Assignment Report

Course name: Microprocessor Based Systems Design

Course code: CSC 471

Sec: A

Traffic Light Control System with Pedestrian Crossing

Submitted To

Md. Alomgir Hossain

Assistant professor, CSE Dept. IUBAT

Submitted by

Rakibul Hasan

21203070

Program: BCSE

Semester: Fall -2024

Submission Date: January 8, 2025

1. Problem Statement

Traffic light systems are a critical component of modern transportation infrastructure, ensuring smooth vehicle flow and pedestrian safety. This project aims to design and simulate a traffic light control system with an integrated pedestrian crossing feature. When a pedestrian button is pressed, the system halts normal traffic flow and allows safe pedestrian passage. The simulation is implemented using Arduino and Proteus software.

2. Objective

- To design and simulate a functional traffic light system with a pedestrian crossing.
- To prioritize pedestrian safety by halting traffic when a button is pressed.
- To utilize Arduino programming and Proteus simulation for implementing the system.

3. Apparatus Required

1. **Software**:
    - Proteus simulation software
    - Arduino IDE for programming

2. **Simulation Components**:
    - Arduino Uno R3 (Microcontroller)
    - LEDs (Red, Yellow, Green, Green LED (Pedestrian))
    - Push Button
    - Virtual Resistors (220 Ω)

4. Description of Assignment

This project involves the design of a traffic light control system with a pedestrian crossing feature. The traffic lights follow a standard sequence (Red ➔ Green ➔ Yellow). When a pedestrian button is pressed, the system halts the normal

sequence, activates the pedestrian crossing light, and then resumes normal traffic flow after 5 seconds.

5. Implementation of Assignment Circuit

Circuit Design

The circuit is designed using Proteus, with the following connections:

- **Traffic Lights**:
  - Red LED connected to Pin 8.
  - Yellow LED connected to Pin 9.
  - Green LED connected to Pin 10.
- **Pedestrian Light**:
  - Pedestrian crossing LED connected to Pin 5.
- **Push Button**:
  - Connected to Pin 2 and GND, using the Arduino's internal pull-up resistor.

Code Implementation

The Arduino code controls the traffic lights and pedestrian crossing functionality. The button press triggers an interrupt in the traffic sequence to prioritize pedestrian safety. The system resumes normal operation after 5 seconds.

```
// Traffic Light Control with Pedestrian Crossing
const int RED_PIN = 8;       // Traffic Red LED
const int YELLOW_PIN = 9;    // Traffic Yellow LED
const int GREEN_PIN = 10;    // Traffic Green LED
const int PED_LED_PIN = 5;   // Pedestrian Walk LED
const int BUTTON_PIN = 2;    // Button Pin

bool pedestrianActive = false;  // Flag to track pedestrian crossing

void setup() {
    pinMode(RED_PIN, OUTPUT);
    pinMode(YELLOW_PIN, OUTPUT);
    pinMode(GREEN_PIN, OUTPUT);
    pinMode(PED_LED_PIN, OUTPUT);
    pinMode(BUTTON_PIN, INPUT_PULLUP); // Use internal pull-up resistor
}
```

```
void loop() {
    // Check if pedestrian button is pressed
    if (digitalRead(BUTTON_PIN) == LOW) { // Button pressed
        pedestrianActive = true;
        pedestrianCrossing();
        pedestrianActive = false;             // Set pedestrian crossing flag
    }
    digitalWrite(RED_PIN, HIGH);    // Traffic Red ON
    digitalWrite(YELLOW_PIN, LOW);  // Ensure Yellow is OFF
    digitalWrite(GREEN_PIN, LOW);   // Ensure Green is OFF
    delay(5000);
    // Check if pedestrian button is pressed
    if (digitalRead(BUTTON_PIN) == LOW) { // Button pressed
        pedestrianActive = true;
        pedestrianCrossing();
        pedestrianActive = false;             // Set pedestrian crossing flag
    }
    digitalWrite(RED_PIN, LOW);     // Traffic Red OFF
    digitalWrite(GREEN_PIN, HIGH);  // Traffic Green ON
    delay(5000);
    // Check if pedestrian button is pressed
    if (digitalRead(BUTTON_PIN) == LOW) { // Button pressed
        pedestrianActive = true;
        pedestrianCrossing();
        pedestrianActive = false;             // Set pedestrian crossing flag
    }
    digitalWrite(GREEN_PIN, LOW);   // Traffic Green OFF
    digitalWrite(YELLOW_PIN, HIGH); // Traffic Yellow ON
    delay(2000);                    // Wait 2 seconds
    digitalWrite(YELLOW_PIN, LOW);
}


// Pedestrian Crossing Sequence
void pedestrianCrossing() {
    // Turn on pedestrian crossing lights
    digitalWrite(RED_PIN, HIGH);       // Traffic Red ON
    digitalWrite(GREEN_PIN, LOW);      // Traffic Green OFF
    digitalWrite(YELLOW_PIN, LOW);     // Traffic Yellow OFF
    digitalWrite(PED_LED_PIN, HIGH);   // Pedestrian Walk Light ON
    delay(5000);                       // Wait 5 seconds for crossing

    // Turn off pedestrian lights and resume normal cycle
    digitalWrite(PED_LED_PIN, LOW);    // Pedestrian Walk Light OFF
```

```
}
```

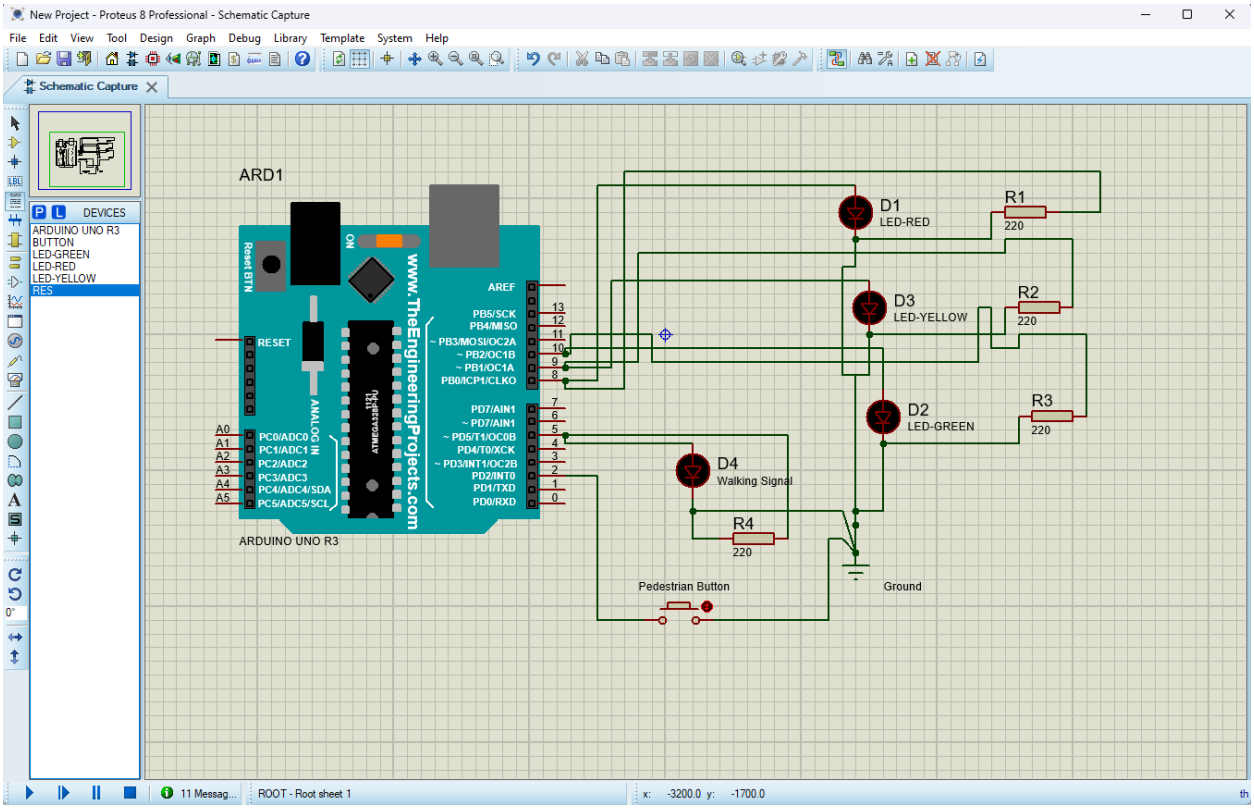
6. Result Analysis

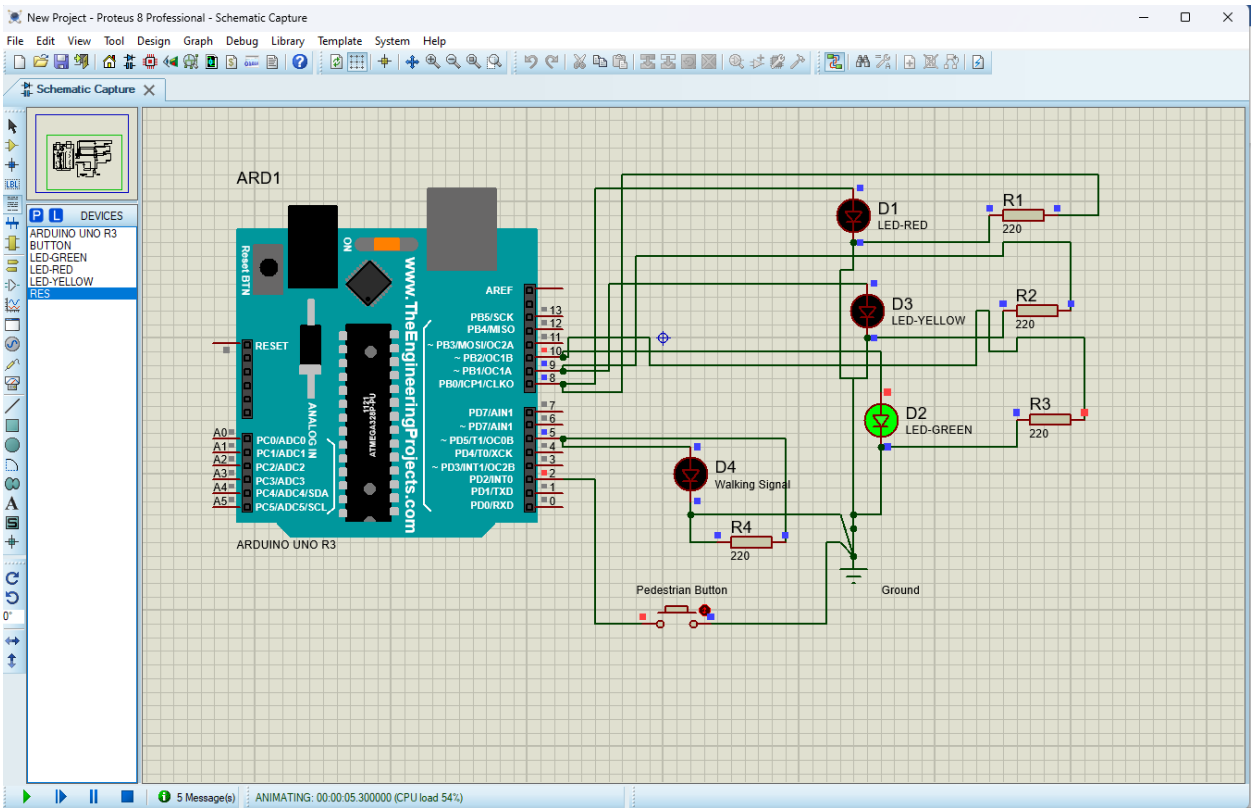
The simulation was run in Proteus. The LEDs successfully followed the programmed sequence:

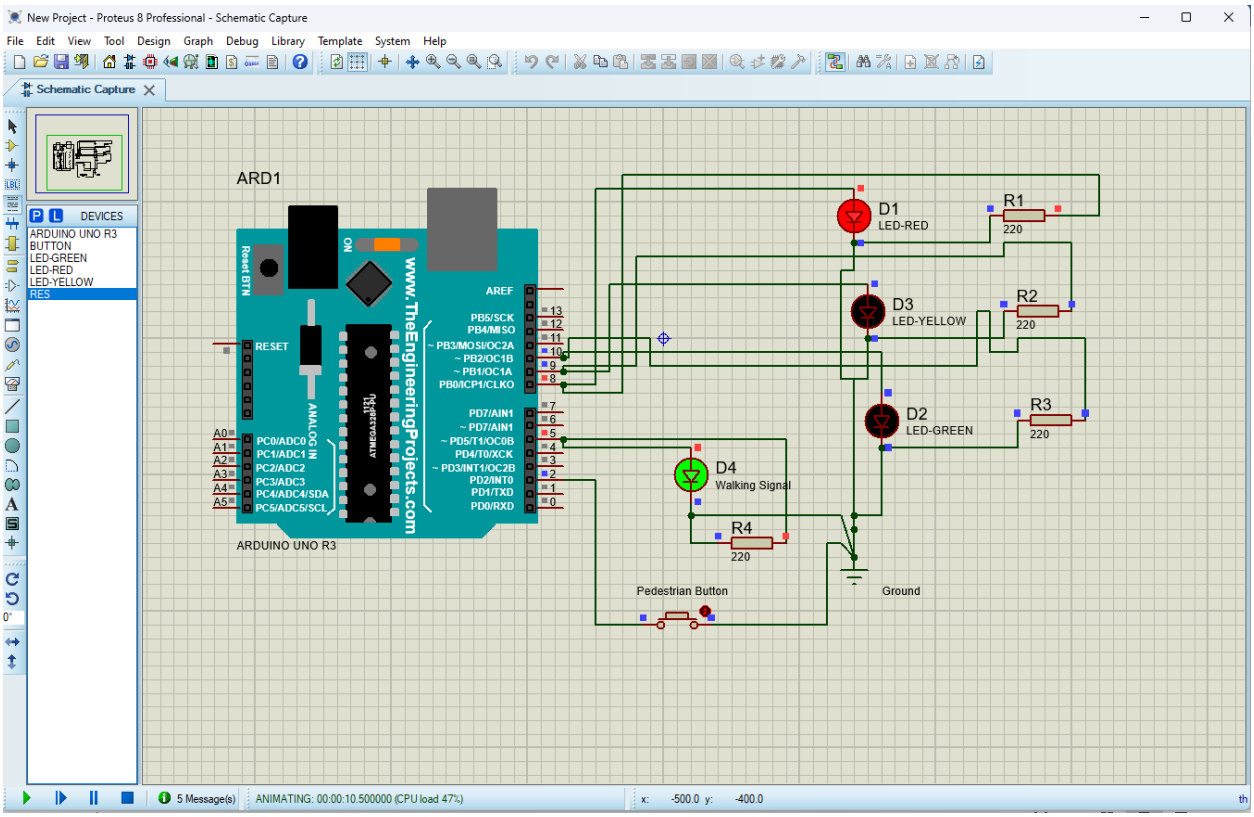
1. Normal traffic light sequence operates as expected (Red ➔ Green ➔ Yellow).
2. Pedestrian button interrupts the sequence, activating the pedestrian crossing light and stopping traffic.
3. After 5 seconds, the system smoothly resumes the traffic light sequence.

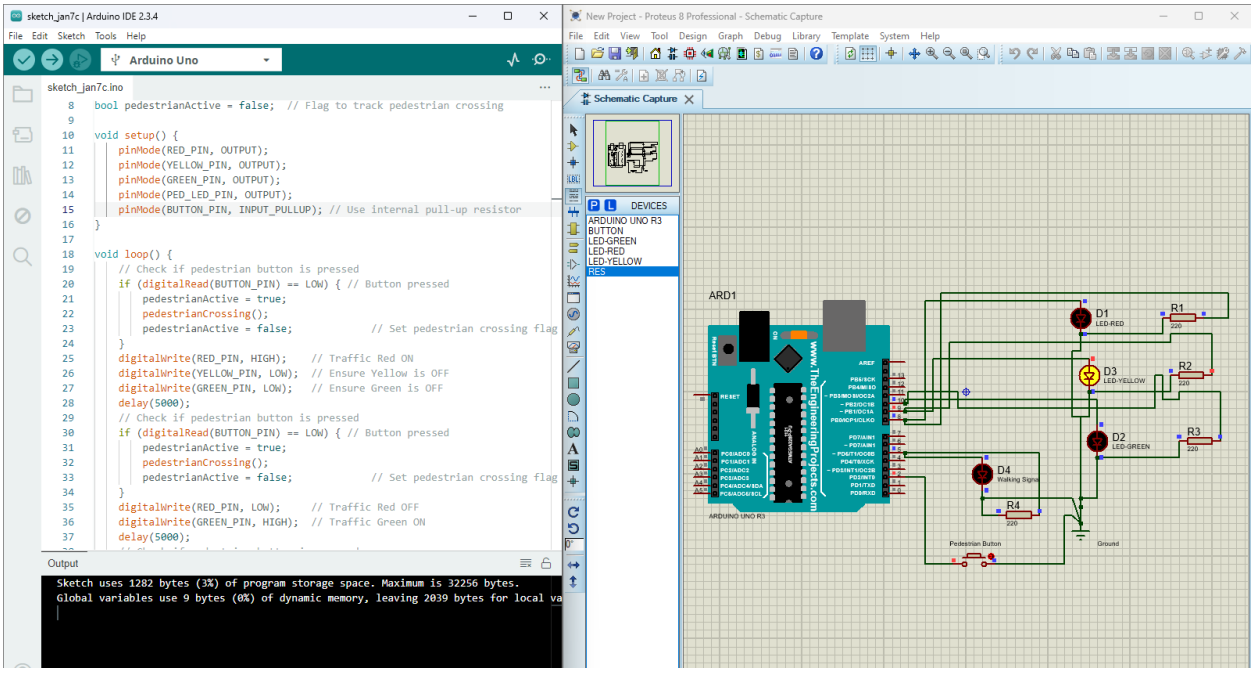
This validated the proper functioning of the traffic light control system.

Simulation Screenshots:




New Project - Proteus 8 Professional - Schematic Capture
File Edit View Tool Design Graph Debug Library Template System Help
Schematic Capture
DEVICES
ARDUINO UNO R3
BUTTON
LED-GREEN
LED-RED
LED-YELLOW
RES
ARD1
Reset BTN
ON
RESET
ANALOG IN
www.TheEngineeringProjects.com
AREF
PB5/SCK
PB4/MISO
~ PB3/MOSI/OC2A
~ PB2/OC1B
~ PB1/OC1A
PB0/ICP1/CLKO
PD7/AIN1
~ PD7/AIN1
~ PD5/T1/OC0B
PD4/T0/XCK
~ PD3/INT1/OC2B
PD2/INT0
PD1/TXD
PD0/RXD
PC0/ADC0
PC1/ADC1
PC2/ADC2
PC3/ADC3
PC4/ADC4/SDA
PC5/ADC5/SCL
ARDUINO UNO R3
D1
LED-RED
R1
220
D3
LED-YELLOW
R2
220
D2
LED-GREEN
R3
220
D4
Walking Signal
R4
220
Pedestrian Button
Ground
5 Message(s)
ANIMATING: 00:00:05.300000 (CPU load 54%)


New Project - Proteus 8 Professional - Schematic Capture
File Edit View Tool Design Graph Debug Library Template System Help
Schematic Capture
DEVICES
ARDUINO UNO R3
BUTTON
LED-GREEN
LED-RED
LED-YELLOW
RES
ARD1
Reset BTN
ON
RESET
ANALOG IN
www.TheEngineeringProjects.com
AREF
PB5/SCK
PB4/MISO
~ PB3/MOSI/OC2A
~ PB2/OC1B
~ PB1/OC1A
PB0/ICP1/CLKO
PD7/AIN1
~ PD7/AIN1
~ PD5/T1/OC0B
PD4/T0/XCK
~ PD3/INT1/OC2B
PD2/INT0
PD1/TXD
PD0/RXD
PC0/ADC0
PC1/ADC1
PC2/ADC2
PC3/ADC3
PC4/ADC4/SDA
PC5/ADC5/SCL
ARDUINO UNO R3
D1
LED-RED
R1
220
D3
LED-YELLOW
R2
220
D2
LED-GREEN
R3
220
D4
Walking Signal
R4
220
Pedestrian Button
Ground
5 Message(s)
ANIMATING: 00:00:10.500000 (CPU load 47%)
x: -500.0 y: -400.0



7. Procedure to Perform

1. Open Proteus and create a new project.
2. Add the Arduino Uno microcontroller and connect the LEDs (Red, Yellow, Green, PED Green Light) to digital pins 8, 9, 10 and 5 through virtual resistors which is 220 ohms.
3. Load the provided C++ code into the Arduino in Proteus.
4. Start the simulation and observe the traffic light sequence.
5. Verify that the lights follow the correct timing and order.

8. Conclusion

This project successfully demonstrates a traffic light system integrated with a pedestrian crossing feature. It prioritizes pedestrian safety by halting traffic on demand and resuming normal operation afterward. The system is efficient, modular, and can be scaled for real-world implementation.