

# Week 9 - IoT Software

MQTT and Node Red



UNIT 3.

# Raspberry Pi with Node-RED

- 3.1. Initial Setting
- 3.2. Access Detection using Node-RED
- 3.3. Send MQTT Payload

# Install Node.js on Raspberry Pi ( Max at Node 16!!! )

- Latest NodeJS Installation

- Install the latest version of Node.js on your Raspberry Pi.

```
$ wget -O - https://raw.githubusercontent.com/meech-ward/NodeJs-Raspberry-Pi/master/Install-Node.sh | sudo bash;
```

- Check your Node.js version on Raspberry Pi.

```
$ node -v
```

```
pi@raspberrypi:~ $ node -v  
v16.10.0
```

- You can test with a simple script if your installation is succeeded.

```
pi@raspberrypi:~ $ node  
Welcome to Node.js v16.10.0.  
Type ".help" for more information.  
> var greeting = "Hello"  
undefined  
> console.log(greeting)  
Hello  
undefined  
>  
pi@raspberrypi:~ $
```

# Install Node-RED

- Before we start Installation, we should update existing libraries.
  - Keep the library up to date by entering the following commands in succession:

```
$ sudo apt update  
$ sudo apt upgrade
```

- Download Node-RED. (it takes quite some time.)
- Enter y for the two questions in the middle.

```
$ bash <(curl -sL https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered) --node16
```

### •Node-RED Installation Procedure

Running Node-RED install for user pi at /home/pi on raspbian

This can take 20-30 minutes on the slower Pi versions - please wait.

```
Stop Node-RED                ✓
Remove old version of Node-RED ✓
Node option not specified    :  --node12 or --node14
Leave existing Node.js        :  v16.10.0   Npm 7.24.0
Clean npm cache              -
Install Node-RED core        ✓  2.0.6
Move global nodes to local
Npm rebuild existing nodes
Install extra Pi nodes
Add shortcut commands
Update systemd script
```

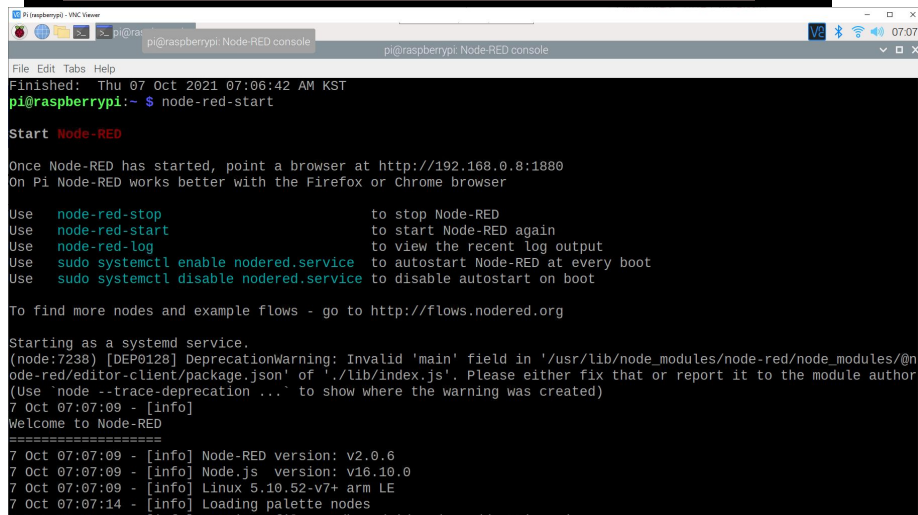
Any errors will be logged to /var/log/nodered-install.log

## Start Node-RED

- Start Node-RED

- Execute with Node JS through the following execution command:

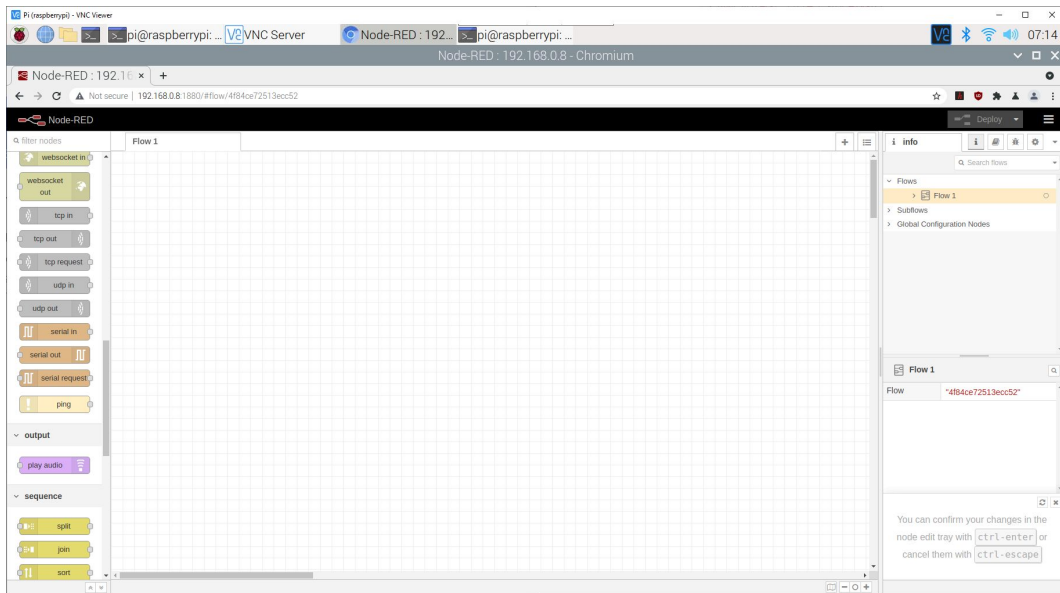
```
$ node-red-pi --max-old-space-size=256
```



```
pi@raspberrypi: Node-RED console
File Edit Tabs Help
Finished: Thu 07 Oct 2021 07:06:42 AM KST
pi@raspberrypi:~$ node-red-start
Start Node-RED
Once Node-RED has started, point a browser at http://192.168.0.8:1880
On Pi Node-RED works better with the Firefox or Chrome browser
Use node-red-stop to stop Node-RED
Use node-red-start to start Node-RED again
Use node-red-log to view the recent log output
Use sudo systemctl enable nodered.service to autostart Node-RED at every boot
Use sudo systemctl disable nodered.service to disable autostart on boot
To find more nodes and example flows - go to http://flows.nodered.org
Starting as a systemd service.
(node:7238) [DEP0128] DeprecationWarning: Invalid 'main' field in '/usr/lib/node_modules/node-red/node_modules/@nodered/editor-client/package.json' of './lib/index.js'. Please either fix that or report it to the module author
(Use 'node --trace-deprecation ...' to show where the warning was created)
7 Oct 07:07:09 - [info]
Welcome to Node-RED
=====
7 Oct 07:07:09 - [info] Node-RED version: v2.0.6
7 Oct 07:07:09 - [info] Node.js version: v16.10.0
7 Oct 07:07:09 - [info] Linux 5.10.52-v7+ arm LE
7 Oct 07:07:14 - [info] Loading palette nodes
```

- Start Node-RED

- Go to `http://{Raspberry Pi's IP Address}:1880` in your browser.
- Both NC browser and Windows browser are available, but here, use the browser in Raspberry Pi itself to access.



UNIT 3.

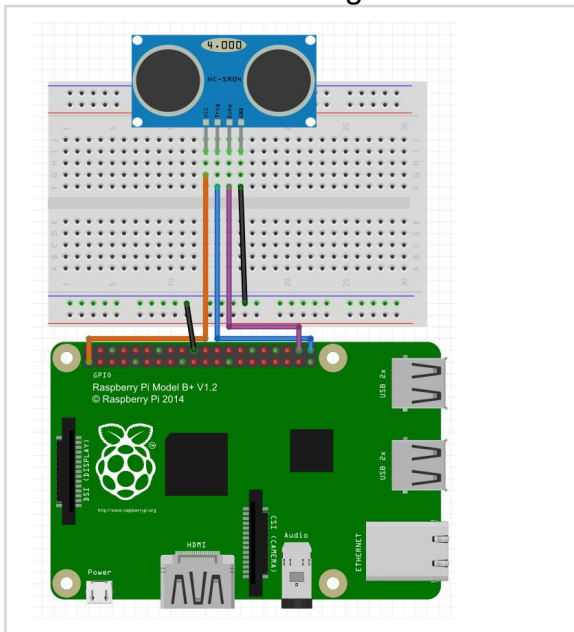
# Raspberry Pi with Node-RED

- 3.1. Initial Setting
- 3.2. Access Detection using Node-RED
- 3.3. Send MQTT Payload



# Access Detection using Node-RED

### • Access Detection using Node-RED

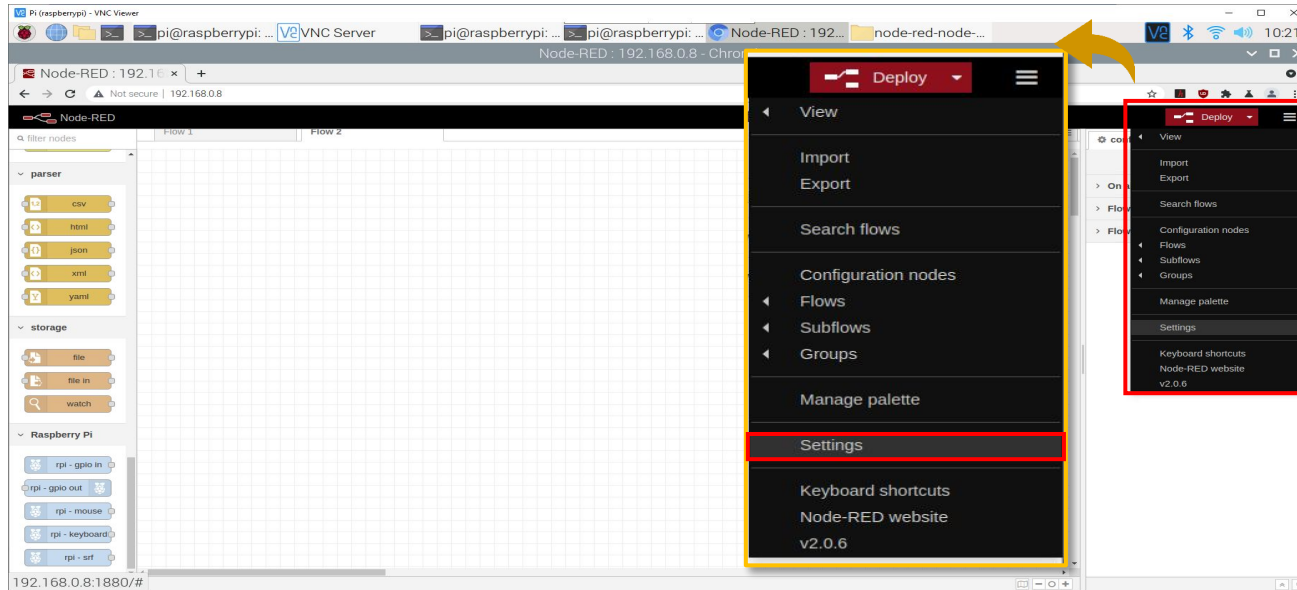


- An example of visualizing the measured distance by measuring the distance using the Ultrasonic Sensor with Node-RED.
- Circuit Diagram
  - HC-SR04 VCC – Raspberry Pi 3.3v
  - HC-SR04 GND – GNDs
  - HC-SR04 Trig – Raspberry Pi GPIO 21
  - HC-SR04 Echo – Raspberry Pi GPIO 20
  - Raspberry Pi GND – GNDs

# Install

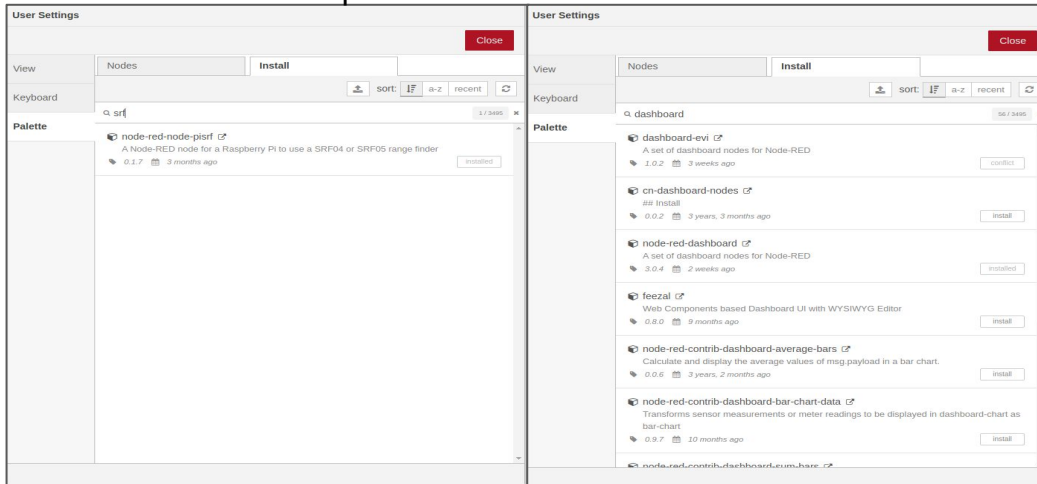
- Install Nodes

- To install Ultrasonic Sensor Node and Social Node, press Menu on the right side of the screen and click Settings.



# Install Nodes

- Install node-red-node-pisrf



- Search for srf in the Palette tab and install pisrf.
- Installation of the ultrasonic sensor to use in Node-RED
- Search dashboard and install node-red-dashboard.
- Installation to use Node-RED's dashboard node to provide simple visualization

### Add Nodes

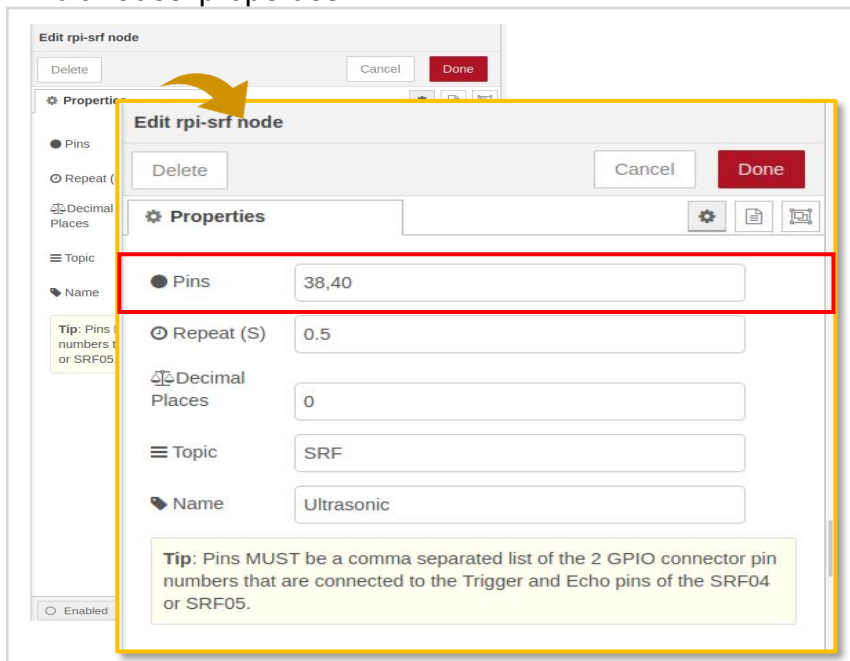
#### • Test example

The screenshot shows the Node-RED web interface in a browser. On the left, the 'Common' node palette is visible. A red box highlights the 'filter nodes' search input field at the top of this palette. A yellow box highlights the search results, which include 'Ultrasonic' and 'Distance' nodes. A red box highlights the 'Ultrasonic' node, and a yellow arrow points from it to a larger, detailed view of the node configuration shown in a separate yellow box. In this detailed view, the 'Ultrasonic' node is connected to a 'Distance' node, which is then connected to a 'msg.payload' node. The 'msg.payload' node displays the value '177.4'. The background of the main interface shows a flow with a 'debug' node and a list of log messages.

- ▶ Enter **srf**, **gauge**, and **debug** in the search input field on the top left where **filter nodes** is written, create nodes one by one, and connect them as shown in the figure on the left.
- ▶ Since the colors are separated, only the arrangement and connection are carried out, and the detailed setting is continued.

### Edit Nodes

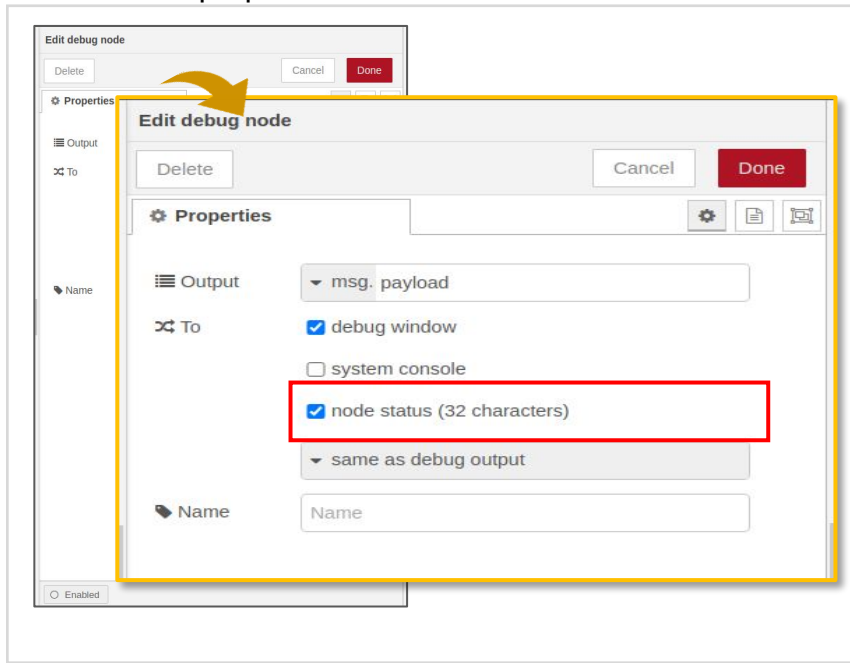
- Edit nodes' properties.



#### ▸ rpi-srf node

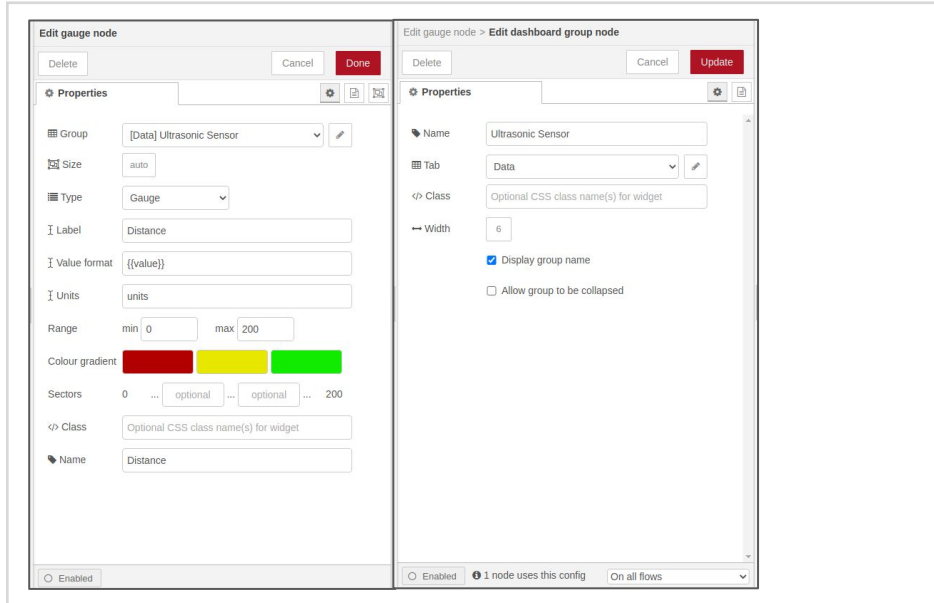
- One thing to note about pins is that you have to input Trig and Echo in order, and write the number of the pin itself, not the gpio pin number.
- Trig and Echo are connected to GPIO 21 and GPIO 20, respectively, and the number of these pins themselves are 38 and 40.
- Repeat : repeat cycle
- Name: Ultrasonic

- Edit nodes' properties.



- debug node
  - Check the node status.

- Edit nodes' properties.

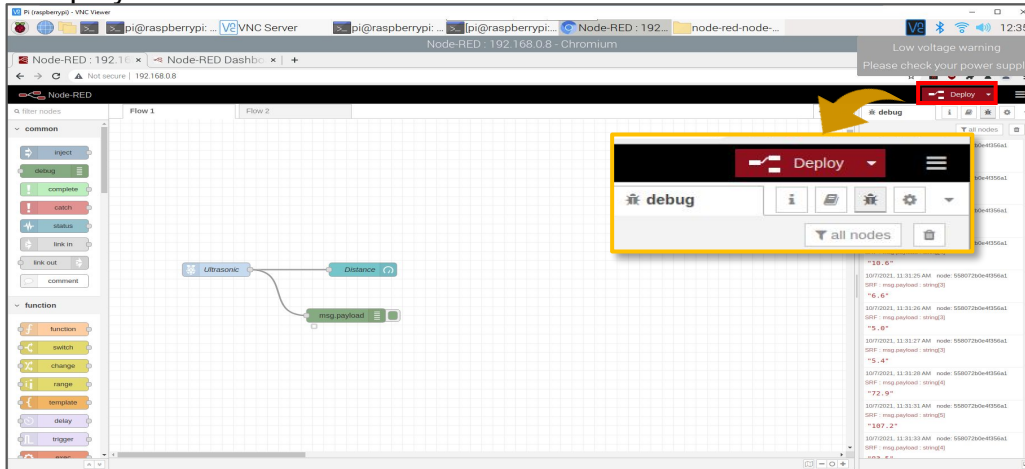


- ▶ gauge node

- Group: Edit nodes' properties through the expand tab and the edit dashboard group node button so that they are in the same state as the screen on the right .
- Label : Distance
- Range: 0 – 200
- Change colors freely
- Name: Distance

# Deploy Nodes

### • Deploy and Check Dashboard



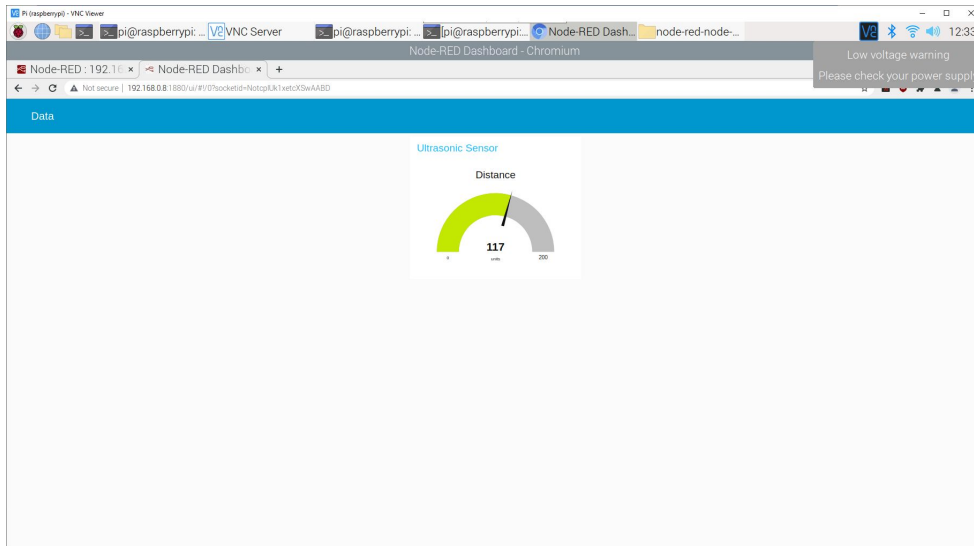
- ▶ Click the “Deploy” button to run the flow.
- ▶ If you click Debug messages, you can see that the observation values are output.



### Check Dashboard

- Deploy and Check Dashboard

- Connect to `http://Raspberry Pi's IP Address:1880:/ui`, so you can see gauge that changes according to the measurements of the sensor.



STOP, Now it's time  
for  
Quick DEMO!

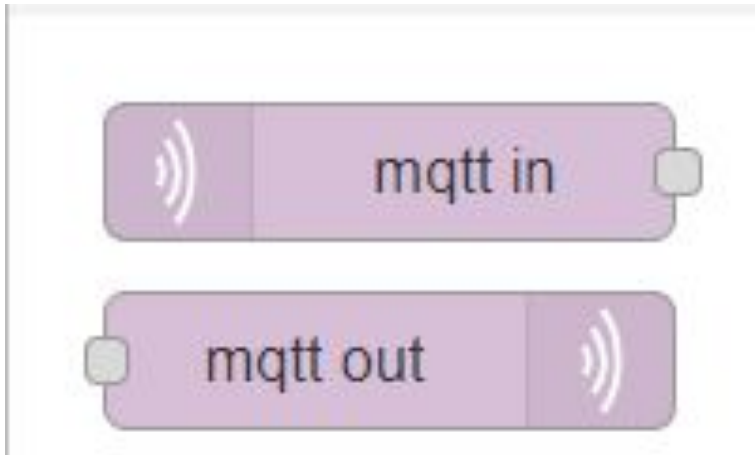


UNIT 3.

# Raspberry Pi with Node-RED

- 3.1. Initial Setting
- 3.2. Access Detection using Node-RED
- 3.3. Send MQTT Payload

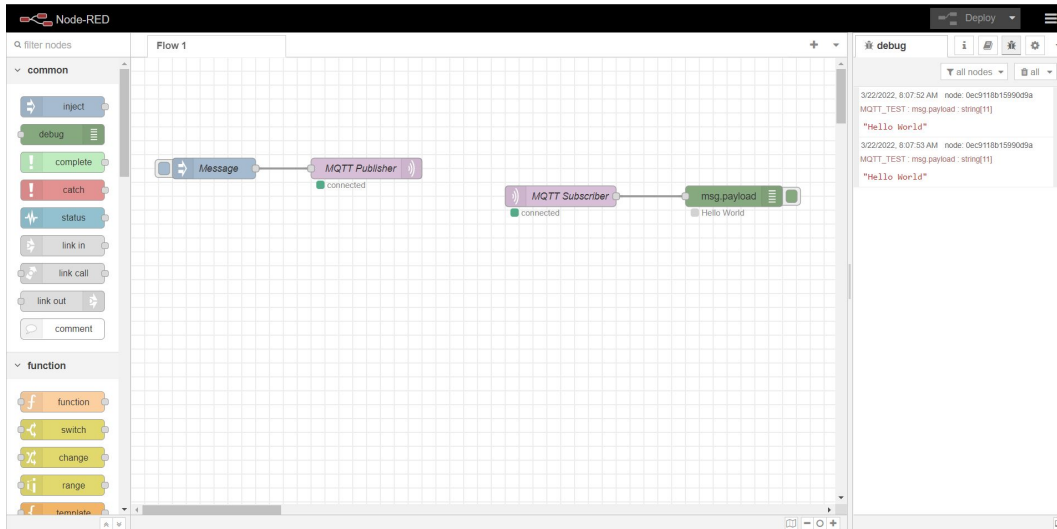
### MQTT Nodes



Untuk berinteraksi (*publish* dan *subscribe*) dengan MQTT Broker:

1. *mqtt in* merupakan *nodes* yang dapat kita gunakan untuk *subscribe* dari sebuah *topic* dan
2. *mqtt out* dapat kita gunakan untuk *publish* ke sebuah *topic* tertentu.

### Initial setup ( overview )



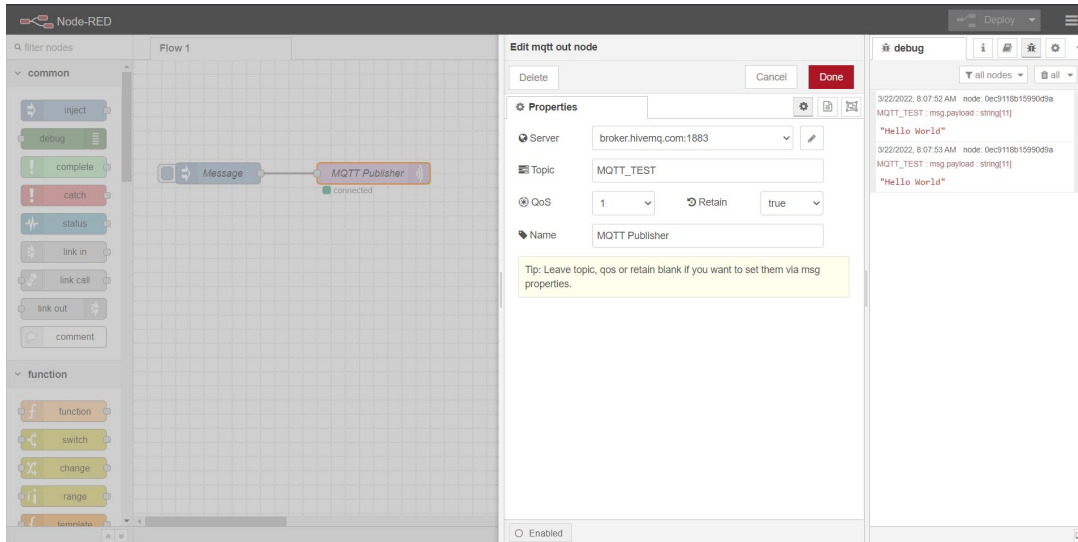
Kita dapat membuat sebuah skema sederhana untuk mempublish serta mensubscribe menggunakan dua nodes tadi seperti gambar di samping!

### Initial setup ( inject node )

The screenshot displays the Node-RED web interface. On the left, the 'common' node palette includes an 'inject' node. The main workspace shows a flow with a 'Message' node connected to an 'MQTT Publisher' node. The 'MQTT Publisher' node is marked as 'connected'. On the right, the 'Edit inject node' dialog is open. The 'Name' field is set to 'Message'. The 'msg.payload' is configured as a string 'Hello World'. The 'Inject once after' checkbox is unchecked, and the 'Repeat' dropdown is set to 'none'. The 'debug' console on the far right shows two log entries: '3/22/2022, 8:07:52 AM node: 0ec9118b15990d9a MQTT\_TEST : msg.payload : string[1]' and '3/22/2022, 8:07:53 AM node: 0ec9118b15990d9a MQTT\_TEST : msg.payload : string[1]', both displaying the payload 'Hello World'.

Dalam *nodes inject* kita perlu mendefinisikan `msg.payload*` sebagai sebuah *string* dan disini kita akan menampilkan data *string* Hello World\*.

### Initial setup ( mqtt out node )



Dalam *node mqtt out*, pertama kita perlu mendefinisikan URL dari MQTT broker kita dan pada contoh kali ini kita akan menggunakan MQTT Broker (broker.hivemq.com) yang *open for public* atau yang kita bisa akses secara gratis.

### Initial setup ( mqtt in node )

The screenshot displays the Node-RED web interface. On the left, the 'common' node palette is visible. The main workspace shows a flow with a 'Message' node connected to an 'MQTT Publisher' node. The 'MQTT Publisher' node is currently in a 'connected' state. On the right, the 'Edit mqtt in node' configuration panel is open, showing the following settings:

- Server: broker.hivemq.com:1883
- Action: Subscribe to single topic
- Topic: MQTT\_TEST
- QoS: 2
- Output: auto-detect (string or buffer)
- Name: MQTT Subscriber

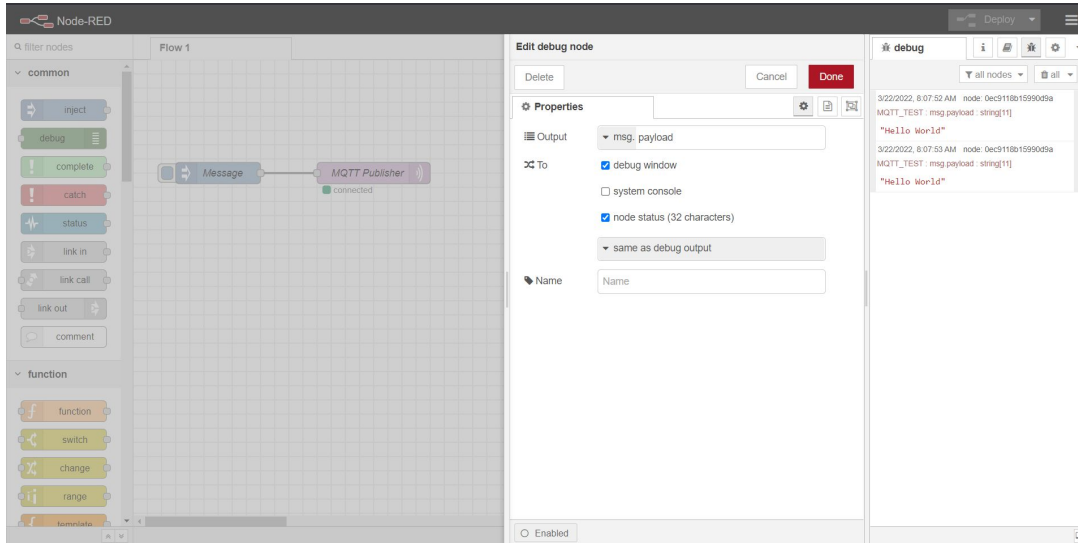
Below the configuration panel, the 'debug' console shows two log entries:

```
3/22/2022, 8:07:52 AM node: 0ec9118b15990d9a  
MQTT_TEST: msg.payload: string[1]  
"Hello World"  
3/22/2022, 8:07:53 AM node: 0ec9118b15990d9a  
MQTT_TEST: msg.payload: string[1]  
"Hello World"
```

Dalam *node mqtt in*, kita perlu memastikan bahwa di bagian server kita menggunakan server yang sama dengan apa yang sudah kita konfigurasi di *node mqtt out*, serta pastikan juga bahwa dibagian **Topic** kita menggunakan **Topic** yang sama dengan apa yang kita konfigurasi di *mqtt out*. Berikut adalah contoh konfigurasi dalam *node mqtt in*.



### Initial setup ( debug node )



Kemudian *setup node debug* seperti konfigurasi di bawah ini agar kita bisa melihat pesan yang masuk melalui *debug console*.

STOP, Now it's time  
for  
Quick DEMO!



# Challenge! [ Berhadiah! :) ]

Buatlah sebuah node-red flow dengan kondisi berikut

1. Dapatkan data sensor ultrasonic/menggunakan sensor yang ada/yang kalian pakai untuk project kalian dan kirimkan ke sebuah mqtt broker
2. Subscribe menggunakan node-red mqtt-in nodes dan tampilkan pada debug nodes