Phase B – Kitchen Timer Beta Release

ENCE-4231 Embedded Systems

May 4, 2023

By: Eric Jacobs

**Project requirements**

The initial project requirements for the kitchen timer started out very basic. The kitchen timer would be able to be set to any time up to one hour (60:00). The timer could be manually set and would alert the user with a noise when the time ran out. The current time left on the timer could be viewed on a display. After the first prototype, additional requirements were added to the kitchen timer. One new requirement was that the timer must be able to connect to nearby devices such as phones or computers and be controlled by them. Another new requirement

**System Design**

The system design for the kitchen timer initially was rather basic and was designed around meeting the initial project requirements. Several additional design decisions were made regarding how the kitchen timer would be designed. One of the first design choices was that any debouncing of the buttons would be done in software rather than adding additional hardware such as capacitors to debounce the button circuits. Another design decision was that a shift register would be used to control the 7-Segment display. This choice was made because the shift register would reduce the complexity of interacting with the display and would not take up a large number of pins on the microprocessor. The initial system design contained the microprocessor, buttons, LEDS, a buzzer, and a display. The initial board was programmed and powered through a USB port.

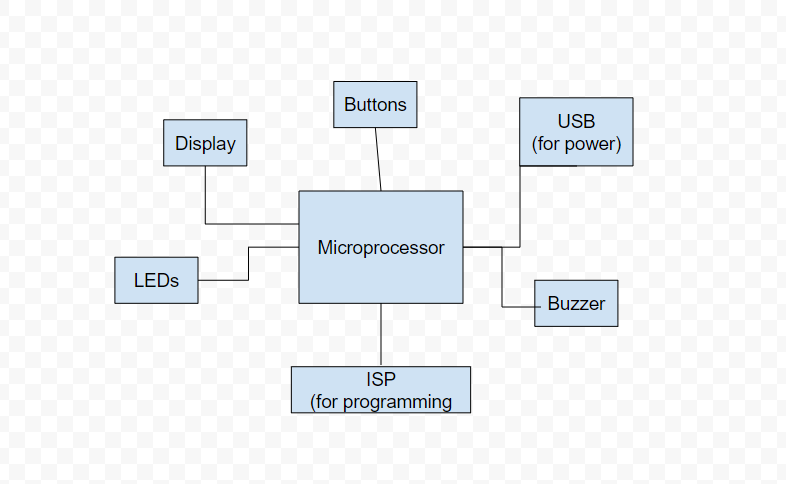


Figure 1 - Initial Design Block Diagram

After the first prototype was tested, the initial design was refined and improved in numerous ways. The new requirements were also considered when making the new design.

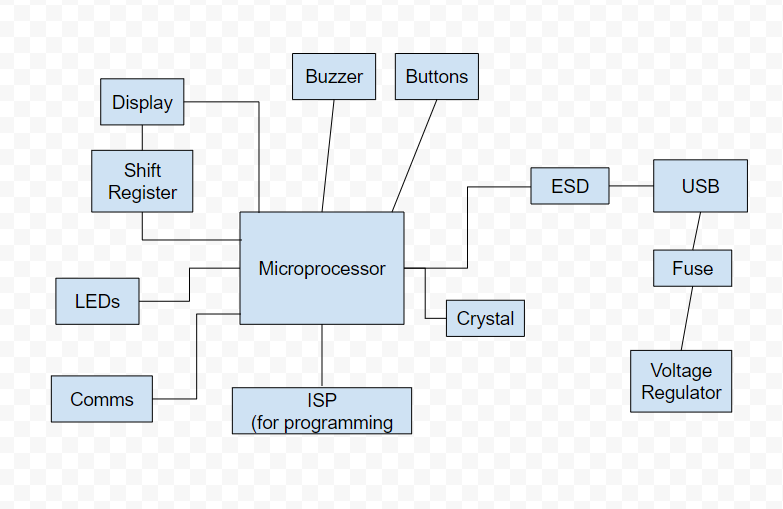


Figure 2 - Second Design Block Diagram

The second design contained all of the things from the initial design, but also incorporated various protective measures such as the ESD protection, Fuse, and Voltage regulator for the USB power/program input. Additionally, the shift register was added to the design and a communication system as added to meet the new requirements.

**Components Selection**

The Kitchen timer contained many different components including resistors, capacitors, a 7-Segment Display and several other parts. These parts were searched for on the website and suitable parts were selected for the timer. For the initial prototype, these parts included:

Arduino Uno/Mega Qt: 1

• 7-Segment Display ($1.60) Qt: 1

• Shift Register 8-Bit (SN74HC595 ($1.05) Qt: 1

• Buttons ($0.55/unit) Qt: 2

• LEDs Qt: 2

• Resistors

• Capacitors

• Mini Speaker ($2.10)

For the improved model, a few additional parts were added including a voltage regulator, a fuse, an ESD chip, and several other parts The list of chosen components can be found below.

* Buttons: [Mouser Electronics](https://www.mouser.com/ProductDetail/CK/PTS125SM43SMTR2LFS?qs=IL6XaBr%252B7X9OCHHnVhtGtA%3D%3D) ($0.53) Qt: 2
* Reset Button: [Mouser Electronics](https://www.mouser.com/ProductDetail/CK/PTS526-SM08-SMTR2-LFS?qs=UXgszm6BlbFgRZzlfGAMFg%3D%3D) ($0.13) Qt: 1
* Microprocessor: [Mouser Electronics](https://www.mouser.com/ProductDetail/Microchip-Technology/ATMEGA32U4-AU?qs=SSucg2PyLi7mKWjHIsNJ3w%3D%3D&mgh=1&gclid=Cj0KCQjwr82iBhCuARIsAO0EAZxfJ9488bka8xAow2-Z2c4hT_TcNGKI2Ijz8MUJdJ85QklEKC0AmFoaAoK9EALw_wcB) ($5.61) Qt: 1
* USB Connector: [Digikey](https://www.digikey.com/en/products/detail/gct/USB2066-05-RBHM-15-STB-00-01-A/9859703?utm_adgroup=USB%2C%20DVI%2C%20HDMI%20Connectors&utm_source=google&utm_medium=cpc&utm_campaign=Shopping_Product_Connectors%2C%20Interconnects&utm_term=&utm_content=USB%2C%20DVI%2C%20HDMI%20Connectors&gclid=Cj0KCQjwr82iBhCuARIsAO0EAZzJ3jvRgFb4Y8DK-t6so2ugBNwET87yoYxLBRj9sfDKtOqghrpFLVkaArLlEALw_wcB) ($0.71) Qt: 1
* Shift Register 8-Bit (SN74HC595): [Sparkfun](https://www.sparkfun.com/products/13699) ($1.05) Qt: 1
* 7-Segment Display: [Sparkfun](https://www.sparkfun.com/products/11405) ($1.60) Qt: 1
* Mini Speaker: [Sparkfun](https://www.sparkfun.com/products/7950) ($2.10) Qt: 1
* ESD chip: [Mouser Electronics](https://www.mouser.com/ProductDetail/STMicroelectronics/USBLC6-2SC6?qs=po45yt2pPpu%2FhNIlwQdTlg%3D%3D&gclid=Cj0KCQjwr82iBhCuARIsAO0EAZwnBHkMzdQFMUwgeY1b0LwQK9PFfOMFQMHEExxpOFfXjX_ddHk9zL4aAg55EALw_wcB) ($0.59) Qt: 1
* Voltage Regulator: [Mouser Electronics](https://www.mouser.com/ProductDetail/Texas-Instruments/LP2985IM5-3.3?qs=7lkVKPoqpbYyrbwrqf2JcA%3D%3D) ($1.37) Qt: 1
* Resistors Qt: 17
* Capacitors Qt: 15
* Fuse Qt: 1
* LEDs Qt: 3
* Wi-Fi Module: [Amazon](https://www.amazon.com/HiLetgo-Wireless-Transceiver-Development-Compatible/dp/B010N1ROQS/ref=sr_1_2_sspa?crid=1YL5QIRZVAPZX&keywords=esp8266&qid=1680154844&sprefix=esp8266%2Caps%2C180&sr=8-2-spons&psc=1&spLa=ZW5jcnlwdGVkUXVhbGlmaWVyPUEyRFNHQUNLTVMyREpNJmVuY3J5cHRlZElkPUEwMDA3NzM0MjE1RjZaVlNHM1M5UCZlbmNyeXB0ZWRBZElkPUEwMDc0Nzc2MUUyTFpMTExUNjZFWCZ3aWRnZXROYW1lPXNwX2F0ZiZhY3Rpb249Y2xpY2tSZWRpcmVjdCZkb05vdExvZ0NsaWNrPXRydWU=) ($9 for 3) Qt: 1

**Build Prototype**

The prototyping stage of the kitchen timer involved the development of the kitchen timer as an Arduino shield. The Arduino shield would be able to use the microprocessor on an Arduino UNO or Arduino Mega to control and power the kitchen timer. The prototype was used to implement a rudimentary version of the kitchen timer without the difficulty of working with a microprocessor. Using Arduino’s coding library, a program was designed to control the kitchen timer. The Arduino shield implementation of the kitchen timer could be started or stopped with the press of a button and the time could be increased with the press of a different button. A 4x 7-Segment display displayed the current time left on the timer. When the time ran out, the buzzer on the PCB would play a noise to alert the user.

**PCB Design**

After completing the prototyping stage of this project, the next step was to design a custom-made PCB. The PCB contained all of the new changes that were made to the original design. The schematic is broken up into several sections which keep it organized and make it easy to locate a specific part of the circuit. From the Arduino shield schematic, the shift registers, 7-Segment display, buttons, LEDs, and buzzer circuits all remained the same. Additional circuits for the crystal, ESD chip, USB connector, Board programmer, Wi-fi chip connector, and voltage regulator were added to the schematic.

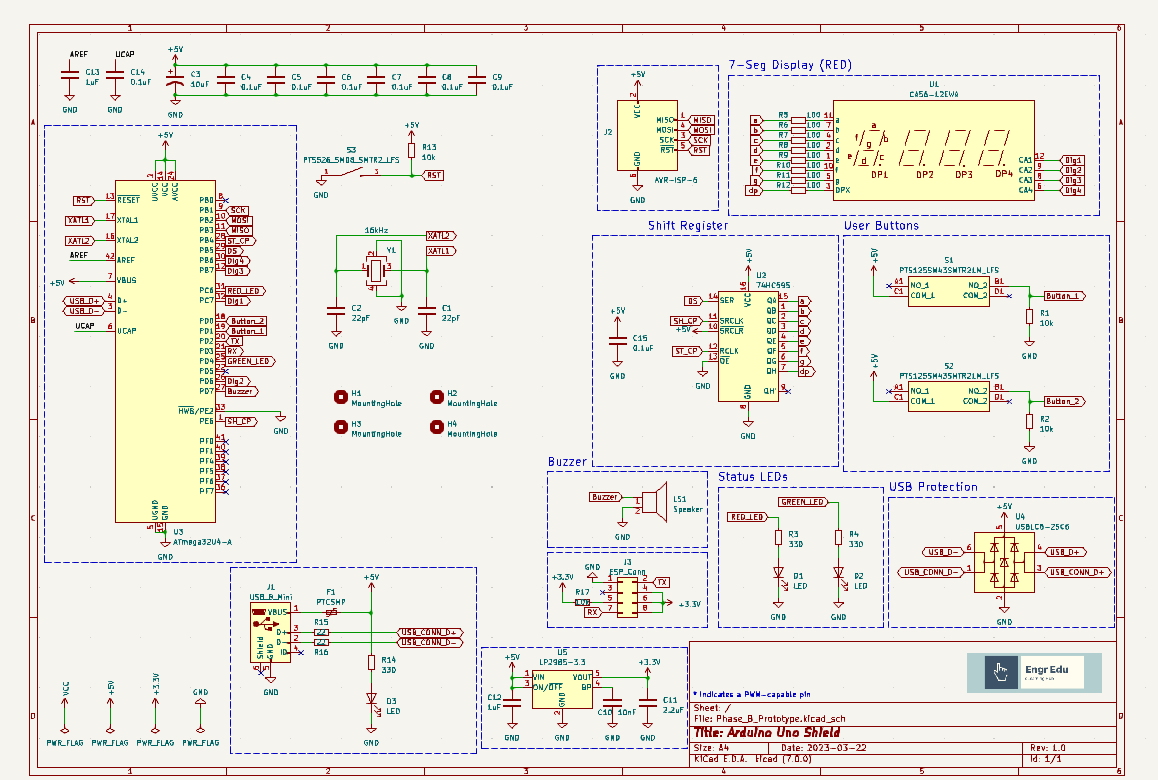


Figure - PCB Schematic

After completing the schematic for the board and checking that it follows all electrical rules. The PCB layout was then decided, and the board was routed. In this design, the 7-Segment display was placed at the top of the board away from other components. This simplified the routing since the 7-Segment display has quite a lot of pins in use. The user buttons are placed on the right side of the board. The microprocessor is placed in a centralized location of the board which allows for easy routing to components on all sides of the board. All of the USB related circuitry is on the left side of the board including the USB connector, ESD chip, and voltage regulator. Since the USB connector needs to be routed with a differential pair, it was important for the components of the USB circuit to be close to one another. The programmer and Wi-Fi chip connectors were placed at the bottom of the board with the Wi-Fi chip connector oriented in a way that the chip would face upwards. This design choice was made so that the Wi-Fi chip would not stick out of the side of the board making it easier to design an enclosure for the board. The parts of the PCB were then routed with a goal of having as little ground plane disturbance as possible. Some possible improvements of this board would be to find a way to route the last trace to the 7-Segment display so that it does not interfere with the ground plane under the differential pair from the USB connector. Additionally, this design could be made to be much more compact.

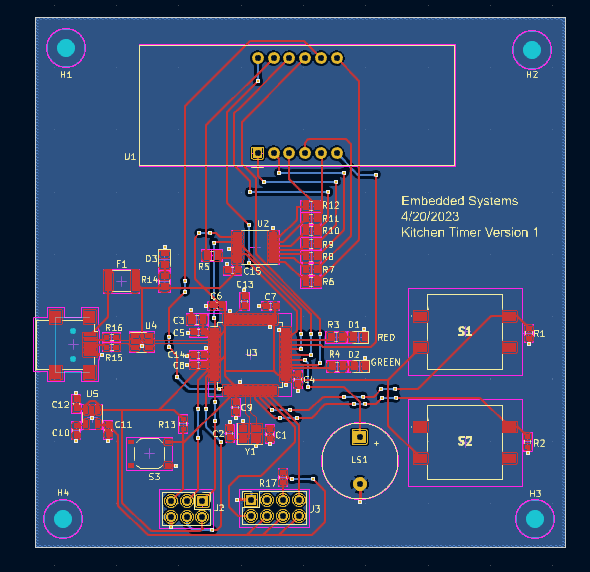


Figure - Routed PCB board

**Assembly Stage**

After designing the.PCBs in KiCAD, the gerber files for the boards were sent to a fab house to be produced. When the fabricated boards arrived, the assembly stage started. The components of the board were soldered one by one starting with the microchips. Then, each of the coupling capacitors were soldered. Once these parts were soldered, the power LED was soldered so that the board could be plugged in and tested for power. After verifying that the boards had power, the red and green LEDs were soldered allowing a program to be tested on the board. Additionally, the programmer connector was soldered so that a boot loader could be uploaded to the board. The boot loader was uploaded to the board and a program designed to blink the LED was uploaded. Once certain that the board was working with the program, the remainder of the components were soldered including the buttons, the 7-segment display, and the buzzer.

A green circuit board with a red display

Description automatically generated with low confidence

Figure - completely soldered PCB board

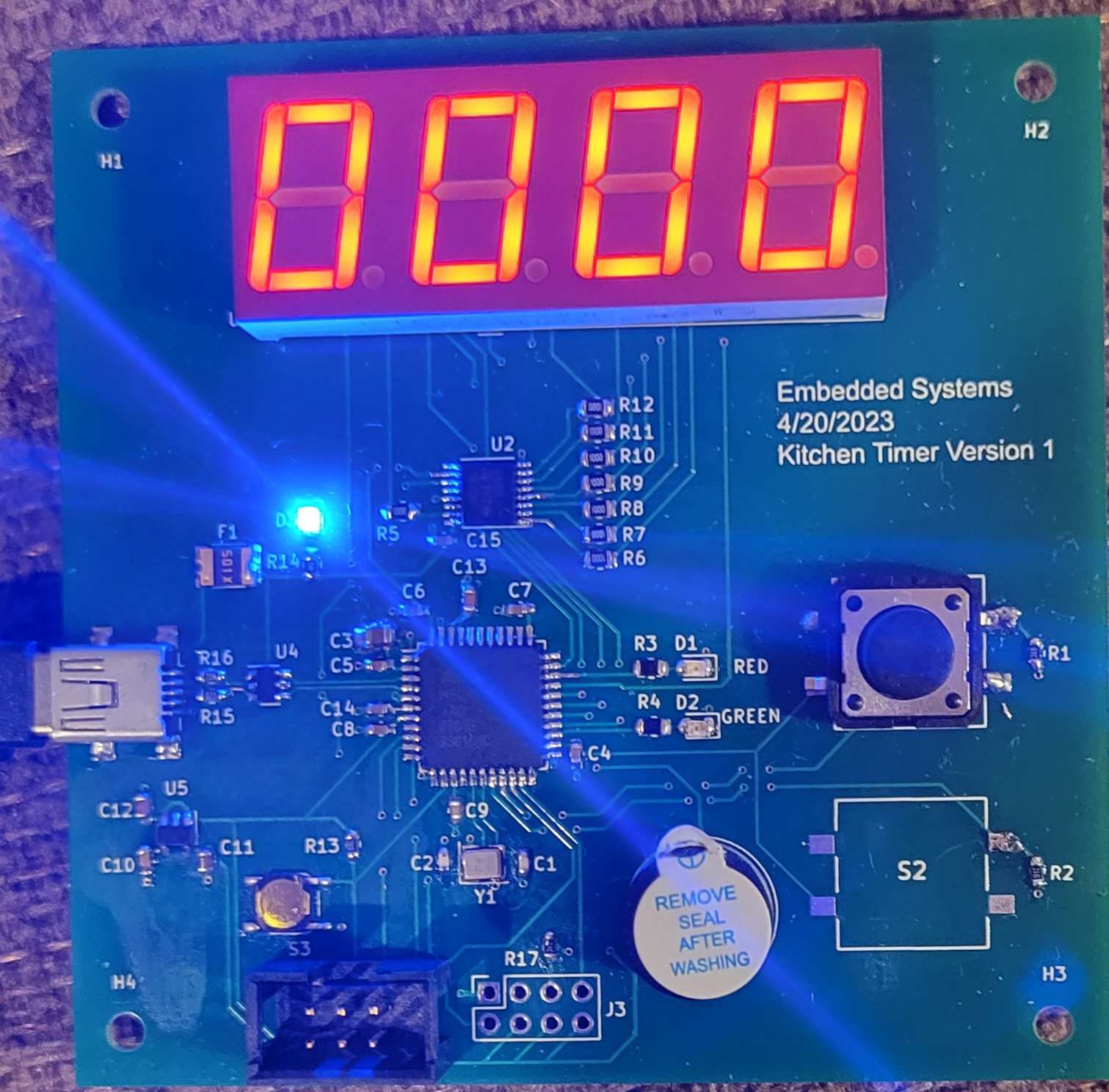


Figure - PCB board with power

**Software Development**

The software development of the kitchen timer consisted of several stages. The first stage of software development occurred during the prototyping period. With the Arduino shield PCB built, the software used to control the timer could be written. This software was responsible for interacting with the buttons, LEDs, buzzer, and display on the Arduino shield. The program would first check if either button was pressed. If the stop/start button as pressed, an interrupt would be triggered and the timer controlling the clock would be toggled on or off. This code would also turn on a green LED to indicate that the timer was paused. If the increment timer button was pressed, an interrupt would be triggered and then the time would be incremented. After checking both buttons, a switch case would be used to check the number in each digit of the display. This data was then displayed on the 7-Segment display using the pins connected to the shift register and the display. If the timer was not paused, the count would then decrease. If the timer reached 0 seconds, the buzzer on the PCB would be powered making an audible noise. After the second model was designed, additional code needed to be added to the program in order to ensure that the code functioned properly with the Wi-Fi chip. The new code was responsible for reading serial data from the Wi-Fi chip. An additional timer was initialized in order to periodically check if any serial data was being received. Any important data started with a ”$” and the program checked for this character. If the character was detected, the following characters were stored in an array until a newline character “\n” was detected. The data between the ”$” and “\n” was then compared to several phrases including “GET”, “SRT”, “STP”, etc. if any of these phrases was found to match the serial data, the corresponding task was completed. “STP” stops the timer, “STR” stars the timer, “GET” sends the current time. This code allowed the kitchen timer to interact with external devices via a website that communicated with the onboard Wi-Fi chip.

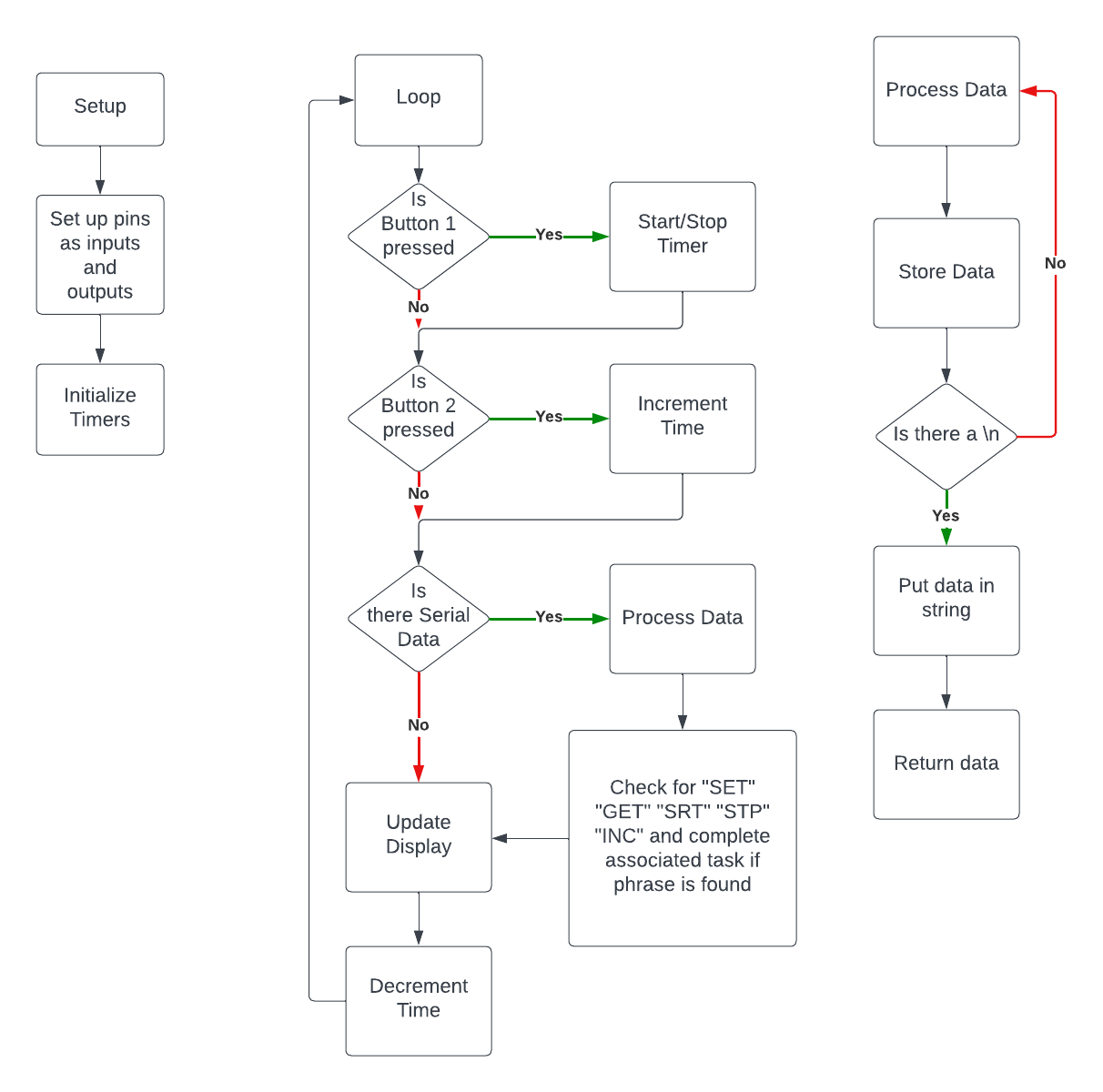


Figure - Code Flow chart

**Enclosure Design**

In order to protect the kitchen timer PCB and improve the aesthetics of the timer, an enclosure was designed for the PCB board to fit into. In order to reduce complexity, the enclosure designed for the kitchen timer was a simple box. The box was designed with enough clearance for the board to comfortably sit inside. Additionally, a small cutout on the side was designed in order to allow access to the USB connector which would be used to power/program the PCB. The enclosure can be seen below along with a 3D model of the PCB inside of it.

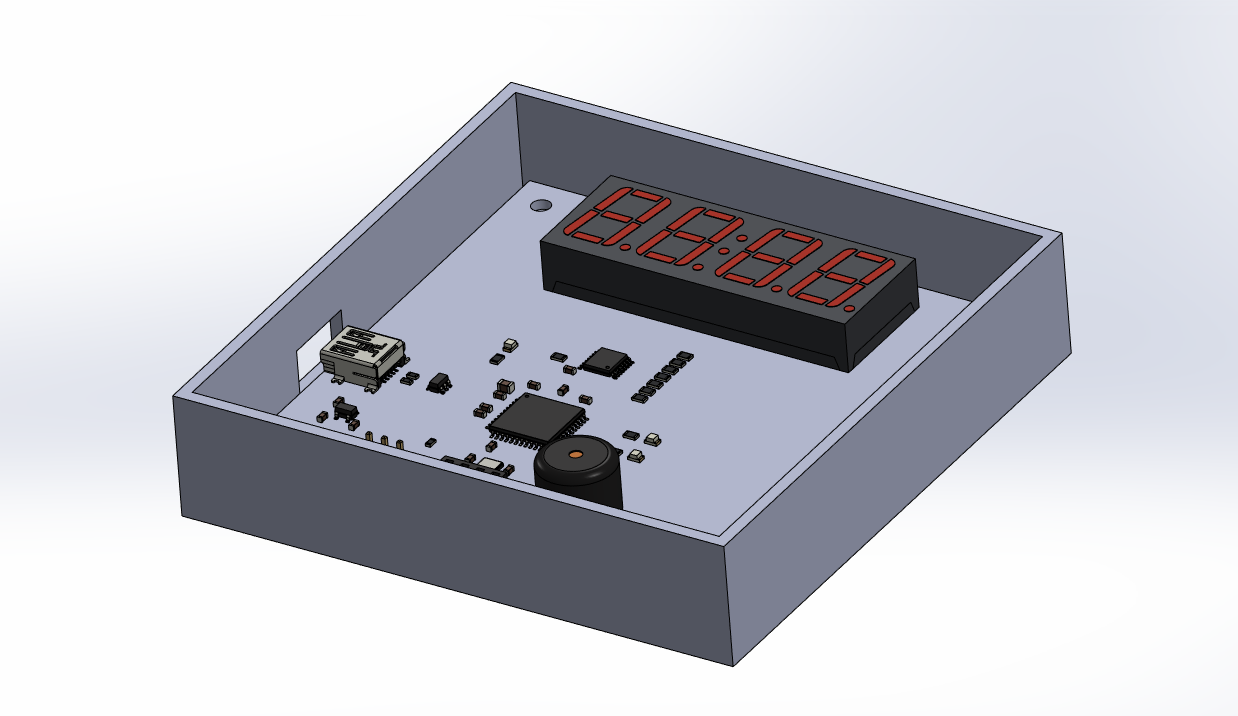


Figure - Enclosure Design with PCB model inside

While this design is quite basic, it protects the back of the PCB from whatever surface the kitchen timer is going to be placed on. Additionally, since the design can be 3D printed, it is able to be produced in numerous colors and several different materials which can improve the overall aesthetic and customer appeal for the system.