# NanoVNA User Guide

Updated and edited with additions by L. Rothman
Updated: Jan 15, 2020
Original Japanese document by: Satoh, Hiroh (cho45) https://github.com/cho45/NanoVNA-manual
Translated from Japanese to English by Google Translate

All Table of Contents links resolve to cho45's translated webpages, not this document.

# 1. Introduction

This document is just one of several user guides available for the original (Version 1) NanoVNA

The ***Original*** URL of this manual is written in **Japanese** and is at https://cho45.github.io/NanoVNA-manual/. It was last updated in Oct 2019.

**NOTE: The user of this guide should be familiar with using Smith Charts, antenna analysers, time domain measurements and other RF measurement techniques.**

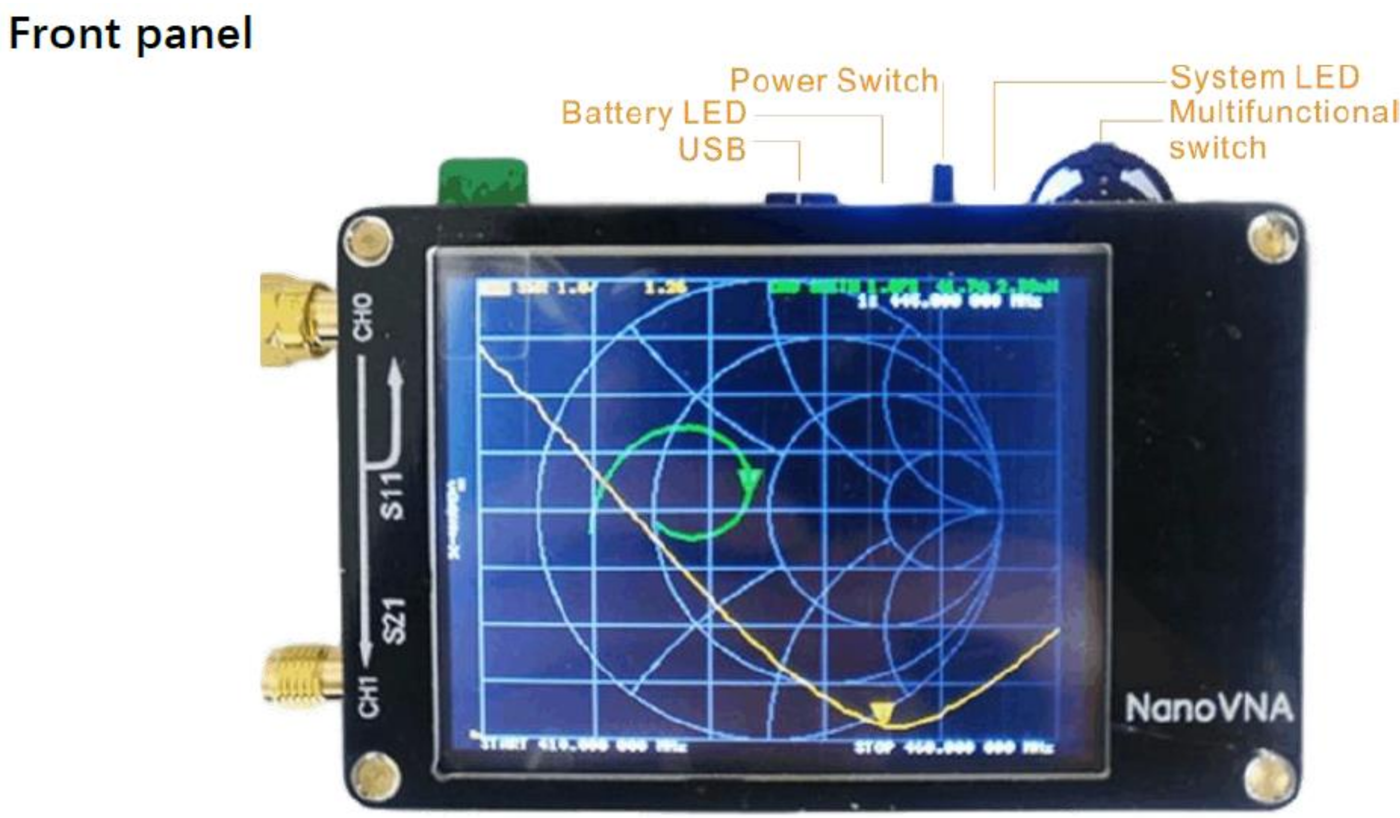
**Learn about the NanoVNA and general VNA Theory in the following online forum areas:**
**Documentation & Update Files: https://groups.io/g/nanovna-users/files**
**Knowledge-based Wiki: https://groups.io/g/nanovna-users/wiki**

## What is the NanoVNA

There are several types of NanoVNA hardware, and this document covers the following hardware:

- ttrftech version (original) ttrftech / NanoVNA
- hugen79 version hugen79 / NanoVNA-H

These hardware versions are almost the same on the circuit, and common firmware can be used.



# What you need to work

The following are required at a minimum.

- A NanoVNA device
- SMA LOAD 50Ω
- SMA SHORT
- SMA OPEN
- SMA Female to Female Through Connector (ensure it is of good quality)
- SMA Male to Male cable x 2

# NanoVNA basics

A Vector Network Analyzer (VNA) measures the frequency characteristics of reflected power and passing power of a high frequency network (RF Network).

The NanoVNA measures the following I (Level) and Q (Phase) elements:

- Input voltage I / Q signal
- Reflected voltage I / Q signal
- Pass voltage I / Q signal

From here we calculate:

- Reflection coefficients S11
- Transmission coefficient S21

Some of the following items that can be calculated from these can be displayed:

- Reflection loss
- Pass-through loss
- Complex impedance
    - resistance
    - reactance
- SWR

Newer firmware versions have already added additional measurement functions such as TDR, on-screen reactance measurement readouts and large-font antenna analyser modes.

## NanoVNA frequency range

The NanoVNA measures the reflection coefficient and transmission coefficient for 101 points in the frequency band to be measured.

The fundamental frequency range of the NanoVNA is 50kHz (10KHz on newer F/W) to 300MHz. For higher frequencies, harmonic mode is used. The fundamental wave is ***not*** attenuated even in harmonic mode**. The usage modes for each frequency are as follows.

- Up to 300 MHz: fundamental wave
- 300MHz to 900MHz: 3rd harmonic
- 900MHz to 1500MHz: 5th harmonic

****Note** that there is always a fundamental wave present at the output (CH0), especially when checking amplifier gain so be wary of saturation issues. In all cases, the RF input is converted to an intermediate frequency of 5kHz. The signal is converted from analog to digital at 48kHz sampling. Digital data is signal processed by the MCU.

| IF frquency (kHz): 5 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| MHz | Measuring range | | Output Fundamental at | | Mixer Fundamental at | | Harmonic used | |
| Band | Start | Stop | Start | Stop | Start | Stop | Test signal | Mixer input |
| 0 | 0.05 | 299.99 | 0.05 | 299.99 | 0.06 | 300.00 | 1 | 1 |
| 1 | 300.00 | 900.00 | 100.00 | 300.00 | 60.00 | 180.00 | 3 | 5 |
| 2 | 900.00 | 1500.00 | 180.00 | 300.00 | 128.57 | 214.29 | 5 | 7 |
| 3 | 1500.00 | 2100.00 | 214.29 | 300.00 | 166.67 | 233.33 | 7 | 9 |

# 2. To do first

Before you can use the NanoVNA, you must first calibrate it.

Always calibrate for the specific frequency range you will be testing. Since each calibration sweep consists of only 101 frequency points, everything between each calibrated point will be interpolated. Because of this, it is always better to use as small a frequency test span as possible so the unit's calibration points are as close as possible to each other.

Calibrate according to the calibration method in section 6 of this guide.

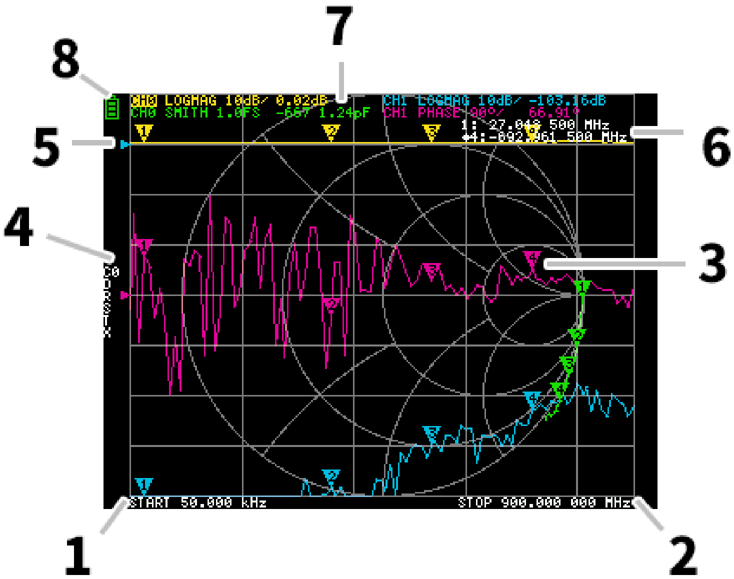
# 3. Input methods

The NanoVNA has the following input controls:

- Touch panel long tap using your finger or a stylus
- Lever (jog) switch
  - Left or Right, Short or Long press
  - Push short press or Push long press
- Power – On/Off slide switch

# 4. How to read the screen

## Main screen




### 1. START frequency

### 2. STOP frequency

The START frequency and STOP frequency are shown along the bottom of the display.

### 3. Marker

The marker position for each trace is displayed as a small numbered triangle. The selected marker can be moved to any of the 101 measured points in the following ways:

- Drag a marker on the touch panel – best to use a stylus for this.
- Press and hold Left or Right on the lever/jog switch.

### 4. Calibration status**

Displays the data number of the calibration being read and the error correction applied. (** See Developer Extended Features section of this document)

- C0 C1 C2 C3 C4 : Each indicates that the corresponding calibration data is loaded.
- c0 c1 c2 c3 c4 : Each indicates that the corresponding number of calibration data is loaded, <u>but the frequency range has been changed after loading</u>, indicating that the error correction is being interpolated.
- D : Indicates that directivity error correction is applied
- R : reflection tracking Indicates that error correction is applied
- S : source match Indicates that error correction is applied
- T : transmission tracking Indicates that error correction is applied
- X : indicates that isolation (crosstalk) error correction is applied

## 5. Reference position

Indicates the reference position of the corresponding trace. You can change the position with:

**`DISPLAY →SCALE →REFERENCE POSITION .`**

## 6. Marker status

The active marker that is selected and one marker that was previously active are displayed top right.

## 7. Trace status

The status of each trace format and the value corresponding to the active marker are displayed.

For example, if the display is showing: **`CH0 LOGMAG 10dB/ 0.02dB`** , read it as follows:

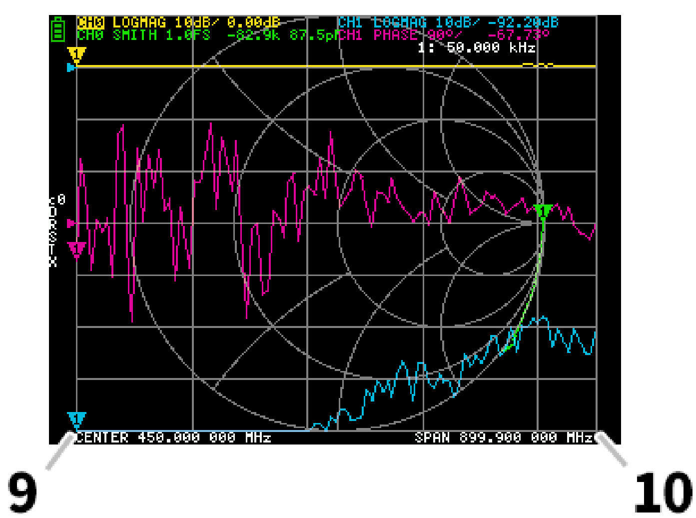
**Channel** CH0 (reflection)
**Format LOGMAG**
**Scale** is 10dB
**Current value** is 0.02dB

For active traces, the channel display is reversed.

## 8. Battery status

If a battery is installed and a **1N4148** surface mount diode is mounted on the PCB at location **D2**, an icon is displayed according to the battery voltage. If the diode is missing, the icon will show an empty battery.

Note: The latest QRP 0.4.3 firmware release allows calibrating the battery level via the console commands. Refer to the current Console Command document at:
https://groups.io/g/nanovna-users/files/

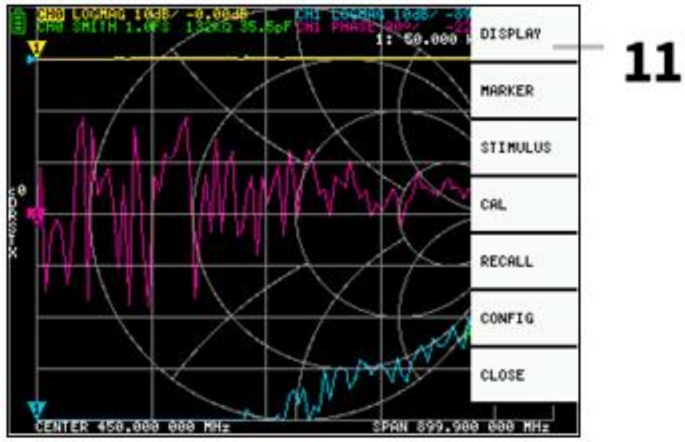



## 9. CENTER frequency

## 10. Span

Each frequency is displayed as above when the CENTER frequency and SPAN are specified

# Menu screen



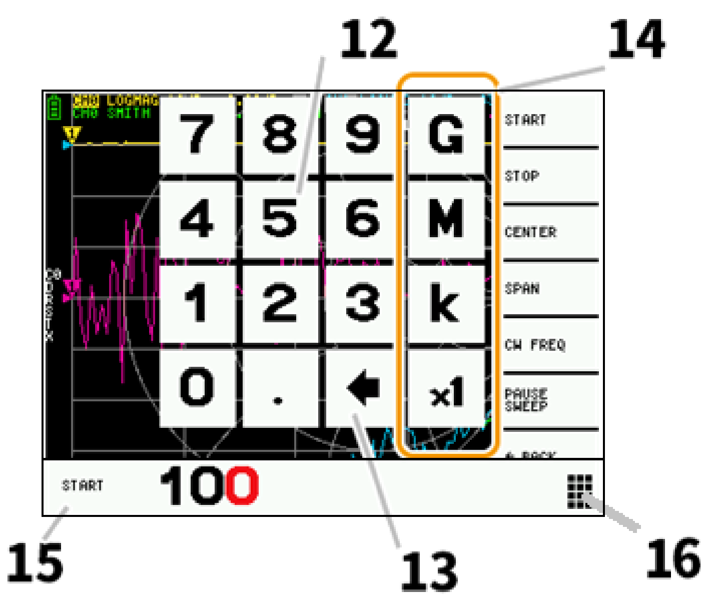

## 11. Menu List

The menu can be displayed by the following operations.

- When a location other than a marker on the touch panel is tapped
- Push the lever switch straight inwards and release.

**An interactive list of all menu commands can be found here:**
https://oristopo.github.io/nVhelp/html/Menu.htm
And listed in Section 11 at the end of this guide.

# Keypad screen




## 12. Numeric keys

Tap a number to enter one character.

## 13. Back key

Delete one character. If no character is entered, the entry is canceled and the previous state is restored.

## 14. Unit key

Multiplies the current input by the appropriate unit and terminates input immediately. In case of × 1, the entered value is set as it is.

## 15. Input field

The name of the item to be entered and the entered number are displayed.

## 16. Keypad icon

The large numeric entry keypad will appear on-screen any time the small keypad icon is pressed.

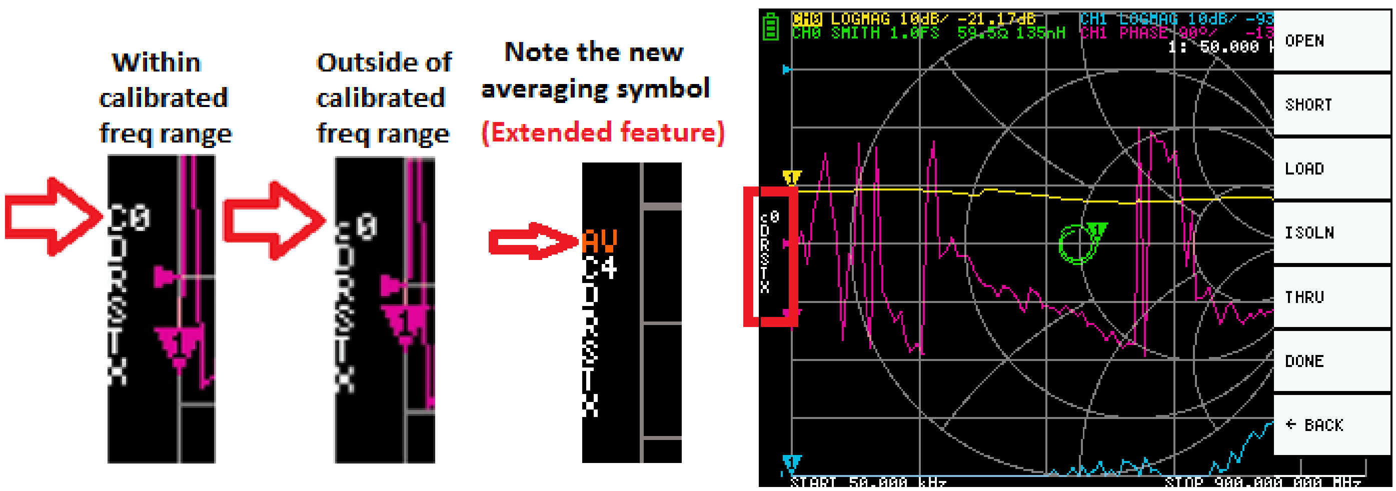
**NOTE:** Older versions of firmware do not show the keypad icon (16) although touching the blank bar at the lower right corner of the display will make the on-screen keypad appear.

# 5. Starting a measurement

## Basic measurement sequence

1. Set the frequency range to be measured – Use START/STOP or SPAN/CENTER
2. Perform calibration (and save!)
3. Connect the Device Under Test (DUT) and measure

# 6. Calibration method




Calibration should basically be performed whenever the frequency range to be measured is changed. If the error has been corrected correctly, the calibration status display on the screen will be:

**Cn DRSTX** Where: **n** is the calibration dataset number being loaded.

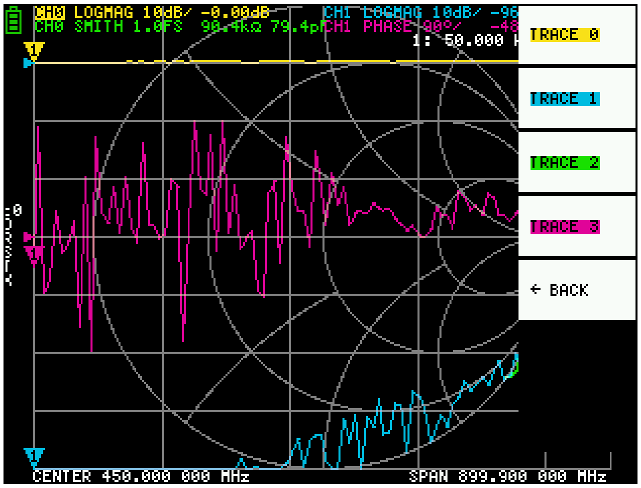
However, the NanoVNA can estimate the existing calibration information and display corrections to some extent. This will happen if the frequency range is changed after loading the calibration data. At this time, the calibration status display on the screen is **cn DRSTX** Where: **n** is the calibration dataset number being loaded. See image above.

1. Reset current calibration state. Select **CAL MENU →RESET** and then **→CALIBRATE**
2. Connect OPEN standard to CH0 port and execute **→OPEN** .
3. Connect SHORT standard to CH0 port and execute **→SHORT** .
4. Connect the LOAD standard to the CH0 port and execute **→LOAD** .
5. Connect the LOAD standard to CH0 and CH1 ports and execute **→ISOLN** .
   If there is only one load, the CH0 port can be left unconnected.
6. Connect a cable between the CH0 and CH1 ports, and execute **→THRU** .
7. Finish calibration and calculate error correction information **→DONE**
8. Specify the dataset number (0 to 4) and save. **→SAVE 0 (0 is the power-on/reset default)**

** Each calibration step should be completed after the display is sufficiently stable. A new averaging feature in QRP73's firmware is described in the Developer Extended Features section.

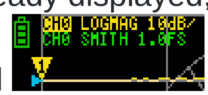
# 7. Function

## Trace display




Up to four traces can be displayed, one of which is the active trace.

Traces can display only what is needed. To switch the display, select **`DISPLAY →TRACE →TRACE n`** .
The following methods can be used to switch the active trace.

- Tap the trace marker you want to activate and then tap it again to toggle its state.
- Select **`DISPLAY →TRACE →TRACE n`** to display. (If already displayed, it will be temporarily hidden)
- The active trace will have its channel text inverted  (CH0 LOGMAG is active)



### Trace format

Although each trace can have its own displayed format, you can only change the format of an ***active trace***.
To assign a format, set the trace to ***active*** (see above) then select: **`DISPLAY →FORMAT`**

The description and unit of measurement of each format is as follows:

- **`LOGMAG`** : Logarithm of absolute value of measured value (dB per div)
- **`PHASE`** : Phase in the range of -180 ° to + 180 ° (90 degree default)
- **`DELAY`** : Delay (pico or nano seconds)
- **`SMITH`** : Smith Chart (Impedance scale is normalized during calibration)
- **`SWR`** : Standing Wave Ratio (can be scaled to show 1, 0.1 or 0.01 per div)
- **`POLAR`** : Polar coordinate format (Impedance scale is normalized during calibration)
- **`LINEAR`** : Absolute value of the measured value (dB per div)
- **`REAL`** : Real number of measured value
- **`IMAG`** : Imaginary number of measured value
- **`RESISTANCE`** : Resistance component of the measured impedance (ohms per div)
- **`REACTANCE`** : Reactance component of the measured impedance (ohms per div)

### Trace channel

NanoVNA has two ports, **CH0** and **CH1** . The following S parameters can be measured at each port.

- CH0 S11 (reflection loss)
- CH1 S21 (insertion loss)

To change the trace channel, select
**DISPLAY →CHANNEL →CH0 REFLECT** or **DISPLAY →CHANNEL →CH1 THROUGH** .

## Scale Menu

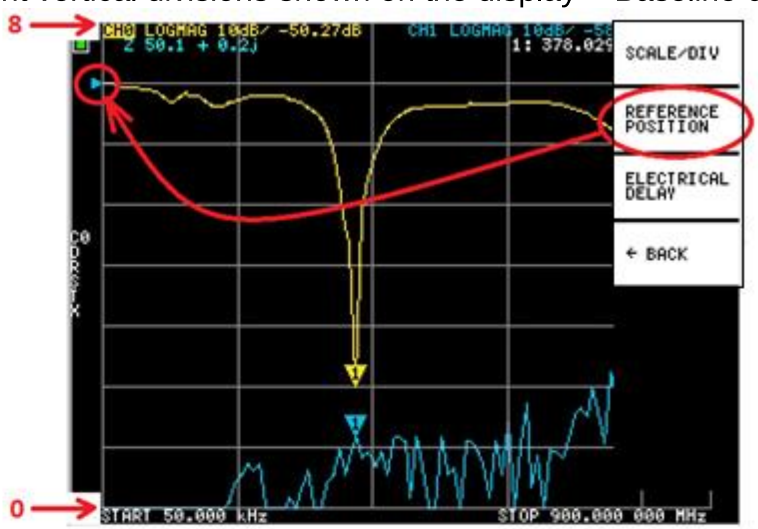
There are three sub-menus that allow adjustment of display scaling
Select **DISPLAY →SCALE**.

### Scale

The various vertical scales change according to the format of the ***active*** trace.
Refer to the previous section on Trace Format and how to select an active trace.

### Reference Position

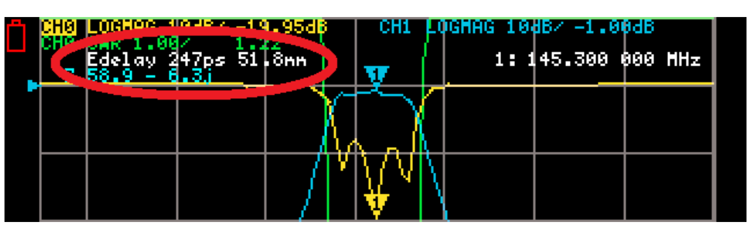
There are eight vertical divisions shown on the display - Baseline 0 through to 8:



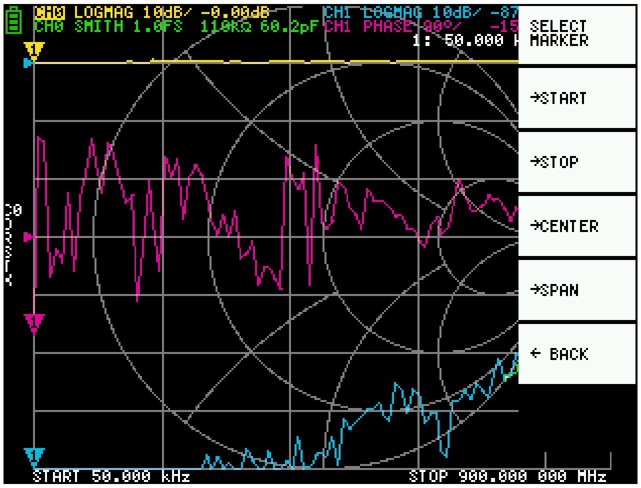

The current **Reference Position** is shown by the small triangle and is **set to 7 here**.

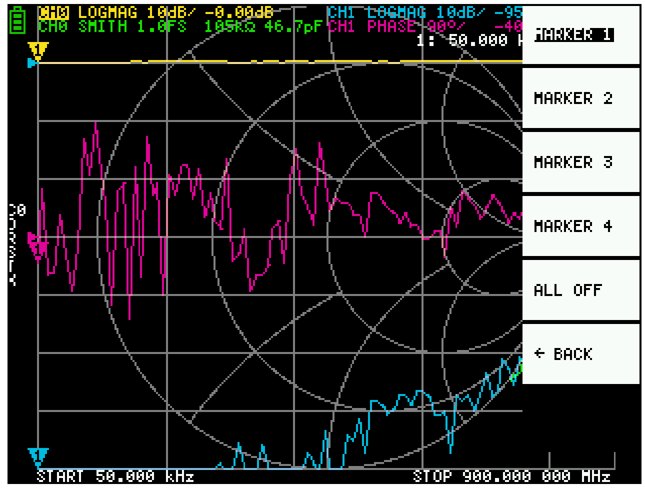
### Electrical Delay

The electrical delay parameter is used to compensate for certain external physical characteristics such as but not limited to electrical length. Care must be taken if this is set to anything other than 0.
Only if the electrical delay value is non-zero, will the current value be displayed:



# Markers




Up to 4 markers can be displayed.
Markers are displayed from **MARKER →SELECT MARKER →MARKER n .**
When you display a marker, the active marker is set to the displayed marker ‘**n**’.

Essentially, these commands allow you to set the frequency **range** or **span** using the position of the **active** marker(s). Make sure the current on-screen frequency span is greater than what you actually require so you can use the marker settings to 'zoom-in'.

## Setting frequencies from marker(s)

You can set the frequency range from the Marker Menu as follows:

- **MARKER →START** Sets the start frequency to the *active* marker’s frequency.

  **Example**: For simplicity, enable only one marker for this.

  Drag the active marker to the starting frequency of the range you want to investigate. You can read the marker’s frequency on-screen at the upper right area of the display as you move it.

  Open the Marker menu and click the **Start** button and you will see the start frequency at the lower left of the display change to the current marker freq.

- **MARKER →STOP** Sets the stop frequency to the *active* marker’s frequency

  Example: Now move the marker to the maximum frequency of the range you're interested in and again, open the Marker menu and press the **Stop** button.

  The upper frequency shown at the lower right of the display will now change to that value. You now know how to set the start and stop frequencies using the marker menu.

- **`MARKER →CENTER`** Sets the frequency of the *active* marker to be the center frequency. The span is adjusted to maintain the current range as much as possible.

  Set the Start frequency as before but the second entry will be to move the marker to the center frequency you're interested in and press the Center menu button. This will automatically adjust the start and stop frequencies to be equal on either side of center.

- **`MARKER →SPAN`** Sets the absolute frequency span to the ***last two active markers***. You need to have any two markers (M1-M4) enabled for the Span button to work. If only one marker is displayed, nothing happens.

  **NOTE:** It doesn't matter if you have 3 markers displayed. If the last two you moved were M2 and M3 and even if M2 is at a higher frequency than M3, the absolute span will be set using the lowest and highest values of the 2 markers.

# Time domain operation

The NanoVNA can simulate time domain measurements by signal processing frequency domain data.

Select **`DISPLAY →TRANSOFRM →TRANSFORM ON`** to convert measurement data to the time domain.
If **`TRANSFORM ON`** is enabled (Inverted white text on black background), the measurement data is immediately converted to the time domain and displayed.
The relationship between the time domain and the frequency domain is as follows.

- Increasing the maximum frequency increases the time resolution
- The shorter the measurement frequency interval (ie, the lower the maximum frequency), the longer the maximum time length

For this reason, the maximum time length and time resolution are in a trade-off relationship. In other words, the *time length is the distance*.

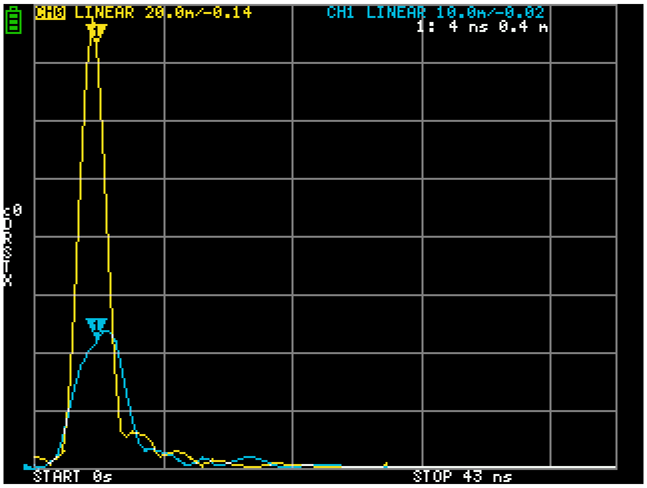
- If you want to increase the maximum measurement distance, you need to lower the maximum frequency.
- If you want to specify the distance accurately, you need to increase the maximum frequency.

  **HINT** – Use a lower frequency to measure a longer length and a higher frequency to measure a shorter length and adjust accordingly for accurate results.

## Time domain bandpass

In bandpass mode, you can simulate the DUT response to an impulse signal.

**NOTE:** The trace format can be set to **`LINEAR, LOGMAG or SWR`** .
The following is an example of the impulse response of a bandpass filter.

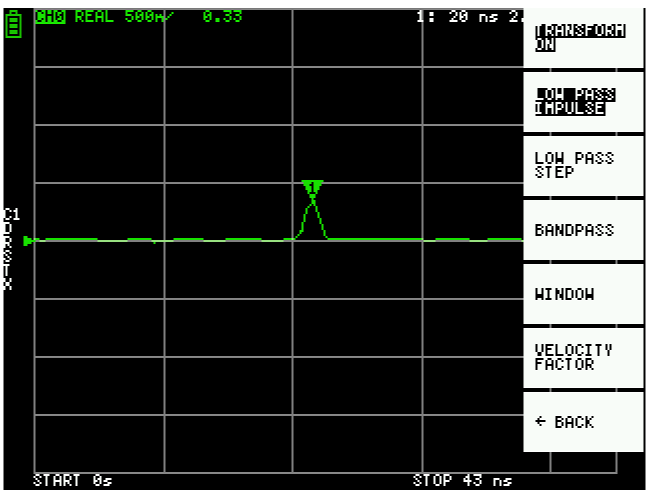



## Time domain low pass impulse

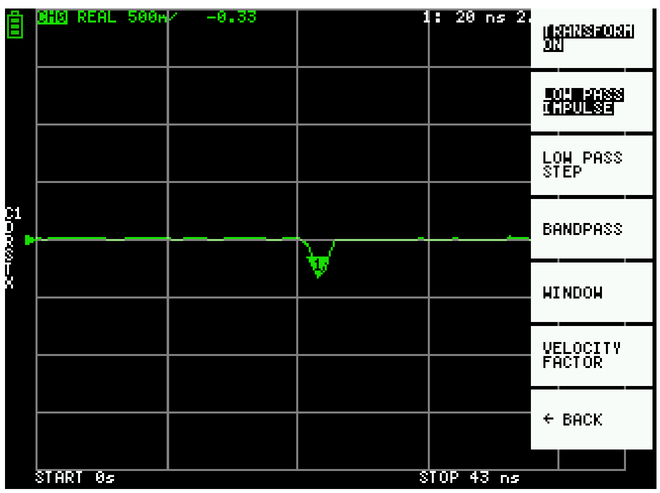
In low-pass mode, you can simulate TDR. In low-pass mode, the start frequency must be set to 50 kHz, and the stop frequency must be set according to the distance to be measured.

The trace format can be set to **`REAL`** .
Examples of Impulse response in open state and impulse response in short state are shown below.

Open:




Short:



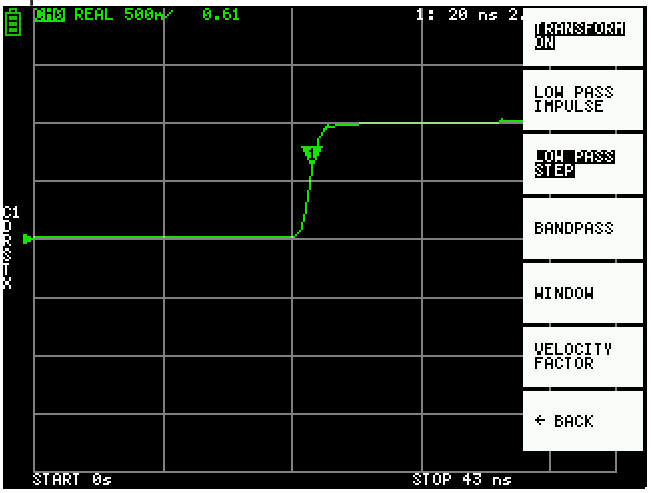

## Time domain low pass step

In low-pass mode, you can simulate TDR. In low-pass mode, the start frequency must be set to 50 kHz, and the stop frequency must be set according to the distance to be measured.

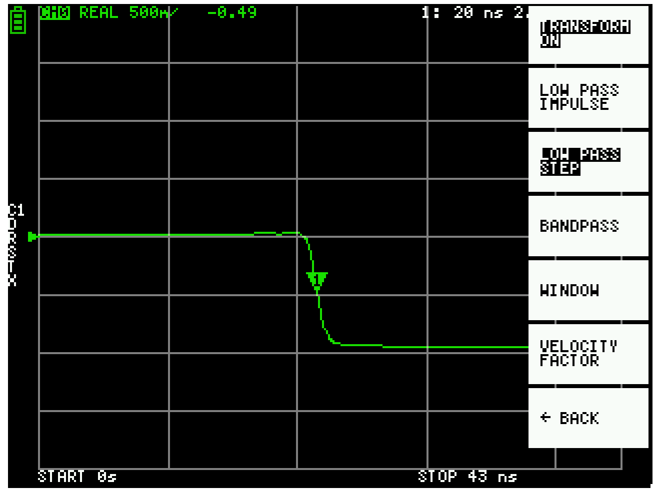
The trace format can be set to **`REAL`** .

Examples of Step response in open state and Step response in short state are shown below.

Open:



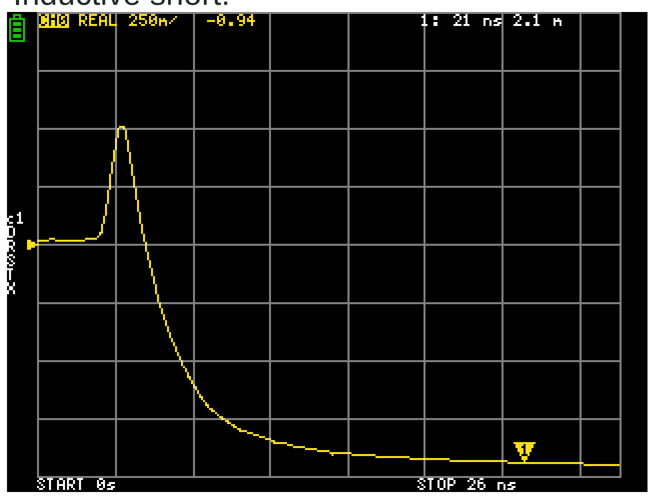

Short:



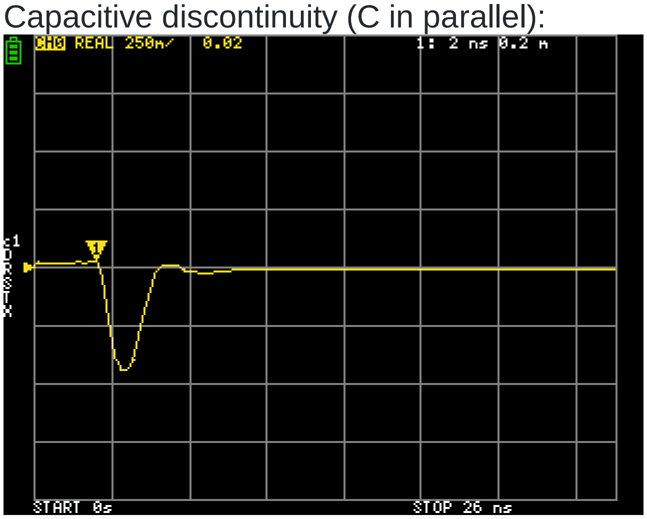

**Step response examples**

Capacitive short:

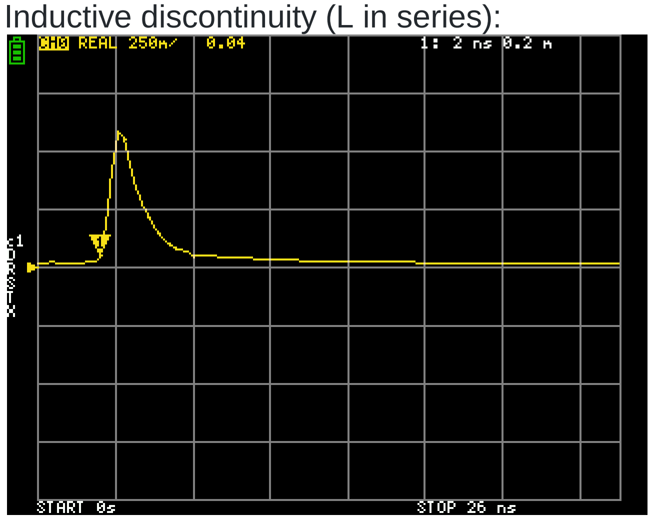

Inductive short:




Capacitive discontinuity (C in parallel):




Inductive discontinuity (L in series):



## Time domain window

The range that can be measured is a finite number, and there is a minimum frequency and a maximum frequency. A window can be used to smooth out this discontinuous measurement data and reduce ringing.

There are three levels of windows.

- MINIMUM (no window, ie: same as rectangular window)
- NORMAL (equivalent to Kaiser window β = 6)
- MAXIMUM (equivalent to Kaiser window β = 13)

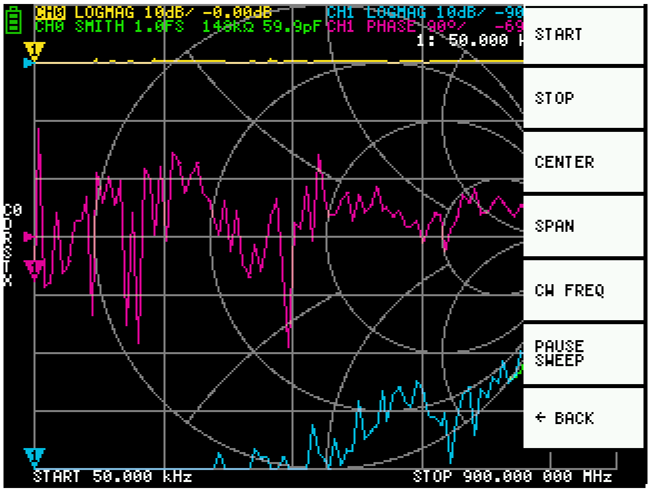
MINIMUM provides the highest resolution and MAXIMUM provides the highest dynamic range. NORMAL is in the middle.

## Setting the wavelength (Velocity) factor in the time domain

The transmission speed of electromagnetic waves in the cable varies depending on the material. The ratio to the transmission speed of electromagnetic waves in vacuum is called the wavelength factor (Velocity Factor, Velocity of propagation). This is always stated in the cable specifications.

In the time domain, the displayed time can be converted into distance. The wavelength shortening ratio used for distance display can be set with **`DISPLAY`** → **`TRANSFORM`** → **`VELOCITY FACTOR`** .
For example, if you measure the TDR of a cable with a wavelength reduction rate of 67%, specify **`67`** for the **`VELOCITY FACTOR`**. (Do not use the decimal point).

# Setting the measurement range




There are three types of measurement range settings.

- Setting the start frequency and stop frequency
- Setting the center frequency and span
- Zero span

## Setting the start frequency and stop frequency

Select and set **`STIMULUS →START`** and **`STIMULUS →STOP`** , respectively.

## Setting the center frequency and span

Select and set **`STIMULUS →CENTER`** and **`STIMULUS →SPAN`** respectively.

## Zero span

Zero span is a mode in which one frequency is sent continuously without frequency sweep.

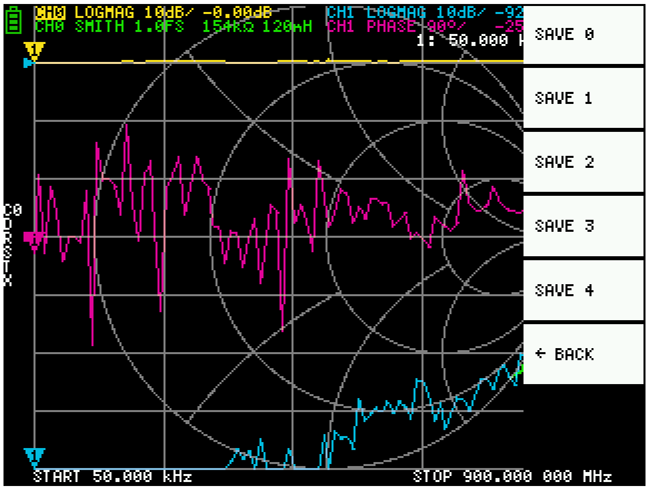
Select and set **`STIMULUS →CW FREQ`** .

## Temporarily stop measurement

**`STIMULUS →PAUSE SWEEP`** is selected, measurement is temporarily stopped.

Note: Versions of firmware prior to Oct 23, 2019 suspended USB data during a pause if the device was not already connected to a computer. Please ensure you are using current firmware.

# Recall calibration and settings

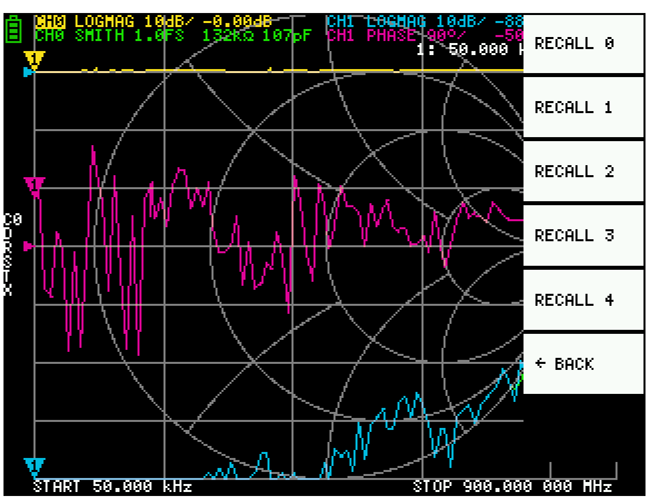


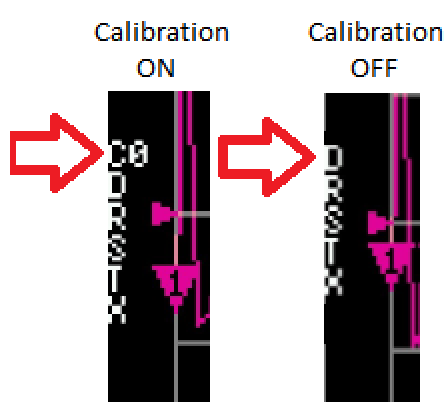

Up to 5 calibration datasets can be saved. NanoVNA loads Saved **0** data immediately after startup.

Calibration data is data that includes the following information:

- Frequency setting range
- Error correction at each of the 101 native measurement points
- Trace setting status
- Marker setting status
- Domain mode settings
- Setting the wavelength shortening rate
- Electrical delay

You can save the current settings by selecting **`CAL MENU →SAVE →SAVE n`** .
Current calibration data can be reset by selecting **`CAL MENU →RESET`** .
**NOTE**: If you want to recalibrate, you ***must*** **reset** but **DO NOT save** (you will corrupt the cal data).
**`CAL MENU →CORRECTION`** indicates whether error correction is currently being performed. You can select this to temporarily stop error correction. (Inverted **button** text=**ON**, Normal button text=**OFF**)




**`RECALL`** the saved settings by selecting **`CAL MENU →RECALL →RECALL n`**

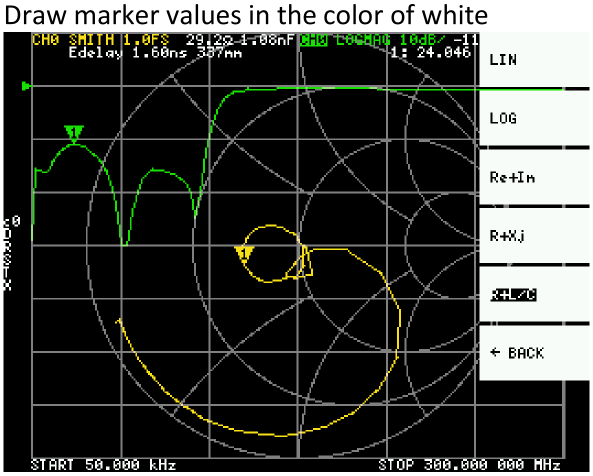
# Developer Extended Features

This section lists new features that _some_ individual NanoVNA Firmware developers are adding to the device.  These features change on a fairly regular basis, please refer to the firmware release areas of the individual developers. Links are in the ***How To Update the Firmware*** section, of this document.

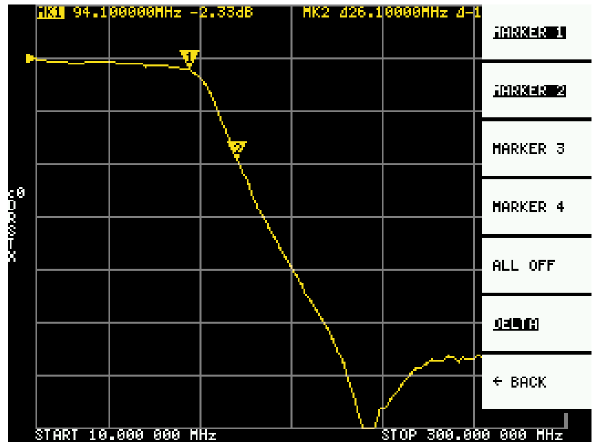
## Edy555/ttrftech:

0.5.4-20191210
improved accuracy of frequency (fixed #54).
There were frequency errors distributed in some tens of Hz in stimulus and IF. But now, the error becomes less than 1Hz. (It's ready to implement bandwidth setting as next issue.)

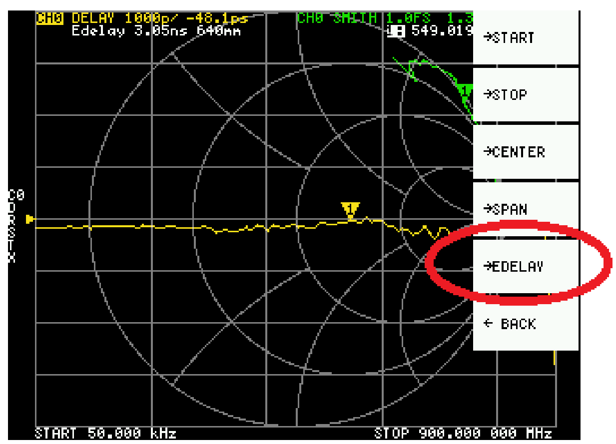
0.5.3-20191130
Add marker smith value menu, which enables to select value format LOG, LIN, ReIm, R+Xj, and R+L/C.
Draw marker values in the color of white



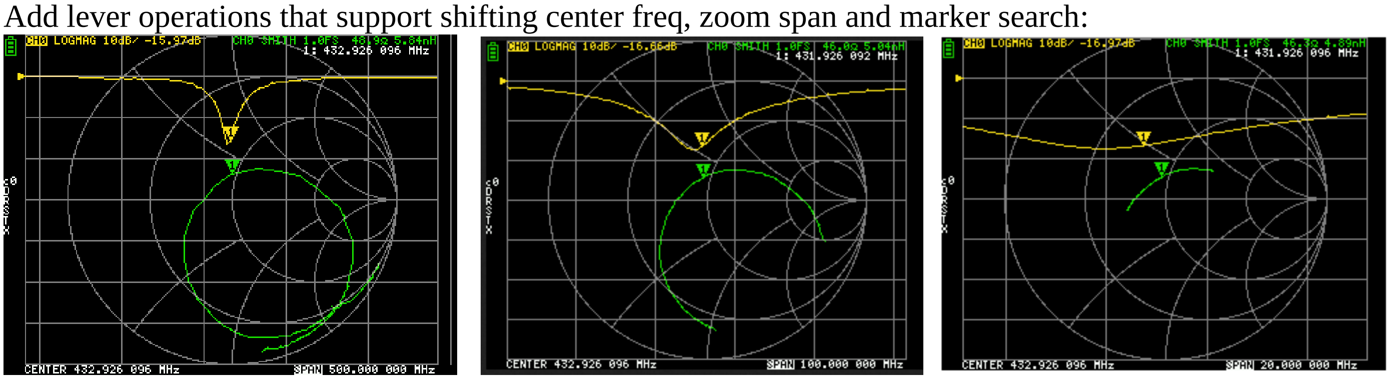

0.5.2-20191129
show multiple marker information
add marker delta mode




0.5.1-20191123 add marker operation groupdelay → edelay




0.5.0-20191117
Add lever operations that support shifting center freq, zoom span and marker search:

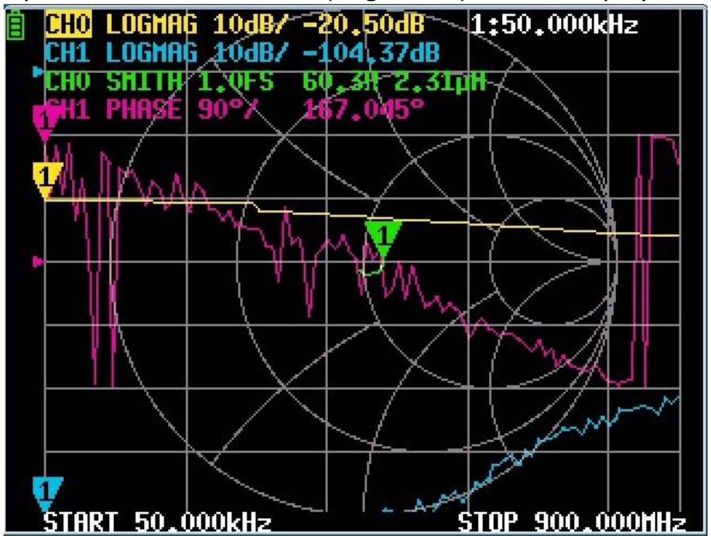



Change touch cal default for 2.8 inch lcd panel and some minor fixes
After updating firmware, recommend to do `touch cal` and do `clearconfig` if traces become corrupted.

## Hugen79:

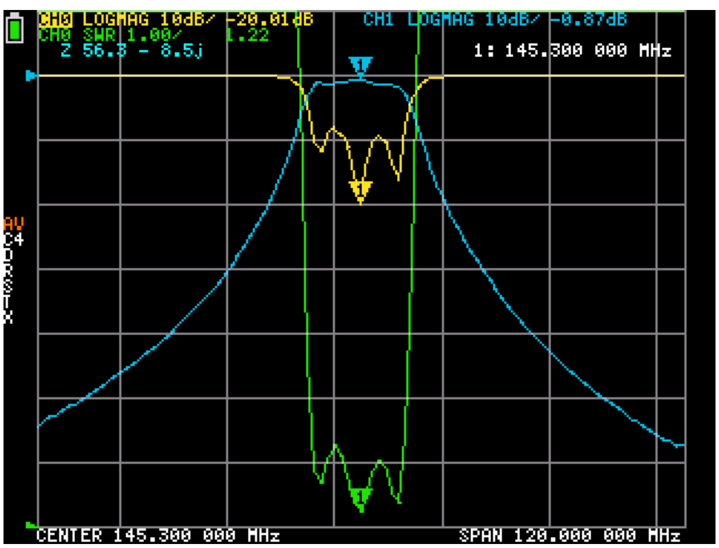
NanoVNA-H_20191125 **v0.4.0-3**: 1. The absolute value of the linear format is displayed.
2. Si5351 default 8mA output. (power=3)
3. The AA version shows 4 traces.

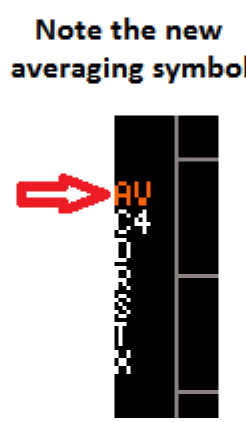
NanoVNA-H_20191115 **v0.4.0**: Using the code of nanoVNA-Q of qrp73, the driving of si5351 and aic3204 is more reasonable. By judging that the si5351 locking state is more reasonable than simply setting a delay, it can effectively avoid the noise caused by the un-stabilization of si5351. Unlike the compilation optimization of QRP73 and edy555, inline optimization is not disabled, and the refresh efficiency is better.
Due to limited flash space DUMP, SCANRAW, COLOR commands are not available.
Optimized for 4 Trace - AA (larger font) version display:




## QRP73:

The NanoVNA-Q-0.4.4-e679893 firmware release added average mode for better calibration.



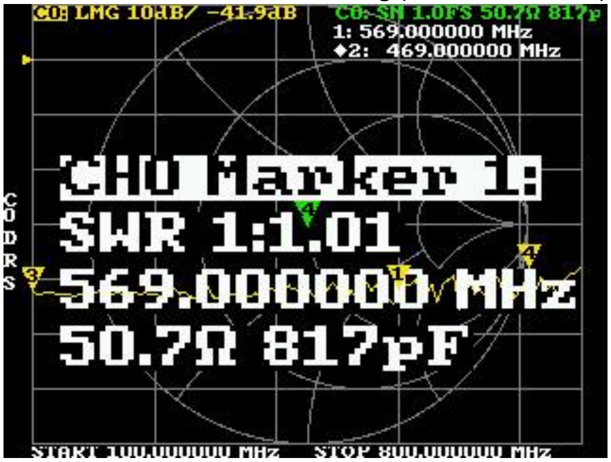




Use menu CAL => AVERAGE to enable average mode before calibration.
Note: average mode sweep time is much slower (about 3 seconds).
When you perform calibration with enabled average mode it allows to get better calibration.

**Reald:**

0.5.4 with bigger font and info screen
`compressed version info`

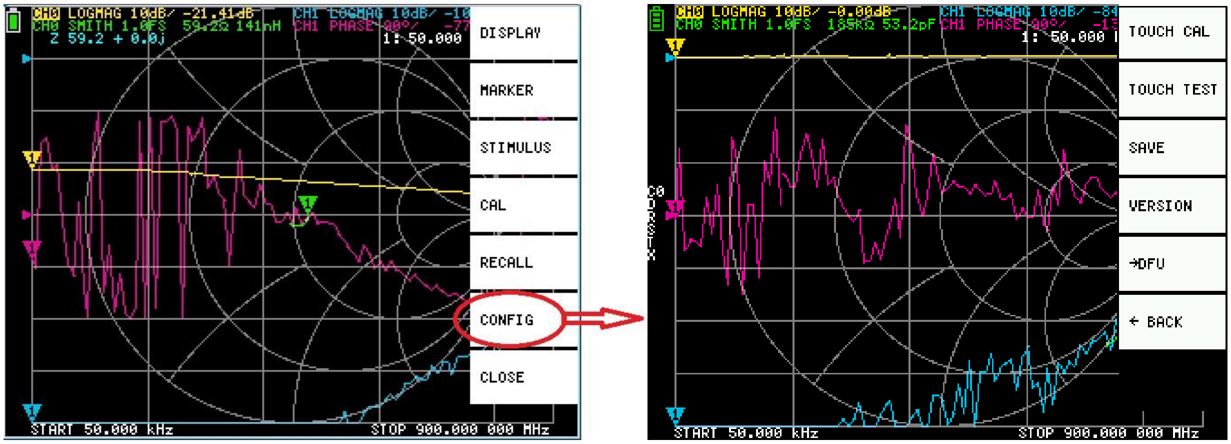
nanovna_191130-0.5.3
Bigger Standard fonts and big info screen for CH0. (Config -> "Info CH0").
Should be much better readable outdoors
Feature: Save infoscreen status to config (CONFIG -> SAVE)




Great for use as an antenna analyser!

# Device settings

The **CONFIG** menu allows you to make general settings for the device:

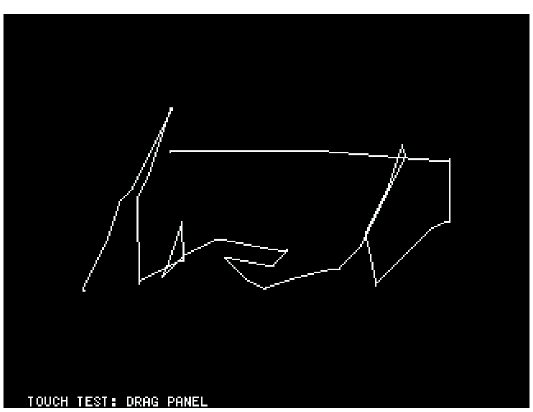



### Touch panel calibration and testing

The LCD touch panel can be calibrated when **CONFIG →TOUCH CAL** is selected if there is a large difference between the actual on-screen tap position and the recognized tap position.
**NOTE**: Be sure to save the settings with **SAVE**.



You can then test the LCD touch panel stylus tracking accuracy by selecting **CONFIG →TOUCH TEST**.
A line is drawn while dragging the stylus along the touch panel. When released from the touch panel, it returns to its original state. Repeat & save the touch screen calibration if tracking is incorrect.



## Saving device settings

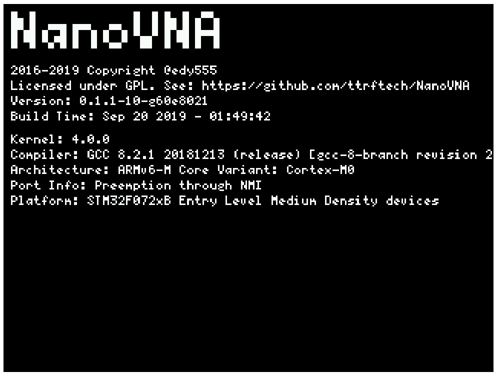
Select **CONFIG →SAVE** to save general instrument settings. General device settings are data that includes the following information:

- Touch panel calibration information
- Grid color
- Trace color
- Calibration data number loaded by default

The **CONFIG →SAVE** command does not apply to calibration settings.

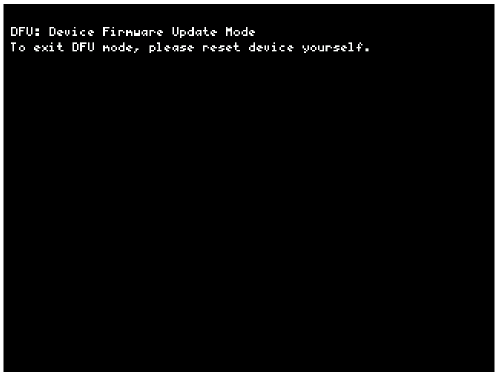
## Display version info

Select **CONFIG →VERSION** to display device version information.




## Firmware update mode via the touchscreen

**CONFIG →DFU RESET** and **ENTER DFU** mode. Select **RESET AND ENTER DFU** to reset the device and enter DFU (Device Firmware Update) mode. In this mode, firmware can be updated via USB.



# 8. How to update the firmware

## How to obtain the firmware

### ttrftech version firmware

Original firmware. It is versioned and frequently developed.

- GitHub releases: https://github.com/ttrftech/NanoVNA/releases
- CircleCI build: https://cho45.stfuawsc.com/NanoVNA/dfu.html

CircleCI has all the firmware with every commit. Use this if you want to try the latest features or check for problems.

### hugen79 version firmware

- Github Release page: https://github.com/hugen79/NanoVNA-H/releases

### QRP version firmware (hardware test and other features)

- Github Release page: https://github.com/qrp73/NanoVNA-Q/releases

### reald version firmware (good antenna analyser mode)

- Github Release page: https://github.com/reald/NanoVNA/releases

## Build yourself

You can easily clone one of the github repositories above and build it yourself. See Section 9.

## How to write (flash) firmware to the nanoVNA

There are various ways to write, but here we will explain using dfu-util . dfu-util is a cross-platform tool, and binaries are also provided on Windows.

### Writing with dfu-util (Ubuntu Linux)

There is dfu-util in the standard package repository.

```
sudo apt-get install dfu-util dfu-util --version
```

Start the device in DFU mode. Use one of the following methods to enter DFU mode.

- Turn on the power while jumpering the BOOT0 pin on the PCB. (Remove the jumper after turning on the power.) The screen turns white, but it is normal.
- **CONFIG →DFU RESET** AND **ENTER DFU** Select **RESET AND ENTER DFU**

Run the following command: build / ch.bin describes the path to the downloaded firmware file .bin.

```
dfu-util -d 0483:df11 -a 0 -s 0x08000000:leave -D build/ch.bin
```

## Writing with dfu-util (macOS)

It is recommended to install using homebrew .

Install the brew command.

```
ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

Install dfu-util command.

```
brew install dfu-util
```

Confirm that the dfu-util command can be started normally.

```
dfu-util --version
```

Start the device in DFU mode. Use one of the following methods to enter DFU mode.

- Turn on the power while jumpering the BOOT0 pin on the PCB. (Remove the jumper after turning on the power.) The screen turns white, but it is normal.
- **`CONFIG`** →**`DFU`** **`RESET`** AND **`ENTER DFU`** Select **`RESET AND ENTER DFU`**

Run the following command: build / ch.bin describes the path to the downloaded firmware file .bin.

```
dfu-util -d 0483:df11 -a 0 -s 0x08000000:leave -D build/ch.bin
```

## Writing with dfu-util (Windows)

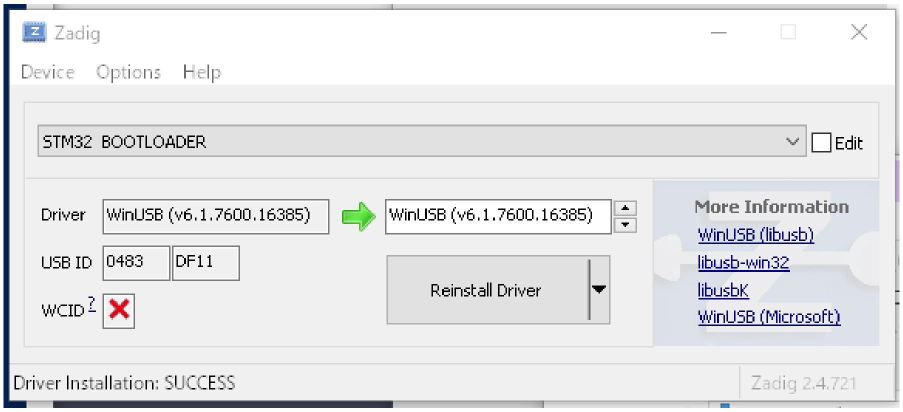
For Windows 7 & 8, the device driver is automatically installed when NanoVNA in DFU mode is connected, but dfu-util cannot be used with this device driver. Here, Zadig is used to replace the driver.

Start the device in DFU mode. Use one of the following methods to enter DFU mode.

- Turn on the power while jumpering the BOOT0 pins on the PCB. (Remove the jumper after turning on the power.) The screen turns white, but it is normal.
- Through the touch menu: **`CONFIG`** →**`DFU`** **`RESET`** AND **`ENTER DFU`** Select **`RESET AND ENTER DFU`**

Start Zadig with NanoVNA in DFU mode connected and use WinUSB as a driver for STM32 BOOTLOADER as follows.

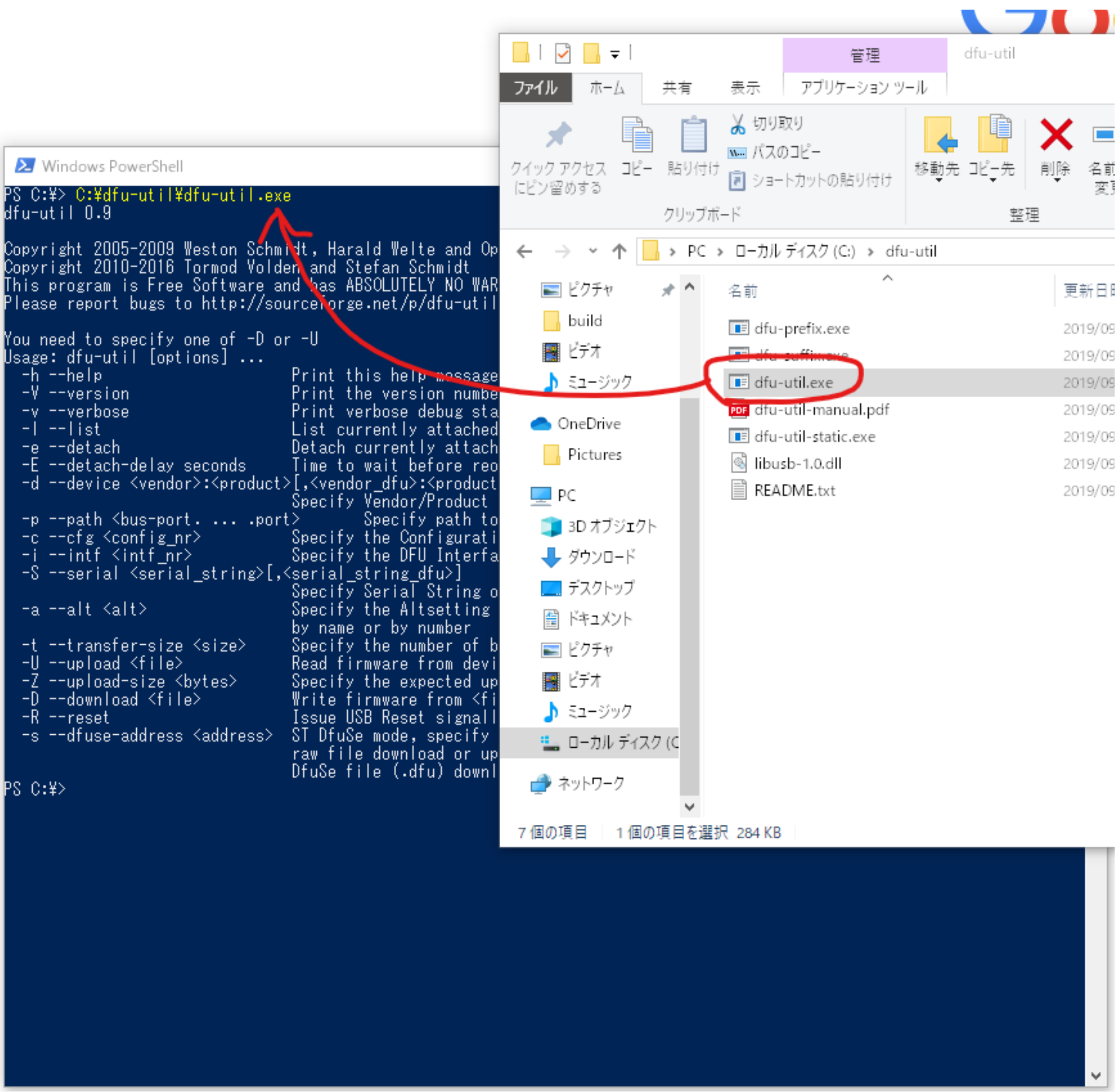
**NOTE:** Windows 10 is able to detect when the nanoVNA is in DFU mode and install the correct driver. If there are detections issues, delete all visible and hidden STM USB driver devices (but not the drivers) using the Windows Device Manager and re-scan for new devices. Windows should detect and install the correct drivers now. Using Zadig on Windows 10 may cause connection issues.




* If you want to restore the driver, find the corresponding device from “Universal Serial Bus Controller” in “Device Manager” and execute “Uninstall Device”. The driver is automatically installed when the USB connector is disconnected and reinserted.

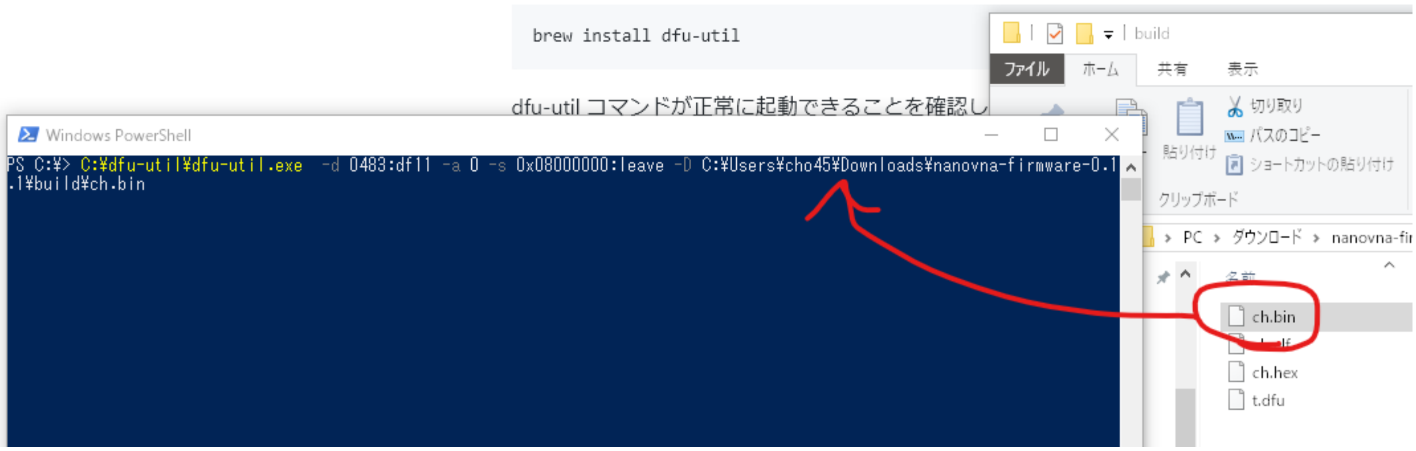
Next, place dfu-util. Download [dfu-util-0.9-win64.zip] from releases and [extract] it. Here, as an example, it is assumed that it is expanded to C: \ dfu-util (it does not matter where folder is located).

Right-click on the Start menu and select Windows PowerShell. A shell screen opens.



When dfu-util.exe is dragged and dropped from Windows Explorer to PowerShell, the path is automatically inserted. Dfu-util version can be displayed by starting with **`--version`** as follows.

```
C:\dfu-util\dfu-util.exe --version
```




Similarly, the firmware file can also be entered by dragging and dropping it from Explorer to PowerShell.

Run the following command: build / ch.bin describes the path to the downloaded firmware file .bin.

```
C:\dfu-util\dfu-util.exe -d 0483:df11 -a 0 -s 0x08000000:leave -D build\ch.bin
```

# How to write (flash) firmware using the Windows GUI

For Windows users, using the ***DfuSE Demo*** tool provided by ST is very straight forward. Although the name includes the word 'Demo', it is a full utility that does not expire or is limited in any way.

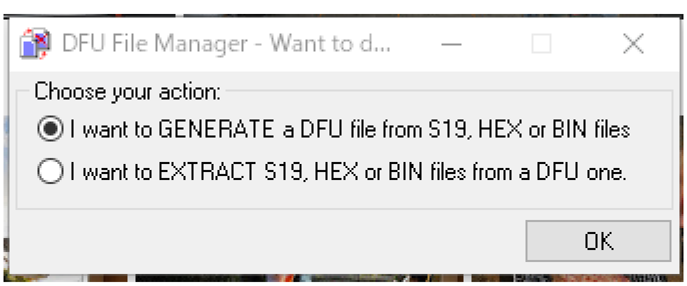
Download link: [STSW-STM32080 from the ST site]

There are 2 utilities installed:

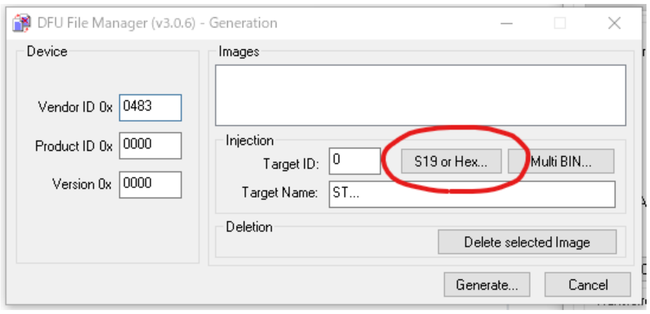
- **DFU File Manager** - a tool to create .dfu files from .bin or .hex files
- **DfuSe Demo** - A tool for writing .dfu files to devices is included.

## HEX or BIN file? Convert the file format with DFU File Manager.

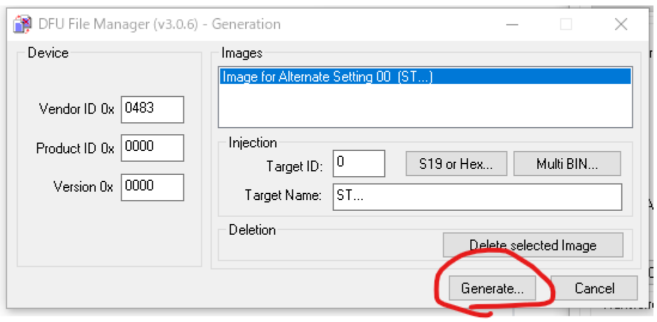
First, start DFU File Manager.




Select: **`I want to GENERATE a DFU file from S19, HEX or BIN files`**.




Click the **S19 or Hex...** button. **ch.hex** firmware **ch.hex** file such as **ch.hex .**



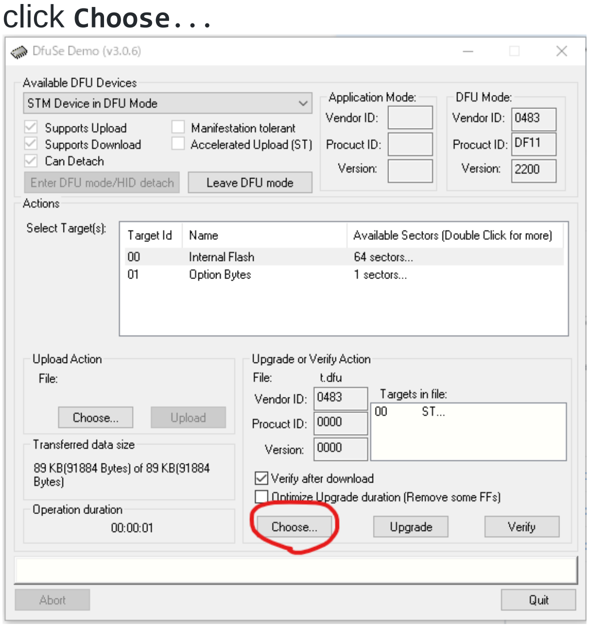

Click the **Generate...** button and create a .dfu file with a suitable name.

## Write (flash) firmware to the NanoVNA with DfuSe Demo

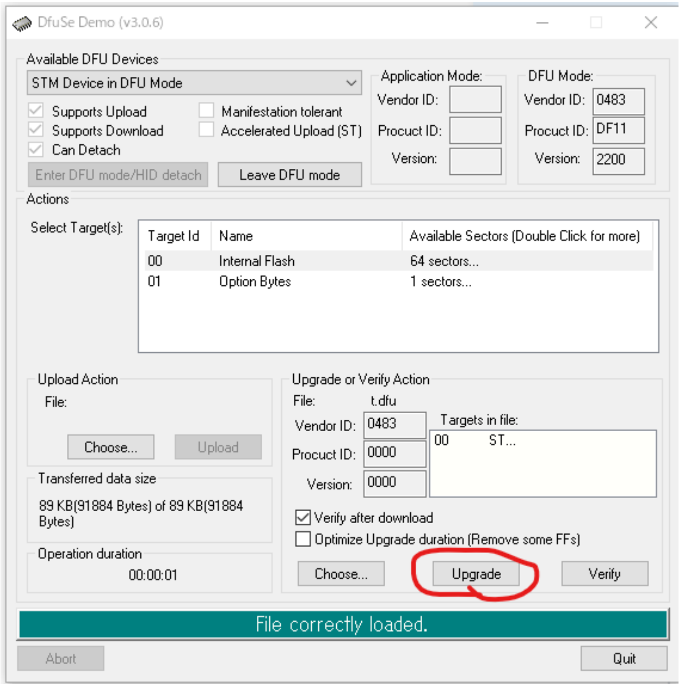
First start the NanoVNA in DFU mode. You can use one of the following methods to enter DFU mode.

- Turn on the power while jumpering the BOOT0 pins on the PCB. (Remove the jumper after turning on the power.) The screen turns white, but it is normal.
- Through the touch menu: **CONFIG →DFU RESET** AND **ENTER DFU** Select **RESET AND ENTER DFU**

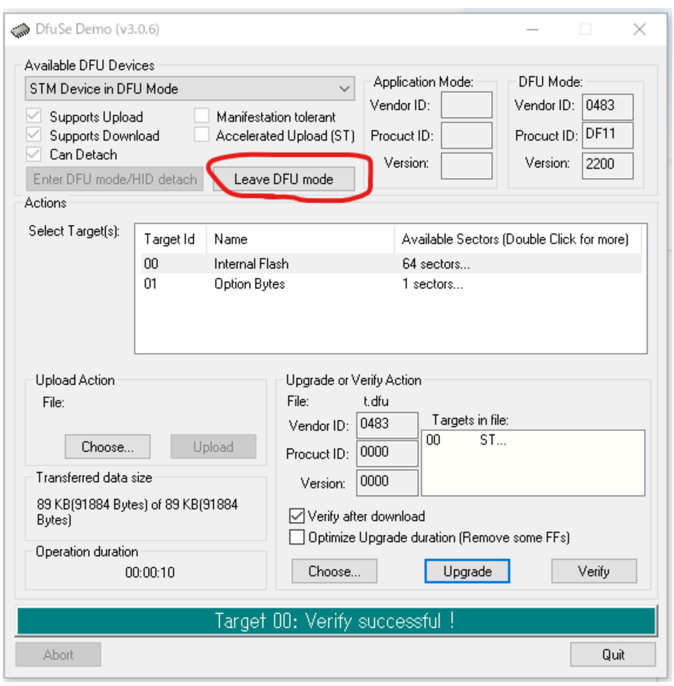
Start DfuSe Demo. Make sure that Available DFU Devices has ***STM Device in DFU Mode*** and click **Choose...**




Select the .dfu file you saved earlier.




Click the **`Upgrade`** button.




This screen will be displayed when writing is complete. Click the **`Leave DFU mode`** button to exit DFU mode. The device will reset and boot with the new firmware however, it is best to cycle the power on the unit.

**NOTE**: In some instances, if you get strange results when trying to calibrate your NanoVNA after flashing it with new firmware, you may need to use the Console Command: clearconfig

Refer to the Console Command reference for more information.

# 9. Firmware development guide

NanoVNA firmware development needs are as follows.

- Git
- gcc-arm-none-eabi
- make

If you already have these, you can build the firmware with **make** .

```
git clone git@github.com:ttrftech/NanoVNA.git cd NanoVNA git submodule update --init --recursive make
```

## Build with Docker

Use docker to build without bothering you. docker is a free, cross-platform container utility. It can be used to quickly reproduce a specific environment (in this case, the build environment).

Just install docker and run the following command:

```
docker run -it --rm -v $(PWD):/work edy555/arm-embedded:8.2 make
```

## On-chip debugging with Visual Studio Code

Visual Studio Code (hereinafter VSCode) is a multi-platform code editor provided free of charge by Microsoft. By installing Cortex-Debug Extension, on-chip debugging can be done with GUI.

The platform-dependent part is omitted, but in addition to the above, the following are required.

- openocd
- VSCode
- Cortex-Debug

Cortex-Debug is searched from Extensions of VSCode and installed.

### tasks.json

First, define a “task” to make the entire NanoVNA on VSCode.

```
{
    "tasks" :   [
        {
            "type" :   "shell" ,
            "label" :   "build" ,
            "command" :   "make" ,
            "args" :   [
            ],
            "options" :   {
                "cwd" :   "${workspaceRoot}"
            }
        }
    ],
    "version" :   "2.0.0"
}
```

Now you can make it as a task on VSCode.

## launch.json

Next, define how to start during Debug. Set as described in Cortex-Debug.

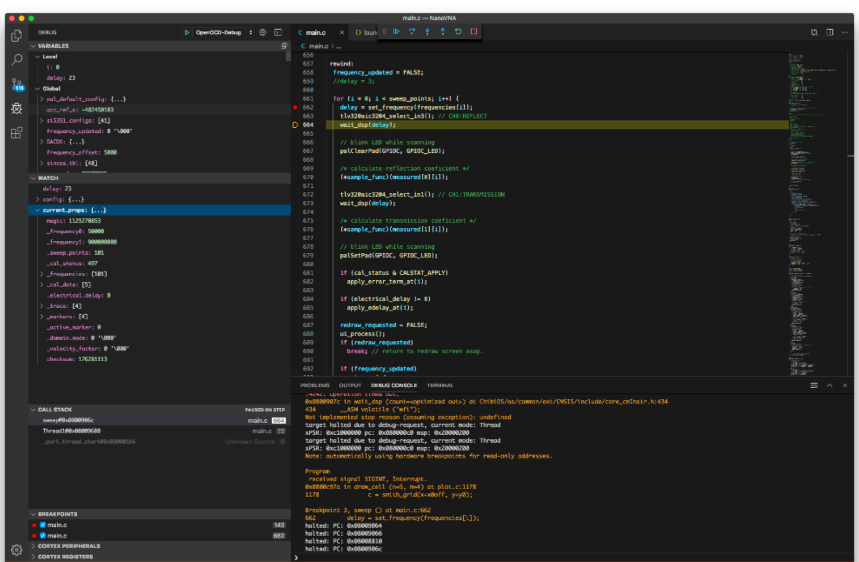
The following settings are for ST-Link. If you use J-Link, replace **interface/stlink.cfg** with **interface/jlink.cfg** .

```
{
    "version" :  "0.2.0" ,
    "configurations" :  [
        {
            "type" :  "cortex-debug" ,
            "servertype" :  "openocd" ,
            "request" :  "launch" ,
            "name" :  "OpenOCD-Debug" ,
            "executable" :  "build/ch.elf" ,
            "configFiles" :  [
                "interface/stlink.cfg" ,
                "target/stm32f0x.cfg"
            ],
            "svdFile" :  "./STM32F0x8.svd" ,
            "cwd" :  "${workspaceRoot}" ,
            "preLaunchTask" :  "build" ,
        }
    ]
}
```

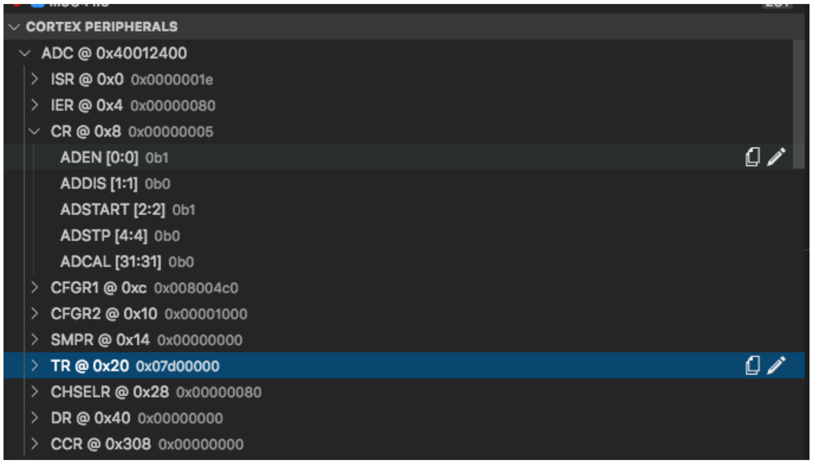
**svdFile** file specified in **svdFile** can be downloaded from the ST site . **svdFile** is not specified, there is no problem in operation.

## Start debugging

When Start Debugging ( **F5** ) is performed, OpenOCD starts automatically after the build by make and the firmware is transferred. When the transfer is complete, the reset handler breaks.

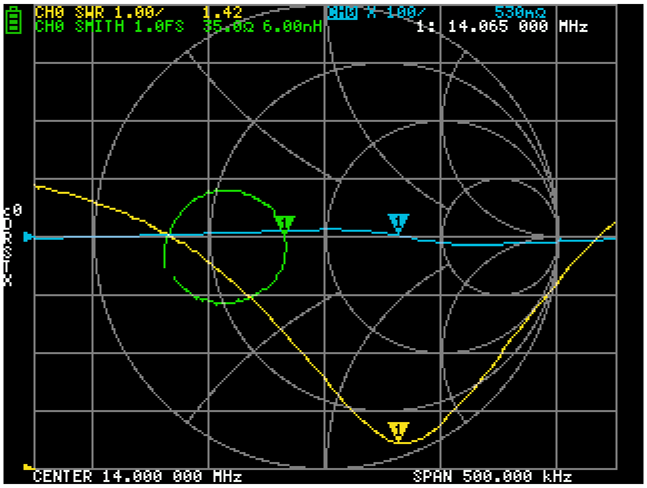


**svdFile** is specified, the defined MCU registers are displayed on the debug screen.


CORTEX PERIPHERALS
ADC @ 0x40012400
ISR @ 0x0 0x0000001e
IER @ 0x4 0x00000080
CR @ 0x8 0x00000005
ADEN [0:0] 0b1
ADDIS [1:1] 0b0
ADSTART [2:2] 0b1
ADSTP [4:4] 0b0
ADCAL [31:31] 0b0
CFGR1 @ 0xc 0x008004c0
CFGR2 @ 0x10 0x00001000
SMPR @ 0x14 0x00000000
TR @ 0x20 0x07d00000
CHSELR @ 0x28 0x00000080
DR @ 0x40 0x00000000
CCR @ 0x308 0x00000000

# 10. Examples of use

## Antenna adjustment (SWR)




An example of using NanoVNA as an antenna analyzer is shown below.

There are two important points in antenna adjustment:

- Whether the antenna is tuned and resonant (i.e., reactance is close to 0 at the desired frequency)
- Is the antenna's SWR low?

### Trace settings

Since only CH0 is used for antenna adjustment, calibration is performed for all items except `THRU` and `ISOLN` .
Set the trace as follows:

- Trace 0: CH0 SWR
- Trace 1: CH0 REACTANCE
- Trace 2: CH0 SMITH
- Trace 3: OFF

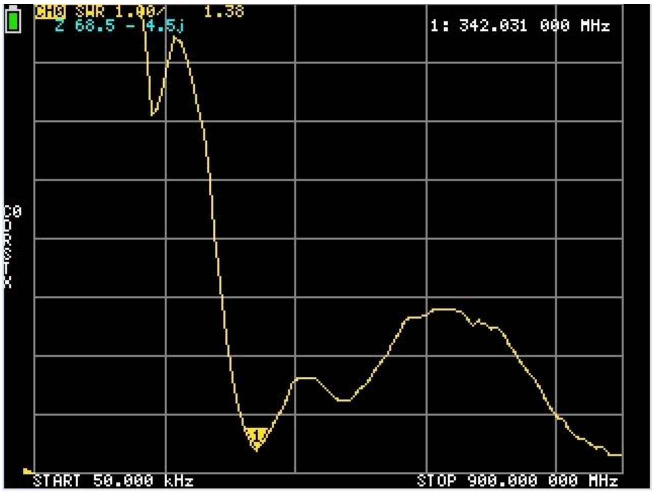
Set the frequency you want to tune the antenna to `CENTER` and set `SPAN` appropriately.
Look for frequencies where trace 1 displaying reactance is close to zero. Since the frequency is the tuning point, adjust the antenna if it is deviated so that the tuning point comes to the target frequency.

If the tuning point is at the desired frequency, check that trace 0 displaying the SWR is displaying a sufficiently low (close to 1) SWR. If the SWR is not enough (less than 2), the Smith chart is used for matching. In this case, matching may be performed using an antenna tuner directly under the antenna.

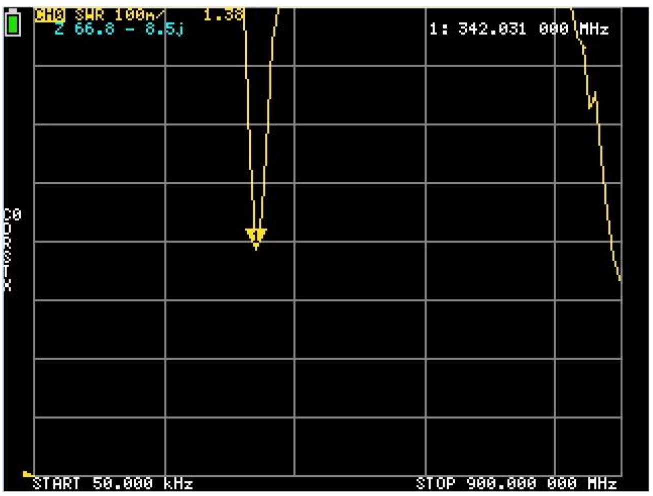
If the SWR drops, the antenna is tuned at the desired frequency and the adjustment of the antenna with a low SWR is complete.

For fine adjustment of SWR, the scale may be set to smaller values as shown:

The Reference is set to 0 (zero) and Scale is set to 1 (one):




The same measurement with even higher accuracy can be displayed with Scale set to 0.1:




And this is what can be displayed with Scale set to 0.01 – unknown how accurate this really is.


The displayed value is rounded to 2 decimal points but the plotted trace is showing 1.065:1

## Adjusting the bandpass filter

TODO

## Check the cable

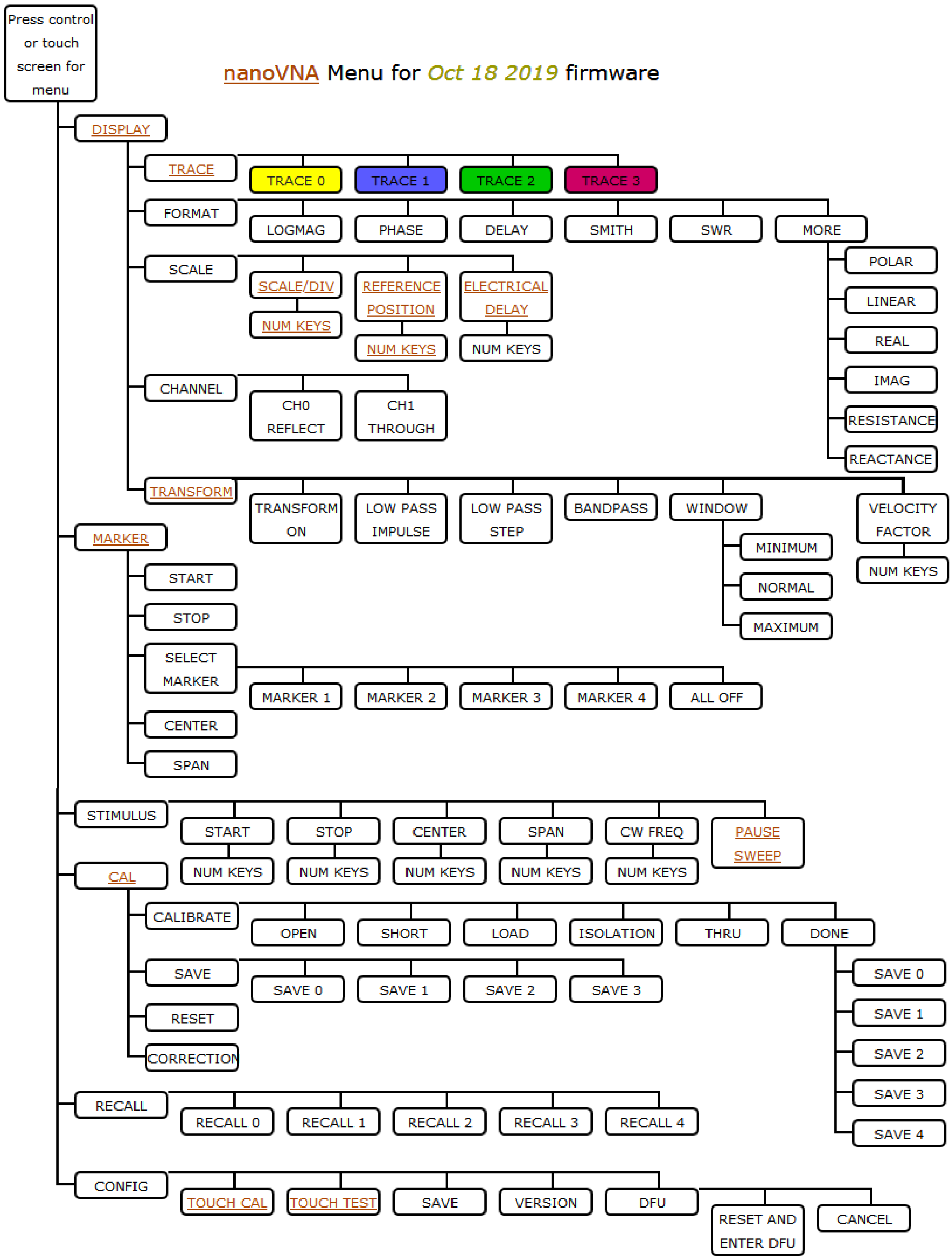
You can simulate TDR using the time domain low pass mode. By using TDR, you can find faults in the transmission path.

TODO

## Common mode filter measurement

TODO

# 11. Typical NanoVNA Touchscreen Menu Map




Source: https://oristopo.github.io/nVhelp/html/Wnano.htm

# 12. Contributors

| | | |
|---|---|---|
| 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 |

Thanks to several additional NanoVNA forum members for their suggestions to improve this guide.