

SPRINT-2

TEAM ID	PNT2022TMID10980
PROJECT NAME	INDUSTRY - SPECIFIC INTELLIGENT FIRE MANAGEMENT SYSTEM
IBM ID	IBM-Project-6098-1658823524

CODE:

```
#include<time.h>
boolexhaust_fan_on = false;
boolsprinkler_on = false;
float temperature = 0;
int gas = 0;
int flame = 0;
Stringflame_status = "";
Stringaccident_status = "";
Stringsprinkler_status = "";
voidsetup() {
  Serial.begin(99900);
}
voidloop() {
  //setting a random seed
  srand(time(0));
  //initial variable
  temperature = random(-20,125);
  gas = random(0,1000);
  intflamereading = random(200,1024);
  flame = map(flamereading,0,1024,0,2);
  //set a flame status
  switch (flame) {
  case0:
```

```
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);
}

```

WOKWI SIMLUATION OUTPUT:

The screenshot displays the Wokwi simulation environment. On the left, the 'sketch.ino' file is open, showing a C++ program that simulates a fire detection system. The code includes variables for exhaust fan status, sprinkler status, temperature, gas, and flame, along with functions for setup and loop. The loop function uses random numbers to simulate sensor readings and a switch statement to determine the flame status based on the flame variable.

On the right, the 'Simulation' window shows the output of the program. It displays the status of the system at three different points in time, separated by dashed lines. The first output shows 'No Fire' detected. The second and third outputs show 'Fire is Detected'.

Simulation Output:

```

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

-----*****-----

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

```