Chandigarh Engineering College Jhanjeri
Mohali-140307
Department of Artificial Intelligence (AI) and Data Sciences

Mid Term Report
On

Adaptive Traffic Light Control System

Engineering Clinics – Multi Disciplinary Project

BACHELOR OF TECHNOLOGY

(Artificial Intelligence (AI) and Data Sciences)

SUBMITTED BY:
Bhoomi Chauhan (2420694)
Bhoomika Rana (2420695)
Bhumika (2420696)
Bitu Sahu (2420694)
Nov 2025

Under the Guidance of
Er. Shubham Sharma J4224
(Assistant Professor)

Department of Computer Science & Engineering
Chandigarh Engineering College
Jhanjeri, Mohali - 140307

Table of Contents

Chapter 1: Introduction

1.1 Main Idea of the Project

Urban traffic congestion is one of the most pressing challenges faced by modern cities, leading to increased travel times, fuel consumption, and environmental pollution. To address these issues, traffic management systems need to be more adaptive and intelligent. Our project introduces an Adaptive Traffic Light Control System (ATLCS) that synchronizes consecutive traffic lights by applying dynamic delays between green signals in a given direction. These delays are updated in real time based on the number of vehicles waiting at each junction. This approach enables vehicles, particularly those leaving the city center, to cover longer distances with fewer stops, thereby reducing the "stop-and-go" effect and improving travel efficiency.

Performance evaluations of the ATLCS have shown promising results. In the synchronized direction, the average travel time of vehicles was reduced by up to 39% compared to conventional non-synchronized fixed-time traffic light systems. Across the entire simulated road network, the overall improvement achieved was 17%.

The broader context of this project lies in the vision of the Smart City, which seeks to create sustainable ecosystems while promoting citizen welfare and economic development. Smart Cities rely on advanced information and communication technologies (ICTs) to monitor and manage critical urban assets such as energy, water, and transportation. Realizing this vision requires collaboration among governments, industries, academia, and communities. Furthermore, enabling technologies such as the Internet of Things (IoT), 5G, cloud computing, artificial intelligence, and connected vehicles play a vital role in achieving sustainable advancements, including smart buildings, renewable energy systems, and intelligent green transportation.

This project contributes to that vision by presenting an innovative traffic management solution that supports sustainable urban mobility, improves travel efficiency, and enhances the overall quality of life in cities.

1.2 Objectives

The primary objective of this project is to design and implement an **Adaptive Traffic Control System (ATCS)** that optimizes traffic signal operations in real time, reducing congestion, waiting times, and environmental impact at urban intersections. Unlike traditional fixed-time or semi-actuated signals, the ATCS dynamically responds to fluctuating traffic patterns, ensuring smoother traffic flow and more efficient road use.

The specific objectives of the project are:

1. **Real-time Traffic Monitoring:** Collect and process traffic data from sensors, cameras, or datasets to track vehicle density, queue lengths, and flow patterns, and detect peak and off-peak periods.
2. **Dynamic Signal Timing:** Adjust green, yellow, and red signal durations dynamically and coordinate multiple intersections to minimize delays and prevent bottlenecks.
3. **Reduction of Travel Time and Congestion:** Apply adaptive algorithms to decrease waiting times and improve traffic flow.
4. **Environmental and Economic Benefits:** Lower vehicle emissions and fuel consumption by reducing idle time, contributing to cleaner air and cost savings.

Chapter 2: System Requirements

The implementation of our **Adaptive Traffic Control System (ATCS)** requires the following hardware and components:

Hardware:

- **Arduino Uno:** The Arduino Uno acts as the main controller for the system. It processes input signals, executes the adaptive traffic control algorithm, and controls the timing and sequencing of the LEDs to simulate traffic signals. Its simplicity, reliability, and versatility make it ideal for prototyping electronic control systems.
- **LEDs (Red, Yellow, Green):** Three LEDs are used to represent a standard traffic light. The **Red LED** indicates stop, the **Yellow LED** signals caution, and the **Green LED** allows traffic to move. These LEDs visually demonstrate the changes in signal states according to the adaptive algorithm.
- **Connecting Wires:** Wires are used to connect the Arduino board to the LEDs and ensure proper electrical continuity. They allow flexible and reconfigurable connections, which are essential for testing and modifying the prototype circuit.
- **Breadboard:** A breadboard provides a solderless platform for assembling the circuit. It allows for easy placement and rearrangement of components, making it convenient to test different configurations and troubleshoot the system during development.

Software:

- **Arduino IDE:** Used to write, compile, and upload the program to the Arduino. It provides tools and libraries needed to control the LEDs and implement the adaptive traffic signal logic. The IDE also allows monitoring of the program's execution and debugging, which helps in testing and improving the system efficiently.

Chapter 3: System Requirements Analysis

1. Purpose of the Software:

- The software controls the traffic light sequence (Red, Yellow, Green) based on adaptive algorithms that simulate real-time traffic conditions.
- It ensures smooth traffic flow, minimizes waiting time, and demonstrates the adaptive behaviour of traffic signals at intersections.

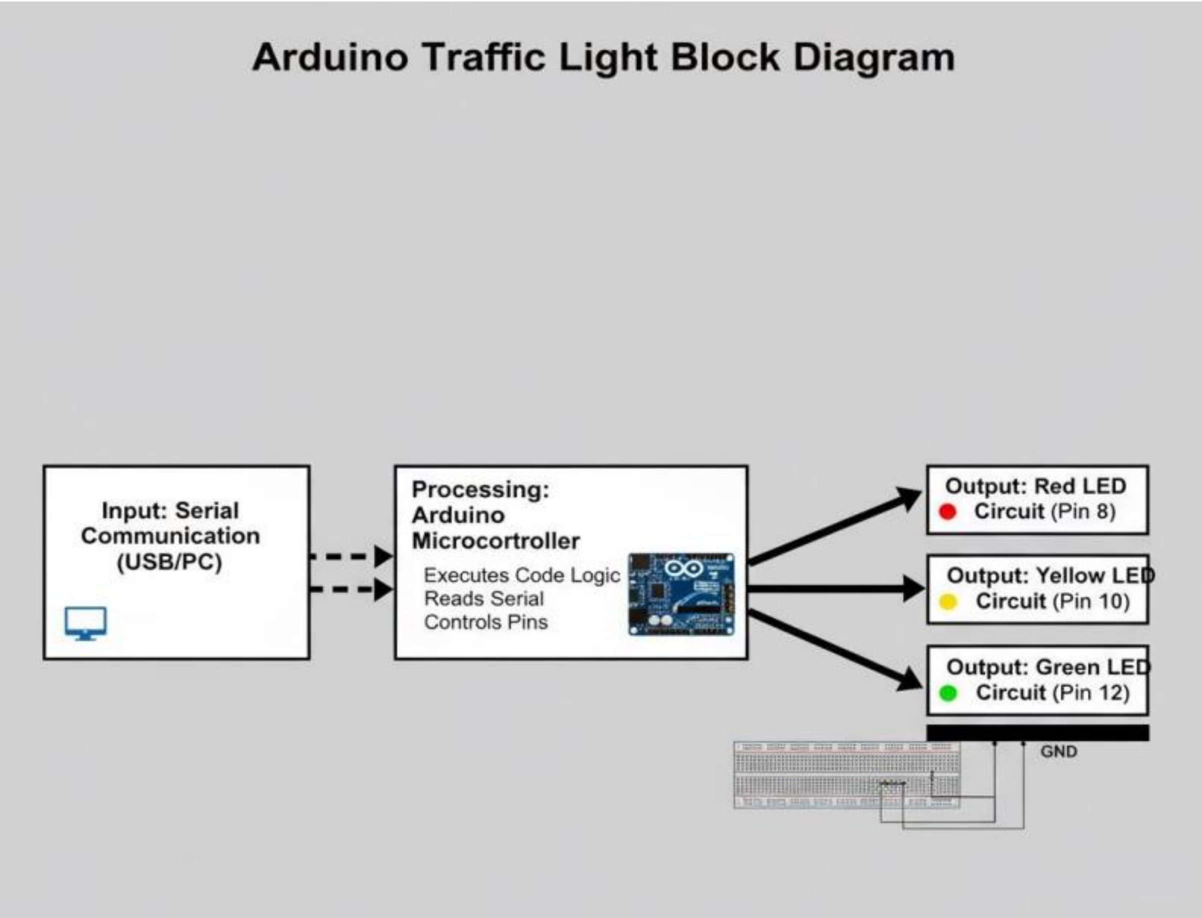
2. Functional Requirements:

- Control the LEDs to represent traffic signal changes.
- Implement timing logic for Red, Yellow, and Green lights based on predefined conditions (or simulated traffic data).
- Allow dynamic adjustment of signal duration to simulate adaptive control.
- Monitor system execution and provide feedback for debugging or testing purposes.

3. Non-Functional Requirements:

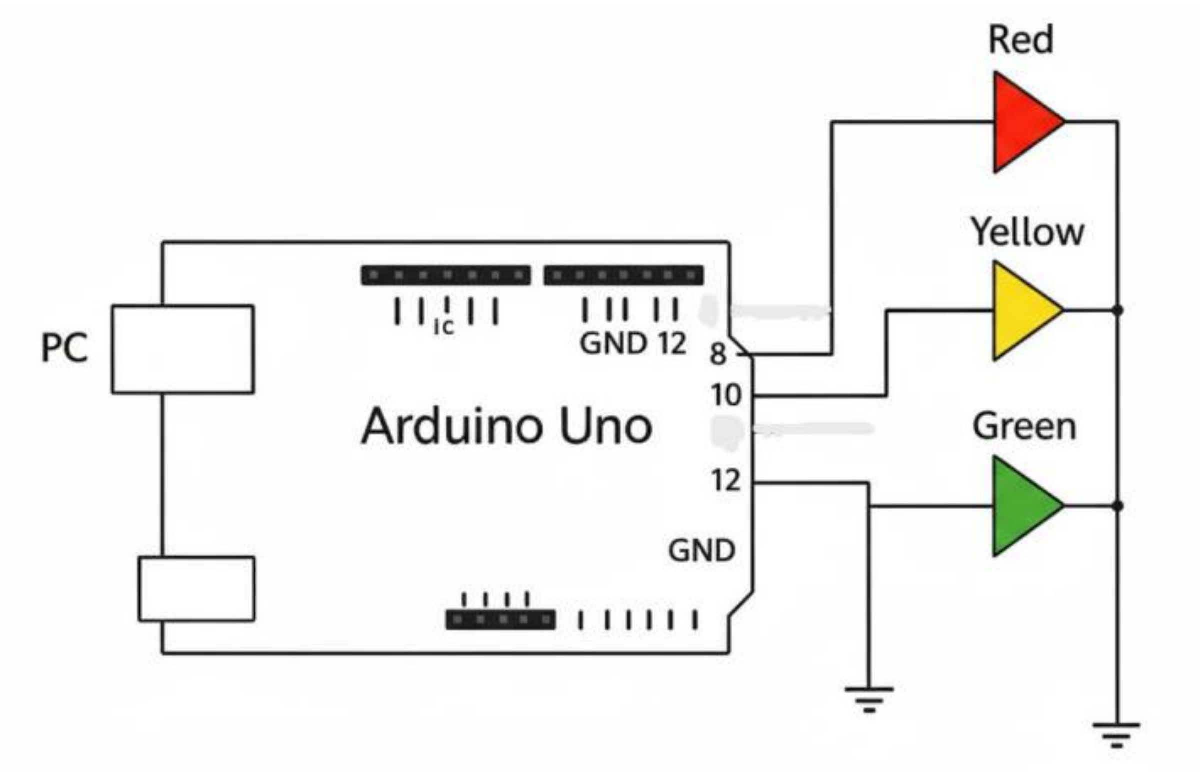
- Reliability: The program should run without errors during testing.
- Efficiency: The system should respond promptly to changes in simulated traffic conditions.
- Usability: The software should be easy to upload and modify via Arduino IDE.

Chapter 4: Circuit Design



The block diagram illustrates the overall functioning of the Arduino-based traffic light system and shows how input, processing, and output components interact within the model. The system begins with the input stage, where commands are sent from a computer to the Arduino through serial communication via a USB connection. These commands determine which traffic light phase—red, yellow, or green—should be activated. The processing stage is handled by the Arduino microcontroller. After receiving a command through the serial interface, the Arduino interprets the input, executes the corresponding section of the program, and controls the appropriate output pins. This stage forms the core of the system, as it contains the programmed logic responsible for switching between different traffic light signals. The output stage consists of three separate LED circuits connected to designated Arduino pins. The red LED is controlled through pin 8, the yellow LED through pin 10, and the green LED through pin 12. When the microcontroller activates a specific pin, the corresponding LED lights up to represent the chosen traffic signal state. Each LED is connected to the breadboard with resistors and is grounded to complete the circuit. Overall, the block diagram provides a simplified representation of how serial input is processed by the Arduino and translated into visual traffic signals through the connected LEDs. It clearly shows the flow of information from input to processing and finally to output, demonstrating the structure and functionality of the implemented traffic light model.

Circuit Diagram:



Arduino Traffic Light Schematic Diagram (Controlled via PC Serial)

Chapter 5: Implementation

The Arduino IDE codes used in the working of the LEDs using the breadboard, Arduino UNO and resistor is given below.

sketch_oct15a > sketch_oct15a.ino

```
const int red = 8;
const int yellow = 10;
const int green = 12;

void setup() {
  pinMode(red, OUTPUT);
  pinMode(yellow, OUTPUT);
  pinMode(green, OUTPUT);
  Serial.begin(9600);  // enable serial communication
}

void loop() {
  if (Serial.available()) {
    char cmd = Serial.read();

    if (cmd == 'R') {
      digitalWrite(red, HIGH);
      delay(5000);
      digitalWrite(red, LOW);
    }

    else if (cmd == 'Y') {
      digitalWrite(yellow, HIGH);
      delay(2000);
      digitalWrite(yellow, LOW);
    }

    else if (cmd == 'G') {
      digitalWrite(green, HIGH);
      delay(5000);
      digitalWrite(green, LOW);
    }
  }
}
```

The Arduino code is designed to control three LEDs representing the red, yellow, and green phases of a traffic signal. At the beginning of the program, the digital pins connected to the LEDs are defined, ensuring clarity and ease of modification. In the setup function, each of these pins is configured as an output, and serial communication is initialized so the Arduino can receive commands from the Serial Monitor. The main logic of the program runs inside the loop function, where the system continuously checks whether any input has been received through the serial interface. When a command is detected, a single character is read and used to determine which traffic light should activate. If the command received is ‘R’, the red LED turns on for a fixed duration and then turns off, simulating the stop phase. When the command is ‘Y’, the yellow LED is activated briefly to represent the caution phase. Similarly, a ‘G’ command triggers the green LED for a defined period, simulating the go signal. Each command activates only one LED at a time, and the use of delay functions ensures that the LEDs remain on for the appropriate amount of time before turning off. This simple control structure demonstrates how traffic light phases can be managed through serial input and digital output, forming the foundation of a basic adaptive traffic signal simulation.

project.py > ...

```python
import serial
import time

# Change 'COM3' to your Arduino's COM port
arduino = serial.Serial(port='COM9', baudrate=9600, timeout=1)

time.sleep(2)  # Wait for Arduino to initialize

def send_signal(signal):
    arduino.write(signal.encode())  # send one character command
    print(f"Signal sent: {signal}")

while True:
    # Green light
    send_signal('G')
    print("🟢 Green ON")
    time.sleep(5)

    # Yellow light
    send_signal('Y')
    print("🟡 Yellow ON")
    time.sleep(2)

    # Red light
    send_signal('R')
    print("🔴 Red ON")
    time.sleep(5)
```

A Python code implemented for the project model. This Python script is used to control the Arduino-based traffic light system by sending commands through serial communication. It establishes a connection between the computer and the Arduino and then repeatedly sends signals that activate the green, yellow, and red LEDs in sequence. The program begins by importing the required libraries: serial for communication with the Arduino and time for implementing delays. A serial connection is then created using the appropriate COM port, baud rate, and timeout value. A brief delay is added to allow the Arduino to initialize properly after the connection is established. A function named send_signal() is defined to simplify sending commands to the Arduino. This function takes a single character ('G', 'Y', or 'R'), encodes it, and transmits it over the serial port. It also prints a confirmation message indicating which signal was sent.

REFRENCES

1. A. Agrahari, MM Dhabu, PS Deshpande, A Tiwari, MA Baig, AD Sawarkar.Artificial Intelligence-based adaptive traffic signal control system: A comprehesive review. Electronics2024.

2. Jing, P.; Huang, H.; Chen, L. An Adaptive Traffic Signal Control in a Connected Vehicle Environment: A Systematic Review. Information 2017, 8, 101.

3. Noaeen, M.; Naik, A.; Goodman, L.; Crebo, J.; Abrar, T.; Abad, Z.S.H.; Bazzan, A.L.C.; Far, B. Reinforcement Learning in Urban Network Traffic Signal Control: A Systematic Literature Review. Expert Syst. Appl. 2022, 99, 116830.

4. Tian, Y.; Liu, S.; Yan, X.; Zhu, T.; Zhang, Y. Active Control Method of Traffic Signal Based on Parallel Control Theory. IEEE J. Radio Freq. Identif. 2024, 8, 334–340.

5. DR Aleko, S Djahel - Information, 2020, An Efficient Adaptive Traffic Light Control System for Urban Road Traffic Congestion Reduction in Smart Cities.

6. SM Shinde -intelligent systems and information management, 2017 - ieeexplore.ieee.org.