Physical Computing Final Project Log

Msc Creative Computing 2019 Physical Computing 1 w/ Phoenix Perry By Stuart Leitch

Background

The "Thinking Cap" is meant to be an attention-preservation device. It is a brain-computer-interface that monitors brain waves, and notes when there are periods of attention on one thing, or this attention changes to something else.

The Thinking Cap is based off of a Muse 2014 headset.

How to use:

How-to turn the Thinking Cap on: i am an amazing coder oh yeah oh yeah. 1. Make sure Bluetooth is running on Linux for the Ard: `ps aux | grep bluetoothd` 2. Power on the Neopixels. Just plug them in. 3. Power on the Feather. Just plug them in. If they're on while connecting to the Muse, the OSC server will connect to both. 4. Export the Slack URL to environment variables. 5. Run `local_connection_BLE.py`. Connect to the Feather. 6. Turn the Muse on by pressing the Circular on/off button. 6a. May need to re-pair the device. Go to the Bluetooth menu and click to connect, and confirm the number shared. 7. Start MuseLab with `MuseLab` and load the configuration file `muselab_configuration.json` Note: The axis is buggy on the second row at the moment. Will fix that in config file later. 8. On the "Outgoing" tab, enter in the IP and path of the OSC server (127.0.0.1 and 5005) plus UDP. Check Forward All Messages. 9. Once MuseLab is running, pair it with muse-io on Ubuntu: `muse-io --osc osc.udp://127.0.0.1:5000 --device 00:06:66:67:0B:0C` 9a. Add `--preset 14` if using the preset 14 configuration file (or any other.) 10. Run `local_osc.py`. This could be done at any point, really. 11. Think. Watch the BossBoxBot for reactions.

Expected outputs:

The server will output the following in a loop as it communicates with the BLE-enabled Thinking Cap.

```
Client state updated to:
{"connected": true, "focused": false, "hat_running": true, "last_reading": 1579023726.023477
Waiting for update
{"focused":false,"wearing":true,"userOverride":false}
{'focused': False, 'wearing': True, 'userOverride': False}
Server state updated to:
```

`{'focused': False, 'mentally_focused': False, 'wearing': True, 'hat_running': True, 'connect`

Preparations:

https://learn.adafruit.com/bluefruit-le-python-library/installation https://learn.adafruit.com/bluefruit-le-python-library/usage

Components used:

1. Adafruit Feather Bluefruit nRF52840 - 1
2. GP2Y0A21YK0F SHARP IR Analog Distance Sensor (10-80cm) - 1

Things I'd improve for this version.

1. Use a more near-range IR sensor.

Week 1 (2019/10/28 - 2019/11/3):

Challenge: Get Arduino and Muse talking to each other.

1. (+) Received the Muse EEG wearable from Sheldon.

2. (+) Charged the Muse and tested it using Muse's iOS app while working on something. There was a 5-minute meditation exercise as introduction to the hardware. Rather than meditate, I did some browsing online. Flipping through tabs and following links. Reviewing the brain wave graph after the conclusion of the 5-minute intro exercise showed I was mainly in the "calm" section which supposedly indicates focus.
3. (-) Attempted to link the Muse with my computer using the open source muse-lsl package. Failed because is only supports the Muse 2016 and later models. This model is in fact a 2014 model. (muse-lsl GitHub repo)
4. (-) Muse doesn't support or even provide their SDK anymore, except in special situations. The documentations site is down. Emailed them re: this project.
5. (-) Muse responded and didn't deign to make the SDK available for my research.
6. (+) Nothing on the internet can ever truly disappear. Found the last version of the SDKs from 2015 for Linux and Windows, along with old versions of the documentation.
7. (-) Windows version doesn't work, since it was pre-Windows 10. It will just stop receiving data without throwing an error.

```
bits/second: 4313       receiving at: 220.00Hz          dropped samples: 0
OSC error 10061: No osc.tcp://[host]:[port] end-point to connect to!
```

8. (+) Linux SDK version still works.
9. (-) Arduino won't talk to Linux at all. Sometimes it will connect for a mere moment, but then disappear from the ports.
10. (+) Further testing with the Muse app has shown that while actively having a dialogue with a reading, I tend to have a baseline in the Neutral area. Not Active, not Calm. While doing something more physical and immediate, such as removing a pesky fly from my airspace or going to get a water, it scores me in the Calm region. Good datapoints, if a bit vague since I can't link it to my own data processing yet.

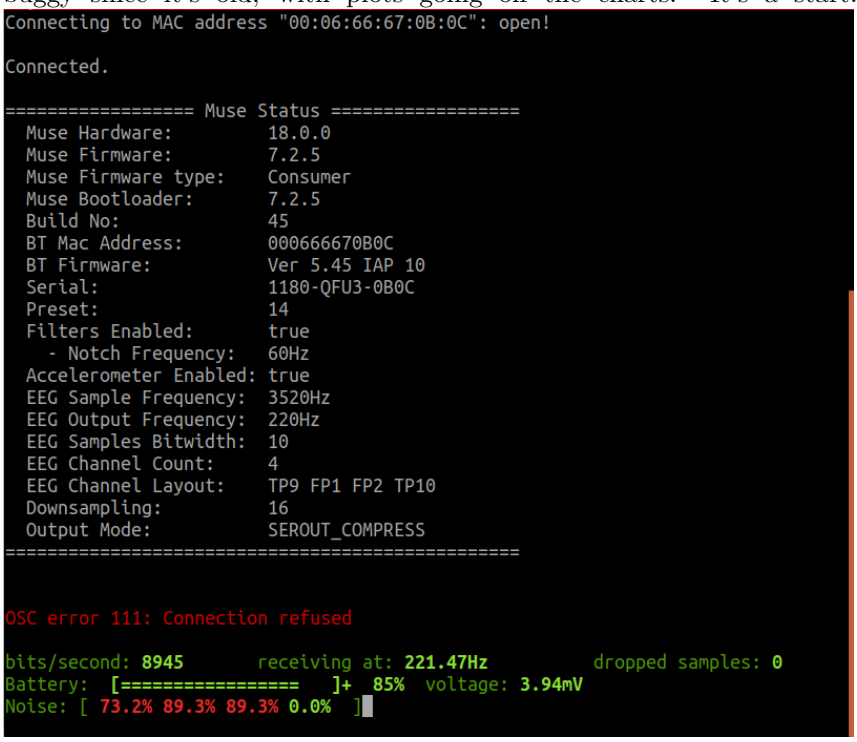
Week 2 (2019/11/4 - 2019/11/10):

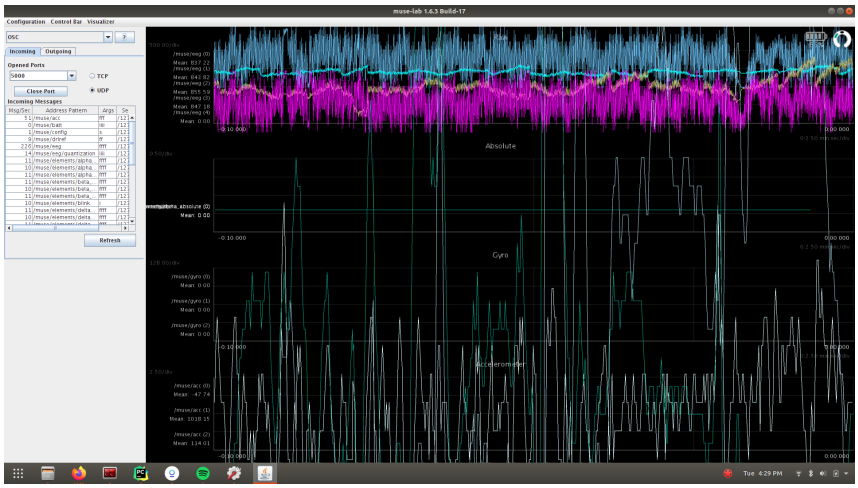
Challenge: Get Linux to recognize the Arduino, so the Muse and Arduino can pass info. Getting Windows to talk to the Muse seems a more difficult task.

1. (+) Tested with the Muse app further, while doing different activities. Contrary to my expectations, playing Zelda didn't elevate my readings to Active, even when solving puzzles. The game is familiar since I've played through it before, but the puzzle's solution I've since forgotten. This still confuses my hypothesis that Active is the area to watch.
2. (-) Arduino stopped talking to Windows, too. Likely as a result of trying

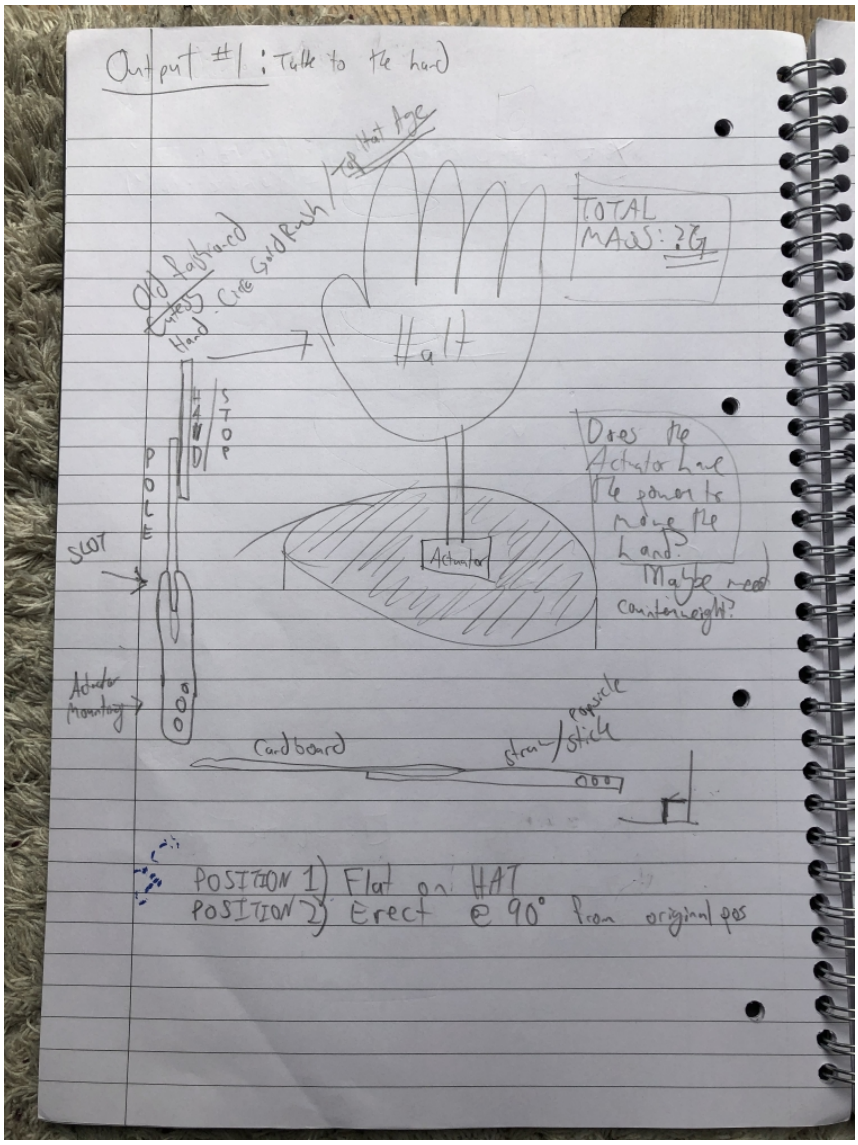
to upload a Sketch via Ubuntu.

3. (+) Thanks to this thread I was able to get the Arduino back in action using Windows.
4. (+) Thanks to this StackExchange Answer I was able to chase down a very annoying bug with Ubuntu that prevented serial ports from being properly assigned to the Arduino in Ubuntu.
5. (+) Combined with #4, by changing the programmer to the same as in Windows (Ubuntu has a different default), I was ablt to write a sketch from Ubuntu to Linux.
6. (+) Both devices talk to Linux. Can proceed with the meat.
7. (+) I started to configure the headset's data transfer protocols. The Muse connects via Bluetooth, which is received by a server running on my machine. The server then streams the data to a charting program, which takes in the 200+ parameters and plots them according to a configuration file. The charting program is buggy since it's old, with plots going off the charts. It's a start.





8. (+) Sketched out how the various inputs/outputs could be mounted on the

top hat. 
Output #1: Talk to the hand
Hat Axle
TOTAL MASS: ?G
Hat
Does the Actuator have the power to move the hand? Maybe need counterweight?
Actuator
SLOT
POLE
STOP
HAND
Actuator mounting
Cardboard
straw/stick
popsicle
POSITION 1) Flat on HAT
POSITION 2) Erect @ 90° from original pos

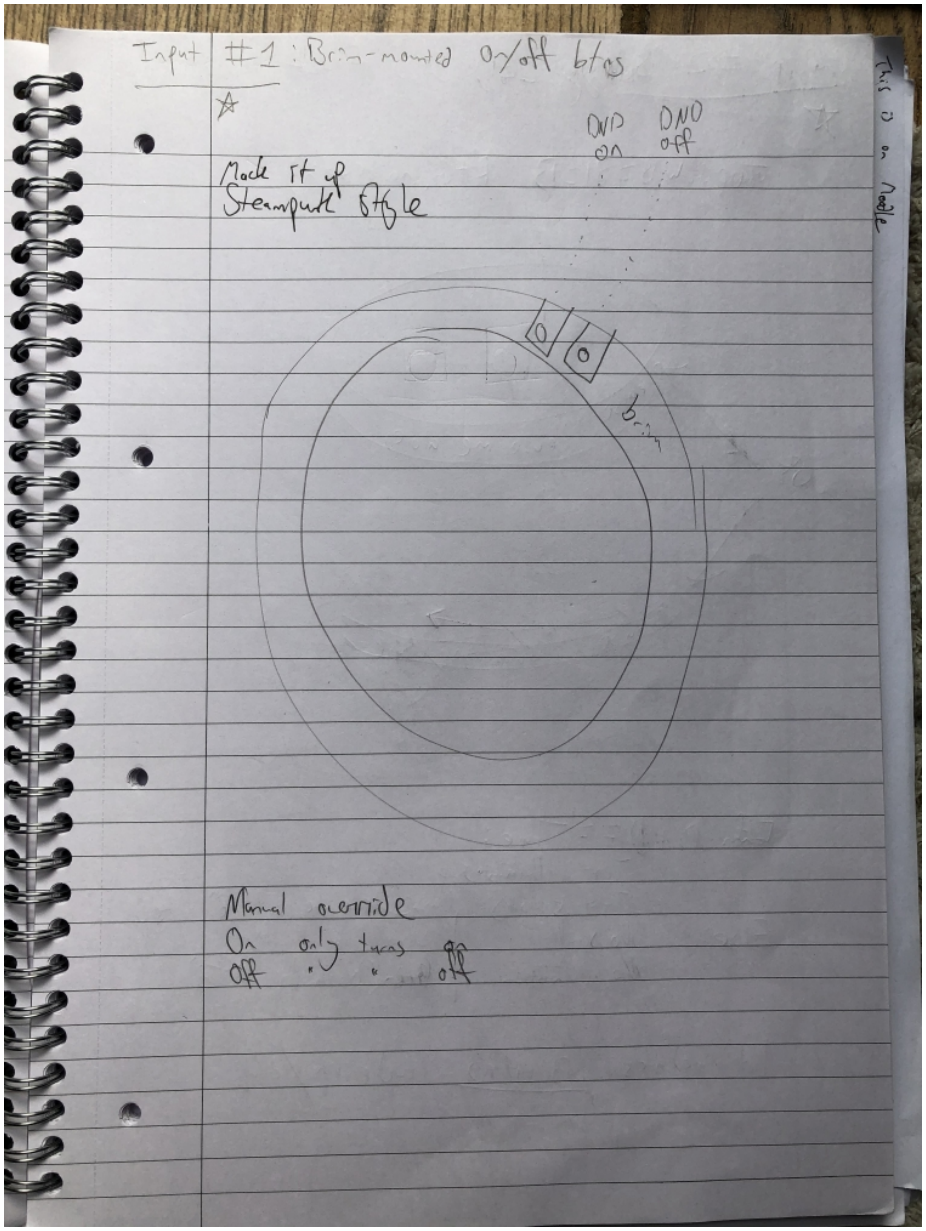


Figure 1: Sketch

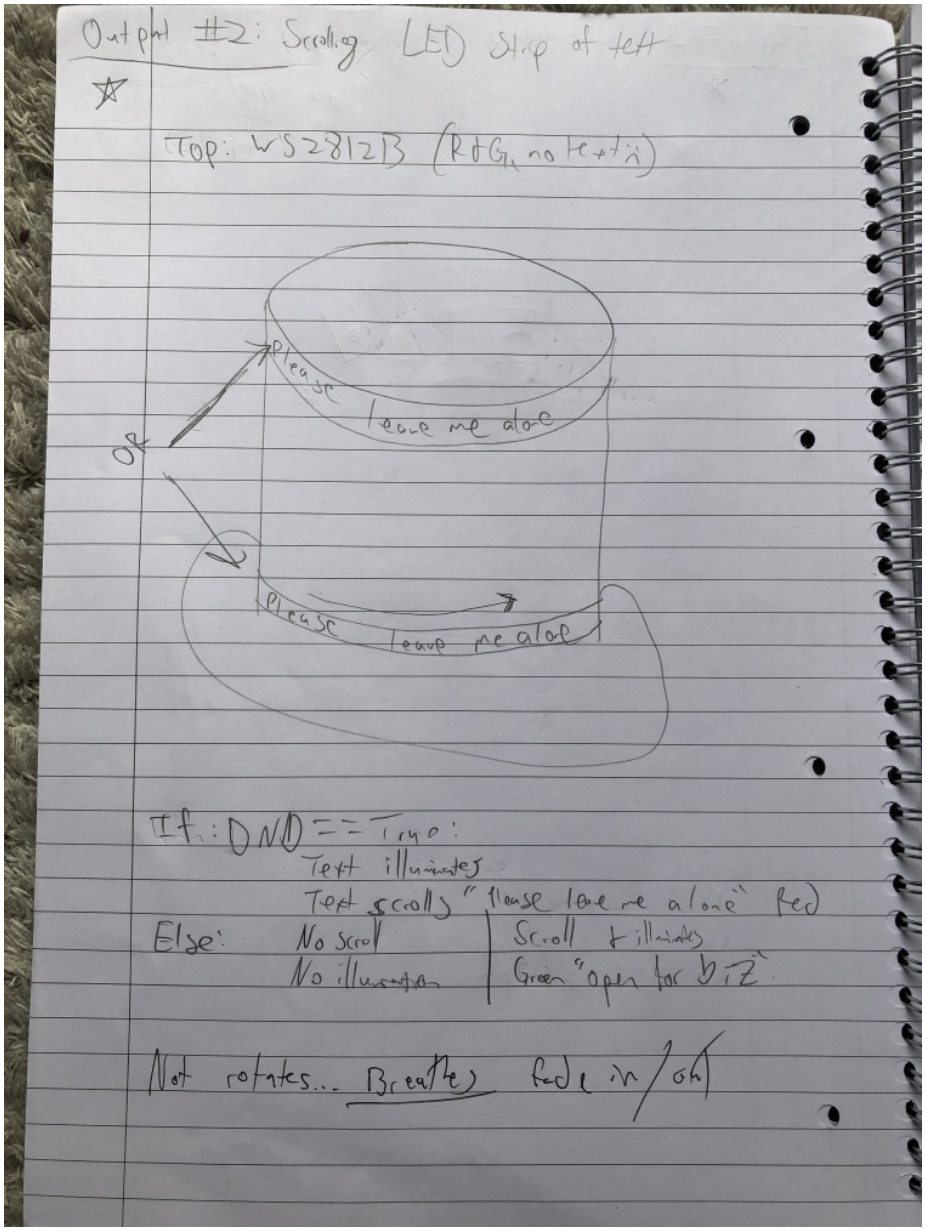


Figure 2: Sketch

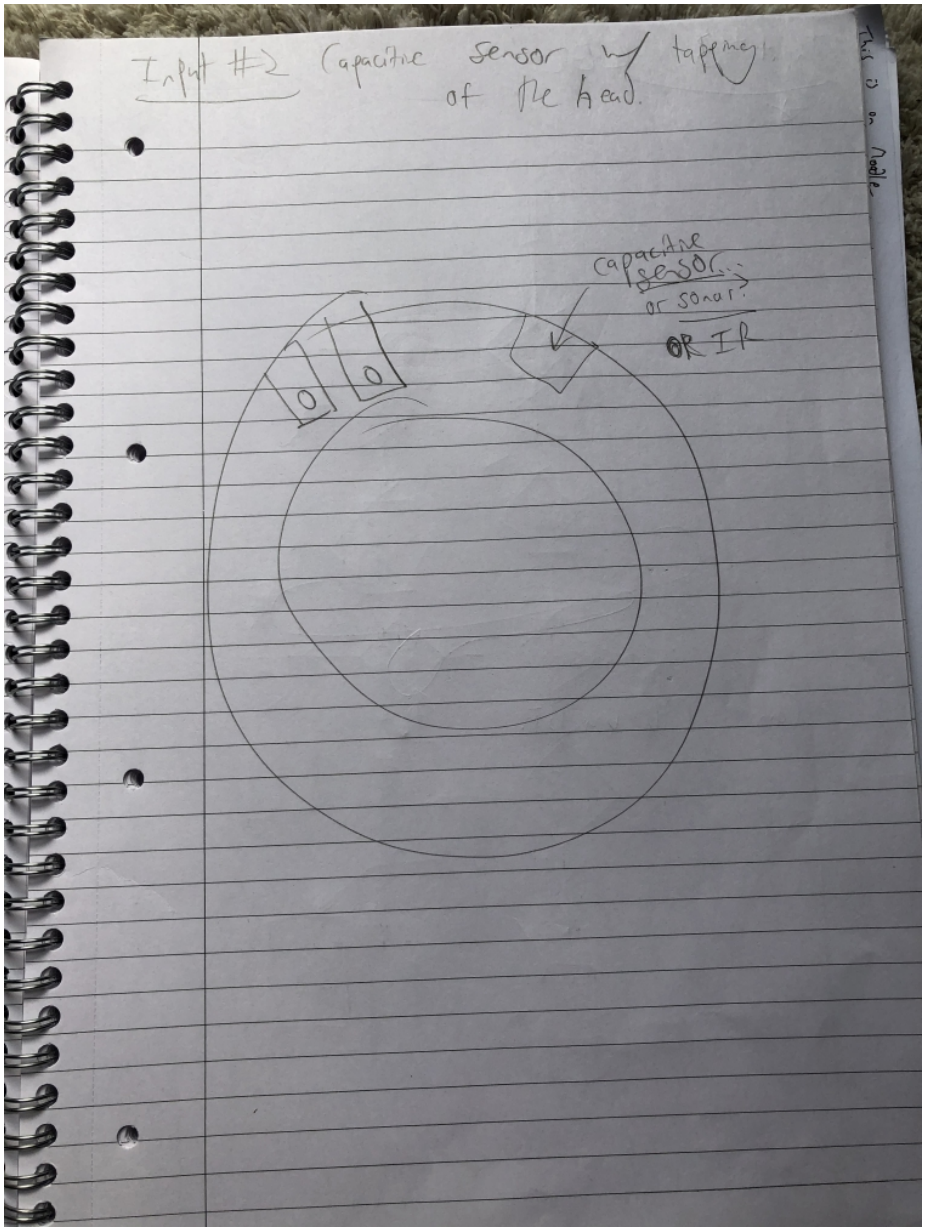


Figure 3: Sketch

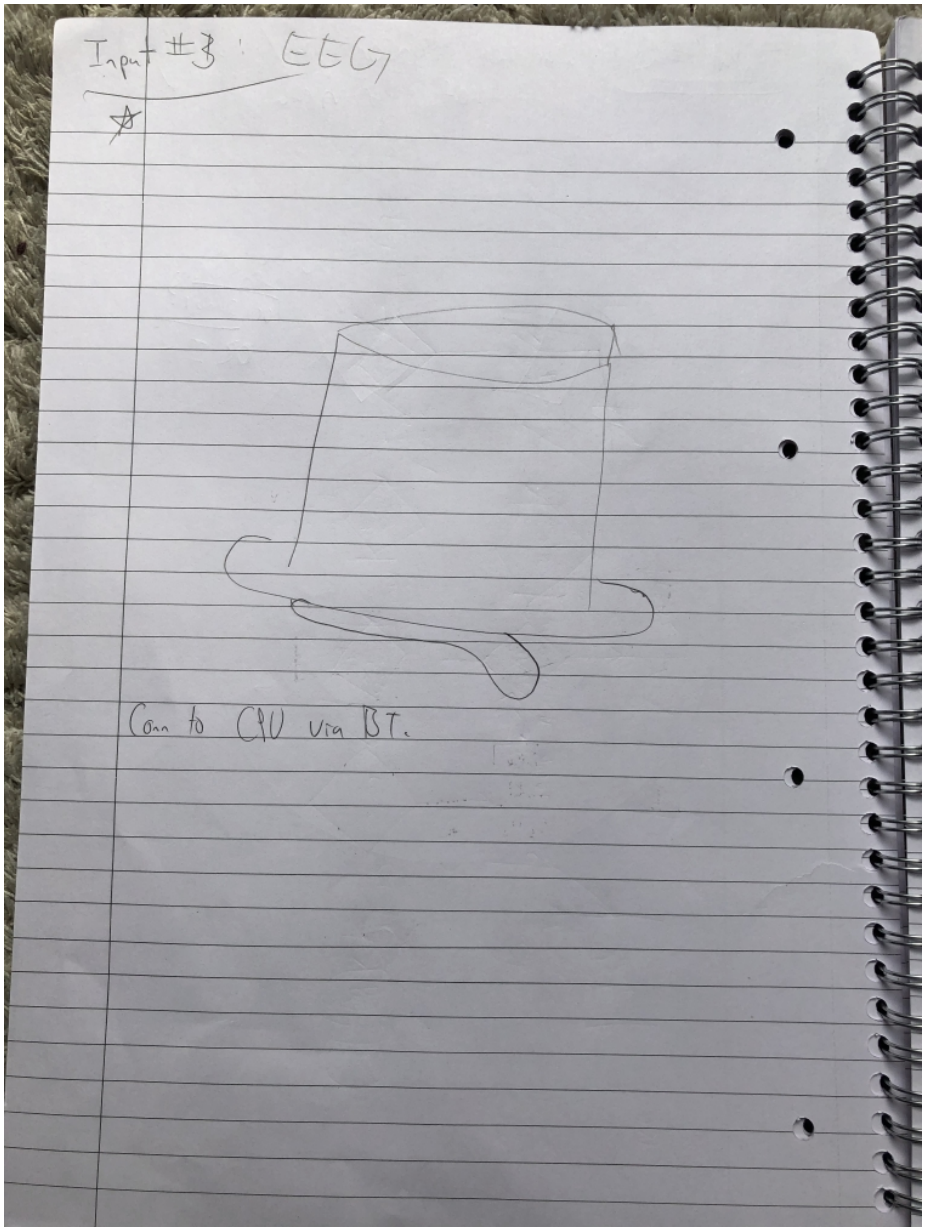


Figure 4: Sketch

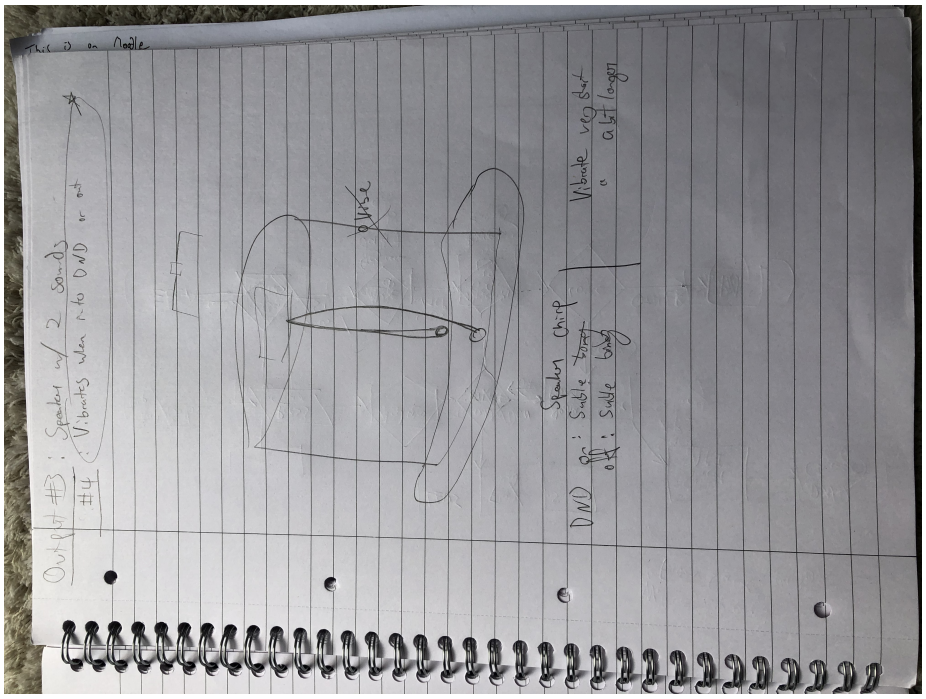
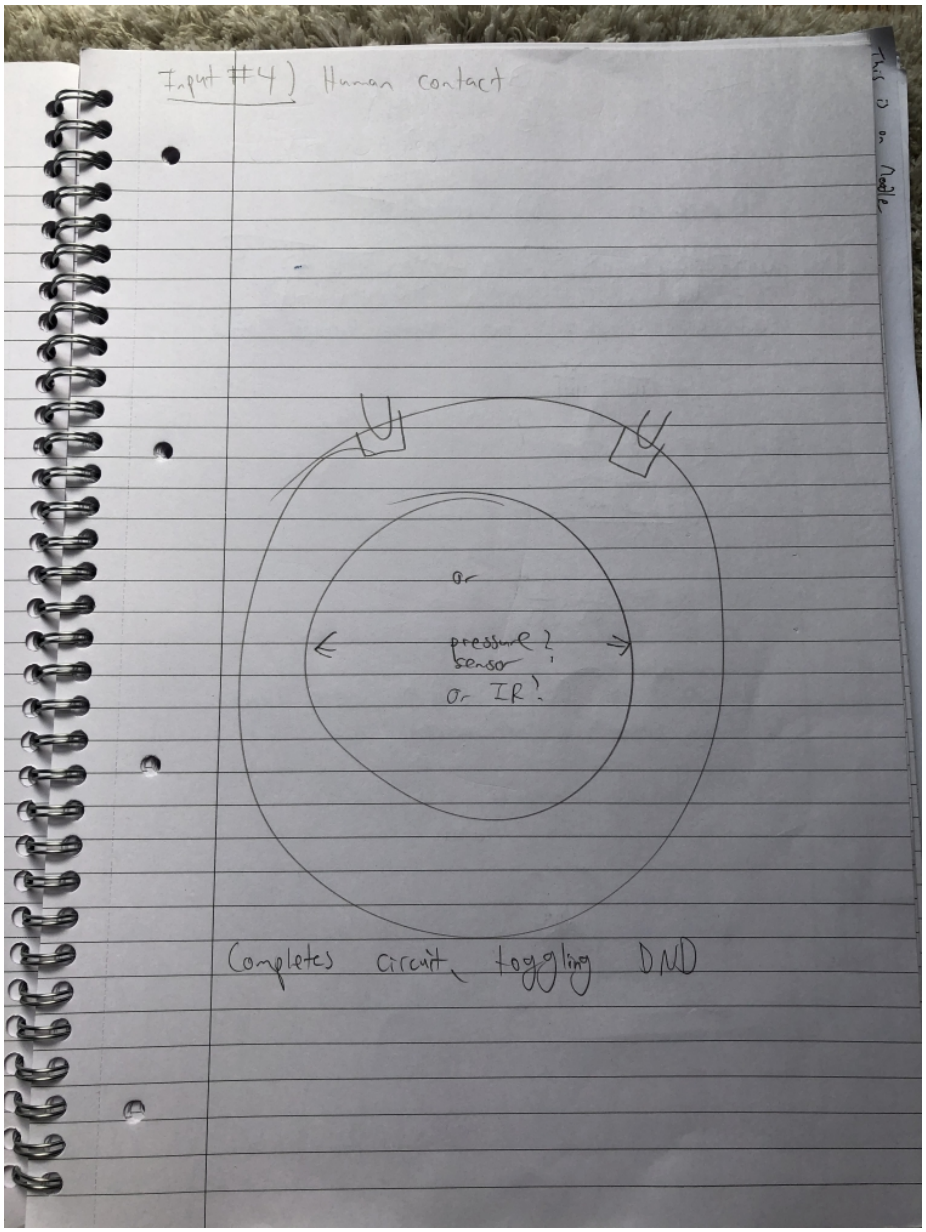
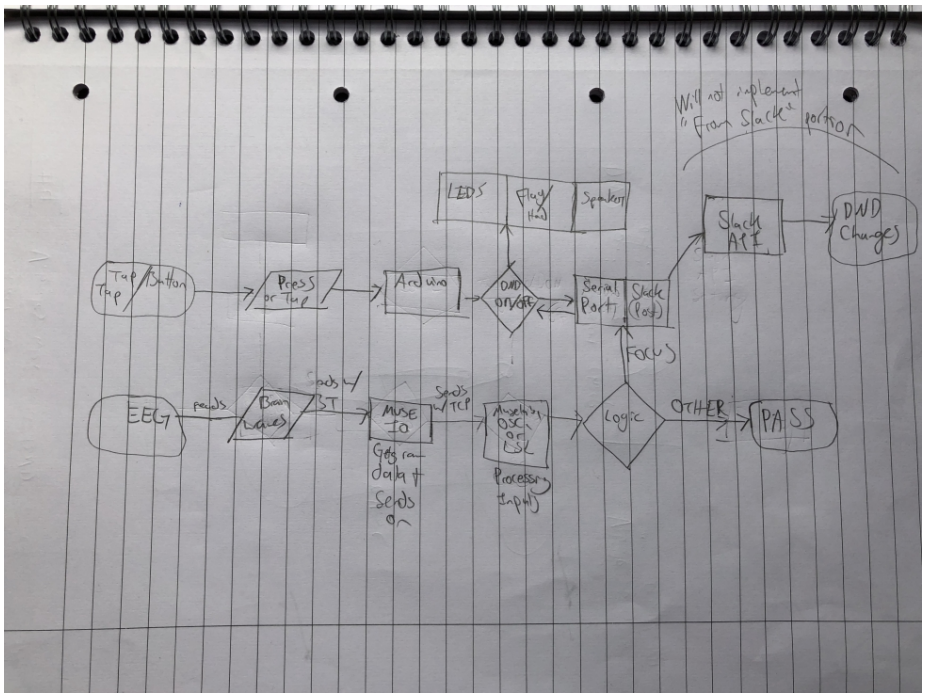


Figure 5: Sketch



9. (+) Sketched systems diagram for the device. Todo: digitize properly for legi-

bility. 

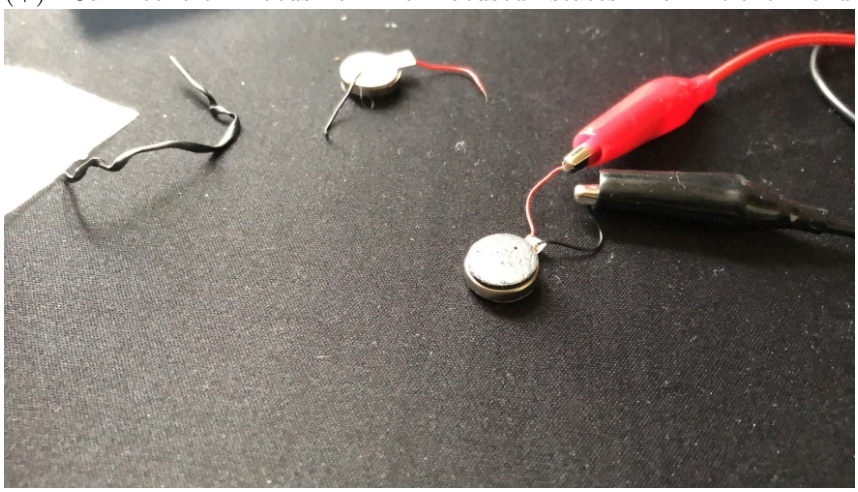
Week 3 (2019/11/11 - 2019/11/17):

Challenge: Order components. Get various bits and bobs controlled by the head (Arduino) and tail (Python)

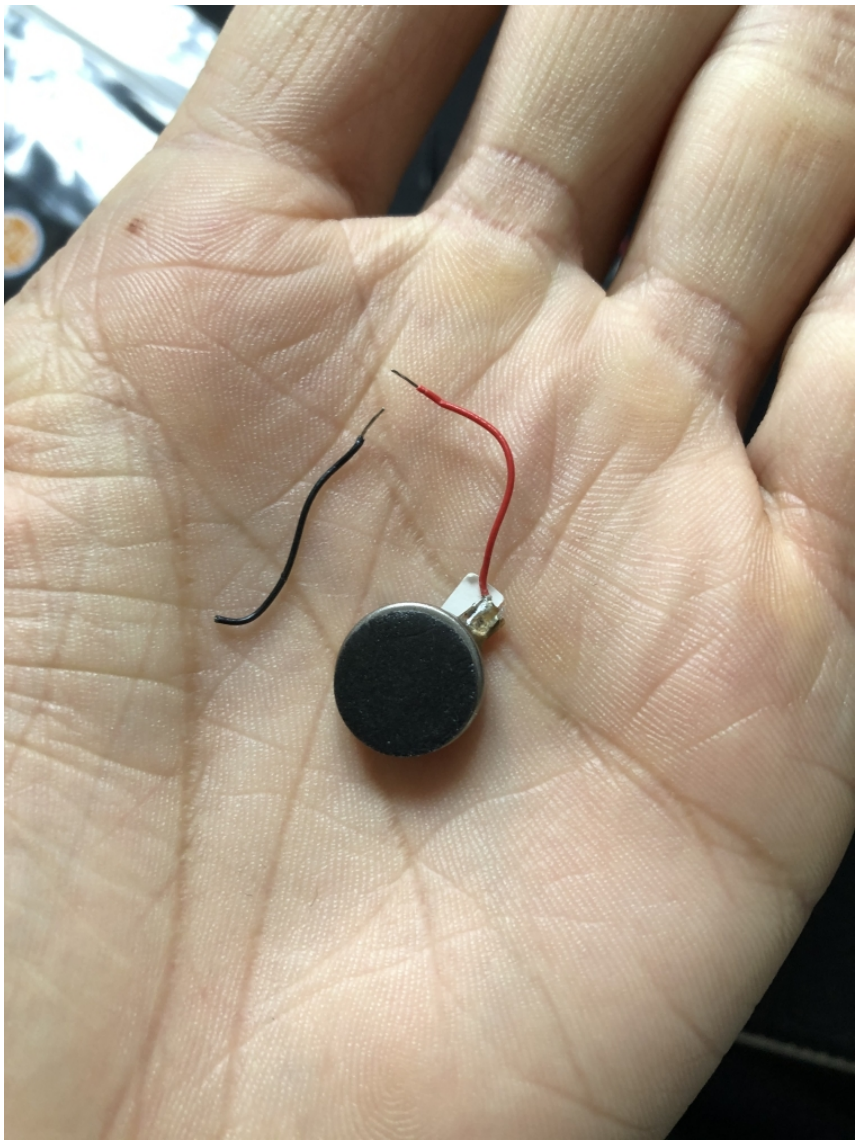
1. (+) Add feature to the hat: Senses when it is being worn.
2. (+) Ordered hat, vibrators, and IR emitters/sensors as well as a 1m LED light strip. Qutie surprised I was unable to source a secondhand top hat after Halloween.

3. (+) Received vibrators and IR emitters/sensors

4. (+) Working with the vibrators, created the beginnings of the communications systems between the Arduino for local control and Python over serial on my machine.
5. (+) Can control Focus or Non-focused states from either end.



6. (-) Still waiting on hat and LEDs to arrive.
7. (-) First time connecting a vibrator to the Arduino, it vibrated so hard that it tore its own wiring out. Luckily I bought 3 and only need 2. Still, not ideal to lose the spare so early. Will have to securely mount the wiring and vibrator so that the vibrations are transferred to the hat, and the vibrator doesn't move at all within its mounting.



Week 3 (2019/11/18 - 2019/11/24):

Challenge: Make an OSC server receive the output from the Muse headband's attention measurement.

1. (+) The hat and LEDs have arrived. Hat was smashed but "reflated" quite well.

2. (+) Introduced to the Adafruit Feather Bluefruit nRF52840 BLE device. Plan to use this instead of the Leonardo for true wireless.

Week 4 (2019/11/25 - 2019/12/1):

null

Week 5 (2019/12/2 - 2019/12/8):

1. (+) Test the fit of the Muse and the tophat. It's actually a better fit with the tophat than without.

Week 6 (2019/12/9 - 2019/12/15):

Challenge: Infrared sensor.

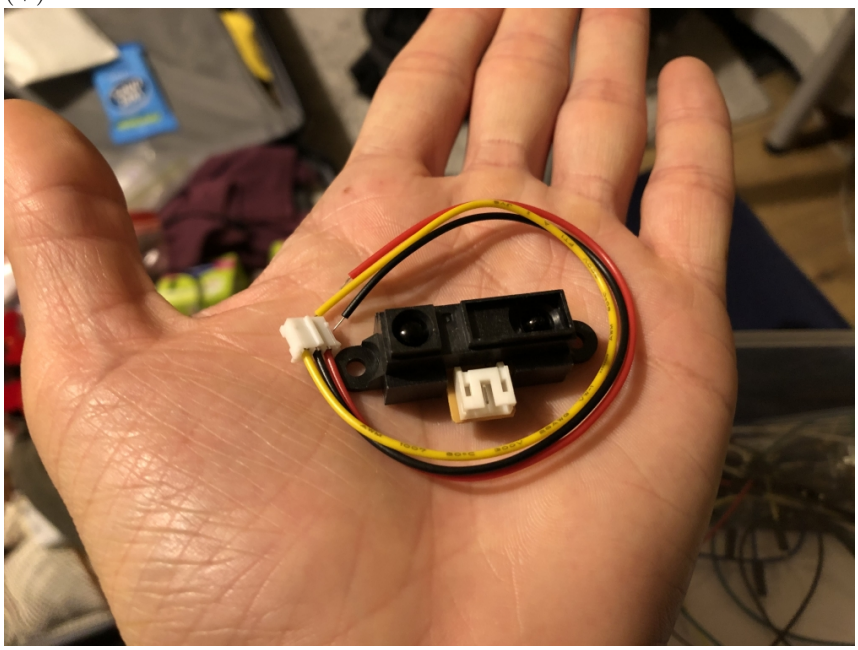
1. (-) Tested the battery with the Bluefruit. It blew up the Bluefruit. Maybe the polarity was reversed? Bought 2 more. One replacement. One backup. Also bought a battery straight from Adafruit. Good thing I'm headed to the USA to pick them up. Also good that I still have the Leonardo.
2. (+) Registered the Slack App.
3. (-) IR distance sensor not working to my liking. Purchased 2 more to test.
4. (+) The Neopixel at least turns on with the 9v battery. I don't dare use the one that fried the Bluefruit.

5. (=) Since the replacement Bluefruits won't reach me until next week, proceeding to develop the server further with only Serial and OSC.
6. (=) Idea. Since I have the potentiometer, maybe I can use that to control the brightness of the LED. https://www.instructables.com/id/Arduino-Potentiometer-Analog-Input-Tinkercad/ https://learn.adafruit.com/adafruit-neopixel-uberguide/arduino-library-use

Week 7 (2019/12/16 - 2019/12/22):

Challenge: Finish the syncing of state between focus and unfocused && the EEG reader and Arduino

1. (+) Bluefruit replacements arrive. Adafruit themselves now also have them in stock. Turns out, I didn't even need to pay $50 each... ah that's life.
2. (+) Finalized User Override button functionality on Ard.
3. (+) Finalized state sharing between Arduino and Machine
4. (+) Finalized EEG reader Concentration reading. Effectively, this means that now the EEG reader can control the Focused state on the Arduino - and thus all the lights etc. Server-side code is essentially done now.
5. (+) First new IR sensor has arrived. This is the better of the two.



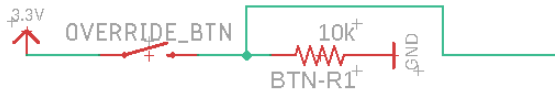
Week 8 (2019/12/23 - 2019/12/29)

Christmas break

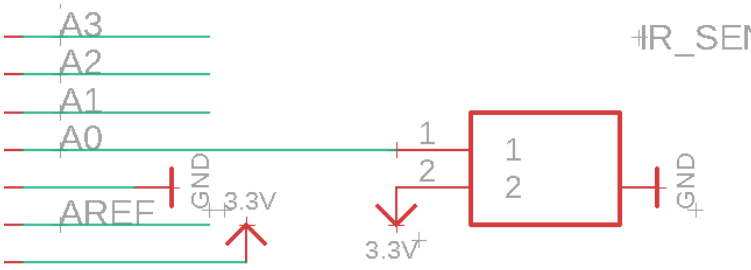
Week 9 (2019/12/30 - 2020/1/5):

Challenge: Get IR reader functional

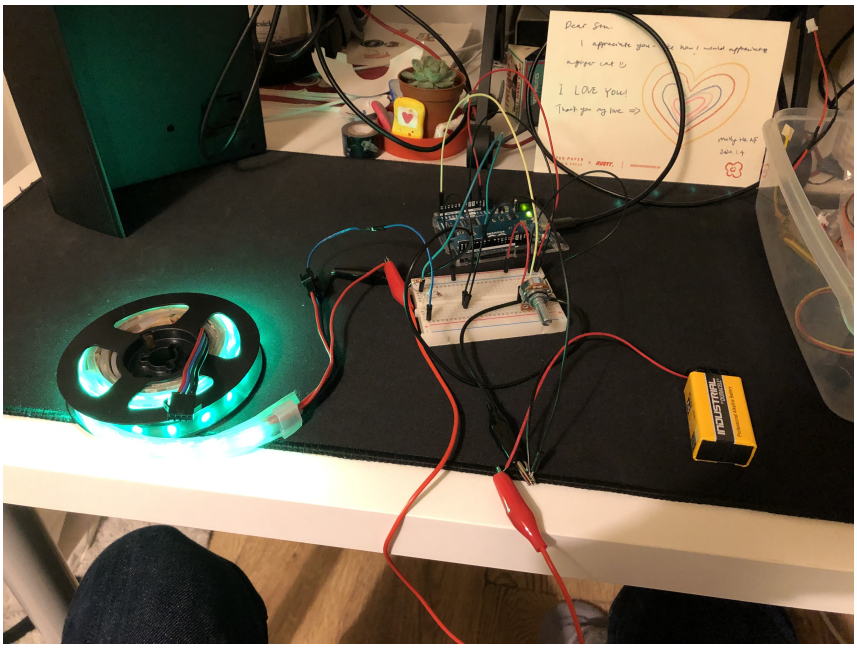
1. (+) Third kind of IR sensor arrived.
2. (+) Located nRF52 Bluefruit Feather open source circuit diagrams for Eagle on GitHub: https://github.com/adafruit/Adafruit-nRF52-Bluefruit-Feather-PCB
3. (+) Comparison between the 2 batteries confirmed that the Adafruit-sold LiPo battery has flipped polarity compared to the CoolComponents one. This is what fried the first Bluefruit. I will re-do the wiring and use it to independently power the Neopixel. I'm not sure which battery is "incorrect."



4. (+) First circuit drawn on Eagle. The override button!
5. (+) The Sharp IR sensor works well enough. It's meant to work on 5v, but it does get accurate enough readings on 3.3v to suit my purposes. Plus it has nice mounting holes for screws.



6. (+) Added IR sensor to Eagle schematic.
7. (+) Complete the Neopixels breathing animation code.
8. (-) Discover that the LiPo battery, despite its size, doesn't supply enough juice to color all the green LEDs I need. Have to rely on the 9V battery. It is, of course, very bright now.
9. (+) Added the Neopixels circuit to the Eagle diagram.
10. (+) Created a dimmer switch for the LEDs, using a potentiometer. Added



to circuit diagram.

11. (-) My Python packages manager got all messed up and I had to waste a couple days getting things back in order. This included some adafruit packages. Why can't every package just get along?
12. (-) Perhaps related to #11, Arduino IDE wouldn't find adafruit-nrfutil in $PATH. Naturally it worked fine in terminal. Worked fine a month ago. Took a day to get working again, with little idea of what went wrong. This was preventing me from compiling for the Feather.

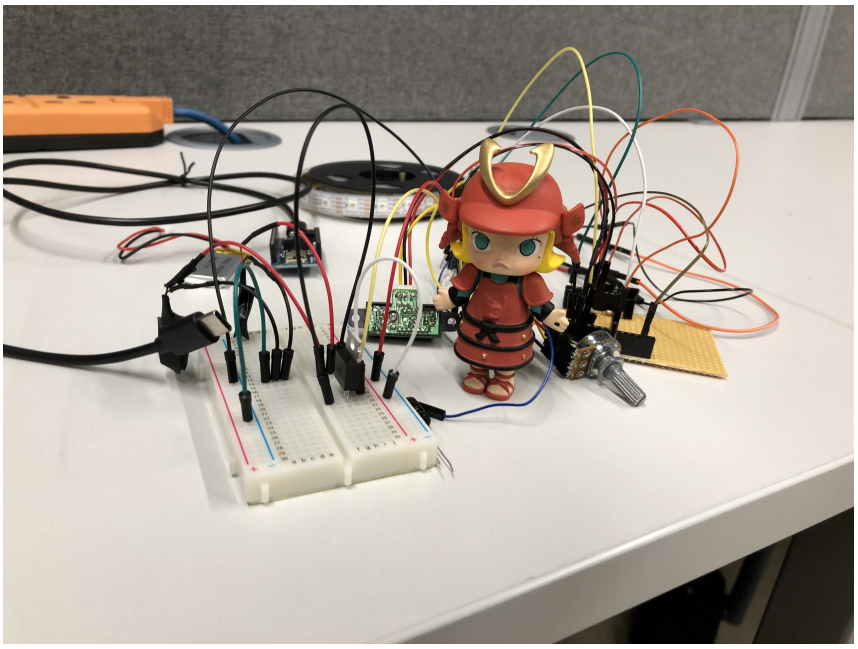
Week 10 (2020/1/6 - 2020/1/12):

Challenge: Get the whole thing functional

1. (-) Soldered on the green perf board. It's a major pain to make connections between the holes. I'm unable for now to get consistent current through what I draw.
2. (+) Switched to strip board. The copper that exists already is much better than my cobbled-together circuits. I also switched to using female headers on it, as Tom said they give a better quality of connection.
3. (+) Discovered current tophat not deep enough. Ordered another one!

Taller.

4. (+) Updated schematic to reflect removal of the 2x 220 ohm resistors. They're distorting the signal, rather than preventing distortion. I left them on the perf board, just will not use them. Instead I will connect them directly, as Tom said the Neopixels don't really need them.
5. (+) Final perfboard completed. It doesn't actually have that many components soldered to it. Didn't really need it.
6. (+) Tested the new battery. After a moment of shock (remedied by installing Blinky on the machine so I could actually see some response), it works!
7. (-) The current supplied over the GPIO pins on the Feather are insufficient to make the vibrators vibrate. 12mA max, 8mA recommended. The vibes work at about 20mA. Must find some way to make them go. There's enough voltage, at least.
8. (+) I tested using an NPN transistor linked to the main Arduino power sup-



ply. Works like a charm!

9. (-) No coin vibrators in Eagle's libraries. Including Adafruit and Sparkfun. Seems like a big thing to miss.
10. (+) Found a similar one from https://www.snapeda.com/home/ that works well enough.
11. (+) Updated vibration code to be non-blocking. Removed the delay and added milli() math instead. Split into 2 functions called separately.
12. (+) Installed the required libraries and connected to the Feather over Bluetooth. Followed this: https://learn.adafruit.com/install-bluez-on-the-raspberry-pi/installation
13. (+) Succeeded in sending strings over Bluetooth in both directions: client-> server-> client.
14. (+) Capacitors arrived for Neopixel (just in case they prove necessary)
15. (+) Made a test 3D print. It failed, the printer head keeps dragging filament and the object bends up from the plate. Still, pretty cool!
16. (-) Second 3D print failure. So bendy. And non-sticky. Weird. At least I got far enough this time to know I need to adjust all my holes by about 1mm wherever there's a hole.
17. (+) Can confirm that bluetooth sending and receiving of my JSON works. Random garbled characters come over the signal, though. Ñ ÅKéB4} Since it's at the end of the string, I can just throw them away and rebuild the JSON for loading.
18. (-) Infrared library doesn't compile for the Feather Express. Does for Leonardo. Problem in the library, not my code.
19. (+) In a true hacker move, I updated the IR library and now it works. Magic. Constant 8's, as intended. Changes committed to this repo.

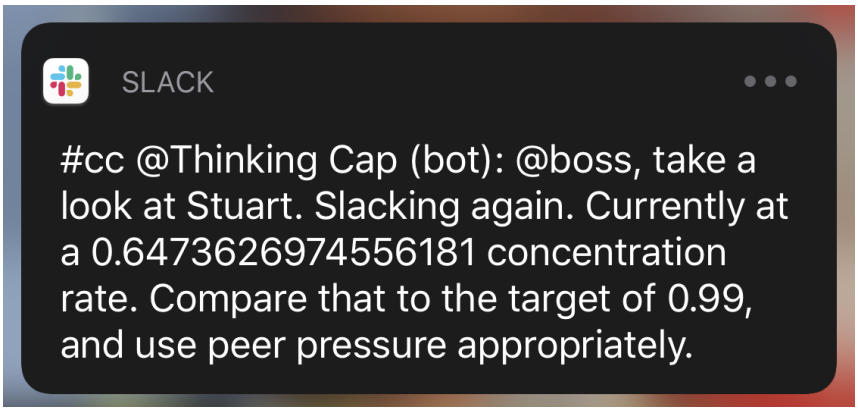
20. (-) Feather isn't recognized on any ttyACM port after being unplugged while on battery. How annoying to test. It "unfreezes" when disconnected.
21. (+) Bluetooth syncing of state now functional. Woo!
22. (-) Won't connect at the same time, if attempting to connect in the wrong order. Has to be: Feather first, brain scanner second. Doesn't seem to be any way to specify the device to connect to in the Adafruit package.
23. (-) Vibrator issues still. My breadboard version and what I soldered don't work the same. Perfboard loses the juice. Re-soldered and fixed. Includes the transistor.
24. (+) Added turn off the lights function for when hat isn't being worn. Everything working over serial, but BLE has some blocking.
25. (-) Concurrent connection still an issue. The main BLE thread doesn't release, so the OSC server which receives the data from the Muse never goes. Two loops that won't release.
26. (+) Split the BLE code and the OSC code and just having the OSC write a float between 0 and 1 to a file, and the BLE code reads it.
27. (+) Had a very hard to track down memory management issue with the JSON. But 'tis a solved thing, now.
28. (+) Had an even harder problem figuring out that blueart.readString is blocking for a whole second, and had to change to .read(), which required a whole rewrite of readState.
29. (+) All assembled. Just need to replace the NOW DEAD 9V for tomorrow.

Week 11 2020/1/13 - 2020/1/20

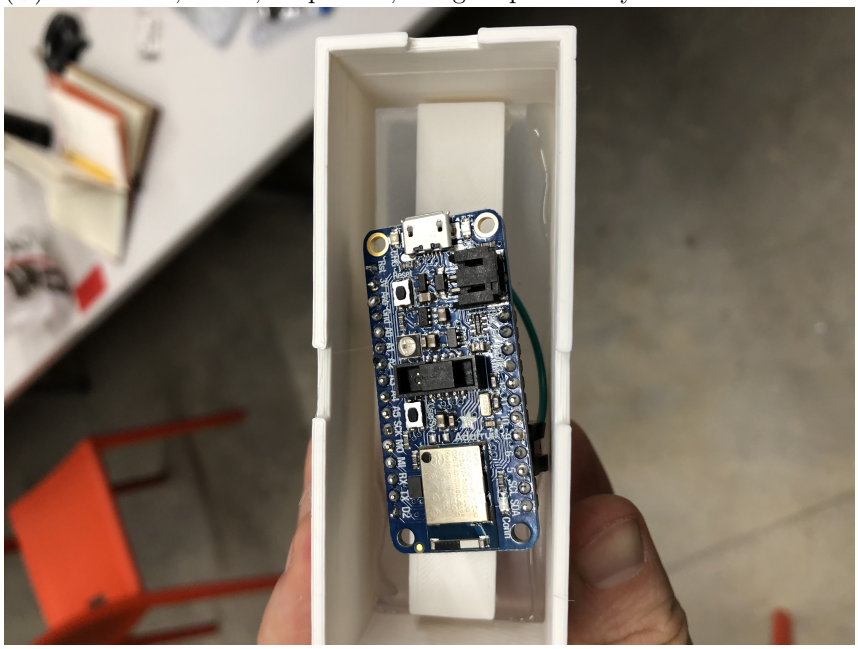
1. (+) Added error handling on server for EOFError. Unfortunately, discovered during the presentation. Great timing.
2. (+) Tried to reglue the Neopixels to the hat. It didn't take, so I just sewed it in.
3. (+) Neopixels working again. Didn't touch them. Naturally.
4. (+) Recorded a demo (thanks Mark) showing it working, including the

brain wave reading.

5. (+) Have some time, attempting (Slack integration)[https://api.slack.com/messaging/webhooks]. Registered this image to the app: *Lego man from 16bit.com*
6. (-) Accidentally committed the Webhooks URL to GitHub. Rookie mistake. That URL has been disabled and a new one made, which is now accessed through environment variables.



7. (+) Slack reactivated.
8. (+) Envisioned, coded, 3D printed, and gave personality to the BossBotBox.



9. (+) Added placeholder animation for BossBotBox for the video.
10. (+) Storyboarded the video.
11. (+) Discovered the culrpit behind the signal getting thrown for the Neopixels. The potentiometer ground keeps coming loose. Swapped for a tighter cable.

12. (+) Filmed everything and assembled, uploaded to YouTube. (Thinking Cap Presentation - Business Corp version)[https://youtu.be/MygsSMPjsX8].
13. (+) Revised presentation to match new Business Corp dystopian theme.
14. (+) Final product:

Final schematic: