CODING AND TESTING

# **SPRINT-3**

|  |  |
| --- | --- |
| DATE | 19-11-2022 |
| TEAM ID | PNT2022TMID26841 |
| PROJECT NAME | Hazardous area monitoring for industrial power plants powered by IoT |

**ALGORITHM:**

1. Start
2. Import 3 modules
3. Create the IBM IoT platform device
4. Give device id
5. Connect the device
6. Introducing my command call back function
7. Get a random temperature and humidity values
8. Loop infinitely
9. Print the random temperature and humidity values on console
10. Publish the values to IBM Watson IoT platform
11. Stop

# **PYTHON CODE:**

#connecting the python to IBM watson IoT platform import wiotp.sdk.device

import time import random myconfig = {

"identity":{ "orgId":"zvvqaf", "typeId":"IoT\_devices", "deviceId":"12345"

},

"auth":{ "token":"qagOTm?(qV+deBQ\*j\*"

}

}

def myCommandCallback(cmd):

print("Message received from IBM IoT platform: %s" % cmd.data['command']) m=cmd.data['command']

if(m=="lighton"):

print("\*\*\*\*\*////LIGHTS ARE ON/////\*\*\*\*\*") elif(m=="lightoff"):

print("\*\*\*\*\*/////LIGHTS ARE OFF/////\*\*\*\*\*") else:

print("\*\*\*\*////WRONG COMMAND////\*\*\*\*\*")

client = wiotp.sdk.device.DeviceClient(config=myconfig, logHandlers=None) client.connect()

while True:

temp=random.randint(-20,125) hum=random.randint(0,100) myData={'temperature':temp, 'humidity':hum}

client.publishEvent(eventId="status", msgFormat="json",data=myData,qos=0,onPublish=None) print("Published data Successfully: %s",myData)

client.commandCallback =myCommandCallback time.sleep(2)

client.disconnect()

**Preconditions:**

User has Device in IBM Watson IoT platform, Node-RED app