American International University- Bangladesh

Department of Computer Science

Lab Report Cover Sheet

Course Name	MICROPROCESSOR AND EMBEDDED SYSTEMS
Lab Report No.	03
Lecturer Name	**MD. ALI NOOR**
Semester	SUMMER 2021-22
Submission Date	23/06/2022
Section	O
Group No.	03

Student Name	**Student ID**	**Contribution (out of 100%)**
1. EMON, MEDEDI HASAN	19-41118-2	
2. ASIF AHAMMED	19-40020-1	
3. EASINUL ABEDIN SAYEM	19-40291-1	
4. MD REDOY SHEIKH	19-40101-1	
5. MD. NAZIM HASAN	19-40118-1	
6. MOHI UDDIN ANANDO	20-43585-1	

Title: Timers: Implementation of a traffic control system.

Objective:

The goal of this exercise is to acquaint yourself with Timers and see how they can be used to construct a traffic management system.

Introduction:

Every electrical part of a sequential logic circuit operates on a time basis. The synchronization of all work is aided by this time base. If devices lacked a time base, they wouldn't know when to perform particular tasks. The timer is therefore a key concept in the field of electronics.

A timer/counter is built into the circuitry of the Arduino controller. It can be used to keep track of time events and works similarly to a clock. A timer is a register whose value changes automatically. 8-bit and 16-bit timers are the two different types of timers found in AVR. A 16-bit timer uses a register that is 16 bits wide, whereas an 8-bit timer uses a register that is 8 bits large.

Apparatus:

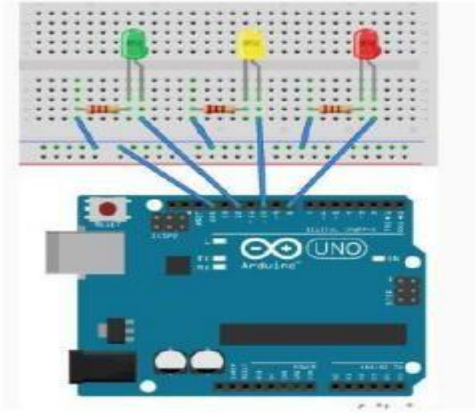
- Arduino Uno/ Arduino Mega
- LED lights (YELLOW, RED, and GREEN)
- Resistors (220 ohms)

Theory and Methodology:

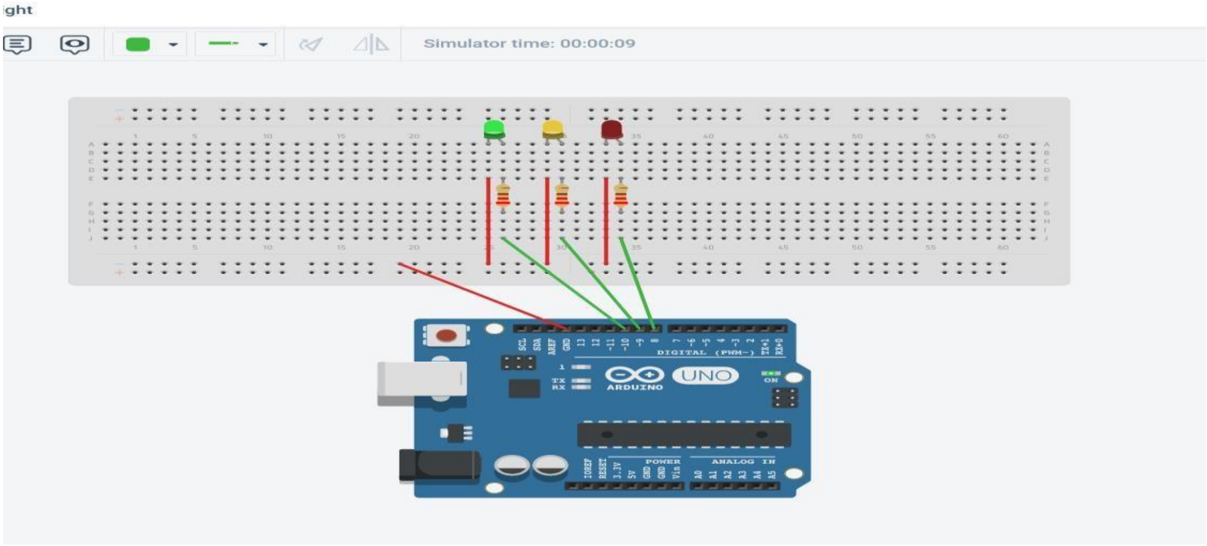
Timers are little gadgets that allow you to set the time between occurrences. The ability to accurately regulate the time between events is particularly frequent in embedded systems design. In multi-threaded operating systems, timers are used to determine how long a task is active before switching to another. To save electricity, timers can be used to pulse width modulate (PWM) an LED. Timers can also be used to calculate an analog signal's sampling rate. Timers are used in almost every embedded software project.

Experiment and Result:

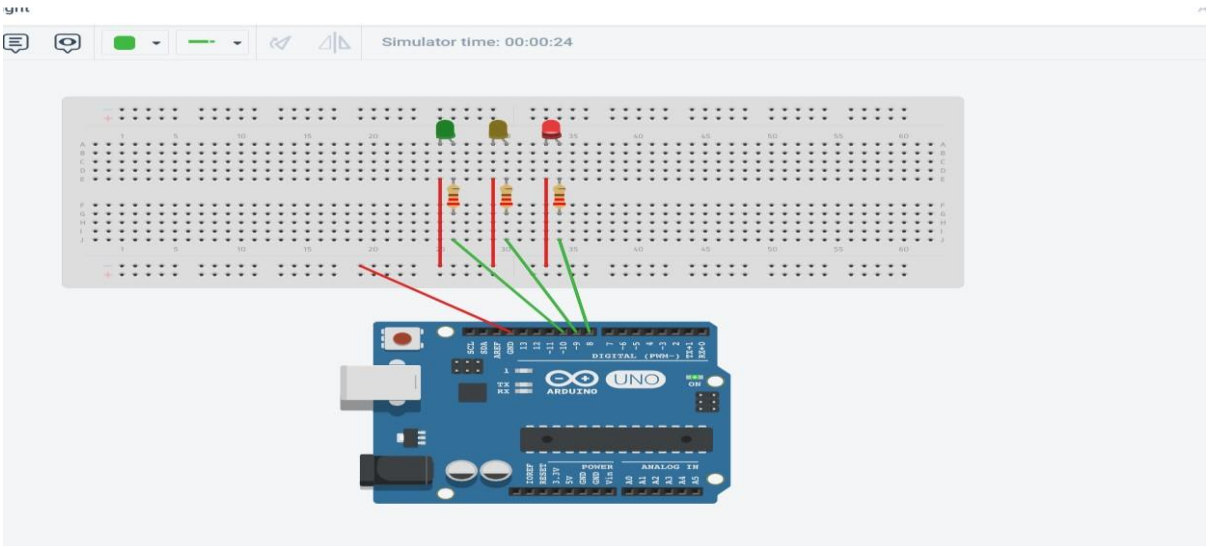
1.First, create the circuit by connecting all of the pieces using the following connection scheme.



2. Understand the normal operation of a traffic control system.



Simulator time: 00:00:09



Simulator time: 00:00:24

p.t.o

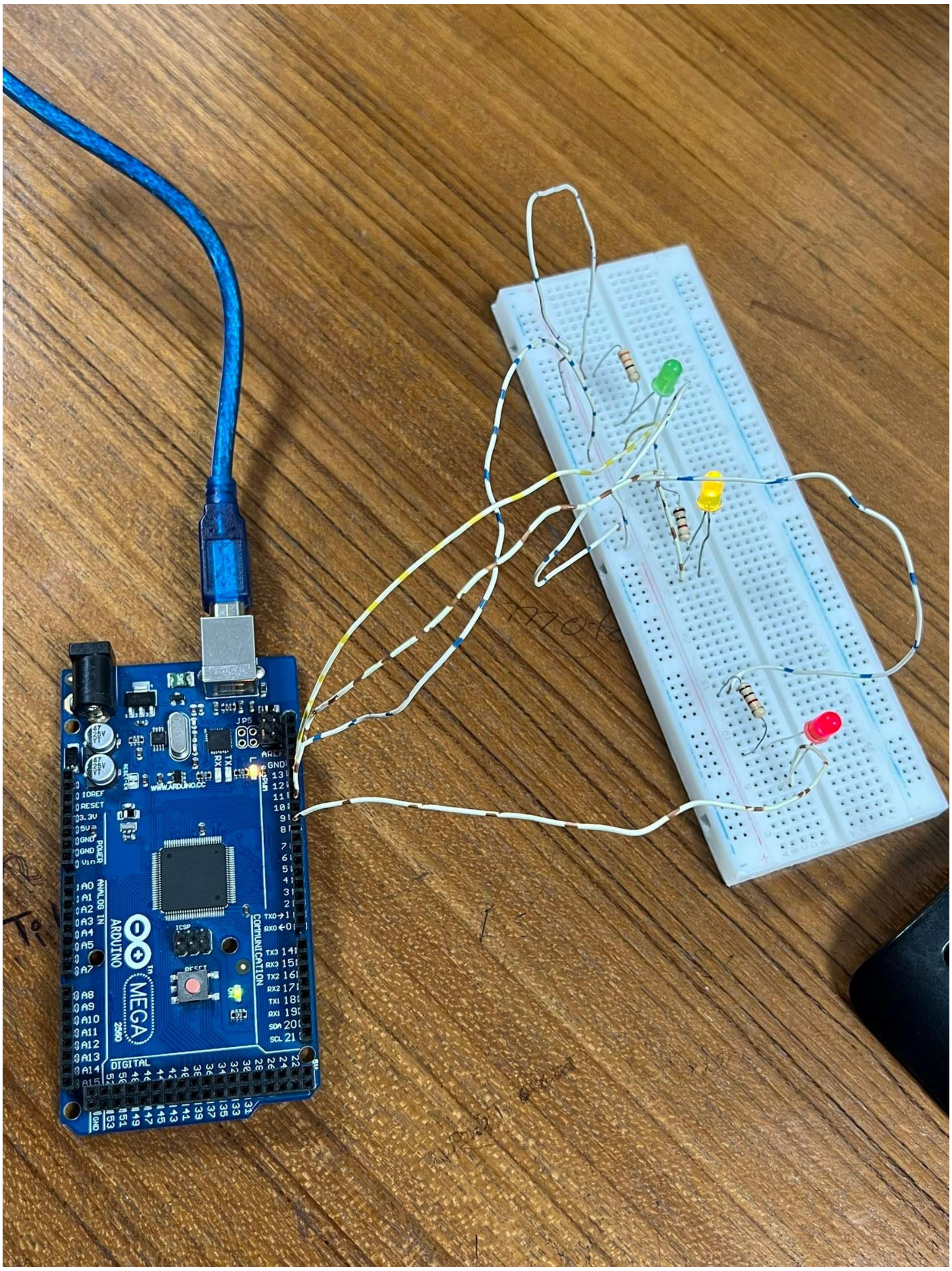


Figure: Circuit of traffic control system

```
}
```

<u>Code implementation:</u>

```
#define RED1_PIN 4
#define RED2_PIN 6
#define YELLOW1_PIN 8
#define YELLOW2_PIN 14
#define GREEN1_PIN 16
#define GREEN2_PIN 18
#define SWITCH_PIN 2

int LED_blink = 700;
int switch_read;
int LED_sequence=1;
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 and the switch as input
 pinMode (RED1_PIN, OUTPUT);
pinMode (RED2_PIN, OUTPUT);
pinMode (YELLOW1_PIN, OUTPUT);
pinMode (YELLOW2_PIN, OUTPUT);
pinMode (GREEN1_PIN, OUTPUT);
pinMode (GREEN2_PIN, OUTPUT);
pinMode (SWITCH_PIN, INPUT);

 TCCR0A = 0b00000000;
 TCCR0B = 0b00000101; //setting prescaler for timer clock
 TCNT0=0;
}
void loop() {
 switch_read=digitalRead(SWITCH_PIN);
if (switch_read==LOW){
 LED_sequence=!LED_sequence;
 }
 if (LED_sequence==1)
 {
 //to turn red1 LED blink
digitalWrite(RED1_PIN, HIGH);
delay_timer(LED_blink);
digitalWrite (RED1_PIN, LOW);

//to turn yellow1 LED blink
digitalWrite(YELLOW1_PIN, HIGH);
```

```
delay_timer(LED_blink);
digitalWrite (YELLOW1_PIN, LOW);

//to make green1 LED blink
digitalWrite(GREEN1_PIN, HIGH);
delay_timer(LED_blink);
digitalWrite(GREEN1_PIN, LOW);

 //red2 blink and so on
digitalWrite(RED2_PIN, HIGH);
delay_timer(LED_blink);
digitalWrite(RED2_PIN, LOW);

 //to turn yellow2 LED blink
 digitalWrite(YELLOW2_PIN, HIGH);
 delay_timer(LED_blink);
 digitalWrite (YELLOW2_PIN, LOW);

//green2 blink and so on
digitalWrite(GREEN2_PIN, HIGH);
delay_timer(LED_blink);
digitalWrite(GREEN2_PIN, LOW);
 }
 else {
digitalWrite(RED2_PIN, HIGH);
delay_timer(LED_blink);
digitalWrite(RED2_PIN, LOW);
digitalWrite(YELLOW2_PIN, HIGH);
delay_timer(LED_blink);
digitalWrite(YELLOW2_PIN, LOW);

digitalWrite(GREEN2_PIN, HIGH);
delay_timer(LED_blink);
digitalWrite(GREEN2_PIN, LOW);

digitalWrite(RED1_PIN, HIGH);
delay_timer(LED_blink);
digitalWrite (RED1_PIN, LOW);
delay_timer(LED_blink);

digitalWrite(YELLOW1_PIN, HIGH);
delay_timer(LED_blink);
digitalWrite(YELLOW1_PIN, LOW);
digitalWrite(GREEN1_PIN, HIGH);
delay_timer(LED_blink);
digitalWrite(GREEN1_PIN, LOW);
 }
}
```

Working Procedure:

- First, we learnt about the many components that make up the Arduino family. The Arduino UNO Microcontroller is one of them. Following that, we have a basic grasp of how Tinkercad software operates.
- To construct the circuit shown above, we must first launch the linkercad online application. Then we put it together by selecting from a selection of components.
  - Following that, we write the code for the traffic light system.
- The circuit is then assembled on a breadboard and connected to the hardware.

Report ques and ans:

1. Include all codes and scripts into the lab report following the writing template mentioned in appendix A of Laboratory Sheet Experiment 3.

   Answer: Attached above

Discussion:

This experiment was carried out in two ways. First, by using an Arduino board, three colored animated. LED lights (red, yellow, and green), three resistors breadboard and connecting wires, the traffic control system was created. After that, the LED lights (red, yellow, and green) were connected to ports 8, 10 and 12. Then some code was written for the traffic control system on Arduino IDE with timers. Then the Arduino board was connected to the computer and the code was run to get the traffic light result. Another way this experiment was done with the help of Tinkercad software. At the time of doing this experiment some issues were faced. The issues were some pin configuration mistakes and some errors in code . But these issues were solved with the help of a lab manual and the internet.

Conclusion:

The goal of this project was to get me acquainted with microcontrollers. We successfully constructed the specified traffic control system as well as our assigned assignment of 'Temperature monitoring system.' We followed our teacher's directions to the letter for this experiment.

Reference(s):

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/