DESIGN AND IMPLEMENTATION
OF
A SMART VEHICLE OPERATION MONITORING & SAFETY SYSTEM

A Project By

Md Robiul Islam	**ME2202027274**
Naziat Saruwar	**ME2202027275**
Tarikul Islam	**ME2202027253**
Md Mejbahul	**ME2202027261**
Md Rashedul Islam	**ME2202027318**

Supervised By:
Hasan Tareq Mahin
Lecturer
Department of Mechanical Engineering (SU)

Department of Mechanical Engineering

SONARGAON UNIVERSITY (SU)

JANUARY 2026

DESIGN AND IMPLEMENTATION
OF
A SMART VEHICLE OPERATION MONITORING & SAFETY SYSTEM

A Project By

Md Robiul Islam	**ME2202027274**
Naziat Saruwar	**ME2202027275**
Tarikul Islam	**ME2202027253**
Md Mejbahul	**ME2202027261**
Md Rashedul Islam	**ME2202027318**

Supervised By:
Hasan Tareq Mahin
Lecturer
Department of Mechanical Engineering (SU)

Department of Mechanical Engineering

SONARGAON UNIVERSITY (SU)

JANUARY 2026

DECLARATION

It is hereby solemnly declared that the work presented in this project report was carried out by us and has not been previously submitted to any university or organization for the award of any degree or certificate. It is hereby ensured that the work presented here does not breach any existing copyright. It is further undertaken that any loss or damage arising from a breach of the foregoing obligation shall be borne, and the university shall be fully indemnified.

Md Robiul Islam
ME2202027274

Md Mejbahul
ME2202027261

Md Rashedul Islam
ME2202027318

Tarikul Islam
ME2202027253

Naziat Saruwar
ME2202027275

Acknowledgement

At first, we would like to express our gratitude to Almighty Allah for giving us the strength and opportunity to complete our dissertation within the scheduled time successfully.

The presentation of the project paper represents the collaborative efforts and generous support of a number of individuals. The Department of Mechanical Engineering (ME), Sonargaon University (SU) provided important subject expertise and information technology services to carry out the project work successfully.

We feel proud to express our deep sense of gratitude to our reverend teacher, guide, and supervisor, for his day-to-day supervision, dexterous management, adept analysis, keen interest, optimistic counseling, and unremitting backup.

We are grateful to **Hasan Tareq Mahin**, the relevant lecturer in the Sonargaon University (SU) Department of ME, for letting us work on an intriguing notion and for resolving any issues that arose throughout its development. We express our gratitude to our administrator for his unwavering support, guidance, and invaluable advice, as well as for allowing us to collaborate on this project.

ABSTRACT

This project deals with the design and development of an IoT based embedded system for detecting the real time vehicle health condition & safety by monitoring various parameters of a vehicle based on some threshold value. In this project, an integrated system is being developed which is a proper blending of hardware (Mechanical & Electronics) and the software to monitor fuel theft, identification of fuel consumption, brake shoe/pad worn out indication as well as load status of a vehicle and transmit it to a web server to store that information along with a unique identification number, different parameters along with date and time. This helps in increasing the life span of the vehicle as well as the engine. Ended up, the owner will be notified of all of the parameter's simulated data through mobile application according to present condition of his vehicle. Using an IoT-based system, the collection of fuel consumption, brake shoe monitoring, temperature measurement, overload checking data followed by real-time analyses at a centralized location, vehicle authorities can easily track movements of their vehicles embedded with sensor devices. In this project, we propose a new system that is based on a capacitive sensor that is open source, coupled to a controller embedded in a vehicle. The function of the controller would be to provide global positioning data over the GPS module for data transfer from almost any location.

TABLE OF CONTENTS

LIST OF FIGURES

List of Abbreviations

In this project, we have used some shortcut keywords. For that reason, all the abbreviations are given below.

a. **APK –** Android Application Package
b. **Arduino –** It's an open-source electronic Platform.
c. **Buzzer –** An electrical device that makes a buzzing noise & used for signalling.
d. **EPA –** Environmental Protection Agency
e. **GPS –** Global Positioning System
f. **HTML** – Hypertext Markup Language
g. **iOS-** Internetwork Operating System.
h. **IoT-Based** – Internet of Things-based
i. **LED –** Light Emitting Diode.
j. **Mine fleet-** It is also a mobile data stream mining environment.
k. **MySQL-** My Structured Query Language (It is an Open-source relational data-based system.
l. **Node MCU** – Open-source electronics Platform**.**
m. **OBD-** On-Board Diagnosis port
n. **OBD II** – On-Board Diagnosis 2 port
o. **PC-** Personal Computer.
p. **PCB –** Printed Circuit Board.
q. **PHP** – Personal Home Page
r. **PVC Board –** Polyvinyl Chloride board.
s. **SAE –** Society of Automotive Engineers
t. **SAWUR**–Situation Awareness with Ubiquitous data mining for Road safety.
u. **USB –** Universal Serial Bus.
v. **VEDAS** – Valcom Environmental Acquisition System. (It is a mobile data stream mining environment)
w. **WAMP** – Windows, Apache, MySQL, and PHP (It is a variation of LAMP for Windows)

CHAPTER 1

INTRODUCTION

1.1 Introduction

In this busy world, everyone uses a vehicle for transportation, but due to a busy schedule, people cannot spare time for its proper maintenance. Many people periodically visit the service center for servicing and maintenance of the vehicle, but many people are not concerned about this issue due to many constraints, it may be lack of time or work overload, etc. If the manufacturing company (service provider) can monitor the health of the vehicle remotely, it would be a beneficial service for both the owners of the vehicle and the concerned company.

Along with the upgradation of technology, we need to develop the service-providing process to ensure proper maintenance of vehicles so that owners can save time and take aim to their regular work. If we can develop the relationship between mechanical and electronics, the owner gets proper information before having any phenomena happen in his vehicle around the systems.

According to relation, we have developed a system with IoT based where the hardware part consists of a microprocessor which is responsible for collecting and processing data based on various parameters from the vehicle using sensors and send to the server over a unique IP address whereas the software part which consists of a web application is responsible for receiving the data and store it in the database which can be retrieved later to create reports of the vehicle for any kind of requirements. As well as where web applications use PHP, HTML, and MySQL for receiving and storing the data. A WAMP server is used to create a dummy server for hosting the application. The health & safety report can be accessed from a centralized location and by multiuser environment using a PC, mobile phone, etc.

With the increase in the number of cars along with other modes of transport such as public transport systems, vehicles for supply chains, and two-wheelers on the road, issues like safety and fuel consumption are of utmost importance, which depend on vehicle condition, road infrastructure, and driver behavior.

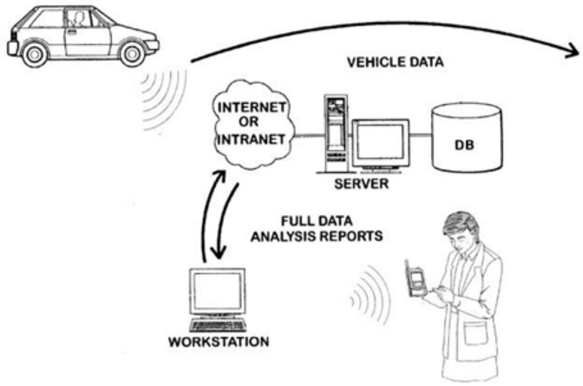


Fig 1.1: Flow diagram of data Fuel Monitoring

This project aims at developing an embedded system for detecting the vehicle condition by monitoring the internal parameters that are used in evaluating the vehicle's current health condition. These parameters are obtained using the OBD2 protocol through a port provided by the manufacturers to the vehicles.

On the other hand, despite the emergence of many novel safety features, the brake system is still the most important way of avoiding an accident or incident. It has a simple purpose, i.e., to convert the vehicle's kinetic energy into heat and to dissipate the heat efficiently, appropriate to the operating envelope of the vehicle. A challenge to the design process of brake systems is manifested by the fact that the driving conditions that a vehicle will experience throughout its life are not known. This applies to both the input conditions (e.g., pedal forces, speeds, and other factors) and the dependent parameters such as deceleration and brake pad temperature. As such, Chand et al. suggested that brake system design with respect to friction material is typically test-centric and follows an iterative development path. The testing procedures are usually designed in such a way that performance parameters are measured and recorded using data- logging equipment. Currently, much of the data- logging equipment required for brake testing has been specialized and correspondingly expensive. With a good grasp of the usage conditions, a brake performance test can be arranged to obtain the most relevant information in the fastest possible time. This suggests a need for data on the factors that underlie drivers' behavior. The lack of reliable and affordable technology to collect such data inspired the development of the system proposed in this paper.

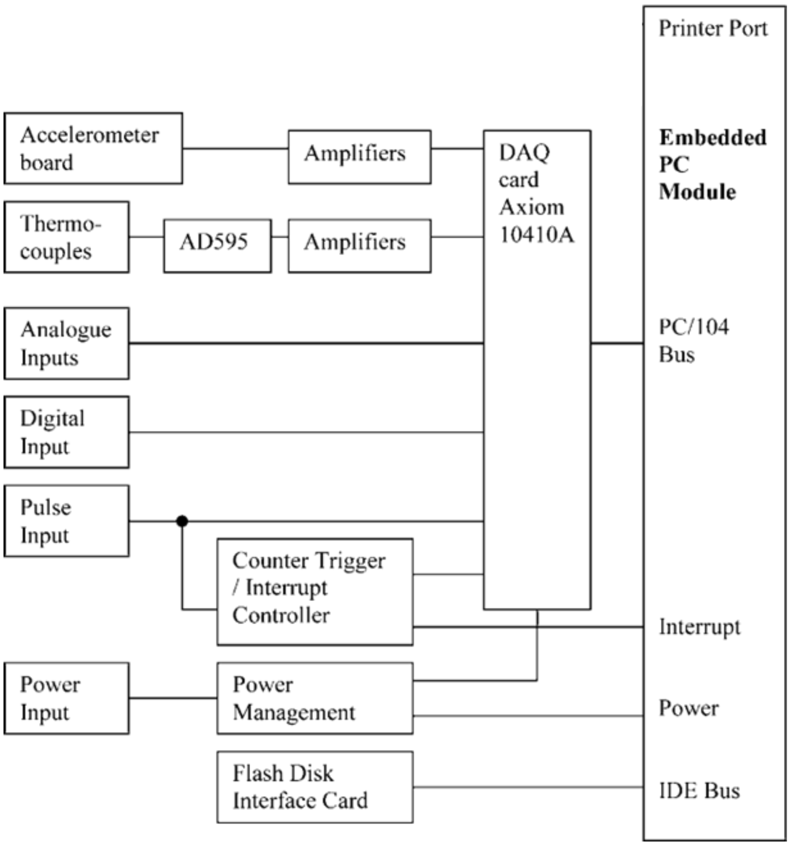


Fig 1.2: A block diagram showing the structure of the proposed system

A real-time evaluation system is being defined that can be used for rapid condition screening and provide reliable information about the vehicle conditions. This real-time evaluation system can be called the Vehicle Health Monitoring System. This system uses an HMI display so that the report and the alerts can be displayed on it, and also feedback from the user can be taken using its touch response. The system model being developed is a standalone on-board model, which will be a black box for the outside world. This model can be extended to identify and report the faults in a car to the authorized service center through wireless communication, a concept of remote diagnostics.

Since 1996, OBD systems have been incorporated into vehicles to help manufacturers meet emission standards set forth by the Clean Air Act in 1990 and the Environmental Protection Agency (EPA). The Society of Automotive Engineers (SAE) developed a set of standards and practices that regulated the development of these diagnostic systems. The SAE expanded on that set to create the OBD-II standards. The OBD-II system allows for the monitoring of most electrical systems on the vehicle. The OBD-II standard specifies the type of diagnostic connector and its pinout, the electrical signaling protocols available, and the messaging format. It also provides a candidate list of vehicle parameters to monitor, along with how to encode the data for each. As a result of this standardization, a single device can query the on-board computer(s) in any vehicle.

1.2 Objectives of the study

- To design and fabricate a system of an automatic signal system.
- To identify the accidents that occurred in the vehicle and get a signal from the vehicle to the owner.
- To measure its performance.
- To estimate the cost.

CHAPTER 2

LITERATURE REVIEW

2.1 Background

‘Vehicle health monitoring concepts and safety techniques utilize the sensor data on the vehicle, mines the data, and predicts the health, along with the safety condition of the vehicle. It provides an idea of when the vehicle would need maintenance. The model can warn when the vehicle becomes overloaded, consumed fuel into specific level, the brake shoe/pad becomes worn out, or in case a major accident occurs. The purpose of this research paper is to provide an overview of the existing ‘Vehicle health monitoring concepts and safety techniques. It discusses the different ways in which Vehicle Health can be monitored. It can be distributed or on-board data mining. Computation or mining in vehicles are restricted by less memory and processor capacity; still, onboard mining is more advantageous than a fully distributed data mining system. This paper surveys the existing models for a computationally efficient Onboard Vehicle Health monitoring system.

The vehicle comes with 60-100 sensors, which help with their effective monitoring. Digital electronic systems make it easy to constantly monitor vital engine parameters like oil pressure, coolant temperature, and exhaust emissions, etc. And report back to the driver when something is amiss. When the sensor data is monitored and mining is performed, it would be possible to predict the good health or bad health of a vehicle.

A certain organization has a huge fleet of trucks. Regular maintenance of vehicles in such a fleet is an important part of supply chain management. Many fleet management companies allow companies that rely on transportation in business to remove or minimize the risks associated with vehicle investment, improving efficiency and productivity. For this purpose, these companies collect the performance data, study the data offline, and send the vehicles to service if necessary.

2.2 Literature Survey

Many of the existing fuel management system uses one of the two approaches: One of them is discussed in. The sensor data from the vehicle is retrieved using a control module, and the parameters are uploaded to the fleet database. In the fleet database, this data is analyzed and the performance parameters of a vehicle to be diagnosed is analyzed with the normal operation of the corresponding parameters to determine whether the vehicle to be diagnosed operates outside or whether it is a normal operation. The disadvantage of this approach bandwidth available is limited; it is not possible or efficient to transmit the performance parameters to the fleet database.

Another approach is VEDAS, which works on the concept of Ubiquitous Data Mining. It is a mobile data stream mining environment. VEDAS analyses the data produced by the various sensors on the vehicle. It continuously monitors the data stream generated by a moving vehicle using an on-board computing device, identifies emerging patterns, and reports these patterns to the central control server using a low wireless network connection if necessary.

Thus, over proposed system is much more advantageous than the existing systems. It could increase functionality and reliability such that it can monitor the vehicle along with various parameters, such as fuel monitoring, fuel theft, brake shoe monitor, etc.

2.3 Focus on Study

The devices that perform analysis on vehicles are PDA. The data mining tasks are usually computationally intensive. The ways to minimize computation should be a major criterion, because of the limited processor available for our system.

To arrive at the best strategy for this problem, it is necessary to understand the techniques used to arrive at the solution. The onboard sensor collects all the data and performs partial data mining. The transfer of all data to the site would occupy more bandwidth. It is designed to do principal component analysis, incremental flourier transformation, and online linear segmentation.

CHAPTER 3

METHODOLOGY

3.1 Introduction

The methodology is the general research strategy that outlines the way in which research is to be undertaken, and among other things, identifies the methods to be used in it. These methods are described in the methodology. Methodology does not define specific methods, even though much attention is given to nature and kinds of processes to be followed in a particular procedure or to attain an objective. When properly applied to a study of methodology, such processes constitute a constructive generic framework and may therefore be broken down into sub-processes, combined, or their sequence changed. A paradigm is like a methodology in that it is also a constructive framework. In theoretical work, the development of paradigms satisfies most or all the criteria for methodology. An algorithm, like a paradigm, is also a type of constructive framework, meaning that the construction is a logical, rather than a physical, array of connected elements.

A paradigm is like a methodology in that it is also a constructive framework. In theoretical work, the development of paradigms satisfies most or all of the criteria for methodology. An algorithm, like a paradigm, is also a type of constructive framework, meaning that the construction is a logical, rather than a physical, array of connected elements.

Any description of a means of calculation of a specific result is always a description of a method and never a description of a methodology. It is thus important to avoid using methodology as a synonym for method or body of methods. Doing this shifts it away from its true epistemological meaning and reduces it to being the procedure itself, or the set of tools, or the instruments that should have been its outcome. A methodology is the design process for carrying out research or the development of a procedure and is not in itself an instrument, or method, or procedure for doing things.

Methodology and methods are not interchangeable. In recent years, however, there has been a tendency to use methodology as a "pretentious substitute for the word method.

Using methodology as a synonym for method or set of methods leads to confusion and misinterpretation and undermines the proper analysis that should go into designing research.

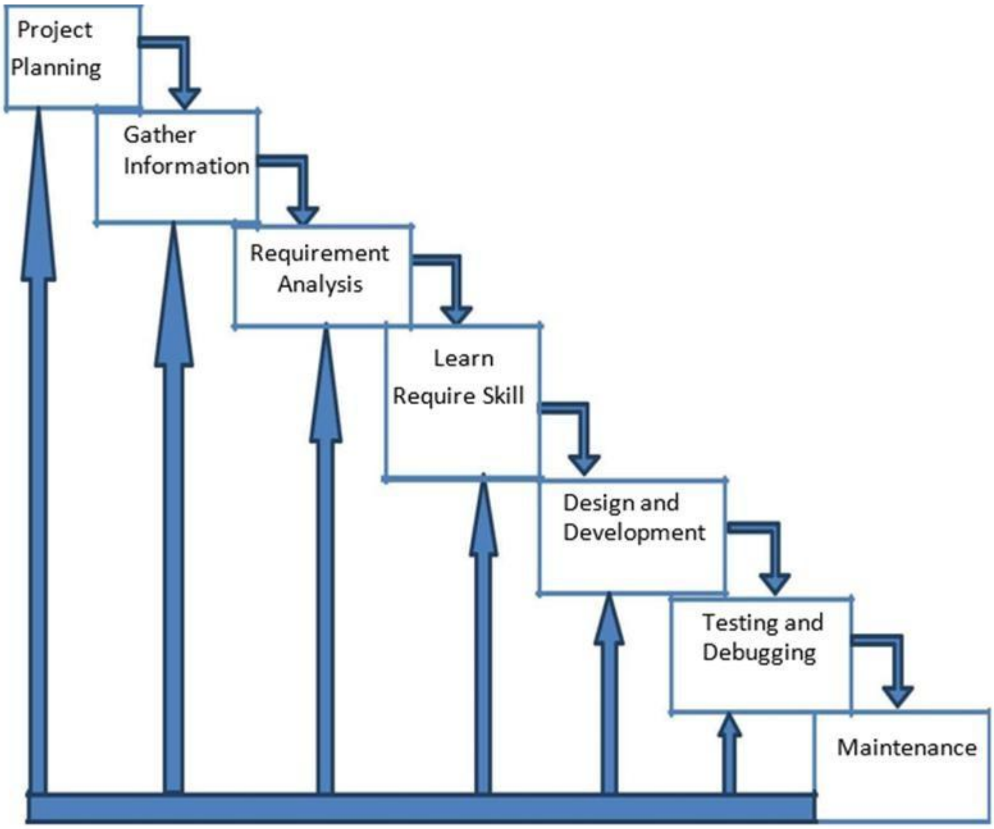


Fig 3.1: Diagram of Methodology

3.2 Justification of Methodology

We have divided this work into several parts. To solve any kind of problem or error, it should arrange the whole work in the segment so that it can provide its validity.
We have collected all possible requirements, which are very well documented, there are no ambiguous requirements. All requirements are clear and fixed.

3.3 Description of Methodology

Full work has been divided into some parts. The work is done part by part

3.4 Project Planning

Project planning is a discipline for stating how to complete a project within a certain timeframe, usually with defined stages, and with designated resources. First, we found some regular problems in our real lives. Then find this problem and think about solving it by some smart way within a certain timeframe. We discussed it with our team members and teachers. To overcome this problem, Authors have taken a plan for a smart system which can detect rpm, and send data to nodeMCU etc. It can also determine the rpm of the motor for the specific command that is needed. The device will help operators with some useful information that supports them in controlling the motor.

3.5 Project Analysis

The project analyst provides critical data support to a technical team. Research and analysis functions may include budget tracking and financial forecasting, project evaluation and monitoring, maintaining compliance with corporate and public regulations, and performing any data analysis relevant to project tasks. Authors have analyzed the need for critical and technical support in this project. Also, authors have analyzed whether this plan is possible after it is planned, its requirements in society, and how it can be applied easily.

3.6 Gather Information

After finding the problem, we gathered some information about this problem and how to solve it smartly. We read a research paper about this problem. We searched the internet to find the solutions.

3.7 Requirement Analysis

Requirements analysis encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product or project.
After gathering the information, we are thinking about requirement to make this project successful. Three sensors have been utilized in the proposed system.

3.8 Learn Required Skill

To complete the project, we had to know about the Android language Java and Hardware connection.

3.9 Hardware

Hardware is the context of technology, which refers to the physical elements. Authors have used many hardware elements such as sensors, registers, sensors are devices, modules, or devices whose purpose is to detect events or changes in their environment and send the information to other electronics, frequently a computer processor. The authors have learned about four sensors and how they work. How does it work? Where is it used?

3.10 Design and Development

System design is the process of defining the components, modules, interfaces, and data for a system to satisfy specified requirements. System development is the process of creating or altering systems, along with the processes, practices, models, and methodologies used to develop them.

Authors have designed system architecture using NodeMCU and a power supply also web interface server, Android app. To develop the device, a Methodology here maintenance some stages.

3.11 Testing and Debugging

A device under test (DUT), also known as equipment under test (EUT) and unit under test (UUT), is a manufactured product undergoing testing, either at first manufacture or later during its life cycle as part of ongoing functional testing and calibration checks. Authors have tested the device in real-life use under the test method. The authors got some results, and it works well.

3.12 Maintenance

Some issues 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 depends on IoT based embedded system, so interfacing of all hardware with Wi-Fi and the internet is a very important part in its functioning. An IoT-based counting system has been proposed in this project to count fuel consumption. Besides, a floating sensor is placed in the fuel tank. It can calculate the level of the fuel tank. From this fuel level, vehicle owners easily monitor fuel consumption. So, if anybody theft fuel from vehicles, vehicle owner will notify them through mobile app. The main advantage of this system is that it will also monitor brake shoe of vehicle. For this reason, it also decreases accidents and vehicle authorities will always monitor vehicle conditions. If in case engine temperature increases too much, it also sends notification to mobile applications. 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.

3.13 Experimental Setup

The process of circuit design can cover systems ranging from complex electronic systems all the way down to individual transistors within an integrated circuit. For simple circuits, the design process can often be done by one person without needing a planned or structured design process, but for more complex designs, teams of designers following a systematic approach with intelligently guided computer simulation are becoming increasingly common. In integrated circuit design automation, the term "circuit design" often refers to the step of the design cycle that outputs the schematics of the integrated circuit. Typically, this is the step between logic design and physical design.

For smart monitoring design, a control system circuit has been developed to monitor and secure the vehicle. A circuit diagram of the vehicle health monitoring concepts and safety techniques has been shown in the figure below.

3.14 Block Diagram

In this circuit, there are several input and output devices. Input devices are a load sensor, a fuel level indicator sensor, a brake shoe/pad worn-out indicator sensor, a vibration sensor, and GPS, etc. Output devices are a buzzer, light indicator. A USB type 5V power supply has been used for running this circuit.

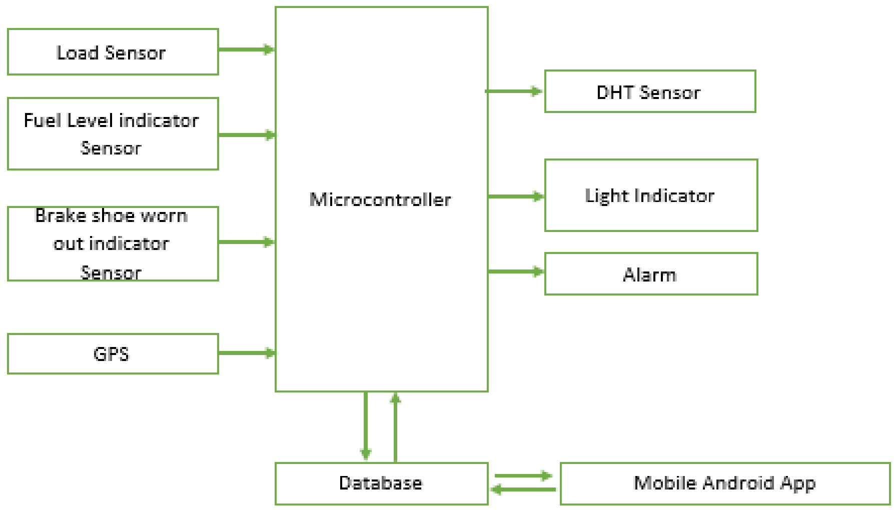


Fig 3.2: Block Diagram of the project

3.15 Working Procedure

The working procedure of this project mainly depends upon the combination of mechanical and electronic control systems. To achieve the design & fabrication of an IoT-based vehicle health monitoring concept and safety techniques, a scope with a fabricated load sensor, fuel level indicator sensor, brake shoe/pad worn out indicator sensor, vibration sensor, LCD Display, Buzzer, GPS, and connecting wire, etc.

- Fuel Level Indicator Sensor:

The fuel level indicator sensor detects the fuel level by measuring the resistivity. If there is high-level fuel in the tank, the resistance changes from high to low from high. According to the voltage dividing rule, we get an analog voltage that the Arduino Nano measures. According to the analog voltage, Arduinos nano provides required digital signal to display on the LCD monitor through as litter unit, as well as notify through mobile apps, and when the fuel level crosses a specific level (below 20 liters), then the buzzer alarms for filling up the fuel. Along with this application, fuel theft would be prevented. If the owner wants to analyze the previous record of fuel consumption from the stored database, he will also find out easily the fuel theft status.

- Load Sensitive Switch:

Load sensors are used here to detect heavy loads in the vehicle. Normally in low weight can't change state. But applying heavy weight, there is a change in state in the load sensor, which sends the signal to Arduino Nano. According to the input signal, Arduinos nano provides required voltage to the buzzer for warning, displays the data on the LCD monitor, and notifies through mobile apps, where the buzzer works as an actuator.

- Brake Shoe/Pad worn-out indicator:

To design and develop a brake shoe/pad, here are used PVC board is used, and thin wires are used to detect the thickness of brake pad. Here, the thin wires are used into brake shoe/pad in four steps within a similar distance from each other. Within the thin wire, electron flows for closing the circuit. If a thin wire becomes open in the closed circuit, the Arduino detects that the thin wire's circuit is open. According to the data, the Arduino makes an output signal to the display as a percentage through the LCD monitor, as well as notifies through mobile apps. If the brake shoe/pad is worn out and crosses the indicated level (below 25%), the arduino provides a signal to the buzzer for beeping so that the driver can be aware. On the other hand, the owner can be notified for replacing the brake shoe/pad via mobile apps as if saving extra cost and preventing accidents.

- NodeMCU:

NodeMCU is used as a microcontroller for the control unit. It converted all the analogue values converted to digital values. The fuel level sensor, pressure switch, and limit switch are connected to the microcontroller. NodeMCU is got from all the input devices and converted the data to signal. Then, send the signal in WiFi module. ESP 8266mod is used as a Wi-Fi module. It is attached with microcontroller. ESP8266 is Wi-Fi enabled system on chip module developed by Espressif system. It is mostly used for development of IoT (Internet of Things). This WiFi module sends all the signals of our base station like a Pocket router.

- Google Firebase:

All the data must be stored in a standard database. So, we are using Google firebase as database. The Firebase Real Time Database can be accessed directly from a mobile device or web browser; there's no need for an application server. First, all the input data or signal send through the Wi-Fi module. Then this data store in firebase for real time monitoring. This firebase is directly connected to our mobile application.

- Mobile Application:

An android application is developed to connect with firebase. It is a user-friendly app. From firebase, we saw our mobile how much fuel is present in the fuel tank and know that passenger's position and passenger's number. If vehicles crash into any place or contact other vehicles, a notification alarm will be sent in our mobile app. Through these apps, vehicle owners easily monitor everything.

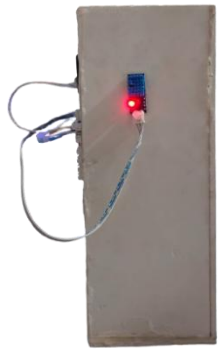
3.16 Project Preview

Fig 3.3: Experimental Setup

3.17 Component overview

Isometric View

Right Side View



Left Side View

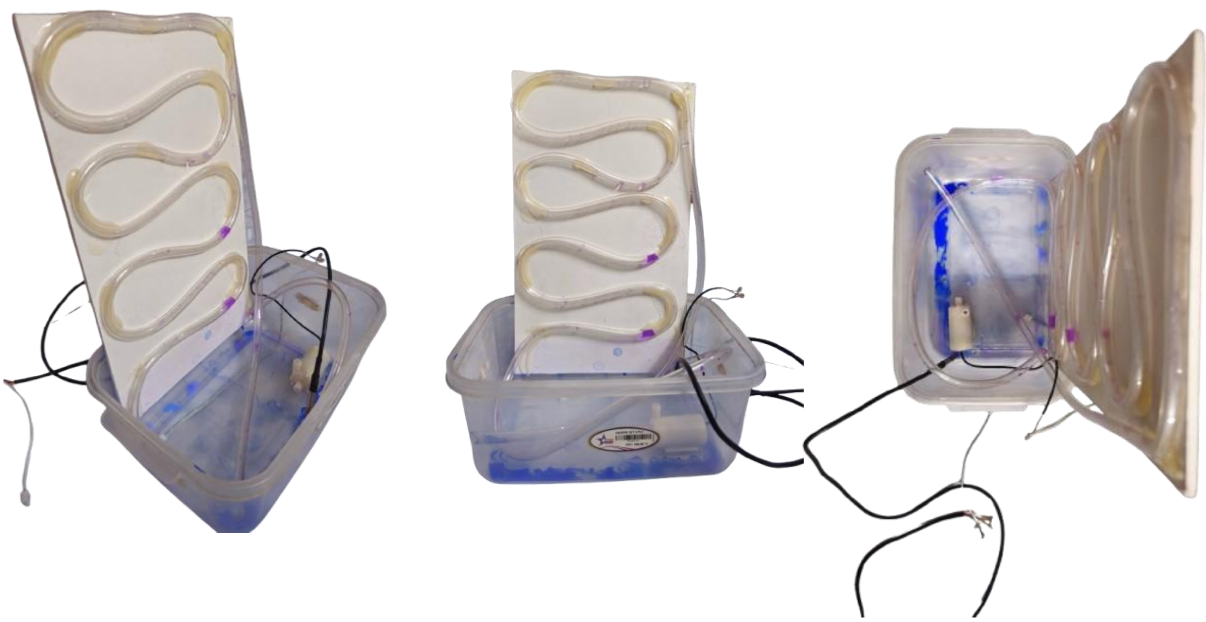
Fig 3.4: Ultrasonic Fuel Level Indicator

Isometric View

Front View

Right Side View

Fig 3.4: Prototype Vehicle with Load & Microcontroller



Isometric View Front View Top View

Fig 3.4: Prototype Vehicle Radiator with coolant, pump and Temperature sensor

Isometric View Top View

Fig 3.4: Brake Pad Model

CHAPTER 4

HARDWARE AND IMPLEMENTATION

Requirement Analysis: Requirement analysis is the technical analysis of a system project that is critical to success or failure. There are two Requirement analysis process.

- Hardware
- Software

➢ Hardware Requirements

- NodeMCU - NodeMCU is an open hardware development board through which a device is designed. It acts as a microcontroller and wi-fi device.
- Power Supply- 12v DC.
- Micro USB Cable
- PCB Board
- DHT-11 Sensor
- Load Sensor
- MQ-4 Sensor
- Connecting Wires
- DS 18B20 Temperature Sensor
- GPS Module
- Pocket Router
- Relay Module

4.1 NodeMCU

NodeMCU could be an Espressif Frameworks Wi-Fi SOC (Framing on a Chip). The ESP8266-12E Wi-Fi module serves as its basis. It's a highly integrated semiconductor that provides a whole Web network in a small package. The Arduino IDE may be used to easily modify it over the USB harborage. With simple code, you may create a Wi-Fi connection similarly to how you would with an Arduino, define input and task legs as needed, and provide a web browser. It can be used to retrieve or send information, have a web garçon or serve as a get-to-point or station. NodeMCU is an open-source, Lua-based firmware and development board specifically designed for Internet of Things (IoT) applications. It has firmware that manages grounding on the ESP-12 module and operates on the ESP8266 from Espressif Systems. It makes use of an ESP 8266mod Wi-Fi module. It is coupled to a microcontroller. It is often used to enhance the Web of Impacts (IoT). This WiFi module, which transmits all signals from the base station, looks a lot like the financial switch. Every piece of information should be stored in a standard database. For this reason, we used Firebase as the database. First and first, the WiFi module must process the supplied data or flag. For real-time observation, this data is stored on Firebase, which your phone will incontinently connect to.

Fig 4.1 ESP 8266

4.1.1 Power

The Arduino can be powered by a surface control restriction or by figuring out USB participation. The control source is properly selected. An AC-to-DC connection can also provide external (non-USB) control. You may use the DC Control Jack, the barrel jack, to control your Arduino board. Most of the time, a separator connector is connected to the barrel jack. Although the sponsor recommends maintaining the board between 7 and 12 volts, it may be powered between 5 to 20 volts. The controllers may warm up at 12 volts, and they may not function at 7 volts. The legs of the control are

i. VIN: by and large the input voltage leg of the Arduino board is utilized to supply input constraints from an exterior control source. The voltage has to be interior the degree indicated.

ii. 5V: This adherence of the Arduino board is utilized as a controlled control drive voltage and it's utilized to supply the board as well as onboard components.

iii. 3.3V: This adhere of the board is utilized to supply a drive of 3.3 V which is made from a voltage controller on the board

iv. GND: Inside the ESP- 8266 pinout, you'll find 5 GND legs, which are all associated. The GND legs are utilized to close the electrical circuit and deliver a common clarification reference position each through your circuit. Persistently make past any mistrustfulness that all GNDs (of the Arduino, peripherals, and variables) are related and have a common ground.

4.1.2 Analog IN

Only one analog pin of the ESP 8266 is used with the Analog Digital Converter (ADC). In addition to being used as analog inputs, these pins can also be used as digital inputs or outputs.

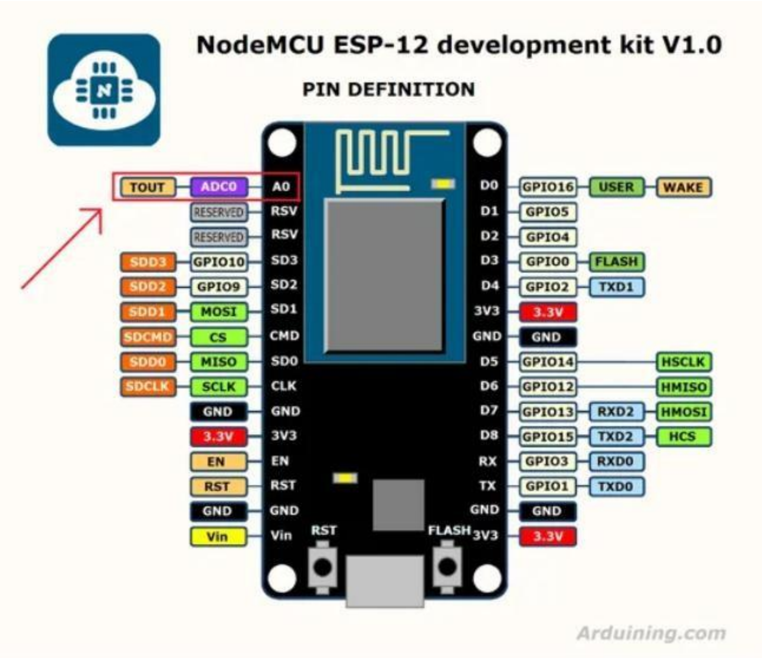


Fig 4.2 Pinout-Analog

4.1.3 Pinout – Digital Pins

Legs 0- 13 of the ESP 8266 serve as advanced input/ issue legs.

Leg 13 of the ESP 8266 is associated with the erected-in Driven.

Within the ESP 8266- legs have PWM capability.

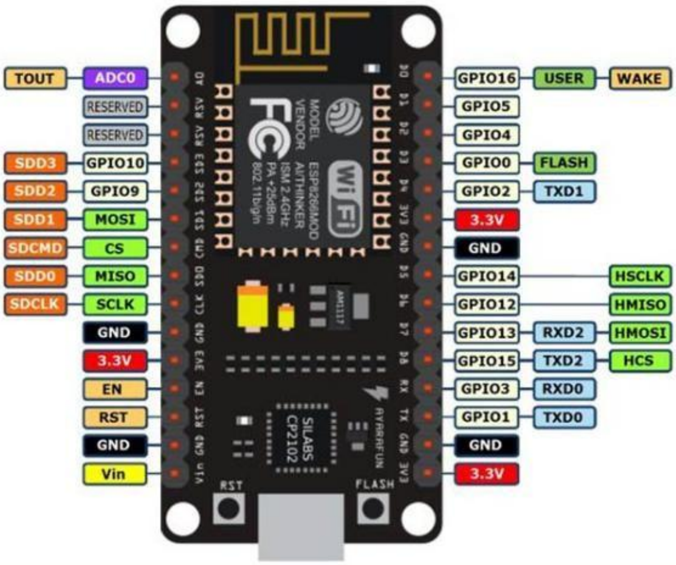


Fig 4.3 Pinout - Digital Pins

It's critical to note that

- Each leg can allow/sink over to 40 mA most extreme. But the suggested current is 20 mA.
- The supreme most extreme current given (or sank) from all legs together is 200 mA.

4.1.4 Others Pins

i. **Pins:** These legs are too known as UART legs. It's utilized for communication between the Arduino board and a computer or other inclination. The transmitter Leg 1 and recipient Leg are utilized to transmit and concede the information resp.

ii. **Outside Hinder Pins:** This pin of the Arduino board is utilized to create the Outside barge and it's done by Legs 2 and 3.

iii. **PWM Legs:** The periodical legs of the board are utilized to change the computerized flag into an analog by changing the run of the palpitation. The Legs and 11 are utilized as a PWM leg.

iv. **SPI Legs:** This can be the periodical supplemental Interface leg, it's used to preserve SPI communication with the assistance of the SPI library. SPI legs incorporate.

v. **SS:** Leg 10 is utilized as a Slave Select.

vi. **MOSI**: Stick 11 is utilized as an Ace Out Slave In.

vii. **MISO:** Leg 12 is utilized as a Ace In Slave Out.

viii. **SCK:** Pin 13 is utilized as a periodical timepiece

ix. **LED Pin:** The board has an inbuilt Driven utilizing computerized leg- 13. The Driven shines as it were when the advanced leg gets to be tall.

x. **AREF Leg:** Typically an analog reference leg of the Arduino board. It's utilized to provide a reference voltage from an outside control drive.

4.1.5 Memory

Microcontroller	Family	Architecture	Flash	SRAM	EEPEROM
ESP 8266	AVR	Harvard	4 MB	64kb	1kb

Table 3.1 Memory of ESP-8266

4.1.6 Specifications

- Voltage: 3.3V.
- Soft-AP and Wi-Fi Direct (P2P).
- Current use ranges from 10 µA to 170 mA.
- Flash memory attachable: up to 16MB (512K is standard).
- Processor clocked at 80–160MHz.
- RAM: 32K + 80K.
- The number of GPIOs is 17 (multiplexed with other operations).
- One input from analog to digital with a resolution of 1024 steps.
- +19.5dBm is the 802.11b mode output power.
- Support for 802.11: b/g/n.
- Five TCP connections can be active at once.

4.2 Ultrasonic Sensor

The Ultrasonic HC-SR04 Sensor Module is a widely used sensor for measuring distance and detecting objects in a variety of applications. It is shown in Figure 3.9. It works in the same way that a radar system does. Ultrasonic sensors operate by producing high-frequency sound waves that humans can't hear. The transmitter emits a high-frequency sound pulse, which is received by the receiver when it reflects from any object's surface.

Fig 4.4 Ultrasonic Sonar Sensor

The sensors detect objects in this manner. It can detect objects or measure distances between 2cm and 400cm. The ultrasonic sensor works in the same way that a bat's object-detecting system does. We can also claim that it works in the same way that a radar system does. The Transmitter (TX) and Receiver (RX) are the two primary components of the ultrasonic or HC-SR04 module (RX).

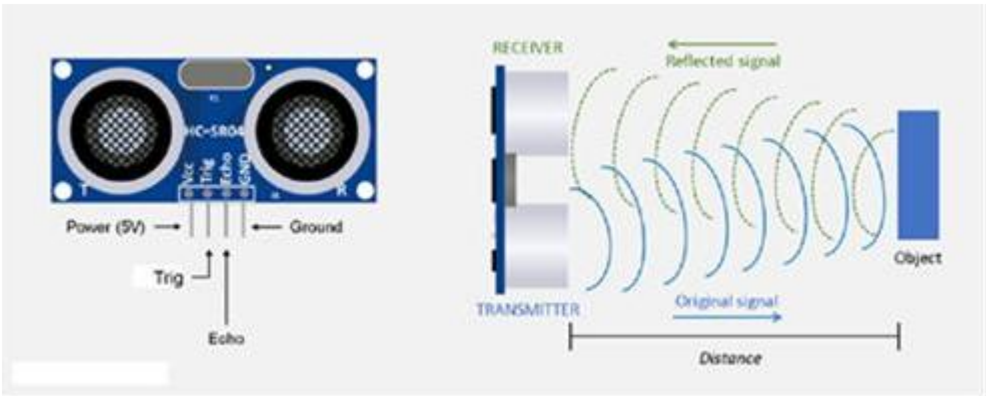


Fig 4.5 Distance measured by using ultrasonic sound

4.2.1 Operating Condition

- Quiescent Current: <2mA
- Working Current: 15mA
- Effectual Angle: <15°
- Ranging Distance: 2cm – 400 cm/1″ – 13ft
- Resolution: 0.3 cm

4.2.2 Working System

Echo and Trig pin of the ultrasonic sensor at storage tank connected with D5 and D6 pin of NodeMCU. Again Echo and Trig pins of the ultrasonic sensor at the feed water tank are connected with the D7 and D8 pins of NodeMCU. The trig pin emits ultrasonic sound, which is received by the echo pin. To make the ultrasonic sound, we must first set the Trig pin to a High State for 10 seconds (microseconds). Where the distance between the sensor and the object is denoted by s. The sound speed in air is v = 0.034cm/s or 340 m/s. The time it takes for sound waves to bounce back from an object's surface is called t. Because time will be doubled when the waves travel and bounce back from the originating point, we must divide the distance value by two Power Supply − +5V DC.

4.3 Connecting Wire

JST Wire is used to establish a connection between the devices. ESP-8266 module, Rain sensor, and Ultrasonic sensor connection were done using this wire.

Fig 4.6 JST Connector [17]

4.4 Power Supply

A power supply is an electrical device that supplies electric power to an electrical load. The primary function of a power supply is to convert electric current from a source to Correct Voltage, current, and frequency to power the load. Also called a power supply unit or PSU, the component that supplies power to a computer. Most personal computers can be plugged into standard electrical outlets. The power supply then pulls the required amount of electricity and converts the AC to the DC. One purpose of a power supply is to convert AC to DC so the computer has proper power to run its components. Another purpose of a power supply is to distribute proper DC voltage to each component. Several cables with connectors come out of the power supply.

Fig 4.7 Power Supply 12v 2amp

Specifications:

- Input: 100~240V AC, 50/60Hz
- Output: 12V DC, 2000mA
- Plug: US plug
- Cable length: 100cm
- Net weight: 128g
- Package size: 9.5 * 7 * 6cm
- Package weight: 144g

4.5 PCB Board

A printed circuit board mechanically supports and electrically connects electronic components or electrical components using conductive tracks, pads, and other features etched from one or more sheet layers of copper laminated onto or between sheet layers of a non-conductive substrate. PCB is an acronym for printed circuit board. It is a board that has lines and pads that connect various points. A PCB allows signals and power to be routed between physical devices.

The soldier is the metal that makes the electrical connections between the surface of the PCB and the electronic components.

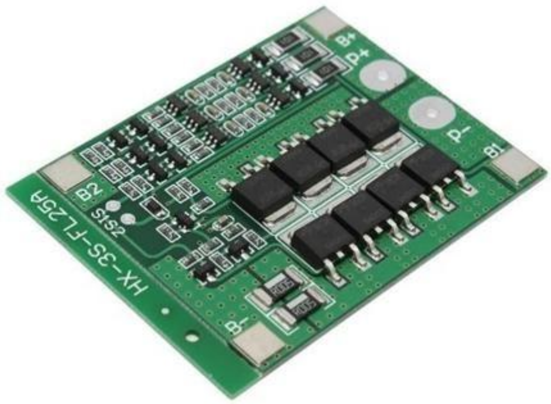


Fig 4.8 PCB Board

The substrate most commonly used in printed circuit boards is a glass fiber reinforced (fiberglass) epoxy resin with a copper foil bonded onto one or both sides. PCB made from paper-reinforced phenolic resin with a bonded copper foil are less expensive and are often used in household electrical devices.

4.6 Hall Sensor

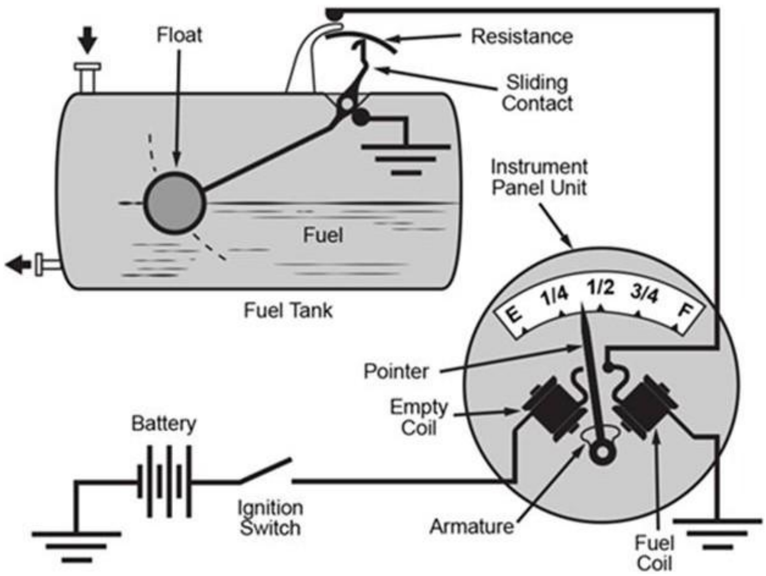


Fig 4.9 Typical Fuel Level Sensing Arrangement

A fuel level sensor (FLS) is used in all automobiles to indicate fuel level. Various methods are used to measure fuel level such as resistive film, discrete resistors, capacitive, and ultrasonic. Resistive-based sensors are most commonly used for this application. These sensors are mechanically connected to a float which moves up or down depending on the fuel level. As the float moves, the resistance of the sensor changes. This sensor is part of a current balance circuit of the fuel gauge display circuit which typically consists of coils for actuation of the display needle. As the resistance of the fuel sensor changes, the position of the needle changes proportionally to the current flowing in the coil.

Specification:

- Supply Voltage (Vs): 8 - 30 VCC @ 5 mA
- Pulse Out (Vo): 0.06 - 4.95V typical (2.5 V at zero field strength)
- Frequency Range: 0 - 30 kHz
- Sensing Distance: Depends on field strength (up to 1.0" @ 3000 Gauss
- Temp. Range: 2TE: -40° to 221° F (-40° to 105° C)

  3TE: -40° to 300° F (-40° to 150° C)

- Thread Length: 1.0" - 6.0" (25-152mm)
- Thread Sizes: 5/8-18, 11/16-24, 3/4-20, M16, M18

4.7 Load Switch

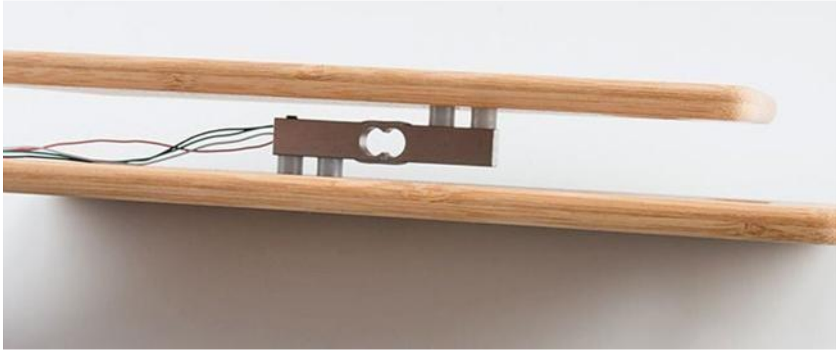


Fig 4.10 Load sensor visualization

A load sensor is a type of force transducer, and another name is a strain gauge. It converts a force such as tension, compression, pressure, or torque into an electrical signal that can be measured and standardized. As the force applied to the load cell increases, the electrical signal changes proportionally.

Besides a Strain gauge (sometimes referred to as a Strain gauge) is a sensor whose resistance varies with applied force; It converts force, pressure, tension, weight, etc., into a change in electrical resistance which can then be measured. When external forces are applied to a stationary object, stress and strain are the result. Stress is defined as the object's internal resistance forces, and strain is defined as the displacement and deformation that occur. The strain gauge is one of the most important sensors of the electrical measurement technique applied to the mechanical quantities. As their name indicates, they are used for the measurement of strain. As a technical the term "strain" consists of tensile and compressive strain, distinguished by a positive or negative sign. Thus, strain gauges can be used to pick up expansion as well as contraction.

4.8 18B20 Temperature Sensor

The DS18B20 is a type of temperature sensor and it supplies 9-bit to 12-bit readings of temperature. The communication of this sensor can be done through a one-wire bus protocol which uses one data line to communicate with an inner microprocessor.

Fig 4.10 Temperature Sensor

Specification:

- Sensor-Type: Digital, Local
- Output-Type: 1-WireR
- Voltage-Supply: 3 V ~ 5.5 V
- Features: Output Switch, Programmable Limit, Programmable Resolution
- Test-Condition: -10°C ~ 85°C (-55°C ~ 125°C)

- Operating-Temperature: -55°C ~ 125°C
- Interface-Type: 1-Wire
- Operating-Supply-Current: 1.5 mA

4.9 GPS Module

GPS modules contain tiny processors and antennas that directly receive data sent by satellites through dedicated RF frequencies. From there, it'll receive timestamp from each visible satellite, along with other pieces of data**.**

One of the global positioning system (GPS) devices utilizes data from satellites to locate a specific point on the Earth in a process named trilateration. Meanwhile, a GPS receiver measures the distances to satellites using radio signals for trilateration. And trilateration is similar to triangulation, which measures angles, depicted in this illustration (Tim Gunther, 2020). GPS modules contain tiny processors and antennas that directly receive data sent by satellites through dedicated RF frequencies. From there, it’ll receive timestamps from each visible satellite, along with other pieces of data. If the module’s antenna can spot 4 or more satellites, it’s able to accurately calculate its position and time.

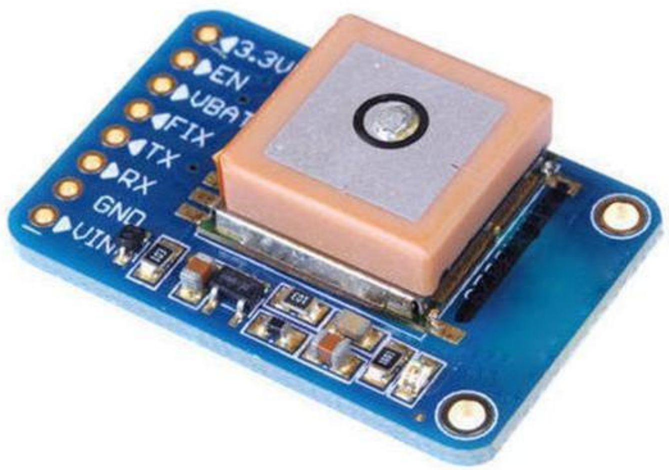


Fig 4.11 GPS Module

Specification:

- Standalone GPS receiver
- Anti-jamming technology
- UART Interface at the output pins (Can use SPI, I2C, and USB by soldering pins to the chip core)
- Under 1-second time-to-first-fix for hot and aided starts
- Receiver type: 50 Channels - GPS L1 frequency - SBAS (WAAS, EGNOS, MSAS, GAGAN)
- Time-To-First-fix: For Cold Start 32s, For Warm Start 23s, For Hot Start <1s
- Maximum navigation update rate: 5Hz
- Default baud rate: 9600bps
- EEPROM with battery backup
- Sensitivity: -160dBm
- Supply voltage: 3.6V
- Maximum DC at any output: 10mA
- Operation limits: Gravity-4g, Altitude-50000m, Velocity-500m/s
- Operating temperature range: -40ºC TO 85°C

4.10 Relay Module

A power relay module is an electrical switch that is operated by an electromagnet. The electromagnet is activated by a separate low-power signal from a microcontroller. When activated, the electromagnet pulls either open or closed an electrical circuit.

A simple relay consists of a wire coil wrapped around a soft iron core, or solenoid, an iron yoke that delivers a low reluctance path for magnetic flux, a movable iron armature, and one or more sets of contacts. The movable armature is hinged to the yoke and linked to one or more sets of the moving contacts. Held in place by a spring, the armature leaves a gap in the magnetic circuit when the relay is de-energized. While in this position, one of the two sets of contacts is closed while the other set remains open.

When an electrical current is passed through a coil, it generates a magnetic field that, in turn, activates the armature. This movement of the movable contacts makes or breaks a connection with the fixed contact. When the relay is de-energized, the sets of contacts that were closed open and break the connection, and vice versa if the contacts were open. When switching off the current to the coil, the armature is returned, by force, to its relaxed position. This force is usually provided by a spring, but gravity can also be used in certain applications. Most power relays are manufactured to operate quickly.

For the distribution of power in high-current applications, GEP Power Products is the industry leader in high-power relay module design and manufacturing. Rated up to 70 amps, GEP's power relay modules are designed for seamless integration in high-power distribution applications. The convenient integral mounting brackets provide easy installation and accessibility. With endless options such as terminal position assurance

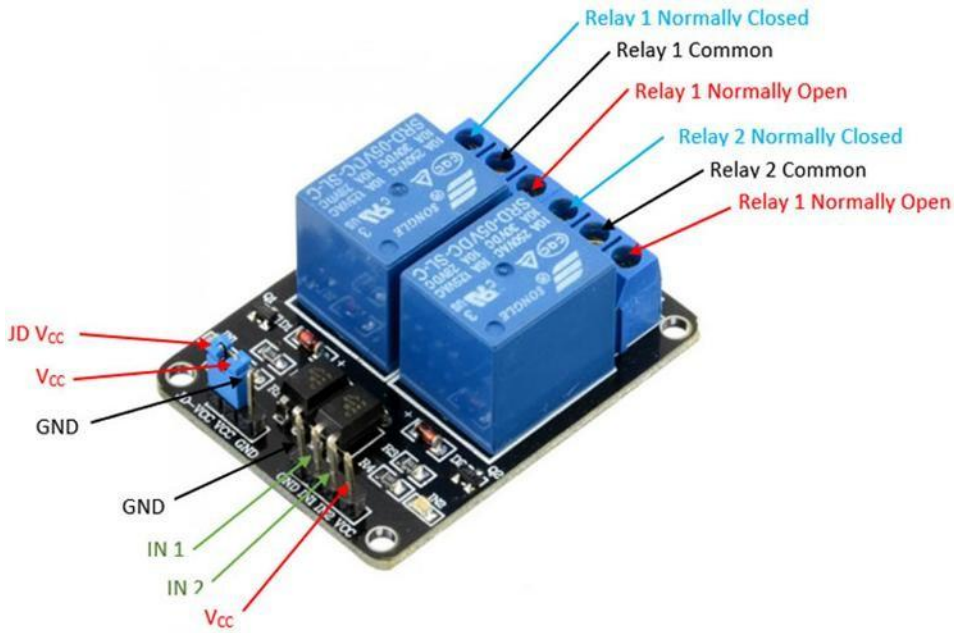


Fig 4.12 Relay 5v

available for wire retention, GEP Power Products' power distribution solutions, and off-road industry knowledge are second to none.

Relay Module Specifications:

- Supply voltage – 3.75V to 6V.
- Quiescent current: 2mA.
- Current when the relay is active: ~70mA.
- Relay maximum contact voltage – 250VAC or 30VDC.
- Relay maximum current – 10A.
- 5V 4-Channel Relay interface board
- Requires 15-20mA signal drive Current

- TTL logic compatible
- High-current AC250V/10A, DC30V/10A relay
- Status LED
- Equipped with 3.1mm screw holes for easy installation
- 61g
- 75 x 55 x 19.3mm (2.95 x 2.16 x 0.76″)

4.11 Software Development

Another crucial subject for this endeavor is software development. For this automated elevator monitoring system to work, the microcontroller has to be coded or programmed. The ESP-8266 is the microcontroller used in this project. The ESP-8266 controller was then controlled using the Arduino IDE. Using the App Controlled Lift mechanism, this step was flawlessly finished.

4.12 Software Arduino IDE

Let's take a closer look at the Arduino IDE's components after learning about them. You may do a number of actions, like creating and saving new files, by selecting the choices located near the top of the program screen. These button icons allow you to easily access some of the most frequently used functions. To check whether your code is error-free, click the verify button.

Fig 4.1 Arduino IDE [19]

You may transfer your code from your laptop to your Arduino and have it execute on your board by selecting Upload. You may input your program through the window, and you can read more about it in the message sections. As we work with the IDE, we'll go into more depth regarding messages, but for now, just know that they alert you to errors in your code and offer details like the RAM usage of the Arduino. Examine the buttons in the code editor's top row in more detail. You may quickly access the activities in the code window using these buttons. These actions involve making a new file, opening an existing file, and saving it. They also include uploading and testing your code on the Arduino board. Let's first discuss what a sketch is, and then we'll use each of these buttons. were kept without a file extension and produced using a text editor. The editor allows you to search for and replace text as well as cut and paste. When exporting and saving, the message section highlights errors and offers feedback. The Arduino Software (IDE) sends information to the terminal, including detailed error warnings.

4.13 Programming Interface

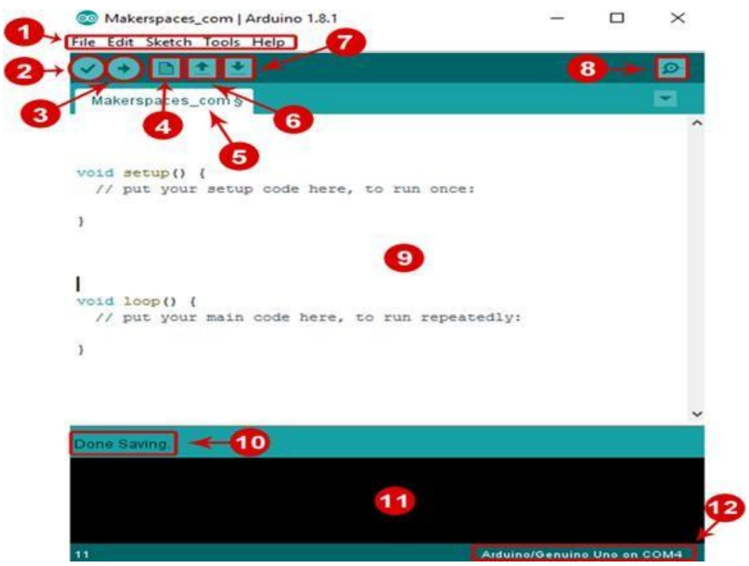


Fig 4.2 Sketch of Arduino IDE

Examine your surroundings and spend some time getting settled.

- Manu Bar: Provides you with the means to create and store Arduino sketches.
- Upload Button: Delivers the code to the attached board, in this example, the Arduino Uno. As the board uploads, the lights will flicker quickly.
- New Sketch: Brings up a new window with an empty drawing in it.
- Sketch Name: This is where the sketch's name appears once it has been saved.
- Open Existing Sketch: This lets you open a drawing that you've saved or one of the samples that are stored.
- Save Sketch: This preserves the drawing that is open at the moment.
- Serial Monitor: This will show the serial number of your Arduino when the board is attached.

- Coding Area: The sketch's code, which instructs the board on what to do, is composed in this section.
- Board and Serial Port: This indicates the board being utilized and the serial port to which it is attached.

You are now prepared to link your Arduino to your PC. Connect the USB cable's other end to the USB port on your computer and insert the first end into the ESP 8266. You must pick ESP 8266 under Tools, Board, then lastly once the board is linked.

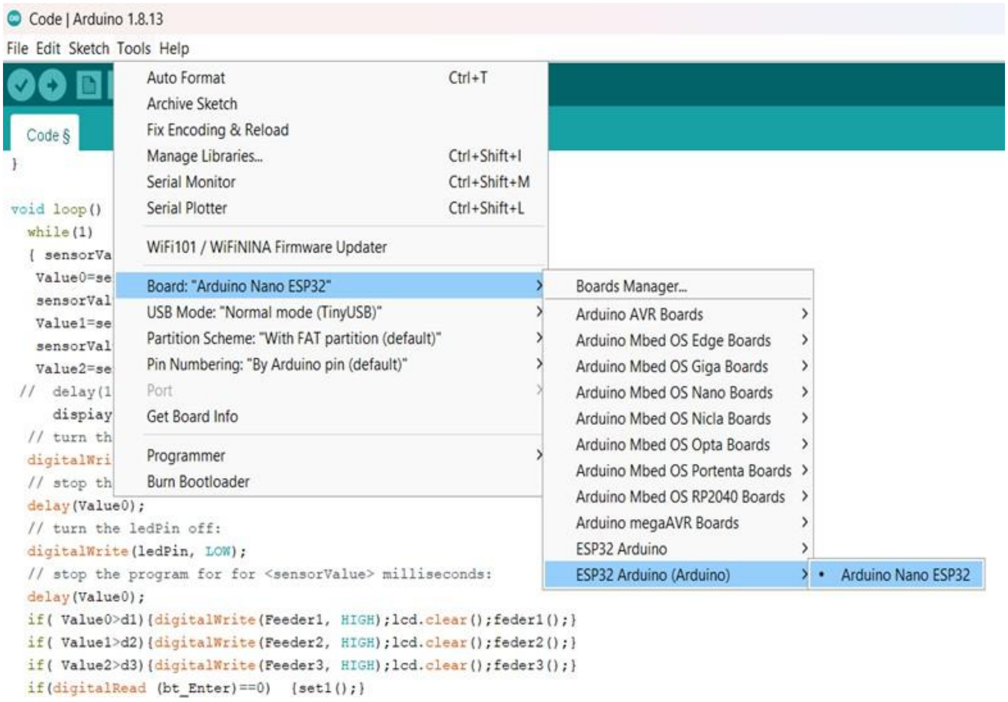


Fig 4.3 Arduino Board Selection

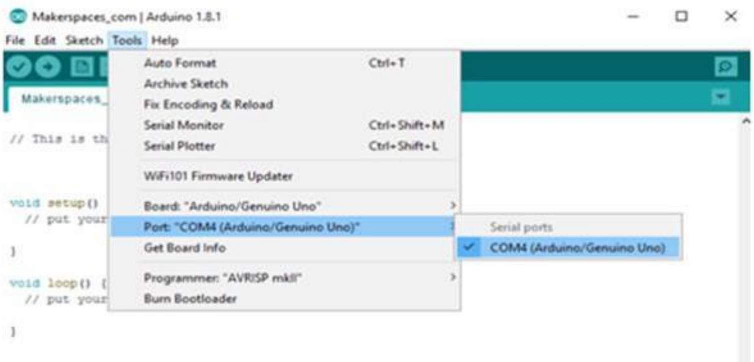
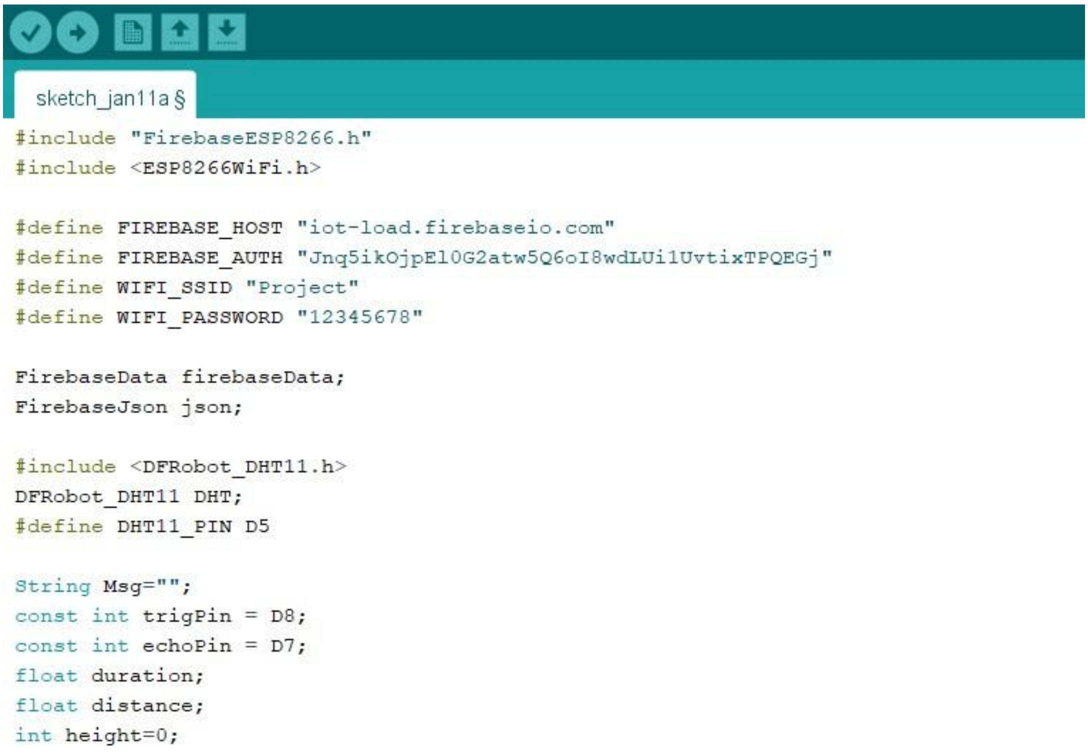


Fig 4.4 Arduino Port Selection

The Arduino needs to know which port on your PC you are currently utilizing. Go to Tools, then Port, and choose the port labeled Arduino to choose the port.

4.14 Arduino Variable Declaration and Library Include




```
#include "FirebaseESP8266.h"
#include <ESP8266WiFi.h>

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

FirebaseData firebaseData;
FirebaseJson json;

#include <DFRobot_DHT11.h>
DFRobot_DHT11 DHT;
#define DHT11_PIN D5

String Msg="";
const int trigPin = D8;
const int echoPin = D7;
float duration;
float distance;
int height=0;
```

Fig 4.5 Arduino IDE screen Library Include and Void Setup

When coding, all of the libraries are first included in the Arduino IDE mane area. Different libraries are needed for different components since different components are used in different projects. We may choose any variable we wish to define as a variable after installing the required library. Here, we have added variables of the character, int, and string types.

4.15 Void Setup Function

File Edit Sketch Tools Help

sketch_jan11a §

```
void setup() {
  pinMode(D4,OUTPUT);
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin, INPUT); // Sets the echoPin as an Input



  Serial.begin(9600);

  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting to Wi-Fi");

  while (WiFi.status() != WL_CONNECTED)
  {
    digitalWrite(D4,1);
    Serial.print(".");
    delay(200);
    digitalWrite(D4,0);
    Serial.print(".");
    delay(200);

  }

  Serial.println();
  Serial.print("Connected with IP: ");
  Serial.println(WiFi.localIP());
  Serial.println();

  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
  Firebase.reconnectWiFi(true);

}
```

Fig 4.6 Void Setup Function

4.16 Void Loop Function

File Edit Sketch Tools Help

sketch_jan11a §

```
void loop() {
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    duration = pulseIn(echoPin, HIGH);
    if(duration>0) distance= duration*0.034/2;

    DHT.read(DHT11_PIN);

    Msg +=distance*100;
    Msg +="-";
    Msg +=analogRead(A0);
    Msg +="-";
    Msg +=DHT.temperature;
    Msg +="-";
    Msg +=DHT.humidity;
    Serial.println(Msg);
    Firebase.setString(firebaseData, "ProjectDevelopment/RainAnalyzer/Data",Msg);
    Msg="";
}
```

Fig 4.7 Void Setup Function

CHAPTER 5
APPS DEVELOPMENT

5.1 Internet of Things (IOT)

The new paradigm known as the Internet of Things (IoT) has made living more technologically advanced than it was in the past. Smart homes, smart transportation, smart industries, smart libraries, smart pollution control, and strength saving are just a few of the advancements made possible by the Internet of Things. The use of IoT to decorate the time period has been the subject of several significant studies, research initiatives, and investigations. There are still many difficult situations and problems that need to be fixed before IoT may be used to its full potential. It is important to consider these difficult situations and problems from a range of IoT-related perspectives, including applications, difficult situations, enabling technology, social and environmental effects, etc. The item also highlights recent research and demonstrates how it adds to IoT-exclusive features. Additionally, the significance of vast amounts of data and how to assess them in the context of the Internet of Things have been discussed. This article may help readers and researchers better comprehend the Internet of Things and its potential applications in practical settings. IoT is gradually becoming an essential part of our lives that is evident everywhere. The Internet of Things (IoT) is a collective creation that combines large systems, smart frameworks, and intelligent gadgets and sensors. Additionally, it makes use of quantum and nanotechnology to reach previously unachievable levels of storage, sensing, and processing speed. With the growing involvement of IoT devices and the technology in daily ordinary lives, a remarkable transformation may be observed.Another essential IoT component is a Smart Health Sensing device (SHSS). The SHSS supports human fitness by means of gadgets and a small intelligent system.To measure and track certain fitness difficulties, health condition, the amount of energy spent in the fitness facility, etc., these gadgets may be utilized both indoors and outdoors.Additionally, it is becoming used to identifying urgent fitness circumstances in hospitals and trauma centers. Because of this, it has changed the condition of science in general by making it possible through complex gadgets and modern technology. Furthermore, scholars and IoT developers are always trying to make elderly and disabled people's current attire better. The majority of people are using such devices and systems because they are highly strong in terms of improvement fees and easily available within a daily charge budget.

5.2 Firebase

5.2.1 Using Firebase Functionalities in an Android App

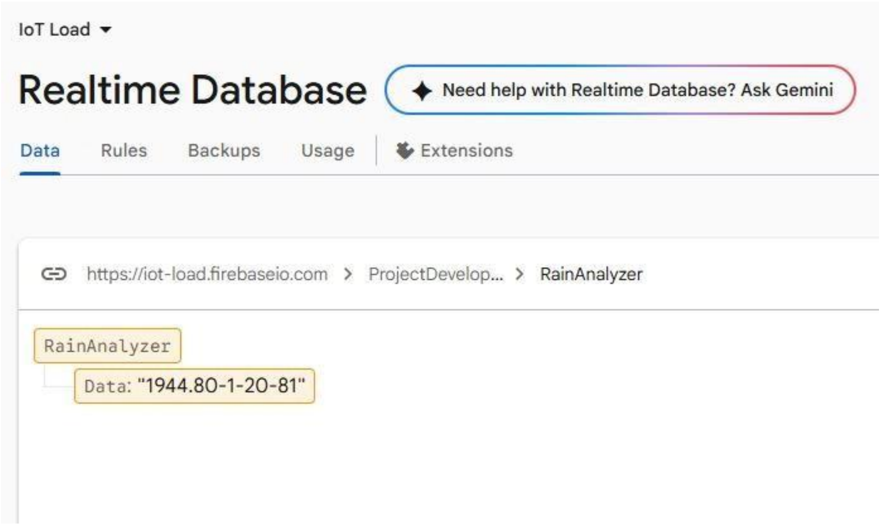


Fig 5.1 Real-Time Database

All of Firebase's features may be used by the Android application with just a few lines of code. Topics including databases, storage, and authentication have all been discussed in this section. To learn more about the capabilities, click the link to the Google Firebase handbook in the reference section. This is how to use a few of the features.

5.2.2 Verification

The user can generate a login ID by using the following code once Firebase and authentication dependencies have been added to the Android application.

```
FirebaseAuthauth=FirebaseAuth.getInstance();

Auth.signInWithEmailAndPassword(email,password).addOnCompleteListener(new
OnCompleteListener())

{

@override

Public Void Oncomplete (task Task)

{ if(task.isSuccessful()) {Firebaseuser User=task.getResult().getResult().getUser();

String email=User.getemail();

//…
```

```
}}
}
```

5.3 Developing Apps

We developed an application to monitor this project. It is an Android application. Using this application is simple. This app was developed using the MIT App Inventor framework. Anyone can develop completely working programs for Android and iOS tablets, as well as iPhones and Android phones, using its intuitive graphical programming environment. Block-based coding programs foster people's intelligence and inventiveness. The four parking spots shown in this app are as follows. It is now in final beta testing as of July 8, 2019. It enables non-programmers to develop apps for Android and iOS, two operating systems (OS). It is free and open-source software, with the source code licensed under a Creative Commons Attribution ShareAlike 3.0 Unported license and the rest of the application under an Apache License 2.0. Users can now create Android apps by dragging and dropping visual objects into a graphical user interface (GUI) that resembles the programming languages Scratch and StarLogo. Work is ongoing on an App-Inventor Companion, which would enable the app to run and debug on iOS devices.

Many different programming languages are used by App Inventor. A new programming language called "App Inventor Language" was developed by MIT App Inventor 2 blocks code. It has not yet been given a name. We're working on a textual version of this language called TAIL, or Textual App Inventor Language.

Many users will find this difficult to comprehend, yet the block code you see on your screen *is* your language's source code. When you save an. aia file (which is just a.zip file), the blocks are saved in an XML textual format that represents the same nested tree structure as the blocks on your screen. So, if you like to think of text as source code, this XML format is the App Inventor program's source code.

5.3.1 MIT App Inventor

A drag-and-drop interactive programming tool called MIT App Inventor may be used to create fully working Android mobile applications. App Inventor promotes a new type of private cell computing that allows people to design, create, and use—in my view—significant cell generation solutions for their daily lives in many circumstances. App Inventor promotes universal virtual literacy by allowing developers to concentrate on making informed programming decisions rather than coding language syntax thanks to its simple programming paradigm and progressive development features. Since it was moved from Google to MIT, certain improvements have been made, and research projects have started.

5.3.2 Drag and Drop Setup of the App

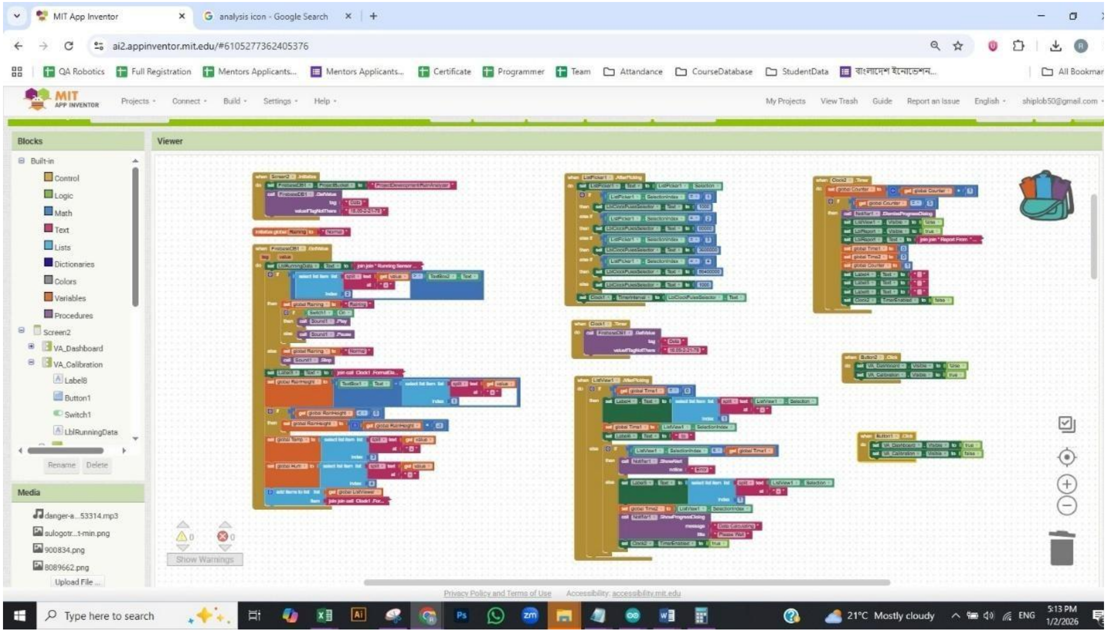


Fig 5.2 Drag and Drop code window

Before starting our major project, we created a few simple apps in App Inventor. The code and output part for the compass are seen above. The purpose of this block program is to inspect or create user-facing elevator buttons. This is linked by a microcontroller to the database.

CHAPTER 6

RESULT & DISCUSSION

The project on IoT-based vehicle health monitoring safety techniques concepts was completed successfully. The monitoring of vehicle data has been developed successfully by an Android application. Here, the load sensor can sense heavy load in the vehicle and send data to mobile apps, the fuel level indicator is measured by a variable resistor of the level of fuel, and it also sends its data. In this project, the fuel tank is 155 ml, and in that case, according to the variable resistance, the fuel is shown in the chart are given below.

GPS indicates the current location of the vehicle. The buzzer and display help driver to recognize the current stage of the vehicle. Finally, all data is stored on an android mobile, which is operated by the owner.

In the Android app, there are more than 2 slots to add more vehicles. That's why every individual owner can monitor the current data of vehicles with a single app.

How We Collect & Measure Data

Data measurement procedures for all projects were carried out under controlled and calibrated conditions to ensure accurate and reliable results. Load and vehicle status data were collected by incrementally applying known loads and observing vehicle performance, recording each condition as working or not working. Radiator temperature testing involved adding measured quantities of hot water step by step and recording stabilized temperature readings using a calibrated sensor. The ultrasonic fuel level indicator used water as a safe substitute for fuel, with level measurements taken incrementally and converted into percentage values. The real-time brake pad status indicator simulated wear by sequentially removing brake pad parts and converting detected conditions into percentage status values. Digital humidity and temperature readings were obtained using a DHT sensor exposed to controlled heat and vapor sources at varying distances. In all cases, measurements were recorded in data tables, verified through repeated trials, and confirmed to be 100% functional and providing accurate data.

6.1 Data Tables

Load & Status

Load No	Implemented Load on Vehicle (Gram)	Status
Load 1	5	**Working**
Load 2	5	**Working**
Load 3	4	**Working**
Load 4	5	**Working**
Load 5	5	**Working**
Load 6	10	**Working**
Load 7	20	**Working**
Load 8	30	**Working**
Load 9	50	**Not Working**
Load 10	-20 (Unload)	**Working**

Table (6.1.1): Data Table For Load & Status of Vehicle.

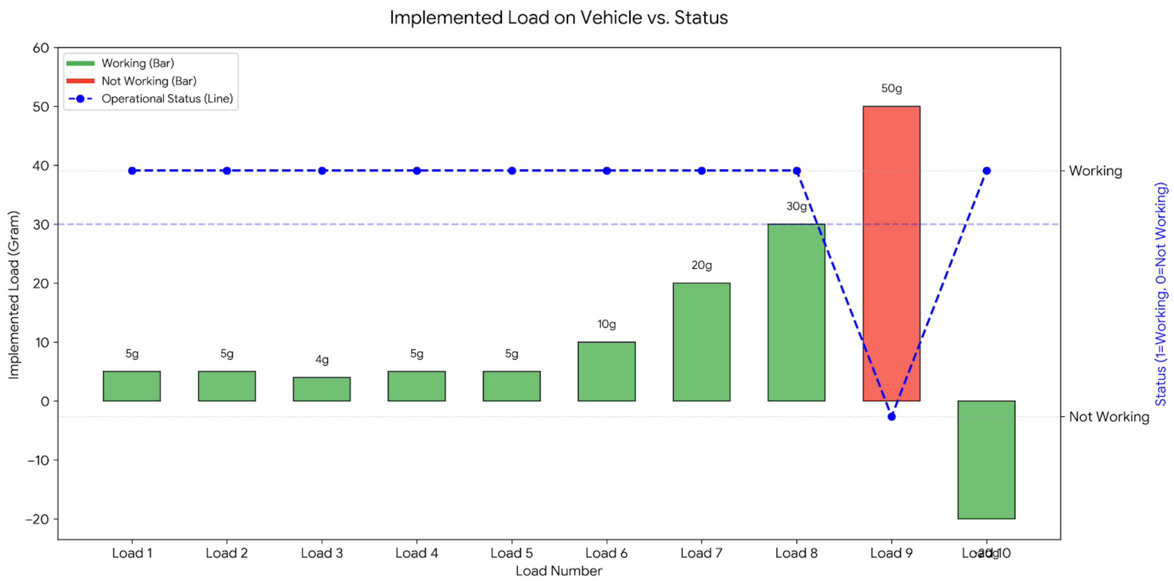


Fig (6.1.2) : Load Vs Status Of Vehicle

Radiator Temperature Testing

Water Quantity After Hot Water Addition (mL)	Displayed Temperature (Degree Celsius)
300	23
400	32
500	38
600	47
700	56
800	62
900	71

Table (6.1.1): Data Table For Radiator Temperature Testing

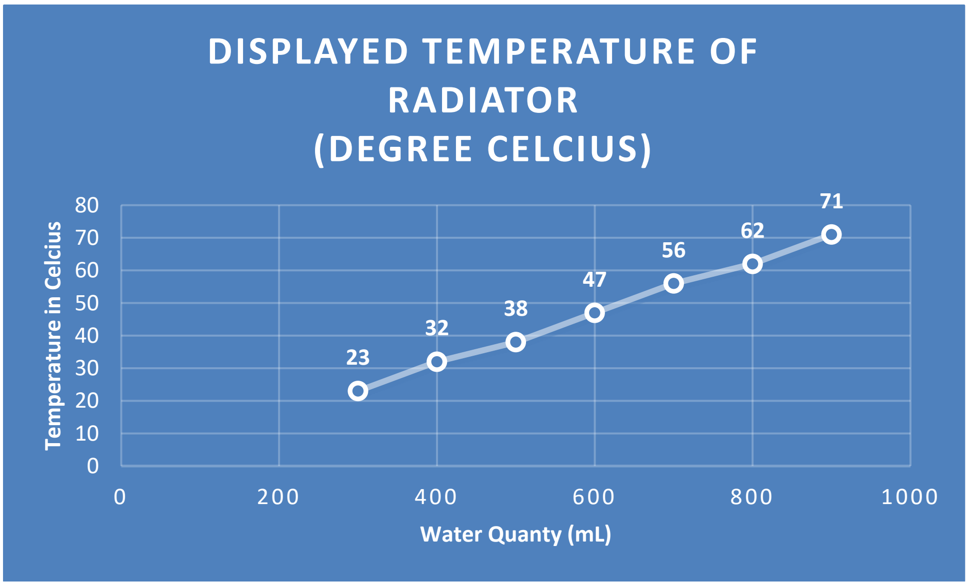


Fig (6.1.2) : Radiator Temperature Testing Graph

Ultrasonic Fuel Level Indicator

Water Quantity (mL)	Percentage (%)
0	0%
500	13%
1000	48%
1500	70%
2000	96%
2200	100%

Table (6.1.1): Data Table Actual Quantity & Displayed Level.

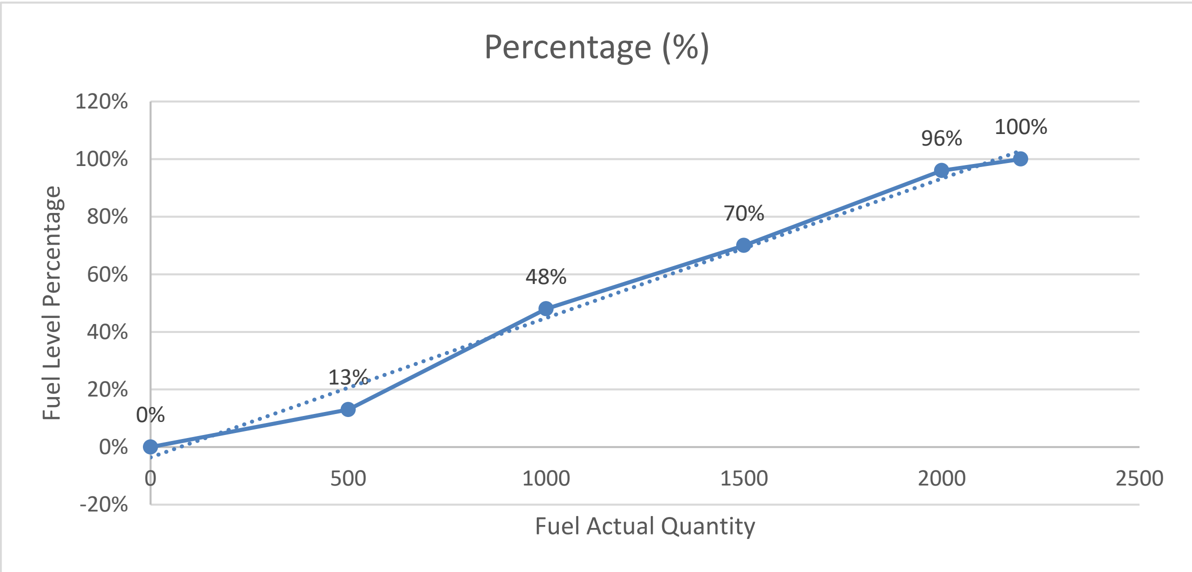


Fig (6.1.2): Actual Fuel Quantity vs Displayed Percentage

Real Time Brake Pad Status Indicator

Attached Part of Brake Pad (Pcs)	Displayed Percentage (%)
4	100%
3	75%
2	50%
1	25%

Table 6.1: Brake Pad Worn Out Table

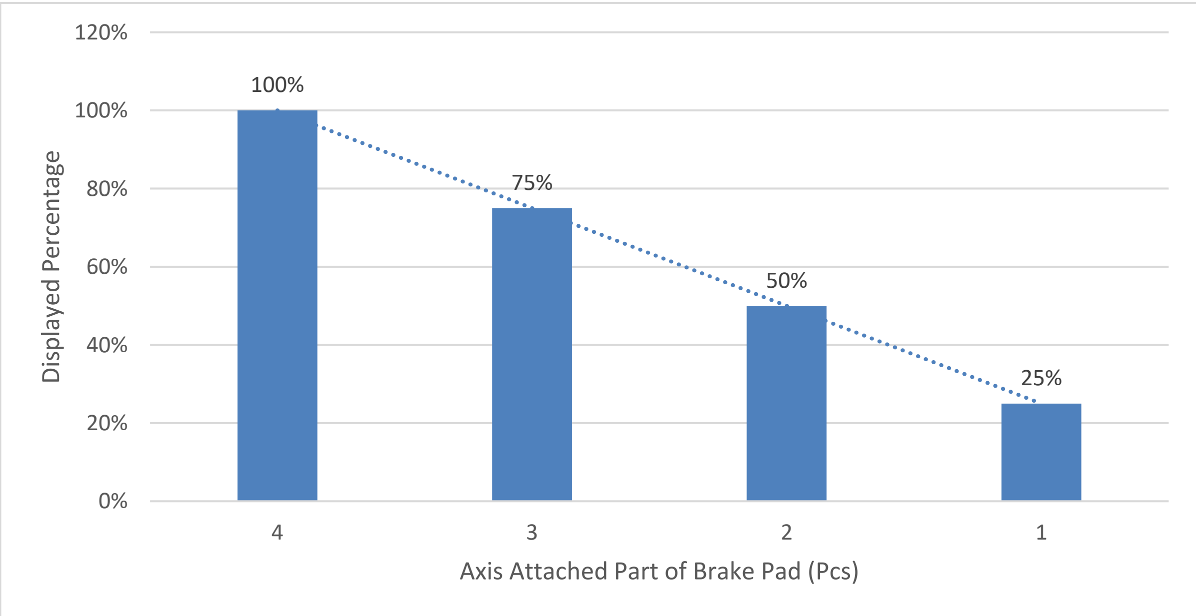


Fig 6.1: Brake Pad Worn Out Graph

Digital Humidity & Temperature (DHT) Sensor

Hot water Vapor Flow from The Distance (Inch)	Displayed Temperature (Degree Celsius)	Humidity (%)
5	24	37
4	29	46
3	37	62
2	44	78
1	49	87

Table 6.1: Digital Humidity Temperature Data Reading Table

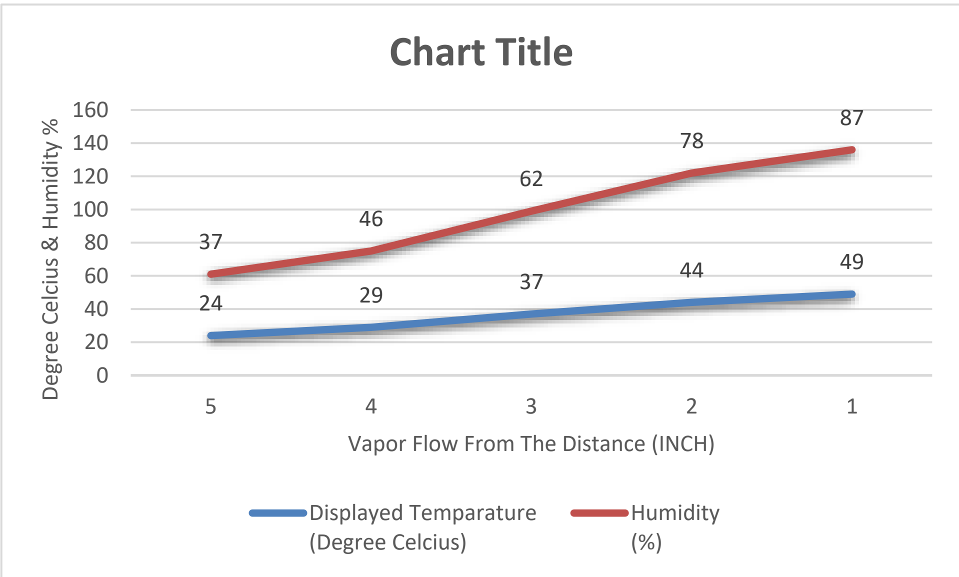


Fig 6.1: Digital Humidity Temperature Data Reading Graph

Data Measurement Procedures:

Load & Status: The data were collected by first calibrating the weighing scale or load measurement device to ensure accurate readings in grams and preparing the vehicle under controlled test conditions. Known loads were then applied incrementally to the vehicle, starting from a low or unloaded condition and increasing step by step, with each load assigned a sequential label. After applying each load, the vehicle was operated under normal conditions and its performance was observed in terms of movement, stability, and response. Based on predefined performance criteria, the operational status for each load was recorded as either "Working" or "Not Working," and any abnormal behavior at higher loads was noted. Finally, all load values and corresponding statuses were entered into a data table, reviewed for consistency, and verified through repeated measurements where necessary.
Status & Condition: 100% in working condition and giving data properly

Radiator Temperature Testing: For the prototype radiator temperature testing project, a temperature sensor was installed on the radiator and the system was powered on and calibrated to ensure accurate readings. Hot water was prepared and added to the radiator in measured quantities using a graduated container, starting from 300 mL and increasing step by step. After each addition, sufficient time was allowed for the heat to distribute and stabilize within the radiator. The temperature corresponding to each water quantity was then measured by the sensor, displayed by the system, and recorded in a data table. Repeated measurements were taken where necessary to verify the accuracy and consistency of the results.
Status & Condition: 100% in working condition and giving data properly

Ultrasonic Fuel Level Indicator: For the ultrasonic fuel level indicator project, the data were collected by first mounting and calibrating the ultrasonic sensor at the top of the tank to establish the empty condition as the reference level. Water was used as a safe substitute for fuel and added to the tank in measured quantities using a graduated container, starting from 0 mL and increasing step by step up to the maximum tank capacity. After each addition, the ultrasonic sensor measured the distance to the water surface, and this value was processed by the microcontroller to determine the corresponding fuel level. The measured level was then converted into a percentage of the total tank capacity. Finally, the water quantity and the corresponding percentage readings were recorded in a data table and verified through repeated measurements to ensure accuracy and consistency.
Status & Condition: 100% in working condition and giving data properly

Real Time Brake Pad Status Indicator : For the real-time brake pad status indicator project, sensors or conductive indicators were installed at predefined wear levels on the brake pad assembly and the system was initialized to detect the presence of each brake pad part. The fully intact brake pad condition with all four parts attached was set as the reference state and calibrated to represent 100% brake pad status. Brake pad wear was then simulated by sequentially removing or disconnecting one part at a time to represent different stages of wear. At each stage, the system detected the number of attached brake pad parts and processed this information using the controller. The detected condition was converted into a percentage value, displayed in real time, and recorded in a data table to verify accuracy and consistency.
Status & Condition: 100% in working condition and giving data properly

Digital Humidity Temperature Data Reading: For the DHT sensor data reading project, the DHT temperature and humidity sensor was connected to a microcontroller, powered on, and allowed to stabilize under normal ambient conditions. Hot water was used as a controlled source of heat and water vapor, and the sensor was positioned at fixed distances from the vapor source ranging from 5 inches to 1 inch. At each distance, the system was allowed to reach stable conditions before the sensor measured the surrounding temperature and relative humidity. The measured values were displayed and recorded in a data table, and repeated observations were taken where necessary to ensure accuracy and consistency of the results.
Status & Condition: 100% in working condition and giving data properly

After analyzing this project, we found different types of results.

- **Signal System:** We have designed a fully automatic vehicle monitoring system. From input devices (Sensors), we get real-time monitoring data, and its send our mobile apps through micro-controller (ESP-8266).
- **Accident Monitoring:** We have identified when accidents occurred and detected the accident spot with the help of the GPS Module.
- **Performance:** We knew that the microcontroller took some time to process the signal. Here, all this data is done through the internet. So, processing time depended on the Internet speed. When the Wi-Fi module crossed the High internet speed zone or connected with high bandwidth, it took 3sec to 5 seconds. Otherwise, it took 5sec to 12sec. Sometimes the controller disconnects from our database due to a poor connection.
- **Cost:** We estimate the project cost carefully. To complete this project, we spent around 14,200 taka. But when this project is applied in the practical field, the cost will vary because of using a more precise sensor.

6.2 Discussion

We have developed vehicle fuel consumption system, brake shoe monitor at a time engine temperature, over loading and starter authentication. This project can be developed with help of our supervisor. He guides us proper way. Withed the co-operation of microelectronics technology, we can apply electronics and programmable related devices within an automobile several systems where it could increase the efficiency, security and longevity of an automobile as well as providing digitalized service for consumers. Doing this project, we can gain electronics knowledge and work hardware software combination. We learned programing C and Arduino programing. Within this project we have tried to show the actual calculation of fuel (in litter) and indication of fuel theft, excess load detection and engine temperature. Each output data driver can see onto the dashboard even owner can know all of it through mobile application wherever owner is staying. We optimize; the mobile application and the systems would be more beneficial for owners. As commercially we can get scopes for attaching or installing our project, it must secure the vehicle, reduce maintenance cost and look after them from wherever they want. Each system of this project is customer demands from the field study we have abled to understand & our objective has fulfilled So, we have been inspired to develop this project.

CHAPTER 7
CONCLUSIONS

Our proposed IoT based design for a real-time fuel consumption system, brake shoe monitoring and engine internal temperature measurement at a time overload checking. Our primary goal was to extend a solution that would allow authority to monitor any number of vehicles in operation, from a central location using an IoT based infrastructure. Monitoring the performance of a brake system makes it possible to conduct real-time assessments of its various parameters of brake shoe. The various measurable parameters relevant to brake performance have been discussed and a fundamental subset of them was then selected for real-time monitoring and reporting. Through our research, we have been able to show fuel consumed, fuel removed, fuel theft and brake shoe monitor, internal temperature, overload monitor and other related parameters. The benefits of our project can be extended to similar other industries where fuel consumption and brake shoe monitoring are of critical concern. Through use of our IoT based monitoring system, we have demonstrated that it is possible to have a detailed operational view of vehicles and engines requiring fuel to operate, monitoring brake shoe and maintaining internal temperature across IoT based systems using cheap yet reliable capacitive sensors.

- **Future Work:** We can be more accurate and valuable by adding this type of future work.

  ➢ We can attach camera inside the vehicle. By doing image processing, we can calculate total passengers. We can easily find autistic people and also identify any person who carries his/her extra bag during journey time.

  ➢ GPRS will also be implemented 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 of this system, we can monitor our vehicles in rural areas.

  ➢ We also monitor Irish camera in front of driver. In this way, we can see our driver's attention. It also provides vehicle safety to assess the driver's alertness and warn the driver if needed and eventually apply the brakes.

  ➢ In future, we will also implement speed meter. When any vehicle drives roughly or passes at high speed, authorities will know this information. It also decreases accidents and gives us safe journeys.

REFERENCE

1. A Proposed System for Automatic Vehicle Monitoring and Accident Detection in Bangladesh. Sanjana Srabanti, Md. Asaduzzaman, Muhammad Kivran Bin Mokter, Tasmiah Tamzid Anannya. IEEE, International Conference-2018.

2. A novel in-vehicle real-time brake-monitoring system. D E Fumi and I A Sultan. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering 2009 223: 793

3. Hillol Kargupta, Ruchita Bhargava, Kua lin, Michel Powers, Patrick Blair, Samuel Bushra, James Dull"VEDAS: A mobile and distributed data stream mining system for real-time vehicle monitoring." Proceedings of the 2004 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics, 2004.

4. Hillol Kargupta, Vasundhara Puttagunta, Martin Klein, Kakali Sarkar"On-board vehicle data stream monitoring using minefleet and fast resource constrained monitoring of correlation matrices." New Generation Computing 25.1 (2006): 5- 32.

5. Last, Mark. "Vehicle failure prediction using warranty and telematics data." Learn. Vol. 29. No. 3. 2011.

6. Kargupta, Hillol. "Onboard vehicle data mining, social networking, advertisement." U.S. Patent No. 8,478,514. 2 Jul. 2013.

7. Stewart,Gilbert W. "Matrix perturbation theory." (1990).

8. Vehicle Health Monitoring System M.Jyothi kiran, S.Ravi teja.

9. Iot Based Vehicle Health Monitoring System Yogesh Kr. Mallick, Anamika Talukdar, Debleena Bhattacharjee, Sandipan Roy

10. Eriksson, Jakob, et al. *"The Pothole Patrol: Using a Mobile Sensor Network for Road Surface Monitoring."* Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services (MobiSys), ACM, 2008.

11. Chen, Feng, et al. *"Vehicle Health Monitoring Using On-Board Diagnostics and Internet of Things."* IEEE Internet of Things Journal, Vol. 4, No. 3, 2017.

12. Wu, Bin, et al. *"Design of an Intelligent Vehicle Accident Detection System Using GPS and GSM."* International Journal of Engineering Research & Technology (IJERT), Vol. 3, No. 2, 2014.

13. Al-Karaki, Jamal N., and Ahmed E. Kamal. *"Routing Techniques in Wireless Sensor Networks: A Survey."* IEEE Wireless Communications, Vol. 11, No. 6, 2004.

14. Gubbi, Jayavardhana, et al. *"Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions."* Future Generation Computer Systems, Vol. 29, No. 7, 2013.

15. Rajasekaran, S., et al. *"CAN Bus Data Analysis for Vehicle Health Monitoring System."* International Conference on Advanced Computing and Communication Systems (ICACCS), IEEE, 2016.

16. Zhang, Y., and V. V. Phoha. *"Vehicle Telematics Data Mining for Intelligent Transportation Systems."* IEEE Transactions on Intelligent Transportation Systems, Vol. 12, No. 4, 2011.

17. Saini, P., and A. K. Sharma. *"IoT Based Smart Vehicle Accident Detection and Notification System."* International Journal of Computer Applications, Vol. 160, No. 8, 2017.

18. Lee, Eun-Kyu, et al. *"Vehicular Cloud Networking: Architecture and Design Principles."* IEEE Communications Magazine, Vol. 52, No. 2, 2014.

APPENDIX

```
#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;
FirebaseJson json;



void setup() {
  pinMode(A0,INPUT);
  pinMode(D2,INPUT_PULLUP);
  pinMode(D5,INPUT_PULLUP);
  pinMode(D6,INPUT_PULLUP);
  pinMode(D7,INPUT_PULLUP);
  pinMode(D8,INPUT_PULLUP);
  Serial.begin(9600);
  pinMode(D4,OUTPUT);
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting to Wi-Fi");
```

```
  while (WiFi.status() != WL_CONNECTED)
  {
    digitalWrite(D4,1);
    Serial.print(".");
    delay(200);
    digitalWrite(D4,0);
    Serial.print(".");
    delay(200);

  }

  Serial.println();
  Serial.print("Connected with IP: ");
  Serial.println(WiFi.localIP());
  Serial.println();

  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
  Firebase.reconnectWiFi(true);

}

void loop() {
  if(WiFi.status() != WL_CONNECTED)
  {
```

```
    while (WiFi.status() != WL_CONNECTED)
     {
       digitalWrite(D4,1);
       Serial.print(".");
       delay(200);
       digitalWrite(D4,0);
       Serial.print(".");
       delay(200);
   }
  }

  Firebase.setInt(firebaseData,
"/ProjectDevelopment/PassengerMonitor/A0",analogRead(A0));
  Firebase.setInt(firebaseData,
"/ProjectDevelopment/PassengerMonitor/D5",digitalRead(D5));
  Firebase.setInt(firebaseData,
"/ProjectDevelopment/PassengerMonitor/D6",digitalRead(D6));
  Firebase.setInt(firebaseData,
"/ProjectDevelopment/PassengerMonitor/D7",digitalRead(D7));
  Firebase.setInt(firebaseData,
"/ProjectDevelopment/PassengerMonitor/D8",digitalRead(D8));

  if(digitalRead(D2)==0)
    Firebase.setInt(firebaseData, "/ProjectDevelopment/PassengerMonitor/D2",0);
  delay(200);

}
```