

SPRINT 2

Date	19 November 2022
Team ID	PNT2022TMID38164
Project Name	Project – Smart Farmer-IoT Enabled smart Farming Application

TEAM MEMBERS:-

BAVADHARANI K	411819106002
SATHISH M	411819106005
ESAKKIRAJAN M	411819106305
ARUNKUMAR A	411819106001

PYTHON SCRIPT :-

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

organization = "zxnybt"
deviceType = "dominators"
deviceId = "12345"
authMethod = "token"
authToken = "123456789"

def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data)
    for key in cmd.data.keys():
        if key == 'motor':
            if cmd.data['motor'] == 'ON':
                print("MOTOR is turned ON")

            elif cmd.data['motor'] == 'OFF':
                print("MOTOR is turned OFF")
    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:
```

```
    temp=random.randint(0,40)  
    Humid=random.randint(0,100)  
    moist=random.randint(0,40)  
    data = { 'temperature' : temp, 'humidity': Humid, 'soil_moisture':moist  
}
```

```
    def myOnPublishCallback():  
        print ('Published Temperature = %s C' % temp, 'Humidity = %s  
%%' % Humid, 'soil moisture =%s' % moist,'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()
```

PYTHON SCRIPT CONFIGURED TO IBM WATSON IoT PLATFORM :-

IBM Watson IoT Platform

sugendran1928@gmail.com
ID: zzxybt

?

?

Grid icon

Dashboard icon

Users icon

Person icon

Network icon

Signal icon

Lock icon

Settings icon

Browse

Action

Device Types

Interfaces

Add Device +

Browse Devices

All DevicesDiagnose

This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.

Search by Device ID

Device Simulator ☒

Filter icon

Refresh icon

	Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
>	<input type="checkbox"/> 12345	Connected	dominators	Device	Nov 3, 2022 3:08 PM	

Items per page 50 | 1-1 of 1 item

1 of 1 page < 1 >

0 Simulations running

THE SENSOR DATAS IN THE PYTHON SCRIPT WILL BE RECEIVED BY IBM WATSON IOT PLATFORM:-

The screenshot displays a dual-pane environment. On the left, a terminal window titled 'IDLE Shell 3.9.8*' shows the execution of a Python script. The script connects to the IBM Watson IoT Platform, and the output indicates a successful connection for device '12345'. The output also shows the device's current temperature and humidity readings.

On the right, a web browser window displays the IBM Watson IoT Platform dashboard. The dashboard shows the device '12345' is connected. Below the device header, there are tabs for Identity, Device Information, Recent Events, State, and Logs. The 'Recent Events' tab is selected, showing a table of live stream data. The table has columns for Event, Value, Format, and Last Received. The events listed show temperature and humidity readings for the device.

Event	Value	Format	Last Received
IoTSensor	{"temperature":1,"humidity":63,"soil_moisture":...	json	a few seconds ago
IoTSensor	{"temperature":22,"humidity":43,"soil_moisture":...	json	a few seconds ago
IoTSensor	{"temperature":39,"humidity":34,"soil_moisture":...	json	a few seconds ago