

# SPRINT DELIVERY –

Team ID	PNT2022TMID32030
Project Name	SMART FARMER - IOT ENABLED SMART FARMINGAPPLICATION SYSTEM

## 5Building Project : Connecting Simulators For Readings:

By using wokwi to generate sensor stimulator output randomly for temperature,humidity,soilmoisture and connect them with the device credentials

OrgID: xhlz7n,

Api key: a-xhlz7n-qlpqdlk0oo

Api token : lzwf0ESiqGkQQcPg\*G

Device type: iotsensor ,

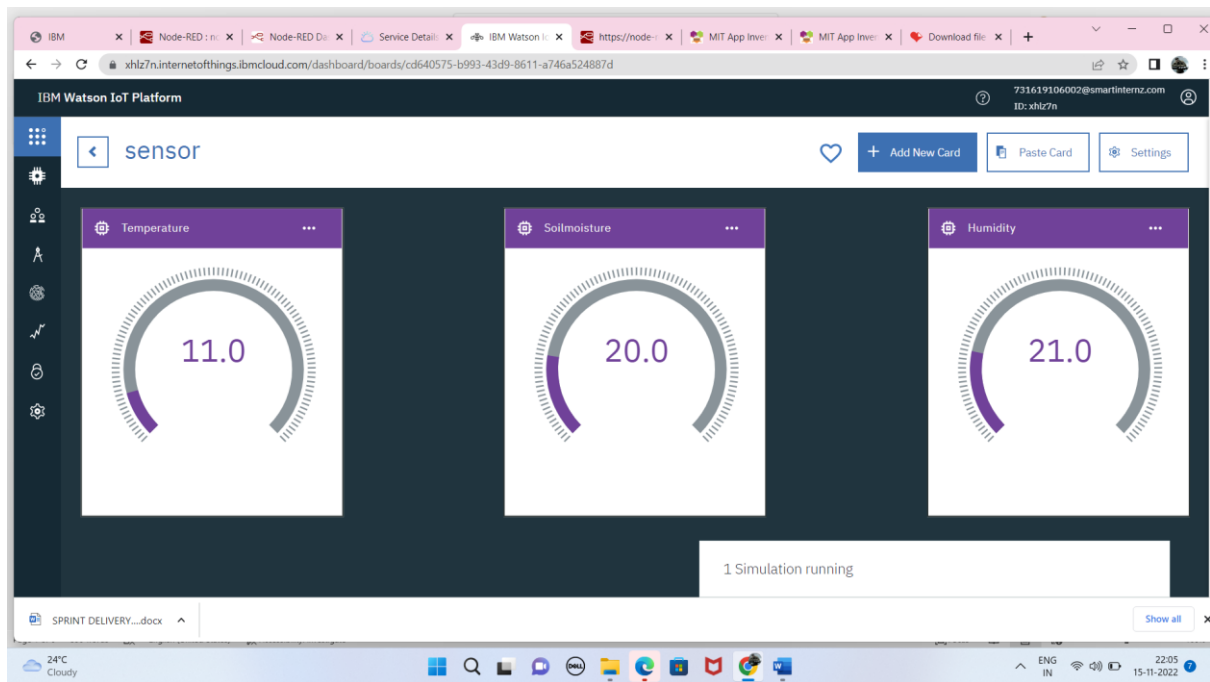
Device ID : iotsensor

Device Token : 2x\_okYGd6qMY77S(S7

You can see the received data in graphs by creating cards in Boards tab:

- To view in board need to add and create new boards with stimulated output value
- The values can be seen in recent tab

The following image is the board value of my team:



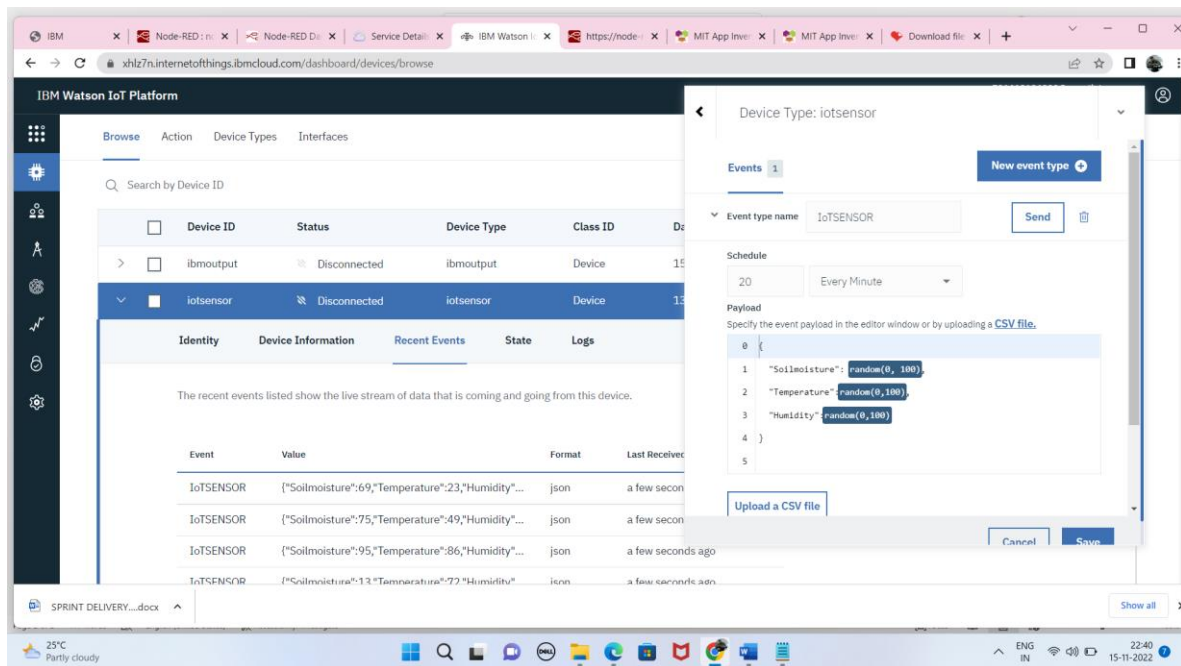
Recent :

The screenshot displays the 'Browse' tab of the IBM Watson IoT Platform dashboard. It shows a list of devices, including 'ibmoutput' and 'iotsensor', both marked as 'Disconnected'. The 'iotsensor' device is selected, and its 'Recent Events' tab is active. Below the device list, a table shows the live stream of data for the selected device.

Event	Value	Format	Last Received
IoTSENSOR	{"Soilmoisture":7,"Temperature":84,"Humidity":6}	json	a few seconds ago
IoTSENSOR	{"Soilmoisture":100,"Temperature":86,"Humidity":...}	json	39 minutes ago
IoTSENSOR	{"Soilmoisture":38,"Temperature":4,"Humidity":...}	json	39 minutes

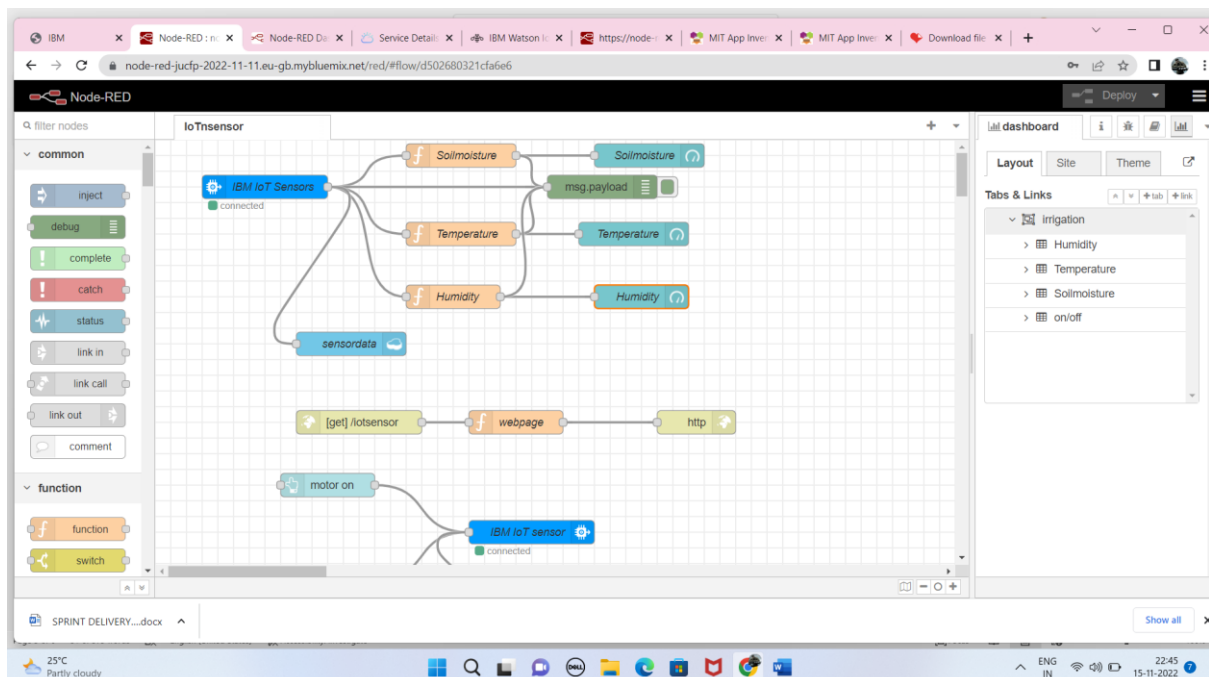
1 Simulation running

You can also stimulate using device stimulator program:



## Configuration of Node-Red to collect IBM cloud data :

The node IBM IoT App In is added to Node-Red workflow. Then the appropriate device credentials obtained earlier are entered into the node to connect and fetch device telemetry to Node-Red.



## Configuring ibm input node :

The screenshot shows the Node-RED web interface in a browser. The main workspace displays a flow with an 'IBM IoT Sensors' node connected to 'Soilmoisture', 'Temperature', and 'Humidity' nodes, which then connect to a 'sensordata' node. A 'motor on' node is also present. The 'Edit ibmiot in node' dialog is open, showing the following properties:

- Authentication: API Key
- API Key: sensor input
- Input Type: Device Event
- Device Type: ☐ All or ☒ iotsensor
- Device Id: ☐ All or ☒ iotsensor
- Event: ☒ All or ☐ IoTSENSOR
- Format: ☐ All or ☒ json
- QoS: 0
- Name: IBM IoT Sensors
- Service: registered
- ☐ Enabled

The right sidebar shows a 'dashboard' tab with a 'Layout' section containing 'Irrigation', 'Humidity', 'Temperature', 'Soilmoisture', and 'on/off'.

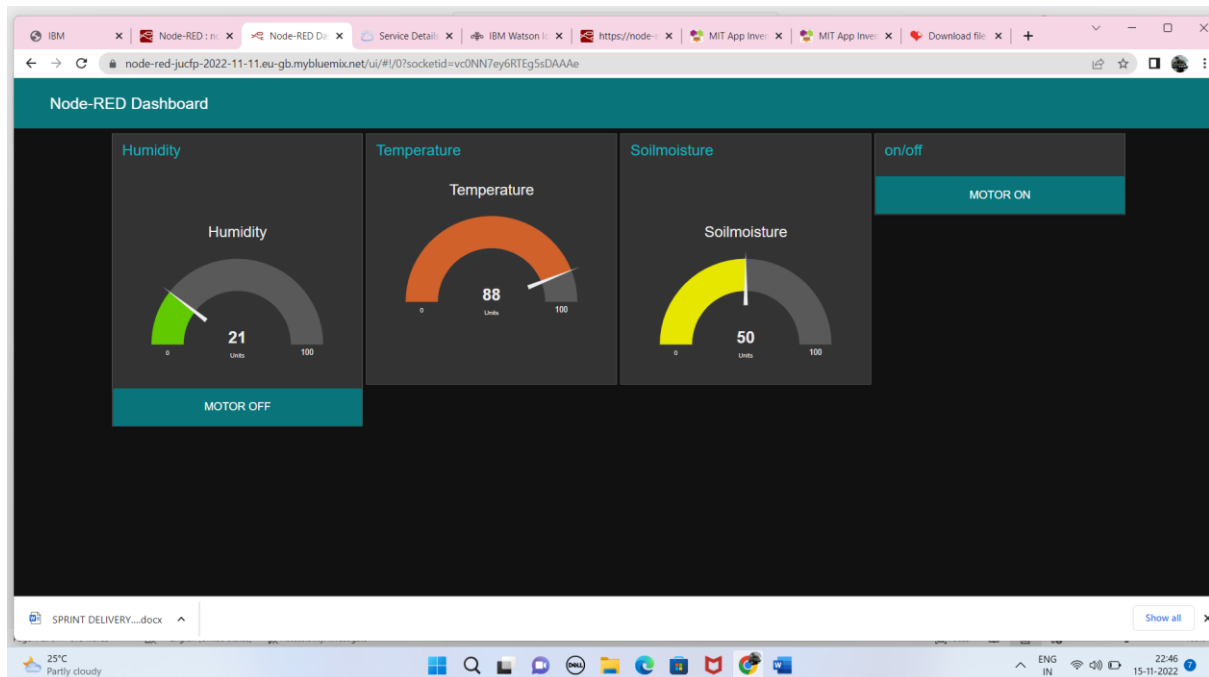
## Configuring Function nodes as sensor input receiver:

The screenshot shows the Node-RED web interface with the 'Edit function node' dialog open. The function node is named 'Soilmoisture' and is configured with the following code:

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

The dialog also shows the 'On Message' tab selected, and the 'Enabled' checkbox is checked. The background flow is the same as in the previous screenshot.

## Creating Dashboard :



## Open Weather Api:

The Node-Red also receive data from the OpenWeather API by HTTP GET request. An inject trigger is added to perform HTTP request for every certain interval. HTTP request node is configured with URL . The data we receive from OpenWeather after request is in below JSON

Generate api code and with api code you can generate a locations weather detail

<http://api.openweathermap.org/data/2.5/weather?q=erode,in&appid=b35d78d080636812518bcb784dad5903>

the link I was generated in the openweatherapi app

The url outcome is:

```
{"coord":{"lon":77.7333,"lat":11.35},"weather":[{"id":802,"main":"Clouds","description":"scattered clouds","icon":"03n"}],"base":"stations","main":{"temp":295.96,"feels_like":296.59,"temp_min":295.96,"temp_max":295.96,"pressure":1013,"humidity":88,"sea_level":1013,"grnd_level":993,"visibility":10000,"wind":{"speed":0.84,"deg":98,"gust":1.19},"clouds":{"all":32},"dt":1668534424,"sys":{"country":"IN","sunrise":1668473127,"sunset":1668514918},"timezone":19800,"id":1272013,"name":"Erode","cod":200}
```

Use the link in HTTP REQUEST :

The screenshot displays the Node-RED web interface in a browser. The main workspace shows a flow with several nodes: 'inject', 'debug', 'complete', 'catch', 'status', 'link in', 'link call', 'link out', 'comment', 'function', 'switch', 'motor off', 'connected', '[get] /command', 'command', 'timestamp', and 'http request'. The 'http request' node is selected, and its configuration panel is open on the right. The configuration panel shows the following settings:

- Method: GET
- URL: <http://api.openweathermap.org/data/2.5/weather>
- Payload: Ignore
- ☐ Enable secure (SSL/TLS) connection
- ☐ Use authentication
- ☐ Enable connection keep-alive
- ☐ Use proxy
- ☐ Only send non-2xx responses to Catch node
- Return: a parsed JSON object
- Name: Name
- Tip: If the JSON parse fails the fetched string is returned as-is.
- ☐ Enabled

The right sidebar shows a 'dashboard' tab with a 'Layout' section containing 'Humidity', 'Temperature', 'Solimmoisture', and 'on/off'.