

Open Source herconfigureerbare instrumenten

Red Pitaya SDRlab en STEMLab

Erwin Serlé- PE3ES / F4VTQ



Open Source herconfigureerbare instrumenten

Moderne laboratorium meetapparatuur die je ook nog als SDR kunt gebruiken.
De Red Pitaya STEMLab en SDRlab niet alleen voor WSPR en FT8

PE3ES
14 maart 2021

STEM is een Engels acroniem voor
Science, Technology, Engineering en
Mathematics.

JN28 – met uitzicht op de buurman



Agenda

Introductie van de begrippen SDR DSP FPGA
Red Pitaya
Wensen van radioamateurs
Basis software
De SDR software op een PC of doen we al het werk op de Red Pitaya
Alternatieve software van Pavel Demin
Meerband ontvangers voor WSPR en FT8
Resultaat in de praktijk
Hardware en software aanpassingen
Hoe leer je al die software kennen
Technische metingen
Alternatieven
Conclusie en discussie

Kernpunten

Wat is SDR, digitale signaalbewerking en een FPGA

Red Pitaya en de verschillen tussen een STEMLab Red Pitaya en een SDRlab Red Pitaya

Een platform met meetinstrumenten

Wensen van radioamateurs

Ontvangers

Extra hardware en wat er nodig is om er een transceiver van te maken

Al die software

Radio programmeren

Instrumenten programmeren

Afronding

Introductie SDR DSP FPGA

Wanneer wordt het een Software Defined Radio?
Mijn Yaesu FRG-7 is geen SDR !



Mijn Kenwood R-1000 met zijn digitale frequentie uitlezing? Nou, nee.



Mijn uBITx met zijn besturing via een programma in een Arduino Zero?



Software Defined Radio

Minstens een deel van de HF signalen is omgezet van analoog naar digitaal door gebruik van een ADC

→ Shannon / Nyquist

Deze signalen worden digitaal verwerkt

De signaalverwerking kan aangepast worden op het niveau van de software (dus zonder solderen of het omprikken van kabels)

Uitgaande signalen (in het geval van een zender) zijn digitaal gecreeerd voor ze door een DAC weer in het analoge domein werden omgezet

Signaalbewerking

Naar ander frequentiegebied omzetten (local oscillator / mengen)

Decoderen (AM, FM, Digimodes, CW)

Filteren (laag, band, hoog)

Versterken

Signaal conditionering

Foutcontrole

Berekeningen

Digitale signaalbewerking

HF signaal voldoende snel bemonsterd (gesampled, Nyquist/Shannon) en omgezet naar een stroom cijfers

Zoals we dat kennen van de geluidskaart met microfoon- of lijningang op de PC

Dat is snel genoeg voor spraak of voor muziek

Voor HF moet de ADC boven 2 x de gebruikte HF-frequentie kunnen werken → direct conversion

En dan moet de signaalbewerking die daarop volgt het ook kunnen bijhouden

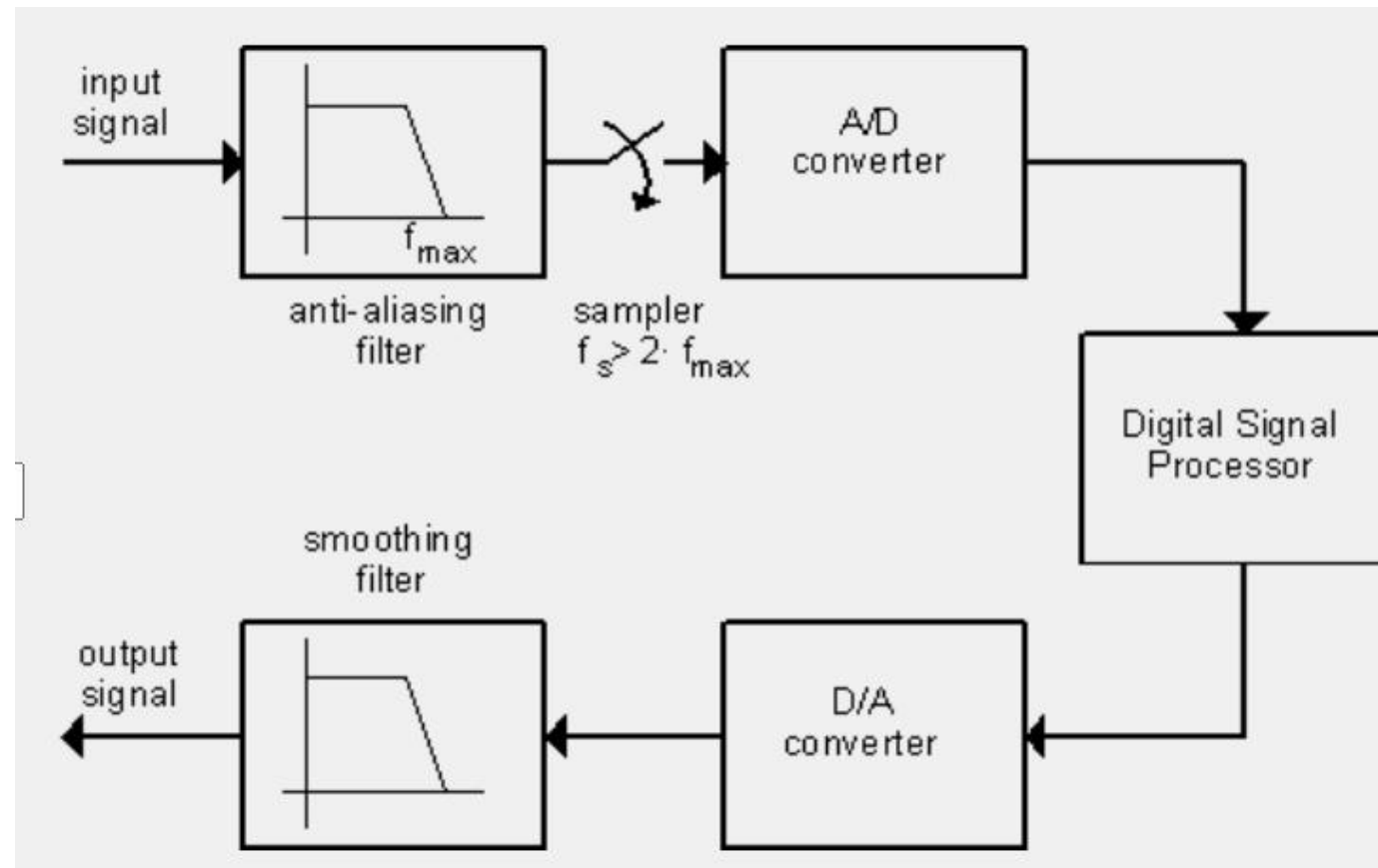
Dat kan met discrete, logische componenten

Of met een snelle PC en de juiste software, zoals GNU-Radio

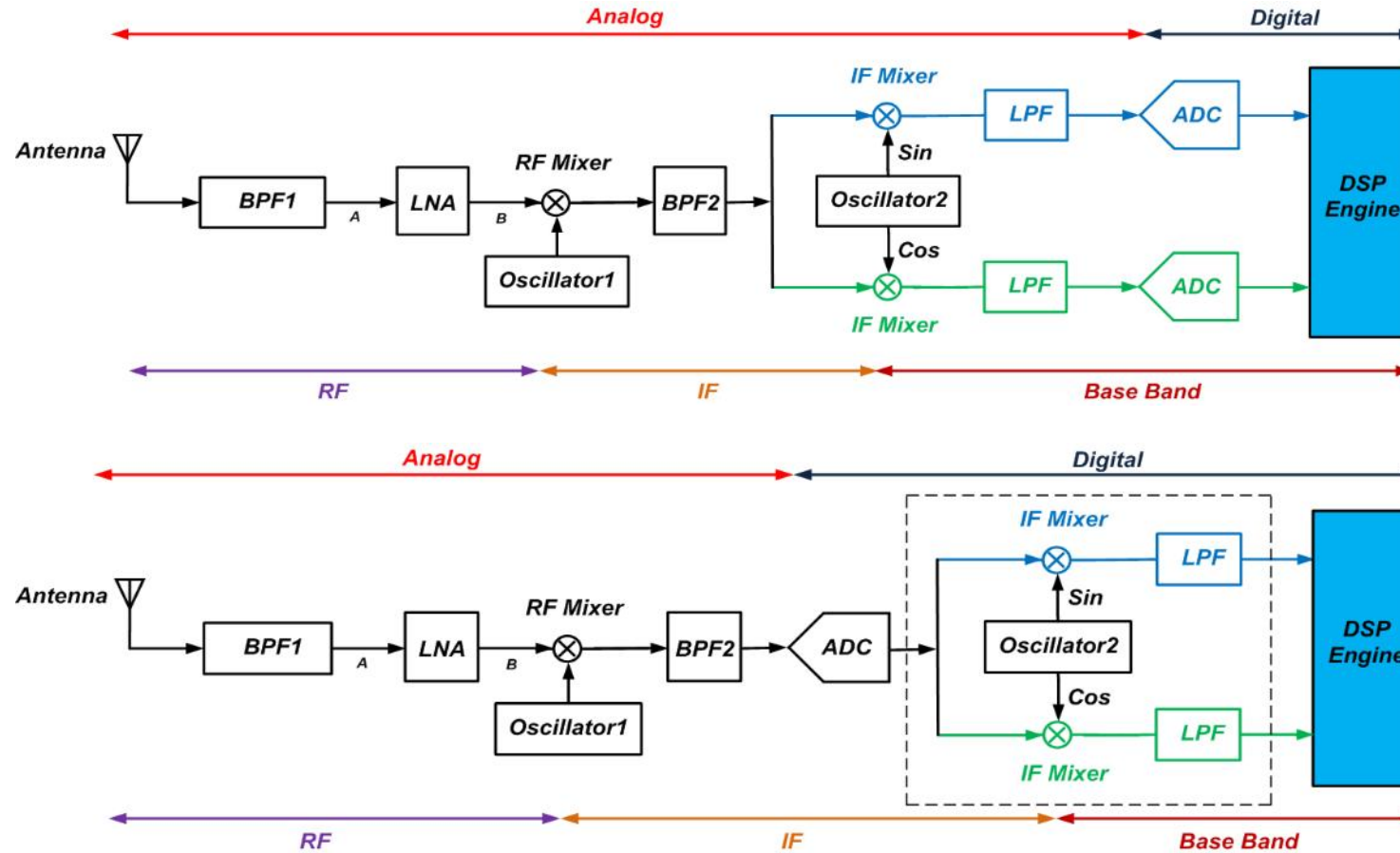
Of met een geïntegreerde oplossing met logische componenten die in een chip gebakken zijn : DSP-chip

Of met een FPGA

Digitale signaalbewerking



Vanaf waar wordt het digitaal ?



Digitale signaalbewerking

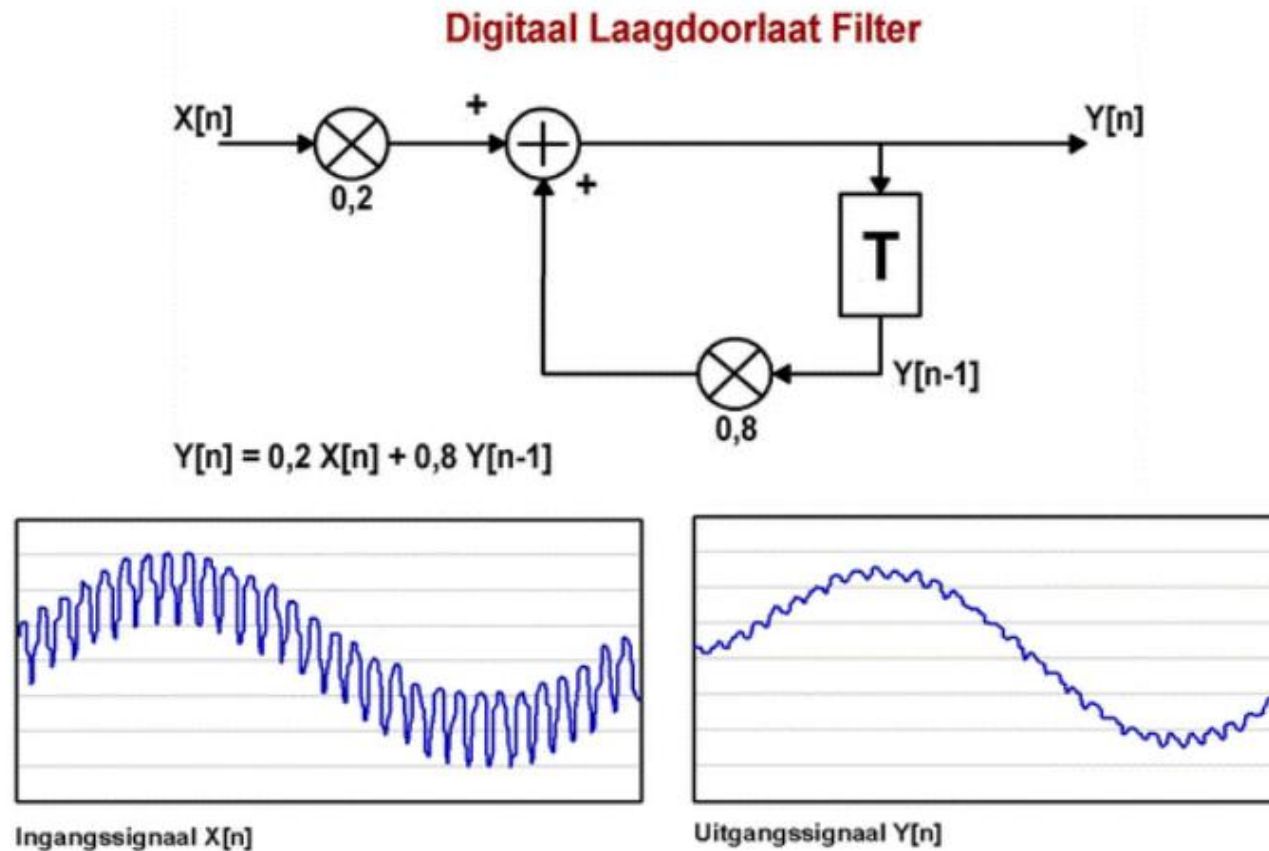
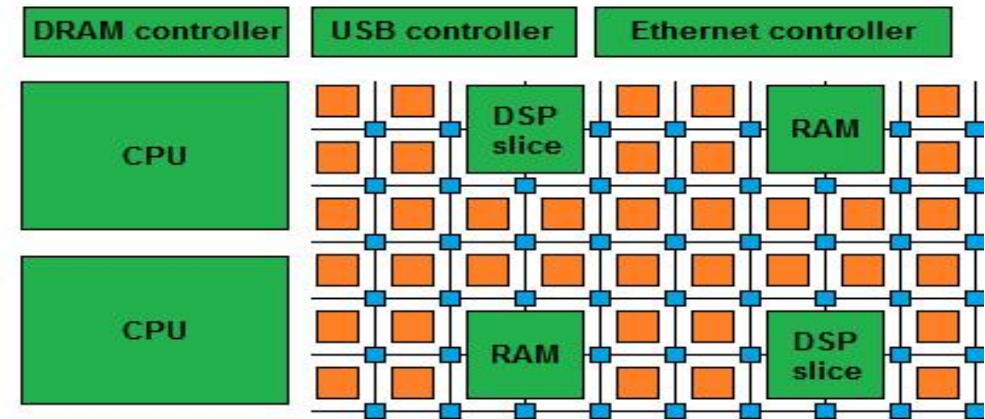
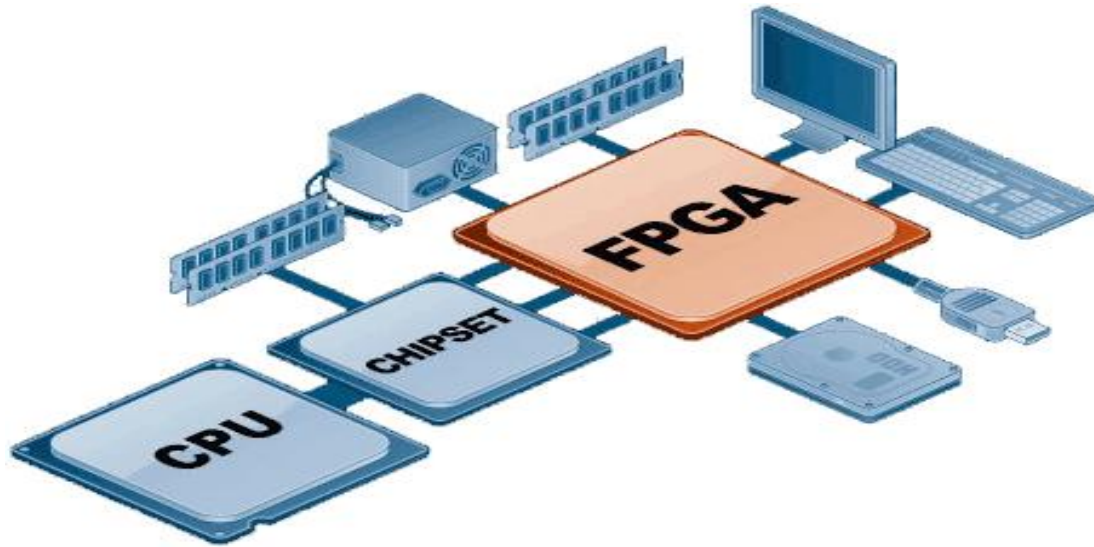


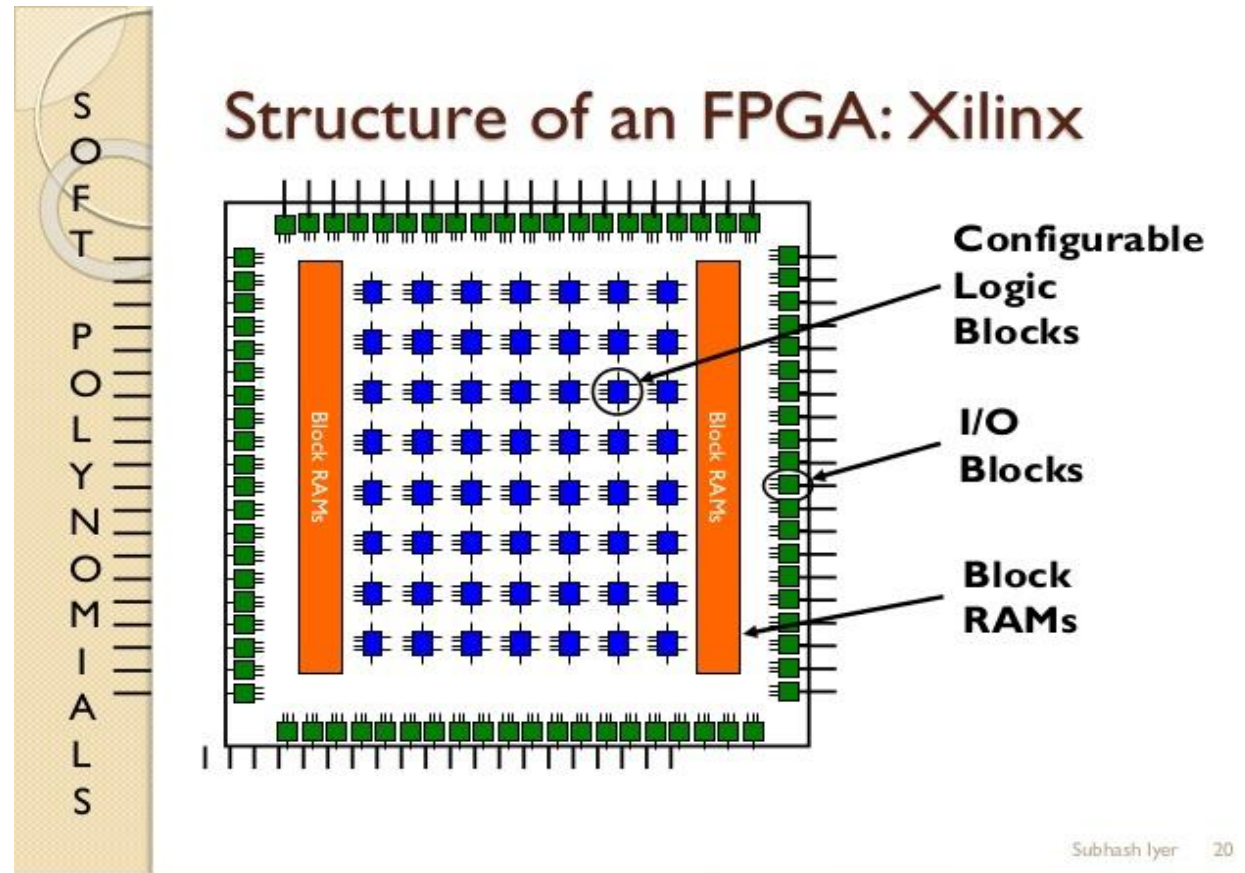
Fig.5a. Voorbeelden van een digitaal laagdoorlaatfilter.

CPU – FPGA - SoC

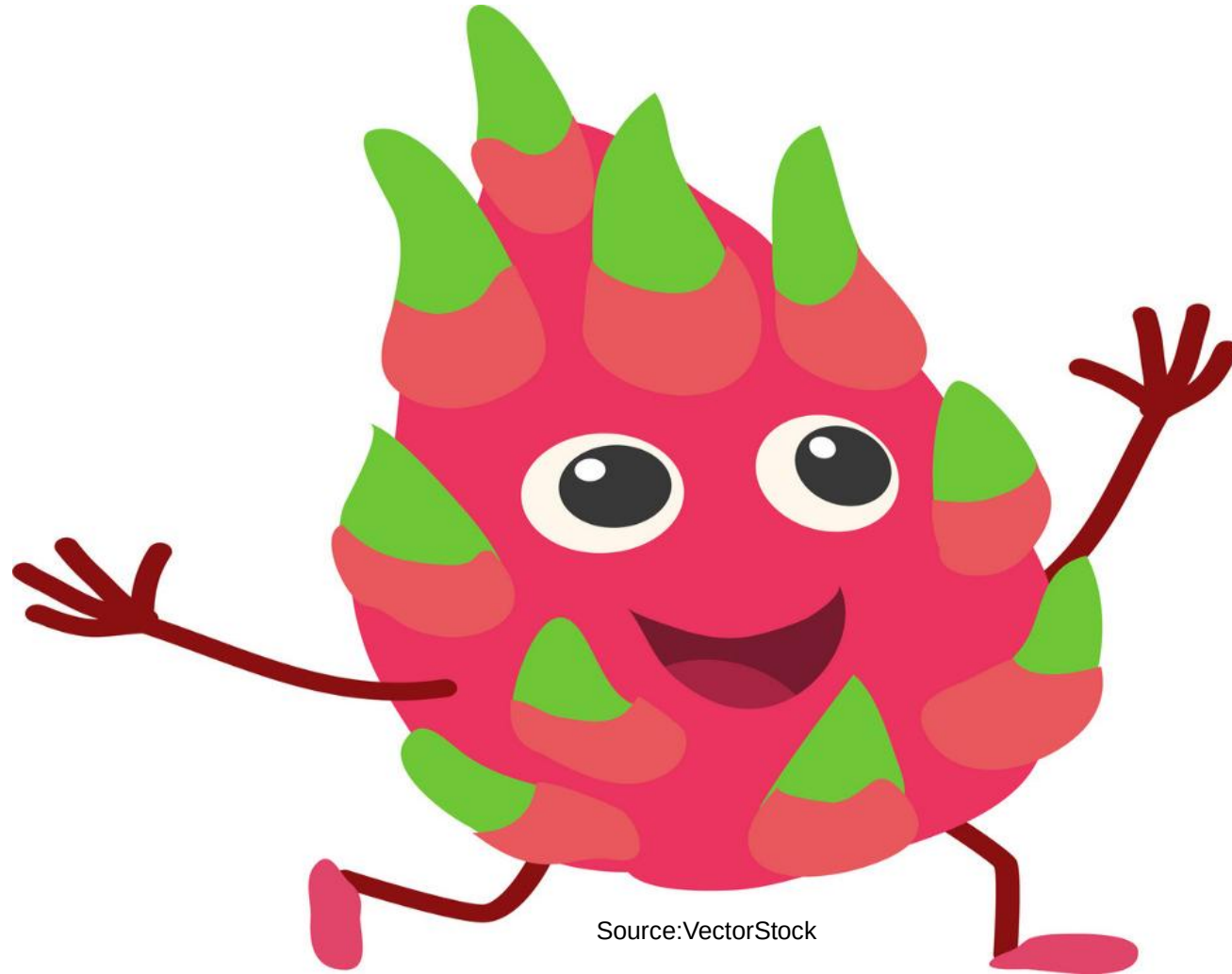


Modern FPGA: lots of **hard**, not-field-programmable gates

Field Programmable Gate Array



Red Pitaya



Source:VectorStock

Red Pitaya - Slovenië

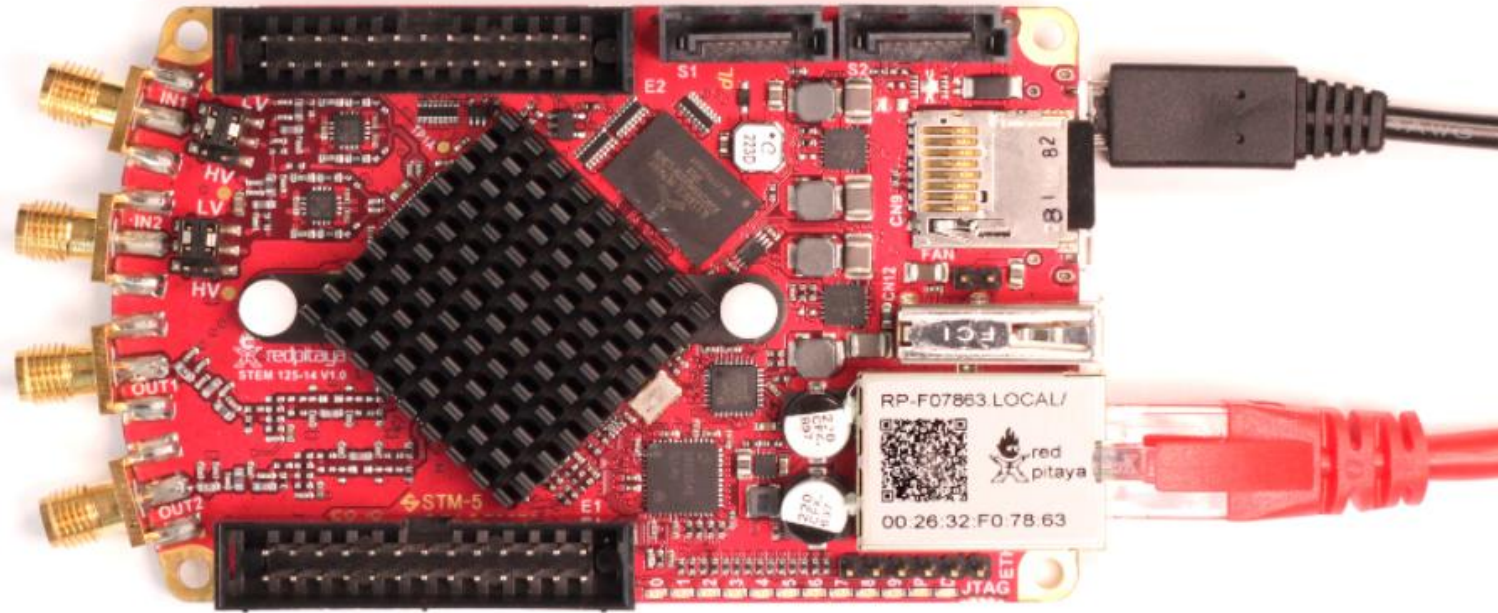


Red Pitaya is a privately-owned company established in 2013 as a result of a spin off from Instrumentation Technologies Inc, the leading company designing and building high performance instruments for particle accelerators. With the help of a successful Kickstarter campaign, deep engineering background and passionate young individuals with innovations in their hearts we are constantly growing and changing the test and measurement market. As a confirmation of our innovative approach we were awarded the prestigious Frost & Sullivan Global Electronic Test&Measurement Tools New Product Innovation Award in 2014.

Our culture is in the passion for what we do, and the pride in the products we deliver.

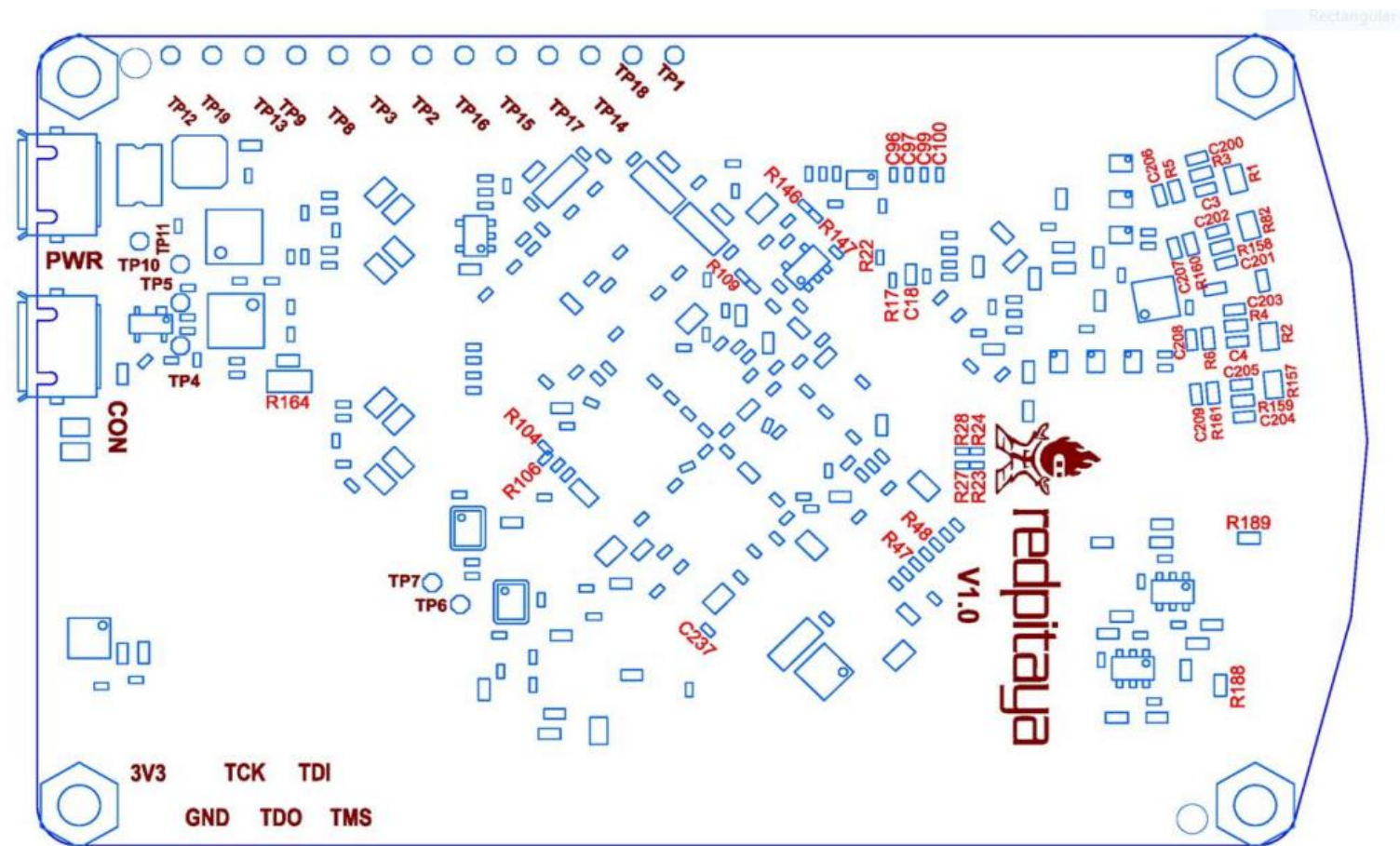
Red Pitaya, the company that is pioneering the move to high performance, open-source, reconfigurable instrumentation with its credit-card-sized STEMLab™ platform.

Red Pitaya STEMLab

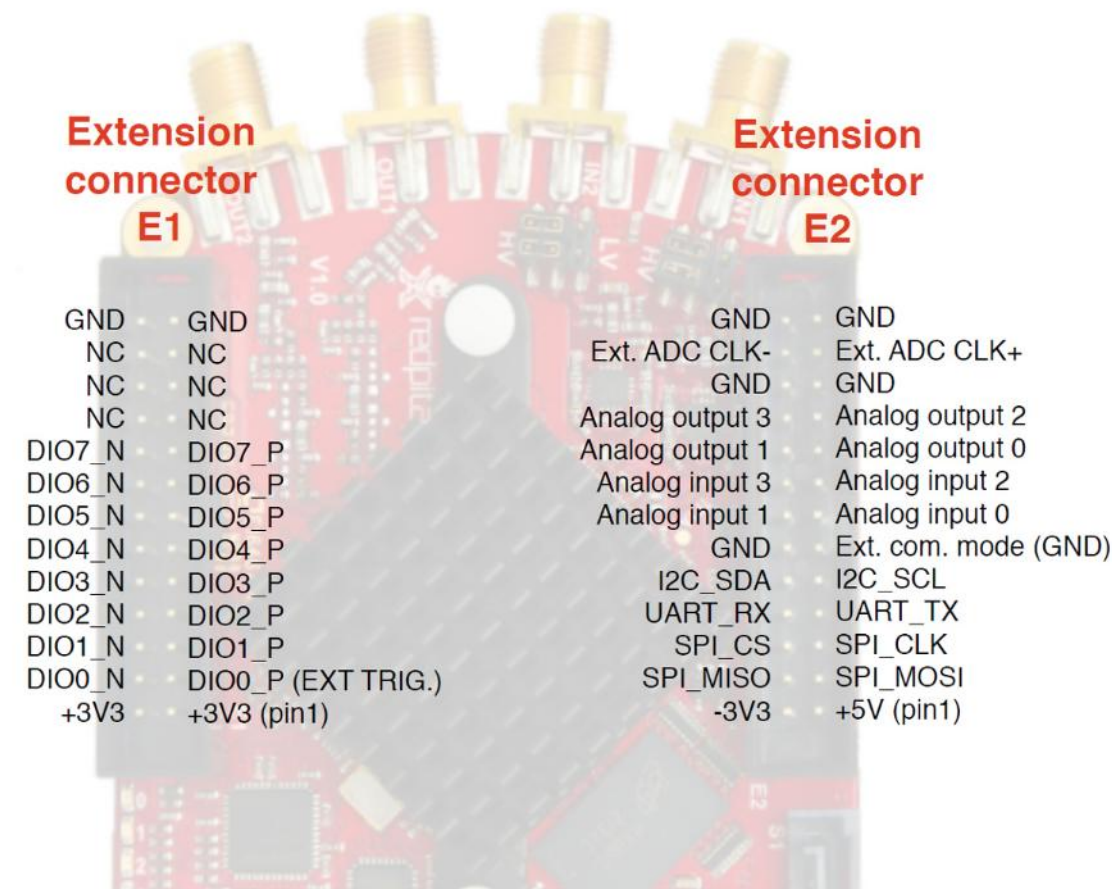


STEM is een Engelse afkorting voor Science, Technology, Engineering en Mathematics.

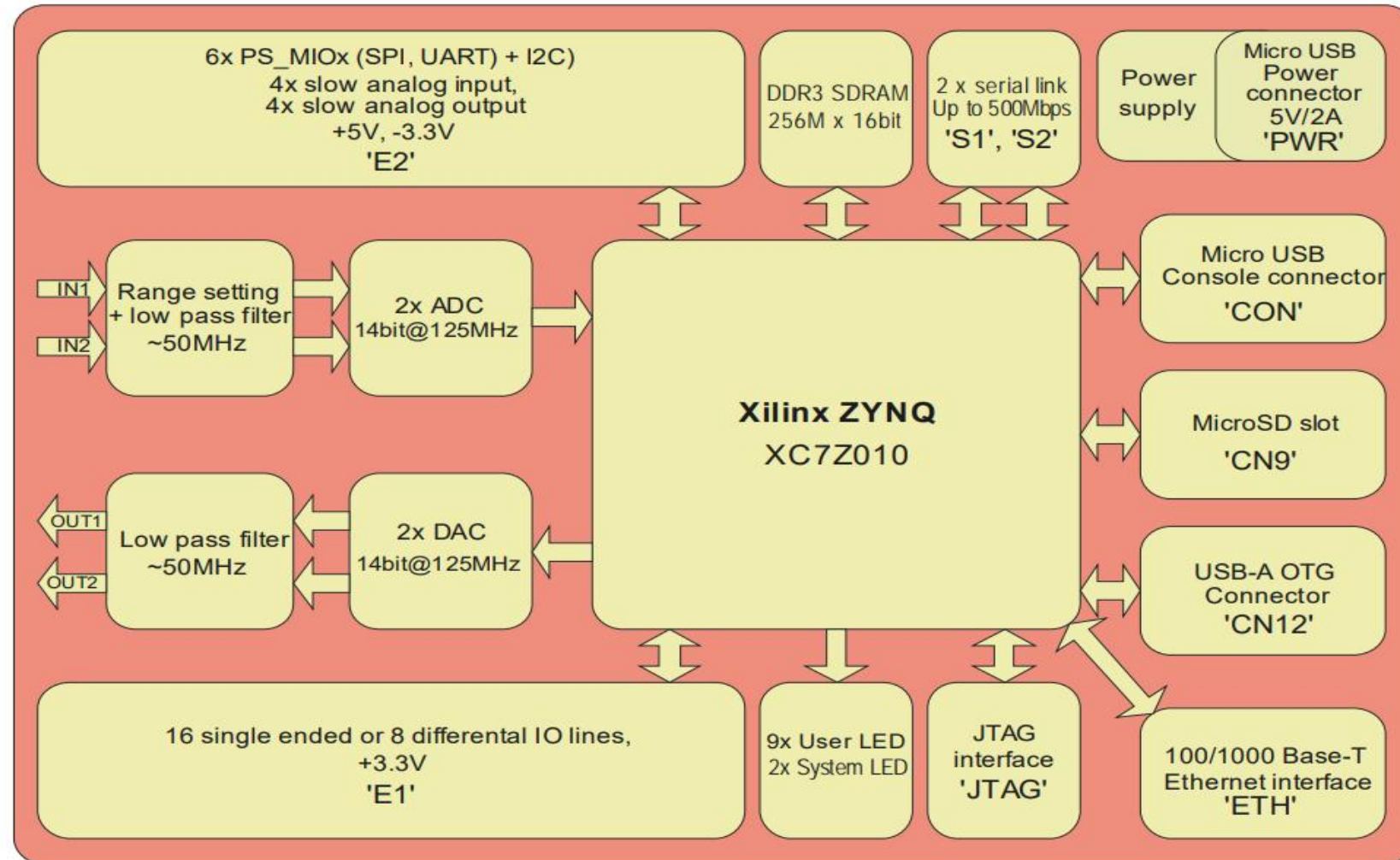
STEMlab onderkant



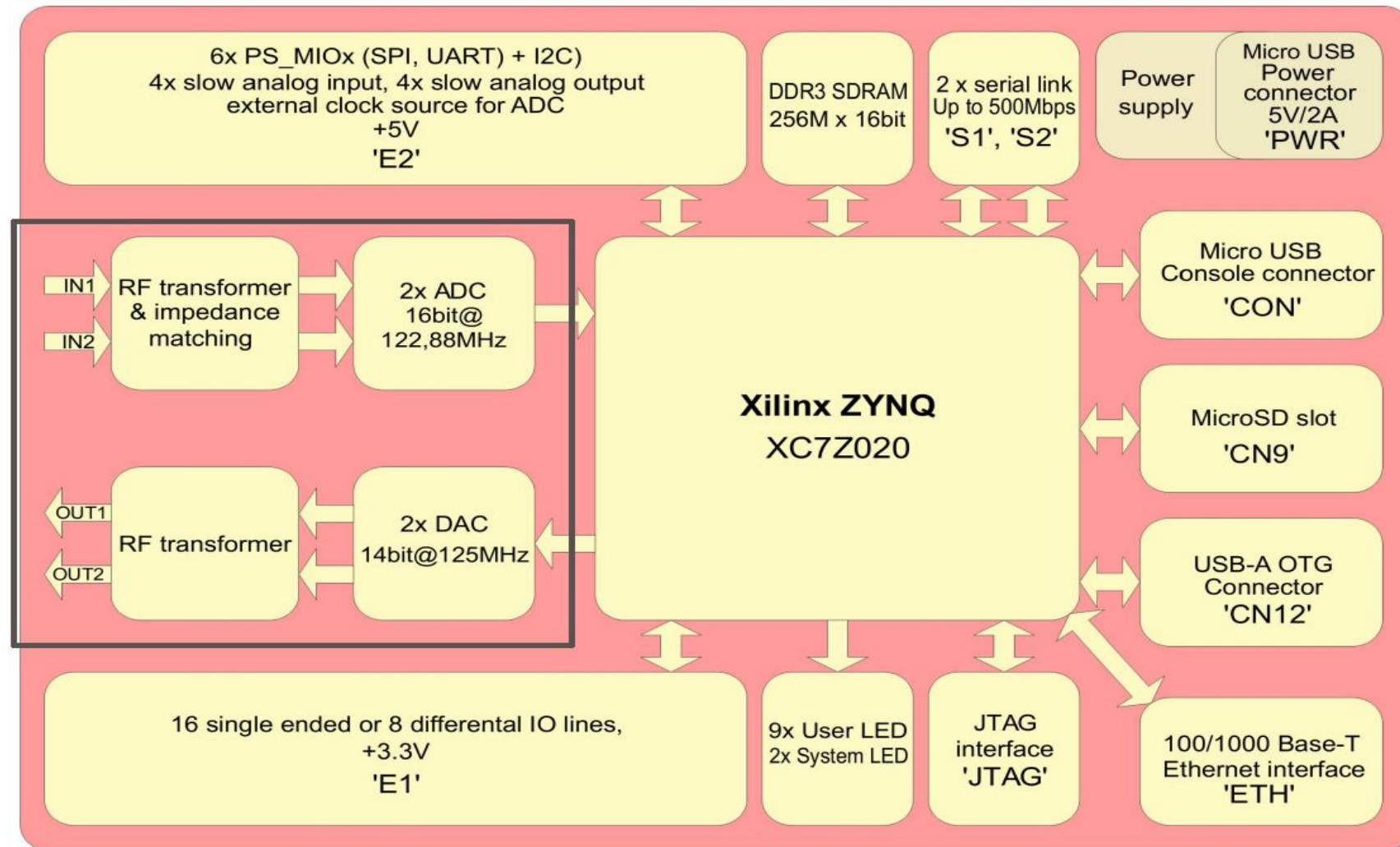
STEMlab aansluitpunten



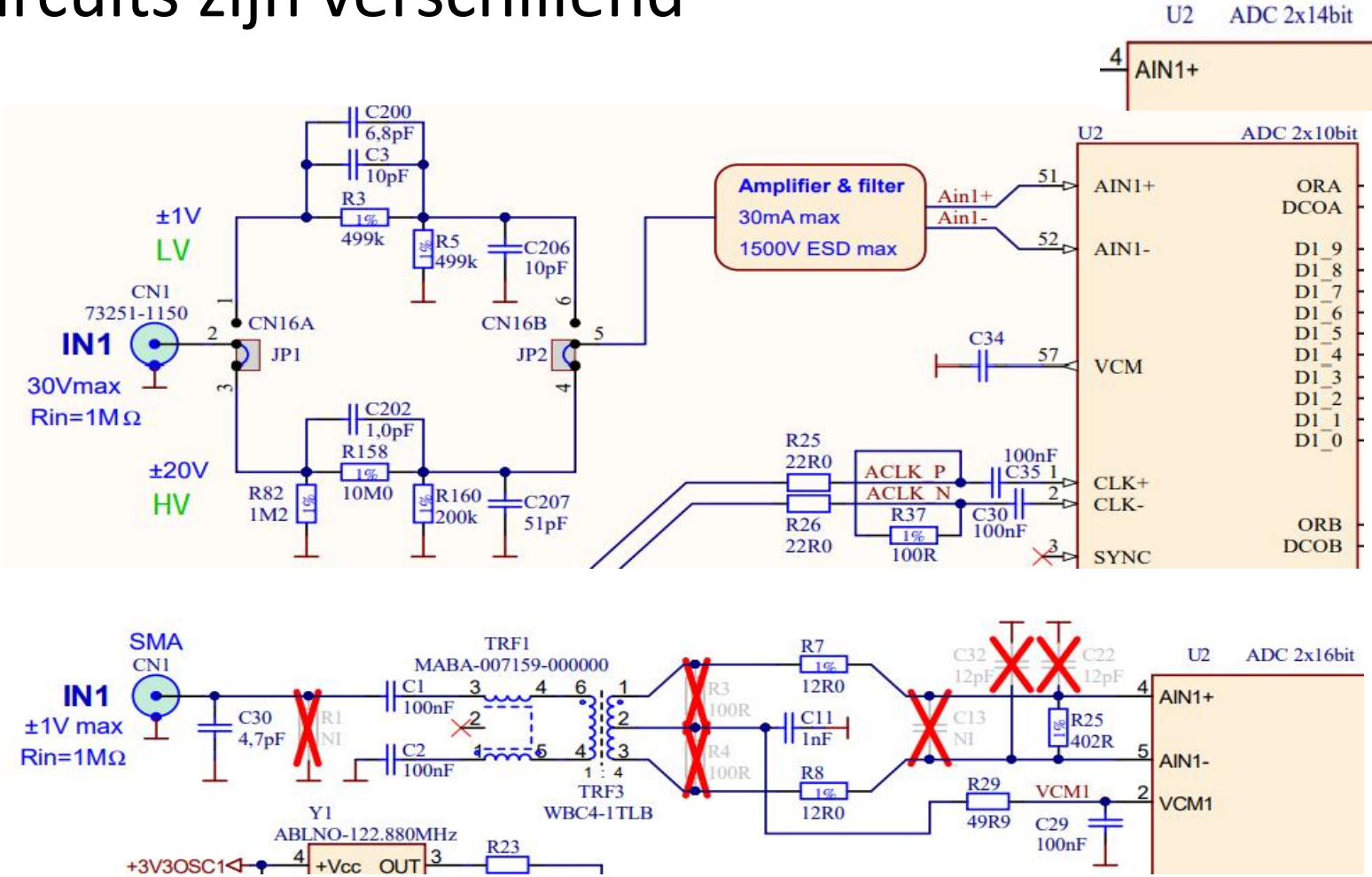
STEMlab schematisch



SDRlab schematisch



Ingangscircuits zijn verschillend



System op een chip

Zynq®-7000 SoC Family											
		Cost-Optimized Devices						Mid-Range Devices			
Device Name		Z-7007S	Z-7012S	Z-7014S	Z-7010	Z-7015	Z-7020	Z-7030	Z-7035	Z-7045	Z-7100
Part Number		XC7Z007S	XC7Z012S	XC7Z014S	XC7Z010	XC7Z015	XC7Z020	XC7Z030	XC7Z035	XC7Z045	XC7Z100
Processing System (PS)	Processor Core	Single-Core ARM® Cortex™-A9 MPCore™ Up to 766MHz			Dual-Core ARM Cortex-A9 MPCore Up to 866MHz			Dual-Core ARM Cortex-A9 MPCore Up to 1GHz ⁽¹⁾			
	Processor Extensions	NEON™ SIMD Engine and Single/Double Precision Floating Point Unit per processor									
	L1 Cache	32KB Instruction, 32KB Data per processor									
	L2 Cache	512KB									
	On-Chip Memory	256KB									
	External Memory Support ⁽²⁾	DDR3, DDR3L, DDR2, LPDDR2									
	External Static Memory Support ⁽²⁾	2x Quad-SPI, NAND, NOR									
	DMA Channels	8 (4 dedicated to PL)									
	Peripherals	2x UART, 2x CAN 2.0B, 2x I2C, 2x SPI, 4x 32b GPIO									
	Peripherals w/ built-in DMA ⁽²⁾	2x USB 2.0 (OTG), 2x Tri-mode Gigabit Ethernet, 2x SD/SDIO									
Programmable Logic (PL)	Security ⁽³⁾	RSA Authentication of First Stage Boot Loader, AES and SHA 256b Decryption and Authentication for Secure Boot									
	Processing System to Programmable Logic Interface Ports (Primary Interfaces & Interrupts Only)	2x AXI 32b Master, 2x AXI 32b Slave 4x AXI 64b/32b Memory AXI 64b ACP 16 Interrupts									
	7 Series PL Equivalent	Artix®-7	Artix-7	Artix-7	Artix-7	Artix-7	Artix-7	Kintex®-7	Kintex-7	Kintex-7	Kintex-7
	Logic Cells	23K	55K	65K	28K	74K	85K	125K	275K	350K	444K
	Look-Up Tables (LUTs)	14,400	34,400	40,600	17,600	46,200	53,200	78,600	171,900	218,600	277,400
	Flip-Flops	28,800	68,800	81,200	35,200	92,400	106,400	157,200	343,800	437,200	554,800
	Total Block RAM (# 36Kb Blocks)	1.8Mb (50)	2.5Mb (72)	3.8Mb (107)	2.1Mb (60)	3.3Mb (95)	4.9Mb (140)	9.3Mb (265)	17.6Mb (500)	19.2Mb (545)	26.5Mb (755)
	DSP Slices	66	120	170	80	160	220	400	900	900	2,020
	PCI Express®	—	Gen2 x4	—	—	Gen2 x4	—	Gen2 x4	Gen2 x8	Gen2 x8	Gen2 x8
	Analog Mixed Signal (AMS) / XADC ⁽²⁾	2x 12 bit, MSPS ADCs with up to 17 Differential Inputs									
Speed Grades	Security ⁽³⁾	AES & SHA 256b Decryption & Authentication for Secure Programmable Logic Config									
	Commercial	-1			-1			-1			-1
	Extended	-2			-2,-3			-2,-3			-2
	Industrial	-1,-2			-1,-2,-1L			-1,-2,-2L			-1,-2,-2L

Notes:

- 1 GHz processor frequency is available only for -3 speed grades in Z-7030, Z-7035, and Z-7045 devices. See [DS190, Zynq-7000 SoC Overview](#) for details.
- Z-7007S and Z-7010 in CLG225 have restrictions on PS peripherals, memory interfaces, and I/Os. Please refer to [UG585, Zynq-7000 SoC Technical Reference Manual](#) for more details.
- Security block is shared by the Processing System and the Programmable Logic.

STEMlab praktijk

Software op de SD kaart bepaalt hoe de Red Pitaya zich gedraagt

1. SD card image van Red Pitaya voor STEMlab

https://downloads.redpitaya.com/downloads/STEMlab-125-1x/STEMlab_125-xx_OS_1.04-7_stable.img.zip

 STEMlab_125-xx_OS_1.04-8_beta.img.zip	2/8/2021 12:13 PM	Compressed (zipp...	587,494 KB
 STEMlab_125-xx_OS_1.04-7_beta.img.zip	11/17/2020 2:47 PM	Compressed (zipp...	587,038 KB
 STEMlab_125-xx_OS_1.03-701_stable.img.zip	11/4/2020 9:15 PM	Compressed (zipp...	584,558 KB
 SDRlab_122-16_OS_1.04-8_beta.img.zip	11/17/2020 9:37 AM	Compressed (zipp...	580,434 KB
 SDRlab_122-16_OS_1.03-6_stable.img.zip	11/17/2020 9:42 AM	Compressed (zipp...	580,023 KB
 red_pitaya_OS-beta_1.03.img.zip	2/7/2021 10:34 PM	Compressed (zipp...	565,025 KB
 red-pitaya-alpine-3.12-armv7-20200628.zip	2/1/2021 11:00 PM	Compressed (zipp...	100,195 KB

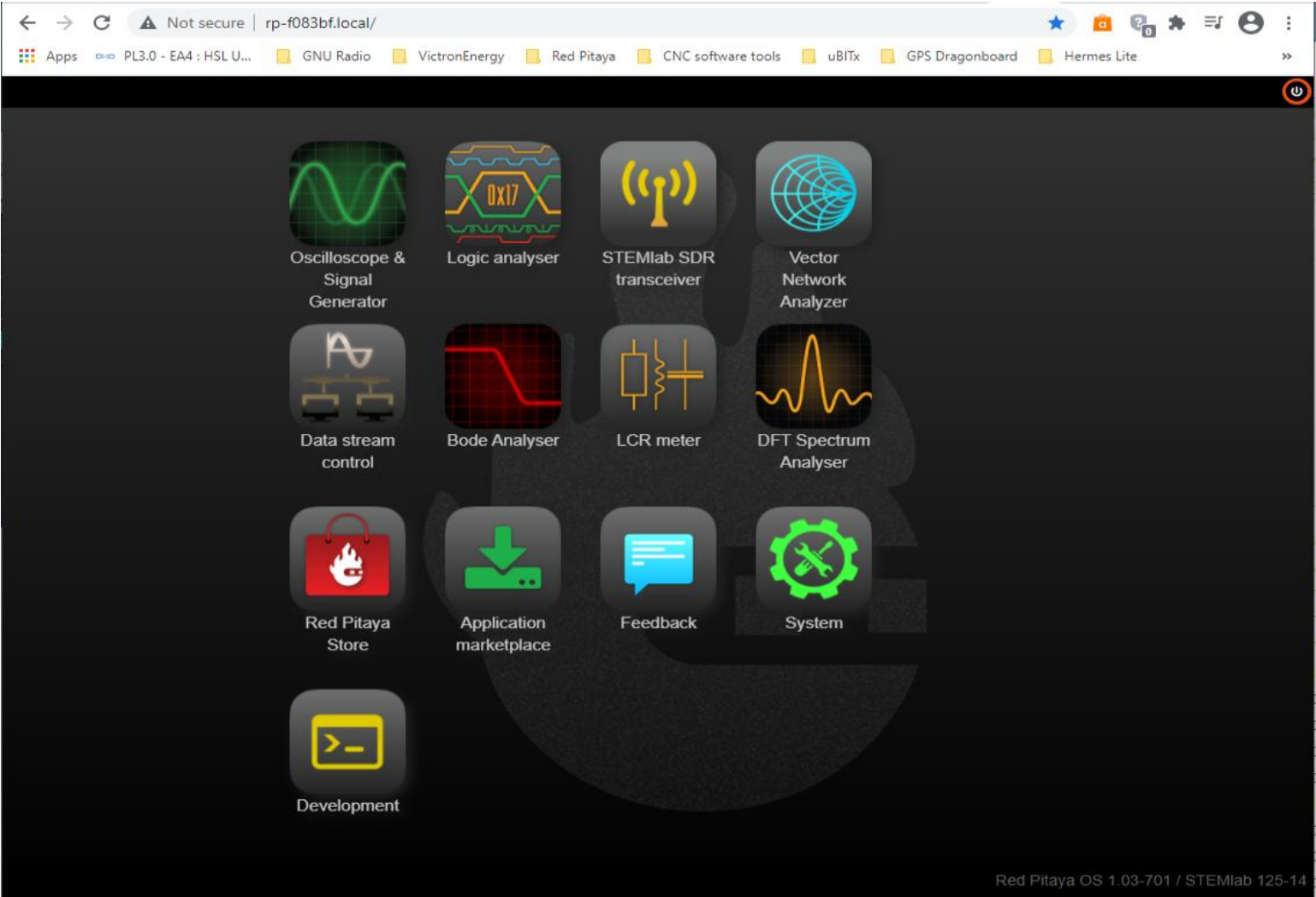
Installatie van de software

Zoals ook bij de Raspberry Pi of andere single board computers

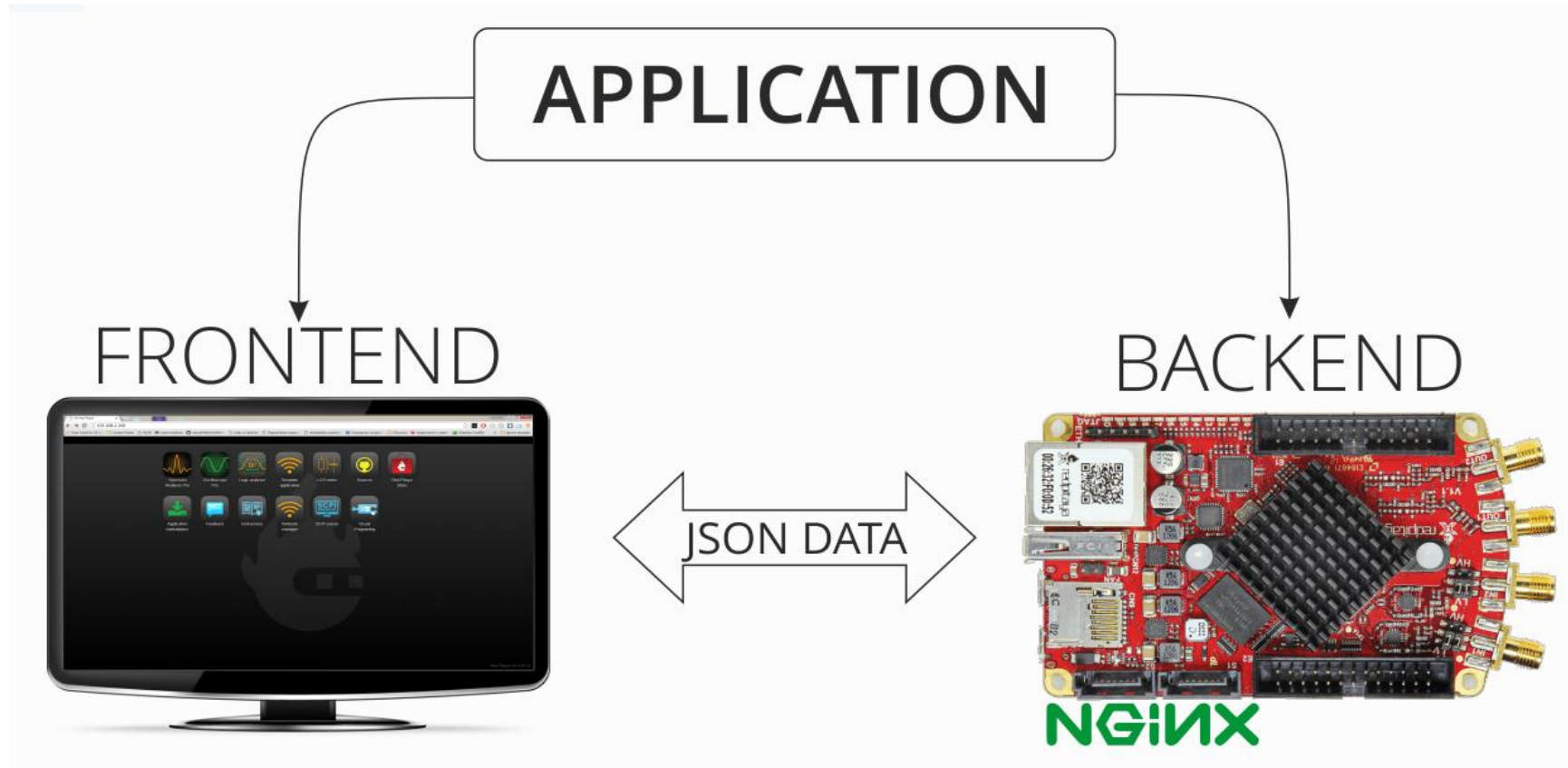
- uitpakken
- image schrijven met Win32DiskImager of balenaEtcher
- SD card in de Red Pitaya
- voeding aan

Red Pitaya krijgt via DHCP een IP address en ook is er een domeinnaam
Open een web browser en kies als adres `rp-xxxxxx.local/`

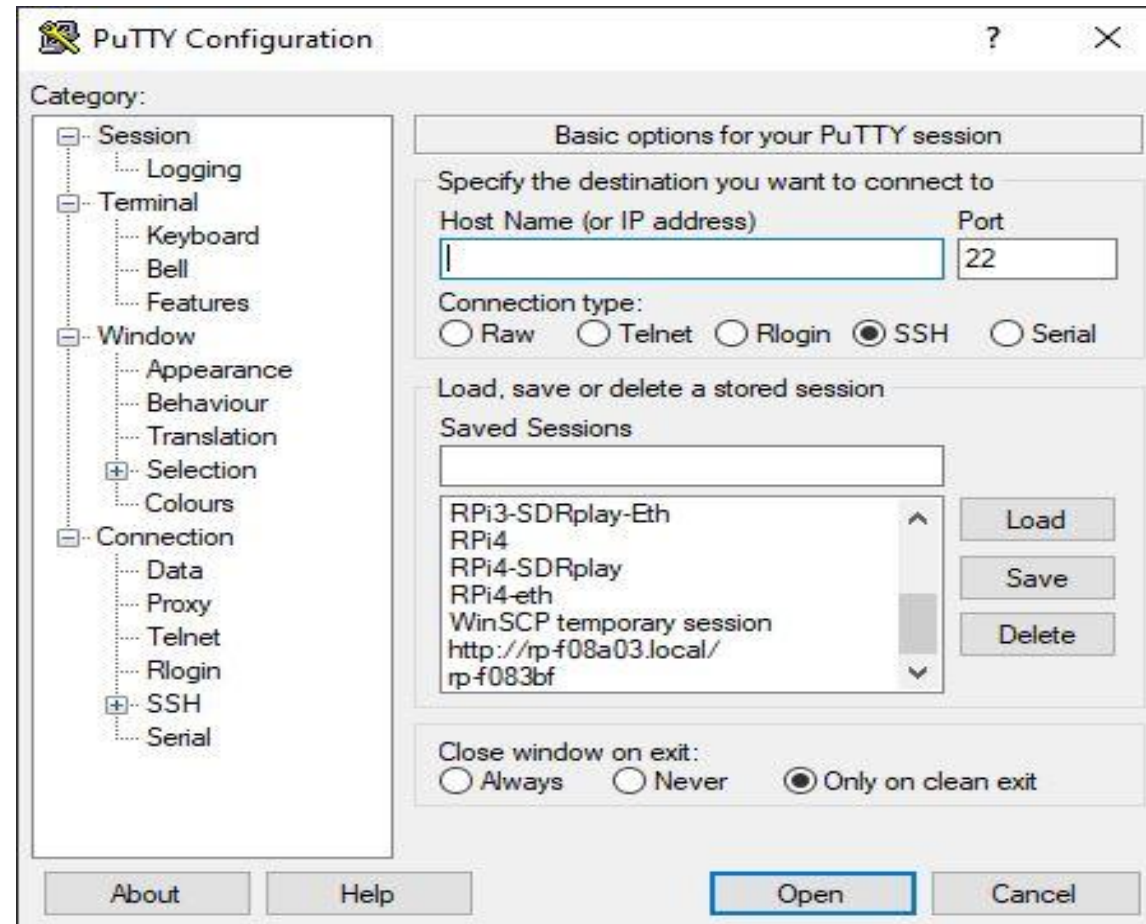
Bediening via de internet browser



Bediening / verwerking



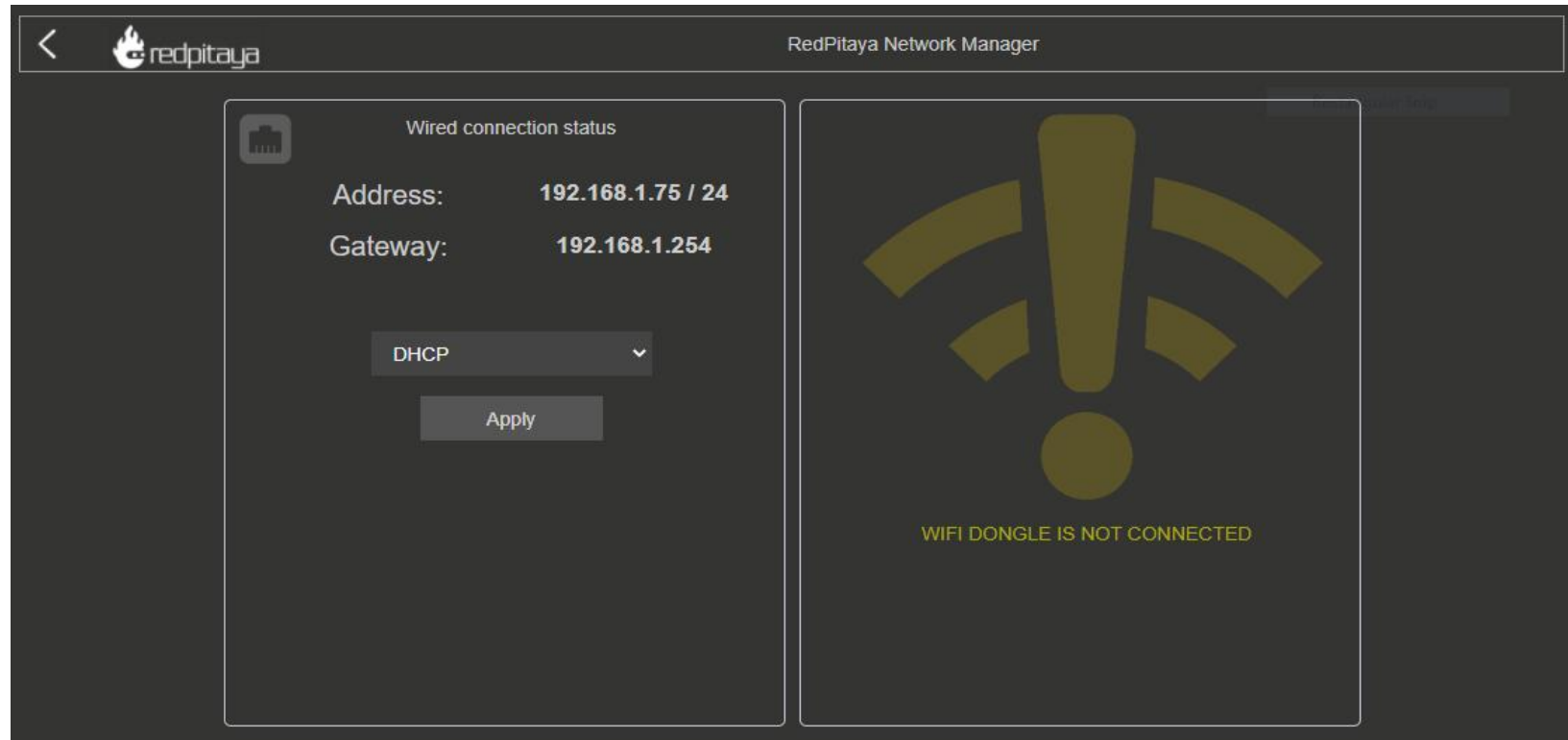
Van domeinnaam naar IP-adres met Putty



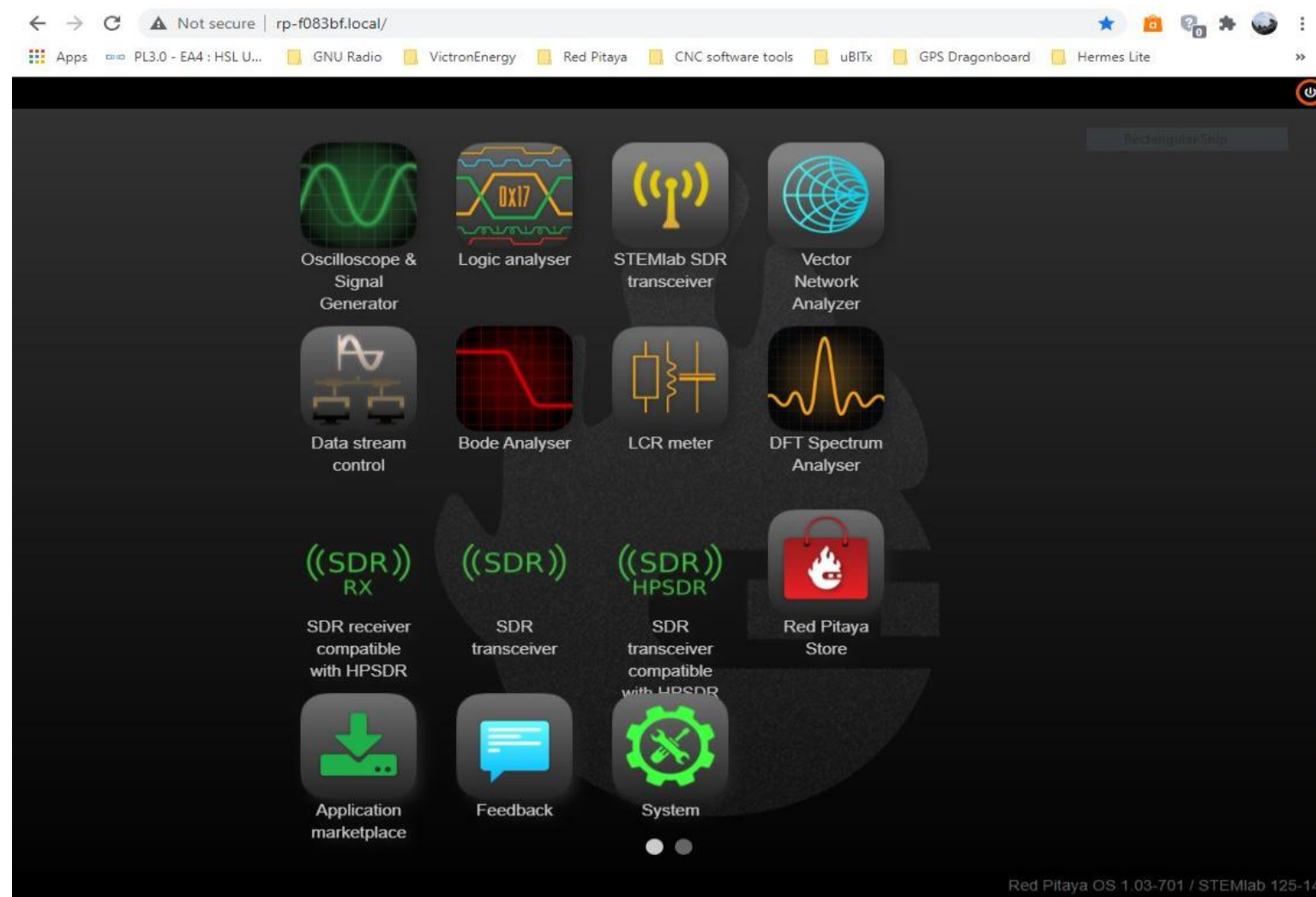
Van domeinnaam naar IP-adres met Putty

```
root@rp-f083bf: ~  
login as: root  
root@rp-f083bf.local's password:  
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.9.0-xilinx armv7l)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
#####  
# Red Pitaya GNU/Linux Ecosystem  
# Version: 1.03  
# Build: 701  
# Branch:  
# Commit: 9ecef3d05607fe99dlf57eef9d100438c827ce22  
# U-Boot: "redpitaya-v2016.4"  
# Linux Kernel: "branch-redpitaya-v2017.2"  
# Pro Applications: ""  
#####  
Last login: Mon Jan 18 19:41:09 2021 from 192.168.1.58  
root@rp-f083bf:~# ip addr  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
    inet6 ::1/128 scope host  
        valid_lft forever preferred_lft forever  
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000  
    link/ether 00:26:32:f0:83:bf brd ff:ff:ff:ff:ff:ff  
    inet 169.254.225.211/16 brd 169.254.255.255 scope link eth0  
        valid_lft forever preferred_lft forever  
    inet 192.168.1.75/24 brd 192.168.1.255 scope global dynamic eth0  
        valid_lft 81568sec preferred_lft 81568sec  
3: sit0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default qlen 1  
    link/sit 0.0.0.0 brd 0.0.0.0  
root@rp-f083bf:~#
```

Webmenu – System – Network Manager



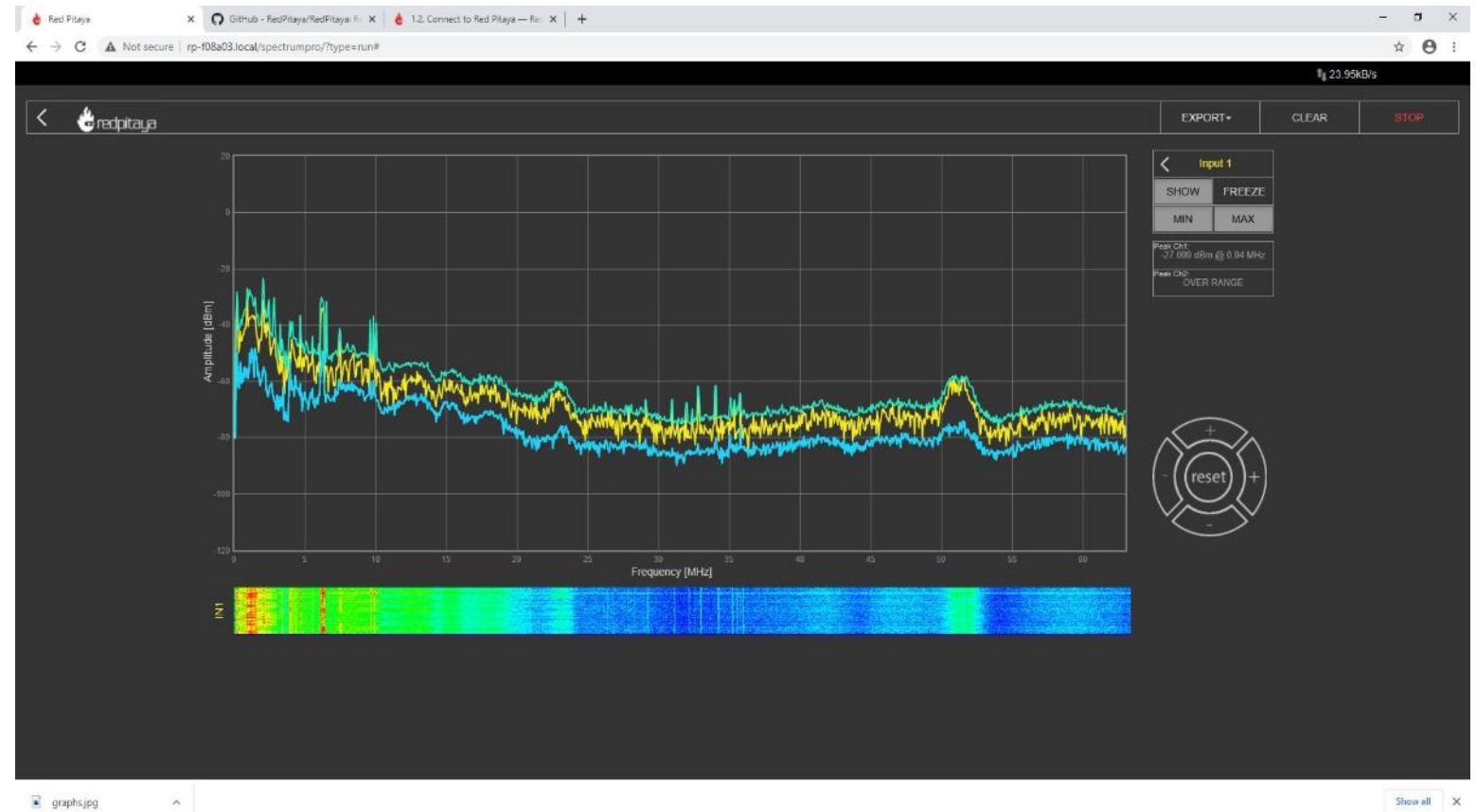
Bediening via de webbrowser



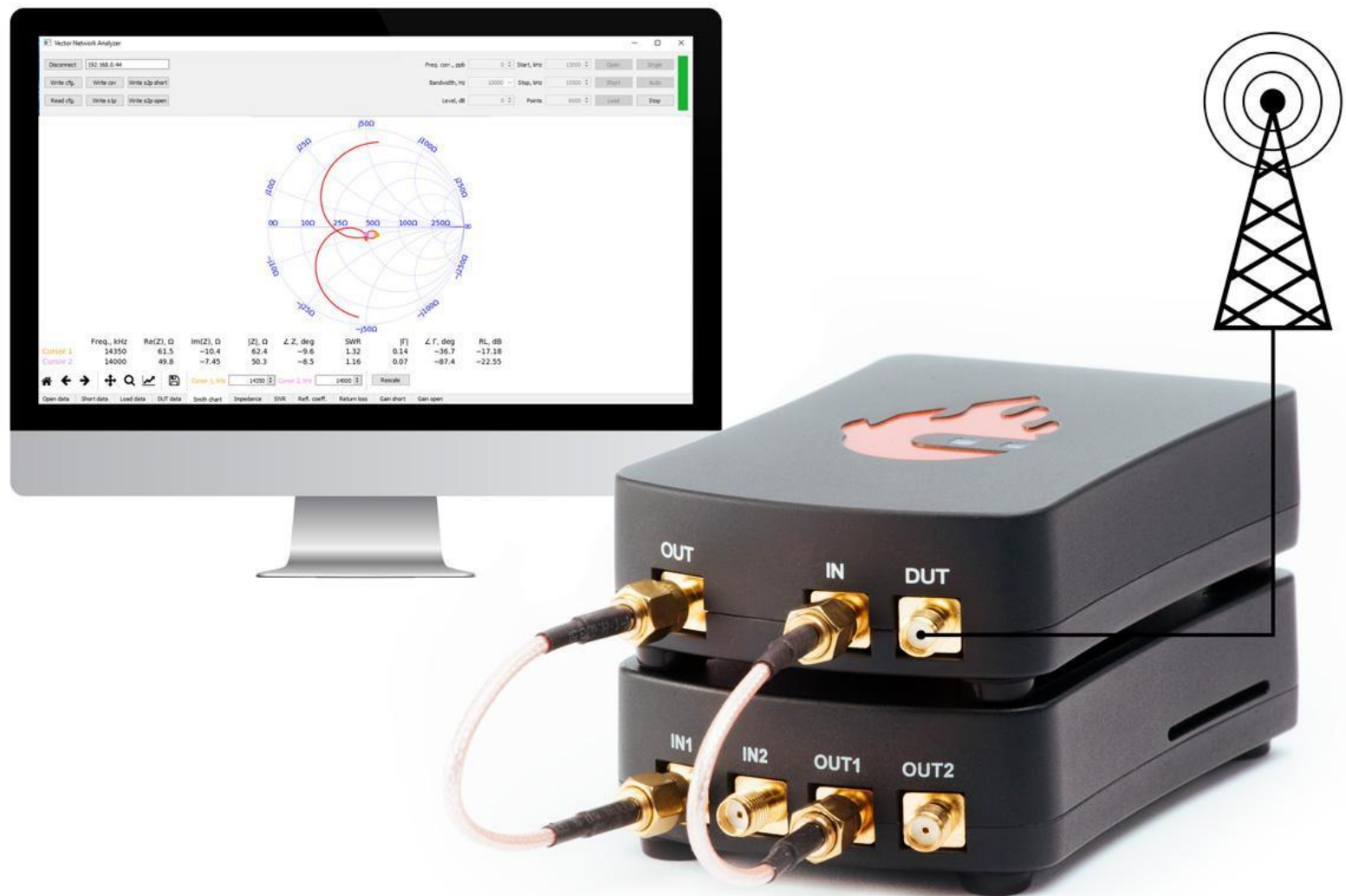
Toepassingen als meetinstrument

Red Pitaya = 2 x HF in, 2 x HF uit, ADC/DAC, CPU, FPGA

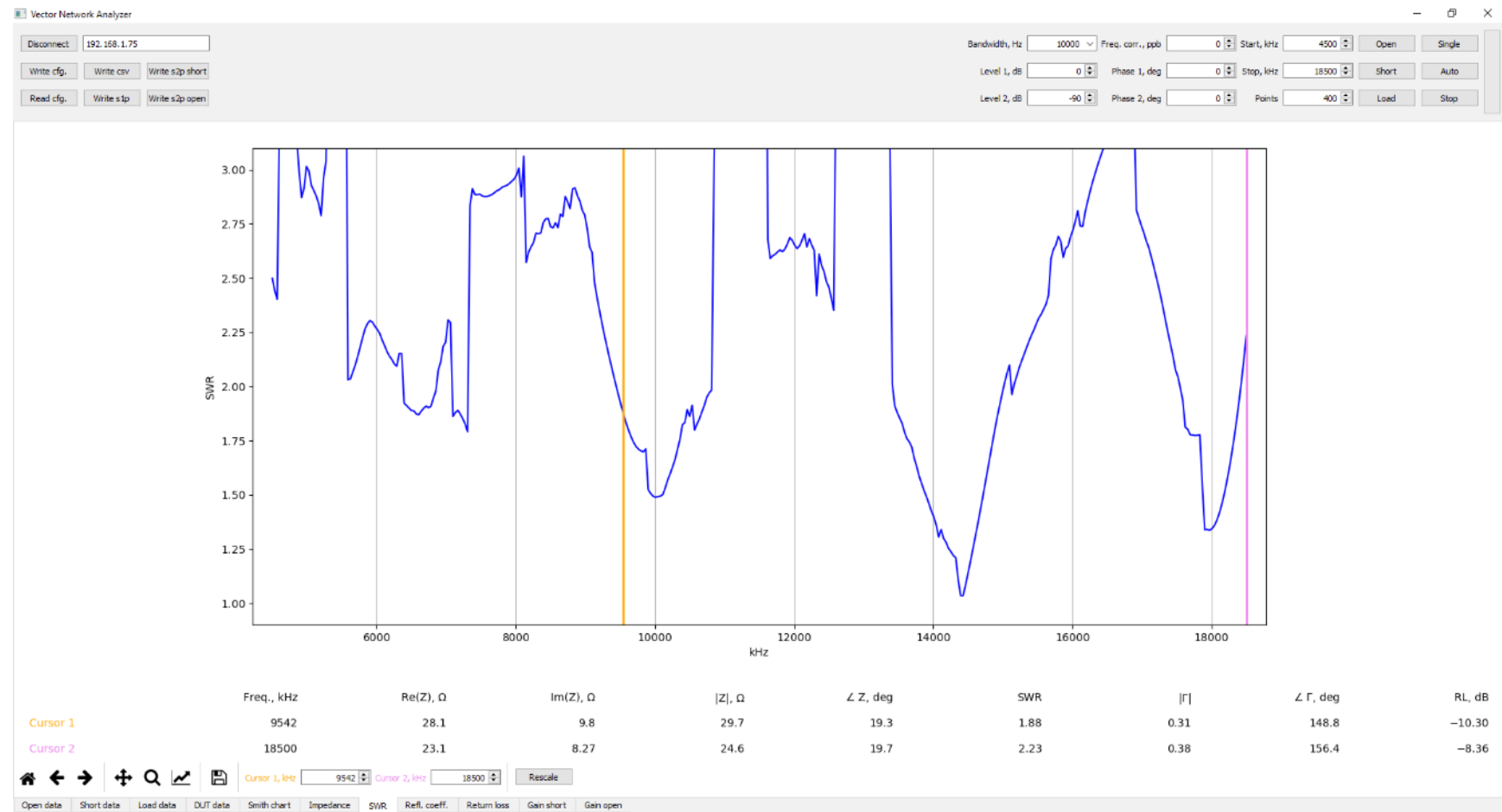
- Signaal generator
- Oscilloscoop
- Spectrum analysator
- VNA
-



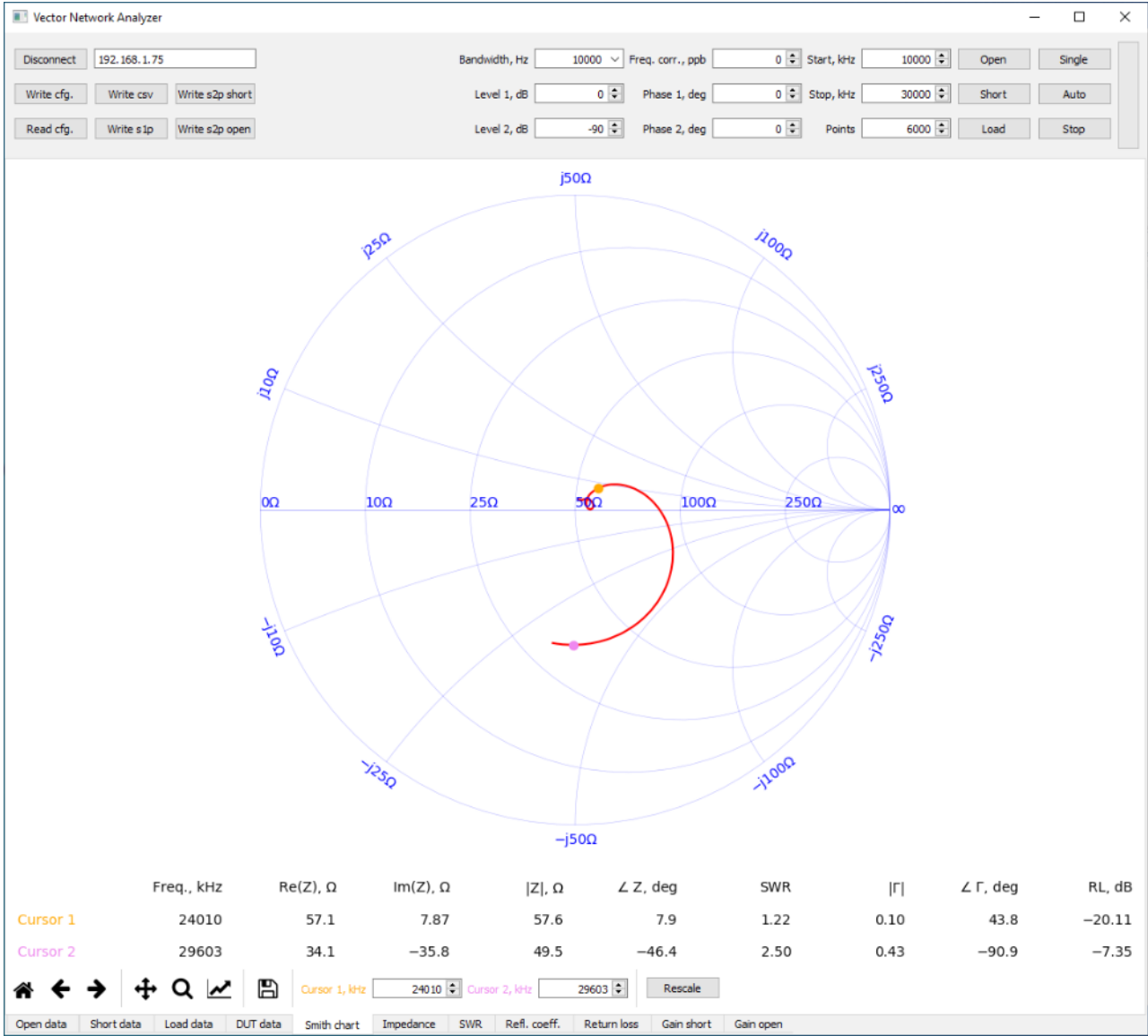
Red Pitaya VNA



Red Pitaya VNA

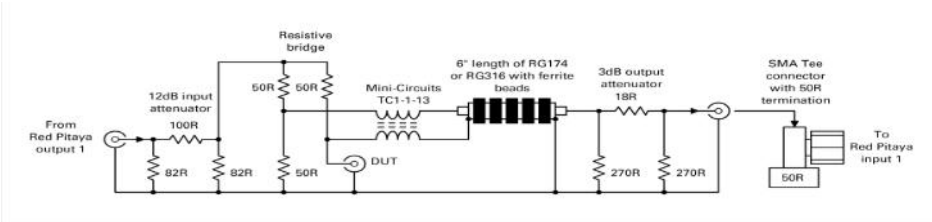
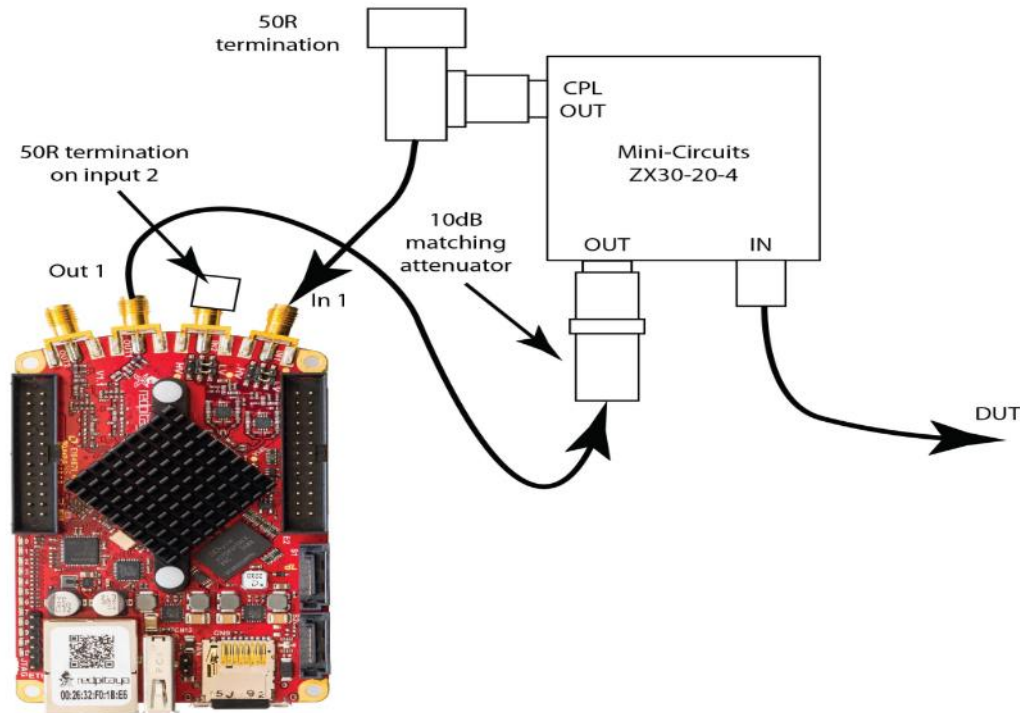


Red Pitaya VNA



Instrumentatie - VNA

VNA toepassing alleen bruikbaar met extra hardware (source: <https://github.com/pavel-demin/red-pitaya-notes/issues/250>)
 Red Pitaya VNA Antenna measurement setup



Wensen van radioamateurs

Onze radio leeft in de 50 Ohm wereld en is onderdeel van het systeem

Meetinstrumenten leven in de “ik mag de schakeling waar ik aan meet niet beïnvloeden” wereld

SDRlab en STEMLab zijn daarom zo verschillend

In de praktijk merk je niet zo veel van die hardware verschillen

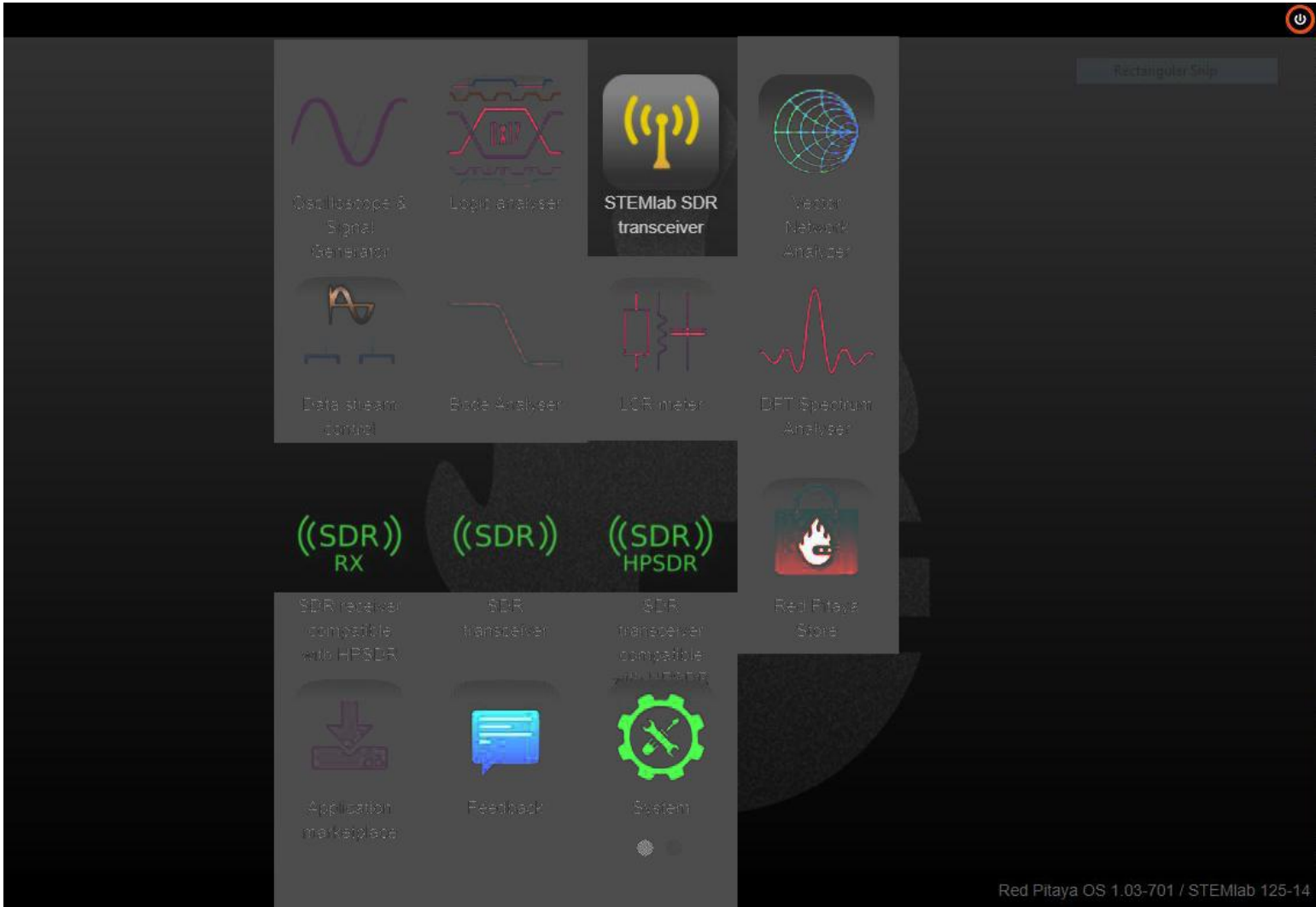
En je kunt eenvoudig aanpassingen doen voor ingangs- en uitgangsimpedantie

Geef mij maar een radio

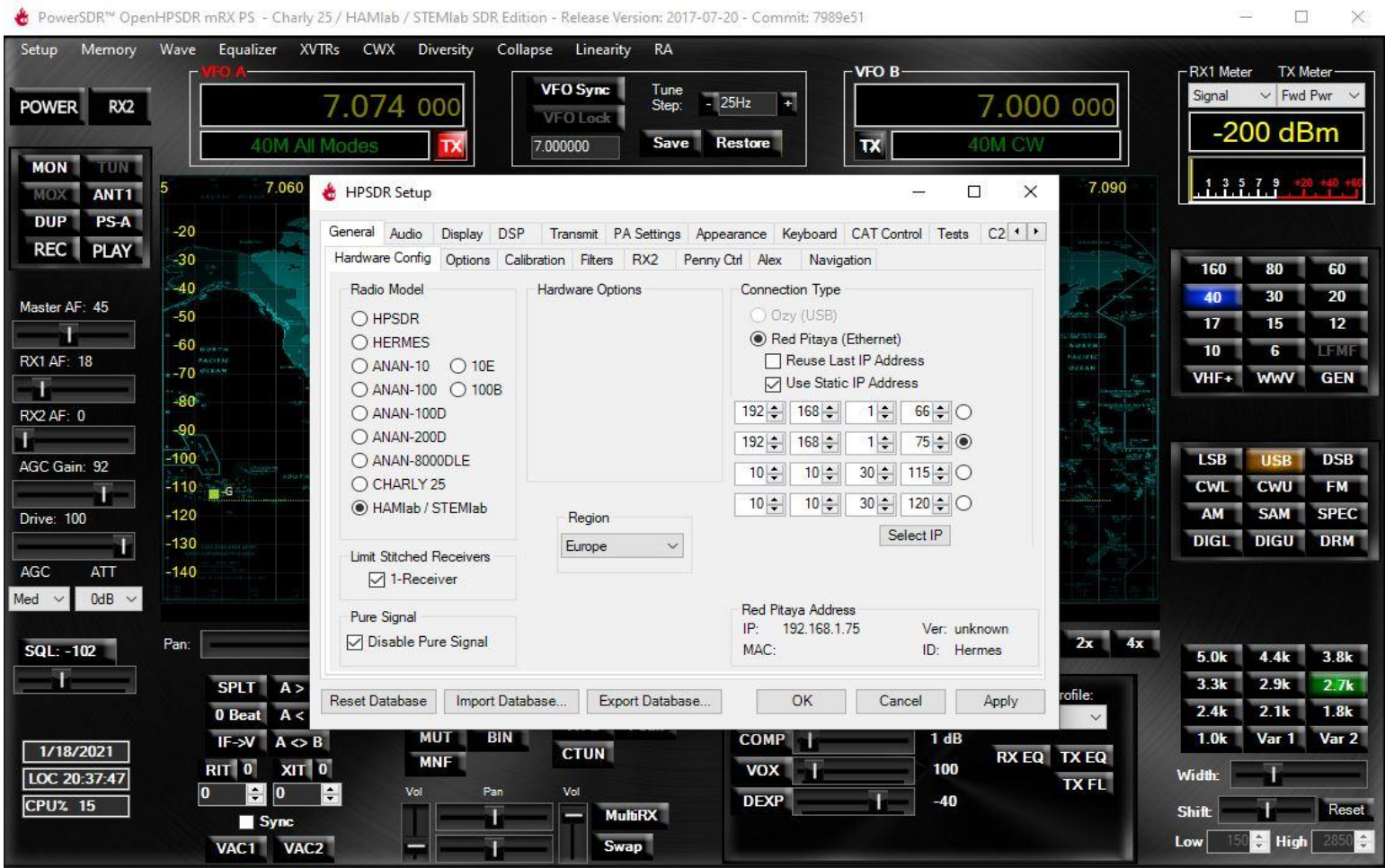
STEMlab met de Red Pitaya software en 3 extra applicaties geïnstalleerd vanaf de Market Place

1. STEMlab SDR Transceiver → PowerSDR STEMlab Edition
2. SDR RX – SDR receiver compatible with HPSSDR
3. SDR – SDR transceiver
4. SDR HPSSDR – SDR transceiver compatible with HPSSDR

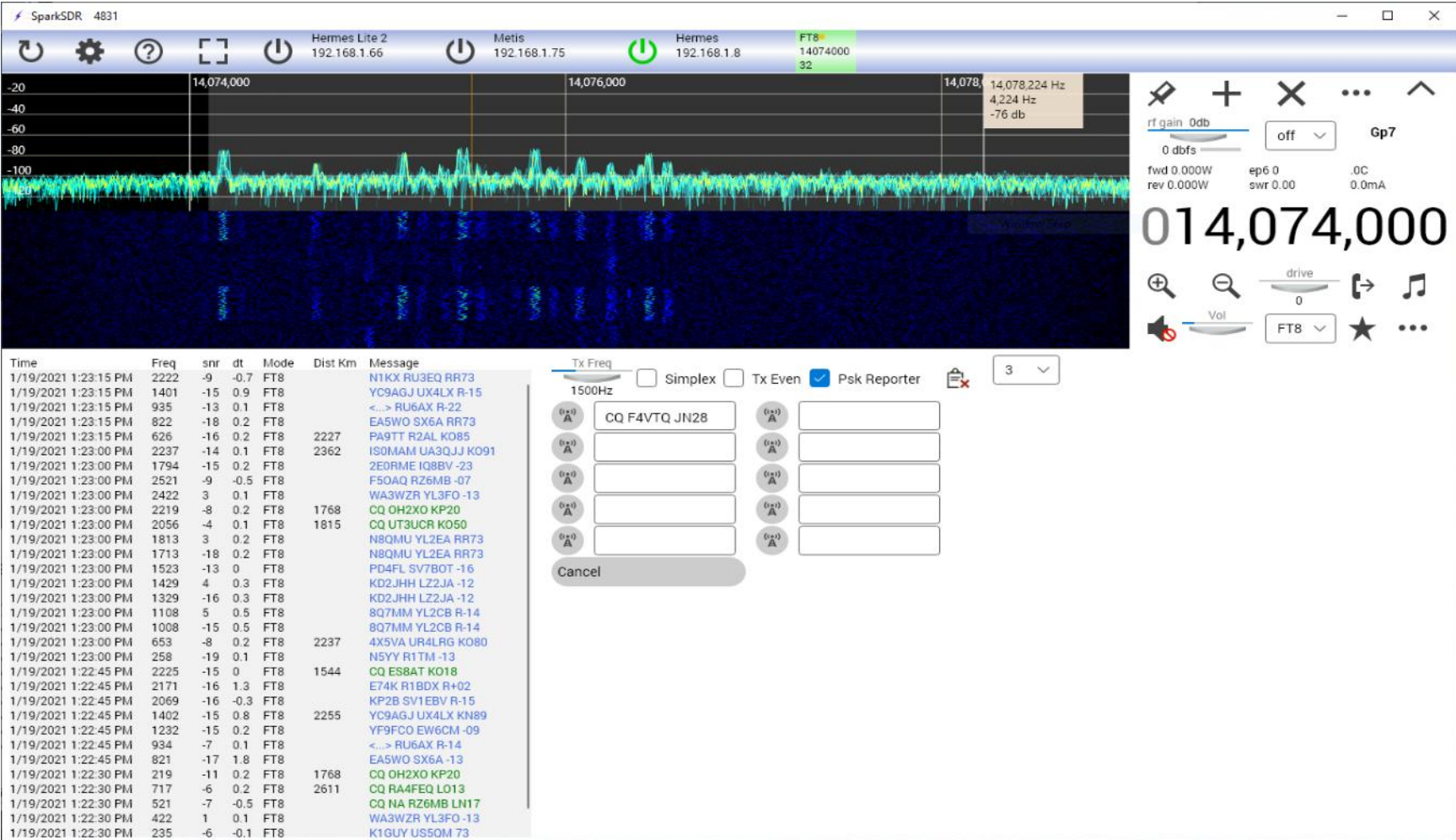
Gevonden op de Application Marketplace



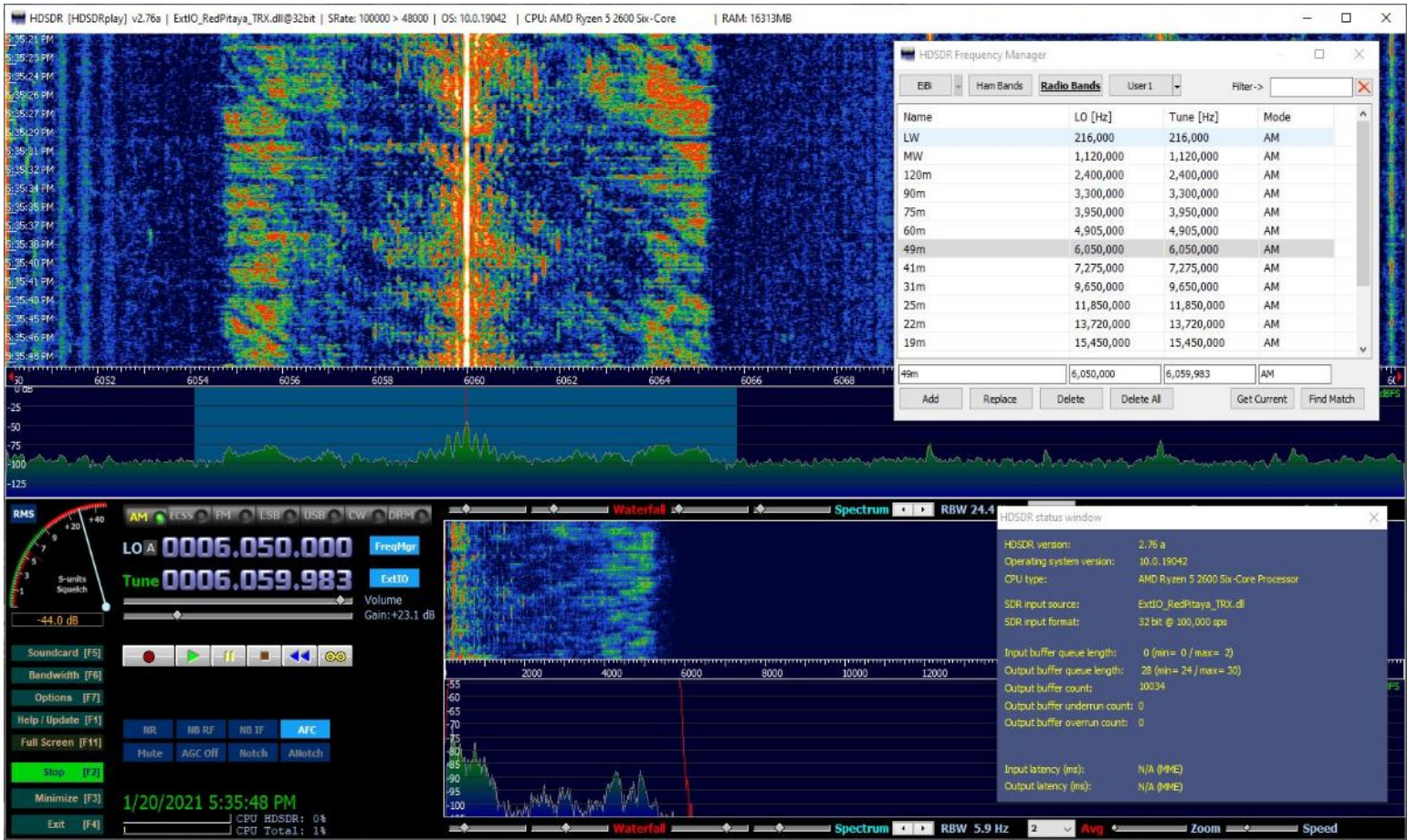
Start en gebruik SDR software



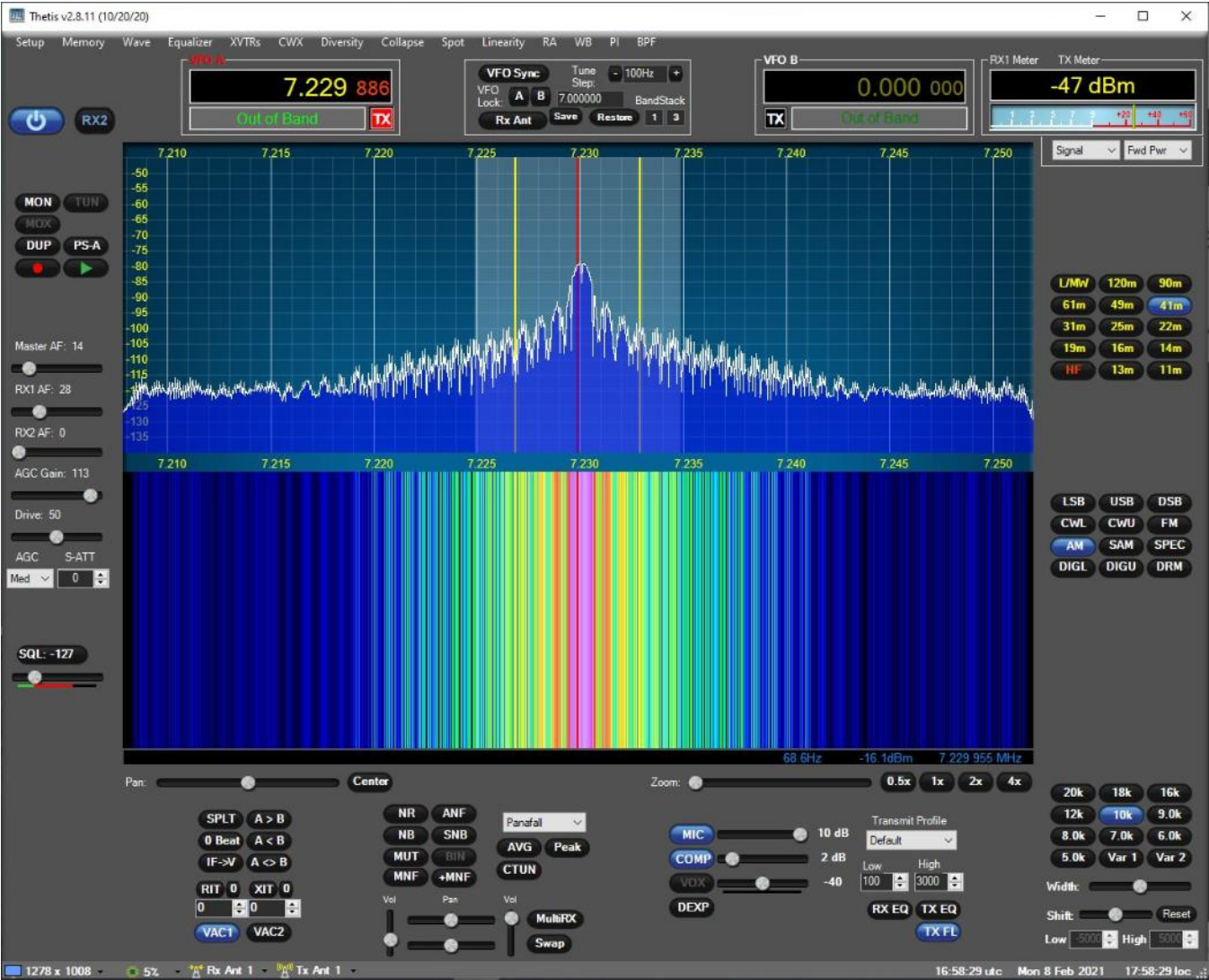
SparkSDR SDRlab FT8



STEMlab met HDSDRplay



STEMlab met Thetis / openHPSDR



Instrumentatie

Gebaseerd op bestaande toepassingen zoals oscilloscoop en signaal generator

T&M Instruments







Oscilloscope

Signal generator

Spectrum analyzer

Bode analyzer

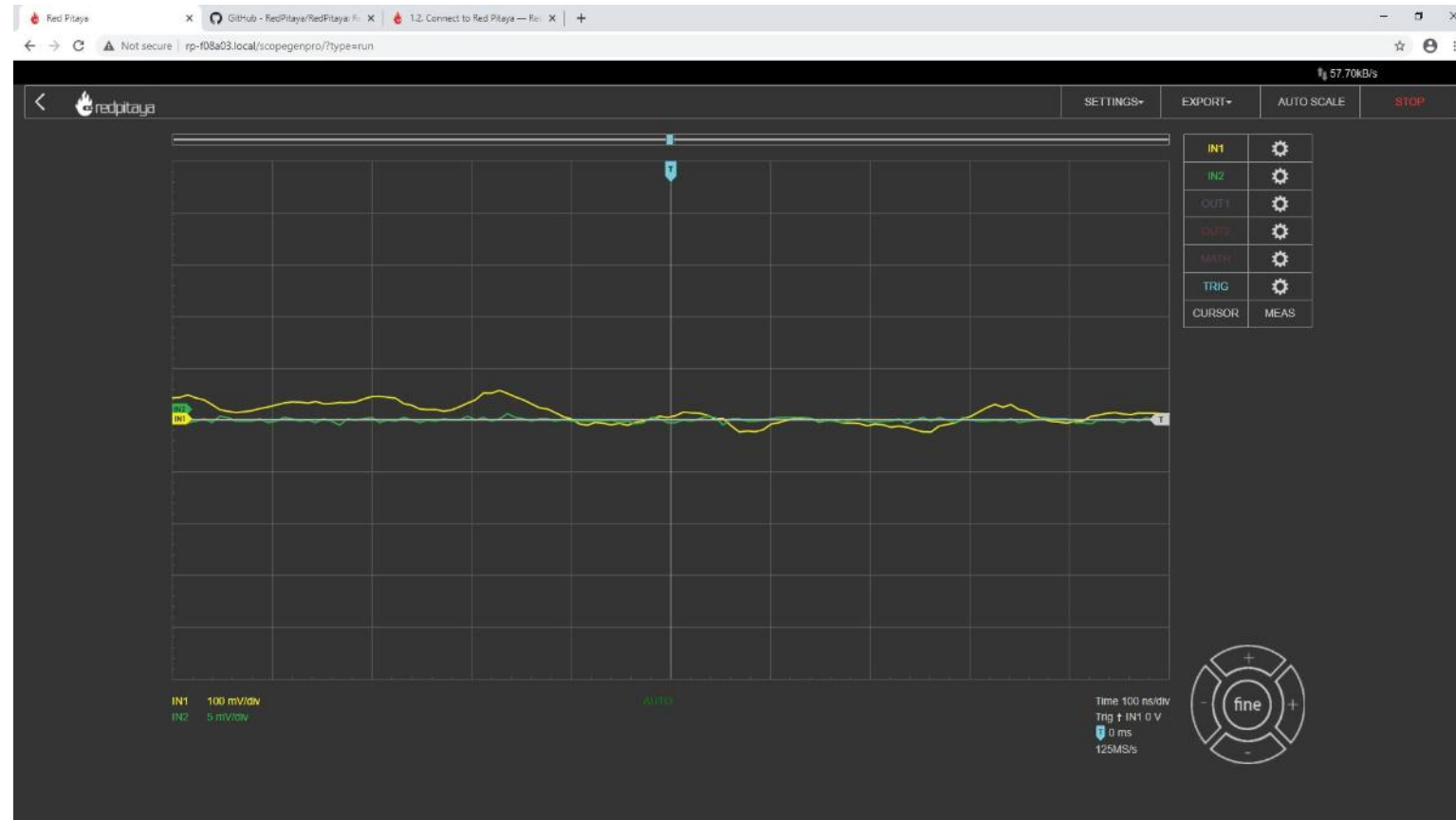
Logic analyzer Pro




LCR meter

Marketplace

Oscilloscope



Is het ook een goede radio ?

SDR

→ met welke SDR software beoordeel je dit en wat vind je belangrijk

Digitale modi

→ FT8

→ WSPR

→ Meerband

→ Ingeroosterd en “band hopping”

Gebruik vanaf afstand, besturing op afstand

Aantal ontvangers en bandbreedte

De bandbreedte hangt af van het programma op de PC :

ghpsdr3-alex	- 6 ontvangers	192 kHz
CW Skimmer Server	- 6 ontvangers	192 kHz
PowerSDR mRX PS	- 2 ontvangers	384 kHz
cudaSDR	- 6 ontvangers	384 kHz
GNU Radio	- 1 ontvanger	2.5 MHz
HDSDR, SDR#, Gqrx	- 2 ontvangers	1.25 MHz
Pothos SDR, CubicSDR		

<https://forum.redpitaya.com/viewtopic.php?t=265&start=360>

Speciale functies

Calibratie van de frequentie

Calibratie van het uitgangsvermogen

Pre-distortion (het vooraf vervormen van het uitgangssignaal ter compensatie van de vervorming daarna in de eindtrap)

Diversity (compenseren van ongewenste signalen door die uit het totale signaal te verwijderen)

Digitale filters

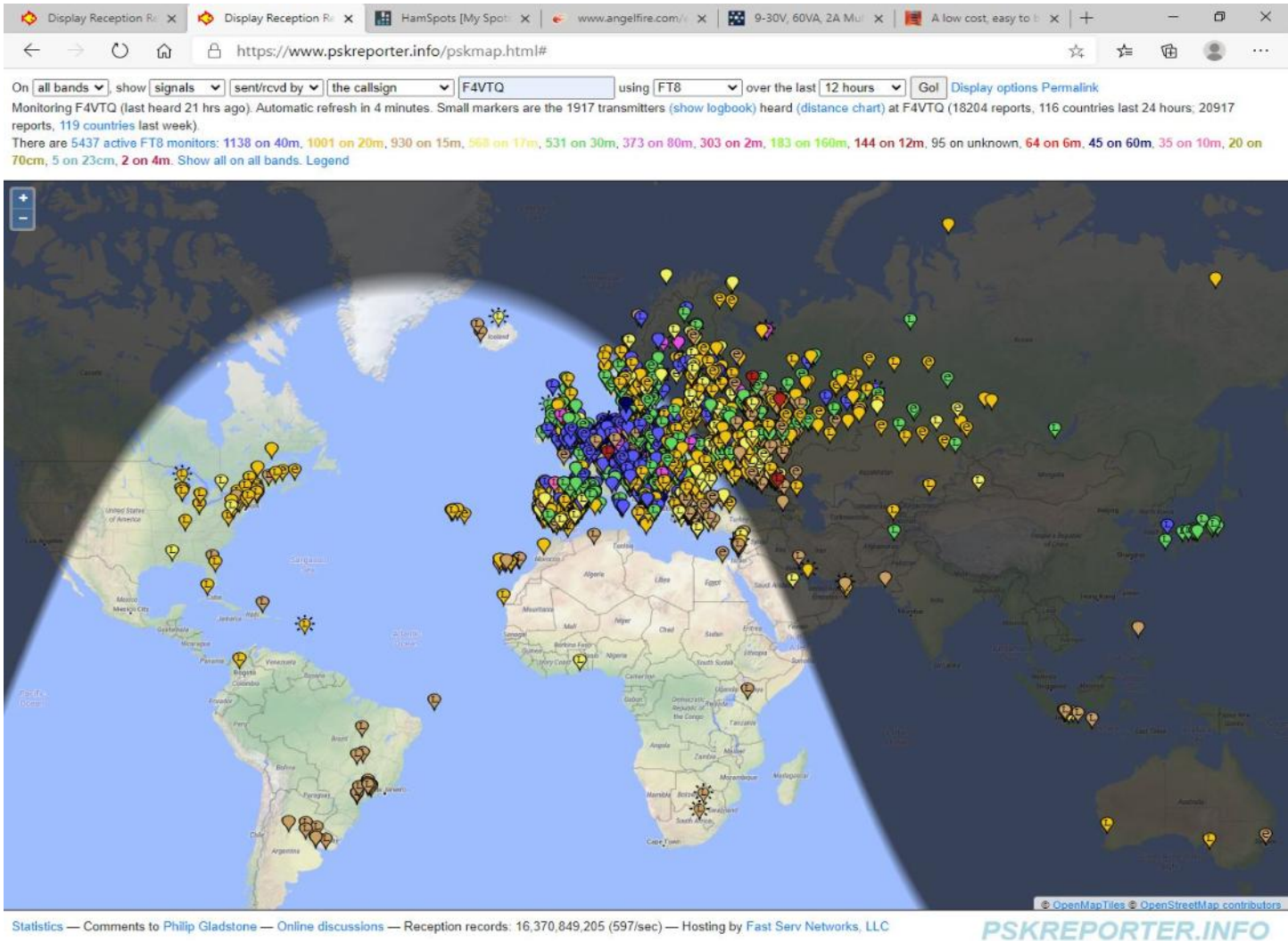
Extra software (plug ins)

FT8 Meerband

Gevoelig

Selectief

Stabiel



Het voordeel van een ethernet radio

In vergelijking met een traditionele of via serieel of usb gekoppelde radio / SDR

Besturing op afstand en overdracht van het signaal met de informatie (spraak, data, cw) !

Zonder tussenliggende, soms moeilijke, niet transparante oplossingen

- USB over Ethernet
- Via een Raspberry Pi of een laptop/pc
- CAT interface / audio kabels

Nuttige aanvullingen voor een radio

Hardware

- GPS
- Voorversterker met ARDUINO aansturing
- LPF en PA om er een “echte” transceiver van te maken



Software

- Kiezen van de antenne en frequenties gebaseerd op dag/nacht ritme



GPS

Waarom?

- Tijd synchronisatie
- Frequentie calibratie

Hoe?



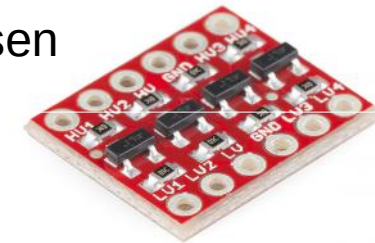
GPS interface

The PPS signal should be connected to the pin DIO3_N of the extension connector E1. The UART interface should be connected to the UART pins of the extension connector E2.

The measurement and correction of the frequency deviation is disabled by default and should be enabled by uncommenting the following line in ft8.cron:

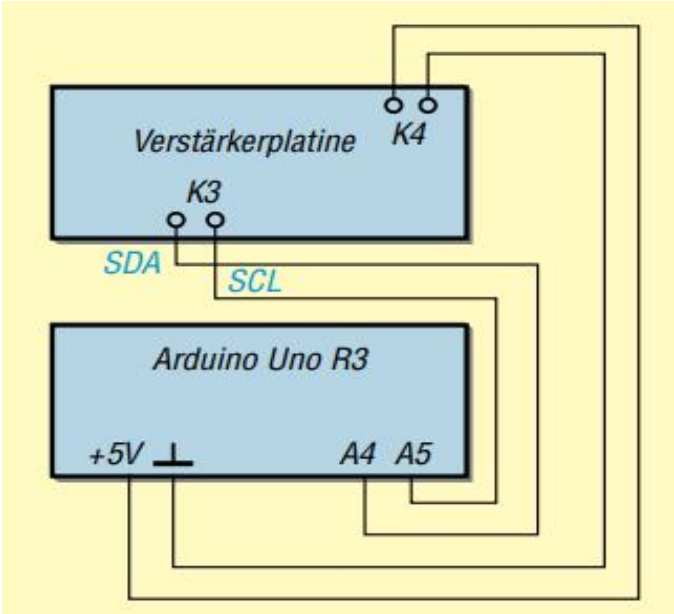
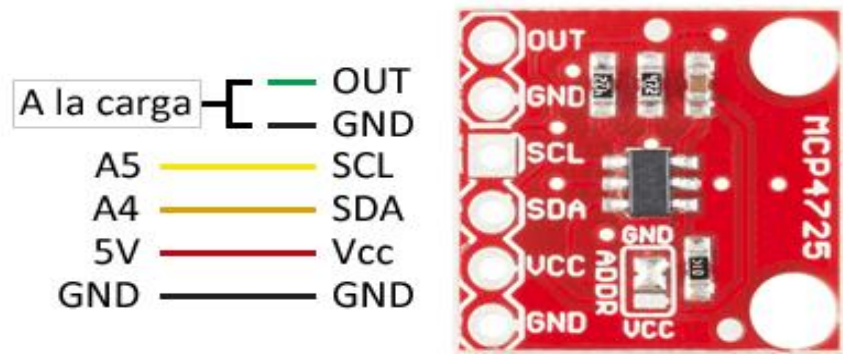
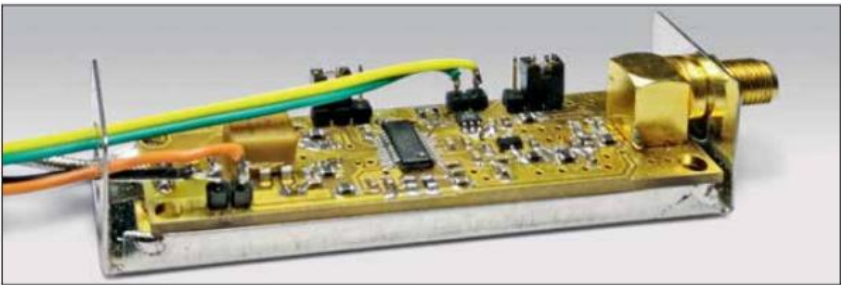
```
***** cd /dev/shm && /media/mmcblk0p1/apps/sdr_transceiver_ft8/update-corr.sh
```

Pas op: 3,3V is niet hetzelfde als 5V → misschien moet er een “level shifter” tussen



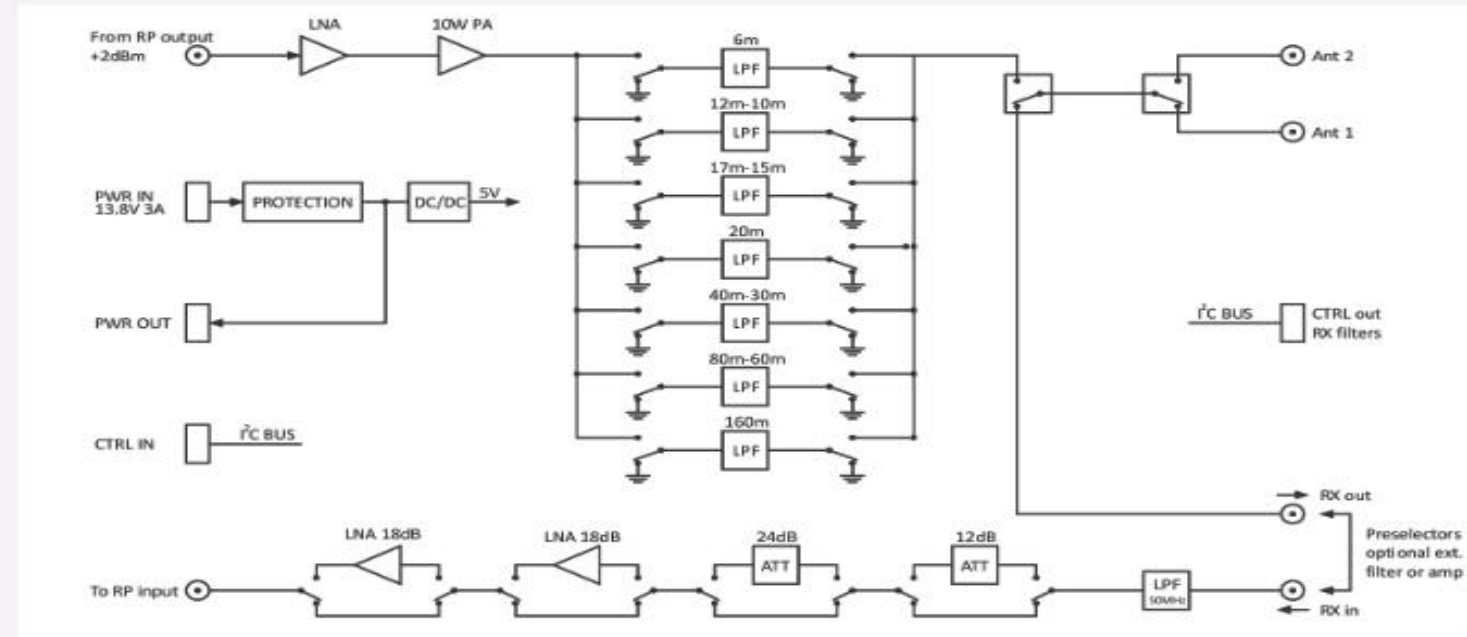
Voorversterker

FUNKAMATEUR voorversterker met een I²C MCP4725 en een kleine Arduino sketch

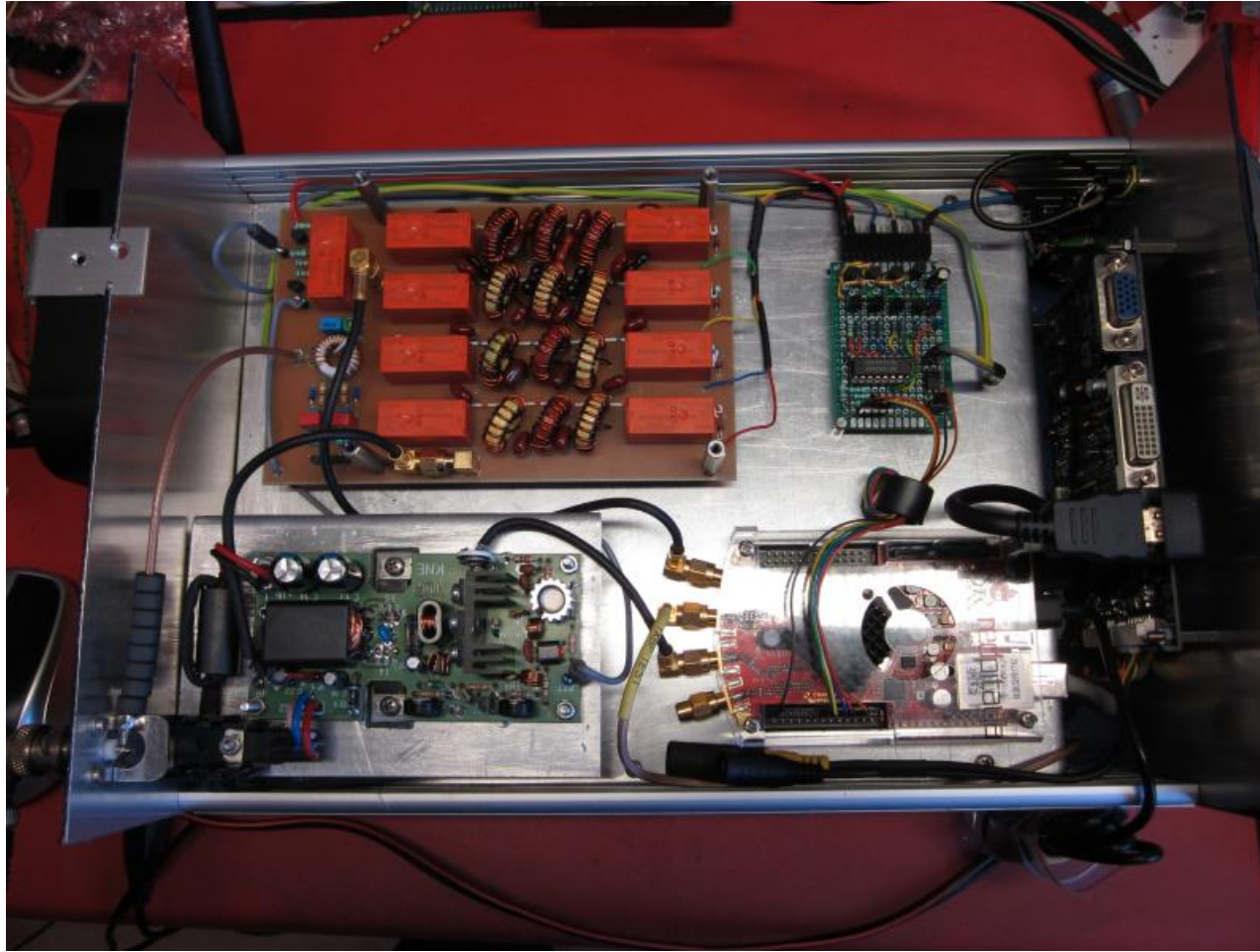


Zenden met een schoon signaal

SDR block diagram



Transceiver voorbeeld PA0EJH



qrz.com - PA0EJH

Software speelterrein

Je gebruikt een aantal eenvoudige Linux bevelen

Als je iets kapot hebt gemaakt, zet je een opgeslagen versie die wel werkte terug

PuTTY en WinSCP geven toegang tot het Linux gedeelte

Begin met te begrijpen hoe de opzet is met .py en .sh scripts en cron

De crontab guru kan je helpen <https://crontab.guru/>

>> Er zijn verschillen tussen de software van Red Pitaya zelf en de Alpine Linux software van Pavel Demin en ook tussen de 2 Red Pitaya varianten STEMLab en SDRlab

Software blokken

Voor ingewikkelde maar langzame processen : Linux ARM processor → Linux, shell script, python, C, C++

De gebruikersinterface : javascript of html

Verbinding tussen de front end en de back end : webserver

Digitale signaal bewerking : FPGA

Red Pitaya basis software

```
root@rp-f083bf:~# ls -ail
total 36
1035 drwx----- 6 root root 4096 Jan 25 12:00 .
2 drwxr-xr-x 21 root root 4096 Jun 18 2020 ..
5573 -rw----- 1 root root 125 Jan 19 21:14 .bash_history
1036 -rw-r--r-- 1 root root 3106 Oct 22 2015 .bashrc
5583 drwx----- 2 root root 4096 Sep 1 2017 .cache
8849 drwxrwxr-x 3 root root 4096 Jan 25 12:00 .config
31345 drwxr-xr-x 2 root root 4096 Sep 1 2017 .jupyter
5593 drwx----- 3 root root 4096 Sep 1 2017 .local
1037 -rw-r--r-- 1 root root 148 Aug 17 2015 .profile
root@rp-f083bf:~# pwd
/root
root@rp-f083bf:~# cd /
root@rp-f083bf:/# ls -ail
total 716
2 drwxr-xr-x 21 root root 4096 Jun 18 2020 .
2 drwxr-xr-x 21 root root 4096 Jun 18 2020 ..
13 drwxr-xr-x 2 root root 4096 Sep 1 2017 bin
1 drwxr-xr-x 10 root root 16384 Jan 1 1970 boot
12 -rw-r--r-- 1 root root 633371 Sep 1 2017 buildlog.txt
3 drwxr-xr-x 11 root root 3040 Jan 18 22:37 dev
189 drwxr-xr-x 74 root root 4096 Sep 1 2017 etc
482 drwxr-xr-x 5 root root 4096 Sep 1 2017 home
483 drwxr-xr-x 11 root root 4096 Sep 1 2017 lib
11 drwx----- 2 root root 16384 Sep 1 2017 lost+found
1031 drwxr-xr-x 2 root root 4096 Aug 1 2017 media
1032 drwxr-xr-x 2 root root 4096 Aug 1 2017 mnt
1033 drwxr-xr-x 3 root root 4096 Sep 1 2017 opt
1 dr-xr-xr-x 102 root root 0 Jan 1 1970 proc
1035 drwx----- 6 root root 4096 Jan 25 12:00 root
1127 drwxr-xr-x 13 root root 400 Jan 27 12:31 run
1045 drwxr-xr-x 2 root root 4096 Sep 1 2017 sbin
1127 drwxr-xr-x 2 root root 4096 Aug 1 2017 srv
1 dr-xr-xr-x 12 root root 0 Jan 1 1970 sys
1129 drwxrwxrwt 10 root root 4096 Jan 24 17:29 tmp
1130 drwxr-xr-x 10 root root 4096 Aug 1 2017 usr
4952 drwxr-xr-x 11 root root 4096 Aug 1 2017 var
```

Pavel Demin Alpine Linux

```
rp-f08a03:~# ls -ail
total 8
 491 drwx----- 2 root    root    120 Jan 27 12:27 .
 418 drwxr-xr-x 21 root    root    420 May 29 2020 ..
1711388 -rw----- 1 root    root    16 Jan 27 12:27 .ash_history
 492 -rw-r--r-- 1 root    root    96 Jun 28 2020 .profile
 520 lrwxrwxrwx 1 root    root    21 Jan 1 1970 apps -> /media/mmcblk0p1/apps
 519 lrwxrwxrwx 1 root    root    21 Jan 1 1970 wifi -> /media/mmcblk0p1/wifi
rp-f08a03:~# pwd
/root
rp-f08a03:~# cd /
rp-f08a03:/# ls -ail
total 0
 418 drwxr-xr-x 21 root    root    420 May 29 2020 .
 418 drwxr-xr-x 21 root    root    420 May 29 2020 ..
 392 drwxr-xr-x 3 root    root    30 Jun 28 2020 .modloop
3248 drwxrwxrwt 4 root    root    80 Jan 1 1970 .modoverlayfs
1180 drwxr-xr-x 2 root    root    1740 Jan 1 1970 bin
 3 drwxr-xr-x 8 root    root    2960 Jan 23 09:22 dev
 422 drwxr-xr-x 30 root    root    1300 May 29 2020 etc
 584 drwxr-xr-x 2 root    root    40 Jan 1 1970 home
 424 drwxr-xr-x 9 root    root    400 Jan 1 1970 lib
 590 drwxr-xr-x 6 root    root    120 Jan 1 1970 media
 594 drwxr-xr-x 2 root    root    40 Jan 1 1970 mnt
 595 drwxr-xr-x 2 root    root    40 Jan 1 1970 opt
 1 dr-xr-xr-x 91 root    root    0 Jan 1 1970 proc
 491 drwx----- 2 root    root    120 Jan 27 12:27 root
2751 drwxr-xr-x 8 root    root    380 Jan 27 12:25 run
 540 drwxr-xr-x 2 root    root    2680 Jan 1 1970 sbin
 598 drwxr-xr-x 2 root    root    40 Jan 1 1970 srv
 1 dr-xr-xr-x 12 root    root    0 Jan 1 1970 sys
 419 drwxrwxrwt 8 root    root    160 Jan 27 12:25 tmp
 541 drwxr-xr-x 9 root    root    180 Jan 1 1970 usr
 427 drwxr-xr-x 11 root    root    260 May 29 2020 var
```

Red Pitaya basis – geïnstalleerde programma's

```
root@rp-f083bf: /boot/www/apps/sdr_transceiver
root@rp-f083bf:/boot/fpga# cd ../www
-bash: cd: ../www: No such file or directory
root@rp-f083bf:/boot/fpga# cd ../www
root@rp-f083bf:/boot/www# ls
apps  client_body_temp  conf
root@rp-f083bf:/boot/www# cd apps
root@rp-f083bf:/boot/www/apps# ls
assets                index.html            sdr_transceiver
ba_pro                info                  sdr_transceiver_hpsdr
error_bazaar_access.html  jupyter_manager      spectrumpro
error_bazaar_install.html  la_pro                stemlab_sdr_transceiver_hpsdr
error_bazaar_proto.html   lcr_meter             stemlab_vna
favicon.ico             network_manager       streaming_manager
idfile.id               scopegenpro           style.css
images                  scpi_manager          ui
impedance_analyzer       sdr_receiver_hpsdr    updater
root@rp-f083bf:/boot/www/apps# cd sdr_transceiver
-bash: cd: sdr_transceiver: No such file or directory
root@rp-f083bf:/boot/www/apps# cd sdr_transceiver
root@rp-f083bf:/boot/www/apps/sdr_transceiver# ls
controllerhf.so  img                js                src
css              index.html         sdr-transceiver  start.sh
fpga.conf        info              sdr_transceiver.bit  stop.sh
root@rp-f083bf:/boot/www/apps/sdr_transceiver#
```

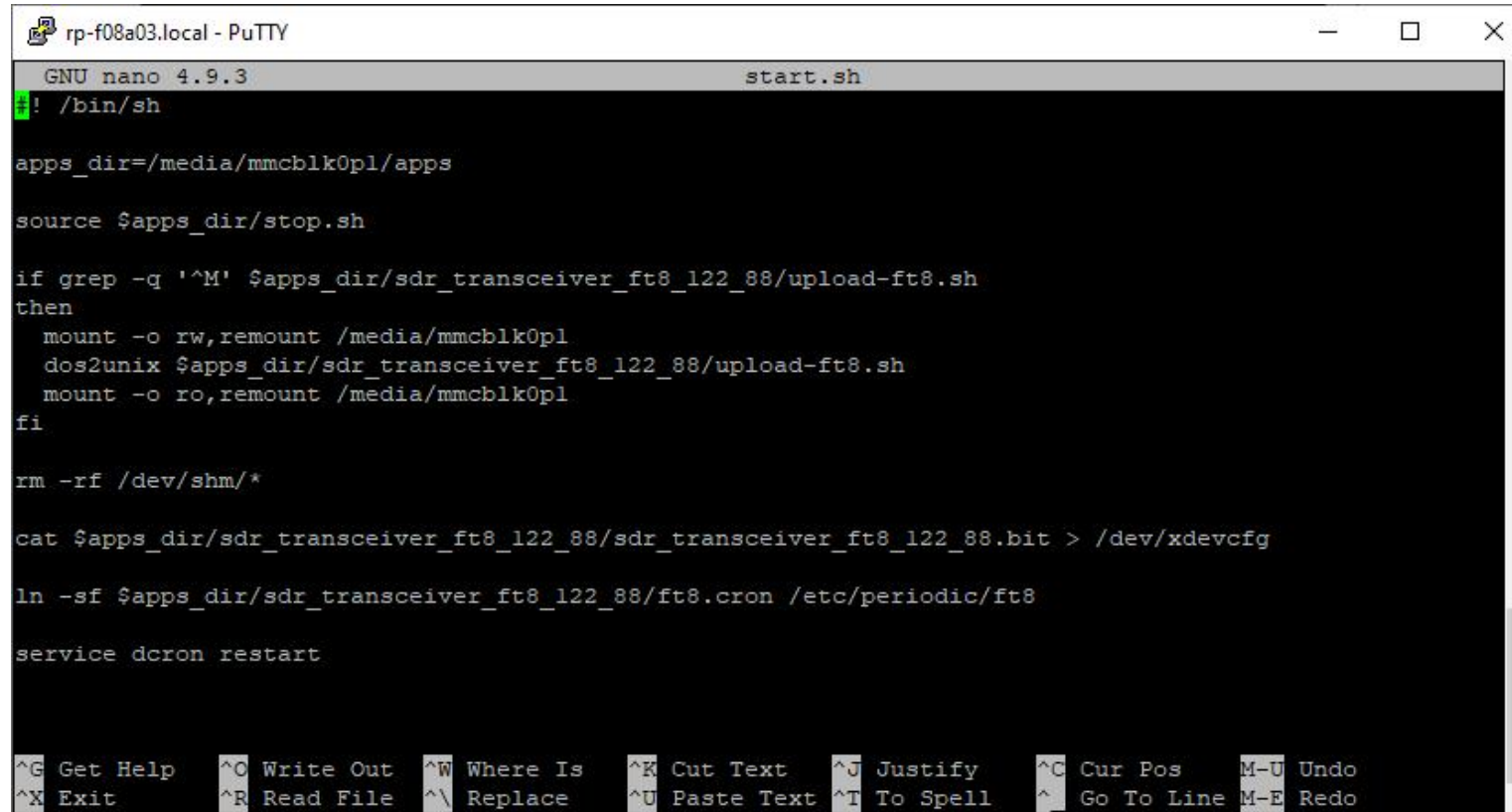

Pavel Demin Alpine Linux – geïnstalleerde toepassingen

```
rp-f08a03.local - PuTTY
rp-f08a03:~# cd /apps
-ash: cd: can't cd to /apps: No such file or directory
rp-f08a03:~# ls
apps  wifi
rp-f08a03:~# cd apps
rp-f08a03:/media/mmcblk0p1/apps# ls
css                               sdr_transceiver_ft8
ft8d                             sdr_transceiver_ft8_122_88
index.html                       sdr_transceiver_hpsdr
index_122_88.html                sdr_transceiver_hpsdr_122_88
led_blinker                      sdr_transceiver_wide
led_blinker_122_88              sdr_transceiver_wspr
mcpha                            sdr_transceiver_wspr_122_88
sdr_receiver_hpsdr              server
sdr_receiver_hpsdr_122_88       stop.sh
sdr_receiver_wide_122_88        vna
sdr_transceiver                 vna_122_88
sdr_transceiver_emb             wsprd
rp-f08a03:/media/mmcblk0p1/apps# cd sdr_transceiver
rp-f08a03:/media/mmcblk0p1/apps/sdr_transceiver# ls
Makefile          sdr-transceiver.c    stop.sh
index.html         sdr_transceiver.bit
sdr-transceiver    start.sh
rp-f08a03:/media/mmcblk0p1/apps/sdr_transceiver#
```


Plek voor start.sh

```
rp-f08a03.local - PuTTY
rp-f08a03:/media# ls
cdrom      floppy      mmcblk0p1  usb
rp-f08a03:/media# cd mmcblk0p1
rp-f08a03:/media/mmcblk0p1# ls
AUTOEXEC.BAT
AlpineRedPitaya-SDRlab-3.12-20200628.txt
COMMAND.COM
KERNEL.SYS
LOCALE
System Volume Information
apps
autorun.ico
autorun.inf
boot.bin
cache
devicetree.dtb
modloop
red-pitaya.apkov1.tar.gz
start.sh
uEnv.txt
uImage
uInitrd
wifi
rp-f08a03:/media/mmcblk0p1#
```

De inhoud van start.sh



```
rp-f08a03.local - PuTTY
GNU nano 4.9.3 start.sh
#!/bin/sh

apps_dir=/media/mmcb1k0p1/apps

source $apps_dir/stop.sh

if grep -q '^M' $apps_dir/sdr_transceiver_ft8_122_88/upload-ft8.sh
then
    mount -o rw,remount /media/mmcb1k0p1
    dos2unix $apps_dir/sdr_transceiver_ft8_122_88/upload-ft8.sh
    mount -o ro,remount /media/mmcb1k0p1
fi

rm -rf /dev/shm/*

cat $apps_dir/sdr_transceiver_ft8_122_88/sdr_transceiver_ft8_122_88.bit > /dev/xdevcfg

ln -sf $apps_dir/sdr_transceiver_ft8_122_88/ft8.cron /etc/periodic/ft8

service dcron restart

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text   ^J Justify    ^C Cur Pos    M-U Undo
^X Exit      ^R Read File  ^\ Replace   ^U Paste Text ^T To Spell   ^_ Go To Line  M-E Redo
```

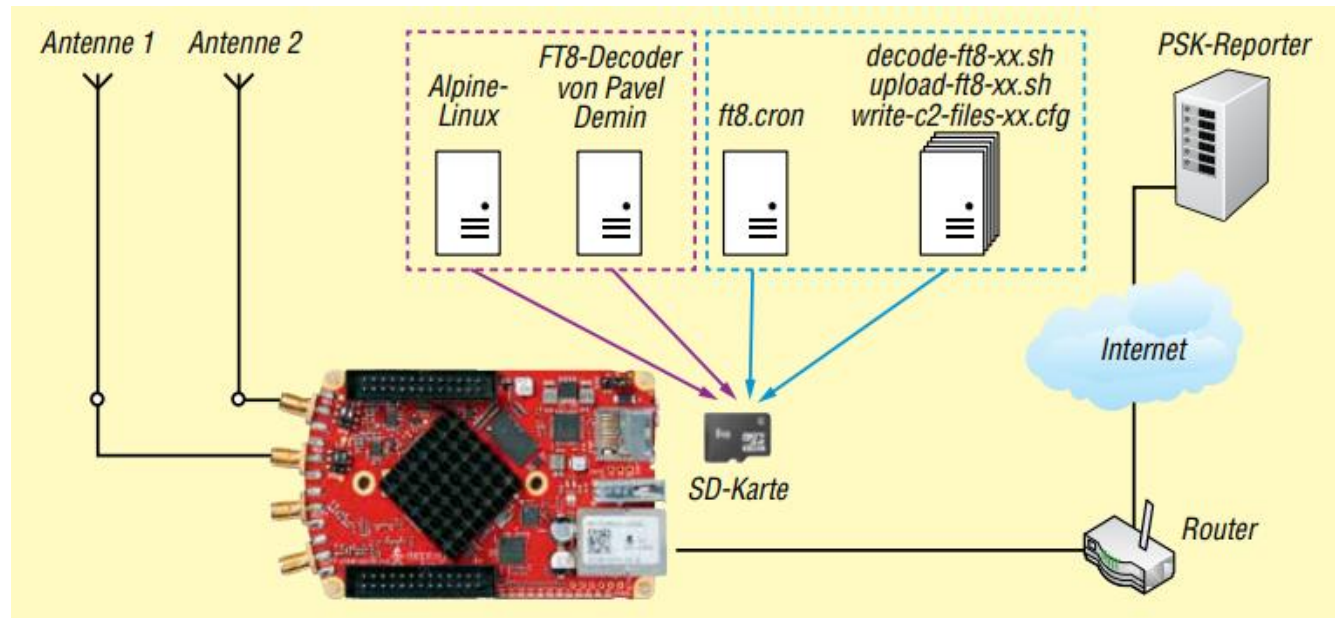
Geknutsel aan cron and .sh - FT8 meerband ontvanger

In .cron vinden we verwijzing naar decode.sh en upload.sh

Decode leest gegevens uit de write.cfg files

Maar er passen maar 8 banden tegelijkertijd in een STEMLab, en niet de 10 HF banden

Met de SDRlab werd het leven al weer leuker met maximaal 16 ontvangers parallel



Antenne ingang 1 Antenne ingang 2 & dag/nacht ritme

Benötigte Dateien (aus dem Download-Bereich auf www.funkamateurl.de)		
Antenne 1	Tag	Nacht
	ft8.cron: Stundenfestlegung für Tag Minuten-Muster für Kanal 1	ft8.cron: Stundenfestlegung für Nacht Minuten-Muster für Kanal 1
	decode-ft8-1D.sh: verwendet Konfiguration für Kanal 1 und obere Bänder, Auswertung für Rufzeichen-1	decode-ft8-1N.sh: verwendet Konfiguration für Kanal 1 und untere Bänder, Auswertung für Rufzeichen-1
	write-c2-files-1D.sh: Diese Konfigurationsdateien enthalten die via GPS empfangene Temperatur-Frequenz-Korrektur. Sie legen außerdem für ein bis acht Bänder sowohl die Frequenz als auch den Antenneneingang #1 fest.	write-c2-files-1N.sh: Diese Konfigurationsdateien enthalten die via GPS empfangene Temperatur-Frequenz-Korrektur. Sie legen außerdem für ein bis acht Bänder sowohl die Frequenz als auch den Antenneneingang #1 fest.
	upload-ft8-1.cfg: In dieser Upload-Konfiguration finden Sie die Parameter CALL (Rufzeichen) und GRID (Locator) sowie optional ANTENNA (Antenne)	upload-ft8-1.cfg: In dieser Upload-Konfiguration finden Sie die Parameter CALL (Rufzeichen) und GRID (Locator) sowie optional ANTENNA (Antenne)
Antenne 2	Tag	Nacht
	ft8.cron: Stundenfestlegung für Tag Minuten-Muster für Kanal 2	ft8.cron: Stundenfestlegung für Nacht Minuten-Muster für Kanal 2
	decode-ft8-2D.sh: verwendet Konfiguration für Kanal 2 und obere Bänder; Auswertung für Rufzeichen-2	decode-ft8-2N.sh: verwendet Konfiguration für Kanal 2 und untere Bänder; Auswertung für Rufzeichen-2
	write-c2-files-2D.sh: Analog wie in write-c2-files-1D.sh bzw. write-c2-files-1N, jedoch für Antenne #2	write-c2-files-2N.sh: Analog wie in write-c2-files-1D.sh bzw. write-c2-files-1N, jedoch für Antenne #2
	upload-ft8-2.cfg: Analog wie in upload-ft8-2.cfg, jedoch für Antenne #2	upload-ft8-2.cfg: Analog wie in upload-ft8-2.cfg, jedoch für Antenne #2

Software vanaf het begin

Voorbeelden op basis van de 4 deling uit het Duits te vinden op de website van Red Pitaya

Dezelfde indeling kan worden gebruikt om de opbouw van Pavel Demin te beschrijven

Voorbeeld Martin Ossman, zie ook Elektor

Red Pitaya DHZ - software

MATLAB, LABview, SCILAB of Python

Python direct vanuit de WEB browser met behulp van Jupyter

C/C++ programmering via C APIs

Red Pitaya DHZ - software

Hoe start je met het programmeren van Red Pitaya of het ontwikkelen van je eigen toepassingen?

Er zijn verschillende manieren om te programmeren / ontwikkelen met Red Pitaya.

Afstandsbediening / programmeren Afstandsbediening kan worden gedaan door gebruik te maken van de meest populaire ontwikkeltools MATLAB, LABview, SCILAB of Python op afstand. Er zijn verschillende voorbeelden beschikbaar.

Programmering Red Pitaya direct vanuit WEB browser / Python Red Pitaya kan worden geprogrammeerd in Python direct vanuit de WEB browser met behulp van Jupyter.

C/C++ programmering -> RedPitaya hardware functies kunnen gemakkelijk worden benaderd via C APIs. Voorbeelden zijn beschikbaar op deze link.

FPGA programmering Informatie over het compileren van Red Pitaya open source FPGA code is hier te vinden.

Maken van eigen WEB applicaties Instructies over hoe je eigen Red Pitaya WEB applicaties kunt maken vind je hier.

<https://github.com/RedPitaya/RedPitaya>

<https://github.com/RedPitaya/RedPitaya#how-to-start-programming-red-pitaya-or-developing-your-own-applications>

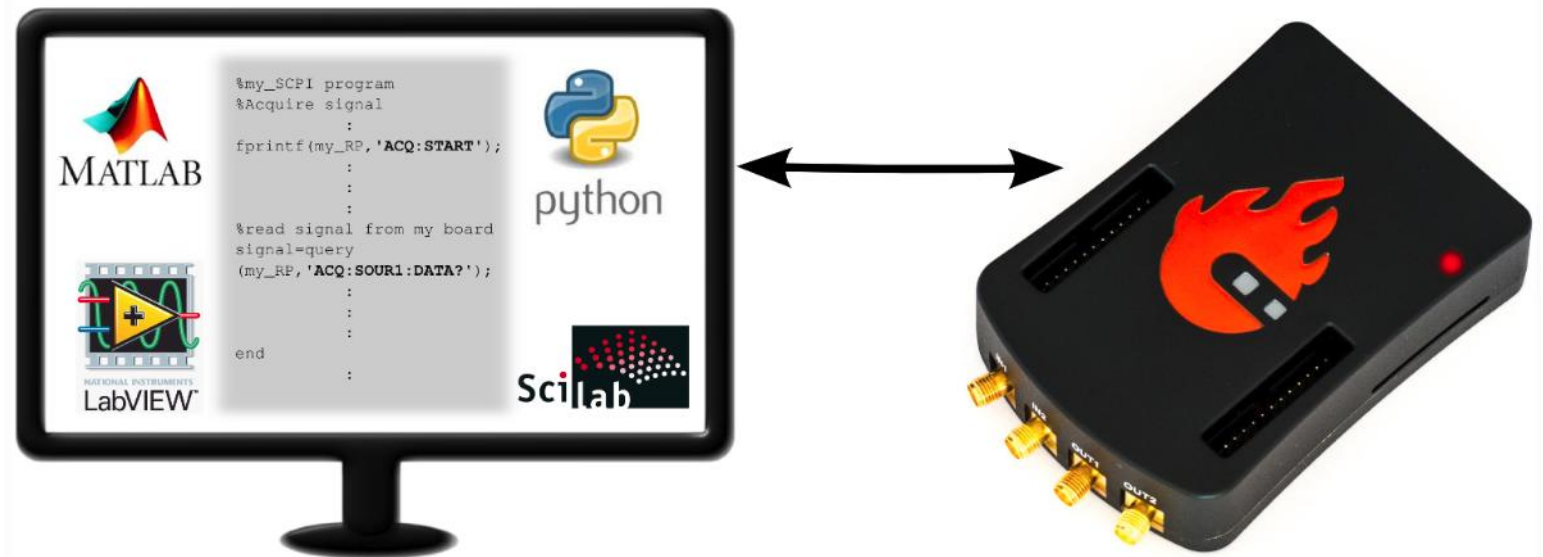
Afstandsbediening voor instrumenten, DSP, signaalverwerking

Matlab, LabVIEW, Scilab of Python en C

<https://redpitaya.readthedocs.io/en/latest/appsFeatures/remoteControl/remoteControl.html>

SCPI-server

NI LabVIEW Community Edition

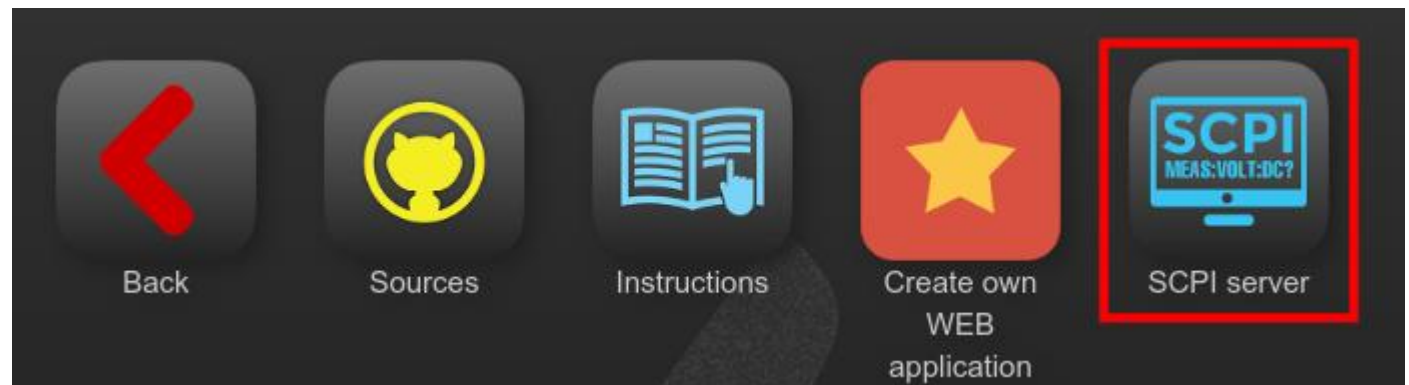


SCPI server toepassing

SCPI geeft toegang tot Red Pitaya hardware (GPIO, snelle en langzame DACS and ADCs, I2C, SPI, UART, LEDs)

Maar niet tot de FPGA, hoewel er wel hulpmiddelen zijn om de FPGA te programmeren in bijvoorbeeld de LabVIEW omgeving (gebaseerd op Vivado)

SCPI = Standard Commands for Programmable Instruments (skippy)



SCPI

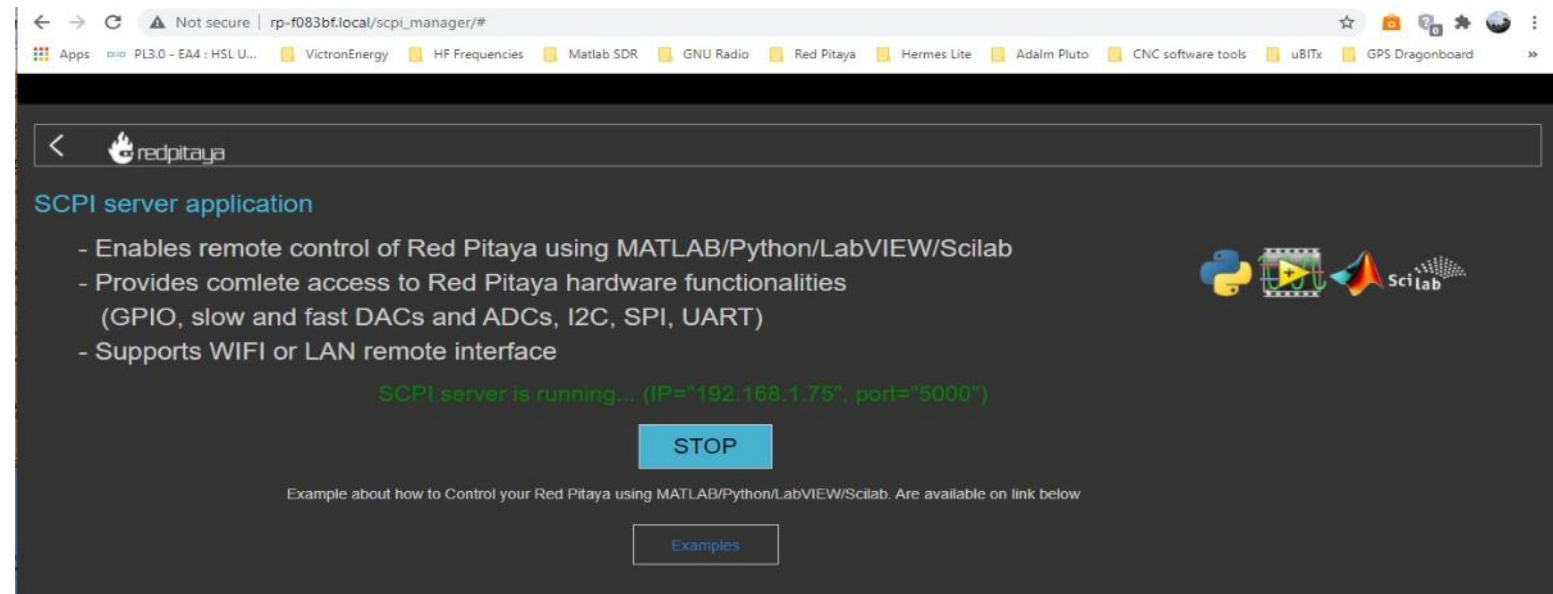
SCPI = Standard Commands for Programmable Instruments (skippy)

Laag bovenop IEEE 488.2

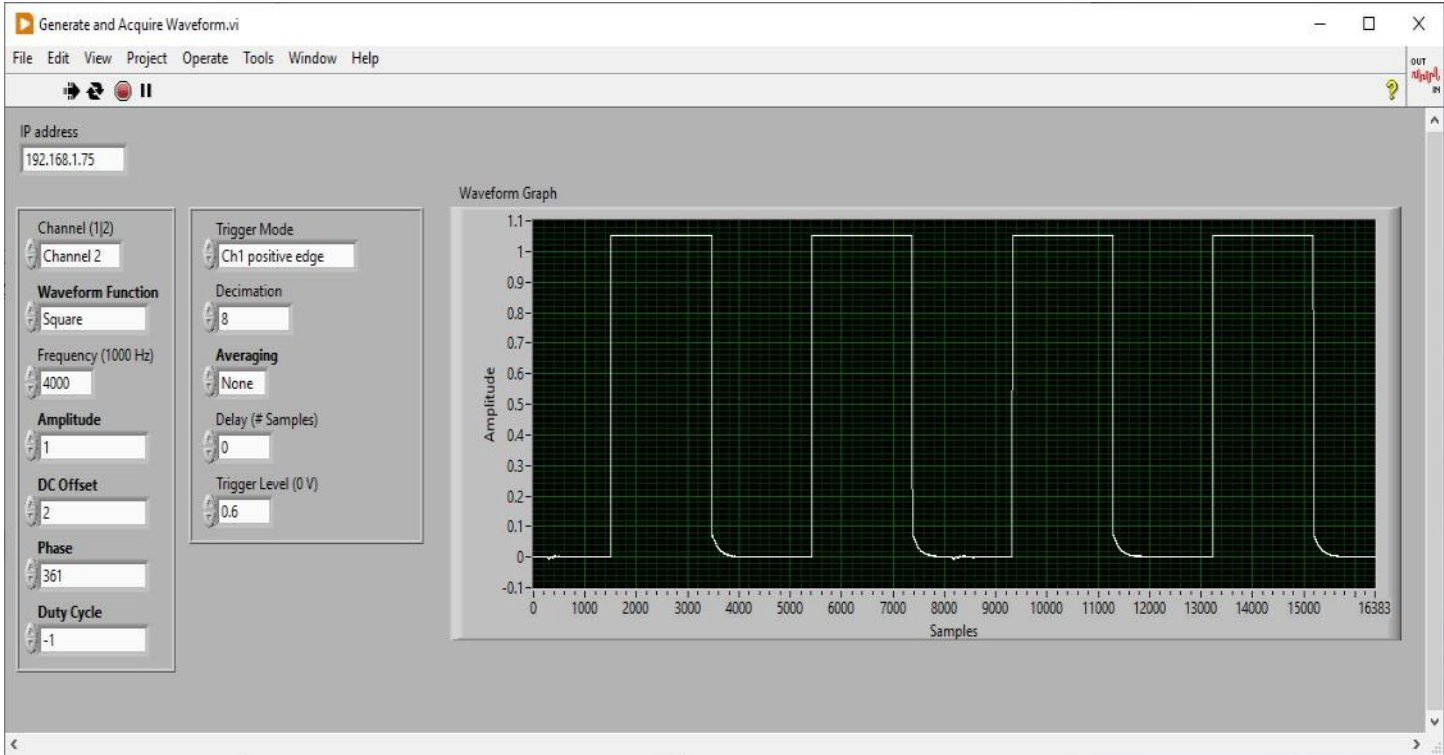
```
:MEASure
  :VOLTage
    :DC?
    :AC?
  :CURRent
    :DC?
    :AC?
  ...
```

VISA = Virtual instrument software architecture

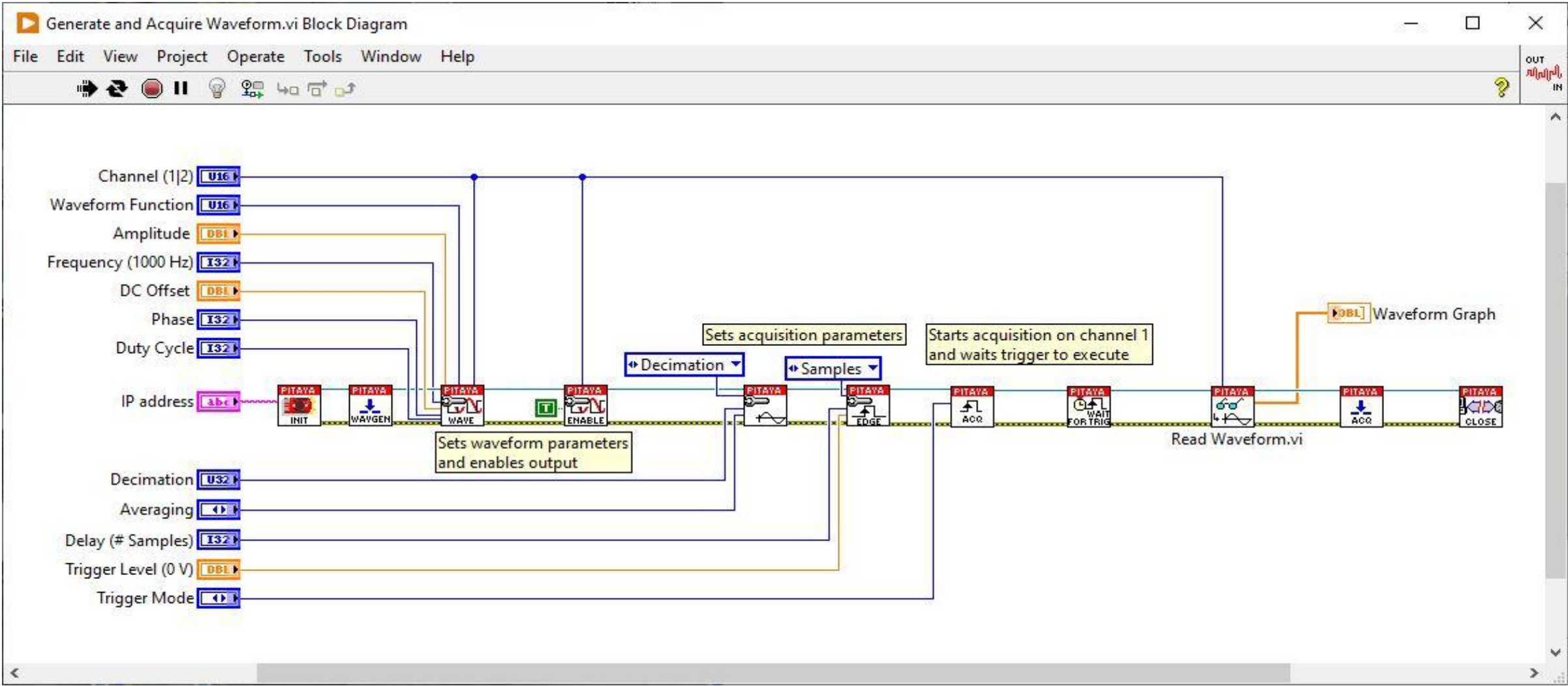
GPIO = General Purpose Interface Bus



NI LabView Community Edition



NI LabView



Mathworks - Matlab

```
>> %% Define Red Pitaya as TCP/IP object
IP= '192.168.1.75';      % Input IP of your Red Pitaya...
port = 5000;
tcpipObj=tcpip(IP, port);

%% Open connection met your Red Pitaya
fopen(tcpipObj);
tcpipObj.Terminator = 'CR/LF';

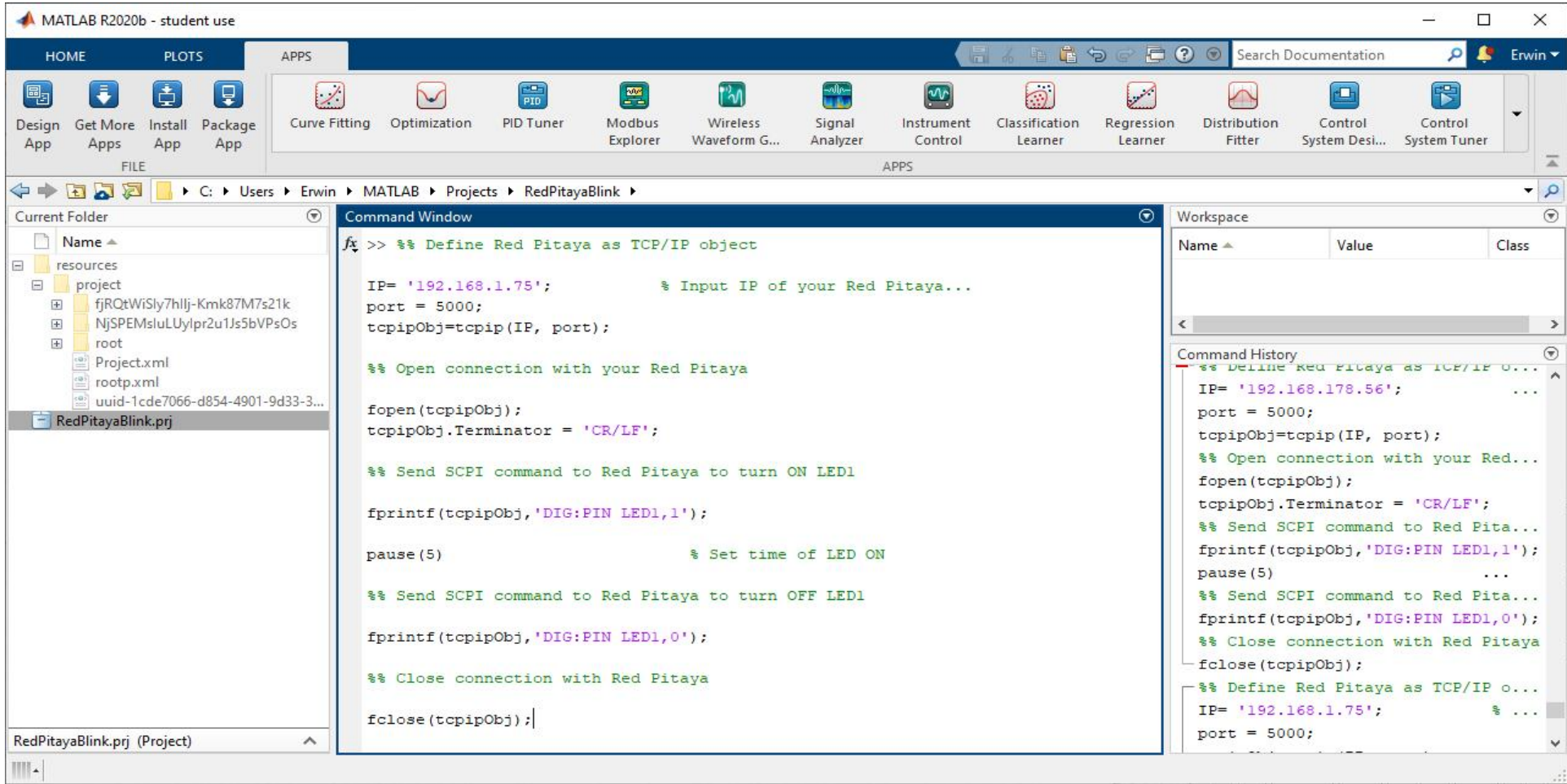
%% Send SCPI command to Red Pitaya to turn ON LED1
fprintf(tcpipObj,'DIG:PIN LED1,1');
pause(5)                % Set time of LED ON

%% Send SCPI command to Red Pitaya to turn OFF LED1
fprintf(tcpipObj,'DIG:PIN LED1,0');

%% Close connection met Red Pitaya
fclose(tcpipObj);
```



Matlab



Slow Morse



```
RedPitaya-blink-repeat.m  RedPitaya-blink-repeat.mlx  +

Define Red Pitaya as TCP/IP object
1  IP= '192.168.1.75';           % Input IP of your Red Pitaya...
2  port = 5000;
3  tcpipObj=tcpip(IP, port);

Open connection with your Red Pitaya
4  fopen(tcpipObj);
5  tcpipObj.Terminator = 'CR/LF';

Send SCPI command to Red Pitaya to turn ON LED1
6  for i= 1 : 5
7      fprintf(tcpipObj,'DIG:PIN LED1,1');
8
9      pause(5)                  % Set time of LED ON

Send SCPI command to Red Pitaya to turn OFF LED1
10     fprintf(tcpipObj,'DIG:PIN LED1,0');
11     pause(2)                  % Set time of LED OFF|
12     end

Close connection with Red Pitaya
13     fclose(tcpipObj);
```

FPGA Verilog HDL

Google: basic radio verilog hdl → en de studie van de Hardware Description Language kan beginnen

https://www.dropbox.com/sh/5fy49wae6xwxa8a/AACI--BhQvcNgjeQLRaiX9dha/ClubVivado2016_Pavel_Demin.pdf?dl=1

Complexiteit op veel niveaus

Afhankelijkheid van de software versie

Versie van Linux / Xilinx-Vivado ontwikkelomgeving versie / standaard locaties / Red Pitaya github materiaal

Alternatieven voor de Red Pitaya's

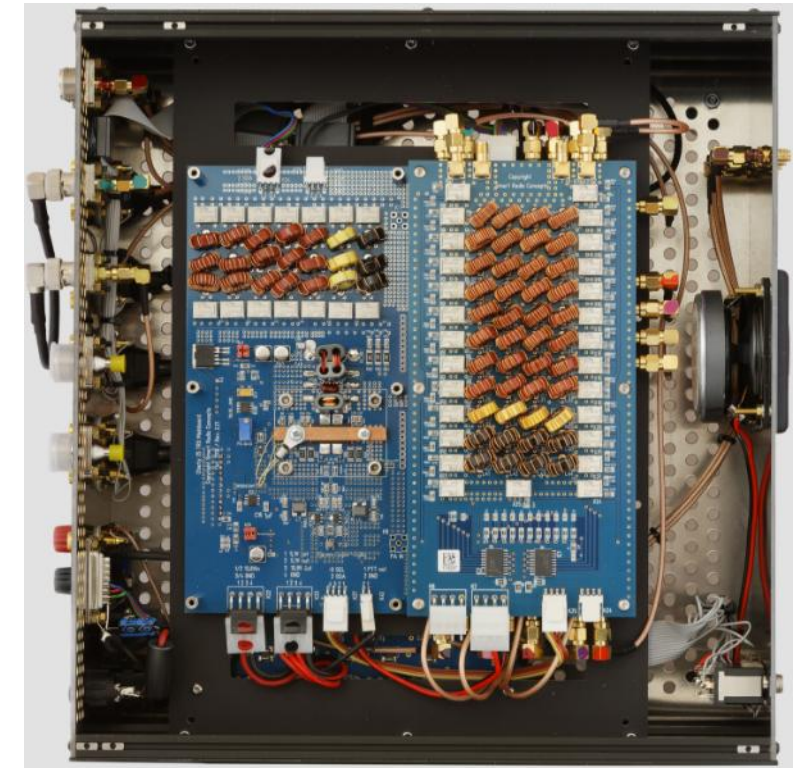
Alleen ontvangst :

Van het hele eenvoudige SDR shield voor de Arduino (Elektor) en de beroemde goedkope RTL-SDR Dongle
SDRPlay RSP1a
FiFi SDR
RX888 (RX666)

..of ook zenden:

Hermes Lite 2
Magnis SDR
Multus SDR
Lime SDR
OSA103 / ZEUS Radio (Russisch)
Tot de geweldenaar Charly25 die de Red Pitaya intern gebruikt

Maar ook andere FPGA ontwikkelkaarten zoals de Intel DE10-Lite en de Digilent Basys3



Complicaties

“...This version of the Red Pitaya SDR ontvanger emulates a Hermes module met eight ontvangers. It may be useful for projects that require eight ontvangers compatible met the programs that support the HPSSDR/Metis communication protocol....”

The modules are compatible met gnuradio and Hermes firmware version 1.8 through 3.2 (known as OpenHPSSDR protocol 1). It is not compatible met the new OpenHPSSDR protocol 2.

Source: <https://github.com/Tom-McDermott/gr-hpsdr>

Reverse Beacon network - Running 2 ontvangers on 1 SDRlab

Note: the SDR_ontvanger_HPSSDR start.sh script starts *two* (2) independent 8-slice SDRs, and the board will end up met two IP addresses and two MAC addresses on your LAN. So you can start two independent versions of Skimmer Server, each referencing a different MAC address by renaming the HermesIntf.dll as described. You can control which bands will get which antenna ports on the Red Pitaya - two are provided

<http://reversebeacon.blogspot.com/2020/06/using-red-pitaya-stemlab-12288-16-on.html>

<http://reversebeacon.blogspot.com/2020/09/using-two-ontvangers-in-your-red-pitaya.html>

To ExtIO or not to ExtIO

I am looking to use HDSDR met extio "ExtIO_RedPitaya.dll" or this one "ExtIO_RedPitaya_TRX.dll" at first. I found these files on the rbn-op forum, it seems to me met links that brought you to your website! like here

[http://pavel-demin.github.io/red-pitaya ... ansceiver/](http://pavel-demin.github.io/red-pitaya...ansceiver/)

These are for the original Red Pitaya 125-14.

Nothing in <http://pavel-demin.github.io/red-pitaya-notes/> works met STEMLab 122.88-16 SDR.

The correct notes for STEMLab 122.88-16 SDR are <https://pavel-demin.github.io/stemlab-sdr-notes/>

Try extio_hpsdr_cmake_mgw-v1.0.5.dll or extio_hpsdr_cmake_mgw-v1.0.4.dll from the following links:

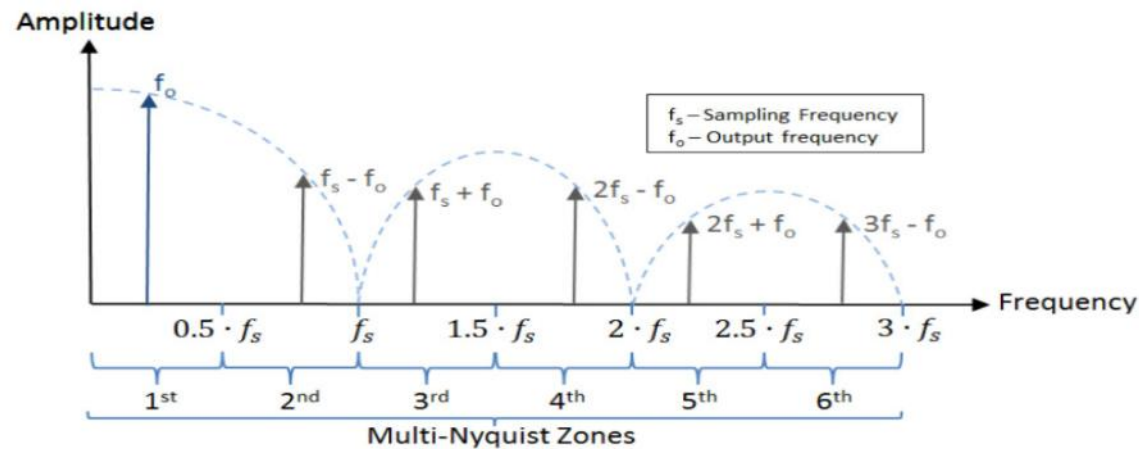
[https://github.com/IW0HDV/extio-iw0hdv/ ... tag/v1.0.5](https://github.com/IW0HDV/extio-iw0hdv/...tag/v1.0.5)

[https://github.com/IW0HDV/extio-iw0hdv/ ... tag/v1.0.4](https://github.com/IW0HDV/extio-iw0hdv/...tag/v1.0.4)

Back to the future

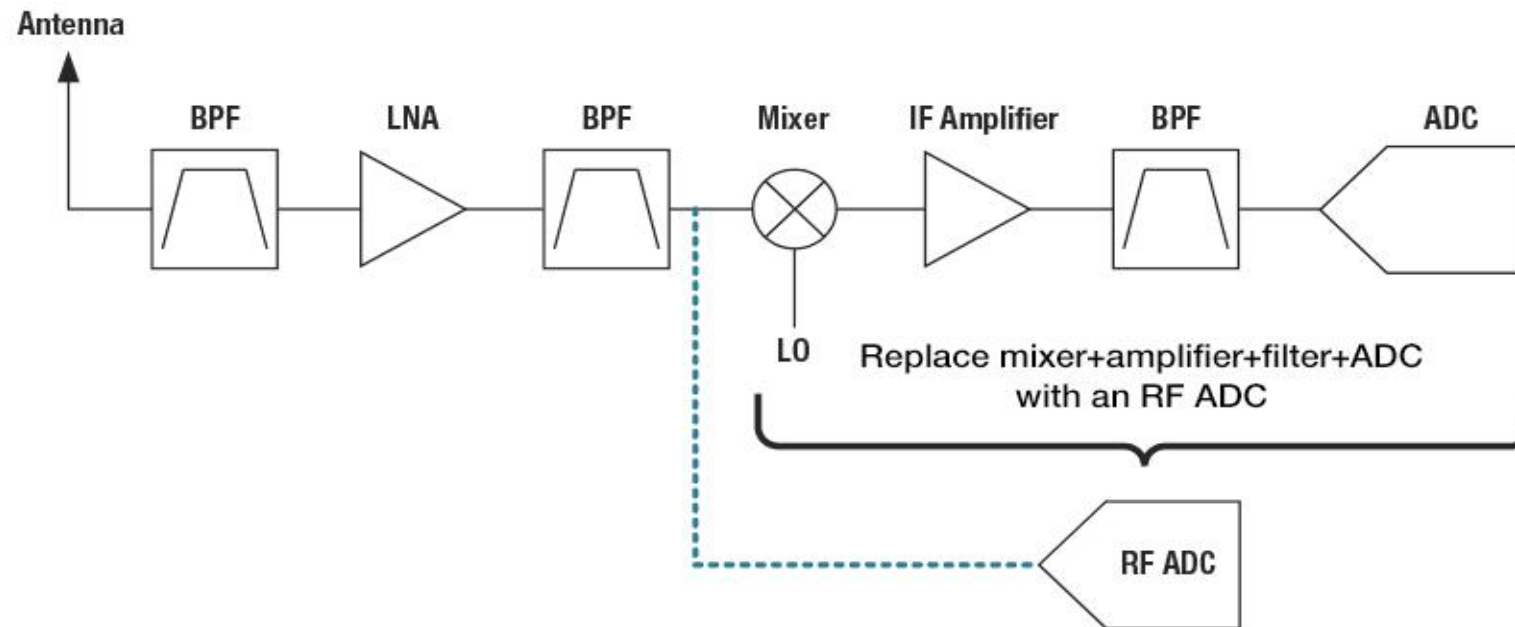
Stelling: als je snel genoeg kunt bemonsteren kun je tot steeds hogere frequenties werken zonder analoge frequentie mixer als front-end

Maar ook: als je de informatie theorie slimmer toepast zijn daarvoor ook andere mogelijkheden:



Of de hardware toekomst ...

<https://www.ti.com/data-converters/adc-circuit/high-speed/rf-sampling.html>



Conclusies

Beide Red Pitaya varianten bieden een stabiel platform voor SDR toepassingen tot zo'n 60 MHz

Voor toepassing als SDR ontvanger of multiband WSPR en FT8 ontvanger zijn ze zonder aanvullende software of hardware direct inzetbaar, met 2 ingangen en 2 uitgangen biedt dit veel flexibiliteit

De prestaties zijn uitzonderlijk goed : gevoelig, selectief, stabiel

Als eenvoudig meetinstrument vervangen ze een dedicated oscilloscoop, een signaalgenerator en een spectrum analyser

Conclusies

Daarnaast is het een zeer interessant platform om zelf met SDR en DSP software aan de gang te gaan

De combinatie van ARM Cortex processor met FPGA is krachtig genoeg voor vele toepassingen

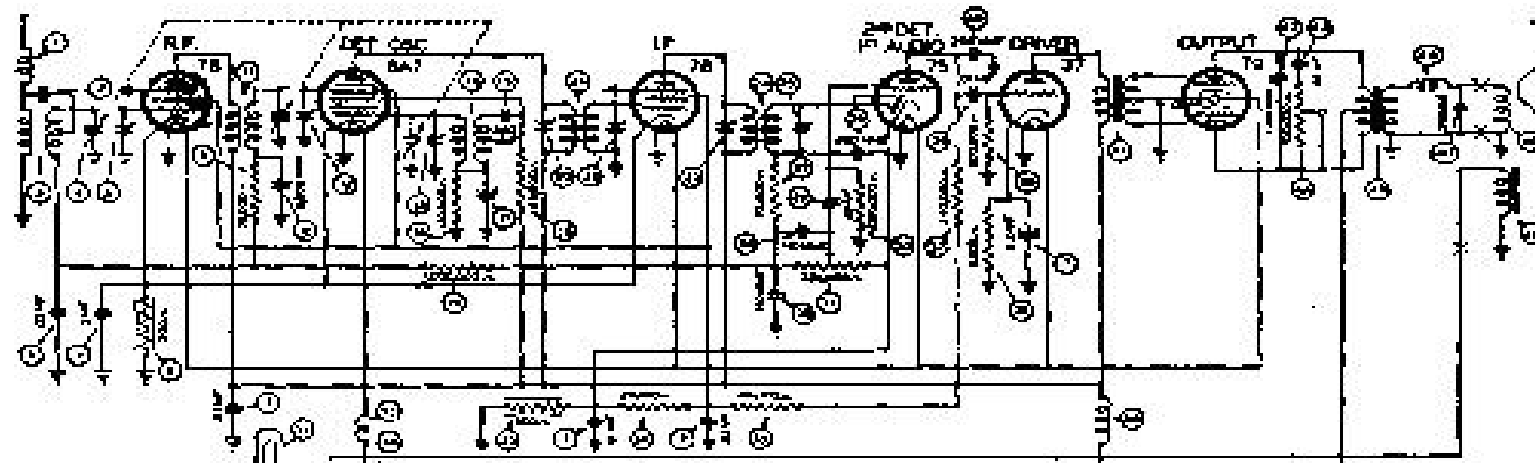
Dat vraagt wel inleren in een nieuwe omgeving, Linux met haar scripting en programmeertalen en Vivado als programmeeromgeving voor FPGA gateway (bitstream software)

Dan raak je meteen thuis in Verilog of System Verilog als Hardware Description Language, en eventueel in software voor in de browser of als app op een smartphone

Conclusies

Hoezo ingewikkeld?

Een buizen heterodyne ontvanger was ook niet je eerste project als radio amateur



Beheersing van complexiteit : ClubVivado2016_Pavel_Demin.pdf

Einde



RADIO AMATEURS LISTEN UP!

**THE RIGHT
SDR FOR
YOU**

Bronnen

<https://audiologieboek.nl/content/9-2-52-digitale-signaalbewerking-in-hoortoestellen-principes/>
<https://www.allaboutcircuits.com/technical-articles/dsp-basics-of-digital-down-conversion-digital-signal-processing/>
https://casper-toolflow.readthedocs.io/projects/tutorials/en/latest/tutorials/skarab/tut_intro.html
<http://www.dspguide.com/>
<https://www.taborelec.com/multi-nyquist-zones-operation-solution-note>
<https://www.facebook.com/redpitayaspark/>
<https://www.redpitaya.com/>
<https://redpitaya.readthedocs.io/en/latest/#>
<http://play.fallows.ca/wp/category/radio/software-defined-radio/>
 FUNKAMATEUR 1/54/2015, 2/158/2015, 6/604/2015, 5/442/2016, 12/1140/2016, 4/336/2018, 6/545/2018, 4/359/2019,
 7/634/2019
 Elektormagazine Nov/Dec 2015 – Martin Ossman

Bronnen FPGA e.a.

<http://pavel-demin.github.io/red-pitaya-notes/>
<https://www.xilinx.com/support/university/vivado/vivado-teaching-material/hdl-design.html>
<https://ln1985blog.wordpress.com/2016/02/07/red-pitaya-dac-performance/>
https://wiki.electrolab.fr/Projets:Lab:2017:Peripheriques_Angelia
<http://antonpotocnik.com/?p=487360>
<https://reversebeacon.blogspot.com/search?q=red+pitaya>
https://wiki.electrolab.fr/Projets:Lab:2018:Red_Pitaya
<http://play.fallows.ca/wp/radio/software-defined-radio/desktopsdr-captivates-matlab-rtl-sdr/#more-3991>
<https://www.ni.com/nl-nl/shop/labview/select-edition/labview-community-edition.html>
met
LabVIEW-FPGA/ta-p/3494460?profile.language=en
https://www.terraelectronica.ru/pdf/show?pdf_file=%252Fz%252FDatasheet%252FR%252Fredpitaya%2Bstemlab%2Bdocumentation.pdf
<https://semiwiki.com/fpga/290990-reverse-engineering-the-first-fpga-chip-xilinx-xc2064/>

SDR Knowledge Base

