

TEAM
ID:PNT2022TMD47947

Smart farm -IOT Enabled Smart Farming Application

IBM NALAIYATHIRAN

SPRINT DELIVERY - 3

Configuration of Node-Red to send command to IBM cloud

inherit node I used to send data from Node-Red to IBM Watson device. So, after it to the flow we need to configure it with credentials on Watson device,

Here we add two buttons in UI

The screenshot shows the 'Edit ibmiot in node' configuration window in Node-Red. At the top, there are 'Delete', 'Cancel', and 'Done' buttons. Below is a 'Properties' section with various settings:

- Authentication:** A dropdown menu set to 'API Key'.
- API Key:** A text input field containing 'IBMIOT APIKEY' with a search icon on the left and an edit icon on the right.
- Input Type:** A dropdown menu set to 'Device Event'.
- Device Type:** A label 'All or' followed by a text input field containing 'abod'.
- Device Id:** A label 'All or' followed by a text input field containing '7654321'.
- Event:** A label 'All or' followed by a text input field containing '+'.
- Format:** A label '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 left, there is an 'Enabled' checkbox which is currently checked.

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;
```

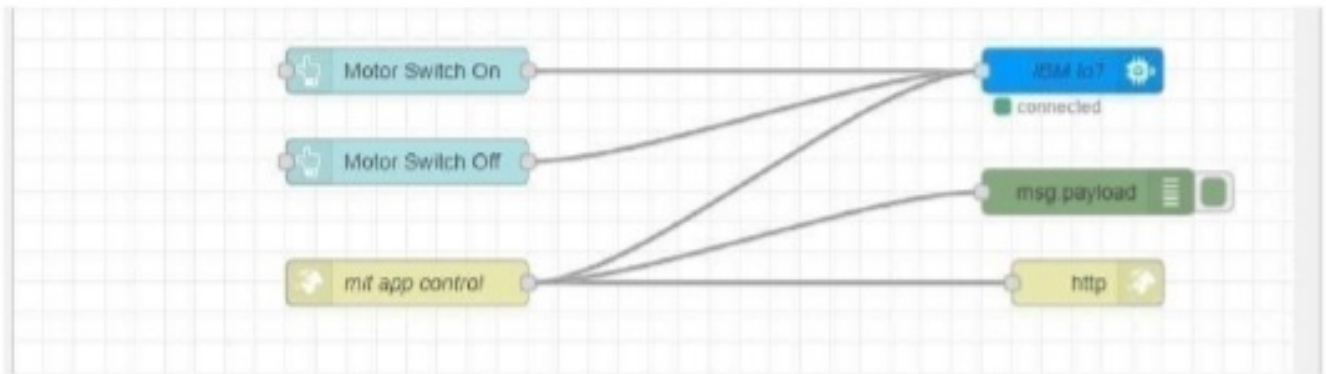


The above images show the java script codes of analyser and state function nodes.

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



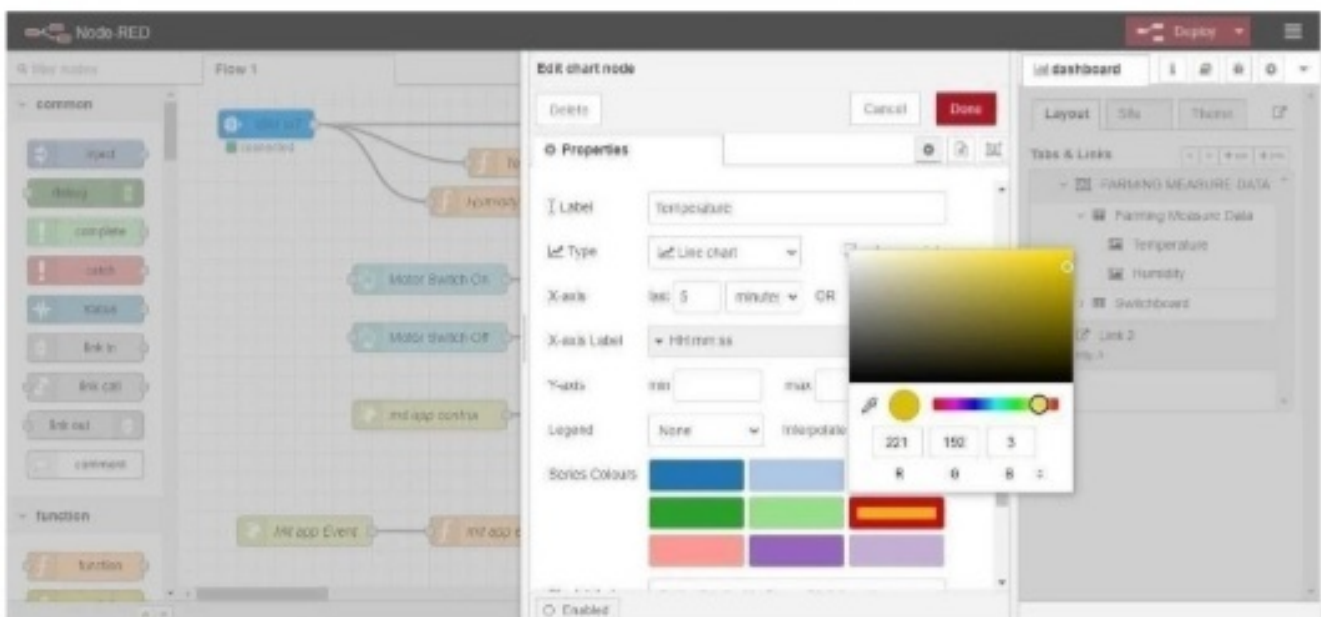
This is the program flow for sending commands to IBM cloud.

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.



The screenshot displays the Node-RED web interface. On the left, a palette lists various nodes categorized by function (input, output, function, etc.). The main workspace shows a flow titled 'Flow 1' with several nodes connected by wires. The flow starts with a 'mqtt in' node, which branches into three parallel paths for 'Temperature', 'Humidity', and 'Moisture'. Each path leads to a corresponding 'mqtt publish' node. Below this, there are 'MQTT Switch On' and 'MQTT Switch Off' nodes, which are connected to a 'mqtt publish' node. Further down, an 'mqtt app control' node is connected to an 'http' node. At the bottom, an 'MQTT App Events' node is connected to an 'mqtt app events' node, which is then connected to an 'http' node. On the right side, a 'debug' console shows the execution log, including messages received from the 'mqtt in' node and the output of the 'http' nodes.

SCREEN – 1

