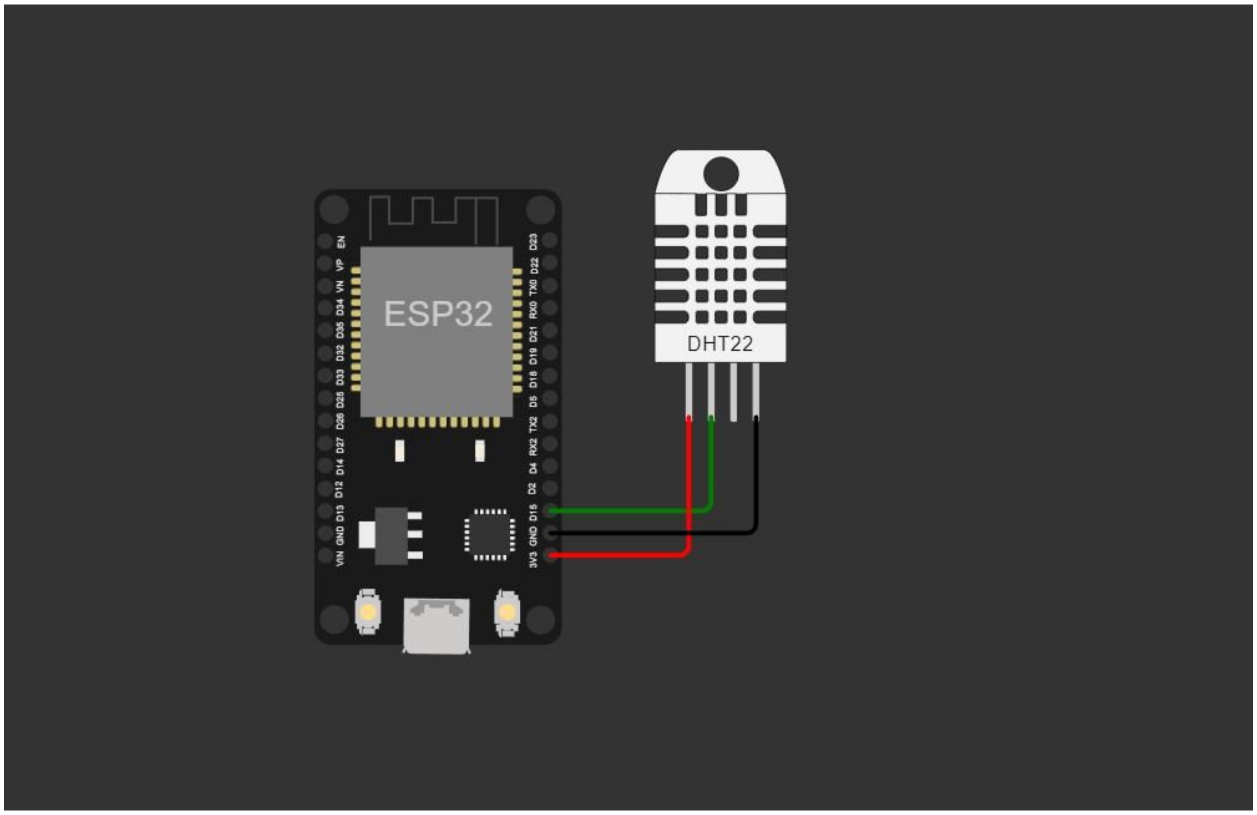
Team ID	PNT2022TMID15675
Project Name	Industry Specific Intelligent Fire Management System

CODE:

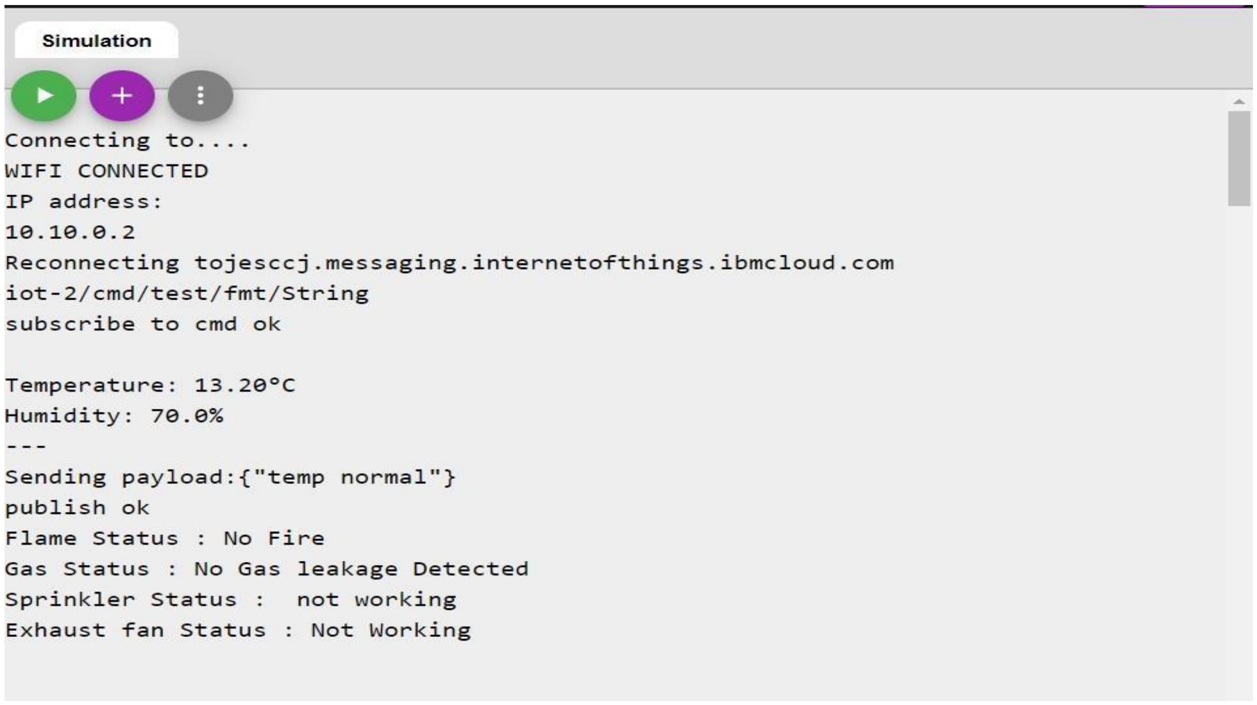
```
#include <WiFi.h>

#include <PubSubClient.h>

#include <time.h> #include "DHTesp.h" #define temp_pin 15

void callback(char* subscribetopic,byte* payload, unsigned int payloadLength); #define ORG
"jesccj"

#define DEVICE_TYPE "ESP32_Controller"

#define DEVICE_ID "Trini"

#define TOKEN "*Vzh&EwwgbRpqohJd+"

String data3;

char server[]= ORG ".messaging.internetofthings.ibmcloud.com"; char
publishTopic[]="iot2/evt/Data/fmt/json"; char subscribeTopic[]="iot-2/cmd/test/fmt/String"; char
authMethod[]="usetoken-auth"; char token[]=TOKEN;

char clientID[]="d:"ORG":"DEVICE_TYPE":"DEVICE_ID;

WiFiClient wifiClient;

PubSubClient client(server,1883,callback,wifiClient);


const int DHT_PIN = 15;


DHTesp dhtSensor;   bool

exhaust_fan_on = false;

bool sprinkler_on = false;


float temperature  = 0;

int gas = 0;  int flame

= 0;
```

```
String flame_status = "";

String accident_status = "";

String sprinkler_status = "";


void setup() {     Serial.begin(99900);


wificonnect();  mqttconnect();


  dhtSensor.setup(DHT_PIN, DHTesp::DHT22);

}


void loop() {

srand(time(0));


    //initial variable


    temperature = random(-20,125);     gas = random(0,1000);     int flamereading = 
random(200,1024);

    flame = map(flamereading,0,1024,0,2);


  TempAndHumidity  data = dhtSensor.getTempAndHumidity();


Serial.println("Temperature: "+ String(data.temperature, 2) + "°C");

Serial.println("Humidity: " + String(data.humidity, 1) + "%");    Serial.println("---

");   delay(1000);

if(data.temperature<38){PublishData1(data.temperature);

flame_status = "No Fire";

        Serial.println("Flame Status : "+flame_status);

    }
    else{ PublishData2(data.temperature);         flame_status = "Fire is Detected";
Serial.println("Flame Status : "+flame_status);


    }
    if(data.humidity<30){
        Serial.println("Gas Status : Gas leakage Detected");
    }
else{
        exhaust_fan_on = false;
        Serial.println("Gas Status : No Gas leakage Detected");
    }

    //send the sprinkler status     if(data.temperature<38){
sprinkler_status = " not working";
        Serial.println("Sprinkler Status : "+sprinkler_status);
    }
else{
        sprinkler_status = " working";
        Serial.println("Sprinkler Status : "+sprinkler_status);
    }

    //toggle the fan according to gas


    if(data.humidity<30){        exhaust_fan_on = true;
        Serial.println("Exhaust fan Status : Working");
    }
else{
        exhaust_fan_on = false;
        Serial.println("Exhaust fan Status : Not Working");
    }
    Serial.println("");     Serial.println("");

    Serial.println(" -------------------------************---------------------");     Serial.println("");
Serial.println("");     delay(1000);     if(!client.loop()){

mqttconnect();

}

}  void PublishData1(float temp){    mqttconnect();

   String payload= "{\"temp normal\"}";    Serial.print("Sending payload:");

Serial.println(payload);


   if(client.publish(publishTopic,(char*)payload.c_str())){

    Serial.println("publish ok");

  } else{

    Serial.println("publish failed");

  }

}

   void PublishData2(float temperature){

mqttconnect();

   String payload = "{\"temp\":";  payload

+= temperature;

payload += ",\"ALERT!!\":""\"temperature greater than 38\""; payload += "}"; Serial.print("Sending
payload: "); Serial.println(payload);

  if(client.publish(publishTopic,(char*)payload.c_str())){

    Serial.println("publish ok");

  } else{

    Serial.println("publish failed");

  }

}

void mqttconnect(){   if(!client.connected()){     Serial.print("Reconnecting to");
Serial.println(server);

    while(!!!client.connect(clientID, authMethod, token)){
      Serial.print(".");
 delay(500);
    }
    initManagedDevice();
    Serial.println();
  }
}

void wificonnect(){
  Serial.println();
  Serial.print("Connecting to");

  WiFi.begin("Wokwi-GUEST","",6);
  while(WiFi.status()!=WL_CONNECTED){     delay(500);     Serial.print(".");
  }
  Serial.println("");
  Serial.println("WIFI CONNECTED");   Serial.println("IP address:");
  Serial.println(WiFi.localIP());
}

void initManagedDevice(){   if(client.subscribe(subscribeTopic)){
Serial.println((subscribeTopic));
    Serial.println("subscribe to cmd ok");
  }else{
    Serial.println("subscribe to cmd failed");
  }
}

void callback(char* subscribeTopic, byte* payload, unsigned int payloadLength){
  Serial.print("callback invoked for topic:");
  Serial.println(subscribeTopic);   for(int i=0; i<payloadLength; i++){
data3 += (char)payload[i];
  }
}
```

CIRCUIT



OUTPUT



```
Connecting to....
WIFI CONNECTED
IP address:
10.10.0.2
Reconnecting tojesccj.messaging.internetofthings.ibmcloud.com
iot-2/cmd/test/fmt/String
subscribe to cmd ok

Temperature: 13.20°C
Humidity: 70.0%
---
Sending payload:{"temp normal"}
publish ok
Flame Status : No Fire
Gas Status : No Gas leakage Detected
Sprinkler Status :  not working
Exhaust fan Status : Not Working
```