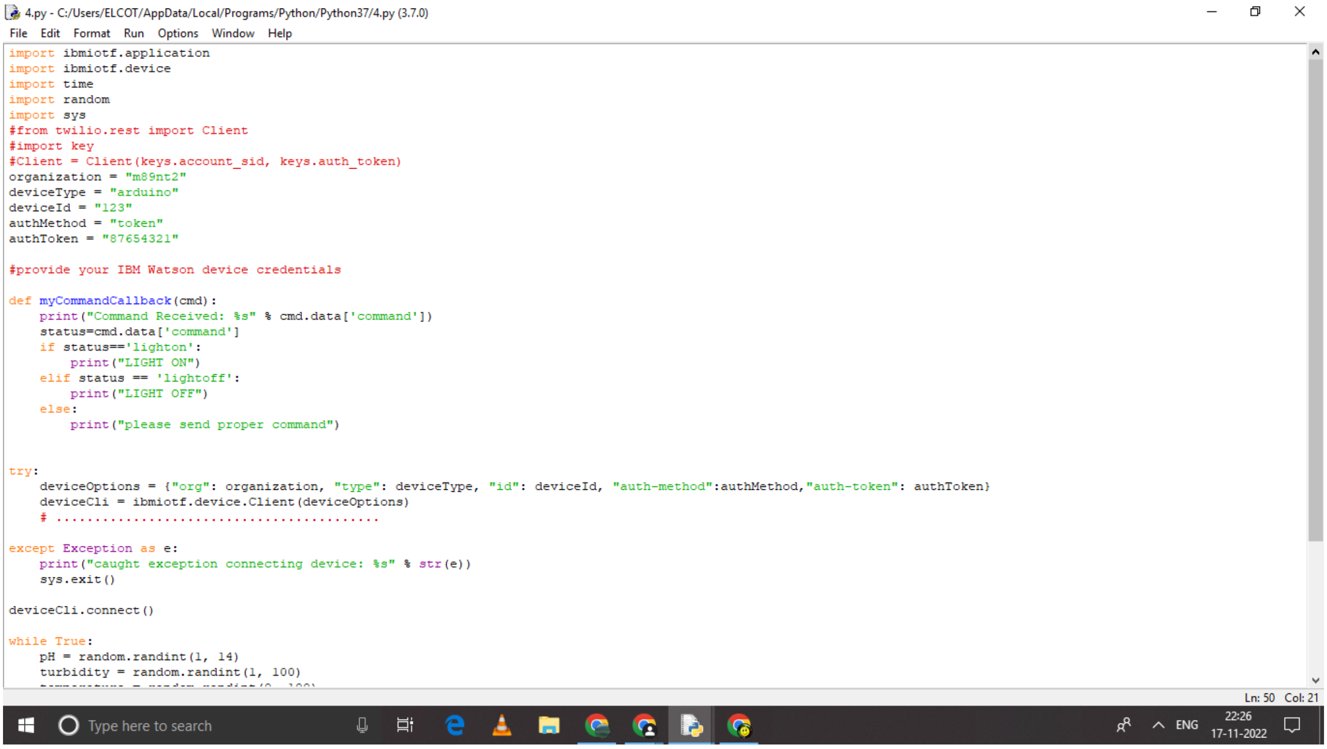
PYTHON SCRIPT TO MONITOR TEMPERATURE, PH, TURBIDITY IN RIVER WATER

PROGRAM:

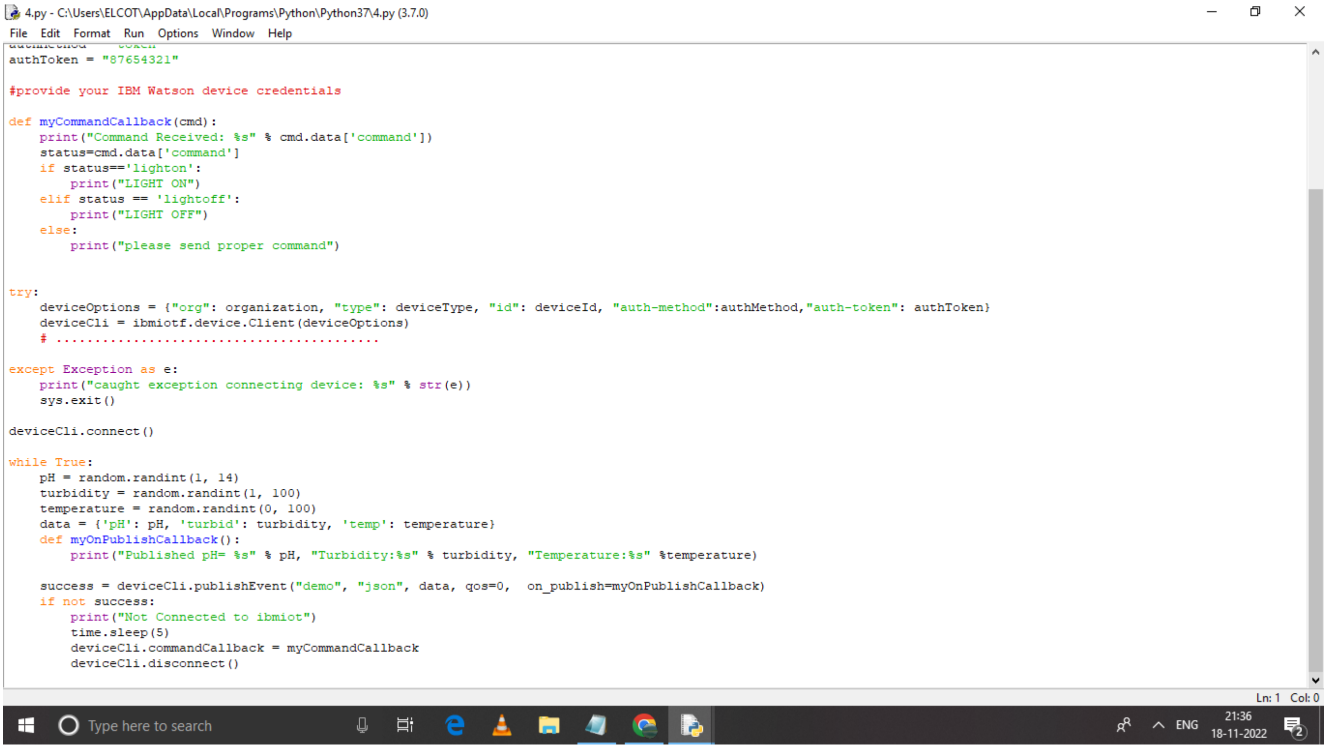
```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random

#provide Your IBM Watson Device Credentials
organization = "m89nt2"
deviceType = "arduino"
deviceID = "123"
authMethod = "token"
authToken ="87654321”



#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" info the cloud as an event of
type"greetings"10 times
deviceCli.connect()


while True:
    #Get sensor Data from DHT11


    temp=random.randint(90,110)
    pH=random.randint(0,14)
    turbidity=random.randint(0,100)


    data = { 'Temperature' : temp, 'pH': pH, 'Turbidity':turbidity }
    #print data
    def myOnPublishCallback():
      print ("published Temperature = %s C" % temp, "pH = is %s %%" % pH, "Turbidity= is
%s %%" % turbidity,"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

deviceCli.disconnect()

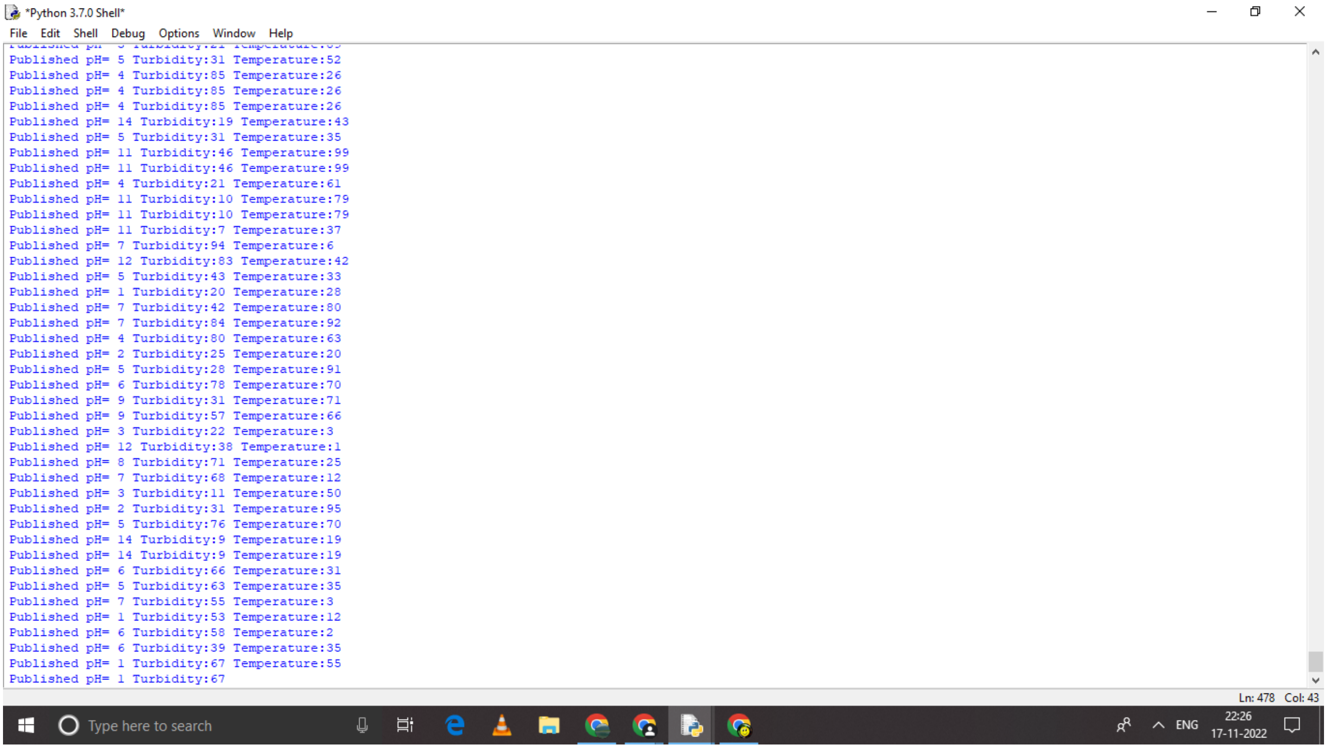

4.py - C:/Users/ELCOT/AppData/Local/Programs/Python/Python37/4.py (3.7.0)
File Edit Format Run Options Window Help
Ln: 50 Col: 21
Type here to search
ENG
22:26
17-11-2022

```
import ibmiotf.application
import ibmiotf.device
import time
import random
import sys
#from twilio.rest import Client
#import key
#Client = Client(keys.account_sid, keys.auth_token)
organization = "m89nt2"
deviceType = "arduino"
deviceId = "123"
authMethod = "token"
authToken = "87654321"

#provide your IBM Watson device credentials

def myCommandCallback(cmd):
    print("Command Received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=='lighton':
        print("LIGHT ON")
    elif status == 'lightoff':
        print("LIGHT 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()

deviceCli.connect()

while True:
    pH = random.randint(1, 14)
    turbidity = random.randint(1, 100)
```


4.py - C:\Users\ELCOT\AppData\Local\Programs\Python\Python37\4.py (3.7.0)
File Edit Format Run Options Window Help
Ln: 1 Col: 0
Type here to search
ENG
21:36
18-11-2022

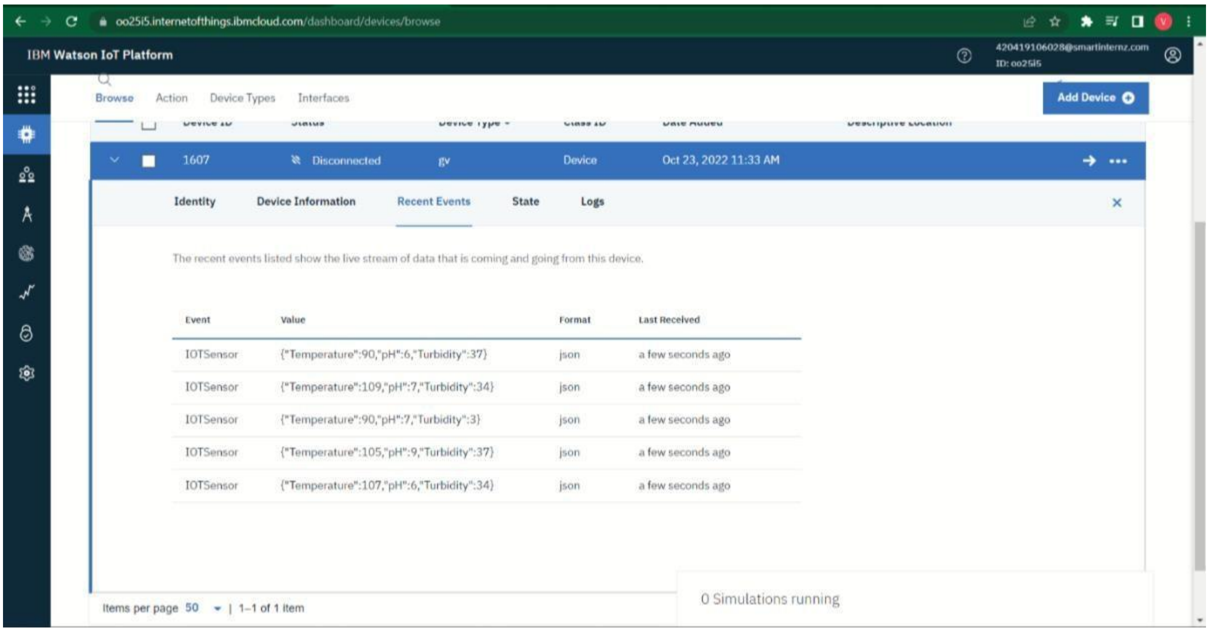
```
authToken = "87654321"

#provide your IBM Watson device credentials

def myCommandCallback(cmd):
    print("Command Received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=='lighton':
        print("LIGHT ON")
    elif status == 'lightoff':
        print("LIGHT 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()

deviceCli.connect()

while True:
    pH = random.randint(1, 14)
    turbidity = random.randint(1, 100)
    temperature = random.randint(0, 100)
    data = {'pH': pH, 'turbid': turbidity, 'temp': temperature}
    def myOnPublishCallback():
        print("Published pH= %s" % pH, "Turbidity:%s" % turbidity, "Temperature:%s" %temperature)

    success = deviceCli.publishEvent("demo", "json", data, qos=0,  on_publish=myOnPublishCallback)
    if not success:
        print("Not Connected to ibmiot")
        time.sleep(5)
        deviceCli.commandCallback = myCommandCallback
        deviceCli.disconnect()
```

OUTPUT:



```
Published pH= 5 Turbidity:31 Temperature:52
Published pH= 4 Turbidity:85 Temperature:26
Published pH= 4 Turbidity:85 Temperature:26
Published pH= 4 Turbidity:85 Temperature:26
Published pH= 14 Turbidity:19 Temperature:43
Published pH= 5 Turbidity:31 Temperature:35
Published pH= 11 Turbidity:46 Temperature:99
Published pH= 11 Turbidity:46 Temperature:99
Published pH= 4 Turbidity:21 Temperature:61
Published pH= 11 Turbidity:10 Temperature:79
Published pH= 11 Turbidity:10 Temperature:79
Published pH= 11 Turbidity:7 Temperature:37
Published pH= 7 Turbidity:94 Temperature:6
Published pH= 12 Turbidity:83 Temperature:42
Published pH= 5 Turbidity:43 Temperature:33
Published pH= 1 Turbidity:20 Temperature:28
Published pH= 7 Turbidity:42 Temperature:80
Published pH= 7 Turbidity:84 Temperature:92
Published pH= 4 Turbidity:80 Temperature:63
Published pH= 2 Turbidity:25 Temperature:20
Published pH= 5 Turbidity:28 Temperature:91
Published pH= 6 Turbidity:78 Temperature:70
Published pH= 9 Turbidity:31 Temperature:71
Published pH= 9 Turbidity:57 Temperature:66
Published pH= 3 Turbidity:22 Temperature:3
Published pH= 12 Turbidity:38 Temperature:1
Published pH= 8 Turbidity:71 Temperature:25
Published pH= 7 Turbidity:68 Temperature:12
Published pH= 3 Turbidity:11 Temperature:50
Published pH= 2 Turbidity:31 Temperature:95
Published pH= 5 Turbidity:76 Temperature:70
Published pH= 14 Turbidity:9 Temperature:19
Published pH= 14 Turbidity:9 Temperature:19
Published pH= 6 Turbidity:66 Temperature:31
Published pH= 5 Turbidity:63 Temperature:35
Published pH= 7 Turbidity:55 Temperature:3
Published pH= 1 Turbidity:53 Temperature:12
Published pH= 6 Turbidity:58 Temperature:2
Published pH= 6 Turbidity:39 Temperature:35
Published pH= 1 Turbidity:67 Temperature:55
Published pH= 1 Turbidity:67
```

PUBLISHING DATA TO IBM CLOUD:



The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
IOTSensor	{"Temperature":90,"pH":6,"Turbidity":37}	json	a few seconds ago
IOTSensor	{"Temperature":109,"pH":7,"Turbidity":34}	json	a few seconds ago
IOTSensor	{"Temperature":90,"pH":7,"Turbidity":3}	json	a few seconds ago
IOTSensor	{"Temperature":105,"pH":9,"Turbidity":37}	json	a few seconds ago
IOTSensor	{"Temperature":107,"pH":6,"Turbidity":34}	json	a few seconds ago