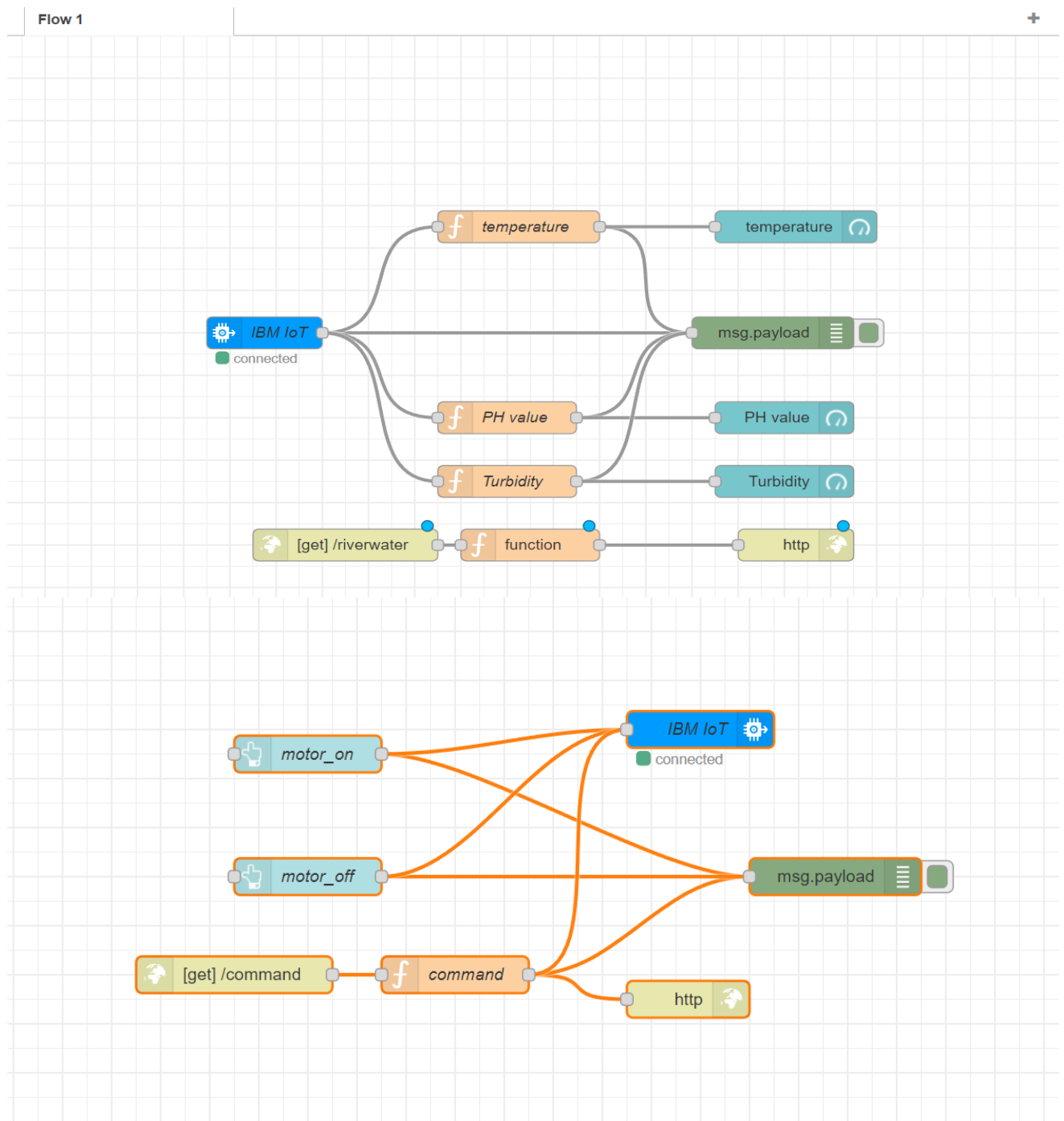


WEB APPLICATION USING NODE RED

Date	03 November 2022
Team ID	PNT2022TMID43374
Project Name	River Water Quality Monitoring and Control System

NODE RED HOME:



IBM IOT NODE:

The image shows the Node-RED web interface. On the left, the 'filter nodes' sidebar is open, showing various nodes. The 'input' section is expanded, and the 'ibmiot in' node is selected. In the main workspace, a flow named 'Flow 1' is visible. It starts with the 'ibmiot in' node, which is connected to a 'function' node. The 'function' node is connected to three output nodes: 'temperature', 'PH value', and 'Turbidity'. The 'function' node is also connected to a '[get] /riverwater' node. The 'Edit ibmiot in node' dialog is open on the right, showing the configuration for the 'ibmiot in' node. The 'Properties' section is expanded, showing the following settings:

- Authentication: API Key
- API Key: IBM API
- Input Type: Device Event
- Device Type: ☐ All or kprp
- Device Id: ☐ All or 2222
- Event: ☒ All or data
- Format: ☐ All or json
- QoS: 0
- Name: IBM IoT
- Service: registered

At the bottom of the dialog, there is a note: "Use the Input Type property to configure this node to receive Events sent by IoT Devices, Commands sent to IoT Devices, Status Messages referring to IoT Devices, or Status Messages referring to". Below the note, there is a checkbox labeled "Enabled".

SENSORS: TEMPERATURE:

The image shows the Node-RED web interface. On the left, the 'filter nodes' sidebar is open, showing various nodes. The 'sequence' section is expanded, and the 'split' node is selected. In the main workspace, a flow named 'Flow 1' is visible. It starts with the 'ibmiot in' node, which is connected to a 'function' node. The 'function' node is connected to three output nodes: 'temperature', 'PH value', and 'Turbidity'. The 'function' node is also connected to a '[get] /riverwater' node. The 'Edit function node' dialog is open on the right, showing the configuration for the 'function' node. The 'Properties' section is expanded, showing the following settings:

- Name: temperature

Below the 'Properties' section, there are four tabs: 'Setup', 'On Start', 'On Message', and 'On Stop'. The 'On Message' tab is selected, showing the following code:

```
1 global.set('Temperature',msg.payload.Temperature)
2 msg.payload=msg.payload.Temperature
3 return msg;
```

PH SENSOR:

Flow 1

IBM IoT
connected

[get] /riverwater

Edit function node

Delete Cancel Done

Properties

Name PH value

Setup On Start On Message On Stop

```
1 global.set('pH',msg.payload.pH)
2 msg.payload=msg.payload.pH
3 return msg;
```

TURBIDITY:

Flow 1

IBM IoT
connected

[get] /riverwater

Edit function node

Delete Cancel Done

Properties

Name Turbidity

Setup On Start On Message On Stop

```
1 global.set('Turbidity',msg.payload.Turbidity)
2 msg.payload=msg.payload.Turbidity
3 return msg;
```

HTTP NODE:

Flow 1

```
graph LR; IoT[IBM IoT] --> temp[f temperature]; IoT --> PH[f PH value]; IoT --> turb[f Turbidity]; get[/riverwater/] --> func[f function];
```

Edit http in node

Delete

Cancel

Done

Properties

Method

GET

URL

/riverwater

Name

Name

Enabled

Node-RED

Flow 1

```
graph LR; IoT[IBM IoT] --> temp[f temperature]; IoT --> PH[f PH value]; IoT --> turb[f Turbidity]; get[/riverwater/] --> func[f function];
```

Edit function node

Delete

Cancel

Done

Properties

Name

Name

Setup

On Start

On Message

On Stop

1 msg.payload={ 'Temperature':global.get("Temperature"), 'pH':global.get(

2 return msg;

Enabled

WEBSITE:

