

Build a Web Application Using Node-RED

Date	17 November 2022
Team ID	PNT2022TMID04463
Project Name	Project – Smart Farmer-IoT Enabled smart Farming Application

Configuration of Node-Red to send commands to IBM cloud

ibmiot out node I used to send data from Node-Red to IBM Watson device. So, after adding it to the flow we need to configure it with credentials of our Watson device.

The screenshot shows the Node-RED web interface in a browser window. The address bar displays the URL: `ed/#flow/c7ddb1462b8a000c`. The main workspace shows a configuration panel for the 'ibmiot out' node, titled 'Edit ibmiot in node'. The panel includes a 'Delete' button, a 'Cancel' button, and a red 'Done' button. Below these buttons is a 'Properties' section with various configuration options:

- Authentication:** A dropdown menu set to 'API Key'.
- API Key:** A text input field containing 'IBMIOT APIKEY'.
- Input Type:** A dropdown menu set to 'Device Event'.
- Device Type:** A checkbox labeled 'All or' followed by a text input field containing 'abcd'.
- Device Id:** A checkbox labeled 'All or' followed by a text input field containing '7654321'.
- Event:** A checkbox labeled 'All or' followed by a text input field containing '+'.
- Format:** A checkbox labeled 'All or' followed by a text input field containing 'json'.
- QoS:** A dropdown menu set to '0'.
- Name:** A text input field containing 'IBM IoT'.

At the bottom of the configuration panel, there is a checkbox labeled 'Enabled' which is currently unchecked. The Windows taskbar is visible at the bottom of the screen, showing icons for PowerShell, File Explorer, Settings, and other applications, along with the system clock and taskbar search bar.

Here we add two buttons in UI

1 -> for motor on

2 -> for motor off

We used a function node to analyses the data received and assign command to each number.

The Java script code for the analyses is:

```
if(msg.payload===1)
```

```
msg.payload={"command": "ON"};
```

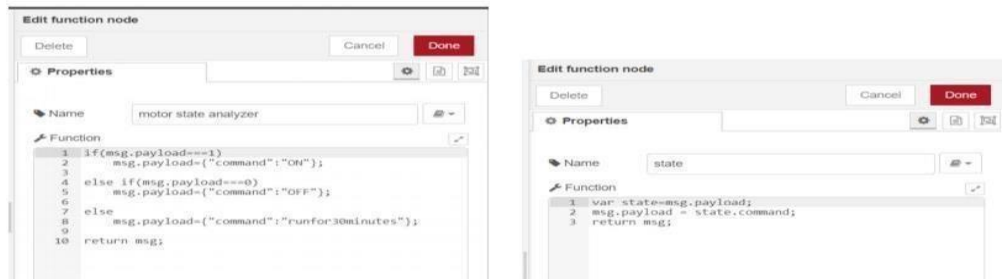
```
else if(msg.payload===0)
```

```
msg.payload={"command": "OFF"};
```

Then we use another function node to parse the data and get the command and represent it visually with text node.

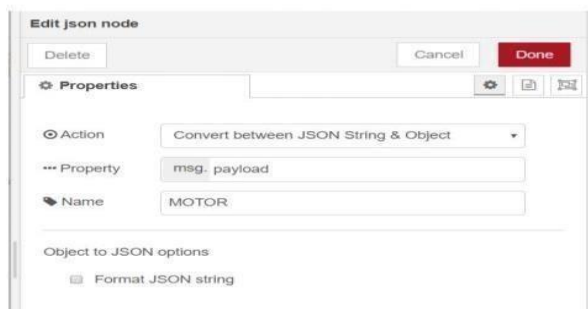
The Java script code for that function node is:

```
var state=msg.payload;  
msg.payload = state.command;  
return msg;
```

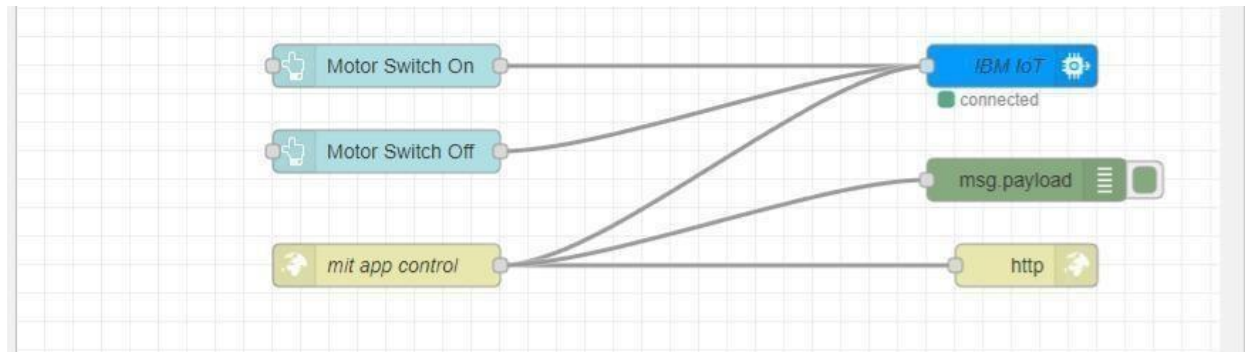


The above images show the java script codes of analyser and state function nodes.

Then we add edit Json node to the conversion between JSON string & object and finally connect it to IBM IoT Out.



Edit JSON node needs to be configured like this



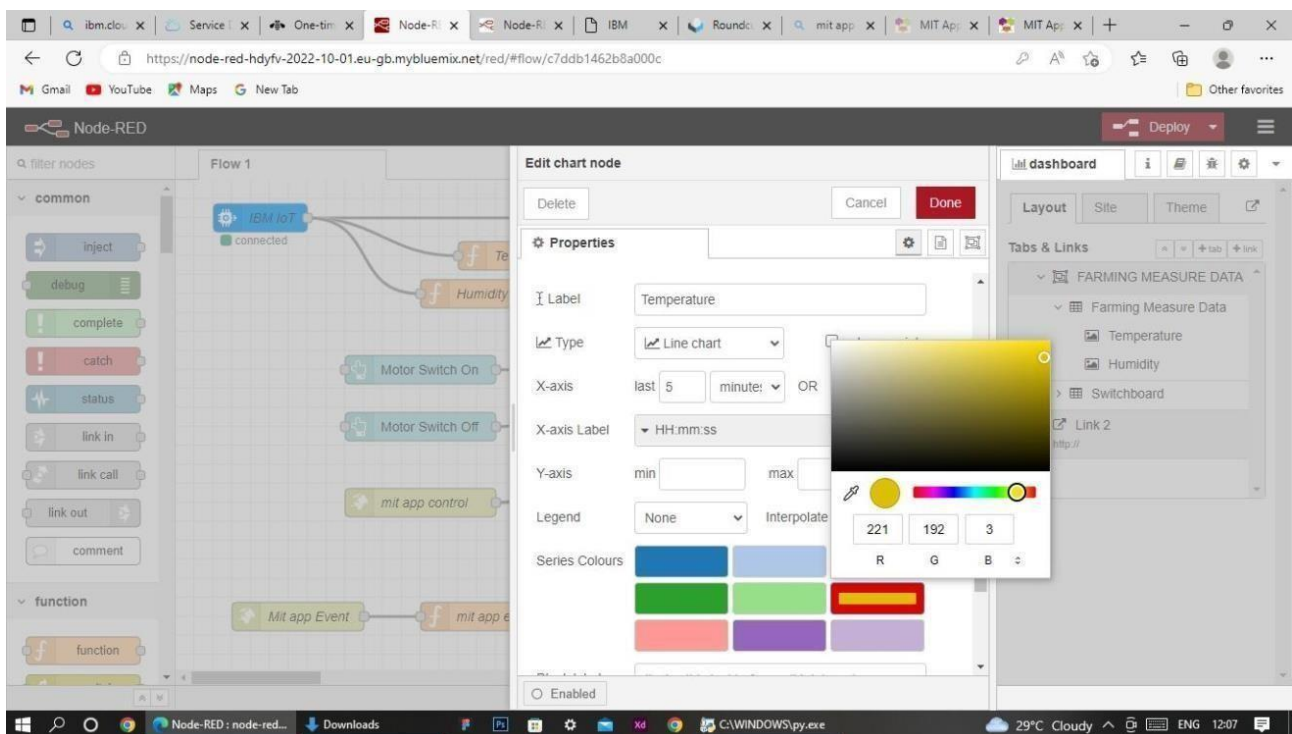
This is the program flow for sending commands to IBM cloud.

Adjusting User Interface

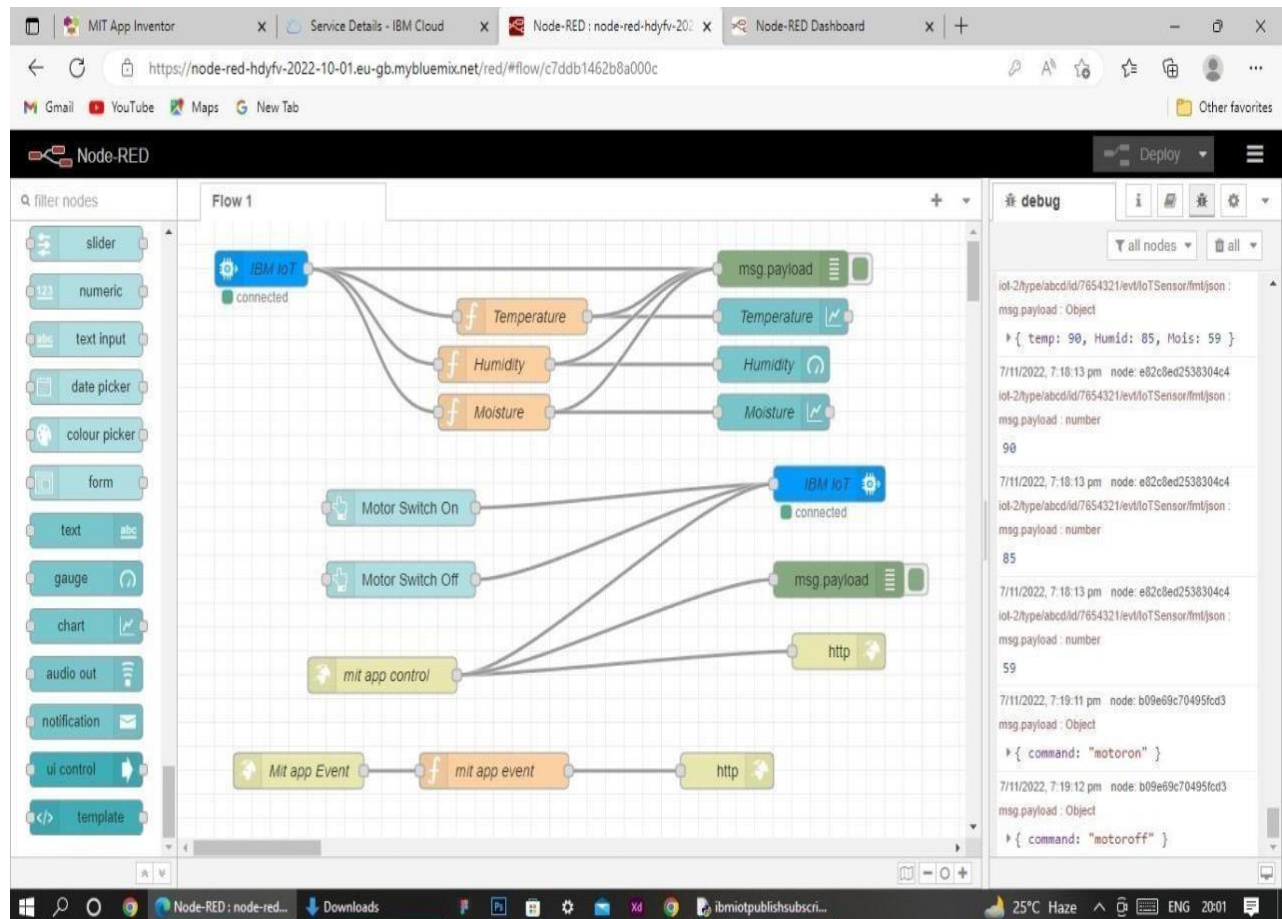
In order to display the parsed JSON data a Node-Red dashboard is created

Here we are using Gauges, text and button nodes to display in the UI and helps to monitor the parameters and control the farm equipment.

Below images are the Gauge, text and button node configurations.



Complete Program Flow



Web APP UI Home Tab

