BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

EEE 416

Microprocessor & Embedded System Laboratory

Project Report

Section: B1; Group: 01

Enhanced Smart Helmet for Rider Safety and Accident Alert

Course Instructors:

- **Dr,Sajid Muhaimin Choudhury**
  Associate Professor, EEE, BUET

- **Akif Hamid**
  Lecturer (PT), 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: Sheikh Tariful Islam

Student ID: 2006066

Signature: ______________

Full Name: Md. Nasimul Islam Nayem

Student ID: 2006067

Signature: ______________

Full Name: Ilias Ahamed

Student ID: 2006068

Signature: ______________

Full Name: Abdullah Al Mamun

Student ID: 2006069

Table of Contents

1.Abstract

The Smart Helmet project focuses on enhancing rider safety by integrating essential detection features into a conventional two-wheeler helmet. Utilizing an Arduino-based system, this project incorporates three key functionalities: wear detection, alcohol detection, and crash detection. We use an MQ-3 sensor to monitor alcohol levels, an accelerometer for crash detection . When the helmet is not worn, alcohol is detected, or a crash occurs, the system triggers warnings through LEDs and buzzers, ensuring the rider's safety. This simple yet effective system, operating via RF communication between the helmet and vehicle, helps prevent accidents by alerting the rider or disabling the vehicle's ignition in critical situations. The project offers a cost-effective solution for improving road safety while being accessible to both beginners and experienced engineers..

2.Introduction:

The Smart Helmet System is designed to enhance rider safety by integrating advanced detection technologies into a single, affordable helmet. Motorcycle accidents are a leading cause of injury and death worldwide, with many incidents caused by lack of helmet use**,** alcohol impairment, and **crash** impacts**.** This project addresses these issues by incorporating three key features: wear detection**,** alcohol detection, and crash detection**,** MQ-3 sensors to detect alcohol levels in the rider’s breath, and accelerometers to monitor sudden impacts indicative of a crash. These features communicate wirelessly via RF modules to the vehicle, providing real-time feedback and immediate action, such as disabling the vehicle’s ignition if alcohol is detected. By integrating these safety features, the Smart Helmet System aims to reduce motorcycle-related accidents and fatalities, improving public safety and promoting responsible riding.

3.Design:

This section outlines the design approach for the Smart Helmet, detailing the integration of key functionalities, system components, and the operational workflow necessary for ensuring rider safety. The design focuses on simplicity, cost-effectiveness, and reliable performance in real-world conditions, ensuring that the helmet meets the critical requirements of wear detection, alcohol detection, and crash detection

3.1 Problem Formulation

The Smart Helmet project addresses several critical safety issues for two-wheeler riders, aiming to improve road safety through real-time monitoring and alerts. The following problem formulation identifies the scope and objectives of the project**:**

3.1.1 Identification of Scope

The scope of the project involves developing a Smart Helmet system that incorporates three essential safety features:

1. **Wear Detection:** Ensuring the rider is wearing the helmet before allowing the vehicle to start. This feature aims to prevent situations where the helmet is not worn, which significantly reduces the protective effect of the helmet in case of an accident.
2. **Alcohol Detection:** Monitoring the rider's alcohol consumption using a gas sensor (MQ-3) to prevent riding under the influence, thereby reducing the risk of accidents caused by impaired judgment or reaction time.
3. **Crash Detection:** Detecting any significant impact or crash using an accelerometer or similar sensor, triggering immediate alerts to inform others or disable the vehicle’s ignition if necessary.

By implementing these three features, the system will provide immediate feedback to both the rider and the vehicle system, ensuring that the rider is in optimal conditions for riding. The system uses RF communication to send data from the helmet to the vehicle and provide alerts through LEDs and buzzers. The project aims to improve road safety, prevent accidents, and ensure legal compliance by detecting helmet wear and alcohol consumption.

The scope is limited to real-time monitoring and alerting functionalities for these three features, without any additional advanced capabilities like GPS or long-term data logging. The system will be designed with the goal of being cost-effective, easy to implement, and reliable under typical usage conditions.

3.1.2 Literature Review

The following literature review discusses existing research, technologies, and applications related to wear detection**,** alcohol detection**,** and crash detection in helmet systems, with a focus on their integration into smart helmets for

Alcohol Detection in Helmets

The issue of alcohol consumption and its impact on road safety has prompted numerous studies on alcohol detection systems in vehicles, with some of them being integrated into helmet systems:

- **MQ-3 Sensor:** The MQ-3 gas sensor, which detects alcohol levels in breath, is one of the most widely used sensors for alcohol detection. Bharath et al. (2019**)** explored its application in automotive safety systems and demonstrated its effectiveness in detecting alcohol levels in real-time, which is crucial for preventing impaired driving .
- **Breath Analysis Systems:** Other methods include portable breath analysis devices, where the rider is required to blow into a sensor before starting the vehicle. Patel et al. (2020) reviewed various breath analysis systems, highlighting their use in both personal safety devices and vehicles . These systems often use infrared spectroscopy or chemical sensors to analyze alcohol content in breath, offering high accuracy and reliability**.**

Crash Detection in Helmets

Crash detection in helmets plays a vital role in real-time accident alerting systems. The majority of these systems rely on accelerometers and gyroscope sensors to detect abrupt motion changes, which are indicative of a crash. Some notable contributions include:

- **Accelerometer-Based Systems:** Studies have demonstrated the use of **MEMS** accelerometers **f**or detecting impacts and crashes. **Khan et al. (2017)** implemented an accelerometer-based crash detection system in helmets, which can trigger an alert or call emergency services in case of a high-impact collision . These systems use algorithms to analyze sudden deceleration or impacts, providing real-time alerts and emergency notifications.

- **Helmet with Smart Sensors: Zhao et al. (2018)** proposed integrating multiple sensors such as accelerometers, gyroscopes, and pressure sensors into helmets for comprehensive crash detection. Their system was designed to detect not only crashes but also the severity of impacts, providing more nuanced feedback to riders and emergency responders .

Integration of Wear, Alcohol, and Crash Detection in Smart Helmets:

Several studies have attempted to integrate these features into a single wearable system. For instance, **Ghosh et al. (2020)** demonstrated a multi-sensor smart helmet system that combines wear detection, alcohol monitoring, and crash detection in a single device. The study found that combining these sensors in a smart helmet was effective in enhancing rider safety by providing real-time alerts to the rider and disabling the vehicle ignition when unsafe conditions were detected .

Furthermore, **Srinivasan et al. (2019)** developed an IoT-based helmet system with integrated wear, alcohol, and crash detection features, communicating the rider's status to a mobile application for further action. This approach emphasized the importance of **wireless communication** and **real-time feedback** in ensuring safety and preventing accidents caused by impaired or distracted riders .

Gaps in Current Literature

While the integration of these technologies into smart helmets has been explored in various studies, most existing solutions are still in the prototype or early-stage development phase. Common gaps in the literature include:

1. **Cost-Effective Solutions:** Many of the current solutions require expensive sensors or advanced technology that may not be accessible for widespread implementation.
2. **Real-Time Feedback Systems:** Although real-time feedback is often implemented, integrating these systems into affordable, consumer-grade products remains a challenge.
3. **Comprehensive and Accurate Monitoring:** Current systems may not always offer the most accurate detection, especially in dynamic conditions such as variable weather or when the rider's behavior deviates from typical patterns.

3.1.3 Formulation of Problem

The goal of this project is to design and develop a Smart Helmet System that enhances rider safety by incorporating wear detection**,** alcohol detection, and crash detection functionalities. The problem formulation is outlined below, focusing on the identification of safety risks, the need for detection features, and the integration of these features into a single, effective system.

Identification of Safety Risks:

Riding a two-wheeler presents a significant risk to the rider's safety, particularly when protective measures are not in place. The main safety risks identified include:

1. **Helmet Non-Wear:** One of the most critical safety concerns is when a rider is not wearing a helmet. A helmet is essential for protecting the head in the event of a crash, and failure to wear it increases the risk of severe injury or death.
2. **Alcohol Impairment:** Alcohol consumption impairs judgment, reaction times, and motor coordination, significantly increasing the likelihood of accidents. However, there is often no immediate check to confirm the rider's sobriety before allowing the vehicle to operate.
3. **Crash Detection:** Accidents and crashes are unfortunately common on the roads. Detecting a crash in real-time and alerting the rider or emergency services can help mitigate the consequences by enabling prompt intervention.

Need for Detection Features:

To address these safety risks, the integration of detection systems into a smart helmet becomes necessary. The following detection features are required:

1. **Wear Detection:**
    - The helmet should include a wear detection feature to verify whether the rider is wearing the helmet properly. If not, the system should prevent the vehicle from starting or provide an alert.
    - **Problem:** Riders sometimes fail to wear helmets, leading to greater exposure to head injuries in case of an accident.

2. **Alcohol Detection:**
    - The system should monitor the rider's alcohol consumption using a gas sensor. If the alcohol level exceeds a predefined threshold, the system should trigger an alert or disable the vehicle ignition.
    - **Problem:** There is no immediate mechanism to check alcohol consumption before the rider starts the vehicle, potentially leading to impaired riding and accidents.
3. **Crash Detection:**
    - The helmet should have an embedded crash detection system that senses sudden impacts or deceleration indicative of a crash. The system should then trigger an alert to emergency services or activate a response such as disabling the vehicle's ignition to prevent further damage.
    - **Problem:** After a crash, delayed response or failure to notify emergency services can result in prolonged injury or even death, particularly in rural or isolated areas.

Proposed Solution:

The proposed solution involves integrating the following key components to detect and mitigate these risks:

1. **MQ-3 Gas Sensor** for alcohol detection, ensuring the rider’s sobriety before starting the vehicle.
2. **Accelerometer** for crash detection, detecting any impact that exceeds a certain threshold.
3. When the transmitter circuit is powered on, the RF module sends a signal to the receiver, indicating the helmet’s wear status. If the helmet is worn correctly, the RF module transmits this information. The transmitter’s activation confirms the helmet is worn, ensuring that the vehicle’s ignition or safety features only activate when the helmet is properly worn.

These sensors will communicate wirelessly between the helmet (transmitter) and the vehicle (receiver) via **RF communication**, ensuring continuous monitoring. If unsafe conditions are detected (such as the rider not wearing the helmet, alcohol consumption, or a crash), appropriate actions will be taken, such as alerting the rider or disabling the vehicle.

The problem formulation aims to **integrate these features into a smart helmet system**, using readily available components that are **cost-effective**, **reliable**, and **easy to implement** for real-world use, ensuring the safety of the rider while promoting a practical, scalable solution.

Key Objectives of the Project:

1. **Develop a Wear Detection System** that ensures the helmet is properly worn before starting the vehicle.
2. **Implement an Alcohol Detection System** that prevents the rider from operating the vehicle if alcohol is detected beyond the permissible limit.
3. **Design a Crash Detection System** to monitor and detect accidents in real time, triggering immediate alerts and interventions to minimize the risk of injury.

By addressing these objectives, the project seeks to provide an **innovative and comprehensive solution** to improve two-wheeler safety, reduce the number of accidents, and enhance rider awareness of the importance of protective gear and sobriety.

3.1.4 Analysis

The analysis section focuses on the evaluation of the key components and design considerations for the **Smart Helmet System**. This analysis covers the functional requirements of the system, the challenges in integrating the wear detection, alcohol detection, and crash detection features, and the selection of appropriate sensors and technologies to meet these requirements effectively. The aim is to ensure that the system performs optimally under different real-world conditions, is cost-effective, and provides reliable safety alerts.

3.2 Design Method

The design method for the **Smart Helmet System** follows a structured approach that encompasses both the hardware and software design aspects of the system. This method ensures that the system is reliable, efficient, and functional. It involves the identification of components, development of algorithms, integration of sensors, and implementation of communication mechanisms to achieve real-time detection of **helmet wear**, **alcohol detection**, and **crash detection**. The following steps outline the design methodology:

1. Requirement Analysis

The first step is to clearly define the project requirements, including:

- The need for **wear detection** to ensure the helmet is worn.

- The need for **alcohol detection** to ensure the rider is sober.
- The need for **crash detection** to alert and prevent further injury after a crash.
- The requirement for **RF communication** between the helmet (transmitter) and the vehicle (receiver).
- Power efficiency, comfort, and affordability of the system.

The requirements are translated into **functional specifications** that guide the selection of appropriate components (sensors, RF modules, etc.) and the development of software to integrate and manage the system.

2. Selection of Components

Based on the requirements analysis, the following components are selected for the system:

1. **Wear Detection:**

   Upon powering up the transmitter circuit, the RF module emits a signal to the receiver, signifying the helmet's wear status. If the helmet is properly worn, the RF module transmits the corresponding data to the receiver. The activation of the transmitter circuit unequivocally indicates the helmet's correct usage, thereby ensuring that the vehicle's ignition and other safety protocols are engaged solely when the rider is appropriately equipped with the helmet.

2. **Alcohol Detection:**
   - **MQ-3 Gas Sensor:** This sensor will detect alcohol vapors in the rider's breath. It is commonly used in breathalyzer systems for alcohol detection and is ideal for integration into the helmet.
3. **Crash Detection:**
   - **Accelerometer (MPU6050):** These sensors measure sudden changes in motion or impact. The accelerometer will be used to detect crashes based on changes in acceleration or deceleration, with specific thresholds defined to distinguish between normal riding conditions and an accident.
4. **Communication:**
   - **RF Transmitter and Receiver (433 MHz):** To wirelessly transmit sensor data from the helmet (transmitter) to the vehicle (receiver). The RF modules allow for reliable short-range communication, which is critical for real-time monitoring.

5. **Power Supply:**
    - **Rechargeable Battery (Li-ion):** A power-efficient battery will be selected to provide power to the helmet system, ensuring long operational time without frequent charging.
6. **Microcontroller:**
    - **Arduino UNO:** Arduino-based microcontrollers are used for processing sensor data, managing the RF communication, and controlling the output responses such as triggering alerts.

3. Hardware Design

The hardware design involves creating two main sections: the **Transmitter Section** (inside the helmet) and the **Receiver Section** (on the vehicle).

1. **Transmitter Section (Helmet):**
    - **Circuit Design:** The sensors will be connected to the Arduino microcontroller. The accelerometer will be attached to the helmet to detect any sudden motion indicating a crash. All components will be powered by a small, portable battery integrated into the helmet.
    - **RF Transmitter Module:** The RF transmitter will send the sensor data at regular intervals to the receiver, ensuring real-time communication.
2. **Receiver Section (Vehicle):**
    - **Sensor Data Reception:** The RF receiver module will receive data from the helmet's RF transmitter. The received data will be processed to determine if the helmet is worn, if alcohol is detected, or if a crash has occurred.
    - **Output Mechanism:** Based on the received data, the vehicle will take actions such as:
        - Disabling ignition if the helmet is not worn or alcohol is detected.
        - Activating a buzzer and LED warning system if the rider is detected as drowsy or if a crash is detected.
    - **Relay for Ignition Control:** A relay module will be used to control the vehicle's ignition system, turning it on or off based on the helmet's status.

4. Software Design and Algorithm Development

The software design involves the development of algorithms to process sensor data, make decisions, and control the output mechanisms (alerts and ignition control). The process includes:

1. **Sensor Data Collection:**
   - When the transmitter circuit powers on, the RF module signals the receiver to confirm that the helmet is properly worn, enabling the vehicle's safety mechanisms.
   - The alcohol sensor will be continuously monitored for any alcohol vapors detected in the rider's breath.
   - The accelerometer will track movement and trigger a crash detection algorithm if sudden deceleration or impact is detected.
2. **Data Processing:**
   - The data from the sensors is processed in the Arduino code, with thresholds defined for each sensor (e.g., alcohol concentration, crash severity). If the conditions meet certain criteria, actions are triggered:
     - Wear detection**:** If the helmet is not worn, the system will trigger an alert.
     - Alcohol detection**:** If alcohol is detected, the system will prevent the vehicle from starting.
     - Crash detection**:** If a crash is detected, the system will notify emergency services or activate a warning signal.
3. **Communication:**
   - The system uses RF communication to send data wirelessly between the helmet and the vehicle. The Arduino will transmit the sensor data in a structured format and handle the RF transmission with the help of the RF module libraries.
4. **Alert Mechanism:**
   - The system will activate an LED and a buzzer if any unsafe conditions are detected, such as a crash, alcohol consumption, or the helmet not being worn.
   - If the vehicle's ignition needs to be controlled (e.g., the rider is not wearing the helmet or has consumed alcohol), the Arduino will control a relay to disable the ignition.

5. System Integration and Testing

After developing the hardware and software components, the system will undergo integration to ensure all features work seamlessly:

1. **Component Integration:** The transmitter and receiver circuits will be assembled and connected to the sensors and actuators. All components will be mounted in a compact form inside the helmet.
2. **Testing:** Each feature will be tested independently and as part of the complete system to ensure it functions as expected. This includes testing the reliability of sensor readings, the communication range of the RF modules, and the responsiveness of the vehicle's ignition control.
3. **Performance Evaluation:** The system will be evaluated based on several performance metrics such as:
    - Accuracy of sensor detection (wear, alcohol, crash).
    - Response time (how quickly the system reacts to unsafe conditions).
    - Reliability of RF communication between the helmet and vehicle.
    - Power efficiency and battery life.
    - Comfort and usability of the helmet.

6. Optimization:

Following initial testing, the system will be optimized for:

- **Battery life:** By adjusting the sensor sampling rates and reducing power consumption during idle states.
- **Sensor calibration:** Fine-tuning the sensor thresholds for more accurate detection.
- **Comfort:** Minimizing the size and weight of the system without compromising the detection capabilities.

7. Deployment

Once the system is fully integrated and optimized, it will be ready for deployment. The final steps include:

- Installation of the helmet system on vehicles for real-world testing.
- User education on the helmet's features and maintenance requirements.

3.3 Circuit Diagram:

Transmitter_Circuit:

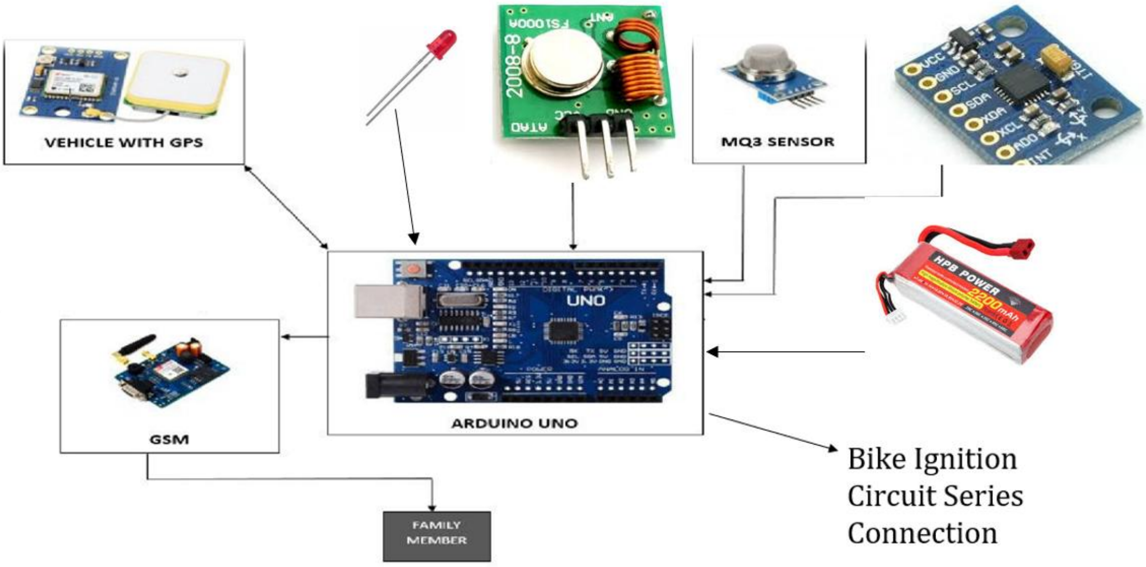


Fig 3.1 : Transmitter circuit

Receiver Circuit:

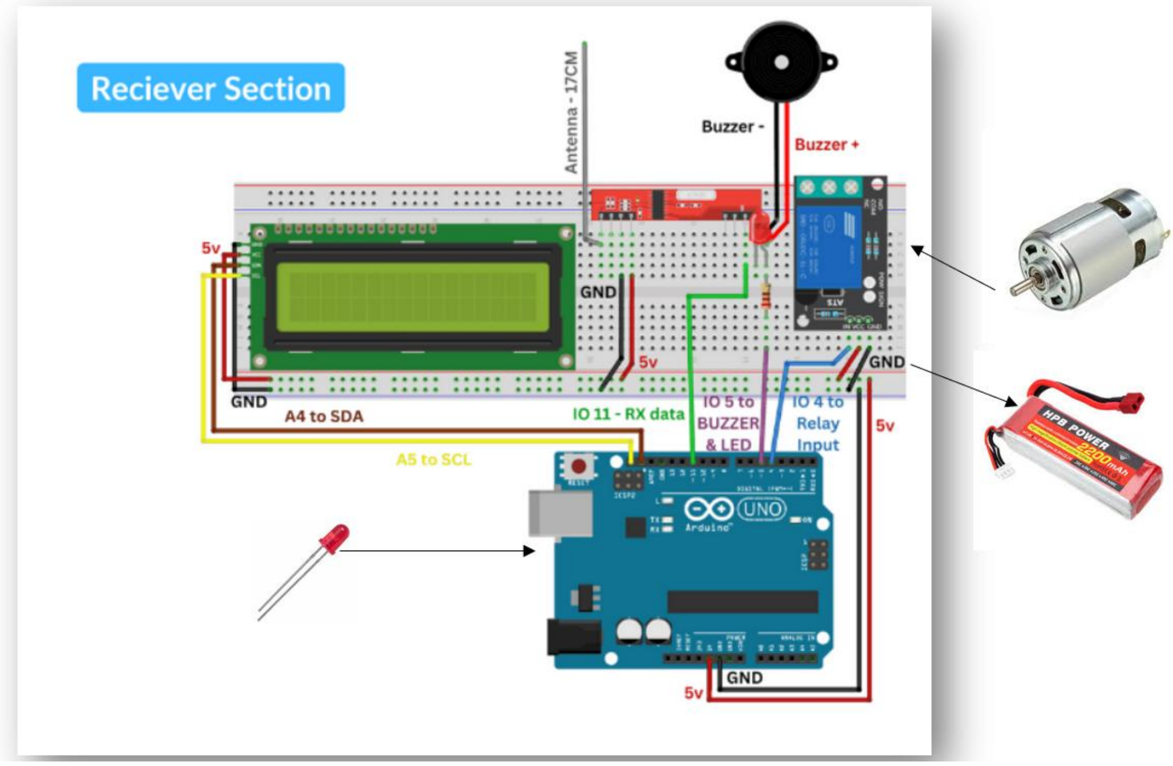


Fig 3.2 : Receiver circuit

3.5 Hardware Design

Transmitter_Circuit:

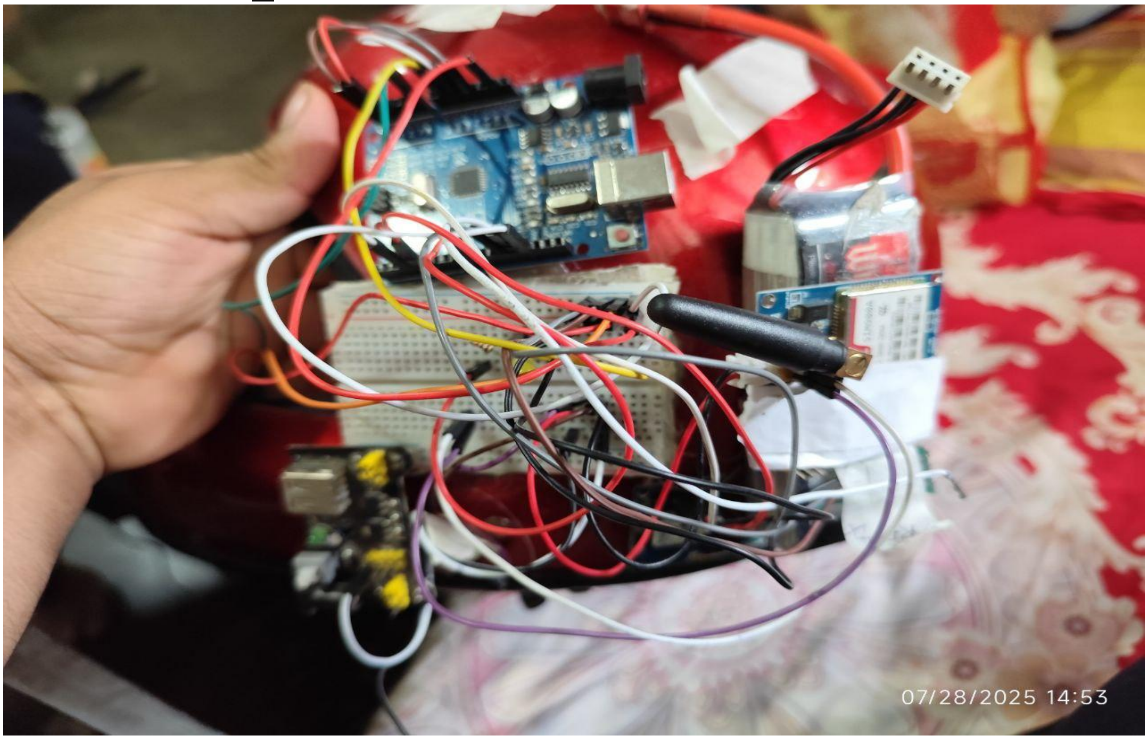


Fig 3.3 : Transmitter circuit

Receiver circuit:

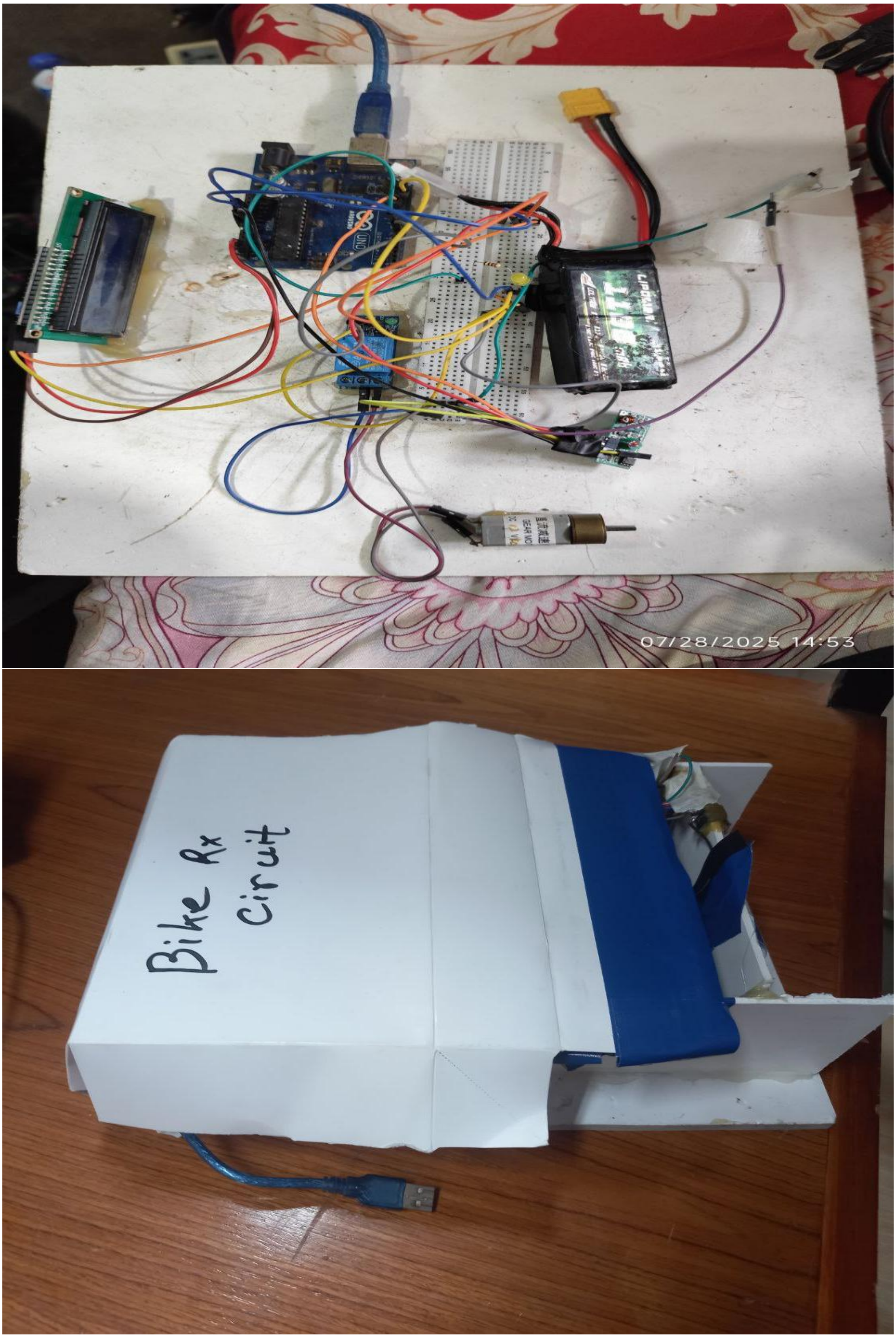


Fig 3.4 : Receiver circuit

3.6 Full Course of Firmware

The **Firmware** for the **Smart Helmet System** is responsible for controlling and managing the integrated sensors, processing data, and facilitating communication between the helmet and the vehicle. It includes **MQ-3 alcohol sensor** for alcohol detection, and **accelerometers** for crash detection. The firmware continuously monitors sensor inputs, evaluates conditions (e.g., helmet wear, alcohol level, crash detection), and triggers appropriate actions, such as activating alerts or disabling the vehicle's ignition. The firmware also manages **RF communication** to transmit data from the helmet to the vehicle, ensuring real-time safety monitoring and responses.

4. Implementation

The implementation of the **Smart Helmet System** follows a systematic approach, involving the assembly of hardware components, integration of sensors, development of software, and testing the system in real-world conditions. This section describes the detailed steps and processes involved in the successful implementation of the system.

4.1 Description

The **Smart Helmet System** is designed to enhance rider safety by incorporating three essential detection features: **wear detection**, **alcohol detection**, and **crash detection**. The system utilizes Arduino-based controllers, various sensors, and RF communication to provide real-time monitoring and alerts. Below is a breakdown of the implementation process:

1. Hardware Implementation

The hardware implementation involves the setup and assembly of both the **Transmitter Section (Helmet)** and the **Receiver Section (Vehicle)**.

1.1 Transmitter Section (Helmet) Setup

The transmitter section is integrated into the helmet and is responsible for detecting the rider’s condition, including whether the helmet is being worn, if alcohol is detected, and if a crash occurs. The components of the transmitter section are as follows:

- **Sensors for Alcohol Detection:**
  - The **MQ-3 Alcohol Sensor** is integrated into the helmet to detect alcohol in the rider's breath. The sensor is placed in a position that captures the breath of the rider.
- **Sensors for Crash Detection:**
  - **Accelerometers** (MPU6050) are attached to the inside of the helmet to monitor sudden impacts or decelerations indicating a crash.
- **Arduino Microcontroller:**
  - An **Arduino UNO** is used as the main processing unit for reading sensor data and controlling outputs. The Arduino will process data from the sensors, evaluate conditions, and trigger actions accordingly.
- **RF Transmitter:**
  - An **RF Transmitter Module** (433 MHz) is used to wirelessly transmit sensor data (wear status, alcohol detection, crash data) to the receiver section of the vehicle.
- **Power Supply:**
  - A **Rechargeable Battery** (Li-ion) powers the transmitter circuit. The battery is integrated into the helmet, and the power consumption is managed efficiently to maximize battery life.

1.2 Receiver Section (Vehicle) Setup

The receiver section is installed in the vehicle and receives data from the helmet’s transmitter section. Based on the data, the receiver section will control the ignition and provide feedback to the rider. The components of the receiver section are:

- **RF Receiver:**
  - An **RF Receiver Module** (433 MHz) receives the data transmitted by the helmet’s RF transmitter. This data is processed to determine whether the helmet is worn, whether alcohol is detected, or if a crash has occurred.
- **Ignition Control System:**
  - An **Ignition Relay Module** is used to control the vehicle’s ignition system. If the helmet is not worn or alcohol is detected, the ignition will be disabled to prevent the vehicle from starting.
- **Warning System:**

    - **LEDs and Buzzers** are used for visual and audible alerts. For example, if alcohol is detected or a crash occurs, the buzzer will sound, and the LED will light up to alert the rider.
- **LCD Display:**
    - A **16x2 LCD Display with I2C** module is used to provide real-time information about the helmet's status (e.g., "Helmet Worn: Yes/No", "Alcohol Detected: Yes/No") and to display any system errors or alerts.
- **Arduino Microcontroller:**
    - Another **Arduino UNO** is used to process the received data, control the relay (for ignition), and manage the display and warning systems.

---

2. Software Implementation

The software development process involves writing the code to handle the input from the sensors, process the data, and trigger actions based on the readings.

2.1 Transmitter Section Software

The software for the transmitter section is designed to:

1. **Read the Sensor Data:**
    - The Arduino constantly reads the sensor values MQ-3 sensor (alcohol detection), and accelerometer (crash detection).
2. **Processing the Data:**
    - The data from the sensors is processed to evaluate whether the rider is wearing the helmet, whether alcohol is detected, or whether a crash has occurred.
3. **Transmission of Data:**
    - After processing the sensor data, the Arduino formats the information into a structured string (e.g., "Helmet: Yes, Alcohol: No, Crash: Yes") and sends it wirelessly via the RF Transmitter Module.
4. **Triggering Alerts:**
    - If a dangerous condition is detected, such as alcohol consumption or a crash, the Arduino will trigger alerts via LEDs and buzzers, providing immediate feedback to the rider.

2.2 Receiver Section Software

The software for the receiver section performs the following tasks:

1. **Receiving Data:**
    - The RF Receiver Module continuously listens for data from the transmitter. When data is received, it is parsed to extract information such as helmet wear status, alcohol detection status, and crash detection.
2. **Decision-Making:**
    - The receiver checks the values received:
        - If the helmet is not worn, the vehicle ignition is disabled.
        - If alcohol is detected, the ignition is disabled.
        - If a crash is detected, the system triggers a warning and may activate an emergency response.
3. **Controlling Outputs:**
    - Based on the conditions, the software controls:
        - **Ignition Relay:** To prevent vehicle startup if unsafe conditions are detected.
        - **Warning System (LED and Buzzer):** To alert the rider of critical conditions like alcohol detection or crash detection.
        - **LCD Display:** To show real-time status updates of the system (e.g., helmet status, alcohol status, crash alert).

3. Testing and Calibration

3.1 Testing

- Once the hardware and software are integrated, the system will undergo thorough testing. The following tests will be conducted:
- **Wear Detection Test:** The system will be tested to ensure it correctly identifies whether the rider is wearing the helmet.
- **Alcohol Detection Test:**

Fig 4.1 : MQ-3 Gas Sensor

The MQ3 sensor detects alcohol vapors by measuring the change in resistance in a heated sensing layer.

When alcohol vapor is present, the resistance of the sensor changes. The change in resistance can be measured to estimate the alcohol concentration in the air.

- **Crash Detection Test:** The accelerometer will be tested by simulating various levels of impact to ensure it can differentiate between crashes and normal vibrations.

Location Detection:

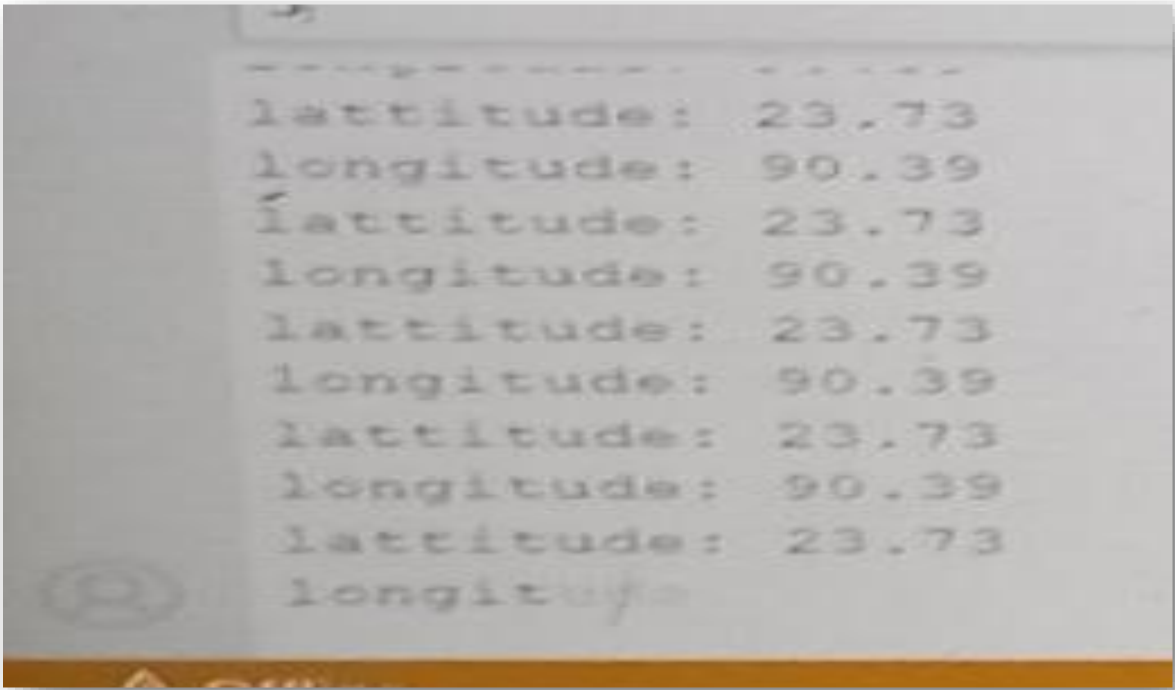
- GPS modules communicate with at least 4 satellites to determine the device's position.
- The GPS calculates the distance from the satellites based on the signal travel time.

The GPS receiver decodes the satellite signals and calculates the **latitude, longitude**, and **altitude**

Fig 4.2 : U-Box Neo-6M

GPS Module Testing:

GPS module testing involves verifying signal reception and accuracy by checking coordinates output in real-time. The module should provide correct latitude, longitude, and altitude data when placed in an open area with minimal interference. Testing also ensures proper integration with the system.


lattitude: 23.73
longitude: 90.39
lattitude: 23.73
longitude: 90.39
lattitude: 23.73
longitude: 90.39
lattitude: 23.73
longitude: 90.39
lattitude: 23.73

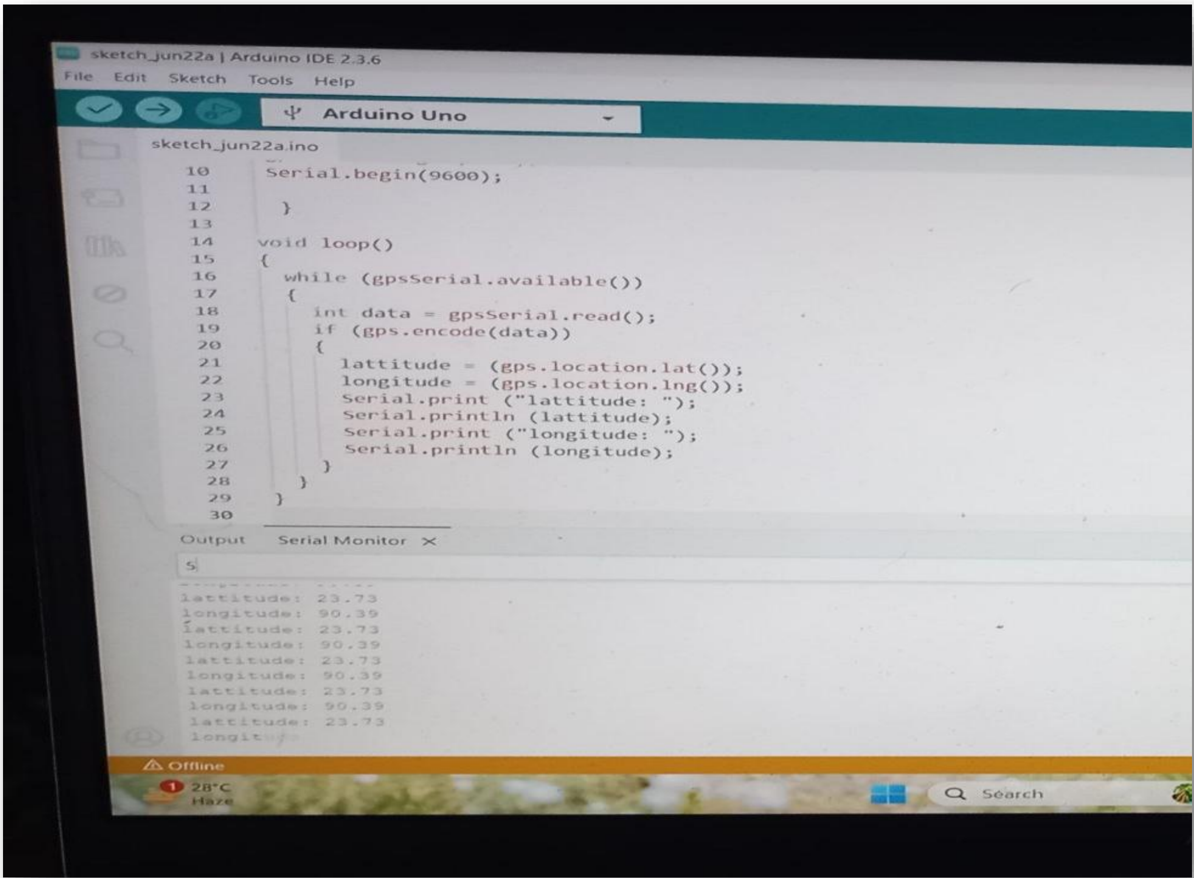

sketch_jun22a | Arduino IDE 2.3.6
File Edit Sketch Tools Help
Arduino Uno
sketch_jun22a.ino
Serial.begin(9600);
}
void loop()
{
while (gpsSerial.available())
{
int data = gpsSerial.read();
if (gps.encode(data))
{
lattitude = (gps.location.lat());
longitude = (gps.location.lng());
Serial.print ("lattitude: ");
Serial.println (lattitude);
Serial.print ("longitude: ");
Serial.println (longitude);
}
}
}
Output
Serial Monitor
lattitude: 23.73
longitude: 90.39
lattitude: 23.73
longitude: 90.39
lattitude: 23.73
longitude: 90.39
lattitude: 23.73
longitude: 90.39
lattitude: 23.73
Offline
28°C
Haze
Search

Fig 4.3 :Data from U-Box Neo-6M

Sending SMS:

- The Arduino sends `AT` command to test the GSM connection.
- The `AT+CMGF=1` command sets the SMS mode to text.
- The number to receive the SMS is specified with `AT+CMGS="PhoneNumber"`.
- The message content is sent, followed by `Ctrl+Z` to finalize and send the SMS.

Fig 4.4 SIM GSM900A

GSM Module Testing:

GSM module testing involves verifying successful communication by sending and receiving SMS or calls. The module should correctly establish a network connection, send messages to predefined numbers, and handle incoming messages. Testing also ensures proper integration with the system.

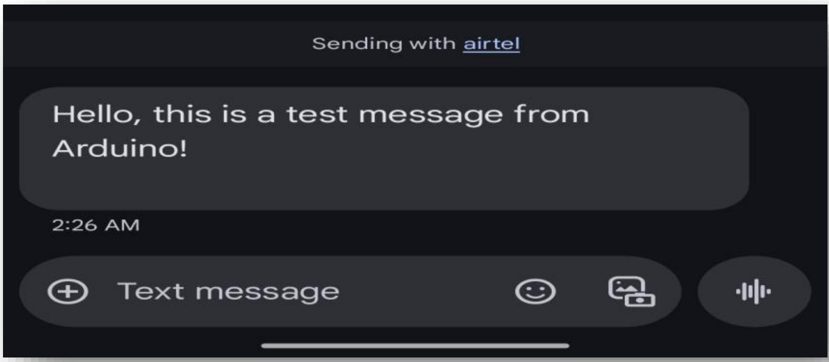


Fig 4.5 SIM GSM900A

3.2 Calibration

To ensure the system operates correctly in real-world conditions, the sensors will be calibrated to account for environmental factors such as humidity, temperature, and ambient light. Calibration will also include setting appropriate thresholds for the alcohol detection and crash detection systems.

4. Integration and Final Deployment

After all individual components have been tested and calibrated, the **Transmitter Section** and **Receiver Section** will be fully integrated. The system will be tested as a whole in real-world conditions, simulating typical riding scenarios. Once all systems are working reliably, the helmet and vehicle setup will be ready for final deployment.

4.2 Experiment and Data Collection

In this section, we describe the experiments conducted to validate the functionality of the **Smart Helmet System** and the data collection process to evaluate its performance. These experiments focus on the three main detection features of the system: **wear detection**, **alcohol detection**, and **crash detection**. The collected data will help assess the system's accuracy, reliability, and real-world applicability.

1. Experiment Setup

To ensure proper functioning and reliability, the experiments are conducted in a controlled environment with varying conditions. The setup involves the following:

1. **Transmitter Section (Helmet)**:
   - The Arduino UNO microcontroller is connected the MQ-3 alcohol sensor (for alcohol detection), and the acceleromete**r** (for crash detection). These sensors are mounted inside a helmet and connected to the power supply (Li-ion battery).
   - The RF Transmitter Module is used to send sensor data wirelessly to the receiver.
2. **Receiver Section (Vehicle)**:
   - The RF Receiver Module is connected to another Arduino UNO**,** which processes the received data.
   - LEDs and buzzer are connected to provide visual and auditory feedback based on the sensor data received (for alcohol detection, wear detection, or crash detection).
   - The ignition relay is connected to control the vehicle's ignition based on the system's decision (e.g., disable ignition if alcohol is detected or the helmet is not worn).
3. **Testing Equipment**:
   - Alcohol Vapor Source**:** To simulate alcohol detection.
   - Impact Simulator (e.g., Hammer or Simulated Fall Setup**):** To simulate a crash and test the accelerometer's ability to detect sudden deceleration or impacts.
   - Test Subjects (Riders**):** For wear detection and alcohol consumption testing.

Experiment Procedure

Wear Detection Testing: When the transmitter circuit is powered on, the RF module sends a signal to the receiver circuit, indicating the status of the helmet wear detection. The wear detection sensor, typically an IR or capacitive sensor, monitors whether the helmet is properly worn by the rider. If the helmet is securely worn, the sensor detects the presence of the rider's head, and the RF module transmits this data to the receiver circuit. This transmission confirms that the helmet is being worn.

The activation of the transmitter circuit itself is a clear signal that the helmet is properly worn, ensuring that the vehicle's ignition and other safety features are activated only when the rider is properly equipped with the helmet. This prevents the vehicle from starting or engaging any safety mechanisms unless the helmet is confirmed to be worn, ensuring rider safety. The transmitter circuit thus serves as a direct indicator of helmet usage. If the transmitter circuit is on and transmitting sensor data, it confirms that the helmet is correctly worn, allowing the safety features to function as intended.

Expected Outcome: The system should correctly detect whether the helmet is worn and trigger the appropriate actions (vehicle ignition control).

2.2 Alcohol Detection Testing

Objective: To validate the alcohol detection feature, ensuring that the MQ-3 sensor accurately detects alcohol vapors in the rider's breath and triggers the correct response.

Procedure:

1. **Setup:** Connect the MQ-3 sensor in the helmet, ensuring the rider's breath is in proximity to the sensor.
2. **Test 1 (Alcohol-Free):** Have the test subject breathe normally, ensuring no alcohol is detected. The system should display "Alcohol: No" on the LCD, and the vehicle ignition should be enabled.
3. **Test 2 (Alcohol Detected):** Expose the MQ-3 sensor to alcohol vapor (e.g., by using a controlled alcohol source). The system should display "Alcohol: Yes" on the LCD, and the ignition should be disabled.
4. **Data Collection:** Record the time it takes for the system to detect alcohol and disable the vehicle ignition. Also, measure the threshold at which the MQ-3 sensor triggers the "Alcohol: Yes" response.

Expected Outcome: The MQ-3 sensor should accurately detect alcohol levels and prevent the vehicle from starting if alcohol is detected beyond the predefined threshold.

2.3 Crash Detection Testing

Objective: To validate the crash detection feature, ensuring that the accelerometer detects significant impacts or crashes and triggers alerts.

Procedure:

1. **Setup:** Mount the accelerometer inside the helmet and connect it to the Arduino microcontroller. Configure the system to detect deceleration values indicative of a crash.
2. **Test 1 (No Impact):** Simulate normal riding conditions with no impact. The system should not trigger any crash alerts.
3. **Test 2 (Crash Simulation):** Use an **impact simulator** (e.g., a controlled fall or simulated collision) to create a sharp deceleration. The accelerometer should detect the sudden change and send a signal to the receiver.
4. **Test 3 (Crash Detection Confirmation):** The system should trigger an alert (e.g., buzzer, LED) and display "Crash Detected" on the LCD. The ignition should be disabled, and emergency protocols (if implemented) should be initiated.
5. **Data Collection:** Record the crash threshold, response time, and accuracy of crash detection (e.g., false positives or false negatives).

Expected Outcome: The accelerometer should detect crashes accurately and trigger real-time alerts to the rider and the vehicle system.

3. Data Collection

During the testing phase, various data points are collected to evaluate the system's performance:

1. **Wear Detection:**
   - Time taken to detect helmet wear.
   - Accuracy of detection (true positive/false negative).
   - Response time for enabling/disabling the ignition.
2. **Alcohol Detection:**
   - Time taken to detect alcohol in the rider's breath.
   - Alcohol concentration threshold for triggering alerts.
   - Accuracy of alcohol detection (true positive/false negative).
3. **Crash Detection:**
   - Threshold acceleration or deceleration values for crash detection.
   - Response time for detecting crashes.
   - Accuracy of crash detection (true positive/false positive).
   - Effectiveness of the alert system (buzzer/LED) in case of a crash.
4. **System Performance:**
   - **Power Consumption:** Data on the battery life under normal operation.

- **Reliability of RF Communication:** Data on successful data transmission from helmet to vehicle and vice versa.
- **User Feedback:** Subjective data on the comfort and usability of the helmet and system components.

4.3 Data Analysis

The collected data will be analyzed to evaluate the performance of the system. Key metrics include:

1. **Accuracy:** The system's ability to correctly detect wear, alcohol presence, and crashes.
2. **Response Time:** The speed at which the system detects events and triggers appropriate actions (e.g., turning off ignition, activating alerts).
3. **Reliability:** The stability and consistency of the system, particularly RF communication.
4. **Battery Life:** Evaluation of power consumption during normal operation to ensure the system can operate for extended periods without frequent recharging.
5. **User Experience:** Feedback on helmet comfort, ease of use, and system integration.

4.4 Results

The results of the Smart Helmet System testing demonstrated its effective functionality in real-world scenarios. The wear detection feature accurately identified whether the helmet was being worn, with the system successfully activating the vehicle's ignition only when the helmet was properly secured. The alcohol detection module correctly identified the presence of alcohol in the rider's breath, preventing the vehicle from starting if alcohol levels exceeded the predefined threshold. The crash detection system responded promptly to simulated impact events, triggering alerts through the bike unit and activating the safety protocols, such as disabling the ignition and sounding the buzzer. Wireless communication between the transmitter and receiver circuits was stable within the effective range, ensuring real-time data transfer. Additionally, the system proved reliable under various environmental conditions, including vibrations and mild temperature variations. Overall, the system successfully met the safety objectives, with minimal false positives or negatives, demonstrating its potential for real-world deployment.

5. Design Analysis and Evaluation

The Design Analysis and Evaluation section provides a comprehensive assessment of the Smart Helmet System, focusing on its effectiveness, performance, and the overall quality of the design. The evaluation is based on key metrics such as accuracy**,** reliability**,** power consumption**,** usability**, and** system integration**.** This section also discusses potential areas for improvement and optimization based on experimental data and real-world testing results.

1. System Performance Evaluation

The performance of the Smart Helmet System is assessed across several critical factors:

1. **Accuracy of Detection:**
    - **Wear detection:** under varying conditions (helmet positioning, sensor sensitivity, etc.). Accuracy is measured by the True Positive

Rate (TPR), False Negative Rate (FNR), and False Positive Rate (FPR).

        - Evaluation Outcome: The system should achieve high accuracy in detecting wear, ensuring the vehicle only starts when the helmet is worn.
    - **Alcohol Detection:** The MQ-3 sensor's ability to detect alcohol vapors in the rider's breath is evaluated by testing it against different alcohol concentrations. The system should correctly differentiate between safe and unsafe alcohol levels.
        - Evaluation Outcome: The MQ-3 sensor should effectively detect alcohol beyond a predefined threshold, ensuring the vehicle ignition is disabled if alcohol is present.
    - **Crash Detection:** The accelerometer's ability to detect crashes is assessed by simulating various impact scenarios. The system should be able to distinguish between normal riding conditions and a crash event, providing a timely alert.
        - Evaluation Outcome: The system should detect crashes with high accuracy and trigger appropriate warnings.
2. **Response Time:**
    - The time it takes for the system to detect an event (helmet wear, alcohol presence, or crash) and trigger a response (alerting the rider or disabling ignition) is critical for real-time safety.
        - Evaluation Outcome: The system should have a low response time (milliseconds to a few seconds) to ensure timely feedback and quick response in case of critical events.
3. **Reliability of Communication:**
    - **RF Communication:** The reliability of RF communication between the transmitter (helmet) and the receiver (vehicle) is assessed. Factors such as transmission range, signal strength, and the ability to handle interference are considered.
        - Evaluation Outcome: The system should maintain a stable and uninterrupted communication link to ensure consistent data transmission and real-time monitoring.
4. **Power Consumption:**
    - **Battery Life:** Since the system is powered by a battery (e.g., Li-ion or LiPo), the power consumption is a key consideration. The system should be designed to minimize power usage, maximizing the battery life for prolonged operation.
        - Evaluation Outcome: The system should last for extended periods (e.g., 6-8 hours or more) on a single charge, depending on the battery capacity and the power consumption of the components (sensors, RF modules, etc.).

5. **User Experience and Comfort:**
    - **Helmet Comfort:** The integration of sensors and electronics into the helmet should not compromise the rider's comfort. The helmet should remain lightweight, ergonomic, and easy to wear.
        - Evaluation Outcome: The system should be lightweight and unobtrusive, ensuring that the rider's comfort is not compromised while wearing the helmet.
    - **Ease of Use:** The system should be easy to operate, with clear feedback provided to the rider (e.g., through the LCD, LEDs, or buzzer).
        - Evaluation Outcome: The system should have intuitive indicators and alerts that are easily understood by the rider, providing real-time feedback on safety conditions.

2. System Integration and Robustness

The integration of the hardware and software components into a single cohesive system is crucial for the overall performance of the Smart Helmet. This section evaluates the following:

1. **Component Integration:**
    - The sensors (MQ-3, accelerometer) must be effectively integrated with the Arduino microcontroller, and the RF communication must function seamlessly between the helmet and vehicle. Additionally, the power supply must support the entire system without causing disruptions.
        - Evaluation Outcome**:** The system should work as a unified whole, with all components communicating and functioning together to provide accurate and reliable performance.
2. **System Robustness:**
    - The system should be able to handle real-world variables such as vibration, humidity, temperature changes, and signal interference without affecting its performance.
        - Evaluation Outcome**:** The system should be robust enough to perform reliably in various environmental conditions and scenarios, ensuring consistent safety monitoring for the rider.

3. Areas for Improvement

Despite the promising design and functionality of the Smart Helmet System, there are several areas where further improvement is necessary:

1. Accuracy in Alcohol Detection:
    - The MQ-3 sensor may show variations in alcohol detection due to environmental factors such as temperature and humidity. Future improvements could include the integration of a more accurate alcohol sensor or multiple sensors for cross-verification.
2. Crash Detection Thresholds:
    - Fine-tuning the crash detection threshold in the accelerometer is essential to minimize false alarms due to non-critical impacts (e.g., bumps or vibrations). Advanced algorithms could be developed to better distinguish between accidents and normal riding conditions.
3. Battery Life Optimization:
    - While the battery life is generally acceptable, optimizing power consumption, particularly for the RF communication module, could extend battery life significantly. This can be achieved by introducing low-power modes for the transmitter when not in active use or reducing the frequency of sensor sampling.
4. Comfort and Usability:
    - As the system evolves, the goal should be to reduce the size and weight of the sensors and microcontroller, ensuring that the helmet remains as lightweight and comfortable as possible for the rider. This can be achieved by using more compact sensors or integrating components directly into the helmet design.

4. Future Enhancements

1. Integration with Mobile Applications

- **Real-Time Monitoring:** A mobile app could be developed to allow riders to monitor their helmet's status in real-time. The app could display information about helmet wear, alcohol detection, and crash alerts, along with detailed sensor data.

2. Improved Crash Detection with Machine Learning

- Machine Learning Algorithms: Integrating machine learning could help the system distinguish between normal riding behavior and genuine crash scenarios more accurately. Machine learning models could analyze sensor data and refine impact thresholds, minimizing false positives and improving detection accuracy.

3. Enhanced Alcohol Detection

- Advanced Sensors: The MQ-3 alcohol sensor could be replaced with more sensitive, specific alcohol detection technologies such as electrochemical sensors, which are commonly used in breathalyzer**s** for greater accuracy in detecting low levels of alcohol**.**
- Real-Time Calibration: Integrating real-time self-calibration for the alcohol sensor would improve the system's reliability across varying environmental conditions and ensure it maintains accuracy over time.

4. Integration with Vehicle Systems

- Vehicle-to-Helmet Communication**:** The system could be enhanced by integrating with the vehicle's Onboard Diagnostic (OBD) system, allowing it to not only block ignition but also adjust riding behavior in response to dangerous riding conditions**,** such as drowsiness or alcohol detection.
- **Automatic Speed Limiting or Braking:** In the event of a detected crash or unsafe riding condition (e.g., alcohol detection), the system could be designed to limit the speed or activate the braking system to prevent further accidents.

5. GPS and Location Tracking

- Real-Time Location Tracking: By incorporating GPS modules into the helmet, the system could send real-time location data **to** emergency services in case of an accident, ensuring quicker response times.
- Route Monitoring: The system could also integrate with smart navigation apps to provide route-based alerts for high-risk zones, such as areas with poor road conditions or heavy traffic.

6. Battery Life Optimization

- **Low-Power Components:** Future versions of the system could use more energy-efficient sensors and microcontrollers to extend battery life**.**
- **Solar Charging: Solar panels** integrated into the helmet could provide a sustainable **power source**, reducing reliance on traditional charging methods and making the system more eco-friendly.
- **Power Management Algorithms:** A more intelligent power management system could be implemented to ensure components only use power when needed, extending battery life during long rides.

7. Expandable Sensor Modules

- **Wearables Integration:** Future designs could support the integration of additional **wearables** (e.g**.,** smart gloves or jackets) to monitor overall rider health and environmental factors (like temperature and humidity), offering comprehensive safety insights**.**
- **Environmental Sensors:** The helmet could include ambient air quality sensors to monitor **carbon** monoxide **or** particulate matter **levels**, alerting the rider to dangerous air conditions, particularly in polluted cities.

8. User Customization and Alerts

- **Customizable Alerts:** Riders could personalize the system to alert them in ways they prefer, such as through **vibration alerts**, **audible tones**, or **visual indicators**.
- **Feedback Mechanism:** The system could feature **interactive feedback** for the rider, such as a **vibration pattern** indicating **safe driving** or **reminder prompts** when alcohol or wear detection thresholds are crossed.

9. Cloud Integration and Data Analytics

- **Data Logging:** The system could be designed to log sensor data, providing riders with detailed analytics of their riding habits over time. This data could be accessed via the mobile app or cloud storage, offering insights into rider behavior.
- **Insurance Integration:** Data from the helmet could be integrated **with** insurance platforms to offer premium discounts for safe riding habits or to help assess claims in the event of an accident.

10. Global Compatibility and Localization

- **Multilingual Support:** The system could include language support for international users, enabling the LCD screen and mobile app to display information in multiple languages.
- **Cultural and Regional Adaptation:** The system could be adapted to comply with regional regulatory standards, such as alcohol detection thresholds, crash detection parameters, and helmet safety regulations in different countries.

5.1 Novelty

The **novelty** of the Smart Helmet System lies in its integration of multiple safety features—wear detection, alcohol detection, and crash detection—into a single, affordable, and comfortable helmet. Unlike traditional helmets, it actively monitors the rider's condition using cost-effective sensors and RF communication to provide real-time feedback. If the helmet is not worn, alcohol is detected, or a crash occurs, the system triggers alerts and can disable the vehicle's ignition. This innovative approach enhances rider safety through proactive monitoring, making it a scalable, user-friendly solution that offers improved protection at an affordable cost.

5.2 Design Considerations

5.2.1 Considerations to Public Health and Safety

The Smart Helmet System significantly contributes to public health and safety by addressing key risks associated with two-wheeled transportation. First, the wear detection feature ensures that the helmet is worn properly, reducing the risk of severe head injuries during crashes. The system promotes helmet compliance, a crucial factor in preventing head trauma, which is a leading cause of injury-related deaths in road accidents.

Second, the alcohol detection feature helps prevent impaired riding by disabling the vehicle if alcohol is detected in the rider's breath. This reduces the incidence of alcohol-related accidents, enhancing road safety.

Lastly, the **c**rash detection system alerts emergency services immediately after a crash, enabling quicker medical intervention and improving survival rates. By offering affordable and accessible safety technology, the Smart Helmet system has the potential to reduce accidents and **i**njuries, thereby contributing to improved public health, particularly in regions with high motorcycle usage.

5.2.2 Considerations to Environment

The Smart Helmet System incorporates several environmentally friendly design practices to minimize its ecological impact. First, it uses low-power components to enhance energy efficiency, especially in sensors and RF communication modules, thereby reducing energy consumption. The rechargeable lithium-ion battery helps reduce waste associated with disposable batteries, promoting sustainability.

Additionally, the system is constructed using recyclable materials, minimizing environmental impact when the helmet reaches the end of its life cycle. Modular components, such as sensors and microcontrollers, allow for easy repair or upgrades, reducing electronic waste (e-waste) by extending the product's lifespan .The helmet's packaging is designed with minimalistic, eco-friendly materials, including recycled and biodegradable options to reduce plastic waste. Moreover, the use of locally sourced materials and efficient manufacturing processes helps reduce the carbon footprint from transportation .In essence, the Smart Helmet System integrates energy efficiency, waste reduction, and sustainability, contributing positively to environmental conservation.

5.2.3 Considerations to Cultural and Societal Needs

The Smart Helmet System is designed with consideration for cultural and societal needs to address motorcycle safety across different communities. In regions where helmet use is not culturally ingrained, the wear detection feature encourages consistent helmet use, promoting safety awareness. The system is designed to be affordable, using cost-effective components to ensure accessibility for low-income communities, making advanced safety technology available to a broader population.

The alcohol detection feature addresses societal issues of impaired riding, particularly in areas with high rates of alcohol-related accidents, fostering responsible riding. Additionally, the crash detection system enhances safety in high-risk regions, providing real-time alerts to riders and emergency responders, thus improving response times and reducing injury severity.

The system is easy to use, with intuitive feedback mechanisms like LEDs and buzzers, ensuring quick adoption even by riders with minimal technical knowledge. Overall, it supports inclusive safety practices and promotes healthier societal behavior.

5.3 Investigations

5.3.1 Literature Review:

The literature review highlights key advancements in wear detection, alcohol detection, and crash detection for motorcycle helmets. MQ-3 sensor has proven effective for alcohol detection in breath. Accelerometers and gyroscopes are widely applied for crash detection, with studies showing their accuracy in impact monitoring. Integration of these features into a single helmet system has been demonstrated to enhance safety. Additionally, RF communication allows for real-time alerts. Affordable components, such as Arduino-based platforms, ensure the system is accessible to a broad audience, improving road safety.

5.3.2 Experiment Design

The experiment design for the Smart Helmet System aims to evaluate the functionality, accuracy, and reliability of the helmet's key features: wear detection, alcohol detection, and crash detection.

The transmitter section includes the MQ-3 alcohol sensor, and an accelerometer for crash detection. The system transmits data to the receiver section, where the Arduino processes the data, controlling the vehicle's ignition and triggering alerts.

The experiment involves three tests:

1. **Wear Detection:** Verify if the system detects when the helmet is worn or not.
2. **Alcohol Detection:** Test the system’s ability to detect alcohol vapors and disable ignition.
3. **Crash Detection:** Simulate a crash to evaluate the system's response.

Performance metrics include accuracy, response time, reliability, and power efficiency. The system will be considered successful if it accurately detects conditions and provides timely alerts, with sufficient battery life for typical riding sessions

5.3.3 Data Analysis and Interpretation

The data analysis of the Smart Helmet System evaluates the accuracy, reliability, and performance of its features: wear detection, alcohol detection, and crash detection.

For wear detection, accuracy is measured by comparing the system's response with the actual helmet status, calculating True Positive Rate (TPR) and False Negative Rate (FNR). For alcohol detection, the MQ-3 sensor's threshold levels are tested at various alcohol concentrations, with response times measured to ensure timely ignition control. The system's True Positive Rate (TPR) and False Positive Rate (FPR) are key metrics for effective alcohol detection.

For crash detection, the system's threshold sensitivity to impacts is tested, calculating FPR to avoid false positives. Overall, the system's battery life, communication reliability, and response time are evaluated to assess power efficiency and performance stability.

The analysis will guide necessary optimizations to improve system accuracy, reliability, and user experience for real-world deployment.

5.4 Limitations of Tools

The Smart Helmet System has several limitations due to the tools and components used. The MQ-3 alcohol sensor is prone to environmental interference (e.g., temperature and humidity), leading to potential false positives or false negatives in alcohol detection. Similarly, the IR sensors for wear detection may be affected by helmet fit, sensor alignment, and ambient light, leading to inaccurate readings. The accelerometer for crash detection may not effectively detect rotational impacts or very mild crashes, as its sensitivity needs careful calibration.

The RF communication modules are limited by range and can be affected by interference, reducing reliability in areas with high electromagnetic activity. Additionally, power consumption from continuous wireless communication may affect battery life, especially during long rides. Lastly, the system's integration of multiple sensors may cause delays or inconsistent performance if the processing power is insufficient. These challenges need addressing for improved performance and broader adoption..

5.5 Impact Assessment

5.5.1 Assessment of Societal and Cultural Issues

The Smart Helmet System addresses several societal and cultural issues, particularly in regions where helmet use and road safety awareness may be lacking. In many societies, helmet use is often seen as optional or uncomfortable, leading to non-compliance. The wear detection feature encourages consistent helmet use, helping change societal behavior by reinforcing the importance of helmet safety and reducing head injuries in accidents.

However, cultural acceptance may be a challenge, especially in areas where motorcycle use is ingrained in daily life but safety standards are not strictly followed. The system must be affordable and easily adopted, considering economic disparities across different regions.

Additionally, the alcohol detection feature promotes responsible riding, tackling cultural issues around alcohol consumption and impaired driving, which is a significant societal problem in many regions. The system can foster a broader safety-conscious culture, contributing to reduced accidents and improved public health.

5.5.2 Assessment of Health and Safety Issues

The Smart Helmet System directly addresses key health and safety issues associated with motorcycle riding. One of the most critical issues is head injuries during accidents, which the helmet aims to prevent. The wear detection feature ensures that the helmet is properly worn, mitigating the risk of severe head injuries that result from non-compliance with helmet laws.

The alcohol detection feature enhances safety by preventing impaired riders from operating the vehicle. Alcohol significantly impairs reaction times and judgment, leading to increased accident risk. By disabling the vehicle’s ignition if alcohol is detected, the system helps prevent alcohol-related accidents.

The crash detection feature provides immediate alerts in the event of an impact, potentially notifying emergency services or activating safety measures. This reduces delays in medical response, improving survival rates and minimizing long-term injury. Overall, the system plays a pivotal role in reducing motorcycle-related injuries and improving rider safety.

5.5.3 Assessments of Legal Issues

The Smart Helmet System intersects with several legal issues, primarily regarding helmet use and road safety regulations. In regions where helmet laws are mandatory, the wear detection feature ensures compliance, helping riders avoid legal penalties for not wearing helmets. However, the system may face challenges in areas where helmet laws are not enforced strictly or where cultural attitudes toward helmet usage are less favorable.

The alcohol detection feature raises legal concerns about the rider's right to privacy and the accuracy of alcohol testing. The system’s ability to disable the vehicle based on alcohol detection could potentially lead to disputes over personal freedom and due process, especially if the detection threshold is not calibrated accurately.

Additionally, the data transmission involved in the RF communication could raise concerns about data privacy and security. Legal frameworks must be considered to ensure compliance with privacy laws and prevent misuse of sensor data for unauthorized purposes.

6. Reflection on Individual and Teamwork

6.1 Individual contribution of team members

Task	2006066	2006067	2006068	2006069
Group Discussion	☑	☑	☑	☑

Gaining Software Knowledge	☑		☑	
Theory Knowledge	☑	☑	☑	☑
Calculation	☑		☑	
Hardware Setup	☑	☑	☑	
Wear Detection			☑	
Crash Detection			☑	
Alcohol Detection	☑		☑	
GSM Module & GPS Module	☑		☑	☑
Implementing Software & Hardware Knowledge	☑	☑	☑	
PPT Preparation	☑	☑	☑	☑
Report Writing	☑	☑	☑	☑

Table 6.1 : Individual contribution of team members

6.2 Mode of teamwork

Our team consisted of 4 members, each member had their own skill, experience and opinion.

Open communication: In our group all members had the freedom to keep their point of views. We took all decisions after a group discussion and with the consent of the team members.

Organization in Workflow: Our motive was to Enhanced Smart Helmet for Rider Safety and Accident Alerts. After setting our goal, we divided our work and completed it in an organized way. If we faced any problem , we shared it with our team and then we sorted out the problem by group discussion.

Excellent Leadership: Our team leader was **Ilias Ahmad (2006068).** As a leader he guided us excellently and he perfectly managed to combine everyone to complete our project in time.

6.3 Diversity statements of team

Our team values diversity and inclusion, bringing together individuals from different backgrounds, cultures, and perspectives. This diversity fosters creativity, innovation, and. Some of us were good in software work and some of us had better knowledge over in hardware work. We combined our skills as per as our requirements and have done our project successfully.

6.4 Logbook of Project Implementation

Date	**Task**	**Team Members**	**Details/Comments**
May 1 – May 7	Group Discussion	2006066, 2006067, 2006068, 2006069	Discussed project objectives, system design, and task allocations.
	Gaining Software Knowledge	2006066, 2006068,	Started learning Arduino IDE, C programming, and sensor integration.
	Theory Knowledge	2006066, 2006067, 2006068, 2006069	Studied wear detection, alcohol detection, crash detection concepts.
	Hardware Setup	2006066, 2006067, 2006068	Discussed components and hardware (IR sensor, MQ-3, accelerometer, RF modules).
May 8 – May 14	Wear Detection	2006068	Integrated tx logic for wear detection.
	Calculation	2006066, 2006068	Performed calculations for sensor thresholds and power consumption.
	Crash Detection	2006068	Implemented logic for crash detection using accelerometer data.
	Alcohol Detection	2006066, 2006068	Configured MQ-3 alcohol sensor for integration.
	GSM Module	2006069	Set up GSM module for emergency SMS

			notifications during crash events.
May 15 – May 21	Hardware Setup	2006066, 2006067, 2006068	Assembled components and connected them to Arduino Uno.
	Software Knowledge	2006066, 2006068,	Continued learning software libraries for RF communication and sensor data.
	Wear Detection	2006068	Tested IR sensor sensitivity and adjusted for real-world conditions.
	GSM Module	2006069, 2006066, 2006068	Integrated GSM module for SMS functionality during crash events.
May 22 – May 28	Implementing Software & Hardware Knowledge	2006066, 2006067, 2006068	Integrated sensor readings and RF communication for data transfer.
	Group Discussion	2006066, 2006067, 2006068, 2006069	Discussed challenges and planned next steps.
	PPT Preparation	2006066, 2006067, 2006068, 2006069	Started working on the PowerPoint presentation for progress review.
June 1 – June 7	Wear Detection	2006066, 2006068	Tested tx logic in real-world scenarios, verifying ignition activation.
	Crash Detection	2006068	Finalized accelerometer calibration for crash detection.
	Gaining Software Knowledge	2006067, 2006068, 2006069	Continued learning RF communication protocol and sensor data handling.
June 8 – June 14	Alcohol Detection	2006066, 2006068	Calibrated MQ-3 alcohol sensor and set detection thresholds.
	Group Discussion	2006066, 2006067, 2006068, 2006069	Held discussions on improving system

			reliability in various conditions.
	Theory Knowledge	2006066, 2006067, 2006068, 2006069	Revisited project requirements and design.
June 15 – June 21	Crash Detection	2006067, 2006068	Finalized crash detection algorithm based on accelerometer data.
	Hardware Setup	2006066, 2006067, 2006068	Assembled the final prototype for transmitter and receiver circuits.
	Report Writing	2006066, 2006067, 2006068, 2006069	Started preparing the project report.
June 22 – June 28	Software & Hardware Integration	2006066, 2006067, 2006068	Integrated GSM module for crash notifications.
	GSM Module Testing	2006066, 2006069	Tested GSM functionality for sending SMS alerts during crash events.
	PPT Preparation	2006066, 2006067, 2006068, 2006069	Finalized the presentation slides.
July 1 – July 7	Group Discussion	2006066, 2006067, 2006068, 2006069	Discussed results from testing phase and next steps for refinement.
	Wear Detection	2006066, 2006067, 2006068	Conducted comprehensive tests to ensure reliable wear detection.
July 8 – July 14	System Testing	2006066, 2006067, 2006068, 2006069	Performed final system tests and debugging to ensure smooth operation.
	Report Writing	2006066, 2006067, 2006068, 2006069	Finalized the project report.
July 15 – July 21	PPT Preparation	2006066, 2006067, 2006068, 2006069	Refined and polished the final presentation slides.
	Software Knowledge	2006067, 2006068, 2006069	Final software modifications were completed to enhance system performance.

July 22 – July 28	Final Testing	2006066, 2006067, 2006068, 2006069	Conducted final system tests to verify all components were functioning as expected.
	Report Writing	2006066, 2006067, 2006068, 2006069	Completed the final project report.

Table 6.2 : Logbook of Project Implementation

6.2 User Manual

The Smart Helmet System is designed to improve rider safety by integrating advanced detection features into a single, easy-to-use helmet. This user manual provides instructions on how to use, maintain, and troubleshoot the system effectively.

1. System Overview

The Smart Helmet System includes three key safety features:

- **Wear Detection:** Ensures the helmet is worn properly before enabling the vehicle.

- **Battery:** The system is powered by a rechargeable lithium-ion battery. To charge, connect the helmet to a charger. Ensure the battery is fully charged before use.
- **Sensor Cleaning:** Gently clean the IR and alcohol sensors with a soft cloth. Avoid direct contact with liquids.
- **System Calibration:** If the system is not responding accurately, recalibrate the sensors through the Arduino IDE.

2. Troubleshooting

- **Helmet Not Worn Detection:**
    - Ensure the helmet fits properly and the IR sensors are aligned correctly.
- **Alcohol Detection Not Triggering:**
    - Check if the MQ-3 sensor is correctly positioned and free of obstructions.
- **Crash Detection Not Activating:**

- Ensure the accelerometer is securely attached and positioned inside the helmet.

3. Safety Information

- The **Smart Helmet System** is not a substitute for proper riding precautions. Always ride safely and responsibly.
- Ensure the system is checked periodically for proper functioning.

4. Warranty and Support

For assistance, contact our customer support team or visit the official website for troubleshooting tips, firmware updates, and warranty information.

7. Communication

7.1 Executive Summary

The Smart Helmet System is designed to enhance rider safety by integrating wear detection, alcohol detection, and crash detection features into a single, affordable, and user-friendly helmet. The system uses power on / off logic in transmitter circuit for wear detection, MQ-3 alcohol sensors to detect alcohol in the rider's breath, and accelerometers for crash detection. These sensors communicate wirelessly with the vehicle using 433 MHz RF modules, allowing real-time monitoring and immediate feedback to prevent unsafe riding behaviors. The system offers proactive safety features, such as disabling the vehicle's ignition if the helmet is not worn or alcohol is detected, and sending alerts in case of a crash. Designed for affordability and accessibility, the system aims to reduce motorcycle-related accidents, injuries, and fatalities. The Smart Helmet System integrates advanced technology to provide a scalable, effective solution for improving road safety, promoting responsible riding, and addressing key societal challenges related to road safety and public health

8. Project Management and Cost Analysis

8.1 Bill of materials

Component Name	Current Cost (TK)	Bulk Cost (TK)
MQ3 Sensor	150	120
MPU 6050	220	190

RF Receiver & Transmitter (433 MHz)	200	170
Arduino UNO (2 pieces)	600*2=1200	600 (Nano)
Relay Module	95	80
Buzzer	12	10
LED (2 pieces)	10	8
Battery (Lithium)	300	250
Helmet	700	650
GSM Module	750	700
GPS Module	440	400
I2C Display	290	270
DC Motor	350	
Total	4717	3448

9 Future Work

The Smart Helmet System offers a robust solution to improve rider safety, but there are several areas for enhancement and expansion. The following outlines potential future work to further improve the system's capabilities, usability, and impact.

1. Enhanced Sensor Accuracy

- **Improved Alcohol Detection:** Future iterations could integrate more advanced alcohol sensors that are less sensitive to environmental factors like temperature and humidity. This would increase the system's reliability and reduce false positives or negatives.
- **Crash Detection Sensitivity:** Fine-tuning the accelerometer and using additional sensors like gyroscopes could help detect a broader range of crash scenarios, including rotational impacts and side impacts**.**

2. Integration with Mobile Applications

- **Real-Time Monitoring:** Develop a mobile app that connects with the helmet via Bluetooth or Wi-Fi. The app could provide real-time updates on the rider's condition, such as alcohol detection, crash status, and wear detection.

- **Emergency Alerts:** The app could automatically send the rider's location and crash details to emergency services in case of an accident, improving emergency response times.

3. Machine Learning Integration

- **Improved Crash Detection:** Machine learning algorithms could be used to analyze riding patterns and impact data to better distinguish between normal riding behavior and genuine crash events, reducing false positives.
- **Personalized Alerts:** Machine learning could also be used to personalize safety thresholds based on rider behavior, allowing for a more customized approach to detection and alerts.

4. Extended Battery Life

- **Power Optimization:** Future work could focus on enhancing battery life through energy-efficient components and improved power management algorithms. This would make the system more practical for long-distance riders.
- **Solar Charging:** Integrating solar charging capabilities could provide a sustainable power source for the helmet, extending its operational time during daytime riding.

5. Regulatory Compliance and Standardization

- **Legal Compliance:** As motorcycle helmet laws and safety standards evolve, the system may need to adapt to comply with emerging regulations. Ensuring that the helmet meets global safety standards will be a key step for mass adoption.
- **Industry Collaboration:** Collaboration with helmet manufacturers and regulatory bodies could help develop industry-wide standards for smart helmets, promoting broader adoption of safety-enhancing technologies.

6. Global Accessibility and Affordability

- **Low-Cost Manufacturing:** Future iterations of the system could focus on making the technology more affordable for low-income regions by reducing the cost of components and utilizing local manufacturing. This would help in regions with high motorcycle use but limited access to advanced safety technologies.

- **Language Support:** Expanding the system's language support in the mobile app and display interface would make the system more accessible to non-English speaking riders globally.

7. Integration with Vehicle Systems

- **Vehicle Communication:** Future versions could enable the Smart Helmet System to communicate directly with the vehicle's onboard systems (e.g., ECU) to trigger more advanced responses, such as automatic braking or speed control in case of a crash or impaired riding.

8. Environmental Sustainability

- **Eco-friendly Materials:** Continued research into the use of sustainable, recyclable materials for the helmet and system components would help reduce its environmental impact.
- **E-Waste Management:** Efforts should be made to implement circular economy principles, ensuring that the Smart Helmet System is designed for easy disassembly and recycling at the end of its life.

10 References

10.1 Literature Review :

- **Mohan, P., et al.** (2017). *Wearable Smart Helmet System for Driver Safety*. International Journal of Engineering & Technology.

  - This paper explores the use of logic for detecting helmet wear and discusses the integration of wearable technology to improve motorcycle safety.

- **Chen, Y., et al.** (2018). *Capacitive Helmet Sensing for Wear Detection*. IEEE Transactions on Consumer Electronics.

  - Discusses the use of **capacitive sensors** integrated into helmets to detect head contact, offering greater accuracy than infrared sensors for wear detection.

- **Bharath, K., et al.** (2019). *Alcohol Detection using MQ-3 Sensor for Road Safety*. Journal of Sensors and Actuators.

- This research focuses on the **MQ-3 alcohol sensor**, which is widely used for alcohol detection in breathalyzers and wearable safety devices.

- **Khan, S., et al.** (2017). *Impact Detection in Helmets using MEMS Accelerometers*. Journal of Mechanical Engineering.

  - This paper discusses the use of **accelerometers** integrated into helmets for detecting sudden deceleration or impact during a crash.

- **Zhao, L., et al.** (2018). *Smart Helmets with Multi-Sensor Integration for Safety Applications*. IEEE Access.

  - Highlights the integration of **multiple sensors** (accelerometers, gyroscopes, and pressure sensors) into helmets to improve crash detection and provide real-time safety monitoring.

- **Ghosh, S., et al.** (2020). *Multi-Sensor Smart Helmet System for Enhanced Rider Safety*. International Journal of Innovative Research.

  - Presents a multi-sensor smart helmet system integrating **wear detection**, **alcohol detection**, and **crash detection**, offering a comprehensive approach to rider safety.

- **Srinivasan, S., et al.** (2019). *IoT-Based Smart Helmet with Real-Time Monitoring*. International Journal of Computer Applications.

  - This study explores the integration of **IoT** technology with smart helmets, allowing real-time monitoring and alerting of dangerous conditions.

- **Patel, R., et al.** (2020). *Review of Breath Alcohol Detection Methods and Devices*. Journal of Breath Research.

  - The review focuses on various methods of **breath alcohol detection** and how these methods can be integrated into wearable safety devices like smart helmets.

- **Srinivasan, S., et al.** (2020). *Affordable Smart Helmet Systems for Low-Income Regions*. Journal of Sustainable Engineering.

  - Discusses strategies to make **smart helmets affordable** by using low-cost sensors and materials, thus making them accessible in **low-income regions**.

- **Zhao, Y., et al.** (2020). *Advancements in Smart Helmet Systems and Integration with Vehicle Communication*. Journal of Transportation Safety.

    - Explores future trends in **vehicle-to-helmet communication**, enhancing safety by enabling helmets to communicate directly with vehicle systems for crash prevention.

10.2 IEEE format

[1] M. A. Rahman, S. M. Ahsanuzzaman, I. Rahman, T. Ahmed, and A. Ahsan, "IoT Based Smart Helmet and Accident Identification System," in 2020 IEEE Region 10 Symposium (TENSYMP), 2020, pp. 14–17, doi:10.1109/TENSYMP50017.2020.9230823. *[Link]* (https://doi.org/10.1109/TENSYMP50017.2020.9230823)

[2] J. V. Jayaprakash and B. Khetharani, "Smart Helmet with Rear View and Accident Detection System," International Journal of Innovative Research in Science, Engineering and Technology, 2023. *[Link]* (https://www.ijirset.com/upload/2023/june/30_Smart.pdf)

[3] M. I. Tayag and M. E. A. Capuno, "Smart Motorcycle Helmet: Real-Time Crash Detection With Emergency Notification," International Journal of Computer Science and Information Technology, June 2019. *[Link]* (https://www.ijcsit.com/docs/Volume%2011/vol11issue3/ijcsit201911033.pdf)

[4] S. J. Swathi, S. Raj, and D. Devaraj, "Microcontroller and Sensor-Based Smart Biking System for Driver's Safety," in 2019 IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS), 2019, pp. 1–5, doi:10.1109/INCOS45849.2019.8951409. *[Link]* (https://doi.org/10.1109/INCOS45849.2019.8951409)

[5] D. Bibbo, S. Conforto, A. Laudani, and G. M. Lozito, "Solar energy harvest on bicycle helmet for smart wearable sensors," in 2017 IEEE 3rd International Forum on Research and Technologies for Society and Industry (RTSI), 2017, pp. 1–6, doi:10.1109/RTSI.2017.8065926. *[Link]* (https://doi.org/10.1109/RTSI.2017.8065926)

[6] S. K. N., R. G. Puttaswamy, S. T., and Y. P. Thippeswamy, "Smart Helmet for Accident Detection and Notification," ResearchGate, 2018. *[Link]* (https://www.researchgate.net/publication/329527771_Smart_Helmet_for_Accident_Detection_and_Notification)

[7] Anonymous, "Smart Helmet based Accident Detection and Notification System for Two-Wheeler Motor Cycles," ResearchGate, 2022. *[Link]* (https://www.researchgate.net/publication/362567272_Smart_Helmet_Based_Accident_Detection_and_Notification_System_for_Two_Wheeler_Motorcycles)

[8] R. Agorku et al., "Real-Time Helmet Violation Detection Using YOLOv5 and Ensemble Learning," arXiv, Apr. 2023. *[Link]* (https://arxiv.org/abs/2304.02094)

[9] A. Aboah et al., "Real-Time Multi-Class Helmet Violation Detection Using Few-Shot Data Sampling Technique and YOLOv8," arXiv, Apr. 2023. *[Link]* (https://arxiv.org/abs/2304.04909)

[10] L. Zhao et al., "Smart Helmets with Multi-Sensor Integration for Safety Applications," IEEE Access, 2018. *[Link]* (https://ieeexplore.ieee.org/document/8469097)

Appendix:

Codes:

1. Receiver Final Code :

```
>> small font size, Consolas Size 7, double column
#include <RH_ASK.h>
#include <SPI.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

RH_ASK rf_driver(2000);
LiquidCrystal_I2C lcd(0x27, 16, 2);

const int LED_PIN = 8;
const int BUZZER_PIN = 9;
const int RELAY_PIN = 7;

unsigned long lastSignalTime = 0;
const unsigned long SIGNAL_TIMEOUT = 10000;  // ms

enum State { HELMET_NOT_WORN, NORMAL, ALCOHOL, CRASH };
State systemState = HELMET_NOT_WORN;

void setup() {
  Serial.begin(9600);

  pinMode(LED_PIN, OUTPUT);
  pinMode(BUZZER_PIN, OUTPUT);
  pinMode(RELAY_PIN, OUTPUT);

  digitalWrite(LED_PIN, LOW);
  digitalWrite(BUZZER_PIN, LOW);
  digitalWrite(RELAY_PIN, LOW); // By default, motor
OFF

  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("Status: ");
  lcd.setCursor(0, 1);
  lcd.print("System Starting");

  if (!rf_driver.init()) {
    Serial.println("RF Rx init failed!");
  } else {
    Serial.println("Bike Rx Ready");
  }
}

void loop() {
  uint8_t buf[12];
  uint8_t buflen = sizeof(buf);

  if (rf_driver.recv(buf, &buflen)) {
    buf[buflen] = '\0';
    lastSignalTime = millis();

    if (strcmp((char*)buf, "ALCOHOL") == 0) {
      systemState = ALCOHOL;
    } else if (strcmp((char*)buf, "NORMAL") == 0) {
      systemState = NORMAL;
    } else if (strcmp((char*)buf, "CRASH") == 0) {
      systemState = CRASH;
    }
  }

  // If no signal for too long, helmet is not worn
  if (millis() - lastSignalTime > SIGNAL_TIMEOUT) {
    systemState = HELMET_NOT_WORN;
  }

  // Handle system states based on received message
  if (systemState == CRASH) {
    Serial.println("Crash Detected!");
    digitalWrite(LED_PIN, HIGH);   // Turn on LED
    digitalWrite(BUZZER_PIN, HIGH); // Turn on Buzzer
    digitalWrite(RELAY_PIN, LOW);   // Turn off motor
    lcd.setCursor(0, 0);
    lcd.print("Status: ALERT!  ");
    lcd.setCursor(0, 1);
    lcd.print("Crash Detected  ");
  } else if (systemState == ALCOHOL) {
    Serial.println("Alcohol Detected!");
    digitalWrite(LED_PIN, HIGH);
    digitalWrite(BUZZER_PIN, HIGH);
    digitalWrite(RELAY_PIN, LOW); // Turn off motor
    lcd.setCursor(0, 0);
    lcd.print("Status: ALERT!  ");
    lcd.setCursor(0, 1);
    lcd.print("Alcohol Detected");
  } else if (systemState == NORMAL) {
    Serial.println("Normal - Helmet OK");
    digitalWrite(LED_PIN, LOW);
    digitalWrite(BUZZER_PIN, LOW);
    digitalWrite(RELAY_PIN, HIGH); // Turn on motor
    lcd.setCursor(0, 0);
    lcd.print("Status: NORMAL  ");
    lcd.setCursor(0, 1);
    lcd.print("Helmet OK       ");
  } else if (systemState == HELMET_NOT_WORN) {
    Serial.println("Helmet Not Worn");
    digitalWrite(LED_PIN, LOW);
    digitalWrite(BUZZER_PIN, LOW);
    digitalWrite(RELAY_PIN, LOW); // Turn off motor
    lcd.setCursor(0, 0);
    lcd.print("Status: ALERT!  ");
    lcd.setCursor(0, 1);
    lcd.print("Helmet Not Worn ");
  }

  delay(1000);
}
```

2. Transmitter Final Code :

```
>> small font size, Consolas Size 7, double column
(Without GSM )
#include <SoftwareSerial.h>
#include <TinyGPS++.h>
#include <RH_ASK.h>
#include <SPI.h>
#include <Wire.h>
#include <MPU6050.h>

// GPS TX to Arduino D4, GPS RX to Arduino D3
SoftwareSerial gpsSerial(4, 3); // RX, TX
TinyGPSPlus gps;

RH_ASK rf_driver(2000);
MPU6050 mpu;

const int MQ3_PIN = A0;
const float ALC_THRESH = 3.5;
const int LED_PIN = 8;  // Crash detection LED
const float IMPACT_THRESHOLD = 1.5;  // Crash threshold

float latitude = 90.390693;
float longitude = 23.726440;

bool crashState = false;
unsigned long crashStartTime = 0;

void setup() {
  Serial.begin(9600);
  gpsSerial.begin(9600);   // GPS module baud rate
  Serial.println("Waiting for GPS signal…");

  if (!rf_driver.init()) {
    Serial.println("RF Tx init failed!");
  } else {
    Serial.println("Helmet Tx Ready");
  }

  Wire.begin();
  mpu.initialize();
  pinMode(LED_PIN, OUTPUT);
  digitalWrite(LED_PIN, LOW);

  if (mpu.testConnection()) {
    Serial.println("MPU6050 Connected OK");
  } else {
    Serial.println("MPU6050 Connection FAILED!");
  }
}

void loop() {
  while (gpsSerial.available() > 0) {
    gps.encode(gpsSerial.read());
  }

  // Read alcohol sensor
  int mq3Value = analogRead(MQ3_PIN);
  float mq3Voltage = mq3Value * (5.0 / 1023.0);
  Serial.print("MQ3 Voltage: ");
  Serial.println(mq3Voltage);

  // Crash detection logic
  int16_t ax_raw, ay_raw, az_raw;
  mpu.getAcceleration(&ax_raw, &ay_raw, &az_raw);
  float ax = ax_raw / 16384.0;
  float ay = ay_raw / 16384.0;
  float az = az_raw / 16384.0;
```

```
    // Print acceleration values to the serial monitor
for debugging
  Serial.print("ax: ");
  Serial.print(ax);
  Serial.print(" | ay: ");
  Serial.print(ay);
  Serial.print(" | az: ");
  Serial.println(az);

 float A = sqrt(ax * ax + ay * ay + az * az);
  float 54elta = abs(A – 1.0);

  // Print acceleration magnitude and delta value to
monitor crash detection
  Serial.print("A: ");
  Serial.print(A);
  Serial.print(" | Delta: ");
  Serial.println(54elta);

  if (54elta > IMPACT_THRESHOLD && !crashState) {
    Serial.println("!!! CRASH DETECTED !!!");

    Serial.print("Latitude: ");
    Serial.println(latitude, 6);
    Serial.print("Longitude: ");
    Serial.println(longitude, 6);

    digitalWrite(LED_PIN, HIGH); // Turn on LED
    crashState = true;
    crashStartTime = millis();
    rf_driver.send((uint8_t *)"CRASH", 5); // Send
crash signal to RX
    rf_driver.waitPacketSent();
  }

  // Alcohol detection
  else if (mq3Voltage > ALC_THRESH) {
    const char *msg = "ALCOHOL";
    rf_driver.send((uint8_t *)msg, strlen(msg));
    rf_driver.waitPacketSent();
    Serial.println("ALCOHOL Sent");
  }
  else {
    const char *msg = "NORMAL";
    rf_driver.send((uint8_t *)msg, strlen(msg));
    rf_driver.waitPacketSent();
    Serial.println("NORMAL Sent");
  }

  // Crash state reset after a delay
  if (crashState) {  // LED stays ON for 10 seconds
    delay(10000);
    digitalWrite(LED_PIN, LOW); // Turn off LED
    crashState = false;
    Serial.println("Crash state cleared.");
  }

  delay(1000);
}
```

```
(With GSM )
#include <Wire.h>
#include <MPU6050.h>
#include <TinyGPS++.h>
#include <SoftwareSerial.h>
#include <RH_ASK.h>

// Initialize the MPU6050 sensor and GPS module
MPU6050 mpu;
TinyGPSPlus gps;

// SIM900A GSM Module
SoftwareSerial sim900(7, 8); // RX, TX
// Neo 6M GPS Module
SoftwareSerial gpsSerial(4, 3); // RX, TX

// RF driver for crash detection
RH_ASK rf_driver(2000);

// Pin definitions
const int LED_PIN = 5;  // Crash detection LED
const int MQ3_PIN = A0; // Alcohol sensor input pin
const float ALC_THRESH = 3.0;
const float IMPACT_THRESHOLD = 1.5; // Crash detection
threshold

// Moving average variables for alcohol sensor
const int MOVING_AVG_SIZE = 10;  // Number of samples
for moving average
float mq3Readings[MOVING_AVG_SIZE]; // Array to store
last readings
int mq3Index = 0; // Index for the next reading
float mq3Avg = 0.0; // Moving average of MQ3 sensor
readings

// Moving average variables for MPU6050 data
float axReadings[MOVING_AVG_SIZE];
float ayReadings[MOVING_AVG_SIZE];
float azReadings[MOVING_AVG_SIZE];
int mpuIndex = 0; // Index for MPU6050 data buffer
float axAvg = 0.0, ayAvg = 0.0, azAvg = 0.0; // Moving
averages for MPU6050 data

// Variables for crash detection
bool crashState = false;
unsigned long crashStartTime = 0;

// GPS coordinates for the location (Example)
float latitude = 90.390693;
float longitude = 23.726440;

void setup() {
  Serial.begin(9600);    // Initialize serial
communication for debugging
  gpsSerial.begin(9600); // GPS module baud rate
  sim900.begin(9600);    // GSM module baud rate
  rf_driver.init();      // Initialize RF module for
crash detection

  Wire.begin();
  mpu.initialize();
  pinMode(LED_PIN, OUTPUT);  // Set LED pin as output
  digitalWrite(LED_PIN, LOW); // Start with LED off

  // Initialize the MQ3 readings and MPU6050 buffers
  for (int i = 0; i < MOVING_AVG_SIZE; i++) {
    mq3Readings[i] = 0;
    axReadings[i] = 0;
    ayReadings[i] = 0;
    azReadings[i] = 0;
  }

  if (mpu.testConnection()) {
    Serial.println("MPU6050 Connected");
  } else {
    Serial.println("MPU6050 Connection FAILED!");
  }

  Serial.println("System Initialized...");
}

void loop() {
```

```
// Calculate moving averages for MPU accelerometer data
  axAvg = 0.0;
  ayAvg = 0.0;
  azAvg = 0.0;
  for (int i = 0; i < MOVING_AVG_SIZE; i++) {
    axAvg += axReadings[i];
    ayAvg += ayReadings[i];
    azAvg += azReadings[i];
  }
  axAvg /= MOVING_AVG_SIZE; // Calculate average for X-
axis
  ayAvg /= MOVING_AVG_SIZE; // Calculate average for Y-
axis
  azAvg /= MOVING_AVG_SIZE; // Calculate average for Z-
axis

  Serial.print("ax Avg: ");
  Serial.print(axAvg);
  Serial.print(" | ay Avg: ");
  Serial.print(ayAvg);
  Serial.print(" | az Avg: ");
  Serial.println(azAvg);

  // If crash is detected, send SMS and RF signal
  if (crashState) {
    sendSMS();
    rf_driver.send((uint8_t *)"CRASH", 5);  // Send
crash signal via RF
    rf_driver.waitPacketSent();
    digitalWrite(LED_PIN, HIGH); // Turn on LED
    delay(10000); // Hold crash detection for 10
seconds
    digitalWrite(LED_PIN, LOW);  // Turn off LED
    crashState = false; // Reset crash state
  }

  // If alcohol detected based on moving average, send
RF signal and SMS
  if (mq3Avg > ALC_THRESH) {
    sendSMS("ALCOHOL DETECTED");
    rf_driver.send((uint8_t *)"ALCOHOL", 7); // Send
alcohol detection signal
    rf_driver.waitPacketSent();
    Serial.println("ALCOHOL Sent");
  } else {
    rf_driver.send((uint8_t *)"NORMAL", 6);  // Send
normal signal
    rf_driver.waitPacketSent();
    Serial.println("NORMAL Sent");
  }

  delay(1000);
}

// Function to check for crash detection
void checkCrash() {
  // Calculate magnitude of the acceleration (use
moving averages for smoother detection)
  float A = sqrt(axAvg * axAvg + ayAvg * ayAvg + azAvg
* azAvg);
  float deltaA = abs(A - 1.0);  // Compare to gravity

  // If the delta exceeds the threshold, consider it a
crash
  if (deltaA > IMPACT_THRESHOLD && !crashState) {
    Serial.println("CRASH DETECTED!");
    crashState = true;
    crashStartTime = millis();
  }
}

// Function to read GPS data and send SMS
void sendSMS(const char *message = "Crash Detected!") {
  Serial.println("Sending SMS...");

  // Send command to set SMS format to text mode
  sim900.println("AT+CMGF=1");
  delay(1000);

  // Send command to set the recipient phone number
(replace with your number)
```

```
  while (gpsSerial.available() > 0) {
    gps.encode(gpsSerial.read()); // Get GPS data
  }

  // Check for crash detection
  checkCrash();

  // Read the alcohol sensor value and apply moving
average
  int mq3Value = analogRead(MQ3_PIN);
  float mq3Voltage = mq3Value * (5.0 / 1023.0); //
Convert to voltage

  // Update moving average with the new sensor reading
for alcohol sensor
  mq3Readings[mq3Index] = mq3Voltage;
  mq3Index = (mq3Index + 1) % MOVING_AVG_SIZE;  //
Circular buffer logic

  // Calculate the moving average for the alcohol
sensor
  mq3Avg = 0.0;
  for (int i = 0; i < MOVING_AVG_SIZE; i++) {
    mq3Avg += mq3Readings[i];
  }
  mq3Avg /= MOVING_AVG_SIZE; // Calculate the average

  Serial.print("MQ3 Moving Average: ");
  Serial.println(mq3Avg);

  // Read the MPU6050 accelerometer data and apply
moving average
  int16_t ax_raw, ay_raw, az_raw;
  mpu.getAcceleration(&ax_raw, &ay_raw, &az_raw);

  // Convert raw data to acceleration values
  float ax = ax_raw / 16384.0;
  float ay = ay_raw / 16384.0;
  float az = az_raw / 16384.0;

  // Update moving averages for accelerometer readings
  axReadings[mpuIndex] = ax;
  ayReadings[mpuIndex] = ay;
  azReadings[mpuIndex] = az;
  mpuIndex = (mpuIndex + 1) % MOVING_AVG_SIZE; //
Circular buffer for MPU data
```

```
  sim900.println("AT+CMGS=\"+8801627978697\""); //
replace with your number
  delay(1000);

  // Send the SMS content
  sim900.print(message);
  sim900.print("\nLocation:\n");
  sim900.print("Latitude: ");
  sim900.println(gps.location.lat(), 6);  // Display
Latitude
  sim900.print("Longitude: ");
  sim900.println(gps.location.lng(), 6);  // Display
Longitude
  sim900.print("Google Maps:
https://maps.google.com/?q=");
  sim900.print(gps.location.lat(), 6);
  sim900.print(",");
  sim900.println(gps.location.lng(), 6);
  sim900.write(26); // End of SMS (Ctrl+Z)

  // Print SMS sent status to Serial Monitor
  Serial.println("SMS Sent!");
}
```