

Vega Carlson - Group 165 - Spring '22, Week 1

Saturday, Jan 15th -- W:0.0 T:0.0
Sunday, Jan 16th -- W:0.0 T:0.0
Monday, Jan 17th -- W:0.5 T:0.5
Tuesday, Jan 18th -- W:0.5 T:0.5
Wednesday, Jan 19th -- W:0.5 T:0.5
Thursday, Jan 20th -- W:0.5 T:0.5
Friday, Jan 21st -- W:3.5 T:3.5

Vega Carlson - Group 165 - Spring '22, Week 2

Saturday, Jan 22nd -- W:0.0 T:3.5
Sunday, Jan 23rd -- W:0.6 T:4.1
Monday, Jan 24th -- W:0.6 T:4.1
Tuesday, Jan 25th -- W:2.8 T:6.3
Wednesday, Jan 26th -- W:4.4 T:7.9
Thursday, Jan 27th -- W:4.6 T:8.1
Friday, Jan 28th -- W:8.2 T:11.7

Vega Carlson - Group 165 - Spring '22, Week 3

Saturday, Jan 29th -- W:0.0 T:11.7
Sunday, Jan 30th -- W:0.0 T:11.7
Monday, Jan 31st -- W:0.9 T:12.6
Tuesday, Feb 1st -- W:0.9 T:12.6
Wednesday, Feb 2nd -- W:1.4 T:13.1
Thursday, Feb 3rd -- W:5.7 T:17.4
Friday, Feb 4th -- W:8.0 T:19.7

Vega Carlson - Group 165 - Spring '22, Week 4

Saturday, Feb 5th -- W:0.0 T:19.7
Sunday, Feb 6th -- W:0.0 T:19.7
Monday, Feb 7th -- W:0.0 T:19.7
Tuesday, Feb 8th -- W:0.0 T:19.7
Wednesday, Feb 9th -- W:0.5 T:20.2
Thursday, Feb 10th -- W:5.3 T:25.5
Friday, Feb 11th -- W:8.0 T:28.2

Vega Carlson - Group 165 - Spring '22, Week 5

Saturday, Feb 12th -- W:0.0 T:28.2
Sunday, Feb 13th -- W:0.0 T:28.2
Monday, Feb 14th -- W:0.8 T:29.0
Tuesday, Feb 15th -- W:0.8 T:29.0
Wednesday, Feb 16th -- W:1.3 T:29.5
Thursday, Feb 17th -- W:4.1 T:32.3
Friday, Feb 18th -- W:8.0 T:36.2

Vega Carlson - Group 165 - Spring '22, Week 6

Saturday, Feb 19th -- W:0.0 T:36.2
Sunday, Feb 20th -- W:0.0 T:36.2
Monday, Feb 21st -- W:1.3 T:37.5
Tuesday, Feb 22nd -- W:4.3 T:40.5
Wednesday, Feb 23rd -- W:4.8 T:41.0
Thursday, Feb 24th -- W:4.8 T:41.0
Friday, Feb 25th -- W:8.0 T:44.2

Vega Carlson - Group 165 - Spring '22, Week 7

Saturday, Feb 26th -- W:0.0 T:44.2
Sunday, Feb 27th -- W:3.1 T:47.3
Monday, Feb 28th -- W:3.1 T:47.3
Tuesday, Mar 1st -- W:4.8 T:49.0
Wednesday, Mar 2nd -- W:6.6 T:50.8
Thursday, Mar 3rd -- W:7.6 T:51.8
Friday, Mar 4th -- W:8.1 T:52.3

Vega Carlson - Group 165 - Spring '22, Week 8

Saturday, Mar 5th -- W:2.1 T:54.4
Sunday, Mar 6th -- W:4.3 T:56.6
Monday, Mar 7th -- W:4.3 T:56.6
Tuesday, Mar 8th -- W:4.3 T:56.6
Wednesday, Mar 9th -- W:4.9 T:57.2
Thursday, Mar 10th -- W:6.7 T:59.0
Friday, Mar 11th -- W:8.3 T:60.6

Mar 12-18 -- Spring Break

Vega Carlson - Group 165 - Spring '22, Week 10

Saturday, Mar 19th -- W:0.0 T:60.6
Sunday, Mar 20th -- W:0.0 T:60.6
Monday, Mar 21st -- W:0.2 T:60.8
Tuesday, Mar 22nd -- W:0.2 T:60.8
Wednesday, Mar 23rd -- W:1.3 T:61.9
Thursday, Mar 24th -- W:2.6 T:63.2
Friday, Mar 25th -- W:8.0 T:68.6

Vega Carlson - Group 165 - Spring '22, Week 11

Saturday, Mar 26th -- W:0.0 T:68.6
Sunday, Mar 27th -- W:0.0 T:68.6
Monday, Mar 28th -- W:3.0 T:71.6
Tuesday, Mar 29th -- W:3.0 T:71.6
Wednesday, Mar 30th -- W:4.0 T:72.6
Thursday, Mar 31st -- W:5.5 T:63.2
Friday, April 1st -- W:8.0 T:74.1

Vega Carlson - Group 165 - Spring '22, Week 12

Saturday, Mar 26th -- W:0.0 T:74.1
Sunday, Mar 27th -- W:0.0 T:74.1
Monday, Mar 28th -- W:0.0 T:74.1
Tuesday, Mar 29th -- W:0.0 T:74.1
Wednesday, Mar 30th -- W:2.7 T:76.8
Thursday, Mar 31st -- W:4.2 T:78.3
Friday, April 1st -- W:8.0 T:82.1

Vega Carlson - Group 165 - Spring '22, Week 1

Saturday, Jan 15th -- W:0.0 T:0.0

0.0 hr -- No Progress today

Sunday, Jan 16th -- W:0.0 T:0.0

0.0hrs -- No Progress today

Monday, Jan 17th -- W:0.5 T:0.5

0.0hr - No progress today - MLK day

Tuesday, Jan 18th -- W:0.5 T:0.5

0.5hr - Met with the group to quickly recap where we were at and do some organizational planning, as well as oggle our new PCBs and check them over real quick.

Wednesday, Jan 19th -- W:0.5 T:0.5

0.0hr - No progress today

Thursday, Jan 20th -- W:0.5 T:0.5

0.0hr - no progress today

Friday, Jan 21st -- W:3.5 T:3.5

3.0hr - Soldered the 3x10 keyboard to a 32u4 and flashed it with QMK firmware. I'm not currently happy with the soldering job, and I have yet to get the LED matrix in place, but at least every button registers and it can *technically* be used now

Week ending 1/21 total = 3.5 hours

Vega Carlson - Group 165 - Spring '22, Week 2

Saturday, Jan 22nd -- W:0.0 T:3.5

0.0 hr -- No Progress today

Sunday, Jan 23rd -- W:0.6 T:4.1

0.6hrs -- Meeting with group

Monday, Jan 24th -- W:0.6 T:4.1

0.0hr - No progress today

Tuesday, Jan 25th -- W:2.8 T:6.3

2.2hr - SSH'd into the pi (After fixing some annoying issues with networking, which ate some time) and ran a `sudo apt-get update` and `sudo apt-get upgrade`. Following that, I got the dev environment working again, and then went to work on code. With the fresh eyes of after-break, I was quickly able to spot some bugs and now there's reliable triggering of the digital drums following the sequence. Still a long way to go, but it's looking good, plus flipping steps live with the keyboard is working now as well, albeit it feels a tad sluggish, as it takes a long time to register button presses. I think I'll be able to remedy this once I'm running the custom keyboard firmware for real, but before I do that I need to re-do the soldering job from last week (I used far too thick of wires, which doesn't leave room for the LED matrix) and add the LED matrix. I'd rather focus on code for now though

Wednesday, Jan 26th -- W:4.4 T:7.9

0.5hr - Meeting. See minutes.

1.1hr - Worked on RP2040 main boards

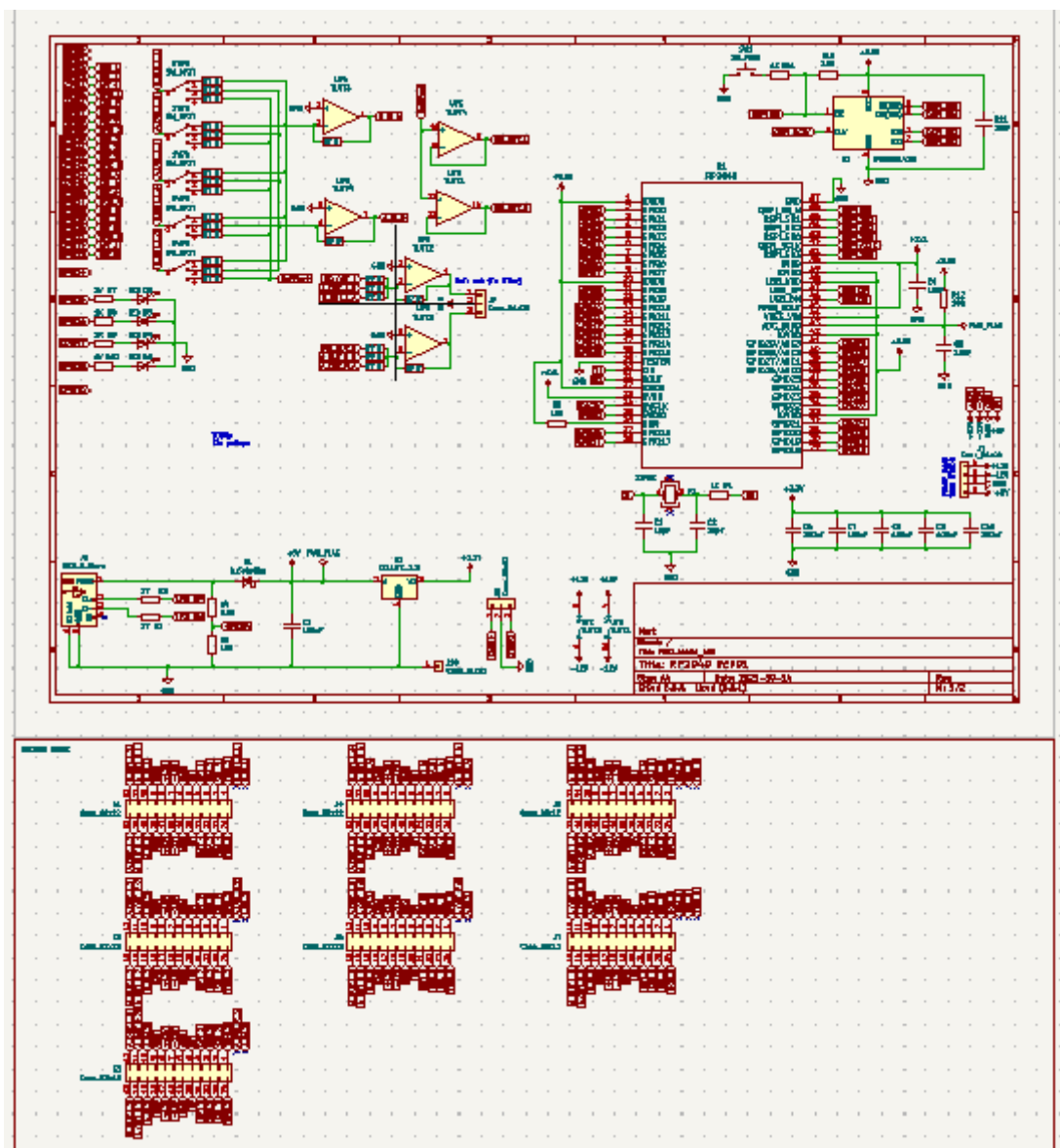
Thursday, Jan 27th -- W:4.6 T:8.1

0.2hr - Talked with group about the structure of the effects bus system to make sure everyone was on the same page

Friday, Jan 28th -- W:8.2 T:11.7

0.6hr - Meeting, see minutes

2.1hr - KiCad work on RP2040 / Mixer board. This was restarted from Wednesday, using the existing KiCad file Mark tossed me, as with that I can be certain the parts can be obtained, via Mark if all else fails. Doing some digging into the RP2040, mixers, buffers, etc. to make sure things are reasonable. I did discover the rp2040 only has 2, multiplex, i2c ports, which should be fine, I'll just have to loop through and turn on/off the particular pins I want associated with each port and drop the data that way. I should still be able to do this at high enough rate. Board is coming along:



0.9hr - Code + work in purr-data to try to get multiple drum tracks working. I can get two tracks working, but when they both try to play at once I only hear one. I need to investigate further.

Week ending 1/21 = 8.0 hours, total = 11.7 hours

Vega Carlson - Group 165 - Spring '22, Week 3

Saturday, Jan 29th -- W:0.0 T:11.7

0.0 hr -- No Progress today

Sunday, Jan 30th -- W:0.0 T:11.7

0.0hrs -- No Progress today

Monday, Jan 31st -- W:0.9 T:12.6

0.9hr - Started work on product spec

Tuesday, Feb 1st -- W:0.9 T:12.6

0.0hr - No Progress today

Wednesday, Feb 2nd -- W:1.4 T:13.1

0.5hr - Meeting. See minutes.

Thursday, Feb 3rd -- W:5.7 T:17.4

1.1hr - Continued work on Product spec

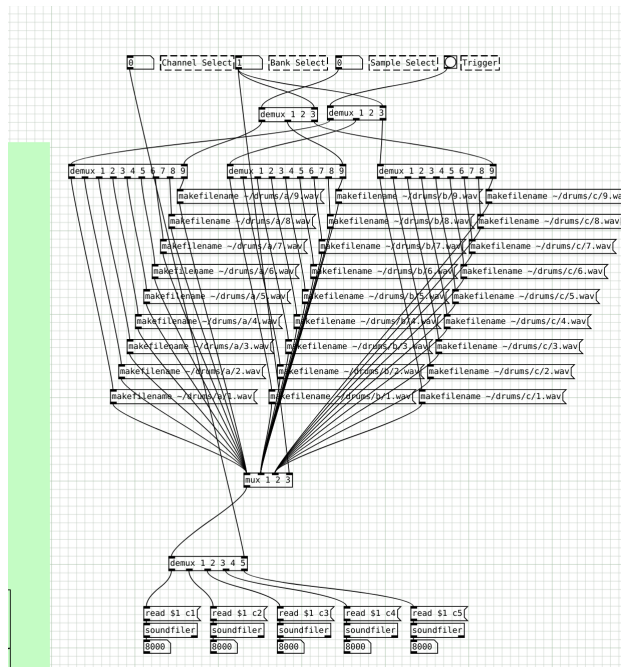
1.5hr - PD work. I decided to re-start with [someone's existing patch](#), as I was having annoying issues with my sample playback patch with occasional clicks. This will need credit in the project, but it is MIT licenced. I also found [This patch](#) which might help with the mess of needing to switch between samples later. Then went on to try playing things back with the pi and python script again, and something is definitely weird - it works fine when triggered with a hardware MIDI controller from my desktop, yet despite the code seemingly being logically correct the same MIDI output channel is being used by all 5 sequencer tracks? Will need to spend time debugging.

1.7hr - Found the bug- python sometimes has issues with reording instructions, espically those from libraries. A message was being sent before it had the channel information actually associated with it, so I just packed that channel info in sooner and voilà, it's working. Now I need to expand the patch, make it pitch and velocity sensative, add effects, and step lock those effects, but this is an actually functional sequencer! It's truly palying back the programmed in steps and working! Then got to work on the PD patch again, took more work than it should have, but I have pitch and velocity input working again.

Friday, Feb 4th -- W:8.0 T:19.7

0.5hr - Meeting, see minutes

1.5hr - More work in Purr-Data. I made a sample selection patch that, while does work, feels a lot like when a beginner programmer uses a ton of `if` s because they don't know about `for` . Still, it works when tested with an external sequencer.



.3hr - Made a little start up script for the pi and tested the new patch on the pi. Script works, updated patch doesn't. Fortunately I *think* I know why, but I can deal with that next week.

Week ending 2/05 = 8.0 hours, **total** = 19.7 hours

Vega Carlson - Group 165 - Spring '22, Week 4

Saturday, Feb 5th -- W0.0 T:19.7

0.0 hr -- No Progress today

Sunday, Feb 6th -- W:0.0 T:19.7

0.0hrs -- No Progress today

Monday, Feb 7th -- W:0.0 T:19.7

0.0hrs -- No Progress today

Tuesday, Feb 8th -- W:0.0 T:19.7

0.0hr - No Progress today

Wednesday, Feb 9th -- W:0.5 T:20.2

0.5hr - Meeting. See minutes.

Thursday, Feb 10th -- W:5.3 T:25.5

4.9hr- First up, when I left off last week the PD script worked on my desktop, but didn't output sound on the pi, so I went to trace that down first. After more debugging than I care to admit, only to discover that I had a master low pass filter set to a cut off of 0Hz, and sample play back is working. It's not the most spectacular thing yet, but it works

Then I needed to work on the UI a bit. I've found having the overview of each pattern's steps is actually hugely helpful, and I think will remain so even when we get the LEDs in the keyboard, so I set up a toggle between the two views. While doing UI, I decided to impliment the pattern lock toggle as well, and then went for getting BPM control, as it was the next biggest thing on the UI... however...

BPM control is going to be a massive problem. Even without any delay/sleeping/etc. the max BPM as-is (without actually measuring anything) feels like 60-ish, which is way to slow. So, I'll need to see how I can shave cycles or thread some things, otherwise this is only going to get worse.

As a quick & hacky solution, I'm going to try using cython (python → c) to see if it gives me a good performance bump... it doesn't.

```
1 cython -3 --embed main.py
2 mv main.c test.c
3 gcc -O3 -o test test.c `pkg-config --cflags --libs python3`
```

... trying `pypy3` instead. Reading, it should, but it should also hurt startup time, that is assuming it works at all.

... it isn't working. Something weird is happening with pypy and numpy, and I can only assume that numpy is being used by one of the other libraries I'm using, as I'm not directly and pypy otherwise works fine. Heck. I have numpy reinstalling (awkwardly slowly? I think runnin gpip made it completely recompile?) in the hope that will work. Otherwise my best bet is a pi-4, which is un-obtainium right now.

I did notice that I'm currently on a 32-bit kernel, but I'm not even going to see if switching to 64 bit helps at all, given the pretty low impact it seems to make: <https://www.sevarg.net/2021/12/18/raspberry-pi-32-bit-vs-64-bit/>

It *might* be practical to do the re-write in C++, but that'd be pushing my C++ skills, and I'd still need to interface with the 2 display hats - both of which only have python libraries available - **and** I'd have to re-solve a bunch of other issues, so I don't think there's time.

Okay, after more reading, it appears that pypy doesn't use the normal library install location. This requires some awkward workarounds, but I think I can make it all work. I basically just have to get every library I'm using... and all of their dependencies, re-installed. I'm starting to regret my choice of working in python now....

Yeah, nope. Numpy and pypy aren't going to get along, which rules out one of my libraries right away, and I haven't even tried to fix the others. I think my only option is to optimize the normaly cython implimention and my code as best I can.

I tried converting the code to work with cython, adding type hints and what not, but the performance gains weren't steller. I also tried profiling with python's `cProfile` and nothing stood out. Othen than throwing a pi-4 at the problem, I'm pretty much out of ideas. Today was highly frustrating.

Did some quick digging into the Unicorn and Microdot Phat's python source. In theory adapting to C(++) if that's the only option should be doable, but I'd still much rather not.

Friday, Feb 11th -- W:8.0 T:28.2

0.5hr - Meeting. See minutes.

2.2hr - Switched over to the pi-4 to test if it'll save my bacon, that required some re-setup. It's definitely faster, but still not up to 120BPM, which is my target (as that's what most drum machines default to). It's currently at ~100BPM though, so significantly better. Doing some work to see if I can find what in the code is eating all the cycles, added a timing function for better anysis.

Then did some reading and discovered that python threading library [may not have actually been helping me any](#), so switched over to the `multiprocessing` library (ughh.) ... and it appears the overhead actually makes things worse. Go figure.

Change	BPM (roughly)
No multithreading (Baseline)	98.25
Multiprocessing the display updates	92.5
No display call (Obviously not a solution)	50,000+

... huh. well, I was intending to do more testing but that's pretty conclusive about where the problem lies, even if I don't have a fix yet.

... and it's fixed. Turns out `uh.show()`, the function which wrote to the RGB LED matrix, was being called way to much and eating everything. I might even be able to switch back to the pi-3 fine. I then brought back in the other display and rounded things off by adding code that effectively acts as as PI, but the results are good- in this image the target is 300 BPM

```
That is 268.54308806969834 BPM
bpm to slow, correcting by 0.0
Pattern Incremented
Last loop took 3.3718583583831787 seconds
That is 284.70946818190913 BPM
bpm to slow, correcting by 0.003145691193030166
Pattern Incremented
Last loop took 3.2737534046173096 seconds
That is 293.2413903399119 BPM
bpm to slow, correcting by 0.004674744374839253
Pattern Incremented
Last loop took 3.2325990200042725 seconds
That is 296.9746615832146 BPM
bpm to slow, correcting by 0.005350605340848063
Pattern Incremented
Last loop took 3.2128303050994873 seconds
That is 298.8019623931781 BPM
bpm to slow, correcting by 0.005653139182526605
Pattern Incremented
Last loop took 3.2022712230682373 seconds
That is 299.7872238567543 BPM
bpm to slow, correcting by 0.005772942943208796
Pattern Incremented
Last loop took 3.202721118927002 seconds
That is 299.745111844651 BPM
bpm to slow, correcting by 0.0057942205575333685
Pattern Incremented
Last loop took 3.200024366378784 seconds
That is 299.997715669383 BPM
```

Switching back to the pi-3, now, since it appears the pi-4 wasn't needed after all, and yep, all good.

Decided to end the night with some code clean up, and then some (off the clock) work to make it easier to navigate and do some show-boating.

Week ending 2/11 = 8.0 hours, **total** = 28.2 hours

Vega Carlson - Group 165 - Spring '22, Week 5

Saturday, Feb 12th -- W0.0 T:28.2

0.0 hr -- No Progress today

Sunday, Feb 13th -- W:0.0 T:28.2

0.0hrs -- No Progress today

Monday, Feb 14th -- W:0.8 T:29.0

0.8hrs -- Worked with Reid in the lab debugging the FM drum and spent a fair amount of time talking with Cole about organizational stuff.

Tuesday, Feb 15th -- W:0.8 T:29.0

0.0hrs -- No Progress today

Wednesday, Feb 16th -- W:1.3 T:29.5

0.5hr - Meeting. See minutes.

Thursday, Feb 17th -- W:4.1 T:32.3

0.8hr - Continued work on the RP2040 controller board

0.8hr - Made Reid some DIY vactrols. Tried to test them, but something is being weird.

1.2hr - Got to the pi for a bit, started with an apt-get update and upgrade (I like to keep things on my network up to date) then started thinking about how to do the per-step values. This is kind of a mess in terms of the object orientation. If I make the steps be the 'parent' of the drums, it implies that each step in the same track can change the drum, which isn't right. If I make it to the track, it's awkward to convey that each step should be able to hold value. I might be able to make the steps children of the drums, which is more logical, but that's still a bit awkward. I think no matter what it's a nasty refactor, so that sounds like the best call, so I've started that. ... okay, some more work in and I've actually decided to remove the track class entirely and just wrap it into the drum. I (think?) I have the refactor done to the point where everything works as before, so now I just need to start bolting the new functionality on that this refactor was done for.

Friday, Feb 18th -- W:8.0 T:36.2

0.8hr - Meeting. See minutes. + some work in the lab with the group

2.4hr - Continued work on the refactor and getting per-step data

~~Updated to python 3.10 on the pi (I wanted to use some new features)~~ This was going to be a pain due to libraries, so while it's installed, I'm not going to use it unless I actually have to.

While I was at it, I went through and did some work to make the code a bit more consistent with python's naming conventions, and it did actually significantly help with readability

Finally finished up the refactor and now each microstep hold data about each drum's parameters. The code is much more sensible now, and I'm quite happy with it.

.7hr - Misc. todos: Getting stuff organized on git, doing some tests on the pi to make sure everything still works as it did before the refactor, worked with purr-data a bit to ensure I can send pan commands to each channel, at least from one of my hardware MIDI controllers

Week ending 2/18 = 8.0 hours, total = 36.2 hours

Vega Carlson - Group 165 - Spring '22, Week 6

Saturday, Feb 19th -- W0.0 T:36.2

0.0 hr -- No Progress today

Sunday, Feb 20th -- W:0.0 T:36.2

0.0hrs -- No Progress today

Monday, Feb 21st -- W:1.3 T:37.5

1.3hrs -- Worked with Reid in the lab debugging the FM drum and spent a fair amount of time talking with Cole about organizational stuff.

Tuesday, Feb 22th -- W:4.3 T:40.5

2.0hrs -- Did significant work on the display code, getting the user interface further along

1.0hrs -- Finished up the code for the per-microstep value display on the RGB matrix. I have yet to actually make it possible to change microstep values live or test them being sent to even the digital drums as CC's, but I have no reason to think it won't work. I'm a bit worried about how to cleanly handle limiting values, but I'll figure something out.

Wednesday, Feb 23rd -- W:4.8 T:41.0

0.5hr - Meeting. See minutes.

Thursday, Feb 24th -- W:4.8 T:41.0

0.0hrs -- No Progress today

Friday, Feb 25th -- W:8.0 T:44.2

0.5hr - Meeting. See minutes.

0.7hr - Worked with Kaleb in the lab on the filter

2.0hr - Next thing up on the code is make it so I can set both the base and per-step-offset values from the keyboards and update the display. This is a non-trivial problem and requires python that's a bit outside my comfort zone, at least to write generically, using `getattr` and `setattr`. In this two hours I managed to get the code written that allows for changing the base values, checking all the steps offset values, and for checking an individual step's offset values and that should work regardless of the datatype. The code isn't fully tested, so next time I need to actually propagate the setters and getters down to make it able to be called by the keyboard and make sure it does get reflected in the UI, then I should finally be able to write the playback function that uses the values.

Vega Carlson - Group 165 - Spring '22, Week 7

Saturday, Feb 26th -- W0.0 T:44.2

0.0 hr -- No Progress today

Sunday, Feb 27th -- W:3.1 T:47.3

1.3hr -- Continued working on microstep value UI code. Pressing and holding buttons for a step and microstep can now be used to edit a per-step value, and it is reflected on the LEDs., Made it so that the name of the value being edited is shown on the character display

0.3hr -- Got the display to hold the step display on the value current being edited

1.5hr -- Why is it sometimes a simple line of code can take way longer than it should? I needed to fix the display code so that the steps actually transistion smoothly. I got it, but that took much longer than it should have.

Monday, Feb 28th -- W:3.1 T:47.3

0.0hr -- No Progress today

Tuesday, Mar 1st -- W:4.8 T:49.0

1.7 -- To start, I made it so that track selection is done using the same buttons as the steps. This is because I'm running out of keys, and it will be that way on the final keyboard anyway

Next, there was a bug where the last edited parameter name didn't go away to show the BPM again. I *mostly* fixed it. It's not perfect, but It's good enough and I don't want to put more time into chasing it.

Next, I needed to make it possible to pick which value is actually being changed by the knob. This has been hardcoded until now.

Finally, I added the necessary code the digital drum `play_step()` function to actually test this out. It's *almost* working. Currently changing the value for any microstep in a group of 4 changes them all, but it does work, and I can hear the changes.

Wednesday, Mar 2nd -- W:6.6 T:50.8

.3hr -- Work with Reid in lab on PCB

0.5hr -- Meeting. See minutes.

1.0hr -- Trying to find where the changing a microstep changes all mircosteps in that step bug is... This is proving to be really hard to trace down. I know the problem - I can confirm it by looking at memory addresses - I just can't figure out *why* it's a problem.

Oh. My. God. I figured it out, Python wasn't making copies of the data objects being passed to each microstep, meaning each microstep's data was actually the same memory. one `copy.copy()` later and Startup time is now awful, but it works.

Thursday, Mar 3rd -- W:7.6 T:51.8

0.0hr -- No Progress today

1.0hr-- Made it so that the base values (that micro step offset values act on) can be set as well. These base values are set per pattern though. Also added printing to the screen to indicate that values have been clamped, and improved the logic for the display a bit. I also quickly made it so the high start up time get's covered up by a pretty start up animation that I made back when first getting things tested.

Friday, Mar 4th -- W:8.1 T:52.3

0.5hr - Meeting. See minutes.

Week ending 3/04 = 8.1 hours, total = 52.3

Vega Carlson - Group 165 - Spring '22, Week 8

Saturday, Mar 5th -- W2.1 T:54.4

2.1 hr -- Quickly made a PCB for a clone of the "Green Ringer" effect pedal to act as one of the effects.

Sunday, Mar 6th -- W:4.3 T:56.6

2.2hr -- Watched two videos as outside lectures.

Monday, Mar 7th -- W:4.3 T:56.6

0.0hr -- No Progress today, but was in lab with Reid while he did some keyboard work, some idle conversation about the project

Tuesday, Mar 8th -- W:4.3 T:56.6

0.0hr -- No Progress today

Wednesday, Mar 9th -- W:4.9 T:57.2

0.1hr -- Talked with cole about the PT2399 effect and options for getting it done quickly

0.5hr -- Meeting. See minutes.

Thursday, Mar 10th -- W:6.7 T:59.0

1.8hr - Working in Pure Data to get all the sample drum parameters working. Things are *mostly* working. I need to add the sample UI and then come back and tweak things so that the values make sense.

Friday, Mar 11th -- W:8.3 T:60.6

0.5hr - Meeting. See minutes.

1.1hr - Added in the sample selection UI, as currently the lack of that is making debugging difficult. Currently, now that the selection UI works the Purr Data patch will need some minor updates for them both to agree on values; however, it does work - samples can be changed and locked per step already.

Week ending 3/04 = 8.3 hours, **total** = 60.6

Mar 12-18 → Spring Break

Vega Carlson - Group 165 - Spring '22, Week 10

Saturday, Mar 19th -- W0.0 T:60.6

0.0hr -- No Progress today

Sunday, Mar 20th -- W:0.0 T:60.6

0.0hr -- No Progress today

Monday, Mar 21st -- W:0.2 T:60.8

0.2hr -- Briefly worked in the lab with Reid on the keyboard

Tuesday, Mar 22nd -- W:0.2 T:60.8

0.0hr -- No Progress today

Wednesday, Mar 23rd -- W:1.3 T:61.9

0.5hr -- Meeting. See minutes.

0.6hr -- Tested the LED matrix on the keyboard. It works!

Thursday, Mar 24th -- W:2.6 T:63.2

0.6hr - Starting by checking the sequencer & Sample drums for bugs and TODOs

- Lost sample drums once as PD crashed. Not sure what to do about this or how to reproduce
- ☐ Should be able to select step for changing parameter without flipping its state
- ☒ ~~Reverb seems to be on even when at 0~~ Reverb parameter is inverted. 127 is fully wet.
- ☒ Note changing is backwards. It's actually backwards in PD, so it needs flipped in the python as it's easier. (I really struggled to get it right in PD)
- ☒ Need to be able to see what parameter is selected without changing value
- ☒ Panning currently doesn't pan, but does go mute at 127 (probably a PD issue)
- ☒ Delay time shouldn't default to 0
- ☒ BDN9 needs to be remapped so that the shift functions on the knobs work
- ☒ Never actually programmed in the ability to change the pattern sequence (!!!)
 - ☒ Should probably add pattern copy-paste if it's easy enough, as building each pattern from scratch is horribly tedious. This might be a very large task though, and obviously time is running thin.
 - I may reuse some of the step sequence entry keys for setting the pattern sequence, and might make it so that the pattern sequence is selected as a sort of 11th track. It's a bit of a hack-y solution, but it should be relatively easy to implement.
- ☒ Song save/restore was made near the beginning, but I suspect other changes have since broken it, and I never made the UI for it. It may just be that we limit it to one song, though I would still like to have a save button combo and for it to reload this state on initialization.
 - I'm not sure, but I think song restore might actually still be happening, just from a really old version?
 - The functionality to change songs is still there, but exposing a UI for it would take time.

0.7hr - Starting with the BDN9 code. Most of the issue was accidentally having something as `x(); if(cond){y();}` when I needed `if(cond){y();}else{x();}`. An easy fix in logic, but quite annoying. Then fixed these in the python code to respect the mappings, fixed the delay time defaulting to 0 and delay mix defaulting to fully wet, flipped the note (pitch) and reverb value mappings to make them logical, and made the selected parameter visible on screen for a time after the parameter select knob has been turned.

Friday, Mar 25th -- W:8.0 T:68.6

0.5hr - Meeting. See minutes.

0.5hr - Helped Kaleb get KiCad (and some other things) working well on his laptop, as some things were horribly slow.

2.0hr - Starting by fixing the panning parameter of the digital drums in Purr Data. I then tried to make it so step parameters could be changed without actually flipping the step state. This is not going to easily fit into this code as it's steep, and realistically the current setup is fine as is, albeit slightly annoying as it means per-step values should really be entered when the step is first toggled on. Next got both manual pattern changing and pattern sequence step entry working.

0.5hr - Got save & load functionality working again, and added the ability to change songs.

0.5hr - Added pattern copy and paste, as the lack of it is very tedious. It also shows some nice error messages on the screen if you try to copy steps between incompatible drum types.

0.8hr - Found a newly introduced bug (from a fix yesterday) where changing the BPM gets covered up by the target parameter. I had to resort to an ugly fix for this, but it works. Also realized I had the bpm knob backwards, so got that too.

Also found a bug in my copy paste code- it wasn't making a deep copy. So, any pattern that is copied is by reference... meaning a change in a pasted pattern affects the original copy. Fixed with `copy.deepcopy()`. Frankly, I think the unintended functionality was actually pretty neat though, so I went ahead and made it a feature on a different key press.

0.6hr - Working on the error handling code to show if the keyboards aren't found. Also started doing a bit of research into what python libraries to use if we can't get the RP2040 board to show up as a MIDI device. Last time I did serial in python it was a major headache. It looks like `pyserial` is still the go-to, and I vaguely remember using `ports = glob.glob('/dev/tty[A-Za-z]*')` from [This Stackoverflow](#) answer before. I'd really rather not do it this way, but if necessary I think it can work. I also went looking for good MIDI libraries supporting the 2040, and all of them appear to be for CircuitPython this is the only one I can find that's in C: <https://github.com/hathach/tinyusb>.

Week ending 3/25 = 8.0 hours, **total** = 68.6

Vega Carlson - Group 165 - Spring '22, Week 11

Saturday, Mar 26th -- W:0.0 T:68.6

0.0hr -- No Progress today

Sunday, Mar 27th -- W:0.0 T:68.6

0.0hr -- No Progress today

Monday, Mar 28th -- W:3.0 T:71.6

!3.0hr -- Worked on getting Linux reinstalled and setup on my laptop so that we can more easily run the sequencer, flash the keyboards, etc. while on campus. (actually about 5 hours, but I don't think it's fair to count all of this) It's still not done, but it's most of the way there.

Tuesday, Mar 29th -- W:3.0 T:71.6

0.0hr -- No Progress today

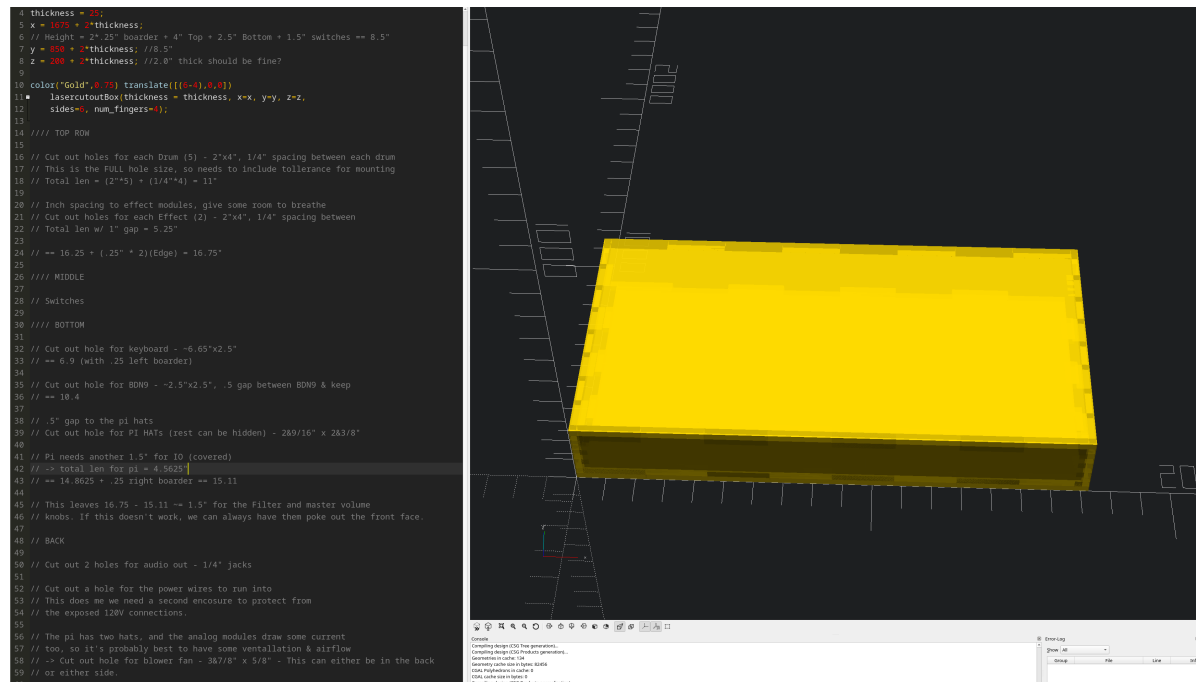
Wednesday, Mar 30th -- W:4.0 T:72.6

0.5hr -- Meeting. See minutes.

0.5hr -- Trying to get the rp2040 board working. Still no dice.

Thursday, Mar 31st -- W:5.5 T:63.2

1.5hr - Starting up OpenSCAD and to try get to get [this Lasercut library](#) working, just to get the very star of the enclosure. Sort of just putting comments in and thinking about the box's layout right now. Sizes as listed are wrong, but should be close. I might switch to cm before moving on to much.



Friday, April 1st -- W:8.0 T:74.1

1.0hr - Meeting. See minutes.

Week ending 3/25 = 6.5 hours, total = 74.1

Vega Carlson - Group 165 - Spring '22, Week 12

Saturday, Mar 26th -- W:0.0 T:74.1

0.0hr -- No Progress today

Sunday, Mar 27th -- W:0.0 T:74.1

0.0hr -- No Progress today

Monday, Mar 28th -- W:0.0 T:74.1

0.0hr -- No Progress today

Tuesday, Mar 29th -- W:0.0 T:74.1

0.0hr -- No Progress today

Wednesday, Mar 30th -- W:2.7 T:76.8

1.7hr -- Got the controller board flashed with Circuit Python & got MIDI input and LED output working

0.5hr -- Meeting. See minutes.

0.5hr -- Registering for the showcase

Thursday, Mar 31st -- W:4.2 T:78.3

1.5hr -- Started out by looking over Reid's OpenSCad work. I see a few things that need changed - namely the master volume & filter pot holes - but otherwise it looks good.

Next up, I needed to look into how to get the LED matrix (using the IS31FL3731) in the keyboard working with QMK. The documentation for it is rather poor, so to start with I looked for a keyboard using the driver and with a similar layout. Strangely, many appear to be using the driver with RGB leds and I'm not even sure how that would be possible.

It appears the "Infinity 60%" is relatively close; however, it has way more code to make it work than I'd expect. Otherwise, [opendeck](#) also looks light it might have good code to yoink'. Unfortunately, I am still going to need to make it so that I can controls the LEDs over serial, which will require a fair amount of work as that's a weird setup for QMK. Reading the docs, I think [Raw HID](#) is the best call and there is a python library for using it, but it's still going to be tricky. As I don't have the handwired keyboard on hand to test, I'm going to start by trying to get rawhid working with the BDN9 ... again, as I tried this last semester and failed.

I am able to at least use the python HID library to get the device name back out, ~~but I'm having a hard time writing data, and there's basically no documentation here~~ Thankfully, my past python experiance came in handy,

```
with hid.Device(0xcb10,0x2133) as h: h.write((256).to_bytes(2,byteorder='big'))
```

seems to work, but I can't read data back out, even though it should be echoing what I write to it. It just sits and waits. As I know I can't get LED control from the many hours I already tried that, I'll see if I can get it to send a key on data reception instead.... nope.

Things are starting to make more sense from reading https://github.com/Drugantibus/qmk-hid-rgb/blob/master/hid_rgb.py... though only a little. The associated keymap, https://github.com/Drugantibus/qmk_firmware/blob/master/keyboards/massdrop/alt/keymaps/drugo/keymap.c, makes it make a bit more sense, but this is going to take some more digesting. I'm going to wait until I have the other keyboard to go much

further, as I really need the simpler LED matrix for debugging.

note, on this board RAW_EPSIZE is 32

Friday, April 1st -- W:8.0 T:82.1

1.5hr - See Minutes + Work on case with Reid, setting up testing environment for the control board with Caleb, discussion about how the entire project will fit together.

2.3 - Started by doing an apt-get update & upgrade on pi as it's been a minute. Then got to work on the 3x10 keyboard. After a lot of banging my head into a wall, I at least have the LED matrix able to be turned on and off via a keypress, though still not through sending messages as I need. ~~I think the host side isn't even sending the messages at all so rather than debug that I'm going to see if I can find an alternative~~

I managed to get it to respond to RAWHID events, so I can now turn the leds on/off with a python command. Beyond needing to actually make it so I can turn on/off LEDs individually, I also need to figure out why the board keeps losing connection. It might just be a bad USB cable, though I also suspect there's a few point in the rats nest of wires that are trying to touch and short.

Week ending 3/25 = 8.0 hours, total = 82.1