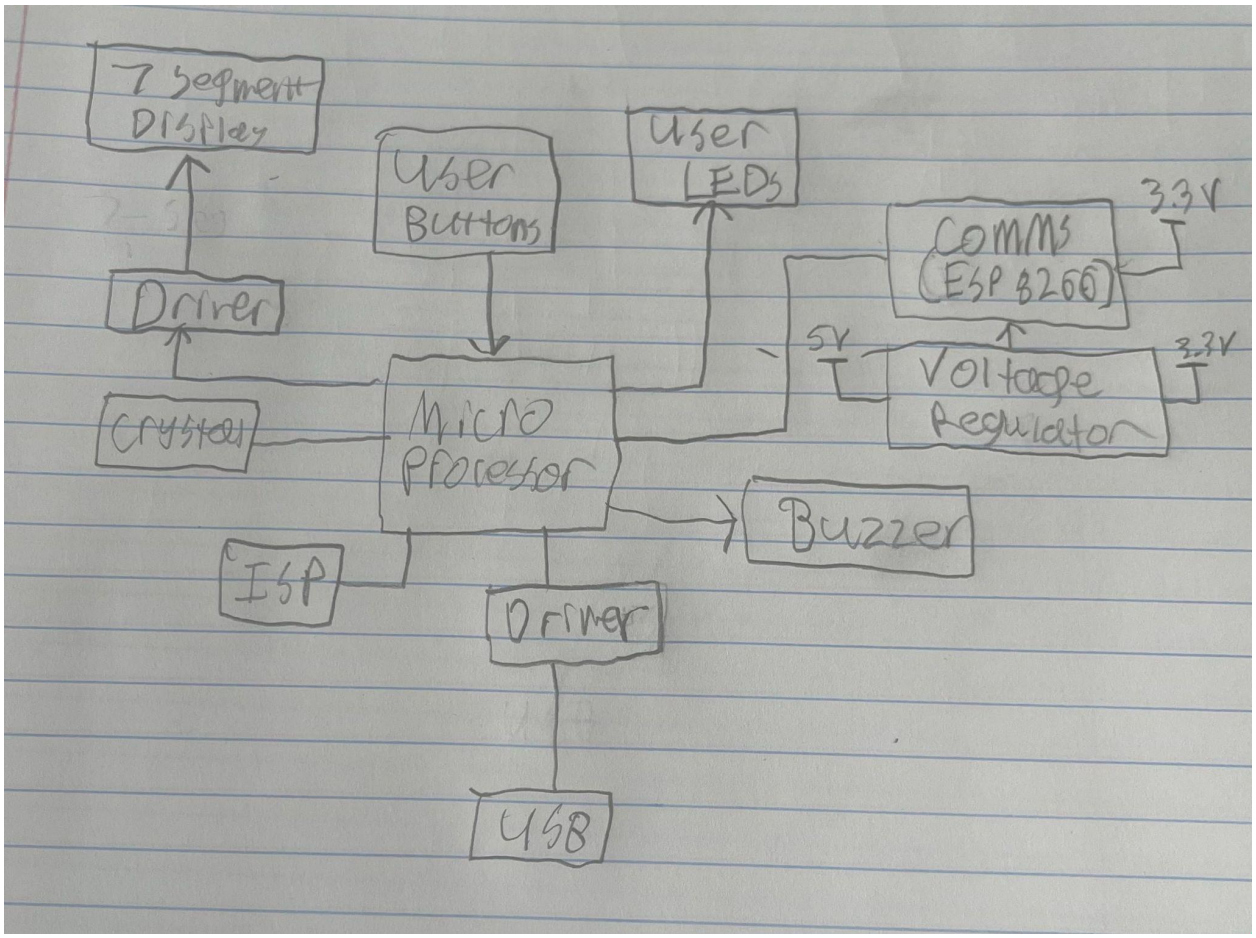
Chris Nodine
ENCE 4012
05/01/2023

Beta Release Report

Project Requirements:

-A kitchen timer that can be incremented, stopped, and started
-A buzzer to sound when the timer is complete
-Can display minutes and seconds(up to one hour)
-A wifi module and web app to control timer
-Should be small
-Cheap to produce

System Design:



Components Selection:

One thing to note is that the WiFi module that we're using, the ESP 8266, requires a 3.3V power supply as opposed to the 5V necessary to power all of the other components. In order to fix this, we need a voltage regulator that inputs 5V and outputs 3.3V, as pictured above.

BOM Table:

	References	Value	Footprint	Quantity
1	C1, C4, C5, C6, C7, C10, C11, C12	0.1uF	C_0603_1608Metric	8
2	C2, C3	22pF	C_0603_1608Metric	2
3	C9, C13	1uF	C_0603_1608Metric	2
4	C8	10uF	C_0805_2012Metric	1
5	C14	2.2uF	C_0603_1608Metric	1
6	C15	10nF	C_0603_1608Metric	1
7	R5, R6, R7, R8, R9, R10, R11, R12	100 ohms	R_0805_2012Metric	8
8	R1, R2	10k ohms	R_0805_2012Metric	2
9	R3, R4	330 ohms	R_0805_2012Metric	2
10	R13, R17	10k ohms	R_0603_1608Metric	2
11	R14, R15	22 ohms	R_0603_1608Metric	2
12	R16	1k ohms	R_0603_1608Metric	1
13	D1, D2, D3	LED	LED_0805_2012Metric	3
14	U1	CA56-12EWA	CA56-12EWA	1
15	U2	74HC595	TSSOP-16_4.4x5mm_P0.65mm	1
16	U3	ATmega32U4-A	TQFP-44_10x10mm_P0.8mm	1

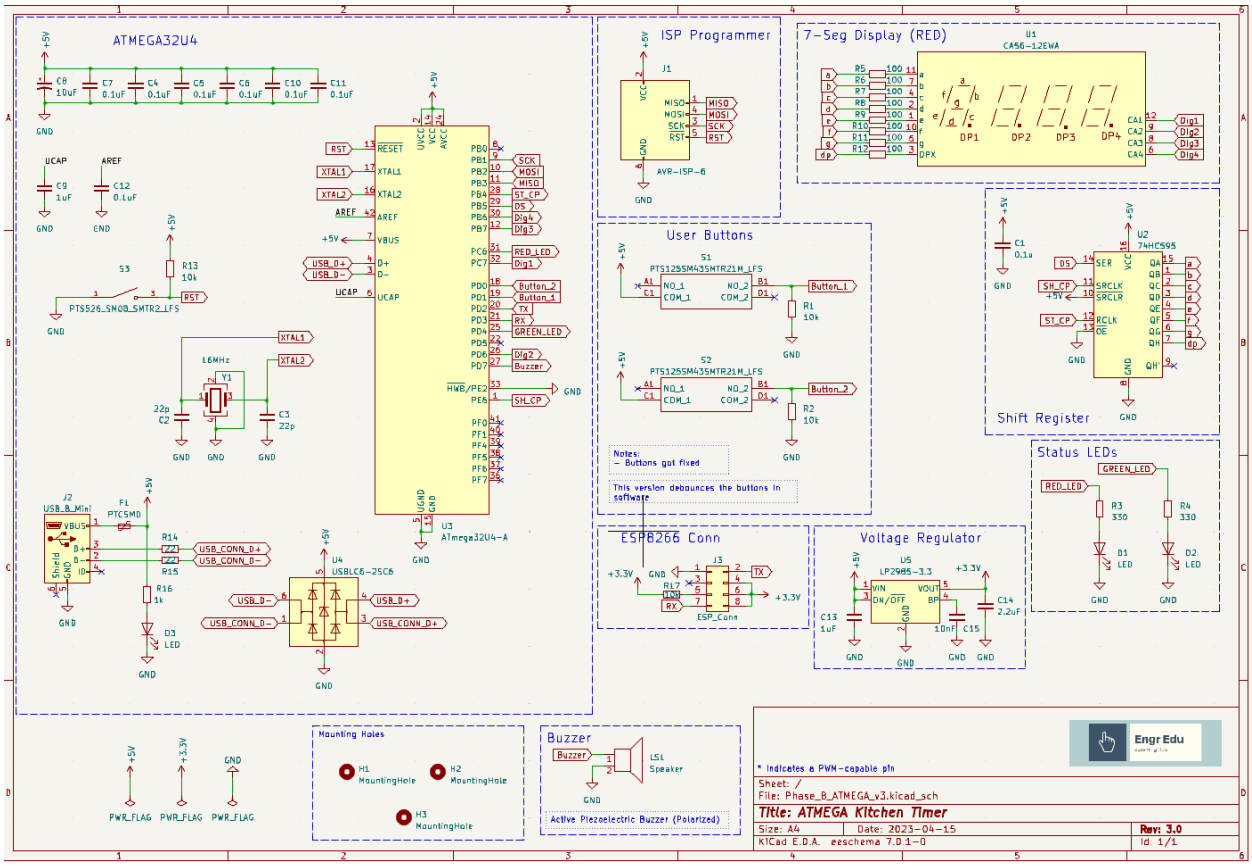
17	U4	USBLC6-2SC6	SOT-23-6	1
18	U5	LP2985-3.3	SOT-23-5	1
19	Y1	16MHz	Crystal_SMD_Abracon_ABM8G-4Pin_3.2x2.5mm	1
20	F1	PTCSMD	Fuse_1812_4532Metric	1
21	H1, H2, H3	MountingHole	MountingHole_2.2mm_M2	3
22	S1, S2	PTS125SM43SMTR21M_LFS	PTS125_SMD_Button	2
23	LS1	Speaker	Buzzer_12x9.5RM7.6	1
24	S3	PTS526_SM08_SMTR2_LFS	PTS526_SMD_Button	1
25	J1	AVR-ISP-6	PinSocket_2x03_P2.54mm_Vertical	1
26	J2	USB_B_Mini	USB_Mini-B_Lumberg_2486_01_Horizontal	1
27	J3	ESP_Conn	PinSocket_2x04_P2.54mm_Vertical	1

Build Prototype:

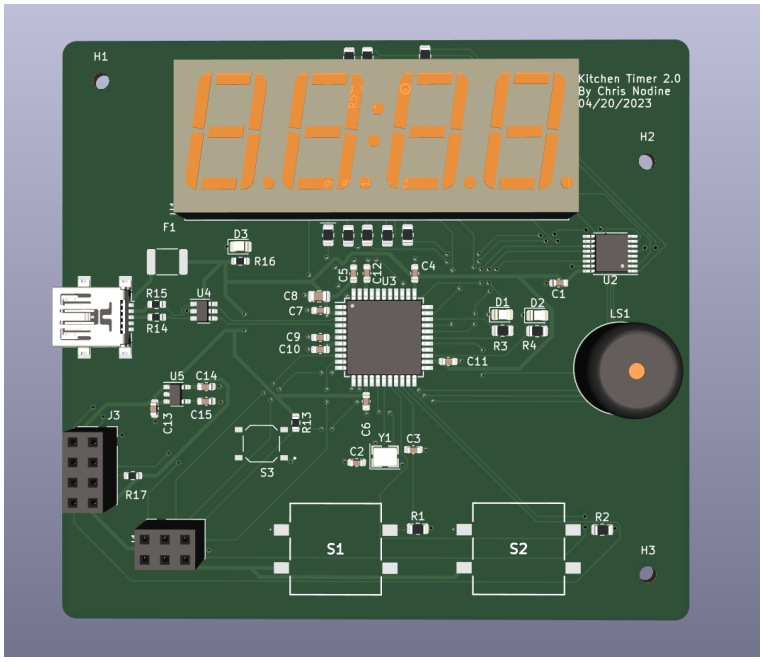
Since we want the footprint of the timer to be quite small, we decided to build the clock prototype board on top of the arduino. Meaning we designed sockets on the bottom of the kitchen timer PCB that directly connect to the pinouts on top of the Arduino. We did this to also make sure we had code that could utilize timers to display numbers on the display and to update every second. Once we got all of our components and code running properly on the Arduino, we moved on to use a different microprocessor to further shrink the footprint of our timer.

PCB Design:

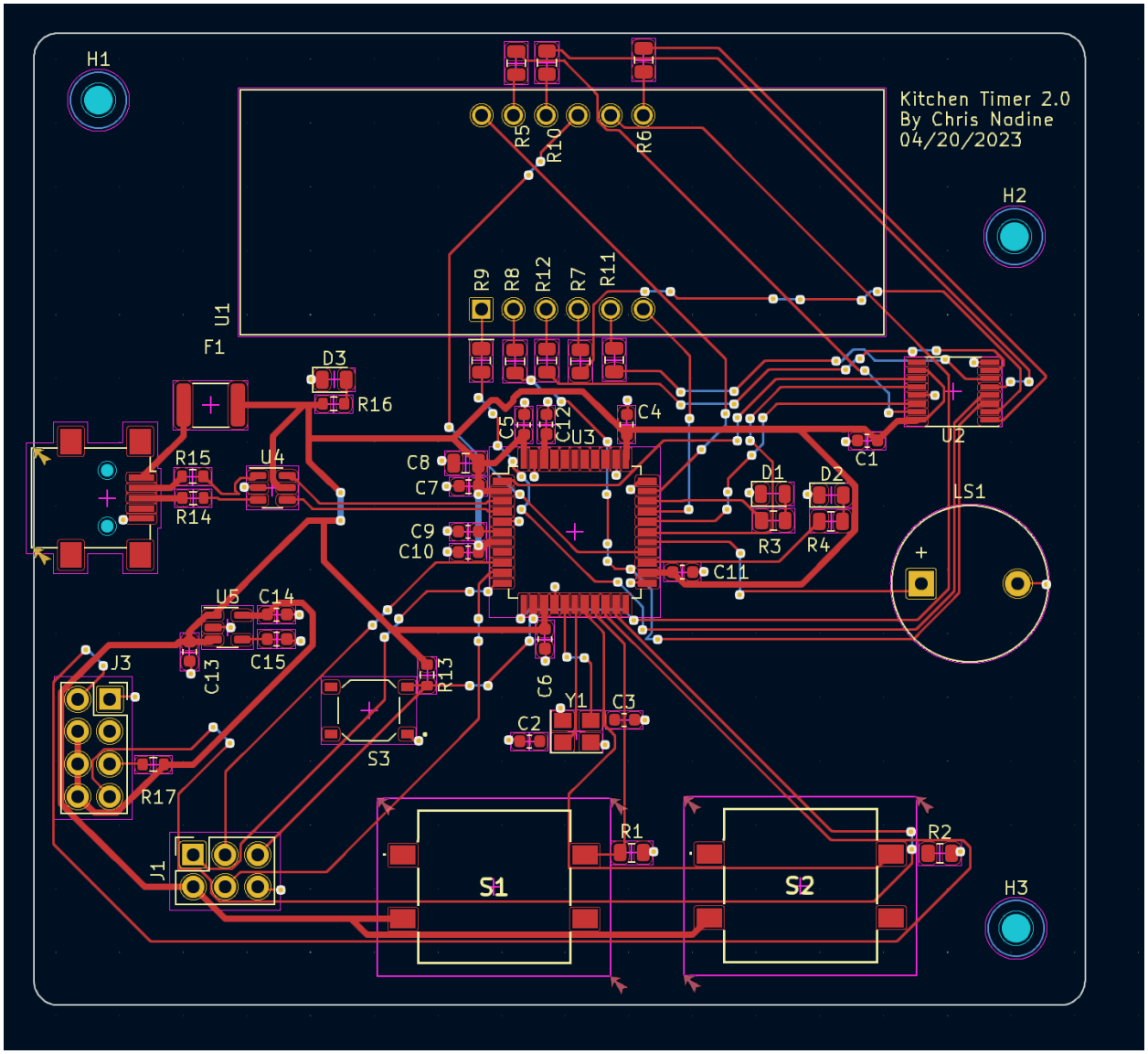
Schematic:



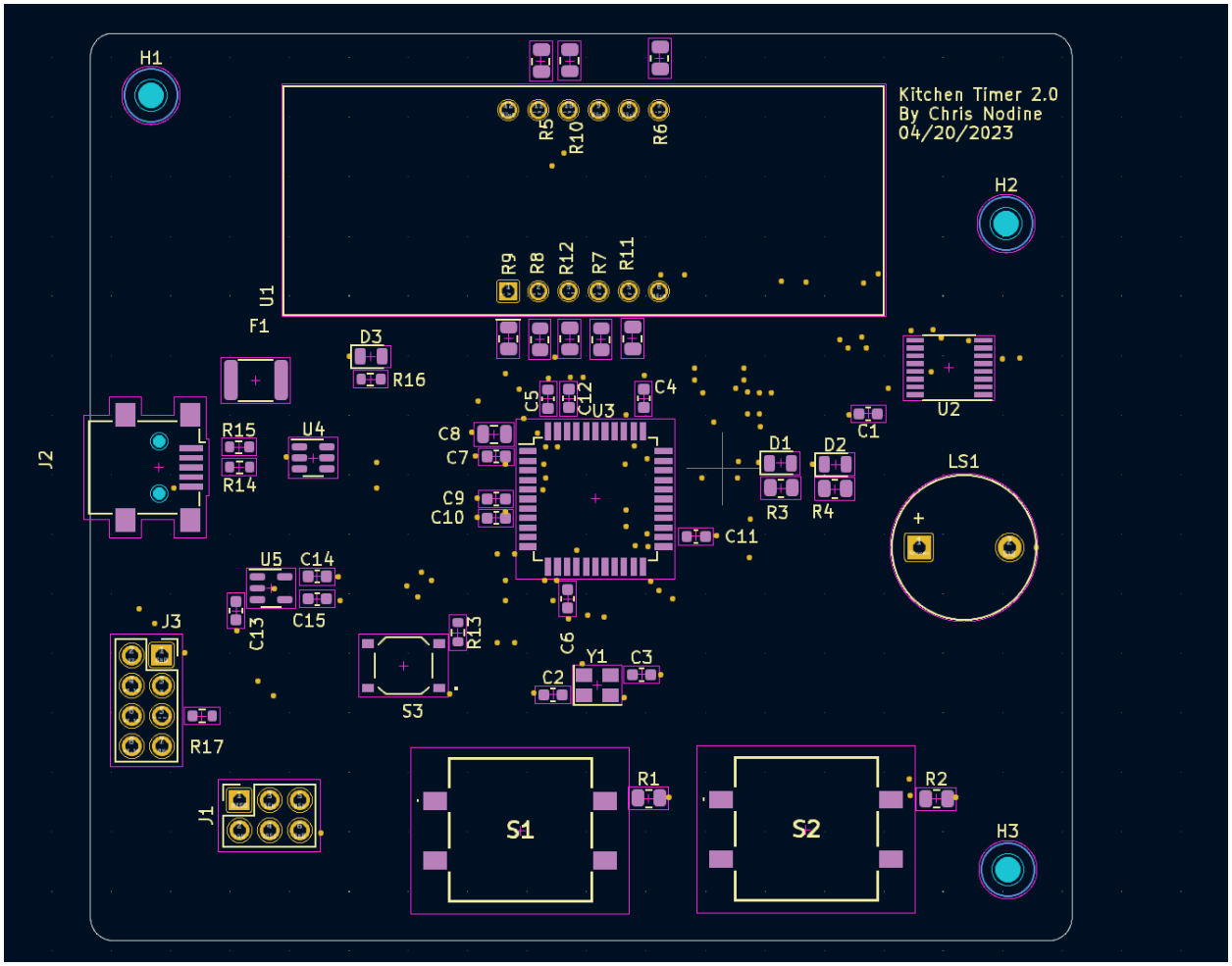
3D Model:



PCB Design (With Routes):



PCB Design (Without Routes):



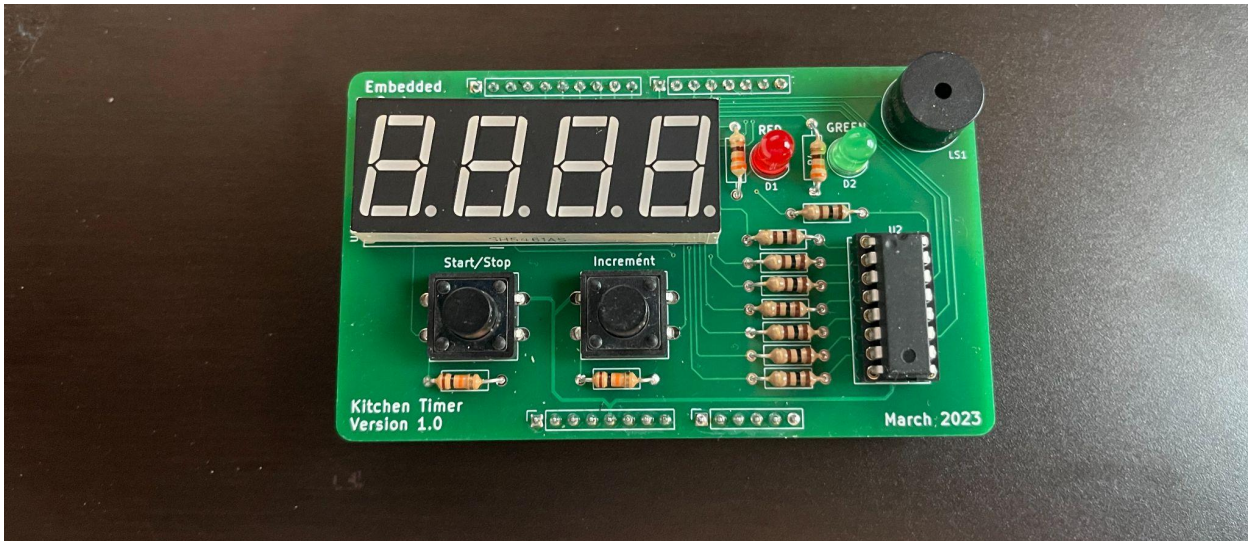
Description:

We put the USB port on the left side of the PCB for easy accessibility. Because of this, we had to put the Fuse and a USB driver nearby. We then placed the microprocessor and all of its decoupling capacitors near the middle of the board in hopes of making routing easy. This requires the crystal nearby. I then placed the display at the top of the PCB and the two user buttons on the bottom. Then, I placed the buzzer below and to the right of the display. Lastly, I placed the two connectors (J1, and J3) at the bottom right of the board. I had to be extremely careful in choosing J3's position and orientation so that the WiFi Module doesn't hang off of the board. Also, I placed the J1 connector on the back side of the PCB to clear up some clutter. Then, I made the routes, generated the gerber files, and sent it off to the Fab-house.

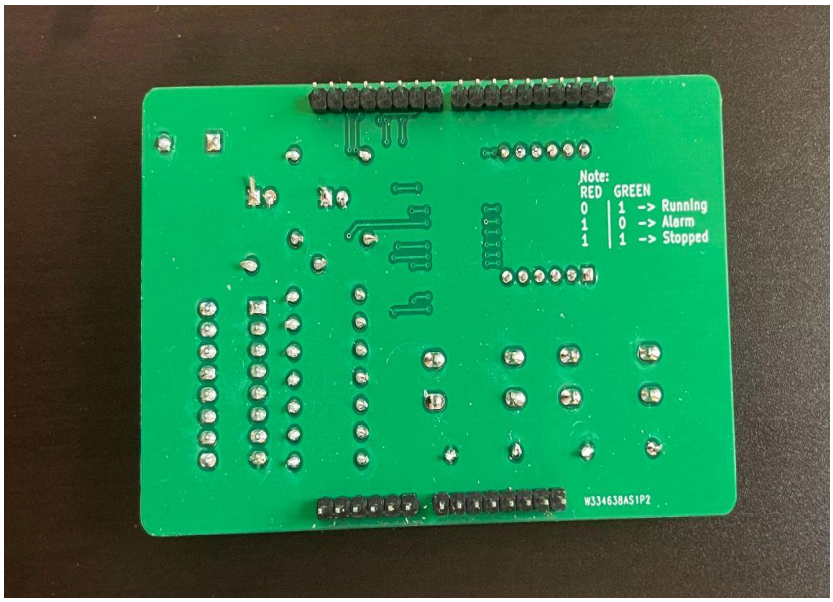
Assembly Stage:

We didn't get the PCBs in time for this report so I've provided some pictures of the assembly of the arduino prototype to get a rough Idea of what the timer will look like.

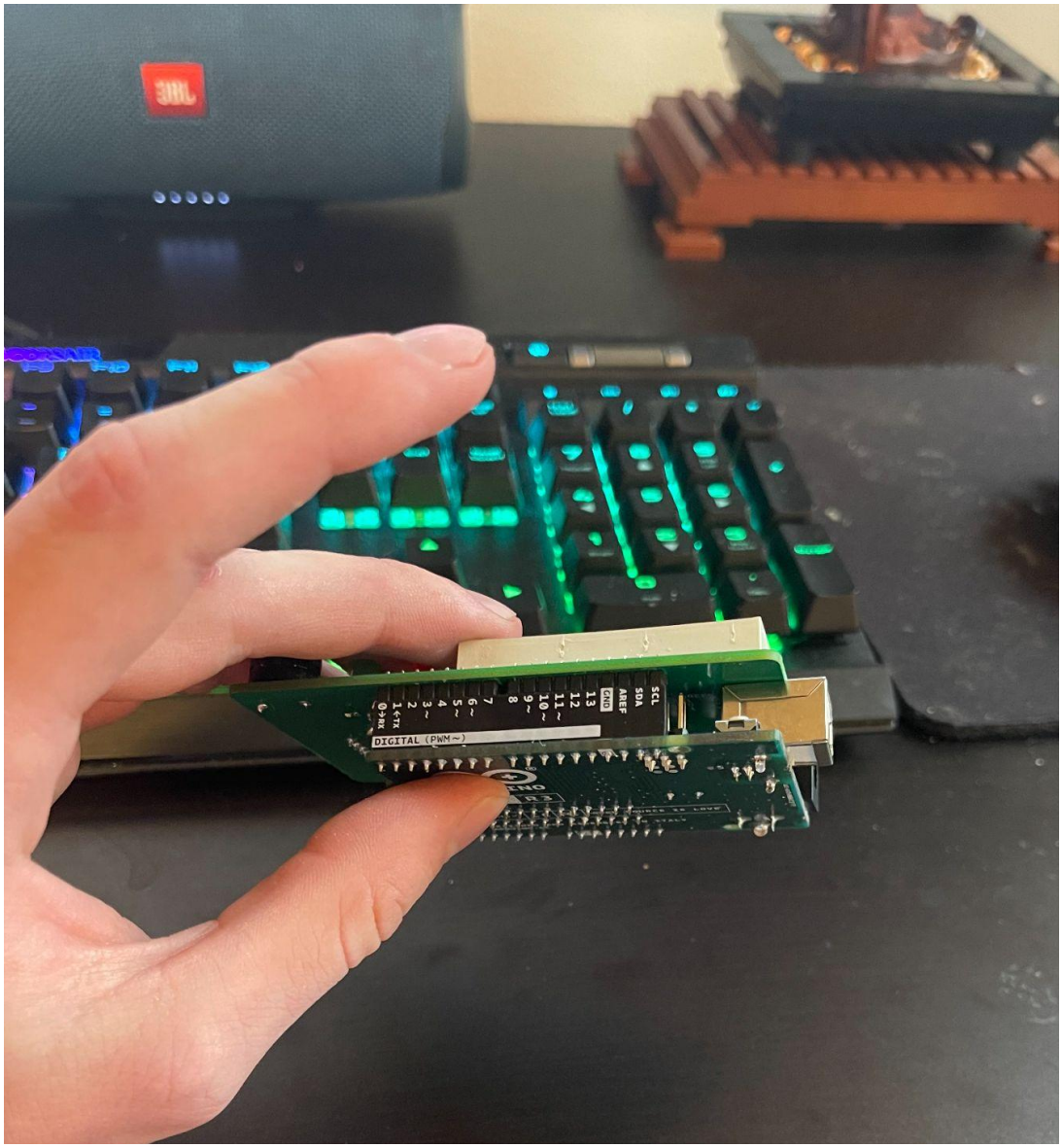
Front of Timer PCB



Back of Timer PCB(Arduino Shield)



Arduino and Timer PCB Connected

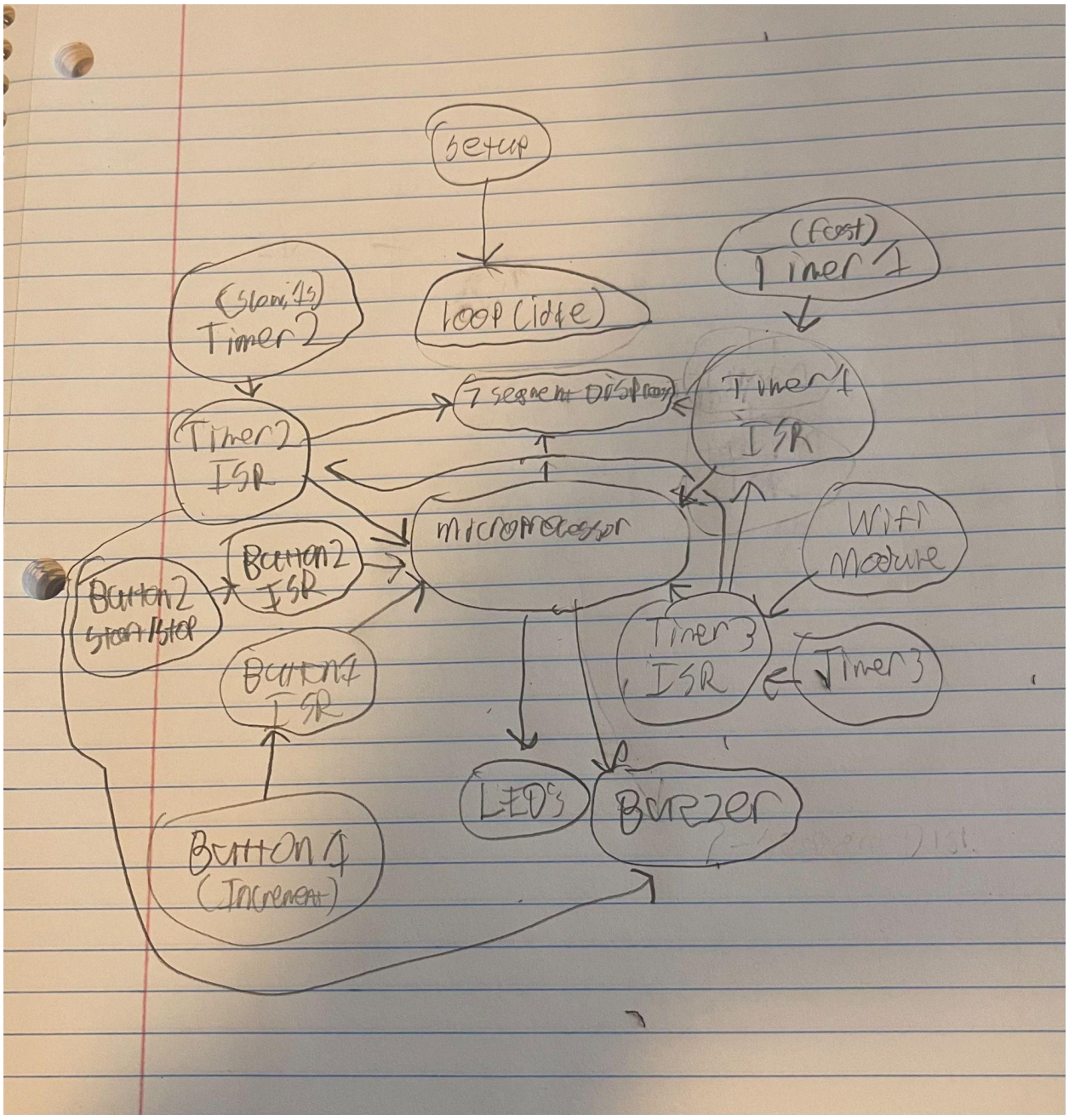


Software Development:
GitHub link here

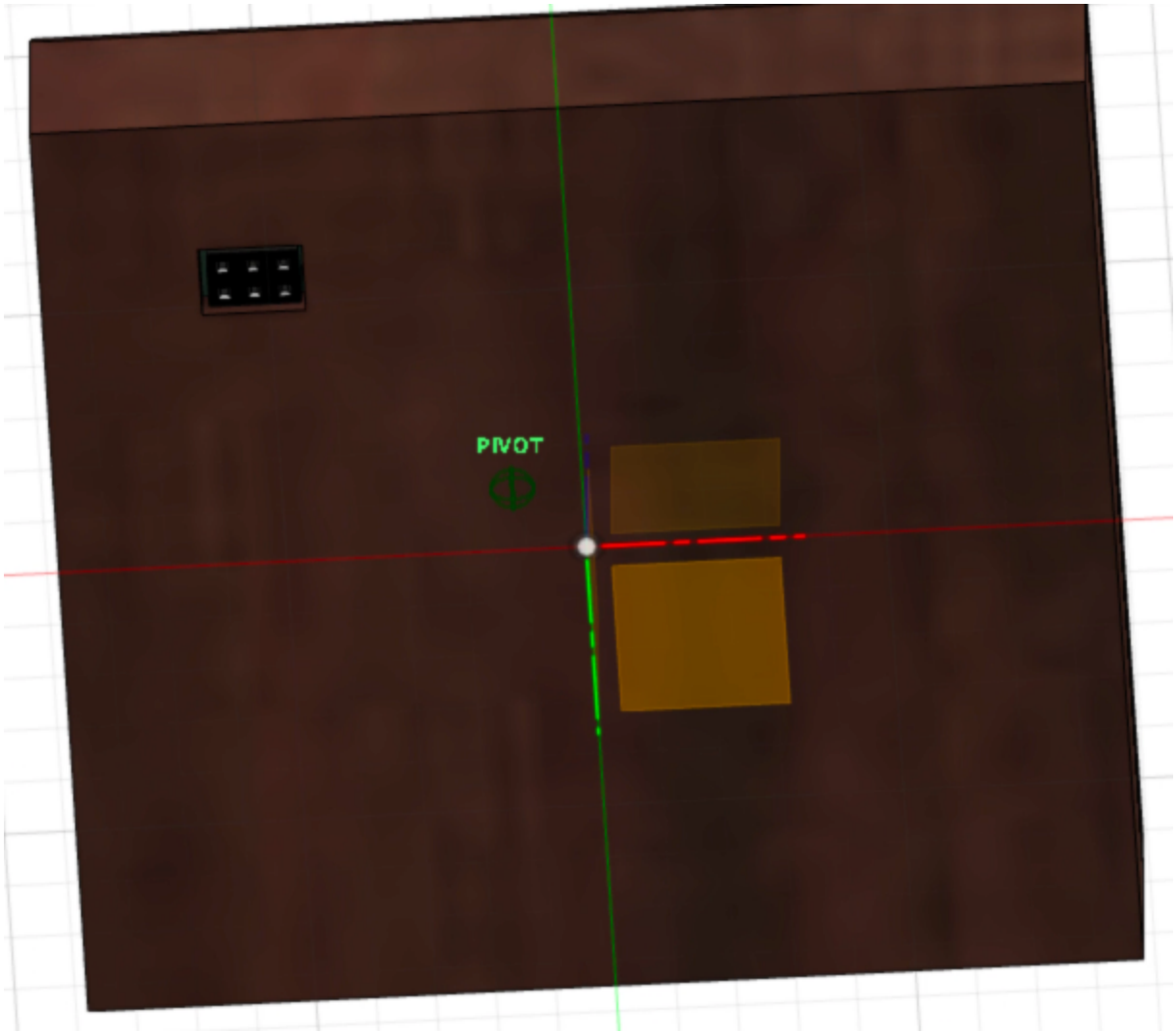
First, I'll discuss the process of making the timer work, then I'll discuss the code with the wifi module. The timer works by utilizing 2 timers and their corresponding interrupts. One of the timers is running extremely quickly(1024 prescaler) and the other is running at one second. We use the first timer's interrupt to quickly update the display by updating, then turning off each digit. Then the second timer interrupt is used to decrement the timer's value every second when the timer is running. Then, to make the Wifi module work, we had to add a third timer into the mix that is set to 4ms which is approximately half the time it takes the wifi module to send a byte to the microprocessor. This timer is used to check if there was a command sent from the Wifi module. If there is, it adds it to a buffer. Then, when the end command character is sent from the

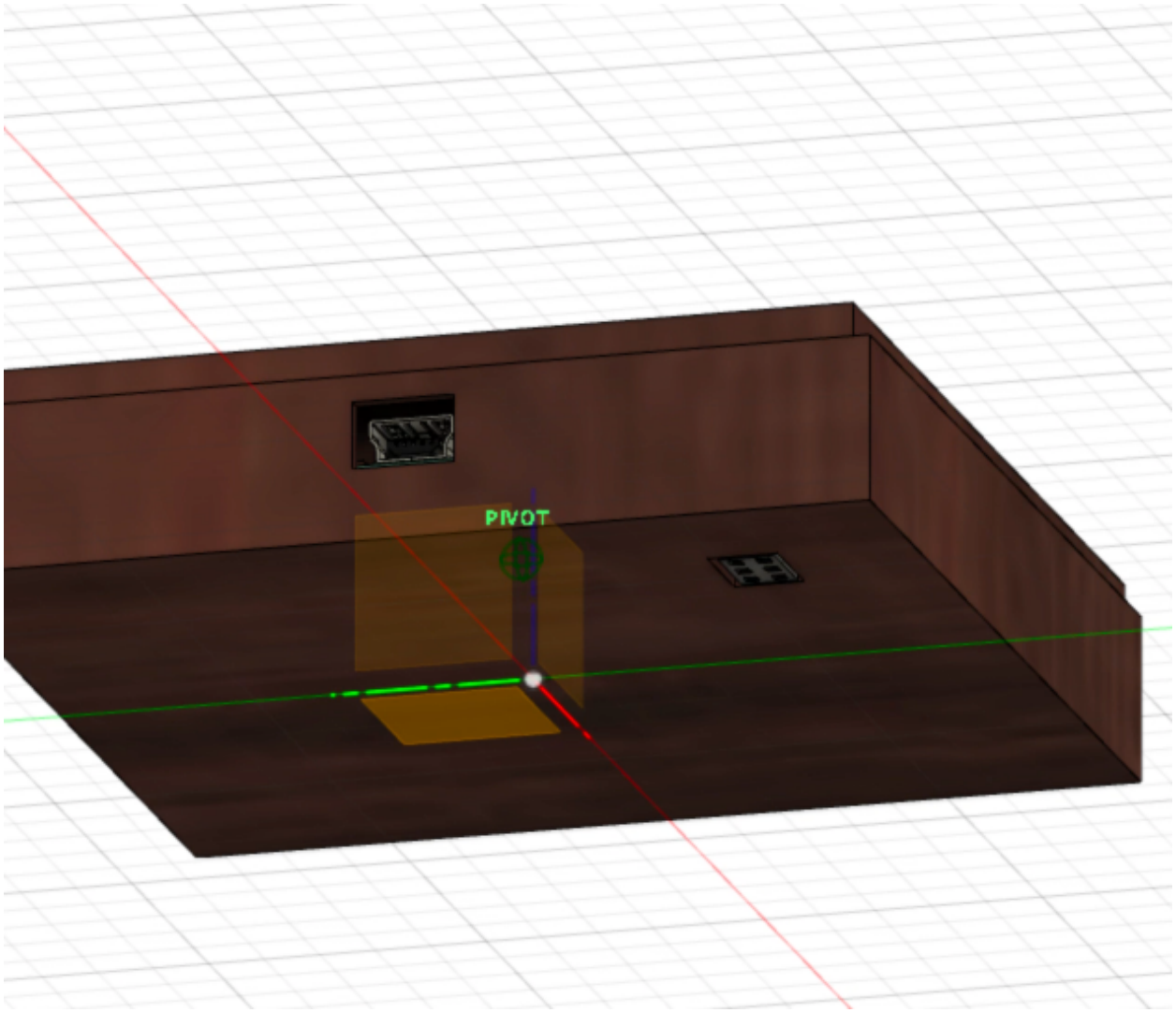
wifi module, the third timer interrupt throws a flag that executes the command sent from the wifi module.

Block Diagram:



Enclosure Design:


PIVOT


PIVOT