VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANA SANGAMA" BELAGAVI-590018, KARNATAKA

A Project Report
On

"MINEGUARD"

Submitted By

SAHANA N SHETTY	1BY23AI133
SANJANA A	1BY23AI141
ZUFSHAN NAAZ	1BY23AI194

Artificial Intelligence and Machine Learning / 3rd Semester
Course: Social Connect and Responsibility (BCSK307)

Under The Guidance Of
Dr. Bharathi M A
Professor and Head of Research and Consultancy Cell
Dept. Of AI&ML

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

B.M.S INSTITUTE OF TECHNOLOGY AND MANAGEMENT
Yelahanka, Bengaluru-560064
2024-2025

B.M.S INSTITUTE OF TECHNOLOGY AND MANAGEMENT

Yelahanka, Bengaluru-560064

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

CERTIFICATE

This is to certify that the project report entitled **MINE GUARD** is a Bonafide work carried out by **Ms. Sahana N Shetty** bearing **USN 1BY23AI133, Ms. Sanjana A** bearing **USN 1BY23AI141, Ms. Zufshan Naaz** bearing **USN 1BY23AI194,** during the year 2024-25. It is certified that all corrections / suggestions indicated have been incorporated in this report.

Signature of the Guide	**Signature of Cluster Head**	**Signature of HoD**
Dr. Bharathi M.A	**Dr. Pradeep K R**	**Dr. Anupama H S**
Professor and Head of Research and Consultancy Cell	Assistant Professor	HOD and Professor

B.M.S INSTITUTE OF TECHNOLOGY AND MANAGEMENT

Department of Artificial Intelligence and Machine Learning

DECLARATION

We, **Sahana N Shetty (1BY23AI133), Sanjana A (1BY23AI141), Zufshan Naaz (1BY23AI194)**; pursuing BE degree from **BMS Institute of Technology & Management**, Yelahanka, Bengaluru, hereby declare that the report entitled "**MINE GUARD**" carried out at **Artificial Intelligence and Machine Learning of BMSIT&M** is a Bonafide record of work done by us during the course of 2024-25 of BE program and all the contents are prepared and presented by us.

Place: Bengaluru

Date: 15-01-2025

TABLE OF CONTENTS

Chapter No.		Title	Page No.
1.		**INTODUCTION**	
	1.1	Overview	**1-2**
	1.2	Objectives of the Project	**2**
2.		**PROBLEM DEFINITION**	
	2.1	Problem Statement	**3**
3.		**LITERATURE SURVEY**	**4-5**
4.		**SYSTEM REQUIREMENT SPECIFICATIONS**	
	4.1	Hardware Requirements	**6**
	4.2	Software Requirements	**7**
		4.2.1 Arduino IDE	**7**
		4.2.2 Blynk Application	**7**
5.		**DESIGN**	
	5.1	System Architecture	**8-9**
6.		**IMPLEMENTATION**	
	6.1	Modules Used	**10-13**
		6.1.1 Software Serial	**10**
		6.1.2 DFRobotDFPlayerMini	**10**
		6.1.3 DHT	**11**
		6.1.4 Wire	**11**
		6.1.5 TinyGPS++	**11-12**
		6.1.6 Queue Array	**12**
		6.1.7 Wifi	**12**
		6.1.8 BlynkSimpleEsp32	**13**
	6.2	Coding	**13-19**
		6.2.1 Code for Arduino Mega 2560	**13-17**
		6.2.2 Code for ESP32	**17-19**
	6.3	Connection	**19-20**
7.		**PROTOTYPE**	**21-22**
8.		**RESULT**	**23-24**
9.		**CONCLUSION**	**25**
10.		**FUTURE ENHANCEMENTS**	**26**
		REFERENCES	**27**

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose constant encouragement and guidance has been a source of inspiration throughout the course.

Our sincere gratitude to our guide, **Dr.Bharathi M. A ,** Professor and Head of Research and Consultancy Cell, Department of Artificial Intelligence and Machine Learning, for her valuable time, patience, suggestion and periodic evaluation that was conducive for the project as a guide/Course coordinator.
We sincerely thank **Dr. Pradeep K R,** Assistant Professor and Cluster Head, Department of Artificial Intelligence and Machine Learning, BMSIT&M, for his valuable support and constant encouragement given to us during this work.
We sincerely thank **Dr. Anupama H S**, Professor and HOD, Department of Artificial Intelligence and Machine Learning, BMSIT&M, for her valuable support and constant encouragement given to us during this work.
It is our privilege and pleasure to express our profound sense of respect, gratitude and indebtedness to **Dr. Sanjay H A**, Principal, BMSIT&M, for guiding and providing facilities for the successful completion of our project.

Last but not the least, we wish to acknowledge our Parents and Friends for giving us moral strength and encouragement.

Sahana N Shetty

Sanjana A

Zufshan Naaz

LIST OF FIGURES

Figure No.	Figure Name	Page No.
5.1	Block Diagram	8
5.2	Circuit Diagram	9
6.1	Flow Chart	20
7.1	Assembled Circuit	21
8.1	Blynk	24
8.2	Output	24
	(a) Gas Warning	
	(b) Proximity Warning	

LIST OF TABLES

Table No.	Table Name	Page No.
3.1	Literature Survey	4-5
8.1	Result	23

Chapter-1

INTRODUCTION

1.1 Overview

The mining sector is one of the most critical industries, contributing significantly to global economic development. It provides essential raw materials for various industries, including construction, manufacturing, and energy production. Despite its importance, mining is considered one of the most hazardous occupations. Miners often face extreme environmental conditions, including toxic gases, high temperatures, and poor visibility, alongside physical risks such as cave-ins and equipment malfunctions. These challenges not only pose serious threats to miners' health and safety but also impact operational efficiency and productivity.

Miners are exposed to dangers like inhalation of toxic gases such as methane and carbon monoxide, extreme heat or humidity leading to dehydration or heatstroke, and reduced oxygen levels in confined spaces. Additionally, long working hours and the physically demanding nature of the job can result in fatigue and cardiovascular stress. In emergency scenarios like gas leaks, explosions, or structural collapses, the lack of real-time monitoring and communication further exacerbates the risks, often delaying rescue operations.

To address these challenges, the MineGuard Project provides an innovative and comprehensive solution. This smart helmet is designed to ensure miners' safety by monitoring environmental conditions, tracking health parameters, and enabling real-time alerts and communication. By integrating advanced sensors, communication modules, and a mobile application interface, MineGuard enhances safety measures and provides miners with immediate feedback during emergencies.

The helmet is equipped with multiple components to tackle various safety and health risks. Gas sensors (MQ2 and MQ7) detect the presence of toxic and flammable gases, while the DHT22 sensor measures temperature and humidity levels to ensure optimal working conditions. An ultrasonic sensor helps detect obstacles, reducing the risk of collisions in low-visibility environments. The pulse sensor monitors the heart rate of miners, providing early warnings for abnormal readings. For location tracking, the Neo 6M GPS module ensures precise positioning, crucial during rescue missions.

Communication is facilitated by the ESP32 module and GSM 800, which transmit data to the control room and send alerts via SMS. The Blynk application provides a user-friendly interface to monitor miners' status in real-time. Audible warnings are generated using the DFPlayer mini and a speaker, alerting miners to imminent dangers. The entire system is powered by a 12V lithium battery, with an LM2596 module to regulate power supply efficiently, ensuring reliable operation in remote areas.

In summary, the MineGuard Project combines advanced technology and user-focused design to address the critical safety issues faced by miners. By providing real-time monitoring, communication, and alerts, it aims to significantly reduce risks, enhance operational safety, and safeguard the lives of miners working in hazardous environments.

1.2 Objectives of the project

1. To monitor miners' safety in real-time by detecting hazardous environmental conditions
2. To track miners' vital signs, including heart rate, ensuring their health and safety.
3. To provide immediate alerts via SMS and a mobile application for emergencies.
4. To notify miners of hazardous situations directly through audible alerts.

Chapter-2

PROBLEM DEFINITION

2.1 Problem Statement

The smart helmet for miners addresses critical safety challenges faced in underground mining environments, where traditional helmets fall short in providing real-time monitoring and hazard detection. Miners work in hazardous conditions, often exposed to risks like toxic gas leaks (e.g., methane and carbon monoxide), poor ventilation, extreme temperatures, physical exhaustion, and the threat of falling debris. Conventional helmets only provide basic head protection, leaving miners vulnerable to these environmental and health risks.

Emergency situations are another challenge, as miners may be unable to communicate in the event of an accident. The smart helmet can include GPS and fall detection systems, enabling automatic alerts and precise location tracking to accelerate rescue efforts. Hence the problem statement :

“Smart helmet for miners with DFPlayer and Blynk for real-time alerts and monitoring.”

Chapter-3

LITERATURE SURVEY

Sl No	Publication	Methodology	Result	Remarks
1.	Amjad Chohan, Mir Sajjad Hussain Talpur, "Smart Helmet For Coal Miners", presented at the International Journal of Computational Intelligence in Control, Vol: 13,2021.	Integrating sensors to monitor the miners health. Communication is enabled through an ESP32 and GSM module.	The smart helmet enhanced miner safety by providing real-time monitoring of hazardous conditions, improving rescue operations through GPS tracking, and ensuring user acceptance with a mobile application interface.	Challenges such as environmental variability, data security concerns, sensor maintenance and the helmet's comfort need to be addressed to improve the system's overall effectiveness.
2.	S. Gaidhane, "Smart Helmet for Coal Miners Using GSM Module Interfaced with RF",presented at the AIP Conference Proceedings,Vol:279, 2023	Utilizes a GSM module and RF interfacing to create a smart helmet system that monitors environmental conditions, while transmitting real-time data to a control center.	It improved overall safety management by providing remote monitoring capabilities to supervisors.	Challenges with signal reliability in remote underground areas, and the helmet's bulk could affect comfort during extended use, leading to potential discomfort for miners.

3.	Abhishek Kr Dubey, Sumit Singh, Deepak Sahu, "Coal Mine Safety Monitoring and Alerting System with Smart Helmet", presented at the Spectrum of Emerging Sciences conference,Vol:4, 2024	Integrated a GPS module for tracking miners' health and location in real-time. The system uses an ESP32 microcontroller for wireless communication.	The GPS module enabled efficient tracking of miners' locations, significantly improving emergency response time.	Challenges include limited battery life, potential signal interference in deep underground mines, and the need for regular calibration of sensors to maintain accuracy.
4.	Sapna Patel, Ayush Goyani, Risit Dhameliya, Bhavik Naiya, "Smart Helmet for Mining Workers",presented at the International Journal for Research in Applied Science and Engineering Technology,Vol:3, 2023	The project develops a smart helmet for mining workers by integrating sensors The system transmits real-time data using a microcontroller, and emergency alerts are sent via GSM to a control room, with GPS tracking for locating miners during emergencies.	The system successfully detected gas leaks, temperature anomalies, and abnormal heart rates, sending timely alerts to the control room and enabling immediate action. GPS functionality enhanced emergency response by tracking the miners precise locations.	Challenges included the need for more reliable communication in deep underground environments, where GSM signals can be weak,
5.	B. Kartik, Dr. Manimaran P, "IoT Based Smart Helmet for Hazard Detection in Mining Industry", presented at the arXiv Conference,2023	The project implements an IoT-based smart helmet that utilizes a variety of sensors, to continuously monitor the safety and health of miners. Data from these sensors is transmitted wirelessly via an ESP32.	The system effectively detected hazardous gases, high temperatures, and abnormal health conditions in miners, sending alerts to the control room in real-time.	System's dependency on GSM for communication, which may be unreliable in underground environments with weak signals, and the bulkiness of the helmet, which could impact miner comfort.

Chapter - 4

SYSTEM REQUIREMENT SPECIFICATIONS

A software requirements specification (SRS) is a comprehensive description of the intended purpose and environment for software under development. The SRS fully describes what the software will do and how it will be expected to perform. Software requirements specification permits a rigorous assessment of requirements before design can begin and reduces later redesign. It should also provide a realistic basis for estimating product costs, risks, and schedules.

The software requirements specification document enlists enough and necessary requirements that are required for the project development. To derive the requirements we need to have clear and thorough understanding of the products to be developed or being developed. This is achieved and refined with detailed and continuous communications with the project team and customer till the completion of the software.

4.1 HARDWARE REQUIREMENTS:

- Arduino Mega 2560
- DFPlayer Mini
- Speaker
- 12v lithium-ion battery
- MQ2 and MQ7 Gas sensor
- DHT22 Temperature and Humidity sensor
- Ultrasonic sensor
- Heart Rate Sensor (pulse sensor)
- ESP32
- GSM 800
- LM2596 Module
- NEO 6M GPS module

4.2 SOFTWARE REQUIREMENTS:

We use the following software requirements: -

- Arduino IDE
- BLynk application

4.2.1 Arduino IDE (Integrated Development Environment):

The Arduino IDE is a software platform used to write, compile, and upload code to Arduino boards. It supports a variety of boards and microcontrollers, allowing developers to create projects for a wide range of applications.

Key features:

- **Code Editor**: Write programs in C/C++ using a simple text editor.
- **Compiler**: Converts the code into machine-readable instructions.
- **Uploader**: Sends the compiled program to the connected Arduino board via USB.
- **Serial Monitor**: Allows communication with the board, useful for debugging.

4.2.2 BLynk Application:

Blynk is an IoT platform that enables users to control and monitor hardware (like Arduino, ESP8266, ESP32) through a mobile app. It offers an easy-to-use interface for building custom dashboards with widgets such as buttons, sliders, and displays. The platform connects the app to the hardware via cloud or local servers, allowing real-time data monitoring and control from anywhere. Users create projects in the Blynk app, generate authentication tokens, and link them to their hardware via simple code.

Chapter - 5

DESIGN

5.1 System Architecture:

System architecture is a conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.

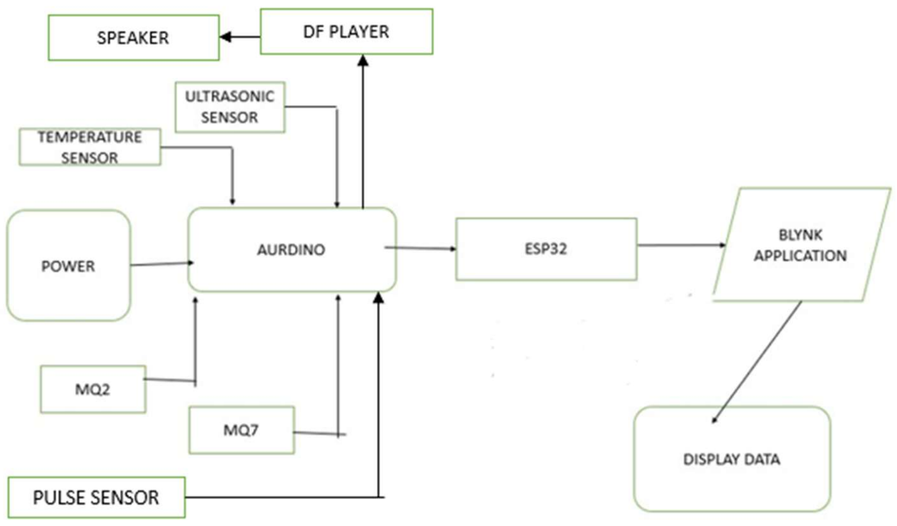


Fig 5.1

The MineGuard system architecture is designed to ensure the safety and health of miners by integrating various sensors, communication modules, and a user interface to monitor and alert in real-time.

1. **Sensors**:

- **Gas Sensors (MQ2, MQ7)**: Detect hazardous gases like methane and carbon monoxide.
- **Temperature and Humidity Sensor (DHT22)**: Monitors the environmental conditions inside the mine.
- **Pulse Sensor**: Monitors the heart rate of the miner to detect any abnormal readings.
- **Ultrasonic Sensor**: Detects obstacles in low-visibility areas.

2. **Microcontroller (Arduino Mega 2560)**:

   The Arduino Mega 2560 acts as the central processing unit, gathering data from all sensors and performing necessary calculations. It also enables communication with other modules for data transmission and alerts.

3. **Communication**:

   - **GSM Module (SIM800L)**: Sends SMS alerts to the control room or other predefined recipients in case of emergencies or hazard detection.
   - **Mobile Application (Blynk)**: Provides a user-friendly interface for real-time monitoring of the miner's health and environmental conditions.

4. **Power Supply**:

   A **12V lithium-ion battery** with an **LM2596 module** is used to power the entire system, ensuring continuous operation during long shifts.

5. **Alert Mechanism**:

   In case of hazardous conditions (e.g., high gas levels, abnormal heart rate, or temperature), the system immediately triggers alerts through both audible alarms in the helmet and SMS notifications to the control center.

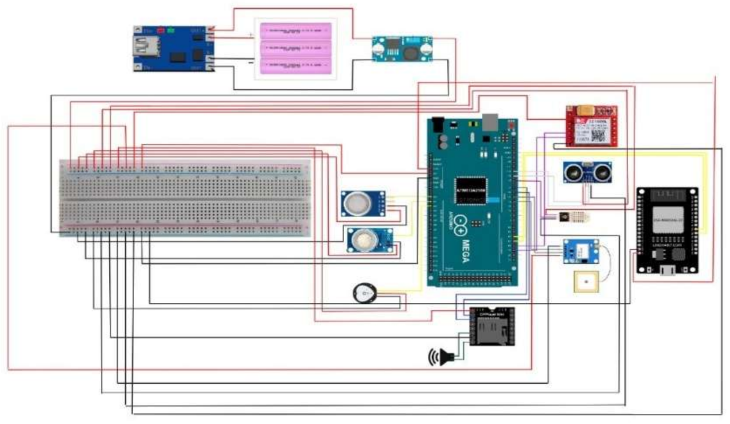


Fig 5.2

Chapter-6

IMPLEMENTATION

6.1 Modules Used:

6.1.1 Software Serial

The SoftwareSerial library allows serial communication on other digital pins of an Arduino board, using software to replicate the functionality. It is possible to have multiple software serial ports with speeds up to 115200 bps. A parameter enables inverted signaling for devices which require that protocol.

This is particularly useful when the hardware UART ports are already in use or are limited in number, as is the case with many Arduino boards. By using the SoftwareSerial library, additional serial communication channels can be created, enabling the microcontroller to communicate with multiple devices such as GSM modules, GPS modules, or other serial peripherals simultaneously. It achieves this by emulating serial communication in software, which involves bit-by-bit transmission and reception of data.

Although it is less efficient than hardware-based UART, SoftwareSerial is invaluable for expanding the communication capabilities of microcontrollers with limited hardware resources. The library supports features like setting baud rates and swapping RX and TX pins for flexibility.

6.1.2 DFRobotDFPlayerMini

The DFRobotDFPlayerMini library is an Arduino-compatible library designed for interfacing with the DFPlayer Mini MP3 module. This compact and cost-effective MP3 module is widely used in projects requiring audio playback functionality. The library simplifies the process of controlling the module, enabling users to play audio files stored on a microSD card with ease. It provides a wide range of features, including playing specific tracks, pausing, stopping, adjusting volume, and even looping audio files. Additionally, it supports advanced features such as playing files in random order or managing audio folders for organized playback.

Communication between the Arduino and the DFPlayer Mini is achieved through a serial interface, and the library works seamlessly with both hardware and software serial communication.This library is ideal for projects that require user feedback, voice alerts such as smart helmets, and home automation solutions.

6.1.3 DHT

The DHT library is a widely used Arduino library designed to interface with DHT series sensors, such as the DHT11 and DHT22, which measure temperature and humidity. This library simplifies the process of retrieving data from these sensors, making it accessible even for beginners in electronics and programming. The DHT library provides functions to initialize the sensor, read temperature in both Celsius and Fahrenheit, and measure relative humidity with reasonable accuracy.

It handles the timing and signal processing required to interpret the data sent by the sensor, which operates on a single-wire communication protocol. The DHT22 offers higher accuracy and a broader measurement range compared to the DHT11, but both sensors are commonly used in projects like weather stations, environmental monitoring systems, and IoT devices. With the DHT library, users can focus on building applications rather than dealing with the complexities of low-level communication, making it a reliable choice for integrating temperature and humidity sensing capabilities into Arduino-based projects.

6.1.4 Wire

The Wire library is a core Arduino library that facilitates communication with devices using the I2C (Inter-Integrated Circuit) protocol. I2C is a popular two-wire communication protocol that allows multiple devices to communicate with a microcontroller using only two lines: a data line (SDA) and a clock line (SCL). The Wire library simplifies the implementation of I2C communication by providing an easy-to-use interface for both master and slave devices.

It includes functions to initialize the bus, send data, request data, and handle communication errors. The library is particularly useful for connecting peripherals like sensors, displays, real-time clocks, and EEPROMs, which often operate on the I2C protocol. It supports features like addressing individual devices through unique 7-bit or 10-bit addresses, enabling communication with multiple devices on the same bus. Its versatility and ease of use make the Wire library an essential tool for Arduino projects involving multi-device integration.

6.1.5 TinyGPS++

The TinyGPS++ library is a lightweight and powerful Arduino library designed to decode NMEA sentences from GPS modules, making it easier to work with GPS data in embedded projects. It supports a wide range of features, including parsing latitude, longitude, altitude, speed, date, time, and the number of satellites in view. TinyGPS++ is highly efficient and optimized for small microcontrollers, offering robust functionality.

The library is compatible with most GPS modules that use serial communication, such as the Neo-6M, and provides a user-friendly interface to access and utilize GPS data. It also includes features for customizing data formats, handling time zones, and validating GPS signals to ensure accurate data. This makes TinyGPS++ an ideal choice for navigation systems, vehicle tracking, geofencing, and other projects requiring real-time location information. With its versatility and ease of use, TinyGPS++ significantly simplifies the integration of GPS functionality into Arduino-based applications.

6.1.6 Queue Array

The **QueueArray** library in Arduino is a lightweight and easy-to-use library designed to implement queue (FIFO - First In, First Out) data structures in Arduino projects. A queue allows data to be stored and processed in the order it was received, making it ideal for applications where sequential processing is critical, such as buffering data streams or managing tasks.

The QueueArray library provides simple functions to enqueue (add), dequeue (remove), peek (view the front element without removing it), and check the size of the queue. It is particularly useful in scenarios where data needs to be temporarily stored and processed in a controlled manner, such as in sensor data handling, communication protocols, or task scheduling systems. The library is optimized for Arduino's constrained environment, offering reliable performance while keeping memory usage minimal.

6.1.7 Wifi

The WiFi library in Arduino provides a simple yet powerful way to add wireless connectivity to projects, enabling devices to connect to Wi-Fi networks for communication and data exchange. This library is designed to work seamlessly with Wi-Fi-enabled hardware like the ESP8266, ESP32, and Arduino Wi-Fi Shields, allowing microcontrollers to interact with the Internet or local networks. It supports a wide range of features, including connecting to secured networks, setting static or dynamic IP addresses, and sending or receiving data via TCP/UDP protocols.

The library also enables HTTP communication, making it ideal for IoT applications like home automation, data logging, and real-time monitoring. With functions to scan available networks, handle disconnections, and configure access points, the WiFi library simplifies the process of building connected devices. Its flexibility and ease of use empower developers to create projects that leverage cloud services, APIs, or peer-to-peer communication, transforming standalone systems into interconnected solutions.

6.1.8 BlynkSimpleEsp32

The BlynkSimpleEsp32 library is a part of the Blynk ecosystem, specifically designed to enable seamless integration of ESP32-based microcontrollers with the Blynk platform. This library provides an efficient way to connect ESP32 devices to the Internet and interact with the Blynk app for IoT applications. With its user-friendly interface, the BlynkSimpleEsp32 library supports a wide range of functionalities, such as real-time control, data visualization, and remote monitoring of sensors and actuators.

It simplifies the process of establishing a Wi-Fi connection, authenticating the device with a unique token, and synchronizing data between the hardware and the Blynk cloud server. The library supports advanced features like virtual pins for custom control, timers for periodic tasks, and notifications for real-time alerts.

6.2 Coding:

6.2.1 Code for Arduino Mega 2560

```
// GPS (SoftwareSerial)
TinyGPSPlus gps;
SoftwareSerial gpsSerial(3, 2); // RX, TX for Neo 6M
SoftwareSerial mySoftwareSerial(6, 5); // RX, TX for DF Player Mini
SoftwareSerial gsmSerial(18, 19); // RX, TX for GSM module

DFRobotDFPlayerMini myDFPlayer; // Create an instance of the DFPlayer Mini
 // DHT22 Setup
DHT dht(DHTPIN, DHTTYPE);

// Queue to store alert events
QueueArray<int> alertQueue(1000);

void setup() {
   Serial.begin(9600);       // Main Serial (Monitor)
   Serial2.begin(9600);      // Start Serial communication with ESP32
   gpsSerial.begin(9600);      // GPS communication
   mySoftwareSerial.begin(9600); // Communication with DFPlayer Mini
   gsmSerial.begin(9600);       // GSM communication
   dht.begin();
   pinMode(ULTRASONIC_TRIG, OUTPUT);
```

```
  pinMode(ULTRASONIC_ECHO, INPUT);
  pinMode(HEART_RATE_PIN, INPUT);

  Serial.println("Arduino Mega is ready.");

  // Initialize DFPlayer Mini
  if (myDFPlayer.begin(mySoftwareSerial)) {
    Serial.println("DFPlayer Mini ready.");
    myDFPlayer.volume(30); // Set volume (0-30)
    myDFPlayer.stop();    // Ensure DFPlayer is stopped on initialization

 }

  // Initialize GSM Module
  gsmSerial.println("AT");   // Test GSM module
  delay(1000);
  gsmSerial.println("AT+CMGF=1"); // Set SMS mode to text
  delay(1000);
  gsmSerial.println("AT+CSCS=\"GSM\""); // Set character set
  delay(1000);
  Serial.println("GSM module ready.");
}

void loop() {
  // Sensor readings
  int gas1 = analogRead(MQ2_PIN);
  int gas2 = analogRead(MQ7_PIN);
  float temperature = dht.readTemperature();
  float humidity=dht.readHumidity();
  int distance = readUltrasonic();
  int heartRate = analogRead(HEART_RATE_PIN);

  // Build the data string with actual sensor values
  String data = String(gas1) + "," + String(gas2) + "," + String(temperature) + "," + String(humidity) + "," +
String(distance) + "," + String(heartRate) + "," + readGPS() + "\n";
```

```
  Serial2.print(data);    // Send data to ESP32

  // Print sensor readings for debugging
  Serial.println("Gas1: " + String(gas1));
  Serial.println("Gas2: " + String(gas2));
  Serial.println("Temperature: " + String(temperature));
  Serial.println("Distance: " + String(distance));
  Serial.println("HeartRate: " + String(heartRate));

  // Trigger alerts based on conditions
  addAlertIfNeeded(gas1 > 400 || gas2 > 400, ALERT_GAS);
  addAlertIfNeeded(heartRate > 120, ALERT_HEART_RATE);

addAlertIfNeeded(temperature > 20, ALERT_TEMPERATURE); // Adjust threshold as needed
  addAlertIfNeeded(distance > 0 && distance < 300, ALERT_OBSTACLE);

  // Process alerts sequentially
  processAlerts();

  delay(100); // Stability delay
}
// Alert handlers
void handleGasAlert() {
  Serial.println("Handling Gas Alert!");
  myDFPlayer.play(2);
  sendSMS("Dangerous gas levels detected!");
}

void handleHeartRateAlert() {
  Serial.println("Handling Heart Rate Alert!");
  myDFPlayer.play(4);
  sendSMS("High Heart Rate Detected!");
}

void handleTemperatureAlert() {
  Serial.println("Handling Temperature Alert!");
  myDFPlayer.play(1);
  sendSMS("High Temperature Detected!");
}

void handleObstacleAlert() {
  Serial.println("Handling Obstacle Alert!");
  myDFPlayer.play(3);
  sendSMS("Obstacle Detected Close!");
}

// Function to read GPS data (not used in current implementation)

String readGPS() {
  String googleMapsLink = "";
  while (gpsSerial.available() > 0) {
    char c = gpsSerial.read();
    gps.encode(c);
    Serial.print(c);  // Debugging: Print raw GPS data to Serial Monitor
  }
  if (gps.location.isValid()) {
    float latitude = gps.location.lat();
    float longitude = gps.location.lng();
    Serial.print("Latitude: ");
    Serial.println(latitude, 6);
    Serial.print("Longitude: ");
    Serial.println(longitude, 6);
  } else {
    Serial.println("Waiting for GPS signal...");
    googleMapsLink = "No GPS Signal";
  }
  return googleMapsLink;
}

// Function to send SMS using GSM module
```

```
void sendSMS(String message) {
   gsmSerial.println("AT+CMGS=\"+9449462809\""); // Replace with recipient's phone number
   delay(1000);
   gsmSerial.println(message); // Send the message content
   delay(1000);
   gsmSerial.write(26); // ASCII code for CTRL+Z to send the SMS
   delay(1000);
   Serial.println("SMS Sent: " + message);
}
```

6.2.2 Code for ESP32: -

```
#define BLYNK_TEMPLATE_ID "TMPL3EkJ2VJjq"
#define BLYNK_TEMPLATE_NAME "Smart helmet for miners"
#define BLYNK_PRINT Serial
#include <WiFi.h>
#include <BlynkSimpleEsp32.h>

char auth[] = "nxCreISOS14U55pK87TZu52MNHa4KS71"; // Blynk Auth Token
char ssid[] = "sanjana";  // Your WiFi SSID
char pass[] = "9449462809"; // Your WiFi Password

String receivedData = ""; // Data buffer for received data
BlynkTimer timer;

// Function to send data to Blynk
void sendToBlynk() {
  if (receivedData.length() > 0) {
    String data = receivedData;
    receivedData = ""; // Clear buffer

    // Debugging: Print received data to the Serial Monitor
    Serial.println("Received Data: " + data);

    // Parse data (assuming the data is comma-separated)
    int gas1 = data.substring(0, data.indexOf(',')).toInt();
    data = data.substring(data.indexOf(',') + 1);

    int gas2 = data.substring(0, data.indexOf(',')).toInt();
    data = data.substring(data.indexOf(',') + 1);

    float temperature = data.substring(0, data.indexOf(',')).toFloat();
    data = data.substring(data.indexOf(',') + 1);

    float humidity = data.substring(0, data.indexOf(',')).toFloat();

data = data.substring(data.indexOf(',') + 1);

    int distance = data.substring(0, data.indexOf(',')).toInt();
    data = data.substring(data.indexOf(',') + 1);

    int heartRate = data.substring(0, data.indexOf(',')).toInt();
    data = data.substring(data.indexOf(',') + 1);
    String gpsLink  = data; // Remaining part of data is GPS information

    // Debugging: Print parsed values to the Serial Monitor
    Serial.print("Gas1: ");
    Serial.println(gas1);
    Serial.print("Gas2: ");
    Serial.println(gas2);
    Serial.print("Temperature: ");
    Serial.println(temperature);
    Serial.print("Humidity: ");
    Serial.println(humidity);
    Serial.print("Distance: ");
    Serial.println(distance);
    Serial.print("HeartRate: ");
    Serial.println(heartRate);
    Serial.print("GPS Link: ");
    Serial.println(gpsLink);
```

```
// Read data from Arduino Mega via Serial
void readFromMega() {
  while (Serial2.available()) {
    char c = Serial2.read();
    receivedData += c;
    if (c == '\n') {
      // Debugging: Print the received data to Serial Monitor
      Serial.print("Received from Mega: ");
      Serial.println(receivedData);
      break;  // End of data packet
    }

  }
}
void setup() {
  Serial. Begin(9600);        // Start Serial Monitor for debugging
  Serial2.begin(9600, SERIAL_8N1, 16, 17); // Start communication with Arduino Mega (RX2, TX2)
  Blynk.begin(auth, ssid, pass); // Connect to Blynk server
  timer.setInterval(1000L, sendToBlynk); // Send data every 1 second
}
```

6.3 Connections:

1.Hardware Integration:

- The Arduino Mega 2560 served as the central controller, interfacing with various sensors and modules:
  - MQ2 and MQ7 for gas detection.
  - DHT22 for temperature and humidity monitoring.
  - Ultrasonic sensor for obstacle detection.
  - Pulse sensor for monitoring heart rate.
- Communication modules, including ESP32 (for wireless data transmission to the Blynk app) and GSM 800 (for SMS alerts), were connected.
- A Neo 6M GPS module was integrated for real-time location tracking.
- A DFPlayer Mini and speaker were used for immediate audio alerts.

2.Power Supply:

- The system was powered using a 12V lithium battery, with an LM2596 module regulating the voltage to required levels.

3.Software and Communication:

- The Arduino was programmed in the Arduino IDE.
- Sensor data was processed and transmitted using the Blynk app and SMS alerts to notify remote operators of hazardous conditions.
- Real-time location tracking and environmental data monitoring ensured continuous situational awareness.

4.Prototype Testing:

- The helmet was tested to validate functionality, including gas detection, obstacle alerting, and communication reliability.

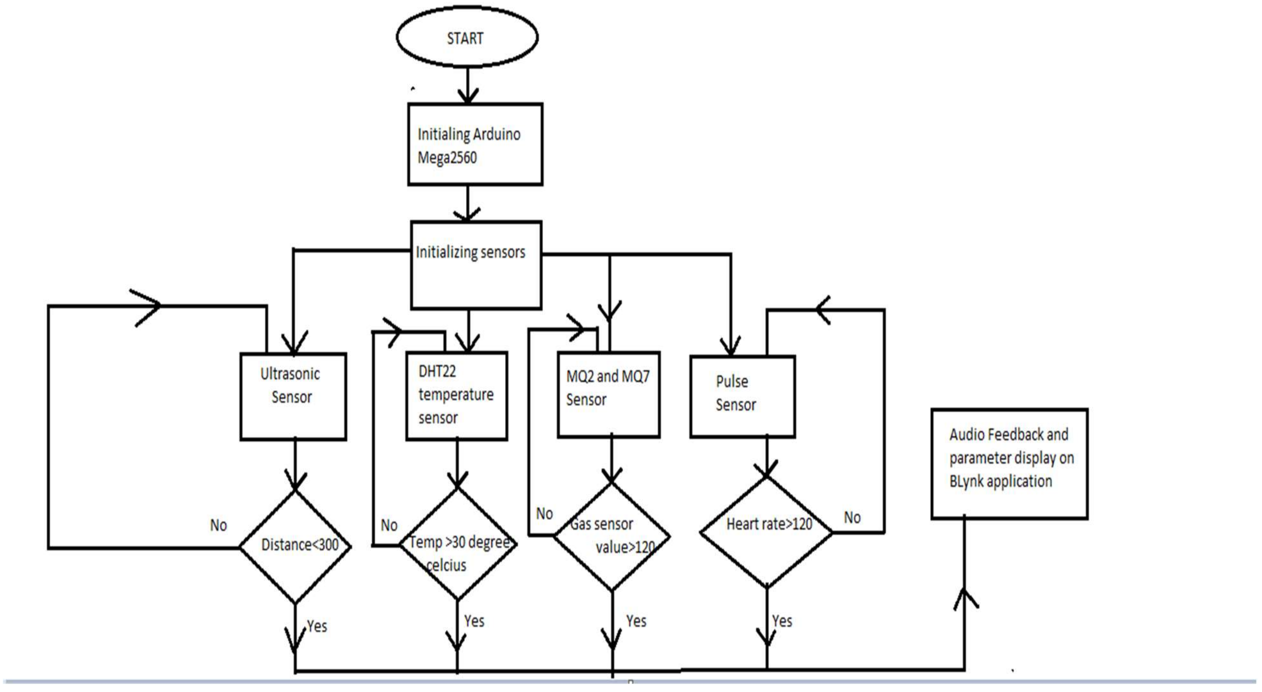


Fig 6.1

Chapter-7

PROTOTYPE

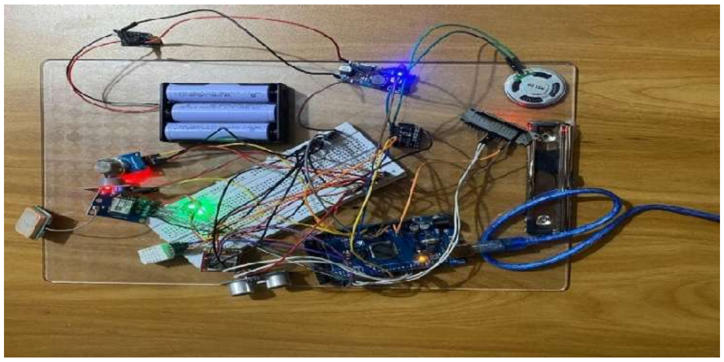


Fig 7.1

The dashboard displays real-time data to monitor the miner's health and environmental conditions. Here's a breakdown of the metrics shown:

1. **MQ2 and MQ7 Sensors**:
   - These are gas sensors.
   - **MQ2 (197 ppm)**: Indicates the level of combustible gases like methane in parts per million (ppm).
   - **MQ7 (385 ppm)**: Measures carbon monoxide (CO) levels in ppm.
2. **Humidity (71%)**:
   - Displays the relative humidity in the miner's environment.

3. **Heart Rate:**
   - Shows the miner's heart rate in beats per minute (bpm), which is critical for monitoring physical health and stress.

4. **Distance:**
    - Likely refers to proximity detection or clearance, which might help ensure the miner's safety in confined spaces.
5. **Temperature:**
    a. Monitors the ambient temperature of the miner's surroundings.

The interface also includes:

- Color-coded indicators: Green (safe), yellow/orange (moderate risk), and red (high risk or critical).
- Icons for notifications and possibly a communication feature (microphone icon).

This smart system helps improve safety in mining environments by alerting users to hazardous conditions or health risks

Chapter - 8

RESULTS

SENSORS	THRESHOLD	OBSERVED VALUE	ACTION TAKEN
MQ2/MQ**7** Sensor	120ppm	MQ2 - 301 ppm MQ7 - 400 ppm	Sensor value sent to ESP32. DFPlayer reads the warning message-" GAS WARNING DETECTED"
Temperature Sensor	30 ºC	26 ºC	Sensor continues to check until the threshold exceeds. No warning given.
Ultrasonic Sensor	300cm	11cm	Sensor value sent to ESP32. DFPlayer reads the warning message-" PROXIMITY ALERT. TAKE NECESSARY PRECAUTIONS"
Pulse Sensor	120bpm	180bpm	Sensor value sent to ESP32. DFPlayer reads the warning message-"HEART RATE ELEVATED"

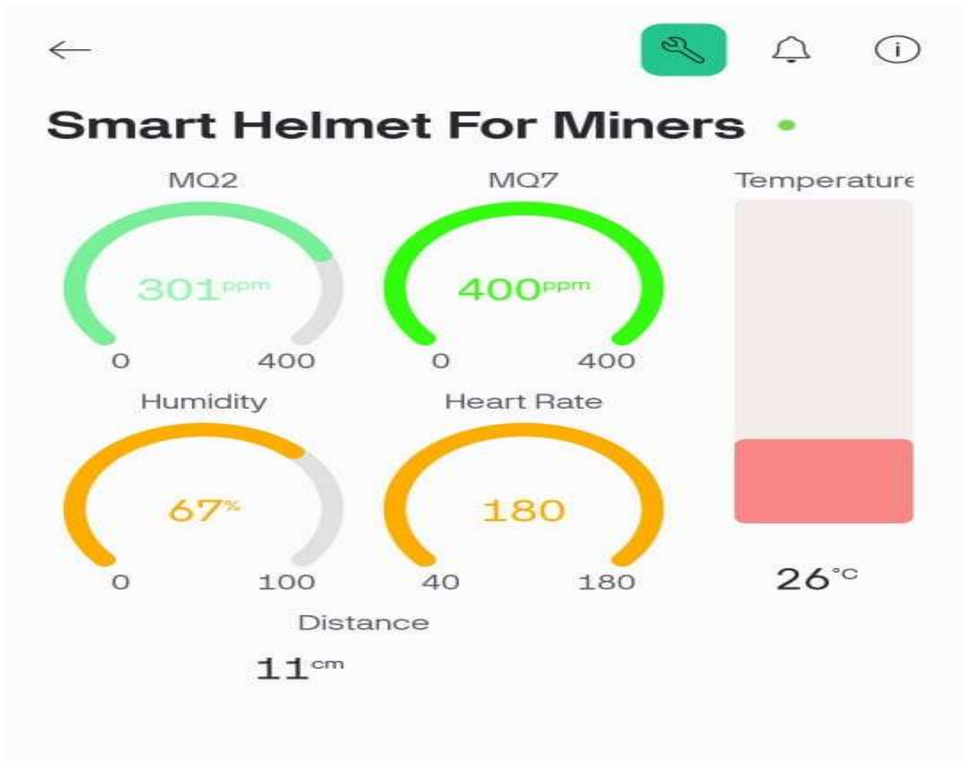
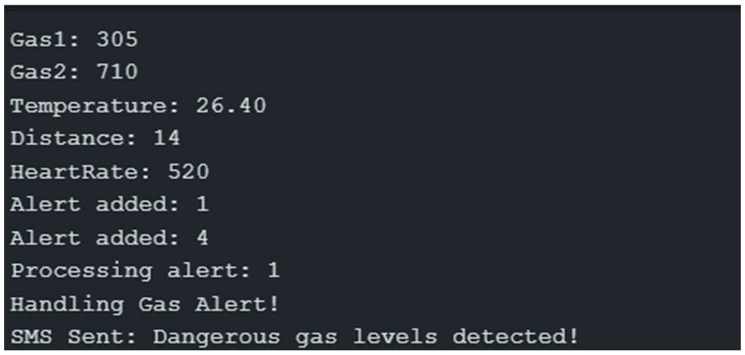
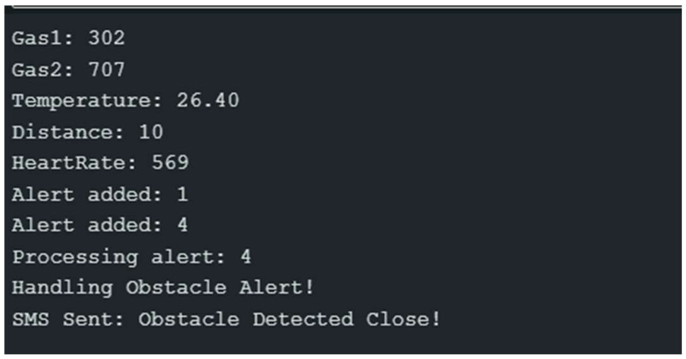


Fig 8.1



```
Gas1: 305
Gas2: 710
Temperature: 26.40
Distance: 14
HeartRate: 520
Alert added: 1
Alert added: 4
Processing alert: 1
Handling Gas Alert!
SMS Sent: Dangerous gas levels detected!
```

Fig 8.2 (a)



```
Gas1: 302
Gas2: 707
Temperature: 26.40
Distance: 10
HeartRate: 569
Alert added: 1
Alert added: 4
Processing alert: 4
Handling Obstacle Alert!
SMS Sent: Obstacle Detected Close!
```

Fig 8.2 (b)

Chapter - 9

CONCLUSION

The smart helmet for miners project is a significant step towards ensuring the safety and well-being of miners. By integrating advanced sensors and communication technology, the helmet provides real-time monitoring of critical parameters such as temperature, humidity, and the presence of dangerous gases like methane and carbon monoxide. The inclusion of an emergency switch allows miners to quickly alert their colleagues and supervisors in case of a hazardous situation.

The DF Player Mini enhances communication capabilities by enabling the transmission of important audio alerts, while the Blynk application facilitates seamless data visualization and remote monitoring. This combination ensures that critical information is always accessible, allowing for timely interventions and decision-making.

Moreover, the GPS tracking feature helps in pinpointing the exact location of miners, which is crucial during rescue operations or when tracking movement within the mine. The real-time data transmitted to the base station via the Blynk app ensures that supervisors are constantly updated on the miners' status and environmental conditions.

Benefits:

- Enhanced Safety: Continuous monitoring and real-time alerts significantly reduce the risk of accidents.
- Efficient Communication: The DF Player Mini and Blynk app enable effective communication between miners and the base station.
- Timely Interventions: Immediate detection of hazardous conditions allows for quick response and mitigation.
- Improved Working Conditions: A reliable safety system boosts miners' confidence and ensures a safer working environment.

Chapter - 10

FUTURE ENHANCEMENTS

1. **Energy Efficiency**:
   - Solar panels for charging, extending battery life in outdoor conditions.
   - Wireless charging for convenience.
2. **Enhanced Connectivity and Data Management**:
   - Integrate cloud platforms like Firebase or AWS to store real-time data for long-term analysis and trend prediction.
3. **AI and Predictive Analytics:**
   - Predict hazardous situations (e.g., gas accumulation trends or potential collapses) using historical and real-time data.
4. **Safety Enhancements:**
   - Use accelerometers and gyroscopes to detect if a miner has fallen and send an immediate alert.
   - Use GPS and ultrasonic data to provide audio directions for the safest escape route during emergencies.
5. **Enhanced User Interface**
   - Include vibration alerts for critical warnings to ensure miners are notified even in noisy environments.

REFERENCES

1. Sumit Kumar Jindal,Rohan Paul,Arpan Karar and Abhirup Datta, "Smart Helmet For Coal Miners", presented at the 1st International Conference on Paradigm Shifts in Communication, Embedded Systems, Machine Learning and Signal Processing (PCEMS), Nagpur, India, May,6-7,2022.

2. N Manikandan,Divakar Mishra and Arnob Banik,Smart Helmet And Monitoring For Miners With Enhanced Protection", presented at the 2nd International Conference on Edge Computing and Applications (ICECAA),Namakkal, India ,June,19-21,2023.

3. Hajira ZaibaRashmi,Rani Samantaray and Sufia Banu, Power Efficient Intelligent Helmet for Coal Mining Security and Alerting", presented at the International Conference on Sustainable Computing and Smart Systems (ICSCSS),Coimbatore, India,June,14-16,2023.

4. Sangharatna Bombarde,Preyas Videkar,Prerana Waghmare and Pankaj Kunekar,"IoT based Smart Security Helmet for Miner's Safety",presented at the 5th Biennial International Conference on Nascent Technologies in Engineering (ICNTE),Mumbai, India,Jan,20-21,2023.

5. Shawana Tabassum,Tanzila Noushin and Suwarna Karna,"IoT based Smart Helmet for Automated and Multi-parametric Monitoring of Underground Miners' Health Hazards",presented at the _IEEE 15th Dallas Circuit And System Conference (DCAS), Dallas, TX, USA,June,17-19,2022.