Grayson Lewis
ECE 342

Developer Guide

1. Abstract

This developer guide describes an egg timer which is intended to be used to time the cooking of hard or soft-boiled eggs. It has an adjustable time between four and ten minutes, and will sound an alarm when the set time has run out. The system uses an ATmega328P microcontroller to receive and process user input, drive a seven-segment display to show the remaining time, and generate a square wave to sound the alarm. The system is powered via a USB connector, it is rated IP43, and is accurate to ±1 second per minute.

2. Electrical Specifications

Supply Voltage: The system is designed to be supplied directly through a USB port. Thus the operating voltage range is the same as the USB voltage output range.

Max supply voltage: 5.25V
Min supply voltage: 4.75V

Supply Current: The seven segment display draws the most current in the system. It drives up to 7 LEDs at a time with 20mA through each LED. Maximum LED duty cycle is approximately 100%, and nominal duty cycle is approximately 50%. This gives the maximum and nominal current draw for the system.

Max Supply Current: 170mA*
* 140mA for the maximum current to the seven segment display, plus 30mA of safety margin.

Nominal Supply Current: 100mA*
* 70mA for the nominal current to the seven segment display, plus 30mA of safety margin.

Max Operating Temperature: The microcontroller in the system has a maximum operating temperature of 125°C. This is derated to 75°C for system longevity.

Min Operating Temperature: The microcontroller in the system has a minimum operating temperature of -55°C. This is derated to 0°C for system longevity.

Nominal Operating Temperature: 25°C

Table 1: Electrical Characteristics Summary

Characteristic	Min	Nominal	Max	Unit
Supply Voltage	4.75	5	5.25	V
Supply Current	-	100	170	mA
Operating Temperature	0	25	75	C

3. User Guide

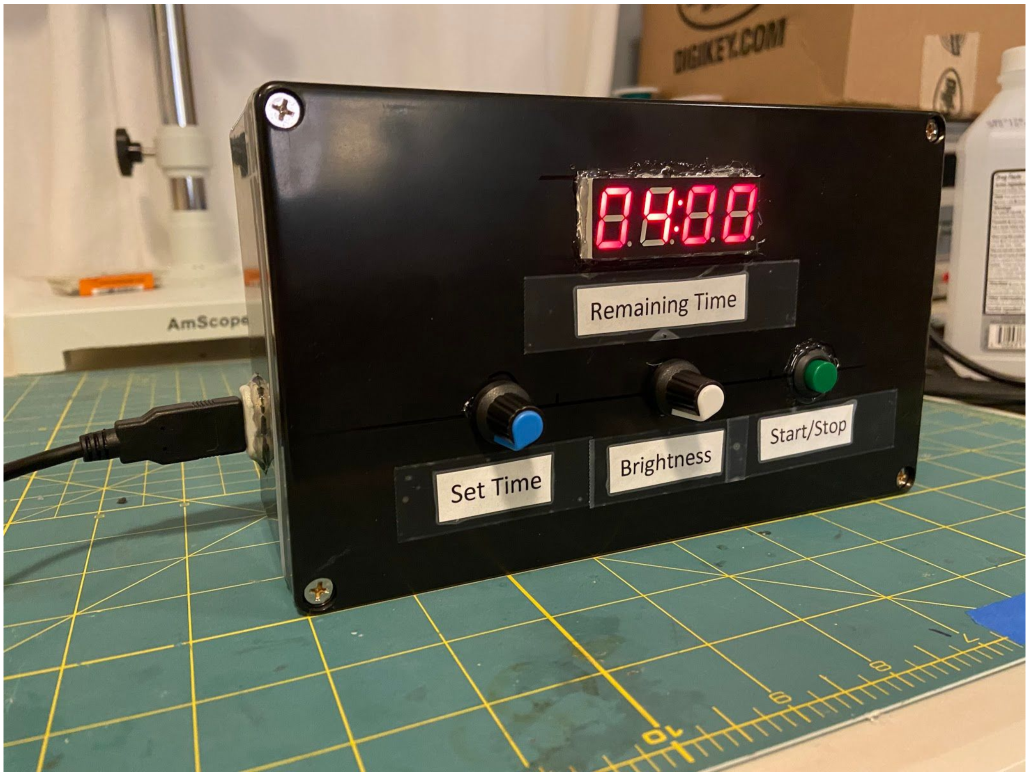


Figure 1: Completed System

1. Setting Up the system

The first step in being able to build this system is being able to communicate with the microcontroller. The microcontroller was soldered to a protoboard, and connectors were soldered to all of the pins, which allowed all of the hardware to be connected to the microcontroller. Once the microcontroller was soldered, an Atmel ICE programmer was used to upload the firmware to the microcontroller.

Once the firmware was uploaded, the seven segment display was built on the custom PCB, and then tested to verify that the firmware was correctly driving the display.

After the seven segment display was tested, the rest of the hardware was connected to the microcontroller. This included the potentiometers to control brightness and the timer duration, the button to start and stop the timer, the audio amplifier module, and the speaker.

Once all of the hardware was connected to the microcontroller, the system was tested electrically to verify that all blocks were working correctly, the system responds correctly to inputs, and rings the alarm when the timer has run out.

After the electrical testing has been completed, the system was assembled into a prototype enclosure similar to the one one designed for the system. This enclosure was sealed using hot glue in order to meet the requirements for ingress protection.

Finally, once the system was fully prototyped, all of the system requirements were tested to ensure that the system met the project requirements.

2. Using the system

Once the system is set up, it is ready to be used.

The timer requires power via a USB cable which can be plugged into the connector in the side of the enclosure. The system will power on and display the current time setting on the seven segment display.

The user can then set the time by rotating the knob labeled “Set Time” The time is adjustable between four and ten minutes in increments of 15 seconds.

The user can adjust the brightness between three settings by rotating the knob labeled “Brightness”
When the user has selected the desired time and brightness, they can press the button labeled “Start/Stop” which will begin the timer countdown.

- At any point while the timer is running the user can press “Start/Stop” to cancel the timer.
- After the timer runs out, the alarm will ring for 30 seconds unless “Start/Stop” is pressed while the alarm is ringing
- If the alarm stops after 30 seconds, the timer will display zero minutes and zero seconds until the “Start/Stop” is pressed.

4. Design Artifact Figures

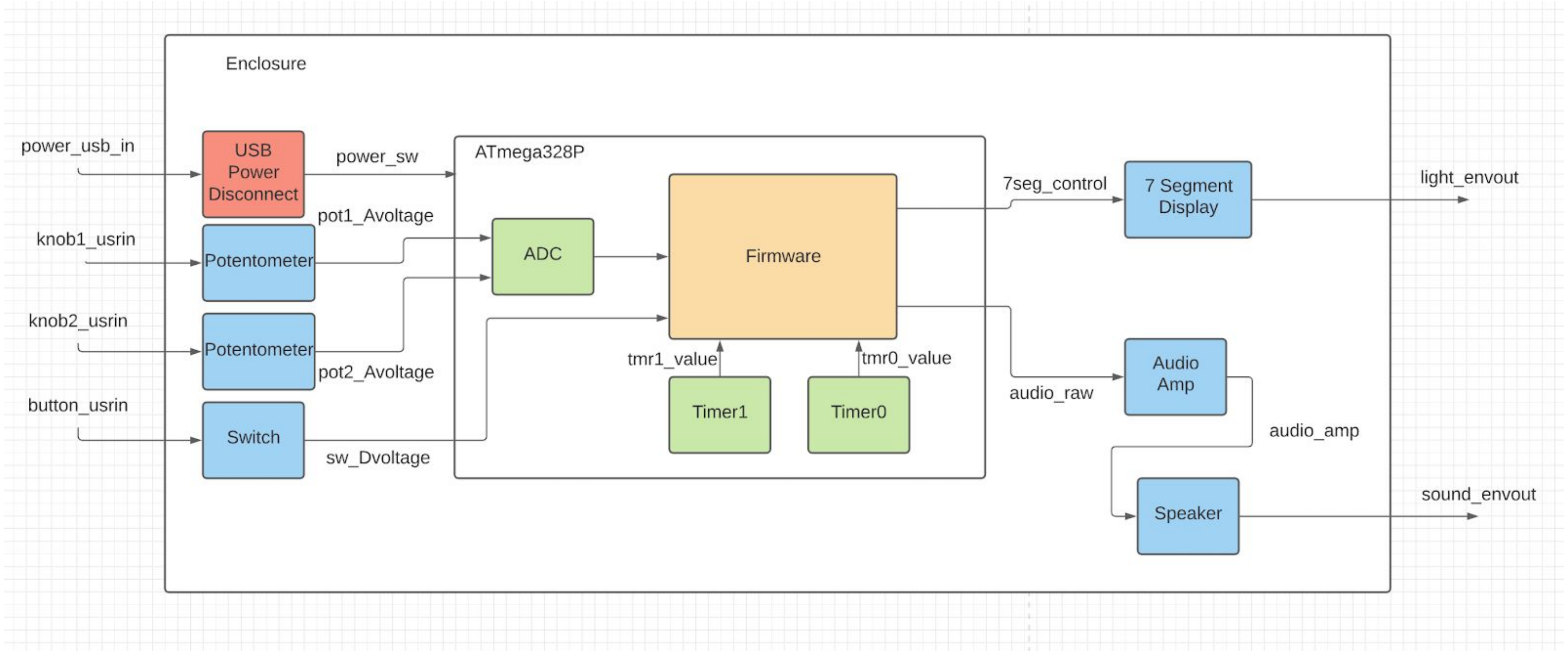


Figure 2: System Block Diagram

The system block diagram shows how the hardware is connected, and how that hardware interfaces with the system firmware to produce the required system outputs. Starting on the input side, the two potentiometers, which are used to set the time and brightness are sampled through the hardware analog to digital converter, which allows a wide range of inputs from the user to be utilized in the firmware. The hardware timers in the microcontroller allow the system to precisely time the control signal to the seven-segment display (which allows for PWM control of the display brightness), as well as the precise timing that allows the timer to meet its accuracy requirement. Once the timer has run out, the firmware generates a square wave, which is then amplified, and passed to the speaker to create the alarm sound that will alert the user when the timer has run out.

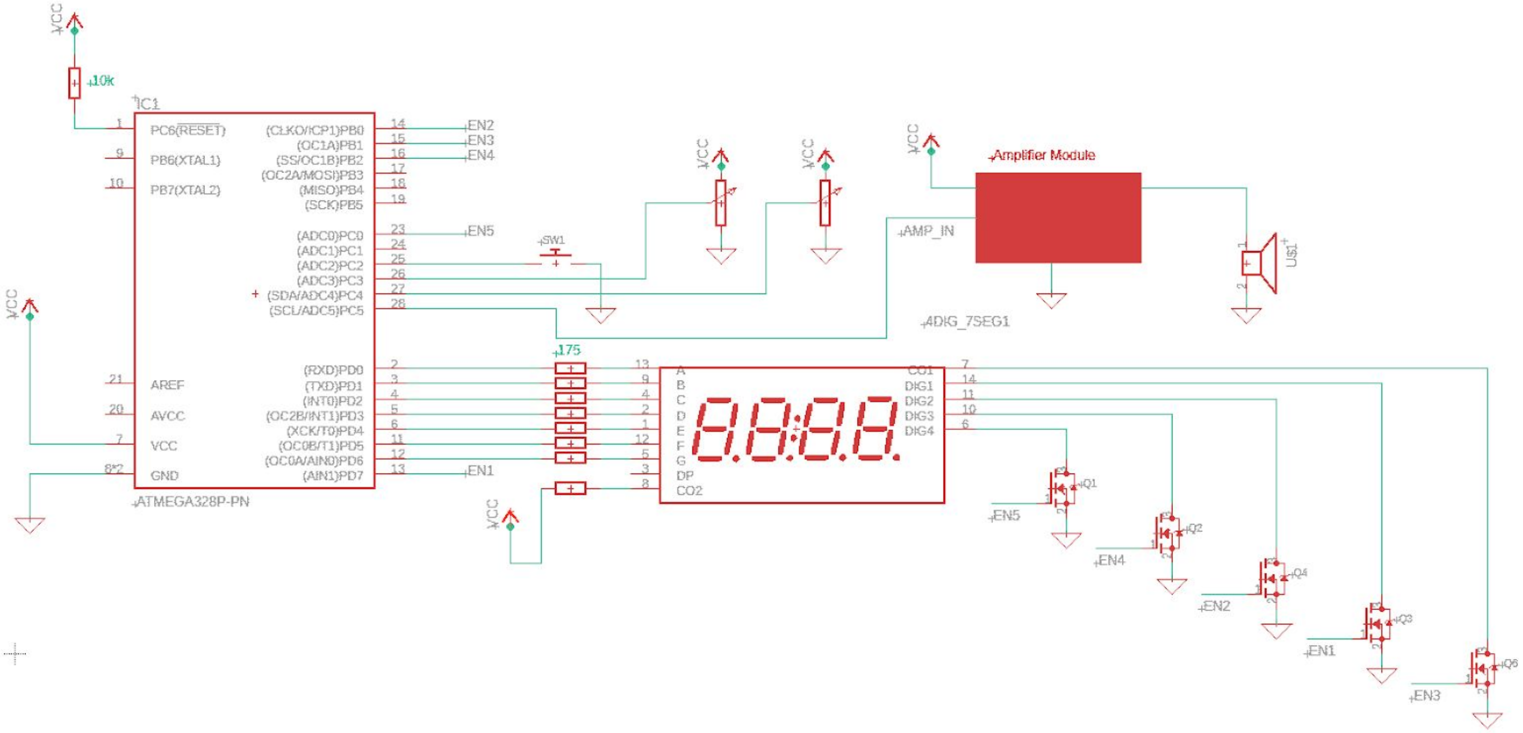


Figure 3: System Electrical Schematic

The electrical schematic shows how the microcontroller electrically connects to all of the additional hardware that is necessary for the system to function. This is useful for anyone who may wish to replicate the system, or modify it. Of particular note is the method of controlling the display, instead of relying on a separate microchip to drive the display, it is driven directly by a single microcontroller. This requires few, inexpensive components, thus reducing the cost of the system were it to be mass produced. This also gives the designer greater control over the display, as they have complete freedom to display whatever they want, and aren’t just limited to numbers.

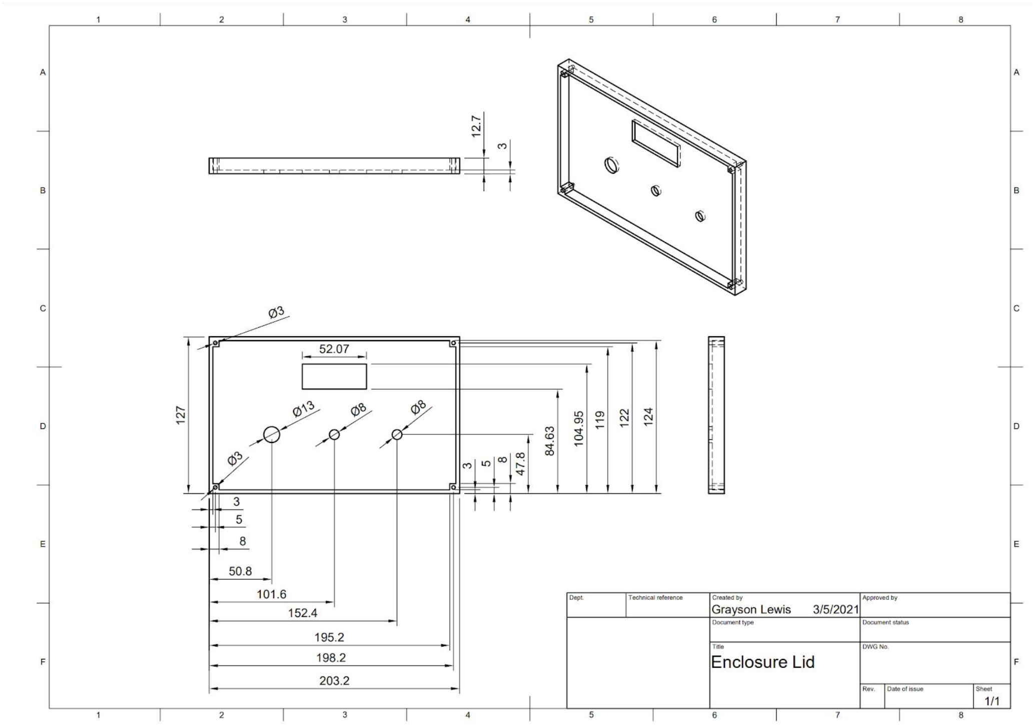


Figure 4: System Enclosure Lid

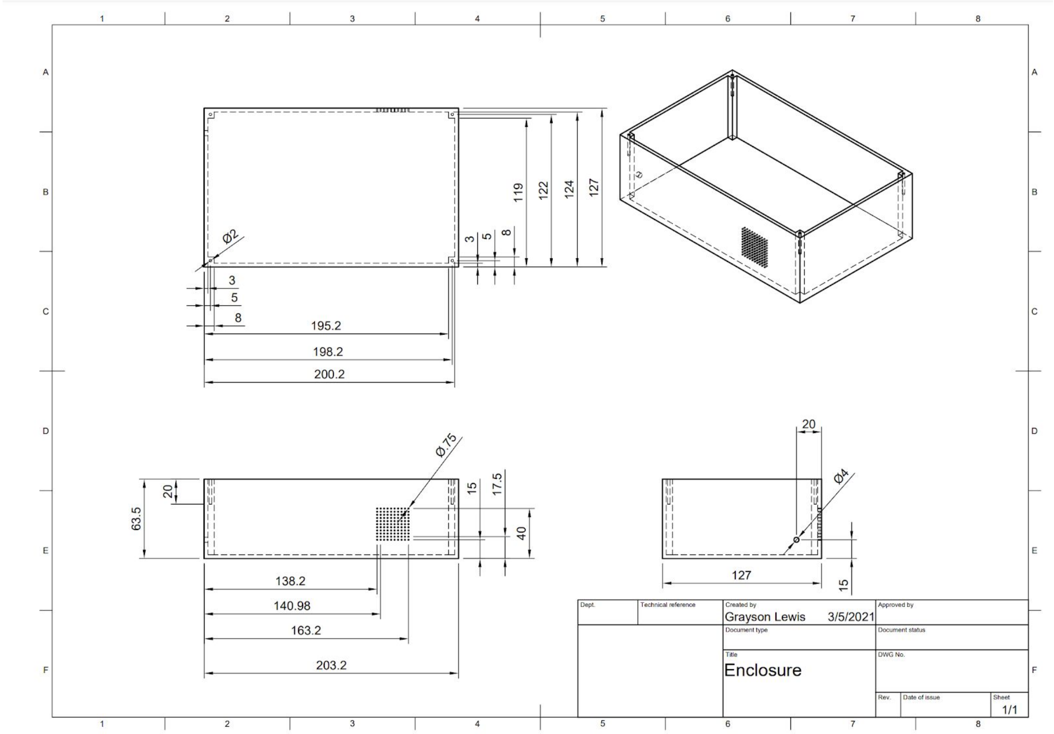


Figure 5: System Enclosure Main Compartment

The system enclosure serves several key functions to the system. It provides mounting for all the user interface components, such as the display, and the potentiometers. It also allows the system to meet its ingress protection requirements for both objects and water (IP43) It has a speaker grill on the bottom for the speaker to be heard through, which means the user is more likely to hear the alarm when it goes off. Finally, it protects the electronics from shock from dropping or shaking, allowing the system to continue working through these events.

5. PCB Information

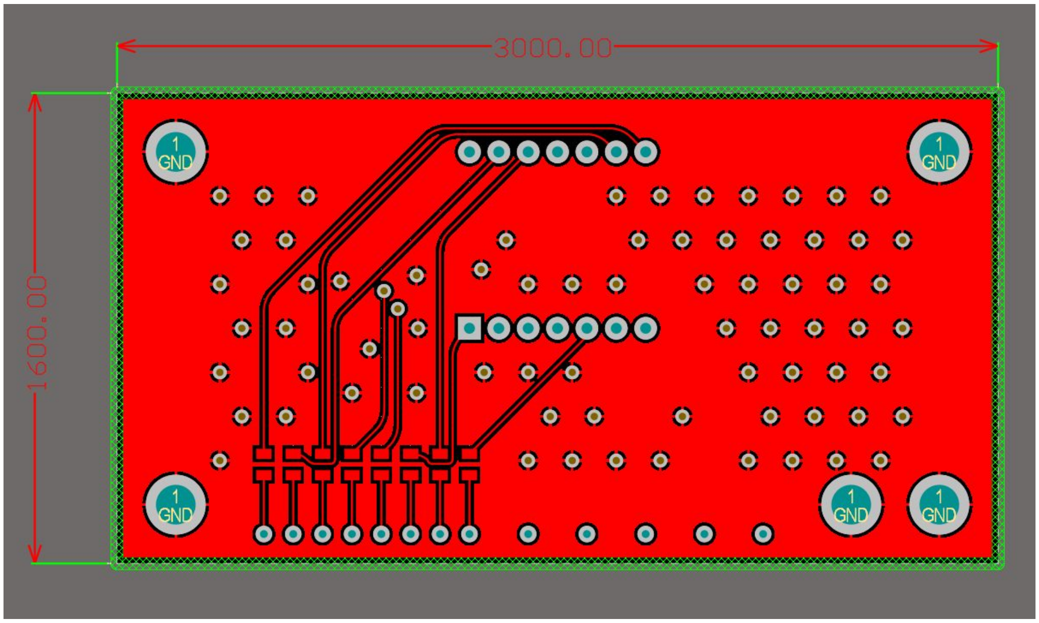


Figure 6: PCB top layer

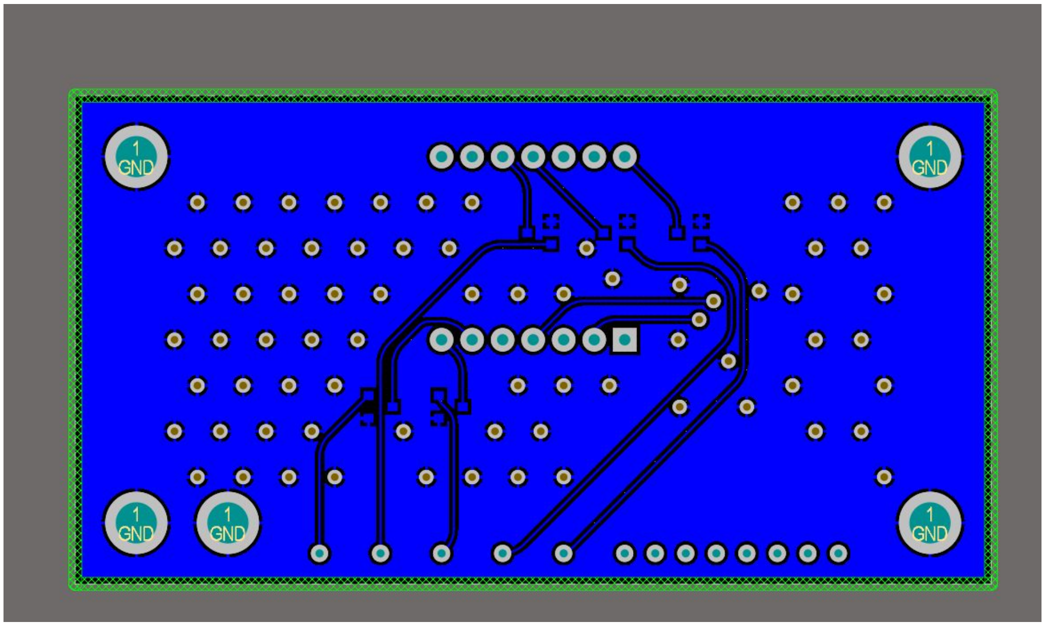


Figure 7: PCB bottom layer

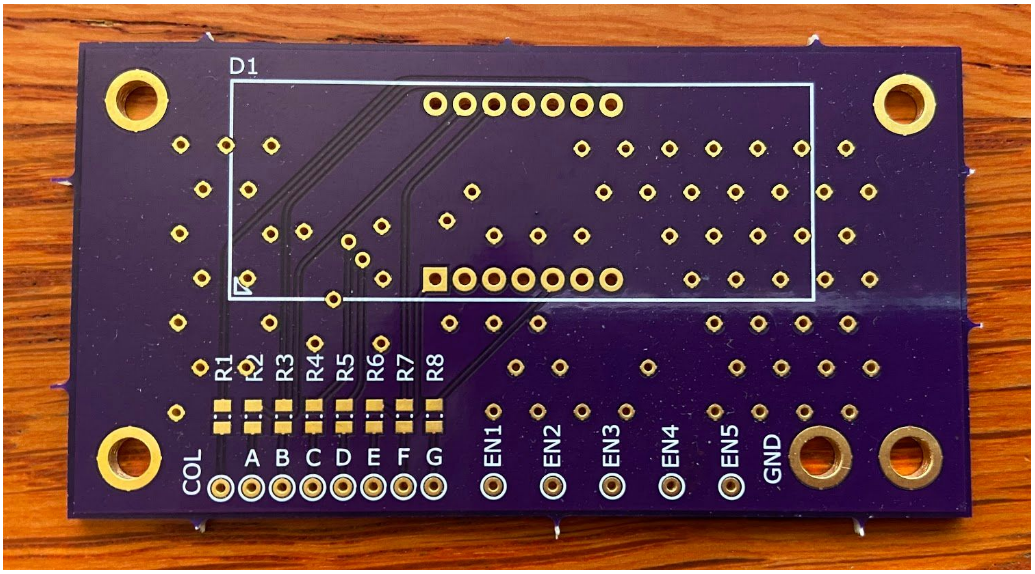


Figure 8: Fabricated PCB top layer

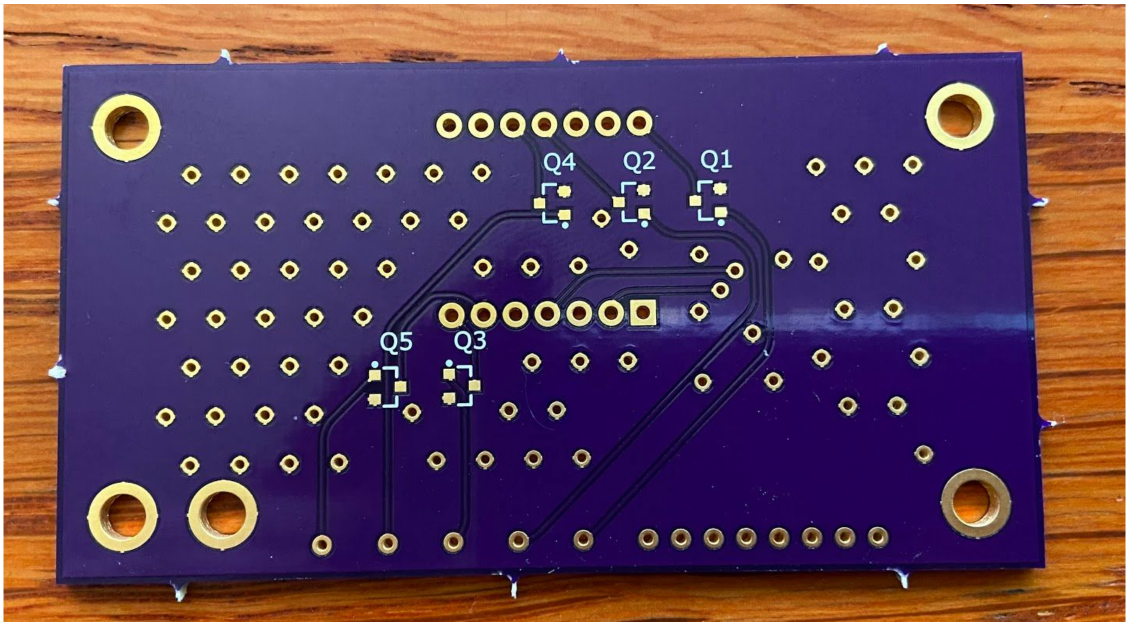
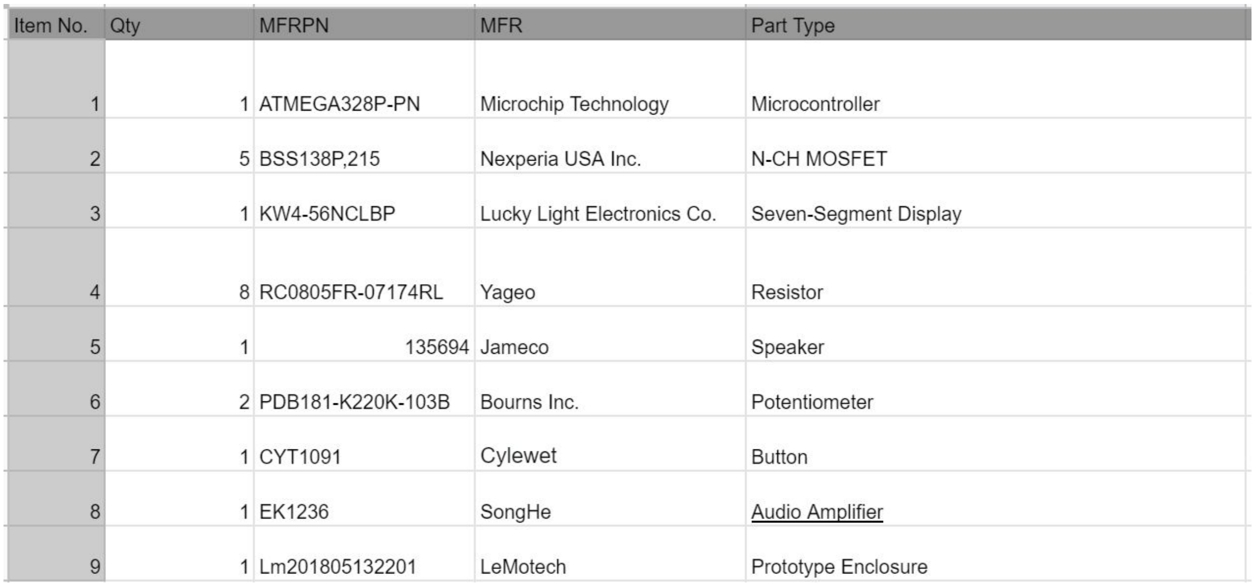


Figure 9: Fabricated PCB bottom layer.

The PCB for this project mounts the seven segment display module, as well as the current limiting resistors for the LEDs, and the MOSFETs which control which of the four digits of the display are enabled.

6. Part Information



Item No.	Qty	MFRPN	MFR	Part Type
1	1	ATMEGA328P-PN	Microchip Technology	Microcontroller
2	5	BSS138P,215	Nexperia USA Inc.	N-CH MOSFET
3	1	KW4-56NCLBP	Lucky Light Electronics Co.	Seven-Segment Display
4	8	RC0805FR-07174RL	Yageo	Resistor
5	1	135694	Jameco	Speaker
6	2	PDB181-K220K-103B	Bourns Inc.	Potentiometer
7	1	CYT1091	Cylewet	Button
8	1	EK1236	SongHe	Audio Amplifier
9	1	Lm201805132201	LeMotech	Prototype Enclosure

Figure 1: Parts Used