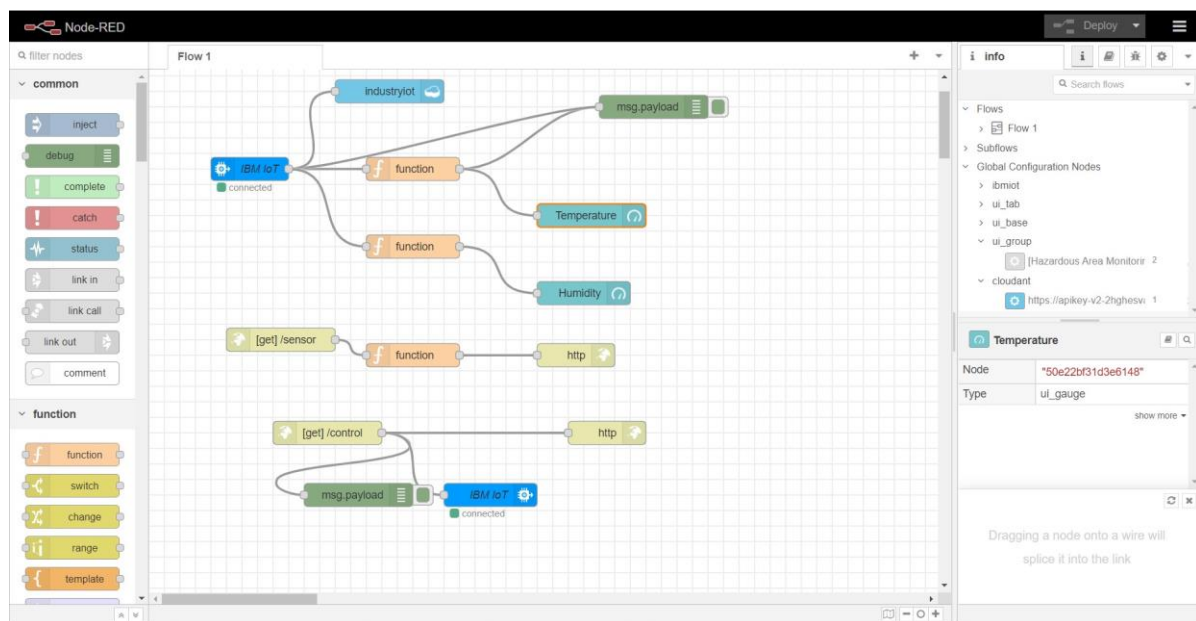


Configure The Application To Receive The Data From Cloud

Date	19 November 2022
Team Id	PNT2022TMID23518
Title	Hazardous Area Monitoring for Industrial Plant using IoT

Node red flow created to get values



Configuring function to fetch the desired value

The screenshot shows the Node-RED web interface. On the left, a flow named 'Flow 1' is visible. It starts with an 'IBM IoT' node (green, connected) which branches into three 'function' nodes (orange). The top 'function' node is connected to an 'industryiot' node (blue). The middle 'function' node is connected to a '[get] /sensor' node (yellow). The bottom 'function' node is connected to a '[get] /control' node (yellow). The '[get] /control' node is connected to a 'msg.payload' node (green). The right pane shows the 'Edit function node' configuration window. The 'Name' field is empty. The 'On Message' tab is selected. The code in the editor is:

```
1 msg.payload = msg.payload.temp;
2 global.set('t',msg.payload)
3 return msg;
```

At the bottom of the configuration window, there is an 'Enabled' checkbox which is currently unchecked.

The screenshot shows the Node-RED web interface. On the left, a flow named 'Flow 1' is visible. It starts with an 'IBM IoT' node (green, connected) which branches into three 'function' nodes (orange). The top 'function' node is connected to an 'industryiot' node (blue). The middle 'function' node is connected to a '[get] /sensor' node (yellow). The bottom 'function' node is connected to a '[get] /control' node (yellow). The '[get] /control' node is connected to a 'msg.payload' node (green). The right pane shows the 'Edit function node' configuration window. The 'Name' field is empty. The 'On Message' tab is selected. The code in the editor is:

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

At the bottom of the configuration window, there is an 'Enabled' checkbox which is currently unchecked.

App Blocks to render the values and display it in app

The screenshot displays the MIT App Inventor web interface for a project named "IoT_Interface". The interface is divided into three main sections: a left sidebar with "Blocks" and "Media", a top navigation bar, and a central "Viewer" area.

Blocks Panel: The "Built-in" category is expanded, showing various block types like Control, Logic, Math, Text, Lists, Dictionaries, Colors, Variables, and Procedures. Under "Screen1", there are blocks for "HorizontalArrangement", "Image1", "VerticalScrollArrangement", "Label1", "Label2", and another "HorizontalArrangement".

Viewer Area: This area shows the visual representation of the app's logic. It contains three event-driven blocks:

- Block 1:** A "when Clock1.Timer" block with a "do" section containing "set Web1.Url to http://159.122.177.234:30241/sensor" and "call Web1.Get".
- Block 2:** A "when Web1.GetText" block with a table for handling response data. The table has columns for "uri", "responseCode", "responseType", and "responseContent". The "do" section contains logic to parse the response: "look up in pairs key temperature" and "look up in pairs key humidity", followed by "call Web1.JsonTextDecode" and "get responseContent".
- Block 3:** A "when Button1.Click" block with a "do" section containing "set Web2.Url to http://159.122.177.234:30241/control?command=rot" and "call Web2.Get".

The interface also includes a "Designer" tab, a "Show Warnings" button, and a "Privacy Policy and Terms of Use" link at the bottom.