VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANA SANGAMA, BELAGAVI – 590 018, KARNATAKA

A Project Report
on

REAL-TIME TRAFFIC MANAGEMENT SYSTEM USING IoT

Submitted in partial fulfillment of the requirements for the VIII Semester of degree of ***Bachelor of Engineering in Information Science and Engineering*** *of VisvesvarayaTechnological University, Belagavi*

by

Samarth R Aithal	**1RN18IS091**	**Ronak H Rathod**	**1RN19IS406**
Varun D S	**1RN18IS120**	**Ullas S Rao**	**1RN19IS410**

Under the Guidance of
Mr. Santhosh kumar
Assistant Professor
Department of ISE

Department of Information Science and Engineering

RNS INSTITUTE OF TECHNOLOGY

Dr. Vishnuvaradhan Road, Rajarajeshwari Nagar Post,
Channasandra, Bengaluru – 560 098

2021-2022

RNS INSTITUTE OF TECHNOLOGY

Dr. Vishnuvaradhan Road, Rajarajeshwari Nagar Post
Channasandra, Bengaluru – 560 098

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the project work entitled ***Real-time Traffic Management System using IoT*** has been successfully completed by **Samarth R Aithal (1RN18IS091), Varun D S (1RN18IS120), Ronak H Rathod (1RN19IS406), and Ullas S Rao (1RN19IS410),** bonafide students of **RNS Institute of Technology, Bengaluru** in partial fulfillment of the requirements for the award of degree **Bachelor of Engineering** in **Information Science and Engineering** of **Visvesvaraya Technological University, Belagavi** during academic year **2021-2022**. The project report has been approved as it satisfies the academic requirements in respect of project work for the said degree.

Mr. Santhosh kumar Project Guide	**Dr. Prakasha S / Mrs. Kusuma S** Project Coordinator
Dr. Suresh L Professor and HOD	**Dr. M K Venkatesha** Principal

External Viva

Name of the Examiners	**Signature with Date**
1. ____________________________	**1.** ____________________
2. ____________________________	**2.** ____________________

DECLARATION

We, **SAMARTH R AITHAL [USN: 1RN18IS91]**, **VARUN D S [USN: 1RN18IS120], RONAK H RATHOD [USN: 1RN19IS406], ULLAS S RAO [USN: 1RN19IS406]** students of VIII Semester B.E. in Information Science and Engineering, RNS Institute of Technology hereby declare that the Project entitled ***Real-Time Traffic Management System using IoT*** has been carried out by us and submitted in partial fulfillment of the requirements for the *VIII Semester of degree of* ***Bachelor of Engineering in Information Science and Engineering*** *of Visvesvaraya Technological University*, *Belagavi* during academic year 2021-2022.

Place: Bengaluru

Date:

SAMARTH R AITHAL	**(1RN18IS091)**
VARUN D S	**(1RN18IS120)**
RONAK H RATHOD	**(1RN19IS406)**
ULLAS S RAO	**(1RN19IS410)**

ABSTRACT

A significant amount of research work carried out on traffic management systems, but intelligent traffic monitoring is still an active research topic due to the emerging technologies such as the Internet of Things (IoT) and Artificial Intelligence (AI). The integration of these technologies will facilitate the techniques for better decision making and achieve urban growth. However, the existing traffic prediction methods mostly dedicated to highway and urban traffic management, and limited studies focused on collector roads and closed campuses. Besides, reaching out to the public, and establishing active connections to assist them in decision-making is challenging when the users are not equipped with any smart devices.

This research proposes an IoT based system model to collect, process, and store real-time traffic data for such a scenario. The objective is to provide real-time traffic updates on traffic congestion and unusual traffic incidents through roadside message units and thereby improve mobility. These early-warning messages will help citizens to save their time, especially during peak hours. Also, the system broadcasts the traffic updates from the administrative authorities. A prototype is implemented to evaluate the feasibility of the model, and the results of the experiments show good accuracy in vehicle detection and a low relative error in road occupancy estimation.

ACKNOWLEDGMENT

The fulfillment and rapture that go with the fruitful finishing of any assignment would be inadequate without the specifying the people who made it conceivable, whose steady direction and support delegated the endeavors with success.

We would like to profoundly thank **Management** of **RNS Institute of Technology** for providing such a healthy and vibrant environment to carry out the project work.

We would like to express our sincere thanks to our beloved Principal **Dr. M K Venkatesha** for his support and inspired me towards the attainment of knowledge.

We wish to place on record our words of gratitude to **Dr. Suresh L,** Professor and Head of the Department, Information Science and Engineering, for being the enzyme and master mind behind our project work.

We would also like to thank our project guide **Mr. Santhosh kumar,** Assistent Professor, Department of ISE, RNSIT, Bengaluru, for his valuable inputs, suggestions, time and guidance.

We place our thanks to project coordinators **Dr. Prakasha S,** Associate Professor and **Ms. Kusuma S,** Assistant Professor, ISE, RNSIT for their timely guidelines and suggestions for carrying out the project work successfully.

We would like to thank all other teaching and non-teaching staff of Information Science & Engineering who have directly or indirectly helped me to carry out the project work.

Place: Bengaluru

Date:

SAMARTH R AITHAL
VARUN D S
RONAK H RATHOD
ULLAS S RAO

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

ESP	Event Stream Processing
IDE	Integrated Development Environment
IoT	Internet of Things
IR	Infrared Sensor
LCD	Liquid Crystal Display
MCU	Microcontroller Unit
PC	Personal Computer
WSN	Wireless Sensor Network

Chapter 1

INTRODUCTION

The sustainability and smartness of the smart city concept rely on the technologies adopted to improve the people's quality of life. The smart city governance is one significant aspect of smart city initiatives, which will facilitate the planning techniques for better decision making. One of the key elements of the smart city governance framework is the public value generated out of the smart services provided.

The emergence of the internet of things (IoT) has evolved the concept of smart cities. In a smart city environment, the physical infrastructures of the city are equipped with smart devices, which continuously produce multidimensional data in different spaces and these data are processed to achieve intelligence for the infrastructure. Ultimately, intelligence is applied to improve the socio-economic activities of the society.

Smart traffic infrastructure is an essential component of smart city initiatives because traffic congestion is a severe issue that grows along with city development. Smart traffic management includes intelligent transport systems with integrated components like adaptive traffic signal controls, freeway management, emergency management services, and roadside units. Such systems collect real-time traffic data and take necessary measures to avoid or minimize any social issue created as part road congestions.

The widely used mobile applications like Google Maps or Apple Maps accurately predict traffic congestion for urban roads based on the sensor data from monitoring devices installed on highways or urban roads. These application providers establish partnerships with various transportation entities to gather traffic information. The transportation governing authorities mostly install the traffic monitoring devices on urban roads, hence such application providers (e.g., Google application programming interface) deliver updates on urban traffic congestion. Besides, such applications also use crowdsourcing with location-based services to improve traffic density prediction. They do expect smart technologies within the vehicle or any smart mobile device with the driver of the vehicle to receive real- time traffic updates. The traffic pattern of urban roads or highways is different from that of collector roads. The users of collector roads include pedestrians, bicycles, motorbikes, and other vehicles; hence, the traffic pattern is different from the highways.

Along with urban roads, the real-time monitoring of collector roads is also, essential to improve the mobility of the entire city. The real-time traffic congestion updates, warnings from traffic authorities on non-recurrent traffic incidents such as accidents spilled loads, VIP visits, ambulance services, or any other unusual road incidents will support the collector road drivers in their decision-making. The real-time traffic updates of roads that connect to the exit points may help the driver for selecting the most suitable route from his current position. The drivers prefer to know about the congestion state of forthcoming intersections to plan themselves and save their time on the road by choosing alternate ways. The question that arises here is how to provide real-time road congestion updates to drivers even if there are no such smart devices with them or within the car, which is the real motivation of this research.

The recent research efforts in intelligent transport systems show that the IoT paradigm can play an important role in traffic management by connecting the physical devices over the internet to exchange information, tracking, and monitoring traffic movement. The global positing systems, sensors, probe vehicles, and vehicle to infrastructure communication are a few ways to collect real-time road data. The sensors such as acoustic and magnetic sensors are cost-effective, power-efficient, and most popular among the recent vehicle monitoring solutions. The collected traffic information from multiple sources can be used to predict and manage traffic congestions. Most of the existing solutions deliver real-time traffic updates of urban roads especially through smart mobile devices and limited attention has been given to collector roads. Therefore, this research proposes an IoT-based system model to provide real- time traffic updates through roadside message units. The proposed system is not limited on its application to collector roads.

Chapter 2

LITERATURE SURVEY

A literature review is a type of review article. A literature review is a scholarly paper, which includes the current knowledge including substantive findings, as well as theoretical and methodological contributions to a particular topic. Literature reviews are secondary sources and so not report new or original experimental work. Most often associated with academic oriented literature, such reviews are found in academic journals and are not to be confused with book reviews that may also appear in the same publications. Literature reviews are a basis for research in nearly every academic field. A narrow scope literature review may be included as a part if a peer reviewed journal article presenting new research, serving to simulate the current study within the body of the relevant literature and to provide context for the reader.

An extensive review if the research undertaken in the domain related to face recognition along with the gender classification and facial age estimation from the face image. Every technical and IEEE papers have their own features which elaborate about particular technique with a certain methodology. It contains both advantages and drawbacks.

In paper [1], they have discussed about an intelligent and real-time traffic information system that was developed based on the video detection and an image processing algorithm to measure traffic- flow according to the average speed of vehicles. Then, traffic status of each pass way is broadcasted to the electronic boards installed on all decision-making entrance / exit. Different levels of congestion related to the routes ahead are shown on the boards with different colors in order to assist commuters. Experimental results are promising due to the proximity of determined traffic status by the system compared to the detection done by traffic experts. Average speed improvement is another result of using this system. This intelligent system developed and implemented in Shiraz city for the first time.

In Paper [2], Increased vehicle density on roads has resulted in plethora of challenges for traffic management in urban cities worldwide. Failure in transporting accident victims, critical patients and medical equipment on time has led to loss of human lives which are the results of road congestions. This paper introduces a framework with the fused concept of vehicular ad hoc networks and Internet of things that aims to prioritize emergency vehicles for smooth passage through the road traffic. The proposed system navigates ambulances in finding the nearest possible path to their destination based on real-time traffic information. A simulation of the designed framework has been demonstrated using Cup Carbon simulator.

In Paper [3], The paper explores the utilization of RFID innovation for traffic congestion and find out the blockage at any intersection of the street by utilizing RFID reader and labels as sensors. The idea behind this paper is to make the fixed and preset activity of traffic signal dynamic. The paper affords a unique method for making the signal timing proportional to the congestion on the roads at any time directly. Proposed intelligent system can maintain the dynamic timings of traffic signals by sensing the density of traffic to minimize the congestion with the help of IoT enabled sensors which provides the advanced and powerful communication technologies for the citizens.

In Paper [4], This paper proposes to implement a smart traffic control system which is based on the measurement of traffic density using real time video processing technique. The video sequences from a camera are analyzed using object detection and counting methods to obtain the most effective way. The computed vehicle density is compared with other parts of the traffic in order to control the traffic signal brilliantly. The system has an advantage of using RFID sensors to ensure law enforcement. Therefore, any car or vehicle which breaks the traffic rules can be easily caught.

In Paper [5], This paper describes a system where ultrasonic sensors are integrated with the Raspberry Pi to operate the lanes of an intersection based on the density of traffic. The current condition of the intersection is updated on a user accessible website. This integration of traffic systems in an Internet of Things (IoT) fashion enables the addition of smart security and road safety devices. As a result, the improvement in traffic system can be incrementally enhanced, which can lead to eventually significant improvement in the overall traffic system.

In Paper [6], In this paper they have discussed on how sensor technology can be integrated with transportation infrastructure to achieve a sustainable Intelligent Transportation System (ITS), since the modern society faces serious problems with transportation system which consists of traffic congestion, safety and pollution. As information technologies have gained popularity when implemented in the modern transportation systems lot of automotive manufacturers started developing in-vehicle sensors and its applications which include safety, traffic management, even government authorities are implementing infrastructures like cameras and sensors for collecting data about traffic. condition, so by this we can leverage the capabilities of smart traffic and transportation systems. Lastly the challenges that come with implementation of ITS environment.

In Paper [7], This paper reports that Magnetic Sensors can we used to detect different types of vehicles classification such as non-axle based and which cannot be obtained from standard loop data, as those networks can be deployed quickly and can be used for only temporary traffic measurement. The capabilities of magnetic sensors were reported on basis of two field experiments, the first experiment collected 2-H trace of measurements on Hearst Avenue in California which had around 99% better vehicle detection rate, the measurements also yield intervehicle spacing, revealing interesting phenomena such a platoon formation downstream of traffic signal. The second experiment was preliminary, sensor data from 37 passing vehicles at the same site were processed and classifies into six types, it has an accuracy of 60% where vehicle length is not used as a feature but if length used it’s believed to give 80-90% accuracy, these wireless magnetic Sensor networks offer an attractive low-cost alternative to inductive loops for traffic measurements in freeways.

In Paper [8], This paper a new sensing device is used that can simultaneously monitor traffic congestion. This sensing devices is based on the combination of passive infrared sensors (PIRs) and ultrasonic rangefinder. The framework relies of dynamic Bayesian Network to fuse heterogeneous data both spatially and temporally for vehicle detection. In order to estimate the speed of incoming vehicles we check the time delay between signals of different sensors and the propose calibration and self- correction model which is based on Bayesian networks. The measurements of the ultrasonic and PIR sensors are used for vehicle classification as a result of validated data it shows 99% accuracy in vehicle detection and a mean error of 5kph vehicle speed estimation, a mean error of 0.7m in vehicle length estimation. With this computational performance of the algorithm which shoes that it can be implemented on low powered devices within the wireless sensor network setting.

In Paper [9], This paper study is based on detecting traffic accidents, spiled loads that lead to traffic congestion, so detecting and finding the position of the occurred accident accurately, quickly can reduce the congestion on traffic and evacuation of the area effected by the accident by using a video based detecting and positioning method by analyzing distribution characteristics of traffic states in a road segment where each lane of the road is monitored and divided into cluster cells, with this we can find average speed and average space occupancy of the congestion in the traffic. On the basis of the parameters, traffic states in the cells are judged via a fuzzy-identification method. For each congested cell, a feature vector is constructed by taking its state together with states of its upstream and downstream neighboring cells in the same lane. Then, a support vector machine classifier is trained to detect incident point.

If a cell is judged to be corresponding to an incident point at least for two successive time periods, an incident is detected and its position is calculated based on the identity number of the cell. Experiments prove the efficiency and practicability of the proposed method.

In Paper [10], This paper is on development of portable roadside magnetic sensor system for classification, speed and counting number of vehicles, this system consists of wireless anisotropic magnetic device which does not require to be embedded in the roadway devices, it measures traffic on the lane adjacent to the sensor. An algorithm based on a magnetic field is proposed to make the system robust to errors caused by large vehicles. In case of misalignment of sensors an algorithm automatically fixes it. TO measure speeds of vehicle a High Accuracy Global Positioning system is used as a reference to evaluate the accuracy, as the results show that maximum error is less than 2.5% over the rage of 5-27m/s. Vehicle classification is done based on the magnetic length and an estimate of the average vertical magnetic height of the vehicle. Vehicle length is estimated from the product of occupancy and estimated speed. This show that the system can be used to count the number of right turns at an intersection with the accuracy of 95% also this sensor is compact, portable, wireless and inexpensive.

In Paper [11], Reliable, real-time traffic surveillance is an integral and crucial function of the 21st century intelligent transportation systems (ITS) network. This technology facilitates instantaneous decision- making, improves roadway efficiency, and maximizes existing transportation infrastructure capacity, making transportation systems safe, efficient, and more reliable. Given the rapidly approaching era of smart cities, the work detailed in this dissertation is timely in that it reports on the design, development, and implementation of a novel, fully-autonomous, self-powered intelligent wireless sensor for real-time traffic surveillance. Multi-disciplinary, innovative integration of state-of-the-art, ultra-low-power embedded systems, smart physical sensors, and the wireless sensor network—powered by intelligent algorithms—are the basis of the developed Intelligent Vehicle Counting and Classification Sensor (iVCCS) platform. The sensor combines an energy-harvesting subsystem to extract energy from multiple sources and enable sensor node self- powering aimed at potentially indefinite life. A wireless power receiver was also integrated to remotely charge the sensor's primary battery. Reliable and computationally efficient intelligent algorithms for vehicle detection, speed and length estimation, vehicle classification, vehicle re- identification, travel-time estimation, time-synchronization, and drift compensation were fully developed, integrated, and evaluated.

In Paper [12], Real-time traffic surveillance is essential in today's intelligent transportation systems and will surely play a vital role in tomorrow's smart cities. The work detailed in this paper reports on the development and implementation of a novel smart wireless sensor for traffic monitoring. Computationally efficient and reliable algorithms for vehicle detection, speed and length estimation, classification, and time-synchronization were fully developed, integrated, and evaluated. Comprehensive system evaluation and extensive data analysis were performed to tune and validate the system for a reliable and robust operation. Several field studies conducted on highway and urban roads for different scenarios and under various traffic conditions resulted in 99.98% detection accuracy, 97.11% speed estimation accuracy, and 97% length-based vehicle classification accuracy. The developed system is portable, reliable, and cost-effective. The system can also be used for short- term or long-term installment on surface of highway, roadway, and roadside. Implementation cost of a single node including enclosure is US $50.

In Paper [13], Acoustic signal processing over wireless acoustic sensor networks (WASN) currently constitutes a topic receiving extensive attention by the scientific community. However, the majority of the related literature fails to consider several types of phenomena directly affecting the smooth operation of such networks, such as sensor faults, aging phenomena, environmental changes, etc. This paper provides as a basis a problem formulation systematizing such obstacles and on top of that builds a sound classification system that is able to consider the appearance of (multiple) sensor faults and environmental noise. The cornerstone of the proposed classifier is an echo state network operating at the sensor level, while the decisions are combined at a higher level via a correlation-based dependence graph. We carried out a thorough experimental campaign utilizing data coming from a WASN composed of 23 sensors aiming at the acoustic classification moving vehicles.

In Paper [14], Automatic video analysis from urban surveillance cameras is a fast-emerging field based on computer vision techniques. We present here a comprehensive review of the state-of-the-art computer vision for traffic video with a critical analysis and an outlook to future research directions. This field is of increasing relevance for intelligent transport systems (ITSs). The decreasing hardware cost and, therefore, the increasing deployment of cameras have opened a wide application field for video analytics. Several monitoring objectives such as congestion, traffic rule violation, and vehicle interaction can be targeted using cameras that were typically originally installed for human operators. Systems for the detection and classification of vehicles on highways have successfully been using classical visual surveillance techniques such as background estimation and motion tracking for some time. The urban domain is more challenging with respect to traffic density, lower camera angles that lead to a high degree of occlusion, and the variety of road users. Methods from object categorization and 3-D modelling have inspired more advanced techniques to tackle these challenges. There is no commonly used data set or benchmark challenge, which makes the direct comparison of the proposed algorithms difficult. In addition, evaluation under challenging weather conditions (e.g., rain, fog, and darkness) would be desirable but is rarely performed. Future work should be directed toward robust combined detectors and classifiers for all road users, with a focus on realistic conditions during evaluation.

In Paper [15], The paper presents a review of various sensing techniques for traffic detection and surveillance. Inductive loop detectors were extensively used for many years. Nowadays, several novel sensing techniques and wireless communication systems have been implemented. The conducted research has resulted in wealth of sensors, which are an essential part of real-time systems. A literature review is performed taking into account the new trends in the developed countries. Furthermore, a comparison between the different sensors is presented by pointing out the advantages and disadvantages of each sensor. Some future trends in this field are also discussed.

In Paper [16], This paper proposes a novel cognitive cellular automata (CA) approach to traffic management that can adapt to immediate requirements, be applied for use in cross-area car societies, enhance system performance, and decrease traffic congestion problems. They propose a mechanism that operates in a cognitive radio mode to increase the channel-reuse rate and decrease the allocation of redundant channels. This approach provides the advantage of a heterogeneous communication interface based on cognitive mechanisms that recognize different transmission modulation modes. The receiver gets messages through different transmission modulation modes. In this work, they postulate vehicles connecting to traffic congestion computing centers by vehicle-to-roadside communications within a car society. Roadside units serve each road segment, and they suppose that every car has a navigation device. They propose an innovative congestion-reduction mechanism that provides directions to a vehicle's navigation device after the driver sets the origin location and the destination. This mechanism calculates the congestion status of the upcoming road segment. By tracking the status of road segments from a point of origin to a destination, our proposed mechanism can handle cross-area car societies. The current study evaluates this approach's performance by conducting computer simulations. Simulation results reveal the strengths of the proposed CA mechanism in terms of increased lifetime and increased congestion- avoidance for urban vehicular networks.

In Paper [17], In India the unbalanced growth in the freight volume as compared to the road length has led to severe issues of traffic congestion, road safety and health hazards, which, as estimated by recent studies will surely escalate very rapidly within coming years. The use of a real time dynamic traffic management system to intelligently navigate the vehicles can optimize the traffic flow on the roads, thus utilizing the infrastructure efficiently and providing an environment conducive to solving gridlock in cities. The authors propose a real time traffic management system (RTMS) consisting of real time traffic monitoring system formed by small network of road side units, junction units and mobile units to dynamically decide the time of traffic lights to discourage formation of gridlock, coupled with a web-based application for vehicle drivers that will derive the data from real time traffic analysis to indicate the local traffic flow and suggest the incoming vehicles to make use of alternative routes in order to further alleviate growth of the gridlock.

In Paper [18], Transportation infrastructure is undergoing major revolutions in most metropolitan areas, which demands for improved operational strategies to meet requirements of smart cities. Such requirements include more convenience more travelers, and higher levels of security, reliability, economics, and societal sustainability in our communities. Given that the wide-area situational awareness is enabled by advanced information and communication technologies, this paper develops a hierarchical operation framework for regulating traffic signals effectively and flexibly in dynamic traffic conditions. The proposed framework which is based on the multi-agent system manages to mitigate potential traffic congestions and minimize drivers' average travel time in metropolitan areas. Further traffic efficiency improvements can be achieved by the utilization of a closed-loop management system. Interactive simulations are conducted in this paper to examine the performance of the proposed framework in a real-world transportation system.

In Paper [19], Since the number of vehicles is increasing day by day, traffic jams are becoming a common scenario in large cities like Dhaka. These frequent traffic jams at major junctions kill a lot of man hours. Thus, it creates a need for an efficient traffic management system. This paper proposes to implement a smart traffic control system which is based on the measurement of traffic density using real time video processing technique. The video sequences from a camera are analyzed using object detection and counting methods to obtain the most effective way. The computed vehicle density is compared with other parts of the traffic in order to control the traffic signal brilliantly. The system has an advantage of using RFID sensors to ensure law enforcement. Therefore, any car or vehicle which breaks the traffic rules can be easily caught. Through this paper they tried to present a progress in the existing manual traffic control system.

In Paper [20], Congestion in traffic is a serious problem nowadays. Although it seems to pervade everywhere, mega cities are the ones most affected by it. And its ever-increasing nature makes it imperative to know the road traffic density in real time for better signal control and effective traffic management. There can be different causes of congestion in traffic like insufficient capacity, unrestrained demand, large Red-light delays etc. While insufficient capacity and unrestrained demand are somewhere interrelated, the delay of respective light is hard coded and not dependent on traffic. Therefore, the need for simulating and optimizing traffic control to better accommodate this increasing demand arises. In recent years, video monitoring and surveillance systems have been widely used in traffic management for traveler's information, ramp metering and updates in real time. The traffic density estimation and vehicle classification can also be achieved using video monitoring systems. This paper presents the method to use live video feed from the cameras at traffic junctions for real time traffic density calculation using video and image processing. It also focuses on the algorithm for switching the traffic lights according to vehicle density on road, thereby aiming at reducing the traffic congestion on roads which will help lower the number of accidents. In turn it will provide safe transit to people and reduce fuel consumption and waiting time. It will also provide significant data which will help in future road planning and analysis. In further stages multiple traffic lights can be synchronized with each other with an aim of even less traffic congestion and free flow of traffic.

Chapter 3

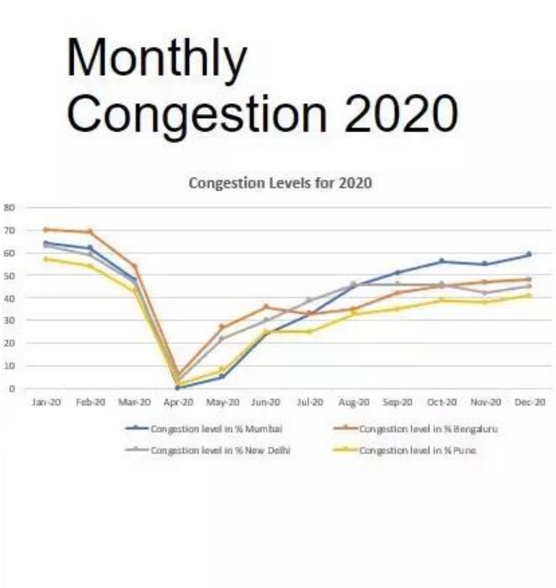
ANALYSIS

3.1 Problem Identification

Real time traffic monitoring systems play a key role in the transition towards smart cities and which help in maintaining traffic congestion and safety when there's an emergency situation on streets and highways. Identifying and measuring congestion is the very first step in traffic management process. The flow, occupancy, density is widely used traffic congestion measures, which are mostly obtained from images or videos captured by the system, based on these measures, the traffic warning messages are broadcasted through smartphones, radio, televisions, light signals, dynamic variable message signs, or display units. Among them, the mobile-based web applications received much attention since it's most common used electronic gadget these days. Most of the resent development in delivering real time traffic updates used the congestion estimates to dynamically control the traffic signal, there are many proposed system such as IoT based real time traffic monitoring for dynamic handling of traffic signals which is based on traffic density also the system uses set of ultrasonic sensors which contains two modules where one is used for vehicle monitoring and other for priority management, these sensors detect vehicles and sent the data to the LCD and the server, but this system sensors couldn't priorities the emergency vehicles such as Police cars, ambulances etc.

Then another system was proposed which was based on ultrasonic sensor which was specifically made for road interactions such as traffic lighting, alarms on any false vehicle activities such as crossing the red signals, another paper prosed a IoT based system for traffic management which manages traffic in local and central servers where the data collection layer uses cameras, sensors which then generates daily report through web applications. Further research developed based on internet connected vehicles was proposed which can collect real time data and provide traffic congestion updates and connected vehicle monitoring provides efficient emergency vehicle management with the help of road side units connected to the internet to notify or alert in such emergency cases where handling them is critical, delay of every second matter but this system works only when vehicle is connected to the internet and was designed only for highways.

The transportation project for the Beijing Olympics is a great example of providing traffic updates through public message units. The project used changeable message boards, radios, television, internet, and in-vehicle displays to monitor and dispatch traffic updates. However, system development was quite expensive due to advanced programs and devices after that, several research efforts were made in this area to provide real-time traffic updates. A system proposed to display traffic intensity through three different light colors which were installed on the electric boards at decision points, in this system the traffic density is calculated from average vehicles speed which are determined by the detection systems and then image processing algorithms are applied to process real time traffic videos and congestion estimation is based on optical flow.



Month	Congestion level in %			
	Mumbai	Bengaluru	New Delhi	Pune
Jan-20	64	70	63	57
Feb-20	62	69	59	54
Mar-20	48	54	47	43
Apr-20	0	6	4	2
May-20	5	27	22	8
Jun-20	24	36	30	25
Jul-20	33	33	39	25
Aug-20	45	35	46	33
Sep-20	51	42	46	35
Oct-20	56	45	46	39
Nov-20	55	47	42	38
Dec-20	59	48	45	41

Fig. 3.1 Traffic Congestion in major cities

Similarly, electronic sign boards are used to avoid congestion by setting different speed limits on the road. This study was tested for highways and updates were delivered through traffic signals or mobile applications, Instead, this research proposes a system model for real-time traffic updates through roadside message units using an IoT platform Nowadays, digital electronic boards are widely used in smart campuses, that can be also reused (if any) to deliver traffic updates during peak hours. Although this system contributes to the real-time traffic management, they provided only few features where some had great accuracy and some didn’t.

As this research does not anticipate any smart devices with the drivers, the traffic updates through roadside message units are analyzed in detail. The graphical message unit displays the upcoming traffic conditions and incidents through messages, signs, or colors. The studies on the impact of dynamic message signs through roadside message units show that it has received acceptance among drivers. The dynamic message signs can be delivered in permanent mode through roadside message units (installed on bridges, toll plazas, tunnels, etc.) or portable units. The portable units are mainly used to warn about unusual traffic incidents. The roadside units mostly display the messages about over spilled roads, planned activities, environmental updates, traffic flow conditions, etc.

3.2 Objectives

The objective is to provide real-time traffic updates on traffic congestion and unusual traffic incidents through roadside message units and thereby improve mobility. These early-warning messages will help citizens to save their time, especially during peak hours. Also, the system broadcasts the traffic updates from the administrative authorities. A prototype is implemented to evaluate the feasibility of the model, and the results of the experiments show good accuracy in vehicle detection and a low relative error in road occupancy estimation.

The main objectives are:

i. To estimate traffic congestions on collector roads using road occupancy measure

ii. Update residents on real-time traffic messages through roadside display units

iii. Monitor the road density of smart campuses especially during peak hours and help to improve mobility

iv. Assist authorities to broadcast important traffic incident messages

v. Provide a real-time dashboard to monitor the traffic updates

3.3 Methodology

Real-time traffic monitoring systems play a key role in the transition toward smart cities. Autonomous traffic sensing is at the heart of smart city infrastructures, wherein smart wireless sensors are used to measure traffic flow, predict congestion, and adaptively control traffic routes. Doing so effectively provides an awareness that enables more efficient use of resources and infrastructure.

Identifying and measuring congestion is the very first step in the traffic management process. The flow, occupancy, density is the widely used traffic congestion measures, which are mostly obtained from images or videos captured by vision systems initially. Based on these measures, the traffic warning messages are broadcasted through smartphones, radio, televisions, light signals, dynamic variable message signs, or display units. Among them, the mobile-based web applications received much attention.

3.3.1 Automated signalling system

Currently all signals are handled manually. A person needs to stand and keep on pressing buttons to change signals. So, in this method using NodeMCU we can make all the signals to change automatically. As no humans are involved, errors will be less.

3.3.2 Density traffic detection

In order to detect the density of the traffic at a particular signal, we place a few Infrared sensors at the distance of 10m 50m 100m.If there are no vehicles or the vehicles are up to 10m infrared sensor then the traffic density will be low. If the number of vehicles is from 10m to 50m infrared sensor then the traffic density will be medium. If the number of vehicles is from 50m to 100m infrared sensor then the traffic density will be high. When the user requires the information on traffic density then they can request, the information will be sent to user with the help of Wi-Fi present in the NodeMCU. Instead of infrared sensor, we can also use ultrasonic sensor. But ultrasonic sensor detects vehicle only if its close to the sensor whereas infrared sensor detects vehicles even if its far away. We can make ultrasonic sensor to detect vehicles that are far from sensor by connecting comparator to it.

3.3.3 Density-based traffic signal control

The traffic signals are controlled depending on the density of the traffic at a particular signal. Let's consider an example of a traffic junction, if the traffic density is low then the time of signals will be 20 seconds. If the density level is medium then the duration of the signal will be 40 seconds. If the density level is high then the duration of the signal will be 60 seconds.

3.3.4. Notification & Cloud Display

All the data that is collected by the NodeMCU is transferred to the user through Cloud.
So, by this the users can get notified about the traffic density. Depending on the traffic density, the user can decide which route to take. In case of emergency vehicles, as they can't stay in traffic, by this feature, they can decide which route has less traffic density and take that route.

3.3.5 Camera photo transmission of signals

In this method, a camera will be placed near each and every traffic signal. This camera is linked to the NodeMCU. Whenever user requests for an image with just click of a button they will receive the image. When the user requests for a photo, at that point the camera present near the traffic signal will click the image and store the image in the cloud. NodeMCU will fetch the image and with the help of Wi-Fi sends that image to the user. In this way user will receive real time image of the traffic.

3.3.6 Components

Node MCU: The ESP8266 is the name of a micro controller designed by Espress if Systems. The ESP8266 itself is a self-contained Wi-Fi networking solution offering as a bridge fromexisting microcontroller to Wi-Fi and is also capable of running self-contained applications. This module comes with a built in USB connector and a rich assortmentof pin-outs. With a micro-USB cable, you can connect a NodeMCU devkit to your laptop and flash it without any trouble, just like Arduino. It is also immediately breadboard friendly.

Fig. 3.2 Node MCU

Specification:

- Voltage: 3.3V.
- Wi-Fi Direct (P2P), soft-AP.
- Current consumption: 10uA~170mA.
- Flash memory attachable: 16MB max (512K normal).
- Integrated TCP/IP protocol stack.
- Processor: Tensilica L106 32-bit.
- Processor speed: 80~160MHz.
- RAM: 32K + 80K.
- GPIOs: 17 (multiplexed with other functions).
- Analog to Digital: 1 input with 1024 step resolution.
- 802.11 support: b/g/n

- Maximum concurrent TCP connections: 5

LCD display: The 16x2 Alphanumeric LCD Display Module is equally popular among hobbyists and professionals for its affordable price and easy to use nature. As the name suggests the 16x2 Alphanumeric LCD can show 16 Columns and 2 Rows therefore a total of (16x2) 32 characters can be displayed. Each character can either be an alphabet or number or even a custom character. This particular LCD gas a green backlight, you can also get a Blue Backlight LCD to make your projects stand our and visually appealing, apart from the backlight color both the LCD have the same specifications hence they can share the same circuit and code. If your projects require more characters to be displayed you can check the 20x4 Graphical LCD which has 20 Columns and 4 Rows and hence can display up to 80 characters.

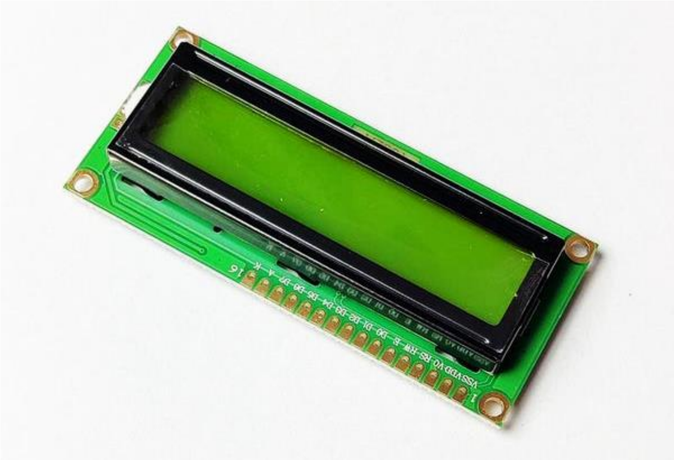


Fig. 3.3 LCD display

Specification:

- Operating Voltage: 4.7V to 5.3V
- Operating Current 1mA (without backlight)
- Can display (16x2) 32 Alphanumeric Characters
- Custom Characters Support
- Works in both 8-bit and 4-bit Mode

IR sensor: This consists of an IR transmitter LED which transmits IR light, this light will then be picked up by an IR receiver LED if it gets reflect by any object in front of it. It is commonly used in Line following robots, proximity sensing, object detection etc. It has three pins, in which two are used to power the sensor and the 3rd pins gives the output as high/low based on the proximity of the object in-front of it.

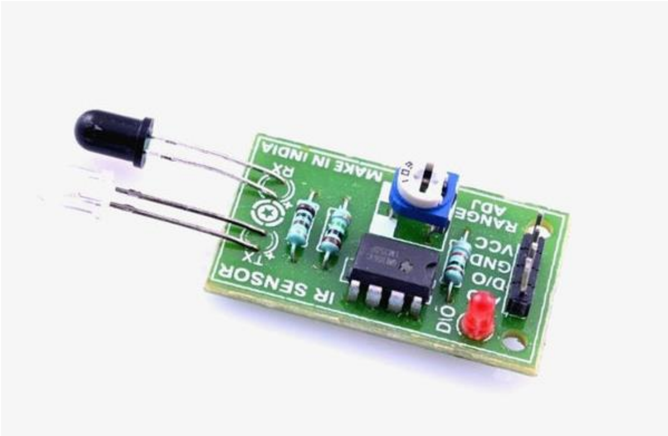


Fig. 3.4 IR sensor

Specification:

- 5VDC Operating voltage
- I/O pins are 5V and 3.3V compliant
- Built-in Ambient Light Sensor
- 20mA supply current
- Mounting hole
- Adjustable Range Detection
- LED Indicators

ESP Camera: The ESP32-CAM is an Original ESP32 CAM Wi-Fi + Bluetooth module board along with OV2640 Camera Module. This product is based on the ESP32 chip with the additional facility of using a camera.

Fig. 3.5 ESP32 Camera

That means, the features of ESP32, especially the Bluetooth and the Wi-Fi connectivity is available along with an additional peripheral OV2640 camera module. This opens up a lot of applications and usage in the field of various IoT applications**.**

Specification:

- 5VDC Operating voltage
- Cellular or Camera Phone
- IoT based surveillance camera
- Web Cam
- Image processing
- Digital still camera
- PC multimedia

Arduino IDE: Arduino first and foremost is an open-source computer hardware and software company. The Arduino Community refers to the project and user community that designs and utilizes microcontroller-based development boards. These development boards are known as Arduino Modules, which are open-source prototyping platforms.The simplified microcontroller board comes in a variety of development board packages. software specification.

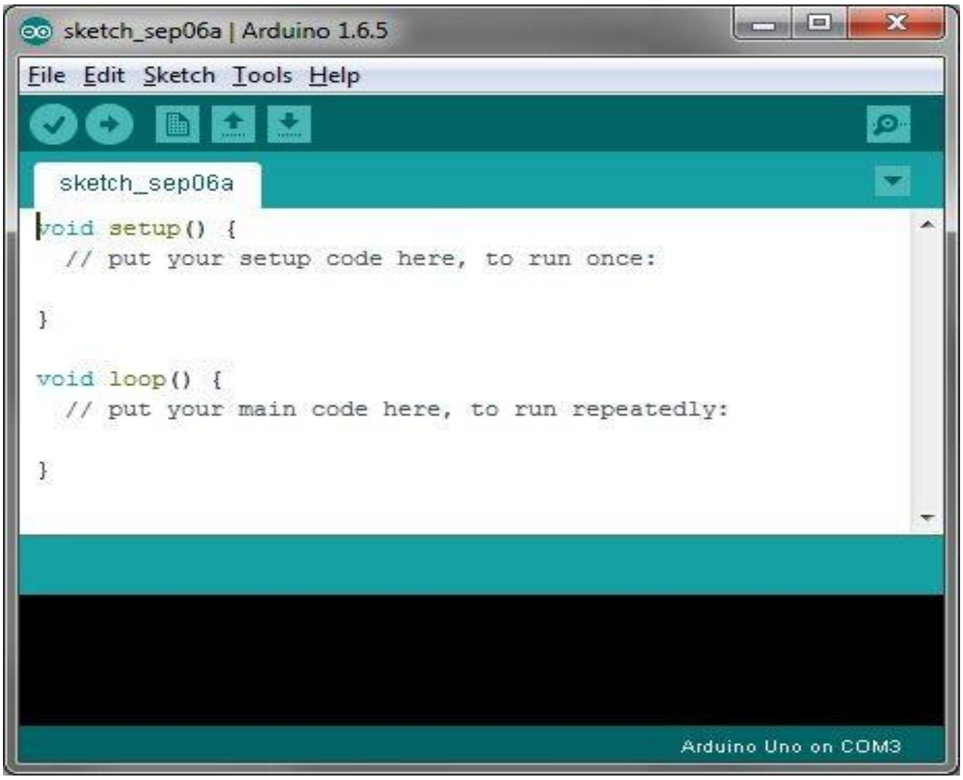


Fig. 3.6 Arduino IDE window

The most common programming approach is to use the Arduino IDE, which utilizes the C programming language. This gives you access to an enormous Arduino Library that is constantly growing thanks to open-source community.

Blynk Cloud: Blynk was designed for the Internet of Things. It can control hardware remotely, it can display sensor data, it can store data, visualize it and do many other cool things. There are three major components in the platform:

- Blynk App - allows to you create amazing interfaces for your projects using various widgets we provide.
- Blynk Server - responsible for all the communications between the smartphone and hardware. You can use our Blynk Cloud or run your private Blynk server locally. It’s open-source, could easily handle thousands of devices and can even be launched on a Raspberry Pi.
- Blynk Libraries - for all the popular hardware platforms - enable communication with the server and process all the incoming and outcoming commands.

Now imagine: every time one presses a Button in the Blynk app, the message travels to the Blynk Cloud, where it magically finds its way to your hardware. It works the same in the opposite direction and everything happens in a blynk of an eye.

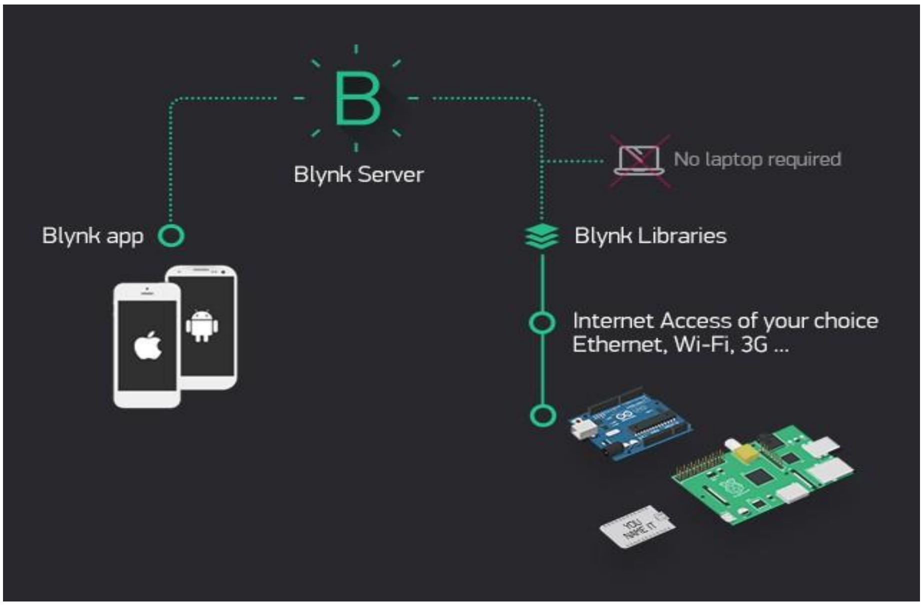


Fig. 3.7 Blynk Cloud

Features

- Similar API & UI for all supported hardware & devices
- Connection to the cloud using: Wi-Fi, Bluetooth, Ethernet, USB (Serial)
- Direct pin manipulation with no code writing.

- Easy to integrate and add new functionality using virtual pins
- Device-to-Device communication using Bridge Widget
- Sending emails, SMS, push notifications, etc.

3.4 System Requirements

3.4.1 Software Requirement Specification

- Windows 7 and above
- Arduino IDE
- Blynk
- Telegram Bot

3.4.2 Hardware Requirement Specification

- Infrared sensor
- Node MCU
- Camera
- LCD display (I2C)
- Jumper wires

3.5 Functional Requirements

A Functional Requirement defines a function of a system or its component, where a function is described as a specification of behavior between outputs and inputs.

- IoT Platforms Leverage Applications: IoT software applications are emerging for businesses in virtually every industry as well as for home users. System should be compatible.
- Remote configuration: Through this requirement it is possible to configure some parameters of the stations remotely, namely the frequency sampling of each sensor.
- Event Notification: This feature is considered a very important requirement, because it allows notification in real-time if disruptive events occur in one

station, e.g. if the water level of one tank gets down the set-point, the system sends an alert to the personin charge of the maintenance.

- Update Information: The system must allow inquiry into stations in order to attain current data. This will allow information of the status of any station and its sensors inreal-time.
- Monitoring stations status: One efficient strategy to reduce the risk of problems in water supply is by better controlling aspects such as the level and quality of the water.Monitoring the stations brings two major benefits, namely real-time analysis of theseparameters and using the data to produce statistical reports.
- Manage a Range of Devices: The number of devices connected to IoT will soon reachanywhere from 28 billion to 50 billion, depending on who you ask. IoT sensors gather information about conditions in their vicinity, such as temperature or moisture level. IoT actuators perform specific tasks, such as turning things on or off and recording information about its triggers and subsequent reactions. Thus, to manage a heterogeneous set of devices.
- Generate Massive Amounts of Data: System must be must be able to support storing massive amounts of data.
- Require Powerful Analytics: The vast volumes of data discussed above have the potential to provide unprecedented insights into consumer behavior and preferences. Unlocking those insights, however, requires powerful analytics tools. A key IoT platform functionality, therefore, is that it is capable of either incorporating — or offering compatibility with —analytics solutions that will translate significant amounts of data into useful and actionable insights.

3.6 Non-Functional Requirements

- Security: Even with the recent attention given to security for IoT devices, it can be easy to overlook the need for end-to-end security for an IoT platform. Every part of a platform should be analyzed for security prospects. From internet connections to the applications and devices to the transmitted and stored data, there is a potential for an attack vector. Without question, the single most important non-functional requirement of an IoT platform is that it offers robust security.
- Scalability: In light of the billions of devices and zettabytes of data discussed earlier, scalability is clearly a requirement in an IoT platform. The best practice for both businesses and consumers is to start small with IoT. However, many IoT solutions achieve their true potential only at scale. The ideal IoT platform is fully able to support

- A small, initial implementation, but also should be able to scale out as your business needs grow.
- Availability: Highly robust public cloud platforms have conditioned us to expect 4 or 5 nines when we think about internet availability. Those same expectations should extend to IoT platforms. In fact, there is a good reason to anticipate even higher levels of availability from IoT platforms. That is because IoT platforms can interact with and control devices that have real-world impact. IoT platforms must, therefore, offer exceptionally high availability.
- Performance: The system response time depends on how sophisticated the sensors are. If the sensors are rough (level sensors), the system will be cheaper but not so accurate when using sophisticated sensors (ultra-sonic).
- Flexibility: The system must be flexible in order to allow the user to insert, remove or edit elements, such as new stations, more sensors or adding mobile phone numbers to deliver alerts.
- Usability: A friendly interface, flexible, with strong graphical capabilities and succinct and clear messages can raise the system efficiency.
- Power supply: In order to solve the problem of remote stations located in isolated places, with difficult access, and without power supply, all these stations need to be equipped with a solar panel and a battery.

Chapter 4

SYSTEM DESIGN

4.1 System Architecture

NodeMCU is an open-source platform based on ESP8266 which can connect objects and let data transfer using the Wi-Fi protocol. NodeMCU will be connected to 3 Infrared Sensors. These Infrared Sensors will detect the traffic density. I2C (Inter-Integrated Circuit) are placed above NodeMCU. Data from NodeMCU will be in the form of binary. So, contents will not be able to understand by humans. I2C converts data from binary to human understandable language and this information is viewed using LCD. NodeMCU will be connected to Traffic Signals. Depending on the results of Infrared Sensors NodeMCU will decide the time for each traffic light. Wi-Fi present in the NodeMCU will be used to transfer data to the user.

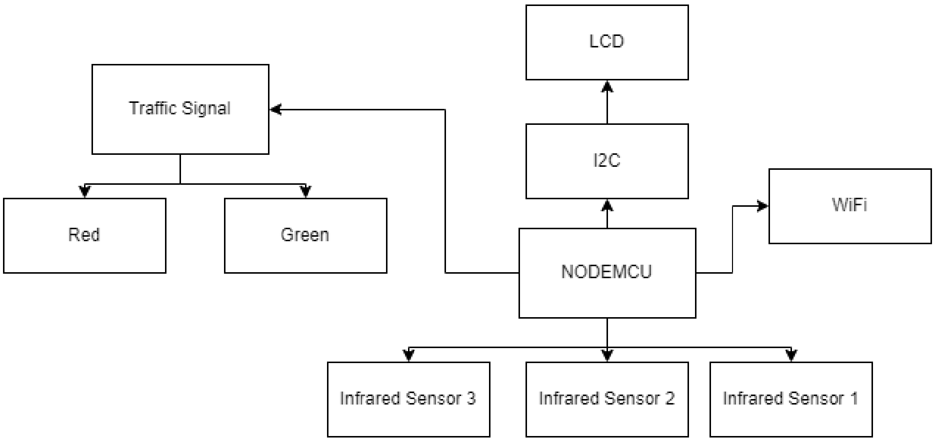


Fig. 4.1 System architecture

4.2 Detailed Design

In detailed system design, every system needs to be broken down to ascertain all activities required and their respective inputs and outputs. In some of the cases, sub systems are broadly defined in the conceptual design phase, but at this stage sub systems are specifically defined to work out every detail concerning the sub-system. Decomposition of the system to operational activities in general is carried out as follows.

4.2.1 High level design

Wi-Fi module: The ESP8266 Wi-Fi Module is a self-contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your Wi-Fi network. The ESP8266 is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor. Each ESP8266 module comes pre-programmed with an AT command set firmware. The ESP8266 module is an extremely cost-effective board with a huge, and ever growing, community.

Fig. 4.2 Wi-Fi module

4.3 Data flow diagram

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. Figure 4.3 shows the flow of data which are collected from sensors are then compared with the standard threshold, if it exceeds the standard value the alert message will be sent to the users.

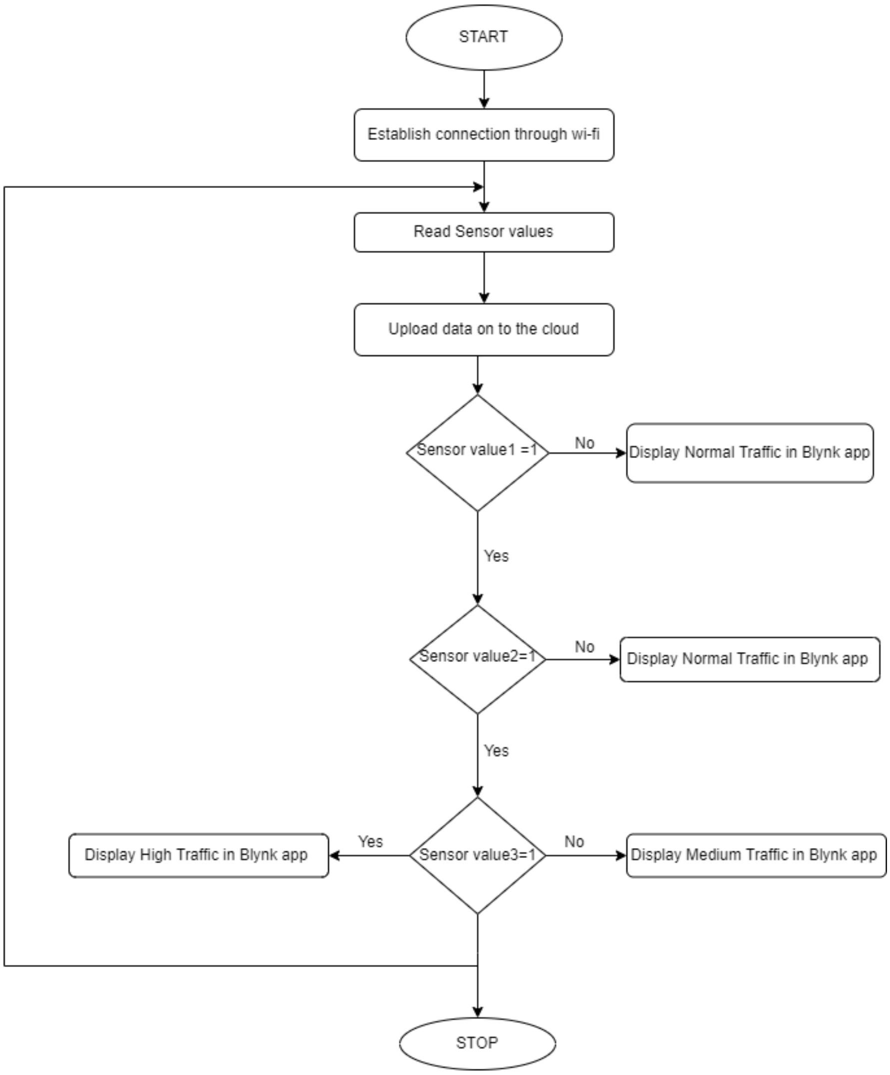


Fig. 4.3 Data flow diagram

4.4 Use case diagram

Use-case diagrams describe the high-level functions and scope of a system. These diagrams alsoidentify the interactions between the system and its actors. The use cases and actors in use-casediagrams describe what the system does and how the actors use it, but not how the system operates internally. Use-case diagrams illustrate and define the context and requirements of either an entire system or the important parts of the system. You can model a complex system with a single use-case diagram, or create many use-case diagrams to model the components of the system. You would typically develop use-case diagrams in the early phases of a project and refer to them throughout the development process.

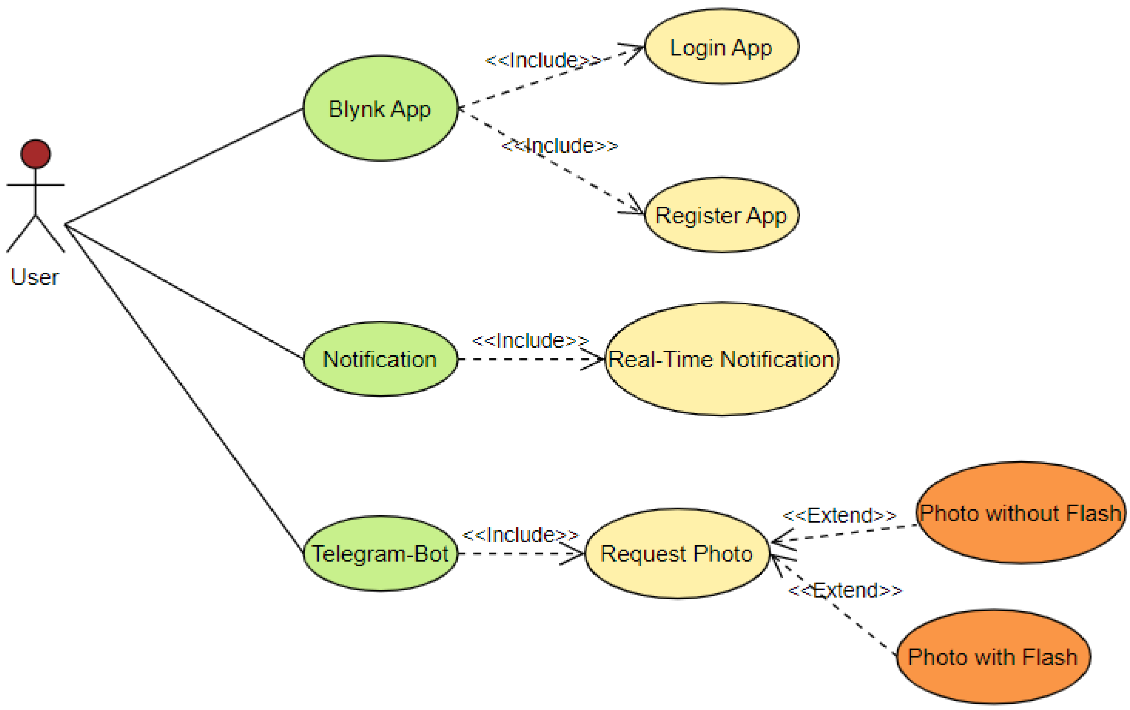


Fig. 4.4 Use case diagram

Figure 4.4 is a use case diagram of smart water quality monitoring system. In this figure,developer is considered as an actor who sets the monitoring mode, deploy sensor, define threshold, sets environment property for monitoring.

4.4 Sequence diagram

A sequence diagram is a Unified Modeling Language (UML) diagram that illustrates the sequence of messages between objects in an interaction. A sequence diagram consists of a group of objects that are represented by lifelines, and the messages that they exchange over time during the interaction.

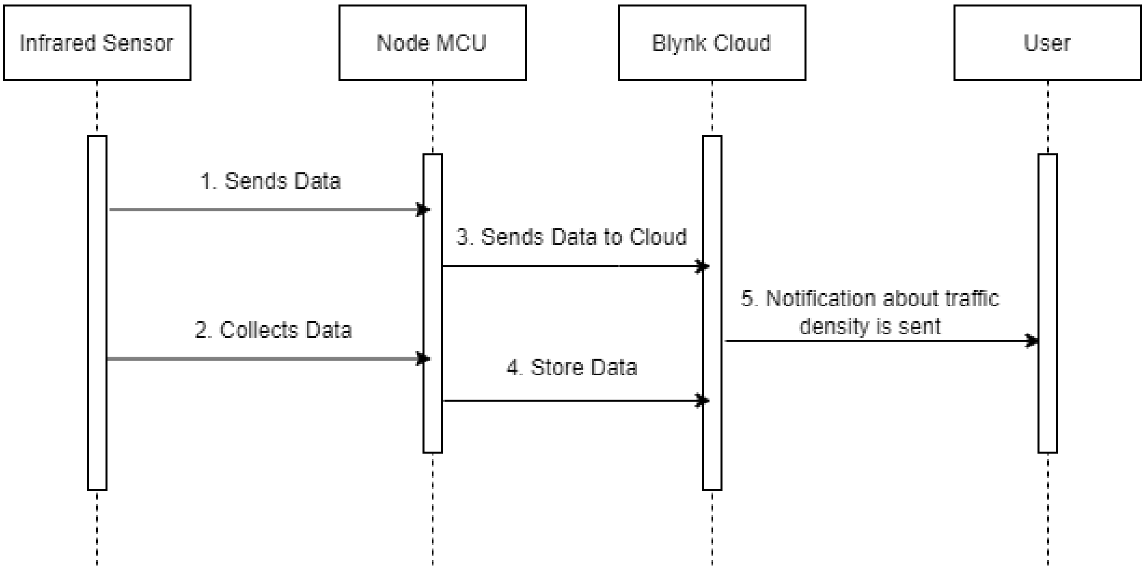


Fig. 4.5 Sequence diagram

Figure 4.5 is a sequence diagram of smart water quality monitoring system. In this system, all the data are collected from sensors and given to microcontroller. From microcontroller to the Blynk cloud. In cloud data get stored and in case any error, notification issent to the user.

.

Chapter 5

IMPLEMENTATION

5.1 Overview of System implementation

There are 3 Infrared Sensors present on the board that are connected to the NodeMCU which is mother board of our project. How Infrared Sensor works is it passes light forward. If the light reflects back then it tells that there is some object in front of it. And depending on the amount of light reflects it detects how far the object is. All the data that is detected by this sensor are passed to NodeMCU for further processing. NodeMCU is connected to traffic light. Depending on the data received by Infrared Sensors, it detects the traffic density and also how long the traffic light should be green or red.

If only Infrared Sensor 1 is detected then traffic light bill be green for 20secs and traffic density is low. If Infrared Sensors 1 and 2 are detected then the traffic light will be green for 30secs and traffic density is medium. If Infrared Sensors 1, 2 and 3 are detected then the traffic light will be green for 50secs and traffic density is high. These traffic density data is transmitted to user through Blynk app which is an open source IoT platform. There is an ESP Camera attached on top of the traffic light. This clicks photo of the traffic and sends it to user whenever they request. Users can request photo through telegram app. There are two types in which photo can be taken that is with flash and without flash.

5.2 Algorithm of proposed system

The proposed system's entire algorithm is shown. Initially, the serial monitor is initialized with 9600 baud rate. Later the ESP Wi-Fi module and the Blynk Server is also initialized. The three IR sensors are being connected and the values are read into the sensors.

The algorithm flow of IR sensor is explained. The IR sensor checks if there is some object detected in front of sensor. These values are sent to NodeMCU for further processing of data and traffic light duration as well as traffic density. Later the same values are sent into the Blynk server and thesame values are updated in the monitor

The user can access the photo from the telegram app using the bot where the user can type in /photo which will click the photo normally and if the user types /flash /photo which will click the photo with flash , this is useful in the night.

The figure 5.1 shows the proposed systems algorithms flow of the IR sensor and connection to the blynk app for monitoring the traffic signals

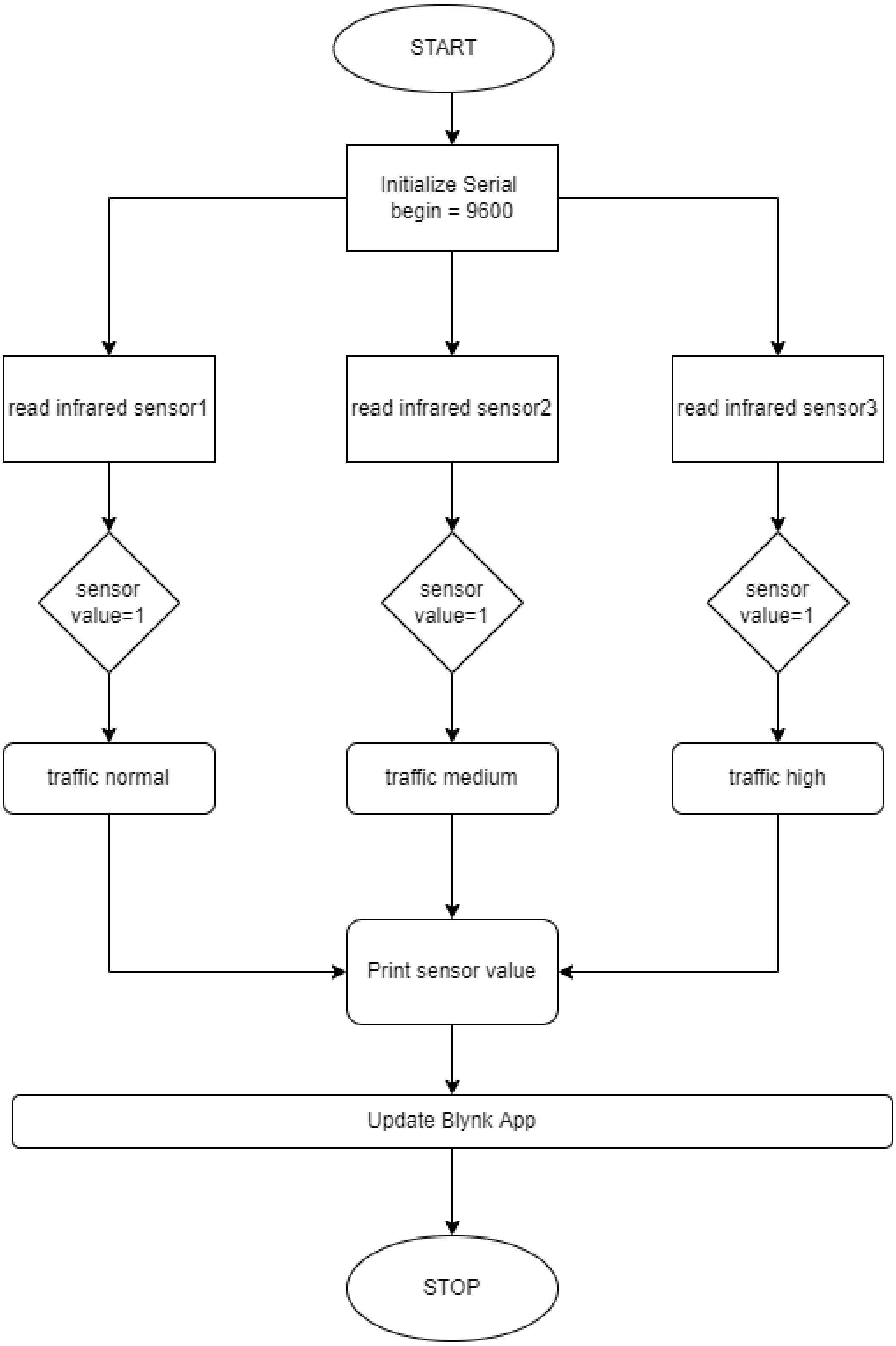


Fig. 5.1 Algorithm

5.3 Pseudocode

In computer science, pseudocode is a plain language description of the steps in an algorithm or another system. Pseudocode often uses structural conventions of a normal programming language, but is intended for human reading rather than machinereading. It typically omits details that are essential for machine understanding of the algorithm, such as variable declarations and language-specific code. The programming language is augmented with natural language description details, where convenient, orwith compact mathematical notation. The purpose of using pseudocode is that it is easier for people to understand than conventional programming language code, and thatit is an efficient and environment-independent description of the key principles of an algorithm. It is commonly used in textbooks and scientific publications to document algorithms and in planning of software and other algorithms.

This project collects the data from sensor and pass that data to a Blynk cloud. Notification is seen in Blynk cloud when data is above the threshold values.

5.3.1 Detecting sensor 1

```
if(traffic_value == 1)  {
   normal_flag = 0;
   traffic = 0;
   traffic_value = 0;
   while(traffic <=19)
   {
     traffic_value1 = digitalRead(traffic_sensor1);
     traffic += 1;
     Serial.println("Ir 1 Count:" + String(traffic));
     lcd.clear();
     lcd.setCursor(0,0);
     lcd.print("Time:"+String(traffic));
     lcd.setCursor(0,1);
     lcd.print("TRAFFIC STARTS");
     digitalWrite(red, LOW);
     digitalWrite(green, HIGH);
     delay(1000);
   }
 }
```

5.3.2 Detecting sensor 2

```
else if(traffic_value1 == 1)
 {
   normal_flag = 0;
   traffic = 0;
   traffic_value1 = 0;
   while(traffic <= 29)
   {
     traffic_value2 = digitalRead(traffic_sensor2);
     traffic += 1;
     Serial.println("Ir 2 Count:" + String(traffic));
     lcd.clear();
     lcd.setCursor(0,0);
     lcd.print("Time:"+String(traffic));
     lcd.setCursor(0,1);
     lcd.print("MEDIUM");
     digitalWrite(red, LOW);
     digitalWrite(green, HIGH);
     delay(1000);
   }
 }
```

5.3.3 Detecting sensor 3

```
else if(traffic_value2 == 1)
 {
   normal_flag = 0;
   traffic = 0;
   traffic_value2 = 0;
   while(traffic <= 49)
   {
     traffic += 1;
     Serial.println("Ir 3 Count:" + String(traffic));
     lcd.clear();
     lcd.setCursor(0,0);
```

```
    lcd.print("Time:"+String(traffic));
    lcd.setCursor(0,1);
    lcd.print("DENSE");
    digitalWrite(red, LOW);
    digitalWrite(green, HIGH);
    delay(1000);
   }
  }
```

5.3.4 Duration of red light

```
else
 {
   normal += 1;
   Serial.println("Count:" + String(normal));
   lcd.clear();
   lcd.setCursor(0,0);
   lcd.print("Time:"+String(normal));
   lcd.setCursor(0,1);
    lcd.print("Normal");
   if(normal >= 15)
   {
    traffic_value = digitalRead(traffic_sensor);

    normal_flag = !normal_flag;
    normal = 0;
   }
   if(normal_flag == 0)
   {
    digitalWrite(red, HIGH);
    digitalWrite(green, LOW);
   }
   else
   {
    digitalWrite(red, LOW);
```

```
    digitalWrite(green, HIGH);
  }
 }
```

5.3.5 Connect camera with telegram bot

```
void setup(){
 WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0);
 // Init Serial Monitor
 Serial.begin(115200);

 // Set LED Flash as output
 pinMode(FLASH_LED_PIN, OUTPUT);
 digitalWrite(FLASH_LED_PIN, flashState);

 // Config and init the camera
 configInitCamera();

 // Connect to Wi-Fi
 WiFi.mode(WIFI_STA);
 Serial.println();
 Serial.print("Connecting to ");
 Serial.println(ssid);
 WiFi.begin(ssid, password);
 clientTCP.setCACert(TELEGRAM_CERTIFICATE_ROOT); // Add root certificate for
api.telegram.org
 while (WiFi.status() != WL_CONNECTED) {
  Serial.print(".");
  delay(500);
 }
 Serial.println();
 Serial.print("ESP32-CAM IP Address: ");
 Serial.println(WiFi.localIP());
}
```

5.3.6 Check whether camera is able to connect with telegram bot or not

```
String sendPhotoTelegram() {
  const char* myDomain = "api.telegram.org";
  String getAll = "";
  String getBody = "";

  camera_fb_t * fb = NULL;
  fb = esp_camera_fb_get();
  if(!fb) {
    Serial.println("Camera capture failed");
    delay(1000);
    ESP.restart();
    return "Camera capture failed";
  }

  Serial.println("Connect to " + String(myDomain));



  if (clientTCP.connect(myDomain, 443)) {
    Serial.println("Connection successful");
```

5.3.7 Camera commands

```
for (int i = 0; i < numNewMessages; i++) {
    String chat_id = String(bot.messages[i].chat_id);
    if (chat_id != CHAT_ID){
      bot.sendMessage(chat_id, "Unauthorized user", "");
      continue;
    }

    // Print the received message
    String text = bot.messages[i].text;
    Serial.println(text);

    String from_name = bot.messages[i].from_name;
```

```
  if (text == "/start") {
    String welcome = "Welcome , " + from_name + "\n";
    welcome += "Use the following commands to interact with the ESP32-CAM \n";
    welcome += "/photo : takes a new photo\n";
    welcome += "/flash : toggles flash LED \n";
    bot.sendMessage(CHAT_ID, welcome, "");
  }
  if (text == "/flash") {
    flashState = !flashState;
    digitalWrite(FLASH_LED_PIN, flashState);
    Serial.println("Change flash LED state");
  }
  if (text == "/photo") {
    //flashState = !flashState;
    sendPhoto = true;
    Serial.println("New photo request");
    //digitalWrite(FLASH_LED_PIN, flashState);
  }
 }
```

5.3.8 Send photo to user

```
void loop() {
 if (sendPhoto) {
  Serial.println("Preparing photo");
  sendPhotoTelegram();
  sendPhoto = false;
 }
 if (millis() > lastTimeBotRan + botRequestDelay)  {
  int numNewMessages = bot.getUpdates(bot.last_message_received + 1);
  while (numNewMessages) {
   Serial.println("got response");
   handleNewMessages(numNewMessages);
   numNewMessages = bot.getUpdates(bot.last_message_received + 1);
```

```
  }
  lastTimeBotRan = millis();
 }
}
```

5.4 Implementation support

Embedded C: This language apply to programming embedded controllers. The languagein which Arduino is programmed is a subset of C and it includes only those features of standard C that are supported by the Arduino IDE.

This does not mean that Arduino C lags anywhere because it is a subset of C. Mostof the missing features of standard C can be easily worked around. Rather, Arduino C is a hybrid of C and C++, meaning it is functional and object-oriented.

The structure of sketches

Essentially, a blank Arduino sketch has two functions: Setup() and loop(). As the Arduino sketch starts executing, the setup() function is called first. It's executed only once and mustbe used to initialize variables, set pin Modes, make settings for hardware components, use libraries, etc.

The loop() function is next to the setup() function and it is iterated infinitely. Any other user-defined functions must be called inside the loop function. This is how microcontrollers execute their firmware code by repeating their code for an infinite number of times while they remain powered on.

If users have programmed other microcontrollers (such as 8051, AVR, PIC, or RX), it's possible to compare the code inside the setup() function with the one outside of the main() loop of an embedded C program — which may have been written to initialize variables and make hardware settings. The setup() and loop() functions have void return types.

A program for a microcontroller must be structured in the same manner as it functions. A microcontroller must be "aware" of its hardware environment and know howto interact with it.

A microcontroller can interact with other hardware components or devices only through these five ways:

- **Digital Input.** This may be received in digital LOW or HIGH from other devices.

- These will be TTL logic levels or voltages converted to TTL logic levels before beingapplied to the GPIO.
- **Digital Output.** This may be output that's digital LOW or HIGH compared to other devices. Again, the output will be TTL logic levels.
- **Analog Input.** It may "sense" analog voltage from other devices. The sensed voltageis converted to a digital value using a built-in, analog-to-digital converter.
- **Analog Output.** It may output analog voltage to other devices. This analog output is not analog voltage but a PWM signal that approximates analog voltage levels.
- **Serial Communication.** It may transmit, receive, or transceiver data with other devices in serial, according to a standard serial data protocol such as UART, USART, I2C, SPI, microwire, 1-wire, and CAN, etc. The serial communication with other devices can be peer-to-peer (UART/USART), half-duplex (I2C), or full-duplex (SPI).

Users that know how to perform these five types of microcontroller interactions can interface any hardware with it.

An Arduino program or any microcontroller program must first have code for initialization. This may include:

- Defining variables and constants
- Setting up pin Modes
- Setting up ADC/PWM channels
- Initializing settings for serial communications

A microcontroller simply intercepts incoming data, processes it according to programmed instructions, and outputs data through its I/O peripherals. This means the program must be organized in specific sections that can handle input data, process data, and control output. Unlike desktop applications, µc programs are not designed to terminate. These programs keep iterating for an infinite number of times until the system is shut down or it meets failure. After a power shutdown, Arduino or any microcontroller resets on the "power resume" and begins execution of its program from the beginning.

The program includes code to handle failures when possible. So, any Arduino program can be visualized as a four-step program as follows:

- **Initialization**
- **Input** – this should include code for data validation and to handle incorrect

orunexpected incoming data.

- **Processing** – this should include code for unexpected failures or exceptions raisedwhile data processing.
- **Output** – this may include code for verification of expected results if the interfaceddevice can also communicate back to the microcontroller.

Chapter 6

TESTING

Software testing is conducted to provide stakeholders with information about the quality of the software product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. It involves the execution of a software component or system component to evaluate one or more properties of interest (a programor application), with the intent of finding software bugs (errors or other defects), and verifying that the software product is fit for use.

6.1 Unit Testing

Unit testing is a software testing method by which individual units of source code, setsof one or more computer program modules together with associated control data, usageprocedures, and operating procedures, are tested to determine whether they are fit for use.

Table 6.1: Test cases for connecting hardware to cloud

Case Id	Description	InputData	Expected Output	Actual Output	Output Status
1	Connecting hardware to the Cloud via Wi-Fi	Specific Wi-Fi name and password	LCD will display “Connected to Traffic”.	Displayed	Pass
2	Connecting hardware to the Cloud via Wi-Fi	Invalid Wi- Fi name andpassword	LCD will display “Traffic not connected”.	Displayed	Pass

Table 6.2: Test cases for IR sensor 1

Case Id	Description	InputData	Expected Output	Actual Output	Output Status
1	If IR sensor 1 detected	Traffic = 1	LCD will display “Normal Traffic”.	Displayed	Pass
2	If IR sensor 1 Not detected	Traffic = 0	LCD will display “Normal Traffic”.	Displayed	Pass

Table 6.3: Test cases for IR sensor 2

Case Id	Description	InputData	Expected Output	Actual Output	Output Status
1	If IR sensor 2 detected	Traffic = 1	LCD will display "Medium Traffic".	Displayed	Pass
2	If IR sensor 2 Not detected	Traffic = 0	LCD will display "Normal Traffic".	Displayed	Pass

Table 6.4: Test cases for IR sensor 3

Case Id	Description	InputData	Expected Output	Actual Output	Output Status
1	If IR sensor 3 detected	Traffic = 1	LCD will display "Dense Traffic".	Displayed	Pass
2	If IR sensor 3 Not detected	Traffic = 0	LCD will display "Normal Traffic".	Displayed	Pass

Table 6.5: Test cases for Camera

Case Id	Description	InputData	Expected Output	Actual Output	Output Status
1	If Camera is detected	/photo	Normal photo	Displayed	Pass
2	If camera flash detected	/flash /photo	Photo with flash	Displayed	Pass

6.2 Integration Testing

Integration testing is the phase in software testing in which individual software modulesare combined and tested as a group. It occurs after unit testing and before validation testing.

Table 6.6: Integration Testing

Case Id	Description	InputData	Expected Output	Actual Output	Output Status
1	If IR sensor 1 is detected	IR1 Traffic = 1	Normal Traffic	Displayed	Pass
2	If IR sensor 1 and 2 is detected	IR1 && IR2 Traffic = 1	Medium Traffic	Displayed	Pass
3	If IR sensor 1,2,3 is detected	IR1 && IR2 && IR3 Traffic =1	Dense Traffic	Displayed	Pass
4	If IR sensor 1 and 3 is detected	IR1 && IR3 Traffic = 1	Normal Traffic	Displayed	Pass
5	If IR sensor 2 and 3 is detected	IR2 && IR3 Traffic = 1	Normal Traffic	Displayed	Pass

Chapter 7

RESULTS

The experiment was demonstrated with the IR sensor and NodeMCU which then collected the data of the traffic density and the following results were obtained from the blynk app.

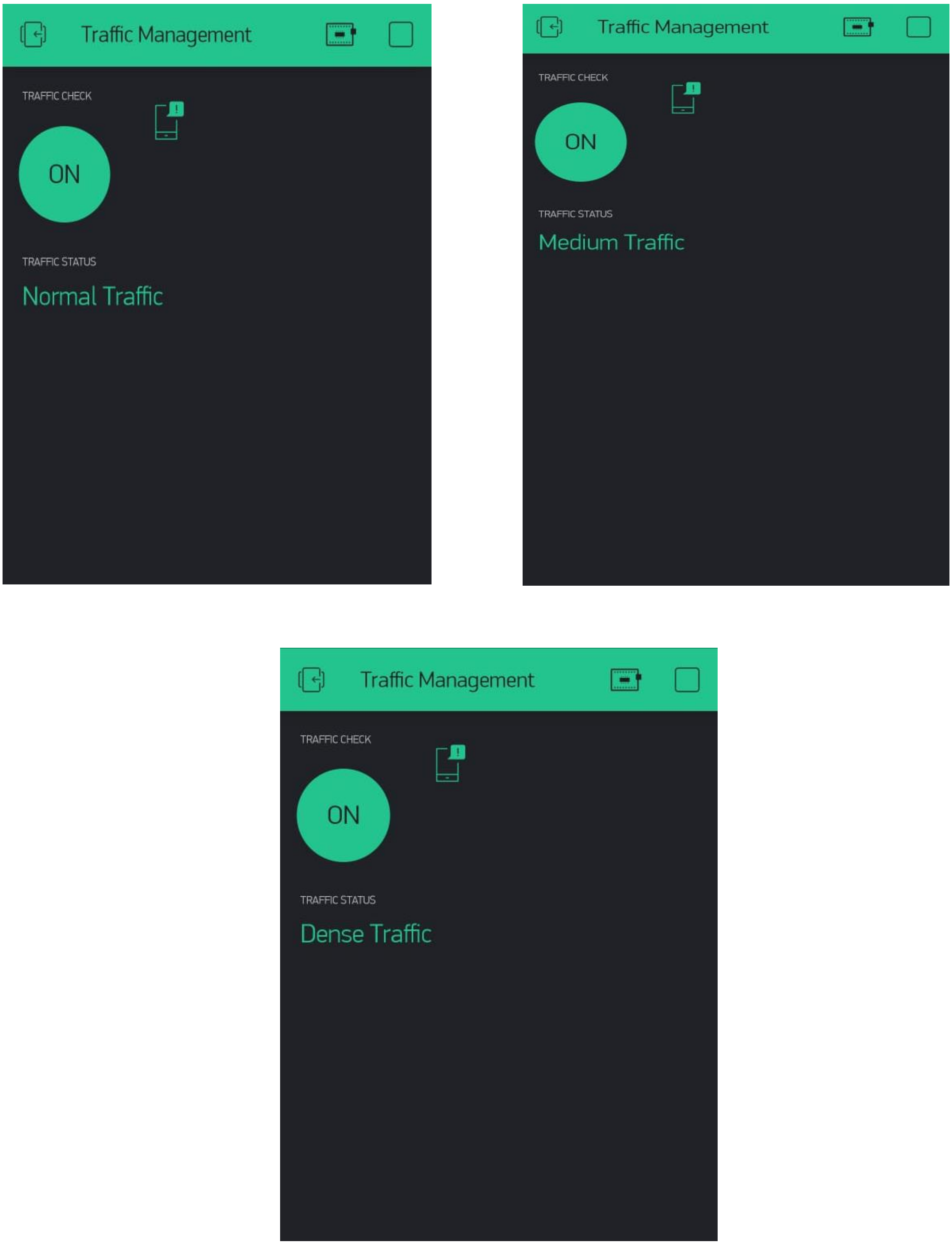


Fig. 7.1 Results from Blynk App

In the above fig 7.1, we can see the results from the Blynk app. If the traffic density is low it will show normal traffic, if the traffic density is medium then it will show medium traffic and if the traffic density is high it will show dense traffic.

Fig. 7.2 Photo without flash

In the above fig 7.2, a photo without flash is been sent to the user via telegram bot.

Fig. 7.3 Photo with flash

In the above fig 7.3, a photo with flash is been sent to the user via telegram bot.

Table 7.1: Sensor results

Sl. No	InfraRed Sensors	Time
1	**IR1**	**20sec**
2	**IR1 && IR2**	**30sec**
3	**IR1 && IR2 && IR3**	**50sec**

In the above table 7.1, we can see if only IR sensor 1 is detected then the time for green light is 20 seconds, if IR sensor 1 and IR sensor 2 is detected then the time for green light is 30 seconds and if IR sensor 1, IR sensor 2, IR sensor 3 is detected then the time for the green light is 50 seconds.

Chapter 8

CONCLUSION AND FUTURE ENHANCEMENT

CONCLUSION:

- This research provided real-time traffic monitoring for traffic updates through Blynk app.
- The users can view messages which will assist in decision making and save their time on roads.
- The real-time data is processed by Wi-Fi-enabled microcontrollers and sent to an IoT platform for further actions.
- The proposed system is expected to be considered in any smart city initiatives such as a smart university campus or any smart premises.

FUTURE ENHANCEMENT:

- Currently the system is built for only one way. In future we can implement this method in a junction and shift traffic signals using round robin methos.
- IR sensor currently only detects if the vehicle is the range of 5cms. In future we can add comparator to it and increase the detection range.
- We can use Raspberry-pi or Arduino UNO to increase the performance and enables us to use high-end cameras.
- In future we can implement for emergency vehicles so that traffic signal opens up when it detects an emergency vehicle.

REFERENCES

[1] Sodagaran, N. Zarei, Z. Azimifar, Intelligent traffic information system a real-time traffic information system on the shiraz bypass, MATEC Web of Conferences 81 (2016), 04003.

[2] L. Sumi, V. Ranga, An IoT-VANET-based traffic management system for emergency vehicles in a smart city, Advances in Intelligent Systems and Computing 708 (2) (2018) 23e31.

[3] Atta, S. Abbas, M.A. Khan, G. Ahmed, U. Farooq, An adaptive approach: smart traffic congestion control system, Journal of King Saud University - Computer and Information Sciences (2018).

[4] M.R. Islam, N.I. Shahid, D.T. Ul Karim, A. Al Mamun, M.K. Rhaman, An efficient algorithm for detecting traffic congestion and a framework for smart traffic control system, in: International Conference on Advanced Communication Technology, ICACT, 2016, pp. 802e807.

[5] M.Z. Talukder, S.S. Towqir, A.R. Remon, H.U. Zaman, An IoT based automated traffic control system with real-time update capability, in: 8th International Conference on Computing, Communications and Networking Technologies, 2017, pp. 1e6.

[6] J. Guerrero-Ib anez, S. Zeadally, J. Contreras-Castillo, Sensor technologies for ~ intelligent transportation systems, Sensors 18 (4) (2018) 1e24.

[7] S. Cheung, S. Coleri, B. Dundar, S. Ganesh, C.-W. Tan, P. Varaiya, Traffic measurement and vehicle classification with single magnetic sensor, Transport. Res. Rec.: Journal of the Transportation Research Board 1917 (January) (2005) 173e181.

[8] E. Odat, J.S. Shamma, C. Claudel, Vehicle classification and speed estimation using combined passive infrared/ultrasonic sensors, IEEE Trans. Intell. Transport. Syst. 19.

[9] Li, Y. Liu, J. Ren, Y. Chen, L. Xin, J. Shi, Detecting and positioning of traffic incidents via video-based analysis of traffic states in a road segment, IET Intell. Transp. Syst.

[10] S. Taghvaeeyan, R. Rajamani, Portable roadside sensors for vehicle counting, classification, and speed measurement, IEEE Trans. Intell. Transport. Syst. 15 (1) (2014) 73e83.

[11] W. Balid, Fully-Autonomous Self-Powered Intelligent Wireless Sensor for Real-Time Traffic Surveillance in Smart Cities, University of Oklahoma, 2016.

[12] W. Balid, H. Tafish, H.H. Refai, Intelligent vehicle counting and classification sensor for real- time traffic surveillance, IEEE Trans. Intell. Transport. Syst. 19 (6) (2018).

[13] S. Ntalampiras, Moving vehicle classification using wireless acoustic sensor networks, IEEE Transactions on Emerging Topics in Computational Intelligence 2 (2) (2018).

[14] N. Buch, S.A. Velastin, J. Orwell, A review of computer vision techniques for the analysis of urban traffic, IEEE Trans. Intell. Transport. Syst. 12 (3) (2011) 920e939.

[15] D.D. Romero, A.S. Prabuwono, Taufik, A. Hasniaty, A review of sensing techniques for real- time traffic surveillance, J. Appl. Sci. 11 (2010) 192e198.

[16] P. Kanathantip, W. Kumwilaisak, J. Chinrungrueng, Robust vehicle detection algorithm with magnetic sensor, in: ECTI-CON 2010 - the 2010 ECTI International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, 2010, pp. 1060e1064. January 2010.

[17] E. Likotiko, D. Petrov, J. Mwangoka, U. Hilleringmann, Real time solid waste monitoring using cloud and sensors technologies, Tosjat 8 (1) (2018) 106e116.

[18]] B. Li, Y. Liu, J. Ren, Y. Chen, L. Xin, J. Shi, Detecting and positioning of traffic incidents via video-based analysis of traffic states in a road segment, IET Intell. Transp. Syst. 10 (6) (2016) 428e437.

[19] K. Nellore, G.P. Hancke, A survey on urban traffic management system using wireless sensor networks, Sensors 16 (2) (2016a).

[20] B. Yang, Y. Lei, Vehicle detection and classification for low-speed congested traffic with anisotropic magnetoresistive sensor, IEEE Sensor. J. 15 (2) (2014) 1132e1138.