2. Objective:

This experiment's major objective is to become familiar with microcontroller timers so that it may analyze the blink test while developing a traffic control system. Tinker cad and the Arduino IDE ~~software~~ were the programs utilized in this experiment. The first one was completed using the Arduino IDE software and equipment that was set up during the lab session. Tinker cad program was used as an alternative way at home. The simulation was carried out utilizing the tinker cad program. This experiment's major objective is to acquaint me with microcontroller timers so that I can study the blink test while using and developing a traffic control system.

3. The apparatus and software name:

- Arduino IDE (any version)
- Arduino Uno (R3) board or Arduino mega 2560 board
- LED lights (RED, GREEN, and YELLOW) and three 200 ~~ohm~~ ohms resistors and jumper wires
- Breadboard

1. Theory and programs:

Each electrical component in a circuit bore sequential logic ~~operators~~ operates on a time foundation. All of the work is kept in sync thanks to this time base. Devices would be enable to determine when to carry out specific activities without a time base. The timer is a crucial idea in the realm of electronics.

The Arduino controller includes a timer/counters as hardware. It functions like a clock and maybe utilized to gauge time-related activities. A register that automatically raises or lowers in value is a timer. In VAR, timers are of two types: 8-bit and 16-bit timers. In an 8-bit timer, the register used is 8-bit wide whereas, in a 16 bit timer, the register width is 16 bits. This means that the 8-bit timer is capable of counting $2^8 = 256$ steps from 0 to 255. Similarly, a 16-bit timer is capable of counting $2^{16} = 65536$ steps from 0 to 65535.

5. A brief procedure:

- First we familiarized ourselves with the different type Arduino family components. Arduino UNO Microcontroller is one of them.
- After that, we know the basic working principle of tinker cad software.
- At first we need to open tinker cad online software to implement the given circuit. Then we select different type of components to implement it.
- Next we write the code to run traffic light system.
- After that we implement the circuit in the bread board and connect this experiment in the hardware.

8

Source code:

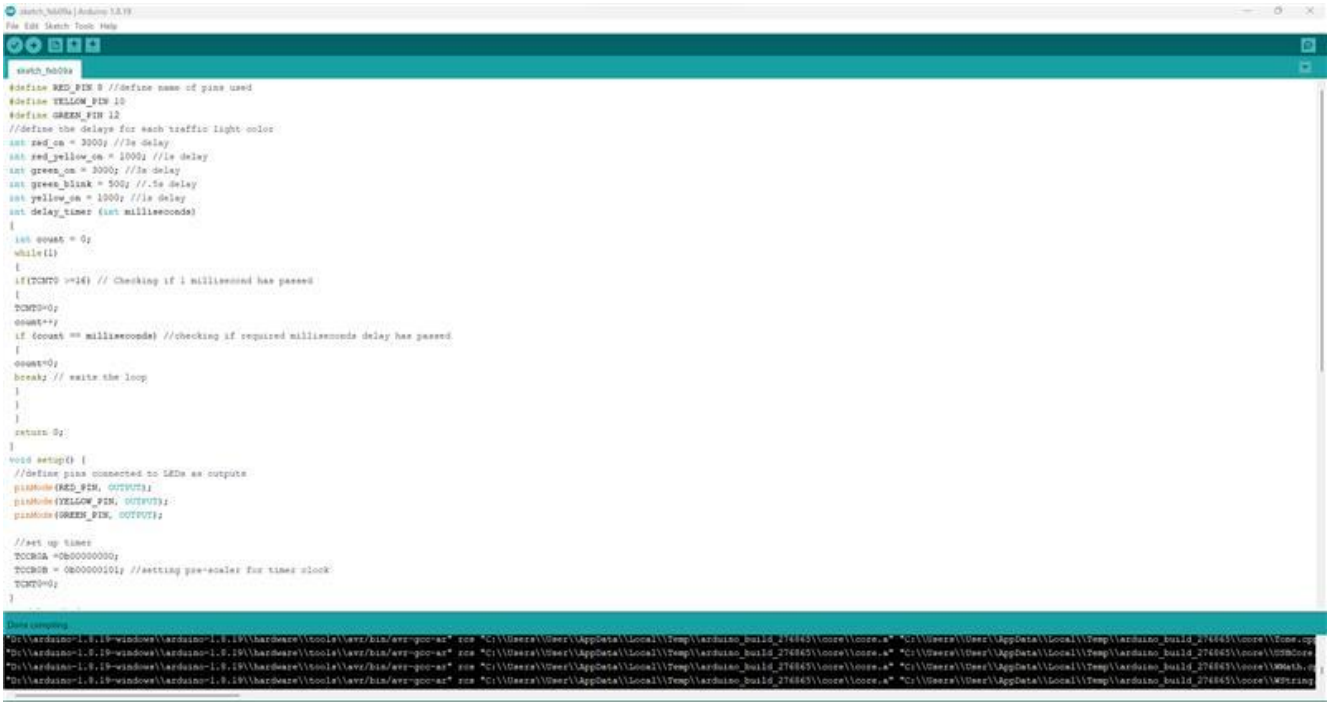

Figure 1.1: Source code for Traffic Control System

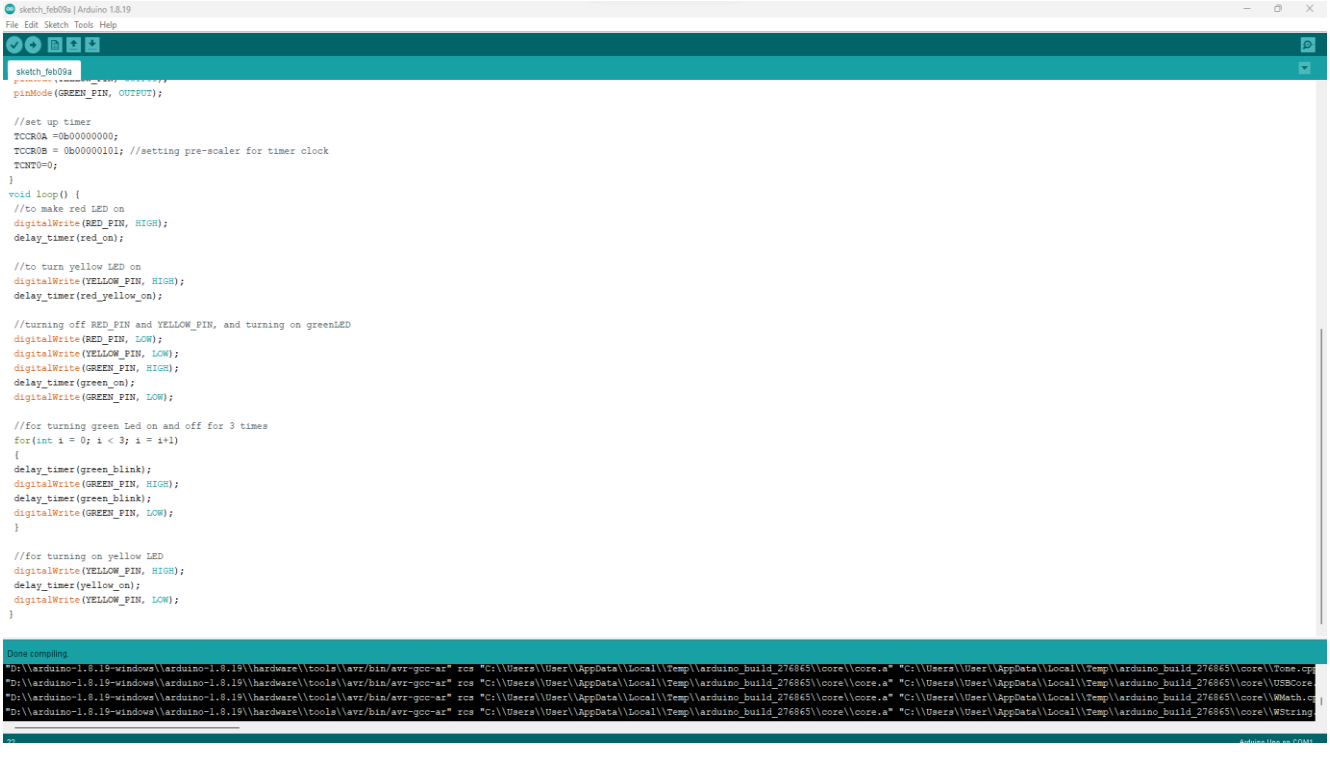

Figure 1.2: Source code for Traffic Control System

Hardware Implementation:

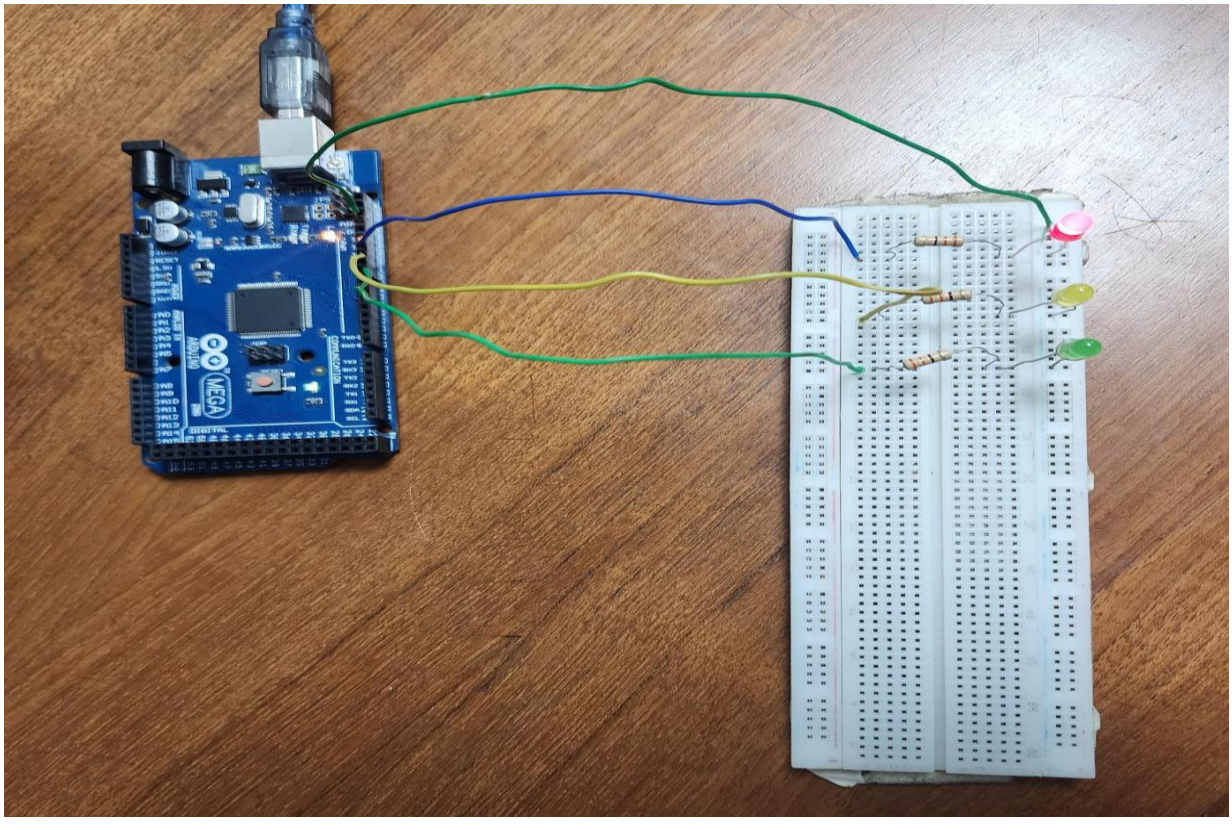

Figure 2.1: Traffic Control System - Red Light On

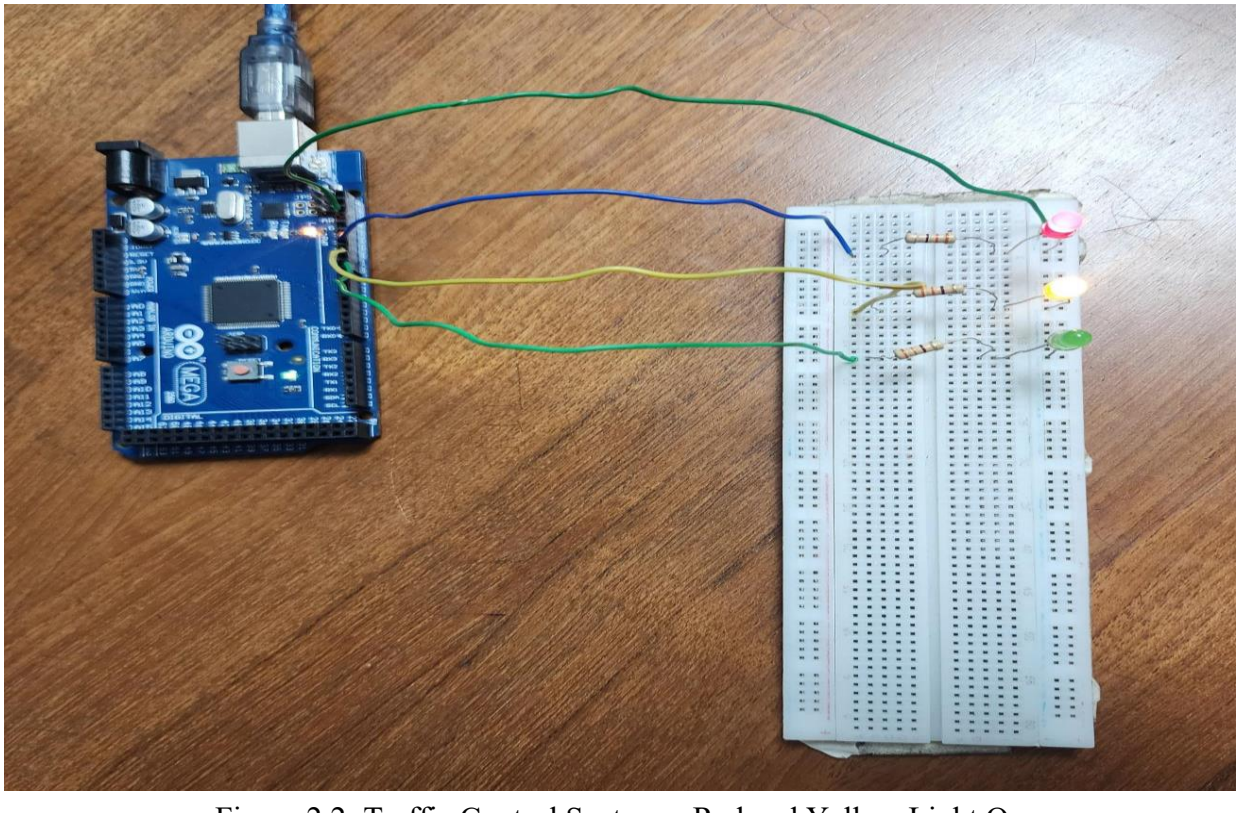

Figure 2.2: Traffic Control System – Red and Yellow Light On

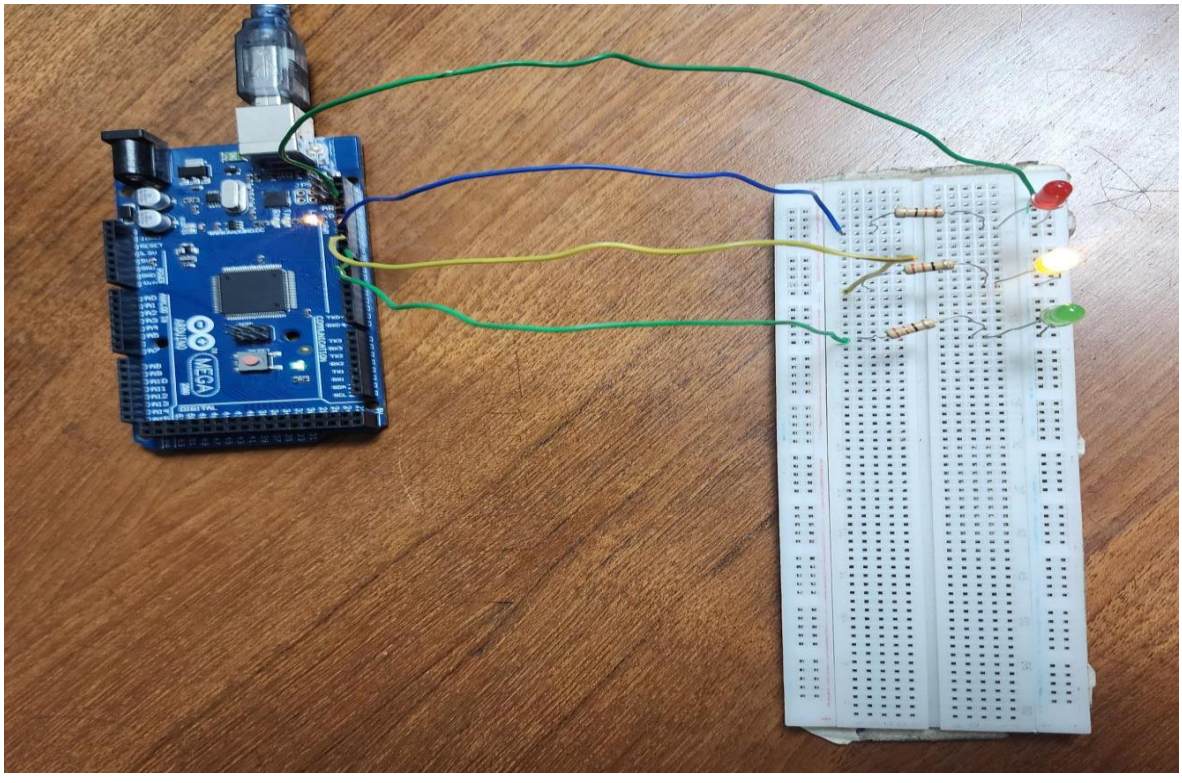

Figure 2.3: Traffic Control System - Yellow Light On

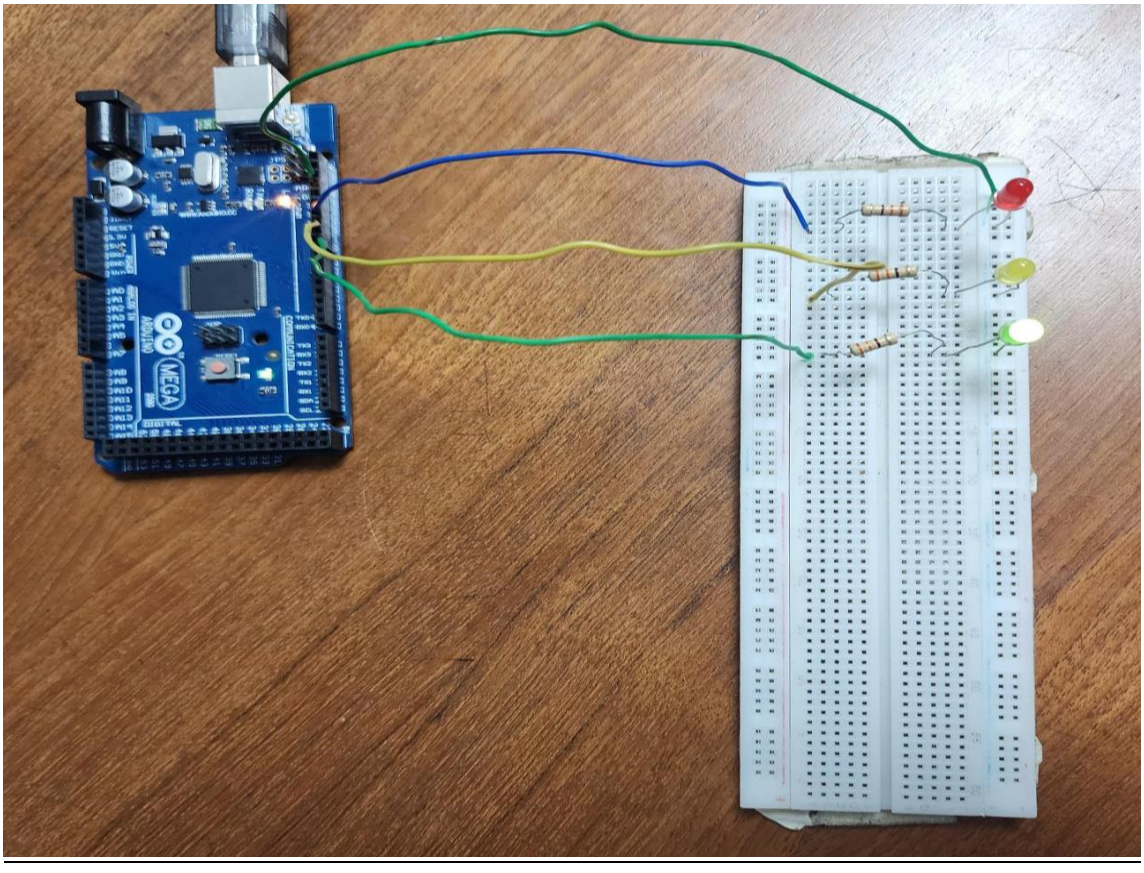

Figure 2.4: Traffic Control System - Green Light On

Simulation:

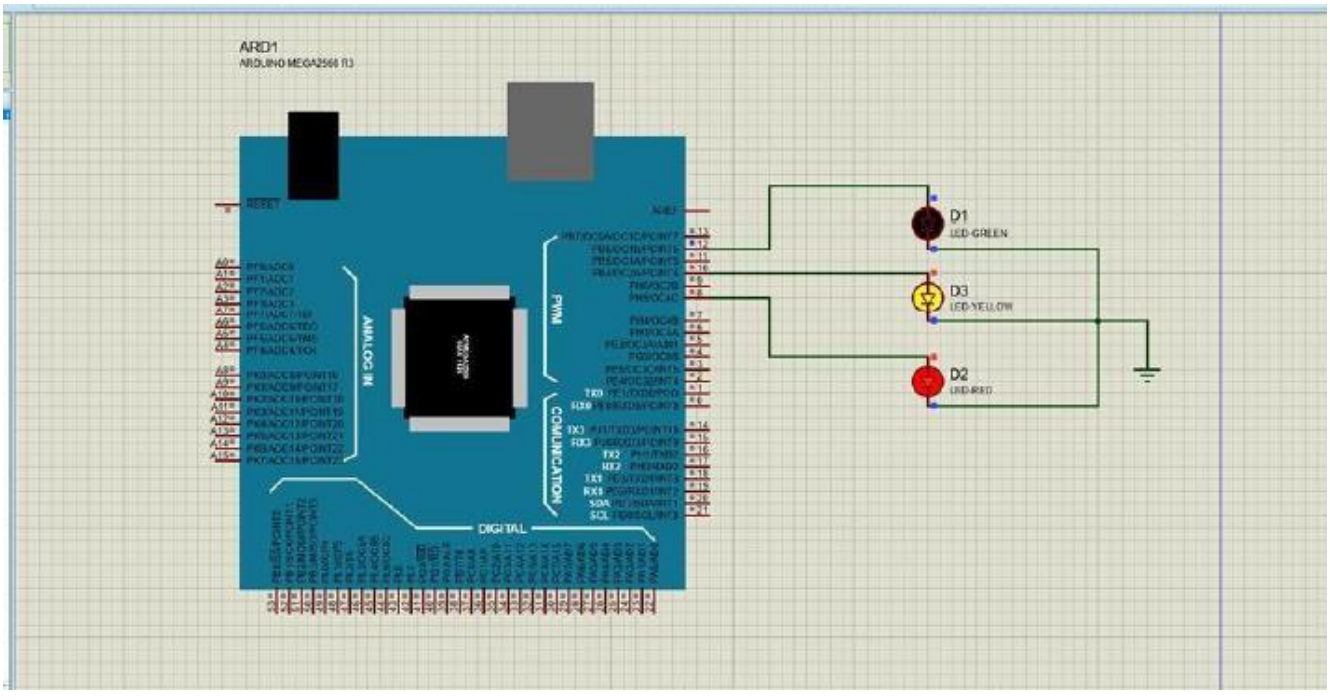

Figure 3.1: Traffic Control System – Red and Yellow Light On

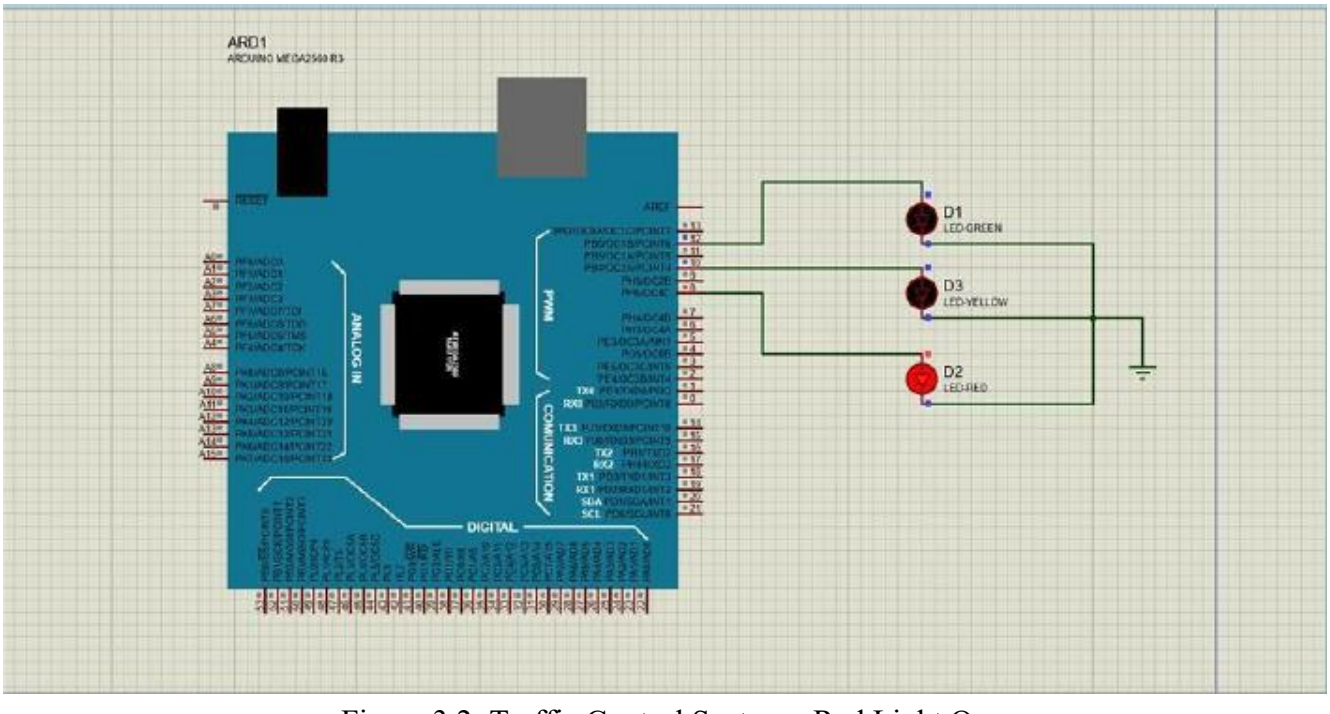

Figure 3.2: Traffic Control System - Red Light On

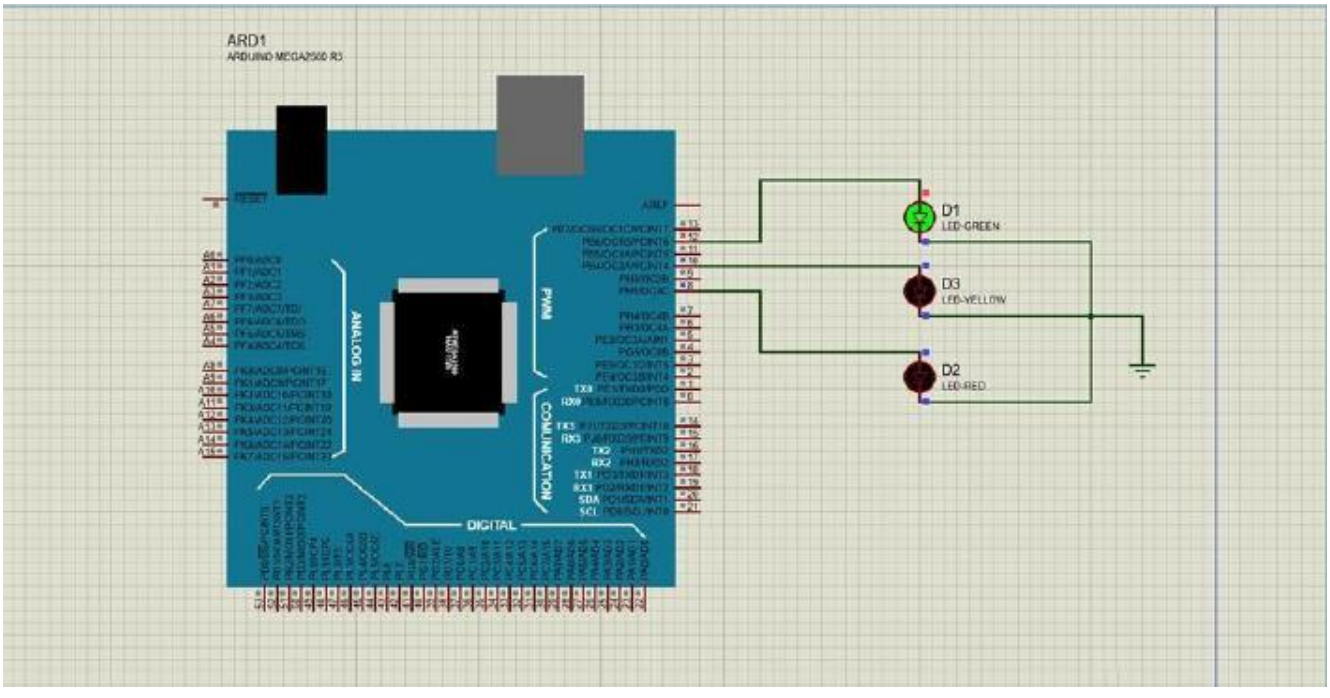

Figure 3.3: Traffic Control System - Green Light On

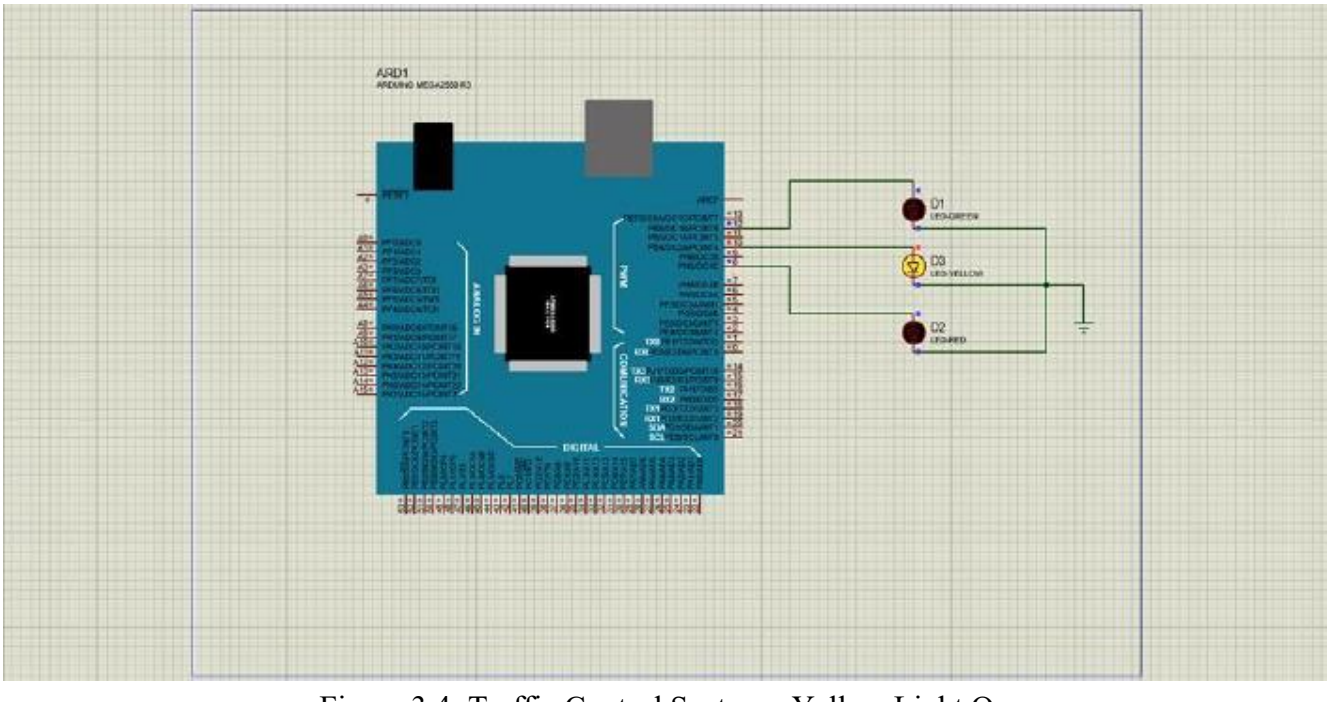

Figure 3.4: Traffic Control System - Yellow Light On

Report:

All codes, scripts and proteus simulation of the blink program and traffic light system is attached above.

8. Discussions:

Two methods were used to conduct this experiment. The traffic control system was first built using an Arduino board, three different colored animated LED lights (Red, yellow, green), three resistors on a breadboard, and connecting cables. Ports 8, 10, and 12 were then linked to the red, yellow and green LED lights. Then, using the Arduino IDE and timers, some code for the traffic control system was written. Following the Arduino board's connection to the computer, the code was executed to determine the outcome of the traffic signal. This experiment was also carried out with the aid of Tinker cad software. Some problems arose as this experiment was being conducted. Both coding flaws and incorrect pin configurations were the problems. However, a lab manual and the internet were used to find the answers to this problems. The traffic system's desired outcome could be attained in both cases.

9. Conclusions:

Microcontrollers were learned while working on this experiment, and at Arduino Uno traffic control system was developed. Our course is instructor explained the entire procedure and guided us through it so that we could properly run this code. Working with Arduino for the first time was difficult at first, but it became easier with time. However, with the assistance of our tutor, the code was eventually properly executed. As a result, we were able to finish our report.