

Project Development

Phase

Sprint - I

Date	29 October 2022
Team ID	PNT2022TMID47980
Project Name	Industry-Specific Intelligent Fire Management System

Link: <https://wokwi.com/projects/322410731508073042>

OUTPUT:

The screenshot displays the Wokwi web IDE interface. On the left, the code editor shows the following C++ code:

```
1 #include <time.h>
2
3 bool exhaust_fan_on = false;
4 bool sprinkler_on = false;
5
6 float temperature = 0;
7 int gas = 0;
8 int flame = 0;
9
10 String flame_status = "";
11 String accident_status = "";
12 String sprinkler_status = "";
13
14 void setup() {
15   Serial.begin(99900);
16 }
17
18 void loop() {
19
20   //setting a random seed
21   srand(time(0));
22
23   //initial variable
24
25   temperature = random(-20,125);
26   gas = random(0,1000);
27   int flamereading = random(200,1024);
28   flame = map(flamereading,0,1024,0,2);
29
30   //set a flame status
31
32   switch (flame) {
33   case 0:
34     flame_status = "No Fire";
35     Serial.println("Flame Status : "+flame_status);
36     break;
```

On the right, the simulation window shows the status of the system components:

Flame Status : No Fire
Gas Status : Gas leakage Detected
Sprinkler Status : not working
Exhaust fan Status : Working

Below this, a separator line is shown, followed by the updated status:

Flame Status : Fire is Detected
Gas Status : Gas leakage Detected
Sprinkler Status : working
Exhaust fan Status : Working

CODE:

```
#include <time.h>

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);
}

void loop() {

    //setting a random seed

    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);

    //set a flame status

    switch (flame) {
    case 0:
```

```
        flame_status = "No
Fire";
        Serial.println("Flame
Status : "+flame_status);
        break;
    case 1:
        flame_status = "Fire is
Detected";
        Serial.println("Flame
Status : "+flame_status);
        break;
    }

    //Gas Detection

    if(gas > 100){
        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(flame){
        sprinkler_status =
"working";
        Serial.println("Sprinkler
Status : "+sprinkler_status);
    }
    else{
        sprinkler_status = "not
working";
        Serial.println("Sprinkler
Status : "+sprinkler_status);
    }

    //toggle the fan according
to gas
```

```

    if(gas > 100){
        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(2000);

}

```

TEST CASES:

S.NO	INPUT	OUTPUT	RESULT
1	Gas:62 Temperature:45.30 Flame:366	Exhaust Fan: Not Working Sprinkler: Not Working Status Logged: Done	PASSED
2	Gas:598 Temperature:51.40 Flame:412	Exhaust Fan: Working Sprinkler: Not Working Status Logged: Done	PASSED

3	Gas:334 Temperature:49.30 Flame:912	Exhaust Fan: Working Sprinkler: Working Status Logged: Done	PASSED
4	Gas:18 Temperature:67.90 Flame:745	Exhaust Fan: Not Working Sprinkler: Working Status Logged: Done	PASSED
5	Gas: 354 Temperature:69.30 Flame:446	Exhaust Fan: Working Sprinkler: Not Working Status Logged: Done	PASSED