A Major Project Stage-II Report On

Tollgate Traffic Monitor & Analyzer Using Raspberry Pi

Submitted in Partial Fulfillment of the Requirements
For the Award of the Degree of

Bachelor of Technology

In

Electronics & Communication Engineering

By

Valpadasu Sai Sudhamsh (HT. NO: 18831A04B2)

Under the guidance of
Mrs. S. Vasanti
Asst. Professor

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

GURU NANAK INSTITUTE OF TECHNOLOGY
(AUTONOMOUS)

(Approved by AICTE, New Delhi, NBA Accredited ECE, ME, CSE & Affiliated to JNTUH) Campus: Ibrahimpatnam, R.R. District – 501506 (T.S.) Ph: (0/95) 8414-202120/21 Fax: 08414-223344 City Office: B2, 2 nd floor, Above Bata, Vikrampuri Colony, Secunderabad – 500009, Ph: 040 – 66323294, 65176117 Fax: 040 – 27892633)

2021-2022

GNIT Guru Nanak Institute of Technology

(Approved by AICTE, New Delhi, NBA Accredited ECE, ME, CSE & Affiliated to JNTU Hyderabad)Campus: Ibrahimpatnam, R.R. District – 501506 (T.S.) Ph: (0/95) 8414-202120/21 Fax: 08414-223344
City Office: B2, 2 nd floor, Above Bata, Vikrampuri Colony, Secunderabad – 500009, Ph: 040 – 66323294, 65176117 Fax: 040 – 27892633

C E R T I F I C A T E

This is to certify that the Project report entitled "***Tollgate traffic monitor & analyzer using IoT***" is being submitted by ***Valpadasu Sai Sudhamsh (18831A04B2)*** in partial fulfillment for the award of the Degree of Bachelor of Technology in ***Electronics And Communication Engineering*** of Jawaharlal Nehru Technological University Hyderabad during the year 2021-2022. The Project report has been approved as it satisfies the academic requirements in respect of Major Project work prescribed for Bachelor Degree.

Internal Guide

Mrs. S. VASANTI
Assistant Professor
Department of ECE

Project Coordinator

Mr. G. MADHU
Assistant Professor
Department of ECE

Head of the Department

Dr. S.P. YADAV
Department of ECE

Principal

Dr. S. SREENATHA REDDY
GNIT

External Examiner

GNIT Guru Nanak Institute of Technology
(Approved by AICTE, New Delhi, NBA Accredited ECE, ME, CSE & Affiliated to JNTU Hyderabad)Campus: Ibrahimpatnam, R.R. District – 501506 (T.S.) Ph: (0/95) 8414-202120/21 Fax: 08414-223344
City Office: B2, 2 nd floor, Above Bata, Vikrampuri Colony, Secunderabad – 500009, Ph: 040 – 66323294, 65176117 Fax: 040 – 27892633

DECLARATION

We, hereby certify that the project work which is being presented in the thesis entitled ***"Tollgate traffic monitor & analyzer using IoT"*** is being submitted by ***Valpadasu Sai Sudhamsh (18831A04B2)***in partial fulfillment of requirements for the award of degree of ***B.Tech. (Electronics And Communication Engineering)*** submitted in the ***Department of ECE*** is an authentic record of our own work carried out during a period from ***August 2021 to February 2022.***The matter presented in this thesis has not been submitted by us in any other University/ Institute for the award of any degree.

Signature of the Students

Mohd Abdul Aziz **R. Praveen** **V. Sai Sudhamsh**

This is certifying that the above statement made by the candidates is correct to best of my knowledge.

Signature of the Internal Guide

Mrs S. Vasanti

The B.Tech Viva-Voice Examination of **Mr. Mohd Abdul Aziz, Mr. Ramjelli Praveen & Mr. Valpadasu Sai Sudhamsh** has been held on ________________________and it is accepted.

Signature of HOD **Signature of External Examiner**

ACKNOWLEDGEMENT

The project entitled ***"Tollgate traffic monitor & analyzer using IoT "*** has seen the light of the day because the success of this project is attributed to the intensity, drive and technical competence of many individuals who contributed to make this project a reality despite all odds. So, we express heartfelt pleasure for the people who helped us in this project and gave a never ending support right from the stage the idea was conceived.

We deeply express our sincere thanks to internal guide **Mrs. S. VASANTI** whose initiation to bring up this project

We express our sincere thanks to **Mr. G. MADHU**, Project Coordinator and my entire faculty, without their constant encouragement and everlasting support this work would not have been possible.

We are very much thankful to **Dr. S.P. YADAV**, Head of ECE Department for his excellent guidance and constant encouragement throughout the project work. His timely advices and endless debates ensured the completion of project report in time.

We would like to express my profound sense of gratitude to **Dr. S. SREENATHA REDDY**, Principal for his constant and valuable guidance.

We are deeply indebted to Honorable **Dr. H. S. SAINI**, Managing Director of Guru Nanak Institutions, for constant Cooperation during the course of work.

Last but not least, we would like to thank all teaching and Non-teaching staff members of ECE department for their support and also, we would like to thank God for all good deeds.

In All Sincerity,

Mr. Valpadasu Sai Sudhamsh(18831A04B2)

GNIT Guru Nanak Institute of Technology

(Approved by AICTE, New Delhi, NBA Accredited ECE, ME, CSE & Affiliated to JNTU Hyderabad)

VISION AND MISSION OF INSTITUTE

Vision:

To be a World-class Educational and Research Institution in the service of humanity by promoting high quality Engineering and Management education. Mission:

M1: Imbibe soft skills and technical skills.

M2: Develop the faculty to reach international standards.

M3: Maintain high academic standards and teaching quality that promote analytical thinking and Independent judgement.

M4: Promote research, innovation and product development in collaboration with reputable Foreign Universities.

M5: Offer collaborative industry program in the emerging areas and spirit of enterprise.

Quality Policy:

GNITis committed to provide quality education through dedicated and talented faculty, world class infrastructure, labs and advanced research center to the students.

VISION and MISSION OF THE DEPARTMENT

The Vision of the Electronics and Communication Engineering department is: To be recognized as a leading Electronics & Communication Engineering department in the region by students and employers and be known for leadership and commitment to foster quality teaching, learning, research and innovation.

The Mission of the Electronics and Communication Engineering department is:

1. Nurture young individuals into knowledgeable, skillful and ethical professionals in their pursuit of Electronics and Communication Engineering.
2. Nurture the faculty to expose them to world-class infrastructure.
3. Sustain high performance by excellence in teaching, research and innovations.
4. Extensive partnerships and collaborations with foreign universities for technology up gradation.
5. Develop Industry-Interaction for innovation and product development.

PROGRAM EDUCATIONAL OBJECTIVES (PEO's):

•**PEO-1:** Graduates shall have the ability to apply knowledge and technical skills in emerging areas of Electronics and Communication Engineering for higher studies, research, employability, product development and handle realistic problems.

•**PEO-2:** Graduates shall possess managerial skills, maintain ethical conduct, sense of responsibility to serve the society and protect the environment.

•**PEO-3:** Graduates shall possess academic excellence with innovative insight, soft skills, leadership qualities, knowledge of contemporary issues for successful professional career.

PROGRAM OUTCOMES (PO's):

1. Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems. Department of ECE,GNIT.
2. Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics natural science, and engineering sciences.
3. Design / development of solutions: Design solutions for complex engineering problems and design system components or process that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, society and environmental consideration
4. Conduct Investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. Ethics: Apply ethics principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOs) of the ECE Department:

After completion of the program the graduate will be able to:

1. Apply the fundamental knowledge and skills of Electronics to design circuits and develop the Integrated Circuits for various applications in the field of communication, signal processing and in system automation to address National/Global issues
2. Product development by using cutting edge software and hardware tools of Electronics and communication engineering.

ABSTRACT

A toll road, also known as a turnpike or toll way, is a public or private roadway for which a fee is assessed for passage. It is a form of road pricing typically implemented to help recoup the cost of road construction and maintenance, which amounts to a form of taxation. Generally, at every tollgate, a person is appointed at every counter. This person's job is to collect the fee from the traveller and enter the details of the vehicle then press a switch to open the gate by giving a receipt to the traveller. There are also other kinds of tollgates that are automated up to a certain degree. But they are not very meticulous in terms of security and reliability. Hence we propose a system to overcome the drawbacks of the systems employed at tollgate that are currently in use. Here in our proposed system, we have integrated a camera and several modules including an RFID tag into a Raspberry Pi. The system works by opening the gate when an RFID tag is swiped at the tollgate only if the data from the RFID tag matches with the data read by the camera (which is the number plate of the vehicle captured by the camera) and also if the read details matches with the data present in the database. The camera implemented in this system works on the principle of Optical Character Recognition (also known as OCR). Moreover, whenever an RFID tag is swiped at the tollgate, the captured data is stored on the local database with other relevant information.

INDEX

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

WSN	-	Wireless Sensor Network
IR	-	Infrared Sensor
RFID	-	Radio Frequency Identification
IDE	-	Integrated Development Environment
OS	-	Operating System
GPIO	-	General Purpose Input/Output
HDMI	-	High-Definition Multimedia Interface
GPL	-	General Public License
UI	-	User Interface
UNIX	-	UNiplexed Information Computing System
HMTL	-	Hyper Text Markup Language
GUI	-	Graphical User Interface
XML	-	Extensible Markup Language
PIL	-	Python Image Library
OCR	-	Optical Character Recognition

CHAPTER 1

INTRODUCTION

As we all know, transport is the backbone of any country's economy. Each and every day more and more vehicles are increasing rapidly and the graph of the rate of buying vehicles is exponential, which has become a major problem at the site of toll booths due to heavy traffic causing endless number of problems such as high petrol/diesel consumption leading to depletion of hydrocarbon deposits below earth's crust and also death causalities due to heavy traffic.

Figure 1.1: Tollgate Lanes

1.1 USE FOR THE SOCIETY:

The sole purpose of this paper is to motivate cashless transactions by installing automated e-toll collection system and the technology that we used is the use of RFID readers/tags and some sensors. This system is capable of saving time as well as fuel conservation which can save a lot of individual's economy. This particular system is far much better and very efficient towards people as they will not stay in a long and lengthy queue thus automated e-toll system will eliminate the hardships of people parking vehicles in a long queue. RFID has the potential of eliminating corruption at local level and also reduce operational costs as well as errors in human operations. WSN's i.e. wireless sensor networks are basically used in different scenarios such as home, office, healthcare, agriculture and also at toll collection plaza which can capture and transmit data from all incoming vehicles and outgoing vehicles because of their consistent and distinctive properties.

1.2 EXISTING SYSTEM:

There are two types of systems that are currently employed for the passage of vehicles near the tollgates.

The first system is the traditional system that is being employed for the past few decades. In this system at each and every gate counter, a person is to be present whose duty is to collect the toll fee and provide a receipt to the traveler. This person has to update the information in the system and also open and close the gate manually.

The second and the latest system that is being employed is with the use of RFID tags attached to vehicles. These RFID tags affixed to the vehicles are scanned by the RFID scanners present at the gates. Based on the authentication from the system the gates are automatically opened and the toll fee is debited from the traveller's account. But these systems are not very efficient.

Figure 1.2: Existing Cash Collecting Tollgate Systems

1.3 DRAWBACKS OF THE EXISTING SYSTEM:

- These existing systems require a lot of manpower.
- These systems lead to queuing up of the following vehicles for a very long time. Considering the average time taken by a vehicle, every year 72,000 vehicles simply

wait for 5 hours by the engine ON condition, thereby aiding pollution and wasting fuel and money.

- Also, there are chances of escaping from the payment of toll tax.
- These methods for toll collection are time-consuming.

1.4 PROPOSED SYSTEM:

The proposed system consists of a Raspberry Pi 3 module for which a camera module, RFID Module (Reader & Tags), IR sensor, and a Servo Motor are integrated. Whenever the RFID tag is swiped through the RFID reader and as soon as a vehicle is detected by the IR sensor the camera module captures the image of the vehicle targeting the number plate of the car. The captured image is processed by acknowledging the vehicle's number plate using Tesseract-based OCR and OpenCV this is stored on the local system. Hence when the RFID is scanned at the toll gate the system verifies if the data processed by the camera matches with the data on the RFID and the data that is already present in the local database, if the data matches the servo motor is turned on to open the gate and close it after a certain time. If the data doesn't match or a fake RFID is being used a buzzer is triggered upon a security person is called. All the data collected is stored on the local system for analysis like traffic density, total revenue, etc.

1.5 ADVANTAGES OF THE PROPOSED SYSTEM:

- By the proposed system the waiting time at the toll gates can be decreased which in turn reduces traffic as well as pollution from the vehicles.
- Vehicle number plate breaches or theft vehicles can be identified easily.
- The use of manpower is reduced on a large scale as everything is automated.

CHAPTER 2

LITERATURE SURVEY

S.No	Authors	Title of the Paper	Components they used	Year
1.	R. Gowri Poornima	IOT Based Vehicle Parking and Toll-Fee Management System	Raspberry Pi 3, Camera, IR Sensor, GSM Module, Servo Motor	2015
2.	Kosemani O. O, Olayiwola O. E, Vincent O. R	Estimating Traffic Intensity at Toll Gates Using Queueing Networks	Camera,Digital Video Recorder, PC	2014
3.	Senthil Kumar Janahan, M.R.M. Veeramanickam, S. Arun, Kumar Narayanan, R. Anandan, Shaik Javed Parvez	IoT based smart traffic signal monitoring system using vehicles counts	Arduino Uno, IR sensor, Camera, Servo Motor	2018
4.	Ranjitha M, Spoorthi Jwanita, Soumya K	IOT Based Smart Traffic Management System.	---	2020
5.	P. S. Hanwate, Narin Meher, Ashlesh Mandke, Manoj Nikam, Manan Mehta	Smart Toll Collection System based on IoT	Arduino Uno, RFID Module, IR Sensor, Stepper Motor	2017
6.	Priyanka .V, Punitha .S.P, Keerthika .K	Automation of Tollgate and Vehicle Tracking System	Atmega 162, RFID Sensor, Proximity Sensor, Relay, GSM Module	2013
7.	Sudha M S , Yashodhre K, Snehil Sarkar, Spoorthy G, Shubham Khosla	IoT Based Toll Booth Management System	Arduino Mega 2560, Matrix Keypad, Wifi Module, Servo Motor, GSM, Buzzer, LCD Screen, RFID Module	2020
8.	P. Keerthana, T. Sangeetha, N. Surekha	Smart Tollgate System Using IoT	Camera, Arduino Uno, LCD Screen, RFID Module, Pressure Sensor, IoT Module, Servo Motor	2018

9.	Narayana Darapaneni; Parikshit Bangade; Amit Mane; Umang Maheshwari; Sushilkumar C Thorawade; Rushikesh Borse	Traffic Monitoring and Analysis At Toll Plaza	AT89S52, RPS, RFID Reader & Tags, ESP8266, Servo Motor	2021
10.	Elmer R. Magsino; Ivan W. H. Ho	An intelligent highway tollgate queue selector for improving server utilization and vehicle waiting time	Raspberry Pi 3, Camera, IR Sensor, RFID Kit, Servo Motor, Matlab	2016

1. R. Gowri Poornima. "IOT Based Vehicle Parking and Toll-Fee Management System". International Conference on Communications, Signal Processing, and their Applications (ICCSPA'15), Sharjah, 2015.

Abstract: The Internet of Things (IOT) refers to the ever-growing network of physical objects that feature an IP address for internet connectivity, and the communication that occurs between these objects and other Internet-enabled devices and systems. IoT extends internet connectivity beyond traditional devices like desktop and laptop computers, smart phones and tablets to a diverse range of devices and everyday things that utilize embedded technology to communicate and interact with the external environment, all via the Internet. The basic desideratum of the vehicle parking and toll-fee management system is to reduce the waiting time of the vehicles. There are huge amount of vehicles passing through Toll Gate Stations and using parking places every day. This IOT based management system collects vehicle toll fee or parking fee automatically. In this system camera is used for capturing the image of the vehicles number plate. The captured image would be converted into the text using Python based OCR (optical character recognition) and the toll would be cut from the user's account and then the gate is opened. The system is divided into the design of three modules, Vehicle Module and the Database Module, Tollgate station. The three modules communicate via GSM modem connected to each module, stolen vehicle are also identified and alarm would be buzzed. For the Identification of the vehicles, the information of the vehicles is already stored on the database. Stored information is also easily exchanged between the motorists and toll or parking authorities, thereby enabling a more efficient toll or parking fee collection by reducing waiting time and traffic.

2. Kosemani O. O, Olayiwola O. E, Vincent O. R. "Estimating Traffic Intensity at Toll Gates Using Queueing Networks". International Journal of Advanced Computer Science and Applications, Vol. 5, No. 4, 2014.

Abstract: Traffic information generation is a routine-like operation that is done on a daily basis at any public gate. A toll gate is a public roadway by which people enter and leave a public organization. The existing models give premium consideration to security over prompt services and as such associated with processes that have high cost of implementation, inaccuracy from complex method as well pose other technical problems such as delay. This research presents an automated procedure for monitoring traffic at toll gates to give the best compromise among the conflicting objectives of payment, security and good services. The system gathers information about the traffic situation with respect to the license plate number captured from each vehicle that passes through the toll gate and as well captures data such as arrival speed, arrival time and date and uses this data as input to generate traffic report/information on a daily basis. Experimentally the system shows that it can effectively capture the vehicle video and detect the license plate in day time, showing accuracy of about 85% to 90%, practical results based on actual data are included.

3. Senthil Kumar Janahan, M.R.M. Veeramanickam, S. Arun, Kumar Narayanan, R. Anandan, Shaik Javed Parvez."IoT based smart traffic signal monitoring system using vehicles counts". International Journal of Engineering & Technology, 7 (2.21) (2018) 309-312.

Abstract: One of the important things in the Internet of things in smart cities is the Intelligent Transportation System (ITS). ITS improves Vehicle to vehicle and Vehicle to Infrastructure communication for improving road facilities rather than increasing road capacities or developing new roads. This is possible because of ITS, it utilizes advanced information and communication, and this communication will be helpful for decreasing traffic congestion and to reduce the accidents on the road, which is dangerous in the urban areas. Traffic signal

management is one of the major problematic issues in the current situation. Such scenarios, every signal are getting 60 seconds of timing on the road at a regular interval, even when traffic on that particular road is dense. As per this proposed model in this article, which will be optimized the timing interval of the traffic signal purely depends on the number of vehicles on that particular roadside. The major advantage of this system is that it can able to decrease the more waiting time for the drivers to cross road signal. In this model, we are using the clustering algorithms model which is based on KNN algorithm. Using this algorithm new model will be liable to determine expected required timing as per provided inputs to the signal which is vehicles count. The input of these systems is vehicles counts on each side of the road from crossing signal. And this input will be determined on much time is to be provided. “Case studies on this system are traffic network and real-time traffic sub-networks are organized to get the effectiveness of the proposed model.

4. Ranjitha M, Spoorthi Jwanita, Soumya K. “IOT Based Smart Traffic Management System”. International Journal of Engineering Research & Technology (IJERT) 2020.

Abstract: The wide variety of motors on the street has risen dramatically in current years. Congestion is a growing trouble that everybody offers with on an everyday basis. Manual site visitors manage through site visitor’s law enforcement officials has now no longer established to be effective. A version is designed to efficiently clear up the above noted troubles through the usage of Internet of Things (IOT). A community of sensors is hired to hint the quantity of motors and the site visitor’s congestion on the intersections on a road, and rerouting can be primarily based totally at the site visitor’s density at the route's lanes. With growing populace and no of motors on road, the site visitors may be predicted to be excessive and control of that site visitors manually may be greater difficult. This task is to offer help to the site visitor’s policemen through developing an interconnection among the motors primarily based totally on cloud connection in order that the site visitors may be monitored automatically. Additionally, automatic ignition primarily based totally at the biometrics permits simplest the customers with allowable license to drive. Violation and site visitor’s offences are effortlessly captured and fined primarily based totally at the wide variety plate of the car and presently logged in user. In case of injuries or emergencies, nearest ambulance will acquire notification consisting of the closest clinic with all required information so the docs can take movement as required or create an alert to folks that set GPS on excessive congestion zones to deviate to a low congestion direction until truly necessary.

5. P. S. Hanwate, Narin Meher, Ashlesh Mandke, Manoj Nikam, Manan Mehta. "Smart Toll Collection System based on IoT". IJSTE - International Journal of Science Technology & Engineering, June 2017.

Abstract: With the growth in the number of vehicles, the need for expansive roads catering to thousands of motor vehicles transverse across India has become inevitable. However, the demand for the construction of these roads cannot be met by public fund alone. Government has taken initiative to construct the road through different methods like PPP models etc. This revenue is collected mainly through toll booths. However, considering the present situation the modern toll system has some limitations. Due to the limited number of toll booths and slow collection process, the average waiting time per vehicle is more. This outcomes in loses worth thousands of cores of rupees regarding fuel wastage. In addition, there are various cases of toll plaza accidents which happen due to the sudden lane changing by drivers for quicker clearance. The use of automated toll collection system in many metropolitan cities would be an efficient step towards the overcrowding of the city highways in heavy congestion of traffic. As we all know, transportation is the strength of our country's economy. There are various implementations, protocols in wireless sensor network such as leach-c and components such as RFID, NFC thus enabling reduction in operation costs and motivating cashless transactions. In case of manual toll collection system time consumption is much far worse as well as fuel depletion and most important is the environment, the amount of air pollution that is created at the toll booth site is at high level, so our developed system will reduce time wastage and not only reduce air pollution but also conserve fuel. The sole purpose of this paper is to reduce the hardships caused by manual toll collection system and pass the subject's vehicle through toll barrier in a matter of few seconds without halt.

6. Priyanka .V, Punitha .S.P, Keerthika .K. "Automation of Tollgate and Vehicle Tracking System". International Journal of Science and Research (IJSR), 2013.

Abstract: Automation of Toll Gate and Vehicle Tracking System is designed to automatically keep track of the vehicle's movement, record the time and the details like Owner's name, contact details, vehicle registration number, vehicle model etc. A computerized system automatically identifies an approaching vehicle and records the vehicle details. RFID-based automation of Toll gate and vehicle tracking system is designed to automatically process the toll gate system without any manual power. This system is very useful for automatic bomb detection, vehicle monitoring, time management and also for automation of payment system. In this paper, we propose an automatic system of toll gate for monitoring and controlling the entry of vehicle. This RFID-based system automatically records time and the details of vehicle's entry time, owner's name, mobile number and vehicle model etc. A passive RFID tag is used to have information about the vehicle registration number etc. And also read by RFID reader which is located at computerized system. An IR communication is proposed to count the number of vehicles which enters the toll gate. In this system, a computerized system automatically checks the details like vehicle owner's name, vehicle registration number and mobile number etc. If the vehicle belongs to the authorized person/group it automatically opens the Toll gate. A proximity sensor is used to detect the presence of weapons in the vehicle without any physical contact with the object. A pre-determined amount will be automatically deducted from vehicle owner's account and it will be shortly intimated to the user by short message service. This automation of Toll gate and vehicle tracking system is proposed to increase the traffic control in a very efficient and effective manner.

7. Sudha M S , Yashodhre K, Snehil Sarkar, Spoorthy G, Shubham Khosla. "IoT Based Toll Booth Management System". International Journal of Progressive Research in Science and Engineering Volume-1, Issue-2, May-2020.

Abstract: In the present days by the ever increasing number of vehicles day by day on road has led to the immense traffic and also made the duration of the journey slow. This slow moving traffic not only is the burden in the cities but also on the highway where people stand in queue just in order to pay toll fee to enter the highway. This automated toll booth

management system is more effective and well organized as people are not stuck in the long queue thereby eliminating the waiting time. RFID based toll system has the capability of eradicating corruptions in highway tax management authorities and also minimizing the operating cost. Implementation of this system can be efficient way to keep a check on fuel wastage. This paper is mainly focused on a RFID based toll booth management system that is monitored using IOT. The database maintains all the data of user accounts and also their balance. Each vehicle owners possess a unique RFID (Radio Frequency Identification) based card mounted on their vehicles which stores their RFID number. The toll booth management system will monitor the cards scanned when a vehicle arrives at the toll booth. The system then connects to the database to check if the card is valid and if valid what is the balance. If the user balance is sufficient, the toll amount is deducted and card scanner system sends signal to the motor that the user has been billed and the billing details are sent to user via SMS. On receiving this signal, the system operates a motor to open the toll gate for that vehicle. The system is controlled by a microcontroller to achieve this purpose. The Wi-Fi module uses a Wi-Fi connection to connect the toll system to IoT cloud platform through which interacts to perform the online verification process, for the stolen vehicles and sends the information to the nearby police station and vehicle owner together. The system allows storing data to all the vehicles passed at particular time intervals for further reference and surveillance, and it also serves for the revenue generation process. This system thus automates the entire toll collection and monitoring process by using RFID plus IoT based system.

8. P. Keerthana, T. Sangeetha, N. Surekha. "Smart Tollgate Using IoT". International Journal of Science and Research (IJSR), 2018.

Abstract: Smart Toll Gate System, an advent of the common place global to the Automated worldwide. It reduces human intervention and the man-made mistakes. Bringing automation in Societies has a primary advantage that is: machine/hardware dependency is expanded (more dependable) and human dependency is decreased. This brings to an end that combining automation with daily life will make lifestyles less difficult and simpler. Nowadays almost all highways toll plazas are operated by hand, where an operator collects cash from the motive force and affords a receipt. Since this manner may be sluggish, we often come upon visitors jams on the toll plazas on busy highways. Toll collection in automated way can save money, energy, and man power. In this work suggest a low cost and efficient approach called Electronic Toll Collection. The use of RFID modules that mechanically collects the toll from shifting automobiles once they cross the toll plaza. This project addresses the problems confronted at toll plaza & also introduce identity gadget for vehicles towards which stolen and coincidence instances are registered the use of RFID.

9. Narayana Darapaneni; Parikshit Bangade; Amit Mane; Umang Maheshwari; Sushilkumar C Thorawade; Rushikesh Borse. "Traffic Monitoring and Analysis At Toll Plaza". IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)2021.

Abstract: This paper presents a solution to Traffic monitoring and analysis at Indian toll plazas. In some of the areas, the work is done on Vehicle detection and localization, vehicle registration detection, and character recognition and vehicle make classifier (Currently concentrated on Indian Cars only). Accurate detecting and localizing of objects in computer vision has always been a core problem, to the rescue of which Tensor Flow Object detection API comes with implementations provided by RCNN family (Basic RCNN, Fast RCNN, and Faster RCNN) and Single-shot detection models. Experimenting over a range of models from Faster RCNN with inception v2, SSD Mobile Net, and SSD inception v2 for vehicle and vehicle registration number detection problem we got higher accuracy with Faster RCNN with inception v2 for RMS Prop optimizer. For the detected vehicle registration number, we used an SVM-based multiclass classifier. A custom image dataset is used to train the model. The vehicle make classifier is implemented with VGG16 and the dataset was selected to contain only Indian cars.

10. Elmer R. Magsino; Ivan W. H. Ho. "An intelligent highway tollgate queue selector for improving server utilization and vehicle waiting time". IEEE Region 10 Symposium (TENSYMP) 2016 .

Abstract: On a highway setup, a vehicle will most probably use a tollgate server that has the shortest queue thinking that it is the fastest exit. In this paper, an intelligent highway tollgate queue selector using fuzzy logic is proposed and simulated in Matlab SimEvents. Its aim is to automatically select the most appropriate tollgate server for a vehicle to ensure the shortest waiting time while trying to balance the server's utilization. Two policies are considered in this study, namely: (1) Shortest Queue (SQ) and (2) Fuzzy Logic-Controlled Queue (FLCQ). Results indicate that the FLCQ policies reduce the average waiting time and queue length by approximately 50% of those obtained from the SQ policy while guaranteeing an equal utilization among available servers. These findings are valid for light and heavy, homogeneous and nonhomogeneous vehicle arrivals. To further improve the decision making of the fuzzy logic controller, traffic flow information collected by remote road-side units can be exploited to allow the control of various system parameters (e.g., service time) in advance.

CHAPTER 3

BLOCK DIAGRAM & EXPLANATION

3.1 BLOCK DIAGRAM:

The block diagram representation of our project, “Tollgate Traffic Monitor & Analyzer Using Raspberry Pi” is designed as shown below;

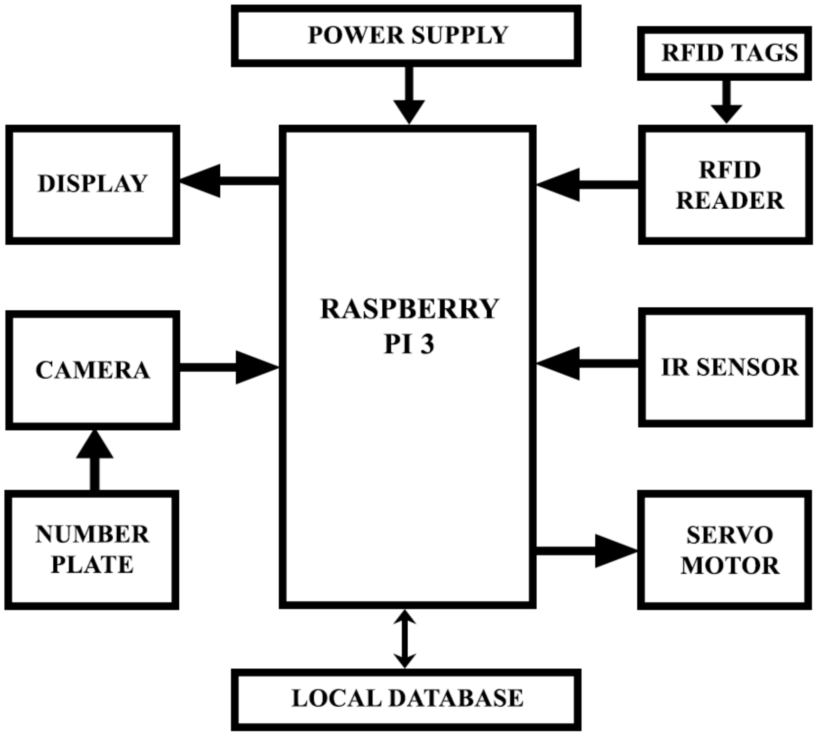


Figure 3.1: Block Diagram

As shown above the block diagram of the system consists of the following components; camera for detecting the number plate, local database for storing the required data, display for presenting the results, power supply for running the complete setup, RFID tags which will be available to the users to scan through the RFID reader present at each of the tollgates, IR sensor for detecting the vehicles at the tollgate and the servo motor for opening and closing the gate.

3.2 METHODOLOGY:

RFID Detection:

The first stage of the process is the detection of the RFID tag and the data on it. This happens when the RFID tag is swiped through the reader available at the tollgate. When RFID is scanned through the reader an active LOW output is sent to the Raspberry Pi also simultaneously fetching the data in the local database.

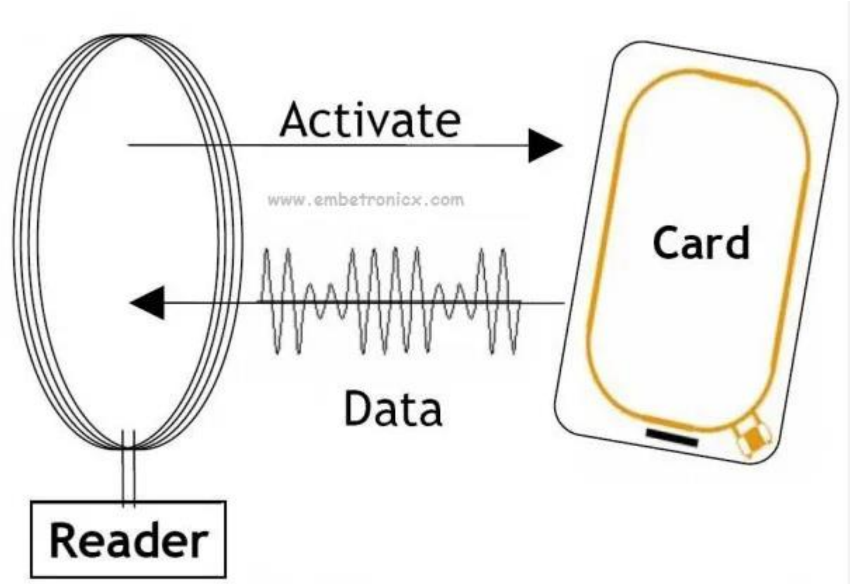


Figure 3.2: RFID Detection

Vehicle Detection:

The second stage of this system is detecting the vehicle when it arrives at the tollgate. This detection can be done by using the infrared sensor and the. It generates the active LOW output when it detects the vehicle. This sensor is connected to the GPIO pins of the Raspberry Pi then whenever it detects the vehicle the sensor generates the signal.

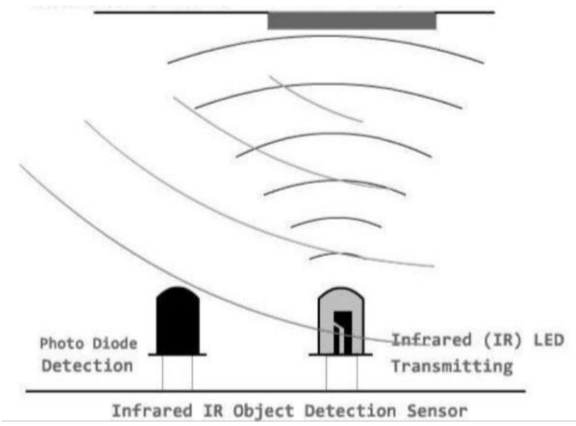


Figure 3.3: Vehicle Detection

Image Capturing:

After the successful detection of the vehicle arrival and scanning of the RFID the controller sends the signal to the camera module then it captures the image of the vehicle targeting its number plate and sends the captured image to process the image to extract the number plate in text format from the captured image.

Figure 3.4: Original Image Captured

Gray Scale Conversion:

The system will convert the captured color image into gray image for processing, because color information doesn't help us identify important edges or other feature. A gray scale image is simply one in which the only colors are shades of gray.

Figure 3.5: Grey Scale of Converted Image.

Bilateral Filtering:

A **bilateral filter** is a non-linear, edge-preserving, and noise-reducing smoothing filter for images. It replaces the intensity of each pixel with a weighted average of intensity values from nearby pixels. This weight can be based on a Gaussian distribution.

Figure 3.6: Bilateral Filter

Canny Edge Detection:

Canny edge detection is a technique to extract useful structural information from different vision objects and dramatically reduce the amount of data to be processed. It is seen that when the characters of the license number are written horizontally the vertical edges of each character appear at regular interval and they have a specific height. This appearance of vertical edge pattern is statistically seen to occur within the license plate, nowhere else within the natural scene of the image. The area of the region should not be less than specified threshold values.

Figure 3.7: Canny Edge Detection

Drawing Lines on Edges:

Contours are defined as the line joining all the points along the boundary of an image that are having the same intensity. Contours can be explained simply as a curve joining all the continuous points (along the boundary), having same color or intensity. The contours are a useful tool for shape analysis and object detection and recognition.

Figure 3.8: Contour Edge Drawing

Number Plate Detection:

Detect the number plate from the edges by using the closed edge finding procedure.

Figure 3.9: Number Plate Detection

Number Plate Extraction:

Extract that number plate from that detected area identifies vehicles by their number plates in which the number plate information is extracted from vehicle's image or from sequence of images without direct human intervention.

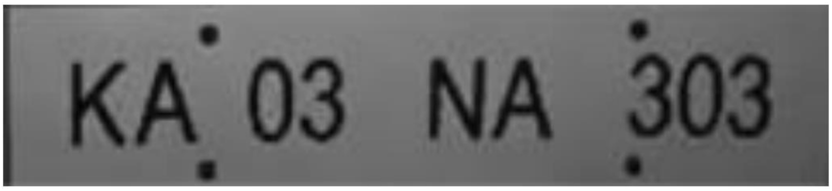


Figure: 3.10 Extract Number Plate

Number Extraction Using OCR:

OCR (optical character recognition) is the use of technology to distinguish printed or handwritten text characters inside digital images of physical documents, such as a scanned paper document. The basic process of OCR involves examining the text of a document and translating the characters into code that can be used for data processing. OCR is sometimes also referred to as text recognition.

KA 03 NA303 34

Opening tollgate automatically:

The data from the OCR, from the RFID, and from the data available on the database are compared and if the all the data at these three levels coincide then the servo motor (gate) is turned ON and OFF for the passage of the vehicle and if the data doesn't match at any level the servo motor is stationary and the buzzer is triggered ON.

When the authentication is successful the following text is displayed on the terminal i.e., "VALID NUMBER PLATE 1 AVAILABLE ON DB" and if the authentication fails due to wrong data the following text is displayed "NUMBER PLATE NOT AVAILABLE ON DB PAY MANUALLY"

CHAPTER 4

SCHEMATIC DIAGRAM & EXPLANATION

4.1 SCHEMATIC DIAGRAM:

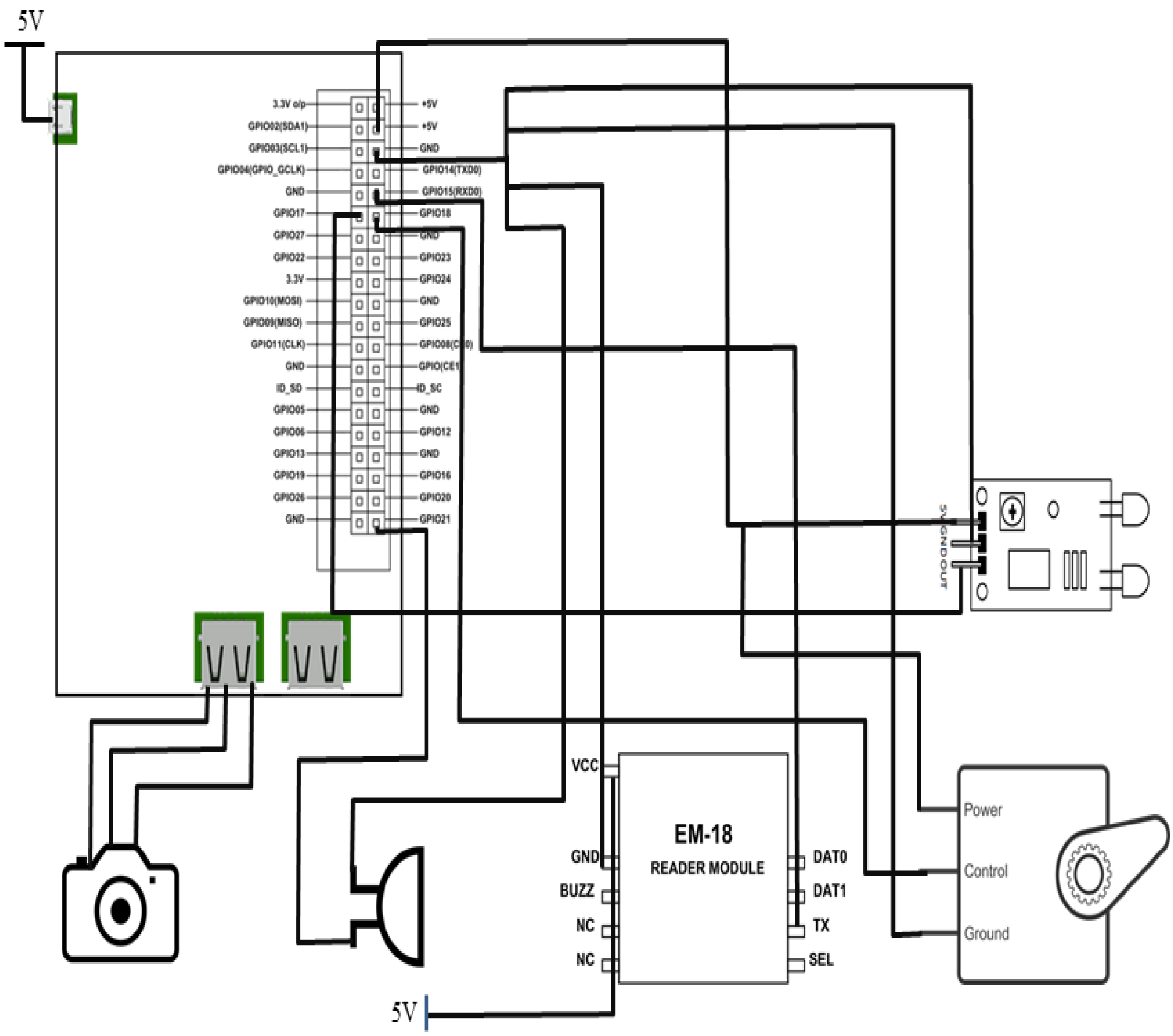


Figure 4.1: Schematic Diagram

4.2 EXPLANATION:

As shown in the figure above a total of six pins were utilized from the raspberry pi board for setting up the system and a USB port is utilized for integrating the camera for image processing. The following components where connected to the respective pins,

- PIN 4(5V POWER) : Parallel Connections Board(VCC)
- PIN 6(GND) : Parallel Connections Board(GND)
- PIN 10(RXD) : RFID TX Pin
- PIN 11(GPIO 17) : IR Sensor OUT Pin
- PIN 12(PCM_CLK) : Servo Motor CONTROL Pin
- PIN 40(PCM_DOUT) : Positive Terminal of Buzzer

Parallel Connections Board:

VCC:

This is used to power up the IR sensor and the servo motor in parallel connection.

GND:

This is used to ground the servo motor, EM 18 module, IR sensor, buzzer in parallel connection.

CHAPTER 5

HARDWARE COMPONENTS

5.1 POWER SUPPLY:

The power supply section is the section which provide +5V for the components to work. IC LM7805 is used for providing a constant power of +5V. The power supply block diagram can be seen below in figure 5.1 and the explanation of each of the block is given below.

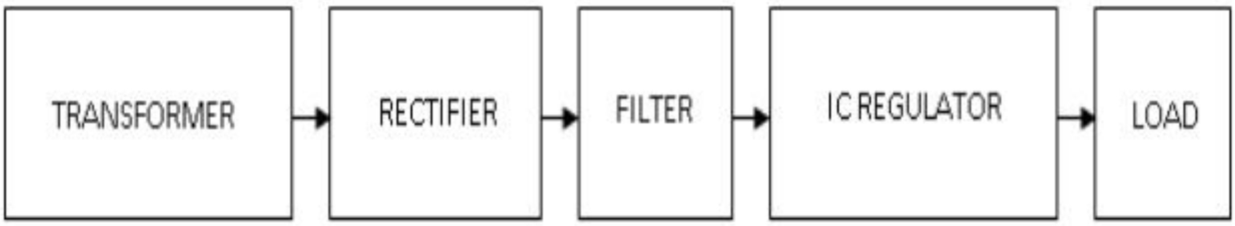


Figure 5.1: Block Diagram of Power Supply

Transformers:

The transformer convert AC electricity from one voltage to another with little loss of power. Transformers work only with AC and this is one of the reasons why mains electricity is AC. Step-up transformers increase voltage, step-down transformers reduce voltage. Most power supplies use a step-down transformer to reduce the dangerously high mains voltage (230V in India) to a safer low voltage. The input coil is called the primary and the output coil is called the secondary. There is no electrical connection between the two coils; instead they are linked by an alternating magnetic field created in the soft-iron core of the transformer. Transformers waste very little power so the power out is (almost) equal to the power in. Note that as voltage is stepped down current is stepped up. The transformer will step down the power supply voltage (0-230V) to (0- 6V) level.

Rectifier:

There are several ways of connecting diodes to make a rectifier to convert AC to DC. The bridge rectifier is the most important and it produces full-wave varying DC. A full-wave rectifier can also be made from just two diodes if a centre-tap transformer is used, but this method is rarely used now that diodes are cheaper. A single diode can be used as a rectifier but it only uses the positive (+) parts of the AC wave to produce half-wave varying DC.

Bridge Rectifier:

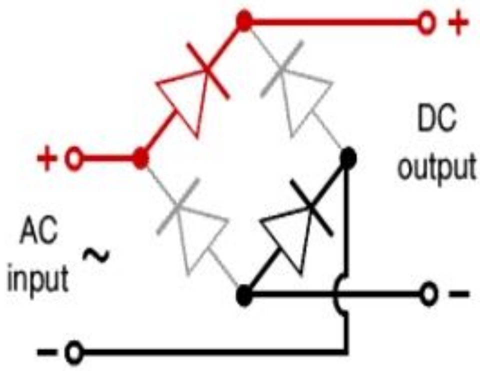


Figure 5.2: Bridge Rectifier

When four diodes are connected as shown in figure, the circuit is called as bridge rectifier. The input to the circuit is applied to the diagonally opposite corners of the network, and the output is taken from the remaining two corners. Let us assume that the transformer is working properly and there is a positive potential, at point A and a negative potential at point B. the positive potential at point A will forward bias D3 and reverse bias D4. The negative potential at point B will forward bias D1 and reverse D2. At this time D3 and D1 are forward biased and will allow current flow to pass through them; D4 and D2 are reverse biased and will block current flow. The advantages of using bridge rectifier are it will give peak voltage output as DC.

Voltage Regulator:

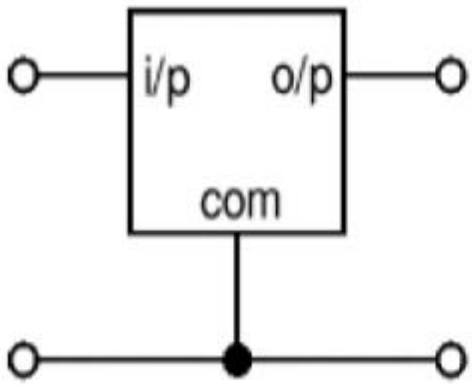


Figure 5.3: Regulator Block Diagram

Voltage regulators comprise a class of widely used ICs. Regulator IC units contain the circuitry for reference source, comparator amplifier, control device, and overload protection all in a single IC. IC units provide regulation of either a fixed positive voltage, a fixed negative voltage, or an adjustable set voltage. The regulators can be selected for operation with load currents from hundreds of milli amperes to tens of amperes, corresponding to power ratings from milli watts to Tens of watts.

A fixed three-terminal voltage regulator has an unregulated dc input voltage, Vi, applied to one input terminal, a regulated dc output voltage, Vo, from a second terminal, with the third terminal connected to ground.

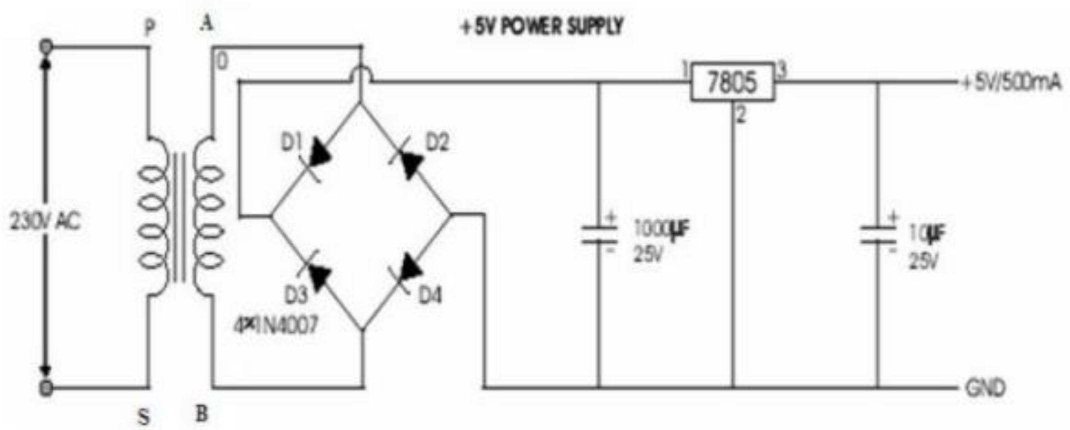


Figure 5.4: Circuit Diagram of Power Supply

5.2 RASPBERRY PI 3:

5.2.1 Introduction:

The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python. It’s capable of doing everything you’d expect a desktop computer to do, from browsing the internet and playing high-definition video, to making spreadsheets, word processing, and playing games.

Raspberry Pi has the ability to interact with the outside world, and has been used in a wide array of digital maker projects, from music machines and parent detectors to weather stations and tweeting birdhouses with infra-red cameras. We want to see the Raspberry Pi being used by kids all over the world to learn to program and understand how computers work.

Figure 5.5: Raspberry Pi Board

The raspberry pi board comprises a program memory (RAM), processor and graphics chip, CPU, GPU, Ethernet port, GPIO pins, Xbee socket, UART, power source connector. And various interfaces for other external devices. It also requires mass storage, for that we use an SD flash memory card.

So that raspberry pi board will boot from this SD card similarly as a PC boots up into windows from its hard disk. Essential hardware specifications of raspberry pi board mainly include SD card containing Linux OS, US keyboard, monitor, power supply and video 24 cables.

Optional hardware specifications include USB mouse, powered USB hub, case, internet connection, the Model A or B: USB Wi-Fi adaptor is used and internet connection to Model B is LAN cable.

5.2.2 <u>Raspberry Pi 3 Technical Specifications</u>:

Raspberry Pi 3 Model B	
SoC	Broadcom BCM2837
CPU	4× ARM Cortex-A53, 1.2GHz
GPU	Broadcom VideoCore IV
RAM	1GB LPDDR2 (900 MHz)
STORAGE	microSD
Ethernet	10/100 Ethernet
Wireless	2.4GHz 802.11n wireless
Video Output	HDMI
GPIO	40-pin header, populated
Ports	HDMI, 3.5mm analogue audio-video jack, 4× USB 2.0, Ethernet, Camera Serial Interface (CSI), Display Serial Interface

Table 5.2.2: Raspberry Pi 3 Technical Specifications

5.2.3 <u>Raspberry Pi 3 Pin Diagram</u>:

The pin diagram for the board is as shown below;

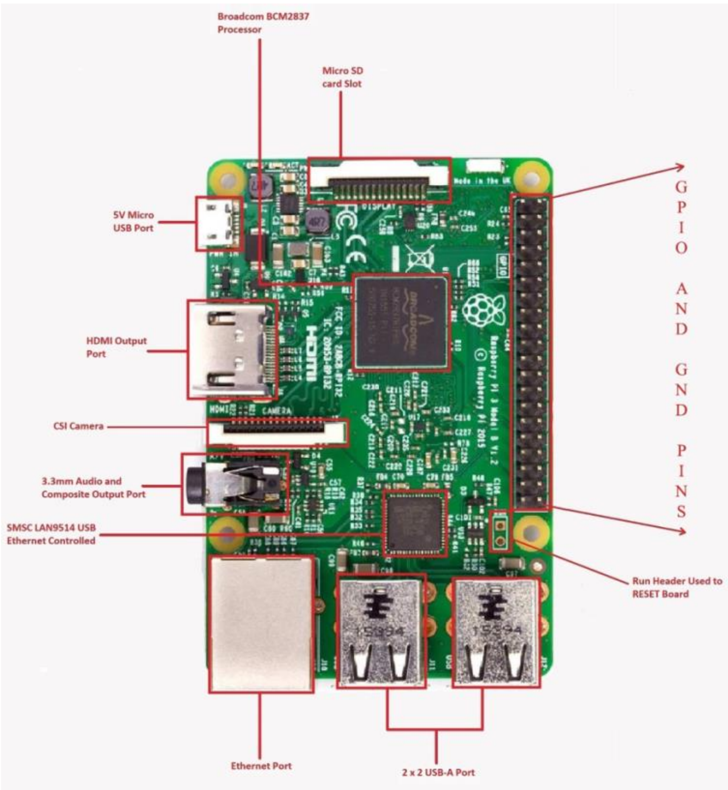


Figure 5.6: Raspberry Pi 3 Pin Diagram

The Raspberry Pi 3 **Model B** is a single-board computer developed by the Raspberry Pi Foundation. This board consists of a 1.2 Ghz 64-bit quad-core ARM processor and an 802.11n Wireless LAN, Bluetooth 4.1, and Bluetooth Low Energy. Like the previous version (the Pi 2) it consists of 1 GB of RAM, 4 USB ports, and full HDMI support. Raspberry pi 3 pin diagram is explained in detail here.

The Processor Used in Raspberry Pi 3:

Broadcom BCM2837:

It is a 1.2GHz 64bit ARM quad-core Cortex A53 processor, with 512 KiB shared L2 cache, dual-core Video Core IV GPU @ 400 MHz supporting OpenGL ES 2.0, hardware-accelerated OpenVG, and 1080p30 H.264 high-profile decode.

Micro SD Card Slot:

This is used for the purpose of storage and also for running the operating system for the raspberry pi 3.

HDMI Output Port:

This port is used for connecting the raspberry pi to an external monitor for monitoring and controlling purposes.

CSI Camera Slot:

This slot is used for connecting the dedicated camera to the raspberry pi also known as the ‘PI Camera’.

3.3mm Audio & Composite Output Port:

This used for audio output, for connecting to external speakers or headphones.

SMSC LAN9514 Microchip:

This chip is especially used in raspberry pi for controlling the LAN Ethernet port and the 4 USB ports present on the board.

GPIO, Power, GND & UART Pins:

A powerful feature of the Raspberry Pi is the row of GPIO (general-purpose input/output) pins along the extreme right edge of the board. Like every Raspberry Pi chipset, it consists of a 40-pin GPIO. A standard interface for connecting a single-board computer or microprocessor to other devices is through General-Purpose Input/Output (GPIO) pins. GPIO pins do not have a specific function and can be customized using the software.

The complete detailed pin diagram to understand these 40 pins is given below,

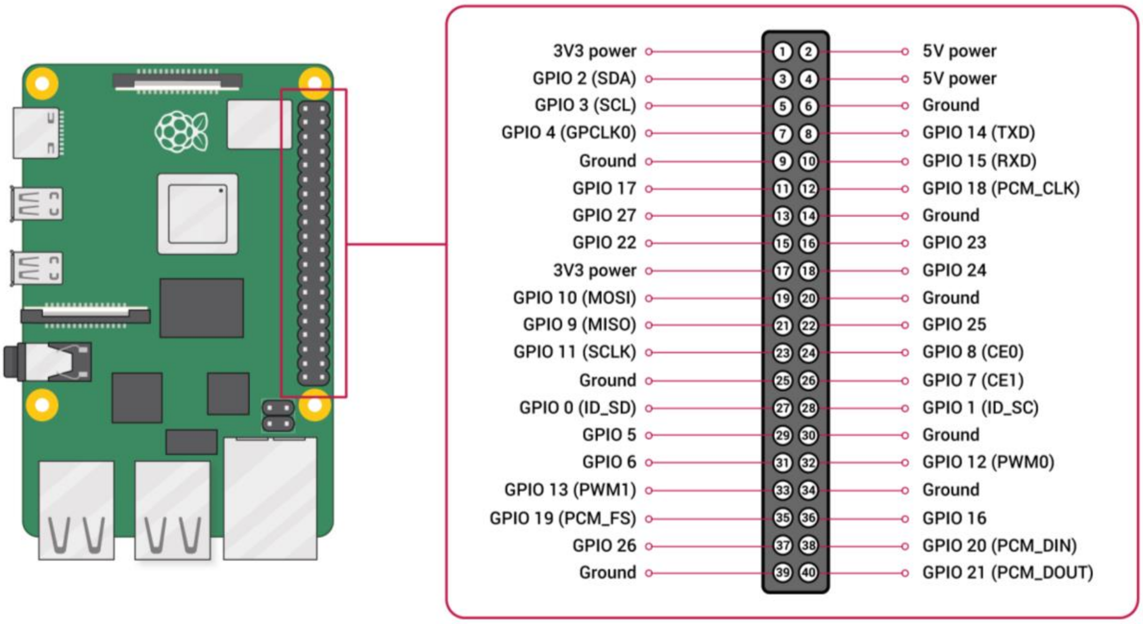


Figure 5.7: Raspberry Pi 3 GPIO Pin Diagram

5.2.4 <u>Features of Raspberry Pi 3</u>:

Some of the main features of raspberry pi 3 are,

- Quad-Core 1.2GHz Broadcom BCM2837 64bit CPU
- 1GB RAM
- BCM43438 wireless LAN and Bluetooth Low Energy (BLE) on board
- 100 Base Ethernet
- 40-pin extended GPIO
- 4 USB 2 ports
- 4 Pole 3.3mm stereo output and composite video port
- Full-size HDMI CSI (Camera Serial Interface) camera port for connecting a camera
- DSI (Display Serial Interface) display port for connecting a touchscreen display
- Micro SD port
- Micro USB power port (up to 2.5A)
- Size – 85 x 56 x 17 mm

5.3 EM-18 RFID READER MODULE:

RFID is an acronym for "**Radio-Frequency Identification**" and refers to a technology whereby digital data encoded in RFID tags or smart labels (defined below) are captured by a reader via radio waves. RFID is similar to barcoding in that data from a tag or label are captured by a device that stores the data in a database. RFID, however, has several advantages over systems that use barcode asset tracking software. The most notable is that RFID tag data can be read outside the line-of-sight, whereas barcodes must be aligned with an optical scanner.

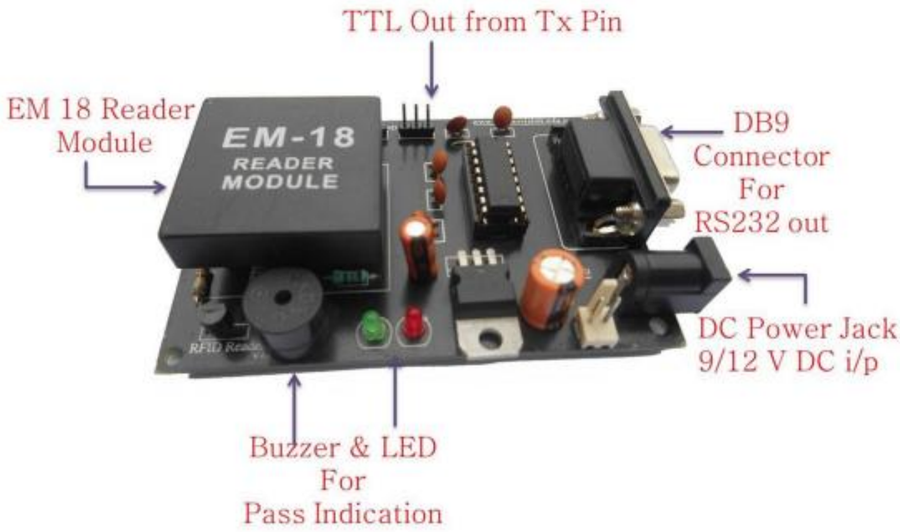


Figure 5.8: EM-18 RFID Module

EM18 is a RFID reader which is used to read RFID tags of frequency 125 kHz. The EM-18 RFID Reader module operating at 125 kHz is an inexpensive solution for your RFID based application. The Reader module comes with an on-chip antenna and can be powered up with a 5V power supply. Power-up the module and connect the transmit pin of the module to receive pin of your microcontroller. Show your card within the reading distance and the card number is thrown at the output.

5.3.1 Working:

RFID Reader has a transceiver that generates a radio signal and transmits it through the antenna. This signal itself is in the form of energy which is used to activate and power the tag.

When the RFID tag comes in a range of signals transmitted by the reader, the transponder in the tag is hit by this signal. A tag draws power from the electromagnetic field created by the reader. Then, the transponder converts that radio signal into usable power. After getting power, the transponder sends all the information it has stored in it, such as a unique ID to the RFID reader in the form of an RF signal. Then, the RFID reader puts this unique ID data in the form of a byte on a serial Tx (transmit) pin. This data can be used or accessed by PC or microcontroller serially using UART communication

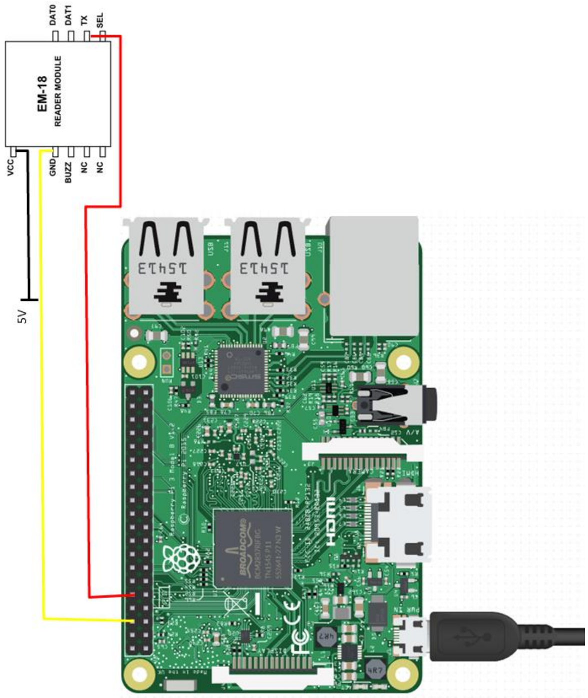


Figure 5.9: EM-18 Configured With Raspberry Pi 3

5.3.2 <u>Pin Diagram</u>:

The pin diagram for the EM-18 RFID Reader module is given below as shown in the figure;

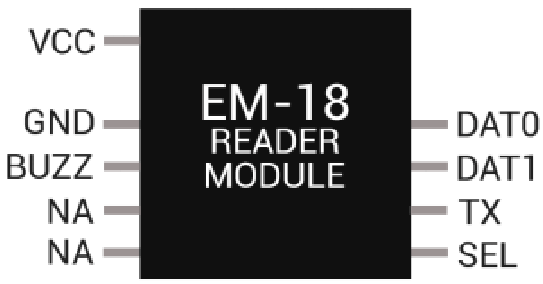


Figure 5.10: Pin Diagram of EM-18 Module

The each pin description is given below in table along with its number;

Pin Number	Description
VCC	Connect to the positive of power source.
GND	Connected to ground.
BUZZ	Connect to BUZZER if used
NC	No Connection
NC	No Connection
SEL	SEL=1 then o/p =RS232 SEL=0 then o/p=WEIGAND
TX	DATA send through TX using RS232 communication
DATA1	WEIGAND interface DATA HIGH pin
DATA0	WEIGAND interface DATA LOW pin

Table 5.3.2: Pin Description of EM-18 Reader Module

Technical Specifications:

- Supply Voltage : 4.6 V to 5.4 V DC
- Current : 65mAmp
- Card/Tag Format : EM4001 or Compatible
- Frequency : 125KHz
- Encoding : Manchester 64-bit, Modulus64
- Operating Temp. : 0 to 85 Celsius

5.3.3 Applications:

- e-Payment, e-Toll Road Pricing, e-Ticketing
- Access Control

5.4 RFID TAGS:

Radio frequency identification (RFID) tags are a substantial part of everyday life and the use of the technology continues to grow. Consider the label on a bottle of alcohol at the store meant to discourage stealing or the small device/sticker on a car windshield that allows fine-free toll road use: those are all RFID tags. While most RFID tags are used for merchandise or to track packages, they can also be used to help track pets or patients in hospital settings.

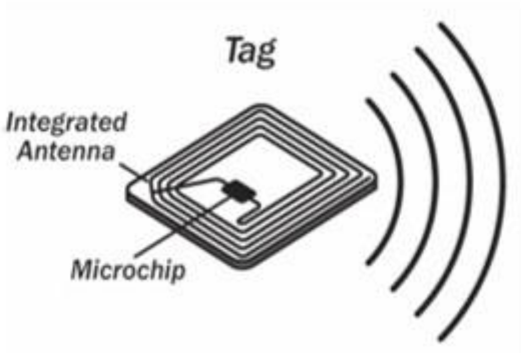


Figure 5.11: RFID Tags

RFID tags are a type of tracking system that uses radio frequency to search, identify, track, and communicate with items and people. Essentially, RFID tags are smart labels that can store a range of information from serial numbers, to a short description, and even pages of data. Some RFID tags include cryptographic security features for a high level of verification and authentication. RFID tags are usually identified by their radio frequencies: low frequency (LF), high frequency (HF), and ultra-high frequency (UHF).

There are two types of RFID tags — passive and active. Passive RFID tags are the most common; they do not require a direct line of sight to a reader but do have a much shorter read range, and are smaller in size and lightweight.

Passive RFID tags are ideal for:

Supply chain and inventory management, Asset and personnel tracking, Logistics, Brand protection and anti-counterfeiting, Real-time location systems (RTLS) using 3D orientation insensitive design, Gate and perimeter access control, Entertainment and travel. Apparel and retail

Active RFID tags are less common and require their own transmitter and power source. These tags tend to be bulkier, rugged, durable, and more expensive. An example of an active RFID tag is a beacon used for RTLS, constant medical monitoring, or theme park attendance.

5.4.1 Working:

An RFID tag works by transmitting and receiving information via an antenna and a microchip — also sometimes called an integrated circuit or IC. The microchip on an RFID reader is written with whatever information the user wants. When a RFID tag is scanned by a reader, the reader transmits energy to the tag which powers it enough for the chip and antenna to relay information back to the reader. The reader then transmits this information back to an RFID computer program for interpretation.

5.4.2 Circuit Diagram:

The integrated circuit diagram of the RFID tags is given below,

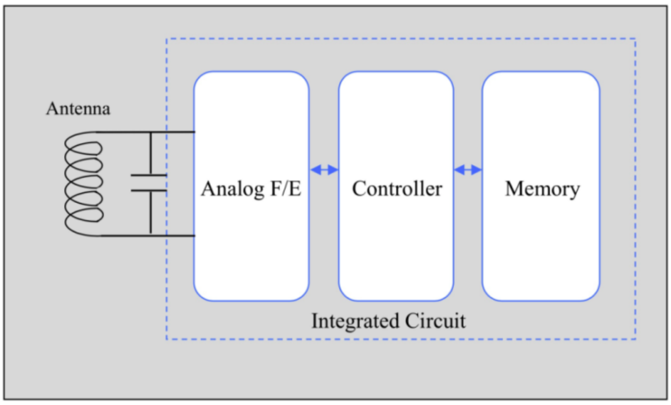


Figure 5.12: Circuit Diagram of RFID Tag

5.4.3 Applications:

- Pet and livestock tracking.
- Inventory management.
- Asset tracking and equipment tracking.
- Cargo and supply chain logistics.
- Customer service and loss control.
- Improved visibility and distribution in the supply chain.

5.5 INFRARED SENSOR:

A device which gives an output by detecting the changes in quantities or events can be defined as a sensor. Generally, sensors produce an electrical signal or optical output signal corresponding to the changes in the inputs. An infrared (IR) sensor is an electronic device that measures and detects infrared radiation in its surrounding environment. Infrared radiation was accidentally discovered by an astronomer named William Herchel in 1800. While measuring the temperature of each color of light (separated by a prism), he noticed that the temperature just beyond the red light was highest. IR is invisible to the human eye, as its wavelength is longer than that of visible light. Anything that emits heat gives off infrared radiation.

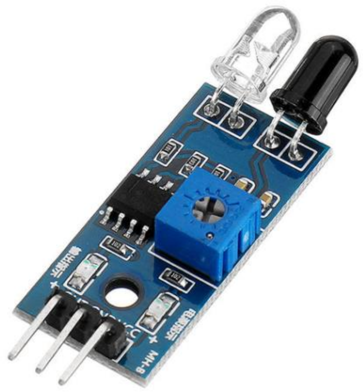


Figure 5.13: Infrared Sensor

There are two types of infrared sensors: active and passive.

Active Infrared Sensors both emit and detect infrared radiation. Active IR sensors have two parts: a light emitting diode (LED) and a receiver.

Passive Infrared (PIR) Sensors only detect infrared radiation and do not emit it from an LED.

5.5.1 Working:

The working principle of an infrared sensor is similar to the object detection sensor. This sensor includes an IR LED & an IR Photodiode, so by combining these two can be formed as a photo-coupler otherwise optocoupler.

IR LED is one kind of transmitter that emits IR radiations. This LED looks similar to a standard LED and the radiation which is generated by this is not visible to the human eye. Once it is used as the combination of an IR transmitter & receiver, then the receiver's wavelength must equal the transmitter. Here, the transmitter is IR LED whereas the receiver is IR photodiode. The infrared photodiode is responsive to the infrared light that is generated through an infrared LED. The resistance of photo-diode & the change in output voltage is in proportion to the infrared light obtained. This is the IR sensor's fundamental working principle.

Once the infrared transmitter generates emission, then it arrives at the object & some of the emission will reflect back toward the infrared receiver. The sensor output can be decided by the IR receiver depending on the intensity of the response.

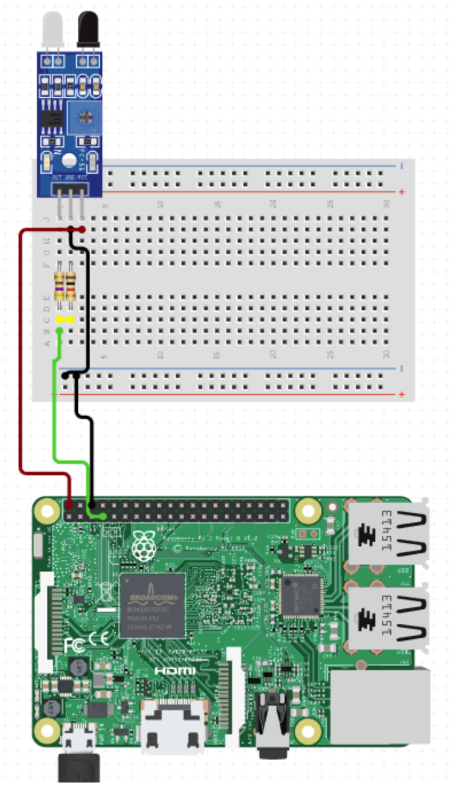


Figure 5.14: IR Sensor Configured With Raspberry Pi 3

5.5.2 <u>Pin Diagram</u>:

The pin diagram of the infrared sensor used in the project is given below,

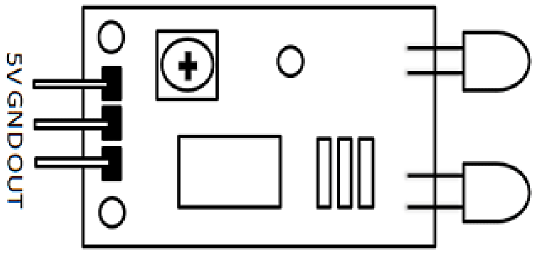


Figure 5.15: Pin Diagram of Infrared Sensor

The each pin description is given below in table along with its number.

Pin Name	Description
VCC	Power Supply +5v
GND	Power Supply Ground
OUTPUT	Active High Output

Table 5.5.2: Pin Description of Infrared Sensor

Technical Specifications:

- The operating voltage is 5VDC
- I/O pins – 3.3V & 5V
- Mounting hole
- The range is up to 20 centimetres
- The supply current is 20mA
- The range of sensing is adjustable
- Fixed ambient light sensor

5.5.3 Applications:

- Climatology
- Flame Monitors
- Gas detectors
- Anaesthesiology testing
- Petroleum exploration
- Rail safety
- Gas Analysers

5.6 SG 90 MICRO SERVO MOTOR:

A **servo motor** is a type of motor that can rotate with great precision. Normally this type of motor consists of a control circuit that provides feedback on the current position of the motor shaft; this feedback allows the servo motors to rotate with great precision. If you want to rotate an object at some specific angles or distance, then you use a servo motor. It is just made up of a simple motor which runs through a **servo mechanism**. A servo motor usually comes with a gear arrangement that allows us to get a very high torque servo motor in small and lightweight packages. Due to these features, they are being used in many applications like toy car, RC helicopters and planes, Robotics, etc.

Figure 5.16: Servo Motor

5.6.1 Working:

It is a closed-loop system where it uses a positive feedback system to control motion and the final position of the shaft. Here the device is controlled by a feedback signal generated by comparing output signal and reference input signal.

Here reference input signal is compared to the reference output signal and the third signal is produced by the feedback system. And this third signal acts as an input signal to the control the device. This signal is present as long as the feedback signal is generated or there is a difference between the reference input signal and reference output signal. So the main task of servomechanism is to maintain the output of a system at the desired value at presence of noises.

Servos are controlled by sending an electrical pulse of variable width, or **pulse width modulation (PWM)**, through the control wire. The rotation can be explained by the figure below.

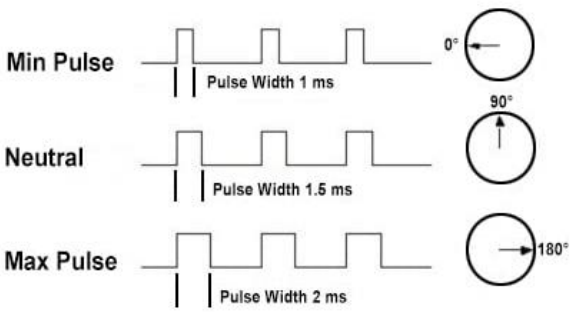


Figure 5.17: Variable Pulse Width Control Servo Position

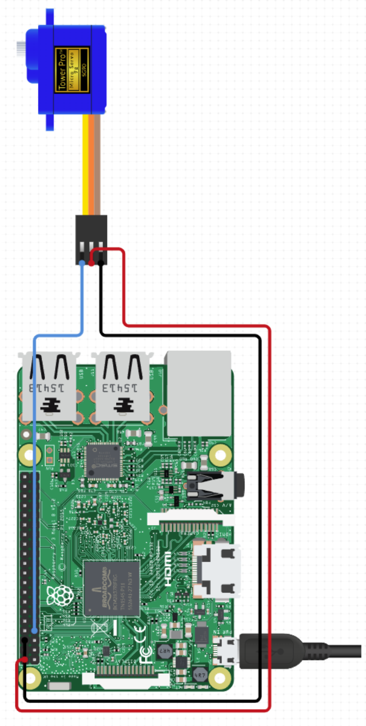


Figure 5.18: Servo Motor Configured With Raspberry Pi 3

5.6.2 Pin Diagram:

The pin diagram of the servo motor used in the project is given below,

Figure 5.19: Pin Diagram of Servo Motor

The each pin description is given below in table along with its number.

Pin Number	Signal Name	Color Scheme 1 (Futaba)	Color Scheme 2 (JR)	Color Scheme 3 (Hitec)
1	Ground	Black	Brown	Black
2	Power Supply	Red	Red	Red or Brown
3	Control Signal	White	Orange	Yellow or White

Table 5.6.2: Servo Connection Colour Coding

Technical Specifications:

- It comes with a 3 horns (arms) and hardware.
- Weight: 9 g
- Dimension: 22.2 x 11.8 x 31 mm approx.
- Stall torque: 1.8 kgf·cm
- Operating speed: 0.1 s/60 degree
- Operating Voltage: 4.8V
- Temperature Range: 0-55 Celsius

5.6.3 Applications:

- Servo motor applications are also commonly seen in remote controlled toy cars for controlling direction of motion
- It is also very commonly used as the motor which moves the tray of a CD or DVD player.

5.7 WEB CAMERA:

A webcam is a video camera that feeds or streams an image or video in real time to or through a computer network, such as the Internet. Webcams are typically small cameras that sit on a desk, attach to a user's monitor, or are built into the hardware. Webcams can be used during a video chat session involving two or more people, with conversations that include live audio and video.

5.20: Web Camera

The term "webcam" (a clipped compound) may also be used in its original sense of a video camera connected to the Web continuously for an indefinite time, rather than for a particular session, generally supplying a view for anyone who visits its web page over the Internet. Some of them, for example, those used as online traffic cameras, are expensive, rugged professional video cameras.

5.7.1 Usage:

Here, we are using the web camera for capturing the vehicle's number plate so that the capture image can be processed to convert it into the next for comparing it with the number plates available on the database.

5.7.2 Applications:

- Video Monitoring
- Video Calling, Video Conferencing, Video Security, Video Clips & Stills
- Input Control Devices, Laser Beam Profiling

5.8 BUZZER:

An audio signaling device like a beeper or buzzer may be electromechanical or piezoelectric or mechanical type. The main function of this is to convert the signal from audio to sound. Generally, it is powered through DC voltage and used in timers, alarm devices, printers, alarms, computers, etc. Based on the various designs, it can generate different sounds like alarm, music, bell & siren.

5.8.1 Usage:

When an alternating voltage is applied to the piezoceramic element, the element extends and shrinks diametrically. This characteristic of piezoelectric material is utilized to make the ceramic plate vibrate rapidly to generate sound waves. This is used for alarm in case of number plate breaches or identification unregistered vehicles

Figure 5.21: Buzzer

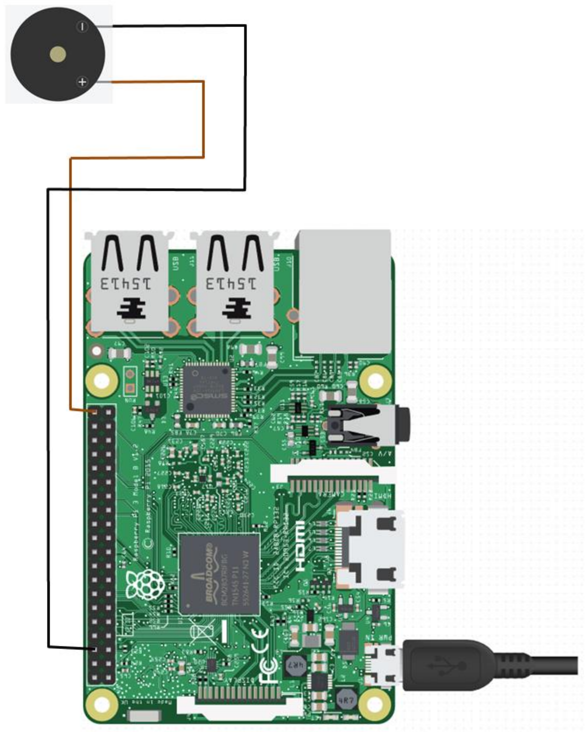


Figure 5.22: Buzzer Configured With Raspberry Pi 3

5.8.2 Applications:

The applications of the buzzer include the following.

- Communication Devices
- Electronics used in Automobiles
- Alarm Circuits
- Portable Devices
- Security Systems
- Timers
- Household Appliances
- Electronic Metronomes
- Sporting Events
- Annunciator Panels
- Game Shows

CHAPTER 6

SOFTWARE SPECIFICATIONS

6.1 RASPBIAN OPERATING SYSTEM:

Raspberry Pi OS (formerly Raspbian) is a debian-based operating system for Raspberry Pi. Since 2013, it has been officially provided by the Raspberry Pi Foundation as the primary operating system for the Raspberry Pi family of compact single-board computers.

Raspberry Pi OS was first developed by Mike Thompson and Peter Green as Raspbian, an independent and unofficial port of Debian to the Raspberry Pi. The first build was released on July 15, 2012. As the Raspberry Pi had no officially provided operating system at the time, the Raspberry Pi Foundation decided to build off of the work done by the Raspbian project and began producing and releasing their own version of the software.

Figure 6.1: Raspbian OS Logo

Raspberry Pi OS is highly optimized for the Raspberry Pi line of compact single-board computers with ARM CPUs. It runs on every Raspberry Pi except the Pico microcontroller.

User interface

Raspberry Pi OS has a desktop environment, PIXEL, based on LXDE, which looks similar to many common desktops, such as macOS and Microsoft Windows. The desktop has a background image. A menu bar is positioned at the top and contains an application menu and shortcuts to a web browser (Chromium), file manager, and terminal. The other end of the menu bar shows a Bluetooth menu, Wi-Fi menu, volume control, and clock.

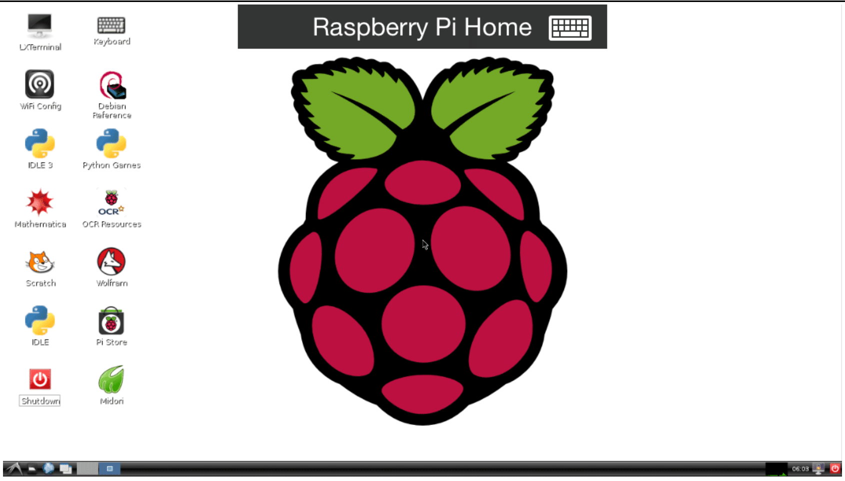


Figure 6.2: Raspberry Pi Home Screenshot

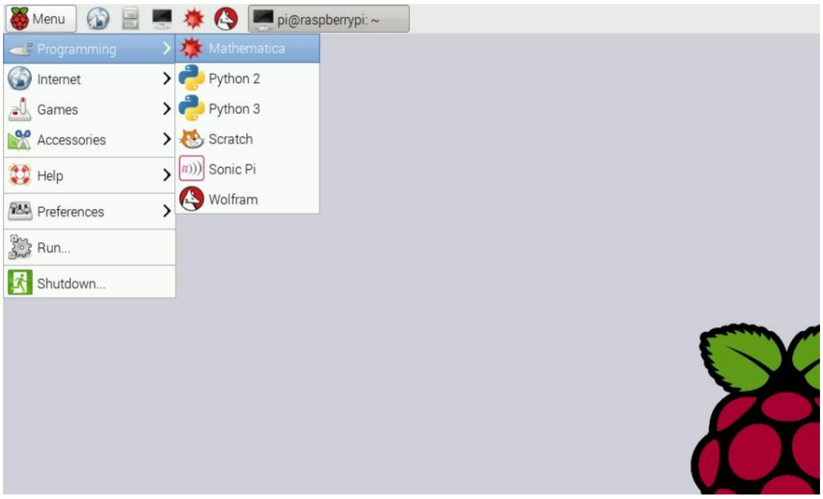


Figure 6.3: Raspberry Pi Screenshot with Available Programming Languages

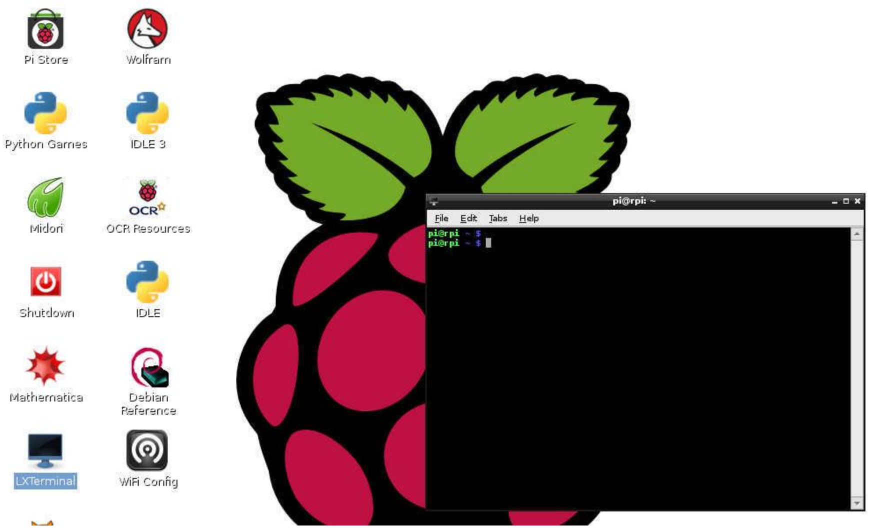


Figure 6.4: Raspberry Pi Screenshot With Terminal

Versions:

Raspberry Pi OS has three installation versions:

- Raspberry Pi OS Lite (32-bit & 64-bit)
- Raspberry Pi OS with desktop (32-bit & 64-bit)
- Raspberry Pi OS with desktop and recommended software (32-bit)

Raspberry Pi OS also has two legacy versions:

- Raspberry Pi OS Lite (Legacy) (32-bit)
- Raspberry Pi OS (Legacy) with desktop (32-bit)

Raspberry Pi OS Lite is the smallest version and doesn't include a desktop environment. Raspberry Pi OS with desktop includes the Pixel desktop environment.

6.2 PYTHON IDE:

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- **Python is Interpreted, Interactive, Object-Oriented , Beginner's Language**

Figure 6.5: Python Logo

History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

Python's features include −

- **Easy-to-learn**
- **Easy-to-read**
- **A broad standard library**
- **Interactive Mode**
- **Portable**
- **GUI Programming** .

Apart from the above-mentioned features, Python has a big list of good features, few are listed below −

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- It supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

Python is available on a wide variety of platforms including Linux and Mac OS X. Let's understand how to set up our Python environment.

Getting Python

The most up-to-date and current source code, binaries, documentation, news, etc., is available on the official website of Python https://www.python.org/

You can download Python documentation from https://www.python.org/doc/. The documentation is available in HTML, PDF, and PostScript formats.

Installing Python

Python distribution is available for a wide variety of platforms. You need to download only the binary code applicable for your platform and install Python..

Python Environment Variables

Here are important environment variables, which can be recognized by Python −

Sr.No.	Variable & Description
1	**PYTHONPATH** It has a role similar to PATH. This variable tells the Python interpreter where to locate the module files imported into a program. It should include the Python source library directory and the directories containing Python source code. PYTHONPATH is sometimes preset by the Python installer.
2	**PYTHONSTARTUP** It contains the path of an initialization file containing Python source code. It is executed every time you start the interpreter. It is named as .pythonrc.py in Unix and it contains commands that load utilities or modify PYTHONPATH.
3	**PYTHONCASEOK** It is used in Windows to instruct Python to find the first case-insensitive match in an import statement. Set this variable to any value to activate it.
4	**PYTHONHOME** It is an alternative module search path. It is usually embedded in the PYTHONSTARTUP or PYTHONPATH directories to make switching module libraries easy.

Table 6.2: Python Built-in Environment Variables

Here is the list of all the available command line options –

Sr.No.	Option & Description
1	**-d** It provides debug output.
2	**-O** It generates optimized bytecode (resulting in .pyo files).
3	**-S** Do not run import site to look for Python paths on startup.
4	**-v** verbose output (detailed trace on import statements).
5	**-X** disable class-based built-in exceptions (just use strings); obsolete starting with version 1.6.
6	**-c cmd** run Python script sent in as cmd string

Table 6.3: Python Command Line Options

Integrated Development Environment

You can run Python from a Graphical User Interface (GUI) environment as well, if you have a GUI application on your system that supports Python.

- **Unix** − IDLE is the very first Unix IDE for Python.
- **Windows** − PythonWin is the first Windows interface for Python and is an IDE with a GUI.
- **Macintosh** − The Macintosh version of Python along with the IDLE IDE is available from the main website, downloadable as either MacBinary or BinHex'd files.

6.3 OPENCV:

Computer Vision

Computer vision is a process by which we can understand the images and videos how they are stored and how we can manipulate and retrieve data from them. Computer Vision is the base or mostly used for Artificial Intelligence. Computer-Vision is playing a major role in self-driving cars, robotics as well as in photo correction apps.

OpenCV

OpenCV is the huge open-source library for the computer vision, machine learning, and image processing and now it plays a major role in real-time operation which is very important in today’s systems. By using it, one can process images and videos to identify objects, faces, or even handwriting of a human. When it integrated with various libraries, such as NumPy, python is capable of processing the OpenCV array structure for analysis. To Identify image pattern and its various features we use vector space and perform mathematical operations on these features.

OpenCV Functionality

- Image/video I/O, processing, display (core, imgproc, highgui)
- Object/feature detection (objdetect, features2d, nonfree)
- Geometry-based monocular or stereo computer vision (calib3d, stitching, videostab)
- Machine learning & clustering (ml, flann)

Applications of OpenCV:

There are lots of applications which are solved using OpenCV, some of them are listed below

- Face recognition
- Automated inspection and surveillance
- Vehicle counting on highways along with their speeds
- Video/image search and retrieval
- Robot and driver-less car navigation and control
- Object recognition
- Medical image analysis
- Movies – 3D structure from motion

How Does A Computer Read An Image?

Consider the below image:

Figure 6.6: Sample Image For OpenCV Process

We are humans we can easily make it out that is the image of a person who is me. But if we ask computer "is it my photo?" The computer can't say anything because the computer is not figuring out it all on its own. The computer reads any image as a range of values between 0 and 255. For any colour image, there are 3 primary channels -red, green and blue.

Image-Processing

Image processing is a method to perform some operations on an image, in order to get an enhanced image and or to extract some useful information from it.

Image processing basically includes the following three steps:

1. Importing the image
2. Analysing and manipulating the image
3. Output in which result can be altered image or report that is based on image analysis

6.4 TESSERACT OCR:

Python-tesseract is an optical character recognition (OCR) tool for python. That is, it will recognize and “read” the text embedded in images.

Python-tesseract is a wrapper for Google’s Tesseract-OCR Engine. It is also useful as a stand-alone invocation script to tesseract, as it can read all image types supported by the Pillow and Leptonica imaging libraries, including jpeg, png, gif, bmp, tiff, and others. Additionally, if used as a script, Python-tesseract will print the recognized text instead of writing it to a file.

INSTALLATION

Installing via pip:

Check the pytesseract package page for more information.

```
pip install pytesseract
```

Or if you have git installed:

```
pip install -U git+https://github.com/madmaze/pytesseract.git
```

Installing from source:

```
git clone https://github.com/madmaze/pytesseract.git

cd pytesseract && pip install -U .
```

Install with conda (via conda-forge):

```
conda install -c conda-forge pytesseract
```

TESTING

To run this project's test suite, install and run `tox`. Ensure that you have `tesseract` installed and in your PATH.

```
pip install tox

tox
```

Functions

- **get_languages** Returns all currently supported languages by Tesseract OCR.
- **get_tesseract_version** Returns the Tesseract version installed in the system.
- **image_to_string** Returns unmodified output as string from Tesseract OCR processing
- **image_to_boxes** Returns result containing recognized characters and their box boundaries
- **image_to_data** Returns result containing box boundaries, confidences, and other information. Requires Tesseract 3.05+. For more information, please check the Tesseract TSV documentation
- **image_to_osd** Returns result containing information about orientation and script detection.
- **run_and_get_output** Returns the raw output from Tesseract OCR. Gives a bit more control over the parameters that are sent to tesseract.

CHAPTER 7

RESULTS

7.1 SIMULATION WORK:

The working of the Toll gate traffic monitor and analyzer starts when the terminal window is invoked as soon as it is invoked the raspberry pi activates the following components RFID Reader, IR Sensor, Camera. First the RFID tag needs to be swiped near the reader as soon as this is done the IR sensor and the camera are activated and once this happens whenever the IR sensor detects the vehicle it captures the image of the vehicle targeting the number plate then the following steps are followed,

Step 1: Capture the image

Step 2: Convert the color image into grey image

Step 3: Smoothing the image by filter

Step 4: Finding the edges using canny edge detection

Step 5: Drawing the edges using the contour line method

Step 6: Finding the number plate shaped rectangle using find contours method

Step 7: Crops the image of the number plate

Step 8: Number extraction from Optical Character Recognition

The resulting vehicle number is then compared with the available database of all the vehicles so as to come up with information about the vehicle and decide to prompt the passage of the vehicle by opening the gate through servo motor or request the traveler to pay manually incase if the vehicle is not registered in the database. The hardware model consists of proximity sensor to detect the presence of

Vehicle, a web camera to capture the image, motors to open/close the road barriers of toll plaza on which this algorithm is executed, Display & a Raspberry pi for controlling all the components of hardware model. As the vehicle arrives at toll plaza, the inductive proximity sensor detects the vehicle and gives a signal to the Raspberry pi. The camera connected to the Raspberry pi captures the image of front view of the vehicle & applies number recognition algorithm on the image to recognize the vehicle's license number. All the information such as time, date, plate number & toll amount is stored in database to maintain the record. PC then sends the signal to Raspberry pi. Then the road barrier is opened for a time by driving motors

& “Please Move Ahead, Valid Number Plate Available In Database” is displayed on the LCD to guide the vehicles and if the number plate is not available on the database then “Number Plate Not Available On DB Pay Manually” is displayed on the LCD and the buzzer is turned ON.

7.2 DESIGN SCREENSHOTS:

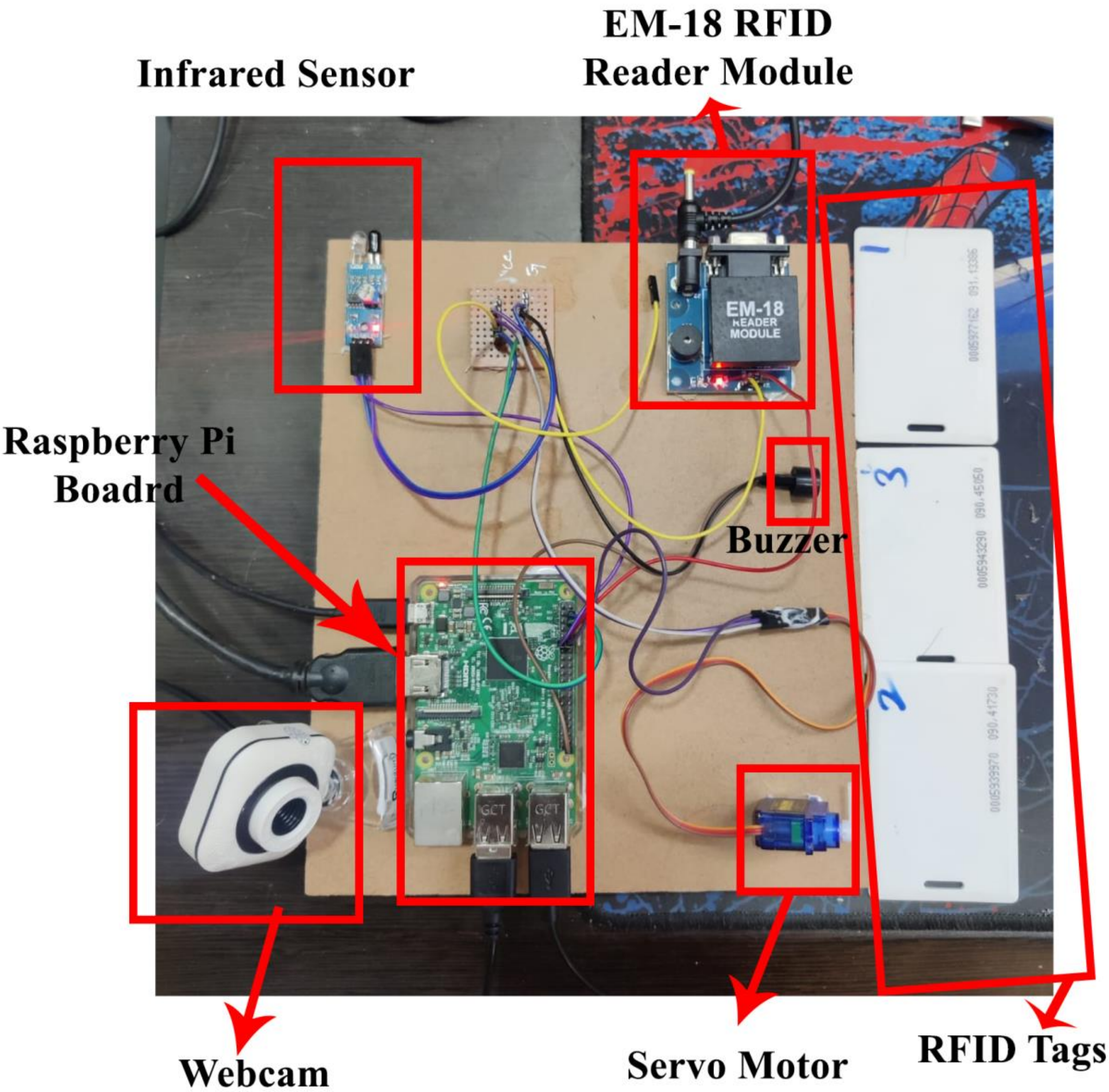


Figure 7.1: Complete System Design

The above figure 7.2 is the complete kit with all the modules and devices integrated to the raspberry pi board with RFID tags to right side of the board beside servo motor

7.3 OUPUT SCREENSHOTS:

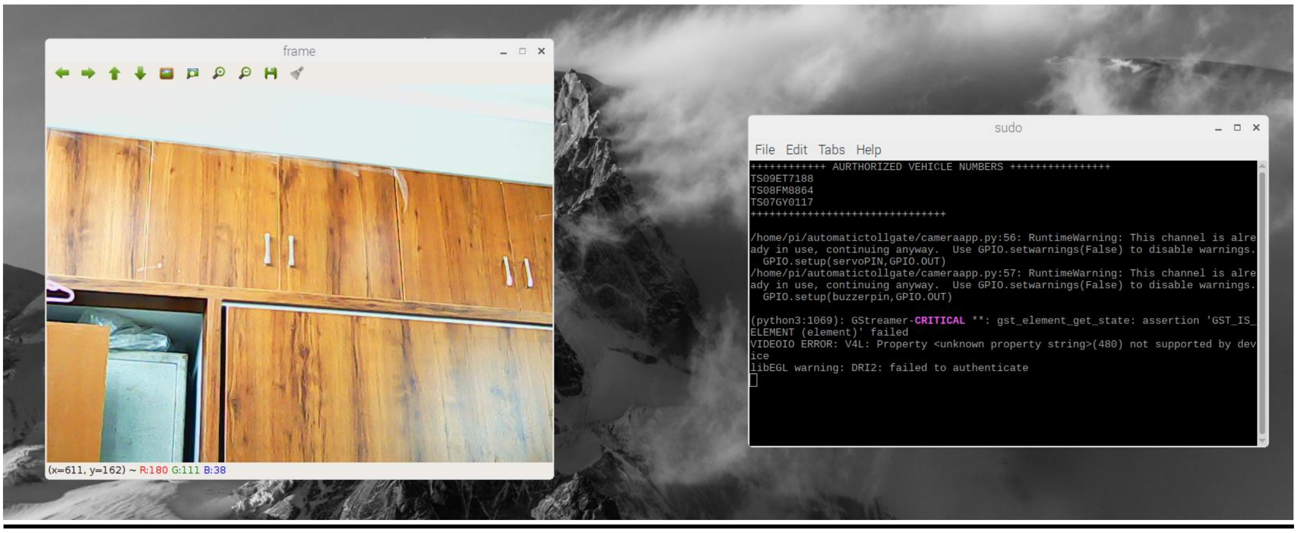


Figure 7.2: Startup Interface

The above screen with the camera and terminal appears immediately when the initialization file is invoked. The left side window is interface to display what the camera is capturing; this camera is used to capture the vehicle number plate and the window to the right side of the screen displays all the information related to the vehicle and authentication when the vehicles tries to pass through the tollgate.

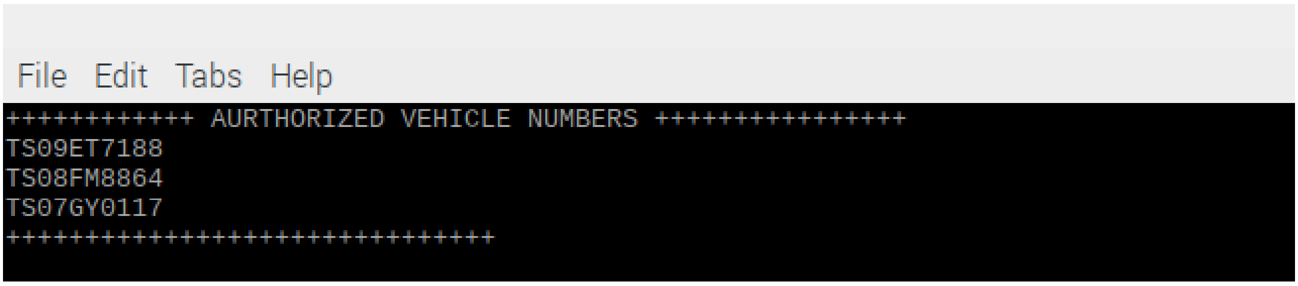


Figure 7.3: Available Data on the Database

Figure 7.4: RFID Scanned through the Reader

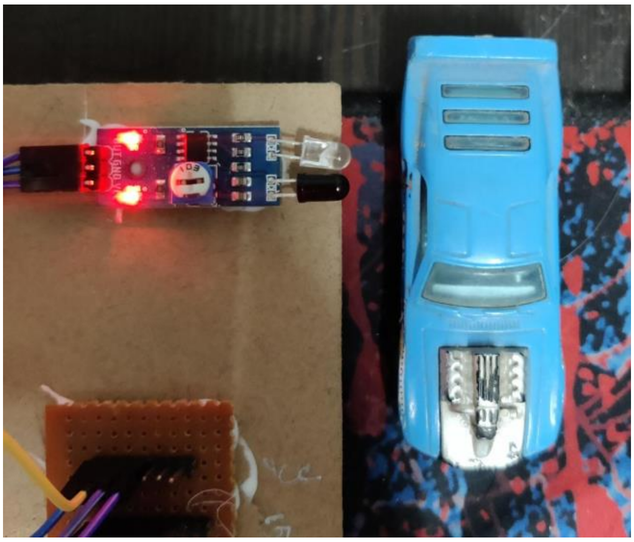


Figure 7.5: Vehicle Detected by IR Sensor

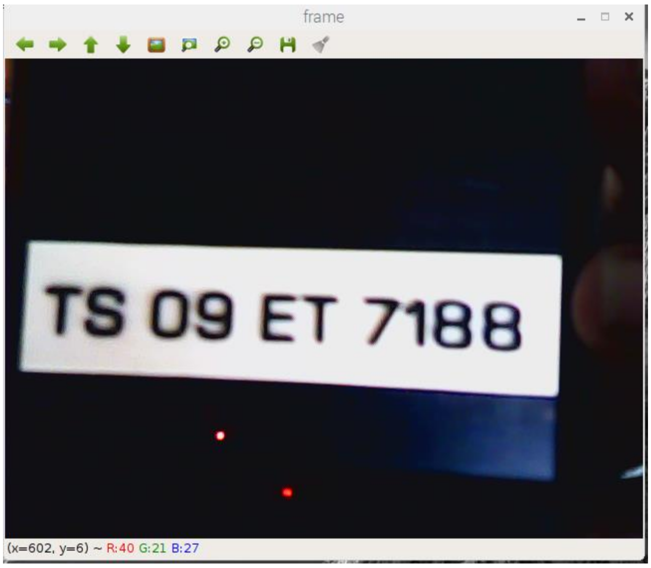


Figure 7.6: Number Plate Captured by Camera

As the data read by the RFID matches with the data read by the camera and also matches with the same data that is available on the database the servo motor turns ON and OFF for the passage of the vehicle and the following information is displayed on the terminal screen.

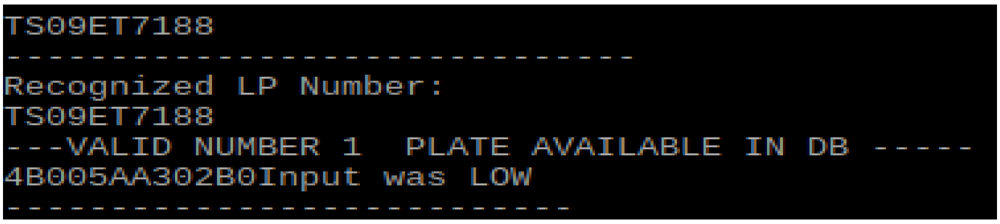


Figure 7.7: Relevant Information Displayed on the Screen


Figure 7.8: Servo Motor Turned ON & OFF

Similarly if the data on database or the data on RFID or the data processed from the camera doesn't match then the servo motor doesn't turn ON, buzzer is triggered ON, and the following information is displayed on the terminal screen.

Figure 7.9: RFID Scanned through the Reader

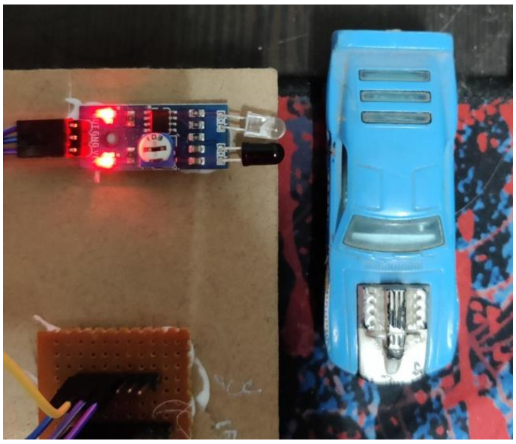


Figure 7.11: Number Plate Captured by Camera

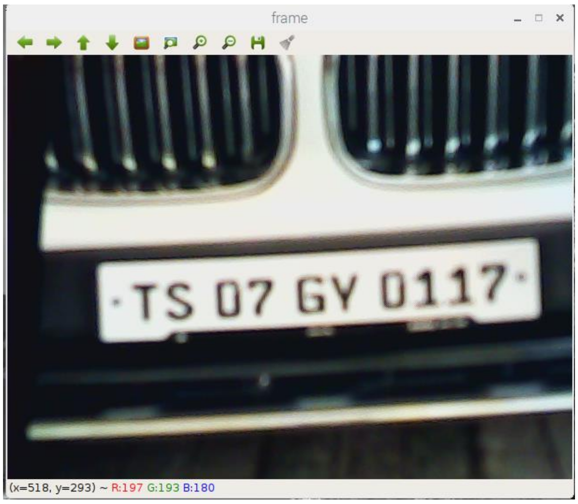


Figure 7.10: Vehicle Detected by IR Sensor

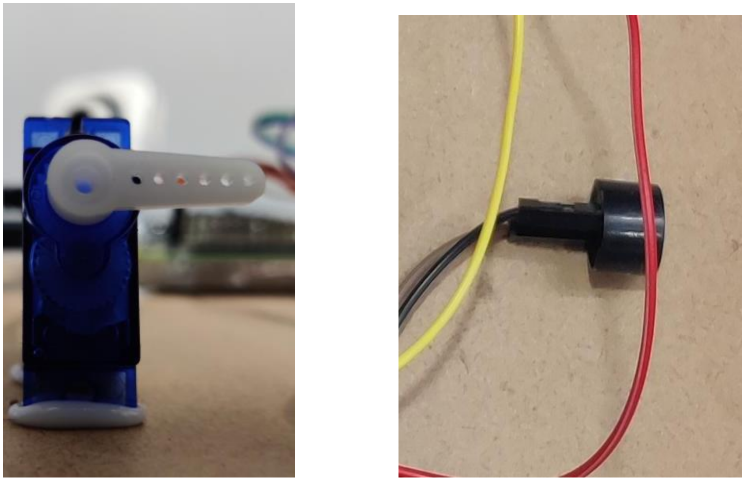
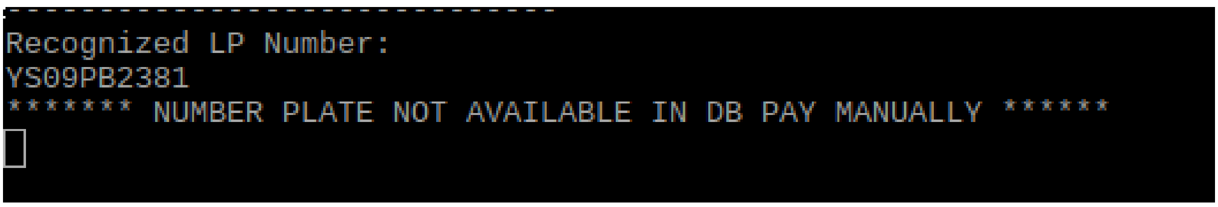


Figure 7.12: Servo Motor is Stationary and Buzzer is Trigger ON.



```
--------------------------------
Recognized LP Number:
YS09PB2381
******* NUMBER PLATE NOT AVAILABLE IN DB PAY MANUALLY ******
```

Figure 7.13: Relevant Information Displayed on the Screen

CHAPTER 8
ADVANTAGES & APPLICATIONS

ADVANTAGES:

- Ease of payment no need to carry cash for the toll transactions which ultimately saves time as the process is fairly simplified.
- Near non-stop movement of vehicles leading to lower fuel cost.
- SMS alerts are sent to the holder for toll transactions, low balance, etc.
- Environmental benefit like reduced air pollution and less use of paper is beneficial.
- Social benefit like less time taking toll payment is hassle-free and analytics for better highway management.
- Economic benefit like the reduced effort in management at the toll plaza and the reduced effort in monitoring centrally is also beneficial.
- Vehicle number plate breaches or theft vehicles can be identified easily.
- The use of manpower is reduced on a large scale as everything is automated.

APPLICATIONS:

- It can be used for parking ticket fee collection in shopping malls, movie theatres, etc.
- It can be included in public properties like airports, railway stations, etc.
- With the further extension of the technology with GPS-based systems, complete toll gates can be eliminated.
- Implementing the complete system with high level image processing will make the toll collection more hassle free as there will be no additional data except the vehicle number plate details.

CHAPTER 9

CONCLUSION & FUTURE SCOPE

The proposed system is a prototype of the toll system using RFID which supports the vehicle to pass across the tollgate with ease and hassle free movement. The system aims in making the toll booths free of long queue and also eliminates the clumsy manual work. The proposed solution also sends the user transaction details and vehicle details to the local database; this makes the transaction open which is free from all corruptions. A set of all the registered vehicles are already present on the database hence it is impossible to bypass the toll without authentication. The information such as the vehicle number, tollgate number and the timings at which the vehicle pass are recorded in the database which helps in analysis for the future unforeseen reason. The easy availability of real time data figures over the database platform will also help the government to collect road tax properly. The combination of RFID system and the image processing through the camera appeared effective to detect stolen vehicles, and prevent their trespassing across states, districts, borders. Thus the existing toll management system can be completely implemented with this technology so that the transaction will be digital without any burden to carry the cash and also the journey becomes ease at the toll. Since most of the fuel is wasted by keeping the engine idle in the toll gates due to traffic congestion, so indirectly it also helps in saving the fuel which is the most precious nonrenewable resource available.

FUTURE SCOPE:

The technology that we are implementing can be used across various applications in the future with minute modifications;

- It can be used for parking ticket fee collection in shopping malls, movie theatres, etc.
- It can be included in public properties like airports, railway stations, etc.

- With the further extension of the technology with GPS-based systems, complete toll gates can be eliminated.
- Implementing the complete system with high level image processing will make the toll collection more hassle free as there will be no additional data except the vehicle number plate details.

REFERENCES

[1] https://data.gov.in/catalog/total-numberregistered-motor-vehicles-india

[2] http://www.indiantollways.com/category/toll-plaza/

[3] www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 3)

[4] Kosemani O. O, Olayiwola O. E, Vincent O. R. “Estimating Traffic Intensity at Toll Gates Using Queueing Networks”. International Journal of Advanced Computer Science and Applications, Vol. 5, No. 4, 2014.

[5] Jacob Fraden “Handbook of Modern Sensors” ISBN: 978-1-4419-6466-3-1993

[6] Naima Kaabooch, Wen-Chen Hu “Big Data Management Technologies and Applications” ISBN: 978-1-4666-4699-5, 2013

[7] Rafael C. Gonzalez and Richard E. Woods “Digital Image Processing” (third edition) ISBN: 978- 0131687288, 1977

[8] Tadayoshi Kohno, Niels Ferguson and Bruce Schneier” Cryptography Engineering: Design Principles and Practical Applications” ASIN: B004NSW9JU, 2011 International Journal of Scientific & Engineering Research Volume 9, Issue 1, January-2018 ISSN 2229-5518 972 IJSER

[9] R. Gowri Poornima. “IOT Based Vehicle Parking and Toll-Fee Management System”. International Conference on Communications, Signal Processing, and their Applications (ICCSPA'15), Sharjah, 2015.

[10] Senthil Kumar Janahan, M.R.M. Veeramanickam, S. Arun, Kumar Narayanan, R. Anandan, Shaik Javed Parvez.“IoT based smart traffic signal monitoring system using vehicles counts”. International Journal of Engineering & Technology, 7 (2.21) (2018) 309-312.

[11] Ranjitha M, Spoorthi Jwanita, Soumya K. “IOT Based Smart Traffic Management System”. International Journal of Engineering Research & Technology (IJERT) 2020.

[12] P. S. Hanwate, Narin Meher, Ashlesh Mandke, Manoj Nikam, Manan Mehta. “Smart Toll Collection System based on IoT”. IJSTE - International Journal of Science Technology & Engineering, June 2017.

[13] Sudha M S , Yashodhre K, Snehil Sarkar, Spoorthy G, Shubham Khosla. “IoT Based Toll Booth Management System”. International Journal of Progressive Research in Science and Engineering Volume-1, Issue-2, May-2020.

[14] Elmer R. Magsino; Ivan W. H. Ho. “An intelligent highway tollgate queue selector for improving server utilization and vehicle waiting time”. IEEE Region 10 Symposium (TENSYMP) 2016 .

[15] P. Keerthana, T. Sangeetha, N. Surekha. “Smart Tollgate Using IoT”. International Journal of Science and Research (IJSR), 2018.

APPENDICES

APPENDIX - A

CODE

For storing the details of the RFID tags:

```
import time
import serial
ser = serial.Serial(
                    port='/dev/ttyUSB0',
                    baudrate = 9600,
                    parity=serial.PARITY_NONE,
                    stopbits=serial.STOPBITS_ONE,
                    bytesize=serial.EIGHTBITS
                    )
                    #timeout=1 # must use when using data.readline()
                    #)
print (" ")
u1rfid="4B005B344A6E"
u2rfid="4B005AA302B0"
u3rfid="4B005AAFFA44"
vrfid1=0
vrfid2=0
vrfid3=0
while(True):
      if (ser.inWaiting()>0):
             #data_str = ser.read(ser.inWaiting()).decode('ascii')
        data_str = ser.read(12).decode('ascii')
        print(data_str, end='')
        if data_str==u1rfid:
```

```
            vrfid1=1
        elif data_str==u2rfid:
            vrfid2=1
        elif data_str==u3rfid:
            vrfid3=1
        data_str=""
    time.sleep(0.01)
```

For reading the stored data & displaying relevant text on the command prompt:

```
import cv2
import os
import time
import RPi.GPIO as GPIO
import os.path
import RPi.GPIO as gp
from time import sleep
import time
import serial
f1 = open("/home/pi/vehiclenumberdatabase/1.txt", "r")
authvech1=f1.read();
f1.close()
f2 = open("/home/pi/vehiclenumberdatabase/2.txt", "r")
authvech2=f2.read();
f2.close()
f3= open("/home/pi/vehiclenumberdatabase/3.txt", "r")
authvech3=f3.read();
f3.close()
print("++++++++++++ AURTHORIZED VEHICLE NUMBERS ++++++++++++++++")
print(authvech1)
print(authvech2)
print(authvech3)
```

```
print("+++++++++++++++++++++++++++++++")
ser = serial.Serial(
                    port='/dev/ttyAMA0',
                    baudrate = 9600,
                    parity=serial.PARITY_NONE,
                    stopbits=serial.STOPBITS_ONE,
                    bytesize=serial.EIGHTBITS
                    )
                    #timeout=1 # must use when using data.readline()
                    #)
print (" ")
u1rfid="4B005B344A6E"
u2rfid="4B005AA302B0"
u3rfid="4B005AAFFA44"
vrfid1=0
vrfid2=0
vrfid3=0
servoPIN=12
manualbutton=13
buzzerpin=40
GPIO.setmode(GPIO.BOARD)
GPIO.setup(11, GPIO.IN, pull_up_down=GPIO.PUD_UP)
GPIO.setup(manualbutton, GPIO.IN, pull_up_down=GPIO.PUD_UP)
GPIO.setup(servoPIN,GPIO.OUT)
GPIO.setup(buzzerpin,GPIO.OUT)
GPIO.output(buzzerpin , 0)
def buzzering():
    GPIO.output(buzzerpin , 1)
    time.sleep(2)
    GPIO.output(buzzerpin , 0)
```

```
def SetAngle(angle):
    pwm=GPIO.PWM(servoPIN,50)
    pwm.start(0)
    duty = angle / 18 + 2
    GPIO.output(servoPIN , True)
    pwm.ChangeDutyCycle(duty)
    time.sleep(1)
    GPIO.output(servoPIN, False)
    pwm.ChangeDutyCycle(0)
    pwm.stop()
def closgate():
    SetAngle(180)
def operategate():
    SetAngle(90)
    time.sleep(5)
    SetAngle(180)
mainruncmmd ="sudo python3 /home/pi/automatictollgate/plate_recognition.py --api-key
ca80ed924a7e3a66431b56db6b7bcdd57ba8dd25 "
timestr=time.strftime("%Y%m%d-%H%M%S")
dir_path = '/home/pi/Images/'
file_name= timestr +'.jpg'
path = dir_path +file_name
withfilepath = mainruncmmd + path
cap = cv2.VideoCapture(0)
cap.set(480, 640)
cap.read()
closgate()
while(True):
    ret, frame = cap.read()
    if (ser.inWaiting()>0):
        data_str = ser.read(12).decode('ascii')
```

```
            print(data_str, end='')
            if data_str==u1rfid:
                vrfid1=1
            elif data_str==u2rfid:
                vrfid2=1
            elif data_str==u3rfid:
                vrfid3=1
        if not GPIO.input(11):
            print('Input was LOW')
            cv2.imwrite(path, frame)
            os.system(withfilepath)
            f = open("/home/pi/automatictollgate/numberdata.txt", "r")
            resultnumber=f.read()
            print("Recognized LP Number:")
            print(resultnumber)
            if resultnumber==authvech1 and vrfid1==1:
                print("---VALID NUMBER 1   PLATE AVAILABLE IN DB -----")
                operategate()
            elif resultnumber==authvech2 and vrfid2==1:
                print("---VALID NUMBER 2 PLATE AVAILABLE IN DB -----")
                operategate()
            elif resultnumber==authvech3 and vrfid3==1:
                print("---VALID NUMBER 3 PLATE AVAILABLE IN DB -----")
                operategate()
            else:
                print("******* NUMBER PLATE NOT AVAILABLE IN DB PAY
MANUALLY ******")
                buzzering()
        if not GPIO.input(manualbutton):
            operategate()
        cv2.imshow('frame',frame)
```

```
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
```

For number plate recognition:

```
from __future__ import absolute_import, division, print_function
import argparse
import collections
import csv
import json
import math
import time
import re
from collections import OrderedDict
from pathlib import Path
import os.path
import requests
from PIL import Image, ImageDraw, ImageFilter, ImageFont
def parse_arguments(args_hook=lambda _: _):
    parser = argparse.ArgumentParser(
        description=
        'Read license plates from images and output the result as JSON.',
        epilog='Examples:\n'
        'To send images to our cloud service: '
        'python plate_recognition.py --api-key MY_API_KEY /path/to/vehicle-*.jpg\n',
        formatter_class=argparse.RawTextHelpFormatter)
    parser.add_argument('-a', '--api-key', help='Your API key.', required=False)
    parser.add_argument(
        '-r',
        '--regions',
```

```
        help='Match the license plate pattern of a specific region',
        required=False,
        action="append")
    parser.add_argument(
        '-s',
        '--sdk-url',
        help="Url to self hosted sdk   For example, http://localhost:8080",
        required=False)
    parser.add_argument('--camera-id',
                        help="Name of the source camera.",
                        required=False)
    parser.add_argument('files', nargs='+', help='Path to vehicle images')
    args_hook(parser)
    args = parser.parse_args()
    if not args.sdk_url and not args.api_key:
        raise Exception('api-key is required')
    return args
def recognition_api(fp,
                    regions=[],
                    api_key=None,
                    sdk_url=None,
                    config={},
                    camera_id=None,
                    timestamp=None,
                    mmc=None,
                    exit_on_error=True):
    data = dict(regions=regions, config=json.dumps(config))
    if camera_id:
        data['camera_id'] = camera_id
    if mmc:
        data['mmc'] = mmc
```

```
    if timestamp:
        data['timestamp'] = timestamp
    response = None
    if sdk_url:
        fp.seek(0)
        response = requests.post(sdk_url + '/v1/plate-reader/',
                                 files=dict(upload=fp),
                                 data=data)
    else:
        for _ in range(3):
            fp.seek(0)
            response = requests.post(
                'https://api.platerecognizer.com/v1/plate-reader/',
                files=dict(upload=fp),
                data=data,
                headers={'Authorization': 'Token ' + api_key})
            if response.status_code == 429:   # Max calls per second reached
                time.sleep(1)
            else:
                break
    if not response:
        return {}
    if response.status_code < 200 or response.status_code > 300:
        print(response.text)
        if exit_on_error:
            exit(1)
    return response.json(object_pairs_hook=OrderedDict)
def blur(im, blur_amount, api_res, ignore_no_bb=False, ignore_list=None):
    for res in api_res.get('results', []):
        if ignore_no_bb and res['vehicle']['score'] == 0.0:
            continue
```

```
        if ignore_list:
            skip_blur = False
            for ignore_regex in ignore_list:
                if re.search(ignore_regex, res['plate']):
                    skip_blur = True
                    break
            if skip_blur:
                continue
        b = res['box']
        width, height = b['xmax'] - b['xmin'], b['ymax'] - b['ymin']
        crop_box = (b['xmin'], b['ymin'], b['xmax'], b['ymax'])
        ic = im.crop(crop_box)

        # Increase amount of blur with size of bounding box
        blur_image = ic.filter(
            ImageFilter.GaussianBlur(radius=math.sqrt(width * height) * .3 *
                                     blur_amount / 10))
        im.paste(blur_image, crop_box)
    return im
def draw_bb(im, data, new_size=(1920, 1050), text_func=None):
    draw = ImageDraw.Draw(im)
    font_path = Path('assets/DejaVuSansMono.ttf')
    if font_path.exists():
        font = ImageFont.truetype(str(font_path), 10)
    else:
        font = ImageFont.load_default()
    rect_color = (0, 255, 0)
    for result in data:
        b = result['box']
        coord = [(b['xmin'], b['ymin']), (b['xmax'], b['ymax'])]
        draw.rectangle(coord, outline=rect_color)
```

```
        draw.rectangle(((coord[0][0] - 1, coord[0][1] - 1),
                        (coord[1][0] - 1, coord[1][1] - 1)),
                       outline=rect_color)
        draw.rectangle(((coord[0][0] - 2, coord[0][1] - 2),
                        (coord[1][0] - 2, coord[1][1] - 2)),
                       outline=rect_color)
        if text_func:
            text = text_func(result)
            text_width, text_height = font.getsize(text)
            margin = math.ceil(0.05 * text_height)
            draw.rectangle(
                [(b['xmin'] - margin, b['ymin'] - text_height - 2 * margin),
                 (b['xmin'] + text_width + 2 * margin, b['ymin'])],
                fill='white')
            draw.text((b['xmin'] + margin, b['ymin'] - text_height - margin),
                      text,
                      fill='black',
                      font=font)
    if new_size:
        im = im.resize(new_size)
    return im
def flatten_dict(d, parent_key='', sep='_'):
    items = []
    for k, v in d.items():
        new_key = parent_key + sep + k if parent_key else k
        if isinstance(v, collections.MutableMapping):
            items.extend(flatten_dict(v, new_key, sep=sep).items())
        else:
            if isinstance(v, list):
                items.append((new_key, json.dumps(v)))
            else:
```

```
                    items.append((new_key, v))
    return dict(items)
def flatten(result):
    plates = result['results']
    del result['results']
    del result['usage']
    if not plates:
        return result
    for plate in plates:
        data = result.copy()
        data.update(flatten_dict(plate))
    return data
def save_results(results, args):
    path = Path(args.output_file)
    if not path.parent.exists():
        print('%s does not exist' % path)
        return
    if not results:
        return
    if args.format == 'json':
        with open(path, 'w') as fp:
            json.dump(results, fp)
    elif args.format == 'csv':
        fieldnames = []
        for result in results[:10]:
            candidate = flatten(result.copy()).keys()
            if len(fieldnames) < len(candidate):
                fieldnames = candidate
        with open(path, 'w') as fp:
            writer = csv.DictWriter(fp, fieldnames=fieldnames)
            writer.writeheader()
```

```
            for result in results:
                writer.writerow(flatten(result))
def custom_args(parser):
    parser.epilog += 'To blur images: python plate_recognition.py --sdk-url
http://localhost:8080 --blur-amount 4 --blur-plates /path/to/vehicle-*.jpg\n'
    parser.epilog += 'To save results: python plate_recognition.py --sdk-url
http://localhost:8080 -o data.csv --format csv /path/to/vehicle-*.jpg\n'
    parser.add_argument('--engine-config', help='Engine configuration.')
    parser.add_argument('-o', '--output-file', help='Save result to file.')
    parser.add_argument('--format',
                        help='Format of the result.',
                        default='json',
                        choices='json csv'.split())
    parser.add_argument(
        '--mmc',
        action='store_true',
        help='Predict vehicle make and model. Only available to paying users.')
    parser.add_argument(
        '--blur-amount',
        help=
        'Amount of blurring to apply on the license plates. Goes from 0 (no blur) to 10.
Defaults to 5. ',
        default=5,
        type=float,
        required=False)
    parser.add_argument(
        '--blur-plates',
        action='store_true',
        help='Blur license plates and save image in filename_blurred.jpg.',
        required=False)
def json_extract(obj, key):
```

```
    """Recursively fetch values from nested JSON."""
    arr = []
    def extract(obj, arr, key):
        """Recursively search for values of key in JSON tree."""
        if isinstance(obj, dict):
            for k, v in obj.items():
                if isinstance(v, (dict, list)):
                    extract(v, arr, key)
                elif k == key:
                    arr.append(v)
        elif isinstance(obj, list):
            for item in obj:
                extract(item, arr, key)
        return arr
    values = extract(obj, arr, key)
    return values
def main():
    args = parse_arguments(custom_args)
    paths = args.files
    results = []
    engine_config = {}
    if args.engine_config:
        try:
            json.loads(args.engine_config)
        except json.JSONDecodeError as e:
            print(e)
            return
    for path in paths:
        with open(path, 'rb') as fp:
            api_res = recognition_api(fp,
                                      args.regions,
```

```
                                        args.api_key,
                                        args.sdk_url,
                                        config=engine_config,
                                        camera_id=args.camera_id,
                                        mmc=args.mmc)
            if args.blur_plates:
                im = blur(Image.open(path), args.blur_amount, api_res)
                filename = Path(path)
                im.save(filename.parent / ('%s_blurred%s' %
                                         (filename.stem, filename.suffix)))
            results.append(api_res)
        if args.output_file:
            save_results(results, args)
        else:
            #print(json.dumps(results, indent=2))
            jtopy=json.dumps(results)
            #print(jtopy)
            results_dict = json.loads(jtopy)[0]
            #print("Type of dict_obj", type(results_dict))
            print("---------------------------")
            #print(results_dict.get['results'])
            plate_values = json_extract(results_dict, 'plate')
            stringplatenum = str(plate_values[0])
            predictednumber = stringplatenum.upper();
            print(predictednumber)
            f = open("/home/pi/automatictollgate/numberdata.txt", "w")
            f.write(predictednumber)
            f.close()
            print("------------------------------")
if __name__ == '__main__':
    main()
```

APPENDIX - B

BIOGRAPHY

NAME : VALPADASU SAI SUDHAMSH **ROLL NO :** 18831A04B2 **EMAIL :** SAIVALPADASU30@GMAIL.COM **PHONE :** 81795986360