

Project Development Phase

Delivery of Sprint 3

Date	12 November 2022
Team ID	PNT2022TMID41967
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 = "3bc6ow" deviceType =

"Gas_Geakage_Detector" deviceId = "6225"

authMethod = "Nodemcu" authToken =

"622519106053"

Initialize GPIO

def myCommandCallback(cmd): print("Command

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

status=cmd.data['command']

if status == "alarmon":

print ("Alarm is on please all Evacuate Fans On")

elif status == "alarmoff":

print ("Alarm is off and Fans Off")

elif status == "sprinkleron":

```

        print ("Sprinkler is On Evacuate Faster")
elif status == "sprinkleroff":
print("Sprinkler is Off")    else:
    print("Please send proper command")
# 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 random function

```

```

    temp=random.randint(0,120)
Humid=random.randint(0,100)    gas=random.randint(0,1500)

```

```

data={'temp':temp,'Humid':Humid,'gas':gas}    #print data

```

```

def myOnPublishCallback():

```

```

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

```

```
        success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,
on_publish=myOnPublishCallback)
```

```
    if not success:
```

```
        print("\n Not connected to IoTF")
```

```
if temp>60 :
```

```
    print("\n Fire Detected due to gas Leak ! Alarm ON! Sprinkler ON! Call The Fire Police \n")
```

```
elif gas>350:
```

```
    print("\n Gas is Leaking \n")
```

```
time.sleep(10)
```

```
deviceCli.commandCallback = myCommandCallback
```

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

The output is in the next page .

OUTPUT :



```
Published Temperature = 5 C Humidity = 56 % Gas_Level = 5 ppm to IBM Watson
Published Temperature = 55 C Humidity = 81 % Gas_Level = 318 ppm to IBM Watson

Fire Detected due to gas Leak ! Alarm ON! Sprinkler ON! Call The Fire Police
Published Temperature = 67 C
Humidity = 2 % Gas_Level = 1041 ppm to IBM Watson

Gas is Leaking
Published Temperature = 13 C
Humidity = 66 % Gas_Level = 784 ppm to IBM Watson

Gas is Leaking
Published Temperature = 5 C
Humidity = 52 % Gas_Level = 424 ppm to IBM Watson
Published Temperature = 57 C Humidity = 14 % Gas_Level = 24 ppm to IBM Watson
Published Temperature = 24 C Humidity = 19 % Gas_Level = 32 ppm to IBM Watson

Fire Detected due to gas Leak ! Alarm ON! Sprinkler ON! Call The Fire Police
Published Temperature = 81 C
Humidity = 55 % Gas_Level = 777 ppm to IBM Watson
Published Temperature = 33 C Humidity = 94 % Gas_Level = 146 ppm to IBM Watson

Fire Detected due to gas Leak ! Alarm ON! Sprinkler ON! Call The Fire Police
Published Temperature = 70 C
Humidity = 51 % Gas_Level = 1307 ppm to IBM Watson
Published Temperature = 19 C Humidity = 25 % Gas_Level = 63 ppm to IBM Watson

Fire Detected due to gas Leak ! Alarm ON! Sprinkler ON! Call The Fire Police
Published Temperature = 50 C
Humidity = 32 % Gas_Level = 1448 ppm to IBM Watson

Fire Detected due to gas Leak ! Alarm ON! Sprinkler ON! Call The Fire Police
Published Temperature = 65 C
Humidity = 96 % Gas_Level = 284 ppm to IBM Watson
```

