AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

Faculty of Engineering

Lab Report

Experiment # 03

Experiment Title: Familiarization with the Timers of an Arduino Microcontroller Board, the study of LED blink test, and implementation of a simple traffic control system using its Timer0 function.

Date of Perform:	March 19, 2025	**Date of Submission:**	April 10, 2025
Course Title:	Microprocessor and Embedded Systems Lab		
Course Code:	EE4103	**Section: G**	
Semester:	Spring 2024-25	**Degree Program:**	BSc in CSE/EEE
Course Teacher:	**Prof. Dr. Engr. Muhibul Haque Bhuyan**		

Declaration and Statement of Authorship:

1. I/we hold a copy of this Assignment/Case Study, which can be produced if the original is lost/damaged.
2. This Assignment/Case Study is my/our original work; no part has been copied from any other student's work or any other source except where due acknowledgment ismade.
3. No part of this Assignment/Case Study has been written for me/us by any other person except where such collaboration has been authorized. by the concerned teacher and is acknowledged in the assignment.
4. I/we have not previously submitted or am submitting this work for any other course/unit.
5. This work may be reproduced, communicated, compared, and archived to detect plagiarism.
6. I/we permit a copy of my/our marked work to be retained by the Faculty Member for review by any internal/external examiners.
7. I/we understand that Plagiarism is the presentation of the work, idea, or creation of another person as though it is your own. It is a form of cheating and is a very serious academic offense that may lead to expulsion from the University. Plagiarized material can be drawn from, and presented in, written, graphic, and visual forms, including electronic data, and oral presentations. Plagiarism occurs when the origin of the source is not appropriately cited.
8. I/we also understand that enabling plagiarism is the act of assisting or allowing another person to plagiarize or copy my/our work.

** Student(s) must complete all details except the faculty use part.*
** Please submit all assignments to your course teacher or the office of the concerned teacher.

Group # 2

Sl No	Name	ID	PROGRAM	SIGNATURE
1	MAHBUB HASAN	22-47419-2	BSc in CSE	
2	MUJTAHID TABASSUM	22-47430-2	BSc in CSE	
3	MD.SALMAN ISLAM	22-47393-2	BSc in CSE	
4	MD. SIAM MEHEDI	22-48342-3	BSc in CSE	
5	MD TOYEB ALI	23-51819-2	BSc in CSE	

Faculty use only

FACULTY COMMENTS	**Marks Obtained**	
	Total Marks	

Table of Contents

Marking Rubrics (to be filled by Faculty):

Level Category	Excellent [5]	Proficient [4]	Good [3]	Acceptable [2]	Unacceptable [1]	No Response [0]
Title and Objectives	Able to clarify the understanding of the lab, no issues are missing and formatting is good.	Able to clarify the understanding of the lab experiment, no issues are missing but its formatting is not good.	Able to clarify the understanding of the lab experiment, but a few issues are wrong, and its formatting is bad.	Able to clarify the understanding of the lab experiment, but it lacks a few important issues of the experiment without maintaining the format.	Unable to clarify the understanding of the lab experiment.	No Response/ copied from others/ identical submissions with gross errors/image file printed
Codes and Methods	Able to explain the experimental codes and simulation methods using Proteus very well.	Able to explain the experimental codes and simulation methods using Proteus but is not formatted well.	Able to explain the experimental codes but simulation method using Proteus is not explained well.	Presents the experimental codes but didn't explain simulation methods using Proteus clearly.	Presents the experimental codes but didn't explain simulation methods using Proteus.	
Results	Key results and images are there. Figures/Tables have all identifications and refer to them properly in the texts.	Key results and images are there. Figures/Tables have all identifications, such as the axis labels, numbers, and captions with a few minor errors; the texts refer them.	Key results and images are there. Figures/Tables lack a few identifications, such as the axis labels, numbers, and captions; the texts refer them.	Misses several key results and images. Figures/Tables lack identification, such as the axis labels, numbers, and captions; the texts don't refer them.	Major results, such as experimental and simulation results' images are not included. Figures and tables are poorly constructed or not presented.	
Discussion and Conclusion	Proper interpretation of results and summarizes the results to draw a conclusion, discusses its applications in real-life situations to connect with the report's conclusion.	Proper interpretation of results and summarizes the results to draw a conclusion but didn't discuss its applications in real-life situations to connect with the conclusion of the report.	Interpretation of results is presented. However, there is a disconnect between the results and discussion.	Misses the interpretation of key results. There is little connection between the results and discussion.	Very poor interpretation of the results. No connection between results and discussions.	
Question and Answer	Able to produce all questions' answers correctly maintaining the lab report format.	Able to produce all questions' answers but didn't maintain the lab report format.	Able to produce all questions' answers but wrong answers to a few questions.	Able to produce all questions' answers but wrong/missing answers to multiple questions.	Unable to produce all questions' answers and completely wrong answers.	
Comments						**Total Marks (25)**

Experiment Title: Familiarization with the Timers of an Arduino Microcontroller Board, the study of LED blink test, and implementation of a simple traffic control system using its Timer0 function.

Objectives:

The objectives of this experiment are to-

1. Study the Timers of an Arduino Microcontroller Board.
2. Learn basic programming commands of the Timer functions.
3. Apply the coding techniques of the Timer functions.
4. Implement the LED blink test using Timer0 of an Arduino Microcontroller Board.
5. Implement a traffic light control system using an Arduino Microcontroller Board.

Equipment List:

1. Arduino IDE (2.0.1 or any recent version)
2. Arduino Microcontroller board
3. LED lights (Red, Green, and Yellow)
4. Three 100 Ω resistors
5. Jumper wires

Circuit Diagram:

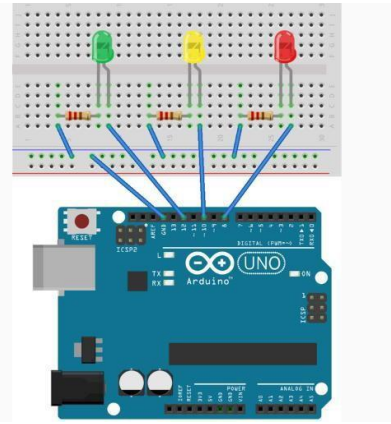

Figure 1: Experimental Setup of a Traffic Control System

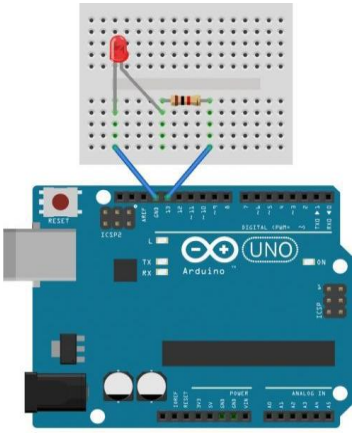

Figure 2: Experimental Setup of a LED Light Blinking test

Experimental Output Results:

For the Led blink test a led was placed in a breadboard following a resistor connected to ground and the anode of the LED went to the Arduino board pin no. 10 as it was configured as output pin. After uploading the code on the Arduino board, the program ran successfully, and the expected output was found. For the Traffic light experiment 3 different colors led was placed in a breadboard following 3 resistors connected to the ground. A ground wire connected to the ground from the Arduino board. Then the anode of the LEDs connected to the pin no. 8, 10 and 12 respectively as it was configured as output pin the Arduino program. The program was uploaded on the Arduino board and the program was run successfully and the desired output was found.

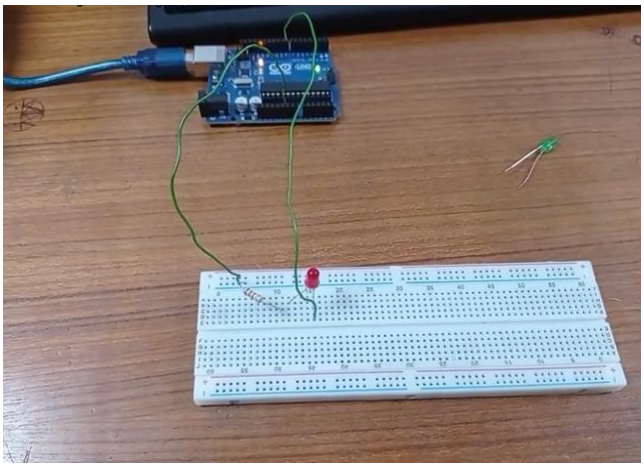

Figure 3: (A) LED Blink Test OFF

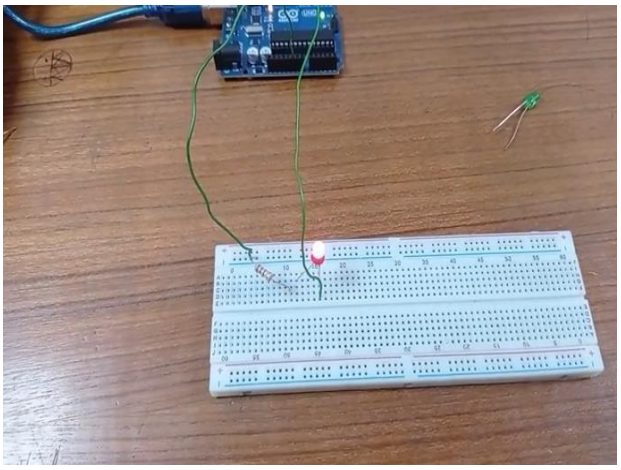

Figure 3: (B) LED Blink Test ON

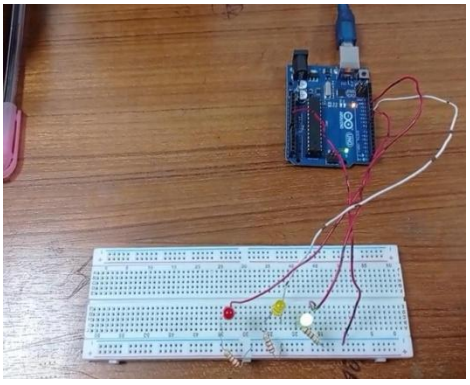

Figure 4: (A) Traffic light Green

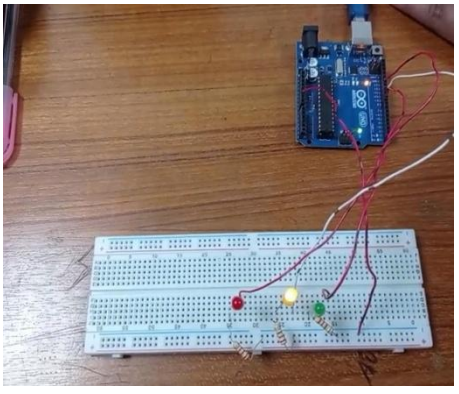

Figure 4: (B) Traffic light Yellow.

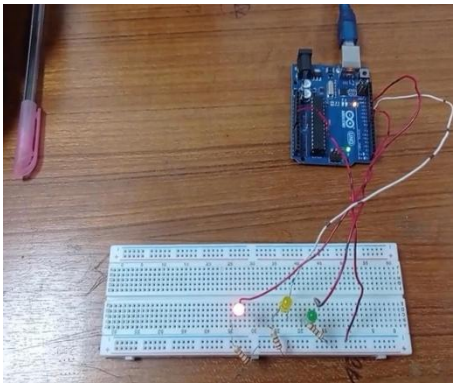

Figure 4: (C) Traffic light Red

Simulation Output Results:

The experimental setup was established according to fig.1 and fig.2 in the proteus app. The code was compiled in Arduino IDE and generated the .hex file. The .hex file was uploaded to the Arduino board on the simulation application. Thus, the program was run successfully.

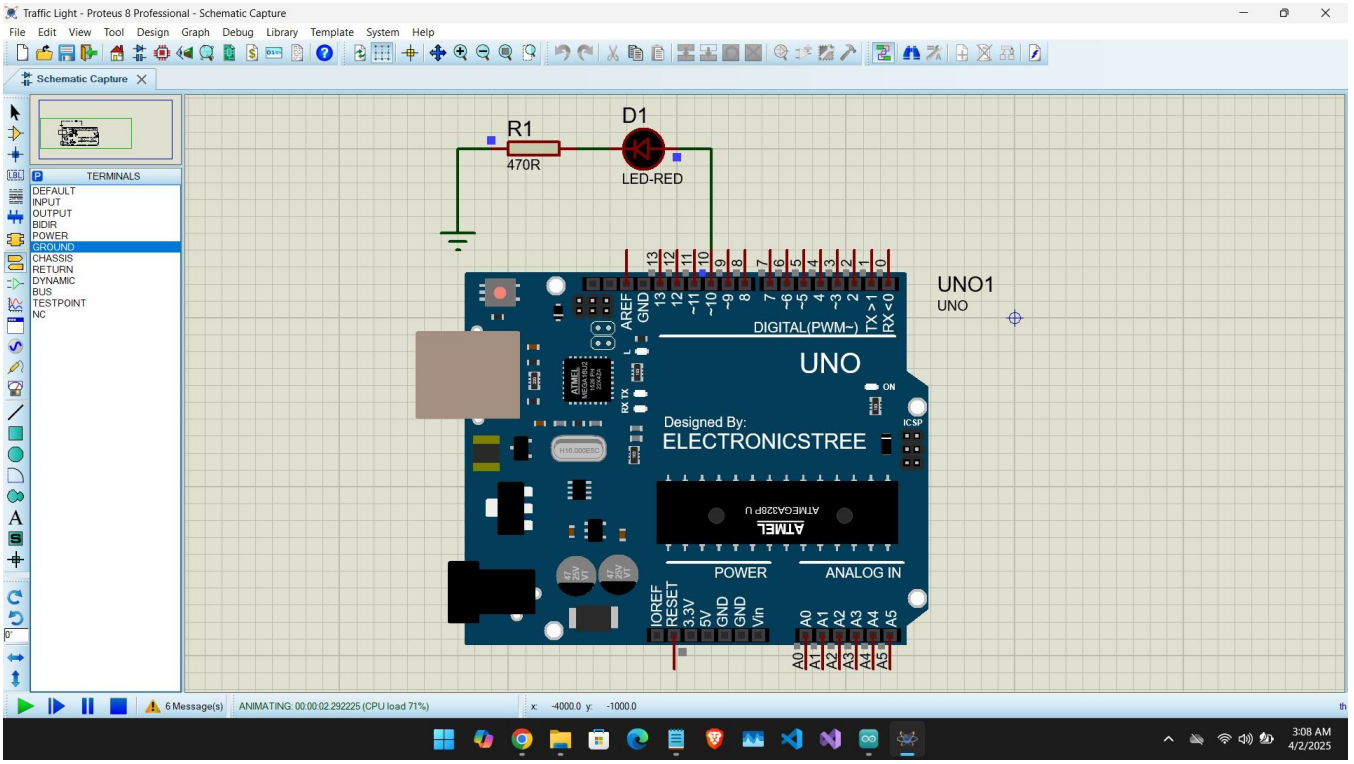

Figure 5: Simulation results of LED off.

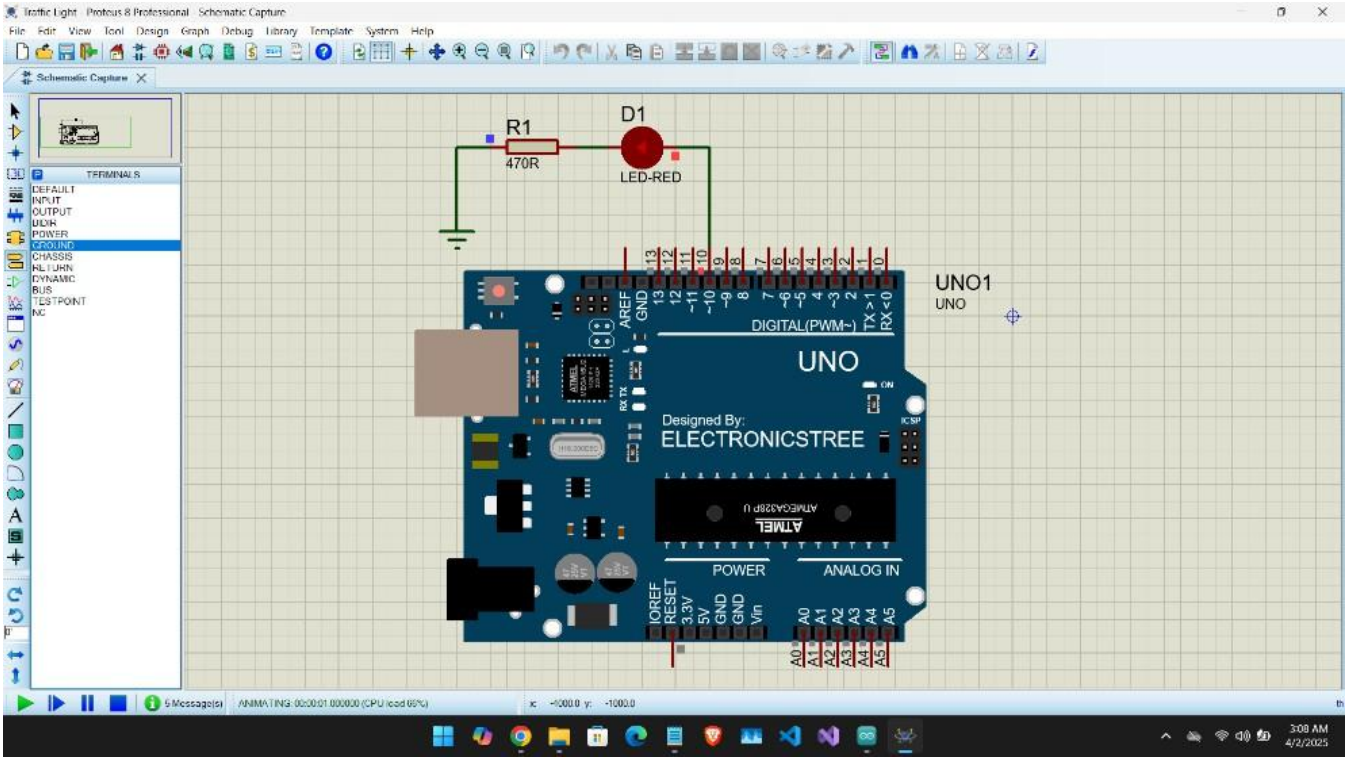

Figure 6: Simulation results of LED on.

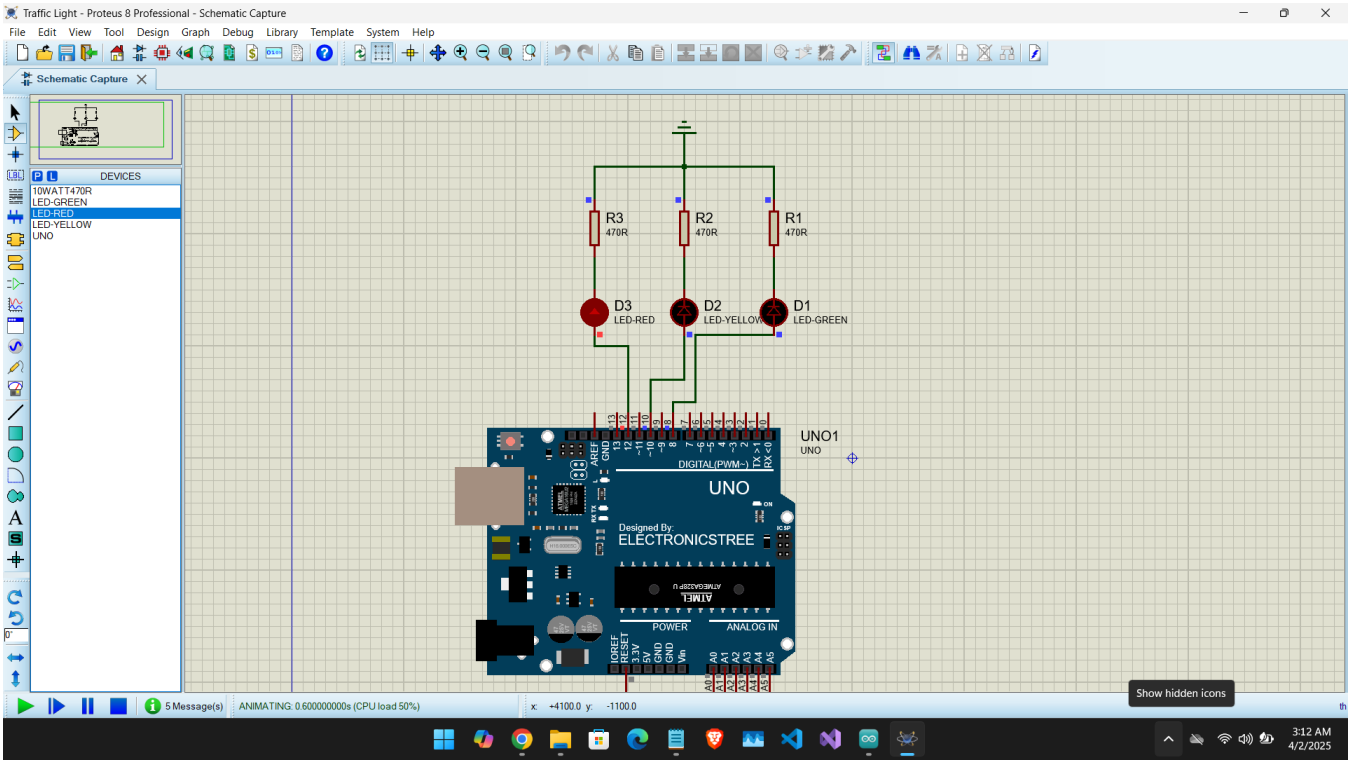

Figure 7: (A) Traffic Light RED

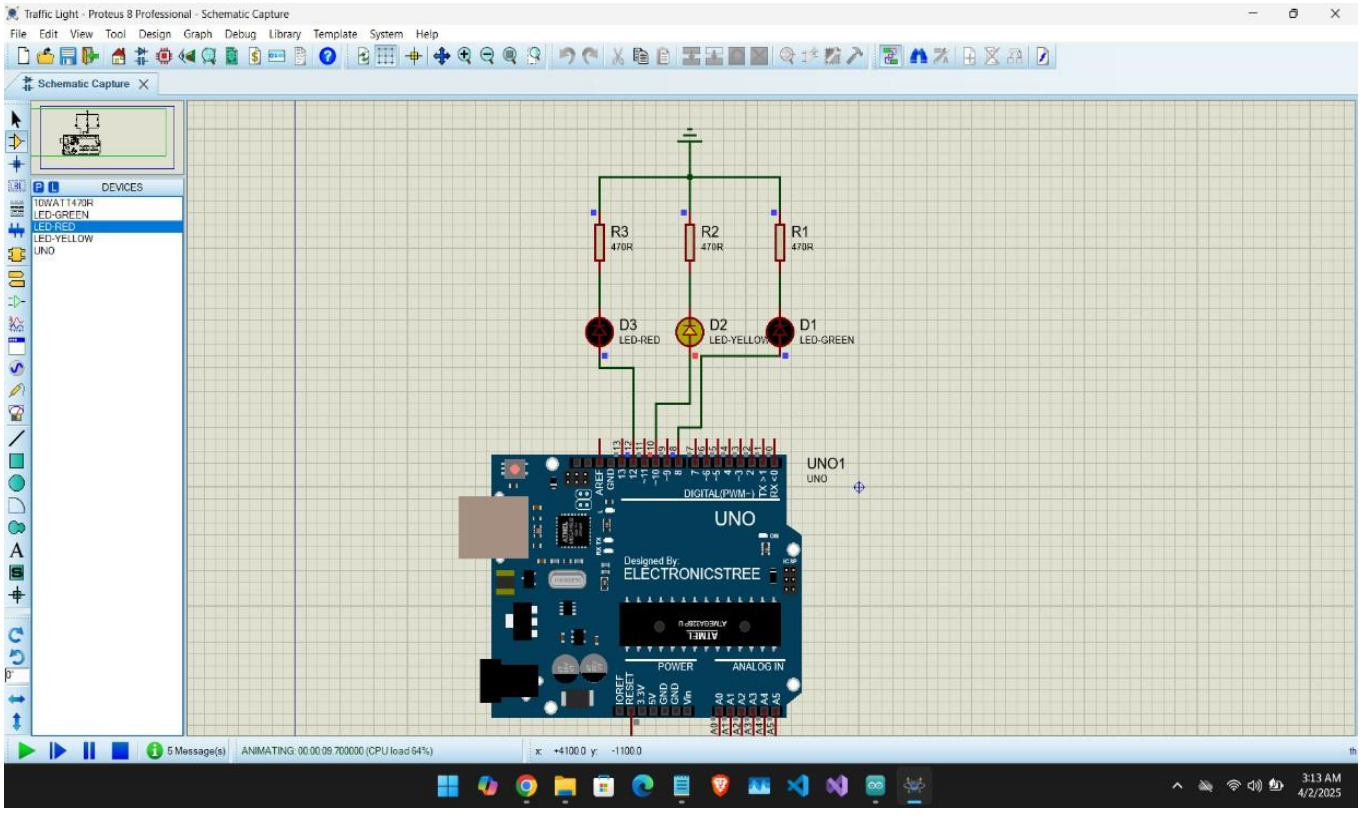

Figure 7: (B) Traffic Light YELLOW

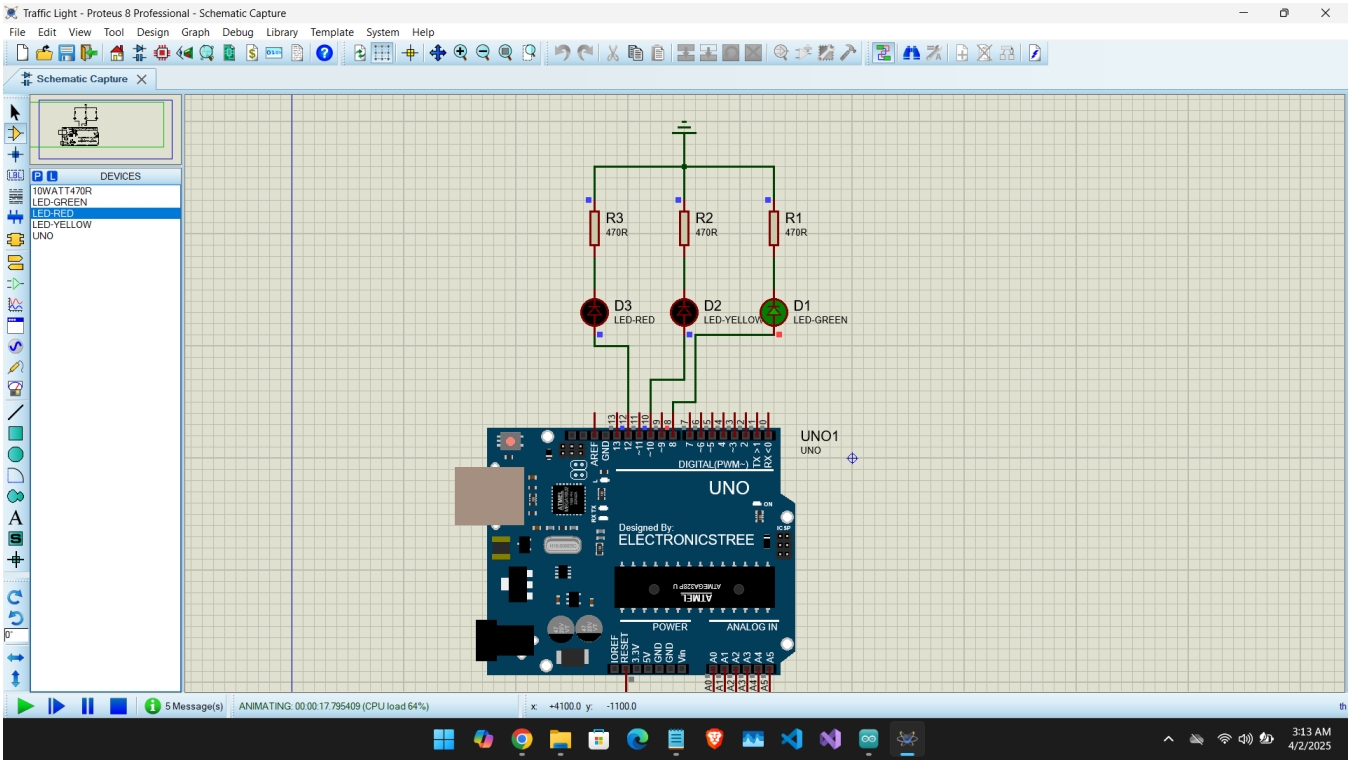

Figure 7: (C) Traffic Light GREEN

Answers to the Questions in the Lab Manual:

1. **Code for LED blink test**

```
// Pin numbers are defined by the corresponding LED colors
#define YELLOW_PIN 10
// The delay amounts are declared in ms; 1 s = 1000 ms
int red_on = 2000;
// Declaring the timer function to count 1 ms of delay instead of calling the delay function
int delay_timer (int milliseconds){
int count = 0;
while(1)
{
if(TCNT0 >= 16) // Checking if 1 millisecond has passed
{
TCNT0=0;
count++;
if (count == milliseconds) //checking if required milliseconds delay has passed
{
count=0;
break; // exits the loop
}
}
}
return 0;
}
void setup() {
//define pins connected to LEDs as outputs
pinMode(RED_PIN, OUTPUT);
//set up the timer using timer registers
TCCR0A = 0b00000000;
TCCR0B = 0b00000101; //setting pre-scaler for timer clock
TCNT0 = 0;
}
void loop() {
//to make the red LED turned on
digitalWrite(RED_PIN, HIGH);
delay_timer(red_on);
//turning off the red LED
digitalWrite(RED_PIN, LOW);
delay_timer(red_on);
}
```

2. **Code for Traffic light control system with timer()**

```
// Pin numbers are defined by the corresponding LED colors
#define RED_PIN 8
#define YELLOW_PIN 10
#define GREEN_PIN 12
//Define the delays for each traffic light color
int red_on = 6000; //3 s delay
int green_on = 3000; //3 s delay
int yellow_blink = 500; //0.5 s delay
// Declaring the timer function to count 1 ms of delay instead of calling the delay function
int delay_timer (int milliseconds){
int count = 0;
while(1)
{
if(TCNT0 >= 16) // Checking if 1 millisecond has passed
{
TCNT0 = 0;
count++;
if (count == milliseconds) //checking if required milliseconds delay has passed
{
count = 0;
break; // exits the loop
}
}
}
return 0;
}
void setup() {
//Define pins connected to LEDs as outputs
pinMode(RED_PIN, OUTPUT);
pinMode(YELLOW_PIN, OUTPUT);
pinMode(GREEN_PIN, OUTPUT);
//Set up the Timer0
TCCR0A = 0b00000000;
TCCR0B = 0b00000101; //setting pre-scaler for timer clock
TCNT0 = 0;
}
void loop() {
//to make the green LED turned on
digitalWrite(GREEN_PIN, HIGH);
delay_timer(green_on);
//to make the green LED turned off
```

```
digitalWrite(GREEN_PIN, LOW);
//for turning the yellow LED on and off for 4 times
for(int i = 0; i < 4; i = i+1)
{
digitalWrite(YELLOW_PIN, HIGH);
delay_timer(yellow_blink);
digitalWrite(YELLOW_PIN, LOW);
delay_timer(yellow_blink);
}
//to make the red LED turned on
digitalWrite(RED_PIN, HIGH);
delay_timer(red_on);
//to make the red LED turned off
digitalWrite(RED_PIN, LOW);
}
```

3. Configuring this system to have delays for output according to my id – 22-47393-2 considering the last 3 digits 3 second for red 9 second for yellow and 3 second for green LED.

```
#define RED_PIN 8
#define YELLOW_PIN 10
#define GREEN_PIN 12

// Updated delays for each traffic light color
int red_on = 3000;      // 3 seconds
int green_on = 3000;    // 3 seconds
int yellow_blink = 500; // 0.5 seconds per blink cycle (ON + OFF = 1 sec)

// Custom delay function using Timer0
int delay_timer(int milliseconds) {
  int count = 0;
  while (1) {
    if (TCNT0 >= 16) { // Approx. 1 millisecond
      TCNT0 = 0;
      count++;
      if (count == milliseconds) {
        count = 0;
        break;
      }
    }
  }
```

```
  return 0;
}

void setup() {
  pinMode(RED_PIN, OUTPUT);
  pinMode(YELLOW_PIN, OUTPUT);
  pinMode(GREEN_PIN, OUTPUT);

  // Timer0 setup
  TCCR0A = 0b00000000;
  TCCR0B = 0b00000101; // Prescaler set to 1024
  TCNT0 = 0;
}

void loop() {
  // Green LED ON for 3 seconds
  digitalWrite(GREEN_PIN, HIGH);
  delay_timer(green_on);
  digitalWrite(GREEN_PIN, LOW);

  // Yellow LED blink for 9 seconds -> 9 total blinks (each blink = 1 sec)
  for (int i = 0; i < 9; i++) {
    digitalWrite(YELLOW_PIN, HIGH);
    delay_timer(yellow_blink);
    digitalWrite(YELLOW_PIN, LOW);
    delay_timer(yellow_blink);
  }

  // Red LED ON for 3 seconds
  digitalWrite(RED_PIN, HIGH);
  delay_timer(red_on);
  digitalWrite(RED_PIN, LOW);
}
```

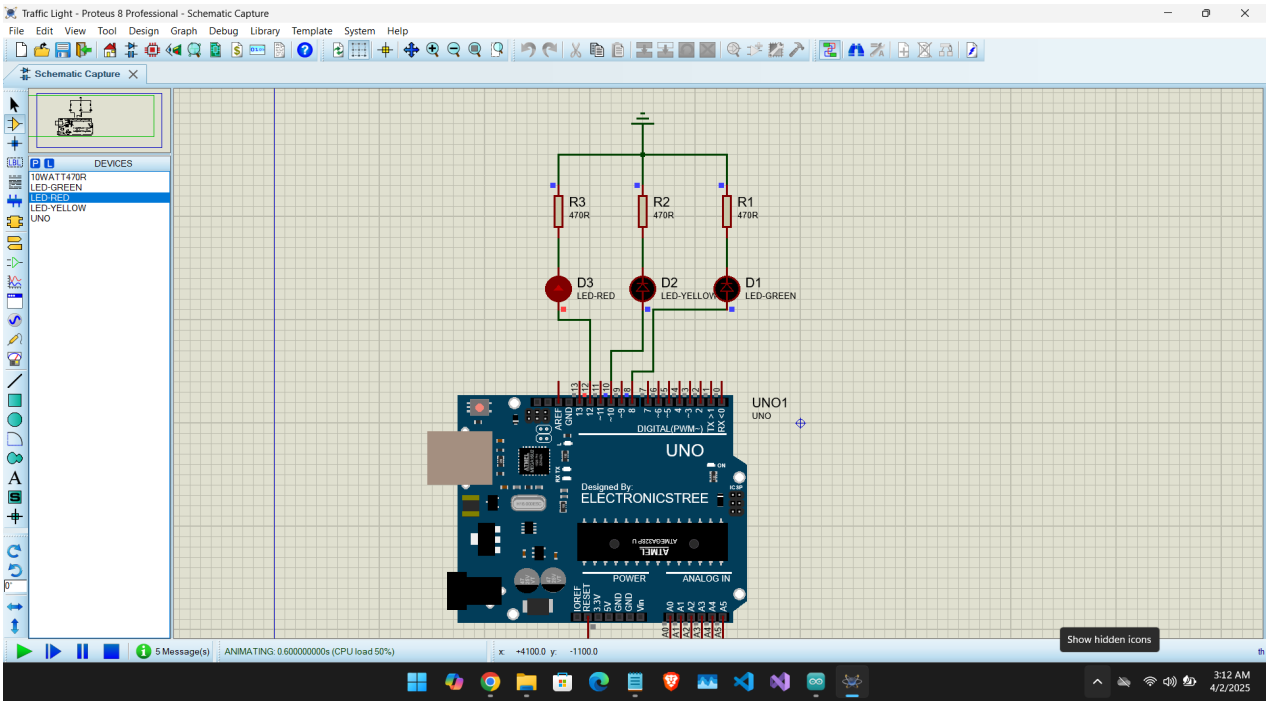

Figure 8: (A) Blinking RED for 3 seconds.

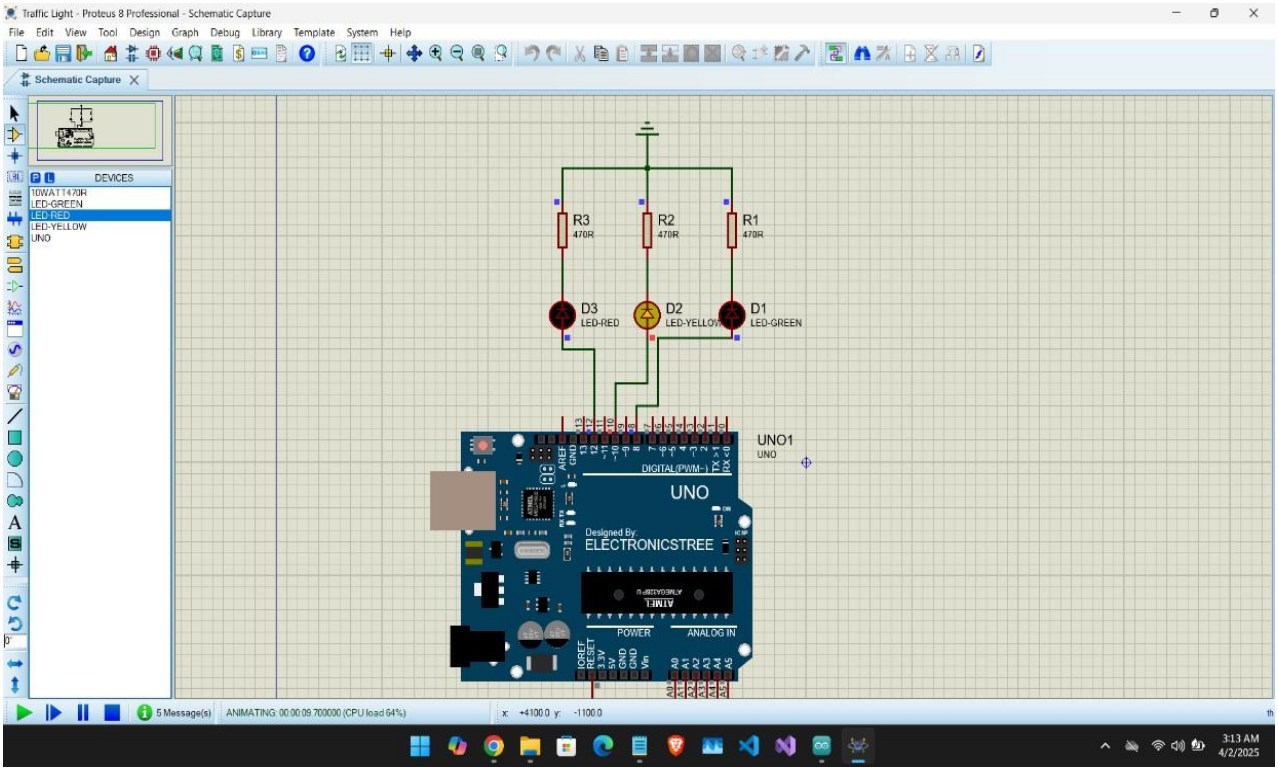

Figure 8: (B) Blinking Yellow for 9 seconds.

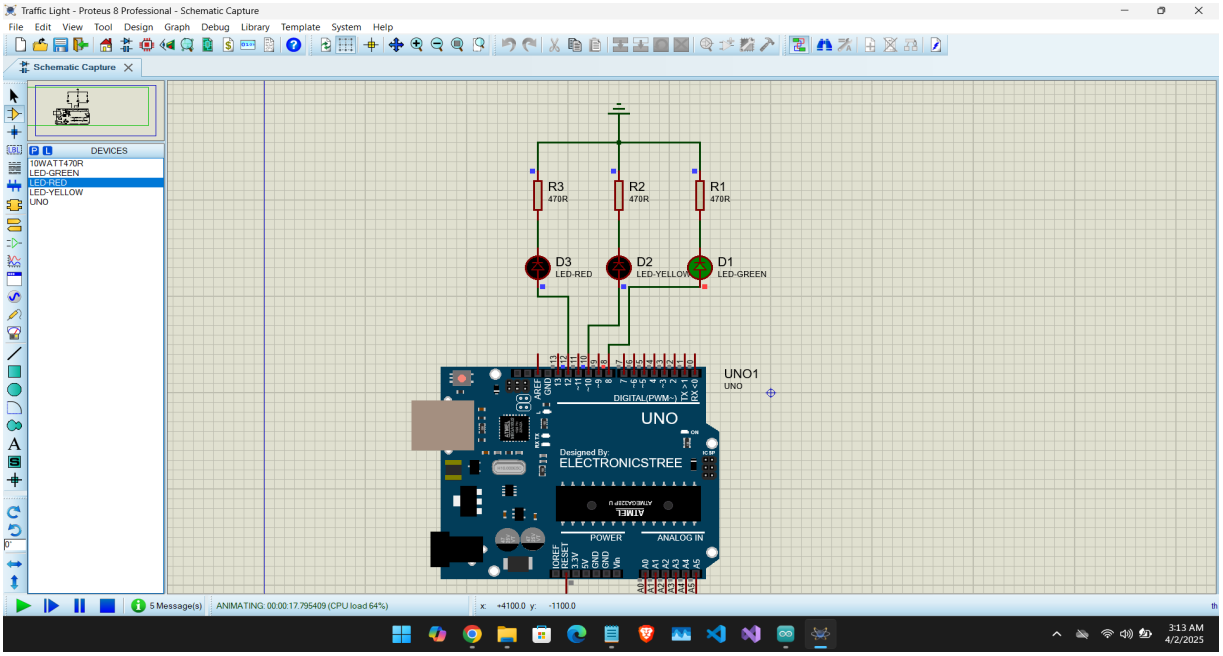

Figure 8: (C) Blinking Green for 3 seconds.

Discussion:

In the lab, both the blinking test and traffic light system experiments were performed. For the blinking test, the components were first set up on an actual Arduino board, with an LED connected to the designated pin. Using Arduino IDE, code was written to control the LED's blinking intervals, which was then compiled into a hex file. After uploading the hex file to the Arduino board, the LED blinked according to the programmed intervals. This hands-on setup verified both the circuit connections and the effectiveness of the code in a real environment. For the traffic light system, the experiment was more complex, requiring multiple LEDs to simulate a standard red-yellow-green traffic light sequence. Each LED was connected to specific pins on the Arduino, and code was written in Arduino IDE to establish the timing and sequence logic. Once compiled, the hex file was uploaded to the Arduino, and the LEDs transitioned between red, yellow, and green, mimicking a real traffic signal. Conducting this experiment in the lab provided practical insight into managing timing sequences and coordinating multiple components in a physical setting, highlighting how software logic translates into hardware functionality in embedded systems.

References:

[1] https://www.arduino.cc/.
[2] ATMega328 manual
[3] https://www.avrfreaks.net/forum/tut-c-newbies-guide-avr-timers
[4] http://maxembedded.com/2011/06/avr-timers-timer0/