

## SPRINT 2

- **Developed the Python script and published the data to the IBM cloud (Watson IoT Platform):**
- Python 3.7.0 version is installed and ibmiot library is imported including the other required libraries. The IBM Watson device credentials are given – organization type, authentication token, etc.

### **Code (Python Script):**

```
import time

import sys

import ibmiotf.application

import ibmiotf.device

import random


#Provide your IBM Watson Device Credentials

organization = "f1pgwv"

deviceType = "raspberrypi"

deviceId = "123"

authMethod = "token"

authToken = "12345678"


# Initialize GPIO

def myCommandCallback(cmd):

    print("Command received: %s" % cmd.data['command'])

    status=cmd.data['command']

    if status=="on":

        print ("control is on")

    else :

        print ("control is off")


    #print(cmd)

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(20,40)
    Turbidit=random.uniform(0.1,5)
    Turbidity=round(Turbidit,2)
    p=random.uniform(6.5,8.5)
    pH=round(p,2)
    Hardness=random.randint(75,300)


    data = { 'temp' : temp, 'Turbidity': Turbidity , 'pH': pH, 'Hardness': Hardness}

    #print data

    def myOnPublishCallback():

        print ("Published Temperature = %s C" % temp, "Turbidity = %s " %
Turbidity,"pH = %s " % pH,"Hardness = %s " % Hardness, "to IBM Watson")


    success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,
on_publish=myOnPublishCallback)

    if not success:

        print("Not connected to IoTf")

```

```
time.sleep(10)
```

```
deviceCli.commandCallback = myCommandCallback
```

```
# Disconnect the device and application from the cloud
```

```
deviceCli.disconnect()
```

- **deviceCli.connect()** – This line of code connects the device with the IBM Cloud. This is done to integrate Python code and the device in IBM Watson Cloud platform.
- Dataset in code for the sensor values is generated using random function and a dictionary of sensed parameters is created. The parameters are ph, turbidity, hardness and temperature.
- Output of the code is shown which has the continuous real timed sensed parameter values sending to IBM Cloud.

## Output:

```
*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\divya\Downloads\Python\publish_subscribe_data.py =====
2022-11-19 13:06:19,487 ibmiotf.device.Client INFO Connected successfully: d:flpgwv:raspberrypi:123
Published Temperature = 37 C Turbidity = 1.84 pH = 7.42 Hardness = 163 to IBM Watson
Published Temperature = 25 C Turbidity = 0.18 pH = 6.99 Hardness = 224 to IBM Watson
Published Temperature = 31 C Turbidity = 4.67 pH = 8.34 Hardness = 167 to IBM Watson
Published Temperature = 35 C Turbidity = 3.87 pH = 8.23 Hardness = 132 to IBM Watson
Published Temperature = 40 C Turbidity = 1.15 pH = 8.02 Hardness = 128 to IBM Watson
Published Temperature = 33 C Turbidity = 2.77 pH = 6.73 Hardness = 98 to IBM Watson
Published Temperature = 39 C Turbidity = 2.39 pH = 7.88 Hardness = 187 to IBM Watson
Published Temperature = 34 C Turbidity = 0.55 pH = 7.13 Hardness = 105 to IBM Watson
Published Temperature = 40 C Turbidity = 4.05 pH = 8.0 Hardness = 282 to IBM Watson
Published Temperature = 23 C Turbidity = 1.91 pH = 6.96 Hardness = 148 to IBM Watson
Published Temperature = 32 C Turbidity = 1.51 pH = 7.12 Hardness = 195 to IBM Watson
Published Temperature = 38 C Turbidity = 0.71 pH = 8.27 Hardness = 132 to IBM Watson
Published Temperature = 40 C Turbidity = 0.26 pH = 8.43 Hardness = 240 to IBM Watson
Published Temperature = 32 C Turbidity = 1.77 pH = 8.48 Hardness = 166 to IBM Watson
Published Temperature = 36 C Turbidity = 3.57 pH = 6.84 Hardness = 289 to IBM Watson
Published Temperature = 20 C Turbidity = 1.12 pH = 7.57 Hardness = 235 to IBM Watson
```

- **Publishing Data to Cloud:**
- These values of the keys in the dictionary are sent to IBM Watson with event name “IoTSensor” in json format.
- After publishing sensor data from python to IBM Watson Cloud platform, the output is shown that it is received.

IBM Watson IoT Platform

Browse

Action

Device Types

Interfaces

123

Connected

raspberrypi

Device

Nov 2, 2022 11:33 AM

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":20,"Turbidity":1.12,"pH":7.57,"Hardnes...	json	a few seconds ago
IoTSensor	{"temp":36,"Turbidity":3.57,"pH":6.84,"Hardnes...	json	a few seconds ago
IoTSensor	{"temp":32,"Turbidity":1.77,"pH":8.48,"Hardnes...	json	a few seconds ago
IoTSensor	{"temp":40,"Turbidity":0.26,"pH":8.43,"Hardnes...	json	a few seconds ago
IoTSensor	{"temp":38,"Turbidity":0.71,"pH":8.27,"Hardnes...	json	a few seconds ago