Kitchen Timer

Kevin Fuentes

Project Requirements

The project is to design and create a kitchen timer that implements a button to start and stop the time, a button to increment the time, two status LED's, a buzzer to notify the user when the time has expired, a seven segment display to show the time remaining, and a way to communicate with Wi-Fi.

System Design

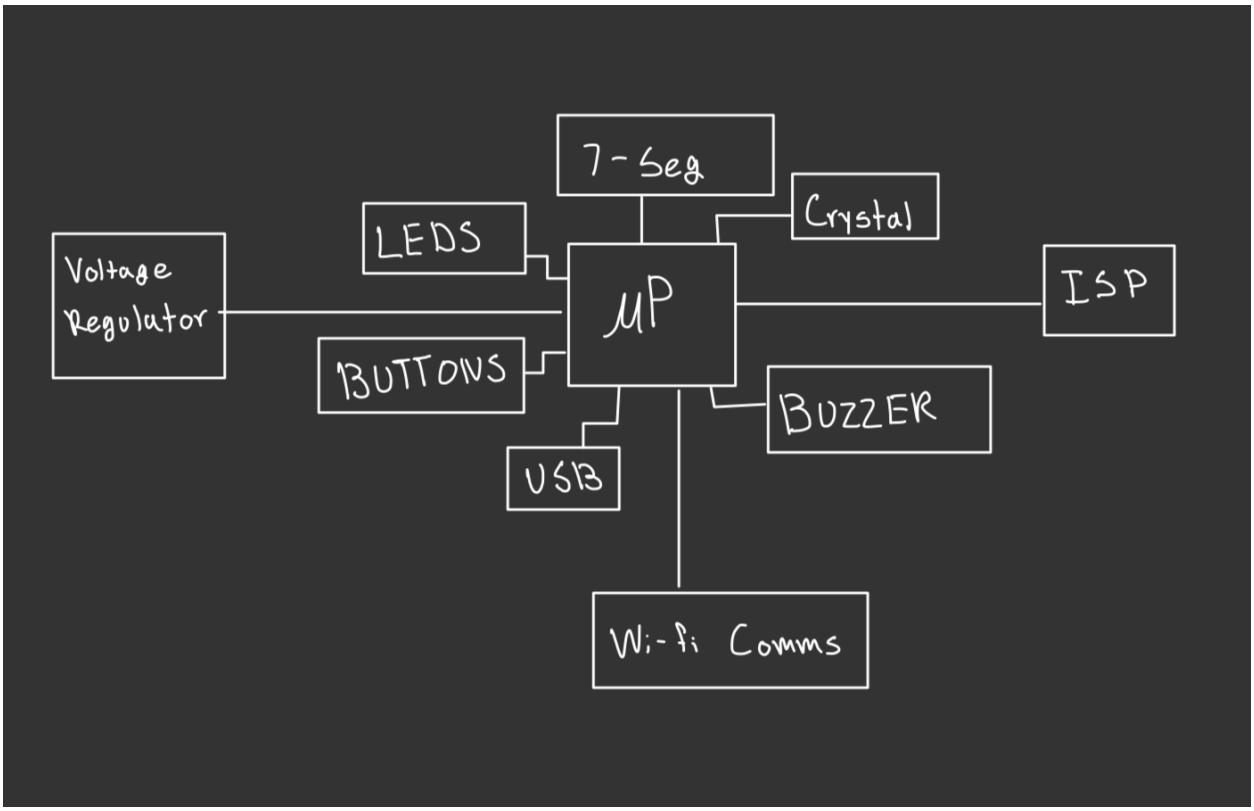

Figure 1: Hardware Block Diagram

The PCB board will include the microprocessor along with the crystal, seven segment display, button, LED's, buzzer, a USB connector to power and code the microprocessor, the ISP module, and a voltage regulator. All component will be routed on the PCB board.

Components Selection

Below is the list of components used.

Microprocessor - ATMEGA32U4-AU

Crystal - ABM8-16.000MHZ-D1X-T ABRACON

Voltage Regulator - USBLC6-2SC6 STMicroelectronics

Buttons - TS526 SK08 SMTR2 LFS and PTS125SM73SMTR21M LFS C&K Switches

LED's - 50080BS75000 Wurth Elektronik

USB - UJ2-MBH-4-SMT-TR CUI Devices

Shift Register - SN74HC595PWRG4 Texas Instruments

Buzzer - CMI-1295-0585T CUI Devices

Seven Segment Display - TC-5723HR Lite-On

ISP - ESP8266

Build Prototype

The components were tested on a breadboard to ensure everything worked correctly. The ESP board was also tested by running a simple program to blink LED's when a command was received.

PCB Design

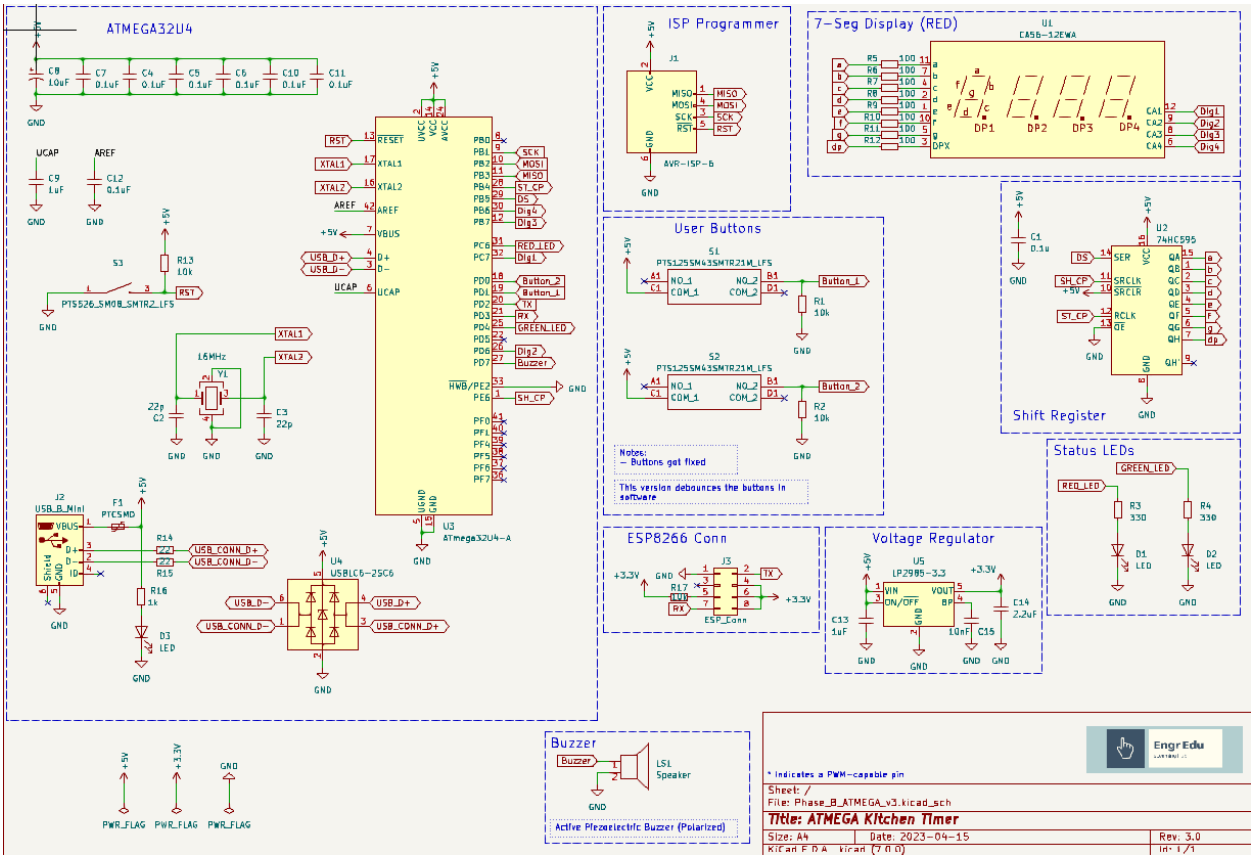

Figure 2: Hardware Schematic

The schematic (Figure 2) shows how the components are wired. Multiple resistors and capacitors were used in order to obtain proper functionality of the devices. The microprocessor is at the center and will be used to read inputs form the buttons, and ESP. As well as output data to the shift register for the seven segment display, and a signal for the buzzer.

Figure 3: PCB Component Layout

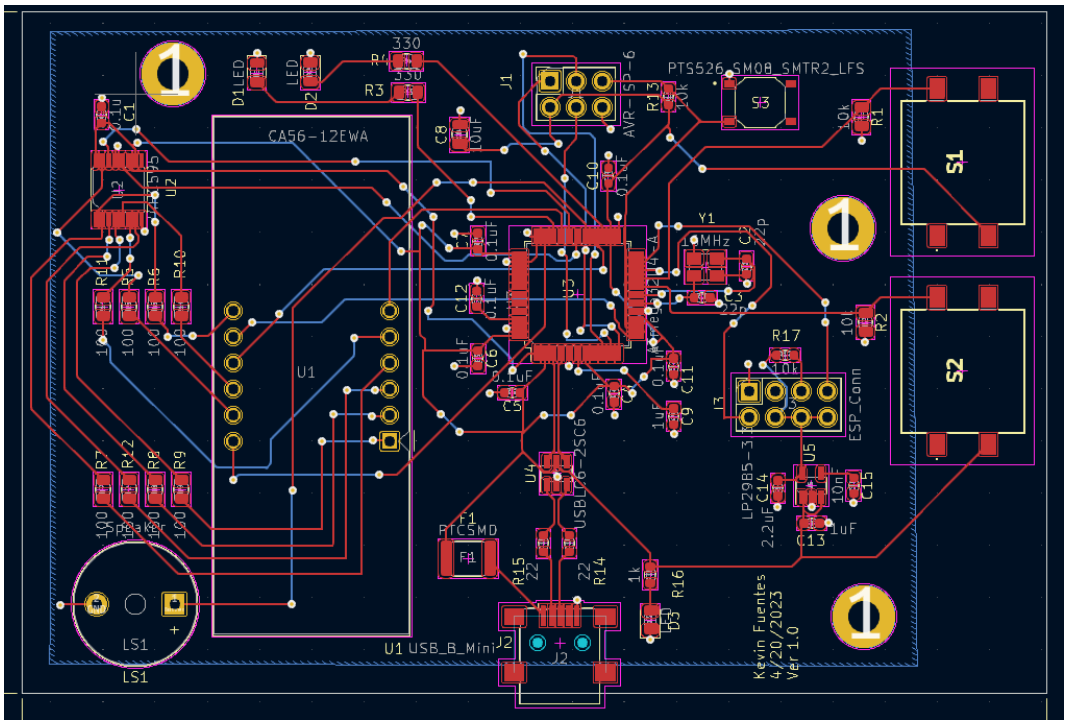

Figure 4: PCB Routing Layout

In Figure 3 the layout of the components on the PCB boar can be seen. In Figure 4 the routing of the PCB board is shown.

Assemble Stage

After soldering all the components the board was connected to a laptop via USB to test if it powers on. Next the boot-loader was installed in order to be able to upload the code using Arduino. After uploading the code the device is able to start/stop the timer and increment the time using the button. The time is displayed on the seven segment, and the buzzer goes off when the time has expired.

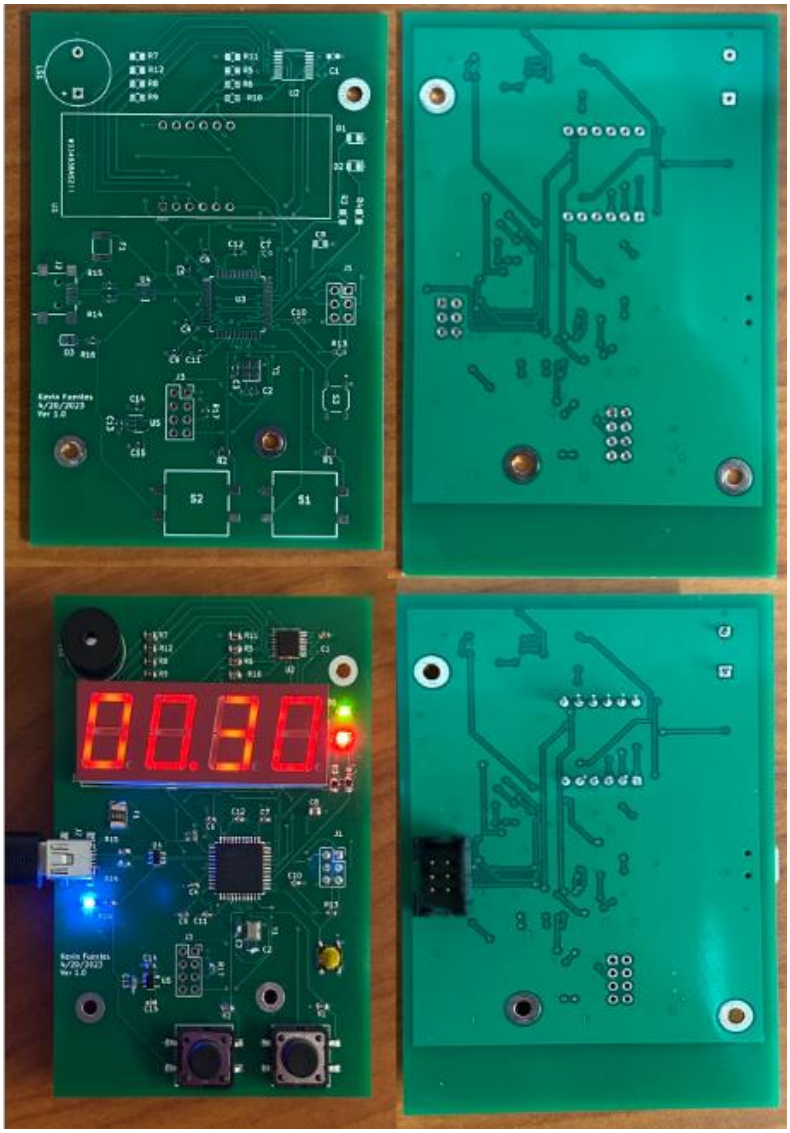

Figure 5: PCB Assembly

Software Development

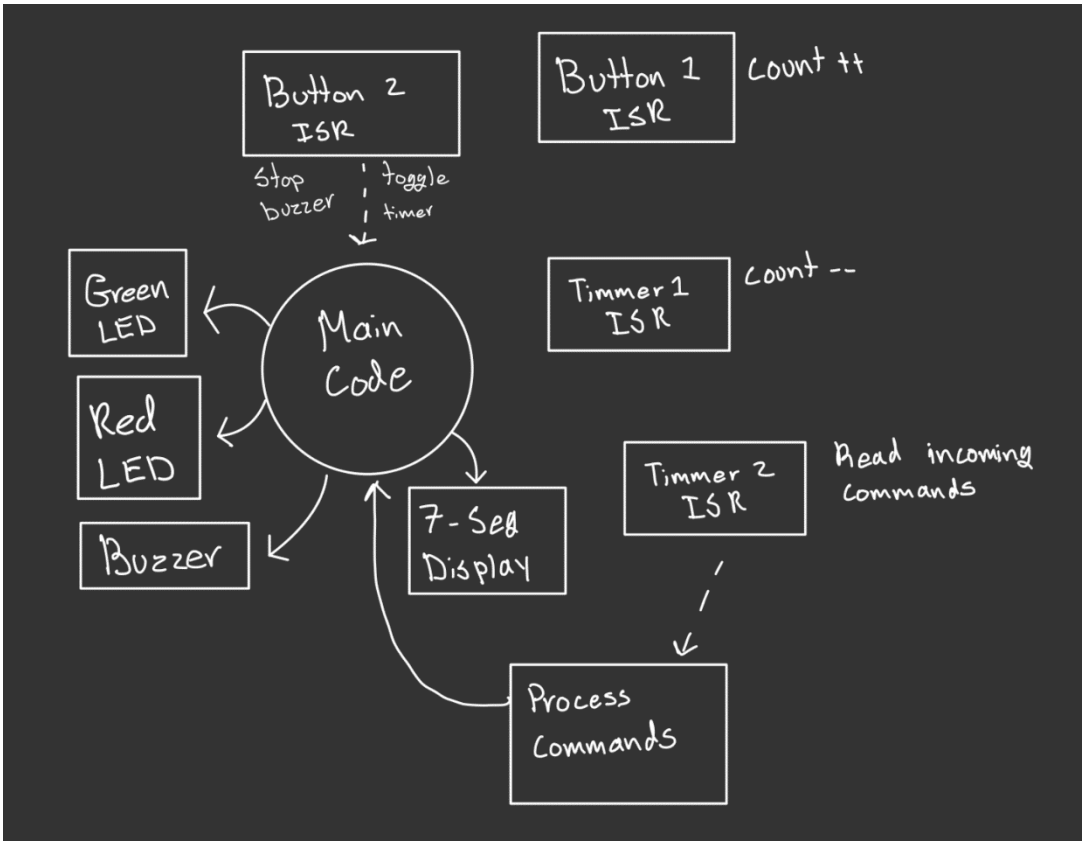

Figure 6: Code Flow

The software uses a combination of timers and interrupt routines to detect inputs and display the time on the seven segment display. The first timer is used to decrement the time amount every second. This only occurs when the timer is enabled. The timer can be toggled by the button two interrupt. This interrupt happens when the button is pressed by the user. The timer can also be toggles when the start/stop command is received from the ESP module. The second timer is used to read incoming characters from the ESP module. It read a character every .4ms. If the character corresponding to the beginning of a command is detected the previous command is cleared and the next characters are saved to be compared to a command. Once the end command character is received no more characters after that are saved. The commands received can trigger different events such as starting or stopping the timer.