

SPRINT-2

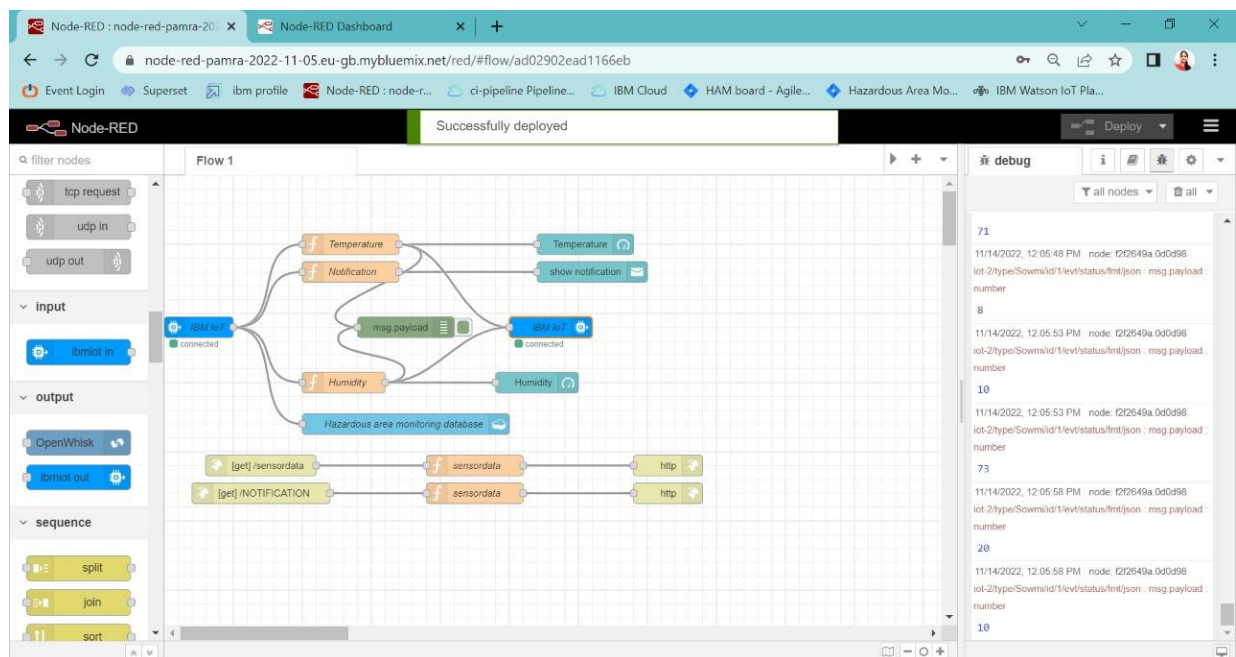
SOFTWARE

Date	13 November 2022
Team ID	PNT2022TMID42278
Project Name	Hazardous Area Monitoring for Industrial Power Plant powered by IOT

PROCEDURE:

- Step1: Develop node red application.
- Step2: Install the required nodes from manage palette option.
- Step3: Connect the node flow. Step4: Deploy the flow.

NODE RED:



FUNCTION NODE COMMAND TO INDICATE THE TEMPERATURE:

```
msg.payload=msg.payload.temperature;  
global.set('temperature',msg.payload); return msg;
```

FUNCTION NODE COMMAND TO INDICATE THE HUMIDITY:

```
msg.payload=msg.payload.humidity;  
global.set('humidity',msg.payload); return msg;
```

FUNCTION NODE COMMAND TO SHOW THE STATUS:

```
temperature=msg.payload.temperature;  
humidity=msg.payload.humidity; if(temperature>50  
& humidity>50){ msg.payload="Temperature and  
Humidity Alert!";  
}  
else if(temperature>50){ msg.payload="Temperature  
Alert!";  
}  
else if(humidity>50){ msg.payload="Humidity  
Alert!";  
}  
else {msg.payload="safe";} return msg;
```

FUNCTION NODE COMMAND TO GET THE HTTP REQUEST FOR SENSOR DATA:

```
msg.payload={  
  "temperature":global.get('temperature'),  
  "humidity":global.get('humidity')  
}  
return msg;
```

FUNCTION NODE COMMAND TO GET THE HTTP REQUEST FOR STATUS:

```
msg.payload={
```

```

"temperature":global.get('temperature'),
"humidity":global.get('humidity')
}

temperature=msg.payload.temperature; humidity=msg.payload.humidity;

if(temperature>50
& humidity>50){ msg.payload="Temperature and
Humidity Alert!";
}

else if(temperature>50){ msg.payload="Temperature
Alert!";
}

else if(humidity>50){ msg.payload="Humidity
Alert!";
}

else {msg.payload="safe";} return msg;

```

NOTIFICATION NODE:

The screenshot shows the Node-RED web interface in a browser. The main workspace displays a flow with an 'IBM IoT' node connected to several other nodes. The 'Edit notification node' dialog is open, showing the following settings:

- Layout:** Top Right
- Timeout (S):** 5
- Border:** (optional) border highlight colour
- Send to all browser sessions:** ☒
- Accept raw HTML/JavaScript input in msg.payload to format popup:** ☐
- Class:** [msg.className]
- Topic:** STATUS
- Name:** Name
- Enabled:** ☐ Enabled

The debug console on the right shows a series of messages with the following payload structure:

```

{
  "temperature": 21,
  "humidity": 42
}

```

The messages are timestamped and include a node ID: 11/13/2022, 2:41:02 PM node: 12f2649a-0d0d98.

GAUGE-TEMPERATURE NODE:

The screenshot shows the Node-RED web interface in a browser. The main workspace displays a flow with an 'IBM IoT' node connected to several output nodes, including a 'Gauge' node. The 'Edit gauge node' dialog is open, showing the following configuration:

- Properties:**
 - Size: auto
 - Type: Gauge
 - Label: Temperature
 - Value format: {{value}}
 - Units: Centigrade
 - Range: min -20, max 125
 - Colour gradient: A gradient bar from green to red.
 - Sectors: -20, optional, optional, 125
- Enabled:** ☒

The debug console on the right shows a series of JSON payloads from the 'iot-2/type/Sowmi/id/1/evt/status/fmt/json' topic, each containing temperature and humidity data. The system clock at the bottom right indicates 14:41 on 13-11-2022.

GAUGE-HUMIDITY NODE:

The screenshot shows the Node-RED web interface with the 'Gauge' node configuration dialog open for humidity. The configuration is as follows:

- Properties:**
 - Group: [Dashboard] Sensor data
 - Size: auto
 - Type: Gauge
 - Label: Humidity
 - Value format: {{value}}
 - Units: Percentage
 - Range: min 0, max 100
 - Colour gradient: A gradient bar from green to red.
- Enabled:** ☒

The debug console on the right displays JSON payloads with temperature and humidity data. The system clock at the bottom right shows 14:41 on 13-11-2022.