

SPRINT-4

TEAM ID	PNT2022TMID18163
PROJECT TITLE	Industry specific intelligent fire management system

PYTHON PROGRAM:

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
```

[#Provide](#) your IBM Watson Device Credentials

```
organization = "a6n32x"
deviceType = "Mainproject"
deviceId = "ibmproject"
authMethod = "token"
authToken = "1234567890"
```

[# Initialize GPIO](#)

```
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="lighton":
        print ("led is on")
    else :
        print ("led is off")
```

[#print](#)(cmd)

```
try:
    deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method": authMethod,
                    "auth-token": authToken}
    deviceCli = ibmiotf.device.Client(deviceOptions)
    #.....
```

```

except Exception as e:
    print("Caught exception connecting device: %s" % str(e))
    sys.exit()

# Connect and send a datapoint "hello" with value "world" into the cloud as an event of type "greeting"
10 times
deviceCli.connect()

while True:
    #Get Sensor Data from DHT22,DHT11,

    Temp=random.randint(-20,120)
    Humidity=random.randint(0,120)
    Flame=random.randint(0,100)
    Gas=random.randint(0,80)

    data = {'Temp' :Temp , 'Humidity' : Humidity, 'Flame' : Flame, 'Gas' : Gas}

    def myOnPublishCallback():
        if Flame > 100:
            data = {'Flame' : Flame}

    print ("Temperature =%s c" % Temp , "Humidity =%s u" % Humidity, "Flame =%s ir" % Flame , "Gas
    =%s ppm" % Gas )
    success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publish=myOnPublishCallback)

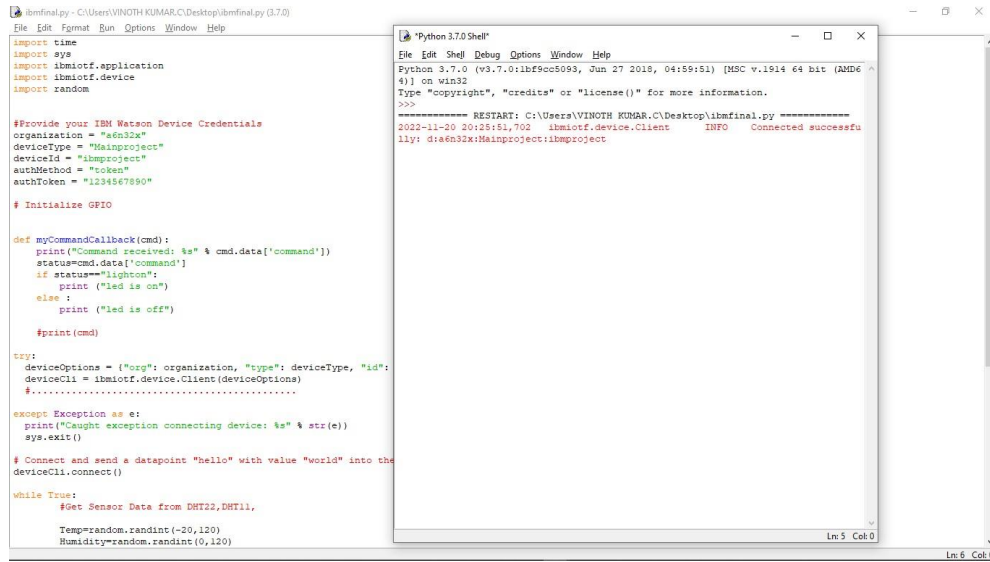
    if not success:
        print("Not connected to IoT")
        time.sleep(1)

    deviceCli.commandCallback = myCommandCallback

# Disconnect the device and application from the cloud
deviceCli.disconnect()

```

PYTHON CODE OUTPUT:



The image shows a Python script in a text editor and its execution output in a terminal window. The script is named `ibmfinal.py` and is located at `C:\Users\VINOTH KUMAR.C\Desktop\ibmfinal.py (3.7.0)`. It imports `time`, `sys`, `ibmiotf.application`, `ibmiotf.device`, and `random`. It defines a `myCommandCallback` function that prints the command received and its status. The script then initializes a device with the following options: `organization = "a6n32x"`, `deviceType = "Mainproject"`, `deviceId = "ibmpjproject"`, `authMethod = "token"`, and `authToken = "1234567890"`. It connects to the device and sends a datapoint "hello" with value "world". Finally, it enters a loop that gets sensor data from DHT22 and DHT11 sensors and prints it.

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random

#Provide your IBM Watson Device Credentials
organization = "a6n32x"
deviceType = "Mainproject"
deviceId = "ibmpjproject"
authMethod = "token"
authToken = "1234567890"

# Initialize GPIO

def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="lighton":
        print ("led is on")
    else :
        print ("led is off")
    #print(cmd)

try:
    deviceOptions = {"org": organization, "type": deviceType, "id":
deviceCli = ibmiotf.device.Client(deviceOptions)
#.....
except Exception as e:
    print("Caught exception connecting device: %s" % str(e))
    sys.exit()

# Connect and send a datapoint "hello" with value "world" into the
deviceCli.connect()

while True:
    #Get Sensor Data from DHT22,DHT11,
    Temp=random.randint(-20,120)
    Humidity=random.randint(0,120)
```

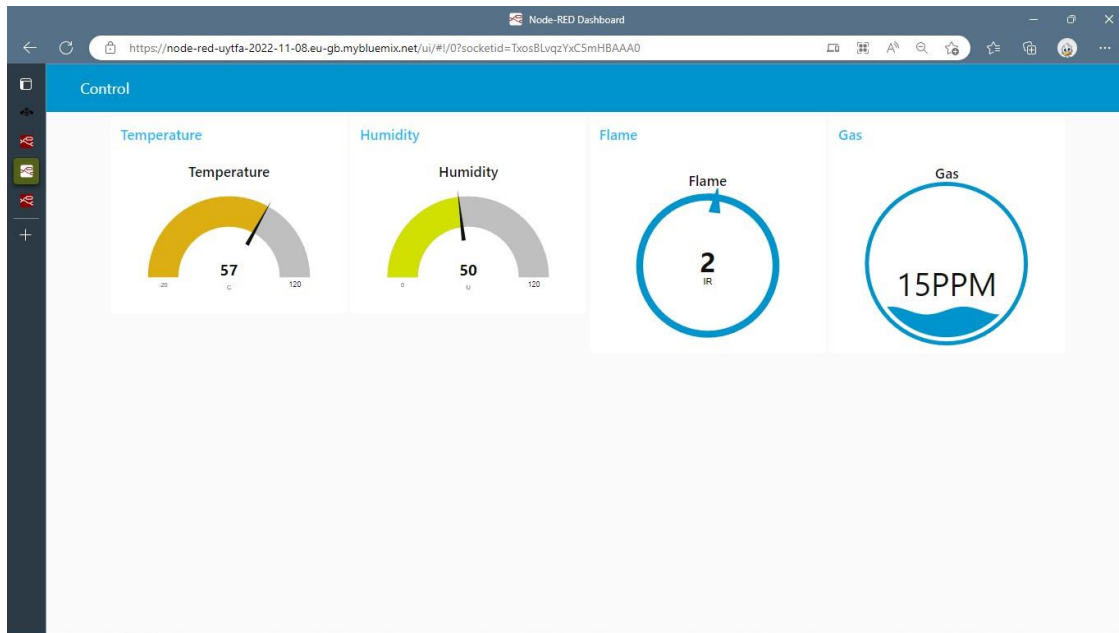
The terminal window shows the output of the script. It starts with a restart message, followed by the connection status: `2022-11-20 20:25:51,702 ibmiotf.device.Client INFO Connected successfully: d:a6n32x:Mainproject:ibmpjproject`.

IBM WATSON OUTPUT:

Browse Action Device Types Interfaces Add Device +				
Event	Value	Format	Last Received	
Data	{"Temp":-40,"Humidity":0}	json	a few seconds ago	
Data	{"Temp":-40,"Humidity":0}	json	a few seconds ago	
Data	{"Temp":-40,"Humidity":0}	json	a few seconds ago	
Data	{"Temp":-40,"Humidity":0}	json	a few seconds ago	
Data	{"Temp":-40,"Humidity":0}	json	a few seconds ago	

Items per page 50 | 1–2 of 2 items1 of 1 page<1>>

NODERED UI OUTPUT:



NODE RED SENSOR READING:

