Noe Sheridan
Beta Release Phase B
Embedded Systems Programming
Dr. Goncalo Martins
4th of May 2023

Project Requirements:

- Cost
  - Requirement not specified by customer. Efforts still made to reduce cost.
- Size
  - Enclosure maximum size 10 by 10 cm.
  - PCB less than the enclosure so that it can fit inside.
- Limitations of Timer
  - Due to size of 7-Segment Display, only four digits of time can be displayed. For clarity purposes this means the maximum time input for the timer is 59 minutes and 59 seconds. (59:59).
- LED Amount
  - Two LEDs used on PCB to communicate with user.
- Button Amount
  - Three buttons total, including the resent button which resets the entire timer, one to increment the timer, and one used to start and stop the timer.
- Type of Buzzer
  -
- Debounce for Buttons
  - To minimize hardware requirements and maximize space on the PCB, the debounce will be done through software.
- Shift Register
  - To minimize the number of Microprocessor pins used, a shift register will be used to communicate between the 7-segment display and the microprocessor.
- How Powered
  - Currently powered through USB connection with the Arduino Uno microprocessor. In the Future will be directly powered through the USB.

System Design

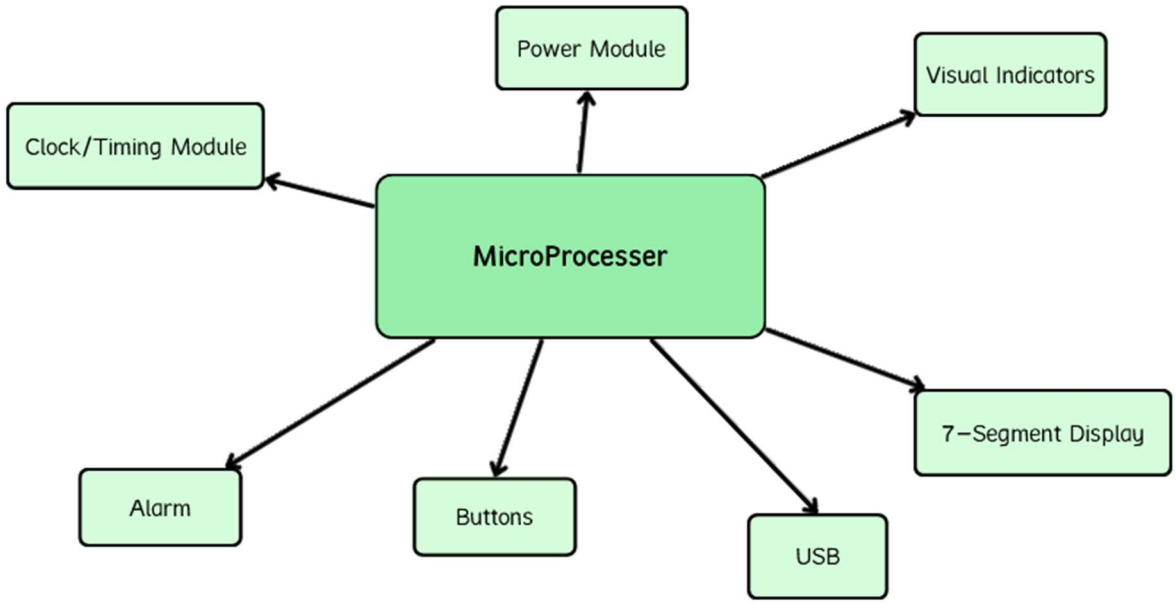

Figure 1: Block Diagram Version 1.0

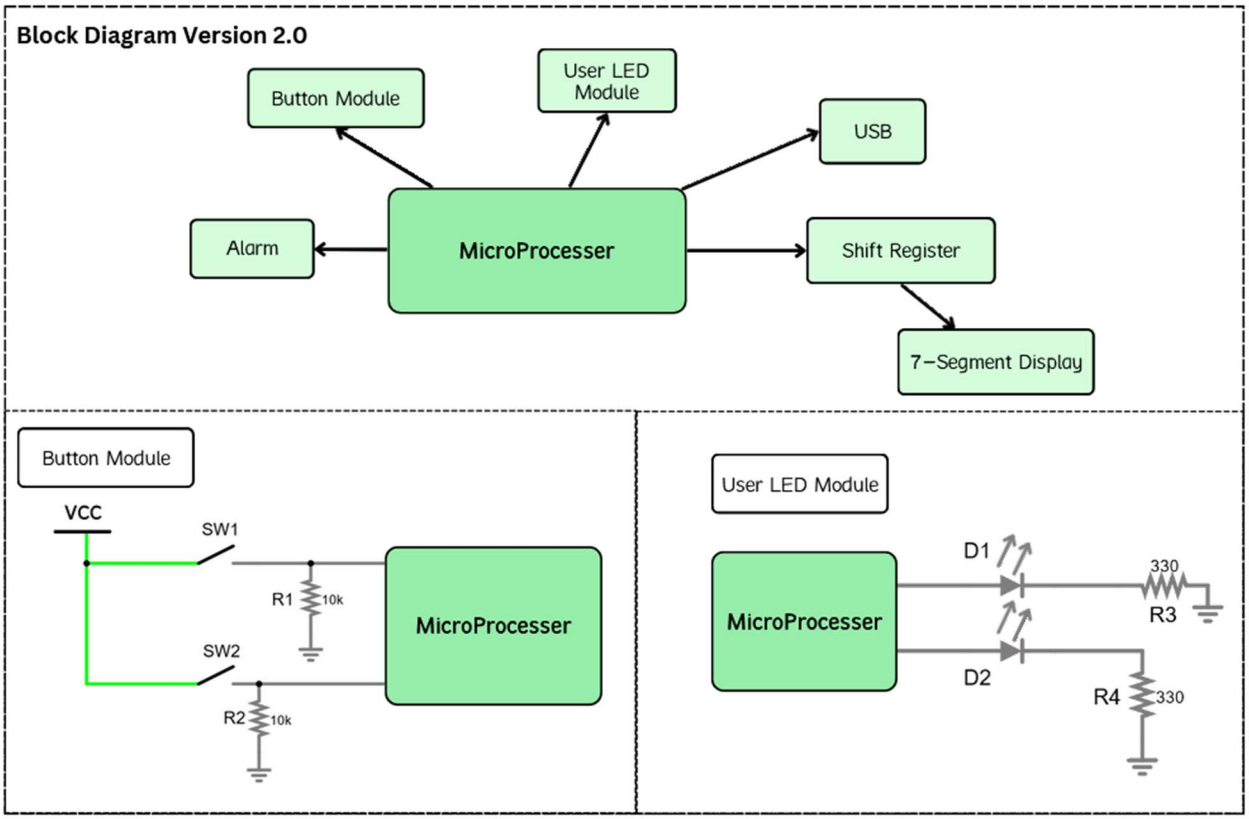

Figure 2: Block Diagram Version 2.0

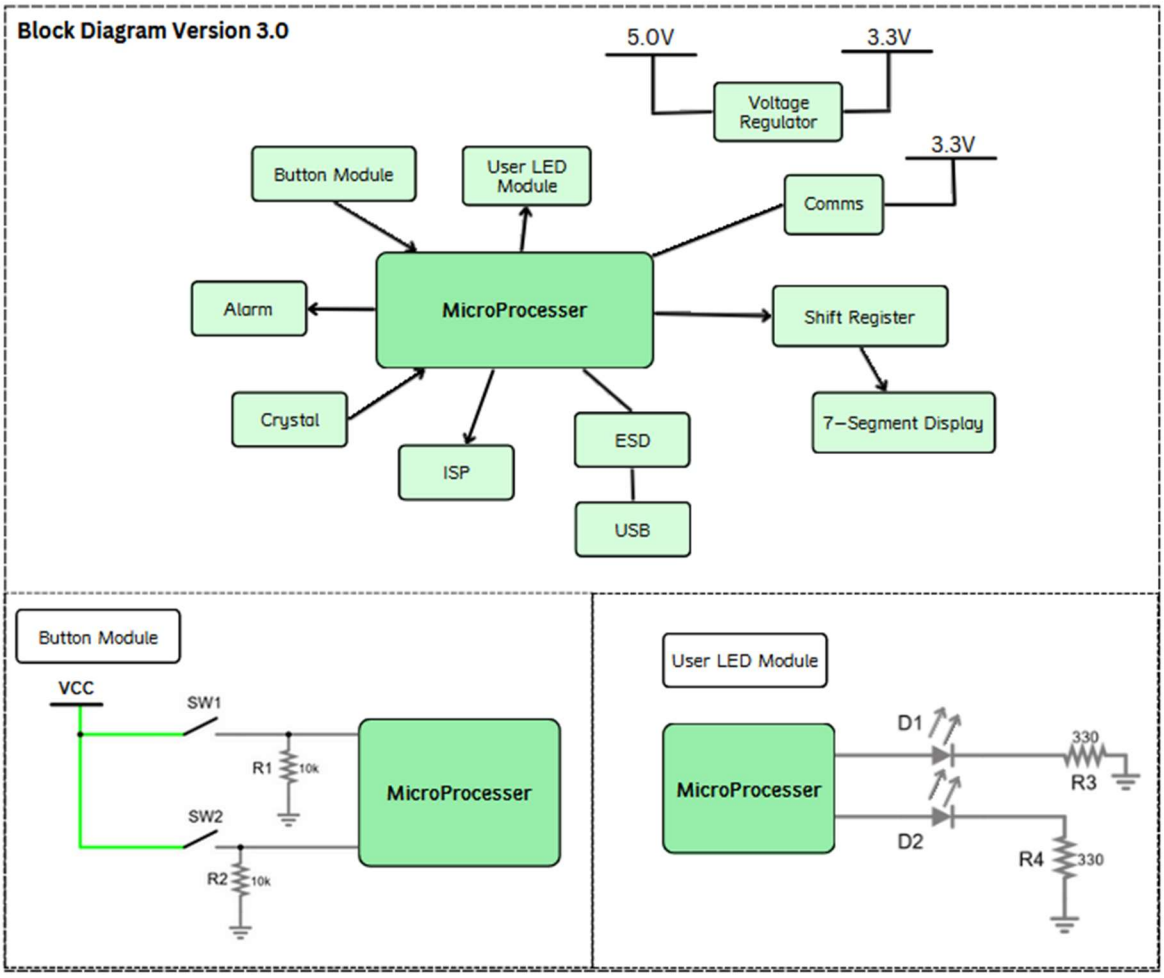

Figure 3: Block Diagram Version 3.0

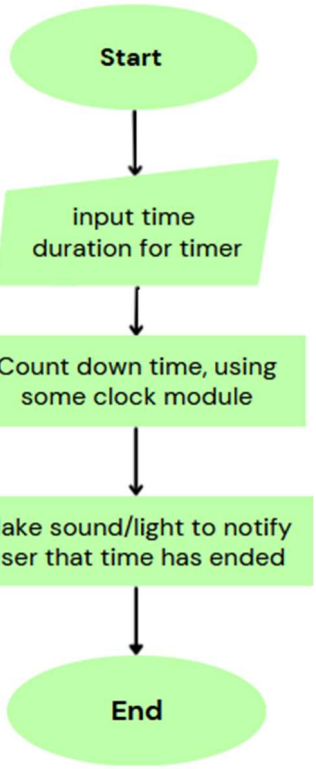

Figure 4: Signal Flow Diagram for System Ver 1.0

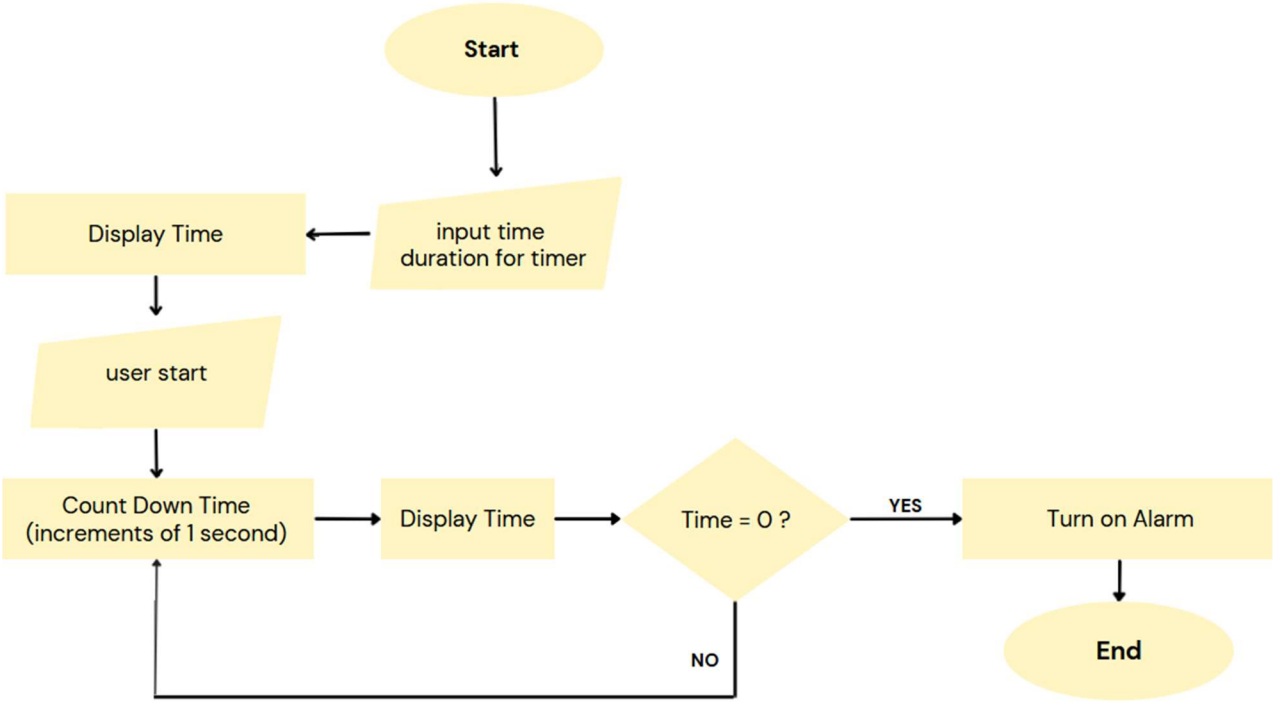

Figure 5: Signal Flow Diagram for System Ver 2.0

The figures above include Block Diagrams of the system design, and Signal Flow Diagrams for the same design, specifying the modules and some components to be used in the design and prototyping and the way in which the system will process signals for intended functionality as a user-programmable timer.

Components Selection

- Arduino Uno (selected for Phase A), Arduino Mega (preferred) Microcontroller
    - Arduino Uno selected for Phase A for accessibility, and already acquired prior to design stage. Used in first round of prototyping.
    - Price: $22.95 on Mouser Electronics.
- AT Mega 32 U4
    - Selected for Phase B thru Final Product development
    - 8-Bit Microcontroller, Arduino IDE compatible, selected as it is smaller than the Uno and more affordable at 24.44% of the Uno price.
    - Price: $5.61 per unit on Mouser.
- 7-Segment Display
    - The display on which the clock time will be displayed for user.
    - LTC-5723HR selected.
    - Price: $2.97 per unit on Mouser
- 8-Bit Shift Register
    - Used to interface between the Uno and the 7-Segment display, reducing the pins on the microcontroller used to program the 7-segment display.
    - Texas Instruments SN74HC595PWRG4 shift register selected for further prototyping.
    - Price: $0.62 per unit on Mouser.
- Buzzer
    - Used as alarm module to notify user when the timer has run out.
    - CMI-1295 Buzzer selected for use.
    - Price: $1.18 per unit on Mouser
- Buttons
    - Phase A Prototype:
        - Only two buttons used for user input, one for Start/Stop commands, and the other to positively increment the timer duration in seconds. Using negative logic.
    - Phase B-Onward Prototypes:
        - Three user buttons used. The two specified in Phase A, and a third button to act as Reset for the timer module. Using negative logic.
        - Start/Stop and Increment Buttons are PTS125 Tactile Switch buttons, $0.43 per unit price on Mouser.
        - Reset Button is PTS526 Tactile Switch buttons, $0.13 per unit price on Mouser.

    - Total Buttons Price: $0.99
- USB Connector
    - Used for communication with programming computers and to connect to external power supply.
    - UJ4-MBH-4-SMT-TR selected for prototyping.
    - Price: $0.88 per unit on Mouser.

Build Prototype

For Phase A of prototyping, a “shield” was made to work with the Arduino Uno. Called a shield as the design fit on top of the Arduino Uno controller board. As this first design used the Uno, internal timers, USB communications, and power supplies were used. As such the shield PCB only needed to include the two user buttons (Start/Stop and Increment), the shift-register, the Buzzer, the two user-LEDs, the 7-Segment display, and all the resistors required for optimal operation of these components. The schematic and a scan of the routed shield design are shown below (Figures 6 and 7).

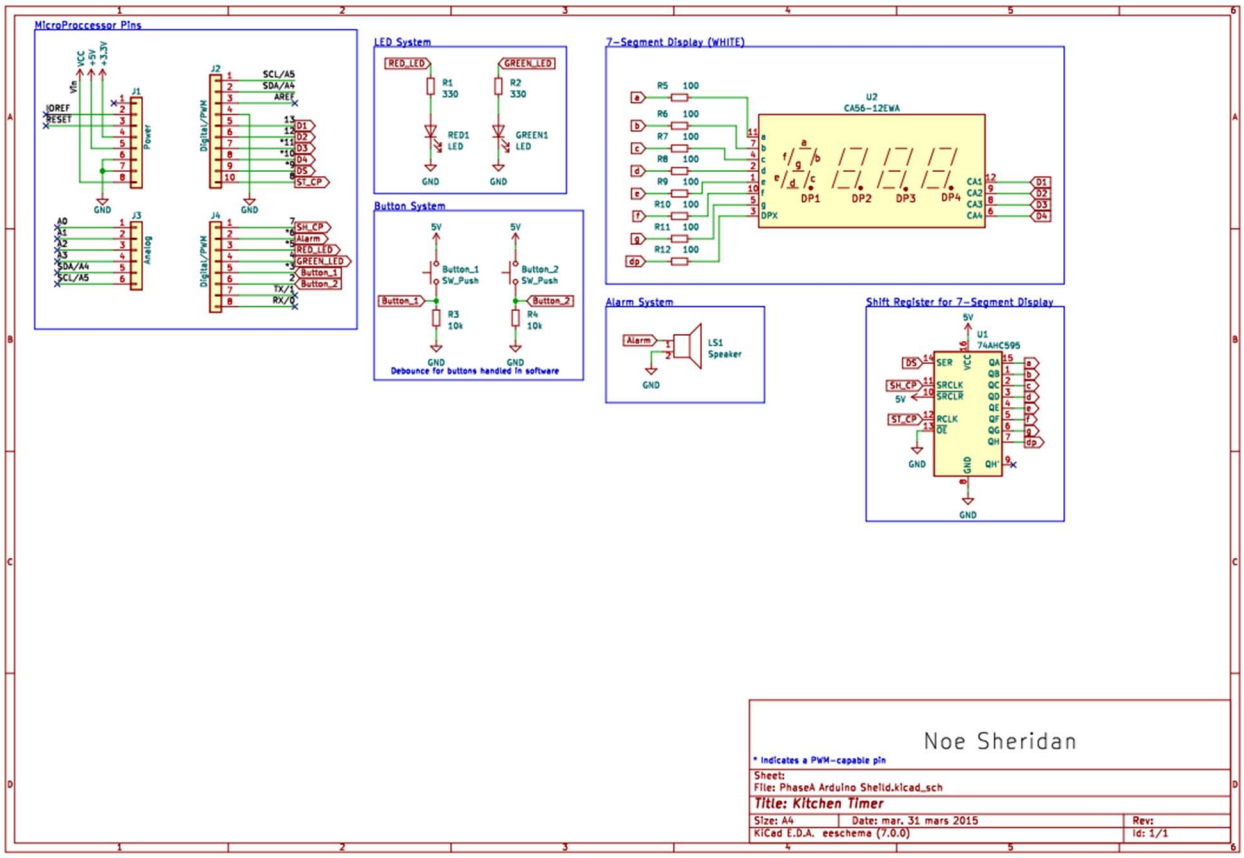

Figure 6: Arduino Uno Shield Prototype Schematic

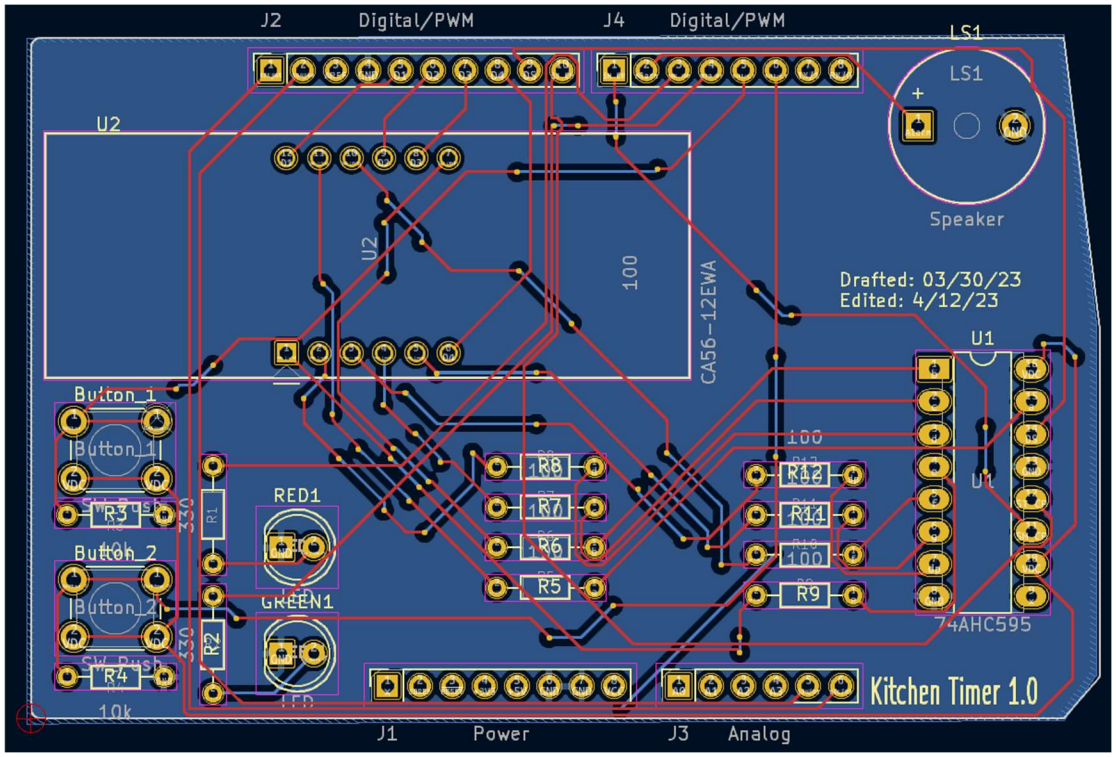

Figure 7: Arduino Uno Shield Routed PCB (Kitchen Timer Ver 1.0)

PCB Design

The figures below detail the current PCB design, which will utilize the AT-Mega microcontroller instead of the Uno. With this change in controller the inclusion of additional modules on the PCB are required for functionality. Including a voltage regulator, a crystal, many capacitors, a reset button, an ISP programmer, and a USB connector. Mounting holes were also added to mount the PCB within the enclosure. The PCB has a ground plane on the back side to reduce the routing necessary. The schematic, routed PCB front and backside, the 3-D view of front and back sides are shown below in Figures 8-12.

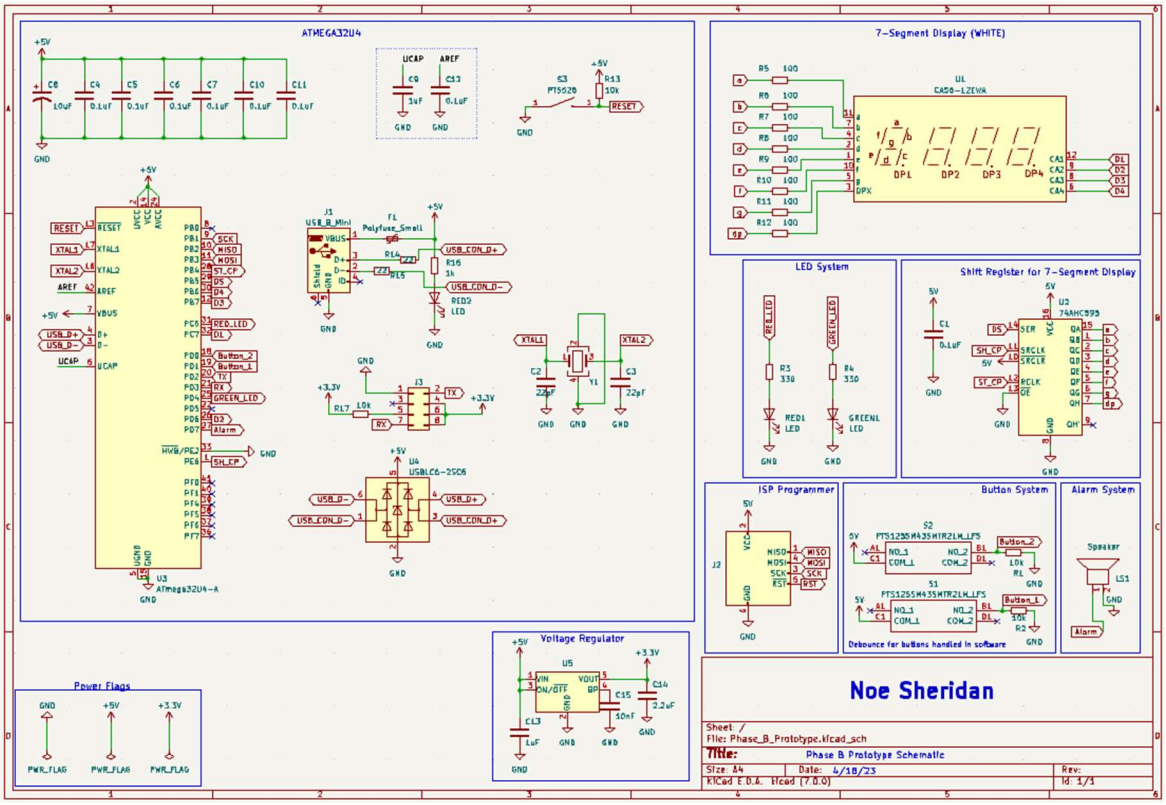

Figure 8: Schematic for the Phase B Prototype

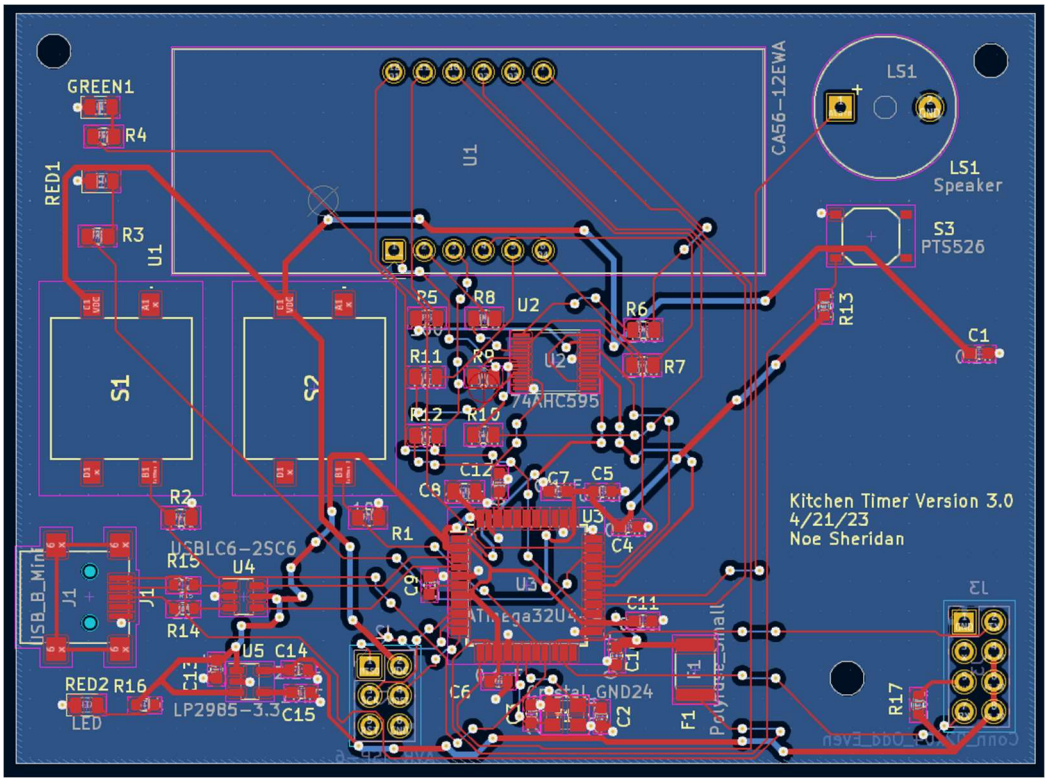

Figure 9: Routed Kitchen Timer PCB

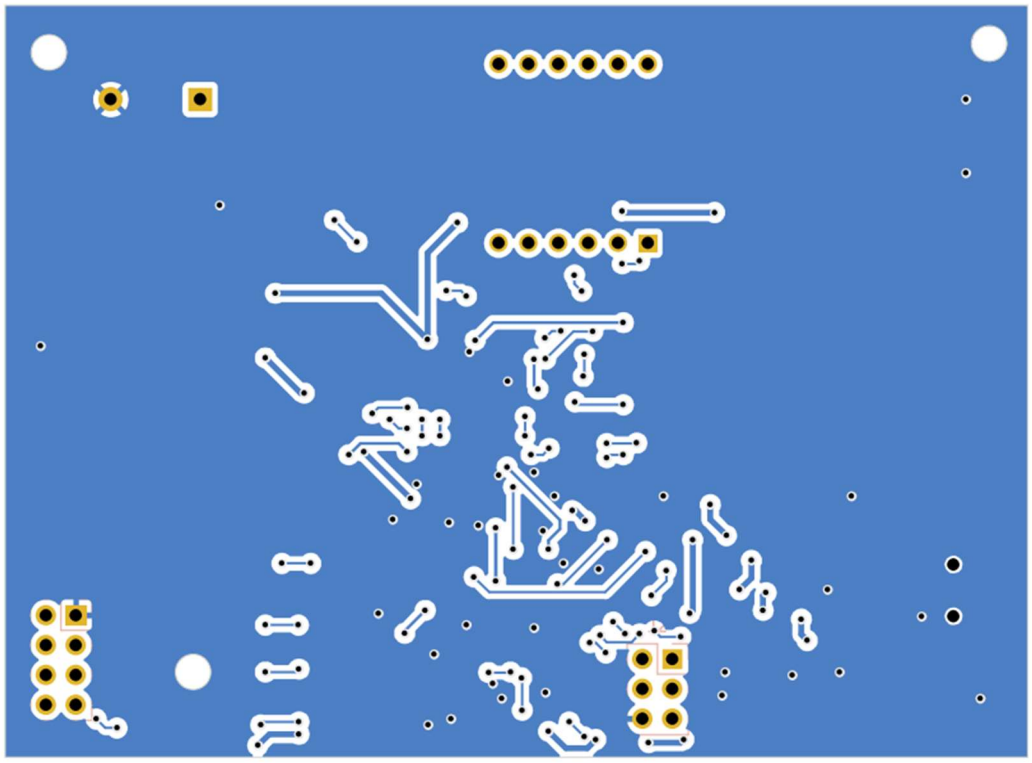

Figure 10: Routed Kitchen Timer PCB Backside (Ground Plane)

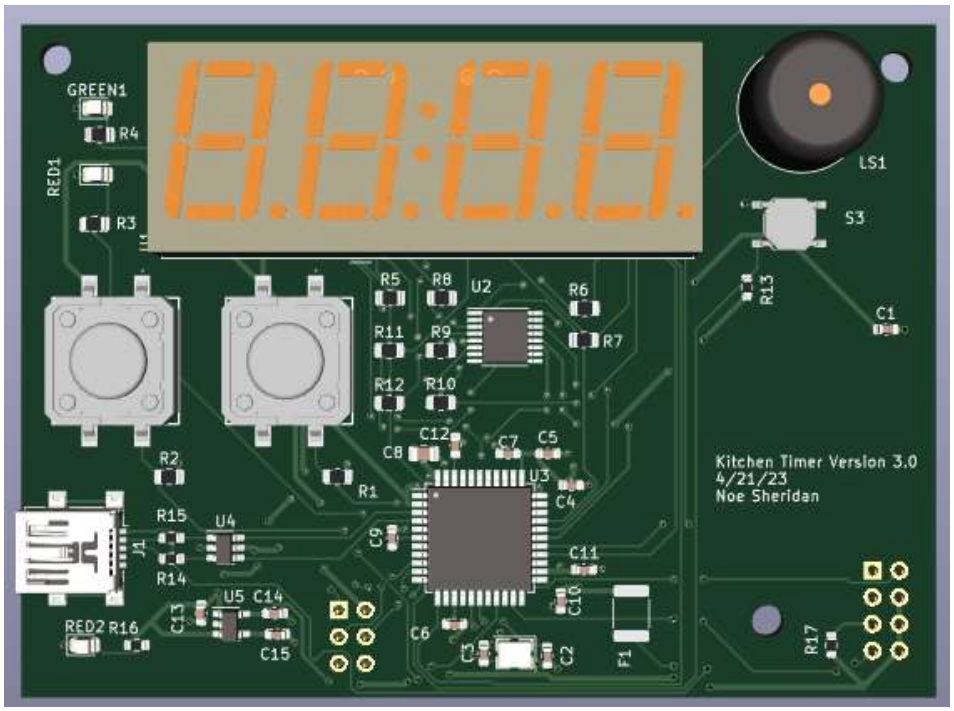

Figure 11: 3-D View of Routed PCB Front Side

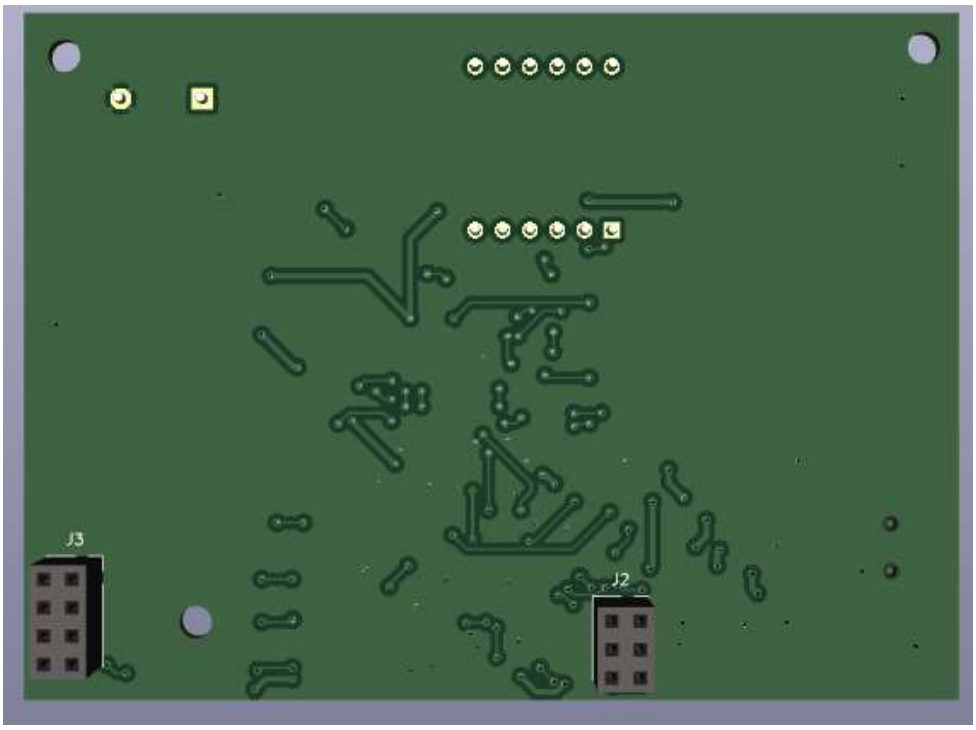

Figure 12: 3-D View of Routed PCB Backside

Assembly Stage

Not applicable as of yet.

Software Design

The code was written using C in the Arduino IDE. The current code titled Phase_B_integrated_attempt_wifi_and_uno.ino it attempts to integrate the WiFi module with the Arduino Uno to update the webpage display. Currently, it turns on an onboard LED (pin 13) when the system has received the start command “STR” and turns the same LED off when it receives the stop command “STP”. It also updates the clock timer when it receives the “GET” command. This portion of the loop body code is shown below in Figure 13.

```
if(processdataflag ==1){//proccess serial commands

  processdataflag = 0; //reset the flag

  if(compareArrays(gCommsMsgBuff, "STR", 3) ==1 ){

    // if start command recognized turn on LED
    digitalWrite(LED, HIGH);
  }
  if(compareArrays(gCommsMsgBuff, "STP", 3) == 1){
    //if stop command recognized turn off LED
    digitalWrite(LED, LOW);
  }
  if(compareArrays(gCommsMsgBuff, "GET", 3) == 1){
    //if GET command recognized send clock data
    Serial.print("$00:01\n");
  }


}
```

Figure 13: Command Processing Loop of Main Body Arduino Code

Enclosure Design

The enclosure designed below is 94mm wide and 65mm tall, (9.4 x 6.5 cm). There are extruded cuts in the front to allow the user access to the buttons, LED, 7-segment display, and to allow room for the buzzer. On the user's left side there is an extruded cut to allow access to the USB connector. And on the back plate there are extruded cuts to give user access to the fuse connector and the connector for the Wi-Fi module, as well as circular extruded cuts which line up with the mounting holes on the PCB for mounting the PCB within the enclosure.

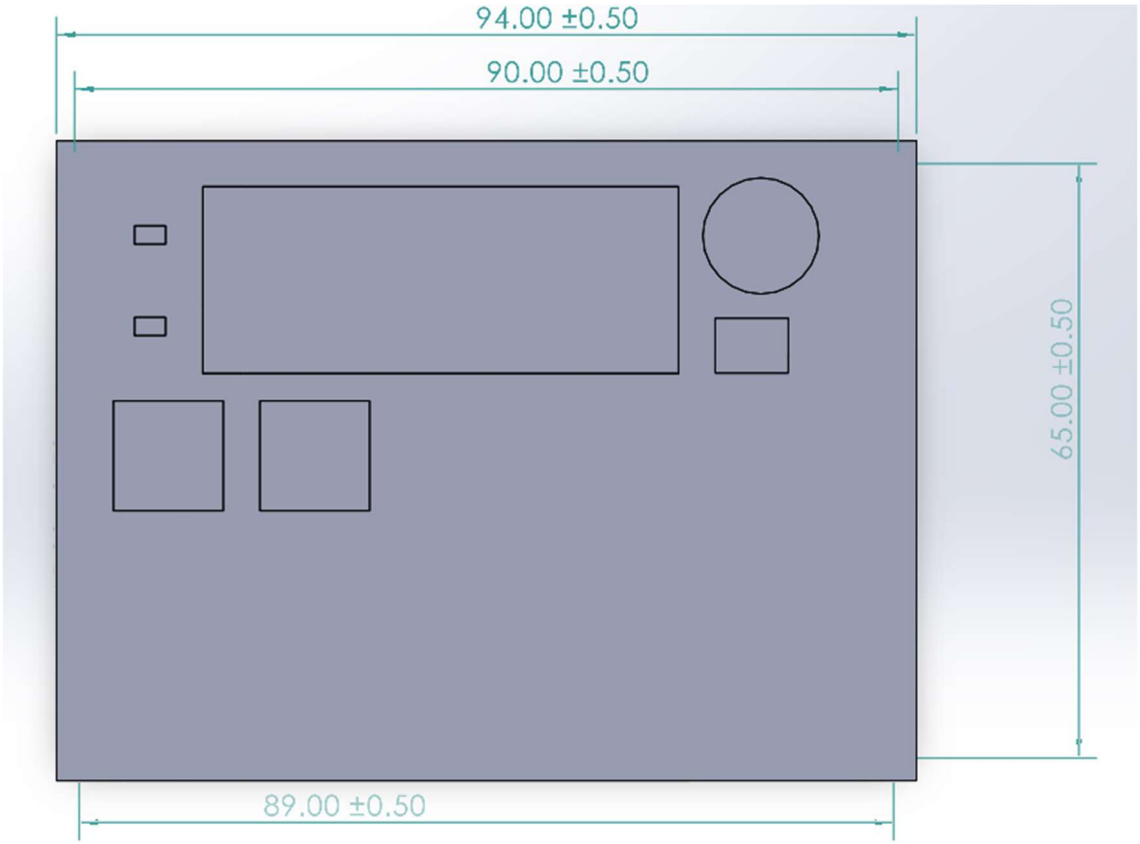

Figure 14: Front View of PCB Enclosure

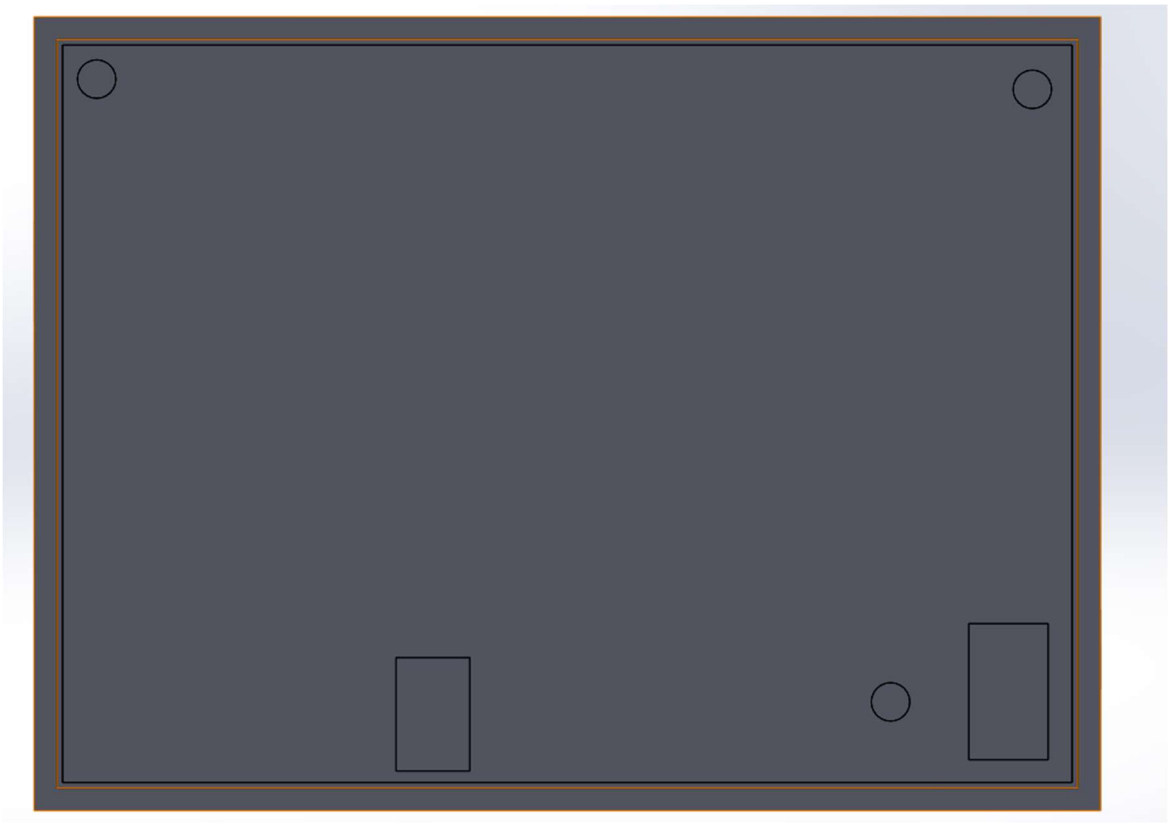

Figure 15: Back view of PCB Enclosure

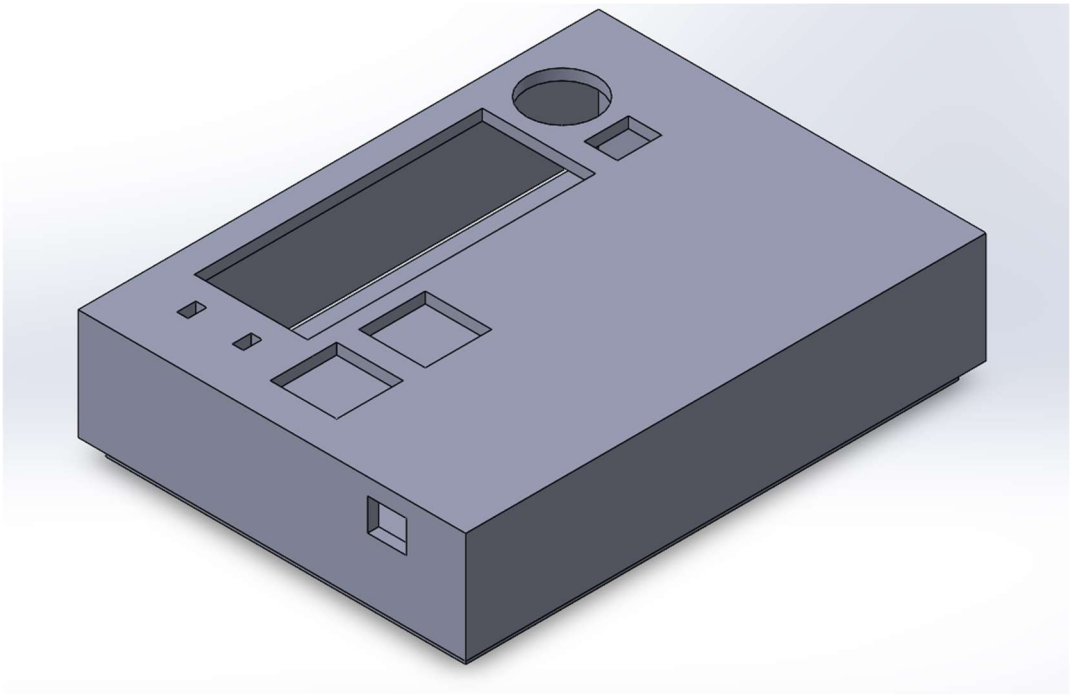

Figure 16: Isometric View of PCB Enclosure