SAFERIDE HELMET SYSTEM

A SUMMER PROJECT REPORT

Submitted by

BARAKATH NISHA S 2022504508

AMIRTHA K 2022504518

ANU PRASANNA R 2022504545

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

DEPARTMENT OF ELECTRONICS ENGINEERING

MADRAS INSTITUTE OF TECHNOLOGY

ANNA UNIVERSITY: CHENNAI 600 044

NOVEMBER 2024

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **“SAFERIDE HELMET SYSTEM”** is the bonafide work of

BARAKATH NISHA S	**2022504508**
AMIRTHA K	**2022504518**
ANU PRASANNA R	**2022504545**

who carried out the project work under my supervision.

SIGNATURE

Dr. D, Meganathan

HEAD OF THE DEPARTMENT

Professor

Department of Electronics Engineering,

Madras Institute of Technology,

Anna University, Chennai-600 044.

SIGNATURE

Dr.C.Sridevi

SUPERVISOR

Associate Professor

Department of Electronics Engineering,

Madras Institute of Technology,

Anna University, Chennai -600 044.

ACKNOWLEDGEMENT

With profound gratitude, we sincerely acknowledge with thanks the opportunity provided to us by our **Dr. K. RAVICHANDRAN** , Dean, Madras Institute of technology, Anna University forproviding good environment and facilities.

We thank our respectful Head of the Department, **Dr. D. MEGANATHAN**, Professor and Head of Electronics Engineering, Madras Institute of technology, Anna University for his encouragement during the Project work.

We sincerely express our gratitude to our guide **Mrs. C. SRIDEVI** ,Associate Professor, Department of Electronics Engineering for their valuable guidance throughout the course of this project.

Place: Chennai

Date:15-11-2024

ABSTRACT

The Smart Helmet project focuses on advancing motorcycle safety by integrating essential safety features into a helmet-based system that operates through an Arduino Mega microcontroller. This system addresses key aspects of road safety, including helmet usage enforcement, alcohol detection, accident detection, and real-time location sharing. Using a helmet detection mechanism, the bike's ignition is disabled if the rider is not wearing the helmet, thereby ensuring compliance with safety protocols.

To prevent incidents caused by intoxicated driving, the system incorporates an MQ-3 alcohol sensor, which checks the rider's breath for alcohol and disables ignition upon detection. The system further improves safety by embedding an accelerometer that continuously monitors for sudden impacts; in the event of an accident, the accelerometer triggers the system's emergency response functions. Upon accident detection, GPS and GSM modules work together to transmit the rider's location to predefined emergency contacts, ensuring prompt aid.

Additionally, the use of RF modules facilitates wireless communication between the helmet and the bike, allowing these features to operate efficiently and reliably. By combining accident prevention and detection with proactive emergency communication, this Smart Helmet project provides a comprehensive safety solution. It not only enforces helmet use and prevents intoxicated driving but also offers real-time location updates during emergencies, aiming to reduce the risks and consequences of motorcycle accidents. This approach presents a practical and technologically advanced solution to improve rider safety on the roads.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER -1

1. INTRODUCTION

The Smart Helmet project is designed to enhance rider safety by integrating various sensors and technologies into a helmet for real-time monitoring and accident prevention. This innovative system includes features such as alcohol detection to prevent the bike from starting, helmet detection to ensure the rider is wearing the helmet, and accident detection with GPS location sharing for quick assistance. The project employs two Arduino Mega 2560 boards, an LCD display, MQ-3 sensor, accelerometer, GPS module, GSM module, and more. Together, these components work to create a safer, smarter riding experience by combining technology with essential safety measures.

1.1 PURPOSE OF STUDY

The purpose of studying the Smart Helmet project is to explore the integration of various electronic components and sensors to improve rider safety and provide real-time solutions for critical situations. By examining the system's functionalities such as alcohol detection, helmet detection, and accident detection with location sharing the project aims to highlight how technology can reduce risks, prevent accidents, and assist in emergencies. This study also serves as an opportunity to understand how embedded systems, sensors, and communication technologies can be applied in practical, life-saving applications, ultimately contributing to the development of smarter, safer transportation solutions

CHAPTER-2

EXISTING METHODOLOGY

Existing smart helmet methodologies focus on distinct safety features like helmet detection, crash detection, alcohol sensing, and GPS integration, each aiming to improve rider safety in specific ways. Helmet detection systems use sensors to ensure the rider wears the helmet by preventing the bike from starting otherwise. Crash detection, using accelerometers and gyroscopes, senses impacts and sends location alerts to emergency contacts upon an accident. Alcohol detection, often through MQ-3 sensors, detects if the rider has consumed alcohol, blocking the bike's ignition if alcohol is present. GPS and GSM modules offer real-time location tracking and emergency communication but usually require manual activation. While effective individually, these systems operate largely as standalone features, lacking seamless integration and proactive safety measures.

Many helmets focus solely on reactive responses, engaging only after an accident occurs. This approach often misses opportunities for real-time intervention and preventive measures, particularly in urgent scenarios where the rider cannot manually activate the safety systems. The Smart Helmet project aims to combine these functions—helmet detection, alcohol detection, and crash detection with automated GPS location sharing—into a single, cohesive solution, providing a more comprehensive, automated, and proactive approach to ensure rider safety and swift response in emergencies.

CHAPTER-3

PROPOSED METHODOLOGY

The proposed methodology for the Smart Helmet project aims to create a comprehensive safety system that integrates multiple protective features. This system leverages an Arduino Mega board as the central controller, combining various sensors and modules to enhance road safety. The helmet is equipped with a detection mechanism that ensures the bike only starts if the helmet is worn. To prevent drunk driving, an MQ-3 alcohol sensor is used to detect the presence of alcohol; if any is detected, the ignition is disabled. An accelerometer monitors for sudden impacts, serving as an accident detection tool. Upon detecting an accident, the system utilizes GPS and GSM modules to share real-time location details with emergency contacts, facilitating rapid response. Additionally, RF modules are employed for wireless communication between the helmet and bike, enabling smooth operation of all safety functions. This integrated approach provides a proactive, multifaceted solution to enhance rider safety, combining prevention, detection, and emergency response in a single system.

3.1HARDWARE IMPLEMENTATION

The hardware implementation of the smart helmet project involves integrating several components to achieve the desired functionalities. The transmitter side, located within the helmet, includes an Arduino Mega microcontroller, an MQ-3 alcohol sensor for detecting the rider's alcohol levels, an MPU-6050 accelerometer and gyroscope module for accident detection, a 433MHz RF transmitter module for wireless communication, and a 3A 250VAC switch to ensure the helmet is worn before starting the bike. The MQ-3 sensor and MPU-6050 are interfaced with the Arduino Mega, with the former connected to analog input pins for alcohol level measurement and the latter using the I2C interface for motion detection. The switch is connected to a digital pin to monitor its status, while the transmitter is used to send signals to the receiver. Additionally, a power supply, such as a 9V battery, powers the system, and LEDs or a buzzer provide alerts. This configuration ensures the helmet fulfills its safety and communication roles effectively.

3.2 HARDWARE USED

3.2.1 TRANSMITTER MODULE

The Motor Driver is a module for motors that allows you to control the working speed and direction of two motors simultaneously. This Motor Driver is designed and developed based on L293D IC.L293D is a 16 Pin Motor Driver IC. This is designed to provide bidirectional drive currents at voltages from 5 V to 36V.

Fig 1 RF transmitter and receiver Module

the RF modules are 433 MHz RF transmitter and receiver modules. The transmitter draws no power when transmitting logic zero while fully suppressing the carrier frequency thus consume significantly low power in battery operation. When logic one is sent carrier is fully on to about 4.5mA with a 3volts power supply. The data is sent serially from the transmitter which is received by the tuned receiver. Transmitter and the receiver are duly interfaced to two microcontrollers for data transfer.

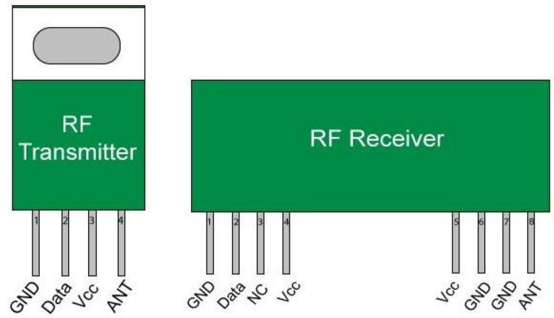


Fig 2 RF transmitter and receiver Module- Pin configuration

Features of RF Module:

- Receiver frequency 433MHz
- Receiver typical frequency 105Dbm
- Receiver supply current 3.5mA
- Low power consumption
- Receiver operating voltage 5v
- Transmitter frequency range 433.92MHz
- Transmitter supply voltage 3v~6v
- Transmitter output power 4v~12v

3.2.2 NEO 6M GPS MODULE

At the heart of the module is a GPS chip from U-blox – NEO-6M. The chip measures. less than a postage stamp but packs a surprising amount of features into its tiny frame. It can track up to 22 satellites over 50 channels and achieve the industry's highest level of tracking sensitivity i.e. -161 dB, while consuming only 45 mA current. Unlike other GPS modules, it can perform 5 location updates in a second with 2.5m horizontal position accuracy. The U-blox 6 positioning engine also has a Time-To-First-Fix (TTFF) of less than 1 second.

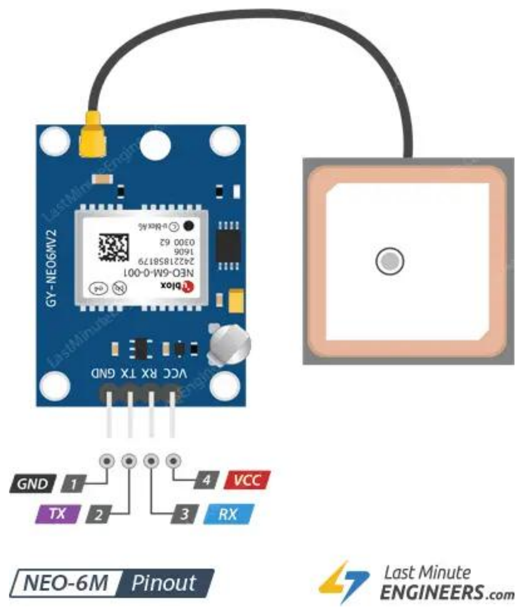


Fig 3 NEO 6M GPS module

One of the best features offered by the chip is Power Save Mode (PSM). This allows a reduction in system power consumption by selectively switching certain parts of the receiver on and off. This dramatically reduces the power consumption of the module to just 11mA making it suitable for power sensitive applications such as GPS wristwatches.
outputs. The sensor's compact PCB measures 3cm x 1.6cm and includes power and digital switch output indicators for easy monitoring. Notably, it activates flame testing if the flame

The required data pins of the NEO-6M GPS chip are broken out to a 0.1″ pitch headers. It contains the pins needed for communication with the microcontroller over the UART. The module supports baud rates from 4800bps to 230400bps with a default baud of 9600.

Position Fix LED Indicator:

There is an LED on the NEO-6M GPS module that indicates the status of the ‘Position Fix’. It will blink at different rates depending on which state it is in:

- No blinking – it is searching for satellites.
- Blink every 1s – Position Fix is found (the module can see enough satellites).

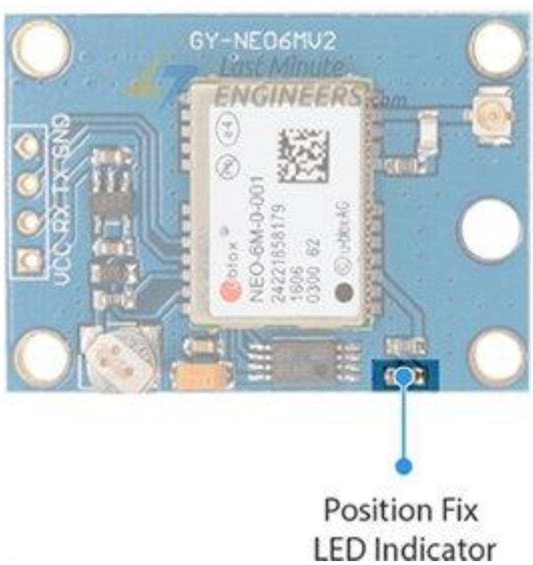


Fig 4 Detection of location

3.2.3 GSM SIM800L MODULE:

At the heart of the module is a SIM800L GSM cellular chip from Simcom. The operating voltage of the chip ranges from 3.4V to 4.4V, making it an ideal candidate for direct LiPo battery supply. This makes it an excellent choice for embedding in projects with limited space.

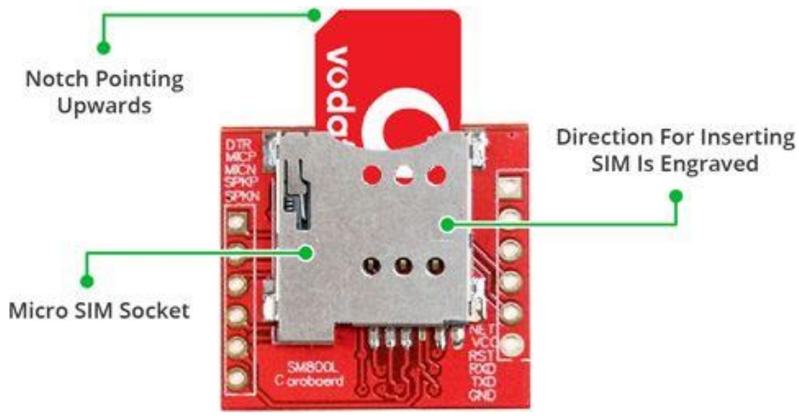
Fig 5 Submersible Water Pump

All the necessary data pins of the SIM800L GSM chip are broken out to a 0.1″ pitch headers, including the pins required for communication with the microcontroller over the UART. The module supports baud rates ranging from 1200 bps to 115200 bps and features automatic baud rate detection.

The module requires an external antenna in order to connect to the network. So the module usually comes with a helical antenna that can be soldered to it. The board also has a U.FL connector If you wish to keep the antenna at a distance from the board.

LED Status Indicators

The SIM800L module has an LED that indicates the status of your cellular network. It will blink at different rates depending on the state it is in.



Blink every 1s

The chip is running but hasn't made a connection to the cellular network yet.

Blink every 2s

The GPRS data connection you requested is active.

The module has made contact with the cellular network and can send/receive voice and SMS.

Blink every 3s

The module has made contact with the cellular network and can send/receive voice and SMS.

Features

- Supports Quad-band: GSM850, EGSM900, DCS1800 and PCS1900
- Connect onto any global GSM network with any 2G SIM
- Make and receive voice calls using an external 8Ω speaker & electret microphone
- Send and receive SMS messages
- Send and receive GPRS data (TCP/IP, HTTP, etc.)
- Scan and receive FM radio broadcasts

3.2.4 LIMIT SWITCH

Limit switch definition is an electromechanical switch that operates by any physical force or the movement of a machine. These switches are very helpful in detecting the absence or presence of an object, counting, detecting speed, detecting movement range, travel limit, positioning, etc. These switches include three terminals NO (Normally Open), NC (Normally Open) & Common.

Fig 6 Limit switch

The controlling of this switch can be done by different factors like temperature, position, and pressure. This switch is mainly designed to operate only once a fixed limit is achieved, and it is generally activated through contact by using an object like a cam. A limit switch working principle is similar to a sensor to identify the presence & absence of an object. This switch can be triggered mechanically by communicating with other substances. When the object contacts the actuator of the switch, then it moves eventually to the boundary of the actuator wherever the location of the contact changes. Most of these switches are mechanical in their operation & include heavy-duty contacts which are capable of switching high current. The limit switch works by using a physical response to function as ON and OFF. This switch needs either a machine or a manually operated switch to change the current flow. These are contact switches. Its working life is shorter because of the internal contacts. These switches are used for the detection of positions in only one direction detection of positions in only one direction.

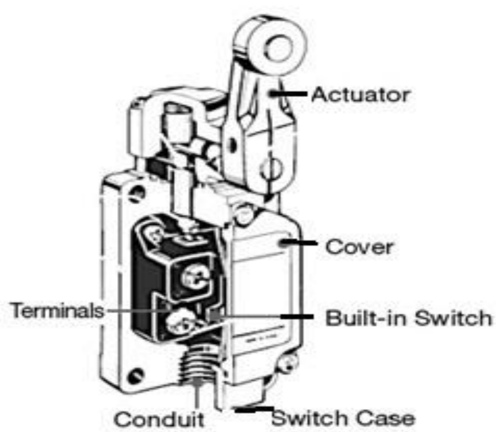


Fig 7 Construction of limit switch

3.2.5 MQ3 SENSOR

Fig 8 MQ3 Sensor

The MQ3 sensor is one of the most widely used in the MQ sensor series. It is a MOS (Metal Oxide Semiconductor) sensor. Metal oxide sensors are also known as Chemiresistors because sensing is based on the change in resistance of the sensing material when exposed to alcohol. The MQ3 alcohol sensor operates on 5V DC and consumes approximately 800mW. It can detect alcohol concentrations ranging from 25 to 500 ppm. When a SnO_2 semiconductor layer is heated to a high temperature, oxygen is adsorbed on the surface. When the air is clean, electrons from the conduction band of tin dioxide are attracted to oxygen molecules. This creates an electron depletion layer just beneath the surface of the SnO_2 particles, forming a potential barrier. As a result, the SnO_2 film becomes highly resistive and prevents electric current flow. In the presence of alcohol, however, the surface density of adsorbed oxygen decreases as it reacts with the alcohol, lowering the potential barrier. As a result, electrons are released into the tin dioxide, allowing current to freely flow through the sensor.

3.2.6 RELAY MODULE

A relay serves as an electric control device, opening or closing contacts to activate other electrical components. Consisting of an electromagnet, mechanically movable contact, switching points, and springs, it includes pins like COM (common), NO (normally open), and NC (normally closed). When triggered, it connects the COM pin to NO, providing power to the load, or disconnects the COM from NC, interrupting power flow. Operating between 3.75V to 6V, the relay has a quiescent current of 2mA, increasing to around 70mA when active.

Fig 9 5V Relay Module

With a maximum contact voltage of 250VAC/30VDC and a current limit of 10, it operates on the electromagnetic attraction principle, energizing the electromagnetic field upon fault detection to move the relay armature and establish or break connections.

Types include electrothermal, electromechanical, solid-state, and hybrid relays, each serving various purposes from voltage protection systems to motor speed control, electricalisolation, home automation, and high-current switching in diverse applications.

3.2.7 ACCELEROMETER

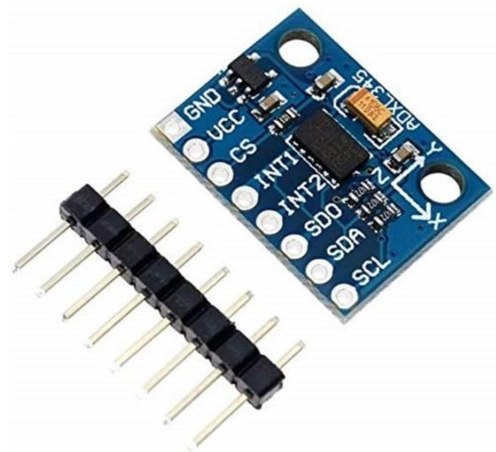


Fig 10 Accelerometer

An accelerometer is a device that measures the vibration, or acceleration of motion, of a structure. The force caused by vibration or a change in motion (acceleration) causes the mass to “squeeze” the piezoelectric material which produces an electrical charge that is proportional to the force exerted upon it. Since the charge is proportional to the force, and the mass is constant, then the charge is also proportional to the acceleration. These sensors are used in a variety of ways – from space stations to handheld devices – and there’s a good chance you already own a device with an accelerometer in it. For example, almost all smartphones today house an accelerometer. They help the phone know whether it undergoes acceleration in any direction, and it’s the reason why your phone’s display switches on when you flip it. In an industrial setting, accelerometers help engineers understand a machine’s stability and enable them to monitor for any unwanted forces/vibrations. An accelerometer works by utilizing an electromechanical sensor that is designed to measure either static or dynamic acceleration. Static acceleration is the constant force acting on a body, like gravity or friction. These forces are predictable and uniform to a large extent. For example, the acceleration due to gravity is constant at 9.8 m/s – and the gravitation force is almost the same at every point on Earth.

Dynamic acceleration forces are non-uniform, and the best example is vibration or shock. A car crash is an excellent example of dynamic acceleration. Here, the acceleration change is sudden when compared to its previous state. The theory behind accelerometers is that they can detect acceleration and convert it into measurable quantities – like electrical signals.

3.2.8 GEAR MOTOR

A gear motor is a type of motor that uses gears to reduce its output speed and increase its output torque. By doing so, it can deliver more power to a load with less energy consumption. Gear motors are widely used in a variety of applications, including industrial automation, robotics, automotive, medical equipment, and more. They are known for their durability, reliability, and efficiency.

Gear motors can be categorized into three types based on the arrangement of the gears:

- Parallel shaft gear motors: the input and output shafts are parallel to each other and are connected by gears that are mounted on them.
- Right-angle gear motors: the input and output shafts are perpendicular to each other, and the gears are mounted on the right-angle gearbox.
- Planetary gear motors: the gears are arranged in a planetary configuration, with the sun gear in the center and planet gears revolving around it.

Fig 11 Gear m

LCD WITH I2C

Fig 12 LCD with I2C module

LCD stands for **Liquid Crystal Display.** LCD is a flat-paneled display**.** It uses liquid crystals combined with polarized to display the content. LCD uses the light modulation property of LCD. LCD is available both in Monochrome and Multicolor. It cannot emit light directly without a backlight. In some LCDs, It displays the content only with the help of a backlight in a dark place.

I2C communication:

I2C or IIC stands for Inter-Integrated Communication. I2C is a serial communication interface to communicate with other I2C devices. I2C uses multi-master / multi slave method. I2C uses 2 lines named **SCL** and **SDA** for transmission/reception and another 2 lines for power supply and ground. Each and every I2C device has I2C address to identify. I2C addresses of multiple devices may have the same address. The address is in the format of “0x20” (Example address). Steps to find out I2C address device is discussed in the following (step 4).

- The serial Clock (SCL) pin is to synchronize the transmitter and receiver.
- Serial Data (SDA) pin is to transfer data.

3.2.8.1 ARDUINO MEGA 2560

Arduino Mega is based on ATmega2560 Microcontroller, an 8-bit AVR Architecture based MCU from ATMEL. It is available in a 100-pin Quad Flat Package.

It is designed and developed to provide more number of IO lines (both Digital and Analog), more flash memory and more RAM when compared to UNO. There is a Type-B USB connector on the left short edge of the board, which is used for powering on the board as well as programming the Microcontroller. There is also a 2.1 mm DC jack to provide external power supply.

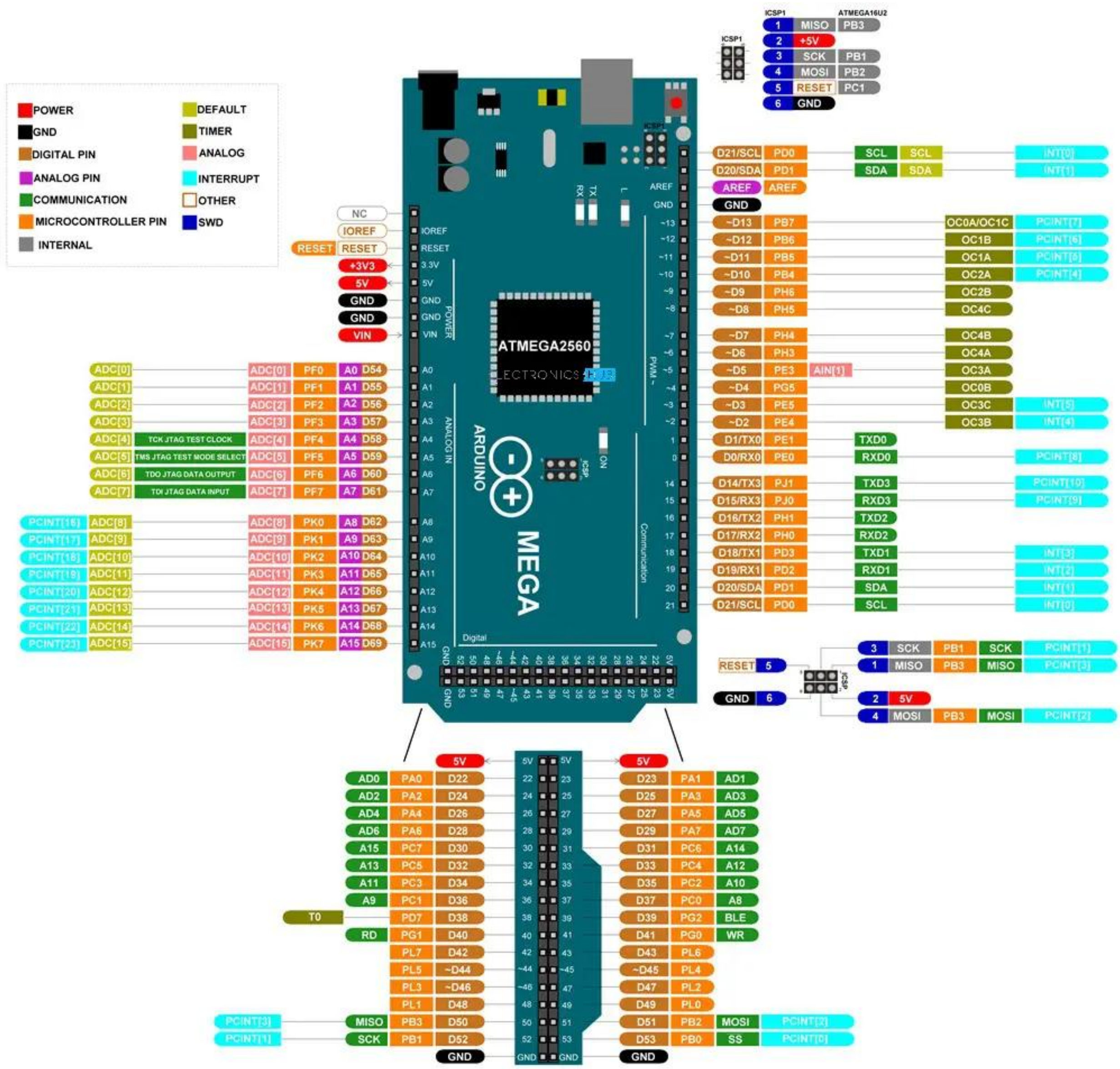
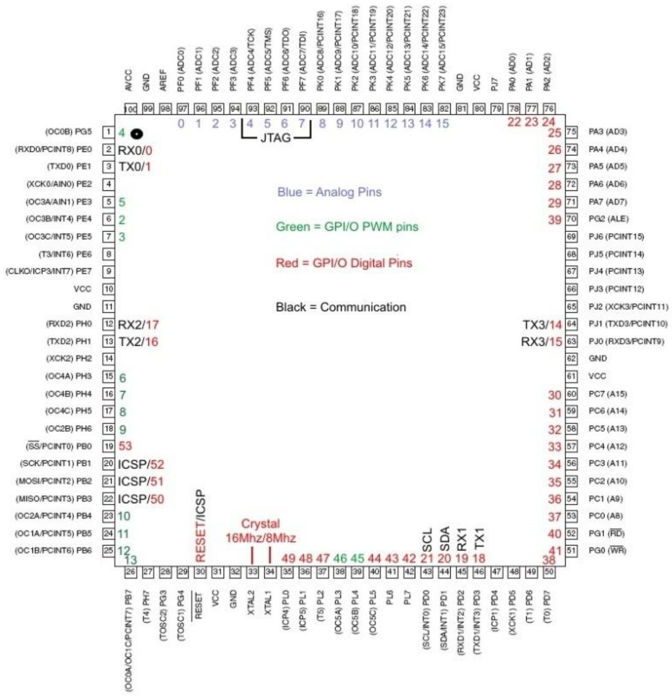


Fig 13 Architecture of Arduino

3.2.8.2 PIN DIAGRAM



14 Pin Diagram of Arduino

What are the Input and Output Pins of Arduino Mega?

Of the 86 pins available on the Mega board, 72 pins are associated with input and output. In that 54 pins (D0 to D53) are true digital IO pins, which can be configured as per you application using pinMode(), digitalWrite() and digitalRead() functions.

All these Digital IO pins are capable of sourcing or sinking 20mA of current (maximum 40mA is allowed). An additional feature of the Digital IO pins is the availability of internal pull-up resistor (which is not connected by default). The value of the internal pull-up resistor will be in the range of 20KΩ to 50KΩ.

There are also 16 Analog Input Pins (A0 to A15). All the analog input pins provide a 10-bit resolution ADC feature, which can be read using analogRead() function.

An important point about Analog Input pins is that they can be configured as Digital IO pins, if required.

Digital IO pins 2 – 13 and 44 – 46 are capable of producing 8-bit PWM Signals. You can use analogWrite() function for this.

Communication Interfaces on Arduino Mega

Arduino Mega supports three different types of communication interfaces. They are:

- Serial
- I2C or I^2C
- SPI

Perhaps the most common communication interface in the Arduino universe is the Serial Communication. In fact, the Arduino boards (UNO or Nano or Mega) are programmed using the serial communication.

Arduino Mega supports four hardware Serial Communication interfaces. Digital IO pins 0 and 1 are used as Serial RX0 and TX0 pins to receive and transmit serial data. These pins are connected to the serial pins of the on-board USB to Serial Converter IC.

Similarly. Digital IO pins 19 and 18 as RX1 and TX1, 17 and 16 as RX2 and TX2 and 15 and 14 as RX3 and TX3 respectively.

Digital IO Pins 20 and 21 can be configured as SDA (20) and SCL (21) to support I2C or I^2C or Two Wire Interface (TWI) communication.

3.3 SOFTWARE REQUIREMENTS

3.3.1 TRANSMITTER CODE

```
#include <Wire.h>
#include <MPU6050.h>

// Pin Definitions
const int switchPin = 7;           // Switch connected to pin 7
const int rfTransmitPin = 10;      // RF Transmitter data pin connected to pin 10
const int mq3Pin = A0;             // MQ-3 sensor connected to analog pin A0

// Constants
const int alcoholThreshold = 300;  // Set threshold based on calibration for safe alcohol levels
const int ACCEL_THRESHOLD = 50000000000; // Adjust based on sensitivity
const int GYRO_THRESHOLD = 5000000;   // Adjust for rotational sensitivity
const int ACCIDENT_DELAY = 2000;   // Delay to avoid repeated accident triggers
const int GSM_BAUD = 9600;

// Global Variables
MPU6050 mpu;
unsigned long lastAccidentTime = 0;
bool accidentDetected = false;

void setup() {
  Serial.begin(9600);            // Serial monitor
  Serial2.begin(GSM_BAUD);       // GSM module
  pinMode(switchPin, INPUT_PULLUP);  // Set switch pin to INPUT with internal pull-up
resistor
```

```
  pinMode(rfTransmitPin, OUTPUT);    // RF transmitter pin as output

  Wire.begin();
  mpu.initialize();

  // Check MPU6050 connection
  if (mpu.testConnection()) {
    Serial.println("MPU6050 connected successfully!");
  } else {
    Serial.println("MPU6050 connection failed!");
  }

  // Initialize SIM800L
  sendATCommand("AT");
  sendATCommand("AT+CMGF=1");     // Set SMS to text mode
  sendATCommand("AT+CNMI=1,2,0,0,0"); // Configure incoming SMS handling
  Serial.println("Setup complete.");
}

void loop() {
  int switchState = digitalRead(switchPin);
  int alcoholLevel = analogRead(mq3Pin);  // Read MQ-3 sensor value

  Serial.print("Switch state: ");
  Serial.println(switchState);  // 1 = not pressed, 0 = pressed
  Serial.print("Alcohol level: ");
  Serial.println(alcoholLevel);

  // Check if helmet is worn and alcohol level is below threshold
  if (switchState == LOW && alcoholLevel < alcoholThreshold) {
```

```
    digitalWrite(rfTransmitPin, LOW);  // Send LOW signal to RF transmitter
    delay(200);                        // Short delay to transmit the signal
    digitalWrite(rfTransmitPin, HIGH); // Reset RF transmitter
    Serial.println("Signal Sent - Helmet Worn and Safe Alcohol Level");
    Serial.println("Alcohol NOT Detected -Signal Sent");

  } else {
    digitalWrite(rfTransmitPin, HIGH); // Ensure RF transmitter is HIGH if conditions are not
met
    if (alcoholLevel >= alcoholThreshold) {
      Serial.println("Alcohol Detected - No Signal Sent");
    } else {
      Serial.println("Helmet Not Worn - No Signal Sent");
    }
  }

  // Accident Detection
  int16_t ax, ay, az, gx, gy, gz;
  mpu.getAcceleration(&ax, &ay, &az);
  mpu.getRotation(&gx, &gy, &gz);

  // Check if accident threshold is exceeded
  if (isAccident(ax, ay, az, gx, gy, gz)) {
    Serial.println("Accident detected! Sending alert...");
    sendAccidentAlert(); // Send alert without GPS location
  }

  delay(2000); // Adjust delay as needed
}
```

```
// Function to check if accident threshold is exceeded
bool isAccident(int16_t ax, int16_t ay, int16_t az, int16_t gx, int16_t gy, int16_t gz) {
  unsigned long currentTime = millis();

  if (currentTime - lastAccidentTime > ACCIDENT_DELAY) {
    if (abs(ax) > ACCEL_THRESHOLD || abs(ay) > ACCEL_THRESHOLD || abs(az) >
ACCEL_THRESHOLD ||
        abs(gx) > GYRO_THRESHOLD || abs(gy) > GYRO_THRESHOLD || abs(gz) >
GYRO_THRESHOLD) {

      lastAccidentTime = currentTime;  // Update last accident time
      accidentDetected = true;
      return true;
    }
  }
  return false;
}

// Function to send accident alert via SMS
void sendAccidentAlert() {
  if (accidentDetected) {
    sendSMS("Accident detected! Location not available.");
    accidentDetected = false; // Reset accident detection flag after sending alert
  }
}

// Function to send an SMS with specified message content
void sendSMS(String message) {
  Serial2.println("AT+CMGS=\"+919791471902\""); // Replace with the actual phone number
  delay(1000);
```

```
  Serial2.println(message);
  delay(1000);
  Serial2.write(26); // Send Ctrl+Z to indicate end of message
  updateSerial();
}

// Function to handle serial communication
void updateSerial() {
  delay(5000);

  // Forward data from Serial Monitor to SIM800L
  while (Serial.available()) {
    Serial2.write(Serial.read());
  }

  // Forward data from SIM800L to Serial Monitor
  while (Serial2.available()) {
    Serial.write(Serial2.read());
  }
}

// Function to send AT commands to the SIM800L and read the response
void sendATCommand(String command) {
  Serial2.println(command);
  delay(1000);
  updateSerial();
}
```

3.3.2 RECEIVER CODE

```
#include <LiquidCrystal.h>

const int rfReceivePin = 10;        // RF Receiver data pin connected to pin 10
const int relayPin = 8;             // Relay control pin connected to pin 8
const int alcoholSensorPin = A0;    // Alcohol sensor analog pin (MQ-3 sensor)
const int alcoholThreshold = 800;   // Threshold for alcohol level (calibrate this for your sensor)
const int stabilityChecks = 10;     // Number of stable readings required
const int delayBetweenChecks = 5;   // Delay between each stability check (ms)

LiquidCrystal lcd(12, 11, 5, 4, 3, 2); // Initialize LCD with the 4-bit parallel interface pins

void setup() {
  pinMode(rfReceivePin, INPUT);    // Set RF receiver pin as input
  pinMode(relayPin, OUTPUT);       // Set relay control pin as output
  digitalWrite(relayPin, HIGH);    // Ensure relay is off at startup

  lcd.begin(16, 2);                // Initialize LCD with 16 columns and 2 rows
  lcd.setCursor(0, 0);
  lcd.print("System Initializing");

  delay(100);
  lcd.clear();
  Serial.begin(9600);
}

void loop() {
  int stableSignalCount = 0;
```

```
// Check for a stable signal over multiple readings
for (int i = 0; i < stabilityChecks; i++) {
  int signal = digitalRead(rfReceivePin);
  if (signal == HIGH) {
    stableSignalCount++;
  } else {
    stableSignalCount = 0;  // Reset if any signal is LOW
  }
  delay(delayBetweenChecks);
}

int alcoholLevel = analogRead(alcoholSensorPin);  // Read alcohol sensor value
Serial.print("Alcohol Sensor Value: ");
Serial.println(alcoholLevel);

lcd.clear();

// Check if the helmet is worn (stable signal detected) and alcohol level is safe
if (stableSignalCount == stabilityChecks) {
  lcd.setCursor(0, 0);
  lcd.print("Helmet Worn");

  if (alcoholLevel < alcoholThreshold) {
    lcd.setCursor(0, 1);
    lcd.print("Safe Alcohol Level");
    Serial.println("Safe Alcohol Level");

    // Turn on the relay (motor runs) if the switch is pressed
    digitalWrite(relayPin, LOW);  // Activate relay
```

```
    Serial.println("Motor Running");
   } else {
    lcd.setCursor(0, 1);
    lcd.print("Alcohol Detected");
    digitalWrite(relayPin, HIGH); // Deactivate relay
    Serial.println("Alcohol Detected");
   }
  }
  // Display "Helmet Not Worn" if no signal detected
  else if (stableSignalCount == 0) {
   lcd.setCursor(0, 0);
   lcd.print("Helmet Not Worn");

   lcd.setCursor(0, 1);
   lcd.print("Motor Off");
   digitalWrite(relayPin, HIGH); // Deactivate relay, turning off the motor
   Serial.println("Helmet Not Worn - Motor Off");
  }

  delay(150);  // Update display every 1.5 seconds
}
```

CHAPTER-4

SYSTEM MODELLING AND DESIGN

4.1 CIRCUIT DIAGRAM

Our project is a smart helmet system that ensures rider safety by requiring helmet use and sobriety to start the bike. It features accident detection with emergency alerts, GPS tracking, and alcohol sensing, integrating RF modules, a GSM module, and relays to create a responsive, safety-focused riding experience.

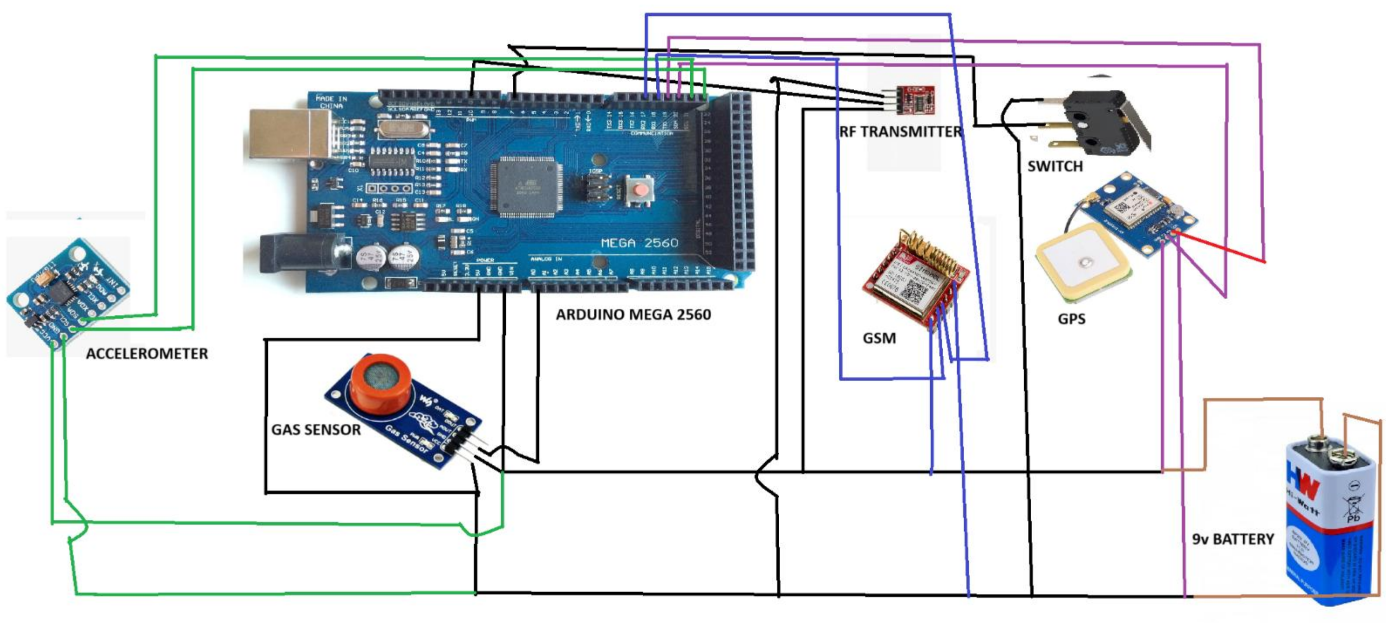


Fig 15 Circuit Diagram- Transmitter side

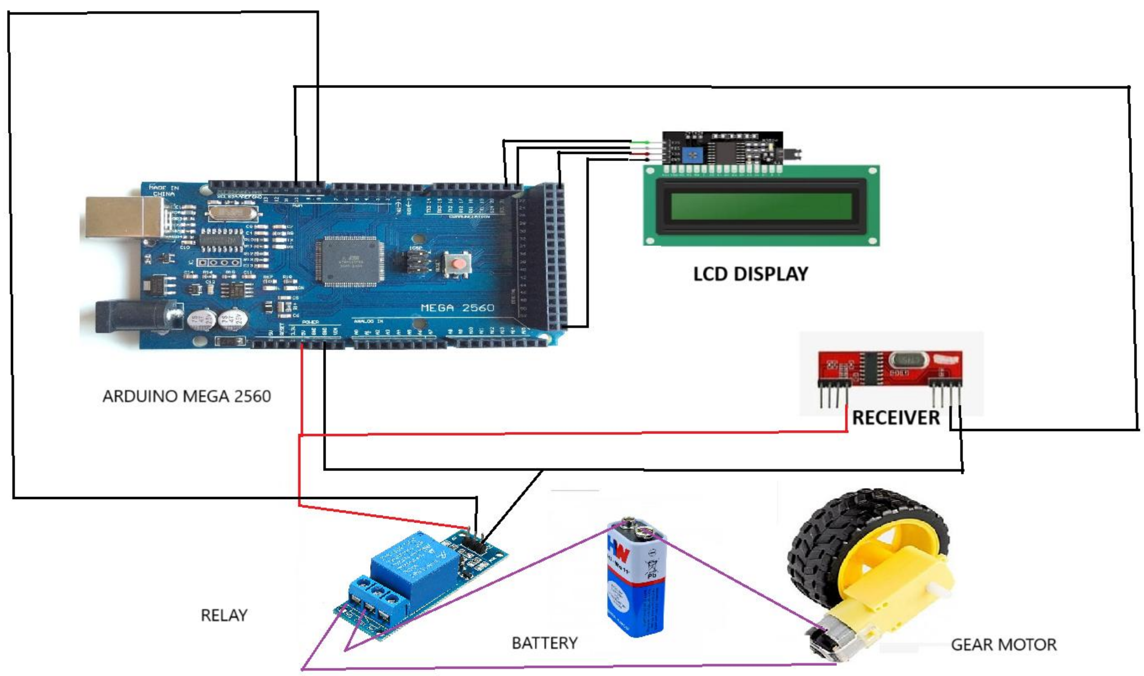


Fig 16 Circuit Diagram- Receiver side

4.2 WORKING

TRANSMITTER HELMET SIDE

The transmitter side is responsible for detecting helmet usage, monitoring alcohol levels, and accident detection. It sends signals to the receiver (bike) based on these conditions.

1. Components and Connections:

MQ-3 Alcohol Sensor: Detects the presence of alcohol. If it senses alcohol levels above a set threshold, it triggers a response to prevent the bike from starting.

Limit Switch: Placed inside the helmet to confirm that the helmet is worn. It sends a signal to the receiver only if the helmet is on.

Accelerometer (MPU6050): Detects sudden movements or impacts to sense an accident. If an accident is detected, it initiates an alert via GSM.

GSM and GPS Modules: Located on the helmet to send accident alerts (using GSM) and location data (using GPS).

RF Transmitter Module: Sends signals to the receiver (on the bike) to indicate whether the helmet is worn and the alcohol level is safe.

2. Operation:

Helmet Detection: If the limit switch is pressed (indicating the helmet is worn), the RF transmitter sends a "helmet worn" signal to the bike. If not, it sends a "no helmet" signal, preventing the bike from starting.

Alcohol Detection: The MQ-3 sensor checks for alcohol. If alcohol levels are above the threshold, the transmitter does not send the "safe to start" signal.

Accident Detection: The MPU6050 checks for sudden impacts or abnormal movement. If detected, it sends an SMS alert via GSM to notify a pre-set contact number of a possible accident, and the GPS provides the location.

RECEIVER SIDE(BIKE)

The receiver side on the bike processes signals from the transmitter and controls the ignition system based on the helmet’s status and alcohol level.

1. Components and Connections:

RF Receiver Module: Receives signals from the helmet's transmitter to check if the conditions to start the bike are met.

Relay Module: Controls the bike’s ignition. The relay is only activated if the "safe to start" signal is received, allowing the bike to start.

2. Operation:

Ignition Control: When the bike receives a signal confirming the helmet is worn and alcohol level is safe, the RF receiver allows the relay to close, enabling the bike’s ignition. If the conditions aren’t met, the relay stays open, preventing the bike from starting.

Safety Check: If either the helmet is not worn or alcohol is detected, the relay remains open, keeping the ignition off and ensuring the rider cannot start the bike without meeting safety conditions.

SUMMARY OF WORKFLOW

1. When the rider wears the helmet, the limit switch and alcohol sensor on the transmitter side check the status.

2. If conditions are safe, the RF transmitter sends a signal to the receiver, which closes the relay, allowing the bike to start.

3. If an accident occurs, the accelerometer detects it and triggers an SMS alert with location data via GSM and GPS

CHAPTER-5

RESULTS AND DISCUSSION

The smart helmet provides a detailed overview of how each module is connected, from the alcohol sensor to the accelerometer and relay system. It showcases the interactions between components and their respective roles, giving a clear picture of the safety measures integrated into the smart helmet.

On the bike, a relay, gear motor, LCD display, and audio-visual modules are connected, ensuring control over ignition and displaying safety messages to the rider.

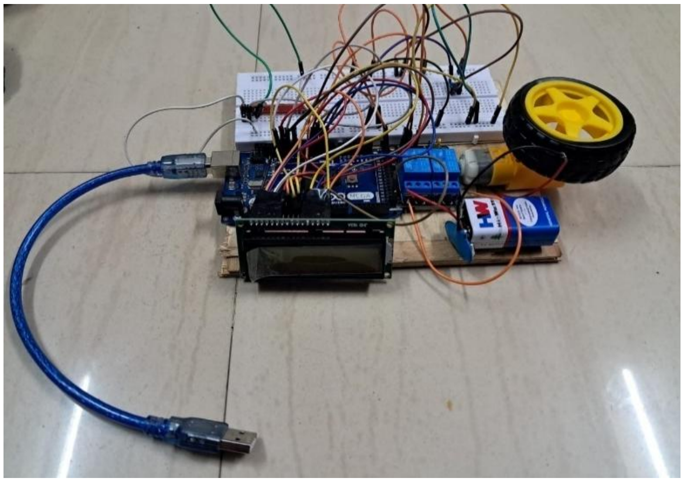


Fig 16 Smart helmet – receiver part

The top view highlights the arrangement of critical components like the GPS and GSM modules, securely placed on top of the helmet. This layout is essential for real-time location sharing and communication. With careful positioning, we've ensured both accessibility and protection for these components from external elements

Fig 17 top view Smart helmet

In the side view, the alcohol sensor is positioned near the mouth area, allowing for easy detection when rotated forward. The outer shell of the helmet is visible, but the internal components remain neatly enclosed and hidden from view.

Fig 18 Side View of helmet

The front view displays only the helmet’s outer structure, along with side plates. The GPS and GSM modules are mounted on top but are not visible from this angle.

Fig 19 Front View of helmet

5.1 IMPLEMENTATION AND TESTING

The MQ-3 sensor detects alcohol levels and, if alcohol is not present, starts the bike and displays a message on the LCD: "Alcohol not Detected”

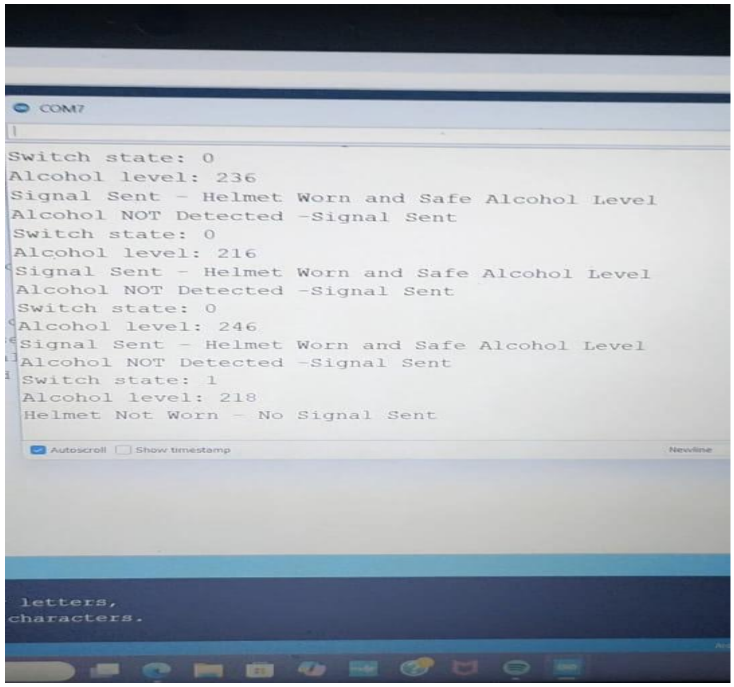


Fig 20Testing of alcohol detection and helmet worn

The MQ-3 sensor detects alcohol levels and, if alcohol is present, stops the bike and displays an alert message on the LCD: "Alcohol Detected."

Fig 21 Testing of alcohol detection and helmet not worn

5.2 SMS INDICATION

The GSM module is tested to ensure SMS is sent to notify emergency contacts during an accident and can receive messages for remote location requests.

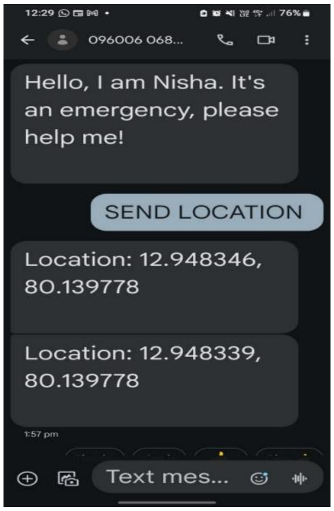


Fig 22 Testing of GPS and GSM Module indication due to Accelerometer

An accelerometer detects impact, triggering an SMS with GPS coordinates to predefined contacts for real-time location sharing on Google Maps.

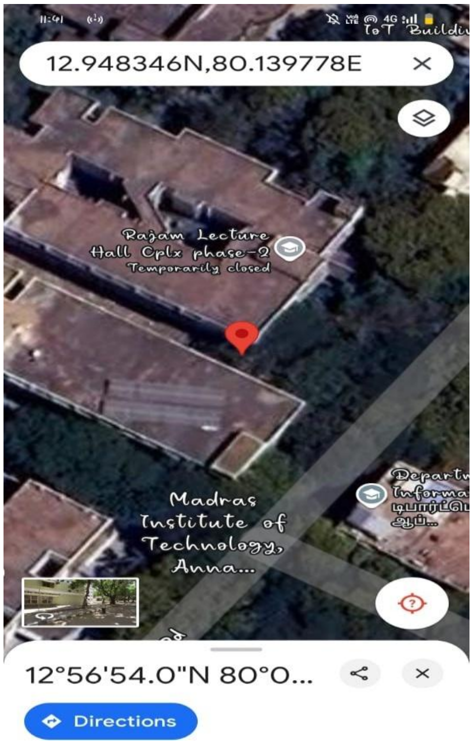


Fig 23 GPS location inferred from

CHAPTER – 6

CONCLUSION AND FUTURE WORKS

In conclusion, the Smart Helmet project successfully integrates essential safety features—helmet detection, alcohol detection, and crash detection with GPS location sharing—into a cohesive system aimed at enhancing rider safety. By automating preventive measures, such as blocking the bike's ignition if alcohol is detected or if the rider is not wearing a helmet, and enabling rapid response through real-time accident alerts, this smart helmet addresses many limitations of existing systems. The project not only enhances rider protection but also demonstrates the potential of embedded systems to create practical solutions in everyday life.

For future work, expanding the helmet's functionality through additional features such as fatigue detection, voice control, and advanced navigation assistance can further elevate rider safety. Implementing a fatigue sensor could alert drowsy riders, preventing accidents caused by exhaustion. Voice control for functions like emergency contact calling could provide hands-free options, ensuring rider focus. Future designs could also use lightweight materials and compact components to improve comfort and usability. Additionally, incorporating wireless connectivity could allow the helmet to sync with mobile apps, giving riders more control over settings and enabling software updates. Together, these advancements will create an even more versatile, comfortable, and responsive safety system for riders.

CHAPTER-7
REFERENCES

[1.] **Smart Helmet Integrated with Motorcycles to Support Rider Awareness and Safety Based Internet of Things**
2022 International Conference on ICT for Smart Society (ICISS)

[2.] https://lastminuteengineers.com/i2c-lcd-arduino-tutorial/

[3.] **Analysis of Smart helmets and Designing an IoT based smart helmet: A cost effective solution for Riders** Divyasudha N.;Arulmozhivarman P.;Rajkumar E.R. 2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT)

[4.] https://www.elprocus.com/mq3-alcohol-sensor/

[5.] **Smart Helmet and Accident Identification System** Valpadasu Hema;Allam Sangeetha;Soleti Navya;Ch. Nimisha Chowdary
2022 International Conference on Advancements in Smart, Secure and Intelligent Computing (ASSIC)