

**ENHANCED SECURITY METHOD FOR IOT BASED VEHICLE CRASH
RESPONSE SYSTEM**

Submitted in partial fulfillment of the requirements for the award of Bachelor of
Engineering Degree in Electronics and Communication Engineering

By

40130215 - SHAURYA

40130233 - THALLAPALLI GAYATHRI



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
SCHOOL OF ELECTRICAL AND ELECTRONICS**

SATHYABAMA

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

CATEGORY -1 UNIVERSITY BY UGC

**Accredited with Grade "A++" by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119**

APRIL – 2024



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

Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

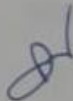
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of SHAURYA (40130215) and THALLAPALLI GAYATHRI (40130233) who carried out the project entitled "ENHANCED SECURITY METHOD FOR IOT BASED VEHICLE CRASH RESPONSE SYSTEM" under the supervision from November 2023 to April 2024.


Internal Guide

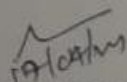
DR. P. KAVIPRIYA, M.E., Ph. D.,
Professor, Department of ECE

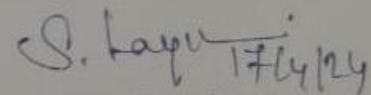


HEAD OF THE DEPARTMENT

Dr. T. RAVI, M.E., Ph. D.,

Submitted for Viva voce Examination held on 17 APR 2024


Internal Examiner


External Examiner

DECLARATION

We **SHAURYA (40130215)** and **THALLAPALLI GAYATHRI (40130233)** hereby declare that the Project Report entitled "**ENHANCED SECURITY METHOD FOR IOT BASED VEHICLE CRASH RESPONSE SYSTEM**" done by us under the guidance of **Dr. P. Kavipriya** Professor, Department of Electronics and Communication Engineering at Sathyabama Institute Of Science And Technology, Chennai, is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Electronics and Communication Engineering.

DATE: 17 APR 2024

PLACE: Chennai

SIGNATURE OF CANDIDATES:

1. **SHAURYA (40130215)**

Shaurya

2. **THALLAPALLI GAYATHRI (40130233)**

T. Gayathri

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to **Board of Management of Sathyabama Institute of Science and Technology** for their encouragement in doing this project and for completing it successfully. We are grateful to them.

We convey our thanks to **Dr. N. M. NANDHITHA, M.E., Ph.D., Dean, School of Electrical and Electronics Engineering** and **Dr. T. RAVI, M.E., Ph.D., Head of the Department, Department of Electronics and Communication Engineering** for providing us necessary support and details at the right time during the progressive reviews.

We would like to express our sincere and deep sense of gratitude to our Project Guide **Dr. P. Kavipriya, Professor**, Department of Electronics and Communication Engineering for her valuable guidance, suggestions and constant encouragement paved the way for the successful completion of our project work.

We wish to express our thanks to all Teaching and Non-teaching staff members of the Department of Electronics and Communication Engineering who were helpful in many ways for the completion of the project

ABSTRACT

Four-wheelers traffic accidents are very common nowadays. Quick medical help saves lives. The system is designed to alert nearby medical facilities to incidents to provide emergency medical assistance. The accelerometer installed in the car can detect the inclination of the car. Thus, the system will decide and send the information to the smart phone connected to the electronic meter with GSM and GPS modules.

In this research work we proposed a new enhanced encryption algorithm to secure the details of injured person.

TABLE OF CONTENT

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	i
	LIST OF TABLES	iii
	LIST OF FIGURES	iv
1	INTRODUCTION	1 - 5
	1.1 BACKGROUND	2
	1.2 MOTIVATION	2
	1.3 IOT DEFINITION	3 - 5
2	LITERATURE SURVEY	6 – 8
	2.1 LITERATURE REVIEW	6 – 8
3	AIM & SCOPE OF PRESENT INVESTIGATION	9 - 11
	3.1 AIM	9
	3.2 OBJECTIVES	9 - 10
	3.3 SCOPE	10 – 11
4	SYSTEM DESIGN	12 - 37
	4.1 EXISTING SYSTEM	12 – 13
	4.2 PROBLEM STATEMENT	13 – 14
	4.3 PROPOSED SYSTEM	14 – 17
	4.4 BLOCK DIAGRAM	17
	4.5 HARDWARE REQUIREMENTS	18
	4.6 HARDWARE DESCRIPTION	18 – 27
	4.7 SOFTWARE REQUIREMENTS	27 – 29
	4.8 ALGORITHM USED	29 - 37
5	RESULTS AND DISCUSSION	38 – 41
6	SUMMARY AND CONCLUSION	42
	6.1 SUMMARY	42
	6.2 CONCLUSION	42
7	REFERENCES	43 - 46

LIST OF TABLES

TABLE NO.	DESCRIPTION	PAGE NO.
4.8.1	ENCRYPTION TABLE	32 - 34
4.8.2	DECRYPTION TABLE	34 – 36
4.9.1	DIFFERENCE BETWEEN ENCRYPTION AND DECRYPTION	36 - 37

LIST OF FIGURES

FIGURE NO.	DESCRIPTION	PAGE NO.
1.1	IOT ARCHITECTURE	4
4.6.1	ARDUINO UNO	18
4.6.2	ATMEGA328P AND ARDUINO UNO PIN MAPPING	19
4.6.3	NODE MCU	20
4.6.4	GSM MODULE	21
4.6.5	GPS MODULE	22
4.6.6	ACCELEROMETER	23
4.6.7	BUZZER	25
4.6.8	POWER SUPPLY	26
4.7.1	ARCHITECTURE OF THINGSPEAK	28
4.8.1	ENCRYPTION ALGORITHM	29
4.8.2	SUB BYTES	31
4.8.3	SHIFT ROWS	31
4.8.4	MIX COLUMNS	32
5.1	PROJECT SETUP	38
5.2	LOCATION OF ACCIDENT	39
5.3	MESSAGE RECEIVED THROUGH SMS FOR HOSPITAL	39
5.4	MESSAGE RECEIVED THROUGH SMS FOR POLICE	40
5.5	MESSAGE RECEIVED THROUGH SMS FOR FAMILY	40
5.6	COMPARISON OF ENCRYPTED MESSAGES	40
5.7	OUTPUT CONSOLE	41

CHAPTER 1

INTRODUCTION

The development of a transportation system has been the generative power for human beings to have the highest civilization above creatures in the earth. Automobile has a great importance in our daily life. We utilize it to go to our work place, keep in touch with our friends and family, and deliver our goods. But it can also bring disaster to us and even can kill us through accidents. Speed is one of the most important and basic risk factors in driving. It not only affects the severity of a crash, but also increases risk of being involved in a crash. Despite many efforts taken by different governmental and non-governmental organizations all around the world by various programs to aware against careless driving, yet accidents are taking place every now and then. However, many lives could have been saved if the emergency service could get the crash information in time. A study by Virtanen et al. shows that 4.6% of the fatalities in accidents could have been prevented only if the emergency services could be provided at the place of accident at the proper time. As such, efficient automatic accident detection with an automatic notification to the emergency service with the accident location is a prime need to save the precious human life. IoT-based vehicle crash response systems hold immense promise in improving road safety and emergency response. These systems can automatically detect accidents, assess their severity, and transmit critical information to emergency services and relevant stakeholders in real-time. They can trigger life-saving actions such as deploying airbags, unlocking doors, or shutting off fuel supplies. Moreover, they can collect valuable data for accident reconstruction and analysis, contributing to safer roads and more efficient emergency services. The integration of the Internet of Things (IoT) into the automotive industry has ushered in a new era of vehicle safety and connectivity. IoT-based vehicle crash response systems have the potential to revolutionize emergency response and enhance the safety of drivers and passengers. These systems can detect accidents in real-time, assess their severity, and automatically initiate a rapid and coordinated response, potentially saving lives and reducing the severity of injuries. However, while IoT technology offers numerous benefits, it also introduces a range of security challenges that must be addressed to ensure the trustworthiness and effectiveness of these systems.

The sensitive nature of the data involved, the potential for unauthorized access, and the need for continuous system availability make security a paramount concern. In the subsequent sections, we will examine the security challenges, regulatory considerations, and the potential impact of security breaches in IoT-based vehicle crash response systems. We will also delve into specific security methods, including secure communication, authentication, access control, data encryption, and physical security measures, all aimed at fortifying the resilience and reliability of these systems. Ultimately, our goal is to create a foundation of trust and security, ensuring that IoT-based vehicle crash response systems fulfil their lifesaving potential while mitigating the associated risks.

1.1 BACKGROUND

Road accidents continue to be a significant global public health concern, resulting in a substantial loss of life and property. According to the World Health Organization (WHO), approximately 1.35 million people die each year due to road traffic accidents, making it a leading cause of death worldwide. Timely and efficient emergency response plays a pivotal role in reducing fatalities and preventing severe injuries resulting from these accidents. Traditional emergency response mechanisms often face challenges in terms of speed and accuracy. Recognizing the limitations of conventional approaches, our Project explores the integration of emerging technologies to enhance the effectiveness of accident detection and emergency notification systems.

1.2 MOTIVATION

The motivation behind this project stems from the pressing need to address the security vulnerabilities inherent in IoT-based Vehicle Crash Response Systems (VCRS). While these systems offer immense potential for improving road safety and emergency response, their effectiveness can be undetermined by security breaches, potentially leading to dire consequences for vehicle occupants and other road users.

1.3 SCOPE OF THE PROJECT

The scope of the project is to design an accidental detection system that detect the accidents and alert rescue team in time. Arduino is major control unit to detect or alert when an accident occurs, which helps in transferring messages to different devices in the system. Receiving pin of GSM module and transmitting pin of GPS module are used to communication GPS module will find the location of the vehicle and the information is fetched by the receiver through the coordinates and the received data is sent to Arduino and the alert to rescue team by GSM module.

1.4 IoT DEFINITION

In this section, we explain some of the IoT definitions. Also, we explain the layered architecture for IoT.

Internet of Things (IoT) has increasingly gained attention in industry to interact with different types of devices. IoT can have influence on industry and society by integrating physical devices into information networks. IoT impacts can be on different perspectives, namely for private and business users. From the perspective of a private user, IoT has effect on both working and personal fields, such as smart homes and offices, e-health and assisted living. From the aspect of a business user the impacts would be in fields such as automation and industrial manufacturing, logistics, business process management, intelligent transportation of people and goods.

IoT integrates physical things into information networks. IoT covers the overall infrastructure, including software, hardware and services, which is used to support these information networks. The integrated physical things can exchange data about the physical properties and information that they sense in their environment. To identify devices, we can use identification technologies like for example RFID, which allow each device be uniquely identified

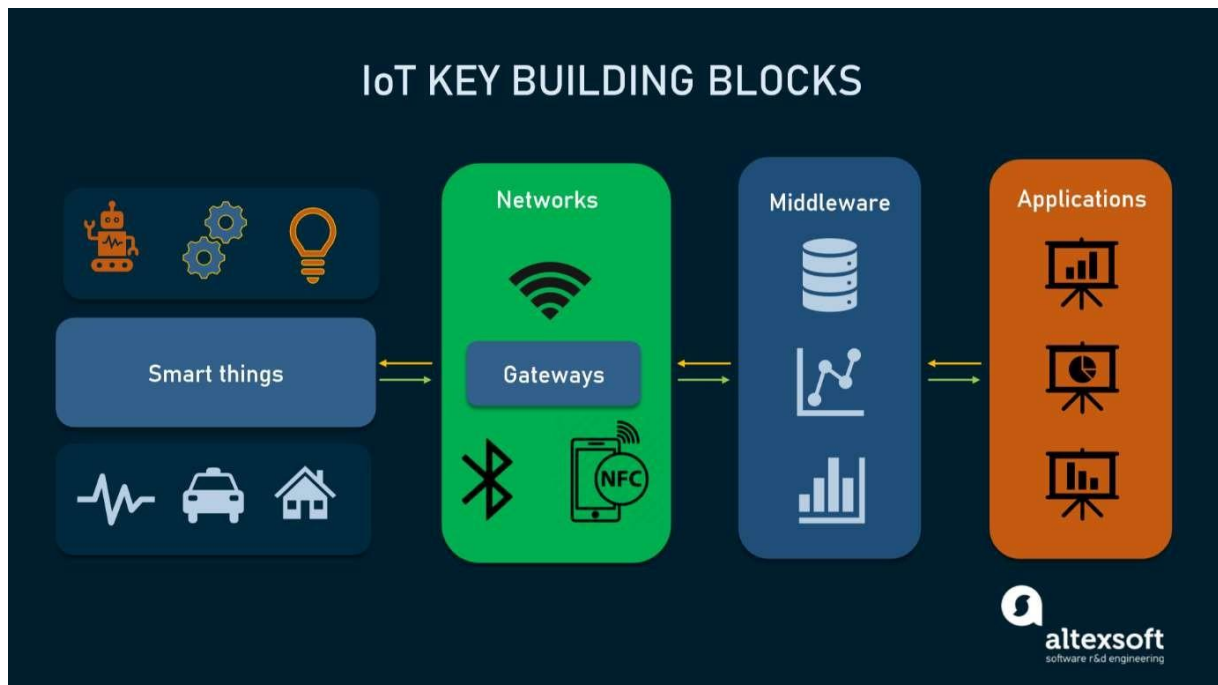


Fig 1.1 IoT Architecture

International Telecommunication Union (ITU) defines IoT as “A global infrastructure for the Information Society, enabling advanced services by interconnecting (physical and virtual) things based on, existing and evolving, interoperable information and communication technologies”.

IoT has a layered architecture designed to answer the demands of various industries, enterprises and society. It shows a generic layered architecture for IoT that consist offive layers, which are discussed, in the following:

- **Edge Technology layer**

This is a hardware layer that consists of embedded systems, RFID tags, sensor networks and all the other sensors in different forms. This hardware layer can perform several functions, such as collecting information from a system or an environment, processing information and supporting communication.

- **Access Gateway layer**

This layer is concerned with data handling, and is responsible for publishing and subscribing the services that are provided by the *Things*, message routing, and hovelling the communication between platforms.

- ***Middleware layer***

This layer has some critical functionalities, such as aggregating and filtering the received data from the hardware devices, performing information discovery and providing access control to the devices for applications.

- ***Application layer***

This layer is responsible for delivering various application services. These services are provided through the middleware layer to different applications and users in IoT-based.

CHAPTER 2

LITERATURE SURVEY

2.1 LITERATURE REVIEW

2.1.1 Title: "A Review on Intelligent Accident Detection and Emergency Notification Systems"

Authors: A. Anitha, Dr. P. Thangaraj

Methodology: This Project provides an overview of various intelligent accident detection systems, emphasizing the need for swift emergency responses. The survey covers methods such as accelerometers, gyroscopes, and computer vision for accident detection. It highlights the importance of integrating GPS and GSM technologies for precise location identification and efficient communication with emergency services.

2.1.2 Title: "A Survey on Intelligent Accident Detection and Reporting System Using IoT"

Authors: A. M. Prabhu, Dr. N. S. Marimuthu

Methodology: The survey focuses on IoT-based accident detection systems, exploring the integration of sensors, communication modules, and cloud platforms. It discusses the potential of MEMS sensors for real-time data collection and emphasizes the role of cloud computing in storing and processing accident-related information. The Project also touches upon energy-efficient designs to enhance the system's reliability.

2.1.3 Title: "Accident Detection and Reporting System Using GPS and GSM Modules"

Authors: Dr. S. P. Chokkalingam, V. Harikrishnan, et al.

Methodology: This work presents a comprehensive accident detection system utilizing GPS and GSM modules. It discusses the integration of accelerometers and GPS for effective accident detection and location tracking. The Project outlines a communication framework to notify emergency services and family members promptly. It provides insights into the practical implementation of the proposed system.

2.1.4. Title: "IoT-Based Smart Helmet for Accident Detection and Reporting System"

Authors: D. B. Dhondge, Dr. R. D. Kharadkar

Methodology: Focused on two-wheeler safety, this Project introduces a smart helmet system equipped with sensors for accident detection. The Project explores the integration of accelerometers and GPS within the helmet. It discusses the use of GSM for communication and proposes a system that can automatically alert nearby hospitals and police stations.

2.1.5. Title: "An Intelligent Accident Detection and Alerting System Using Wireless Communication"

Authors: R. Karthik, Dr. S. Sudha

Methodology: The Project explores wireless communication techniques for accident detection. It discusses the integration of accelerometers and GPS modules, emphasizing the utilization of Zigbee communication. The Project proposes a decentralized approach where vehicles communicate with each other to enhance the overall effectiveness of accident detection and notification.

2.1.6. Title: "Real-Time Vehicle Accident Detection and Notification System Using GSM and GPS"

Authors: C. S. R. Anjaneyulu, Dr. D. Ramakrishna, et al.

Methodology: This Project delves into real-time accident detection systems with a focus on GSM and GPS technologies. The authors propose an algorithm using accelerometer data to identify accidents and employ GPS for location tracking. The Project also discusses the integration of GSM for timely notification to emergency services and family members.

2.1.7. Title: "A Survey on Vehicular Ad Hoc Networks for Intelligent Transportation Systems"

Authors: N. Kumaravel, K. Baskaran

Methodology: While not exclusively focused on accident detection, this survey explores Vehicular Ad Hoc Networks (VANETs) for intelligent transportation systems. It covers communication protocols and network architectures relevant to

the context of real-time information exchange in vehicular environments, providing insights into potential applications for accident detection systems.

2.1.8. Title: "A Review on Smart Accident Detection and Reporting System using IoT and Machine Learning"

Authors: D. S. Sable, M. H. Zaheer, et al.

Methodology: Combining IoT and machine learning, this review explores advanced accident detection systems. It discusses the integration of various sensors, including accelerometers, and emphasizes the role of machine learning algorithms for accurate accident identification. The Project presents a holistic approach, incorporating multiple technologies for enhanced performance.

2.1.9. Title: "Accident Detection and Reporting System using Smartphone Sensors"

Authors: S. A. Ladhake, Dr. A. M. Sapkal

Methodology: Focusing on leveraging Smartphone sensors for accident detection, this Project discusses the integration of accelerometers and GPS modules within smart phones. The Project proposes an application-based approach, utilizing the computing power of smart phones for real-time accident detection and communication with emergency services.

2.1.10. Title: "Development of an Automatic Vehicle Accident Detection and Messaging System"

Authors: R. A. Shaikh, A. V. Deorankar

Methodology: This work introduces an automatic vehicle accident detection system using accelerometers and GPS. It outlines the integration of GSM for communication and proposes a messaging system for notifying emergency services and family members. The Project highlights the practical implementation aspects and the potential for deployment in real-world scenarios.

CHAPTER 3

AIM AND SCOPE OF PRESENT INVESTIGATION

3.1 AIM

The aim of our project is to develop a robust and comprehensive Accident Detection System, leveraging the capabilities of Arduino, GSM, GPS, Buzzer, and AES (Advanced Encryption Standard). This method seeks to address the challenges associated with existing accident detection systems, particularly focusing on real-time response, precise location identification, efficient communication, and secure transmission of distress messages. The accidental detection and alert system is designed to detect accidents and alert the rescue team in time. Arduino is a major control unit to communicate between devices when an accident occurs, which helps in transferring messages to different devices in the system. The receiving pin of the GSM module and the transmitting pin of the GPS module are used for communication. The GPS module will find the location of the vehicle and the information is fetched by the receiver through the coordinates the received data is sent to Arduino and the alert to the rescue team by the GSM module. The accelerometer and gyroscope detect the accident occurrence by the reading produced by the movements of the vehicle. The vibration sensor is also used to detect the accident by producing voltage from the impact of vehicle movements.

3.2 OBJECTIVES

3.2.1 Prompt Accident Detection

- Develop an algorithm that accurately identifies potential accidents based on data from MEMS sensors.
- Implement real-time processing on the Arduino microcontroller to ensure swift decision-making.

3.2.2 Precise Location Identification

- Utilize GPS technology to determine the exact location of the accident.
- Integrate GPS data into the emergency notification system to enable rapid response.

3.2.3 Efficient Emergency Notification

- Employ the GSM module to send encrypted distress messages to nearby hospitals and police stations.
- Design a communication protocol for seamless integration with emergency services.

3.2.4 Family Notification System

- Implement a secure mechanism to notify pre-defined family members about the accident.
- Ensure encrypted communication to protect the privacy of individuals involved.

3.2.5 Auditory Alert Mechanism

- Integrate a Buzzer for an audible alert at the accident site, aiding first responders and passerby awareness.

3.2.6 AES Encryption Implementation

- Incorporate AES encryption to secure distress messages during transmission.
- Validate the effectiveness of AES in safeguarding sensitive information in the context of emergency communications.

3.3 SCOPE

The scope of this Project lies in its potential to revolutionize the landscape of emergency response systems by combining cutting-edge technologies for efficient accident detection and secure communication. By addressing the challenges associated with traditional methods, our proposed system aims to significantly reduce the response time of emergency services, thereby increasing the likelihood of saving lives and minimizing the severity of injuries resulting from road accidents.

The integration of AES encryption adds a crucial layer of security to the system, acknowledging the importance of protecting sensitive information during emergency communications. This Project not only contributes to the field of intelligent transportation systems but also sets a precedent for the implementation

of robust security measures in real-time applications where privacy and data integrity are paramount.

In the subsequent sections, we will delve into the technical details of the proposed system, discussing the design and implementation of each component and presenting the results of experimental validations. Through this comprehensive exploration, we aim to showcase the potential of our Accident Detection System as a viable and secure solution for enhancing emergency response in the face of road accidents.

CHAPTER 4

SYSTEM DESIGN

4.1 EXISTING SYSTEM

Many researchers carried out their studies on accident detection system. Traditional traffic accident prediction uses long-term traffic data such as annual average daily traffic and hourly volume. In contrast to traditional traffic accident prediction, real-time traffic accident prediction relates accident occurrences to real-time traffic data obtained from various detectors such as induction loops, infrared detector, camera etc. Real-time traffic accident prediction focuses on the change of traffic conditions before an accident occurrence, while traffic incident detection studies are concerned with the change of traffic conditions after an incident occurrence. However, the performance of these detection and prediction system is greatly restricted by the number of monitoring sensor, available fund, algorithms used to confirm an accident, weather, traffic flow etc.

4.1.1 DRAWBACKS OF THE EXISTING SYSTEM

- **Security Concerns:** Security is a significant disadvantage. IoT devices in vehicles are susceptible to hacking, making them potential targets for cyberattacks. Unauthorized access to the system can compromise the safety of the vehicle's occupants and the privacy of their data.
- **Privacy Issues:** These systems collect and transmit data, including the vehicle's location and occupants' information, to service providers and emergency responders. Privacy concerns may arise regarding who has access to this data and how it is used.
- **Complexity and Reliability:** IoT-based systems involve complex hardware and software components that need to function reliably. System failures or glitches can lead to false alarms or delayed responses during emergencies.

- **Cost of Implementation:** Implementing IoT technology in vehicles can be expensive. This includes the cost of hardware, software development, connectivity, and ongoing maintenance.
- **Maintenance Challenges:** IoT devices in vehicles require regular updates and maintenance. Ensuring that all devices are up-to-date and functioning correctly can be a logistical challenge.
- **Data Overload:** These systems generate a vast amount of data, which can be overwhelming for emergency responders and service providers. Managing and analysing this data effectively is crucial.
- **Lack of Real-Time Data:** Traditional systems may not provide real-time information about accidents. They rely on manual reporting, which can result in delays in emergency response.
- **Limited Accuracy:** Older systems may lack the ability to accurately detect the severity of an accident. They might not differentiate between minor incidents and life-threatening crashes, leading to unnecessary emergency responses.

4.2 PROBLEM STATEMENT

The escalating frequency of road accidents globally has underscored the urgent need for effective and rapid emergency response systems. Existing solutions often fall short in terms of prompt accident detection and timely notification, leading to increased fatalities and severity of injuries. Conventional methods lack the integration of advanced technologies that could significantly enhance the speed and accuracy of accident identification.

The primary problem lies in the reliance on manual reporting or basic sensor technologies that struggle to discern subtle but critical changes in vehicle dynamics during accidents. The delay in detecting and notifying emergency services

hampers their ability to respond swiftly, jeopardizing lives and exacerbating the impact of accidents.

Additionally, the lack of secure communication channels in existing systems poses a significant concern. Transmitting distress messages to hospitals, police stations, and family members without robust encryption puts sensitive information at risk of unauthorized access, potentially compromising the privacy and security of individuals involved in the accident.

Addressing these challenges requires the development of an intelligent Accident Detection System that seamlessly integrates MEMS sensors, GPS technology, GSM communication, and advanced encryption protocols. This Project aims to bridge these gaps and contribute to the establishment of a reliable, rapid, and secure emergency response system for mitigating the devastating consequences of road accidents.

4.3 PROPOSED SYSTEM

The proposed method aims to develop a robust and comprehensive Accident Detection System, leveraging the capabilities of Arduino, GSM, GPS, Buzzer, MEMS (Micro Electro-Mechanical Systems), and AES (Advanced Encryption Standard). This method seeks to address the challenges associated with existing accident detection systems, particularly focusing on real-time response, precise location identification, efficient communication, and secure transmission of distress messages.

4.3.1 Hardware Integration

The foundation of the proposed method lies in the integration of key hardware components. The Arduino microcontroller serves as the central processing unit, coordinating the functionalities of MEMS sensors, GPS module, GSM module, and the Buzzer.

- ***MEMS Sensors:*** MEMS sensors, specifically accelerometers, are employed to continuously monitor the vehicle's acceleration and deceleration. These sensors play a pivotal role in detecting abrupt changes in motion, indicative of a potential accident. The real-time data from MEMS sensors serve as input for the accident detection algorithm.
- ***GPS Module:*** The GPS module is utilized to determine the precise location of the accident. Accurate location data is crucial for notifying emergency services and family members effectively. The GPS coordinates are integrated into the distress message for immediate response and location tracking.
- ***GSM Module:*** The GSM module facilitates wireless communication for the transmission of distress messages. Upon detecting an accident, the system utilizes the GSM module to send encrypted messages to nearby hospitals, police stations, and pre-defined family members. This communication ensures that relevant stakeholders are promptly informed and can initiate necessary actions.
- ***Buzzer:*** An audible alert system in the form of a Buzzer is incorporated to provide an immediate and easily perceptible signal at the accident site. This serves a dual purpose: aiding first responders in locating the incident swiftly and alerting passersby to the presence of an emergency situation.

4.3.2. Accident Detection Algorithm

The Arduino microcontroller hosts an accident detection algorithm that processes real-time data from MEMS sensors. The algorithm is designed to recognize patterns associated with sudden changes in acceleration or deceleration that are characteristic of an accident. Upon detection, the algorithm triggers the subsequent actions, including GPS data retrieval, message encryption, and communication through the GSM module.

4.3.3 Communication Protocol

The communication protocol is a critical aspect of the proposed method, ensuring that distress messages are transmitted efficiently and securely. The system employs a GSM-based communication protocol to send messages in encrypted format using the AES algorithm.

AES-Encrypted Communication: Messages containing information about the accident, including location coordinates, are encrypted using the Advanced Encryption Standard (AES). AES is a widely accepted encryption algorithm known for its security and efficiency. Encrypting the messages ensures that sensitive information remains confidential during transmission, addressing concerns related to unauthorized access.

4.3.4 Message Recipients

The proposed method considers three primary categories of message recipients: nearby hospitals, police stations, and pre-defined family members.

- **Nearby Hospitals:** Distress messages are sent to nearby hospitals, providing them with encrypted information about the accident and its location. This prompt notification enables hospitals to prepare for incoming emergency cases, potentially reducing the response time for medical assistance.
- **Police Stations:** Similar to hospitals, nearby police stations receive encrypted distress messages containing accident details. This aids law enforcement in swiftly responding to the scene, managing traffic, and coordinating rescue operations if necessary.
- **Family Members:** Pre-defined family members receive encrypted messages notifying them about the accident. This ensures that the immediate circle of individuals related to the accident is informed promptly. The encrypted format adds a layer of security, preventing unauthorized individuals from accessing sensitive information.

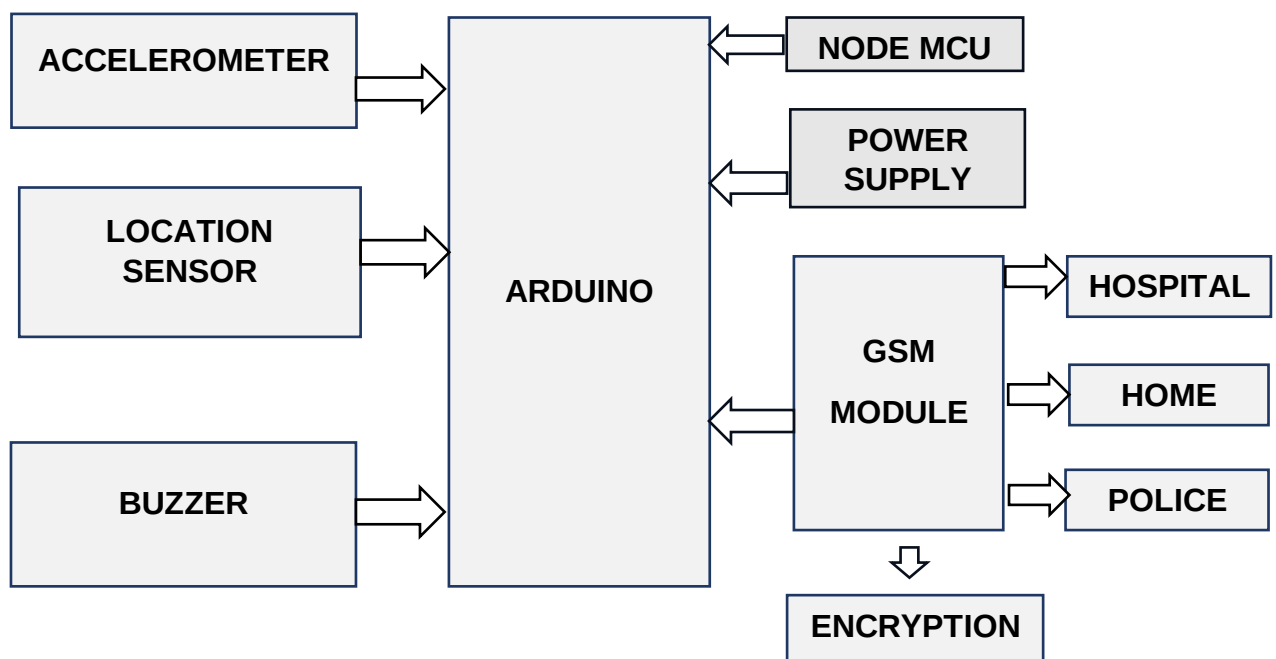
4.3.5 User Interface and System Feedback

The proposed system can incorporate a user interface, possibly through a mobile application, providing real-time feedback to the user. The interface can display the status of distress messages, confirmations from recipients (hospitals, police stations, family members), and additional information such as emergency contact details.

4.3.6 Implementation and Testing

The proposed method will undergo rigorous testing to evaluate its effectiveness and reliability in real-world scenarios. Controlled experiments simulating various accident scenarios will be conducted to validate the accuracy of the accident detection algorithm, the efficiency of the communication protocol, and the security of AES encryption.

4.4 BLOCK DIAGRAM OF PROPOSED SYSTEM



4.5 HARDWARE REQUIREMENTS

- Arduino UNO
- Node MCU
- GSM Module
- GPS Module
- Accelerometer
- Location Sensor
- Buzzer
- Power Supply
- Connecting Wires

4.6 HARDWARE DESCRIPTION

4.6.1 Arduino UNO



Fig 4.6.1 Arduino UNO

The Arduino Uno is one of the most popular and widely used microcontroller boards in the Arduino family. It is designed for beginners and professionals alike, offering a versatile platform for a wide range of electronic projects. The Arduino Uno is based on the ATmega328P microcontroller, which is a low-power, high-performance chip from the Atmel AVR 8-bit family. The ATmega328P microcontroller on the Arduino Uno runs at a clock speed of 16MHz, providing sufficient processing power for most embedded applications. The Arduino Uno features a total of 14 digital input/output (I/O) pins, of which 6 can be used as PWM (Pulse Width Modulation) outputs. In

In addition to digital I/O pins, the Arduino Uno has 6 analog input pins, marked A0 through A5. The Arduino Uno can be powered via a USB connection or an external power supply. A reset button is provided on the Arduino Uno board, allowing users to reset the microcontroller and restart the program execution. The Arduino Uno features female headers for connecting external components and shields. A versatile microcontroller which is widely used throughout for various electronic project is the Arduino Uno dependent on the AT Mega 328P. It has 14 computerized i/o pins, 6 analogous inputs, a Universal Serial Bus, and a reset button. Comprises of all that anticipated to assist the microcontroller; interface Personal Computer as well as provide power to via a battery or a connector. There are various applications for which an Arduino board can be use. Such as:

- Modelling prototypes and systems of electronic nature
- Endless possibilities of DIY (Do-It-Yourself) Undertakings
- Simple beginner level usage
- Undertakings requiring multiple I/O pins

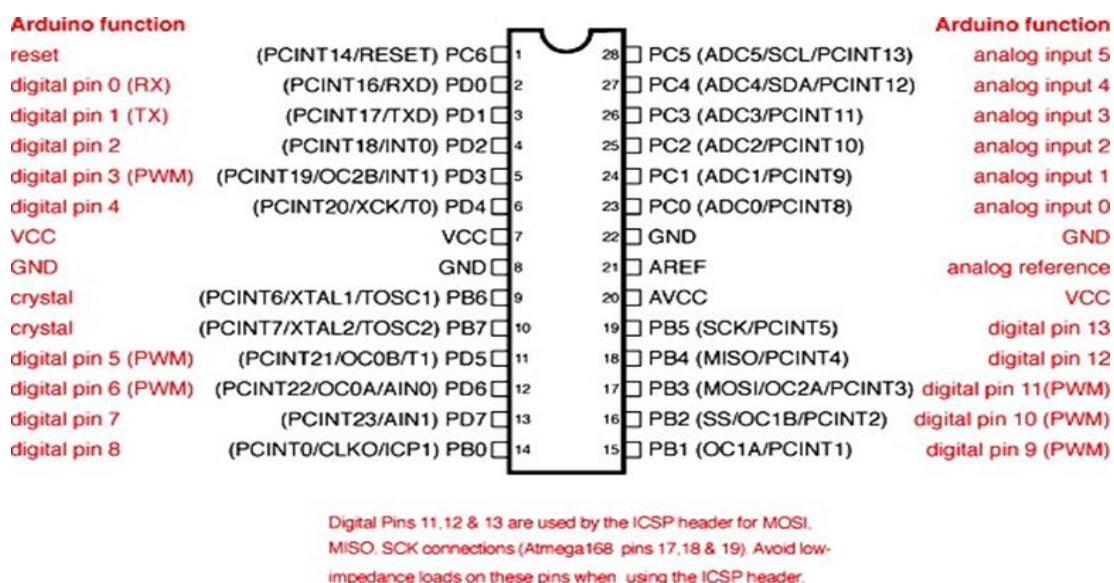


Fig 4.6.2 ATmega328P and Arduino Uno Pin Mapping

4.6.2 Node MCU



Fig 4.6.3 Node MCU

Node MCU is an open-source firmware and development kit that enables the rapid prototyping of IoT (Internet of Things) projects. It is based on the ESP8266 Wi-Fi module, which integrates a microcontroller and a Wi-Fi transceiver, making it an ideal platform for building connected devices and IoT applications. Node MCU is built around the ESP8266 microcontroller, which features a 32-bit Ten silica Xtensa LX106 CPU running at clock speeds up to 80 MHz. The ESP8266 is known for its low cost, low power consumption, and built-in Wi-Fi connectivity, making it well-suited for IoT applications. Node MCU comes with pre-installed firmware based on the Lua scripting language. Lua is a lightweight, powerful scripting language that is easy to learn and use, making it ideal for rapid prototyping and development of IoT projects. In addition to Lua, Node MCU also supports programming using the Arduino IDE (Integrated Development Environment). This allows users to write code in C/C++ using familiar Arduino libraries and syntax, making it accessible to a wider audience of developers and hobbyists. One of the key features of Node MCU is its built-in Wi-Fi connectivity, which allows devices to connect to the internet and communicate with other devices over a wireless network. This enables a wide range of IoT applications, including remote monitoring, home automation, and smart devices. Node MCU features multiple General-purpose Input/Output (GPIO) pins, which can be used to interface with external sensors, actuators, displays, and other electronic components. These pins support digital input/output, analog input, PWM (Pulse Width Modulation) output, and other standard GPIO functionalities. Node MCU is equipped with a USB interface for programming and power supply.

4.6.3 GSM Module



Fig 4.6.4 GSM Module

GSM (Global System for Mobile Communications, originally Groupe Special Mobile) is a standard developed by the European Telecommunications Standards Institute (ETSI). It was created to describe the protocols for second-generation (2G) digital cellular networks used by mobile phones and is now the default global standard for mobile communications – with over 90% market share, operating in over 219 countries and territories.

A GSM module or a GPRS module is a chip or circuit that will be used to establish communication between a mobile device or a computing machine and a GSM or GPRS system. The modem (modulator-demodulator) is a critical part here. These modules consist of a GSM module or GPRS modem powered by a power supply circuit and communication interfaces (like RS-232, USB 2.0, and others) for computers. A GSM modem can be a dedicated modem device with a serial, USB, or Bluetooth connection, or it can be a mobile phone that provides GSM modem capabilities.

4.6.4 GPS Module



Fig 4.6.5 GPS Module

A GPS (Global Positioning System) module is a device that receives signals from satellites orbiting the Earth to determine its precise geographic location.

These modules are commonly used in a wide range of applications, including navigation systems, vehicle tracking, asset management, and outdoor recreation. The core component of a GPS module is the GPS receiver, which receives signals from multiple satellites in the GPS constellation. Each satellite broadcasts a signal containing information about its position and the current time. By triangulating signals from multiple satellites, the GPS receiver calculates its own position, including latitude, longitude, altitude, and sometimes speed and heading. The GPS constellation consists of a network of satellites orbiting the Earth, each transmitting signals on different frequencies. Currently, the GPS constellation comprises 24 satellites arranged in six orbital planes, always ensuring global coverage and availability of signals. GPS modules are equipped with an antenna to receive signals from GPS satellites. The antenna may be integrated into the module itself or provided as an external component, depending on the design and form factor of the module. GPS modules typically provide location data in various formats, including NMEA (National Marine Electronics Association) sentences, which are standardized ASCII strings containing information such as latitude, longitude, altitude, speed, and time. The accuracy of a GPS module depends on several factors, including the number of satellites in view, the geometry of satellite positions relative to the receiver, atmospheric conditions, and the quality of the

receiver's hardware and algorithms. GPS modules provide highly accurate time synchronization based on signals received from GPS satellites. GPS modules require a power supply to operate, typically in the range of 3.3 to 5 volts DC.

Applications

- GPS modules are widely used in automotive applications for vehicle tracking, navigation, and fleet management.
- GPS modules are popular among outdoor enthusiasts for activities such as hiking, camping, geocaching, and biking.
- GPS modules are essential components in aircraft navigation systems, providing pilots with precise position information, flight planning.
- GPS modules are utilized in asset tracking systems to monitor the location, status, and movement of valuable assets, equipment, and inventory items.

4.6.5 Accelerometer

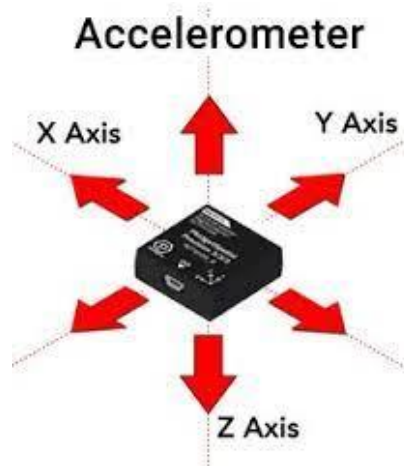


Fig 4.6.6 Accelerometer

An accelerometer is a sensor that measures acceleration, either static (such as gravity) or dynamic (such as motion or vibrations), in one or more axes of movement. It detects changes in velocity along one or more axes and converts these changes into electrical signals, which can then be processed and analysed by electronic circuits or microcontrollers. Accelerometers work based on the principle of inertial sensing. They contain one or more microstructures (such as piezoelectric crystals, capacitive plates, or MEMS - Microelectromechanical Systems) that move in response to acceleration. When acceleration is applied to

the sensor, the mass inside the sensor experiences a force, causing the microstructures to deform or move relative to the sensor housing. This deformation generates an electrical signal proportional to the applied acceleration. Accelerometers can measure acceleration along one, two, or three axes of movement, depending on the sensor's configuration. Single-axis accelerometers measure acceleration along a single axis (typically the vertical axis), while dual-axis and tri-axis accelerometers measure acceleration along two or three orthogonal axes (X, Y, and Z). Accelerometers have a specified sensitivity, which determines the amount of electrical output produced per unit of acceleration applied to the sensor. The sensitivity is typically expressed in units such as millivolts per g (mV/g) or volts per g (V/g), where "g" represents the acceleration due to gravity (approximately 9.81 m/s^2). The measurement range of an accelerometer defines the minimum and maximum acceleration values that the sensor can accurately measure without saturation or clipping.

Applications

- Accelerometers are used in a wide range of applications across various industries, including automotive, aerospace, consumer electronics, healthcare, and industrial monitoring.
- Automotive: Vehicle stability control, airbag deployment, rollover detection, and navigation systems.
- Consumer Electronics: Motion sensing in smartphones, tablets, gaming controllers, and wearable devices.
- Healthcare: Fall detection, activity monitoring, and medical device instrumentation.
- Industrial Monitoring: Condition monitoring of machinery, vibration analysis, and structural health monitoring.
- Aerospace: Aircraft attitude and heading reference systems (AHRS), inertial navigation systems (INS), and flight control systems.

4.6.6 Buzzer



Fig 4.6.7 Buzzer

An audio signaling device like a beeper or buzzer may be electromechanical or piezoelectric or mechanical type. The main function of this is to convert the signal from audio to sound. Generally, it is powered through DC voltage and used in timers, alarm devices, printers, alarms, computers, etc. Based on the various designs, it can generate different sounds like alarm, music, bell & siren.

The pin Configuration of the buzzer is shown below. It includes two pins namely positive and negative. The positive terminal of this is represented with the '+' symbol or a longer terminal. This terminal is powered through 6Volts whereas the negative terminal is represented with the '-' symbol or short terminal and it is connected to the GND terminal.

The specifications of the buzzer include the following.

- Colour is black
- The frequency range is 3,300Hz
- Operating Temperature ranges from – 20° C to +60°C
- Operating voltage ranges from 3V to 24V DC
- The sound pressure level is 85dBA or 10cm
- The supply current is below 15mA

A buzzer is available in different types which include the following.

- Piezoelectric

- Electromagnetic
- Mechanical
- Electromechanical
- Magnetic

The working principle of a buzzer depends on the theory that, once the voltage is given across a piezoelectric material, then a pressure difference is produced. A piezo type includes piezo crystals among two conductors. Once a potential disparity is given across these crystals, then they thrust one conductor & drag the additional conductor through their internal property. So this continuous action will produce a sharp sound signal.

4.6.7 Power Supply, Connecting Wires and Bread Board

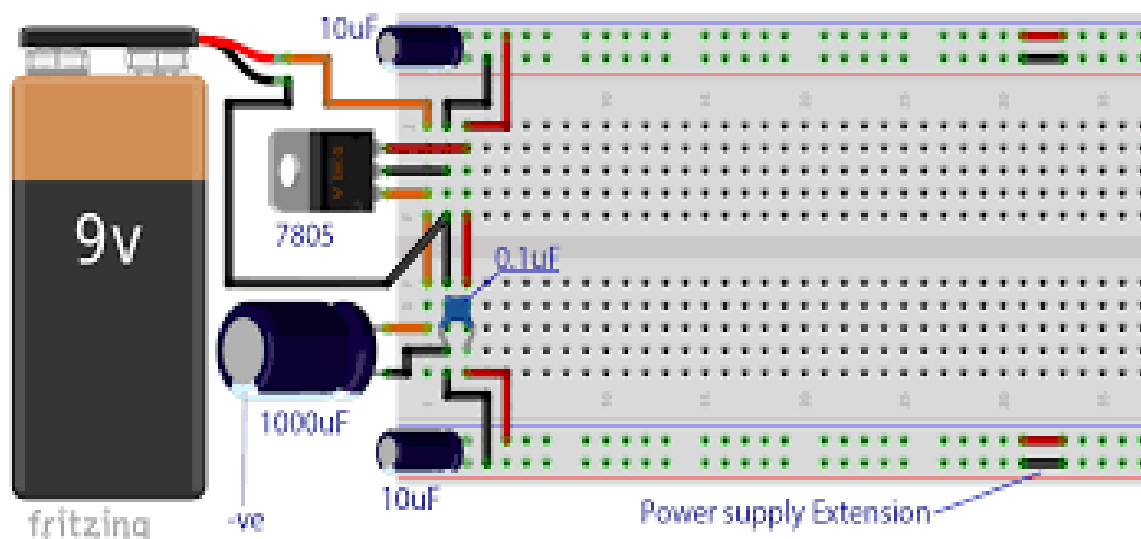


Fig 4.6.8 Power Supply

A power supply breadboard is a device used in electronics prototyping to provide a stable source of electrical power to circuits built on a breadboard. It typically consists of a compact module that plugs into the breadboard and supplies regulated DC voltage and current to power electronic components.

- **Voltage Regulation:** It provides adjustable or fixed DC output voltages, typically ranging from a few volts to tens of volts, depending on the model.

- **Current Limiting:** Many power supply breadboards include current-limiting capabilities to protect circuits from overcurrent conditions.
- **Input Power Source:** Power supply breadboards can be powered from various sources, such as AC mains (via a wall adapter), batteries, USB ports, or external DC power supplies.
- **Display and Controls:** Some power supply breadboards feature built-in displays (e.g., LCD or LED) to indicate output voltage and current values.

Connecting wires, also known as jumper wires or hookup wires, are essential components used to establish electrical connections between components on a breadboard or between a breadboard and external devices. They are typically made of flexible insulated wire with connectors (e.g., pins, sockets, or alligator clips) at each end.

4.7 SOFTWARE REQUIREMENTS

4.7.1 *Arduino IDE*

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards can read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing. Arduino boards are equipped with microcontrollers, which act as the brain of the system. The Arduino Software (IDE) provides an integrated development environment for writing and uploading code to the Arduino board. Arduino programming involves writing code, known as sketches, using the Arduino IDE. One of the key features of Arduino is its versatility and expandability. Users can easily connect various electronic components, such as sensors, actuators, displays, and communication modules, to Arduino boards using standard connectors (e.g., digital and analog pins, serial interfaces).

4.7.1 ThingSpeak

ThingSpeak is an open-source Internet of Things (IoT) application and API to store and retrieve data from things using the HTTP and MQTT protocol over the Internet or via a Local Area Network. ThingSpeak enables the creation of sensor logging applications, location tracking applications, and a social network of things with status updates”. ThingSpeak was originally launched by IoT Bridge in 2010 as a service in support of IoT applications. ThingSpeak has integrated support from the numerical computing software MATLAB from MathWorks, allowing ThingSpeak users to analyse and visualize uploaded data using MATLAB without requiring the purchase of a MATLAB license from MathWorks. ThingSpeak has a close relationship with MathWorks, Inc. In fact, all of the ThingSpeak documentation is incorporated into the MathWorks' MATLAB documentation site and even enabling registered MathWorks user accounts as valid login credentials on the ThingSpeak website.



Fig 4.7.1 Architecture Of Thingspeak

ThingSpeak enables sensors, instruments, and websites to send data to the cloud where it is stored in either a private or a public channel. Once data is in a **ThingSpeak** channel, you can analyse and visualize it, calculate new data, or interact with social media, web services, and other devices. ThingSpeak is an IoT analytics platform service from MathWorks, the makers of MATLAB and Simulink. ThingSpeak allows you to aggregate, visualize, and analyse live data streams in

the cloud. ThingSpeak provides instant visualizations of data posted by your devices or equipment. Execute MATLAB code in ThingSpeak, and perform online analysis and processing of the data as it comes in. ThingSpeak accelerates the development of proof-of-concept IoT systems, especially those that require analytics. You can build IoT systems without setting up servers or developing web software. For small- to medium-sized IoT systems, ThingSpeak provides a hosted solution that can be used in production.

4.8 ALGORITHM USED

4.8.1 AES ENCRYPTION ALGORITHM

The advanced encryption standard (AES) is based on symmetric-key matrix calculation. Later it becomes U.S. Government standard for secure and arrange the information in encrypted and decrypted form. Data are encrypted to inhibit from data hacker. Symmetric encryption technique has similar secret passkey on both encryption and decryption sides. If the passkey is endangered, the encryption is useless. AES encryption uses 128-bits, 192-bits and 256-bits passkeys. While talking about key lengths, usually to measure the strength of the key based on the number of trials. AES has ability to built-in flexibility of key length.

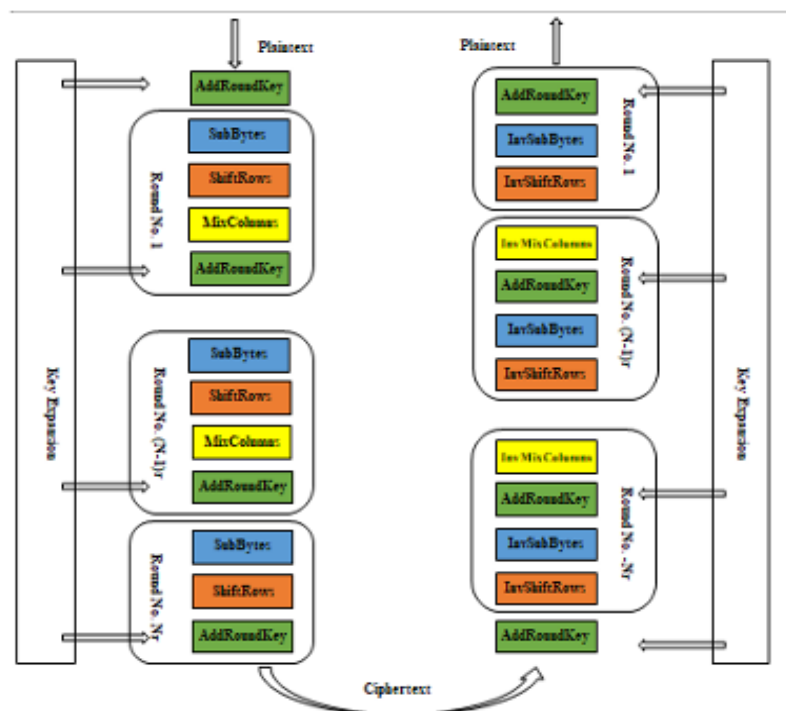


Fig 4.8.1 Encryption Algorithm

The encryption period of AES can be broken into three steps: initial round, the main rounds, and the final round. Most of the steps use a similar sub-task in various blends as follows:

Initial Round

- Add Round Key

Main Rounds

- Sub Bytes
- Shift Rows
- Mix Columns
- Add Round Key

Final Round

- Sub Bytes
- Shift Rows
- Add Round Key

Add Round Key

- The Add round key activity is the leading period of AES encryption.
- In this activity, the contribution to the round is ex- or operation with the secret round key.

Sub Bytes

In sub bytes steps, states of plaintext are replaces with corresponds to substitution box or s-box. In contrast to DES, AES utilizes a similar s-box for all bytes. Above s-box table, the byte input is distributed into two 4-bit parts in columns. The primary half decides the line and the second half decides the segment. If state matrix of cell value is 19 is replaced with first row and ninth columns of s-box cell value d4.

	0	1	2	3	4	5	6	7	8	9	0a	0b	0c	0d	0e	0f
0	63	7c	77	7b	f2	6b	6f	c5	30	1	67	2b	fe	d7	ab	76
1	ca	82	c9	d7	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	4	c7	23	c3	18	96	5	9a	7	12	80	e2	eb	27	b2	75
4	9	83	2c	1a	1b	6e	5a	a0	52	3b	d6	d3	29	e3	2f	84
5	53	d1	0	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	Cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	2	7f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	3d	f5	bc	b6	da	21	10	ff	f3	d2
8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	Db
a	e0	32	3a	0a	49	6	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	4e	A9	6c	56	f4	ea	65	7a	ae	8
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	3	f6	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	Df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Fig 4.8.2 Sub Bytes

Shift Rows

- After sub-bytes operation, we perform shift rows operation on 128-bits or 4x4 matrix data, where every cell contains a byte.
- In the Shift Rows, every one of these rows are moved to left by a set of sums:
- First row of state matrix is does not moved by any, second row is moved by one byte left, third row is moved by 2 bytes left and fourth row is moved by 3 bytes left.

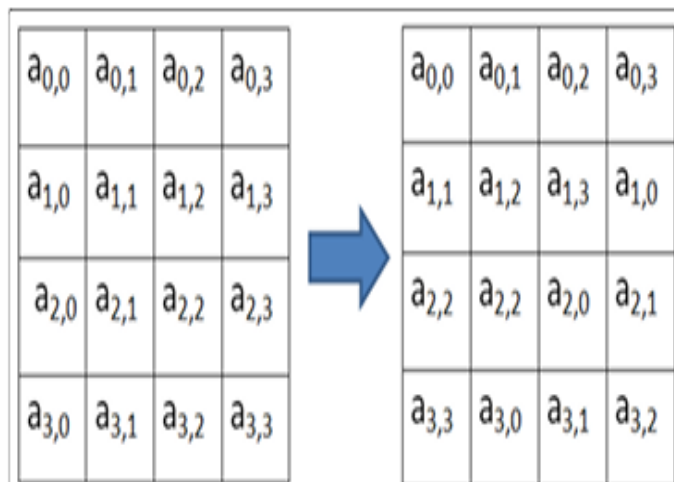


Fig 4.8.3 Shift Rows

Mix Columns

- After the shift rows of AES, the mix columns stage gives diffusion by blending the input contribution.
- It is not as shift rows, mix columns operations split the matrix column wise and each column (4x1) are multiple by 4x4 matrix and produced again a different column matrix (4x4).

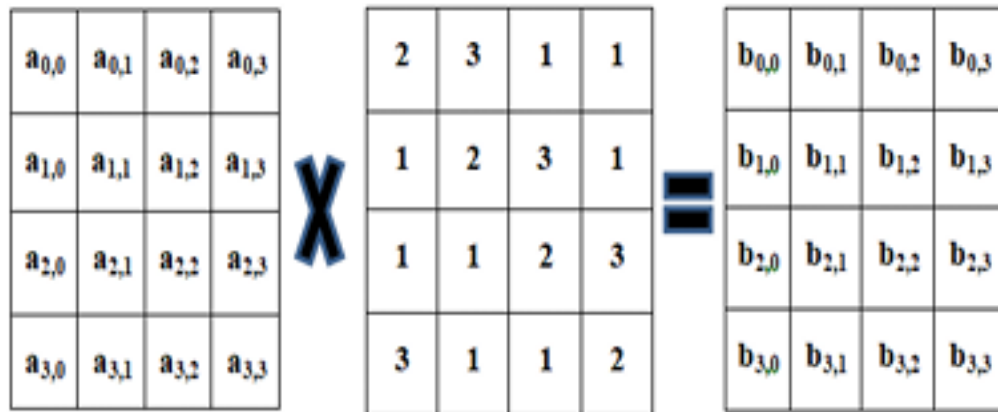


Fig 4.8.4 Mix Columns

4.8.2 Encryption Algorithm

The process or the key used for encrypting the patient data is shown below in Table

Table 4.8.1 Encryption Table

ACTUAL DATA	ENCRYPTED DATA
A	=
B	-
C	+
D	—
E	)
F	(

G	*
H	&
I	^
J	%
K	\$
L	#
M	@
N	!
O	`
P	~
Q	/
R	?
S	.
T	>
U	<
V	{
W	}
X	[
Y	]
Z	\
0	a
1	b
2	c

3	d
4	e
5	f
6	g
7	h
8	i
9	j

Decryption Process

Decryption is the process of returning encrypted and unreadable data to its unencrypted state. When decrypting, the system will extract the corrupted data and convert it into easy text and images that the reader and the system can understand. Decryption can be done manually or automatically. It can also be done using keys or passwords.

Advantages

The reasons for using decryption vary, but adequate security is one of the advantages. This method is especially important in terms of ensuring control of the organization. It's easy to see how this system would benefit security experts because it prevents filtering of duplicates that use encryption to hide sensitive information.

4.8.3 Decryption Algorithm

The process or the key used for decrypting the patient data is shown below in Table

Table 4.8.2 Decryption Table

ENCRYPTED DATA	DENCRYPTED DATA
=	A

-	B
+	C
—	D
)	E
(	F
*	G
&	H
^	I
%	J
\$	K
#	L
@	M
!	N
`	O
~	P
/	Q
?	R
.	S
>	T
<	U
{	V
}	W

[	X
]	Y
\	Z
a	0
b	1
c	2
d	3
e	4
f	5
g	6
h	7
i	8
j	9

4.9 DIFFERENCE BETWEEN ENCRYPTION AND DECRYPTION

Table 4.9.1 Difference Between Encryption and Decryption

ENCRYPTION	DECRYPTION
Encryption is the process of converting normal message into meaningless message.	While decryption is the process of converting meaningless message into its original form.
Encryption is the process which takes place at sender's end.	While decryption is the process which take place at receiver's end.

Its major task is to convert the plain text into cipher text.	While its main task is to convert the cipher text into plain text.
Any message can be encrypted with either secret key or public key.	Whereas the encrypted message can be decrypted with either secret key or private key.
In encryption process, sender sends the data to receiver after encrypted it.	Whereas in decryption process, receiver receives the information (Cipher text) and convert into plain text.

CHAPTER 5

RESULTS AND DISCUSSION

The results of implementing an enhanced security method based on a vehicle crash response system can vary depending on the specific approach taken, the technologies employed, and the effectiveness of the implementation. The enhanced security method can enhance the resilience of the vehicle crash response system against various cyber threats, such as hacking, data breaches, and malware attacks. The implementation of secure communication protocols and data encryption techniques can help ensure the confidentiality and integrity of sensitive information transmitted within the vehicle crash response system. By implementing real-time monitoring solutions and predictive analytics algorithms, the enhanced security method can enable faster detection and response to vehicular accidents or emergencies.

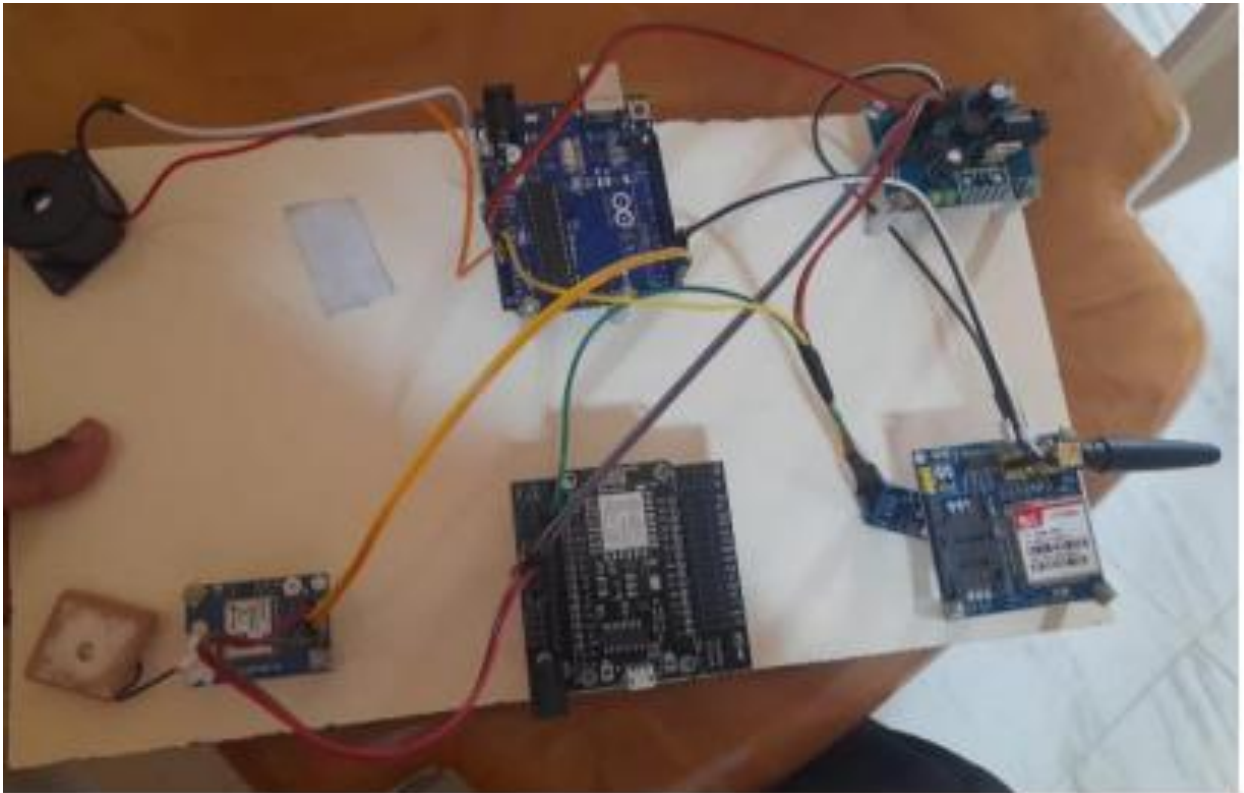


Fig 5.1 Project Setup

After accident occurred, the nearest hospital and ambulance should be notified. Delay in hospitalization may lead to loss of life. This collision detection device solves the problem and notifies the nearest repair center. Below figure shows the message received from SMS with its location. The system can provide latitude and longitude information as well as status location details.

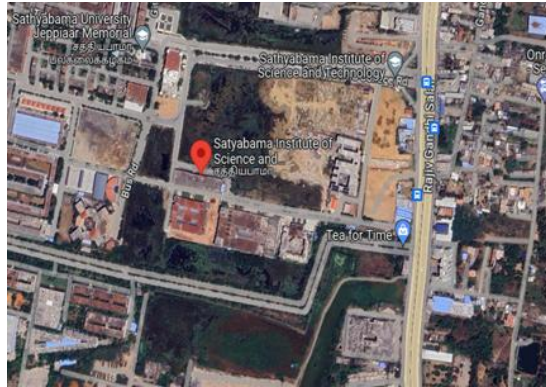


Fig 5.2 Location of Accident

Recipients: The plan will consider three main types of recipients: nearby hospitals, police stations and family members are planned first.

a. Nearby Hospitals: Distress messages are sent to nearby hospitals, giving them secrets about the situation and whereabouts. This timely notification will reduce the time it takes to respond to medical assistance, allowing hospitals to prepare for future emergencies.



Fig 5.3 Message received through SMS For Hospital

b. Police stations: Like hospitals, nearby police stations also receive distress reports with emergency details. This will help police respond quickly to the situation, control traffic and coordinate rescue operations if necessary.



Fig 5.4 Message received through SMS For Police

c. Family members: Family members identified before receiving an encrypted message informing them of the situation. In this way, those involved in the incident are quickly informed. The encrypted format adds a layer of security to prevent unauthorized people from accessing sensitive information.



Fig 5.5 Message received through SMS For Family

HOME			
NAME	LOCATION		
ABCD	12°52'18.2"N 80°13'22.3"E		

POLICE STATION			
NAME	AGE	MOBILE NUMBER	LOCATION
ABCD	20	912345678	12°52'18.2"N 80°13'22.3"E

HOSPITAL			
NAME	AGE	BLOOD GROUP	LOCATION
ABCD	20	A+	12°52'18.2"N 80°13'22.3"E

Fig 5.6 Comparison Of Encrypted Messages

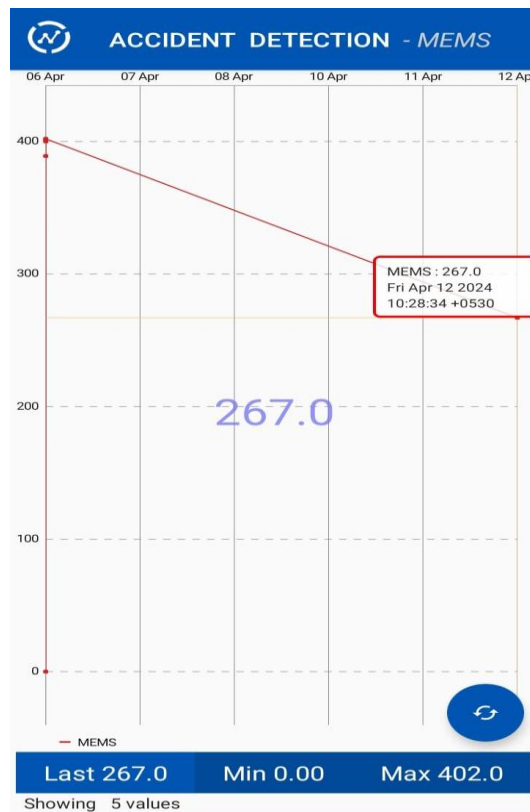
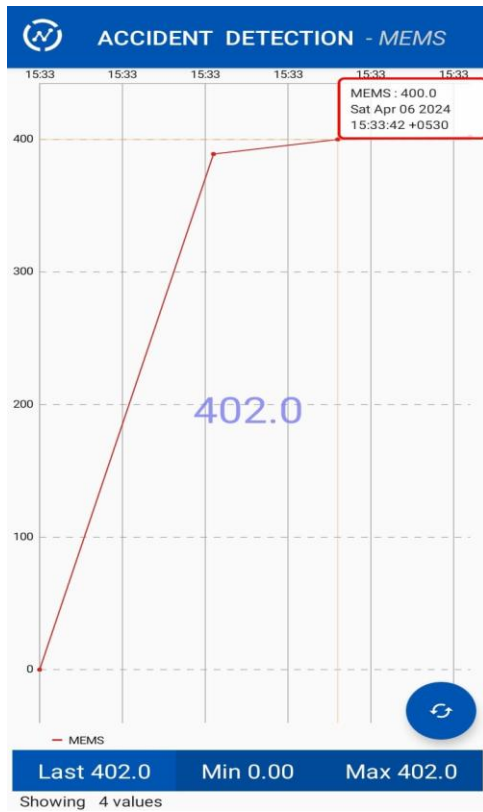


Fig 5.7 Output Console

CHAPTER 6

SUMMARY AND CONCLUSION

6.1 SUMMARY

The results of the proposed Accident Detection System demonstrate its efficacy in real-time accident identification, swift communication, and secure message transmission. The integration of Arduino, GSM, GPS, Buzzer, MEMS, and AES encryption ensures accurate detection of accidents, with prompt notifications sent to nearby hospitals, police stations, and pre-defined family members. The system's GPS module enables precise location tracking, aiding emergency responders. The AES-encrypted communication ensures the confidentiality of transmitted distress messages, enhancing the security of sensitive information. The incorporation of a Buzzer provides an immediate audible alert at the accident site. Overall, the proposed method showcases a holistic approach, addressing key challenges in existing systems and offering a reliable solution for intelligent and secure accident detection and notification.

6.2 CONCLUSION

A system to detect an event of accident has been developed. The proposed system deals with accident alerting and detection. It reads the exact latitude and longitude of the vehicle involved in the accident and sends this information to nearest emergency service provider. Arduino helps in transferring the message to different devices in the system. Accelerometer monitors the accident happening direction and gyroscope is used to determine rollover of the vehicle. The information is transferred to the registered number through GSM module. Using GPS, the location can be sent through tracking system to cover the geographical coordinates over the area.

REFERENCES

1. A. Alrajhi, K. Roy, L. Qingge and J. Kribs, "Detection of Road Condition Defects Using Multiple Sensors and IoT Technology: A Review," in IEEE Open Journal of Intelligent Transportation Systems, vol. 4, pp. 372-392, 2023, doi: 10.1109/OJITS.2023.3237480.
2. A. Balaji, U. Tripathi, V. Chamola, A. Benslimane and M. Guizani, "Toward Safer Vehicular Transit: Implementing Deep Learning on Single Channel EEG Systems for Microsleep Detection," in IEEE Transactions on Intelligent Transportation Systems, vol. 24, no. 1, pp. 1052-1061, Jan. 2023, doi: 10.1109/TITS.2021.3125126.J. p
3. A. S. Naik, S. K. Reddy and M. G. Raj, "RTEPMS: Real-Time Environmental Parameters Monitoring System Using IoT-Based LoRa 868-MHz Wireless Communication Technology in Underground Mines," in IEEE Access, vol. 12, pp. 7430-7455, 2024, doi: 10.1109/ACCESS.2024.3350429.
4. B. Wang, H. Zhang and Y. -X. Guo, "Radar-Based Soft Fall Detection Using Pattern Contour Vector," in IEEE Internet of Things Journal, vol. 10, no. 3, pp. 2519-2527, 1 Feb.1, 2023, doi: 10.1109/JIOT.2022.3213693.
5. C. Mouradian et al., "An IoT Platform-as-a-Service for NFV-Based Hybrid Cloud/Fog Systems," in IEEE Internet of Things Journal, vol. 7, no. 7, pp. 6102-6115, July 2020, doi: 10.1109/JIOT.2020.2968235.
6. C. Wisultschew, G. Mujica, J. M. Lanza-Gutierrez and J. Portilla, "3D-LIDAR Based Object Detection and Tracking on the Edge of IoT for Railway Level Crossing," in IEEE Access, vol. 9, pp. 35718-35729, 2021, doi: 10.1109/ACCESS.2021.3062220.

7. D. Na and S. Park, "Blockchain-Based Dashcam Video Management Method for Data Sharing and Integrity in V2V Network," in *IEEE Access*, vol. 10, pp. 3307-3319, 2022, doi: 10.1109/ACCESS.2022.3140419.
8. H. U. R. Siddiqui, A. A. Saleem, M. A. Raza, K. Zafar, K. Munir and S. Dudley, "IoT Based Railway Track Faults Detection and Localization Using Acoustic Analysis," in *IEEE Access*, vol. 10, pp. 106520-106533, 2022, doi: 10.1109/ACCESS.2022.3210326.
9. I. Gokasar, A. Timurogullari, S. S. Azkan, M. Deveci and Z. Lv, "MSND: Modified Standard Normal Deviate Incident Detection Algorithm for Connected Autonomous and Human-Driven Vehicles in Mixed Traffic," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 11, pp. 13151-13160, Nov. 2023, doi: 10.1109/TITS.2022.3190667
10. J. Marecek, S. Maroulis, V. Kalogeraki and D. Gunopulos, "Low-Rank Methods in Event Detection With Subsampled Point-to-Subspace Proximity Tests," in *IEEE Access*, vol. 10, pp. 32525-32536, 2022, doi: 10.1109/ACCESS.2022.3152206.
11. K. Choi, J. Yi, C. Park and S. Yoon, "Deep Learning for Anomaly Detection in Time-Series Data: Review, Analysis, and Guidelines," in *IEEE Access*, vol. 9, pp. 120043-120065, 2021, doi: 10.1109/ACCESS.2021.3107975.
12. L. C. Choy, J. Wu, C. Long and Y. -B. Lin, "Ubiquitous and Low Power Vehicles Speed Monitoring for Intelligent Transport Systems," in *IEEE Sensors Journal*, vol. 20, no. 11, pp. 5656-5665, 1 June1, 2020, doi: 10.1109/JSEN.2020.2974829.
13. L. F. P. de Oliveira, L. T. Manera and P. D. G. D. Luz, "Development of a Smart Traffic Light Control System With Real-Time Monitoring," in *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3384-3393, 1 March1, 2021, doi: 10.1109/JIOT.2020.3022392.

14. M. A. Khan, T. Nawaz, U. S. Khan, A. Hamza and N. Rashid, "IoT-Based Non-Intrusive Automated Driver Drowsiness Monitoring Framework for Logistics and Public Transport Applications to Enhance Road Safety," in IEEE Access, vol. 11, pp. 14385-14397, 2023, doi: 10.1109/ACCESS.2023.3244008.
15. M. H. M. Ghazali and W. Rahiman, "Vibration-Based Fault Detection in Drone Using Artificial Intelligence," in IEEE Sensors Journal, vol. 22, no. 9, pp. 8439-8448, 1 May1, 2022, doi: 10.1109/JSEN.2022.3163401.
16. N. Kumar, D. Acharya and D. Lohani, "An IoT-Based Vehicle Accident Detection and Classification System Using Sensor Fusion," in IEEE Internet of Things Journal, vol. 8, no. 2, pp. 869-880, 15 Jan.15, 2021, doi: 10.1109/JIOT.2020.3008896.
17. N. Sharma and R. D. Garg, "Real-Time IoT-Based Connected Vehicle Infrastructure for Intelligent Transportation Safety," in IEEE Transactions on Intelligent Transportation Systems, vol. 24, no. 8, pp. 8339-8347, Aug. 2023, doi: 10.1109/TITS.2023.3263271.
18. Q. Xie, X. Hu, L. Ren, L. Qi and Z. Sun, "A Binocular Vision Application in IoT: Realtime Trustworthy Road Condition Detection System in Passable Area," in IEEE Transactions on Industrial Informatics, vol. 19, no. 1, pp. 973-983, Jan. 2023, doi: 10.1109/TII.2022.3145858.
19. R. A. Khalil, N. Saeed, M. Masood, Y. M. Fard, M. -S. Alouini and T. Y. Al-Naffouri, "Deep Learning in the Industrial Internet of Things: Potentials, Challenges, and Emerging Applications," in IEEE Internet of Things Journal, vol. 8, no. 14, pp. 11016-11040, 15 July15, 2021, doi: 10.1109/JIOT.2021.3051414.
20. R. Tong, Q. Jiang, Z. Zou, T. Hu and T. Li, "Embedded System Vehicle Based on Multi-Sensor Fusion," in IEEE Access, vol. 11, pp. 50334-50349, 2023, doi: 10.1109/ACCESS.2023.3277547.

21. S. Aoki, T. Yonezawa and N. Kawaguchi, "RobotNEST: Toward a Viable Testbed for IoT-Enabled Environments and Connected and Autonomous Robots," in *IEEE Sensors Letters*, vol. 6, no. 2, pp. 1-4, Feb. 2022, Art no. 6000304, doi: 10.1109/LENS.2021.3139624.
22. S. Li, "Fall Detection With Wrist-Worn Watch by Observations in Statistics of Acceleration," in *IEEE Access*, vol. 11, pp. 19567-19578, 2023, doi: 10.1109/ACCESS.2023.3249191.
23. T. Vaiyapuri, E. L. Lydia, M. Y. Sikkandar, V. G. DÃaz, I. V. Pustokhina and D. A. Pustokhin, "Internet of Things and Deep Learning Enabled Elderly Fall Detection Model for Smart Homecare," in *IEEE Access*, vol. 9, pp. 113879-113888, 2021, doi: 10.1109/ACCESS.2021.3094243.
24. U. Alvi, M. A. K. Khattak, B. Shabir, A. W. Malik and S. R. Muhammad, "A Comprehensive Study on IoT Based Accident Detection Systems for Smart Vehicles," in *IEEE Access*, vol. 8, pp. 122480-122497, 2020, doi: 10.1109/ACCESS.2020.3006887.
25. V. Agarwal, S. Tapaswi, P. Chanak and N. Kumar, "Intelligent Emergency Evacuation System for Industrial Environments Using IoT-Enabled WSNs," in *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1-12, 2023, Art no. 9514612, doi: 10.1109/TIM.2023.3328071.

SOURCE CODE

```
void gps(); void smss();

#include <SoftwareSerial.h>

SoftwareSerial mySerial(2, 3); //RX, TX

int rxPin = 0;           // RX PIN

int txPin = 1;           // TX TX

//GPS

int byteGPS=-1;

char linea[300] = "";

char comandoGPR[7] = "$GPRMC";

int cont=0;

int bien=0;

int conta=0;

int indices[13];

char flag,flag1;

void setup()

{

    Serial.begin(9600);

    mySerial.begin(9600);

    pinMode(15,OUTPUT);

}

void loop()

{

    int MEMS=analogRead(A0);
```

```

Serial.print("M:");Serial.println(MEMS);delay(1000);

if(MEMS>400 || MEMS<200)

{

    gps();smss();delay(2000);

}

else

{

    digitalWrite(15,0);

}

}

void smss()

{

    digitalWrite(15,1);

    Serial.println("fsdfsfdsvjmbnccxgsggsddsdsgdsvsgssdffzfsff");

    mySerial.print(linea[21]);mySerial.print(linea[22]);mySerial.print(linea[23]);mySerial
    .print(linea[24]);mySerial.print(linea[25]);mySerial.print(linea[26]);mySerial.print(lin
    ea[27]);

    mySerial.print(linea[28]);mySerial.print(linea[29]);mySerial.print(linea[30]);mySerial
    .print(linea[31]);mySerial.print(linea[32]);mySerial.print(linea[33]);mySerial.print(lin
    ea[34]);

    mySerial.print(linea[35]);mySerial.print(linea[36]);mySerial.print(linea[37]);mySerial
    .print(linea[38]);mySerial.print(linea[39]);mySerial.print(linea[40]);mySerial.print(lin
    ea[41]);

    mySerial.print(linea[42]);mySerial.print(linea[43]);mySerial.print(linea[44]);

    delay(1500);

    mySerial.println((char)26);

```

```

delay(2000);

Serial.println("fvuvyvfuevfvf agcyacyscyauacsfcucauucsavfvayvfvavfscfuyacfefv
gv");
mySerial.print(linea[21]);mySerial.print(linea[22]);mySerial.print(linea[23]);mySerial
.print(linea[24]);mySerial.print(linea[25]);mySerial.print(linea[26]);mySerial.print(lin
ea[27]);
mySerial.print(linea[28]);mySerial.print(linea[29]);mySerial.print(linea[30]);mySerial
.print(linea[31]);mySerial.print(linea[32]);mySerial.print(linea[33]);mySerial.print(lin
ea[34]);
mySerial.print(linea[35]);mySerial.print(linea[36]);mySerial.print(linea[37]);mySerial
.print(linea[38]);mySerial.print(linea[39]);mySerial.print(linea[40]);mySerial.print(lin
ea[41]);

mySerial.print(linea[42]);mySerial.print(linea[43]);mySerial.print(linea[44]);

delay(1500);

mySerial.println((char)26);

delay(2000);

Serial.println("bjbjsbaikjj iuvsuavdvsadsaduwwqtttyefrecf xf aj chs akj
akfvafvdjvdkvjfvdvfkjabkjfbkjabf");

mySerial.print(linea[21]);mySerial.print(linea[22]);mySerial.print(linea[23]);mySerial
.print(linea[24]);mySerial.print(linea[25]);mySerial.print(linea[26]);mySerial.print(lin
ea[27]);
mySerial.print(linea[28]);mySerial.print(linea[29]);mySerial.print(linea[30]);mySerial
.print(linea[31]);mySerial.print(linea[32]);mySerial.print(linea[33]);mySerial.print(lin
ea[34]);
mySerial.print(linea[35]);mySerial.print(linea[36]);mySerial.print(linea[37]);mySerial
.print(linea[38]);mySerial.print(linea[39]);mySerial.print(linea[40]);mySerial.print(lin
ea[41]);

mySerial.print(linea[42]);mySerial.print(linea[43]);mySerial.print(linea[44]);

delay(1500);

mySerial.println((char)26);

```

```

delay(2000);

digitalWrite(15,0);

}

void gps()
{
    byteGPS=Serial.read();

    if (byteGPS == -1)
    {
        delay(100);
    } else {
        linea[conta]=byteGPS;

        conta++;

        if (byteGPS==13)
        {
            cont=0;

            bien=0;

            for (int i=1;i<7;i++)
            {
                if (linea[i]==comandoGPR[i-1])
                {
                    bien++;
                }
            }

            if(bien==6)

```

```

{
    for (int i=0;i<300;i++){
        if (linea[i]==',')
        {
            indices[cont]=i;
            cont++;
        }
        if (linea[i]=='*')
        {
            indices[12]=i;
            cont++;
        }
    }
    Serial.println("");
    Serial.println("");
    for (int i=2;i<5;i++)
    {
        if(i==2)
        {
            for (int j=indices[i];j<(indices[i+1]-1);j++)
            {
                ((linea[j+1]));
            }
            if(i==4)

```

```

{
    Serial.print(",");
    for (int j=indices[i];j<(indices[i+1]-1);j++)
    {
        ((linea[j+1]));
    }
    smss();flag=0;
}
}
conta=0;
for (int i=0;i<300;i++){
    linea[i]=' ';
}
}
}
}

```