Tactical rescue analysis system

Using IoT

Presented by Group 23 of 5th Semester.

Presentation Overview

POINTS FOR DISCUSSION

01 Problem statement, Abstract and Literature review

02 Module-wise breakdown

03 How did we Implement it?

04 The Demonstration

05 Result Analysis

06 The conclusion

07 Declaration and contributions

The Problem Statement

- When discussing disaster management, we are well apprised of the fact that the most habitual type of natural disasters are caused by the weather.
- These disasters include tornadoes, hurricanes, drought and extreme heat/extreme cold weather. Here the rescue operation comes into picture.
- Their first course of action would be the analysis of the surroundings and data gathering and analysis.

Abstract

- This Iot based device comprises of few sensors to measure environmental parameters which in turn monitor logging system to analyse these parameters.

- So, this scheme will be conceptualized to build a wireless IoT based system which can measure critical environmental parameters like Temperature, Humidity, Pressure, altitude and concentration of gas.

- Our rescue analysis system is ideally used to analyse the situation of our surrounding environment which will assist us to determine the risks of the rescue operation.

- Also, since our station is IoT enabled, we can send these parameters to a ThingSpeak channel (IoT cloud) where we can store, analyse and access the data remotely.

- Sending this information to ThingSpeak enables live monitoring from anywhere around the world and we can also view the logged data which will be stored on their website, plus, graph it over time to analyse it.

Literature review

Serial Number	Paper Name	Author Name	Aim of the Paper	Year
1	Disaster Relief and Data Gathering Rover	S. V. V. Srinivas Aditya Kumar Singh, Aman Raj Abhishek Shukla Rachit Patel Aviral Malay Department of ECE, ABESIT, Ghaziabad, India	The objective of this work is to provide the concerned authorities with a tool to help them gather data by reconnaissance using the rover and help them to form the strategy for the rescue operation that is cost effective, efficient, fast as well as secure for the rescue workers.	2018
2	Earthquake rescue robot.	Shubham Sarkar Akash Patil Aditya Hartalkar Aditya Wasekar Electrical Engineering, Fr.C. Rodrigues Institute of Technology, Vashi, Navi Mumbai, India	This paper intercepts the need of the hour to efficiently pose a path of saving lives and precious time during an earthquake. Earthquake is accompanied by huge hue and loss, along with accumulation of debris leading to make search operation difficult. This paper suggests the use of rescue robot car for such rescue operation, to determine the people stuck in a particular region.	2017
3	Design and Development of a low cost Rescue robot.	Ahalya Ravendran Poom Ponpai Parinthorn Yodvanich Wimonsiri Faichokchai Chung-Hao Hsu School of Manufacturing Systems and Mechanical Engineering, Thammasat University, Pathum Thani, Thailand	This research study focuses on design and development of low cost mobile robot with various task operations that are adaptable for unknown terrain and environmental conditions.	2019
4	Multi-Robot-Multi-Target Path Planning and Position Estimation for Disaster area	Shin nyeong Heo Sheng yu Lu Ji-sun Shin Hee-hyol Lee Graduate school of Information, Production and Systems, Waseda University, Japan	This paper proposes a multi-robot-multi-target path planning for disaster area and position estimation.	2018

Module-wise breakdown

➢ **Module 1:**Environment assessment

- As the very first step suggests, identification of the risk, our analysis system will first begin to analyze the external conditions like temperature, pressure, and humidity.
- So, after analyzing these basic environmental conditions, our analysis system further decides on how to get a better approach on the given circumstances of the situation.

➢ **Module 2:**Collecting data

- As our analysis system follows up on the given conditions, we will further determine the case much more effectively by collecting the data and keeping a record of the on-going situation which will in turn be stored in our thingspeak channel.

➢ **Module 3:**Analyzing the collected data

This is our crucial step of the project where we observe and analyse the recorded data on thingspeak channel, which will help us determine the risk factors involved in our situation.

- **Module 4:**Determining the risk factor

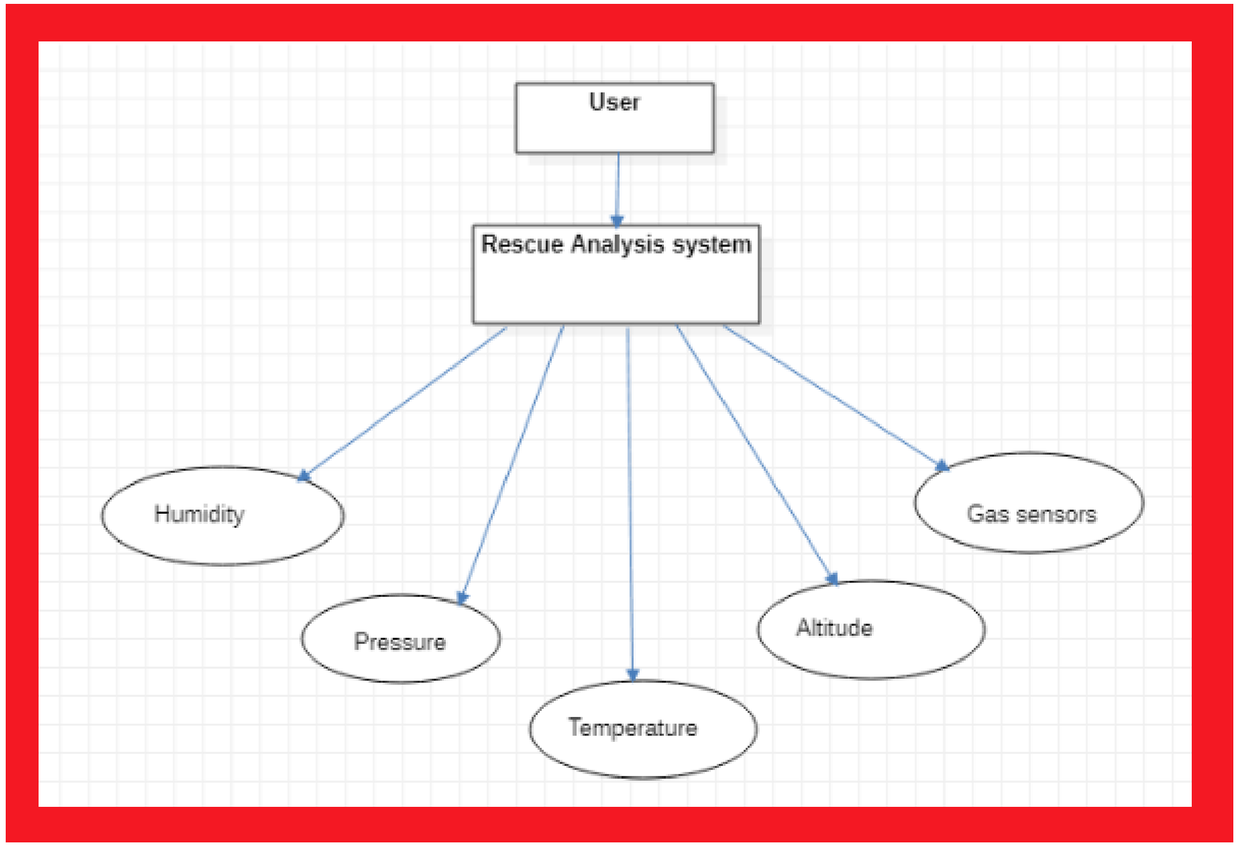
  - So, once the data is analyzed carefully, our rescue team decides the extremities of the risk, with the help of other functional requirements used in our project like a camera which will be mounted on our analysis system and help us in visually monitoring the situation.
  - We also have a MQ2 gas sensor, which will determine the presence of inflammable gas leaks.
  - Our switches will also play a crucial part in assisting the rescue teams aid the person in danger by simply sending out a message through the switches, which will send the signal to the other crew members of the rescue teams.

- **Module 5:**Execution and personal aid

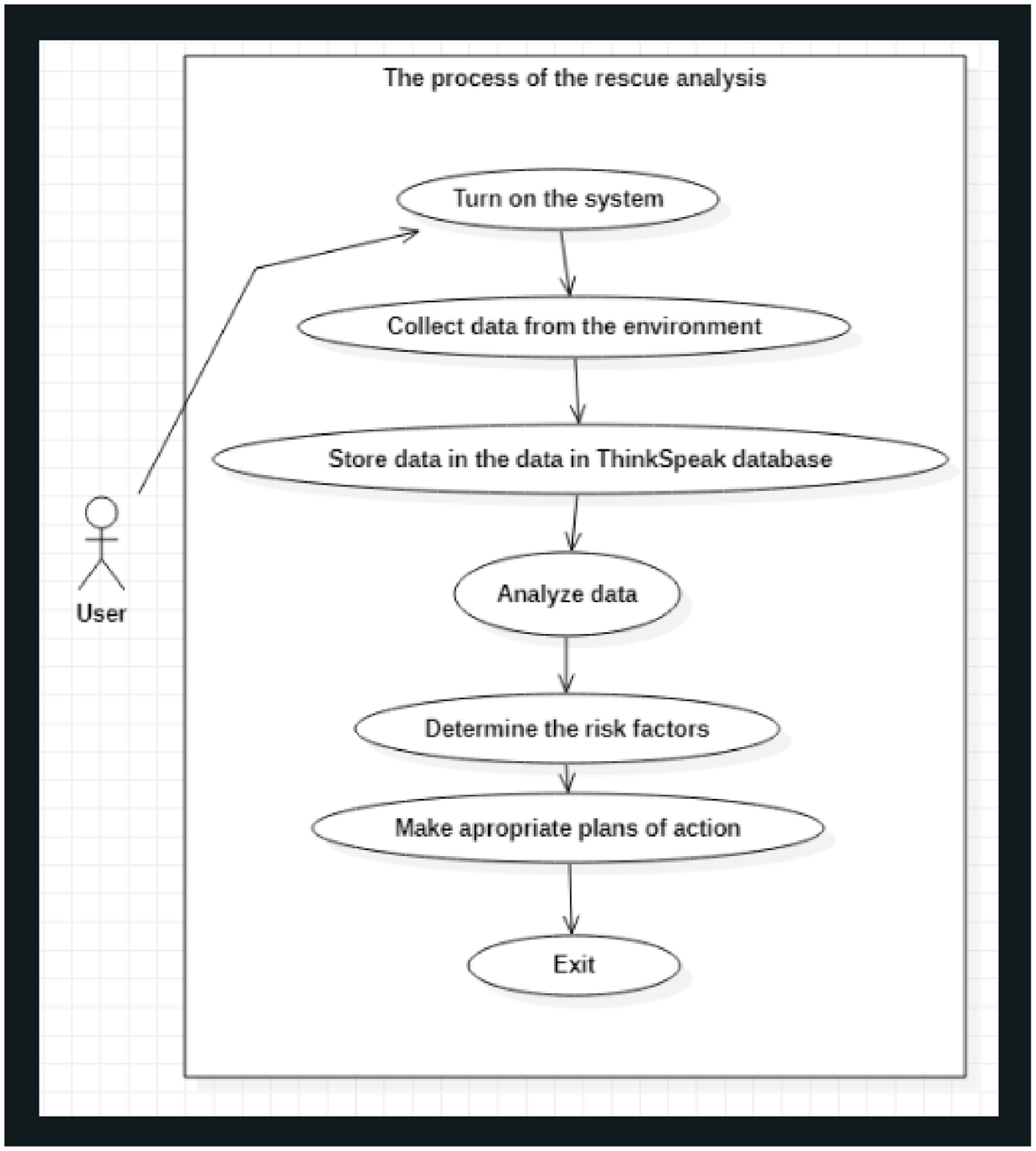
  At the final stage, the rescue team chooses an appropriate plan of action considering the risk factors involved in the rescue operation.

Design diagrams

ER Diagram



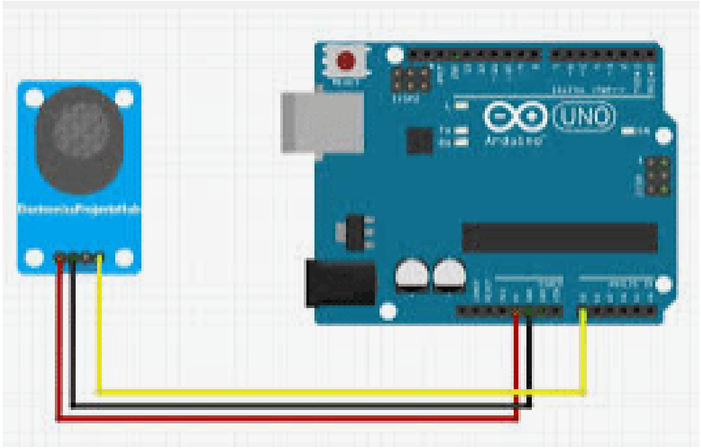
Use case diagram

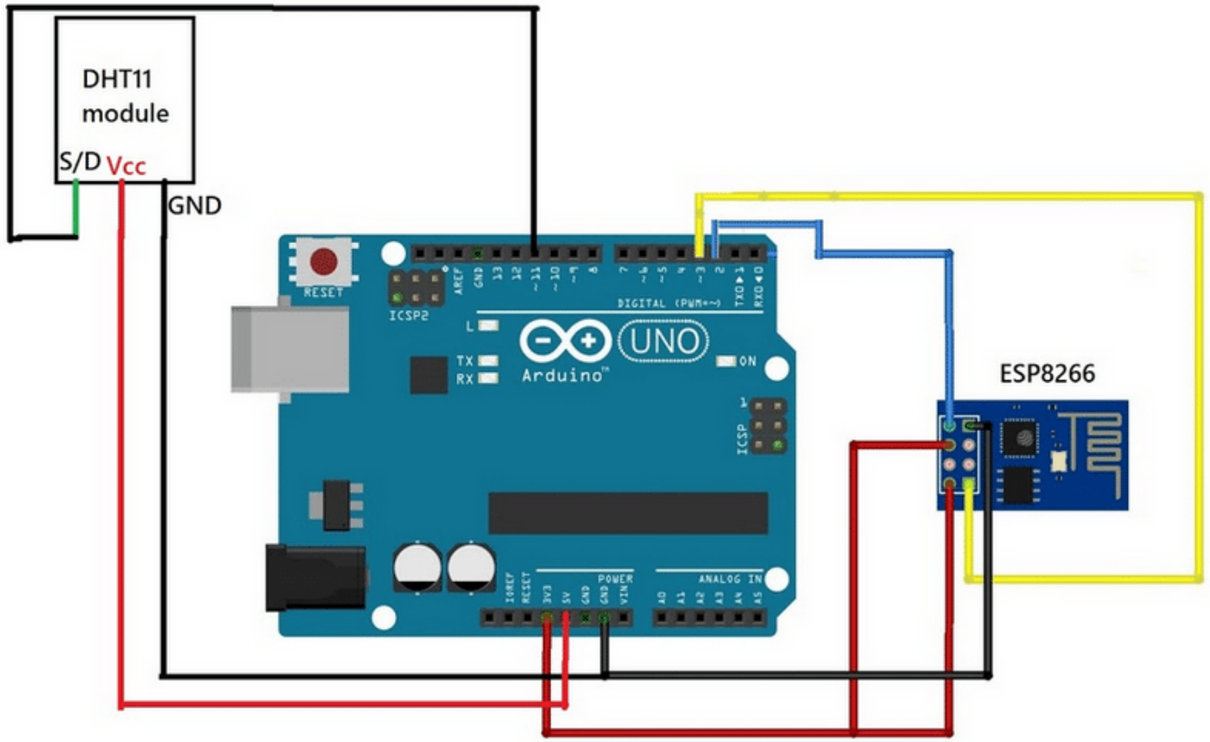


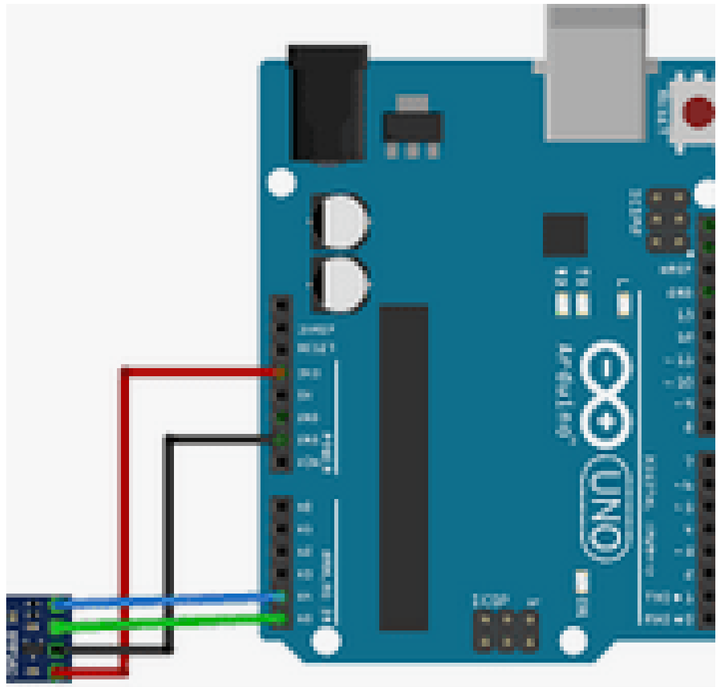
How did we implement it?

LET'S TAKE A LOOK AT OUR CODE

Circuit diagram







IMPLEMENTATION

PART 1: Inclusion of all libraries and declaring the global variables

PART 2: Setup and loops functions

PART 3: Functions for individual sensors

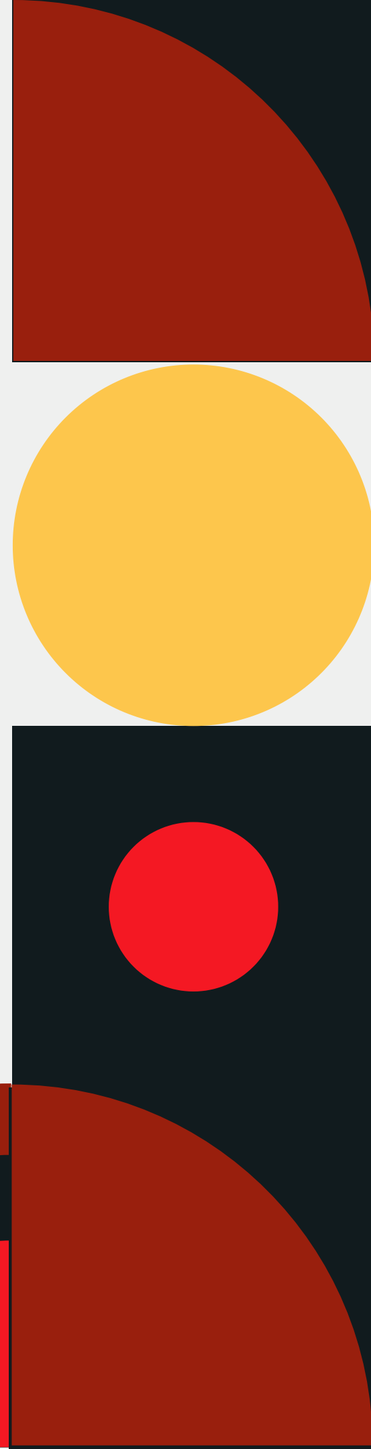
```
#include <SoftwareSerial.h>
#include <dht11.h>
#include <Wire.h>
#include <Adafruit_BMP085.h>

float pressure = 0.0;
float altitudee = 0.0;
Adafruit_BMP085 bmp;
#define RX 2
#define TX 3
#define dht_apin 11
dht11 dhtObject;
String AP = "HiraChak";
String PASS = "SnehLata79";
String API = "4V8TGO5TT855YACW";
String HOST = "api.thingspeak.com";
String PORT = "80";
int countTrueCommand;
int countTimeCommand;
boolean found = false;
int valSensor = 1;
int buzzer = 10;
int smokeA0 = A1;
int sensorThres = 400;

SoftwareSerial esp8266(RX,TX);
String pressureread();
String altituderead();
```

```
void setup()
{
        Serial.begin(9600);
        esp8266.begin(115200);
        bmp.begin(9600);
        sendCommand("AT",5,"OK");
        sendCommand("AT+CWMODE=1",5,"OK");
        sendCommand("AT+CWJAP=\""+ AP +"\",\""+ PASS +"\"",20,"OK");
}


void loop()
{
      String getData = "GET /update?api_key="+ API
      +"&field1="+getTemperatureValue()+"field2="+getHumidityValue()+"field3="+getGasdata()+"field4="+pressureread()+"&field5="+altituderead();

      sendCommand("AT+CIPMUX=1",5,"OK");
       sendCommand("AT+CIPSTART=0,\"TCP\",\""+ HOST +"\","+ PORT,15,"OK");
       sendCommand("AT+CIPSEND=0," +String(getData.length()+4),4,">");
       esp8266.println(getData);delay(1500);countTrueCommand++;
       sendCommand("AT+CIPCLOSE=0",5,"OK");
}
```

```
String getTemperatureValue()
{
  dhtObject.read(dht_apin);
  Serial.print(" Temperature(C)= ");
  int temp = dhtObject.temperature;
  Serial.println(temp);
  delay(50);
  return String(temp);
}

String getHumidityValue()
{
  dhtObject.read(dht_apin);
  Serial.print(" Humidity in %= ");
  int humidity = dhtObject.humidity;
  Serial.println(humidity);
  delay(50);
  return String(humidity);
}
```

```
String pressureread()
{
  pressure = bmp.readPressure() / 100.0;
  Serial.print("Pressure = ");
  Serial.print(pressure / 100.0);
  Serial.println(" hPa");
  return String(pressure);
}

String altituderead()
{
  // Calculate altitude assuming 'standard' barometric
  // pressure of 1013.25 millibar = 101325 Pascal
  altitudee = bmp.readAltitude();
  Serial.print("Altitude = ");
  Serial.print(altitudee);
  Serial.println(" meters");
  return String(altitudee);
}
```

```
String getGasdata()
{
        pinMode(buzzer, OUTPUT);
        pinMode(smokeA0, INPUT);
        Serial.begin(9600);
        int analogSensor = analogRead(smokeA0);

        Serial.print("Gas sensor: ");
        Serial.println(analogSensor);
  // Checks if it has reached the threshold value
        if (analogSensor > sensorThres)
        {
                tone(buzzer, 1000, 200);
                return String(analogSensor);
        }
        else
        {
                noTone(buzzer);
                return String(analogSensor);
        }
        Serial.println(analogSensor);
        delay(100);
}
```

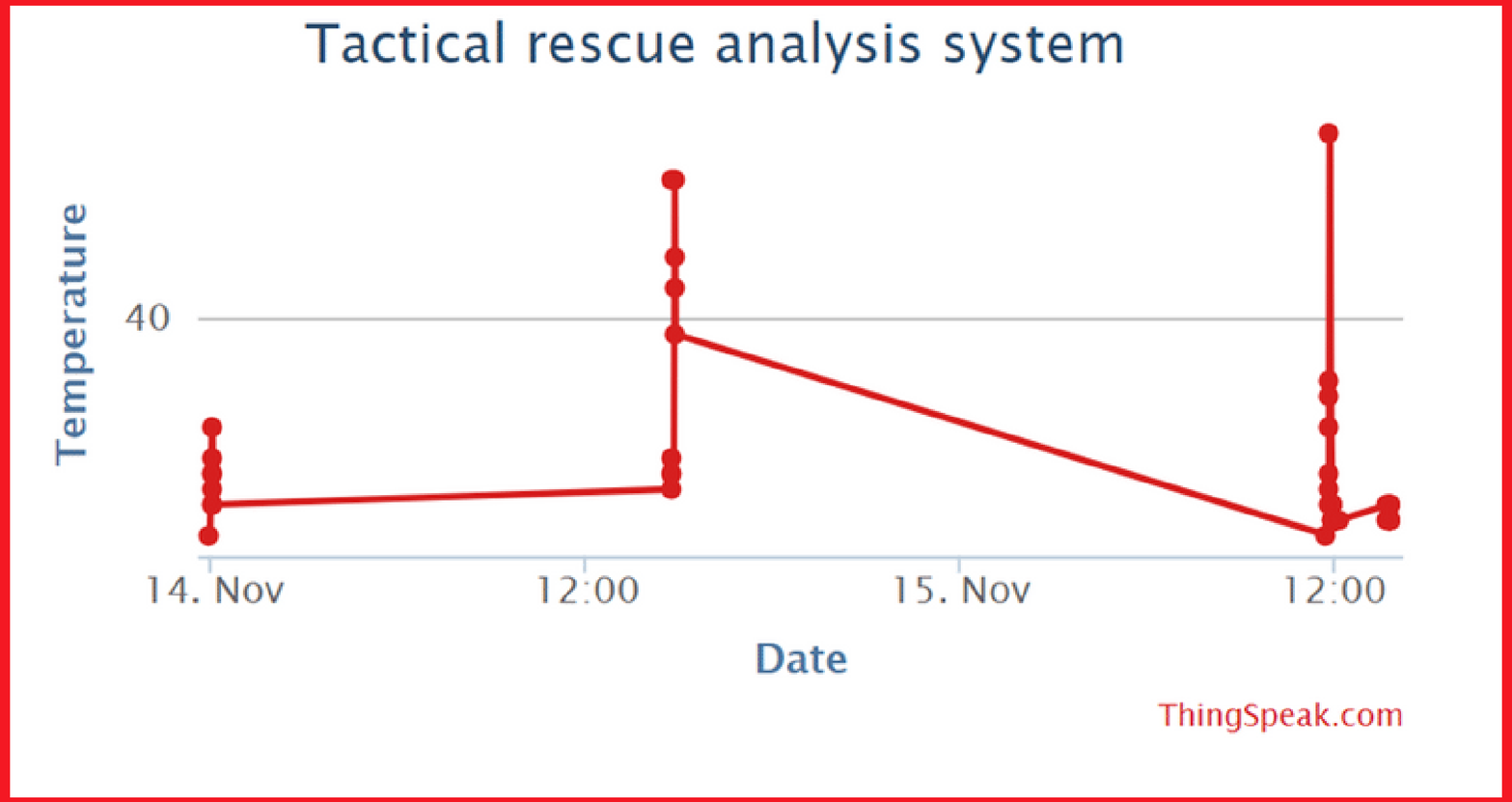


The demonstration

LET'S MAKE IT WORK

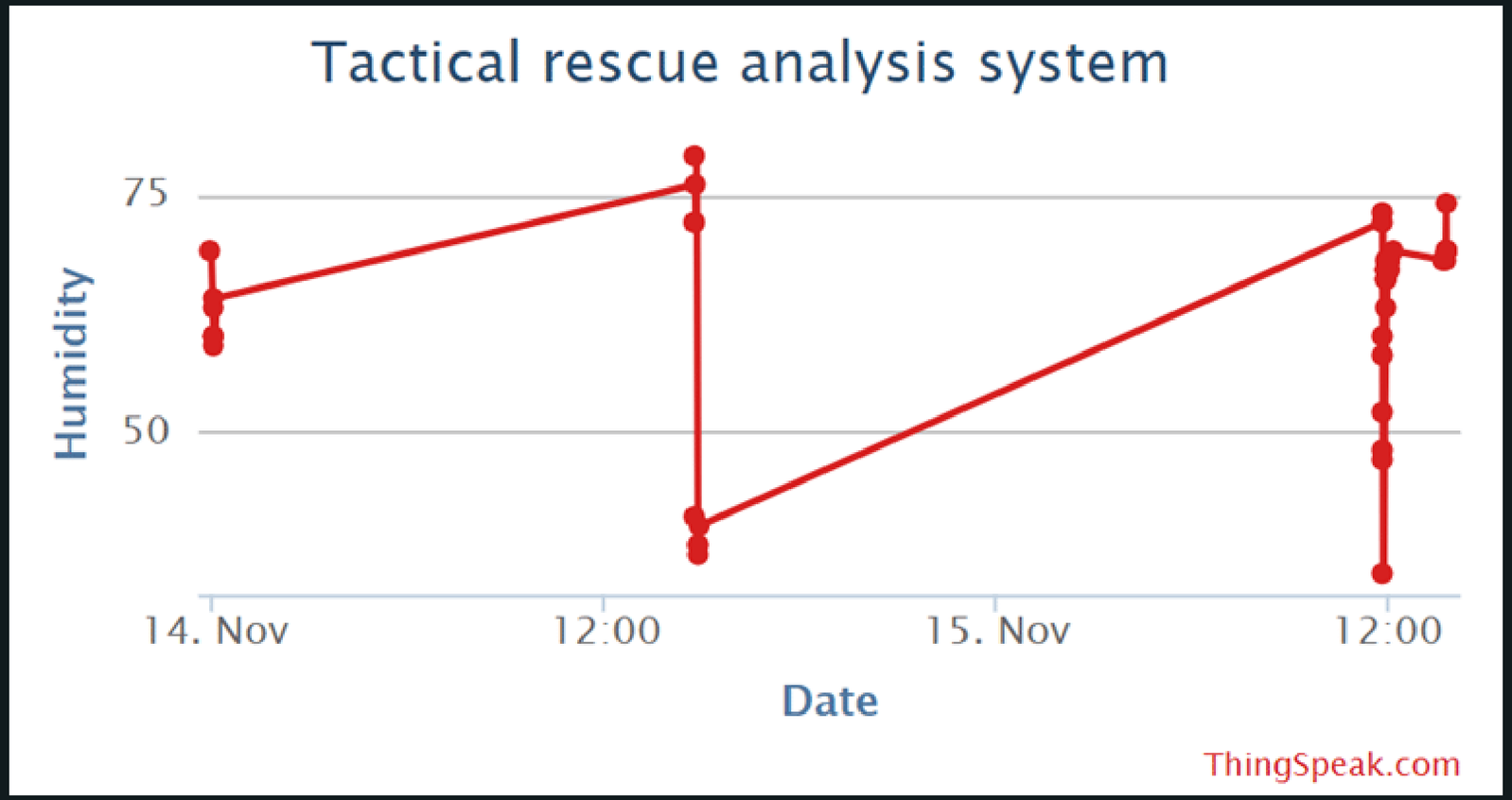
Result analysis

Since our project is result analysis oriented, our main focus will be towards the analysis of the collected data. To analyze our data we visualize our collected information in the forms of graphs. Here we shall explain each graph and their attributes.



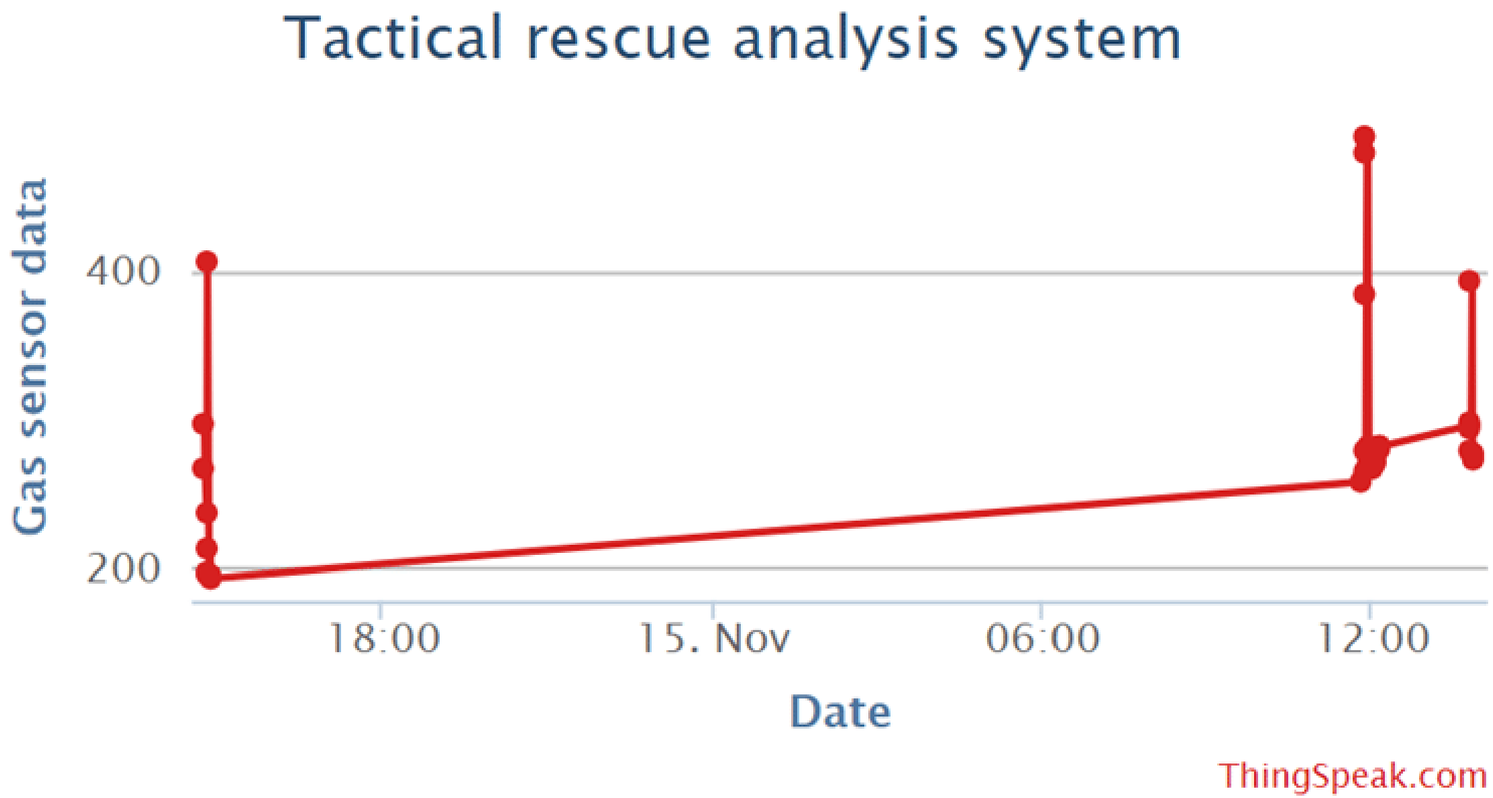
Humidity Graph

The humidity is temperature dependant. Hence when the temperature rised there is a fall in humidity. We can see such variations in the graph. Thats why we decided to calculate the temperature first then the humidity.



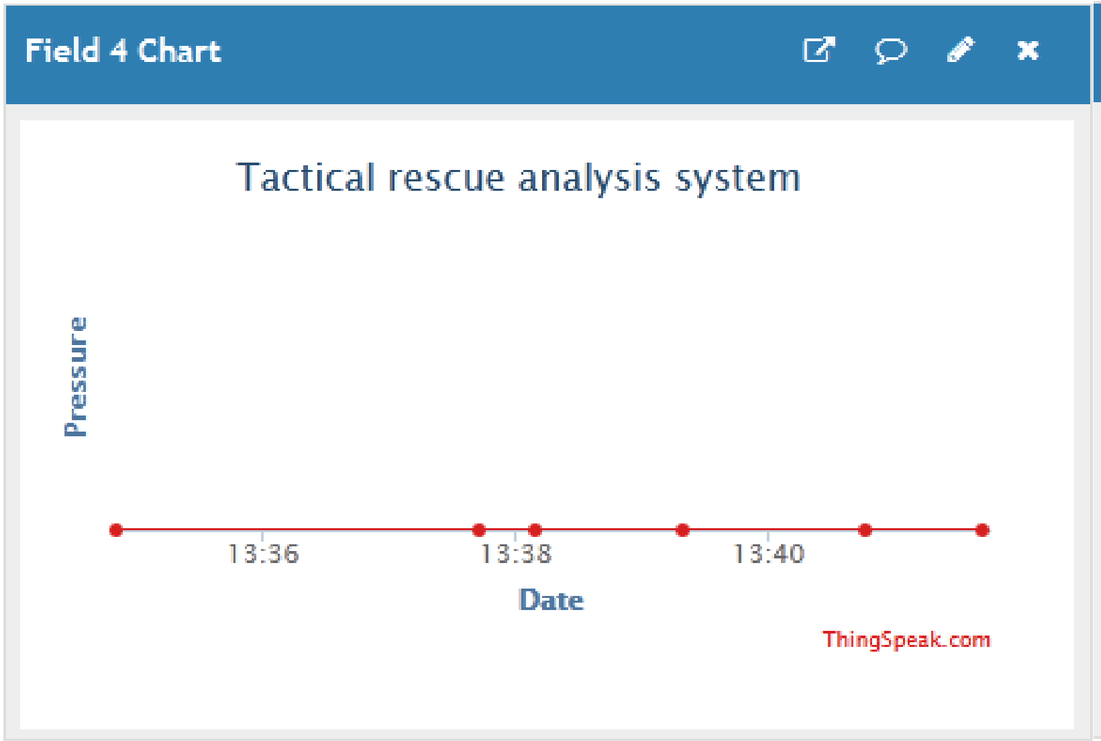
Concentration of gas graph

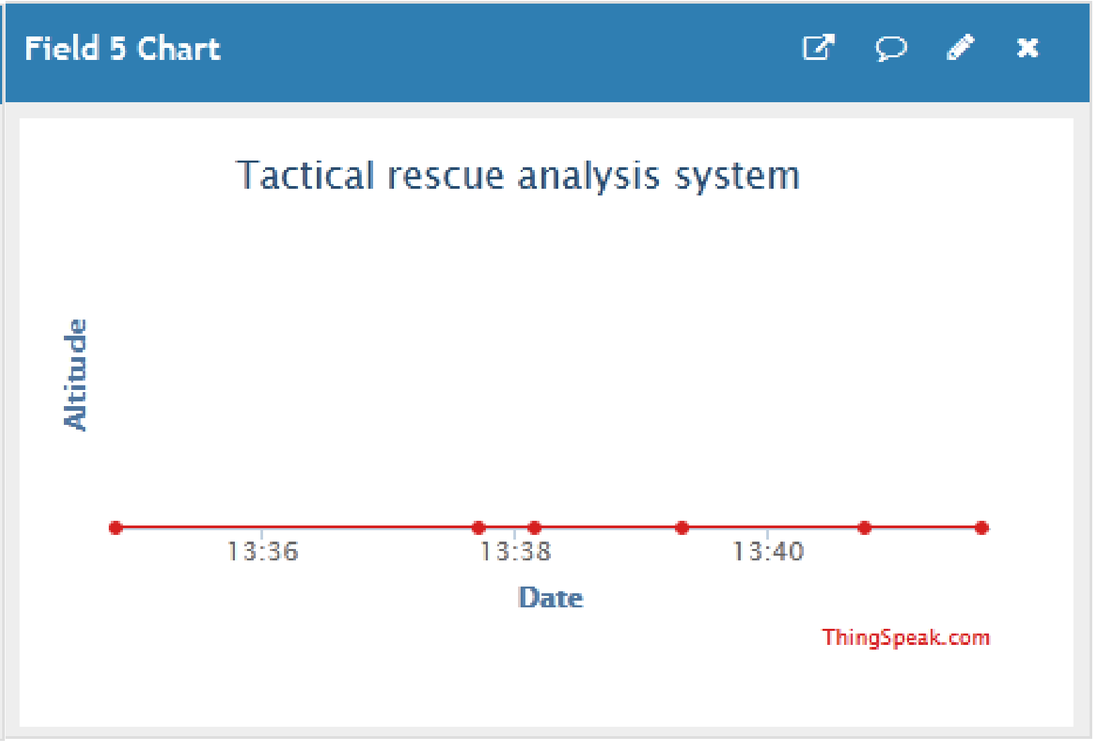
The gas sensors gathers information about the concentration of the gasses that are present. It calculates then in part per million units. The resultant data is then graphed out to see any abnormalities in the atmosphere.



The Pressure and altitude graph

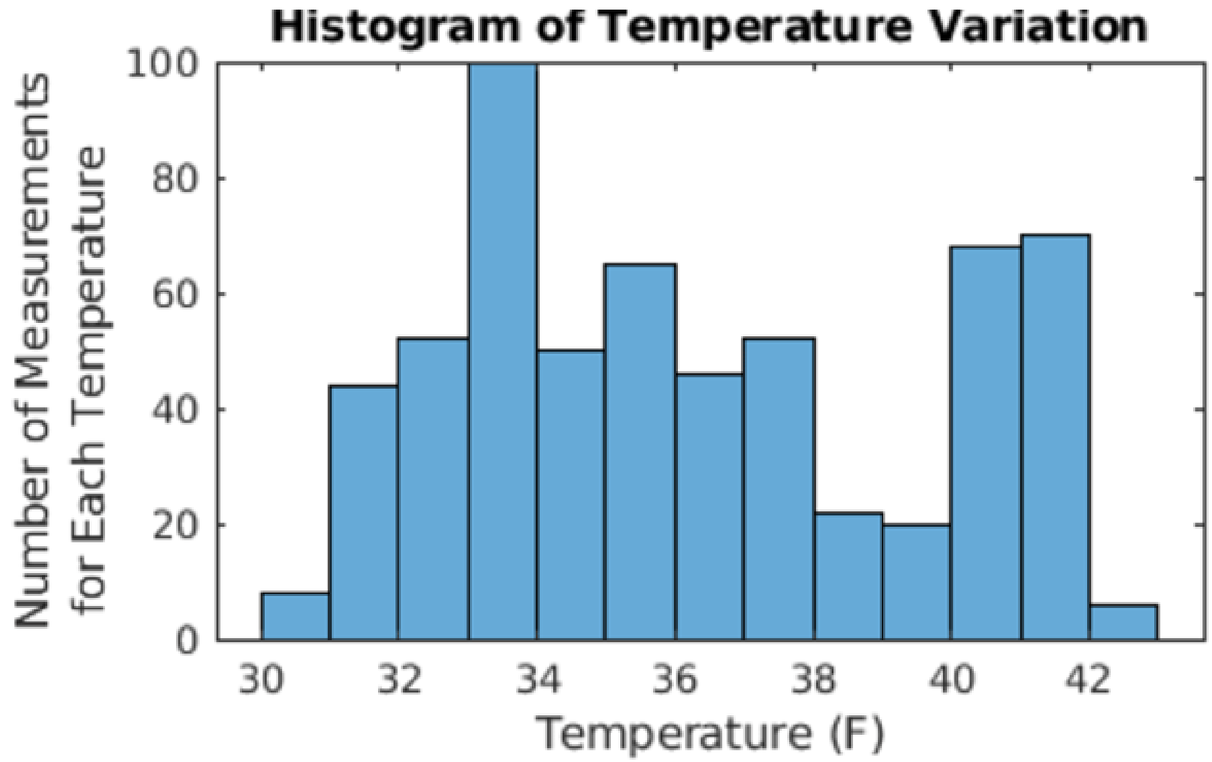
Here the altitude and the pressure is being measured my the BMP180 sensors. The features are inbuilt ance diminishing the need for getting seperate sensors for each of them.





HISTOGRAM OF TEMPERATURE VARIATION

Since a 2 x 2 graph about temperature can be a little hard to read or analyze, we can create a histogram graph that would be far more visually convenient to grasp the data fluctuations.



Conclusion

- The objective of our project was to detect extremities in the temperature and humidity levels, it also contributed to monitor the presence of harmful gas leaks or explosive gases.

- Through which our rescue teams could identify and analyze data from our ThingSpeak channel. This would immensely benefit the analysis phase of the rescuers.

- We demonstrate the collected data in the form of graphs which is visually accessible since it forms a live graph. It helps us in learn about the environmental conditions and predict unforeseen events.

Declaration

- This project was inspired by the recent abnormalities and severe climate change we witnessed.
- We referred many papers on disaster management, different Iot sensors and how we can implement it as well as many real life technologies such and rescue robots and drones.
- This is a declaration slide where we state how much we have taken of the internet and how much was done on our own.

20%

The commands for the **send** function in Arduino were reffered on the internet

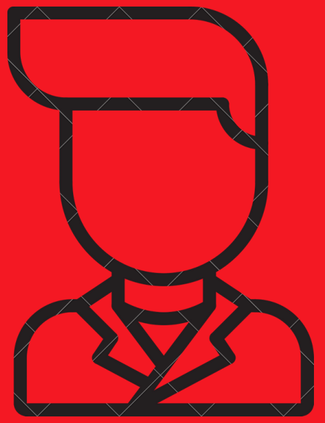
80%

The coding for indivisual sensors and the thinkspeak channel was created and customised by us.

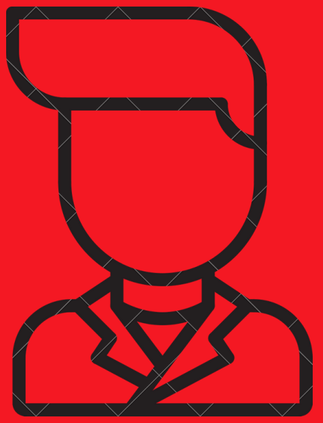
Contribution

Sl.No	Name	Code(%)	Research(%)	PPT(%)	Report(%)	Total
1	V A S Kiranmayee	30%	20%	10%	20%	80
2	Srishti Ranjan	30%	15%	10%	20%	80
3	Shreyansh JV	20%	10%	20%	30%	80
4	Shylesh Suresh	15%	20%	25%	20%	80

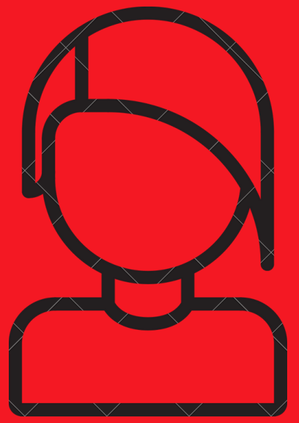
Presenters



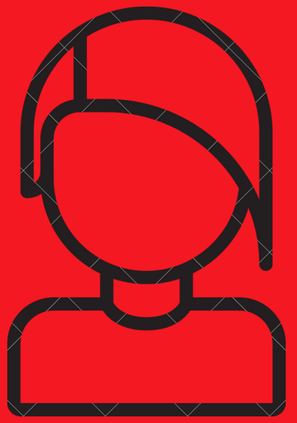
Shreyansh JV
ENG18CS0268



Shylesh S
ENG18CS0274



Srishti Ranjan
ENG18CS0284



V.A.S Kiranmayee
ENG18CS0304