

SPRINT_4

Team ID	PNT2022TMID32030
Project Name	SMART FARMER - IOT ENABLED SMART FARMINGAPPLICATION SYSTEM

Receiving commands from IBM cloud using Python :

```
import time

import sys

import ibmiotf.application

import ibmiotf.device

import random

#Provide your IBM Watson Device Credentials

organization = "xhlz7n"

deviceType = "iotsensor"

deviceId = "iotsensor"

authMethod = "token"

authToken = "2x_okYGd6qMY77S(S7"

# Initialize GPIO

def myCommandCallback(cmd):

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

    status=cmd.data['command']
```

```

if status=="motoron":
    print ("motor is on")
elif status == "motoroff":
    print ("motor 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

    temp=random.randint(90,110)

    Humid=random.randint(60,100)

    Mois=random. Randint(20,120)

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

#print data

def myOnPublishCallback():

    print ("Published Temperature = %s C" % temp, "Humidity = %s %" %
Humid, "Moisture =%s deg c" % Mois "to IBM Watson")

    success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,
on_publish=myOnPublishCallback)

```

if not success:

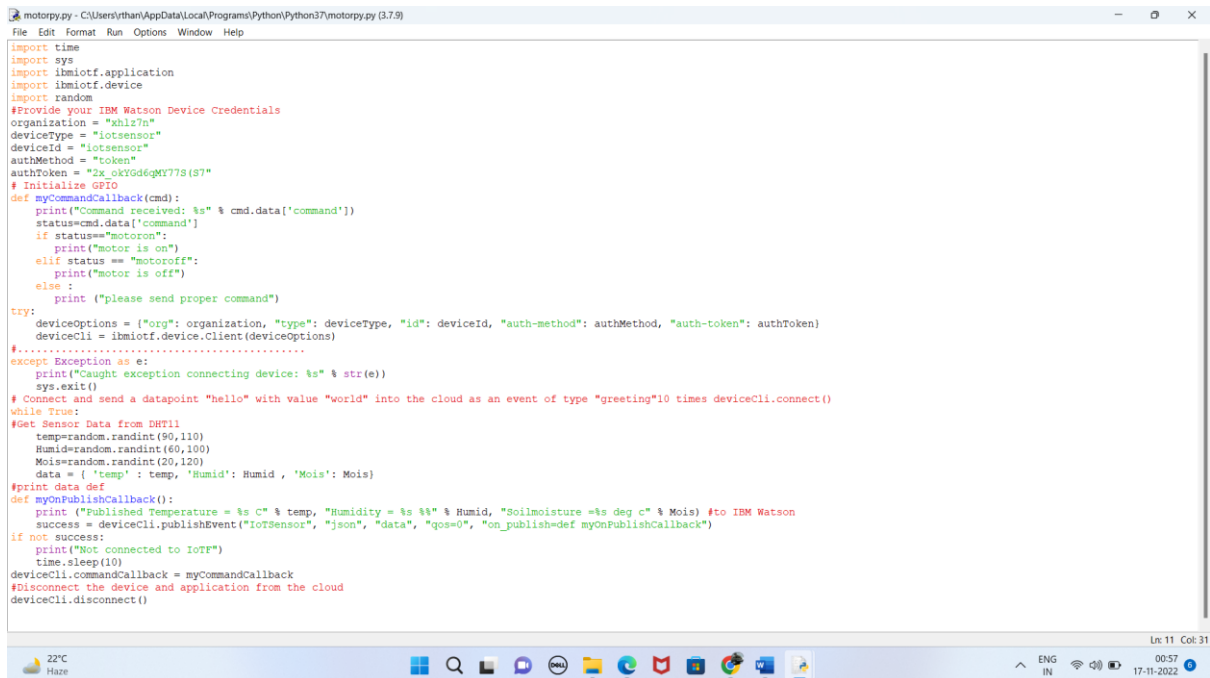
```
print("Not connected to IoTTF")
```

```
time.sleep(10)
```

```
deviceCli.commandCallback = myCommandCallback
```

Disconnect the device and application from the cloud

```
deviceCli.disconnect()
```



```
motorpy.py - C:\Users\ythan\AppData\Local\Programs\Python\Python37\motorpy.py (3.7.9)
File Edit Format Run Options Window Help

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

#Provide your IBM Watson Device Credentials
organization = "xh1z7n"
deviceType = "iotsensor"
deviceId = "iotsensor"
authMethod = "token"
authToken = "2k_okYGd6qMY778(S7)"

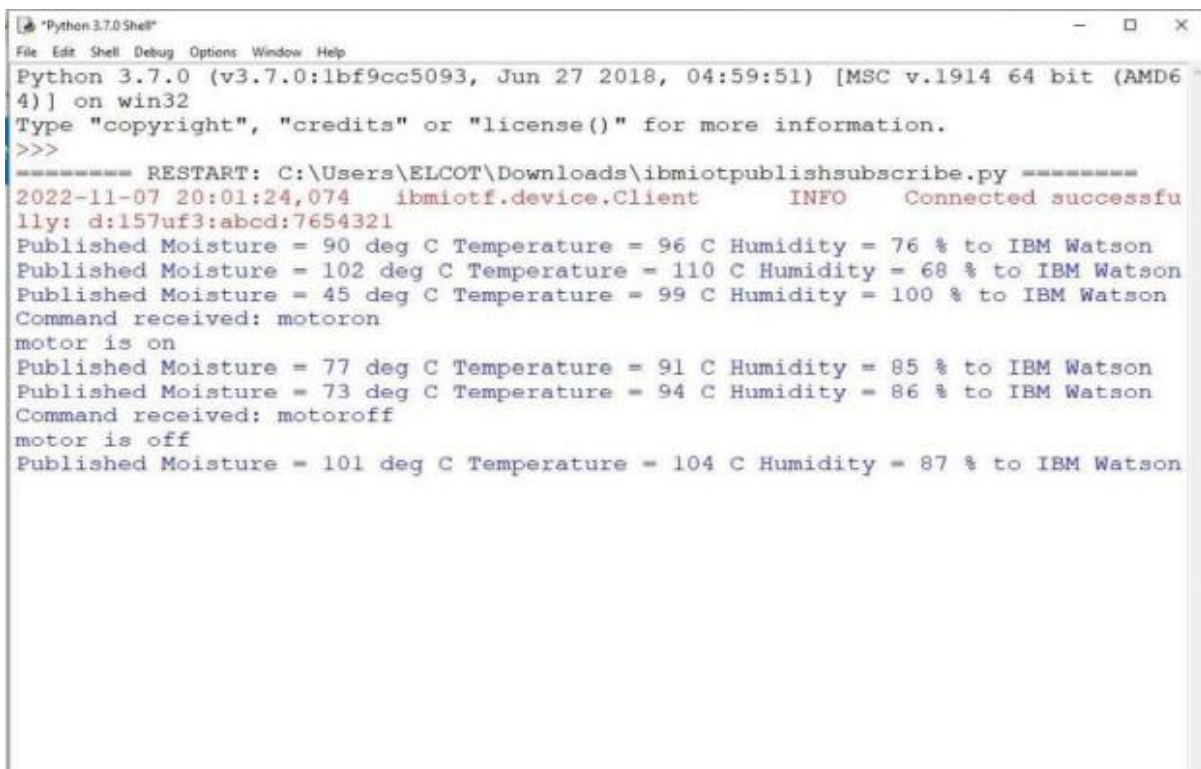
# Initialize GPIO
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="motoron":
        print("motor is on")
    elif status == "motorooff":
        print("motor is off")
    else :
        print ("please send proper command")

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
    temp=random.randint(90,110)
    Humid=random.randint(60,100)
    Mois=random.randint(20,120)
    data = { 'temp' : temp, 'Humid': Humid , 'Mois': Mois}
    #print data def
    def myOnPublishCallback():
        print ("Published Temperature = %s C" % temp, "Humidity = %s %" % Humid, "Soilmoisture =%s deg c" % Mois) #to IBM Watson
        success = deviceCli.publishEvent("IoTSensor", "json", "data", "qos=0", "on_publish=def myOnPublishCallback")
    if not success:
        print("Not connected to IoTTF")
        time.sleep(10)
    deviceCli.commandCallback = myCommandCallback
    #Disconnect the device and application from the cloud
    deviceCli.disconnect()
```

The python code to command the motor ON and OFF

OBSERVATION AND RESULT:



```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\ELCOT\Downloads\ibmiotpublishsubscribe.py =====
2022-11-07 20:01:24,074 ibmiotf.device.Client INFO Connected successfully: d:157uf3:abcd:7654321
Published Moisture = 90 deg C Temperature = 96 C Humidity = 76 % to IBM Watson
Published Moisture = 102 deg C Temperature = 110 C Humidity = 68 % to IBM Watson
Published Moisture = 45 deg C Temperature = 99 C Humidity = 100 % to IBM Watson
Command received: motoron
motor is on
Published Moisture = 77 deg C Temperature = 91 C Humidity = 85 % to IBM Watson
Published Moisture = 73 deg C Temperature = 94 C Humidity = 86 % to IBM Watson
Command received: motoroff
motor is off
Published Moisture = 101 deg C Temperature = 104 C Humidity = 87 % to IBM Watson
```

The running of motor depend on the sensor readings is done successfully done using node-red ,openweatherapi,mit app inventor , ibm cloud and watson