

## Build a Web Application Using Node-RED

Date	17 November 2022
Team ID	PNT2022TMID09732
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 with a browser tab titled 'Node-RED : node-red-hdyfv-202'. The address bar shows the URL 'ed/#flow/c7ddb1462b8a000c'. The main workspace displays the 'Edit ibmiot in node' configuration panel. At the top of this panel are 'Delete', 'Cancel', and 'Done' buttons. Below is the 'Properties' section with various settings:

- Authentication:** API Key (dropdown menu)
- API Key:** IBMIOT APIKEY (text input with a search icon)
- Input Type:** Device Event (dropdown menu)
- Device Type:** ☐ All or abcd (text input)
- Device Id:** ☐ All or 7654321 (text input)
- Event:** ☒ All or + (text input)
- Format:** ☐ All or json (text input)
- QoS:** 0 (dropdown menu)
- Name:** IBM IoT (text input)

At the bottom of the configuration panel is an 'Enabled' checkbox, which is currently unchecked. The Windows taskbar at the bottom shows icons for PowerShell, Task View, Settings, Mail, Visual Studio Code, Chrome, and a file explorer window titled 'C:\WINDOWS\py.exe'.

Here we add two buttons in UI

1 -> for motor on

2 -> for motor off

We used a function node to analyse 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"};
```

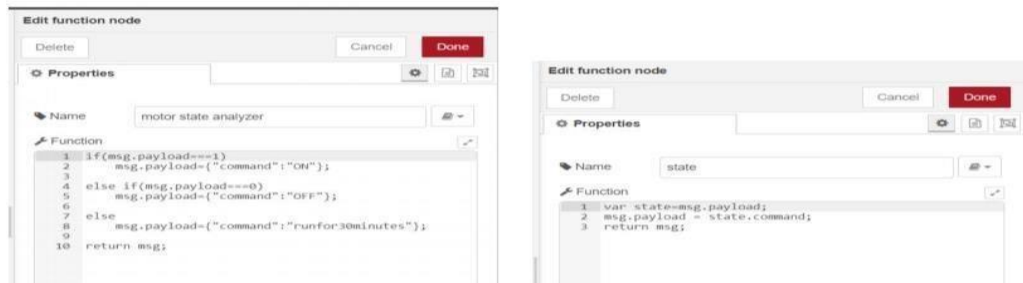
```
else if(msg.payload===0)
```

```
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.

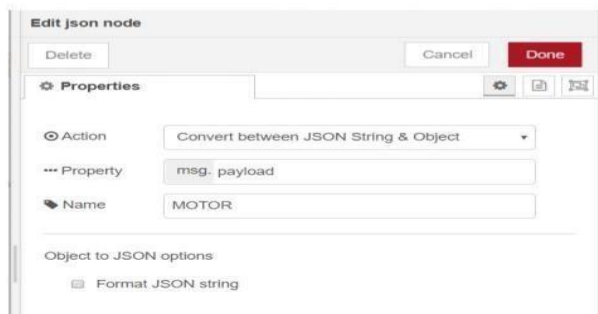
The Java script code for that function node is:

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

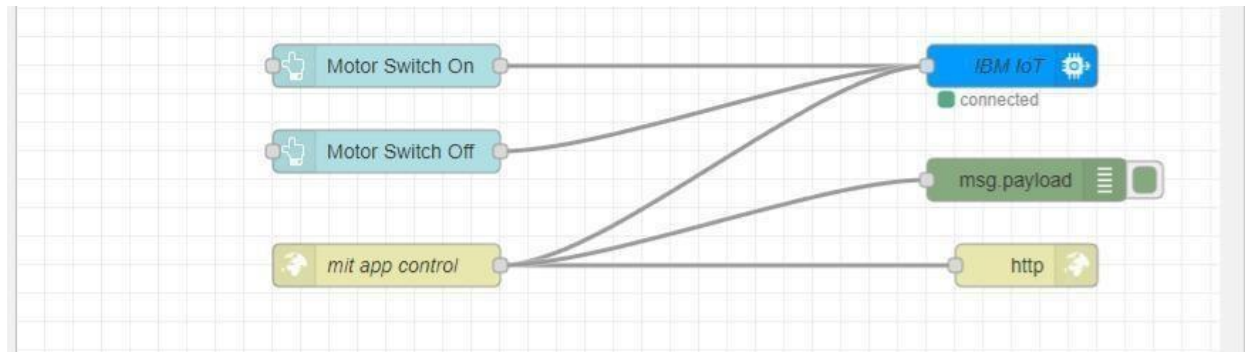


The above images show the java script codes of analyser and state function nodes.

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



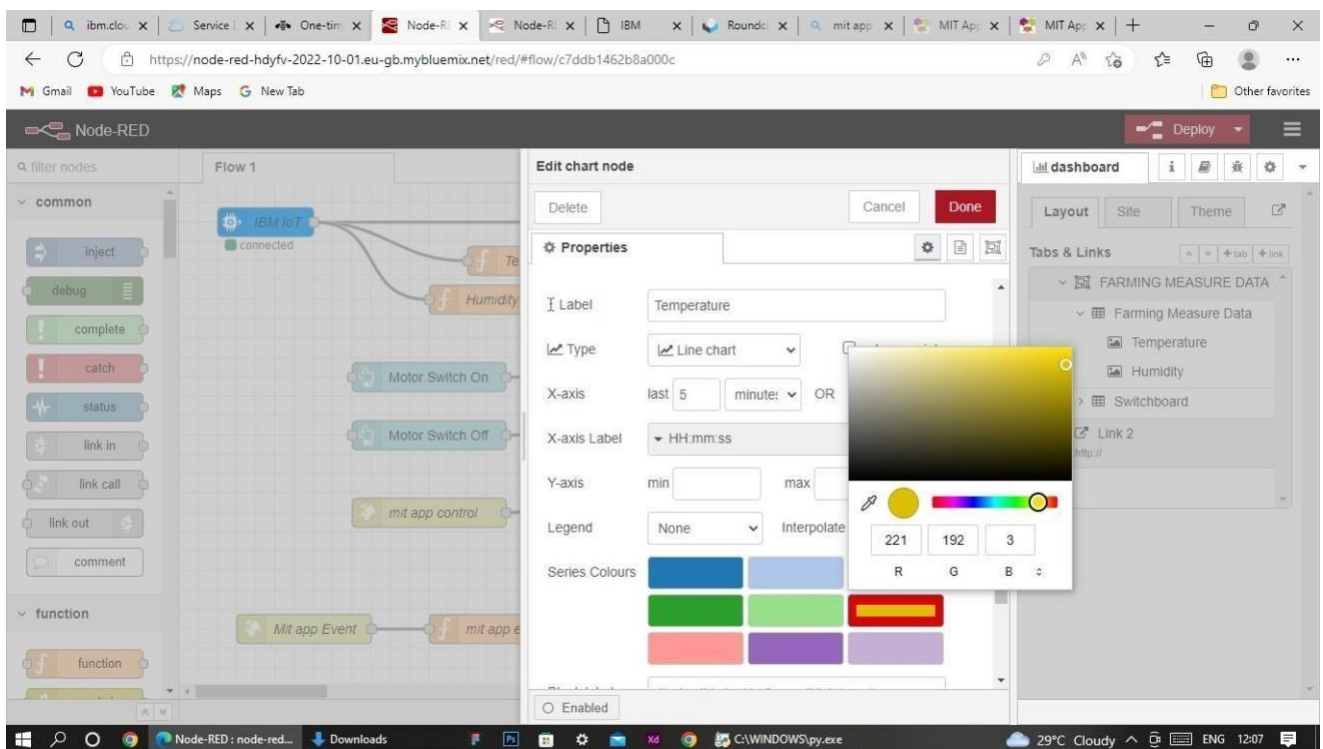
This is the program flow for sending commands to IBM cloud.

## Adjusting User Interface

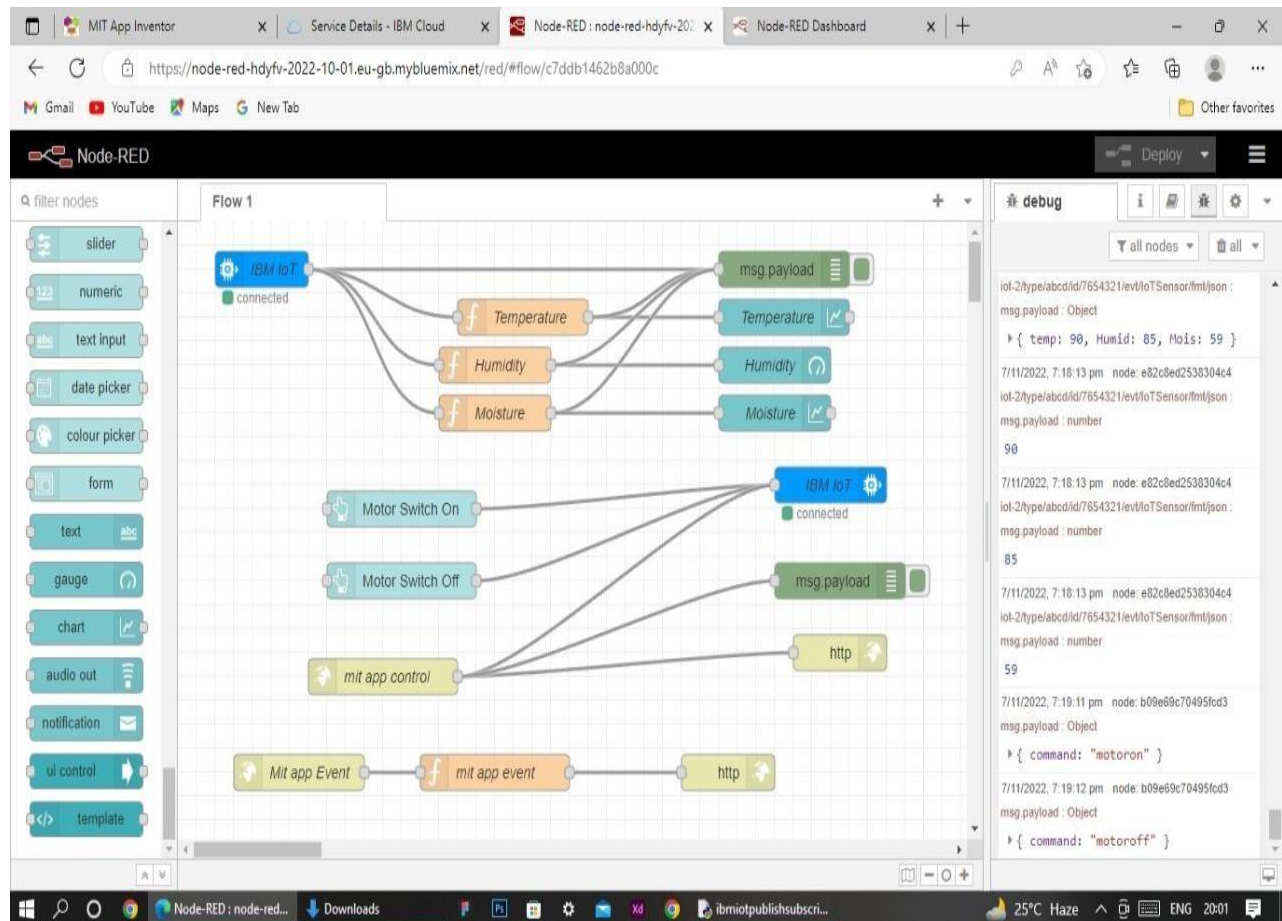
In order to display the parsed JSON data a Node-Red dashboard is created

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

