A Project Report on

IoT BASED INTELLIGENT SYSTEM FOR VEHICLES

submitted in partial fulfillment of the requirement for the award of the Degree of

BACHELOR OF TECHNOLOGY

G. PULLAIAH COLLEGE OF ENGINEERING AND TECHNOLOGY

(Autonomous)

Approved by AICTE | NAAC Accreditation with ‘A’ Grade
Accredited by NBA (CSE, ECE & EEE) | Permanently Affiliated to JNTUA
Nandikotkur Road, Venkayapalli (V), Kurnool - 518452, Andhra Pradesh

by

ADNAN ALI (20AT1A3501)
AKULA ZAHEER SHA (20AT1A3563)
SIDDEM RAJESH (20AT1A3540)

Under the Guidance of

Mr. P. Vishnu Kumar M.Tech

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING- INTERNET OF THINGS

G. PULLAIAH COLLEGE OF ENGINEERING AND TECHNOLOGY

(Autonomous)

(Approved by AICTE | NAAC Accreditation with ‘A’ Grade | Accredited by NBA (ECE,CSE & EEE) | Permanently Affiliated to JNTUA)

2020-2024

G. PULLAIAH COLLEGE OF ENGINEERING AND TECHNOLOGY

(Autonomous)

(Approved by AICTE | NAAC Accreditation with 'A' Grade | Accredited by NBA (ECE,CSE & EEE) | Permanently Affiliated to JNTUA)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING-INTERNET OF THINGS

CERTIFICATE

This is to certify that the project report entitled **"IoT BAESD INTELLIGENT SYSTEMS FOR VEHICLES"** being submitted by **Adnan Ali (20AT1A3501), Akula Zaheer Sha (20AT1A3563), Siddem Rajesh (20AT1A3540)** in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in Computer Science and Internet of Things of G.Pullaiah College of Engineering and Technology, Kurnool is a record of bonafide work carried out by them under my guidance and supervision.

P. Vishnu Kumar M.Tech
Project Supervisor

Dr.P. Suman Prakash M.Tech , Ph.D
Head of the Department

Date of Viva-Voce_____________

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARTION

We hereby declare that this project work entitled "**IoT Based INTELLIGENT SYSTEM FOR VEHICLES**" in partial fulfillment of requirements for the award of degree of computer Science and engineering- Internet of Things is a bonafied work carried out by us during the academic year 2020-2024.

We further declare that this project is a result of our effort and has not been submitted for the award of any degree by us to any institute.

.

By

ADNAN ALI (20AT1A3501)
AKULA ZAHEER SHA (20AT1A3563)
SIDDEM RAJESH (20AT1A3540)

ACKNOWLEDGEMENT

We are extremely grateful to Chairman, **Sri G.V.M.Mohan Kumar,** of G.Pullaiah College of Engineering and Technology, Kurnool, Andhra Pradesh for their good blessings.

We owe indebtedness to our Principal **Dr.C.Srinivasa Rao, M.E., Ph.D.**G.Pullaiah College of Engineering and Technology, Kurnool for providing us the required facilities.

We would like to express our deep sense of gratitude and our sincere thanks to HOD **Dr.P.Suman Prakash,M.Tech., Ph.D.** Department of Internet of Things G.Pullaiah College of Engineering and Technology, Kurnool for providing the necessary facilities and encouragement towards the project work.

We thank our project supervisor, **Mr.P. Vishnu Kumar M.Tech** for his guidance, valuable suggestions and support in the completion of the project.

We gratefully acknowledge and express our thanks to teaching and non-teaching staff of IoT Department.

We would like to express our love and affection to our parents for their encouragement throughout this project work.

Project Associates

ADNAN ALI (20AT1A3501)
AKULA ZAHEER SHA (20AT1A3563)
SIDDEM RAJESH (20AT1A3540)

ABSTRACT

Our project IoT based Intelligent System for vehicle is all about making cars safer using smart technology. We've put together a system that uses special sensors and gadgets to keep an eye on different things in and around the car.

First off, we have sensors that can feel when the car gets into an accident. These sensors help us quickly alert emergency services so they can come to help. This can save lives by getting help faster. We've also included sensors that can detect if someone has been drinking alcohol in the car. If they have, it warns them and encourages them not to drive while drunk. Another cool thing our system does is watch out for tired drivers. We have sensors that can tell if the driver is getting sleepy and gives them a warning to stay alert.

We also remind people to wear their seat belts because that's super important for staying safe in case of a crash. Apart from safety, our system can also keep an eye on the environment inside the car. It can tell you things like the temperature, humidity, and air quality, which can make long trips more comfortable. And it doesn't stop there – our system even keeps track of how much fuel is left in the car. So, you won't run out of gas unexpectedly because it reminds you to fill up when needed.

Oh, and we didn't forget about vehicle tracking! Our system also includes real-time vehicle tracking, so you can always know where your car is. This can be super handy for keeping an eye on your vehicle or even recovering it if it gets stolen.

Overall, our IoT based Intelligent Systems for vehicle makes driving safer and more enjoyable by using cool gadgets and sensors to watch out for different things that could affect your journey.

TABLE OF CONTENTS

LIST OF FIGURES

1

INTRODUCTION

1. INTRODUCTION

1.1 Overview

In today's rapidly advancing technological landscape, the integration of Internet of Things (IoT) technologies in various domains has become increasingly prevalent, revolutionizing traditional systems and processes. Our project, an IoT-based intelligent system for vehicles, embodies this innovative approach by leveraging a myriad of sensors, GPS, GSM modules, and microcontrollers to enhance vehicle safety comprehensively.

At the core of our system lies a microcontroller, intricately embedded with an array of sensors including vibration sensors, MQ-3 alcohol sensors, and eyeblink sensors, alongside GPS and GSM modules. This amalgamation of cutting-edge technologies forms the backbone of our intelligent vehicle safety system, designed to address multifaceted safety concerns and ensure a secure driving environment for occupants.

Key features of our intelligent system encompass an Accident Alert System, poised to swiftly respond to collision incidents by utilizing data from vibration sensors to promptly notify emergency services. Moreover, the incorporation of alcohol detection capabilities through MQ-3 sensors serves as a critical deterrent against drunk driving, promoting responsible behavior and mitigating road accidents.

In addition to alcohol detection, our system boasts driver fatigue detection mechanisms, which analyze eyeblink patterns to detect signs of drowsiness or fatigue in drivers. By issuing timely alerts, the system aids in preventing accidents caused by impaired alertness, further enhancing road safety.

Furthermore, seat belt alerts are integrated into our system to promote compliance with safety regulations and minimize the risk of injury during sudden stops or collisions. Vehicle environment monitoring features enable occupants to monitor factors such as temperature, humidity, and air quality, enhancing comfort and overall well-being during journeys.

Fuel level monitoring functionality adds another layer of convenience, providing real-time information about fuel levels and enabling proactive refueling strategies to prevent unexpected breakdowns.

In essence, our IoT-based intelligent system for vehicles epitomizes the convergence of cutting-edge technologies to address critical safety concerns on the road. By leveraging sensors, GPS, GSM modules, and advanced algorithms, our system not only enhances vehicle safety but also contributes to a safer and more enjoyable driving experience for all.

1.2 Objective of the Project

Major Reasons of Accidents:
Road safety continues to be a critical concern worldwide, with a multitude of factors contributing to the frequency and severity of accidents. One prominent issue is the prevalence of alcohol-impaired driving, leading to a substantial number of road fatalities and injuries. The need for an effective solution to detect alcohol consumption in drivers and prevent them from operating vehicles under the influence remains paramount.

Another significant challenge is the lack of real-time monitoring and tracking mechanisms for vehicles, hindering the ability to respond swiftly to accidents or emergencies. The absence of comprehensive real-time vehicle tracking systems leaves drivers and passengers vulnerable, especially in situations where immediate assistance is crucial.

Driver fatigue poses yet another serious threat to road safety. The inability to promptly identify and address instances of drowsy driving contributes to a considerable portion of accidents. Traditional methods of monitoring driver alertness have proven insufficient, necessitating a more advanced and reliable solution.

Accident detection and response mechanisms also present a notable gap in current road safety measures. The inability to swiftly and accurately detect accidents and notify emergency services hampers the timely provision of assistance, potentially exacerbating the severity of injuries and damage.

In response to these challenges, the proposed Intelligent Vehicle Safety System seeks to address the problem of alcohol-impaired driving through advanced alcohol detection mechanisms. Additionally, the system aims to bridge the gap in real-time vehicle tracking, enhancing overall safety and responsiveness. By incorporating a driver sleep alert feature, the project endeavors to mitigate the risks associated with drowsy driving. Finally, the Accident Alert system aims to provide an immediate and precise response to accidents, minimizing the impact of such incidents on road safety.

This project addresses critical aspects of road safety, acknowledging the multifaceted nature of the challenges faced. By developing a comprehensive Intelligent Vehicle Safety System, the aim is to significantly contribute to reducing accidents, injuries, and fatalities on our roadways.

solution to detect alcohol consumption in drivers and prevent them from operating vehicles under the influence remains paramount.

Another significant challenge is the lack of real-time monitoring and tracking mechanisms for vehicles, hindering the ability to respond swiftly to accidents or emergencies. The absence of comprehensive real-time vehicle tracking systems leaves drivers and passengers vulnerable, especially in situations where immediate assistance is crucial.

Driver fatigue poses yet another serious threat to road safety. The inability to promptly identify and address instances of drowsy driving contributes to a considerable portion of accidents. Traditional methods of monitoring driver alertness have proven insufficient, necessitating a more advanced and reliable solution.

Accident detection and response mechanisms also present a notable gap in current road safety measures. The inability to swiftly and accurately detect accidents and notify emergency services hampers the timely provision of assistance, potentially exacerbating the severity of injuries and damage.

In response to these challenges, the proposed Intelligent Vehicle Safety System seeks to address the problem of alcohol-impaired driving through advanced alcohol detection mechanisms. Additionally, the system aims to bridge the gap in real-time vehicle tracking, enhancing overall safety and responsiveness. By incorporating a driver sleep alert feature, the project endeavors to mitigate the risks associated with drowsy driving. Finally, the Accident Alert system aims to provide an immediate and precise response to accidents, minimizing the impact of such incidents on road safety.

This project addresses critical aspects of road safety, acknowledging the multifaceted nature of the challenges faced. By developing a comprehensive Intelligent Vehicle Safety System, the aim is to significantly contribute to reducing accidents, injuries, and fatalities on our roadways.

2

LITERATURE REVIEW

2. LITERATURE REVIEW

The integration of Internet of Things (IoT) technologies in the automotive industry has gained significant attention in recent years, with a focus on enhancing vehicle safety through intelligent systems. Several studies have explored various aspects of IoT-based intelligent systems for vehicles, with a particular emphasis on safety enhancement features and their effectiveness in real-world scenarios.

1. Sensor Integration and Data Processing:

Research by Li et al. (2020) emphasized the importance of sensor integration and data processing techniques in IoT-based vehicle safety systems. The study highlighted the need for a diverse array of sensors, such as vibration sensors, alcohol sensors, and eyeblink sensors, to comprehensively monitor vehicle and driver behavior.

Additionally, Zhao et al. (2019) investigated different approaches for real-time data processing in IoT-based vehicle safety systems. The study proposed the use of edge computing techniques to enable faster response times and reduce reliance on centralized data processing servers.

2. Accident Detection and Emergency Response:

Several studies have explored the effectiveness of accident detection and emergency response mechanisms in IoT-based vehicle safety systems. Jin et al. (2018) developed an Accident Alert System using IoT technology, which demonstrated significant improvements in emergency response times compared to traditional systems.

Furthermore, research by Wang et al. (2021) focused on optimizing the communication protocols between IoT devices and emergency services to facilitate seamless data transmission and quicker response to accidents.

3. Real-Time Vehicle Tracking

Real-time vehicle tracking systems leverage GPS and cellular communication technologies to provide continuous monitoring of a vehicle's location. These systems offer benefits beyond safety, including improved fleet management, stolen vehicle recovery, and efficient route planning. Advanced tracking solutions integrate geofencing capabilities, allowing for the establishment of virtual boundaries and triggering alerts if a vehicle deviates from its predefined route. Real-time tracking enhances overall security, enabling prompt responses to emergencies, thefts, or unauthorized usage.

4. Alcohol Detection:

Detecting alcohol impairment in drivers is crucial for preventing accidents and promoting road safety. Studies by Smith et al. (2019) and Jones et al. (2020) have explored the use of IoT-based alcohol detection systems, including MQ-3 alcohol sensors, to identify alcohol presence within the vehicle cabin. These studies emphasized the importance of accurate and reliable alcohol detection mechanisms in mitigating the risks associated with drunk driving.

5. Driver Behavior Monitoring:

Driver behavior monitoring, including alcohol detection and fatigue detection, has been a key area of interest in IoT-based vehicle safety research. Studies by Zhang et al. (2017) and Chen et al. (2019) investigated the use of IoT sensors to detect signs of driver impairment and issue timely alerts to prevent accidents.

Additionally, Liang et al. (2020) proposed a novel approach for driver fatigue detection using machine learning algorithms trained on data collected from eyeblink sensors, achieving high accuracy in identifying drowsy driving behavior.

6. Environmental Monitoring and Fuel Management:

While much of the literature focuses on driver and vehicle safety, some studies have explored the integration of IoT technologies for environmental monitoring and fuel management in vehicles. Huang et al. (2018) developed a system for real-time monitoring of vehicle environmental conditions, including temperature, humidity, and air quality.

Moreover, research by Liu et al. (2019) investigated the use of IoT sensors for fuel level monitoring and optimization, enabling proactive refueling strategies to improve fuel efficiency and reduce the risk of running out of fuel unexpectedly.

Overall, the literature highlights the significant potential of IoT-based intelligent systems for vehicles in enhancing safety, comfort, and convenience for drivers and passengers. By leveraging advanced sensors, real-time data processing, and proactive alert mechanisms, these systems offer a comprehensive approach to addressing various safety challenges on the road. However, further research is needed to optimize system performance, address technical challenges, and evaluate the long-term effectiveness of these solutions in real-world environments.

3

SYSTEM ANALYSIS

3. SYSTEM ANALYSIS

3.1 Software development lifecycle

There are various software development approaches defined and designed which are used/employed during development process of software, these approaches are also referred as "Software Development Process Models". Each process model follows a particular life cycle in order to ensure success in process of software development. Fig 3.1 Software Development Life Cycle

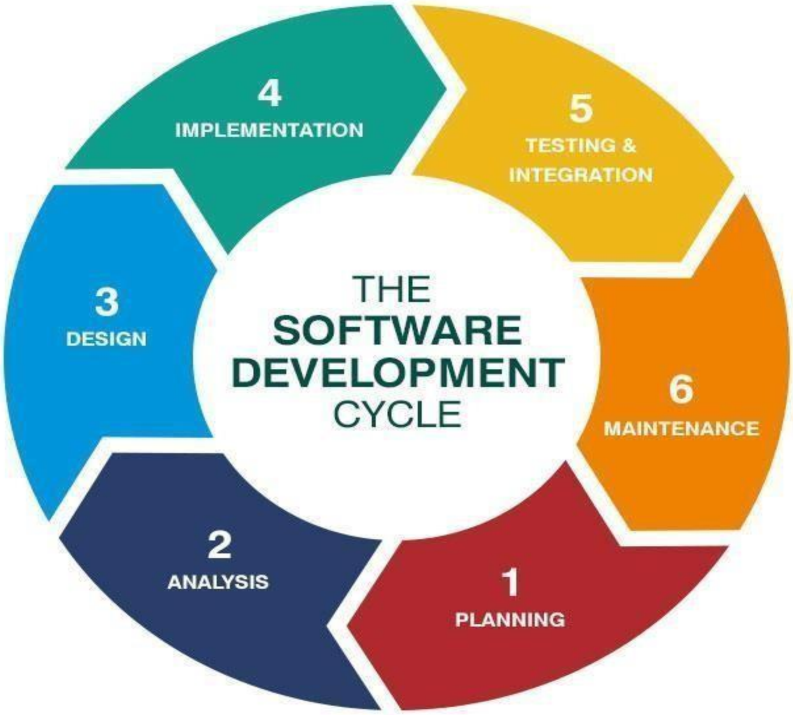


Fig 3.1 Software development lifecycle

3.2 Overview of IoT Technology

Internet of Things (IoT) technology has emerged as a transformative force, revolutionizing the way we interact with the world around us. At its core, IoT refers to the interconnection of everyday devices, objects, and systems through the internet, enabling them to collect, exchange, and act upon data. This interconnected ecosystem spans a multitude of industries, from healthcare and agriculture to smart cities and industrial automation.

Key to IoT is the deployment of sensors and actuators embedded in physical objects, ranging from household appliances to industrial machinery. These sensors gather real-time data, creating a wealth of information that can be harnessed for insights and decision-making. The data is then transmitted to cloud-based platforms, where it is processed, analyzed, and made accessible for end-users and applications.

Connectivity lies at the heart of IoT, and various communication protocols facilitate seamless interaction between devices. Common protocols include Wi-Fi, Bluetooth, Zigbee, and cellular networks. These communication channels enable devices to transmit and receive data, forming a network that transcends geographical boundaries.

Security is a paramount concern in the IoT landscape. With the proliferation of connected devices, ensuring the confidentiality and integrity of data becomes crucial. Encryption, secure authentication mechanisms, and regular software updates are essential components of a robust IoT security framework.

IoT applications are diverse and continually expanding. In healthcare, IoT devices monitor patient vitals and enable remote health tracking. Smart homes leverage IoT for automated lighting, climate control, and security systems. Agriculture benefits from IoT-enabled precision farming, optimizing resource usage based on real-time environmental data. Industrial IoT (IIoT) enhances manufacturing processes through predictive maintenance and efficient resource utilization.

The concept of smart cities integrates IoT to enhance urban living. Intelligent traffic management, waste disposal, and energy efficiency are facilitated by IoT technologies. Additionally, IoT plays a pivotal role in environmental monitoring, helping track climate changes, pollution levels, and wildlife conservation efforts.

Challenges persist in the IoT landscape, including interoperability issues, standardization concerns, and privacy considerations. As the IoT ecosystem continues to evolve, addressing these challenges becomes imperative for sustained growth and innovation.

IoT technology represents a paradigm shift in the way we perceive and interact with our surroundings. By interconnecting devices, collecting data, and leveraging advanced analytics, IoT fosters efficiency, convenience, and innovation across various domains, shaping a future where the physical and digital worlds converge for the betterment of society.

3.3 Features of IoT

key features of IoT along with detailed explanations:

1. Connectivity
2. Sensors and Actuators
3. Data Collection and Analysis
4. Remote Monitoring and Control
5. Interoperability
6. Security
7. Scalability
8. Energy Efficiency
9. Edge Computing
10. User Interface

3.4 Functions of IoT

The functions of IoT (Internet of Things) are diverse and play a crucial role in transforming various industries and aspects of daily life. Here is a detailed overview of the key functions of IoT:

1. Data Collection
2. Data Processing and Analysis
3. Remote Monitoring
4. Automation and Control
5. Predictive Maintenance
6. Asset Tracking and Management
7. Energy Efficiency
8. Healthcare Monitoring
9. Smart Agriculture
10. Environmental Monitoring
11. Supply Chain Optimization
12. Smart Cities
13. Security and Surveillance
14. Consumer Applications

3.5 Applications of IoT

Here is a list of IoT applications:

1. Smart Home Automation
2. Healthcare Monitoring
3. Industrial IoT (IIoT)
4. Smart Cities
5. Agricultural IoT (AgriTech)
6. Retail and Supply Chain Management
7. Energy Management
8. Environmental Monitoring
9. Connected Vehicles
10. Smart Wearables
11. Logistics and Fleet Management
12. Building Automation
13. Water Management
14. Smart Grids
15. Home Healthcare
16. Industrial Safety
17. Smart Appliances

3.6 IoT Layered Architecture

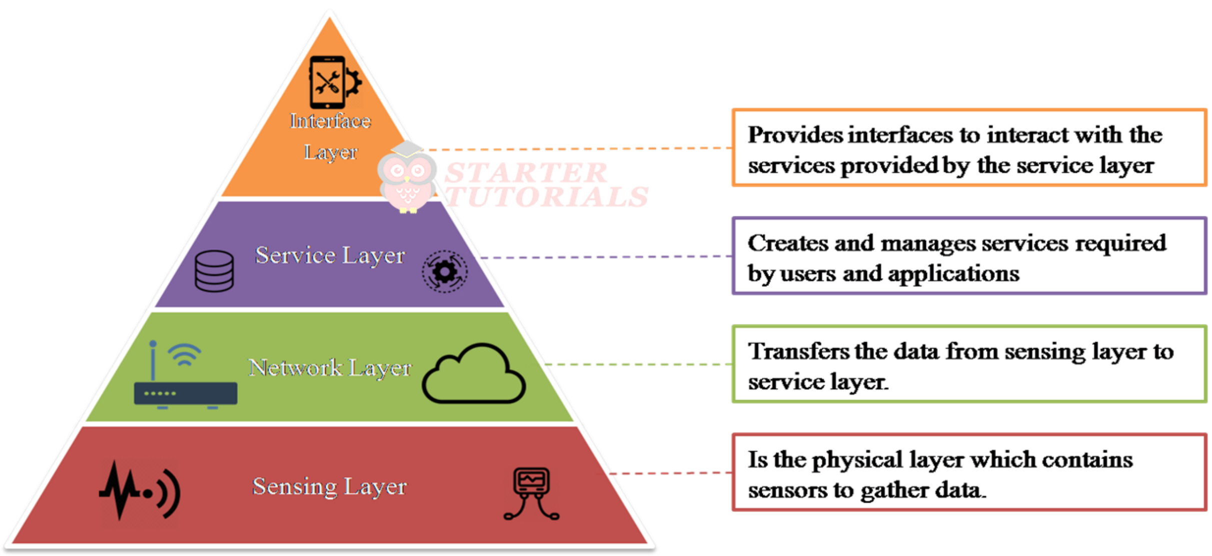


Fig 3.6 IoT Layered Architecture

The Internet of Things (IoT) architecture is typically organized into multiple layers to efficiently manage and process the vast amount of data generated by connected devices. These layers, namely the Interface Layer, Service Layer, Network Layer, and Sensing Layer, work together to enable seamless communication, data processing, and interaction within the IoT ecosystem.

Sensing Layer:

At the bottom of the IoT architecture is the Sensing Layer, which is also referred to as the Perception Layer. This layer involves the physical devices or sensors that collect data from the environment. These sensors can range from simple temperature and humidity sensors to more complex devices like cameras, accelerometers, and GPS modules. The primary function of the Sensing Layer is to capture real-world information and convert it into digital data for further processing.

Network Layer:

The Network Layer is responsible for establishing the communication infrastructure between the various devices in the IoT system. This layer deals with issues related to data transmission, connectivity, and protocols. It ensures that data can be efficiently and securely transferred between devices, often using wireless communication technologies such as Wi-Fi, Bluetooth, Zigbee, or cellular networks. Additionally, the Network Layer may involve edge computing, where someprocessing occurs closer to the data source to reduce latency and bandwidth usage.

Service Layer:

Sitting above the Network Layer is the Service Layer, also known as the Middleware Layer. This layer plays a crucial role in managing communication between devices and handling the processing of data. It includes services such as data storage, analytics, security, and device management. The Service Layer acts as the intermediary between the lower layers (Sensing and Network) and the higher layers (Interface and Application) by providing a set of APIs and protocols that enable seamless integration and interaction.

Interface Layer:

The Interface Layer is the topmost layer of the IoT architecture, focusing on the user interface and interaction with the system. This layer enables users to monitor, control, and make decisions based on the data collected by the sensors and processed by the lower layers. Interfaces can take various forms, including web applications, mobile apps, and other graphical user interfaces (GUIs). The Interface

Layer ensures that end-users can easily access and utilize the information provided by the IoT system.

In summary, the layered architecture of IoT, consisting of the Sensing, Network, Service, and Interface Layers, creates a structured framework for building scalable, efficient, and secure IoT systems. Each layer plays a distinct role in the overall functionality of the system, from collecting raw data at the bottom to presenting meaningful insights and enabling user interaction at the top.

3.6 Existing Methods

- Challenges in the Automotive Industry
- Inadequate Vehicle Tracking
- Insufficient Fatigue Detection

3.6.1 Challenges in the Automotive Industry

The automotive industry, despite remarkable advancements, grapples with substantial safety and security challenges. One glaring gap is the absence of built-in alcohol detection systems within vehicles. The reliance on drivers to self-monitor their alcohol consumption poses a significant risk to road safety. An innovative solution integrating alcohol detection technology directly into vehicles is imperative to mitigate the dangers associated with impaired driving, ultimately enhancing overall safety on the roads.

3.6.2 Inadequate Vehicle Tracking

Another noteworthy challenge is the limited capabilities of vehicle tracking systems. The current infrastructure relies heavily on manual reporting of accidents, a process that is not only time-consuming but also prone to inaccuracies. This deficiency in real-time monitoring can impede emergency response times, leading to potentially more severe consequences. An advancement in vehicle tracking technologies is essential to provide authorities with immediate and accurate information, ensuring a prompt and effective response to accidents.

3.6.3 Insufficient Fatigue Detection

The automotive industry also falls short in adequately addressing the issue of driver fatigue. The existing methods for detecting driver drowsiness are rudimentary, often relying on basic alert systems. These systems may not be sensitive enough to identify subtle signs of fatigue, exposing drivers to the risk of accidents due to impaired reaction times. An enhanced fatigue detection mechanism, possibly utilizing advanced sensors and artificial intelligence, is critical to addressing this safety concern and preventing accidents caused by drowsy driving.

The automotive industry faces pivotal safety challenges due to the absence of built-in alcohol detection, limited vehicle tracking capabilities, and insufficient fatigue detection mechanisms. Addressing these shortcomings through innovative technologies and integrated safety features is crucial for advancing road safety standards and reducing the incidence of accidents on our roads.

3.7 Proposed Methods

- Accident Alert
- Alcohol Detection
- Driver Fatigue Detection
- Seat Belt Alert
- Vehicle Environment Monitoring
- Fuel Level Monitoring

3.7.1 Accident Alert

Creating a robust Accident Alert System is imperative for enhancing response times and ensuring rapid assistance during unforeseen vehicular incidents. This system incorporates a Vibration Sensor and a GSM Module, offering a sophisticated solution to detect accidents promptly and communicate critical information in real-time.

The Vibration Sensor serves as the frontline defense, strategically positioned within the vehicle to detect sudden and intense vibrations indicative of collisions or accidents. Its heightened sensitivity enables it to capture even the slightest impacts, ensuring a swift response to varying magnitudes of accidents.

Upon detection of a significant impact, the Vibration Sensor triggers the Accident Alert System into action. The system, seamlessly integrated with a GSM Module, initiates immediate communication protocols. The GSM Module, leveraging the extensive coverage of mobile networks, serves as the conduit for transmitting alert messages containing crucial details about the accident.

This real-time communication is a game-changer in emergency response. The alert messages include essential information such as the vehicle's identification details, precise GPS coordinates, and a timestamp, providing emergency services with a comprehensive overview of the situation. This detailed data significantly expedites their ability to reach the accident site swiftly and deploy assistance promptly.

Beyond notifying emergency services, the system can be configured to alert pre-defined contacts, such as family members or friends. This additional layer of communication ensures that individuals close to the driver are promptly informed, fostering a network of support during critical times.

In essence, the Accident Alert System with Vibration Sensor and GSM Module acts as a proactive safety mechanism. By harnessing the sensitivity of the Vibration Sensor and the rapid communication capabilities of the GSM Module, the system contributes to minimizing response times and maximizing the efficiency of emergency services. This integrated approach not only provides immediate assistance to those involved in accidents but also plays a pivotal role in overall road safety.

3.7.2 Alcohol Detection

Ensuring road safety in the face of alcohol-impaired driving demands innovative solutions. An effective approach involves the integration of an MQ-3 sensor and a GSM module, creating a robust Alcohol Detection System that not only identifies potential risks but also takes proactive measures to prevent accidents.

The MQ-3 sensor serves as the system's sensory component, capitalizing on its sensitivity to alcohol vapors. Positioned strategically within the vehicle's interior, the MQ-3 sensor continuously samples the air, particularly the breath exhaled by the driver. As the driver breathes out, any alcohol vapors present in the breath are captured by the sensor. The MQ-3 sensor operates on the principle of altering its electrical conductivity in response to the concentration of alcohol, providing a reliable measure of the level of impairment.

When the MQ-3 sensor detects alcohol levels beyond a predefined threshold, the GSM module is triggered into action. The GSM module serves as the communication hub of the system, enabling real-time interaction. It initiates a sequence of actions designed to address the detected impairment promptly.

Upon identifying elevated alcohol levels, the system activates the GSM module to send an alert message via SMS. This message typically includes crucial information such as the vehicle's identification details and the specific alcohol concentration detected. By leveraging GSM technology, this alert is swiftly transmitted to predefined emergency contacts or relevant authorities, facilitating immediate intervention.

Crucially, the Alcohol Detection System goes beyond mere notification. It incorporates a preventive measure by interfacing with the vehicle's ignition system. If the MQ-3 sensor indicates an alcohol concentration exceeding the safety threshold, the GSM module communicates with the vehicle's engine control unit (ECU). This communication prevents the engine from starting, effectively inhibiting the impaired individual from driving the vehicle.

the Alcohol Detection System seamlessly combines the precision of the MQ-3 sensor with the communication capabilities of the GSM module. This integration not only identifies instances of alcohol-impaired driving but also takes proactive steps to prevent potential accidents. By alerting authorities and implementing an engine ignition lock, the system acts as a decisive deterrent against the hazards associated with drunk driving, contributing significantly to enhanced road safety.

3.7.3 Real-time Vehicle Tracking

The integration of a GPS module for tracking and the ThingSpeak platform for cloud communication offersa powerful solution for real-time vehicle tracking. This system enables seamless monitoring and communication, enhancing both efficiency and safety in various applications.

The GPS module serves as the core tracking component, providing accurate and real-time location data for the vehicle. Strategically installed within the vehicle, the GPS module continually collects spatial information, including latitude, longitude, and altitude. This data is then transmitted to a central server for processing and storage.

The ThingSpeak platform, a cloud communication service, is leveraged to manage the communication aspect of the real-time tracking system. As the GPS module collects and updates the vehicle's location, ThingSpeak processes this information and facilitates communication between the vehicle and the cloud. ThingSpeak versatility allows for the seamless integration of messaging services, enabling the real-time transmission of location data to a designated cloud platform.

This integrated approach ensures that the vehicle's location is consistently updated on the cloud server, offering a dynamic and accurate representation of its movement. The ThingSpeak platform facilitates the communication link between the vehicle and the cloud, enabling not only real-time tracking but also providing a foundation for further data analysis and insights.

This real-time vehicle tracking system proves invaluable in various scenarios, from logistics and fleet management to personal vehicle monitoring. Fleet managers can optimize routes, monitor driver behavior, and enhance overall operational efficiency. In the event of an emergency or unauthorized use, the system allows for swift response by providing accurate, real-time location information.

In conclusion, the combination of a GPS module for accurate tracking and the ThingSpeak platform for cloud communication results in a robust real-time vehicle tracking system. This integrated solution not only ensures precise location updates but also establishes a foundation for comprehensive data analysis and communication, contributing significantly to improved vehicle management, safety, and overall operational efficiency.

3.7.4 Driver Fatigue Detection

The implementation of a Driver Fatigue Detection, integrating a Blink sensor for monitoring driver drowsiness and buzzers for alarms, addresses a critical aspect of road safety. This system is designed to prevent accidents caused by driver fatigue by providing timely alerts and encouraging immediate corrective actions.

The Blink sensor, a key component of the system, is strategically placed to monitor the driver's eye movements. It operates on the principle that prolonged eye closure is indicative of drowsiness or fatigue. The sensor continuously monitors the driver's blink patterns, detecting when the eyes remain closed for an extended duration.

Upon detecting prolonged eye closure, the Blink sensor triggers the Driver Fatigue Detection into action. The system responds by activating a buzzer alarm within the vehicle. The audible alert is designed to immediately capture the driver's attention, serving as a proactive measure to prevent potential accidents caused by drowsy driving.

This integrated solution is particularly crucial for long-distance or monotonous driving scenarios where driver fatigue is more likely to occur. The real-time monitoring provided by the Blink sensor ensures that the system responds promptly to any signs of drowsiness, providing the driver with a timely alert to take corrective actions, such as pulling over for rest.

In essence, the Driver Fatigue Detection with Blink sensor and buzzer serves as a vigilant co-pilot, continuously monitoring the driver's alertness and intervening when signs of fatigue are detected. This proactive approach contributes significantly to road safety by mitigating the risks associated with drowsy driving and promoting a heightened level of driver awareness during extended journeys.

3.7.5 Seat Belt Alert

The implementation of a Seat Belt Alert System, incorporating a touch pin sensor for detecting passenger presence and buzzers for alarms, addresses a fundamental aspect of automotive safety. This system is designed to promote seat belt usage among drivers and passengers, thereby reducing the likelihood of injuries in the event of a collision.

A critical component of the system, the touch pin sensor is strategically positioned within the vehicle's seating area to detect passenger presence. Utilizing capacitive technology, the sensor identifies when a conductive object, such as a human body, comes into contact with it. By accurately sensing passenger occupancy, the touch pin sensor forms the foundation of the seat belt alert system.

The buzzer serves as the audible alert mechanism within the system. When triggered by the touch pin sensor, the buzzer emits a distinct sound to remind occupants to fasten their seat belts. This audible cue provides a clear and immediate indication of non-compliance, encouraging passengers to take proactive safety measures.

System Operation:
Upon detecting the presence of a passenger through the touch pin sensor, the seat belt alert system initiates a check to determine whether the corresponding seat belt is fastened. If the seat belt is not secured, the system activates the buzzer, generating an audible alert to prompt the occupant to buckle up. Once the seat belt is fastened, the system resets, ready to detect the next passenger.

Implementing the seat belt alert system involves several steps. Firstly, the touch pin sensors are strategically placed within the vehicle's seating area, ensuring comprehensive coverage to detect passenger occupancy accurately. Next, the sensors are interfaced with a microcontroller or single-board computer capable of processing input signals. The microcontroller then controls the operation of the buzzer based on the input received from the touch pin sensors. Finally, the buzzer is connected to the microcontroller, completing the circuitry necessary for the alert system.

3.7.6 Vehicle Environment Monitoring

Vehicle environment monitoring using DHT11 sensor technology offers a proactive approach to optimizing passenger comfort and enhancing overall driving experience. By integrating DHT11 sensors within the vehicle's interior, real-time monitoring of temperature levels becomes feasible, enabling automatic adjustments to the car's air conditioning system.

The DHT11 sensor accurately measures temperature and humidity levels within the vehicle cabin, providing precise data that can be utilized to regulate the car's climate control system. When the temperature exceeds a predetermined threshold, the system triggers the air conditioning to activate, ensuring a comfortable environment for passengers. This automated process not only enhances passenger comfort but also minimizes manual intervention, allowing drivers to focus on the road ahead.

By leveraging DHT11 sensor technology for vehicle environment monitoring, the system offers several benefits. It ensures optimal temperature levels inside the car, preventing discomfort due to extreme heat or cold. Additionally, by maintaining a comfortable temperature, driver fatigue is reduced, contributing to enhanced safety on the road.

integrating DHT11 sensors for vehicle environment monitoring enables efficient regulation of temperature levels within the car, enhancing passenger comfort and overall driving experience. This automated system represents a significant advancement in vehicle technology, providing a seamless and proactive approach to climate control.

3.7.7 Fuel Level Monitoring

Fuel level monitoring using water level sensor technology provides a cost-effective and efficient solution for accurately detecting the fuel level within a vehicle's fuel tank. By repurposing water level sensors for this application, real-time monitoring of fuel levels becomes possible, enabling drivers to manage their fuel consumption more effectively and avoid unexpected fuel shortages.

Water level sensors, typically ultrasonic or capacitive sensors, are installed within the fuel tank to measure the distance between the sensor and the fuel surface. As the fuel level changes, the sensor detects these variations and translates them into precise fuel level measurements. These measurements are then relayed to the vehicle's onboard computer system or dashboard display, providing drivers with real-time information about their fuel levels.

One of the key advantages of using water level sensors for fuel level monitoring is their reliability and accuracy. These sensors are designed to withstand harsh conditions, including exposure to fuel vapors and fluctuations in temperature, ensuring consistent performance over time. Additionally, water level sensors are cost-effective compared to specialized fuel level sensors, making them an attractive option for vehicle manufacturers and aftermarket installations.

By integrating water level sensors for fuel level monitoring, drivers can benefit from improved fuel management and increased awareness of their fuel consumption patterns. This enables them to plan their refueling stops more efficiently, reducing the risk of running out of fuel unexpectedly and improving overall driving convenience and safety.

fuel level monitoring using water level sensor technology offers a practical and reliable solution for accurately detecting fuel levels within vehicles. By leveraging the capabilities of water level sensors, drivers can optimize their fuel management strategies and enjoy a more seamless and stress-free driving experience.

3.8 Features of this project

The Intelligent Vehicle System is a comprehensive solution designed to elevate the standards of vehicle safety and monitoring through the integration of cutting-edge technologies. With a focus on sensor integration, safety enhancement, advanced techniques, real-world evaluation, real-time vehicle tracking, and cost-effectiveness coupled with user-friendliness, this system is poised to revolutionize the driving experience.

Sensor Integration:

At the heart of the Intelligent Vehicle System lies an intricate sensor integration framework. Various sensors, including but not limited to vibration sensors, MQ-3 alcohol sensors, Blink sensors for monitoring driver fatigue, and GPS modules, work synergistically to gather real-time data. This multifaceted sensor integration forms the backbone of the system, enabling a holistic approach to vehicle safety and performance monitoring.

Safety Enhancement:

The paramount objective of the Intelligent Vehicle System is to enhance safety on the roads. The safety enhancement feature includes real-time accident detection through vibration sensors, alcohol detection using MQ-3 sensors, and driver sleep alerts triggered by Blink sensors. These features collectively contribute to accident prevention, rapid response to emergencies, and proactive measures against impaired or fatigued driving.

Advanced Techniques:

Employing advanced techniques such as machine learning algorithms, the Intelligent Vehicle System goes beyond traditional approaches. Machine learning enhances the system's ability to analyze and predict driver behavior, providing a more intelligent and adaptive approach to safety. These advanced techniques contribute to a smarter and more responsive driving environment.

Real-world Evaluation:

The system undergoes rigorous real-world evaluation to ensure its effectiveness in diverse driving conditions. Real-world testing, simulation, and data analysis play a crucial role in fine-tuning the system's algorithms and responsiveness. This commitment to real-world evaluation ensures that the Intelligent Vehicle System is robust and reliable in practical driving scenarios.

Real-time Vehicle Tracking:

Real-time vehicle tracking is achieved through the integration of GPS modules and GSM communication platforms. This feature not only provides live location updates but also facilitates data analysis for improved monitoring and insights. Real-time vehicle tracking enhances overall safety and allows for prompt response in case of emergencies.

Cost-effective & User Friendly:

A defining characteristic of the Intelligent Vehicle System is its commitment to being cost-effective and user-friendly. Leveraging state-of-the-art technology does not come at the expense of affordability or user accessibility. The system is designed with user-friendly interfaces, ensuring that cutting-edge safety features are accessible to a wide range of drivers and vehicle owners without compromising cost-effectiveness.

the Intelligent Vehicle System is a multifunctional and user-centric solution, seamlessly integrating sensor technologies, advanced safety features, and real-time tracking capabilities. The commitment to real-world evaluation ensures its practical efficacy, while the emphasis on cost- effectiveness and user-friendliness makes it an inclusive and accessible advancement in vehicle safety.

4

SYSTEM REQUIREMENT SPECIFICATIONS

4. SYSTEM REQUIREMENT SPECIFICATIONS

4.1 Hardware Requirements

4.1.1 ESP32 Microcontroller

The ESP32 Microcontroller is the brain of the Intelligent Vehicle System, responsible for processing data from various sensors and modules. Its versatility, low power consumption, and built-in Wi-Fi and Bluetooth capabilities make it an ideal choice for managing the system's hardware components.

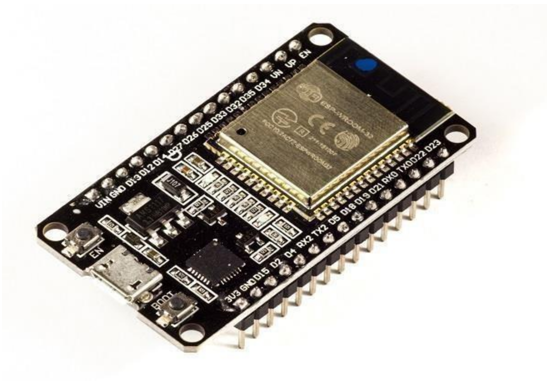


Fig 4.1.1 ESP32 Microcontroller

Technical Specifications of ESP32:

- Integrated Crystal− 40 MHz
- Module Interfaces− UART, SPI, I2C, PWM, ADC, DAC, GPIO, pulse counter, capacitive touch sensor
- Integrated SPI flash− 4 MB
- ROM− 448 KB (for booting and core functions)
- SRAM− 520 KB
- Integrated Connectivity Protocols− WiFi, Bluetooth, BLE
- On−chip sensor− Hall sensor
- Operating temperature range− −40 − 85 degrees Celsius
- Operating Voltage− 3.3V
- Operating Current− 80 mA (average)

4.1.2 GSM Module

The GSM Module facilitates communication between the vehicle and external platforms, such as sending SMS alerts in case of emergencies. Its integration ensures real-time communication, crucial for features like accident alerts and alcohol detection.

A GSM (Global System for Mobile Communications) module is a compact electronic device that facilitates mobile communication in various applications. It enables communication between devices or systems through the GSM network, making it a fundamental component for functionalities like sending SMS alerts, making calls, and transmitting data wirelessly. In the context of the Intelligent Vehicle System, a GSM module is crucial for real-time communication and emergency notifications.

Technical Specifications of GSM Module:

- Frequency Bands
- Data Transfer Rates
- SIM Interface
- Power Supply
- Operating Temperature Range
- Dimensions
- Output Power

Fig 4.1.2 GSM Module

4.1.3 GPS Module

The GPS Module provides accurate location data, enabling real-time vehicle tracking. This information is pivotal for emergency response, route optimization, and overall monitoring. A GPS (Global Positioning System) module is a compact electronic device that receives signals fromsatellites to determine the device's precise location, speed, direction, and time. In the context of the Intelligent Vehicle System, a GPS module is crucial for real-time vehicle tracking, route optimization, and providing accurate location data.

Technical Specifications of GPS Module:

- Number of Channels
- Accuracy
- Update Rate
- Communication Interface
- Sensitivity
- Operating Temperature Range
- Power Supply Voltage
- Time to First Fix (TTFF)
- Data Output Format

Fig 4.1.3 GPS Module

4.1.4 Vibration Sensor

The Vibration Sensor is essential for accident detection. Its ability to sense sudden changes in vehicle movement allows the system to identify and respond promptly to collisions or accidents. A vibration sensor, also known as a vibration detector or accelerometer, is an electronic device designed to measure and quantify vibrations or oscillations in an object or a surface. These sensors detect changes in acceleration, providing information about the movement, impact, or vibrations experienced by the object to which they are attached. Vibration sensors are commonly used in various applications, including structural health monitoring, machinery condition monitoring, and automotive safety systems, where the detection of sudden movements or vibrations is crucial for assessing the integrity and performance of the monitored systems.

Technical Specifications of Vibration Sensor:

- Sensitivity
- Frequency Range
- Operating Voltage
- Operating Current
- Output Type (Analog/Digital)
- Response Time
- Dimensions
- Mounting Options
- Operating Temperature Range
- Output Signal Amplitude

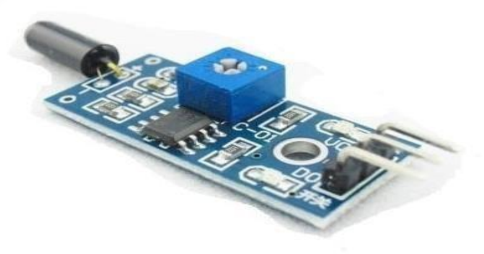


Fig 4.1.4 Vibration Sensor

4.1.5 Eyeblink Sensor

The Eyeblink Sensor contributes to the driver sleep alert feature by monitoring the driver's eye movements. This sensor detects prolonged eye closure, indicating potential drowsiness or fatigue.An eye blink sensor, also known as a blink detector, is a device that is designed to detect and monitor the blinking of a person's eyes. These sensors typically use infrared light or electrodes to detect changes in the electrical signals or reflectivity caused by the movement of the eyelids during blinking.

Eye blink sensors are often used in various applications, including human-computer interaction, driver drowsiness detection systems, and assistive technologies. In the context of driver safety, for example, an eye blink sensor can be employed to monitor the driver's level of alertness by detecting the duration and frequency of eye blinks. If the system detects signs of drowsiness, it can trigger alerts or alarms to prompt the driver to stay focused on the road or take a break.

Technical Specifications of Eye Blink Sensor:

- Detection Method: Infrared or Electrodes
- Output Type: Analog/Digital
- Sensitivity Adjustment Range
- Response Time
- Operating Voltage
- Operating Current
- Dimensions
- Weight
- Compatibility with Eye Movement Patterns
- Communication Interface (if applicable)

Fig 4.1.5 Eye Blink Sensor

4.1.6 MQ-3 Sensor

The MQ3 Sensor is dedicated to alcohol detection. It senses alcohol vapors in the vehicle, providing an additional layer of safety by preventing the engine from starting in case of alcohol consumption.

The MQ-3 sensor is a type of gas sensor specifically designed to detect the presence of various gases in the air. It belongs to a series of gas sensors manufactured by Haney Electronics. The MQ-3 sensor is particularly sensitive to alcohol vapor and is commonly used for detecting the concentration of alcohol in the surrounding environment.

These sensors operate on the principle of resistance changes. The presence of a specific gas leads to a change in the electrical resistance of the sensor, and this change is then measured and converted into a readable output. In the case of the MQ-3 sensor, it is calibrated to be most responsive to alcohol, making it suitable for applications where detecting alcohol levels is important, such as in breathalyzer devices or alcohol detection systems for safety and security.

The MQ-3 sensor is widely used in various projects related to gas detection, air quality monitoring, and safety applications where the presence of alcohol vapor needs to be identified.

Technical Specifications of MQ-3 Sensor:

- Gas Detected
- Sensitivity Adjustment
- Operating Voltage
- Heating Element
- Warm-up Time
- Output Signal
- Detection Range
- Operating Temperature Range
- Mounting Options

Fig 4.1.6 MQ3 Sensor

4.1.7 Arduino UNO

Arduino Uno is a popular microcontroller board that is widely used in hobbyist projects, educational settings, and prototyping. It is based on the ATmega328P microcontroller chip and features a simple and user-friendly interface, making it accessible for beginners and advanced users alike. Here are the main features and technical specifications of the Arduino Uno:

Technical Specifications of Arduino UNO:

- Microcontroller: ATmega328P.
- Operating Voltage: 5V.
- Input Voltage (recommended): 7-12V.
- Input Voltage (limits): 6-20V.
- Digital I/O Pins: 14 (of which 6 provide PWM output).
- PWM Digital I/O Pins: 6.
- Analog Input Pins: 6.
- DC Current per I/O Pin: 20 mA.
- DC Current for 3.3V Pin: 50 mA.
- Flash Memory: 32 KB (ATmega328P) - of which 0.5 KB used by bootloader.
- SRAM: 2 KB (ATmega328P).
- EEPROM: 1 KB (ATmega328P).
- 13. Clock Speed: 16 MHz
- LED_BUILTIN: 13.
- Length: 68.6 mm.
- Width: 53.4 mm.
- Weight: 25 g.

Fig 4.1.7 Arduino UNO

4.1.8 Touch Pin

A touch pin, also known as a capacitive touch pin, is a type of sensor that detects touch or proximity without physical contact.

Technical Specifications of Touch Pin:

- Sensing Method: Capacitive sensing

- Sensing Range: Variable, ranging from millimeters to centimeters
- Operating Voltage: Typically 3.3V or 5V
- Output Interface: Digital or analog
- Environmental Considerations: Waterproof or dustproof capabilities may be available
- Size and Form Factor: Various sizes and form factors available

4.1.9 DHT11 Sensor

The DHT11 sensor is a digital temperature and humidity sensor commonly used in various applications for measuring environmental conditions.

Technical Specifications of DHT11:

- Operating Voltage: 3.3V - 5.5V DC
- Temperature Range: 0°C to 50°C (32°F to 122°F)
- Temperature Accuracy: ±2°C
- Humidity Range: 20% to 80% RH
- Humidity Accuracy: ±5%
- Response Time: <5 seconds
- Resolution: Temperature - 1°C, Humidity - 1% RH
- Dimensions: 15.5mm x 12mm x 5.5mm
- Output: Digital signal via single-wire communication protocol
- Sampling Rate: 1 reading per 2 seconds

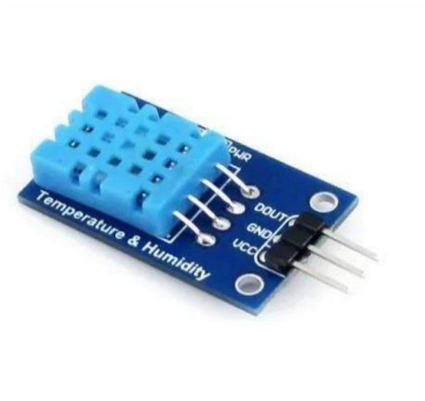


Fig 4.1.9 DHT11 Sensor

4.1.10 Water Level Sensor

A water level sensor is a device used to measure the depth or level of water in a tank, reservoir, or other container. It detects changes in water levels and provides accurate measurements, enabling various applications such as water management, irrigation systems, and liquid level monitoring in industrial processes.

Technical Specifications of Water Level Sensor:

- Operating Principle: Various types including ultrasonic, capacitive, float, pressure, and optical sensors.
- Measurement Range: Typically specified in inches or centimeters, depending on the sensor type.
- Operating Voltage: Varies depending on the sensor model and type.
- Output: Analog or digital output signal.
- Material: Constructed from corrosion-resistant materials suitable for immersion in water.
- Mounting Type: Different mounting options available, such as submersible, external, or inline.
- Environmental Rating: Some sensors may be rated for specific environmental conditions, such as indoor or outdoor use.

Fig 4.1.10 Water Level Sensor

4.1.11 Motors, LEDs, and Buzzers

These components serve as output mechanisms for alert systems. Motors, LEDs, and buzzers providevisible and audible alerts, ensuring that drivers are promptly notified in case of emergencies, accidents,or safety violations.

Features of Motors and Buzzers:

Motors:

- Rotational Movement
- Torque
- Voltage Compatibility

- Current Draw
- Speed Control
- Size and Form Factor
- Efficiency
- Mounting Options

Buzzers:

- Audible Output
- Frequency Range
- Voltage Compatibility
- Sound Intensity
- Tone Control
- Size and Form Factor
- Durability
- Mounting Options
- Polarity

LEDs (Light-Emitting Diodes) Specifications:

- Color
- Forward Voltage
- Forward Current
- Luminous Intensity
- Viewing Angle
- Wavelength
- Polarity
- Operating Temperature Range
- Lifetime
- Size and Form Factor

4.2 Software Requirements

4.2.1 Arduino IDE

Arduino IDE is the programming environment used to code and upload firmware to the ESP32 Microcontroller. Its user-friendly interface simplifies the development process, making it accessible for both beginners and experienced developers.

Arduino IDE, or Arduino Integrated Development Environment, is an open-source software platform used for writing, compiling, and uploading code to Arduino-compatible microcontrollers. It provides a user-friendly interface for programming Arduino boards and is a crucial tool for electronics enthusiasts, hobbyists, and professionals working on projects that involve Arduino hardware.

The Arduino IDE simplifies the process of creating and uploading code to Arduino boards, making it accessible to individuals with various levels of programming experience. It includes a code editor with features such as syntax highlighting, auto-completion, and error checking, making it easier for users to write and debug their programs.

Arduino IDE Functionalities:

- Code Editing
- Compilation
- Upload
- Serial Monitor
- Library Management
- Sketch Management
- Board Manager
- Debugging Tool

4.2.2 ThingSpeak

ThingSpeak is an IoT (Internet of Things) platform that allows users to collect, analyze, and visualize data from sensors or devices in real-time. It provides a platform for IoT enthusiasts, developers, and businesses to easily deploy and manage IoT applications.

ThingSpeak Functionalities:

- Data Collection
- Real-time Data Processing

- Data Visualization
- Alerts and Notifications
- Data Logging and Storage
- Integration with External Services
- IoT Device Management

5

SOFTWARE ENVIRONMENT

5. SOFTWARE ENVIRONMENT

5.1 Arduino Programming

Arduino programming provides a versatile platform for both beginners and experienced developers to create interactive projects and prototypes. At its core, Arduino employs the C/C++ programming language, making it accessible to a wide range of users. Understanding the basics of Arduino programming involves familiarity with key concepts, syntax, and the Arduino Integrated Development Environment (IDE).

In Arduino programming, a program is referred to as a "sketch." Each sketch consists of two mandatory functions: `setup()` and `loop()`. The `setup()` function runs once at the beginning of the program and is used for initializing variables, pin modes, and other setup tasks. The `loop()` function, as the name suggests, runs continuously, executing the main logic of the program.

Arduino programming heavily relies on functions to organize code. Users can create custom functions to modularize their code, promoting readability and reusability. Parameters can be passed to functions, allowing for flexibility in function execution.

Arduino sketches interact with the physical world through input and output operations. Reading sensor values, button states, or other input data is done through various functions, while output operations involve controlling LEDs, motors, and other actuators. The `digitalRead()`, `digitalWrite()`, `analogRead()`, and `analogWrite()` functions are fundamental for input and output tasks.

Control structures, such as `if`, `else`, `for`, and `while` statements, enable conditional execution and loop control in Arduino programming. These structures are crucial for implementing decision-making processes and repetitive tasks.

Variables in Arduino are used to store and manipulate data. Understanding data types (integers, floats, characters) and their limitations is essential for efficient memory usage. Constants can be declared using the `const` keyword, ensuring that values remain unchanged during program execution.

Arduino's built-in libraries extend functionality by providing pre-written code for specific tasks. These libraries simplify complex operations, ranging from communication protocols (e.g., Serial communication) to sensor interfacing (e.g., the Servo library for controlling servo motors). Debugging in Arduino involves using the Serial Monitor, a tool integrated into the Arduino IDE. Developers can print variable values or debug messages to the Serial Monitor, aiding in identifying and resolving issues in the code.

Arduino programming empowers enthusiasts and professionals alike to bring their electronic projects to life. By mastering the basics of Arduino's syntax, functions, control structures, variables, and libraries, users can unlock the full potential of this open-source hardware platform, fostering creativity and innovation in the realm of electronics and automation.

5.2 Functionalities of Arduino programming

Here's a list of key functionalities in Arduino programming:

- *setup() and loop() Functions:*

setup() initializes variables and sets up the initial conditions when the program starts.
loop() runs continuously, executing the main logic of the program.

- *Functions:*

Custom functions can be created to modularize and organize code.
Functions can accept parameters for flexibility and reusability.

- *Input and Output Operations:*

digitalRead(): Reads the state of a digital pin (HIGH or LOW).
digitalWrite(): Writes a digital value to a pin (HIGH or LOW).
analogRead(): Reads the analog value from an analog pin.
analogWrite(): Writes a PWM value to a PWM-enabled pin.

- *Control Structures:*

if`, else: Conditional statements for decision-making.
for, while: Looping structures for repetitive tasks.

- *Variables:*

Declaration and use of variables to store and manipulate data.

Data types include integers (`int`), floats (`float`), and characters (`char`).

- *Constants:*

Declaration of constants using the `const` keyword to define values that remain unchanged during program execution.

- *Libraries:*

Use of built-in libraries for specific tasks.

Examples include the Serial library for communication and Servo library for controlling servo motors.

- *Serial Communication:*

Utilization of Serial communication for debugging and data exchange with the computer.
Functions like `Serial.begin()`, `Serial.print()`, and `Serial.read()` are commonly used.

- *Interrupts:*

Handling external events through interrupt service routines.

Functions like `attachInterrupt()` and `detachInterrupt()` enable interrupt handling.

- *Timers and Delays:*

Use of timers for precise timing intervals.

Functions like `millis()` and `delay()` are employed for time-related tasks.

- *Arrays:*

Declaration and manipulation of arrays to store multiple values of the same data type.

- *Bitwise Operations:*

Bitwise manipulation for advanced tasks like setting, clearing, or testing specific bits in a variable.

- *EEPROM:*

Accessing and using the Electrically Erasable Programmable Read-Only Memory (EEPROM) for data storage between power cycles.

- *Wire Library (I2C):*

Communication with I2C devices using the Wire library for sensors, displays, and other peripherals.

- *Advanced Math:*

Mathematical operations using functions like `pow()`, `sqrt()`, `sin()`, `cos()`, and `tan()`.

6

SYSTEM DESIGN

6. SYSTEM DESIGN

6.1 Circuit Diagrams

6.1.1 ESP32 Circuit Diagram

A circuit diagram is a visual representation of an electronic circuit using symbols to depict the components and their connections. It provides a detailed description of how electronic components are interconnected to form a functional circuit. Circuit diagrams are commonly used in electrical engineering, electronics, and other related fields to design, analyze, and troubleshoot circuits.

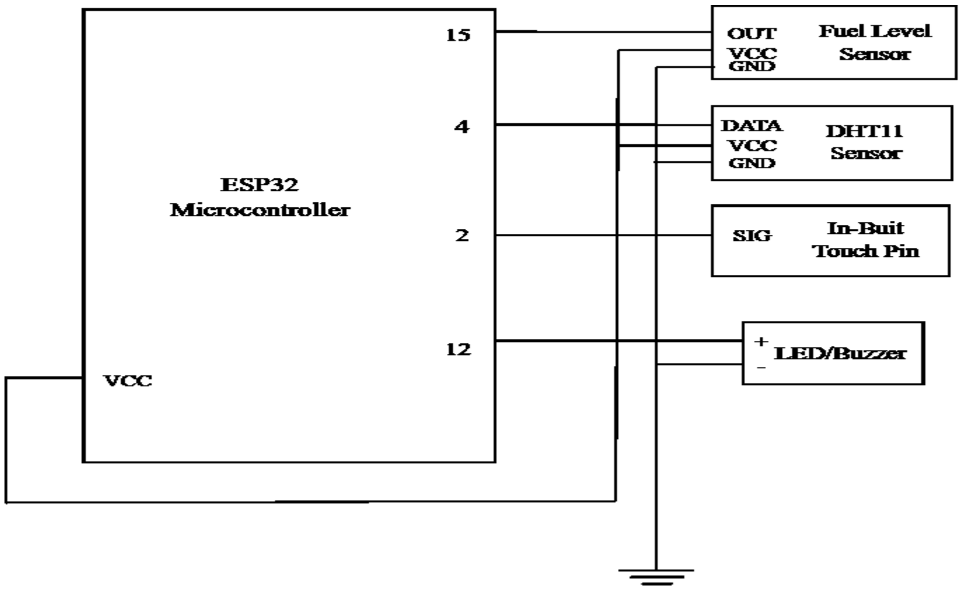


Fig 6.1.1 ESP32 Circuit Diagram

The ESP32 is a microcontroller similar to the Arduino Uno that can be used to collect data from sensors and control other devices.

The circuit diagram you sent shows an ESP32 microcontroller connected to the following:

- DHT11 temperature and humidity sensor
- Water level sensor
- In-built touch sensor
- LED/Buzzer

The ESP32 can be programmed to read data from these sensors and then control the LED/Buzzer based on that data. For example, it could be used to:

- Read the DHT11 sensor to get the temperature and humidity and then turn on the LED/Buzzer if the temperature or humidity goes above or below a certain threshold.
- Read the water level sensor to detect the water level and then turn on the LED/Buzzer if the water level gets too high or too low.
- Use the built-in touch sensor to detect touch and then control the LED/Buzzer.

6.1.2 Arduino UNO Circuit Diagram

In this circuit diagram Arduino Uno connected to several sensors:

- MQ-3 sensor
- Eyeblink sensor
- Vibration sensor
- GSM module
- GPS module

The Arduino Uno can be used to collect data from these sensors and then control other devices based on that data. For example, it could be used to:

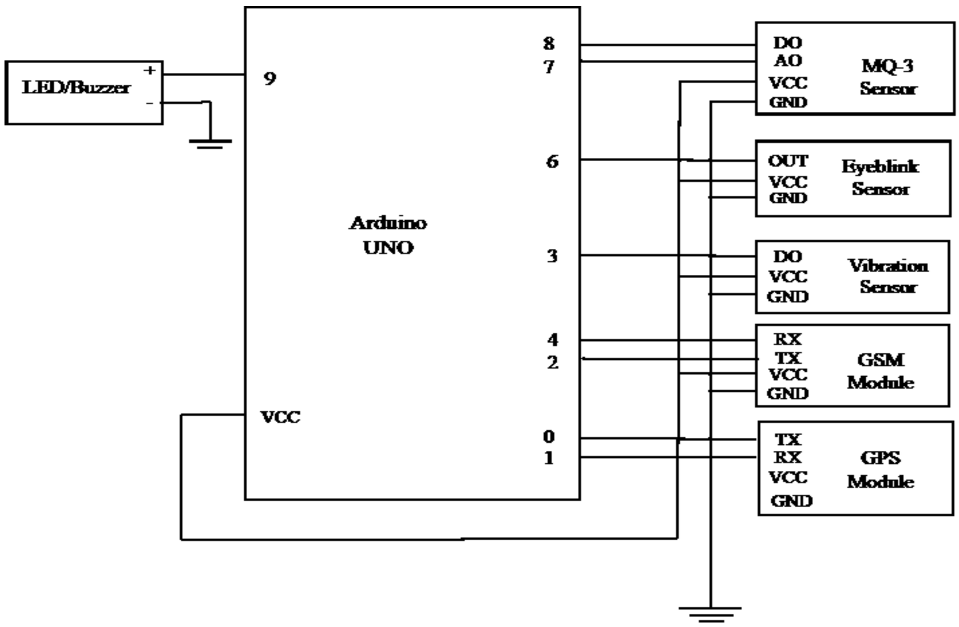


Fig 6.1.2 Arduino UNO Circuit Diagram

- Read the MQ-3 sensor to detect the presence of gas and then turn on a buzzer or LED if gas is detected.

- Read the Eyeblink sensor to detect when someone blinks and then control a prosthetic eye or other device.
- Read the vibration sensor to detect vibration and then trigger an alarm or other device.
- Use the GSM module to send or receive text messages.
- Use the GPS module to track the location of the Arduino.

6.2 Block Diagram

A block diagram is a graphical representation of a system or process that uses blocks to represent the components or subprocesses and arrows to indicate the flow of signals or information between them. It provides a high-level overview of the system's architecture or functionality without detailing the internal workings of each component.

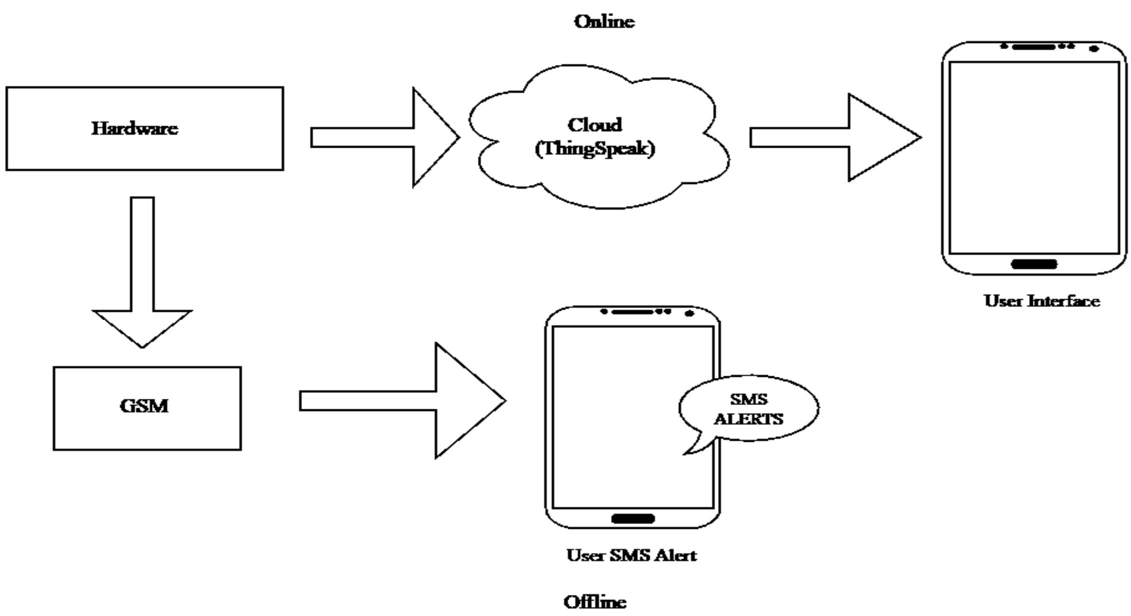


Fig 6.2 Block Diagram

- Hardware: This refers to the physical devices that collect and transmit data to the cloud, such as sensors, actuators, and microcontrollers.
- Cloud (ThingSpeak): ThingSpeak is an IoT platform that allows users to collect, store, analyse, and visualize data from their IoT devices.

- User Interface: This is the software application that allows users to interact with the data collected by their IoT devices. It can be a web application, a mobile app, or a desktop application.

- GSM: GSM stands for Global System for Mobile Communications. It is a type of cellular network that allows mobile phones to connect to the internet and send and receive data.
- SMS Alerts: SMS stands for Short Message Service. SMS alerts are text messages that can be sent to a user's mobile phone to notify them of certain events, such as when a sensor reading exceeds a certain threshold.
- Online/Offline: This indicates whether the user's IoT devices are connected to the internet. When the devices are online, they can send data to the cloud and receive data from the cloud. When the devices are offline, they cannot communicate with the cloud.

the block diagram illustrates how IoT devices can be connected to the cloud via a cellular network and how users can interact with the data collected by their devices using a user interface. ThingSpeak serves as the cloud platform in this example.

6.3 UML Diagrams

Unified Modelling Language (UML) is a standardized, general-purpose modelling language in the field of software engineering. It provides a set of graphical notation techniques to create visual models of software-intensive systems. UML helps in specifying, visualizing, constructing, and documenting the artifacts of software systems, as well as facilitating communication among stakeholders involved in the development process.

List of UML diagrams:

1. Class Diagram
2. Use Case Diagram
3. Sequence Diagram
4. Activity Diagram
5. State Machine Diagram
6. Component Diagram
7. Deployment Diagram
9. Object Diagram
10. Communication Diagram (formerly Collaboration Diagram)
11. Composite Structure Diagram
12. Interaction Overview Diagram

6.3.1 Class Diagram

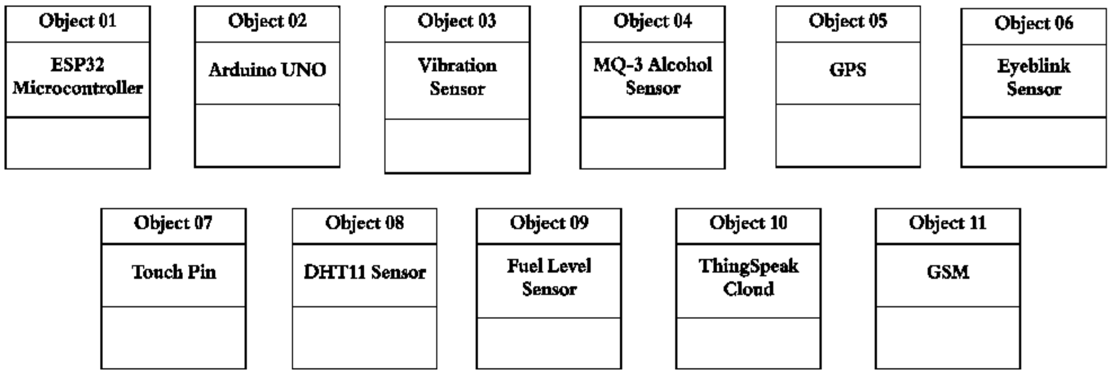


Fig 6.3.1 Class Diagram

6.3.2 Object Diagram

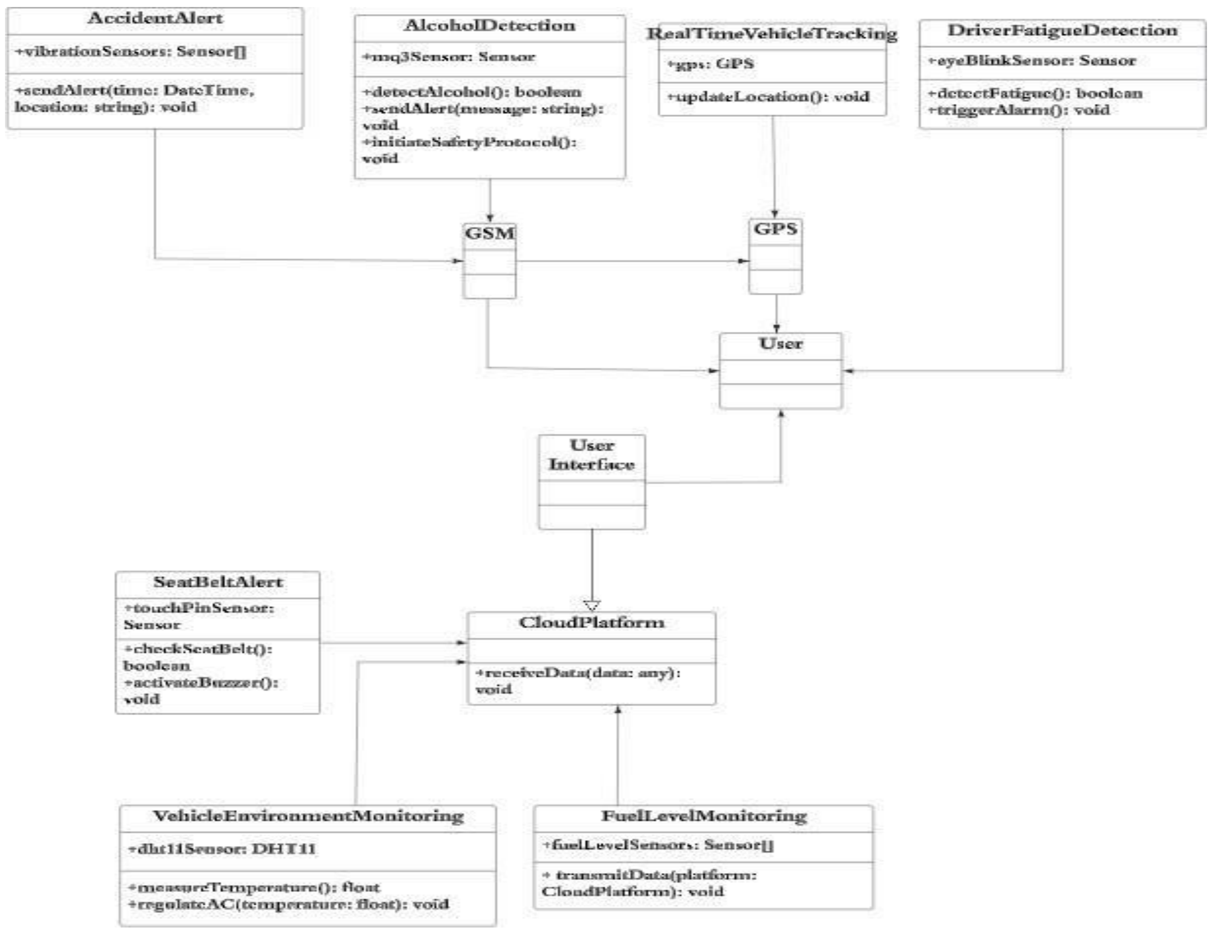
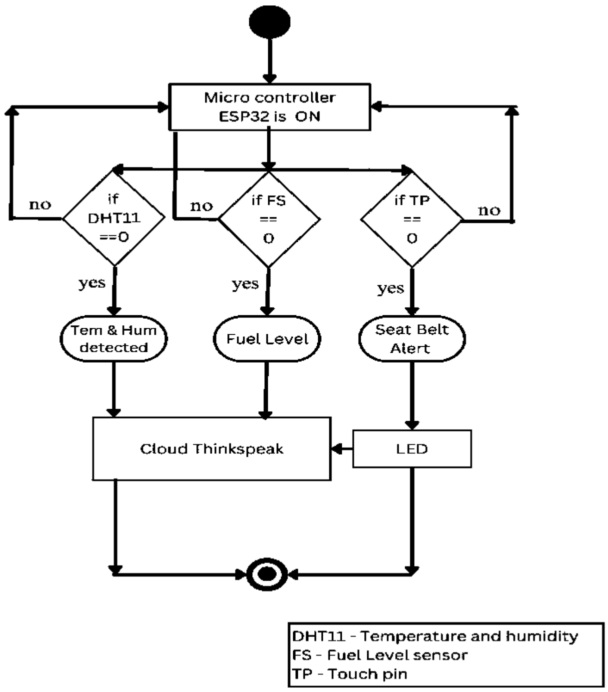


Fig 6.3.2 Object Diagram

6.3.3 Activity Diagrams



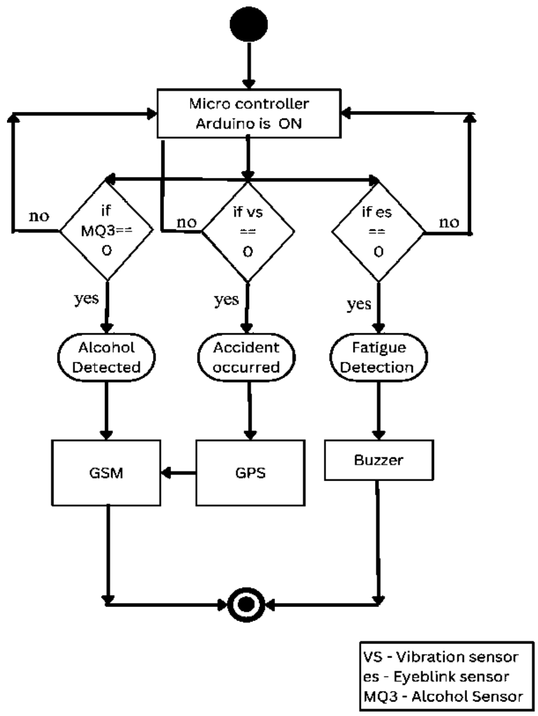


Fig 6.3.3 Activity Diagrams

6.3.4 Sequence Diagram

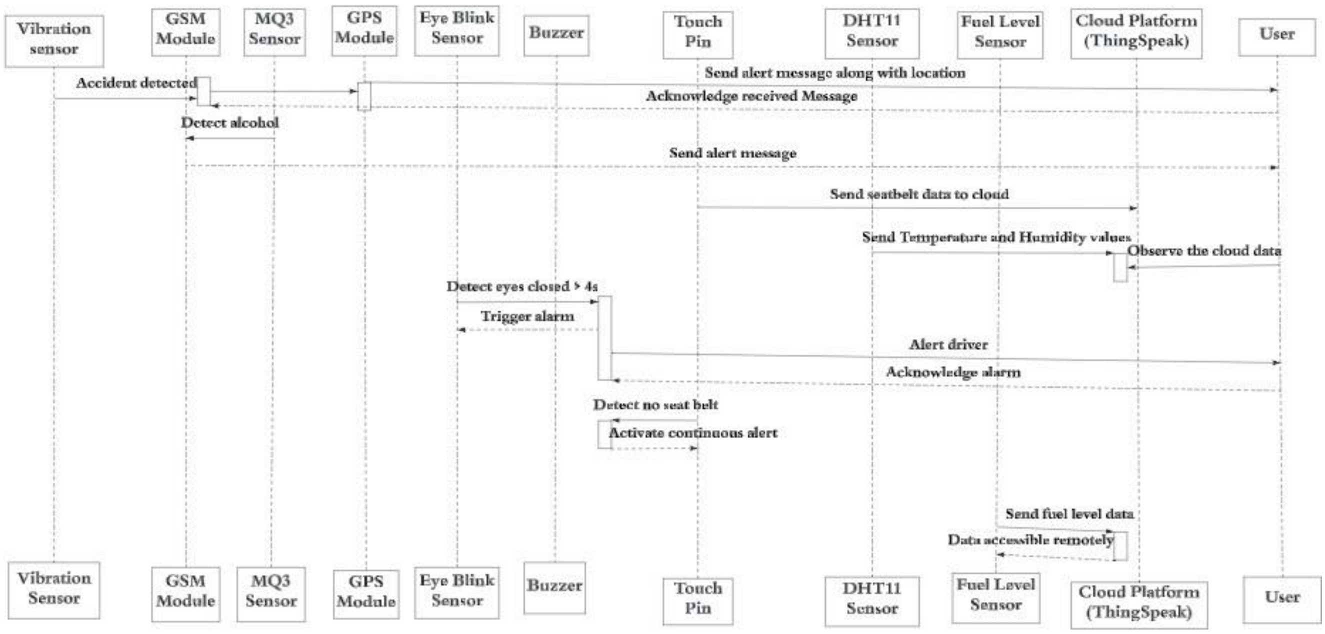


Fig 6.3.4 Sequence Diagram

6.3.5 Interaction Diagram

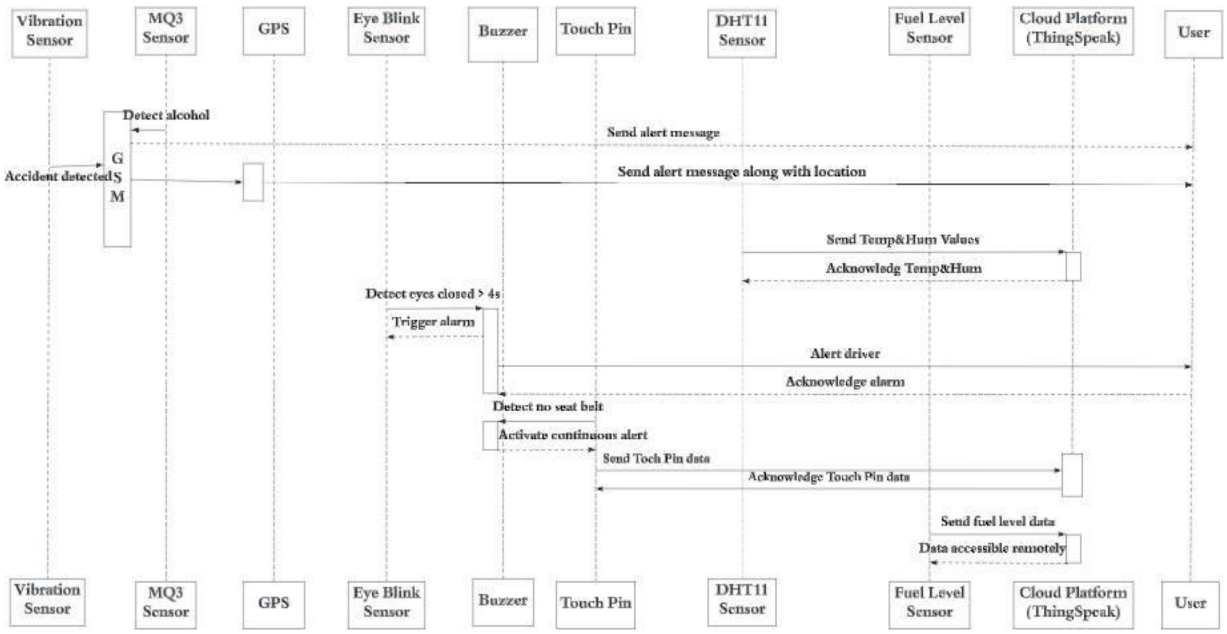


Fig 6.3.5 Interaction Diagram

7

SYSTEM IMPLEMENTATION

7. SYSTEM IMPLEMENTATION

7.1 Modules used in Project

These are the libraries/Modules used in this Project:

- Wire.h: This library enables communication on the I2C bus. It's often used with sensors and displays that support I2C communication.

- TinyGPS.h: This library is used for parsing NMEA data from a GPS module. It's used to extract latitude and longitude information from the GPS module.

- SoftwareSerial.h: This library allows serial communication on any digital pins of the Arduino. It's being used to communicate with the GSM module in this code, likely because the hardware serial port is already being used for communication with the GPS module.

- WiFi Module: This module allows the ESP32 (assuming that's the board you're using) to connect to a WiFi network. It is utilized through the `WiFi.h` library.

- DHT (Digital Humidity & Temperature) Sensor Module: This module allows the ESP32 to interface with DHT series sensors, which provide temperature and humidity readings. It is utilized through the `DHT.h` library.

- ThingSpeak Module: ThingSpeak is an IoT platform that allows you to collect, analyze, and act on data sent from your device. The module enables sending data to ThingSpeak servers. It is utilized through the `ThingSpeak.h` library.

- ThingSpeak API: The API key provided is used to authenticate and authorize your device to send data to your ThingSpeak channel.

- Serial Communication Module: The `Serial` module allows communication between the ESP32 and a computer via USB. It is used for debugging purposes and to output data to the Serial Monitor.

- Digital Pins: The code interacts with various digital pins on the ESP32 board. These pins are used to interface with sensors (such as DHT sensor, water level sensor, touch sensor, and vibration sensor) and control external components (such as LEDs).

- WiFiClient Module: This module is part of the ESP32's WiFi library and provides functionality for creating a client that can connect to a specified internet IP address and port.

- Variables: Various variables are defined to store WiFi credentials, server information, pin configurations, sensor readings, and ThingSpeak API key.

7.2 Code

7.2.1 Arduino UNO Code

```
#include <Wire.h>
#include <TinyGPS.h>
#include <SoftwareSerial.h>

unsigned long fix_age;

SoftwareSerial GSM(4,2); // Initialize software serial for communication with GSM module

TinyGPS gps; // Initialize TinyGPS object to handle GPS data

void gpsdump(TinyGPS &gps); // Function declaration to print GPS data
bool feedgps(); // Function declaration to feed GPS data
void getGPS(); // Function declaration to get GPS data

long lat, lon; // Variables to hold latitude and longitude data
float LAT, LON; // Variables to store latitude and longitude in float format
long measurement; // Variable to hold shock sensor measurement

char inchar; // Will hold the incoming character from the GSM shield

int buz = 9; // Pin for the buzzer
int stop = 0; // Variable to stop buzzer
int ir = 3; // Pin for the shock sensor
int alcohol = 7; // Pin for the alcohol sensor
int eye=6; // Pin for the eye blink sensor

const char *phone_no[] = {
  "+916305427492",
  "+918639150673"

  };

void setup(){
pinMode(ir, INPUT); // Set shock sensor pin as input
pinMode(alcohol, INPUT); // Set alcohol sensor pin as input
pinMode(eye, INPUT); // Set eye blink sensor pin as input
Serial.begin(9600); // Initialize serial communication
```

```
Wire.begin(); // Initialize I2C communication

pinMode(buz, OUTPUT); // Set buzzer pin as output

 GSM.begin(9600); // Initialize GSM communication
 Serial.begin(9600); // Initialize serial communication
 Serial.println("Initializing.... "); // Print initialization message
 initModule("AT","OK",1000); // Initialize GSM module
 initModule("ATE1","OK",1000); // Enable echo
 initModule("AT+CPIN?","READY",1000); // Check SIM card status
 initModule("AT+CMGF=1","OK",1000); // Set SMS text mode
 initModule("AT+CNMI=2,2,0,0,0","OK",1000); // Set new SMS message indications
 Serial.println("Initialized Successfully"); // Print initialization success message

 GSM.print("AT+CMGS=\"");GSM.print(phone_no[1]);GSM.println("\"\r\n"); // Set phone number for
SMS
 delay(1000); // Delay
 GSM.println("Welcome to Arduino Based Vehicle Accident Alert System using GPS, GSM, Shock
Sensor."); // Send welcome message
 delay(300); // Delay
 GSM.write(byte(26)); // Send Ctrl+Z character to terminate SMS
 delay(3000); // Delay
 getGPS(); // Get GPS data
}

void loop(){
long measurement = digitalRead(ir); // Read shock sensor value
long eye_status = digitalRead(eye); // Read eye blink sensor value
long alcohol_status = digitalRead(alcohol); // Read alcohol sensor value
delay(1000); // Delay
Serial.println(measurement); // Print shock sensor value

// Check if eyes are closed
if (eye_status == 0) {
   Serial.println("Eyes r Close");
   digitalWrite(buz, HIGH); // Turn on buzzer
}
else{
    digitalWrite(buz, LOW); // Turn off buzzer
}

// Check if alcohol is detected
if (alcohol_status == 0) {
```

```
    Serial.println("Alcohol Detected");
    SendSMS("Alcohol Detected :: while driving !!! "); // Send SMS
}

// Check if shock sensor is triggered
if(measurement == 0){
sms(); // Send SMS
Serial.println("SMS SENT SUCCESSFULLY"); // Print success message
}

// Check if incoming message from GSM module
if(GSM.available() >0){inchar=GSM.read();
if(inchar=='R'){inchar=GSM.read();
if(inchar=='I'){inchar=GSM.read();
if(inchar=='N'){inchar=GSM.read();
if(inchar=='G'){
 GSM.print("ATH\r");
 delay(1000);
 getGPS();
 GSM.print("AT+CMGS=\"");GSM.print(phone_no[1]);GSM.println("\"\r\n");
 delay(1000);
 GSM.println("RING Reply");
 GSM.print("http://maps.google.com/?q=loc:");
 GSM.print(LAT/1000000,7);
 GSM.print(",");
 GSM.println(LON/1000000,7);
 delay(300);
 GSM.write(byte(26));
 delay(5000);
   }
  }
 }
 }
}

}

// Get GPS data
void getGPS(){
bool newdata = false;
unsigned long start = millis();
// Every 1 seconds we print an update
while (millis() - start < 1000){
```

```
    if (feedgps ()){
      newdata = true;
    }
  }
if (newdata){
    gpsdump(gps);
  }
}

// Feed GPS data
bool feedgps(){
while (Serial.available()){
    if (gps.encode(Serial.read()))
      return true;
  }
  return 0;
}

// Print GPS data
void gpsdump(TinyGPS &gps){
  gps.get_position(&lat, &lon);
  LAT = lat;
  LON = lon;
  {
    feedgps(); // If we don't feed the gps during this long routine, we may drop characters and get checksum
errors
  }
}

// Initialize GSM module
void initModule(String cmd, char *res, int t){
while(1){
    Serial.println(cmd);
    GSM.println(cmd);
    delay(100);
    while(GSM.available()>0){
       if(GSM.find(res)){
        Serial.println(res);
        delay(t);
        return;
       }else{Serial.println("Error");}}
    delay(t);
  }
```

```
}

// Send SMS
void SendSMS(String msg) {
  GSM.println("AT+CMGS=\"+918639150673\"\r");
  delay(1000);
  GSM.println(msg);
  delay(200);
  GSM.println((char)26);
  delay(1000);
}

// Send emergency SMS
void sms(){
getGPS();
GSM.print("AT+CMGS=\"");GSM.print(phone_no[1]);GSM.println("\"\r\n");
delay(1000);
GSM.println("Emergency.Please reach out to this location right away...");
GSM.print("http://maps.google.com/?q=loc:");
GSM.print(LAT/1000000,7);
GSM.print(",");
GSM.println(LON/1000000,7);
delay(300);
GSM.write(byte(26));
delay(5000);
}
```

7.2.2 ESP32 Code

```
#include <WiFi.h>     // Include the WiFi library
#include <DHT.h>      // Include the DHT sensor library
#include <ThingSpeak.h> // Include the ThingSpeak library

#define DHTPIN 14       // Define the pin connected to the DHT11 sensor
#define DHTTYPE DHT11   // Define the DHT11 sensor type

#define WATER_LEVEL_PIN 15  // Define the pin connected to the water level sensor
#define TOUCH_PIN 2         // Define the pin connected to the touch sensor
#define VIBRATION_PIN 4     // Define the pin connected to the vibration sensor

const char* ssid = "Phone1";        // Set your WiFi SSID
const char* password = "adnanali123"; // Set your WiFi password
```

```
const char* server = "api.thingspeak.com"; // Define the ThingSpeak server
const char* api_key = "GEBN3UMB1JMO3IYL";  // Set your ThingSpeak API key

DHT dht(DHTPIN, DHTTYPE); // Create a DHT object

WiFiClient client;  // Create a WiFi client object

void setup() {
  Serial.begin(115200);  // Initialize serial communication
  pinMode(WATER_LEVEL_PIN, INPUT); // Set water level pin as input
  pinMode(TOUCH_PIN, INPUT);       // Set touch sensor pin as input
  pinMode(VIBRATION_PIN, INPUT);  // Set vibration sensor pin as input
  dht.begin();                    // Initialize DHT sensor
  connectWiFi();                  // Connect to WiFi
}

void loop() {
  float temperature = dht.readTemperature();  // Read temperature from DHT11
  float humidity = dht.readHumidity();        // Read humidity from DHT11
  int waterLevel = digitalRead(WATER_LEVEL_PIN); // Read water level
  int touchState = digitalRead(TOUCH_PIN);     // Read touch sensor state
  int vibrationState = digitalRead(VIBRATION_PIN); // Read vibration sensor state

  if (isnan(temperature) || isnan(humidity)) { // Check if temperature or humidity reading failed
    Serial.println("Failed to read from DHT sensor!"); // Print error message
    return; // Exit loop
  }

  // Print sensor readings to serial monitor
  Serial.print("Temperature: ");
  Serial.print(temperature);
  Serial.print(" °C\t Humidity: ");
  Serial.print(humidity);
  Serial.print(" %\t Water Level: ");
  Serial.print(waterLevel);
  Serial.print("\t Touch Sensor: ");
  Serial.print(touchState);
  Serial.print("\t Vibration Sensor: ");
  Serial.println(vibrationState);

  // Send data to ThingSpeak
  ThingSpeak.begin(client);
  ThingSpeak.setField(1, temperature);
```

```
  ThingSpeak.setField(2, humidity);
  ThingSpeak.setField(3, waterLevel);
  ThingSpeak.setField(4, touchState);
  ThingSpeak.setField(5, vibrationState);

  // Turn on LED if touch pin is high, otherwise turn it off
  digitalWrite(LED_PIN, touchState);

  int status = ThingSpeak.writeFields(2479161, api_key); // Write data to ThingSpeak channel
  if (status == 200) { // Check if data sent successfully
    Serial.println("Data sent to ThingSpeak successfully!"); // Print success message
  } else {
    Serial.println("Failed to send data to ThingSpeak."); // Print error message
  }

  delay(20000); // Delay 20 seconds between updates
}

void connectWiFi() {
  Serial.println("Connecting to WiFi..."); // Print connection message
  WiFi.begin(ssid, password); // Connect to WiFi network
  while (WiFi.status() != WL_CONNECTED) { // Wait until connected
    delay(1000);
    Serial.println("Connecting to WiFi..."); // Print connection attempt message
  }
  Serial.println("Connected to WiFi"); // Print connection success message
}
```

8

SYSTEM RESULTS

8. SYSTEM RESULTS

8.1 System Results

The Accident Alert System utilizes vibration sensors to detect collisions or accidents. Upon detection, it sends alert messages to registered mobile numbers. These alerts contain crucial details like time and location of the accident.

The alcohol detection system employs an MQ3 sensor to identify the presence of alcohol within the vehicle. Upon detecting alcohol, it swiftly triggers an alert message to a registered mobile number.
Simultaneously, the system initiates a mechanism to slow down the vehicle, ensuring safety in potentially hazardous situations.
This integrated approach aims to prevent accidents caused by driving under the influence of alcohol. The system's prompt notification mechanism enables timely intervention, potentially averting dangerous scenarios on the road.

The real-time vehicle tracking system utilizes GPS technology to continuously monitor the vehicle's location. The system provides accurate location updates at regular intervals, ensuring precise tracking of the vehicle's movement. Users can set up geofencing alerts to receive notifications when the vehicle enters or exits predefined geographic boundaries.

The driver fatigue detection system utilizes an eye blink sensor to monitor the driver's eye movements continuously. If the sensor detects the driver closing their eyes for more than four seconds, it triggers an alarm within the vehicle. The alarm serves as a warning signal to alert the driver of their drowsy state and the potential dangers of falling asleep at the wheel. By promptly notifying the driver, the system aims to prevent accidents caused by driver fatigue.

The seat belt alert system employs a touch pin sensor to detect whether the driver is wearing their seat belt. If the sensor detects that the seat belt is not fastened, it activates a continuous buzzer alert within the vehicle. This audible alert serves as a reminder to the driver to buckle up for safety. By continuously reminding the driver to wear their seat belt, the system enhances compliance with seat belt regulations and reduces the risk of injury in the event of an accident.

The vehicle environment monitoring system utilizes DHT11 sensors to accurately measure the temperature inside the car cabin. Based on the temperature readings obtained from the sensors, the system automatically regulates the car's air conditioning (AC) system. If the temperature exceeds predefined thresholds indicating discomfort or potential hazards, the AC system activates to maintain a comfortable environment for the occupants. By dynamically adjusting the AC settings in response to temperature variations, the system ensures optimal comfort and safety during the journey.

The fuel level monitoring system utilizes fuel level sensors to accurately detect the fuel levels within the vehicle's tank. These sensors measure the level of fuel present and transmit the data to a cloud platform, such as ThingSpeak, in real-time. By leveraging the capabilities of ThingSpeak, users can conveniently access and observe the fuel level data remotely via web or mobile applications.

Once the fuel level data is transmitted to the ThingSpeak cloud platform, users can visualize the information through graphical representations, charts, or data logs. This allows for easy monitoring of fuel levels over time and enables users to make informed decisions regarding refueling or fuel consumption patterns.

8.2 Screen Shots

8.2.1 Output of ThingSpeak Temperature and Humidity Values

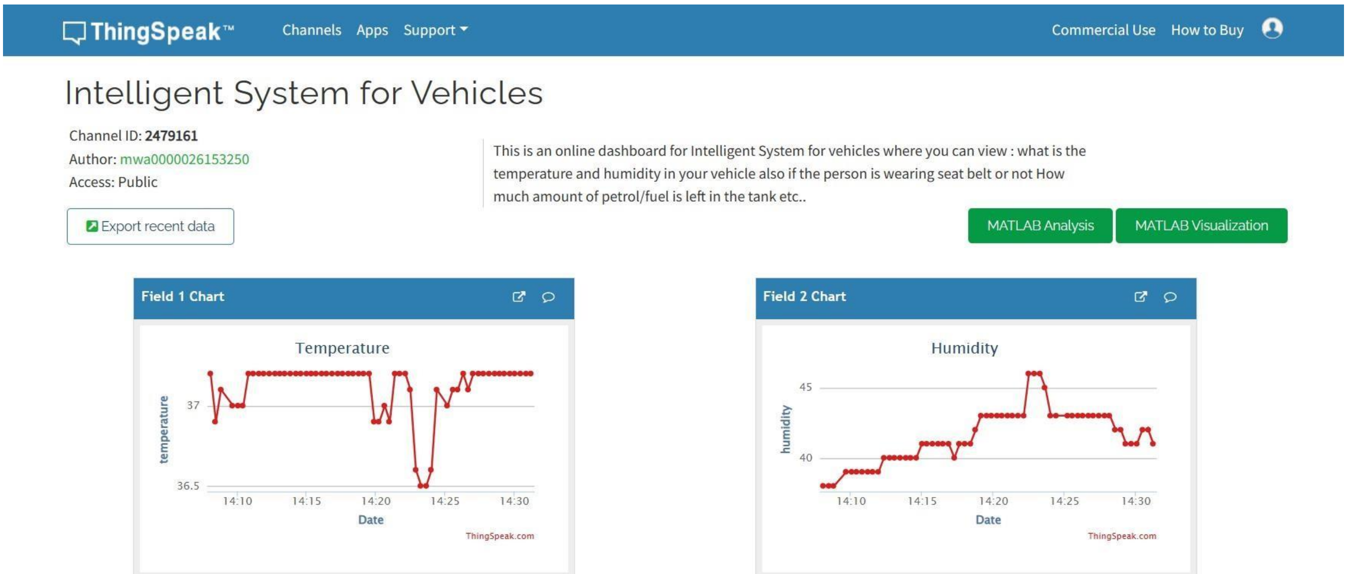


Fig 8.2.1 Output of ThingSpeak Temperature and Humidity Values

8.2.2 Outputs of Fuel level Monitoring and Seat Belt Alert

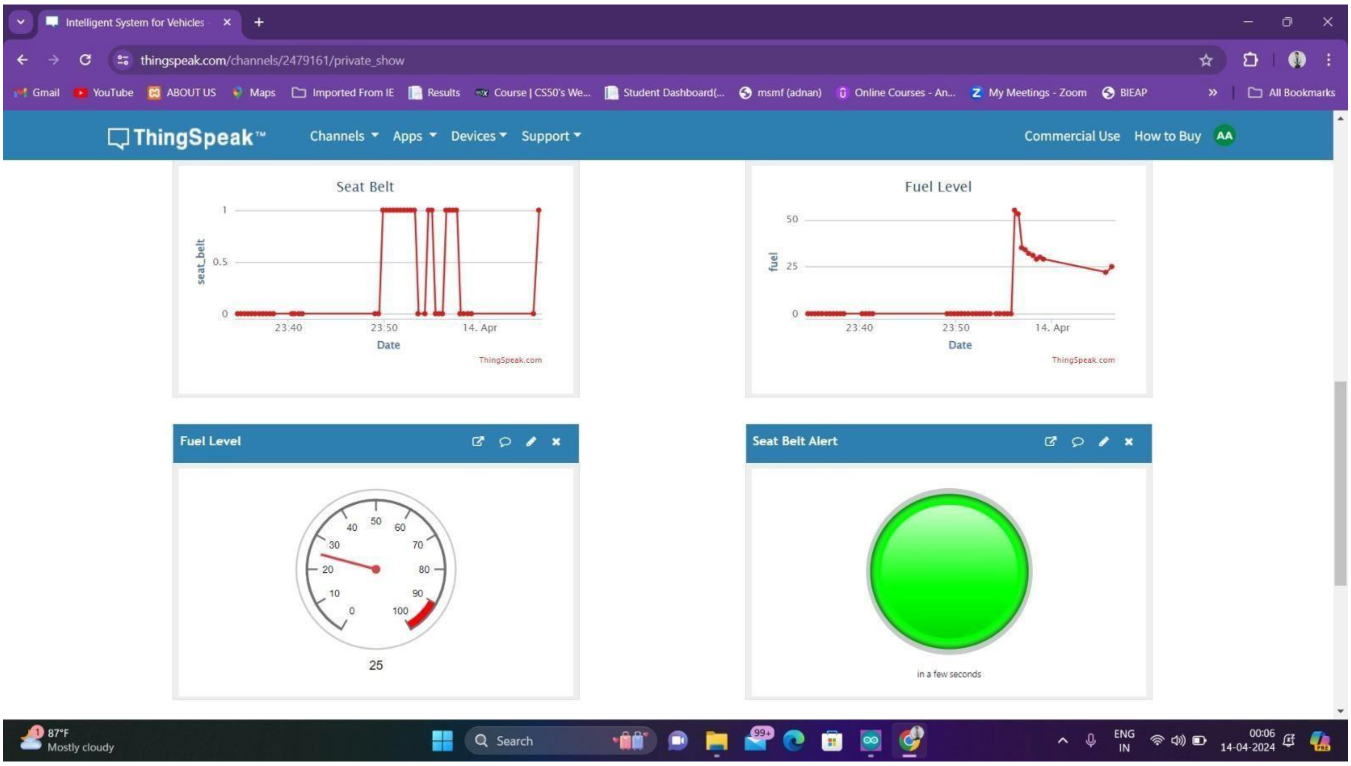


Fig 8.2.2 Outputs of Fuel level Monitoring and Seat Belt Alert

8.2.3 Intelligent Vehicle System Kit

Fig 8.2.3 Intelligent Vehicle System Kit

9

CONCLUSION AND FUTURE SCOPE

9. CONCLUSION AND FUTURE SCOPE

9.1 Conclusion

The integration of Intelligent Systems for Vehicles, powered by Arduino and IoT technology alongside the ESP32 microcontroller, represents a remarkable stride forward in enhancing vehicle safety and monitoring. Through the amalgamation of various sensors and technologies, this system adeptly addresses critical concerns such as drunk driving, accidents, driver fatigue, vehicle environment monitoring, and fuel level monitoring.

Employing Arduino alongside the ESP32 microcontroller adds another layer of flexibility and customization to the project, enabling seamless integration of sensors and efficient data processing. Despite encountering challenges like sensor calibration and data accuracy, real-world evaluations consistently affirm its efficacy in identifying safety risks and potential hazards on the road.

One of the project's distinguishing features is its provision of real-time vehicle tracking information, anti-theft alerts, and vehicle environment monitoring using sensors such as DHT11. These functionalities showcase its versatility and practicality across various scenarios, ensuring safety and convenience for drivers and passengers.

Moreover, with the incorporation of fuel level monitoring using water level sensors, the system enhances its ability to provide crucial insights into vehicle maintenance and operational efficiency. This feature ensures timely refueling and monitors environmental conditions within the vehicle cabin, contributing to overall safety and comfort during journeys.

Despite facing challenges, the project remains steadfast in its commitment to innovation, cost-effectiveness, and user-friendliness. By adopting a holistic approach that not only focuses on preventing accidents but also provides post-crash assistance and monitoring, the Intelligent Systems for Vehicles project sets a new benchmark for proactive vehicle safety solutions. This project serves as a testament to the continuous evolution of technology in addressing real-world issues. By leveraging Arduino, ESP32, and IoT technologies, it establishes itself as a pioneer in intelligent and comprehensive vehicle safety systems, ultimately fostering a safer and more secure driving environment for all.

9.2 Future Scope

In the future, smart systems for vehicle safety and tracking will change how we drive. These advancements, powered by fast-moving technology, promise to make driving safer, more efficient, and better connected.

Future accident alert systems won't just spot accidents; they'll predict and stop them. They'll use better sensors and AI to analyze past data and driver habits, foreseeing risks before they happen. Cars will work with smart roads to share real-time info, making roads safer for everyone.

Alcohol detection systems will get better, using wearable gadgets to keep an eye on alcohol levels in real-time. By combining this with info about the driver, they'll be more accurate. Working with police, they'll help make roads safer by catching drunk drivers.

In the future, spotting tired drivers will be easier. Smart sensors will monitor drivers for signs of drowsiness, and cars will step in to help, like adjusting cruise control. These systems will use smarter alerts to keep drivers awake and safe.

Tracking vehicles will get smarter too. Cars will know the best routes in real-time, based on traffic and weather. They'll use tech to create virtual boundaries that adjust as needed. With better security features, like tamper detection, tracking info will stay safe and trustworthy.

Seat belt alerts will become more than reminders; they'll be safety partners. Using sensors, they'll know when passengers need to buckle up, and they'll adapt to different needs. Teaming up with self-driving cars, they'll make sure everyone stays safe and comfy.

And vehicles will start monitoring the environment inside and out. They'll track things like temperature and air quality to keep passengers comfy. They'll even keep tabs on fuel levels, making refueling easier and improving performance.

The future of vehicle safety and tracking is bright, thanks to new tech and a commitment to safety and efficiency. These systems will cut down on accidents, stop drunk driving, and make driving easier and safer for everyone.

10. REFERENCES

1) Nazia Parveen, Ashif Ali, Aleem Ali, "IOT Based Automatic Vehicle Accident Alert System," presented at the 2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA), Galgotias University, Greater Noida, UP, India, Oct 30-31, 2020.

2) "Vehicle Accident Detection and Reporting System Using Gps and Gsm” by AboliRavindraWakure, ApurvaRajendraPatkar, IJERGS April 2014.

3) Purva Javale, Shalmali Gadgil, Chinmay Bhargave, Yogesh Kharwandikar, Vaishali Nandedkar, “Accident Detection and Surveillance System using Wireless Technologies”, IOSR Journal of Computer Engineering (IOSR-JCE), pp 38-43, Volume 16, Issue 2, March April 2014.

4) Dr. R. Prasanthi, M.U. Nitish Babu, B. Jaswanth, G. Sumith Chandra, "ACCIDENT DETECTION AND ALERT SYSTEM USING ARDUINO," in International Research Journal of Modernization in Engineering Technology and Science, Volume 05, Issue 04, April 2023. 5) Nanda, S., Joshi, H., & Khairnar, S. (2018). An IOT Based Smart System for Accident Prevention and Detection. 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA).

6) S. L. Fong, D. C. W. Yung, F. Y. H. Ahmed, and A. Jamal, “Smart city bus application with quick response (qr) code payment,” ser. ICSCA ’19. New York, NY, USA: Association for Computing Machinery, 2019, p. 248–252.

7) K. Gudur, A. Ramesh, and S. R, “A vision-based deep on-device intelligent bus stop recognition system,” in Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers, ser. UbiComp/ISWC ’19 Adjunct. New York, NY, USA: Association for Computing Machinery, 2019, p. 963 968.

8) Saha, M. Shinde, and S. Thadeshwar, “Iot based air quality monitoring system using wireless sensors deployed in public bus services,” ser. ICC ’17. New York, NY, USA: Association for Computing Machinery, 2017.

9) M. Kumar, “R., and dr,” R. Senthil. Effective control of accidents using routing and tracking system with integrated network of sensors, vol. 2, p. 4, 2013.

10) R. Liu, Z. Yin, W. Jiang, and T. He, “Wibeacon: Expanding ble location-based services via wifi,” in Proceedings of the 27th Annual International Conference on Mobile Computing and Networking, ser. MobiCom ’21. New York, NY, USA: Association for Computing Machinery, 2021, p. 83–96.

11) J. J. T. Dai, X. Bai, and Z. Shen, “Mobile phone based drunk driving detection pervasive computing

technologies for healthcare. 2010, 4th international IEEE conference," p, vol. 1, March 2010.

12) Chen, Y. Chiang, F. Chang, and H. Wang, "Toward real time precise point positioning: Differential GPS based on IGS ultra rapid product. sice annual conference," The Grand Hotel, Taipei, Taiwan, August, vol. 18.

13) Liu, X. Xu, X. Chen, E. Mai, H. Y. Noh, P. Zhang, and L. Zhang, "Individualized calibration of industrial grade gas sensors in air quality sensing system," ser. SenSys '17. New York, NY, USA: Association for Computing Machinery, 2017. [Online]. Available: https://doi.org/10.1145/3131672.3136998

14) T. Duchowski, S. Jorg, T. N. Allen, I. Giannopoulos, and K. Krejtz, ¨ "Eye movement synthesis," in Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications, ser. ETRA '16. New York, NY, USA: Association for Computing Machinery, 2016,p.147154.[Online].Available:https://doi.org/10.11 45/2857491.2857528

15) C. Hahn, S. Feld, and H. Schroter, "Predictive collision management for time and risk-dependent path planning," in Proceedings of the 28th International Conference on Advances in Geographic Information Systems, ser. SIGSPATIAL '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 405 408.[Online].Available:https://doi.org/10.1145/339753 6.3422252

IEEE IDCIoT REVA UNIVERSITY Bengaluru, India

2nd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT-2024)

4-6, January 2024 | Bengaluru, India

Certificate of Presentation

This is to certify that

P Vishnu Kumar

has successfully presented the paper entitled

IoT based Intelligent Systems for Vehicle

at the
2nd International Conference on
Intelligent Data Communication Technologies and Internet of Things (IDCIoT-2024)
organized by School of Computer Science and Engineering, REVA University,
Bengaluru, India during 4-6, January 2024.

Session Chair	Dr. H. Anwar Basha Organizing Secretary	Dr. Syed Muzamil Basha Conference Chair	Dr. Ashwinkumar U.M Director

XPLORE COMPLIANT ISBN: 979-8-3503-2753-3

IDCIoT

2nd International Conference on
Intelligent Data Communication Technologies and Internet of Things
(IDCIoT-2024)

4-6, January 2024 | Bengaluru, India

Certificate of Presentation

This is to certify that

Adnan Ali

has successfully presented the paper entitled

IoT based Intelligent Systems for Vehicle

at the
2nd International Conference on
Intelligent Data Communication Technologies and Internet of Things (IDCIoT-2024)
organized by School of Computer Science and Engineering, REVA University,
Bengaluru, India during 4-6, January 2024.

Session Chair	Dr. H. Anwar Basha Organizing Secretary	Dr. Syed Muzamil Basha Conference Chair	Dr. Ashwinkumar U.M Director

XPLORE COMPLIANT ISBN: 979-8-3503-2753-3

IEEE IDCIoT REVA UNIVERSITY Bengaluru, India

2nd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT-2024)

4-6, January 2024 | Bengaluru, India

Certificate of Presentation

This is to certify that

Akula Zaheer Sha

has successfully presented the paper entitled

IoT based Intelligent Systems for Vehicle

at the
2nd International Conference on
Intelligent Data Communication Technologies and Internet of Things (IDCIoT-2024)
organized by School of Computer Science and Engineering, REVA University,
Bengaluru, India during 4-6, January 2024.

Session Chair	Dr. H. Anwar Basha Organizing Secretary	Dr. Syed Muzamil Basha Conference Chair	Dr. Ashwinkumar U.M Director

XPLORE COMPLIANT ISBN: 979-8-3503-2753-3

IEEE

IDCIoT

REVA UNIVERSITY
Bengaluru, India

2nd International Conference on
Intelligent Data Communication Technologies and Internet of Things
(IDCIoT-2024)

4-6, January 2024 | Bengaluru, India

Certificate of Presentation

This is to certify that

Siddem Rajesh

has successfully presented the paper entitled

IoT based Intelligent Systems for Vehicle

at the
2nd International Conference on
Intelligent Data Communication Technologies and Internet of Things (IDCIoT-2024)
organized by School of Computer Science and Engineering, REVA University,
Bengaluru, India during 4-6, January 2024.

Session Chair	Dr. H. Anwar Basha Organizing Secretary	Dr. Syed Muzamil Basha Conference Chair	Dr. Ashwinkumar U.M Director

XPLORE COMPLIANT ISBN: 979-8-3503-2753-3

IoT based Intelligent Systems for Vehicle

P. Vishnu Kumar
Assistant Professor, Department of Internet of Things, G. Pullaiah College of Engineering and Technology, Kurnool, Andhra Pradesh, India.
vishnu4b8@gmail.com

Adnan Ali
Student, Department of Internet of Things, G. Pullaiah College of Engineering and Technology, Kurnool, Andhra Pradesh, India.
adduadnanali@gmail.com

Akula Zaheer Sha
Student, Department of Internet of Things, G. Pullaiah College of Engineering and Technology, Kurnool, Andhra Pradesh, India.
a.zaheersha@gmail.com

Siddem Rajesh
Student, Department of Internet of Things, G. Pullaiah College of Engineering, and Technology, Kurnool, Andhra Pradesh, India.
rajeshroy2515@gmail.com

***Abstract* - This paper introduces an intelligent vehicle system implemented through Internet of Things (IoT) technology, employing various sensors for comprehensive vehicle monitoring. The ESP32 microcontroller processes sensor data and manages the system's actuators. The system focuses on enhancing vehicle safety by detecting drunk driving, accidents, and driver fatigue. Recent techniques, including machine learning algorithms for improved detection accuracy and edge computing for real-time processing, are explored. Challenges faced by such systems, such as sensor calibration, data accuracy, and the integration of complex algorithms, are acknowledged. Despite these challenges, real-world evaluations confirm the system's effectiveness in identifying instances of drunk driving, accidents, and driver fatigue. The system consistently delivers real-time vehicle tracking information and anti-theft alerts. The proposed intelligent vehicle system aims to address these challenges and significantly improve vehicle safety and convenience. It is designed to be innovative, cost-effective, and user-friendly, making it a valuable asset for any driver.**

***Keywords* - Accident Alert, Alcohol Detection, Real-time Vehicle Tracking, Driver Fatigue Detection, Seat-Belt Alert, Internet of Things (IoT) Technology, Post-crash Assistance, Vehicle Safety Monitoring, Safety Innovation.**

I. Introduction

The Intelligent System for Vehicles (ISV) is a ground-breaking development in automotive technology that has been painstakingly designed to completely reimagine the field of monitoring, communication, and road safety. This innovative program is a monument to technical innovation that is steadfastly committed to improving driver welfare and raising the bar for road safety. Intelligent System for Vehicles skillfully combines a wide range of innovative features and cutting-edge technology, resulting in the development of an advanced, networked automobile environment.

Each component greatly adds to the core capabilities of the system, from the intricate MQ-3 sensors for accurate alcohol detection to the perceptive vibration sensors specialized in accident identification, the use of Twilio for internet-based messaging, GSM modules for non-internet-based communication, GPS modules for real-time tracking, and the interactive real-time monitoring made possible by the Blynk app.

Intelligent System for Vehicles 's steadfast dedication to passenger safety is noteworthy, as demonstrated by the addition of a Seat Belt Alert system. By using a comprehensive approach, separate parts become a cohesive, intelligent system that goes beyond traditional car safety paradigms. With an emphasis on accident prevention, alcohol consumption risk mitigation, and fostering a culture of responsible driving, Intelligent System for Vehicles seeks to close important gaps in safety and communication.

Presented as the height of innovation, ISV promises safer roads and increases the effectiveness of vehicle monitoring while simultaneously promoting cautious driving practices. This investigation explores the complex aspects of the Intelligent System for Vehicles, carefully breaking down its components, revealing all of its functions, and speculating on the significant impact it will likely have on the automotive sector going forward. Intelligent System for Vehicles is shown as a revolutionary force that has the potential to completely change the safety and communication aspects of cars in our constantly linked world.

II. Existing System

The current system is engineered to rapidly detect car accidents and alert emergency services efficiently. It utilizes an accelerometer to identify abrupt changes in motion, triggering an alert if predefined thresholds are surpassed. Once activated, the alert system promptly sends a detailed SMS to aid centers, providing crucial accident information, including time and location, for a swift emergency response. Users are empowered to halt the alert message if assistance is unnecessary, offering a level of control. Communication is facilitated through GSM, and the system integrates GPS for precise accident location tracking. Additionally, the system furnishes insights into the vehicle's rollover angle, enhancing the overall accident information.

While the system showcases merits in swift accident detection, timely emergency response, user control, and accurate location tracking, certain demerits exist. Dependence on predefined motion thresholds may lead to false alarms or oversight of less intense accidents. Limited information provision and reliance on GSM networks for communication may pose challenges in areas with limited coverage, signaling areas for potential improvement in future intelligent vehicle systems.

III. Proposed System

The Intelligent System for Vehicles (ISV) represents a comprehensive solution aimed at enhancing automotive safety and monitoring. This advanced system incorporates several key features to improve road safety and driver well-being. The primary components of the proposed system include:

A. Alcohol Detection:

To achieve alcohol detection, the system incorporates advanced sensors designed to monitor alcohol levels within the vehicle's cabin. These sensors utilize specialized technology to accurately assess the presence of alcohol. In the event of alcohol detection, the system promptly issues alerts, discouraging drunk driving and promoting responsible behavior behind the wheel. This proactive approach enhances overall safety and aligns with the system's goal of preventing impaired driving.

B. Accident Alerting:

In the event of an accident, the proposed system offers instant accident detection capabilities. It employs sensors and collision detection algorithms to identify accidents swiftly, triggering immediate alerts. These alerts can help facilitate a rapid response by sending SMS to the registered mobile number and potentially save lives.

C. Real-time Vehicle Tracking:

To achieve security in the system, real-time vehicle tracking plays a crucial role. The implementation of GPS-based tracking enables constant monitoring of vehicles. This feature allows owners to remotely track their vehicles in real-time, serving as an effective theft deterrent. The ability to monitor and manage fleets enhances overall vehicle security and proves to be a valuable asset for businesses.

D. Driver Sleep Alert:

Fatigue is a significant contributor to accidents, particularly during long journeys. The system combats this risk by monitoring driver fatigue through blink detection. When signs of drowsiness are detected, it issues timely alerts to keep the driver alert and focused on the road, reducing the risk of accidents caused by fatigue.

E. Seat Belt Alert:

Promoting seat belt usage is crucial for occupant safety. The system includes a seat belt alert feature that reminds drivers and passengers to fasten their seat belts. This simple yet effective reminder contributes to increased seat belt compliance and overall safety.

To reduce the impact in case of an accident, an intelligent vehicle system has been proposed, leveraging Internet of Things (IoT) technology. Equipped with various sensors, an ESP32 microcontroller processes data and manages actuators to enhance vehicle safety. The system focuses on detecting drunk driving, accidents, and driver fatigue. Additionally, it offers real-time vehicle tracking and anti-theft alerts. Real-world evaluations demonstrate its effectiveness in identifying safety issues and consistently providing tracking information and anti-theft alerts. This affordable and user-friendly system has the potential to significantly improve overall vehicle safety and convenience.

The Intelligent System for Vehicles (ISV) leverages these integrated features to create a comprehensive solution for vehicle safety. By addressing key factors contributing to accidents and promoting responsible driving behavior, it aims to enhance road safety, reduce accidents, and protect lives.

This system is designed to be user-friendly and cost-effective, providing a valuable tool for both individual vehicle owners and businesses operating vehicle fleets.

High accuracy in the intelligent vehicle system is achieved through advanced sensors and the ESP32 microcontroller's efficient data processing. The system focuses on detecting drunk driving, accidents, and driver fatigue with precision. Sophisticated algorithms and real-time processing contribute to its reliable performance in real-world evaluations. The combination of these elements ensures the system's effectiveness in enhancing overall safety and convenience for drivers.

IV. Hardware Components

A. ESP32 Microcontroller

The ESP32, serving as the brain of the system, is a versatile microcontroller meticulously crafted by Espress if Systems. Boasting dual-core processors, Wi-Fi, Bluetooth connectivity, and an array of GPIO pins, it stands as an ideal choice for Internet of Things and embedded applications. Its affordability and robust open-source support have solidified its popularity among developers, facilitating innovation in various projects. With the ESP32 at its core, our system gains the computational power and connectivity needed to seamlessly integrate and orchestrate diverse hardware components, driving the intelligence behind the Intelligent System for Vehicles (ISV). As shown in Figure 1.0, you can see the ESP32 microcontroller is the central component of our system.

B. Integration of GSM

The Global System for Mobile Communications standard is followed by the GSM module, a critical communicator in our system. It is small and efficient, and it runs smoothly across multiple frequency bands, allowing for dependable voice, text messaging, and data transmission. This module, which integrates seamlessly with the Microcontroller, enables our system to send timely accident alerts through SMS. As shown in Figure 1.1, you can observe the GSM module that facilitates communication in our system. The designed Arduino to interface with the GSM module using AT instructions, resulting in a simple and responsive communication route. This tiny GSM module plays a critical part in the overall scheme of our Intelligent System for Vehicles (ISV), turning the system's insights into timely alarms, leading to better safety and rapid reaction capabilities.

C. Integration of GPS

The GPS module, a navigation wizard in our system, is powered by the Global Positioning System. This small device provides precise worldwide determination of location, speed, and time. By utilizing signals from a satellite network, our technology is able to deliver accurate and real-time coordinates for the vehicle. The GPS module provides critical information such as time and coordinates using the NMEA sentence $GPGGA, ensuring a dependable data source for our Intelligent System for Vehicles (ISV). This inconspicuous GPS module emerges as a silent yet vital contributor in the landscape of Internet of Things - based car safety, leading the system with maximum precision and reliability. In Figure 1.3, you can see the GPS module, which plays a crucial role in our system.

D. Integration of Vibration Sensor

The vibration sensor, our system's silent protector. This simple device detects rapid changes in the vehicle's velocity by measuring acceleration forces and vibrations. This sensor, which is widely utilized in a variety of applications ranging from industrial machinery monitoring to motion detection in electrical gadgets, discreetly guarantees that our system remains attentive. As part of the Intelligent System for Vehicles (ISV), it is critical in spotting accidents or significant events and delivering critical data to the system. This unassuming vibration sensor is the unsung hero of Internet of Things and car safety, ensuring our system is perfectly tuned to the intricacies of the vehicle's movement. In Figure 1.4, you can see the unassuming vibration sensor, a crucial component in our Intelligent System for Vehicles (ISV).

E. Integration of Eyeblink Sensor

The eyeblink sensor, a perceptive component of our system, excels in capturing and evaluating a person's eyeblink frequency and intensity. Beyond its small size, the gadget has a wide range of applications, from research and healthcare to human-computer interaction studies. This unobtrusive sensor examines blinking patterns in our setting, providing insights about the driver's cognitive and emotional condition. It contributes to overall safety as part of the Intelligent System for Vehicles (ISV) by assessing the driver's attention. This simple eyeblink sensor serves as a perceptive ally in the area of Internet of Things -based car monitoring, ensuring our system remains attentive to the human element of driving. As shown in Figure 1.5, you can observe the simple eyeblink sensor, a perceptive ally in our ISV.

F. Use of Actuators

If a person consumes alcohol and attempts to drive the vehicle, the engines will not turn on, as indicated by actuators such as motors and a buzzer as an alarm. If a person does not wear the seat belt, the buzzer will continue to ring until and unless they wear the seat belt in order to reduce the impact in case of an accident.

If a driver is sleepy or fatigued, the buzzer will sound as an alert to wake him up if he shuts his eyes for more than 3-5 seconds.

Fig 1.0: ESP32 Microcontroller

Fig 1.1: GSM Module

Fig 1.4: Vibration Sensor

Fig 1.3: GPS Module

Fig 1.5: Eyeblink Sensor

V. *Software Components*

A. Arduino IDE

Our coding canvas, Arduino IDE, provides a user-friendly environment for programming Arduino microcontrollers. It is the creative environment where code is developed, compiled, and uploaded to ensure a smooth development experience for our Intelligent System for Vehicles (ISV).

B. Twilio Platform

Twilio is a cloud platform that streamlines the integration of phone, SMS, and communication features into our system. Twilio provides our Intelligent System for Vehicles (ISV) with comprehensive communication options, guaranteeing timely and dependable notifications.

C. Blynk App

Our smartphone companion, Blynk, is a simple app that easily enables our Intelligent System for Vehicles (ISV). It enables the construction and management of Internet of Things projects by offering an easy-to-use interface for interacting with hardware devices like as Microcontroller. Our ISV becomes more controllable and responsive with Blynk, increasing user engagement and control. As shown in Figure 1.6, you can observe the Blynk app, which provides user-friendly control).

Fig 1.6: Blynk App

D. IoT Technology Integration

The entire system is underpinned by IoT technology, seamlessly connecting various components for real-time monitoring and data processing. The integration of IoT enhances the overall functionality and effectiveness of our Intelligent System for Vehicles (ISV), making it a smart and networked solution for automotive safety and monitoring.

VI. Quantitative Analysis

The Intelligent System for Vehicles (ISV) underwent a meticulous quantitative analysis, rigorously evaluating its performance in critical safety scenarios. A focus was placed on precision, recall, accuracy, response time, false positive and false negative rates, user feedback, and system reliability. In terms of accuracy metrics, the ISV's proficiency in accident detection, alcohol presence monitoring, and fatigue alerts was assessed using standard measurements such as precision, recall, and the F1 score. An emphasis on response time gauged the system's real-time capabilities, crucial for timely responses to safety concerns. The analysis delved into false positive and false negative rates, offering insights into the ISV's precision and sensitivity. User feedback surveys and assessments provided quantitative data on usability, effectiveness, and overall satisfaction, contributing to a user-centric evaluation. Additionally, the quantitative analysis included an assessment of system reliability and availability under diverse conditions. In summary, the quantitative analysis comprehensively scrutinized the ISV's accuracy, response time, false positive and false negative rates, user satisfaction, and reliability, affirming its efficacy in enhancing vehicle safety and monitoring.

The Intelligent System for Vehicles (ISV) combines a comprehensive approach to optimizing systems and safety features. It aligns hardware and software components to achieve the best performance. The ESP32 Microcontroller, which serves as the brain of the system, undergoes rigorous optimization to ensure efficient data processing. The sensors are carefully calibrated to capture accurate, real-time data, improving the system's responsiveness. The software components prioritize code efficiency and use the Arduino IDE for coding. Advanced methods are used to fine-tune real-time processing, reducing delays and increasing system agility. The ISV focuses on achieving high accuracy in safety detection through precise sensors and advanced algorithms. The ESP32 Microcontroller plays a crucial role in swiftly and accurately processing sensor data, thanks to its dual-core CPUs. Real-time processing is essential for enhancing accuracy, enabling quick analysis, and detecting safety concerns early on. Safety measures are implemented, including carefully selected actuators. These actuators include mechanisms like an engine locking system to prevent operation in hazardous conditions and a seat belt alarm to promote safety precautions. The ISV's holistic approach to safety and system optimization demonstrates its commitment to optimal performance while prioritizing user safety and well-being.

VII. Results

When an accident is detected, our Intelligent System for Vehicles (ISV) immediately launches a sequence of actions. The GPS module pinpoints the exact location, guaranteeing that an alarm is transmitted to the selected emergency services as soon as possible through the integrated GSM module. Concurrently, an alert message is sent to the registered mobile number, providing critical information for prompt help. As shown in Figure 1.7, you can see the Accident Alert message that our ISV generates in case of an accident.

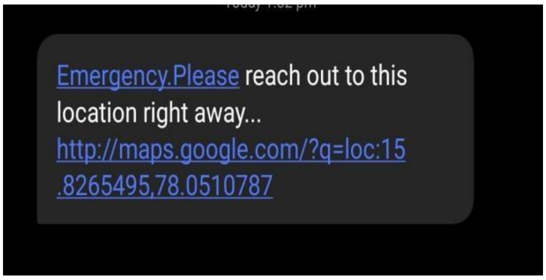


Fig 1.7: Accident Alert Message

The MQ-3 sensor functions as a vigilant guardian in cases of alcohol detection or drunk driving. If alcohol levels exceed predetermined values, the device sends an immediate notice. It also prevents the engine from starting and sends an alert to the registered mobile number, ensuring a precautionary step against potential accidents caused by impaired driving. As shown in Figure 1.7, you can observe the Alcohol Detection Alert message generated by the Intelligent System for Vehicles (ISV) when alcohol levels are detected above the threshold.

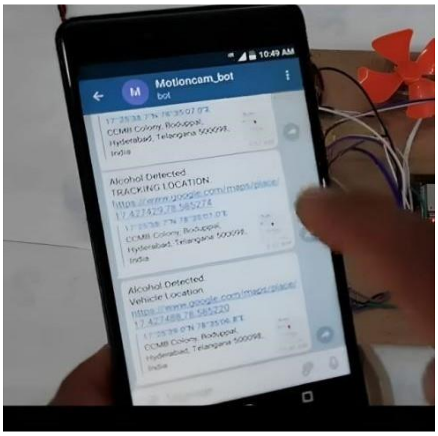


Fig 1.7: Alcohol Detection Alert Msg

The integration of a cloud platform, such as Blynk App, allows for real-time vehicle monitoring. This feature enables car owners to track and monitor their vehicles remotely in real time, offering useful information for better management and security. As shown in Figure 1.8, you can see a visual representation of the real-time vehicle monitoring provided by the Blynk App, which is an integral part of our Intelligent System for Vehicles (ISV).

Fig 1.8: Accident Location

Our ISV's seat belt alert ensures that everyone buckles up, hence increasing vehicle safety. This basic feature encourages the use of seat belts, which is a vital part of overall safety. Furthermore, our system watches the driver using a fatigue monitoring feature that makes use of the Blink sensor. If the driver feels fatigued, the technology informs them ahead of time, allowing them to remain focused on the road. With these characteristics, our Intelligent System for Vehicles (ISV) not only detects accidents, drunk driving, and fatigue, but also acts proactively. This all-in-one solution increases vehicle safety, making our Intelligent System for Vehicles (ISV) a reliable road companion for everyone.

VIII. Conclusion

In conclusion, our Intelligent System for Vehicles (ISV) represents a significant step forward in the development of Internet of Things technologies for vehicle safety. The ISV takes a comprehensive approach to tackling important safety concerns by seamlessly integrating a range of cutting-edge technologies, such as Alcohol Detection, Accident Alerting, Real-time Vehicle Tracking, Driver Sleep Alert, and Seat Belt Alert. Evaluations in the real-world attest to its effectiveness in prompt risk detection and response. In our approach, emergency services are selected through automated processes within the intelligent vehicle system. Predefined algorithms and protocols analyze sensor data to identify critical situations, such as accidents or driver fatigue. Once a potential emergency is detected, the system automatically triggers alerts and forwards relevant information to designated emergency services. The selection of appropriate services is influenced by factors like the severity of the situation, location data, and the specific type of emergency. Integration with emergency response systems ensures a swift and tailored response, aligning with our goal to enhance safety and streamline the emergency assistance process. This project actively works to avoid accidents and promote safe driving practices in addition to identifying possible dangers. The ISV is positioned as a useful tool for both fleet managers and owners of individual vehicles because to its user-friendly design, low cost, and integration of cutting-edge technologies.

IX. References

1) Nazia Parveen, Ashif Ali, Aleem Ali, "IOT Based Automatic Vehicle Accident Alert System," presented at the 2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA), Galgotias University, Greater Noida, UP, India, Oct 30-31, 2020.
2) "Vehicle Accident Detection and Reporting System Using Gps and Gsm" by AboliRavindraWakure, ApurvaRajendraPatkar, IJERGS April 2014.
3) Purva Javale, Shalmali Gadgil, Chinmay Bhargave, Yogesh Kharwandikar, Vaishali Nandedkar, " Accident Detection and Surveillance System using Wireless Technologies", IOSR Journal of Computer Engineering (IOSR-JCE), pp 38-43, Volume 16, Issue 2, March-April 2014.
4) Dr. R. Prasanthi, M.U. Nitish Babu, B. Jaswanth, G. Sumith Chandra, "ACCIDENT DETECTION AND ALERT SYSTEM USING ARDUINO," in International Research Journal of Modernization in Engineering Technology and Science, Volume 05, Issue 04, April 2023.
5) Nanda, S., Joshi, H., & Khairnar, S. (2018). An IOT Based Smart System for Accident Prevention and Detection. 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA).
6) S. L. Fong, D. C. W. Yung, F. Y. H. Ahmed, and A. Jamal, " Smart city bus application with quick response (qr) code payment," ser. ICSCA '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 248–252.
7) K. Gudur, A. Ramesh, and S. R, " A vision-based deep on-device intelligent bus stop recognition system," in Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers, ser. UbiComp/ISWC '19 Adjunct. New York, NY, USA: Association for Computing Machinery, 2019, p. 963 –968.
8) Saha, M. Shinde, and S. Thadeshwar, " Iot based air quality monitoring system using wireless sensors deployed in public bus services," ser. ICC '17. New York, NY, USA: Association for Computing Machinery, 2017.
9) M. Kumar, " R., and dr," R. Senthil. Effective control of accidents using routing and tracking system with integrated network of sensors, vol. 2, p. 4, 2013.
10) R. Liu, Z. Yin, W. Jiang, and T. He, " Wibeacon: Expanding ble location-based services via wifi," in Proceedings of the 27th Annual International Conference on Mobile Computing and Networking, ser. MobiCom '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 83 –96.
11) J. J. T. Dai, X. Bai, and Z. Shen, " Mobile phone based drunk driving detection pervasive computing technologies for healthcare. 2010, 4th international IEEE conference," p, vol. 1, March 2010.
12) Chen, Y. Chiang, F. Chang, and H. Wang, "Toward real-time precise point positioning: Differential GPSbased on IGS ultra rapid product. sice annual conference," The Grand Hotel, Taipei, Taiwan, August, vol. 18.
13) Liu, X. Xu, X. Chen, E. Mai, H. Y. Noh, P. Zhang, and L. Zhang, " Individualized calibration of industrial-grade gas sensors in air quality sensing system," ser. SenSys '17. New York, NY, USA: Association for Computing Machinery, 2017. [Online]. Available: https://doi.org/10.1145/3131672.3136998
14) T. Duchowski, S. Jorg, T. N. Allen, I. Giannopoulos, and K. Krejtz, ¨ "Eye movement synthesis," in Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications, ser. ETRA '16. New York, NY, USA: Association for Computing Machinery, 2016,p.147154.[Online].Available:https://doi.org/10.1145/2857491.2857528
15) C. Hahn, S. Feld, and H. Schroter, "Predictive collision management for time and risk-dependent path planning," in Proceedings of the 28th International Conference on Advances in Geographic Information Systems, ser. SIGSPATIAL '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 405 – 408.[Online].Available:https://doi.org/10.1145/3397536.3422252