**Smart City Project - Setup Guide**

This project simulates a smart city traffic management system using a Raspberry Pi, Flask for the web interface, and a 7-segment display for countdowns. It dynamically assigns traffic light timings based on random traffic conditions or sensor inputs and displays the data on a web dashboard. LEDs or GPIO-controlled devices simulate the traffic lights for four directions.

1. **Components and Tools Required**

**Hardware :**

* RaspberryPi
* 7-Segment Display with 74HC595 Shift Register.
* Traffic Lights Simulation.
* Power Supply.
* Ultra Sonic Sensor.
* Stepper Motor.

The Raspberry Pi and Arduino models along with necessary sensors for the project.

**Software :**

* Python(3.10 or 3.11)
* Other python libraries(will be mentioned in installation.
* VS Code on laptop and RaspberryPi OS on raspberryPi.
* Set Up the Flask Application.

**2.** **Libraries Used**

* Flask: Used for creating the web application and serving HTML templates and JSON data.
* RPi.GPIO: Used for interacting with the GPIO pins on the Raspberry Pi to control hardware components like LEDs and 7-segment displays.
* threading: Used to run the traffic light simulation in a separate thread to keep the Flask web server responsive.
* time: Used for adding delays, timing the countdowns, and controlling the flow of the simulation.
* random: Used for generating random traffic light timings for simulation purposes.

**Configuration Steps for Smart City Traffic**

* + Install the Raspberry Pi OS.
  + Install Python and Required Libraries.
  + Connect the Hardware.
  + Run the Flask App.
    - python3 smart\_city.py
  + Access the Web Interface.

http://<Raspberry\_Pi\_IP>:5000

* **GPIO Pin Configuration**: Double-check the GPIO pin numbers in your Python script against your wiring to ensure correctness.
* **Test Hardware**: Run simple scripts to test each component (e.g., stepper motor and sensor) before integrating them into the full project.
* **Power Supply**: Make sure the Raspberry Pi and all connected devices have a stable power source.
* **Web Server Security**: For more secure deployments, consider setting up password protection or using a reverse proxy.

These steps will help you configure and set up project successfully.

**Snapshots**

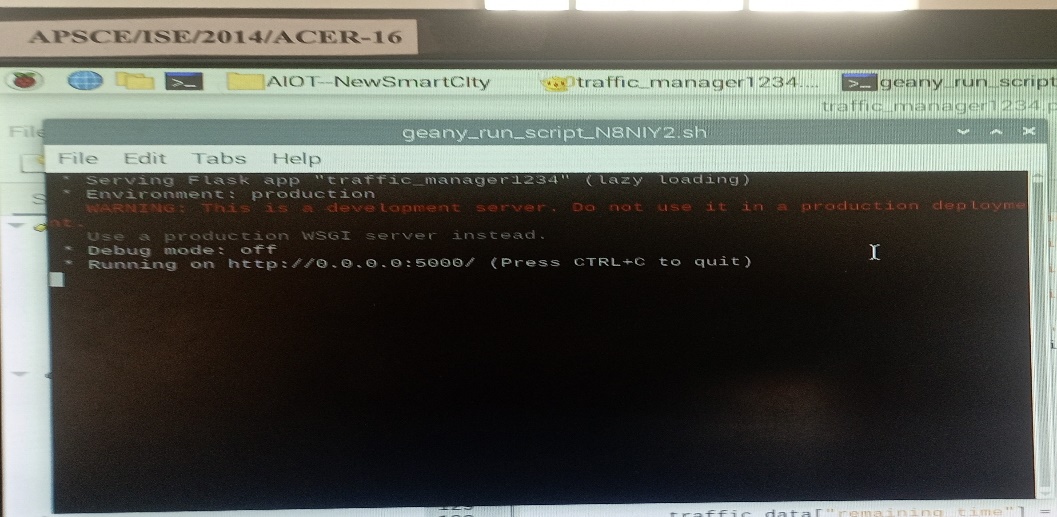


Fig 1 Running the traffic\_manager1234.py

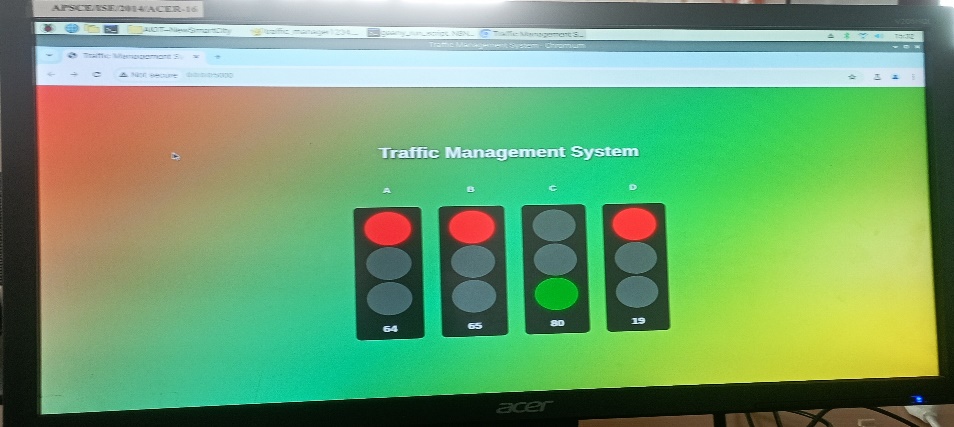


Fig 2. Web Page

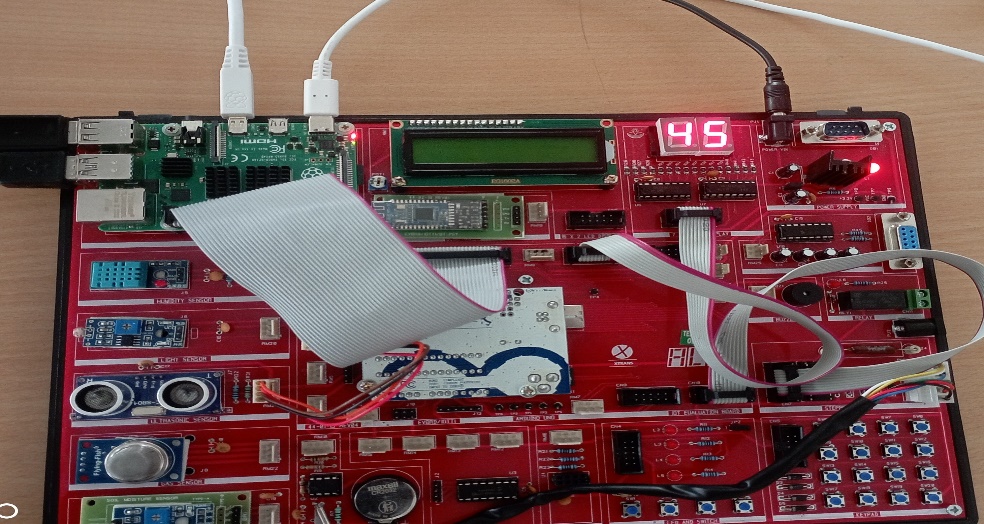
****

Fig 3 Raspberry pi kit and 7segment lcddisplay