Computer Technology Project 1

CMPE3002
SEMESTER 2 – 2025
Progress Report

Project Name:
IoT-Based & Cloud Integrated Accident Detection and Emergency Alert System

Student Name: Dhrubo Jouti Das Troyee

Student ID: 22663281

Course : Bachelor of Technology (Computer systems and Networking)

Date of Submission: October 17, 2025

Table of Contents

Abstract:

The report explained about the development of an Internet of Things-based accident detection system that uses real-time monitoring and alerting to enhance road safety. An Arduino MKR WiFi 1010 microcontroller connected to several sensors is used by the system to identify dangerous situations and auto accidents. Twilio SMS services and Firebase Cloud Functions are used to send accident notifications, which facilitates quick communication with specified emergency contacts. Real-time data visualization and history accident tracking are available for future investigation via a web-based dashboard. Research and system design, including cloud communication techniques, user engagement procedures, and IoT framework choices, are the main objectives of current project phase. Key issues like sensor accuracy, data reliability and secure cloud connectivity were addressed in the paper, which also contrasts developments with the initial design plan.

Introduction

Worldwide, car crashes are a primary source of deaths and serious injuries. Even with improvements in traffic planning and vehicle design, human error, poor vision, and postponed emergency response still generate many deaths.

Nearly 1.3 million people die and 50 million suffer injuries in traffic accidents annually, according to a World Health Organization (WHO) report [1] .Due to either unreported accidents or accidents that are discovered too late, many of these incidents happen in places where assistance is delayed. Modern reporting techniques primarily depend on human witnesses or phone calls, even though immediate treatment during this time the only way of survival.

Real-time data gathering, processing, and connection between sensors and cloud-based systems are made accessible by IoT. IoT sensors can be installed in cars to evaluate environmental factors, recognize odd motion patterns, and automatically sound an alarm in the event of a collision.

This project aims to design and create an Internet of Things (IoT)-based accident detection and emergency alert system that can use a variety of sensors to detect collision events while sending out real-time notifications via cloud platforms. Enhancing emergency response efficiency, decreasing dependence on human reporting, and eventually improving road safety results are the goals of the system.

Background and problem statement

IoT technology has changed a few sectors in recent years, including transportation, agriculture, and healthcare. IoT also provide features to detect the accident using the cloud connectivity and so on. However, existing accident alarm systems frequently depend on manual reporting or GSM-based communication, which may not always be available in remote locations or may be delayed or unreliable [1] , [2] . Moreover, fear of police case involvement, legal consequences and being held responsible for the children also delays the emergency response .

Researchers are gradually utilizing multi-sensor fusion (accelerometers, gyroscopes, vibration sensors, GPS) combined with cloud connectivity and automatic notifications to identify collisions in real time, highlighting the promise of new advances in IoT-based accident detection systems. Yet, there are still issues including false positives brought on by rough roads or abrupt brakes, network coverage restrictions, excessive costs, and data security issues[1].

As a result, an effective infrastructure is required that can securely send precise GPS location and sensor data, reliably detect accidents , differentiate between actual damage and suspicious activity, and promptly notify emergency contacts when needed also store the event record for future insurance purposes.

Project overview

The goal of the Internet of Things-based accident detection and emergency alert system is to immediately notify emergency contacts and automatically detects vehicle accidents. It collects and processes data from several sensors using an Arduino MKR WiFi 1010 microcontroller. Upon detecting a collision, the system informs users via Twilio SMS of the precise location of the incident and transmits data to Firebase Cloud. Real-time monitoring and historical data viewing are made possible using a web dashboard. Road safety is enhanced by this IoT and cloud technology integration, which minimizes emergency actions.

Objectives:

- ✓ Building a prototype integrating multi -sensor accident detection .
- ✓ Posting JSON over HTTPS to firebase Cloud functions using security device,
- ✓ Capturing GPS tracking , capturing coordinates, converting them into readable address.
- ✓ Securing the system with firebase Auth, fire store rules, restricted Api keys.
- ✓ Automating Alerts to notify predefined family member with neatest emergency services via Twilio over WiFi /cloud (without GSM.)
- ✓ Structural Cloud database such as fire store to log all the data for future insurance or research.
- ✓ Creating a web dashboard for visualising GPS location, live status , alert status, charting historical data.

Equipment and Tools:

The system are combination of sensors, hardware components and cloud enable software tolls to identify accident event , process data and deliver the emergency notification.

1. Sensors

- ✓ **DHT11 (Temperature sensor) :** senses excessive heat or overheating inside the vehicle , which works as a preventive safety warning [3].
- ✓ **Vibration Sensor (SW-420): identifies** shocks or mechanical vibrations in the event of a
- ✓ collision.
- ✓ **MQ2 (gas/smoke sensor):** detects smoke, methane or leaked gas after crash to identify fire[4], [5].
- ✓ **GPS Module(NEO 6M) :**provides precise real time latitude and longitude of accident location for tracing[6].
- ✓ **MQ3 (Alcohol Sensor): detects** alcohol in driver cabin to determine drunk driving condition[7]
- ✓ **MPU6050 (Built with 3 axis Accelerometer & 3 axis Gyroscope):**identify vehicle impact, tilt, and rotational movement by measuring sudden motion change.

2. Hardware Tool:

1. **Micro controller Arduino MKR wifi 1010**
   i. The Nina-W102 built-in Wi-Fi module enables direct cloud access without the need for external shields.
   ii. ARM Cortex-M0+ 32-bit microprocessor, which outperforms Uno or Nano in terms of performance.
   iii. Serieral digital/analog pins accommodate buttons, MPU6050, DHT11, GPS, OLED, and other sensors.

2. **OLED display (0.96 I²C)**
   i. Displays current system information, including GPS and accident alerts, temperature, and seatbelt status.
   ii. Low-power and small (0.96), perfect for embedded and automotive applications.

3. **Push Buttons (Seatbelt Detection & Cancel Alarm):**Detect seatbelt status and cancel false alert before data transmission
4. **Buzzer :**Produce an audible warning for accidents, overheating or countdown alerts.
5. **Breadboard:** used for connecting all the sensors and together
6. **Jumpier wires (M-M, F-M, M-F**): used for connecting Arduino with sensor, breadboard and electronical components.
7. **USB Cable** : used for programming and power supply

3. Software & programming tools

1. **Arduino IDE:** main environment to write, compile and upload code to MKR Wi-Fi 1010 microcontroller.
2. **Firebase fire store :**cloud NoSQL database to store all the accident event.
3. **Firebase Cloud functions :b**ackend logic to verify device data, process GPS address, trigger alert.
4. **Twilio API :**sent SMS alert service without GSM
5. **Google Maps and Places API :** converts GPS coordinates into a readable address also finds the neatest emergency service .
6. **CSS/HTM / JavaScript :** frontend technologies to create web dashboard for monitoring.
7. **Chart.js** : display historical graphs, bar of accident statistic on dashboard. Which creates clear visual feedback.
8. **Arduino Library :**
   - **WiFiNINA.h** :connects Arduino MKR Wi-Fi 1010 to Wi-Fi networks
   - **ArduinoHttpClient.h :**enables HTTPS POST/GET communication with cloud or backend.
   - **TinyGPS++ :**responsible to decode GPS data (longitudes, latitude) from NEO-6M to know the exact accident location
   - **ArduinoJson :** Responsible for converting data in Json format

System Processing flow

Here is a figure of how the system will process suing all the sensor and using other IOT devices.

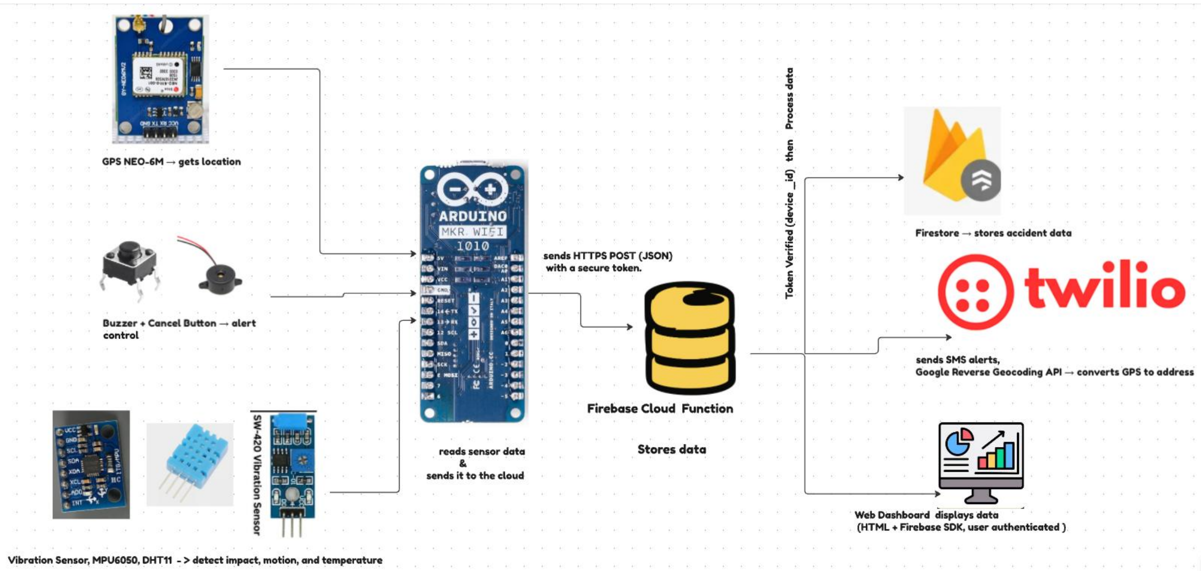

Figure1 : System processing Flow

Collecting sensor data
First, the Arduino MKR WiFi 1010 is used to continuously gather real-time sensor data from all of the modules that are attached. The MQ2, MQ3, and DHT11 sensors analyze environmental and safety hazards such the presence of alcohol, smoke, or unexpected temperatures, while motion-based sensors (MPU6050 and SW-420) identify unusual vibrations or crash impacts. To determine the position of the vehicle, the GPS module creates coordinates in real time.

Threshold checking and alert
Arduino compared against predefined thresholds. if any of these thresholds exceeds (impact, overheated) it will trigger an accidence detection. The buzzer will be warned the driver.

The system displays a 10-second confirmation interface on the OLED panel like "*Accident detected*" "*Alert cancelled*" in the moment when it detects a possible incident. The driver can press the cancel button during this period to avoid receiving false notifications from speed bumps and others.

Data communication (Cloud communication)
The device securely sends the JSON alert—which includes all sensor readings, GPS data, timestamp, and system status—to the Firebase Cloud Function via HTTPS with device authentication if the alert is not cancelled.

Connecting Firebase / Fire store/Web Dashboard
The cloud function confirms and maintains the info in Firebase Fire store, uses the Google Reverse Geocoding API to transform the GPS coordinates into a readable location, and uses the Google Places API to find the closest emergency services[8]. Lastly, Twilio updates the web-based dashboard in real time for monitoring and issue tracking, and it sends an SMS alert to emergency contacts.

Precautionary features:

In addition to recognizing crashes, this system includes intelligent driver safety notifications to prevent risky scenarios before an accident occurs:

- ✓ **Checking temperature:**
  - Warns Driver if above 35 °C by displaying OLED message *"Careful !! engine is heating !!"* to prevent engine fire risk.
  - If exceeds above 70 °C OLED message "*Overheat Detected !! sending alert !"* , will trigger accident alert.
- ✓ **Buzzer activation :**
  The buzzer will be activated if the temperature exceeds a predefined value (70 °C)
  - giving an audible warning.
  - Caution alert = short beep (24 °C)
  - Ctitical condition =Long beep (8 0 °C)
- ✓ **GPS zone Awareness:**
  Google places API helps detecting the school or hospital zone
  - OLED prompt the driver to slow down when driving in critical zone
  - OLED "Slow down!! School zone ahead!" Or "Drive carefully!! Hospital nearby !!"

Expected Outcomes:

- ✓ **Accident detection and Alerting:**
  Utilizing multi-sensor fusion (MPU6050 + SW-420) to detect collisions accurately and sending SMS alerts to pre-defined contacts with a GPS map link.

- ✓ **Vehicle and Driver Safety Monitoring**
  identifying high-risk situations (such as the presence of alcohol using MQ-3, smoke or gas using MQ-2, overheating using DHT11, or failure to wear a seatbelt) and adding them to logs and alerts for context.

- ✓ **Technical and Educational Impact:**
  improving knowledge of embedded systems, cloud data management, and IOT incorporates multi-sensor logic and IoT cloud platforms.

- ✓ **IoT Adaptability & Upcoming Cooperation** :
  Providing a cloud-based infrastructure built for future smart city integration and multiple cars.

- ✓ **Cloud Data logging and insurance support :**
  Utilizing Firebase Firestore to store structured incident information (sensor readings, timestamps, severity, and geocoded address) to facilitate research, inspections, and insurance claims.
  .

Challenges :

1. **Sensor Accuracy and Calibration**
   The selection of suitable sensors (e.g., DHT11 vs. DHT22, MQ-series, MPU6050, SW-420) requires comparative research to work in parallel without interference.

2. **Integration of cloud platforms**
   In order to protect location and identity data, integrating Firebase (Authentication, Cloud Functions, Fire store) required careful schema design, input validation, and basic Fire store rules.

3. **Connectivity and Network Dependency**: Signal loss or network instability may cause delays in warnings or data synchronization, particularly in rural or low-coverage locations, as the system relies on Wi-Fi and Firebase for real-time data transfer.

4. **Collection of data simultaneously**
   Timing and interference issues were introduced when accelerometer, vibration, temperature, gas, alcohol, seatbelt, and GPS modules were used in simultaneously. Maintaining real-time performance required effective I^2C/UART bus management,

Progress update and comparison:

Work done so far :

1. For understanding accident detection systems and IoT safety processes, a thorough literature review and problem examination were conducted.
2. Comparative analysis was used to choose appropriate sensors (such as the MPU6050, DHT11, MQ2, MQ3, GPS, and SW-420) while guaranteeing trustworthiness and compatibility.
3. Researched done before choosing Firebase for integration, a thorough examination of several cloud systems was conducted, including Firebase, Arduino Cloud, MQTT, and Blynk.
4. The system architecture and idea were done, outlining how the sensors, microcontroller, and cloud components would interact.
5. Investigated and combined google Maps/Places API and Twilio API were to allow precise position monitoring and automated alerting.
6. Decided of using HTTPS communication with firebase cloud function
7. Created a system flowchart to generate the idea of this project.

Schedule Progress :

1. Week 1-2 :drafted initial concept for my project, did literature review.
2. Week3-4 : compared sensors and selected MQ2, 3 DHT11 NEO-6M .
3. Week5-6 :Research cloud service option (firebase vs Arduino cloud) firebase is selected. And submitted the project video 1.
4. Week7-8 Completed system processing flow, planned for web dashboard(HTML/CSS)
5. Week9-10 :API research completed (Twillio , Maps, reverse geocoding) and submitted project video 2.
6. Week11-12 :started writing the report section and project presentation 3 slides.
7. Week13 :Compiled all the video 1,2,3 in a final formal report and submitted the project final video .

The project is on schedule and prepared to implementation and testing for CTP2.

Comparison with Project Form:

Topic	Updated plan	Previous plan
Communication Strategy	GSM Module (SIM800L)	Switch to HTTPS
Cloud service	Arduino IOT Cloud	Firebase, Firestore
Data storage	Not included	Cloud storage for history monitoring
Dashboard	Not included	Upgraded to real time web dashboard
Alert sending Way	IFTT	Twilio API
Precautionary Features	Not included	GPS zone- based , Buzzer activation , temperature
Future implementation	Not included	Planned for Lora Wan offline communication mode
GPS map link in a readable format	Not included	GPS reverse Geocoding API

Figure2 : comparison between original & project Form

Future Implementation

These are some of the list for future implementation:.

- ✓ Begin hardware integration using all the sensor mentioned above.
- ✓ Implement accident detection logic
- ✓ Integrate firebase Cloud function, fire store, Twilio Alert system
- ✓ Build and test web dashboard.
- ✓ Adding offline mode using Lora WAN communication
- ✓ Add fatigue detection such as (eye Blink Monitor , Time based alert , voice alert)
- ✓ Improve dashboard UI/UX

Reflection

It was completely a new experience while working on this project. The variety of IoT sensors, cloud platforms, and communication channels available initially made me feel overwhelmed. But after much investigation and comparison, I was able to determine which technologies were best for real-time alerting and hazard detection .My knowledge of sensor, network communication, and ,cloud integration with Firebase and Twilio was also improved by this project. I learned how theoretical research relates to real-world application and how structural planning can reduce problems like false alarms and connection delays. But its true at the very first of this project I was having less knowledge about IOT devices and their connectivity. Also it took me a long time to research about which device will be compatible with which and how they are going to work like (DHT11 vs DHT22). Additionally, I also had to study why HTTPS is being used rather than MQTT, as MQTT requires more complicated setup, whereas HTTPS is simpler to secure for small-scale projects. However, it was a great experience while doing this.

Conclusion:

To increase road safety, the IoT-Based Accident Detection and Emergency Alert System effectively integrate sensor technologies, cloud computing, and automated communication. The system is capable of precisely detecting environmental risks and accidents in real time by utilizing multiple sensors with the Arduino MKR WiFi 1010 [9], [10].Moreover, by integrating Firebase Cloud and Twilio, it guarantees that responders receive emergency warnings and location information promptly, cutting down on response time and perhaps saving lives. Overall, the project provides a scalable and affordable solution for intelligent transportation systems, illustrating how IoT may be used to address practical safety issues.

References:

[1] S. Aslam, S. Islam, N. Nigar, Sunday Adeola Ajagbe, and M. O. Adigun, “An IoT-Based Automatic Vehicle Accident Detection and Visual Situation Reporting System,” *Journal of Advanced Transportation*, vol. 2024, pp. 1–14, May 2024, doi: https://doi.org/10.1155/2024/4719669.

[2]S. Nanda, H. Joshi, and S. Khairnar, “An IOT Based Smart System for Accident Prevention and Detection,” *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, Aug. 2018, doi: https://doi.org/10.1109/iccubea.2018.8697663.

[3]Components101, “DHT11 Sensor Pinout, Features, Equivalents & Datasheet,” Components101, Jul. 16, 2021. https://components101.com/sensors/dht11-temperature-sensor.

[4]“Gas Leakage Detector With SMS Alert,” Jun. 14, 2023. https://justdoelectronics.com/gas-leakage-detector-with-sms-alert/.

[5]“Fire Alarm using Arduino, GSM and Flame sensor - Electronics Projects,” Apr. 22, 2022. https://electronicsprojects.in/fire-alarm-using-arduino-gsm-and-flame-sensor/

[6]“NEO-6M GPS Module — 25×25mm Antenna | Arduino & Raspberry Pi,” *REES52*, 2024. https://rees52.com/products/neo-6m-gps-module-25x25-active-antenna (accessed Oct. 17, 2025).

[7]“Alcohol Detector using Arduino, MQ3 Gas Sensor and 0.96” OLED Display Module - Electronics Projects,” *Electronics Projects*, Apr. 22, 2022. https://electronicsprojects.in/alcohol-detector-using-arduino-mq3-gas-sensor-and-0-96-oled-display-module/ (accessed Oct. 18, 2025).

[8] Muhriddin Eshqulov, Valiyev Muhammadali, and Jamshid Abdunazarov, “VEHICLE ACCIDENT ALERT SYSTEM BUILT USING ARDUINO, AN ADXL335 ACCELEROMETER, GPS, AND A GSM MODULE,” May 19, 2023. https://www.researchgate.net/publication/371685799_VEHICLE_ACCIDENT_ALERT_SYSTEM_BUILT_USING_ARDUINO_AN_ADXL335_ACCELEROMETER_GPS_AND_A_GSM_MODULE

[9]IRJET Journal, “IRJET- IOT BASED ACCIDENT TRACKING AND REPORTING,” *Academia.edu*, Jan. 07, 2020. https://www.academia.edu/41544513/IRJET_IOT_BASED_ACCIDENT_TRACKING_AND_REPORTING (accessed Oct. 18, 2025).

[10]A. Shaik *et al.*, “Smart Car: An IoT Based Accident Detection System,” *2018 IEEE Global Conference on Internet of Things (GCIoT)*, Dec. 2018, doi: https://doi.org/10.1109/gciot.2018.8620131.