

SPRINT 4

Framework (Cloud deployment)

Date	12 November 2022
Team ID	PNT2022TMID27071
Project Name	Project - Gas Leakage Monitoring and Alerting System for Industries.

Cloud Deployment:

- On cloud, analyse and store the data and communicate wirelessly for further analysis is possible. Anyone can access the leakage data from anywhere using any Internet enabled device like PC, tablet or smart phone, and analyse it.
- The fire caused by gas leakage not only harms the owner but also people who are not far from the fire. From these problems, the authors make a design of cloud computing-based detection system of gas leak using a microcontroller NodeMCU Esp8266 that can provide notifications via smartphone in case of fire and automatically do the first treatment by turning on the exhaust. Notification sends via the smartphone appear not only when opening the application, but also when it does not open the application.

5.5 Receiving commands in IBM cloud using Python program:

#IBM Watson IOT Platform

#pip install wiotp-sdk

import wiotp.sdk.device

import time

import random

#Provide your IBM Watson Device Credentials

```
myConfig = {
    "identity": {
        "orgId": "yf0dyy",
        "typeId": "Kumaran",
        "deviceId": "12345"
    },
    "auth": {
        "token": "VJTDPRX@f&4Vuox8ms"
    }
}
```

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

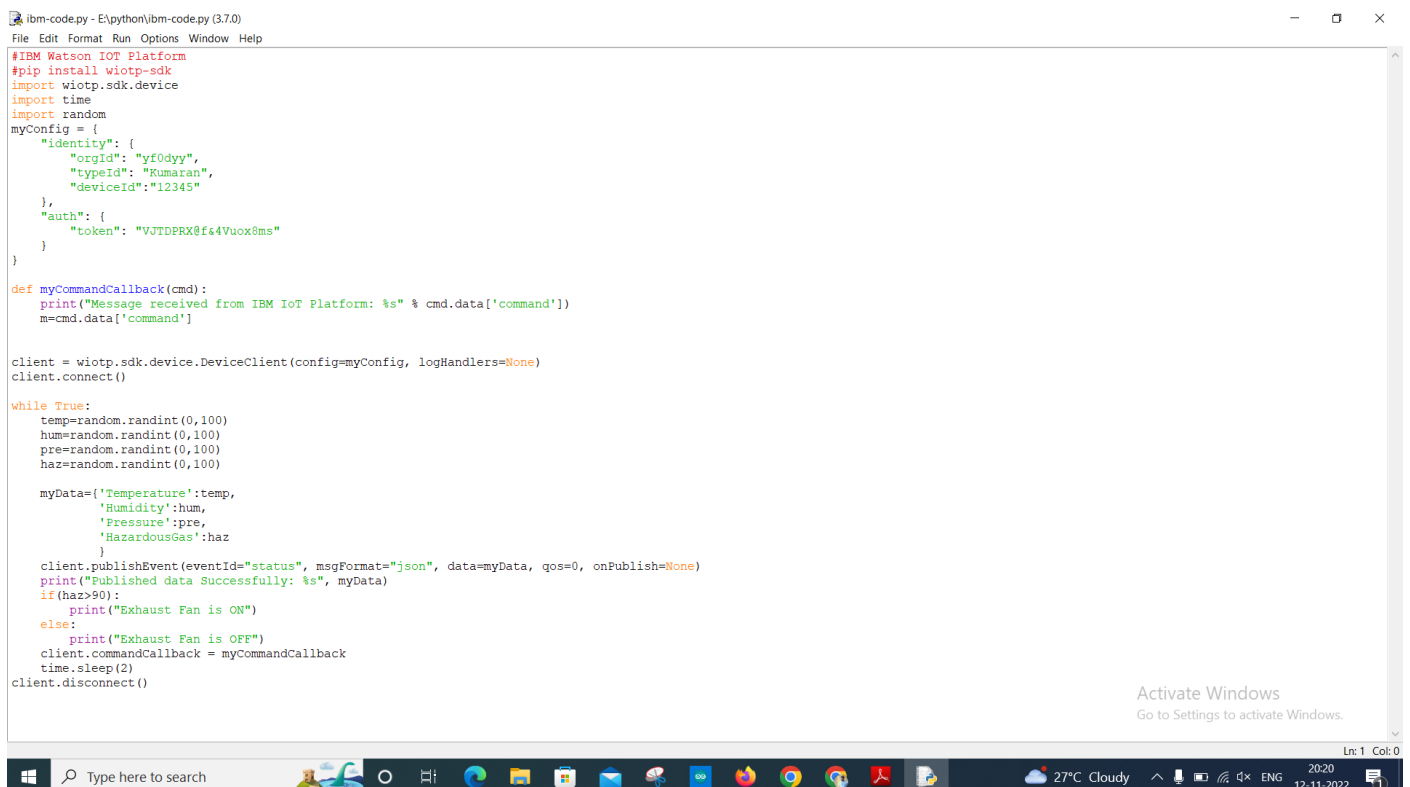
```
client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()
```

#Conditions

```
while True:
    temp=random.randint(0,100)
    hum=random.randint(0,100)
    pre=random.randint(0,100)
    haz=random.randint(0,100)

    myData={'Temperature':temp,
            'Humidity':hum,
            'Pressure':pre,
            'HazardousGas':haz
            }

    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
    print("Published data Successfully: %s", myData)
    if(haz>90):
        print("Exhaust Fan is ON")
    else:
        print("Exhaust Fan is OFF")
    client.commandCallback = myCommandCallback
    time.sleep(2)
client.disconnect()
```



```
ibm-code.py - E:\python\ibm-code.py (3.7.0)
File Edit Format Run Options Window Help
#IBM Watson IoT Platform
#pip install wiotp-sdk
import wiotp.sdk.device
import time
import random
myConfig = {
    "identity": {
        "orgId": "yf0dy",
        "typeId": "Rumaran",
        "deviceId": "12345"
    },
    "auth": {
        "token": "VJTDPRX@f64Vuox8ms"
    }
}

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:
    temp=random.randint(0,100)
    hum=random.randint(0,100)
    pre=random.randint(0,100)
    haz=random.randint(0,100)

    myData={'Temperature':temp,
            'Humidity':hum,
            'Pressure':pre,
            'HazardousGas':haz
            }

    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
    print("Published data Successfully: %s", myData)
    if(haz>90):
        print("Exhaust Fan is ON")
    else:
        print("Exhaust Fan is OFF")
    client.commandCallback = myCommandCallback
    time.sleep(2)
client.disconnect()
```

Activate Windows
Go to Settings to activate Windows.

Ln: 1 Col: 0

27°C Cloudy 2020 12-11-2022

```
*Python 3.7.0 Shell*
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: E:\python\libm-code.py =====
2022-11-12 20:20:55,781 wiotp.sdk.device.client.DeviceClient INFO Connected successfully: diyf0ddy:Kumaran:12345Published data Successfully: %s
{'Temperature': 35, 'Humidity': 76, 'Pressure': 37, 'HazardousGas': 44}
Exhaust Fan is OFF
Published data Successfully: %s {'Temperature': 62, 'Humidity': 65, 'Pressure': 77, 'HazardousGas': 20}
Exhaust Fan is OFF
Published data Successfully: %s {'Temperature': 72, 'Humidity': 54, 'Pressure': 69, 'HazardousGas': 10}
Exhaust Fan is OFF
Published data Successfully: %s {'Temperature': 89, 'Humidity': 65, 'Pressure': 69, 'HazardousGas': 99}
Exhaust Fan is ON
```

Activate Windows
Go to Settings to activate Windows.

Ln: 5 Col: 0

6.Observations & Results:

```
*Python 3.7.0 Shell*
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: E:\python\libm-code.py =====
2022-11-12 20:20:55,781 wiotp.sdk.device.client.DeviceClient INFO Connected successfully: diyf0ddy:Kumaran:12345Published data Successfully: %s
{'Temperature': 35, 'Humidity': 76, 'Pressure': 37, 'HazardousGas': 44}
Exhaust Fan is OFF
Published data Successfully: %s {'Temperature': 62, 'Humidity': 65, 'Pressure': 77, 'HazardousGas': 20}
Exhaust Fan is OFF
Published data Successfully: %s {'Temperature': 72, 'Humidity': 54, 'Pressure': 69, 'HazardousGas': 10}
Exhaust Fan is OFF
Published data Successfully: %s {'Temperature': 89, 'Humidity': 65, 'Pressure': 69, 'HazardousGas': 99}
Exhaust Fan is ON
```

Activate Windows
Go to Settings to activate Windows.

Ln: 5 Col: 0

IBM Watson IoT Platform

?

310819106044@smartinternz.com

ID: yf0dy

Browse

Action

Device Types

Interfaces

Add Device +

Device ID

Status

Device Type

Class ID

Date Added

Descriptive Location

▼

■

12345

Disconnected

Kumaran

Device

Oct 31, 2022 11:38 AM

→ ...

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

event_1

{"Hazardous Gas":61,"Temperature":88,"Humidit...

json

a few seconds ago

event_1

{"Hazardous Gas":20,"Temperature":36,"Humidit...

json

a few seconds ago

event_1

{"Hazardous Gas":79,"Temperature":56,"Humidit...

json

a few seconds ago

event_1

{"Hazardous Gas":52,"Temperature":82,"Humidit...

json

a few seconds ago

event_1

{"Hazardous Gas":26,"Temperature":33,"Humidit...

json

a few seconds ago

1 Simulation running

Activate Windows

Go to Settings to activate Windows.

The screenshot displays the IBM Watson IoT Platform dashboard. At the top, the header shows the user ID '310819106044@smartinternz.com' and the email 'ID: yf0dy'. The main dashboard area is divided into several sections:

- My Board:** A section with a back arrow and a heart icon. It contains a line chart for 'Hazardous Gas' with a pink background and a red line. The chart shows a peak at 20:42:30 and a dip at 20:43. Below the chart is a '1 minute' dropdown and a 'now' button.
- Humidity:** A card showing 'Humidity' with a blue circle icon and '99.0 gallon'.
- Value:** A green card showing '52.0 %' for 'Pressure'. Below the value is a table with two columns: 'Time' and 'Value'.
- Gauge:** A purple card showing a gauge for '48.0 °C'.

Time	Value
10/31/2022, 8:43:03 PM	52.0 %
10/31/2022, 8:43:00 PM	25.0 %
10/31/2022, 8:42:57 PM	14.0 %
10/31/2022, 8:42:54 PM	78.0 %
10/31/2022, 8:42:51 PM	69.0 %

Node-RED interface showing a flow for monitoring environmental data. The flow starts with an **IBM IoT** node (connected), which feeds into four function nodes: **Hazardous Gas**, **Temperature**, **Humidity**, and **Pressure**. These function nodes are connected to a **switch** node, which then feeds into an **http request** node. The output of the **http request** node is connected to a **msg.payload** node. The **msg.payload** node is also connected to four output nodes: **Hazardous Gas**, **Temperature**, **Humidity**, and **Pressure**. The **debug** console on the right shows the following log entries:

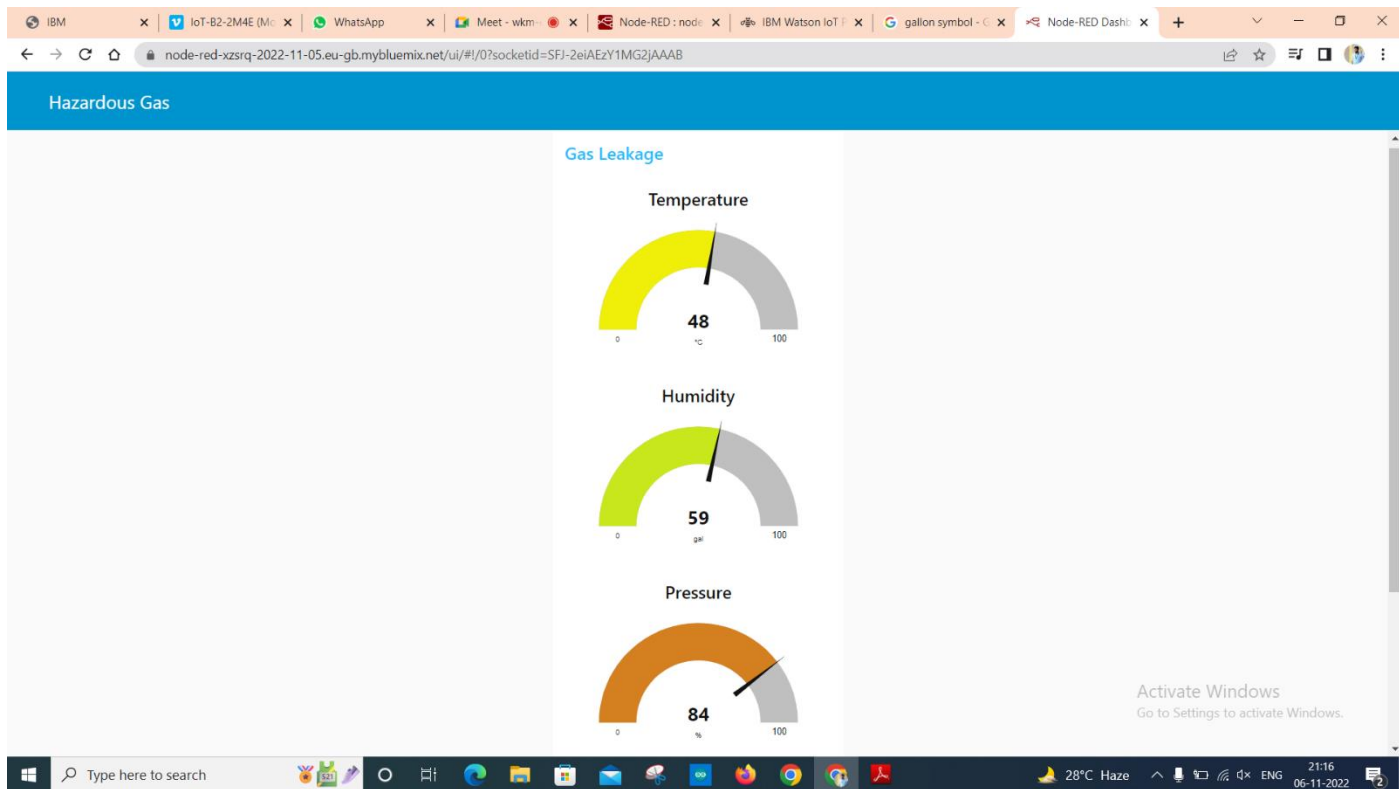
```
msg.payload : number
73
11/12/2022, 8:27:48 PM node: b0ec530feac71d47
iot-2/type/KumaranId/12345/evl/status/fmt/json :
msg.payload : Object
{ Temperature: 30, Humidity: 71,
  Pressure: 45, HazardousGas: 86 }
11/12/2022, 8:27:48 PM node: b0ec530feac71d47
iot-2/type/KumaranId/12345/evl/status/fmt/json :
msg.payload : number
30
11/12/2022, 8:27:48 PM node: b0ec530feac71d47
iot-2/type/KumaranId/12345/evl/status/fmt/json :
msg.payload : number
71
11/12/2022, 8:27:48 PM node: b0ec530feac71d47
iot-2/type/KumaranId/12345/evl/status/fmt/json :
msg.payload : number
45
11/12/2022, 8:27:48 PM node: b0ec530feac71d47
iot-2/type/KumaranId/12345/evl/status/fmt/json :
msg.payload : number
86
11/12/2022, 8:27:48 PM node: b0ec530feac71d47
iot-2/type/KumaranId/12345/evl/status/fmt/json :
msg.payload : string(22)54
11/12/2022, 8:27:48 PM node: b0ec530feac71d47
iot-2/type/KumaranId/12345/evl/status/fmt/json :
msg.payload : string(22)54
```

Node-RED Dashboard interface showing three gauges for monitoring environmental data. The gauges are labeled **Humidity**, **Pressure**, and **Hazardous Gas**. The values displayed on the gauges are 61, 90, and 72 respectively. The dashboard also includes an **Activate Windows** watermark.

Humidity: 61 gal

Pressure: 90 %

Hazardous Gas: 72 ppm



7. Advantages & Disadvantages:

Advantages:

- Get real-time alerts about the gaseous presence in the atmosphere.
- Prevent fire hazards and explosions.
- Supervise gas concentration levels.
- Ensure worker's health.
- Real-time updates about leakage.
- Cost-effective installation.
- Data analytics for improved decisions.
- Measure oxygen level accuracy.
- Measure oxygen level accuracy.

Disadvantages:

- Only one gas can be measured with each instrument.
- Poor stability leads to greater environmental impact.
- When heavy dust, steam or fog blocks the input of the sensor.

8. Conclusion and Future Work:

In this paper we use IOT technology for enhancing the existing safety standards. While making this prototype has been to bring a revolution in the field of safety against the leakage of harmful and toxic gases in environment and hence nullify any major or minor hazard being caused due to them. We have used the IOT technology to make a Gas Leakage Detector for society which having Smart Alerting techniques involving sending text message to the concerned authority and an ability performing data analytics on sensor. This system will be able to detect the gas in environment using the gas sensors. This will prevent form the major harmful problem.

9. References:

IBM cloud reference: <https://cloud.ibm.com/>

IoT simulator: <https://watson-iot-sensor-simulator.mybluemix.net/>

Fast 2 SMS: <https://www.fast2sms.com/>