Traffic Management System

[Phase 3 - Development Part 1]

M.Thirumalaiselvi - 952621106015
S.Veerasamy Chettiar College of Engineering and Technology - 9526 (Puliyankudi)

Introduction:

In today's increasingly connected world, the use of IoT (Internet of Things) technology to manage and monitor various aspects of our environment has become a pivotal aspect of smart city initiatives. One such application is the implementation of an IoT-enabled traffic management system, which leverages the power of microcontrollers, sensors, and network communication to efficiently control traffic flow. In this project, we've explored the development of an IoT traffic management system using an ESP32 board and MicroPython within the Wokwi simulator.

Procedure:

Step 1: Set Up Wokwi Account

- If you haven't already, sign in to your Wokwi account (https://wokwi.com/).

Step 2: Create a New Project

- Click on "My Projects" and then "New Project" to create a new project.

Step 3: Add Components to the Circuit

- In the Wokwi simulator, you need to add the components for your traffic management project:
- ESP32: Search for "ESP32" in the Components Library, and add it to your project.
- Traffic Light: You can create a simple traffic light using LEDs (e.g., red, yellow, green) and resistors. Add these components to your project.

Step 4: Connect the Components

- Use the "Add Wire" option to establish the necessary connections between the ESP32 and the LEDs to create a traffic light setup. For example, connect the ESP32 pins to the LEDs' anodes and cathodes.
- You may need to use resistors to limit current for the LEDs, depending on your LED specifications.

Step 5: Add MicroPython Code

- Click on the "Code" button in the left sidebar to access the code editor.
- Write the MicroPython code for the traffic light control system. You can use the code I provided in a previous response and adapt it to work in the Wokwi simulator.
- Configure the Wi-Fi and MQTT settings to simulate data transmission. Ensure the code controls the LEDs to simulate a traffic light sequence.

Step 6: Configure Virtual Components

- Click on the "Components" tab on the left sidebar.

- Configure the properties of each component, such as specifying the LED colors for the traffic light and resistor values.

Step 7: Set Up Simulation Parameters

- Click on the “Simulation” tab on the left sidebar.
- Configure the simulation parameters, such as the simulation speed and simulation duration.

Step 8: Start the Simulation

- Click the “Start Simulation” button to run your project in the Wokwi simulator.
- You will be able to see the traffic light simulation and observe how the ESP32 controls the LEDs in the virtual traffic light.

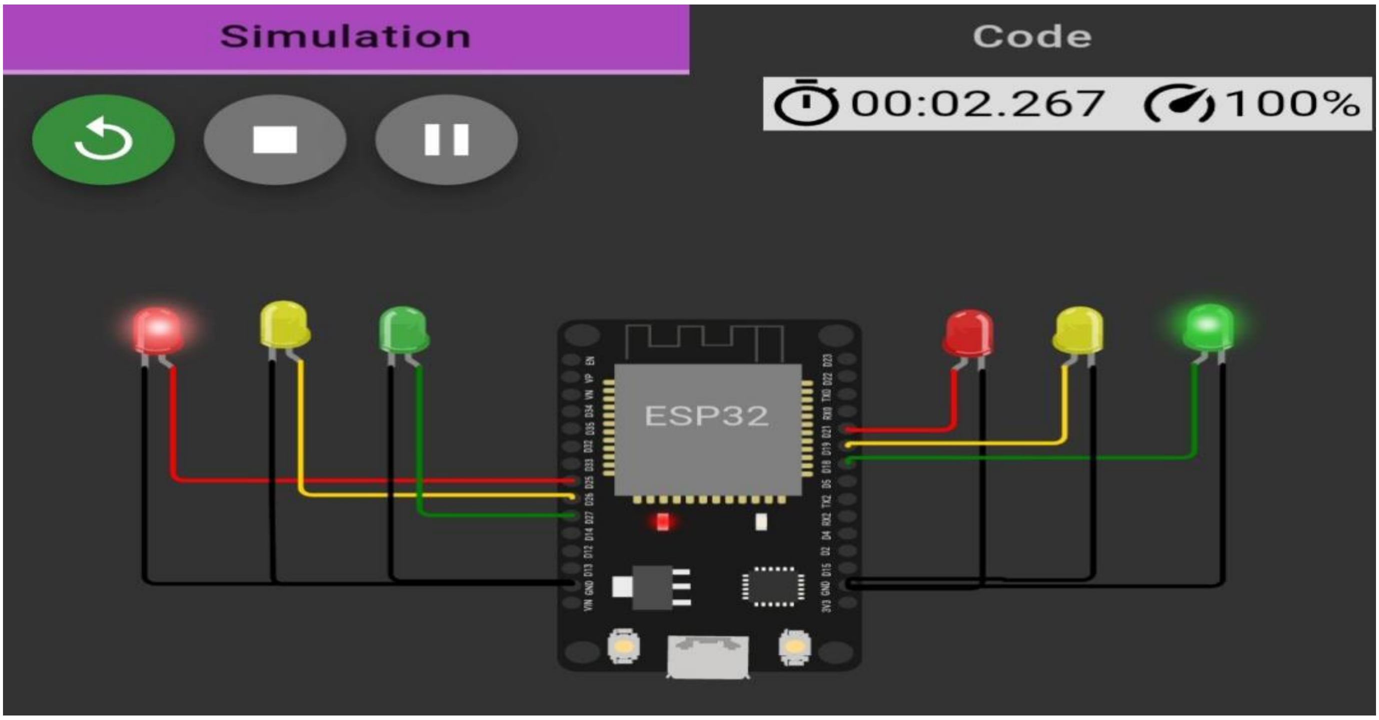
Step 9: Monitor and Test

- While the simulation is running, you can monitor the behavior of the traffic lights and test the traffic management system’s functionality.
- Use the console and visualization features in Wokwi to see the results of your code.

Step 10: Debug and Experiment

- If you encounter issues or want to experiment with different scenarios, you can make adjustments to your code and components and rerun the simulation.
- By following these steps, you can create a traffic management project in Wokwi using MicroPython and simulate a traffic light control system with ESP32. Wokwi provides a convenient and interactive environment for simulating and testing your IoT projects.

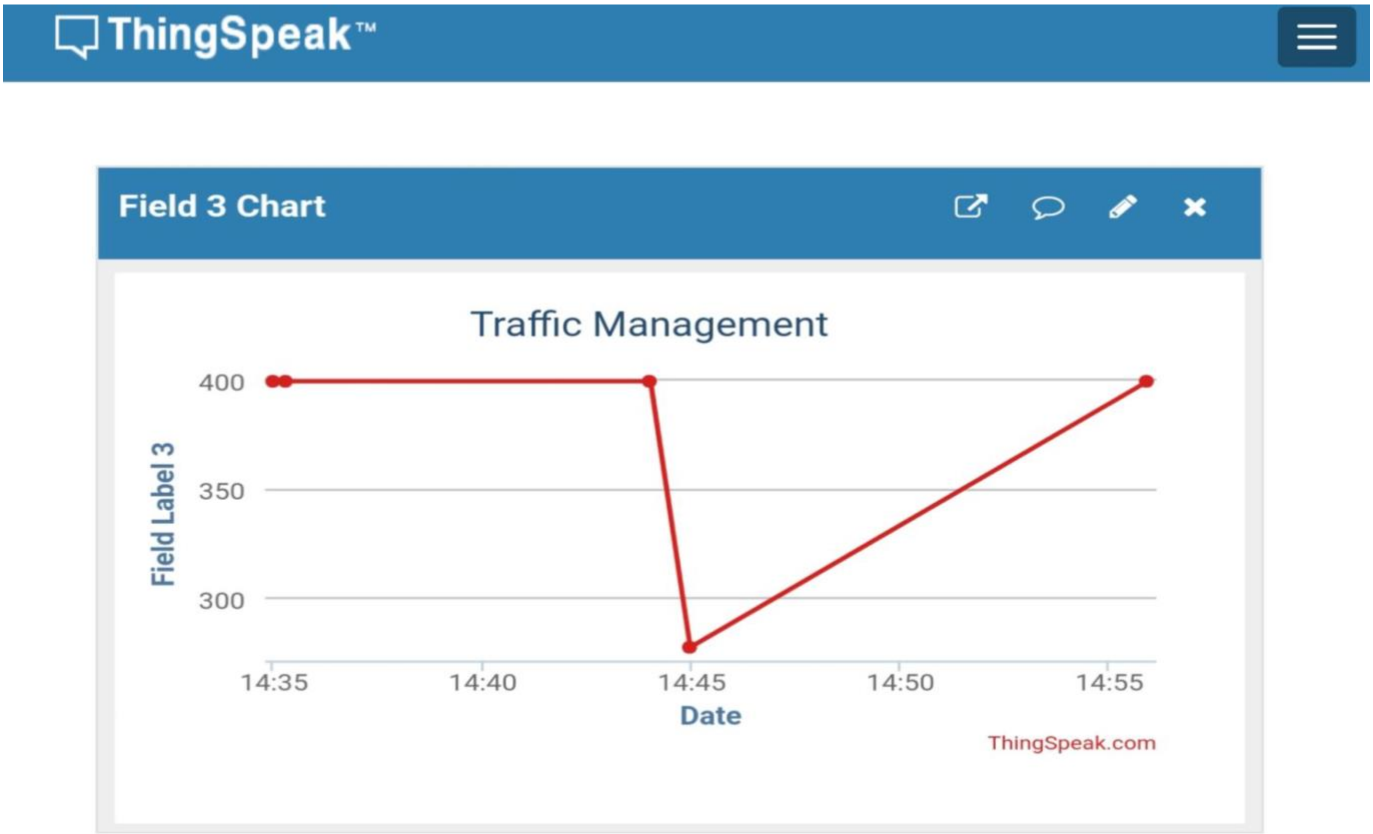
Simulation:



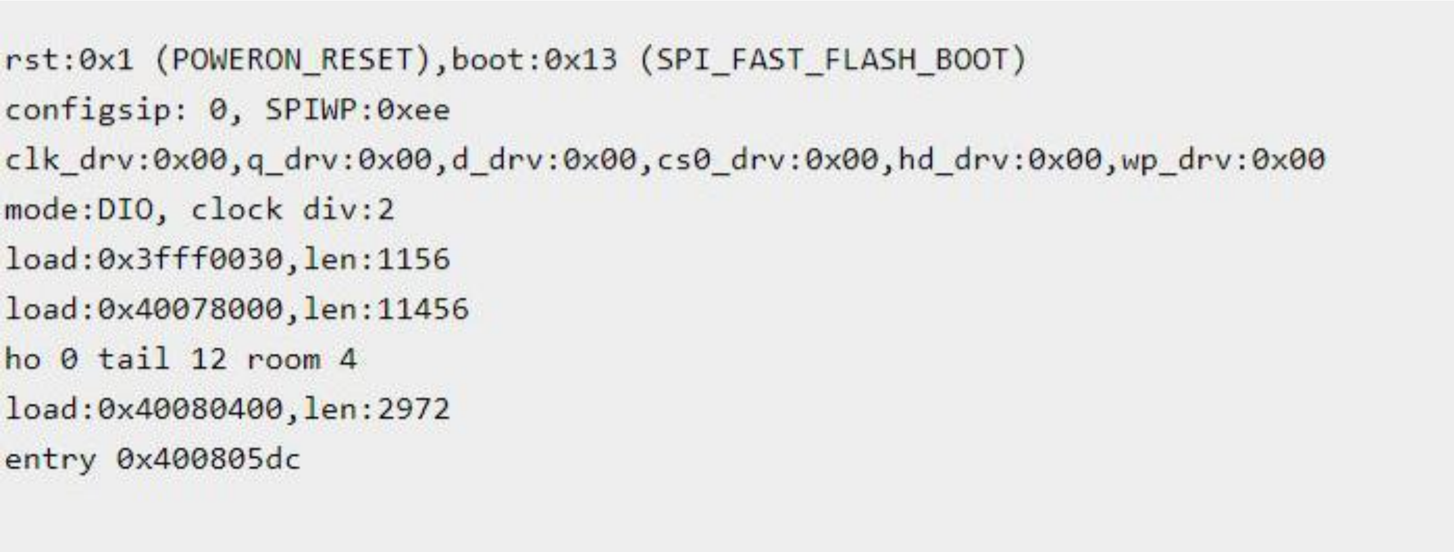
Source code :

```
Void setup()
 {
Serial.begin(115200);
pinMode(25,OUTPUT);
pinMode(26,OUTPUT);
pinMode(27,OUTPUT);
pinMode(21, OUTPUT);
pinMode(19, OUTPUT);
pinMode(18, OUTPUT);
}
Void loop()
 {
digitalWrite(25, HIGH);
digitalWrite(18, HIGH);
delay(3000);
digitalWrite(25, LOW);
digitalWrite(18, LOW);
delay(3000);
digitalWrite(26, HIGH);
digitalWrite(19, HIGH);
delay(3000);
digitalWrite(26, LOW);
digitalWrite(19, LOW);
delay(3000);
digitalWrite(27, HIGH);
digitalWrite(21, HIGH);
delay(3000);
digitalWrite(27, LOW);
digitalWrite(21, LOW);
delay(3000);}
```

Real time output:



Source Source output:



```
rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:2
load:0x3fff0030,len:1156
load:0x40078000,len:11456
ho 0 tail 12 room 4
load:0x40080400,len:2972
entry 0x400805dc
```

Conclusion:

The IoT traffic management project within the Wokwi simulator showcases the versatility and potential of IoT devices for real-world applications. By implementing a virtual traffic light control system, we’ve illustrated how an ESP32 board running MicroPython can effectively simulate traffic light sequences. This not only provides valuable insights into traffic management strategies but also serves as a foundation for real-world implementations.