

Installing NodeJS, NodeRED and mqtt broker Mosquitto

1. Options for computers

These installations can be done on a normal Windows computer with Win 10 operating system. Please note that the installations typically write some information on C:\Users\... -folders. This means that only the user who made the installation will be able to run the system. There are instructions on the installation pages on how to install on system level. With system level installation all users will be able to run the system.

All these installations can be done on a linux computer, too. Even a Raspberry Pi will be enough to run the system. The instructions are given on installation pages of each component.

1. NodeJS

Look at the NodeRED installation guidelines which version they recommend to install of the Node.js. It is not necessary the latest version!

<https://nodered.org/docs/faq/node-versions#installing-nodejs>

Install the Node.js. Typically it is best to install to the default directory with default options.

<https://nodejs.org/en/download/>

On a computer with Windows operating system the Node.js files can be found in library C:\Program Files\nodejs .

To ensure that the installation was successful you can run following commands:

Using Powershell: `node --version; npm --version`

Using cmd: `node --version && npm --version`

The following lines with green print are from a Powershell command window.

```
PS C:\Users\xxxxxx> cd..  
PS C:\Users> cd..  
PS C:\> node --version;npm -version  
v12.13.1  
6.12.1  
PS C:\>
```

This installation was done earlier in 2019. You might already get higher version numbers.

2. Node-RED

The node-RED installation on Linux and on Windows operating systems is explained in the "Running Node-RED locally" pages

<https://nodered.org/docs/getting-started/local>

Installation especially on Windows computer is explained in pages

<https://nodered.org/docs/getting-started/windows>

The installation is done with command:

```
npm install -g --unsafe-perm node-red
```

```
PS C:\> npm install -g --unsafe-perm node-red
C:\Users\xxxx\AppData\Roaming\npm\node-red ->
C:\Users\xxxx\AppData\Roaming\npm\node_modules\node-red\red.js
C:\Users\xxxx\AppData\Roaming\npm\node-red-pi ->
C:\Users\xxxx\AppData\Roaming\npm\node_modules\node-red\bin\node-
red-pi

> bcrypt@3.0.6 install
C:\Users\xxxx\AppData\Roaming\npm\node_modules\node-
red\node_modules\bcrypt
> node-pre-gyp install --fallback-to-build

node-pre-gyp WARN Using request for node-pre-gyp https download
[bcrypt] Success:
"C:\Users\xxxx\AppData\Roaming\npm\node_modules\node-
red\node_modules\bcrypt\lib\binding\bcrypt_lib.node" is installed via remote
+ node-red@1.0.1
added 68 packages from 36 contributors, removed 73 packages and updated
108 packages in 29.004s
PS C:\>
```

This install the Node-RED into the system bath. The Node-RED can be started now on any directory.

The Node-RED installation can be tested on power shell command:

```
C:>node-red
```

```
PS C:\> node-red
6 Oct 17:35:34 - [info]

Welcome to Node-RED
=====

6 Oct 17:35:34 - [info] Node-RED version: v1.0.1
6 Oct 17:35:34 - [info] Node.js version: v10.16.3
6 Oct 17:35:34 - [info] Windows_NT 10.0.18362 x64 LE
6 Oct 17:35:35 - [info] Loading palette nodes
```

```
6 Oct 17:35:36 - [warn] Missing node modules:
6 Oct 17:35:36 - [warn] - node-red-node-email (0.1.27): e-mail, e-mail in
6 Oct 17:35:36 - [warn] - node-red-node-feedparser (0.1.12): feedparse
6 Oct 17:35:36 - [warn] - node-red-node-twitter (0.1.13): twitter-credentials,
twitter in, twitter out
6 Oct 17:35:36 - [info] Removing modules from config
6 Oct 17:35:36 - [info] Settings file : \Users\xxxx\.node-red\settings.js
6 Oct 17:35:36 - [info] Context store : 'default' [module=memory]
6 Oct 17:35:36 - [info] User directory : \Users\xxxx\.node-red
6 Oct 17:35:36 - [warn] Projects disabled : set
editorTheme.projects.enabled=true to enable
6 Oct 17:35:36 - [info] Flows file : \Users\xxxx\.node-red\flows_DESKTOP-
IIMRDS7.json
6 Oct 17:35:36 - [info] Server now running at http://127.0.0.1:1880/
6 Oct 17:35:36 - [warn]
```

Your flow credentials file is encrypted using a system-generated key.

If the system-generated key is lost for any reason, your credentials file will not be recoverable, you will have to delete it and re-enter your credentials.

You should set your own key using the 'credentialSecret' option in your settings file. Node-RED will then re-encrypt your credentials file using your chosen key the next time you deploy a change.

```
6 Oct 17:35:36 - [info] Starting flows
6 Oct 17:35:36 - [info] Started flows
6 Oct 17:35:37 - [info] [mqtt-broker:HAMK MQTT on xxxxxxxxxxxx] Connected
to broker: NodeRed1@mqtt://xxxxxxxxxxxxx:1883
```

On the organization ICT administration managed Windows computer the PowerShell command “node-red” caused a security error “.....because running scripts is disabled on this system....”. But the same command “node-red” on the same computer executed in the Command prompt did not cause any security errors!

You can pick up the Node-RED server address from the lines from PS terminal. In my installation the server is in local address

`http://127.0.0.1:1880/`

Please open that address with a browser.

The last lines on the command window are generated by the Node-RED flow already earlier created in the Node-RED.

To run the Node-RED the PS terminal window needs to be open. If you would like to use the Node-RED for an application and not just for testing and developing please read the instruction on page

<https://nodered.org/docs/getting-started/windows#running-on-windows>

Similar instructions are available for a linux installation.

3. MQTT broker Mosquitto

As an mqtt broker we might use the Mosquitto. An existing installation on some server can be used or the Mosquitto can be installed locally on the same computer where the Node.js and Node-RED are installed.

Instructions for installation are available at

<https://mosquitto.org/>

<https://mosquitto.org/download/>

You can download the suitable .exe file for the installation and run the file as administrator. The file can be for example the “mosquitto-1.6.7-install-windows-x64.exe” . The installation will be done with the default parameters.

The mosquito documentation can be found at

<https://mosquitto.org/man/mosquitto-8.html>

Please go the latest documentation page if you are using the latest version of the Mosquitto.

The mosquito files can be found under

C:\Program Files\mosquitto

Please study the content of the files

readme-windows.txt

mosquitto.conf

Don't close the the PowerShell or Command prompt terminal which might be open after the Node-RED installation. Please open a new PowerShell terminal.

To test your installation write in the power shell command prompt in the mosquito directory:

.\mosquitto

Some common mistakes and solutions from power shell are seen on the print below:

```
PS C:\Users> cd..
```

```
PS C:\> cd C:\Program Files\mosquitto
```

```
Set-Location : A positional parameter cannot be found that accepts argument
'Files\.....
PS C:\> cd "C:\Program Files"
PS C:\Program Files> cd mosquitto
PS C:\Program Files\mosquitto> mosquitto
mosquitto : The term 'mosquitto' is not recognized as the name of a cmdlet,
function, script file, or operable program.....
.....Windows PowerShell does not load commands from the current location
by default. If you trust this command, instead type: ".\mosquitto". See "get-
help about_Command_Precedence" for more details.
PS C:\Program Files\mosquitto> .\mosquitto
```

This will start the mosquito server. It starts with the configuration set on the file mosquito.conf .

The Mosquitto installation can be tested by opening a separate PowerShell command terminal and typing:

```
.\mosquitto_sub -v -t "test/firsttopic"
PS C:\> cd "C:\Program Files"
PS C:\Program Files> cd mosquitto
PS C:\Program Files\mosquitto> .\mosquitto_sub -v -t "test/firsttopic"
```

Leave that PowerShell terminal open. Open again a new PowerShell terminal. Type the command:

```
.\mosquitto_pub -t "test/firsttopic" -m "helloThere"
PS C:\Users\xxxxx> cd "C:\Program Files\mosquitto\"
PS C:\Program Files\mosquitto> .\mosquitto_pub -t "test/firsttopic" -m
"helloThere"
PS C:\Program Files\mosquitto>
```

Now you should get in the previous PowerShell terminal a new line:

```
test/firsttopic helloThere
```

To find out the IP address where the Mosquitto broker is listening please open once more a new PowerShell window and type:

```
netstat -an | findstr 1883
PS C:\> netstat -an | findstr 1883
TCP 0.0.0.0:1883 0.0.0.0 LISTENING
TCP 192.168.0.17:60736 193.167.167.59:1883 ESTABLISHED
TCP [::]:1883 [::]:0 LISTENING
TCP [::1]:1883 [::1]:60969 ESTABLISHED
TCP [::1]:60969 [::1]:1883 ESTABLISHED
PS C:\>
```

The Mosquitto broker is listening at address 0.0.0.0 in port 1883 .

4. Testing the installation

The installation can be tested with the flow seen on picture below:

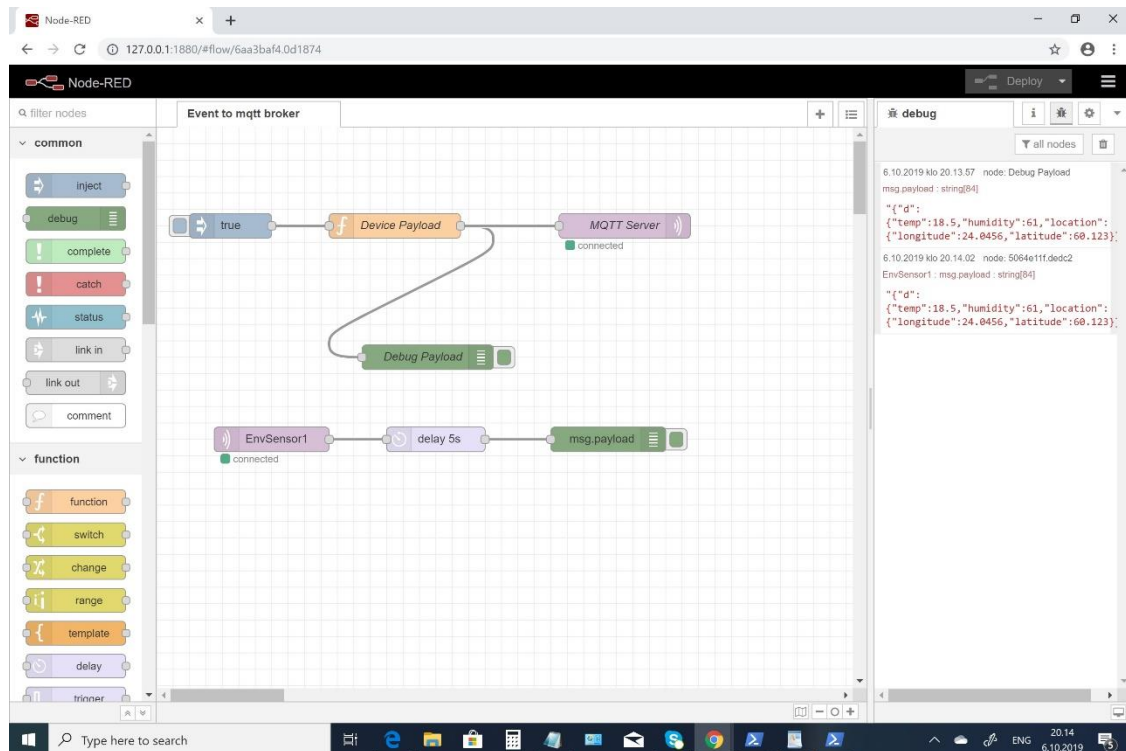


Fig. 4.1 The Node-RED flow for testing

The flow can be copied to your node red by copying the following JSON formatted text first into notepad to remove invisible symbols from the word or pdf file and then copying the text into clipboard. The clipboard can be imported to Node-RED.

```
[{"id":"6aa3baf4.0d1874","type":"tab","label":"Event to mqtt  
broker","disabled":false,"info":"","z":"63e03741.5d6cf8","type":"inject","z":"6aa3baf4.0d1874","  
name":"","topic":"","payload":"true","payloadType":"bool","repeat":"","crontab":"","once":false,"  
onceDelay":0.1,"x":90,"y":120,"wires":[["398df5fb.e9e4ea"]],{"id":"398df5fb.e9e4ea","type":"func  
tion","z":"6aa3baf4.0d1874","name":"Device Payload","func":"\nvar longitude1 = 24.0456;\nvar  
latitude1 = 60.123;\n\n// Array of pseudo random temperatures\nvar temp1 =  
[15,17,18.5,20,21.5,23,24,22.2,19,18];\n\n// Array of pseudo random relative humidities\nvar  
humidity1 = [50,55,61,68,65,60,53,49,45,47];\n\n// Counter to select from array.\nvar counter1 =  
context.get('counter1') || 0;\nncounter1 = counter1+1;\nif(counter1 > 9) counter1 =  
0;\ncontext.set('counter1',counter1);\n\n// Create MQTT message in JSON\nmsg = {\n  payload:  
JSON.stringify(\n    {\n      d:{\n        \"temp\": temp1[counter1],\n        \"humidity\":  
humidity1[counter1],\n        \"location\": {\n          \"longitude\": longitude1,\n          \"latitude\": latitude1\n        }\n      }\n    }\n  );\n  return  
msg;\n},\"outputs\":1,\"noerr\":0,\"x\":290,\"y\":120,\"wires":[[\"8f56361b.779e98\",\"36e34c50.b57c04\"]],{\"i  
d\":\"8f56361b.779e98\",\"type\":\"debug\",\"z\":\"6aa3baf4.0d1874\",\"name\":\"Debug  
Payload\",\"active\":true,\"tosidebar\":true,\"console\":false,\"tostatus\":false,\"complete\":\"payload\",  
\"x\":330,\"y\":280,\"wires":[[]],{\"id\":\"36e34c50.b57c04\",\"type\":\"mqtt  
out\",\"z\":\"6aa3baf4.0d1874\",\"name\":\"MQTT  
Server\",\"topic\":\"EnvSensor1\",\"qos\":\"0\",\"retain\":\"false\",\"broker\":\"426dbfae.68222\",  
\"x\":570,\"y\":120  
,\"wires":[[]],{\"id\":\"772b8d63.6c67f4\",\"type\":\"mqtt  
in\",\"z\":\"6aa3baf4.0d1874\",\"name":"","topic\":\"EnvSensor1\",\"qos\":\"0\",\"datatype\":\"auto\",  
\"broker\":  
\"426dbfae.68222\",  
\"x\":140,\"y\":380,\"wires":[[\"8e9a06ec.bdd718\"]],{\"id\":\"5064e11f.dedc2\",\"type\":  
\"debug\",  
\"z\":\"6aa3baf4.0d1874\",  
\"name\":\"\",  
\"active\":true,  
\"tosidebar\":true,  
\"console\":false,  
\"tostatus\":  
false,  
\"complete\":  
\"false\",  
\"x\":550,\"y\":380,\"wires":[[]],{\"id\":\"8e9a06ec.bdd718\",  
\"type\":\"delay\",  
\"z\":\"6a  
a3baf4.0d1874\",  
\"name\":\"\",  
\"pauseType\":  
\"delay\",  
\"timeout\":  
\"5\",  
\"timeoutUnits\":  
\"seconds\",  
\"rate\":  
\"1\",  
\"nbRateUnits\":  
\"1\",  
\"rateUnits\":  
\"second\",  
\"randomFirst\":  
\"1\",  
\"randomLast\":  
\"5\",  
\"randomUnits\":  
\"sec  
onds\",  
\"drop\":false,  
\"x\":340,\"y\":380,\"wires":[[\"5064e11f.dedc2\"]],{\"id\":\"426dbfae.68222\",  
\"type\":  
\"mqtt-  
broker\",  
\"z\":\"\",  
\"name\":  
\"LocalInstallation\",  
\"broker\":  
\"0.0.0.0\",  
\"port\":  
\"1883\",  
\"clientId\":  
\"\",  
\"usetls\":fals  
e,  
\"compatmode\":  
false,  
\"keepalive\":  
\"60\",  
\"cleansession\":  
true,  
\"birthTopic\":  
\"\",  
\"birthQos\":  
\"0\",  
\"birthPa  
yload\":  
\"\",  
\"closeTopic\":  
\"\",  
\"closeQos\":  
\"0\",  
\"closePayload\":  
\"\",  
\"willTopic\":  
\"\",  
\"willQos\":  
\"0\",  
\"willPayload  
\":\""}]
```

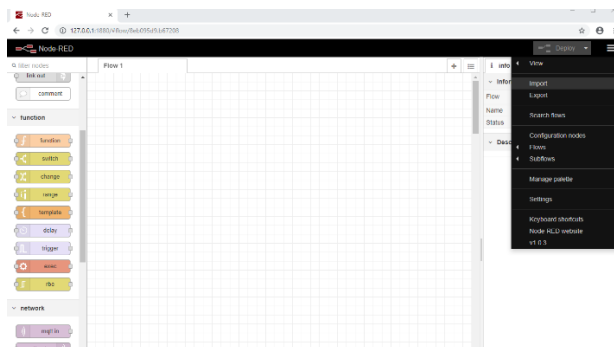


Fig 4.2 Click the Import on pull down meny on top right.

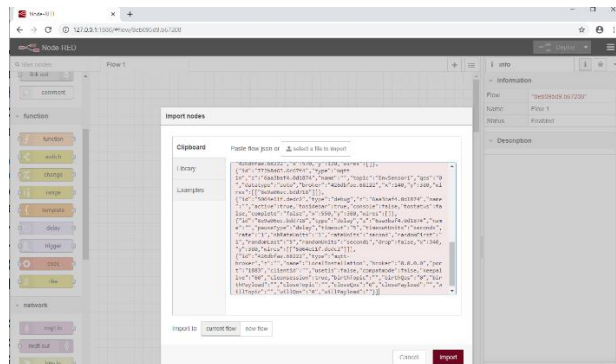


Fig 4.3 Import the JSON formatted text from the clipboard

The other option is to copy the file EventToMqttBroker_flow.json with the same Import tool.

Deploy the new flow in the NodeRED.

You may get a message about lost credentials.

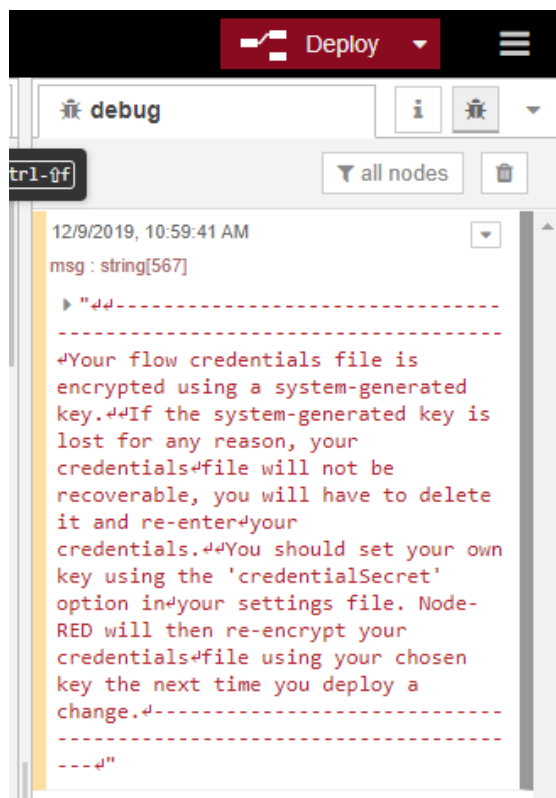


Fig 4.4 Lost credentials

To get it working again you should go to the library C:\Users\<your user name>\.node-red . Open the file settings.js . Uncomment the last line seen below:


```
// By default, credentials are encrypted in storage using a generated key. To
// specify your own secret, set the following property.
// If you want to disable encryption of credentials, set this property to false.
// Note: once you set this property, do not change it - doing so will prevent
// node-red from being able to decrypt your existing credentials and they will be
// lost.
credentialSecret: "a-secret-key",
```

Save the file. The NodeRED should be able to deploy the flow changes to the server.

Open once more a new Powershell command window.

Type there the settings to make it listen the same mqtt topic what is defined in the NodeRED flow.

In the NodeRED flow open the nodes MQTT Server and EnvSensor1 and read what is the topic there.

Enter the same topic in the Powershell command window.

In the Powershell command window you should get the same payload as in the NodeRED.

```
PS C:\Users\xxxx> cd..
PS C:\Users> cd..
PS C:\> cd "program files\mosquitto\"
PS C:\program files\mosquitto> .\mosquitto_sub -v -t "EnvSensor1"
EnvSensor1 {"d":{"temp":18.5,"humidity":61,"location":{"longitude":24.0456,"latitude":60.123}}}
```