| **JAI SHRIRAM ENGINEERING COLLEGE**Difference Between NAAC & NBA Accreditation - Haq Se EngineerJai Shriram Engineering College (@JSREC09) / TwitterPREC LONI  **TIRUPPUR – 638 660**  Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai  Recognized by UGC & Accredited by NAACandNBA (CSE and ECE) |
| --- |

**DEPARTMENT OF**

**ELECTRONICS AND COMMUNICATION ENGINEERING**

**IBM - Naan Mudhalvan**

**Internet of Things Group 3**

**Phase 4 - Development part II**

**Tittle: Traffic Management**

**NAME : K.PRAVEENKUMAR**

**NM ID :AU711221106023**

**YEAR : III**

**INTRODUCTION:**

A Traffic Management System (TMS) through the Internet of Things (IOT) is a sophisticated system designed to monitor, control, and optimize traffic flow and transportation infrastructure using IOT technologies. It involves the integration of various IOT devices, sensors, and communication networks to collect real-time data from roadways, intersections, and vehicles. This data is then processed and analyzed to make informed decisions, improve traffic flow, enhance safety ,And reduce congestion on roads.

**DEVELOPMENT PROCESS :**

**Planning and Analysis:**

* Identify traffic issues and goals.
* Gather data on traffic patterns, congestion, and safety.
* Develop a comprehensive traffic management plan.

**Design and Engineering:**

* Design traffic control systems, including signals, signage, and road layouts.
* Consider intelligent transportation systems (ITS) for advanced traffic management.

**Implementation:**

* Install traffic control devices such as traffic lights, signs, and cameras.
* Integrate ITS technologies for real-time data collection and control.
* Implement traffic management strategies like adaptive signal control.

**Monitoring and Data Collection:**

* Continuously monitor traffic conditions using sensors and cameras.
* Collect data on traffic flow, congestion, and incidents.
* Use data for real-time decision-making and future planning.

**Evaluation and Optimization:**

* Assess the effectiveness of traffic management strategies.
* Make adjustments based on data analysis and feedback.
* Optimize traffic control to improve safety and efficiency.

**Public Engagement and Education:**

* Communicate with the public about traffic management changes.
* Educate drivers and pedestrians about new traffic rules.
* Gather public input for further improvements.

**Emergency Response:**

* Develop protocols for managing traffic during emergencies.
* Coordinate with law enforcement, emergency services, and other agencies.

**Future Planning:**

* Continuously plan for future growth and changes in traffic patterns.
* Consider long-term infrastructure improvements and innovations.

**EXISTING SYSTEM:**

The existing System is generally controlled by traffic police. The main drawback of our system which is controlled by traffic police is that the system is not smart enough to deal with the traffic congestion. The traffic police official can either block a road for more amount of time or let the vehicle on the other road pass by i.e. the decision making may not be smart enough and it entirely depends on the official’s decision.

**PROGRAM :**

**#define BLYNK\_TEMPLATE\_ID "TMPL26V4fGv5q"**

**#define BLYNK\_TEMPLATE\_NAME "Test"**

**#define BLYNK\_AUTH\_TOKEN "XEHxNF\_Ur1Nt2p7wB5B20dNI1ZUwj34P"**

**#include <WiFi.h>**

**#include <WiFiClient.h>**

**#include <BlynkSimpleEsp32.h>**

**int duration1 = 0;**

**int distance1 = 0;**

**int duration2 = 0;**

**int distance2 = 0;**

**int dis1 = 0;**

**int dis2 = 0;**

**int dis\_new1 = 0;**

**int dis\_new2 = 0;**

**int entered = 0;**

**int left = 0;**

**int inside = 0;**

**#define LED 2**

**#define PIN\_TRIG1 15**

**#define PIN\_ECHO1 14**

**#define PIN\_TRIG2 13**

**#define PIN\_ECHO2 12**

**BlynkTimer timer;**

**char auth[] = BLYNK\_AUTH\_TOKEN;**

**char ssid[] = "Wokwi-GUEST"; // your network SSID (name)**

**char pass[] = "";**

**#define BLYNK\_PRINT Serial**

**long get\_distance1() {**

**// Start a new measurement:**

**digitalWrite(PIN\_TRIG1, HIGH);**

**delayMicroseconds(10);**

**digitalWrite(PIN\_TRIG1, LOW);**

**// Read the result:**

**duration1 = pulseIn(PIN\_ECHO1, HIGH);**

**distance1 = duration1 / 58;**

**return distance1;**

**}**

**long get\_distance2() {**

**// Start a new measurement:**

**digitalWrite(PIN\_TRIG2, HIGH);**

**delayMicroseconds(10);**

**digitalWrite(PIN\_TRIG2, LOW);**

**// Read the result:**

**duration2 = pulseIn(PIN\_ECHO2, HIGH);**

**distance2 = duration2 / 58;**

**return distance2;**

**}**

**void myTimer() {**

**Serial.println("100");**

**dis\_new1 = get\_distance1();**

**dis\_new2 = get\_distance2();**

**if (dis1 != dis\_new1 || dis2 != dis\_new2){**

**Serial.println("200");**

**if (dis1 < dis2){**

**Serial.println("Enter loop");**

**entered = entered + 1;**

**inside = inside + 1;**

**digitalWrite(LED, HIGH);**

**Blynk.virtualWrite(V0, entered);**

**Blynk.virtualWrite(V2, inside);**

**dis1 = dis\_new1;**

**delay(1000);**

**digitalWrite(LED, LOW);**

**}**

**if (dis1 > dis2){**

**Serial.println("Leave loop");**

**left = left + 1;**

**inside = inside - 1;**

**Blynk.virtualWrite(V1, left);**

**Blynk.virtualWrite(V2, inside);**

**dis2 = dis\_new2;**

**delay(1000);**

**}**

**}  
}**

**void setup() {**

**Serial.begin(115200);**

**pinMode(LED, OUTPUT);**

**pinMode(PIN\_TRIG1, OUTPUT);**

**pinMode(PIN\_ECHO1, INPUT);**

**pinMode(PIN\_TRIG2, OUTPUT);**

**pinMode(PIN\_ECHO2, INPUT);**

**Blynk.begin(auth, ssid, pass, "blynk.cloud", 8080);**

**timer.setInterval(1000L, myTimer);**

**}**

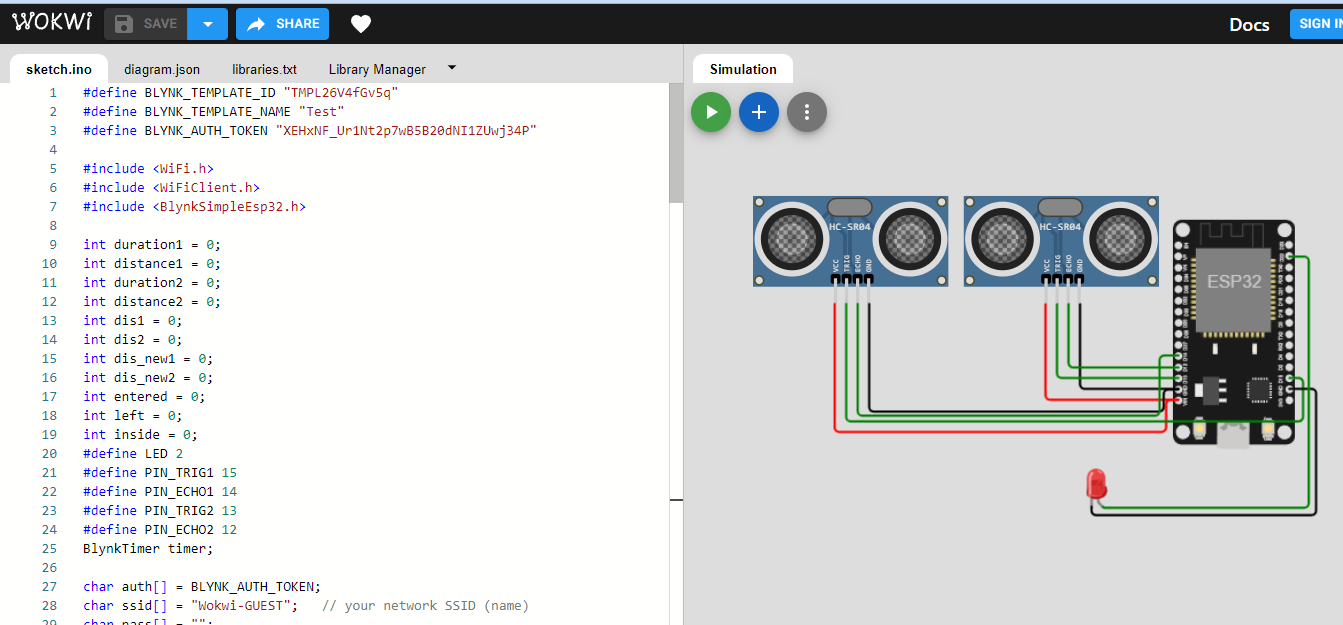
**void loop() {**

**Blynk.run();**

**timer.run();**

**}**

**SIMULATION :**

****