

Smart Farmer-IOT Enabled Smart Farming Application

IBM NALAIYATHIRAN

SPRINT DELIVERY – 3

TITLE	Smart Farmer-IOT Enabled Smart Farming Application
DOMAIN NAME	INTERNET OF THINGS
TEAM ID	PNT2022TMID44924

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 'Add new ibmiot config node' configuration window in Node-Red. The window has a title bar 'Edit ibmiot in node > Add new ibmiot config node' and two buttons: 'Cancel' and 'Add'. Below the title bar is a 'Properties' tab with a settings icon and a document icon. The configuration fields are as follows:

- Name:** A text input field with the placeholder 'Name'.
- API Key:** A text input field containing 'a-ck2tf0-yutwjnphx'.
- API Token:** A text input field with masked characters (dots).
- Server-Name:** A text input field containing 'orgid.messaging.internetofthings.ibmcloud.com'.
- Scalable:** A checkbox that is currently unchecked.
- Application ID:** A text input field.
- Keep Alive:** A radio button selected, followed by a text input field containing '60' and the label 'Seconds'.
- Use Clean Session:** A checked checkbox.

At the bottom of the configuration window, there is a status bar with the following information:

- ☐ Enabled
- 0 nodes use this config**
- A dropdown menu currently set to 'On all flows'.

The background of the screenshot shows a Node-Red flow editor with a 'msg.payload' node and a list of nodes on the right side.

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.

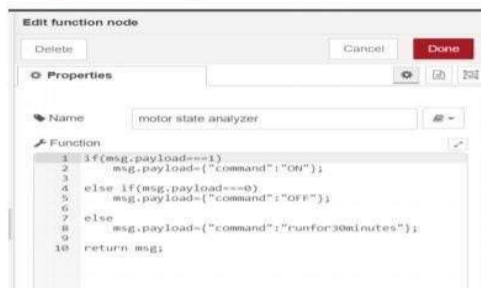
The Java script code for the analysis is:

```
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.

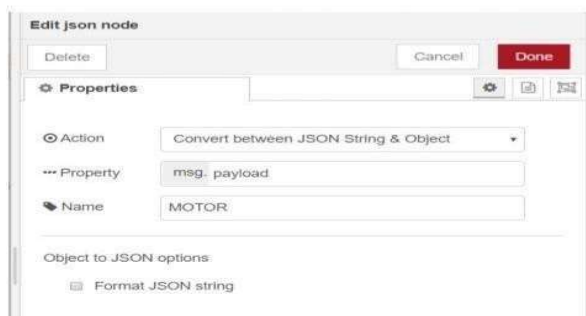
The Java script code for that function node is:

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

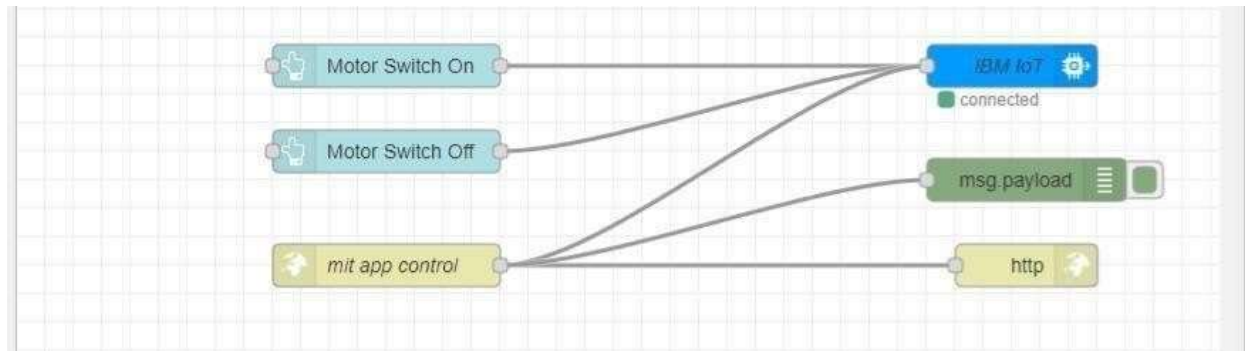


The above images show the java script codes of analyser and state function nodes.

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



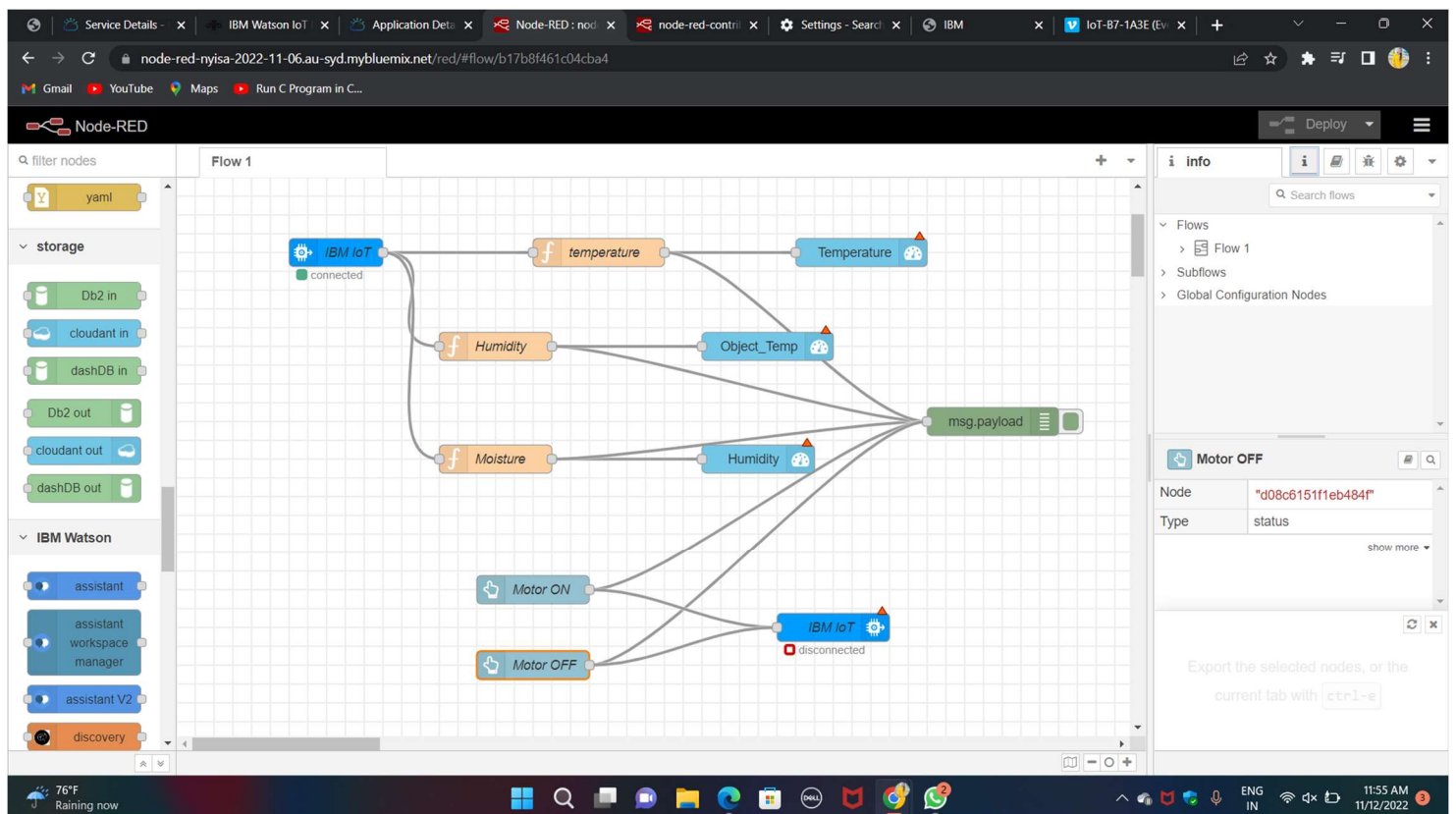
This is the program flow for sending commands to IBM cloud.

Adjusting User Interface

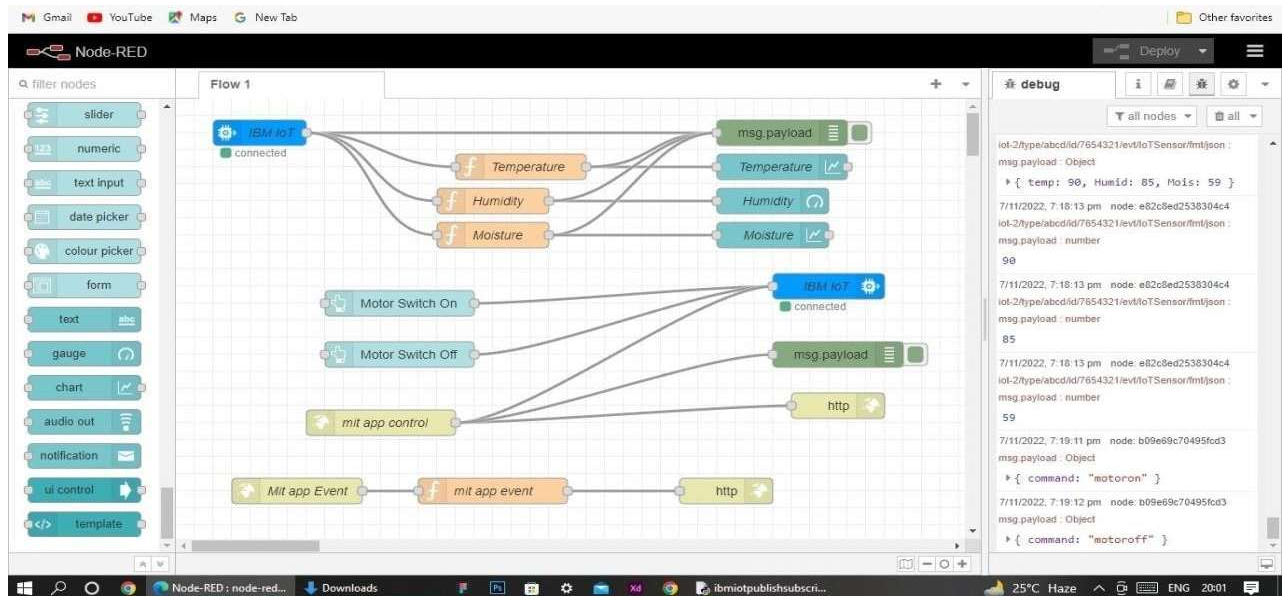
In order to display the parsed JSON data a Node-Red dashboard is created

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 we started to create the flow 1



COMPLETE PROGRAM FLOW :



MOBILE APP WEB : BLOCK DIAGRAM

