

The IBM logo is displayed in white text on a blue arrow-shaped background, which is part of a larger blue horizontal bar. To the left of this bar is a thick, dark blue vertical bar. In the bottom-left corner, there are several thin, curved, light blue lines that sweep upwards and to the right.

# SPRINT 1 REPORT

**TEAM ID – PNT2022TMID19528**

- **TEAM LEADER : SUBAHAN.K**
- **TEAM MEMBER : SAI KUMAR.G**
- **TEAM MEMBER : K.MD.HASSAIN**
- **TEAM MEMBER : NIKHIL.K**

# 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

<b>S.NO</b>	<b>Tools &amp; Technology Used</b>
1	Python 3.7.0
2	IBM Cloud
3	Node-Red

## Python Script:

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

#Provide your IBM Watson Device Credentials
organization = "x0fxss" #replace the ORG ID
deviceType = "Testing"#replace the Device type wi
deviceId = "Testdevice1"#replace Device ID
authMethod = "token"
authToken = "123456789" #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 == "motoroff" :
        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 %"
%ssoilmoisture, "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:

We are running python script to send data to IBM cloud and data is displayed in web-ui by using node-red.

```
*Python 3.7.0 Shell*
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

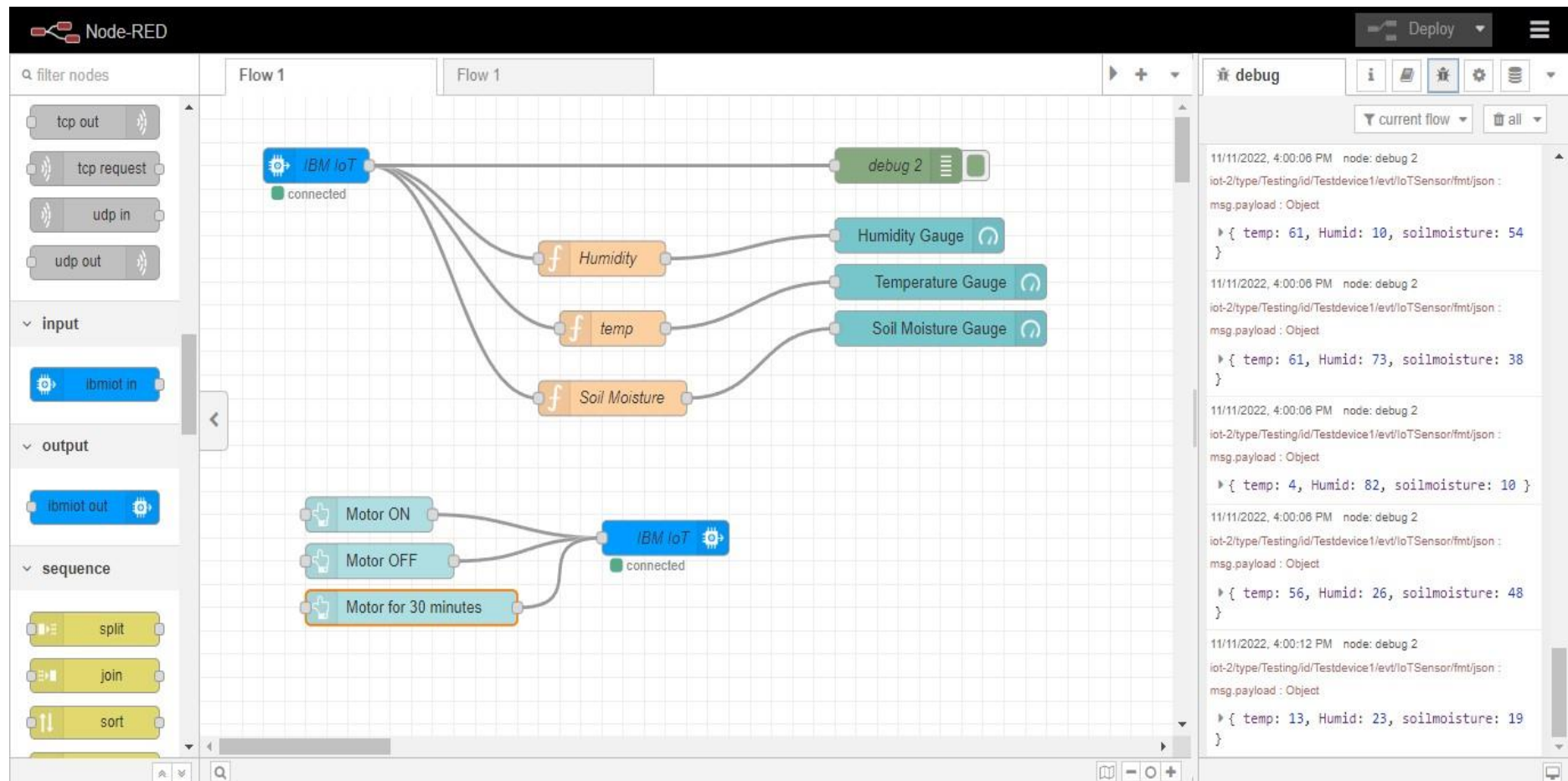
State

Logs

×

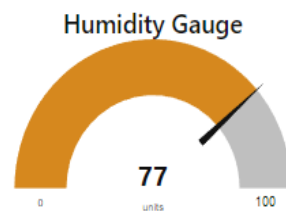
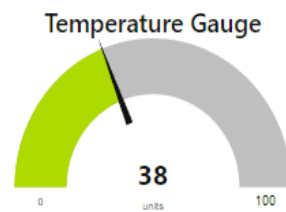
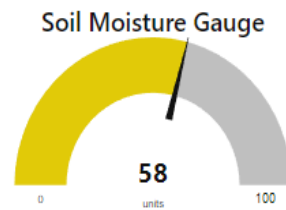
The recent events listed show the live stream of data that is coming and going from this device.

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





Default



Group 2

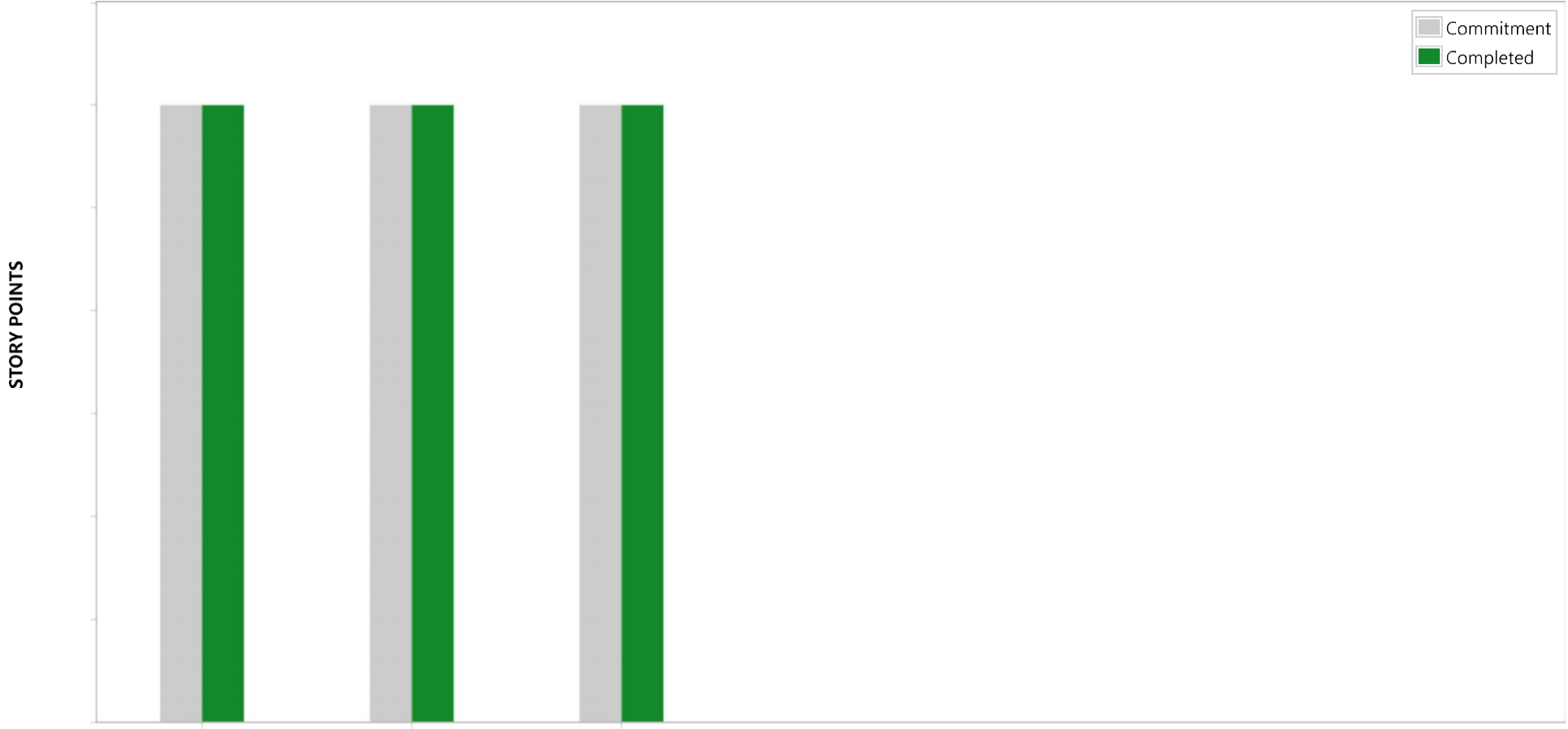
MOTOR ON

MOTOR OFF










MOTOR FOR 30 MINUTES

Data are successfully received and displayed.

Velocity Chart



Sprint	Commitment	Completed
SFDP Sprint 1	15	15
SFDP Sprint 2	15	15
SFDP Sprint 3	15	15

	NOV				NOV								NOV			
	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
Sprints	SFDP Sprint 2				SFDP Sprint 3											
Releases																
>  <u>SFDP-1</u> This Epic is to accomplish user should able...																
>  <u>SFDP-2</u> This Epic is to accomplish user should able...																
>  <u>SFDP-3</u> This Epic is to create a dashboard in our ap...																
>  <u>SFDP-4</u> This Epic is to accomplish the IoT Device C...																
>  <u>SFDP-5</u> This Epic is to accomplish to solve the user...																
>  <u>SFDP-6</u> This Epic is to accomplish to solve the user...																
>  <u>SFDP-7</u> This Epic is to accomplish to solve particula...																
>  <u>SFDP-8</u> This Epic is to accomplish the conection wit...																
>  <u>SFDP-9</u> This Epic is to accomplish to application Cr...																

## Burndown Chart



SFDP Sprint 3

Story Points ▼

