PROJECT DEVELOPMENT - DELIVERY OF SPRINT 2

Date:	09 November 2022
Team ID:	PNT2022TMID00966
Name:	Real-Time River Water Quality Monitoring and Control System

PYTHON CODE

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



#Provide your IBM Watson Device
Credentialsorganization = "udjkcs"
deviceType = " Water_quality_monitoring"
deviceId = '
Temperature_Monitoring"
authMethod = "token"
authToken =
"APTPlwZ&Wtb75ga@5&"

# Initialize GPIO


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

    command")#print(cmd)
```

```python
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(0,1
    00)
    pH=random.randint(0,14)

    data = { 'temp' : temp, 'Humid': Humid ,'pH' :
    pH }#print data
    def myOnPublishCallback():
       print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % Humid,
       "pHValue =
%s" % pH,  "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
clouddeviceCli.disconnect()
```

```
Python 3.9.8 (tags/v3.9.8:bb3fdcf, Nov  5 2021, 20:48:33) [MSC v.1929 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
============== RESTART: C:\Users\skcoder\OneDrive\Documents\wiotp.py =============
2022-11-02 12:19:40,909   wiotp.sdk.device.client.DeviceClient  INFO    Connecte
d successfully: d:eoic67:testdevicetype:123456
Published data Successfully: %s {'temperature': 75, 'humidity': 28}
Published data Successfully: %s {'temperature': 58, 'humidity': 7}
Published data Successfully: %s {'temperature': 78, 'humidity': 80}
Published data Successfully: %s {'temperature': 13, 'humidity': 10}
Published data Successfully: %s {'temperature': 11, 'humidity': 10}
Published data Successfully: %s {'temperature': -18, 'humidity': 60}
Published data Successfully: %s {'temperature': 61, 'humidity': 47}
Published data Successfully: %s {'temperature': 39, 'humidity': 36}
Published data Successfully: %s {'temperature': 107, 'humidity': 33}
Published data Successfully: %s {'temperature': 74, 'humidity': 87}
Published data Successfully: %s {'temperature': 30, 'humidity': 40}
Published data Successfully: %s {'temperature': -8, 'humidity': 76}
Published data Successfully: %s {'temperature': 70, 'humidity': 17}
Published data Successfully: %s {'temperature': 51, 'humidity': 73}
Published data Successfully: %s {'temperature': 83, 'humidity': 95}
Published data Successfully: %s {'temperature': 124, 'humidity': 75}
Published data Successfully: %s {'temperature': 111, 'humidity': 53}
Published data Successfully: %s {'temperature': 70, 'humidity': 30}
Published data Successfully: %s {'temperature': 72, 'humidity': 38}
Published data Successfully: %s {'temperature': 93, 'humidity': 25}
Published data Successfully: %s {'temperature': 10, 'humidity': 34}
```