

DELIVERY OF SPRINT-1

Team ID	PNT2022TMID53803
Team Members	Subash – 714019106110 Wasim Ansari – 714019106133 Yuvan Shankar -714019106136 Siddiq Bala - 714019106104
Project Title	Gas Leakage Monitoring And Alerting System For Industries

The goal of creating this prototype was to revolutionize environmental safety by eliminating any major or minor hazards brought on by the release of hazardous and dangerous gases into the environment. We created a Gas Leakage Detector for society using IOT technology, and it has the ability to perform data analytics on sensors and Smart Alerting techniques that send text messages to the relevant authorities. Using gas sensors, this system will be able to identify any gases present in the surrounding area. This will shield us from the main detrimental issue.

Thus the data has to be published to the IBM cloud. In the python script, the values for the gas, temperature, humidity and fire have been generated and published to IBM cloud platform. This is achieved by importing the required libraries in the python script and also specifying the organization, deviceType, deviceid, authMethod and authToken to integrate with the specific cloud account, so that the data will be published to IBM cloud platform. A threshold value has been fixed for each module and if any value exceeds this threshold value, then an alert message has been generated.

PYTHON CODE:

```
import ibmiotf.application
```

```
import ibmiotf.device
```

```
import time
```

```
import sys
```

```
#ibm watson device credentials
```

```
organization="griwxv"
```

```
deviceType="ESP32" deviceid="12345678"
```

```
authMethod="token"
```

```
authToken="12345678"
```

```
#generate random values for gas leakage
```

```
def myCommandCallback(cmd):
```

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

```
    print (cmd) try:
```

```
        deviceOptions={"org": organization,"type": deviceType,"id":  
deviceid,"authmethod":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 sending data for gas leakage
```

```
deviceCli.connect()
```

```
while True:
```

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

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

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

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

```
    data={'Gas':Gas,'Temp':Temp,'Hum':Hum,'Fire':Fire}
```

```
    print(data)
```

```
    def myOnPublishCallBack():
```

```
        print("published Gas  %s " %Gas)    print("published Temp %s " %Temp)    print("published  
Hum %s " %Hum)    print("published Fire %s " %Fire)
```

```
    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 from the cloud  
  
deviceCli.connect()
```

OUTPUT:

```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
{'Gas': 80, 'Temp': 89, 'Hum': 30, 'Fire': 44}
published Gas 80
published Temp 89
published Hum 30
published Fire 44
{'Gas': 54, 'Temp': 82, 'Hum': 89, 'Fire': 60}
published Gas 54
published Temp 82
published Hum 89
published Fire 60
{'Gas': 19, 'Temp': 50, 'Hum': 96, 'Fire': 8}
published Gas 19
published Temp 50
published Hum 96
published Fire 8
{'Gas': 47, 'Temp': 76, 'Hum': 14, 'Fire': 77}
published Gas 47
published Temp 76
published Hum 14
published Fire 77
{'Gas': 86, 'Temp': 89, 'Hum': 55, 'Fire': 63}
published Gas 86
published Temp 89
published Hum 55
published Fire 63
{'Gas': 68, 'Temp': 46, 'Hum': 54, 'Fire': 29}
published Gas 68
published Temp 46
published Hum 54
published Fire 29
Ln 18 Col 51
```