IOT Based Vehicle Accident Detection and Emergency Alert System

[1]Prof. R. A. Kadu, [2]Siddharth Pardeshi, [3]Saideep Kulkarni, [4]Mahesh Kanawade

[1]Assistant Professor,[2]Student,[3]Student,[4]Student
[1,2,3,4]Department of Electronics & Telecommunication

[1,2,3,4]Pravara Rural Engineering College, Loni India.

ABSTRACT :

'IoT Based Vehicle Accident Detection and Emergency Alert System' is designed to enhance road safety and ensure timely assistance during accidents. This system utilizes an MPU6050 accelerometer to monitor the vehicle's motion and detect abrupt changes in acceleration indicative of a collision. Upon detecting an accident, the GPS module retrieves the vehicle's precise location, and the GSM module sends SMS alerts to multiple pre-configured emergency contacts, such as ambulance services, police, and family members.

By integrating IoT capabilities, the system provides real-time monitoring and communication, ensuring immediate response to emergencies. The scalable design allows for future enhancements, such as cloud-based data storage, integration with emergency service networks, and AI-based predictive analytics for proactive accident prevention. This affordable and efficient solution aims to minimize response time, reduce fatalities, and improve overall road safety.

Keywords:

IoT, Internet of Things, Vehicle Accident Detection, Emergency Alert System, GSM Module, GPS Module, Accelerometer, MPU6050, Real-Time Monitoring, Road Safety, Accident Notification, Emergency Response, Collision Detection, SMS Alerts, Location Tracking, Automotive Safety, Predictive Analytics

I. INTRODUCTION :

The "IoT Based Vehicle Accident Detection and Emergency Alert System" is a modern technological solution designed to address the critical issue of delayed emergency response during road accidents. In many cases, the lack of immediate notification to emergency services contributes significantly to the loss of lives. This project leverages IoT technologies to provide a real-time system that automatically detects accidents and promptly sends alerts to designated emergency contacts.

The system integrates key components, including an MPU6050 accelerometer for collision detection, a GPS module for precise location tracking, and a GSM module for sending SMS alerts. By monitoring the vehicle's motion and detecting sudden acceleration changes, the system identifies potential collisions. Upon detection, it retrieves the vehicle's coordinates and transmits them to emergency services, family members, and other predefined contacts via SMS.

This project aims to enhance road safety by significantly reducing the response time for accident victims. The solution is affordable, scalable, and easy to implement, making it suitable for integration into both personal and commercial vehicles. Future enhancements could include cloud-based monitoring, integration with IoT networks, and predictive analytics for proactive safety measures.

II. LITERATURE REVIEW :

[1] Singh et al. (2018) – Identified delays in emergency response due to manual accident reporting.
[2] Patel and Sharma (2019) – Proposed an MPU6050-based accident detection system but faced false alerts.
[3] Kumar et al. (2020) – Improved accuracy using a gyroscope sensor with the accelerometer.
[4] Gupta and Rao (2021) – Developed a GPS-GSM-based system for accident alerts via SMS.
[5] Choudhury et al. (2022) – Added Google Maps integration but faced network dependency issues.
[6] Desai et al. (2021) – Introduced a cloud-based IoT accident detection system for real-time monitoring.
[7] Mukherjee and Verma (2023) – Used AI for accident prediction, improving accuracy but requiring high computational power.

III. TECHNOLOGY STACK :

[1] Hardware Components & Their Functions

Microcontroller: Arduino Uno

- Purpose: The core processing unit that reads sensor data, detects an accident, and triggers SMS alerts.
- Specifications: ATmega328P, 14 digital I/O pins, 6 analog inputs, 16 MHz clock speed.

Accelerometer & Gyroscope: MPU6050

- Purpose: Detects sudden acceleration changes (impact detection) and tilting of the vehicle.
- Specifications: 3-axis accelerometer and 3-axis gyroscope, I2C interface.

GPS Module: NEO-6M

- Purpose: Determines the vehicle's real-time location coordinates.
- Specifications: Supports UART communication, accuracy within 2.5 meters.

GSM Module: SIM800L

- Purpose: Sends an emergency SMS alert to predefined numbers with accident location coordinates.
- Specifications: Quad-band GSM, operates on

3.7V-4.2V, communicates via UART.

Buzzer

- Purpose: Provides an audible alert when an accident is detected.
- Specifications: 5V operation, connected to a digital output pin.

LED Indicators

- Purpose: Indicates system power, accident detection, and successful SMS sending.
- Specifications: 5mm LEDs, connected to digital pins via resistors.

Power Supply: 12V Battery / Adapter

- Purpose: Provides power to Arduino and other components.
- Specifications: 12V DC, connected via a voltage regulator if necessary.

[2] Software & Programming Tools

Arduino IDE

- Purpose: Writing, compiling, and uploading the code to the Arduino Uno.
- Programming Language: Embedded C / C++.

Proteus

- Purpose: Testing the circuit design before implementation.

Fritzing

- Purpose: Creating circuit diagrams for documentation.

[3] Communication Protocols Used

- **I2C Protocol:** Used for communication between the MPU6050 sensor and Arduino.
- **UART Protocol:** Used for communication between Arduino, GPS module, and GSM module.

IV. PROPOSED SYSTEM :

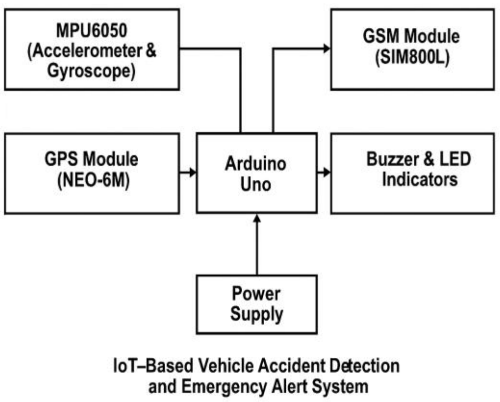
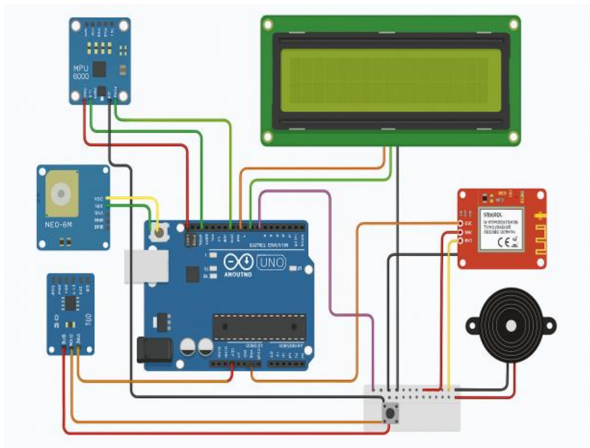


fig .Block Diagram

The proposed system is an IoT-based solution designed to automatically detect vehicle accidents and rapidly notify emergency contacts while providing an option for the user to cancel alerts if there are no injuries. Using an MPU6050 accelerometer, the system continuously monitors the vehicle's motion for sudden changes indicative of collisions. When an abnormal spike in acceleration is detected, the Arduino processes this information and triggers a series of responses: the system activates a buzzer for audible alert, displays an "Accident Detected" message on a 16x2 LCD, and simultaneously initiates the process to fetch the current location from a NEO-6M GPS module. Before sending out emergency notifications, a user-activated "No Casualty" button gives the driver an opportunity to cancel the alert within a set timeframe, thereby sending a cancellation message if no injuries occurred. If no input is received, the system utilizes a SIM800L GSM module to send an SMS with the accident details and GPS coordinates to multiple predefined emergency contacts. This integrated approach ensures that in the event of a collision, emergency response is both swift and accurate, while also reducing false alerts by incorporating user confirmation for non-critical incidents.

V. CIRCUIT DIAGRAM :



Arduino Uno (Central Controller):

- The Arduino Uno is placed at the center of the diagram since it is the brain of the system. All peripherals connect to it for power and data communication. The VCC and GND lines from the Arduino supply power to the connected components.

MPU6050 (Accelerometer & Gyroscope):

- The MPU6050 is connected via the I²C interface. In the diagram, its SDA and SCL lines are connected to the Arduino's A4 and A5 pins respectively. This sensor continuously monitors vehicle movements and provides acceleration (and tilt) data to the Arduino so that the system can determine if an accident has occurred.

NEO-6M GPS Module:

- The GPS module is connected to the Arduino using a separate pair of connections. Its TX (transmit) and RX (receive) pins are linked to digital pins (in our setup, they were assigned to Pins 4 and 3 on the Arduino, respectively, using a SoftwareSerial instance). The GPS module fetches real-time location data (latitude and longitude) when an accident is detected.

SIM800L GSM Module:

- The SIM800L module is used to send SMS alerts. In the Fritzing diagram, its TX and RX are connected to other digital pins of the Arduino (Pins 7 and 8 as per our setup, via SoftwareSerial for reliable communication). Additionally, power connections (5V or an external regulated voltage if required) and GND are shown. When an accident is confirmed, the GSM module sends out SMS messages containing the accident details and GPS coordinates to multiple emergency contacts.

16x2 LCD Display (I2C Interface):

- The LCD display is connected using the I2C protocol, which means it shares the SDA and SCL lines with the MPU6050. In the diagram, these connections also attach to A4 and A5. The LCD is used to show system messages such as "System Initializing", "Accident Detected", "No Casualty", and other status information, providing visual feedback.

No Casualty Button:

- The diagram shows a push button (labeled as the “No Casualty” button) connected to one of the Arduino’s digital pins (Pin 5). One side of the button is connected to the digital input pin with a pull-up resistor (internal or external), while the other side goes to GND. This button allows the driver to confirm that there are no casualties after an accident, thereby triggering a specific message.

Buzzer:

- A buzzer is also included in the diagram and is connected to a digital output pin (Pin 6). Its purpose is to provide an audible alert when an accident is detected.

Power and Ground Connections:

- Throughout the diagram, all components have proper power (5V, sometimes 3.3V for certain sensors) and GND connections to ensure they operate within their specified voltages. These connections are clearly shown with color-coded wires to maintain clarity in the layout.

VI. ALGORITHM :

[1] Initialization:

- Power on the Arduino and initialize all connected components.
- Initialize the MPU6050 sensor for acceleration and motion detection.
- Start communication with the NEO-6M GPS module to be ready for location retrieval.
- Initialize the SIM800L GSM module and set it to SMS text mode.
- Initialize the 16x2 LCD display to show system messages.
- Set the digital pin for the "No Casualty" button (with internal pull-up) and configure the buzzer pin.

[2] Continuous Monitoring:

- Enter the main loop where the Arduino continuously reads acceleration data from the MPU6050 sensor.
- Calculate the magnitude of acceleration from the sensor readings.

[3] Accident Detection:

- Compare the calculated acceleration magnitude with a predefined threshold.
- Decision:
  - If the acceleration does not exceed the threshold:
    - Continue monitoring sensor data.
  - If the acceleration exceeds the threshold:
    - Flag that an accident has been detected.
    - Display “Accident Detected” on the LCD and activate the buzzer to alert the driver.

[4] User Confirmation for No Casualty:

- Start a timer (e.g., for 30 seconds) to wait for user input.
- Decision: If the "No Casualty" button is pressed within the time period:
    - Display “No Casualty” on the LCD.
    - Cancel the emergency alert process.
    - Send an SMS to all predefined emergency contacts indicating that the accident was detected but there are no casualties.
  - If the button is not pressed within the allotted time:
    - Proceed to obtain the accident's location using the GPS module.

[5] GPS Location Retrieval:

- Retrieve the current GPS coordinates (latitude and longitude) from the NEO-6M module.
- Format the obtained coordinates into a human-readable string or URL (e.g., a Google Maps link).

[6] Emergency Alert Transmission:

- For each emergency contact in the predefined list, use the GSM module to send an SMS alert containing the accident message and the GPS location.
- Display a confirmation on the LCD that the emergency SMS has been sent.

[7] System Reset and Resume Monitoring:

- Once the SMS transmission is completed (either the emergency or no casualty alert), reset the system state.
- Return to the continuous monitoring step to await further events.

VII. CHALLENGES :

[1] Accurate Accident Detection:

- Need to distinguish between actual collisions and false positives (e.g., sudden braking, bumps, rough road surfaces).

[2] Calibration and Sensitivity:

- Extensive calibration required for MPU6050 to set the right sensitivity.
- Filtering algorithms must be fine-tuned to minimize false triggers.

[3] GPS Signal Reliability:

- Difficulty obtaining accurate GPS fixes in tunnels, urban canyons, or areas with poor satellite visibility.

[4] GSM Network Dependence:

- SIM800L GSM module performance depends on reliable cellular network coverage, which may be inconsistent in remote areas.

[5] Integration of Multiple Modules:

- Challenges in interfacing accelerometer, GPS, GSM, LCD, and user input reliably.
- Handling different voltage levels and potential electromagnetic interference between components.

[6] Power Management:

- Ensuring efficient power usage across all connected modules.
- Providing sufficient battery backup for uninterrupted operation.

[7] User Interface Reliability:

- Dependence on the user to press the "No Casualty" confirmation button in critical situations.
- Risk that, under stress, the interface might not function as intended.

VIII. ADVANCEMENTS :

[1] AI/ML Integration:

- Use machine learning algorithms to distinguish between real accidents and false triggers, improving detection accuracy.

[2] Cloud Connectivity:

- Enable real-time data logging and remote monitoring of accidents through cloud platforms, facilitating data analysis and fleet management.

[3] Advanced Communication Protocols:

- Incorporate 5G, NB-IoT, or LoRa for more reliable and low-latency transmission of emergency alerts in various environments.

[4] Enhanced V2X Communication:

- Integrate vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication to enable coordinated responses and traffic rerouting in emergency situations.

[5] Additional Safety Sensors:

- Add biometric sensors (e.g., heart rate, temperature) to monitor driver health and provide additional data during accidents.

[6] Improved User Interface:

- Enhance system feedback via more advanced displays or mobile apps that offer real-time status updates and accident information.

[7] Blockchain Integration:

- Use blockchain technology to secure and verify accident data, ensuring tamper-proof records for legal and insurance purposes.

IX. REAL WORLD EXAMPLES :

OnStar by General Motors is a well-known example of an automated accident detection system integrated into GM vehicles. It utilizes accelerometers and GPS to detect crashes and automatically notifies emergency services. When an accident occurs, the system transmits real-time location data and vehicle condition details to emergency responders, ensuring a faster and more efficient rescue operation.

Tesla vehicles also feature an advanced emergency response system that relies on AI-based accident detection. This system processes data from vehicle sensors, GPS, and cloud connectivity to determine the severity of an accident. In case of a major crash, Tesla's system can autonomously alert emergency services and provide crucial information such as the vehicle's location and impact force, improving the chances of a quick response.

Google has introduced a crash detection feature in Android smartphones, which functions similarly to in-car accident detection systems. Using built-in sensors such as accelerometers and gyroscopes, the smartphone can detect sudden impacts and immediately send alerts to emergency contacts. If the user is unresponsive after a detected crash, the device can automatically share their GPS location with first responders, enhancing personal safety even when a dedicated vehicle system is unavailable.

Mercedes-Benz has implemented the eCall system, a built-in emergency call feature that activates when airbags deploy during a collision. This system automatically contacts emergency services and transmits real-time crash data, including the vehicle’s GPS location and accident severity. By streamlining the emergency response process, the eCall system ensures that help arrives as quickly as possible, potentially saving lives.

In addition to in-car systems, IoT-based smart helmets for motorcyclists have been developed by various startups and researchers. These helmets integrate accelerometers, GSM modules, and GPS to detect crashes and send emergency alerts. If a motorcyclist falls or meets with an accident, the helmet system can automatically notify emergency services and designated contacts, improving safety for riders who are more vulnerable on the road.

X. FUTURE DIRECTIONS & RESERCH TRENDS :

The future of IoT-based vehicle accident detection and emergency alert systems is expected to evolve with advancements in artificial intelligence, cloud computing, and smart transportation networks. AI-driven accident prediction models are becoming more sophisticated, allowing systems to not only detect accidents but also anticipate potential collisions based on driving patterns, road conditions, and vehicle telemetry data. Machine learning algorithms will continue to refine impact detection, reducing false positives and improving real-time response accuracy.

Another key research trend is the integration of cloud-based IoT platforms, enabling real-time monitoring of vehicles by emergency services, fleet operators, and insurance companies. Cloud computing and big data analytics will allow historical crash data to be analyzed for traffic safety improvements and accident prevention strategies. Additionally, blockchain technology is emerging as a solution for secure and tamper-proof accident reporting, ensuring reliable data transmission between vehicles, authorities, and insurance providers.

The use of advanced communication technologies such as 5G, LoRaWAN, and NB-IoT is set to enhance the reliability of emergency alerts, especially in remote areas where traditional GSM networks may be weak. These technologies will enable low-latency, high-speed communication, allowing for faster and more efficient accident response. Future developments also include vehicle-to-everything (V2X) communication, where vehicles can exchange safety data with nearby cars, traffic signals, and emergency response units to provide a more interconnected accident management system.

Additionally, biometric monitoring and health assessment features are being explored to enhance post-accident response. Wearable sensors, embedded within vehicles or integrated into driver accessories, can monitor heart rate, body temperature, and other vital signs to determine the severity of an injury. This data can be transmitted to emergency responders along with accident location details, helping medical teams prepare appropriate treatment in advance.

XI. CONCLUSION :

The proposed IoT-based accident detection and emergency alert system effectively enhances road safety by providing real-time monitoring and instant communication during vehicular accidents. By integrating the MPU6050 accelerometer, NEO-6M GPS module, SIM800L GSM module, and LCD display with an Arduino microcontroller, the system ensures prompt detection of accidents and rapid transmission of location-based alerts to multiple emergency contacts. This timely response can significantly reduce fatalities and improve emergency services' efficiency. Overall, the project demonstrates a cost-effective, scalable, and reliable solution that contributes to safer transportation and smart mobility.

XII. REFERENCES :

[1] "Practical Arduino: Cool Projects for Open Source Hardware" by Jonathan Oxer and Hugh Blemings: A comprehensive guide for beginners to Arduino, which includes projects involving GPS and GSM modules. It’s useful for building the hardware part of your project

[2] "Internet of Things with Arduino Cookbook" by Marco Schwartz: This book includes IoT projects using Arduino, perfect for your accident detection project where you need to integrate sensors, GPS, and GSM.

[3] MPU6050 Datasheet: The official datasheet for the MPU6050 accelerometer and gyroscope, which helps understand how to use the sensor for detecting vehicle accidents.

[4] SIM800L GSM Module Datasheet: The datasheet for the SIM800L GSM module, which you can use to send SMS alerts.

[5] Neo-6M GPS Module Datasheet: The datasheet for the Neo-6M GPS module, which will help you extract location coordinates.

[6] "Vehicle Accident Detection and Notification System Using IoT" by P. S. S. Pradeep Kumar, K. R. Anusha, R. K. Siva, and M. N. Durga: This paper discusses the integration of IoT technologies like GPS and GSM to detect vehicle accidents and send notifications.

[7] "IoT-Based Vehicle Accident Detection and Emergency Alert System" by S. M. J. Rao et al.: An IoT system using sensors for vehicle accident detection, with detailed implementation of the accident detection and alert process.