

FINAL CODE

DATE	17 NOVEMBER 2022
TEAM ID	PNT2022TMID41673
PROJECT NAME	REAL TIME RIVER WATER QUALITY MONITORING AND CONTROL SYSTEM

CODE:

#importing Random function

to generate the value and
required IoT and System

Libraries

import random as rand

import time

import ibmiotf.application

import ibmiotf.device

import sys

from clrprint import *

#defining credentials of device

```
organization = "sfq80i"
```

```
deviceType = "IoTDevice"
```

```
deviceId =
```

```
"PNT2022TMID30308"
```

```
authMethod = "token"
```

```
authToken =
```

```
"dAdKBdtr*Er(mud*0x"
```

```
def motorON():
```

```
    clrprint("\nMotors Turned  
ON",clr='r')
```

```
def motorOFF():
```

```
    clrprint("\nMotors Turned  
OFF",clr='r')
```

```
    #time.sleep(0)
```

```
# Initialize GPIO
```

```
# code to activate the motor  
comes here in Sprint 4
```

```
def
```

```
myCommandCallback(cmd):
```

```
    # Command Call back
```

```
    clrprint("\nCommand  
received: %s" %  
cmd.data['command'],clr='r')
```

```
    if(cmd.data['command'] ==  
"Motor On"):
```

```
        motorON()
```

```
    elif(cmd.data['command']  
== "Motor Off"):
```

```
        motorOFF()
```

```
    else:
```

```
        clrprint("\nInvalid  
Command",clr='r')
```

```
try:
```

```
    deviceOptions = {"org" :  
organization, "type":  
deviceType, "id" : deviceId,  
"auth-method" : authMethod,  
"auth-token" : authToken}
```

```
    deviceCli =  
ibmiotf.device.Client(deviceO  
ptions)
```

```
except Exception as e:
```

```
    print("Caught exception  
connecting device: %s"  
%str(e))
```

```
    sys.exit()
```

```
deviceCli.connect()
```

```
while True:
```

```
    print("Welcome to Real-  
Time River Water Quality  
Monitoring and Control  
System")
```

```
    temperature =  
int(rand.randint(0,100))
```

```
    pH = int(rand.randint(0,14))
```

```
    DO =  
int(rand.randint(0,150))
```

```
    Turbidity =  
int(rand.randint(0,20))
```

```
    TSS =
```

```
int(rand.randint(0,3700))
```

```
Manganese =  
int(rand.randint(0,1000))
```

```
Copper =  
int(rand.randint(0,2000))
```

```
ammoniaNitrate =  
int(rand.randint(0,100))
```

```
Hardness =  
int(rand.randint(0,1000))
```

```
Zinc =  
int(rand.randint(0,100))
```

```
Conductivity =  
int(rand.randint(0,2000))
```

```
Chloride =  
int(rand.randint(0,200))
```

```
Sulphate =  
int(rand.randint(0,1000))
```

```
data = {"Temperature":  
temperature,
```

```
    "pH": pH,
```

```
    "DO": DO,
```

```
    "Turbidity" : Turbidity,
```

```
"TSS": TSS,  
  
"Manganese":  
Manganese,  
  
"Copper": Copper,  
  
"AmmoniaNitrate":ammoniaN  
itrate,  
  
"Hardness":Hardness,  
  
"Zinc": Zinc,  
  
"Conductivity":  
Conductivity,  
  
"Chloride": Chloride,  
  
"Sulphate": Sulphate  
  
}
```

```
#These variables store  
value of random data to be  
shared to the cloud
```

```
print(data)
```

```
#printing the values
```

```
def
```

```
myOnPublishCallback():
```

```
    print("Published all data  
to IBM Watson")
```

```
    success =  
deviceCli.publishEvent("IotSe  
nsor", "json", data, qos=0, on_pu  
blish=myOnPublishCallback)
```

```
    if not success:
```

```
        print("Not connected to  
IoT Device")
```

```
        time.sleep(20)
```

```
deviceCli.commandCallback =  
myCommandCallback
```

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

```
deviceCli.disconnect()
```