

- 
- ▶ SPRINT 2 REPORT
 - ▶ SMARTFARMER – IoT ENABLED SMART FARMING APPLICATION

TEAM ID – PNT2022TMID22142

- TEAM LEADER : NAVEEN V M
- TEAM MEMBER : GHAJENDHIRAN J
- TEAM MEMBER : GANESA MOORTHY M
- TEAM MEMBER : Koushik P H

Project Tracker

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	15	5 Days	26 Oct 2022	30 Oct 2022	15	30 Oct 2022
Sprint-2	15	7 Days	31 Oct 2022	06 Nov 2022	15	07 Nov 2022
Sprint-3	15	6 Days	07 Nov 2022	12 Nov 2022	15	13 Nov 2022
Sprint-4	15	6 Days	13 Nov 2022	18 Nov 2022		18 Nov 2022 – 19 Nov 2022

S.NO	Tools & Technology Used
1	Python 3.7.0
2	IBM Cloud
3	Node-Red

Python Script:

```
RED NODE.py - F:\RED NODE.py (3.11.0)
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 = "382b93" #replace the ORG ID
deviceType = "GGT" #replace the Device type w/
deviceId = "1234" #replace Device ID
authMethod = "token"
authToken = "q3SrVFQ4Qm_q79v1" #Replace the authToken
# Initialize GPIO

#Receives Command from Node-red
def myCommandCallback(cmd):
    print ("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="motoron":
        print ("motor is on")
    elif status == "motordoff" :
        print ("motor is off")
    elif status == "motor30" :
        print ("motor is on for 30 minutes")

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)
    soilmoisture=random.randint(0,100)

    data = { 'temp': temp, 'Humid': Humid, 'soilmoisture': soilmoisture }
    #print data
    def myOnPublishCallback():
        print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % Humid, "soilmoisture = %s %%"
        %soilmoisture, "to IBM Watson")

    success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publish=myOnPublishCallback)
    if not success:
        print("Not connected to IoT")
        time.sleep(5)

deviceCli.commandCallback = myCommandCallback

# Disconnect the device and application from the cloud
deviceCli.disconnect()
```

OUTPUT:

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\charu\Downloads\ibmiotpublishsubscribe.py =====

2022-11-11 15:56:49,907 ibmiotf.device.Client INFO Connected successfully: d:x0fxss:Testing:Testdevice1

Published Temperature = 8 C Humidity = 44 % soilmoisture = 3 % to IBM Watson

Published Temperature = 13 C Humidity = 95 % soilmoisture = 43 % to IBM Watson

Published Temperature = 78 C Humidity = 83 % soilmoisture = 83 % to IBM Watson

Published Temperature = 100 C Humidity = 52 % soilmoisture = 60 % to IBM Watson

Published Temperature = 45 C Humidity = 93 % soilmoisture = 16 % to IBM Watson

Published Temperature = 53 C Humidity = 12 % soilmoisture = 59 % to IBM Watson

Published Temperature = 15 C Humidity = 49 % soilmoisture = 32 % to IBM Watson

Published Temperature = 37 C Humidity = 73 % soilmoisture = 25 % to IBM Watson

IBM Watson IoT Platform

?

312819106035@smartinternz.com
ID: x0fxss

Browse

Action

Device Types

Interfaces

Add Device +

>

1234

Disconnected

Noder

Device

24 Oct 2022 09:50

▼

Testdevice1

Connected

Testing

Device

11 Nov 2022 15:08

→ ...

Identity

Device Information

Recent Events

State

Logs

×

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
IoTSensor	{"temp":59,"Humid":96,"soilmoisture":100}	json	a few seconds ago
IoTSensor	{"temp":26,"Humid":59,"soilmoisture":99}	json	a few seconds ago
IoTSensor	{"temp":74,"Humid":13,"soilmoisture":96}	json	a few seconds ago
IoTSensor	{"temp":79,"Humid":24,"soilmoisture":28}	json	a few seconds ago

The screenshot displays the Node-RED web interface in a browser. The main workspace shows a flow named 'Flow 1'. The flow starts with a 'load up i' node (blue) which is connected to three parallel processing nodes: 'temp' (orange), 'humidity' (orange), and 'soil moisture' (orange). Each of these nodes is connected to a corresponding output node: 'temperature' (teal), 'humidity' (teal), and 'soil moisture' (teal). Additionally, the 'temp' node is connected to a 'msg.payload' node (green). Below this, there are two input nodes: 'motor on' (teal) and 'motor off' (teal), both connected to a 'text input' node (teal). The 'text input' node is connected to an 'IBM IoT OUT' node (blue). The right sidebar shows a 'debug' console with a log of messages. The log includes a '750N Message expected' error and several JSON objects representing weather data, such as:

```
{ id: 701, weather: "Mist", detail: "mist", icon: "50n", tempk: 299.16 }
```

IBM

IBM-EPBL/IBM-Proje...

Node-RED.pdf

IBM Watson IoT Platform

Node-RED : node-red

Node-RED Dashboard

IBM Cloud and Node-RED

Download file (Love?)

IBM App Development

←

→

↻

🔒

🌐

https://node-red-zefid-2022-11-08.au-syd.mybluemix.net/ui/#/0?socketid=W5f-mtrUapWNTIAAAJ

100%

☆

📁

⬇

🐦

☰

Smart Agriculture System

Welcome!!

soil moisture

0

10

cm

temprature

0

10

cm

humidity

0

10

cm

7c98b456c8900513

MOTOR ON

MOTOR OFF