National Textile University

Project Proposal

Embedded IoT Systems – Fall 2025

🚦 **Project Title:**

IoT-Based Smart Traffic Management System Using ESP32

Group Name:

Tech X

Group Members & Assigned Tasks (Updated)

Member	Responsibility
Saim Mustafa	ESP32 firmware development, IoT integration, web dashboard programming

Member	Responsibility
Ibtisam Butt	Hardware wiring, LED signal assembly, IR sensor calibration
Shahzaman Dhillon	Circuit design, PCB layout, power distribution, testing & debugging

---

🎯 Objectives of the Project

1. To design and implement a smart traffic management system using the ESP32 microcontroller.
2. To use IR sensors to detect real-time vehicle density for intelligent traffic control.
3. To enable IoT-based monitoring using a built-in ESP32 Web Server dashboard.
4. To understand signal timing, preemption logic, and safe switching mechanisms in traffic control.
5. To apply embedded systems, IoT, and sensor interfacing knowledge in a real-world application.
6. To reduce unnecessary waiting time and improve road traffic flow through automation.
7. To implement accurate sensor-based vehicle counting using debouncing and filtering.

🛠️ Hardware Components Used

- **ESP32**

- **IR Sensors × 4 (Vehicle detection for 4 lanes)**
- **Red, Yellow, Green LEDs × 12**
  - **3 LEDs per lane**
- **220Ω Resistors for LEDs**
- **Custom PCB Board for:**
  - **LED signal distribution**
  - **Sensor power lines**
  - **Neat wiring**
- **5V Power Supply / USB Adapter**
- **Jumper Wires (Male–Female / Male–Male)**
- **Breadboard (optional)**
- **5V–3.3V voltage regulation (if needed)**

💻 Software Stack

- Arduino IDE (Programming ESP32)
- ESP32 Board Package
- HTML/CSS for Web Dashboard Interface
- ESPAsyncWebServer / WebServer library
- WiFi library
- IR sensor input handling with debounce logic

🚦 Key Features of This IoT Smart Traffic System

✓ **Real-Time Vehicle Detection**

Each lane uses an IR sensor that detects cars with 95–99% accuracy**.**

✓ **Automatic Density-Based Traffic Control**

When one lane has more cars than others, it automatically gets the green light.

✓ **Intelligent Preemption**

If 2 or more vehicles are detected, the system overrides normal mode and gives immediate green to that lane.

✓ **Built-In Web Dashboard (IoT)**

ESP32 hosts a local WiFi dashboard showing:

- Current lane signals (R/Y/G)
- Vehicle count of each lane
- System mode (Normal / Preemption)
- Real-time refresh without external cloud

✓ **Vehicle Counter Reset Logic**

Counters reset only after that lane turns green.

✓ **Safety Delay (All-Red Time)**

Before switching signals, the system activates 2 seconds of all-red to prevent collisions.

✓ **Debounce Filtering (80ms delay)**

Removes false triggers caused by:

- sunlight

- vibration
- sensor noise
- pedestrians

✓ **WiFi Control & Monitoring**

Since ESP32 has built-in WiFi, no extra module is needed.
The dashboard works on any phone or laptop browser.

⚙️ **How the System Works (Full Operation Flow)**

1. Normal Cycle Mode

- When traffic is low, signals rotate in a fixed order:
  Lane 1 → Lane 2 → Lane 3 → Lane 4

2. Vehicle Counting Mode

- IR sensors detect vehicles when the lane is red.
- Counters increment in real time.

3. Preemption Mode

Triggered automatically when:
A lane has 2 or more vehicles waiting.
System immediately switches green to that lane.

4. Safe Switching

Before switching to green:
✓ **All signals turn RED for 2 seconds**
✓ **Prevents accidents**

5. Web Dashboard Monitoring

The ESP32 WiFi dashboard shows:

- Vehicle count per lane
- Live signal colors
- Total vehicles detected
- System mode

6. Automatic Reset

When a lane turns green, its counter resets to 0.

Circuit Diagram Summary

The circuit is designed around the ESP32, which handles everything:

Inputs:

- **4 × IR sensors**
    - **Connected to GPIO pins**
    - **Powered by 5V (or 3.3V depending on module)**

Outputs:

Each of the 4 lanes has:

- **1 Red LED**
- **1 Yellow LED**
- **1 Green LED**
  (12 LEDs in total)

All LEDs are connected to:

- ESP32 digital output pins

- Current-limiting resistors (220Ω)

Power System:

- **Entire system powered with 5V**
- **ESP32 regulated to 3.3V internally**
- **All grounds tied together**

IoT:

- **ESP32 creates a WiFi network**
- **Web server dashboard accessible through any browser**

This setup ensures safe, automated, IoT-enabled traffic control based on density.

Circuit Diagram of IoT-Based Traffic Light Controller

The detailed circuit diagram below shows the complete wiring schematic for this IoT-based traffic light controller: