Home

Jump to bottom

Yury K edited this page on 14 Jun 2020 · 13 revisions

New

The code has been ported to STM32H745I DISCOVERY board, it can be used with the same RF frontend. See the README.md in the separate repo for details.

Introduction

The EU1KY antenna analyzer V3 is an open source project to build your own, reasonably cheap but very functional antenna analyzer that is a handful tool for tuning coax-fed shortwave ham radio HF/VHF, and CB antennas. Big color TFT LCD, full control using capacitive touchscreen, many features outperform many expensive antenna analyzers present on the market. Moreover, you have fun building this tool on your own and save some money.

Features

The device is actually a 1-port VNA, it allows measuring the parameters of any passive load connected to it in the range of 500 kHz ... 150 MHz (and even up to 450 MHz, depending on your hardware, see below), and displaying the results in graphic form. Additionally, it functions as TDR (Time Domain Reflectometer).

Open-Short-Load calibration is supported to compensate transmission line influence on measured parameters.

All the settings, as well as calibration files, screenshots and Touchstone S1P files, are stored on SD card which can be accessed via the on-board USB HS port (it works as card reader).

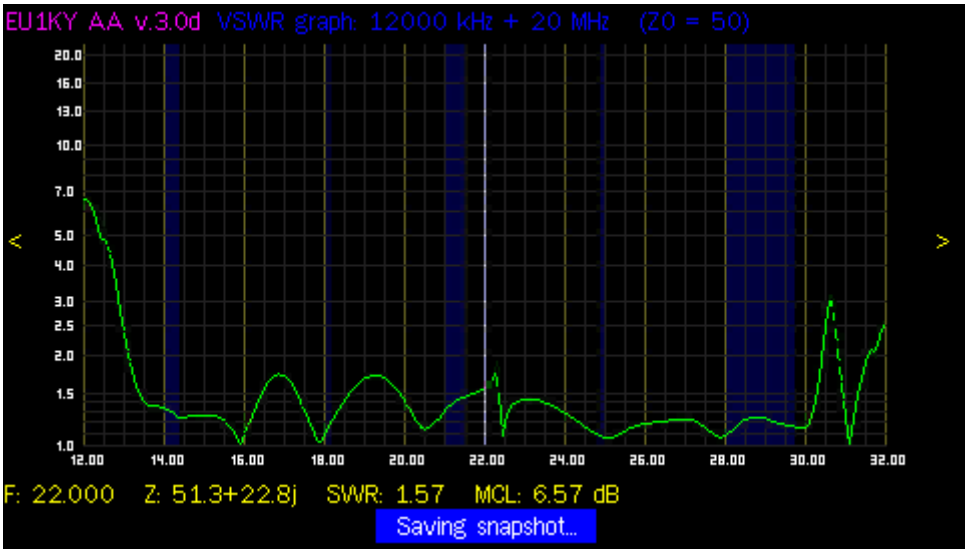
ST-Link 2.1 debbugger built into the dev board provides virtual COM port that, when the device is connected to PC with Mini-USB cable, can be used to control device remotely using RigExpert AntScope software when the device shows main menu window. The device emulates some features of the RigExpert AA-1000 antenna analyzer.

*Si5351 revision B specs dated 2015 declare that the chip can output maximum frequency of **200 MHz** instead of 160 MHz as they stated before. I am not sure that Si5351 chips manufactured before 2015 can output this frequency reliably in the entire operational temperature range, but, I hope, most of them can. Actually, the IC is designed so that 225 MHz is its maximum output, but the manufacturer limits it in the spec, as you can see. There must be a reason for this limitation.*

Anyway, you can try using device up to 200 MHz on the first harmonic by enabling hidden parameter in the menu. But only if your Si5351a is able to output these frequencies reliably.

*And, moreover, the device is able to work **up to 450 MHz** (using 3rd harmonics of both signal and LO above 150 MHz, or above 200 MHz if confgured as above). Although the IF signal is 20 dB weaker, this is not a big issue since there are no strong interfering signals induced in the UHF antennes, unlike on HF. But the performance of many 602/612 type mixers degrde significantly at the frequencies above 200..300 MHz. Probably, increasing supply voltage of the mixers (only) to 8 Volts could help, but I did not try. In order to extend the working frequencies up to 450 MHz, enable hidden params in configuration menu, and choose the upper bound. Note that above 150 (or 200 if configured so) MHz the measurements will appear noisy (the signal is 20..40 dB weaker). I recommend to minimize another hidden param (LIN ATTENUATION), set it to 0 in this case. But still some mixers, or your RF choke may limit the maximum usable frequency to a significantly lower bound. E.g. in my case the signal level drops rapidly at frequencies above 320 MHz. Another possible reason of this drop is the half-wave resonance of the coax shield length on the RF choke, make it not longer than 1/4 wavelengths at the highest frequency. Also note that full recalibration of the device (includng HW cal) is needed if you increase the maximum frequency, and the calibration runs much longer.*

Some screenshots and photos



Hardware

Device is implemented using STM32F746G-DISCO Development Board, which is cheap but full of features needed for this project:

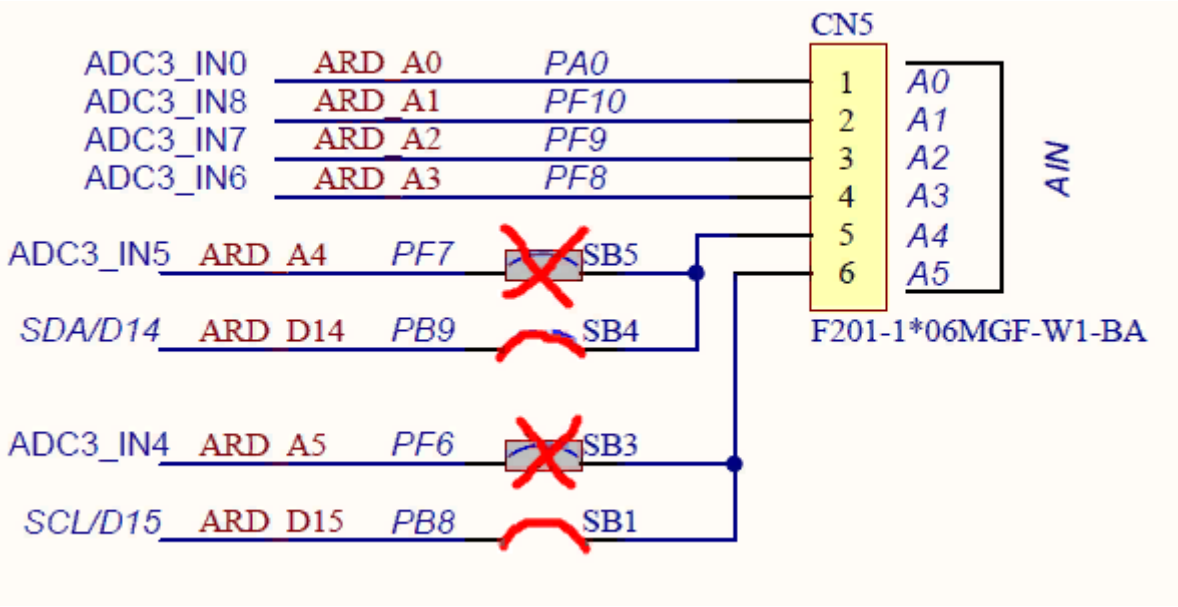
- fast but low power ARM Cortex M7 CPU core running at 216 MHz, with floating point hardware accelerator that boosts digital signal processing
- big and fast 480x272 4.3" TFT LCD display, driven by controller built into the CPU and mapped directly to the on-board RAM
- capacitive touchscreen on the display - no buttons are needed
- excellent on-board audio codec with stereo line input, a great addition for signal processing
- on-board ST-Link V2.1 in-circuit debugging interface
- Arduino Uno compatible shield socket
- a lot of peripheral interfaces
- open-source code of the drivers for all peripherals and on-board devices, provided by ST Microelectronics. Tons of bugs in it, but it was very challenging to find and fix them.
- DSP math libraries provided by ARM, with FFT code highly optimized for speed

And, finally, there is no more zoo of various poor quality boards and displays that Chinese guys supply and that were used in version 2. This development board is designed and supported by ST Microelectronics.

You'll only need to build and add a small RF Frontend in the form of Arduino Uno similar shield, and make some additional interconnections.

You'll need also a Micro-SD card (4 GB is more than enough) where the board will store it's configuration files and screenshots, and a 5 V power supply (any cheap power bank used to autonomously charge smartphones can be used).

** Development board should be modified a little to control the RF frontend via I2C: solder bridges SB5 and SB3 should be removed. Solder bridges SB1 and SB4 should be soldered in instead. All these bridges are near the Arduino Uno shield connector CN5 at the corner of the board, near the SD card slot, they are clearly marked. See the picture below. **



RF frontend schematic

Yes, it is very simple and contains no parts that are difficult to find. Though it works significantly better than the RF frontend of Version 2.

The frequency synthesizer is built on a well proven Si5351a chip, like in version 2 of the device.

There are two channels in the device: Voltage channel (V) and current channel (I). The device actually measures magnitude and phase difference between these channels to calculate impedance of the load connected to the input.

There is a noticeable novelty in the measurement bridge schematic. It avoids mixer imbalance influence on magnitude measurements due to strong common mode signal, i.e. the mixer no longer "hangs" with high level RF signal applied to both inputs. No matter what you do, the actual in-phase signal suppression is 25-30 dB in this case, this significantly limits the device precision. This new design introduces low Ohm measurement resistors connected to the ground. The inputs of the mixers are not exposed to a strong in-phase signal when the load impedance is high in this case. The RF choke at input has a very high impedance and does not shunt low-Ohm resistor to noticeable extent. Any phase and magnitude change it brings is easily compensated by calibration. I did not see this idea used before in similar devices, though, later I've found it in the article by Joel Dunsmore: Broadband_Coupler_Dunsmore.pdf. Good ideas never come to one head. :)

The RF choke is 3-4 turns of 4mm diameter 50-Ohm coax (I used RG-303) on a ferrite bead taken from the signal cable (D-SUB) of an old CRT monitor. The impedance of it is several orders higher than that of the current measurement resistor (5.1 Ohms used). But the length of the cable wound on the core, from the bridge to the grounding point, must not exceed 1/4 wavwlength for planned device's maximum operating frequency in order to avoid shunting the measurement resistor due to resonance. But this can aggravate performance on the lowest operational frequency if you plan to use device up to 450 MHz. It is up to you to find the best core to satisfy these requirements. Please note that the shield of the coax **must be grounded to PCB** right behind the choke, see the schematic. Also take into account that since the RF choke is included into the bridge, it should be properly mechanically fixed, otherwise the device calibration may be severely disturbed if the choke is moved, especially on VHF frequencies.

After many experiments I've found the RF choke design to be the optimal in the entire frequency range. It utilizes a tiny Amidon BN61-2402 binocular core, and the line wound on it is just a twisted pair made of dia. 0.2mm enameled wire, Z0 is about ~60 Ohm. The length of the line is short enough to avoid parasitic resonances, though the impedance in the entire frequency range is very satisfying. The line is wound three turns through one hole, and then it continues three turns in another one, see the picture. The idea of it came from the article.pdf

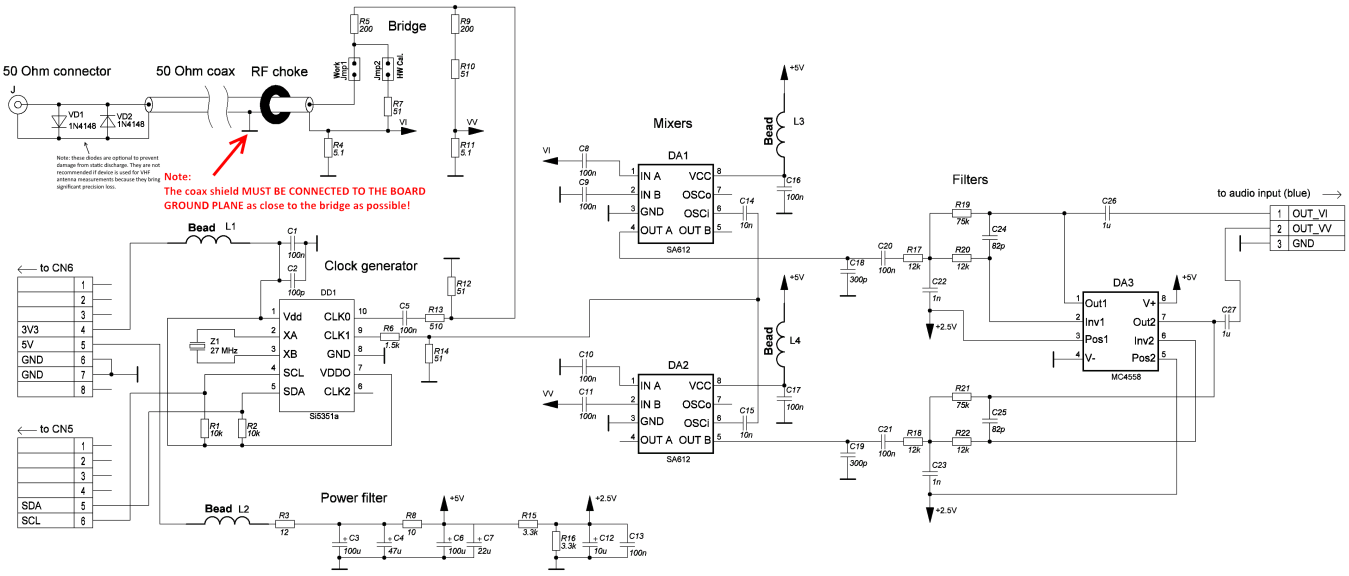
found somewhere in Internet.

Also, Mikko AB6RF had a time to build and measure properties of different types of chokes for the analyzer. His article in the discussion group is really helpful.

Two mixers (cheap and widely available 612 type) convert the I and V signals to the intermediate frequency of 10031 Hz which is then passed to Low pass filters.

OUT_VI is connected to the **left channel** of linear audio input (blue 3.5 mm connector), OUT_VV to the **right channel** (at the same blue connector) of the Discovery board. The rest is handled mathematically by the firmware.

Open the schematic in a new window to see it full size



Parts:

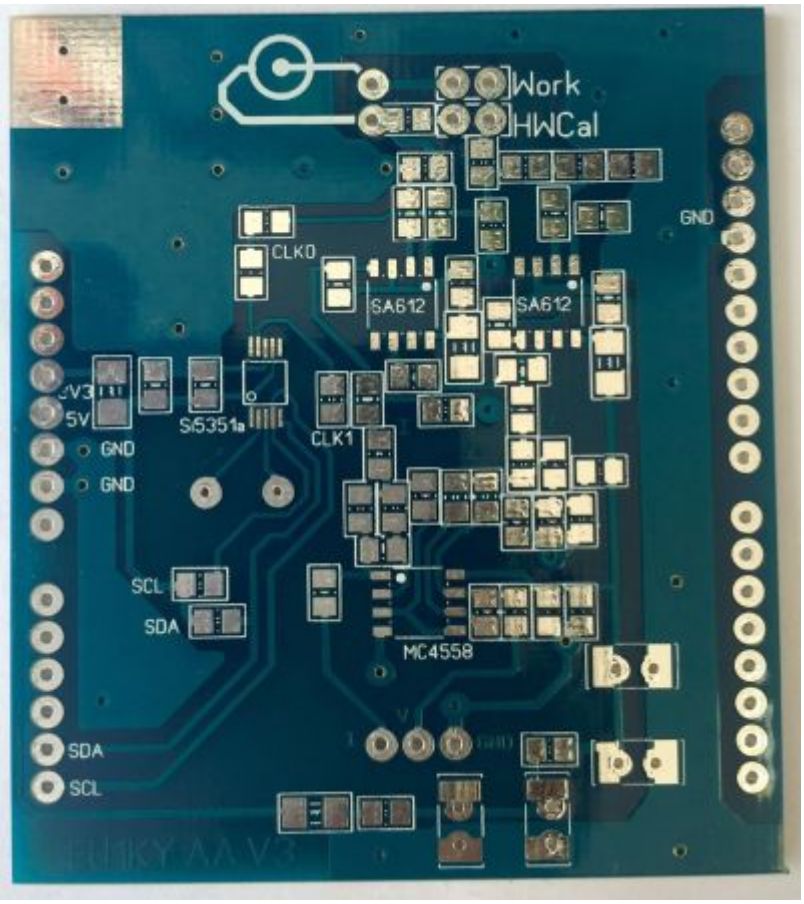
- All SMD resistors: 0805 1%
- All SMD capacitors: 0805, >=16V
- Electrolytic capacitors: V >= 16V
- Beads: any, 0805
- SA612 (can be replaced with SA602, NE612, NE602): in SO-8 package
- MC4558: in SO-8 package. Can be replaced with any opamp in the same package with the same pin assignment, that works from 4.5V unipolar voltage, and with GBW product > 3 MHz.
- Si5351a: 10-MSOP package
- Crystal: any 27000 kHz fundamental frequency quartz (or 25000 kHz can be used as alternative, provided that it is properly chosen in device's configuration)

RF frontend PCB

The prototype PCB design in Sprint Layout 6.0 format can be downloaded in the Downloads section, the assembly drawing is also there. There is also an edited version by YL2GL.

The PCB is double-sided. The other side (not shown in the file) is not etched and is used as ground plane. Some vias should be connected to it, but those vias that should not be grounded (e.g., unused pins of the shield connector) must be countersunk at the ground plane to prevent short circuits.

Industrially manufactured RF frontend PCB can be ordered from R3KBL (yuraws@gmail.com)



Cloning the source code

You'll need to install GIT to download and pull changes to the source code.

```
$ git clone https://github.com/EU1KY/eu1ky_aa_v3
```

Build environment

If you don't want to build the firmware from sources, just take the prebuilt binary in the Downloads section of this page.

The project binaries can be built in Windows with EmBitz IDE, there is project F7Discovery.ebp file that should be opened with the IDE. The IDE is free, and allows in-circuit debugging via the ST-Link debug interface that is present on the devboard. Choose Release build target for optimized firmware, or Debug target if you want to debug the code.

Built binary file can be downloaded to the board with STM32 ST-Link utility (available from ST Microelectronics site) with MiniUSB cable (it is not supplied with the board, find it yourself).

Commissioning

Having assembled the device and downloaded the firmware into it, you need to test it and prepare to operation.

Ensure you have the **empty** (formatted) micro-SD card inserted in its slot on the Discovery board, otherwise the firmware refuses to boot. Power on the device. The main menu window should appear.

Move the jumper from JP1 to JP2 on the RF frontend shield.

Select "DSP" menu. You should see the two spectrum windows showing the two powerful carriers. You should see no noise in the near vicinity of the powerful signal peaks which should look as 5-pixels wide "towers". Do not consider noise in the left (low frequency) parts of the windows, it is normal influence of AC mains surrounding you.

The magnitude of the measured signals in DSP window should be around 1200-1600 (ideally they should be equal), the frequency displayed should be 10031 Hz (bin 107). If something is wrong (no signal, too much noise, big difference between the signals, or detected frequency is not 10031 Hz) - find the reason and fix it before proceeding. *FYI: in the DSP window, the device puts 3500 kHz signal to the RF bridge. Horizontal lines are each 10 dB, so you can easily estimate signal to noise ratio.*

It may happen that you don't see the signals in neither of the windows. The most probable reason is that there are lots of Si5351a chips with the bus address which differs from the default address published in specs (C0h). Try choosing other known addresses in Configuration menu (C4h, CEh), or select "Autodetect". If autodetection succeeds, green message will appear and the correct address will be stored. If it fails and red message appears, Si5351a has not been detected, it's up to you to find out why.

Return to main menu by tapping the lower left corner of the screen (*this is the common way to exit to main menu from any window*).

Then enter to "Generator" menu. You should see stable magnitude and phase readings on various frequencies (set by tapping at the top of the screen). Ensure the magnitude differences between the channels does not exceed +/-1.5 dB, and phase difference does not exceed 10 degrees. If it is not so, find the reason and fix it before proceeding.

In the "Generator" window, set the frequency of your crystal connected to Si5351a, e.g. 27000 kHz, and listen to this frequency on your favorite receiver. Find the carrier from the device and measure how far it is from 27000 kHz. It is common to see the device transmitting several kHz above, e.g at 27,004,821 Hz. So, the measured difference is 4821 Hz. Remember it. Return to main menu and enter "Configuration" window. Select configuration parameter SI5351_CORR and set the measured frequency difference (e.g. +4821). Save and exit from configuration window.

Ensure that the jumper on the board is set in JP2, and removed from JP1. Now run "HW calibration". The device will scan the magnitude and phase difference coefficients in the entire of its band. AFter it finishes, return to main menu. Get in the Generator window and ensure that it shows now 0.0 dB magnitude and 0.0 degrees phase differences between the channels in the entire frequency range. Now remove the jumper from JP2 ("HWCal") position and set it to JP1 ("Work").

After this select active OSL calibration file in the configuration window and run OSL calibration procedure. After this procedure the device is ready for use.

Do not forget to save changes in configuration and results of OSL calibration!

Other info

- Configuration file contents reference.
- Version history
- Инструкция по настройке прибора после сборки
- There is a discussion forum for this project (in Russian)
- There is a discussion forum for this project (in Polish)
- A good article to understand TDR feature of the analyzer: "Time Domain Analysis Using a Network Analyzer" (http://literature.cdn.keysight.com/litweb/pdf/5989-5723EN.pdf)
- DH1AKF firmware mod
- Analyzer plastic case model for 3D printer by Martin DL8CQ
- Discussion group at the groups.io

License

The entire project, as a whole, is in public domain, and even more, under the terms of WTFPL Version 2 license, as long as you **retain notice of my (EU1KY) authorship**. See http://www.wtfpl.net/ for details.

Note that some parts of the source code are under other freeware license terms, see inside the files for information.

Clone this wiki locally

https://github.com/EU1KY/eu1ky_aa_v3.wiki.git