Phase B – Kitchen Timer (Beta Release)

Evan Hill
ENCE 4231
Embedded Systems Programming
Dr. Goncalo Martins

Project Requirements

When developing ideas for the kitchen timer, examples were sought out on the internet to find possible features to be included in the design. Examples of these were the capability to be stood up or mounted, be solar-powered, using dials or buttons to control the timer, and being able to control the volume. For the kitchen timer, several requirements and constraints were decided on that needed to be met with the design. Starting off, the total size of the PCB and enclosure must be within 10 cm. by 10 cm. It would run solely on an external power source through a USB cable. Two user buttons would be implemented to make full use of the product. The kitchen timer will use a 4 digit, 7 segment display to display the timer count down. The kitchen timer would be capable of increasing the time before the count down, starting, pausing, and resetting the time. After the kitchen timer reaches a time count of 0, it will then buzz to alert the user. As for the enclosure, the size requirements remain the same as described. The user should have access to all buttons, the USB port and the display.

System Design

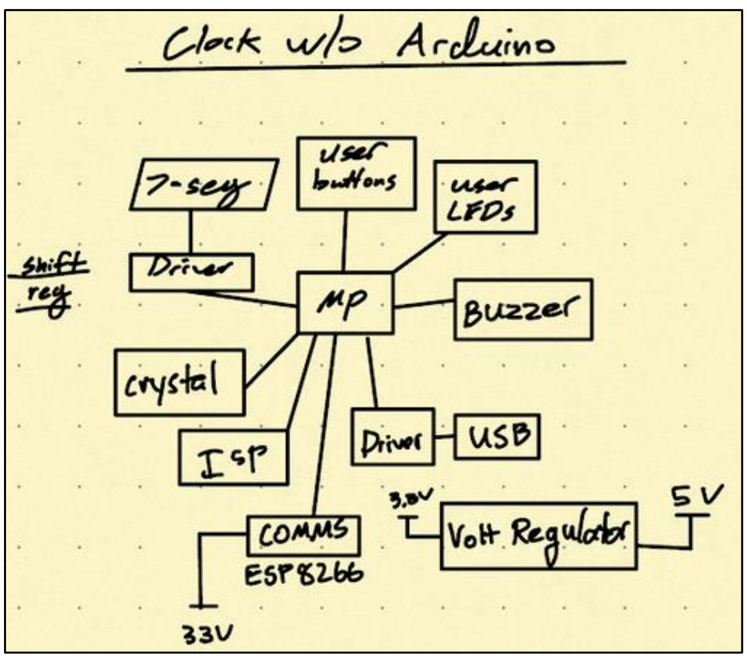

Figure 1 – PCB Design Block Diagram

Above depicts the diagram developed with the ideas for the requirements and constraints for the kitchen timer PCB. At the center of everything is the microprocessor, which handles the computations and controls all the peripherals. There are 3 user buttons to control the functions of the kitchen timer, and LEDs to indicate the status of the kitchen timer. A buzzer is implemented to indicate when the time has reached 0, and an alarm flag is being raised. A USB driver is used to provide power to the kitchen timer, alongside voltage regulator to provide 3.3 V for those peripherals which require 3.3 V rather than 5 V. For COMMS, an ESP8266 was selected to provide the capabilities to interact with external devices through Wi-Fi. The ISP provides a port to program the kitchen timer. The crystal provides a clock pulse at 16 MHz for the microprocessor. And finally, the 4 digits, 7 segment display is used to provide a visual indication of the current time.

Components Selection

Item	Qty	Reference(s)	Value	LibPart	Footprint
1	2	C2, C3	22 pF	Device:C_Small	Capacitor_SMD:C_0603_1608Metric
2	7	C4, C5, C6, C7, C10, C11, C12	0.1 uF	Device:C_Small	Capacitor_SMD:C_0603_1608Metric
3	1	C8	10 uF	Device:C_Polarized_Small_US	Capacitor_SMD:C_0805_2012Metric
4	2	C9, C13	1 uF	Device:C_Small	Capacitor_SMD:C_0603_1608Metric
5	1	C14	2.2 uF	Device:C_Small	Capacitor_SMD:C_0603_1608Metric
6	1	C15	10 nF	Device:C_Small	Capacitor_SMD:C_0603_1608Metric
7	3	D1, D2, D3	LED	Device:LED	LED_SMD:LED_0805_2012Metric
8	1	F1	PTC SMD	Device:Polyfuse_Small	Fuse:Fuse_1812_4532Metric
9	1	J1	AVR-ISP-6	Connector:AVR-ISP-6	Connector_PinSocket_2.54mm:PinSocket_2x03_P2.54mm_Vertical
10	1	J2	USB_B_Mini	Connector:USB_B_Mini	Connector_USB:USB_Mini-B_Lumberg_2486_01_Horizontal
11	1	J3	ESP_Conn	Connector_Generic:Conn_02x04_Odd_Even	Connector_PinSocket_2.54mm:PinSocket_2x04_P2.54mm_Vertical
12	1	LS1	Speaker	Device:Speaker	Buzzer_Beeper:Buzzer_12x9.5RM7.6
13	2	R1, R2	10k	Device:R	Resistor_SMD:R_0805_2012Metric
14	2	R3, R4	330	Device:R	Resistor_SMD:R_0805_2012Metric
15	8	R5, R6, R7, R8, R9, R10, R11, R12	100	Device:R_Small	Resistor_SMD:R_0805_2012Metric
16	1	R13	10k	Device:R	Resistor_SMD:R_0603_1608Metric
17	2	R14, R15	22	Device:R	Resistor_SMD:R_0603_1608Metric
18	1	R16	1k	Device:R	Resistor_SMD:R_0805_2012Metric
19	1	R17	10k	Device:R_Small	Resistor_SMD:R_0603_1608Metric
20	2	S1, S2	PTS125SM43SMTR21M_LFS	PTS125_SMD_Button:PTS125SM43SMTR21M_LFS	PTS125_SMD_Button:PTS125_SMD_Button
21	1	S3	PTS526_SM08_SMTR2_LFS	PTS526_SMD_Button:PTS526_SM08_SMTR2_LFS	PTS526_SMD_Button:PTS526_SMD_Button
22	1	U1	CA56-12EWA	Display_Character:CA56-12EWA	Display_7Segment:CA56-12EWA
23	1	U2	74HC595	74xx:74HC595	Package_SO:TSSOP-16_4.4x5mm_P0.65mm
24	1	U3	ATmega32U4-A	MCU_Microchip_ATmega:ATmega32U4-A	Package_QFP:TQFP-44_10x10mm_P0.8mm
25	1	U4	USBLC6-2SC6	Power_Protection:USBLC6-2SC6	Package_TO_SOT_SMD:SOT-23-6
26	1	U5	LP2985-3.3	Regulator_Linear:LP2985-3.3	Package_TO_SOT_SMD:SOT-23-5
27	1	Y1	16 MHz	Device:Crystal_GND24	Crystal:Crystal_SMD_Abracon_ABM8G-4Pin_3.2x2.5mm

Table 1 – Component Table

Build Prototype

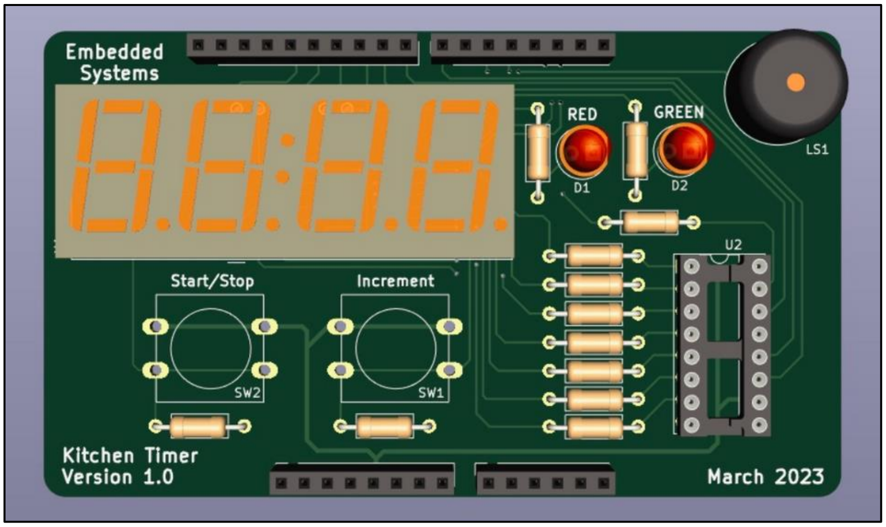

Figure 2 – Kitchen Timer Arduino Shield Prototype 3D Render

A PCB shield Kitchen Timer for the Arduino Uno was employed for the purposes of testing and developing the code for the kitchen timer while the PCB boards were being manufactured offsite. Depicted above is a 3D rendering of the PCB shield used. Compared to the final PCB design, there are some similarities and some differences. Like the final design, a 4 digit 7 segment display is used to display the time, and a buzzer is used to play the alarm signal. Two user pushbuttons are used to control the kitchen timer functionalities, as well as 2 LEDs to indicate the kitchen timer's status. However, what's different is rather than a SMD shift register, a simple 2x8 IC socket is used instead to implement a shift register IC. All the capacitors are missing, as well as the ISP, Crystal, Voltage regulator, and ESP8266, which are either not present, or already included in the Arduino Uno. Pin sockets are provided so that the shield can be mounted directly on top of the Arduino Uno for seamless programming.

PCB Design

For the purposes of designing the PCB, KiCad was used to develop the circuit schematics, PCB Design and PCB layout.

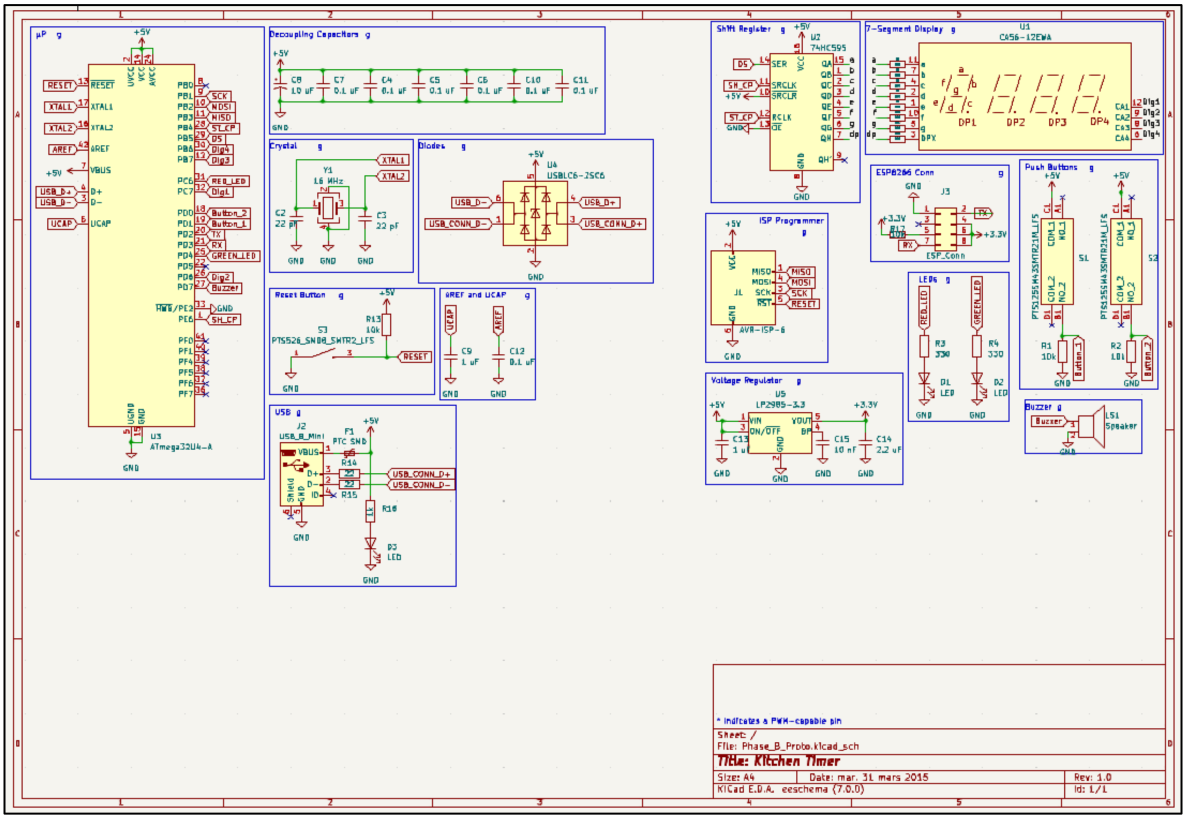

Figure 3 – PCB Circuit Schematic made in KiCad

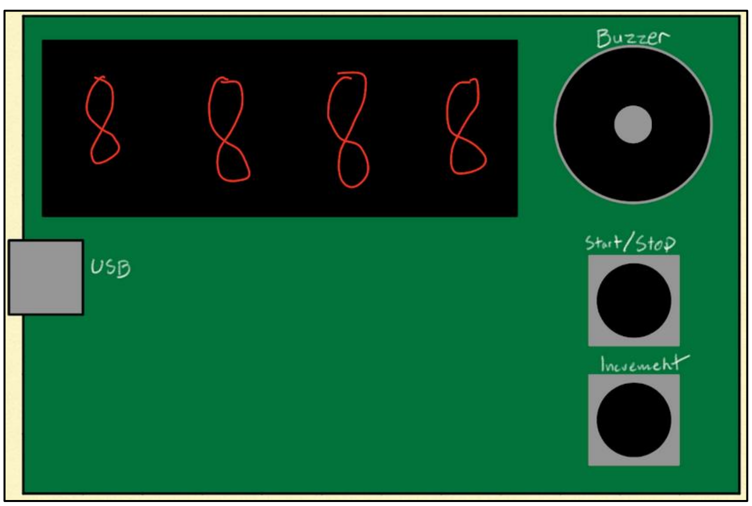

Figure 4 – PCB Layout Mockup

Before beginning designing the layout of the PCB board, I wanted to mockup a simple design for the main peripherals so that I could better direction myself while placing each of the components. The display needs to be in the corner to provide plenty of space for the other components. The USB needs to be on the opposite side from the user buttons so that the USB doesn't interfere with interacting with the user buttons. The user buttons should be close to each other as well as on the edge for ease of use. And the buzzer also occupies a large space, so placing it far away in the corner provides space for other components.

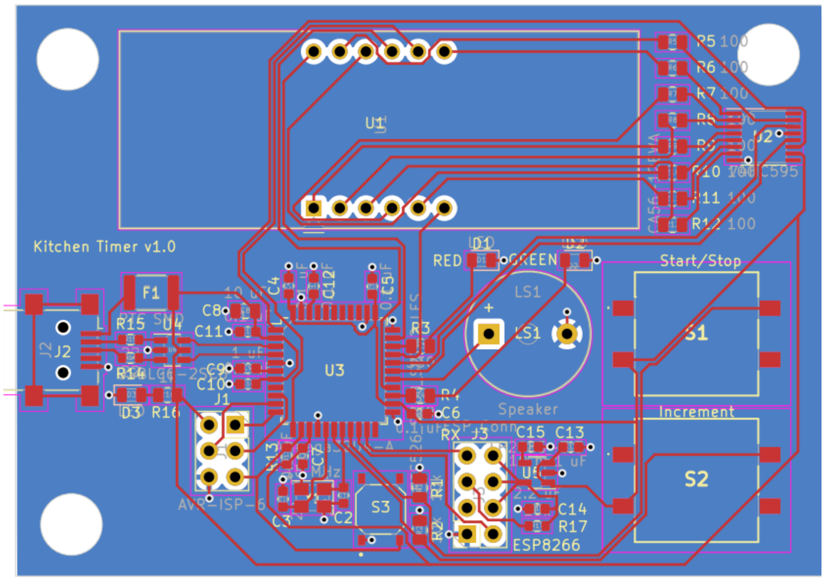

Figure 5 – PCB Layout with routing

Depicted above is the final PCB layout. Compared to the original mockup, the display, user buttons, and USB remain in similar places. However, while routing, the buzzer needed to be relocated so that the shift register, along with all its resistors, could have a close position to the display. Thankfully, there was ample space beside the buttons to be placed alongside the LEDs to centralize all of the indicators. The microprocessor needed to be placed in the most central location possible, since it would have the most routings to it. Beyond the placement of the rest of the peripherals, they were decided based on the space needed, as well as the ease of routing.

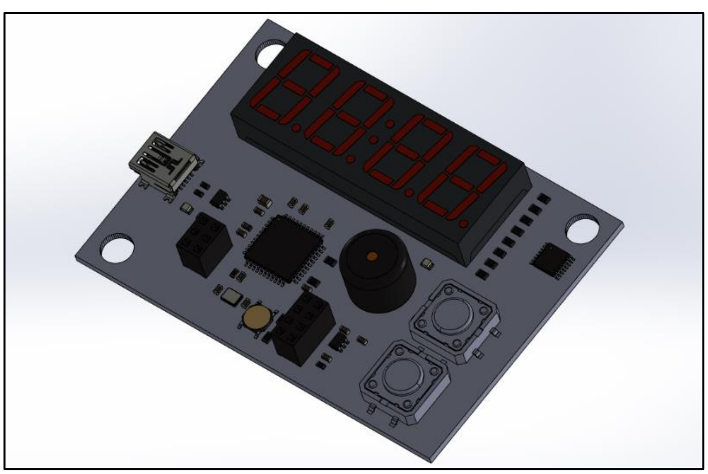

Figure 6 – 3D Render of final PCB Design in SolidWorks

Assembly Stage

Assembly of the PCB boards took place on Tuesday from 4 to 6 pm during weeks 9 and 10. Dr. Goncalo Martins set up the workspaces to solder the PCB boards, including a soldering iron, solar flux kit, soldering wire, tip cleaner, and a tip tinner. Dr. Goncalo Martins also all of the PCB materials required as he guided and aided students through the soldering process, as for many, it was either their first time soldering, or still a relatively new skill. While soldering was difficult at first, soldering so many components allowed me to build confidence and learn methods to solder better. The experience also helped me to practice untensing my shoulders so that my hands could be less shaky. While I couldn't document the process along the way, I am able to provide pictures of the final product. Thankfully, my board had little no faults, and was functioning properly upon being provided with power.

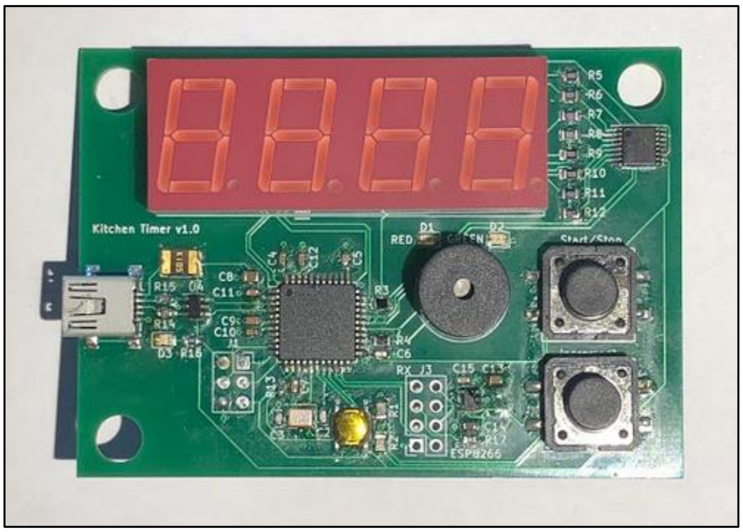

Figure 7 – Fully-Soldering PCB Top-view

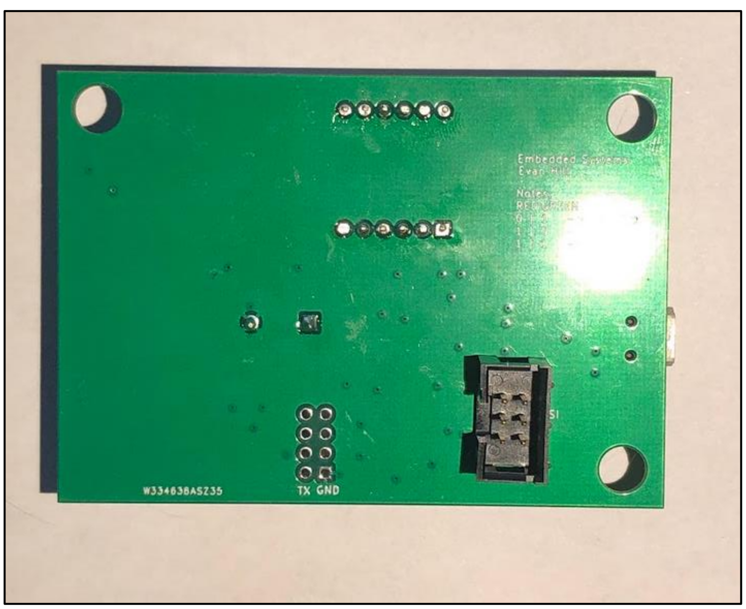

Figure 8 – Fully-Soldering PCB Bottom-view

Figure 9 – Fully-Soldering PCB Side-view

Software Development

Enclosure Design

Overall, the enclosure design and result went well. Thankfully, KiCad provides a measurement tool to measure the dimensions of components and PCB boards via the 3D model. With this, I was able to design an enclosure to exact specifications, and with the .step file for the PCB from KiCad, I was able to verify that the PCB would fit inside the enclosure perfectly by rendering all 3 components in a SolidWorks assembly. Unfortunately in the end, I didn't realize that the ISP programmer would be soldered on the bottom side of the PCB, instead of the front, where I designed an access hole for it. This ultimately means that the enclosure does not work for the PCB, however, the top of the enclosure fits perfectly to the PCB, and means that with minor adjustments to accommodate for the ISP, the enclosure should fit.

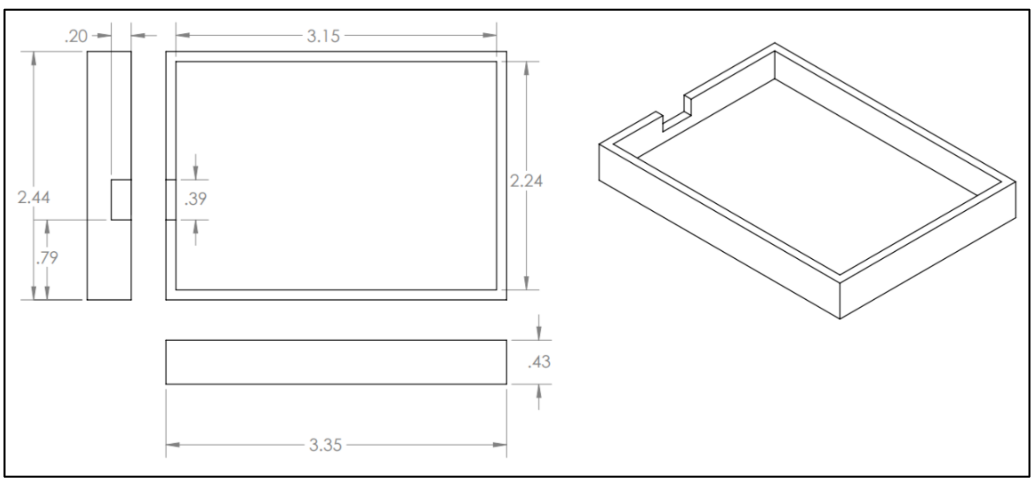

Figure 10 – Enclosure Bottom Design

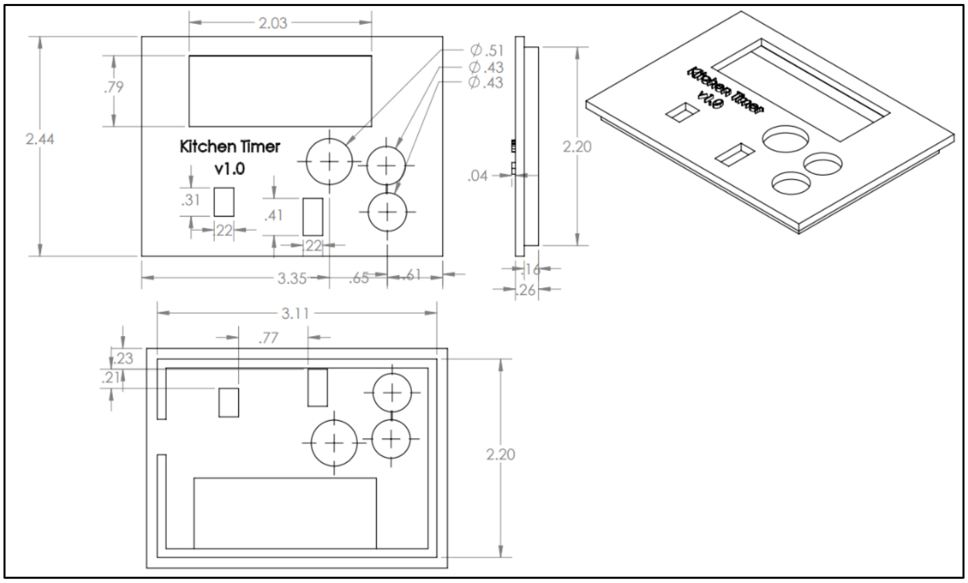

Figure 11 – Enclosure Top Design

Figure 12 – Enclosure Bottom 3D Print Top-view

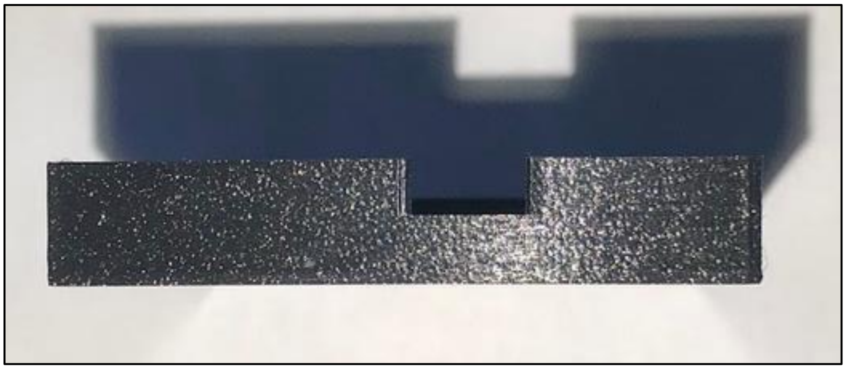

Figure 13 – Enclosure Bottom 3D print side-view

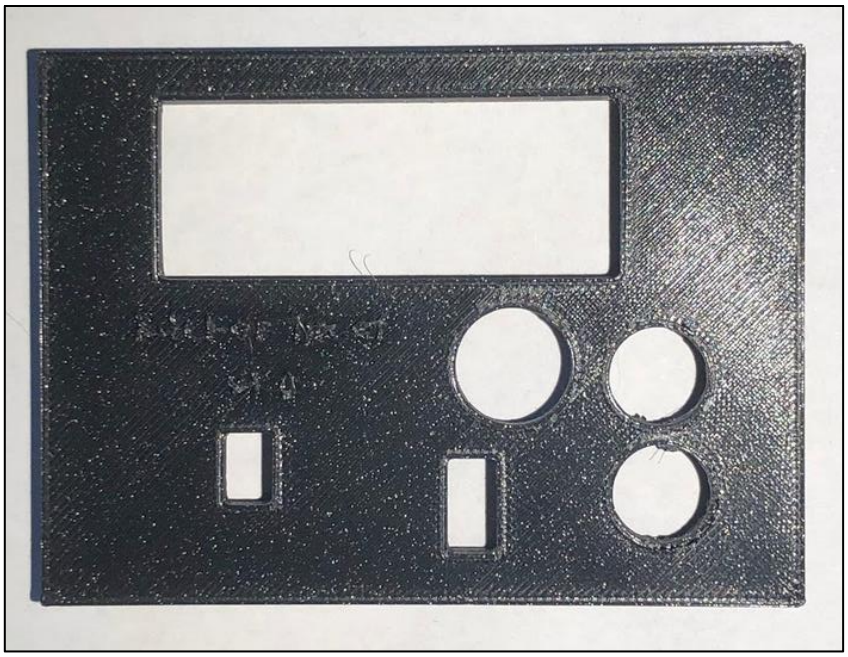

Figure 14 – Enclosure Top 3D print top-view

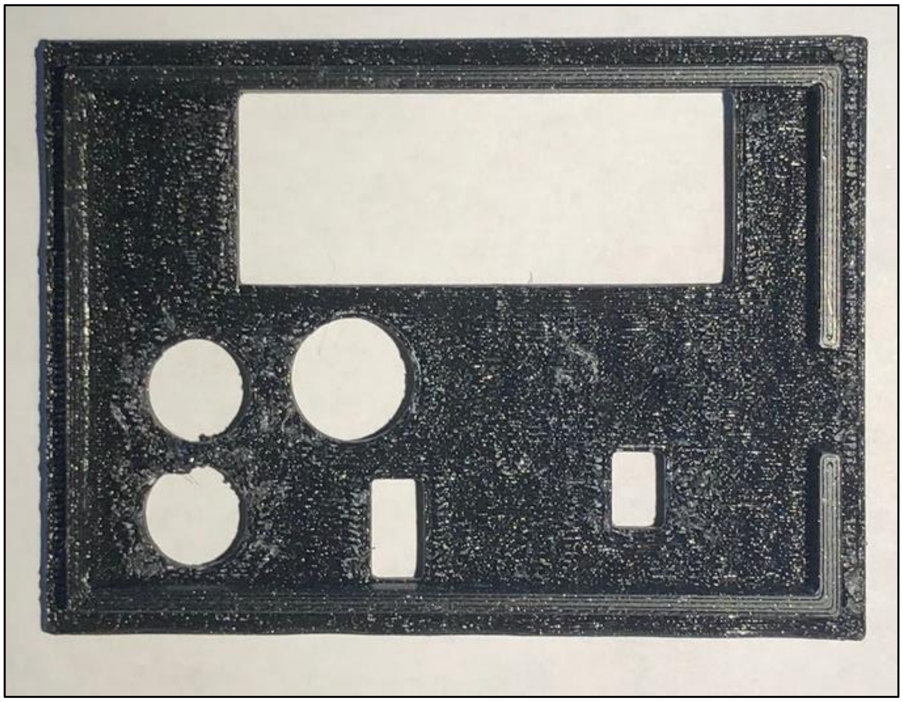

Figure 15 – Enclosure Top 3D print bottom-view

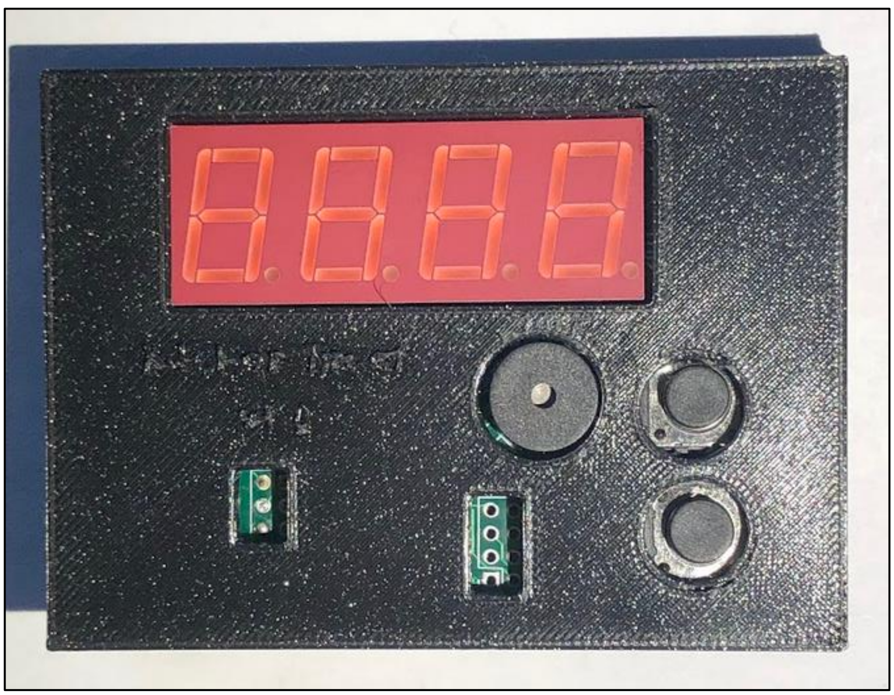

Figure 16 – Enclosure with PCB top-view

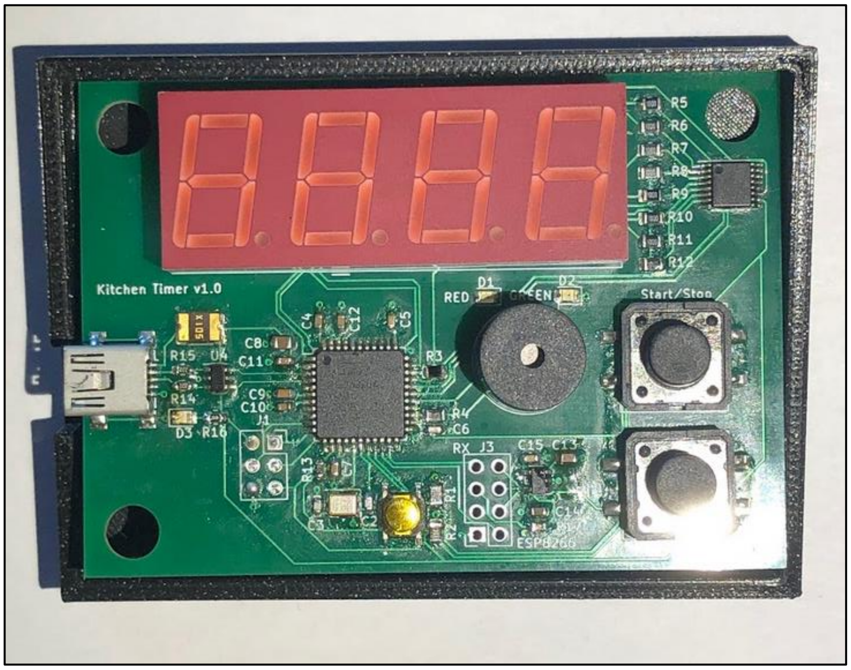

Figure 17 – Enclosure bottom with PCB top-view

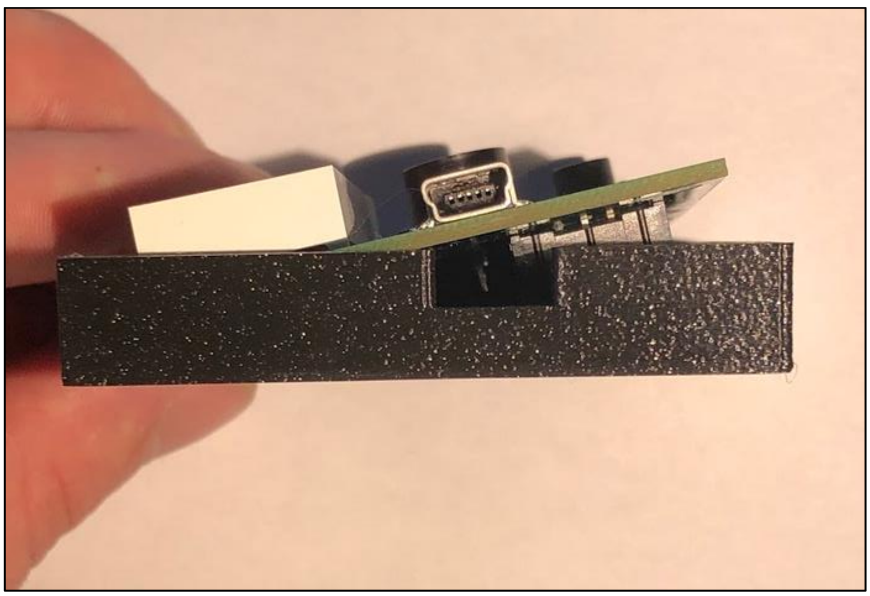

Figure 18 – Enclosure bottom with PCB side-view

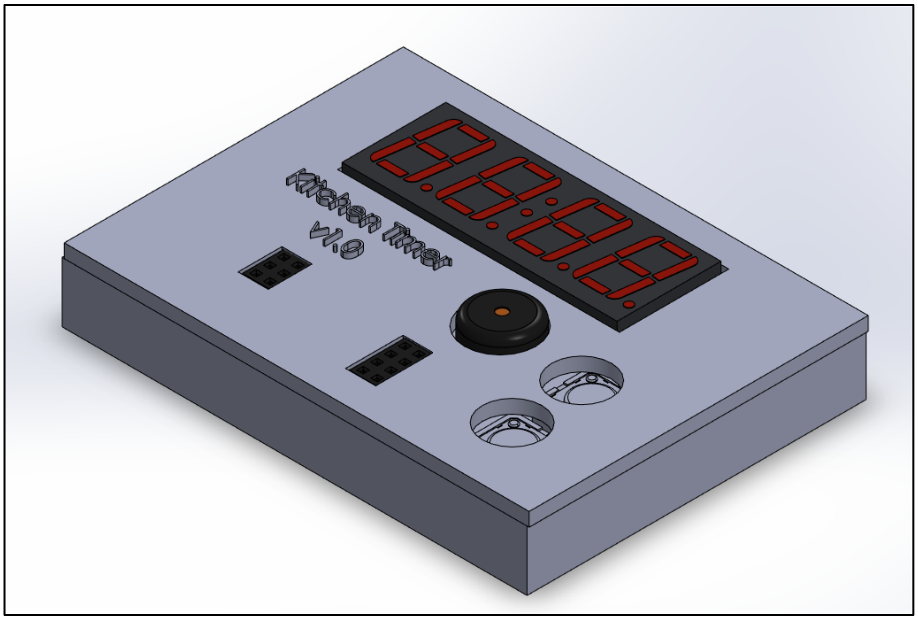

Figure 19 – PCB and Enclosure Assembly 3D rendering in SolidWorks