ABSTRACT

Rapid and reliable accident detection has become a critical requirement due to the rising number of road fatalities and delays in emergency response systems. Manual reporting of accidents is often inefficient, especially in remote areas, leading to severe injuries or loss of life. This project proposes a **smart Accident Detection and Emergency Alert System** capable of autonomously identifying collision events and instantly notifying rescue authorities. The system incorporates an **accelerometer (or gyroscope/impact sensor)** to continuously monitor the vehicle's motion dynamics, detecting sudden changes in acceleration that exceed pre-defined crash thresholds. A **microcontroller unit** processes this sensor data in real time and intelligently distinguishes between normal vibrations and actual collision signatures using a threshold-based algorithm.

Once an accident is confirmed, the system automatically activates a **GPS module** to obtain the precise geographical coordinates of the incident location. Simultaneously, a **GSM module** is triggered to transmit an alert message containing the victim's exact latitude, longitude, time of accident, and vehicle identification details to emergency contacts, medical services, or nearby authorities. To reduce false alarms, a **manual override/cancel button** is included, allowing the user to abort accidental triggers within a specified grace period. Additional features such as a **buzzer for alert indication**, **data logging**, and **real-time tracking** can also be integrated depending on the application.

Designed to be compact and low-cost, the proposed system is suitable for two-wheelers, cars, commercial fleets, and even industrial vehicles. Its ability to drastically reduce emergency response time can significantly improve survival rates and enhance road safety. The solution aligns with smart transportation initiatives and demonstrates strong potential for implementation in modern Intelligent Transportation Systems (ITS).

LIST OF CONTENTS

CHAPTER 1

INTRODUCTION

Road safety has become a major concern worldwide due to the continuously rising number of traffic accidents. Every year, thousands of lives are lost because victims do not receive timely medical assistance. In many cases, the delay occurs not because help is unavailable, but because the accident is not reported quickly enough. This is especially true in remote or rural areas where there is limited human presence or poor communication facilities. Therefore, an automated and reliable accident detection mechanism has become essential to reduce fatalities and prevent severe injuries.

The **Accident Detection and Emergency Alert System** is designed to address this problem by automatically sensing a collision and immediately informing emergency contacts. The system uses sensors such as an accelerometer or gyroscope to monitor the motion of the vehicle. When a sudden impact or abnormal acceleration is detected, the system interprets it as a potential accident. To confirm the event and prevent false triggers caused by normal road bumps or speed breakers, the microcontroller applies threshold-based verification.

Once the system confirms an accident, a **GPS module** acquires the real-time location of the vehicle, and a **GSM module** sends an alert message containing the exact coordinates to predefined contacts such as family members, ambulance services, or nearby hospitals. A manual override or cancel switch is included to give the user a chance to stop the alert in case of false detection. This ensures accuracy and preserves the reliability of the system.

By significantly reducing the time taken to report an accident and locate the victim, this system can greatly improve the chances of survival. The solution is cost-effective, easy to implement, and suitable for bikes, cars, commercial vehicles, and even autonomous transport systems. The project aligns with modern Intelligent Transportation System (ITS) initiatives and supports safer mobility with the help of embedded electronics and communication technology.

CHAPTER-2

LITERATURE SURVEY

Several research works have explored different approaches to accident detection using embedded systems, IoT, and smart sensors.

1. **Sensor-Based Systems:**

   Many studies utilize accelerometers, gyroscopes, and vibration sensors to detect abnormal acceleration patterns indicating a crash. These methods are efficient, low-cost, and easy to implement.

2. **Smartphone-Based Solutions:**

   Mobile sensors (accelerometer + GPS) are used to detect accidents; however, they rely heavily on user interaction, battery life, and cellular network availability.

3. **IoT-Enabled Systems:**

   Advanced systems integrate cloud platforms, GPS modules, and GSM communication to send alerts. These solutions offer scalability but require network availability and higher cost.

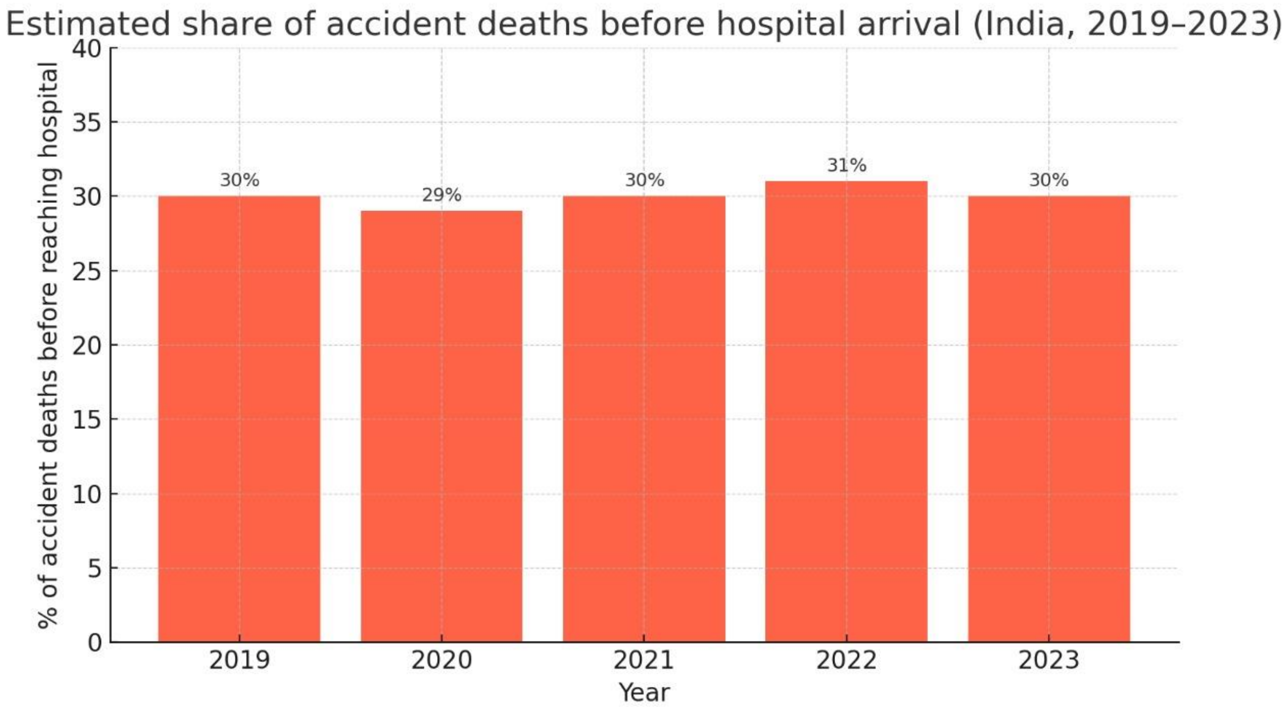
4. **Machine Learning-Based Detection:**

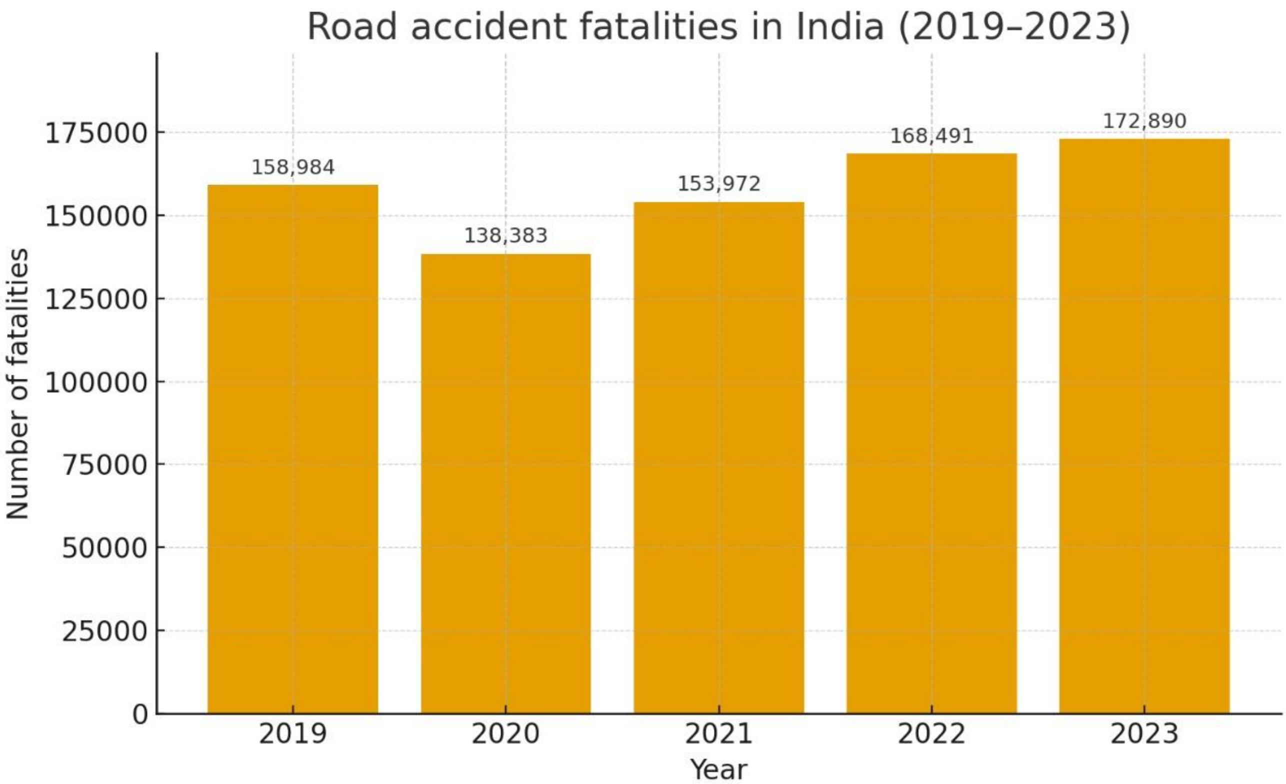
   Some works propose ML models to classify normal vs crash events. These solutions are accurate but computationally heavy and not ideal for low-cost real-time implementation.

Gap Identified:

Most existing systems either depend on user interaction, require expensive hardware, or suffer from false triggering. The proposed system focuses on reliability, affordability, and immediate emergency communication using simple sensors and embedded modules.

ROAD ACCIDENT DEATHS RECENT REPORT



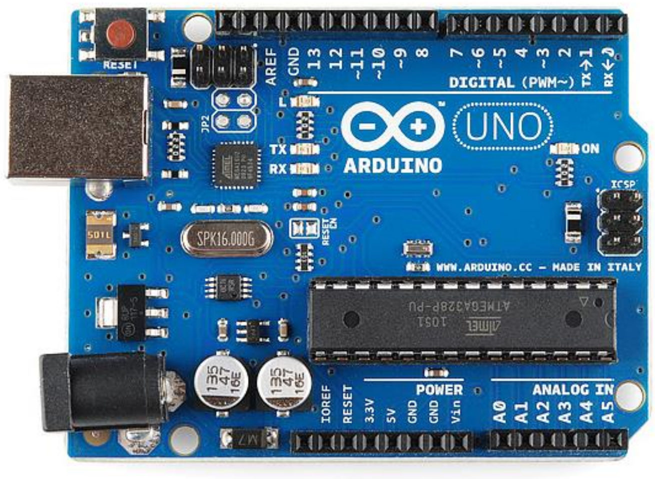


CHAPTER-3

SYSTEM DESIGN AND METHODOLOGY

3.1 COMPONENTS USED

- **Microcontroller:** Arduino UNO



Key Points of Arduino

Since you prefer a non-table format, here are the essential key points about the Arduino platform, designed for clarity and quick reading:

1. Open-Source Ecosystem

Arduino is fundamentally an open-source platform. This means that both the hardware designs (schematics) and the software (the Arduino IDE) are free and publicly available. This philosophy fosters a massive global community, allowing anyone to manufacture compatible boards, contribute code, and share knowledge freely.

2. The Microcontroller Core

At the heart of every Arduino board is a microcontroller (like the ATmega328P chip on the Uno). This is a tiny, programmable computer that reads the program (called a "sketch") and executes the instructions to control connected components.

3. High Accessibility

The platform was specifically designed for beginners, artists, and educators. Its primary goal is to make complex electronics and programming accessible to people with little to no prior technical background.

4. Simplified Programming

Arduino uses a simplified version of the C/C++ programming language, often referred to as the Arduino Language. It's easy to write because it uses straightforward functions (e.g., digitalWrite() to turn a pin on or off) and is compiled and uploaded using the user-friendly Arduino IDE (Integrated Development Environment).

5. Interaction through I/O Pins

Arduino boards feature dedicated Input/Output (I/O) pins that serve as the interface to the real world.

- Digital Pins handle simple ON/OFF signals (like reading a button press or lighting an LED).
- Analog Pins can read a range of variable values from sensors (like temperature or light intensity).

6. The Shield and Library Ecosystem

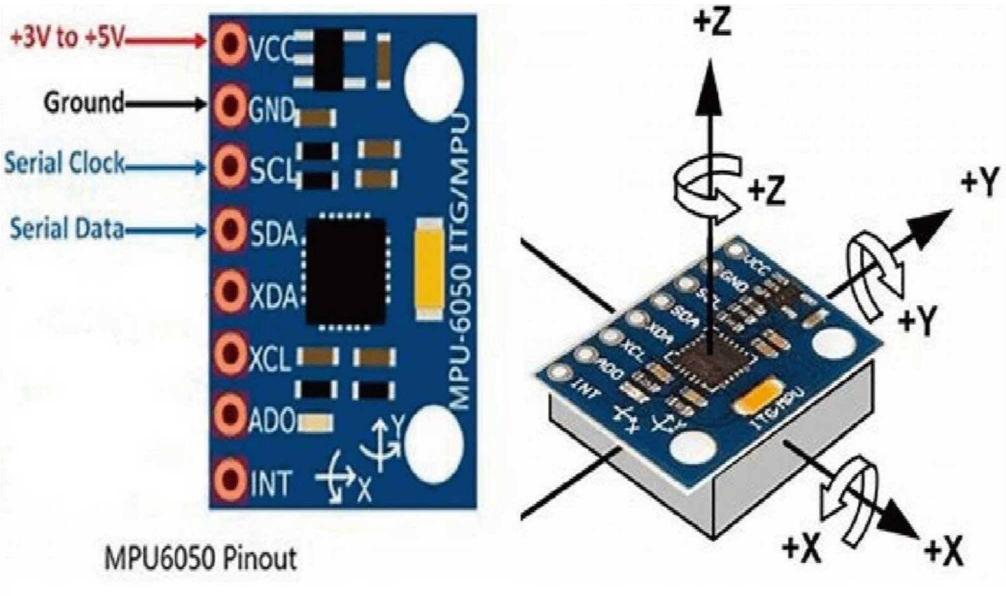
To quickly add functionality, the Arduino platform relies on two major components:

- Shields: These are pre-made circuit boards that plug directly on top of the main Arduino board to add specific features (e.g., an Ethernet Shield for networking or a Motor Shield for robotics).
- Libraries: These are collections of pre-written code that let you easily control complex components (like LCD screens or specific sensors) without having to write all the low-level code yourself.

7. Vast Application Range

Due to its versatility, Arduino is used for an enormous range of projects, including: home automation, robotics, interactive art installations, educational tools, and rapid prototyping in engineering.

Accelerometer / Gyroscope: MPU6050



MPU-6050 Key Points

1. 6-Axis Motion Tracking (IMU)

- What it is: The MPU-6050 is a single-chip Inertial Measurement Unit (IMU) that provides 6 Degrees of Freedom (6-DOF).
- Components: It integrates a 3-axis Accelerometer (measures linear acceleration/gravity) and a 3-axis Gyroscope (measures angular velocity/rate of rotation) on a single silicon die.

2. Digital Motion Processor (DMP)

- Onboard Processing: The chip includes an embedded Digital Motion Processor (DMP).
- Function: The DMP is designed to run complex sensor fusion algorithms (like calculating Yaw, Pitch, and Roll angles) directly on the chip, significantly reducing the computational load on the host microcontroller (like the Arduino).
- Output: It can output orientation data as Quaternions or Euler angles (Yaw/Pitch/Roll).

3. Communication Protocol

- Interface: It communicates primarily using the I2C (Inter-Integrated Circuit) serial protocol, requiring only two data lines (SDA and SCL) plus power and ground.
- Address: Its I2C address is typically 0x68 (can be changed to 0x69 using the AD0 pin).

4. Data and Output

- High Resolution: It features 16-bit Analog-to-Digital Converters (ADCs) for highly accurate measurement on all six axes.
- Units: Raw output is converted via libraries to:
- Temperature: It also includes an embedded temperature sensor.

5. Application & Limitations

- Core Use: Essential for orientation-sensitive projects like drones/quadcopters, self-balancing robots, gimbals, and gesture-controlled interfaces.
- Yaw Drift: As a 6-axis sensor (lacking a magnetometer), the calculated Yaw (rotation around the vertical axis) will drift over time. Achieving stable, absolute Yaw requires integrating an external magnetometer (e.g., using a 9-DOF sensor like the MPU-9250 or a separate compass chip).

GPS Module: Neo-6M

1. Core Functionality (Position, Velocity, Time)

- Purpose: To receive signals from GPS satellites to calculate its precise Position (Latitude and Longitude), Velocity (Speed and Heading), and Time (UTC time). This is often called PVT data.
- Performance: It offers high sensitivity (tracking up to -162 dBm), which helps maintain a GPS Fix even under challenging conditions.
- Accuracy: Under good conditions, it provides a typical horizontal positional accuracy of approximately 2.5 meters.

2. Communication and Interfacing

- Protocol: It communicates with the Arduino using the UART (Serial) protocol via its TX (Transmit) and RX (Receive) pins.
- Baud Rate: The default serial communication speed is typically 9600 baud.
- Data Format: The position data is output in the standard NMEA (National Marine Electronics Association) sentence format, which consists of comma-separated ASCII text strings (e.g., $GPGGA, $GPRMC).

3. Key Components and Features

- Built-in Memory: It often includes an onboard EEPROM or battery-backed RAM (with a small lithium battery or capacitor) to store configuration data and the latest satellite almanac.
- Cold/Hot Start: Storing the almanac allows for a much faster Hot Start (Time-To-First-Fix, TTFF, in $\sim 1\text{ second}$) compared to a Cold Start (TTFF in $\sim 38\text{ seconds}$).
- Antenna: Most modules come with a ceramic patch antenna, and sometimes a dedicated connector for an external active antenna for better reception.

4. Use with Arduino

- Libraries: The TinyGPS++ library is the most widely used Arduino library for the Neo-6M. It handles parsing the raw NMEA strings and converting them into usable data types (floats for coordinates, time objects, etc.).

- Connection: For boards like the Arduino Uno that have only one hardware serial port, the module is usually connected using a Software Serial library on different digital pins to free up the main hardware serial port for debugging.

GSM Module: SIM800L

1. Primary Functions (2G Connectivity)

- **GSM & GPRS:** The module supports **Quad-band GSM/GPRS (2G)** networks (850/900/1800/1900 MHz), allowing it to work globally, provided a 2G network is available.
- **Communication Features:** It enables:
  - **SMS Messaging:** Sending and receiving text messages.
  - **Voice Calls:** Making and receiving phone calls (requires external microphone and speaker).
  - **GPRS Data:** Basic internet connectivity (using TCP/IP and HTTP) for sending and receiving small amounts of data.

2. Communication Protocol

- **UART Serial:** It interfaces with the Arduino using the **UART (Serial)** protocol via its **TX** (Transmit) and **RX** (Receive) pins.
- **AT Commands:** The module is controlled entirely by sending it **AT (Attention) commands** (e.g., AT+CMGS for sending SMS, AT+CSQ for checking signal quality).

- **Baud Rate:** The default communication speed is typically **9600 bps**, but it supports auto-baud detection up to 115200 bps.

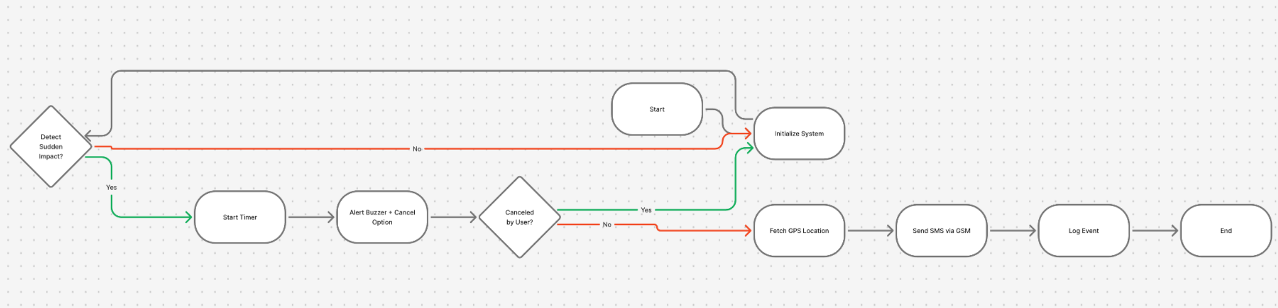
3. Critical Power Requirements (The Biggest Challenge)

- **Voltage Range:** The module requires a very specific, narrow operating voltage range: **3.4V to 4.4V DC** (4.0V recommended). **Connecting it directly to an Arduino's 5V pin can damage it.**
- **Current Spikes:** This is the most crucial point. During a transmission burst (e.g., making a call or sending data), the current draw can momentarily **peak up to 2 Amps (2A)**.
- **Power Solution:** The Arduino's onboard 3.3V or 5V regulator **cannot** supply this current spike. A separate, stable, high-current power source (like a regulated **4V switching power supply** or a direct **3.7V Li-ion battery**) is **mandatory** for reliable operation. A large capacitor (e.g., $1000 \mu\text{F}$) is often added across the power pins to smooth out the current spikes.

4. Integration with Arduino

- **Logic Level Shifting:** Because the SIM800L works at 3.3V/4.0V logic and the Arduino Uno works at 5V logic, the **Arduino's 5V TX pin must be level-shifted down** to the SIM800L's RX pin (usually using a voltage divider or logic level converter) to prevent damage.
- **SoftwareSerial:** Since the Arduino Uno only has one hardware serial port, the **SoftwareSerial** library is typically used to create a second serial port for communicating with the SIM800L.
- **Antenna:** It requires an external antenna (often a small coil or a flexible PCB antenna) connected via a U.FL connector or solder pad for network reception.

FLOW DIAGRAM



WORKING PROCEDURE

1. System Power-On

When the system is powered on:

- The microcontroller initializes all modules (GSM, GPS, MPU-6050 accelerometer, ultrasonic sensor, LCD).
- GSM network registration is verified using AT commands.
- The MPU-6050 accelerometer is calibrated by averaging initial readings.
- GPS module is activated and starts searching for satellite lock.

The LCD displays the initialization status during this process.

2. Continuous Monitoring

Once initialized, the system enters the monitoring mode. The microcontroller performs these tasks continuously:

(a) Acceleration Monitoring

- The MPU-6050 accelerometer measures acceleration along X, Y, and Z axes.
- These readings are compared with pre-calibrated offset values.
- Any sudden deviation beyond the threshold (±1000 raw units ≈ 0.1–0.2g) indicates a possible crash.

(b) Obstacle Detection

- The ultrasonic sensor (HC-SR04) measures the distance from the front of the vehicle.
- If an object is detected within 10 cm, a warning is shown on the LCD.

(c) Vehicle Location & Speed

- The GPS module continuously updates:
    - Latitude
    - Longitude
    - Speed (in knots)
    - Satellite validity

This data is displayed on the LCD in real-time.

3. Accident Detection Logic

The system detects an accident when:

- The acceleration deviation crosses the set threshold AND
- The deviation is significantly different from normal vibration patterns.

If such a condition occurs:

1. The accident alert screen is shown on the LCD.
2. A buzzer is activated to notify the driver.
3. A timer (grace period) starts (e.g., 8 seconds).
4. If the driver presses the cancel button during this period, the alert is aborted.

This prevents accidental or false triggers.

4. Emergency Alert Activation

If the driver does not cancel during the grace period, the system performs two critical actions:

(a) Automatic Call

- The GSM module dials the emergency contact number.
- This gives immediate voice communication in case the victim is conscious.

(b) SMS with GPS Location

After the call is initiated, the system sends an SMS containing:

- A message indicating an accident
- Latitude & longitude values (up to 6 decimal precision)
- Speed before impact
- A direct Google Maps link

http://maps.google.com/maps?q=lat,lon

This allows the emergency responder to reach the exact spot quickly.

5. Confirmation and Reset

Once the call and SMS are sent:

- The LCD displays “CALL & SMS SENT”.
- The system enters a cool-down mode (e.g., 1 minute) to prevent duplicate alerts.
- After the cooldown, it returns to normal monitoring mode.

PIN CONNECTION

GSM TX → Arduino D2

GSM RX → Arduino D3

GPS TX → Arduino D10

GPS RX → Arduino D11

MPU6050 SDA → A3

MPU6050 SCL → A2

LCD SDA → A4

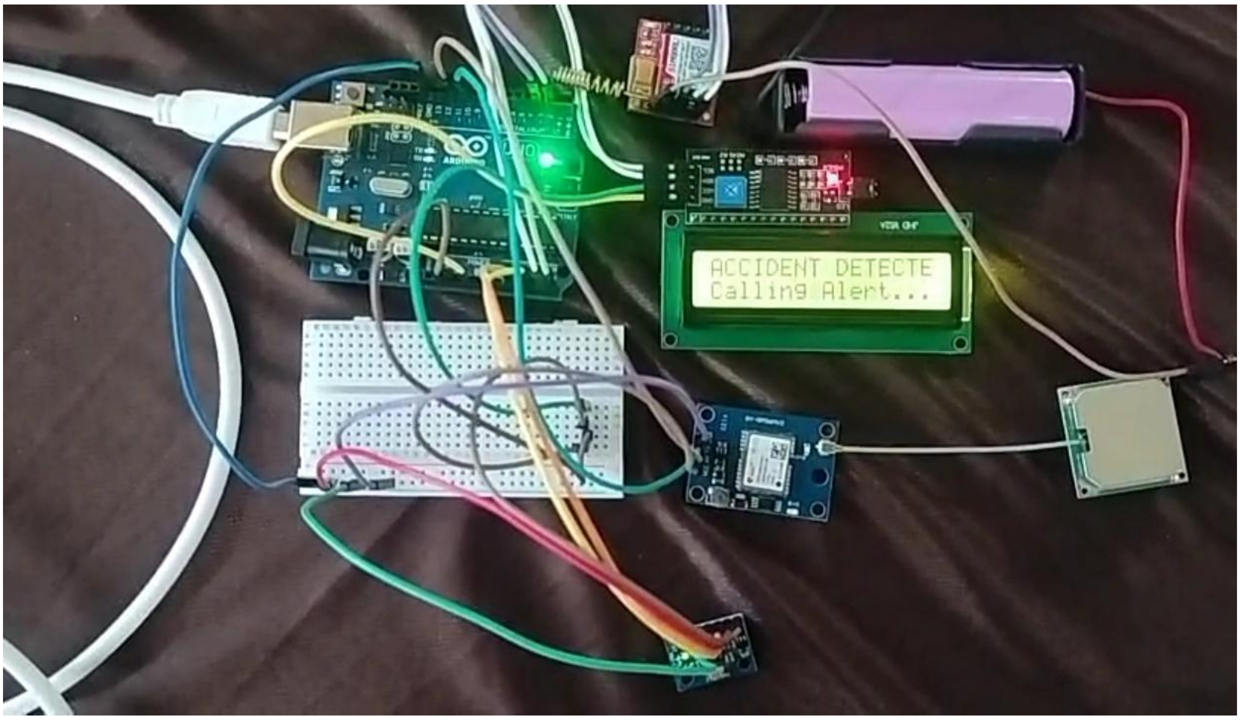
LCD SCL → A5

Ultrasonic TRIG → D7

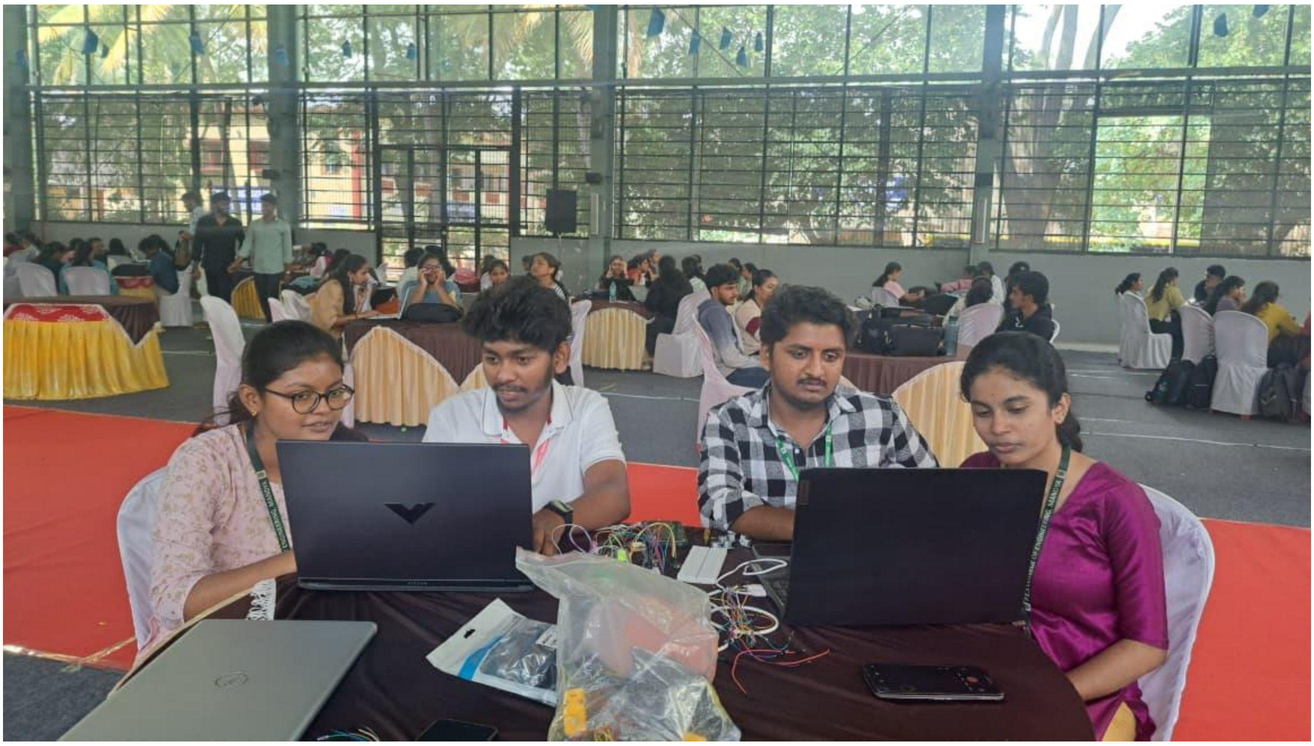
Ultrasonic ECHO → D8

Buzzer → D5

Cancel Button → D6 → GND



TEAM MEMBER



CHAPTER-4

Future Scope

- Integration with cloud IoT platforms for real-time monitoring
- Implementation of machine learning to improve detection accuracy
- Camera integration for capturing accident snapshots
- Vehicle-to-Infrastructure (V2X) communication
- Smartphone app for live tracking and alert management
- Automatic engine shut-off during major collisions

CONCLUTION

The Accident Detection and Emergency Alert System successfully demonstrate how embedded electronics and communication technologies can be combined to improve road safety and reduce emergency response time. By using sensors such as the MPU6050 accelerometer and the HC-SR04 ultrasonic sensor, the system accurately detects sudden impacts or potential collision conditions. With the integration of GPS and GSM modules, it automatically sends the precise accident location and initiates an emergency call, ensuring that help can reach the victim as quickly as possible.

The system operates reliably, is cost-effective, and can be implemented in two-wheelers, four-wheelers, and commercial vehicles. The inclusion of features such as a cancel button, buzzer alert, and threshold-based detection helps reduce false alarms and makes the system practical for real-world use. Overall, this project provides a smart, efficient, and life-saving solution that can significantly improve post-accident response and support the development of safer transportation systems.