SPRINT-1

DATE	**10-11-2022**
TEAM ID	**PNT2022TMID27922**
PROJECT NAME	**Gas leakage monitoring and alerting system.**

PYTHON CODE:

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random

#Provide your IBM Watson Device Credentials
organization = "msi400"
deviceType = "Gasleak"
deviceId = "6068"
authMethod = "token"
authToken = "123456781"

# Initialize GPIO
def myCommandCallback(cmd):
   print("Command received: %s" % cmd.data['command'])
   status=cmd.data['command']
   if status=="lighton":
      print ("led is on")
   elif status == "lightoff":
      print ("led is off")
   else :
      print ("please send proper command")

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
        gas_level=random.randint(25,500)
        temp=random.randint(90,110)
        Humid=random.randint(60,100)

        data = {'Gas_level':gas_level, 'temp' : temp, 'Humid': Humid }
        #print data
        def myOnPublishCallback():
            print ("Toxicity_of_the_gas=%s ppm"%gas_level,",Published_Temperature = %s C" % temp,",Humidity = %s %%" % Humid, "to IBM Watson")

        success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publish=myOnPublishCallback)
        if not success:
            print("Not connected to IoTF")
        time.sleep(10)

        deviceCli.commandCallback = myCommandCallback

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

OUTPUT: