TRAFFIC MANAGEMENT system using IOT

Abstract — *Traffic is a major obstacle faced by all metropolitan cities; this is due to the exponential increase in the number of vehicles on the road but the infrastructure for road transportation remains the same. The average travelling time consumption by people is increasing year by year for the same given way points. Most of the cities still rely on conventional traffic signaling which is controlled manually or time based. This conventional system used around the world is not efficient as it lacks useful data from reliable real-time sources to clear the way for emergency vehicles (ambulances, firetrucks and police vehicles) during heavy traffic conditions. Manual interaction involves manual errors, which leads to more fuel consumption and leads to health issues. Due to this Traffic flowing to a junction from all the directions at a given time is unequal. A smart traffic management is a system, where traffic is controlled by the management system, which controls the traffic lights in accordance with the real time situation of traffic moving from all different directions in a junction. This real time data is collected either from google maps (future work) or from various sensors placed at equal intervals of distance at a junction. This data is collected and brought to a control system which autonomously calculates the optimum time for the release of the green signal.*

We are aiming to solve the issue of traffic by efficiently controlling the signaled intersections in cities by presenting present an algorithm based on comparative real time data analysis using IOT [2]. Considering this review, we identify a range of important possibilities for contributions to traffic management, detection technology and flexible optimization techniques that use various kinds of automated learning.

Keywords — *smart traffic, IOT, Emergency vehicle, autonomous management, google maps.*

I. INTRODUCTION

Traffic flowing to a junction from all the directions is most of the time unequal. This factor is not taken into account by the conventional traffic management system leading to unbalanced distribution of traffic from all directions which leads to air pollution and sever health issues, as there are around 67 lakhs on vehicles in single metropolitan city like Bangalore where people are facing this issue.

A smart traffic management system is, where traffic is constrained by the administration framework, which controls the traffic lights as per the continuous circumstance of traffic moving from every extraordinary bearing in an intersection. This continuous information is gathered from different sensors set at equivalent interims of separation at an intersection. This information is gathered and brought to a control framework which independently ascertains the ideal time for the arrival of the green sign to every specific heading in an intersection so as to counteract traffic heaping up[1].

The subsequent significant issue which we are taking a gander at in this venture is about the development of crisis vehicles (fire engines, ambulances and police vehicles) in packed intersections of a city. Because of the absence of foundation present in our nation to address the issue, concerning the speedy development of crisis vehicles in a city, has carried us to the decision about making a self-governing framework to help the crisis vehicles to travel through the city without issue

In this framework, we are proposing a powerful technique wherein these vehicles can be recognized by utilizing radio wave flagging strategy. The crisis vehicles will be distinguished and will be the included inside a framework that discharges a SOS signal which can be identified by a sign discovery unit which sends a crisis trigger to the traffic the executive's framework. In this framework, the crisis vehicle will be recognized around 1 kilo meter from the sign and when it arrives at 500 meters from the sign then the control unit gives green sign toward the path in which the crisis vehicle is drawing closer so as to give a sign free hall and to avert the crisis vehicle stalling out an automobile overload. An order of precedence is set if there is more than one emergency vehicle approaching from many directions. The order of precedence is given as follows:

- Ambulance
- Fire truck
- Police vehicles

The order of precedence also takes into account the factor that includes a combination of more than two types of emergency vehicles approaching the signal from the same direction then the trigger is fired and

the order of precedence is given to this particular direction compared to the direction a single emergency vehicle is approaching.
This provides an effective way to provide a signal free path autonomously to the emergency vehicles that presently do not have any proper infrastructure for the signal free movement of emergency vehicles.

The proposed plan was to use ultra-violet or proximity sensors at a stipulated distance at each side in accordance to the vehicle flow rate at each side. However, the disadvantage of this methodology is that it only provides us with limited amount of data. Using this method, the sensors can only provide us the data about whether or if there is presence of vehicle at the position where the sensors are placed. The accuracy could highly depend on the distance between the two sensors. The exact distance of the traffic jam cannot be predicted using ultra-violet sensors or proximity sensors. Therefore, this proposed method was found not effective.

Thus, to overcome the many disadvantages of the above proposed plan and to get more accurate results, GPS tracking data is utilized to determine real-time traffic situation. References like Google maps and Bing maps are used to calculate the exact value of the vehicle flow rate on each road. Hence, using this accurate data, the calculations done are more precise and optimum time for the release of signal can be calculated precisely.

This method will provide us a vast number of advantages as this could help us save on the cost of procurement of ultra-violet proximity sensors that have to be placed at particular intervals on each road leading to an intersection. This can also help us reduce hardware complexity and data handling as this method can provide with simpler and organized form of data which can be easily utilized by the traffic management system.

Therefore, this data provided is much more accurate and precise when compared to the previous system.

Radio waves are a kind of electromagnetic radiation which has wavelength in the electromagnetic range which is longer than the infrared light. Radio waves have frequencies as high as 300 GHz to as low as 30 Hz. They are created by electric charges experiencing quickening. Normally happening radio waves are produced by lightning and astronomical objects.

II. COMPONENTS

Smart traffic management system consists the Following components[6].

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

Radio signal detector

In a radio, the device that receives and takes the information from a modulated radio frequency current or voltage is called a detector.
While differentiating to the present-day radio stations which transmit sound (for instance: a sound signal) on an undisturbed bearing wave, radio stations in the early days at the outset transmitted data by radiotelegraphy. 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.

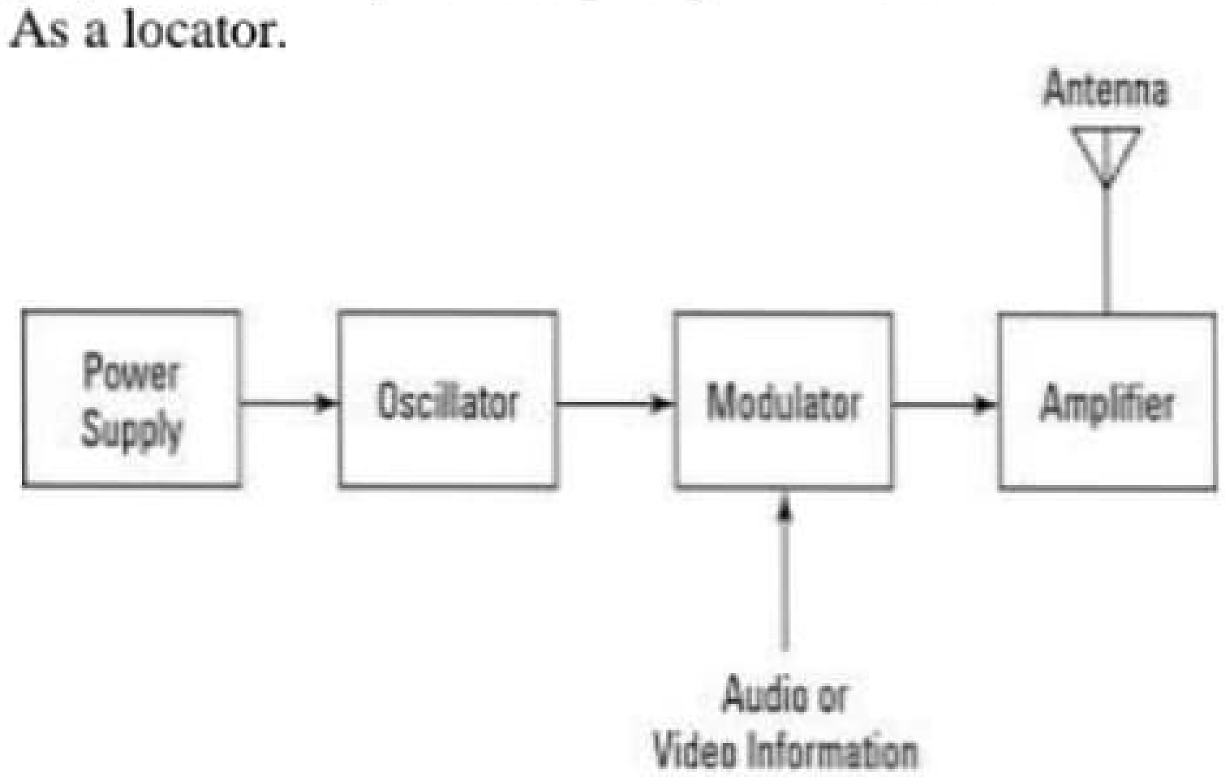

Fig.1 sensor working principle

The collector utilizes a receiving wire so it can catch radio waves, process these waves to retain just those waves that are vibrating at the necessary recurrence, extricates the sound flag that were added to those waves, intensifies the sound sign, lastly plays them on a speaker.
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. This can be satisfied utilizing a basic transistor intensifier circuit.

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.

If we bring some magnetic field near the plate we would disturb the straight flow of the charge carriers due to a force, called Lorentz Force. In such a case the electrons would deflect to one side of the plate and the positive holes to the other side of the plate. This means if we put a meter now between the other two sides we will get some voltage which can be measured.

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.

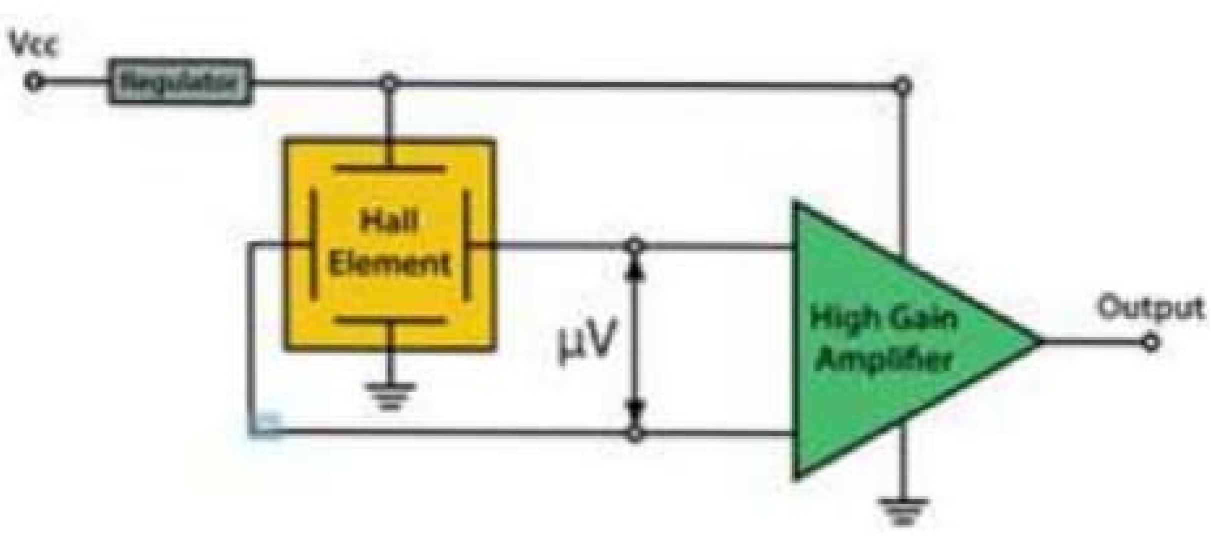

Fig.2-Hall effect sensor

There are two types of Hall Effect sensors, one providing analog and the other digital output. The analog sensor is composed of a voltage regulator, a Hall Element and an amplifier. The digital output sensors provide just two output states, either "ON" or "OFF".

Advantages

- High speed operation over 100 KHz possible. Whereas at high frequencies the inductive or capacitive sensor output begins to distort.
- Non contact operation so there is no wear and friction, hence unlimited number of operating cycles.
- When packed immune to dust, air, water where as capacitive sensor may get triggered by dust.
- It can measure zero speed.
- Highly repeatable operation.
- Capable of measuring large current.

Disadvantages

- It may be affected by external interfering magnetic field.
- Large temperature drift.
- Large offset voltage

Raspberry PI

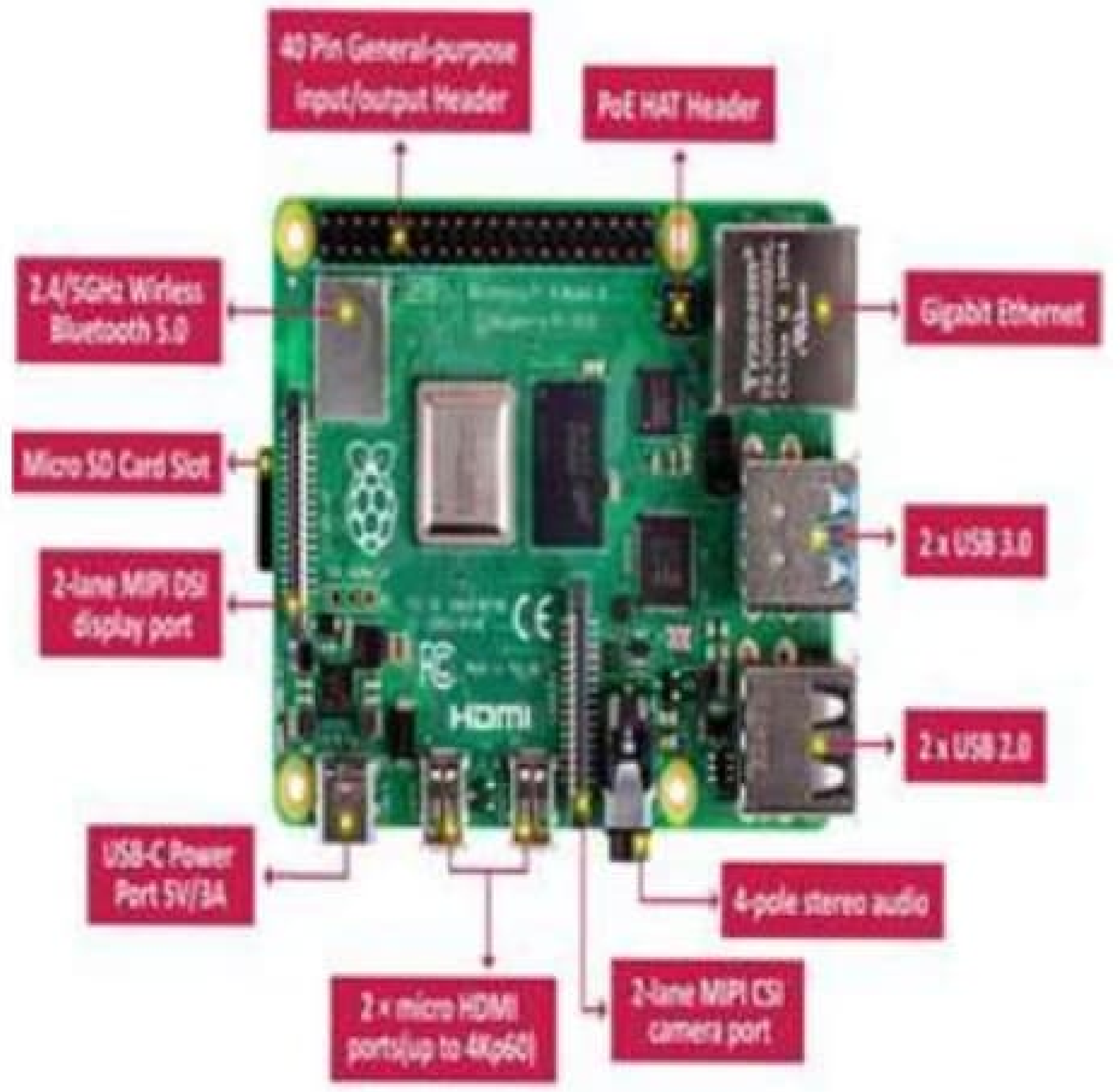

Fig.3-Raspberry pi

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.

Despite the fact that the first aim of the Raspberry Pi venture was to PCs with programming choices under the control of understudies, the Raspberry Pi has been taken by a various objective group of spectators. Software engineers over the world have taken the modest stage for ventures which are from reproducing retro formed cupboards to controlling robots and to setting up modest however amazing home media gadgets. Coming up next are the upsides of Raspberry Pi over PCs and comparable gadgets. Advantages

•It is a solitary board PC

•It is very cost effective.

•A completely fledged working framework and it very well may be utilized as an everyday PC.

•The availability of GPIO (General Purpose Input Output pins) recognizes a Raspberry Pi from conventional PCs. These pins can be associated with sensors and other outside parts which help in interfacing with them automatically a language, for example, Python or Java. This enables you to manufacture and model or any Internet of Things gadgets that can detect this present reality

•Newer models have Wi-Fi and Bluetooth options worked in. This permit bringing ventures into remote mode effectively.

Light-Emitting Diode

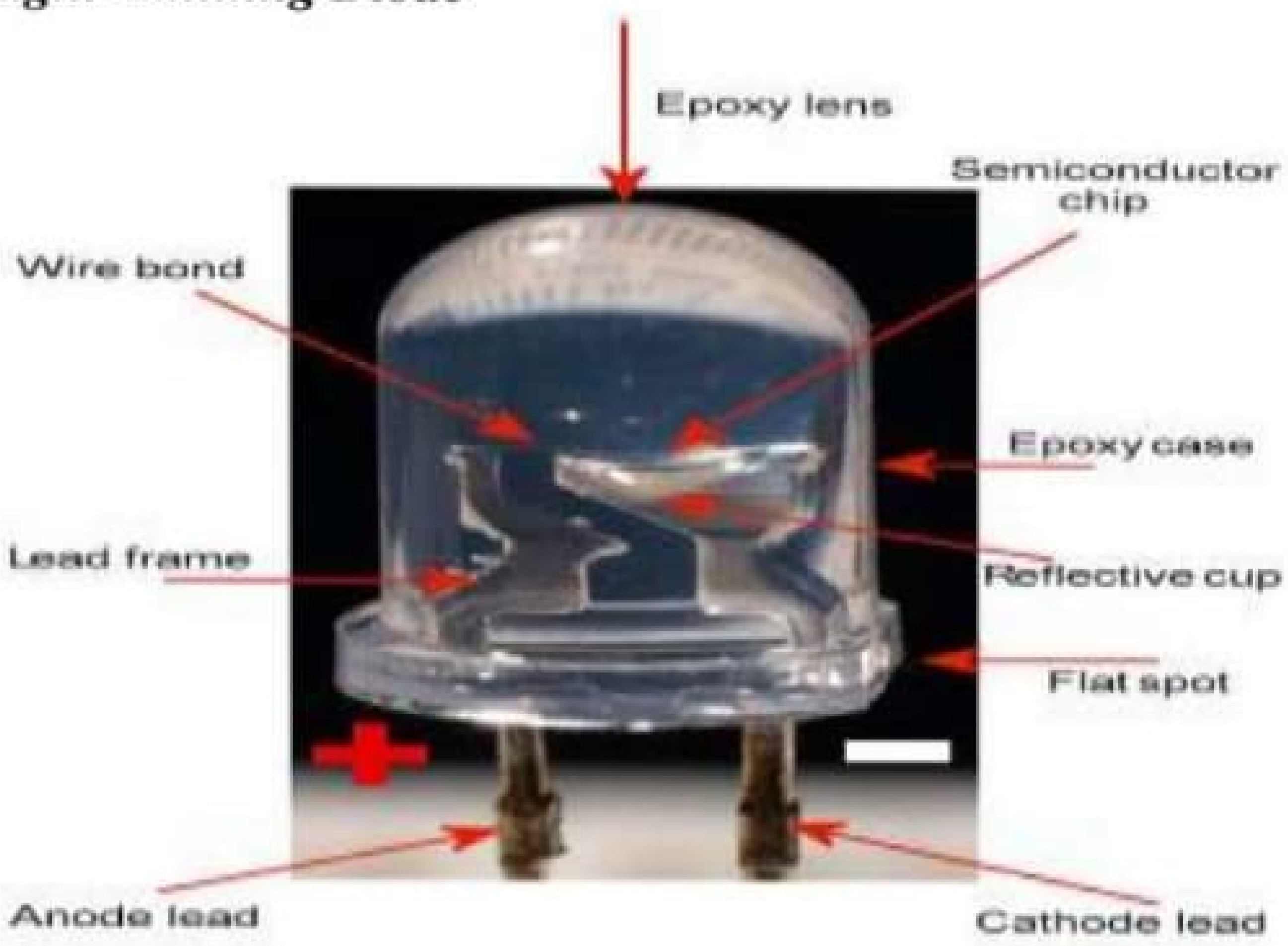

Fig. 4- Light emitting diode

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.

The shade of the light (comparing to the vitality obtained by the photons) is found by the vitality required for electrons to cross the vitality hole of the semiconductor. White light is gotten by utilizing a few different semiconductors. 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

In a light discharging diode, the recombination of electrons and electron openings in a semiconductor creates light (or infrared radiation). This procedure is designated "electroluminescence". The wavelength of the light is relied upon the hole of the vitality band in the kind of semiconductors utilized. Since these materials have a high list of refraction, plan highlights of the gadgets, for example, extraordinary optical coatings are important to emanate light effectively with low wastage.

III. METHODOLOGY

Basic flowchart[5]

- This is the basic ideology behind controlling the timing of traffic lights with respect to present time traffic conditions.
- The sensor collects data of the real-time density of vehicles present on the road.
- The data from the sensors are collected and stored in the cloud.
- This information is fed to the microcontroller which determines the change in signal for each lane.
- During the case of an emergency, the data is directly fed into the microcontroller hence terminating previous loop and changing the signal immediately.

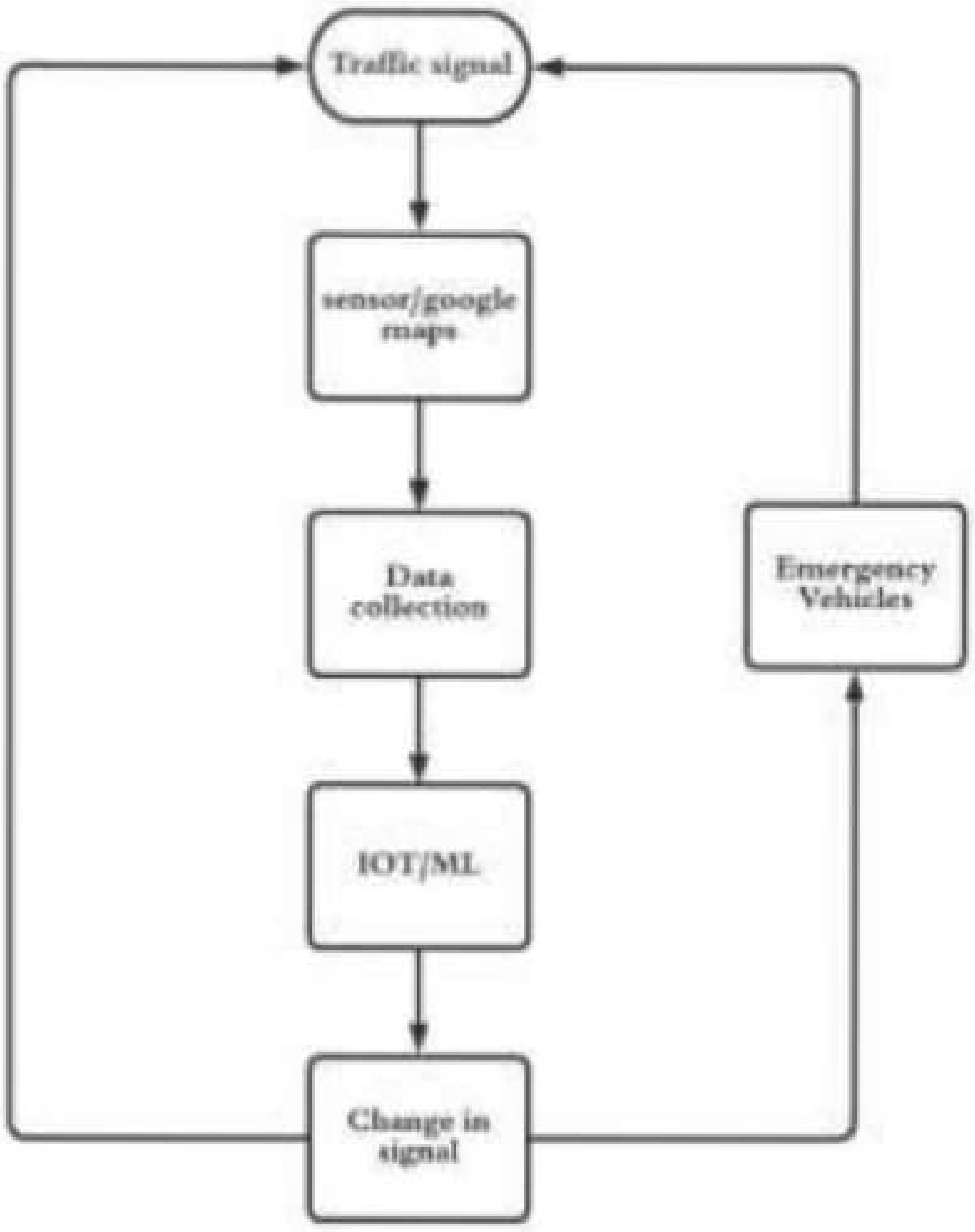

Fig. 5- Process flow chart

Complex flow chart

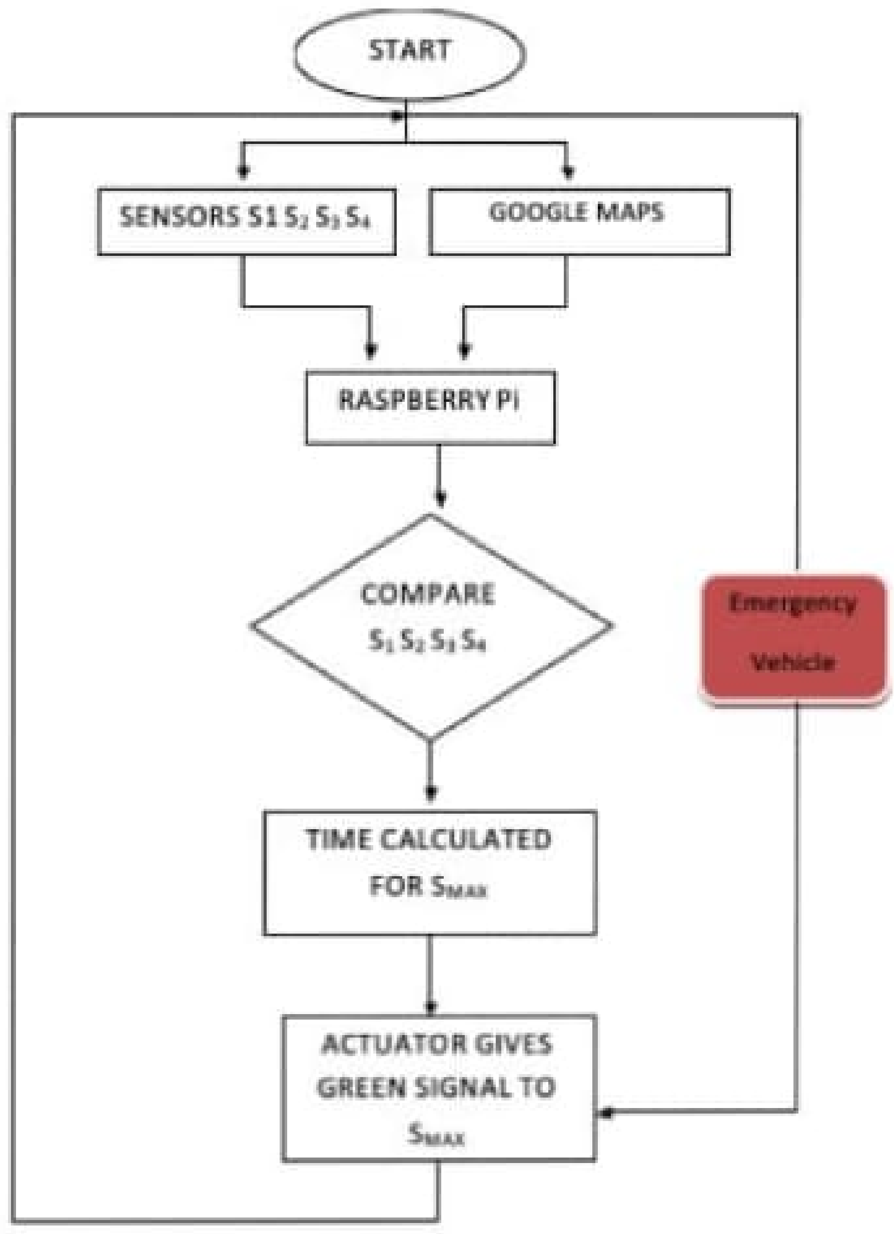

Fig. 6- data collection and priority line up

Proposed system

This traffic management control system consists of hall effect sensor, LED lights (red and green) and Raspberry Pi. The Raspberry Pi microcontroller contains the python programming code which controls time delay of led lights.

The Hall effect sensors placed on the road surface at regular intervals of distance to detect the presence of vehicles on the road till where it is placed. The government is suggested to place a radio wave transmitter on all emergency vehicles hence the emitted signal can be collected by the radio detector to identify the presence of any emergency vehicle[8]. The radio waves detector is placed at every junction in order to detect the presence of radio wave emitter that is placed inside the emergency vehicle. Raspberry Pi microcontroller is placed at every junction who collects the information from the various sensors placed at different locations and the resultant time is calculated and the LED lights are turned on by the microcontroller for the stipulated time calculated[3].

Phase-1:-

We are assuming 4 different breadboards to imitate a four-direction intersection junction and have placed 3 hall effect sensors on 3 sides along with one green light and red light on each side. All the terminals from the hall effect sensor and the Led lights are connected to the gpio pins of the raspberry pi. The instantaneous data of a particular moment is collected from all the sensors from every direction and the program compares and identifies the side having maximum traffic volume and the green light is given to the particular side with maximum volume and the other three sides will remain red. As the vehicles moves out, the moment the last active sensor detects the absent of vehicle, the program goes on to check which other side has maximum traffic. Considering there are four direction intersection junctions, once the green light is given to side from next iteration onwards preference is given to the other three side[9]s. During the third cycle preference is given to the last two sides and the onwards.

Phase-2:-

When the autonomous managing traffic control system is switched on, it collects all the information from various sensors like the ultrasonic detectors and radio wave detectors. The sensors (S1 S2 S3 S4) data is collected by the control unit and the side with the maximum traffic is identified along with the distance of traffic on each side. The raspberry pi controller then calculates the optimum time for the release of green signal to the side that has been identified with the maximum distance of traffic. At the end of the time provided by the microcontroller, the sensor data from all the other sensors are collected and the side with the highest traffic distance has been identified and the green signal is given to that side. In the next case the last two sensor data is ignored and the other sensors readings are identified and green light signal preference is given to the side which has the highest traffic distance compared to the other sensor values. This process is continued till all the sides leading to an intersection has been given at least one green signal in the first loop[10].

At the beginning of the second loop, all the sensor data are again taken and the side with the highest traffic is identified and the microcontroller determines the optimum time for release of green signal in that particular direction. This process continues till the system is turned off manually.

If there is a scenario in which the radio wave detectors detect the presence of an incoming radio signal which signifies that an emergency vehicle is approaching. Thus, a trigger is fired and the system is alerted about the approaching emergency vehicle. This radio wave signal is again monitored by the control unit and as the radio wave emitter reaches the 500 meters from the junction then the microcontroller fires the final trigger and all the running process is cancelled and the preference is transferred in the direction of emergency vehicle is approaching[4].

The microcontroller immediately changes the other direction signals to red and provides green signal to the direction in which the emergency vehicle is approaching. If there are emergency vehicles approaching in more than one direction then the order of preference is as follows:

Ambulance> Fire Truck > Police trucks

In the condition in which there are two emergency vehicles approaching in one direction and single emergency vehicle in another direction, the preference is given to the direction in which there are 2 or more than 2 emergency vehicles are approaching[7].

IV. PROGRAMMING

Python Programming :

Python is an exceptionally significant level of programming and is an open-source programming language. It was made by a Dutch developer Guido van Rossum. It was first discharged route in 1991. Python's plan study accentuated on code meaningfulness and understandability with its calculable usage of huge whitespace. Its language utilizes object-situated methodology target to assist software engineers with composing clear and sensible codes for little, medium and enormous scale ventures. Python is otherwise called a multi-worldview programming language. This implies Python underpins both item and angle arranged programming. Item situated programming and organized writing computer programs are completely upheld, and a significant number of its highlights bolster utilitarian programming and viewpoint arranged programming. Numerous different ideal models are helped by means of augmentations, including configuration by contract and sensible programming. Python utilizes lively composing and a whole mix of reference checking and a cycle-identifying junk jockey for dealing with the memory size. It likewise comprises of dynamic name goals (late official), which joins names of technique and variable during the execution of any program. Python writing computer programs is limitlessly expanding in applications as it is route more obvious and code when contrasted with other programming dialects. Preferences of Python over other programming languages: Presence of Third-Party Modules: The Python Package Index contains a few outsider modules that help in better collaboration when contrasted with the vast majority of different dialects and stages. Extensive Support Libraries: Python gives a tremendous default library which comprises territories like web conventions, string tasks, web administrations apparatuses and working framework (OS) interfaces. Open Source and Community Development: Python language is an open source programming language which implies that it is allowed to utilize and convey, including for business and business purposes.

User-accommodating Data Structures:

This programming language has a worked in rundown and word reference information structures which can be utilized to manufacture quick runtime information structures. Further it likewise gives the alternative of extremely spirited and a significant level information composing which confines the length of help code that is fundamental.

Snippet of program

```
import RPi.GPIO as GPIO
from time import sleep
hallpin1=8
#LED1=8
hallpin2=10
hallpin3=12
#hallpin4=24
hallpin11=22
hallpin12=24
hallpin13=26
hallpin21=38
hallpin22=40
hallpin23=37
hallpin31=31
hallpin32=29
hallpin33=23
LED1=16
LED2=18
LED11=32
LED12=36
LED21=35
LED22=33
LED31=21
LED32=19
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)
GPIO.setup(LED1, GPIO.OUT, initial=GPIO.LOW)
GPIO.setup(LED2, GPIO.OUT, initial=GPIO.LOW)
GPIO.setup(hallpin1, GPIO.IN)
#GPIO.setup(LED2, GPIO.OUT, initial=GPIO.LOW)
GPIO.setup(hallpin2, GPIO.IN)
GPIO.setup(hallpin3, GPIO.IN)
GPIO.setup(LED11, GPIO.OUT, initial=GPIO.LOW)
GPIO.setup(LED12, GPIO.OUT, initial=GPIO.LOW)
GPIO.setup(hallpin11, GPIO.IN)
GPIO.setup(hallpin12, GPIO.IN)
GPIO.setup(hallpin13, GPIO.IN)
GPIO.setup(LED21, GPIO.OUT, initial=GPIO.LOW)
GPIO.setup(LED22, GPIO.OUT, initial=GPIO.LOW)
GPIO.setup(hallpin21, GPIO.IN)
GPIO.setup(hallpin22, GPIO.IN)
GPIO.setup(hallpin23, GPIO.IN)
GPIO.setup(LED31, GPIO.OUT, initial=GPIO.LOW)
GPIO.setup(LED32, GPIO.OUT, initial=GPIO.LOW)
GPIO.setup(hallpin31, GPIO.IN)
GPIO.setup(hallpin32, GPIO.IN)
GPIO.setup(hallpin33, GPIO.IN)
while True:
  print("----------------------------")
  if(GPIO.input(hallpin1)==True):
   # GPIO.output(LED1, GPIO.HIGH)
    a1=1
    print("magnet 1")
    print("detected")
  if(GPIO.input(hallpin1)==False):
    a1=0
    print("magnet 1")
    print("not detected")
  if(GPIO.input(hallpin2)==True):
    a2=1
    print(" magnet 2")
    print(" detected")
  if(GPIO.input(hallpin2)==False):
    a2=0
    print(" magnet 2")
    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")
    print("  detected")

  if(GPIO.input(hallpin12)==False):
    b2=0
    print("magnet 12")
    print(" not detected")
  if(GPIO.input(hallpin13)==True):
    b3=1
    print(" magnet 13")
    print(" detected")
  if(GPIO.input(hallpin13)==False):
    b3=0
    print(" magnet 13")
    print(" not detected")
  print("-----------------------------")
```

```
    if(GPIO.input(hallpin21)==True):
      c1=1
      print(" magnet 21")
      print(" detected")
    if(GPIO.input(hallpin21)==False):
      c1=0
      print("magnet 21")
      print("not detected")
    if(GPIO.input(hallpin22)==True):
      c2=1
      print("magnet 22")
      print("detected")
    if(GPIO.input(hallpin22)==False):
      c2=0
      print(" magnet 22")
      print("not detected")
    if(GPIO.input(hallpin23)==True):
      c3=1
      print("magnet 23")
      print(" detected")
    if(GPIO.input(hallpin23)==False):
      c3=0
      print("magnet 23")
      print("not detected")
  print("------------------------------")
    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("
  magnet 32")
      print("
  not detected")
    if(GPIO.input(hallpin33)==True):
      d3=1
      print("
  magnet 33")
      print("
  detected")
    if(GPIO.input(hallpin33)==False):
      d3=0
      print("
  magnet 33")
      print("
  not detected")
    sum1=a1+a2+a3
    sum2=b1+b2+b3
    sum3=c1+c2+c3
    sum4=d1+d2+d3
    print(sum1)
    print(sum2)
    print(sum3)
    print(sum4)
    f1=0
    f2=0
    f3=0
    f4=0
    if(f1==1)and(f2==1)and(f3==1)and(f4==1):
      f1=0
      f2=0
      f3=0
      f4=0
  if(f1==0):

  if(sum1>sum2)and(sum1>sum3)and(sum1>sum4):
          GPIO.output(LED1, GPIO.HIGH)
          GPIO.output(LED12, GPIO.HIGH)
          GPIO.output(LED22, GPIO.HIGH)
          GPIO.output(LED32, GPIO.HIGH)
          sleep(15)
          GPIO.output(LED1, GPIO.LOW)
          GPIO.output(LED12, GPIO.LOW)
          GPIO.output(LED22, GPIO.LOW)
          GPIO.output(LED32, GPIO.LOW)
          f1=1
    if(f2==0):
  if(sum2>sum1)and(sum2>sum3)and(sum2>sum4):
          GPIO.output(LED11, GPIO.HIGH)
          GPIO.output(LED2, GPIO.HIGH)
          GPIO.output(LED22, GPIO.HIGH)
          GPIO.output(LED32, GPIO.HIGH)
          sleep(15)
          GPIO.output(LED11, GPIO.LOW)
          GPIO.output(LED2, GPIO.LOW)
          GPIO.output(LED22, GPIO.LOW)
          GPIO.output(LED32, GPIO.LOW)
          f2=1
    if(f3==0):
  if(sum3>sum1)and(sum3>sum2)and(sum3>sum4):
          GPIO.output(LED21, GPIO.HIGH)
          GPIO.output(LED2, GPIO.HIGH)
          GPIO.output(LED12, GPIO.HIGH)
          GPIO.output(LED32, GPIO.HIGH)
          sleep(15)
          GPIO.output(LED21, GPIO.LOW)
          GPIO.output(LED2, GPIO.LOW)
          GPIO.output(LED12, GPIO.LOW)
          GPIO.output(LED32, GPIO.LOW)
          f3=1
  if(f4==0):
  if(sum4>sum1)and(sum4>sum2)and(sum4>sum3):
          GPIO.output(LED31, GPIO.HIGH)
          GPIO.output(LED2, GPIO.HIGH)
```

```
        GPIO.output(LED12, GPIO.HIGH)
        GPIO.output(LED22, GPIO.HIGH)
        sleep(15)
        GPIO.output(LED31, GPIO.LOW)
        GPIO.output(LED2, GPIO.LOW)
        GPIO.output(LED12, GPIO.LOW)
        GPIO.output(LED22, GPIO.LOW)
        f4=1
    sleep(2)
```

Fig working prototype

V. CONCLUSION

smart traffic management system has given the best results to with waiting & travelling time of a passenger has been reduced and emergency vehicles can move without obstacles or barriers. The pollution rate can be reduced by implementing this smart traffic management system in all prime locations. The suggested traffic management system can be implemented in all metropolitan cities as it is most suitable and reliable for the day.