DEVELOP THE PYTHON CODE

DATE	19-11-2022
TEAM ID	PNT2022TMID38668
PROJECT TITLE	Gas Leakage Monitoring And Alerting System For Industries

```
import time

import sys

import ibmiotf.application

import ibmiotf.device

import random

#Provide your IBM Watson Device Credentials

organization = "bd91hr"

deviceType = "android"

deviceId = "1902"

authMethod = "token"

authToken = "12345678"

# Initialize GPIO

    def mycommandCallback(cmd):

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

  status = cmd.data['command']

  if status == "NO LEAKAGE":

    print("OPEN PIPELINE")

  elif status == "LEAKAGE":

    print("CLOSE PIPELINE")

  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

          temp=random.randint(0,100)

    Humid=random.randint(0,100)

        Gas=random.randint(0,100)

          data = { 'temp' : temp, 'Humid': Humid, 'Gas':Gas }

     #print data

        def myOnPublishCallback():

            print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % Humid, "Gas Concentration = %s"%Gas ,"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()
```