

NOTE: The following is an **evolving** description of the NanoVNA's USB **Console Command** list and is updated as new CONSOLE commands are added/removed or functionality of existing commands changes. Refer to the **NanoVNA User Guide** for on-screen GUI commands.

For additional info on a command, ask about it in the forum and/or read the firmware's release notes.

New Commands or F/W News:

Hugen has removed the DUMP, SCANRAW, COLOR commands from v0.4.0 Nov 15/19 release & up to free up memory

Console Commands (in no particular order):

help – **Note:** Commands in () may or may not be present for an individual developer's F/W release.

Prints the following command list, which is subject to F/W version, to the console:

```
help, exit, (info), threads, version, reset, freq, offset, time, dac,
saveconfig, clearconfig, data, dump, frequencies, port, stat, gain, power,
sample, (scan), (scanraw), sweep, test, touchcal, touchtest, pause, resume
, cal, save, recall, trace, marker, edelay, (capture), (vbat), transform,
threshold, (color), (vbat_offset)
```

exit – exits and then restarts console mode

outputs: "ChibiOS/RT Shell"

info – prints firmware info on console. Can be used to verify F/W build date.

- This is similar the *Version* selection in the on-screen GUI menu

```
Kernel:      4.0.0
Compiler:    GCC 7.2.1 20170904 (release) [ARM/embedded-7-branch revision 255204]
Architecture: ARMv6-M
Core Variant: Cortex-M0
Port Info:   Preemption through NMI
Platform:    STM32F072xB Entry Level Medium Density devices
Board:       NanoVNA
Build time:  Sep 19 2019 - 08:34:02
```

version – returns simple version name and/or release – Not the same as the GUI version command. See *info* command.

eg: "edy555 0.2.2 extended with scan command" or "0.3.0m"

threads – prints running threads (mainly for F/W debug use):

stklimit	stack	addr	refs	prio	state	name
20000200	2000039C	20001BF8	0	128	WTEXTIT	main
20001AF8	20001B3C	20001B90	0	1	READY	idle
20002910	20002BA4	20002C98	0	128	READY	sweep
20002CD0	20002D4C	20002F18	0	129	CURRENT	noname

reset – Performs POR (power-on reset) of the device

This is **NOT** the Calibration Reset command.

Outputs to the console: "Performing reset"

reset dfu – places device into DFU mode (only found in F/W after Nov 1/19)

freq – Used to set the CW frequency

Usage: **freq {frequency (Hz)}**

offset – Used to offset frequency of second osc in the SI5351 (experimental & debug use)

Usage: **offset {frequency offset (Hz)}**

Default is 5KHz (5000)

transform – allows controlling the transform function from the console

-Usage: **transform {on|off|impulse|step|bandpass|minimum|normal|maximum} [...]**

threshold - set the threshold for switching to harmonics mode. The default is 300MHz.

- After changing, you can use the save configuration command to permanently store the new value.

- This greatly simplifies testing if your SI5351 has difficulty reaching 300MHz and if so, provides an easy workaround.

usage: **threshold {frequency in harmonic mode}**

output: current: 300000000 <<Defaults to 300MHz with no input

See <https://groups.io/g/nanovna-users/message/4627>

Use saveconfig console command to save value

scanraw - The 'scanraw' command is found in [QRP73's](https://github.com/qrp73/NanoVNA-Q/releases) releases of firmware v0.4.1 and above

created by QRP-RX for the NanoVNA and is available only through the command-line console interface.

F/W release location: <https://github.com/qrp73/NanoVNA-Q/releases>

From the NanoVNA's console interface:

```
ch> scanraw
```

```
usage: scanraw {channel (0|1)} {start (Hz)} {step (Hz)} {step-count} [average]
```

The 'scanraw' command is similar to the 'scan' command, which allows you to perform measurements for an unlimited number of points. However, you can also specify which port (CH0 or CH1) along with a user-specified number of points to average - up to 1000x.

NOTE: As with the 'scan' command (see deprecated commands section), the 'scanraw' command returns raw measured values, with no calibration applied so you will need to apply your own calibration corrections to the results.

Example: The following command will return S11 measurements (with no calibration applied) from 50 kHz to 10.050 MHz with 1 kHz step (10000 points) and with 5x averaging:

```
scanraw 0 50000 1000 10000 5
```

where:

0 - channel CH0 (S11)

50000 - starting frequency (Hz)

1000 - frequency per step (Hz) 100Hz minimum**

10000 - total number of steps/points

5 - number of times to average

** QRP firmware release 0.4.2 only, has 1Hz steps: <https://github.com/qrp73/NanoVNA-Q/releases>

time – get the time in:

timespec.year+1980, timespec.month, timespec.day, timespec.millisecond

outputs: "1980/1/1 8203499"

Cannot be set - it is the Linux "time" command - Useful only if you have a RTC in the system.

sample - Samples gamma, amplifier or reference signal states – (used for NanoVNA debugging)

Usage: **sample {gamma|ampl|ref}**

gamma - amplitude / reference signal << **this is what you want, do not use the others** (per Erik)

ampl - measure the absolute strength of the bridge/CH1 signal

ref - measure the absolute strength of the reference signal

capture – dumps the screen content so you can make a "screenshot"

- capture returns 153600 bytes, which represent the 320x240 pixel image with 2 bytes per pixel. The pixel color is coded with RGB565 format. For some reason edy555 decided to swap LSB and MSB bytes of the pixel word on firmware side. So you need to reverse bytes in pixel word.
- Running the capture command will freeze and dump the ILI9343's internal display RAM to the console output. Not all ILI9343 compatible (ie: clone) displays can be read and some may only return FF's.
- Ensure you set up a read array large enough for the 153600 bytes. (per QRP)
- See this thread for more technical info: <https://groups.io/g/nanovna-users/message/5941>

dump - returns buffer with audio data received from audio codec. This is ADC output of SA602 mixers.

Format:

```
FB8D FF8D F84C FF8A F824 FF85 FA9B FF80 FEE5 FF7A 03C6 FF74 075A FF74 0783 FF79
03FF FF81 FE9C FF88 F9C5 FF89 F7C2 FF8C F8C7 FF85 FC41 FF81 00ED FF7C 058C FF75
07F6 FF74 0672 FF7C 01B5 FF86 FC35 FF8E F8D2 FF8B F803 FF88 FA0C FF84 FE31 FF7C
02F2 FF76 06DE FF75 07D1 FF78 04DB FF7E FF8A FF87 FA7C FF8A F7FF FF8B F868 FF83
FB74 FF7F 0028 FF79 04E6 FF73 07B7 FF75 06FC FF7C 02CA FF85 FD45 FF8A F90C FF8D
F7C2 FF8E F9A3 FF84 FD5A FF7B 021A FF78 0677 FF74 07EE FF75 056B FF7F 0066 FF85
```

dac – Used to output or set the DAC level

It seems to set the output voltage of an analog output of the STM uP

The adc gain is automatically set depending on the selected frequency

Set usage: **dac {value(0-4095)}**

With no parameter: ch> dac

usage: **dac {value(0-4095)}**

current value: 1923

saveconfig – Used to save the touchscreen calibration, vbat_offset calibration (in 'Q' F/W) and threshold values

Outputs: "Config saved"

clearconfig - usage: **clearconfig {password}**

Where password is hardcoded as "1234"

A wrong input gives: "Key unmatched" error. Correct key gives: "Config and all cal data cleared"

Used to clear ALL flash memory with configuration, **BUT NOT RAM.**

Restart the Nanovna and calibrate the touchscreen and RF sections after using this command !!

NOTE: The old config remains in the RAM and if you make save it will be stored back into flash. When you power off NanoVNA after clearconfig, and then power on, the firmware will see at boot time that there is no config in the flash and default config will be loaded. **Saving without a power restart will restore bad config data.**

port – Sets port of DSP tlv320aic3204

Usage: **port {0:TX 1:RX}**

frequencies – prints a long list of all standard F/W 101 sweep points from the *start* frequency to the *stop* frequency.

```
50000
9049500
18049000
45047500
...up to...
846003000
855002500
864002000
891000500
900000000
```

example: This is a list of 101 frequency points

from 50KHz to 900MHz

stat – outputs processor status (used for NanoVNA debugging)

average: -138 -137

rms: 3597 8

callback count: 9126700

awd: 27

gain – sets left and right channel gain of DSP tlv320aic3204

Usage: **gain {lgain(0-95)} [rgain(0-95)]**

power - Used to set the output level of the Si5351A clock generator

- Usage: **power {0-3}** **Read F/W release notes for default value!!

- Output level is usually increased when in harmonic mode. Refer to NanoVNA forum posts on this topic.

data - usage: data [array]

example: **data [100 200]**

produces output of 101 points:

```
0.999902069 -0.003914059
0.999517679 -0.003805434
0.999603867 -0.003855689
0.999514162 -0.004347831
...up to...
0.998815953 -0.006629021
0.997710704 -0.006602317
0.998088240 -0.006423361
0.998069703 -0.006385779
0.998046994 -0.006417039
```

NOTE: Left & right columns above - the Real part and the Imaginary part, respectively, of the reflection coefficient.

Reading Calibration Data:

The firmware version **edy555-0.2.2** allows dumping the internal calibration table with the data command:

data [0-6] Where: 0: S11

1: S21

and

2: /* error term directivity */

3: /* error term source match */

4: /* error term reflection tracking */

5: /* error term transmission tracking */

6: /* error term isolation */

BUT - it is currently NOT possible to upload calibration tables into the NanoVNA (at this time)

sweep – sets the start/stop for the next automatic sweep, but you never know when that will happen or when that will be finished and issuing a pause command stops the automatic sweep (per QRP)

- Usage: **sweep {start(Hz)} [stop] [points]**

- If no inputs, then prints current setup eg: 300000000 500000000 101

touchcal – produces touchscreen calibration touch points in the upper left and lower right corners

outputs: "first touch upper left, then lower right..."

"done"

"touch cal params: A B C D (upper left and lower right x-y co-ords)

Use the **saveconfig** console command to save.

touchtest – test touch accuracy

– touch, hold and drag pointer to draw a line on the display to verify tracking accuracy

pause – pause LCD display refresh

NOTE: Update to the latest F/W after Oct 24, 2019 to prevent the GUI pause function from stopping all USB communication if not already connected to a PC.

resume – resumes LCD display output refresh – this is the opposite of the console's **pause** command.

cal – Allows an external application to perform a system port calibration of the device

Usage: **cal** [**load**|**open**|**short**|**thru**|**isoln**|**done**|**reset**|**on**|**off**|**in**]

The first command to use here is: **cal reset**, followed by a complete OSLT calibration sequence.

After calibration, use the **save** command to save it to one of 5 memories: 0 to 4

save - **save** {**id**} where 'id' is the calibration & system state memory location 0 to 4

NOTE: 1. Memory **0** is always recalled at power-on.

2. The console save is **NOT** the same as the GUI save button in the on-screen Config menu.

trace – prints status of either 2 or 4 traces depending on F/W version used.

- first number is a scale of trace, see menu DISPLAY => SCALE/DIV

- second number is a reference position of trace, see menu DISPLAY => REFERENCE POSITION

```
0 SWR          CH0 1.000000000 0.000000000
1 LOGMAG       CH1 1.000000000 7.000000000
2 SMITH        CH0 1.000000000 0.000000000
3 PHASE        CH1 1.000000000 4.000000000
```

marker – With no parameters supplied, outputs info on all visible markers:

```
ch> marker
```

```
1 44 396028000 \ This assumes all 4 markers are enabled.
2 35 315032500 \ Format:
3 20 180040000 / marker-number index marker-frequency
4 92 828004000 /
```

Usage: **marker** [**n**] [**off**|{**index**}]

Where: **Index** is the number of the point [0-100] (Any one of 101 points along the trace)

color – allows user to read or change the colour of each of the four individual traces**

Returns the following with no parameters:

```
ch> color
```

```
usage: color {id} {rgb24}
```

```
-3: 0x77EFBF
```

```
-2: 0xFFFFFFFF
```

```
-1: 0x878387
```

```
0: 0xFFE31F
```

```
1: 0x00BFE7
```

```
2: 0x1FE300
```

```
3: 0xE7079F
```

NOTE: Colour parameters need to be entered with '0x' hex prefix:

Example: **color 3 0xffffffff** (This turns trace 4 from purple to white)

Refer to this post: <https://groups.io/g/nanovna-users/message/5914>

Examples: if you want to change track colors, just use the following commands:

```
color 0 0xFF0000
```

it will change TRACE 0 color to red (web color #FF0000)

```
color 1 0x00FF00
```

will change TRACE 1 color to green (web color #00FF00)

```
color 2 0x0000FF
```

will change TRACE 2 color to blue (web color #0000FF)

```
color 3 0x808080
```

will change TRACE 3 color to gray (web color #808080)

```
color -1 0xFFFFFFFF
```

will change grid color to white (web color #FFFFFF)

You can play with it and at any time, you can revert colors back just with power off and power on.

If you want to store selected colors permanently, you need to use menu CONFIG => SAVE

** Only with QRP's v0.4.2 and higher firmware release

vbat – returns the approximate voltage of the device’s internal battery in mV.

Latest firmware with SMT 1N4148 diode added to PCB at location D2. (F/W assumes either 0.2Vf or 0.6Vf drop across diode. Refer to firmware release notes to determine diode type – Si or Schottky)

Returns battery voltage in this format: **4213mV**

vbat_offset – allows the battery voltage shown on the GUI config screen to be ‘calibrated’ by specifying a millivolt offset value (typically the Vf of the diode at D2).

Usage: `vbat_offset [mV]` Returns currently saved value if [mV] is blank.

Use **saveconfig** command to save offset value.

Example:

1. measure raw battery voltage with external voltmeter
2. connect Nanovna to terminal and run **vbat_offset 0** (0mV offset applied)
3. read voltage shown on nanovna version screen and subtract it from raw battery voltage [= **result**]
4. run **vbat_offset [result]** and then save it using **saveconfig** command
5. **VBAT** on the version screen will now display the accurate battery voltage

recall - recall {id} where ‘id’ is calibration & system state memory location 0 to 4

edelay – electrical delay in **picoseconds**

Usage: **edelay [n]** where **n** is in picoseconds

Output with no parameters supplied shows current setting.

Eg: 20.00000000

Commands to be depreciated by individual developers

scan – used to set the scanning boundaries and number of points (per Erik, Oct 22 2019)

To be replaced by QRP’s scanraw command (see above)

format: `scan start(Hz) step(Hz) points`

Erik K. added a command to the eddy firmware to enable on demand scans of arbitrary length (you can scan with one million steps or much more if you want)

How to use:

First pause the continuous scanning with "pause" and then use the "scan" command to:

`scan [from frequency in Hz] [increment frequency in Hz] [number of steps]`

The frequency increment step is for now an integer

The scan command outputs:

```
start
frequency s11_real s11_imag s21_real s21_imag \
...
frequency s11_real s11_imag s21_real s21_imag /
done
```

NOTE: The scan command returns raw measured values, with no calibration applied so you will need to apply your own calibration corrections to the results.

Example:

```
ch> pause
ch> scan 5000000 20 5
ch> start
5000000 0.001503840 0.000420701 -0.306770563 0.018568072
5000020 0.000695601 0.000503197 -0.306792527 0.018579231
5000040 0.000532656 0.000520238 -0.306793421 0.018573865
5000060 0.000495833 0.000512704 -0.306819111 0.018593480
5000080 0.000520689 0.000523833 -0.306812644 0.018576323
done
```

Hardware check

QRP's firmware has a built-in si5351 hardware check starting with his 0.3.0m release and higher. So, if you see the **red PLL** indicator under the battery icon, it means that you might have max-freq-limited si5351 chip which may become unstable above its 200 MHz fundamental maximum operating frequency.

Possible fix – run the **clearconfig 1234** console command, reboot and re-calibrate RF to see if that resolves the error indicator. Refer to clearconfig console command info, above.

This hardware check has been incorporated in the Hugen's firmware github tree as well as of mid Nov 2019 and any developer that uses his source code.

NanoVNA Firmware Developers on GihHub:

Edy555	- https://github.com/ttrftech/NanoVNA	Original developer of the NanoVNA
Hugen	- https://github.com/hugen79/NanoVNA-H	Made HW, SW & UI improvements to edy555's design
Erik	- https://github.com/erikkaashoek/NanoVNA	SW improvements
QRP	- https://github.com/qrp73/NanoVNA-Q	SW & UI improvements
Reald	- https://github.com/reald/NanoVNA	UI improvements
Oristo	- https://oristopo.github.io/nVhelp/html/firmware.htm	An excellent firmware reference webpage

Removed or Disabled Commands:

-Echo -systime
-test -mem

Hugen has removed the DUMP, SCANRAW, COLOR commands from his v0.4.0 Nov 15/19 release & up to free up memory.

Changes:

LR - July 2 2019 - Original

LR - July 5 2019 - Added better command output descriptions

LR – Aug 6 2019 – Cleaned up formatting

LR – Sept 5 2019 – added **scan** console command to list

LR – Sept 25 2019 – updated new command list, improved command descriptions

LR – Oct 4 2019 – added Erik's corrections on console command usage

LR – Oct 4 2019a – added Erik's info on reading calibration data

LR – Oct 7 2019 – Added two new console commands and updated others

LR – Oct 15 2019 – Added 'reset dfu' command and example of scan command

LR – Oct 16 2019 – Added QRP's 'scanraw' command

LR – Oct 24 2019 – Depreciated scan command, improved command descriptions, new section

LR – Oct 30 2019 – Updated scan & capture, added color command and document reformat

LR – Nov 4 2019 – Added new vbat_offset command for QRP's firmware release 0.4.3, added list of NanoVNA GitHub developer firmware repositories, added more command details

LR – Dec 3 2019 - Misc updates and corrections to clearconfig command

LR – Dec 9 2019 – Updated command list, improved descriptions.

COPYRIGHT 2019 – L. Rothman

For more detailed explanations of commands, please ask the developers via the NanoVNA forum.

This list may be copied freely and updated so long as my original copyright is left in place.

Please advise me of any errors or updates through a message on the NanoVNA-Users group at:

<https://groups.io/g/nanovna-users/messages>