SMART TRAFFIC LIGHT SYSTEM FOR EMERGENCY & TRAFFIC OPTIMIZATION

"Adaptive Signal Control for Emergency Vehicles and Efficient Traffic Flow"

FIRST REVIEW

Presented By : Shreya Bakre 22BEC0184
M Saisharan 22BEC0603

TABLE OF CONTENTS

PROJECT PRESENTATION

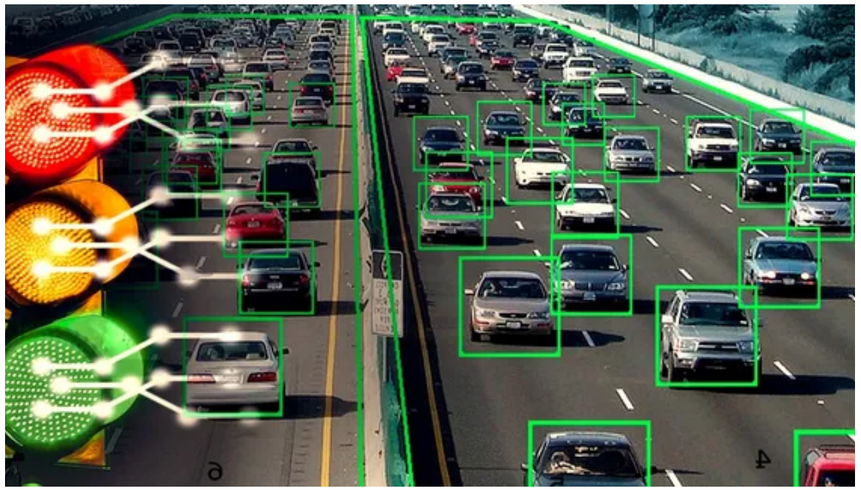
ABSTRACT

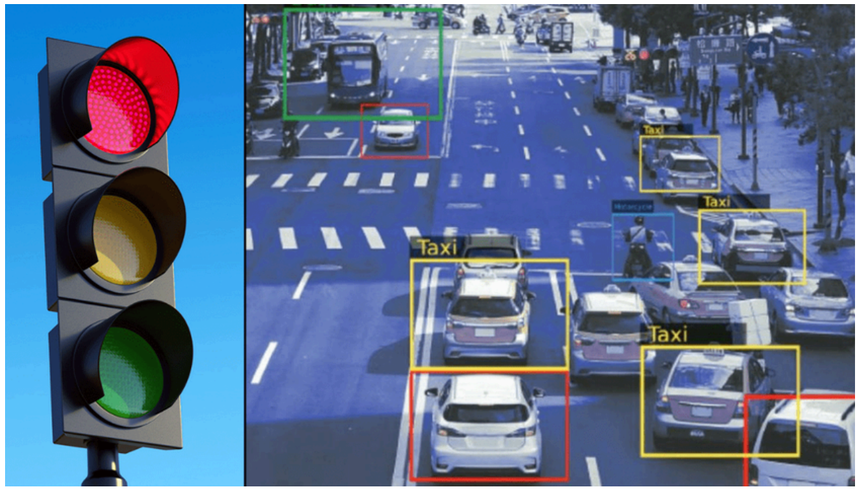
This project introduces an intelligent traffic light system that dynamically adjusts signal timings based on real-time traffic conditions while prioritizing emergency vehicles. By using computer vision (YOLO) and sensor-based detection, the system detects emergency vehicles and grants them green lights to clear their path.

The system employs a microphone module to detect siren frequencies (400–1600 Hz), ensuring that the signal stays green while an emergency vehicle approaches. Once the siren fades, the system reverts to its normal operation.

With the integration of AI-driven automation, this solution reduces congestion, enhances emergency response times, and ensures smoother, more efficient traffic flow for all road users.

INTRODUCTION





- Urban traffic congestion is a major issue, causing delays, increased pollution, and higher fuel consumption. Traditional fixed-schedule traffic lights fail to adapt to varying traffic conditions and prioritize emergency vehicles, worsening bottlenecks and response times. This can lead to critical delays in emergencies, where the Golden Hour is vital.

- To address these challenges, we propose a Smart Traffic Light System that adjusts signal timings based on real-time traffic data from sensors. The system prioritizes emergency vehicles, such as ambulances, fire trucks, and police cars, by using siren detection and additional technologies like GPS or RFID, ensuring faster passage.

- By dynamically optimizing traffic flow and improving emergency response times, this system creates a safer, more efficient urban traffic environment, potentially saving lives and reducing congestion.

PROBLEM DEFINITION 📊🚗💡

Traffic Congestion

- Urban traffic congestion causes wasted time, increased fuel consumption, and higher emissions.
- Fixed-cycle traffic lights fail to adapt to varying traffic conditions, especially during peak times.

Emergency Vehicle Challenges

- Emergency vehicles struggle to pass through congested intersections.
- Delayed response times can result in critical consequences, including loss of lives.

Inefficient Traffic Management

- Conventional traffic systems don't prioritize emergency vehicles or adjust to real-time traffic flow.
- Traffic bottlenecks and slow emergency vehicle passage worsen overall efficiency.

OBJECTIVE 🎯

Adaptive Traffic Signal Control 🚦

- Establish smart traffic management that adjusts signal timings based on real-time traffic volume.
- Provide longer green lights for lanes with high traffic density and shorter durations for lanes with low traffic.

Preemptive Pre-emption for Emergency Vehicles 🚑

- Detect emergency vehicles (ambulances, fire trucks, police cars) using sound sensors, RFID, or GPS.
- Automatically prioritize the emergency lane by switching the signal to green, ensuring quick passage.

Efficient Traffic Flow Management 🚗

- Optimize signal cycles to reduce congestion and improve overall traffic flow.
- Minimize delays for both regular traffic and emergency vehicles, ensuring smooth movement.

Incorporating Sensor Technologies 🛠️

- Use a combination of traffic sensors (infrared, radar, cameras) and sound detectors for emergency vehicle detection.
- Ensure adaptability to all vehicle types and varying traffic conditions.

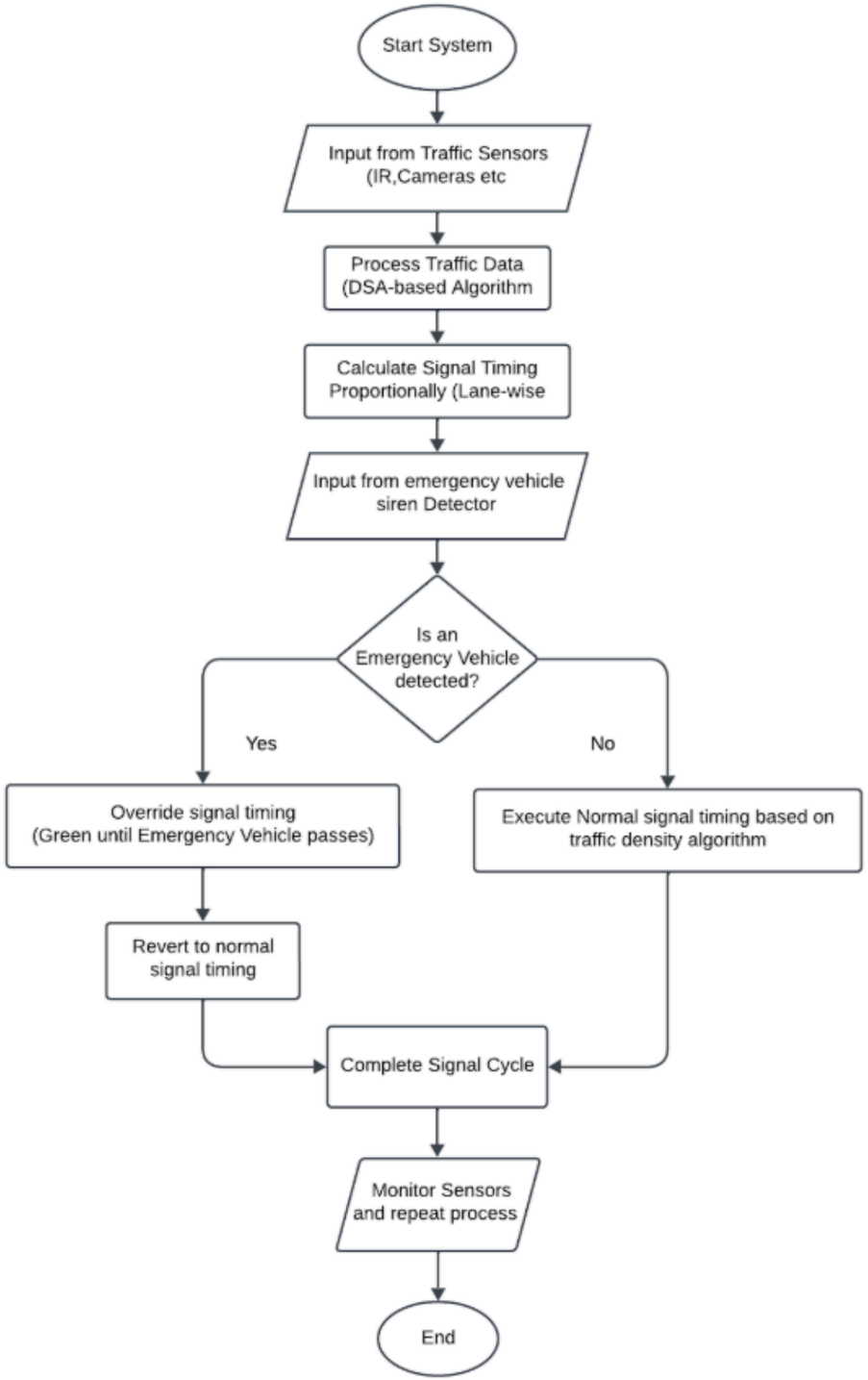
Scalability 🌍

- Use a combination of traffic sensors (infrared, radar, cameras) and sound detectors for emergency vehicle detection.
- Ensure adaptability to all vehicle types and varying traffic conditions.

Real-Time Adaptation ⏱️

- Continuously monitor and adapt the system to real-time traffic conditions.
- Ensure optimal performance by adjusting the system as needed for changing traffic patterns.

FLOW DIAGRAM



Start System 🖥️

- Input from Traffic Sensors (IR, Cameras, etc.) 📷
- Process Traffic Data (DSA-based Algorithm) 🤖
- Calculate Signal Timing Proportionally (Lane-wise) ⏱️

Input from Emergency Vehicle Siren Detector 🚨

- Is an Emergency Vehicle Detected?
  - Yes ✅
    - Override Signal Timing (Green until Emergency Vehicle Passes) 🚦🚑
  - No ❌
    - Execute Normal Signal Timing Based on Traffic Density Algorithm 📊

Revert to Normal Signal Timing 🔄

- Complete Signal Cycle 🔄
- Monitor Sensors and Repeat Process 🔄

End 🛑

EXPERIMENTAL RESULTS 📊📈

SECTION A :COUNTING VEHICLES AND TIME CALCULATION

Image 1

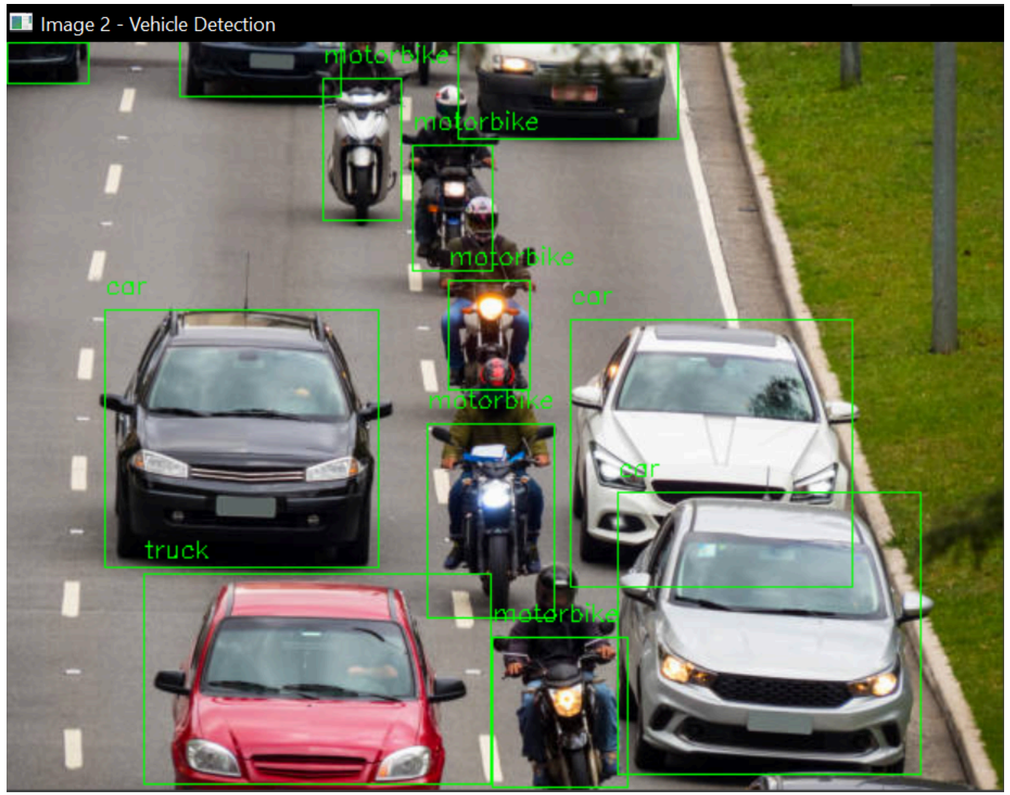


Image 2

EXPERIMENTAL RESULTS 📊📈

SECTION A :COUNTING VEHICLES AND TIME CALCULATION

Image 3

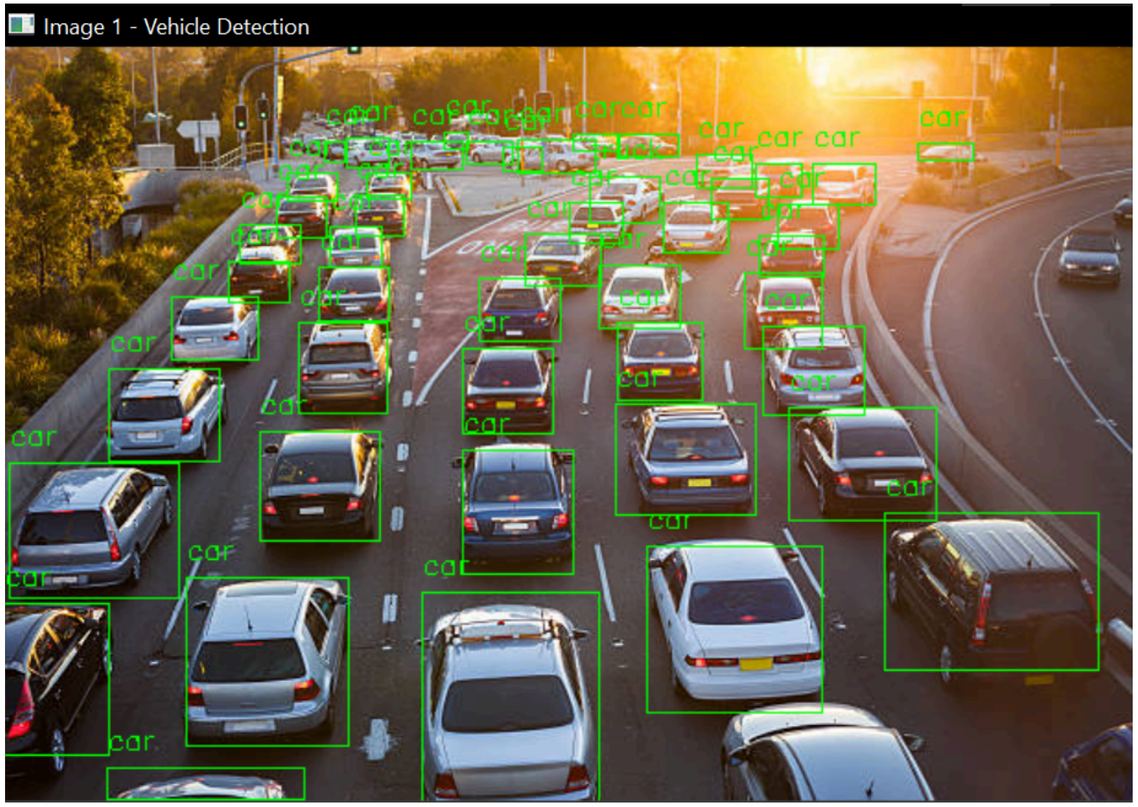


Image 4

OUTPUT RESULT 🚦

Traffic Clearance and Remaining Vehicles:

Vehicle Type	Crossed in Image 1	Crossed in Image 2	Crossed in Image 3	Crossed in Image 4
car	9	6	45	29
bus	0	0	0	2
truck	0	1	0	2
motorbike	0	5	0	6
Total Vehicles Counted	9	12	48	51
Total Vehicles Crossed	9	12	45	39
Total Time Used (s)	18	25.5	90	89

Remaining in Image 1	Remaining in Image 2	Remaining in Image 3	Remaining in Image 4
0	0	2	0
0	0	0	0
0	0	1	0
0	0	0	12

EXPLANATION🔍

This code is designed to process images of traffic, detect vehicles in those images, count how many vehicles can cross a traffic light in a given time (90 seconds), and then display a table comparing the results across multiple images.

Step-by-Step Breakdown:

1. Loading the YOLO Model

- YOLO (You Only Look Once) is a powerful model used to detect objects in images, including vehicles like cars, trucks, buses, and motorbikes.
- The code loads the YOLO model using three files:
  - yolov3.cfg: A configuration file that defines how YOLO works.
  - yolov3.weights: Pre-trained data that helps YOLO recognize objects.
  - coco.names: A list of object classes, including "car", "bus", "truck", etc.

2. Detecting Vehicles in an Image

- For each image, the code processes the image to detect vehicles.
- It uses the YOLO model to scan the image and draw a box around each detected vehicle (e.g., car, truck, bus, etc.).
- It also counts how many vehicles of each type (car, bus, truck, motorbike) are in the image.

3. Calculating Vehicles That Can Cross in 90 Seconds

- After detecting vehicles in each image, the code calculates how many vehicles can cross the traffic light in 90 seconds.
  - Different vehicle types take different amounts of time to cross.
  - The code checks how many vehicles can cross within the given time and how many are left for the next round.

4. Comparing the Results

- The code does this for multiple images (you provide a list of images to compare).
- It creates a table showing:
  - How many vehicles of each type crossed in each image.
  - The total number of vehicles in each image.
  - The total time used for crossing vehicles in each image.

5. Displaying the Results

- The program displays:
  - Bounding boxes around detected vehicles in the images.
  - A table summarizing how many vehicles crossed and the total time used for each image.
  - A conclusion about which image has the most vehicles remaining for the next round.

Some general assumptions taken

Each vehicle type is assigned a time required to pass the signal:

- Car = 2 seconds
- Bus = 5 seconds
- Truck = 6 seconds
- Motorbike = 1.5 seconds

Max Time = 90 seconds:
The system assumes that the total time available for vehicles to cross the intersection is fixed at 90 seconds.
Fixed Time Limit:
This means the code will calculate how many vehicles can pass based on the assumption that the traffic light will stay green for 90 seconds, regardless of the traffic flow or conditions.

Code Link:
https://docs.google.com/document/d/1m-pADVcz1I4pdLIP6VaNrl25Gtf2DsurMC8Q2F8T5kw/edit?usp=sharing

SECTION B:
HARDWARE IMPLEMENTATION FOR AUTOMATED TRAFFIC SIGNAL CONTROL

Proposed Methodology

Ambulance Detection:

- Sound Detection: A MAX9814 microphone module captures the siren’s intensity.
- Wireless Signal Transmission: The ambulance transmits its location using LoRa/NRF24L01.

Traffic Signal Adjustment:

- If an ambulance is detected, opposite signals turn red, while the ambulance lane turns green.
- A blue LED blinks to alert nearby drivers.
- The OLED display shows a visual alert.

Manual Override:

- If needed, RFID or a keypad allows manual traffic light control.

.

ADDITIONAL HARDWARE FOR ENHANCED IMPLEMENTATION

Better Sound Detection	**MAX9814 Microphone Module (AGC)**	**Ensures better sound capture and automatic gain control (better than KY-037).**
Frequency Analysis	**DFRobot Gravity Analog Sound Level Meter**	**Captures dB levels more accurately for siren intensity detection.**
Wireless Traffic Control	**NRF24L01 / LoRa Module**	**Sends signals to multiple traffic lights for a coordinated response.**
Real-Time Cloud Monitoring	**ESP32 with MQTT Protocol**	**Logs siren detection and traffic light status in a cloud database.**
LCD/LED Display	**OLED Display (0.96")**	**Shows “ Ambulance Detected – Green Light ON” message.**
Emergency Manual Override	**RFID/Keypad Module**	**Traffic officers can manually control the signal for special cases.**

VIRTUAL IMPLEMENTATION

Why Use a Potentiometer?

The online simulation platform does not support a microphone module for real-time sound detection.

A potentiometer is used to simulate sound intensity variations, mimicking the microphone’s output.

How It Works in Our Simulation?

The microphone detects siren sound and converts it into an analog signal.

Instead, we use a potentiometer to manually adjust intensity levels .

High value → Siren detected → Green light ON
Low value → No siren → Normal traffic cycle resumes

Outcome

Successfully simulates emergency siren detection in the simulation.

Allows us to test the traffic light control logic without a real microphone.

Prepares us for future hardware implementation with an actual microphone module.

ONLINE SIMULATION

Simulation Logic

Potentiometer → Simulates sound sensor (higher values = louder siren).

Arduino reads potentiometer values and determines ambulance presence.

LEDs switch based on the ambulance status.

16x2 LCD displays messages for different states.

Button → Manual override for emergency situations.

Real Hardware	Tinkercad Substitute
ESP32	Arduino Uno/Nano
MAX9814 Microphone	KY-038 / KY-037 Sound Sensor
OLED Display	16x2 LCD
LoRa/NRF24L01	Simulated using Serial Monitor
RFID Manual Override	Push Button

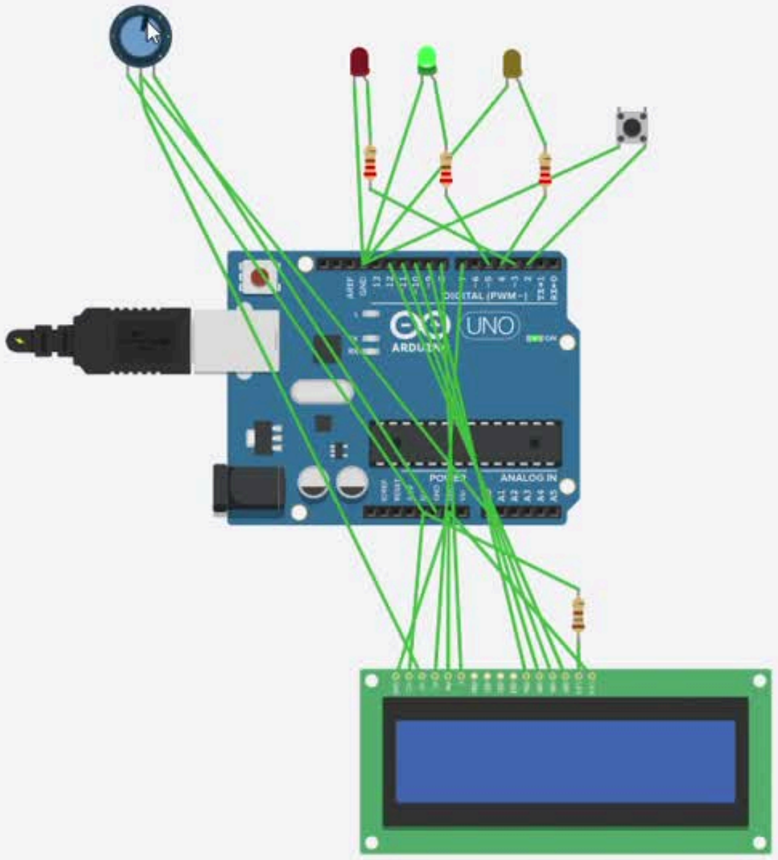
Stunning Kasi-Fyyran

All changes saved

Simulator time: 00:00:05

Code | Stop Simulation | Send To

1 (Arduino Uno R3)

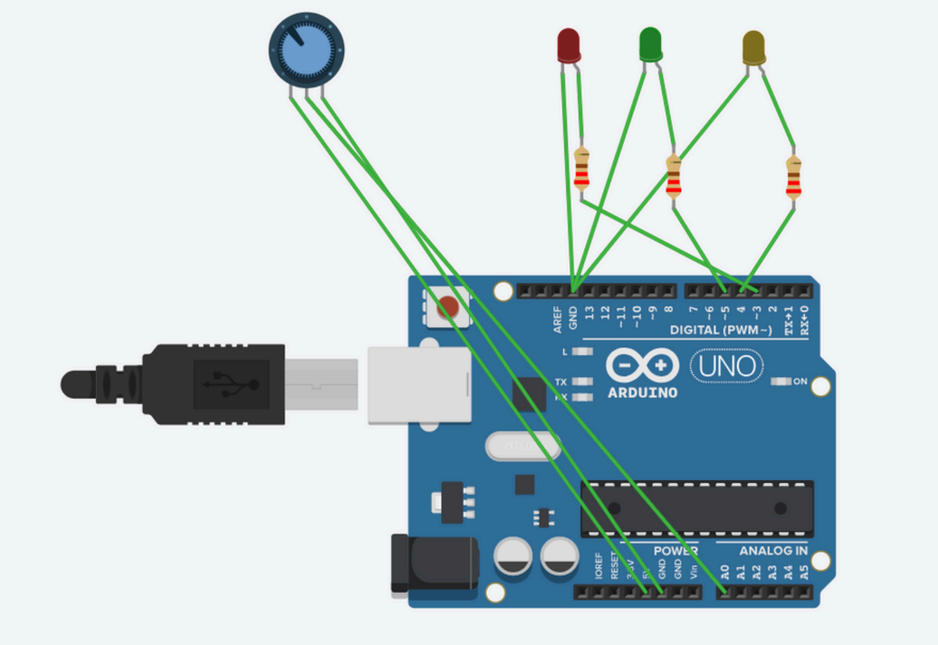


```
61          lcd.print("Green Light ON");
62
63          digitalWrite(RED_LIGHT, LOW);
64          digitalWrite(YELLOW_LIGHT, LOW);
65          digitalWrite(GREEN_LIGHT, HIGH);
66      }
67      else {
68          Serial.println(" 🚦 Normal Traffic Mode");
69          lcd.setCursor(0, 0);
70          lcd.print("Normal Traffic");
71          lcd.setCursor(0, 1);
72          lcd.print("Red Light ON");
73
74          digitalWrite(GREEN_LIGHT
75          digitalWrite(YELLOW_LIGH
```

Serial Monitor

```
Sound Level: 647
ð□□□ Ambulance Detected! Turning Green Light ON
Sound Level: 647
ð□□□ Ambulance Detected! Turning Green Light ON
Sound Level: 647
ð□□□ Ambulance Detected! Turning Green Light ON
Sound Level: 647
ð□□□ Ambulance Detected! Turning Green Light ON
Sound Level: 647
ð□□□ Ambulance Detected! Turning Green Light ON
Sound Level:
```

Send | Clear



Our project ensures real-time, accurate traffic light automation for emergency vehicles by leveraging frequency detection and modulation analysis, enabling faster and safer emergency response.

Drive link of Code	https://drive.google.com/file/d/1pDRAJKYq6w52Xi2Nay_hrisp5fdhpKH5/view?usp=drive_link
Link of Simulation	https://www.tinkercad.com/things/6m6dgSgVuDi-stunning-kasi-fyyran/editel?returnTo=https%3A%2F%2Fwww.tinkercad.com%2Fdashboard%2Fdesigns%2Fcircuits