



TRAFFIC LIGHT – ENHANCED

Jawad Kadri – 101147056

April 11, 2022

Contents

Implementation	2
Tinkercad.....	3
Components Used.....	3
Topics Covered.....	4
Design – Flowchart.....	4
Issues Faced	5
Extra Features	5
References / Inspiration.....	6
Video Showcase	6

Implementation

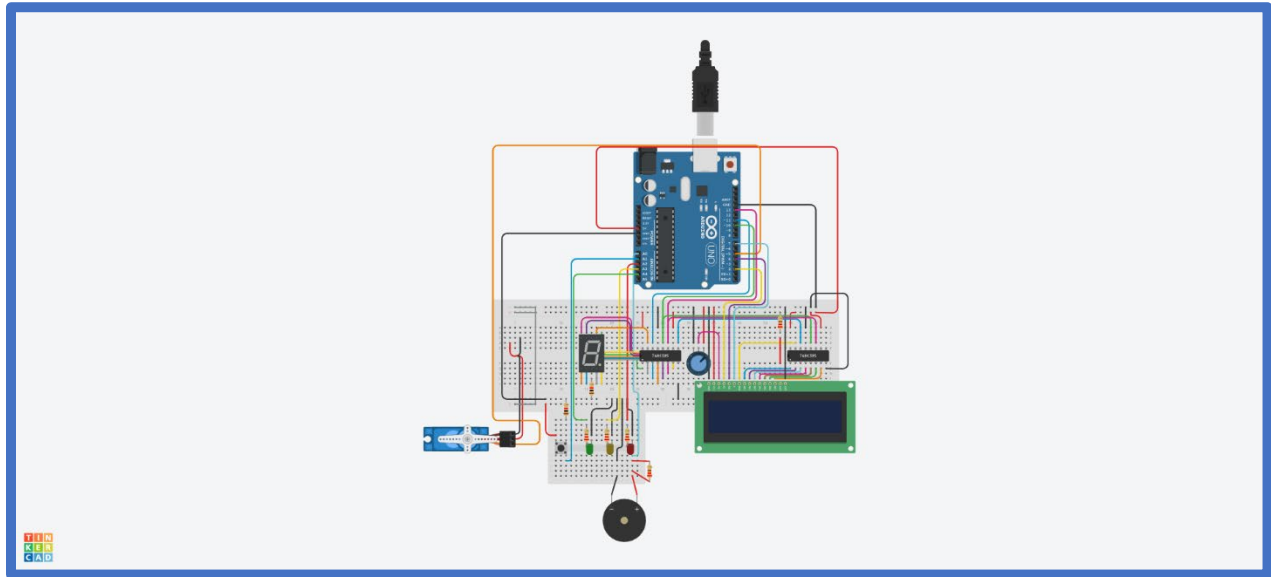
We implemented an enhanced version of the original traffic light program from Assignment 1. A lot of changed from visual to the code itself have been made for the creation of this project. Let's begin with what each element is doing.

- LEDs – The LEDs act as the visual view of the traffic light.
 - Green → Go
 - Yellow → Slow Down / Yield
 - Red → Stop
- Button – When the user pushes the button the delay is reduced by half so the cross walkers don't have to wait for a long period of time to cross the road.
- Buzzer – Unfortunately the original assignment wasn't friendly for the visually impaired, so we fixed it by adding an Active Buzzer to sound when its safe for people to cross.
- 7 Segment 1 Digit Display – When the last 10 seconds of the timer is left (9-0) the display is used to notify the people that they'll be able to cross the road soon.
- Servo Motor – Acts as the stopper so cars don't cross the intersection when a train is coming.
- LCD – Displays the intersection and what is currently happening. This provides us with a fancy view of our intersection.

In terms of the code for the program we aren't using the original `_delay_ms` function nor are we using an if statement to check if the button was pressed just like how we did it from Assignment 1. Instead, we are using timers as a delay and interrupts for when the button is pressed. Furthermore, we introduced more features such as the Servo Motor, LCD, and Buzzer that wasn't in the original Assignment.

You can probably tell that our Arduino wouldn't have enough pins for all these components however with the implementation of Shift Registers and Serial Peripheral Interface we are able to fit them all.

Tinkercad



Components Used

The following table below is all the devices that are implemented.

* Refers to Approximately

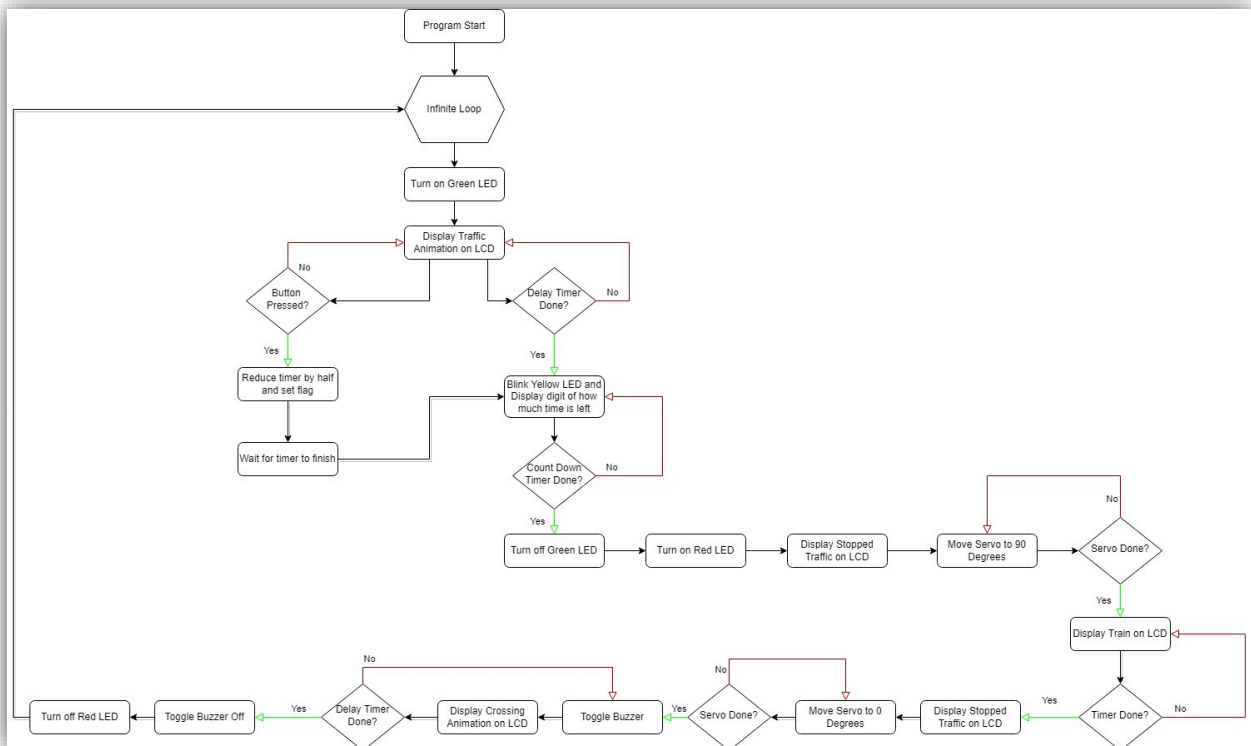
	#	Components Used
Mandatory	3	LEDs
	1	Push Button
	1	LCD
	1	1 Digit 7-Segment LED with Potentiometer
	2	74HC959 Shift Register
Optional	1	Servo Motor
	1	Active Buzzer
Other	1	Power Supply
	2	1K Resistor
	5	220 Ω Resistor
	56*	Jumper Wires
	1	Breadboard
	1	Mini-Breadboard
	1	Arduino Board

Topics Covered

The list below is all the topics that are covered in this project.

- Bitwise Operators
- Interrupts (SEI) → Push button
- Timer 0 (PWM) → Servo Motor
- Timer 1 (Normal Mode) → Delay Timer
- LCD
- 7 Segment 1 Digit Display
- Servo Motor
- Serial Peripheral Interface
- Rand → Function to get random value

Design – Flowchart



Issues Faced

- Couldn't/Failed to get 4-digit 7 segment display. Due to lack of understanding and limitation of time we didn't have the ability to get the 4-digit led working.
- The servo motor and delay timer couldn't be used on the same timer. What would happen is when the timer was configured in PWM the LCD would no longer work because the delayMs was using the same timer. The solution was to use Timer0 PWM for the Servo Motor and Timer 1 Normal Mode for the delay timer.
- The servo motor wouldn't run smoothly not because of the timer, instead it was due to low battery life. If the battery is low the servo motor turning speed is affected drastically.
- Lots of wiring issues. Jawad Kadri constantly had to re-adjust which wires were connected to what pins to compensate for which pins needed to be connected to PWM or in a normal pin.
- Due to a lack of Arduino Pins we required another Shift Register, so Jawad Kadri got another one (He choose to purchase Elegoo's Upgraded Electronics Fun Kit).
- Power issues. Due to the servo motor and LCD being power hungry we needed to split the load between the two. So, we decided to use a power supply and off load some of the power so the Arduino wouldn't shut off.

In terms of bugs, we don't believe we have any. The program works exactly as intended.

Extra Features

- Custom Icons for LCD
 - Train
 - Stop sign
- Added Two more LEDs to simulate a visual warning that the train is coming
- Proper/Fluid animation of LCD instead of flashing between two frames
- Not exactly a feature, but to fix how the wiring looks. It's very messy and could be tidied up a bit.

References / Inspiration

If you noticed every piece of this project is based off labs or topics covered within the course.

The biggest inspiration is based off assignment 1 from the course. In that assignment we had to create a basic traffic light using 3 LEDs, 1 digit 7 Segment LED, and a button. However, in that assignment all the code was created before we had the knowledge of timers and interrupters. Meaning most of the code had to be re-worked, however the main purpose and the interaction the user has with the program remains almost the same.

A few examples of what projects we incorporated into the program are:

- Obviously, the LEDs, Button, and 7 Segment Display are all based off assignment 1.
- The Servo Motor + Power Supply were based off a lab where we used PWM timers to adjust the angle of the Servo Motor.
- The LCD is based on multiple labs where we used the LCD and became comfortable using it.
- The Shift Registers are also based on labs we covered towards the end of the course.

We're extremely glad we could use the traffic light concept for this project because comparing our traffic light from the original assignment we've learned a lot since then and come to understand how each part of a Real-Time System works.

Video Showcase

Below Links a video of our program in action. Link: <https://youtu.be/8LEq2iYbog0>

