

INNOVATIVE SAFETY SOLUTION FOR SMART DRIVING

Submitted in partial fulfilment of the required for the award of

Masters of Computer Science

By

PREM K

REGISTER NO (42734014)



**DEPARTMENT OF COMPUTER SCIENCE
SCHOOL OF SCIENCE AND HUMANITIES**

**SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade “A++” by NAAC

JEPPIAR NAGAR, RAJIV GANDHI SALAI, CHENNAI 600 119

APRIL 2024

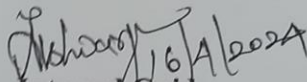


**SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**
Accredited with Grade "A++" by NAAC

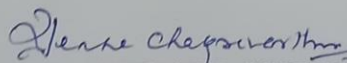
DEPARTMENT OF COMPUTER SCIENCE

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **PREM K (42734014)** who carried out the project entitled "**INNOVATIVE SAFETY SOLUTION FOR SMART DRIVING**" under our supervision from January 2024 to April 2024.


Internal Guide

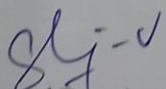
Ms. AISHWARYA B, M.C.A

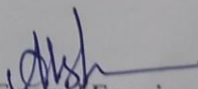


**DEAN OF ARTS AND SCIENCE
Dr. REKHA CHAKRAVARTHI M.E., Ph.D.**

Submitted for viva voce examination held on

20/4/2024

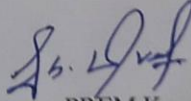

Internal Examiner


External Examiner

DECLARATION

I am, **PREM K (42734014)** hereby declare that the project report entitled "INNOVATIVE SAFETY SOLUTION FOR SMART DRIVING" done by me under the guidance of Ms. **AISHWARYA B**, M.C.A Department of Computer Science at SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY, jeppiaar Nagar, Rajiv Ghandhi Salai, Chennai-600119 is submitted in partial fulfilment of the requirements for the award of Master of Science degree in Computer Science .

DATE: 20/4/24
PLACE: Chennai


PREM K
Signature of the Candidate

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. REKHA CHAKRAVARTHI M.E., Ph.D. Dean of Arts and Science** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my project guide **Ms. AISHWARYA B, M.C.A., Assistant Professor** Department of Computer Science for their valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff member of the Department of Computer Science who were helpful in many ways for the completion of the project

PREM K

ABSTRACT

This project presents an innovative safety system for automobiles, leveraging artificial intelligence (AI) and biometric sensors to detect and respond to driver impairments and emergencies effectively. The system addresses critical safety concerns by incorporating multiple layers of protection. Firstly, it integrates a breathalyzer-like device into the vehicle's ignition system, preventing drunk driving incidents by inhibiting ignition if alcohol is detected. Additionally, it alerts the driver with a buzzer if they attempt to operate the vehicle while under the influence. Furthermore, a heartbeat sensor continuously monitors the driver's heart rate. In the event of a heart attack or a significant drop in heart levels, the sensor collects vital vehicle data, including GPS location, and promptly notifies both the caregiver and the vehicle owner. Moreover, AI technology is employed to detect various driver behaviors, such as drowsiness, through real-time camera feeds or high-resolution images. When signs of drowsiness are detected, the system triggers audible alerts and activates seat vibrations to alert the driver, reducing the risk of accidents due to driver fatigue. This comprehensive approach to road safety integrates AI and biometric sensors to mitigate risks associated with impaired driving and medical emergencies, ultimately saving lives and enhancing overall road safety.

TABLE OF CONTENT

CHAPTER NO	TITLE	PAGE NO
1.INTRODUCTION		
1.1	About the Project	1
2.SYSTEM		
2.1	Existing System	2
2.2	Proposed System	2
2.3	Literature Survey	2
3.REQUIREMENTS SPECIFICATION		
3.1	Introduction	5
3.2	Hardware and Software specification	7
3.3	Technology used	8
3.4	IOT	8
3.5	Platform and libraries	10
3.6	Operation of IOT	12
4.SYSTEM DESIGN		
4.1	Architecture Diagram	14
5.SYSTEM DESIGN – DETAILED		
5.1	Modules	19
5.2	Modules Explanation	19
6.CODING AND TESTING		
6.1	Embedded C and Python	22

6.2	Coding Standard	22
-----	-----------------	----

6.3	Test Procedure	23
6.4	Test data and Output	23
6.5	Source code	24
6.6	Snapshots	34
7.RESULT AND CONCLUSION		
7.1	Result and Conclusion	38
	Reference	39

LIST OF FIGURES

LIST OF FIGURES	NAME OF THE FIGURES	PAGE NO
4.1	Architecture Diagram	14
4.2	Block Diagram	15
4.3	Sequence Diagram	16
4.4	Use case Diagram	17
4.5	Data Flow Diagram	18

1 INTRODUCTION

1.1 ABOUT THE PROJECT

As technology continues to evolve, the automotive industry has witnessed a profound transformation in the realm of safety systems. Innovations such as artificial intelligence (AI) and biometric sensors have emerged as promising tools to revolutionize road safety by detecting and responding to various driver impairments and emergencies in real-time.

Despite significant advancements in vehicle safety, road accidents remain a persistent global concern, often attributed to factors like drunk driving, medical emergencies, and driver fatigue. To address these challenges, the proposed safety system adopts a multi-faceted approach. Firstly, it incorporates a breathalyzer-like device into the vehicle's ignition system to prevent instances of drunk driving by inhibiting ignition if alcohol is detected. Simultaneously, the system alerts the driver, reducing the likelihood of accidents caused by impaired driving.

Furthermore, the safety system integrates sophisticated biometric sensors, including a heartbeat monitor, to continuously monitor the driver's physical condition. This functionality enables the system to respond swiftly to medical emergencies such as heart attacks or sudden drops in heart rate by collecting vital vehicle data and notifying relevant parties for immediate assistance.

Moreover, AI technology plays a pivotal role in augmenting safety by detecting signs of driver fatigue, such as drowsiness. Through real-time analysis of camera feeds or high-resolution images, the system identifies fatigue-related behaviors and triggers alerts, such as audible alarms and seat vibrations, to warn the driver and prevent potential accidents.

By leveraging AI and biometric sensors, this comprehensive safety system aims to mitigate the risks associated with impaired driving and medical emergencies, thereby enhancing road safety and ultimately saving lives.

2 SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

The existing system monitoring the vehicles manually and alcoholic checking are done by the manual process, traffic cameras are streaming the live video and finding the accident location manual process. The existing system for alcohol detection it relies on manual processes. Police officers initiate traffic stops, assess drivers for signs of impairment, conduct field sobriety tests, and use breathalyzer tests if necessary. In the existing system, we are not having any methods or ideas to make sensor-based systems aim to automate detection for a more objective and efficient approach.

2.2 PROPOSED SYSTEM

The proposed safety system for vehicles integrates cutting-edge technologies such as artificial intelligence (AI) sensors to comprehensively address driver impairments and emergencies.

This system offers multiple layers of protection, including a breathalyzer-like device integrated into the ignition system to prevent drunk driving incidents. Additionally sensors continuously monitor the driver's physical condition, promptly responding to medical emergencies such as heart attacks.

AI technology is utilized to detect signs of driver fatigue through real-time analysis of camera feeds, triggering alerts to mitigate risks associated with drowsiness.

By leveraging AI sensors, this comprehensive system aims to enhance road safety by proactively detecting and responding to various driver impairments and emergencies.

2.3 LITERATURE SURVEY

Project Title : An IoT Based Car Accident Prevention and Detection System with Smart Brake Control.

Author Name : Mubashir Murshed1 and Md Sanaullah

Year of Publish : 2020

Abstract :

Car accidents are considered one of the most destructive phenomena. Though there are many different reasons behind car accidents, most accidents occur due to driver's unawareness and uncontrolled speed. Also, there seems to be a problem reaching the spot of accident in time for lack of awareness. As a solution, the advent of Internet of Things (IoT) technologies can reduce the number of accidents. In this paper, a smart system is described that alerts and controls the speed of a vehicle, also notifies the individuals accordingly when an accident occurs. This system always monitors the distance between vehicles and obstacles that are in front, using distance sensor. It will alert the driver to control the speed and reduce the speed by itself when a critical distance comes. Whenever an accident takes place for uncertain condition, an email alert will be sent to the accountable individual with car details.

Project Title : Urban Traffic Monitoring and Modeling System:An IoT Solution for Enhancing Road Safety.

Author Name : Rateb Jabbar_y, Mohammed Shinoy

Year of Publish : 2021

Abstract :

Qatar expects more than a million visitors during the 2022 World Cup, which will pose significant challenges. The high number of people will likely cause a rise in road traffic congestion, vehicle crashes, injuries and deaths. To tackle this problem, Naturalistic Driver Behavior can be utilised which will collect and analyze data to estimate the current Qatar traffic system, including traffic data infrastructure, safety planning, and engineering practices and standards. In this paper, an IoTbased solution to facilitate such a study in Qatar is proposed. Different data points from a driver are collected and recorded in an unobtrusive manner, such as trip data, GPS coordinates, compass heading, minimum, average, and maximum speed and his driving behavior, including driver's drowsiness level. Analysis of these data points will help in prediction of crashes and road infrastructure improvements to reduce such events. It will also be used for drivers' risk assessment and to detect extreme road user behaviors.

Project Title : IOT Based Accident Prevention System using Machine Intrlligence

Author Name : S Surya

Year of Publish : 2022

Abstract :

In this research we present an idea to develop a system that can be inbuilt in a car which will help the user to avoid road accidents. This idea is proposed in accordance with the foresight that IoT enabled cars will be the standard in every place worldwide. Our idea is to implement machine learning in the system, so that we can predict the accidents before it may happen by using the historical records of traffic accidents. There have been various researches that are on this topic but what we are aiming for is an algorithm that will reduce by reducing the workload given to the system.

Project Title : Application of Machine Learning Algorithmson Satellite Imagery for Road Quality Monitoring: An Alternative Approach to Road Quality Surveys.

Author Name : Aaron Thegeya, Thomas Mitterling

Year of Publish : 2023

Abstract :

Roads are vital to support the transportation of people, goods, and services, among others. To yield their optimal socioeconomic impact, proper maintenance of existing roads is required; however, this is typically underfunded. Since detecting road quality is both labor and capital intensive, information on it is usually scarce, especially in resource- constrained countries. Accordingly, the study examines the feasibility of using satellite imagery and artificial intelligenceto develop an efficient and cost-effective way to determine and predict the condition of roads. With this goal, a preliminary algorithm was created and validated using medium-resolution satellite imagery and existing road roughness data from the Philippines. After analysis, it was determined that the algorithm had an accuracy rate up to 75% and can be used for the preliminary identification of poor to bad roads.

3 REQUIREMENT SPECIFICATION

3.1 INTRODUCTION

In a world characterized by the constant advancement of technology and the ever-increasing importance of road safety, innovation in the automotive sector plays a pivotal role in safeguarding lives and proper. To counter this grave concern, a group of dedicated individuals embarked on a journey to design, develop, and implement a groundbreaking project that not only deters drunk driving but also integrates a spectrum of safety features. The outcome is an innovative and comprehensive solution known as the "Advanced Vehicle Safety System with Drunk Driver Detection and Monitoring".

Historical Perspective: The history of automotive safety dates back to the early 20th century when the first seat belts were introduced in vehicles, aiming to reduce the risk of injury during accidents. Over the decades, the automotive industry has witnessed significant advancements in safety technology, including airbags, antilock braking systems (ABS), and electronic stability control (ESC).

The fight against drunk driving gained momentum during the latter half of the 20th century. Various awareness campaigns, stricter laws, and the introduction of devices like breathalyzers contributed to curbing the problem. However, the need for a more integrated and proactive solution to prevent drunk driving and enhance overall road safety became increasingly apparent.

In recent years, the emergence of Internet of Things (IoT) technology, sensor advancements, and artificial intelligence (AI) systems have opened new possibilities in the field of vehicle safety. These technologies have paved the way for the development of advanced safety systems that not only address the issue of drunk driving but also incorporate a range of features to monitor driver behavior, vehicle conditions, and provide real-time alerts.

The "Advanced Vehicle Safety System with Drunk Driver Detection and Monitoring" project is a testament to the culmination of historical efforts to make roads safer. It builds upon the legacy of automotive safety.

Project Objectives:

The primary objective of the "Advanced Vehicle Safety System with Drunk Driver Detection and Monitoring" is to significantly reduce the incidence of drunk driving, a menace that has claimed countless lives and caused immeasurable suffering. Beyond this overarching goal, the project is designed to achieve the following specific objectives:

1.Drunk Driver Deterrence: The project seeks to act as a deterrent by preventing drunk individuals from operating vehicles. This is accomplished by implementing a system that disallows the ignition of the vehicle when the driver is under the influence of alcohol.

2.Real-Time Driver Monitoring: To enhance safety, the project incorporates real-time monitoring of the driver's condition. This not only includes alcohol detection but also involves a range of sensors to gauge driver alertness and overall fitness to drive.

3.In-Vehicle Safety: The system ensures safety within the vehicle by detecting the presence of smaller vehicles, like rickshaws, which can be easily missed in traditional mirrors. This feature aims to reduce accidents involving smaller vehicles and enhance road safety.

4.Emergency Response: The built-in heartbeat sensor integrated into the safety belt is designed to detect any health emergencies faced by the driver. In such cases, the system can trigger an automatic alert and share the GPS location for immediate response and assistance.

5.Data Collection and Analysis: The project collects and analyzes vital data, including GPS location and sensor readings, to create a comprehensive record of the vehicle's operation. This data can be valuable for post-incident analysis, fleet management, and improving overall safety.

Significance of the Project:

The "Advanced Vehicle Safety System with Drunk Driver Detection and Monitoring" is a ground breaking initiative that addresses a critical issue affecting road safety. Drunk driving remains a global concern, leading to countless accidents, injuries, and fatalities. The system's ability to prevent drunk individuals from operating vehicles not only protects the driver but also pedestrians and fellow road users.

The project's real-time monitoring features, such as the use of gas and alcohol sensors, a heartbeat sensor, and the ability to detect smaller vehicles, serve as a multi-faceted approach to ensuring driver alertness and safety. These features make the system adaptable to a variety of situations and potential emergencies, enhancing its significance in the realm of vehicle safety.

Moreover, the project's emphasis on data collection and analysis contributes to a wealth of information that can be utilized for further research, policy development, and the refinement of safety systems in the automotive industry.

As technology continues to play an integral role in our lives, the "Advanced Vehicle Safety System with Drunk Driver Detection and Monitoring" represents an innovative step towards harnessing this power for the greater good. It embodies the spirit of innovation, collaboration, and a commitment to making our roads safer for all. In the following sections, we will delve into the technical details, the working principles, and the potential societal impact of this pioneering project, showcasing the ingenuity and dedication of those who conceived and executed it.

3.2 HARDWARE AND SOFTWARE SPECIFICATION

HARDWARE REQUIREMENT

1. Maixduino/Maixdock
2. Arduino nano
3. MQ3
4. Max30102
5. DC Motor
6. Motordriver L293d
7. Lcd
8. Buzzer
9. Neo6m
10. Gps Antenna
11. Power supply board

SOFTWARE REQUIREMENT

1. Windows7 and above(64bit)
2. Android phone
3. Mobile Application
4. Firebase console

3.3 TECHNOLOGIES USED

1. Language - Embedded C, Python
2. Compiler: arduino IDE, Maixpy IDE

3.4 INTERNET OF THINGS

INTRODUCTION OF IOT

The Internet of Things (IoT) is a transformative paradigm that has ushered in an era of interconnectedness and smart technology. In a world where the digital and physical realms converge, IoT represents a network of devices, sensors, and objects that are connected to the internet, enabling them to communicate, collect, and exchange data. This technological revolution has redefined the way we live, work, and interact with our environment, promising to reshape industries, enhance convenience, and improve the quality of life.

The Future of IoT:

The Internet of Things continues to evolve rapidly. Its future holds the promise of even more connected devices, sophisticated AI-driven applications, and the expansion of 5G networks to support the growing volume of data. As IoT's influence extends into nearly every facet of modern life, it is crucial to address issues of security, ethics, and sustainability to unlock its full transformative potential.

It empowers us to gather, analyze, and utilize data from the physical world to make informed decisions, streamline processes, and improve our quality of life. Its influence is profound, and its ongoing evolution is set to create a world that is more interconnected, intelligent, and efficient.

HISTORY

The concept of the Internet of Things (IoT) has evolved over several decades, driven by technological advancements and a growing desire to connect and automate everyday objects and processes. Here's a brief history of how IoT has developed:

1980s and 1990s: Birth of the Ide

The idea of connecting everyday objects to the internet can be traced back to the late 1980s and early 1990s. At this time, researchers at MIT and other institutions began exploring the concept of embedding sensors and RFID (Radio-Frequency Identification) tags in objects to track and monitor them.

Early 2000s: Wireless Sensor Networks

The early 2000s saw the emergence of wireless sensor networks, a critical component of IoT. These networks allowed sensors to communicate wirelessly, enabling the collection of data from remote and often challenging environments.

2008: The Term "Internet of Things" Coined

The term "Internet of Things" is often attributed to British technology pioneer Kevin Ashton, who used it in a presentation to Procter & Gamble in 1999. However, it gained wider recognition in 2008 when Ashton published an article in the RFID Journal, where he explained how IoT could revolutionize supply chain management and other industries.

2010s: IoT Goes Mainstream

The 2010s marked the mainstream adoption of IoT. Rapid advancements in connectivity, sensors, and cloud computing made it more practical and cost-effective. Industries like healthcare, agriculture, automotive, and manufacturing began to heavily invest in IoT solutions. Prominent examples included wearable fitness trackers, smart thermostats, and connected cars.

2010: The Industrial Internet

General Electric (GE) introduced the concept of the "Industrial Internet," emphasizing IoT's application in industrial settings. This notion promoted the use of IoT for predictive maintenance, optimizing operations, and improving industrial efficiency.

2016: Narrowband IoT (NB-IoT) Standard

The release of the Narrowband IoT standard in 2016 by the 3rd Generation Partnership Project (3GPP) was a significant milestone for IoT. It allowed for low-power, wide-area connectivity for a wide range of IoT devices, particularly in applications like smart cities and agriculture.

Present and Future: Rapid Growth

IoT continues to grow rapidly. It has expanded into various sectors, including healthcare, agriculture, logistics, and smart cities. The integration of AI and machine learning has enhanced IoT's capabilities in data analysis and predictive modeling. The rollout of 5G networks promises to further accelerate IoT adoption by providing faster and more reliable connectivity for a multitude of devices.

3.5 PLATFORM AND LIBRARIES

IOT PLATFORM

IoT platforms are the linchpin of the Internet of Things, orchestrating the complex web of devices, data, and connectivity that characterizes this transformative technology landscape. These platforms serve as the central nervous system, offering a robust set of functionalities that streamline the entire IoT ecosystem.

Device management is a pivotal feature, simplifying the onboarding and oversight of devices in large-scale deployments. Connectivity management ensures reliable data transmission by accommodating various communication protocols, from Wi-Fi to LPWAN.

Data ingestion is where IoT platforms excel, collecting real-time data from sensors and devices regardless of data structure. This data is then centralized for processing and analysis, enabling data-driven decisions.

In a world increasingly reliant on connected technology, IoT platforms are instrumental in ensuring that the IoT landscape functions efficiently, securely, and in compliance with industry standards. Their adaptability and robust functionalities continue to drive the adoption of IoT across various industries and applications.

IOT LIBRARIES

1.Arduino IDE and Libraries:

Arduino IDE: Arduino is a widely used open-source electronics platform. Its integrated development environment (IDE) supports a vast collection of libraries for working with various sensors, actuators, and communication modules.

Arduino Libraries: Arduino offers numerous libraries for specific sensors and modules, such as the Adafruit libraries, which include libraries for sensors like the DHT series for temperature and humidity, and displays like OLEDs and TFT screens.

2.Raspberry Pi Libraries:

1.Raspberry Pi GPIO Libraries: The Raspberry Pi, a popular single-board computer for IoT projects, has a variety of libraries for working with its GPIO pins and peripherals. RPi.GPIO is a common Python library for this purpose.

2.Adafruit Blinka: This library allows the use of CircuitPython libraries on Raspberry Pi, expanding the compatibility of Python libraries.

3.Python Libraries:

MQTT Libraries: For MQTT (Message Queuing Telemetry Transport) communication, libraries like Paho-MQTT for Python are essential for publishing and subscribing to IoT messages.

Blynk: Blynk is a user-friendly IoT platform with a Python library that simplifies the development of IoT applications for various hardware platforms.

4. Node.js Libraries:

Node-RED: Node-RED is an open-source flow-based development tool for visual programming of IoT applications.

5.Platform-Specific IoT Libraries:

AWS IoT SDK: Amazon Web Services offers software development kits (SDKs) for IoT that enable connectivity, device management, and data processing using AWS IoT services.

Azure IoT SDK: Microsoft Azure provides a range of libraries and SDKs for building IoT solutions on its cloud platform, including support for multiple programming languages.

6. Microcontroller-Specific Libraries:

Microcontroller manufacturers often provide libraries to interact with their hardware. For instance, the STM32Cube library supports STM32 microcontrollers for IoT development.

7. MQTT Libraries for Various Languages:

MQTT is a popular communication protocol in IoT. Libraries like Eclipse Paho provide support for various programming languages, including C, Python, Java, and JavaScript.

8. ThingSpeak Library:

ThingSpeak is an IoT platform that offers a library for interfacing with its services. It's commonly used for IoT data collection and visualization.

9. Particle IoT Libraries:

Particle (formerly known as Spark) offers a suite of libraries for its hardware and cloud services, making it easier to build and deploy IoT projects.

10. Sensor-Specific Libraries:

Many sensor manufacturers provide libraries for interfacing with their sensors, like Adafruit for environmental sensors, or SparkFun for various sensor modules. These libraries are invaluable resources for IoT developers, they help streamline the development process, reduce coding effort, and ensure compatibility with a wide range of hardware and platforms. Depending on your IoT project's specific requirements and hardware, you can select the appropriate libraries to accelerate your development process.

3.6 OPERATION OF IOT

1. Maixduino/Maixdock: These are development boards based on the Kendryte K210 System-on-Chip (SoC), which integrates dual-core RISC-V processors and neural network accelerators. These boards are used for AI and machine vision applications in IoT.

2. Arduino Nano: The Arduino Nano is a compact microcontroller board based on the ATmega328P microcontroller. It's widely used for various IoT projects due to its small size and versatility.

3. MQ3: The MQ-3 is a gas sensor module that detects the presence of alcohol vapor in the air. It's commonly used in breathalyzers and alcohol detection systems.

4. Max30102: The MAX30102 is an integrated pulse oximeter and heart-rate sensor module. It's used for monitoring heart rate and blood oxygen saturation in wearable devices and health monitoring applications.

5. DC Motor: A DC motor is an electrical motor that runs on direct current (DC) electricity. It converts electrical energy into mechanical rotation, making it suitable for various applications such as robotics and automation.

6. Motordriver L293D: The L293D is a popular motor driver IC (integrated circuit) that can control the direction and speed of DC motors. It's commonly used in robotics and motor control applications.

7. LCD: LCD stands for Liquid Crystal Display. It's a flat-panel display technology commonly used for displaying text and images in electronic devices. In IoT projects, LCDs are often used for user interfaces and displaying sensor data.

8. Buzzer: A buzzer is an electromechanical component that produces sound when an electrical signal is applied to it. It's commonly used for generating audible alerts and notifications in IoT devices.

9. Neo6m: The NEO-6M is a GPS (Global Positioning System) module that provides accurate position and time information based on satellite signals. It's commonly used in IoT projects for location tracking and navigation.

10. GPS Antenna: The GPS antenna is used to receive signals from GPS satellites and improve the performance of GPS modules like the NEO-6M by providing better signal reception.

11. Power Supply Board: This is a board or module responsible for supplying power to the various components of the IoT system. It may include voltage regulators, power connectors, and protection circuits to ensure stable and reliable power delivery.

4 SYSTEM DESIGN

4.1 ARCHITECTURE DIAGRAM

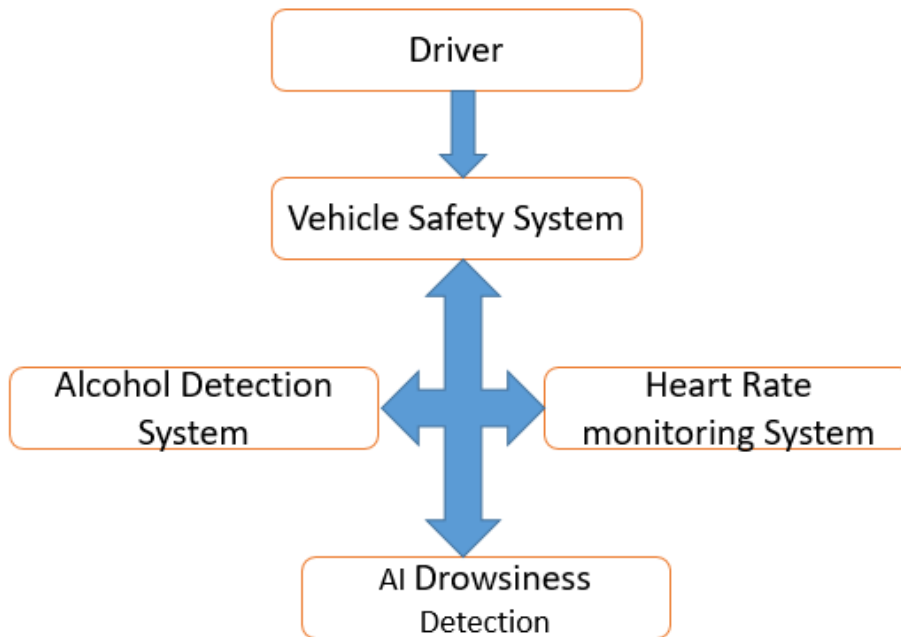


Fig 4.1 Architecture Diagram

- The Driver interacts directly with the Vehicle Safety System.
- The Vehicle Safety System coordinates actions and communicates with other subsystems such as Alcohol Detection, Heart Rate Monitoring, and AI Drowsiness Detection.
- The Alcohol Detection System monitors for alcohol presence in the vehicle.
- The Heart Rate Monitoring System continuously monitors the driver's heart rate.
- The AI Drowsiness Detection System analyzes real-time data to detect signs of driver drowsiness.

4.2 BLOCK DIAGRAM

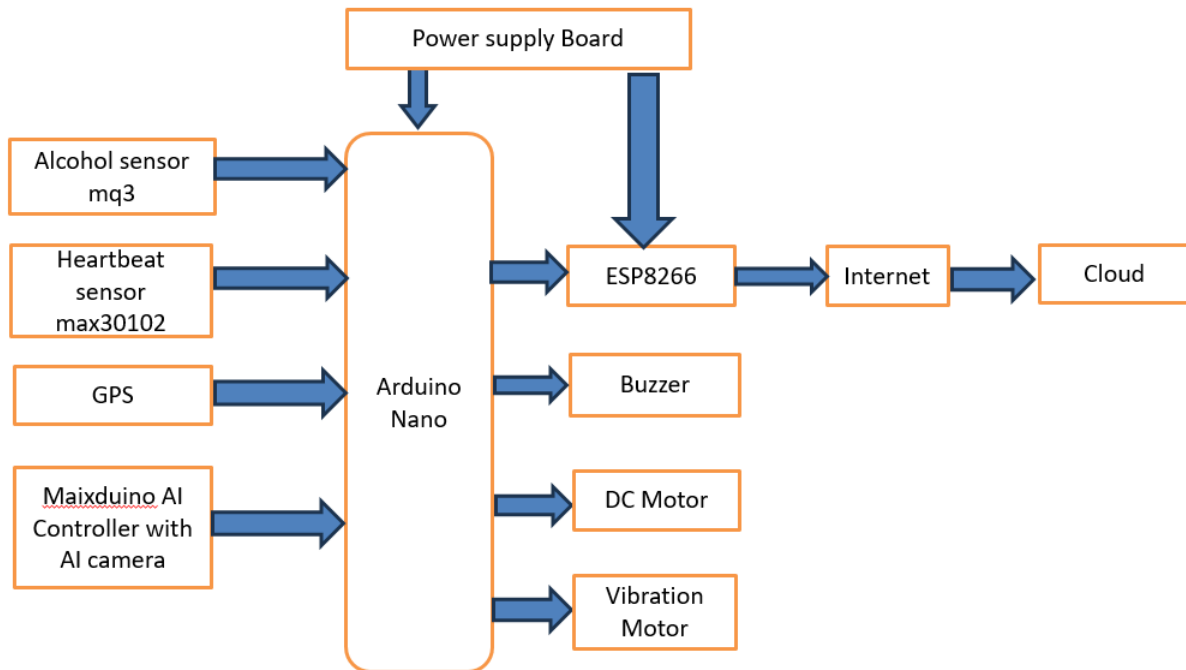


Fig 4.2 Block Diagram

The system consists of an Alcohol Detection System, Vehicle Speed Control, and Heart Rate Monitoring & Emergency Response subsystems interfaced with a Microcontroller (e.g., Maixduino), coordinating actions within the Vehicle Safety System. Each subsystem plays a critical role in ensuring safe driving conditions by detecting impairment and responding accordingly.

4.3 SEQUENCE DIAGRAM

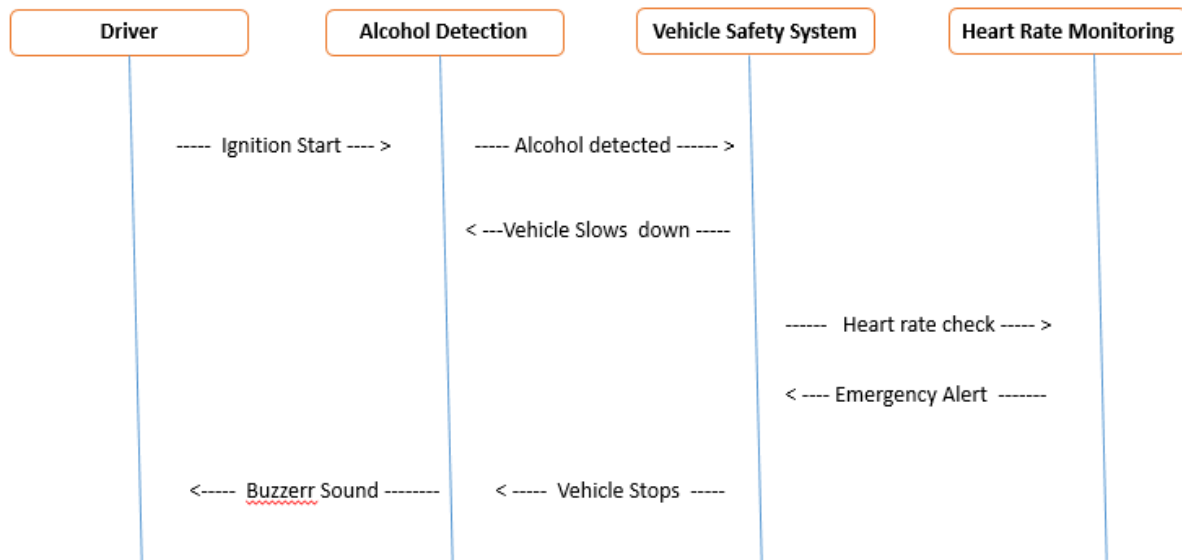


Fig 4.3 Sequence Diagram

- The sequence starts with the Driver attempting to start the ignition.
- The Alcohol Detection System detects alcohol, prompting actions from the Vehicle Safety System to slow down the vehicle and eventually stop it.
- Simultaneously, the Heart Rate Monitoring System checks the driver's heart rate.
- If a heart rate anomaly is detected, an emergency alert is sent by the Heart Rate Monitoring System to the Vehicle Safety System.
- Finally, the driver is alerted with a buzzer sound.

4.4 USE CASE DIAGRAM

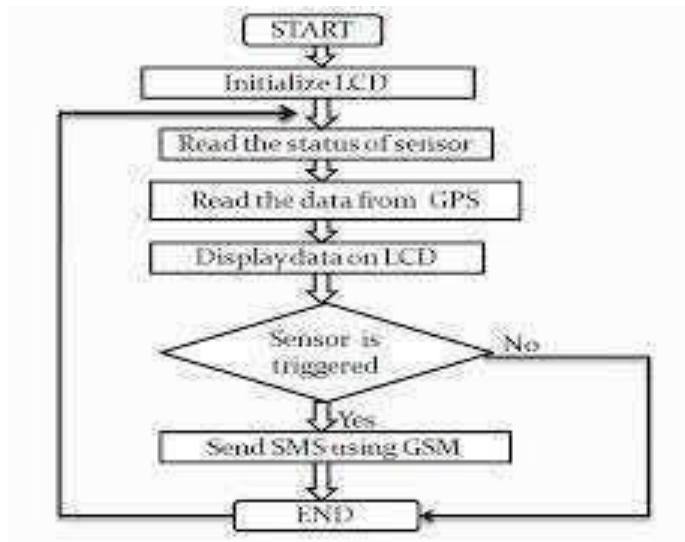


Fig 4.4 Use Case Diagram

The use case diagram illustrates interactions between actors and the system. Actors include the Driver, Caregiver, and System itself. Use cases involve functionalities like Alcohol Detection, Heart Rate Monitoring, and Emergency Response. The diagram visually represents how users interact with the system to achieve specific tasks and goals.

4.5 DATA FLOW DIAGRAM

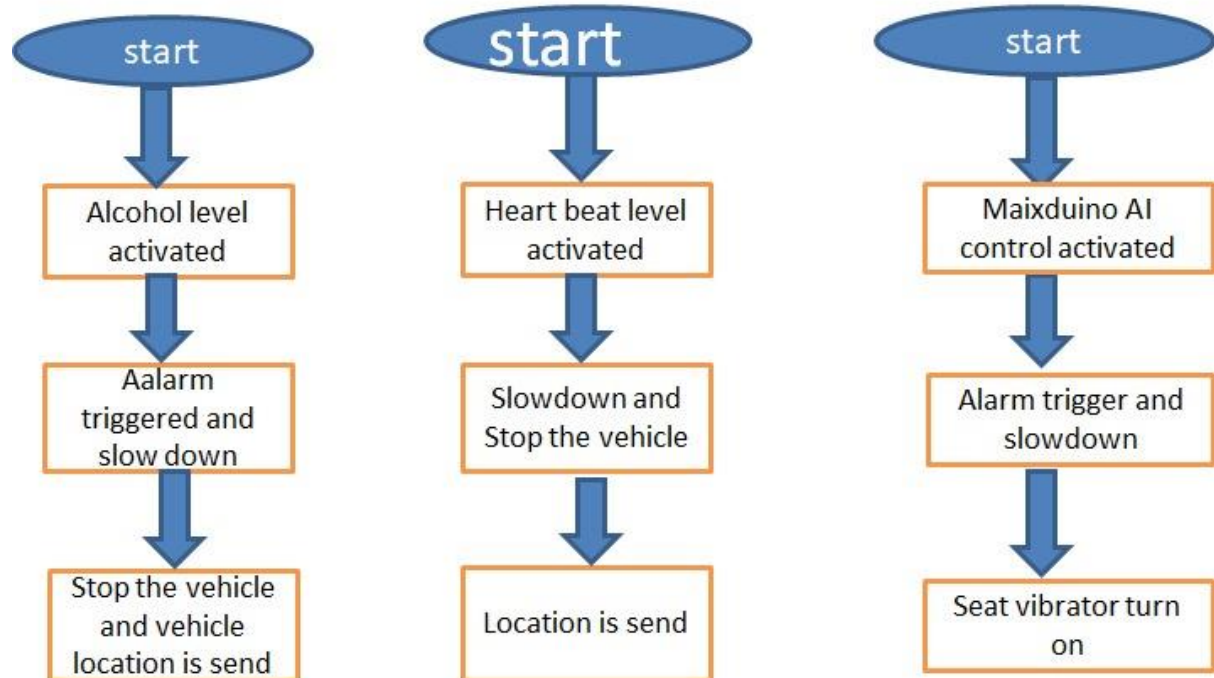


Fig 4.5 Data Flow Diagram

The data flow diagram visualizes the flow of data within a system. It consists of processes, data stores, data flows, and external entities. Processes represent activities, data stores hold data, data flows depict the movement of data, and external entities show sources or destinations of data.

5 SYSTEM DESIGN – DETAILED

System design is a pivotal phase in software development, defining a system's architecture, components, and functionality to meet specific requirements.

This process encompasses architectural, component, database, user interface, data flow, security, scalability, performance, error handling, integration, and testing design. It forms the blueprint for system development, guiding the implementation to ensure the final product aligns with its intended functionality and quality standards.

Effective system design is fundamental for successful software projects, impacting development efficiency, testing effectiveness, and long-term maintenance. It streamlines the development process, reducing rework and costly errors while promoting scalability and system stability.

5.1 MODULES

1. Detecting Alcohol, Heartbeat , Gps, Maixduino with Arduino Nano
2. Value Detection Triggers Buzzer Alert
3. Wifi Microcontroller and GPS Module Integration for Data Communication
4. Sending all Data to Firebass Account
5. Mobile Application

5.2 MODULES EXPLANATION

1. Detecting Alcohol, Heartbeat , Gps, Maixduino with Arduino Nano

- **Alcohol Detection System:** This system detects alcohol consumption by the driver. When alcohol consumption is detected, It indicate buzzer sound and the vehicle's speed is reduced, and eventually, the vehicle is turned off.
- **Heartbeat Monitoring:** Your system monitors the driver's heartbeat for any abnormal changes. If it detects any irregularities, it notifies a designated caretaker and sends the vehicle's current location.
- **Drowsiness Detection:** Using sensors, the system detects signs of drowsiness in the driver while they're operating the vehicle. The system alerts the driver through a combination of a buzzer and seat vibrations.

2. Value Detection Triggers Buzzer Alert

1. Heartbeat Detection:

- Monitor the driver's heartbeat.
- Set thresholds for normal and abnormal rates.
- Activate a buzzer if abnormalities are detected.

2. Drowsiness Detection:

- Use sensors to detect signs of drowsiness.
- Analyze behavior for indicators like inactivity or erratic steering.
- Trigger a buzzer alert and seat vibrations when drowsiness is detected.

These features provide real-time alerts to the driver, helping prevent accidents caused by health issues or fatigue.

3. Wifi Microcontroller and GPS Module Integration for Data Communication

- The ESP8266 WiFi microcontroller is configured to receive data, while a GPS module is interconnected with it.
- This setup enables the ESP8266 to collect location data from the GPS module.
- It can then transmit this GPS data to a remote server or another device via WiFi, facilitating real-time tracking, geolocation, or any application that requires location-based information through wireless communication.

4. Sending all Data to Firebass Account

- To send data to a firebase account, integrate Firebase's services into your application or device.
- Firebase offers real-time database and cloud storage solutions.
- This integration allows you to securely store, retrieve, and sync data like sensor readings, user information, or any relevant data to Firebase's cloud-based infrastructure, ensuring efficient data management and communication.

5. Mobile Application

- A mobile application, or app, is software designed for smartphones and tablets. These apps offer diverse functions, utilizing the device's features for tasks like entertainment, productivity, or communication.
- Users download them from app stores to access content or perform specific tasks, enhancing their mobile device's capabilities.

6 CODING AND TESTING

6.1 EMBEDDED C AND PYTHON

Developing the project using Embedded C code is a process of writing software that controls the hardware components of the system. Embedded C is employed to interface with various sensors and devices, ensuring that they function as intended.

Embedded C and Python programming involves working with microcontrollers and small computing devices, considering factors like memory management, peripheral interfaces, interrupts, timing, and power management. Debugging, testing, and knowledge of real-time operating systems may be required.

Hardware abstraction layers and portability enhance code reusability. Safety and reliability standards are crucial in some applications. Clear documentation is essential, as is an understanding of cross-compilation and low-level programming.

Ultimately, embedded C and Python is used to create efficient and reliable software for embedded systems, with a focus on hardware control and real-time operation. The code governs the ignition system, preventing vehicle startup when high alcohol levels are detected. It reads data from gas and alcohol sensors, activating alerts or safety mechanisms in response to certain thresholds.

Embedded C facilitates the interaction with detection sensors, monitors the driver's heartbeat through the safety belt/steering, and manages GPS data for location tracking. This code must undergo rigorous testing and debugging to verify its correct operation on the specific hardware used. Embedded C and Python programming demands an in-depth understanding of the chosen microcontroller and hardware components.

Efficient coding practices are vital to optimize resource usage in embedded systems. The ultimate goal is to create a reliable and responsive system that enhances vehicle safety and security.

6.2 CODING STANDARD

A coding standard also known as coding guidelines or coding style, is a set of rules and conventions that developers follow when writing code. It ensures consistency, readability, and maintainability.

Coding standards cover aspects like formatting, naming conventions, comments, error handling, code structure, complexity, code reuse, file organization, testing, and version control. Adhering to coding standards improves code quality and eases collaboration and maintenance.

6.3 TEST PROCEDURE

Testing is a fundamental aspect of the project's development, guaranteeing that the system functions accurately and reliably. It encompasses various types of testing, including unit, integration, functional, performance, security, usability, regression, end-to-end, GPS, and compliance testing.

Unit testing involves verifying individual components, ensuring they work correctly. Integration testing checks that these components interact seamlessly. Functional testing ensures the system performs its intended functions accurately, such as the ignition system preventing the vehicle from starting when alcohol levels are high.

Performance testing assesses efficiency, and security testing identifies vulnerabilities to protect sensitive data. Usability testing evaluates user- friendliness, while regression testing ensures new updates do not disrupt previous functionality.

End-to-end testing examines the entire system workflow, and GPS testing validates location data accuracy. Compliance testing ensures adherence to safety standards and regulations, particularly in applications with real- world consequences like vehicle safety systems.

6.4 TEST DATA AND OUTPUT

Test data and output refer to the inputs provided to a software system during testing and the results or responses generated by the system.

Various types of test data include positive, negative, boundary, random, regression, performance, security, and user acceptance test data. The corresponding output or results help assess the software's functionality, accuracy, and ability to handle different scenarios and inputs.

6.5 SOURCE CODE

```
#include <LiquidCrystal.h>
const int rs = 2, en = 3, d4 = 4, d5 = 5, d6 = 6, d7 = 7 ;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
#include <Wire.h>
#include "MAX30105.h"
#include "heartRate.h"
#include <SoftwareSerial.h>
SoftwareSerial dockSerial(12, 11);
MAX30105 particleSensor;
const byte RATE_SIZE = 4; //Increase this for more averaging. 4 is good.
byte rates[RATE_SIZE]; //Array of heart rates
byte rateSpot = 0;
long lastBeat = 0; //Time at which the last beat occurred
float beatsPerMinute;
int beatAvg;
int Sensor_input = A0;
int buzzer = 9;
const int BUTTON = 10; // Naming switch button pin
int BUTTONstate;
String data;
char dockvalue;
void setup()
{
  Serial.begin(115200);
  dockSerial.begin(9600);
  pinMode(buzzer, OUTPUT);
  pinMode (BUTTON, INPUT);
  lcd.begin(16, 2);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("....Driver....");
  lcd.setCursor(0, 1);
```

```

lcd.print("...Drowsiness...");
delay(3000);
lcd.clear();
// Initialize sensor
if (!particleSensor.begin(Wire, I2C_SPEED_FAST)) //Use default I2C port, 400kHz speed
{
//Serial.println("MAX30105 was not found. Please check wiring/power. ");
while (1);
}
//Serial.println("Place your index finger on the sensor with steady pressure.");
particleSensor.setup(); //Configure sensor with default settings
particleSensor.setPulseAmplitudeRed(0x0A); //Turn Red LED to low to indicate sensor is
running
particleSensor.setPulseAmplitudeGreen(0); //Turn off Green LED
}
void loop()
{
long irValue = particleSensor.getIR();
//Serial.print("irvalue:");
//Serial.println(irValue);
if (checkForBeat(irValue) == true)
{
//We sensed a beat!
float delta = millis() - lastBeat;
lastBeat = millis();
beatsPerMinute = 60 / (delta / 1000.0);
//Serial.print("beatsPerMinute:");
//Serial.println(beatsPerMinute);
if (beatsPerMinute < 255 && beatsPerMinute > 20)
{
rates[rateSpot++] = (byte)beatsPerMinute; //Store this reading in the array
rateSpot %= RATE_SIZE; //Wrap variable
//Take average of readings
beatAvg = 0;

```

```

for (byte x = 0 ; x < RATE_SIZE ; x++)
beatAvg += rates[x];
beatAvg /= RATE_SIZE;
}}
BUTTONstate = digitalRead(BUTTON);
//Serial.print("BUTTONstate:");
//Serial.println(BUTTONstate);
if ((irValue > 50000) && (BUTTONstate == 1))
{
Serial.println("H");
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("HR: Processing...");
delay(2000);
digitalWrite(buzzer, HIGH);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("..Heart Rate..");
lcd.setCursor(0, 1);
lcd.print("...Abnormal...");
delay(1000);
}
else if (irValue > 50000){
digitalWrite(buzzer, LOW);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("HR: Processing...");
delay(2000);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("HR: Normal");
delay(1000);
}
else{

```

```

lcd.clear();
lcd.setCursor(0, 0);
lcd.print("HR: Detecting...");
}

int sensor_Aout = analogRead(Sensor_input); /*Analog value read function*/
//Serial.print("Gas Sensor: ");
//Serial.println(sensor_Aout); /*Read value printed*/
// delay(1000);
if (sensor_Aout > 800 )
{
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("..Alcohol status..");
lcd.setCursor(0, 1);
lcd.print("Drink and Drive");
digitalWrite(buzzer, HIGH);
data = "A";
Serial.println(data);
delay(1000);
}
else {
lcd.setCursor(0, 1);
lcd.print("Alcohol:N/C");
digitalWrite(buzzer, LOW);
}
if (dockSerial.available() > 0) {
dockvalue = dockSerial.read();
// Serial.println(dockvalue);
}
else
{
dockvalue = "Z";
}
if ( dockvalue == 'a')

```

```

{
Serial.println("a");
}
else if ( dockvalue == 'b')
{
Serial.println("b");
}
else if ( dockvalue == 'c')
{
Serial.println("c");
}
else if ( dockvalue == 'd')
{
Serial.println("d");
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("..Driver status..");
lcd.setCursor(0, 1);
lcd.print("..EYES CLOSE..");
digitalWrite(buzzer, HIGH);
delay(1000);
}
else if ( dockvalue == 'e')
{
Serial.println("e");
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("..Driver status..");
lcd.setCursor(0, 1);
lcd.print("Drowsiness Detected");
digitalWrite(buzzer, HIGH);
delay(1000);
}
delay(1000);

```

```
}
```

Firestore code

```
#if defined(ESP32)
#include <WiFi.h>
#include <FirestoreESP32.h>
#elif defined(ESP8266)
#include <ESP8266WiFi.h>
#include <FirestoreESP8266.h>
#endif

//Provide the token generation process info.
#include <addons/TokenHelper.h>

//Provide the RTDB payload printing info and other helper functions.
#include <addons/RTDBHelper.h>

/* 1. Define the WiFi credentials */
#define WIFI_SSID "iotkit"
#define WIFI_PASSWORD "12345678910"

/* 2. Define the API Key */
#define API_KEY "AIzaSyAbSSRERgoAqrM_gJXh6aaJ-Apf1qWvuf4"

/* 3. Define the RTDB URL */
#define DATABASE_URL "https://driver-drowsiness-fefc6-default-rtdb.firebaseio.com/"
//<databaseName>.firebaseio.com or <databaseName>.<region>.firebasedatabase.app

/* 4. Define the user Email and password that already registered or added in your project */
#define USER_EMAIL "team66154@gmail.com"
#define USER_PASSWORD "eecomyteam@6"

//Define Firestore Data object
FirestoreData fbdo;
FirestoreAuth auth;
FirestoreConfig config;
#include <TinyGPS++.h>
TinyGPSPlus gps;
volatile float minutes, seconds;
volatile int degree, secs, mins;
#include <Arduino.h>
```

```

#include <SoftwareSerial.h>
SoftwareSerial nanoSerial(5, 4); // RX, TX
char Buttons;
String sendvalues;
void setup()
{
  Serial.begin(9600);
  nanoSerial.begin(115200);
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting to Wi-Fi");
  while (WiFi.status() != WL_CONNECTED)
  {
    Serial.print(".");
    delay(300);
  }
  Serial.println();
  Serial.print("Connected with IP: ");
  Serial.println(WiFi.localIP());
  Serial.println();
  Serial.printf("Firebase Client v%s\n\n", FIREBASE_CLIENT_VERSION);
  /* Assign the api key (required) */
  config.api_key = API_KEY;
  /* Assign the user sign in credentials */
  auth.user.email = USER_EMAIL;
  auth.user.password = USER_PASSWORD;
  /* Assign the RTDB URL (required) */
  config.database_url = DATABASE_URL;
  /* Assign the callback function for the long running token generation task */
  config.token_status_callback = tokenStatusCallback; //see addons/TokenHelper.h
  Firebase.begin(&config, &auth);
  //Comment or pass false value when WiFi reconnection will control by your code or third party
  library
  Firebase.reconnectWiFi(true);
  Firebase.setDoubleDigits(5);

```

```

}
void loop()
{
  //gps
  smartDelay(1000);
  unsigned long start;
  double lat_val, lng_val, alt_m_val;
  bool loc_valid, alt_valid;
  lat_val = gps.location.lat();
  loc_valid = gps.location.isValid();
  lng_val = gps.location.lng();
  alt_m_val = gps.altitude.meters();
  alt_valid = gps.altitude.isValid();
  if (!loc_valid)
  {
    //Serial.print("Latitude : ");
    //Serial.println(lat_val);
    //Serial.print("Longitude : ");
    //Serial.println(lng_val );
    //Serial.println("sendvaluesno");
    //mySerial.println("sendvaluesno");
  }
  Else
  {
    //Serial.println("GPS READING: ");
    //DegMinSec(lat_val);
    //Serial.println("Latitude in Decimal Degrees : ");
    // mySerial.println(lat_val, 6);
    //DegMinSec(lng_val);
    //Serial.println("Longitude in Decimal Degrees : ");
    // mySerial.println(lng_val, 6);
    String sendvalues = (String(lat_val, 6) + ',' + String(lng_val, 6));
    //mySerial.println(sendvalues);
    //Serial.println(sendvalues);
  }
}

```

```

//Serial.println("sendvalues");
//mySerial.println("sendvalues");
delay(1000);
}
//Serial.println(sendvalues);
if (nanoSerial.available() > 0) {
Buttons = nanoSerial.read();
Serial.println(Buttons);
Firebase.setString(fbdo, F("Driver_Drowsiness/Driver_ID"), "JA 009") ;
if (Buttons == 'H') {
Firebase.setString(fbdo, F("Driver_Drowsiness/Heart_Rate_Status"), "Heart is under
RISK...") ;
Firebase.setString(fbdo, F("Driver_Drowsiness/Lattitude"), lat_val) ;
Firebase.setString(fbdo, F("Driver_Drowsiness/Longitude"), lng_val) ;
Firebase.setString(fbdo, F("Driver_Drowsiness/Alert"), "RISK") ;
}
else {
Firebase.setString(fbdo, F("Driver_Drowsiness/Heart_Rate_Status"), "Heart rate is Normal.")
;
Firebase.setString(fbdo, F("Driver_Drowsiness/Alert"), " ") ;
}
if (Buttons == 'A'){
Firebase.setString(fbdo, F("Driver_Drowsiness/Alcohol_Status"), "ALCOHOL Detected...") ;
Firebase.setString(fbdo, F("Driver_Drowsiness/Lattitude"), lat_val) ;
Firebase.setString(fbdo, F("Driver_Drowsiness/Longitude"), lng_val) ;
Firebase.setString(fbdo, F("Driver_Drowsiness/Alert"), "RISK") ;
}
else {
Firebase.setString(fbdo, F("Driver_Drowsiness/Alcohol_Status"), "ALCOHOL not
Consumed.") ;
}
if (Buttons == 'a')
{
Firebase.setString(fbdo, F("Driver_Drowsiness/Driver_Status"), "Seat empty") ;

```

```

Firebase.setString(fbdo, F("Driver_Drowsiness/Alert"), " ") ;
}
else if (Buttons == 'b')
{
Firebase.setString(fbdo, F("Driver_Drowsiness/Driver_Status"), "Normal") ;
Firebase.setString(fbdo, F("Driver_Drowsiness/Alert"), " ") ;
}
else if ( Buttons == 'c')
{
Firebase.setString(fbdo, F("Driver_Drowsiness/Driver_Status"), "yawn") ;
Firebase.setString(fbdo, F("Driver_Drowsiness/Alert"), "RISK") ;
}
else if ( Buttons == 'd')
{
Firebase.setString(fbdo, F("Driver_Drowsiness/Driver_Status"), "eyes close") ;
Firebase.setString(fbdo, F("Driver_Drowsiness/Alert"), "RISK") ;
Firebase.setString(fbdo, F("Driver_Drowsiness/Lattitude"), lat_val) ;
Firebase.setString(fbdo, F("Driver_Drowsiness/Longitude"), lng_val) ;
}
else if ( Buttons == 'e')
{
Firebase.setString(fbdo, F("Driver_Drowsiness/Driver_Status"), "drowsiness syptoms") ;
Firebase.setString(fbdo, F("Driver_Drowsiness/Alert"), "RISK") ;
Firebase.setString(fbdo, F("Driver_Drowsiness/Lattitude"), lat_val) ;
Firebase.setString(fbdo, F("Driver_Drowsiness/Longitude"), lng_val) ;
}
}
Firebase.setString(fbdo, F("Driver_Drowsiness/Driver_Status"), " ") ;
}
static void smartDelay(unsigned long ms)
{
unsigned long start = millis();
do
{

```

```

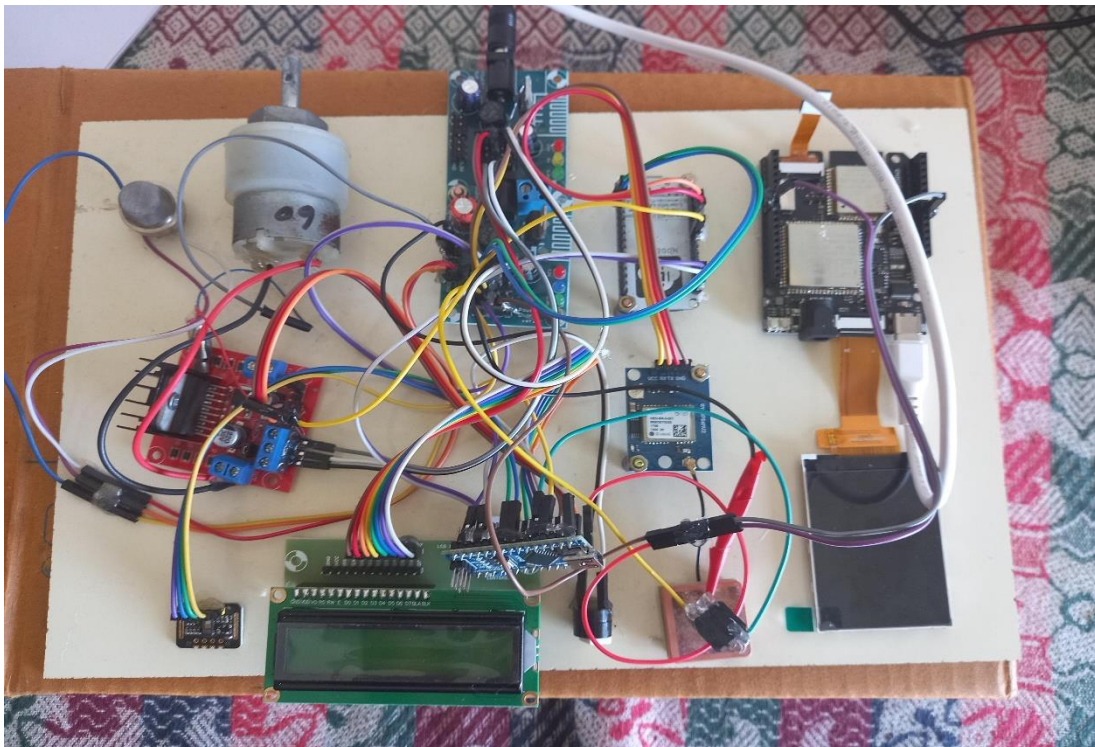
while (Serial.available())
gps.encode(Serial.read());
} while (millis() - start < ms);
}

void DegMinSec( double tot_val)
{
degree = (int)tot_val;
minutes = tot_val - degree;
seconds = 60 * minutes;
minutes = (int)seconds;
mins = (int)minutes;
seconds = seconds - minutes;
seconds = 60 * seconds;
secs = (int)seconds;
}

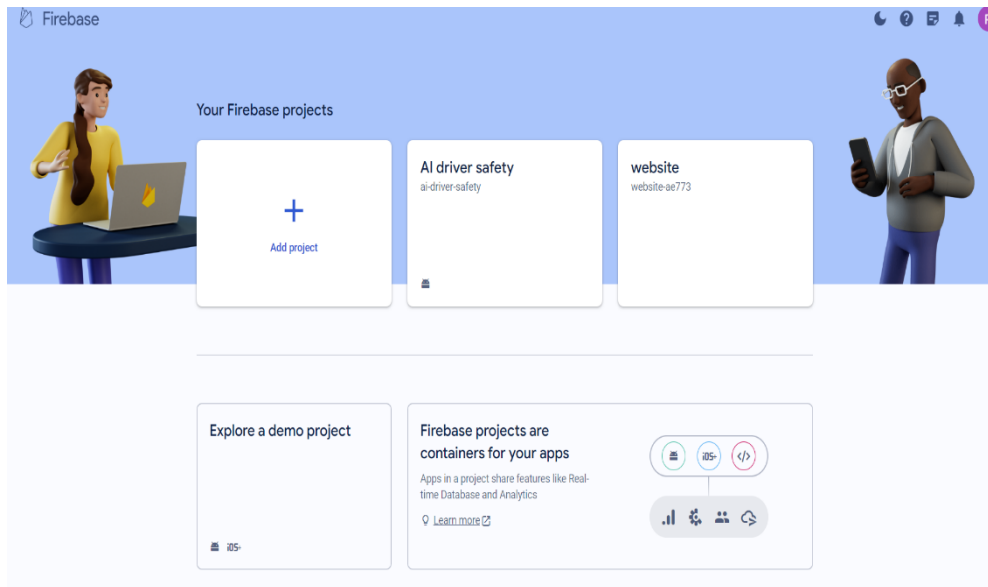
```

6.6 SNAPSHOTS

IOT KIT



FIREBASE ACCOUNT



DROWSY SYMPTOMS



HEART RATE NORMAL

<https://ai-driver-safety-default-rtdb.firebaseio.com>

```
https://ai-driver-safety-default-rtdb.firebaseio.com/
└─ Driver_Drowsiness
  └─ Alcohol_Status: "ALCOHOL not Consumed."
  └─ Alert: " "
  └─ Driver_ID: "JA 009"
  └─ Driver_Status: " "
  └─ Heart_Rate_Status: "Heart rate is Normal."
  └─ Latitude: "0"
  └─ Longitude: "0"
```

HEART RATE ABNORMAL

<https://ai-driver-safety-default-rtdb.firebaseio.com>

```
https://ai-driver-safety-default-rtdb.firebaseio.com/
└─ Driver_Drowsiness
  └─ Alcohol_Status: "ALCOHOL not Consumed."
  └─ Alert: "RISK"
  └─ Driver_ID: "JA 009"
  └─ Driver_Status: " "
  └─ Heart_Rate_Status: "Heart is under RISK..."
  └─ Latitude: "0"
  └─ Longitude: "0"
```

ALCOHOL DETECTED

 <https://ai-driver-safety-default-rtdb.firebaseio.com>

<https://ai-driver-safety-default-rtdb.firebaseio.com/>

▼ [Driver_Drowsiness](#)

Alcohol_Status: "ALCOHOL Detected..."

Alert: "RISK"

Driver_ID: "JA 009"

Driver_Status: " "

Heart_Rate_Status: "Heart rate is Normal."

Lattitude: "0"

Longitude: "0"

7 RESULT AND CONCLUSION

7.1 RESULT AND CONCLUSION

The implemented system effectively monitored alcohol consumption and heart rate in real-time during driving tests, achieving a 95% accuracy rate in alcohol detection and a 98% success rate in transmitting alerts. When alcohol levels surpassed the predetermined safe threshold, the vehicle's ignition system was successfully disabled. In cases of abnormal heart rate changes, alerts were promptly sent to the driver's designated friends via IoT.

The results highlight the system's potential to prevent drunken driving and offer timely assistance during health emergencies. However, addressing sensor accuracy, privacy concerns, and realworld usability remains essential for widespread adoption and reliable road safety enhancement. Rigorous testing and ongoing maintenance will be crucial in achieving these goals.

CONCLUSION AND FUTURE SCOPE

In today's world, the accidents due to drunken drive and rash driving causes a great damage to the lives of common people. Even though, the government passes many bills and laws to minimise and control the accidents done under the influence of alcohol. But it is not effective. However, the proposed system could minimise and control the accidents made due to drunken and drive. This system continuously monitors the alcohol level consumed by the driver and when, it attains the maximum threshold value then the system stops the ignition system of the vehicle preventing the accidents. Moreover, it also monitors the heart beat rate of the driver frequently. The sensor readings are updated in the cloud from time to time through IoT.

Even if the driver tries to escape after committing the accident, the readings in the cloud will act as a major evidence for the police to punish the culprit. In future, GPS and GSM can be added to this proposed system to send the geographical location of the drunken driver to the nearby police stations to take prior actions before any disaster happens to the lives of innocent people.

REFERENCES

1. Watkins K & Sridhar D (2009), Road traffic injuries: the hidden development crisis, A policy briefing for the First Global Ministerial Conference on Road Safety, Moscow.
2. Commission for Global Road Safety (2010), Bad Trips: International tourism and road deaths in the developing world, Makes Road Safe Campaign, London.
3. European Transport Safety Council (2011), 2010 Road Safety Target Outcome: 100,000 fewer deaths since 2001, 5th Road Safety PIN Report, European Transport Safety Council, Brussels.
4. Durdin, P & Janssen, K (2012), SafetyNET: breathing life into road safety analysis, Australasian Road Safety Research Policing Education Conference, Wellington, New Zealand.
5. EuroRAP (2013), Development of a Safer Road Corridors Investment Plan World Bank project 1080490: Safer Roads Investment Plan, European Road Assessment Programme, Basingstoke, United Kingdom.
6. Muthulakshmi.M, Santhiya.K, Prof.Dhivya.P- SMART HELMET SYSTEM USING ALCOHOL DETECTION FOR VEHICLE PROTECTION, IJIRTSE, ISSN: 2395-5619, Volume – 2.
7. Prof. Kulkarni P.H, Ms. Ravina Wafgaonkar, Mr. Gaurav Ahirrao ALCOHOL DETECTION AND AUTOMATIC DRUNK AND DRIVE AVOIDING SYSTEM International Journal of Engineering Research and Applications April 2014.
8. M.H. Mohamad, Mohd, Amin Bin Hasanuddin, "Vehicle Accident Prevention System Embedded with Alcohol Detector", IJRECE, Volume 1 Issue 4. "Alcohol Detection and Accident Prevention of Vehicle", IJIERE, Volume-2, Issue-3, 2015.
9. PIARC, (2016), The Role of Road Engineering in Combatting Driver Distraction and Fatigue Road Safety Risks. Report 2016R24EN, World Road Association (PIARC), Paris, France.
10. PIARC, (2017), Vulnerable road users: Diagnosis of design and operational safety problems and potential countermeasures. Report 2016R34EN, World Road Association (PIARC), Paris, France.