

TRAFFIC MANAGEMENT

Name	Register No :	Nan muthalvan ID
S. Aarthi	811221106001	au811221106001
M. Arockia Jenifer	811221106003	au811221106003
C. Thamilarasi	811221106033	au811221106033
S. Vellaisamy	811221106036	au811221106036
C. Dhinesh Babu	811221106008	au811221106008

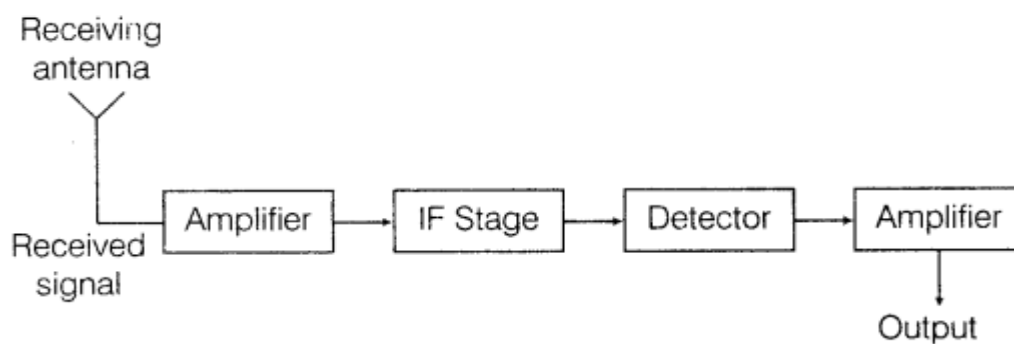
ABSTRACT:

- Traffic congestion is a major problem in many cities of India along with other countries.
- Failure of signals, poor law enforcement and bad traffic management has lead to traffic
- congestion. One of the major problems with Indian cities is that the existing
- infrastructure cannot be expanded more, and thus the only option available is better
- management of the traffic. Traffic congestion has a negative impact on economy, the
- environment and the overall quality of life. Hence it is high time to effectively manage
- the traffic congestion problem. There are various methods available for traffic
- management such as RFID, infrared sensors, inductive loop detection, wireless sensor
- network, etc. All these methods are effective methods of smart traffic management. But
- the problem with these systems is that the installation time, the cost incurred for the
- installation and maintenance of the system is very high. Hence a new technology using
- image processing with IOT is introduced which can be coupled with the existing
- signaling system that can act as a key to smart traffic management in real time. This
- new technology which will require less time for installation with lesser costs as
- compared to other methods of traffic congestion management. Use of this new
- technology will lead to reduced traffic congestion.

COMPONENTS:

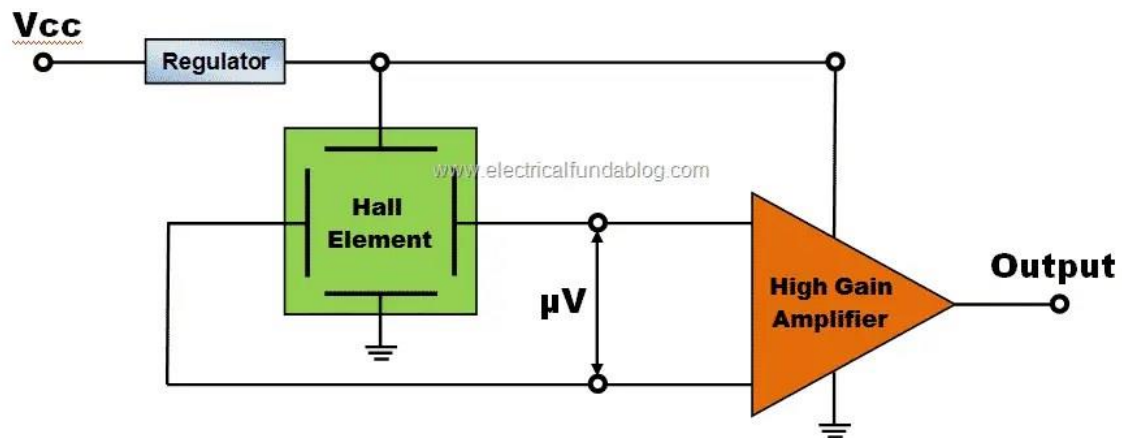
- Radio signal detector
- Radio waves transmitter
- Ultra-sonic sensor/Hall Effect sensor
- Raspberry Pi
- Python programming
- Light Emitting Diode

BLOCK DIAGRAM:

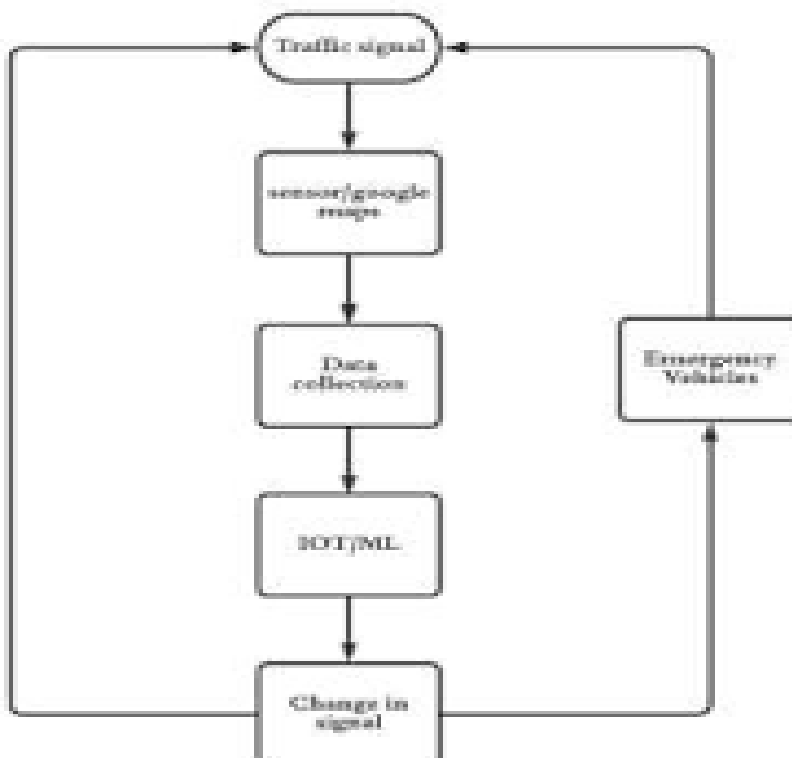


HALL EFFECT SENSOR:

Analog Output Hall Effect Sensor



PROCESS FLOW CHART:

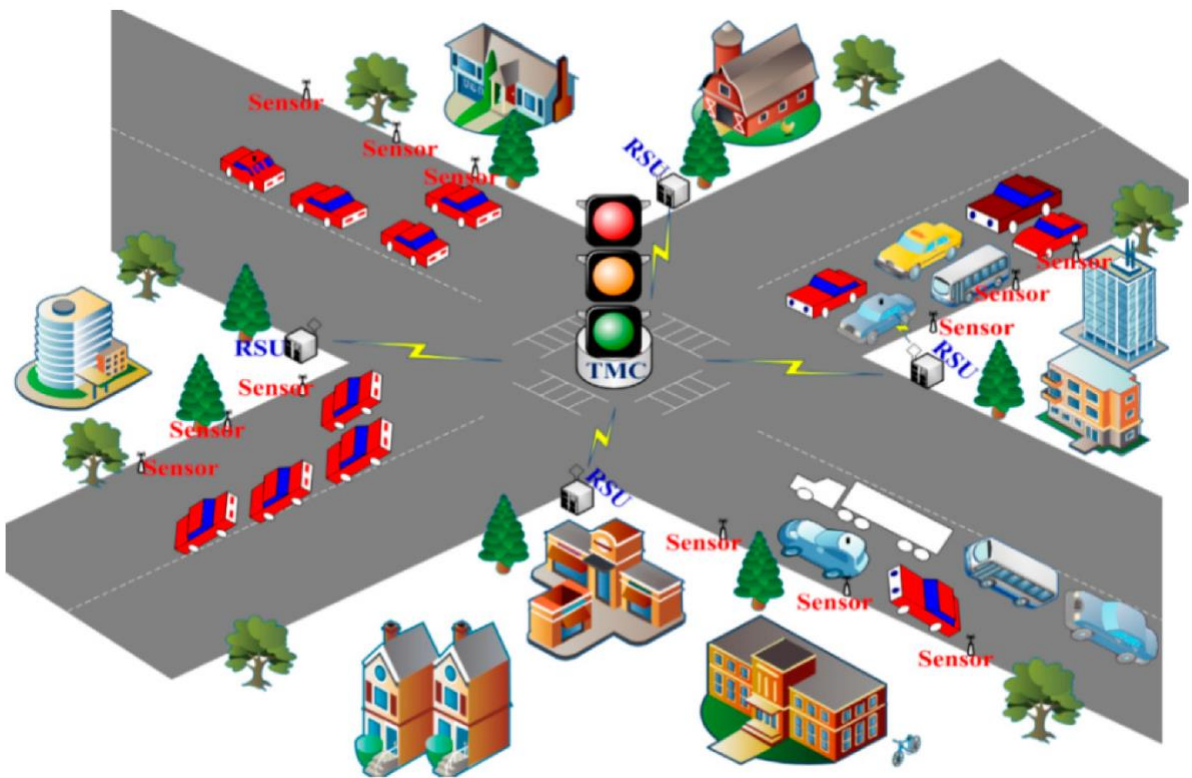


PROGRAM :

```
f = open("out.txt", "r")
no_of_vehicles=[]
no_of_vehicles.append(int(f.readline()))
no_of_vehicles.append(int(f.readline()))
no_of_vehicles.append(int(f.readline()))
no_of_vehicles.append(int(f.readline()))
baseTimer = 120 # baseTimer = int(input("Enter the base timer value"))
timeLimits = [5, 30] # timeLimits = list(map(int,input("Enter the time limits ").split()))
print("Input no of vehicles : ", *no_of_vehicles)

t = [(i / sum(no_of_vehicles)) * baseTimer if timeLimits[0] < (i / sum(no_of_vehicles)) * baseTimer <
timeLimits[1] else min(timeLimits, key=lambda x: abs(x - (i / sum(no_of_vehicles)) * baseTimer)) for i
in no_of_vehicles]
print(t, sum(t))
```

WORKING PROTOTYPE:



CONCLUSION:

*We are working with a python code to implement the above setup.
The python code will be uploaded in the upcoming phases of the
project.*

THANK

YOU