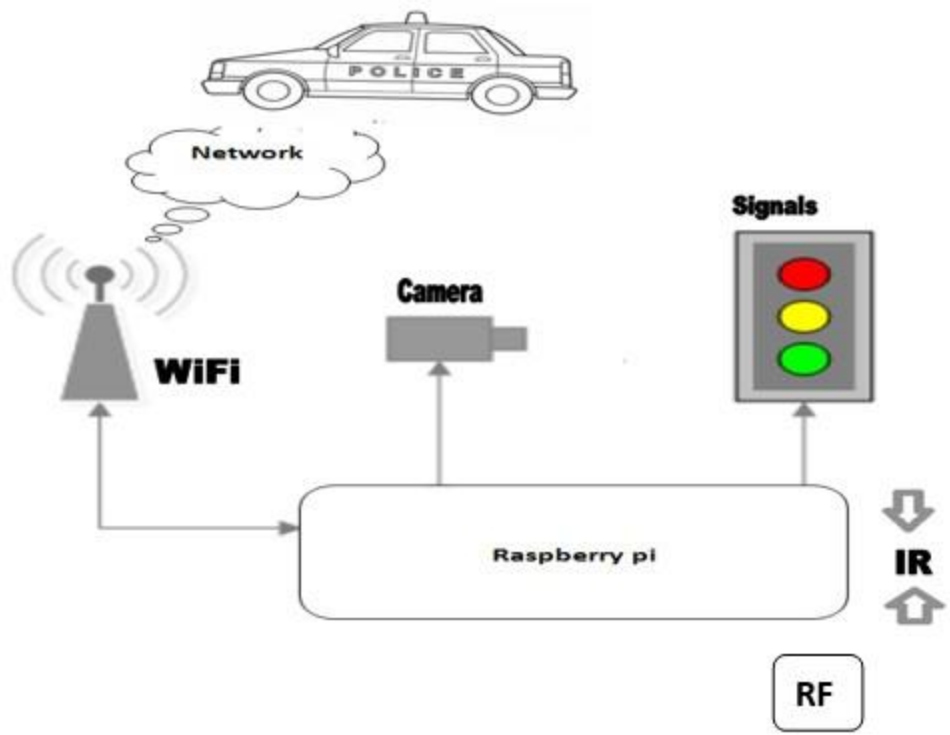
IOT BASED TRAFFIC MANAGEMENT

ABSTRACT

- In the contemporary world, urban mobility is one of the unprecedented challenges to be tackled in the administration of a big city. This paper analyses the ever growing urban population around the globe and discusses about the traffic systems in densely populated cities like Los Angeles and Amsterdam. Further, an advanced traffic management system is proposed, implemented using Internet of Things IOT. The system is supported by a circuit embedded in the vehicle, which operates using RFID with clustered systems. The functionalities of the system include efficient traffic light control, parking space identification and anti-theft security mechanism. The proposed architecture and working with big data analytics involving Hadoop is presented. Moreover, supervised learning methodologies are proposed that would help in determining the standard of roads, estimating overall traffic flow, calculating average speed of distinct vehicle types on a road and travel path of a vehicle.

WORKING PRINCIPLE

- There will 8 sensors across the 4 lanes with each lane having 2 sensors each to give the data how much dense the lane .If the entire lanes have less traffic the system will work normally means there lanes sequence will be First A lane then D lane then C lane and at last D lane.


POLICE
Network
Signals
Camera
WiFi
Raspberry pi
IR
RF

PROBLEM STATEMENT

- Traffic congestion problems consist of incremental delay, vehicle operating costs such as fuel consumption, pollution emissions and stress that result from interference among vehicles in the traffic stream, particularly as traffic volumes approach a road's capacity.

TRAFFIC MANAGEMENT SOLUTION

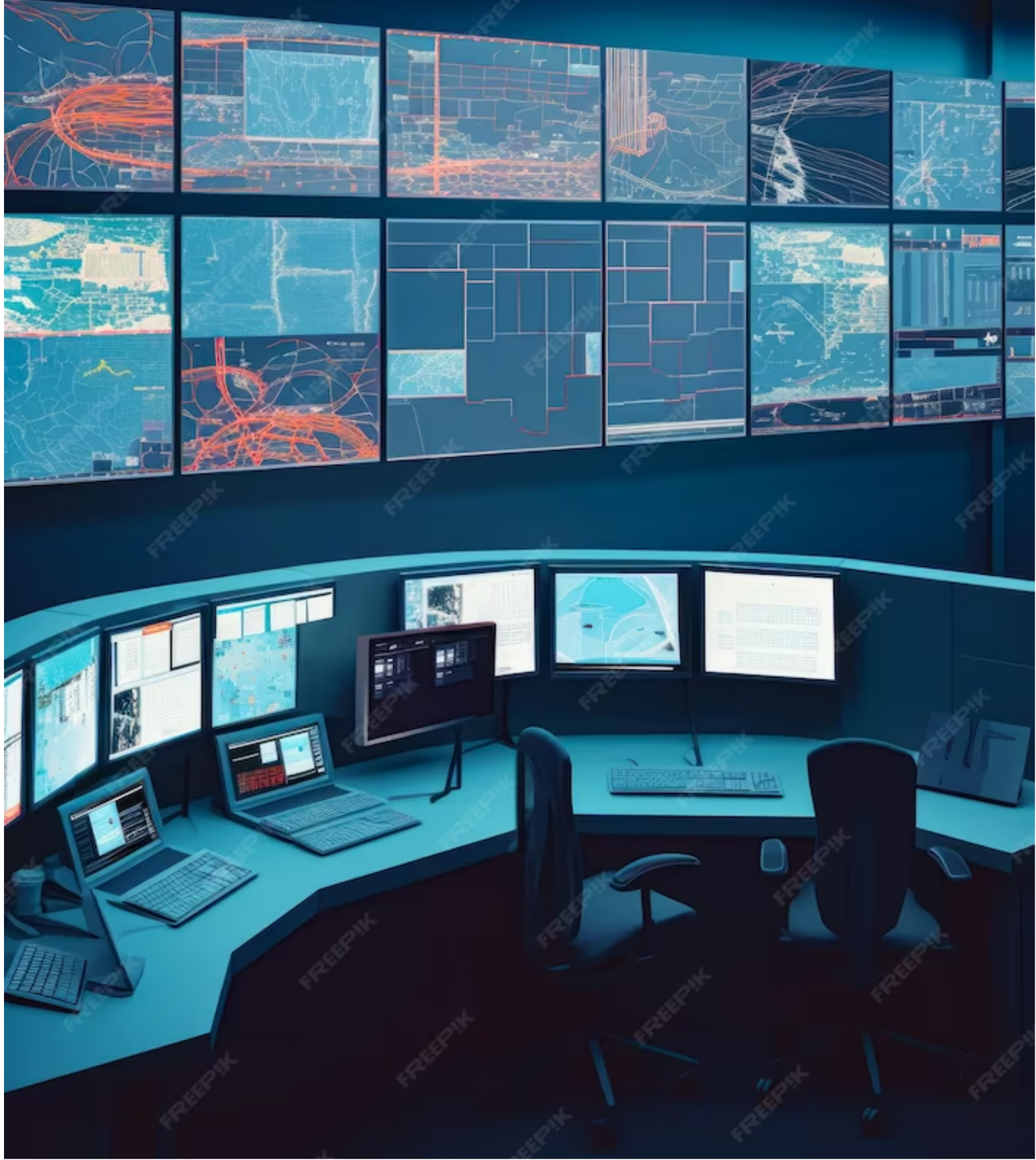
- An Internet of Things (IOT)-enabled intelligent traffic management system can solve pertinent issues by leveraging technologies like wireless connectivity & intelligent sensors. Considered a cornerstone of a smart city, they help improve the comfort and safety of drivers, passengers & pedestrians.

IoT Sensors for Traffic Management

IoT sensors can be used to collect real-time data on traffic flow, vehicle speeds, and road conditions. This data can be analyzed to optimize traffic light timings, reroute traffic, and alert drivers of hazards. By leveraging IoT sensors, traffic management can become more efficient and proactive.

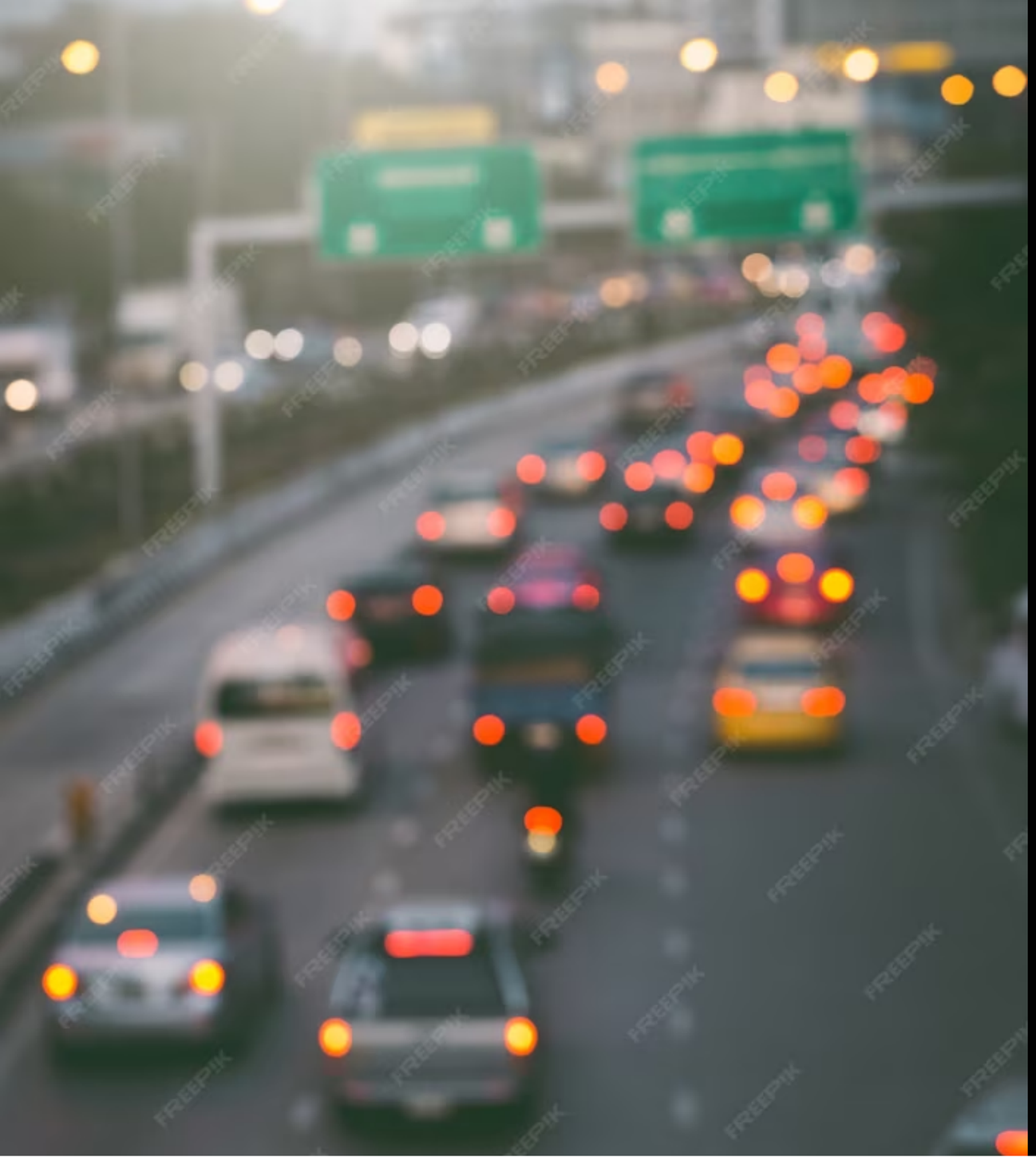
Smart Traffic Management Systems

Smart traffic management systems use IoT data to optimize traffic flow, reduce congestion, and improve safety. These systems can automatically adjust traffic light timings based on real-time traffic conditions, reroute traffic to avoid accidents or congestion, and provide drivers with real-time alerts and information.



Enhancing Road Safety with IoT

IoT can enhance road safety by providing real-time data on road conditions, weather, and accidents. This data can be used to alert drivers of hazards, reroute traffic to avoid accidents, and provide emergency services with real-time information. By leveraging IoT, road safety can be improved for drivers, pedestrians, and cyclists.



Sustainability Benefits of IoT for Traffic Management

IoT can help reduce traffic congestion and emissions by optimizing traffic flow and reducing idling times. By reducing congestion, IoT can also improve the overall efficiency of transportation networks. This can lead to reduced fuel consumption and emissions, making traffic management more sustainable.

IOT BASED TRAFFIC MANAGEMENT

Hardware Components Needed:

- Raspberry Pi (Raspberry Pi 3 or 4 recommended)
- Ultrasonic distance sensors (HC-SR04 or similar)
- Jumper wires
- Breadboard (optional)

Python Code for Traffic Monitoring:

```
import RPi.GPIO as GPIO
import time

# Configure GPIO pins for sensors
TRIG = 18  # Ultrasonic sensor trigger pin
ECHO = 24  # Ultrasonic sensor echo pin

GPIO.setmode(GPIO.BCM)
GPIO.setup(TRIG, GPIO.OUT)
GPIO.setup(ECHO, GPIO.IN)

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

    while GPIO.input(ECHO) == 0:
        pulse_start = time.time()
```

```
  while GPIO.input(ECHO) == 1:
        pulse_end = time.time()

    pulse_duration = pulse_end - pulse_start
    distance = pulse_duration * 17150  # Speed of sound = 343 m/s

    return round(distance, 2)

try:
    while True:
        distance = measure_distance()
        print("Distance:", distance, "cm")
        time.sleep(1)

except KeyboardInterrupt:
    GPIO.cleanup()
```

Example Output:
Assuming that the ultrasonic sensor detects a vehicle within its range, the output would look like this

Distance: 15.23 cm
Distance: 14.91 cm
Distance: 15.67 cm
…

Introduction

A smart traffic management system utilizing cameradata, communication and automated algorithms is to be developed to keep traffic flowing more smoothly. The aim is to optimally control the duration of green or red light for a specific traffic light at an intersection. The traffic signals should not flash the same stretch of green or red all the time, but should depend on the number of vehicles present. When traffic is heavy in one direction, the green lights should stay on longer; less traffic should mean the red lights should be on for a longer time interval

System design

1) Raspberry Pi
2) LED lights which are used for the purpose of signaling.
3) Traffic cameras which are used for monitoring traffic.
4) Node MCU Microcontroller

Block diagram

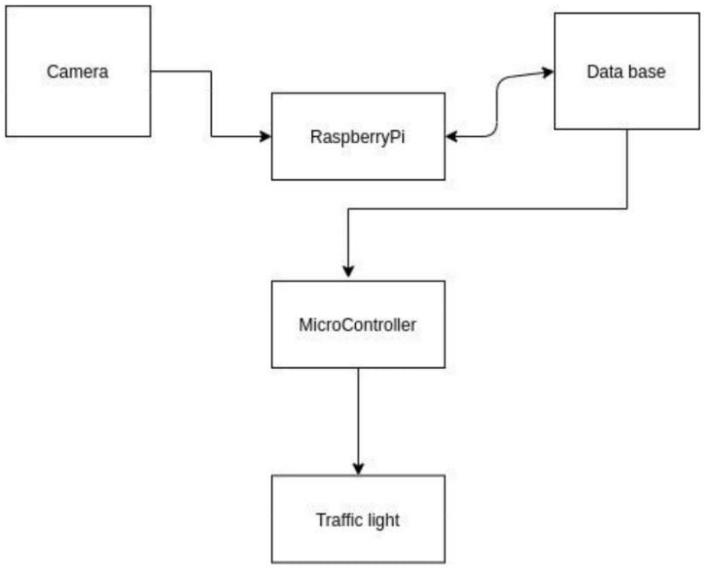


Fig. 1 - Flow Chart

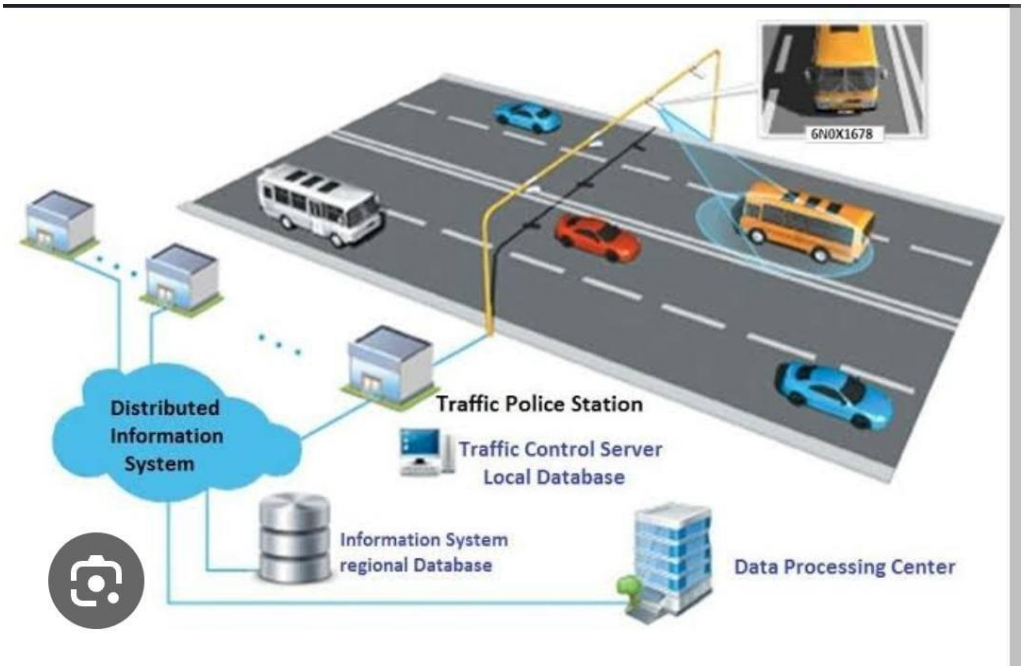
SYSTEM IMPLEMENTATION

Steps in the proposed system for controlling traffic light 1:

1. Camera: Continuously record traffic video.
2. Read Image: Read frames of the traffic image.
3. Grayscale Image Conversion: It converts color image to grayscale image. This method is based on different color transforms. According to the R, G, B value in the image, it calculates the value of grayscales and converts the image into a grayscale image.
4. Image Binarization: Grayscale image is converted into black and white image.
5. Traffic Signal Control: Based on vehicle count signal timings are changed and the respective LED glows.

Steps for controlling traffic light 2:

1. Initialize System.
2. Configure ESP 8266 module for multi access point through AT commands.
3. Connect WI-FI module to WI-FI network.
4. Start UDP local port in WI-FI module.
5. Establish UDP connection to Raspberry pi.
6. Wait for data.
7. Change traffic light signal 2 depending upon their received data from raspberry Pi.



Components

Smart traffic management system consists the Following components.

- Radio signal detector
- Radio waves transmitter
- Ultra-sonic sensor/Hall Effect sensor
- Raspberry Pi
- Python programming
- Light Emitting Diode

Radio signal detector

The transmitter was turned on and off to create short or extensive stretches of radio waves, illuminating messages in various codes, similar to that of Morse code. In this manner, the early radio recipients had just a single application or function that was to separate between the nearness or nonappearance of any radio sign. The gadget that played out this specific capacity was known As a locator.

The parts of a radio detector are: -

• Antenna: It helps in catching the radio waves. Commonly, the reception apparatus is basically a long wire. At the point when this wire is vulnerable to radio waves, the waves cause a little substituting current (AC) inside the receiving wire.

•RF enhancer: It is a sense speaker that enhances the exceptionally weak radio recurrence signal from the reception apparatus with the goal that the sign can be prepared by the tuner.

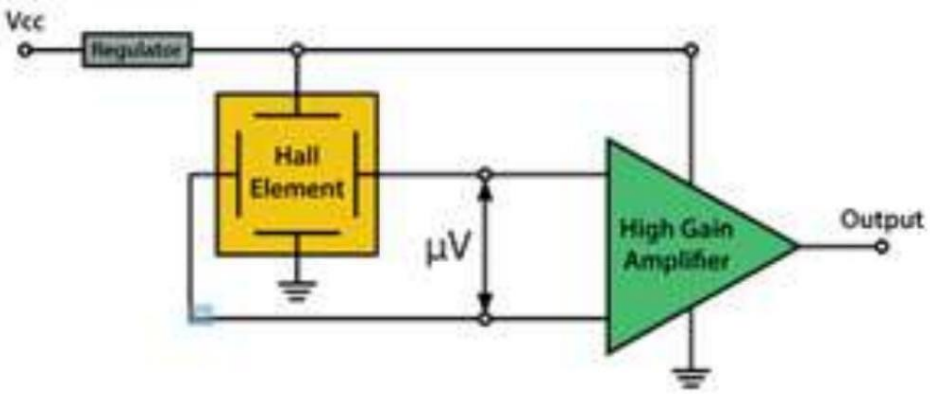
•Tuner: A circuit that can pull back sign of a specific recurrence from a blend of sign of various frequencies.

• Detector: This part is answerable for isolating the sound data from the transporter wave. For AM (Amplitude balance) flag, this can be satisfied with the assistance of a diode that just amends the rotating current sign.

• Audio speaker: The motivation behind this part is to enhance the powerless sign that originates from the identifier with the aim that it tends to be heard by anybody

Hall Effect Sensor

The Hall Effect is the most common method of measuring magnetic field and the Hall Effect sensors are very popular and have many contemporary applications. For example, they can be found in vehicles as wheel speed sensors as well as crankshaft or camshaft position sensors.The basic Hall Element of the Hall Effect magnetic sensors mostly provides very small voltage of only a few micro volts per Gauss, so therefore, these devices are usually manufactured with built-in high gain amplifiers.



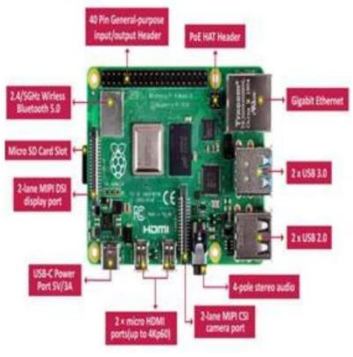
Raspberry PI

The Raspberry Pi is a small sized personal computer (PC) which is structured and fabricated by the Raspberry Pi Foundation (a non-benefit association) which is dedicated to making PCs and programming guidelines as effectively open as conceivable to the intended interest group. Software engineers over the world have taken the modest stage for ventures which are from reproducing setting up modest however amazing home media gadgets.

Advantages

- It is a solitary board PC
- It is very cost effective.

Raspberry PI



Light-Emitting Diode

A light-producing diode is a semiconductor which has a light source that conveys light when current is permitted to move through it. Electrons in the semiconductor join with the electron gaps, along these lines discharging vitality as photons. LEDs have numerous points of interest over other radiant light sources.

They are:

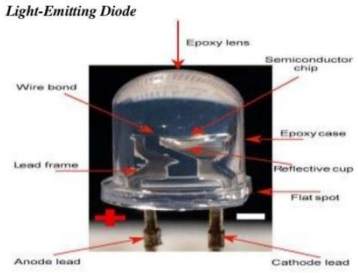
•Lower vitality utilization

• Longer lifetime

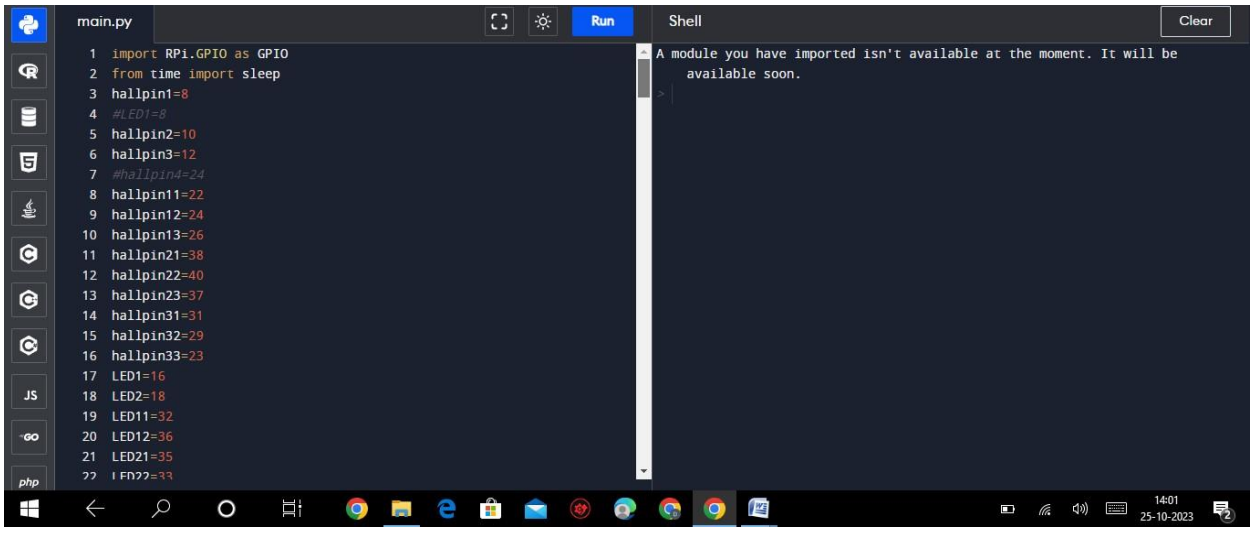
•Improved strength

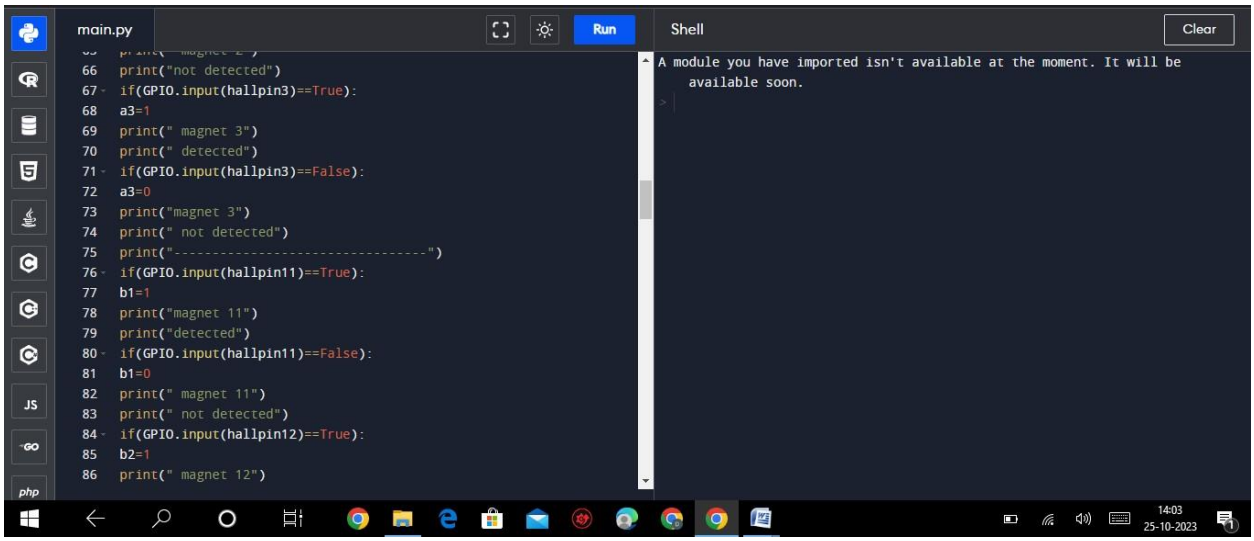
•Small in size

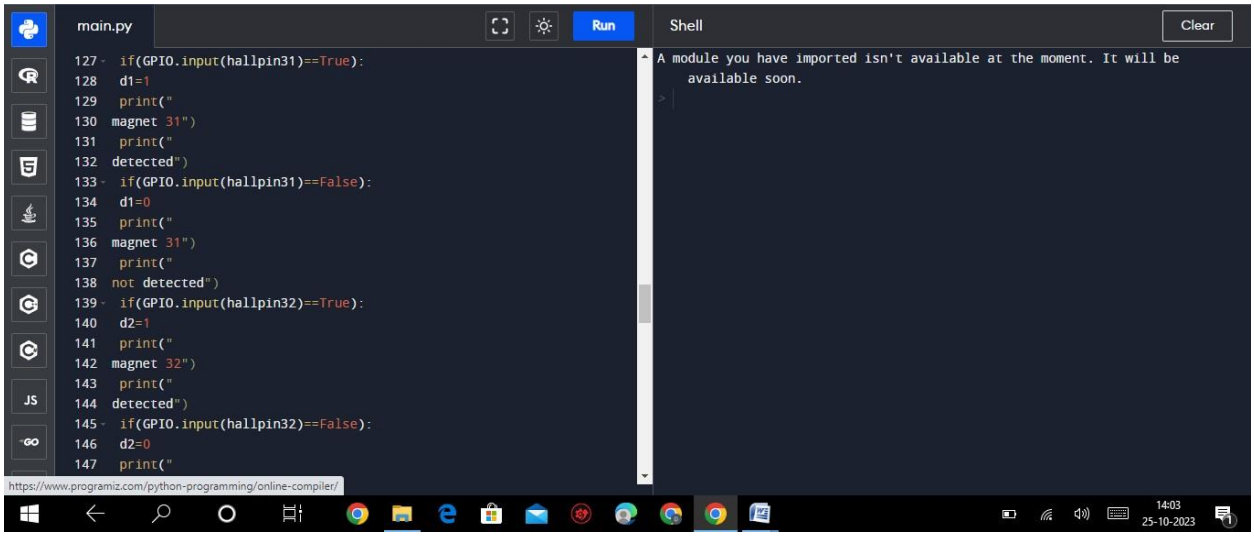
• Faster rate of exchanging



Program




main.py
Run
Shell
Clear
print("not detected")
if(GPIO.input(hallpin3)==True):
a3=1
print(" magnet 3")
print(" detected")
if(GPIO.input(hallpin3)==False):
a3=0
print("magnet 3")
print(" not detected")
print("--------------------------------")
if(GPIO.input(hallpin11)==True):
b1=1
print("magnet 11")
print("detected")
if(GPIO.input(hallpin11)==False):
b1=0
print(" magnet 11")
print(" not detected")
if(GPIO.input(hallpin12)==True):
b2=1
print(" magnet 12")
A module you have imported isn't available at the moment. It will be available soon.
14:03
25-10-2023


main.py
Run
Shell
Clear
if(GPIO.input(hallpin31)==True):
d1=1
print("
magnet 31")
print("
detected")
if(GPIO.input(hallpin31)==False):
d1=0
print("
magnet 31")
print("
not detected")
if(GPIO.input(hallpin32)==True):
d2=1
print("
magnet 32")
print("
detected")
if(GPIO.input(hallpin32)==False):
d2=0
print("
A module you have imported isn't available at the moment. It will be available soon.
https://www.programiz.com/python-programming/online-compiler/
14:03
25-10-2023

Output

Moderate traffic. Alternating signals.

High traffic. Red signal for the main road.

Low traffic. Green signal for the main road. Moderate traffic. Alternating signals.

Moderate traffic. Alternating signals.

High traffic. Red signal for the main road.

High traffic. Red signal for the main road.

High traffic. Red signal for the main road.

Conclusion

Smart Traffic Management System has been developed by using multiple features of hardware components in IoT. Traffic optimization is achieved using IoTplatform for efficient utilizing allocating varying time to all traffic signal according to available vehicles count in road path.