

SPRINT DELIVERY – IV

Project Name: SmartFarmer-IoT Enabled SmartFarming Application.

Team ID: PNT2022TMID32489

Develop A Mobile Application

- ✓ Node-Red Contains /Data and /Command Requests
- ✓ /Data command is used to access the MIT App Inventor
- ✓ /Command is used to send and receive the Motor Control response and request(on/off)
- ✓ Every one second the parameter is update
- ✓ App inventor contains blocks, with these help of these blocks we can build our application

Source Code

```
import wiotp.sdk.device
import time
import os
import datetime
import random
myConfig = {
    "identity": {
        "orgId": "7hx5v5",
        "typeId": "Nodemcu",
        "deviceId": "8100"
    },
    "auth": {
        "token": "8100192050"
    }
}
client = wiotp.sdk.device.DeviceClient(config = myConfig, logHandlers = None)
client.connect()
def myCommandCallback(command):
    print("Message recieved from IBM IoT Platform:
%s"%command.data['command'])
    m = command.data['command']
    if(m=="motoron"):
        print("Motor is switched on")
    elif(m=="motoroff"):
        print("Motor is switched Off")
    while True:
        moisture = random.randint(0,100)
        temp = random.randint(0,125)
        hum = random.randint(0,100)
```

The screenshot shows the Visual Studio Code interface with a Python file named 'myCommandCallback.py' open. The code is as follows:

```

1 import datetime
2 import random
3 myConfig = {
4     "identity": {
5         "orgId": "7hx5v5",
6         "typeId": "Nodemcu",
7         "deviceId": "8100"
8     },
9     "auth": {
10        "token": "8100192050"
11    }
12 }
13 client = wiotp.sdk.device.DeviceClient(config = myConfig, logHandlers = None)
14 client.connect()
15
16 def myCommandCallback(command):
17     print("Message received from IBM Iot PlatForm: %s"%command.data['command'])
18     m = command.data['command']
19     if(m=="motoron"):
20         print("Motor is switched on")
21     elif(m=="motoroff"):
22         print("Motor is switched off")
23
24 while True:
25     moisture = random.randint(0,100)
26     temp = random.randint(0,125)
27     hum = random.randint(0,100)
28     myData = {'soil_moisture':moisture,'temperature':temp,'humidity':hum}
29     client.publishEvent(eventId = "status", msgFormat = "json", data = myData, qos=0, onPublish = None)
30     print("Published data Successfully: %s"%myData)
31     time.sleep(2)
32     client.commandCallback = myCommandCallback
33
34 client.disconnect()
35

```

The interface also shows the Explorer sidebar on the left with options like 'Open Folder' and 'Clone Repository'. The bottom status bar indicates the file is at line 21, column 16, and the Python version is 3.10.5 64-bit.

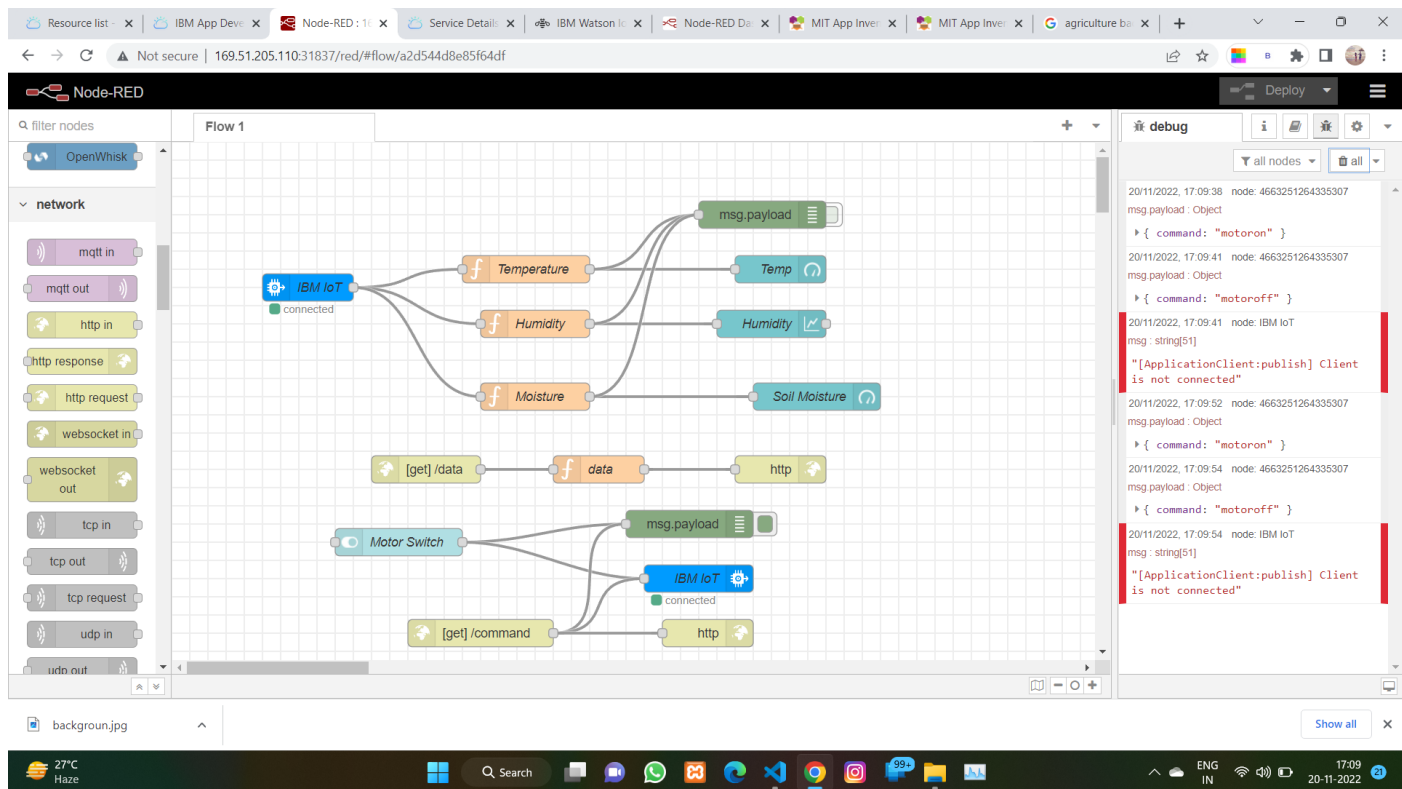
The image shows a Windows desktop environment. In the foreground, a terminal window is open, displaying the execution of a Python script. The script uses the `wiotp.sdk.device.client.DeviceClient` to connect to an IBM Watson IoT Platform instance and publishes 20 data points. Each data point contains `'soil_moisture'`, `'temperature'`, and `'humidity'` values. The output shows successful connections and data publishing for each point.

```

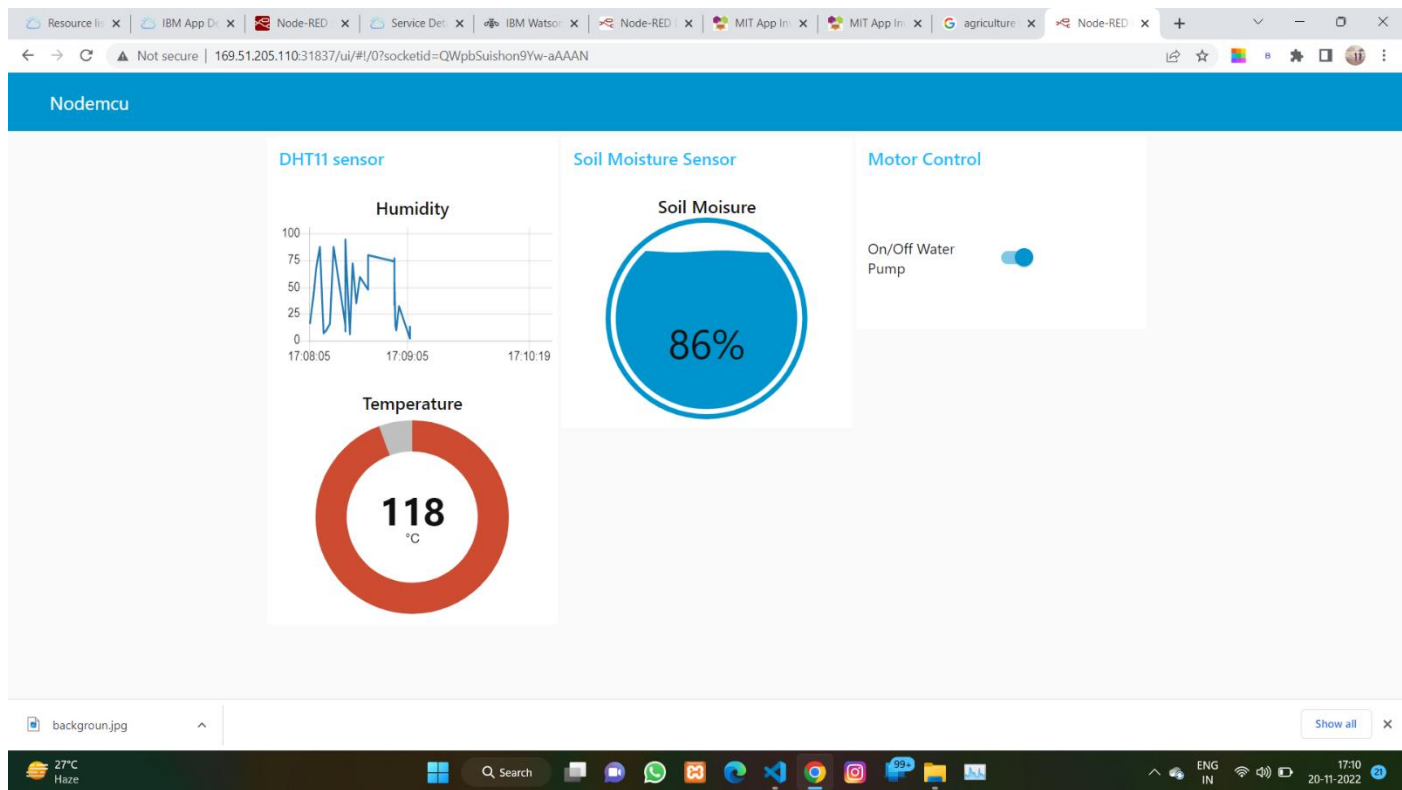
D:\IBM Project>python pythoncode.py
2022-11-20 17:08:05,775 wiotp.sdk.device.client.DeviceClient INFO Connected successfully: d:7hx5v5:Nodecmcu:8100
Published data Successfully: {'soil_moisture': 45, 'temperature': 43, 'humidity': 16}
Published data Successfully: {'soil_moisture': 68, 'temperature': 75, 'humidity': 43}
Published data Successfully: {'soil_moisture': 2, 'temperature': 59, 'humidity': 68}
Published data Successfully: {'soil_moisture': 16, 'temperature': 44, 'humidity': 87}
Published data Successfully: {'soil_moisture': 92, 'temperature': 62, 'humidity': 7}
Published data Successfully: {'soil_moisture': 59, 'temperature': 32, 'humidity': 10}
Published data Successfully: {'soil_moisture': 26, 'temperature': 3, 'humidity': 16}
Published data Successfully: {'soil_moisture': 28, 'temperature': 28, 'humidity': 87}
Published data Successfully: {'soil_moisture': 44, 'temperature': 28, 'humidity': 15}
Published data Successfully: {'soil_moisture': 10, 'temperature': 98, 'humidity': 9}
Published data Successfully: {'soil_moisture': 33, 'temperature': 119, 'humidity': 94}
Published data Successfully: {'soil_moisture': 12, 'temperature': 64, 'humidity': 73}
Published data Successfully: {'soil_moisture': 2, 'temperature': 87, 'humidity': 6}
Published data Successfully: {'soil_moisture': 88, 'temperature': 47, 'humidity': 72}
Published data Successfully: {'soil_moisture': 61, 'temperature': 91, 'humidity': 35}
Published data Successfully: {'soil_moisture': 34, 'temperature': 79, 'humidity': 59}
Published data Successfully: {'soil_moisture': 32, 'temperature': 91, 'humidity': 48}
Published data Successfully: {'soil_moisture': 97, 'temperature': 28, 'humidity': 80}
Published data Successfully: {'soil_moisture': 80, 'temperature': 88, 'humidity': 74}
Published data Successfully: {'soil_moisture': 14, 'temperature': 83, 'humidity': 62}
Published data Successfully: {'soil_moisture': 25, 'temperature': 109, 'humidity': 77}
Published data Successfully: {'soil_moisture': 100, 'temperature': 40, 'humidity': 33}
Published data Successfully: {'soil_moisture': 90, 'temperature': 98, 'humidity': 57}
Published data Successfully: {'soil_moisture': 59, 'temperature': 108, 'humidity': 60}
Published data Successfully: {'soil_moisture': 90, 'temperature': 55, 'humidity': 57}
Published data Successfully: {'soil_moisture': 33, 'temperature': 75, 'humidity': 15}
Published data Successfully: {'soil_moisture': 97, 'temperature': 119, 'humidity': 10}
  
```

In the background, a web browser window displays the IBM Watson IoT Platform interface. The interface includes a sidebar with navigation icons and a main content area with tabs for 'Browse', 'Action', 'Device Types', and 'Int'. The browser's address bar shows the URL `7hx5v5.internetofthings.ibmcloud.com/dp`.

Node-Red Config

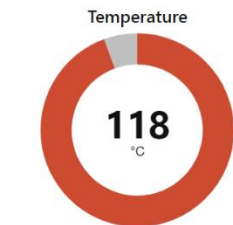
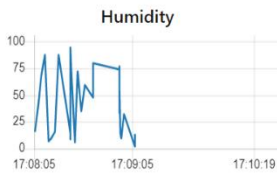


Application Output

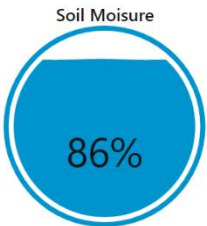


Nodemcu

DHT11 sensor



Soil Moisture Sensor



Motor Control

On/Off Water Pump



backgroun.jpg

Show all



27°C
Haze



Search



ENG
IN



17:10
20-11-2022