Hub-4 mode 2 and 3

This page explains how to use a Multi/Quattro as a bidirectional inverter operating parallel to the grid, integrated into a customer designed system (PLC or other). In mode 2, Advanced, or more 3, Custom.

Note that it is also possible to run a Victron grid parallel storage system without designing and implementing your own control loop, logic or measurements. See the Hub-4 manual.

For firmware version requirements, refer to the main hub-4 manual.

Three/split-phase systems are not possible yet, expected date is 2016 Q2. No hardware changes will be necessary, there will only be firmware upgrades.

1. Overview of Hub-4 operating modes

1.1 Mode 1 - Standard

Standard
The system runs automatically, and uses excess energy harvested during the day to fill the gaps when there is not enough PV power available. Typically in the evening and night. Easy configuration in Assistants and on the Color Control GX.

Necessary Victron equipment:

1. Multi or Quattro inverter/charger
2. Color Control GX
3. AC sensor used for control loop

For details refer to the main hub-4 manual.

1.2 Mode 2 - Advanced

Same as standard operating mode, but more flexibility is given to implement time shifting, load management or other energy management optimization algorithms. Either by ModbusTCP commands or by running additional self implemented code on the Color Control GX. Often the best of both worlds: complete flexibility and benefit from the VRM Portal and all other functionality already available on the CCGX, without having to add additional cost of extra PLCs or other control modules.

Necessary Victron equipment:

1. Same as mode 1, Standard.

Available control points:

- Grid power setpoint
- Maximum charge percentage
- Maximum discharge percentage

1.3 Mode 3 - Custom

Customer self implements their control loop and grid measurements, and uses the MultiPlus and/or Quattros as simple, remote controllable, bidirectional inverter/chargers that can be set to either charge or discharge an x amount of Watts. Note that the point of control is the AC-input, not the charger itself.

```
Power to/from AC-input = Power to/from battery + Power to/from AC-output
```

Necessary Victron equipment:

1. Multi or Quattro inverter/charger
2. Color Control GX
3. Note that there is no AC sensor necessary, since the inverter/charger will act as a 'dumb' bi-directional inverter/charger. It will act on the external command given, which can for example be 'take 2000W from AC in, or feed back 100W through AC in'.

Available control points:

- Switch (on, charger-only, inverter-only, off)
- Power setpoint in Watts: regulates the power on the ac-input.
- Disable charge flag
- Disable feedback flag

2. Details of operating mode 2 - Advanced

2.1 Operation

- Grid power setpoint. A positive value: power is taken from grid. Negative value: power fed back to grid. Note that the maximum charge and discharge percentages documented below have priority over the grid power setpoint.
    - D-Bus: com.victronenergy.hub4 /AcPowerSetpoint
    - Modbus: Register 2700, unit ID 0
- Maximum charge percentage. Limits the power taken in by the Multi through AC-In.
    - 0%: disable charging. This setting may be used for time shifting. For example by disabling charging the battery when feeding back to the grid is profitable, and leaving battery capacity for later.
    - 100%: no charge power not limited by CCGX (max charge current on multi still applies).
    - >0% and <100%: The maximum allowed AC-In power is
      max charge percentage * max charge current * battery voltage / 100
      The maximum charge current is taken from the settings on the Multi (which can be changed using VE configure).
      Because this setting limits the power taken in on the Ac-In of the multi, it may give unexpected results when the AC-Out of the multi is also in use (for example to connect critical loads).
    - D-Bus: com.victronenergy.hub4 /MaxChargePercentage
    - Modbus: Register 2701, unit ID 0

- Maximum discharge percentage. Can be used to enabled or disable battery discharge. Value < 50 means that discharge is disabled. Otherwise discharge current is limited only by the battery and/or the Multi.
    - D-Bus: com.victronenergy.hub4 /MaxDischargePercentage
    - Modbus: Register 2702, unit ID 0

2.1 Notes

- The above settings are intended for power management, and will be overridden by battery safety mechanisms. For example: if the multi is in sustain mode - because the battery is almost empty - it will start charging the battery, event if the grid power setpoint is negative or the maximum charge percentage is set to zero.
- Sustain will always remain in operation. Disabling charge will not disable sustain. Note that this was added to the Hub-4 Assistant released in january 2016 (version 154).
- In mode 2 the CCGX will change the Multi's AC-In power setpoint (see Mode 3) continuously, so it does not make sense to change the setpoint in this mode.
- See modbustcp excel sheet for scaling and data types, available here.

3. Details of operating mode 3 - Custom

3.1 Operation

- When the grid is available, the Multi will be connected to the grid. You can control its operation with the Power setpoint. When positive, energy will flow to the grid. When negative, energy will be taken from the grid. Some examples:
    - When set to +400W, it will feed 400W back through its input. This energy will be taken from the battery. If there is also a 200W AC load connected to AC output the total energy taken from the batteryt will be 600W. The batteries will always be discharged when the setpoint is positive.
    - When set to -400W, it will take 400W from the AC input. When the load on the output is lower than 400W, it will charge the battery with the difference. When the load on the AC output is higher, it will discharge the battery with the difference. So with a negative setpoint charge/discharge depends also on the connected loads.
    - **Important note**: When using the CCGX to communicate with the HUB-4 assistant (see further on) you should bear in mind that the CCGX inverts the setpoint. So positive becomes negative and vice versa.
- Note that it will always remain within battery and maximum inverter power limits: when the battery is full, or when the maximum charge current as configured in VEConfigure is already met, it will not draw more power. And when instructed to discharge with 10000 watt, while it is a 2500W inverter, it will discharge with 2500W, until the battery is empty.
- To force the Multi to Inverter Mode, set the switch to Inverter-only. Note that when you do that, there will be no grid assist, meaning that on an overload the Inverter will switch off, signalling overload alarm. Instead of switching back to grid.
- When no grid is available, it will automatically switch to inverter mode (seamless), unless the switch is set to charger-only. While in inverter mode, the Power setpoint is in-effective.

3.2 Notes

If no new value is sent to the Multi for longer than 500 periods (= 10 seconds at 50 Hz), the Multi will automatically fall back to 'bypass' that is: No charge/No discharge. The internal transfer relay is closed when AC input is available. In that case any loads on the AC output are fed by the AC input. If there is no AC input available, the unit will switch to inverter mode. (Unless it is set to 'Charger Only' or receives a 'low cell' signal from the VE.Bus BMS).

- Send a target value to the CCGX via ModbusTCP:
    - unit-id 246 (= CCGX VE.Bus port)
    - register-id 37 (= /Hub4/AcPowerSetpoint)
      Important: The CCGX <u>inverts</u> the value of the setpoint. So use a positive value to instruct charging and a negative one to tell it to discharge. The unit is Watts.
    - register-id 38 (= /Hub4/DisableCharge)
      Value 0: charge enabled, Value 1: charge disabled.
      Important: Disabling charge will also disable feedback.
    - register-id 39 (= /Hub4/DisableFeedback)
      Value 0: feedback enabled, Value 1: feedback disabled.
      If feedback is disabled, the battery will not discharge. In case of grid failure, this parameter will be ignored and power will be supplied to AC Out.
- Note that it can very well be that the Multi does not do what you want it to do, because the battery is either empty, or full for example. Therefore, for your control algorithm, read the actual power back from the Multi, unit-id 246 and register-id 12.
- See modbustcp excel sheet for scaling and data types, available [here](#).
- Note that the mechanism is slow, which to our opinion is not perfect, but also no problem.
- The 'VE.Bus to VE.Can interface' does not yet support reading and writing the hub4/grid-parallel setpoint.
- If your system contains a Hub-4 compatible AC-Sensor which is set up as grid meter, the CCGX will automatically enter mode 1 and start updating the AC power setpoint continuously. You can stop this by setting the value of /Settings/CGwacs/Devices/D<serial>/Hub4Mode in the settings D-Bus service (com.victronenergy.settings) to 3 (=hub4 control disabled). Replace <serial> with the serial number of the grid meter.

3.3 Using the MK2 directly instead of CCGX

In this setup, it is not necessary to use a CCGX. Instead an MK2 is used. We have both MK2-RS232 and an MK2-USB available, see the pricelist. A direct connection to the VE.Bus RS485, without MK2 or CCGX, is not possible.

Note that, as also indicated on the Datacommunication whitepaper, the MK2 protocol is not an easy protocol. That is unfortunate, but it is what it is. And we cannot give support on it unless there is a huge commercial value behind the project, read thousands of Multis or Quattros.

And now, after all the warnings, the information:

Use RAM ID 128 and higher. Every RAM ID is a word (2 bytes)

The assistant RAM will be filled from ID 128 with 'assistant RAM records'. Each record starts with a word that contains the AssistantID and the size of the record, and then a number of AssistantRAM

words.

RAM ID	contents
128	ID_Size (1st Assistant)
129	1st AssistantRAM0
130	1st AssistantRAM1
…	…
RAMn	1st AssistantRAMn
RAMn+1	ID_Size (2nd Assistant in this VE.Bus device, if any)
RAMn+2	2nd AssistantRAM0
RAMn+3	2nd AssistantRAM1
etcetera	

The ID_Size word contains 0xZZZY, where:

- ZZZ = the Assistant ID. This is where you can recognize the function of the assistant (HUB4 ID = 3)
- Y = the number of ramIDs that will follow after the ID_Size. At the moment of writing this is 2 for the HUB-4 Assistant version. This can be increased in future releases if more parameters become available.

So you have to scan the Assistant RAM records by looking at each ID_size record.

Search for HUB4, and then RAM0 is the setpoint in Watts and RAM1 contains the flags:

- bit0 of RAM1 contains the flag 'disable charge'
- bit1 of RAM1 contains the flag 'disable feedback'

Above information is an addendum to the 'Interfacing with VE.Bus products' document. Available from our whitepapers.

EXAMPLE:
Below you will find a communication example (assumed is that the HUB-4 assistant is the 1st assistant in the assistant list using AssistantRAM. So this means that the ID_Size is at RamID 128.)

Reading the ID_Size:

→ 0x05, 0xFF, 0x57, 0x30, 0x80, 0x00*, 0xF5
(Length, 0xFF, 'W', 0x30, Lo(ID), Hi(ID), Checksum)

Response:
← 0x07, 0xFF, 0x57, 0x85, 0x32, 0x00, 0x52, 0x5A, 0x40
(Length, 0xFF, 'W', 0x85, Lo(ValueA), Hi(ValueA), Lo(ValueB)*, Hi(ValueB)*, Checksum)

ValueA is the contents of RAMID 128. In this example it is 0x0032 which indicates HUB4 with 2 extra RAMIDs.

*) Please note that you will get an extra ValueB. This is a feature of newer Multi firmware versions. Because the IDs range from 0..255 the Hi(ID) field would always be 0. Newer Multi firmwares allow you to specify a second ID in this field. So in this case ValueB is the value of RAMID 0 because the 0x00 is interpreted as the second ID. RAMID 0 corresponds with UMains (This can be found in paragraph 7.3.11 of the 'Interfacing with VE.Bus products' document.) So in this example the UMains

value is 0x5A52 ⇒ 231.22V NOTE: You will always get a ValueB in the response. You can make handy use of this by reading an extra RAMID or you can ignore it if you don't need it.

Writing the set point with +200 Watt:
We should write RAM ID 129 to +200.
Writing requires 2 frames, one frame specifying the ID to write and one frame specifying the data to write.

→ 0x05, 0xFF, 0x57, 0x32, 0x81, 0x00, 0xF2
(0x32=Command Write RAMID, 81 00 = 129, Note that 2nd byte will always be 0 in this command)

→ 0x05, 0xFF, 0x57, 0x34, 0xC8, 0x00, 0xA9
(0x34=Command Write Data, C8 00 = 200)
Note: If a negative set point is needed you should specify this as the 2's complement value so for example -200W must be specified as 0xFF38 (=65536-200) so that would result in: → 0x05, 0xFF, 0x57, 0x34, 0x38, 0xFF, 0x3A

Response:
← 0x05, 0xFF, 0x57, 0x87, 0x00, 0x00, 0x1E Write OK (87 means successful. Just ignore the rest of the frame (00,00 in this case))
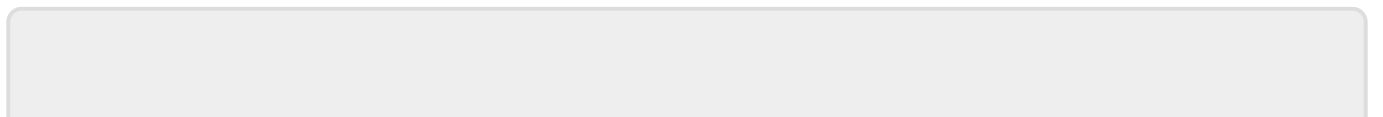
Please refer to the 'Interfacing with VE.Bus products' document, available from our whitepapers, for more info.

4. Combining Hub-4 with Lithium batteries

- Victron Lithium batteries with a VE.Bus BMS: select this in the Hub4 assistant and all will be done automatically.
- Non-Victron batteries:
    - when you do not want to discharge, set the switch to charger-only. Note that when the switch is set to charger-only, and there is no grid available, there will still be a small power draw on the battery to power the control board.
    - when you do not want to charge, disable charging via the disable-charge flag.
- As an alternative to running the control loop externally, using ModbusTCP, it is also possible to run code on the CCGX itself and update the AcPowerSetpoint via D-Bus. We have one customer that is running a MQTT client on the CCGX, written in Python, that gets the control-loop output as updates from a MQTT broker. And the Python script sends them to the Multi, using D-Bus service com.victronenergy.vebus.ttyO1, and path /Hub4/AcPowerSetpoint

DISQUS

~~DISQUS~~