AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

Faculty of Engineering

Lab Report

Experiment # 01

Experiment Title: Familiarization with a microcontroller, the study of blink test and implementation of a traffic control system using microcontrollers.

Date of Perform:	3 March 2025	**Date of Submission:**	10 March 2025
Course Title:	Microprocessor and Embedded Systems Lab		
Course Code:	EE4103	**Section:**	P
Semester:	Spring 2024-25	**Degree Program:**	BSc in CSE
Course Teacher:	**Prof. Dr. Engr. Muhibul Haque Bhuyan**		

Group # 01

Sl No	Name	ID	PROGRAM	SIGNATURE
1.	Md.Saikat Hossain	23-51242-1	BSc in CSE	
2.	Md.Mosharof Hossain Khan	23-51259-1	BSc in CSE	
3.	Rimal Banik	23-51260-1	BSc in CSE	
4.	Md.Rahidul Islam	23-51269-1	BSc in CSE	
5.	Rahat Ahmed	21-44911-2	BSc in CSE	

Faculty use only

FACULTY COMMENTS	**Marks Obtained**	
	Total Marks	

Contents

Objectives:

The objective of this experiment is to get familiarized with Microcontrollers. Besides, the specific objectives of this experiment are to-

a) Make an LED blink using an Arduino and its delay function.
b) Implement a traffic control system using Arduino and its delay function and LEDs.

Apparatus:

1. Arduino IDE 2.3.5
2. Arduino Uno (R3) board
3. LED lights (RED, GREEN, and YELLOW)
4. Three 100 ohms resistors, and
5. Jumper wires

Circuit Diagram:

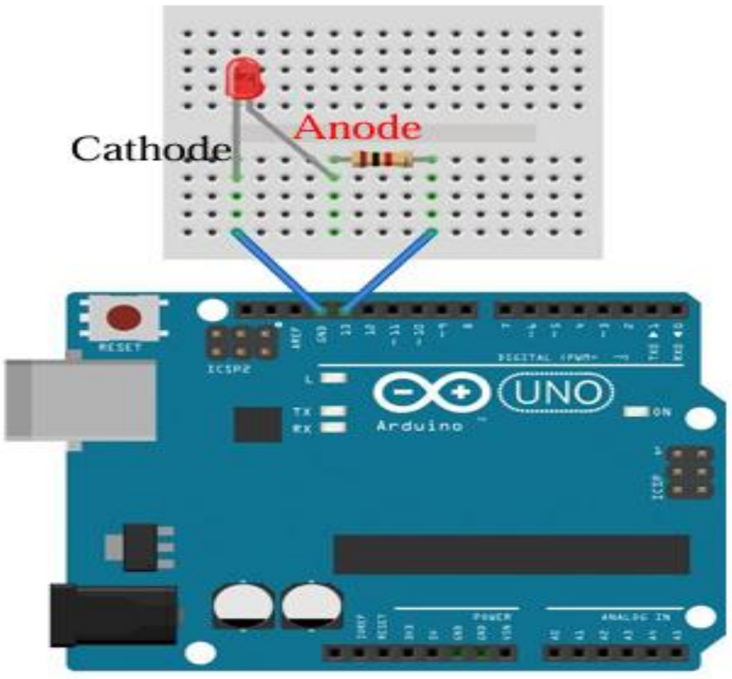

Figure 01: Experimental Setup of Blink Test using an Arduino Microcontroller Board.

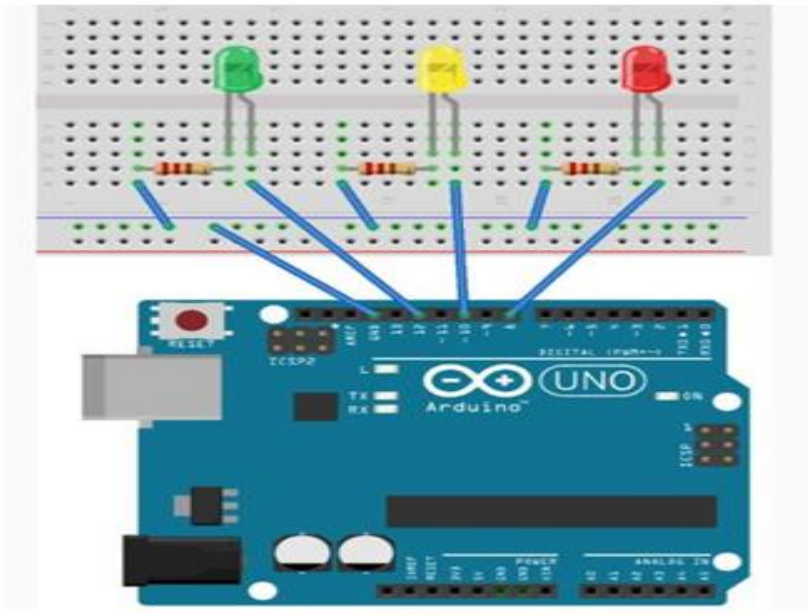

Figure 02: Experimental Setup of a Traffic Control System using an Arduino Microcontroller Board.

Code Explanation:

1.Program for LED blink test:

```
void setup() {
  // Pin type declaration for the red LED
pinMode(5, OUTPUT);
}



void loop() {
  // Turning on the voltage at output pin 5 (for red LED)

digitalWrite(5, HIGH);
delay(1000);    //  LED is turned on for 1 second
digitalWrite(5, LOW);
delay(1000);    //  LED is turned off for 1 second
}
```

Explanation: This program makes a red LED connected to pin 5 of an Arduino board blink on and off. In the setup() function, pin 5 is set as an output to control the LED. In the loop() function, the LED is turned ON for 1 second by setting pin 5 to HIGH, then turned OFF for 1 second by setting pin 5 to LOW. This cycle repeats indefinitely, causing the LED to blink with a 1-second interval. The delay(1000) function is used to control the timing of the LED's ON and OFF states.

2. Program for Traffic Light Control System:

```
void setup() {
  // pin connections for the LED lights

pinMode(8, OUTPUT);
pinMode(10, OUTPUT);
pinMode(12, OUTPUT);
}

void loop() {
// Turning on the voltage at the output pin 8 (for green LED)
digitalWrite(8, HIGH);
delay(3000);     // green LED is on for 3 seconds
// Turning off the voltage at output pin 8 (for green LED)
digitalWrite(8, LOW);   // green LED is off

// Turning the yellow LED on and off for 4 times
for (int i = 0; i < 4; i = i+1)
{
digitalWrite(10, HIGH);
delay(500);    // yellow LED is on for 0.5 seconds
```

```
digitalWrite(10, LOW);
delay(500);    // yellow LED is off for 0.5 seconds
}

// Turning on the voltage at the output pin 12 (for red LED)
digitalWrite(12, HIGH);
delay(6000);       // red LED is on for 6 seconds
// Turning off the voltage at the output pin 12 (for red LED)
digitalWrite(12, LOW);    // red LED is off

} // void loop() function ends and repeats
```

Explanation: This program simulates a traffic light control system using an Arduino. It controls three LEDs (Green, Yellow, and Red) connected to pins 8, 10, and 12. In the loop() function, the Green LED stays ON for 3 seconds, the Yellow LED blinks 4 times with a 0.5-second delay, and the Red LED stays ON for 6 seconds. The cycle repeats continuously, simulating a real traffic light sequence. The pinMode() function in the setup() phase configures the pins as outputs, and the digitalWrite() and delay() functions control the LEDs' states and timing.

Hardware Implementation and Explanation:

1.LED blink test:

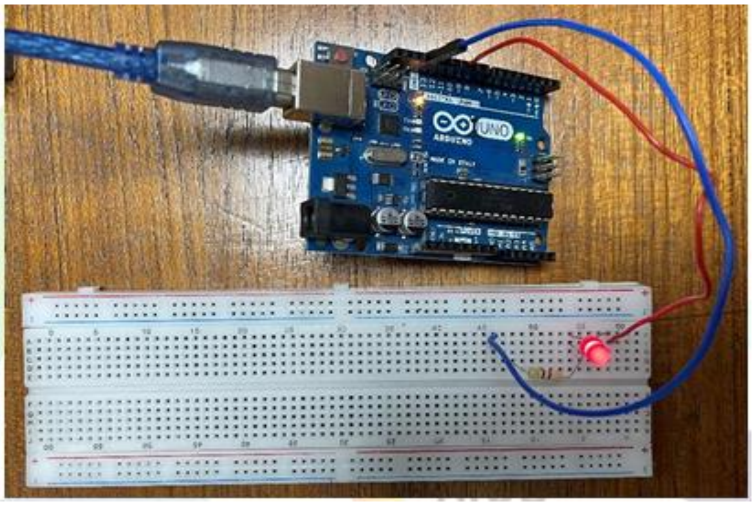

Figure 03: Hardware implementation for LED blink test

Explanation: In this implementation, a jumper wire was connected at the pin 5 of the Arduino Uno board. The wire was then connected on the breadboard. The cathode of an LED light was connected with the wire connected with pin 5. The anode of the LED light was connected with a 100 Ω resistor. The following resistor was then connected with the Ground (GND) of the Arduino Uno board. The Arduino Uno board was connected with a computer to compile and import required code.

2.Traffic light control system:

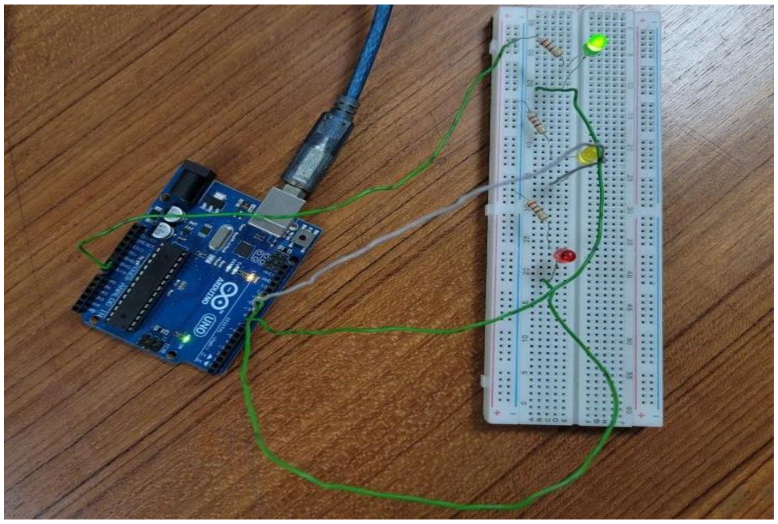

Figure 04: Hardware implementation for traffic light control system

Explanation: In this experiment, jumper wires were connected from pin 8, 10 and 12 of the Arduino Uno board to the anodes of the GREEN, YELLOW and RED LED sequentially. Three 100 Ω resistors were connected at each cathode of all the LED.The following resistor was then connected with the Ground of the Arduino board. The Arduino board was connected with a computer to compile and import required code.

Experimental Output Results:

1. LED Blink Test:

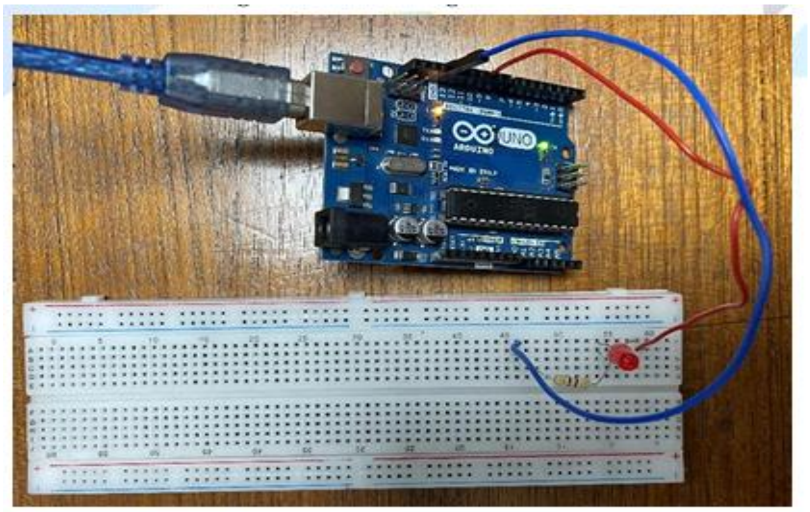

Figure 05: Light blink test (When LED is off)

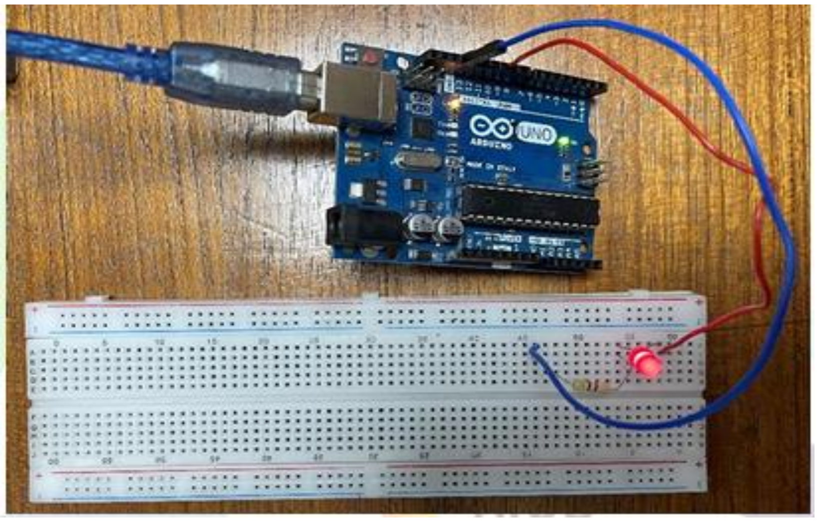

Figure 06: Light blink test (When LED is on)

Explanation: When the code is implemented and the loop() function starts, the code implements a digitalWrite() operation on the microcontroller and provides a high voltage at the pin 5 as output. Hence, the LED that was connected at pin 5 turns ON (Figure 06). A delay occurs afterwards for a certain time. This time was defined in milliseconds. According to the code that was implemented, a 1000 millisecond delay occurs after the light is turned ON. Then, another digitalWrite() operation occurs and the pin 5 was set to LOW as output. Therefore, the LED turns OFF (Figure 05). Another 1000 milliseconds delay occurs after the LED was set to low. The delay() operation was executed by using delay() function in the loop () function. Thus, these results were happening until the microcontroller was turned off or removed from the power source/computer.

2.Traffic Light Control System:

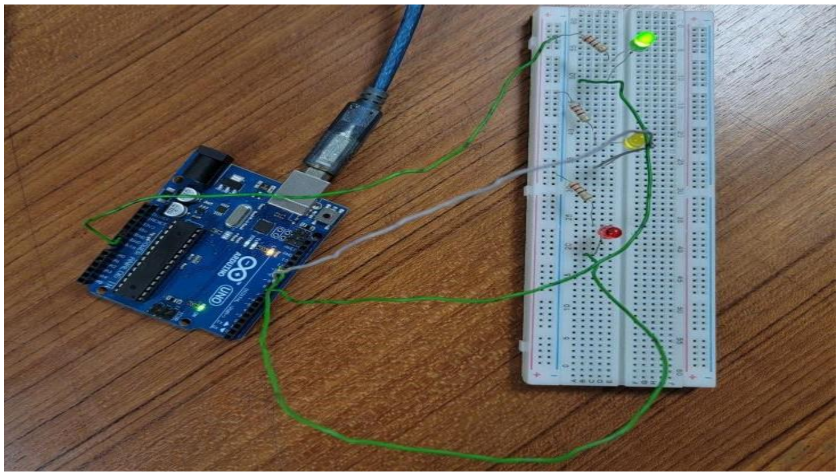
Figure 07: Traffic light control system(When GREEN LED is on)

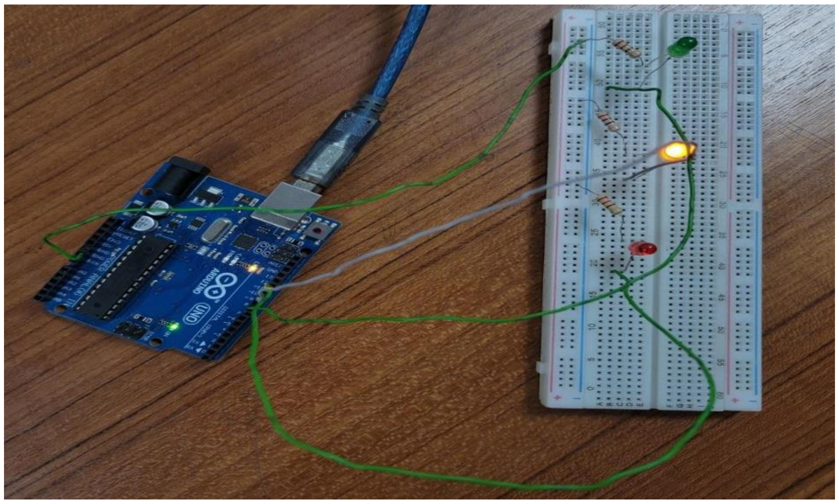
Figure 08: Traffic light control system(When YELLOW LED is on)

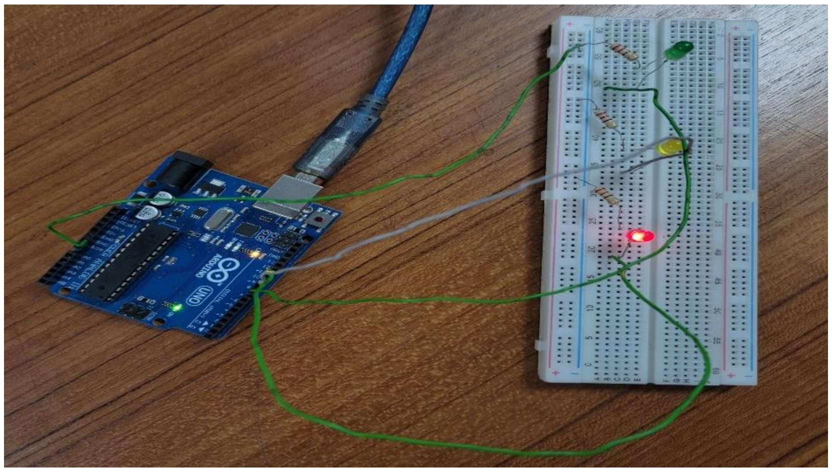
Figure 09: Traffic light control system(When RED LED is on)

Explanation: When the code is implemented and the loop() function starts, the code implements a digitalWrite() operation on the microcontroller and provides a high voltage at pin 8 as output. As a result, the green light turns ON (Figure 07). Then, the light stayed on for 3000 milliseconds due to the delay() function called green_on.After that,the light turned off as a digitalWrite() was performed on the microcontroller and output at pin 8 was LOW.Then, a digitalWrite() operation on the microcontroller provides a high voltage at pin 10 as output. As a result, the yellow light turns ON and OFF four times due to the for loop introduced in the code (Figure 08). Each blink lasted 500 milliseconds because of the delay() function called yellow_blink.After the yellow LED blinked four times, a digitalWrite() operation on the microcontroller provides a high voltage at pin 12 as output. As a result, the red light turns ON (Figure 09). Then, the light stayed on for 6000 milliseconds due to the delay() function called red_on. After that, the light turned off as a digitalWrite() was performed on the microcontroller and output at pin 12 was LOW.All the operations kept occurring as all of the codes were performed in a loop.

Simulation Output Results:

1.LED Blink Test:

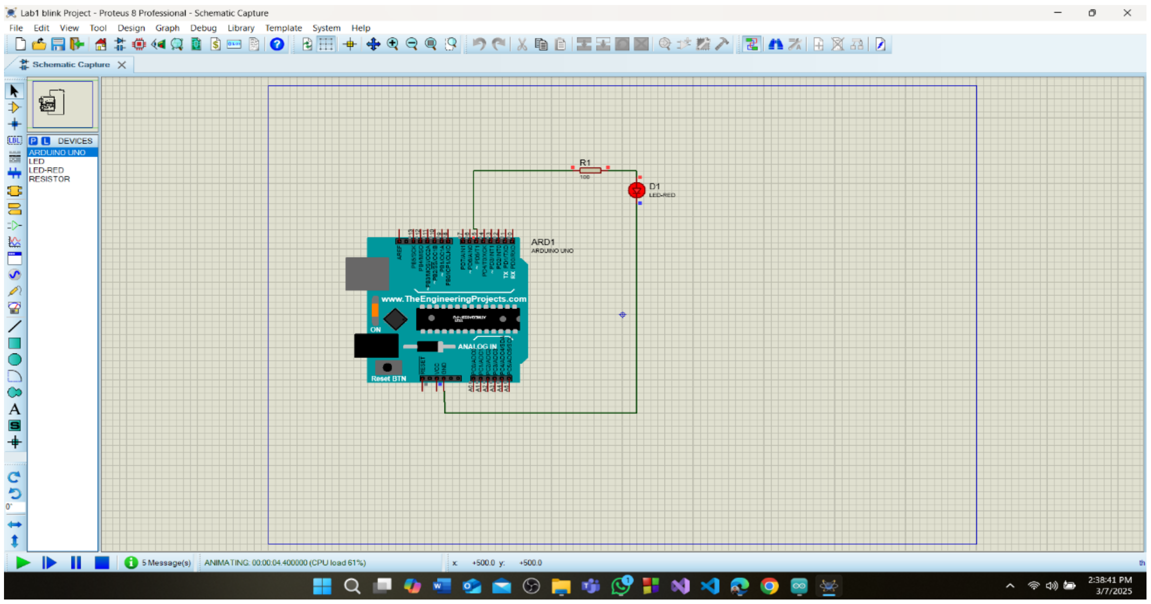
Lab1 blink Project - Proteus 8 Professional - Schematic Capture
File Edit View Tool Design Graph Debug Library Template System Help
Schematic Capture
DEVICES
ARDUINO UNO
LED
LED-RED
RESISTOR
R1
D1
LED-RED
ARD1
ARDUINO UNO
www.TheEngineeringProjects.com
ANALOG IN
Reset BTN
5 Message(s)
ANIMATING: 00:00:04.400000 (CPU load 61%)
2:38:41 PM
3/7/2025

Figure 10: LED Blink Test on Simulation(When LED is on)

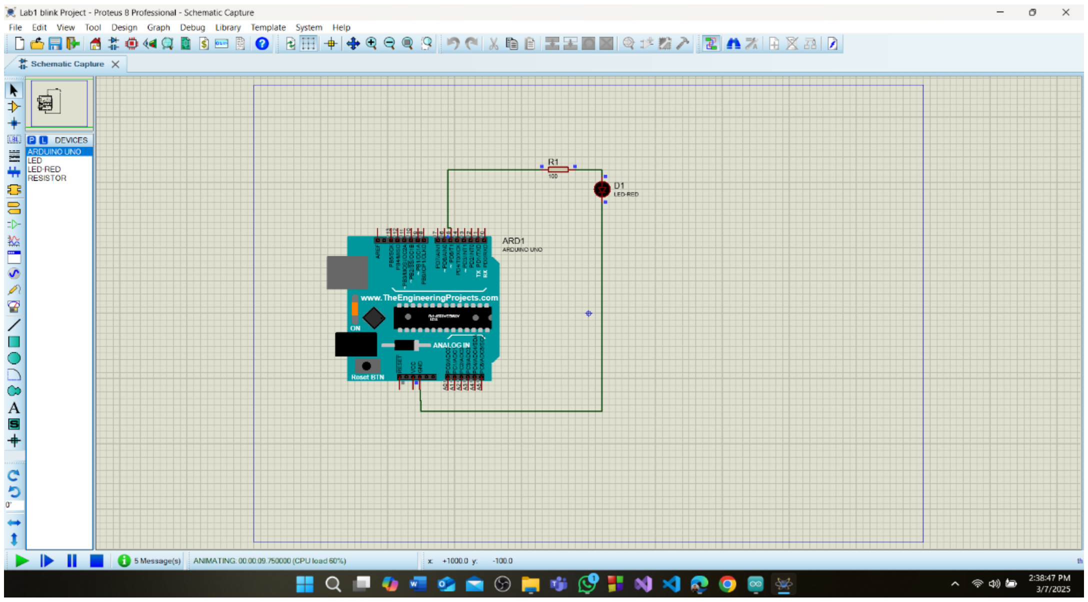
Lab1 blink Project - Proteus 8 Professional - Schematic Capture
File Edit View Tool Design Graph Debug Library Template System Help
Schematic Capture
DEVICES
ARDUINO UNO
LED
LED-RED
RESISTOR
R1
D1
LED-RED
ARD1
ARDUINO UNO
www.TheEngineeringProjects.com
ANALOG IN
Reset BTN
5 Message(s)
ANIMATING: 00:00:09.750000 (CPU load 60%)
2:38:47 PM
3/7/2025

Figure 11: LED Blink Test on Simulation(When LED is off)

Explanation: In this simulation, the Arduino Uno board, 100Ω resistor, and LED were set up according to the hardware configuration used in the lab. The program was written and verified in the Arduino IDE 2.3.5, which generated a HEX file. This file was then loaded into the Proteus simulation to simulate the behavior of the circuit. After running the simulation, the results were observed and compared with the actual hardware results to check for consistency and verify the performance of the system.

2.Traffic Light System Control:

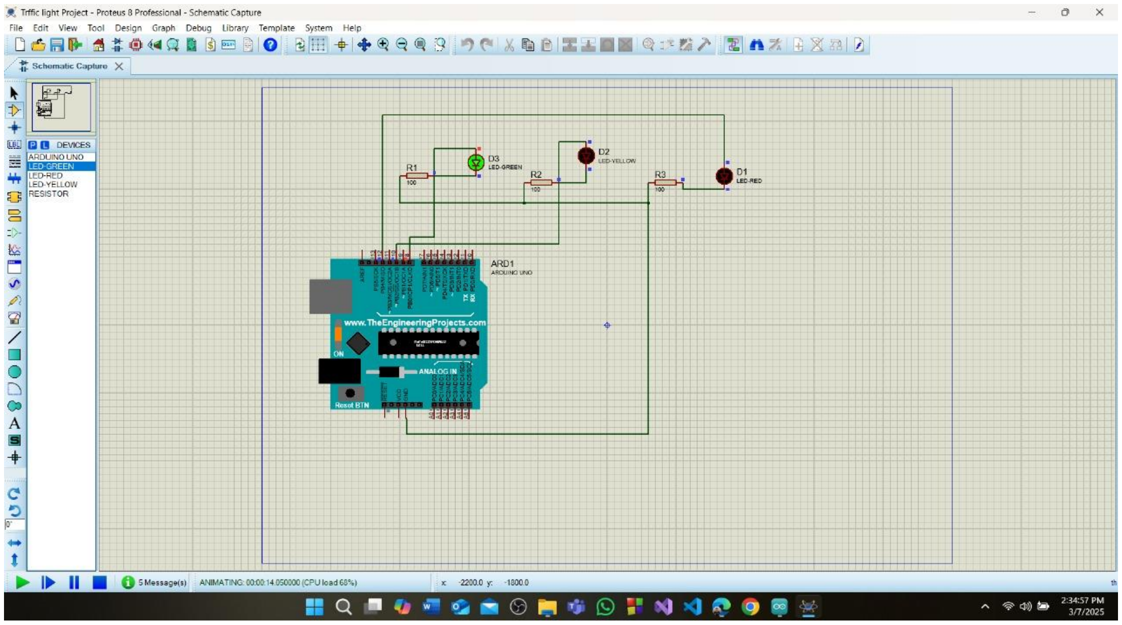
Trffic light Project - Proteus 8 Professional - Schematic Capture
File Edit View Tool Design Graph Debug Library Template System Help
Schematic Capture
DEVICES
ARDUINO UNO
LED-GREEN
LED-RED
LED-YELLOW
RESISTOR
R1
D3
LED-GREEN
R2
D2
LED-YELLOW
R3
D1
LED-RED
ARD1
ARDUINO UNO
www.TheEngineeringProjects.com
ANALOG IN
Reset BTN
5 Message(s)
ANIMATING: 00:00:14.050000 (CPU load 68%)
2:34:57 PM
3/7/2025

Figure 12: Traffic Light System Control on simulation(Green LED is on)

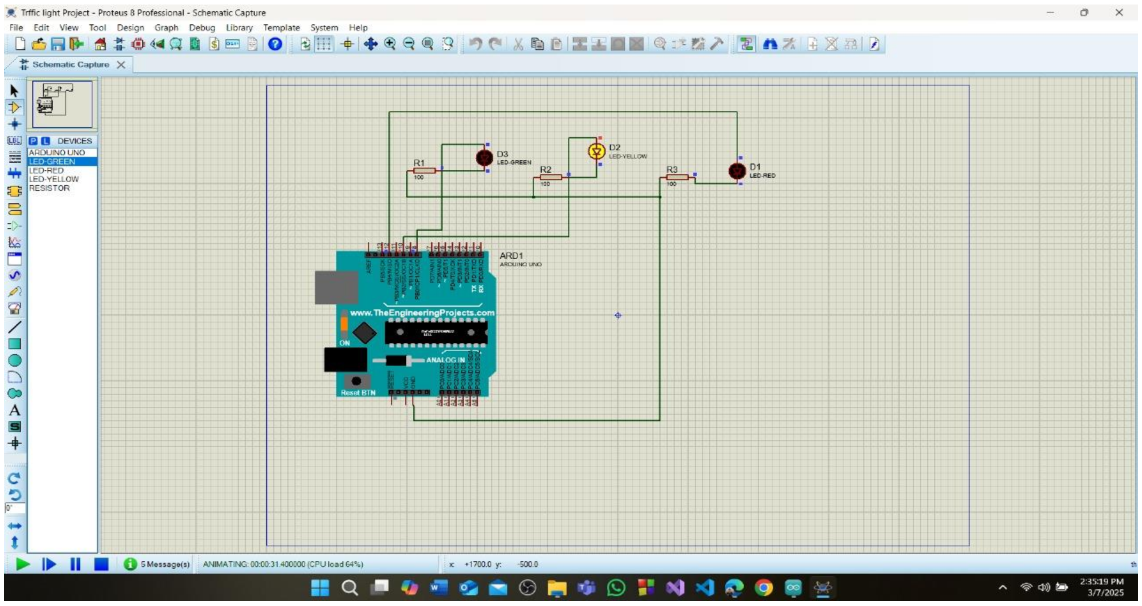
Trffic light Project - Proteus 8 Professional - Schematic Capture
File Edit View Tool Design Graph Debug Library Template System Help
Schematic Capture
DEVICES
ARDUINO UNO
LED-GREEN
LED-RED
LED-YELLOW
RESISTOR
R1
D3
LED-GREEN
R2
D2
LED-YELLOW
R3
D1
LED-RED
ARD1
ARDUINO UNO
www.TheEngineeringProjects.com
ANALOG IN
Reset BTN
5 Message(s)
ANIMATING: 00:00:31.400000 (CPU load 64%)

Figure 13: Traffic Light System Control on simulation(Yellow LED is on)

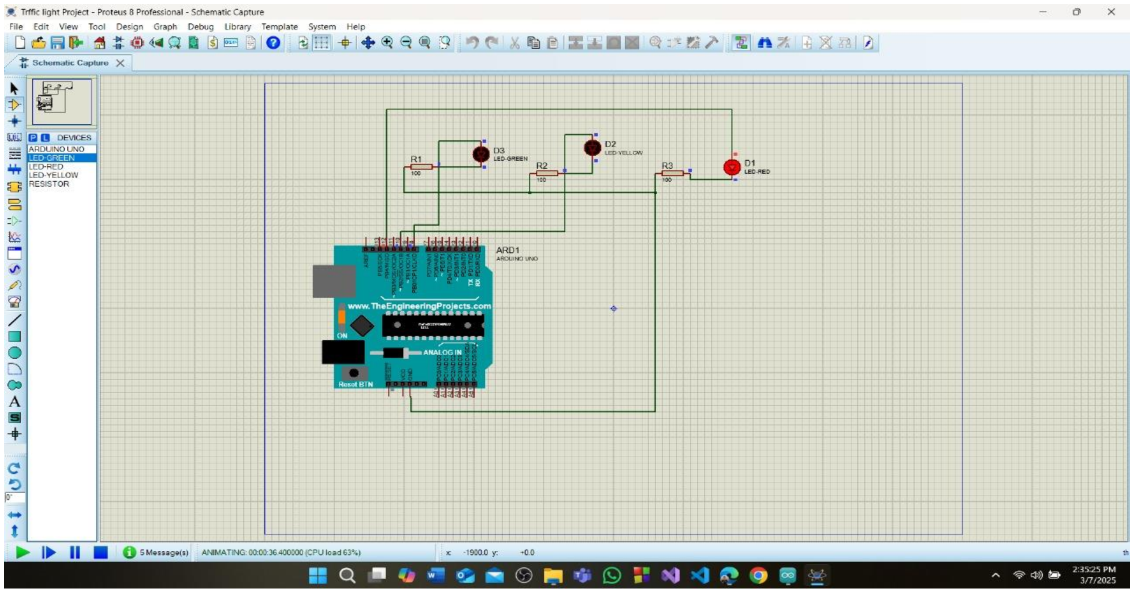
Trffic light Project - Proteus 8 Professional - Schematic Capture
File Edit View Tool Design Graph Debug Library Template System Help
Schematic Capture
DEVICES
ARDUINO UNO
LED-GREEN
LED-RED
LED-YELLOW
RESISTOR
R1
D3
LED-GREEN
R2
D2
LED-YELLOW
R3
D1
LED-RED
ARD1
ARDUINO UNO
www.TheEngineeringProjects.com
ANALOG IN
Reset BTN
5 Message(s)
ANIMATING: 00:00:36.400000 (CPU load 63%)

Figure 14: Traffic Light System Control on simulation(Red LED is on)

Explanation: In this simulation, the Arduino Uno board, three 100Ω resistors,Green, Yellow,Red LEDs were set up according to the hardware configuration used in the lab. The program was written and verified in the Arduino IDE 2.3.5, which generated a HEX file. This file was then loaded into the Proteus simulation to simulate the behavior of the circuit. After running the simulation, the results were observed and compared with the actual hardware results to check for consistency and verify the performance of the system.

Discussion:

In this experiment, we implemented two microcontroller-based applications: a basic LED blink test and a traffic light control system using an Arduino Uno. The blink test helped in understanding the fundamental concepts of digital output control using microcontrollers, where an LED was programmed to turn ON and OFF at a set interval using digitalWrite() and delay(). This simple experiment demonstrated how a microcontroller can precisely control external components through programmed logic. Expanding on this, the traffic control system simulated real-world signal operations using multiple LEDs. The system successfully replicated the sequence of a standard traffic light: green(go), yellow(caution)**,** and red(stop), ensuring a structured and repeating cycle. This experiment reinforced our understanding of timing control, loop execution, and hardware-software interaction.

Through this practical implementation, we gained hands-on experience with microcontroller programming, digital electronics, and embedded system design. Additionally, we understood the importance of precise timing in automation and control systems.

Here are some real life scenario and application of this experiment:

1.Smart Traffic Management:

Advanced microcontroller-based traffic lights use sensors and AI to adjust traffic signals dynamically based on real-time traffic flow, reducing congestion in busy intersections.

Example: Modern cities implement smart traffic systems that prioritize emergency vehicles by using microcontrollers to detect and clear their path.

2.Pedestrian Crosswalk Signals:

Traffic lights controlled by button-based microcontrollers allow pedestrians to request crossing time, ensuring safer road usage.

Example: Many urban areas use smart crosswalk systems where microcontrollers activate signals when pedestrians press a button.

3.Railway Crossing Signals:

Microcontrollers manage automatic railway crossings, where LED signals and barriers operate when a train approaches, reducing human intervention.

Example: In automated railway crossings, a microcontroller turns on red lights and lowers the gate when it detects an incoming train.

4.Factory and Industrial Automation:

Industries use microcontroller-based LED indicators for automated machinery, signaling operating status, machine errors, and safety warnings.

Example: Manufacturing plants use blinking LEDs to indicate machine cycles, ensuring operators are aware of running processes.

5.Home Automation Systems:

Microcontrollers enable automatic lighting by turning LEDs or bulbs on/off based on motion detection, voice commands, or mobile apps.

Example: Smart home systems automatically switch LED lights ON when a person enters a room and OFF when they leave.

6.Medical Equipment and Emergency Signals:

Microcontroller-controlled indicator LEDs are used in medical devices to alert healthcare workers to patient conditions.

Example: In hospitals, heart rate monitors and emergency alert systems use blinking LEDs to indicate vital signs and warnings.

Conclusion:

This experiment deepened our understanding of microcontroller-based automation through the LED blink test and traffic light control system. The blink test introduced fundamental digital output control, while the traffic system demonstrated sequential execution, precise timing, and real-world automation. We gained valuable insights into hardware-software integration, loop execution, and embedded system design, essential for advanced applications. This knowledge is crucial for developing smart traffic management, industrial automation, and IoT-based systems. Overall, this experiment enhanced our technical proficiency and problem-solving skills, laying a strong foundation for future advancements in intelligent and adaptive automation technologies.

References:

[1] https://www.arduino.cc/.
[2] https://www.educba.com
[3] https://www.geeksforgeeks.org/led-blinking-using-arduino