

# Internet Of Things

## Lab - 9

Aadhitya Swarnesh



23 October 2020

### Aim :

To Apply and use the MQTT protocol with the help of Node-RED and concepts of IoT.

### Software :

Node-Red Software and HiveMQ MQTT broker.

### Methodology :

This video instructions are followed : <https://www.youtube.com/watch?v=LCYIFoyBn2I>

### Simulation And Output :

#### 1) Implementing an RFID chip scanner

##### (1.1) MQTT In Node

**Edit mqtt in node**

Delete Cancel Done

**Properties**

Server: RFID Reader@broker.mqttdashboard.cc

Topic: RFID Reader

QoS: 2

Output: auto-detect (string or buffer)

Name: RFID Reader

## (1.2) MQTT Server Configuration

Edit mqtt in node > **Edit mqtt-broker node**

Delete

Cancel

Update

**Properties**

Name

Name

Connection

Security

Messages

Server

broker.mqttdashboard.com

Port

1883

☐ Enable secure (SSL/TLS) connection

Client ID

RFID Reader

Keep alive time (s)

60

☒ Use clean session

☐ Use legacy MQTT 3.1 support

## (1.3) JSON Node

**Edit json node**

Delete

Cancel

Done

**Properties**

Action

Convert between JSON String & Object

▼

... Property

msg. payload

Name

Name

Object to JSON options

☐ Format JSON string

## (1.4) Debug Node

Edit debug node

Delete

Cancel

Done

⚙ Properties

⚙

📄

🖨

☰ Output

▼ msg. payload

🔗 To

☒ debug window

☐ system console


☐ node status (32 characters)

🏷 Name

Name

## (1.5) HiveMQ Window

⚠ Not Secure | hivemq.com/demos/websocket-client/

 **HIVEMQ**  
ENTERPRISE MQTT BROKER

Websockets Client Showcase

Connection

● connected

⌵

Host

broker.mqttdashboard.com

Port

8000

ClientID

clientId-YGUuKXGlt

Disconnect

Username

Password

Keep Alive

60

Clean Session

✕

Last-Will Topic

Last-Will QoS

0

Last-Will Retain

☐

Last-Will Message

Publish

⌵

Topic

RFID Reader

QoS

0

Retain

☐

Publish

Message

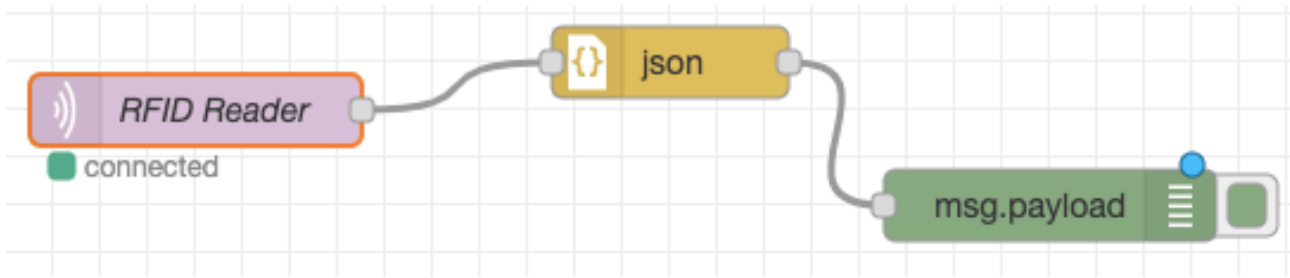
{"Name":"Aadhitya Swarnesh", "Organization":"VIT", "RegNo":"18BCE1087"}

Subscriptions

⌵

Add New Topic Subscription

## (1.6) Complete Circuit Flow



## (1.7) Complete Circuit Flow Code

```
[{"id":"3a2877a0.3a12b8","type":"tab","label":"RFID Scanner","disabled":false,"info":""},{ "id":"c6bf52d5.82e86","type":"mqtt in","z":"3a2877a0.3a12b8","name":"RFID Reader","topic":"RFID Reader","qos":"2","datatype":"auto","broker":"7d9cbc85.412e24","x":240,"y":140,"wires":[["3e74b8c6.94c8c8"]]}, {"id":"3e74b8c6.94c8c8","type":"json","z":"3a2877a0.3a12b8","name":"","property":"payload","action":"","pretty":false,"x":440,"y":120,"wires":[["5ef106ab.7245b8"]]}, {"id":"5ef106ab.7245b8","type":"debug","z":"3a2877a0.3a12b8","name":"","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload","targetType":"msg","statusVal":"","statusType":"auto","x":600,"y":180,"wires":[]}, {"id":"7d9cbc85.412e24","type":"mqtt-broker","z":"","name":"","broker":"broker.mqttdashboard.com","port":"1883","clientId":"RFID Reader","usetls":false,"compatmode":false,"keepalive":"60","cleansession":true,"birthTopic":"","birthQos":"0","birthPayload":"","closeTopic":"","closeQos":"0","closePayload":"","willTopic":"","willQos":"0","willPayload":""}]
```

## (1.8) Output

```
23/10/2020, 13:19:08 node: 5ef106ab.7245b8
RFID Reader : msg.payload : Object
▼ object
  Name: "Aadhitya Swarnesh"
  Organization: "VIT"
  RegNo: "18BCE1087"
```

## 2) Implementing a Speedometer

### (2.1) MQTT In Node

**Edit mqtt in node**

Delete Cancel Done

**Properties**

Server Speed Sensor

Topic Speed Sensor

QoS 2

Output auto-detect (string or buffer)

Name Speed Sensor

### (2.2) MQTT Server Configuration

**Edit mqtt in node > Edit mqtt-broker node**

Delete Cancel Update

**Properties**

Name Speed Sensor

**Connection** Security Messages

Server broker.mqttdashboard.com Port 1883

☐ Enable secure (SSL/TLS) connection

Client ID Speed Sensor

Keep alive time (s) 60 ☒ Use clean session

☐ Use legacy MQTT 3.1 support

### (2.3) JSON Node

**Edit json node**

Delete Cancel Done

**Properties**

Action Convert between JSON String & Object

Property msg. payload

Name Name

Object to JSON options

☐ Format JSON string

## (2.4) Gauge Node

Edit gauge node

Delete Cancel Done

⚙️ Properties 📄 🖼️

🗃️ Group [Bike UI] Bike ✎

🖼️ Size auto




☰ Type Gauge ▼

🏷️ Label Speedometer

🏷️ Value format {{value}}

🏷️ Units units

Range min 0 max 100

Colour gradient   

Sectors 0 ... optional ... optional ... 100

🔍 Name Speedometer

## (2.5) UI Tab

Edit gauge node > Edit dashboard group node > Edit dashboard tab node

Delete Cancel Update

⚙️ Properties 📄

🔍 Name Bike UI

🖼️ Icon dashboard

🔌 State ☒ Enabled

🔌 Nav. Menu ☒ Visible

## (2.6) UI Group

Edit gauge node > Edit dashboard group node

Delete Cancel Update

⚙️ Properties 📄

🔍 Name Bike

🗃️ Tab Bike UI ▼ ✎

↔️ Width 6

☒ Display group name

☐ Allow group to be collapsed

## (2.7) Text Node

Edit text input node

Delete

Cancel

Done

⚙️ Properties

⚙️

📄

🖼️

📁 Group

[Bike UI] Bike

✎

📏 Size

auto

🏷️ Label

Speed Slider

📖 Tooltip

optional tooltip

⌨️ Mode

text input

⌚ Delay (ms)

300

➔ If **msg** arrives on input, pass through to output:

☒

☒ When changed, send:

Payload

Current value

Topic

🏷️ Name

Speed Slider

## (2.8) Speed Indicator Node

Edit slider node

Delete

Cancel

Done

⚙️ Properties

⚙️

📄

🖼️

📁 Group

[Bike UI] Bike

✎

📏 Size

auto

🏷️ Label

Speed Indicator

📖 Tooltip

optional tooltip

↔️ Range

min

0

max

100

step

1

🔊 Output

continuously while sliding

▼

➔ If **msg** arrives on input, set slider to new payload value:

☒

☒ When changed, send:

Payload

Current value

Topic

🏷️ Name

Speed Indicator

## (2.9) Debug Node

Edit debug node

Delete

Cancel

Done

⚙ Properties

⚙

📄

🖨

☰ Output

▼ msg. payload

🔗 To

☒ debug window

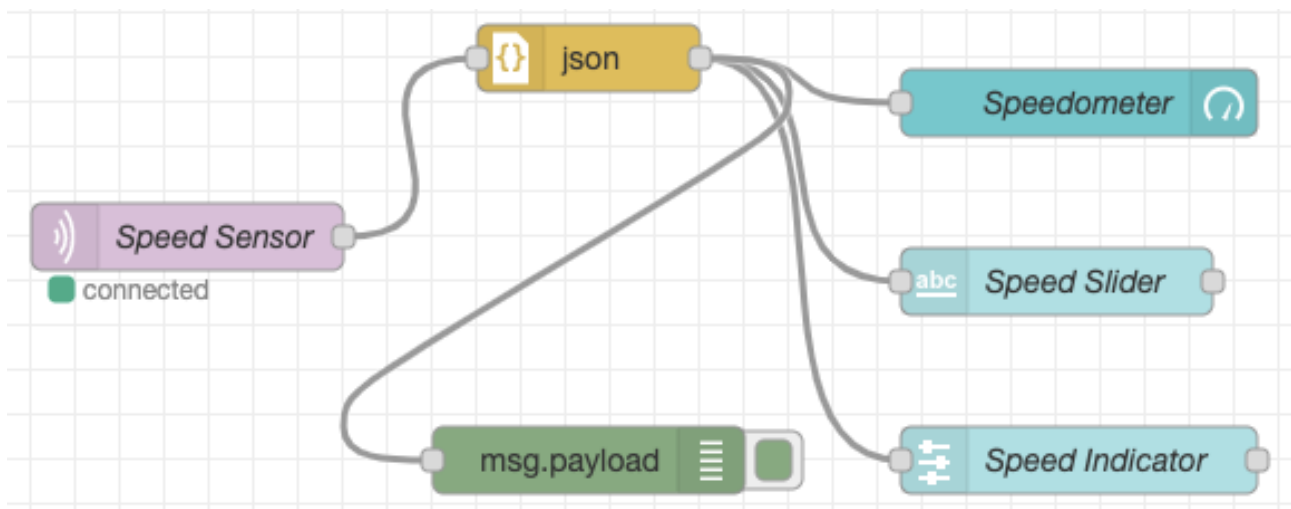
☐ system console

☐ node status (32 characters)

🔍 Name

Name

## (2.10) Complete Circuit Flow






## (2.11) Complete Circuit Flow Code

```
[{"id":"34557bbc.d95494","type":"tab","label":"Speed
Sensor","disabled":false,"info":""},{id:"865124dc.fa4818","type":"mqtt
in","z":"34557bbc.d95494","name":"Speed Sensor","topic":"Speed
Sensor","qos":"2","datatype":"auto","broker":"7c19841a.1ec52c","x":210,"y":180,"wires":[["1644041c.69b39c"]]},
{"id":"1644041c.69b39c","type":"json","z":"34557bbc.d95494","name":"","property":"payload","action":"","pretty":false,"x":390,"y":100,"wires":
[["7a011c5f.3874d4","d1ec4bd2.1099e8","765821bb.f3efc","431c0808.65b448"]]},
{"id":"431c0808.65b448","type":"debug","z":"34557bbc.d95494","name":"","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":false,"statusVal":"","statusType":"auto","x":390,"y":280,"wires":[]},
{"id":"7a011c5f.3874d4","type":"ui_gauge","z":"34557bbc.d95494","name":"Speedometer","group":"16c53fcc.777ba","order":0,"width":0,"height":0,"gtype":"gage","title":"Speedometer","label":"units","format":"{{value}}","min":0,"max":"100","colors":["#00b500","#e6e600","#ca3838"],"seg1":"","seg2":"","x":610,"y":120,"wires":[]},
{"id":"d1ec4bd2.1099e8","type":"ui_text_input","z":"34557bbc.d95494","name":"Speed Slider","label":"Speed Slider","tooltip":"","group":"16c53fcc.777ba","order":1,"width":0,"height":0,"passthru":true,"mode":"text","delay":300,"topic":"","x":600,"y":200,"wires":[[]]},
{"id":"765821bb.f3efc","type":"ui_slider","z":"34557bbc.d95494","name":"Speed Indicator","label":"Speed Indicator","tooltip":"","group":"16c53fcc.777ba","order":2,"width":0,"height":0,"passthru":true,"outs":"all","topic":"","min":0,"max":"100","step":1,"x":610,"y":280,"wires":[[]]},{id:"7c19841a.1ec52c","type":"mqtt-broker","z":"","name":"Speed Sensor","broker":"broker.mqttdashboard.com","port":"1883","clientid":"Speed Sensor","usetls":false,"compatmode":false,"keepalive":"60","cleansession":true,"birthTopic":"","birthQos":"0","birthPayload":"","closeTopic":"","closeQos":"0","closePayload":"","willTopic":"","willQos":"0","willPayload":""},
{"id":"16c53fcc.777ba","type":"ui_group","z":"","name":"Bike","tab":"d4af2056.81534","order":2,"disp":true,"width":"6","collapse":false},
{"id":"d4af2056.81534","type":"ui_tab","z":"","name":"Bike UI","icon":"dashboard","disabled":false,"hidden":false}]
```

## (2.12) HiveMQ Dashboard Portal

⚠ Not Secure | hivemq.com/demos/websocket-client/



**HIVEMQ**  
ENTERPRISE MQTT BROKER

Websockets Client Showcase

**Connection**

connected

⌵

Host

broker.mqttdashboard.com

Port

8000

ClientID

clientId-YGUuKXGigt

Disconnect

Username

Password

Keep Alive

60

Clean Session

☒

Last-Will Topic

Last-Will QoS

0

Last-Will Retain

☐

Last-Will Message

**Publish**

⌵

Topic

Speed Sensor

QoS

0

Retain

☐

Publish

Message

80

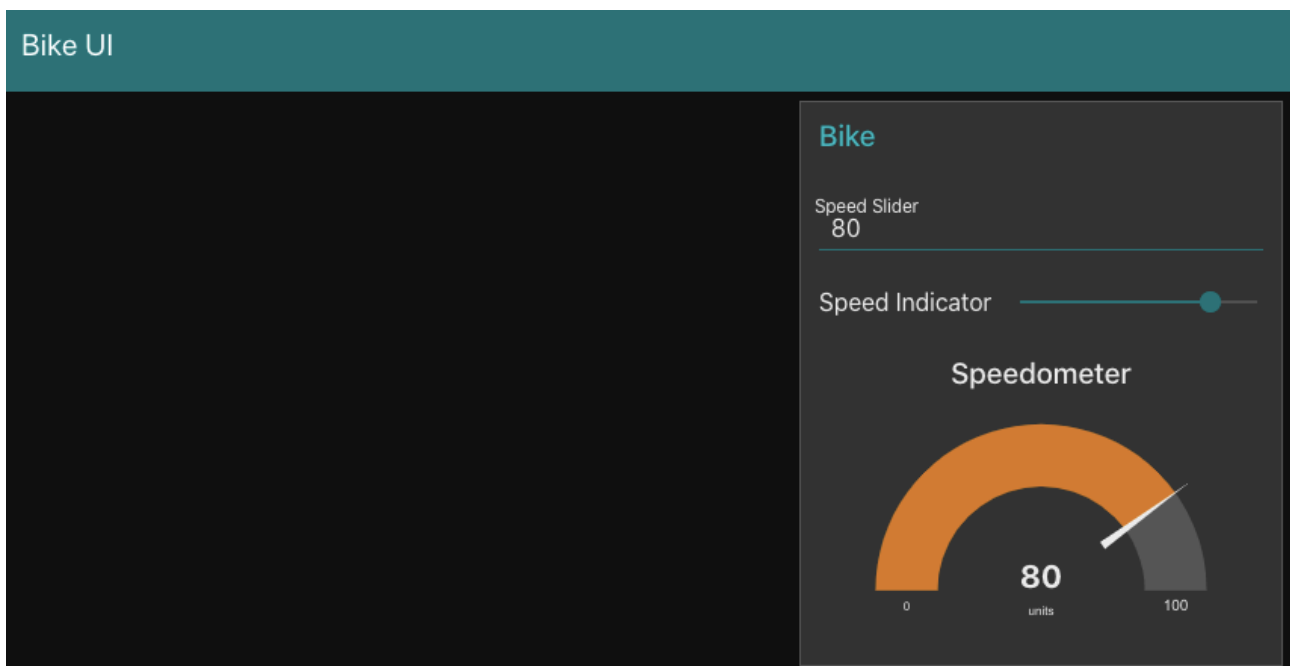
**Subscriptions**

⌵

Add New Topic Subscription

## (2.13) Output

```
23/10/2020, 13:42:11  node: 431c0808.65b448
Speed Sensor : msg.payload : number
80
```



### 3) Implementing a IoT based smart Fridge

#### (3.1) Inject Nodes

**Edit inject node**

Delete Cancel Done

**Properties**

Name Vegetable Controller

msg. payload = a<sub>z</sub> Tomatoes not available

msg. topic = a<sub>z</sub> Fridge Sensor

**Edit inject node**

Delete Cancel Done

**Properties**

Name Milk Controller

msg. payload = a<sub>z</sub> Three Packets of milk available

msg. topic = a<sub>z</sub> Fridge Sensor

**Edit inject node**

Delete Cancel Done

**Properties**

Name Egg Controller

msg. payload = 0<sub>9</sub> 12

msg. topic = a<sub>z</sub> Fridge Sensor

### (3.2) Debug Node

Edit debug node

Delete

Cancel

Done

⚙ Properties

⚙

📄

🖨

📄 Output

▼ msg. payload

🔗 To

☒ debug window

☐ system console

☐ node status (32 characters)

🔑 Name

Name

### (3.3) MQTT Out Node

Edit mqtt out node

Delete

Cancel

Done

⚙ Properties

⚙

📄

🖨

🌐 Server

Refrigerator

▼

✎

📄 Topic

Fridge Sensor

⚙ QoS

▼

🔄 Retain

▼

🔑 Name

Name

### (3.4) MQTT Server Configuration

Edit mqtt out node > Edit mqtt-broker node

Delete

Cancel

Update

⚙ Properties

⚙

📄

🔑 Name

Refrigerator

Connection

Security

Messages

🌐 Server

broker.mqttdashboard.com

Port

1883

☐ Enable secure (SSL/TLS) connection

🔑 Client ID

Leave blank for auto generated

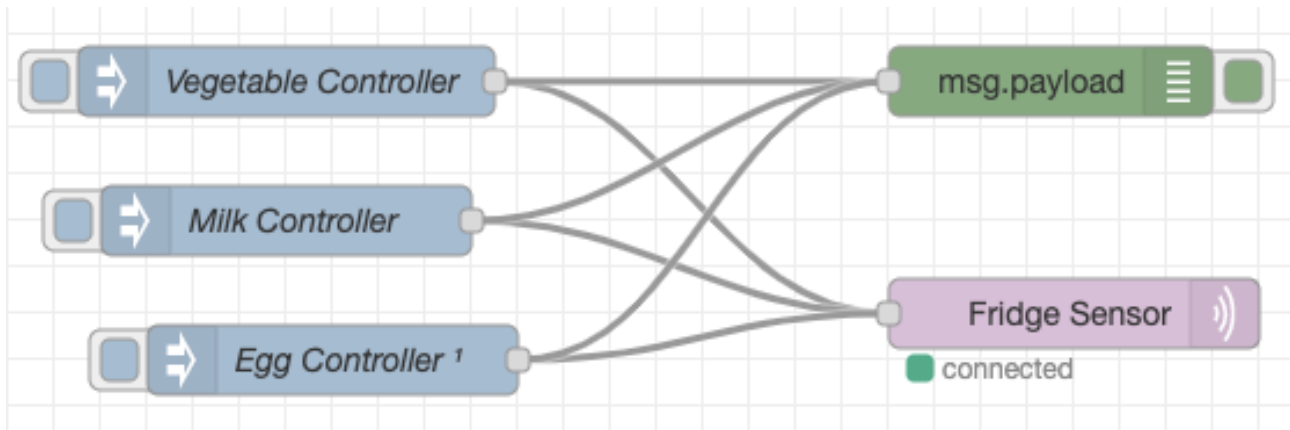
☐ Keep alive time (s)

60

☒ Use clean session

☐ Use legacy MQTT 3.1 support

### (3.5) Complete Circuit Flow




### (3.6) Complete Circuit Flow Code

```
[{"id":"5bb40bc2.5f7b94","type":"tab","label":"Refrigerator",
"disabled":false,"info":""},
{"id":"51f9f6c8.c31fb8","type":"inject","z":"5bb40bc2.5f7b94","name":"Vegetable Controller","props":[{"p":"payload"},
{"p":"topic","vt":"str"}],"repeat":"","crontab":"","once":false,"onceDelay":0.1,"topic":"Fridge Sensor","payload":"Tomatoes not available","payloadType":"str","x":160,"y":100,"wires":[["c52fe37.0882b2","a36731d5.8a1ba"]]},
{"id":"98d45325.9a81e","type":"inject","z":"5bb40bc2.5f7b94","name":"Milk Controller","props":[{"p":"payload"},
{"p":"topic","vt":"str"}],"repeat":"","crontab":"","once":false,"onceDelay":0.1,"topic":"Fridge Sensor","payload":"Three Packets of milk available","payloadType":"str","x":160,"y":160,"wires":[["c52fe37.0882b2","a36731d5.8a1ba"]]},
{"id":"af6fb88.ab7eb48","type":"inject","z":"5bb40bc2.5f7b94","name":"Egg Controller","props":[{"p":"payload"},
{"p":"topic","vt":"str"}],"repeat":"","crontab":"","once":true,"onceDelay":"0.2","topic":"Fridge Sensor","payload":"12","payloadType":"num","x":180,"y":220,"wires":[["c52fe37.0882b2","a36731d5.8a1ba"]]},
{"id":"a36731d5.8a1ba","type":"debug","z":"5bb40bc2.5f7b94","name":"","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":false,"statusVal":"","statusType":"auto","x":490,"y":100,"wires":[]},
{"id":"c52fe37.0882b2","type":"mqtt out","z":"5bb40bc2.5f7b94","name":"","topic":"Fridge Sensor","qos":"","retain":"","broker":"7d9cbc85.412e24","x":500,"y":200,"wires":[]},
{"id":"7d9cbc85.412e24","type":"mqtt-broker","z":"","name":"Refrigerator","broker":"broker.mqttdashboard.com","port":"1883","clientId":"","usetls":false,"compatmode":false,"keepalive":}
```

60","cleansession":true,"birthTopic":"","birthQos":"0","birthPayload":"","closeTopic":"","closeQos":"0","closePayload":"","willTopic":"","willQos":"0","willPayload":""}]

(3.7) HiveMQ Window

Not Secure | hivemq.com/demos/websocket-client/



**HIVEMQ**  
ENTERPRISE MQTT BROKER

Websockets Client Showcase

Connection

connected

Host

broker.mqttdashboard.com

Port

8000

ClientID

clientId-tWDJ7uNc7o

Disconnect

Username

Password

Keep Alive

60

Clean Session

x

Last-Will Topic

Last-Will QoS

0

Last-Will Retain

Last-Will Message

Publish

Messages

Subscriptions

Add New Topic Subscription

Qos: 2

Fridge Sensor

x

Color



QoS

2

Subscribe

Topic

Fridge Sensor

### (3.8) Output

```
23/10/2020, 14:08:45 node: a36731d5.8a1ba
Fridge Sensor : msg.payload : string[22]
"Tomatoes not available"

23/10/2020, 14:08:47 node: a36731d5.8a1ba
Fridge Sensor : msg.payload : string[31]
"Three Packets of milk available"

23/10/2020, 14:08:49 node: a36731d5.8a1ba
Fridge Sensor : msg.payload : number
12

23/10/2020, 14:08:50 node: a36731d5.8a1ba
Fridge Sensor : msg.payload : string[31]
"Three Packets of milk available"
```

**Publish**

**Messages**

2020-10-23 14:08:50	Topic: Fridge Sensor	Qos: 0
Three Packets of milk available		
2020-10-23 14:08:49	Topic: Fridge Sensor	Qos: 0
12		
2020-10-23 14:08:47	Topic: Fridge Sensor	Qos: 0
Three Packets of milk available		
2020-10-23 14:08:45	Topic: Fridge Sensor	Qos: 0
Tomatoes not available		
2020-10-23 14:08:39	Topic: Fridge Sensor	Qos: 0
Tomatoes not available		

**Subscriptions**

Add New Topic Subscription

Qos: 2

Fridge Sensor

X

## Result :

Thus, with the help of Node-RED we have implemented the MQTT protocol utilising HiveMQ, and have also visualised outputs using Node Red dashboard and have thus applied it in a practical use case scenario.