Design and Development of IoT Based Comprehensive System for Emergency Assistance

By

Sahriar Habib
16121066
Zawata Afnan
16121060
Sakib Anam Chowdhury
19121157
Sarah Altaf Chowdhury
19121156

A thesis submitted to the Department of Electrical and Electronic Engineering in partial fulfillment of the requirements for the degree of
B.Sc. in Electrical & Electronic Engineering

Department of Electrical and Electronic Engineering
BRAC University
April 2020

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. I/We have acknowledged all main sources of help.

Student's Full Name & Signature:

Sahriar Habib 16121066	**Zawata Afnan** 16121060
Sakib Anam Chowdhury 19121157	**Sarah Altaf Chowdhury** 19121156

Approval

The thesis/project titled "Design and Development of IoT Based Comprehensive System for Emergency Assistance" submitted by

1. Sahriar Habib (16121066)
2. Zawata Afnan (16121060)
3. Sakib Anam Chowdhury (19121157)
4. Sarah Altaf Chowdhury (19121156)

of Spring, 2020 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Electrical & Electronic Engineering on 27th June 2020.

Examining Committee:

Supervisor:
(Member)

____

Abu S.M. Mohsin, Ph.D.
Assistant Professor
Department of Electrical & Electronics Engineering
Brac University

Program Coordinator:
(Member)

____

Abu S.M. Mohsin, Ph.D.
Assistant Professor
Department of Electrical & Electronics Engineering
Brac University

Departmental Head:
(Chair)

____

Shahidul Islam Khan, Ph.D.
Professor & Chairperson
Department of Electrical &Electronics Engineering
Brac University

Abstract/ Executive Summary

For any society to make significant unabridged progress, it is imperative to utilize technical skills or otherwise, and ensure personal safety before anything else. This includes reducing the time of response to an accident to the minimum, among other things. The development of the Internet of Things (IoT) Based Emergency Response system is an attempt towards minimizing the response time of essential emergency services such as ambulance services provided by hospitals and fire stations and police services. By approaching this issue, we are also hoping to reduce rates of casualty and injuries occurring to people involved in household fire and road accidents in the longer run. The system is fully automated to be able to send emergency notifications to nearby hospital, police station or fire station and an emergency contact in case of an accident. This is done via a combination of a device that has to be installed in the vehicle or/and in the home of the consumer and a remote server system that responds to the device accordingly. The motive is to provide the live location of the accident and measure the severity of the damage caused so that the appropriate degree of help can be provided as fast and efficiently as possible using routes with the lowest traffic congestion. We believe this system will reduce the response time of service providers after accidents and, in the process, bring down casualty rates in a society.

Keywords: IoT; Accident Detection; IoT Fire Alarm; Emergency response to an accident; IoT System.

Acknowledgement

We are grateful to the Almighty Allah for his blessings and mercy that helped us to meet the challenges to complete the project.

There are some people without whose contributions it would not have been possible to reach the climax. Many thanks to our adviser, Dr Abu S.M. Mohsin, who read our numerous revisions and helped make some sense of the confusion.

Also, thanks to all faculty and staff members of EEE department who offered guidance and support. And at long last, much appreciated to our guardians and various companions who endured this long process with us, continuously delivering support and love.

Funding Acknowledgement

We would like to thank "The Information and Communication Technology Division (ICT-Division), Ministry of Posts, Telecommunications and Information Technology, People's Republic of Bangladesh" for awarding us the "ICT Special fund", providing financial support to complete building this system.

Table of Content

Declaration .... **ii**

Approval .... **iii**

Abstract/ Executive Summary .... **iv**

Acknowledgement .... **v**

Funding Acknowledgement .... **vi**

Table of Content .... **vii**

List of Tables .... **xi**

List of Figures .... **xii**

List of Acronyms .... **xiv**

Chapter 1 Introduction **1**

1.1 Motivation of the thesis 1

1.2 Introduction to IoT 2

1.3 IoT Architecture Layer 4

1.4 Literature review of IoT System and application 7

1.5 IoT Technologies 10

1.6 Network Topologies 17

1.7 Thesis Structure 20

Chapter 2 The IoT Based Home Device **22**

2.1 Literature Review 22

2.2 Home Device Design 25

2.2.1 Component Selection 25

2.2.2 Component Description 26

2.3 Hardware Connection 29

2.3.1 Connection Diagram 29

2.3.2 Schematic Diagram 30

2.3.3 PCB Layout 30

2.3.4 Implemented Picture 31

2.4 Working Mechanism 32

2.4.1 Data Sensing Mechanism 32

2.4.2 Arduino Code 33

2.4.3 Wi-Fi Connectivity Mechanism 34

2.4.4 NodeMCU Code 34

2.5 Result and Discussion 36

2.5.1 Sensor Data Receive 36

2.5.2 Sensor Data Characteristics 40

2.6 Cost analysis 44

Chapter 3 The IoT Based Vehicle Device 46

3.1 Literature Review 46

3.2 Vehicle Device Design 49

3.2.1 Component Selection 50

3.2.2 Component Description 50

3.3 Hardware Connection 54

3.3.1 Connection Diagram 54

3.3.2 Schematic Diagram 55

3.3.3 PCB Layout 56

3.3.4 Implemented Picture 57

3.4 Working Mechanism.. 58

3.4.1 Data Sensing Mechanism 58

3.4.2 GPRS Connection Mechanism 59

3.4.3 Arduino Code 60

3.5 Result and Discussion 64

3.5.1 Sensor Data Receive 64

3.5.2 Sensor Data Characteristics 67

3.6 Cost analysis.. 73

Chapter 4 Comprehensive System Design.. 74

4.1 Significance of the Comprehensive system 74

4.2 System design in stages.. 75

4.3 System Working Mechanism 77

4.3.1 Overall System Data flow 77

4.3.2 Server Data Receive 78

4.2.3 Data Routing and Storing 81

4.3.4 Algorithm and Notification 83

4.3.5 Algorithm Testing 86

4.3.6 User App Realtime Data Feed and Alert.. 90

4.3.7 Emergency Responder Notification System... 92

4.3.8 Emergency Notification simulation 93

4.3 System Security and Privacy 95

4.3.1 Security Literature Review: 95

4.3. Security and Privacy analysis of our system: 98

Chapter 5 Conclusion 99

5.1 System Limitations and Drawbacks 99

5.1.1 Limitations 99

5.1.2. Drawbacks 99

5.2 Future Work and Possibilities 101

5.3 Conclusion 102

Publications 103

References: 104

Appendix A. 107

List of Tables

Table 1: Test Data for Home Device 38

Table 2: Home Device Threshold value chart 44

Table 3: Vehicle Device Data Set for Stage one and two of Test case 65

Table 4: Vehicle Device Data set for Stage three and four of Test case 66

Table 5: Vehicle Device Data set for Stage five of Test case 67

Table 6: Vehicle Device Threshold value chart 72

Table 7: Threshold values for home and vehicle kit 83

Table 8: Detection levels in case of Fire 87

Table 9: Detection levels in case of Gas Leakage 87

Table 10: Detection levels in case of Fire due to Short-Circuit 87

Table 11: Detection Values in case of Fire in the Vehicle Engine 88

Table 12: Detection Values in case of Gas Leakage in the Vehicle 88

Table 13: Detection Values in case of Vehicle Collision 88

Table 14: System Response for Test Cases 89

List of Figures

Figure 1: Different Types of Network Topologies 18

Figure 2: Home Device Components 26

Figure 3: Home Device Connection Diagram 29

Figure 4: Home Device Schematic Diagram 30

Figure 5: Home device PCB Layout 30

Figure 6: Prototype of Home Device 31

Figure 7: LPG Data graph of home device 40

Figure 8: Smoke Data Graph of home device 41

Figure 9: Flame Sensor response of the Home Device 42

Figure 10: Temperature Data graph of Home Device 43

Figure 11: Vehicle Device Components 51

Figure 12: Vehicle Device Connection Block Diagram 54

Figure 13: Vehicle Device Schematic Diagram 55

Figure 14: PCB Layout of Vehicle Device 56

Figure 15: Prototype of Vehicle Device 57

Figure 16: Temperature Data Graph of Vehicle device 68

Figure 17: CNG Data Graph of Vehicle device 69

Figure 18: Smoke Data Graph of Vehicle device 70

Figure 19: Acceleration Data Graph of Vehicle device 71

Figure 20: Stage Diagram of the Comprehensive System 77

Figure 21: Data flow of the Comprehensive System 78

Figure 22: MQTT Broker General working Diagram 79

Figure 23: NodeRED and Mosquitto logo 79

Figure 24: NodeRed Flow for Two Devices 80

Figure 25: Overall Data Routing diagram 81

Figure 26: PHP Script Data Conditioning algorithm 81

Figure 27: Data stored and Database 82

Figure 28: Home Device Threshold checking algorithm 84

Figure 29: Vehicle Device Threshold checking algorithm 85

Figure 30: Data Flow for User Application 90

Figure 31: Mobile Application 91

Figure 32: Emergency Responder Web Dashboard 93

Figure 33: Emergency response Simulation for the notification system 94

List of Acronyms

IoT	Internet of Things
MCU	Microcontroller Unit
UART	Universal asynchronous receiver-transmitter
LPG	Liquefied petroleum gas
CNG	Compressed Natural Gas
SQL	Structured Query Language
GPS	Global Positioning System
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communication
ADC	Analogue to Digital Converter

Chapter 1

Introduction

In this era of information technology and automation, we are trying to automate another aspect of our need with implementing some IoT technology in our life. This chapter starts by discussing the motivation of the Thesis then moves on to introduce IoT in terms of its Architecture Layer, Literature Review, and Technologies. It then describes Network Topologies, finally concluding with Hardware Description. In the end, we discussed our proposed system and presented a systematic overview of implementing the proposed system.

1.1 Motivation of the thesis

Bangladesh is a small country in the south-Asian region. Besides many existing crises due to its crowding population, one of the most alarming problems it faces is an alarming rise in road and fire incidents, which cost thousands of lives every year. According to an article by Bangladesh Journals Online named "Road Traffic Accidents in Bangladesh." , regularly around eight people get killed in street auto collisions. The genuine pace of casualty is probably going to be considerably higher. The quantity of mishaps has expanded by 43% between 1982 and 2000, while the amount of fatalities has grown by around 400% inside the equivalent period. It builds more just before festival celebrations. Over the most recent 21 years, 84 thousand road accidents happened, 56 thousand individuals got killed, and 63 thousand were injured. The reasons that play behind road accidents are multifactorial. Fast urbanization and mechanization can be distinguished as fundamental components prompting a higher number of street traffic mishaps. On the other hand, fire incidents contribute no less to this sector. According to another article by Dhaka Tribune named "Fire incidents in Bangladesh triple over 22 years.", the quantity of fire hazards has expanded more than triple across Bangladesh since 1997; with the year 2018 saw an average daily incident of 53. Fire Service and Civil Defense measurements demonstrated that around 250,000 flames happened in the nation between January 1, 1997, and December 31, 2018, as indicated by the recorded database. These flames additionally caused an expected money related loss of around Tk6,400 crore to the country. One of the primary reasons for these accidents is the unconsciousness of people and often, the lack of awareness about it causes respective precautions. The aftermath of such accidents is even more catastrophic in terms of materialistic and infrastructure damages, and of course, the enormous loss of lives. In case of an accident, the response time of emergency services like fire, ambulance, and police services are unfortunately prolonged. This might be because of the delay in information being sent to the appropriate authorities (for example: calling an ambulance), heavy traffic and insufficient data regarding the accident, etc. This societal crisis has highly inspired our project development. Our goal is to minimize the time difference between the occurrence and the emergency response by automatizing the detection process of the accident and alerting the nearest hospitals for prompt medical assistance and Police or Fire Service when required.

1.2 Introduction to IoT

Internet of Things has been the most significant milestone to date in the world of technology. It is one of the buzzing concepts in the revolutions IT sector, which successfully took over the future of technology in a brief time and playing an absolute game-changer. Technologies considered profound when they camouflage and become indistinguishable in their field of application: in the workplace or everyday works. The numerous aspects of applications of the Internet of things made it widely famous technology within the industrial elements as well as day-to-day human life. Although its development has been initially driven by the needs of large-scale corporative tasks, at present, it has been modified to incorporate its benefits into building a smart societal system. The architecture of IoT According to the recommendations of the International Telecommunication Union, the network architecture of IoT consists of the sensing layer, the access layer, the network layer, the middleware layer, and application layers.

The phrase Internet of things, shortly known as IoT, is focused around two words-"Internet" and "Things." The internet is a global system consisting of interconnected networks that use the standard communication protocols (TCP/IP), providing a variety of information and communication facilities to serve of millions of private, public, academic, business, and government networks, both in the local and global scope, that are linked by arrays of electronic, wireless and optical networking technologies. The term "Things" identifies both living and non-living elements distinguishable by the real world. Objects typically identifiable by the internet are electronic devices accessed to online networks. Internet of things determines a broader span of things that are as well non-electronic, i.e., food, clothing, tools and types of machinery, animals, trees, cars, humans, etc. Due to its vast potential and revolutionary aspects that vary within scientists and developers, there has not been any unique standardized definition universally for IoT. The common theme of this technological concept can be summarized as- "An open and interoperable network of intelligent objects that can auto-organize, share information, data, and resources, reacting and responding to real-time situations and changes in the environment." With each passing day, the development of the Internet of things is the most dynamic, the concept updates with a more mature and sophisticated version. Within the past decade, its applications are not only confined within the commercial arena but fueled the vision of a global infrastructure where every physical element connect and communicate. It is now considered as a worldwide network that can connect human-to-human, human-to-things, things-to-things by providing every object with a unique, recognizable identity and allow to communicate among them. Physical objects are made identifiable within the network by fitting in sensors are actuators and are linked through a wired or wireless network. The network IP of the internet can connect the networks of the physical system as well. This hugely facilitates development in many social sectors, such as health, education, infrastructure, entertainment, etc. Internet of things made it possible to automize systems in the production line to healthcare to traffic management to home automation and a lot more places, which led the system to be much more time-efficient, cost-efficient, and free of human errors.

Internet of things focused on building a network of communication where each physical object can intelligently communicate and respond to the complex environment and changes without any human intervention. These networks accumulate vast volumes of data from the external environment and factors captured by the embedded sensors and actuators, which are processed through an analytical approach. These pieces of information are a vital part of designing a system algorithm in building automized networks where the objects can identify the physical signals and process the complexity of the external environment and respond logically. System automation means a system that can react intelligently to physical changes without human interaction, and that's where the Internet of things plays a unique role in interconnecting physical objects. It does it by coding and networking objects and them individually machine-readable and recognizable on the internet. Most of the current content in the Internet of Things has been created through coded RFID tags and IP addresses linked into an Electronic Product Code network.

1.3 IoT Architecture Layer

The concept of IoT is vast and multi-opinionated across the world and among different groups of concern. This has been a problem when it comes to looking for a unified architectural construction. For the idea of IoT to work, it must include the selection of sensors, networks, communications, and computing technologies, amongst others. According to the International Telecommunication Union (ITU), the architecture model of the Internet of Things is recommended as follows:

(a) The Sensing Layer

(b) The Access Layer

(c) The Network Layer

(d) The Middleware Layer

(e) The Application Layers

This layered architecture is like the open systems interconnection (OSI) reference model in network and data communication.

Sensing layer: This layer includes sensors, other hardware such as; embedded systems, RFID tags, and readers and others. The sensors establish a connection between the physical and digital worlds allowing real-time information to be collected and processed. The miniaturization of hardware has enabled robust sensors to be produced in much smaller forms, which are integrated into objects in the physical world. There is a variety of sensors designed to measure different physical data, such as temperature, air quality, movement, electricity, etc. In some cases, they may also have a degree of memory, enabling them to record a certain number of measurements. Sensors are adapted to take physical data and convert to machine-readable signals. Sensors are grouped according to their unique purpose, such as environmental sensors, body sensors, home appliance sensors, and vehicle telemetric sensors, etc. The main features of this layer are to capture data collected by various types of sensors, identify intelligently, and share the obtained information in the related units in the network.

The access layer: this layer's primary function is to transfer information from the sensing layer to the network layer through existing mobile networks, wireless networks, wireless LANs, satellite networks, and other infrastructure. Besides, it regulates Flow control & Reliability – Qos – Energy Optimization. Also, it performs cross-platform communication, if required. The

IoT web portal is in this layer. All decisions related to communications and measurements of the flow and its quality and energy consumed are made in this layer.

Network layer: this layer's primary function is to integrate the information resources of the network into a sizeable cognitive system with the Internet platform and establish an efficient and reliable infrastructure platform for upper-class service management and large-scale industry applications. It performs the following functions; Gateway – Routing & Addressing – Network Capabilities – Transport Capabilities – Error detection & Correction. Also, it takes care of message routing, publishing, and subscribing. With the demand needed to serve a broader range of IoT services and applications such as high-speed transactional services, context-aware applications, etc., multiple networks with various technologies and access protocols are needed to work with each other in a heterogeneous configuration. These networks can be in the form of private, public, or hybrid models and are built to support the communication requirements for latency, bandwidth, or security.

The middleware layer: this layer's primary function is management and control network information in real-time, as well as providing an excellent user interface for the upper-layer application. It includes various business support platforms, management platforms, an information processing platform, and an intelligent computing platform. Executes the following functions; Qos Manager – Device Manager – Business Process Modeling – Business Process Execution – Authorization – Key Exchange & Management – Trust & Reputation - Identity Management. In this layer, all actions related to the control, security, and management of the application are made.

Application layer: this layer's primary function is to integrate the role of the bottom system, and build the practical application of various industries, such as smart grids, smart logistics, intelligent transportation, precision agriculture, disaster monitoring, and distance medical care. This layer is at the top of the architecture and is responsible for the delivery of various applications to different users in IoT. The applications can be from different industry segments such as manufacturing, logistics, retail, environment, public safety, healthcare, food, and drug, etc. With the increasing maturity of RFID technology, numerous applications are evolving, which will be under the umbrella of IoT.

1.4 Literature review of IoT System and application

As the number of accident cases is increasing at a higher pace worldwide, researchers are focusing more on how to minimize the accidents or at least control the number of deaths causing due to road accidents. With the same thinking as us, many researchers have tried to develop a system that can detect car accidents and notify them of emergency services. Shivani S. and Sebastian S.[1] in their paper has built an accident detecting kit using the accelerometer and vibrating sensors for detecting and sending the signal to family members. But only using these sensors won't be that useful in sensing the fatality of the accidents and can be a wrong call for emergency assistance. Patel [2], in his paper, developed an android system that can detect vehicular accidents using only the accelerometer, which is also cannot give proper data or information to know about the fatality of the accident and whether it requires any emergency medical assistance. Thompson C. [3] has proposed a system that uses sensors in the smartphone to detect accidents. The smartphone uses the cellular internet connection to recognize the accidents and sends the signal information. The major drawback of the system is it is not fully automated at times, and one must manually assign the accident information to the desired receiving ends which certainly delays the process. E. Krishna Priya, P. Manju [4] developed a system that can detect an accident and sends the signal to hospitals using an accelerometer and ultrasonic sensor and Atmega162 microcontroller. In their order, it is difficult to estimate the fatality of the accident due to a lack of data from a limited number of sensors. This might result in the wrong call for emergency assistance. Though they have put an option of pressing a switch to terminate the alert signal to the emergency response service, it is not reliable as it is manual, and the person might not be in the conscious mind to think about pressing the switch. Chaturvedi N. and Srivastava P. [5] established a system that will detect accidents and sends alert messages using GSM modem to the hospital, police station, and relatives of the victim. In their paper, they have not mentioned any specific sensor which will detect the accident, making the system very unclear to the readers and researchers. Mane A. and Rana J. [6] have tried to give a solution to the ongoing problem of deaths due to road accidents in their country. They build a system with a Micro Electromechanical System (MEMS) sensor and vibration sensor to detect vehicle accidents and sends the signal to emergency response like hospitals and police stations. GPS modem is incorporated into the system for knowing the exact accident location. Md. Schedule, Jalil J, and MBI Reaz [7] have proposed a system that will detect an accident and send the alert message to the rescue services, especially ambulance using GPS, GPRS, and GSM technology. The speed of the vehicle will be continuously monitored and compared with the previous rate through a microcontroller. If the speed goes under a certain value it will assume that an accident has occurred.

Fire accidents in homes are becoming a significant concern worldwide. The more worrying factor is the delay in emergency services. Based on that, many researchers are trying to minimize the delay for emergency services to arrive and take necessary actions. Apart from us, many researchers worldwide have also made approaches to dealing with this alarming issue. R.Angeline, Adithya S., and Narayanan A. [8] in their paper have proposed a system that detects fire incidents and sends them to the emergency service required. The drawback of this system is the message to the fire service is not immediate. The signal first goes to the watchman, and upon verification, the watchman sends the message to the fire service. This whole process will take a bit of time, and the emergency response will be slower. E. Saraswathi and Kumar A. [9], in their paper, have showcased an Arduino based home automation system to detect fire incidents and sends the signal to the fire service. The disadvantage of the process is energy control depends on specific optimum power consumption, which might not be available to different consumers. Johnsaida N. ,Lakkishetty V. Rahul, and Shalini T. [10] have developed a fire emergency response system with only one flame sensor, which will not be valid and have more chance of false call for emergency service. M. Samarasinha Reddy and K. Raghava Rao[11] has designed a fire accident detection and prevention monitoring system by using temperature sensor LM35 and flame sensor to detect fire. LM 35 might not be useful at times, and with a limited number of sensors and data, the emergency call for assistance might not be correct all the time. Imteaz A., Rahman T. [12], in their paper, has proposed a system that will detect fire incidents and sends an emergency signal to fire brigade when necessary. The system is mainly developed for industries and factories in the country. The system comprises of microcontroller, several sensors, and a camera. Upon detecting fire by the sensors like gas sensors, flame sensors, the signal is sent via GSM modem with an image of the spot captured by the camera. Then drawback of the system is before the emergency signal is sent, there must be confirmation from an admin, which makes the whole process lengthy. Gosrani S., Jadhav A. [13] has developed a system using Arduino, which will detect fire and sends an alert message to the fire station, hospitals with the data collected by the system. The system is built with multiple sensors: flame detection sensor (PT33b Infrared sensor), gas sensor (MQ-6), and temperature sensor (LM35) to detect fire incidents. Kodali and Ravi [14] have developed a system with a Wi-Fi microcontroller and sensors like smoke detection sensor (MQ-2), flame detection sensor, and flammable gas sensor(MQ-5) to detect fire in households or industries. GPS module is also incorporated into the system. So, whenever any fire incident occurs, the alert message goes to the cloud service with the location. In the cloud service, all rescue services are connected like fire departments and police. By getting the alert message, fire service will take necessary measures.

1.5 IoT Technologies

RFID:

Radio Frequency Identification is a system that transmits the identity of an object or person wirelessly using radio waves in the form of a serial number [15]. RFID device was first founded and used in the second World War in Britain, and the identification of a friend was first used in the year 1948. In 1999 the RFID technology was first introduced to the world by MIT at Auto-ID centre. This technology contributes significantly to the IoT sector. In IoT, the primary role of RFID technology is to identify people or objects with a GPS module and other necessary sensors installed in them. RFID can be classified into three types: Active RFID, Passive RFID, and Semi Passive RFID. One of the significant advantages of RFID technology is it is very cost-effective. The other benefits include good accuracy of results and reliability. RFID has a wide range of wireless applications such as patient monitoring, distribution, tracing, military apps, etc. [16].

Internet Protocol:

Internet Protocol (IP) is a way by which data can be sent from one computer to another computer on the internet. Each machine, which is termed as host, has at least one IP address, which identifies that it is different from all other computers on the internet. It was first developed in the year 1970. There are two versions of Internet Protocol, which are commonly used: IPv4 and IPv6. Each version has its IP addresses, which are different from each other. In IPv4, there are five classes of varying IP ranges. They are Class A, Class B, Class C, Class D, and Class E. From these, only Class A, B, and C are most readily used. IPv6 is the latest version of the Internet Protocol and is often called the 21st century Internet Protocol. The actual protocol provides for 4.3 billion IPv4 addresses, while the IPv6 provides availability to more than 85,000 trillion addresses [17].

Electronic Product Code (EPC):

Electronic Product Code is an RFID code which contains specific products' information which may include manufacturers name and other unique details related to the product. EPC was first developed in the Auto-ID centre at MIT in 1999. Since its development, it has been readily used worldwide. EPC has four different sections within it: header, EPC manager, Object class, serial number. The header, which is usually 8 bits, is the number that corresponds to the EPC format used by the tag. EPC manager, which is generally 8-35 bits long, is the unique number which is given by the EPC Global to the company which is manufacturing their products. Object class, which is 39-56 bits long, is the specific product number for a product made by a company. This number is always different for different product classes. The fourth and last section is the serial number, which is 69-95 bits long is the number given by the company to a different product. Each has its unique serial number. EPC consists of four components, which are Object Naming Service (ONS), EPC information service (EPCIS), EPC Discovery Service (EPCDS), and EPC Security Service(EPCSS).

Barcode:

A barcode is a way of representing specific products that are machine-readable. It can be scanned by optical scanners, and now with the invention of smartphones, it can be scanned via a barcode scanner application through cameras. It can also be defined as the series of parallel black lines of varying widths separated by white spacing. Every product has its pattern of parallel lines, which consists of the product's specific information. In supermarkets, the barcode is mostly used to identify the product by scanning the product with optical scanners and storing the data into the computer system. The invention of barcode has been a massive revolution for supermarkets and retail stores for identifying and storing the sold product information into the computer system. Recently another kind of barcode has been in use, which is named Quick Response (QR) code. As the name suggests, the QR code has fast readability and faster than all the other data entry methods. Barcode is usually of three types: Alpha Numeric, Numeric, and two dimensional. Among the types of barcode 2D barcodes are the most exclusive ones. They can store more data compared to other barcode types. 2D barcodes are also errored protective. Even if there is an error or is damaged, the data is still error-free and can be scanned. The QR code that is mentioned before is a type of 2D barcode.

The Barcode and EPC can be distinguished by some key points such as:

Features	Electronic Product Code (EPC)	Barcode
Mode of transmission	It uses radio waves to transmit data utilizing RFID technology.	It needs optical recognition by optical scanners to transmit data.
Data modification	EPC has an electronic memory and can be dynamically updated.	The information printed on a barcode is permanent and cannot be modified.
Scanning.	The scanning process is automatic.	It needs manual scanning.

Wireless Fidelity (Wi-Fi):

Wi-Fi network is a wireless network that is connected to a router, and devices like phones, laptops, and other smart devices can be connected to the internet via Wi-Fi. The concept of Wi-Fi was first established in 1990 in the research group of IEEE 802.11 Working Groups of Wireless LANs and was led by Victor Hayes. He is also known as the father of Wireless Fidelity (Wi-Fi). The emergence of the Wi-Fi network has outcast the use of wired networks. Wi-Fi networks communicate using radiofrequency. After the invention of the Wi-Fi network, it has been readily accepted by the people. It is now supported by almost every smart device, video game controllers, major operating systems, etc. Wi-Fi is also playing a significant role in the IoT world. Many IoT devices and systems are using Wi-Fi technology as their communicating medium. The merits of Wi-Fi technology are vast in number. The installation is swift and easy. Experts are not needed for the installation of the network. Only basic knowledge can get the job done. It is very cost-effective as many users can use at a time with a reasonable monthly subscription based on the internet limit. The data connection is also high-speed.

Bluetooth:

Bluetooth is a type of wireless technology that transfers data from device-to-device within a short-range, usually within 100 meters. Bluetooth is cheap compared to other data-sharing technologies. The data that Bluetooth can transfer are text, pictures, videos, and audios. The term Bluetooth was first introduced by Ericson Mobile Communication Company in 1994, aiming to build a Personal Area Networks (PAN). Within a short period, Bluetooth became famous worldwide for its fast and hassle-free transmission of data. At that time, smartphones were not developed, so Bluetooth was the only means of transferring data from one device to another. Cellphone companies started to get the Bluetooth technology in almost all their handsets. As years passed, Bluetooth was not limited to only cellphones but also computers, laptops, headphones, keyboards, and mouse. With the advancement of technology, Bluetooth is also used in smart home applications and Remote-Controlled Cars contributing to robotics and smart home development. The only drawback of Bluetooth is they have a short-range for data transmission.

Wi-Fi and Bluetooth can be distinguished by some key features, such as:

Features	**Bluetooth**	**Wi-Fi**
Purpose of designing	To connect short-range devices for sharing data.	Wi-Fi offers high-speed internet access through a router.
Range	Range of transmission is around 10 meters.	Range of transmission is 100- 200 meters.
Flexibility	A limited number of users can be connected at a time.	Greater number of users can be connected.
Bandwidth	Low Bandwidth	High Bandwidth
Power Consumption	Power Consumption is low.	Power Consumption is high.

ZigBee:

ZigBee is a wireless sensor network specified for communication in a Wireless Personal Area Network (WPAN). WPAN is a network enclosed in a limited place, usually home or office space leading to the fact that ZigBee has a short-range for transmission. ZigBee technology was developed in the year 2001 by the ZigBee Alliance. ZigBee technology mostly works in a mesh topology, but it can also work in star and cluster tree topology. Mesh topology is most suitable in the sense that if one computer stops working, then other computers can communicate with each other. It has a little cost and low data rate. Therefore, installation and maintenance are also cheap. It is also low powered, and a network protocol based on the IEEE 802.15.4 standard. There are several applications of Zigbee, which include home automation, industrial automation, medical monitoring system, which is patient monitoring and fitness monitoring, power systems, consumer electronics such as television, DVD, CD, remote control, etc.

Near Field Communication (NFC):

Near Field Communication (NFC) is a short-range wireless technology having a frequency of 13.56MHz. NFC technology came from RFID technology, which means they use radio frequency for transmission. It transmits data like pictures, videos, and contact information within two NFC supported devices. It needs a distance of 4-10 cm for the proper transfer of data. The data exchange rate is approximately 424 kbps. NFC was first introduced by Philips and Sony companies. NFC can be another data-sharing medium to Bluetooth with an added number of advantages. Unlike Bluetooth, it doesn't need any pairing of devices; it simply gets connected automatically when two NFC supported devices are within the radius of transmission. However, it will need permission from the user before transmission of any data. Another application of NFC is digital money transactions and serves as an alternative to credit cards. The security is high, and the technology is reliable, and so people have started considering NFC as a money transaction medium. Moreover, it can also work in extreme condition and are easy to connect and install.

Wireless Sensor Networks (WSN):

Wireless Sensor Network (WSN) is a wireless network comprising of the base station and some nodes which are wireless sensors that monitor physical or environmental conditions, such as temperature, pressure, vibration, sound, heat, light, etc. and forward the data through the network to the main location. The output of the signal is an electrical signal which goes to a controller for further processing. Though recently, the use of WSN has been enormous throughout the world, it was introduced a long time back in the year 1967 during the Vietnam war. It was developed by the US military forces during the war. The importance of Wireless Sensor Network is huge in the IoT world. There are several applications of WSN in the IoT sector. It works in star, mesh, and tree topology. There are five types of Wireless Sensor Networks (WSN): i) terrestrial WSNs ii) underground WSNs iii) underwater WSNs iv) multimedia WSNs v) mobile WSNs. The applications of WSN based on IoT are distributed in many areas such as military, homeland security, healthcare, manufacturing, habitat monitoring, forest fire, and flood detection, and so on[18]. It can also be used in the power sector, such as in smart grids and energy control systems, transportation sectors. Besides, WSN contributes to smart city and smart home development.

Actuators:

The actuator is a device that converts energy to motion or eventually to a mechanical system. The source of power for this device is electric, hydraulic fluid or compressed air. The most common examples which are in use are electric motors, solenoids. It has a short range of operation for a distance of about 30 feet. There are three types of actuators: i) Electrical actuators: electric motors and solenoids. ii)Hydraulic actuators: use hydraulic fluid to generate motion. iii)Pneumatic actuators: use compressed air to generate motion. Hydraulic and pneumatic actuators are ancient devices founded in the year 1938 in World War II. Since its invention, it has been in use still today. Electric Actuators came a bit late between the 1960s and 1970s with the uprising of the oil and chemical industry. The most commonly used actuators today are electrical actuators. It has several advantages which include better accuracy, less noise, less environmental damage. Electric actuators are costlier than the other actuators and might overheat due to long use. The features of hydraulic actuators are high horsepower-to-weight ratio, loss of pressure is minimum, can hold force constant for a longer period. The drawbacks of hydraulic actuators are chances of leakage of fluids, need additional parts for functioning which can be motors, pumps, or noise emitting components. The pneumatic actuator is a cheaper option compared to other actuators. It works fine at a wide range of temperatures, less hazardous. The drawback of the pneumatic actuator is similar to hydraulic actuators that is they need extra components for proper functioning, which includes valves, regulators, and compressors.

IoT technology for the designed system

Home Device

The main motive behind developing the home kit is to keep specific internal environment factors under constant monitoring. The collected data is needed for further speculations of hazards or take any automotive measures if there any accident takes place. Nowadays, most families have access to Wi-Fi in their houses, which encouraged to design the kit to connect over Wi-Fi. The IoT circuit includes NodeMCU that establishes the connection between the kit and the available Wi-Fi network. The data that is continuously feeding into the Arduino needs to be sent to a remote server from it will be further interfaced with other links. Wi-Fi allows prompt access to the internet without the hassle of a specific access point and connects at almost every corner of the house without human involvement. Besides, its budget-friendly and high-speed connectivity make this technology highly productive for the kit.

Vehicle Device

In contrast to the Home kit, the Vehicle kit is more sophisticated in terms of its features and circuit combination. Vehicles keep on changing places now and then, so the kit needs to be supported with a secured internet connection that can follow with it to every location. Since the kit designed for mobile objects like vehicles, it uses a GPRS module to get access to sim operated internet connection. It supports Mobile Communication Internet Protocol to keep the system online and exchange data that is fed into the Arduino from the embedded sensors to the respective server. GPRS replaced the wired network with a more simplified system to access the packet data networks like the internet. The packet radio principle is employed by GPRS to transport user data packets between GSM mobile stations and external packet data networks. Its open architecture, convenient billing policy, and consistent IP services with a fast data rate perfectly blend with the system requirement.

1.6 Network Topologies

Network Topology refers to the format of a network and how distinctive hubs in a network are associated with each other and how they communicate. Topologies are of two types: physical (the physical format of devices on a network) and logical(the way that the signals interpret within the network or the way that the information is exchanged through the network from one device to the other).

Types of Network Topologies are as follows:

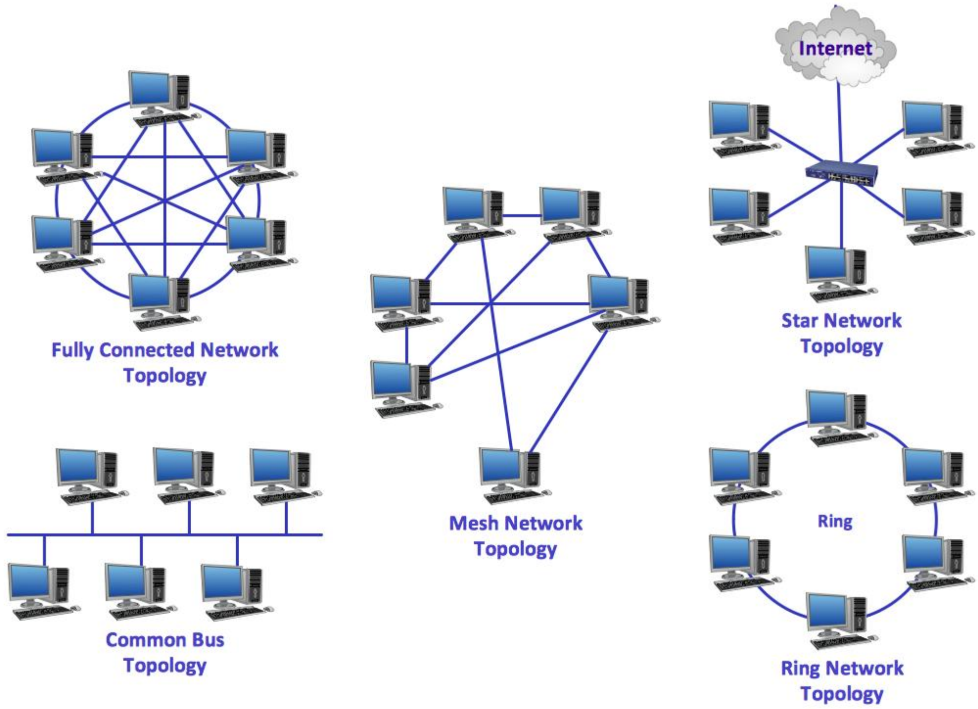


Figure 1: Different Types of Network Topologies

1.Star topology- In star topology, each device within the network is associated with a central device called the central hub. Not at all like Mesh topology, star topology doesn't permit coordinate communication between devices; a device must need to communicate through a central hub. On the occasion that one device needs to send data to another device, it must begin by sending the information to the central hub. After that, the central hub transmits that information to the targeted device.

2.Ring Topology- A ring topology is a system set up where devices are connected in a roundabout way. Each device is associated with two others, similar to focuses on a circle. Together, gadgets in a ring topology are alluded to as a ring system. In a ring system, packets of information pass from one node then onto the next until they arrive at their goal. Most ring topologies permit parcels to travel just one way, called a unidirectional ring system. Another license information to move in either bearing is called bidirectional.

3. Bus Topology-Bus topology is the sort of system topology where each gadget on the system, is associated with an independent central link line. Information is transmitted in a solitary course, from one point to the next. We can't send data in two ways. At the point when this topology has exactly two endpoints, it is known as Linear Bus Topology. It commonly used for small systems.

4. Mesh topology- Mesh topology is the sort of topology, where all the devices are channelled with one another. Work topology is a point-to-point association. It has n(n-1)/2 system channels to interface n devices. It has two strategies for transmission of information: steering and flooding. In the steering strategy, it follows a logic that figures out the minimum possible route, to send out the data to the designated hub. In the flooding system, all the system hubs get similar information, leaves no requirement for directing algorithm.

5.Hybrid Topology- It is a network topology that combines two or more topologies to form a network system. It is a dependable and adaptable topology; however, at the same time, it is costly. It gets the benefits and negative marks of the topologies used to fabricate it.

Our devices in this project will follow the Star topology based on the functional requirement.

Few merits of Start Topology are listed below:

1. Less costly since each device required one I/O port and should be connected with the central hub with one link.
2. Simpler to install
3. Less sum of cables required since each device is linked to the central hub only.
4. Strong, if one interface falls off, the other joins will work fairly fine.
5. The link with disruption and error can be easily identified.

Few demerits of Star Topology are listed below:

1. On the off chance that the central hub goes down, it halts the entire system as the devices cannot work without it.
2. Central hub requires more resources and routine maintenance since it is the central framework of star topology.

The embedded sensors and actuators in the home and vehicle kit accumulate data from the physical environment and feed to the Arduino through multiple ports. Arduino sends the data (converted to digital signal) to the server immediately. The server sends out the data to the web dashboard for further processing and interpretation. The goes through a formative analysis within the dashboard, and the final interpretation is sent to the server. Different thresholds are for different measures. The final signal from the server is sent out to the targeted nodes, which then responds according to the system logic.

1.7 Thesis Structure

The title of the paper clearly reflects on the idea that will be represented in this paper that is to design an integrated accident detection system cultivating the concept of IoT.

The first chapter is designed on the discussions on the revolution of the Internet of things and the aspect that motivated the work on this topic. It is composed of subchapters that consecutively discuss the architectural layers of IoT: that has been followed to design the system, some significant past works in the field, IoT technologies, network topologies and descriptions of the related components that have been used.

Chapter 2 and chapter 3 have been designed is a similar manner that delivers in-depth knowledge and analysis of the design of individual parts of the integrated system. Chapter 2 introduces the first one of two devices we built called Home Device. It is aimed to monitor House or kitchen environment. A design process is shown in the chapter, along with the collected data sample of sensor data and analyses for future simulations. Chapter 3 introduces the second device named the Vehicle device that is designed to identify vehicle accidents and some small signs of malfunctions of the engine. The build process Vehicle device followed by the collection and analyzation of sensor data of the vehicle device is also shown in this chapter.

Chapter 4 explores the comprehensive design of our IoT system. It discusses how the different devices work in union and are connected to a single server system for data and notification processing in detail with several simulations of possible emergency situations based on data characteristics collected from chapters 2 and 3.

Finally, in chapter 5, we concluded the thesis with some discussions on the limitations, drawbacks, and future work possibilities of our system.

Chapter 2

The IoT Based Home Device

This chapter focuses on the Home Device. It starts by discussing the literature review and then breaks down the design of the device in terms of the components used and the reasons behind it. This is followed by the hardware connection, which breaks down the connection diagram, schematic diagram, PCB layout followed by the implemented picture of the prototype. Next comes the working mechanism which discusses the data sensing mechanism, Arduino code, Wi-Fi connectivity mechanism, and NodeMCU Code. Finally, results and discussion elaborate more on the sensor data and its characteristics and concludes by discussing the cost analysis.

2.1 Literature Review

Accidents involving fire in households is a common issue that has existed since aeons. Therefore, preventive measures to put out small fires were invented as early as the 1800s. Inventors first began experimenting with automatic sprinklers around 1860. The first automatic sprinkler system was patented by Philip W. Pratt of Abington, Massachusetts in 1872. This was followed by the first smoke detector that was invented in 1970. There are various sensors available in the market today that can be used to build integrated systems for fire monitoring, detection, and prevention. Various gas sensors such as the ones we used in making our kit, MQ2, MQ8, MQ135 can be used to detect alarming levels of Methane, Butane, Propane, Carbon-dioxide, Carbon monoxide in a given environment. Then there are flame, temperature, and humidity sensors as well to detect the presence of a fire or an alarming increase in temperature. Our home kit was designed with the idea of bigger fire accidents in mind. Most of the modern-age homes today are designed with keeping fire hazards in mind, so most of them have built-in fire alarm systems with sprinklers to put out small fires in the kitchen, etc. Also, there are fire extinguishers available on every floor of a building. But when these small fires turn into something much bigger before they have had a chance to be put out by built-in systems, fire stations must be notified. If that notifying process is dependent on a manual system, then it poses a huge threat to property and more importantly, potential lives involved in the incident. That is where the novelty of our kit being IoT based automated emergency response provider comes in.

There have been quite a few research papers with an idea similar to ours. One of which proposed by Su [19], uses a multisensory fire detection algorithm. A firefighting robot is designed with three flame sensors inside that helps in fire detection. If a fire accident is true, the robot can find out fire source using the proposed method by the fire detection system and move to the fire source to fight it using an extinguisher.

In a different and improved approach proposed by Cho et al. [20], image processing is used to detect fire using a statistic colour model. In this paper, an automatic fire detection system without the heuristic fixed threshold values was studied. An automatic method using the statistical colour model and the binary background mask was used instead. The experiment was done using 600 frames from 6 typical different fire video clips, that showed promising results. The proposed method showed a good performance of about average 85% detection rate without false positive, compared with the other methods with the heuristic fixed threshold values.

In another paper developed by Imteaj et al. [12], a similar approach was taken with the replacement of Raspberry Pi 3. Raspberry Pi 3 has been used to control multiple Arduino which are integrated with a couple of sensors and a camera. A 360° relay motor is assembled with the camera so that it can snap the image in whatever angle the fire is detected. A confirmation of the fire suspecting system is provided to avoid any false alarm. The system will immediately send a message along with the image of the affected spot and Arduino's location. An admin can confirm or deny the impeachment, and if the admin confirms the situation as breaking out of the fire, then the system will immediately raise the alarm, and an automatic message will be sent to the nearby fire brigade. In this paper, there must be the involvement of an admin to determine whether to respond to the fire or not. This is where the novelty of our kit is reinforced as the notification is directly sent to the fire station without seeking manual approval of any sort.

The paper that was most similar to ours except for the notification process was the one proposed by Saeed et al. [21] who also used a GSM module. In their paper, they designed and evaluated a wireless sensor network using multiple sensors for early detection of house fires. Besides, the Global System for Mobile Communications (GSM) was used to avoid false alarms. To test the results of their fire detection system, a fire was simulated in a smart home using the Fire Dynamics Simulator and a language program. The simulation results showed that the system could detect early fire, even when a sensor is not working while keeping the energy consumption of the sensors at an acceptable level.

Muhammad et al. [22] concluded in a paper that fire could be detected using a vision-based system during surveillance using convolutional neural networks (CNNs). However, such methods generally need more computational time and memory, restricting its implementation in surveillance networks. This proposed method, although it does not involve IoT, but is intended to fight the same issue as us.

The novelty of our system compared to other systems mentioned above, is we are considering multiple factors to determine any fire incident. The multiple factors include temperature, humidity, flame, and levels of different gases. We measure the levels of these multiple parameters by specific sensors dedicated to each. In other papers, they are mostly working with one or two sensors making the system unreliable and also has some major drawbacks already mentioned. As the home device is part of a comprehensive system, multiple devices' data can be monitored from a single app and also give notification through a single app.

2.2 Home Device Design

The purpose of the Home device is to send the Gas ppm values, temperature, and flame detection data to the server. So, the possible fire or any kitchen gas leaked can be detected and take necessary steps to prevent possible disaster or response to an unlikely scenario that happened already. As almost every modern home has fire sprinkler which works only on heat emission, in design consideration, we assumed house has fire sprinkler already installed, and we will give the IoT support.

2.2.1 Component Selection

The components for the Home device consist of a microcontroller unit, three gas sensors, one temperature sensor, one flame sensor taking gas data, and another microcontroller to send the data to the server via Wi-Fi.

- Arduino Nano
- NodeMCU
- MQ-2
- MQ-4
- MQ-135
- DHT11
- IR based Digital Flame sensor
- 5V Wall power adapter

2.2.2 Component Description

The components used here all project grade components to minimize the system development cost. Here are the components:

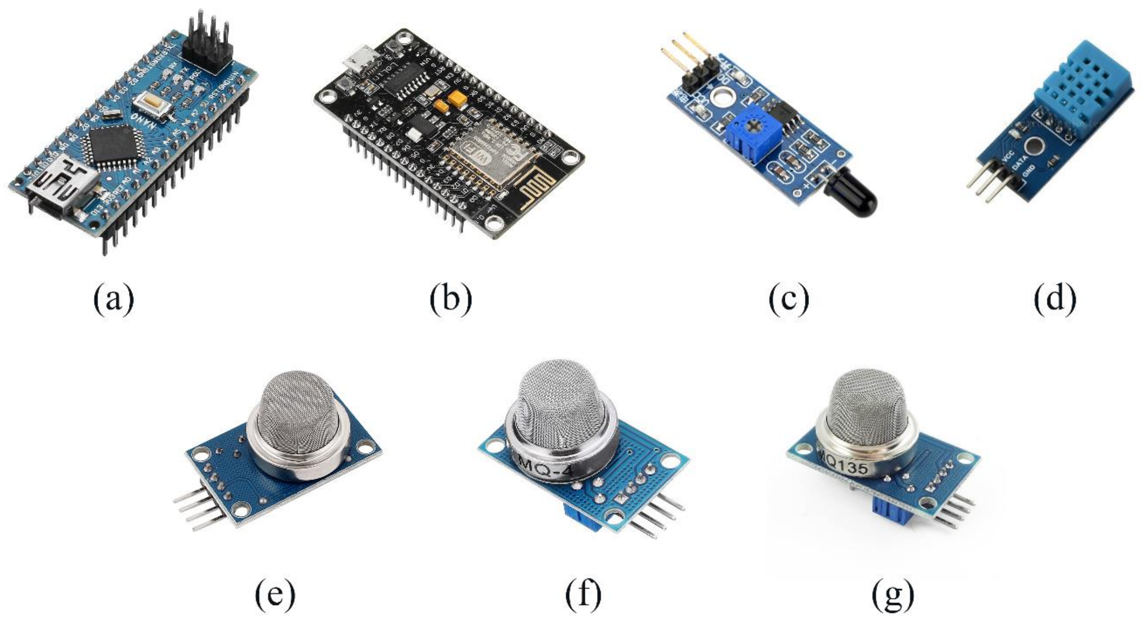


Figure 2: Home Device Components. (a) Arduino Nano, (b) NodeMCU, (c) Flame Sensor, (d) DHT11, (e) MQ-2 gas sensor, (f) MQ-4 Gas Sensor, (g) MQ-135 gas sensor

Arduino Nano

Arduino Nano is a microcontroller of the Arduino series, which is much smaller in size than the other boards based on Atmega328p. This board is also breadboard friendly. The features are quite like Arduino UNO, but the size is much smaller with a limited number of digital and analog pins. It has 14 digital pins, eight analog pins, two reset pins, and six power pins. There is only one way to power the board, which is the mini USP support and also used for uploading the code from the Computer. DC power jack is not present in Arduino Nano. The crystal oscillator is the same as other Arduino boards, which is of frequency 16MHz. The flash memory is 16 KB or 32KB, depending on the AT Mega board. SRAM can fluctuate from 1KB to 2KB, and EEPROM can be 512 bytes or 1 KB.

NodeMCU

NodeMCU is an Arduino like hardware IO and Lua based IOT platform, which is specially made for ESP8266 Wi-Fi chip. This hardware codes like Arduino but interacts in Lua script. One single development board consists of 10 GPIO, I2C, 1-wire, and ADC. There are numerous numbers of features of this development kit, which makes it popular in the IoT sector. The features include cost-effective, smart, Wi-Fi enabled, open-source, interactive, and programmable.

MQ-2

MQ-2 sensor is a gas sensor that detects levels of various gases in households or factories. The gases which it can detect are methane, carbon monoxide, hydrogen gas, LPG, smoke, and alcohol. The range of concentration level it can detect is from 200ppm to 1000ppm. The concentration level of different gases can be detected by varying the resistance of the sensing material of the sensor. By voltage division rule, the levels of various gases can be known. Then the resistance of the sensing material can vary from 10kΩ to 60kΩ. A potentiometer is generally used for adjustment of the sensitivity of the sensor. The major advantage of this sensor is it has a longer lifespan and has a wider range for detection. The other features include quick response and high sensitivity.

MQ-4

MQ-4 sensor is a gas sensor that detects concentration levels of different gases in homes and industries. MQ-4 gas sensor mainly detects combustible gas like methane and also natural gas. The range of concentration level it can detect is 300ppm to 10000ppm (According to Datasheet). This sensor detects the levels of methane gas in the atmosphere and gives the output as an analog voltage. It also has a built-in ADC which converts the analog signal to digital. Therefore, it can give output value as an analog and also digital. There is a sensing material inside a steel-plated mesh that detects the difference in the concentration level of the gases and changes the resistance accordingly. Upon the change, resistance to the current flow and potential difference is varied, which represents the concentration level of different gases. The sensing material is made of tin dioxide (SnO_2). The operating voltage of this sensor is 5V DC. There is a special feature of this gas sensor. It can give both analog and digital output. Other features include longer lifespan and quick response.

MQ-135

MQ-135 is another gas sensor we used, which is generally used for detecting different chemical contents in the air. Generally, with MQ-135 sensor pollution level in the atmosphere is determined. The gases it detects are NH3, alcohol, benzene, Carbon dioxide, etc. The sensing material is tin dioxide (SnO_2). By varying the sensing resistance of the material with the help of a potentiometer, the concentration level of gases can be determined. The sensing resistance varies from 30kΩ to 200kΩ. The operating voltage is 5V DC. The features are similar to other gas sensors, which are low cost, longer lifespan, wide range detection, and quick response.

Flame sensor

A flame sensor is a sensor that can detect fire or flames within the range within the range of 100 cm distance. The output value that this sensor is giving can be analog or digital. This sensor is widely used to detect fire beforehand and notify or warn people in homes, offices, or industries. There is an electromagnetic radiation receiver at one end of the sensor which receives infrared radiation from the fire. The sensor is worked through a layer of oil, dust, and water vapour. The operating voltage varies from 3.3V to 5V. The advantages of this sensor are high photosensitivity, quick response time, and adjustable sensitivity. It is useful in fire alarms in houses and industries and industrial machinery involving heating at high temperatures.

DHT11

DHT11 is a type of humidity and temperature sensor. They can detect and measure humidity and temperature by separate methods. The humidity sensing element is used for measuring humidity, and a negative temperature coefficient thermistor is used for determining temperature. DHT11 can measure temperature from 0 to 50 degrees Celsius. The range of humidity it can measure is 20% to 80%. The operating voltage of this sensor is 3V to 5V DC. This sensor can interact with microcontrollers like Arduino and raspberry pi. A pull up resistor of 5kΩ to 10kΩ is present for communicating with microcontrollers. Due to its small size, low cost, and high accuracy rate, it is widely used.

5V Wall power adapter

To give power from a wall outlet.

2.3 Hardware Connection

All the sensors are connected to Arduino Nano as input. The microcontroller takes the data and sends it to the NodeMCU via serial connection. And the NodeMCU sends that data to the server.

2.3.1 Connection Diagram

Here shows the Connection Block Diagram

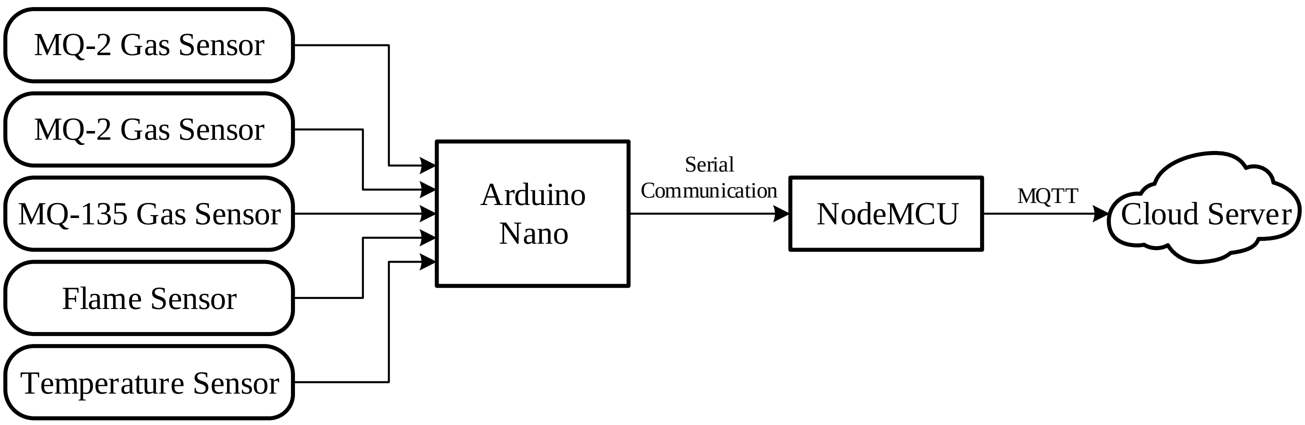


Figure 3: Home Device Connection Diagram

2.3.2 Schematic Diagram

Here shows the Schematic of the home device. It shows the pin by pin Component connection of the home device

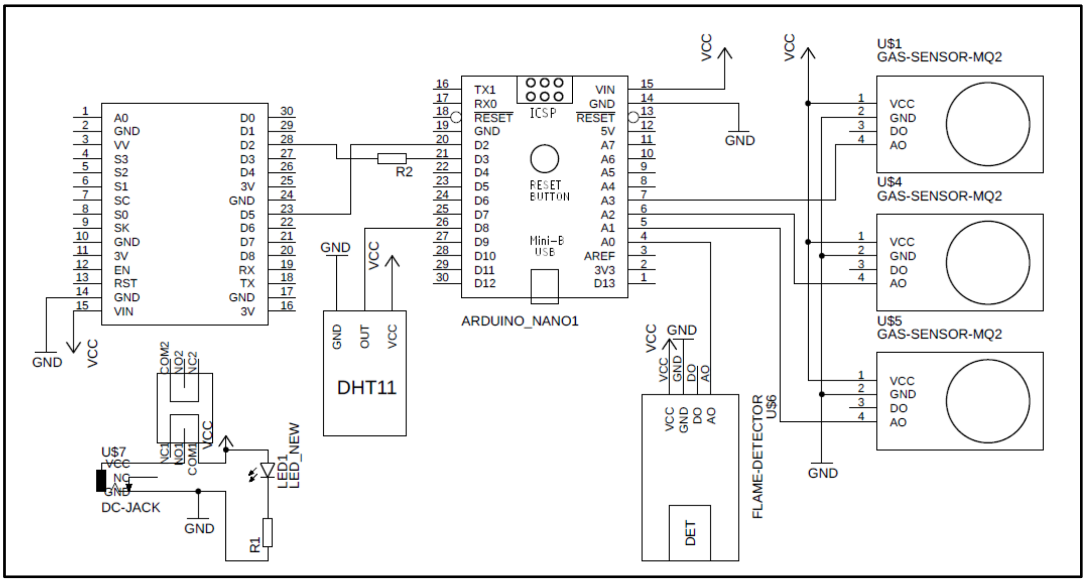


Figure 4: Home Device Schematic Diagram

2.3.3 PCB Layout

Here show the PCB Layout and component placement of the home device, which is build based on the schematic diagram shown previously. This PCB board is used to build our Home Device prototype.

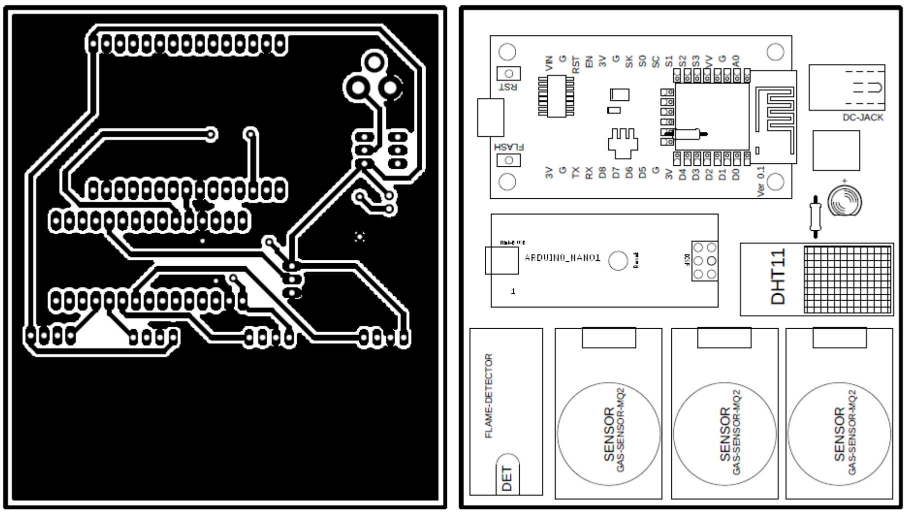


Figure 5: Home device PCB Layout

2.3.4 Implemented Picture

Here shows the Prototype of the home device build from previously shown schematic and PCB.

This prototype is used for our all home device data collection.

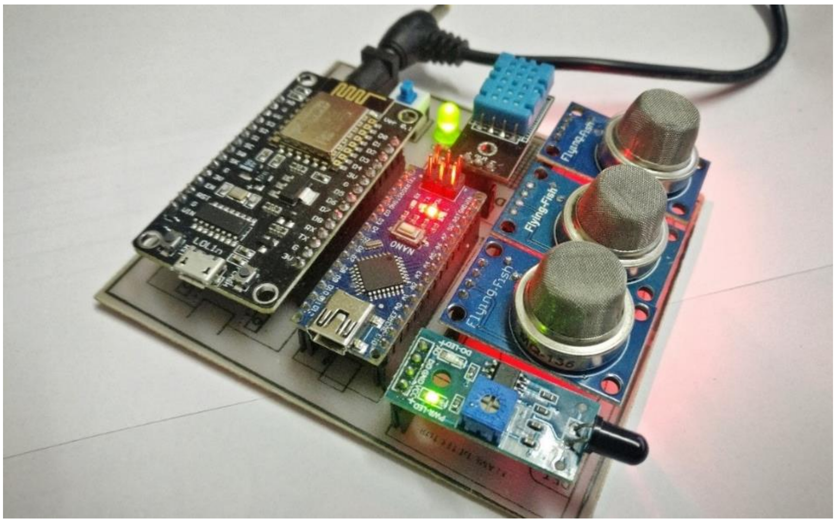


Figure 6: Prototype of Home Device

2.4 Working Mechanism

In this section, we will discuss the working mechanism of the Home device until the data sent to the server. Before sending the data, we will collect the data and check some basic conditions and send this to the server.

2.4.1 Data Sensing Mechanism

Data Sensing happens in 2 steps,

- Raw data sensing
- Conditioning

First of all, Arduino read the MQ2, MQ4, MQ135, DHT11, and Flame sensor data. These data reads happen in several ways. For MQ gas sensors data, we used ADC reading by MCU, then the corresponding Arduino library process the ADC value with the resistance value of the sensor to get the ppm values and those stores in a variable. Then the Corresponding DHT11 library decodes the temperature values sent from the DHT11 temperature sensor. DHT11 sends a digital signal to the MCU which needs to be decoded. The flame sensor does not have to need any library; it changes its analog signal value with the flame IR reading. It can give different values to a different amount of flame detected.

Secondly, the flame data is checked with a previously stored sensor value to check if there is any fire, that turned a flag ON so that further processes can be done by emergency responder immediately.

Finally, all the sensor data, along with the flag to determine emergencies, pre-determined GPS location of the device, and a unique device ID is cascaded into a string and send that to NodeMCU via serial communication for further processing.

2.4.2 Arduino Code

First of all, we used some library, to take the sensor data more efficiently. We used DHT, mq2, mq135 to take sensor data, and used SoftwareSerial to transfer the data to NodeMCU to send it to the server.

```
#include <SoftwareSerial.h>   //library for serial communication
#include "DHT.h"                  //library for temperature
#include <MQ2.h>               //library for mq2
#include <MQ135.h>
```

After that, we defined the variables and parameters, and the fix gps address for the location. We defined the pins and created different function to take different sensor value.

```
void getDHT(){
  temperature = dht.readTemperature(); // take temperature
}
void getMQ2(){
  lpg = mq2.readLPG(); //take mq2 reading
  co = mq2.readCO();    //take CO reading
  smoke = mq2.readSmoke();  // take smoke reading
}
void getMQ4(){
  mq4 = analogRead(mq4_pin); // take mq4 reaing
}
void getMQ135(){
  mq135 = analogRead(mq135_pin); //take mq135 reading
}
void getflame(){
  flame = analogRead(flame_pin); //take flame sensor reading
}
```

In the main loop, we take all the sensor values and merge them into a single string to send to the NodeMCU. Additionally, we add a variable named Alarm, and it checks the flame sensor value to find immediate flame, and in the server, this alarm value will be crosschecked for a false alarm. Here in the code, Server is the Serial pins defined by SoftwareSerial which will send the data to NodeMCU.

```
void loop() {
// calling different functions for different sensors to take reading
  getDHT();
  getMQ2();
  getMQ4();
  getMQ135();
  getflame();

//Storing all the sensor data to a string for sending via serial to NodeMCU
data=(String)alarm+','+(String) temperature+','+ (String)lpg+','+ (String)co+','+
(String)smoke+','+ (String)mq4+','+(String)mq135+','+(String)flame+','+Location;

  Serial.println(data); //serial print to bug fix
  server.println(data); //sending string to NodeMCU
  delay(500);
}
```

2.4.3 Wi-Fi Connectivity Mechanism

Connecting to the local Wi-Fi of the user and sending the data received by Arduino is done by NodeMCU. NodeMCU connects to the Wi-Fi, then it connects itself to the server via MQTT protocol. It publishes data to the server MQTT client with a unique topic that later determines where the data should be sent for further process.

NodeMCU receives data from the Arduino Nano as a Character buffer via serial communication. It sends that data to Server via Arduino Pub-Sub client to the server.

2.4.4 NodeMCU Code

To access the internet with Wi-Fi, we need to use a library in NodeMCU named ESP8266WiFi. As we will be sending the data via MQTT protocol, we will use PubSubClient. And SoftwareSerial for serial communication with Arduino.

```
#include <ESP8266WiFi.h>   //library for wifi
#include <PubSubClient.h>  //library for MQTT
#include <SoftwareSerial.h> //library for Serial communication
```

}

For PubSubClient, we need to define the server address to the target MQTT server, the port, username, password for the MQTT broker, and a specific topic name so it can be identified that the data is from a home device.

```
//Setting home wifi credential to connect
const char* ssid = "test";
const char* password = "test";
// setting MQTT broker credential and server formation to connect to server
const char* mqttServer = "139.180.140.52";
const int mqttPort = 1883;
const char* mqttUser = "ehet";
const char* mqttPassword = "0000";
const char* topic_name = "home_kit";
```

In the setup, we will make a loop to connect to the Wi-Fi, then to the mqtt server.

```
void setup() {
  Serial.begin(115200);
  mySerial.begin(115200);
  WiFi.begin(ssid, password);    // connect to wifi
// checking and re connecting to wifi if not connected.
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.println("Connecting to WiFi..");
  }
  Serial.println("Connected to the WiFi network");
//connection to the server via MQTT broker
  server.setServer(mqttServer, mqttPort);
  while (!server.connected()) {
    Serial.println("Connecting to MQTT...");
    if (server.connect(topic_name,mqttUser,mqttPassword)) {
      Serial.println(" connected ");
    } else {
      Serial.print(" failed with state ");
      Serial.print(server.state());
      delay(2000);
    }
  }
}
```

After successful connection to the server, we will receive the data coming from Arduino and publish it to the server.

```
void loop() {
  server.loop();
  if (mySerial.available()) {
    data = mySerial.readStringUntil('\n');  //receiving data from Arduino
  }
  Serial.println(data);   // serial print for debugging
//sending data to the server via publishing the to the server
  str_length = data.length()+1;
  char send_data[str_length];
  data.toCharArray(send_data,str_length);
  Serial.println(send_data);
  server.publish(topic_name,send_data);
  data="";   // resetting data string after sending to the server
  delay(1000);
}
```

2.5 Result and Discussion

In result and discussion, we will observe the sensor data characteristics and try to determine the possible response of the corresponding sensor in an unlikely or possible emergency scenario, for example, gas leakage or fire in the house or kitchen. We will also try to determine the **safe value, low-risk value, and high-risk value** of the particular sensor.

2.5.1 Sensor Data Receive

The data collected from the Home device have several parameters. All of them are not used in our final calculations. For the Home device, we focused on just Three parameters: temperature, Flame, LPG value, and Smoke data to analyze.

Test Case for Home Device: To test the response of the sensors of Home Module**,** we conduct a controlled test in a home kitchen, where initially, we keep the device approximately in 3 ft distance from the stove. Here are the test environment details:

Date	Time (GMT+6)	Kitchen size	Temperature*
5th April, 2020	20:25 (08:25 pm)	5’ x 6’	26 °C

**Temperature data are taken from the "AccuWeather, Inc." website.*

Data were taken for 6:00 minutes in total. The data is taken in 4 stages.

Firstly, we recorded the data when normal cooking is going on when the device was approximate 3 ft away from the stove. Secondly, we slowly increased then decreased the LPG value of open-air by opening the LPG valve 1/5th of the maximum pressure. While doing so, we kept the device around 10inch away from the cylinder valve as making LPG concentration high while keeping the sensor far from source will cause much risk to the house. Thirdly we have increased the smoke in the kitchen by putting some vegetables in very hot oil, and we have added some burning paper smoke to add with the cooking smoke to make the smoke intensity much high on air. This time as well, we keep the sensor almost 10 inches away from the source for the same reasons. And finally, we placed the device 2 feet away from the stove's burner, both directly in front of it, and indirectly. By indirectly, we are representing the flame source in an angle and slightly far away from the device. We then took the response data of flame.

Here given data sample for every 12 seconds. Starting from the time at 20:25:01 (HH:MM:SS).

Time (HH:MM:SS)	Temperature (°C)	LPG (ppm)	Smoke (ppm)	Flame (ADC value)
20:25:01	31.2	0	0	1023
20:25:13	31.3	0	0	1023
20:25:25	31.5	0	2	1023
20:25:37	31.8	0	0	1023
20:25:49	31.8	0	0	1023
20:26:01	31.8	0	0	1023
20:26:13	32.1	547	2	1023
20:26:25	32.3	801	2	1023
20:26:37	32.3	954	2	1023
20:26:49	32.3	2472	14	1023
20:27:01	32.3	1493	6	1023

20:27:13	32.6	0	0	1023
20:27:25	33.8	1	4	1023
20:27:37	34.8	2	78	1023
20:27:49	36.4	1	498	1023
20:28:01	36.4	1	693	1023
20:28:13	37.4	2	962	1023
20:28:25	37.8	2	1038	1023
20:28:37	35.9	2	886	1023
20:28:49	35.9	1	456	1023
20:29:01	34.9	0	180	1023
20:29:13	34.2	0	12	1023
20:29:25	33.9	0	4	1023
20:29:37	34.3	0	0	830
20:29:49	34.4	0	0	498
20:30:01	34.4	0	0	503
20:30:13	34.3	0	0	1023
20:30:25	34.3	0	0	0
20:30:37	34.3	0	2	167
20:30:49	33.9	0	0	284
20:31:01	33.9	0	0	1003

Table 1: Test Data for Home Device

Here, the first section of data shows the ambience data of the kitchen, which is in the first one minute (20:25:01 to 20:26:01) of data. There was no excessive smoke or LPG value with no detectable flame outside of the stove, and although the stove was on, that does not usually get detected by the flame sensor without taking the sensor directly near the flame. Here, sensor values are fairly normal, and smoke LPG and flame sensor are in their initial values.

Secondly, we did the LPG concentration test from 20:26:02 to 20:27:13. We slowly increase and then decrease the LPG value of open-air by opening the LPG valve to 1/5th of the maximum pressure and taking the sensor closer to the opening of the valve to simulate the increasing LPG concentration in the air. And after that, we pull back the sensor and close the valve to reduce the LPG value back to the initial value. Here it shows the LPG value increases and smoke value gives some random data between 1-15, which is fairly insignificant in the smoke value perspective.

Thirdly, after testing the LPG concentration, we test the smoke concentration in the air from 20:27:25 to 20:29:25. We have increased the smoke in the kitchen by putting some vegetables in very hot oil. It makes an excessive amount of smoke for some time. After one minute, we added some burning paper smoke to add with the cooking smoke, which eventually leads to more than 1000 ppm while putting the sensor very closed to the smoke. While putting it on the smoke, the temperature slightly increases and then eventually decreases. It also appears that the LPG value gives some random values of 1, 2, which is insignificant concerning the response data.

Finally, we checked the flame sensor response. We keep the sensor directly and indirectly in front of an open flame. By putting it indirectly, we are representing the flame source in an angle and slightly far away from the device. From 20:29:37 to 20:30:10, the flame was indirectly in front of the flame which indicates high fire but not an excessive amount. Then from 20:30:20 to 20:30:50, when we keep the flame directly in front of the sensor, the situation corresponds to a high flame. By keeping the sensor in front of the flame, it decreases value from 1023 to almost 0 depending on the flame quantity.

2.5.2 Sensor Data Characteristics

Data taken on the home device here are data collected from the kitchen environment. In a living room or bedroom, these data will have a similar threshold. Seeing data in the table is not an easy way to determine the critical points of the data or sensor readings. Therefore, we will use graphical representation to understand the sensor responses to different scenarios more efficiently and try to determine **low-risk, minimum risk,** and **high-risk** range of the response.

LPG Data:

In our controlled test environment, we have got three levels of LPG reading from the sensor, which is represented more understandably in the graph below.

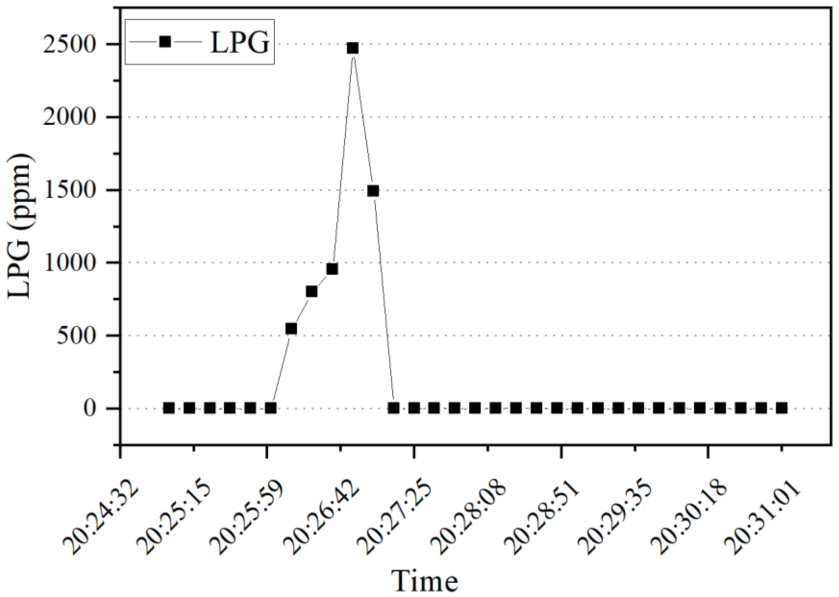


Figure 7: LPG Data graph of home device

We considered 20:25:59 to 20:26:42 as medium risk as that time the LPG level in the air was low, the sensors were much far away from the Gas cylinder. We can see that data ranges from 500 to 1000. After this time when we took the sensor much closer to the valve of the cylinder, the value picked to almost 2500, so we can consider 2000 as a high-risk level, as a medium level mostly gives a value of around 500 to 1000.

So, we can determine low risk, medium risk, and high-risk value range will be respectively, 0 to 499, 500 to1999, and 2000 to inf.

Smoke data:

As the cooking going on, smoke is very natural in some cases. This data can help us identify excessive smoke in case of fire.

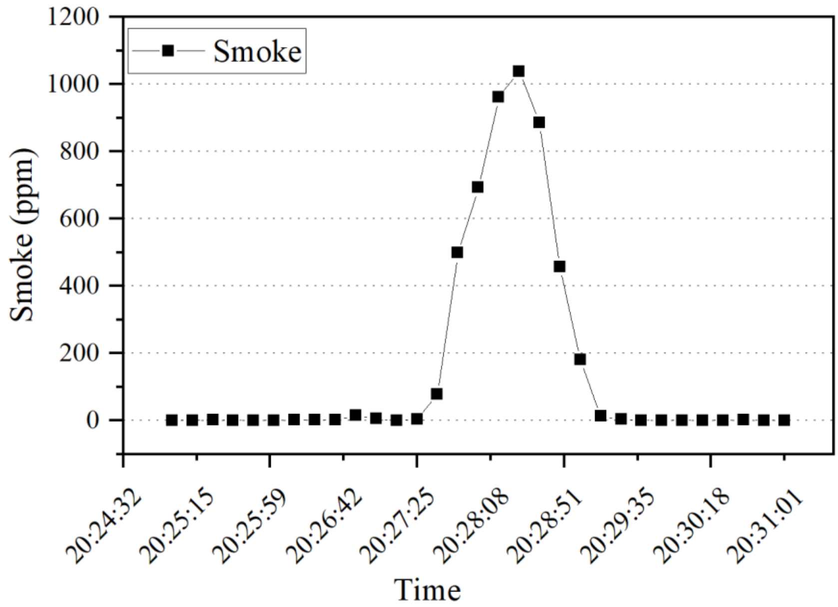


Figure 8: Smoke Data Graph of home device

The data showed here have maximum time zero value due to no smoke. After testing the LPG response when we switch to smoke firstly, we gave small smoke from oil in a pot and simulating high smoke cooking process. From 20:27:25 to 20:28:08 in this circumstance, the value picked to 500-700ppm. But when we added burning paper with that, to simulate the starting of an unnatural fire with smoke, the value quickly rises to 1000+.

So, High value surely will be 1000+ values. And medium risk can be set to 600, which is mean of 500 and 700. So below 600 ca be identifying as low risk or no risk value.

Flame Sensor Data:

In the graph we can see, the data ranges from 900 to 1023 whine in a no flame or fire and active flame in front, it gives 0-200 value.

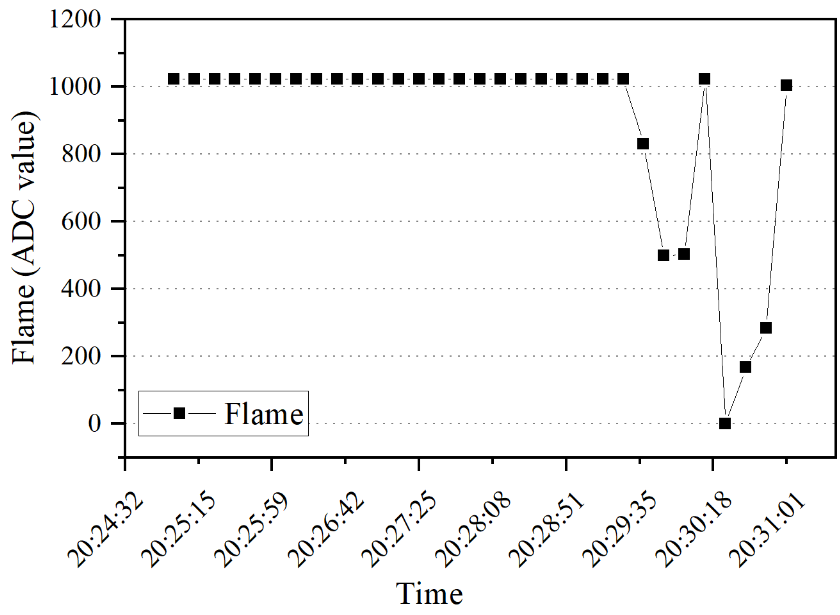


Figure 9: Flame Sensor response of the Home Device

This data changes with the amount of the flame in front of it. In the test, when we keep the flame indirectly in front of the sensor, we get a response of around 600+ in ADC value. So below that level, we can consider the low risk or no risk level. But when we take the flame directly after 20:30:00, we get almost 0 in ADC value, and slightly fluctuating to 300, so we can set 300 as higher risk value.

As the value is decreasing from a high value to low value on flame, our pre-determined low risk, medium risk, and high-risk value range will be respectively, 1023 to 601, 600 to301, and 300 to 0.

Temperature Sensor Data:

The temperature on the test day was 25°C-35°C as being a relatively hot day; the ambient temperature on the kitchen was around 31°C.

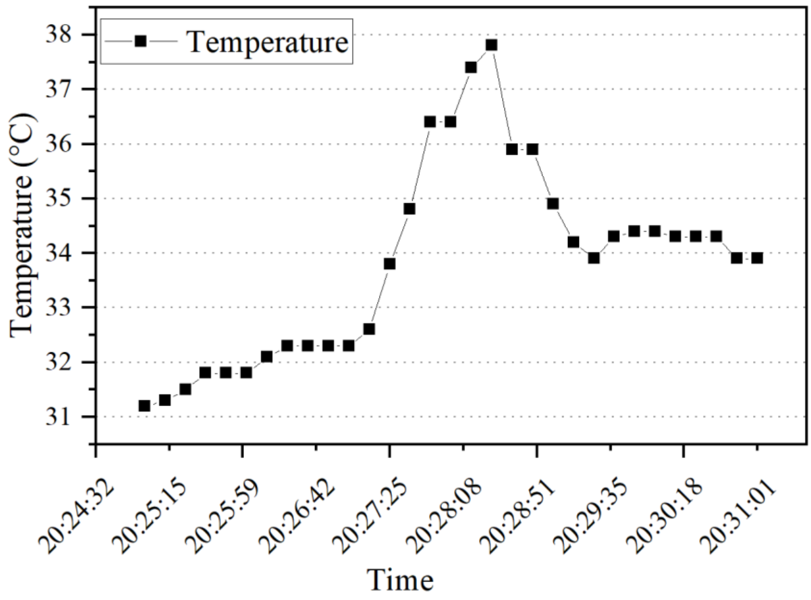


Figure 10: Temperature Data graph of Home Device

We can see a sudden temperature rise in between 20:27:25 and 20:29:35, which was due to the smoke test. When we started the smoke test, the temperature rises due to the sensors were too close to the stove, and hot smoke was coming out of it. After the smoke test, the temperature slowly came back to normal during the flame test. The flame test brings the temperature to rise a little bit, which is eventually insignificant. But with small smoke the temperature went up to more than 38°C, in normal working time, with much more to cook, even though the LPG or smoke will not increase, the temperature can be more than 45 °C. According to the DHT11 datasheet, it will give a better temperature value in the range of 0-50°C. So, in the event of a fire, though the actual temperature will be higher this sensor will give a maximum of 50°C - 55°C. so as without any option, we have to set both low and high threshold values to 49°C.

To summarize the response characteristics of the sensors, we can represent these by our determined low, medium, and high-risk levels.

Sensor	Risk Level	Value Range
LPG	Low	0 - 499
	Medium	500 - 1999
	High	2000 - inf
Smoke	Low	0 - 599
	Medium	600 - 999
	High	1000 - inf
Flame	Low	601-1023
	Medium	301 - 600
	High	0 - 300
Temperature	Low	0 - 48
	Medium	49 – inf
	High	49 - inf

Table 2: Home Device Threshold value chart

The ranges can be more dynamic, depending on the situation. This can be done by more dynamic test cases which were not possible due to lack of resources. These range values will be used in the server to notify mobile app and emergency responders depending on whether it is a medium risk situation or high-risk situation. Details about this will be described in Chapter 4.

2.6 Cost analysis

With the advancement in technology, there have been various inventions of devices that are used in home security. Different kinds of CCTV cameras, Smart locks, etc. along with in-built fire sprinklers in modern age homes are many of the new, advanced approaches while the good old fire extinguisher remains in most cases. Although there are several papers on this topic, there is yet to be a device available on the market that automatically notifies the fire station in case of a major accident. Therefore, there cannot be a fair comparison between our device and other fire safety devices on the market. The cost of the home safety device that is designed to notify nearby fire responders in case of a major fire accident where fire sprinklers are not enough is approximately BDT. 2000 only (USD 24). According to the prices collected from Bdstall, Fire alarms available in Dhaka, Bangladesh start at the lowest BDT. 650 (Fire Alarm Motor Siren) and can go up to BDT.3500 (Context Plus Conventional Photoelectric Smoke Detector). However, both these products function on the principle of creating a loud sound to notify the consumer of an accident, and it is up to the consumer to call for help. A regular 5kg fire extinguisher ranges from BDT. 1500 to BDT.3000 while a mini version costs BDT. 500. There are other safety measures than can be taken, such as escape masks that are priced at BDT. 750 and evacuation Mask 30 Minutes Smoke Protector priced at BDT. 2550. There are also slightly more advanced smoke detectors that monitor carbon monoxide levels ranging from USD 29 to USD 43 when ordering from Amazon, excluding shipping costs. Although these assist in notifying the user, it will still be the user's responsibility to analyze the situation and call for help.

To conclude the chapter, we like to point out the key takeaway of this chapter. We have discussed the build and coding with its working mechanism. We have seen and analyze the sensor data to determine the minimum and maximum threshold value to use in the server, which will be a comprehensive system in chapter 4.

Chapter 3

The IoT Based Vehicle Device

This chapter focuses on the Vehicle Device. It starts by discussing the literature review and then breaks down the design of the device in terms of the components used and the reasons behind it. This is followed by the hardware connection, which breaks down the connection diagram, schematic diagram, PCB layout followed by the implemented picture. Next comes the working mechanism which discusses the data sensing mechanism, GPRS connection mechanism, and Arduino code. Finally, results and discussion elaborate more on the sensor data and its characteristics and concludes by discussing the cost analysis.

3.1 Literature Review

There have been many measures taken to reduce the magnitude of impact during accidents, such as the invention and innovation of airbags, ABS (Anti-lock Braking System), traction control, and improved electronic stability control. Windshields are made using shatter-resistant glass that, in case of an accident breaks into multiple pieces that are comparatively less harmful than regular shards of glass. Safety measures as basic as bumpers and seat belts were also invented in the 1800s to reduce the impact of damage caused in accidents followed by airbags that were not invented much later during the 1900s. The anti-lock braking system was also improved from rear wheels only to all four wheels to prevent skidding and allow the driver to maintain control of the steering wheel while applying the brakes, especially on smooth surfaces. Furthermore, modern technologies such as pre-collision technology enable many modern vehicles to use sensors to prepare for an impending collision by tightening of seatbelts, alignment of airbags, and preloading of brakes to reduce shock.

However, there is yet to be a full system that caters to the aftereffect of an accident if and when one occurs. These measurements have been designed to be implemented on the vehicle itself to reduce its chances of incurring an accident and reducing the impact on the vehicle and passenger to the bare minimum level. However, there are very few comprehensive systems in the market that caters to situations where emergency response is needed after an accident has already taken place.

But there are quite a few published research papers on accident detection that resemble our idea of minimizing response time after an accident. For instance, Faiz et al. [23] have proposed the development of an Android-based application that detects an accidental situation and sends an emergency alert message to the nearest police station and health care centre. This application is integrated with an external pressure sensor to extract the outward force of the vehicle body. It measures the speed and change of tilt angle with GPS and accelerometer sensors respectively on an Android phone. By checking conditions, this application is also capable of reducing the rate of false alarm. In a similar paper, Ali and Alwan [24] concluded that most of the smartphone-based accident detection systems rely on the high speed of the vehicle (extracted from the smartphone GPS receiver) and the G-Force value (extracted from smartphone accelerometer sensor) to detect an accident. As many references assure that 90% of road traffic accidents occur at the low speed of the vehicle; therefore, in addition to high-speed accident detection, this paper concentrated on low-speed car accident detection. The main obstacle that encounters the low-speed accident is how to differentiate whether the user is inside the vehicle or outside the vehicle, walking or slowly running. The effect of this obstacle is minimized, in this work, by a proposed mechanism that distinguishes between the speed variation of the low-speed vehicle and walking or slowly running person. The proposed system consists of two phases; the detection phase, which is used to detect car accidents at low and high speeds. The notification phase, and immediately after an accident is indicated, is used to send detailed information such as images, video, accident location, etc. to the emergency responder for fast recovery.

There is also a GPS and Map matching based vehicle accident detection system where Amin et al.[25] proposed that whenever the speed is falling below the safe calculated threshold speed, the system will generate an accident situation. It will check the vehicle location from a map matching module and generate an accident situation if the vehicle is found outside the road network. This will reduce false accident detection drastically. The map matched accident location is then sent by utilizing the GSM network.
The main aspect among others that differentiates these papers from ours is the inclusion of IoT to build a more comprehensive system. Although there are papers where IoT was applied unlike our system, it used the combination of IoT and smartphones such as one proposed by Bhatti et al. [26], where the systems rely on the ubiquity of sensor-rich mobile phones for the detection of accidents. The proposed system uses multiple smartphone sensors, including accelerometer, GPS, pressure, and microphone acquisition to detect accidents. A smartphone application is developed that continuously reads data from the sensors and transmits this information to the cloud for further computation. An accident is detected through threshold analysis. Although the notification process of this system is much like that of ours, the identification process differs in terms of many important factors. Firstly, our system does not rely on a smartphone alone to receive and transmit data as that poses quite a few problems. Being dependent on a smartphone would require it to be operating on a minimum level of battery power, and if the phone runs out of charge, it will stop recording data. Secondly, it is difficult to localize the position of the smartphone; whether it is inside or outside the vehicle.

In a slightly different paper developed by Yee and Lau[27], a more manual approach was taken. Vehicle collision detection (VCD) system is an Android Application that requires the assistance of on-board sensors such as GPS and Accelerometer. GPS is used to calculate the car speed, while the accelerometer is used to calculate the acceleration force. Their idea is to combine both sensors' decisions to evaluate an accident condition, i.e. the severity of potential accidents. Connectivity is required, as the database is located at the backend using back endless API, for data creation/updates. User can be a service provider or driver; providers will receive accidents sent by driver and driver has to provide information surrounding the trip. When a potentially dangerous event data occurs, it will prompt the user to confirm whether an accident has occurred. Users can cancel if no accident has happened, whereas if there is no response from the user for 20 seconds, the application will search for the nearest providers within a 10km radius. If no providers are found within a 10km radius, a message will be sent to 911. If a Provider is found within a 10km radius, details of the accidents, including the number of passengers, current victim, and location will be stored in a database and assigned to the provider. Lastly, a notification will be delivered to the assigned provider. This is the crucial difference that sets our system apart from theirs. In our case, the vehicle kit does all the work of collecting the data via not just two, but multiple sensors and sending the data to service providers when appropriate after analyzing threshold values. Instead of manually putting in information manually

Compared to those, in our system is we are using multiple sensors to measure various parameters. In addition to gas sensors and flame sensors commonly used in other systems, the piezo element is incorporated into the system. The piezo element collects native collision sensor data which helps in determining the severity of the accident. The GPS module is integrated into the kit, so additional GPS tracker is not necessary. Therefore, multiple functions can be performed by this vehicle kit.

3.2 Vehicle Device Design

The purpose of the Vehicle device is to send the Live GPS Location to the server along with the Gas ppm values, temperature, and flame detection data. So, a possible accident can be detected, and the necessary steps can be taken to rescue the passenger. On top of that, the excessive temperature of gas emission can be detected and used to notify the user to conduct maintenance of the vehicle to avoid a possible accident.

3.2.1 Component Selection

The components for the Vehicle device consist of a microcontroller unit, three gas sensors, one temperature sensor, one flame sensor, GPS and GPRS module taking data, and sending to the server.

- Arduino Mega
- MQ-2
- MQ-4
- MQ-135
- DHT11
- IR based Digital Flame sensor
- Piezo Element
- MPU6050
- Quectel L80-R GPS Module
- SIM800L GPRS Module

3.2.2 Component Description

The components used here all project grade components to minimize the system development cost. Here are the components:

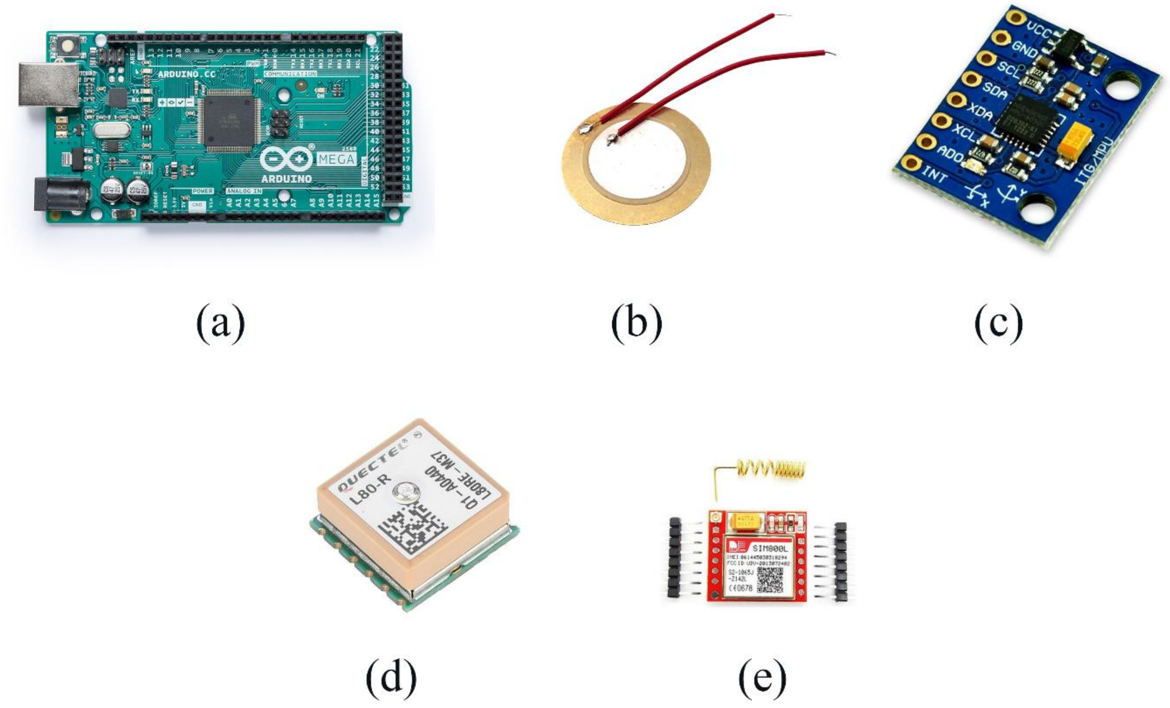


Figure 11: Vehicle Device Components. (a) Arduino Mega, (b) Piezo Element, (c) MPU6050, (d) Quectel L80-R GPS Module (e) SIM800L GPRS Module

Arduino Mega

Arduino Mega is a type of microcontroller of the Arduino series, which is based on the At mega 2560 chip. The memory space is much larger than the other boards of Arduino series containing a greater number of digital and analog pins. It has 54 digital I/O pins and 16 analog pins on the board. A crystal oscillator of 16MHz frequency is also present on the board. There are three ways to power this Arduino board. The most common one is through a USB cable port connected with the Computer, which is also used for uploading code to the Arduino board. The other two ways are DC power jack and through Vin pin of the board. A reset button and four hardware serial ports called USART is incorporated into the board for producing maximum speed for communication. There is one unique feature of this Arduino board that it has an ICSP header, which connects to the Computer to upload Codes and to debug.

Piezo Element

The piezo element is a sensor that detects collision by measuring pressure and vibration. It can be interfaced with microcontrollers. The electric signal will be generated as an output signal from the collision of the materials. A thin membrane is placed on a large base to transmit the force to the piezoelectric element. The piezoelectric element will start to generate electric voltage depending on the amount of pressure being applied to the thin membrane. The amount of electrical voltage is proportional to the amount of pressure applied. The suitable temperature in which it can work is -20 degrees Celsius to 60 degrees Celsius. Quartz is the most suitable material for the piezoelectric sensor because the flexibility of the material is very high.

MPU6050

MPU6050 or GY-521 is a sensor that allows measurement for both a 3-axis gyroscope and a 3-axis accelerometer consisting of a built-in Digital Motion Processor (DMP). It can communicate with microcontrollers like Arduino through I2C serial data bus using two wires called Serial Clock Line (SCL) and Serial Data Line (SDL). SCL is used for giving clock pulse, and SDL is used for transmitting data through I2C communication. According to the datasheet, the range of accelerometer is ±2, ±4, ±8, and ±16g. The range of gyroscope is ±250, 500, 1000, and 2000 degrees/s. The sensor consists of a 16-bit analog to digital converting for converting the output value to digital. It also has a temperature sensor embedded inside it.

Quectel L80-R GPS Module

L80 is an ultra-compact GPS POT (Fix on Best) module with an implanted 15.0mm × 15.0mm × 4.0mm fix radio wire as antenna. This space-saving plan makes L80 the culminate module for the scaled-down gadgets. Received by the LCC bundle and coordinates with fix receiving wire, L80 has extraordinary execution both in acquisition and tracking. Combining progressed AGPS called EASY™ (Embedded system) and demonstrated AlwaysLocate™ innovation, L80 accomplishes the most elevated execution and ultimately meets the mechanical standard. EASY™ innovation guarantees L80 can calculate and automatically predict orbits utilizing the ephemeris information (up to 3 days) put away in inside Smash memory so that L80 can settle position rapidly indeed at indoor flag levels with moo control utilization. With AlwaysLocate™ innovation, L80 can adaptively alter the on/off time to adjust between situating exactness and control utilization agreeing to the natural and movement conditions.

L80 is programmed in a way that supports automatic switching between antennas. It can accomplish the switching between the internal fixed antenna and the external dynamic antenna. Additionally, it keeps situating amid the exchanging process. The Fitness Low Power (FLP) highlight gives low power GPS arrangement for wellness applications. It is an optimized arrangement for wearable, fitness and tracking gadgets, and the power cost is almost 50% control utilization of normal mode. With its compact plan, high accuracy, and affectability, L80 is flawlessly appropriate for a broad run of M2M applications such as convenient gadgets, car, personal following, security, and mechanical PDA, particularly suited for different applications, like GPS mouse and OBD.

SIM800L

SIM800L module is a miniature GSM/GPRS modem. Its features make it suitable to integrate into many IoT projects. It can be used to carry out tasks that are quite similar to a regular cellphone in terms of essential communication methods. The SIM800L module supports quad-band GSM/GPRS network, which allows us to send and receive SMS and voice calls and connect to the internet through GPRS, TCP/IP, and more. This component communicates with the microcontroller through UART port, supports command including 3GPP TS 27.007, 27.005, and SIMCOM enhanced AT Commands. When it is supplied with power, it starts up, searches for the cellular network, and logs in automatically. It is highly affordable and has a super convenient size. On top of it, the quad-band frequency ensures long-range connectivity, which means it can work anywhere in the world in the projects that require the features.

MQ-2, MQ4, MQ135, DHT11, IR based Flame sensor

These components are described in Chapter 2.2.2 in corresponding

MQ-2,

MQ-4, **MQ-135**, **DHT11**, **Flame sensor** Sections.

3.3 Hardware Connection

All the sensors are connected to Arduino mega along with the GPRS module to send data to the server. The Microcontroller used the sensors as input and GPRS modules as output.

3.3.1 Connection Diagram

Here is shown the Connection Block Diagram.

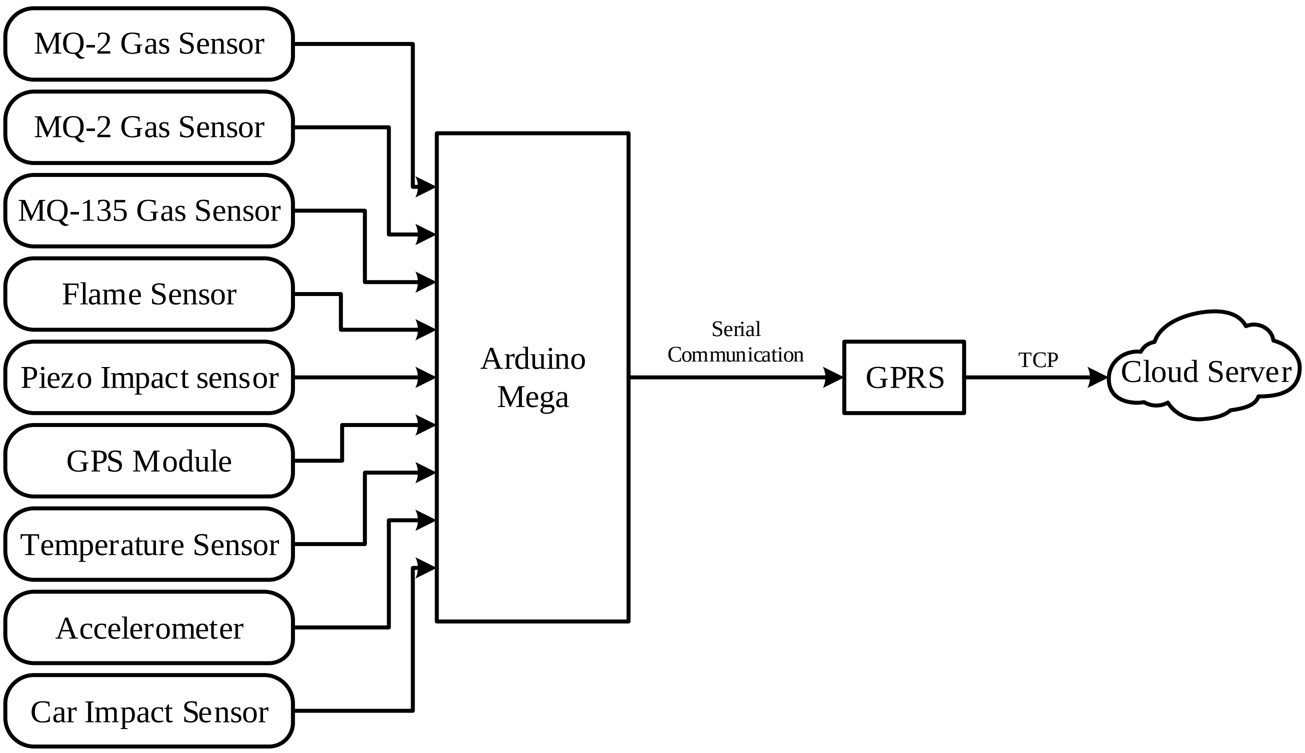


Figure 12: Vehicle Device Connection Block Diagram

3.3.2 Schematic Diagram

Here is shown the Schematic of the Vehicle device. It shows the pin by pin Component connection of the device.

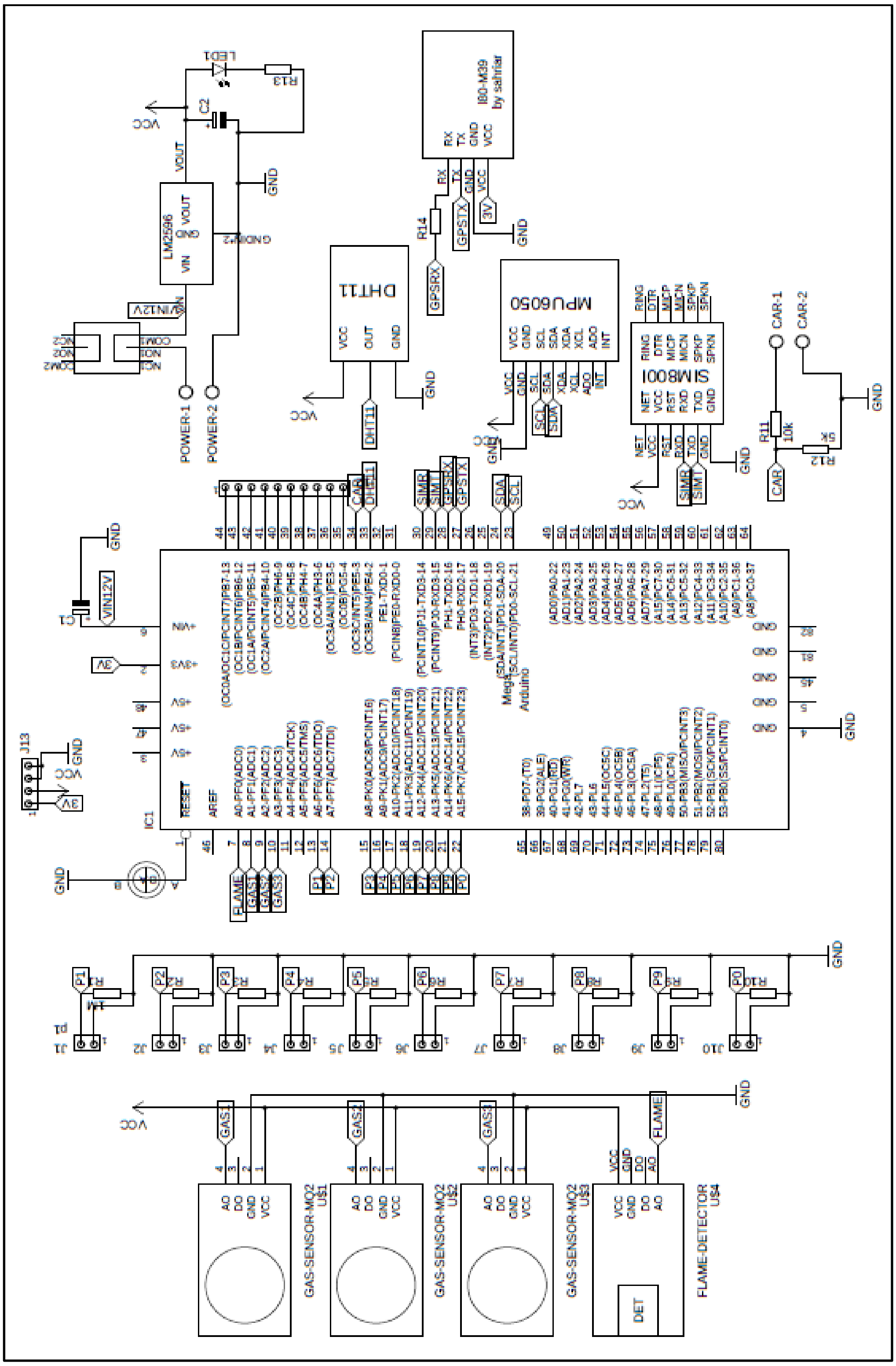


Figure 13: Vehicle Device Schematic Diagram

3.3.3 PCB Layout

Here is shown the PCB Layout and component placement of the vehicle device. This is built based on the schematic diagram shown previously. This PCB board is used to build our vehicle Device prototype.

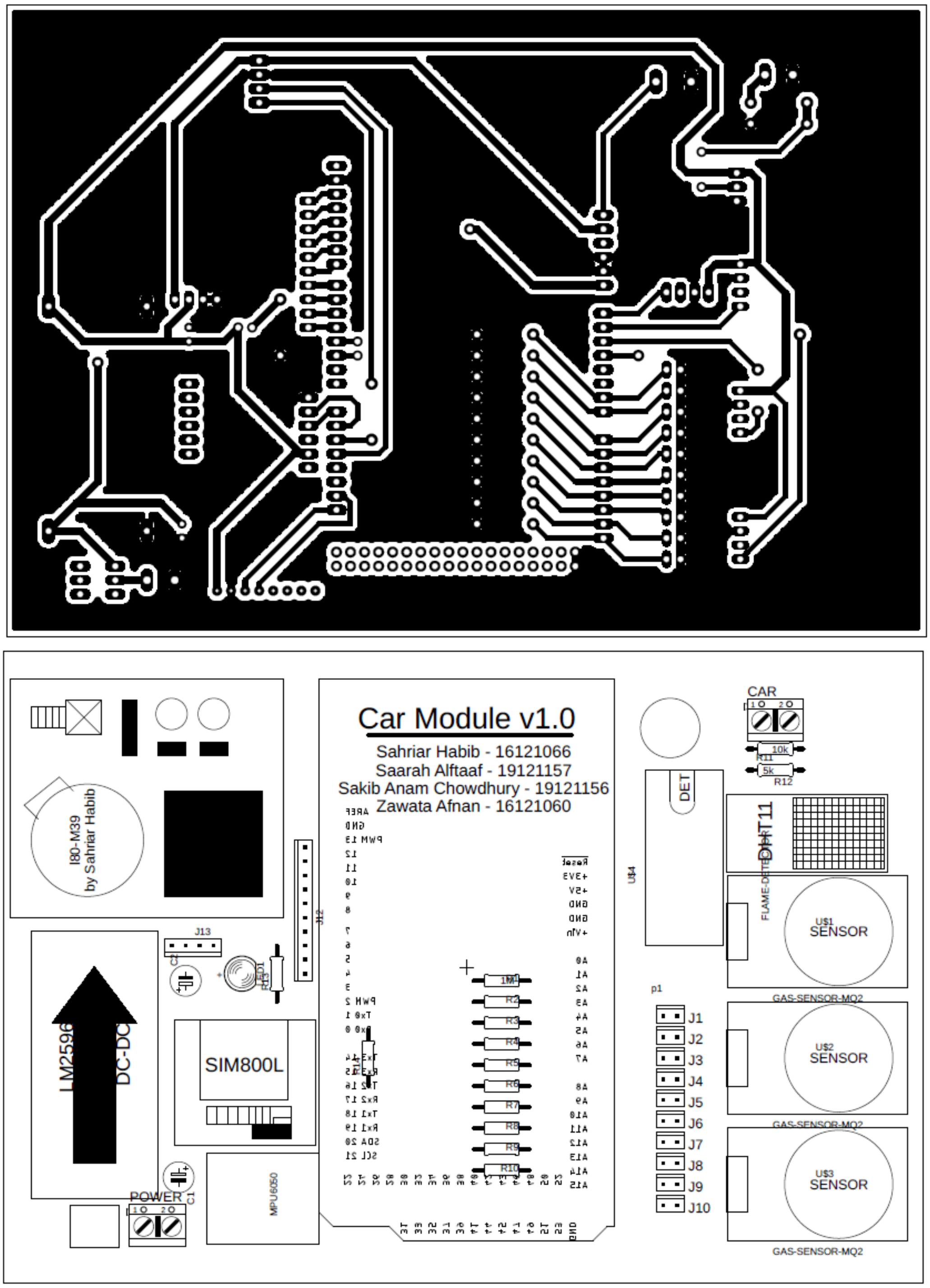


Figure 14: PCB Layout of Vehicle Device

3.3.4 Implemented Picture

Here shows the Prototype of the vehicle device build from previously shown schematic and PCB. This prototype is used for all vehicle device data collection.

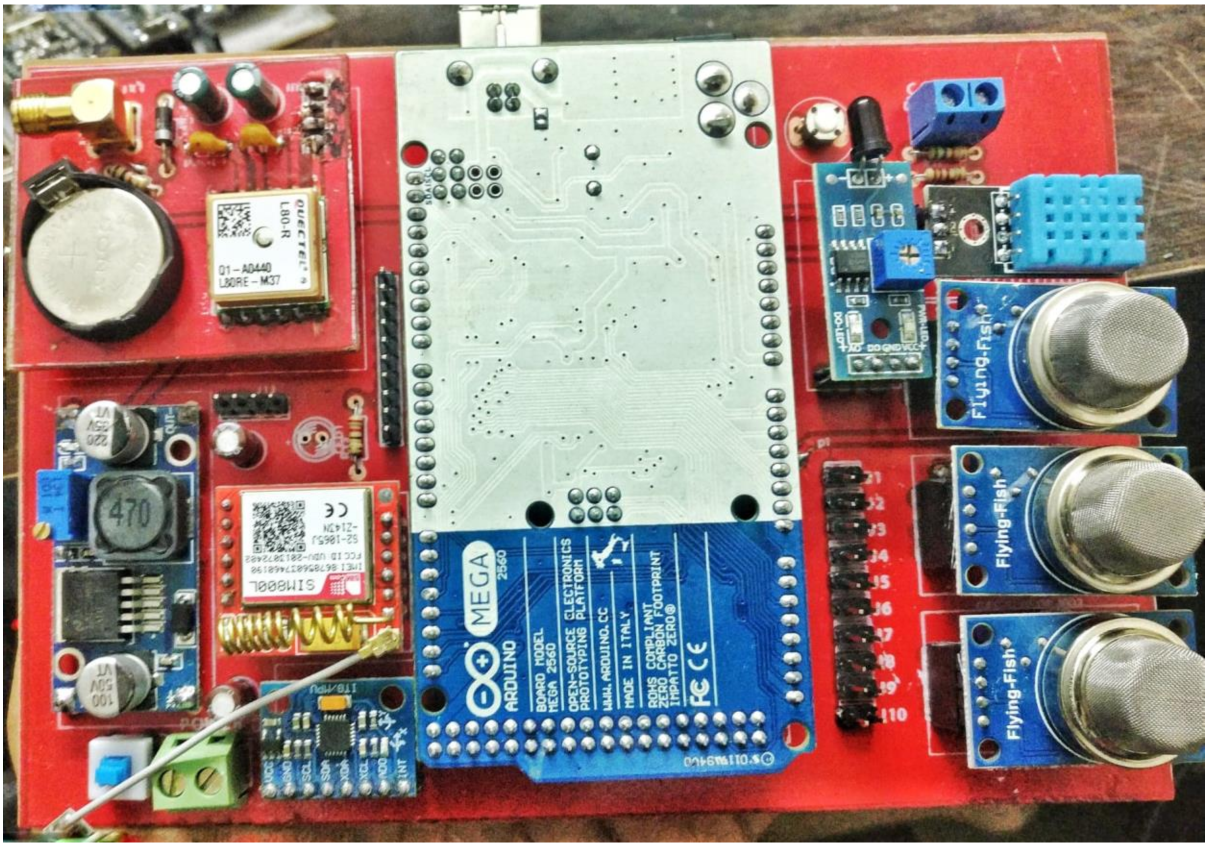


Figure 15: Prototype of Vehicle Device

3.4 Working Mechanism

In this section, we will discuss the working mechanism of the Vehicle device until the data is sent to the server. Before sending the data, we will collect the data and check some basic conditions and send this to the server.

3.4.1 Data Sensing Mechanism

Data Sensing happens in 2 steps,

- Raw data sensing
- Conditioning

First of all, Arduino read the MQ2, MQ4, MQ135, DHT11, Flame sensor, MPU6050, Piezo, and GPS data. Those data reads happen in several ways. For gas sensors data, we used ADC reading by MCU, then the corresponding Arduino library process the ADC value with the resistance value of the sensor to get the ppm values and those stores in a variable. Then the Corresponding DHT11 library decodes the temperature values sent from the DHT11 temperature sensor. DHT11 sends a digital signal to the MCU which needs to be decoded. The flame sensor does not need any library. It changes its analog signal value with the flame IR reading. It can give different values to different levels of flame detected. The piezo gives Analog Data multiple piezo is connected through the ADC pins of the Arduino.

MPU6050: gives acceleration data which needs some numerical calculation to use as a meter per second values. Raw acceleration data is collected from reading the register values of the sensor. According to MPU6050 datasheet and Documentation, those data need to divide by 16384.0 to get the usable numerical value.

GPS Module: GPS modules gives NMEA data. NMEA data is a data format developed by The National Marine Electronics Association, USA. The raw NMEA data looks like below:

```
$GPRMC,090917.00,A,2352.00512,N,09024.28788,E,0.370,,131119,,,A*7A

$GPVTG,,T,,M,0.370,N,0.685,K,A*2C

$GPGGA,090917.00,2352.00512,N,09024.28788,E,1,04,4.40,14.2,M,-49.3,M,,*48

$GPGSA,A,3,31,32,10,14,,,,,,,,,11.83,4.40,10.98*0E

$GPGSV,2,1,05,10,48,129,39,14,50,351,32,18,,,22,31,77,302,44*4E

$GPGSV,2,2,05,32,45,036,31*4B

$GPGLL,2352.00512,N,09024.28788,E,090917.00,A,A*6D
```

The data is passed throw the library "TinyGPS" to get the Latitude and longitude value by parsing the strings from "$GPRMC" shown above. According to the "GPS Information" website, we found our module fives NMEA v2.0 types of data.

Secondly, the flame data, Impact data, and vehicle's airbag deployment data is checked with a previously stored sensor value to check if there is any fire, that turned a flag ON so that further processes can be done by emergency responder immediately.

Finally, all the sensor data, along with the flag to determine emergencies, GPS location of the device, and a unique device ID is cascaded into a string and send that to the server via GPRS Module using TCP protocol for further processing.

3.4.2 GPRS Connection Mechanism

GPRS Module we use is SIM800L. To communicate via GSM or GPRS, we need to read and write AT Commands. According to the article in the "ElectronicForU" website, AT Commands are commands to control modems where AT stands for Attention. There are some well-established commands for a certain task to answer from the GPRS module. For example, Writing "AT" to the module, it will reply with "OK" which signifies a successful connection with the proper power supply to the module. Here we used some commands to check the sim connection like, “AT+CCID” to check SIM is connected or not, if connected, it will give a unique SIM ID. For our current connection which a TCP connection to a remote server, we will use “AT+CIPSTART” to connect to the server with server IP and listening port. For example, we used the command below, where ‘139.180.140.52” is the server IP address and “1850” is the port.

```
"AT+CIPSTART=\"TCP\",\"139.180.140.52\",\"1850\""
```

While sending a certain String to the server, we use the "AT+CIPSEND" command. Sending this command will make the module ready to receive sending data. After that, we can simply send the data to the module, and the module will automatically send the data to the previously specified server on the specified port.

3.4.3 Arduino Code

We used DHT, mq2, mq135, wire, and TinyGPS to take sensor values. And SoftwareSerial to send data to server via GPRS module. Using TinyGPS, we can simplify code for string parsing from GPS NMEA data.

```
#include <SoftwareSerial.h>  // library for serial communication
#include "DHT.h"  // library for temperature sensor
#include <MQ2.h>  // library for gas sensor
#include<Wire.h>  //I2C library for communication.
#include <TinyGPS.h> // GPS library
```

After defining all the pins and variables needed for the code, we define different functions to collect different sensor data.

```
. void getDHT() {
  temperature = dht.readTemperature();  //DHT temperature
}
void getMQ2(){
  lpg = mq2.readLPG(); //lpg value reading
  co = mq2.readCO(); // CO value reading
  smoke = mq2.readSmoke(); // smoke value reading
```

```
}
void getMQ4(){
 mq4 = analogRead(mq4_pin); //mq4 data read
}
void getMQ135(){
 mq135 = analogRead(mq135_pin); //mq135 data read
}
void getflame(){
 flame = analogRead(flame_pin); // flame data read
}
void accelerometerSetup(){   //setting up accelerometerwith I2C address
 Wire.begin();
 Wire.beginTransmission(MPU);
 Wire.write(0x6B);
 Wire.write(0);
 Wire.endTransmission(true);
}
void get_accelerometer(){ //get data of accelerometer
 ax = 0.00;
 ay = 0.00;
 az = 0.00;
 for (int i = 0; i < 10; i++) {       //to take average of 10 data
   Wire.beginTransmission(MPU);
   Wire.write(0x3B);
   Wire.endTransmission(false);
   Wire.requestFrom(MPU, 12, true);
   AcX = Wire.read() << 8 | Wire.read();
   AcY = Wire.read() << 8 | Wire.read();
   AcZ = Wire.read() << 8 | Wire.read();
   GyX = Wire.read() << 8 | Wire.read();
   GyY = Wire.read() << 8 | Wire.read();
   GyZ = Wire.read() << 8 | Wire.read();
   ax = ax + (AcX / 16384.0);
   ay = ay + (AcY / 16384.0);
   az = az + (AcZ / 16384.0);
 }
 ax = ax / 10;
 ay = ay / 10;
 az = az / 10;
 Wire.endTransmission(true);
}
void get_gps(){
 while (gpsSerial.available()) { // check for gps data
```

```
    if (gps.encode(gpsSerial.read())){      // encode gps data
      gps.f_get_position(&lat, &lon); // get latitude and longitude
    }
  }
  latitude = String(lat, 6);
  longitude = String(lon, 6);
}
void get_piezo(){       /piexo data
  p1 = analogRead(piezo1);
  p2 = analogRead(piezo2);
  …
  p10 = analogRead(piezo10);
}
void get_car(){
  car = digitalRead(car_in);
}
```

We also make separate functions to define GPRS module. A set of AT command is needed to do so. Here, we connect to the internet and also connect to the server and port to data send.

```
void sim_init(){      // sim set to our specify mode
  Serial.println("Initializing...");
  delay(1000);
  Serial3.println("AT");  // check if module connected
  updateSerial();
  Serial3.println("AT+CSQ"); //Signal quality test, value range is 0-31 , 31 is the best
  updateSerial();
  Serial3.println("AT+CCID"); //Read SIM information to check the SIM is plugged
  updateSerial();
  Serial3.println("AT+CREG?"); //Check whether it has registered in the network
  updateSerial();
  Serial.println("Done Initializing.");
  Serial3.println("AT+CSTT=\"blweb\""); // setting the apn
  updateSerial();
  delay(1000);
  Serial3.println("AT+CIICR"); // printing the netword unique id
  updateSerial();
  delay(1000);
  Serial3.println("AT+CIFSR");  // printing the IP address of the device
  updateSerial();
  delay(1000);
  Serial3.println("AT+CIPSTART=\"TCP\",\"139.180.140.52\",\"1850\"")//MQTT connect
  updateSerial();
```

```
  delay(1000);
}
```

In the main loop we take all the data , merge those to a string and send to the server.

```
void loop(){
//calling different functions to take reading of different sensors
  getDHT();
  getMQ2();
  getMQ4();
  getMQ135();
  getflame();
  get_accelerometer();
  get_gps();
  get_car();
//some in device parameter check
  if(car==1){
    alarm =1;
  }else if(flame <200){
    alarm =1;
  }else if(ax<-10 && (p1>800 || p2>800 || p3>800 || p4>800 || p5>800 || p6>800 || p7>800 ||
p8>800 || p9>800 || p10>800) && smoke >5000){
    alarm =1;
  }else{
    alarm =0;
  }
// all sensor data to a string to send
  data = (String)alarm+','+(String)temperature + ',' + (String)lpg + ',' + (String)co + ',' +
(String)smoke + ',' + (String)mq4 + ',' + (String)mq135 + ',' + (String)flame + ',' + (String)ax
+ ',' + (String)ay + ',' + (String)az + ',' +latitude+ ',' +longitude+ ',' +(String)car+ ','
+(String)p1+ ',' +(String)p2+ ',' +(String)p3+ ',' +(String)p4+ ',' +(String)p5;
  Serial.println(data);
  send_data(data);  //data send via GPRS function
}
 void send_data(String dat)
{
  Serial3.println("AT+CIPSEND");  //taking the module to send mode
  delay(300);
  Serial3.print(dat);  // sending the data
  Serial3.println((char)26);  //signaling the module that data sending is done
  delay(500);
}
```

3.5 Result and Discussion

In result and discussion, we will observe the sensor data characteristics and try to determine the possible response of the corresponding sensors in an unlikely or possible emergency scenario, for example, an Engine fire or a collision with another object which could be a possible accident. We will also try to determine the **safe value, low-risk value,** and **high-risk value** of the particular sensors.

3.5.1 Sensor Data Receive

The data collected from the Vehicle Device have several parameters. All of them are not used in our final calculations. For Vehicle device, we focused on mainly five parameters to analyze in detail. These are **Temperature, Forward Acceleration or deceleration, CNG, Flame and Smoke data.** Other data need less analysis to determine the response, such as Piezo sensor for impact detection and vehicle's native collision detection trigger.

Test Case for Vehicle Device: To test the response of the sensors of Vehicle Device**,** we conduct a controlled test, where the device was placed under the car bonnet at the side of the car engine. Here are the test environment details:

Date	Time (GMT+6)	Vehicle Model	Engine and Fuel Type	Ambience Temperature*
5th April 2020	18:00 (06:00 pm)	Toyota NOAH, Model 1998	CNG Converted Gasoline Engine	24 °C

**Temperature data are taken from the "AccuWeather, Inc." website.*

Data were taken for 2:00 Hours in total. We will discuss specific sensor response to a specific task done by the vehicle rather than discussing the whole data. The data is taken in 5 stages.

Firstly, we keep the module under the bonnet in its place for a few minutes to take the ambience data under the car bonnet. Secondly, after that, we started the car engine, and without driving it for some time, we took the normal data of a started engine. Thirdly, we tested the CNG leakage by slightly releasing the CNG valve of the engine to test the potential gas leakage. Fourthly, we tested the smoke sensor response by putting artificial smoke under the bonnet. And finally, we drove the car to a safe area to test the acceleration and its related smoke release from the engine.

For stage one and two data collection, we took sample data in 20 seconds interval to represent the response of the sensors. Here the data shown below in Table 3 are the first and second stages of our test.

Time (HH:MM:SS)	Temperature (°C)	CNG (ppm)	Smoke (ppm)	Acceleration (ms^{-2})
17:58:01	30.7	1	0	0.294
17:58:21	31.1	1	0	0.098
17:58:41	31.2	4	0	0.098
17:59:01	31.2	2	0	0.098
17:59:21	31.3	1	0	0.098
17:59:41	31.4	2	0	0.098
18:00:01	31.4	1	0	0.098
18:00:21	31.5	1	0	0.098
18:00:41	38.4	5	19	0.098
18:01:01	42.3	7	0	0.098
18:01:21	45.8	8	0	0.098
18:01:41	49.1	8	0	0.098
18:02:01	53.2	13	0	0.098
18:02:21	54.6	7	0	0.098
18:02:41	54.6	30	0	0.098
18:03:01	53.4	51	0	0.098
18:03:21	53.7	39	0	0.098
18:03:41	53.7	62	0	0.098

Table 3: Vehicle Device Data Set for Stage one and two of Test case

Firstly, the data shows the ambience data while the engine is turned off from time 17:58:01 to 18:00:01 where all the data are in their initial value. Secondly, after starting the engine, the temperature and CNG data starts rising. Here from 18:00:21 to 18:03:41, the CNG data shown are very low, which is from burning CNG in the engine. This does not show any leakage of excessive gas.

For stage two and three, we took sample data in 10 seconds interval to represent the data set.

Time (HH:MM:SS)	CNG (ppm)	Smoke (ppm)	Acceleration (ms^{-2})
18:09:00	0	0	0.098
18:09:10	114	0	0.098
18:09:20	290	4	0.098
18:09:30	589	7	0.098
18:09:40	741	1	-0.218
18:09:50	13	1	0.098
18:10:00	13	0	0.098
18:10:10	0	16	0.098
18:10:20	1	162	0.098
18:10:30	12	387	0.098
18:10:40	4	588	0.098
18:10:50	16	763	0.098
18:11:00	5	265	0.098
18:11:10	0	1	0.098

Table 4: Vehicle Device Data set for Stage three and four of Test case

In the third stage, we make an artificial gas leakage from the engine and tested the response of the gas sensor. The data of this stage is shown in table 4 from 18:09:00 to18:10:00. Here in the first 40 seconds, we keep the bonnet closed and let the gas concentration rise, then we open the bonnet and let all the gas release from the chamber.

In the fourth stage, we make artificial black smoke using plastic and paper. As creating black smoke in an engine is very risky, we take a large piece of paper with some plastic and burn them to create a high-density smoke and put those under the bonnet right after stopping the fire and get an approximate smoke value. The data in table 4 from 18:10:10 to 18:11:10 is the smoke sensor response data.

To represent the fifth stage where we get the acceleration sensor response, we took data in 25 seconds interval to represent the data set.

Time (HH:MM:SS)	CNG (ppm)	Smoke (ppm)	Acceleration (ms^{-2})

18:41:09	177	0	0.981
18:41:34	4	0	2.5506
18:41:59	63	0	-0.0981
18:42:24	83	0	0.981
18:42:49	83	0	-0.981
18:43:14	65	0	1.8639
18:43:39	78	0	1.4715
18:44:04	5	0	1.0791
18:44:29	2	4	-14.715
18:44:54	9	3	-1.6677
18:45:19	18	0	0.3924
18:45:44	23	0	2.4525
18:46:09	44	0	-0.981
18:46:34	49	0	-0.1962
18:46:59	43	0	-0.981
18:47:24	32	0	1.5696
18:47:49	11	0	0.4905
18:48:14	12	0	1.3734
18:48:39	12	0	1.2753
18:49:04	26	0	1.3734

Table 5: Vehicle Device Data set for Stage five of Test case

The final stage data is shown in table 5, where we take a car to a remote place where there are few vehicles on the road to test the sensor response for hard braking which eventually represent the hard break for safety reason or hard break due to collision. Data taken from 18:41:09 to 18:49:04 represent the driving pattern of a typical driver better; that is why we took this portion of data to analyze further.

3.5.2 Sensor Data Characteristics

The data shown here is data from a controlled environment. By seeing data in the table is not an easy method to determine the critical points of the data or sensor readings. Therefore, we will use graphical representation to understand the sensor response to a different scenario more efficiently and try to determine **low-risk minimum risk and high-risk range** of the response.

Temperature data:

Here, the temperature is fairly static when the car is stopped. When the car engine is started, the temperature rises gradually and reaches its saturation point.

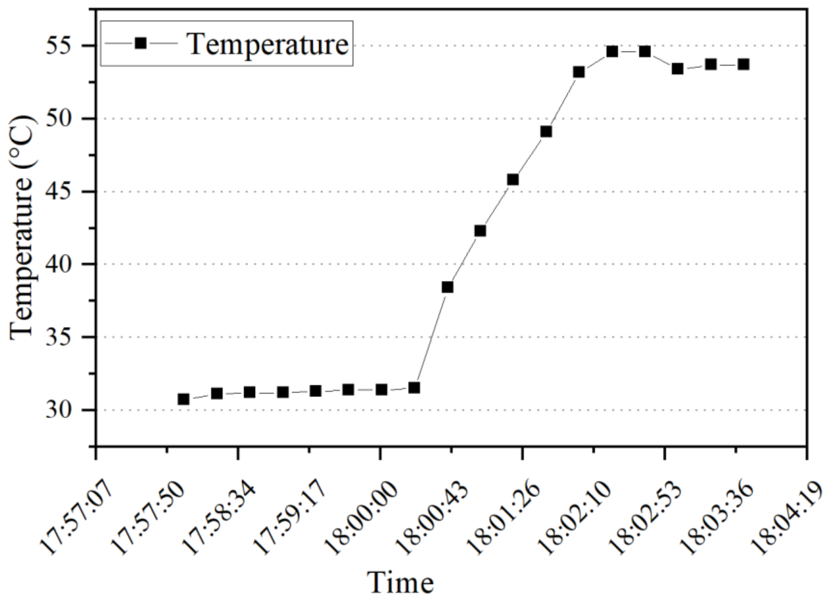


Figure 16: Temperature Data Graph of Vehicle device

In the graph above the data is taken from table 3. In our controlled test, the Temperature data in our DHT11 sensor started from 30 °C and rose to 55 °C. But the actual temperature in a car engine area can be close to 100°C, but the sensor was limited to around 0~50 °C according to datasheet. So, the actual data could not be taken in any project grade component available in the Bangladesh market. As we could not take any usable data from this particular sensor, we could not make any low, medium, and high-risk data range for this particular sensor.

CNG Data:

In our test environment, we tested CNG leakage but artificially made a leakage in the engine. Here, the graph below shows the response of the sensor during that time taken from table 4.

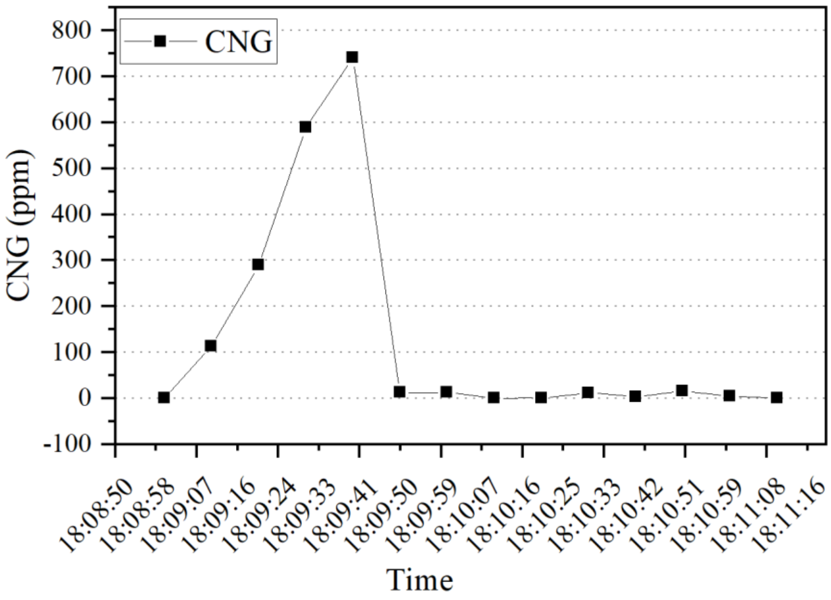


Figure 17: CNG Data Graph of Vehicle device

In this particular scenario, we let the gas leak for almost 45 seconds. We could not do more than that due to safety reasons. Within this time from 18:09:00 to 18:09:45, we let the gas leak. During this time, the gas data rises much faster than we initially expected. In less than 30 seconds gas data reach 500. In a potential gas leakage time, this can be alarming so, we decided to make and high-risk range for this gas higher than 500. Bellow 500 it is the safe range.

Smoke Data:

In the test, smoke data gradually increases and decreases. In table 4, we saw the smoke response from 18:10:10 to 18:11:10.

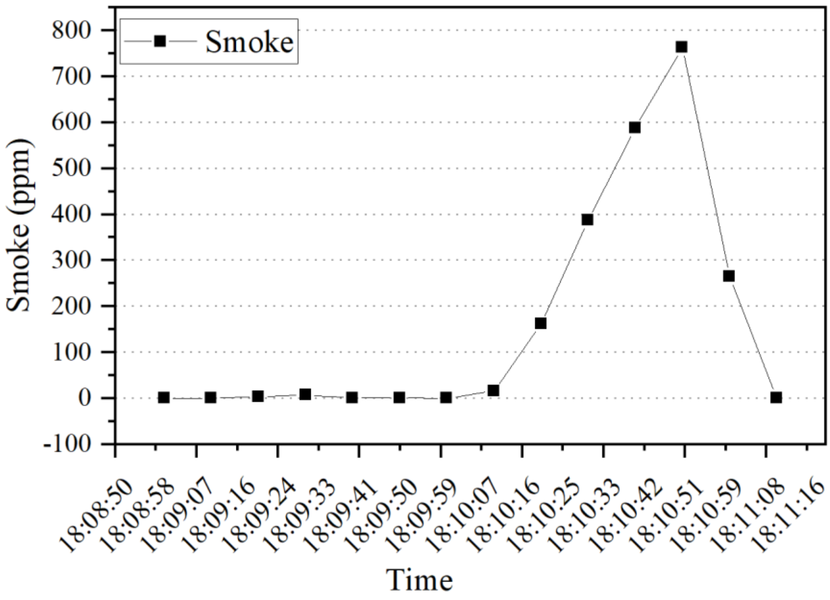


Figure 18: Smoke Data Graph of Vehicle device

Here, the data rises for almost 50 seconds. Then we opened the bonnet to release the gas. In that time, the data of the sensor gradually decreases rather than rapidly changing. Here the maximum data reached almost 800 in 50 seconds. Just like CNG value, this data also increases faster than we expected. Considering the low amount of smoke, we decided to make low-risk, medium-risk and high-risk range from 0 to 399, 400 to 599 and 600 to infinity respectively.

Acceleration Data:

Acceleration and deceleration occur simultaneously when humans drive. We tend to speed up a little and apply brakes a little to maintain a constant speed. Hard braking is fairly rare without any issues. From this data, we can find where whether the vehicle hit the brakes or not.

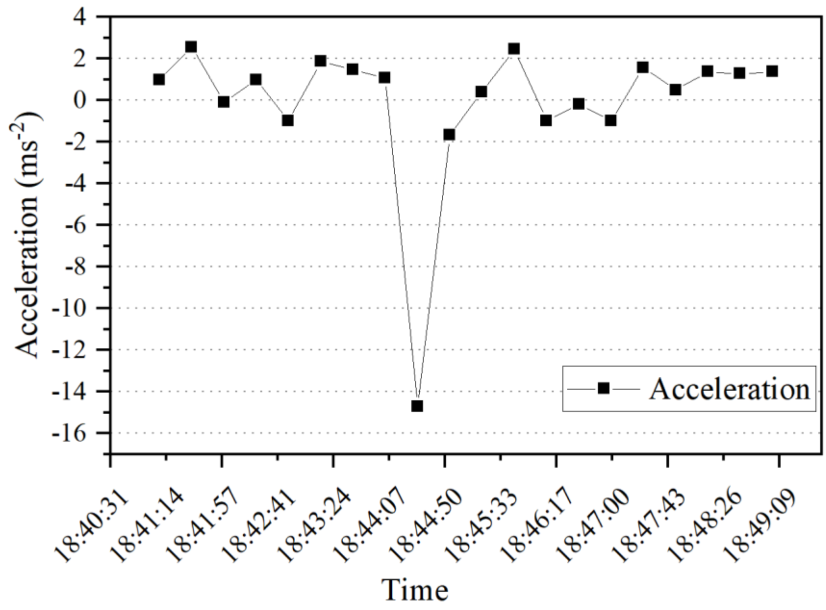


Figure 19: Acceleration Data Graph of Vehicle device

This graph mainly demonstrates data from table 5. From there, we focused on only acceleration values. Here, if the vehicle is accelerating, the data will be positive and negative for when the vehicle is decelerating. Here we get one very hard brake of -14.7 ms^{-2}. These kinds of sudden negative peaks actually determine the hard brakes. Being present in that situation, we determine the low, medium and high value for the acceleration; 0 to 9.9, 10 to 16.9, and 17 to infinity respectively.

Piezo Element Data:

Other than these five data, we also used a piezo sensor. But those data are fairly static. According to the datasheet of the piezo element, it has a resistance change due to impact. With correct resistance set up around it, we can take ADC value ranging from 0-1023. We could not test any real car accident, but with lab testing to the sensors, we could find with a very large impact, piezo will give around 900-1023 value, for medium-impact, it ranges from 500 to 899 and for small shaking or vibration, 0-200 value. So, we decided to make low-risk, medium-risk, and high-risk value according to these data, which was, 0 to 599, 600 to 899 and 900 to 1023 respectively.

Flame sensor:

As we do not need the level of flame, as any kind of flame in a car could result in a catastrophe, especially in a CNG powered car, we tried to use this sensor as digital sensor despite giving an analog sensor. As this sensor gives 1023 ADC in no flame and 0 in high flame, considering flame in a closed dark area will definitely be high amount to sensor despite the amount, we make a threshold value of 500 to determine if there is any flame under the bonnet or not.

To summarize the response characteristics of the sensors, we can represent these by our determined low, medium, and high-risk levels.

Sensor	Risk Level	Value Range
CNG	Low	0 - 499
	Medium	500 – inf.
	High	500 - inf
Smoke	Low	0 - 399
	Medium	400 - 599
	High	600 - 1023
Piezo	Low	0-599
	Medium	600 - 899
	High	900 - 1023
Acceleration	Low	0 – 9.9
	Medium	10 – 16.9
	High	17 - inf
Flame	Low	1023-500
	Medium	499 - 0
	High	499 - 0

Table 6: Vehicle Device Threshold value chart

The ranges can be more dynamic, depending on the situation. This can be achieved by more dynamic test cases which were not possible to carry out due to lack of resources. These values will be used in the server to notify mobile application and emergency responders depending on whether it is a medium risk situation or high-risk situation. Details about this will be described in Chapter 4.

3.6 Cost analysis

The approximate cost of building our vehicle device is BDT.8500. Most of the safety features that we get to see nowadays come in-built in vehicles, such as ABS (anti-lock braking system), seatbelts, back-up cameras, auto-steering, airbag, etc. However, there are quite a few monitoring systems available. For instance, Easytrax Limited in Dhaka, Bangladesh, offers a range of monitoring services such as monitoring multiple vehicles, live traffic update, fuel monitoring, route playback, etc. that ranges from packages of BDT. 4000, BDT.6000 to BDT.13500. Other than that, there are a variety of cameras that can be installed externally on the dashboard. Another company by the name ONSTAR provides automatic crash response via crisis assistance that operates on a more manual but similar principle as our device. However, none of these options is as hands-free, automated and comprehensive as ours; therefore, there is no scope for a fair evaluation and comparison.

To conclude the chapter, we like to point out the key takeaway of this chapter. We have discussed the build and coding with its working mechanism of the vehicle deice. We have seen and analyze the sensor data to determine the minimum and maximum threshold value to use in the server, which will be a comprehensive system in chapter 4.

Chapter 4

Comprehensive System Design

In this chapter, the Comprehensive System Design is broken down into several stages. The System Working Mechanism includes the discussion of Server Data Receive, Data Routing and Storing, Functions and Notifications, User App Realtime Data Feed, Emergency Responder Notification System and the Overall System Data Flow.

4.1 Significance of the Comprehensive system

Our comprehensive system consists of two individual devices connected to a common Cloud Server, and it responds to two kinds of emergencies: road accidents in vehicles and fire accidents in homes. There are devices available in the market which focus on reducing the chances of an accident occurring or try to reduce the impact of said accidents. However, there are very few systems that deal with the aftermath of an accident in terms of notifying the assistance providers. The novelty of our system is it specializes in the response, i.e. the aftermath of an accident and more important does so in a completely automated manner. This system was designed with the idea of reducing response rates to provide help to those who need it as fast as possible when they need it, and if they need it. And any process that is manually operated will incorporate more lag due to human processing and reaction time as opposed to something that is fully automated. The complete automated response is what sets our comprehensive system apart from anything that is available in the market.

Now, why it is essential to connect these individual devices to a larger, more comprehensive system is because it allows the devices to send their data to a single server system where everything is connected instead of smaller systems that cater to an individual device's needs more aptly. That makes monitoring and processing of the information sent from the sensors and GPS easier and more efficient. By being part of one comprehensive system, it avoids repeating the same operations for both the systems via different platforms, thus making it cost-effective and quicker and more streamlined. Furthermore, it makes it possible to simplify the notification process from the analyzed data collected from the sensors. If we were to build the notification process for an individual system only (for example the car device), it would not be possible to incorporate such a thorough, extensive and systematic approach as each system would result in different feedback being sent to the user and would fail to be cost-effective.

4.2 System design in stages

Our system is designed to work in four stages.

- **Stage 1:** Networking things or the sensing layer, there is a collection of data from several sensors along with GPS Location Data. In the vehicle kit, the sensors used are temperature, gas, flame, GPS, piezo, and accelerometer. The Devices will be placed inside the vehicle, and real-time data of multiple variables such as temperature, various gas concentrations such as that of methane, butane, humidity, a flame in case of a fire, acceleration, and impact, etc. surrounding the kit will be continuously recorded. These sensors are connected to Arduino Mega2560, which is built in the PCB along with the GPRS module and sensors mentioned.

- **Stage 2:** Data acquisition (DAQ) systems where the collected data is sent to the microcontroller unit for analog to digital conversion, storage, and further processing. Instead of using a separate DAQ board, the Arduino Mega2560 acts as the DAQ system itself, which not only reduces cost but also helps in making the design of the system more compact. The sensors will be sending the collected data to the Arduino Mega2560 using various protocols such as I2C, serial, analog, etc. These are then converted to digital signals, amplified, and sent to the Cloud Server for further processing.

- **Stage 3:** Cloud analytics where the already processed data is sent to the cloud server for further in-depth processing via the GPRS module. Cloud computing consists of three categories, Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). Our system is designed using the IaaS category, which uses a dedicated physical server. In this stage, the user also gets real-time data update through their Mobile App. The data collected from sensors are sent to the cloud server via TCP Protocol using GPRS Module and MQTT protocol using NodeMCU. Once in the cloud server, if specific parameters are crossmatched in case of an accident, a notification to the appropriate authority will be sent.
- **Stage 4:** Identification stage where the Server identifies a situation where the data indicates whether the vehicle needs any assistance. If it does, it uses that data to notify the nearest assistance provider, for example, Fire Service or Hospital, by analyzing the vehicle's location and sending it to them. From the start, data received sent to the written programs, where the values from the sensors are matched by the parameters for minimum and maximum threshold values as set by us, which represent the minimum threat and maximum threat levels depending on the experimental values of the Sensor. Upon matching the said parameters, notification is sent to the mobile application and the assistance providers and emergency contact when applicable. The system keeps track of the live location of the vehicle. It sends it to its nearest assistance centre (Hospital, police station, fire station) in case of an emergency if and when there is assistance needed.

To make a quick rescue or response, it is imperative for the system to register a considerable range of hospitals, fire stations, and ambulance services in a given area. Once they are registered, using a combination of real-time location tracking via GPS (Global Position

Tracking System) and the now registered assistance centers in the Server, live location of accident can soon be sent to them.

Here is a visual representation of the system stages.

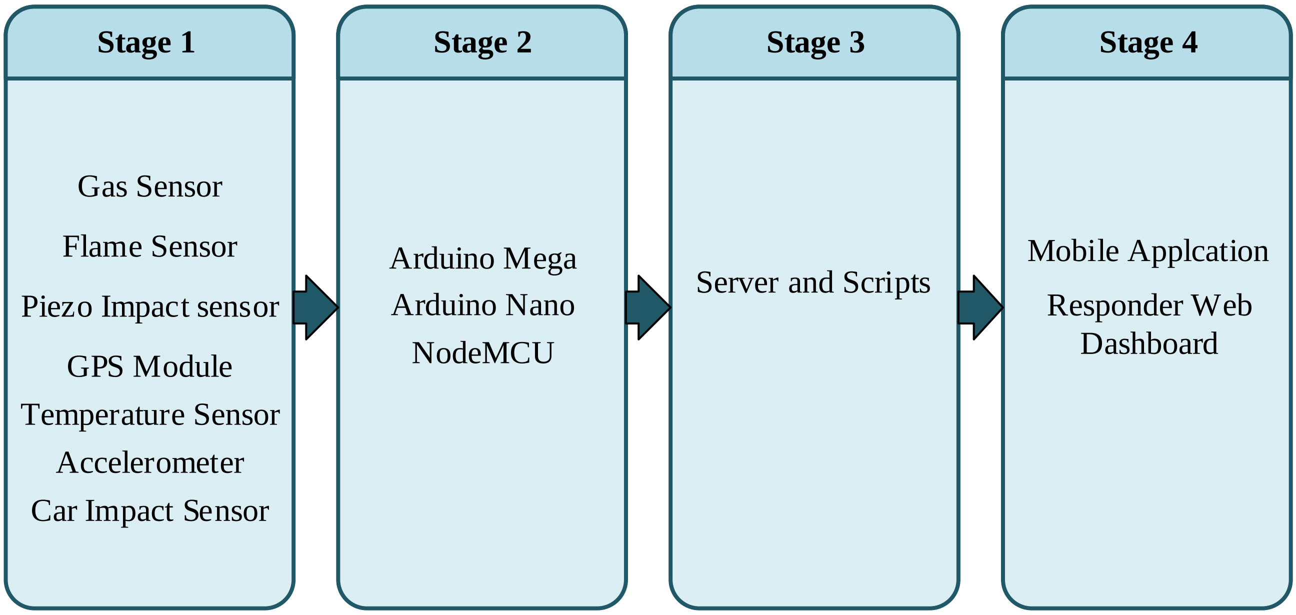


Figure 20: Stage Diagram of the Comprehensive System

4.3 System Working Mechanism

This IoT system works on several steps of data filtering in both data storing and feedback. Each of them is used by different Script and server applications.

4.3.1 Overall System Data flow

To describe the overall data flow, we can simplify the whole system to a simple system, where data collected from sensors is stored to MCUs and then sent to the Server. The Server takes the data and immediately stores the Data for future use. Then the Server considers the last data stored and checks with the minimum and maximum threshold values to determine whether it needs a user warning or emergency responder (Details in section 4.2.4) and acts accordingly. It can be shown in a simplified data flow diagram below:

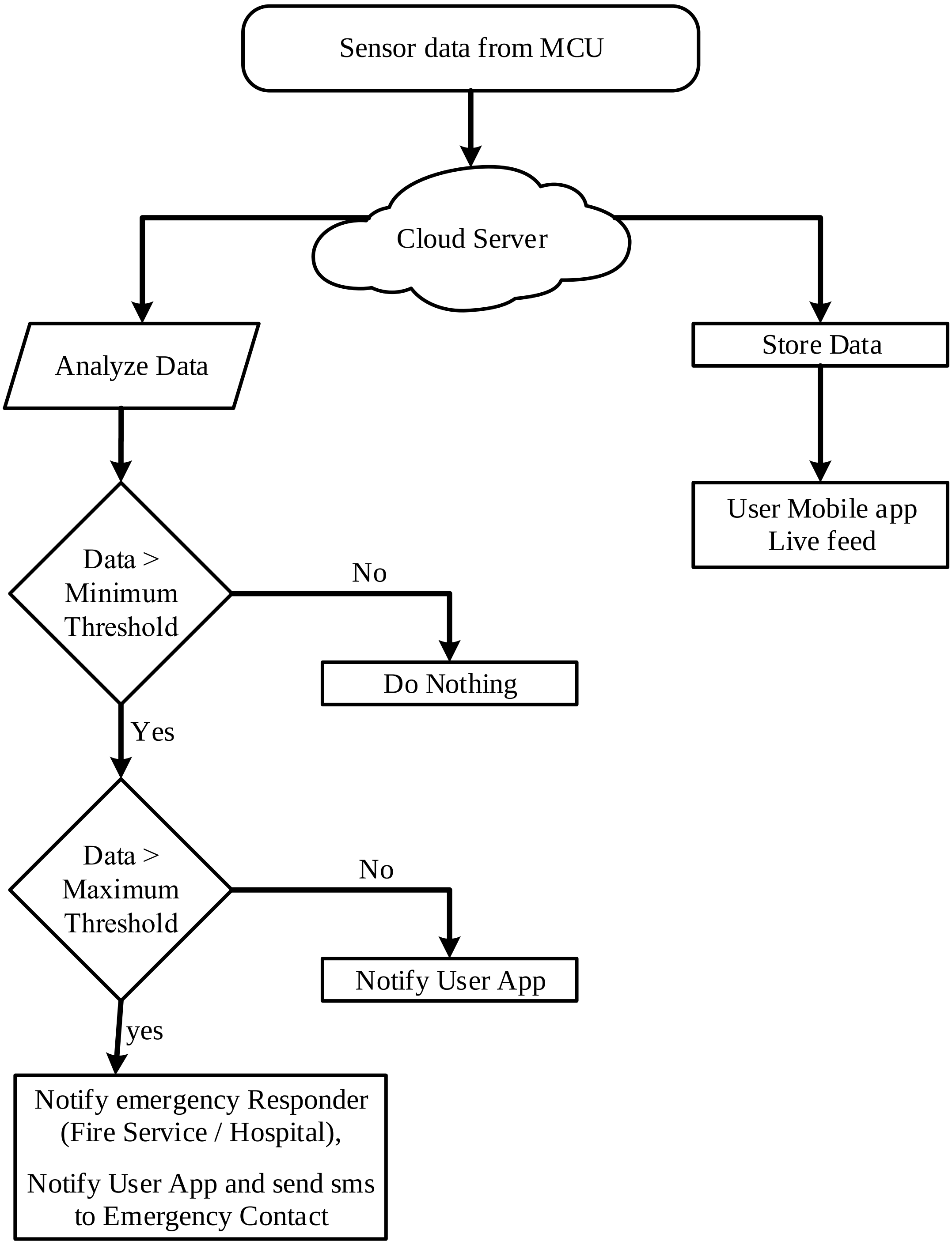


Figure 21: Data flow of the Comprehensive System

4.3.2 Server Data Receive

To receive the data coming from both Home device and vehicle device, we first need to set up the Server accordingly. We used a Virtual Private Server (VPS) as the brain of our system. The Server runs on CentOS 7, a Linux based operating system where we use NodeRed, a cross-platform data transfer software. NodeRed receives and sends data from a variety of method.

As we are using different connection protocols for home and vehicle device, NodeRED is very helpful for that. We use TCP and MQTT protocol to receive data on the Server. To collect data via MQTT, there should be an MQTT broker. MQTT Broker is responsible for obtaining the data and sending to different clients.

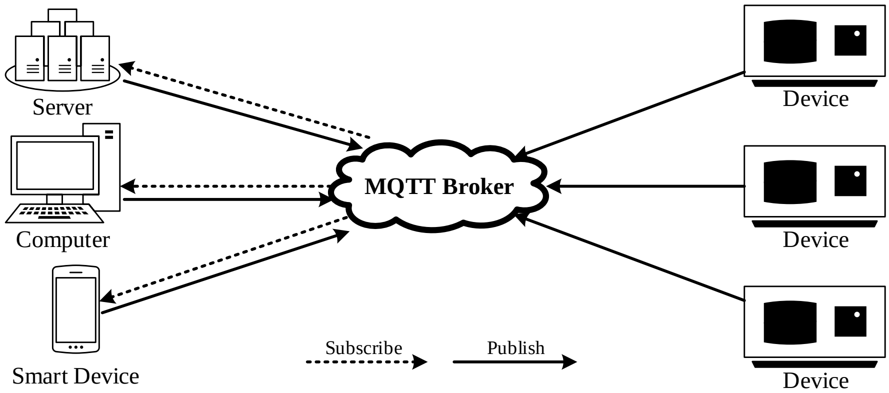


Figure 22: MQTT Broker General working Diagram

We used “Mosquitto,” a very famous free MQTT broker in our Server for connection. Together with Mosquitto and NodeRED, all of the data can flow from our devices to the Server.

Figure 23: NodeRED and Mosquitto logo

Data is received by the NodeRED in two different ways for two kinds of devices.

Home Device: Data from Home Device is sent via MQTT protocol from NodeMCU. MQTT Broker “Mosquitto” needs to set up some usernames and passwords and receives data in specific “Topic” to differentiate the different sources of data. From Home Device NodeMCU, Data is sent using the username and password and a topic name of “home kit”. When data specified in this specified topic name, in the server IP and Port, the MQTT broker receives the data and sends it to the NodeRED sketch, where we can address those data to further processing.

Vehicle Device: Data is received from the vehicle device in a much simpler way. It uses the TCP protocol. In TCP, no extra broker is needed. Data is openly sent to a specific IP address using allocated Port for TCP; in our case, it is 1850. Data can be received directly from NodeRED and sent to other services for further processing.

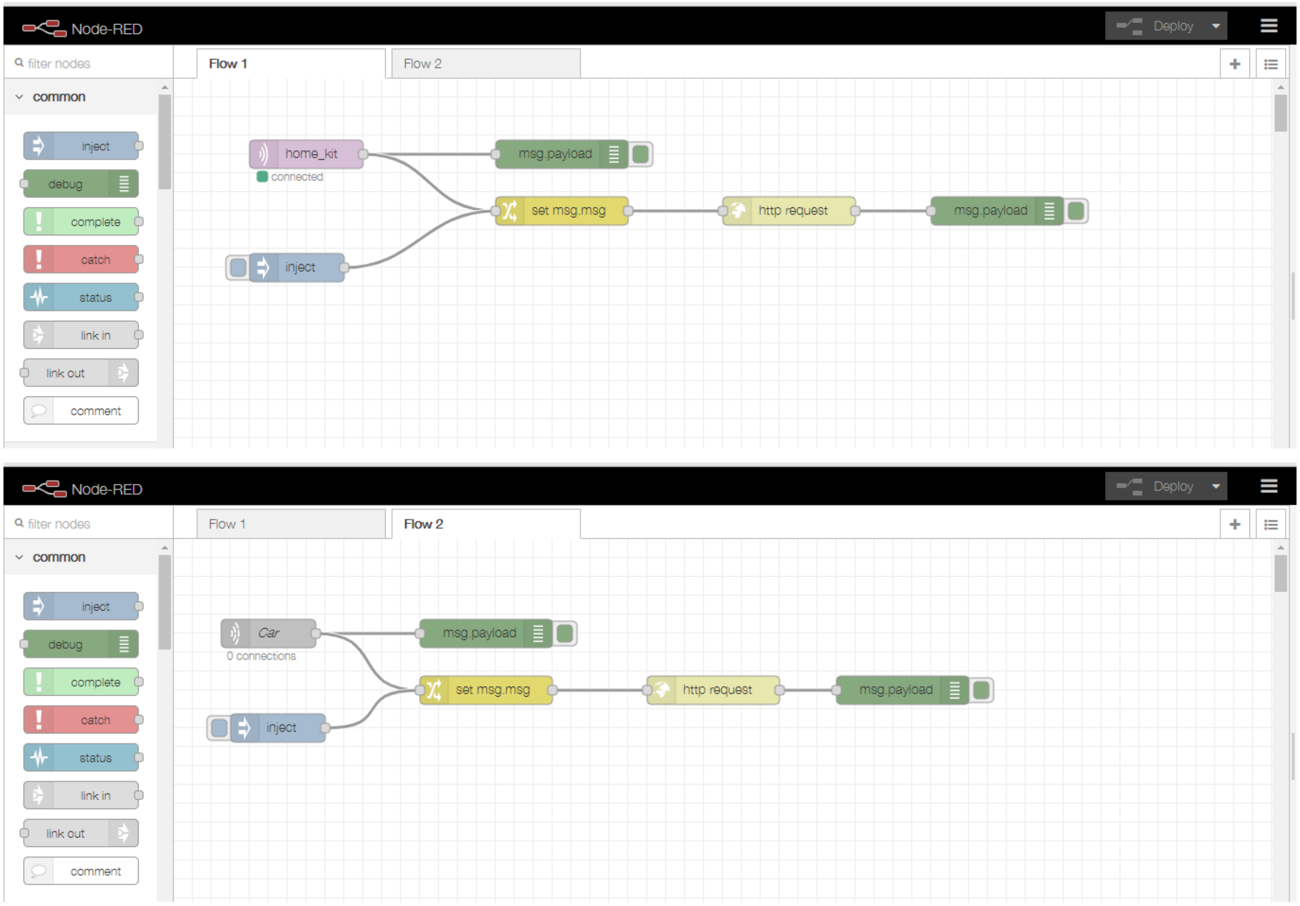


Figure 24: NodeRed Flow for Two Devices

After Receiving data, NodeRED can be set up for doing multiple tasks. In our case, we take the data to convert it to a string message payload and sent it for further processing.

4.2.3 Data Routing and Storing

After receiving data, those need to be sorted and stored. All the data collected is in a String format. In a lengthy text separated by "," all the data from a device sent at a time is received. Those data need to be saved in a MySQL database for conditioning and monitoring. To insert the data into a MySQL database, we used PHP. PHP is a server-side scripting language. We used a script to take the received data and split those data by checking ",".

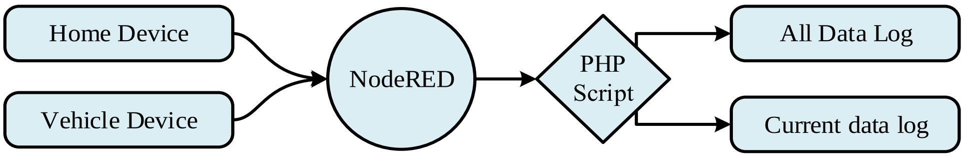


Figure 25: Overall Data Routing diagram

After splitting the data, the Script will save each device data into two separate tables. In one, all the data will be stored one by one. In the other, only the current data will be stored in just one row, which will later be used in conditioning and monitoring. The storing table or the data log is to collect data for future improvement of the system. And to minimize data reading errors, we approached to store current data in a separate table to collect the data.

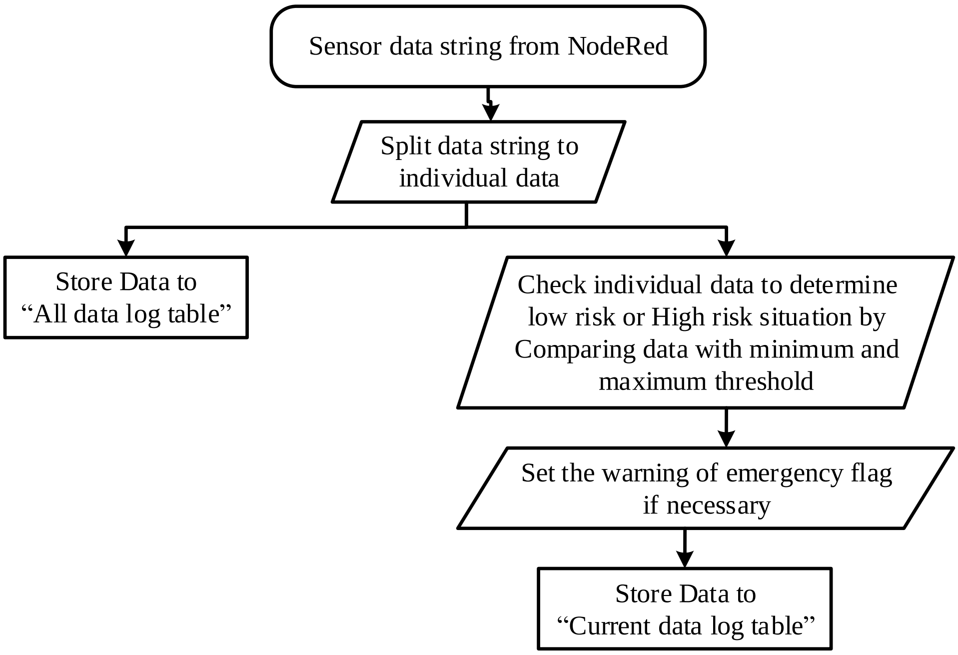


Figure 26: PHP Script Data Conditioning algorithm

This PHP data conditioning algorithm shows how the script is designed to work. The details on checking the risk levels are in chapter 4.3.4. the image below shows the phpMyAdmin console for our SQL database where our data is stored.

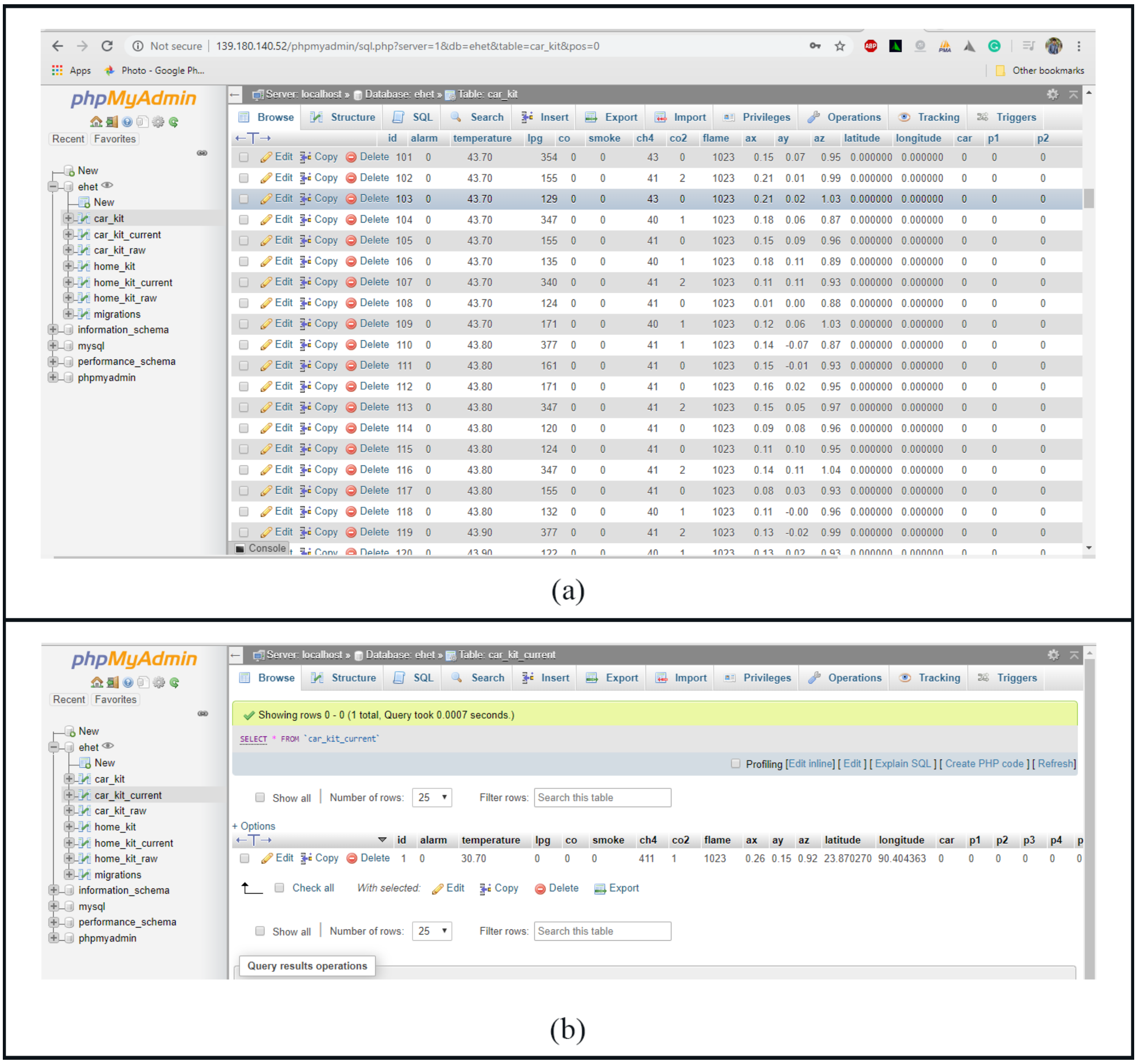


Figure 27: Data stored and Database (a) all data log, (b) Current data log

After data is stored, another PHP Script runs to check all the latest data and set the emergency parameter, whether it needs a user lookup or professional help. Depending on the criteria, it places data that requires a warning to the user or sends the notification to the Emergency responder for support.

4.3.4 Algorithm and Notification

The algorithm in the Script checks the current parameter with previously given parameters and changes the emergency flag data value of the existing data table between 0 and 1. Changing flag value will lead the emergency responder service like fire service or Hospital to respond to the emergency. In other cases, the mobile app will get a notification warning about a low emergency warning that can be handled without any professional help.

To determine whether it is a simple warning only for the application, or for the emergency responder, we determine two threshold values for some sensors to identify three situations.

- No risk
- Medium risk
- High risk

Minimum threshold value helps to determine if the situation is no risk or medium risk. And the maximum threshold value helps to determine the high-risk condition when help from hospital or fire service is needed.

The table below shows the minimum and maximum values for each sensor data we used to determine the risk factor. These values were determined in Chapter 2 and Chapter 3 in the section name Sensor Data Characteristics in Table 2 and Table 6.

	Sensor data	Minimum threshold	Maximum threshold
Home Device	Flame Sensor	600	300
	LPG	500	2000
	Smoke	600	1000
	temperature	49	49
Vehicle Device	Flame Sensor	500	500
	Smoke	400	600
	CNG	500	500
	Piezo sensor	600	900
	Acceleration	-5	-10
	GPS Signal	Not found	N/A

Table 7: Threshold values for home and vehicle kit

After crossing the minimum threshold value, we will make the warning flag High, which will give notification to a smartphone app. Then we will check if the maximum threshold value is crossed and matched with other corresponding values to make the emergency flag high, which will eventually direct the notification to the emergency responder.

The Home Device algorithm works as follows:

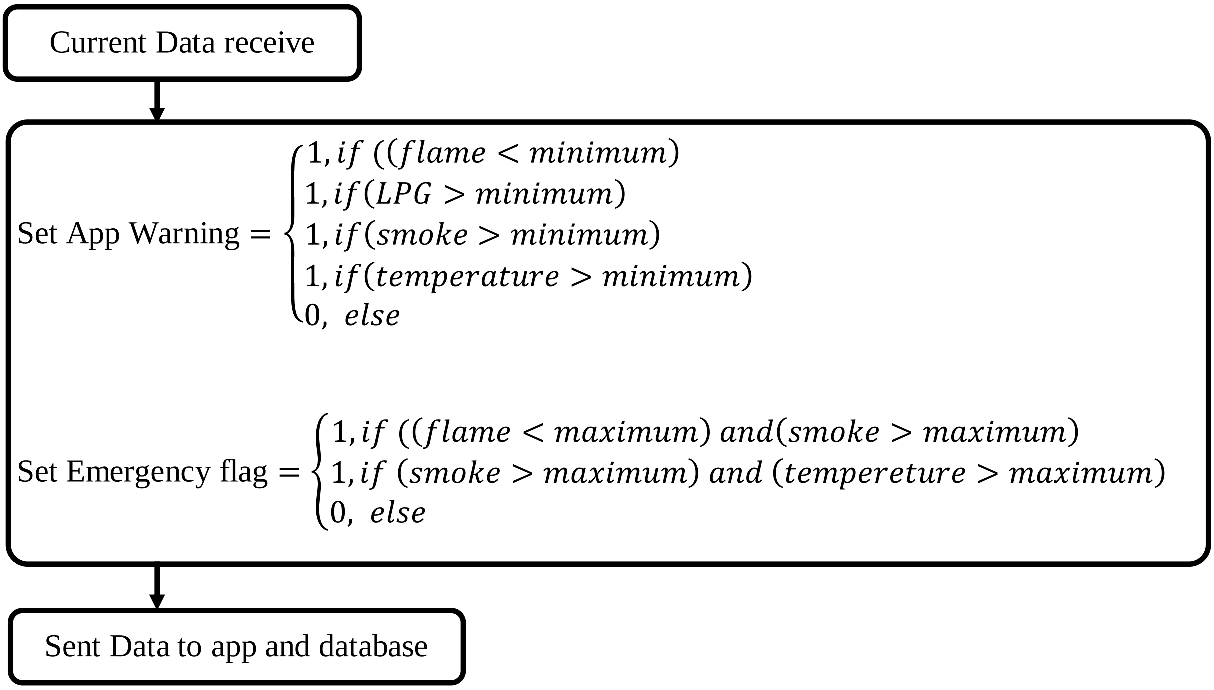


Figure 28: Home Device Threshold checking algorithm

Here the minimum values will be checked and used to give warning to the mobile app we developed to communicate with the server. And for multiple emergency data, for example, flame and smoke will check at the same time to determine whether an emergency exists or not. Higher smoke can make the flame invisible, so we also checked the smoke and temperature.

The Vehicle Device algorithm works as follows:

Current Data receive

↓

$$\text{Set App Warning} = \begin{cases} 1, if\ (piezo\ impact > minimum) \\ 1, if\ (acceleration < \ minimum) \\ 1, if\ (\ flame > minimum) \\ 1, if (Smoke > minimum) \\ 1, if\ (CNG > minimum) \\ 1, if (\ GPS\ not\ found) \\ 0,\ \ else \end{cases}$$

$$\text{Set Emergency flag} = \begin{cases} 1, if\ (car\ airbag\ triggers) \\ 1, if\ (piezo > maximum)\ and\ (acceleration < maximum) \\ 1, if\ (\ flame < maximum)\ and\ (Smoke > maximum) \\ 0, else \end{cases}$$

↓

Sent Data to app and database

Figure 29: Vehicle Device Threshold checking algorithm

Here warning will check the minimum values and give warning to the mobile app. And for multiple emergency data, for example, piezo and acceleration will be checked at the same time to determine possible collation. There is an extra check of a vehicle's native collision sensor, which activates in modern vehicles in a severe collision.

4.3.5 Algorithm Testing

To Simulate these algorithms, we make many test scenarios from the reference of gathered response data before. Some of the key scenarios are,

- Fire in the kitchen
- Gas leakage in the kitchen
- Fire in the living room
- Fire in the car engine
- Gas leakage in the car
- Heavy collision on the side of the car.

Different scenarios will cause different types of action based on given algorithm parameters shown in Table 7. Sensor data used here to simulate are in the range of ideal data value to trigger a situation.

For kitchen Data,

If data crossed minimum threshold values of Home device, then warning=1

If data crossed maximum threshold values of Home device, then emergency = 1

For Car Data,

If data crossed minimum threshold values of Vehicle device, then warning=1

If data crossed maximum threshold values of Vehicle device, then emergency = 1

Here are the test cases with selected sensor data for Simulation. All the sensor data here are selected based on the sensor response we discussed in the previous chapters.

Fire in the Kitchen: Kitchen fire will generate fire, heat, and there will be a significant amount of smoke. It is less likely to be any LPG to detect in a fire so, the test sensor values for this test case selected below are as follows.

Flame	Temperature	LPG	Smoke
150	53	3	1050

Table 8: Detection levels in case of Fire

With the data shown in Table 8, A notification should be sent to mobile, and emergency responder service should get a notification as well.

Gas leakage in the Kitchen: Gas leakage will increase the LPG value only, but the ambience remains the same. So, the test sensor value for this test case selected below.

Flame	Temperature	LPG	Smoke
900	32	700	3

Table 9: Detection levels in case of Gas Leakage

With the data shown in Table 8, A notification should be sent to the mobile user app only.

Fire in the living room: Due to electricity short circuit or any possible scenario living room fire will have no LPG. Still, flame should be a very low value, and temperature will be higher than normal temperature, smoke can be huge or low depends on the severity of the fire. So, the test sensor value for this test case selected for two possible scenarios given below.

No.	Flame	Temperature	LPG	Smoke
01	200	40	1	680
02	170	42	2	1260

Table 10: Detection levels in case of Fire due to Short-Circuit

For the first scenario, A notification should be sent to the mobile user app only.

For the second scenario, both user app and emergency responder service should get a notification to attend to the emergency.

Fire in the car engine: Fire in the engine will cost flame value to go low and form a huge black smoke. Here ax stands for acceleration and P1, P2, P3, P4 and P5 stand for Piezo values. So, the test sensor value for this test case selected below.

Flame	Smoke	CNG	ax	airbag	GPS	P1	P2	P3	P4	P5
50	630	93	0.3	0	23.770616,90.397149	100	115	103	94	137

Table 11: Detection Values in case of Fire in the Vehicle Engine

For this test case, data is shown in table 11. A notification should be sent to mobile user application as well as to the emergency responder to conduct rescue and fire suppression operation.

Gas leakage in the car: Gas leakage will make the LPG or CNG values to go high and other values to be normal. So, the test sensor value for this test case is as selected below.

Flame	Smoke	CNG	ax	airbag	GPS	P1	P2	P3	P4	P5
1023	2	543	0.12	0	23.770616,90.397149	103	105	104	103	98

Table 12: Detection Values in case of Gas Leakage in the Vehicle

For data in table 12, A notification should be sent to the mobile user app, and the data values will be shown.

Heavy collision on the side of the car: Collision will be more likely to cause an increase in the piezo sensors' values referred to as P1 to P5. Other values will remain as previously shown. So, the test sensor value for this test case selected below.

Flame	Smoke	CNG	ax	airbag	GPS	P1	P2	P3	P4	P5
980	340	2	-6.5	1	23.770616,90.397149	102	115	863	917	137

Table 13: Detection Values in case of Vehicle Collision

Here, A notification should be sent to the mobile user app as well as to the emergency responder to conduct a rescue operation.

With all of these test cases, responses of the system are shown below:

Test	Test Case	The expected response of the system	Outcome
1	Fire in the kitchen	Both app and emergency response service will receive a notification.	Success
2	Gas leakage in the kitchen	Only smartphone app will get a notification	Success
3	Fire in the living room 1	Only smartphone app will get a notification	Success
4	Fire in the living room 2	Both app and emergency response service will receive a notification.	Success
5	Fire in the car engine	Both app and emergency response service will receive a notification.	Success
6	Gas leakage in the car	Only smartphone app will get a notification	Success
7	Heavy collision on the side of the car	Both app and emergency response service will receive a notification.	Success

Table 14: System Response for Test Cases

However, these test cases can be configured in thousands of ways. And threshold values and notification algorithm can be updated over time for better efficiency and for more security.

4.3.6 User App Realtime Data Feed and Alert

For the user, there is an Android application to look up the data stored in real-time. This app is also responsible for giving alert and sending SMS in High-risk situations mentioned previously.

If the user wants to see the Home device data, the app will call a PHP script written specifically to show data in the "Current data" table, which will eventually return the data from those table's specific rows and columns. Every time the script is called that returns the current data. That data will be shown in the application one data each second.

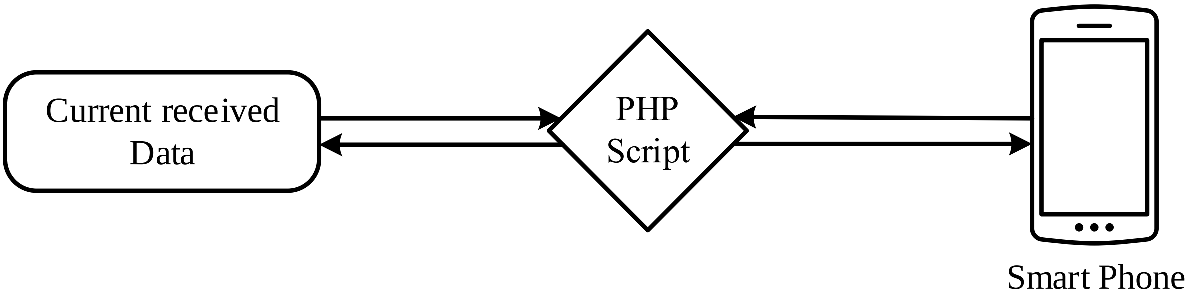


Figure 30: Data Flow for User Application

The user app has both home and vehicle device connectivity, so one app can monitor multiple devices. The Mobile Application checks the data every time to check if there is an emergency or if an accident has happened. If so, there will be an alert on the Mobile Application to notify the user about the incident. And in case of High-risk value, it will send SMS to 3 contact number from the mobile itself. User can update these Emergency contact any time they want.

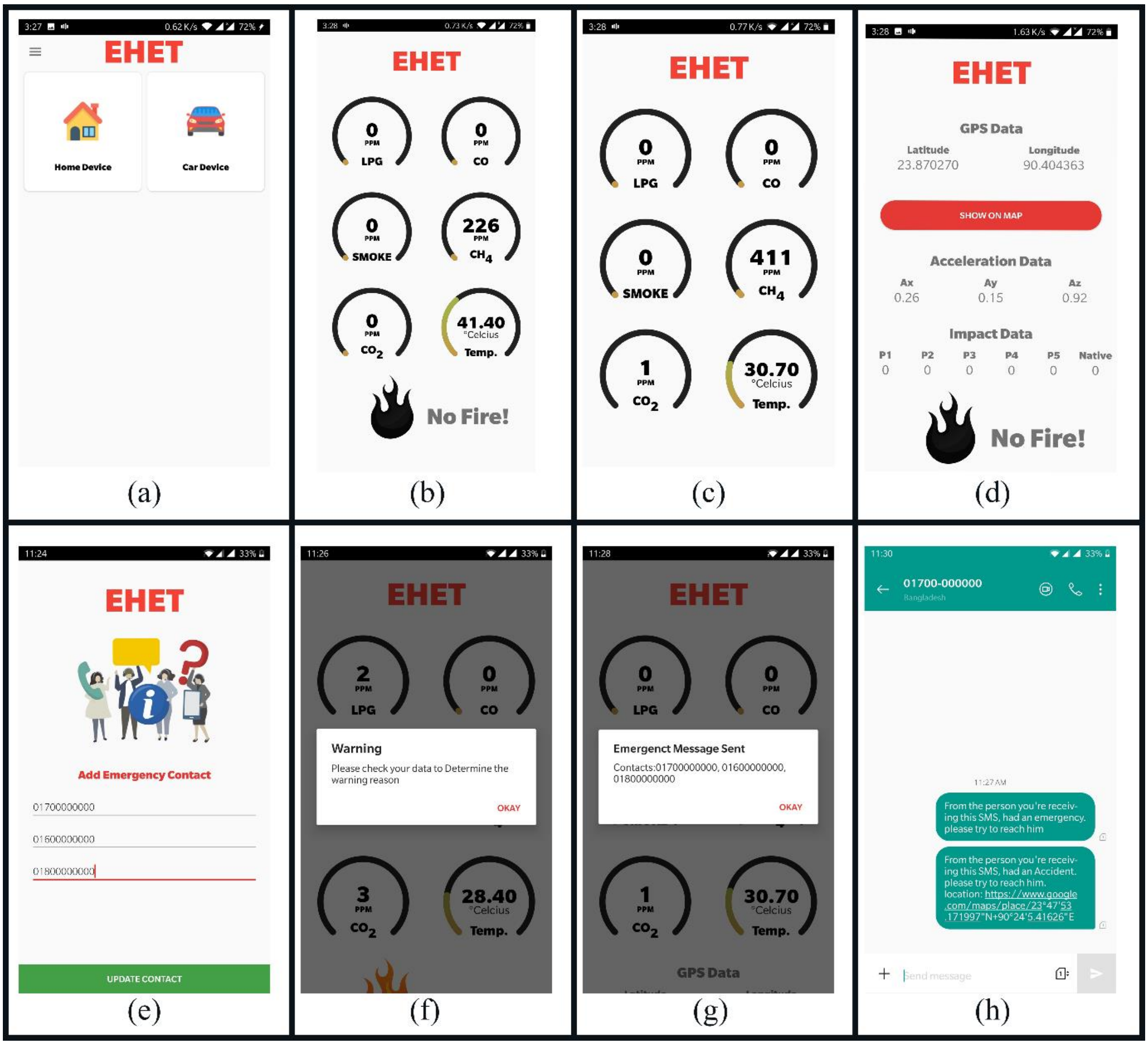


Figure 31: Mobile Application. (a)Different devices in the system, (b)Home device data feed without warning, (c)Car device Gas data feed, (d) car device location speed and impact data feed, (e) Emergency Contact information save, (f) Medium Risk warning, (g) High-risk warning and emergency Contact SMS send alert, (h) SMS sent from mobile to emergency contact.

4.3.7 Emergency Responder Notification System

As referring to the term “Emergency responder”, we are primarily referring to the Hospitals, Ambulance Service provider, Fire Service, and civil defence. They are the critical emergency responders in our system. As they need to respond to emergencies of the users, the more area with responder is registered to the system; the more efficient is the service of the system.

Every Emergency Responder has a web dashboard, where they can find the notification emergency, for example, fire in the building or car accident nearby. They will have a different panel with their location as the home location for them. When an emergency is determined by the PHP scripts as previously explained, there will be an emergency flag in the database.

The Dashboard checks all of those flag fields through an API, which is another PHP Script. If any of the tables is found with an emergency flag ON, that device location is pulled by the Script immediately and starts to look for nearby hospitals and fire service. For example: in our current web dashboard, we have entered five hospitals located to demonstrate the system. They are Dhaka Medical College Hospital, Ibn Sina Hospital, Square Hospital, Kurmitola General Hospital, Apollo Hospital. These five hospitals spread across Dhaka City. While in an emergency, the location is compared with these five hospital locations to find the nearest Hospital using “Google Maps Location API,” then only the nearest Dashboard gets permission to show the emergency notification with visual and sound.

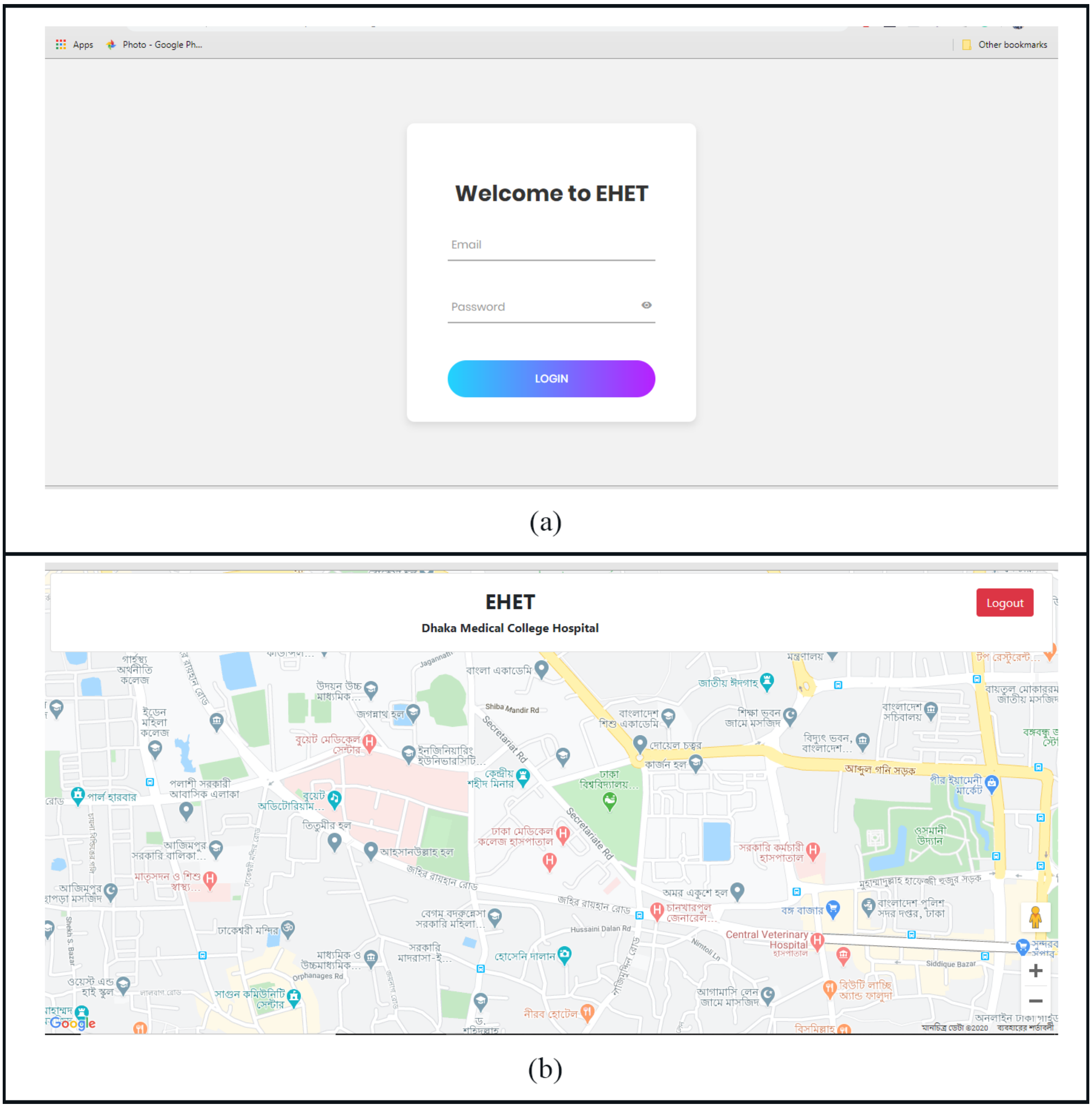


Figure 32: Emergency Responder Web Dashboard. (a)Login page, (b)Normal page with the respective hospital or fire service-centered map,

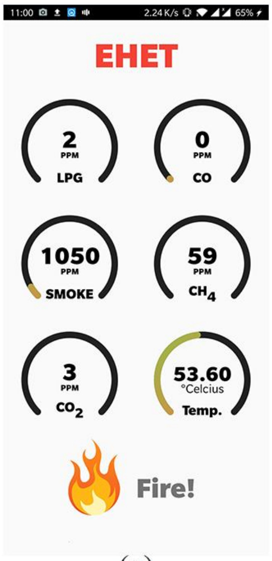
Here is shown the Dashboard of Dhaka Medical college after login. There is a simple map showing the hospital home location.

4.3.8 Emergency Notification simulation

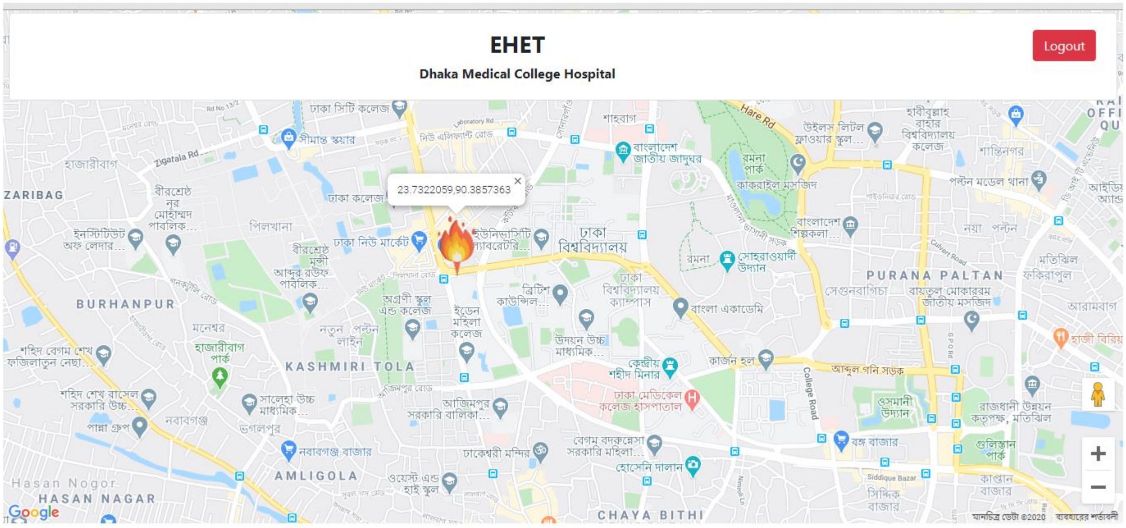
If we take our first notification simulation from Table 4, and the location is Near Nilkhet, let's assume the Latitude and Longitude location is 23.7322059, 90.3857383.

Flame	Temperature	LPG	Smoke
150	53	3	1050

The response of our mobile app and web Dashboard as follows,



(a)



(b)

Figure 33: Emergency response Simulation for the notification system. (a) Smartphone application notification indicating fire, (b) Web dashboard of the nearby hospital with location showing the emergency.

When there is an emergency, the mobile app will get the notification, and the web dashboard will show the location of near Nilkhet, it directly indicates a fire icon with a loud sound.

Clicking the fire icon there shows the exact location of the incident and where to send the emergency responder team to help. After this point, it is up to the emergency responder team to help those in need.

4.3 System Security and Privacy

Internet of things (IoT) has emerged globally for its functionality and services. In the past 5-6 years, the usage of IoT smart devices has grown at an exceptional scale, reaching billions of users worldwide. People rely more on smart home devices because they can monitor their home and control their home appliances remotely which include baby and older people monitoring, pet monitoring, light, fan, heater etc. The worrying factor is that the smart devices connecting to the internet is not that much secured, and there can be a breach of privacy from malicious attacks. The attackers can intrude into family activities and can take full control of the smart devices installed in houses. For the past few years, this has been a major concern in the IoT industry as millions of IoT devices have been compromised by attackers. To overcome this problem, there has to be more device-level protection and network-level security. The types of attacks include network-level attack, software and encryption attacks. For minimizing and overcoming these attacks, researchers worldwide have come up with different schemes and modules which we will be discussing in the next segment.

4.3.1 Security Literature Review:

A tremendous amount of research has been made globally for security and privacy control for IoT devices, but still, the research is at beginner level as the attacks are still occurring in large numbers. The research made by the different researchers mainly focuses on the potential attacks possible and try to stop that by proposing different countermeasures and schemes. Sivaran S. and Vishwanath A.[28] in their paper focuses on network-level security and propose to implement more security measures in the network that usually the device manufacturers lack. They described their reasoning for choosing to work on network-level defence rather than device-level protection. The first reason for their choice is network-level security can be implemented for an entire range of IoT devices rather than device-level security which focuses only on a particular device. The second reason is that network-level security adds an additional layer of protection that can increase device-level security provided by the manufacturers. They proposed an external entity called the SMP (Security Management Provider) which provides extra security to the users at the network level for the IoT devices installed in their homes. SMP interacts with ISP network on one side and home users on the other side. The role of SMP is to exercise limited configuration control over the ISP network or home router on behalf of the user/client, with not being present on the data path. They also built and implemented a prototype of their system, which includes ISP access switch, ISP network controller, SMP security orchestrator and web GUI (JavaScript/ HTML) controlled by the SMP. They tested their prototype with two devices: Hue light bulb and nest smoke alarm and promises to work with more IoT devices in their future work to increase the effectiveness of their system.

Javaid U. [29] in his paper focuses on DDoS attacks and how it can be stopped by using blockchains, specifically Ethereum: a type of blockchain. DDoS attack is basically sending a huge number of data from one or more devices to create network traffic and then compromise the system. In their approach, they used blockchains which use smart contracts. Smart contracts are self-executable computer programs. The smart contract is basically coded to differentiate between legitimate and illegitimate devices. The smart contract is developed using Solidity, which is a contract-oriented, high-level language for the Ethereum virtual machine environment. There are two phases of operation of Solidity, which are initialization and deployment. In the initialization stage, a server node brings the smart contract into effective action which will help in recognizing the legitimate host. There is a gas limit associated with each device to limit the resources of the device, which will ultimately defend against DDoS attacks. In the deployment stage, different IoT devices' nodes in the system will communicate with the server node. The server will verify devices and register them in the contract list of legitimate devices. If it finds anything suspicious about any device, it will delete the specific node from the system immediately and deny access.

Ho and De Leung[30] have discussed security issues of smart lock system installed in houses. This smart lock system can operate even if the owner is a bit distant from the door. An attacker corrupting the smart lock system can unlock the door without the owner knowing which is indeed scary. They proposed a solution of wearing a wearable device which communicates wirelessly with a hand-held vibrator installed in the door. When the owner wearing the wearable device comes in a range of the door, the smart lock establishes a secure connection, and when the owner touches the lock, the smart lock sends a one-bit intent signal to the wearable device. The BLE's pairing-based channel or DTLS could be used for the secure connection. After receiving the signal, the owner will send an 'Unlock' message to the smart lock, and finally, the door will be unlocked.

Sharaf and W. Saad [31] proposed an approach which uses the device's fingerprint features to verify an object's legitimacy. They used the idea of transfer learning for the authentication process. They worked on a two-fold approach. The first approach is that they made sure whether the signal is sent by a single object. After that, it verifies the legitimacy of the device. Infinite Gaussian Mixture Mode (IGMM) is used for the first phase of the research where fingerprints for different objects follow a multivariate Gaussian distribution. The next phase is just comparing the results from IGMM with the usual cluster shape for the device. They tested their approach and found better result for authentication compared to other conventional techniques.

4.3. Security and Privacy analysis of our system:

Our comprehensive system consists of two devices: the home kit and the vehicle kit. Each kit will have a large number of users taking the services. All the data of different sensors from different users are collected and stored in a MySQL database for monitoring and conditioning. The data collected from different sensors are sent to a cloud server using TCP protocol and MQTT protocol. We can assume huge data are being stored in the server every day and from the manufacturer's side, it is our duty to protect all the data coming from different clients. We have studied and analyzed the security and privacy issues of IoT devices which we discussed in the previous segments. After thorough research, we have decided to give more emphasis on data encryption and limiting the IP address access to minimize the chances of illegal access. We have set up a specific username and password for both MySQL database and MQTT broker "Mosquito" to secure the connection. We can say the security is double in data sharing and storing as both database, and MQTT broker is password protected. In addition to the data encryption, we have limited the IP address access, which means our server and database cannot be accessed using any other IP address except the specific IP address which we have set. This increases security at a considerable amount. Another thing that we took into consideration is the range of data that will be shared in the dashboard to the emergency services. Our clients' personal data or device data will not be shared with the emergency responders for their privacy. Only the concerned data will be sent like name, contact number and live location. For future implementation, we have plans to use Google firewall service for correct authentication of the user.

Chapter 5

Conclusion

The last chapter entitled the Limitations, Possibilities and Conclusion discuss the limitations and drawbacks followed by potential future work and possibilities of the comprehensive system and individual devices.

5.1 System Limitations and Drawbacks

5.1.1 Limitations

Before releasing the kits on the market, hundreds, if not thousands of trial runs must be made for data collection. As this system is designed to detect accidents, we would need to create a similar mock environment to collect optimum data. This would mean the purposeful occurrence of accidents in many cases which is not totally feasible and might require a lot of resources and increase expenditures.

5.1.2. Drawbacks

a) The emergency contact will be notified via SMS using the consumer's phone. Therefore, in the case of an accident, if the consumer's mobile device is out of charge or out of credit balance, the emergency contact will be failed to be notified.

   When a consumer buys the device, necessary information such as his name, mobile number, address and more important details of an emergency contact will be provided and stored in the server for future use. Hence, even if the consumer's mobile device fails to notify the emergency contact, the stored information in the server can be used instead to let the emergency contact know about any accident that may have taken place.

b) If the magnitude of the accident is too extreme, it might destroy the vehicle or home kit which would require getting replaced later. Or in a worse and rare scenario, the kit might get destroyed before being able to notify one of the service providers.

   More sensors can be attached to the device to attain more variety of data that can help to pick up more sensitive changes in parameters around the vehicle that will in turn help to send an alert earlier than when a situation gets too extreme.

c) In the case of the vehicle kit, as a GSM module establishes the constant connection between the kit and cloud sensor, travelling to a remote area with no network would disrupt that connection.

   When the overall economic infrastructure of a country improves, it ensures that telecommunication companies can reach the farthest corners of the country and provide their network.

d) If one of the sensors on the kit gets disabled or damaged due to some reason, it would need to be repaired manually before it can record data again.

e) In case of the home kit, we have assumed that the home where the kit will be installed in will already have a basic fire alarm system as is the case in most modern-age households. It would not make much sense to, therefore, implement the house kit in a place without a basic fire alarm system with sprinklers.

   With the help of an initial profit generation that can be invested in the company or funding from a trusted source, the home device can be modified to come with an accompanying fire sprinkler system in the future.

5.2 Future Work and Possibilities

As with most projects, our project also has a multitude of options for improvement and further development.

a). With the addition of a fingerprint sensor, we can personalize the kits to a higher degree and make it more exclusive to the user. This not only adds a more personal feel to the system but also improves security and can be used to store unique information of the user that will help in quicker identification in case of an accident.

b). In the case of vehicle kits, with the help of a little research, the more accident-prone roads and road junctions can be localized in a selected area where cameras can be installed. The rapid enhancement of image processing can be utilized to record real-time pictures from the location of the accident, thus solidifying the data being collected from the sensors. Real-time pictures will give a clearer perspective on the location and magnitude of the accident. Not only externally, but cameras can also be installed inside the vehicle to give a better understanding of the situation of the passengers inside.

c). A health-monitoring band can also be developed and added to the entire system which consumers can wear and can be used to record more accurate data of real-time physical condition of them in both day-to-day lives and unfortunate case of an accident. For example, in the case of the vehicle-kit, if the passengers wear a health-monitoring band, variables such as their heart rate, blood pressure, body temperature can also be sent to the nearest hospital for better analysis of the situation.

d). A separate band can be made, specifically keeping the safety of women and children in mind. The band can be designed in a way wherewith the press of a button a notification can be sent to the nearest police station in case of a possible threat of any sort.

e). In case of the home kit, we can develop a sprinkler system which can be activated upon analyzing the data from the kit and be used to put out small fires before notifying the fire station unnecessarily.

5.3 Conclusion

The problem we are seeking a remedy for is a very common scenario in our day-to-day life. We hope to improve this inconvenience by many folds. According to a recent report by the World Health Organization (WHO), approximately 1.35 million people die each year as a result of road traffic crashes, and Road traffic crashes cost most countries 3% of their gross domestic product. Moreover, 93% of the world's fatalities on the roads occur in low- and middle-income countries, even though these countries have approximately 60% of the world's vehicles. To add to that, Road traffic injuries are the leading cause of death for children and young adults aged 5-29 years. These are extraordinary numbers that show how a simple issue such as road congestion or lack of communication might result in the loss of many human lives. Even though this is the kind of project that has room for incredible further enhancements in the future, the current model is already more reliable and sophisticated than anything available in the market in the present day. Ours is a full system that can be accessed by both victim and emergency response service for faster and most cases, automatic alert, and relatable response in the lowest possible period. In an ever-so-changing modern world with ground-breaking technology time, and, most importantly, lives should not be lost at the cost of such trivial issues that could have a simplified yet sophisticated solution.

Publications

Conference:

- Sahriar Habib, Zawata Afnan, Sakib Anam Chowdhury, Sarah Altaf Chowdhury and Abu S.M. Mohsin "IoT Based Accident Detection and Emergency Response System" The 2020 5th International Conference on Cloud Computing and Internet of Things (CCIOT 2020), Okinawa, Japan during 22-24, September 2020 (Scopus Indexed Conference).

Journal:

- "Design and development of an automated comprehensive system for emergency assistance service" (Expected to submit Elsevier - Journal of Network and Computer Applications, Impact Factor 5.5)

References

[1] S. Shivani and S. Shoney, "IoT based car accident detection and notification algorithm for general road accidents. ," *International Journal of Electrical and Computer Engineering (IJECE).* vol. 9, no. 5, 2019, doi: 10.11591/ijece.v9i5.pp4020-4026

[2] K. Patel, "Utilizing the Emergence of Android Smartphones for Public Welfare by Providing Advance Accident Detection and Remedy by 108 Ambulances.," *International Journal of Engineering Research & Technology (IJERT),* vol. 2, no. 9, 2013.

[3] C. Thompson, J. White, B. Dougherty, A.Albright, and D. C. Schmidt, "Using smartphones to detect car accidents and provide situational awareness to emergency responders," *Proceedings of the International Conference on Mobile Wireless Middleware, Operating Systems, and Applications,* 30 June- 2 July 2010.

[4] E. K. Priya, P. Manju, V. Mythra, and S. Umamaheshwary, "IOT Based Vehicle Tracking and Accident Detection System," *International Journal of Innovative Research in Computer and Communication Engineering,* vol. 5, no. 3, 2017.

[5] N. Chaturvedi and P. Srivastava, "Automatic Vehicle Accident Detection and Messaging System Using GSM and GPS Modem," *International Research Journal of Engineering and Technology (IRJET),* vol. 5, no. 3, 2018.

[6] A. Mane and J. Rana, "Vehicle Collision Detection and Remote Alarm Device using Arduino," *International Journal of Current Engineering and Technology,* vol. 4, no. 3, 2014.

[7] M. S. Amin, J. Jalil, and M. B. I. Reaz, "Accident detection and reporting system using GPS, GPRS and GSM technology," *2012 International Conference on Informatics, Electronics & Vision (ICIEV),* 2012, doi: 10.1109/ICIEV.2012.6317382.

[8] R. Angeline, A. S., and A. Narayanan, "Fire Alarm System Using IOT," *International Journal of Innovative Technology and Exploring Engineering (IJITEE),* vol. 8, no. 6S3, 2019.

[9] E. Saraswathi, A. Kumar, J. Singh, J. Mohanty, and Y. Mishra, "Arduino Based Home Automation System Using MQTT Protocol Incorporating Internet of Things (IOT)," *Journal of Network Communications and Emerging Technologies (JNCET),* vol. 8, no. 5, 2018.

[10] N. Johnsaida, V. R. Lakkisetty, and Tadi Shalini, "IOT Based Smart Fire Emergency Response System," *International Journal of Advance Research and development,* vol. 3, no. 2, 2018.

[11] M. Reddy and R. K, "Fire Accident Detection and Prevention monitoring System using Wireless Sensor Network enabled Android Application," *Indian Journal of Science and Technology,* vol. 9, 05/18 2016, doi: 10.17485/ijst/2016/v9i17/93097.

[12] A. Imteaz, T. Rahman, M. K. Hossain, M. S. Alam, and S. A. Rahat, "An IOT based Fire Alarming and Authentication System for Workhouse using Raspberry Pi 3," *International Conference on Electrical, Computer and Communication Engineering (ECCE),* February 16-18 2017, doi: 10.1109/ECACE.2017.7913031.

[13] S. Gosrani, A. Jadhav, K. Lekhak, and D. Chheda, "Fire Detection, Monitoring and Alerting System based on IOT," *International Journal of Research in Engineering, Science and Management,* vol. 2, no. 4, April 2019.

[14] R. K. Kodali and S. Yerroju, "IoT Based Smart Emergency Response System for Fire Hazards," *International Conference on Applied and Theoretical Computing and Communication Technology* December 2017.

[15] C. Sun, "Application of RFID Technology for Logistics on Internet of Things," *2012 AASRI Conference on Computational Intelligence and Bioinformatics,* vol. 1, pp. 106-111, 2012.

[16] D. Moeinfar, H. Shamsi, and F. Nafar, "Design and Implementation of a Low-Power Active RFID for Container Tracking at 2.4 GHz Frequency " *International Journal of Advances in Internet of Things,* vol. 2, pp. 13-22, January 10 2012, doi: 10.4236/ait.2012.22003.

[17] S. Madakam, R. Ramashawami, and S. Tripathy, "Internet of Things (IOT): A Literature Review," *Journal of Computer and Communications,* vol. 3, 2015.

[18] T. Arampatzis, "A Survey of Security Issues in Wireless Sensors Networks, in Intelligent Control. Proceeding of the IEEE International Symposium on," *Mediterrean Conference on Control and Automation,* pp. 719-724, 2005.

[19] K. L. Su, "Automatic Fire Detection System Using Adaptive Fusion Algorithm for Fire Fighting Robot," *2006 IEEE International Conference on Systems, Man and Cybernetics,* pp. 966-971, 2006.

[20] B. Cho, J. Bae, and S. Jung, "Image Processing-Based Fire Detection System Using Statistic Color Model," *2008 International Conference on Advanced Language Processing and Web Information Technology,* pp. 245-250, 2008.

[21] F. Saeed, A. Paul, A. Rehman, W. H. Hong, and S. H., "IoT-Based Intelligent Modeling of Smart Home Environment for Fire Prevention and Safety," *Journal of Sensor and Actuator Networks,* vol. 7, no. 11, 2018, doi: 10.3390/jsan7010011.

[22] K. Muhammad, J. Ahmad, I. Mehmood, S. Rho, and S. W. Baik, "Convolutional Neural Networks Based Fire Detection in Surveillance Videos," *IEEE Access,* vol. 6, pp. 18174-18183, 2018.

[23] A. B. Faiz, A. Imteaj, and M. Chowdhury, "Smart vehicle accident detection and alarming system using a smartphone," *2015 International Conference on Computer and Information Engineering (ICCIE),* pp. 66-69, 2015.

[24] A. Zainab and A. Hamid, "Car Accident Detection and Notification System Using Smartphone," *International Journal of Computer Science and Mobile Computing (IJCSMC),* vol. 4, pp. 620-635, 2015.

[25] M. S. Amin, M. A. S. Bhuiyan, M. B. I. Reaz, and S. S. Nasir, "GPS and Map matching based vehicle accident detection system," in *2013 IEEE Student Conference on Research and Developement,* 16-17 Dec. 2013 2013, pp. 520-523.

[26] F. Bhatti, M. A. Shah, C. Maple, and S. U. Islam, "A Novel Internet of Things-Enabled Accident Detection and Reporting System for Smart City Environments," *Sensors (Basel, Switzerland)* vol. 19, no. 9, 3 May 2019, doi: 10.3390/s19092071.

[27] T. H. Yee and P. Y. Lau, "Mobile vehicle crash detection system," in *2018 International Workshop on Advanced Image Technology (IWAIT),* 7-9 Jan. 2018 2018, pp. 1-4.

[28] V. Sivaran, H. Gharakhelli, A. Vishwanath, R. Boreli, and O. Mehani, "Network-Level Security and Privacy Control for Smart-Home IoT Devices," *IEEE WiMob Workshop on Selected Topics in Wireless and Mobile Computing (STWiMob),* 2015, doi: 10.1109/WiMOB.2015.7347956.

[29] U. Javaid, A. K. Siang, M. N. Aman, and B. Sikdar, "Mitigating IoT Device based DDoS Attacks using Blockchain," *CryBlock'18: 1st Workshop on Cryptocurrencies and Blockchains for Distributed Sysytems,* June 15 2018, doi: 10.1145/3211933.3211946.

[30] G.Ho, D. Leung, P. Mishra, A. Hosseini, D. Song, and D. Wagner, "Smart Locks: Lessons for Securing CommodityInternet of Things Devices," *Proceedings of the 11th*

ACM on Asia Conference on Computer and Communications Security, pp. 461-472, 2016.

[31] Y. Sharaf-Dabbagh and W. Saad, "On the authentication of devices in the Internet of Things," *2016 IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM),* pp. 1-3, 2016.

Appendix A.

To execute our device for all our test porous, we used the code bellow for the devices mentioned.

Home Device Arduino Code

```
#include <SoftwareSerial.h>  //library for serial communication
SoftwareSerial server(2,3); // RX, TX pin set for the serial communication

// including library for DHT11 and setting variables for that
#include "DHT.h"
#define DHTPIN 8
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
float temperature=0.00;

// including library for Gas sensors and setting variables for that
#include <MQ2.h>
#define mq2_pin A3
int lpg, co, smoke;
MQ2 mq2(mq2_pin);
#define mq4_pin A2
#define mq135_pin A1
#define flame_pin A0
int mq4,mq135,flame;

boolean alarm =0;
String Location = 23.7706914,90.4001676; // native location for the home device
String data="";

void setup() {
  Serial.begin(9600);
  server.begin(115200);
  pinMode(mq2_pin, INPUT);
  pinMode(mq4_pin, INPUT);
  pinMode(mq135_pin, INPUT);
  pinMode(flame_pin, INPUT);
  dht.begin();
  mq2.begin();
}

void loop() {
```

```
// calling different functions for different sensors to take reading
  getDHT();
  getMQ2();
  getMQ4();
  getMQ135();
  getflame();

  if(flame<150){
    alarm = 1;
  }
  else{
    alarm = 0;
  }
//Storing all the sensor data to a string for sending via serial to NodeMCU
data=(String)alarm+','+(String) temperature+','+ (String)lpg+','+ (String)co+','+
(String)smoke+','+ (String)mq4+','+(String)mq135+','+(String)flame+','+Location;

  Serial.println(data); //serial print to bug fix
  server.println(data); //sending string to NodeMCU
  delay(500);
}
void getDHT(){
  temperature = dht.readTemperature(); // take temperature
}
void getMQ2(){
  lpg = mq2.readLPG(); //take mq2 reading
  co = mq2.readCO();    //take CO reading
  smoke = mq2.readSmoke();  // take smoke reading
}
void getMQ4(){
  mq4 = analogRead(mq4_pin); // take mq4 reaing
}
void getMQ135(){
  mq135 = analogRead(mq135_pin); //take mq135 reading
}
void getflame(){
  flame = analogRead(flame_pin); //take flame sensor reading
}
```

Home Device NodeMCU Code

```
#include <ESP8266WiFi.h>   //library for wifi
#include <PubSubClient.h>  //library for MQTT
#include <SoftwareSerial.h> //library for Serial communication

SoftwareSerial mySerial(4,14); // RX, TX for serial communication to Arduino

//Setting home wifi credential to connect
const char* ssid = "test";
const char* password = "test";
// setting MQTT broker credential and server formation to connect to server
const char* mqttServer = "139.180.140.52";
const int mqttPort = 1883;
const char* mqttUser = "ehet";
const char* mqttPassword = "0000";
const char* topic_name = "home_kit";

WiFiClient espClient;
PubSubClient server(espClient);

String data="";
int str_length;

void setup() {
  Serial.begin(115200);
  mySerial.begin(115200);
  WiFi.begin(ssid, password);    // connect to wifi
// checking and re connecting to wifi if not connected.
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.println("Connecting to WiFi..");
  }
  Serial.println("Connected to the WiFi network");
//connection to the server via MQTT broker
  server.setServer(mqttServer, mqttPort);
  while (!server.connected()) {
    Serial.println("Connecting to MQTT...");
    if (server.connect(topic_name,mqttUser,mqttPassword)) {
      Serial.println(" connected ");
    } else {
      Serial.print(" failed with state ");
      Serial.print(server.state());
      delay(2000);
```

```
    }
  }
}

void loop() {
  server.loop();
  if (mySerial.available()) {
    data = mySerial.readStringUntil('\n');  //receiving data from Arduino
  }
  Serial.println(data);   // serial print for debugging

//sending data to the server via publishing the to the server
  str_length = data.length()+1;
  char send_data[str_length];
  data.toCharArray(send_data,str_length);
  Serial.println(send_data);
  server.publish(topic_name,send_data);
  data="";   // resetting data string after sending to the server
  delay(1000);
}
```

Car Device Arduino Code

```
#include <SoftwareSerial.h>  // library for serial communication
#include "DHT.h"  // library for temperature sensor
#include <MQ2.h>  // library for gas sensor
#include<Wire.h>  //I2C library for communication.
#include <TinyGPS.h> // GPS library
////////////DHT11 variable set/////////////////////////////////
#define DHTPIN 2
#define DHTTYPE DHT11   // DHT 11
DHT dht(DHTPIN, DHTTYPE);
float temperature = 0.00;
/////////////GAS variable set //////////////////////////
#define mq2_pin A1
int lpg, co, smoke;
MQ2 mq2(mq2_pin);
#define mq4_pin A2
#define mq135_pin A3
#define flame_pin A0
int mq4, mq135, flame;
/////////MPU6050 variable set ///////////////////////////
const int MPU = 0x68;
```

```
int16_t AcX, AcY, AcZ, Tmp, GyX, GyY, GyZ;
float ax, ay, az;
////// GPS  variable set /////////////////////
float lat = 00.0000, lon = 00.000;
String latitude,longitude;
#define gpsSerial Serial2
TinyGPS gps;
/////// piezo variable set ////////////////
#define piezo1 A6
#define piezo2 A7
#define piezo3 A8
#define piezo4 A9
#define piezo5 A10
#define piezo6 A11
#define piezo7 A12
#define piezo8 A13
#define piezo9 A14
#define piezo10 A15
int p1,p2,p3,p4,p5,p6,p7,p8,p9,p10;
///////////car in variable set //////////////////
#define car_in 3
int car=0;
boolean alarm =0;
String data = "";

void setup()
{
  Serial.begin(9600);
  Serial3.begin(9600); //GPS
  gpsSerial.begin(9600);
  pinMode(mq2_pin, INPUT);
  pinMode(mq4_pin, INPUT);
  pinMode(mq135_pin, INPUT);
  pinMode(flame_pin, INPUT);
  pinMode(piezo1, INPUT);
  pinMode(piezo2, INPUT);
  pinMode(piezo3, INPUT);
  pinMode(piezo4, INPUT);
  pinMode(piezo5, INPUT);
  pinMode(piezo6, INPUT);
  pinMode(piezo7, INPUT);
  pinMode(piezo8, INPUT);
  pinMode(piezo9, INPUT);
```

```
  pinMode(piezo10, INPUT);
  pinMode(car_in, INPUT);
  dht.begin();
  mq2.begin();
  accelerometerSetup();
  sim_init();
}

void loop(){
//calling different functions to take reading of different sensors
  getDHT();
  getMQ2();
  getMQ4();
  getMQ135();
  getflame();
  get_accelerometer();
  get_gps();
  get_car();
//some in device parameter check
  if(car==1){
    alarm =1;
  }else if(flame <200){
    alarm =1;
  }else if(ax<-10 && (p1>800 || p2>800 || p3>800 || p4>800 || p5>800 || p6>800 || p7>800 ||
p8>800 || p9>800 || p10>800) && smoke >5000){
    alarm =1;
  }else{
    alarm =0;
  }
// all sensor data to a string to send
  data = (String)alarm+','+(String)temperature + ',' + (String)lpg + ',' + (String)co + ',' +
(String)smoke + ',' + (String)mq4 + ',' + (String)mq135 + ',' + (String)flame + ',' +
(String)ax + ',' + (String)ay + ',' + (String)az + ',' +latitude+ ',' +longitude+ ',' +(String)car+
',' +(String)p1+ ',' +(String)p2+ ',' +(String)p3+ ',' +(String)p4+ ',' +(String)p5;
  Serial.println(data);
  send_data(data);  //data send via GPRS function
}
void send_data(String dat)
{
  Serial3.println("AT+CIPSEND");  //taking the module to send mode
  delay(300);
  Serial3.print(dat);  // sending the data
  Serial3.println((char)26);  //signaling the module that data sending is done
```

```
  delay(500);
}

void updateSerial(){      //function to check if the GPRS returns anything
  delay(500);
  while (Serial.available()){
    Serial3.write(Serial.read());//Forward what Serial received to Software Serial Port
   }
  while (Serial3.available()){
    Serial.write(Serial3.read());//Forward what Software Serial received to Serial Port
   }
}
void sim_init(){      // sim set to our specify mode
  Serial.println("Initializing...");
  delay(1000);
  Serial3.println("AT");  // check if module connected
  updateSerial();
  Serial3.println("AT+CSQ"); //Signal quality test, value range is 0-31 , 31 is the best
  updateSerial();
  Serial3.println("AT+CCID"); //Read SIM information to confirm whether the SIM is
plugged
  updateSerial();
  Serial3.println("AT+CREG?"); //Check whether it has registered in the network
  updateSerial();
  Serial.println("Done Initializing.");
  Serial3.println("AT+CSTT=\"blweb\""); // setting the apn
  updateSerial();
  delay(1000);
  Serial3.println("AT+CIICR"); // printing the netword unique id
  updateSerial();
  delay(1000);
  Serial3.println("AT+CIFSR");  // printing the IP address of the device
  updateSerial();
  delay(1000);
  Serial3.println("AT+CIPSTART=\"TCP\",\"139.180.140.52\",\"1850\""); //MQTT
connection
  updateSerial();
  delay(1000);
}
void getDHT() {
  temperature = dht.readTemperature();  //DHT temperature
}
void getMQ2(){
```

```
  lpg = mq2.readLPG(); //lpg value reading
  co = mq2.readCO(); // CO value reading
  smoke = mq2.readSmoke(); // smoke value reading
}
void getMQ4(){
  mq4 = analogRead(mq4_pin); //mq4 data read
}
void getMQ135(){
  mq135 = analogRead(mq135_pin); //mq135 data read
}
void getflame(){
  flame = analogRead(flame_pin); // flame data read
}
void accelerometerSetup(){   //setting up accelerometerwith I2C address
  Wire.begin();
  Wire.beginTransmission(MPU);
  Wire.write(0x6B);
  Wire.write(0);
  Wire.endTransmission(true);
}
void get_accelerometer(){ //get data of accelerometer
  ax = 0.00;
  ay = 0.00;
  az = 0.00;

  for (int i = 0; i < 10; i++) {      //to take average of 10 data
    Wire.beginTransmission(MPU);
    Wire.write(0x3B);
    Wire.endTransmission(false);
    Wire.requestFrom(MPU, 12, true);
    AcX = Wire.read() << 8 | Wire.read();
    AcY = Wire.read() << 8 | Wire.read();
    AcZ = Wire.read() << 8 | Wire.read();
    GyX = Wire.read() << 8 | Wire.read();
    GyY = Wire.read() << 8 | Wire.read();
    GyZ = Wire.read() << 8 | Wire.read();
    ax = ax + (AcX / 16384.0);
    ay = ay + (AcY / 16384.0);
    az = az + (AcZ / 16384.0);

  }
  ax = ax / 10;
  ay = ay / 10;
```

```
  az = az / 10;
  Wire.endTransmission(true);
}
void get_gps(){
  while (gpsSerial.available()) { // check for gps data
    if (gps.encode(gpsSerial.read())) // encode gps data
    {
      gps.f_get_position(&lat, &lon); // get latitude and longitude
    }
  }
  latitude = String(lat, 6);
  longitude = String(lon, 6);
}

void get_piezo(){       /piexo data
  p1 = analogRead(piezo1);
  p2 = analogRead(piezo2);
  p3 = analogRead(piezo3);
  p4 = analogRead(piezo4);
  p5 = analogRead(piezo5);
  p6 = analogRead(piezo6);
  p7 = analogRead(piezo7);
  p8 = analogRead(piezo8);
  p9 = analogRead(piezo9);
  p10 = analogRead(piezo10);
}

void get_car(){
  car = digitalRead(car_in);
}
```