ttrftech / **NanoVNA**

Very Tiny Palmtop Vector Network Analyzer

211 commits | **1** branch | **6** releases | **2** contribu

Branch: **master** ▾ | **New pull request** | **Find File** | **Clone**

edy555 add command battery | Latest commit be456

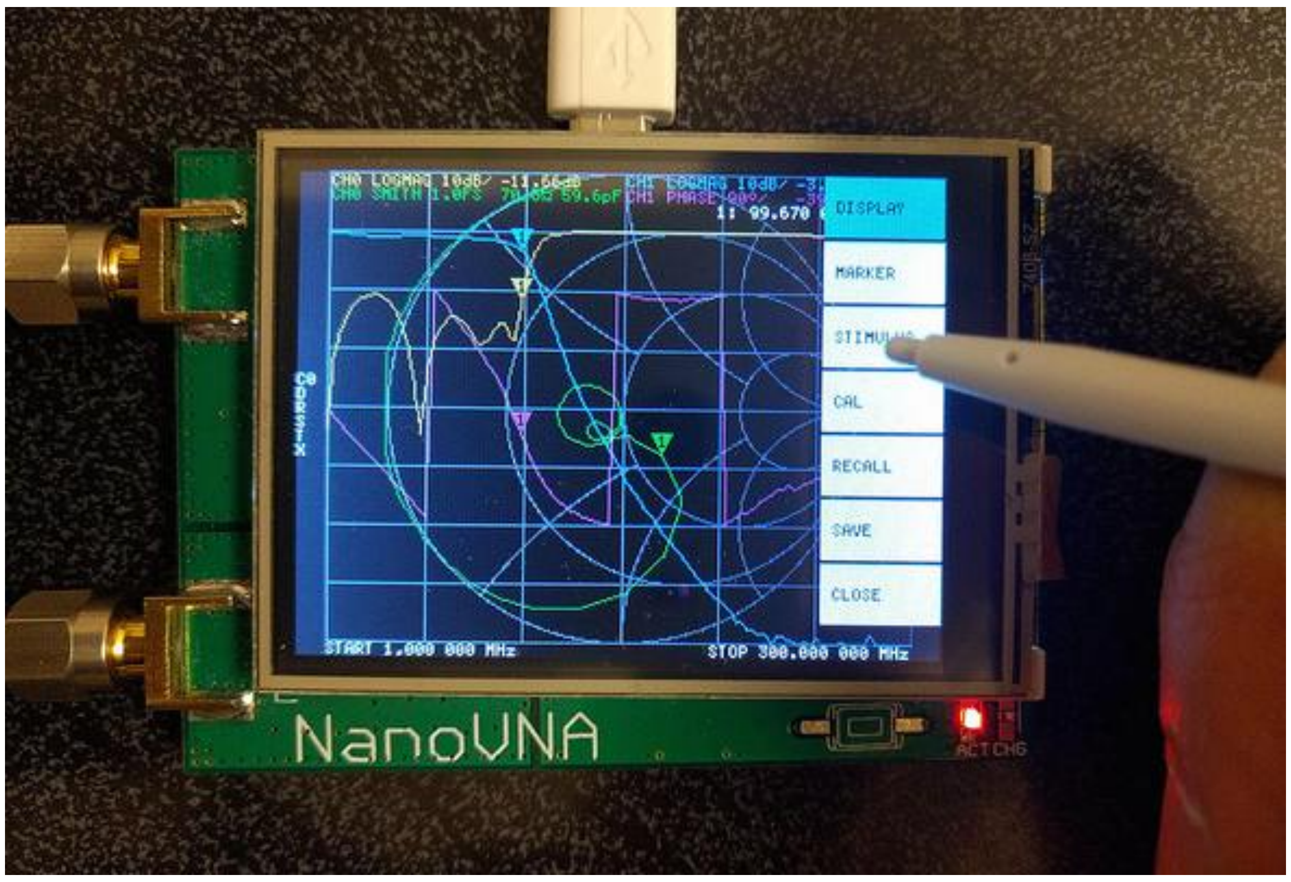
.circleci	ci: remove CIRCLE_TAG from build job
.vscode	env: add vscode configulation
ChibiOS @ 669d4bb	revert chibios fix
NANOVNA_STM32_F072	move `enter dfu` to submenu of config
doc	add schematics, block diagram and pcb photo, update README.md
python	add time domain reflectmetry
.gdbinit	add hard_fault handler
.gitignore	env: update .gitignore
.gitmodules	fix: correct branch of submodule ChibiOS
Font5x7.c	view: add left arrow on menu items
Makefile	add target dfu in Makefile
README.md	doc: update README
STM32F072xB.ld	add save and restore dac value in flash
adc.c	show battery indicator (requires implementing D2 on board)
chconf.h	static inline reduce stack memory usage
dsp.c	gain adjust for frequency
ffconf.h	initial commit
flash.c	add interpolation on cal
halconf.h	change adc driver not to use chibios hal
ili9341.c	append version info screen to UI
main.c	add command battery
mcuconf.h	change adc driver not to use chibios hal
nanovna.h	add command battery
numfont20x24.c	feat: add 'n', 'p' and pad symbol
plot.c	show battery indicator (requires implementing D2 on board)

prog.sh	initial commit
si5351.c	feat: adjust gain and frequency band, enpower LO drive strength.
si5351.h	add argument rdiv in si5351 setupMultisynth
tlv320aic3204.c	fix: omit adc filter
ui.c	move `enter dfu` to submenu of config
usbcfg.c	initial commit
usbcfg.h	initial commit

README.md

NanoVNA - Very tiny handheld Vector Network Analyzer

release v0.0.9 circleci passing



About

NanoVNA is very tiny handheld Vector Network Analyzer (VNA). It is standalone with lcd display, portable device wit battery. This project aim to provide an RF gadget but useful instrument for enthusiast.

This repository contains source of NanoVNA firmware.

Prepare ARM Cross Tools

Requires gcc-4.9 to build firmware from source code. (Not work gcc-5.4 or lator, because of SRAM shortage that th runtime use more SRAM)

MacOSX

Install cross tools and firmware updating tool.

```
$ brew tap px4/px4
$ brew install gcc-arm-none-eabi-49
$ brew install dfu-util
```

Otherwise, use toolchains included inside LPCxpresso. Like this.

```
$ PATH=$PATH:/Applications/lpcxpresso_7.8.0_426/lpcxpresso/tools/bin
```

Linux (ubuntu)

Download arm cross tools from here. This version is 32-bit binary, so additional lib32z1 and lib32ncurses5 package required.

```
$ wget https://launchpad.net/gcc-arm-embedded/4.9/4.9-2015-q3-update/+download/gcc-arm-none-eabi-
4_9-2015q3-20150921-linux.tar.bz2
$ sudo tar xfj -C /usr/local gcc-arm-none-eabi-4_9-2015q3-20150921-linux.tar.bz2
$ PATH=/usr/local/gcc-arm-none-eabi-4_9-2015q3/bin:$PATH
$ sudo apt install -y lib32z1 lib32ncurses5
$ sudo apt install -y dfu-util
```

Fetch source code

Fetch source and submodule.

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

Build

Just make in the directory.

```
$ make
```

Build firmware using docker

You can build firmware using this docker image without installing arm toolchain.

```
$ cd NanoVNA
$ docker run -it --rm -v $(PWD):/work edy555/arm-embedded:4.9 make
```

Flash firmware

Boot MCU in DFU mode. To do this, jumper BOOT0 pin at powering device. Then, burn firmware using dfu-util via US

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

Or do simply

```
$ make flash
```

Control from PC

See python directory.

Note

Hardware design material is disclosed to prevent bad quality clone. Please let me know if you would have your own u

Reference

- Schematics
- PCB Photo
- Block Diagram
- Kit available from https://ttrf.tk/kit/nanovna
- Credit: @edy555

[EOF]