SKILL LAB PROJECT

**Smart Traffic Light System using IOT**

*“A Smart way to Control Traffic Control System"*



**ACKNOWLEDGEMENT**

It is my great fortune that I have got opportunity to carry out this project work under the Supervision of Prof. **(Dr.) Arnab Pal** in the Department of Electrical & Electronics Engineering, **Silicon**

**University(SU)**, Silicon Hills, DLF Cybercity, Patia, Bhubaneswar, Odisha -751024, India.

I express my sincere thanks and deepest sense of gratitude to my guide for his constant support, unparalleled guidance and limitless encouragement. I wish to convey my gratitude to Prof. **(Dr.) Rabi**

**Narayan Mishra**, HOD, Department of Electrical & Electronics Engineering, Silicon University and to the authority of Silicon

University for providing all kinds of infrastructural facility towards the research work. I would also like to convey my gratitude to all the

faculty members and staffs of the Department of Electrical &

Electronics Engineering, Silicon University for their whole-hearted cooperation to make this work turn into reality.

# **ABSTRACT**

This project presents the design and implementation of a Smart Traffic Management System using IoT technologies to enhance the efficiency and automation of urban traffic control. Leveraging components such as the ESP32 microcontroller, RFID modules, ultrasonic sensors, and vibration detectors, the system dynamically manages traffic lights based on real-time vehicle density and emergency vehicle detection.

The RFID module identifies authorized vehicles, granting them priority passage, while ultrasonic sensors monitor traffic buildup to adjust signal timing. In the event of accidents, vibration sensors trigger alerts for immediate response.

The control logic is programmed using C++ and deployed via the Arduino IDE, enabling modular and responsive traffic light control. The system successfully demonstrates real-time traffic sensing, priority-based signaling, and accident detection. This scalable solution holds significant potential for smart city applications, offering future enhancements such as cloud integration, mobile app support, and license plate recognition.

# INTRODUCTION

The rapid urbanization and increasing number of vehicles have led to severe traffic congestion and delays in many cities.

Traditional traffic systems are often static and cannot adapt to real-time traffic conditions. This project aims to design and implement a Smart Traffic Management System using Internet of Things (IoT) technology to enhance traffic efficiency and safety through automation and intelligent decision-making.

# SYSTEM COMPONENTS

The Smart Plant Irrigation System consists of the following components:

### **1.ESP32 Microcontroller**

* **Function**: Acts as the brain of the system.
* **Features**:
  + Dual-core processor with integrated Wi-Fi and Bluetooth.
  + Multiple GPIO pins for connecting sensors and actuators.
  + Fast processing speed allows real-time data handling.
  + Compatible with the Arduino IDE, allowing seamless code development and deployment.



### **2. RFID Module (MFRC522)**

* **Function**: Used for identifying vehicles equipped with RFID tags.
* **Features**:
  + Operates at 13.56 MHz (ISO/IEC 14443 standard).
  + Can read multiple types of RFID tags (e.g., MIFARE).
  + Communicates with ESP32 via SPI interface.

### **3.Ultrasonic Sensors (e.g., HC-SR04)**

* **Function**: Measure distance to detect vehicle presence and density.
* **Features**:
  + Uses sound waves to detect objects within a range (typically 2 cm to 4 m).
  + High accuracy and fast response time.
  + Outputs distance data in real-time.

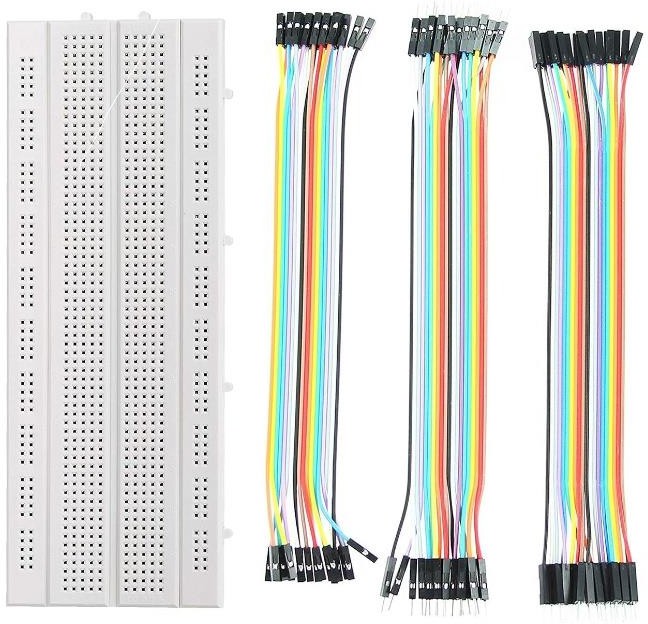
### **4.Traffic Light LEDs**

* **Function**: Simulate traffic signals using colored LED lights (Red, Yellow, Green).
* **Features**:
  + High-brightness LEDs for visibility.
  + Controlled using digital signals from ESP32 via relays or transistors.



### **5.Breadboard** Jumper Wires

* **Function**: Provide regulated power and temporary prototyping platform.
* **Features**:
  + 5V/3.3V regulated power supply for ESP32 and sensors.
  + Breadboard allows flexible testing and wiring of circuit components.



### **6.Power Supply and Breadboard**

* **Function**: Provide regulated power and temporary prototyping platform.
* **Features**:
  + 5V/3.3V regulated power supply for ESP32 and sensors.
  + Breadboard allows flexible testing and wiring of circuit components.

**Code**

**#include <SPI.h>**

**#include <MFRC522.h>**

**// RFID Pins**

**#define SS\_PIN 5**

**#define RST\_PIN 4**

**MFRC522 rfid(SS\_PIN, RST\_PIN);**

**// Traffic Light Pins**

**#define R1 25**

**#define Y1 26**

**#define G1 27**

**#define R2 14**

**#define Y2 12**

**#define G2 13**

**// Ultrasonic Sensor Pins**

**#define TRIG1 15**

**#define ECHO1 2**

**#define TRIG2 17**

**#define ECHO2 16**

**// Settings**

**const int vehicleThreshold = 10; // cm**

**int greenTime = 5000; // 5 seconds**

**int yellowTime = 3000; // 3 seconds**

**int allRedPause = 3000; // 3 seconds**

**void setup() {**

**Serial.begin(115200);**

**SPI.begin();**

**rfid.PCD\_Init();**

**// Set pin modes**

**pinMode(TRIG1, OUTPUT); pinMode(ECHO1, INPUT);**

**pinMode(TRIG2, OUTPUT); pinMode(ECHO2, INPUT);**

**pinMode(R1, OUTPUT); pinMode(Y1, OUTPUT); pinMode(G1, OUTPUT);**

**pinMode(R2, OUTPUT); pinMode(Y2, OUTPUT); pinMode(G2, OUTPUT);**

**allRed();**

**Serial.println("🚦 Smart Traffic System Initialized");**

**}**

**void loop() {**

**// RFID Emergency Vehicle Detection**

**if (rfid.PICC\_IsNewCardPresent() && rfid.PICC\_ReadCardSerial()) {**

**if (isMasterCard()) {**

**Serial.println("🚨 Emergency Vehicle Detected → Road 1 Green");**

**greenLight(1, 20000);**

**allRed(); delay(2000);**

**} else {**

**String scannedUID = getUIDString();**

**Serial.print("Unknown RFID UID: ");**

**Serial.println(scannedUID);**

**}**

**rfid.PICC\_HaltA();**

**return;**

**}**

**// Normal Traffic Cycle: Road 1**

**digitalWrite(R1, LOW); digitalWrite(G1, HIGH); digitalWrite(Y1, LOW);**

**digitalWrite(R2, HIGH); digitalWrite(G2, LOW); digitalWrite(Y2, LOW);**

**delay(greenTime);**

**digitalWrite(G1, LOW); digitalWrite(Y1, HIGH);**

**delay(yellowTime);**

**digitalWrite(Y1, LOW); digitalWrite(R1, HIGH);**

**delay(allRedPause);**

**// Road 2**

**digitalWrite(R2, LOW); digitalWrite(G2, HIGH); digitalWrite(Y2, LOW);**

**delay(greenTime);**

**digitalWrite(G2, LOW); digitalWrite(Y2, HIGH);**

**delay(yellowTime);**

**digitalWrite(Y2, LOW); digitalWrite(R2, HIGH);**

**delay(allRedPause);**

**// Check for vehicle presence**

**bool road1Busy = isVehicleConstantlyPresent(TRIG1, ECHO1);**

**bool road2Busy = isVehicleConstantlyPresent(TRIG2, ECHO2);**

**if (road1Busy) {**

**Serial.println("📈 Road 1 is busy → Extending Green");**

**greenLight(1, greenTime + 5000);**

**} else if (road2Busy) {**

**Serial.println("📈 Road 2 is busy → Extending Green");**

**greenLight(2, greenTime + 5000);**

**}**

**// Print distances**

**long distance1 = getDistance(TRIG1, ECHO1);**

**long distance2 = getDistance(TRIG2, ECHO2);**

**Serial.print("Distance on Road 1: ");**

**Serial.print((distance1 == -1) ? "Error" : String(distance1) + " cm");**

**Serial.print(" | Distance on Road 2: ");**

**Serial.println((distance2 == -1) ? "Error" : String(distance2) + " cm");**

**delay(1000);**

**}**

**// Get Distance from Ultrasonic Sensor**

**long getDistance(int trig, int echo) {**

**digitalWrite(trig, LOW);**

**delayMicroseconds(2);**

**digitalWrite(trig, HIGH);**

**delayMicroseconds(10);**

**digitalWrite(trig, LOW);**

**long duration = pulseIn(echo, HIGH, 30000); // timeout 30ms**

**if (duration == 0) return -1;**

**return duration \* 0.0343 / 2;**

**}**

**// Check if vehicle consistently present**

**bool isVehicleConstantlyPresent(int trig, int echo) {**

**int count = 0;**

**for (int i = 0; i < 5; i++) {**

**long d = getDistance(trig, echo);**

**if (d > 0 && d < vehicleThreshold) count++;**

**delay(200);**

**}**

**return count >= 4; // majority detected**

**}**

**// Convert UID to String**

**String getUIDString() {**

**String content = "";**

**for (byte i = 0; i < rfid.uid.size; i++) {**

**if (rfid.uid.uidByte[i] < 0x10) content += "0";**

**content += String(rfid.uid.uidByte[i], HEX);**

**if (i < rfid.uid.size - 1) content += " ";**

**}**

**content.toUpperCase();**

**return content;**

**}**

**// Check if Scanned UID Matches Emergency UID**

**bool isMasterCard() {**

**String uidStr = getUIDString();**

**Serial.print("Scanned UID: ");**

**Serial.println(uidStr);**

**return (uidStr == "83 22 29 3"); // Replace with your actual emergency UID**

**}**

**// Give green light to specified road**

**void greenLight(int road, int duration) {**

**if (road == 1) {**

**digitalWrite(G2, LOW); digitalWrite(Y2, HIGH);**

**delay(2000);**

**digitalWrite(Y2, LOW); digitalWrite(R2, HIGH);**

**digitalWrite(R1, LOW); digitalWrite(G1, HIGH);**

**} else {**

**digitalWrite(G1, LOW); digitalWrite(Y1, HIGH);**

**delay(2000);**

**digitalWrite(Y1, LOW); digitalWrite(R1, HIGH);**

**digitalWrite(R2, LOW); digitalWrite(G2, HIGH);**

**}**

**delay(duration);**

**digitalWrite(G1, LOW); digitalWrite(G2, LOW);**

**}**

**// Turn all lights to red**

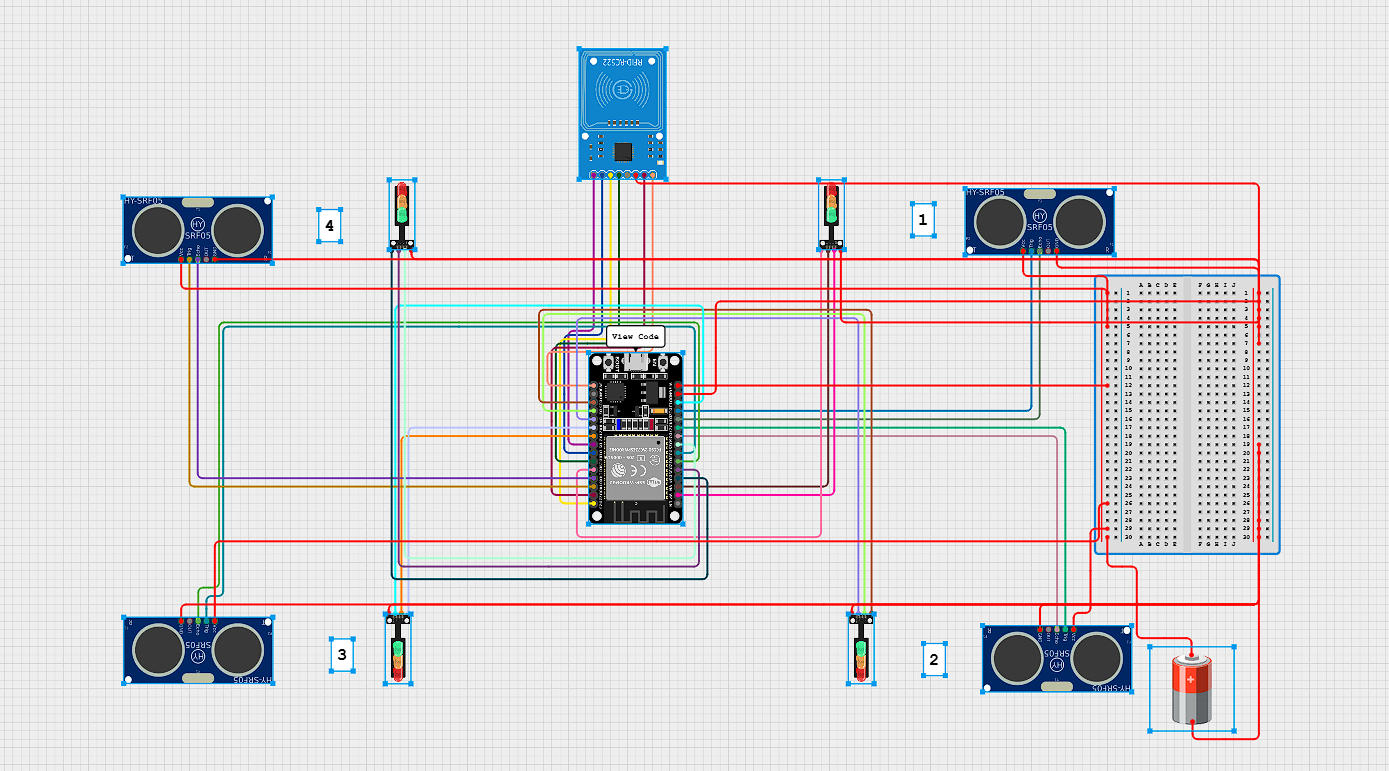
**void allRed() {**

**digitalWrite(R1, HIGH); digitalWrite(Y1, LOW); digitalWrite(G1, LOW);**

**digitalWrite(R2, HIGH); digitalWrite(Y2, LOW); digitalWrite(G2, LOW);**

**}**

**HARDWARE CONNECTION**

****

**ALGORITHM FLOWCHART**

START  
 |  
 V  
Initialize ESP32, RFID, Ultrasonic Sensors, Relays  
 |  
 V  
Check RFID Reader ------ YES --> Prioritize RFID Vehicle and Switch Lane to GREEN  
 | |

NO V  
 Hold Other Lanes Temporarily  
 V  
 Read Ultrasonic Sensors  
 |  
 V  
Determine Lane with Highest Traffic  
 |  
 V  
Adjust Signal Timing Dynamically  
 |  
 V  
Control Traffic Lights via Relay Module  
 |  
 V  
Repeat Loop

**WORKING**

## ****Working Principle****

* **Vehicle Detection**: Ultrasonic sensors are placed at junctions to detect the presence and number of vehicles. Traffic light durations are dynamically adjusted based on density.
* **Vehicle Prioritization**: RFID readers at entry points detect RFID tags attached to authorized or emergency vehicles, allowing them signal priority.
* **System Control**: ESP32 collects sensor data and processes it using pre-programmed logic to manage traffic lights through relay modules.
* **Software**: Developed using C++ on the Arduino IDE, offering modular, efficient, and scalable code.

# **METHODOLOGY**

The Smart Traffic System project follows a structured approach integrating hardware, software, and IoT-based logic to dynamically manage traffic flow. The methodology can be broken down into the following phases:

### **1. Problem Identification**

Urban traffic congestion and emergency vehicle delays are growing concerns. The project identifies the need for an adaptive, real-time traffic control system capable of addressing these issues using smart sensing and control.

### **2. System Design**

* A block diagram and logical flow were conceptualized to guide system behavior.
* Design decisions were made to prioritize modularity and scalability.

### **3. Component Selection**

* **ESP32** was chosen as the central controller due to its processing power and built-in Wi-Fi.
* **RFID module (MFRC522)** enables recognition of emergency or authorized vehicles.
* **Ultrasonic sensors** assess vehicle density to adapt traffic signal timing.
* **Relay modules and LEDs** simulate traffic light behavior.
* **9V battery** provides portable power, enabling off-grid operation and demonstrations.

### **4. Sensor Integration and Programming**

* Sensors and modules were interfaced with ESP32 using the Arduino IDE.
* Code written in **C++** handles:
  + Real-time sensor polling.
  + Signal prioritization.
  + Dynamic LED control.
* The logic was structured to process inputs continuously and trigger appropriate outputs without manual intervention.

### **5. System Testing**

* The prototype was tested under various simulated traffic conditions.
* Emergency vehicle scenarios were validated using RFID tags.
* The system dynamically adjusted traffic signals based on real-time inputs.

### **6. Result Analysis**

* Successful real-time operation with prioritized signaling and dynamic control.
* Validated sensor accuracy and response time under different conditions.
* Demonstrated the system’s potential for expansion to multiple junctions.

# **FUTURE WORK**

While the current prototype of the Smart Traffic System demonstrates effective real-time control and vehicle prioritization, several enhancements can further improve its functionality, scalability, and integration with modern urban infrastructure:

### **1. Cloud Integration**

* Connect the system to a cloud platform (e.g., Firebase, AWS IoT) for real-time data storage, analysis, and monitoring.
* Enable remote control and dashboard visualization for traffic authorities.

### **2. GPS Module Integration**

* Use GPS to track vehicle movement and automatically identify approaching emergency vehicles without relying solely on RFID.
* Enhance route optimization and tracking for public or emergency vehicles.

### **3. Mobile Application Support**

* Develop a mobile app for users and administrators to:
  + View real-time traffic conditions.
  + Receive traffic alerts and updates.
  + Monitor system performance and configure settings remotely.

### **4. Camera-Based License Plate Recognition**

* Integrate a camera module with OCR (Optical Character Recognition) to identify vehicle license plates.
* Enable automated fine issuance, entry logging, or further vehicle prioritization based on license.

### **5. AI-Powered Traffic Prediction**

* Implement machine learning models to predict traffic flow trends based on historical data.
* Use predictive analytics to preemptively adjust traffic signals before congestion builds up.

### **6. Solar Power Integration**

* Add solar panels and smart energy management to power the system sustainably in outdoor environments.

### **7. Multiple Junction Synchronization**

* Develop communication protocols to synchronize traffic signals across multiple intersections.
* Create a city-wide adaptive traffic management system.

# **APPLICATIONS**

### 1. **Urban Traffic Management**

* Automatically adjusts traffic signals based on real-time vehicle density.
* Helps reduce traffic congestion in metropolitan areas.
* Ensures smoother traffic flow during peak hours.

### 2. **Emergency Vehicle Priority**

* Detects ambulances, fire trucks, and police vehicles via RFID.
* Automatically gives them a green light to minimize delays.
* Enhances emergency response times and saves lives.

### 3. **Accident Detection & Alerting**

* Vibration sensors detect potential collisions.
* Sends immediate alerts to traffic control centers and emergency services.
* Reduces response time for accident victims.

### 4. **Smart City Data Analytics**

* Vehicle count data is sent to the cloud.
* Authorities can analyze traffic patterns, congestion hotspots, and plan infrastructure upgrades accordingly.

### 5. **Environmental Benefits**

* Reduces vehicle idling time at signals, lowering fuel consumption and air pollution.
* Contributes to a cleaner, greener urban environment.

### 6. **Law Enforcement & Regulation**

* Helps monitor road usage and enforce traffic rules.
* Can be integrated with cameras or license plate readers for automated ticketing or surveillance.

### 7. **Public Transport Optimization**

* Enhances traffic signal coordination for buses and public transport, reducing delays.
* Supports smart routing and scheduling.

# **CONCLUSION**

The Smart Traffic System using IoT offers a forward-looking solution to the growing challenges of urban traffic management. By leveraging real-time sensor data, automated signal control, and cloud integration, this system enhances road safety, reduces congestion, and prioritizes emergency response. Its ability to detect vehicles, manage traffic flow efficiently, and respond dynamically to road conditions demonstrates the powerful role IoT can play in building smarter and more sustainable cities. With continued development and integration, such systems can significantly contribute to the vision of intelligent, connected urban infrastructure