

Smart Farmer – IOT Enabled Smart Farming Application

DEVELOP A PYTHON SCRIPT TO PUBLISH AND SUBSCRIBE TO IBM IOT PLATFORM

Date	29 October 2022
Team ID	PNT2022TMID11578
Project Name	Project – Smart Farmer-IoT Enabled smart Farming Application

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

#Provide your IBM Watson Device Credentials
organization = "1jk4ps"
deviceType = "PNT2022TMID51719"
deviceId = "Smart_Farmer"
authMethod = "token"

authToken = "11*53hCIhmEbf!&Es&"

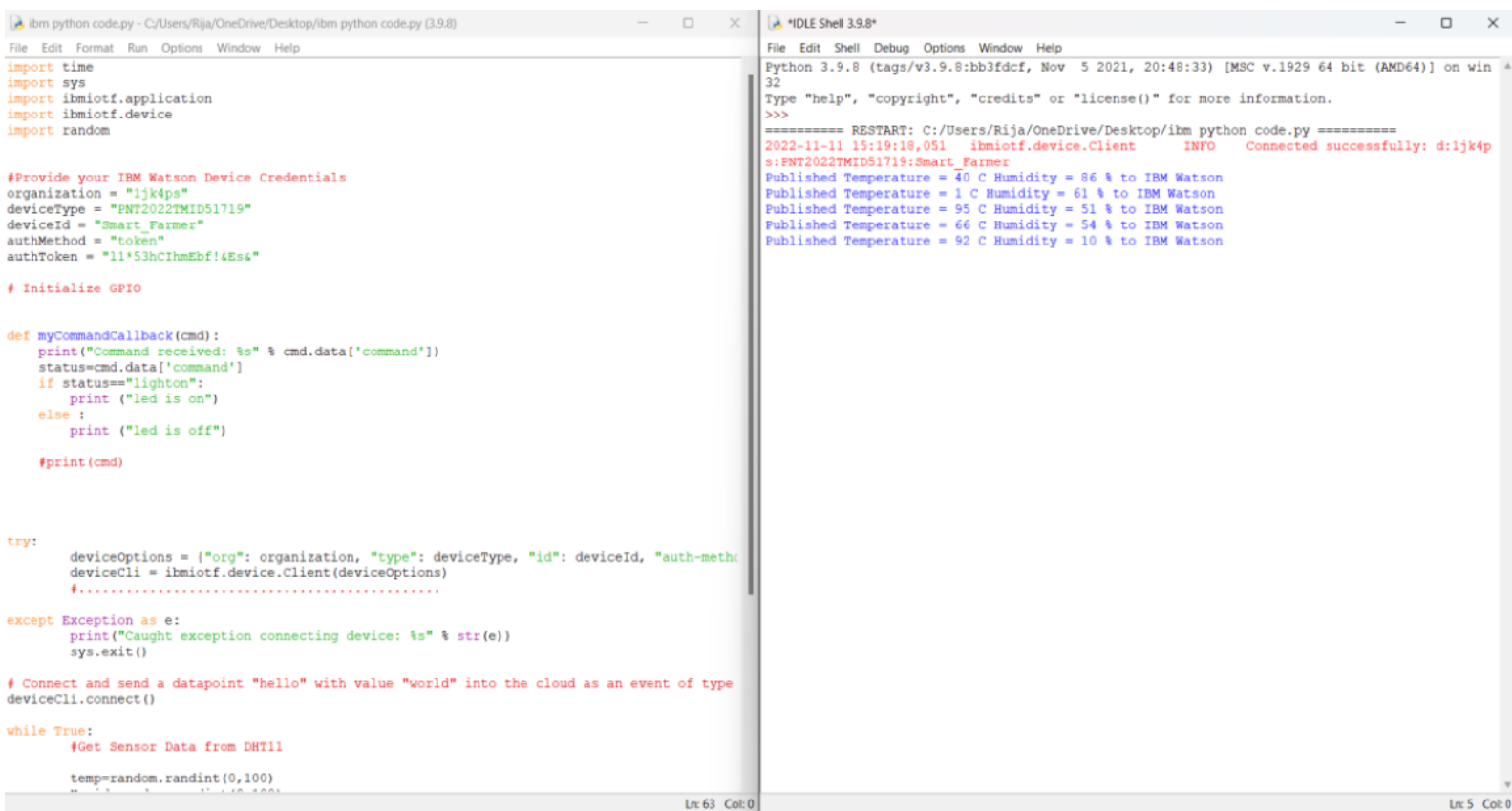
# Initialize GPIO
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="lighton":
        print ("led is on")
    else :
        print ("led is off")
    #print(cmd)
try:
```

```

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)
Humid=random.randint(0,100)
Data = { 'temp','Humid':Humid}
#print data
Def myOnPublishCallback():
Print("Published Temperature = %s C" % temp, "Humidity = %s %% " %
Humid, "to IBM Watson")
success = deviceCli.publishEvent("SDFRN", "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:



The image shows a screenshot of a code editor and a terminal window. The code editor on the left contains a Python script named `ibm python code.py`. The script imports `time`, `sys`, `ibmiotf.application`, `ibmiotf.device`, and `random`. It defines device credentials and initializes the `ibmiotf.device.Client`. A `myCommandCallback` function is defined to handle commands like 'light on' and 'light off'. The script then connects to the cloud and enters a loop to send random temperature data to IBM Watson IoT.

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

#Provide your IBM Watson Device Credentials
organization = "ijk4ps"
deviceType = "PNT2022TMID51719"
deviceId = "Smart_Farmer"
authMethod = "token"
authToken = "11*53hC1hmEbf!&Es"

# Initialize GPIO

def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="lighton":
        print ("led is on")
    else :
        print ("led is off")

    #print(cmd)

try:
    deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method": authMethod, "device-id": deviceId}
    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 "hello"
deviceCli.connect()

while True:
    #Get Sensor Data from DHT11
    temp=random.randint(0,100)
```

The terminal window on the right shows the execution of the script. It displays the restart command, the connection status, and the published data points for temperature and humidity.

```
Python 3.9.8 (tags/v3.9.8:bb3fddf, Nov 5 2021, 20:48:33) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/Rija/OneDrive/Desktop/ibm python code.py =====
2022-11-11 15:19:18,051 ibmiotf.device.Client INFO Connected successfully: d:ijk4ps:Smart_Farmer
Published Temperature = 40 C Humidity = 86 % to IBM Watson
Published Temperature = 1 C Humidity = 61 % to IBM Watson
Published Temperature = 95 C Humidity = 51 % to IBM Watson
Published Temperature = 66 C Humidity = 54 % to IBM Watson
Published Temperature = 92 C Humidity = 10 % to IBM Watson
```