

REAL TIME RIVER WATER QUALITY MONITORING AND CONTROL SYSTEM USING IoT

Submitted by

SWATHI.A.P	(113219041120)
SOWMYA.A	(113219041114)
MADHUMITHA.S	(113219041060)
KOKILA.B	(113219041053)

**BACHELOR OF ENGINEERING IN
ELECTRONICS AND
COMMUNICATION DEPARTMENT**

WORK FLOW

Team ID	PNT2022TMID23523
Project Name	Real-time river water quality monitoring and control system

COMMUNICATION AMONG MIT APP, NODE-RED, IBM IOT WATSON AND PYTHON

Python code:

```
File Edit Format Run Options Window Help
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random

#Provide your IBM Watson Device Credentials
organization = "ks8pti"
deviceType = "ESP32"
deviceId = "143143"
authMethod = "token"
authToken = "123456789"

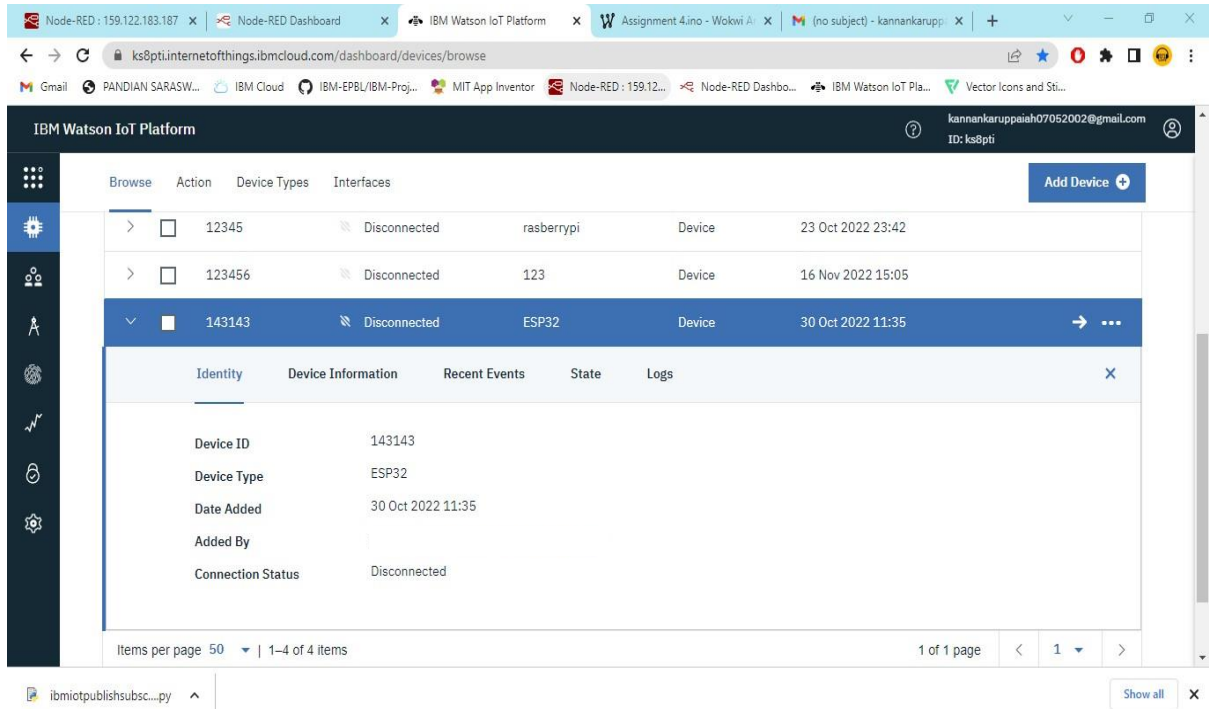
# Initialize GPIO

def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="START":
        print ("Motor is Started")
    elif status=="STOP":
        print ("Motor is OFF state")
    elif status=="LEFT":
        print ("Left Side is Closed")
    elif status=="RIGHT":
        print ("Right Side is Closed")
    elif status=="FORWARD":
        print ("Message is Forward to the chief")
    else :
        print ("Send a proper command")

    #print(cmd)

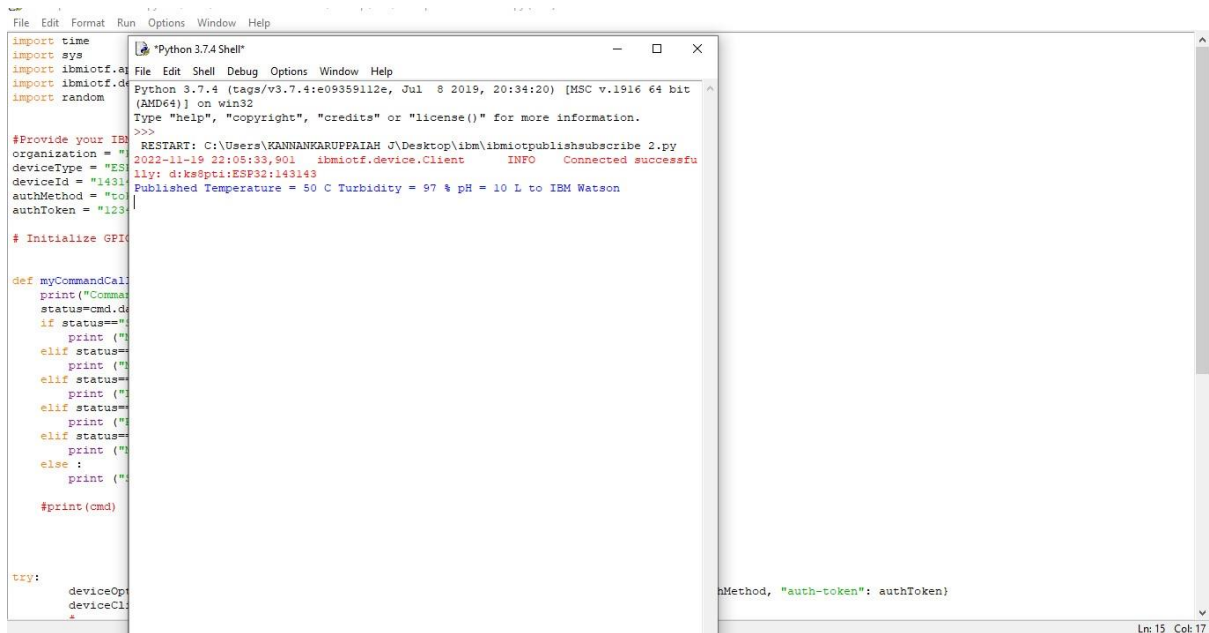
try:
    deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method": authMethod, "auth-token": authToken}
    deviceCli = ibmiotf.device.Client(deviceOptions)
```

Before running the python code, The IOT platform is disconnected

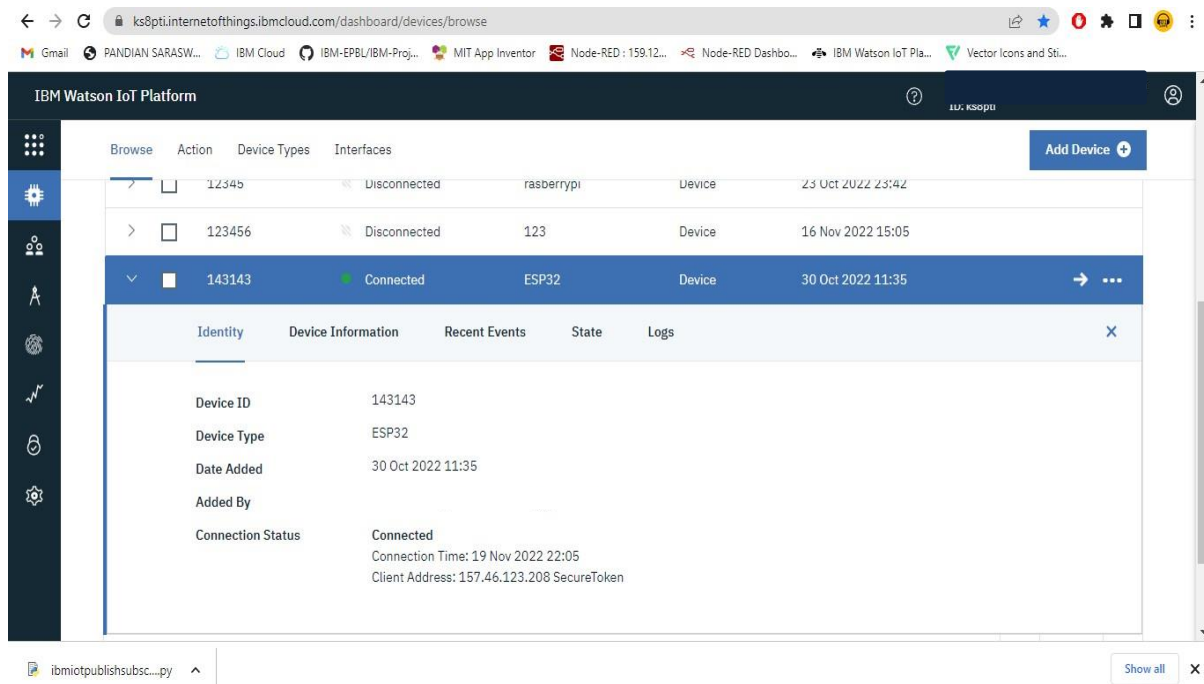


Run the Python code:

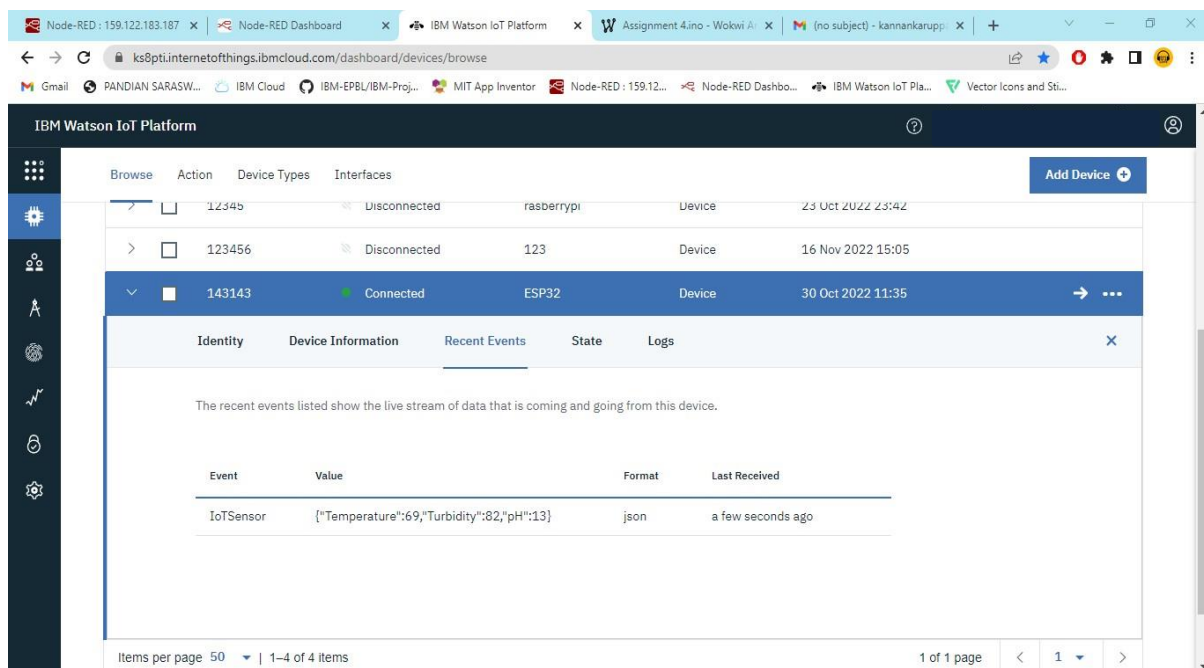
After running the python code the data's are shown in IDLE



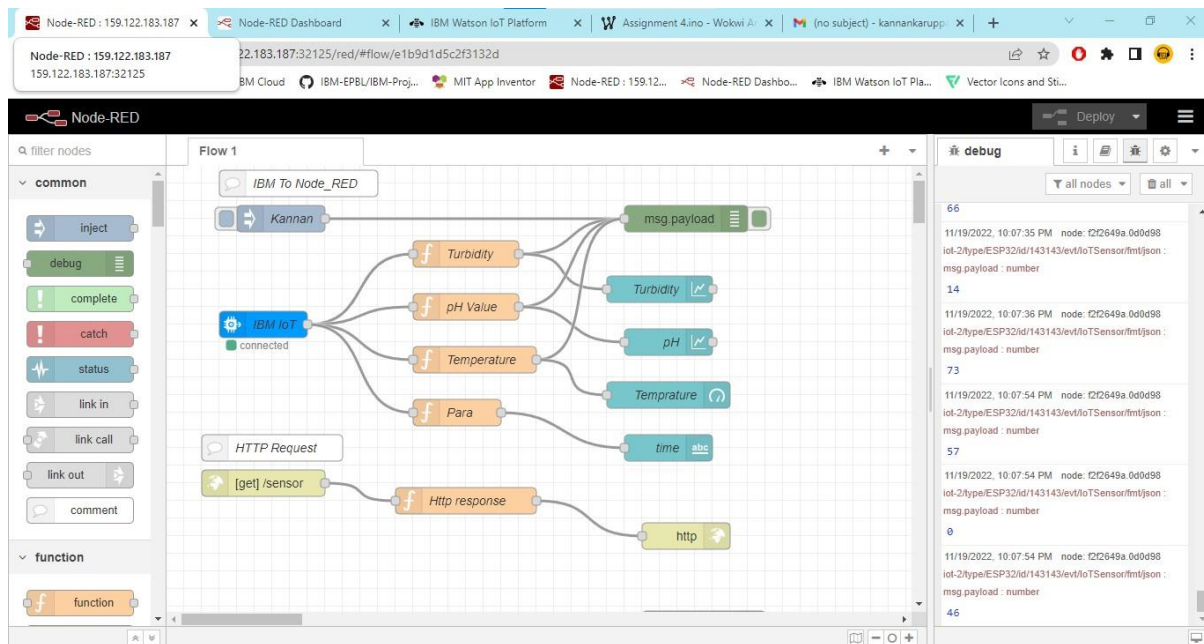
Now the IBM IoT Watson platform is connected



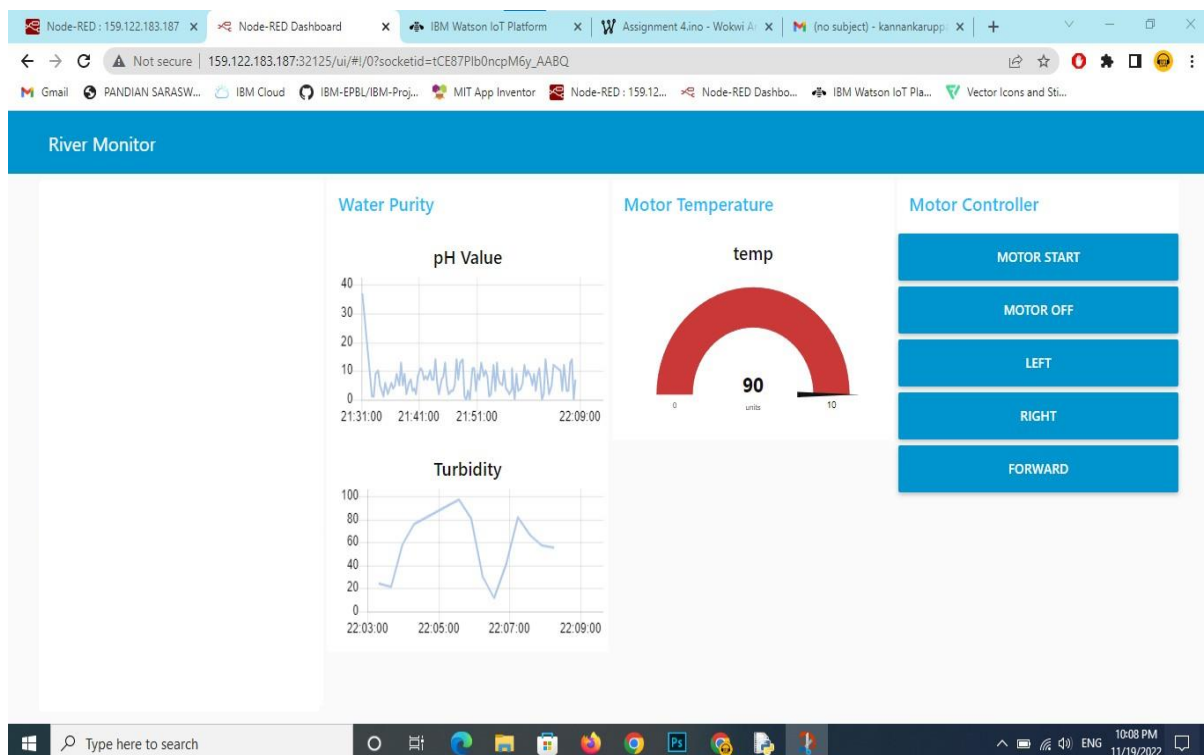
IoT Device ESP32 is connected with python code ,Then the data's are collected and shown in recent events.



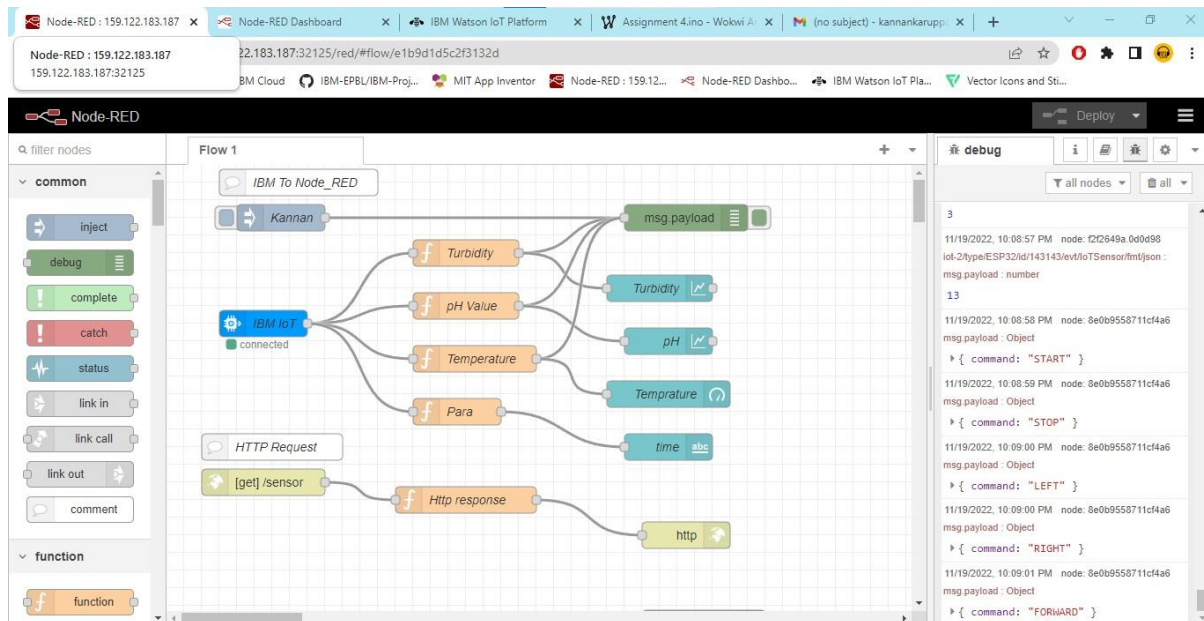
- The Node-RED is connected with the IBM IoTplatform .
- IBM IoT passes the data to the Node-Red.
- Node-RED collects all the data and display in debug window.



- Node-RED Dashboard is Showing all the data.
- When we click the buttons in dashboard the result will publish both Node-RED and Python



Node-RED OUTPUT



Python OUTPUT

The screenshot shows a Python 3.7.4 Shell window. The output of the script is as follows:

```
Python 3.7.4 (tags/v3.7.4:09359112e, Jul 8 2019, 20:34:20) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users\KANNANKARUPPAIAH J\Desktop\ibm\ibmiotpublishsubscribe 2.py
2022-11-19 22:05:33,901 ibmiotf.device.Client INFO Connected successfully: d:\k88pti:ESP32:143143
Published Temperature = 50 C Turbidity = 97 & pH = 10 L to IBM Watson
Published Temperature = 74 C Turbidity = 81 & pH = 3 L to IBM Watson
Published Temperature = 32 C Turbidity = 30 & pH = 12 L to IBM Watson
Published Temperature = 27 C Turbidity = 12 & pH = 3 L to IBM Watson
Published Temperature = 97 C Turbidity = 42 & pH = 3 L to IBM Watson
Published Temperature = 69 C Turbidity = 82 & pH = 13 L to IBM Watson
Published Temperature = 73 C Turbidity = 66 & pH = 14 L to IBM Watson
Published Temperature = 46 C Turbidity = 57 & pH = 0 L to IBM Watson
Published Temperature = 90 C Turbidity = 55 & pH = 7 L to IBM Watson
Published Temperature = 41 C Turbidity = 35 & pH = 2 L to IBM Watson
Published Temperature = 13 C Turbidity = 46 & pH = 3 L to IBM Watson
Command received: START
Motor is Started
Command received: STOP
Motor is off state
Command received: LEFT
Left Side is Closed
Command received: RIGHT
Right Side is Closed
Command received: FORWARD
Message is Forward to the chief
Published Temperature = 39 C Turbidity = 85 & pH = 1 L to IBM Watson
```

- This is my mobile app screen.
- Its show the pH and Turbidity values of water andtemperature of motor.
- When I'm clicking the control buttons in this screen the result are publish in Node-RED & Python.

10:09



Vo LTE 4G 89

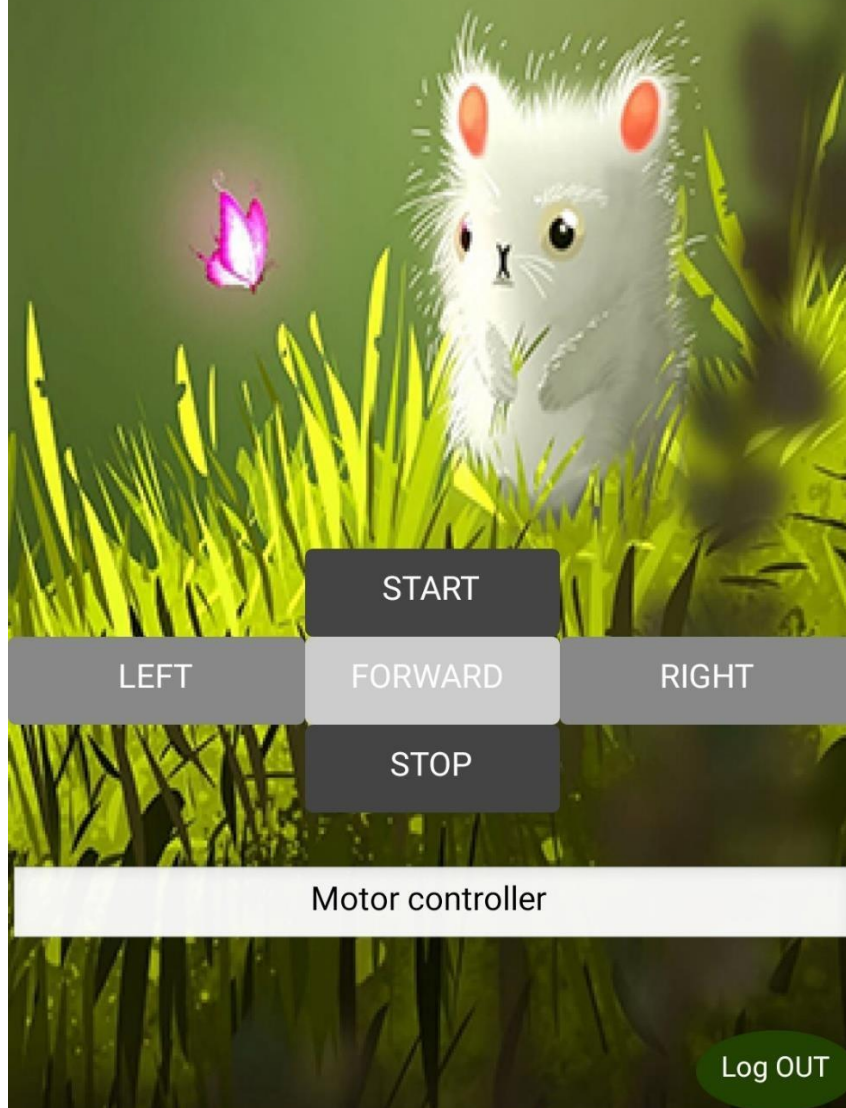
Real Time Water Quality Monitoring

MONITORING WINDOW

pH Value : 6

Turbidity : 55

Temperature: 13



START

LEFT

FORWARD

RIGHT

STOP

Motor controller

Log OUT



Python Output

The screenshot shows the Node-RED interface with a Python shell window open. The shell displays the output of a script that publishes sensor data to IBM Watson IoT. The output includes a list of sensor readings (Temperature, Turbidity, pH) and a series of commands received from the IoT platform (START, STOP, LEFT, RIGHT, FORWARD). The debug console on the right shows the messages being received by the Node-RED flow.

```
RESTART: C:\Users\KANNANKARUPPAIAH J\Desktop\ibmiotpublishsubscribe 2.py
2022-11-19 22:05:33,901 ibmiotf.device.Client INFO Connected successfully
11/19/2022, 10:09:58 PM node: 4fe0547e5fa53b21
msg.payload: number
52
11/19/2022, 10:09:58 PM node: 4fe0547e5fa53b21
msg.payload: Object
{ command: "START" }
11/19/2022, 10:09:58 PM node: 4fe0547e5fa53b21
msg.payload: Object
{ command: "FORWARD" }
11/19/2022, 10:09:58 PM node: 4fe0547e5fa53b21
msg.payload: Object
{ command: "LEFT" }
11/19/2022, 10:09:58 PM node: 4fe0547e5fa53b21
msg.payload: Object
{ command: "STOP" }
11/19/2022, 10:09:58 PM node: 4fe0547e5fa53b21
msg.payload: Object
{ command: "RIGHT" }
```

Node-RED

