

Accessing 1-wire device over owfs from Node-RED

In this article 1-wire devices, e.g. DS18B20, could be accessed from Node-RED and its values displayed on dashboard if needed.

This application note is based on **Debian Bullseye**, **Node.js v16.x** and depends on **owfs** and **node-red-contrib-owfs**¹ package.

1 Installation prerequisites

Install packages `ca-certificates` and `wget`; usually these are already installed.

```
$ apt-get install -y ca-certificates wget
```

2 Prepare owfs

First install `owserver`. You can install `owhttpd` and `owftpd`, but they aren't needed.

```
$ apt-get install -y owserver
```

Then connect an 1-wire bus master or a USB thermometer to a USB port. Once the device is connected, a device file is created, e.g. `/dev/ttyUSB0`. A corresponding device file could be seen in the kernel message buffer or in realtime with `dmesg -w` when the device is being plugged in.

Open `owserver` configuration file `/etc/owfs.conf` and comment out (prepend `#`) line beginning with `server` and including `FAKE` word to get rid of simulated (unreal) devices.

```
# This part must be changed on real installation
server: FAKE = DS18S20,DS2405
```

Then add a new line

```
server: passive = /dev/ttyUSB0 # for DS9097E and USB thermometer
```

or

```
server: device = /dev/ttyUSB0 # for DS9097U
```

Restart service `owserver`.

```
$ /etc/init.d/owserver restart # for sysvinit
$ systemctl restart owserver # for systemd init
```

¹<https://flows.nodered.org/node/node-red-contrib-owfs>

3 Install Node.js

Since Node-RED recommends² version 16, this version is going to be installed³.

```
$ wget -qO- https://deb.nodesource.com/setup_16.x | bash -  
$ apt-get install -y nodejs
```

4 Install Node-RED

This is a quick setup routine to get Node-RED up and running, not usable for worldwide production, e.g. no authentication, automatical start at boot is missing etc.

It is recommended that services run under user account, so create a system user *nodered*.

```
$ mkdir -p /home/nodered  
$ useradd --gid users --home-dir /home/nodered --system --shell /bin/bash nodered  
$ chown nodered -R /home/nodered
```

Login as user *nodered*.

```
$ su - nodered
```

Then install *node-red* and *node-red-contrib-owfs*.

```
# npm install node-red  
# npm install node-red-contrib-owfs
```

There could be errors when installing *node-red-contrib-owfs* with *bcrypt* package resulting in errors like *gyp ERR! configure error* or *ImportError: No module named ast*. In this case exit user *nodered* and install some packages as root and log in back to user.

```
# exit  
$ apt-get install -y g++ make python2.7  
$ su - nodered
```

As Node-RED was installed with *npm* without *global* flag, packages were installed under user directory *node_modules*. Run Node-RED from there.

```
# ./node_modules/node-red/red.js
```

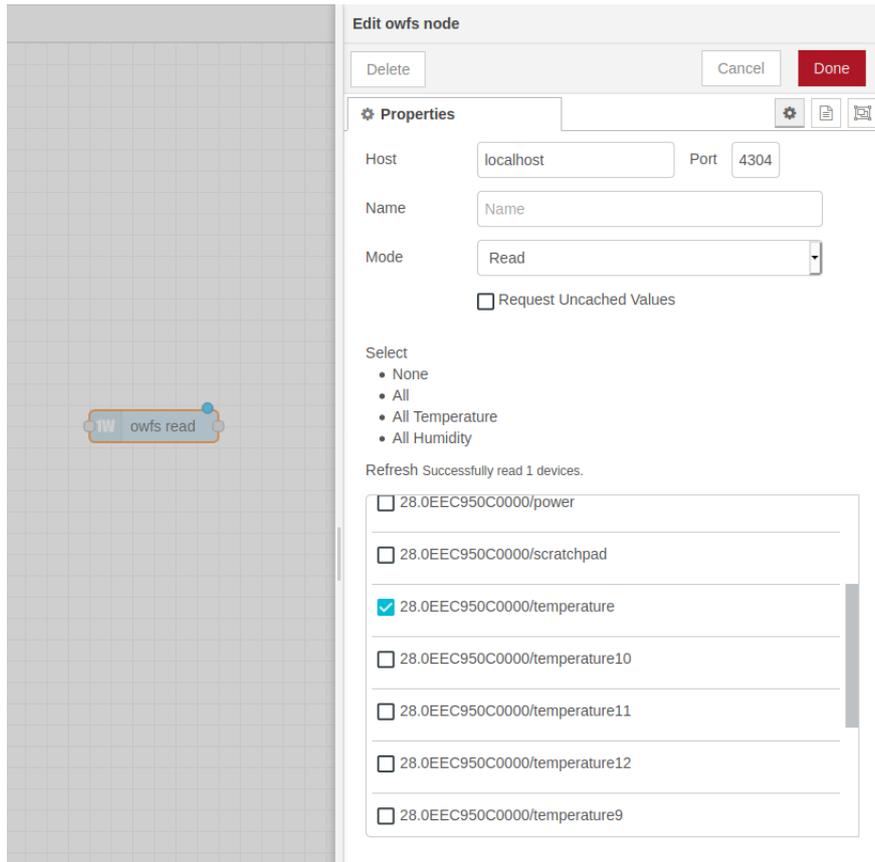
5 Accessing Node-RED

Open browser and navigate to *[x]:1880*, where *[x]* is IP address or a domain name of the computer running Node-RED. There will be 1-wire node on the bottom of node list and after adding (dragging) it

²<https://nodered.org/docs/faq/node-versions>

³<https://github.com/nodesource/distributions/blob/master/README.md>

to a flow it needs some configuration, e.g. select some interesting values like temperature.



Device list is automatically populated by detected 1-wire devices.

Create and deploy a simple flow to check working.



6 Troubleshooting

In case owfs node reports *Error: connect ECONNREFUSED 127.0.0.1:4304* open properties of owfs node and set host to `:::1` or in configuration file `/etc/owfs.conf` replace `localhost` with `127.0.0.1` at the line beginning with `server: port`.

```
server: port = localhost:4304
```