

# TRAFFIC MANAGEMENT SYSTEM IN IOT(INTERNET OF THINGS)

## INRODUCTION

"The Internet of Things (IoT) is a revolutionary paradigm that empowers everyday objects to communicate and exchange data over the internet. By connecting physical devices and sensors, IoT is reshaping industries, homes, and cities, enabling greater automation, efficiency, and insights into our increasingly interconnected world."

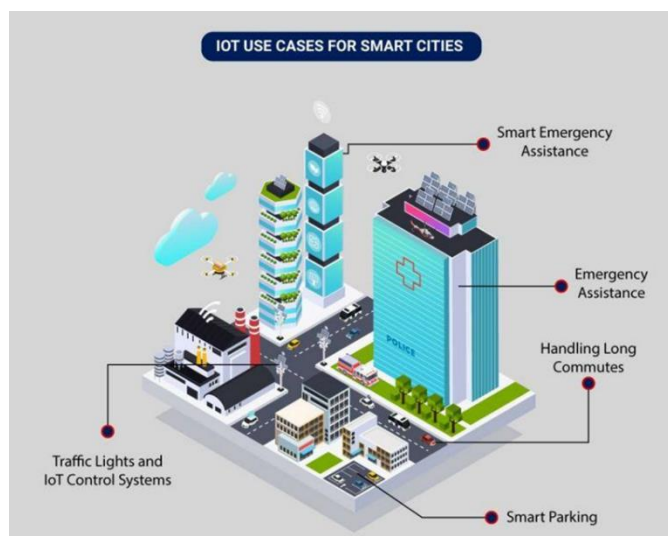
## Application

***Parking Enabled through IoT:*** Smart meters and mobile apps make on-street parking spaces easily accessible with instant notifications. Drivers receive alerts whenever a parking spot is available to reserve it instantly. The app gives easy directions to the parking spot with a convenient online payment option.

***Emergency Assistance through IoT:*** A traffic monitoring system using IoT technology enables emergency responders to speed up the care mechanism in case of accidents late at night or in isolated locations. The sensors on the road detect any accident, and the problem is immediately reported to the traffic management system. This request is passed on to relevant authorities to take corrective action. Emergency response personnel would include medical technicians, police officers, and fire departments for enhanced responsiveness and timely intervention.

***Traffic Lights and IoT Control Systems:*** Smart traffic signals may look like a typical stoplight, yet they utilize an array of sensors to monitor real-time traffic. Usually, the goal is to help cars reduce the amount of time

spent idle. And IoT technology enables the various signals to communicate with each other. This is while adapting to changing traffic conditions in real time. The outcome is less time spent in traffic jams and even reduced carbon emissions.



## Problem solving and design things

### Project Overview:

In this project, we'll create a basic traffic management system that can monitor traffic at an intersection and control traffic lights accordingly. We'll use Raspberry Pi, ultrasonic sensors, and LEDs to simulate the intersection

### Components Needed:

- Raspberry Pi (with Raspbian OS)
- Ultrasonic Distance Sensors (HC-SR04)
- LEDs
- Breadboard and jumper wires
- 

### Python Libraries Required:

- [RPi.GPIO](#) (for controlling GPIO pins)
- `time` (for adding delays)
- `threading` (for parallel execution)

## python code for traffic management system

```
import RPi.GPIO as GPIO
import time
import threading
```

```
# GPIO Pins for sensors and LEDs
```

```
SENSOR1_TRIG = 17
```

```
SENSOR1_ECHO = 18
```

```
SENSOR2_TRIG = 23
```

```
SENSOR2_ECHO = 24
```

```
RED_LED = 25
```

```
GREEN_LED = 8
```

```
# Setup GPIO pins
```

```
GPIO.setmode\(GPIO.BCM\)
```

```
GPIO.setup(SENSOR1_TRIG, GPIO.OUT)
```

```
GPIO.setup(SENSOR1_ECHO, GPIO.IN)
```

```
GPIO.setup(SENSOR2_TRIG, GPIO.OUT)
```

```
GPIO.setup(SENSOR2_ECHO, GPIO.IN)
```

```
GPIO.setup(RED_LED, GPIO.OUT)
```

```
GPIO.setup(GREEN_LED, GPIO.OUT)
```

```

def measure_distance(trigger_pin, echo_pin):
    GPIO.output(trigger_pin, True)
    time.sleep(0.00001)
    GPIO.output(trigger_pin, False)

    start_time = time.time()
    end_time = time.time()

    while GPIO.input(echo_pin) == 0:
        start_time = time.time()

    while GPIO.input(echo_pin) == 1:
        end_time = time.time()

    pulse_duration = end_time - start_time
    distance = pulse_duration * 17150 # Speed of sound in cm/s
    return round(distance, 2)

def traffic_lights_controller():
    while True:
        distance1 = measure_distance(SENSOR1_TRIG, SENSOR1_ECHO)
        distance2 = measure_distance(SENSOR2_TRIG, SENSOR2_ECHO)

        if distance1 < 30 or distance2 < 30:
            # Traffic is detected, stop one direction
            GPIO.output(RED_LED, GPIO.HIGH)
            GPIO.output(GREEN_LED, GPIO.LOW)
        else:
            # No traffic, let one direction go
            GPIO.output(RED_LED, GPIO.LOW)
            GPIO.output(GREEN_LED, GPIO.HIGH)

        time.sleep(1)

try:
    # Create and start the traffic lights controller thread
    traffic_thread = threading.Thread(target=traffic_lights_controller)
    traffic\_thread.start()

    # Main program loop
    while True:
        pass

```

except KeyboardInterrupt:

# Cleanup on Ctrl+C

[GPIO.cleanup\(\)](#)

In this code:

- We initialize the GPIO pins for the ultrasonic sensors and LEDs.
- The measure\_distance function calculates the distance using ultrasonic sensors.
- The traffic\_lights\_controller function monitors the distance from the sensors and controls the traffic lights accordingly.
- We use threading to run the traffic lights controller in parallel with the main program.

## **Advantages of a Smart Traffic Management System(IOT)**

Cleaner, greener, safer, and more accessible roads are a few benefits of implementing IoT and intelligent technology.

It helps with the following:

- ✓ Reducing traffic jams and accidents on the streets
- ✓ Ensuring immediate clearance for emergency vehicles
- ✓ Facilitating safer and shorter commute times
- ✓ Reducing congestion & energy consumption at intersections
- ✓ Offering significant productivity benefits with real-time monitoring of crucial infrastructures
- ✓ Reducing operating costs with efficient traffic management processes
- ✓ Ensuring compliance with the regulations for reducing the carbon footprint

## **Disadvantages of traffic management system in IOT**

**Vulnerability to Cyberattacks:** IoT-based TMS is susceptible to cyberattacks, which can disrupt traffic control systems and compromise data security.

**Privacy Concerns:** Continuous data collection from sensors and cameras can raise privacy concerns among the public.

**Initial High Costs:** Implementing an IoT-based TMS can involve significant upfront costs for hardware, infrastructure, and technology deployment.

**Maintenance Challenges:** Regular maintenance of IoT devices and infrastructure is necessary to ensure system reliability, which can be costly and time-consuming.

**Data Accuracy Issues:** The accuracy of data collected by IoT devices can be affected by various factors, potentially leading to incorrect traffic management decisions.

**Dependence on Technology:** Reliance on technology makes TMS vulnerable to technical failures, glitches, or power outages.

**Digital Divide:** Not all areas may have equal access to advanced IoT-based TMS, exacerbating traffic management disparities.

**Environmental Impact:** The production and disposal of IoT devices can have negative environmental consequences.