

**TEAM ID : PNT2022TMID47920**

**PROJECT TITLE : Industry-Specific Intelligent Fire Management System**

## **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 fire = 0; int
```

```
flow = 0;
```

```
String fire_status = "";
```

```
String accident_status = "";
```

```
String sprinkler_status = ""; DHTesp
```

```
dhtSensor;
```

```
void setup() {
```

```

Serial.begin(99900);

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

void loop() {

    TempAndHumidity data = dhtSensor.getTempAndHumidity();

    srand(time(0));

    temperature = data.temperature;  gas_ppm =
    rand()%1000;  int firereading = rand()%1024;  fire
    = map(firereading,0,1024,0,1024);  int firerange =
    map(firereading,0,1024,0,3);      int flow =
    ((rand()%100)>50?1:0);

    switch (firerange) {  case 2:
    fire_status = "Close Fire";
    break;  case 1:    fire_status =
    "Distant Fire";  break;  case 0:
    fire_status = "No Fire";  break;

    }

```

```
    if(gas_ppm > 100){    is_exhaust_fan_on =
true;

    }

    else{    is_exhaust_fan_on =
false;

    }

    if(temperature < 40 && firerange ==2){
accident_status = "need auditing";
is_sprinkler_on = false;

    }

    else if(temperature < 40 && firerange ==0){
accident_status = "nothing found";
is_sprinkler_on = false;

    }

    else if(temperature > 50 && firerange == 1){
is_sprinkler_on = true;    accident_status =
"moderate";

    }

    else if(temperature > 55 && firerange == 2){
is_sprinkler_on = true;    accident_status =
"severe";

    }else{    is_sprinkler_on =
false;    accident_status =
"nil"; }

    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";

}


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\"fire\":" +String(fire)+", "; out+="\n\t\t\"flow\":" +String(flow)+",\n\t}";
out+="\n\t\"output\":{"; out+="\n\t\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}";

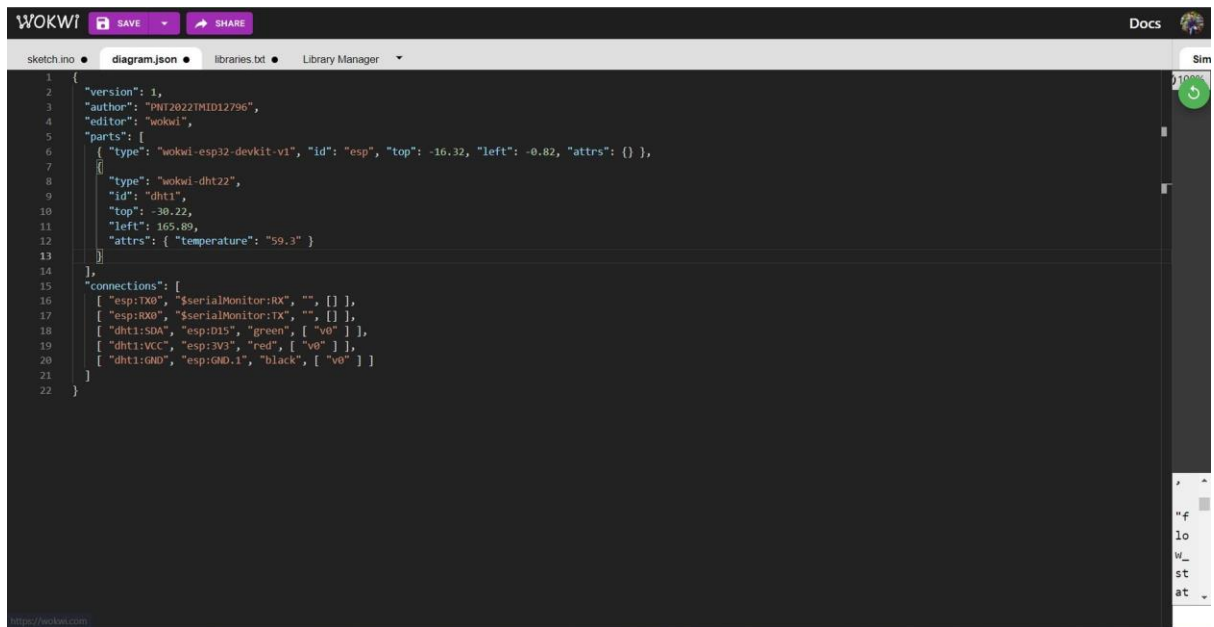
    out+="\n\t\"messages\":{";

out+="\n\t\t\"fire_status\":" +fire_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);

```

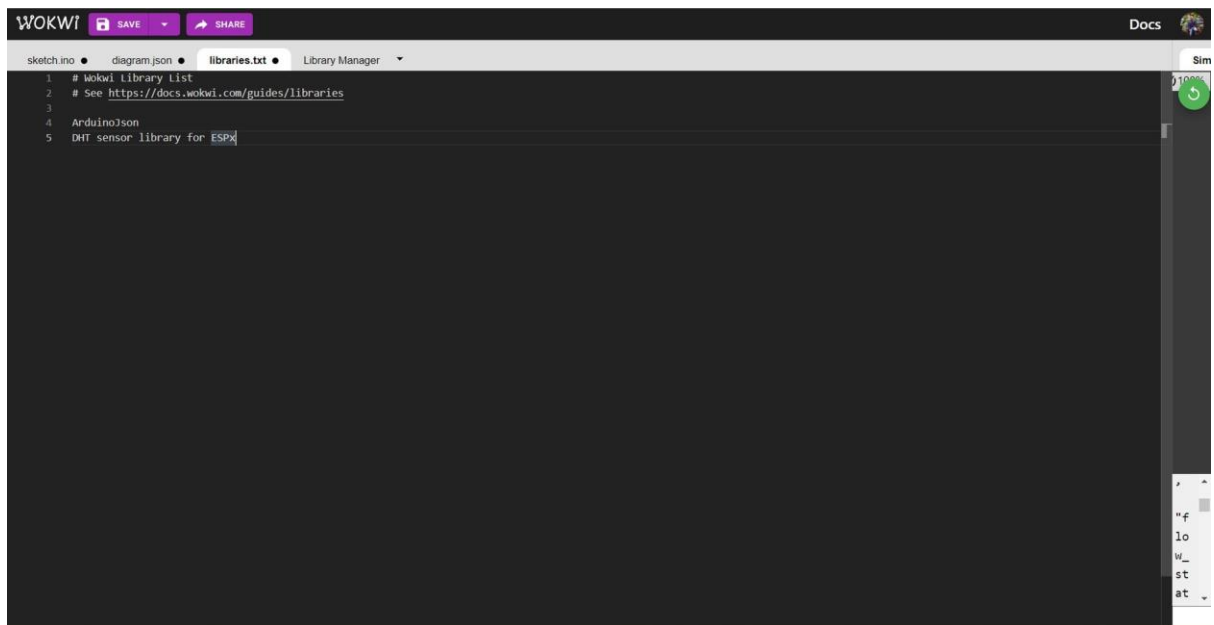
```
delay(1000);  
  
}
```

## DIAGRAM.JSON:



```
1 {  
2   "version": 1,  
3   "author": "PMT2822TMD12796",  
4   "editor": "wokwi",  
5   "parts": [  
6     {  
7       "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": -16.32, "left": -0.82, "attrs": {} },  
8     {  
9       "type": "wokwi-dht22",  
10      "id": "dht1",  
11      "top": -30.22,  
12      "left": 165.89,  
13      "attrs": { "temperature": "59.3" }  
14    }  
15  ],  
16  "connections": [  
17    [ "esp:TX0", "$serialMonitor:RX", "", [] ],  
18    [ "esp:RX0", "$serialMonitor:TX", "", [] ],  
19    [ "dht1:SDA", "esp:D15", "green", [ "v0" ] ],  
20    [ "dht1:VCC", "esp:3V3", "red", [ "v0" ] ],  
21    [ "dht1:GND", "esp:GND.1", "black", [ "v0" ] ]  
22  ]  
23 }
```

## LIBRARIES TEXT:



```
1 # Wokwi library list  
2 # See https://docs.wokwi.com/guides/libraries  
3  
4 ArduinoJson  
5 DHT sensor library for ESP8266
```

## CIRCUIT:

