IOT BASED VEHICLE ACCIDENT DETECTION AND EMERGENCY ALERT SYSTEM

Submitted to

SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE

ELECTRONICS AND TELECOMMUNICATION ENGINEERING

By

Mr. Pardeshi Siddharth Manoj

Mr. Kulkarni Saideep Rahul

Mr. Kanawade Mahesh Nitin

Under The Guidance of

Mr. R.A.Kadu

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION ENGINEERING

PRAVARA RURAL EDUCATION SOCIETY

PRAVARA RURAL ENGINEERING COLLEGE, LONI
TAL-RAHATA, DIST- AHMEDNAGAR (M.S.), INDIA
413736

2024-2025

Vision Statement of College

"Enrich the youth with skills and values to enable them to contribute in the development of society: nationally and globally".

Mission Statement of College

To provide quality technical education through effective teaching-learning and research to foster youth with skills and values to make them capable of delivering significant contribution in local to global development.

Program Vision

To develop technical ability in Electronics and Telecommunication Engineering graduates by providing quality education that creates professionals for the benefit of society.

Program Mission

To educate students in the advanced technologies in the field of Electronics and Telecommunication for emphasizing technical expertise, professional attitude, ethical values and inspire the students to utilize their education for the betterment of society

Program Educational Objectives (PEOs)

PEO1	Engage in designing, testing, operating and manufacturing systems in the field of Electronics and Telecommunication Engineering
PEO2	Solve problems of allied areas by applying the knowledge of Electronics and Telecommunication Engineering
PEO3	Ability to update knowledge with emerging technologies to embrace professional and ethical attitude in multidisciplinary projects
PEO4	Work effectively as an individual and as a team member to make effective contributions to the benefit of the workplace and community

Program Specific Outcomes (PSOs)

PSO1	Ability to exhibit competency in the areas of Electronics and Telecommunication Engineering like Electronic Circuits and Communication.
PSO2	Design and implement the Embedded systems to resolve societal and industrial problems using modern engineering hardware and software tools.
PSO3	Ability to inculcate software proficiency skills for Industry

Program Outcomes

POs	Description of POs
1	**Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2	**Problem analysis:** Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3	**Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4	**Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5	**Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6	**The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7	**Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts and demonstrate the knowledge of, and need for sustainable development.
8	**Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9	**Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10	**Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11	**Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12	**Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Course Outcomes

CO1	Identify the problem statement based on interested domain in recent trends.
CO2	Apply engineering knowledge for the selection of appropriate software, hardware and development of circuit diagram and software to solve the identified problem.
CO3	Develop problem-solving skills by identifying and resolving issues encountered during the testing of Mini Project.
CO4	Prepare a technical report based on the Mini project.
CO5	Develop communication and presentation skills by effectively deliver technical seminar based on the Mini Project work.

PRAVARA RURAL EDUCATION SOCIETY's

PRAVARA RURAL ENGINEERING COLLEGE, LONI

Affiliated to Savitribai Phule Pune University, Pune

CERTIFICATE

This is to certify that the mini project Report on **: IOT Based Vehicle Accident Detection and Emergency Alert System** has been successfully carried out mini-project project report is submitted **by Mr.** Pardeshi Siddharth Manoj**, Mr.** Kulkarni Saideep Rahul **, Mr.** Kanawade Mahesh Nitin the students of the third year in Electronics and Telecommunication Engineering during the Academic year 2024-25.

Mini Project Guide   Head of Department

ACKNOWLEDGEMENT

Throughout the journey of this work till the date, we realized more strongly how much selfless effort. So, we should not forget them while praising flower. It is matter of gratification for us to pay our respects and acknowledgements to all those who have imparted knowledge and help us to complete our final mini project report.

We would like to acknowledge the great contribution and support received in this endeavor from our project guide Mr. R.A.Kadu for this depth knowledge, guidance and inspiration for will be of great importance to clarify and kind of problem likely to us in the future.

Mr. Pardeshi Siddhart Manoj

Mr. Kulkarni Saideep Rahul

Mr. Kanawade Mahesh Nitin

ABSTRACT

This project presents an IoT-based vehicle accident detection and emergency alert system designed to improve road safety. The system utilizes an MPU6050 accelerometer to detect sudden motion or impact, a NEO-6M GPS module to obtain location data, and a SIM800L GSM module to send emergency SMS alerts. Additionally, an LCD display shows system status messages, and a push button allows the user to cancel alerts in case of no injury. The system is cost-effective, scalable, and designed for rapid emergency response. It ensures that accident victims receive timely medical help, especially in areas where manual intervention may be delayed or unavailable.

This project introduces an intelligent and low-cost IoT-based system capable of detecting vehicle collisions in real time and notifying emergency contacts with precise GPS coordinates using GSM technology.

It integrates sensors like the MPU6050 accelerometer and modules such as the NEO-6M GPS and SIM800L GSM with a user-friendly LCD display, providing both automated alerting and manual control through a “No Casualty” button, enhancing safety and minimizing false alarms.

INDEX

Chapter 1
INTRODUCTION

1. Introduction

As vehicular accidents continue to rise worldwide, the consequences can be devastating, often resulting in serious injuries or fatalities. In many cases, the delay in emergency response time further escalates the severity of the incident. Traditional accident reporting systems rely heavily on human intervention—either from the victim or a passerby—to inform emergency services. This reliance leads to delayed assistance, especially in remote or less populated areas where immediate help is not readily available.

In today's technologically advanced world, the integration of Internet of Things (IoT) solutions in transportation systems presents a new opportunity to mitigate such issues. IoT-based accident detection systems can automatically identify collisions, determine the location of the incident, and notify emergency services or contacts without any human involvement. These systems improve the response time dramatically and potentially save lives by enabling quicker medical assistance.

This project is designed to create a functional prototype of an IoT-based vehicle accident detection and emergency alert system. It uses a combination of sensors and modules including the MPU6050 for motion detection, the NEO-6M for GPS location tracking, and the SIM800L for sending real-time SMS alerts. An LCD display is included to provide status updates to the user, and a manual override button gives the driver the ability to cancel the alert if there is no injury. This user-friendly and cost-effective system demonstrates the potential to revolutionize accident response, making our roads safer and smarter.

2. Motivation Of Project

In many developing and even developed countries, delays in emergency response to vehicle accidents have led to preventable injuries and fatalities. Often, bystanders are hesitant or unable to contact emergency services in time, and victims themselves may be unconscious or immobilized. These situations demand a system that can act autonomously and immediately to notify the appropriate authorities or emergency contacts.

With the rise of IoT-enabled technologies, there is a growing potential to leverage connected sensors and communication modules to address this issue. By automating the process of accident detection and response, such systems can bridge the gap between incident occurrence and emergency intervention.

This project is driven by the motivation to create a solution that not only reduces emergency response time but also eliminates the reliance on third-party reporting. It aims to empower vehicles with the ability to self-report critical incidents, ensure accurate GPS tracking, and support manual overrides to prevent false alarms. The ultimate goal is to build a cost-effective, accessible system that can be implemented in any vehicle, regardless of its make or model, and save lives by delivering timely assistance.

3. Aim of Project

To design and implement an IoT-based system that detects vehicle accidents and automatically sends SMS alerts with location details to multiple emergency contacts. This system aims to be accurate, timely, cost-efficient, and user-friendly.

4. Objectives Of Project

- Detect accidents using real-time motion data from an MPU6050 accelerometer.
- Use the NEO-6M GPS module to determine the exact location of the vehicle at the time of the accident.
- Utilize the SIM800L GSM module to send alert messages to multiple emergency contacts.
- Include a manual override through a push button that cancels alerts if no injury has occurred.
- Provide real-time feedback and system status via a 16x2 I2C LCD display.

5. Existing Systems and Drawbacks

Advanced accident detection systems like OnStar by General Motors or eCall in European vehicles provide built-in emergency alert mechanisms. These systems are typically embedded into premium models and rely on proprietary technology. While effective, they are expensive and inaccessible for most users in developing nations.

Additionally, many of these systems lack the ability for users to intervene or cancel alerts in the event of a false alarm, leading to misuse of emergency resources. Our proposed system aims to fill these gaps by offering an affordable, user-centric, and easily customizable solution that integrates a manual override, real-time feedback, and open-source technology.

6. Proposed System

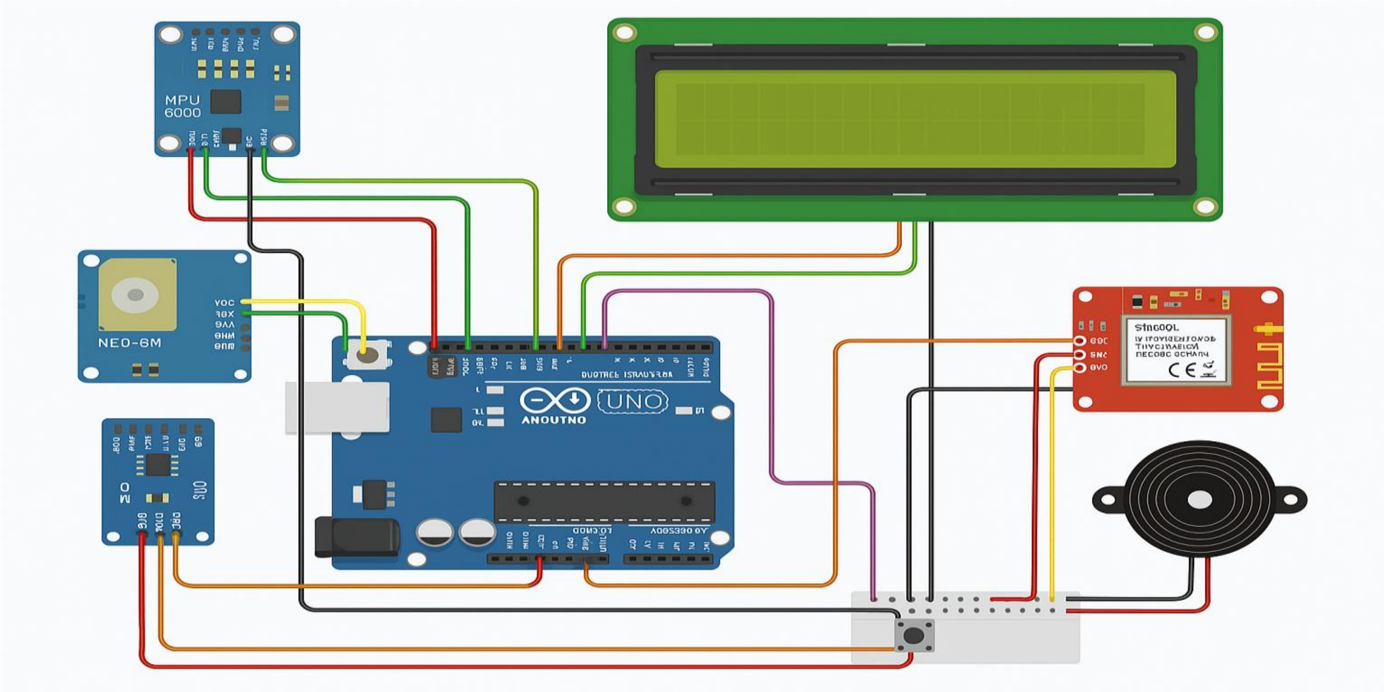
The proposed IoT-based system consists of a set of sensors and modules connected to an Arduino Uno microcontroller. The MPU6050 motion sensor constantly monitors vehicle movements. Upon detecting a sudden or abnormal change in acceleration that might signify a collision, the system triggers a 30-second timer during which the user may cancel the alert using a push button.

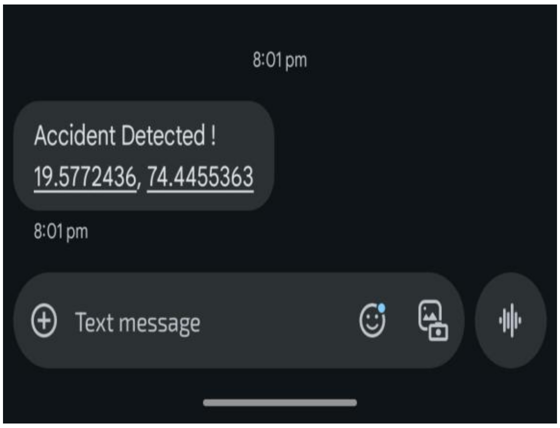
If no intervention is detected within this grace period, the NEO-6M GPS module captures the current location coordinates, and the SIM800L GSM module dispatches SMS alerts to multiple predefined emergency contacts. The LCD display updates continuously with system messages such as "System Ready," "Accident Detected," and "Alert Sent/Canceled," helping the user stay informed. The buzzer provides audible feedback during emergency states, increasing situational awareness.

Chapter 2

Theoretical Details of Project

1. Circuit Diagram





2. Description of Components

Arduino Uno Microcontroller :-

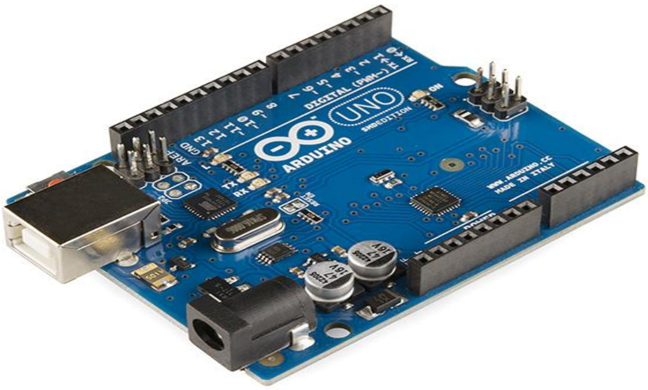


Fig. Arduino UNO

Acts as the central processor that manages input signals from various sensors and executes the alert logic. It is based on the ATmega328P microcontroller and provides digital and analog I/O pins, making it suitable for interfacing multiple modules simultaneously.

Some of the key features of the Arduino Uno include:

- Open-source design. Large community at arduino.cc/forum/ of people using and troubleshooting it.
- Easy USB interface. The chip on the board plugs straight into your USB port and registers on your computer as a virtual serial port. This allows us to serially communicate which is an extremely easy protocol
- Convenient power management and built-in voltage regulation. 12v can easily beregulated to both 5v and 3.3v
- Easy-to-find and cheap, microcontroller
- Countless number of hardware features like timers, PWM pins, external and internal interrupts, and multiple sleep modes.

MPU6050 Sensors :-

Fig.MPU6050 Sensor

The MPU6050 is a MEMS (Micro-Electro-Mechanical Systems) based 6-axis sensor that combines a 3-axis gyroscope and a 3-axis accelerometer. It is capable of detecting linear acceleration and angular velocity. In this project, it monitors real-time motion and detects sudden jerks, which could indicate a vehicular collision. Its I2C communication protocol makes it easy to interface with the Arduino. Sensitivity thresholds can be adjusted in the code for accurate detection.

Neo 6M GPS Module :-

Fig. MPU6050 Sensor

This GPS module provides accurate real-time position data, including latitude and longitude coordinates. It operates using the UART serial communication protocol and includes a built-in ceramic antenna for signal reception. The module can achieve a positional accuracy of around 2.5 meters, making it suitable for identifying accident locations precisely. It is capable of saving the latest GPS fix in its internal memory, which can be retrieved after a power failure.

SIM900L GSM Module :-

Fig. SIM900L GSM Module

The SIM800L is a compact GSM module used for SMS-based communication. It supports quad-band frequencies and is controlled through AT commands sent via a serial interface. It allows the system to send emergency text messages containing GPS coordinates to pre-defined emergency contacts. A stable power supply (3.7V–4.2V) is essential for its proper functioning, and an external antenna is typically used to improve network reception.

LCD Display (16x2 with I2C Interface):-

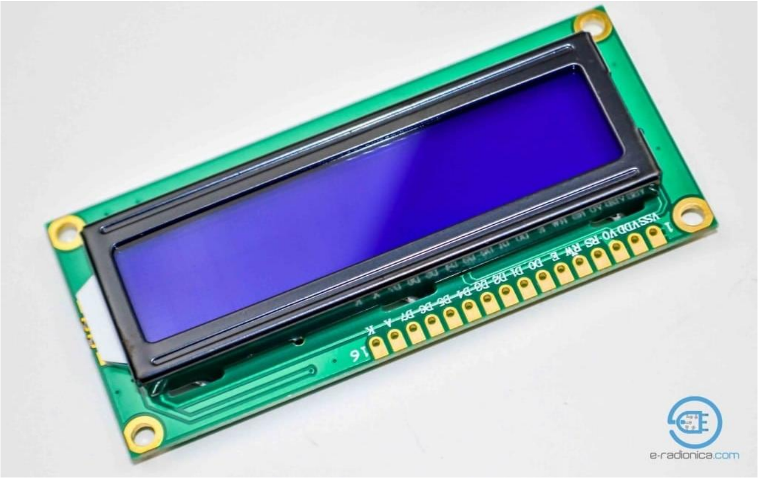


Fig. LCD Display

The 16x2 LCD module is used to display system messages to the user. The I2C interface board attached to the LCD allows communication with just two data lines (SDA and SCL), freeing up other digital pins for other components. It provides feedback like "System Ready," "Accident Detected," and "Alert Sent," helping users understand the current state of the system in real-time.

Push Button:-

Fig. Push Button

This push button serves as a manual override that enables the user to cancel the alert message in case of a false trigger or a minor collision with no injuries. It is connected to one of the digital pins on the Arduino and is typically used with a pull-down resistor to maintain signal stability. The system waits for a predefined time (e.g., 30 seconds) for the user to press the button before sending the SMS.

Buzzer:-

Fig. Buzzer

The buzzer acts as an audible alert device that beeps when an accident is detected. This feature alerts nearby people and draws attention to the vehicle, especially in remote or low-visibility areas. It is also a useful tool during testing, helping confirm that the sensor logic is functioning correctly.

Breadboard and Jumper Wires:-

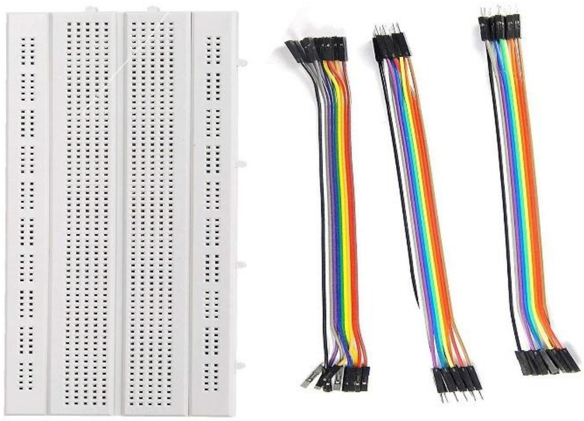
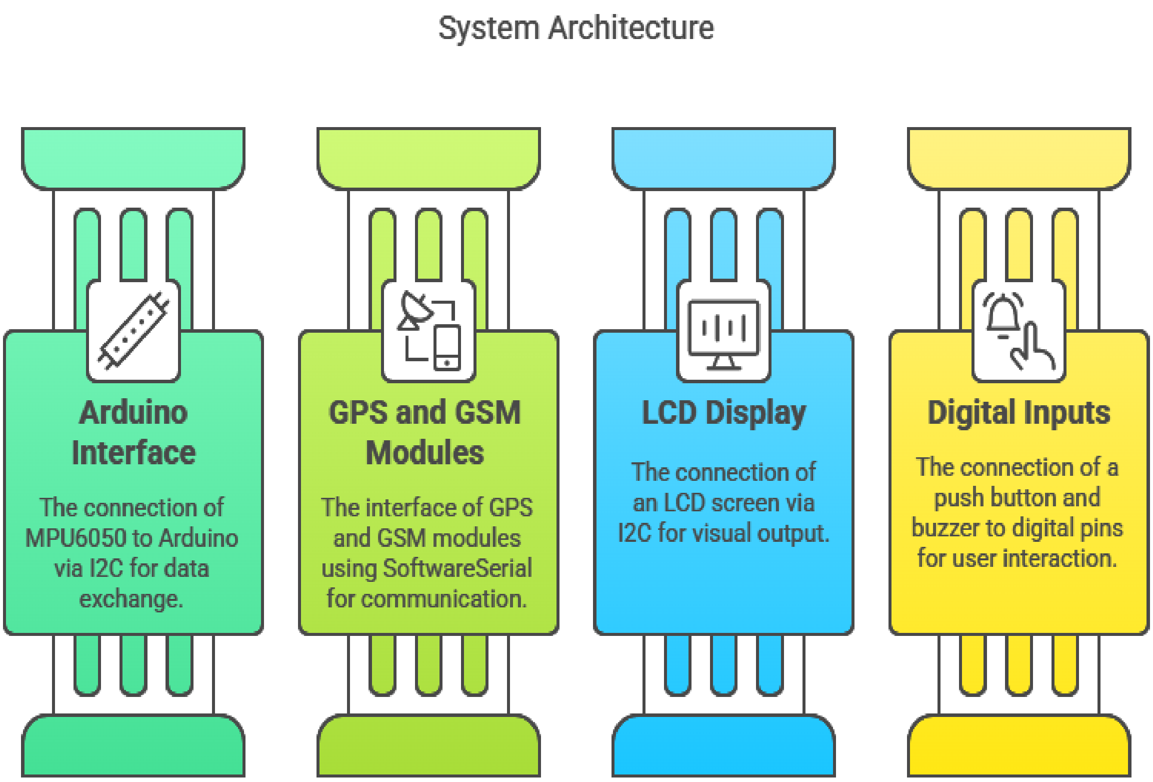


Fig. Breadboard and Wires

These are essential tools for prototyping. The breadboard allows for quick assembly and reconfiguration of the circuit without soldering, which is useful during testing and debugging. Jumper wires facilitate clean and organized connections between the modules and the Arduino. For the final deployment, the components can be soldered onto a PCB or perf board for durability.

3. System Architecture


System Architecture
Arduino Interface
The connection of MPU6050 to Arduino via I2C for data exchange.
GPS and GSM Modules
The interface of GPS and GSM modules using SoftwareSerial for communication.
LCD Display
The connection of an LCD screen via I2C for visual output.
Digital Inputs
The connection of a push button and buzzer to digital pins for user interaction.

Chapter 3

SOFTWARE & CODE

1. Software Details

- **Arduino IDE:** Used for programming and uploading code to the Arduino Uno board. Core libraries include Wire.h for I2C, SoftwareSerial.h for GSM/GPS communication, and TinyGPS++ for GPS data parsing.
- **Fritzing:** Used to visually design and share the hardware circuit layout.
- **Proteus (Optional):** Can be employed for pre-implementation simulation and verification of circuit functionality.

2. Code Section:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <Adafruit_MPU6050.h>
#include <Adafruit_Sensor.h>
#include <SoftwareSerial.h>
#include <TinyGPS++.h>  // Use TinyGPS++ for better GPS parsing

// LCD (16x2)
LiquidCrystal_I2C lcd(0x27, 16, 2);

// MPU6050
Adafruit_MPU6050 mpu;

// SoftwareSerial: GPS (RX->3, TX->4), GSM (RX->8, TX->7)
SoftwareSerial gpsSerial(4, 3);
SoftwareSerial gsmSerial(7, 8);
TinyGPSPlus gps;

// Pins
#define NO_CASUALTY_BUTTON 5
#define BUZZER 6
```

```
// Emergency Contacts
const char* emergencyNumbers[] = {"+911234567890", "+919876543210"};
const int numContacts = 2;

bool accidentDetected = false;

void setup() {
  Serial.begin(9600);
  gpsSerial.begin(9600);
  gsmSerial.begin(9600);
  delay(3000);

  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("System Starting");

  Wire.begin();
  if (!mpu.begin()) {
    lcd.setCursor(0, 1);
    lcd.print("MPU6050 Error");
    while (1);
  }

  pinMode(NO_CASUALTY_BUTTON, INPUT_PULLUP);
  pinMode(BUZZER, OUTPUT);
  digitalWrite(BUZZER, LOW);

  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Initialized");

  gsmSerial.println("AT");
```

```
  delay(1000);
  while (gsmSerial.available()) {
    Serial.write(gsmSerial.read());
  }
}

void loop() {
  // Read GPS in background
  while (gpsSerial.available()) {
    gps.encode(gpsSerial.read());
  }

  // MPU6050 readings
  sensors_event_t a, g, temp;
  mpu.getEvent(&a, &g, &temp);

  if (!accidentDetected && (abs(a.acceleration.x) > 2 || abs(a.acceleration.y) > 2 || abs(a.acceleration.z)
> 2)) {
    accidentDetected = true;
    lcd.setCursor(0, 0);
    lcd.print("Accident Detected");
    digitalWrite(BUZZER, HIGH);
    sendAlert();
    delay(5000);
  }

  if (accidentDetected && digitalRead(NO_CASUALTY_BUTTON) == LOW) {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("No Casualty");
    sendNoCasualtyAlert();
    digitalWrite(BUZZER, LOW);
    accidentDetected = false;
  }
```

```
  delay(500);
}

void sendAlert() {
  String location = getGPSLocation();
  for (int i = 0; i < numContacts; i++) {
    gsmSerial.println("AT+CMGF=1");
    delay(500);
    gsmSerial.print("AT+CMGS=\"");
    gsmSerial.print(emergencyNumbers[i]);
    gsmSerial.println("\"");
    delay(500);
    gsmSerial.print("Accident Detected! Location: ");
    gsmSerial.print(location);
    gsmSerial.write(26); // Ctrl+Z
    delay(5000);
  }
}

void sendNoCasualtyAlert() {
  for (int i = 0; i < numContacts; i++) {
    gsmSerial.println("AT+CMGF=1");
    delay(500);
    gsmSerial.print("AT+CMGS=\"");
    gsmSerial.print(emergencyNumbers[i]);
    gsmSerial.println("\"");
    delay(500);
    gsmSerial.print("No Casualty. Ignore Alert.");
    gsmSerial.write(26);
    delay(5000);
  }
}
```

```
String getGPSLocation() {
  if (gps.location.isValid()) {
    String lat = String(gps.location.lat(), 6);
    String lng = String(gps.location.lng(), 6);
    return "Lat:" + lat + ", Lon:" + lng;
  } else {
    return "GPS Unavailable";
  }
}
```

Chapter 4
Advantages & Applications

Advantages:

- **Automatic Accident Detection:**
  Detects accidents in real-time without the need for human intervention.
- **Instant Emergency Alerts:**
  Sends SMS alerts with GPS coordinates to multiple emergency contacts immediately.
- **Reduces Emergency Response Time:**
  Helps ambulance and rescue teams reach the accident location faster, potentially saving lives.
- **Low-Cost and Scalable:**
  Built using affordable components, making it suitable for large-scale deployment in vehicles.
- **User Confirmation Feature:**
  Includes a "No Casualty" button to avoid sending unnecessary alerts when the driver is safe.
- **Portable and Easy to Integrate:**
  Can be easily added to both new and existing vehicles.

Applications:

- **Private Vehicles:**
  Can be installed in cars, motorcycles, or scooters to ensure driver safety.
- **Public Transport:**
  Useful in buses, taxis, and school vans to monitor accidents and ensure passenger safety.
- **Logistics and Fleet Management:**
  Transport companies can track vehicle safety and respond quickly in case of crashes.
- **Remote Area Vehicles:**
  Especially helpful where medical help is far away or delayed.
- **Smart City Integration:**
  Can be a part of larger smart transportation or road safety systems in smart cities.

Chapter 5

RESULTS & CONCLUSION

Result

The system was tested under various simulated conditions that mimicked vehicular impact. When the MPU6050 detected a sudden jerk or shock, the LCD displayed "Accident Detected," and the countdown began. If the button was not pressed, the GPS module collected the coordinates and the GSM module successfully transmitted SMS alerts to the predefined contacts.

In cases where the user pressed the button within the 30-second window, the system canceled the alert and displayed "Alert Canceled" on the screen. The buzzer effectively signaled emergency states, validating the alert mechanism. Overall, the system performed as expected in multiple test cases.

Conclusion

This project effectively demonstrates a functional, affordable, and efficient IoT-based vehicle accident detection and emergency alert system. It bridges the critical gap between accident occurrence and response time, especially in regions lacking adequate human monitoring or fast-responding emergency services. Its real-time alerting capability, ease of deployment, and adaptability make it suitable for mass adoption.

In future iterations, enhancements could include GSM call capabilities, app-based interfaces, or AI for accident pattern prediction. This would further improve reliability and utility. The successful implementation highlights the transformative role of IoT in enhancing safety and saving lives on the road.

References

1. **MPU6050 Datasheet**
   - Provides technical specifications, electrical characteristics, communication protocols (I2C), and application guidelines for using the MPU6050 accelerometer and gyroscope module. It was referenced for setting the correct sensitivity thresholds for accident detection and understanding its motion sensing capabilities.
2. **NEO-6M GPS Module Datasheet**
   - Offers detailed information about the NEO-6M GPS receiver module, including its pin configuration, communication method (UART), accuracy, and setup instructions. It was used to understand how to retrieve latitude and longitude coordinates efficiently and how to interface the module with Arduino.
3. **SIM800L GSM Module Datasheet**
   - Describes the working of the SIM800L GSM module, AT command sets, power requirements, and network registration procedures. This helped in coding the SMS sending part, handling network delays, and managing SMS formats while maintaining reliable communication.
4. **Arduino Documentation**
   - Official Arduino website and tutorials provided valuable resources for programming the Arduino Uno, interfacing different sensors and modules, handling serial communication, and debugging issues during project development. It was particularly useful for learning about SoftwareSerial, LCD libraries, and power management.
5. **IoT Research Articles and Academic Papers**
   - Several research papers and articles related to IoT-based accident detection and smart emergency response systems were reviewed. These papers discussed existing solutions, challenges faced in real-world implementation, optimization techniques, and future scope, which influenced the design and improvement of the proposed system.
6. **Books on IoT and Embedded Systems**
   - Textbooks like *"Internet of Things: Principles and Paradigms"* and *"Embedded Systems: Introduction to ARM Cortex-M Microcontrollers"* provided theoretical background and practical considerations for building real-time, embedded IoT systems.
7. **Online Forums and Tutorials (Arduino, Stack Overflow, Instructables)**
   - Various community-driven platforms like Stack Overflow, Arduino forums, and Instructables tutorials offered solutions to common errors faced during interfacing GPS

and GSM modules. Community insights helped optimize sensor readings, manage serial communication conflicts, and handle module initialization failures.

8. **Datasheets of Supporting Components (LCD 16x2, Push Button, Buzzer)**
    - Technical datasheets and manuals for basic components such as LCDs, buzzers, and push buttons were consulted to understand operating voltages, interfacing methods, and best practices for stable integration with the Arduino Uno.