, [there is a full getting started guide here](https://adafru.it/cpy-welcome) (<https://adafru.it/cpy-welcome>).

For more info on the Feather nRF52840 Express, [check out this guide](https://adafru.it/DLI) (<https://adafru.it/DLI>).

Adafruit suggests using the Mu editor to edit your code and have an interactive REPL in CircuitPython. [You can learn about Mu and its installation in this tutorial](https://adafru.it/ANO) (<https://adafru.it/ANO>).

Follow this guide for instructions on installing the latest release version of CircuitPython for the Feather nRF52840 Express:

Installing CircuitPython

<https://adafru.it/Amd>

Note: This project uses the Bluetooth functionality available in CircuitPython 5.0.0-beta.0 and later, on nRF52840 boards such as the Feather nRF52840 Express and Circuit Playground Bluefruit. Make sure you are using that version or later.

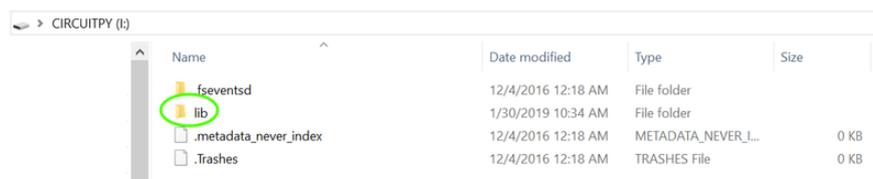
Libraries

You'll also need to add the following libraries for this project. [Follow this guide \(https://adafru.it/Cqa\)](https://adafru.it/Cqa) on adding libraries.

Plug your nRF Feather board into your computer via a USB cable. Please be sure the cable is a good power+data cable so the computer can talk to the board.

A new disk should appear in your computer's file explorer/finder called **CIRCUITPY**. This is the place we'll copy the code and code library. If you can only get a drive named **FTHR840BOOT**, load CircuitPython per the guide above.

Download the latest CircuitPython libraries to your computer using the green button below. **Match the library you get to the version of CircuitPython you are using.** Save to your computer's hard drive where you can find it.



Download the latest CircuitPython library bundle

<https://adafru.it/y8E>

With your file explorer/finder, browse to the bundle and open it up. Copy the following folders and files from the library bundle to your **CIRCUITPY lib** directory you made earlier:

The ones you'll need are:

- **adafruit_ble** (folder)

- `adafruit_bluefruit_connect` (folder)
- `adafruit_bus_device` (folder)
- `adafruit_crickit.mpy` (file)
- `adafruit_motor` (folder)
- `adafruit_seesaw` (folder)

All of the other necessary code is baked into CircuitPython!

CircuitPython Code

Copy the program below and paste it into a new document in Mu. Then, save it from Mu onto your **CIRCUITPY** flash drive as **code.py**. As soon as CircuitPython restarts, the servo arm will move into its neutral position and await your Bluetooth connection!

```
# SPDX-FileCopyrightText: 2019 John Edgar Park for Adafruit Industries
#
# SPDX-License-Identifier: MIT

# BLE Crickit Light Switch
# Use with the Adafruit BlueFruit LE Connect app
# Works with CircuitPython 5.0.0-beta.0 and later
# running on an nRF52840 Feather board and Crickit FeatherWing
# micro servo, 3D printed switch actuator

import time

import board
import digitalio

from adafruit_crickit import crickit

from adafruit_ble import BLERadio
from adafruit_ble.advertising.standard import ProvideServicesAdvertisement
from adafruit_ble.services.nordic import UARTService

from adafruit_bluefruit_connect.packet import Packet
# Only the packet classes that are imported will be known to Packet.
from adafruit_bluefruit_connect.button_packet import ButtonPacket

# Prep the status LEDs on the Feather
blue_led = digitalio.DigitalInOut(board.BLUE_LED)
red_led = digitalio.DigitalInOut(board.RED_LED)
blue_led.direction = digitalio.Direction.OUTPUT
red_led.direction = digitalio.Direction.OUTPUT

ble = BLERadio()
uart_service = UARTService()
advertisement = ProvideServicesAdvertisement(uart_service)

UP_ANGLE = 180
NEUTRAL_ANGLE = 120
DOWN_ANGLE = 60
crickit.servo_1.angle = NEUTRAL_ANGLE

angle = NEUTRAL_ANGLE # use to track state

print("BLE Light Switch")
print("Use Adafruit Bluefruit app to connect")
```

```

while True:
    blue_led.value = False
    ble.start_advertising(advertisement)

    while not ble.connected:
        # Wait for a connection.
        pass
    blue_led.value = True # turn on blue LED when connected
    while ble.connected:
        if uart_service.in_waiting:
            # Packet is arriving.
            red_led.value = False # turn off red LED
            packet = Packet.from_stream(uart_service)
            if isinstance(packet, ButtonPacket) and packet.pressed:
                red_led.value = True # blink to show a packet has been received
                if packet.button == ButtonPacket.UP and angle != UP_ANGLE: # UP
                    # UP
                    # button pressed
                    # of goal angle
                    # ramp to goal angle
                    angle = NEUTRAL_ANGLE - 45 # set anticipation angle, opposite
                    for a in range(angle, UP_ANGLE+1, 1): # anticipation angle,
                        crckit.servo_1.angle = a
                        time.sleep(0.1) # wait a moment
                        crckit.servo_1.angle = NEUTRAL_ANGLE # then return to neutral
                    angle = UP_ANGLE # set state to prevent redundant hits
                elif packet.button == ButtonPacket.DOWN and angle != DOWN_ANGLE: #
                    # DOWN button
                    angle = NEUTRAL_ANGLE + 45
                    for a in range(angle, DOWN_ANGLE-1, -1):
                        crckit.servo_1.angle = a
                        time.sleep(0.1) # wait a moment
                        crckit.servo_1.angle = NEUTRAL_ANGLE # then return to neutral
                    angle = DOWN_ANGLE # set state to prevent redundant hits

```

How the Program Works

The first part of the program sets up a **UARTServer**, which is a BLE service that wirelessly sends streams of characters between a BLE device and a client computer (which could be a phone or tablet). The program then starts advertising its UART service to anyone listening. The client computer receives the advertisement, and tells the user it's available. The user can then choose to connect as a client of the service.

Once a connection is established, the program code waits for incoming command packets from the client, specifically a **ButtonPacket**. When one arrives, the program either adjusts or sees which button has been pressed: up-arrow or down-arrow.

Servo Motion

The servo will initially go to a neutral, half-way position of 120°.

In order to create a nice motion with the servo arm that can stylishly flip the switch, and still allow manual use of it, we'll do a few things when the button packet is received.

First, we'll anticipate the motion. This means that if the goal is to move the switch up, first the arm will swing back 45° away from the neutral position of 120°, sort of like pulling your arm back before a throw, or squatting down before a jump.

Then, we'll rotate to the goal angle of 180° for up or 60° for down. Rather than jump immediately to the goal value, the code moves the servo one degree at a time in the specified direction, until it hits the numerical limit. This gives us a slower, smoother motion.

Once the goal angle is reached, we pause for a brief moment and then return the arm to the neutral angle.

Control the Light Switch with the Bluefruit LE Connect App

Download the App

To control the servo and NeoPixels from a client device (phone or tablet), you use the Adafruit [Bluefruit LE Connect App \(https://adafru.it/iCi\)](https://adafru.it/iCi). Install it from the [Apple App Store \(https://adafru.it/BYj\)](https://adafru.it/BYj) or [Google Play App Store \(https://adafru.it/f4G\)](https://adafru.it/f4G).

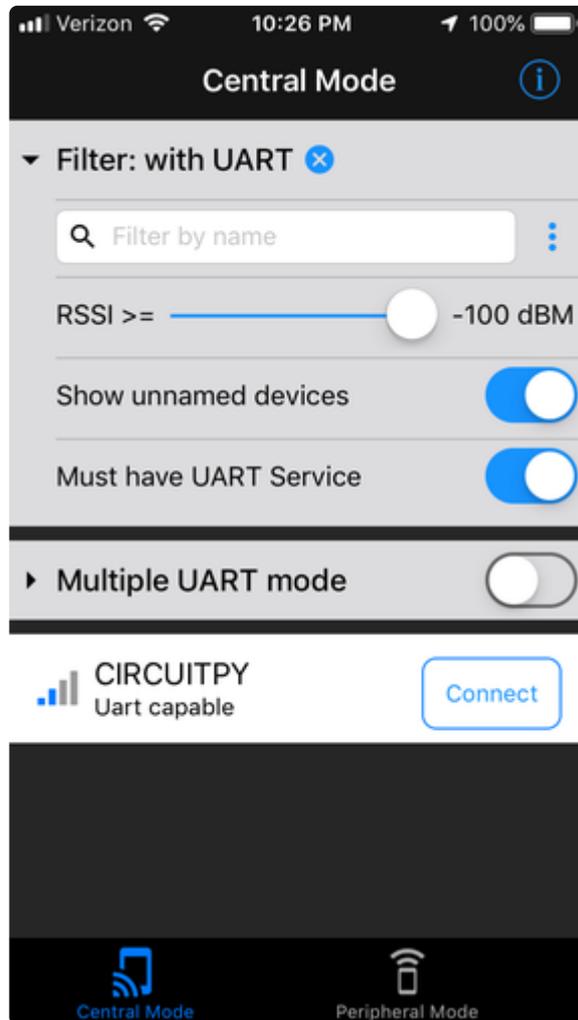
Connect to Feather

With the Crickit plugged into DC power and turned on with its on-board power switch, the Feather will also get power and automatically start running your code. Then start up the Bluefruit LE Connect app, and make sure it's in Central Mode (left button on the bottom). When you start the app, you should see a device named **CIRCU** or **CIRCUITPY**. If there's a long list of devices, you can shorten it by turning on the "**Must have UART Service**" switch.

To connect to the Light Switch, touch the **Connect** button. You should see "**Connecting**" and then "**Discovering Services**".

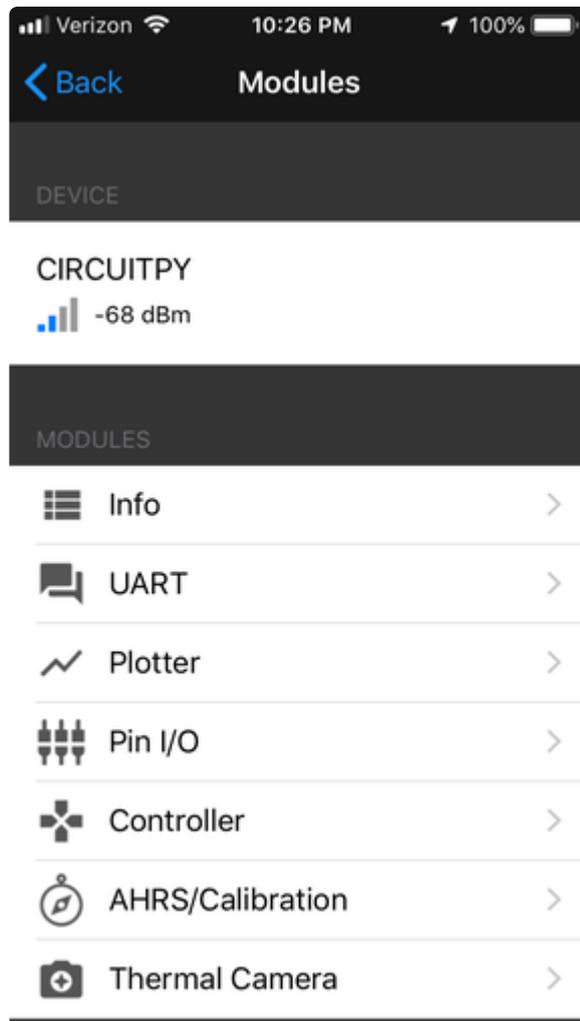
Troubleshooting

If you don't see **CIRCU** or **CIRCUITPY** right away, try pulling down to refresh. If that doesn't work, try turning Bluetooth off and back on on your phone or tablet and restarting the app.



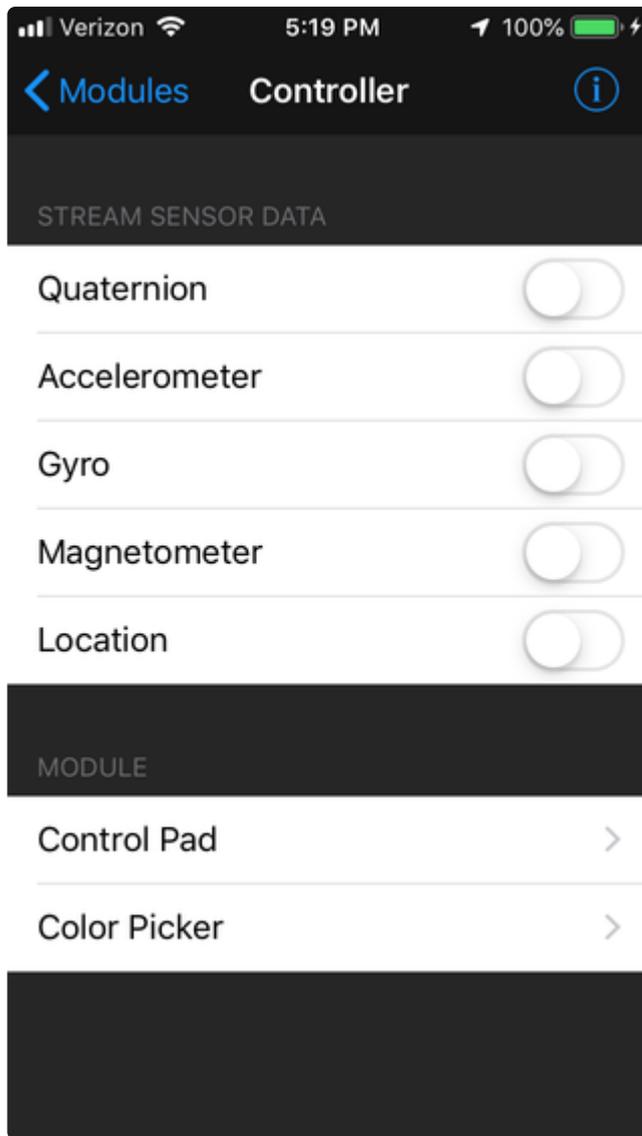
Device Menu

After you connect, you'll see a menu of how you can interact with the device. Choose **Controller**.



Controller Menu

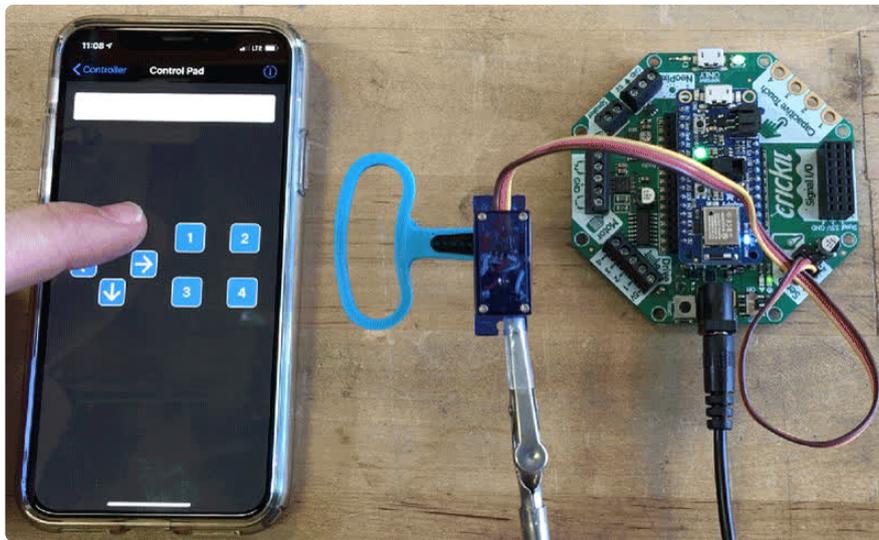
After you choose Controller, you'll see another screen with more choices. **Control Pad** is used for this project.



Move the Servo Arm

Here's the **Control Pad** screen. Use the **Up** and **Down** buttons to move the servo arm up and down. Later, we'll use this to flip the lights on and off!

You should see the arm start in a neutral center position, and then when you press the **up** ↑ button, the arm will travel down a bit and then fully up to the top of the range chosen in the code, finally returning to its neutral position. The same motion pattern applies in reverse for the **down** ↓ button.



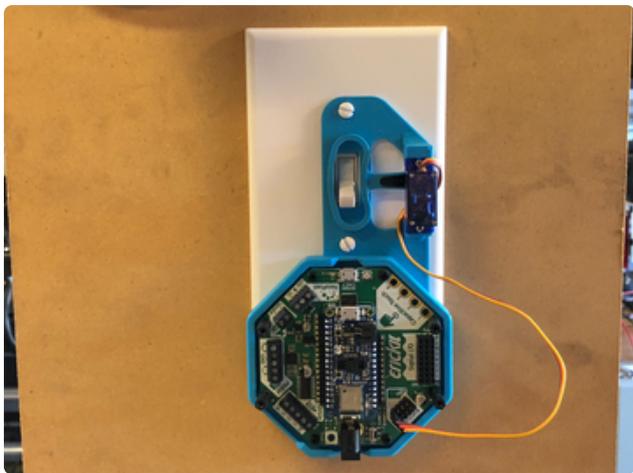
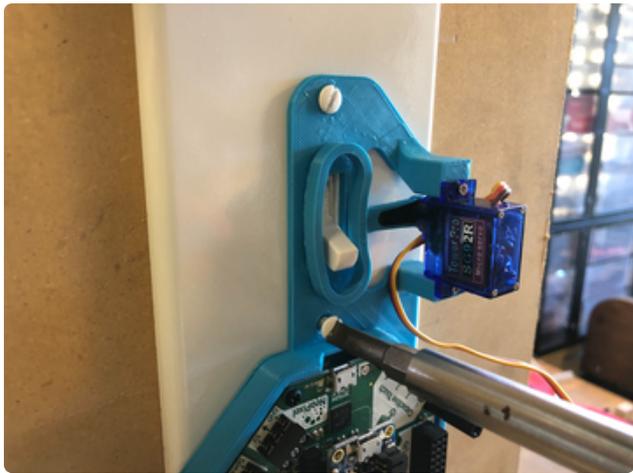
Next, we'll hook it up to the actual light switch plate!

Mount and Use the Remote Switch

Now, we'll mount the remote to the light switch plate and put it to use!



Try to avoid actually taking the switch plate off (just take out the screws). This way you will avoid the high voltage mains wiring behind the plate.

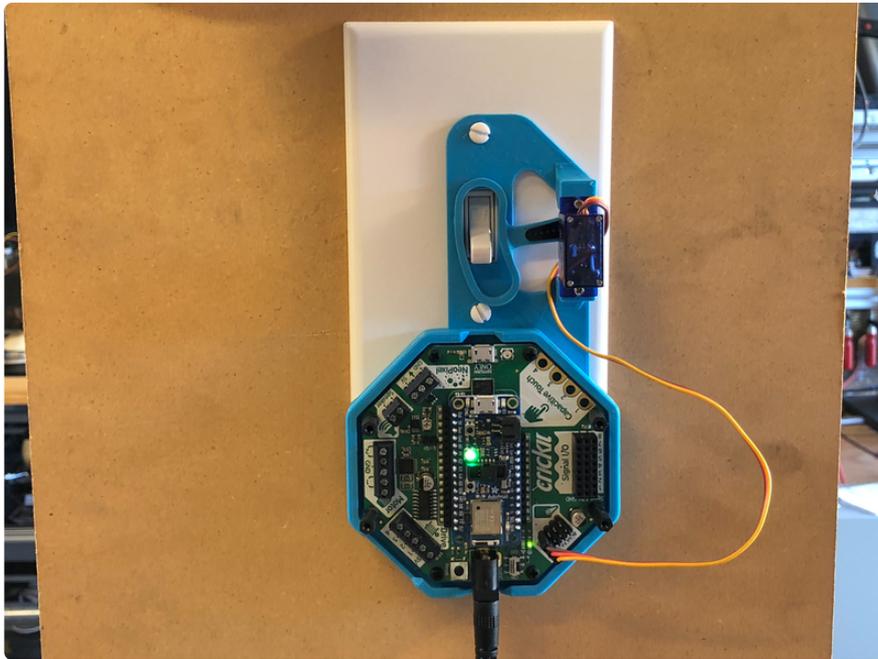


Mounting

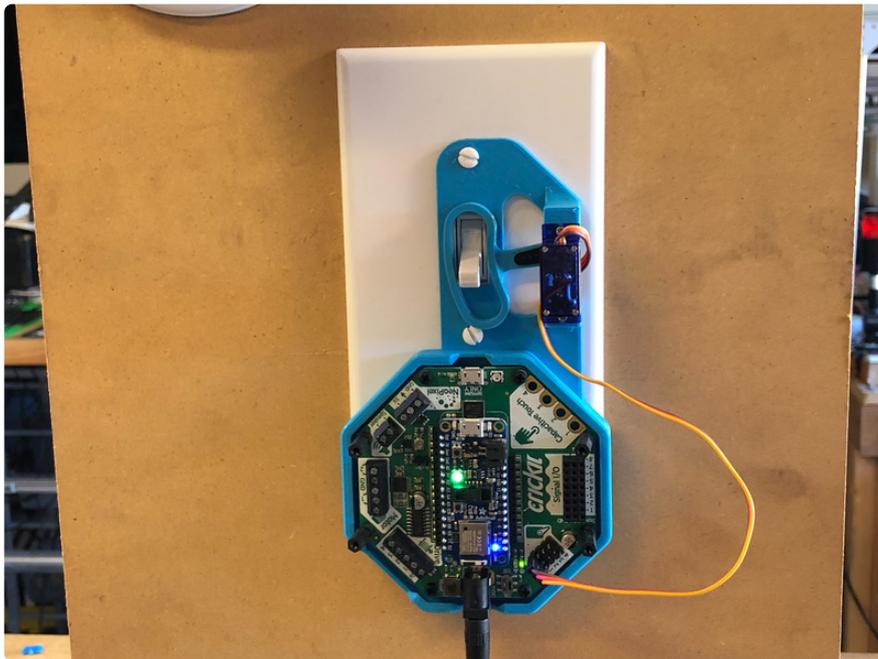
Unscrew the two screws from the switchplate.

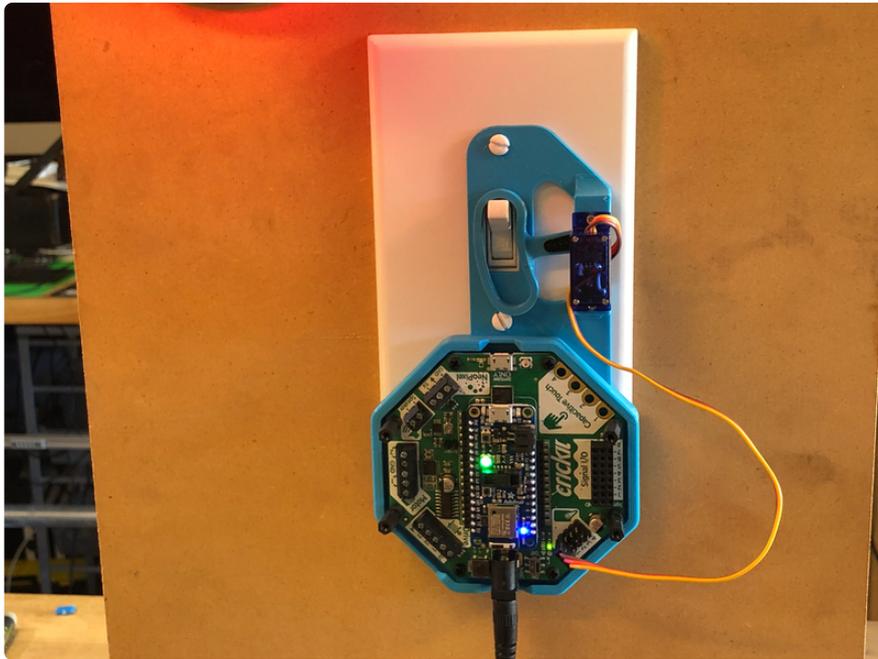
Place the mount over the plate and screw back in.

Plug the 5V 2A power supply into the wall and then into Crickit's DC power port.



It's ready for testing! Use the Adafruit Bluefruit app to connect to the device, then, use the up and down buttons in the control section to turn the servo up and down.





Cover

Once it's working to your satisfaction, you can neaten up the servo wiring and screw the cover into place.

