**Smart Helmet with Sensor-Based Crash Detection and Automated Emergency Communication**

A Project report is submitted in partial fulfillment of the requirements for the award of Degree of Bachelor of Science in Electrical and Electronic Engineering.

**Submitted by**

Kawsar Mahmud Tanveer Khan

221-33-1740

Nobab Bin Shahanur

221-33-1580

Nazmul Islam Bahar

221-33-1594

**Supervised by**

Dr. Tama Fouzder

Associate Professor

Department of Electrical and Electronic Engineering



Department of Electrical and Electronic Engineering

Faculty of Engineering

DAFFODIL INTERNATIONAL UNIVERSITY

**January, 2026**

Declaration

We hereby declare that this project entitled “Smart Helmet with Sensor-Based Crash Detection and Automated Emergency Communication” represents our own work and has been carried out in the laboratories of the Department of Electrical and Electronic Engineering under the Faculty of Engineering, Daffodil International University, in partial fulfillment of the requirements for the degree of Bachelor of Science in Electrical and Electronic Engineering.

This project has not been previously submitted, in whole or in part, for the award of any degree, diploma, or other academic qualification at this or any other institution. We have made sincere efforts to identify and address all potential risks associated with this research during its execution. All relevant ethical and safety considerations have been properly followed where applicable, and we have duly acknowledged our responsibilities and the rights of all participants involved in this project.

**Signature of the candidates**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Kawsar Mahmud Tanveer Khan

ID: 221-33-1740

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Nobab Bin Shahanur

ID: 221-33-1580

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Nazmul Islam Bahar

ID: 221-33-1594

Approval

The project entitled **“Smart Helmet with Sensor-Based Crash Detection and Automated Emergency Communication”** submitted by **Kawsar Mahmud Tanveer Khan (ID: 221-33-1740)**, **Nobab Bin Shahanur (ID: 221-33-1580)**, and **Nazmul Islam Bahar (ID: 221-33-1594)** has been carried out under my supervision and is accepted as satisfactory in partial fulfillment of the requirements for the degree of **Bachelor of Science in Electrical and Electronic Engineering** in **January, 2026**.

Signed

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Dr. Tama Fouzder**

Associate Professor

Department of Electrical and Electronic Engineering

Faculty of Engineering

Daffodil International University

Dedicated

To

**“Our Beloved Parents”**

Table of Contents

|  |  |
| --- | --- |
| **List of Figures** | **vii** |
| **List of Tables** | **viii** |
| **List of Abbreviations** | **ix** |
| **Acknowledgment** | **x** |
| **Abstract** | **xi** |

[**CHAPTER 1: INTRODUCTION 1**](#_Toc118501734)

[1.1 Motivation 1](#_Toc118501736)

[1.2 Problem Statement and Proposed solution(s) 2](#_Toc118501737)

[1.3 Objectives 3](#_Toc118501738)

[1.4 Brief Methodology 4](#_Toc118501739)

[1.5 Implementation Schedule 4](#_Toc118501740)

[1.6 Structure of the Report 5](#_Toc118501741)

[**CHAPTER 2:**](#_Toc118501742)[**LITERATURE REVIEW 6**](#_Toc118501743)

[2.1 Introduction 6](#_Toc118501744)

[2.2 Related Research 6](#_Toc118501745)

[2.3 Compare and Contrast 7](#_Toc118501746)

[2.4 Summary 8](#_Toc118501747)

[**CHAPTER 3**](#_Toc118501748)**:** [**MATERIALS AND METHODS 10**](#_Toc118501749)

[3.1 Introduction 10](#_Toc118501750)

[3.2 Methods and Materials or System Design and Components 10](#_Toc118501751)

[3.3 Design Specifications. Standards and Constraints 13](#_Toc118501752)

[3.4 System Analysis or Design Analysis 15](#_Toc118501753)

[3.5 Experimental Setup 16](#_Toc118501754)

[3.6 Summary 18](#_Toc118501755)

[**CHAPTER 4**](#_Toc118501756)**:** [**RESULTS AND DISCUSSIONS 19**](#_Toc118501757)

[4.1 Results 19](#_Toc118501758)

[4.2 Discussions 23](#_Toc118501759)

[**CHAPTER 5**](#_Toc118501760)**:** [**PROJECT MANAGEMENT 25**](#_Toc118501761)

[5.1 Task, Schedule and Milestones 25](#_Toc118501762)

[5.2 Resources and Cost Management 25](#_Toc118501763)

[5.3 Lesson Learned 27](#_Toc118501764)

[**CHAPTER 6**](#_Toc118501765)**:** [**IMPACT ASSESSMENT OF THE PROJECT 29**](#_Toc118501766)

[6.1 Economical, Societal and Global Impact 29](#_Toc118501767)

[6.2 Environmental and Ethical Issues 30](#_Toc118501768)

[6.3 Utilization of Existing Standards or Codes 30](#_Toc118501769)

[6.4 Other Concerns 30](#_Toc118501770)

[**CHAPTER 7**](#_Toc118501771)**:**[**CONCLUSIONS AND RECOMMENDATIONS 32**](#_Toc118501772)

[7.1 Conclusions 32](#_Toc118501773)

[7.2 New Skills and Experiences Learned 32](#_Toc118501774)

[7.3 Future Recommendations 32](#_Toc118501775)

[**REFERENCES 34**](#_Toc118501776)

List of Figures

|  |  |  |
| --- | --- | --- |
| *Figure No* | *Figure Name* | *Page No.* |
| Fig. 3.1 | Circuit diagram of Smart Helmet. | 15 |
| Fig. 3.5 | Hardware setup of Smart Helmet. | 17 |
| Fig. 4.1 | G-Force and Angular Velocity during Normal Riding Condition | 19 |
| Fig. 4.2 | Sensor Response during Sudden Acceleration | 20 |
| Fig. 4.3 | Sensor Response during Sudden Braking | 20 |
| Fig. 4.4 | Sensor Response on Speed Breaker and Rough Road | 21 |
| Fig. 4.5 | MPU6050 Sensor Response during Controlled Helmet Drop Test on Soft Surface | 22 |
| Fig. 4.6 | Sample emergency SMS received with GPS coordinates | 23 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

List of Tables

|  |  |  |
| --- | --- | --- |
| *Table No* | *Table Name* | *Page No.* |
| Table 4.1 | False alarm handling test results | 22 |
| Table 5.1 | Equipment and component cost breakdown | 27 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

List of Abbreviations

|  |  |
| --- | --- |
| DIU | Daffodil International University |
| GPS | Global Positioning System |
| GSM | Global System for Mobile Communication |
| MPU | Motion Processing Unit |
| OLED | Organic Light Emitting Diode |

Acknowledgement

First and foremost, we express our sincere gratitude to **Almighty Allah** for granting us the strength, patience, and guidance to successfully complete this project.

We want to pay our utmost respect to our Supervisor **Dr. Tama Fouzder, Associate Professor** of the **Department of EEE, Daffodil International University** for who has given us the chance to work on an impactful idea and taken care of every issue of development of this concept. Then we would like to take this opportunity to express gratitude to our supervisor for being dedicated in supporting, motivating and guiding us throughout this project. This project can’t be done without her useful advice and help. Also thank her very much for giving us the opportunity to work with this project.

We are also thankful to the **Department of Electrical and Electronic Engineering, Daffodil International University**, for providing the necessary laboratory facilities, equipment, and academic environment required to carry out this project successfully. Our sincere appreciation goes to the faculty members and laboratory staff of the department for their cooperation and technical assistance.

Abstract

Motorcycle accidents are very common, particularly in developing countries where bikes are used every day. In many serious accidents, riders become unconscious or badly injured and not able to call for help. This problem becomes even worse at night or in rural areas, where animals such as dogs or foxes may suddenly jump onto the road, causing the motorcycle to crash. If the rider slips off the road with the bike and becomes unconscious, it becomes very difficult to get help, as rural roads and highways often have very little traffic. In some cases, the rider may remain unnoticed for hours, and emergency assistance does not arrive quickly. Normal helmets only protect the head and do nothing after an accident happens.

To solve this problem, our project introduces a **Smart Helmet** that can automatically detect accidents and ask for help. The helmet uses an accelerometer and a gyroscope to monitor sudden changes in force and angular movement. When these values exceed the threshold value, the system understands that a crash has occurred. A buzzer and cancel button are included so the rider can stop the alert in case of a small fall or false detection.

If the system observes an accident, it automatically sends an emergency text message with the rider's live GPS location and a Google Maps link to certain number. This allows help to get to rider fast without anyone having to do anything. The system is reliable and cheap because it runs on a rechargeable battery and uses cheap parts.

This Smart Helmet can reduce emergency response time and help save lives, making motorcycle travel safer.

***Keywords: Smart Helmet, Accident Detection, GPS, GSM, Emergency Communication, Rider Safety***

Chapter 1

Introduction

1.1 Motivation

Motorcycle accidents are still a serious safety problem in many areas, particularly in developing countries where motorcycles are used for daily transportation. Due to lower physical protection, riders are highly exposed to severe injuries during accidents. The risk becomes more higher when fast medical care is not available, which is a common situation in rural locations and during night-time travel.

Accidents occurring in low-traffic or poorly lighted regions may remain unnoticed for a long period. In such situation, the rider may become unconscious or physically debilitated, making it hard to contact emergency services or family members. Delayed communication in these serious times often leads to late rescue operations, increasing the chance of severe damage or death.

Traditional helmets only protect us passively and do not help us deal with emergencies after an accident. Smart protection systems can make riding safer thanks to recent improvement in embedded sensor, wireless communication, and location-tracking technologies. But one of the worst things about automatic accident detection systems is

that they often go off for no reason.   
  
 Not every impact that is seen produces substantial damage. A rider may suffer a modest accident, such falling at a slow speed, and stay alert and able to move with only minimal damage. If this happens, automatically sending an emergency SMS could make family member panic for no reason and make the system less reliable. Sensor based systems could also sometimes mistake fast movements or changes in traffic for accidents.  
  
The purpose of this Smart Helmet project is to develop a smart system that can quickly find accidents and give alarms, but it should still work well in real life. If the rider is conscious and safe after an accident, they can turn off the alarm and prevent SMS to be sent by pressing the manual cancellation button. This feature also lets the user turn off notifications that are sent because of false detections. This makes the system more accurate and the user more likely to trust it.

The project also focuses in the real world uses and not costing too much. A lot of the smart safety systems that are already out there are challenging to implement on a wide scale and cost a lot of money. This study is about a design that is affordable, reliable, and easy to implement with parts that are easy to find. This makes it an ideal choice for everyday use.

1.2 Problem Statement and Proposed solution(s)

**Problem Statement**

* Motorcycle riders are at a high risk of serious injury or death because they do not have much physical protection.
* One of the main reasons riders die in accidents is because help comes too late.
* Accidents sometimes go unnoticed, especially when:
  + In the rural areas
  + While riding at night
  + When there isn't much traffic
* Riders who are hurt may:
  + Lose consciousness
  + Become unable to move physically
  + Not being able to ask for help
* In serious accidents, manual communication in an emergency is not reliable.
* Traditional helmets simply offer passive safety and don't help in an emergency.
* Smart safety systems that are already in use generally have:
  + A lot of money
  + Architecture of a complicated system
  + Not very good at adapting to real-life riding situations
* False alarms are common in accident detection systems because of:
  + Road type
  + Misinterpretation of sensors
* False alarms lower:
  + Reliability of the system
  + Trust in the user
* There is a pressing demand for a system that:
  + Dependable
  + Cheap
  + Smart
  + Able to find accidents automatically and send emergency messages
  + Able to deal with misleading or non-critical detections well

**Suggested Solution(s)**

* The idea suggests a technique for detecting accidents and sending emergency alerts that uses a Smart Helmet.
* The system brings together:
  + Sensors for motion built in
  + GSM communication without wires
  + Tracking your location with GPS
* Motion sensors keep an eye on how riders move in real time.
* To find serious accidents, they look for strange patterns of acceleration and collision.
* After finding a possible crash:
  + The system goes into a warning phase.
  + A loud buzzer alert goes off.
* During the warning phase, the rider can:
  + If you are aware and safe, cancel the emergency process.
* This cuts down on bogus emergency warnings by a lot.
* If the alert isn't canceled within a certain amount of time:
  + The system sends an emergency SMS on its own.
* The message for emergencies says:
  + GPS location in real time
  + Sent to a set contact through the GSM module
* The technology operates without the rider having to do anything, even if the rider is:
  + Not aware
  + Not able to talk
* The answer is based on:
  + Not too expensive
  + Simple design
  + Usability in the real world
* Uses parts that are easy to find, therefore it can be used for:
  + Regions that are growing
  + People that ride motorcycles every day
* The mechanism for canceling false alarms:
  + Makes the system more reliable
  + Makes users more likely to trust you
* Overall, the Smart Helmet that has been suggested is a useful, smart, and effective way to keep people safe when they are in a motorbike accident.

1.3 Objectives

The main objectives of the project are as follows:

* To design and build a smart helmet system that can automatically detect motorcycle accidents using sensors.
* To detect real accident and reduce false alarm produced by some minor impacts.
* To create an automated emergency alert system that sends text messages to certain people without the rider having to do anything.
* To use GPS to share the real time location of the accident so that emergency help can reach the place quickly.
* To provide a manual cancel button for the rider to stop emergency SMS in case of small accidents or false alarms.

1.4 Brief Methodology

The Smart Helmet system is consist of motion sensors, a microcontroller, GPS, and GSM modules. When the bike is moving normally, the motion sensor keeps an eye on the acceleration and angular velocity data all the time.

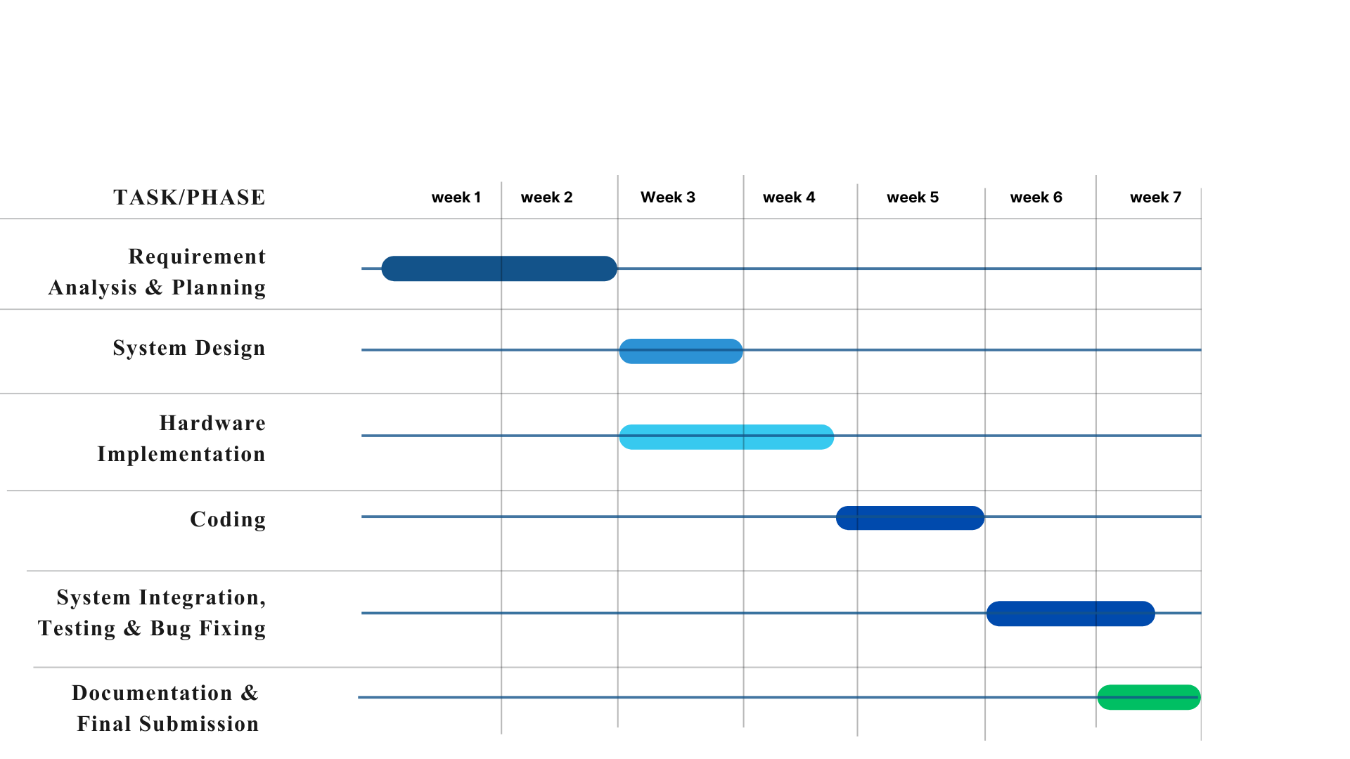
An algorithm for detecting accidents uses sensor data in real time to find unusual impact patterns that are linked to serious crashes. The system sounds an alarm and initiates a short delay period when it thinks there might be an accident. This gives the rider time to cancel the emergency alert if the accident is minimal or the detection is wrong.

If there is no cancellation within the set time, the system immediately gets the rider's GPS location and sends an emergency SMS to a list of contacts using the GSM network. After that, the system is tested in multiple situations to make sure it can accurately identify things, communicate reliably, and work well overall.

1.5 Implementation Schedule

A systematic Work Breakdown Structure (WBS) is used to plan the execution of the proposed Smart Helmet system. The project's tasks are broken down into logical phases, with each one depending on the successful execution of the one before it. A tentative time schedule is made to keep track of progress and make sure the project is finished on time.

The activities are scheduled over a **7-week timeline**, as shown in the Gantt Chart below. Each phase follows a sequential dependency to ensure smooth progress.



1.6 Structure of the Report

There are seven chapters in this study. They go from finding the problem to making final findings and suggestions for the future. This way, the proposed Smart Helmet system is presented in a straightforward and organized way.

Chapter 1 gives an overview of the project by talking about its background and motivation, defining the issue statement, listing the goals, describing the methodology, showing the implementation timetable, and discussing how the report is organized as a whole.

Chapter 2 gives a full assessment of the literature on current smart helmet systems and technologies for finding accidents. It talks about other studies, compares the different methods that are already out there, points out areas where more research is needed, and sums up the results that support the recommended solution.

Chapter 3 talks about the materials, processes, and system design of the Smart Helmet that is being suggested. This chapter talks about the system's hardware parts, architecture, design requirements, limitations, and the steps taken to build it.

Chapter 4 shows the results and observations from testing the system. It also has a long talk about how well the system works, how accurate it is, and what its limits are when it is used in different ways.

Chapter 5 talks on project management, such as organizing tasks, setting milestones, using resources, keeping costs down, and what was learnt during the project development process.

Chapter 6 talks about the project's effects on the economy, society, the environment, and ethics to see how it will affect them. It also talks about the rules, codes, and standards that are important for the system design.

Chapter 7 wraps up the report by going over the project's main successes. It talks about new abilities and experiences learned and gives suggestions on how to make the proposed Smart Helmet system better and add to it in the future.

Chapter 2

Literature Review

2.1 Introduction

The rapid growth of intelligent transportation systems and embedded safety technologies has led to increasing research interest in motorcycle rider protection and accident prevention mechanisms. Among these, smart helmet systems have emerged as a promising solution to enhance rider safety by combining sensing, communication, and automation technologies.

This chapter presents a comprehensive review of existing research works related to smart helmets, accident detection techniques, and emergency alert systems. The purpose of this literature review is to analyze previously proposed methods, identify commonly used technologies, and understand the strengths and limitations of existing solutions. Particular attention is given to sensor-based accident detection, GPS-based location tracking, GSM-based emergency communication, and mechanisms used to reduce false alarms.

By examining relevant journals, conference papers, technical reports, and existing prototypes, this chapter highlights current trends and research gaps in smart helmet development. The insights gained from this review provide a strong foundation for the proposed system design and help justify the need for a reliable, low-cost, and practical Smart Helmet solution suitable for real-world application, especially in rural and low-resource environments.

2.2 Related Research

Road safety for motorcycle riders has been an active area of research due to the high fatality rate associated with two-wheeler accidents. According to the World Health Organization (WHO), motorcyclists represent a significant portion of global road traffic deaths, particularly in low- and middle-income countries [1]. This has motivated researchers to explore intelligent safety systems that can reduce accident severity and improve post-accident emergency response.

A **smart helmet** can be defined as a protective headgear integrated with electronic sensors, processing units, and communication modules to enhance rider safety beyond conventional passive protection [2]. Most smart helmet research focuses on three core aspects: accident detection, emergency communication, and rider condition monitoring.

Early studies primarily addressed **preventive safety mechanisms**. For example, systems were proposed to ensure helmet usage and prevent drunk driving by integrating pressure sensors and alcohol sensors with vehicle ignition control [3]. While effective in enforcing safety rules, these systems do not address situations where an accident has already occurred.

To handle post-accident scenarios, researchers introduced **sensor-based accident detection techniques**, commonly using accelerometers. Sudden changes in acceleration beyond predefined threshold values are interpreted as crash events [4]. However, accelerometer-only approaches are sensitive to road irregularities, sudden braking, or sharp turns, which often leads to false accident detection.

To improve reliability, later research incorporated **gyroscopes along with accelerometers**, allowing detection of both linear acceleration and angular motion [5]. These multi-sensor approaches showed better accuracy in identifying real crashes. Nevertheless, many of these systems assume that every detected crash requires emergency assistance, which may not be true in cases of low-speed falls or minor accidents.

Several studies integrated **GPS and GSM technologies** to enable automatic emergency alert systems. Upon accident detection, these systems send SMS notifications containing the rider’s geographic location to predefined contacts [6]. Although this significantly reduces emergency response time, a major limitation identified in the literature is the lack of **false alarm management**, which can cause unnecessary panic and reduce user trust.

Recent research highlights the importance of **user intervention mechanisms**, such as delay-based alerts and manual cancellation buttons, to address false positives [7]. Additionally, researchers emphasize the need for **low-cost and energy-efficient designs** to ensure suitability for developing countries and rural deployment [8].

The design approach of the proposed Smart Helmet system in this project is supported by prior academic coursework, including embedded systems, microcontrollers, sensors and instrumentation, communication systems, and IoT-based system design. Concepts such as sensor interfacing, signal thresholding, serial communication, and power management learned during coursework have directly influenced the system architecture and implementation strategy.

From the reviewed literature, it is evident that although existing smart helmet systems provide valuable safety features, challenges remain in achieving accurate accident detection, minimizing false alarms, ensuring affordability, and maintaining practical usability. These limitations form the basis for the proposed solution presented in this project.

2.3 Compare and Contrast

Research on smart helmet systems shows that they have made a lot of progress in making motorcycle riders safer by using sensing and communication technology. But when these systems are tested in real-world riding situations, a close comparison shows that they have a lot of problems.

The first smart helmets were mostly about safety features that would keep people from getting hurt, such detecting when someone was wearing a helmet or drinking alcohol. These systems do a good job of encouraging safe riding, but they don't help when someone needs medical help right away after an accident. The proposed system, on the other hand, focuses on emergency reaction after an accident, which makes it good for situations that could be life-threatening.

A number of research used accelerometer-based accident detection methods to find sudden collisions. These systems are easy to use and cheap, but they are quite sensitive to bumps in the road, sudden stops, and quick curves, which might lead to false accident detection. Later studies made detection more accurate by combining data from an accelerometer and a gyroscope. This cuts down on false positives, but it doesn't get rid of them completely. The suggested method also uses data from multiple sensors, but it makes it even more reliable by adding a way for the user to cancel it.

Many current systems use GPS and GSM modules together to send out emergency notifications automatically after an accident is found. This method does speed up response time, but most systems presume that every mishap needs emergency help. This assumption can cause unwanted alarms when there are small accidents or falls at moderate speeds. The proposed system, on the other hand, adds an auditory warning and a countdown period. This lets the rider dismiss the alert if they are awake and safe, which makes it easier to use and more trustworthy.

Another problem with current designs is that they don't focus enough on real-world deployment issues like power management, network instability, and usability in rural areas. Most studies offer conceptual or simulation-based solutions that do not take into account battery-operated operation and GSM power fluctuations. The suggested Smart Helmet directly tackles these practical limitations by prioritizing energy conservation, steady performance, and utility in rural and nocturnal contexts where human communication may be unfeasible.

Also, many of the described systems are either expensive or hard to use, which makes it hard for developing areas to use them. The proposed design puts price and simplicity first by employing parts that are easy to get while still being reliable. Compared to many other systems, this makes the system better for wider use.

In conclusion, the suggested system builds on existing smart helmet systems by making false alarm handling better, making them more reliable in the real world, and focusing on user-centered and low-cost design. These changes make the proposed Smart Helmet a better option for everyday motorcycle riders because it is easier to use and install.

2.4 Summary

This chapter looked at previous studies on smart helmet systems, ways to find accidents, and ways to send emergency alerts. The literature emphasizes the significance of incorporating sensors, communication modules, and automation to improve motorcycle rider safety, especially in post-accident situations where prompt help is essential.

The analyzed studies show that accelerometer-based and multi-sensor systems for detecting accidents can find crashes, however they typically have false alarms and don't let users intervene. In the same way, GPS and GSM-based emergency notification systems cut down on response time, but they often assume that all incidents need emergency help. Also, a lot of current designs don't take into account real-world problems including limited power, usability in rural areas, and cost-effectiveness.

The comparison of current systems shows that there are major research gaps in managing false alarms, putting systems into use in the real world, and designing systems with people in mind. These restrictions strongly support the suggested Smart Helmet system, which emphasizes dependable accident detection, user-managed alarm cancelation, cost-effective deployment, and adaptability to actual riding conditions.

The findings from this literature analysis serve as the basis for the system design and methodology outlined in the following chapter, which examines the proposed Smart Helmet architecture and implementation specifics.

Chapter 3

Materials and Methods

3.1 Introduction

This chapter describes how the planned Smart Helmet system was designed and built. The system is meant to fill the research gaps found in the literature study by providing reliable accident detection, quick emergency communication, and excellent false alarm handling, all while keeping costs low and usability high.

The chapter talks about the general system architecture, the hardware parts, and the design choices that were made when making the Smart Helmet. It also talks about how to combine sensor, processing, communication, and power management modules into one working system. Real-world working conditions are very important, such as being able to use the device in the country, being able to use it with a battery, and being able to interact with the rider in an emergency.

The goal of this chapter is to help you understand how the proposed system is set up and how each part helps the project reach its goals. The next sections give detailed explanations of system design, specifications, limitations, and implementation techniques to help with the experimental evaluation that will be talked about in later chapters.

3.2 Methods and Materials

**Methodology**: The process of development is modular and progressive. At first, the system requirements were based on safety demands, research, and how people really ride in the real world. Then, the general system architecture was planned to make sure it worked reliably, used little power, and was easy to add to.

The method for finding accidents is based on constantly watching motion characteristics using inertial sensors. The microcontroller processes acceleration and angular velocity data in real time to find strange patterns that could mean a collision is about to happen. There are set threshold levels and time intervals that help tell the difference between typical riding conditions and accidents.

When the system sees a possible accident, it goes into alert mode, which makes a sound and starts a countdown. This lets the rider turn off the warning by hand in case of a small mishap or a false alarm. If the alert isn't canceled within the time limit, the system automatically gets the rider's location and sends an emergency SMS to a set contact using a wireless communication module.

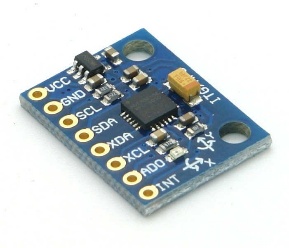
The system runs in a monitoring loop all the time to make sure it is always responsive and reliable.

**Materials**: The Smart Helmet system is implemented using the following hardware components:

* **ESP32 Development Board:**

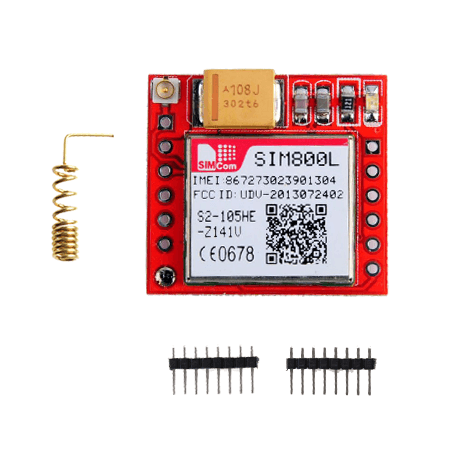


* **MPU6050 Accelerometer and Gyroscope Module:**



* **u-blox NEO-8M GPS Module:**



* **SIM800L GSM/GPRS Module:**
* **Passive Buzzer (1207 Type):**



* **Push Button (Manual Cancel Switch):**



* **Lithium-Polymer (Li-Po) Battery (3.7 V):**



* **Battery Charging and Power Management Module (TP4056):**



* **OLED Display:**



* All components are selected based on cost-effectiveness, availability in local markets, low power consumption, and suitability for real-world embedded system deployment

3.3 Design Specifications. Standards and Constraints

**Design Specifications**

**Functional Specifications:**

* The system shall continuously monitor rider motion using inertial sensors to detect potential accident conditions.
* The system shall identify severe impacts by analyzing acceleration and angular velocity data within predefined threshold limits.
* The system shall generate an audible warning and initiate a countdown period after detecting a possible accident.
* The system shall allow the rider to cancel the emergency alert during the countdown phase using a manual push button.
* The system shall automatically send an emergency SMS containing real-time GPS coordinates if the alert is not canceled.
* The system shall operate independently using a battery-powered supply.

**Technical Specifications:**

* Microcontroller: ESP32 development board
* Motion Sensor: MPU6050 (3-axis accelerometer and 3-axis gyroscope)
* GPS Module: u-blox NEO-8M
* GSM Module: SIM800L
* Power Supply: 3.7 V Li-Po battery with charging and regulation circuit
* Communication Interfaces: I²C for sensor communication, UART for GPS and GSM modules
* Alert Mechanism: Passive buzzer and push-button input
* **Display Module:** OLED display used to show battery charge level, GSM network status, GPS connectivity, and overall system readiness

**Non-Technical Specifications:**

* The system shall be compact and lightweight to ensure rider comfort.
* The system shall be affordable for deployment in developing regions.
* The system shall be easy to operate and require minimal user interaction.
* The system shall be suitable for use in rural and night-time riding conditions.

**Standards and Regulations**

* The system design follows general **embedded system safety practices** related to low-voltage electronics.
* SMS communication is implemented in accordance with **GSM communication standards** supported by SIM800L.
* Sensor interfacing and communication protocols adhere to standard **I²C and UART communication specifications**.
* Power management follows safe handling guidelines for **Li-Po battery charging and operation**.

Although no dedicated international helmet safety certification is implemented in this prototype, the electronic system is designed to operate without interfering with the structural integrity of the helmet.

**Design Constraints**

* **Power Constraint:** The system must work well even when the battery is low and the GSM module needs a lot of current.
* **Size and Weight Constraint**: All electronic parts must be small and light so that the rider doesn't feel uncomfortable.
* **Cost Constraint:** The system must use inexpensive parts to make sure it is affordable and can be used in real life.
* **Environmental Constraint:** The system must work well even when there is vibration, movement, or changes in the environment.
* **Network Dependency Constraint**: The GSM network must be available for emergency communication, however this may not always be the case in remote places.
* **False Detection Constraint:** The system must keep real accidents sensitive while minimizing false alarms.These specifications, standards, and constraints collectively guide the design decisions and ensure that the proposed Smart Helmet system meets functional requirements while remaining practical, safe, and cost-effective.

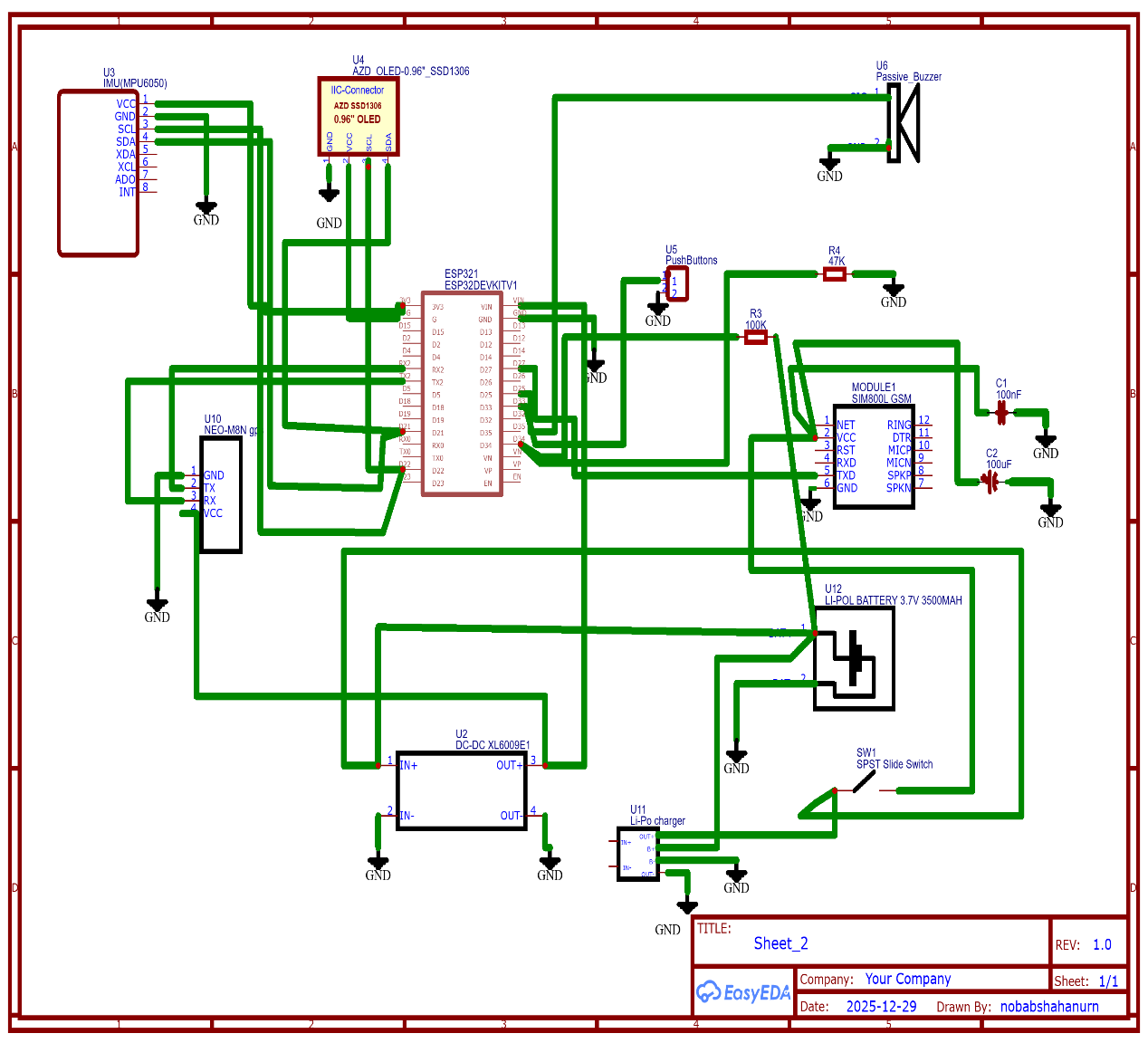


Fig. 3.1 Circuit diagram of Smart Helmet.

* 1. System Analysis

**Threshold Selection and Detection Logic:**

Accident detection in the proposed Smart Helmet system is based on a threshold-driven analysis of motion data obtained from the MPU6050 accelerometer and gyroscope. Two key parameters are monitored continuously: resultant linear acceleration (G-force) and angular velocity magnitude.

Based on extensive experimental testing under normal riding conditions, sudden maneuvers, rough road conditions, and controlled helmet drop tests, the following threshold values were selected:

* **Resultant Acceleration Threshold:** ≥ **2.2 g**
* **Angular Velocity Threshold:** ≥ **190 °/s**
* **Validation Time Window:** ≥ **120 ms**

An accident is detected only when both the acceleration and angular velocity thresholds are exceeded simultaneously for at least the specified validation time window. This dual-parameter and time-based approach ensures reliable accident detection while minimizing false alarms caused by short-duration disturbances such as sudden braking, speed breakers, or accidental helmet drops.

The detection logic can be summarized as:

If (Resultant Acceleration ≥ 2.2 g) AND

(Angular Velocity ≥ 190 °/s)

sustained for ≥ 120 ms

Then

Accident Detected

Else

No Accident

3.5 Experimental Setup

**Hardware Setup**

The full Smart Helmet prototype was put together by putting together the ESP32 development board, MPU6050 motion sensor, u-blox NEO-8M GPS module, SIM800L GSM module, passive buzzer, manual cancel button, and a 3.7 V Li-Po battery with power management circuitry. We put all the parts on a prototype board and powered them with the battery to make it work like a real helmet.

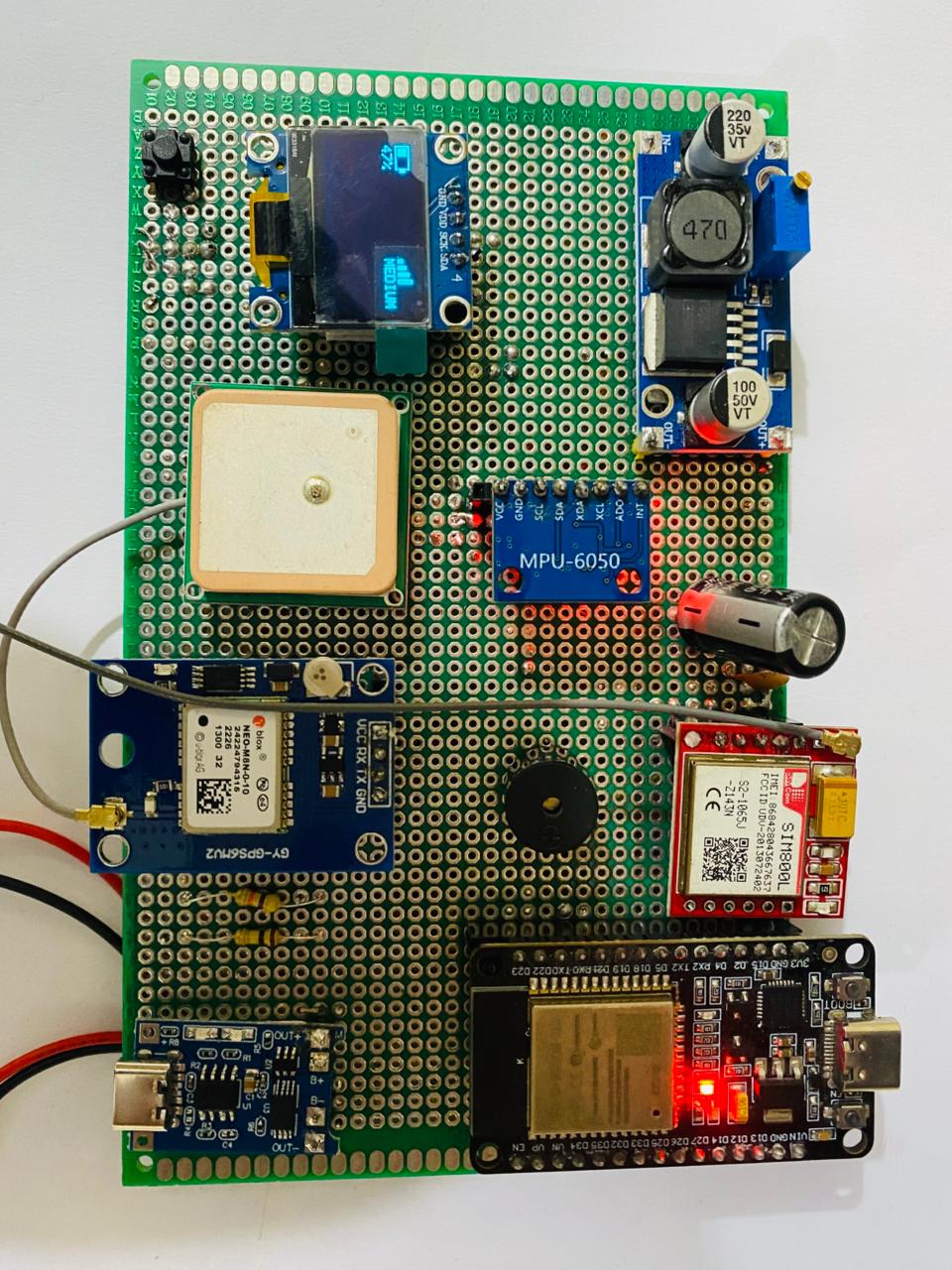
  
The ESP32 and the MPU6050 sensor talked to each other using the I²C communication protocol. The GPS and GSM modules, on the other hand, talked to each other through independent UART interfaces. To keep the system running well, especially during GSM transmission, which needs a lot of peak current, the voltage was properly regulated and the ground was properly connected. The ESP32's digital input/output pins were used to link the buzzer and cancel button.

Fig. 3.5 Hardware setup of Smart Helmet.

**Experimental Procedure**

The experiment was conducted in multiple stages:

1. **Sensor Verification Test:**

The MPU6050 sensor was first tested to make sure it could correctly get acceleration and gyroscope data when the object was moving. Through serial monitor, real-time sensor readings were seen.

1. **Accident Detection Test:**

We manually reproduced sudden impacts and quick movements to make the conditions look like an accident. We tested the threshold-based detection algorithm to make sure that the system correctly detects unusual motion patterns while disregarding conventional riding actions. We rode the motorcycle on normal roads and over speed breakers, applied sudden braking, and collected this data to determine the threshold values.

1. **False Alarm Handling Test:**

We made up scenarios where the rider stays awake and the impact is small. In these situations, the buzzer alert and countdown mechanism were used, and the manual cancel button was clicked to make sure that the emergency SMS transmission was stopped.

1. **Emergency Alert Test:**

The system was tested to finish the countdown period without canceling it for real accident situations. The GPS module got the rider's location coordinates, and the SIM800L module sent an emergency SMS with the rider's location to a number that had been set up ahead of time.

1. **Power and Stability Test:**

The system was tested under battery-powered operation to observe voltage stability, GSM power surges, and continuous monitoring performance.

**Result Observation and Analysis**

The behavior of the system was tracked using serial output logs, SMS message delivery confirmation, and the timing of buzzer alerts. We looked at detection accuracy by comparing sensor readings during normal movement and fake accidents. We did several test runs to make sure that SMS delivery and GPS accuracy worked well every time.

3.6 Summary

This chapter explained how the Smart Helmet system was designed, how it was built, and how it was put into action. It talked about the general system architecture, the hardware parts that were chosen, and the ways that sensor, processing, communication, and power management units were brought together into a single working design.

To explain important design choices, they talked about the design parameters, relevant standards, and real-world limitations. The experimental setting explained how the implementation process and testing environment were used to check the system's performance, including how well it could find accidents, send emergency alerts, and deal with false alarms.

The results of this chapter lay the groundwork for performance evaluation and result analysis, which will be explored in the next chapter.

Chapter 4

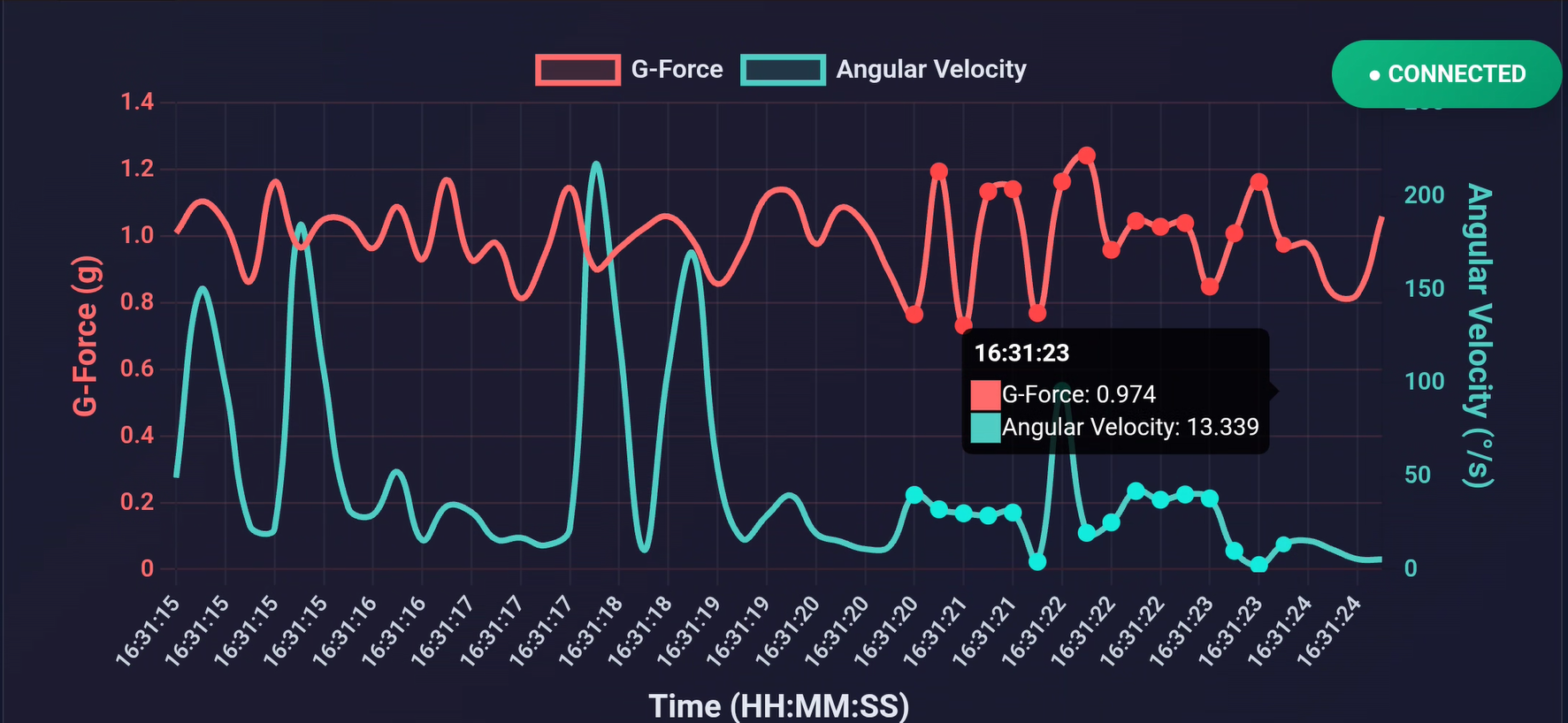
Results and Discussions

4.1 Results

This section presents the experimental results obtained from the MPU6050 accelerometer and gyroscope during real-time riding conditions. The data were collected under different scenarios such as normal riding, sudden acceleration, sudden braking, speed breakers, and rough road conditions. The graphs represent the resultant acceleration (G-force) and angular velocity variations over time.

**4.1.1 Normal Riding Condition:**

During normal riding conditions, the recorded G-force remained close to 1 g, which corresponds to gravitational acceleration. Angular velocity values showed fluctuations due to minor steering and road irregularities.

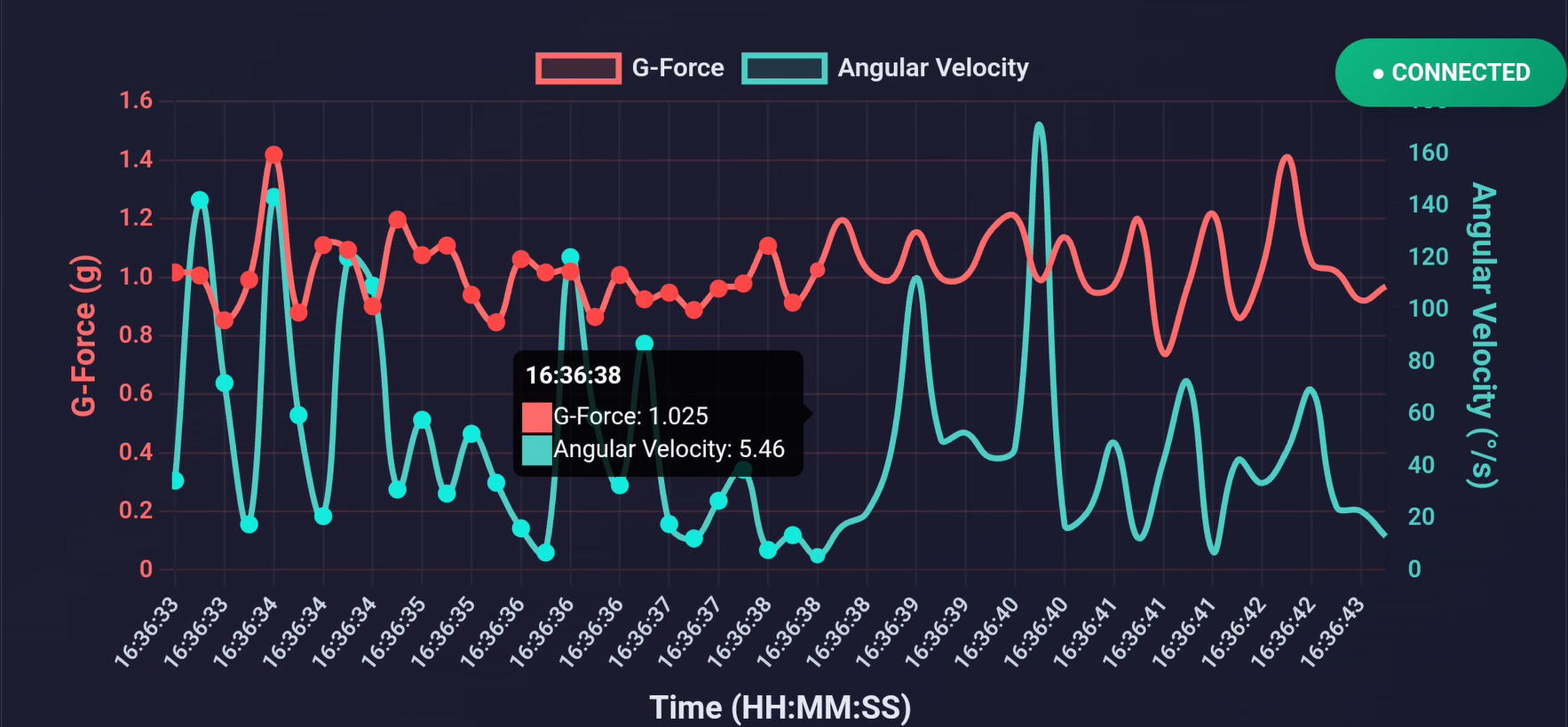


**Figure 4.1:** G-Force and Angular Velocity during Normal Riding Condition

**Observation:**  
No abnormal spikes were observed, indicating stable riding behavior without accident-like conditions.

**4.1.2 Sudden Acceleration:**

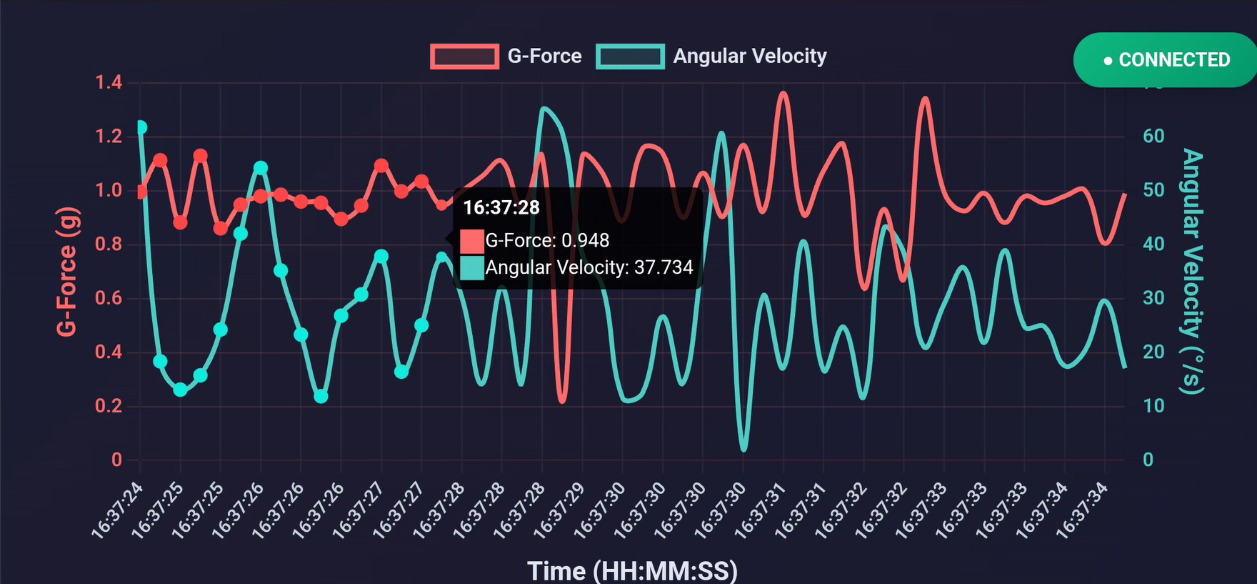
When sudden acceleration was applied, a noticeable increase in G-force was observed. However, angular velocity remained within a moderate range.



**Figure 4.2:** Sensor Response during Sudden Acceleration

**Observation:**  
Although acceleration increased briefly, the combined threshold conditions for crash detection were not satisfied, preventing false accident detection.

**4.1.3 Sudden Braking:**

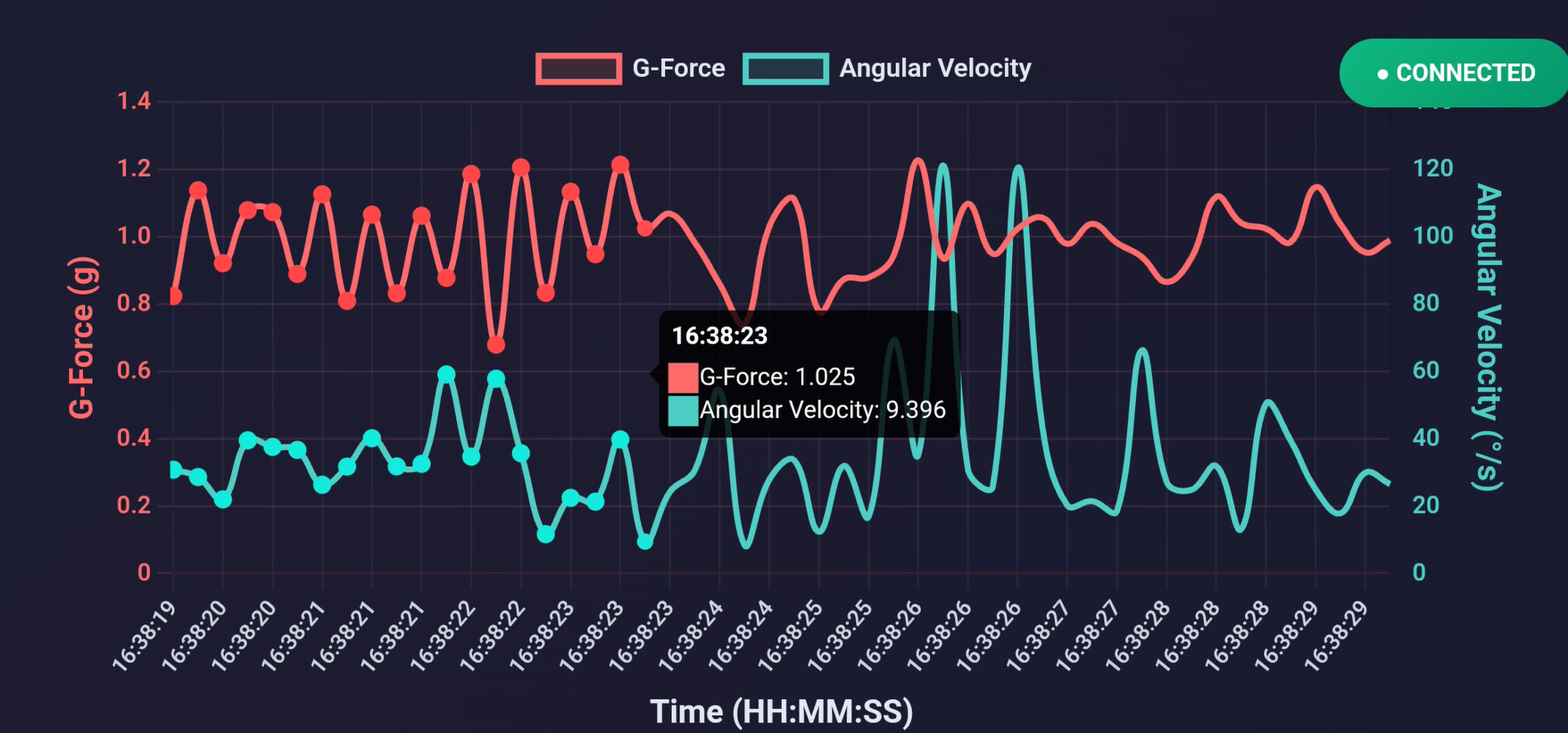
Sudden braking caused a short-duration drop and spike in G-force values along with moderate angular velocity changes.

**Figure 4.3:** Sensor Response during Sudden Braking

**Observation:**  
The system correctly identified this as a non-accident event, demonstrating effective discrimination between braking and crash conditions.

**4.1.4 Speed Breaker and Rough Road Condition**

While crossing speed breakers and uneven road surfaces, short spikes in acceleration were observed. Angular velocity fluctuations were also present due to vertical motion.

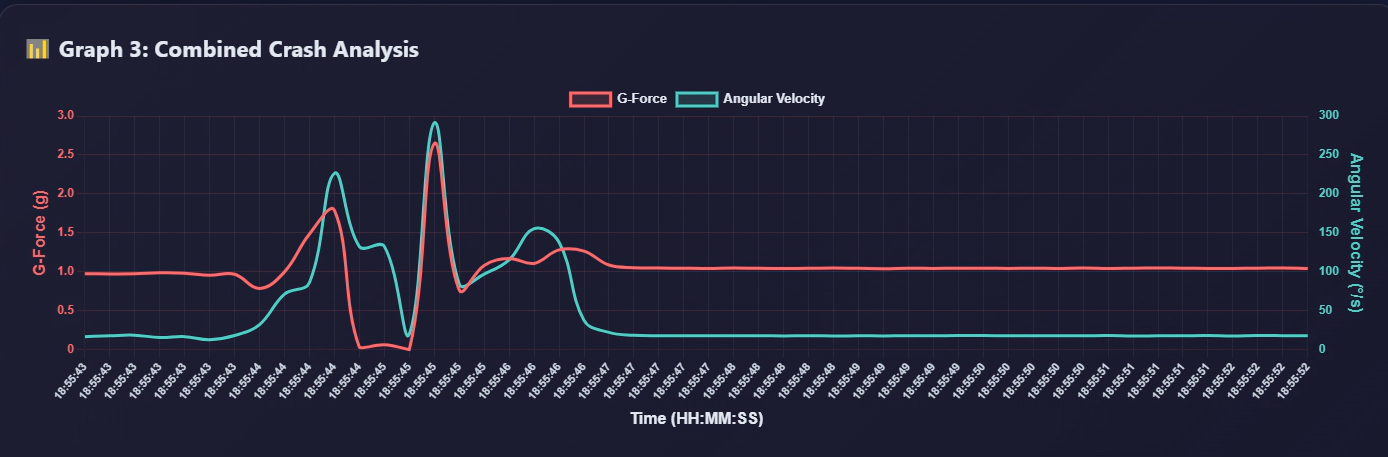


**Figure 4.4:** Sensor Response on Speed Breaker and Rough Road

**Observation:**  
Despite noticeable disturbances, the motion does not exceed the defined crash thresholds for a sustained duration, demonstrating effective rejection of false alarms in rough road conditions.

**4.1.6 Controlled Helmet Drop Test (Non-Injury Impact)**

Figure **4.5** illustrates the MPU6050 sensor response during a controlled helmet drop test on a soft surface. A short-duration acceleration spike occurs at the moment of impact.

Figure 4.6: MPU6050 Sensor Response during Controlled Helmet Drop Test on Soft Surface

Observation: Although a short-duration acceleration spike was detected, the system did not classify this event as an accident because the combined threshold condition of high acceleration and high angular velocity within a defined time window was not satisfied. This demonstrates the effectiveness of the proposed algorithm in preventing false accident detection during non-critical impact events.

**4.1.2 False Alarm Handling Results**

To evaluate false alarm handling, minor impact and low-speed fall scenarios were simulated. In these cases, the system activated the buzzer and initiated the countdown period. When the cancel button was pressed within the countdown duration, the emergency alert process was successfully terminated and no SMS was sent.

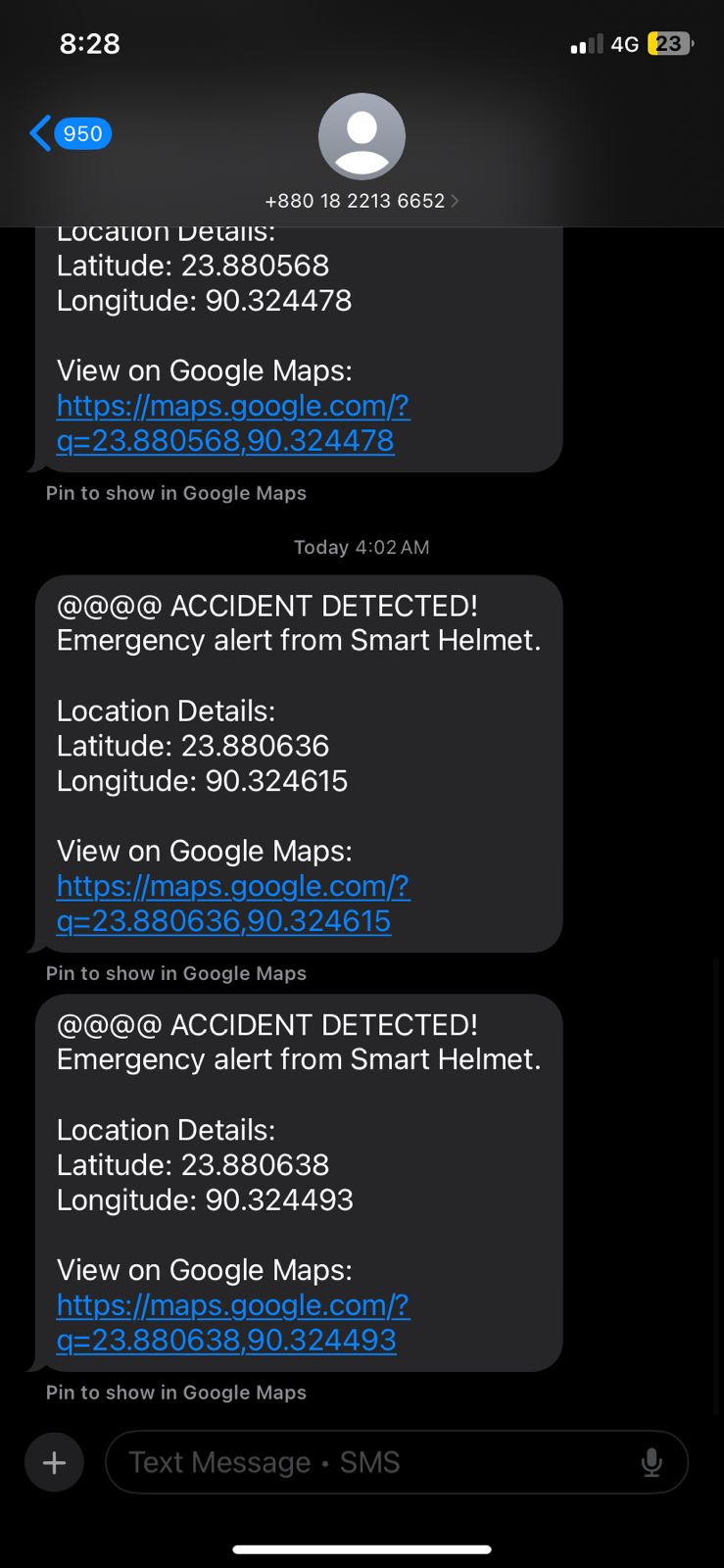
**Table 4.1:** *False alarm handling test results*

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Scenario** | **Accident Detected** | **Cancel Button Pressed** | **SMS Sent** |
| Minor impact | Yes | Yes | No |
| Low-speed fall | Yes | Yes | No |
| Normal riding | No | Not required | No |
| Severe impact | Yes | No | Yes |

The results confirm that the manual cancellation mechanism effectively prevents unnecessary emergency alerts during non-critical situations.

**4.1.3 Emergency Alert and GPS Results**

For confirmed accident scenarios where the alert was not canceled, the system successfully acquired GPS coordinates and transmitted an emergency SMS using the GSM module.

****

**Fig. 4.3:** *Sample emergency SMS received with GPS coordinates*

The GPS coordinates provided accurate location information, enabling easy identification of the accident site through map services.

4.2 Discussions

The testing results show that the Smart Helmet system satisfies its main design goals by properly identifying accident conditions, reducing false alarms, and making sure that emergency communication is trustworthy. We tested the system in a number of real-world and controlled situations, and the results show that the proposed approach is strong.  
  
The performance of accident detection shows that using both accelerometer and gyroscope data together makes detection much more reliable. Normal riding situations, such as abrupt acceleration, braking, and crossing bumpy roads, did not set off the accident detection system. However, simulated accident situations that went above set threshold values were always recognized. This shows that multi-sensor fusion works better than single-sensor-based methods because it makes the system less sensitive to vibrations from the road and short, non-critical motion disturbances.

One of the best things about the system was how well it handled false alarms. Controlled helmet drop tests and modest low-speed collisions caused transient bursts of motion that briefly went beyond magnitude thresholds but didn't last long enough to meet the time-based validation condition. Because of this, emergency alarms didn't go off. Also, the addition of an audible warning and a cancellation button that the user controls lets conscientious riders stop unneeded emergency alerts. This solution with a human in the loop fixes a big problem with many smart helmet systems that have too many false alerts, which makes users less likely to trust the system.

The chosen threshold values resultant acceleration of 2.2 g, angular velocity of 190 °/s, and a validation time window of 120ms were established through experimental observations and corroborated by crash biomechanics literature. Experimental findings indicated that non-accidental occurrences generally produce motion spikes lasting around 100 ms, whereas actual crash dynamics entail prolonged anomalous motion exceeding this timeframe. So, the specified time limit clearly separates real accident conditions from temporary disturbances and makes it easier for emergency responders to get there faster.

The results of emergency communication tests show that the system works well. The successful sending of SMS notifications with correct GPS locations shows that the GSM and GPS modules work well together. This capability is especially useful in rural, nighttime, or low-traffic areas where help from a person may not be accessible right away. The results show that using GSM for emergency alerts in real life is a useful and effective way to do so.

Observations of power stability show how important it is for wearable safety equipment to have strong power management. Even though there were short voltage drops during GSM transmission because of increased current demand, the system kept working without any resets or breaks in service. This shows that the design for the power supply and regulation is good for real-world use with batteries.

The system works well, however there are still some problems with it. Emergency communication relies on the availability of the GSM network, which may not always be reliable in remote regions. It may take longer to get a GPS signal indoors or in places where there are obstacles. These restrictions show that there is room for improvement in the future, such as support for communication over many networks or hybrid positioning systems.

The overall discussion shows that the suggested Smart Helmet system is a good and practical option since it combines precision, less false alarms, user intervention, and low cost. The trial results show that the system is good for real-world motorcycle safety uses and could cut down on the time it takes for emergency responders to get to the scene of a serious collision.

Chapter 5

Project Management

5.1 Task, Schedule and Milestones

Good project management was necessary to make sure that the Smart Helmet system was finished on time and under budget. The project was carried out in an organized way by breaking down the whole job into a series of clear tasks and milestones. This method made it possible to keep an eye on progress at all times and finish each phase on time.

The project started with figuring out what was needed and making plans. This is where the problem was found, the necessary literature was read, and the final system requirements were set. This part laid the groundwork for the next steps in design and development, and it was finished on time for the project.

The next step was to design the system, which meant making the system architecture, block diagram, and circuit schematic. During this stage, the final decisions about component selection, power management, and communication interfaces were made.

After the design process, the hardware was put together by connecting and assembling all of the electronic parts. We reached this milestone when we checked each of the following: steady sensor readings, GPS communication, and GSM transmission.   
At the same time, software was developed to process sensor data, find accidents, handle false alarms, and send emergency alerts. This milestone was reached when fundamental algorithms were successfully run and the system behaved in a stable way.

The system integration and testing process made sure that the hardware and software parts worked together properly. We ran a lot of different tests to make sure that the accident detection worked, the alerts went off at the right time, the GPS data was collected correctly, and the GSM connectivity was reliable.   
The last step was documentation and reporting, which involved analyzing the results, writing reports, and making presentations. Finishing this step meant that the project was a success.

The project went mostly according to plan, with each milestone being reached in order. During testing, small changes were made to make the system more stable and reliable. This showed that the project team was able to manage tasks and plan ahead throughout the project's life cycle.

5.2 Resources and Cost Management

Managing resources and costs well was an important part of the Smart Helmet project to make sure it was finished on time and within budget. The project goals were met without spending too much money by carefully planning and using both human and technical resources.

**Human Resources:**

The project team members were in charge of designing the system, putting together the hardware, writing the software, testing it, and writing the documentation. The project supervisor helped with important parts of the project, like reviewing the design, validating the tests, and writing the report. Properly dividing up tasks among team members helped them get more done and cut down on development time.

**Hardware and Technical Resources:**

The main technical resources were tools for developing embedded systems, electronic parts, and labs. We chose parts like the ESP32 development board, MPU6050 sensor, GPS module, GSM module, and power management circuits based on how easy they were to get, how cheap they were, and how well they worked together. To keep expenses down, we used existing lab equipment like computers, power supplies, and multimeters.

**Software Resources:**

The whole project utilizes open-source and free software platforms. The Arduino IDE was used for programming and development, and serial monitoring and data visualization tools were employed for testing and fixing bugs. This method got rid of license fees and made it easier to change things during development.

**Cost Management:**

The project was planned with a lot of focus on how to do it cheaply. The choice of components was based on how much they cost and how easy they were to find in the local market. To save money, they used reusable parts and shared lab equipment whenever they could. The total cost of the project stayed under the budget, which shows good financial planning.

Keeping an eye on spending and resource use on a regular basis helped keep costs from going over budget. Instead than buying more things, we used design optimization to meet any new needs that came up during development.

In general, smart helmet project was finished successfully and on a budget since resources were used wisely and costs were kept in check.

**Equipment and Component Cost:**

The major cost of the project was associated with electronic components and development materials. Table 5.1 presents the list of equipment used along with their approximate costs.

**Table 5.1: Equipment and Component Cost Breakdown**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Sl. No.** | **Equipment / Component** | **Quantity** | **Unit Cost (BDT)** | **Total Cost (BDT)** | | 1 | ESP32 Development Board | 1 | 600 | 600 | | 2 | MPU6050 Accelerometer & Gyroscope | 1 | 350 | 350 | | 3 | SIM800L GSM/GPRS Module | 1 | 450 | 450 | | 4 | NEO-8M GPS Module | 1 | 1500 | 1500 | | 5 | Li-Po Battery (3.7 V) | 2 | 500 | 1000 | | 6 | Battery Charging & Power Module | 1 | 40 | 40 | | 7 | Passive Buzzer | 1 | 40 | 40 | | 8 | Push Button Switch | 1 | 10 | 10 | | 9 | Connecting Wires & PCB / Breadboard | 1 | 1000 | 1000 | | 10 | Helmet & Mechanical Setup | 1 | 800 | 800 | |

**Total Estimated Project Cost = 5,110 BDT**

(Note: Costs are based on local market prices.)

5.3 Lesson Learned

Participating in the initiative advanced our technical skills and managerial capabilities. One of the most valuable insights obtained was the essential significance of methodically designing and constructing the system in modular components. Debugging, testing, and system integration were more efficient and effective when the

system was decomposed into smaller functional elements.   
  
The initiative improved hands-on expertise in embedded systems, with a focus on sensor interfacing, real-time data processing, and serial communication. The deployment of motion sensors, GPS, and GSM modules deepened my comprehension of the integration between hardware and software, especially in the execution of time-critical duties. Addressing power management challenges, especially when the GSM module demanded significant current, highlighted the importance of a robust power

architecture in battery-powered devices.

Another vital insight I acquired was the significance of effectively managing false alarms in safety-critical systems. Creating a cancellation system that users can control underscores the importance of accounting for both technological accuracy and human considerations. This experience reaffirmed the notion that system reliability relies not solely on sensor accuracy but also on user engagement and operational efficiency.   
  
From a project management standpoint, meeting deadlines requires efficient task assignment, meticulous time management, and regular progress assessments. The experience improved writing and reporting abilities, emphasizing the importance of precise communication in technical terminology.

This undertaking improved problem-solving capabilities, practical engineering expertise, and collaborative experience, all of which will be advantageous for future academic pursuits and professional engineering initiatives.

Chapter 6

Impact assessment of the project

6.1 Economical, Societal and Global Impact

The proposed Smart Helmet system has the potential to create significant economical, societal, and global impacts by improving road safety and reducing the consequences of motorcycle accidents.

**Economical Impact:**

From an economic point of view, the Smart Helmet can lower the cost of medical care for major accident injuries by making it easier for emergency responders to get there faster. Timely medical intervention sometimes lowers the costs of staying in the hospital and the costs of long-term rehabilitation. The suggested method is affordable, which makes it available to a wide range of users and makes it easy for many people to utilize without putting a pressure on their finances. If accidents are less severe, people and families may lose less productivity, which would be good for the economy as a whole.

**Societal Impact:**

The Smart Helmet has social effects that mostly have to do with making riders safer and making it easier for emergency responders to accomplish their jobs. The system can save lives by automatically finding incidents and calling emergency contacts. This is especially helpful in remote and low-traffic areas where help is sometimes delayed. A false alarm canceling system increases user trust and encourages safe use. The strategy promotes a culture of safety awareness among motorcycle riders and helps reduce the number of deaths from road accidents.

**Global Impact:**

Car accidents are a big problem for public health all throughout the world, but they are especially bad in poor countries. The Smart Helmet that has been suggested satisfies global road safety targets by using cheap technology to help individuals deal with problems that come up after an accident. The system can be utilized in many places throughout the world since it uses standard communication infrastructure and parts that are easy to find. The project helps the world work to lower the number of deaths and injuries caused by traffic by getting people to move about more safely.   
The Smart Helmet project is a useful, cheap, and scalable way to aid the economy, make society safer, and reach worldwide road safety goals.

6.2 Environmental and Ethical Issues

It is necessary to think about the environmental and ethical issues of the proposed Smart Helmet system to make sure that the project helps with sustainable development and responsible engineering.

Problems with the environment: The Smart Helmet system is made to run on batteries and use very little electricity, which helps save energy while it is in use. Using a rechargeable lithium-polymer battery makes you less reliant on single-use batteries, which cuts down on electronic waste. The method also requires fewer electronic parts, which cuts down on the amount of materials used and the damage done to the environment during production.

But electronic waste that is made when a product is no longer useful is still a problem. To protect the environment, it is important to properly throw away and recycle electrical parts and batteries. The design makes it easy to use standard, reusable parts that may be replaced or recycled on their own, instead of throwing away the whole system.

Ethical Issues: The Smart Helmet technology was made with the main goal of making people safer and saving lives. The system only sends out emergency information when it's really needed, and there is a manual cancelation option to give the rider more control and stop unwanted emergency alerts.

Privacy is another major moral issue. The system only collects and sends location data for emergencies and only after it sees a possible mishap. There is no ongoing surveillance or data storage, which lowers the chance of personal information being misused.

The project follows ethical engineering rules by putting safety first, reducing harm, and making sure that technology is used responsibly. The Smart Helmet system strikes a good mix between making technology better, being responsible for the environment, and being morally responsible.

6.3 Utilization of Existing Standards or Codes

The design of the proposed Smart Helmet system follows the right engineering standards and best practices to make sure it is safe, reliable, and works with other systems. Even though the project is only a prototype, the design and development process took into account a number of current standards and codes.

The system's electronic design follows conventional safety standards for low-voltage embedded systems, making sure that all parts work safely within their rated voltage and current limits. To keep users safe, the power management circuit follows prescribed rules for charging and protecting lithium-polymer (Li-Po) batteries, such as preventing overcharging and short-circuiting.

Standardized protocols are used to make sure that the different parts of the system can talk to each other. The motion sensor talks to the microcontroller using the I²C communication protocol, while the GPS and GSM modules talk to each other using UART serial communication. Both of these are conventional ways that businesses use to talk to each other. Emergency SMS delivery follows GSM communication standards, which makes sure that it works with all cell networks.

From a software point of view, the system development follows typical embedded programming principles, such as modular coding, managing errors, and safe initialization processes. These practices make the system more reliable and easier to keep up with.

The electrical system is made to work on its own without changing the helmet's structural integrity or safety function, even if formal helmet safety certifications such impact protection standards are not part of this prototype. Future versions may include following accepted helmet safety standards and rules for electromagnetic compatibility.

In general, using current standards and rules makes sure that the proposed Smart Helmet system is safe, reliable, and ready for more development and use in the real world.

6.4 Other Concerns

There were a lot of additional things that were thought about when making the Smart Helmet system, in addition to the economic, environmental, ethical, and standards-related issues that were already mentioned.

One key thing to think about is how available the network is. The emergency alert system needs GSM network coverage, which might not be available or might not perform well in rural or distant places. In these situations, it may take longer to send emergency messages. This limitation shows how important it is to provide features in the future, such support for several networks or other ways to communicate.

Another worry is whether GPS signals will be available. To get an accurate location, you need to be able to see the satellites clearly. GPS may not work as well in places like tunnels, indoors, or cities with a lot of buildings close together. This doesn't change how the system works, but it could change how accurate location information is in some cases.

It's also crucial to think about how comfortable and easy to use the product is for the user. The electronic parts make the helmet heavier, which could make it less comfortable on extended rides. To reduce discomfort, parts must be carefully placed and packed tightly.

There are also worries about system maintenance and durability. For the device to keep working, the battery has to be charged regularly. Over time, vibration, dust, or moisture can harm the electrical parts. For long-term reliability, it's important to design the enclosure correctly and do regular maintenance.

Chapter 7

Conclusions and Recommendations

7.1 Conclusions

Along with the economic, environmental, ethical, and standards-related problems already discussed, the development of the Smart Helmet system also dealt with other issues.

One of the most important things is network access. The emergency alert system needs the GSM network to work, but this network may not be available in isolated or rural regions. In certain situations, it may take longer to send out emergency notifications. This constraint shows that improvements are needed in the future, including making it function with more than one network or adding alternative communication methods.

Another problem is that GPS signals aren't always easy to get to. To find the right position, you need to be able to see satellites clearly. GPS may not work as well indoors, in tunnels, or in cities with a lot of people. This doesn't affect how the system works, but it could change how accurate location data is in some cases.  
User comfort and ease of use are quite important. The electronic parts make the helmet heavier, which could make it less comfortable to wear for lengthy periods of time. To make things more comfortable, it's important to carefully organize and bundle the parts.

Maintaining the system and making it last longer are harder tasks. For ongoing operation, it is important to have a reliable way to charge the battery. Over time, vibration, dust, or moisture can damage electrical parts. For reliability to last, the enclosure must be well-designed and kept up with regularly.

7.2 New Skills and Experiences Learned

The Smart Helmet project gave me several chances to improve my technical and professional skills. One of the most important results of this project was the chance to work on designing and building embedded systems. Testing and troubleshooting over and over again made me better at programming microcontrollers, connecting them to sensors, and processing data in real time.

Using motion sensors, GPS, and GSM modules let me learn more about how hardware and software work together and how serial communication protocols work. The research helped people learn more about how to manage power in battery-powered devices, especially when it comes to meeting high current needs and keeping the system stable.

The project helped people learn how to solve problems and use technology. To deal with false alarms, sensor noise, and communication delays in real-world situations, we had to use analytical thinking and make changes to the design over and over again. These experiences made it clear how important it is to think about how users interact with the system and the limits that come with building it.

The initiative made it easier for people to work together, helped them organize their time better, and increased the quality of technical documentation. Improving project management by coordinating well, achieving deadlines, and writing well-organized reports. I've learned enough from these skills and experiences to do academic research and professional engineering work in the future.

7.3 Future Recommendations

The proposed Smart Helmet system works well, but there are a number of improvements that might be made to make it more useful, reliable, and practical.

Future versions of the system may include advanced accident detection techniques, such as machine learning models, to improve accuracy and reduce false alarms in a wider range of riding circumstances. Adding extra sensors, such heart-rate or pressure monitors, could help doctors better assess a rider's condition after an accident.  
  
To make communication easier, looking into multi-network compatibility, such LTE, NB-IoT, or internet-based messaging, could make emergency warnings more reliable in areas where GSM coverage is poor. Cloud-based data storage and mobile app integration can make it easier to keep an eye on things and analyze data in real time.   
  
Better battery management and power optimization can make the system last longer and work better. Using small, weatherproof enclosures would make them last longer and better able to handle the elements.

In the future, efforts may focus on following helmet safety and electromagnetic rules, as well as doing a lot of field testing. These improvements would make it easier to sell the Smart Helmet system and get a lot of people to use it as a reliable way to keep riders safe.

References

[1] World Health Organization, Global Status Report on Road Safety, WHO Press, Geneva, Switzerland, 2018.

[2] A. Al-Hourani and K. Gomez, “Smart Wearable Safety Systems: A Review,” IEEE Access, vol. 8, pp. 212345 212360, 2020.

[3] P. Ajay, M. Ramesh, and K. Kumar, “Smart Helmet for Safe Riding,” International Journal of Engineering Research & Technology, vol. 6, no. 4, pp. 45 49, 2017.

[4] S. Kumar and R. Verma, “Accident Detection Using Accelerometer and Microcontroller,” International Journal of Computer Applications, vol. 104, no. 2, pp. 25 28, 2018.

[5] R. Patel, A. Shah, and V. Mehta, “Motorcycle Accident Detection Using MEMS Sensors,” IEEE International Conference on Smart Technologies, pp. 112 117, 2019.

[6] M. Sharma, P. Singh, and A. Gupta, “GPS and GSM Based Smart Helmet for Accident Detection,” International Conference on Communication and Signal Processing, pp. 978 982, 2020.

[7] A. Singh and N. Jain, “False Alert Reduction Techniques in Smart Helmet Systems,” IEEE Access, vol. 9, pp. 55621 55630, 2021.

[8] H. Rahman, S. Islam, and T. Ahmed, “Low-Cost IoT-Based Smart Helmet for Rider Safety,” International Journal of Advanced Electronics and Communication Systems, vol. 5, no. 3, pp. 61 66, 20

[9] S. S. Shinde and R. S. Deshmukh, “IoT Based Smart Helmet for Accident Detection and Location Tracking,” *International Journal of Engineering Research and Technology (IJERT)*, vol. 9, no. 6, pp. 312–316, 2020.

[10] N. Gupta, R. Kumar, and S. Verma, “Real-Time Accident Detection and Emergency Alert System Using IoT,” *International Conference on Intelligent Systems and Control (ISCO)*, pp. 145–150, 2019.

[11] F. A. Berg, “Motorcycle Impacts into Roadside Barriers: Real-World Crash Test,”

*in Proc. 19th Int. Tech. Conf. on the Enhanced Safety of Vehicles (ESV),*

NHTSA, 2005.

Appendix A

Turnitin Report

*Include here the 1st page of Turnitin Report*

Every supervisor has his/her own Turnitin account. If not, then supervisors are requested to get the account from Library as soon as possible.

Appendix B

Complex Engineering Problem Solving and Engineering Activities

|  |  |  |
| --- | --- | --- |
| Complex Engineering Problems (P) Solving | | |
|  | Attributes | Statement from students |
| P1 | Depth of knowledge required | This project required knowledge from multiple engineering areas, including embedded systems, sensors, communication systems, and power management. We applied concepts learned from microcontrollers, sensors and instrumentation, and communication engineering courses to design and implement the system successfully. |
| P2 | Range of conflicting requirements | While designing the system, we faced conflicting requirements such as ensuring accurate accident detection while avoiding false alarms, maintaining low power consumption while using a high-current GSM module, and keeping the system compact while ensuring reliability. These issues were resolved through careful design decisions and testing. |
| P3 | Depth of analysis required | Detailed analysis was needed to determine suitable sensor threshold values, handle sensor noise, and ensure stable operation during GSM transmission. Several test cases were performed to analyze system behavior under different conditions. |
| P4 | Familiarity of issues | Although we were familiar with individual components from previous coursework, integrating all components into a single wearable safety system introduced new challenges that required additional learning, experimentation, and troubleshooting. |
| P5 | Extent of applicable codes | The project followed standard embedded system design practices, serial communication protocols (I²C and UART), GSM communication guidelines, and basic safety rules for lithium-polymer battery usage. |
| P6 | Extent of stakeholder involvement and conflicting requirements | The primary stakeholders are motorcycle riders and their emergency contacts. A key concern was sending emergency alerts quickly without causing unnecessary panic. This was addressed by adding a manual cancel option for false or minor accident cases. |
| P7 | Interdependence | The system modules are closely connected with each other. Sensor data processing, power supply, GPS location tracking, and GSM communication must work together. A problem in one part directly affects the overall system performance. |

|  |  |  |
| --- | --- | --- |
| Complex Engineering Problems (P) Solving | | |
|  | Attributes | Statement from students |
| A1 | Range of resources | The project employed a lot of different things, like hardware parts, programming tools, datasheets, lab instruments, and online technical material. These materials were important for both development and testing. |
| A2 | Level of interaction | There was a lot of interaction between the hardware and software parts. To make sure everything worked right, the sensors, microcontroller, GPS, GSM module, and power system had to work together all the time. |
| A3 | Innovation | The project introduces a practical improvement by combining automatic accident detection with a user-controlled cancellation feature. This makes the system more reliable and user-friendly compared to many existing solutions. |
| A4 | Consequences of society and environment | The system makes society better by making it safer for riders and speeding up emergency response times. Using a rechargeable battery and low-power parts helps reduce the impact on the environment. |
| A5 | Familiarity | Some of the tasks were similar to what we learned in school, but using them on a real-world, safety-critical wearable device needed new abilities, hands-on experience, and problem-solving that went beyond what we did in the lab. |

Appendix C

Program Code

*/\**

*\* =====================================================*

*\* Smart Helmet with Sensor-Based Crash Detection and Automated Emergency Communication*

*\* =====================================================*

*\**

*\* This system detects motorcycle/bike accidents using MPU6050*

*\* sensor and automatically sends emergency SMS with GPS location*

*\* via SIM800L GSM module.*

*\* PASSIVE BUZZER FEATURES:*

*\* 10-second warning tone on crash detection*

*\* 3-second beep pattern before SMS*

*\* SMS can be aborted during beep pattern by pressing button*

*\* Version: 1.0*

*\* Date: 2026*

*\*/*

*#include <Wire.h>*

*#include <TinyGPSPlus.h>*

*#include <Adafruit\_GFX.h>*

*#include <Adafruit\_SSD1306.h>*

*/\* ================= OLED DISPLAY ================= \*/*

*#define SCREEN\_WIDTH 128*

*#define SCREEN\_HEIGHT 64*

*#define OLED\_RESET -1*

*Adafruit\_SSD1306 display(SCREEN\_WIDTH, SCREEN\_HEIGHT, &Wire, OLED\_RESET);*

*/\* ================= MPU6050 ================= \*/*

*#define MPU\_ADDR 0x68 // I2C address of MPU6050*

*int16\_t AcX, AcY, AcZ, GyX, GyY, GyZ; // Raw sensor values*

*float gyroBiasX = 0, gyroBiasY = 0, gyroBiasZ = 0; // Calibration offsets*

*/\*===========Battery===========\*/*

*#define BATTERY\_PIN 34*

*/\* ================= GPS ================= \*/*

*TinyGPSPlus gps; // GPS parser object*

*HardwareSerial gpsSerial(2); // Use UART2 for GPS communication*

*#define GPS\_RX 16 // ESP32 RX pin connected to GPS TX*

*#define GPS\_TX 17 // ESP32 TX pin connected to GPS RX*

*/\* ================= SIM800L ================= \*/*

*HardwareSerial sim800(1); // Use UART1 for SIM800L communication*

*#define SIM\_RX 26 // ESP32 RX pin connected to SIM800L TX*

*#define SIM\_TX 27 // ESP32 TX pin connected to SIM800L RX*

*const char PHONE\_NUMBER[] = "+8801747744721";*

*/\* ================= PASSIVE BUZZER & BUTTON ================= \*/*

*#define BUZZER\_PIN 25 // Passive buzzer for alert*

*#define CANCEL\_BTN 33 // Push button to cancel false alarms*

*/\* ================= MUSICAL NOTES FOR PASSIVE BUZZER ================= \*/*

*#define NOTE\_C5 523*

*#define NOTE\_E5 659*

*#define NOTE\_G5 784*

*#define NOTE\_A5 880*

*/\* ================= THRESHOLDS ================= \*/*

*// These values are tuned for motorcycle/bike crash detection*

*// Adjust based on testing in real conditions*

*#define ACC\_THRESHOLD 1.2 // g-force (1g = normal gravity, 2.5g = significant impact)*

*#define GYRO\_THRESHOLD 80.0 // degrees per second (rapid rotation indicates fall)*

*#define IMPACT\_TIME\_MS 30 // milliseconds (sustained impact to filter vibrations)*

*/\* ================= BUZZER TIMING ================= \*/*

*#define WARNING\_TONE\_DURATION 10000 // 10 seconds warning tone*

*#define BEEP\_PATTERN\_DURATION 3000 // 3 seconds beep pattern*

*#define BEEP\_CYCLE\_TIME\_MS 400 // 400ms per beep cycle (on + off)*

*#define BEEP\_ON\_TIME\_MS 200 // 200ms beep on duration*

*/\* ================= STATE VARIABLES ================= \*/*

*bool crashDetected = false; // Flag: crash has been confirmed*

*bool smsSent = false; // Flag: emergency SMS has been sent*

*bool smsAborted = false; // Flag: SMS sending was aborted by user*

*unsigned long impactStart = 0; // Timestamp when impact first detected*

*/\* ================= BUZZER STATE MACHINE ================= \*/*

*enum BuzzerState {*

*BUZZER\_IDLE, // No sound*

*BUZZER\_WARNING\_TONE, // 10-second continuous warning*

*BUZZER\_BEEP\_PATTERN, // 3-second beep pattern*

*BUZZER\_SILENT // After SMS sent or cancelled*

*};*

*BuzzerState buzzerState = BUZZER\_IDLE;*

*unsigned long warningToneStart = 0;*

*unsigned long beepPatternStart = 0;*

*/\* ================= OLED DISPLAY STATE ================= \*/*

*int signalStrength = 0; // RSSI value (0-31) from SIM800L*

*int lastSignalStrength = -1; // Previous signal strength for change detection*

*bool gpsConnected = false; // GPS fix status*

*bool lastGpsConnected = false; // Previous GPS status for change detection*

*int batteryPercent = 100; // Battery level (default 100%)*

*int lastBatteryPercent = -1; // Previous battery level for change detection*

*bool displayAvailable = false; // Flag: OLED initialized successfully*

*bool showingSMSMessage = false; // Flag: SMS notification is being displayed*

*unsigned long smsDisplayTime = 0; // Timestamp when SMS message was shown*

*/\* ================= UTILITIES ================= \*/*

*// Reads response from SIM800L with timeout*

*// Returns the complete response as a String*

*String readSIMResponse(unsigned long timeout = 3000) {*

*String response = "";*

*unsigned long start = millis();*

*while (millis() - start < timeout) {*

*while (sim800.available()) {*

*response += char(sim800.read());*

*}*

*}*

*return response;*

*}*

*// Checks if SIM800L is ready and registered on network*

*// Returns true if module responds and is registered*

*bool sim800Ready() {*

*// Test basic communication*

*sim800.println("AT");*

*delay(100);*

*if (readSIMResponse(1000).indexOf("OK") == -1) {*

*Serial.println("❌ SIM800L not responding to AT");*

*return false;*

*}*

*// Check network registration*

*sim800.println("AT+CREG?");*

*delay(100);*

*String r = readSIMResponse(2000);*

*// +CREG: 0,1 (registered, home network) or +CREG: 0,5 (registered, roaming)*

*if (r.indexOf(",1") == -1 && r.indexOf(",5") == -1) {*

*Serial.println("❌ SIM800L not registered on network");*

*Serial.println(r);*

*return false;*

*}*

*// Check signal strength*

*sim800.println("AT+CSQ");*

*delay(100);*

*String csq = readSIMResponse(1000);*

*Serial.print("📶 Signal: ");*

*Serial.println(csq);*

*return true;*

*}*

*/\* ================= PASSIVE BUZZER FUNCTIONS ================= \*/*

*// Plays power-on melody when system starts*

*void playPowerOnTone() {*

*Serial.println("🔊 Playing power-on melody...");*

*tone(BUZZER\_PIN, NOTE\_C5, 200);*

*delay(250);*

*tone(BUZZER\_PIN, NOTE\_E5, 200);*

*delay(250);*

*tone(BUZZER\_PIN, NOTE\_G5, 300);*

*delay(350);*

*noTone(BUZZER\_PIN);*

*Serial.println("✅ Power-on melody completed");*

*}*

*// Starts 10-second warning tone (continuous)*

*void startWarningTone() {*

*Serial.println("🚨 Starting 10-second warning tone...");*

*buzzerState = BUZZER\_WARNING\_TONE;*

*warningToneStart = millis();*

*tone(BUZZER\_PIN, NOTE\_A5); // Continuous 880 Hz tone*

*}*

*// Stops warning tone and starts beep pattern*

*void startBeepPattern() {*

*Serial.println("🔔 Starting 3-second beep pattern...");*

*noTone(BUZZER\_PIN);*

*buzzerState = BUZZER\_BEEP\_PATTERN;*

*beepPatternStart = millis();*

*}*

*// Stops all buzzer sounds*

*void stopBuzzer() {*

*noTone(BUZZER\_PIN);*

*buzzerState = BUZZER\_SILENT;*

*Serial.println("🔇 Buzzer stopped");*

*}*

*// Handles beep pattern during SMS preparation*

*void handleBeepPattern() {*

*if (buzzerState != BUZZER\_BEEP\_PATTERN) return;*

*unsigned long elapsed = millis() - beepPatternStart;*

*unsigned long cycleTime = elapsed % BEEP\_CYCLE\_TIME\_MS;*

*if (cycleTime < BEEP\_ON\_TIME\_MS) {*

*tone(BUZZER\_PIN, NOTE\_E5);*

*} else {*

*noTone(BUZZER\_PIN);*

*}*

*}*

*/\* ================= MPU FUNCTIONS ================= \*/*

*// Reads raw accelerometer and gyroscope data from MPU6050*

*// Data is stored in global variables AcX, AcY, AcZ, GyX, GyY, GyZ*

*void readMPU() {*

*Wire.beginTransmission(MPU\_ADDR);*

*Wire.write(0x3B); // Starting register for accelerometer data*

*Wire.endTransmission(false);*

*Wire.requestFrom(MPU\_ADDR, 14, true); // Read 14 bytes: 6 accel + 2 temp + 6 gyro*

*// Read accelerometer values (3 axes, 16-bit each)*

*AcX = Wire.read() << 8 | Wire.read();*

*AcY = Wire.read() << 8 | Wire.read();*

*AcZ = Wire.read() << 8 | Wire.read();*

*Wire.read(); Wire.read(); // Temperature - not used*

*// Read gyroscope values (3 axes, 16-bit each)*

*GyX = Wire.read() << 8 | Wire.read();*

*GyY = Wire.read() << 8 | Wire.read();*

*GyZ = Wire.read() << 8 | Wire.read();*

*}*

*// Calibrates the gyroscope by computing bias values*

*// Device must be kept still during calibration*

*void calibrateGyro() {*

*Serial.println("🟡 Calibrating gyro (keep helmet still)...");*

*delay(2000);*

*long sx=0, sy=0, sz=0;*

*// Take 500 samples to calculate average bias*

*for(int i=0; i<500; i++){*

*readMPU();*

*sx += GyX;*

*sy += GyY;*

*sz += GyZ;*

*delay(5);*

*}*

*// Calculate average bias for each axis*

*gyroBiasX = sx / 500.0;*

*gyroBiasY = sy / 500.0;*

*gyroBiasZ = sz / 500.0;*

*Serial.println("✅ Gyro calibrated");*

*Serial.print("Bias X: "); Serial.print(gyroBiasX);*

*Serial.print(" Y: "); Serial.print(gyroBiasY);*

*Serial.print(" Z: "); Serial.println(gyroBiasZ);*

*}*

*/\* ================= SMS FUNCTION ================= \*/*

*// Sends emergency SMS with accident location via Google Maps link*

*// Returns true if SMS was successfully sent, false otherwise*

*bool sendSMS(float lat, float lon) {*

*if (!sim800Ready()) {*

*Serial.println("❌ SIM800L NOT READY / NO NETWORK");*

*return false;*

*}*

*// Construct emergency message with location details*

*String msg = "🚨 EMERGENCY ALERT 🚨\n\n";*

*msg += "Possible accident detected by Smart Helmet.\n";*

*msg += "The rider may be injured or unconscious.\n\n";*

*msg += "📍 Location:\n";*

*msg += "Lat: " + String(lat, 6) + "\n";*

*msg += "Lon: " + String(lon, 6) + "\n\n";*

*msg += "🗺️ Google Maps:\n";*

*msg += "https://maps.google.com/?q=";*

*msg += String(lat, 6) + "," + String(lon, 6);*

*msg += "\n\n⚠️ Please contact the rider or send help immediately.\n";*

*msg += "\n— Smart Helmet Safety System";*

*Serial.println("📤 Preparing to send SMS...");*

*// Set SMS to text mode*

*sim800.println("AT+CMGF=1");*

*delay(100);*

*String resp1 = readSIMResponse(1000);*

*if (resp1.indexOf("OK") == -1) {*

*Serial.println("❌ Failed to set text mode");*

*return false;*

*}*

*// Send recipient number*

*sim800.print("AT+CMGS=\"");*

*sim800.print(PHONE\_NUMBER);*

*sim800.println("\"");*

*delay(500);*

*// Wait for '>' prompt from SIM800L*

*String prompt = readSIMResponse(2000);*

*if (prompt.indexOf(">") == -1) {*

*Serial.println("❌ SMS PROMPT FAILED - No '>' received");*

*Serial.println(prompt);*

*return false;*

*}*

*// Send message content*

*sim800.print(msg);*

*delay(100);*

*sim800.write(26); // Send CTRL+Z to indicate end of message*

*// Wait for confirmation*

*String result = readSIMResponse(10000); // SMS can take time*

*if (result.indexOf("+CMGS:") != -1 || result.indexOf("OK") != -1) {*

*Serial.println("📨 SMS SUCCESSFULLY SENT");*

*return true;*

*}*

*Serial.println("❌ SMS SENDING FAILED");*

*Serial.println(result);*

*return false;*

*}*

*/\* ================= OLED DISPLAY FUNCTIONS ================= \*/*

*// Queries signal strength from SIM800L using AT+CSQ command*

*// Returns RSSI value (0-31), or 0 if query fails*

*int querySignalStrength() {*

*sim800.println("AT+CSQ");*

*delay(100);*

*char buffer[64] = {0};*

*unsigned long start = millis();*

*int idx = 0;*

*// Read response with timeout*

*while (millis() - start < 2000 && idx < 63) {*

*if (sim800.available()) {*

*buffer[idx++] = sim800.read();*

*}*

*}*

*buffer[idx] = '\0';*

*// Parse "+CSQ: <rssi>,<ber>" response*

*int rssi = 0;*

*const int CSQ\_PREFIX\_LENGTH = 6; // Length of "+CSQ: "*

*char\* csqPos = strstr(buffer, "+CSQ:");*

*if (csqPos != NULL) {*

*sscanf(csqPos + CSQ\_PREFIX\_LENGTH, "%d", &rssi);*

*return rssi;*

*}*

*return 0; // No signal or error*

*}*

*// Draws battery icon with fill level at specified position*

*// Parameters: x, y = top-left position, percent = battery level (0-100)*

*void drawBatteryIcon(int x, int y, int percent) {*

*const int BATTERY\_WIDTH = 18;*

*const int BATTERY\_HEIGHT = 10;*

*const int BATTERY\_BORDER = 2;*

*// Draw battery outline*

*display.drawRect(x, y, BATTERY\_WIDTH, BATTERY\_HEIGHT, SSD1306\_WHITE);*

*// Draw battery tip (2x4 pixels)*

*display.fillRect(x + BATTERY\_WIDTH, y + 3, 2, 4, SSD1306\_WHITE);*

*// Calculate fill width based on percentage*

*// Inner area = BATTERY\_WIDTH - (2 \* BATTERY\_BORDER)*

*int maxFillWidth = BATTERY\_WIDTH - (2 \* BATTERY\_BORDER);*

*int fillWidth = (percent \* maxFillWidth) / 100;*

*if (fillWidth > 0) {*

*display.fillRect(x + BATTERY\_BORDER, y + BATTERY\_BORDER, fillWidth, BATTERY\_HEIGHT - (2 \* BATTERY\_BORDER), SSD1306\_WHITE);*

*}*

*// Display percentage text below icon*

*display.setCursor(x, y + 12);*

*display.setTextSize(1);*

*display.print(percent);*

*display.print("%");*

*}*

*// Draws signal strength bars at specified position*

*// Parameters: x, y = top-left position, strength = RSSI value (0-31)*

*void drawSignalBars(int x, int y, int strength) {*

*// Determine number of bars based on signal strength*

*// Signal mapping: 0-10=WEAK(1 bar), 11-20=MEDIUM(3 bars), 21-31=STRONG(4 bars)*

*// Note: Intentionally skips 2 bars to provide clearer visual distinction between levels*

*int numBars = 0;*

*const char\* label = "NO SIG";*

*if (strength >= 21) {*

*numBars = 4;*

*label = "STRONG";*

*} else if (strength >= 11) {*

*numBars = 3;*

*label = "MEDIUM";*

*} else if (strength > 0) {*

*numBars = 1;*

*label = "WEAK";*

*}*

*// Draw 4 signal bars with varying heights*

*int barHeights[] = {4, 6, 8, 10};*

*for (int i = 0; i < 4; i++) {*

*int barX = x + (i \* 5);*

*int barY = y + 10 - barHeights[i];*

*if (i < numBars) {*

*// Filled bar for active signal*

*display.fillRect(barX, barY, 3, barHeights[i], SSD1306\_WHITE);*

*} else {*

*// Outlined bar for inactive*

*display.drawRect(barX, barY, 3, barHeights[i], SSD1306\_WHITE);*

*}*

*}*

*// Display signal quality label below bars*

*display.setCursor(x, y + 12);*

*display.setTextSize(1);*

*display.print(label);*

*}*

*// Main display update function - refreshes OLED screen*

*void updateOLEDDisplay() {*

*if (!displayAvailable) return; // Skip if display not initialized*

*display.clearDisplay();*

*if (showingSMSMessage) {*

*// Display SMS sent notification*

*display.setTextSize(2);*

*display.setCursor(10, 20);*

*display.print("SMS SENT");*

*display.setTextSize(1);*

*display.setCursor(30, 40);*

*display.print("SUCCESS!");*

*} else {*

*// Display regular status screen*

*// Battery icon (top-left)*

*drawBatteryIcon(5, 5, batteryPercent);*

*// Signal strength bars (top-right)*

*drawSignalBars(88, 5, signalStrength);*

*// GPS connection status (bottom)*

*display.setTextSize(1);*

*display.setCursor(10, 50);*

*if (gpsConnected) {*

*display.print("GPS: Connected");*

*} else {*

*display.print("GPS: Connecting...");*

*}*

*}*

*display.display();*

*}*

*/\* ================= SETUP ================= \*/*

*int readBatteryPercent() {*

*const float ADC\_REF = 3.3;*

*const float ADC\_MAX = 4095.0;*

*float sum = 0;*

*for (int i = 0; i < 20; i++) {*

*sum += analogRead(BATTERY\_PIN);*

*delay(2);*

*}*

*float adc = sum / 20.0;*

*float vAdc = adc \* (ADC\_REF / ADC\_MAX);*

*// 100k / 47k divider*

*float batteryVoltage = vAdc \* (147.0 / 47.0);*

*int percent = map(batteryVoltage \* 100, 330, 420, 0, 100);*

*percent = constrain(percent, 0, 100);*

*return percent;*

*}*

*void setup() {*

*// Initialize serial communication for debugging*

*Serial.begin(115200);*

*delay(1000);*

*// ✅ ESP32 ADC configuration (PLACE HERE)*

*analogReadResolution(12); // 0–4095*

*analogSetAttenuation(ADC\_11db); // allows up to ~3.6V*

*Serial.println("\n\n========================================");*

*Serial.println(" SMART HELMET - PASSIVE BUZZER ");*

*Serial.println("========================================\n");*

*// Configure buzzer and button pins*

*pinMode(BUZZER\_PIN, OUTPUT);*

*pinMode(CANCEL\_BTN, INPUT\_PULLUP);*

*noTone(BUZZER\_PIN); // Ensure buzzer is off initially*

*// Initialize I2C for MPU6050 (SDA=21, SCL=22)*

*Serial.println("🔧 Initializing MPU6050...");*

*Wire.begin(21, 22);*

*// Wake up MPU6050 (it starts in sleep mode)*

*Wire.beginTransmission(MPU\_ADDR);*

*Wire.write(0x6B); // Power management register*

*Wire.write(0); // Wake up (set sleep bit to 0)*

*Wire.endTransmission();*

*delay(100);*

*Serial.println("✅ MPU6050 initialized");*

*// Calibrate gyroscope for accurate readings*

*calibrateGyro();*

*// Initialize OLED Display*

*Serial.println("🔧 Initializing OLED Display...");*

*if(!display.begin(SSD1306\_SWITCHCAPVCC, 0x3C)) {*

*Serial.println("⚠️ SSD1306 allocation failed - display disabled");*

*displayAvailable = false;*

*} else {*

*displayAvailable = true;*

*display.clearDisplay();*

*display.setTextColor(SSD1306\_WHITE);*

*display.setTextSize(1);*

*display.setCursor(0, 0);*

*display.println("Smart Helmet");*

*display.println("Initializing...");*

*display.display();*

*Serial.println("✅ OLED Display initialized");*

*}*

*// Initialize GPS module on UART2*

*Serial.println("🔧 Initializing GPS...");*

*gpsSerial.begin(9600, SERIAL\_8N1, GPS\_RX, GPS\_TX);*

*Serial.println("✅ GPS module initialized (waiting for fix...)");*

*// Initialize SIM800L on UART1*

*Serial.println("🔧 Initializing SIM800L...");*

*sim800.begin(9600, SERIAL\_8N1, SIM\_RX, SIM\_TX);*

*delay(3000); // Give SIM800L time to boot up*

*// Test SIM800L connectivity and play power-on tone if ready*

*if (sim800Ready()) {*

*Serial.println("✅ SIM800L ready and registered on network");*

*playPowerOnTone(); // Play melody when system is ready*

*} else {*

*Serial.println("⚠️ SIM800L not ready - check power and SIM card");*

*}*

*Serial.println("\n========================================");*

*Serial.println(" SMART HELMET SYSTEM READY ");*

*Serial.println(" Monitoring for accidents... ");*

*Serial.println("========================================\n");*

*}*

*/\* ================= LOOP ================= \*/*

*void loop() {*

*// ===== CANCEL BUTTON CHECK =====*

*// Allow user to cancel false alarm by pressing button*

*if (digitalRead(CANCEL\_BTN) == LOW) {*

*// If in beep pattern phase, abort SMS*

*if (buzzerState == BUZZER\_BEEP\_PATTERN) {*

*Serial.println("❌ SMS ABORTED BY USER DURING BEEP PATTERN");*

*smsAborted = true;*

*}*

*// Cancel all alerts*

*crashDetected = false;*

*smsSent = false;*

*smsAborted = false;*

*showingSMSMessage = false;*

*stopBuzzer();*

*Serial.println("❌ ALERT CANCELLED BY USER");*

*delay(1000); // Debounce delay*

*}*

*// Check if SMS message should be cleared after 2 seconds*

*if (showingSMSMessage && (millis() - smsDisplayTime >= 2000)) {*

*showingSMSMessage = false;*

*updateOLEDDisplay();*

*}*

*// ===== BUZZER STATE MACHINE =====*

*// Handle warning tone timing (10 seconds)*

*if (buzzerState == BUZZER\_WARNING\_TONE) {*

*if (millis() - warningToneStart >= WARNING\_TONE\_DURATION) {*

*startBeepPattern();*

*}*

*}*

*// Handle beep pattern timing (3 seconds)*

*if (buzzerState == BUZZER\_BEEP\_PATTERN) {*

*unsigned long beepElapsed = millis() - beepPatternStart;*

*if (beepElapsed >= BEEP\_PATTERN\_DURATION) {*

*// Beep pattern complete - proceed to send SMS if not aborted*

*stopBuzzer();*

*if (!smsAborted && gps.location.isValid() && !smsSent) {*

*Serial.println("📍 Beep pattern complete - sending emergency SMS...");*

*if (sendSMS(gps.location.lat(), gps.location.lng())) {*

*smsSent = true;*

*showingSMSMessage = true;*

*smsDisplayTime = millis();*

*updateOLEDDisplay();*

*} else {*

*Serial.println("⚠️ SMS failed - will retry...");*

*delay(5000); // Wait before retry*

*}*

*}*

*} else {*

*// Continue beep pattern*

*handleBeepPattern();*

*}*

*}*

*// ===== READ SENSOR DATA =====*

*readMPU();*

*// Convert raw accelerometer values to g-force (1g = 16384 LSB at ±2g scale)*

*float ax = AcX / 16384.0;*

*float ay = AcY / 16384.0;*

*float az = AcZ / 16384.0;*

*// Convert raw gyroscope values to degrees per second (dps)*

*// At ±250°/s scale: 131 LSB/(°/s)*

*// Subtract bias calculated during calibration*

*float gx = (GyX - gyroBiasX) / 131.0;*

*float gy = (GyY - gyroBiasY) / 131.0;*

*float gz = (GyZ - gyroBiasZ) / 131.0;*

*// Calculate magnitude of acceleration and rotation vectors*

*float accMag = sqrt(ax\*ax + ay\*ay + az\*az);*

*float gyroMag = sqrt(gx\*gx + gy\*gy + gz\*gz);*

*// Display real-time sensor readings*

*Serial.print("ACC(g): "); Serial.print(accMag, 2);*

*Serial.print(" | GYRO(dps): "); Serial.print(gyroMag, 1);*

*// ===== CRASH DETECTION LOGIC =====*

*// A crash is characterized by:*

*// 1. High acceleration (sudden jerk/impact)*

*// 2. High rotation (bike/helmet tilting)*

*// 3. Both conditions sustained for IMPACT\_TIME\_MS*

*if (accMag > ACC\_THRESHOLD && gyroMag > GYRO\_THRESHOLD) {*

*// Potential crash detected - start timing*

*if (impactStart == 0) {*

*impactStart = millis();*

*Serial.print(" [IMPACT DETECTED]");*

*}*

*// Verify impact sustained for minimum time to avoid false triggers*

*if (millis() - impactStart > IMPACT\_TIME\_MS) {*

*if (!crashDetected) {*

*crashDetected = true;*

*smsAborted = false; // Reset abort flag*

*startWarningTone(); // Start 10-second warning tone*

*Serial.println("\n🚨🚨🚨 CRASH CONFIRMED 🚨🚨🚨");*

*}*

*}*

*} else {*

*// Conditions no longer met - reset timer*

*impactStart = 0;*

*}*

*Serial.println(); // New line for readability*

*// ===== GPS DATA PROCESSING =====*

*// Continuously feed GPS data to parser*

*while (gpsSerial.available()) {*

*gps.encode(gpsSerial.read());*

*}*

*// Update GPS connection status*

*gpsConnected = gps.location.isValid();*

*// ===== GPS STATUS MONITORING =====*

*// Inform user if waiting for GPS fix after crash*

*if (crashDetected && !gps.location.isValid() && !smsSent) {*

*Serial.print("⏳ Waiting for GPS fix... ");*

*Serial.print("Satellites: "); Serial.println(gps.satellites.value());*

*}*

*// Display GPS info when available*

*if (gps.location.isValid()) {*

*Serial.print("📍 GPS: ");*

*Serial.print(gps.location.lat(), 6);*

*Serial.print(", ");*

*Serial.print(gps.location.lng(), 6);*

*Serial.print(" | Sats: ");*

*Serial.println(gps.satellites.value());*

*}*

*// Query signal strength periodically (every 5 seconds)*

*static unsigned long lastSignalQuery = 0;*

*static bool firstUpdate = true; // Track if we need initial update*

*if (millis() - lastSignalQuery >= 5000) {*

*signalStrength = querySignalStrength();*

*lastSignalQuery = millis();*

*}*

*// 🔋 Battery percentage update (every 2 seconds)*

*static unsigned long lastBatteryRead = 0;*

*if (millis() - lastBatteryRead >= 2000) {*

*batteryPercent = readBatteryPercent();*

*lastBatteryRead = millis();*

*}*

*// Update display when state changes or on first loop iteration*

*bool stateChanged = (signalStrength != lastSignalStrength) ||*

*(gpsConnected != lastGpsConnected) ||*

*(batteryPercent != lastBatteryPercent) ||*

*firstUpdate;*

*if (!showingSMSMessage && stateChanged) {*

*updateOLEDDisplay();*

*// Update tracking variables only if they actually changed*

*if (signalStrength != lastSignalStrength) lastSignalStrength = signalStrength;*

*if (gpsConnected != lastGpsConnected) lastGpsConnected = gpsConnected;*

*if (batteryPercent != lastBatteryPercent) lastBatteryPercent = batteryPercent;*

*firstUpdate = false;*

*}*

*delay(200); // Loop delay for sensor sampling rate (~5 Hz)*

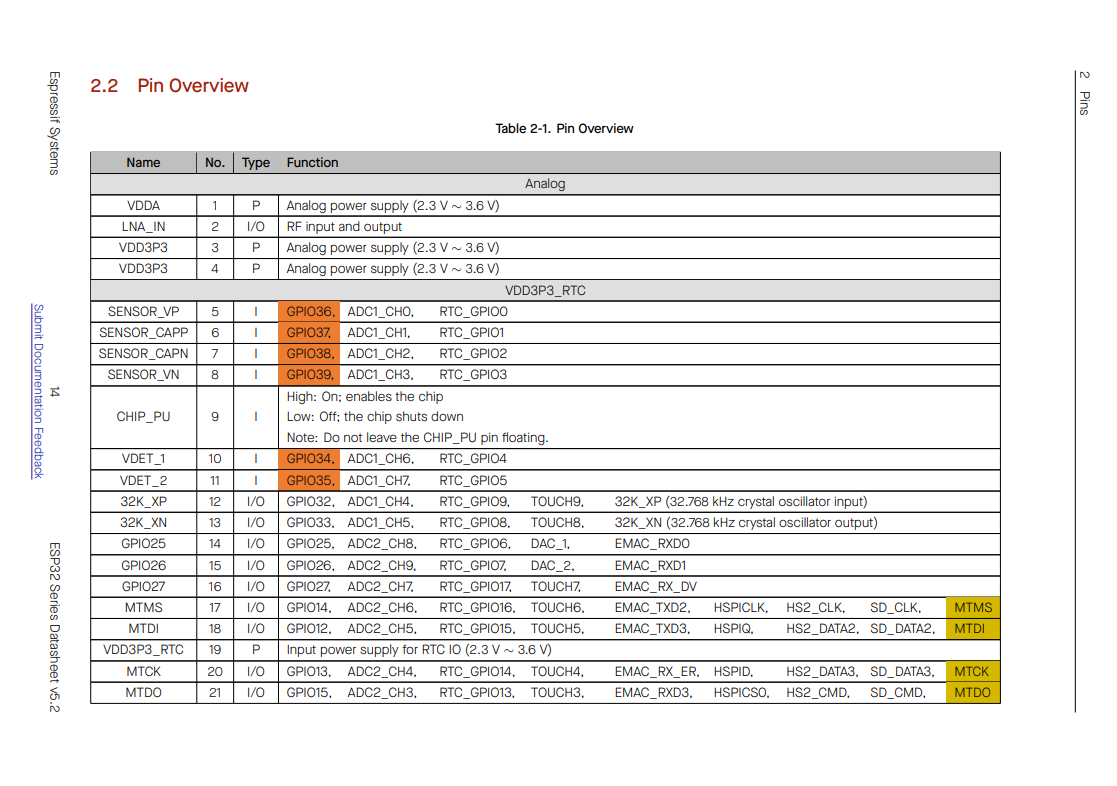
*}*

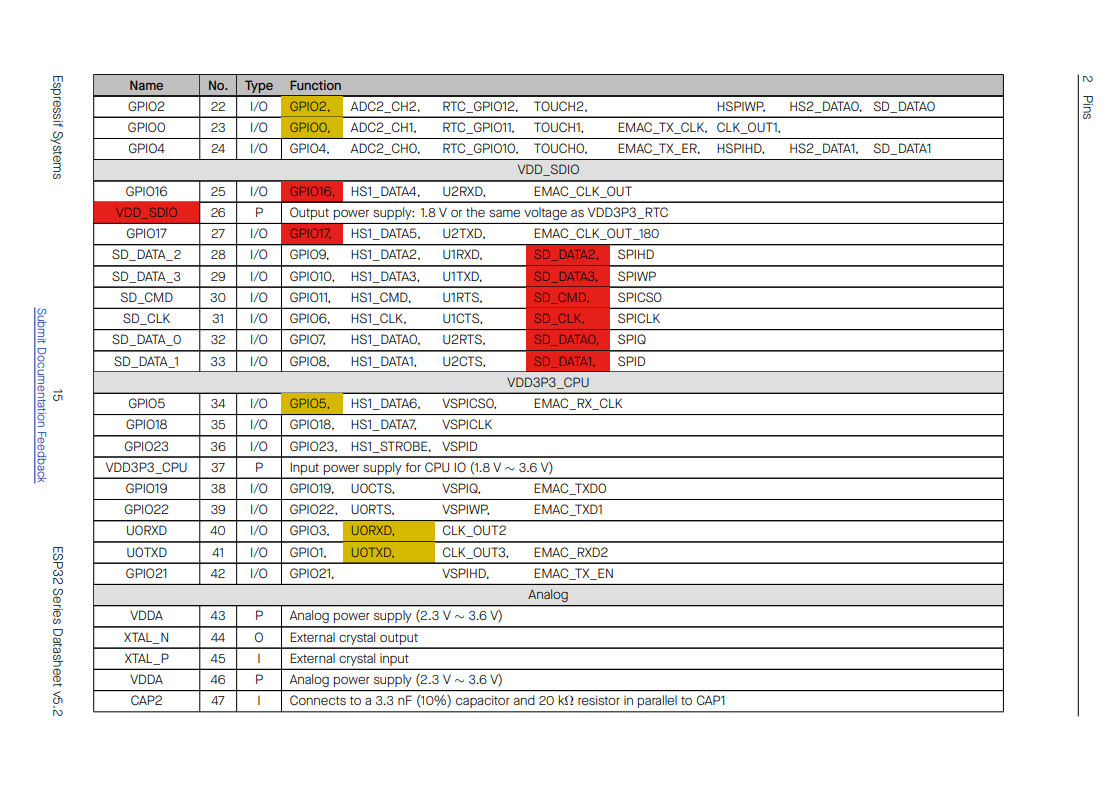
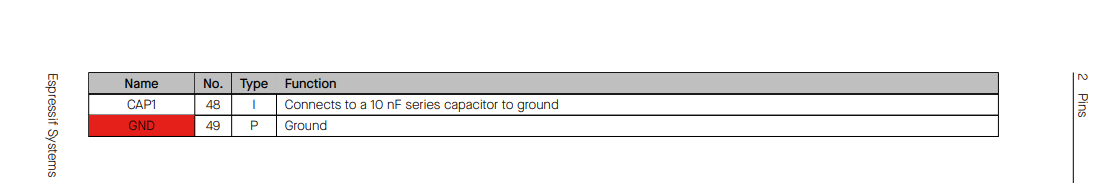
Appendix D

Datasheet of components

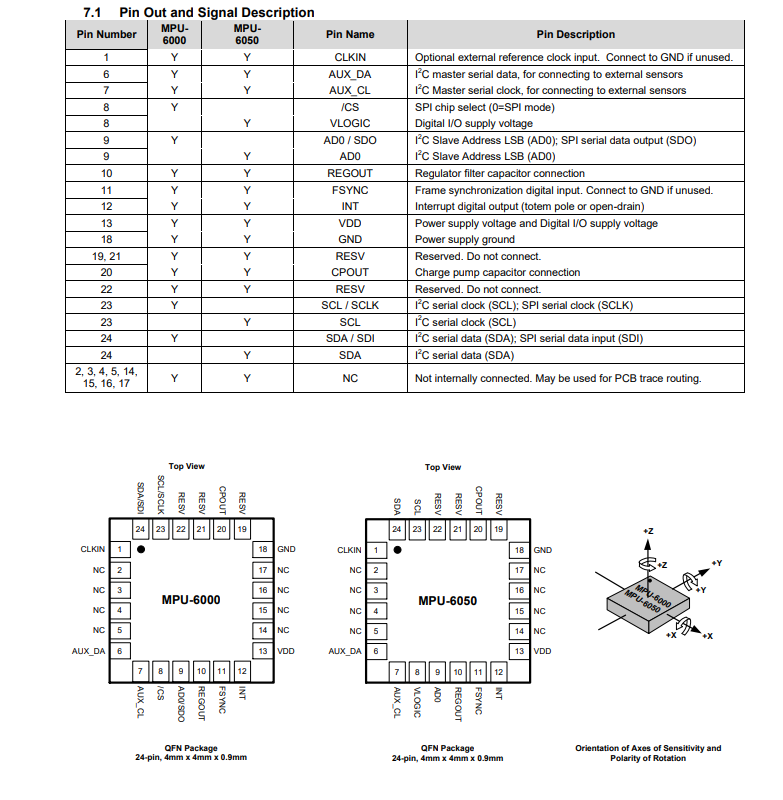
This appendix includes the official datasheets of the primary electronic components used in the development of the Smart Helmet with Sensor-Based Crash Detection and Automated Emergency Communication system. Each datasheet provides detailed technical specifications, electrical characteristics, pin configurations, and other relevant information necessary for accurate design, integration, and safe operation of the component in the system.

**ESP32 Development Board:**

****

**** ****

**MPU6050 Accelerometer and Gyroscope Sensor:**



**u-blox NEO-M8N GPS Module**

