

# Venus OS operational commandline manual

## 1. Introduction

This page explains how to change settings and/or update firmware for various Victron products which are connected to a [GX device](#), for example the Color Control GX, via the command line, ie ssh.

This document is part of the Venus OS developer documentation. The main document is [the Venus OS wiki on github](#).

For more software developer oriented command line tips & tricks, see [the Venus OS commandline manual for software development](#) on the Venus wiki.

### 1.1 No support warning

Note regarding support: this document is made available for convenience only. There is no support on this type of using our products. Using the command line is outside normal use;

In case of questions, you can use the [Venus developers mailing list](#), or the [modifications space on Victron Community](#).

Our support staff on all official channels, Disqus and elsewhere, does not know how these tools can be used.

### 1.2 Use a normal tool instead of the command line

It is not at all necessary to use below command line instructions. We have far easier to use normal computer applications available:

#### VE.Bus products (Inverters, Multis, Quattros)

Changing settings with products featuring a VE.Bus comm. port (Inverters, Multis and Quattros) is normally done with VEConfigure.

Firmware updates are normally done with VEFirmware, a computer tool that is installed together with VEConfigure, or VictronConnect.

To change settings remotely, use [Remote VEConfigure](#).

Remotely updating firmware of a VE.Bus product is only possible from the command line, see below.

#### VE.Can products (Skylia-i, MPPT Solar Charger 150/75 & 150/85, and other products)

Updating VE.Can connected products is done with VE Power Setup, see [VE Power Setup manual](#). Both

for a local update, as for a remote update via the internet.

Changing settings of a VE.Can connected product (except for the Lynx Shunt VE.Can, which can be altered on the Venus GUI, and of course then also via the d-bus from the command-line). Normally done on the LCD panel of the product itself.

## **VE.Direct products (MPPTs, BMV, Phoenix Inverter VE.Direct, and other products)**

Both firmware updating as well as changing settings is normally done with [VictronConnect](#).

[VictronConnect Remote](#) allows remote changing of settings. It is not available from the command line.

Updating firmware remotely is available via the [Remote Firmware Update feature of VRM](#) (in the device list), and via the command line, see below.

### **1.3 Connecting to the command line and file transfer**

See the [ccgx root access page](#) for an explanation on how to connect to the command line.

Besides accessing the command line, you might also need to send files back and forth. Use the [Secure copy, or scp](#), for that.

## **2. Venus software updates**

Check firmware version:

```
root@CCGX:~# cat /opt/victronenergy/version
v2.06
Color Control GX
201612090913
```

Then to update, this depends on which version you are running. Up until v1.74 we were using opkg package updates. And then we switched to image based updates, and jumped with the version number to v2.00.

See [normal user documentation](#) for how to update from v1.74 (or earlier) to v2.00 (or later). It is not possible to that remotely or from the commandline.

### **2.2 Venus v2.00 and later**

Checking for updates and triggering a new update is all done with `/opt/victronenergy/swupdate/check-updates.sh`.

For an introduction, run `checkupdate.sh -help`. For more details, read [its code](#).

## 2.3 Venus v1.74 and before

### Updating to v2.00 and later

This needs to be done with an sdcard. There is no other way. See instructions here:

[https://www.victronenergy.com/live/ccgx:firmware\\_upgrade\\_to\\_v2](https://www.victronenergy.com/live/ccgx:firmware_upgrade_to_v2)

### Updating Venus version

To trigger an update check, run:

```
/opt/color-control/opkg-scripts/check-updates.sh
```

If there are new updates found, and if the GUI is running and automatic updates are enabled, the GUI will automatically initiate an update followed by a reboot. Otherwise, trigger the installation yourself with:

```
/opt/color-control/opkg-scripts/do-updates.sh
```

To change the Update-to setting, use set-feed.sh.

### Automatic update setting

To check if the device is configured to automatically update, query /Settings/System/AutoUpdate

```
dbus -y com.victronenergy.settings /Settings/System/AutoUpdate GetValue
```

The response will be either:

Status	Meaning
0	Automatic updates are disabled
1	Automatic updates are enabled

To change the setting, simply do the following (replace # with either 0 or 1):

```
dbus -y com.victronenergy.settings /Settings/System/AutoUpdate SetValue #
```

## 3. VE.Bus products

### 3.1 Changing settings

## Step 1. Reading the settings

To read the settings use:

```
/opt/color-control/mk2vsc -r -s  
dbus://com.victronenergy.vebus.tty01/Interfaces/Mk2/Tunnel
```

-s specifies the serial port to use. In the above example we use the DBus MK2 tunnel. This is available when the vecan-mk2 service is active, which it normally is. For a direct connection use -s /dev/tty01

Optionally -f can be used to specify a custom file name to write the settings to

The generated file (.rvsc or .rvms) can be opened and edited on a PC using [VE Configure](#)

## Step 2. Modifying the settings

Transfer the file to a computer running Windows, and which has the [VEConfigure toolset](#) installed.

On that computer, open the file by double clicking it, make the modifications, and save the file again.

Then transfer the new file back onto the [GX device](#).

## Step 3. Writing the new settings to the system

To write the settings use:

```
/opt/color-control/mk2vsc -w -s  
dbus://com.victronenergy.vebus.tty01/Interfaces/Mk2/Tunnel
```

## 3.2 Updating VE.Bus firmware

WARNING: At Victron we don't use this tool. Part of the code is shared with the VE.Bus firmware update functionality in VictronConnect, and therefore proven. But not all. So use at you're own risk.

Requires Venus v2.00 or later.

Using the vbdup tool. This can update firmware of the following products:

- Multi versions 26xxyyy/27xxyyy (minimum originally installed firmware, yyy needs to be 2xx or higher)
- MK3

Updating other VE.Bus devices, such as the VE.Bus BMS and Digital Multi Control panels is not possible with vbdup.

Updating a MultiGrid and also a MultiPlus-II is not (yet) possible with vbdup.

Make sure that the **GX device** (fe. CCGX), as well as routers or any other used communications equipment, are not powered by the VE.Bus devices. If they are, you'll lose power and connection as soon as the update is started, resulting in a dead system.

The product that you want to update needs to have application firmware running: just bootloader is not enough. The update starts with requesting the hardware IDs (VE.Bus discovery) from the running application. Those hardware IDs are then used to address the VE.Bus devices once in bootloader mode. After the discovery, the hardware IDs are stored in a file. In case, for example because of a power loss, the update fails, vbdup will automatically use the hardware IDs stored in that file.

Read the console help before use:

```
root@beaglebone:~# vbdup --help
vbdup version 0.05
```

Vebus device firmware updater.

Supported products:

- MK3
- Multi versions 26xyyy/27xyyy  
where yyy >= 2xx

**WARNING:**

Updating will reset any user defined settings to default!

If you want to keep your settings you must use VeConfigure, local or remote, to readout and save the settings.

After updating, you can use VeConfigure again to restore the settings.

Firmware updates currently do not work through the dbus tunnel. Please close the vecan-mk2 before using this feature

Note, When updating to Multi firmware version 4xx

Serial numbers were introduced with Multi firmware 4xx.

In VEFash the user is requested to enter the serial number manually.

This code has no user interaction. We do not know the update order when multiple Multi's are connected.

Therefore the serial number will be generated using the unique target ID

Serial number format = tid##### where ##.. is the hexadecimal representation of the unique target ID.

Existing serial numbers when updating within the 4xx range will remain unchanged

**Usage:**

**VBDUP OPTIONS:**

- f, --file  
vff/vfd vebus device firmware file.

**COMMON OPTIONS:**

- h, --help  
Print this help.

`--log-before <count>`

When set to non zero <count> number of log lines, which are not enabled themselves, are nevertheless shown before an log which is enabled. For example an error message will be preceeded by <count> warning and info messages which occurred before the error. Combined with `-v` also warning messages will be preceeded by info messages

`--log-after <count>`

Like `--log-before`, but temporarily enables tracing of disabled messages after a warning or error occurred.

`-s, --serial`

the serial port to use (default `/dev/ttyS0`).

`-i, --use-host-io-lines`

Use host processor IO lines to control MK2/MK3 reset and VE.Bus standby.

`-v --verbose`

Enable log messages. Specifying multiple times increases verbosity.

Victron Energy B.V.

Prior to updating a Multi the vecan-mk2 service should be stopped:

```
svc -d /service/vecan-mk2
```

To update run:

```
vbdup -f 2610408.vff -vvv -s /dev/tty01
```

`-vvv` is not required and only included to output messages on the console in order to see the tool at work.

Start the vecan-mk2 up again by:

```
svc -u /service/vecan-mk2
```

## 4. VE.Can products

First simply run `vup` to see connected products, and then follow onscreen instructions. No need to stop the canbus service, you can run an update, and when done, it will automatically recognize the updated product.

```
root@bpp3:~# vup
Querying devices
| NAD  | man | serial          | boot  | bmode| vup| vup#| id   |
description      | version |
```

0x20	358	0013122	1.02	-	+	0	100	BMV
to NMEA2000 interface		1.06						
0x20	358	0013122	1.02	-	-	1	x	x
x								
0x24	358	0013463	HQ1227H7JSJ   1.00	-	+	0	108	
BlueSolar Charger MPPT 150/70			2.02					

To update the first device in the above list, use the following command:

```
vup -n 0x20 -f new_firmware.vup
```

For other devices, change the address, as found in the first column (NAD). The name of the firmware file depends on the product.

E.g. `skyllai_control_v1.10.vup` for updating a Skylla-i Control to version 1.10.

## 4.2 Restoring VE.Can products to their defaults

WARNING: for chargers etc this is a bad idea to do remotely if you are not sure what you are doing, since charge voltages etc might be reset as well. It needs a ccgx version  $\geq 1.30$ . `-n` is address, the same as for `vup`, see above (and just type `vup` to get it).

```
vreg -n 0x24 -s 4
```

## 4.3 Restarting a VE.Can device

A bit of a trick: below code first sets them to bootloader mode, and then back out of it:

```
root@ccgx:~# vup -n 0x24 -u
Querying devices
Found device, enabling UDF.
UDF is ready to accept update.
Udf enabled, update can begin.

root@ccgx:~# vup -n 0x24 -e
Querying devices
Update end send
```

## 4.4 vreg: changing settings from the command line

```
# vreg --help
vreg - Victron registers

-b, --bin
    Output in binary format.

-g, --get
```

Get a register.

-n, --nad

The NAD of the target device.

-r, --run

Keep running.

-s, --set

Set a register. Data is set with e.g. un16:2800 un16:12

#### COMMON OPTIONS:

-c, --canbus

Specifies the CAN-bus interface to use, formatted as [driver]:[channel].  
kvaser:1 will for example select the first Kvaser channel.

-c ? will list the available gateways.

-d, --dump

Dump the contents of the CAN bus frames. Specifying multiple times increases verbosity.

-h, --help

Print this help.

--log-before <count>

When set to non zero <count> number of log lines, which are not enabled themselves, are nevertheless shown before an log which is enabled. For example an error message will be preceeded by <count> warning and info messages which occurred before the error. Combined with -v also warning messages will be preceeded by info messages

--log-after <count>

Like -log-before, but temporarily enables tracing of disabled messages after a warning or error occurred.

--silent

Only listen, no messages or acks are sent

-v --verbose

Enable log messages. Specifying multiple times increases verbosity.

Victron Energy B.V.

Switching all devices on VE.Can off (careful, when having NMEA2000-out enabled, or having an Inverter RS, also AC will be switched off!):

```
vreg -c socketcan:can0 -s 0x0200 un32:0x0000004
```



switching a specific device off:

```
vreg -c socketcan:can0 -s 0x0200 un32:0x0000004 -n 0x24
```

Note that in above example the VREG data field (un32) is padded with zeros; current firmware requires that, future fw won't any more.

Switch a specific device on:

```
vreg -c socketcan:can0 -s 0x0200 un32:0x0000001 -n 0x24
```

Seeing which devices are off and which are on goes like this. Press ctrl-C to stop the command.

```
vreg -c socketcan:can0 -g 0x200 -r
```

## 5. VE.Direct products

### 5.1 See connected devices

```
# svstat /service/* | grep vedirect
/service/vedirect-interface.tty02: up (pid 2020) 1210220 seconds, normally
down
/service/vedirect-interface.tty04: up (pid 2105) 1210218 seconds, normally
down
```

### 5.2 Firmware update

First stop the driver:

```
/opt/victronenergy/serial-starter/stop-tty.sh /dev/ttyUSB0
```

Then update the product:

```
root@bpp3:~# dup -s /dev/tty02 -f filename.dup
VE.Direct Updater
Updating device on /dev/tty02
Locating device
Found BlueSolar Charger MPPT 75/15 (A042)
Retrieving calibration data
Found application v1.14
Entering bootmode
Found bootloader v1.01
Erasing device
Programming
Starting new firmware
Verifying new version
Found bootloader v1.01
```

```
Update successful
Restoring calibration data

root@bpp3:~#
```

and when done, release the tty back to the serial-starter:

```
/opt/victronenergy/serial-starter/start-tty.sh /dev/ttyUSB0
```

## 5.3 Changing settings & other remote commands

See here: <https://github.com/victronenergy/venus/wiki/dbus-api>.

That API allows to read and write VREGs. See MPPT, BMV and other HEX Protocol documents on our white paper section on the website for more information. Once you understood it all; its possible to reset a BMV too 100% with a one liner from the command line, ie “dbus -y com.victronenergy.battery.ttyO1 [find the right path] SetVreg [VREG] [DATA].

The API will be used for a feature thats in progress: remotely changing settings of MPPTs, BMVs and other products connected to a GX device on the internet.

## 6. Screenshot

To make a screenshot press and hold the upbutton, and then press the down button. You'll then find `/opt/color-control/gui/snapshot.png`

## 7. Enabling Remote Console

If you have a CCGX already installed, but Remote Console on VRM is not enabled, follow these steps to enable it remotely:

1. SSH into CCGX (make a root password by following instructions [here](#))

```
1. dbus -y com.victronenergy.settings /Settings/System/VncInternet
   SetValue 1
```

```
2. dbus -y com.victronenergy.settings /Settings/System/VncLocal
   SetValue 1
```

```
3. touch /data/conf/vncpassword.txt
```

2. Reboot CCGX via SSH:

```
reboot
```

3. Now you should be able to access Remote Console feature from VRM
4. If you want to set a password, go to Settings → Remote Console → Enable password check

## 8. Enabling VRM two-way communication (Remote VEConfigure aka vrmpubnub)

If you have a CCGX already installed, but VRM two-way communication is not enabled, follow these steps to enable it remotely:

1. SSH into CCGX (make a root password by following instructions [here](#))
  1. `dbus -y com.victronenergy.settings /Settings/Services/Vrmpubnub SetValue 1`
2. VE.Configure should now be able to connect to the CCGX

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

Permanent link:  
[https://www.victronenergy.com/live/open\\_source:ccgx:commandline?rev=1614768403](https://www.victronenergy.com/live/open_source:ccgx:commandline?rev=1614768403)

Last update: **2021-03-03 11:46**

