jankae / **LibreVNA** Public

100kHz to 6GHz 2 port USB based VNA

GPL-3.0 license

447 stars 84 forks

Star Notifications

Code Issues 6 Pull requests 6 Actions Projects Wiki Security

master Go to file

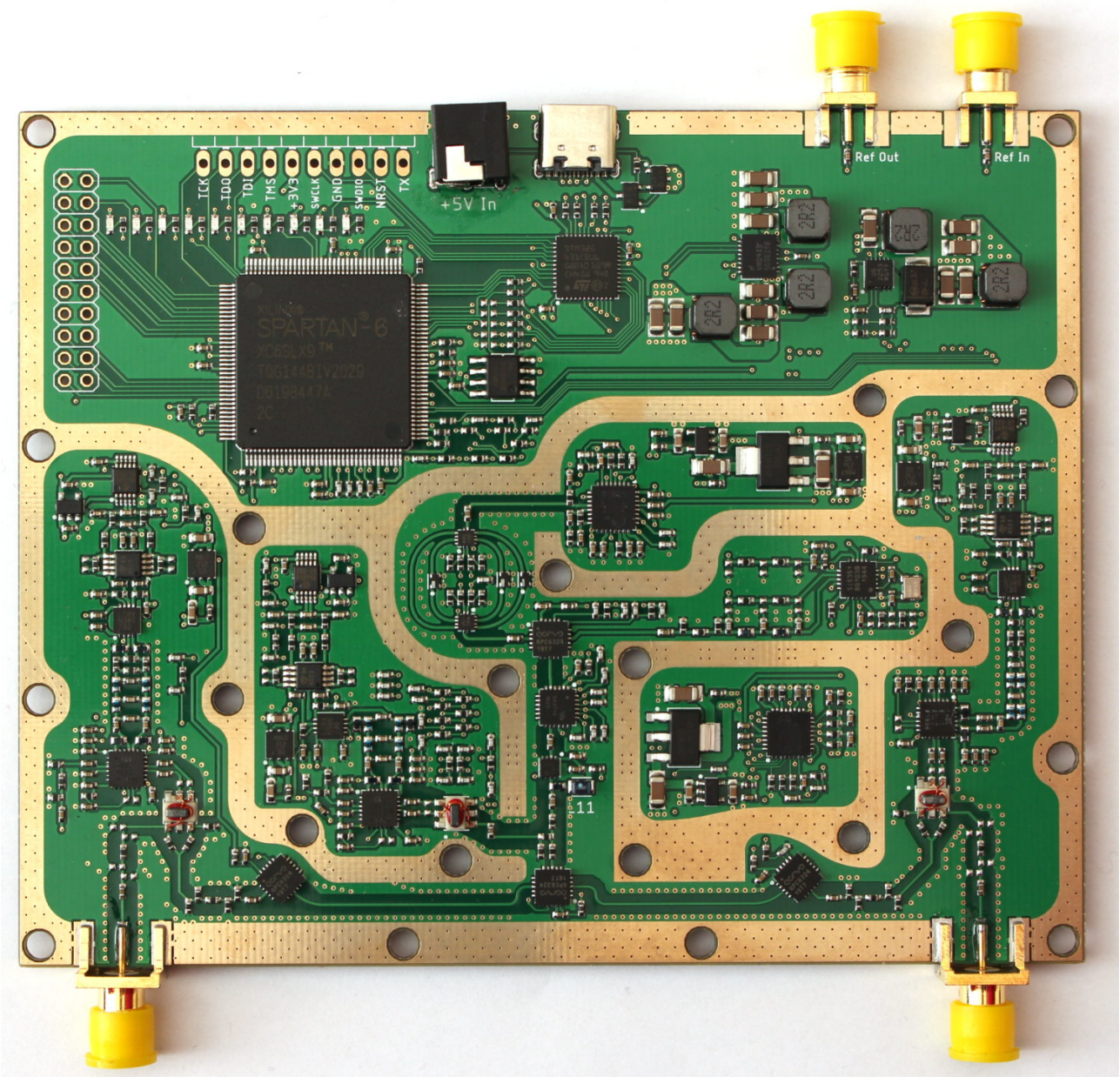
jankae Merge pull request #122 from sophiekovalevsky/implement-sc... ... 4 days ago 514

View code

README.md

100kHz to 6GHz VNA

This is the improved version of my first attempt at a VNA.



GUI Installation

Windows

- Download the latest Release and unpack the zip file
- Start the included LibreVNA-GUI.exe
- No installation or driver is required, simply move the extracted folder somewhere convenient

Ubuntu

- Download the latest Release and unpack the zip file
- Install the required libraries:

```
sudo apt install libqt5widgets5
```

- Install the udev rule (otherwise you don't have the permissions to access the USB device):

```
wget https://raw.githubusercontent.com/jankae/LibreVNA/master/Software/PC_Applicatic
sudo cp 51-vna.rules /etc/udev/rules.d
```

- Either reboot or reload the udev rules manually:

```
sudo udevadm control --reload-rules
sudo udevadm trigger
```

- You can now start GUI:

```
cd $UNPACKED_ZIP_FOLDER$
./LibreVNA-GUI
```

Quick Start

- You can find released versions of the GUI application and the device firmware here.
- If you would like to try out the newest features, the compiled versions of each commit can be found here (but keep in mind that some features might be unstable or incomplete).
- An (incomplete) user manual is also available.
- If you would like to control the LibreVNA with a script, see the SCPI Programming Guide.

Please also take a look at the FAQ.

If you notice bugs or have ideas for improvements, please create an issue for that.

For general questions or discussions, the LibreVNA group is probably the best place.

Preliminary specifications

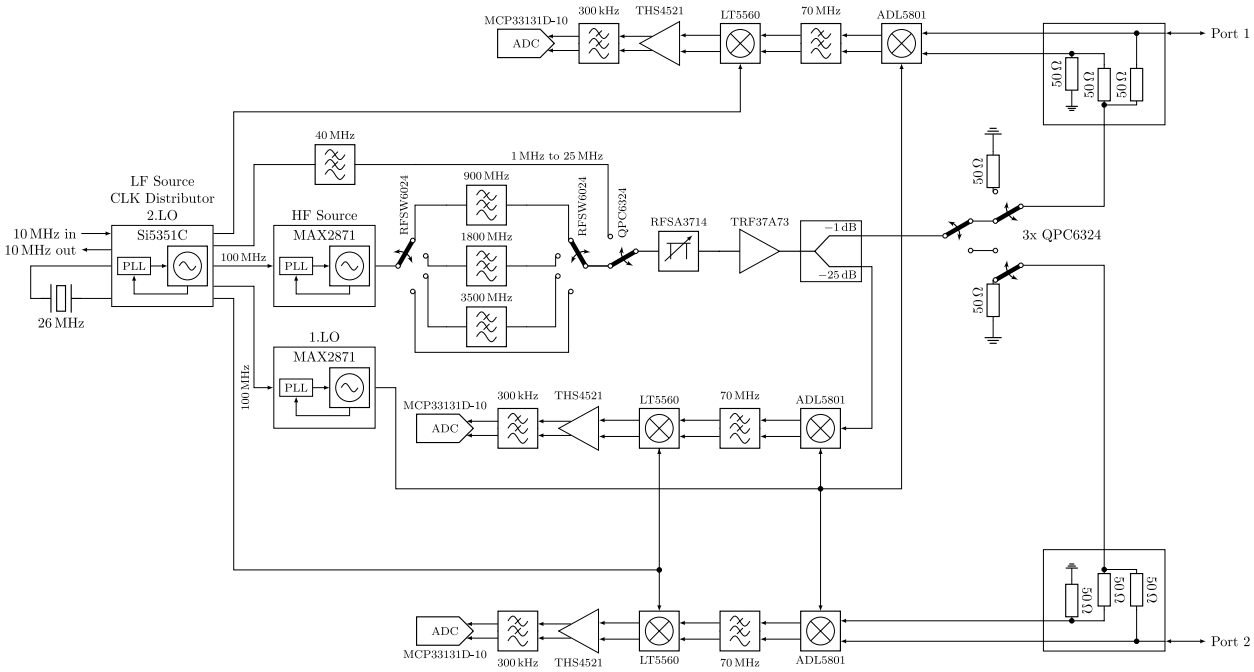
Some specifications are available but this project is still under development and the specifications might change.

There are also some initial example measurements.

How does it work?

The PCB is really only the RF frontend with some processing power. Everything else is handled in the PC application once the data is transferred via USB. You can try out the application without the PCB (obviously no measurements are possible, but you can import provided example measurements and get an idea about what it can and can't do).

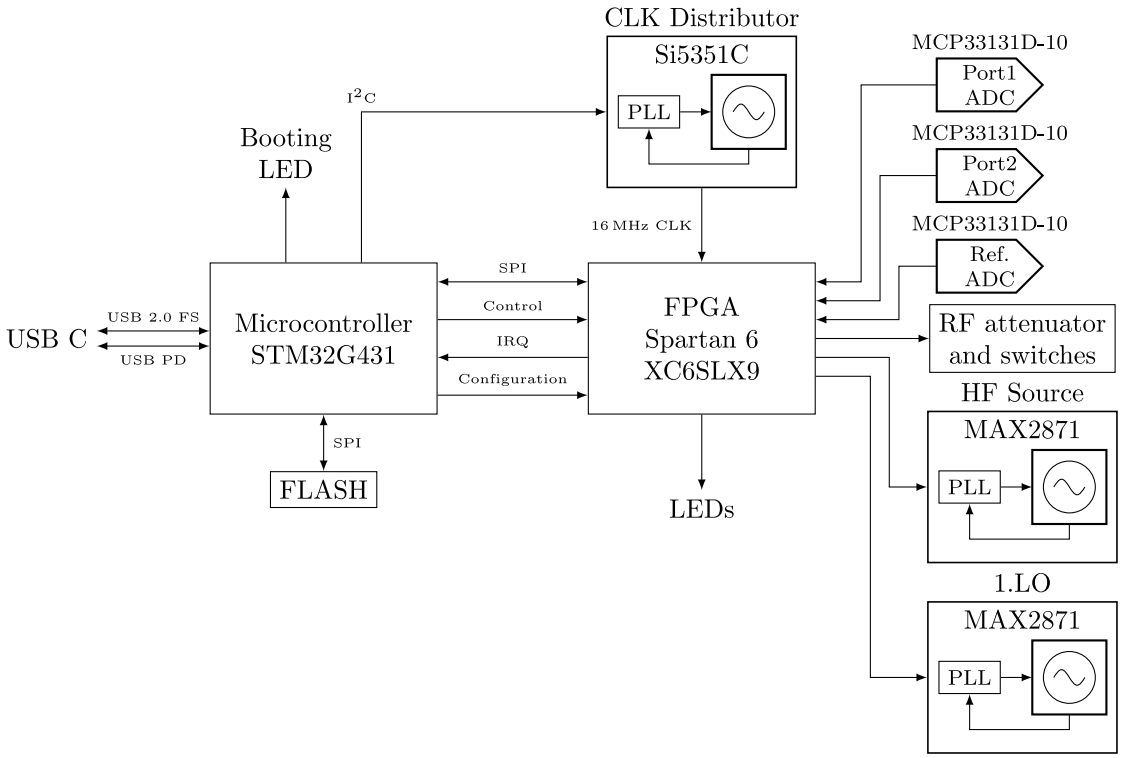
RF path:



- The main clock source is an Si5351C, providing all the required clocks for the different blocks. It also serves as the stimulus source for frequencies below 25MHz. Its own reference clock is either a 26MHz crystal or an external 10MHz signal.
- The stimulus source for frequencies above 25MHz is a MAX2871. Its output signal is slightly filtered to reduce the amount of harmonics.
- The stimulus signal power can be adjusted between approximately -42 and -10dbm with a digital attenuator (RFSA3714).
- After the amplifier(TRF37A73) the signal is split and the weaker part of it fed into the reference receiver.
- The stronger part of the signal can be routed to either port. In each signal path, two RF switches are used in series to achieve higher isolation between the ports.
- Instead of directional couplers, resistive return-loss-bridges are used (easier to implement for wide bandwidth).

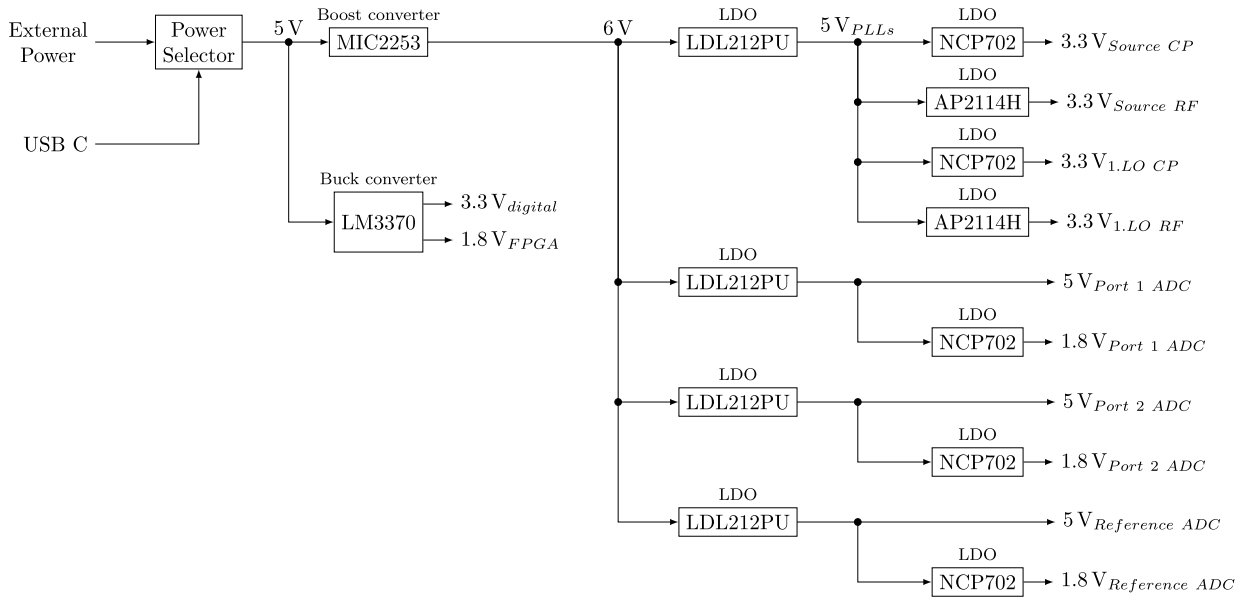
- Both ports have completely separated receive paths. This increases BOM cost but allows measuring two parameters at once (S11 and S21 or S22 and S12). It also avoids potential isolation issues that could arise if the receive paths would merge into a single mixer/ADC.
- Each receiver consists of two down-convert mixers. The 1.IF sits at 60MHz, the 2.IF 250kHz.
- The ADCs are sampling the final IF with 16bit@800kHz.

Digital section:



- The central element is the Spartan6 FPGA. It handles all communication with the RF blocks and samples the ADCs. This allows for nearly instant switching of the measurement frequency, only limited by the settling time of the PLLs.
- The microcontroller handles the setup of the sweep in the FPGA, extracts and preprocesses the measurements and passes them on through USB.
- The flash contains the FPGA bitstream. Because the microcontroller has access to the flash, no FPGA-related hardware tools (such as JTAG programmers) are needed, everything can be updated via USB.

Power supply:



- Everything is powered from USB (or optionally by external 5V DC)
- Almost every RF block has its own local regulater, preventing noise and signals coupling into the supply lines from propagating across the whole PCB

Releases 9

v1.3.0 Latest
on Apr 20

+ 8 releases

Contributors 7

Languages

C 72.3% C++ 26.1% VHDL 1.2% Assembly 0.1% QMake 0.1% Python 0.1% Makefile 0.1%