

PROJECT DEVELOPMENT PHASE

| | |
|---------------|--|
| Date | 17 November 2022 |
| Team ID | PNT2022TMID42644 |
| Project Title | INDUSTRY – SPECIFIC INTELLIGENT FIRE MANAGEMENT SYSTEM |

SPRINT 1 : (Simulation Creaction using Wokwi and Connecting IBM Cloud using Python Code) :

Simulation Creaction using Wokwi :

Code :

```
#include <time.h>
```

```
#include <WiFi.h>
```

```
#include <PubSubClient.h>
```

```
bool exhaust_fan_on = false;
```

```
bool sprinkler_on = false;
```

```
float temperature = 0;
```

```
int gas_level = 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_level = random(0,1000);  
int flamereading =  
random(200,1024);  
flame =  
map(flamereading,0,1024,0,  
2);  
//set a flame status  
Serial.print("Temperature : ");  
Serial.println(temperature);  
  
Serial.print("Gas_level : ");  
Serial.println(gas_level);  
  
Serial.print("Flame : ");  
Serial.println(flame);  
  
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_level > 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 =
"Sprinkler ON";
Serial.println("Sprinkler Status : "+sprinkler_status);
}
else{
sprinkler_status = "Sprinkler OFF";
```

```
Serial.println("Sprinkler Status : "+sprinkler_status);
}
//toggle the fan according to gas
if(gas_level > 100){
  exhaust_fan_on = true;
  Serial.println("Exhaust fan Status : Fan ON");
}
else{
  exhaust_fan_on = false;
  Serial.println("Exhaust fan Status : Fan OFF");
}
Serial.println("");
Serial.println("");
Serial.println(" -----#####-----
----");
Serial.println("");
Serial.println("");
delay(1000);
}
```

Simulation Output :

WOKwi

SAVE

SHARE

Docs

sketch.ino • diagram.json • libraries.txt • Library Manager ▾

```
1  #include <time.h>
2  #include <WiFi.h>
3  #include <PubSubClient.h>
4
5  bool exhaust_fan_on = false;
6  bool sprinkler_on = false;
7  float temperature = 0;
8  int gas_level = 0;
9  int flame = 0;
10 String flame_status = "";
11 String accident_status = "";
12 String sprinkler_status = "";
13 void setup() {
14   Serial.begin(999000);
15 }
16 void loop() {
17   //setting a random seed
18   srand(time(0));
19   //initial variable
20   temperature = random(-
21   20,125);
22   gas_level = random(0,1000);
23   int flamereading =
24   random(200,1024);
25   flame =
26   map(flamereading,0,1024,0,
27   2);
28   //set a flame status
29   Serial.print("Temperature : ");
```

Simulation

00:36.986 79%

↺

■

⏸

Temperature : 46.00
Gas_level : 670
Flame : 1
Flame Status : Fire is Detected
Gas Status : Gas leakage Detected
Sprinkler Status : Sprinkler ON
Exhaust fan Status : Fan ON

-----#####-----

Temperature : 2.00
Gas_level : 670
Flame : 0
Flame Status : No Fire
Gas Status : Gas leakage Detected
Sprinkler Status : Sprinkler OFF
Exhaust fan Status : Fan ON



sketch.ino

diagram.json

libraries.txt

Library Manager

Simulation

00:41.783 79%

```
1 #include <time.h>
2 #include <WiFi.h>
3 #include <PubSubClient.h>
4
5 bool exhaust_fan_on = false;
6 bool sprinkler_on = false;
7 float temperature = 0;
8 int gas_level = 0;
9 int flame = 0;
10 String flame_status = "";
11 String accident_status = "";
12 String sprinkler_status = "";
13 void setup() {
14   Serial.begin(99900);
15 }
16 void loop() {
17   //setting a random seed
18   srand(time(0));
19   //initial variable
20   temperature = random(-
21     20,125);
22   gas_level = random(0,1000);
23   int flamereading =
24     random(200,1024);
25   flame =
26     map(flamereading,0,1024,0,
27     2);
28   //set a flame status
29   Serial.print("Temperature : ");
```



Temperature : 18.00

Gas_level : 63

Flame : 0

Flame Status : No Fire

Gas Status : No Gas leakage Detected

Sprinkler Status : Sprinkler OFF

Exhaust fan Status : Fan OFF

-----#####-----

Temperature : 68.00

Gas_level : 110

Flame : 1

Flame Status : Fire is Detected

Gas Status : Gas leakage Detected

Sprinkler Status : Sprinkler ON

Exhaust fan Status : Fan ON



sketch.ino

diagram.json

libraries.txt

Library Manager

Simulation

```
6 bool sprinkler_on = false;
7 float temperature = 0;
8 int gas_level = 0;
9 int flame = 0;
10 String flame_status = "";
11 String accident_status = "";
12 String sprinkler_status = "";
13 void setup() {
14   Serial.begin(999000);
15 }
16 void loop() {
17   //setting a random seed
18   srand(time(0));
19   //initial variable
20   temperature = random(-
21     20,125);
22   gas_level = random(0,200);
23   int flamereading =
24     random(200,1024);
25   flame =
26     map(flamereading,0,1024,0,
27     2);
28   //set a flame status
29   Serial.print("Temperature : ");
30   Serial.println(temperature);
31
32   Serial.print("Gas_level : ");
33   Serial.println(gas_level);
34
35   Serial.print("Flame : ");
```



00:19.872 79%

Temperature : 108.00

Gas_level : 51

Flame : 0

Flame Status : No Fire

Gas Status : No Gas leakage Detected

Sprinkler Status : Sprinkler OFF

Exhaust fan Status : Fan OFF

Temperature : 37.00

Gas_level : 85

Flame : 1

Flame Status : Fire is Detected

Gas Status : No Gas leakage Detected

Sprinkler Status : Sprinkler ON

Exhaust fan Status : Fan OFF

Connecting IBM Cloud using Python Code :

Code:

```
import wiotp.sdk.device
import time
import random
myConfig = {
    "identity": {
        "orgId": "0bm892",
        "typeId": "NodeMcu",
        "deviceId": "1234"
    },
    "auth": {
        "token": "12345678"
    }
}

def myCommandCallback(cmd):
    print("Message received from IBM IoT Platform: %s" %
cmd.data['command'])
    m=cmd.data['command']

client = wiotp.sdk.device.DeviceClient(config=myConfig,
logHandlers=None)
client.connect()

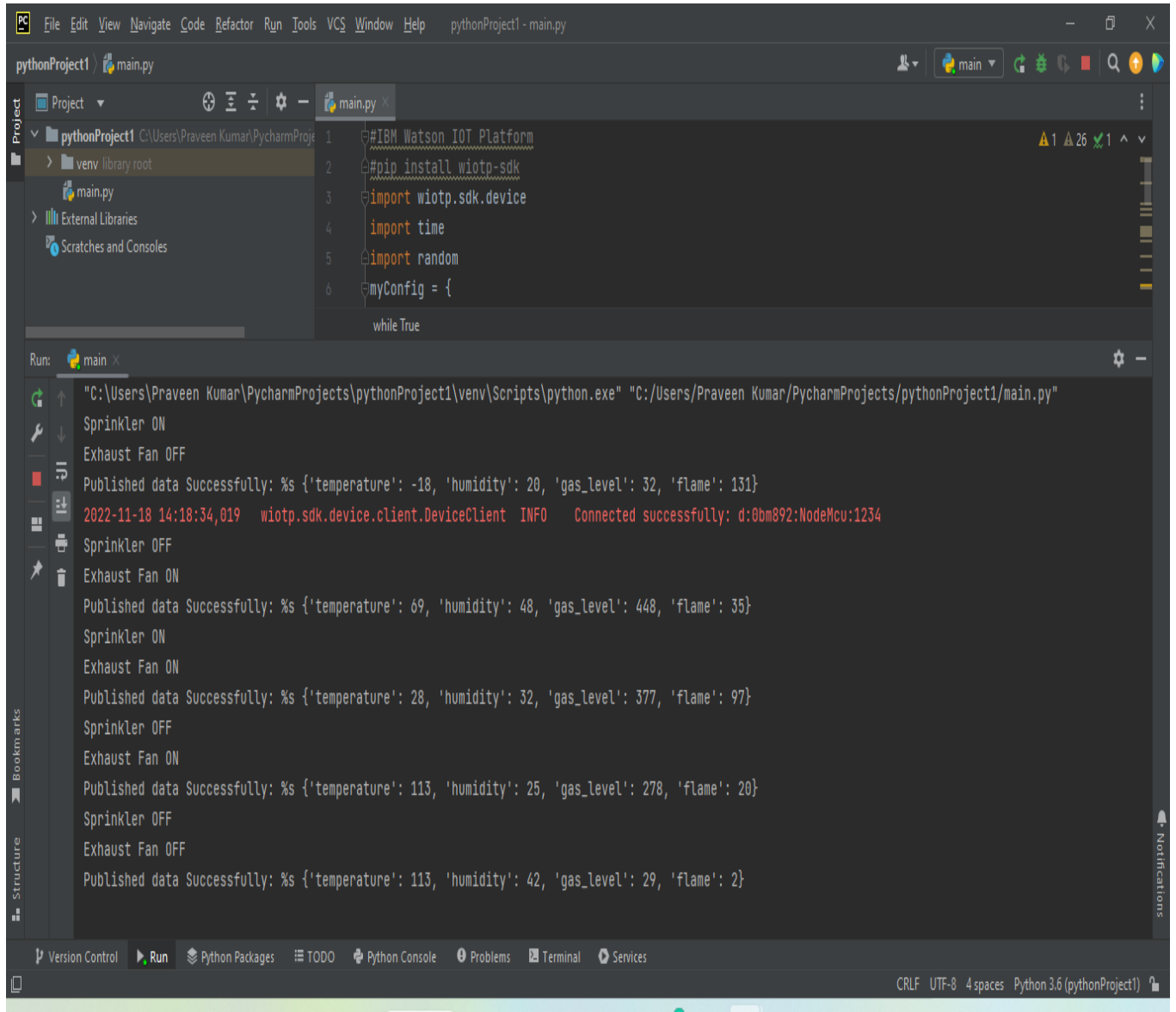
while True:
    temperature=random.randint(-20,125)
    humidity=random.randint(0,100)
    gas_level =random.randint(0,500)
    flame=random.randint(0,200)

    if (flame>80):
        print("Sprinkler ON")
    else:
        print("Sprinkler OFF")

    if (gas_level>100):
        print("Exhaust Fan ON")
    else:
        print("Exhaust Fan OFF")

    myData={'temperature':temperature, 'humidity':humidity,
'gas_level': gas_level,'flame':flame}
    client.publishEvent(eventId="status", msgFormat="json",
data=myData, qos=0, onPublish=None)
    print("Published data Successfully: %s", myData)
    client.commandCallback = myCommandCallback
    time.sleep(2)
client.disconnect()
```


OUTPUT :



The screenshot displays the PyCharm IDE interface. The top toolbar includes menus for File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The project name is 'pythonProject1 - main.py'. The left sidebar shows the Project view with 'pythonProject1' and its subfolders: 'venv', 'library', 'root', 'main.py', 'External Libraries', and 'Scratches and Consoles'. The main editor window shows the code for 'main.py'.

```
1 #IBM Watson IOT Platform
2 #pip install wiotp-sdk
3 import wiotp.sdk.device
4 import time
5 import random
6 myConfig = {
    while True
```

The bottom panel shows the Run output for 'main.py'. The command executed is: `"C:\Users\Praveen Kumar\PycharmProjects\pythonProject1\venv\Scripts\python.exe" "C:/Users/Praveen Kumar/PycharmProjects/pythonProject1/main.py"`. The output shows a sequence of actions: Sprinkler ON, Exhaust Fan OFF, Published data Successfully: %s {'temperature': -18, 'humidity': 20, 'gas_level': 32, 'flame': 131}, 2022-11-18 14:18:34,019 wiotp.sdk.device.client.DeviceClient INFO Connected successfully: d:0bm892:NodeMcu:1234, Sprinkler OFF, Exhaust Fan ON, Published data Successfully: %s {'temperature': 69, 'humidity': 48, 'gas_level': 448, 'flame': 35}, Sprinkler ON, Exhaust Fan ON, Published data Successfully: %s {'temperature': 28, 'humidity': 32, 'gas_level': 377, 'flame': 97}, Sprinkler OFF, Exhaust Fan ON, Published data Successfully: %s {'temperature': 113, 'humidity': 25, 'gas_level': 278, 'flame': 20}, Sprinkler OFF, Exhaust Fan OFF, Published data Successfully: %s {'temperature': 113, 'humidity': 42, 'gas_level': 29, 'flame': 2}.

The bottom status bar shows: CRLF UTF-8 4 spaces Python 3.6 (pythonProject1).

OUTPUT IN IBM CLOUD :

The screenshot displays the IBM Watson IoT Platform interface. At the top, the header shows the platform name, a user profile with email 'praveenkumarraj111@gmail.com' and ID '0bm892', and navigation tabs: 'Browse', 'Action', 'Device Types', and 'Interfaces'. A blue 'Add Device +' button is on the right. The main content area shows details for a device with ID '1234', status 'Connected', type 'NodeMcu', and a timestamp '13 Nov 2022 14:23'. Below this, a tabbed interface includes 'Identity', 'Device Information', 'Recent Events' (selected), 'State', and 'Logs'. A message states: 'The recent events listed show the live stream of data that is coming and going from this device.' A table follows with columns: 'Event', 'Value', 'Format', and 'Last Received'. It lists five 'status' events with JSON payloads and a 'a few seconds ago' timestamp. A bottom notification bar indicates '1 Simulation running'.

IBM Watson IoT Platform

praveenkumarraj111@gmail.com
ID: 0bm892

Browse Action Device Types Interfaces

Add Device +

1234 Connected NodeMcu Device 13 Nov 2022 14:23

Identity Device Information Recent Events State Logs

The recent events listed show the live stream of data that is coming and going from this device.

| Event | Value | Format | Last Received |
|--------|---|--------|-------------------|
| status | {"temperature":120,"humidity":8,"gas_level":35,... | json | a few seconds ago |
| status | {"temperature":-6,"humidity":69,"gas_level":485... | json | a few seconds ago |
| status | {"temperature":113,"humidity":4,"gas_level":21... | json | a few seconds ago |
| status | {"temperature":4,"humidity":3,"gas_level":199,"f... | json | a few seconds ago |
| status | {"temperature":13,"humidity":7,"gas_level":448... | json | a few seconds ago |

1 Simulation running