

PROJECT TITLE : REAL TIME RIVER WATER QUALITY MONITORING AND CONTROL SYSTEM

TEAM ID : PNT2022TMID27339

PROJECT CODE

```
import time
```

```
import sys
```

```
import ibmiotf.application
```

```
import ibmiotf.device
```

```
import random
```

```
# Provide your IBM Watson Device Credentials
```

```
organization = "f7du5q"
```

```
deviceType = "RaspberryPi"
```

```
deviceId = "979"
```

```
authMethod = "token"
```

```
authToken = "12345678"
```

```
# Initialize GPIO
```

```
def myCommandCallback(cmd):
```

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

```
    status = cmd.data['command']
```

```
    if status == "motoron":
```

```
        print("Motor is on")
```

```
    else:
```

```
        print("Motor 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(0, 100)
```

```
    pH = random.randint(0, 14)
```

```
    turbidity = random.randint(0,1000)
```

```
    data = {'temperature': temp, 'pH': pH, 'turbidity':turbidity}
```

```
    # print data
```

```
    def myOnPublishCallback():
```

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

```

if ((temp > 70)&(pH>6)&(turbidity>500)):
    import requests

    url = 'https://www.fast2sms.com/dev/bulkV2'
    message = 'water quality is poor'
    numbers = 9790828557
    payload = f'sender_id =
FastSM&message={message}&route=v3&language=english&numbers={numbers}'
    headers = {
        'authorization':
'S4iYQnRsA8kMj0GOvKJTux3WE6czyewV5NCIDPoXmUdaLp9bHBm0ItCj5UhK7DSds9LkWPEF1RwVYO
le',
        'content-Type': 'application/x-www-form-urlencoded'
    }
    response = requests.request("POST", url=url, data=payload, headers=headers)
    print(response.text)

    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()

```