

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 resemble stylized grass or reeds.

SPRINT 3 REPORT

IOT ENABLED SMART
FARMING APPLICATION

TEAM ID – PNT2022TMID25066
SOWNDHARYASREE.S
SHREEYAZHINI.A
RIDANYA.K
SWETHA.P

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:

```
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
```

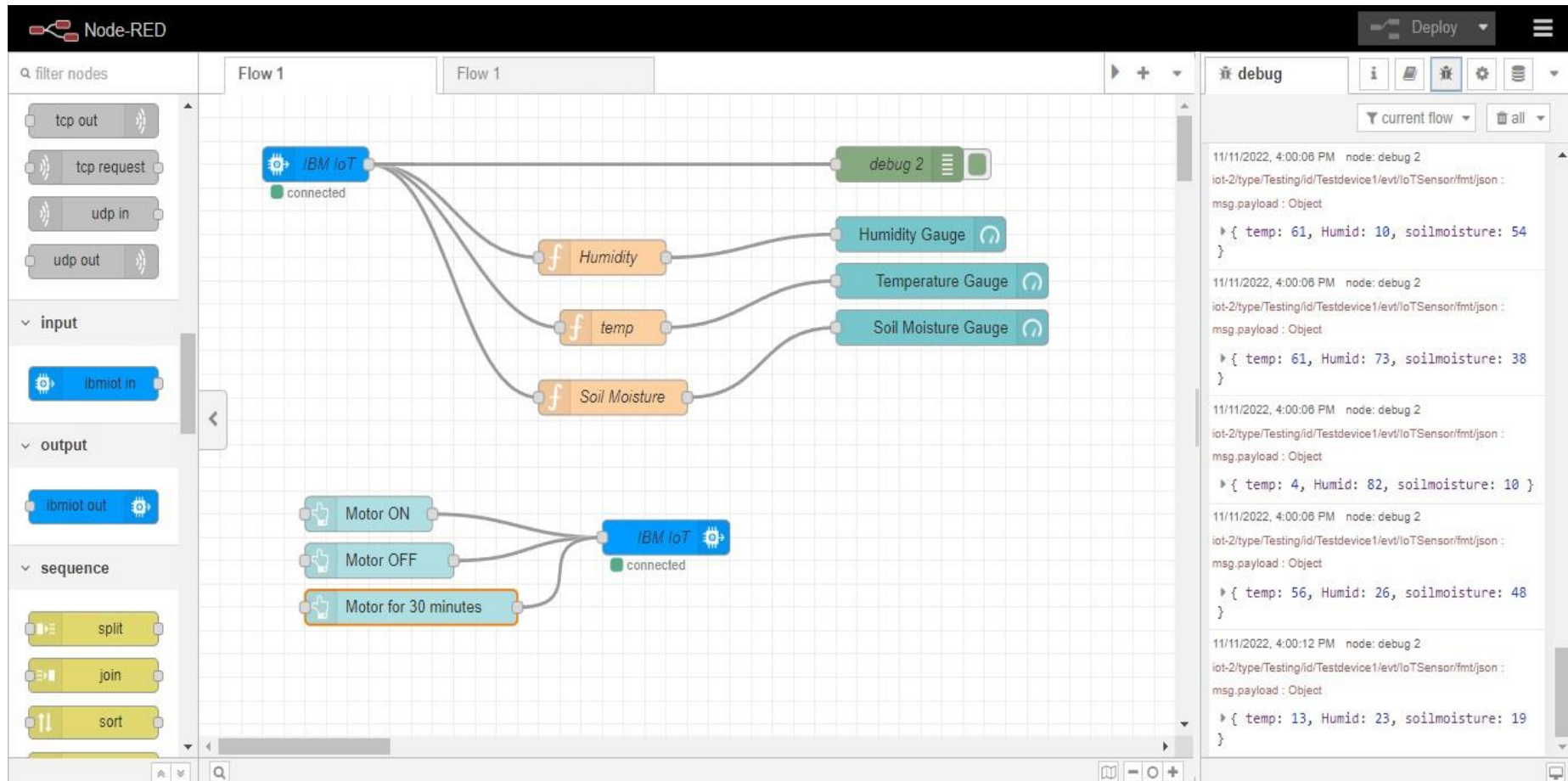
Add Device +

> 1234 Disconnected Nuder Omit 24OmZ02209:B0

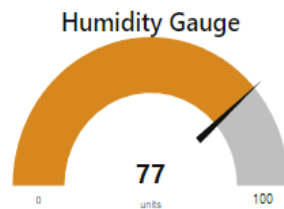
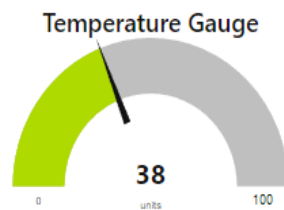
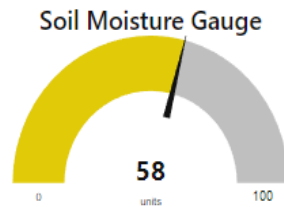
Identity Device Information Recent Events State Logs X

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

Ewent	Vahie	Formet	Last Reue?ued
IoTSénsor	{"temp":59,"Humid":96,"soilmoisture":100}	json	a few seconds ago a
IoTSénsor	{"temp":Z6,"Humid":59,"soilmoisture":99}	jsnn	few seconds ago a
IoTSensor	{"temp":74,"Humid":13,"soilmoisture":96}	jsnn	few seconds ago
IoTSénsor	{"temp":79,"Humid":24,"soilmoisture":2B}	jsnn	a few seconds ago



Default



Group 2

MOTOR ON

MOTOR OFF

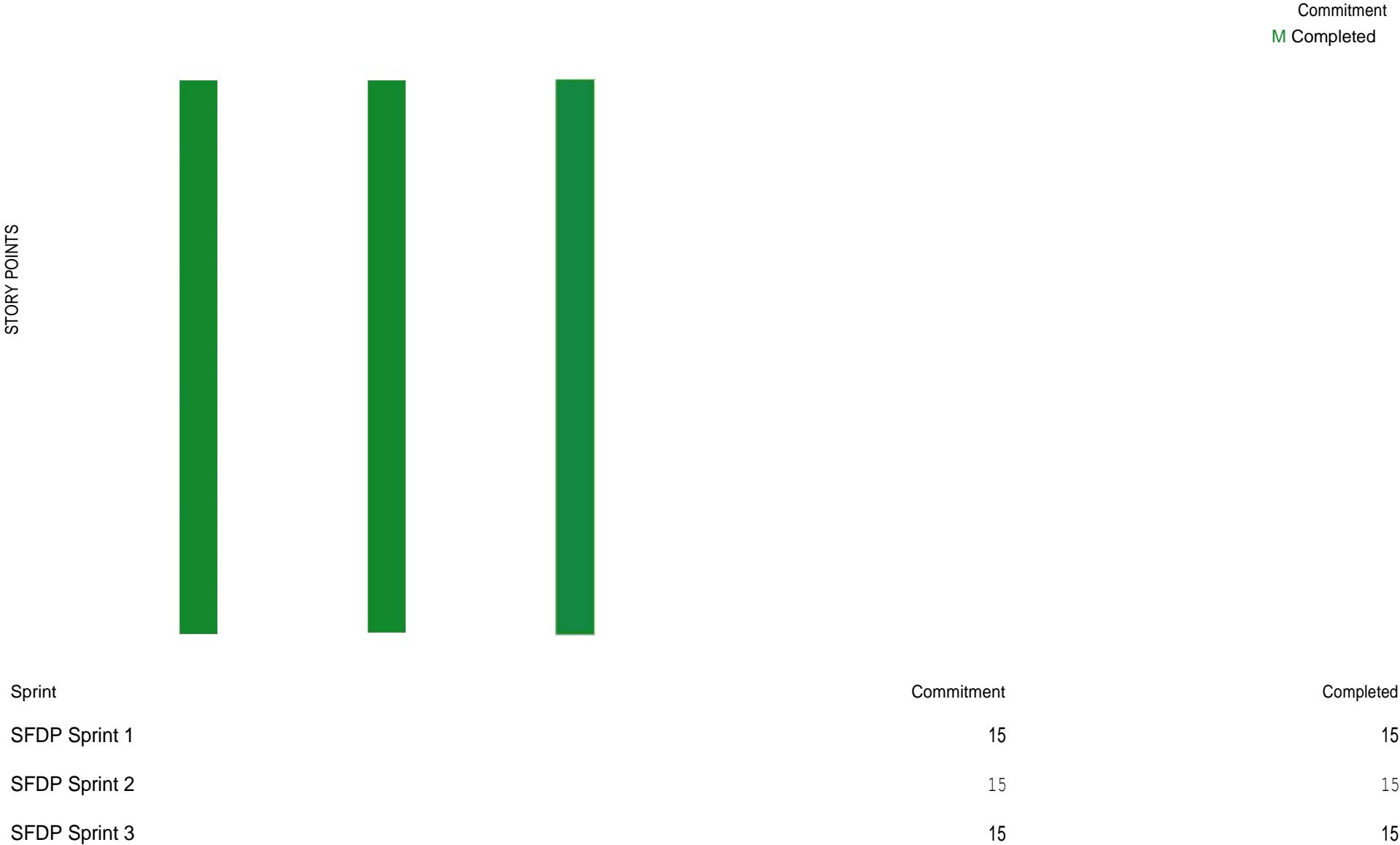
MOTOR FOR 30 MINUTES

Data are successfully received and displayed.



y

Velocity Chart



Sprints

SFDP Sprint 2

SFDP Sprint 3

Releases

» @ SFDP-1 This Epic is to accompiisn user should able.

> B SFDP-Z This Epic is to accomplish user should able...

> @ SFOP-3 This Epic is to create a dasno rd in our ap.

› B srD -q This Epic is to accomplish the IoT Device C...

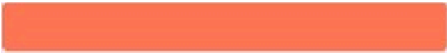
› a sro -s This Epic is to accomplish to solve the user...

* A SFDP-6 ThlsEpic is to accomplish io solve the user. .

» 0 SFDP-7 This Epic is to accomplish to solve particula.

› 0 sro g This Epic is to accomplish ihe conection we.

* @ SF DP-9 This Epic is to accomplish to application Cr ..



Burndown Chart



SFDP Sprint 3

Story Points ▾

