

SPRINT 4

Date	14 November 2022
Team ID	PNT2022TMID46860
Project Name	Project – Smart Farmer- IoT based Smart Farming Application

PYTHON CODE TO CONNECT IBM CLOUD

The screenshot shows a Windows desktop with a text editor on the left and a Python 3.7.0 Shell on the right. The text editor contains a Python script that connects to IBM Watson IoT and prints sensor data. The Python Shell shows the output of the script, which includes a list of sensor data points.

```
python to iot.py - C:\Users\ELCOT\AppData\Local\Programs\Python\Python37\python
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 = "j3bgcj"
deviceType = "nodeMCU"
deviceId = "1234"
authMethod = "token"
authToken = "12345678"

# Initialize GPIO

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

try:
    deviceOptions = {"org": organization, "type": deviceType, "id": deviceId}
    deviceCli = ibmiotf.device.Client(deviceOptions)
    #.....

Restart: C:\Users\ELCOT\AppData\Local\Programs\Python\Python37\python to iot.py
2022-11-18 18:18:42,868 ibmiotf.device.Client INFO Connected successfully: dj3bgcjnodeMCU
:1234
Published Temperature = 5 C Humidity = 10 % soil_moisture =47C to IBM Watson
Published Temperature = 3 C Humidity = 17 % soil_moisture =37C to IBM Watson
Published Temperature = 13 C Humidity = 58 % soil_moisture =37C to IBM Watson
Published Temperature = 14 C Humidity = 53 % soil_moisture =31C to IBM Watson
Published Temperature = 16 C Humidity = 71 % soil_moisture =30C to IBM Watson
Published Temperature = 11 C Humidity = 40 % soil_moisture =49C to IBM Watson
Published Temperature = 11 C Humidity = 35 % soil_moisture =48C to IBM Watson
Published Temperature = 0 C Humidity = 6 % soil_moisture =34C to IBM Watson
Published Temperature = 17 C Humidity = 78 % soil_moisture =47C to IBM Watson
Published Temperature = 22 C Humidity = 67 % soil_moisture =42C to IBM Watson
Published Temperature = 27 C Humidity = 9 % soil_moisture =34C to IBM Watson
Published Temperature = 1 C Humidity = 91 % soil_moisture =46C to IBM Watson
Published Temperature = 16 C Humidity = 79 % soil_moisture =34C to IBM Watson
Published Temperature = 10 C Humidity = 18 % soil_moisture =37C to IBM Watson
Published Temperature = 24 C Humidity = 6 % soil_moisture =39C to IBM Watson
Published Temperature = 17 C Humidity = 37 % soil_moisture =42C to IBM Watson
Published Temperature = 14 C Humidity = 1 % soil_moisture =50C to IBM Watson
Published Temperature = 30 C Humidity = 23 % soil_moisture =45C to IBM Watson
Published Temperature = 13 C Humidity = 28 % soil_moisture =36C to IBM Watson
Published Temperature = 0 C Humidity = 74 % soil_moisture =32C to IBM Watson
Published Temperature = 11 C Humidity = 3 % soil_moisture =33C to IBM Watson
Published Temperature = 6 C Humidity = 98 % soil_moisture =31C to IBM Watson
Published Temperature = 16 C Humidity = 61 % soil_moisture =34C to IBM Watson
Published Temperature = 21 C Humidity = 61 % soil_moisture =44C to IBM Watson
Published Temperature = 9 C Humidity = 61 % soil_moisture =50C to IBM Watson
Published Temperature = 1 C Humidity = 55 % soil_moisture =49C to IBM Watson
Published Temperature = 10 C Humidity = 20 % soil_moisture =44C to IBM Watson
Published Temperature = 12 C Humidity = 32 % soil_moisture =44C to IBM Watson
```

UI INTERFACE USING WEB API KEY ON IBM

The screenshot shows the Node-RED web interface in a browser. The top bar includes tabs for Node-RED, an email inbox, and IBM Cloud services. The main workspace displays a flow with a 'Hello Node-RED!' node, an 'IBM IoT' node (connected), and three function nodes for 'temperature', 'humidity', and 'soilmoisture'. A 'gauge' node is also present, displaying a value of 38. The 'Edit gauge node' panel is open, showing properties for the gauge, including Group, Size, Type, Label, Value format, Units, Range, and Colour gradient. The 'debug' console on the right shows a series of log messages from the IoT node.

TEMPERATURE

The screenshot shows the Node-RED web interface with the same flow as the previous image. The 'Edit chart node' panel is open, showing properties for the line chart, including Group, Size, Type, Label, X-axis, Y-axis, Legend, and Series Colours. The 'debug' console on the right shows a series of log messages from the IoT node.

COMMAND FUNCTION TO GET

The screenshot shows the Node-RED web interface in a browser. The top bar includes tabs for Node-RED, an email inbox, and IBM Cloud services. The main workspace displays a flow with an 'IBM IoT' node (labeled 'connected') that triggers a 'Hello Node-RED!' message. This node is connected to a function node. The 'Edit function node' panel is open, showing the following JavaScript code:

```
1 msg.payload = {  
2   "temp":global.get("t"),  
3   "Humid":global.get("h"),  
4   "soil_moisture":global.get("s")  
5 }  
6 return msg;
```

The right sidebar shows a 'debug' console with a list of messages. The bottom of the screen shows a Windows taskbar with various application icons and a system clock indicating 18:32 on 18-11-2022.

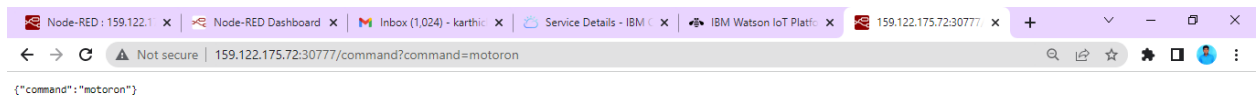
COMMAND DISPLAY ON OTHER WINDOW SCREEN

The screenshot shows a web browser window with the URL `159.122.175.72:30777/sensor`. The page displays a JSON object representing sensor data:

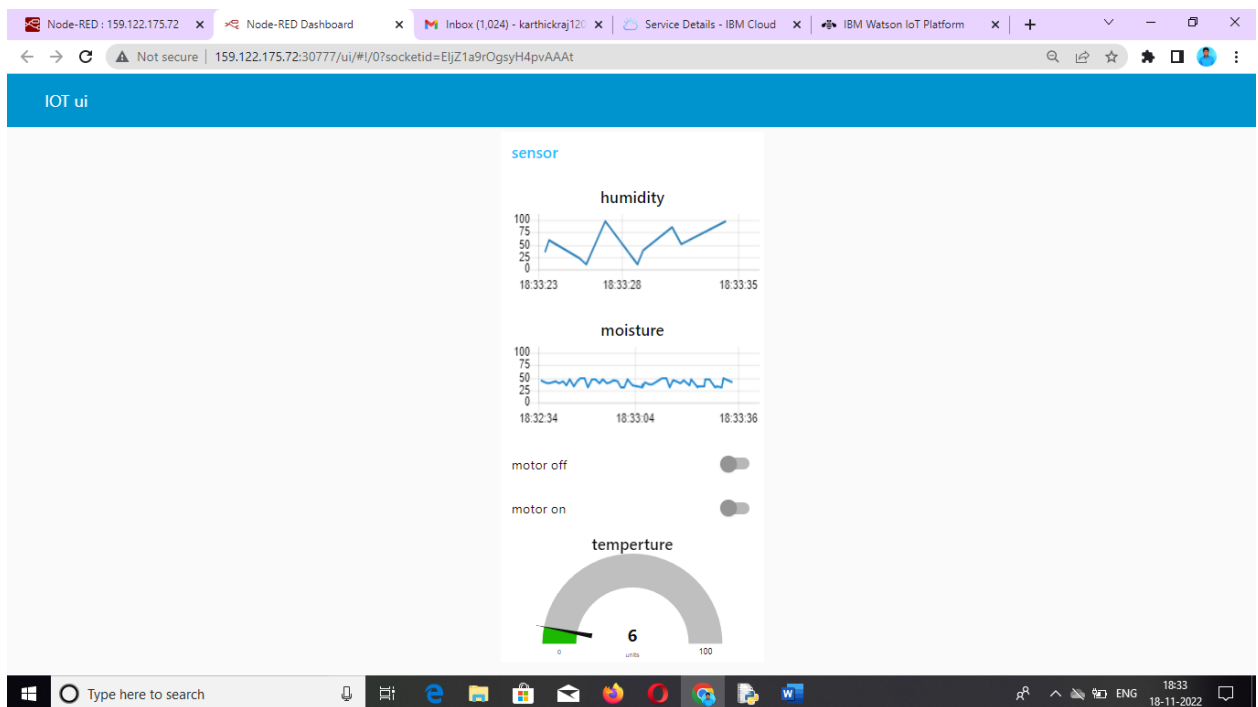
```
{ "temp":12,"Humid":84,"soil_moisture":34 }
```

The browser's address bar and tabs are visible at the top, and a Windows taskbar is at the bottom, showing the system clock as 18:35 on 18-11-2022.

COMMAND FOR MOTOR ON



WEB UI USER INTERACT WITH SOFTWARE



MOBILE APPLICATION

