

SPRINT 2

Team ID : PNT2022TMID29941

IBM ID : IBM-Project-31889-1660205917

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(3000);
```

```
}
```

Diagram.json:

```
{  
  "version": 1,  
  "author": "Divya Selvakumar",  
  "editor": "wokwi",  
  "connections": []  
}
```

Simulation Output:

The screenshot displays the Wokwi IDE interface. On the left, the code editor shows a C++ program named 'sprint2.ino'. The code includes a header file, initializes variables for exhaust fan, sprinkler, temperature, gas, and flame, and contains a loop that generates random values for these variables and prints their status. The right side of the interface shows the 'Simulation' tab with a timeline and a console output area. The output displays the status of the system at three different points in time, separated by dashed lines. The first two outputs show 'Fire is Detected' and 'Gas leakage Detected', with the sprinkler working and the exhaust fan working. The third output shows 'No Fire At Now', with the sprinkler not working and the exhaust fan still working.

```
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   //setting a random seed  
20  
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 At Now";  
35       Serial.println("Flame Status : "+flame_status);  
36       break;  
37     case 1:  
38       flame_status = "Fire is Detected";  
39
```

Simulation

00:10.666 97%

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

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

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