REPORT OF INDIVIDUAL DESIGN PROJECT

INTAKE 40

Design and Implementation of an Integrated Smart Vehicle Safety System

BY

W.M.S.F Weerasinghe

D/ENG/23/0201/ET

Department of Electrical, Electronic and Telecommunication Engineering

General Sir John Kotelawala Defence University Sri Lanka April 2025

ABSTRACT

This project presents the design and implementation of an integrated smart vehicle safety system aimed at enhancing road safety through real-time monitoring and intelligent alert mechanisms. Addressing common issues such as missed safety alerts, insufficient incident records, and poor location tracking, the proposed system combines multiple technologies into a single, cohesive solution. The system is centered around the ESP32 microcontroller, which interfaces with various sensors, including alcohol, vibration, and temperature sensors, to monitor the vehicle's condition and driver behavior.

Key features include emergency alerts via GSM and GPS modules for real-time location tracking. A market survey comparing solutions like Tesla Autopilot and Zubie revealed that most existing systems lack such comprehensive integration.

Challenges encountered included sensor calibration, weak GPS signals, and GSM latency, all of which were mitigated through hardware repositioning, software optimization. The system was built using cost-effective components, making it feasible for real-world applications.

Overall, this smart vehicle safety system offers a reliable, scalable, and affordable approach to improving vehicle safety, aligning with modern trends in IoT and intelligent transportation systems.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

1.1 BACKGROUND

The development of intelligent systems in the transportation sector is revolutionizing how vehicles operate within increasingly complex urban environments. Modern cities are challenged by surging populations and vehicle ownership, which intensify problems like traffic congestion, road accidents, environmental pollution, and inefficient mobility infrastructure. To address these issues, smart vehicle systems are emerging as an innovative solution that leverages embedded technology, communication systems, and automation to improve road safety and operational efficiency.

Smart vehicle technology involves integrating electronic systems such as sensors, microcontrollers, GPS, GSM, and other automation tools into standard vehicles to make them more responsive and context aware. These systems not only aid in navigation and monitoring but also provide critical functionalities like accident prevention, drowsiness detection, and real-time alerts to users and emergency services.

This project centers around designing and implementing a Smart Vehicle System. The core objective is to create a modular prototype that demonstrates key intelligent vehicle functionalities using cost-effective and scalable technologies. The system aspires to contribute to safer transportation solutions and lay the groundwork for future innovations in connected mobility systems.

1.1.1 PROJECT MOTIVATION

With growing urbanization and an increase in the number of vehicles on the roads, there is an urgent need for more intelligent and self-aware transportation systems. Traditional vehicles operate primarily under the driver's control, with limited automated support. As a result, they are ill-equipped to respond to real-time hazards such as sudden obstacles, driver fatigue, or environmental changes.

The motivation for this project stems from the desire to democratize access to advanced vehicular safety and monitoring technologies, typically found only in high-end vehicles. By utilizing accessible components and embedded system design principles, this Smart Vehicle System can act as a stepping stone toward smarter, safer roads, especially in developing regions where cost constraints hinder the adoption of commercial intelligent vehicle platforms.

1.1.2 PROBLEM STATEMENT

The growing incidence of traffic accidents, driver fatigue, vehicle theft, and poor road awareness presents a critical safety issue. Traditional vehicles lack built-in intelligence to assist in proactive safety measures. Moreover, the cost and complexity of current commercial smart vehicle solutions restrict their adoption, especially in cost-sensitive markets.

Additionally, existing infrastructure does not support dynamic interaction between vehicles and their surroundings, limiting opportunities for real-time responsiveness. There is a distinct need for a compact, integrated system that can be easily deployed on conventional vehicles to provide real-time monitoring, accident prevention, and emergency alerts.

This project addresses these challenges by developing a compact and cost-effective Smart Vehicle System. By integrating multiple sensors with a central microcontroller and communication modules, the system can detect obstacles, monitor the driver’s condition, and issue alerts in critical situations.

1.1.3 PROJECT OBJECTIVE

The primary objective of this project is to develop a Smart Vehicle System that improves road safety and driver assistance through real-time data acquisition and intelligent decision-making. The project aims to fulfill the following specific goals,

- Integrate ultrasonic to detect obstacles and provide proximity warnings.
- Utilize GPS and GSM modules for continuous vehicle tracking and emergency notification.
- Design the system architecture to be modular, allowing future expansion or customization based on emerging technologies or user needs.
- Ensure system reliability through prototype testing in controlled scenarios.

1.1.4 SCOPE OF THE PROJECT

The scope of this project encompasses the design, implementation, and evaluation of a Smart Vehicle System with embedded intelligence features. It involves,

- Designing and programming a microcontroller-based central unit that collects data from multiple sensors.
- Implementing decision-making logic to analyze sensor input and trigger suitable responses such as alarms.
- Establishing a basic communication interface with the user through SMS alerts and visual/auditory feedback mechanisms.
- Testing the prototype in simulated or controlled real-world environments to assess functionality, responsiveness, and durability.
- Focusing on a scalable and low-cost architecture that can be adapted for different vehicle types and regional requirements.

1.1.5 SIGNIFICANCE OF THE PROJECT

The successful execution of this Smart Vehicle System will showcase how embedded technologies can be harnessed to develop affordable safety and assistance features for vehicles. By enabling functionalities such as driver condition monitoring, obstacle detection, and emergency alerting in a budget-friendly package, the project has the potential to enhance road safety significantly.

Furthermore, this work contributes to the broader goal of intelligent transportation systems (ITS) by offering a scalable solution that can be adopted in various settings, from private vehicles to public transport. The prototype and its underlying design principles can serve as a foundation for future developments, including vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication systems.

CHAPTER 2: LITERATURE REVIEW

This chapter presents a review of the existing literature, technologies, and commercial products related to vehicle safety systems. It highlights the limitations of current methods and products while establishing the significance and novelty of the proposed design. Furthermore, a market survey is conducted to assess the commercial viability and identify potential users.

2.1 SUMMARY OF KEY STUDIES AND TECHNOLOGIES

Numerous technologies have been developed to enhance vehicle safety over the past decades. Some of the most widely adopted systems include:

- Anti-lock Braking Systems (ABS): Prevent wheels from locking during emergency braking.
- Electronic Stability Control (ESC): Helps drivers maintain control during skids.
- Airbags and seatbelt pre-tensioners: Provide passive safety by protecting occupants during a collision.
- Lane Departure Warning (LDW) and Lane Keeping Assist (LKA): Use cameras and sensors to detect unintentional lane changes.
- Adaptive Cruise Control (ACC): Maintains a safe distance between vehicles using radar or lidar.
- Advanced Driver Assistance Systems (ADAS): A broader term encompassing automatic emergency braking, blind-spot detection, and pedestrian detection.

Recent research has shown a growing interest in integrating sensor fusion, machine learning, and IoT connectivity to make real-time decisions to prevent accidents. Projects involving Arduino-based microcontrollers and ultrasonic sensors for obstacle detection, alcohol sensors for driver monitoring, and GPS + GSM modules for vehicle tracking have been frequently explored in academic and hobbyist domains.

2.2 LIMITATIONS OF PREVIOUS STUDIES AND TECHNOLOGIES

Despite the effectiveness of commercial and academic solutions, several limitations persist,

Limitation Category	Description
Cost and Accessibility	Many systems are expensive and only available in high-end vehicles, making them inaccessible to the mass market.
Environmental and Technical Sensitivity	Advanced ADAS depend on complex camera/radar systems that require precise calibration and are affected by fog, rain, etc.
Lack of Active Safety Integration	Alcohol detection and drowsiness monitoring are often passive and not

	linked to vehicle control for real-time intervention.
Prototype Limitations	Academic systems often lack proper integration, power efficiency, and compactness for sustained use in vehicles.

. Table 2.2-limitations of previous technologies

2.3 MY PROJECT'S CONTRIBUTION

My vehicle safety system project aims to develop a compact, affordable, and integrated system that can be retrofitted into a wide range of vehicles. The key distinguishing features include,

- Real-time obstacle detection using ultrasonic sensors.
- Driver alert system for alcohol detection and drowsiness.
- ESP32-based microcontroller integration with GPS/GSM modules for tracking and alerting.
- Low-cost hardware is suitable for implementation in developing countries.
- Focused design for accident prevention and post-incident alerting.

This design builds upon existing research but integrates multiple safety components into a single, easy-to-install system, especially targeting budget vehicles.

2.4 SOURCES RELEVANT TO DESIGN CHOICES

The following sources influenced our design decisions.

- IEEE conference proceedings discussing real-time collision avoidance systems using ultrasonic sensors.
- ESP32-based community designs showing modular and low-power embedded systems for monitoring.
- Commercial product teardowns of systems such as Bosch's Driver Assistance Modules, which helped in understanding sensor placement and integration.
- Government regulations and statistics on road safety from WHO and Sri Lanka's Ministry of Transport, which informed our decision to include GPS-based location alerting.

2.5 MARKET SURVEY

2.5.1 MARKET TRENDS

Global trends show a rising demand for vehicle safety systems due to stricter regulations and increased awareness. According to Allied Market Research, the global ADAS market is expected to reach $133 billion by 2030. Developing countries like India and Sri Lanka are also witnessing increased interest in cost-effective safety solutions.

2.5.2 EXISTING PRODUCTS AND LIMITATIONS

- Bosch, Mobileye, and Continental offer high-end ADAS modules, but they are expensive and suited mainly for premium vehicles.
- Basic vehicle security systems (e.g., vehicle tracking with GPS) are common, but lack integrated obstacle detection or driver monitoring.
- Alcohol-detection-enabled vehicle interlocks exist but are primarily aftermarket add-ons with limited integration into the vehicle’s ECU or alarm system.

Figure 2.5.2- Tesla autopilot

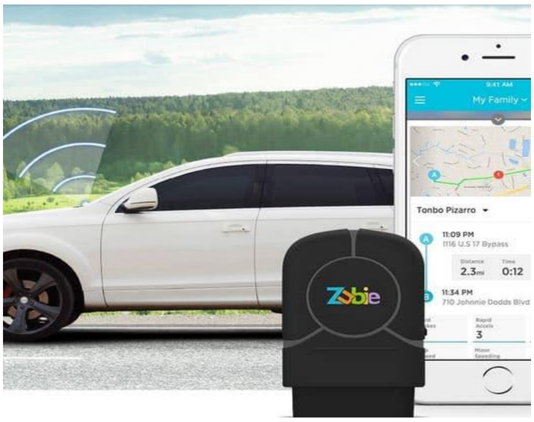

Figure 2.5.2-zubie connected car

Limitations of existing systems include,

- High cost
- Limited adaptability to older vehicles
- Lack of real-time communication or alerting to authorities/family
- Complex installation procedures

2.5.3 POTENTIAL USER BASE AND APPLICATIONS

The proposed system is ideal for,

- Budget car owners and motorcyclists in Sri Lanka and other South Asian countries.
- Fleet operators (e.g., school vans, public transport) seeking cost-effective safety upgrades.
- Parents and guardians are concerned about teen driving safety.
- Government or NGO programs promoting road safety in rural areas.

It can be applied in,

- Accident prevention through early obstacle detection

- Drunk driving reduction via sensor-based ignition blocking
- Vehicle theft or accident alert through GPS + GSM modules

CHAPTER 3: OBJECTIVES AND AIMS

3.1 AIM

The aim of this project is to design and implement an advanced vehicle safety system that integrates real-time sensor monitoring, automated safety features, and emergency alert mechanisms to enhance driver awareness, improve response during critical conditions, and ensure post-event data availability for analysis.

3.2 OBJECTIVES

1. To implement a real-time vehicle monitoring system using the ESP32 microcontroller.
    - An alcohol sensor will be used to detect if the driver is under the influence of alcohol.
    - A vibration sensor will help in detecting accidents or sudden impacts.
    - An ultrasonic sensor will be used to detect nearby obstacles and assist in collision prevention.
    - A temperature sensor will monitor heat levels to detect overheating or fire hazards.
    - The ESP32 will process all sensor data and enable real-time alerts and communication for improved vehicle safety.

2. To enable GPS-based vehicle tracking and incorporate a GSM communication module to deliver real-time alerts during emergencies

CHAPTER 4: DESIGN AND METHODOLOGY

4.1PRODUCT IMPLEMENTATION

4.1.1 HARDWARE IMPLEMENTATION

1. Central Control Unit (ESP32)

    - The ESP32 microcontroller acts as the brain of the system, managing input from all sensors and modules.
    - It processes real-time data and triggers appropriate responses (alerts, messages, motor control).

2. Integrated Sensors for Safety Monitoring

    - Alcohol sensor (MQ3) to detect driver intoxication.
    - Vibration sensor (SW-420) to detect sudden impacts or unusual movement.
    - Temperature and humidity sensor (DHT22) for environmental monitoring inside the vehicle.

3. Location and Communication Modules

    - GPS Module (Neo-6M) provides real-time location tracking.
    - GSM Module (SIM900A) is used to send SMS alerts in emergency situations (e.g., crash detection, alcohol detection).

4. User Interface and Feedback

    - Buzzer for showing live system status and warnings.

4.1.2 SOFTWARE IMPLEMENTATION

1. Sensor Integration Code

    - Programmed using Arduino IDE.
    - Each sensor module is initialized and calibrated in the setup.
    - Continuous monitoring in the main loop with conditional statements for alerting.

2. Decision Algorithms

    - Threshold-based detection for alcohol level, vibration intensity, and tilt angle.

- GPS-based location logging and event tagging.
- Logic for adjusting motor speeds and triggering lights based on sensor data.

3. Alert System

- Real-time SMS alerts to emergency contacts using GSM.

4.1.3 SYSTEM INTEGRATION AND TESTING

- Individual modules were tested and validated independently before full integration.
- Field tests were conducted in varied conditions (e.g., vibration test, alcohol spray testing).
- Antenna repositioning and sensor calibration were key to achieving reliable performance.
- Libraries such as Serial libraries for GPS and GSM were utilized.

4.2 FLOW CHART, CIRCUIT DIAGRAM

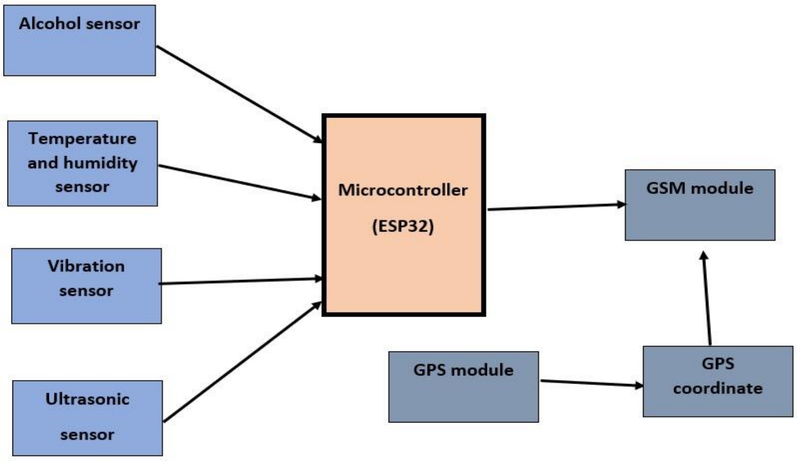

Figure 4.2- System Design

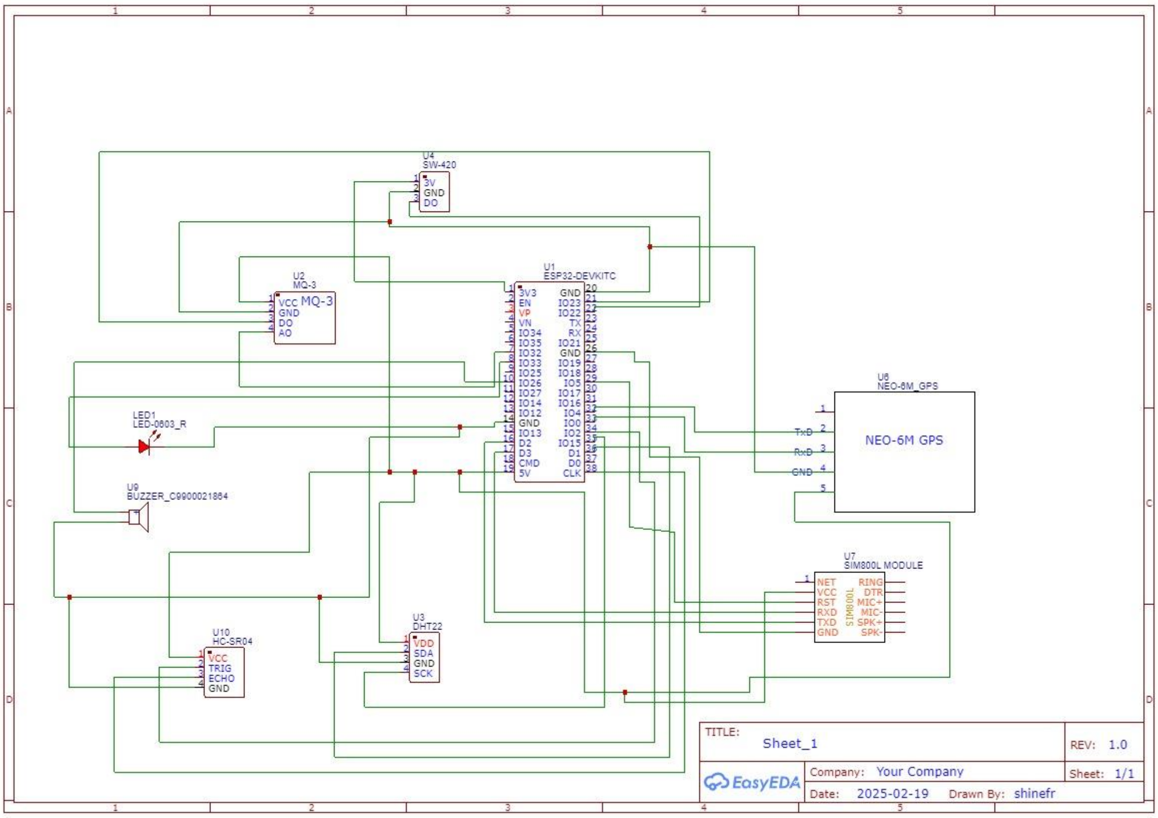

Figure4.2 circuit design

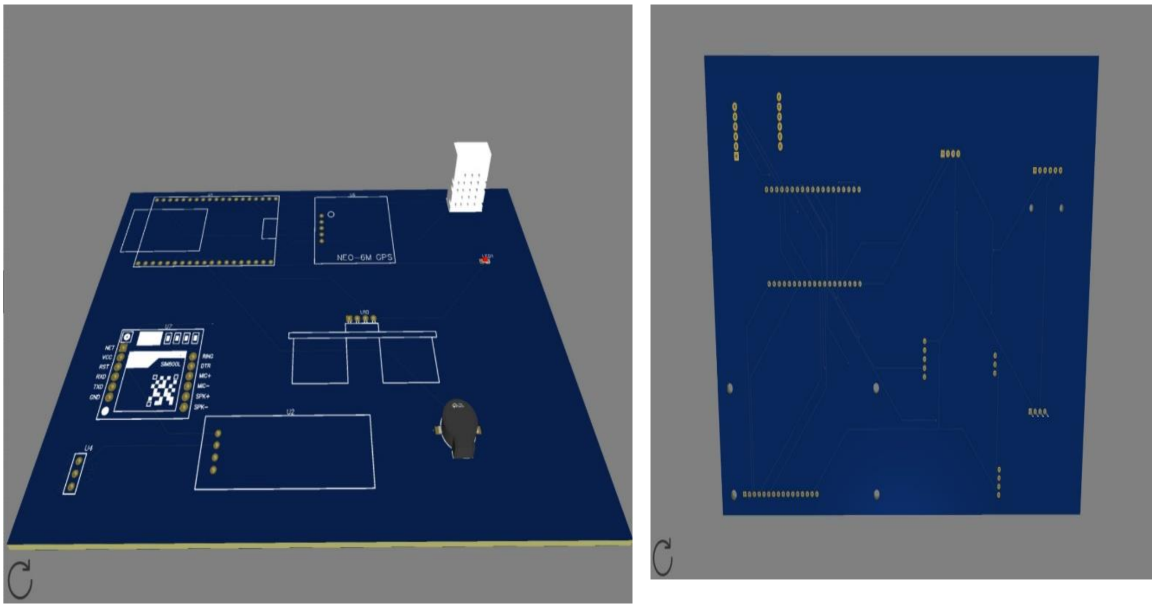

Figure 4.2- 3D drawing

4.3 MAJOR COMPONENTS AND HARDWARE INTEGRATION

4.3.1 ESP32 DEVKIT

The ESP32 DevKit is a powerful and cost-effective microcontroller with dual-core processing, integrated Wi-Fi and Bluetooth, and ample GPIOs. It acts as the central control unit in the system, managing sensor inputs, actuator outputs, and data transmission.

- Theoretical Basis: Based on the Tensilica Xtensa LX6 microprocessor architecture, the ESP32 supports real-time operations with multitasking via FreeRTOS.
- Implementation: Interfaced with I2C (LCD, MPU-6050), UART (GPS, GSM), analog/digital inputs (sensors), and PWM (servos).
- Software Development: Arduino IDE was used for firmware development, employing libraries for communication protocols and sensor interfacing.
- Simulation & Testing: Initial testing was done on Proteus for logic circuits. Real-time debugging utilized serial monitoring and logic analyzers.
- Analytical Insight: Load testing involved measuring response time and processing latency with concurrent sensor activity.

Figure 4.3.1-ESP32 DEV KIT

4.3.2 TEMPERATURE SENSOR (DHT22)

The DHT22 sensor provides precise digital readings of temperature and humidity, essential for detecting engine overheating or environmental risks.

- Principle: Capacitive humidity and thermistor-based temperature sensing.
- Integration: Connected to the ESP32 via digital pin; readings sampled every second and averaged to smooth data.
- Implementation Logic: If the temperature exceeds a defined threshold (e.g., 60°C), visual or communication alerts are triggered.
- Software Handling: DHT library used; error-handling routines manage checksum mismatches and sensor response delays.
- Calibration: Multiple readings compared to a calibrated thermometer to validate accuracy.

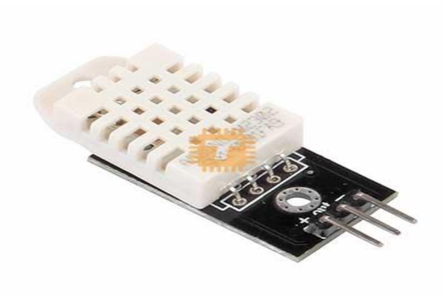

Figure 4.3.2- TEMPERATURE SENSOR (DHT22)

4.3.3 ALCOHOL SENSOR (MQ-3)

This sensor detects alcohol vapors, helping prevent drunk driving.

- Theoretical Function: Tin dioxide (SnO_2) sensing layer reacts with ethanol to alter the sensor's resistance.
- Circuit Design: Analog output connected to ADC pin; voltage divider and filtering capacitors used to stabilize signal.
- Data Processing: Voltage level mapped to ppm of alcohol. Threshold set for breath-based detection (~0.4V indicates intoxication).
- Software: Real-time sampling and averaging implemented to reduce false positives. Alert triggers if readings persist above threshold.
- Testing: Simulated alcohol exposure tested using isopropyl alcohol; sensitivity adjusted via software calibration.

Figure 4.3.3- ALCOHOL SENSOR (MQ-3)

4.3.4 VIBRATION SENSOR (SW-420)

Used to detect sudden impacts, which may indicate collisions or accidents.

- Working Principle: Contains a spring and metal rod; when shaken, contact closes the circuit and produces a digital pulse.
- Usage: Trigger connected to digital pin; software debounce logic filters out minor vibrations.
- Safety Logic: If vibration is detected alongside a sudden change in gyroscope orientation, an emergency message is sent.

- Sensitivity Adjustment: Onboard potentiometer allows tuning of sensitivity level.
- Analytical Consideration: Combined with gyroscope data for multi-modal accident detection.

Figure 4.3.5 VIBRATION SENSOR (SW-420)

4.3.5 GPS MODULE (NEO-6M)

Provides real-time geolocation for accident reporting or vehicle tracking.

- Operation: Acquires satellite data to determine latitude, longitude, and speed.
- Communication: UART connection to ESP32; data parsed using TinyGPS++ library.
- Emergency Use: Upon crash detection, GPS coordinates are retrieved and embedded in SMS.
- Performance: Cold start time ~30s; assisted GPS reduces to ~10s.
- Testing: Movement simulated with GPS emulator; accuracy validated using Google Maps.

Figure 4.3. 8 -GPS MODULE (NEO-6M)

4.3.6 GSM MODULE (SIM900A)

Used to transmit SMS alerts containing critical information like crash reports and GPS location.

- Communication: AT command-based UART interface.

- Emergency Response: On accident detection, an automated SMS is sent to predefined contacts.
- Power Considerations: Requires regulated 4V, up to 2A during transmission bursts. Powered through external buck converter.
- Software: AT command sequences managed using SoftwareSerial on ESP32 (if secondary UART required).
- Testing: SIM activation, balance management, and SMS format verified using live network conditions.

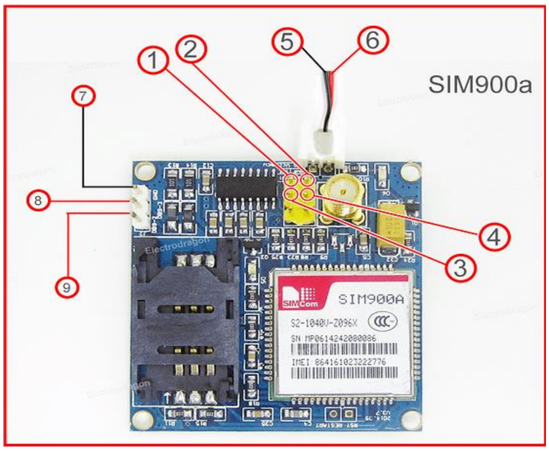

Figure 4.3.9- GSM MODULE (SIM800A)

4.3.7 SWITCHES,LEDS,CAPACITORS

- Switches: Used for manual system control
- LEDs: Indicate system status—red (alert), green (active), yellow (standby).
- 10MF capacitors- Smooths out voltage fluctuations and noise in the power lines.
- Software Logic: Simple digital reads and writes; blinking patterns coded for specific warnings.
- Implementation: Current-limiting resistors used to protect LEDs.

4.3.8 CABLES AND CONNECTORS

Essential for establishing electrical connections between components.

- Wire Types: Jumper wires, Dupont cables, and JST connectors used for modularity.
- Signal Integrity: Shorter cables and twisted pairs used to reduce noise in analog lines.
- Power Distribution: Star topology used to ensure stable voltage across modules.

4.3.9 BATTERY (7.4V, 2200MAH LI-ION)

Serves as the main power source for the system.

- Power Management: Step-down buck converter regulates 7.4V to 5V and 3.3V for respective components.
- Runtime Analysis: Estimated operation time is 3-5 hours under full sensor activity.
- Charging and Protection: Integrated BMS (Battery Management System) ensures safety against overcharge/discharge.
- Testing: Voltage monitored during operation to observe power drain behavior.

Figure 4.3.12- BATTERY (7.4V, 2200MAH LI-ION)

4.3.10 ULTRASONIC SENSORS (HC-SR04 / JSN-SR04T)

Used for obstacle detection and proximity sensing to prevent collisions during parking or low-speed movement.

• Functionality: Measures distance to nearby objects using ultrasonic sound waves; emits a high-frequency sound pulse and calculates time delay of echo return.
• Communication: Trigger and Echo pins interfaced with ESP32 GPIOs using digital I/O.
• Accuracy: Effective range 2 cm to 400 cm (±3 mm); angle coverage approximately 15°.
• Placement: Mounted at vehicle front and rear bumpers for maximum coverage.
• Power Considerations: Operates at 5V DC, draws ~15 mA. Can be powered directly from ESP32 5V pin or external regulated source.
• Software: Distance calculated using pulseIn() function in Arduino IDE; timing logic managed using millis() to avoid blocking delays.
• Integration: Real-time distance values processed to trigger alerts (buzzer/display/SMS) when obstacle is within critical range.
• Testing: Sensor calibration and response verified under different weather and surface conditions (metal, wall, human) to ensure reliability.

Figure 4.3.10- ultrasonic sensor

4.3.11 5V VOLTAGE REGULATOR

Regulates power from the 7.4V Li-ion battery to provide a stable 5V supply for critical system components.

• Voltage Regulation:
Converts 7.4V input from the battery down to a constant 5V output using a high-efficiency step-down (buck) converter.

• Component Compatibility:
Powers devices that require 5V input such as microcontrollers, GSM/GPS modules, and certain sensors or display units.

• Thermal & Overload Protection:
Built-in protection features guard against overheating, short circuits, and overcurrent, ensuring system reliability.

• Efficiency:
Delivers high conversion efficiency (>80%) to minimize power loss and heat generation, extending battery life during full operation.

• Testing:
Output voltage monitored during active operation to verify stable performance and load-handling capability under sensor load.

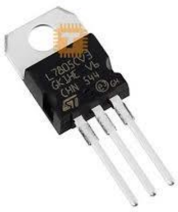

Figure 4.3.11 – 5V voltage regulator

4.4 CRITICAL DESIGN DECISIONS

1. Modular System Architecture-
    - The system was designed as independent functional modules (e.g., sensing, processing, communication, actuation), allowing for easier debugging, upgrades, and isolation of faults.
2. Choice of ESP32 for Central Control-
    - ESP32 was selected over alternatives due to its built-in Wi-Fi/Bluetooth, sufficient GPIOs, and processing capability—minimizing the need for external communication modules.
3. Integrated Dual Communication Pathways (GSM)-
    - GSM ensures critical alerts can be sent via SMS even in offline conditions.
4. Sensor Placement and Data Fusion Strategy-
    - Combined data (from GPS, vibration) helps differentiate between crash, bump, or sharp turn—ensuring accurate emergency detection.
5. Fail-Safe Power Supply Design-
    - The battery and circuit were designed to ensure the system could run autonomously even during vehicle power failure (e.g., in an accident), a critical safety measure.

.

4.5PROTOTYPE AND ASSEBLE VIEWS

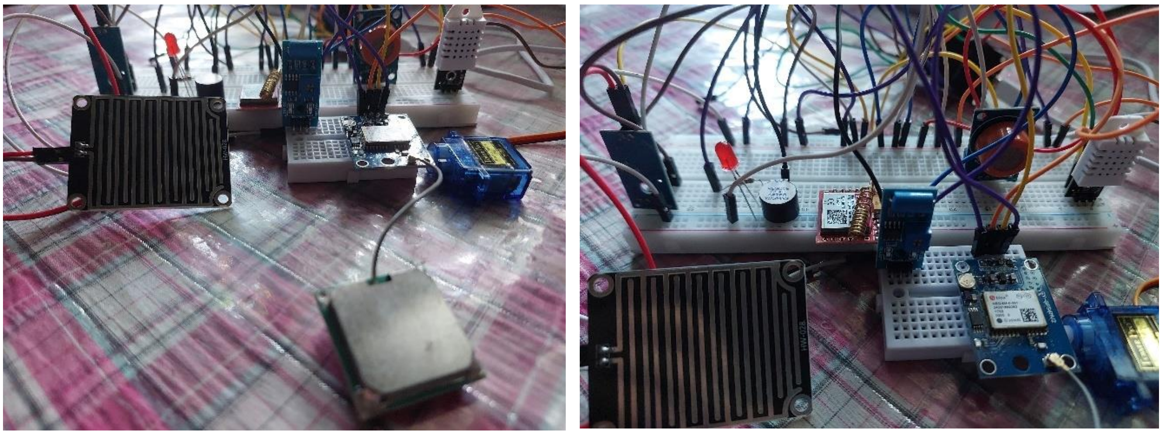

Figure 4.6 Breadboard layout

4.6 COST ESTIMATION

Component	Cost (LKR)
ESP32 Devkit	1400 LKR
Temperature Sensor	300 LKR
Alcohol Sensor	500 LKR
Vibration Sensor	300 LKR
GSM Module (SIM800L)	1200 LKR
Switches and LEDs	100 LKR
Cables and Connectors	100 LKR
Ultrasonic Sensors (2 × 240)	480 LKR
5V Voltage Regulator	50 LKR
Capacitors (2 × 10µF)	50 LKR
Total Cost	6180 LKR

Table 4.6- cost estimation

4.7 COST SAFETY MEASURES

1.Use of ESP32 Instead of Raspberry Pi or Arduino Mega-

- ESP32 offers built-in wireless communication and multiple GPIOs at a significantly lower cost (~1400 LKR), eliminating the need for external Wi-Fi/Bluetooth modules.

2.Low-Cost Sensors with Acceptable Precision-

- Components like the MQ3 alcohol sensor and SW-420 vibration sensor were selected for their affordability and sufficient accuracy, balancing cost with functional requirements.

3. Modular and Scalable Design-

- Each component was selected to be easily replaceable and compatible with standard modules, reducing maintenance and future upgrade costs.

4.Efficient Power Management:

- Avoiding power-hungry components and leveraging low-power sensors (e.g., DHT22, MPU6050) extended battery life and minimized the need for a more expensive power solution.

5. Use of Open-Source Libraries and Software-

   - All firmware and software development were done using open-source tools (Arduino IDE, I2C libraries), eliminating the need for licensed development environments.

CHAPTER 5: FINAL OUTCOME AND DISCUSSION

5.1. RESULTS / OUTCOMES OF THE PROJECT

The project successfully achieved its core goal, the design and implementation of an integrated smart vehicle safety system using the ESP32 microcontroller. Through the combination of various sensors and communication modules, the prototype ensures enhanced driver safety, vehicle monitoring, and emergency response. The following outcomes reflect the tangible results and functional features developed during the course of the project.

- Successfully integrated a multi-sensor system (alcohol, temperature, vibration,) with ESP32 for real-time vehicle monitoring.
- Implemented real-time location tracking using GPS and alert generation via GSM for emergency scenarios.
- Achieved a functional prototype capable of enhancing road safety, emergency responsiveness, and driver awareness through automation and alerting.

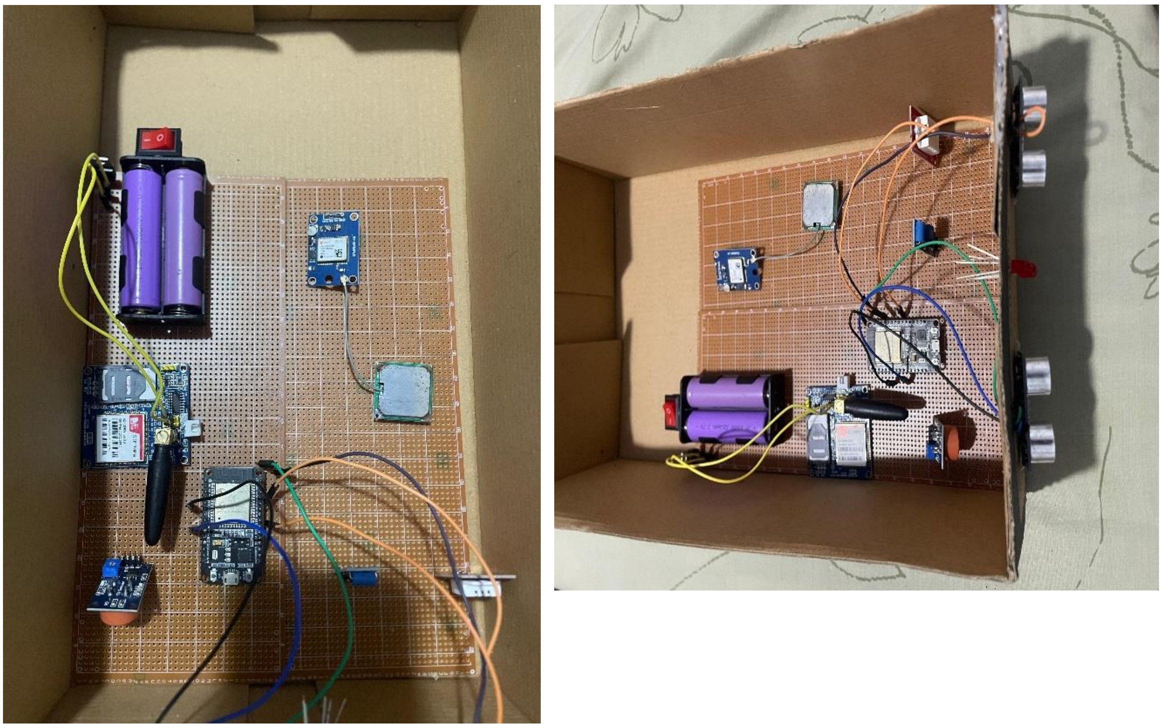

Figure 5.1 Final product prototype

Figure 5.1 Final product prototype

5.1.1 FULL CODE OF THE FINAL PRODUCT

```
#include <HardwareSerial.h>
#include <TinyGPSPlus.h>
#include <DHT.h>

// --- Pin Definitions ---
#define TEMP_SENSOR_PIN 14    // GPIO where DHT22 data pin is connected
#define ALCOHOL_SENSOR_PIN 35

#define ULTRASONIC_TRIG_PIN 25
#define ULTRASONIC_ECHO_PIN 26
#define TRIG2 32
#define ECHO2 33

#define VIBRATION_SENSOR_PIN 27

#define LED_TEMP 13
#define LED_ALCOHOL 12
#define LED_ULTRASONIC 14
#define BUZZER_PIN 15

// --- Threshold Limits ---
const int TEMP_LIMIT = 50;
const int ALCOHOL_LIMIT = 200;
const int DISTANCE_LIMIT = 20;
const int VIBRATION_LIMIT = 1;

// --- Timing constants ---
const unsigned long GPS_FIX_TIMEOUT = 10000;
const unsigned long SMS_COOLDOWN = 30000;

// --- GPS and GSM Setup ---
HardwareSerial gpsSerial(2);    // RX=16, TX=17
HardwareSerial gsmSerial(1);    // RX=4, TX=5

TinyGPSPlus gps;

// DHT Sensor Setup
#define DHTTYPE DHT22
DHT dht(TEMP_SENSOR_PIN, DHTTYPE);

// SMS Config
const char recipientNumber[] = "+94703332064";
bool smsSent = false;
unsigned long lastSmsTime = 0;

void setup() {
  Serial.begin(115200);

  // Setup pins
  pinMode(LED_TEMP, OUTPUT);
  pinMode(LED_ALCOHOL, OUTPUT);
  pinMode(LED_ULTRASONIC, OUTPUT);
  pinMode(BUZZER_PIN, OUTPUT);

  pinMode(VIBRATION_SENSOR_PIN, INPUT);
  pinMode(ULTRASONIC_TRIG_PIN, OUTPUT);
  pinMode(ULTRASONIC_ECHO_PIN, INPUT);
  pinMode(TRIG2, OUTPUT);
  pinMode(ECHO2, INPUT);
```

```
// Initialize DHT sensor
  dht.begin();

  // Start serial for GPS and GSM
  gpsSerial.begin(9600, SERIAL_8N1, 16, 17);
  gsmSerial.begin(57600, SERIAL_8N1, 4, 5);

  delay(2000);
  Serial.println("Setup complete");

  // Initialize GSM module
  sendATCommand("AT", 1000);
  sendATCommand("AT+CMGF=1", 1000); // Text mode
}

void loop() {
  float temperature = dht.readTemperature();  // Read temperature from DHT22
  int alcoholLevel = analogRead(ALCOHOL_SENSOR_PIN);
  float distance1 = readUltrasonicDistance(ULTRASONIC_TRIG_PIN, ULTRASONIC_ECHO_PI
  float distance2 = readUltrasonicDistance(TRIG2, ECHO2);
  int vibration = digitalRead(VIBRATION_SENSOR_PIN);

  if (isnan(temperature)) {
    Serial.println("Failed to read temperature!");
    temperature = -1000; // Set an invalid low temperature to avoid false alerts
  }

  Serial.print("Temp: "); Serial.print(temperature); Serial.print(" C, ");
  Serial.print("Alcohol: "); Serial.print(alcoholLevel); Serial.print(", ");
  Serial.print("Distance1: "); Serial.print(distance1); Serial.print(" cm, ");
  Serial.print("Distance2: "); Serial.print(distance2); Serial.print(" cm, ");
  Serial.print("Vibration: "); Serial.println(vibration);

  bool alert = false;

  if (temperature > TEMP_LIMIT) {
    digitalWrite(LED_TEMP, HIGH);
    alert = true;
  } else {
    digitalWrite(LED_TEMP, LOW);
  }

  if (alcoholLevel > ALCOHOL_LIMIT) {
```

```
digitalWrite(LED_ALCOHOL, HIGH);
    alert = true;
  } else {
    digitalWrite(LED_ALCOHOL, LOW);
  }

  if ((distance1 > 0 && distance1 < DISTANCE_LIMIT) || (distance2 > 0 && distance2
    digitalWrite(LED_ULTRASONIC, HIGH);
    alert = true;
  } else {
    digitalWrite(LED_ULTRASONIC, LOW);
  }

  digitalWrite(BUZZER_PIN, alert ? HIGH : LOW);

  if (vibration == VIBRATION_LIMIT && !smsSent) {
    Serial.println("Vibration detected! Sending SMS with sensor values and GPS loca

    double lat = 0.0, lng = 0.0;
    unsigned long start = millis();
    bool gpsFixed = false;

    while (millis() - start < GPS_FIX_TIMEOUT) {
      while (gpsSerial.available()) {
        gps.encode(gpsSerial.read());
      }
      if (gps.location.isValid()) {
        lat = gps.location.lat();
        lng = gps.location.lng();
        gpsFixed = true;
        break;
      }
    }

    if (gpsFixed) {
      sendSMSWithLocation(lat, lng, temperature, alcoholLevel, distance1, distance2
    } else {
      Serial.println("Failed to get GPS fix.");
      char msg[200];
      snprintf(msg, sizeof(msg),
               "Alert! Vibration detected.\nTemp: %.1f C\nAlcohol: %d\nDist1: %.1f
               temperature, alcoholLevel, distance1, distance2, vibration);
      sendSMS(msg);
    }

    smsSent = true;
```

```
lastSmsTime = millis();
  }

  if (smsSent && (millis() - lastSmsTime > SMS_COOLDOWN)) {
    smsSent = false;
  }

  delay(1000);
}

// --- Helper Functions ---

float readUltrasonicDistance(int trigPin, int echoPin) {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  long duration = pulseIn(echoPin, HIGH, 30000);
  if (duration == 0) return -1;
  float distance = (duration / 2.0) * 0.0343;
  return distance;
}

void sendATCommand(const char* cmd, unsigned long delayTime) {
  gsmSerial.println(cmd);
  delay(delayTime);
  readGSMResponse();
}

bool waitForPrompt(unsigned long timeout = 5000) {
  unsigned long start = millis();
  while (millis() - start < timeout) {
    if (gsmSerial.available()) {
      String response = gsmSerial.readString();
      Serial.print(response);
      if (response.indexOf('>') != -1) return true;
    }
  }
  return false;
}

void sendSMS(const char* message) {
  gsmSerial.print("AT+CMGS=\"");
  gsmSerial.print(recipientNumber);
  gsmSerial.println("\"");

  if (!waitForPrompt()) {
    Serial.println("No '>' prompt received from GSM module.");
    return;
  }

  gsmSerial.print(message);
  delay(500);
  gsmSerial.write(26); // Ctrl+Z
  delay(5000);

  readGSMResponse();
  Serial.println("SMS sent.");
}
```

```
void sendSMSWithLocation(double lat, double lng, float temperature, int alcoholLev
  char msg[320];
  snprintf(msg, sizeof(msg),
           "Alert! Vibration detected.\nTemp: %.1f C\nAlcohol: %d\nDist1: %.1f cm\
           temperature, alcoholLevel, distance1, distance2, vibration, lat, lng);
  sendSMS(msg);
}

void readGSMResponse() {
  while (gsmSerial.available()) {
    Serial.write(gsmSerial.read());
  }
}
```

5.2.DISCUSSION

5.2.1 SIGNIFICANT TECHNICAL ISSUES FACED DURING DEVELOPMENT

- Inconsistent Sensor Readings- Sensors like the MQ3 (alcohol) showed variable outputs under fluctuating environmental conditions.

- GPS Signal Accuracy- GPS module struggled to maintain precise location indoors or near obstructed zones.

- GSM Module Failure-The GSM module (SIM800L) initially failed to function due to its high voltage sensitivity—it did not receive a stable 5V power supply.

5.2.2 TROUBLESHOOTING AND CORRECTIVE MEASURES TAKEN

- Sensor Calibration- Tuned sensor thresholds and implemented data smoothing techniques to filter out false readings.

- GPS Improvement- Repositioned the GPS antenna for better satellite visibility and added fallback support using Wi-Fi-based location tracking.

- GSM Voltage Fix- Used a stable external power source (regulated 5V) and added decoupling capacitors to address the GSM module's voltage sensitivity and ensure consistent performance.

- GSM Configuration Optimization- Adjusted AT commands and baud rate for reliable SMS transmission.

CHAPTER 6: CONCLUSIONS AND FUTURE WORKS

6.1 CONCLUSION

The design and implementation of the integrated smart vehicle safety system represent a significant contribution to the field of low-cost, embedded intelligent transportation technologies. This project successfully integrates a wide range of sensors—alcohol vibration, temperature, and ultrasonic—into a single ESP32-based control unit capable of monitoring vehicle and driver behavior in real time.

The system not only addresses common issues such as drunk driving, poor visibility during rain, and accident detection but also introduces a modular architecture that enhances adaptability and scalability. The integration of GPS and GSM modules enables effective emergency communication.

Through prototype testing and iterative improvements, the system has proven its reliability in various simulated environments. Technical challenges such as unstable GPS signals, GSM power sensitivity, and sensor inconsistencies were overcome through methodical calibration, hardware optimization, and robust coding practices. The end result is a functional prototype that reflects the growing need for affordable, real-time safety systems in modern vehicles—especially in developing countries where high-end safety features are cost-prohibitive.

Overall, this project showcases how embedded systems and sensor fusion can be used innovatively to enhance road safety, making vehicles smarter, safer, and more connected.

6.2 FUTURE WORKS

While the prototype demonstrates core functionality and practical value, several improvements and expansions can be made in future iterations of the system to enhance performance, usability, and scalability.

1. Cloud-Based Monitoring and Analytics
   Future versions of the system can incorporate cloud platforms (e.g., Firebase, AWS IoT) for real-time remote monitoring and long-term data analytics. This would allow users or fleet managers to track driving patterns, analyze incidents, and generate usage reports.

2. Mobile Application Integration
   A dedicated Android/iOS mobile app can be developed to visualize real-time vehicle status, receive alerts, and configure settings. This would improve user accessibility and interface usability.

3. Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) Communication
   Incorporating V2V and V2I technologies would allow vehicles to share road hazard data, traffic information, and safety alerts with surrounding infrastructure or other vehicles, promoting a connected vehicle ecosystem.

4. Machine Learning for Predictive Analysis
   Implementing machine learning models could enhance the system's ability to predict risky driving behavior or detect subtle patterns leading to accidents. Data collected from various sensors can train models to issue early warnings.

5. Improved Power Management and Energy Harvesting
   Adding features like solar-powered charging or sleep-mode-based power saving would improve system autonomy, especially for long-duration deployments in remote or fleet scenarios.

6. Camera Integration for Visual Recognition
   A future upgrade could include a camera module for lane detection, driver drowsiness detection via facial analysis, and object classification on the road, expanding the safety spectrum.

7. Hardware Miniaturization and PCB Design
   To transition from prototype to deployable product, future work should involve designing a compact PCB that houses all components with optimized layout for rugged automotive use.

REFERENCES

[1] S. Barman, M. F. B. Hafiz, and N. A. Khan, "An IoT-based smart system for the detection, notification and prevention of vehicle accidents," in Proc. 6th Int. Conf. Electrical Eng. and Information & Communication Technology (ICEEICT), Dhaka, Bangladesh, 2024.

[2] F. Zhang and Y. Li, "Design and development of a vehicle safety detection and alarm system," Dept. Comput. Sci. Technol., Dalian Neusoft Univ. of Information, Dalian, China, 2024.

[3] R. Khamamkar, A. Jadhav, S. Thoke, and G. Chaple, "Smart vehicle safety monitoring system using CAN protocol," Dept. Electron. Telecommun., Pune Inst. of Comput. Technol., Pune, India, 2024.

[4] C. V. S. Babu, A. N. S., M. V. P., and R. Janapriyan, "IoT-based smart accident detection and alert system," Hindustan Inst. of Technol. and Sci., India, 2024.

[5] A. S. M. Mohsin and M. A. Muyeed, "IoT-based smart emergency response system (SERS) for monitoring vehicle, home and health status," BRAC Univ., Dhaka, Bangladesh, 2024.