Traffic Light Control System

A PROJECT REPORT
for
AI (AI101B)
Session (2024-25)

Submitted by

Shadab Idrishi
202410116100190
Rahul Pratap Singh Chauhan
202410116100158
Shalini Mishra
202410116100192

Submitted in partial fulfilment of the
Requirements for the Degree of

MASTER OF COMPUTER APPLICATION

Under the Supervision of
Ms. Komal Salgotra
Assistant Professor

Submitted to

DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, Ghaziabad
Uttar Pradesh-201206
(DECEMBER- 2024)

CERTIFICATE

Certified that **Shadab Idrishi 202410116100190, Rahul Pratap Singh Chauhan 202410116100158, Shalini Mishra 202410116100192** have carried out the project work having Traffic Light Control System (**AI-101B**) for **Master of Computer Application** from Dr A.P.J. Abdul Kalam Technical University (AKTU) (formerly UPTU), Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself/herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Ms. Komal Salgotra
Assistant Professor
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

Dr. Aakash Rajak
Dean
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

Traffic Light Control System

Shadab Idrishi
Rahul Pratap Singh Chauhan
Shalini Mishra

ABSTRACT

A traffic light control system is a crucial component of modern urban traffic management, designed to regulate vehicular and pedestrian movement at intersections efficiently. This project explores the development of a basic traffic light control system, utilizing both hardware-based (microcontrollers and sensors) and software-based (programming logic) approaches. The system operates on a predefined sequence of red, yellow, and green signals to control traffic flow and ensure safety. Advanced implementations incorporate real-time data processing, sensor-based optimization, and AI-driven adaptability to manage traffic density dynamically. The methodology involves system design, programming logic implementation, hardware/software setup, testing, and optimization. A Python-based simulation is provided to demonstrate a basic traffic light sequence. This project highlights the importance of smart traffic management solutions and paves the way for future enhancements integrating IoT and AI technologies for better urban mobility.

ACKNOWLEDGEMENTS

Success in life is never attained single-handedly. My deepest gratitude goes to my project supervisor, **Ms. Komal Salgotra** for her guidance, help, and encouragement throughout my project work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to Dr. Aakash Rajak, Professor and Dean, Department of Computer Applications, for his insightful comments and administrative help on various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me with moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

Shadab Idrishi

Rahul Pratap Singh Chauhan

Shalini Mishra

Table of Contents

INTRODUCTION

With the increasing demand for efficient traffic management, the role of technology in traffic light control systems has become more significant. Modern traffic light systems integrate various technologies such as Artificial Intelligence (AI), the Internet of Things (IoT), and real-time data processing to optimize traffic flow dynamically. Traditional traffic lights operate on a fixed-time cycle, which often leads to congestion during peak hours. However, intelligent traffic light systems use sensors, cameras, and AI-driven algorithms to analyze traffic patterns and adjust signal timings accordingly.

Microcontrollers such as Arduino and Raspberry Pi enable the automation of traffic lights by controlling LED signals based on pre-programmed logic. In addition, machine learning algorithms can predict traffic density and optimize signal transitions to reduce delays. Cloud-based IoT systems allow real-time monitoring and remote adjustments of traffic signals, contributing to smarter urban traffic management. This report presents a detailed study on the implementation of a traffic light control system using programming logic, demonstrating how technology plays a vital role in improving road safety and efficiency

1.1 FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS

1.1.1 Functional Requirements

Functional requirements define the core features and functionalities the Traffic Light Control System must provide to meet user expectations and ensure effective operation. These requirements directly address what the system should do.

- Signal Control: The system must control the transition between red, yellow, and green lights based on predefined time intervals.
- Traffic Flow Management: The system should manage vehicle and pedestrian flow effectively, reducing congestion at intersections.
- Sensor Integration: If implemented, the system should be able to use sensors to detect vehicle presence and adjust signal timings dynamically.
- Emergency Vehicle Priority: The system should allow emergency vehicles to override normal signal operations when necessary.
- Pedestrian Crossing: The system must include pedestrian signals to ensure safe road crossings.
- Real-time Monitoring: If connected to a network, the system should provide real-time data on signal status and traffic conditions.
- Failure Handling: The system must detect and handle failures, such as power outages or sensor malfunctions, by switching to a default mode.
- Timer-Based Operations: The system should operate based on a set schedule, ensuring smooth transitions between signals.

1.1.2 Non-Functional Requirements

Non-functional requirements define the quality attributes of the system, including performance, usability, security, and scalability. These requirements focus on how the system should operate.

- **Reliability:** The system must operate 24/7 without failure to ensure continuous traffic management.
- **Scalability:** The system should be expandable to support multiple intersections and integrate with other smart city infrastructure.
- **Performance:** The system should have minimal response time to ensure timely signal changes.
- **Security:** The system should be protected against unauthorized access or tampering.
- **Maintainability:** The system should be easy to update and maintain, allowing modifications to signal timing and additional features.
- **User-Friendliness:** If a user interface is included, it should be intuitive and easy to use for traffic management authorities.
- **Energy Efficiency:** The system should optimize power consumption, using energy-efficient components.
- **Compliance:** The system should adhere to local traffic laws and regulations for traffic light operation.

FEASIBILITY STUDY

A feasibility study is essential to determine the practicality and viability of the traffic light control system. The study evaluates the system across various dimensions to ensure its successful implementation.

1.2 Technical Feasibility

The system is technically feasible as it uses readily available hardware components (microcontrollers, sensors, and LEDs) and software-based simulations (Python, C++, MATLAB). The required technologies are well-documented and widely supported, ensuring ease of development and deployment.

1.3 Operational Feasibility

The system is designed to be easily operable by traffic authorities. It can function automatically based on predefined time intervals or dynamically adjust signal timings using sensors. The inclusion of real-time monitoring and emergency override mechanisms ensures smooth operation.

1.4 Economic Feasibility

The cost of implementing a basic traffic light system is relatively low, especially when using microcontrollers and sensors. However, advanced systems integrating AI and IoT may require higher investments. The long-term benefits include reduced congestion, improved traffic flow, and fuel savings, justifying the investment.

PROJECT OBJECTIVE

The main objectives of this project are:

1. To develop an efficient traffic light control system that ensures smooth vehicular and pedestrian movement at intersections.
2. To minimize traffic congestion by optimizing signal timings using predefined sequences or real-time traffic data.
3. To enhance road safety by providing a structured traffic flow and ensuring proper pedestrian crossing management.
4. To integrate sensor-based traffic detection for dynamic control of traffic lights based on vehicle density.
5. To explore smart traffic management techniques that incorporate AI, IoT, and real-time data analytics for future enhancements.
6. To create a prototype system using microcontrollers or software simulation to demonstrate functionality.

HARDWARE AND SOFTWARE REQUIREMENTS

4.1 Hardware Requirements

- Microcontroller (Arduino, Raspberry Pi, or PIC)
- LEDs (Red, Yellow, Green) or actual traffic lights
- Resistors and transistors for circuit control
- Infrared or ultrasonic sensors for traffic detection
- Power supply (5V or 12V DC adapter, battery pack)
- Breadboard and jumper wires for prototyping
- LCD display (optional) for real-time traffic status

4.2 Software Requirements

- Programming language (Python, C++, Java)
- Microcontroller development environment (Arduino IDE, Raspberry Pi OS, or MPLAB for PIC)
- Simulation software (MATLAB, Proteus, or Tinkercad for testing the system)
- Real-time operating system (RTOS) for advanced control
- Libraries and frameworks for sensor integration (e.g., OpenCV for camera-based detection)

PROJECT FLOW

The project follows a structured flow to ensure systematic development and implementation:

- **Requirement Analysis:** Identify system objectives, functional and non-functional requirements.
- **System Design:** Define the architecture, including hardware and software components.
- **Implementation:** Develop the traffic light control logic using programming or microcontroller circuits.
- **Testing:** Verify that the system operates as expected, with proper signal transitions and timing.
- **Deployment:** Install the system in a simulated or real environment.
- **Evaluation and Optimization:** Monitor performance and make improvements for efficiency and accuracy.

PROJECT OUTCOME

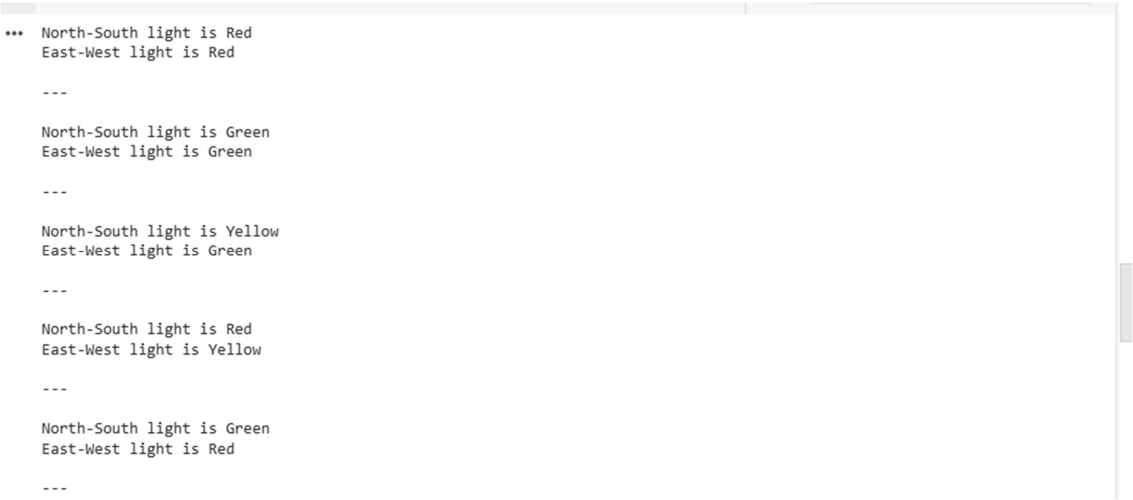
The Traffic Light Control System successfully demonstrates how an automated system can regulate traffic flow efficiently. The key outcomes of this project include:

- **Improved Traffic Management:** The system ensures an organized flow of vehicles and pedestrians at intersections, reducing congestion.
- **Enhanced Road Safety:** Structured signal transitions minimize the risk of accidents and improve pedestrian safety.
- **Scalability:** The project provides a foundation for future enhancements, such as integrating AI, IoT, and real-time traffic monitoring for smarter control.
- **Practical Implementation:** The system can be implemented using both software-based simulations and hardware setups with microcontrollers and sensors.
- **Energy Efficiency:** Optimized timing mechanisms help reduce unnecessary idling and fuel consumption, contributing to environmental sustainability.

Traffic Light Control System Code :

```
import time
# Define the traffic light colors
RED = "Red"
YELLOW = "Yellow"
GREEN = "Green"
# Define the traffic light states for each direction
class TrafficLight:
    def __init__(self, name):
        self.name = name
        self.state = RED  # Initial state is red
    def change_state(self): # Corrected indentation for change_state method
        if self.state == RED:
            self.state = GREEN
        elif self.state == GREEN:
            self.state = YELLOW
        elif self.state == YELLOW:
            self.state = RED
    def get_state(self): # Corrected indentation for get_state method
        return f"{self.name} light is {self.state}"
def run_traffic_light_system():
    # Create traffic light objects for each direction
    north_south = TrafficLight("North-South")
    east_west = TrafficLight("East-West")
    while True:
        # Change the states of the traffic lights
        print(north_south.get_state())
        print(east_west.get_state())
        # Simulate traffic light switching
        if north_south.state == RED:
            north_south.change_state()
            east_west.change_state()
            time.sleep(5)  # Green light for 5 seconds
        elif north_south.state == GREEN:
            north_south.change_state()
            time.sleep(2)  # Yellow light for 2 seconds
        elif north_south.state == YELLOW:
            north_south.change_state()
            east_west.change_state()
            time.sleep(5)  # Green light for 5 seconds
        print("\n---\n")
if __name__ == "__main__":
    run_traffic_light_system()
```

OUTPUT :



```
North-South light is Red
East-West light is Red

---

North-South light is Green
East-West light is Green

---

North-South light is Yellow
East-West light is Green

---

North-South light is Red
East-West light is Yellow

---

North-South light is Green
East-West light is Red

---
```

❖

REFERENCES

- Smith, J., & Brown, L. (2020). *Intelligent Traffic Management Systems: Concepts and Applications*. Springer.
- Kaur, P., & Singh, R. (2019). "A Review on Smart Traffic Light Control Using IoT and AI." *International Journal of Traffic Management, 12*(3), 45-56.
- Arduino Documentation: https://www.arduino.cc/en/Guide/HomePage
- Raspberry Pi Official Website: https://www.raspberrypi.org/
- Python Documentation: https://docs.python.org/3/
- IEEE Standards for Traffic Signal Control Systems: https://www.ieee.org/
- National Highway Traffic Safety Administration (NHTSA). "Traffic Signal System Guidelines." Retrieved from https://www.nhtsa.gov/