CHAPTER 1

INTRODUCTION

The accident is any uncertain activity due to unavoidable circumstances and carelessness of some people. This incident is happening continuously all around the world. Underground mining operations proves to be a risky venture as far as the safety and health of workers are concerned. These risks are due to different techniques used for extracting different minerals. The deeper the mine, the greater is the risk. These safety issues are of grave concern especially in case of coal industries. Thus, safety of workers should always be of major consideration in any form of mining, whether it is coal or any other minerals.

Underground coal mining involves a higher risk due to the problems of ventilation and potential for collapse. Coal has always been the primary resource of energy in India, which has significantly contributed to the rapid industrial development of the country. But the production brings with it the other byproducts, which proves to be a potential threat to the environment and the people associated with it. So the present work is a sincere attempt in analyzing and designing a real time monitoring system of detection for coal mine safety.

CHAPTER 2

COMPONENTS REQUIRED

2.1 NODEMCU ESP8266

NodeMCU is a low-cost open source IoT platform. It initially included firmware which runs on the ESP8266 Wi-Fi SoC, and hardware which was based on the ESP-12 module. Later, support for the ESP32 32-bit MCU was added.

2.2 ARDUINO GAS SENSOR

The MQ-9 Gas Sensor module is useful for gas leakage detecting in home and industry. It can detect methane content. The output is an analog signal and can be read with an analog input of the Arduino.

2.3 ARDUINO HUMIDITY & TEMPERATURE SENSOR

The DHT-11 is a digital-output, relative humidity, and temperature sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air, and sends a digital signal on the data pin. It is a temperature as well as humidity sensor is an easy way to measure temperature using an Arduino. The sensor can measure a fairly wide range of temperature (-50°C to 125°C), is fairly precise (0.1°C resolution), and is very low cost, making it a popular choice.

FIG 2.1 NODEMCU ESP8266

FIG 2.2 GAS SENSOR

FIG 2.3 HUMIDITY & TEMPERATURE SENSOR

CHAPTER 3

DESIGN, ESTIMATION AND PLANNING

3.1 OBJECTIVE

This project investigates the presence of toxic gases in critical regions and their effects on miners. A real time monitoring system using wireless sensor network which includes multiple sensors, is developed. A real-time monitoring systems may assist in monitoring and control over the mining environment.

Thus, the primary objective of this project is decided to design an efficient real-time monitoring system so that various leaked mine gases could be identified at times and preventive measures could be taken accordingly. The investigations to be carried out with the following objectives:

- Detection of methane
- Measurement and detection of humidity and temperature
- Design of a real-time monitoring system

3.2 WORKING

In this system, the coal mine safety systems are fixed with a gas sensor, temperature/humidity sensor & buzzer. All sensors are integrated into NodeMCU. A gas sensor detects the gas in the coal mine environment; if the gas level exceeds the threshold level then the buzzer gets high so that the mineworkers get notified. These sensor values are continuously uploaded to the cloud for analysis and also for further use. These 3 readings of graph are uploaded to thingspeak and then to ioT platform, then there will be 3 such graphs.

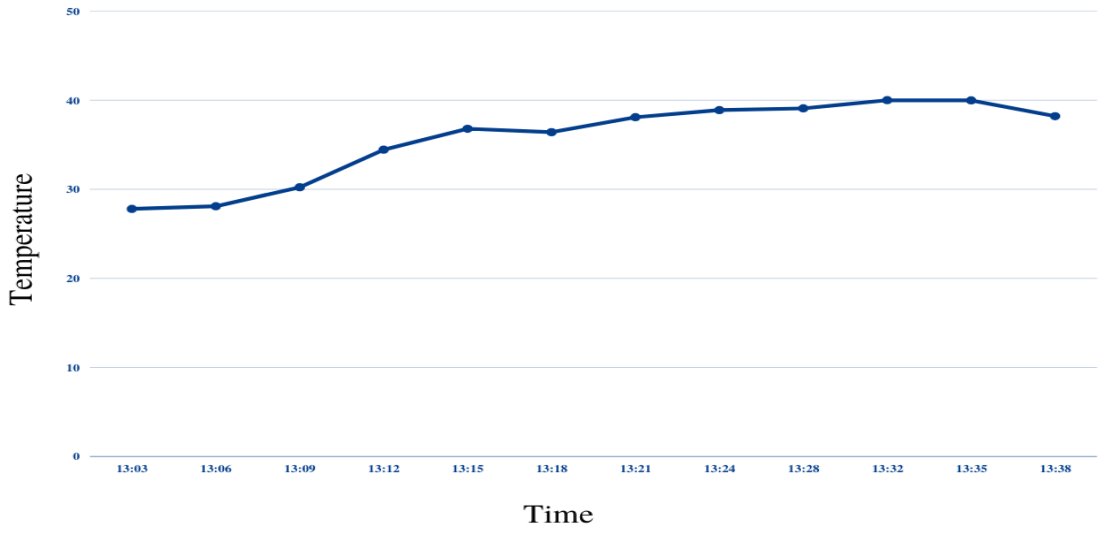


FIG 3.1 TIME VS TEMPERATURE

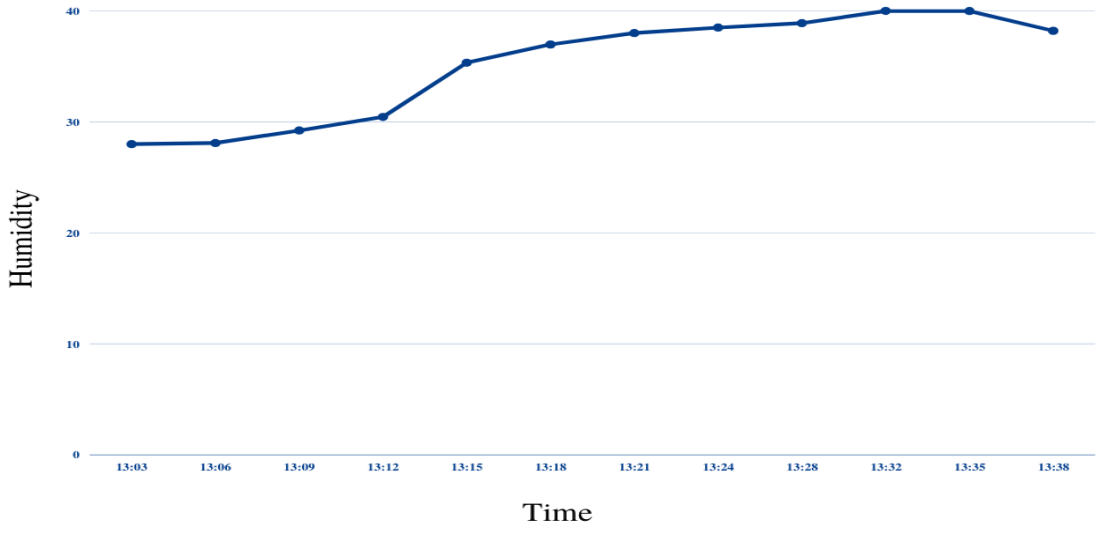


FIG 3.2 TIME VS HUMIDITY

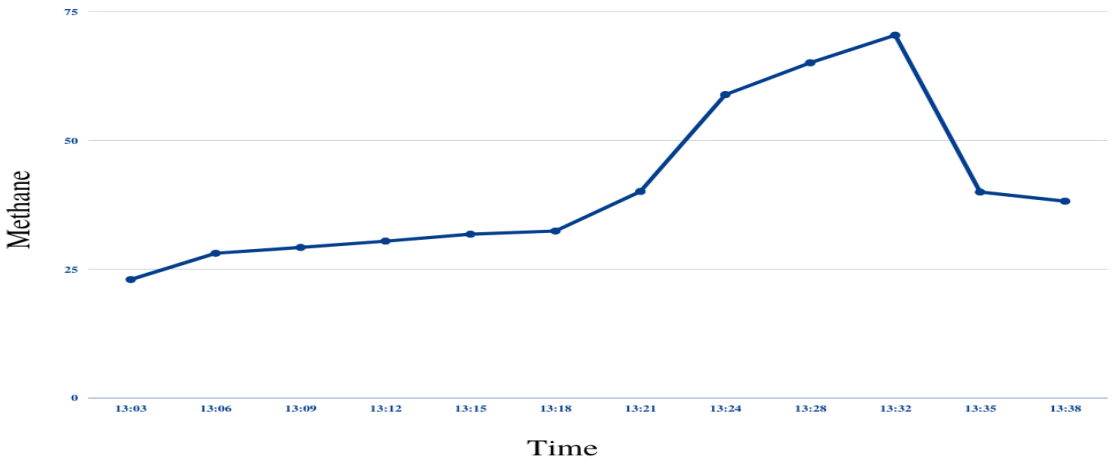
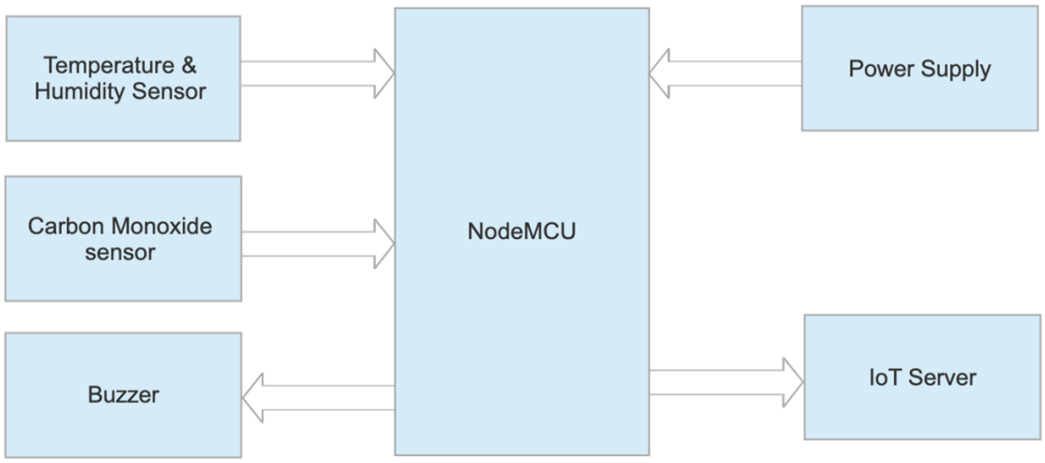


FIG 3.3 TIME VS METHANE CONTENT

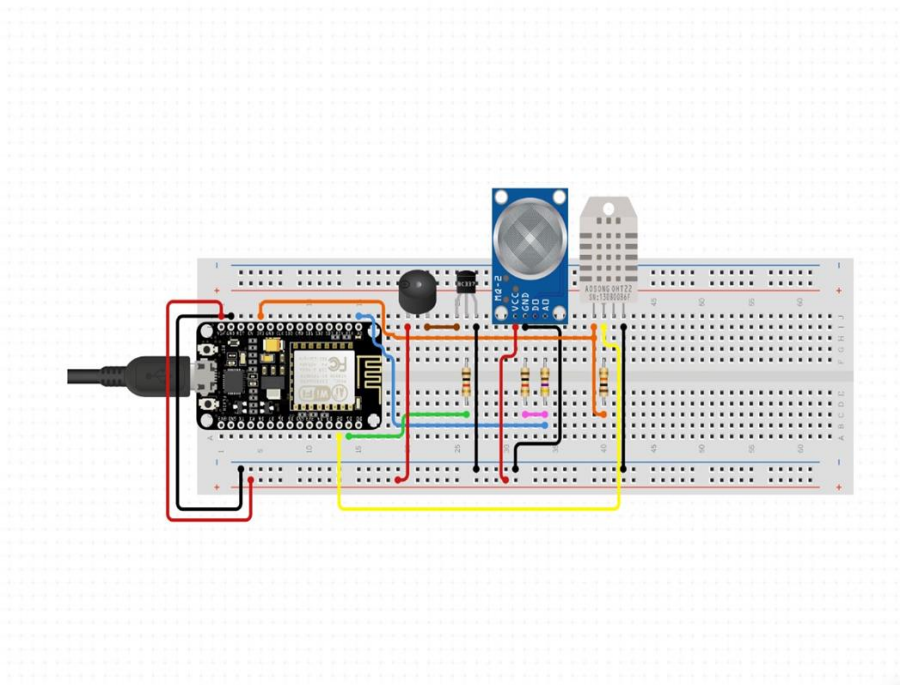
3.4 SOFTWARE REQUIREMENT

The Arduino Integrated Development Environment (IDE) is a cross-platform application that is written in functions from C and C++. It is used to write and upload programs to Arduino compatible boards, but also, with the help of third-party cores, other vendor development boards.

3.5 BLOCK DIAGRAM



3.5 CIRCUIT DIAGRAM



CHAPTER 4

ADVANTAGES AND DISADVANTAGES

4.1 ADVANTAGES

- There is no need for cables to lay and easy installation in blind areas, reducing the cost of the monitoring system. The number of nodes can be increased to eliminate blind areas. Also, it offers a general communication and allocation of the goal.
- The dense nodes ensure the data acquisition with high accuracy and optimum data transmission and further realization of a real-time monitoring system for the environment.
- A little computing ability, storage capacity with data fusion of sensor nodes make them suitable for the remote monitoring system.

4.2 LIMITATIONS

- In wired system the arrangement will be complex. In case of a fire the cable will be burnt.
- Complex analysis of sensor reading are not possible in this system
- A maximum of three sensors cloud be connected at time.
- The sensors must be placed in close proximity to main board

CHAPTER 5

APPLICATION

Here, this safety system can monitor and detect, air temperature, humidity, also methane gases present. It can also save miners life before any casualty occurs by giving alarm when sensors cross a certain value. It can also store the values for future use.

CHAPTER 6

CONCLUSION

The above safety system proposes the basic idea for the life saving measures for the miners. The sensors used for demonstration of concept are general. The MQ-9 gas sensor is more sensitive to methane but can sense some of the other toxic gases too .We found more heating of sensor if operated for long time. It is noise free and has low power platform. With use of sensors, the system can work with more accuracy in real time. It can be modified in industrial monitoring as well. A real time monitoring system is developed to provide clearer and more point to point perspective of the underground mine. This system is displaying the parameters on the monitoring unit; it will be helpful to all miners present inside the mine to save their life before any casualty occurs. Alarm triggers when the sensor values cross the threshold level. This system also stores all the data in the computer for future inspection.

REFERENCES

- https://en.wikipedia.org/wiki/Arduino_IDE
- https://wifinowglobal.com/news-and-blog/underground-wi-fi-makes-coal-mines-safer-and-more-productive-in-south-africa
- http://ijesi.org/papers/Vol%202(7)/Version-2/D0272014019.pdf
- https://www.academia.edu/33478285/Coal_Mine_Safety_Monitoring_and_Alerting_System

APPENDIX

```
#include <DHT.h>
#include <ESP8266WiFi.h>
String apiKey = "QGQ0J3P7QUP10INY9";
const char *ssid =  "Nokia 5.1 Plus";
const char *pass =  "12345678";
const char* server = "api.thingspeak.com";
#define DHTPIN 14
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHT11);
WiFiClient client;
void setup()
{
pinMode(12,OUTPUT);
Serial.begin(115200);
delay(10);
dht.begin();
Serial.println("Connecting to Nokia 5.1 Plus");
Serial.println(ssid);
WiFi.begin(ssid, pass);
while (WiFi.status() != WL_CONNECTED)
{
delay(450);
Serial.print(".");
}
Serial.println("");
Serial.println("Connected");
}

void loop()
```

```
{
float methane = analogRead(A0);
float humidity = dht.readHumidity();
float temperature = dht.readTemperature();
if (isnan(humidity) || isnan(temperature ||isnan(methane))) {
Serial.println("Failed to read from  sensor!");
return;
}
if (client.connect(server,80)) // api.thingspeak.com
{
String postStr = apiKey;
postStr +="&field1=";
postStr += String(temperature) ;
postStr +="&field2=";
postStr += String(humidity);
postStr += "&field3=";
postStr += String(methane/1023*100);
postStr += "\r\n\r\n\r\n";

client.print("POST /update HTTP/1.1\n");
client.print("Host: api.thingspeak.com\n");
client.print("Connection: close\n");
client.print("X-THINGSPEAKAPIKEY: "+apiKey+"\n");
client.print("Content-Type: application/x-www-form-urlencoded\n");
client.print("Content-Length: ");
client.print(postStr.length());
client.print("\n\n");
client.print(postStr);

Serial.print("Temperature: ");
```

```
Serial.print(temperature );
Serial.print(" °C, Humidity: ");
Serial.print(humidity);
Serial.print(" %, Methane Level: ");
Serial.println(methane/1023*100);
if(methane/1023*100 >= 60){
  digitalWrite(12,HIGH);
}
else {
  digitalWrite(12,LOW);
}
}
client.stop();
delay(1000);
}
```