

Smart Farmer-IOT Enabled Smart Farming Application

SPRINT DELIVERY – 3

TITLE	Smart Farmer-IOT Enabled Smart Farming Application
DOMAIN NAME	INTERNET OF THINGS
TEAM ID	PNT2022TMID28757
LEADER NAME	A.Iswarya
TEAM MEMBER NAME	A.Uday Kiran N.Sireesha S.Manoj

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' section with a gear icon and a document icon. The configuration fields are as follows:

- Name:** A text input field with the placeholder text 'Name'.
- API Key:** A text input field containing the value 'a-ck2tf0-yutwjjanphx'.
- API Token:** A text input field with masked characters '*****'.
- Server-Name:** A text input field containing the value 'orgid.messaging.internetofthings.ibmcloud.com'.
- Scalable:** A checkbox that is currently unchecked.
- Application ID:** A text input field.
- Keep Alive:** A section containing a text input field with the value '60', the word 'Seconds', and a checked checkbox labeled 'Use Clean Session'.

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

- A radio button labeled 'Enabled'.
- A message icon and the text '0 nodes use this config'.
- A dropdown menu currently showing 'On all flows'.

The background of the Node-Red interface is visible, showing a flow with a 'msg.payload' node and a sidebar with a list of nodes.

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.

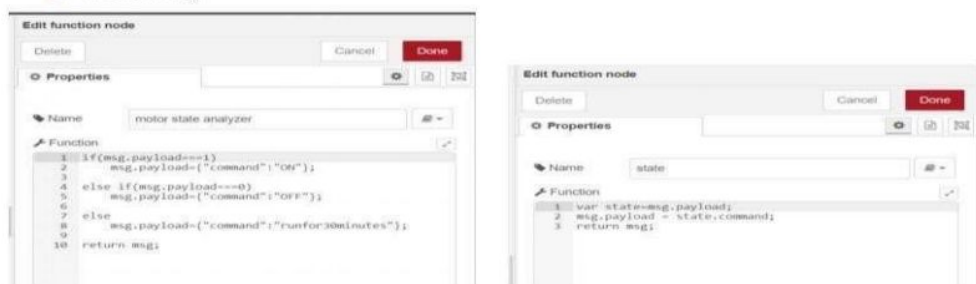
The Java script code for the analysis is:

```
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;
```

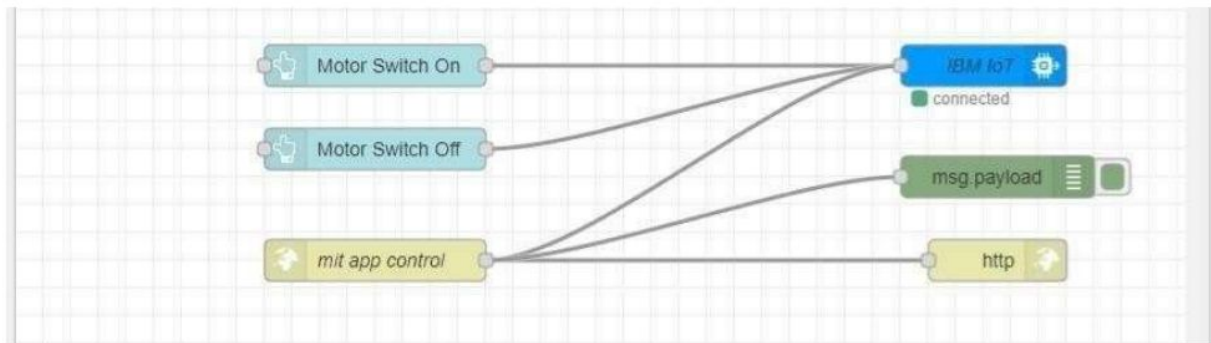


The above images show the JavaScript codes of the analyser and state function nodes.

Then we add an 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



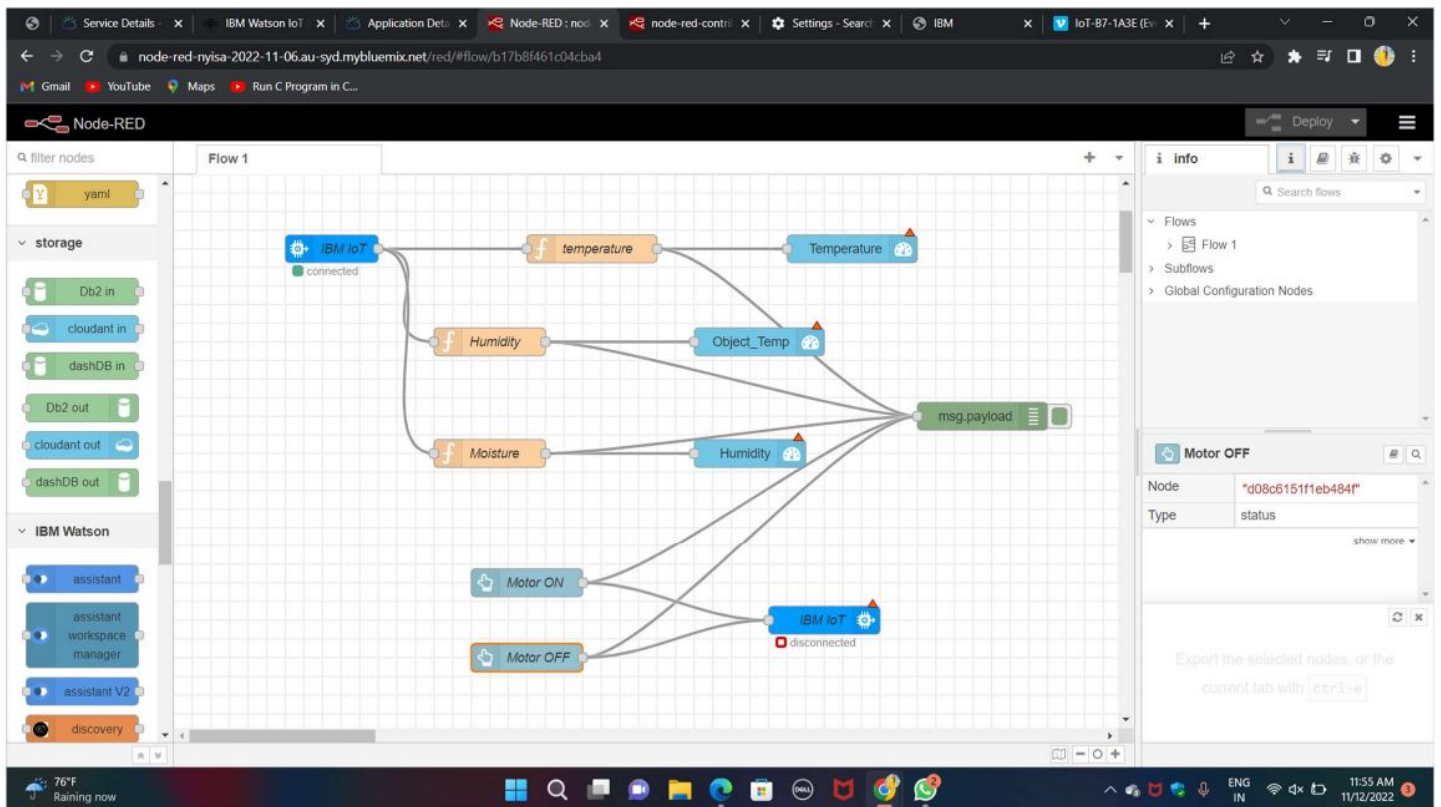
This is the program flow for sending commands to IBM cloud.

Adjusting User Interface

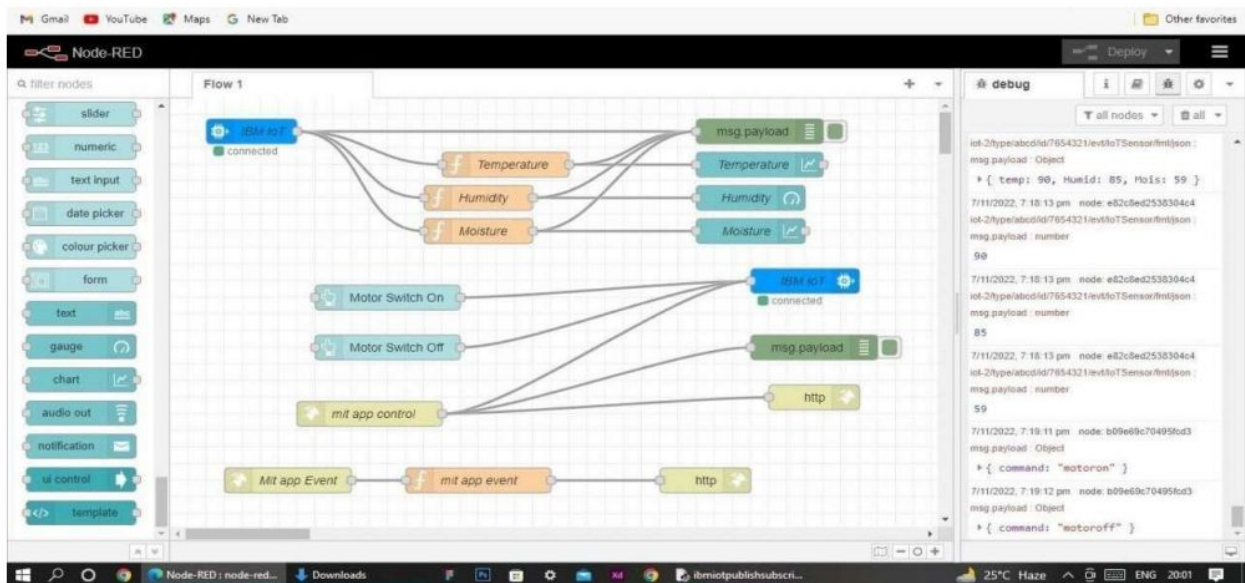
In order to display the parsed JSON data a Node-Red dashboard is created

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 :

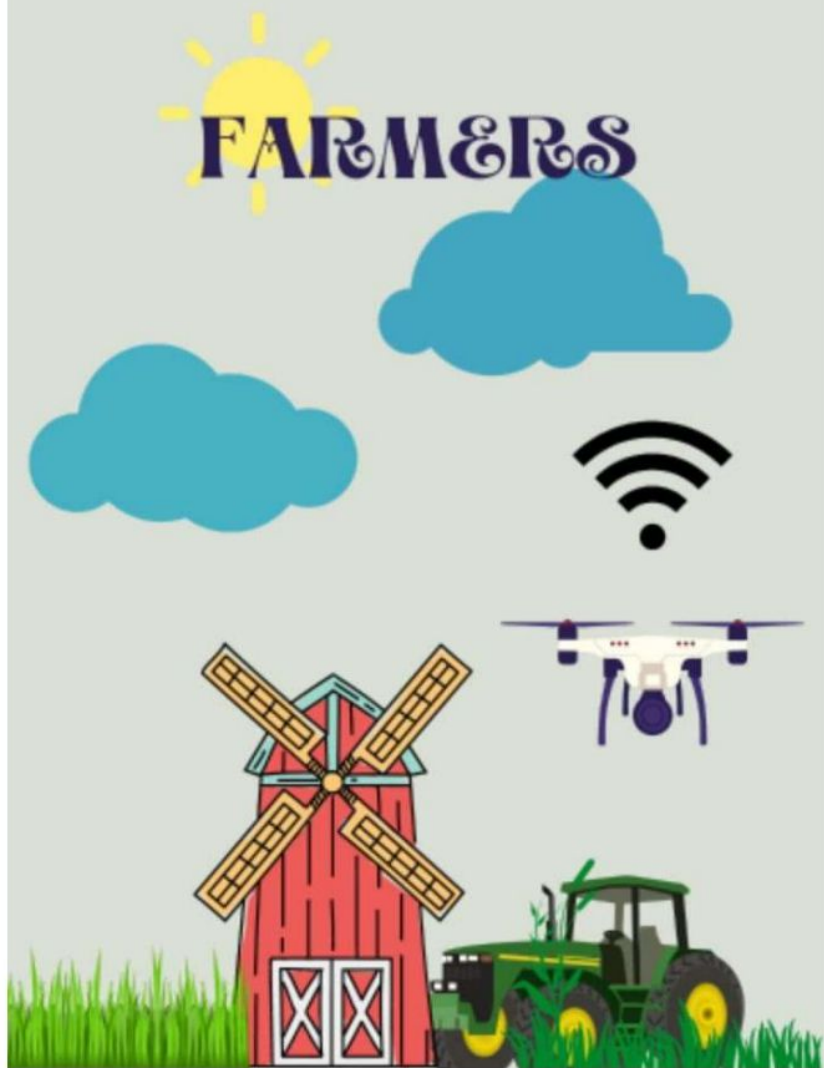


MOBILE APP WEB : BLOCK DIAGRAM



SMART

FARMERS



SMART

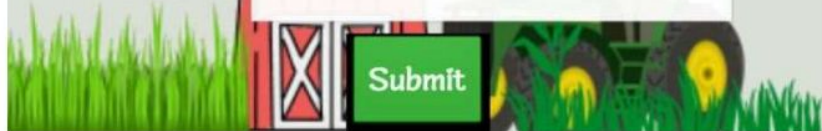
FARMERS



User Name

Password

Submit



Screen3

Smart Agriculture

Temperature

60

Humidity

95

Moisture

90

MOTOR ON

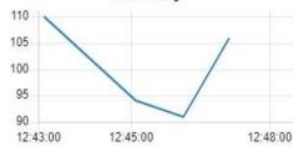
MOTOR OFF



Smart Farming

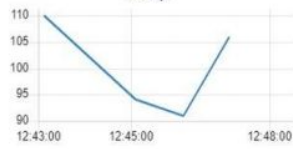
Farming

Humidity



LIGHT OFF

Temp



Garden

Moisture



Switch Board

LIGHT ON