Density Based Traffic Light Controlling

Vemula Haran Ritvick CS21B1033

I. AIM

The aim of this project is to revolutionize traditional traffic signal management systems by introducing a dynamic and responsive approach to traffic control. By leveraging the capabilities of the ESP32 micro-controller and ultrasonic sensors, the project seeks to develop a smart traffic management system that can adaptive adjust signal timings based on real-time traffic conditions. The primary objective is to optimize traffic flow, minimize waiting times at intersections, and enhance overall urban mobility and efficiency.

II. Apparatus/Software Required

Hardware:

1) ESP32 WiFi Module (1): The ESP32 serves as the central processing unit for the traffic management system, responsible for reading data from ultrasonic sensors and controlling traffic signals.
2) Ultrasonic Sensor HC-SR04 (4): These sensors detect the presence of vehicles at each intersection by measuring the distance to the nearest object.
3) Red, Yellow, and Green LED's (4 each): These LED's are used to indicate the status of traffic signals at each intersection.
4) Connecting Wires: Wires are used to establish connections between components on the breadboard.
5) Micro-USB Cable: The USB cable is used to power the ESP32 module.
6) Breadboard: Provides a platform for assembling and connecting the hardware components.

Software:

1) Blynk Cloud Platform: The Blynk platform is utilized for real-time monitoring and control of the traffic management system. It enables users to remotely monitor traffic conditions and adjust signal timings as needed.

III. Procedure

A. Introduction

The traditional traffic signal management systems often face challenges in efficiently regulating traffic flow, leading to congestion and delays. The Density-Based Traffic Light Controller project seeks to revolutionize this by introducing a dynamic and adaptive approach to traffic control. Through the integration of ESP32 micro-controllers, ultrasonic sensors, and the Blynk Cloud platform, the project aims to create a sophisticated traffic management system capable of real-time optimization. By harnessing data on traffic density and patterns, this innovative system promises to enhance road safety, reduce commute times, and minimize environmental impact, ultimately contributing to more sustainable and efficient urban mobility.

B. Methodology/Problem Statement

The project addresses the problem of inefficient traffic signal management in urban areas, where fixed timing schedules often lead to unnecessary delays and congestion at intersections. By developing a smart traffic management system that dynamically adjusts signal timings based on real-time traffic conditions, the project aims to mitigate these issues and improve overall traffic efficiency.

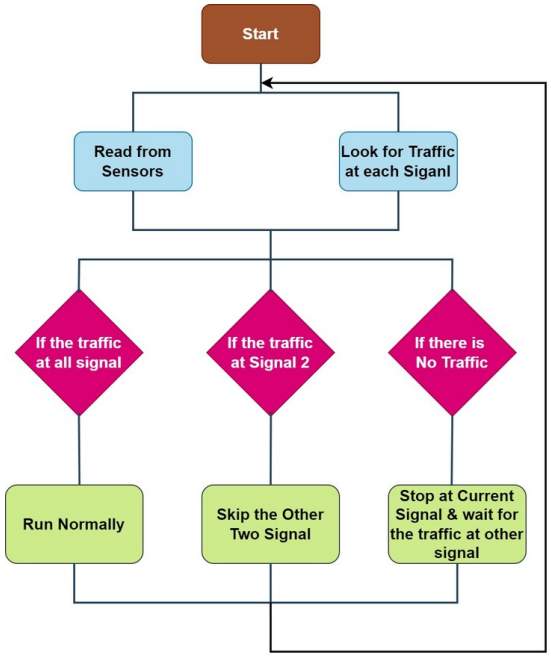

Fig. 1: Algorithm Flow Chart

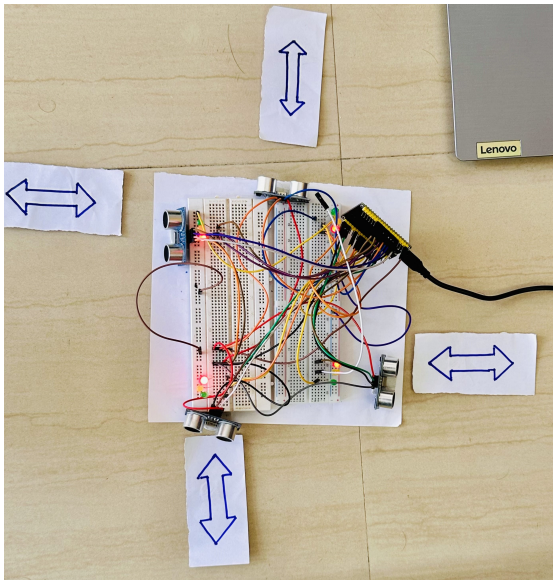

Fig. 2: Hardware Setup (Top view)

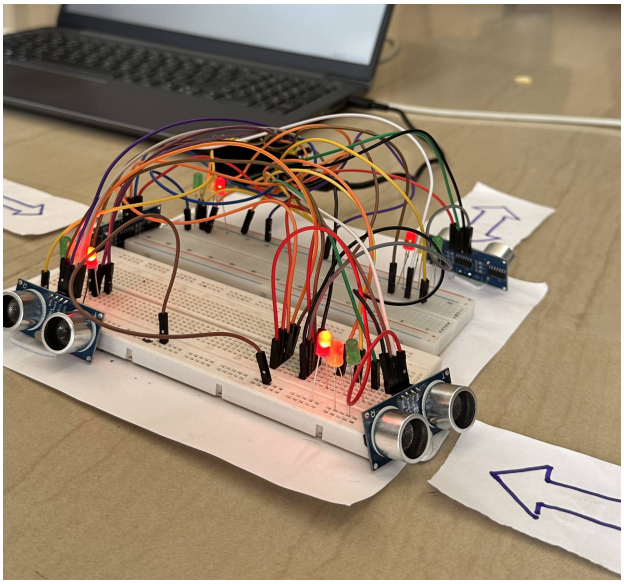

Fig. 3: Hardware Setup (Side view)

C. Implementation

1) Hardware Design: In the hardware design phase, meticulous attention is given to connecting the ESP32 micro-controller, ultrasonic sensors, and LED's on a breadboard, adhering closely to the provided circuit diagram. Each component such as ultrasonic sensors, LED's are strategically placed and wired to ensure seamless communication and accurate data transmission. The 5V power supply is distributed to power all components adequately, while meticulous attention is paid to grounding and signal routing to avoid interference and ensure proper functionality in various environmental conditions.

2) Executable Code:

```
#define BLYNK_TEMPLATE_ID "TMPL3nZ3AdpkD"
#define BLYNK_TEMPLATE_NAME "traffic light controlling"
#define BLYNK_AUTH_TOKEN "VKsYczcok_wb2hwzsExMWRdKXetqjt2s"

#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>

char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "vishnu";
char pass[] = "vishnu@2003";

// GPIO pins for each traffic signal light (red, yellow, green)
int signal1[] = {26, 25, 33};
int signal2[] = {19, 18, 5};
int signal3[] = {32, 3, 21};
int signal4[] = {17, 16, 15};

// Timing delays for light changes (milliseconds)
int redDelay = 5000;
int yellowDelay = 2000;

// Ultrasonic sensor pins
int triggerpin1 = 13;
int echopin1 = 12;
int triggerpin2 = 27;
int echopin2 = 14;
int triggerpin3 = 22;
int echopin3 = 23;
int triggerpin4 = 2;
int echopin4 = 4;

long duration; // Time for ultrasonic signal to return
int S1, S2, S3, S4; // Distance measurements from each sensor

int t = 5; // Threshold distance to check for traffic presence

void setup()
{
  Serial.begin(115200);
  Blynk.begin(auth, ssid, pass);

  // Initialize traffic signal LEDs as outputs
  for (int i = 0; i < 3; i++)
  {
    pinMode(signal1[i], OUTPUT);
    pinMode(signal2[i], OUTPUT);
    pinMode(signal3[i], OUTPUT);
```

```
    pinMode(signal4[i], OUTPUT);
  }

  // Initialize ultrasonic sensor pins
  pinMode(triggerpin1, OUTPUT);
  pinMode(echopin1, INPUT);
  pinMode(triggerpin2, OUTPUT);
  pinMode(echopin2, INPUT);
  pinMode(triggerpin3, OUTPUT);
  pinMode(echopin3, INPUT);
  pinMode(triggerpin4, OUTPUT);
  pinMode(echopin4, INPUT);
}

void loop()
{
  Blynk.run();

  // Measure distance from each ultrasonic sensor
  S1 = readDistance(triggerpin1, echopin1);
  S2 = readDistance(triggerpin2, echopin2);
  S3 = readDistance(triggerpin3, echopin3);
  S4 = readDistance(triggerpin4, echopin4);

  // Print the distances measured by each sensor
  Serial.print("Distance from Sensor 1: ");
  Serial.print(S1);
  Serial.println(" cm");
  Serial.print("Distance from Sensor 2: ");
  Serial.print(S2);
  Serial.println(" cm");
  Serial.print("Distance from Sensor 3: ");
  Serial.print(S3);
  Serial.println(" cm");
  Serial.print("Distance from Sensor 4: ");
  Serial.print(S4);
  Serial.println(" cm");

  // Check and update the status of each traffic signal
  if (S1 < t)
  {
    signal1Function();
  }
  else if (S2 < t)
  {
    signal2Function();
  }
  else if (S3 < t)
  {
    signal3Function();
```

```
  }
  else if (S4 < t)
  {
    signal4Function();
  }

  // Send updated signal status to Blynk
  updateBlynk();

  // Wait a bit before next loop iteration
  delay(1000);
}

int readDistance(int triggerPin, int echoPin)
{
  digitalWrite(triggerPin, LOW);
  delayMicroseconds(2);
  digitalWrite(triggerPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  return duration * 0.034 / 2; // Convert to cm
}

// Define functions for handling each signal
void signal1Function()
{
  controlSignal(signal1, V1, V2, V3);
}
void signal2Function()
{
  controlSignal(signal2, V4, V5, V6);
}
void signal3Function()
{
  controlSignal(signal3, V7, V8, V9);
}
void signal4Function()
{
  controlSignal(signal4, V10, V11, V12);
}

// Controls the signal light changes and updates Blynk
void controlSignal(int signalPins[], int vPinRed, int vPinYellow, int vPinGreen
{
  lowAll(); // Reset all signals to red before changing the current one
  digitalWrite(signalPins[0], HIGH); // Turn on red LED
  Blynk.virtualWrite(vPinRed, 255);
  delay(redDelay);
```

```
  digitalWrite(signalPins[0], LOW); // Turn off red LED
  Blynk.virtualWrite(vPinRed, 0);

  digitalWrite(signalPins[2], HIGH); // Turn on green LED
  Blynk.virtualWrite(vPinGreen, 255);
  delay(redDelay);

  digitalWrite(signalPins[2], LOW); // Turn off green LED
  Blynk.virtualWrite(vPinGreen, 0);

  digitalWrite(signalPins[1], HIGH); // Turn on yellow LED
  Blynk.virtualWrite(vPinYellow, 255);
  delay(yellowDelay);

  digitalWrite(signalPins[1], LOW); // Turn off yellow LED
  Blynk.virtualWrite(vPinYellow, 0);

  lowAll();
}


// Check if there is traffic at any signal
bool anyTraffic()
{
  return (S1 < t || S2 < t || S3 < t || S4 < t);
}

// Set all signals to red
void lowAll()
{
  for (int i = 0; i < 4; i++)
  {
    digitalWrite(signal1[0], HIGH);
    digitalWrite(signal1[1], LOW);
    digitalWrite(signal1[2], LOW);
    digitalWrite(signal2[0], HIGH);
    digitalWrite(signal2[1], LOW);
    digitalWrite(signal2[2], LOW);
    digitalWrite(signal3[0], HIGH);
    digitalWrite(signal3[1], LOW);
    digitalWrite(signal3[2], LOW);
    digitalWrite(signal4[0], HIGH);
    digitalWrite(signal4[1], LOW);
    digitalWrite(signal4[2], LOW);
  }
}

// Update Blynk with the current status of traffic signals
void updateBlynk()
{
```

```
  Blynk.virtualWrite(V1, digitalRead(signal1[0]) == HIGH ? 255 : 0);
  Blynk.virtualWrite(V2, digitalRead(signal1[1]) == HIGH ? 255 : 0);
  Blynk.virtualWrite(V3, digitalRead(signal1[2]) == HIGH ? 255 : 0);

  Blynk.virtualWrite(V4, digitalRead(signal2[0]) == HIGH ? 255 : 0);
  Blynk.virtualWrite(V5, digitalRead(signal2[1]) == HIGH ? 255 : 0);
  Blynk.virtualWrite(V6, digitalRead(signal2[2]) == HIGH ? 255 : 0);

  Blynk.virtualWrite(V7, digitalRead(signal3[0]) == HIGH ? 255 : 0);
  Blynk.virtualWrite(V8, digitalRead(signal3[1]) == HIGH ? 255 : 0);
  Blynk.virtualWrite(V9, digitalRead(signal3[2]) == HIGH ? 255 : 0);

  Blynk.virtualWrite(V10, digitalRead(signal4[0]) == HIGH ? 255 : 0);
  Blynk.virtualWrite(V11, digitalRead(signal4[1]) == HIGH ? 255 : 0);
  Blynk.virtualWrite(V12, digitalRead(signal4[2]) == HIGH ? 255 : 0);
}
```

3) Software Implementation: The software architecture includes integrating the ESP32 micro-controller with the Blynk Cloud platform for real-time monitoring and control of the traffic management system. The Blynk app provides a user-friendly interface for remotely monitoring traffic conditions and adjusting signal timings as needed.

IV. Results/Output

The results and outputs of the Density-Based Traffic Light Controller project showcase significant improvements in traffic flow and efficiency. When density is detected in any lane at an intersection, the corresponding traffic signal promptly turns green, ensuring minimal delays for vehicles in congested lanes and facilitating smoother traffic flow.

In scenarios where density is detected in multiple lanes simultaneously, a priority-based approach is employed, allowing vehicles that arrived first in each lane to proceed through the intersection. This fair allocation of green signal time optimizes the use of intersection capacity and minimizes unnecessary delays. Overall, the project achieves optimized traffic management by dynamically adjusting signal timings based on real-time traffic conditions, leading to reduced congestion and enhanced urban mobility. Additionally, the real-time monitoring and analysis capabilities of the traffic management system provide valuable insights into traffic patterns, enabling informed decision-making for future traffic management strategies.

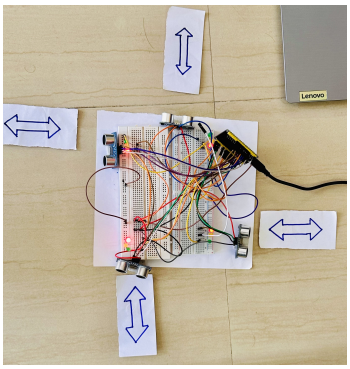

Fig. 4: When lanes are free - Red signal

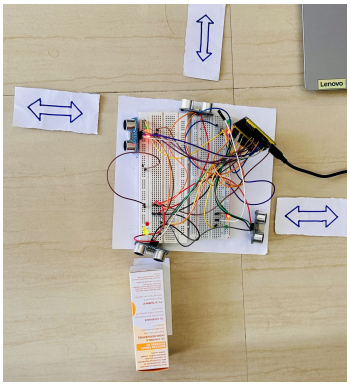

Fig. 5: When vehicle arrived - Green signal

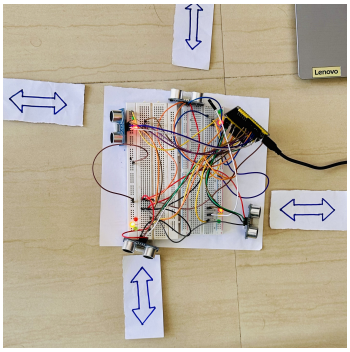

Fig. 6: When vehicle just left - Yellow signal

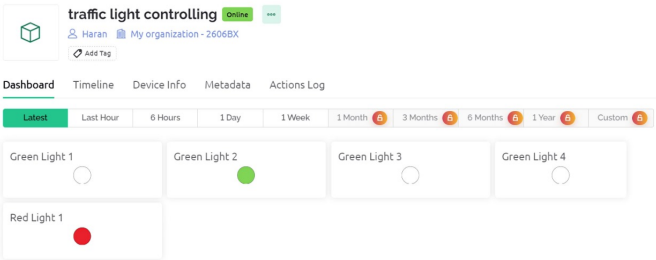

Fig. 7: Cloud Virtual Signal

V. Conclusion and Future Directions

In conclusion, the Density-Based Traffic Light Controller project demonstrates the potential of smart traffic management systems in optimizing traffic flow and improving urban mobility. By implementing a dynamic and adaptive approach to traffic control, the project offers a practical solution to the inefficiencies of traditional traffic signal management systems.

Future directions of the project may include further refining the dynamic traffic control algorithm, integrating additional sensors for more accurate traffic monitoring, and exploring opportunities for scalability and deployment in larger urban areas. Additionally, ongoing research and development efforts could focus on enhancing the system's capabilities through machine learning algorithms and advanced sensor technologies.

References

- P.Pujithsai, O.Goutham Sai, Varshith, Bagubali A "Density Controlled Traffic Lights System using Aurdino Mega 2560" International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075 (Online), Volume-9 Issue-5, March 2020
- Pramod Sharma "Density Based Intelligent Traffic Control System Using IR Sensors" Intl journal of scientific research.
- M. A.A. Parkhi, Mr. A.A. Peshattiwar, Mr.K.G. Pande "Intelligent Traffic System Using Vehicle Density". Yeshwantrao Chavan College of Engg., Nagpur. International Journal of Electrical and Electronic Engoneers, 2016.
- S.Sundara Mahalingam1 , S.Arockiaraj2 1 Assistant Professor, "DENSITY BASED TRAFFIC LIGHT CONTROL USING ARDUINO " Electrical and Electronics Engineering, Mepco Schlenk Engineering College , Tamil Nadu, India .