Real time river water quality monitoring and control system

SPRINT – 4

TITLE	Real time river water quality monitoring and control system
DOMAIN NAME	INTERNET OF THINGS
TEAM ID	PNT2022TMID20521

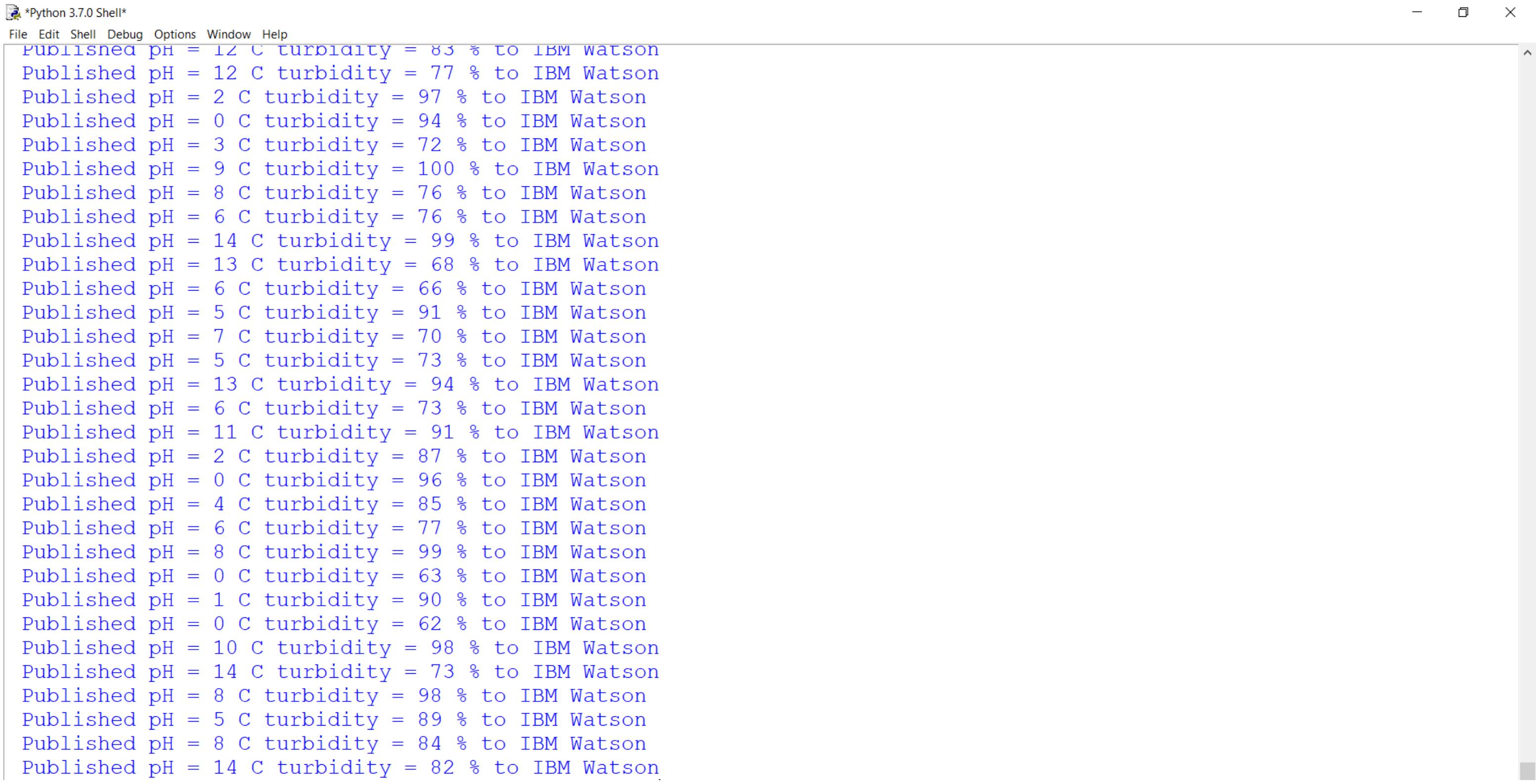
Receiving commands from IBM cloud using Python program

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


#Provide your IBM Watson Device Credentials
organization = "fwe3x0"
deviceType = "IOT_device"
deviceId = "1911010"
authMethod = "token"
authToken = "1911010abcdefgh"
```

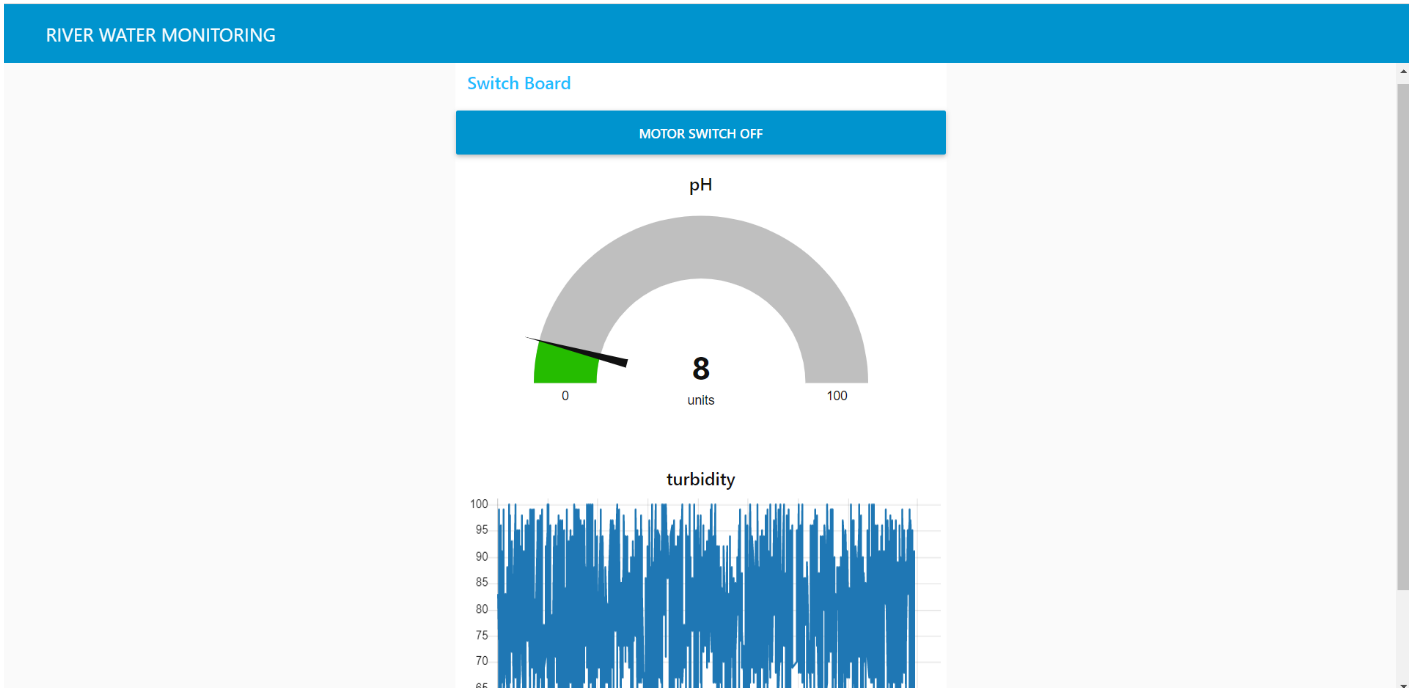
```
# Initialize GPIO


def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status == "motoron":
        print ("motor is on")
    elif status == "motor":
        print ("motor is off")
    else:
        print("Please send proper command")
    #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 random function

        pH=random.randint(0,14)
        turbidity=random.randint(60,100)
        data={'pH':pH,'turbidity':turbidity}
        #print data
        def myOnPublishCallback():
            print (" Published pH = %s " % pH, "turbidity = %s %" % turbidity,  "to IBM Watson")

        success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publish=myOnPublishCallback)
        if not success:
            print("\n Not connected to IoTF")
```

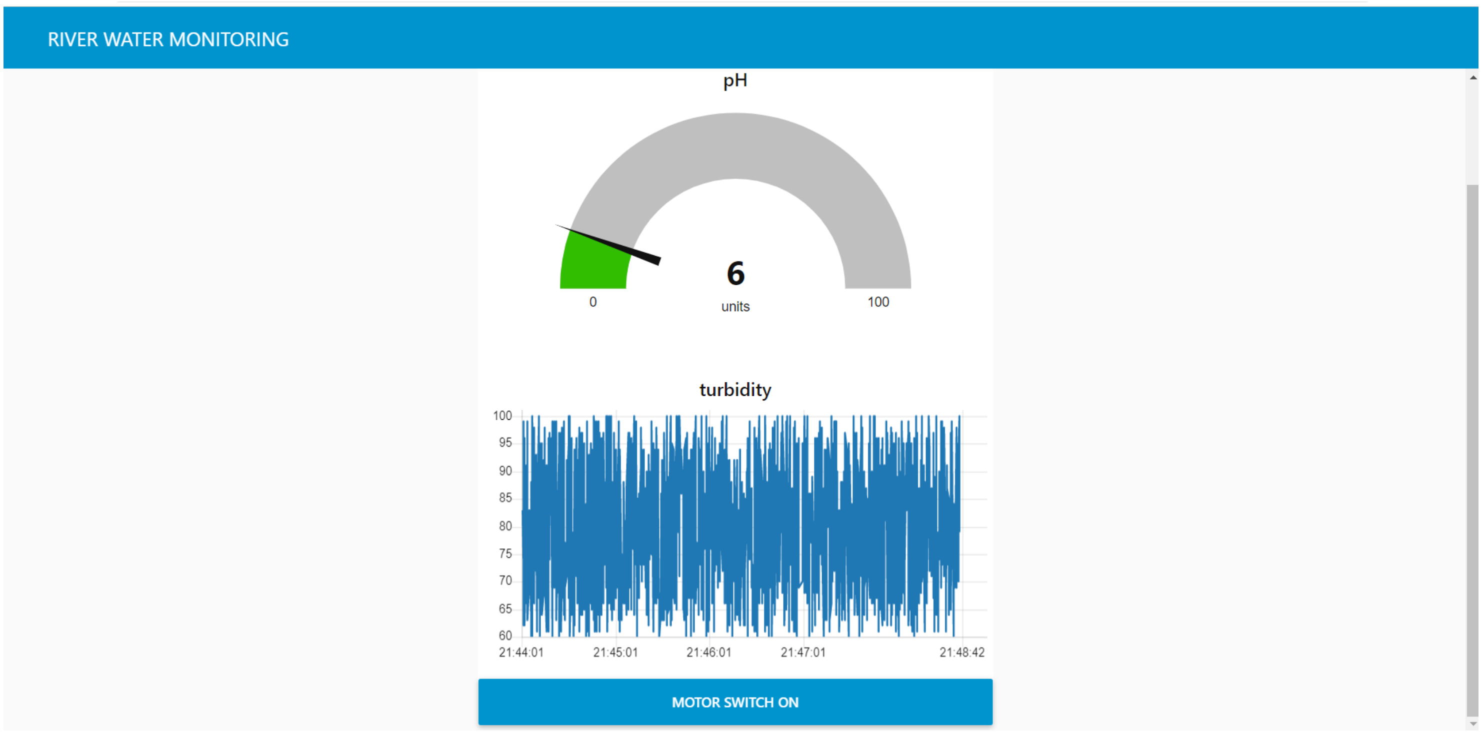
```python
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status == "motoron":
        print ("motor is on")
    elif status == "motor":
        print ("motor is off")
    else:
        print("Please send proper command")
    #print(cmd)



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 random function

        pH=random.randint(0,14)
        turbidity=random.randint(60,100)
        data={'pH':pH,'turbidity':turbidity}
        #print data
        def myOnPublishCallback():
            print (" Published pH = %s " % pH, "turbidity = %s %" % turbidity,  "to IBM Watson")

        success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publish=myOnPublishCallback)
        if not success:
            print("\n Not connected to IoTF")
```

y



```
Published pH = 12 C turbidity = 83 % to IBM Watson
Published pH = 12 C turbidity = 77 % to IBM Watson
Published pH = 2 C turbidity = 97 % to IBM Watson
Published pH = 0 C turbidity = 94 % to IBM Watson
Published pH = 3 C turbidity = 72 % to IBM Watson
Published pH = 9 C turbidity = 100 % to IBM Watson
Published pH = 8 C turbidity = 76 % to IBM Watson
Published pH = 6 C turbidity = 76 % to IBM Watson
Published pH = 14 C turbidity = 99 % to IBM Watson
Published pH = 13 C turbidity = 68 % to IBM Watson
Published pH = 6 C turbidity = 66 % to IBM Watson
Published pH = 5 C turbidity = 91 % to IBM Watson
Published pH = 7 C turbidity = 70 % to IBM Watson
Published pH = 5 C turbidity = 73 % to IBM Watson
Published pH = 13 C turbidity = 94 % to IBM Watson
Published pH = 6 C turbidity = 73 % to IBM Watson
Published pH = 11 C turbidity = 91 % to IBM Watson
Published pH = 2 C turbidity = 87 % to IBM Watson
Published pH = 0 C turbidity = 96 % to IBM Watson
Published pH = 4 C turbidity = 85 % to IBM Watson
Published pH = 6 C turbidity = 77 % to IBM Watson
Published pH = 8 C turbidity = 99 % to IBM Watson
Published pH = 0 C turbidity = 63 % to IBM Watson
Published pH = 1 C turbidity = 90 % to IBM Watson
Published pH = 0 C turbidity = 62 % to IBM Watson
Published pH = 10 C turbidity = 98 % to IBM Watson
Published pH = 14 C turbidity = 73 % to IBM Watson
Published pH = 8 C turbidity = 98 % to IBM Watson
Published pH = 5 C turbidity = 89 % to IBM Watson
Published pH = 8 C turbidity = 84 % to IBM Watson
Published pH = 14 C turbidity = 82 % to IBM Watson
```





Advantages & Disadvantages:

- River water can be monitored and controlled remotely.
- Increase in convenience to control by the government.

- Less labor cost.

- Better standards of living

Disadvantages:

- Lack of internet/connectivity issues.

- Added cost of internet and internet gateway infrastructure.

Conclusion

Thus the objective of the project to implement an IoT system in order to help farmers to control and monitor their farms has been implemented successfully.

Bibliography

IBM cloud reference: https://cloud.ibm.com/

IoT simulator :

https://watson-iot-sensor-simulator.mybluemix.net/