IOT BASED BIOMETRIC AUTHORIZATION FOR VEHICLE START UP AND MONITORING OF FUEL

SUPERVISED BY

Md. Din Al-Amin

Lecturer and Assistant Coordinator

Department of Mechanical Engineering

Sonargaon University (SU)

SUBMITTED BY

Md. Shakil Hossain
Student ID: BME1803016500

Md. Mosharef Hossain
ID : BME1803016499

Md. A. Halim
ID : BME1803016501

Md. Shahadat Hossain
ID : BME1803016496

Md. Muhaimenul
ID : BME1803016517

An undergraduate thesis

Submitted to

The Department of Mechanical engineering, Sonargaon University, in partial fulfillment of the requirements for the degree of **"Bachelor of Science in Mechanical Engineering".**

Department Of Mechanical Engineering

Sonargaon University (SU)

APPROVAL SHEET

This thesis entitled "IoT Based Biometric Authorization for Vehicle Start Up and Monitoring of Fuel" prepared and submitted by Md. Shakil Hossain (BME1803016500), Md. Mosharef Hossain (BME1803016499), Md. A. Halim (BME1803016501), Md. Shahadat Hossain (BME1803016496) and Md. Muhaimenul (BME1803016517) in partial fulfillment of the requirements for the degree of "Bachelor of Science in Mechanical Engineering" has been examined and recommended for approval and acceptance.

Md. Din Al-Amin

Thesis Supervisor

ACKNOWLEDGEMENT

All praise is due to Almighty Allah who created us. First of all, we would like to pay our respects and special thanks to our Honorable sir Md. Mostafa Hossain Sir, He is the Head of our Mechanical Engineering Department. We would also like to express our gratitude and sincere thanks to our assistant coordinator, Mohammad Din Al-Amin. Sir helped us with the direction of writing this thesis. Also, we would like to thank our classmates for their hard work and time in helping us with this thesis. While collecting information and data for this thesis we have received help and advice from many people and professionals and have collected information from many books which are mentioned in the reference. Without all this help, it would have been difficult to complete this thesis successfully. In addition, we have received help from many people whose name is this not on the acknowledgement, we are forever grateful to them.

ABSTRACT

Transportation plays a vital role in human daily activities, resulting that the number of urban automobiles are quickly increasing. On the contrast car theft is now a great problem across the country various vehicle antitheft devices have been developed latterly, but the outcome is still unsatisfactory since every kind of device has its drawbacks. Therefore, an enhanced system has been proposed in this report to ensure vehicle safety and track the vehicle in the event of theft. This proposed system includes a smart vehicle security starting system based on the internet of things using fingerprint authentication to enable the engine ON in addition to the key mechanism. The user has to use both fingerprint and key, to access the vehicle. Even if one input out of the two is available, the vehicle cannot be turned on. At a time fuel has been monitored of the fuel tank. This report mainly focuses on the development of an android app that improves the problems of transport burglary, tracking and monitoring system. Fingerprint sensor has been used as biometric recognition of the vehicle's user all the time, but in case of emergency (e.g., accidents or user's sickness etc.) Limit switch has been placed for door security system. Anyone can take the control of the vehicles taking the permission from the authorized person. The grand feature of the presented vehicle security system is that, it can track the vehicles from anywhere of the globe with the help of GPS module.

TABLE OF CONTENTS

LIST OF FIGURE

LIST OF TABLE

ABBREVIATION

IoT	Internet of Things
GPS	Global Positioning System
LCD	Liquid Crystal Display
Wi-fi	Wireless Fidelity
PCB	Printed Circuit Board
NEO	New Earth Object
I2C	Inter-Integrated Circuit
VCC	Voltage Common Collector
GND	Ground

CHAPTER 1

INTRODUCTION

1.1 Background

The present age, everything is now in the control of the Internet. Modern control methods have made life easier through the use of information technology. If the terrestrial communication system can be controlled through information technology [1]- [2], it will be a revolutionary change. This is especially true of vehicle owners or agency dealers. It often faces various problems like vehicle theft, accidents, driver fraud, not getting the car to the destination on time etc. Various high-priced, luxurious automobiles are available in today's world of comfort and luxury. Many of these automobiles came equipped with security systems from the start. Nonetheless, even despite of the fact that a significant amount of money is being invested in some sectors, vehicle theft is still continuing on the rise, in spite of the improvements in vehicle security system. [3]-[4] If it is possible to provide a second level of security in this case against vehicles stealing with duplicate keys or otherwise, then it will prevent vehicle stealing in a great extent.

In the case of vehicle agencies, taking a driver other than the designated driver of a designated vehicle is a schedule disaster. Many a times the car owners who rent a car cannot collect the rent on time and they cannot control it. If the biometric system can be used, then no unregistered person will be able to drive. There are various Biometric methods like face detection, optical detection, fingerprint etc. are available, however this project uses fingerprint biometric, which is readily available and its usage is low.

Fuel management system is the important portion in our vehicle. Fuel management systems are designed to effectively measure and manage the use of fuel within the transportation. This information can be then stored in computerized systems and reports generated with data to inform management practices. This enables consumption control, cost analysis and tax accounting for fuel purchases. Modern vehicle tracking systems commonly use GPS technology for locating the vehicle. As fuel cost is rising constantly, vehicle owners are facing fuel theft from his vehicle. In that case, vehicle owners do not calculate how much quantity of fuel needs in a day or in a week. Time is important for efficient goods transportation system such as real time tracking and reducing financial fuel loss of vehicles is rapidly increased. This can be overcame by using real time tracking vehicle system, such as a system which monitors current amount of fuel via mobile application. Using this approach, a vehicle owner will help the user to monitor and control all of the vehicles from a remote location. Again, real time fuel-filled and fuel consumption in vehicles is not maintained. For this kind of problem, when the driver starts filling fuel in the tank, the floating sensor gets activated and store data on the mobile application. Some of the drivers can drive without asking his vehicle owner. At that time, the driver used the vehicle for rental or personal use. By utilized the mobile application, the owner can trace all the vehicles in the same period, find very nearest fuel pump from the vehicle location, notify when fuel goes to a certain level

The whole process will be controlled by a mobile based app. In this case, the fingerprints of the designated drivers will be taken in the database in advance. Each vehicle will have a fingerprint sensor in addition to the car start switch. The fingerprint sensor will have to give his fingerprint when the designated driver of a particular car starts the car and that fingerprint will be stored in the database via a Wi-Fi module. The owner of the car will be notified on the app by the number of the vehicle and the identity of the driver of the vehicle. Now if the owner of the car does not allow permission, no one will be able to start that car. Now in many cases it can happen that for some reason someone else has to drive the car in the absence of the specific driver or the driver has been injured in a car accident, it is needed to remove that car from the road by second person. On the basis of the foregoing the presented project having options of both enable and disable fingerprint is so designed that the owner can allow the car to run without fingerprints when needed.

This project is talking about a mobile based app that have been developed with the help of MIT App Inventor. Also, Google Firebase is used as a database to store all the data, which will act as the database of the mobile app. In the process Wi-Fi module is needed to send the fingerprint from the car to the database. A microcontroller is also needed to handle the entire process. Many projects are managed using GSM but ESP8266 microcontroller is used because it has Wi-Fi module facility.

IoT (Internet of Things): The Internet of Things (IoT) is a new paradigm that has converted traditional lifestyles into high-tech lifestyles. IoT changes include smart cities, smart homes, pollution management, energy conservation, smart transportation, smart industries, and smart libraries. In order to advance technology via IoT, numerous key research studies and investigations have been done. However, a number of challenges and issues must be overcome in order to fully fulfill the potential of IoT. These issues and challenges must be assessed from a range of angles, including applications, roadblocks, enabler technologies, social and environmental effects, and so on. [3]

The Internet of Things (IoT) refers to physical things (or groups of such objects) which are connected with sensors, computing power, software, and other technologies to interconnect and transfer data with other devices and networks over the Internet or other communication systems. [1]

ESP 8266: In this system the data transfer in google firebase database is managed through IOT by ESP8266 (shown in figure 1.2).The ESP8266 is nothing but a low cost Wi-Fi microchip. Espressif Systems in Shanghai, China design the ESP8266 Wi-Fi module to support both the TCP/IP networking software and the microcontroller access to any Wi-Fi network. [6]-[7]

Embedded Systems produces the ESP8266 wi-fi SOC (system on a chip). It's a fully integrated chip that's meant to fit into a compact packaging and give complete internet access. The ESP8266 may be used as an external i-fi module by connecting it to any microcontroller through the serial UART and using the regular AT Command set Software, but can also be used direct as a Wireless microprocessor by using the included SDK to create a new software.

MIT app inventor: MIT app inventor is visual programing system. Through this anyone as well as children can make app very easily. After the invention of MIT app inventor there is no need to write program and it also saves time. Professor Hal Abelson leads a small group of CSAIL personnel and students who constitute the basis of an international movement of inventors. This core team manages the free online app creation environment that serves over 6 million registered users, in addition to directing educational outreach and doing research on its benefits. [5]

MIT App Inventor is a drag-and-drop visual programming tool for designing and building fully functional mobile apps for Android. App Inventor promotes a new era of personal mobile computing in which people are empowered to design, create, and use personally meaningful mobile technology solutions for their daily lives, in endlessly unique situations. App Inventor’s intuitive programming metaphor and incremental development capabilities allow the developer to focus on the logic for programming an app rather than the syntax of the coding language, fostering digital literacy for all. Since it was moved from Google to MIT, a number of improvements have been added, and research projects are underway.

Android app: Android Inc. was founded in Palo Alto, California, in October 2003 by Andy Rubin, Rich Miner, Nick Sears, and Chris White. Rubin described the Android project as having "tremendous potential in dev eloping smarter mobile devices that are more aware of its owner's location and preferences" Android, operating system for cellular telephones and tablet computers (shown in figure 1.3).

Android began in 2003 as a project of the American technology company Android Inc., to develop an operating system for digital cameras. In 2004 the project changed to become an operating system for

Smart phones. Android Inc., was bought by the American search engine company Google Inc.in 2005. At Google, the Android team decided to base their project on Linux, an open-source operating system for personal device.

On November 5, 2007, Google announced the founding of the Open Handset Alliance, a consortium of dozens of technology and mobile telephone companies, including Intel Corporation, Motorola, Inc., NVIDIA Corporation, Texas Instruments Incorporated, LG Electronics, Inc., Samsung Electronics, Sprint Nextel Corporation, and T-Mobile (Deutsche Telekom). The consortium was created in order to develop and promote Android as a free open-source operating system with support for third-party applications. Android-based devices use wireless networks in order to take full advantage of features such as one-touch Google searches, Google Docs (e.g., word editors, spreadsheets), and Google Earth (satellite mapping software).

The two main types of software are system software and application software. System software controls a computer's internal functioning, chiefly through an operating system, and also controls such peripherals as monitors, printers, and storage devices. Application software, by contrast, directs the computer to execute commands given by the user and may be said to include any program that processes data for a user. Application software thus includes word processors, spreadsheets, database management, inventory and payroll programs, and many other "applications." A third software category is that of network software, which coordinates communication between the computers linked in a network.

1.2 Objectives

The aim of the presented project is to implement an app-based vehicle monitoring and security system. The main motive of this project is to increase the safety of vehicles. The owner of the vehicle should be able to monitor the condition of the vehicle around all the time. The vehicle system of this country with new technology can be developed using the applied method. That is to say, our system will have the following objectives:

- To start the vehicle through finger print sensor.
- To monitor the level of fuel in the fuel tank.
- To ensure the door safety system for vehicle by using limit switch.
- To show notifications on Android app via IoT.
- To enable or disable the engine of vehicle through the android app.
- To check the real time location of vehicle via app.

1.3 Scope and Limitations of The Study

- Response of driver could not identify. So that some accident would be occurred for incognizant of driver.
- Total load could not measure of our vehicle. When it was over load, it was risk for passengers and also destroyed our vehicle lifetime. Implementation load cell sensor inside vehicle, we will know total weight of our vehicle. At that time vehicle will safe and accident will decrease.
- This monitoring system would be lose in rural area. For this reason, we could not monitor this system in our mobile application when vehicle went to rural area.
- We could not calculate vehicle speed through the mobile apps. If driver drive the vehicle roughly, we could not understand. Accident will be held for this reason.
- GPRS will also implement in future. So that, we will not need to use pocket router. It is a packet oriented mobile data standard on the 2G and 3G cellular communication network’s global system communications. After implementation this system, we can monitor our vehicle in rural area.

1.4 Methodology of the Study

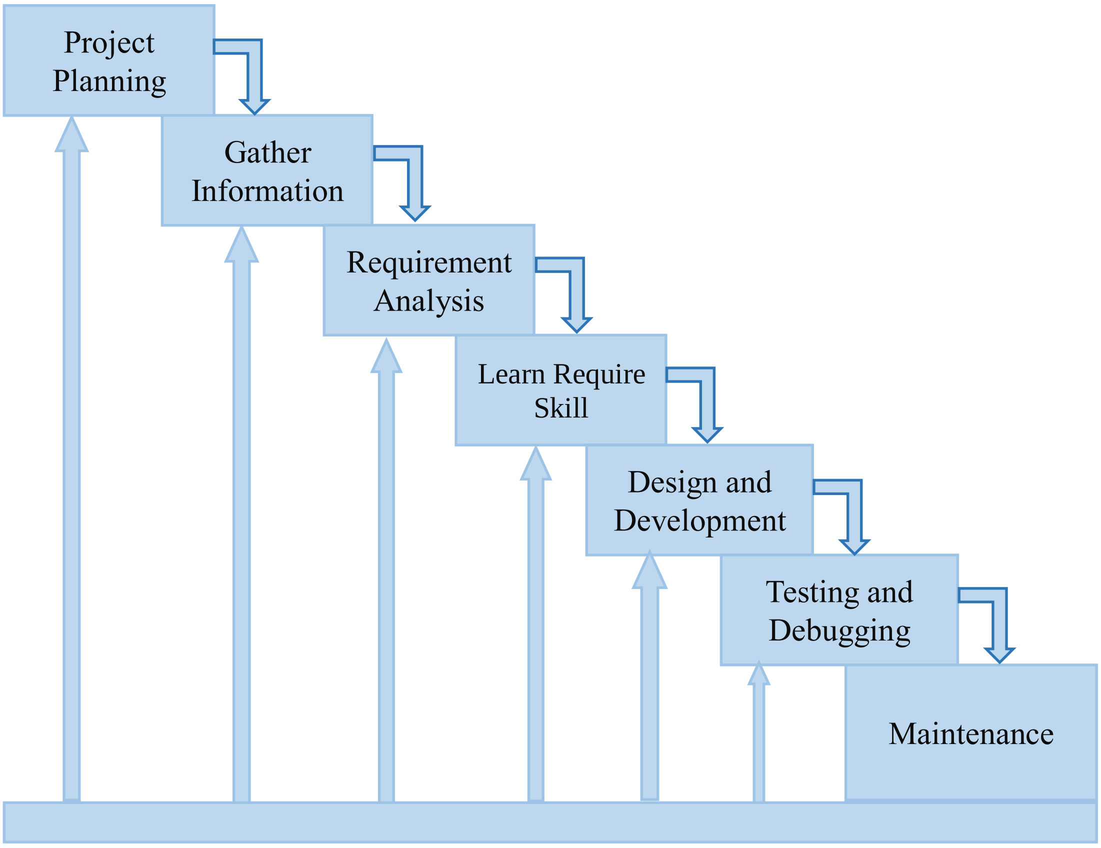


Fig 1.1: Methodology diagram of the project

Methodology is the systematic, theoretical analysis of the methods applied to a field of study. Actually Methodology is some theoretical step or work schedule that flowed by any project. It comprises the theoretical analysis of the body of methods and principles associated with a branch of knowledge. Typically, it encompasses concepts such as paradigm, theoretical model, phases and quantitative or qualitative techniques. Our project work has completed by following some strategy, which are given below.

1.4.1 Project Planning

Project planning is a discipline that specifies how to accomplish a project in a specific amount of time, typically with specified stages and resources. First, we discovered a few common issues in our day-to-day lives. Then locate this issue and consider a clever solution within a set time limit. With our team members and teachers, we had a discussion. We have devised a clever mechanism that can feed data to nodeMCU. For the particular command that is required, it can also determine the condition of the vehicle.

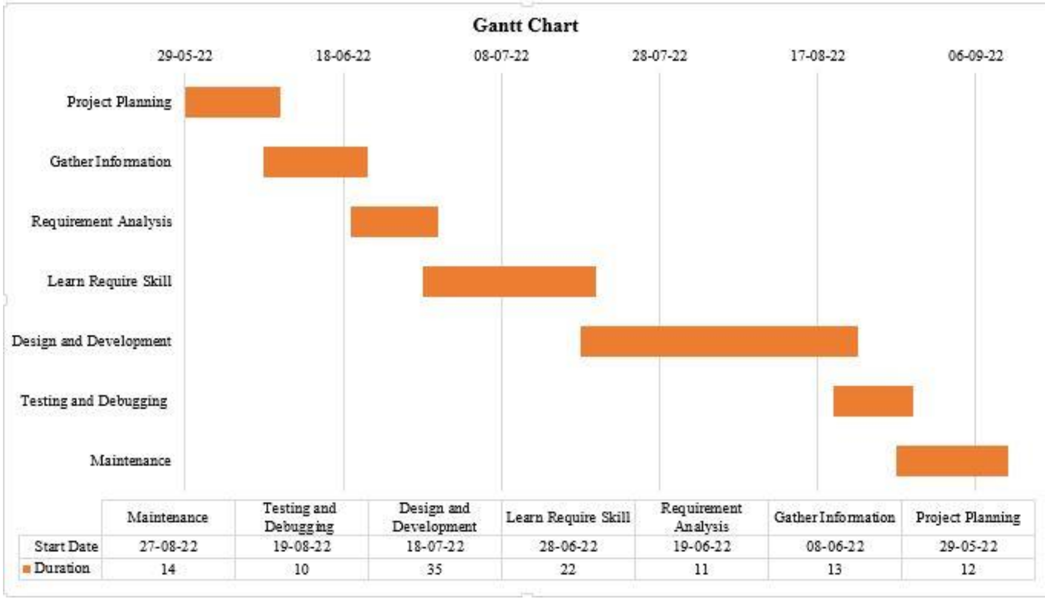


Fig 1.2: Gantt chart of this project

1.4.2 Gather Information

After finding problem we gathered some information about this problem that how to solve it in smartly. We read some research paper about this problem. We search in internet to find the solutions. We collect information from different vehicle system and their operation system.

1.4.3 Requirements Analysis

The tasks involved in figuring out the requirements or conditions that must be met for a brand-new or modified project or product are included in requirements analysis. We are considering the requirements to make this project effective after acquiring the necessary data. The suggested system makes use of three sensors. The writers had to study a lot of hardware-related information as well as a few computer languages in order to finish this project like Arduino, numerous sensor programming, the C programming language, etc. The information that authors have gathered includes.

1.4.4 Learn Required Skill

For completing the project we had to know about Android language Java, Hardware connection. We practices Arduino IDE programing and developed code in this software. At a time, we learned about block programming for developing android apps. We have use MIT App Inventor to develop the apps. We have trained about DC circuit design which is implemented our project. We develop our primary circuit in the bread board. Different type of sensor, module and micro-controller were used to develop this project. We also created a database to store data in server.

1.4.5 Design and Development

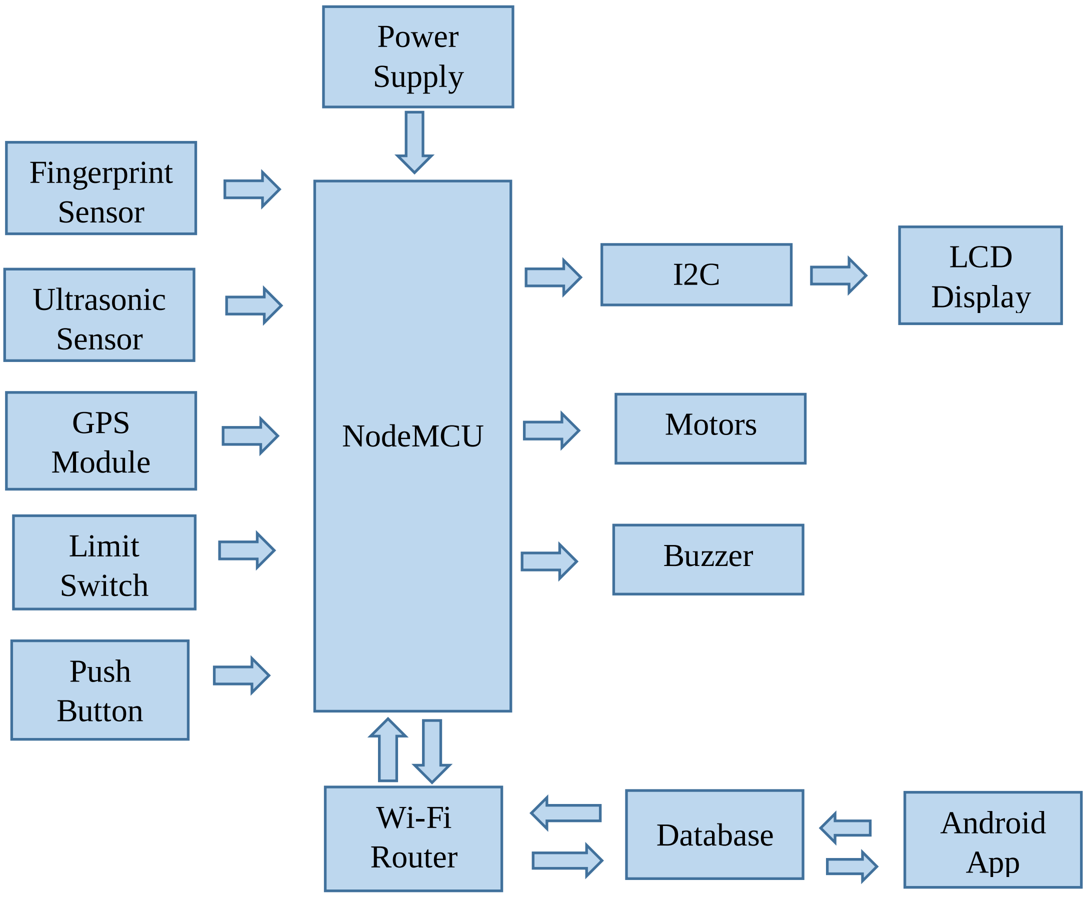


Fig 1.3: Block diagram of the system

The process of defining a system's modules, interfaces, components, and data in order to meet predetermined requirements is known as system design. The process of building or changing a system, as well as the procedures, techniques, models, and development approaches, is referred to as system development. Utilizing NodeMCU, a power supply, a web interface server, and an Android app, we created the system architecture. Here, some stage of a methodology was employed to construct the device.

1.4.6 Testing and Debugging

A device that is being tested is also known as equipment under test or unit under test, is a manufactured product that is being tested, either during its initial manufacturing or later in its life cycle as part of continuing functional testing and calibration checks. We check the value of sensor at maximum and minimum level of fuel in the tank. Also check the trig point of limit switch. The whole process was checked the part to part segment.

1.4.7 Maintenance

There are some issues which come up in the client environment. To fix those issues patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment. The main methodology of this entire project is depends on IoT based embedded system so inter facing of all hardware with Wi-Fi and internet is very important part in its functioning. An ultrasonic sensor was placed in the fuel tank. It could calculate the level of fuel tank. From this fuel level, vehicle owners easily monitor fuel consumption. So anybody theft fuel from vehicle, vehicle owner will notify through mobile app. A GPS is placed in the front part of the vehicle and it gives longitude and latitude values. The values of sensor and GPS are collected by NodeMCU as it has inbuilt Wi-Fi module all the data is transferred to the cloud through Wi-Fi and analysis is done in mobile application and notifications are sent according to the conditions.

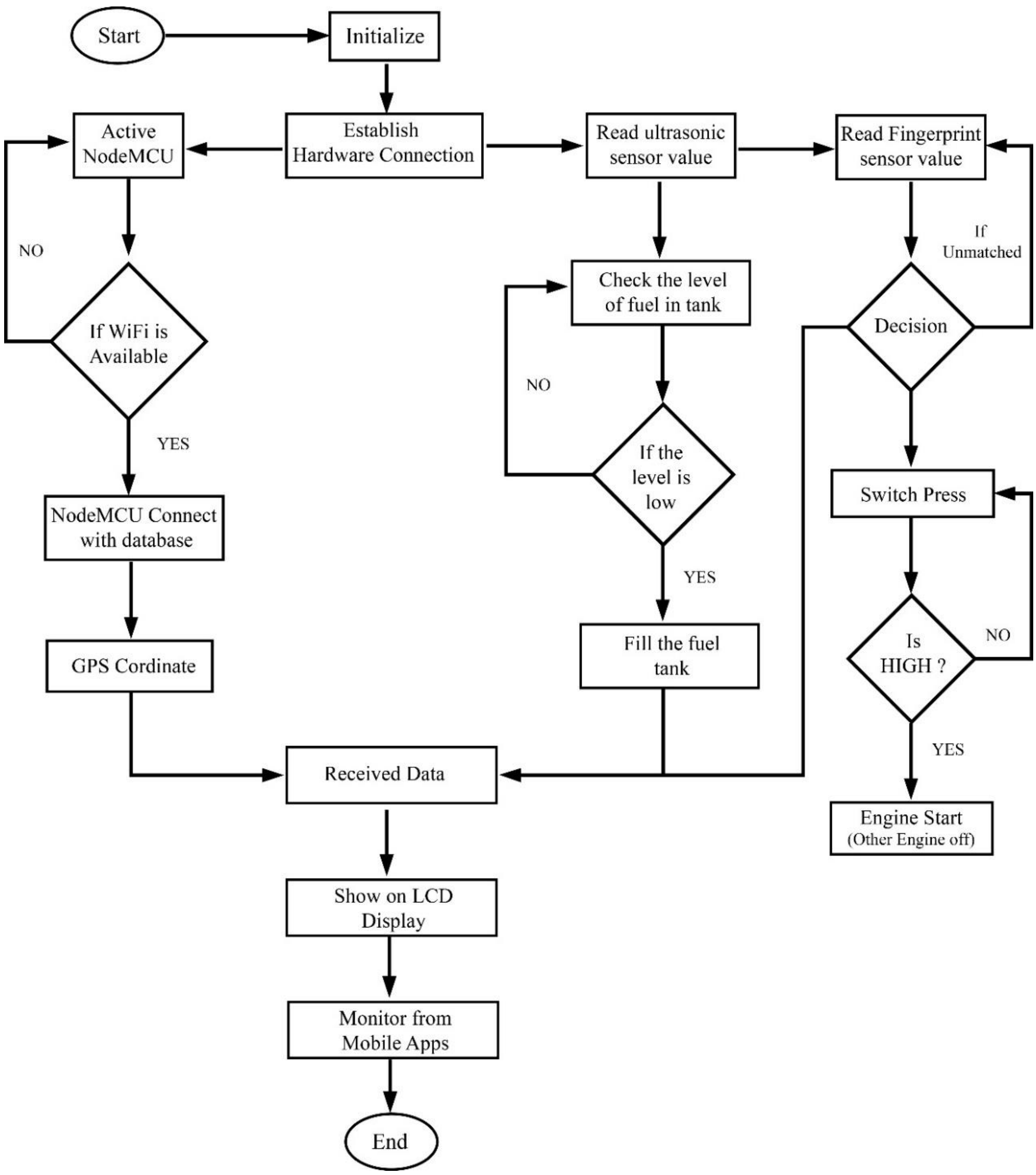


Fig 1.4: Flow Chart of the project

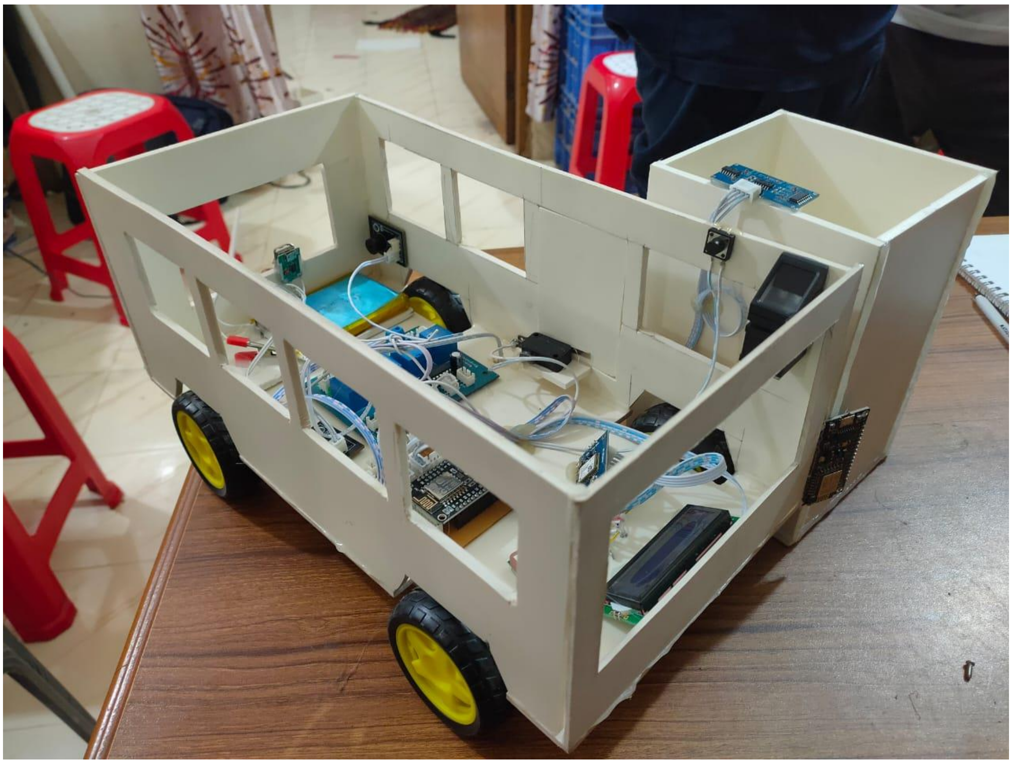


Fig 1.5: Total setup of whole system

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In this chapter, a brief literature survey in this field is presented. Some researchers used new sensor to improve the work efficiency, where some others used GSM and PIC18F4620 as micro controlling system. In literature we have found that most of the existing works presented a security system focusing on complete app-based network.

2.2 Literature Review

A prototype of a car ignition system using a fingerprint sensor was proposed [9]. The automobiles can't be stolen with this technology in place. It is designed to use a fingerprint scanner to control the vehicle's ignition. This system comprises of a GSM SIM 900 that is connected to the Arduino, the project's microcontroller to make certain only approved fingerprints are matched with the Arduino to start the ignition, making the system safe. Vehicles start if the registered fingerprint is validated with the database's biometrics, however users who have no matched in the firebase are prevented from starting the car. A burglar alarm is activated by a buzzer, and a notice is sent to the landlord's handset through GSM SIM 900 and the LCD monitor screen are relevant signals to the operator. The ESP8266 module having Wi-Fi module used rather than GSM SIM900 makes the presented project more convenient.

A finger print-based automobile starting system was designed to prevent burglary and illegal use [10]. They used the fingerprint module, PIC18F4620 microcontroller, and Liquid Crystal Display (LCD) module to do this. In this paper, they don’t develop android apps. Also, ESP8266 module has been used rather than PIC18F4620, which includes wi-fi module.

An app that minimizes traveler wait times at bus stops by exchanging bus information between drivers and passengers was designed [11]. The precise arrival and departure times of buses are provided to passengers so that they may conveniently wait. The most crucial feature of the presented app. is Real-time location tracking, which will keep the user of the vehicles being informed. Bus and ETA information will be shown on checkpoints, as well as SMS and email warnings. However they only designed it for MUET campus and check the travelling time and reduce it.

IoT is extensively used in everyday object and its popularity is increasing day by day. In this paper it includes the design and development of an IoT and mobile-based vehicle fuel activities such as real time fuel monitoring and GPS tracking system. The proposed IoT device measures the amount of fuel by using ultrasonic fuel sensor.

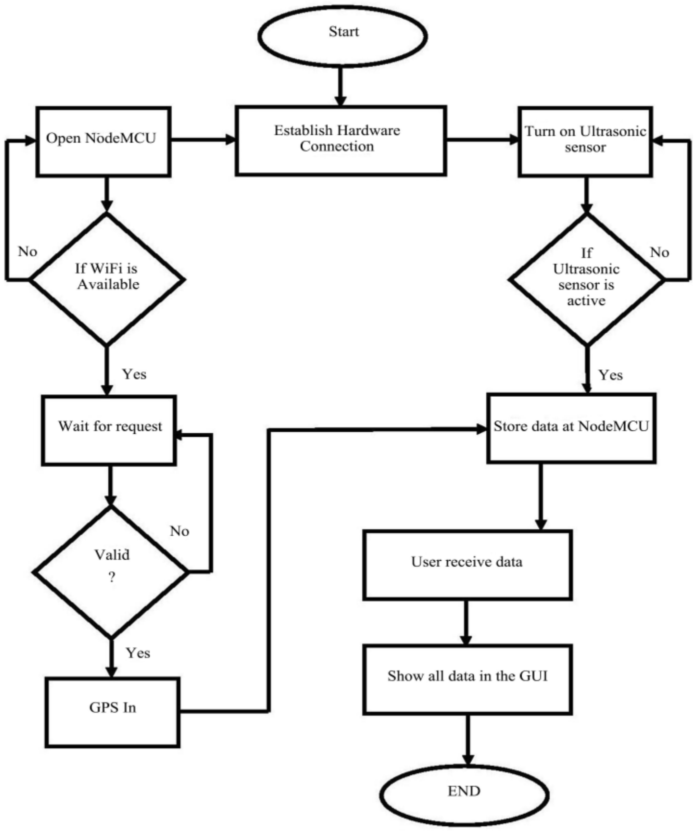


Fig 2.1: System Block Diagram-1

When the vehicle tank of fuel reaches a certain level, driver gets notification through mobile application and also searches the nearest pump location for reloading fuel. The proposed system used GPS tracking for showing current location of vehicle and finding nearest pump location.

A software prototype that utilizes the Global Positioning System (GPS) on a Mobile phone to detect the position of a car as well as stores the coordinates in a Firebase Real-time Database was designed [12] Vehicle monitoring software is essential for tracking the movement of vehicles including buses, trains, and taxis. The precision of vehicle tracking systems is one of the difficulties. It is feasible with the aid of technologies. Enabling the drivers of the vehicles to offer an accurate time estimate. The time of arrival for the users is the goal of this research is to enhance the accuracy of predicted vehicle time so that users know what to expect the vehicle's appearance. In addition, the vehicle's projected arrival time is calculated using the Google distance matrix Application Programming Interface (API). Based on suggested path and traffic statistics, the Distance Matrix API can yield traveling timing sites. The software prototype underwent numerous levels of system testing and review. Based on customer input, the methods have improved its estimation of travel times. Throughout the project controlling precision is improved. The vehicle owners can stop the car anytime as well as can access to start the car through app.

A prototype of vehicle's working system was proposed to access the car using a node MCU [13] If the activation key is missing or misplaced, accessing the vehicle's working system is impossible. It allows drivers to maintain a connection with their car. When the starter device is stolen, the car cannot be started because the ignite mechanism is still frozen on the vehicle start, and only the registered user with the pairing of ignition key and mobile phone application will be able to resume the vehicle at a sensible time. In order to evaluate the system, this research recommends using a node MCU, Bluetooth low energy (BLE), transistors, power relays, and a Mobile platform. It is also cost-effective, and once placed in the car, it requires no more servicing.

The development of a car guard and alarm system based on IoT technology that uses biometric authentication was designed [14]. For vehicle security concerns, the suggested system VSS IoT, which is based on the interface of a Raspberry Pi 3 Model B+ development board, Pi camera, PIR sensor, and smart-phone, grants only complete access to recognized vehicle drivers. As a result, if the suggested system identifies an alleged infringer inside the car, it will inform and communicate his picture, as well as the vehicle's position, to the vehicle's owner and/or a police workstation through the Internet, in case the vehicle is robbed or harmed. The suggested solution is tested on two datasets: ORL and our own.

A GSM technology was applied to send messages [15]. The user may manage the vehicle and, if necessary, switch it off. The system additionally uses an alphanumeric pin (with a total of 3 attempts) to regulate the entrance of a security vault door and the using of a safety belt. If a window burglar is detected, the IR sensor identifies the thief, as well as any obstacles, and delivers a signal to the microcontroller. A Bluetooth module and a burglar alarm are both attached to the device. The system sends an alarm sound to the dashboard (which is essentially a smartphone), which then provides an alarm condition to the user's smartphone.

A technology for automatic person identification was employed [15] Each buses and trucks company is used to rent their vehicle. Similarly, protecting the car against theft is critical. Vehicle theft can be thwarted remotely by a trusted individual. Embedded computer technology is a new field that has employed in a variety of applications. Embedded systems, as well as the Global System for Mobile (GSM) and Fingerprint Recognition, are used to create a reliable car security system. The assessment focuses on two primary technologies for automatic person identification: fingerprint recognition and existing car security systems.

2.3 Problem Statement of the Study

- Here we only monitored the fuel consumption. We could not monitor the passengers and driver response. So that something accident would be happened.
- We informed that this monitoring system would be lose in rural area. For this reason, we could not monitor this system in our mobile application when vehicle went to rural area.
- We could not calculate vehicle speed through the mobile apps. If driver drive the vehicle roughly, we could not understand. Accident will be held for this reason.
- We could not place any shock sensor in the vehicle. So we cannot monitor accidental area zone.

2.4 Conclusion

All the literature has tried to establish a security system controlled by microprocessor. However, all the proposed security system has some disadvantages, which limits the application in practical use. There are a lot of scope to improve the security system of the vehicles using advanced android and IoT technique. In android and IoT based security system, effectiveness of the system is monitored using authentication system

CHAPTER 3

HARDWARE SETUP AND PROGRAMING

3.1 Introduction

This project is specially designed for vehicle theft, and driver-to-vehicle conflict, at the same time more different types of problems are seen. Project assessment and monitoring, compliance with business and government requirements, and any data analysis pertinent to project tasks the authors have examined how this initiative will require critical and technical support. The needed to identify the components, modules, interfaces, and data for a system in order to meet specific criteria is known as system design. Mobile authentication signal provide from authority so that driver can start the vehicle. ESP-8266, GPS Module, limit switch finger print sensor, relay module, motor and wheel was used to construct the body of the body. In the project environment, there are a few challenges that arise. Patches have been provided to address these issues. Fingerprint sensors, ultrasonic sensors GPS module, limit switch and LCD display others modules were connected with ESP-8266.

3.2 Hardware Setup

It is always critical to have accurate knowledge about all of the project's equipment and computer software. To create idealize, we used electronics components. The main and most important component in our extension is the nodeMCU. Because it contains all of the computer program information, we consider this section to be the heart of our extension. We have utilized fingerprint and ultrasonic sensors. LCD display unit have placed for showing data which size is 16x2. GPS module was placed for check the coordinate and finf the exact location of the vehicle. We have used lots of hardware and software components in our project. Now let’s discuss about those components in below:

- Fingerprint Sensor
- Ultrasonic Sensor
- ESP-8266
- GPS Module
- LCD Display
- I2C Module

- Yellow Motor
- Yellow Wheel
- Limit Switch
- Push Button

3.2.1 Finger Print Sensor Connection

Fingerprint readers are used to recognize and authenticate an individual's fingerprint. Fingerprint scanners are safe and dependable security authentication equipment. It was used a digital camera to capture a visual image of the fingerprint. This scanners were created an image of the fingerprint by using capacitors and hence electrical current. In terms of precision, this type of scanner excels. It was a type of electronic security system that uses fingerprints for biometric authentication to grant a user access to information or to approve transactions. This finger sensors was very finger touch sensing signal output, low effective, sensing circuit standby current is very low, less than 5uA.

Fig 3.1: Fingerprint sensor

Working System:

At first VCC and GND was supplied from the both site of the controller and sensor. Then TX pins of finger print sensor was connected with D7 pin and RX pin of figure was connected with D6 pin of ESP-8266.

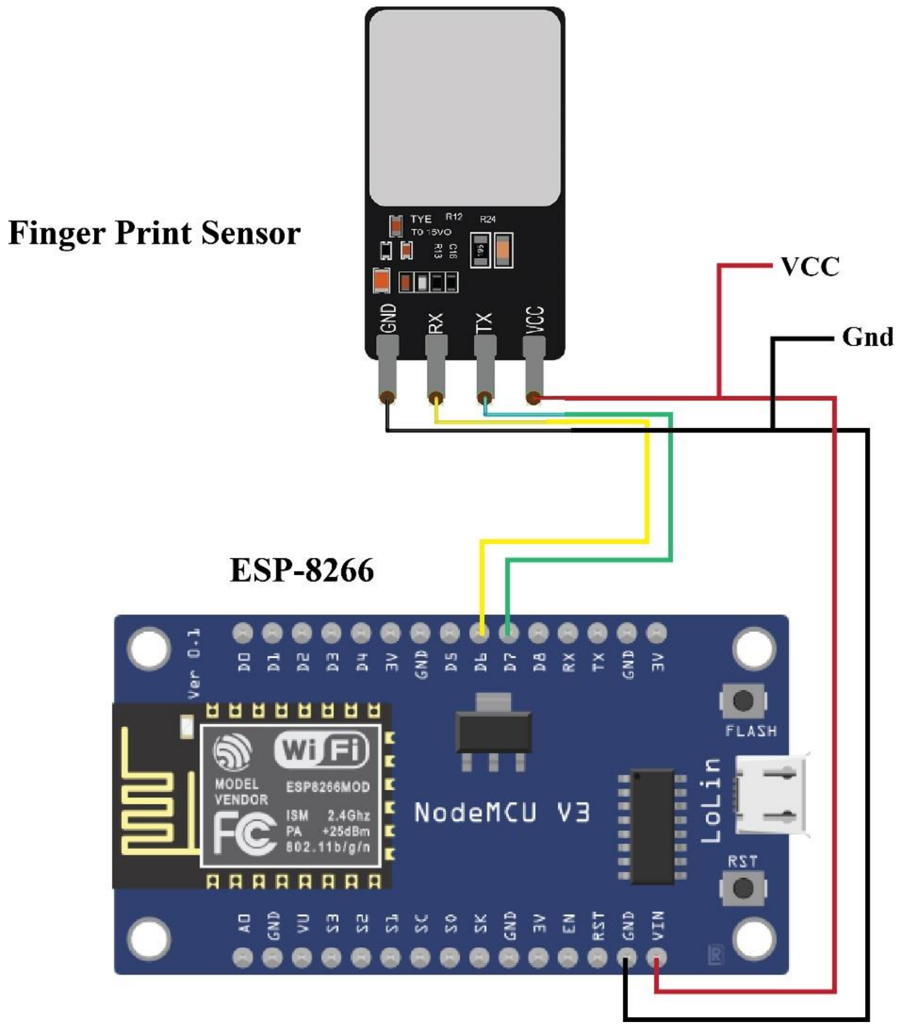


Fig 3.2: Finger Print Sensor Connection with ESP-8266

Through using this communication system was developed sensor and controller. This sensor was given an optical view of finger. This view was compared by actual view of finger. At first it was taken optical view of drivers. So only driver finger was matched. Other it was not matched. The accuracy of the sensor was good.

Fig 3.3: Finger Print Sensor attached in vehicle

3.2.2 Ultrasonic Sensor

Ultrasonic sensing is one of the best ways to sense proximity and detect levels with high reliability. It is an instrument that measures the distance to an object using ultrasonic sound waves. An ultrasonic sensor uses a transducer to send and receive ultrasonic pulses that relay back information about an object's proximity. In this time, we are used this sensor for measuring level of fuel. From this sensor, we know that actual fuel level of our system.

Fig 3.4: Ultrasonic Sensor

Specification:

- Power Supply: DC 5V
- Working Current: 15mA
- Working Frequency: 40Hz
- Ranging Distance: 2cm – 400cm/4m
- Resolution: 0.3 cm
- Measuring Angle: 15 degree
- Trigger Input Pulse width: 10uS

Working System:

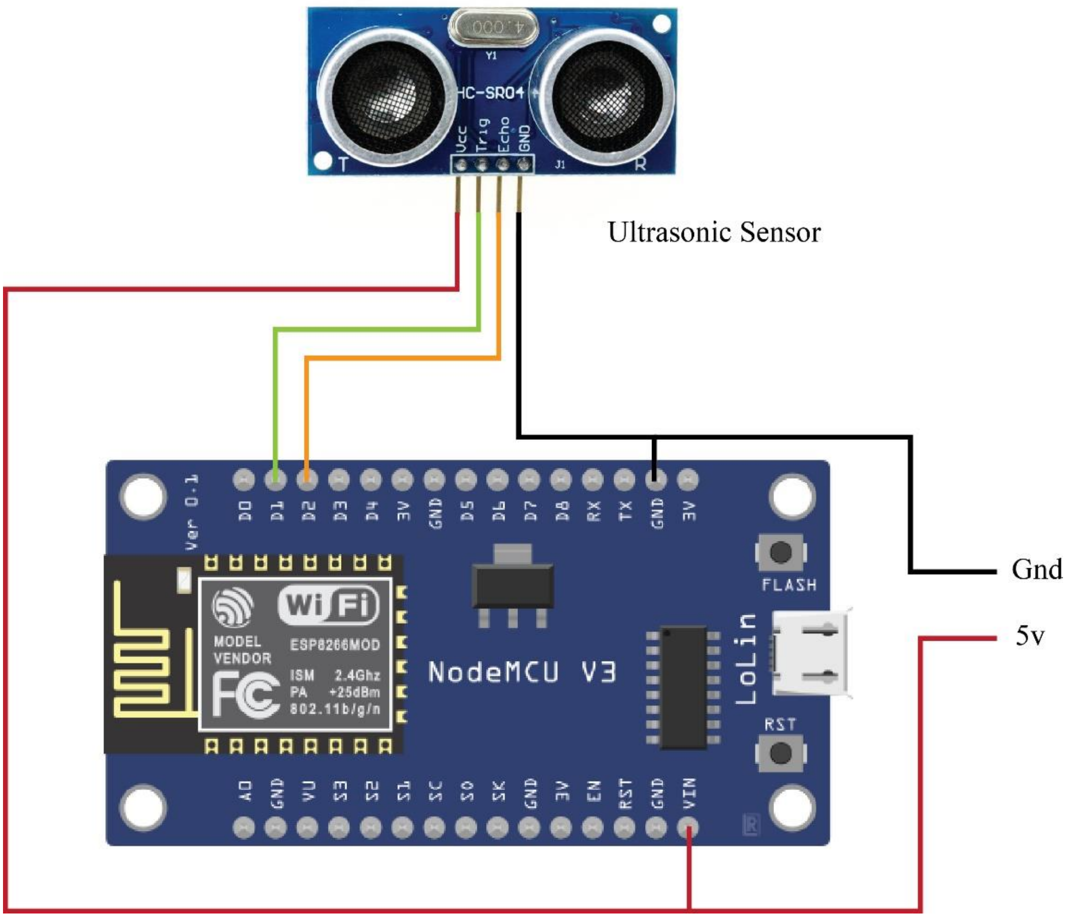


Fig 3.5: Connection between ultrasonic sensor and nodeMCU

The sensor measures the distance between the fuel level and bottom surface of the fuel tank. The sonar sensor has 4 Pin Examples: VCC, GND, TX (TRIG), and RX (ECHO). We will connect sonar sensor-1 TX Pic to the nodeMCU D1 no. Pin and RX Pin to the nodeMCU D2 no. Pin. And also connected VCC and GND Pin to the nodeMCU VCC and GND Pin. When fuel filled up in the tank, the ultrasonic sensor sends a signal to the microcontroller. At first, trig pin was sent an ultrasonic wave. This waves pushed back to the ECHO pin of sensors. Microcontroller processed the signal and converted into distance. From this distance, we calculated the level of fuel in the tank. This output was showed on this to the LCD display.

3.2.3 NodeMCU

NodeMCU is an electronics platform or device, as well as the software that runs on it. ESP-8266 was used in creating interactive objects or surroundings can use electronics more accessible. NodeMCU is an open-source hardware and software platform, project, and user community that creates single-board microcontrollers and microcontroller developer for the creation of digital devices. It was operated 5v. Wi-Fi MOD was attached with ESP-8266 to transfer the data in database. This MOD connected with Wi-Fi router by mentioning the address. NodeMCU was connected to a computer via USB and then to the Arduino software platform (IDE).

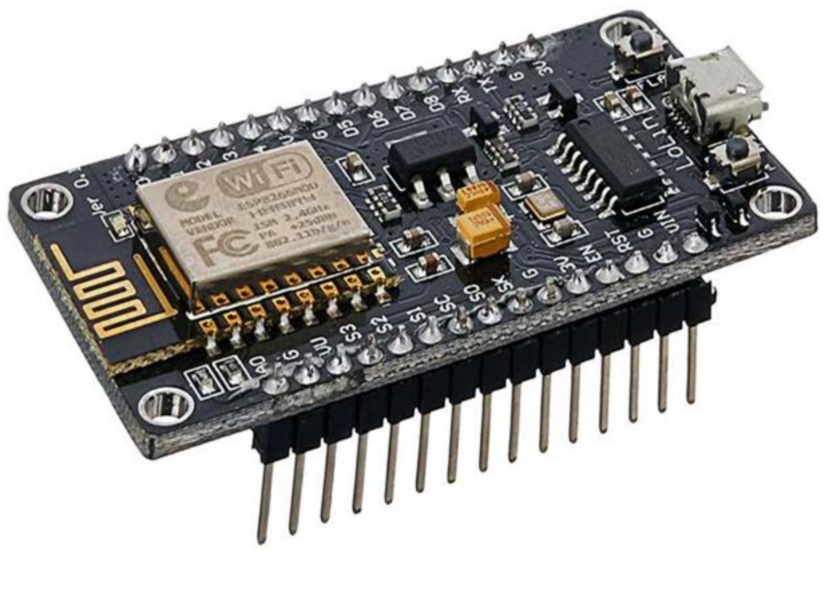


Fig 3.6: NodeMCU Controller

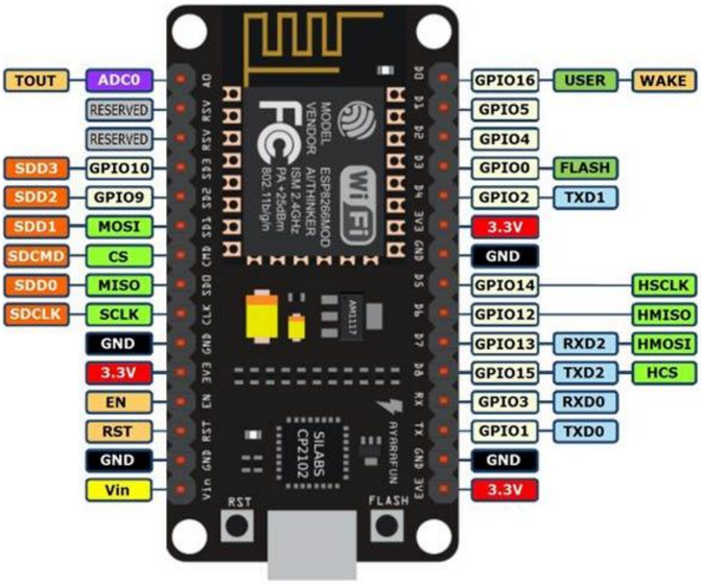


Fig 3.7: NodeMCU Pin Configuration

Specification of NodeMCU:

- Operating Voltage: 3.3V
- Input Voltage: 7-12V
- Digital I/O Pins (DIO): 16
- Analog Input Pins (ADC): 1
- Flash Memory: 4 MB
- SRAM: 64 KB
- Clock Speed: 80 MHz

Working System:

NodeMCU is the microcontroller of this project. It has some digital pin, one Rx Tx pin and one analog pin etc. This pin is also used to control input and output devices. For analog input signal, we have only used A0 pin. For digital input signal, we have used D0-D8 pins. These pins is also used as digital input/output. We have connected an external power supply for activating this controller. Here, positive point of power supply connect with vin pin and negative point connect with GND pin. All the input and output devices connect with between D0 to D8. For input signal, we have declared this pin as input and for output signal we have also declared this as output.

3.2.4 GPS Module

A GPS module is a tiny board on which a GPS sensor and other components are placed. In addition to the GPS module, a GPS receiver comprises a data display and additional components including such memory for data storage.

Fig 3.8: GPS Module

A signal was used from orbiting satellites and ground stations on Earth to identify its precise location on the planet. Microcontroller was interfaced the signal, the NEO-6M GPS receiver module uses USART communication. It was checked the coordinate of vehicle position. The distance was measured to each satellite by the amount of time it took to receive a transmitted signal. Vehicle positioned can be determined and displayed it. With distance measurements from a few more satellites.

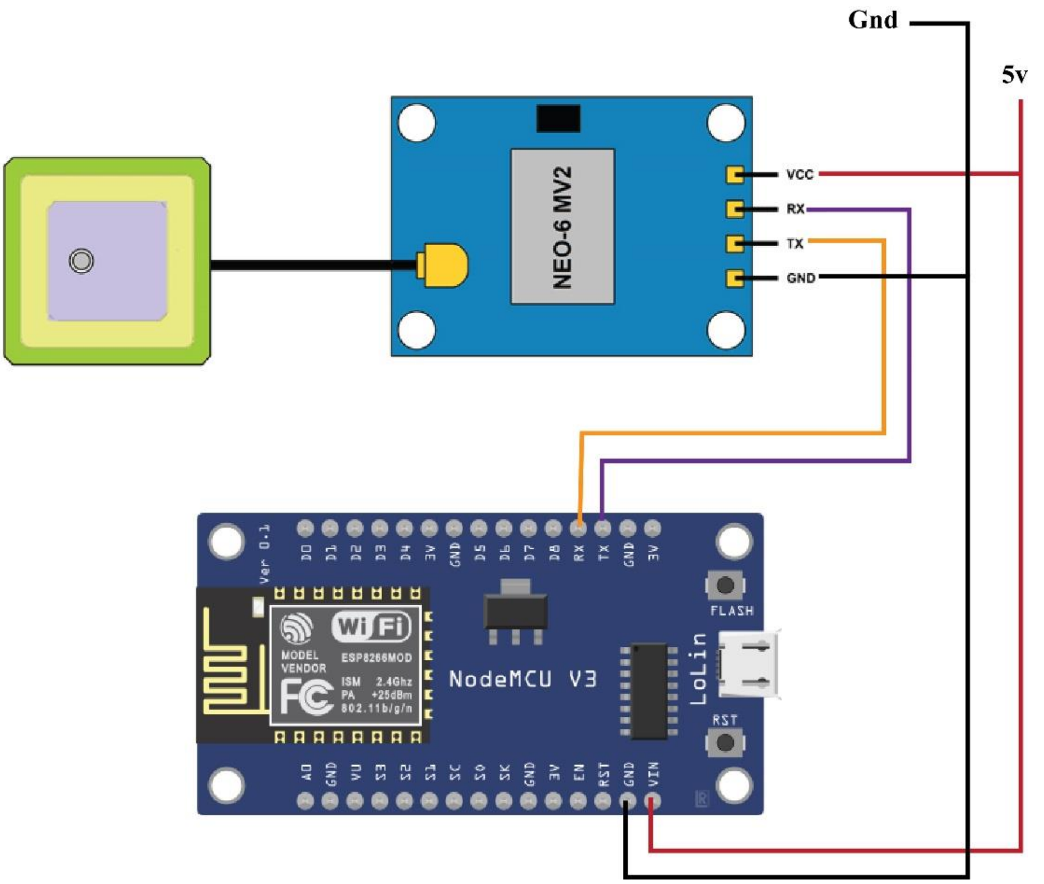


Fig 3.9: GPS Module Connection with ESP-8266

Working System:

RX and TX pin of GPS module was connected with TX and RX. Power supplied was connected from both devices so that this was module was activated properly. GPS module send data in micro-controller. It was used in vehicle to find the actual location of the vehicle. It was taken some time to initialize the device with coordinate. Then it was sent data in ESP-8266. ESP-8266 was read data and output was shown in mobile apps. So authority or owner vehicle was easily known the location of the vehicle.

Fig 3.10: GPS Setup at Vehicle

3.2.5 LCD Display

A 16x2 LCD can display 16 characters per line on each of its two lines. Each character is presented in a 5x7 pixel matrix on this LCD. The 224 distinct characters and symbols can be displayed on the 16 x 2 intelligent alphanumeric dot matrix display. Command and Data are the two registers on this LCD.

Fig 3.11: LCD Display

Working System:

LCD was connected with I2C module. It had 2 row and 16 columns. So at a 16 variables was printed on display. Display was used to show the monitoring system of the vehicle. In initializing position, staring and waiting command was shown on LCD display. Then connection ID was also shown there.

Fig 3.12: Status showing on LCD Display

3.2.6 I2C Module

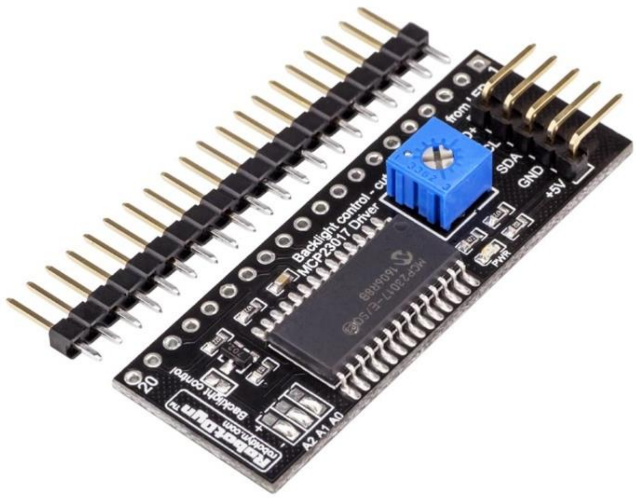


Fig 3.13: I2C Module

Working System:

I2C is a single-ended serial bus that is synchronous, multi slave, multi master packet switched. Multiple chips can be connected to the same bus, for example. Serial Data Line (SDA) and Seria Clock Line (SCL), both bidirectional open collector and open drain lines, are pulled up with resistors in I2C. SCL pin connected with D4 pin and SDA pin connected with D3 pin for buss communication. Again VCC and GND pin connected with Vin and Gnd of nodeMCU. Regulator was used to set the display brightness. This I2C module reduced the used of pin for nodeMCU. At a time, it was worked as a protector for LCD display. A2, A1 and A0 pins were used for changing the addressed of I2C. By changing the address, we connected more I2C by using one nodeMCU for buss communications.

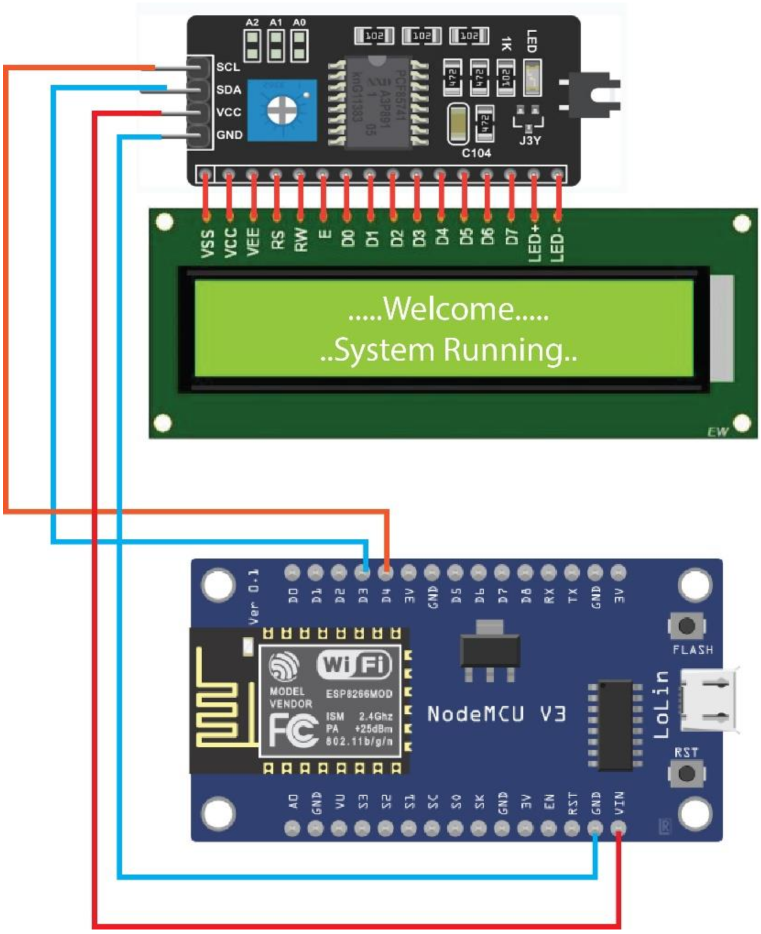


Fig 3.14: I2C and LCD Connection in ESP-8266

3.2.7 Push Button

Fig 3.15: Push Button

A Push Button was a sort of switch that operates on a basic "Push-to-Make" operation. When pressed, it remains in the off or generally open position, but when pressed, it permits current to flow through it, or we can say it completes the circuit.

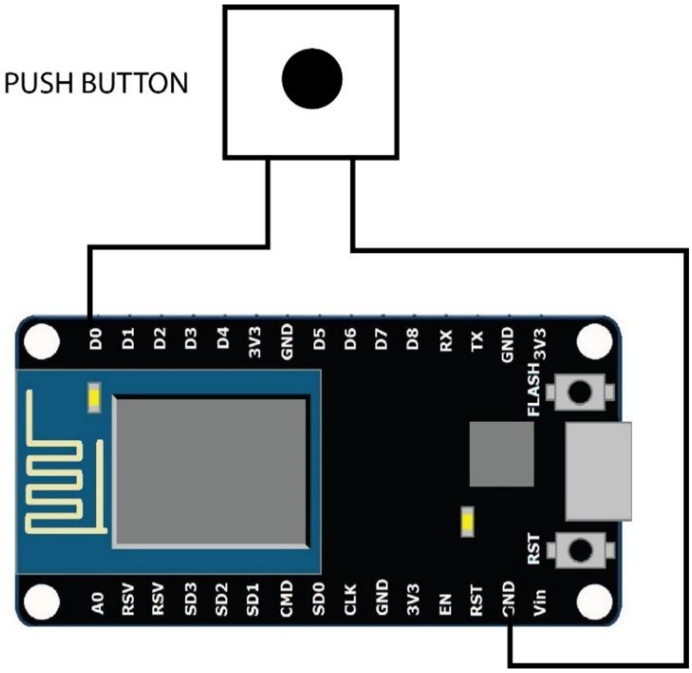


Fig 3.16: Push Button Connections

Working System:

One side of the push button was connected with D0 pins and other side was connected with GND pin of ESP-8266. It was implemented here in input pullup system. It was used to start the vehicle. At first drivers was checked the figure print. If figure was matched with authorized person, then it would be permit for starting the vehicle. Then push was pressed for starting the vehicle.

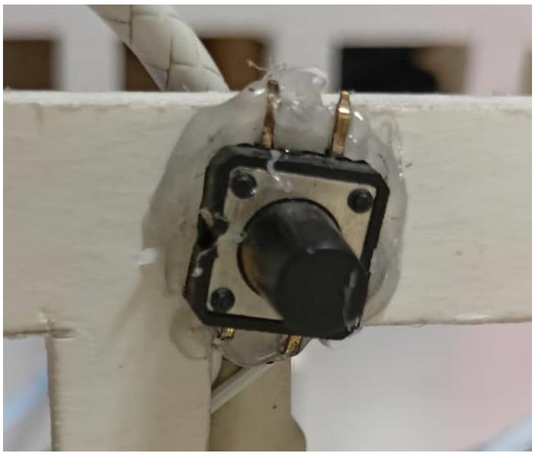


Fig 3.17: Push Button at Vehicle

3.2.8 Limit Switch

Fig 3.18: Limit Switch

Limit switch is used for controlling machinery as part of a control system, as a safety interlocks, or to count objects passing a point. It is an electromechanical device that consists of an actuator mechanically linked to a set of contacts. When an object comes into contact with the actuator, the device operates the contacts to make or break an electrical connection.

Limit switches are used in a variety of applications and environments because of their ruggedness, ease of installation, and reliability of operation. We can determine the presence or absence.

Specification:

- Actuator: Coil Spring
- Maximum Current: 10 Amp
- Housing Material: Die Cast Zinc
- Maximum AC Voltage: 300 V
- Maximum DC Voltage: 300 V
- Contact Type: Snap Action.

3.2.9 Relay Module

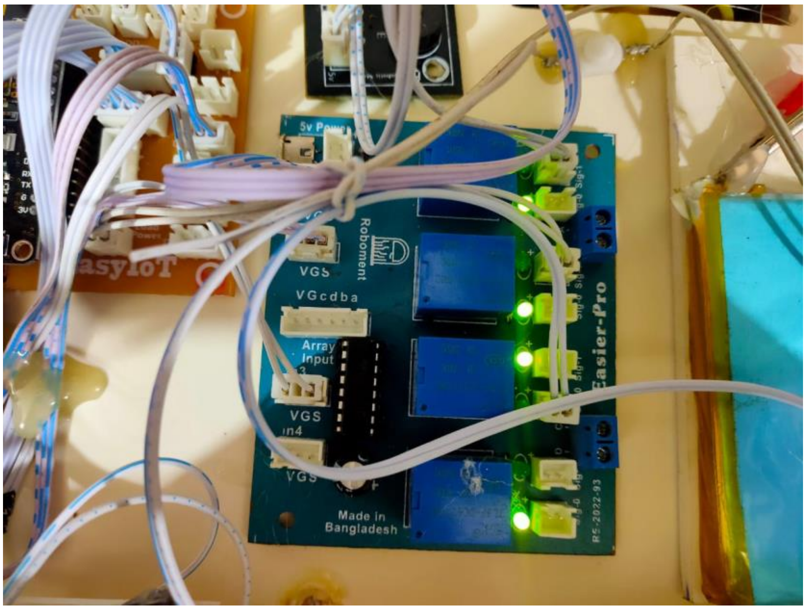


Fig 3.19: Relay Module Setup

The relay module is an electrically controlled switch that can be turned on or off, allowing or disallowing current flow. Relay module was used to drive the motor. Motor should not connect with micro-controller. Because controller could be destroyed to follow the back EMF from motor. So that reason motor was operated by using relay module. Relay switched the motor with power supply. They're made to run on low voltages like 3.3V (like the ESP32, ESP8266, and others) or 5V (like our Arduino).

3.2.10 Motor

This Yellow Dual Axis Gear Motor is perfect for a robotic automobile or a line-tracing robot. The DC gear motor is 2.5 inches long, 0.85 inches wide, and 0.7 inches thick, with a bright yellow plastic structure. It was used to show the engine starting. If motor was started so vehicle was run.

- Motor Voltage 3-9 V
- Gear Ratio: 48:1
- No-Load Current: at 3V 0.12 A
- No-Load Speed: at 3V 110 RPM

Fig 3.20: Yellow Motor

Fig 3.21: Yellow Motor Attached with Vehicle

3.3 Software Development

Fig 3.22: Arduino IDE Software

Writing code and uploading it to the board is simple with the open-source Arduino Software (IDE). Any Arduino board could be used with this software. The Arduino software was written in C++. C/C++ had the option of writing our code in inline assembler, assembler, or a combination of these languages. Although the Arduino Uno was a wonderful board for novices, it had certain restrictions. On an Arduino Uno, memory restrictions might be a concern, especially when working on larger projects like establishing an Arduino web server.

3.3.1 Introduction of Arduino

Function of Arduino IDE software was shown there. There were shown some function able portion. Menu bar, serial monitor, variable declaration, void setup, void loop, verify button and output pane were shown here. Menu was used to file save and create a new file. Verify port was used upload and compile the code. Tools port was used to select the control board and port. Error was shown to check the error.

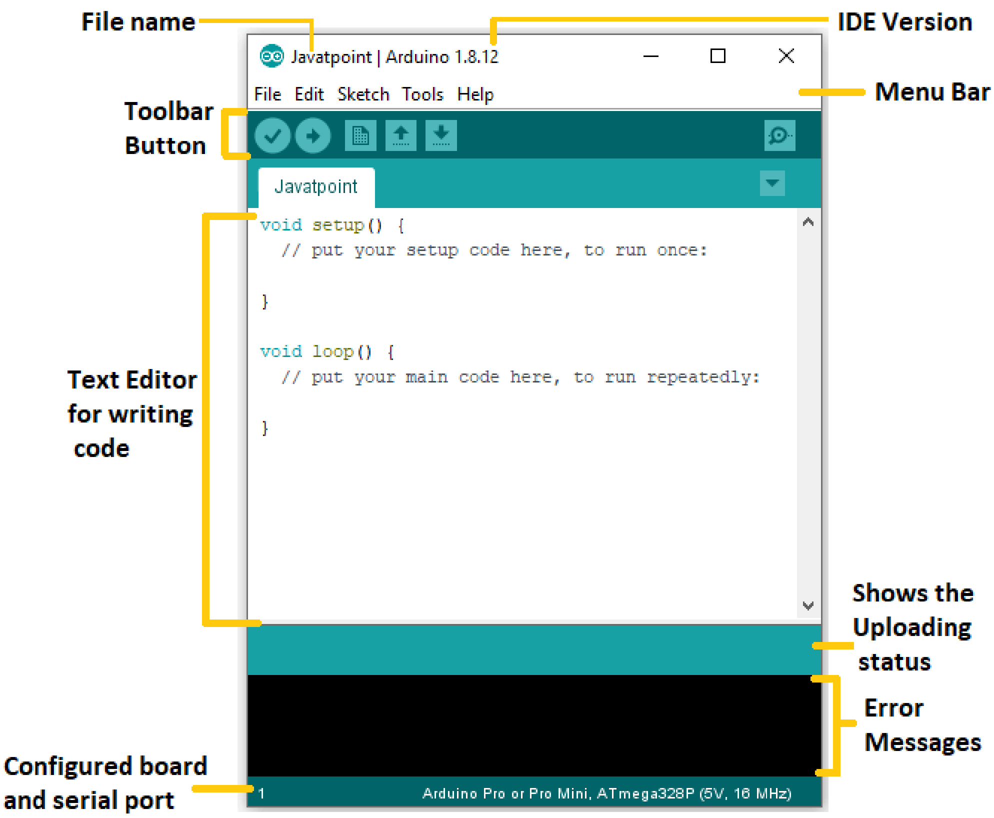


Fig 3.23: Arduino Sketch Part

3.3.2 Variables Declaration and Library Included

```
FirebaseData firebaseData,loadData;
FirebaseJson json;
int Connector=D4;

int Connector1=D0;
int StartKey=D3;
int Engine=D5;
int scaned=0;
int loadR(String field)
  {
    if (Firebase.getString(loadData, "/DUET/Biometric/"+field))
    {
      return loadData.stringData().toInt();
    }
  }

#if (defined(__AVR__) || defined(ESP8266)) && !defined(__AVR_ATmega2560__)
// For UNO and others without hardware serial, we must use software serial...
// pin #2 is IN from sensor (GREEN wire)
// pin #3 is OUT from arduino  (WHITE wire)
// Set up the serial port to use softwareserial..
```

Fig 3.24: Variable Declaration

Variables were declared in this part. Also variable was initialized because so Garber value was shown in serial print. Connector1, StartKey, Engine, scaned and Connector variables was declared as integer type data.

```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2);
#include <Adafruit_Fingerprint.h>
#include "FirebaseESP8266.h"
#include <ESP8266WiFi.h>

#define FIREBASE_HOST "iot-load.firebaseio.com"
#define FIREBASE_AUTH "Jnq5ikOjpEl0G2atw5Q6oI8wdLUi1UvtixTPQEGj"
#define WIFI_SSID "Vehicle"
#define WIFI_PASSWORD "12345678"

FirebaseData firebaseData,loadData;
FirebaseJson json;
int Connector=D4;
```

Fig 3.25: Library Included

Some library was added in this Arduino IDE software. Because this library was not included in Arduino IDE. #include was used for included the library. #include <LiquidCrystal_I2C.h> ,<Adafruit_Fingerprint.h>, "FirebaseESP8266.h" and <ESP8266WiFi.h> was included. I2C was used to control the I2C module. Besides other library was used to fully work this code.

Some variable was defined in this code. IREBASE_HOST, FIREBASE_AUTH , WIFI_SSID and WIFI_PASSWORD were defined in this code.

#define FIREBASE_HOST "iot-load.firebaseio.com"

#define FIREBASE_AUTH "Jnq5ikOjpEl0G2atw5Q6oI8wdLUi1UvtixTPQEGj"

#define WIFI_SSID "Vehicle"

#define WIFI_PASSWORD "12345678"

3.3.3 Void Setup and Loop Function

Fig 3.26: LCD Initialize and Pin Mode Functions

StartKey was included as a INPUT_PULLUP system. Engine was declared as an OUTPUT. lcd init sub function was used to initialize the LCD display. lcd was set at certain cursor. Printing the stating status for checking. Connector and Connector1declared as an OUTPUT.

WiFi.begin(WIFI_SSID, WIFI_PASSWORD);

Serial.print("Connecting to Wi-Fi");

Wi-Fi status and password were called in the setup functions.

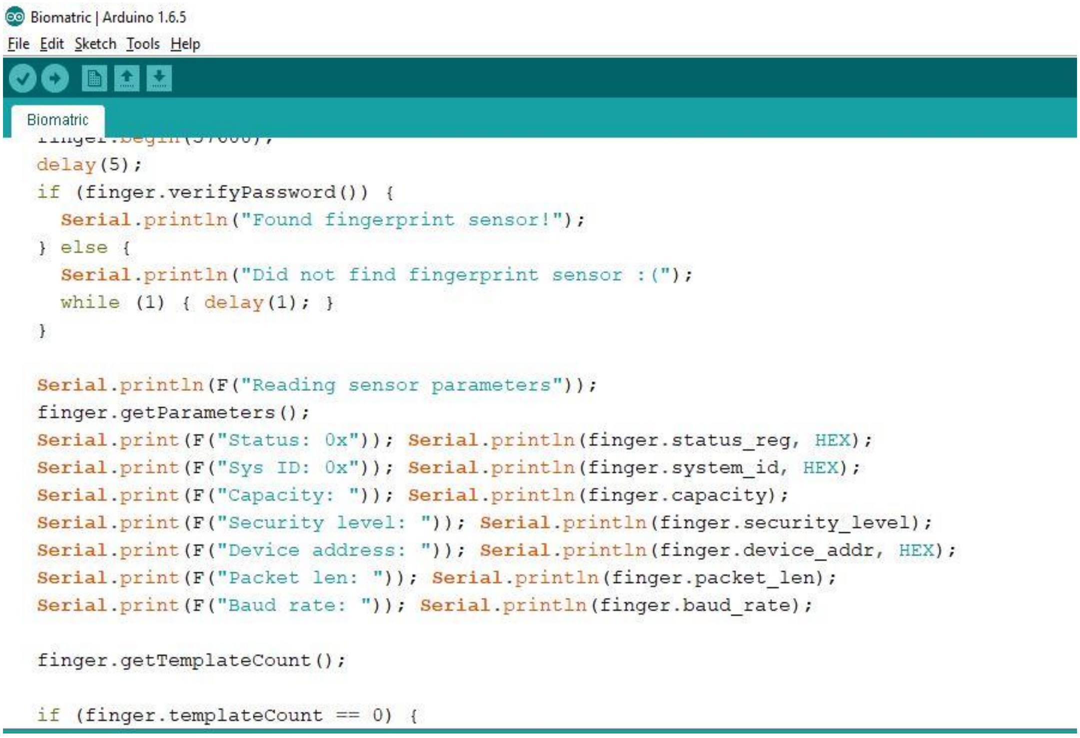


Fig 3.27: Finger Print Sensor Program

Finger print registration was checked. finger.system_id, finger.capacity, finger.security_level, finger.device_addr, finger.device_addr and finger.baud_rate were valued was shown in the serial monitor.

Firebase.setString(firebaseData, "/ Biometric/Permission","0");

Firebase.setString(firebaseData, "/Biometric/Requested","System_Restarted");

At first Firebase address was set in this link. If the author and password was right. Then it would be checked the b URL ID and set string

Biomatric | Arduino 1.6.5

File Edit Sketch Tools Help

Biomatric

```
  else {
    Serial.println("Waiting for valid finger...");
      Serial.print("Sensor contains "); Serial.print(finger.templateCount); Serial.print
  }
  Serial.println();
  Serial.print("Connected with IP: ");
  Serial.println(WiFi.localIP());
  Serial.println();
  lcd.setCursor(0,0);
    lcd.print("Connected     ");
    lcd.setCursor(0,1);
    lcd.print(WiFi.localIP());
  delay(6000);
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
  Firebase.reconnectWiFi(true);
  Firebase.setString(firebaseData, "/DUET/Biometric/Permission","0");
  Firebase.setString(firebaseData, "/DUET/Biometric/Requested","System_Restarted");
}

void loop()                         // run over and over again
{
  if(WiFi.status() != WL_CONNECTED)
```

Fig 3.28: Firebase Declaration

Biomatric | Arduino 1.6.5

File Edit Sketch Tools Help

Biomatric

```
void loop()                         // run over and over again
{
  if(WiFi.status() != WL_CONNECTED)
  {
      while (WiFi.status() != WL_CONNECTED)
        {
     lcd.setCursor(0,0);
     lcd.print("Connecting       ");
     lcd.setCursor(0,1);
     lcd.print("to Wi-Fi        ");
     digitalWrite(Connector,1);
     Serial.print(".");
     delay(200);
     lcd.setCursor(0,1);
     lcd.print("                ");
     digitalWrite(Connector,0);
     Serial.print(".");
     delay(200);
     }
  }
```

Fig 3.29: Wi-Fi Status Checked in Loop Function

In this loop, Wi-Fi was checked by the WL_CONNECTED. Then LCD was showed the connection LCD was printed second line to Wi-Fi connect.

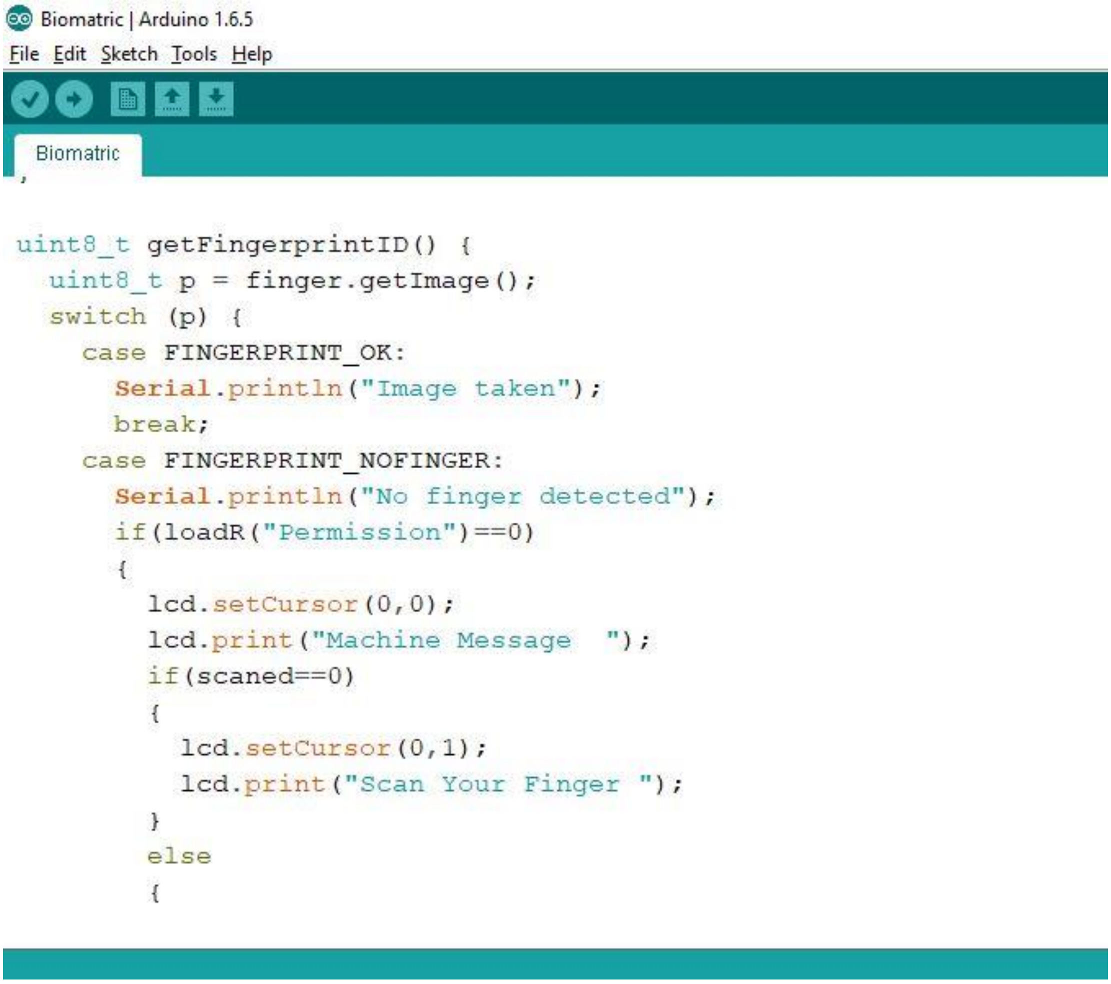


Fig 3.30: Finger Checked

Then code was checked by case function. Then image was taken from the finger print sensor.

```
if(loadR("Permission")==0)  {
        lcd.setCursor(0,1);
         lcd.print("Scan Your Finger ");
}
```

This condition was applied to check the authorized person

3.3.4 Code Compile & Upload

Fig 3.31: Compile of Arduino Program

After coding, it would be tested for checking the any compile error. If any error did not found. So code had no error. Other error was found, it was fixed by changing parameters. When compile is done, this type screen was shown in Figure 3.31.

After compiling, this code was uploaded in Arduino by using USB cable. Before uploading we select the nodeMCUboard and select the port. The system was shown in Figure 3.32.

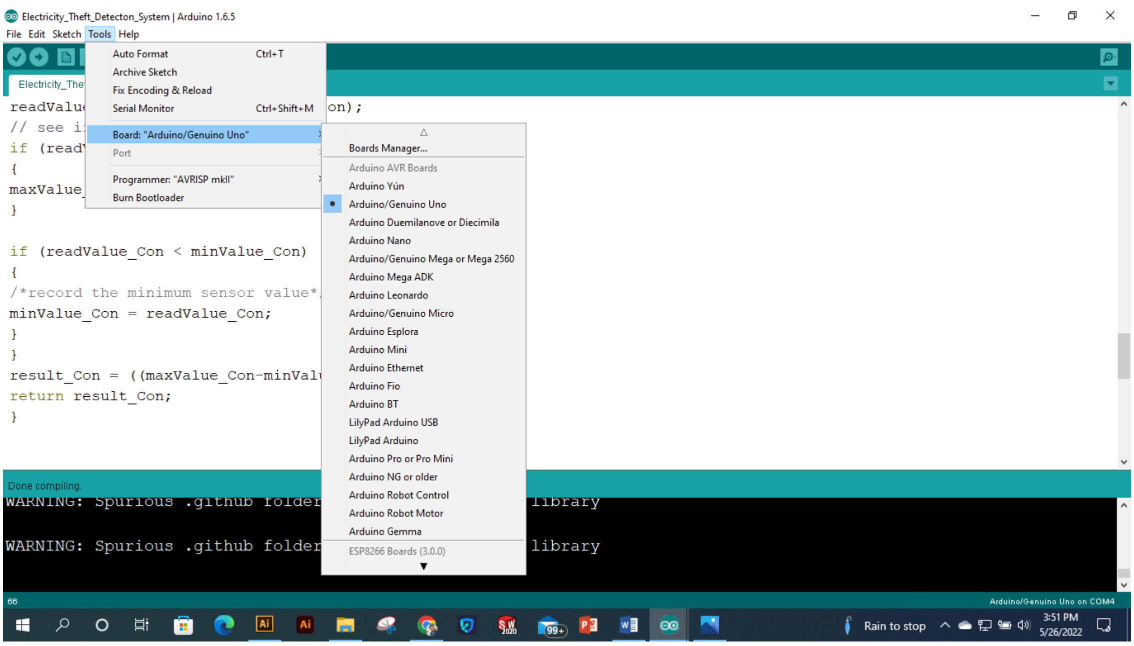


Fig 3.32: Board Selection of Arduino

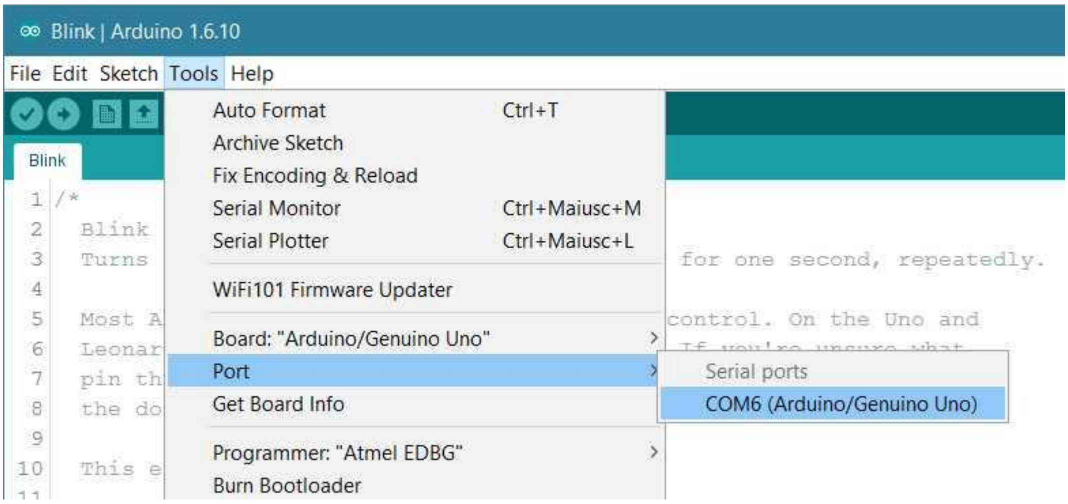


Fig3.33: Port Selection of Arduino

Port was selected by using this format. Then click the upload button for uploading code into Arduino. The procedure was almost done.

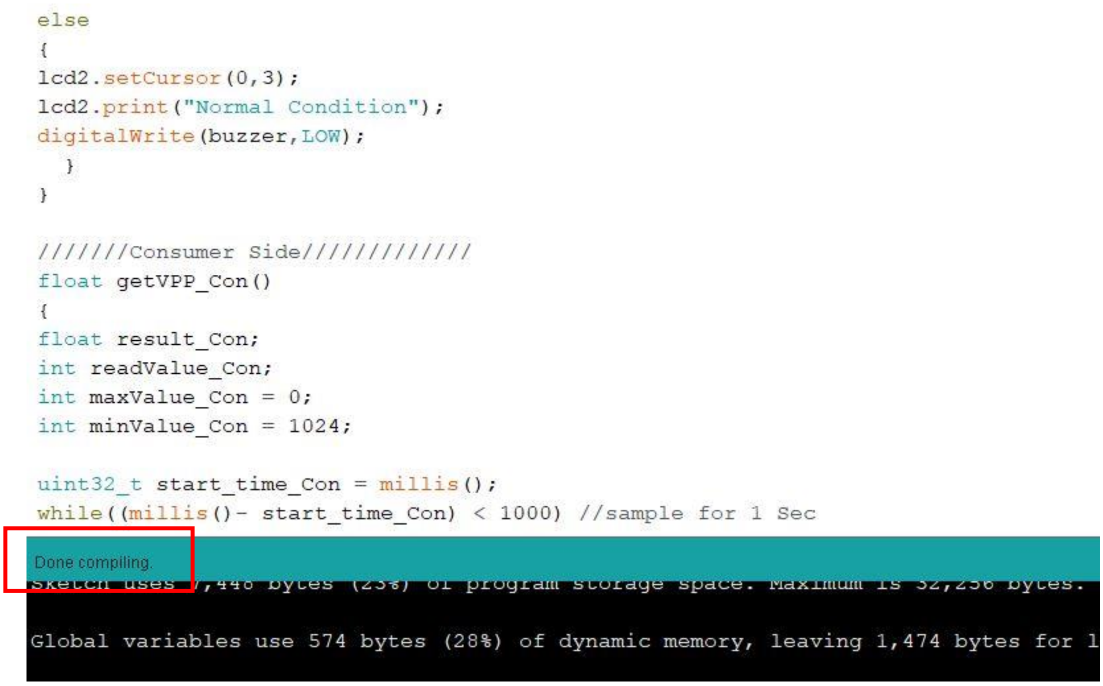


Fig 3.34: Compile Coding in Arduino IDE Software

Finally compile this code and upload this code in Arduino by using loader cable. This is the final code of Vehicle Theft Alert & Engine Lock System. Done compiling was shown in Figure 3.33 and 3.34.

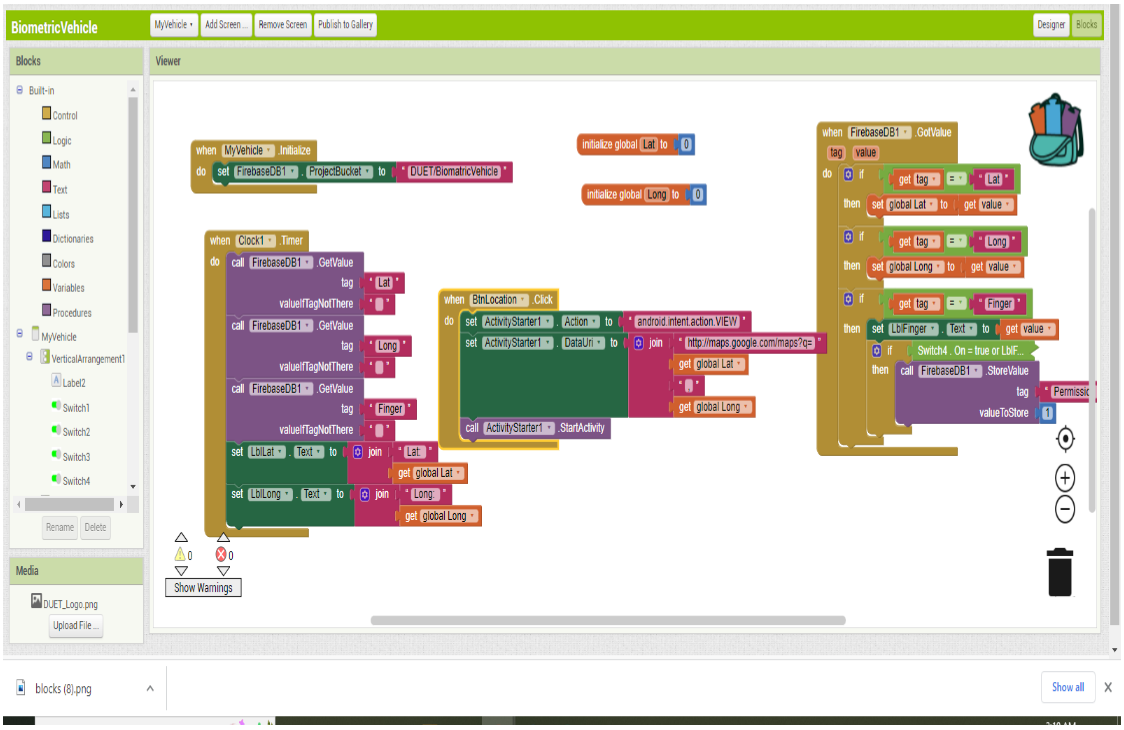
3.4 Apps Development

We develop an app for monitoring this project. It is an android app. It is a user friendly app. This app is built with help of MIT app inventor platform. It is an intuitive, visual programming environment that allows everyone to build fully functional apps for Android phones, iPhones, and Android/iOS tablets. Blocks-based coding programs inspire intellectual and creative empowerment. In this app, four parking slot are shown here. At first home page design for android apps. University logo and authority name was included here.

Fig 3.35: Front Page Development

This is the main front page of this project. This is built by block programing in MIT app inventor. All most every information's are given here. It is an IoT based vehicle safety and cloud monitoring app which is shown here in title name of this app.

This was done by using block programing methods. Some lable and tag value. Text was added by using add functions.



.

Fig 3.36: Monitoring Page Development

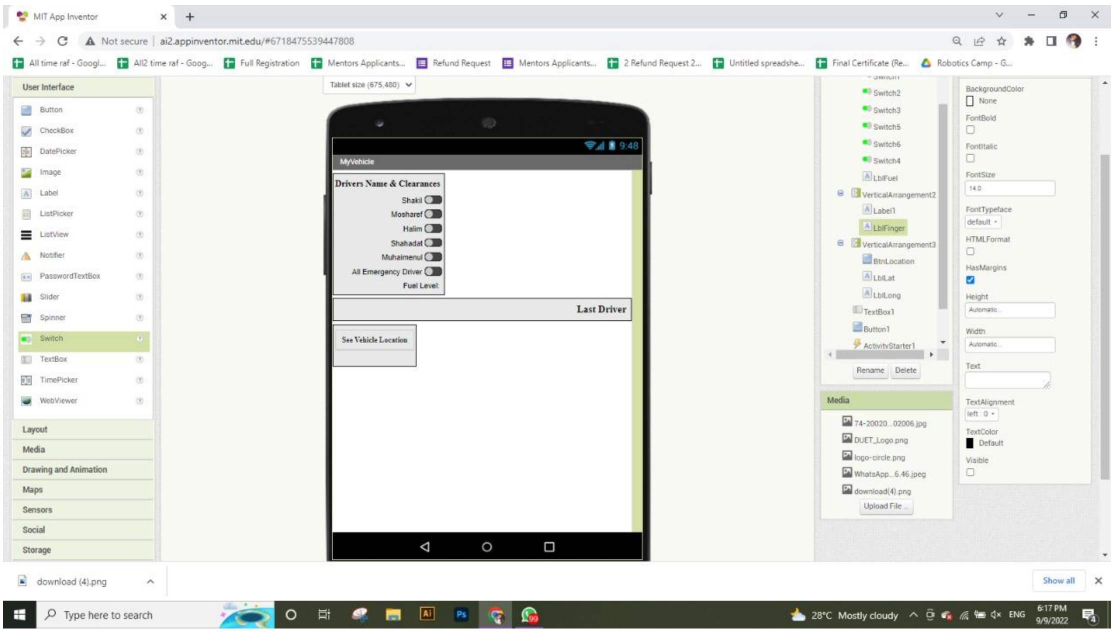


Fig 3.37: User interface of monitoring page

Second page was design by using if condition and some variable. This app was connected with database. Value was read from database and send any value into the database using second page. All notification shows in this page.

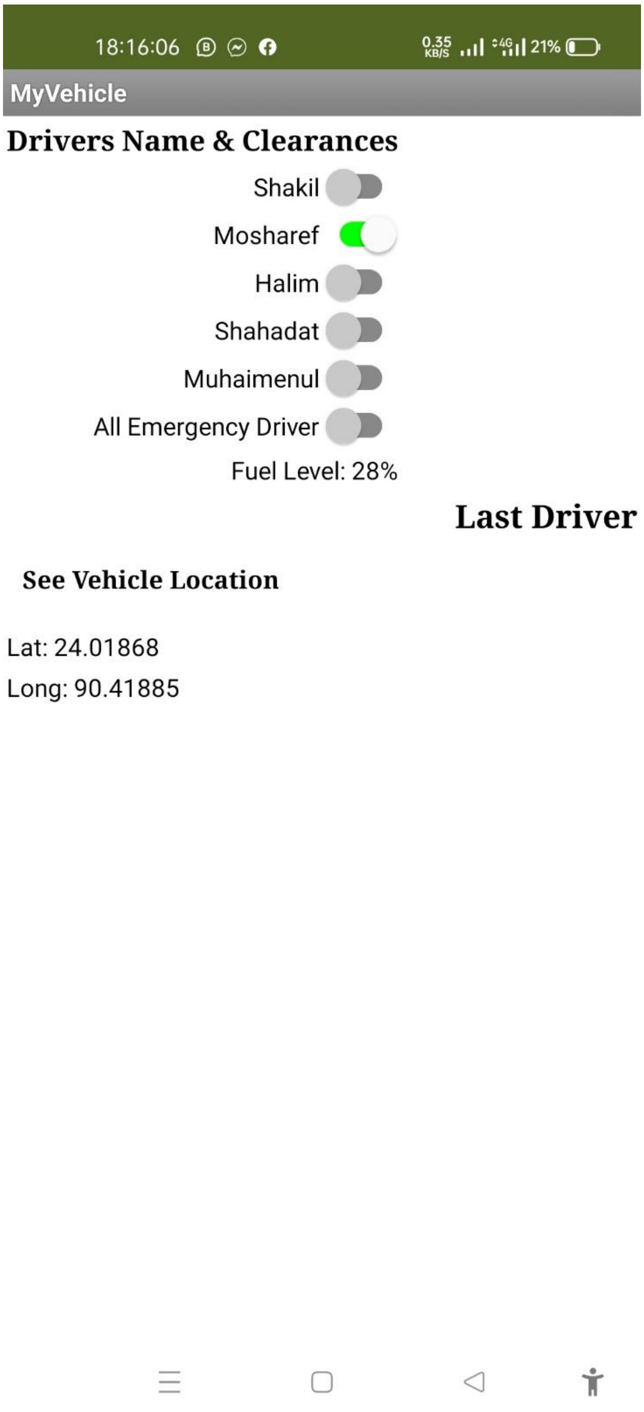


Fig 3.38: System monitoring page from mobile

CHAPTER 4

DATA COLLECTION AND ANALYSIS

4.1 Introduction

Data collection and analysis tools are defined as a series of charts, maps, and diagrams designed to collect, interpret, and present data for a wide range of applications. The choice of data collection method is a critical point in the research process. Quantitative data collection typically involves one or more of the following: Surveys, tests, or questionnaires administered in groups, one-on-one, by mail, or online, reviews of records or documents. It is the process of gathering and measuring information on variables of interest, in an established systematic fashion that enables one to answer stated research questions, test hypotheses, and evaluate outcomes. We have collected data from simulation software. Also calculated time with respect to internet speed.

4.2 Data Collection Method

Methods of quantitative data collection and analysis include questionnaires with closed-ended questions, methods of correlation and regression, mean, mode and median and others. Quantitative methods are cheaper to apply and they can be applied within shorter duration of time compared to qualitative methods. We have collected data by observing, online tracing and histories. We have collected data of ultrasonic sensor by using serial monitor in Arduino IDE software. Flash memory (program space), is where the Arduino sketch is stored. SRAM (static random access memory) is where the sketch creates and manipulates variables when it runs. We can use the serial port on any Arduino, including one with a USB connection, to send data to a computer running a terminal program like Putty, Hyperterm, TeraTerm, etc. and "log" the data to a file.

4.3 Data Collected

We have measured distance of ultrasonic sensor by using this formula.

Distance = (Time x Speed of Sound) / 2.

The "2" is in the formula because the sound has to travel back and forth. First the sound travels away from the sensor, and then it bounces off of a surface and returns back. The easy way to read the distance as centimeters is to use the formula: Centimeters = ((Microseconds / 2) / 29). For example, if it takes 100µs (microseconds) for the ultrasonic sound to bounce back, then the distance is ((100 / 2) / 29) centimeters or about 1.7 centimeters.

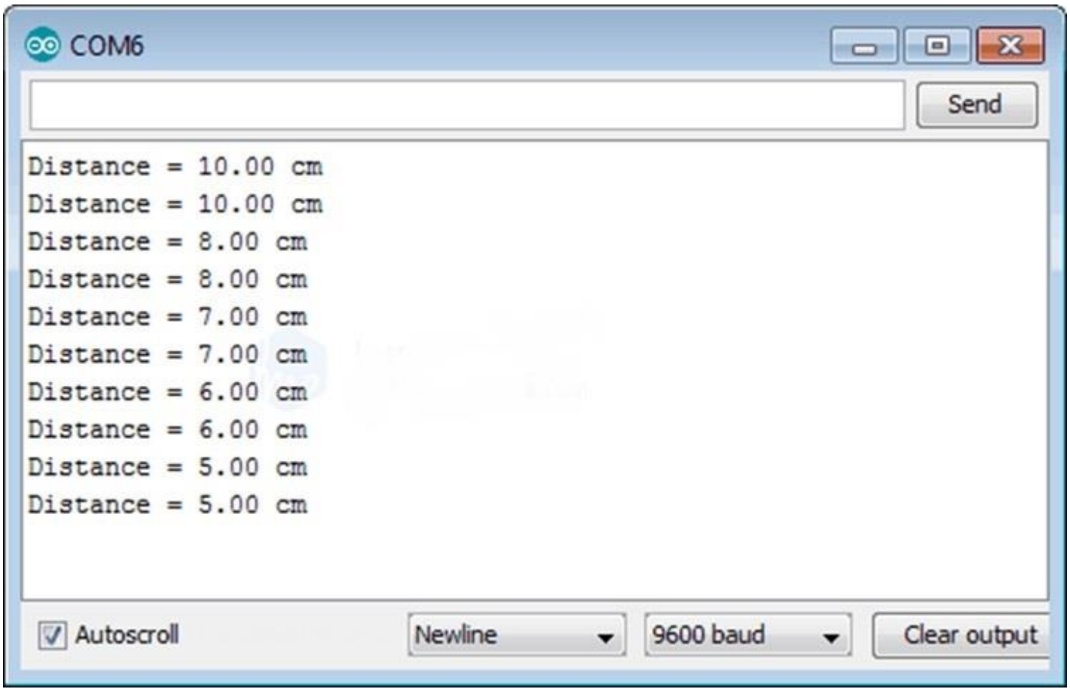


Fig 4.1: Measured the distance of fuel level in the tank

We have also collected the value of GPS module. GPS uses a lot of complex technology, but the concept is simple. The GPS receiver gets a signal from each GPS satellite. The satellites transmit the exact time the signals are sent. By subtracting the time the signal was transmitted from the time it was received, the GPS can tell how far it is from each satellite. The latitude and longitude value of each coordinated show in the serial monitor. Latitude and longitude are broken into degrees, minutes, seconds and directions, starting with latitude. For instance, an area with coordinates marked 41° 56' 54.3732" N, 87° 39' 19.2024" W would be read as 41 degrees, 56 minutes, 54.3732 seconds north; 87 degrees, 39 minutes, 19.2024 seconds west.

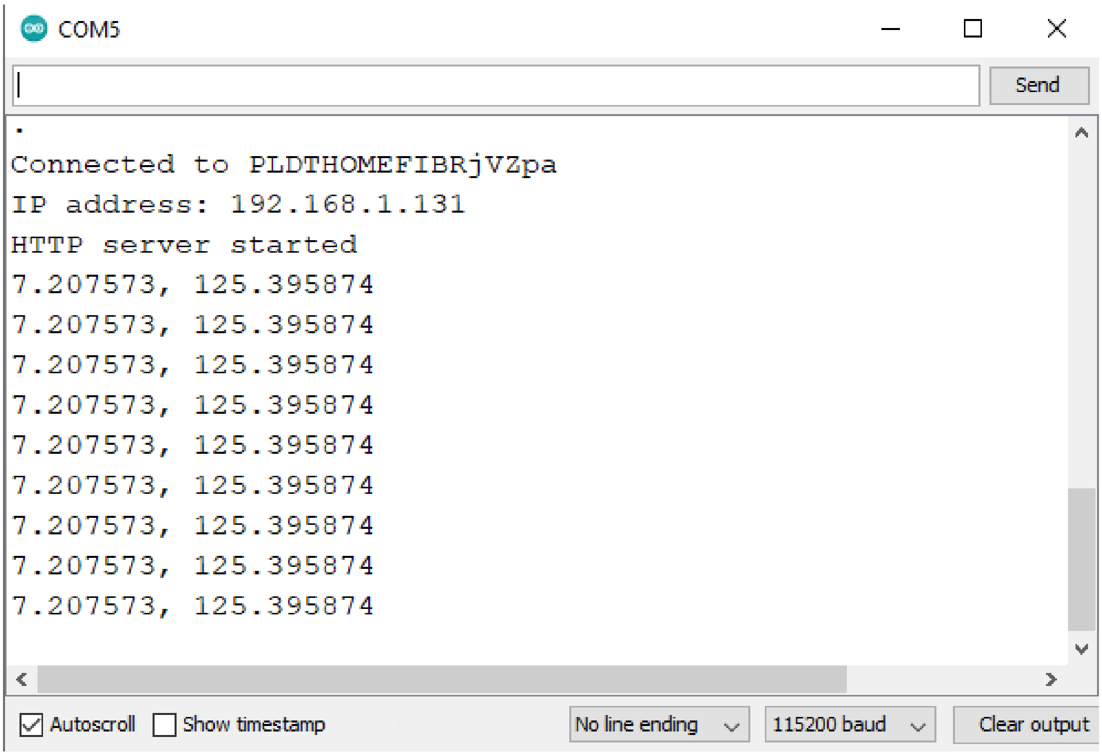


Fig 4.2: GPS coordinate value showed in serial monitor

We have collected some data which based on data transmission speed. It was taken little more delay to transfer data from nodeMCU to database or firebase. This time was depended on the 2G, 3G and 4G network speed. We collected data for 3G and 4G network which was shown in table 4.1.

Table 4.1: Operating Time Calculation for 3G and 4G network

Operating System	Time for 3G	Time for 4G
Delay for two-way data transmission	5 seconds	3 seconds
From Scanner to Database	2 seconds	1 second
Database to App	3 second	2 second
App to Microcontroller	2 seconds	1 second

We have also measured voltage and current for each components and modules. Pump and motor were gained maximum voltage and current. Others equipment's were absorbed little voltage which was around 3.3v. Average current of the components was around 30 Amp.

Table 4.2: Measured Voltage and Current for each components.

Operating Time	Voltage (V)	Current (mA)
Finger Print Sensor	3.3	50.1
GPS Module	3.28	38.6
LCD Display	3.3	20.2
I2C Module	3.3	20.6
Relay Module	3.31	90.1
Push Button	3.3	10.3
Motor	4.99	104

4.4 Data Analysis

After collecting this data, we have to analysis the data very carefully. We have done number of trial for observing this data. At first, we checked the fingerprint sensor. We have taken 100 number of trial for this experiment.

Table 4.3: Scanning Accuracy for Fingerprint Sensor

Matched	Unmatched
93	7

Accuracy = 93%

Authorized ID Detection Accuracy = 93%

Unauthorized ID Rejection Accuracy = 100%

Sometimes it was failed for the position of finger. It was very careful to set the finger on the sensor.

Table 4.4: Performance test of android app

Item	Success	Failure	Total Trial
Disable System	50	0	50
Day- 1	30	0	30
Day- 2	30	0	30
Dayr 3	30	0	30

No of Success rate of App performance for disabling system 50 trials = 100%

No of Success rate for App performance for specific driver for 30 trial each = 100%

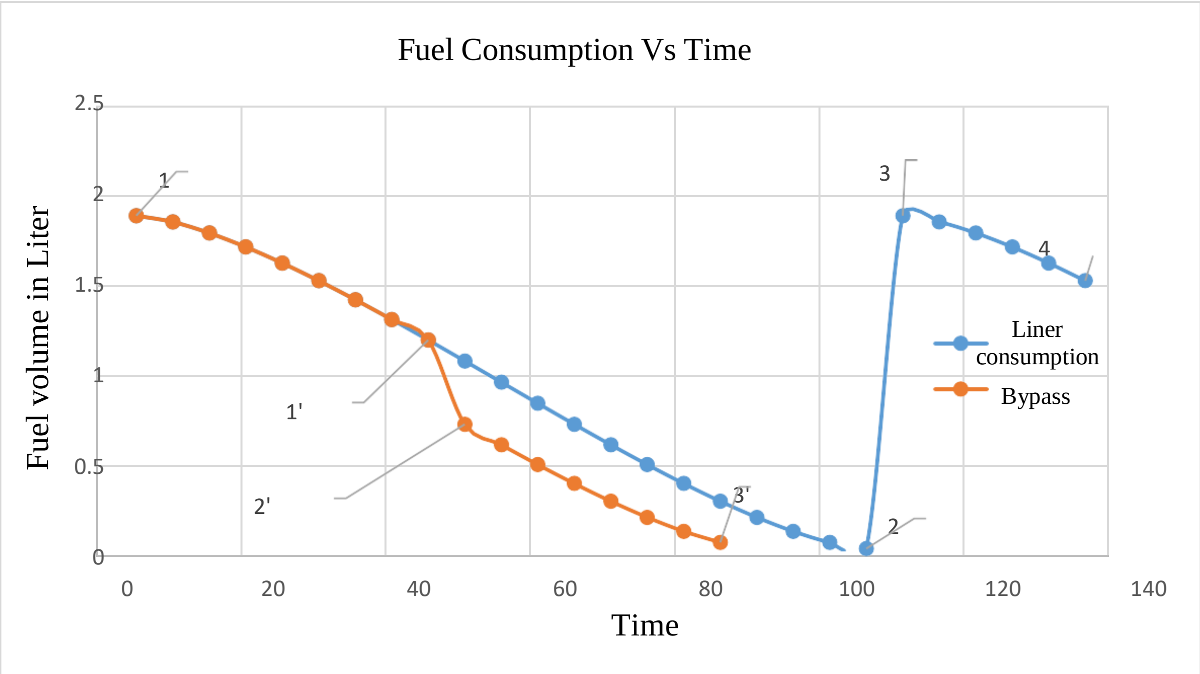


Fig 4.3: Linear and bypass Fuel consumption system

In this graph, fuel consumption rate was shown here. Blue line represented the linear consumption and orange line represented the bypass line of fuel. Anybody theft fuel through this line. Area between 1` and 2` was represented the fuel consumption rate was so much high. In this point, fuel theft was happened. So authority easily know after analyzing this graph.

CHAPTER 5
RESULT AND DISCUSSIONS

5.1 Introduction

Realizing a project physically has lots to do with research, choice of component and testing of the components. After carrying out lots of simulations on Arduino IDE, the project was implemented and tested to ensure proper operation under stated instruction. The system intelligent agents were able to communicate well and appropriate output is given under user input. The system requests for user's finger, process it and give appropriate output based on if the finger is stored in the fingerprint module or not. The android app ensures the improved vehicle security has been developed in MIT APP inventor.

5.2 Summary Result

Fig 5.1: Wi-Fi Connecting

ESP-8266 was connected with mobile data or Wi-Fi router. Then it was connected with database. This message proved that nodeMCU connected with Wi-Fi and database successfully.

Fig 5.2: Request for scanning fingerprint

Then the system requested for scanning our finger. After we gave our finger print in the fingerprint sensor. If the fingerprint was matched, then the system go to next step. If it is not matched, then result show on display.

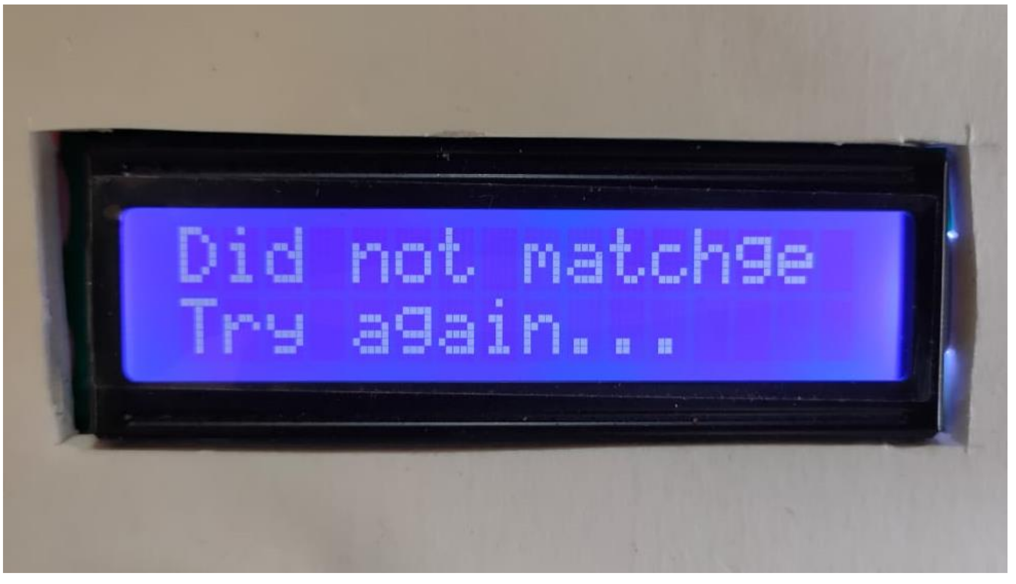


Fig 5.3: Result of wrong fingerprint condition

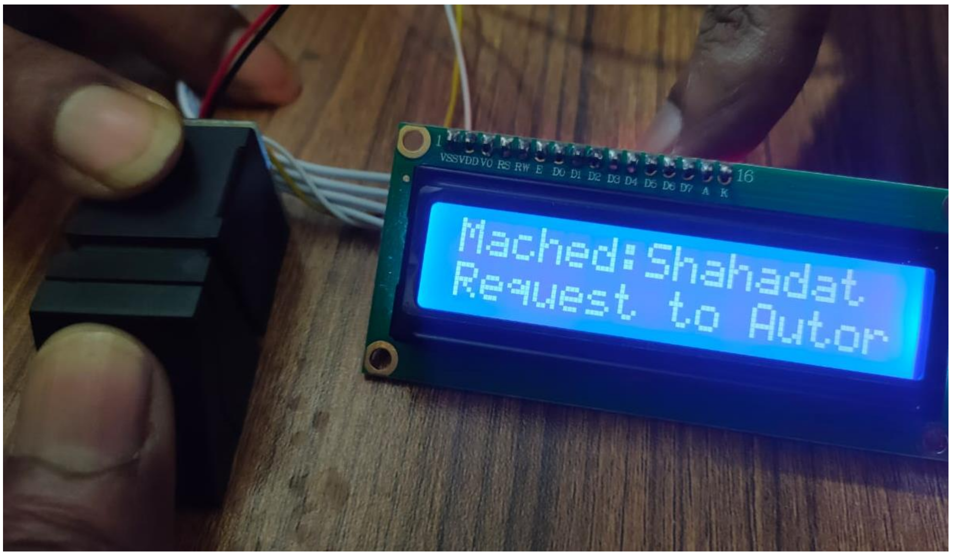


Fig 5.4: Fingerprint matched condition

When authorize person gave his/her finger print, then this result showed on the LCD display. This result was given an authentication signal for running the system.

Fig 5.5: Message request for authentication from apps

Then an authentication signal came from authority via mobile apps. This message requested us to wait for some time. In this time, author gives permission drive to start the vehicle by using push button switch.

5.3 Discussions on Result

This work is a well operating prototype of a sensor based vehicle staring system, fuel monitoring, door locking and safety system. The authorized users of the vehicles have to use fingerprint detection technique to start the vehicle. The function of tracking of the proposed vehicle security system is to monitor the Real time location with the help of GPS module. Data transmission rate for 3G system is a bit slower than 4G system. This is because the ITU-R has set standards for 4G connectivity. It is requiring all networks described as 4G to adhere to the required set of speed and connection standards. In table 4.3 we calculated the accuracy here the security is 100% protected and accurate for 100 trials authorized ID detection accuracy = 93%, unauthorized ID rejection accuracy = 100%. That means the system is accurate. In table 4.4 the app performance for disabling the system where vehicle starts only by keys is 100%. In table 4.5 app performance for specific driver is 100%. The application is highly targeted especially for those people who are not capable of leaving their private cars and reaching their desired destination in due course. The prototype of application went through several stages of software testing and evaluation.

CHAPTER 6

CONCLUSIONS & FURTHER STUDY

6.1 Conclusions

- This project describes the implementation of a vehicle security starting system based on the internet of things and has completed its whole goal of protecting vehicle starting system to improve existing auto security system based on the results acquired.
- The installed phone application aids in the control of the ignition system and prevents unauthorized persons from accessing the automobile unless they have both the ignition key and the application to enable vehicle starting.
- The system alert has also been utilized to wear the automobile owner if they attempt to use just the ignition keys. The verification shows that the IOT based advanced vehicle system is realistic and able to control the stealing automatically.
- The response time delay is also less compared to previously presented methods. This IoT based advance vehicle system enables user safety by seat belt compulsion, key less locking /unlocking system to operate the vehicles.
- The system can be used for any vehicles too by using these components and modules used in this project.
- IoT based advance vehicle system offers utmost efficiency, convenience, safety & reliability. Several tests were done and it is noticed that this system works as required and couldn't be readily hacked by unauthorized persons. This implemented method is an ideal solution for users to improve vehicle's security.

6.2 Recommendations

- RFID reader will used to start the vehicle.
- Face recognition system will be included in future.
- Web interface will be shown on the monitor.
- Float sensor will be implemented here in future.

6.3 Suggestion for Further Study

- Our project is particularly use to safeguard the vehicles. But by adding some more features we can make it use for cruise control system with the help of which we can easily make a certain limit on speed of vehicle from anywhere to avoid fatal accidents.
- As the system takes some time to send confirmation via IoT, so to minimize this delay we can use remote locking feature with our system. As user wants to go somewhere he can turn ON the system from a distance, by doing so, he gets the OTP in the meantime he reaches his / her vehicle, vehicle is ready for access.

REFERENCE

[1] Motoyoshi, Gen, Yoshiyuki Fujino, Hiromitsu Wakana, Amane Miura, and Naokazu Hamamoto. "Overall Architecture and Traffic Dynamic Control Method in the Satellite/Terrestrial Integrated Mobile Communication System." In 29th AIAA International Communications Satellite Systems Conference (ICSSC-2011), p. 8030. 2011.

[2] Umehira, Masahiro, Sunao Fujita, Zhen Gou, and Jing Wang. "Location-based centralized dynamic channel assignment with channel segregation for multi-beam mobile satellite networks." In 2014 Sixth International Conference on Wireless Communications and Signal Processing (WCSP), pp. 1-5. IEEE, 2014.

[3] Song, Hui, Sencun Zhu, and Guohong Cao. "Svats: A sensor-network-based vehicle anti-theft system." In IEEE INFOCOM 2008-The 27th Conference on Computer Communications, pp. 2128-2136. IEEE, 2008.

[4] Alsayaydeh, Jamil Abedalrahim Jamil, Win Adiyansyah Indra, W. A. Y. Khang, Vadym Shkarupylo, and D. A. P. P. Jkatisan. "Development of vehicle ignition using fingerprint." ARPN Journal of Engineering and Applied Sciences 14, no. 23 (2019): 4045-4053.

[5] Folorunso, C. O., L. A. Akinyemi, A. A. Ajasa, and K. Oladipupo. "Design and development of fingerprint based car starting system." In NIEEE International Conference and Exhibition on Power and Telecommunications. 2015.

[6] Chadil, Noppadol, Apirak Russameesawang, and Phongsak Keeratiwintakorn. "Real-time tracking management system using GPS, GPRS and Google earth." In 2008 5th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, vol. 1, pp. 393-396. IEEE, 2008.

[7] Shibghatullah, Abdul S., Abdurrahman Jalil, Mohd H. Abd Wahab, Joseph Ng Poh Soon, Kasthuri Subaramaniam, and Tillal Eldabi. "Vehicle Tracking Application Based on Real Time Traffic." (2022).

[8] Claude, TUYISENGE Jean, Ishimwe Viviane, Iradukunda Jean Paul, and Mukanyiligira Didacienne. "Development of Security Starting System for Vehicles Based on IoT." In 2021 International Conference on Information Technology (ICIT), pp. 505-510. IEEE, 2021.

[9] Elngar, Ahmed A., and Mohammed Kayed. "Vehicle security systems using face recognition based on internet of things." Open Computer Science 10, no. 1 (2020): 17-29.

[10] Mukhopadhyay, Debajyoti, Megha Gupta, Tahesin Attar, Prajakta Chavan, and Vidhi Patel. "An attempt to develop an IOT based vehicle security system." In 2018 IEEE International Symposium on Smart Electronic Systems (iSES)(Formerly iNiS), pp. 195-198. IEEE, 2018.

[11] Hardesty, Larry. "The MIT roots of Google's new software." MIT News Office, Retrieved October 1 (2010): 2015.

[12] Rezwan, Sifat, Wasit Ahmed, Mahrin Alam Mahia, and Mohammad Rezaul Islam. "IoT based smart inventory management system for kitchen using weight sensors, LDR, LED, Arduino Mega and NodeMCU (ESP8266) Wi-Fi module with website and app." In 2018 Fourth International Conference on Advances in Computing, Communication & Automation (ICACCA), pp. 1-6. IEEE, 2018.

[13] Srivastava, Prakhar, Mohit Bajaj, and Ankur Singh Rana. "Overview of ESP8266 Wi-Fi module based smart irrigation system using IOT." In 2018 Fourth International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB), pp. 1-5. IEEE, 2018.

[14] Kiruthiga, N., and L. Latha. "A study of biometric approach for vehicle security system using fingerprint recognition." International Journal of Advanced Research Trends in Engineering and Technology (IJARTET) 1, no. 2 (2014).

[15] Parihar, Yogendra Singh. "Internet of things and nodemcu." Journal of Emerging Technologies and Innovative Research 6, no. 6 (2019): 1085.

APPENDIX

Codes of the Project

```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2);
#include <Adafruit_Fingerprint.h>
#include "FirebaseESP8266.h"
#include <ESP8266WiFi.h>
#define FIREBASE_HOST "iot-load.firebaseio.com"
#define FIREBASE_AUTH "Jnq5ikOjpEl0G2atw5Q6oI8wdLUi1UvtixTPQEGj"
#define WIFI_SSID "Vehicle"
#define WIFI_PASSWORD "12345678"

FirebaseData firebaseData,loadData;
FirebaseJson json;
int Connector=D4;
int Connector1=D0;
int StartKey=D3;
int Engine=D5;
int scaned=0;
int loadR(String field)
  {
    if (Firebase.getString(loadData, "/DUET/Biometric/"+field))
    {
      return loadData.stringData().toInt();
    }
  }

#if (defined(__AVR__) || defined(ESP8266)) && !defined(__AVR_ATmega2560__)
// For UNO and others without hardware serial, we must use software serial...
// pin #2 is IN from sensor (GREEN wire)
// pin #3 is OUT from arduino  (WHITE wire)
// Set up the serial port to use softwareserial..
SoftwareSerial mySerial(D7, D6);

#else
// On Leonardo/M0/etc, others with hardware serial, use hardware serial!
// #0 is green wire, #1 is white
#define mySerial Serial1
#endif
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);
void setup()
{
  pinMode(StartKey,INPUT_PULLUP);
```

```
pinMode(Engine,OUTPUT);
lcd.init();
lcd.clear();
lcd.backlight();
lcd.setCursor(0,0);
lcd.print("....Starting....");
lcd.setCursor(0,1);
lcd.print("Please Wait...");
delay(3000);
Serial.begin(9600);
pinMode(Connector,OUTPUT);
pinMode(Connector1,OUTPUT);
WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
Serial.print("Connecting to Wi-Fi");
while (WiFi.status() != WL_CONNECTED)
{
  lcd.setCursor(0,0);
  lcd.print("Connecting       ");
  lcd.setCursor(0,1);
  lcd.print("to Wi-Fi       ");
  digitalWrite(Connector,1);
  Serial.print(".");
  delay(200);
  lcd.setCursor(0,1);
  lcd.print("                ");
  digitalWrite(Connector,0);
  Serial.print(".");
  delay(200);
}
while (!Serial);  // For Yun/Leo/Micro/Zero/...
delay(100);
Serial.println("\n\nAdafruit finger detect test");

// set the data rate for the sensor serial port
finger.begin(57600);
delay(5);
if (finger.verifyPassword()) {
  Serial.println("Found fingerprint sensor!");
} else {
  Serial.println("Did not find fingerprint sensor :(");
  while (1) { delay(1); }
}
Serial.println(F("Reading sensor parameters"));
finger.getParameters();
```

```
  Serial.print(F("Status: 0x")); Serial.println(finger.status_reg, HEX);
  Serial.print(F("Sys ID: 0x")); Serial.println(finger.system_id, HEX);
  Serial.print(F("Capacity: ")); Serial.println(finger.capacity);
  Serial.print(F("Security level: ")); Serial.println(finger.security_level);
  Serial.print(F("Device address: ")); Serial.println(finger.device_addr, HEX);
  Serial.print(F("Packet len: ")); Serial.println(finger.packet_len);
  Serial.print(F("Baud rate: ")); Serial.println(finger.baud_rate);
  finger.getTemplateCount();
  if (finger.templateCount == 0) {
    Serial.print("Sensor doesn't contain any fingerprint data. Please run the 'enroll'
example.");
  }
  else {
    Serial.println("Waiting for valid finger...");
      Serial.print("Sensor contains "); Serial.print(finger.templateCount);
Serial.println(" templates");
  }
  Serial.println();
  Serial.print("Connected with IP: ");
  Serial.println(WiFi.localIP());
  Serial.println();
  lcd.setCursor(0,0);
    lcd.print("Connected     ");
    lcd.setCursor(0,1);
    lcd.print(WiFi.localIP());
  delay(6000);
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
  Firebase.reconnectWiFi(true);
  Firebase.setString(firebaseData, "/DUET/Biometric/Permission","0");
  Firebase.setString(firebaseData,
"/DUET/Biometric/Requested","System_Restarted");
}
void loop()                    // run over and over again
{
  if(WiFi.status() != WL_CONNECTED)
  {
      while (WiFi.status() != WL_CONNECTED)
        {
    lcd.setCursor(0,0);
    lcd.print("Connecting      ");
    lcd.setCursor(0,1);
    lcd.print("to Wi-Fi      ");
    digitalWrite(Connector,1);
    Serial.print(".");
```

```
    delay(200);
    lcd.setCursor(0,1);
    lcd.print("                ");
    digitalWrite(Connector,0);
    Serial.print(".");
    delay(200);
    }
  }
  getFingerprintID();
  delay(50);            //don't ned to run this at full speed.
}
uint8_t getFingerprintID() {
  uint8_t p = finger.getImage();
  switch (p) {
    case FINGERPRINT_OK:
      Serial.println("Image taken");
      break;
    case FINGERPRINT_NOFINGER:
      Serial.println("No finger detected");
      if(loadR("Permission")==0)
      {
        lcd.setCursor(0,0);
        lcd.print("Machine Message  ");
        if(scaned==0)
        {
          lcd.setCursor(0,1);
          lcd.print("Scan Your Finger ");
        }
        else
        {
                lcd.setCursor(0,1);
          lcd.print("Autonticating....");
        }
      }
      else
      {
    lcd.setCursor(0,0);
    lcd.print("Machine Message  ");
    lcd.setCursor(0,1);
    lcd.print("Have Safe Driving");
    if(digitalRead(StartKey)==0)
      {
        digitalWrite(Engine,1);
      }
```

```
    }
    return p;
  case FINGERPRINT_PACKETRECIEVEERR:
    Serial.println("Communication error");
    return p;
  case FINGERPRINT_IMAGEFAIL:
    Serial.println("Imaging error");
    return p;
  default:
    Serial.println("Unknown error");
    return p;
}

// OK success!
p = finger.image2Tz();
switch (p) {
  case FINGERPRINT_OK:
    Serial.println("Image converted");
    break;
  case FINGERPRINT_IMAGEMESS:
    Serial.println("Image too messy");
    return p;
  case FINGERPRINT_PACKETRECIEVEERR:
    Serial.println("Communication error");
    return p;
  case FINGERPRINT_FEATUREFAIL:
    Serial.println("Could not find fingerprint features");
    return p;
  case FINGERPRINT_INVALIDIMAGE:
    Serial.println("Could not find fingerprint features");
    return p;
  default:
    Serial.println("Unknown error");
    return p;
}

// OK converted!
p = finger.fingerSearch();
if (p == FINGERPRINT_OK) {
  Serial.println("Found a print match!");
} else if (p == FINGERPRINT_PACKETRECIEVEERR) {
  Serial.println("Communication error");
  return p;
} else if (p == FINGERPRINT_NOTFOUND) {
```

```
    Serial.println("Did not find a match");
    lcd.setCursor(0,0);
    lcd.print("Did not match");
    lcd.setCursor(0,1);
    lcd.print("Try again...     ");
    digitalWrite(Connector1,1);
    delay(1000);
    digitalWrite(Connector1,0);
    Firebase.setString(firebaseData, "/DUET/Biometric/Requested","Unknown
Person");
    return p;
  } else {
    Serial.println("Unknown error");
    return p;
  }

  // found a match!
  Serial.print("Found ID #"); Serial.print(finger.fingerID);
  Serial.print(" with confidence of "); Serial.println(finger.confidence);
  scaned=1;
  lcd.setCursor(0,0);
    lcd.print("Mached:");
    if(finger.fingerID==1 ||finger.fingerID==5)
    {
    lcd.setCursor(7,0);
    lcd.print("Mehedy   ");
    Firebase.setString(firebaseData, "/DUET/Biometric/Requested","Mehedy");
    }
    else if(finger.fingerID==2 ||finger.fingerID==6)
    {
    lcd.setCursor(7,0);
    lcd.print("Humaun   ");
    Firebase.setString(firebaseData, "/DUET/Biometric/Requested","Humaun");
    }
    else if(finger.fingerID==3 ||finger.fingerID==7)
    {
    lcd.setCursor(7,0);
    lcd.print("Saddam   ");
    Firebase.setString(firebaseData, "/DUET/Biometric/Requested","Saddam");
    }
    digitalWrite(Connector1,1);
    delay(50);
    digitalWrite(Connector1,0);
    delay(50);
```

```
    digitalWrite(Connector1,1);
    delay(50);
    digitalWrite(Connector1,0);
    lcd.setCursor(0,1);
    lcd.print("Request to Autor ");
    lcd.print(finger.fingerID);
  return finger.fingerID;
}
// returns -1 if failed, otherwise returns ID #
int getFingerprintIDez() {
  uint8_t p = finger.getImage();
  if (p != FINGERPRINT_OK)  return -1;
  p = finger.image2Tz();
  if (p != FINGERPRINT_OK)  return -1;
  p = finger.fingerFastSearch();
  if (p != FINGERPRINT_OK)  return -1;

  // found a match!
  Serial.print("Found ID #"); Serial.print(finger.fingerID);
  Serial.print(" with confidence of "); Serial.println(finger.confidence);

  return finger.fingerID;
}
```