

## SPRINT - 4

<b>Team ID</b>	<b>PNT2022TMID17768</b>
<b>Project Title</b>	<b>Gas Leakage Monitoring And Alerting System</b>
<b>Date</b>	<b>15.11.2022</b>

### PYTHON CODE EXECUTION:

```
code.py - C:\Users\bala\AppData\Local\Programs\Python\Python36-32\code.py (3.6.0)
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 = "29xrcm"
deviceType = "ESP32"
deviceId = "1234"
authMethod = "token"
authToken = "12345678"

# Initialize GPIO

def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="sprinkleron":
        print ("Sprinkler is on")
    else :
        print ("Sprinkler 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 DHT11
    #.....
    #Send data to cloud as an event of type "greeting" 10 times
```

```
code.py - C:\Users\bala\AppData\Local\Programs\Python\Python36-32\code.py (3.6.0)
File Edit Format Run Options Window Help

#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)
    gasconcentration=random.randint(0,100)

    data = { 'temp' : temp, 'Humid': Humid, "gasconcentration": gasconcentration}

    #print data
    def myOnPublishCallback():
        print ("Published Temperature = %s C" % temp, "Humidity = %s %" % Humid, "gasconcentration = %s %" % gasconcentration, "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()

Ln: 57 Col: 0
```

```
code.py - C:\Users\bala\AppData\Local\Programs\Python\Python36-32\code.py (3.6.0)
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 = "s9xrcm"
deviceType = "ESP32"
deviceId = "1234"
authMethod = "token"
authToken = "12345678"

# Initialize GPIO

def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="sprinkleron":
        print ("Sprinkler is on")
    else :
        print ("Sprinkler 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
deviceCli.connect()

while True:
    #Get Sensor Data from DHT11

    temp=random.randint(0,100)
    Humid=random.randint(0,100)
    gasconcentration=random.randint(0,100)

    data = { 'temp' : temp, 'Humid': Humid, "gasconcentration": gasconcentration}

    #print data
    def myOnPublishCallback():
        print ("Published Temperature = %s C" % temp, "Humidity = %s %" % Humid, "gasconcentration = %s %" % gasconcentration, "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()

Ln: 57 Col: 0
```

```
*Python 3.6.0 Shell*
File Edit Shell Debug Options Window Help

Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 07:18:10) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:\Users\bala\AppData\Local\Programs\Python\Python36-32\code.py ==
2022-11-16 02:20:17,405 ibmiotf.device.Client INFO Connected successfully: d:s9xrcm:ESP32:1234
Published Temperature = 88 C Humidity = 59 % gasconcentration = 78 % to IBM Watson
Published Temperature = 28 C Humidity = 99 % gasconcentration = 87 % to IBM Watson
Published Temperature = 32 C Humidity = 60 % gasconcentration = 8 % to IBM Watson
Published Temperature = 41 C Humidity = 54 % gasconcentration = 67 % to IBM Watson
Published Temperature = 81 C Humidity = 64 % gasconcentration = 17 % to IBM Watson
Published Temperature = 51 C Humidity = 93 % gasconcentration = 38 % to IBM Watson
Published Temperature = 5 C Humidity = 1 % gasconcentration = 79 % to IBM Watson
Published Temperature = 44 C Humidity = 88 % gasconcentration = 69 % to IBM Watson
Published Temperature = 76 C Humidity = 54 % gasconcentration = 27 % to IBM Watson
Published Temperature = 37 C Humidity = 78 % gasconcentration = 10 % to IBM Watson

Ln: 5 Col: 0
```

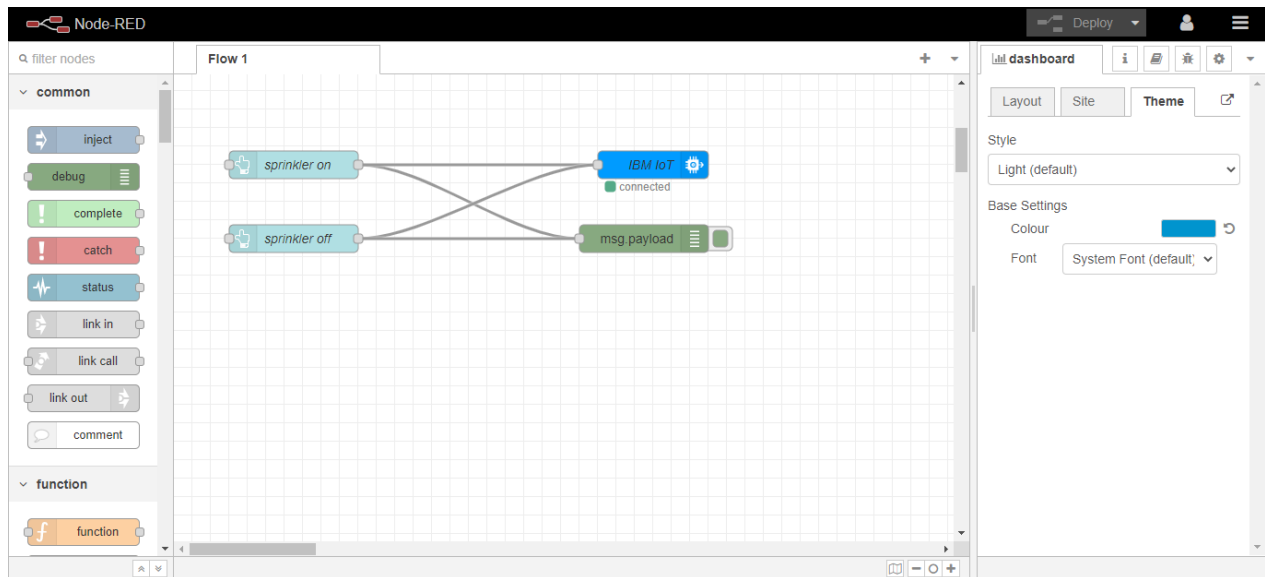
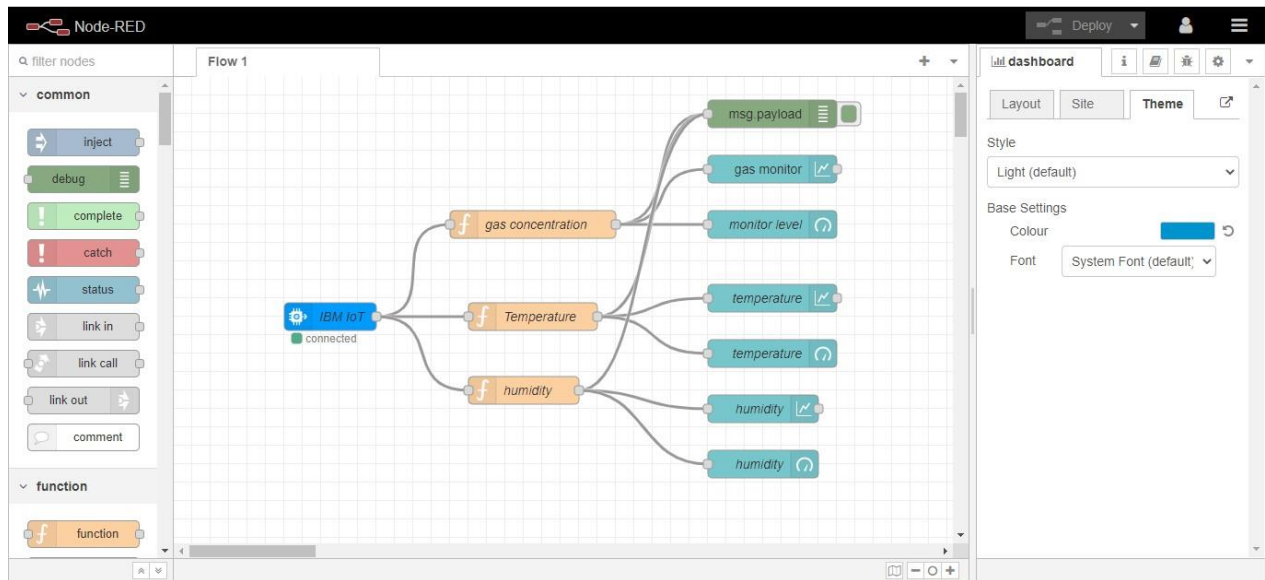
## Recent Events in IBM WATSON IOT Platform :

The screenshot displays the IBM Watson IoT Platform interface. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. A sidebar on the left contains icons for various functions. The main content area is titled 'Recent Events' and shows a table of live data streams. The table has four columns: 'Event', 'Value', 'Format', and 'Last Received'. Below the table, a status bar indicates '2 Simulations running'.

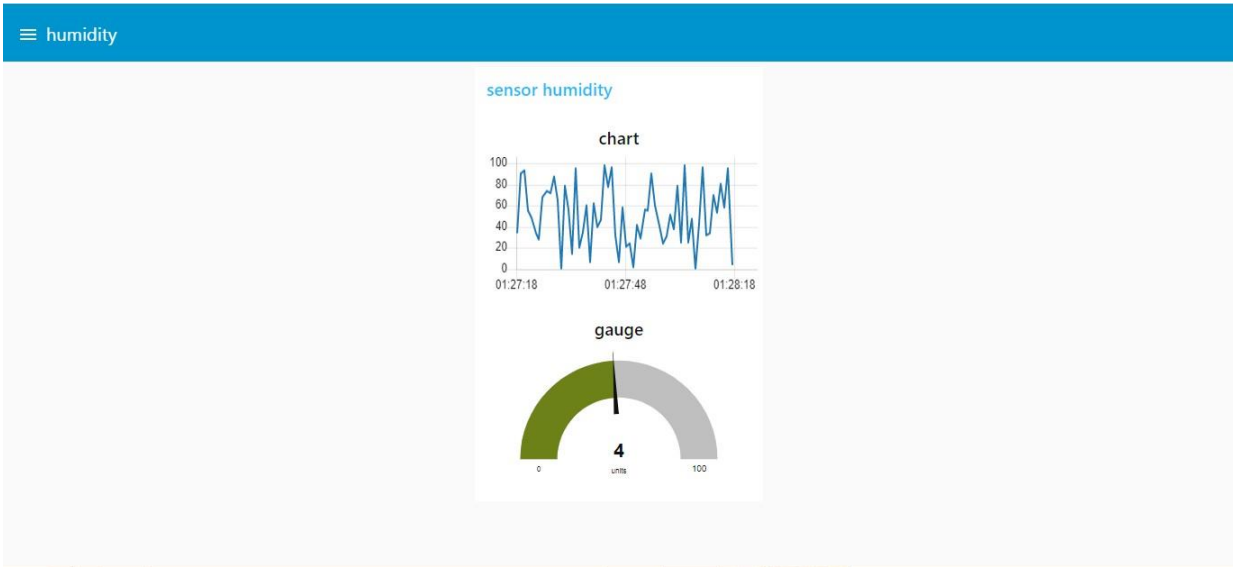
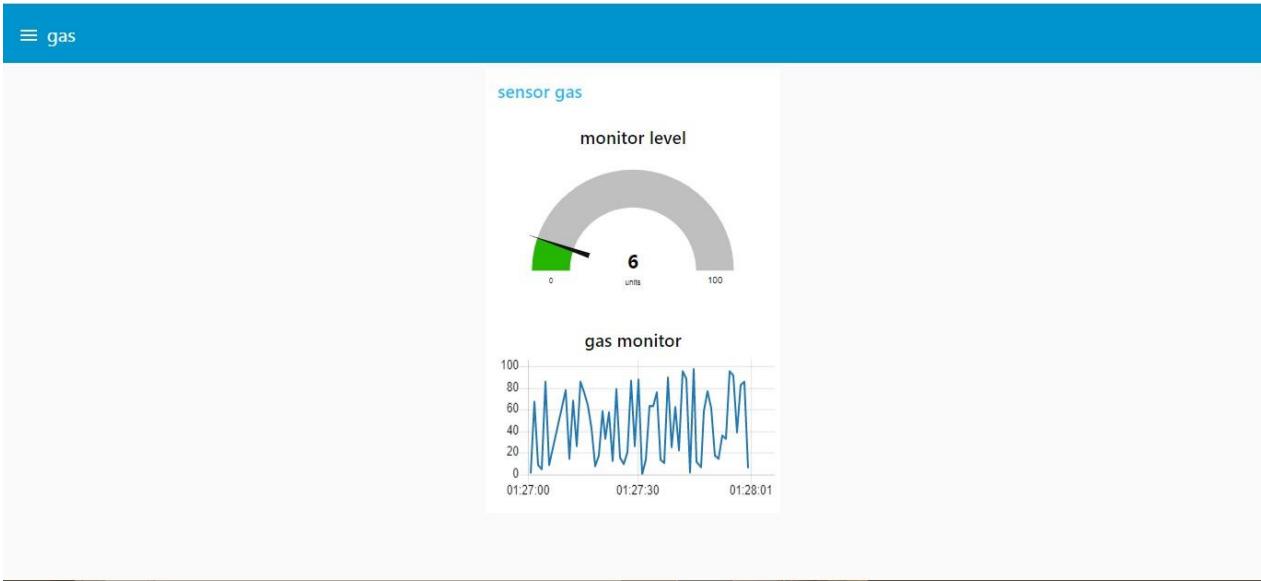
Event	Value	Format	Last Received
IoTSensor	{"temp":17,"Humid":97,"gasconcentration":12}	json	a few seconds ago
IoTSensor	{"temp":61,"Humid":49,"gasconcentration":48}	json	a few seconds ago
IoTSensor	{"temp":91,"Humid":49,"gasconcentration":77}	json	a few seconds ago
IoTSensor	{"temp":51,"Humid":79,"gasconcentration":43}	json	a few seconds ago
IoTSensor	{"temp":52,"Humid":52,"gasconcentration":57}	json	a few seconds ago

The screenshot displays the IBM Watson IoT Platform interface with a line chart titled 'gas leakage'. The chart shows three data series: 'temp' (blue), 'Humid' (orange), and 'gasconcentration' (green). The y-axis ranges from 0 to 100. The x-axis shows a time range from 01:54 to 01:54:30. A legend at the bottom identifies the series. A status bar at the bottom right indicates '2 Simulations running'.

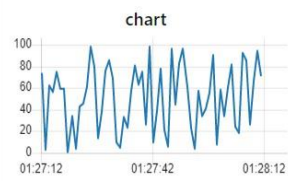
## Node Red Flow :



Dashboard Created Using Node :



sensor temperature



gauge



## Testing :

Switch Case

SPRINKLER ON

SPRINKLER

Node-RED

Deploy

filter nodes

Flow 1Flow 2

common

inject

debug

complete

catch

status

link in

link call

link out

comment

function

function

IBM IoT

connected

gas concentration

Temperature

humidity

msg.payload

gas monitor

monitor level

temperature

temperature

humidity

humidity

debug

all nodes

all

11/16/2022, 1:34:02 AM node:8ec7522603dfac16

msg.payload : Object

{ command: "switchon" }

11/16/2022, 1:34:03 AM node:8ec7522603dfac16

msg.payload : Object

{ command: "switchoff" }

11/16/2022, 1:34:04 AM node:8ec7522603dfac16

msg.payload : Object

{ command: "switchon" }

11/16/2022, 1:34:05 AM node:8ec7522603dfac16

msg.payload : Object

{ command: "switchoff" }

11/16/2022, 1:34:06 AM node:8ec7522603dfac16

msg.payload : Object

{ command: "switchon" }

11/16/2022, 1:34:08 AM node:8ec7522603dfac16

msg.payload : Object

{ command: "switchoff" }