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

PROGRAM:

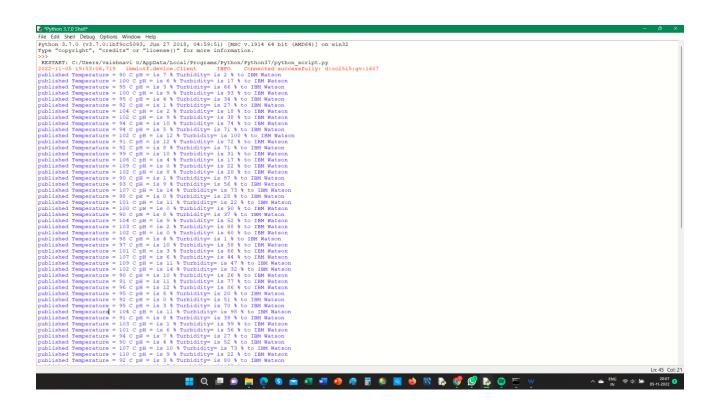
```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
#provide Your IBM Watson Device Credentials
organization = "oo25i5"
deviceType = "gv"
deviceID = "1607"
authMethod = "token"
authToken = "12341607"
#Initialize GPIO
def myCommandCallback(cmd):
  print ("command 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()

OUTPUT:



PUBLISHING DATA TO IBM CLOUD:

