Kitchen Timer (IoT) – Phase B

By Harsha Karunanayaka

• Project Requirements

In this stage we develop a new PCB design based on the microprocessor ATMEGA32U4 that has similar CPU specifications as Arduino UNO but with more availability of inputs. Also, includes additional components to enhance the product i.e., interact the clock with Wi-Fi communication.

Hardware Requirements

- Buzzer
- User buttons.
- Status LEDs.
- 7-Segment Display.
- EPS8266 module for communication.

Software Requirements

- Turn on/off the time.
- Increment the timer.
- Ability to interact with time over the internet, phone, or computer using a web application.

Communication Requirements

- Ability to control the timer remotely.

• System Design

The block diagram was designed accordingly with the new microprocessor to meet the project requirements.

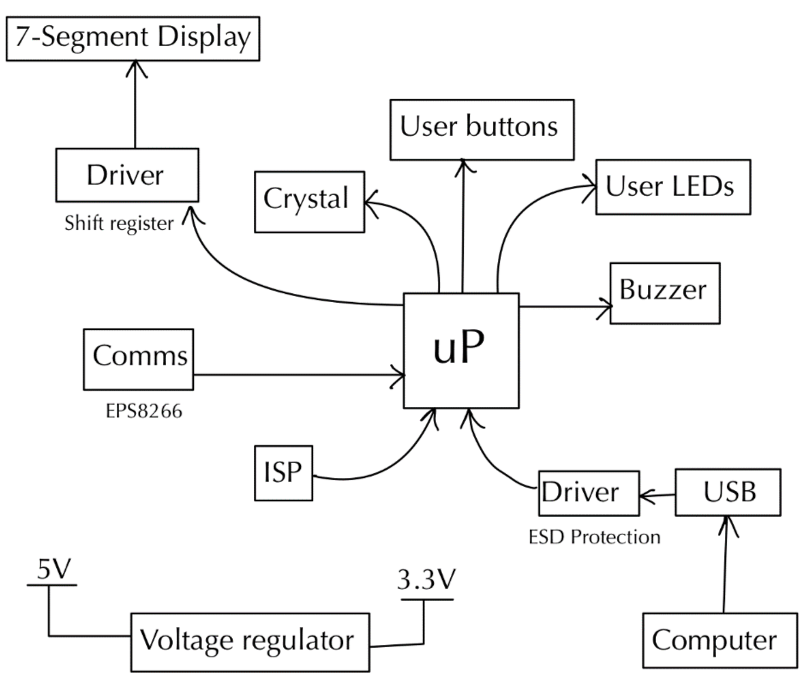

Figure 1: System Design

Microprocessor

The microprocessor used in this stage of the project is ATMEGA32U4. As support to the microprocessor module, ISP programmer, Crystal, and USB modules are added to the PCB design. The ATMEGA32U4 has an additional timer and additional pins that can be connected to the communication module.

Communication Module

EPS8266 is the communication module used in this project. The module works with a 3.3V supply; hence, a voltage regulator is required to drop the voltage. This module can create a Wi-Fi hotspot by itself or can be connected to any Wi-Fi network. In this project, the module is used to create its Wi-Fi network to interact with devices like computers and phones.

• Component Selection

In this stage, the following components are used to design the microprocessor module. To reduce the usage of the pins of the microprocessor, an 8-bit shift register is used as a driver for the 7-segment display.

Components

- x1 ATMEGA32U4 microprocessor chip
- x1 16MHz crystal
- x1 8-bit shift register (SN74HC595)
- x2 LEDs
- x2 Buttons
- x1 Buzzer
- x1 EPS8266 module
- Resistors and Capacitors

• Build Prototype

As the prototype, an Arduino shield is designed and built that can be connected to an Arduino UNO. The shield consisted of every component except the ATMEGA chip and the EPS communication module. Arduino UNO did not have enough pinouts to connect the communication module with other components. However, the EPS82866 module tested separately with the Arduino UNO without the shield.

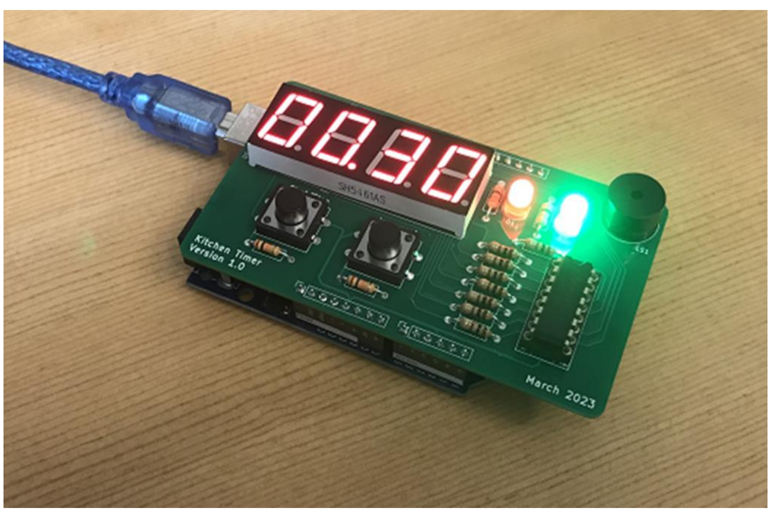

Figure 2: Arduino Sheild connected to the Arduino UNO

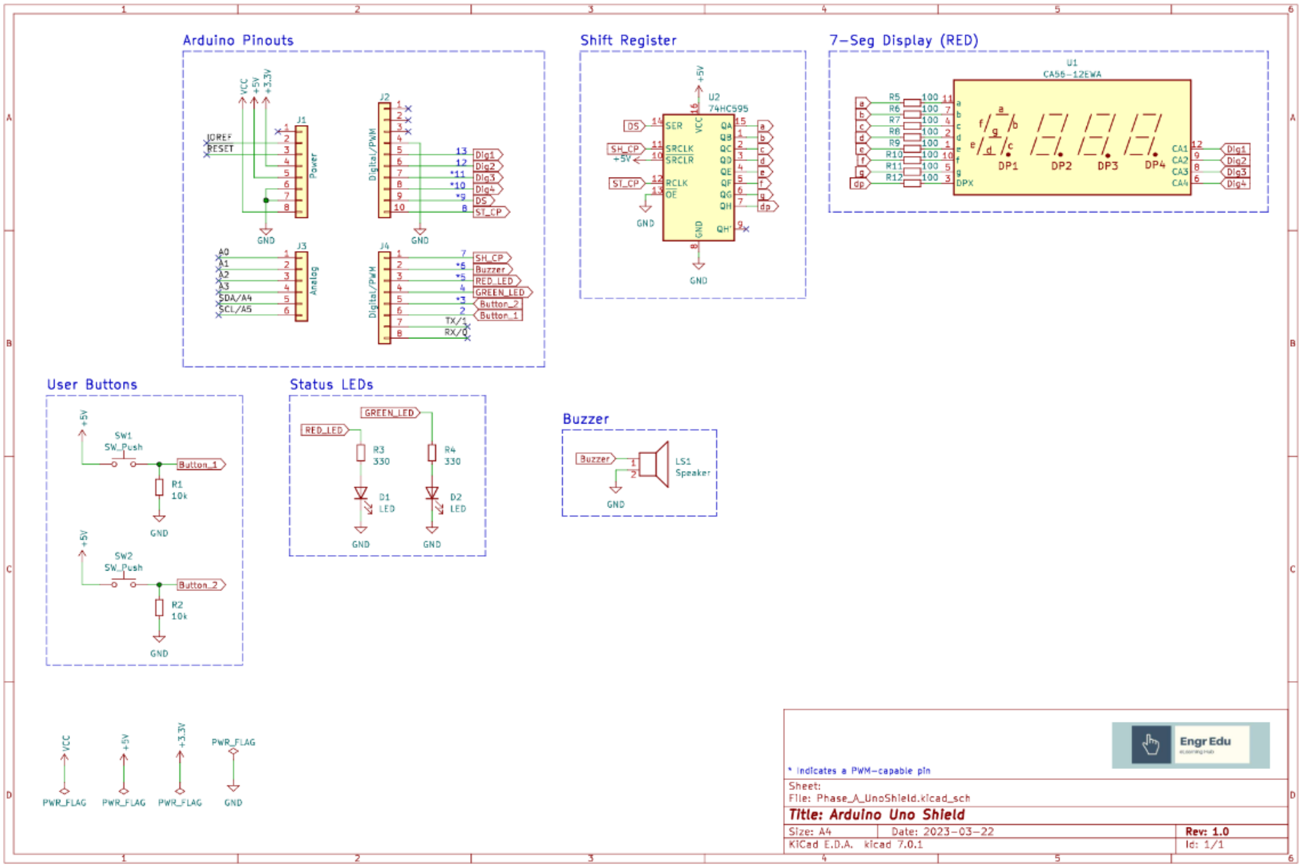

Figure 4: Schematic of the Arduino sheild

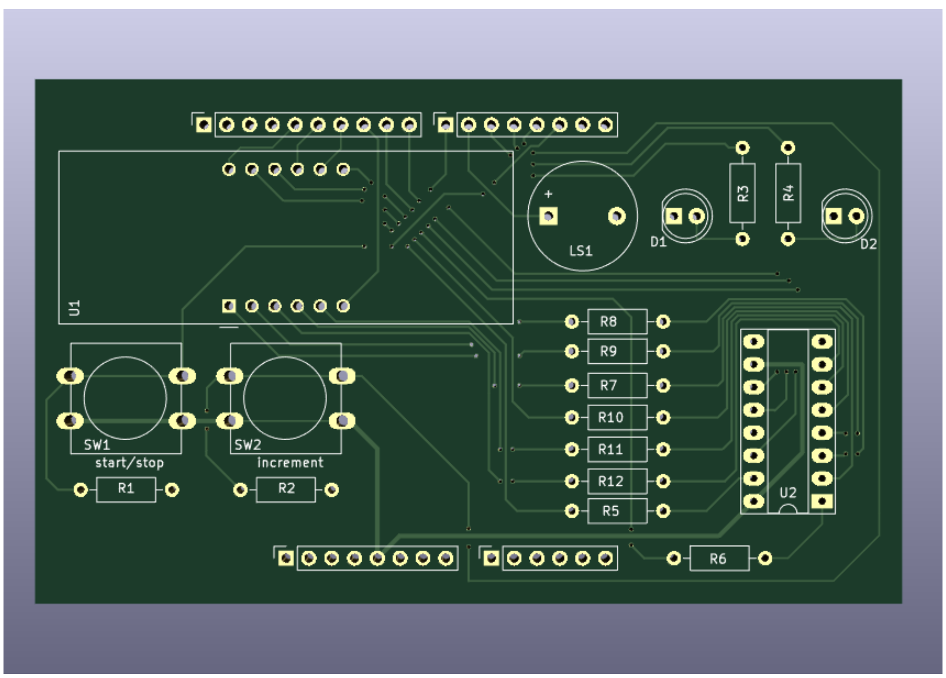

Figure 3: PCB of the Arduino sheild

- ## PCB Design

In this section schematics and PCB layout of the newly designed board are presented. The schematic diagram shows how the individual components are connected in the fabrication of the PCB. Most of the connections are similar to those in the prototype shield and instead of the Arduino board, the new microprocessor is assembled with the EPS communication module.

The size of the PCB is 100mm x 100mm where the bottom layer is used as the ground plane. Appropriated header pins are used to connect the EPS module and the IPS programmer.

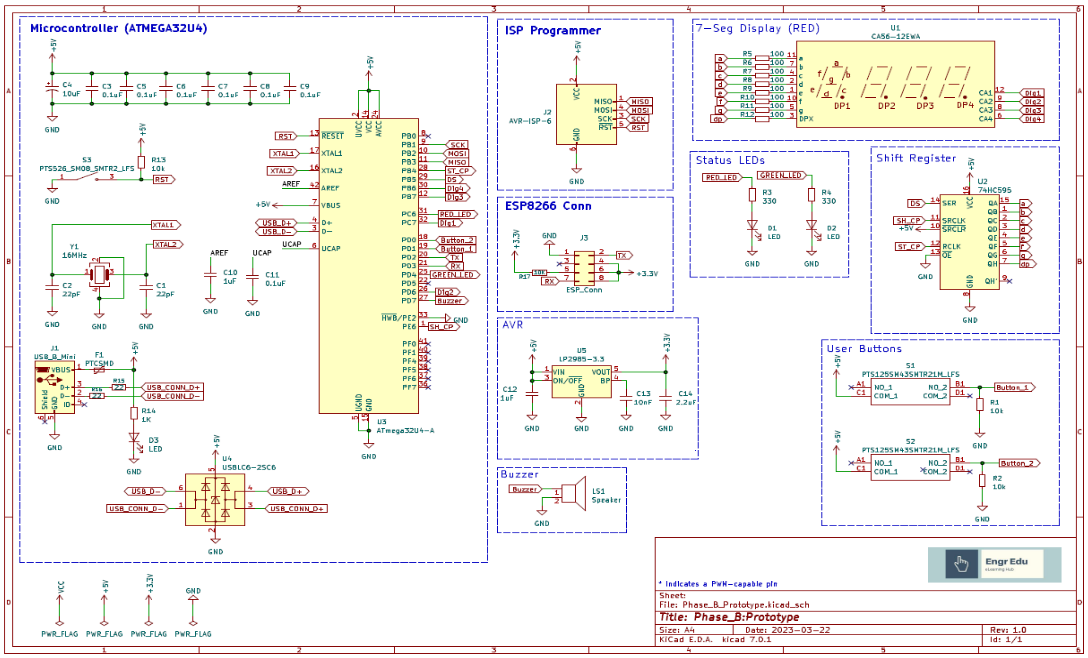

Figure 5: Schematics of the new PCB

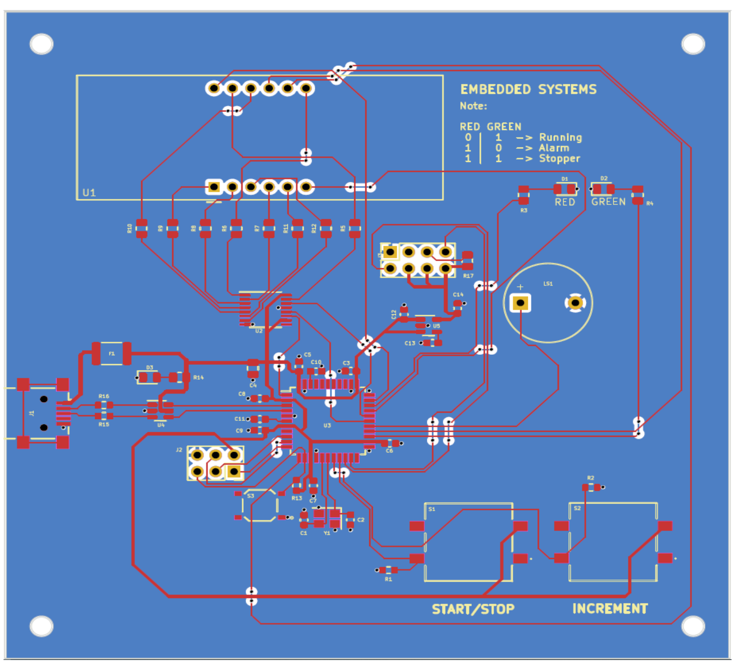

Figure 6:PCB layout

- ## Assemble Stage

In the assembly stage, all the components solder to the PCB and test with the final version of the code. Then, PCB is covered with an enclosure for finalizing the kitchen timer. The following figure shows the final rendering of the PCB with all the components.

Figure 7: PCB render. Top (left), Bottom(right)

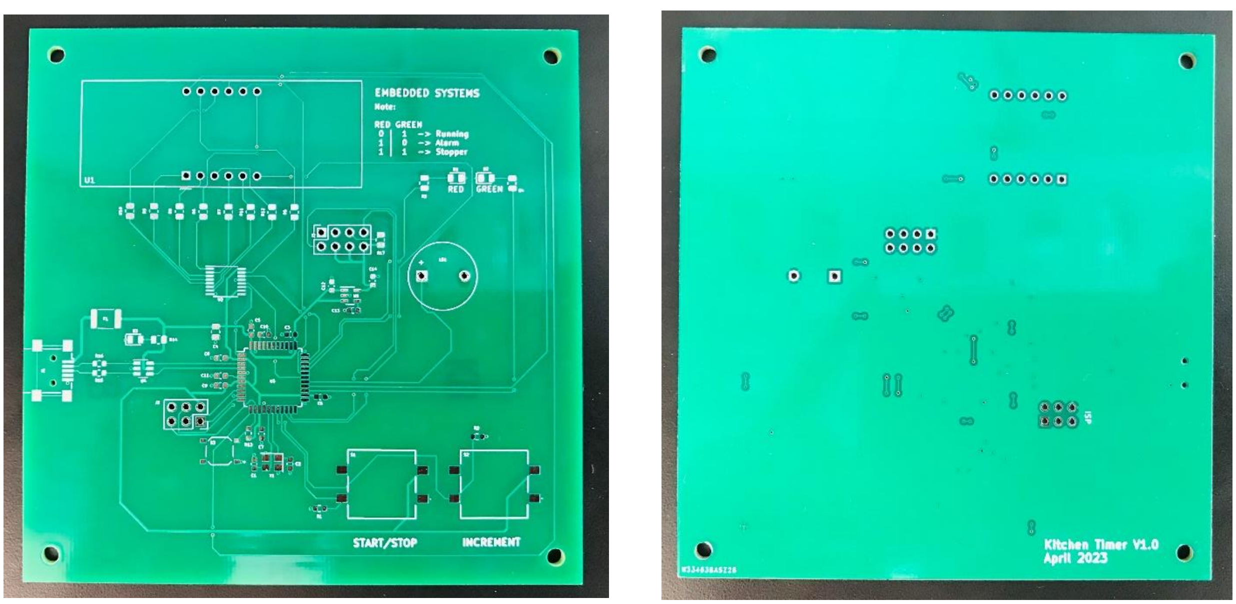

Figure 8: PCB before assembly. Top(left), Bottom (right)

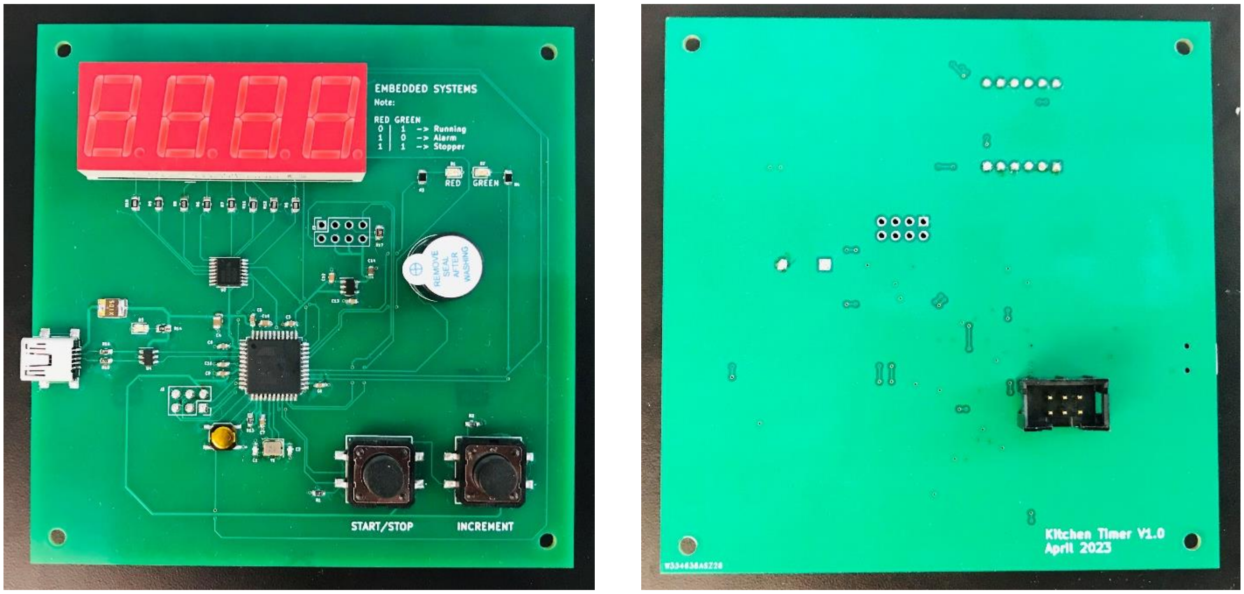

Figure 9: PCB after assebmbling the components. Top(left), Bottom(right)

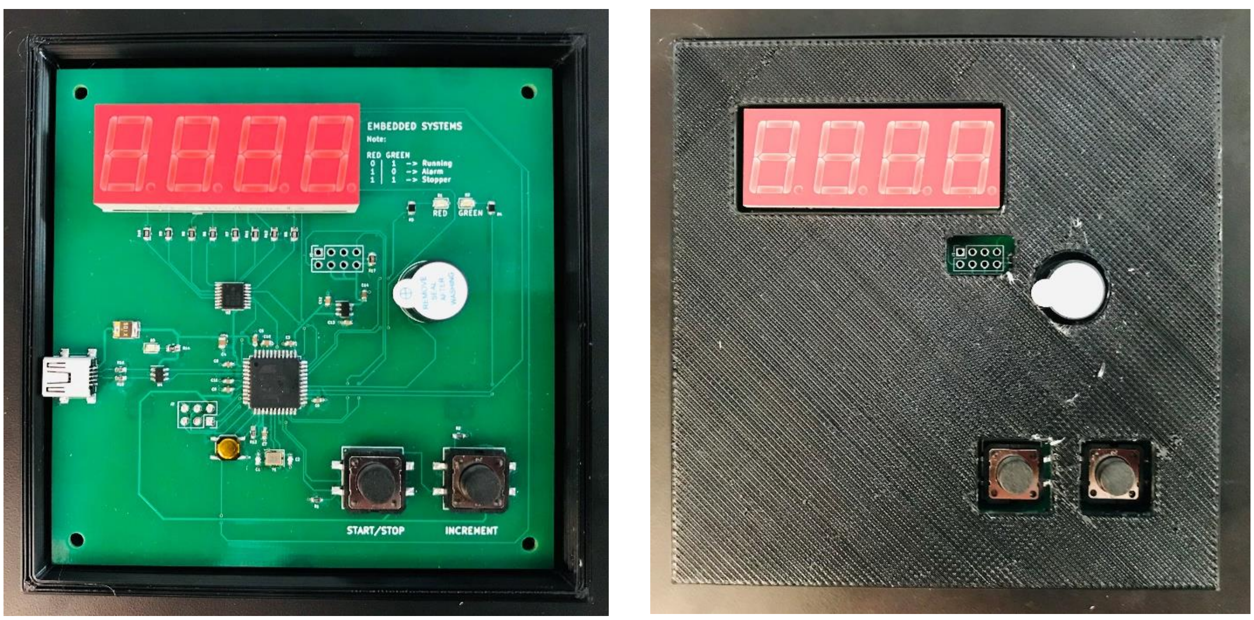

Figure 10: PCB with enclosure. Without enclosure top(left), With enclosure top(right)

- # Software development

This section explains how the code is developed with interrupts and the EPS module is used to communicate with the timer.

<u>With Interrupts</u>

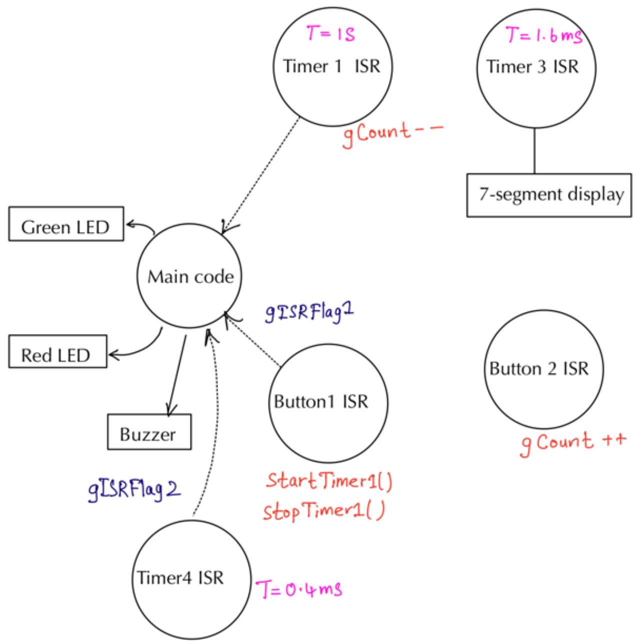

Figure 11: Code block diagram with the communication module

The code block diagram in Figure 12 shows how the code interacts with peripherals, communication with interrupt routines, and the main loop. Timer 1 is used to count down the timer after setting it to a specific value and the Timer 3 is dedicated to refreshing the 7-segment display every 1.6 ms. Button 1 which is the start/stop button, is used as an external interrupt for starting and stopping the timer. Also, button 1 used to increment the time is connected with another external interrupt routine to increment the timer count. This guarantee that the code runs concurrently without any delay.

<u>Integration of the EPS communication module to the existing code</u>

The ATMEGA32F4 chip consists of a 4th timer that uses here for the serial communication between the module and the microprocessor. It is set as an interrupt routine that runs every 0.4 ms reading the serial data received by the module. The following code block diagram shows how the main loop is interacting with the peripherals and timers.

In this code, the kitchen time can interact with a web application developed on a web server. In addition to the user command buttons, the timer can be started, stop, and incremented using this application. Also, a specific value for the timer can be set using this, and every second the value displayed on the timer is updated on the application.

The following figure shows how the main loop of the code is interacting with the external and internal interrupt routines, running the code with concurrency.

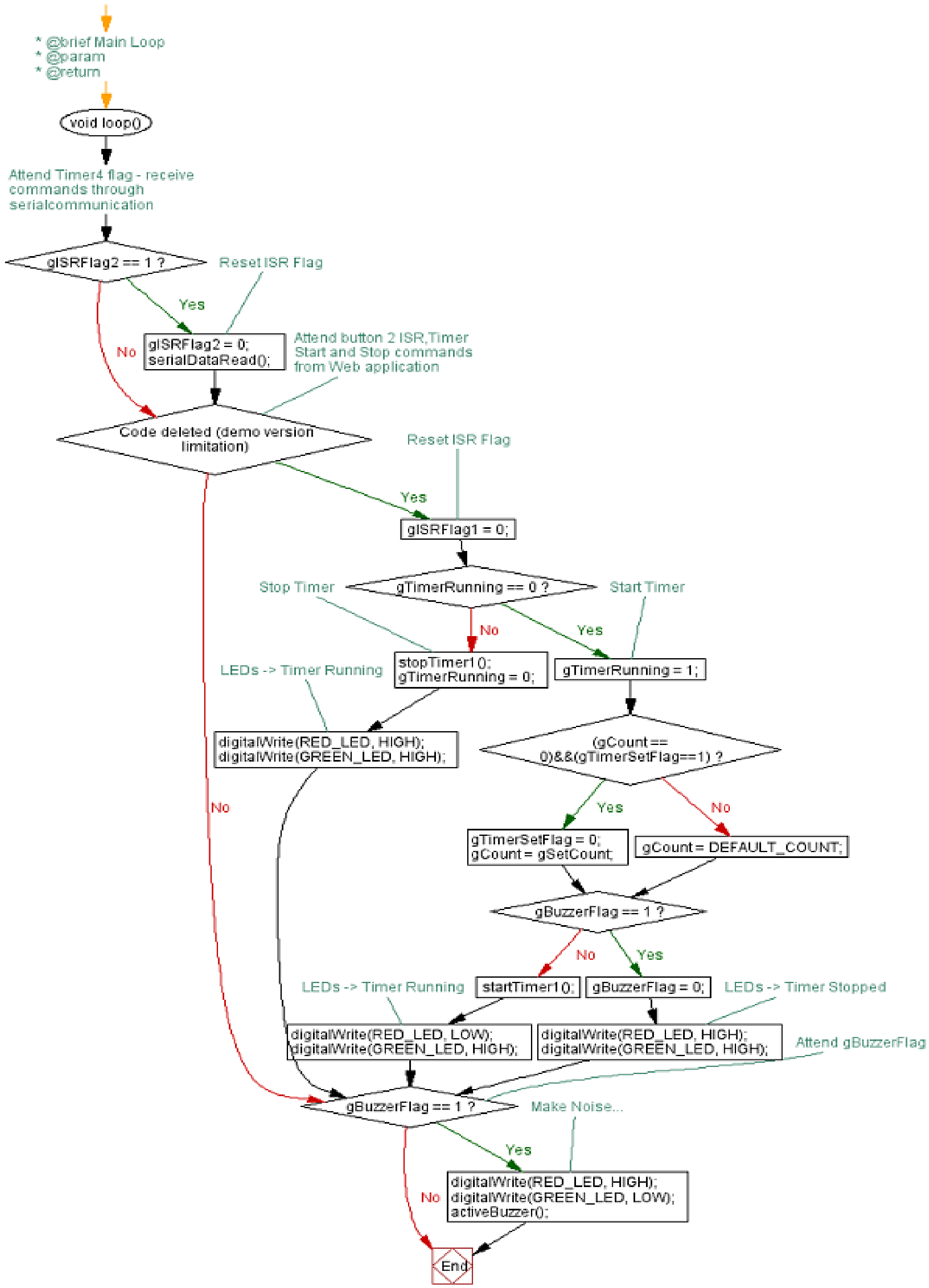

Figure 13: Flow chart of the main loop in the code

- ## Enclosure Design

The enclosure for the kitchen timer is designed with SolidWorks and the figures of the Enclosure design are presented in this section.

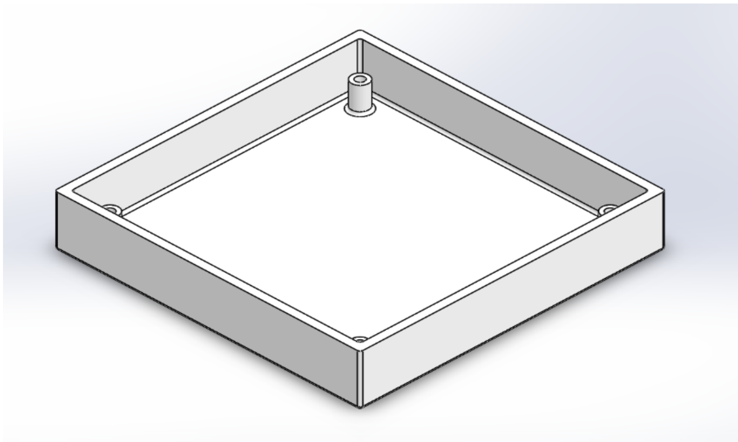

Figure 14: Enclosure Base

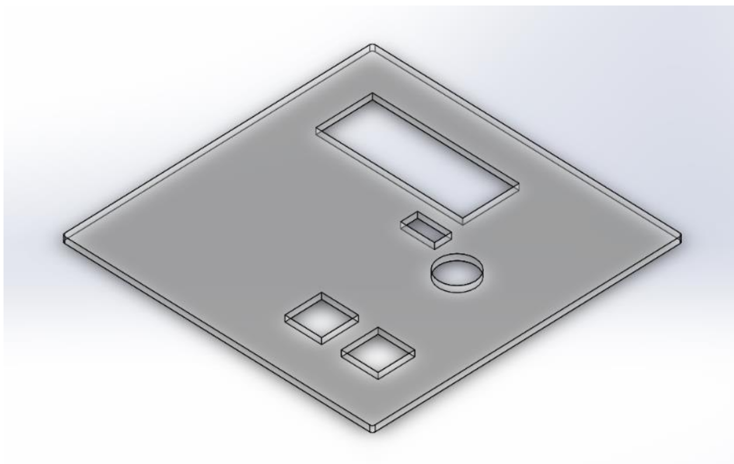

Figure 15: Enclosure Top

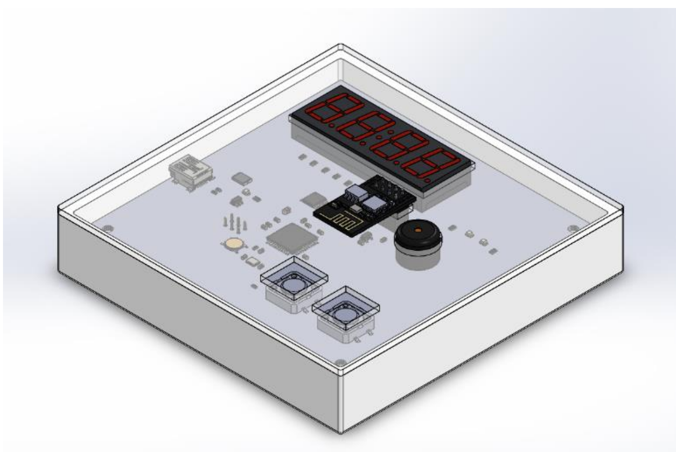

Figure 16: Kitchen Timer final assembly

3D printed final design of the enclosure

The following figures show the 3D-printed enclosure with PETG.

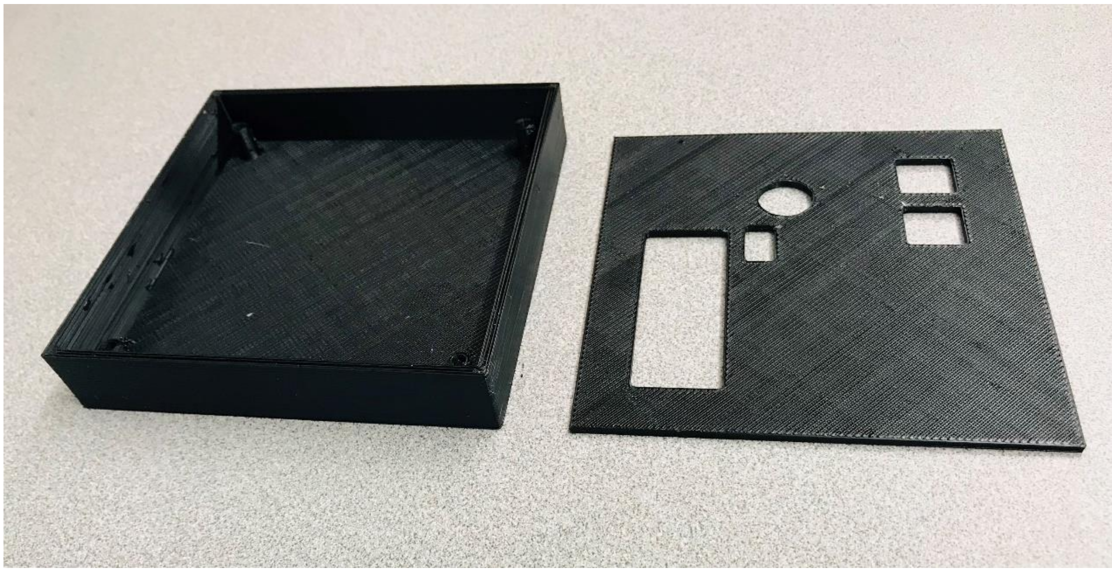

Figure 17: Enclosure base(Left), Enclosure top(Right)

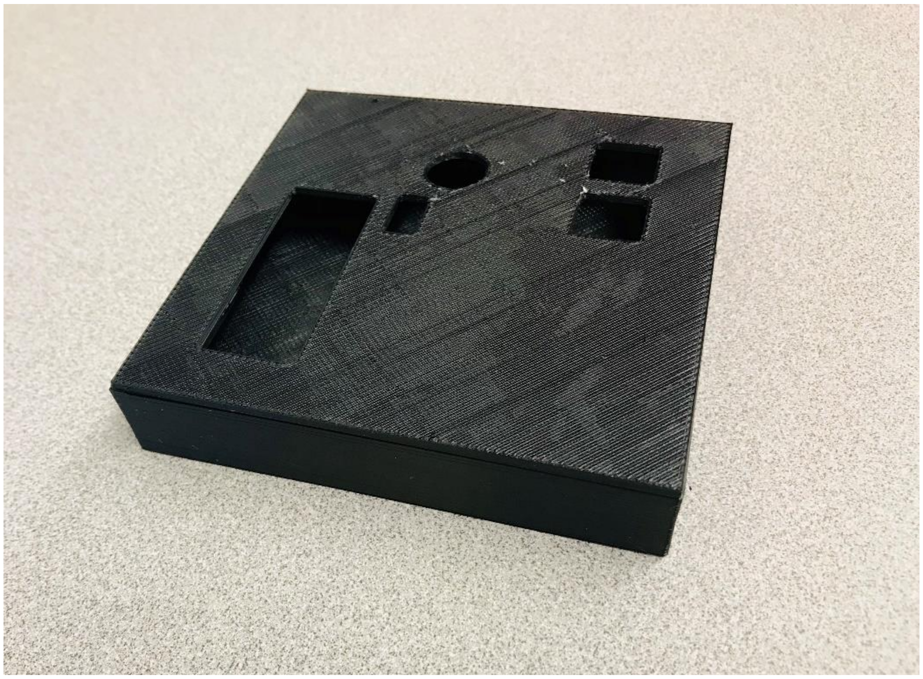

Figure 18: Final Assembly of the enclosure

- **References**

All the files related to this project, including schematics, PCB layout, the final version of the code, and CAD models are included in the following GitHub repository.
https://github.com/harshakarunanayaka/ENCE_3220_Class2023/tree/main/Phase_B