

# SPRINT-1

PROJECT	INDUSTRY-SPECIFIC INTELLIGENT FIRE MANAGEMENT SYSTEM
TEAM ID	PNT2022TMID15035

## PROGRAM

```
#include "DHTesp.h"
#include <cstdlib>
#include <time.h>

const int DHT_PIN = 15;

bool is_exhaust_fan_on = false;
bool is_sprinkler_on = false;

float temperature = 0;

int gas_ppm = 0;
int flame = 0;
int flow = 0;

String flame_status = "";
String accident_status = "";
String sprinkler_status = "";

DHTesp dhtSensor;
```

```

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

  /*** sensor pin setups ***/
  dhtSensor.setup(DHT_PIN, DHTesp::DHT22);
  //if real gas sensor is used make sure the sensor is heated up for accurate readings
  /*
   - Here random values for readings and stdout were used to show the
     working of the devices as physical or simulated devices are not
     available.
  */
}

void loop() {

  TempAndHumidity data = dhtSensor.getTempAndHumidity();

  //setting a random seed
  srand(time(0));

  //initial variable activities like declaring , assigning
  temperature = data.temperature;
  gas_ppm = rand()%1000;
  int flamereading = rand()%1024;
  flame = map(flamereading,0,1024,0,1024);
  int flamerange = map(flamereading,0,1024,0,3);
  int flow = ((rand()%100)>50?1:0);

  //set a flame status based on how close it is.....
  switch (flamerange) {
  case 2:    // A fire closer than 1.5 feet away.

```

```

    flame_status = "Close Fire";
    break;
case 1:    // A fire between 1-3 feet away.
    flame_status = "Distant Fire";
    break;
case 0:    // No fire detected.
    flame_status = "No Fire";
    break;
}

//toggle the fan according to gas in ppm in the room
if(gas_ppm > 100){
    is_exhaust_fan_on = true;
}
else{
    is_exhaust_fan_on = false;
}

//find the accident status 'cause fake alert may be caused by some mischief activities
if(temperature < 40 && flamerange ==2){
    accident_status = "need auditing";
    is_sprinkler_on = false;
}
else if(temperature < 40 && flamerange ==0){
    accident_status = "nothing found";
    is_sprinkler_on = false;
}
else if(temperature > 50 && flamerange == 1){
    is_sprinkler_on = true;
    accident_status = "moderate";
}

```

```

else if(temperature > 55 && flamerange == 2){
    is_sprinkler_on = true;
    accident_status = "severe";
}else{
    is_sprinkler_on = false;
    accident_status = "nil";
}

```

```

//send the sprinkler status
if(is_sprinkler_on){
    if(flow){
        sprinkler_status = "working";
    }
    else{
        sprinkler_status = "not working";
    }
}
else if(is_sprinkler_on == false){
    sprinkler_status = "now it shouldn't";
}
else{
    sprinkler_status = "something's wrong";
}

```

```

//Obviously the output.It is like json format 'cause it will help us for future sprints
String out = "{\n\t\"senor_values\":{";
out+="\n\t\t\"gas_ppm\": "+String(gas_ppm)+", ";
out+="\n\t\t\"temperature\": "+String(temperature,2)+", ";
out+="\n\t\t\"flame\": "+String(flame)+", ";
out+="\n\t\t\"flow\": "+String(flow)+", \n\t}";

```

```
out+="\n\t\"output\":{";  
out+="\n\t\t\"is_exhaust_fan_on\":\""+String((is_exhaust_fan_on)?"true":"false")+",";"  
out+="\n\t\t\"is_sprinkler_on\":\""+String((is_sprinkler_on)?"true":"false")+",";"  
out+="\n\t}";  
out+="\n\t\"messages\":{";  
out+="\n\t\t\"fire_status\":\""+flame_status+",";"  
out+="\n\t\t\"flow_status\":\""+sprinkler_status+",";"  
out+="\n\t\t\"accident_status\":\""+accident_status+",";"  
out+="\n\t}";  
out+="\n}";  
Serial.println(out);  
  
delay(1000);  
}
```

## DIAGRAM.JSON:

```
WOKWI  SAVE  SHARE  ESP322-DHT22 
```

```
sketch.ino diagram.json libraries.txt Library Manager ▼
```

```
1 {
2   "version": 1,
3   "author": "PNT2022TMID34516",
4   "editor": "wokwi",
5   "parts": [
6     { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": -16.32, "left": -0.82, "attrs": {} },
7     {
8       "type": "wokwi-dht22",
9       "id": "dht1",
10      "top": -30.22,
11      "left": 165.89,
12      "attrs": { "temperature": "59.3" }
13    }
14  ],
15  "connections": [
16    [ "esp:TX0", "$serialMonitor:RX", "", [ ] ],
17    [ "esp:RX0", "$serialMonitor:TX", "", [ ] ],
18    [ "dht1:SDA", "esp:D15", "green", [ "v0" ] ],
19    [ "dht1:VCC", "esp:3V3", "red", [ "v0" ] ],
20    [ "dht1:GND", "esp:GND.1", "black", [ "v0" ] ]
21  ]
22 }
```

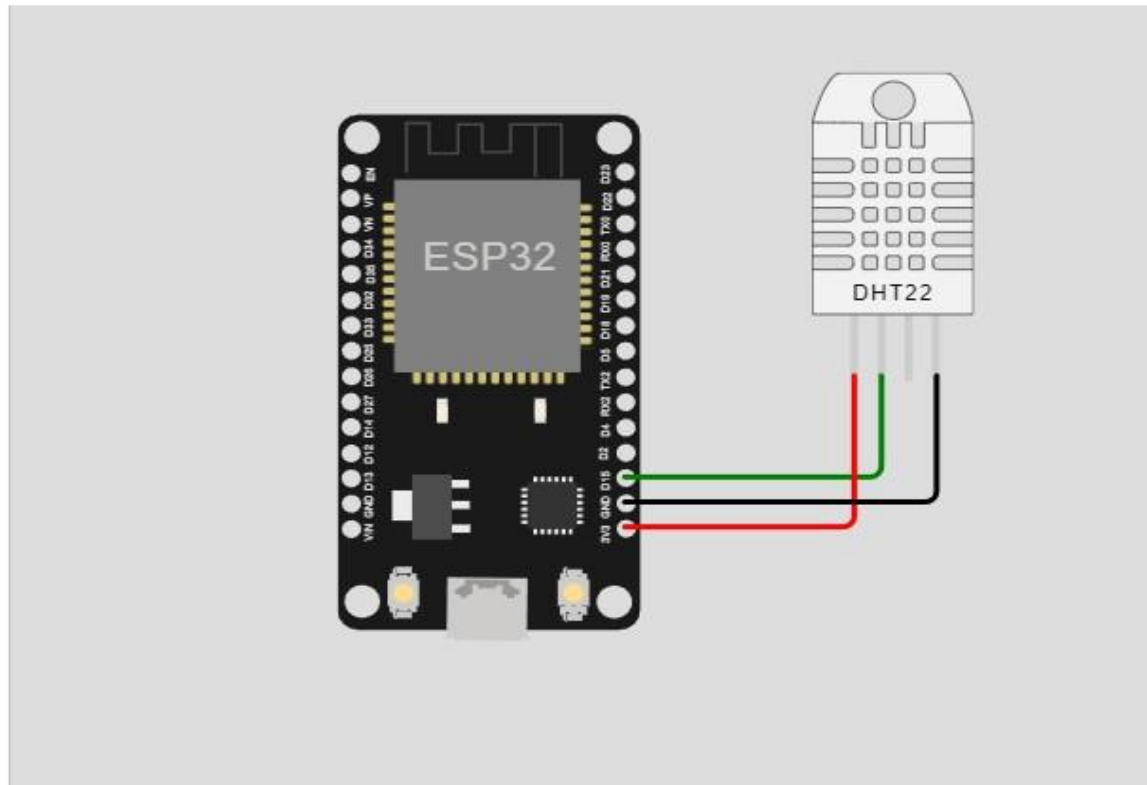
## LIBRARIES TEXT:



The screenshot shows the Wokwi IDE interface. At the top, there is a dark header bar with the Wokwi logo, a 'SAVE' button, a 'SHARE' button, a heart icon, and the text 'ESP322-DHT22' with an edit icon. Below the header, there is a light gray bar with tabs for 'sketch.ino', 'diagram.json', and 'libraries.txt' (which is active). To the right of the tabs is a 'Library Manager' dropdown menu. The main area shows the content of 'libraries.txt' with line numbers 1 through 5 on the left. The text in the editor is as follows:

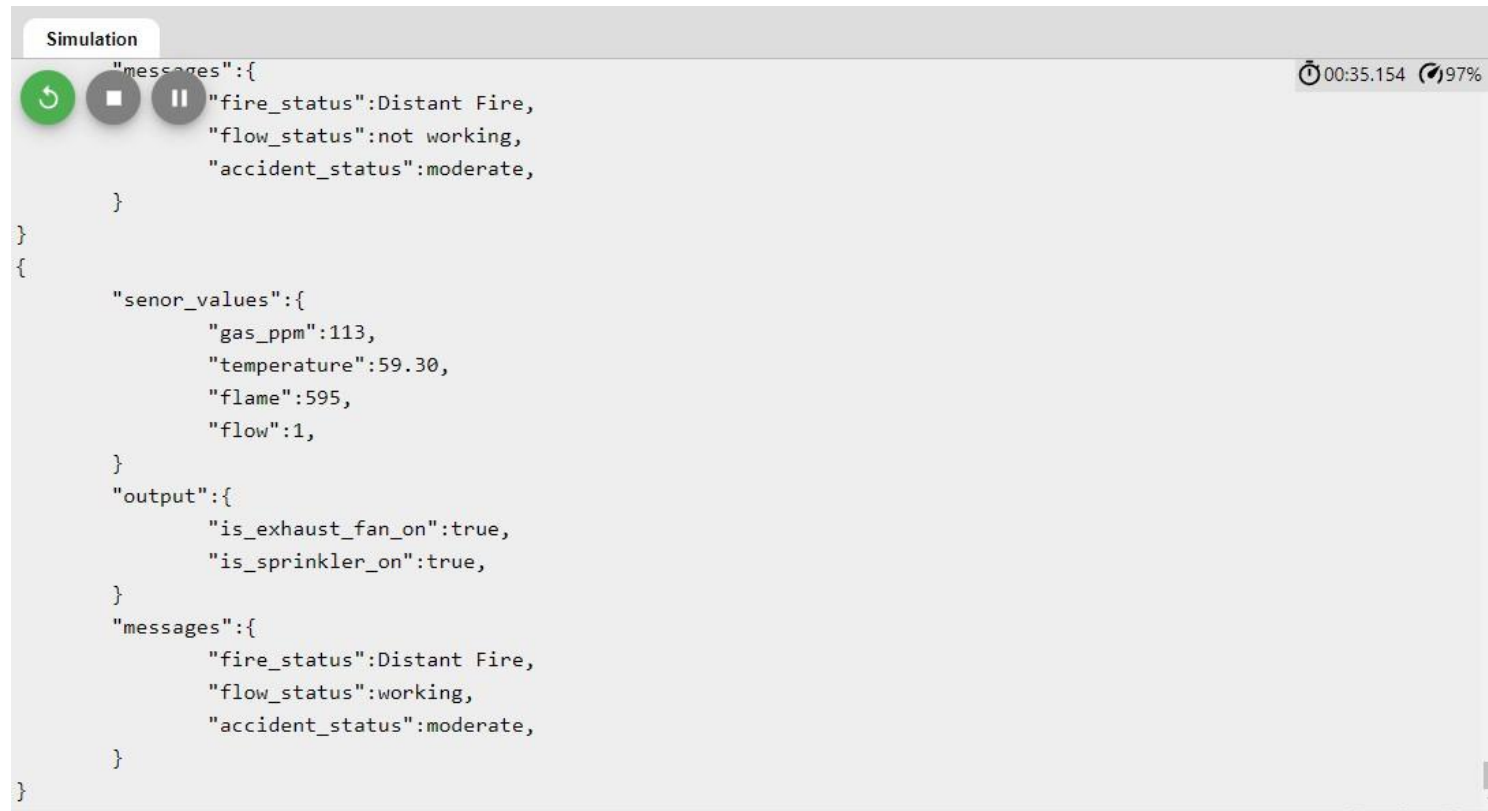
```
1 # Wokwi Library List
2 # See https://docs.wokwi.com/guides/libraries
3
4 DHT sensor library for ESPx
5 ArduinoJson
```

## CIRCUIT:





## OUTPUT:



The screenshot shows a Wokwi simulation window titled "Simulation". In the top-left corner, there are three circular icons: a green play button, a grey square stop button, and a grey pause button. The top-right corner displays a timer at "00:35.154" and a battery level at "97%". The main area of the window contains a JSON object representing simulation data. The JSON is as follows:

```
{
  "messages": {
    "fire_status": "Distant Fire",
    "flow_status": "not working",
    "accident_status": "moderate",
  }
},
{
  "senor_values": {
    "gas_ppm": 113,
    "temperature": 59.30,
    "flame": 595,
    "flow": 1,
  },
  "output": {
    "is_exhaust_fan_on": true,
    "is_sprinkler_on": true,
  },
  "messages": {
    "fire_status": "Distant Fire",
    "flow_status": "working",
    "accident_status": "moderate",
  }
}
```

## WOKWI LINK:

<https://wokwi.com/projects/347926417442341458>