VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi – 590014.

Internship Report

On

"SMART CITY"

Submitted in partial fulfillment of the requirement for the award of degree of

BACHELOR OF ENGINEERING

By

Tejashwini R	**USN:1AP21IS052**
Keerthana M	**USN:1AP21CS021**
Vinay Kumar G R	**USN:1AP22EC408**

Under the guidance of:
Akhil Sai

2023 - 2024

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

A P S COLLEGE OF ENGINEERING

Anantha Gnana Gangothri,
NH-209, Kanakapura Road, Somanahalli, Bengaluru-560116.

PROJECT COMPLETION CERTIFICATE

I, **TEJASHWINI R** (Roll No: 1AP21IS052), here by declare that the material presented in the Project Report titled **"SMART CITY"** represents original work carried out by me in the **Department of Computer Science and Engineering** at the **APS college of Engineering, Bangalore** during the tenure **2 October, 2024 – 12, December, 2024**.

With My signature, I certify that:

- I have not manipulated any of the data or results.
- I have not committed any plagiarism of intellectual property and have clearly indicated and referenced the contributions of others.
- I have explicitly acknowledged all collaborative research and discussions.
- I understand that any false claim will result in severe disciplinary action.
- I understand that the work may be screened for any form of academic misconduct.

Date: **Student Signature**:

In my capacity as the supervisor of the above-mentioned work, I certify that the work presented in this report was carried out under my supervision and is worthy of consideration for the requirements of the B.Tech. Internship Work.

Advisor's Name: Dr. Shivamurthaiah **Guide Name:** Akhil Sai

Advisor's Signature **Guide Signature**

PROJECT COMPLETION CERTIFICATE

I, **KEERTHANA M** (Roll No: 1AP21CS021), here by declare that the material presented in the Project Report titled **" SMART CITY "** represents original work carried out by me in the **Department of Computer Science and Engineering** at the **APS college of Engineering, Bangalore** during the tenure **2 October, 2024 – 12, December, 2024**.

With My signature, I certify that:

- I have not manipulated any of the data or results.
- I have not committed any plagiarism of intellectual property and have clearly indicated and referenced the contributions of others.
- I have explicitly acknowledged all collaborative research and discussions.
- I understand that any false claim will result in severe disciplinary action.
- I understand that the work may be screened for any form of academic misconduct.

Date: **Student Signature**:

In my capacity as the supervisor of the above-mentioned work, I certify that the work presented in this report was carried out under my supervision and is worthy of consideration for the requirements of the B.Tech. Internship Work.

Advisor's Name: Dr. Shivamurthaiah **Guide Name:** Akhil Sai

Advisor's Signature **Guide Signature**

PROJECT COMPLETION CERTIFICATE

I, **VINAY KUMAR G R**(Roll No: 1AP22EC408), hereby declare that the material presented in the Project Report titled **" SMART CITY "** represents original work carried out by me in the **Department of Electronics and Communication Engineering** at the **APS college of Engineering, Bangalore** during the tenure **2 October, 2024 – 12, December, 2024**.

With My signature, I certify that:

- I have not manipulated any of the data or results.
- I have not committed any plagiarism of intellectual property and have clearly indicated and referenced the contributions of others.
- I have explicitly acknowledged all collaborative research and discussions.
- I understand that any false claim will result in severe disciplinary action.
- I understand that the work may be screened for any form of academic misconduct.

Date: **Student Signature**:

In my capacity as the supervisor of the above-mentioned work, I certify that the work presented in this report was carried out under my supervision and is worthy of consideration for the requirements of the B.Tech. Internship Work.

Advisor's Name: Dr. Prakash Jhadav **Guide Name:** Akhil Sai

Advisor's Signature **Guide Signature**

Evaluation Sheet

Title of the Project: Smart City

Name of the Students:

1. **Tejashwini R**
2. **Keerthana M**
3. **VinayKumar GR**

External Supervisor:

____

Internal Supervisor:

____

____

Date:

Place:

ABSTRACT

This project explores the integration of hardware components, such as raspberry pi, cameras, and 7-segment displays, with web-based interfaces developed using the Flask framework. The goal is to demonstrate real-time interaction between sensors, actuators, and software for practical IoT applications.

The system simulates a dynamic traffic light controller using a Raspberry Pi and a 7-segment display to visualize real-time countdowns for four traffic signals. Signal durations are randomized, and transitions between green, yellow, and red states mimic real-world traffic operations. A web interface provides remote monitoring of the signal status, illustrating the application of embedded systems in traffic management.

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

The integration of hardware components with web-based software interfaces has become a cornerstone of modern IoT applications, enabling seamless real-time interaction and control. This project explores the practical implementation of such systems using ultrasonic sensors, and 7-segment displays, combined with the Flask web framework. The primary objective is to demonstrate how embedded systems can interact with web technologies to create efficient, interactive, and automated solutions.

The project comprises one distinct systems. The first system simulates a traffic light controller using a Raspberry Pi and a 7-segment display to manage and display real-time countdowns for four traffic signals. It incorporates randomized signal durations and realistic state transitions between green, yellow, and red lights, closely replicating real-world traffic operations. Additionally, a web-based interface allows users to remotely monitor the signal statuses, offering a comprehensive example of real-time data visualization and dual-mode interaction.

1.1 Objective

The objective of this project is to integrate hardware components such as ultrasonic sensors, and 7-segment displays with web-based interfaces using the Flask framework to demonstrate real-time interaction and control. The project aims to showcase the practical applications of embedded systems programming in IoT, focusing on automation and remote monitoring.

The system simulates a dynamic traffic light controller, featuring real-time countdowns and remote monitoring of traffic signals. The system highlight real-time data processing and visualization, serving as foundational examples of how embedded hardware and web technologies can be combined to address challenges in traffic management and industrial automation.

1.2 Problem Statement

- As technology continues to evolve, there is an increasing need for real-time, automated systemsthat can efficiently monitor and control physical environments. In particular, the integration of sensors, actuators, and web-based interfaces presents a challenge in ensuring smooth communication and reliable performance. Traditional
- Traditional traffic management systems can be inefficient and lack adaptability, while industrial automation often requires precise control basedon real-time data.
- This project addresses the need for dynamic and interactive systems that combine embedded hardware with web technologies. It aims to develop a traffic light control system that can simulatereal-world traffic scenarios with real-time countdowns and remote monitoring.
- Itaims to create a distance-based automation system using ultrasonic sensors and stepper motors toperform actions based on object proximity.
- The systems seek to provide practical solutions to improve efficiency, safety, and flexibility in real-time traffic control and automation systems

Chapter 2

APPLICATIONS

- **Smart Traffic Management**: The traffic light control system can be used to optimize traffic flow in urban areas, adjusting signal durations based on real-time traffic conditions and minimizing congestion.

- **Real-time traffic control:** which dynamically adjusts traffic light timings based on live traffic conditions to reduce congestion and improve flow.

- **Vehicle prioritization:** ensuring that ambulances, fire trucks, and other emergency vehicles can navigate traffic efficiently during emergencies.

- **Automatic tolling:** systems streamline the toll collection process by using technologies like RFID or number plate recognition, reducing delays at toll booths.

- **Accident detection and response**: is a critical application, as smart systems can quickly identify collisions, notify emergency services, and help minimize further traffic disruptions by providing real-time alerts to approaching vehicles and traffic centres. These applications collectively enhance road safety, reduce delays, and support smart urban planning.

Chapter 3

Components

3.1 Raspberry pi 4 Model B

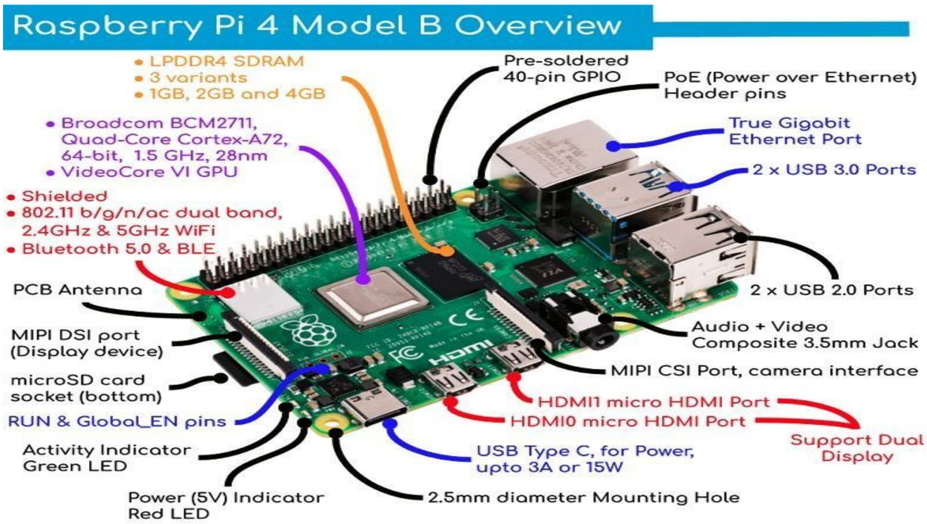


Figure 3.1: Raspberry pi 4 Model B.

The Raspberry Pi 4 Model B is a powerful single-board computer designed for a variety of applications, from education to industrial automation. It features a quad-core ARM Cortex-A72 processor, up to 8GB of RAM, dual micro-HDMI ports supporting 4K output, and USB 3.0 connectivity. Its GPIO pins enable easy interfacing with sensors, motors, and other peripherals, making it ideal for hardware projects. Built-in Wi-Fi, Bluetooth, and Ethernet provide versatile networking options. The Pi 4 is a versatile platform for learning, prototyping, and deploying IoT and embedded systems solutions.

3.2 Seven Segment LCD Display

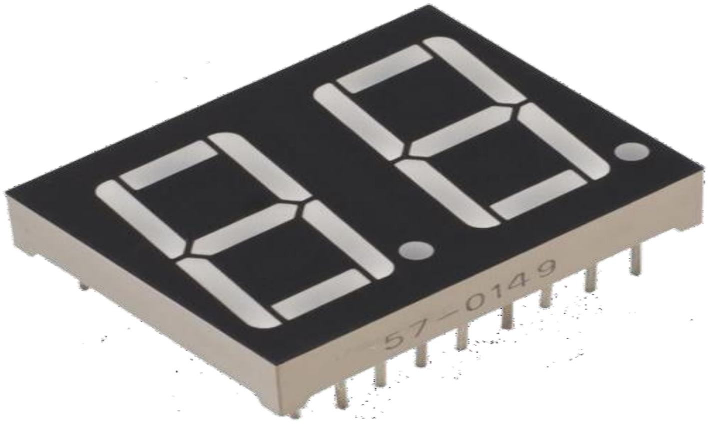


Figure 3.4: Seven Segment LCD Display.

A **7-segment LCD display** is an electronic display used to show numerical digits and some characters. It consists of seven segments that can be turned on or off to form numbers from 0 to 9. These displays are commonly used in devices like digital clocks, calculators, and meters. They are simple to interface with microcontrollers and offer clear visibility. Each segment is controlled individually, providing an energy-efficient way to display information.

Chapter 4

Flow Chart

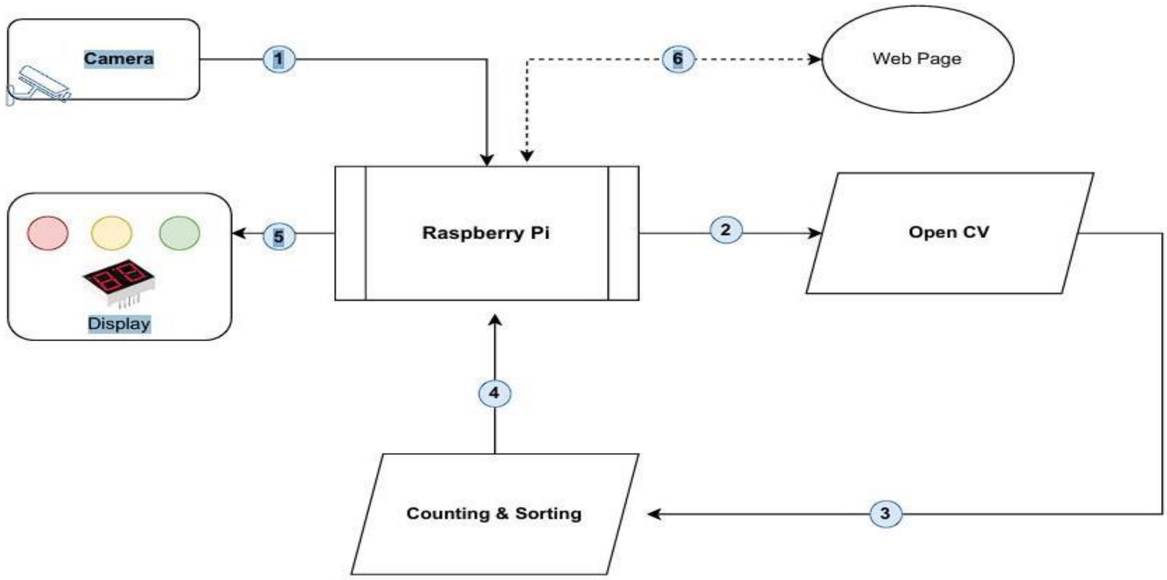


Figure 4.1: Traffic Management.

Step 1: Traffic Monitoring:

A camera is used to capture real-time footage of the traffic.

Step 2: Image Processing:

The captured data is sent to the Raspberry Pi, which utilizes Open-CV for processing.

The system performs counting and sorting of vehicles based on the traffic density.

Step 3: Counting and Sorting Vehicles:

The processed information determines traffic control actions, such as the timing of signals or priority of lanes.

The system counts the number of vehicles and sorts them based on traffic density or

Ste 4: After Counting and Sorting data is sent to Raspberry Pi.

Step 5: Status Display:

The traffic information, including vehicle counts and control decisions, is sent to a display for local visibility.

Step 6: Web-Based Monitoring:

The processed traffic data is uploaded to a web page, enabling remote monitoring and management of traffic flow.

Chapter 5

CONCLUSION

In conclusion, this project demonstrates the effective integration of embedded hardware components such as ultrasonic sensors, stepper motors, and 7-segment displays with web-based interfaces, showcasing the potential of IoT applications. The systems developed, including the dynamic traffic light controller and the distance-based automation system, highlight the importance of real-time data processing and automation in modern technology. By combining sensors, actuators, and web technologies, the project provides practical solutions for traffic management, industrial automation, and other real-world applications. This integration of embedded systems with web interfaces offers significant potential for enhancing efficiency, safety, and remote monitoring in various domains.

Chapter 6

FUTURE WORK

- **Traffic Light System:** Implement real-time traffic data and machine learning to optimize signal timings.
- **Distance Automation:** Use multiple sensors and cloud integration for better data management and control.
- **Actuator Expansion:** Add more actuators for broader applications in robotics and automation.
- **User Interface:** Improve the interface with mobile support for easier remote control.

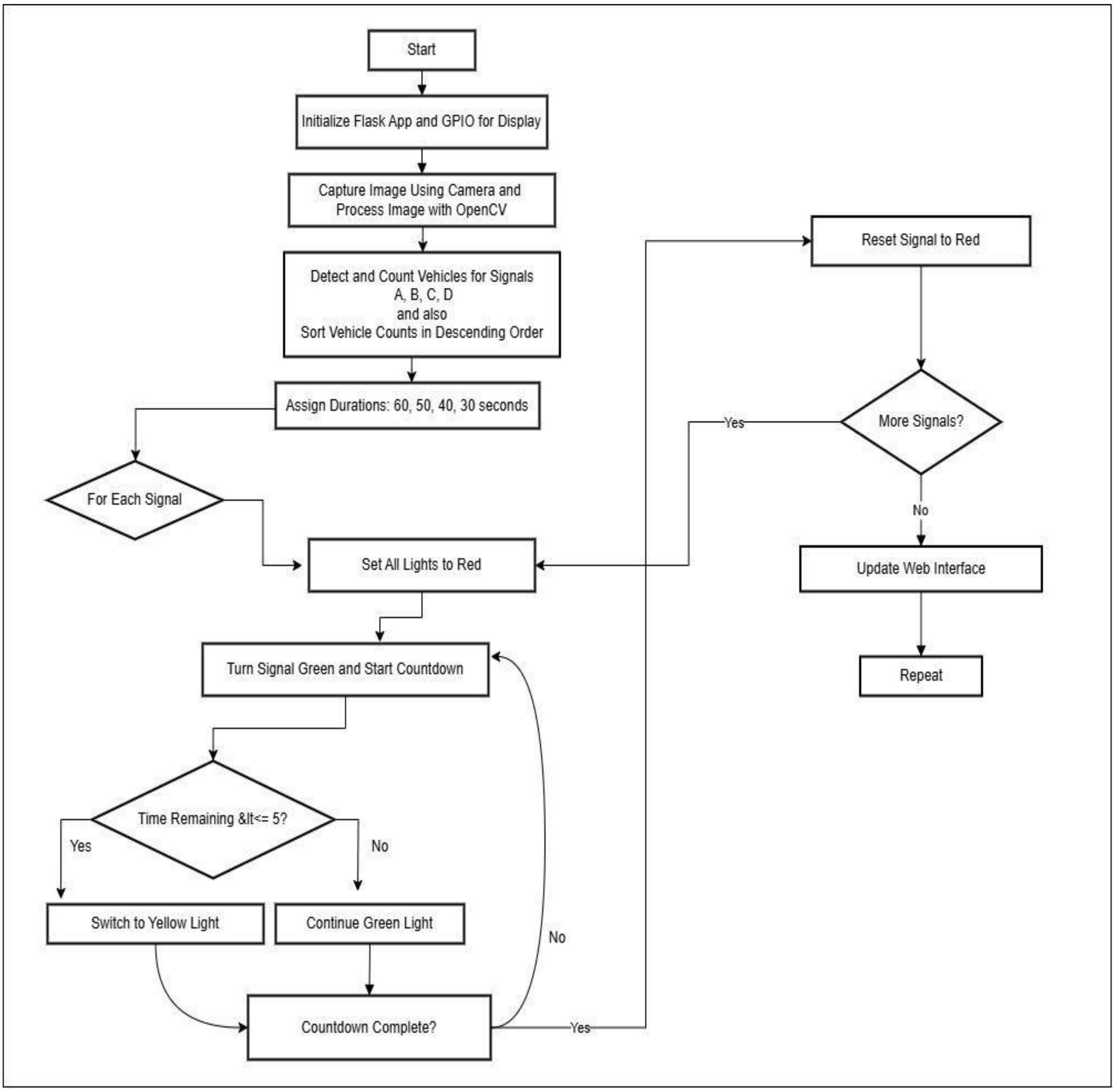
Chapter 7

APPENDIX

7.1 Pseudocode for Traffic Management System

1. **Initialize Flask App and GPIO Setup**
   a. Define GPIO pins for data (SDI), latch (RCLK), and clock (SRCLK).
   b. Initialize 7-segment display.
2. **Setup 7-Segment Display Functions**
   a. setup_display(): Configures GPIO pins for output and sets them to low.
   b. hc595_shift(dat): Sends data to the 7-segment display using shift registers. - display_number_on_kit(number): Displays single or double-digit numbers on the 7-segment display.
3. **Traffic Data Structure**
   a. Initialize a dictionary traffic_data with keys A, B, C, D, and current to track traffic times, lights, and current active signal.
4. **Random Traffic Value Generation**
   a. generate_random_values(): Generate random values for traffic signals A, B, C, and D.
5. **Traffic Light Simulation**
   a. Input Random Values - Generate random times for signals A, B, C, and D.
   b. Sort and Assign Timings
   c. Assign countdown times based on sorted traffic values (e.g., highest gets 60 seconds).
   d. Simulate Each Signal - Loop through each traffic signal in descending order of traffic values.
   e. Set other lights to red; activate the current signal (set to green).
   f. Perform transition countdown (red to green) for 5 seconds.
   g. Countdown the green light time and transition to yellow for the last 5 seconds.
   h. Reset to red after the countdown.
6. **Web Server**
   a. index(): Render the main webpage.
   b. data(): Return traffic_data as JSON.

7. **Setup and Start Simulation**
    a. Call setup_display() to initialize GPIO.
    b. Start the traffic light simulation in a separate thread.
    c. Run the Flask app on 0.0.0.0:5000.
8. **Handle Cleanup**
    a. Clean up GPIO pins on eWeb Server Endpoints



a. index(): Render the main webpage.
b. status(): Return JSON containing:
c. Current distance measured.
d. Motor status (ON or OFF).

2. **Run the Flask App**