

Team ID	PNT2022TMID19466
Project Title	Smart Farmer – IoT enabled Smart Farming Application
Date	16-11-2022

Build a web Application using NODE RED SERVICE:

- Open node red website
- Connect it with the iot Watson platform.
- Create the device and provide the credentials.
- Provide the device id, device type etc.
- Once filling the details provide finish and start simulation.
- Return back to node red and deploy the function gauges and chart.

IBM IOT IN NODE:

The screenshot displays the Node-RED web interface in a browser. The main workspace shows a flow with an 'ibmiot in' node connected to 'Soil Moisture', 'Humidity', and 'Temperature' nodes, which then connect to a 'switch' node and an 'http request' node. The right-hand panel is open to the 'Edit ibmiot in node' configuration window.

Edit ibmiot in node configuration:

- Authentication:** API Key
- API Key:** 2c533d245eb7b902
- Input Type:** Device Event
- Device Type:** All or +
- Device Id:** 0223
- Event:** All or +
- Format:** All or json
- QoS:** 0
- Name:** IBM IoT
- Service:** registered

Below the configuration fields, there is a note: "Use the Input Type property to configure this node to receive Events sent by IoT Devices, Commands sent to IoT Devices, Status".

The right sidebar shows a list of nodes, with 'ibmiot in' selected. A help panel for 'ibmiot in' is also visible, providing details about its functionality: "Input node that can be used with Watson IoT Platform to receive events sent from devices, receive commands sent to devices, or receive status updates concerning devices or applications. It produces an object called msg and sets msg.payload to be a String containing the payload of the incoming message." It also explains how device and application IDs are stored in the message object.

SOIL MOISTURE FUNCTION:

The screenshot displays the Node-RED web interface in a browser. The main workspace shows a flow named 'Flow 1' with the following components: an 'IBM IoT' node (connected), a 'switch' node, and an 'http request' node. The flow is connected to three function nodes: 'Soil Moisture', 'Humidity', and 'Temperature'. The 'Soil Moisture' function node is selected, and its configuration is shown in the 'Edit function node' panel. The 'Name' is 'Soil Moisture', and the 'On Message' tab is active, containing the following JavaScript code:

```
1 global.set('moist',msg.payload.soil_moisture)
2 msg.payload=msg.payload.soil_moisture
3 return msg;
```

The right sidebar shows the 'help' panel for the 'function' node, providing information about its usage and conventions. The bottom status bar indicates the system is 'Enabled' and shows the date and time as 12:44 PM on 16-11-2022.

SOIL MOISTURE GAUGE:

The screenshot shows the Node-RED web interface in a browser. The main workspace displays a flow named 'Flow 1'. The flow starts with an 'IBM IoT' node (green) connected to a 'switch' node (yellow). The 'switch' node has three outputs, each leading to a function node: 'Soil Moisture', 'Humidity', and 'Temperature'. These function nodes are connected to an 'http request' node (green), which then connects to an 'Ina11ria' node (green). The right sidebar is open to the 'Edit gauge node' configuration for the 'Soil Moisture' gauge. The configuration includes: Group: '[Home] Soil Moisture', Size: 'auto', Type: 'Gauge', Label: 'Soil Moisture', Value format: '{{value}}', Units: 'units', Range: min 0, max 100, Colour gradient: a gradient from green to red, Sectors: 0, optional, optional, 100, Class: 'Optional CSS class name(s) for widget', and Name: (empty). The bottom status bar shows '28°C Cloudy' and the time '12:38 PM 16-11-2022'.

HUMIDITY FUNCTION:

The screenshot shows the Node-RED web interface in a browser. The main workspace displays a flow named 'Flow 1'. The flow starts with an 'IBM IoT' node (green) connected to a 'switch' node (yellow). The 'switch' node has three outputs, each leading to a function node: 'Soil Moisture', 'Humidity', and 'Temperature'. These function nodes are connected to an 'http request' node (green), which then connects to an 'Ina11ria' node (green). The right sidebar is open to the 'Edit function node' configuration for the 'Humidity' function. The configuration includes: Name: 'Humidity', and the 'On Message' tab is selected. The code in the 'On Message' tab is:

```
1 global.set('hum',msg.payload.humidity)
2 msg.payload=msg.payload.humidity
3 return msg;
```

 The bottom status bar shows '28°C Cloudy' and the time '12:47 PM 16-11-2022'.

HUMIDITY CHART:

The screenshot shows the Node-RED web interface in a browser. The main workspace displays a flow named 'Flow 1'. It starts with an 'IBM IoT' node (connected), which branches into three function nodes: 'Soil Moisture', 'Humidity', and 'Temperature'. These function nodes are connected to a 'switch' node, which then connects to an 'http request' node. The 'http request' node is connected to an 'Inet/r1ata' node, which is connected to a 'r1ata' node. The 'Edit chart node' panel is open on the right, showing the configuration for the 'Humidity' chart node. The configuration includes: Group: '[Home] Soil Moisture', Size: 'auto', Label: 'Humidity', Type: 'Line chart', X-axis: 'last 1 hours OR 1000 points', X-axis Label: 'HH:mm:ss', Y-axis: 'min' and 'max' fields, Legend: 'None', Interpolate: 'linear', and Series Colours: a grid of colored squares. The 'help' panel on the far right provides documentation for the 'chart' node.

Edit chart node

Properties

Group: [Home] Soil Moisture

Size: auto

Label: Humidity

Type: Line chart ☐ enlarge points

X-axis: last 1 hours OR 1000 points

X-axis Label: HH:mm:ss ☐ as UTC

Y-axis: min max

Legend: None Interpolate: linear

Series Colours: [Color grid]

☐ Enabled

chart

Plots the input values on a chart. This can either be a time based line chart, a bar chart (vertical or horizontal), or a pie chart.

Each input `msg.payload` value will be converted to a number. If the conversion fails, the message is ignored.

Minimum and Maximum Y axis values are optional. The graph will auto-scale to any values received.

Multiple series can be shown on the same chart by using a different `msg.topic` value on each input message. Multiple bars of the same series can be shown by using the `msg.label` property.

The X axis defines a time window or a

TEMPERATURE FUNCTION:

The screenshot shows the Node-RED web interface in a browser. The main workspace displays a flow named 'Flow 1'. It starts with an 'IBM IoT' node (connected), which branches into three function nodes: 'Soil Moisture', 'Humidity', and 'Temperature'. These function nodes are connected to a 'switch' node, which then connects to an 'http request' node. The 'http request' node is connected to an 'Inet/r1ata' node, which is connected to a 'r1ata' node. The 'Edit function node' panel is open on the right, showing the configuration for the 'Temperature' function node. The configuration includes: Name: 'Temperature', Setup tab selected, and a JavaScript code block with the following code:

```
1 global.set('temp',msg.payload.temperature)
2 msg.payload=msg.payload.temperature
3 return msg;
```

The 'help' panel on the far right provides documentation for the 'function' node.

Edit function node

Properties

Name: Temperature

Setup On Start On Message On Stop

```
1 global.set('temp',msg.payload.temperature)
2 msg.payload=msg.payload.temperature
3 return msg;
```

☐ Enabled

function

A JavaScript function to run against the messages being received by the node.

The messages are passed in as a JavaScript object called `msg`.

By convention it will have a `msg.payload` property containing the body of the message.

The function is expected to return a message object (or multiple message objects), but can choose to return nothing in order to halt a flow.

The **On Start** tab contains code that will be run whenever the node is started. The **On Stop** tab contains code that will be run when the node is stopped.

If the **On Start** code returns a Promise object

TEMPERATURE CHART:

The screenshot shows the Node-RED web interface in a browser. The main workspace displays a flow with an IBM IoT node connected to three function nodes labeled 'Soil Moisture', 'Humidity', and 'Temperature'. These function nodes are connected to a 'switch' node, which then connects to an 'http request' node. The 'http request' node is connected to an 'Inet/data' node, which is connected to a 'data' node. The 'data' node is connected to a 'chart' node. The 'chart' node is configured with the following properties:

- Group: [Home] Soil Moisture
- Size: auto
- Label: Temperature
- Type: Line chart
- X-axis: last 1 hours OR 1000 points
- X-axis Label: HH:mm:ss
- Y-axis: min max
- Legend: None
- Interpolate: linear
- Series Colours: (A grid of color swatches)

The right sidebar shows the 'help' panel for the 'chart' node, which provides detailed information about its usage and configuration options.

SWITCH OPTION:

The screenshot shows the Node-RED web interface in a browser. The main workspace displays a flow with an IBM IoT node connected to three function nodes labeled 'Soil Moisture', 'Humidity', and 'Temperature'. These function nodes are connected to a 'switch' node, which then connects to an 'http request' node. The 'http request' node is connected to an 'Inet/data' node, which is connected to a 'data' node. The 'data' node is connected to a 'MOTOR ON' node. The 'MOTOR ON' node is connected to a 'data' node. The 'data' node is connected to a 'switch' node. The 'switch' node is configured with the following properties:

- Name: Name
- Property: msg.payload
- Rules: (A list of rules, currently empty)
- Checking all rules: (checked)
- Recreate message sequences: (unchecked)

The right sidebar shows the 'help' panel for the 'switch' node, which provides detailed information about its usage and configuration options.

MOTOR ON BUTTON:

The screenshot shows the Node-RED web interface in a browser. The main workspace displays a flow named 'Flow 1' with several nodes: a switch, an http request, a data node, a function node, and two output nodes labeled 'MOTOR ON' and 'MOTOR OFF'. The 'MOTOR ON' node is selected, and the 'Edit button node' panel is open on the right. In this panel, the 'Label' is set to 'MOTOR ON', the 'Payload' is set to 'a_2', and the 'Topic' is set to 'msg.topic'. The 'When clicked, send:' section is also visible. The bottom status bar shows the system time as 12:52 PM on 16-11-2022.

MOTOR OFF BUTTON:

This screenshot is similar to the one above, showing the same Node-RED flow. However, the 'MOTOR OFF' node is now selected, and the 'Edit button node' panel is updated accordingly. The 'Label' is set to 'MOTOR OFF', while the 'Payload' remains 'a_2' and the 'Topic' remains 'msg.topic'. The 'When clicked, send:' section is also visible. The bottom status bar shows the system time as 12:52 PM on 16-11-2022.

IBM IOT OUT NODE:

The screenshot shows the Node-RED web interface in a browser. The main workspace displays a flow named 'Flow 1' with several nodes: 'inject', 'connected', 'Temperature', 'switch', 'http request', 'data', 'MOTOR ON', 'MOTOR OFF', and 'msg'. The 'Edit ibmiot out node' configuration panel is open on the right, showing the following settings:

- Authentication: API Key
- API Key: 2c533d245eb7b902
- Output Type: Device Event
- Device Type: 2.2.2
- Device Id: device id e.g. ab12cd231a21
- Event Type: event type e.g. blink
- Format: json
- Data: payload information e.g. data points
- QoS: 0
- Name: IBM IoT
- Service: registered

The 'Enabled' checkbox is checked. The help panel on the far right provides information about the 'ibmiot out' node, including its purpose and message properties.

NODE-RED:

The screenshot shows the Node-RED web interface with a flow named 'Flow 1'. The flow includes the following nodes and connections:

- Inputs:** 'Soil Moisture', 'Humidity', and 'Temperature' sensors.
- Processing:** Each sensor is connected to a corresponding function node (e.g., 'Soil Moisture' to 'f Soil Moisture'). These function nodes are connected to a 'switch' node.
- Outputs:** The 'switch' node is connected to an 'http request' node, which then connects to 'msg.payload' nodes.
- Actuators:** 'MOTOR ON' and 'MOTOR OFF' nodes are connected to an 'IBM IoT' node.
- Other:** A 'data' node is connected to an 'http' node.

The 'IBM IoT' node is currently 'disconnected'. The help panel on the right shows the 'button' node configuration.

IBM App Developer x Node-RED: node x IBM Cloud Account x Service Details x IBM Watson IoT x Create a Cluster x MIT App Inventor x MIT App Inventor x

node-red-apdtc-2022-11-15.eu-gb.mybluemix.net/red/#flow/3cd27c109e543c30

www.eroderic.in/tc... Inbox (16) - atjaiy@... New Tab New Tab Attempt Login User Gmail YouTube Maps Full Video: Nastu N...

Node-RED

filter nodes

common

- inject
- debug
- complete
- catch
- status
- link in
- link call
- link out
- comment

function

- function
- switch
- change
- range

Flow 1

switch

http request

msg.payload

[get]/data

data

http

MOTOR ON

MOTOR OFF

msg.payload

[get]/command

http

IBM IoT

disconnected

help

Search help

- audio out
- Dashboard
- button
- chart
- colour picker
- status, nic/gar

button

Adds a button to the user interface.

Clicking the button generates a message with `msg.payload` set to the **Payload** field. If no payload is specified, the node id is used.

The **Size** defaults to 3 by 1.

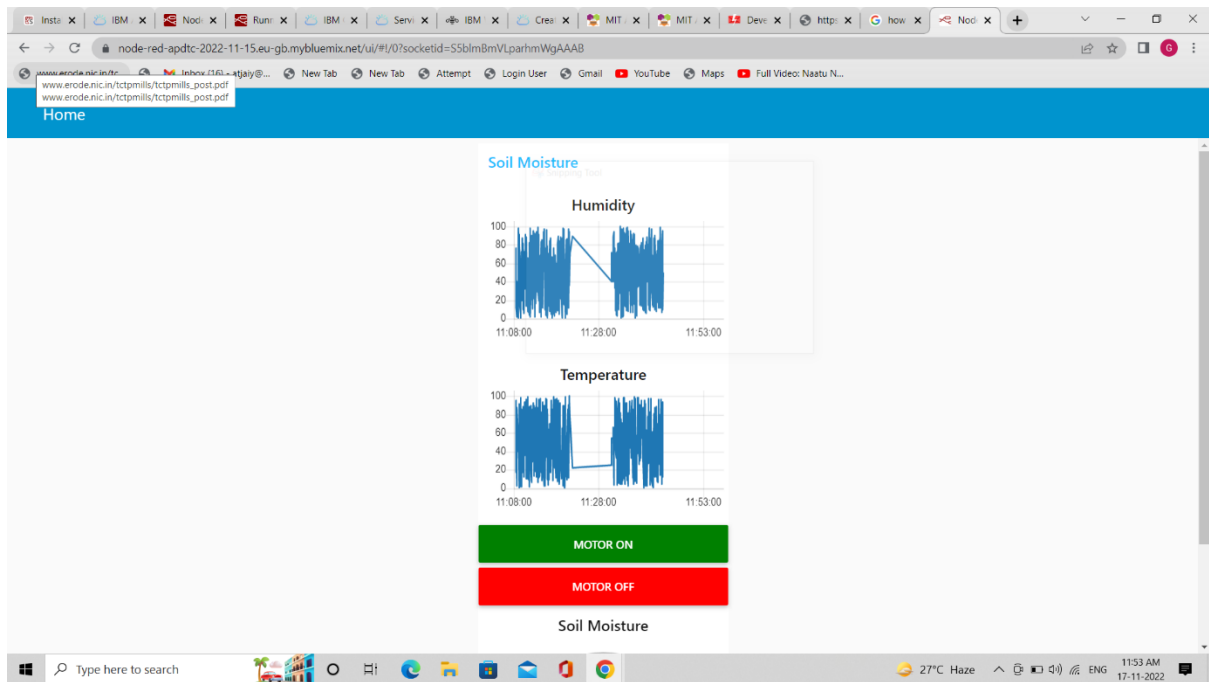
The **Icon** can be defined, as either a **Material Design icon** (e.g. 'check', 'close') or a **Font Awesome icon** (e.g. 'fa-fire'), or a **Weather icon**. You can use the full set of google material icons if you add 'mi-' to the icon name. e.g. 'mi-videogame_asset'.

The colours of the text and background may be set. They can also be set by a message property by setting the field to the name of the property, for example `{{background}}`.

Type here to search

28°C Cloudy 12:54 PM 16-11-2022

NODE RED DASHBOARD HOME:



NODE RED DEBUG CONSOLE:

