

Project Development Phase Sprint-3

Date				14th November 2022		
Team ID				PNT2022TMID13653		
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 sensor data such as temperature, humidity, visibility to the IBM IoT platform.	7	High	P.Yogiramji V.Thamu G.Selvam T.Prakash raja
Sprint-3		US-2	After developing python code, commands are received print the statements which represent the control of the devices.	5	Medium	P.Yogiramji V.Thamu G.Selvam T.Prakash raja
Sprint-3		US-3	Publish Data to the IBM Cloud.	8	High	P.Yogiramji V.Thamu G.Selvam T.Prakash raja

US-1 Develop 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 = "PNT2022TMID47485"
deviceId = "PNT2022TMID47485"
authMethod = "token"
authToken = "BGM(9-Tgfy&lrHmgIp"
```

#Intialize GPIO

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

#print(cmd)

```
try:
    deviceOptions = {"org": organization,"type":
deviceType,"id":deviceId,"authmethod":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)    visi=random.randint(0,100)
```

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

#print data

```
def myOnPublishCallback():
    print("Published temperature=%s C" %temp,"humidity =%s %"
%humid,"visibility =%s %" %visi,"to IBM Watson")
```

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

```
    if not success:
        print("Not connected to IoT")
time.sleep(1)
```

```
deviceCli.commandCallback= myCommandCallback
```

#Disconnect the device and application from the cloud

```
deviceCli.disconnect(
    )
```