AMERICAN INTERNATIONAL UNIVERSITY–BANGLADESH (AIUB)

FACULTY OF SCIENCE & TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MICROPROCESSOR AND EMBEDDED SYSTEMS

Spring 2024-2025

Section: Q, **Group: 08**

LAB REPORT ON

Implementation of traffic runway lights using timer functions.

Supervised By

Protik Parvez Sheikh

Submitted By

MD. TANJIL TASHRIK ZIM

Group Member

No.	Name	ID
1.	**MD. TANJIL TASHRIK ZIM**	**22-48021-2**
2.	MD. AL-IMRAN SAYEM	22-48023-2
3.	MD. ABRAR RAFID SHARIAR	22-48055-2
4.	HUMAIRA ZANNAT	22-47789-2
5.	SANJUKTA DAS	22-47801-2

Date of Submission: March 18, 2025

Title: Implementation of traffic runway lights using timer functions.

Objective: Traffic control systems are vital components of modern urban infrastructure, ensuring the orderly and efficient movement of vehicles and pedestrians. The increasing complexity and volume of traffic in cities necessitate the use of advanced technologies for effective traffic management. One such technology involves the use of microprocessors and timers to automate and optimize traffic light operations. In this lab experiment, the implementation of a traffic control system using an Arduino Mega, a popular microcontroller platform known for its versatility and ease of use has been explored. The Arduino Mega is equipped with a variety of features, including multiple timers, which can be harnessed to create precise and reliable timing sequences essential for traffic light control.

So, it can be said that the main objectives of this experiment are to:

1. Get familiarized with the implementations of Arduino UNO/Mega microcontroller.
2. Get familiarized with Timer.
3. Implement the functionality of Timer0 through programming.
4. Implement a basic Traffic Light Control System.

Apparatus:

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

Experimental Setup:

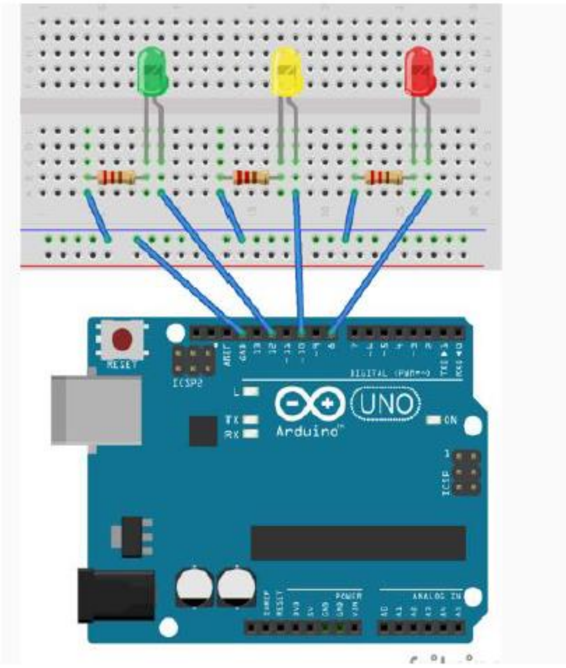
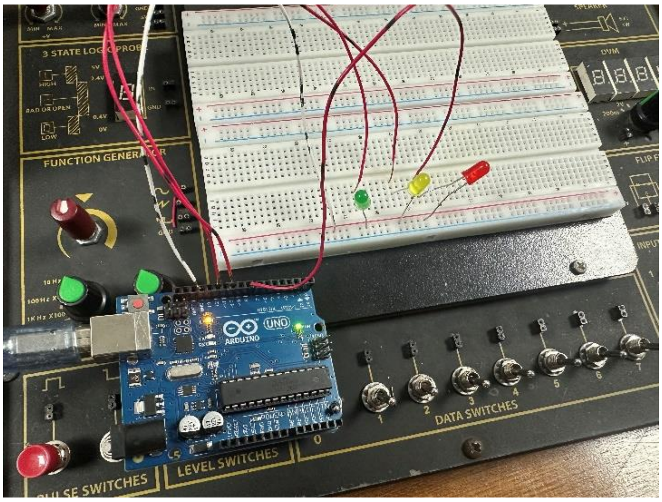


Fig 1: Experimental Setup a TLC using Arduino Mega (Timer0)



Code:

Code for Traffic Control System

```
#define RED_PIN 8  //define name of pins used
#define YELLOW_PIN 10
#define GREEN_PIN 12

//define the delays for each traffic light color
int red_on = 3000; //3s delay
int red_yellow_on = 1000; //1s delay
int green_on = 3000; //3s delay
int green_blink = 500; //.5s delay
int yellow_on = 1000; //1s delay
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() {
  // put your setup code here, to run once:
  pinMode(RED_PIN, OUTPUT);
  pinMode(YELLOW_PIN, OUTPUT);
  pinMode(GREEN_PIN, OUTPUT);
  TCCR0A = 0b00000000;
  TCCR0B = 0b00000101; //setting pre-scaler for timer clock
  TCNT0 = 0;
}

void loop() {
  // put your main code here, to run repeatedly:
  //to make red LED on
    digitalWrite(RED_PIN, HIGH);
    delay_timer(red_on);

    //to turn yellow LED on
    digitalWrite(YELLOW_PIN, HIGH);
    delay_timer(red_yellow_on);

    //turning off RED_PIN and YELLOW_PIN, and turning on greenLED
    digitalWrite(RED_PIN, LOW);
    digitalWrite(YELLOW_PIN, LOW);
    digitalWrite(GREEN_PIN, HIGH);
    delay_timer(green_on);
    digitalWrite(GREEN_PIN, LOW);

    //for turning green Led on and off for 3 times
```

```
   for(int i = 0; i < 3; i = i+1)
   {
   delay_timer(green_blink);
   digitalWrite(GREEN_PIN, HIGH);
   delay_timer(green_blink);
   digitalWrite(GREEN_PIN, LOW);
   }

   //for turning on yellow LED
   digitalWrite(YELLOW_PIN, HIGH);
   delay_timer(yellow_on);
   digitalWrite(YELLOW_PIN, LOW);
}
```

Experimental Output Results:

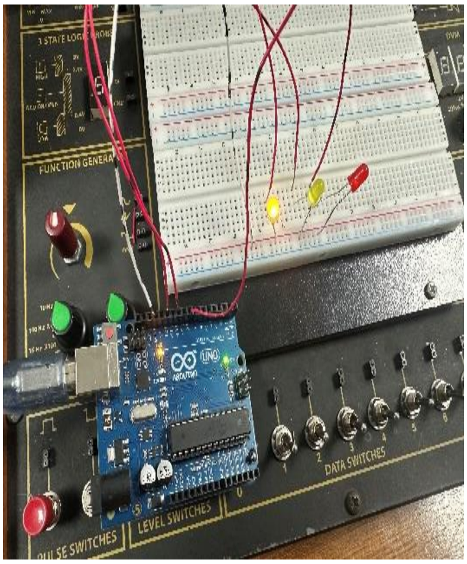


Fig 2: Green On

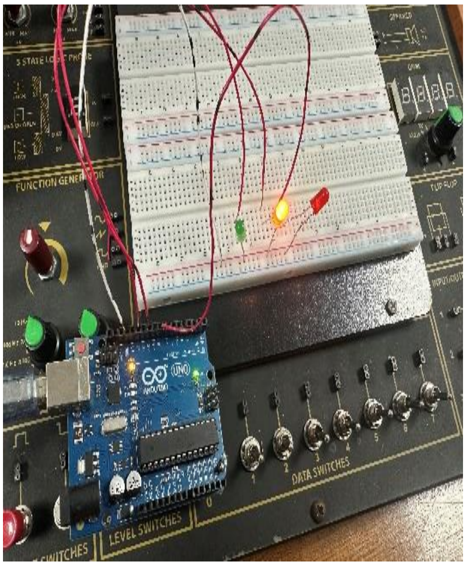


Fig 3: Yellow On

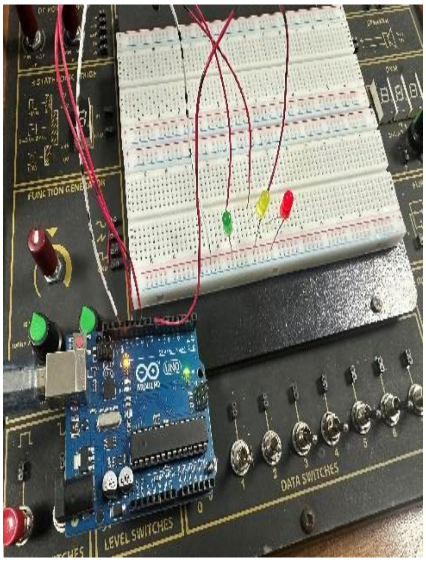


Fig 4: Red on

Simulation Output Results:

Here are the simulation output results of Traffic Light System on Tinkercad simulation and explanation:

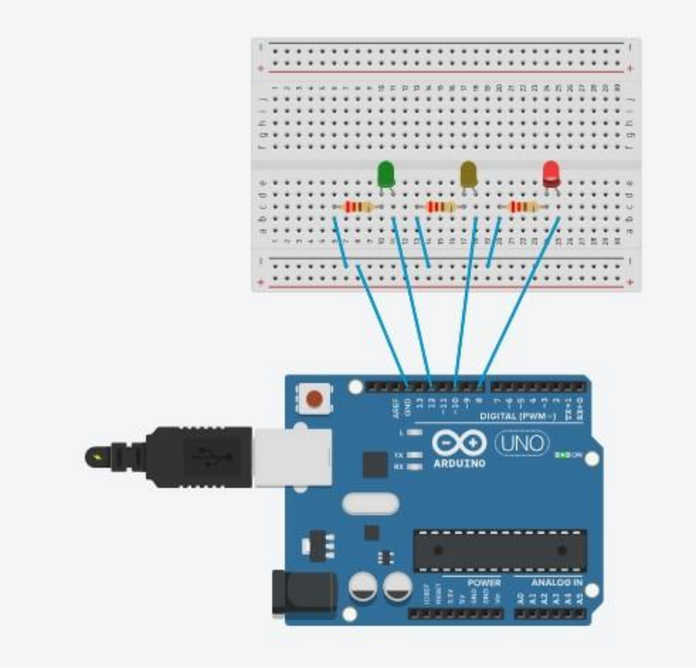


Fig 5: Red LED is ON in Traffic Light System on Tinkercad Simulation

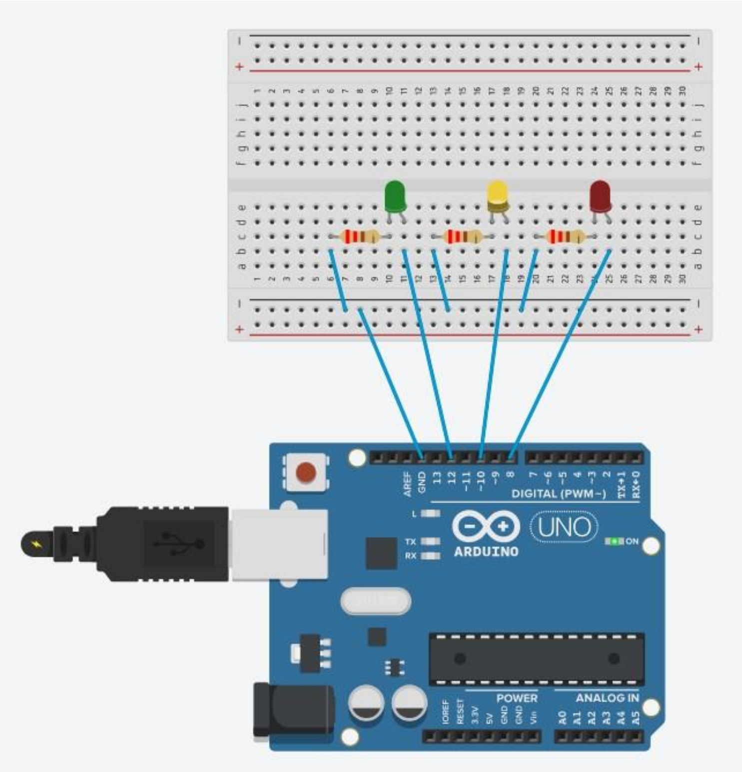


Fig 6: Yellow LED is ON in Traffic Light System on Tinkercad Simulation

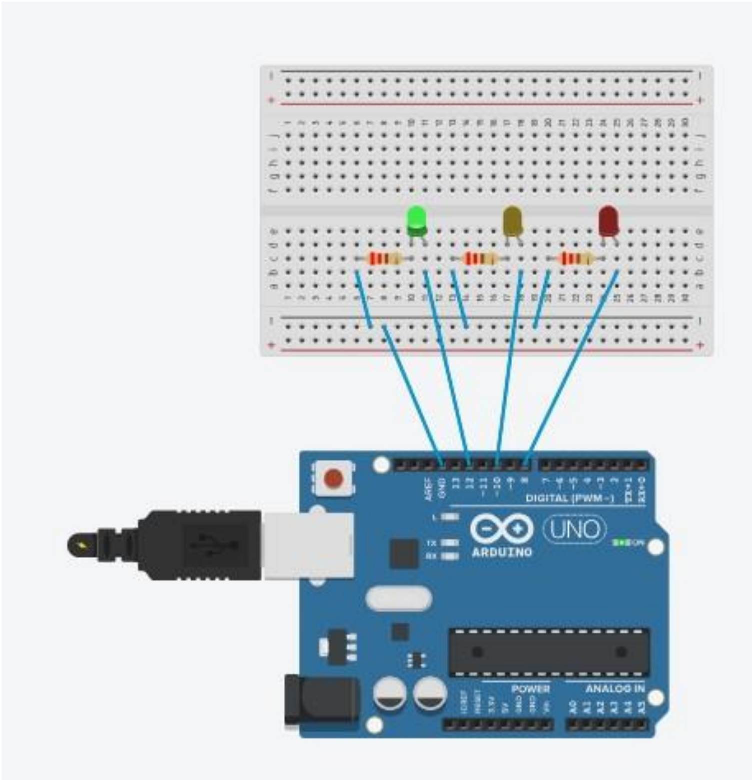


Fig 7: Green LED is ON in Traffic Light System on Tinkercad Simulation

Explanation:

The traffic light simulation implemented on Tinkercad. When the code is implemented and the loop () function starts, the code implements a digital write operation on the microcontroller and provides a high voltage at the pin 8 as output. As a result, the red light turns ON (Fig. 5). Then the light stayed on for 3000 milliseconds due to the delay function called 'red_on'. After that, the light turned off as a digital write was performed on the microcontroller and output at 8 was low. Then the yellow LED turns ON (Fig. 6) and stayed on for 1000 milliseconds using the 'yellow_on' delay function. After that the yellow LED light turned off. Then, green LED light turns ON for 3000 milliseconds using the 'green_on' delay function. After that a for loop was introduced in the code which made the microcontroller blink the green light 3 times with an interval of 500 milliseconds (Fig. 7) for the delay caused by the value set on 'green_blink'. When the green LED blinked for 3 times, then the yellow LED turns ON (Fig.3). It was turned on after 1000 milliseconds which was set in the 'yellow_on' variable and was used a delay function. After that the red LED turns ON (Fig. 5) and turned off after 3000 millisecond which was set in the 'red_on' variable and was used a delay function. All the operations kept occurring as all of the codes were performed in a loop.

Report Questions:

Question: Configure the system to have delays for outputs according to your ID. Consider the last three digits from your ID (if your ID is XX-XXABC-X then consider A for the RED light, B for the YELLOW LED, and C for GREEN LED). Include the program and results within your lab report.

Answer:

Using my ID, 22-48021-2, the delays for each LED, based on the last three digits, would be:

- **RED LED**: 5 seconds (for digit 0, replaced by 5)
- **YELLOW LED**: 2 seconds (for digit 2)

- **GREEN LED**: 1 second (for digit 1)

Configured Program:

```
// Pin numbers are defined by the corresponding LED colors
#define RED_PIN 8
#define YELLOW_PIN 10
#define GREEN_PIN 12

// Define delays for each traffic light color based on the last three digits of your ID
int red_on = 5000;      // 5 s delay for RED LED
int yellow_blink = 2000; // 2 s delay for YELLOW LED
int green_on = 1000;    // 1 s delay for GREEN LED

// Timer function to count delay without using delay()
int delay_timer(int milliseconds) {
  int count = 0;
  while (1) {
    if (TCNT0 >= 16) {  // 1 ms has passed
      TCNT0 = 0;
      count++;
      if (count == milliseconds) { // Check if delay time is reached
        count = 0;
        break; // exits the loop
      }
    }
  }
  return 0;
}

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

```
    // Setup Timer0
    TCCR0A = 0b00000000;
    TCCR0B = 0b00000101; // Set prescaler
    TCNT0 = 0;
}

void loop() {
    // Green LED sequence
    digitalWrite(GREEN_PIN, HIGH);
    delay_timer(green_on);
    digitalWrite(GREEN_PIN, LOW);

    // Yellow LED blink sequence
    for (int i = 0; i < 4; i++) {
        digitalWrite(YELLOW_PIN, HIGH);
        delay_timer(yellow_blink);
        digitalWrite(YELLOW_PIN, LOW);
        delay_timer(yellow_blink);
    }

    // Red LED sequence
    digitalWrite(RED_PIN, HIGH);
    delay_timer(red_on);
    digitalWrite(RED_PIN, LOW);
}
```

Discussion:

The experiment aimed to design simple and complex systems using the Timer0 function, which is a hardware component found on Arduino microcontroller boards. The delay() function was used to pause programs, but it had drawbacks because it paused all operations, which could disrupt necessary operations. Timers were proposed as a solution to this problem. Timer0 was found in the LED blink test and Traffic Control System, and its operation was observed and documented. The methodology for writing code using this function was also observed and incorporated into the systems developed. This experiment established a solid foundation for effectively using timers in future Arduino projects, allowing for precise timing control and event synchronization. The experiment's objectives were observed and met, providing a solid foundation for future Arduino projects. The experiment aimed to control timing, synchronize events, and develop advanced applications that required precise timing control, demonstrating Timer0's effectiveness in Arduino projects.

Conclusion:

We successfully fulfilled the given objectives by experimenting with the development of a traffic control system utilizing timers. We were acquainted with the Arduino microcontroller and received useful knowledge about timer functions in the Arduino environment. We were successful in creating a basic circuit to make an LED light blink by using the Timer0 function, illustrating a fundamental use of timers. The experiment's highlight was the deployment of a traffic control system, which demonstrated how timers may be used to govern timing intervals in real-world circumstances. Overall, the experiment taught us valuable skills in Arduino programming and timer-based control system design, creating a solid platform for future research in embedded systems and microcontroller applications.

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/