

Project Development Phase
Sprint-3

Date	18thNovember2022
Team ID	PNT2022TMID38378
Project Name	Project: Signs with Smart Connectivity for Better Road Safety.
Marks	20 Marks

Sprint	functional requirement	User Story Number	User Story/Task	Story Points	Priority	Team members
Sprint-3		US-1	Develop a python script to publish random sensordata such as temperate, humidity, and visibility to the IBM IoT Platform.	7	High	R. Janani S. Sowmiyaa E. Anandhi K. Jenifer K. Pavithra
Sprint-3		US-2	After developing the python code, commands are received and the statement representing the devices' control are printed.	5	Medium	R. Janani S. Sowmiyaa E. Anandhi K. Jenifer K. Pavithra
Sprint-3		US-3	Publish Data tothe IBM cloud.	8	High	R. Janani S. Sowmiyaa E. Anandhi K. Jenifer K. Pavithra

US-

1Develop a python script to publish random sensor data such as temperature, humidity, and visibility to the IBM IoT Platform

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

#Provide your IBM Watson Device Credentials

```
organization="33lnun"
deviceType="PNT2022TMID38378" deviceId
="PNT2022TMID38378"
authMethod="token" authToken=
"F*IPB)eaQ5+CqaU(Gj"
```

#Intialize GPIO

```
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command']) status=cmd.data['command']
    if status=="light on": print
        ("led is on")
    else:
        print("led 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 data point "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)
```

```
visi=random.randint(0,100)
```

```
data={'temperature'=temp,'humidity'=humid,'visibility'=visi}
```

```
#printdata
```

```
defmyOnPublishCallback(): print("Publishedtemperature=%sC"%temp,"humidity=%s%% "%  
% humid,"visibility=%s%% "%visi,"IoTBMWatson")
```

```
success=deviceCli.publishEvent("IoTSensor","json",data,qos=0,on_publish=my  
OnPublishCallback)
```

```
ifnotsuccess:
```

```
print("NotconnectedtoIoTF")time.sleep(1)
```

```
deviceCli.commandCallback=myCommandCallback
```

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

```
deviceCli.disconnect()
```