1. **General Information**

**Student Name: Chris Worthley**

**Student ID number: 02024207**

**Team Name/Number: ???**

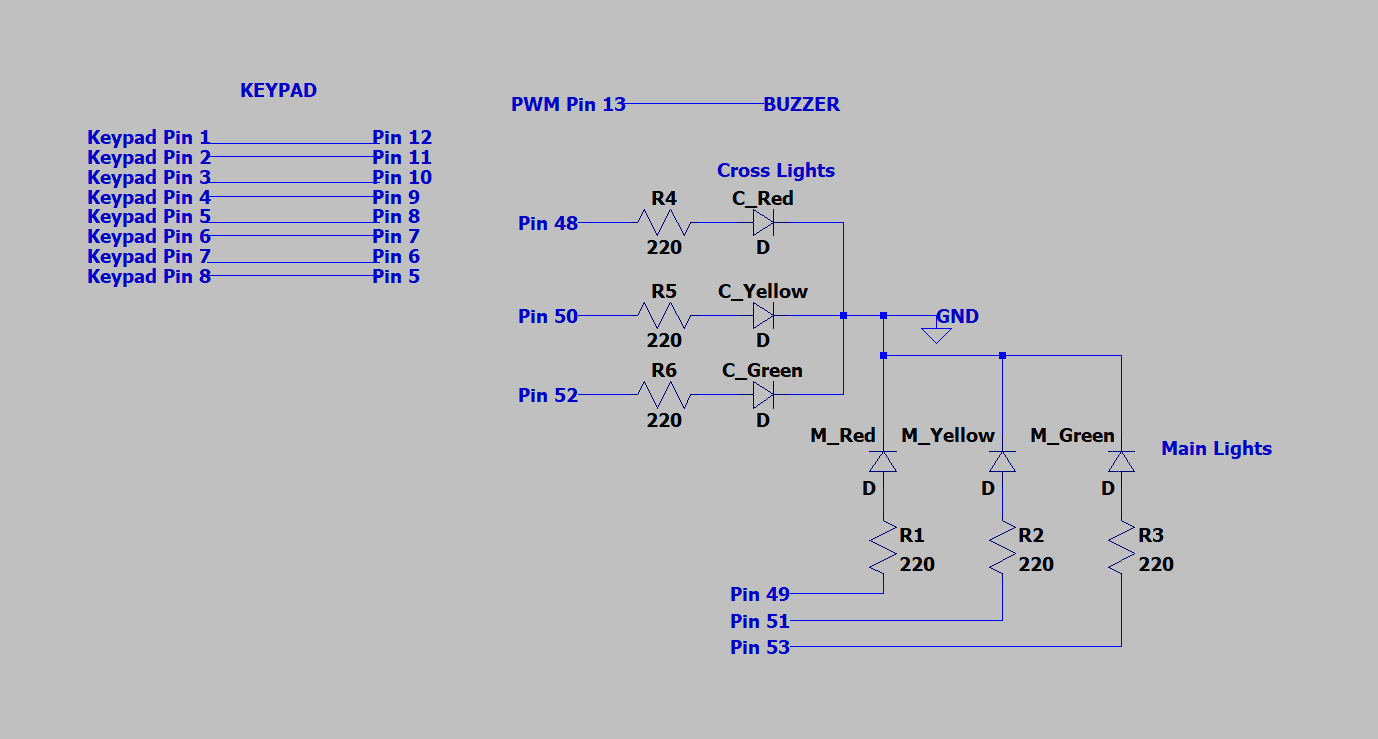
**Team member names: Daniel Burns, Himadri Saha**

**Date of completion: 10/12/25**

**Demonstration method: zoom**

1. **Design:**
   1. **Hardware Design**

Hardware schematics listed below:



**Figure 1. Hardware Schematic (Except 7-Segment Display)**

A circuit board with wires and numbers

AI-generated content may be incorrect.

**Figure 2. 7-Segment Display Wiring**

The hardware setup uses digital pins 48 through 53 to operate the LEDs for both traffic signals. The main traffic signal is connected to pins 52, 50, and 48 for the green, yellow, and red lights respectively, while the cross-traffic signal uses pins 53, 51, and 49 for its corresponding lights. Each LED is connected to its output pin through a current-limiting resistor, which is 220 ohms, to prevent excessive current draw. The Arduino outputs logic-high signals to illuminate specific LEDs according to the programmed timing sequence.

A 16-button keypad is connected to pins 12 through 5 and serves as a user input device for controlling or modifying light operation modes. The buzzer is attached to pin 13, which is a PWM capable output. The seven-segment display is interfaced through three digital pins using a shift-register configuration to reduce the number of required I/O lines. The clock, latch, and data connections for the display are assigned to pins 4, 3, and 2 respectively.

All components share a common 5V supply and ground, provided directly by the Arduino board. During operation, the Arduino cycles through the programmed states, turning on the main road green light while keeping the cross-traffic red, then proceeding through yellow and red phases before switching priority to the cross-traffic lights. The seven-segment display provides a visible countdown for each phase, while the buzzer sounds briefly to indicate transitions.

1. Personal Contribution to the Lab (Technical Details):

For this lab, I did a couple of things. Firstly, I designed the state machine used for the design of the stop light algorithm, which was the basis of the software. I set the pin configuration for the software and sourced the drivers for the keypad and wrote code to be able to receive messages from it, which is in the main code in lines 1-65. I created the hardware configuration and hardware schematic (Figures 1 and 2) for our setup, as well as authoring the hardware design section of this lab report. I also created a test function to test the hardware setup without using interrupts to confirm our hardware design before developing the final software script, which was the skeleton for our final software script.