

## Project Development Phase

### Delivery of Sprint 3

Date	12 November 2022
Team ID	PNT2022TMID28798
Project Name	Project –Gas leakage monitoring and alerting system for industries
Marks	20 marks

**Code:** import time import

sys import

ibmiotf.application import

ibmiotf.device import

random

#Provide your IBM Watson Device Credentials

organization = "6a4pz2" deviceType =

"Node\_1" deviceId = "12345" authMethod =

"use-token-auth" authToken = "12345678"

# Initialize GPIO

def myCommandCallback(cmd): print("Command

received: %s" % cmd.data['command'])

status=cmd.data['command'] if status=="alarmon":

print ("Alarm is on") elif (status == "alarmoff") :

print ("Alarm is off")

elif status == "sprinkleron":

print("Sprinkler is OFF") elif

status == "sprinkleron":

print("Sprinkler is ON")

```
#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 DHT11
```

```
    temp=random.randint(0,100)
```

```
    Humid=random.randint(0,100)    gas=random.randint(0,100)
```

```
    data = { 'temp' : temp, 'Humid': Humid, 'gas' : gas }
```

```
    #print data
```

```
    def myOnPublishCallback():
```

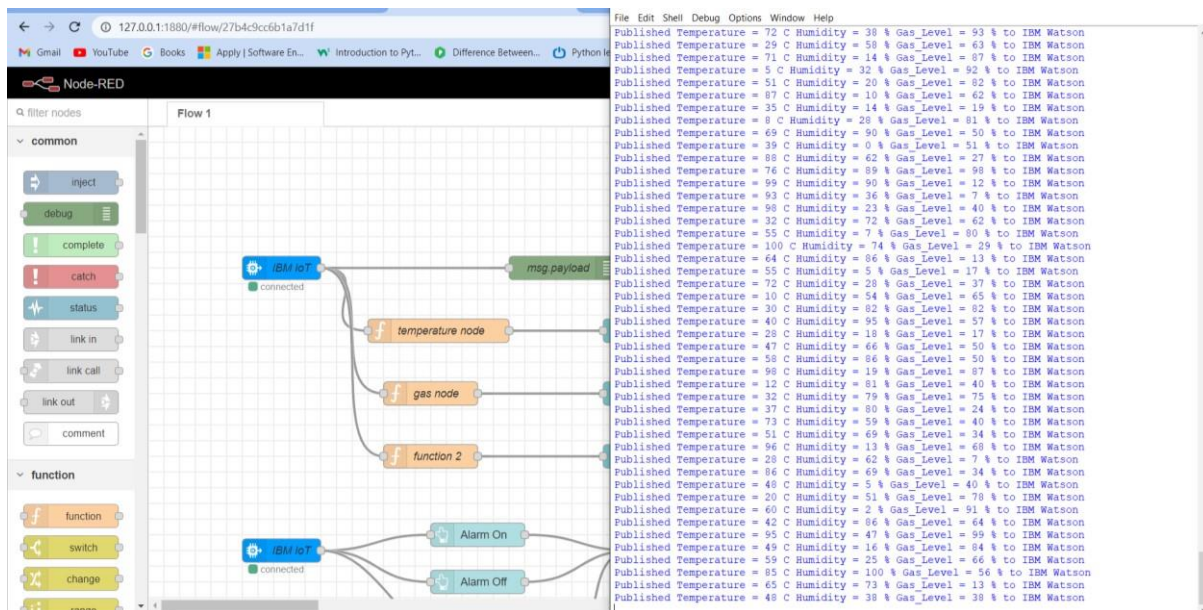
```
        print ("Published Temperature = %s C" % temp, "Humidity = %s %" % Humid, "Gas_Level = %s  
%" % gas, "to IBM Watson")
```

```
        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()
```



```

File Edit Format Run Options Window Help
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random

#Provide your IBM Watson Device Credentials
organization = "q4n8ny"
deviceType = "PNT2022TMD47483"
deviceId = "PNT2022TMD47483DVICEID"
authMethod = "token"
authToken = "0vZoxRf@LrhADWKjbl"

# Initialize GPIO

def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="alarmon":
        print ("Alarm is on")
    elif (status == "alarmoff"):
        print ("Alarm is off")
    elif status == "sprinkleron":
        print("Sprinkler is OFF")
    elif status == "sprinkleroff":
        print("Sprinkler is ON")
    #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
deviceCli.connect()

while True:
    #Get Sensor Data from DHT11

```

The code is a Python script that connects to an IBM Watson IoT device and sends sensor data. It includes a command callback function and a main loop that connects to the device and sends data.