Develop a python script

Date	17 September 2022
Team ID	PNT2022TMID4740
Project Name	Project - Industry-Specific Intelligent Fire Management System

Python script for generating the random sensor values - Temperature, Flame Level and Gas Level to the IBM Watson IoT Platform.

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


#Provide your IBM Watson Device Credentials

organization = "4aqwut"
deviceType = "12345678dt"
deviceId = "12345678did"
authMethod = "token"
authToken = "*PrtsGAO?B@_tTPEKT"


# Initialize GPIO

def myCommandCallback(cmd):
   print("Command received: %s" % cmd.data['command'])
   status=cmd.data['command']
   if status=="sprinkleron":
      print ("Sprinkler is on")
   elif status == "sprinkleroff":
      print ("Sprinkler is off")
   elif status == "exhaustfanon":
      print ("Exhaust Fan ON")
   elif status == "exhaustfanoff":
      print ("Exhaust Fan 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



       temp=random.randint(0,100)
       flame_level=random.randint(0,100)
       gas_level = random.randint(0,100)


       data = { 'Temperature' : temp, 'Flame_Level' : flame_level, 'Gas_Level' : gas_level }

       #print data

       def myOnPublishCallback():
           print ("Published Temperature = %s C" % temp, "Flame_Level = %s %%" % flame_level,
"Gas_Level = %s %%" %gas_level ,"to IBM Watson")

       success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,
on_publish=myOnPublishCallback)
       if not success:
           print("Not connected to IoTF")
       time.sleep(1)

       deviceCli.commandCallback = myCommandCallback


# Disconnect the device and application from the cloud

deviceCli.disconnect()
```

Output :

```
Published Temperature = 3 C Flame_Level = 88 % Gas_Level = 30 % to IBM Watson
Published Temperature = 22 C Flame_Level = 51 % Gas_Level = 16 % to IBM Watson
Published Temperature = 80 C Flame_Level = 32 % Gas_Level = 88 % to IBM Watson
Published Temperature = 98 C Flame_Level = 81 % Gas_Level = 34 % to IBM Watson
Command received: sprinkleroff
Sprinkler is off
Command received: exhaustfanoff
Exhaust Fan OFF
Command received: sprinkleron
Sprinkler is on
Published Temperature = 93 C Flame_Level = 77 % Gas_Level = 43 % to IBM Watson
Command received: exhaustfanon
Exhaust Fan ON
Published Temperature = 18 C Flame_Level = 37 % Gas_Level = 88 % to IBM Watson
Published Temperature = 61 C Flame_Level = 53 % Gas_Level = 65 % to IBM Watson
Published Temperature = 95 C Flame_Level = 76 % Gas_Level = 90 % to IBM Watson
Published Temperature = 56 C Flame_Level = 14 % Gas_Level = 27 % to IBM Watson
Published Temperature = 34 C Flame_Level = 33 % Gas_Level = 51 % to IBM Watson
Published Temperature = 9 C Flame_Level = 56 % Gas_Level = 80 % to IBM Watson
Published Temperature = 42 C Flame_Level = 51 % Gas_Level = 18 % to IBM Watson
```