

Venus OS Large image: Signal K and Node-RED

1. Introduction

1.1 Venus OS Large, Node-RED and Signal K server?

This manual documents Venus OS Large, an extended build of Venus OS, that adds [Node-RED](#) and [Signal K server](#).

Node-RED is a tool for connecting hardware devices, APIs and online services in new and interesting ways. It provides a browser-based editor that makes it easy to wire together flows. With it, one can for example program a functionality such as activating a relay based on a temperature measurement. Or make far more complex algorithms, tying relays, measurements, or other data available from Venus OS or elsewhere together. All without having to write real source code, as Node-RED calls Low-code programming for event-driven applications.

Also Node-RED features a [fully customisable dashboard](#), viewable in a webbrowser - both locally but also remotely, via the VRM Servers.

Signal K server is aimed for yachts, and multiplexes data from NMEA0183, NMEA 2000, Signal K and other sensor inputs. It makes all that data available for Apps, such as [WilhelmSK](#), a highly customizable boat instrument display. The [Sailing with Signal K blogpost on our website](#) is a great read for more information and ideas.

1.2 Status and where this development is going

Currently, these images are available as beta versions. They work well, several users are using this without any issues; and are super happy with the provided functionality.

The remaining work are mostly small improvements, clean ups and testing.

There will be two flavours of Venus OS to install:

1. the normal one as we've been building for years now. With all normal & supported functionality.
2. the large flavour. All the same as the normal one, but with addition of Signal K & Node-RED.

Switching between them will be easy: a simple switch in the gui. Updating will be easy as well, the online updates will work, just like they do now for the normal images.

For now not that easy yet: installing and updating is done using the offline install method: with usb stick or sd-card.

Why not keep it all in one like it has been until now? To save download and install time for the (vast majority) of users that will not use these features. These new features increase the download (& installed) size enormously, and also they are really cool; but do not fall within normal Victron support.

Which is why we want to keep them separate.

More details of the development status here:

- <https://github.com/victronenergy/venus/issues/378> (this also contains the latest Venus OS Large release notes)
- <https://github.com/SignalK/signalk-server-node/issues/517>

2. Where to find support

There is and will be **no official support** on the functionalities included in this image. For support, turn to the [Modifications Space on Victron Community](#).

For support on SignalK, there is a #victron channel in [the SignalK Slack](#) workspace.

3. Requirements

3.1 GX Device compatibility

The compatible GX devices are:

- Cerbo GX
- Venus GX
- Octo GX
- MultiPlus-II GX
- EasySolar-II GX

The CCGX and CanVU GX are unfortunately not able to run Venus OS Large.

Venus OS large is also available for certain RaspberryPi models. Installation files below, and installation & other raspberrypi specific instructions [here](#).

3.2 CPU & Memory resources

Both SignalK as well as Node-red are rather resource intensive software applications to run on the GX Device. From a CPU usage point of view as well as RAM memory.

Ofcourse, how intensive depends on what kind of flows (node-red) or how many messages & handling of them (SignalK) happens in the system.

From a GX device point of view, from the support models the Venus GX is the one with the least resources. Running both SignalK and Node-RED at the same time will be “pushing it” to say it simply. The Cerbo GX is already much faster, as it has a dual core CPU as well as much more RAM. More details about CPU & RAM differences are in this document:

<https://github.com/victronenergy/venus/wiki/machines>.

When running out of RAM or CPU, the device will slow down. And eventually also reboot by itself due

to the watchdog.

3. How to install

Step 1. Preparations

- [Cerbo GX](#) - No preparations required.
- [Venus GX](#) - First install & boot v2.73,or a later version. And only thereafter install Venus OS Large. The reason for this is to make sure the partitioning of the flash is per requirements of Venus OS Large.
- [MultiPlus-II GX](#) - No preparations required.
- [EasySolar-II GX](#) - No preparations required.
- [RaspberryPi 3](#) - No preparations required; unless the Pi is running a version prior to v2.70. In that case, the sdcard will first need to be reflashed with a v2.70 or later version, using a .wic.gz file, [details here](#).
- [RaspberryPi 4](#) - No preparations required.

Note, when installing on a RaspberryPi, do make sure to familiarise with [the Venus OS for raspberrypi documentation](#) as well.

Step 2. Download the right file

Download the right file from [this dropbox link](#).

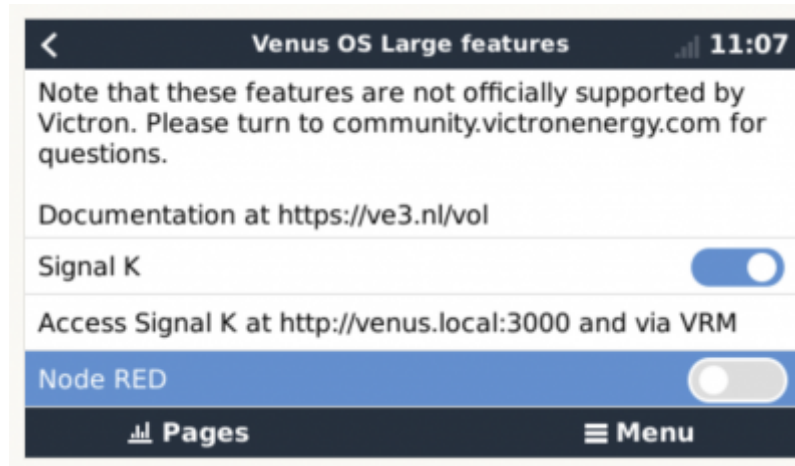
Which file to use?	
Cerbo GX	venus-swu-einstein...swu
Venus GX	venus-swu-beaglebone...swu
MultiPlus-II GX	venus-swu-nanopi...swu
EasySolar-II GX	venus-swu-nanopi...swu
RaspberryPi3	venus-swu-raspberrypi2...swu
RaspberryPi4	venus-swu-raspberrypi4...swu

The change-log is embedded in the (rather long) [issue 378 here](#).

Step 3. Install the file

1. Disable Auto updates. See Settings → Firmware → Online updates in Remote Console.
2. Install the downloaded file using the [GX Offline install method](#). Requires a USB-stick or Micro SD-Card.
3. Verify that the image was successfully installed: go to the Settings menu, and check that the Venus OS Large menu is available.
4. Enable either Signal K or Node-RED. Note that though its possible to enable both, doing that on a Venus GX is “pushing it”, the system might become slow. This is less of a problem on a Cerbo GX, as it has more resources. Installer access level is required to enable/disable these settings.

Visible once successfully installed, the Venus OS Large menu:



4. How to update

Unlike the normal Venus OS, this version can only be updated manually. Meaning by downloading the latest .swu file from above link; saving that on an SD-card or USB-stick, and inserting that in the GX device and then follow onscreen instructions.

Having the online update functionality work is still being worked on.

5. Node-RED

5.1 Introduction

Node-RED is a tool for connecting hardware devices, APIs and online services in new and interesting ways. It provides a browser-based editor that makes it easy to wire together flows. With it, one can for example program a functionality such as activating a relay based on a temperature measurement. Or make far more complex algorithms, tying relays, measurements, or other data available from Venus OS or elsewhere together. All without having to write real source code, as Node-RED calls Low-code programming for event-driven applications.

Also Node-RED features a [fully customisable dashboard](#), viewable in a webbrowser - both locally but also remotely, via the VRM Servers.

A good tutorial on how to work with Node-RED is here: <https://nodered.org/docs/tutorials/first-flow>.

The Node-RED editor is at <http://venus.local:1880/>. For some systems, you'll have to replace `venus.local` by the ip-address.

Please note that updating the `@victronenergy/node-red-contrib-victron` module is not (yet) possible from either the Node-RED UI as the commandline. The recommended way to get to the new `node-red-contrib-victron` version is to await a new Venus OS large build.

5.2 Installing extra nodes

Node-RED comes with a core set of useful nodes, but there are many more available from both the

Node-RED project as well as the wider community.

For example nodes that give access to weather data, or with timer functionality, and so forth.

You can search for available nodes in the Node-RED library.

The Victron Palette comes pre-installed.

Read the [Adding nodes to the Palette - guide by Node-RED](#) to learn how to install and manage node palettes.

To install new nodes/palettes requires a working internet access.

Palettes will be installed on the data partition on the GX Device. Which means that they will remain installed when updating Venus OS to a new version.

Refer to the remove button in the palette manager to remove self-installed palettes.

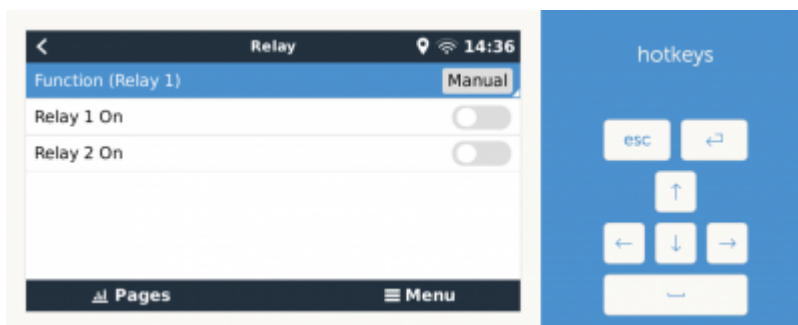
When installing a newer version of a palette than what is pre-installed in Venus OS, node-red will use that newer version.

Make sure that there is always 30MB of free space remaining on the data partition. TODO: how can a user see that? Also mem usage / mem free & some cpu stats would be nice to see. Maybe with a dashboard & flow? ;o)

5.3 Controlling relays

GX Device relays

To be able to control the relay(s) on the GX Device, configure them to be Manually controlled:



BMV Relay

The relay in a BMV can be controlled. Make sure to set its option in the BMV configuration to REM, which is short for Remote Controlled.

MPPT Relay

Currently it is not possible to control the relay in a MPPT Solar Charger, even though there is a Victron

node offering such functionality. The reason it doesn't work, is that VictronConnect does not have the required "Manual control" option in the list of selectable Relay modes for the MPPTs. This will be fixed at some point.

Note that only the larger model solar chargers feature a relay.

Multi & Quattro Relays

Includes alarm relays as well as the AC-out 2 relay: its not possible to control those; and there is no plan to make them controllable from within Node-RED.

5.4 Node-RED Dashboard

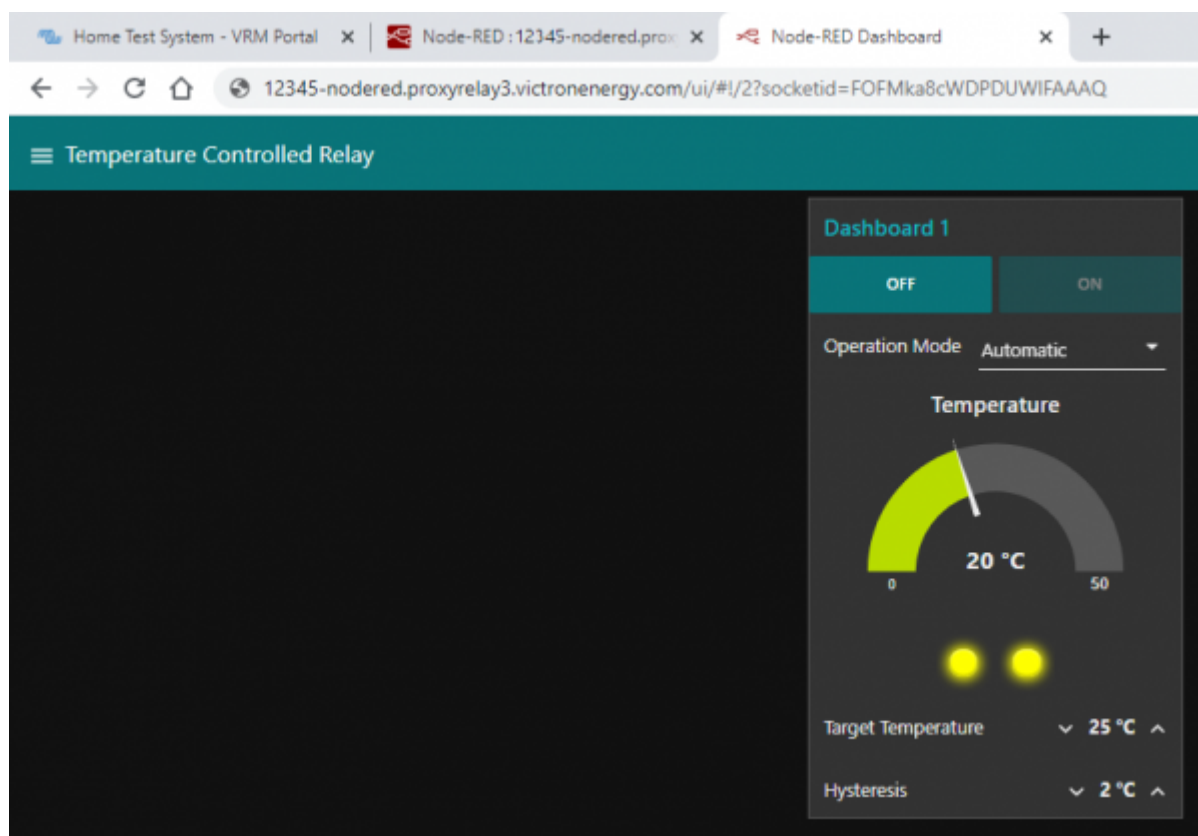
Besides the flow and automation functionality, Node-RED also offers a fully customisable dashboard. Including graphs, indicators as well as controls such as switches and sliders.

It is viewable in a web browser - both locally as well as remote, via the VRM Servers. See chapter 6.2 for how to see it remotely.

The dashboard needs to be installed, just like other extra palettes, using the Manage Palette feature, on the right of the screen in the menus.

Once that is done, you can find the nodes in the palettes on the left.

Example dashboard:



5.5. Node-RED FAQ

Q: How can I update the Victron node palette

As that palette is part of the total Venus OS large package, it can only be updated by installing a newer Venus OS large version.

In case you see that there is a newer version Victron palette available online, and that's not yet shipping in a Venus OS Large version, then the only option is to wait: a new Venus OS Large version is usually made available shortly after updating that palette.

Q: How can I password protect Node-RED

Password protection of Node-Red is linked to the remote console. If a password is set for the remote console, this password is also used for securing Node-Red. The username is admin, the password is the remote console password.

6. Signal K

6.1 Introduction

6.1.1 Why have Signal K on your GX Device?

By adding Signal K, the GX device becomes a data hub for all marine electronics on your yacht. For example audio, boat speed and wind data, and more. The Signal K is the interface between all those devices and protocols on one side, and mobile Apps on phones, tablets and the alike on the other. Note that SignalK itself is not a visualisation tool, its purpose is purely to gather all the data and make it available for web- phone-, and tablet-apps for the visualisation.

Also, there are SignalK plugins to forward the data to influxdb & Grafana.

Popular Apps compatible with Signal K:

- [Wilhelm SK](#) - WilhelmSK is a highly customizable boat instrument display available for iPhone, iPad, Apple Watch and AppleTV.
- ... (welcome to add more here)

6.1.2 Instruments compatible with Signal K

- All Victron products that can be connected to a GX device
- NMEA0183 and NMEA2000 GPS data, AIS data, Wind Temperature and other environmental data, tbd
- Engine data available on NMEA2000 and SAE 1939
- Fusion Stereo

And many more.

6.1.3 What is Signal K ?

Signalk-server is a software application that, in the context of this manual, runs on your GX Device and interface to all Victron protocols and more. It uses plugins, and there is a large community of enthusiasts around it, making it one of the most flexible and complete pieces of software that understands the plethora of protocols of all different manufacturers that make marine electronics.

For more information on Signal K, read [the Signal K website](#). And our blog post of a while ago.

Note that Signal K is not a Victron project. At Victron, we are great fans of sailing, open source as well as community driven projects. And Signal K is a perfect example of such project. And when running on a GX Device its a perfect marriage.

6.1.4 Signal K - the data format

Strictly speaking, Signal K is not software but a data format. And Signalk-server is the name of the software.

This section explains the rationale behind the data format.

Signal K is a modern and open data format for marine use. Its goal is to

- create a shared data model for marine related data - make marine related data easy to access, handle and build applications on

The shared data model allows you to handle the data in a uniform way independent of how the data is produced. In practice this means that you can refer to your battery voltage with ``electrical/batteries/1/voltage`` and not care where data came from: Victron BMV, NMEA 2000 bus or some other voltage sensor.

Easy access means that you can use modern standards like HTTP, WebSockets and JSON that are in widespread. There is a wealth of knowledge and skilled people are easy to find, with no special marine related knowledge needed to get started.

In Venus OS it brings all the data available in the GX system, including NMEA 2000 data, to a single data model and programming interface.

6.1.3 Considerations on how to install Signal K

The type of install described in this document, running Venus OS including Signalk, has these advantages: TODO elaborate. Doing this was born out of a wish to make Signalk more mainstream, less of a raspberrypi / hobbyists/tinkerers product.

The other common type of Signalk install, installed on a RaspberryPi running Raspbian, has the advantages of more flexibility, and ease of installing extra components such as Grafana or Influx. The disadvantage is that you are basically setting up a Linux computer yourself, which you'll also have to

maintain yourself. Whereas the Venus OS type is more of a complete product; no need to bother about keeping individual components up to date and worrying about their compatibility.

There is no one right way for everyone to do this; the best solution will depend on the requirement.

For raspberrypi owners, note that its very easy to try both methods and change between them: simply have two sdcards.

6.2 Where to find support?

As above already introduced, use [the Modifications space on Victron Community](#). As an extra, for Signal K we recommend the Victron channel in the [SignalK Slack](#).

Do not contact normal Victron support for any help on SignalK. Its based on community support, not official support.

6.3 Signal K details

TODO: remove these versions, and replace by a link to the change-log.

And instead add information on what specifics we do in the Signal K install.

1. Signalk-node version is 1.37.5
2. Victron Venus Plugin, v1.22.0

6.4 First use & configuring

Step 1. Opening the dashboard

Open a web browser, and go to the SignalK Dashboard by navigating to <http://venus.local:3000>.

Further documentation about that is ... (some link to signalk-node documentation?)

Step 2. Configure serial connections

Use this to, for example, configure a USB-connected NMEA0183 connection in SignalK. To do so, go to the dashboard (<http://0.0.0.0:3000>), and there to Server → Connections.

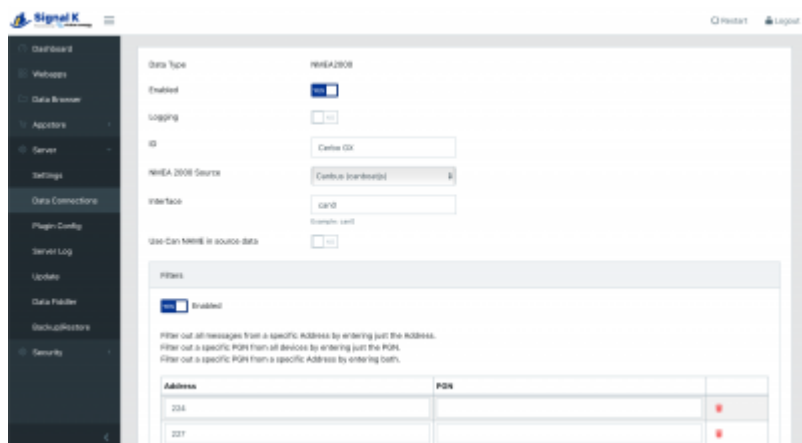
Make sure to not add a connection using any of the built-in serial ports (/dev/ttyO0, /dev/ttyO2, /dev/ttyO4, /dev/ttyO5). These are the VE.Bus- and VE.Direct ports on the Venus GX, and it is not necessary to configure them in SignalK. Data from any devices there connected is already available in SignalK, by means of the “Victron Venus Plugin”; see Dashboard → Server → Plugin Config.

Note that once configured in SignalK, for example a USB-serial port, that port is no longer normally available in Venus OS. For example, when plugging in a USB-GPS; and then adding a connection for its corresponding tty in the SignalK, makes for that GPS to be no longer be recognised and available in

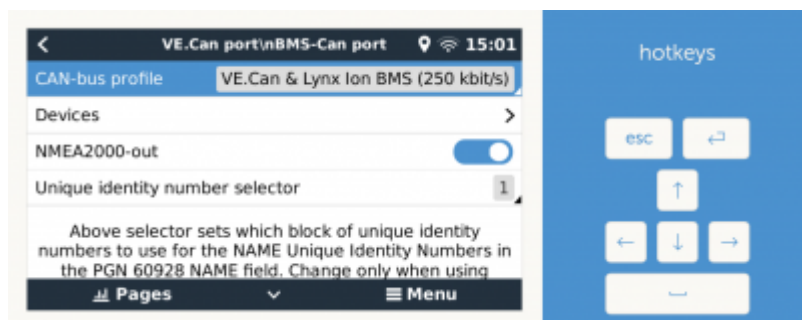
Venus OS.

Step 3. Configure can-bus connections

For the Venus GX, Signalk comes pre-configured with a NMEA2000 listener for the VE.Can ports, its named "n2k-on-ve.can-socket", and uses the canboatjs.



Note that, to keep that operational, that port must be configured as a 250kbit VE.Can & NMEA2000 port in the Remote Console.



Step 4. Adding Signalk Plugins or Webapps

TODO: explain & warn about data partition size & space (since thats where plugins are installed by signalk on Venus OS). Filling up the datapartition can corrupt Venus OS. More information about data partition on Venus OS here: <https://github.com/victronenergy/venus/wiki/data-partition>

6.5 Signal K FAQ

Q1 How to best handle & install a GPS

You want the data available in Signalk and in Venus OS, so that it gets to VRM. And preferably a quality GPS.

USB GPS Dongle

Plugging it into the GX Device will make it be recognised automatically; within Venus OS itself as well

as in Signalk, using the victron venus plugin. No need to configure anything.

NMEA2000 GPS

Connect it to the GX Device on the canbus port. Consider using our VE.Can to NMEA2000 cable to go from the NMEA2000 Micro-C cables to the RJ-45 on the GX Device.

In Venus OS v2.60, support for NMEA2000 GPS-es was added.

Signal K can then get it in two ways: directly from the NMEA2000 canbus in case thats configured as a data provider in Signal K. And via the native Venus OS functionality through the Victron-signalk-plugin.

Conclusion (untested), when using a NMEA2000 GPS, it will be available to both Venus OS and Signal K.

Part of a NMEA0183 network

Tbd. Actually a NMEA0183-network is probably something that doesnt exist: you can't have NMEA0183 in a network, its point to point.

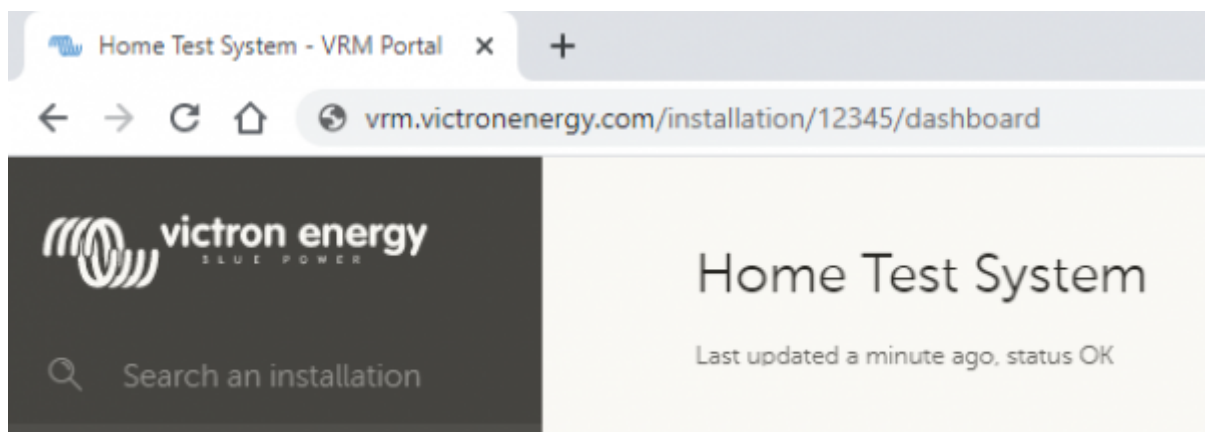
7. Remote access via VRM

7.1 Node-RED flow editor and Signal K

It it possible to access Node-RED flow editor and Signal K remotely via VRM - the interface and functionality is identical to access via LAN (ie. via <http://venus.local:1880> or <http://venus.local:3000>) .

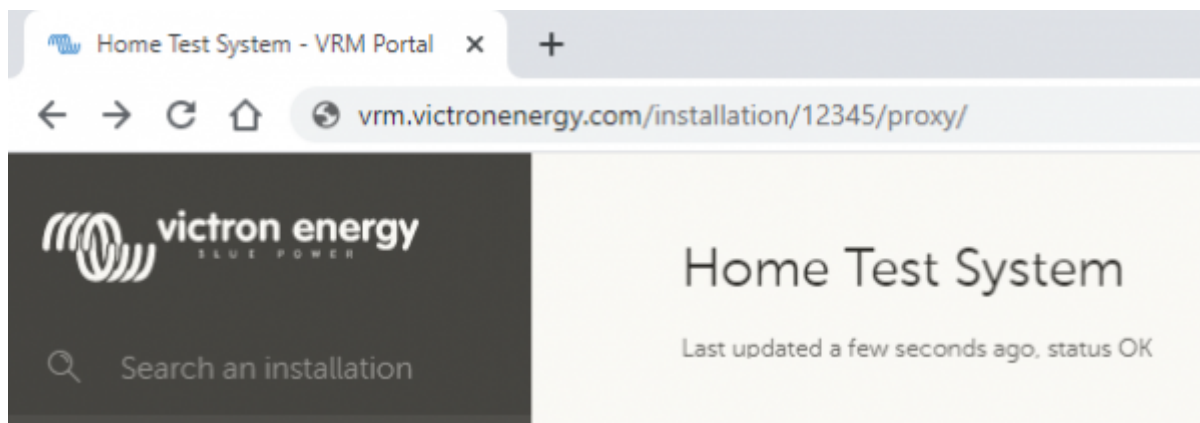
Step by step instructions

- **Step 1.** Log into/open the regular VRM 'dashboard' or 'advanced' page for the site

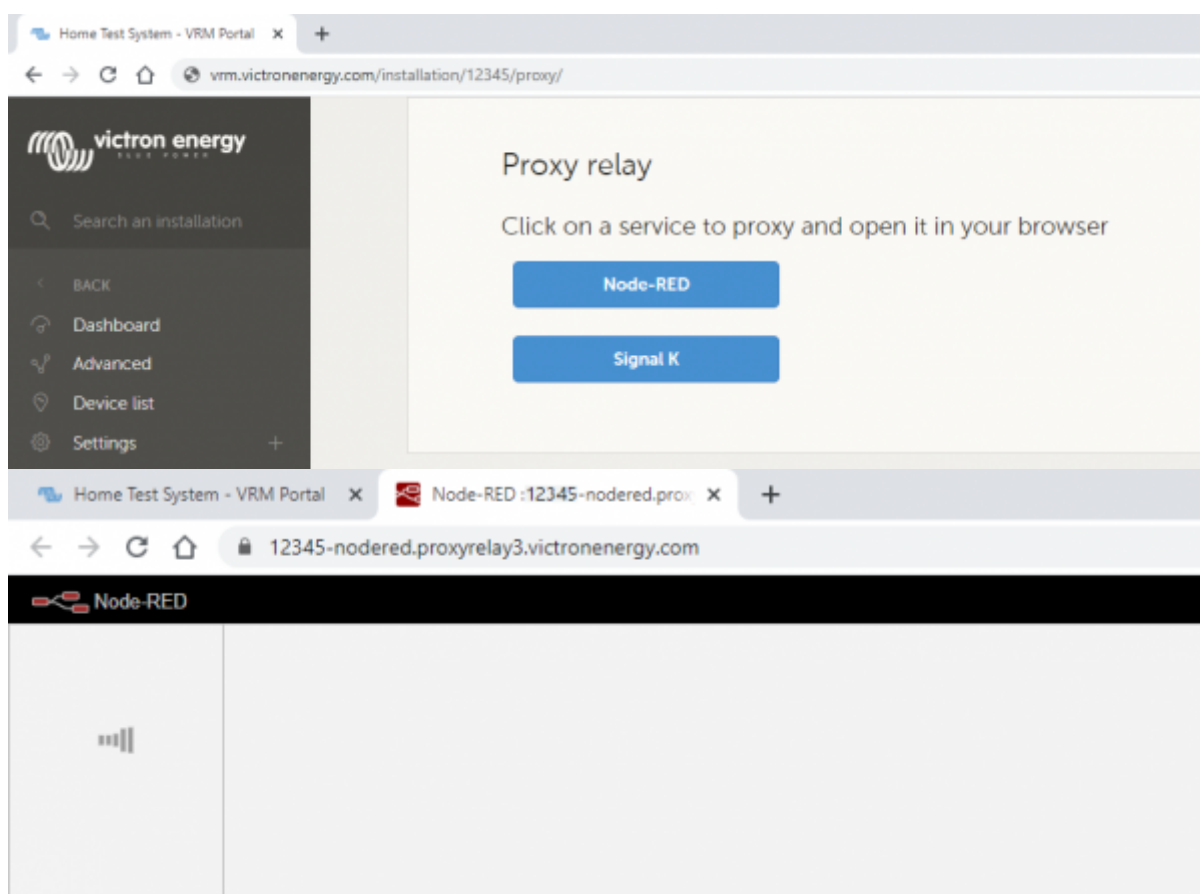


- **Step 2.** Modify the URL to include **/proxy/** after the site URL ID number (note that the last / is required)

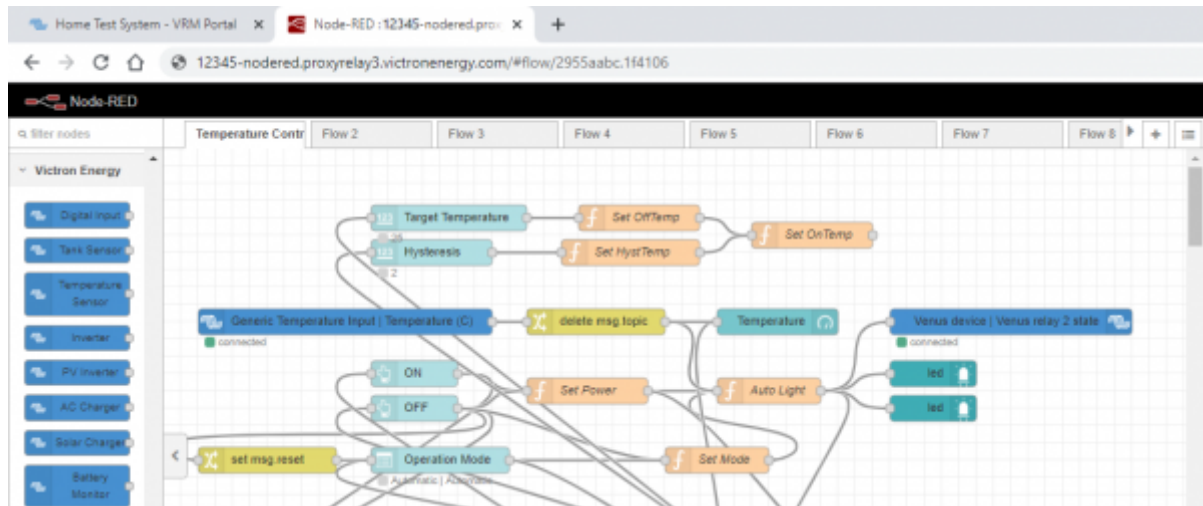
As an example: <https://vrn.victronenergy.com/installation/12345/proxy/>.



- **Step 3.** Click on the **Node-RED** or **Signal K** button in the Proxy relay screen; depending on location and the speed of internet connection, there may be a short delay while loading



- **Step 4.** The Node-RED flow editor or Signal K interface screen will appear



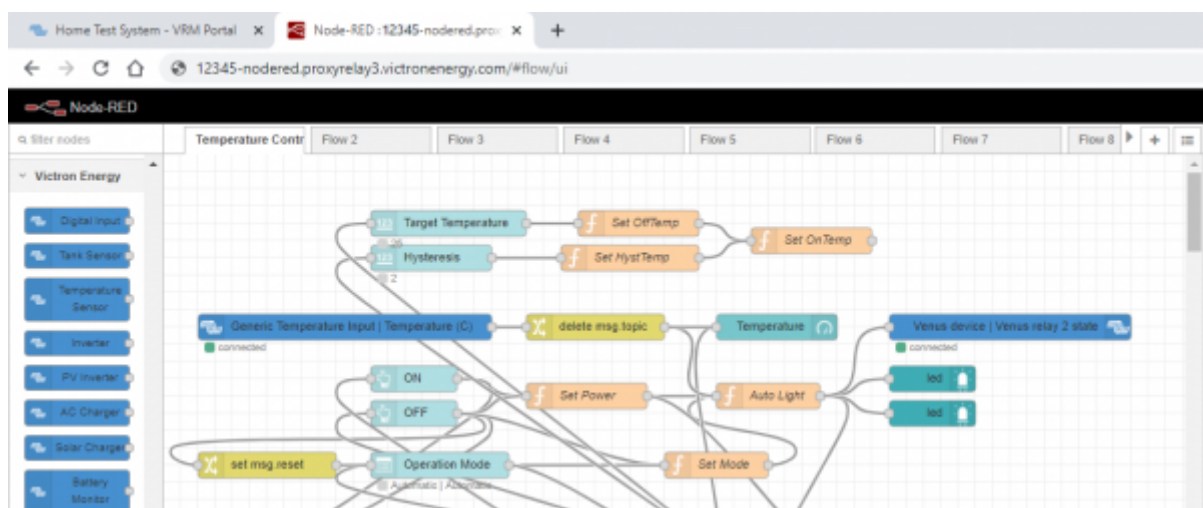
7.2 Node-RED Dashboard/User Interface

It is also possible to access the Node-RED dashboard/User Interface remotely via VRM - the interface and functionality is identical to access via LAN (ie. via <http://venus.local:1880/ui>)

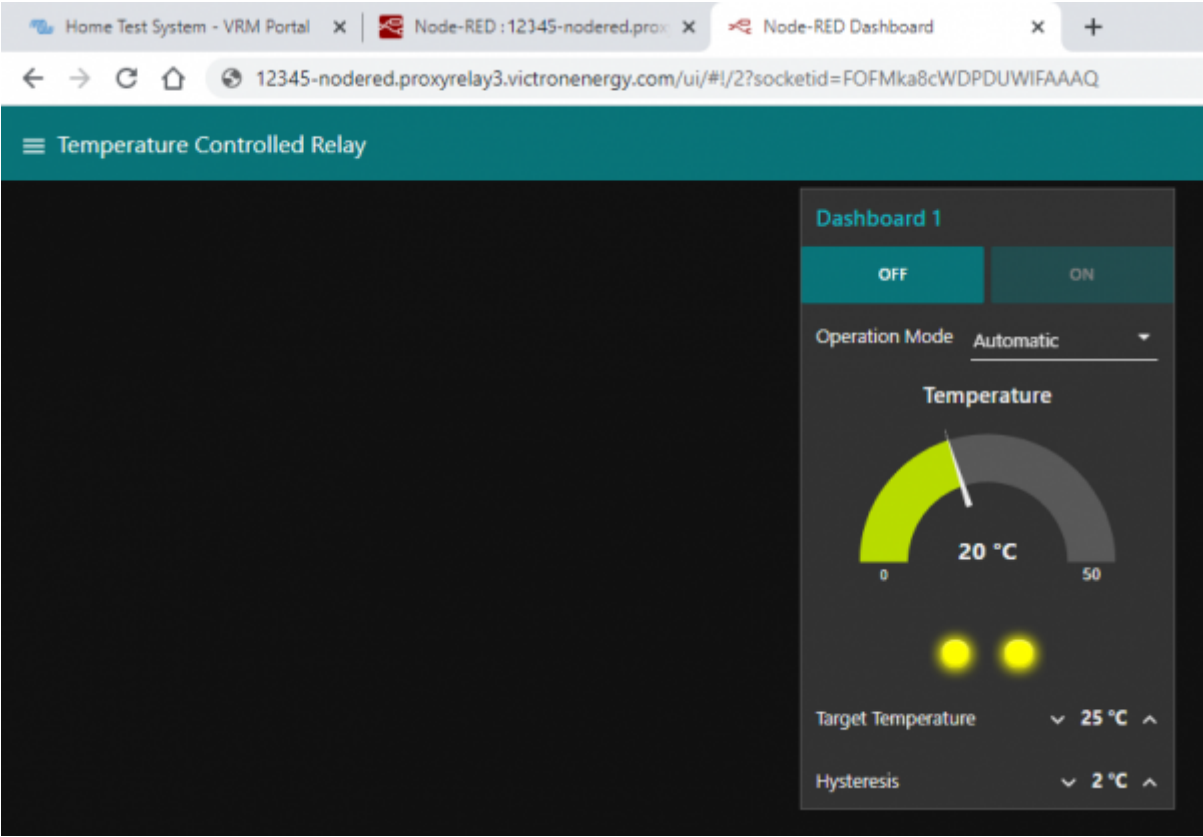
Step by step instructions

- **Step 1.** Install the Node-RED dashboard addon, see chapter 4.4
- **Step 2.** Follow steps #1 to 4 in section 6.1 above in order to 'open' the remote connection
- **Step 3.** The 'new' proxy URL needs to be modified to include **/ui** after **.....victronenergy.com**

As an example: <https://12345-nodered.proxyrelay3.victronenergy.com/ui> where **12345** is the unique site URL ID



- **Step 4.** The Node-RED dashboard/user interface screen will appear



From:
<https://www.victronenergy.com/live/> - **Victron Energy**

Permanent link:
<https://www.victronenergy.com/live/venus-os:large?rev=1631212628>

Last update: **2021-09-09 20:37**

