Smart Traffic Light Controller

PBL Mini Project Report

Course Name → PBL_1 (CSE20030)
Academic Year → 2025–26
Semester → III
Course → SY B.Tech (CSE)
Department Name → School of Computer Science and Engineering
Panel → A2

Name	Roll No	Contribution	Prn
Mohammed Azad	27	PPT & Report	1262240680
Rupesh Kumar Garg	28	Hardware & Simulation	1262240690
Satyam Jha	32	Hardware & Code	1262240703
Shivam Mane	71	Simulation & Report	1262253260

2. Abstract

This mini-project details the design and implementation of an **Intersectional Traffic Light Controller** using an **Arduino Uno Microcontroller**.

The aim of our project was to replicate how a traffic light system works in tandem w/ each other at a crossroad. In this proof-of-concept model, a four-way intersection is simulated using high-intensity LEDs (Red, Yellow, Green) to represent the actual traffic lights. The Arduino serves as the central controller, programmed with precise hardcoded time delays and a coordinated sequence to manage the signal changes. This ensures a synchronized and continuous operation, preventing the conflicting signals that lead to gridlock.

While this implementation uses pre-set cycles, it provides a robust and scalable foundation for a fully adaptive control system. The architecture is intentionally designed for future expansion, such as the integration of IR or ultrasonic sensors to detect actual vehicle presence. This would allow the system to dynamically adjust signal timing based on real-time traffic density.

While this model is in no way challenging the traditional traffic light system logic, it serves as a proof-of-concept learning opportunity for us PBL students to get hands-on w/ hardware.

3. Table of Contents

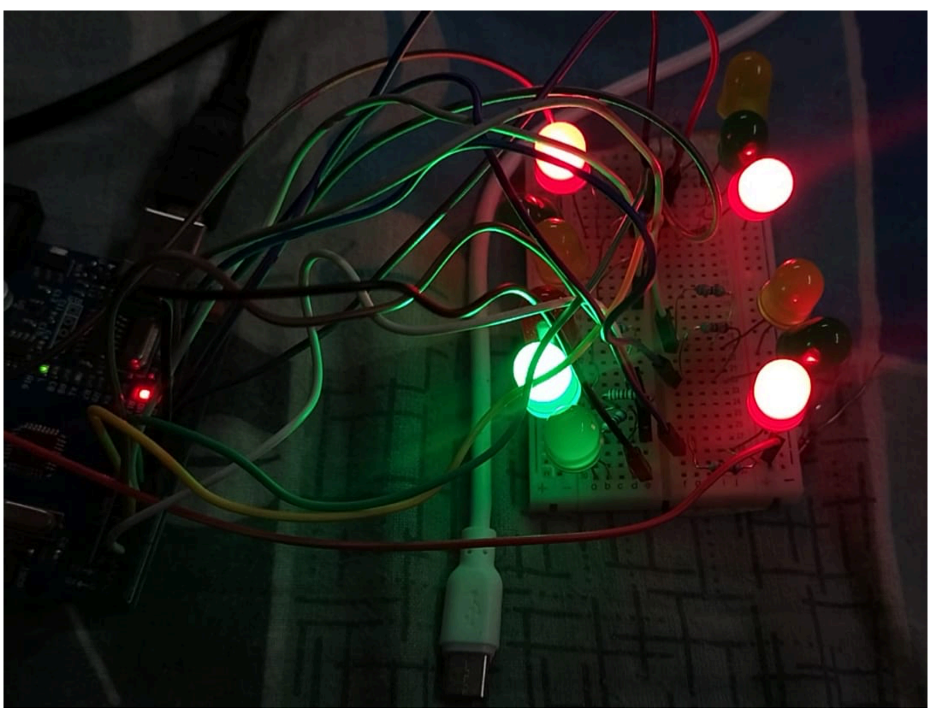
4. Introduction

Objective of the Project

The objective of this project is to design and implement an imitated traffic light system using the Arduino Uno microcontroller. This system provides accurate signal transitions at a crossroad junction and ensures traffic management without human intervention.

Project Motivation

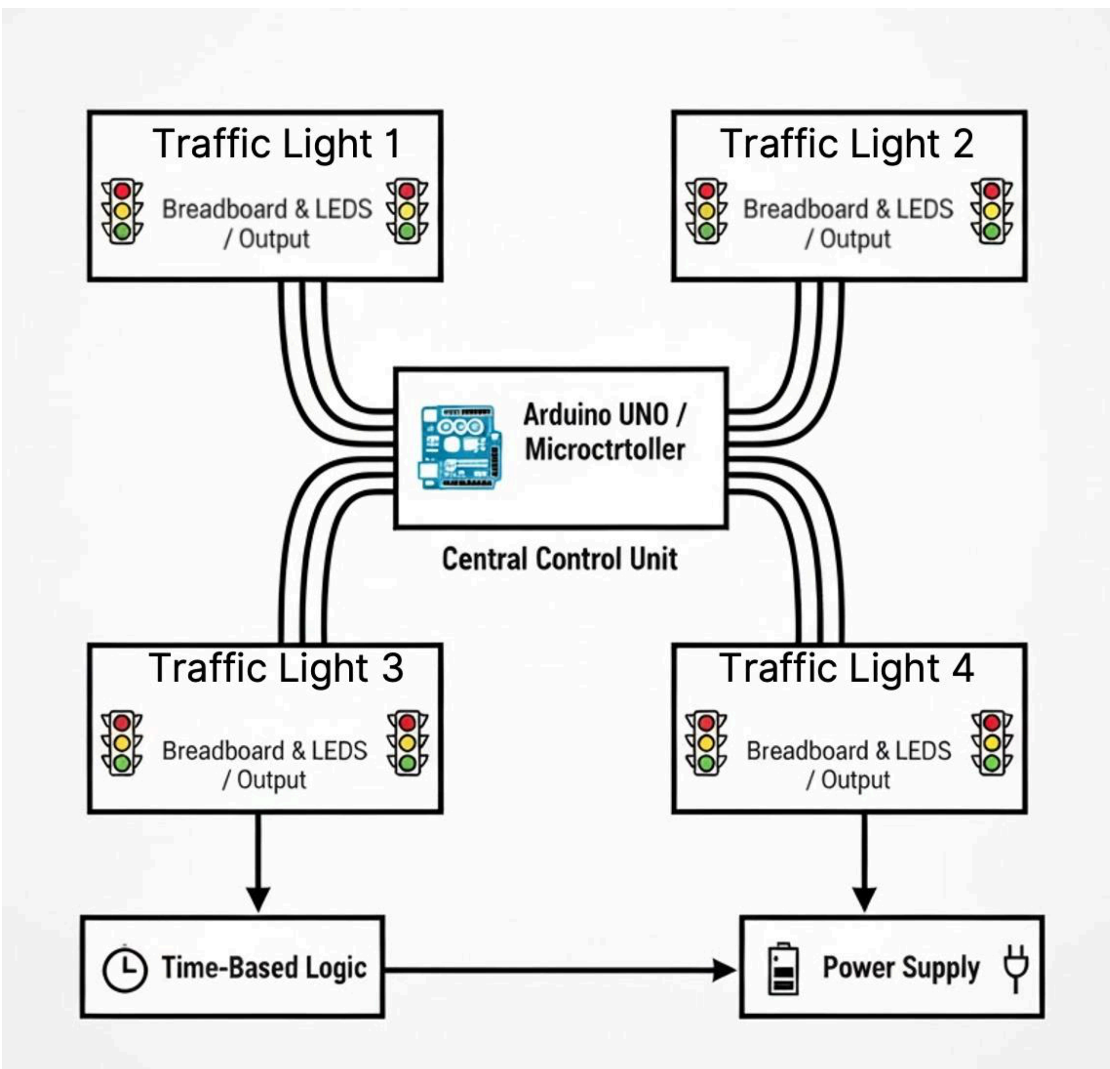
Learning to work with hardware as predominantly software students by course. We wish to apply practically our programming skills to simulate a real world scenario use-case, it may not be robust as of now, however this project does serve as a valuable early exposure to hardware programming.



5. Implementation

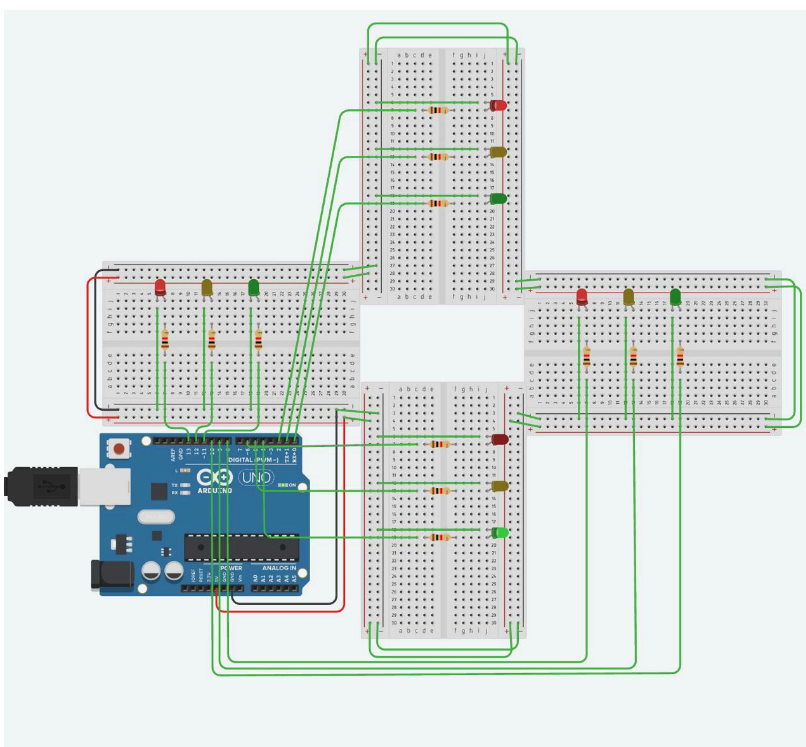
5.1 Block Diagram/Logic Diagram

The system is composed of an **Arduino Uno**, **12 LEDs** (Red, Yellow, Green) representing four directions, **220Ω resistors**, and a **5V power source**. The Arduino controls each LED according to hardcoded (pre-set time) intervals to simulate real-time signal changes.



5.2 Explanation of Components

Component	Function in the System	Microcontroller Interaction (Pins)
Arduino Uno	Main controller executing programmed signal logic.	Digital Pins 2–13
LEDs (Red, Yellow, Green)	Represent traffic signals for each direction.	Digital Pins 2–13
220Ω Resistors	Protect LEDs from excessive current.	In series with LEDs
Breadboard & Jumper Wires	Provide connections for circuit setup.	N/A
5V Power Supply	Provides DC power to the entire system.	Vcc, GND



5.3 Implemented Code

The following C++ code controls the sequence of lights in each direction using delay-based timing. (deployed & ported using **Arduino IDE**)

```cpp
const int redPins[4]    = {2, 8, 6, 13};
const int yellowPins[4] = {7, 9, 5, 12};
const int greenPins[4]  = {3, 10, 4, 11};

const unsigned long GREEN_TIME  = 5000;
const unsigned long YELLOW_TIME = 2000;

void setup() {
  for (int i = 0; i < 4; i++) {
    pinMode(redPins[i], OUTPUT);
    pinMode(yellowPins[i], OUTPUT);
    pinMode(greenPins[i], OUTPUT);

    digitalWrite(redPins[i], HIGH);
    digitalWrite(yellowPins[i], LOW);
    digitalWrite(greenPins[i], LOW);
  }

  delay(500);
}

void loop() {
  for (int dir = 0; dir < 4; dir++) {
    activateDirection(dir);
  }
}

void activateDirection(int dir) {
  for (int i = 0; i < 4; i++) {
    digitalWrite(redPins[i], HIGH);
    digitalWrite(greenPins[i], LOW);
    digitalWrite(yellowPins[i], LOW);
  }

  digitalWrite(redPins[dir], LOW);
  digitalWrite(greenPins[dir], HIGH);
  delay(GREEN_TIME);

  digitalWrite(greenPins[dir], LOW);
  digitalWrite(yellowPins[dir], HIGH);
  delay(YELLOW_TIME);

  digitalWrite(yellowPins[dir], LOW);
  digitalWrite(redPins[dir], HIGH);
}
```

5.4 Hardware and Software Specification

Hardware Specifications:

Specification	Detail
Microcontroller	Arduino Uno
LEDs	12 (Red, Yellow, Green)
Resistors	12 × 220Ω Resistors
Connections	Breadboard and Jumper Wires
Power Supply	5V DC Power Supply

Software Specifications:

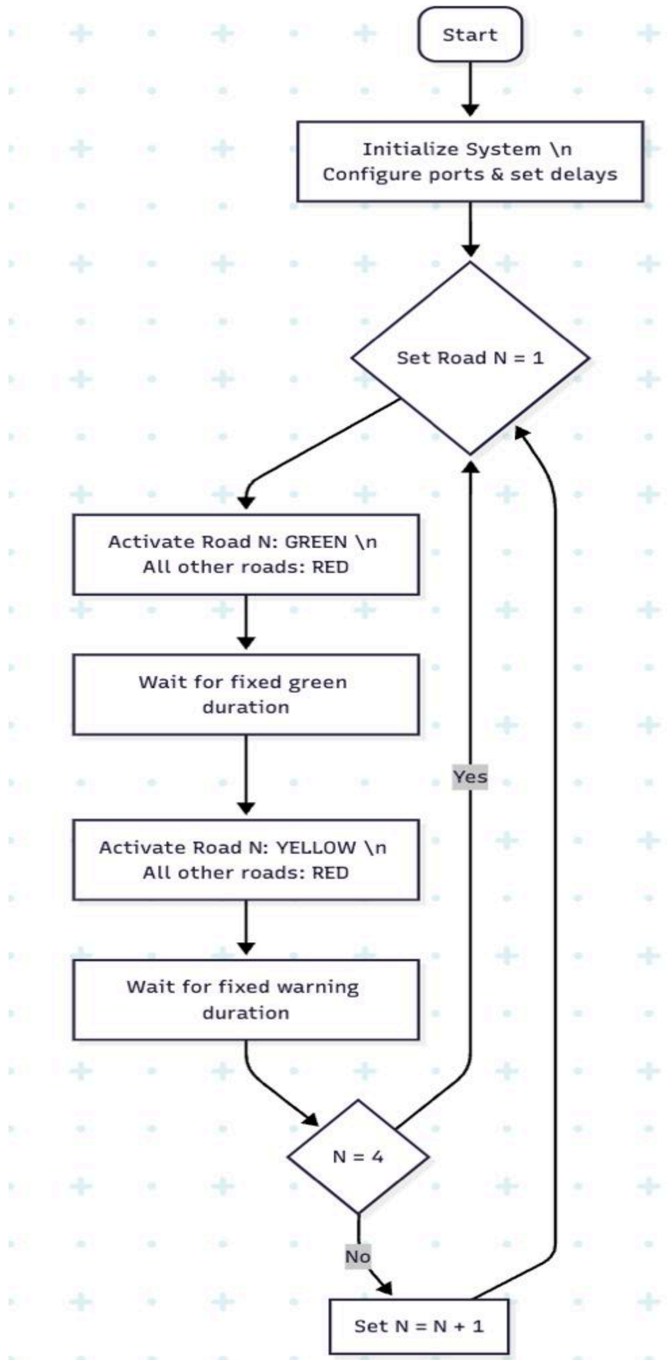
Specification	Detail
Programming Language	Arduino C/C++
IDE	Arduino IDE
Simulation Tool	TinkerCAD
Compiler	Built-in Arduino Compiler

5.5 Flowchart / Algorithm

Algorithm:

1. Initialize Arduino and configure all pins (2-13) as output.
2. Activate the green LED for the first direction (green1) and red for all others (red2, red3, red4).
3. After 5 seconds, switch green1 LOW and activate yellow1 for 2 seconds as a warning.
4. Switch yellow1 LOW. Switch red2 LOW and green2 HIGH.
5. Repeat this pattern (Green-Yellow transition) for direction 2, then direction 3, then direction 4.
6. Repeat the entire sequence indefinitely.

Flowchart:


Start
Initialize System \n
Configure ports & set delays
Set Road N = 1
Activate Road N: GREEN \n
All other roads: RED
Wait for fixed green
duration
Activate Road N: YELLOW \n
All other roads: RED
Wait for fixed warning
duration
N = 4
Yes
No
Set N = N + 1

6. Result

6.1 Test Cases

Test Case No.	Input	Expected Output	Actual Results	Status
1	Run Program	Sequential green signal for each road	Matched expected output	PASS
2	Run Program	Yellow activates before each transition	Matched expected output	PASS
3	Run Program	Red lights remain ON for other roads while one is green/yellow	Matched expected output	PASS

6.2 Results

The Smart Traffic Light Controller system operated successfully during testing and simulation. Each direction received appropriate signal durations, and transitions occurred smoothly. The timing logic executed continuously, confirming the reliability of the programmed sequence.

7. Conclusion

The Smart Traffic Light Controller using Arduino was successfully developed and tested. It automates signal control. This system forms a foundation for future development involving IR sensors and IoT modules for adaptive control.

8. References

1. **Arduino Official Documentation** → https://www.arduino.cc
2. **TinkerCAD Simulation Platform** → https://www.tinkercad.com
3. **MIT-WPU Microcontroller Lab Materials**
4. **Course** → Microprocessor, Microcontroller and Applications