# NanoVNA User Guide

iii. Cable check

iv. Common mode filter measurement

# Introduction

This document is an unofficial user guide for NanoVNA. The URL is https://cho45.github.io/NanoVNA-manual/.

It is managed by the github repository .

Please send Pull-request when there are corrections such as discrepancies with the latest firmware.

It is also available in PDF format on the Releases page on GitHub.

- https://github.com/cho45/NanoVNA-manual/releases

## What is 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 have almost the same components on the circuit, and common firmware is available.

## What is required for operation

You need at least the following:

- NanoVNA body
- SMA LOAD 50Ω
- SMA SHORT
- SMA OPEN

- SMA Female to Female Through Connector
- SMA Male to Male cable × 2

## Basics of NanoVNA

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

NanoVNA measures the following factors:

- Input voltage I/Q signal
- I/Q signal of reflected voltage
- I/Q signal of passing voltage

From here we calculate:

- Reflection coefficients S11
- Transmission coefficient S21

The following items that can be calculated from these can be displayed.

- Reflection loss
- Passage loss
- Complex impedance
  - resistance
  - reactance
- SWR

Such.

# Oscillation frequency of NanoVNA

The NanoVNA measures the reflection coefficient and transmission coefficient at 101 points in the measurement frequency band.

The local frequency of NanoVNA is 50kHz to 300MHz. Frequencies above this use harmonic mode. The fundamental wave is not attenuated even in harmonic mode. The usage modes for each frequency are as follows.

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

Especially when checking the gain of the amplifier, it is necessary to be careful that the fundamental wave is always input.

The input is in each case converted to an intermediate frequency of 5kHz. The signal is analog-to-digital converted at 48kHz sampling. The digital data is processed by the MCU.

# First thing

It must always be calibrated first before it can be used. Initially calibrate as follows.

- Make sure START is at 50kHz
- Make sure STOP is 900MHz
- Calibrate according to the calibration method

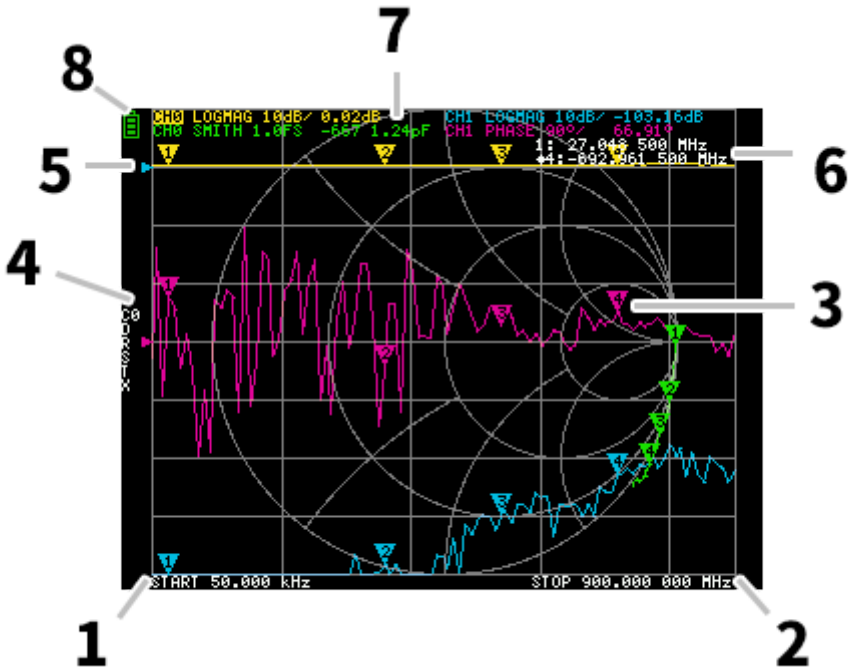
# input method

The NanoVNA has the following inputs.

- Touch panel long tap
- Lever switch
  - L/L long press
  - R / R long press
  - Push / Push and hold
- Power slide switch

# How to read the screen

## Main screen




### 1. START 2. STOP frequency

Displays the respective frequencies when start/stop is specified.

### 3. Marker

The position of each marker on the trace is displayed. The selected marker can be moved by the following methods.

- Drag the marker on the touch panel
- Press and hold LR on the lever switch

## 4. Calibration status

The data number of the calibration being read and the error correction applied are displayed.

- `C0 C1 C2 C3 C4` : Each indicates that the corresponding number of calibration data is loaded.
- `c0 c1 c2 c3 c4` : Each shows that the corresponding number of calibration data has been loaded, but the frequency range has changed after loading and indicates that compensation is being used for error correction.
- `D` : directivity Indicates that error correction is applied
- `R` : refrection 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` : isolation (crosstalk) indicates that error correction is applied

## 5. Reference position

Indicates the reference position of the corresponding trace. `DISPLAY SCALE REFERENCE POSITION` You can change the position with.

## 6. Marker status

You will see the active marker selected and one previously active marker.

## 7. Trace status

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

For example `CH0 LOGMAG 10dB/ 0.02dB` , read as follows.

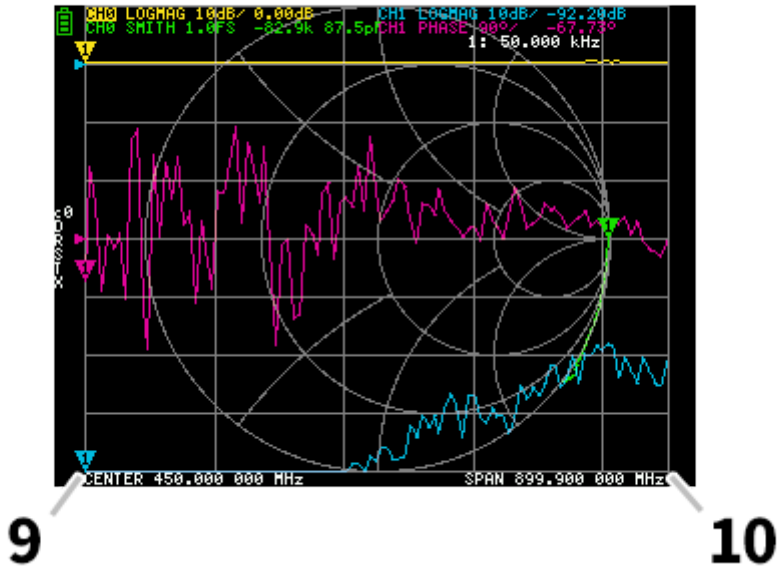
- Channel CH0 (reflection)
- format `LOGMAG`
- Scale is 10 dB
- Current value is 0.02dB

In addition, the channel display is reversed for the active trace.

### 8. Battery status

If the battery is installed and is `D2` already mounted on the PCB , an icon will be displayed according to the battery voltage.

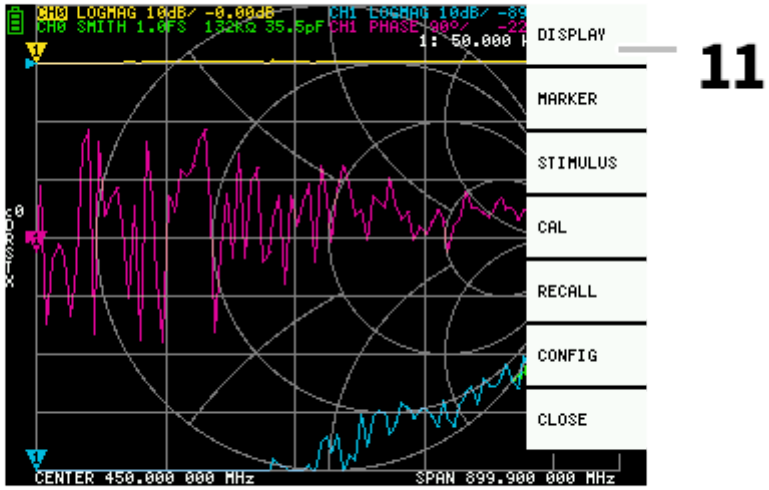
## Main screen 2



## 9. CENTER frequency 10. span

When the center frequency and span are specified, the respective frequencies are displayed.

## Menu screen



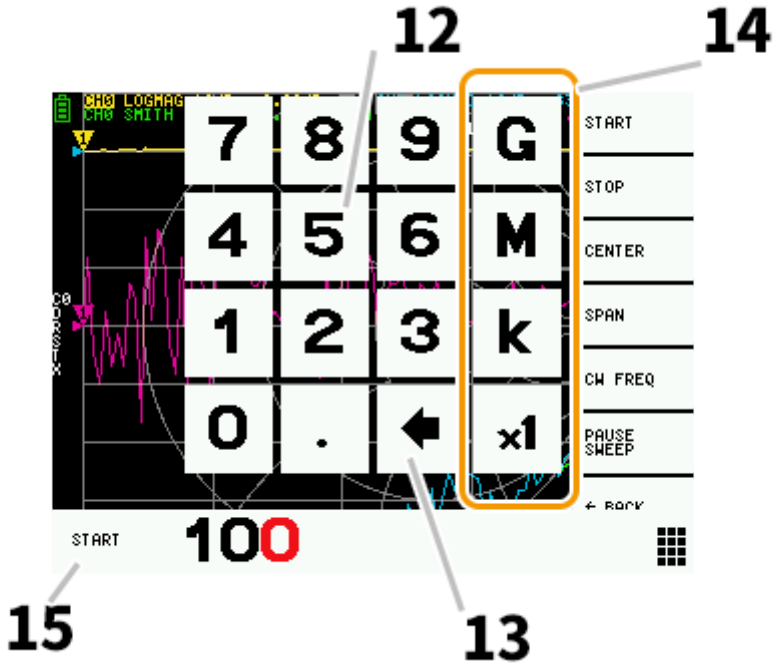

11

## 11. Menu

You can display the menu by the following operations.

- When you tap anywhere on the touch panel other than the marker
- Push in the lever switch

# Keypad screen




## 12. Numeric keys

Tap a number to enter one character.

## 13. Back key

Delete one character. If you have not entered any characters, the entry is canceled and the previous status is restored.

## 14. Unit key

Immediately ends the input by multiplying the current input by the appropriate unit. In case of ×1, the entered value will be set as it is.

## 15. Input field

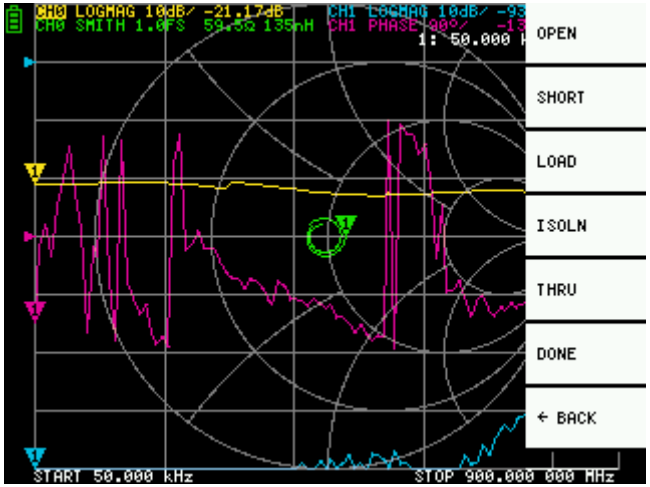
The input item name and the entered number are displayed.

# Start measurement

## Basic measurement sequence

1. Set the frequency range to measure
2. Calibrate
3. Connect DUT

# Calibration method




Calibration basically needs to be done every time the frequency range to be measured is changed. If the error is corrected correctly, the calibration status display on the screen will `Cn D R S T X` be. n is the data number you are loading.

However, the NanoVNA can supplement the existing calibration information and give a somewhat correct display. This happens if you change the frequency range after loading the calibration data. At this time, the display of the calibration status on the screen is `cn D R S T X` . n is the data number you are loading.

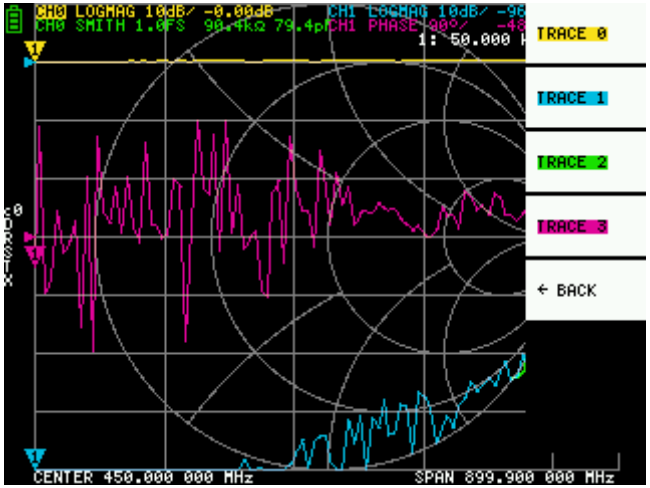
1. Reset current calibration state `CAL RESET`
2. Connect the OPEN standard to the CH0 port `CAL CALIBRATE OPEN` and execute.
3. Connect the SHORT standard to the CH0 port `CAL CALIBRATE SHORT` and run.
4. Connect the LOAD standard to the CH0 port `CAL CALIBRATE LOAD` and run.
5. Connect the LOAD standard to the CH0 and CH1 ports `CAL CALIBRATE ISOLN` and execute. The CH0 port can be left unconnected if there is only one load.

6. Connect the cables to the CH0 and CH1 ports, connect the cables with the through connector, `CAL CALIBRATE THRU` and execute.
7. Finish calibration and calculate error correction information `CAL CALIBRATE DONE`
8. Specify the data number and save. `CAL CALIBRATE SAVE SAVE 0`

*It is necessary to import each calibration data after the display is sufficiently stable.

# function

## Trace display




You can display up to four traces, one of which is the active trace.

The trace can display only what you need. `DISPLAY TRACE TRACE n` Select to switch the display .

There are the following methods to switch the active trace.

- Tap the trace marker you want to activate
- `DISPLAY TRACE TRACE n` Select to display. (If it is already displayed, you need to hide it temporarily)

### Trace format

Each trace can have its own format. To change the format of the active trace, `DISPLAY FORMAT` select the format you want to change.

The display of each format is as follows.

- `LOGMAG` : Logarithm of absolute value of measured value
- `PHASE` : Phase in the range of -180° to +180°
- `DELAY` : Delay
- `SMITH` : Smith chart
- `SWR` : Standing Wave Ratio
- `POLAR` : Polar coordinate format
- `LINEAR` : Absolute value of measured value
- `REAL` : Real number of measured value
- `IMAG` : Imaginary number of measured value
- `RESISTANCE` : Resistance component of the measured impedance
- `REACTANCE` : Reactance component of impedance of measured value

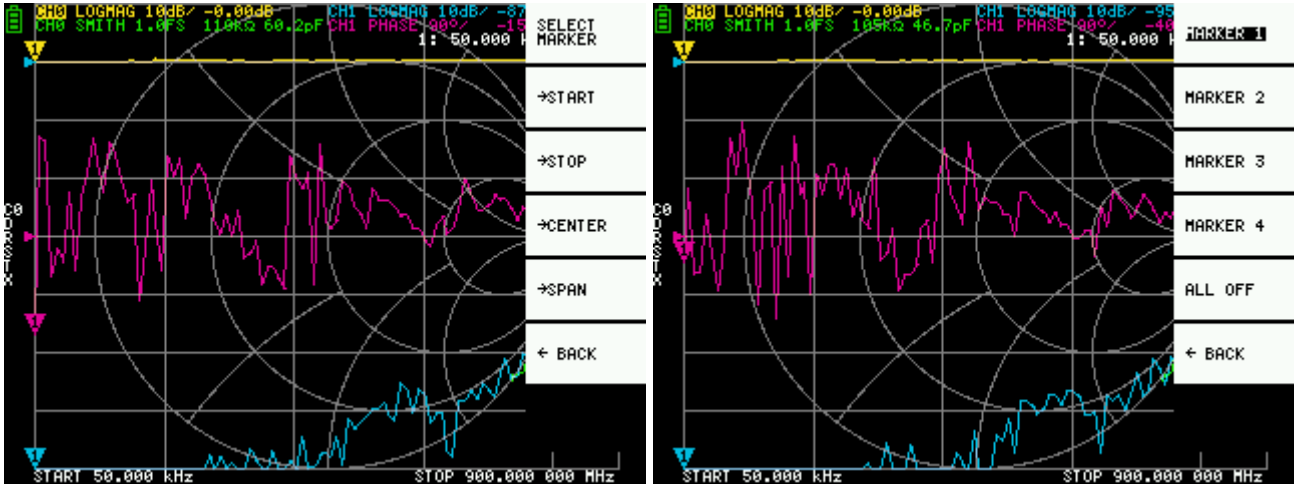
### Trace channel

The NanoVNA has `CH0` `CH1` two ports. The following S-parameters can be measured at each port.

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

To change the channel of the trace `DISPLAY` `CHANNEL` of `CH0 REFLECT` or `CH1 THROUGH` to select a.

## marker




You can display up to four markers. The marker is displayed `MARKER` `SELECT MARKER` `MARKER n` from. When you display a marker, the active marker is set to the displayed marker.

## Time domain operation

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

`DISPLAY` `TRANSOFRM` `TRANSFORM ON` Select to convert the measurement data to the time domain . `TRANSFORM ON` If is enabled, the measurement data is immediately transformed and displayed in the time domain.

The time domain and frequency domain have the following relationship.

- Time resolution increases as the maximum frequency increases
- The closer the measurement frequencies are (that is, the lower the maximum frequency is), the longer the maximum time length is.

Therefore, there is a trade-off between maximum time length and time resolution.

When the time length is called distance, the following can be said.

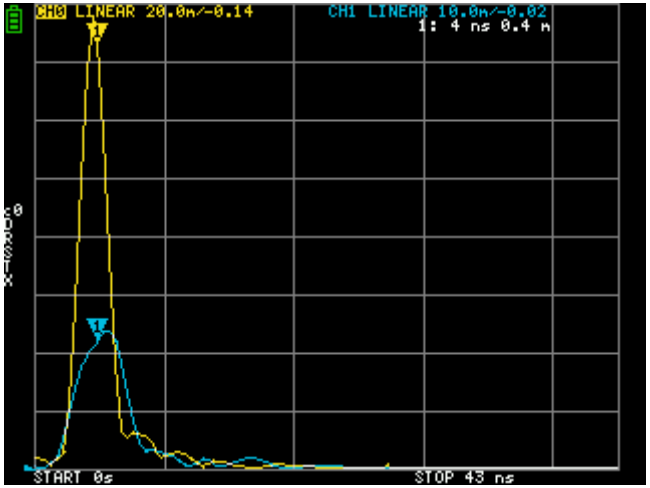
- If you want to increase the maximum measuring distance, you need to lower the maximum frequency.
- If you want to specify the distance accurately, you need to increase the maximum frequency.

## Time domain bandpass

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

The trace format `LINEAR` `LOGMAG` `SWR` can be set to.

Below is an example of the impulse response of a bandpass filter.

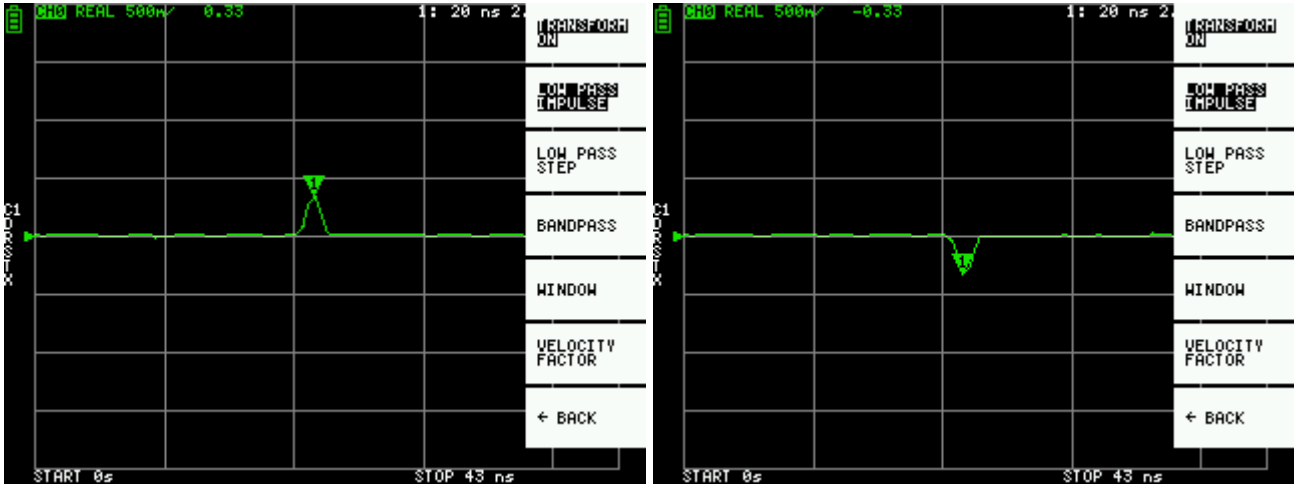



## Time domain low pass impulse

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

`REAL` You can set the trace format to .

An example of the step response in the open state and the impulse response in the short state is shown below.

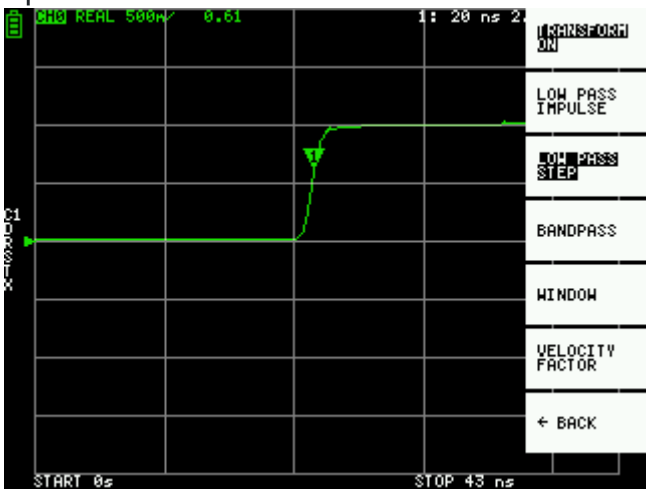



## Time domain lowpass step

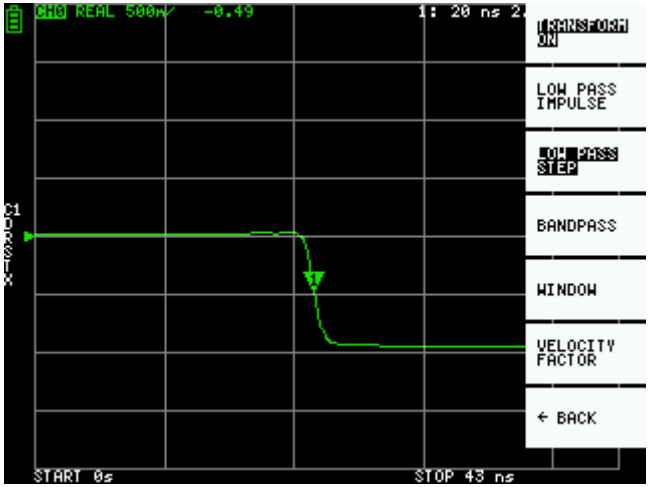
In lowpass mode, you can simulate TDR. In low pass mode, the start frequency must be set to 50kHz and the stop frequency must be set according to the distance you want to measure.

REAL You can set the trace format to .

open:



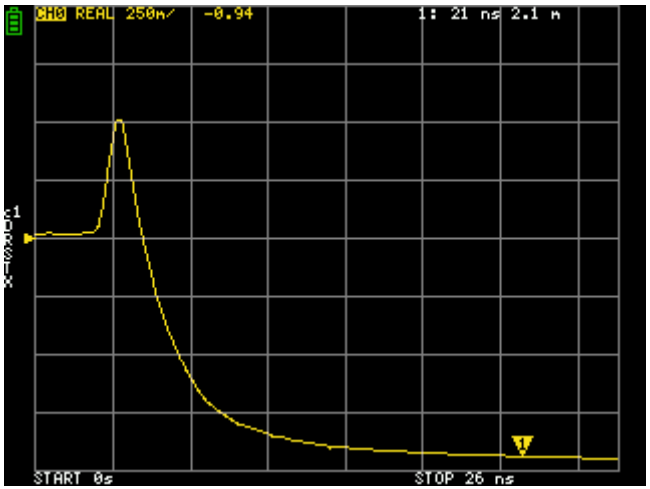
short:



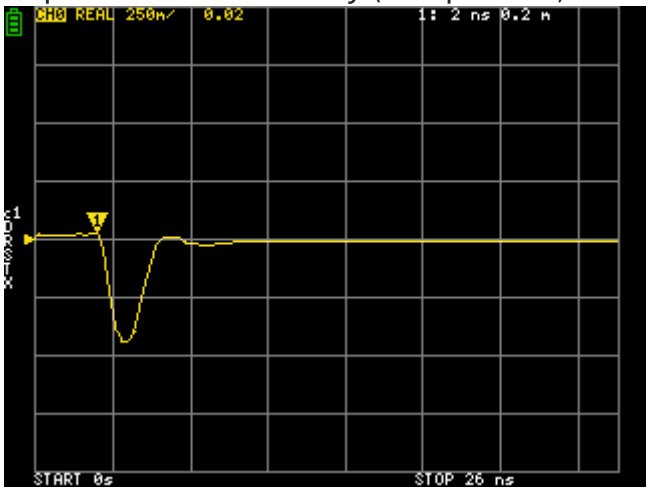

### Step response example

Capacitive short:

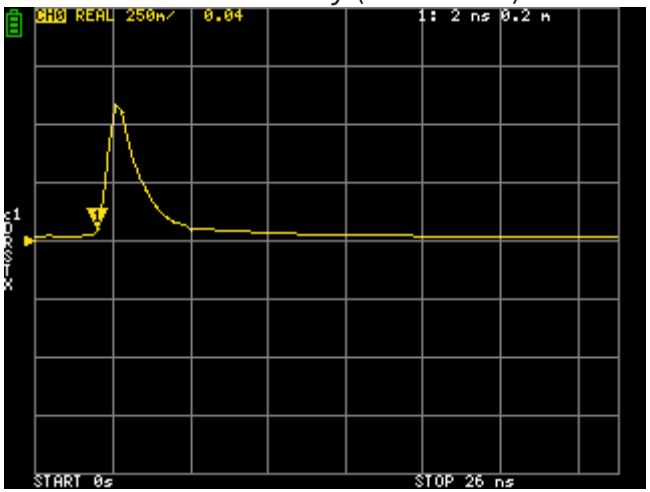
Inductive short:



Capacitive discontinuity (C in parallel):




Inductive discontinuity (L in series):




## Time domain window

The measurable range is a finite number, and there are minimum and maximum frequencies. Windows can be used to smooth this discontinuous measurement data and reduce ringing.

The window has three stages.

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

MINIMUM has the highest resolution, MAXIMUM has the highest dynamic range. NORMAL is somewhere in between.

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

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

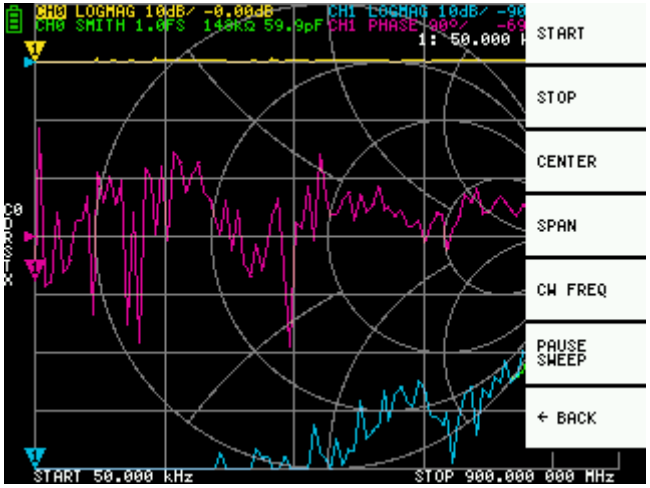
In the time domain, the displayed time can be displayed in terms of distance. The wavelength reduction rate used for distance display `DISPLAY TRANSFORM VELOCITY FACTOR` can be set with. For example, if you measure the cable of TDR with a wavelength reduction rate of 67%, `VELOCITY FACTOR` in `67` the specified.

### Set frequency from marker

You can set the frequency range from the marker as shown below.

- `MARKER →START` Set the frequency of the active marker to the start frequency
- `MARKER →STOP` Set the frequency of the active marker to the stop frequency
- `MARKER →CENTER` Set the frequency of the active marker to the center frequency. The span is adjusted to keep the current range as close as possible.
- `MARKER →SPAN` Sets the two visible markers to the span, including the active marker. If only one marker is displayed, nothing happens.

## Setting the measurement range




There are three types of measurement range settings.

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

## Setting the start frequency and stop frequency

Each `STIMULUS START` , `STIMULUS STOP` and selecting and setting.

## Set center frequency/span

Each `STIMULUS CENTER` , `STIMULUS SPAN` and selecting and setting.

## Zero span

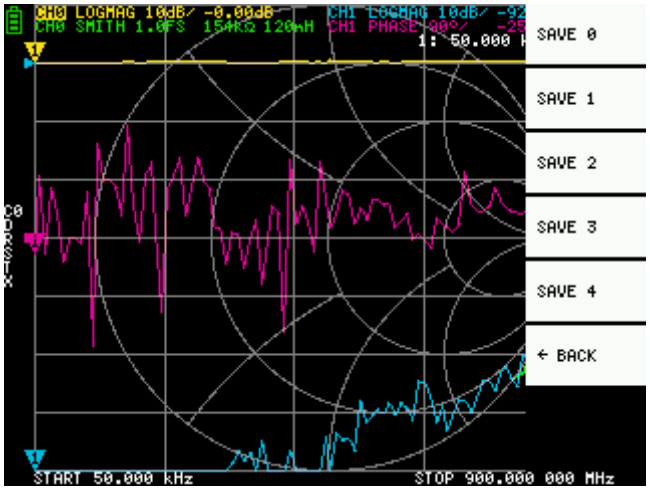
Zero span is a mode in which one frequency is sent continuously without frequency sweep.

`STIMULUS CW FREQ` Select and set.

### Temporarily stop measurement

`STIMULUS` `PAUSE SWEEP` Select to temporarily stop the measurement.

## Calibration and setup recall




You can save up to 5 calibration data. The NanoVNA loads the data of number 0 immediately after starting.

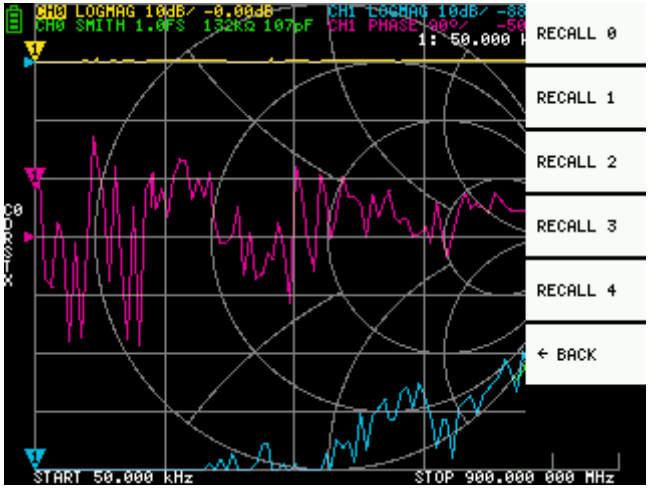
Calibration data is data that includes the following information.

- Frequency setting range
- Error correction at each measurement point
- Trace setting status
- Marker setting status
- Domain Mode Settings
- Setting wavelength reduction rate
- electrical delay

`CAL` `SAVE` `SAVE n` You can save the current settings by selecting.

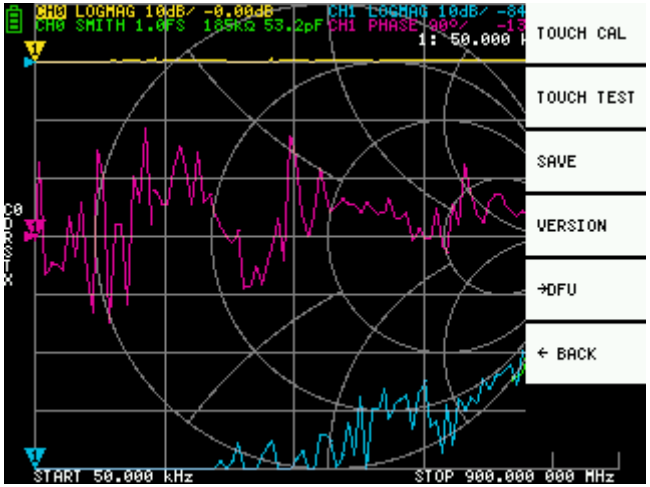
`CAL` `RESET` You can reset the current calibration data by selecting. If you `RESET` want to recalibrate, you must do.

`CAL` `CORRECTION` Indicates whether error correction is currently being performed. You can select this to temporarily stop error correction.



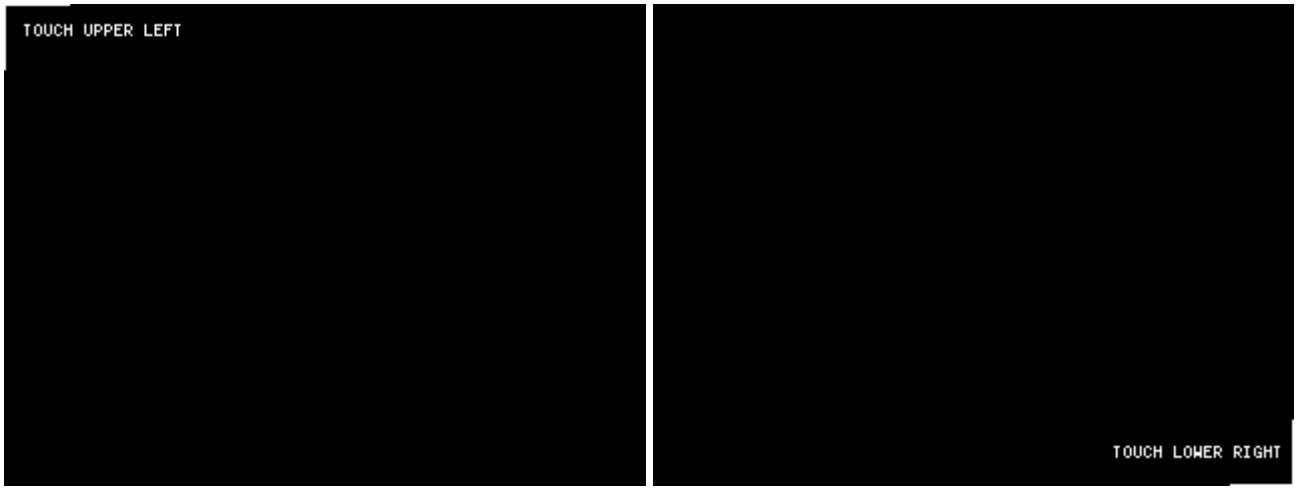

`RECALL` `RECALL n` You can recall the saved settings by selecting.

## Equipment settings

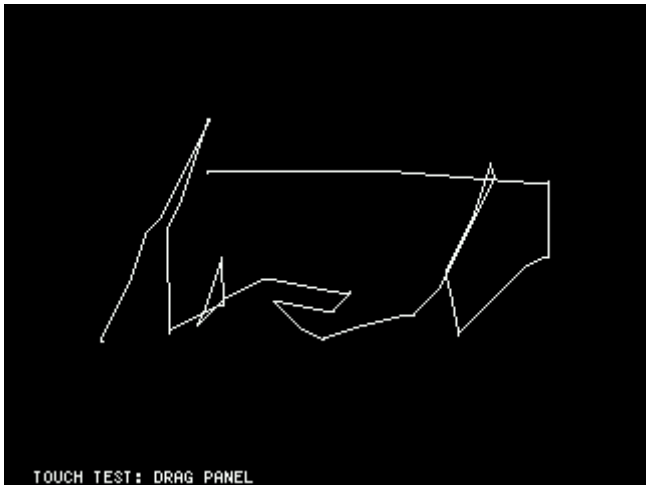



CONFIG Below you can make general settings for the device.

## Touch panel calibration and testing




CONFIG TOUCH CAL Select to calibrate the touch panel. If there is a large difference between the actual tap position and the recognized tap position, this can be solved. TOUCH CAL After TOUCH TEST performing, confirm that the SAVE settings are correct by performing , and save the settings with.




`CONFIG` `TOUCH TEST` Touch to test the touch panel. A line is drawn while tapping the touch panel. When you release it from the touch panel, it returns to the original state.

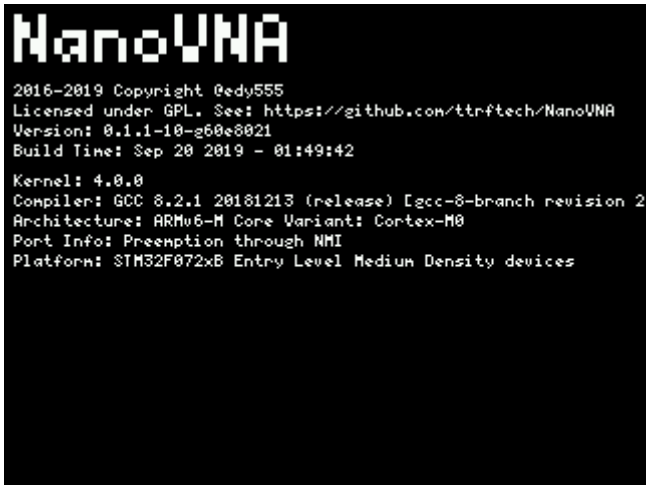
## Save instrument settings

`CONFIG` `SAVE` Select to save the general settings of the device. General device settings are data that includes the following information:

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

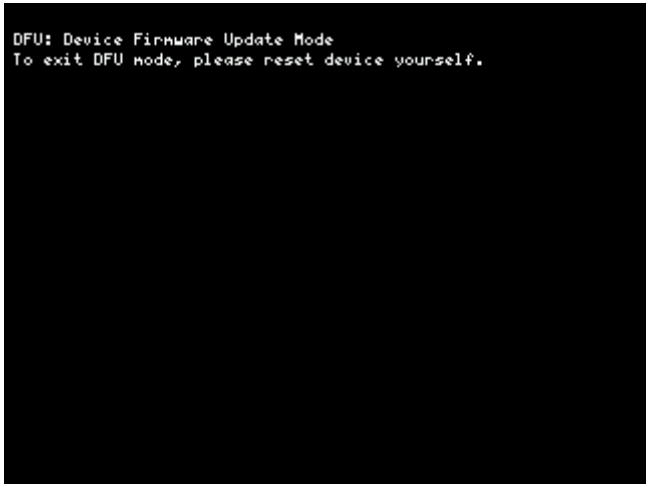
There is currently no way to set anything other than touch panel calibration information.

## Show version




`CONFIG VERSION` Select to display the version information of the device.

## Firmware update




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

# How to update the firmware

## How to get the firmware

### ttrftech version firmware

Original firmware. It is versioned and is frequently developed.

- GitHub releases
- CircleCI build

GitHub releases has a reasonably stable release version of the firmware.

CircleCI has all the firmware per commit. Use this if you want to try the latest features or check for defects.

### hugen79 version firmware

- Google Drive

The latest firmware is installed in Google Drive.

### Build it yourself

You can easily clone the github repository and build it yourself.

## How to write the firmware

There are various ways to write it, but here I will explain using dfu-util . dfu-util is a cross-platform tool, and binaries are also provided for Windows, so you can download and use it.

## Writing with dfu-util (Ubuntu)

You can find dfu-util in the standard package repository.

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

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

- Turn on the power by jumpering the BOOT0 pin on the PCB. (After turning on the power, remove the jumper.) The screen becomes white but it is normal.
- `CONFIG →DFU RESET AND ENTER DFU` Select

Execute the following command. build/ch.bin describes the path to the .bin of the downloaded firmware file.

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

## Writing with dfu-util (macOS)

We recommend using homebrew to install.

Install the brew command.

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

Install the dfu-util command.

```
brew install dfu-util
```

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

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

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

- Turn on the power by jumpering the BOOT0 pin on the PCB. (After turning on the power, remove the jumper.) The screen becomes white but it is normal.
- `CONFIG →DFU RESET AND ENTER DFU` Select

Execute the following command. build/ch.bin describes the path to the .bin of the downloaded firmware file.

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

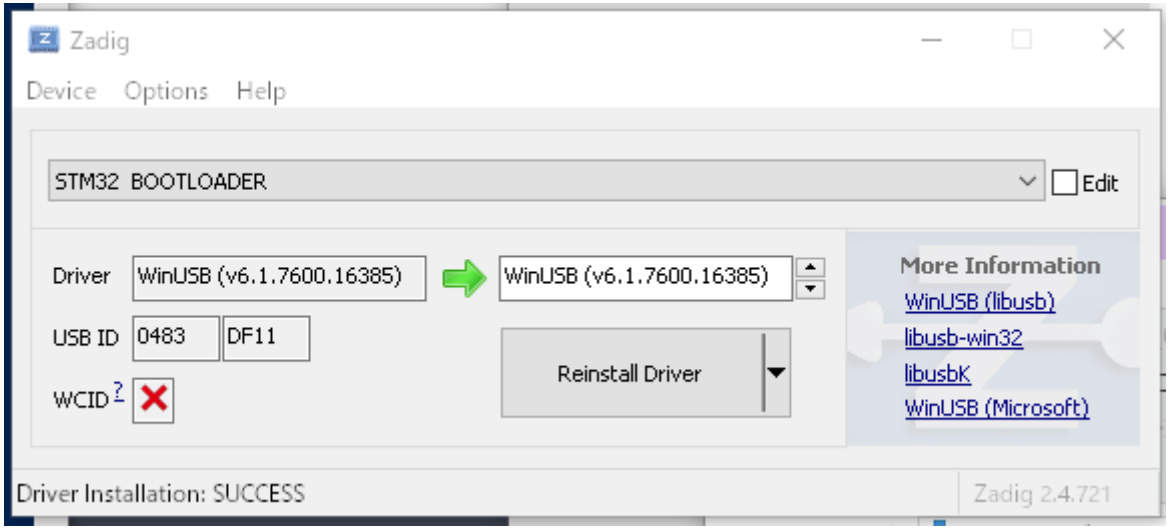
## Writing with dfu-util (Windows 10)

For Windows, connecting the NanoVNA in DFU mode will automatically install the device driver, but this device driver cannot use dfu-util. Here we use Zadig to replace the driver.

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

- Turn on the power by jumpering the BOOT0 pin on the PCB. (After turning on the power, remove the jumper.) The screen becomes white but it is normal.
- `CONFIG →DFU RESET AND ENTER DFU` Select

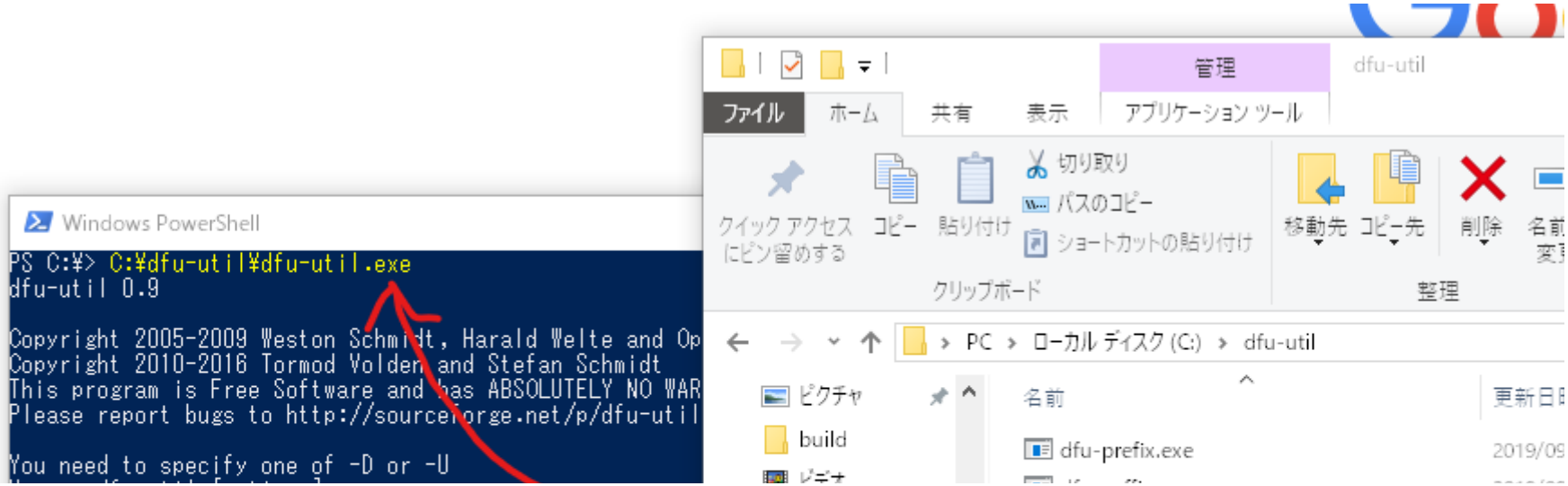
Start Zadig with NanoVNA in DFU mode connected, and use WinUSB as a driver for STM32 BOOTLOADER as follows.

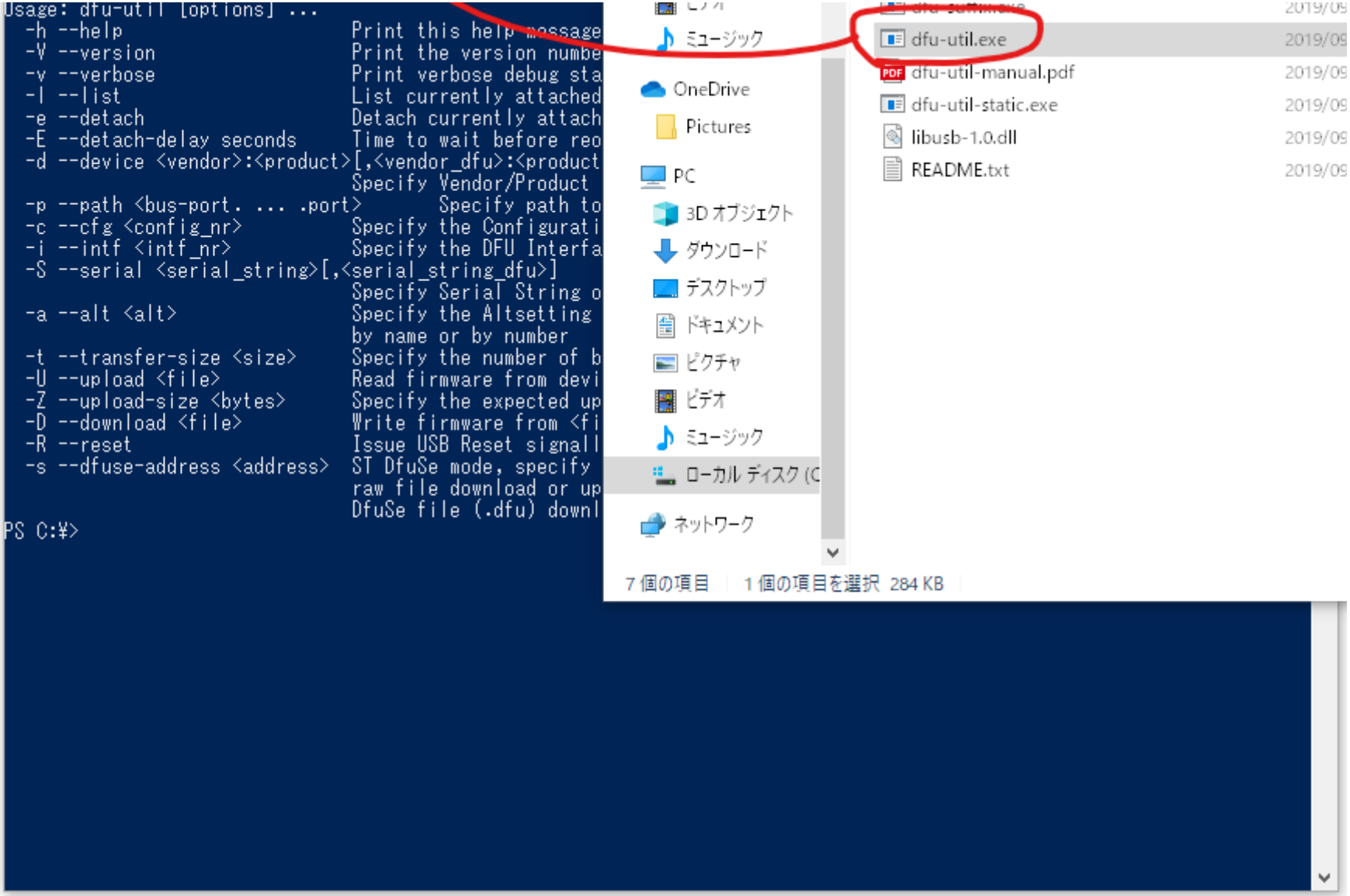



* If you want to restore the driver, find the corresponding device in "Universal Serial Bus controllers" of "Device Manager" and execute "Uninstall device". The driver will be installed automatically when the USB connector is removed and reinserted.

Next, place dfu-util. Download and extract dfu-util-0.9-win64.zip from releases . As an example, I'll assume that you have expanded it to C:\dfu-util (it can be anywhere).

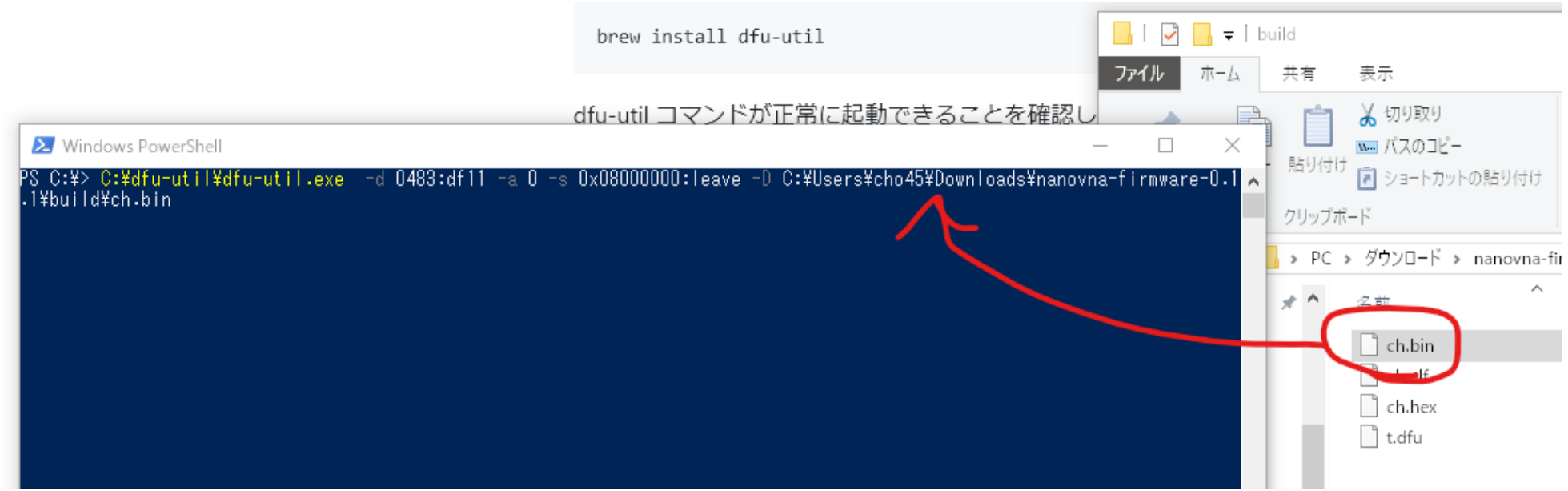
Right-click on the Start Menu and select Windows PowerShell. The shell screen opens.






Drag and drop dfu-util.exe from Explorer to PowerShell to insert the path automatically. The following so `--version` you can start to when the version display of dfu-util wearing.

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




Similarly, the path of the firmware file can be input by dragging and dropping it from Explorer to PowerShell.

Execute the following command. build/ch.bin describes the path to the .bin of the downloaded firmware file.

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

## How to write the firmware (Windows GUI)

For those who are not familiar with CUI, some complicated steps are required, but I will also introduce the writing method using the DfuSE Demo tool provided by ST for reference.

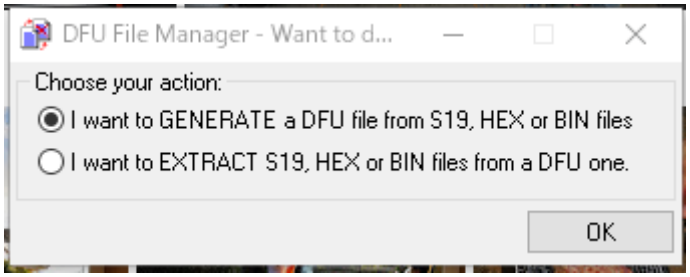
Download STSW-STM32080 from ST site .

- DFU File Manager: Tool to create .dfu files from .bin or .hex
- DfuSe Demo: A tool to write .dfu files to your device

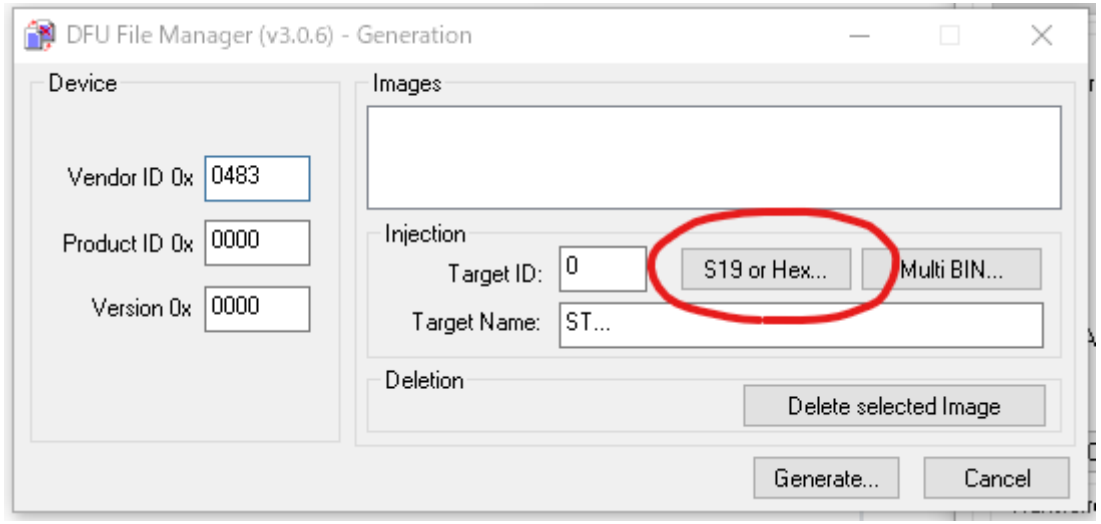
is included.

## Convert the file format with DFU File Manager.

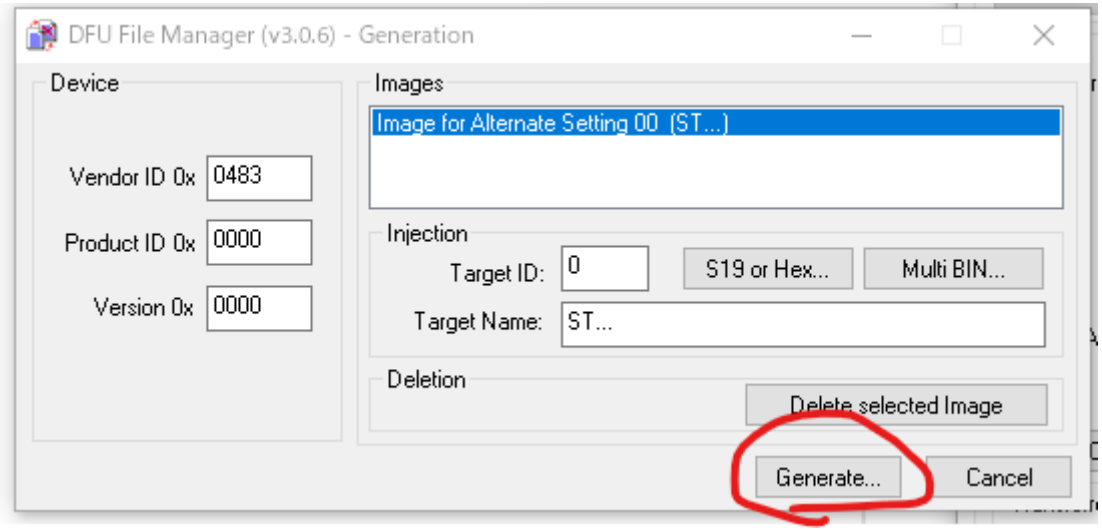
First, start DFU File Manager.




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




`S19 or Hex...` Click the button. `ch.hex` Select the firmware .hex file, etc.



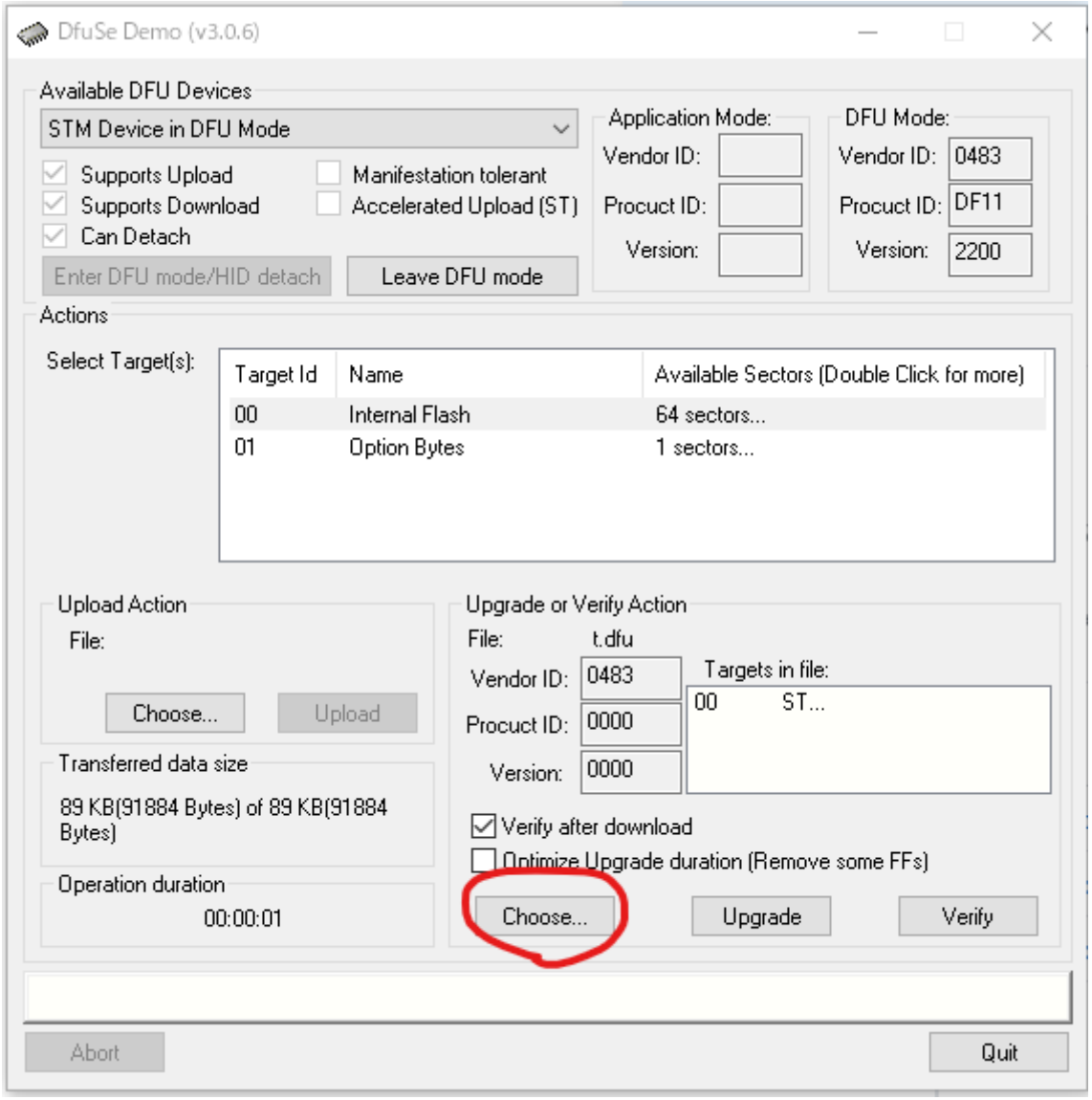

`Generate...` Click the button and create a .dfu file with a suitable name.

## Writing firmware with DfuSe Demo

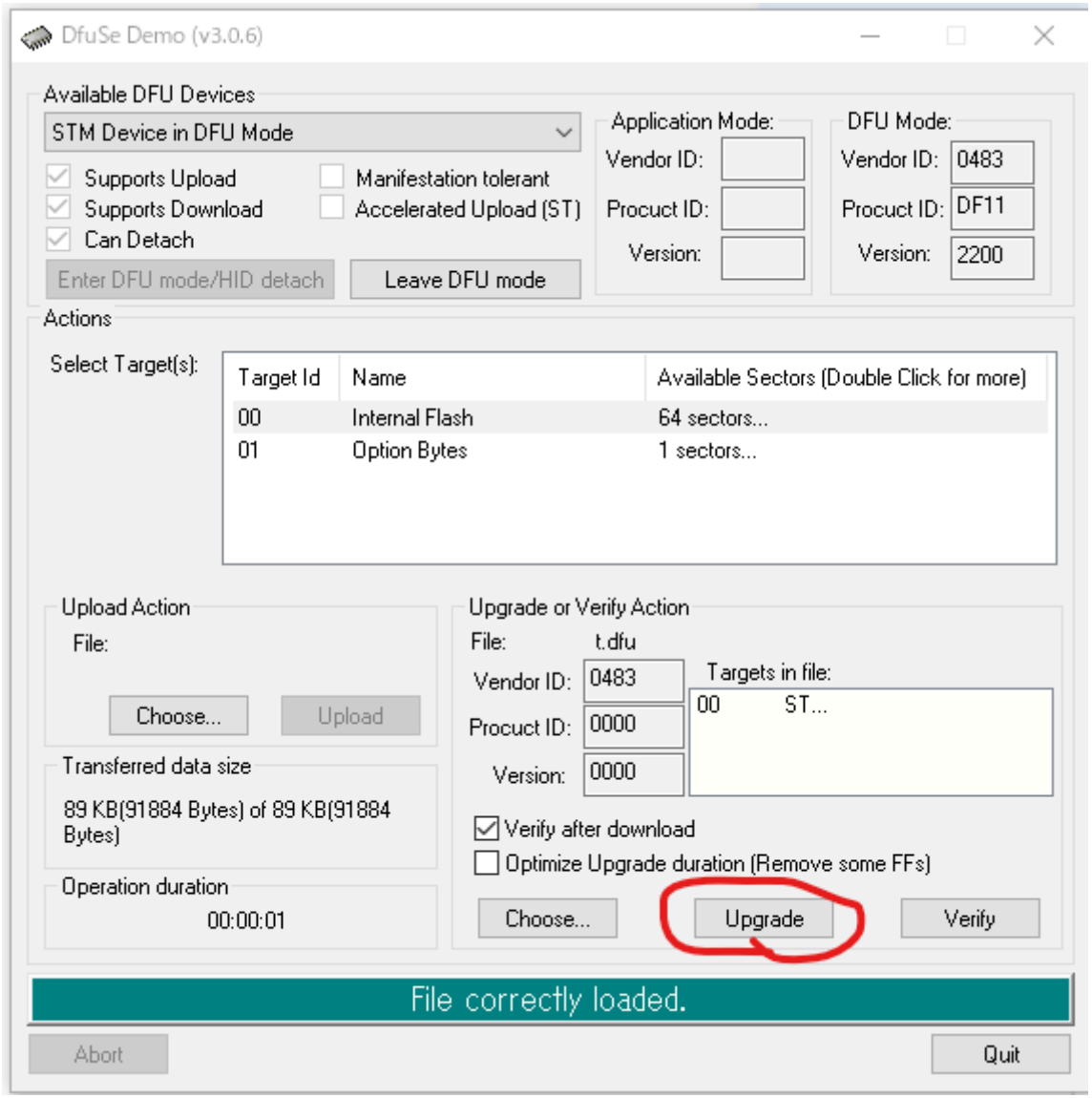
First boot the device in DFU mode. Use one of the following methods to enter DFU mode.

- Turn on the power by jumpering the BOOT0 pin on the PCB. (After turning on the power, remove the jumper.) The screen becomes white but it is normal.
- `CONFIG →DFU RESET AND ENTER DFU` Select

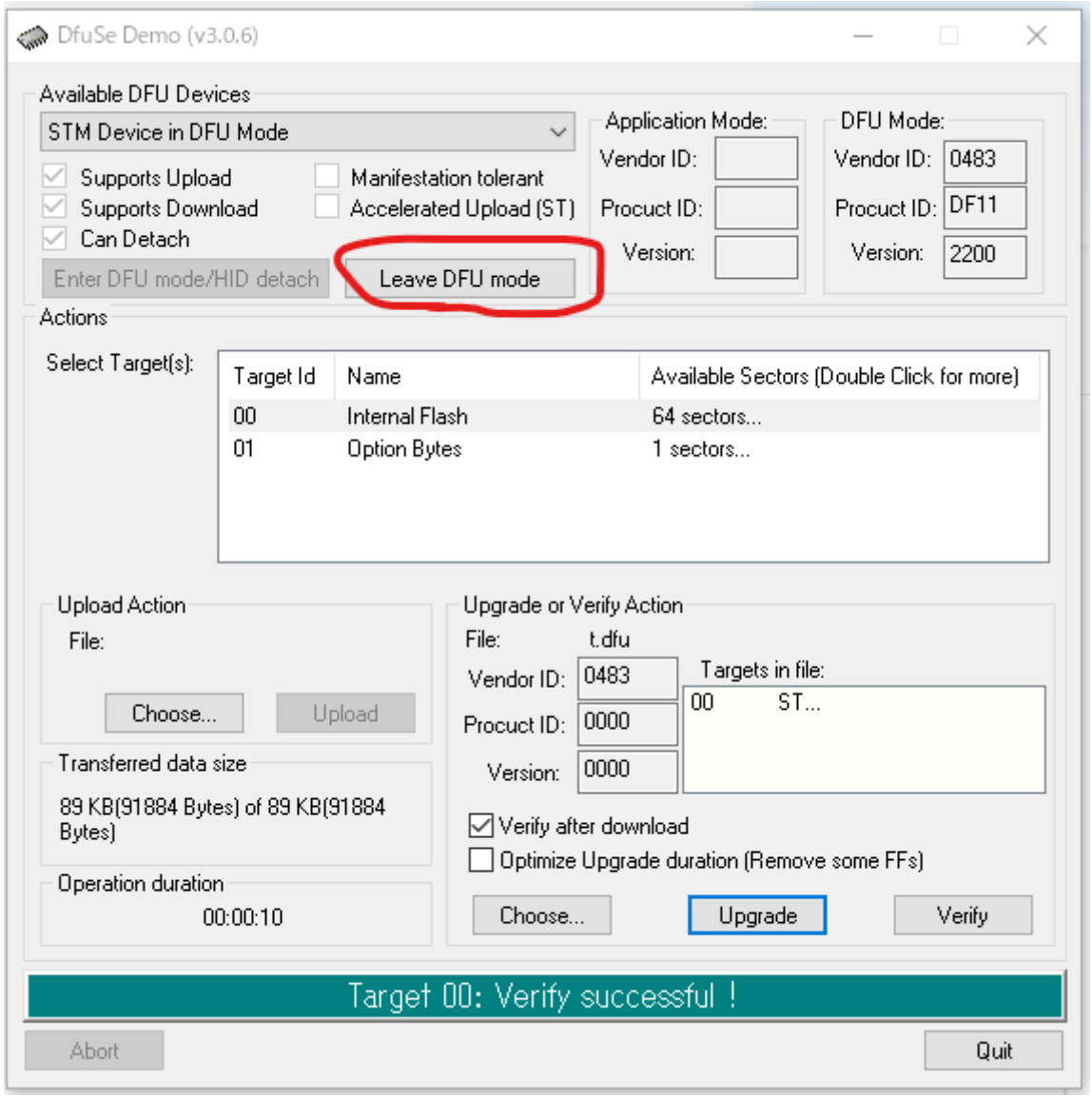
Launch the DfuSe Demo. Make `STM Device in DFU Mode` sure that it is in Available DFU Devices `Choose...` and click.




Select the .dfu file you saved earlier.




`Upgrade` Click the button.




After writing, this screen will `Leave DFU mode` appear. Click the button to exit DFU mode. The device will reset and boot with the new firmware.

# Firmware Development Guide

The following are the development requirements for the NanoVNA firmware.

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

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

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

## Build with Docker

You can use docker to build without any hassles. docker is a free, cross-platform container utility. It can be used to quickly reproduce a particular environment (in this case, the build environment).

Simply install docker and then 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 (VSCode) is a multi-platform code editor provided by Microsoft for free. On-chip debugging can be done by GUI by introducing Cortex-Debug Extension.

I will omit the platform-dependent part, but in addition to the above, the following is required.

- openocd
- VSCode
- Cortex-Debug

Cortex-Debug is searched from VSCode Extensions 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 do make as a task on VSCode.

## launch.json

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

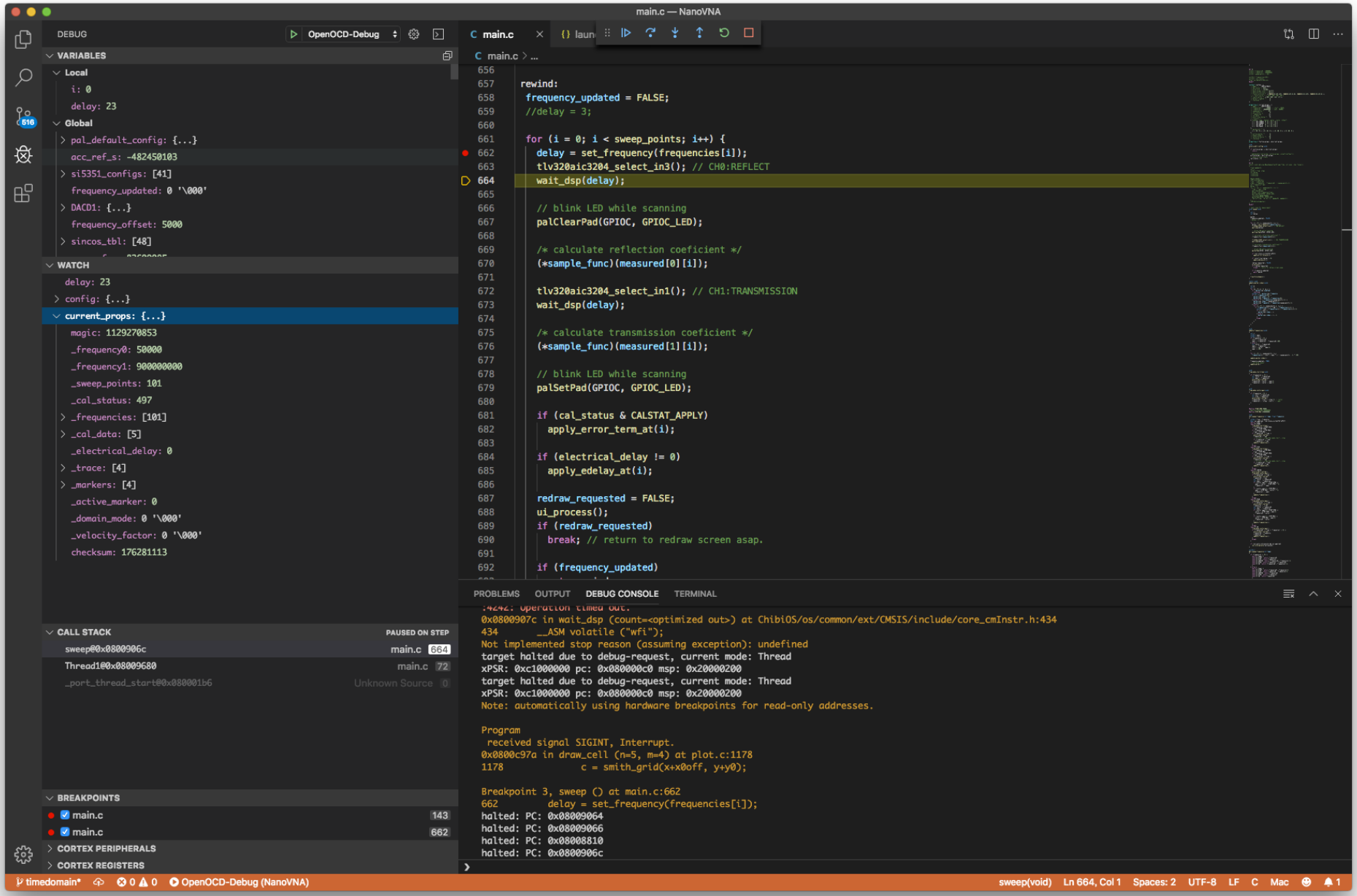
The following is the setting when using ST-Link. If If you are using the J-Link `interface/stlink.cfg` wo `interface/jlink.cfg` replace it with.

```
{
    "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",
        }
    ]
}
```

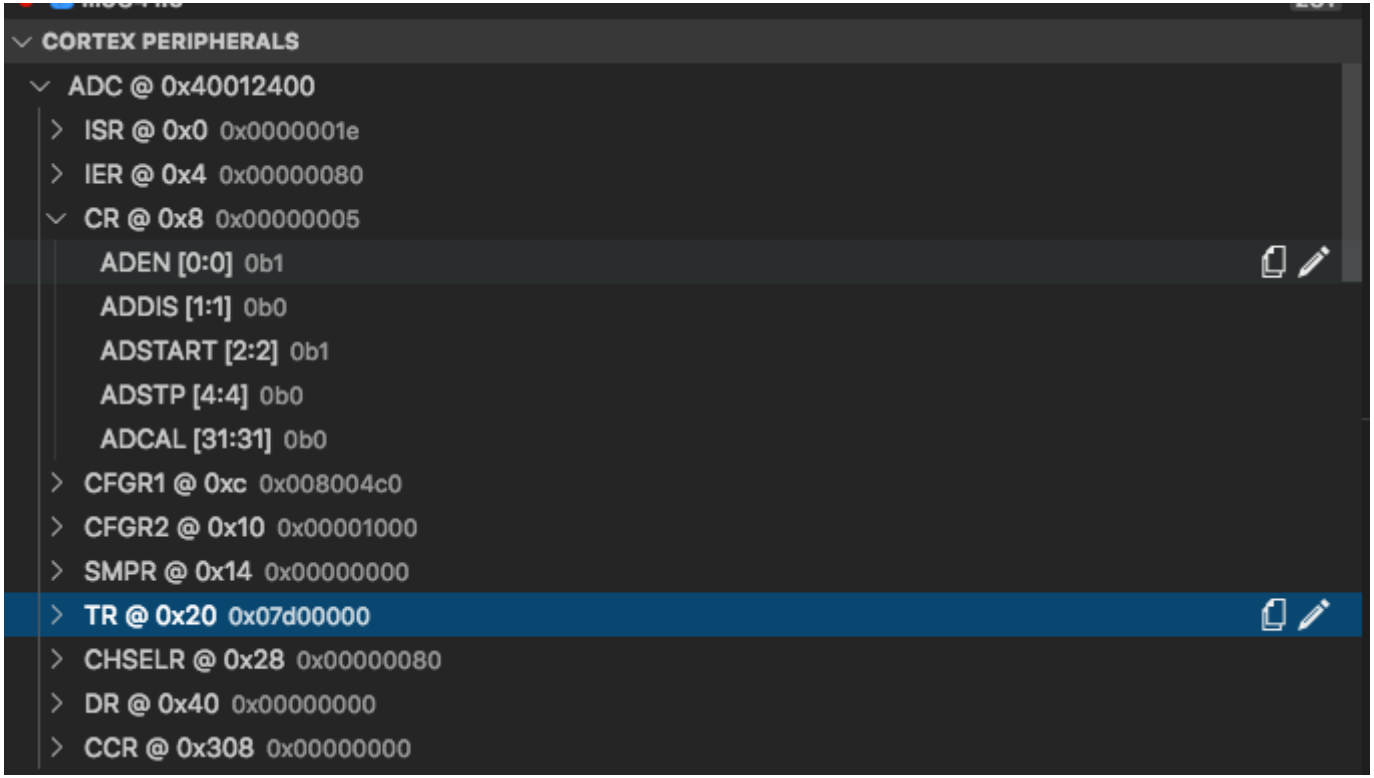
`svdFile` The file specified for can be downloaded from the ST site . `svdFile` There is no problem in operation even if is not specified.

## Start debugging

If you do Start Debugging ( F5 ), OpenOCD will automatically start and the firmware will be transferred after the build by make. When the transfer is completed, the reset handler breaks.




`svdFile` If is specified, the defined MCU registers will be displayed in the debug screen.

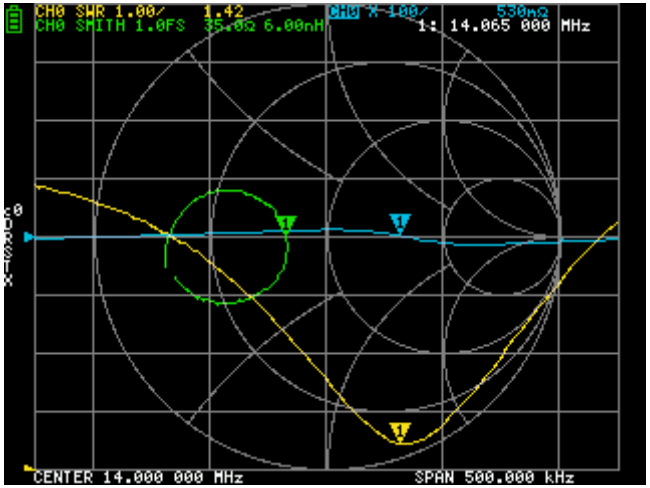

CORTEX PERIPHERALS
ADC @ 0x40012400
ISR @ 0x0 0x0000001e
IER @ 0x4 0x00000080
CR @ 0x8 0x00000005
ADEN [0:0] 0b1
ADDIS [1:1] 0b0
ADSTART [2:2] 0b1
ADSTP [4:4] 0b0
ADCAL [31:31] 0b0
CFGR1 @ 0xc 0x008004c0
CFGR2 @ 0x10 0x00001000
SMPR @ 0x14 0x00000000
TR @ 0x20 0x07d00000
CHSELR @ 0x28 0x00000080
DR @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

# Example of use

## Bandpass filter adjustment

TODO

## Antenna adjustment




Here is an example of using NanoVNA as an antenna analyzer.

The following two points are important in adjusting the antenna.

- Whether the antenna is in a tuned/resonant state (that is, the reactance is close to 0 at the desired frequency)
- Low SWR of antenna (good matching)

### Trace settings

Only CH0 is used for antenna adjustment, so calibrate all items except `THRU` and `ISOLN` .

The trace settings are as follows.

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

`CENTER` Set the frequency you want to tune the antenna to, `SPAN` and set appropriately.

Look for frequencies where trace 1 displaying the reactance is near 0. Since the frequency is the tuning point, if there is a deviation, adjust the antenna so that the tuning point comes to the target frequency.

If the tuning point is at the desired frequency, make sure that trace 0, which is displaying SWR, is displaying SWR that is low enough (close to 1). If it does not show enough SWR (less than or equal to 2), use Smith chart to match. At this time, you can use an antenna tuner directly below the antenna for matching.

If the SWR goes down, you're tuned to the frequency you want, and you're done with the low SWR antenna.

## Cable check

You can simulate the TDR by using the time domain lowpass mode. TDR can be used to find defects in transmission lines.

TODO

## Common mode filter measurement

TODO