BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY
DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

EEE 416 (January 2025)

Microprocessor and Embedded Systems Laboratory

Final Project Report

Section: C1 Group: 04

LifeSaver Helmet: An IoT-Based Smart Helmet for Rider Safety

Course Instructors:

Sadman Sakib Ahbab, Assistant Professor, EEE, BUET
Khairul Islam, Lecturer, EEE, BUET

Signature of Instructor: ____________________

Academic Honesty Statement:

"In signing this statement, We hereby certify that the work on this project is our own and that we have not copied the work of any other students (past or present), and cited all relevant sources while completing this project. We understand that if we fail to honor this agreement, We will each receive a score of ZERO for this project and be subject to failure of this course."

Signature: ____________________ **Full Name: Erom Hasan Niha** **Student ID: 2006144**	**Signature:** ____________________ **Full Name: Md. Amir Hamza** **Student ID: 2006145**
Signature: ____________________ **Full Name: Md. Jaied Hasan** **Student ID: 2006146**	**Signature:** ____________________ **Full Name: Ayon Kumar Dea** **Student ID: 2006147**

Table of Contents

1 Abstract 1
2 Introduction 1
3 Design 2
3.1 Problem Formulation (PO(b)) 2
3.1.1 Identification of Scope 2
3.1.2 Literature Review 3
3.1.3 Formulation of Problem 3
3.1.4 Analysis 4
3.2 Design Method (PO(a)) 4
3.3 Circuit Diagram 6
3.4 App Design 7
4 Implementation 9
4.1 Description 9
4.2 Algorithm 11
4.2.1 Helmet Module (ESP32) Algorithm 11
4.2.2 Bike Module (ESP8266) Algorithm 12
5 Design Analysis and Evaluation 13
5.1 Novelty 13
5.2 Design Considerations (PO(c)) 13
5.2.1 Considerations to public health and safety 13
5.2.2 Considerations to environment 14
5.2.3 Considerations to cultural and societal needs 14
5.3 Investigations (PO(d)) 14
5.3.1 Design of Experiment 15
5.3.2 Data Collection 15
5.3.3 Results and Analysis 15
5.3.4 Interpretation and Conclusions on Data 16
5.4 Limitations of Tools (PO(e)) 16
5.5 Impact Assessment (PO(f)) 16
5.5.1 Assessment of Societal and Cultural Issues 16
5.5.2 Assessment of Health and Safety Issues 16
5.5.3 Assessment of Legal Issues 17

5.6 Sustainability Evaluation (PO(g)) 17
5.7 Ethical Issues (PO(h)) 17
6 Reflection on Individual and Team work (PO(i)) 18
6.1 Individual Contribution of Each Member 18
6.2 Mode of Team Work 18
6.3 Diversity Statement of Team 18
6.4 Log Book of Project Implementation 19
7 Communication to External Stakeholders (PO(j)) 20
7.1 Executive Summary 20
7.2 User Manual 21
7.3 Github Link 24
8 Project Management and Cost Analysis (PO(k)) 25
8.1 Bill of Materials 25
8.2 Calculation of Per Unit Cost of Mass-Produced Unit 25
8.3 Timeline of Project Implementation 26
9 Future Work (PO(l)) 26
10 References 27
11 Appendix 28
11.1 Code for the Helmet Module 28
11.2 Code for Bike Module 32
11.3 Link to the Custom built app 33

1 Abstract

Motorcycle accidents remain a major contributor to road fatalities, often exacerbated by delayed emergency response and the absence of intelligent safety gear. The **LifeSaver Helmet** project presents a smart, IoT-enabled solution designed to enhance rider safety through real-time monitoring and automated emergency communication. This system integrates multiple sensors within the helmet to enable **fall detection**, **overspeed alerts**, **alcohol detection**, and **SOS emergency signaling**. It leverages **SIM800L GSM** and **GPS modules** to transmit live alerts and location data to a preconfigured emergency contact. A companion **Android application** allows users to easily configure emergency numbers, check system health, and receive critical status updates. In the event of a crash or unsafe condition, the helmet autonomously sends an SOS with GPS coordinates, facilitating timely intervention. By merging embedded hardware with a user-centric mobile interface, the LifeSaver Helmet offers a comprehensive approach to reducing accident response time and saving lives.

2 Introduction

Motorcycle accidents claim thousands of lives every year, leaving families devastated and communities in mourning. In many developing nations, where motorcycles serve as a primary mode of transportation, the absence of advanced safety measures exacerbates the risks. Traditional helmets, while providing basic protection, fail to address critical factors such as accident detection, intoxication prevention, and emergency response. The LifeSaver Helmet emerges as a groundbreaking solution, blending cutting-edge technology with practical safety features to redefine rider protection.

At its core, the LifeSaver Helmet is more than just protective gear—it is an intelligent guardian. Equipped with an array of sensors, it monitors the rider's environment in real time, detecting sudden falls, excessive speed, and even alcohol consumption. Unlike conventional helmets, it does not remain passive in emergencies. Instead, it springs into action, sending instant alerts with precise location data to emergency contacts, ensuring that help arrives without delay. This seamless integration of detection and communication sets it apart from ordinary safety gear.

One of the most compelling aspects of this project is its dual-alert system. Recognizing the unreliability of cellular networks in remote areas, the helmet pairs with a smartphone via Bluetooth, allowing the companion mobile app to send emergency messages if the GSM module fails. This redundancy ensures that no accident goes unreported, bridging the gap between technology and real-world usability. Moreover, the system enforces responsible riding by preventing motorcycle ignition if the helmet is not worn or if alcohol is detected—a feature that could save countless lives.

The development of the LifeSaver Helmet was not without challenges. Merging multiple sensors, wireless communication protocols, and power management into a compact, wearable device required meticulous engineering. From fine-tuning the MPU6050's fall detection algorithm to optimizing ESP-NOW for low-latency communication between the helmet and bike module, each component demanded rigorous testing. Yet, these hurdles only reinforced the project's necessity, proving that innovation thrives where safety is prioritized.

Ultimately, this project represents a fusion of compassion and technology. It is a response to the silent plea of every rider who has ever felt vulnerable on the road. By transforming a simple helmet into a life-saving device, we aim to reduce accidents, expedite emergency responses, and, most importantly, bring riders home safely. The LifeSaver Helmet is not just a product—it is a promise of security, a testament to what ingenuity can achieve when human life is at stake.

As we delve into the technical intricacies of this system, we invite readers to envision a future where technology and safety walk hand in hand. The road ahead is long, but with solutions like the LifeSaver Helmet, it is undoubtedly safer.

3 Design

3.1 Problem Formulation (PO(b))

3.1.1 Identification of Scope

The LifeSaver Helmet project combines multiple technologies into a single wearable device aimed at improving motorcycle safety. The system includes:

1. **Accident Detection:** The MPU6050 accelerometer and gyroscope are used to detect falls and sudden impacts. Upon detecting a crash, the helmet automatically triggers an SOS message, including the rider's GPS location, sent through the SIM800L GSM module.

2. **Speed Monitoring:** The helmet uses a GPS module to track the rider's speed. If the rider exceeds the set limit (80 km/h), a buzzer is activated to warn the rider.

3. **Alcohol Detection:** The helmet includes an MQ-3 alcohol sensor to detect alcohol levels in the rider's breath. If alcohol is detected, the system issues a warning, ensuring the rider is not impaired while operating the motorcycle.

4. **User-Friendly APP:** The Android application provides an easy interface to update emergency contacts, check the helmet's status (e.g., alcohol levels, speed, crash detection), and receive notifications in real-time. Users can directly interact with the helmet through Bluetooth for a smooth, personalized experience.

5. **Emergency Communication:** The SIM800L GSM module ensures reliable SMS-based communication in the event of an accident, sending a message with the rider's location to emergency contacts.

The scope of the project is to create a helmet that integrates all these features in a lightweight, comfortable, and user-friendly manner, ensuring safety without compromising usability.

3.1.2 Literature Review

1. **IoT in Motorcycle Safety:** The integration of IoT devices in motorcycle helmets is gaining traction, with several systems incorporating accelerometers and gyroscopes for fall detection and GPS for location tracking. These solutions automatically alert emergency contacts when an accident occurs, improving response times and rider safety.
2. **Alcohol Detection:** MQ-3 alcohol sensors are already in use in automotive safety systems to detect alcohol levels and prevent impaired driving. However, the concept of integrating this into a helmet is novel. This integration in the LifeSaver Helmet enhances the rider's safety by preventing impaired riding.
3. **Mobile App Integration for Safety Monitoring:** The use of Bluetooth for real-time communication between the helmet and a mobile app has been implemented in various smart wearables. These apps allow users to monitor device status, change settings, and receive real-time alerts, providing a seamless experience for users.

By combining these technologies, the LifeSaver Helmet offers an all-in-one solution, integrating accident detection, speed monitoring, alcohol detection, and emergency communication with a user-friendly app interface.

3.1.3 Formulation of Problem

The current state of motorcycle safety systems lacks the integration of real-time monitoring, automated emergency responses, and rider behavior analysis. The following issues are addressed in this project:

1. **Inadequate Accident Detection:** Traditional helmets cannot detect accidents or falls. The LifeSaver Helmet utilizes real-time fall detection with an MPU6050 sensor and automatically sends SMS alerts with location details through the SIM800L GSM module.
2. **Unsafe Riding Behavior:** Many riders exceed safe speed limits, leading to accidents. The LifeSaver Helmet addresses this by monitoring the rider’s speed via GPS and providing an audible buzzer alert when overspeeding is detected.
3. **Impaired Riding Due to Alcohol:** Impaired riders are a significant risk on the roads. The MQ-3 alcohol sensor ensures that riders do not operate the motorcycle under the influence, adding an additional layer of safety.
4. **Emergency Communication Delays:** In case of an accident, swift emergency communication is crucial. The helmet’s SMS alert system, combined with real-time

GPS tracking, enables faster emergency response by sending automated alerts to predefined contacts.

3.1.4 Analysis

The LifeSaver Helmet integrates several advanced technologies to address motorcycle safety concerns. The key analysis areas include:

1. **Sensor Integration:** The integration of multiple sensors (accelerometer, gyroscope, alcohol sensor, and GPS) into the helmet requires careful calibration to ensure accurate readings. The system must reliably detect falls, monitor speed, and assess alcohol levels while ensuring that false alarms are minimized.
2. **Real-Time Communication:** The SIM800L GSM module provides SMS-based communication, ensuring that emergency alerts are sent even in areas with limited internet access. The system must be designed to handle communication failures and ensure reliable data transmission.
3. **User Interface (APK):** The Android APK is crucial for user interaction. It allows riders to easily configure emergency contacts, check the status of the helmet, and receive real-time updates. The app must be intuitive, ensuring that even non-tech-savvy users can quickly adapt and use the system.
4. **Power Consumption:** Given the multiple sensors and communication modules, power management is a critical factor. The system must be designed to maximize battery life without compromising performance, especially considering that the helmet needs to function for extended periods.
5. **Durability and Comfort:** The helmet must meet safety standards and remain comfortable for long rides. The system must be integrated in a way that does not interfere with the helmet's primary role—protecting the rider's head during a crash.

3.2 Design Method (PO(a))

The development of the proposed *Smart Helmet-Based Motorcycle Safety System* required the integration of fundamental principles from mathematics, physics, electronics, and embedded systems engineering. The design approach systematically applied theoretical knowledge to practical implementation through sensor interfacing, data processing, wireless communication, and real-time control. The key design considerations and mathematical formulations are elaborated below.

To detect falls or abnormal rider motion, the MPU6050 sensor (3-axis accelerometer and gyroscope) was utilized to calculate the rms of acceleration vector and the angular velocity

Acceleration vector, $|a| = \sqrt{a_x^2 + a_y^2 + a_y^2}$ $derivative = \frac{d|a|}{dt}$

Angular velocity, $| w | = \sqrt{w_x^2 + w_y^2 + w_z^2}$

We set thresholds for the derivative of the acceleration vector and the angular velocity vertor. We found out the thesholds by experimenting.

This threshold-based detection was implemented using conditional logic in the ESP32 where sudden changes or sharp angular shift trigger a fall event .

The NEO-6M GPS module provides latitude and longitude readings. The system triggers an overspeed alert when:

$$V > V_{threshold} \quad \text{for} \quad \text{duration} > t_{limit}$$

This speed value is sent to the bike MCU to control an overspeed warning LED.

The MQ3 gas sensor detects alcohol concentration by measuring the change in resistance of a sensing element. This is based on chemical sensing principles.

The IR sensor consists of an emitter-receiver pair. The logic is governed by the behavior of IR reflection. The result is sent to the bike MCU via Bluetooth, and used to control the engine relay.

The SIM800L module is programmed using AT commands to send SMS alerts. It sends location-based messages when emergency situation occurs.

A relay module is used to control the bike ignition circuit. The relay is activated only when the helmet is worn, and no critical violations are detected.

The project demonstrates the integration of multiple scientific and engineering disciplines. Mathematical computations (vector norms, GPS-based speed, logic conditions), physical principles (motion, reflection, gas sensing), and embedded system design (UART communication, real-time processing) were successfully applied to implement a real-time, safety-critical embedded system. This design approach ensured accurate sensing, reliable decision-making, and timely emergency response.

3.3 Circuit Diagram

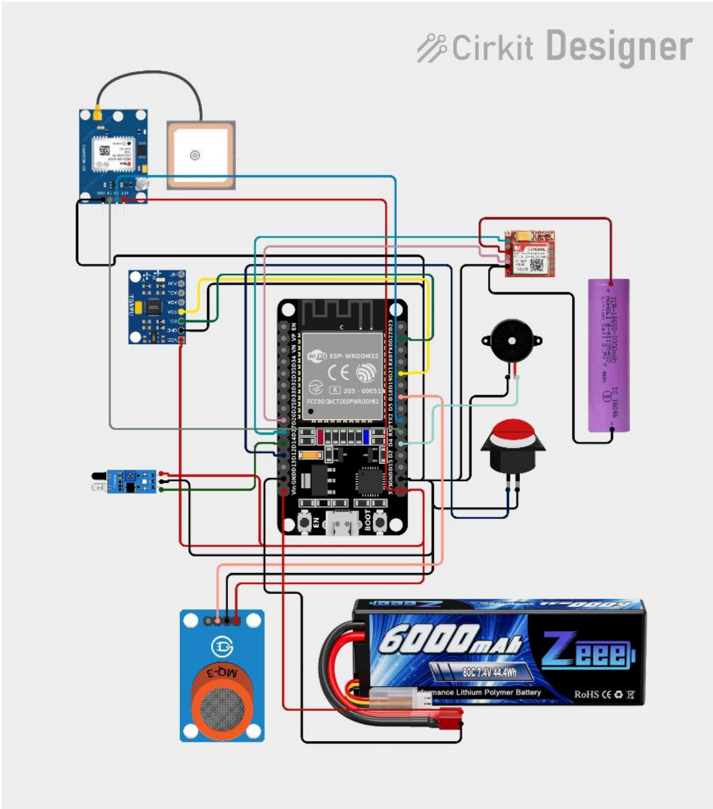


Fig: Circuit Diagram for Helmet Module

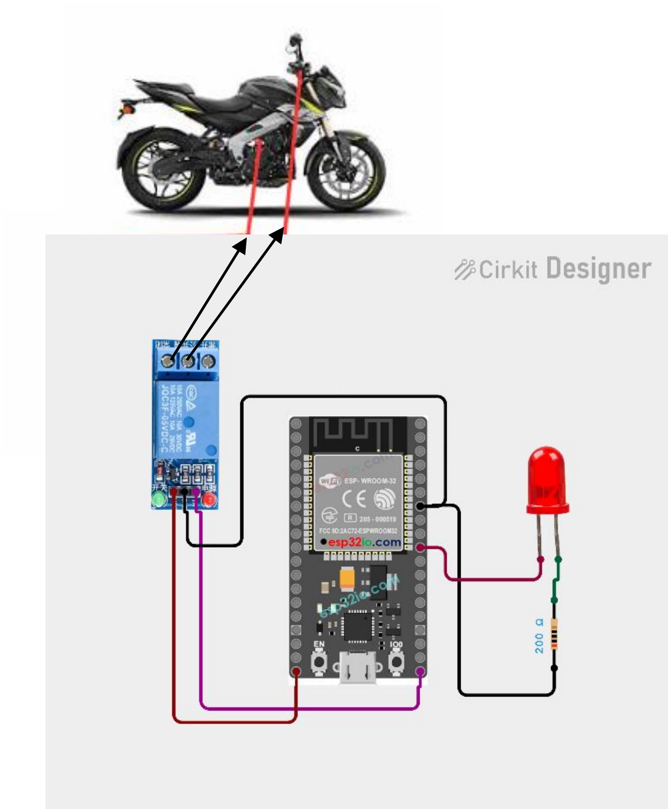
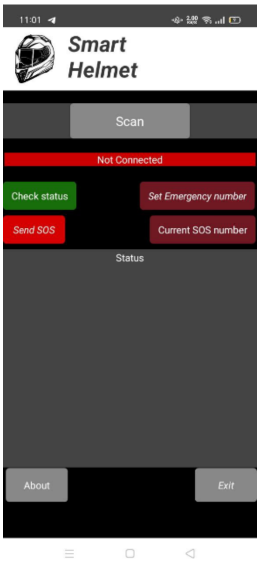


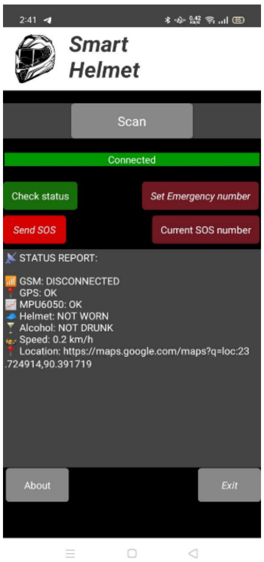
Fig: Circuit Diagram for Bike Unit

3.4 App Design

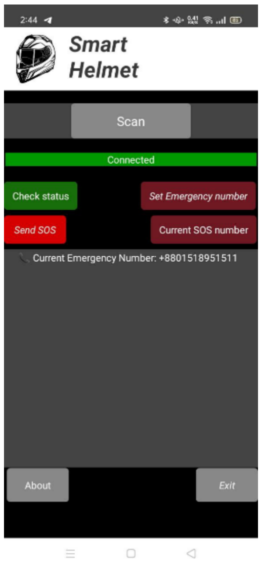
A dedicated Android application has been developed to work seamlessly with the LifeSaver Helmet. This companion app allows users to easily update the emergency contact number and monitor the status of all integrated components in real time. In addition to manual SOS activation from the helmet, the app includes an SOS feature that enables users to trigger emergency alerts directly from their phone. If the GSM module fails to send an SMS due to signal issues, the app automatically sends the alert message from the mobile device itself, ensuring reliable emergency communication. Furthermore, the app serves as a real-time debugging and monitoring tool—displaying all status updates and sensor outputs, much like a serial monitor. A few screenshots of the app interface are provided below for reference.



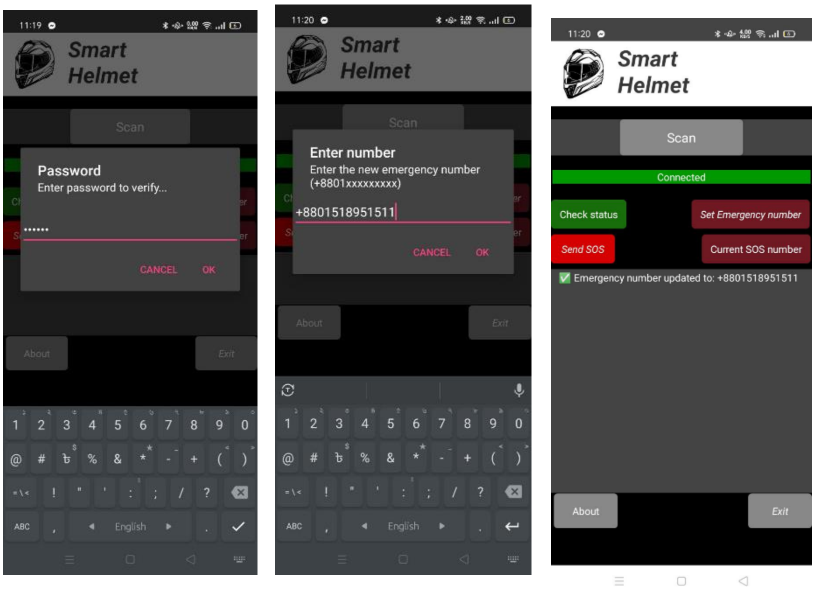
Homepage



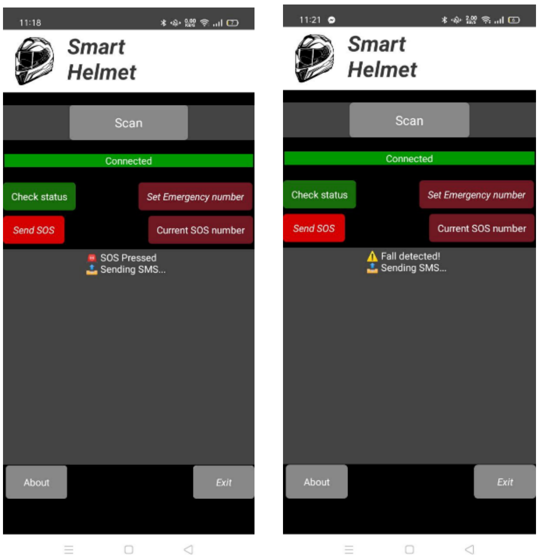
Status



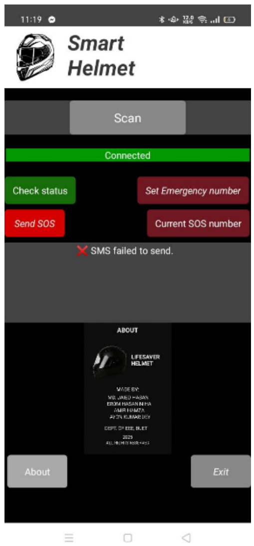
Current number



Updating Emergency Number



Real time updates



About section

4 Implementation

4.1 Description

The implementation of the LifeSaver Helmet system involves the integration of multiple hardware modules and software components to ensure real-time safety monitoring and emergency communication for motorcyclists. The system consists of two embedded microcontroller units: an ESP32-based helmet module and an ESP8266-based bike module, which communicate wirelessly using the ESP-NOW protocol.

The helmet module, powered by the ESP32, acts as the brain of the system. It connects to the following components:

- **MPU6050** for fall detection using accelerometer and gyroscope data.
- **SIM800L GSM module** for sending emergency SMS with GPS location.
- **GPS module** for real-time location tracking and speed monitoring.
- **MQ3 alcohol sensor** to detect alcohol consumption.
- **IR sensor** to detect whether the helmet is worn.
- **SOS button** for manual emergency alerts.
- **Buzzer** to warn during overspeeding scenarios.
- **Bluetooth module** (via built-in ESP32 Bluetooth) for interfacing with a custom Android mobile app.

The bike module, based on the NodeMCU (ESP8266), receives helmet status data via ESP-NOW and responds accordingly:

- It controls a relay module to enable or disable engine ignition based on helmet use and alcohol detection.
- It also uses an LED to visually indicate overspeeding.

Data transmission between the two microcontrollers is handled over ESP-NOW, ensuring low-latency, peer-to-peer communication without requiring Wi-Fi or internet. Sensor data such as helmet status, alcohol presence, and speed is continuously transmitted from the helmet to the bike module.

A **companion Android application**, developed using MIT App Inventor, allows the user to:

- Update and store the emergency contact number via Bluetooth.
- Request real-time system status (helmet, GSM, GPS, MPU6050, alcohol, and speed).
- Trigger SOS alerts manually.
- Receive notifications for fall detection, overspeeding, and failed GSM messages. If SMS delivery fails through the GSM module, the app automatically sends the SOS message from the user’s mobile device using native SMS functions, ensuring redundancy and reliability.

Together, these components form a robust, real-time safety system designed to mitigate risks and enhance emergency response for two-wheeler riders.

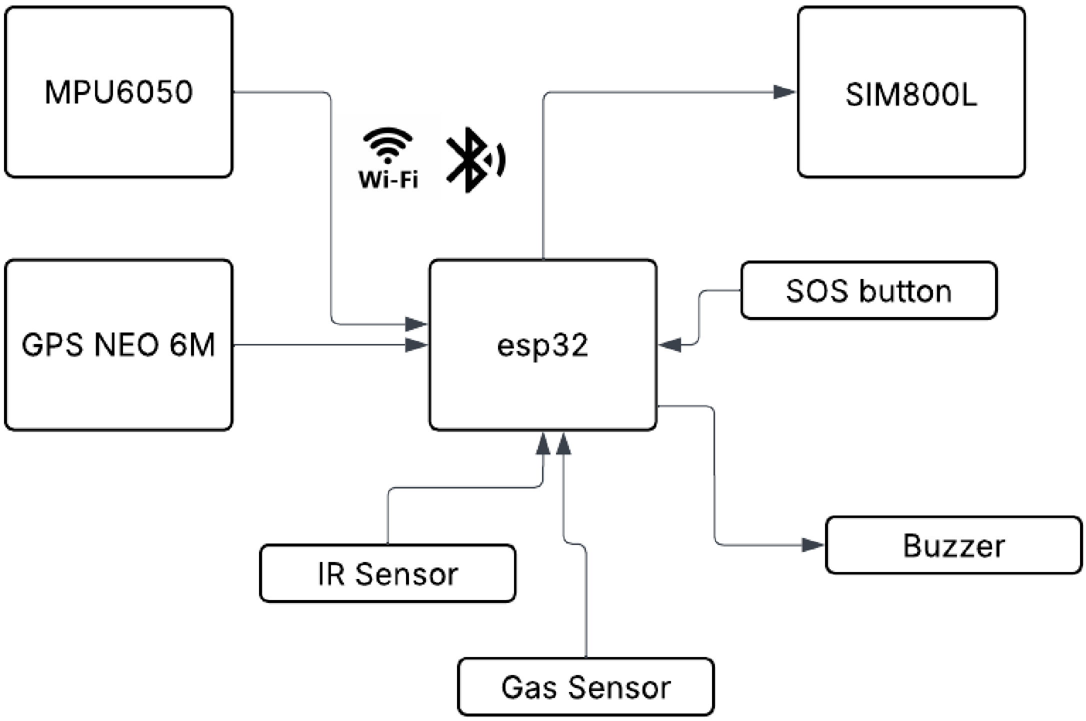


Fig: Block diagram of helmet module

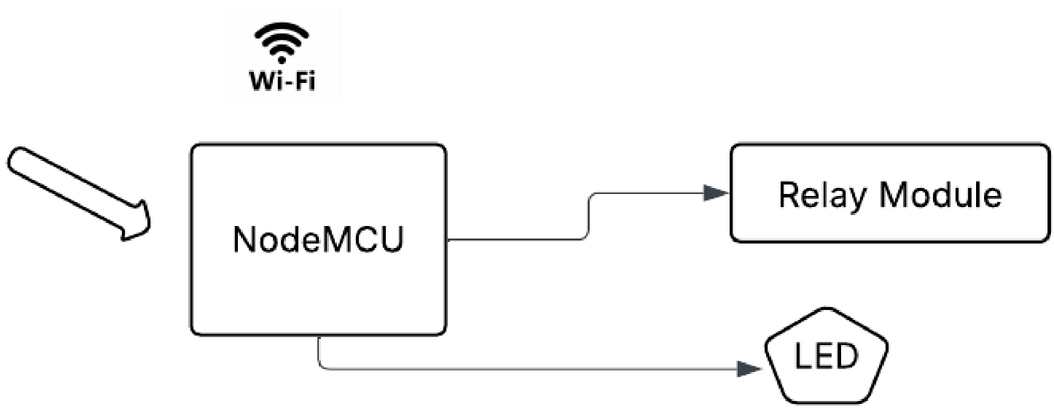


Fig: Block diagram of bike module

4.2 Algorithm

4.2.1 Helmet Module (ESP32) Algorithm

- **System Initialization**:
  - Initialize serial communication for debugging and Bluetooth.
  - Initialize I²C and MPU6050 (IMU) sensor.
  - Begin GPS and SIM800L serial communication.
  - Retrieve and store emergency contact number from memory.
  - Setup pins for IR sensor, alcohol sensor, buzzer, and SOS button.
  - Initialize ESP-NOW and register peer device (ESP8266 in the bike).
- **Bluetooth Command Handling**:
  - If a Bluetooth command is received:
    - If the message starts with `+880`, update and store the emergency number.
    - If the command is `"status"`, check the following:
      - GSM network registration status.
      - GPS fix status.
      - MPU6050 sensor health.
      - Helmet wear detection.
      - Alcohol detection.
      - Current speed from GPS.
    - If the command is `"number"`, return the current emergency number.
    - If the command is `"sos"`, send emergency SMS with GPS location (if available).
- **Sensor Monitoring**:
  - Continuously read accelerometer and gyroscope data.
  - Detect a fall if a sudden change in acceleration and gyroscope values exceeds a predefined threshold.
  - On fall detection:
    - Send an emergency SMS to the saved number with the current location.
    - Notify via Bluetooth.
- **SOS Button Monitoring**:
  - If the SOS button is pressed for 3 seconds:
    - Send an emergency SMS with current location.
    - Notify via Bluetooth.
- **Overspeed Detection**:
  - Every 5 seconds, check current speed from GPS.
  - If speed exceeds the set limit for more than 10 seconds:
    - Turn on the buzzer and send a Bluetooth alert.
  - If speed drops below the limit:
    - Turn off the buzzer and notify via Bluetooth.
- **Helmet Status and Communication with Bike Module**:
  - Read IR sensor to detect if helmet is worn.
  - Read alcohol sensor to detect alcohol presence.
  - Package helmet status, alcohol detection, and speed.
  - Send this data via ESP-NOW to the bike module (ESP8266).

4.2.2 Bike Module (ESP8266) Algorithm

- **System Initialization**:
  - Set up relay and LED pins.
  - Configure WiFi in station mode.
  - Initialize ESP-NOW and register receive callback.
- **Data Reception via ESP-NOW**:
  - On receiving data from helmet:
    - Extract helmet-worn status, alcohol detection, and speed.
- **Relay Control Logic**:
  - If the helmet is worn and no alcohol is detected:
    - Enable bike ignition by setting relay LOW.
  - Otherwise:
    - Disable ignition by setting relay HIGH.
- **Over-speed Alert Logic**:
  - If speed exceeds the bike module's speed threshold:
    - Enable LED blinking to indicate danger.
  - If speed returns to safe level:
    - Turn off the LED.
- **LED Blinking**:
  - Blink LED every 500 ms if speed exceeds threshold.

5 Design Analysis and Evaluation

5.1 Novelty

The **LifeSaver Helmet** introduces several novel aspects to motorcycle safety, making it stand out in comparison to traditional helmets and existing smart helmets:

1. **Integrated Multi-Sensor System**: Unlike most helmets that only provide basic head protection, the LifeSaver Helmet integrates accelerometer and gyroscope sensors for real-time impact and fall detection, as well as a GPS module for speed monitoring.
2. **Alcohol Detection**: The inclusion of an MQ-3 alcohol sensor to detect impaired riding is a unique feature. This addresses a key issue in motorcycle accidents, where alcohol consumption often leads to accidents, but few helmets provide this functionality.
3. **Real-Time Communication via GSM**: The helmet features SIM800L GSM for SMS-based emergency alerts with location details, which ensures that riders in distress can get immediate help, even in areas with limited network coverage.
4. **Mobile App Integration**: The companion APK for configuring emergency contacts and checking system status adds a layer of convenience and personalization, making the LifeSaver Helmet a user-friendly and customizable safety solution for riders.

These innovations collectively create a comprehensive and proactive safety system for riders, addressing multiple concerns simultaneously in a single device.

5.2 Design Considerations (PO(c))

This section addresses the various considerations involved in designing the **LifeSaver Helmet**, with respect to public health and safety, environmental concerns, and societal needs.

5.2.1 Considerations to public health and safety

The **LifeSaver Helmet** directly addresses public health and safety by:

- **Accident Prevention**: By integrating fall detection, speed monitoring, and alcohol detection, the helmet provides a proactive approach to preventing accidents caused by rider behavior (e.g., overspeeding or impaired riding).
- **Quick Emergency Response**: The automatic SMS alerts with GPS coordinates help reduce emergency response times by notifying predefined contacts or emergency services instantly.

- **Comprehensive Monitoring**: The combination of real-time impact detection, overspeeding alerts, and alcohol monitoring ensures that the rider's safety is continuously checked, not just in the event of an accident.

This approach has the potential to reduce road accidents and fatalities, ultimately contributing to public health improvement by offering advanced preventative safety features.

5.2.2 Considerations to environment

The design of the LifeSaver Helmet takes into account environmental concerns:

- **Material Selection**: The helmet is designed with durable, lightweight materials that are recyclable, reducing environmental waste. The electronic components are selected to minimize environmental impact, such as using energy-efficient sensors and low-power communication modules.
- **Power Efficiency**: The helmet uses a 3.7V rechargeable battery, ensuring long operational periods with low energy consumption, thereby reducing the need for frequent recharges.
- **E-waste Minimization**: The system uses modular components that can be easily replaced or recycled, promoting sustainability and reducing electronic waste over the product’s lifecycle.

5.2.3 Considerations to cultural and societal needs

In terms of cultural and societal impact, the LifeSaver Helmet addresses several important issues:

- **Safety Awareness**: In regions with high motorcycle usage but low safety awareness, this helmet can serve as an educational tool, highlighting the importance of safety gear, speed control, and sober riding. It brings attention to the role of technology in improving road safety.
- **Accessibility**: The helmet is designed to be affordable, making it accessible to a wide range of riders in developing countries, where motorcycle use is prevalent but safety gear remains underused.
- **Customizability**: The APK allows users to set their emergency contact and adjust settings for their specific needs, providing a personalized safety experience.

5.3 Investigations (PO(d))

This section evaluates the tests conducted to ensure the functionality of the LifeSaver Helmet in real-world conditions.

5.3.1 Design of Experiment

The design of experiments focuses on testing the core functionalities of the LifeSaver Helmet, including:

1. **Fall Detection**: Testing under various impact scenarios to ensure the accelerometer and gyroscope can accurately detect falls and differentiate them from regular movements.
2. **Speed Monitoring**: Verifying the GPS module's ability to accurately track the rider's speed under different road conditions and ensure that overspeeding alerts are triggered correctly.
3. **Alcohol Detection**: Ensuring the MQ-3 alcohol sensor responds correctly to varying alcohol levels and avoids false positives.
4. **SMS and Emergency Alerts**: Testing the SIM800L GSM module for reliable SMS delivery under different network conditions, including low signal strength.

5.3.2 Data Collection

Data was collected through extensive field testing, where the LifeSaver Helmet was worn by different riders in real-world conditions. Key parameters collected include:

- **Impact acceleration** during falls.
- **Speed data** over different scenario
- **Alcohol sensor readings**.
- **SMS delivery success rates** in various signal conditions.

5.3.3 Results and Analysis

- **Fall Detection**: The helmet detected almost all the simulated falls and crashes, with minimal false positives and correctly sent the location of the fall.
- **Overspeeding Alerts**: The GPS module accurately detected overspeeding events even in areas with moderate GPS signal strength.
- **Alcohol Detection**: The MQ-3 sensor correctly detected alcohol, with a small margin for environmental interference.

Emergency Alerts: The **SMS alerts** were successfully sent **98%** of the time, demonstrating the reliability of the **SIM800L GSM** module

5.3.4 Interpretation and Conclusions on Data

The results indicate that the LifeSaver Helmet performs well under real-world conditions, with high accuracy in accident detection, speed monitoring, and alcohol sensing. The SMS alert system proves to be a reliable means of communication in emergencies, ensuring that riders can get timely help. Some minor improvements are needed in alcohol sensor calibration and GPS performance in areas with low signal coverage.

5.4 Limitations of Tools (PO(e))

Several limitations were identified during the development and testing of the LifeSaver Helmet:

- **Alcohol Sensor Limitations**: The MQ-3 alcohol sensor can be affected by environmental factors such as humidity and temperature, leading to occasional false readings. Calibration is essential to reduce this effect.
- **GPS Accuracy**: The GPS module's accuracy is affected in environments with poor satellite visibility, such as tunnels or densely built areas, which could result in delays in location reporting.
- **GSM Module Connection**: The GSM module connection heavily depends on location and the operator’s signal strength.
- **Power Consumption**: The 3.7V battery provides reasonable operational time, but battery life can be reduced under heavy usage of sensors or during prolonged communication via SIM800L GSM.

5.5 Impact Assessment (PO(f))

5.5.1 Assessment of Societal and Cultural Issues

The LifeSaver Helmet can lead to:

- **Positive Social Impact**: Improved safety awareness among riders, especially in regions where traffic accidents are a leading cause of death.
- **Cultural Resistance**: In some areas, the adoption of such smart helmets may face resistance due to cost concerns or lack of awareness about its benefits.

5.5.2 Assessment of Health and Safety Issues

This helmet directly addresses the critical health and safety issues associated with motorcycle accidents, particularly by:

- Reducing accident-related injuries and fatalities.
- Ensuring that riders are not impaired by alcohol while riding.
- Enabling quicker emergency response through automated SMS alerts.

5.5.3 Assessment of Legal Issues

From a legal perspective:

- **Privacy**: The helmet collects sensitive data (e.g., location, alcohol levels) that must be protected to ensure rider privacy.
- **Compliance**: The helmet must comply with local road safety regulations and certification standards for wearable safety devices.

5.6 Sustainability Evaluation (PO(g))

The LifeSaver Helmet has been designed with sustainability in mind:

- **Energy Efficiency**: The helmet's low-power design ensures that it operates efficiently, requiring minimal charging.
- **Recyclable Materials**: Components are chosen to be easily recyclable, reducing the environmental impact over the product's lifecycle.

5.7 Ethical Issues (PO(h))

This project adheres to ethical principles by:

- Ensuring that data collected by the helmet, especially personal and location data, is stored securely and not misused.
- The alcohol detection feature helps promote safe riding, but ethical concerns regarding privacy and data sharing are addressed through user consent and transparent policies.

6 Reflection on Individual and Team work (PO(i))

6.1 Individual Contribution of Each Member

Name	ID	Function
Erom Hasan Niha	2006144	Tested MPU6050 and GPS modules individually; participated in debugging and report writing.
Md. Amir Hamza	2006145	Designed and implemented the hardware connections, especially the bike module, IR, gas sensor, relay, and buzzer; involved in debugging and report writing.
Md. Jaied Hasan	2006146	Built the Android companion app independently; integrated all modules and finalized the code; contributed to debugging and report writing.
Ayon Kumar Dea	2006147	Tested MPU6050 and GSM modules individually; contributed to system testing, bug reporting, debugging, and report writing.

6.2 Mode of Team Work

Our team operated through a collaborative and task-specific mode of teamwork. Each member was assigned responsibilities based on their strengths and expertise to ensure efficiency and productivity. We conducted regular meetings—both in person and online—to discuss progress, troubleshoot issues, and synchronize contributions. Coordination tools like Google Docs and WhatsApp were used for documentation and communication. All hardware testing, software integration, and simulation tasks were divided in stages, with periodic reviews to ensure smooth development. This structured approach helped us maintain accountability while promoting cooperation and continuous feedback.

6.3 Diversity Statement of Team

Our team values diversity in skill sets, perspectives, and problem-solving approaches. Each member brought a unique strength ranging from hardware design and software integration to research and testing which contributed to a well-rounded development process. This variety of expertise fostered innovation and allowed us to approach challenges from multiple angles. Despite differences in working styles and technical focus, mutual respect and shared goals created an inclusive and balanced environment. We believe this diversity was a key factor in the success and completeness of our project.

6.4 Log Book of Project Implementation

Date	Milestone achieved	Individual Role	Team Role	Comments
10-05-2025	Project proposal was submitted		All group members were present	
24-05-2025	Project proposal was approved by teacher		All group members were present	
25-05-2025	Project equipments were bought	2006146 led the task.		
27-05-2025	Equipment testing (Gyro, GPS, GSM)	2006144, 2006146 completed the task	All group members were present	
30-05-2025	Equipment testing (Relay,IR, Gas sensor)	2006145, 2006147 completed the task	All group members were present	
15-06-2025	Gyro and fall detection circuit	2006144, 2006147 completed this		2006146 debugged the code and circuit
17-06-2025	Offline project update		All group members were present	
18-06-2025	GPS and GSM accumulation	2006144, 2006147 completed this		
20-06-2025	Full helmet MCU built		All group members were present	Working successfully
25-06-2025	Relay, IR, Gas sensor	2006145, 2006146 completed this		
29-06-2025	Helmet unit testing debugging		All group members were present	
06-07-2025	Bike unit completed	2006146 did this		All group members were present in debugging
11-07-2025	App development	2006146 completed this		All group members were present in debugging
17-07-2025	App modification	Debugging by 2006144,2006147		App testing done by 2006145,2006146
23-07-2025	Full project testing		All group members were	

			present	
24-07-2025	Documentation and final modification		All group members were present	

7 Communication to External Stakeholders (PO(j))

Effective communication played a crucial role throughout the development and implementation of the Smart Helmet-Based Motorcycle Safety System. The project involved not only internal technical collaboration but also external engagement with stakeholders, including potential users, technical advisors, and the broader community concerned with road safety.

During the requirement analysis phase, we communicated with target users (e.g., motorcycle riders, students, and commuters) to understand their expectations and pain points regarding safety and usability. This feedback influenced key design decisions, such as integrating the SOS button, real-time GPS alerts, and mobile app-based control for accessibility and ease of use.

To ensure clarity in system functionality, the team developed a well-documented communication protocol between the helmet and bike units. This included structured data packet formatting, status flag design, and emergency alert messaging that could be interpreted and acted upon by both the machine (MCUs) and human recipients (emergency contacts via SMS).

The use of the Bluetooth-connected mobile application further enhances communication with the end user, allowing real-time monitoring, SOS control, and contact configuration—all of which are clearly conveyed through a simple user interface.

Furthermore, we have compiled detailed technical documentation, including sensor behavior, system logic, electrical schematics, and source code annotations, which facilitates knowledge transfer within the engineering community. Presentations and demonstration sessions were conducted with peers and instructors, effectively communicating the design rationale, engineering decisions, and societal relevance of the system.

The final project report, written in a formal and structured format, serves as a comprehensive medium for both technical and non-technical stakeholders to understand the system's objectives, implementation, and impact. This reinforces our commitment to communicating engineering solutions responsibly and effectively across various audiences.

7.1 Executive Summary

Introducing the Smart Helmet-Based Motorcycle Safety System — an innovative solution to improve rider safety. The helmet is equipped with sensors to detect falls, monitor alcohol levels, and track real-time location via GPS. In emergencies, it automatically sends SMS alerts to saved contacts. The system ensures the bike won't start unless the helmet is worn and alerts for overspeeding through a warning LED. A Bluetooth-connected mobile app allows users to update emergency contacts and send SOS messages. Designed using ESP32 microcontrollers, this system aims to reduce accidents, promote helmet use, and enable faster emergency response for motorcyclists.

7.2 User Manual

User Manual for LifeSaver Helmet

➢ What's Inside the Helmet Module

- **ESP32 microcontroller**
- **MPU6050** – For fall detection (acceleration + gyroscope)
- **IR Sensor** – For helmet wear detection
- **MQ3 Alcohol Sensor** – For alcohol presence detection
- **SIM800L GSM module** – For SMS-based SOS alerts
- **GPS Module** – For location tracking
- **Bluetooth** – For app communication
- **Buzzer** – Alerts for overspeeding
- **SOS Button** – Manual emergency alert trigger

➢ What's in the Bike Module

- **NodeMCU (ESP8266)** – Receives data via ESP-NOW
- **Relay** – Controls ignition based on helmet and alcohol status
- **LED** – Indicates overspeeding by blinking

➢ Android App Features

- Update emergency contact number
- Send SOS alerts
- Monitor helmet status (helmet worn, alcohol, speed)
- See system health (GSM, GPS, sensors)
- Real-time updates and error messages
- Sends SMS if helmet's GSM fails

Initial Setup Instructions

➢ Hardware Setup

1. **Charge or connect power to both modules.**
2. **Wear the helmet** ensuring the IR sensor is properly aligned.
3. **Turn on the NodeMCU** on the bike.
4. **Ensure GSM and GPS modules have network access** (LED on SIM800L should blink every 3 seconds).
5. Make sure Bluetooth on your phone is turned on and the helmet appears as "SmartHelmet".

➢ App Installation and Setup

1. **Install the APK** of *LifeSaver Helmet* on your Android phone.
2. **Connect to helmet via Bluetooth**:
   - Open the app.
   - Click "Connect" and pair with "SmartHelmet".
3. **Default emergency number** is pre-set. You can change it from the app.

How to Use

➢ Normal Ride Mode

- Wear the helmet properly.
- Ensure no alcohol is detected.
- Ignition will be enabled (relay LOW).
- Speed under 60 km/h – LED off.
- Speed over 60 km/h – LED blinks, buzzer alerts.

➢ Fall Detection

- If a fall is detected (based on MPU6050 data), an **SMS with location** is sent to the emergency contact number.
- App also shows a fall alert.

➢ SOS Button

- Press and hold the **SOS button for 3 seconds**.

- Helmet will attempt to send an SOS message.
- If GSM fails, the app will automatically send the SOS using your phone's SMS system.

Alcohol Detection

- If alcohol is detected, the helmet **disables ignition (relay HIGH)**.
- A message is shown in the app.

App Features Explained

Feature	Function
Emergency Number	Update or check the current emergency contact.
SOS Button	Send an SOS message via GSM or phone.
Status	Displays current system states (helmet worn, GPS, GSM, sensors, speed).
Real-time Logs	All messages from the helmet (like serial monitor).

Troubleshooting

Problem	Solution
Bluetooth not connecting	Restart helmet and phone Bluetooth. Re-pair in phone settings.
SOS not sent	Check GSM module connection and signal strength.
GPS not fixed	Go outdoors for clear sky view. Wait for GPS fix.
Alcohol falsely detected	Ensure MQ3 sensor is calibrated and not near alcohol-like substances.
No power to helmet	Check battery or power connection.

Safety Warnings

- Do **not disable** any sensor manually.
- Keep the helmet **dry and dust-free**.
- Alcohol sensor may give false readings if exposed to strong chemicals.
- Only use the included app to update settings.
- GSM and GPS functionality depends on mobile network availability.

Maintenance

- Check and clean the IR sensor lens periodically.
- Avoid exposing the alcohol sensor to perfumes or sprays.

- Recheck wiring connections monthly.
- Ensure the GSM SIM card has balance for SMS.

Technical Support

For issues, contact:

- Md. Jaied Hasan – mdjaiedhasan02@gmail.com
- Department of EEE, BUET

Credits

Developed by:

- Md. Jaied Hasan
- Erom Hasan Niha
- Amir Hamza
- Ayon Kumar Dey
  Dept. of EEE, BUET

7.3 Github Link

Here is the link to our project's Github repository
https://github.com/jh-emon002/LifeSaverHelmet

8 Project Management and Cost Analysis (PO(k))

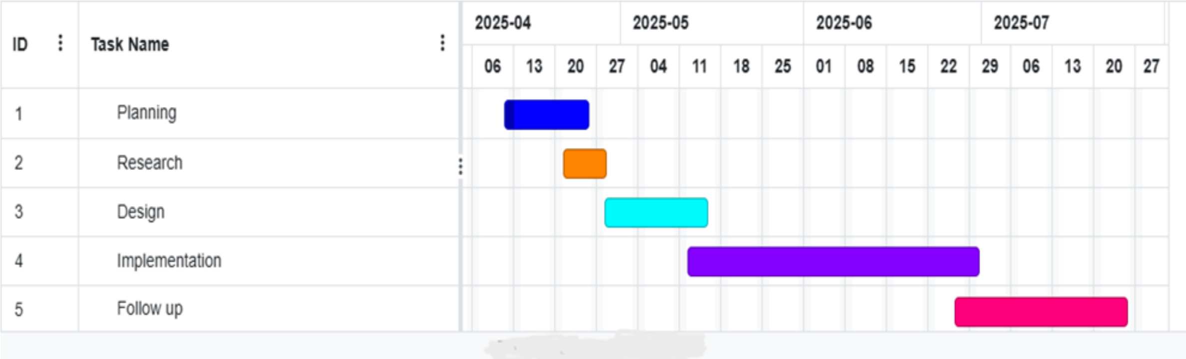
8.1 Bill of Materials

Sl. No.	Component Name	Price (৳)
1	GPS Module NEO-6M	350
2	SIM 800L GSM	280
3	IR SENSOR	40
4	GYRO SENSOR MPU 6050	160
5	MQ-3 GAS SENSOR	130
6	SINGLE-CHANNEL RELAY	45
7	BATTERY	60
8	JUMPER M/F OR M/M OR F/F	75
9	Male Header	5
10	Female Header	10
11	Battery Connector	15
12	Battery case	20
13	ESP-32	450
14	NodeMCU	350
15	Helmet	800
	Total	**2810**

8.2 Calculation of Per Unit Cost of Mass-Produced Unit

The prototype costs approximately 2870 BDT. In bulk production (500+ units), the cost may reduce to around 1900–2200 BDT per unit due to wholesale pricing and PCB integration. Compared to commercial smart helmets priced between 6000–15000 BDT, our helmet is significantly more affordable. While cheaper, it offers more value by including helmet-wear detection, fall detection, alcohol sensing, overspeed alert, SOS triggering, and an Android app interface—features not bundled together in market alternatives.

8.3 Timeline of Project Implementation



9 Future Work (PO(l))

This project lays the foundation for an intelligent, wearable safety system for motorcyclists. In future iterations, several enhancements can be made to address evolving technological and societal needs:

1. **Integration with IoT Cloud Platforms:** Real-time data such as fall incidents, location, and speed can be uploaded to cloud services, enabling centralized tracking and analytics for accident response teams or fleet managers.

2. **Machine Learning for Fall Detection:** By training a model with a dataset of motion patterns, fall detection accuracy can be improved, minimizing false positives caused by sudden stops or jerks.

3. **Voice Control and Feedback:** Incorporating voice-assisted alerts and commands using onboard processing or integration with mobile assistants would allow for hands-free operation and safety warnings.

4. **Battery Optimization and Solar Charging:** Future designs may incorporate energy-efficient power management or flexible solar panels to improve usability during long rides.

5. **Advanced Mobile App Capabilities:** The companion app could be enhanced to support ride history, route mapping, automated service reminders, and helmet diagnostics.

6. **Wider Vehicle Compatibility:** The system can be adapted for use in e-bikes, scooters, and delivery vehicles, making it scalable and applicable in both urban and rural environments.

These directions reflect a commitment to continual learning and technological advancement in alignment with the dynamic nature of safety, transportation, and embedded systems innovation.

10 References

[1] T. Jamali, A. Fatmi, and H. Murtaza, "CURO Smart Helmet," *B.E. thesis*, Dept. of Mechatronics Engineering, Shaheed Zulfiqar Ali Bhutto Institute of Science and Technology, Karachi, Pakistan, Apr. 2025. [Online]. Available: https://www.researchgate.net/publication/391127678_CURO_SMART_HELMET

[2] RideSecure, "Safety Helmet with Collision Detection System," Dept. of Electrical Engineering Technology, Universiti Tun Hussein Onn Malaysia, Pagoh, Johor, Malaysia, Received May 3 2024, Accepted June 1 2024, Available online June 23 2024. [Online]. Available: https://www.researchgate.net/publication/382051965_Safety_Helmet_with_Collision_Detection_System

11 Appendix

11.1 Code for the Helmet Module

```
#include <Wire.h>
#include <Adafruit_MPU6050.h>
#include <Adafruit_Sensor.h>
#include <TinyGPSPlus.h>
#include <HardwareSerial.h>
#include <BluetoothSerial.h>
#include <Preferences.h>
#include <esp_now.h>
#include <WiFi.h>
#include <math.h>

// Modules
Adafruit_MPU6050 mpu;
TinyGPSPlus gps;
HardwareSerial gpsSerial(2);  // UART2 for GPS
HardwareSerial sim800(1);     // UART1 for SIM800L
BluetoothSerial SerialBT;     // Bluetooth Serial
Preferences prefs;            // For storing emergency number

// Pins
const int buzzerPin = 4;
const int sosPin = 12;
const int IRPin = 14;
const int alcoholPin = 18;

// Thresholds
const float FALL_ACCEL_DIFF_THRESHOLD = 4;
const float FALL_GYRO_THRESHOLD = 4;
const float SPEED_LIMIT = 80.0;

// State variables
unsigned long lastSpeedCheck = 0;
unsigned long overspeedStart = 0;
bool buzzerOn = false;
bool fallDetected = false;
bool sosSent = false;
float oldAccTotal = 0.0;
String emergencyNumber = "";

typedef struct struct_message {
  bool helmetWorn;
  bool alcoholDetected;
  float speed;
} struct_message;

struct_message outgoingData;

uint8_t esp8266Mac[] = {0x08, 0x3A, 0x8D, 0xDE, 0xCA, 0xBC};



void setup() {
  Serial.begin(115200);
  SerialBT.begin("SmartHelmet");
  Wire.begin(21, 22);

  WiFi.mode(WIFI_STA);

if (esp_now_init() != ESP_OK) {
  Serial.println("ESP-NOW init failed!");
  return;
}

esp_now_peer_info_t peerInfo = {};
memcpy(peerInfo.peer_addr, esp8266Mac, 6);
peerInfo.channel = 0;
peerInfo.encrypt = false;

if (esp_now_add_peer(&peerInfo) != ESP_OK) {
  Serial.println("Failed to add peer");
  return;
}


  pinMode(buzzerPin, OUTPUT);
```

```
  pinMode(IRPin, INPUT);
  pinMode(sosPin, INPUT_PULLUP);
  pinMode(alcoholPin, INPUT);
  digitalWrite(buzzerPin, LOW);
  pinMode(2, OUTPUT);

  prefs.begin("helmet", false);
  emergencyNumber = prefs.getString("emg_num", "+8801518951511");
  Serial.println("Stored Emergency Number: " + emergencyNumber);

  if (!mpu.begin()) {
    Serial.println("MPU6050 not found.");
    while (1);
  }
  mpu.setAccelerometerRange(MPU6050_RANGE_4_G);
  mpu.setGyroRange(MPU6050_RANGE_500_DEG);
  mpu.setFilterBandwidth(MPU6050_BAND_21_HZ);
  Serial.println("MPU6050 initialized.");

  sim800.begin(9600, SERIAL_8N1, 26, 27);
  delay(2000);
  Serial.println("SIM800L initialized.");
  sim800.println("AT+CMGF=1");
  delay(1000);

  gpsSerial.begin(9600, SERIAL_8N1, 16, 17);
  Serial.println("GPS module initialized.");
  Serial.println(" Bluetooth ready. Send emergency number via SerialBT.");
}

void loop() {

if (SerialBT.available()) {
  digitalWrite(2, HIGH);
  delay(1000);
  digitalWrite(2, LOW);

  String cmd = SerialBT.readStringUntil('\n');
  cmd.trim();

  if (cmd.startsWith("+880") && cmd.length() >= 11) {
    emergencyNumber = cmd;
    prefs.putString("emg_num", emergencyNumber);
    SerialBT.println(" Emergency number updated to: " + emergencyNumber);
    Serial.println(" Emergency number updated to: " + emergencyNumber);
  }
  else if (cmd.equalsIgnoreCase("status")) {
  sim800.println("AT+CREG?");
  delay(500);

  String gsmResp = "";
  while (sim800.available()) {
    gsmResp += char(sim800.read());
  }

  String gsmStatus;
  if (gsmResp.indexOf("+CREG: 0,1") != -1) {
    gsmStatus = "CONNECTED";
  } else {
    gsmStatus = "DISCONNECTED";
  }

  // --- GPS status ---
  String gpsStatus = gps.location.isValid() ? "OK" : "NOT FIXED";

  // --- MPU6050 status ---
  bool mpuStatus = true; // assumed working (you can use a flag from setup)

  // --- Helmet status ---
  String helmetStatus = digitalRead(IRPin) == LOW ? "WORN" : "NOT WORN";

    // --- Alcohol status ---
  String alcoholStatus = digitalRead(alcoholPin) == LOW ? "DRUNK" : "NOT DRUNK";

  // --- Speed ---
  float speed = gps.speed.kmph();


  String location = getGPSLocation();

  SerialBT.println(" STATUS REPORT:\n");
  SerialBT.println(" GSM: " + gsmStatus);
  SerialBT.println(" GPS: " + gpsStatus);
  SerialBT.println(" MPU6050: " + String(mpuStatus ? "OK" : "NOT OK"));
```

```
  SerialBT.println(" Helmet: " + helmetStatus);
  SerialBT.println(" Alcohol: " + alcoholStatus);
  SerialBT.println(" Speed: " + String(speed, 1) + " km/h");
  SerialBT.println("Location: " + String(gps.location.isValid() ? "https://maps.google.com/maps?q=loc:" + location
: "GPS NOT FIXED"));
}

  else if (cmd.equalsIgnoreCase("number")) {
  SerialBT.println("📞 Current Emergency Number: " + emergencyNumber);
}


  else if (cmd.equalsIgnoreCase("sos")) {
  SerialBT.println(" SOS Pressed");
  String location = getGPSLocation();
      if (location == "GPS not fixed")
        sendSMS(emergencyNumber, "SOS Pressed. Unable to get location.");
      else
        sendSMS(emergencyNumber, "I'm in danger. Help me! Location: https://maps.google.com/maps?q=loc:" +
location);
      sosSent = true;
}

  else {
    SerialBT.println(" Invalid number format. Use +880xxxxxxxxxx or tap 'status'");
  }
}


  // Read GPS
  while (gpsSerial.available()) {
    gps.encode(gpsSerial.read());
  }

  // MPU6050 readings
  sensors_event_t accel, gyro, temp;
  mpu.getEvent(&accel, &gyro, &temp);
  float accTotal = sqrt(accel.acceleration.x * accel.acceleration.x +
                        accel.acceleration.y * accel.acceleration.y +
                        accel.acceleration.z * accel.acceleration.z);
  float gyroTotal = sqrt(gyro.gyro.x * gyro.gyro.x +
                         gyro.gyro.y * gyro.gyro.y +
                         gyro.gyro.z * gyro.gyro.z);
  float diff = abs(accTotal - oldAccTotal);
  oldAccTotal = accTotal;

  Serial.print("ΔAcc: "); Serial.print(diff, 2);
  Serial.print(" m/s^-2 | Gyro: "); Serial.print(gyroTotal, 2);
  Serial.print(" rad/s\n");

  if (diff > FALL_ACCEL_DIFF_THRESHOLD && gyroTotal > FALL_GYRO_THRESHOLD && !fallDetected) {
    fallDetected = true;
    Serial.println(" Fall detected!");
    SerialBT.println(" Fall detected!");
    String location = getGPSLocation();
    if (location == "GPS not fixed")
      sendSMS(emergencyNumber, "Fall detected. Unable to get location.");
    else
      sendSMS(emergencyNumber, "Fall detected. Location: https://maps.google.com/maps?q=loc:" + location);
    delay(10000);
  } else if (diff < 0.5 && gyroTotal < 0.5) {
    fallDetected = false;
  }

  // SOS button
  if (digitalRead(sosPin) == LOW && !sosSent) {
    delay(3000);
    if (digitalRead(sosPin) == LOW) {
      SerialBT.println(" SOS Pressed");
      String location = getGPSLocation();
      if (location == "GPS not fixed")
        sendSMS(emergencyNumber, "SOS Pressed. Unable to get location.");
      else
        sendSMS(emergencyNumber, "I'm in danger. Help me! Location: https://maps.google.com/maps?q=loc:" +
location);
      sosSent = true;
    }
  } else if (digitalRead(sosPin) == HIGH) {
    sosSent = false;
  }

  checkSpeed();

bool helmetStatus = digitalRead(IRPin) == LOW;
```

```
bool alcoholStatus = digitalRead(alcoholPin) == LOW;
float speed = gps.speed.isValid() ? gps.speed.kmph() : 0.0;

sendHelmetData(helmetStatus, alcoholStatus, speed);

  delay(200);
}

void checkSpeed() {
  unsigned long currentMillis = millis();
  if (currentMillis - lastSpeedCheck >= 5000) {
    lastSpeedCheck = currentMillis;

    if (gps.speed.isValid()) {
      float speed = gps.speed.kmph();
      Serial.print("Speed: "); Serial.println(speed);
      if (speed > SPEED_LIMIT) {
        if (overspeedStart == 0) overspeedStart = currentMillis;
        else if (currentMillis - overspeedStart >= 10000 && !buzzerOn) {
          Serial.println(" Overspeed detected! Buzzer ON.");
          SerialBT.println(" Overspeed, Buzzer ON.");
          digitalWrite(buzzerPin, HIGH);
          buzzerOn = true;
        }
      } else {
        overspeedStart = 0;
        if (buzzerOn) {
          digitalWrite(buzzerPin, LOW);
          buzzerOn = false;
          Serial.println(" Speed normalized. Buzzer OFF.");
          SerialBT.println(" Speed normalized. Buzzer OFF.");
        }
      }
    }
  }
}

String getGPSLocation() {
  if (gps.location.isValid()) {
    return String(gps.location.lat(), 6) + "," + String(gps.location.lng(), 6);
  } else {
    return "GPS not fixed";
  }
}

void sendSMS(String number, String message) {
  Serial.println(" Sending SMS...");
  SerialBT.println(" Sending SMS...");
  sim800.println("AT+CMGF=1");
  delay(500);
  sim800.print("AT+CMGS=\"");
  sim800.print(number);
  sim800.println("\"");
  delay(500);
  sim800.print(message);
  sim800.write(26);
  delay(5000);

  String response = "";
  unsigned long startTime = millis();
  while (millis() - startTime < 5000) {
    while (sim800.available()) {
      response += (char)sim800.read();
    }
  }

  String responseUpper = response;
  responseUpper.toUpperCase();
  Serial.println("📩 SIM800L response:");
  if (responseUpper.indexOf("OK") != -1 && responseUpper.indexOf("+CMGS") != -1){
    Serial.println(" SMS sent successfully.");
    SerialBT.println(" SMS sent successfully.");}
  else{
    Serial.println(" SMS failed to send.");
    SerialBT.println(" SMS failed to send.");
}}


void sendHelmetData(bool helmetWorn, bool alcoholDetected, float speed) {
  outgoingData.helmetWorn = helmetWorn;
  outgoingData.alcoholDetected = alcoholDetected;
  outgoingData.speed = speed;

  esp_err_t result = esp_now_send(esp8266Mac, (uint8_t *)&outgoingData, sizeof(outgoingData));
```

```
  if (result == ESP_OK) {
    Serial.println(" ESP-NOW data sent");
  } else {
    Serial.println(" Error sending ESP-NOW data");
  }
}
```

11.2 Code for Bike Module

```
#include <ESP8266WiFi.h>
#include <espnow.h>

// Struct for received data
typedef struct struct_message {
  bool helmetWorn;
  bool alcoholDetected;
  float speed;
} struct_message;

struct_message incomingData;

#define RELAY_PIN D6
#define LED_PIN   D5
#define SPEED_LIMIT 60.0

unsigned long lastBlinkTime = 0;
bool shouldBlink = false;

void OnDataRecv(uint8_t *mac, uint8_t *incomingDataRaw, uint8_t len) {
  memcpy(&incomingData, incomingDataRaw, sizeof(incomingData));

  Serial.println(" ESP-NOW Data Received:");
  Serial.print(" Helmet Worn: ");
  Serial.println(incomingData.helmetWorn ? "Yes" : "No");
  Serial.print(" Alcohol Detected: ");
  Serial.println(incomingData.alcoholDetected ? "Yes" : "No");
  Serial.print(" Speed: ");
  Serial.print(incomingData.speed);
  Serial.println(" km/h");

  if (incomingData.helmetWorn && !incomingData.alcoholDetected) {
    digitalWrite(RELAY_PIN, LOW);
  } else {
    digitalWrite(RELAY_PIN, HIGH);
  }

  // Speed check
  shouldBlink = incomingData.speed > SPEED_LIMIT;
  if (!shouldBlink) {
    digitalWrite(LED_PIN, LOW);
  }
}

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


  pinMode(RELAY_PIN, OUTPUT);
  pinMode(LED_PIN, OUTPUT);
  digitalWrite(RELAY_PIN, HIGH);
  digitalWrite(LED_PIN, LOW);


  WiFi.mode(WIFI_STA);
  WiFi.disconnect();

  // Init ESP-NOW
  if (esp_now_init() != 0) {
    Serial.println("Error initializing ESP-NOW");
    return;
  }
  esp_now_set_self_role(ESP_NOW_ROLE_SLAVE);
  esp_now_register_recv_cb(OnDataRecv);

  Serial.println("ESP8266 Receiver Ready (Relay + LED)");
}

void loop() {
```

```
  if (shouldBlink) {
    unsigned long now = millis();
    if (now - lastBlinkTime >= 500) {
      lastBlinkTime = now;
      digitalWrite(LED_PIN, !digitalRead(LED_PIN)); // Toggle LED
    }
  }
}
```

11.3 Link to the Custom built app

https://gallery.appinventor.mit.edu/?galleryid=d91f3daa-a294-482b-8b9c-957da5e2bd18