**Advance Traffic Control**

**System With Real Adaptive**

**Time Allotment**

**Introduction To IOT**

**Course Project Report**

***Submitted By***

**Manu Sai 221000032**

**Priyanshu Pradhan 221000041**

**Pushpraj Singh Paikra 221000042**



**Dr. Shyama Prasad Mukherjee International Institute Of Information Technology, Naya Raipur**

**June-2023**

## Abstract

By increasing of population, the usage of vehicles has been increasing and controlling of traffic is one of the challenging works. The frequent traffic jams at major junctions call for an efficient traffic management system in place. The resulting wastage of time and increase in pollution levels can be eliminated on a city-wide scale by these systems. Previously the traffic control techniques used like magnetic loop detectors, induction loop detectors are buried on the road side provide the limited traffic information and require separate systems for traffic counting and for traffic surveillance. Here the project proposes to implement an artificial density traffic control system using Raspberrypi. The hardware here we are using is IR sensor, Raspberry pi, Jumper wire, Red and Green led and the software used is MATLAB.

In this project the camera is get interfaced with a Raspberry pi. The image sequences from a camera are analysed using thresholding method to find the density of vehicles. Subsequently, the number of vehicles at the intersection is evaluated and traffic is efficiently managed. In this project we implemented a real-time emergency vehicle detection system. In case an emergency vehicle is detected, the lane is given priority over all the others.

1. **Introduction**

Current traffic control techniques involving magnetic loop detectors buried in the road, infra-red and radar sensors on the side provide limited traffic and require separate systems for traffic counting and for traffic surveillance. Inductive loop detectors do provide a cost-effective solution; however, they are subject to a high failure rate when installed in poor road surfaces, decrease pavement life and obstruct traffic during maintenance and repair

Infrared sensors are affected to a greater degree by fog than video cameras and cannot be used for effective surveillance. In contrast, our project is designed to solve the major and still unsolved problem in Indian transportation system. Most effective way to reduce traffic in current use is by using traffic light. which is quite ineffective It has major drawback; the major drawback is that its non-adaptive and continuously allot same time to each regardless of the density of vehicle. E.g., sometimes allocating same amount of time to the lane with no vehicle and sometimes giving not enough time to lane with more vehicle**.**

Traffic congestion reduction: One of the primary motivations is to address the issue of traffic congestion. Advanced traffic control systems can help optimize traffic flow, minimize bottlenecks, and reduce delays. By improving the efficiency of traffic management, the system can contribute to reducing congestion on road networks, leading to smoother traffic flow and shorter travel times.

Enhanced efficiency and productivity: Advanced traffic control systems can improve the efficiency and productivity of transportation networks. By employing technologies such as real-time data collection, predictive modelling, and adaptive control algorithms, the system can optimize traffic signal timings, adjust lane configurations, and allocate resources effectively. This leads to improved utilization of existing infrastructure, reduced travel times, and increased capacity of road networks.

Urban areas witness a mix of vehicles, including different sizes, speeds, and maneuvering capabilities. The existing traffic control systems may not adequately accommodate this variety, leading to suboptimal traffic flow, increased congestion, and safety concerns. The research aims to develop a prototype system that can intelligently handle the heterogeneity of traffic, optimizing signal timings, and traffic management strategies accordingly. Real-time Data Integration: The effectiveness of an advanced traffic control system relies on accurate and timely data acquisition. However, integrating real-time data from various sources, such as traffic sensors, surveillance cameras, and connected vehicles, presents a significant challenge. The research problem involves designing mechanisms to efficiently collect, process, and analyze heterogeneous data streams, ensuring their seamless integration into the traffic control system for enhanced decision-making.

Scalability and Adaptability: Developing a prototype system that is scalable and adaptable to changing urban environments and emerging technologies is a significant contribution. The system can be designed to accommodate future growth in traffic volume and evolving transportation trends, such as connected and autonomous vehicles. Its scalability ensures long-term effectiveness, while its adaptability allows for the seamless integration of new technologies and data sources, making it a valuable asset for urban planning and transportation management. Improved Traffic Efficiency: By incorporating advanced algorithms, real-time data analysis, and adaptive control mechanisms, the prototype system can significantly enhance traffic efficiency. It can optimize traffic signal timings, adapt to changing traffic patterns, and dynamically allocate resources to reduce congestion, minimize delays, and improve overall traffic flow. This contribution leads to reduced travel times, increased capacity utilization, and enhanced productivity of transportation networks.

1. **Objective**

* The objective of our project is to design adaptive control system which can adapt accordingly in real time and allot time accordingly.
* It will be a modified traffic light system with ability to sense the density of vehicle in each lane allot time optimally.

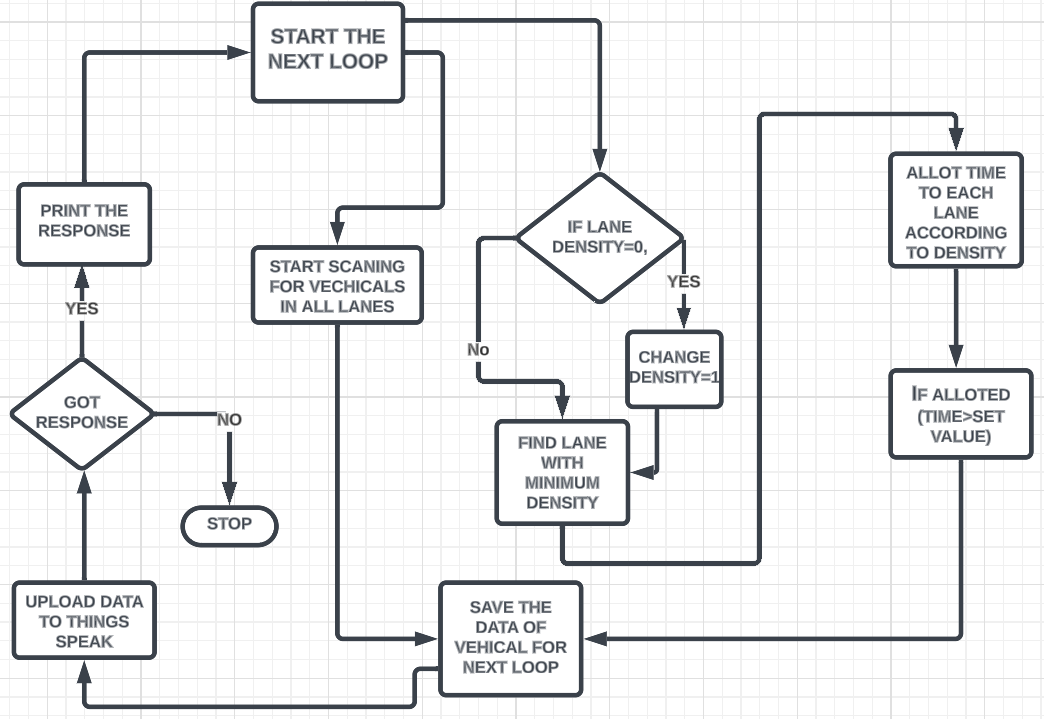


**Fig – 1 (Indian – traffic system)**

1. **Methodology**

We will use following algorithm to find number of vehicle and allocation of time for each and every lane**.**

* Problem Definition and Requirements Analysis: The first step is to clearly define the problem statement and conduct a comprehensive analysis of the requirements. This involves understanding the existing traffic control challenges, identifying specific goals and objectives, and defining the desired functionalities and performance metrics for the prototype system. Stakeholder consultation and literature review play a crucial role in this phase.
* Data Collection and Preprocessing: Gathering relevant data is essential for developing an effective prototype system. Various data sources, including traffic sensors, surveillance cameras, and connected vehicle data, are collected. Data preprocessing techniques are then applied to clean, validate, and integrate the collected data, ensuring its quality and compatibility with the subsequent analysis and modeling stages.
* Traffic Flow Modelling and Simulation: Traffic flow models are developed to represent the behaviour and dynamics of the transportation network. Different modelling techniques, such as macroscopic, mesoscopic, or microscopic models, can be employed based on the specific requirements and objectives. Simulation tools, such as VISSIM, SUMO, or TRANSIMS, are used to evaluate the performance of the traffic control system under various scenarios, helping to identify optimal control strategies.
* Algorithm Design and Optimization: Advanced control algorithms and optimization techniques are developed to improve traffic control strategies. These algorithms can include adaptive signal control, predictive modelling, machine learning, and optimization algorithms. The design of these algorithms focuses on optimizing traffic signal timings, coordinating signal operations, and efficiently allocating resources to achieve the desired traffic flow objectives.
* Prototype System Implementation: The developed algorithms and control strategies are implemented in a software-based prototype system. This involves coding the algorithms, integrating them with the data acquisition and processing modules, and developing a user interface for system control and monitoring. The prototype system should be designed to handle real-time data inputs, perform calculations, and generate control outputs effectively.
* System Integration and Testing: The prototype system is integrated with various components, such as traffic signal controllers, communication networks, and data interfaces. Integration testing ensures the proper functioning of all system components and their interaction. The prototype system is then subjected to extensive testing under different traffic scenarios and conditions to evaluate its performance, validate its effectiveness, and identify any potential issues or limitations.

****

**Fig-2 (Block – diagram)**

**Code implementing following algorithm**

import rpi.gpio as gpio

import time

import threading

import requests

# your thing speak api key and url

sensor1=15

sensor2=16

sensor3=22

led1=13

led2=11

led3=18

gpio.setmode(gpio.board)

gpio.setup(sensor1,gpio.in)

gpio.setup(sensor2,gpio.in)

gpio.setup(sensor3,gpio.in)

gpio.setup(led1,gpio.out)

gpio.setup(led2,gpio.out)

gpio.setup(led3,gpio.out)

lane1=0

lane2=0

lane3=0

def set(a):

if(a>3):

a=3

def sensor1call():

global lane1

global objcnt1

objcnt1=0

prev=0

while(true):

curr=gpio.input(sensor1)

if curr!=prev:

if curr==1:

objcnt1+=1

prev=curr

lane1=objcnt1

time.sleep(0.01)

def sensor2call():

global lane2

global objcnt2

objcnt2=0

prev=0

while(true):

curr=gpio.input(sensor2)

if curr!=prev:

if curr==1:

objcnt2+=1

prev=curr

lane2=objcnt2

time.sleep(0.01)

def sensor3call():

global lane3

global objcnt3

objcnt3=0

prev=0

while(true):

curr=gpio.input(sensor3)

if curr!=prev:

if curr==1:

objcnt3+=1

prev=curr

lane3=objcnt3

time.sleep(0.01)

t1=threading.thread(target=sensor1call)

t2=threading.thread(target=sensor2call)

t3=threading.thread(target=sensor3call)

density1=1

density2=1

density3=1

t1.start()

t2.start()

t3.start()

try:

while true:

# your thingspeak api key and url

write\_api\_key = 'bgu9uvnufhojr7b2'

write\_api\_url = f'https://api.thingspeak.com/update?api\_key={write\_api\_key}'

# the data you want to send to thingspeak

# send the data to thingspeak

field1 = density1

field2 = density2

field3 = density3

data = {'field1': field1, 'field2': field2, 'field3': field3}

response = requests.post(write\_api\_url, data=data)

# check if the data was successfully sent

if response.ok:

print(f'data sent to thingspeak: field1={field1}, field2={field2}, field3={field3}')

else:

print(f'failed to send data to thingspeak: {response.text}')

#setting value for density fo each lane

objcnt1=0

objcnt2=0

objcnt3=0

print(lane1,lane2,lane3)

minval=min(min(density1,density2),density3)

if minval==0:

minval=1

if density1==0:

density1=1

if density2==0:

density2=1

if density3==0:

density3=1

density3=density3/minval

density2=density2/minval

density1=density1/minval

set(density1)

set(density2)

set(density3)

timeal=2

#coverting density into time for each lane

time1=(density1//1)\*timeal

time2=(density2//1)\*timeal

time3=(density3//1)\*timeal

#turning light on/off on the basis of data given

#light for lane 1

gpio.output(led1,true)

time.sleep(time1)

gpio.output(led1,false)

#light for lane 2

gpio.output(led2,true)

time.sleep(time2)

gpio.output(led2,false)

#light for lane 3

gpio.output(led3,true)

time.sleep(time3)

gpio.output(led3,false)

#time to process data taken by ir sensor till this point

print("done1")

#processing for lane 1

#processing for lane 2

#processing for lane 3

#setting density after processing data

density1=lane1

density2=lane2

density3=lane3

except keyboardinterrupt:

gpio.cleanup()

1. **IoT Architecture**

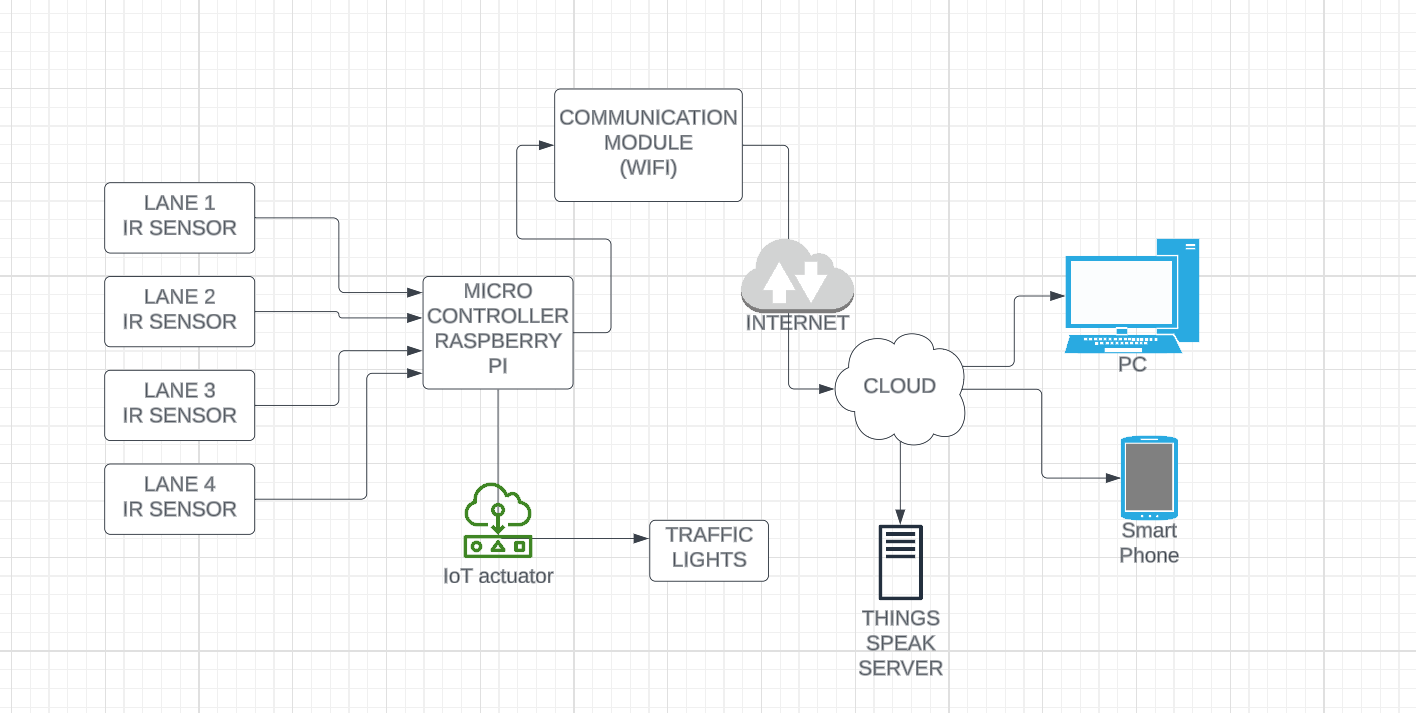
****

Fig - 3 (IoT – Architecture)

An infrared (IR) sensor is a device that detects and measures infrared radiation emitted by objects or individuals. These sensors work by detecting changes in the intensity of infrared radiation, allowing them to trigger appropriate responses or actions based on the detected signals. In our project we have connected 3 IR sensors to detect the object to the raspberry.

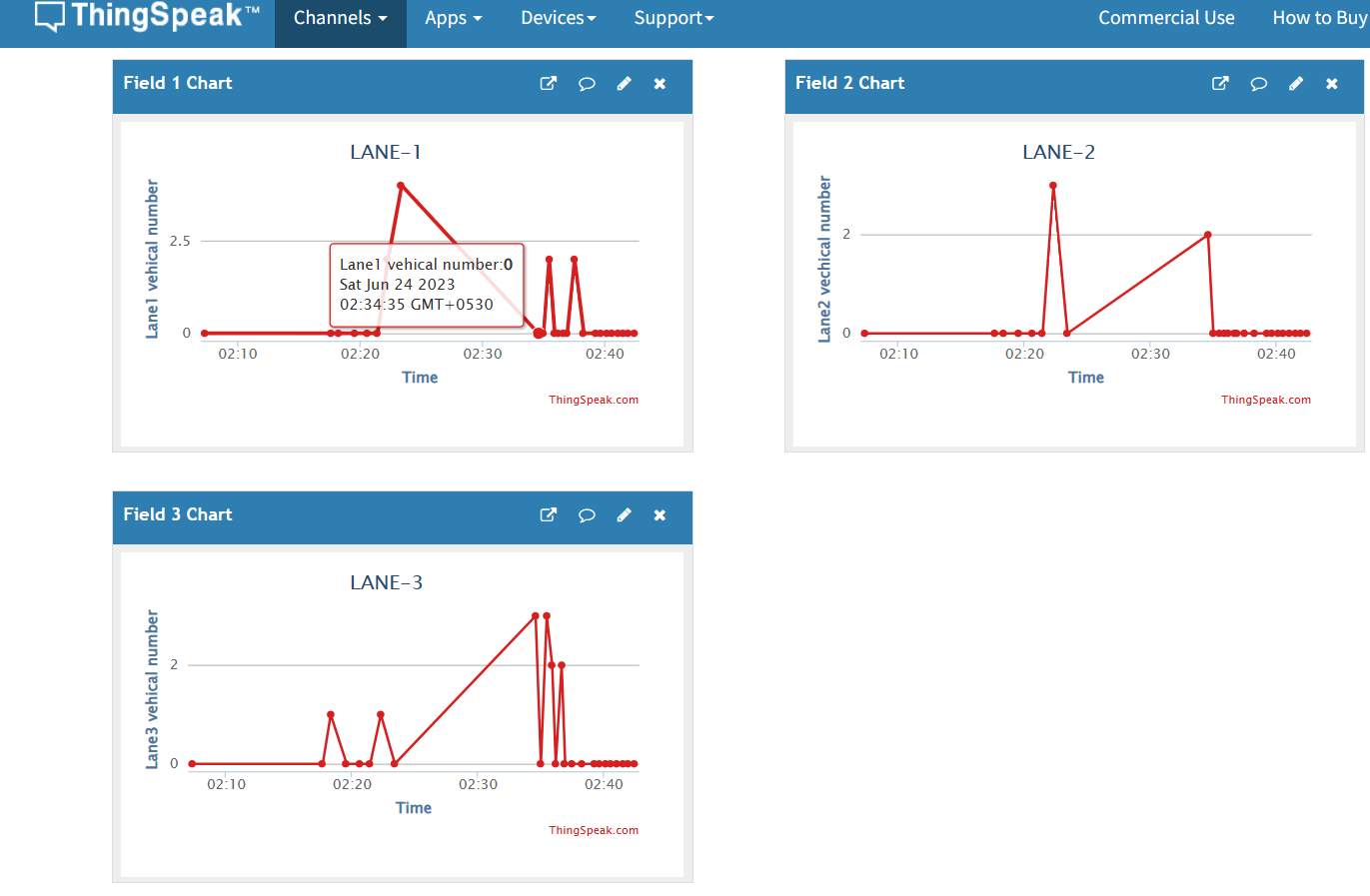
Raspberry Pi boards have powerful processing capabilities, support various operating systems, and offer a wide range of input/output options, making them suitable for a diverse range of applications such as home automation, robotics, media centres, and more.

Cloud IoT Platforms: Raspberry Pi can connect to cloud IoT platforms like AWS IoT, Google Cloud IoT Core, or Azure IoT Hub. These platforms provide SDKs and libraries that allow developers to easily integrate Raspberry Pi with cloud services, enabling seamless data streaming and device management. We have used think speak to upload the number of vehicles

1. **Results**

It will solve these major problems in our transportation system

* We will get that each of the lane is allocated time directly proportional to number of vehicles in it. such that the model will result in ease of travelling in every junction increasing the efficiency and decreasing the requirement of making extra roads in the city.
* It will be an adaptive model therefore it can also adapt in situation like holidays, festivals etc.
* Such that programmer or larger authorities can use that data in the future to plan city or generate an app to show the route with least traffic.
* Since, its adaptive in future it can easily modified accordingly to traffic scenarios. multiple devices such as fire sensor, motion sensor and microphones can be also adder for further development of this model.

Fig – 4 (Think Speak)

* In case of curfew or via incoming the system can be edited to perform required task since the traffic light are fully controlled by the microcontroller connected to the internet. We will get data in thing speak about the number of vehicles in each lane.



Fig – 5 (Prototype)

* Google can use this to data to increase the precision of its traffic predicting model in google maps. Local traffic departments can predict the traffic situation in real time and handle it accordingly.

1. **Comparison of our solution with existing one**

* Our model is fully modifiable therefore it can be modified accordingly to the situation where it is used and can be accessed through the internet.
* It on real time basis uploads traffic data to the server. hence, we can integrate multiple models in every junction to design the real time traffic model of the city and do data analysis in it.
* The sensor used in out model is quite robust, cheap and accurate. hence, it's easily reproducible and easy to implement.

CURRENT SOLUTION OUR SOLUTION

|  |  |
| --- | --- |
| * Use camera which is quite inaccurate. | * Our model uses IR sensor which gives accurate data. |
| * Camera is quite fragile and can’t withstand temperature changes. | * IR sensor is durable and can be easily replaceable. |
| * Setting up this is quite expensive and it will cost too much to implement on large scale. | * Very cheap to setup and parts are easily available. Hence, it can be implanted on larger scale. |
| * It doesn’t upload real time data of traffic to the server. | * It uploads real time data of traffic to the server. |

1. **Learning outcomes**

The learning Outcome of this project are as follows:

* Technical Expertise: Developing a prototype advanced traffic control system requires a deep understanding of traffic engineering principles, data analysis, optimization techniques, and algorithm development. The project provides an opportunity to enhance technical expertise in these areas and gain hands-on experience in implementing advanced control strategies.
* Data Management and Integration: The prototype system involves collecting, preprocessing, and integrating various data sources, such as traffic sensors, cameras, and connected vehicles. This process allows for a better understanding of data management challenges, data quality assurance, and techniques for integrating heterogeneous data streams. It enhances knowledge in data integration and analysis, which can be applied to future transportation-related projects.
* Algorithm Development and Optimization: The development of advanced control algorithms and optimization techniques is a key aspect of the project. It involves designing algorithms that adapt to real-time traffic conditions, optimize signal timings, and allocate resources effectively. Through this process, participants gain expertise in algorithm development, optimization methodologies, and simulation-based evaluation.
* Systems Thinking and Integration: Developing a prototype traffic control system requires considering the integration of various components, including data acquisition, processing, algorithm implementation, and user interfaces. This fosters a system thinking approach, where participants learn to identify interdependencies, manage system integration challenges, and ensure seamless communication between different system elements.
* Practical Implementation Considerations: Developing a prototype system provides an opportunity to understand the practical challenges associated with deploying advanced traffic control systems. Participants gain insights into real-world implementation considerations, such as infrastructure requirements, system scalability, maintenance needs, and integration with existing traffic management frameworks. This knowledge helps in developing feasible and practical solutions that can be implemented in the field.
* Strategic Planning: Developing a prototype advanced traffic control system contributes to strategic planning and policy formulation. Participants gain an understanding of how the system aligns with broader transportation goals, sustainability objectives, and urban planning initiatives. They learn to consider the strategic implications of implementing advanced traffic control systems and their potential impact on transportation management strategies.

1. **References**

* Raspberry pi documentation <https://www.raspberrypi.com/documentation/>
* YouTube (multithreading in python | python tutorial)

[https://www.youtube.com/watch?v=icbu6zaktqq](https://www.youtube.com/watch?v=ICbU6zAKtqQ)

* Chat gpt

<https://openai.com/blog/chatgpt>

* Python documentation

<https://docs.python.org/3/>

* Think speak documentation

<https://www.mathworks.com/help/thingspeak/>