Sentinel – A Smart Fire Fighting Helmet

PROJECT REPORT

SUBMITTED FOR THE COURSE:

IOT DOMAIN ANALYST

(ECE3502)

PRIYANSHU TRIVEDI	18BEE0203
SUMIT JAISWAL	18BEE0107
SOMDYUTI DAS ADHIKARY	18BEE0112
MANAS DIXIT	18BEE0117

GUIDIED BY:

PROF. Geetha M

Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

March 2021

TABLE OF CONTENTS

OBJECTIVE

Fire-fighting has been a problem since the medieval ages. Even with the advent of modern technology , we haven 't been able to reduce fire accidents significantly . Our proposed project aims to use all relevant data, to remedy any fire accident. The project uses a variety of sensors and the collected data is processed and utilized to alert authorities and display meaningful results. All our sensors are integrated with one central Micro-controller. The Micro-controller is responsible for relaying the data to a shared database, from where the processed data is retrieved. Location data will be broadcast from the server to all clients in the nearby area.

The main idea of this project is to monitor the physical parameters around the fire-fighter in hazardous situations, the whole project is supposed to be mounted on the helmet of the individual. The parameters needed to be monitored are temperature, humidity, air condition i.e. levels of toxic and inflammable gases in the atmosphere nearby including the atmospheric pressure of the area.

There will be two interfaces of the project a mobile application and a web interface. The mobile application which will report any abrupt change and will be used to send SOS message to call for help and the web interface will be monitored by someone from outside so just in case if anything goes wrong or in case of abrupt parameter change, one can send help from outside.

INTRODUCTION

FIRE FIGHTING HELMET WITH INDICATORS AND DATA SHARING

Firefighting is one of the toughest job in today 's world ,each and every day hundred and thousands of firefighters put their life on stake for helping both nature and humanity According to NFPA statistics over 100 casualties were recorded in 2019.

Sentinel is a novel helmet designed to advance current firefighting methods . By attaching various crucial sensors on the helmet , we will be able to gather important vitals about the environment such as temperature , CO_2 levels , atmospheric pressure , volume of total organic compounds , and much more while simultaneously sending this data in real-time over the Internet of Things platform to a remote database . We will then retrieve this information and create a unique app user interface that responds to the real-time changes in data to relay information across all individuals over watching the situation.

The Sentinel firefighter helmet sends and retrieves information by using three sensors connected to a central micro-controller capable of connecting to nearby WIFI networks . When information from the environment is extracted , we are capable of sending this information using WIFI following the IOT protocol to a remote real-time database where it is stored and processed . When this information is retrieved , it is used to create an app user interface that displays the information in a real-time framework to show changes as they happen . This interface allows all subscribers to the web app to monitor the situation different firefighters are and providing assistance when needed.

The micro-controller we used for this project is the NODE-MCU board which is an Internet of Things enabled board built on top of the Arduino framework . We then attached several sensors to the board (DHT 11 Temperature and Humidity Sensor , BMP 280 Atmospheric and Pressure Sensor, and MQ3 –Flammable gas sensor) following the I2C (Inter -Integrated Circuit) protocol where we used a multi -master /multi -slave architecture to allow sensors to publish and receive data to the bus system . When the data is received by the NodeMCU micro -controller , the data is sent over WiFi to a real - time Google Firebase database where it is categorized by sensor and type of data. This information is then retrieved and displayed on an app user interface updated in real- time. The app could be built using kotlin .

COMPONENTS USED

The project was built on Raspberry Pi initially, but due to economy concern we've decided to switch the project platform to NodeMCU microcontroller.

SENSORS USED :

- ❖ DHT11 – Temperature and Humidity Sensor
- ❖ MQ3 – Flammable gas Sensor
- ❖ BMP280 – Air Pressure Sensor
- ❖ NODE-MCU

SOFTWARE COMPONENTS

- ❖ THINGSPEAK DATA COLLECTION PLATFORM

METHODOLOGY

- Researchers have developed the new software framework enable to integrate a wide range of devices and services and efficiently manage resources.

- In addition, based on this, they developed a smart helmet to respond the disaster safety accidents. The smart helmet collects, generates, and converts information on sensors (infrared camera, electro Optical camera, drone camera, oxygen residual sensor, 6- axis inertial sensor, and smart watch) and it can be monitored through head mounted display(HMD) and the Command Center.

- Finally, they developed a simulator and generated data based on scenarios, and also tested all devices and service outdoors.

SOFTWARE COMPONENT

THINGSPEAK DATA COLLECTION PLATFORM

- ThingSpeak is an IoT analytics platform service that allows you to aggregate, visualize and analyze live data streams in the cloud. ThingSpeak provid instant visualizations of data posted by your devices to ThingSpeak.

- With the ability to execute MATLAB code in ThingSpeak you can perform online analysis and processing of the data as it comes in. ThingSpeak is often used for prototyping and proof of concept IoT systems that require analytics.

Fig5.ThingSpeakSoftware

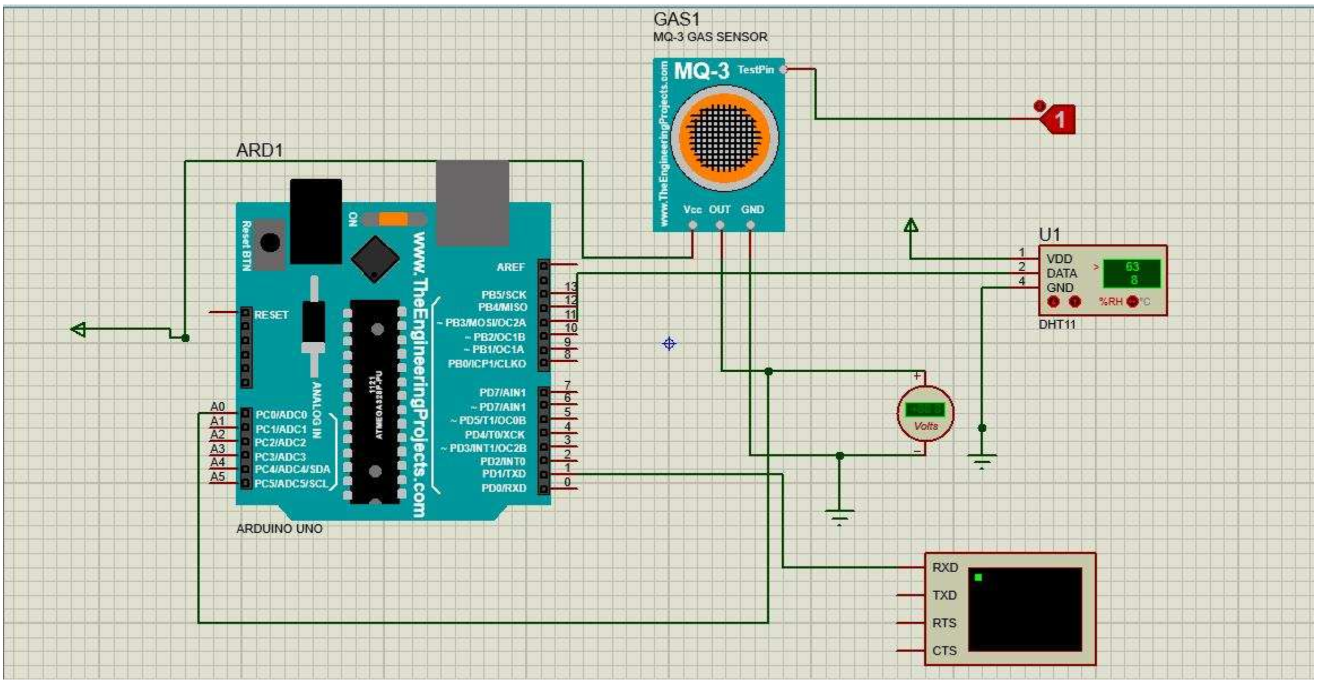
- The MVVM architecture which is used widely by many companies, was found to be the best fit for this application . Research was also done for finding the best dependency injection framework for the application .The benchmarks for the Kotlin and Dagger framework seemed to be promising .We proceeded with the Kotlin framework as it provides a simpler DSL (Domain Specific language) than the Dagger framework.

- The application uses the ThingSpeakKotlin API to retrieve the data from the NodeMCU microcontroller.

PROTEUS PROGRAM

1. BLOCK DIAGRAM

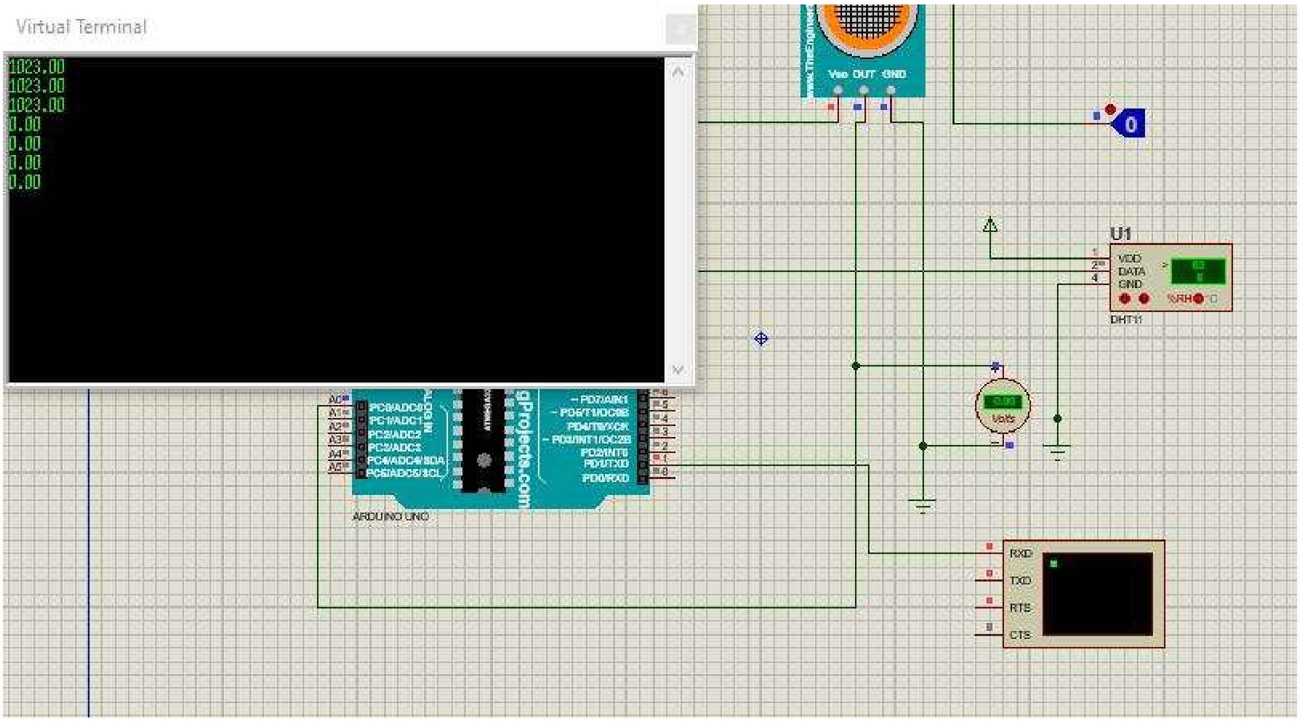
Due to non availability of support for simulation of Node MCU in simulation software we are unable to demonstrate live functionality , instead we have used arduino for schematic and simulation of input output.

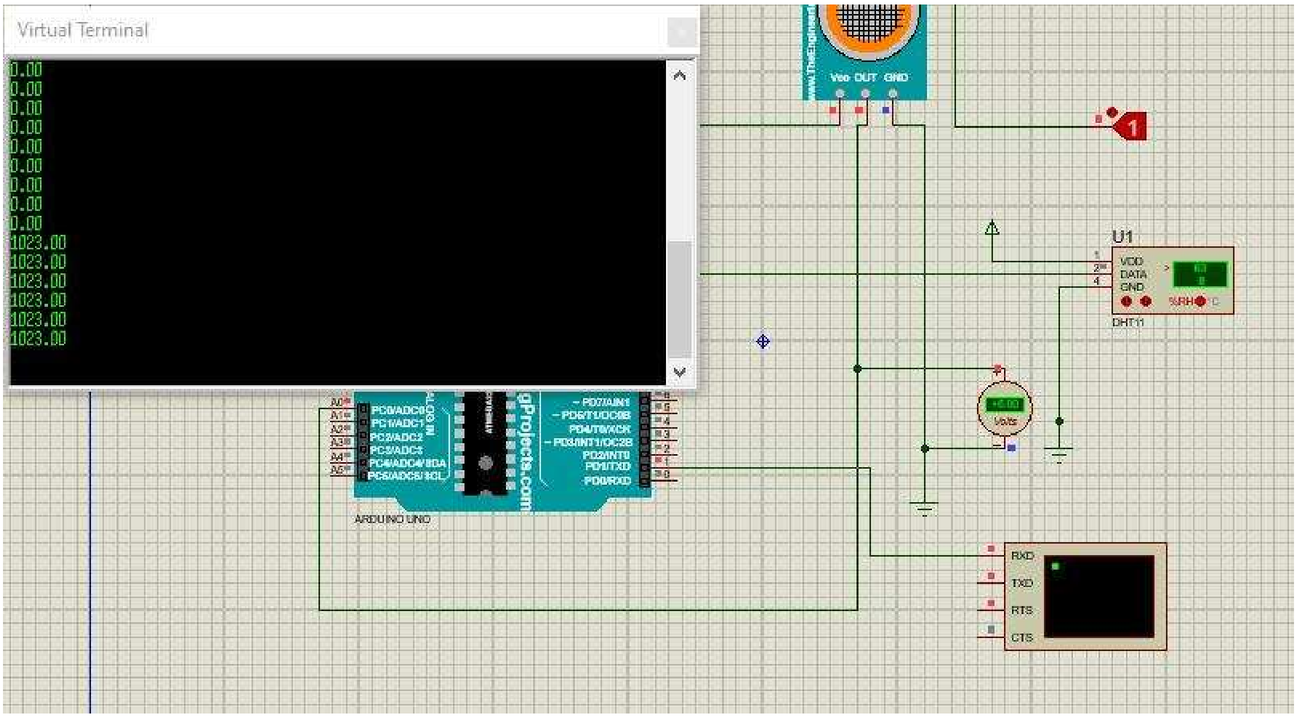



```
mq3

#define dht_apin 11 // Analog Pin sensor is connected to

dht DHT;
const int AOUTpin=A0;

void setup() {
  Serial.begin(9600);//sets the baud rate
  pinMode(AOUTpin, INPUT);
  pinMode(dht_apin, INPUT);

}

void loop() {
  DHT.read11(dht_apin);
  /*Serial.print(DHT.humidity);
  Serial.print(DHT.temperature); */

  float x;
  x = analogRead(AOUTpin);
  Serial.println(x);
  delay(1000);
}
```

```
Done compiling.

ls\\avr/bin/avr-gcc" -w -Os -g -flto -fuse-linker-plugin -Wl,--gc-sections -mmcu=atmega328p -o "C:\\Users\\Sumit\\AppData\\Local\\Temp\\arduino_build_830302/mq3.ino.elf" "C:\\Users\\Sumit\\AppD
ls\\avr/bin/avr-objcopy" -O ihex -j .eeprom --set-section-flags=.eeprom=alloc,load --no-change-warnings --change-section-lma .eeprom=0 "C:\\Users\\Sumit\\AppData\\Local\\Temp\\arduino_build_830
ls\\avr/bin/avr-objcopy" -O ihex -R .eeprom "C:\\Users\\Sumit\\AppData\\Local\\Temp\\arduino_build_830302/mq3.ino.elf" "C:\\Users\\Sumit\\AppData\\Local\\Temp\\arduino_build_830302/mq3.ino.hex"
er: C:\Users\Sumit\Documents\Arduino\libraries\DHTstable
ls\\avr/bin/avr-size" -A "C:\\Users\\Sumit\\AppData\\Local\\Temp\\arduino_build_830302/mq3.ino.elf"
space. Maximum is 32256 bytes.
memory, leaving 1834 bytes for local variables. Maximum is 2048 bytes.
```

2. Virtual Terminal PANEL





The Temperature and humidity sensor data will be uploaded on thingspeak server

HARDWARE COMPONENTS

NODE MCU

Fig 1. Node MCU

❖ NodeMCU is a low-cost open source IoT platform. It initially included firmware which runs on the ESP 8266 Wi -Fi SoC from Espressif Systems , and hardware which was based on the ESP -12 module .[6][7] Later ,support for the ESP32 32-bit MCU was added.

❖ Both the firmware and prototyping board designs are open source.[9]

❖ The firmware uses the Lua scripting language. The firmware is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. It uses many open source projects, such as lua-cjson[10] and SPIFFS.[11] Due to resource constraints, users need to select the modules relevant for their project and build a firmware tailored to their needs. Support for the 32-bit ESP32 has also been implemented.

DHT11 – TEMPERATURE AND HUMIDITY SENSOR

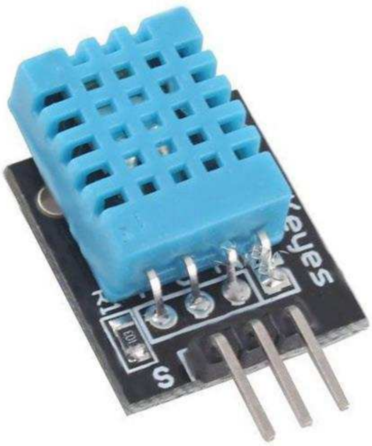


Fig 2. DHT11 Sensor

❖ The DHT11 is a basic, ultra low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air, and spits out a digital signal on the data pin (no analog input pins needed). Its fairly simple to use, but requires careful timing to grab data.

❖ Provides high reliability and long term stability.

BMP280 – AIR PRESSURE SENSOR

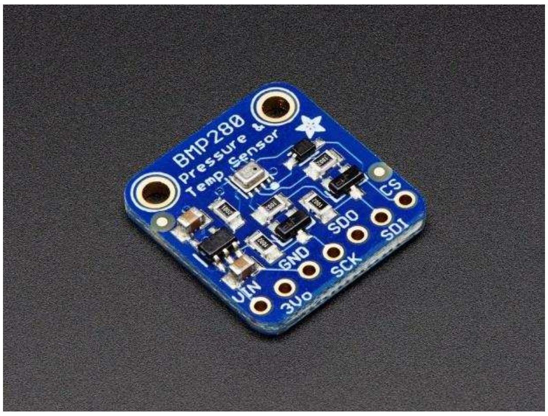


Fig 3. BMP280 Sensor

❖ An environmental sensor with temperature, barometric pressure that is the next generation upgrade to the BMP085/BMP180/BMP183. This sensor is great for all sorts of weather sensing.

❖ This precision sensor from Bosch is the best low- cost, precision sensing solution for measuring barometric pressure with ±1 hPa absolute accuraccy, and temperature with ±1.0°C accuracy.

MQ3 – FLAMMABLE GAS SENSOR

❖ MQ3 – Flammable gas sensor for detecting a wide range of gases, including Alcohol, benzene, CO and methane.

❖ Ideal for use in office or factory. MQ3 gas sensor has high sensitivity to methane, carbon mono-oxide and LPG , also sensitive to inflammable gases.

❖ It is with low cost and particularly suitable for gas index monitoring application.

❖ High sensitivity to Alcohol, LPG and Methane.

❖ Stable and Long Life

❖ The sensitivity of the sensor can be adjusted by using the potentiometer.

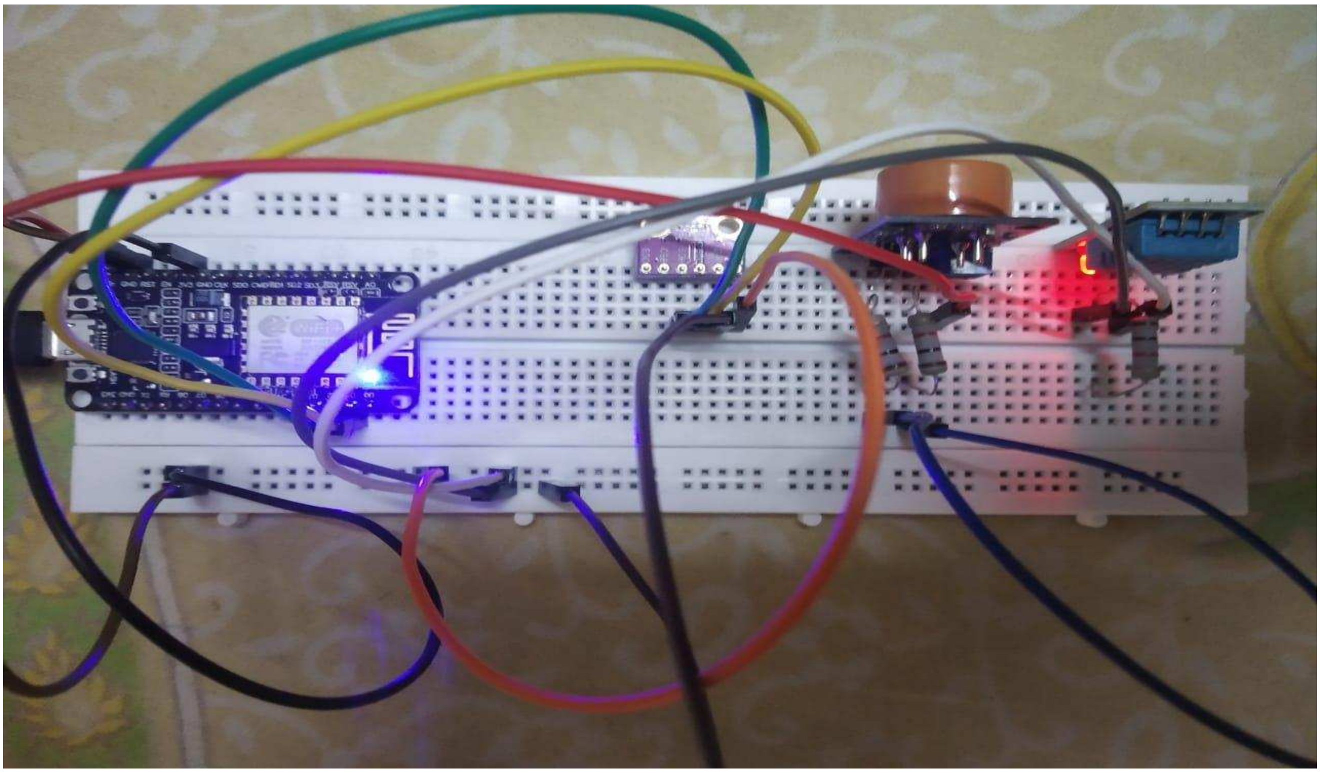
Fig 4. MQ3 Sensor

HARDWARE CONNECTIONS

Wiring Instructions:

The Vin of 3V is taken from the pin at the end. The ground voltage is chanlled from the ground pins on the other side of Vin. DHT 11 has three ports, VCC, GND and DATA. DATA port is mapped to D3/GPIO pin in MCU. MQ3 sensor has 4 pins, Vcc, GND, DO(digital output) and AO (analog output. We have mapped AO to the A0 pin in MCU. BMP 280 has 6 pins. Vcc, GND, and SDA for data input to D3 pin of MCU.

Final connection:



CODE AND THINGSPEAK PROGRAM

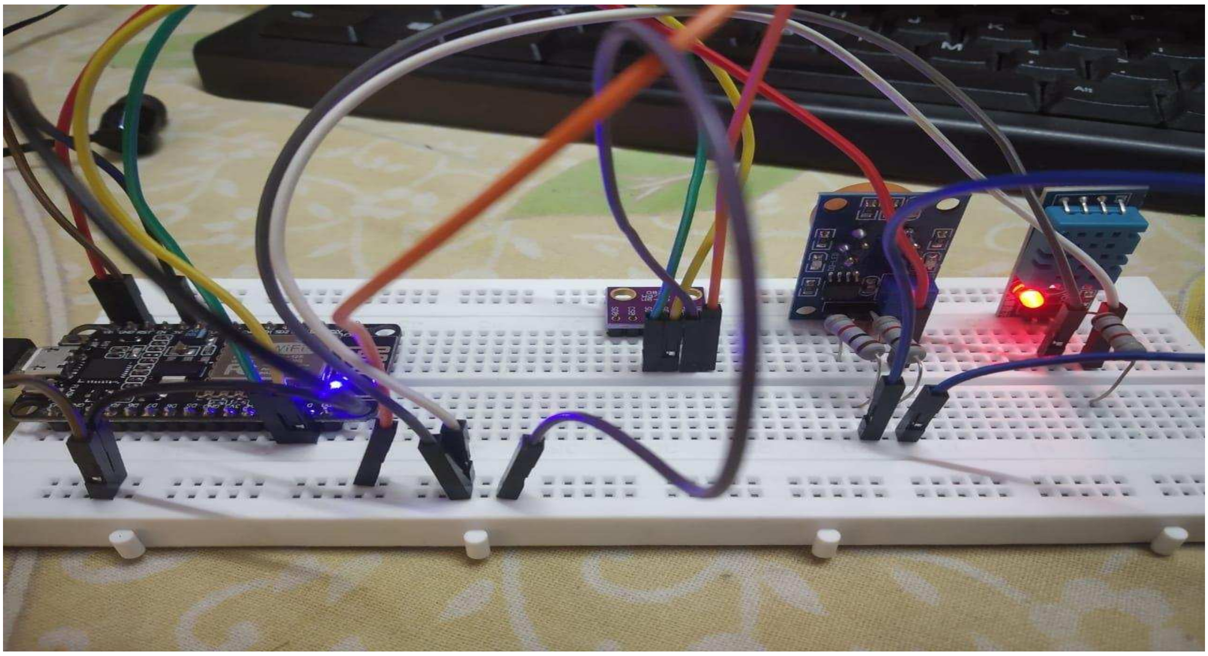
CODE

```
#include <DHT.h>  // Including library for dht
#include <ESP8266WiFi.h>
String apiKey = "HGVDLXSDY1HB56DJ";     //  Enter your Write API key from ThingSpeak
const char *ssid =  "KM TRIVEDI";     // replace with your wifi ssid and wpa2 key
const char *pass =  "krishna@123"
;const char* server = "api.thingspeak.com";
#define DHTPIN 0          //pin where the dht11 is connected
DHT dht(DHTPIN, DHT11);
WiFiClient client;
void setup() {
   Serial.begin(115200);
   delay(10);
   dht.begin();
   Serial.println("Connecting to ");
   Serial.println(ssid);
   WiFi.begin(ssid, pass);
   while (WiFi.status() != WL_CONNECTED)      {
       delay(500);
       Serial.print(".");
   }
  Serial.println("");
  Serial.println("WiFi connected");
}
void loop() {
      float h = dht.readHumidity();
      int p = random(10093502,10094106);
      int ht = random(4184,4206);
      float t = dht.readTemperature();
      float gas = analogRead(A0);
      if (isnan(gas))             {
            Serial.println("Failed to read from MQ-5 sensor!");
         return;
      }
      if (isnan(h) || isnan(t))                {
           Serial.println("Failed to read from DHT sensor!");
             return;
      }
       if (client.connect(server,80))   //   "184.106.153.149" or api.thingspeak.com
     {
        String postStr = apiKey;
        postStr +="&field1=";
        postStr += String(t);
        postStr +="&field3=";
        postStr += String(p/100);
        postStr +="&field5=";
        postStr += String(ht/10);
        postStr +="&field2=";
        postStr += String(h);
        postStr +="&field4=";
        postStr += String(gas/1023*100);
        postStr += "\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(t);
Serial.print(" degrees Celcius, Humidity: ");
Serial.print(h);
Serial.print(" Gas Detection: ");
Serial.print(gas);
Serial.println("%. Send to Thingspeak.");
 }
client.stop();
Serial.println("Waiting...");    // thingspeak needs minimum 15 sec delay between
updates  delay(1000);
}
```

RESULTS

The code runs perfectly and the modules connect to the wifi and they start transmitting data to the thingspeak server.



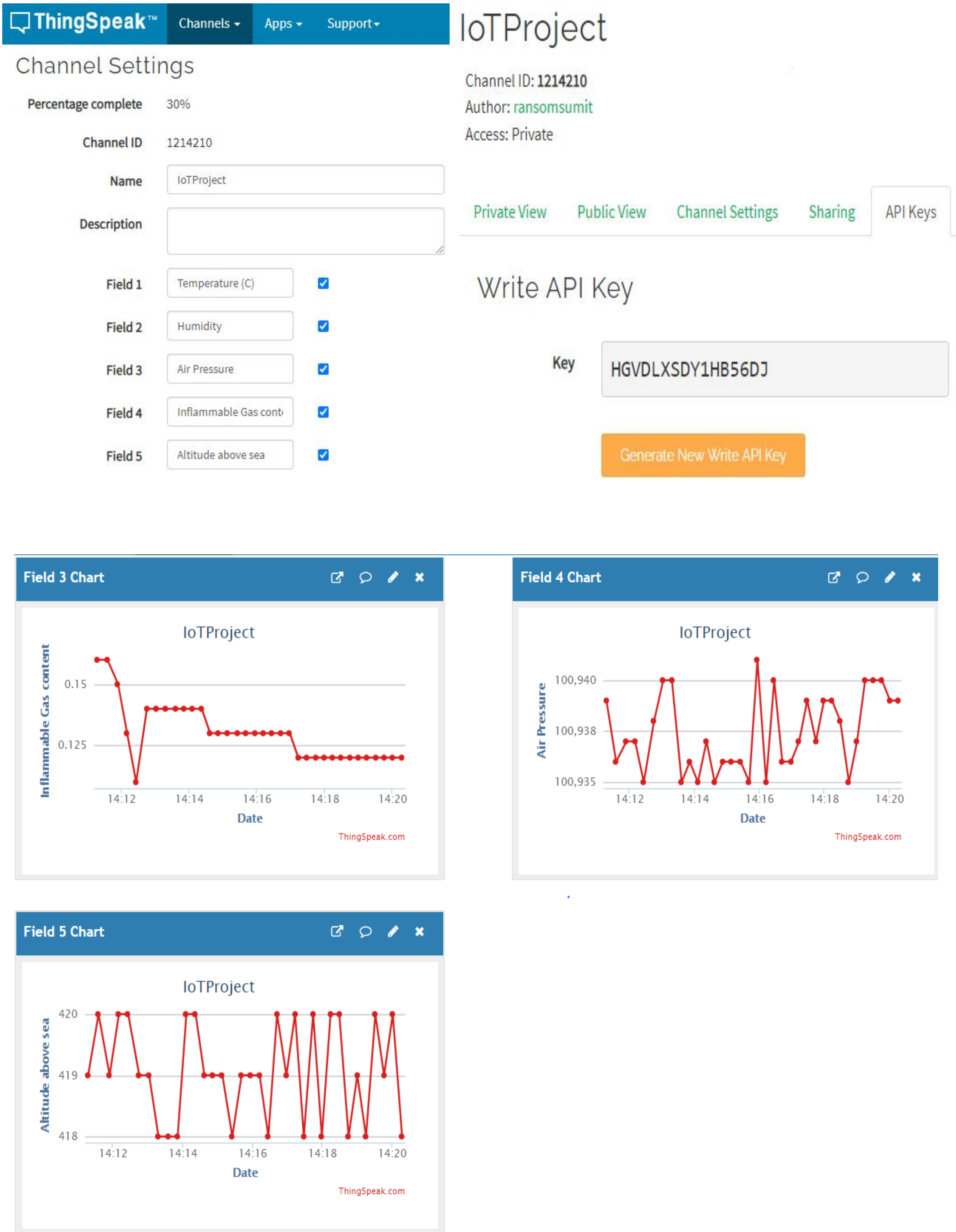


Fig 7. Graph plot of readings from MQ3 and BMP280 sensor

Fig8.Graph plot of readings from sensor DHT11 temp and humidity sensor

APP INTERFACE OF SENTINEL

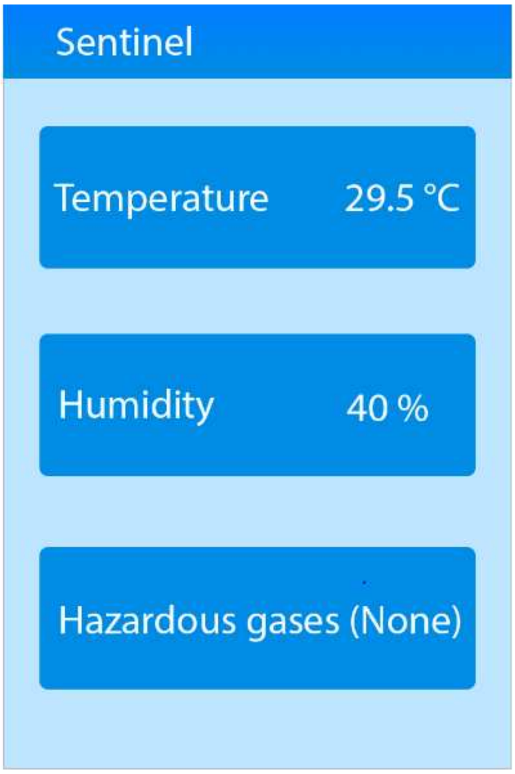


Fig 9. No hazardous gas detected

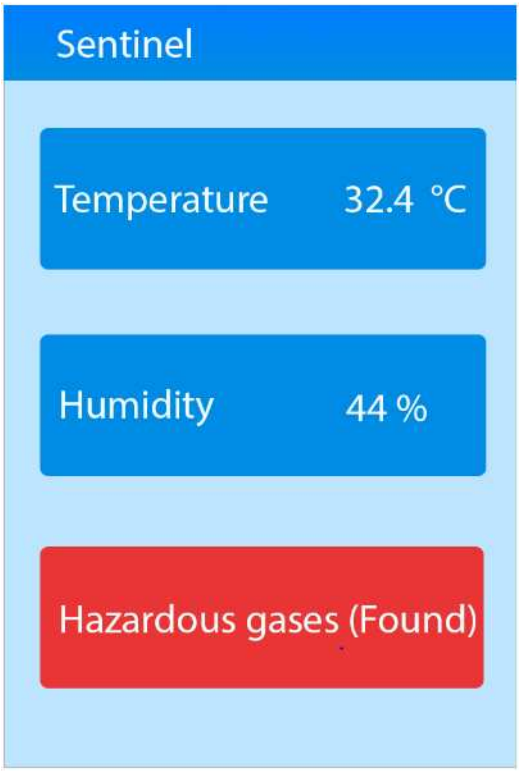


Fig 10. hazardous gas detected

APPLICATION

Compared with the traditional techniques of forest fire detection, a wireless sensor network paradigm based on a ZigBee technique was proposed. The proposed technique is in real time, given the exigencies of forest fires. The architecture of a wireless sensor network for forest fire detection is described.

The Navy program, Damage Control -Automation for Reduced Manning is focused on enhancing automation of ship functions and damage control systems. A key element to this objective is the improvement of current fire detection systems. As in many applications, it is desired to increase detection sensitivity and,more importantly increase the reliability of the detection system through improved nuisance alarm immunity.

A novel video smoke detection method using both color and motion features is presented. The result of optical flow is assumed to be an approximation of motion field. Background estimation and color-based decision rule are used to determine candidate smoke regions. The Lucas Kanade optical flow algorithm is proposed to calculate the optical flow of candidate regions

The possibility of detecting small forest fires with the help of a simple and cheap lidar operating at 0.532-μm wavelength up to distances of about 6.5 km is demonstrated. The values of the signal-to-noise ratio (SNR) achieved in the experiments are consistent with theoretical estimations obtained by computational modeling of the lidar detection process, including simulation of the smoke-plume shape and of the laser beam–plume interaction.

An integrated system based on video surveillance is presented for automatic fire detection and suppression. The system is composed of two modules, including fire detection and fire suppression. The fire detection module makes full use of traditional CCD cameras for fire recognition.

CONCLUSION

At the end of completing this project we aim to achieve the development of following product features. The sentinel fire fighter helmet will send and retrieve information by using six sensors connected to a central microcontroller capable of connecting to nearby WiFi networks. When information from the environment will be extracted , we will be capable of sending this information using WiFi following the IOT protocol to a remote real-time database where it is stored and processed. When this information will be retrieved, it will be used to create a app user interface that will display the information in a real-time framework to show changes as they happen . This interface will allow all subscribers to the app to monitor the situation different firefighters are and providing assistance when needed.

REFERENCES

[1] Junguo ZHANG, Wenbin LI, Ning HAN, Jiangming KAN E , Forest fire detection system based on a ZigBee wireless sensor network, Higher Education Press and Springer-Verlag 2019

[2]Susan L. Rose-Pehrsson , Sean J. Hart, Thomas T. Street, Frederick W. Williams , and Mark H. Hammond , Early Warning Fire Detection System Using a Probabilistic Neural Network , Naval Research Laboratory, Chemistry Division, Washington, © Springer Nature Singapore Pte Ltd. 2019

[3] Yu Chunyu , Fang Jun, Wang Jinjun and Zhang Yongming , Video Fire Smoke Detection Using Motion and Color Features , State Key Laboratory of Fire Science, USTC, Number 96 Jin Zhai Road, Hefei, Anhui, China 2019 Springer Science + Business Media, LLC.

[4]Utkin, A., Lavrov, A., Costa, L. et al. , Detection of small forest fires by lidar , ApplPhys B 74, 77–83 29 November 2019 • © Springer-Verlag 2019

[5]VinayDubey, Prashant Kumar and Naveen Chauhan , Forest Fire Detection System Using IoT and Artificial Neural Network , Department of Computer Science and Engineering, National Institute of Technology Hamirpur, Hamirpur, Himachal Pradesh, India © Springer Nature Singapore Pte Ltd. 2019

[6] FeiniuYuan ,An integrated fire detection and suppression system based on widely available video surveillance , © Springer-Verlag 2019

[7] Masum A.K.M., Shan-A-Alahi A., Al Noman A., Uddin M.N., Azam K.I., Rakib M.G.S. , IoT-Based Smart Monitoring System to Ensure Worksite Safety —A Context of Garment Industry in Bangladesh (2020), vol 89. Springer 2020, Singapore

[8] Ouni, S., TrabelsiAyoub, Z. &Kamoun, F. , Auto-organization approach with adaptive frame periods for IEEE 802.15.4/zigbee forest fire detection system , Auto-organization approach with adaptive frame periods for IEEE 802.15.4/zigbee forest fire detection system , Springer-Verlag 2019

[9] Ramesh P.A., Siva Prasad Y.A., Chidananda G. , Monitor and Abolish the Wildfire Using Internet of Things and Cloud Computing , Advances in Intelligent Systems and Computing, vol 1054. Springer (2020), Singapore

[10] "Development and Application of the Smart Helmet for Disaster and Safety ", MingiJeongIoT Research Division Electronics and Telecommunications Research Institute Daejeon, Republic of Korea , Hyesun Lee IoT Research Division Electronics and Telecommunications Research Institute (ETRI) Daejeon , Republic of Korea , Kang Bok Lee IoT Research Division Electronics and Telecommunications Research Institute Daejeon, Republic of Korea, 978-1-5386-5041-7/18/$31.00 ©2019 IEEE

[11] A Cloud and Contiki based Fire Detection System using Multi-Hop Wireless Sensor Networks, ICEMIS '18: Proceedings of the Fourth International Conference on Engineering & MIS 2018June 2018 Article No.: 66 Pages 1–5

[12] Aguirre J., Ordóñez A., Ordóñez H. (2017) Low-Cost Fire Alarm System Supported on the Internet of Things. In: Solano A., Ordoñez H. (eds) Advances in Computing . CCC 2017.Communications in Computer and Information Science, vol 735. Springer, Cham

[13] Datta S., Das A.K., Kumar A., Khushboo , Sinha D. (2020) Authentication and Privacy Preservation in IoT Based Forest Fire Detection by Using Blockchain – A Review. In: Nain N., Vipparthi S. (eds) 4th International Conference on Internet of Things and Connected Technologies (ICIoTCT), 2019. ICIoTCT 2019.Advances in Intelligent Systems and Computing, vol 1122. Springer, Cham

[14] Ahlawat H.D., Chauhan R.P. (2020) Detection and Monitoring of Forest Fire Using Serial Communication and Wi-Fi Wireless Sensor Network. In: Singh P., Bhargava B., Paprzycki M., Kaushal N., Hong WC. (eds) Handbook of Wireless Sensor Networks : Issues and Challenges in Current Scenario's. Advances in Intelligent Systems and Computing, vol 1132. Springer, Cham

[15] Fan Y., Zhang L., Du Y. (2020) A New Type Building Fire Protection Facility Monitoring System. In: Abawajy J., Choo KK., Islam R., Xu Z., Atiquzzaman M. (eds) International Conference on Applications and Techniques in Cyber Intelligence ATCI 2019. ATCI 2019.Advances in Intelligent Systems and Computing, vol 1017. Springer, Cham

[16] Lourenço P., Fantoni A., Vieira M. (2019) Simulation of an Early Warning Fire System. In: Camarinha-Matos L., Almeida R., Oliveira J. (eds) Technological Innovation for Industry and Service Systems. DoCEIS 2019. IFIP Advances in Information and Communication Technology, vol 553. Springer, Cham

[17] Tomar R., Tiwari R., Sarishma (2019) Information Delivery System for Early Forest Fire Detection Using Internet of Things. In: Singh M., Gupta P., Tyagi V., Flusser J., Ören T., Kashyap R. (eds) Advances in Computing and Data Sciences. ICACDS 2019.Communications in Computer and Information Science, vol 1045. Springer, Singapore

[18] Apiecionek Ł., Krieger U. (2019) Monitoring - The IoT Monitoring Tool for Fire Brigades. In: Choraś M., Choraś R. (eds) Image Processing and Communications Challenges 10. IP&C 2018. Advances in Intelligent Systems and Computing, vol 892. Springer, Cham

[19] Khoumeri EH., Cheggou R., Bekhouche M.EA., Oubraham S. (2018) A Safety IoT- Based System for a Closed Environment. In: Mehmood R., Bhaduri B., Katib I., Chlamtac I. (eds) Smart Societies, Infrastructure, Technologies and Applications. SCITA 2017. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 224. Springer, Cham

[20] T. Sahithi, T. Pranathi and A. Pravin, Automatic Fire Rescue System using IoT,International Conference on Communication and Signal Processing, April 4-6, 2019, India,978-1-5386-7595-3/19/$ 31.00 ©2019 IEEE

Plagiarism Verification Report

Submissions Overview

Background Information [what is this?]

Batch file name: IoT Paper.docx

Report generated on: 27/05/2021, 05:40:35 PM

Checking Parameters [what is this?]

Matching scope(s): Within submission, Internet

Leniency: Detailed matching with threshold 70%

Minimum sentence length: Sentences with more than or equal to 3 meaningful words were checked

Similarity Statistics

Similarity Statistics [what is this?]

Total number of documents: 1

Number of documents which can be processed: 1

Number of documents which cannot be processed: 0

Show 10 entries Search:

Entry	Document	Status	Similarity	Action
1	IoT_Paper.docx	processed	10/180=5.55%	View details

Showing 1 to 1 of 1 entries

First Previous 1 Next Last