

Entwicklung und Selbstbau von SDR-Transceivern

Henning Paul DC4HP

3. Mai 2024

Inhaltsverzeichnis

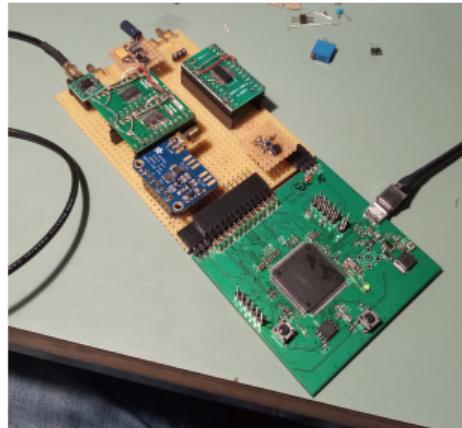
- 1 Einführung
- 2 FPGA-basierte SDRs
- 3 SDR-Transmitter
- 4 Software
- 5 Ein anderer Ansatz
- 6 Ausblick

Inhalt

- 1 Einführung
- 2 FPGA-basierte SDRs
- 3 SDR-Transmitter
- 4 Software
- 5 Ein anderer Ansatz
- 6 Ausblick

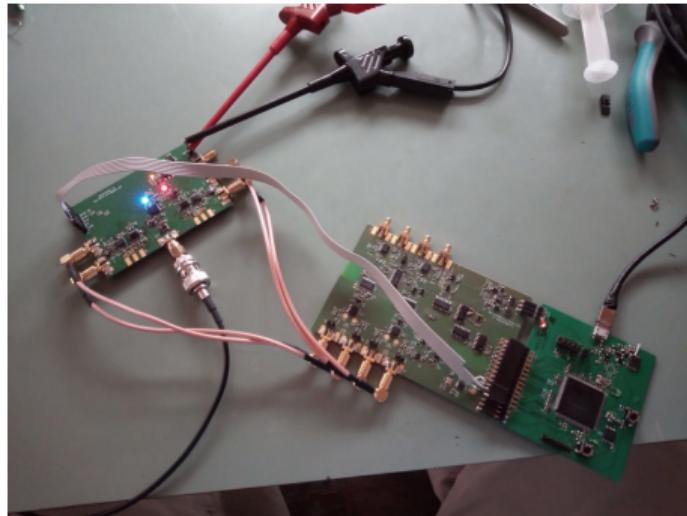
Erstes Echtzeitsystem: hnchBoard

- KiCad-Projekt des HackRF One ebenso wie Firmware- und Hostsoftware-Sourcen auf Github verfügbar
github.com/greatscottgadgets/hackrf/
- Ansatz: HackRF One als Grundlage, aber modularer Klon mit differentiellen I/Q-Ein- und Ausgängen um unterschiedliche Modulator- und Demodulatorboards anschließen zu können
- Hardware:
 - ▶ NXP LPC4330 Mikrocontroller mit modifizierter HackRF One-Firmware
 - ▶ MAX5189 2-Kanal-DAC, 2x ADC1175 (8 bit, 20MSps)
 - ▶ 74VHC GlueLogic (an Stelle des CPLDs des HackRF One)
 - ▶ ±5V Dual supply für den Analogteil (mit LT1945 Schaltregler + LDOs erzeugt)
 - ▶ Modifizierte libhackrf erlaubt Betrieb mit jeder HackRF-kompatiblen Software (bspw. GQRX)



Erstes Echtzeitsystem (2)

- VHF-Demodulatorboard mit ADRF6806
- Konfiguration des Demodulatorbausteins über SPI vom LPC4330 aus
- Integration in Firmware zur transparenten Ansteuerung

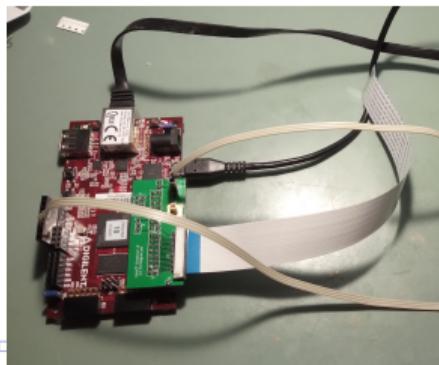


Inhalt

- 1 Einführung
- 2 FPGA-basierte SDRs
- 3 SDR-Transmitter
- 4 Software
- 5 Ein anderer Ansatz
- 6 Ausblick

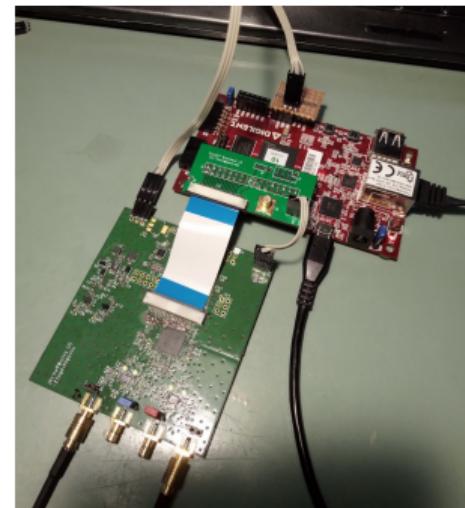
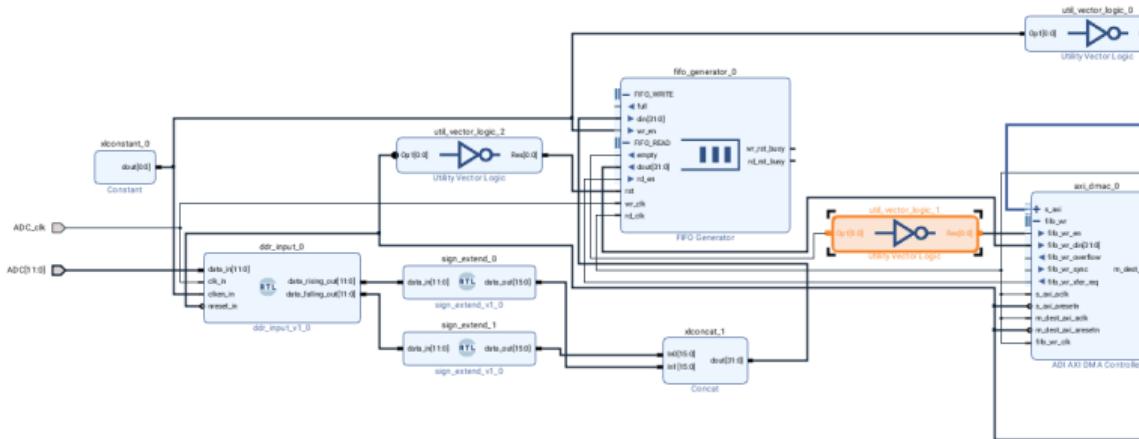
FPGA-Evaluationboard

- USB-Schnittstelle ist Flaschenhals, Umstieg auf eine schnellere Schnittstelle sinnvoll
- Quellen des ADALM-PLUTO von Analog Devices auf Github veröffentlicht
- Xilinx Zynq-7000 SoC offensichtliche Wahl, einziges erschwingliches Evalboard: Digilent Cora Z7
- Zentrale Funktionsblöcke aus der Library: DMA-Core & Dezimatorblock
- Erste Versuche mit dem AD-Teil des hncBoard mittels Flexadapter



FPGA-Evaluationboard (2)

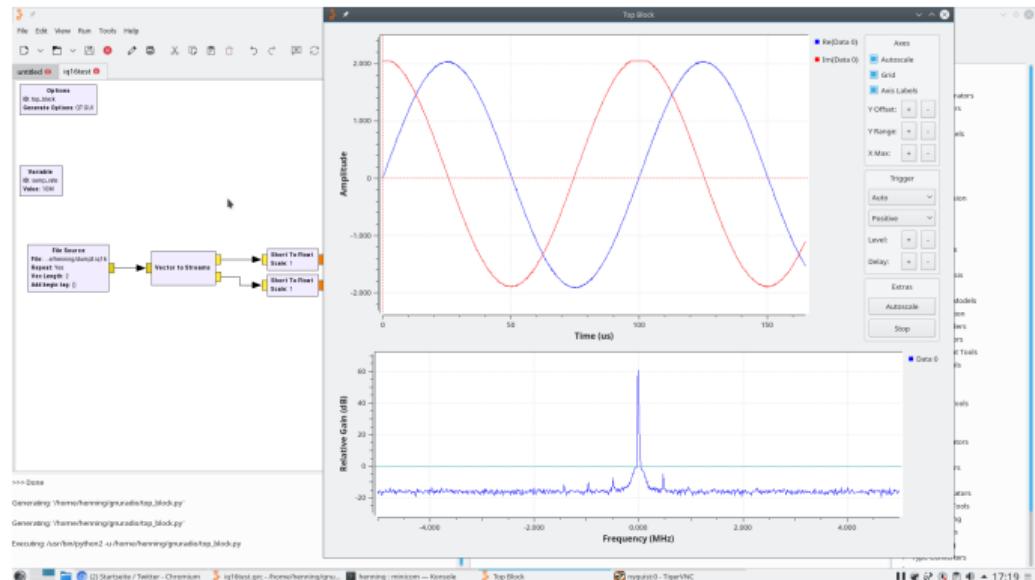
- Selbst layouttes Testboard zur Erhöhung der ADC-Auflösung auf 12 bit mit einem LTC2291 (25MSps)
 - Kernelmodul für Petalinux zur Kommunikation mit dem ADC ist Abwandlung des Moduls für den AD9361 des Pluto



```
root@rpkguesttest:~# lio_redaeadue -s 1921 buch_dumyadec > dump.lq16
root@rpkguesttest:~# hwdump dmpg_1q16 1 head -n 20
00000000 0fff 0000 0fff 0000 0004 0000 0002 0000
00000010 0002 0000 0001 0002 0002 0001 0fff
00000020 ffff 0fff 0000 0fff 0000 ffff ffff 0fff
00000030 ffff 0fff 0fff 0fff 0001 0001 0001
00000040 0000 0001 0fff 0002 0fff 0002 0001 0002
00000050 0001 0001 0004 0000 0005 0005 0003 0000
00000060 0005 0001 0006 0000 0005 0001 0004 0fff
00000070 0003 0000 0004 0000 0003 0003 0003 0001
00000080 0001 0001 0001 0002 0000 0002 0000 0001
00000090 0000 0001 0000 0001 0000 0000 0000 0000
000000a0 0000 0000 0000 0000 0000 0fff 0001 0fff 0001
000000b0 0000 0000 0000 0fff 0001 0001 0000 0fff
000000c0 ffff 0fff 0fff 0fff 0000 0fff 0fff 0fff
000000d0 ffff 0fff 0fff 0fff 0fff 0fff 0fff 0fff
000000e0 0001 0fff 0000 0fff 0000 0fff 0000 0fff
000000f0 0001 0fff 0001 0fff 0004 0fff 0002 0fff
00000100 0001 0fff 0004 0fff 0001 0fff 0000 0fff
00000110 0001 0fff 0fff 0fff 0fff 0fff 0fff 0fff
00000120 ffff 0fff 0000 0fff 0001 0000 0fff 0001
00000130 0002 0000 0002 0000 0005 0fff 0006 0fff
```

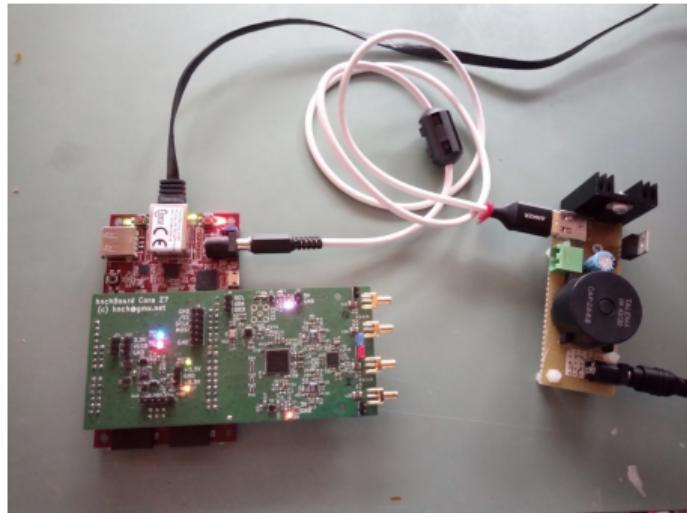
FPGA-Evaluationboard (3)

- IIO-Interface ermöglicht einfache Anbindung an GNU Radio



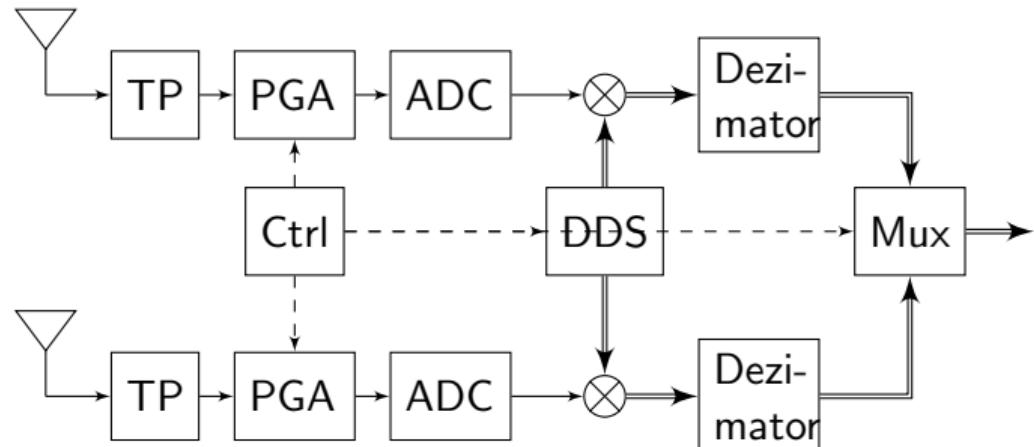
Aktueller SDR-Empfänger

- Aufsteckboard für das Cora Z7 mit 40MSps/12bit LTC2292 ADC (Upgrade mit pin-kompatiblen ADCs aus der Familie bis zu 105MSps/14bit möglich)
- Mit I/Q-Demodulatorboards für VHF oder UHF nutzbar
- Kurzwelle ohne Extraboard nutzbar - dann 2 unabhängige ADC-Kanäle
- github.com/hennichodernich/gr-iio und github.com/hennichodernich/gr-osmosdr ermöglichen Ansteuerung der Demodulatorboards aus GNU Radio oder GQRX heraus
- Kernelmodul:
github.com/hennichodernich/hnch-iio



Design des SDR-Empfängers

- $40 \text{ MSps} \times 2 \text{ Kanäle} \times 16\text{bit} \rightarrow 160 \text{ MByte/s}$, zu viel für Gigabit Ethernet
- Realistisch: 60MByte/s
- Digitaler Mischer und Dezimator nötig
- Pavel Demin's SDR-Firmware für den Red Pitaya:
CIC-Dezimator, beliebige Mischfrequenzen
- Hier: FIR-Dezimation mit Faktor 8, Mischfrequenzen
 $n \cdot f_s / 8 = n \cdot 5 \text{ MHz}$



Inhalt

- 1 Einführung
- 2 FPGA-basierte SDRs
- 3 SDR-Transmitter
- 4 Software
- 5 Ein anderer Ansatz
- 6 Ausblick

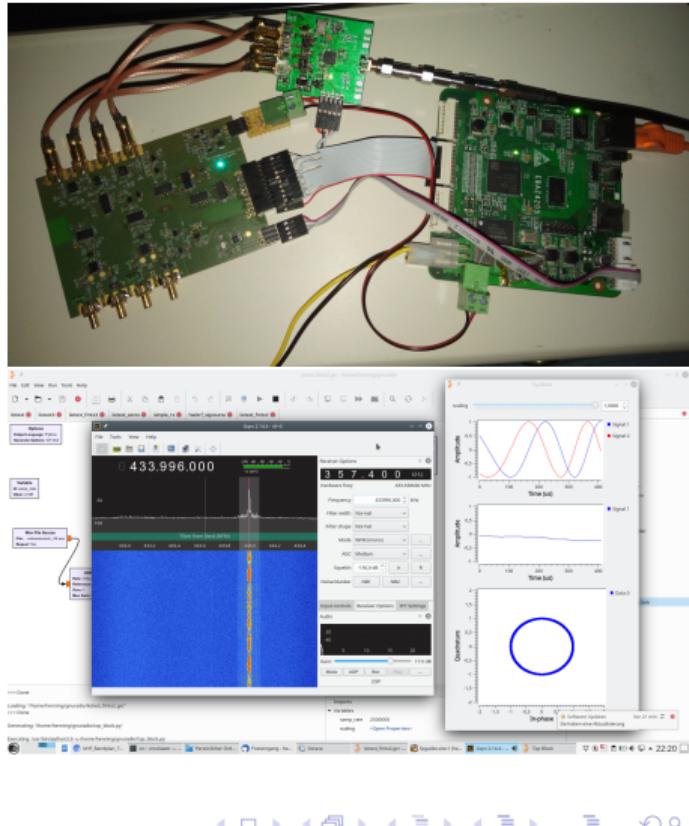
Motivation

- Cora Z7 hat nicht genug IOs um ADC und DAC für Vollduplex gleichzeitig bedienen zu können
- Zynq-7000-Boards mit ausreichend IOs sind vergleichsweise teuer (bspw. Zedboard für >600€)
- Idee: Gebrauchte EBAZ4205 Bitcoin-Mining-Controller von AliExpress (damals 15€, heute ca 35€) auf Eignung überprüfen

The screenshot shows the Hackaday homepage with a prominent feature article titled "HACKING THE FPGA CONTROL BOARD FROM A BITCOIN MINER" by Tom Nardi from November 18, 2020. The article discusses the use of an old EBAZ4205 board for mining cryptocurrency. The page includes a search bar for "ebaz4205" and a "SEARCH" button, along with a "NEVER MISS A HACK" section and a "SUBSCRIBE" button.

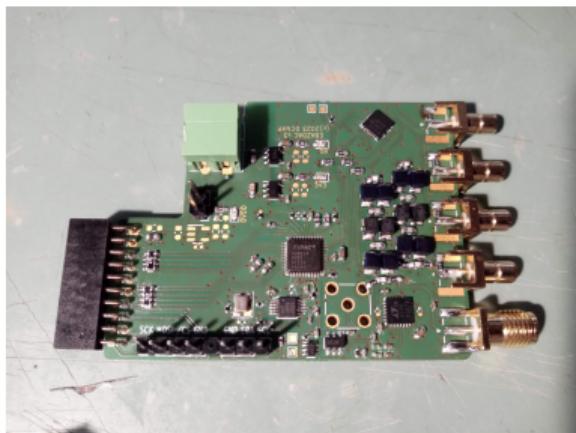
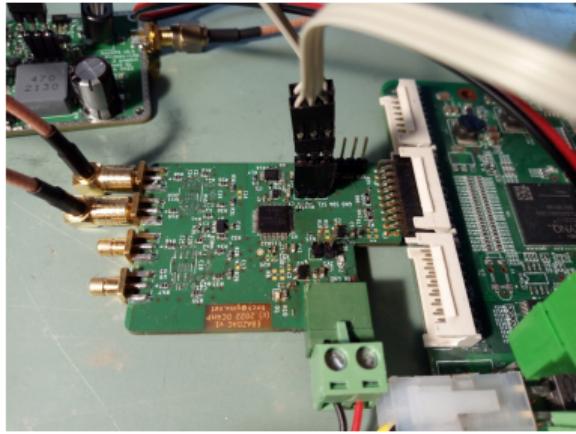
Erste Schritte mit dem EBAZ4205

- Ziel: Petalinux auf dem EBAZ4205 zum Laufen bekommen
- Schwieriger als erwartet
 - ▶ Studium des reverse engineereden Schaltplans github.com/xjtuecho/EBAZ4205 und
 - ▶ intensive Beschäftigung mit dem Linux-Devcetreesystems waren notwendig um Ethernet zum laufen zu bekommen
- Sendevereuche wieder mit dem hnchBoard und dem ADRF6720 I/Q-Modulator von 2016
- Test: FM auf 70cm mit GNU Radio direkt in ein RTL-SDR



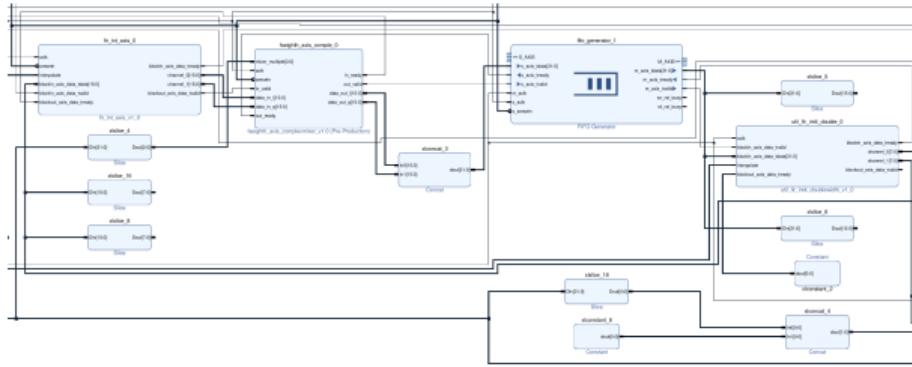
Kurzwelle mit dem Tx-SDR

- DAC-Board designt (oben: 1. Version, unten aktuelle Version)
- DAC-Baustein unterstützt bis zu 125MSamp/s, 96MSamp/s erfolgreich nachgewiesen
- EBAZ4205 hat nur 100Mbit Ethernet → Beschränkung auf 8 bit Wortbreite und Interpolation notwendig
- Interpolation um den Faktor 32:
 - ▶ 3 MSamp/s Schnittstellenrate
 - ▶ 6 MByte/s Datenrate
 - ▶ 96 MSamp/s DAC-Rate
- Variation der Schnittstellenrate z.T. sinnvoll, damit gewünschtes Kurzwellenband “mittig” im darstellbaren Frequenzband zu liegen kommt (bspw. $n = 9$:
 $f = 25,5..28,5\text{MHz}$, für 10m ungünstig)



Digitaler Mischer

- 2 interpolierende Mischstufen ($\times 8$ und $\times 4$)
- Mischung mit $n \cdot f_s/4$ kommt ohne Multiplikationen aus (1, 0, -1, 0)
- Mischung mit $n \cdot f_s/8$ sehr gut approximierbar ($256/\sqrt{2} = 181.02$)
- Insgesamt Mischerfrequenzen 0..31· Schnittstellenrate möglich

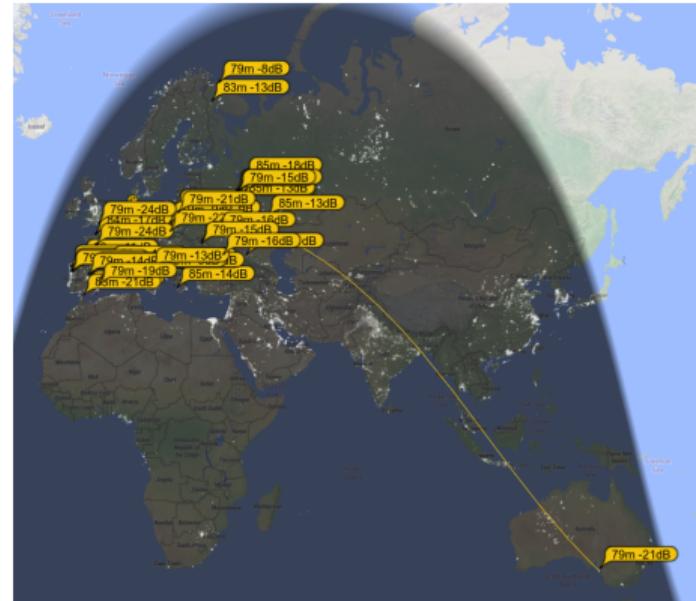


```
architecture arch_lmp of fsleigh_axis_complexmixer_v1_0 is
begin
    -- Add user logic here
    bits_i <= std_logic_vector(sum_i);
    bits_q <= std_logic_vector(sum_q);
    data_out_i <= bits_i(internal_width-1) & bits_i(internal_width-2-bit_skip downto internal_width-out_width-bit_skip);
    data_out_q <= bits_q(internal_width-1) & bits_q(internal_width-2-bit_skip downto internal_width-out_width-bit_skip);

    do_fetch <= '1' when (
        ((state=s_idle) and (in_valid='1'))
        or ((state=s_reload) and (in_valid='1') and (out_ready='1'))
        or ((state=s_running) and (in_valid='1') and (out_ready='1'))
        or ((state=s_stalled) and (in_valid='1'))
        or ((state=s_restarting) and (in_valid='1') and (out_ready='1'))
    ) else '0';
    do_process <= '1' when (
        --((state=s_reload) and (in_valid='1') and (out_ready='1'))
        --((state=s_running) and (in_valid='1') and (out_ready='1'))
        --((state=s_stalled) and (out_ready='1'))
        --((state=s_restarting) and (out_ready='1'))
    ) else '0';
    in_ready <= '1' when (
        (state=s_idle)
        or (state=s_stalled)
        or ((state=s_restarting) and (out_ready='1'))
        or ((state=s_reload) and (out_ready='1'))
        or ((state=s_running) and (out_ready='1'))
    ) else '0';
    out_valid <= '1' when (
        (state=s_running)
        or (state=s_restarting)
    ) else '0';
end architecture;
```

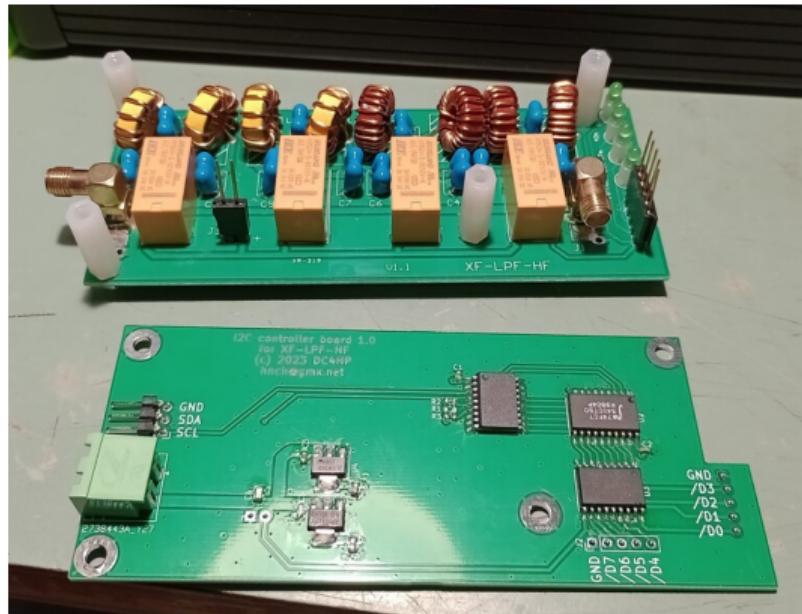
Endstufe

- Leistungsverstärker vom Hermes-Lite 2 inspiriert
(KiCad-Projekt unter
github.com/softerhardware/Hermes-Lite2 verfügbar)
- Ausgangsleistung 3-5W je nach Band
- Arbeitspunkt per I2C-Digitalpoti einstellbar
- Enable-Pin per Open Collector-Ausgang schaltbar



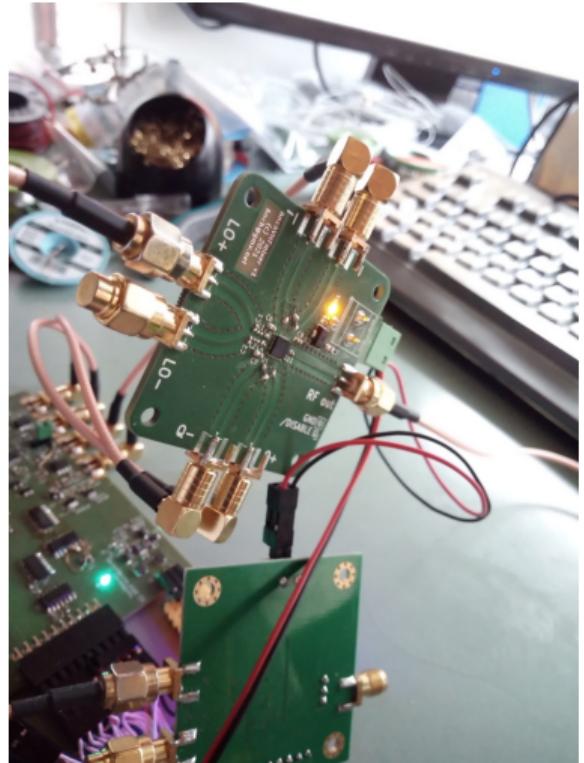
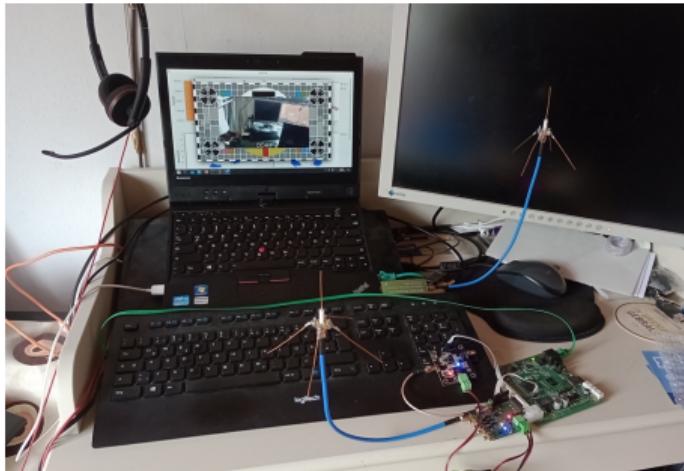
Tiefpassfilter

- Schaltbares Tiefpassfilter von AliExpress
- Aufsteckboard zur Ansteuerung per I2C
- IIO-Kernelmodul verfügbar auf
github.com/hennichodernich/mcp23009



VHF und UHF

- Problem: ADRF6720-Board nur für UHF geeignet
- Abhilfe: Universeller I/Q-Mischer LTC5598, rechts als separates Board, unten auf DAC-Board integriert
- LO über ADF4351 erzeugt
- Anwendungsbeispiel: DATV auf 23cm

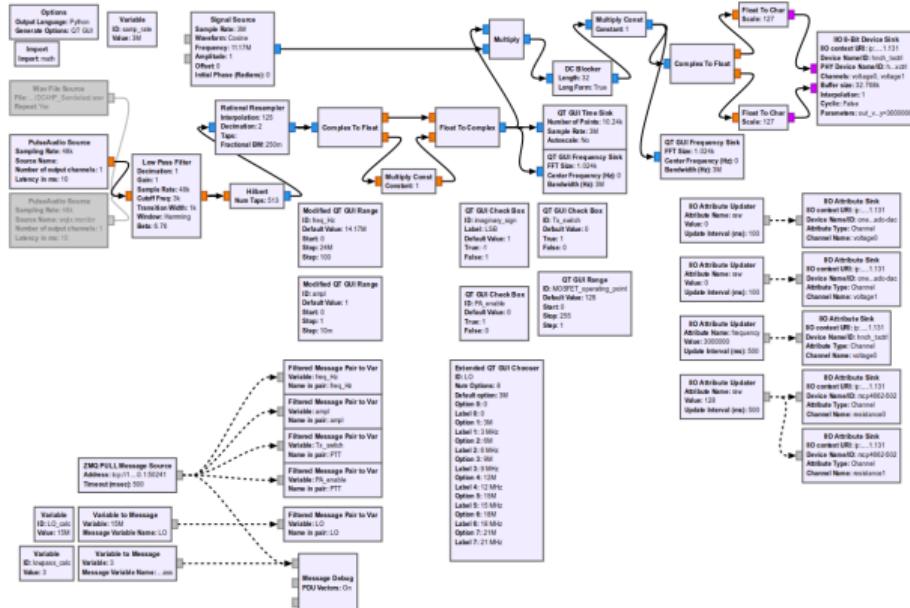


Inhalt

- 1 Einführung
- 2 FPGA-basierte SDRs
- 3 SDR-Transmitter
- 4 Software
- 5 Ein anderer Ansatz
- 6 Ausblick

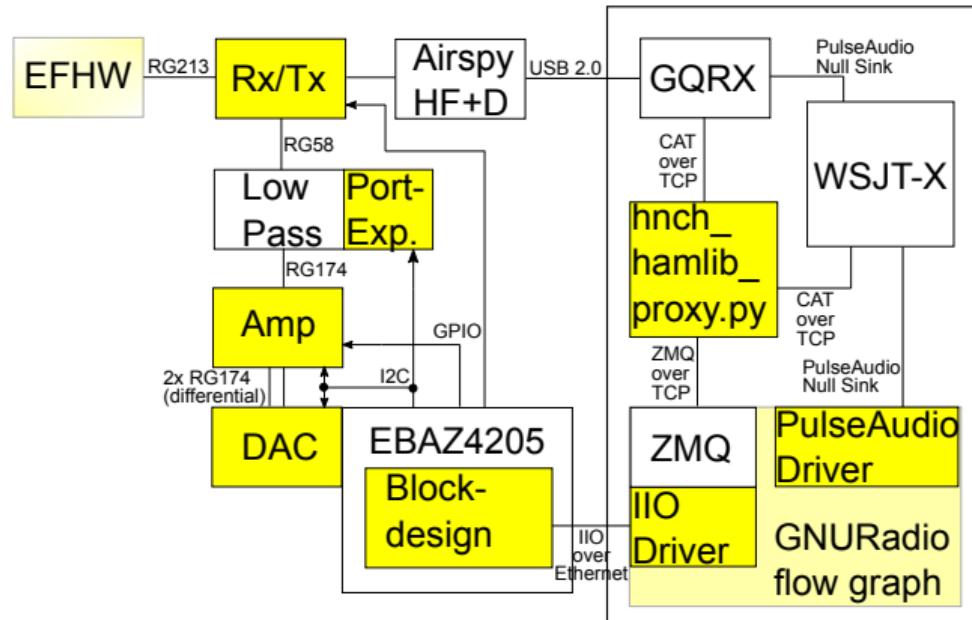
GNU Radio-basierter SSB-Tx

- GNU Radio-Flowgraph auf dem PC implementiert die eigentliche Signalverarbeitung
- Kommunikation mit Flowgraph über PulseAudio (NF) bzw. über ZMQ (PTT und Frequenz)
- Kommunikation des Flowgraphs mit dem FPGA-Boards über IIO
⇒ verschiedene IIO-Treiber mussten geschrieben oder angepasst werden



Zusammenspiel der Komponenten

- Pythonskript basierend auf <http://james.ahlstrom.name/hamlibserver.py> implementiert ein rigctld-kompatibles CAT interface für Programme wie WSJT-X und fldigi
- Rechtssstehend eigene Anteile in gelb, unveränderte in weiß
- Airspy HF+ Discovery als Empfänger

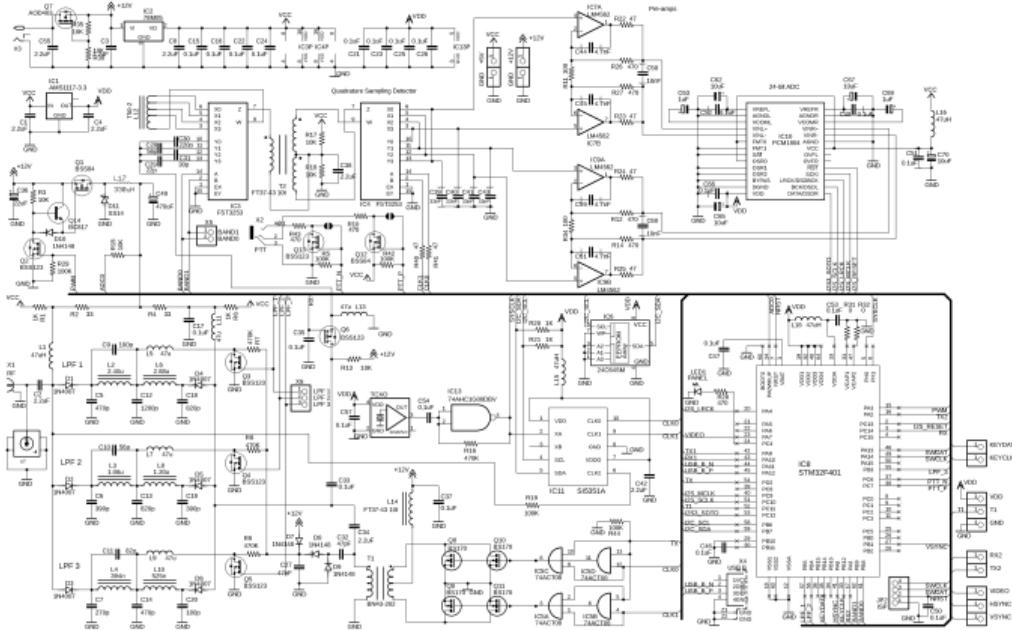


Inhalt

- 1 Einführung
- 2 FPGA-basierte SDRs
- 3 SDR-Transmitter
- 4 Software
- 5 Ein anderer Ansatz
- 6 Ausblick

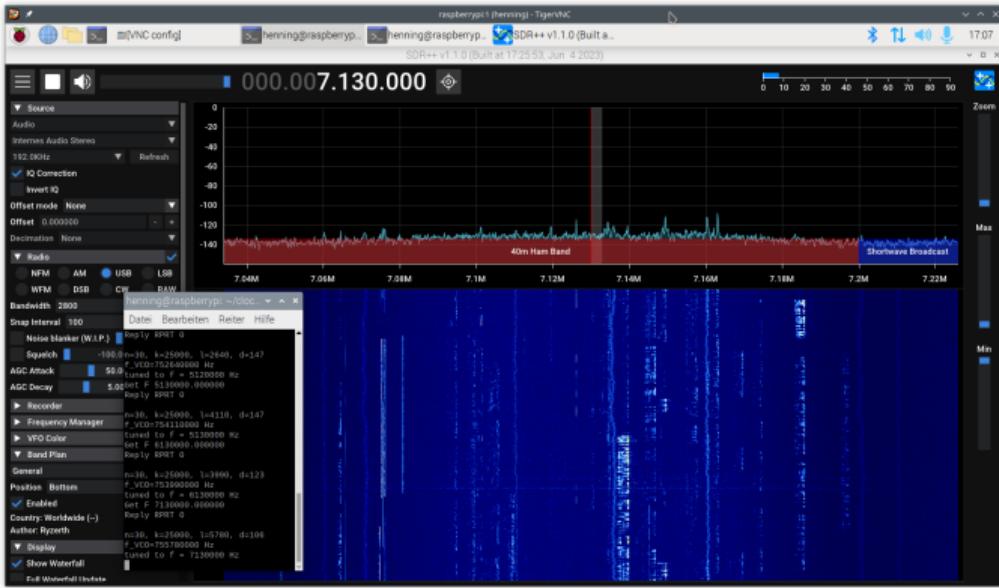
QDX als Vorbild

- Schaltplan des QDX von den QRP Labs online verfügbar:
- “Digitale” (schaltendene) Endstufe aus einem einzelnen Digitalsignal getrieben
- Schaltmischer (“H-mode mixer”) mit Hifi-ADC dahinter
- Idee:
 - ▶ Empfänger als Raspberry Pi-Hat nachbauen
 - ▶ Mit Endstufe experimentieren



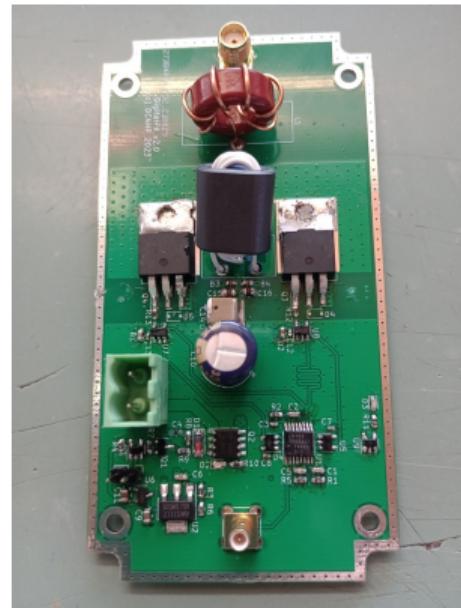
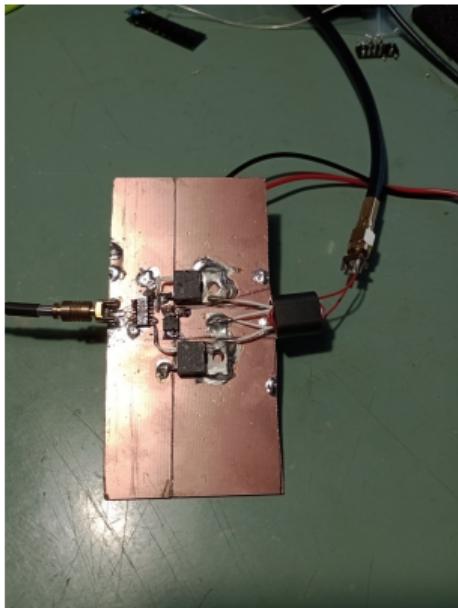
Empfängerboard

- 24bit/192kHz-I2S-ADC (Kernelmodulsourcen unter github.com/hennichodernich/WM8786_RasPi)
- LO (wie beim QDX) mit Si5351A erzeugt, rigctld-kompatibles Python Skript setzt Register



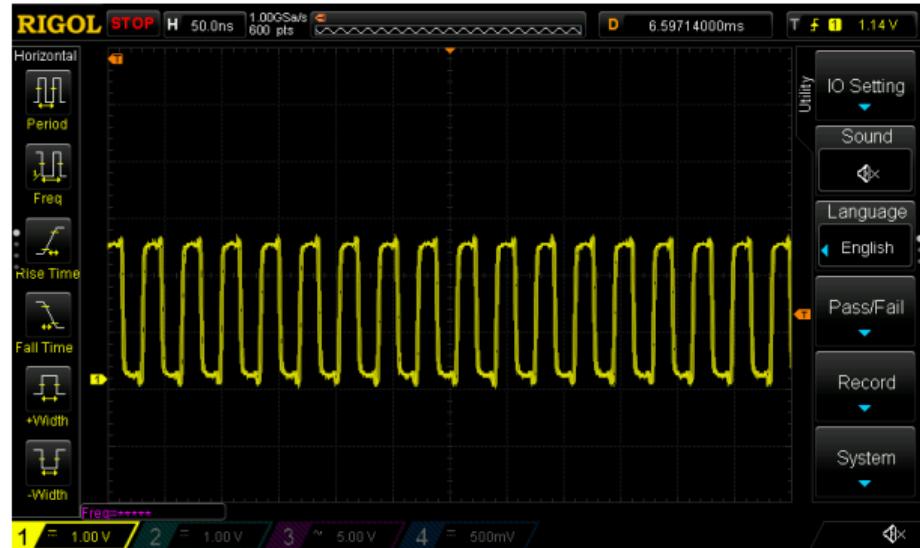
Schaltende Endstufe

- 2 Allzweck-MOSFETs (IRF510) im Gegentakt-Schaltbetrieb
- 4 A-FET-Treiber
- Monoflop stellt Höchst-Einschaltdauer sicher (wg. thermischen Effekten)
- 22 W in der Grundwelle
- github.com/hennichodernich/digital-pa



Sendesignalerzeugung

- Nicht mit dem Si5351 wie im QDX, sondern mit rpitx von Evariste F5OEO
github.com/F5OEO/rpitx
- Clocksynthesizer des Raspberry Pi wird missbraucht um beliebige Taktsignale an Pin 7 zu erzeugen
- Ursprünglich nur FM/FSK möglich - seit dem RPi 2 auch AM (Kombi FM+AM ergibt SSB bzw. allgemein I/Q)
- Anbindung an GNU Radio übernimmt
github.com/jmfriedt/gr-rpitx



Gesamtgerät

- Weitere integrierte Baugruppen:
 - ▶ DC/DC-Wandler 12-20V → 5V 4A
 - ▶ Eingangsstrom- und -spannungsmessung
 - ▶ RTC-Modul
 - ▶ Temperaturfühler auf dem Endstufentransistor
 - ▶ Externes GPS-Modul für exakte Uhrzeit

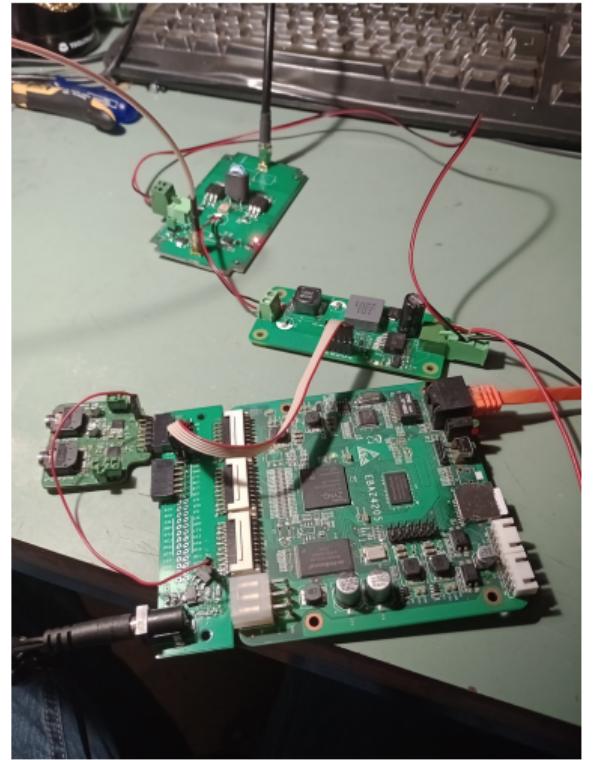


Inhalt

- 1 Einführung
- 2 FPGA-basierte SDRs
- 3 SDR-Transmitter
- 4 Software
- 5 Ein anderer Ansatz
- 6 Ausblick

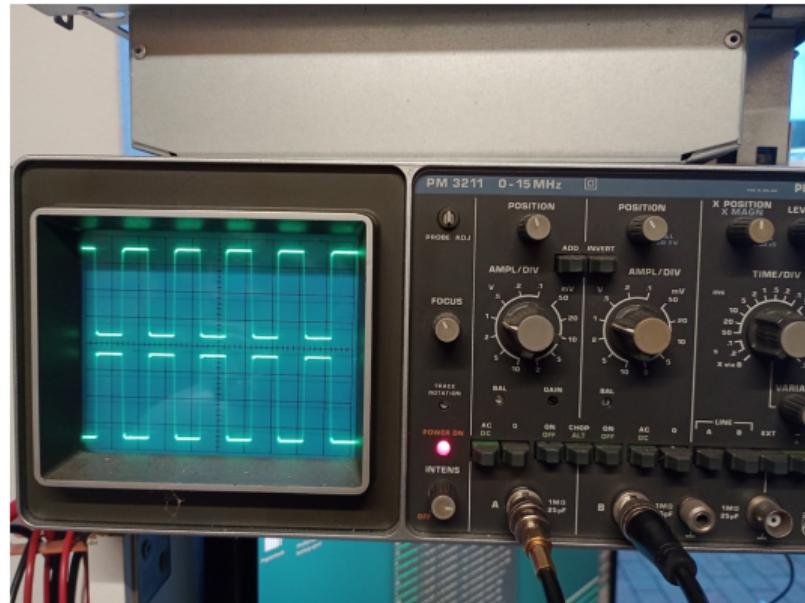
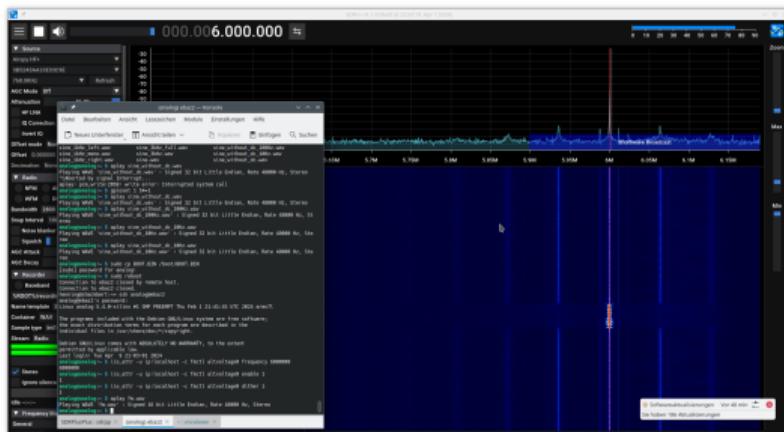
Work in progress

- Aktuelle Digitalendstufe erlaubt nur FM/FSK
- für SSB zusätzliche AM notwendig
 - ▶ Theorie: L.R. Kahn 1952, "Single Sideband Transmission by Envelope Elimination and Restoration"
 - ▶ bspw. angewendet bei Satellitentranspondern in Form von HELAPS (K. Meinzer DJ4ZC 1973)
- Durch Verarbeitung im FPGA ist Trennung von Amplitude und Phase einfach möglich
- Experiment: Brückenspannung der Digitalendstufe per DC/DC-Wandler dynamisch verstellen



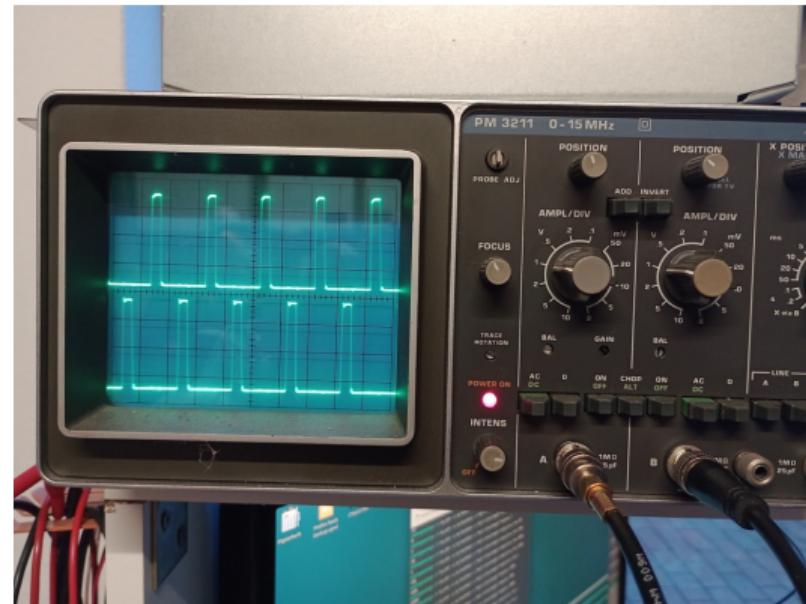
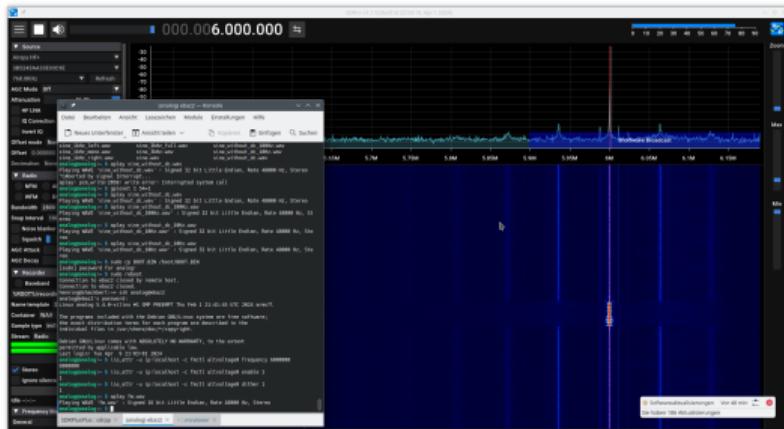
Work in progress (2)

- Frequenzmodulator im FPGA, MOSFETs der schaltenden Endstufe werden einzeln gesteuert



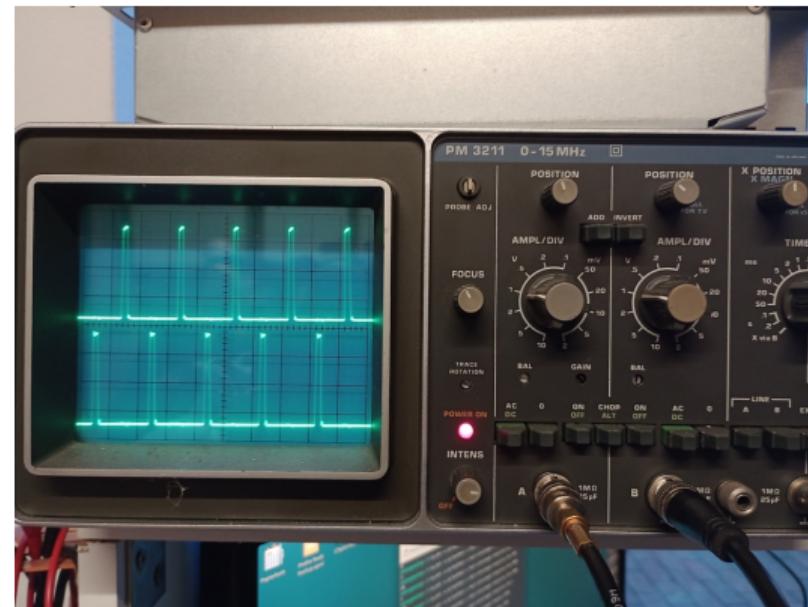
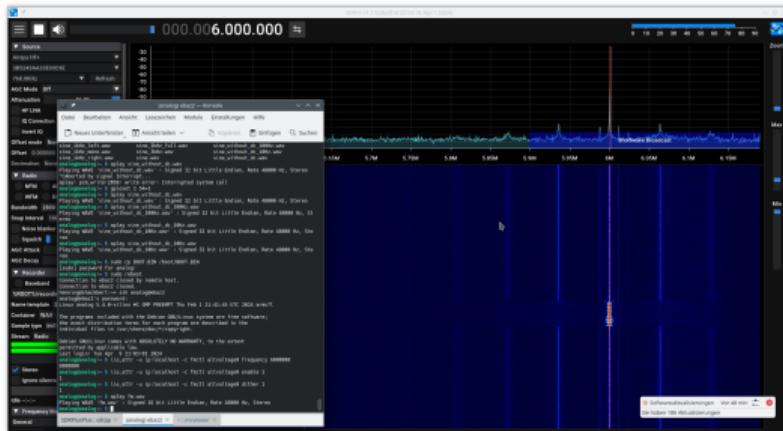
Work in progress (2)

- Frequenzmodulator im FPGA, MOSFETs der schaltenden Endstufe werden einzeln gesteuert



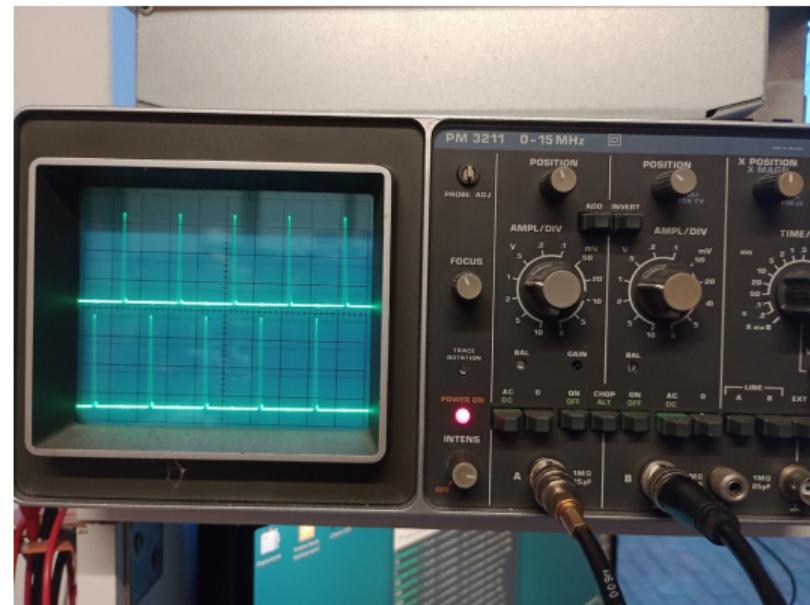
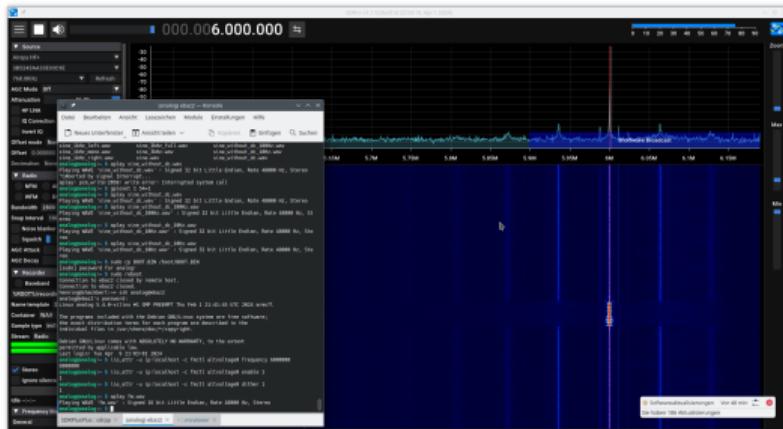
Work in progress (2)

- Frequenzmodulator im FPGA, MOSFETs der schaltenden Endstufe werden einzeln gesteuert



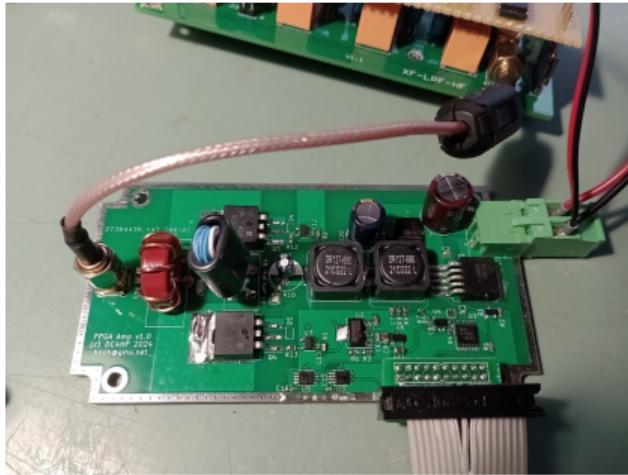
Work in progress (2)

- Frequenzmodulator im FPGA, MOSFETs der schaltenden Endstufe werden einzeln gesteuert



Work in progress (3)

- Brückenspannung für die Digitalendstufe über Step-Down-Wandlers per DAC einstellbar
- Linker Stereokanal bewirkt AM, rechter Stereokanal legt aktuelle Frequenzauslenkung fest
- Zur Zeit noch Vorverarbeitung notwendig



Vielen Dank für eure Aufmerksamkeit!