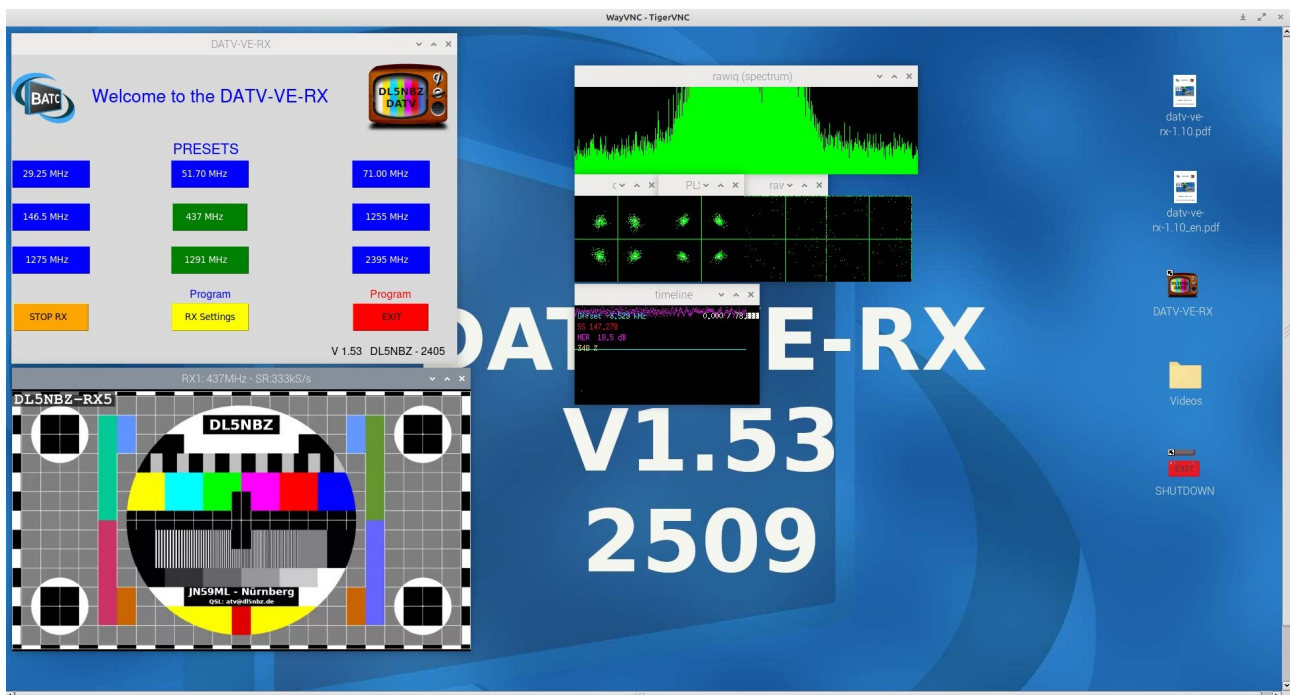




DATV-VE-RX



DL5NBZ - 2405 - V1.53
September 2025

Inhaltsverzeichnis

Vorwort.....	3
1. Konzept.....	4
Aufbau und Baugruppen.....	5
Funktionsbeschreibung.....	6
Technische Daten.....	7
Hinweise zur Software.....	7
Bestandteile des System und Lizenzen.....	8
2. Beschreibung der Software.....	9
1. datv-ve-rx.py.....	9
2. settings.py.....	10
SDR-Parameter.....	10
RX-Parameter.....	11
RX-OPTIONS.....	11
STREAMING.....	12
EDIT PRESET-No.....	13
Anzeigen der Daten aus der Service Description Table (SDT).....	14
3. basic_conf.py.....	15
Auto Start Channel (0-9):.....	15
Init Time s (1-9):.....	15
Selectivity:.....	16
Receiver No (1-9):.....	16
RX-NAME-:.....	16
Button DATV-VE-RX:.....	16
Button RX-SETTINGS:.....	16
4. presets.csv.....	17
5. Virtuelle Tastatur.....	17
6. VNC-Server.....	18
7. FTP-Server.....	18
8. Installation der Software auf dem Raspi 5 oder Raspi4.....	19
Entwicklungsfortschritt.....	20
QO-100-Empfang.....	22
Appendix.....	23
Unterstützte RTL-SDR Hardware.....	23

Rainer Floesser, DL5NBZ

Flensburger Strasse 6

D-90427 Nuernberg

Germany

Tel: +49 911 325466

Mobile: +49 170 3870810

Fax: +49 911 325418

eMail: dl5nbz@dl5nbz.de

Vorwort

Jede gute Lösung ist einfach

Als ich im Mai 2024 das erste Mal Bilder auf 23cm mit dem DATV-VE-RX in der Version 0.90 empfangen habe, war mir noch nicht klar, was ich da losgetreten habe. Nachdem ich das Projekt auf der HamRadio 2024 der AGAF gezeigt hatte, sollte ich es unvorbereitet im internationalen DATV-Forum vorstellen. Wenn ich mich heute daran erinnere, sehe ich eine Traube von Leuten um mich herum, die ungläubig auf die einzelnen Komponenten auf dem Tisch schauten, wo in einem kleinen Display ein DATV-Signal aus meinem tragbaren DATV-Sender zu sehen war. In der Ausgabe 11/24 der CQ/DL erschien dann als Titelstory die Beschreibung zur Version 1.03, Innerhalb kurzer Zeit erhielt ich viele positive Rückmeldung und Erfahrungsberichte. Einige dieser Hinweise haben mich dazu veranlasst an manchen Stellen des Projekts noch ein „Fein-Tuning“ zu machen. Das Ergebnis war, das nun etwa alle 6 Monate ein Update kam. Nun liegt die Version 1.53 vor, die einige Neuerungen mit bringt. Ohne die qualifizierten Kommentare und Hinweise meiner Tester und Nutzer wäre das nicht möglich gewesen. Vielen Dank dafür!

Inzwischen habe ich gelernt, dass der Empfänger in der ersten Liga mitspielt, obwohl er deutlich kostengünstiger ist. Aber ob das in Zukunft mit Updates alle 6 Monate noch machbar ist, ist zweifelhaft. Die Luft wird dünner.

In der Version 1.53 bin ich das Problem der Selektion angegangen, die jetzt in zwei Stufen schaltbar ist. Stufe 0 ist wie bisher, Stufe 1 nähert sich der tatsächlichen Bandbreite eines Signals an. Diese Stufe 1 sollte aber zunächst als Feldtest verstanden sein.

Ich habe auch versucht die Benutzeroberfläche etwas zu ordnen und übersichtlicher zu machen.

Hier noch ein wichtiger Hinweis: Der Grundgedanke war das Projekt mit dem Raspberry PI 5 zu machen, um bei der Leistung zukunftssicher zu sein. Inzwischen hat sich herausgestellt, dass bei portablen Einsatz der Raspberry PI 4 besser geeignet ist. Der Raspi4 benötigt nicht eine Stromversorgung mit „Power Delivery“ wie der Raspi5 und kann deshalb mit jeder Power Bank betrieben werden. Die beschriebene Software läuft ohne Änderungen auf beiden Modellen.

1. Konzept

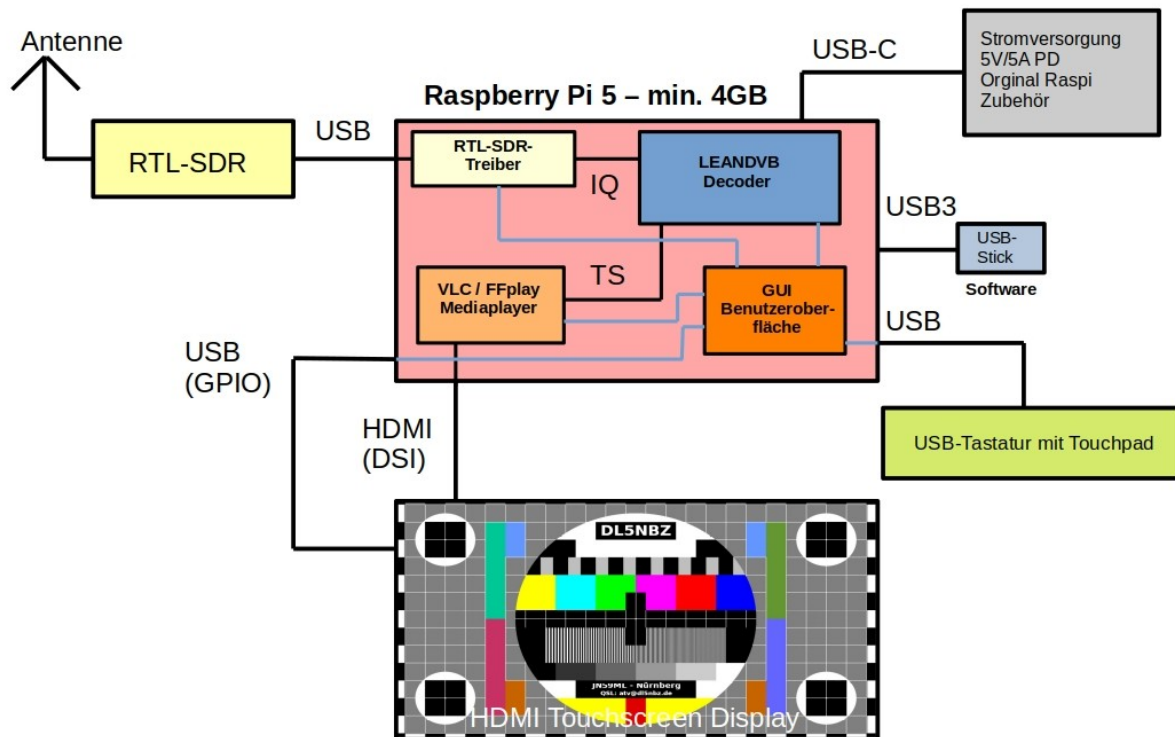
In Deutschland ist das niedrigste, nutzbare Frequenzband für DATV 435 MHz (70 cm), als weiteres Frequenzband bietet sich 1275 MHz (23 cm) an. Höhere Frequenzen erfordern spezielle Antennen und waren meiner Meinung nach für das Projekt nicht im Fokus. Diskrete Empfänger für 70 cm und 23 cm erfordern besondere Leiterplatten und SMD-Bauteile, die ein gewisses Können beim Löten erfordern. Deshalb schloss ich so eine Lösung aus.

Da ich bereits Erfahrungen mit den RTL-SDR-Sticks hatte und auch noch mehrere in der Bastelkiste liegen hatte, lag es nahe, damit erste Versuche zu machen. Die Verarbeitung der Signale aus dem RTL-SDR sollte in einem Raspberry Pi erfolgen. Die Bildwiedergabe kann auch in einem Fernseher oder Monitor mit HDMI-Eingang erfolgen. Ein Display am Raspi über die DSI-Schnittstelle ist auch möglich.

Folgende Komponenten sind nötig:

- der **RTL-SDR-Stick** – **bevorzugt das Original Version 3** (RTL-SDR.COM) oder ein kompatibler Stick (der berühmte blaue DVB-T-Stick). Die limitierte Version 4 wird nun mit neuen Treiber richtig unterstützt. Das Original hat ein Metallgehäuse und schützt besser vor Einstrahlungen des Rechners.
- ein **Raspberry Pi5** (min. 4 GB) – da das Konzept mehrere Jahre Bestand haben sollte, habe ich mich für das aktuelle Modell mit dem aktuellen Betriebssystem „Debian Bookworm“ (Debian 12) entschieden. Für portable Anwendungen eignet sich der Raspi4 mehr, da er aus einer normalen Powerbank betrieben werden kann. Der Raspi5 benötigt bei der Stromversorgung „Power Delivery“, was nicht alle Powerbanks beherrschen, zudem ist der Stromverbrauch höher.
- ein **Display** – min. 1024x600 Pixel, gerne auch als Touch-Screen, alternativ ein Fernseher oder Monitor mit HDMI-Eingang, dann aber mit Maus und Tastatur. Displays kleiner 7“ werden nicht mehr unterstützt.
- Eine geeignete **Stromversorgung** für den Raspi5 (**Achtung:** 5V/27W mit Power-Management)
- Ausreichende **Kühlung** des Raspi5, der Standard-Lüfter mit Kühlkörper reicht vollkommen.
- die nötige Software, basierend auf **Raspian Bookworm** (Basis Debian 12) mit der Empfangssoftware mit Benutzeroberfläche. Liegt als Image vor.

Aufbau und Baugruppen



DATV-VE-RX – Baugruppen und Verschaltung

Der DATV-VE-RX wird aus fertigen Baugruppen zusammen gesteckt. Für die Verbindung der einzelnen Baugruppen werden handelsübliche Kabel verwendet. Die Bedienung ist über den Touch-Screen möglich. Bei Verwendung eines Fernsehers oder Monitors am HDMI-Anschluss kommt man um die USB-Tastatur mit Touchpad oder Maus nicht herum. Den Zusammenbau und die Verbindung der einzelnen Baugruppen zeigt das folgende Schema.

Hinweis: die Software (Betriebssystem, Empfangssoftware und Benutzeroberfläche) befinden sich auf einem USB-Stick (bevorzugt USB3 an einem der USB3-Anschlüsse). Ab dem Raspi4 ist ein Booten auch von USB ab Werk möglich, wenn keine MicroSD-Karte eingesteckt ist. Die Methode mit dem USB-Stick ist wesentlich robuster. MicroSD-Karten sind viel empfindlicher und neigen bei rauer Behandlung zu Ausfällen. Sollte man auf den vierten USB-Anschluss angewiesen sein, bleibt die Verwendung der MicroSD-Karte oder einer SSD mit Adapter-Karten.

Beim Raspi5 ist auch ein Booten von einer NVME-SSD, die an PCIe angeschlossen werden kann, möglich. Dazu muss die Boot-Reihenfolge im EEPROM des Raspi5 geändert werden.

Funktionsbeschreibung

Die Software besteht aus folgenden Teilen:

- Das Linux-Betriebssystem auf der Basis von **Debian Bookworm**
- Die graphische **Benutzeroberfläche (GUI)** in **Python3** programmiert
- Das Programm **RTL_SDR** als Treiber und Empfänger für den RTL-SDR-Stick
- Das Programm **leandvb** als Decoder des DVB-S/DVB-S2 Transportstreams
- Das Programm **vlc bzw. ffplay** zur Umwandlung des Transportstreams in einen MPG-2, MPG-4, AAC-Stream und dessen Anzeige bzw. Audio-Wiedergabe.

Nach dem Booten des Betriebssystems (Linux Debian Bookworm) in den Desktop wird ein in Python3 geschriebenes Graphic User Interface (GUI) gestartet (datv-ve-rx.py).

Es stehen zunächst 9 voreingestellte Frequenzen zur Verfügung (Presets).

Nach Drücken eines der Preset-Buttons in der GUI wird die gewählte Frequenz an rtl_sdr und leandvb übergeben und der Empfänger gestartet.

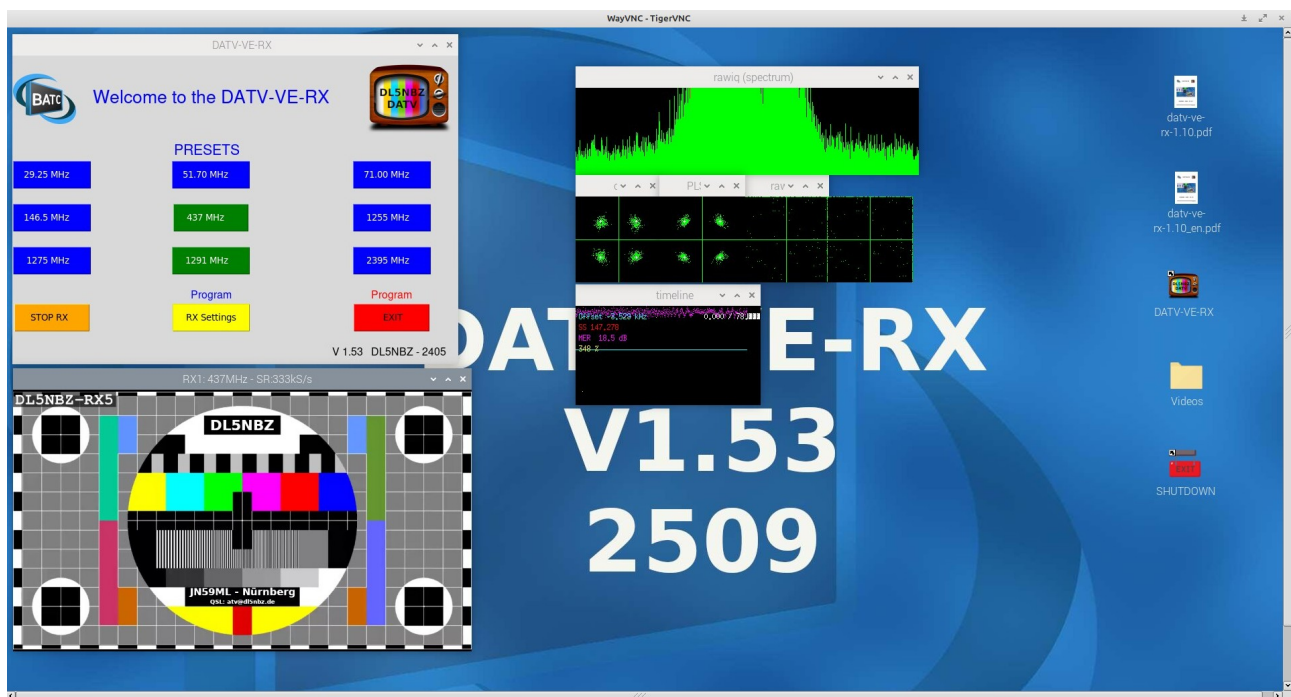
rtl_sdr bekommt die Abtastrate, die Verstärkung und die Empfangsfrequenz übergeben und wird gestartet, danach steht ein IQ-Signal bereit.

leandvb bekommt die Symbolrate, das FEC, die Betriebsart (DVB-S/DVB-S2) und einige andere Parameter, mit denen die Ausgabe gesteuert wird, übergeben. Standardmäßig öffnen sich nach Start von **leandvb** fünf Fenster (zwei Spektrumsanzeigen, zwei Constellationsanzeigen und eine Anzeige der Empfangsdaten). Bei kleinen Displays verdecken sich manche Anzeigen, die wichtigen sind immer oben. Das ist so gewollt, um die Anzeige übersichtlich zu halten.

Wird ein gültiges Signal empfangen, wird **ffplay (oder vlc)** gestartet und das Bild in einem weiteren Fenster angezeigt. Der Ton steht auf dem HDMI-Signal bereit. Hat der HDMI-Monitor kein Audio, muss man entweder einen Audio-HAT oder eine USB-Audio-Karte anschließen, um den Ton wiederzugeben. Es ist auch möglich ein BT-Headset oder einen BT-Lautsprecher mit dem Raspi zu verbinden und so den Ton wiederzugeben.

Häufig genutzte Frequenzen können durch grüne Buttons gekennzeichnet werden.

Über den gelben Button „RX-Settings“ wird ein weiteres Fenster geöffnet, in dem beliebige Frequenzen und andere Parameter eingegeben werden können. Sind alle Parameter eingegeben, kann man den Empfänger durch drücken des grünen „Start“-Buttons starten.



Technische Daten

Empfänger (RTL-SDR-Stick V3)

Frequenzbereich: 24 MHz bis 1766 MHz

Maximale Empfängerbandbreite: 3,2 MHz max. / 2,56 MHz stabil

Modulationsarten: QPSK, 8PSK, 16APSK, 32APSK – selbst erkennend

Symbolraten: 18 kS/s – 2000 kS/s

Standards: DVB-S, DVB-S2

Der mögliche Frequenzbereich ist abhängig vom Tuner des Sticks. Der RTL-SDR-Stick V3 ist für diesen Empfänger am Besten geeignet, da er in einem Metallgehäuse eingebaut ist und über einen TXCO verfügt.

Tuner	Frequenzbereich
Elonics E4000	52 - 2200 MHz mit einer Lücke von 1100 MHz bis 1250 MHz (unterschiedlich)
Rafael Micro R820T	24 - 1766 MHz
Rafael Micro R828D	24 - 1766 MHz
Fitipower FC0013	22 - 1100 MHz (FC0013B/C, FC0013G haben einen extra L-band Eingang, der bei den meisten Sticks nicht angeschlossen ist)
Fitipower FC0012	22 - 948.6 MHz
FCI FC2580	146 - 308 MHz und 438 - 924 MHz (mit einer Lücke dazwischen)

Hinweise zur Software

Der DATV-VE-RX besteht aus mehreren Software-Paketen. Nur im

Zusammenspiel dieser einzelnen Pakete funktioniert der Empfänger so wie er soll. Die einzelnen Pakete erfordern Abhängigkeiten im Betriebssystem (Raspian OS auf Basis Bookworm), die installiert sein müssen. Alles in Allem ist der Empfänger ein komplexes Softwaresystem, das richtig installiert sein muss, um zu funktionieren.

Der Raspi5 besitzt einen ARM-Prozessor, der einen anderen Befehlssatz als die gängigen INTEL- oder AMD-Prozessoren hat. Deshalb müssen einige Teile der Software auf dem Raspi5 kompiliert werden, um auf diesem Prozessor zu laufen. Auch dafür sind einige Abhängigkeiten erforderlich.

Aus diesem Grund habe ich mich entschieden, kein Installationsskript zu veröffentlichen, sondern nur ein komplettes Image des Systems. Ich erhoffe dadurch eine einfachere Inbetriebnahme des Empfängers. Die Zeit, die ich so am Support einsparen kann, kann ich sinnvoller in die Weiterentwicklung stecken.

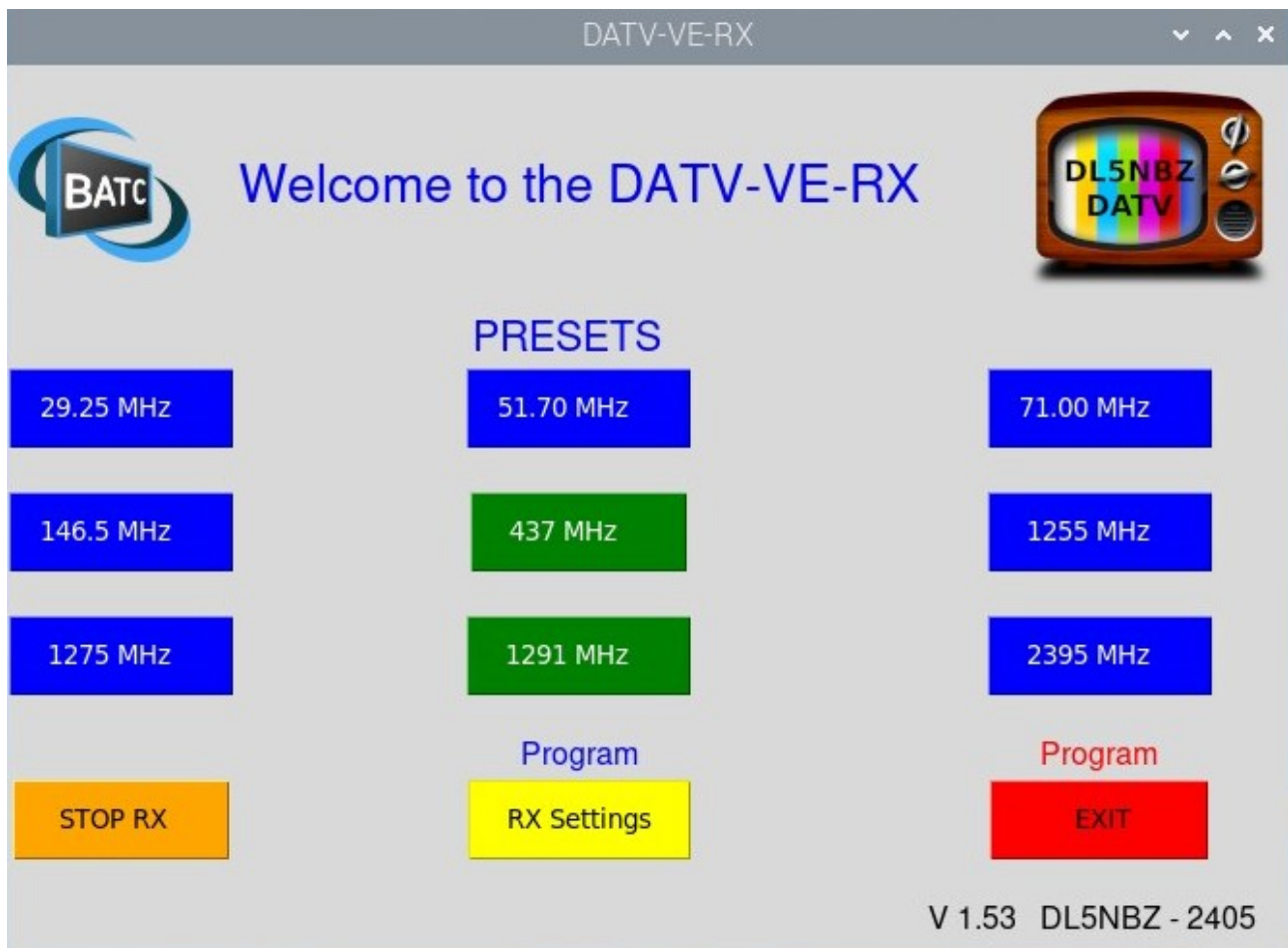
Grundsätzlich ist es auch möglich, den Empfänger auf einem Laptop zu installieren. Voraussetzung ist auch hier ein Betriebssystem auf der Basis von Debian 12 Bookworm (i.e. Ubuntu, Linux Mint, usw). Es ist nicht auszuschließen, dass ich irgendwann dafür ein Installationsskript erstelle. Dieses Projekt hat aber derzeit eine sehr niedrige Priorität. Eine Windows-Version wird es definitiv nicht geben.

Bestandteile des System und Lizenzen

1. RASPIAN OS Bookworm
Raspberry PI Foundation, freie und Open-Source-Software-Lizenzen, hauptsächlich GPL
2. RTL_SDR
osmocom.org, Open-Source-Software
3. ffmpeg (ffplay)
ffmpeg.org, GNU Lesser General Public Licence (LGPL) version 2.1
4. leandvb
pabr.org, F5OEO, K4KDR, W6RZ, G4GUO, AMSAT-DL, BATC, GNU General Public License version 3
5. Python3
Python Software Foundation, GPL

2. Beschreibung der Software

1. datv-ve-rx.py



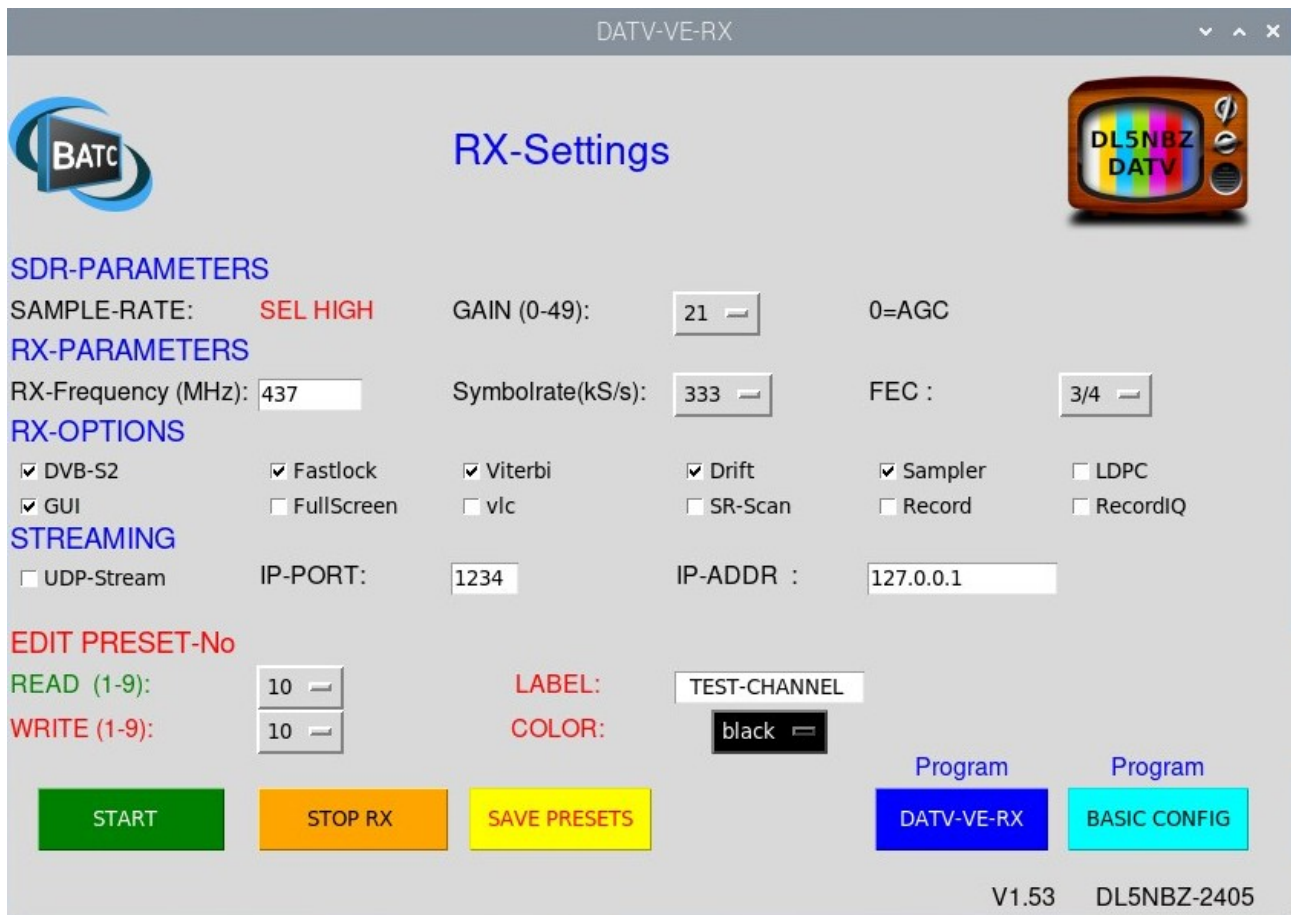
datv-ve-rx.py ist die zentrale Benutzeroberfläche. Sie wird nach Start des RasPi5 automatisch gestartet. Es zeigt sich dann folgende Maske (siehe oben): Hier stehen neun voreingestellte Frequenzen zur Verfügung. Die Parameter dieser Frequenzen sind in der Datei **presets.csv** gespeichert. Ist diese Datei nicht vorhanden, wird sie beim Schließen des Programms über den Button **EXIT** mit Defaultwerten angelegt.

Häufig benutzte Frequenzen können mit einem grünen Button-Hintergrund markiert werden. Durch klicken auf einen der neun Frequenz-Buttons wird der Empfänger mit den hinterlegten Parametern für diese Frequenz gestartet. Mit einem Klick auf **STOP RX** kann der Empfänger wieder deaktiviert werden.

Durch klicken auf den Button **RX Settings** gelangt man in ein Programm, mit dem man eine selbst gewählte Frequenz eingeben und starten kann. Ebenfalls kann man hier PRESETS ändern und bearbeiten.

Mit dem Button **EXIT** werden alle Presets in der Datei **presets.csv** gespeichert und das Programm beendet. Das Programm **datv-ve-rx.py** hat den geplanten Funktionsumfang erreicht.

2. settings.py



Das Bild zeigt die Version 1.53 des Programms **setting.py**. Die Version ermöglicht, eine Frequenz und alle nötigen Parameter von Hand einzugeben. Die Eingabe-Maske ist in vier Bereiche unterteilt:

SDR-Parameter

Hier wird die Verstärkung des Empfangsteils eingestellt.

Sample-Rate: in der Version 1.53 muss die Samplerate nicht mehr eingestellt werden. Sie wird aus der Symbolrate ermittelt. Die Samplerate bestimmt maßgeblich die Bandbreite und Selektivität des Empfängers. Wird bei SAMPLE-RATE **SEL STANDARD** angezeigt ist alles wie in früheren Versionen, bei **SEL HIGH** wird die Bandbreite an die Symbolrate angepasst. Umgeschaltet wird das in der **basig_conf.py**

GAIN: Die Grundeinstellung für GAIN ist 0. Mit dieser Einstellung übernimmt die automatische Verstärkungsregelung (AGC) des RTL-SDR-Stick. Sollte das Grundrauschen (Noise Floor) zu stark sein, kann man die Verstärkung zu kleineren Werten (1-49) aus einer Tabelle in 2dB Schritten ändern. Damit sollte das Rauschen geringer werden und den Empfangskanal nicht mehr so stark

stören. Die beiden Spektrum-Anzeigen nach Start des Empfängers geben Aufschluss darüber.

RX-Parameter

Hier wird zunächst die **Empfangsfrequenz** in MHz eingegeben. Bei der Eingabe von „ungeraden“ Frequenzen ist der Dezimal-Punkt zu verwenden (z.B. 51.7). Die **Symbol-Rate** wird in kS/s (kiloSymbols/Sekunde) eingegeben. Hier werden die gängige Werte (125 kS/s, 250 kS/s, 333 kS/s, 500 kS/s, 1000 kS/s, usw.) angeboten.

Die **Fehlerkorrektur (FEC)** wird als Bruch eingegeben. Je nach System (DVB-S / DVB-S2) werden gültige Werte in einer Tabelle angeboten. Bei DVB-S2 erkennt der Empfänger den richtigen Wert selbstständig.

Die derzeitige Fehlerkorrektur erfüllt nicht den Standard für DVB-S/DVB-S2. Das eingebaute LDPC-Filter ist einen großen Schritt vom Ziel entfernt. Ein externes LDPC-Filter konnte ich bisher auf dem ARM-Prozessor noch nicht kompilieren.

Anmerkung: LDPC steht für **Low Density Parity Check**.

RX-OPTIONS

Weiter gibt es noch einige **CheckBoxes**, die aber in der Version 1.53 noch nicht alle aktiviert sind. Folgende CheckBoxes sind freigeschaltet:

DVB-S2: Standardmäßig arbeitet der Empfänger im DVB-S-Standard. Will man DVB-S2 empfangen, muss diese Box getickt sein. Bei DVB-S-Empfang muss zwingend ein korrektes FEC eingegeben sein. Es sind folgende FECs erlaubt: $\frac{1}{2}$, $\frac{2}{3}$, $\frac{3}{4}$. Bei DVB-S2 findet der Empfänger selbst das richtige FEC, eventuelle Einstellungen werden ignoriert. Bei DVB-S ist ein FASTLOCK nicht möglich und kann nicht getickt werden.

FASTLOCK: Ein rechenintensiver Algorithmus, der ein schnelleres Synchronisieren auf das Empfangssignal erlaubt.

VITERBI: ein dynamisches Demodulationsverfahren für optimale Leistung auch auf verschiedenen Plattformen (rechenintensiv)

DRIFT: gleicht Frequenzversatz aus. Hilfreich beim Empfang der ISS, um die Dopplerverschiebung zu kompensieren. Ab V1.02 freigeschaltet.

LDPC: Siehe im Abschnitt Fehlerkorrektur. Hier kann später der externe LDPC-Filter aktiviert werden. In V1.53 noch ohne Funktion.

GUI: Nach dem Starten des Empfängers machen zwei Spektrum-Anzeigen und zwei Constellations-Anzeigen und ein Fenster mit dem zeitlichen Verlauf des Empfangssignals auf. Diese Anzeigen sind hilfreich zum justieren der Antenne und der Einstellung der Empfangsverstärkung. Ist diese Box getickt, startet der Empfänger diese Anzeigen. Bei festen und getesteten Strecken ist das unnötig und kann weggelassen werden.

Sampler: Getickt aktiviert sich ein rechenintensives Bandpassfilter auf der Eingangsfrequenz.

FullScreen: Ist diese Box getickt, wird das empfangene Bild Bildschirmfüllend angezeigt.

VLC: Standardmäßig stellt der Empfänger Signale mit dem Mediaplayer **FFPLAY** dar. Mit dem **FFPLAY** werden Signalen innerhalb von 15 Sekunden nach einrasten des Empfängers angezeigt. **FFPLAY** verfügt aber über weniger Möglichkeiten als der VLC. Durch ticken dieser Box wird mit dem VLC angezeigt und nicht mit FFPLAY. Bei Verwendung des VLC können bei laufenden Empfang die Daten der Service Description Table (SDT) unter Werkzeuge/Codec-Informationen angezeigt werden.

SR-Scan: Hiermit wird der Scanner für die Symbolraten aktiviert. Beim Start des Empfängers werden alle gängigen Symbolraten (125kS/s – 1000kS/s) durchprobiert, bis das passende SR gefunden wird. **Achtung:** Wenn SR-Scan wieder abgewählt wurde, muss wieder eine Symbolrate aus der Liste ausgewählt werden, sonst startet der Empfänger ins Leere.

RECORD: Ist diese Box getickt, wird der empfangene Transportstream aufgezeichnet. Es wird eine Datei **record-YY-MM-DD-HH-MM-SS.ts** in das Verzeichnis **/home/pi/Desktop/Videos/** geschrieben. Das Verzeichnis ist vom Desktop aus zu erreichen.

RECORDIQ: Ist diese Box getickt, wird der empfangene IQ-Datenstrom aufgezeichnet, um ihn später mit geeigneter Software auszuwerten. Es wird eine Datei **record-YY-MM-DD-HH-MM-SS.iq** in das Verzeichnis **/home/pi/Desktop/Videos/** geschrieben. Das Verzeichnis ist vom Desktop aus zu erreichen. **ACHTUNG!: Bei der Aufzeichnung des IQ-Signals entstehen sehr große Dateien. Deshalb diese Funktion mit Bedacht benutzen.**

STREAMING

Ab der Version V1.10 kann der Empfänger den empfangenen Transportstrom via UDP an einen anderen Computer streamen. Auf dem anderen Computer kann man das Bild dann z.B. mit dem VLC darstellen (Medien/Netzwerkstream öffnen `udp://@:1234`).

UDP-Stream: Durch ticken wird das Streaming aktiviert.

IP-PORT: Hier wird der Port für das Streaming festgelegt. Der Default-Wert ist 1234

IP-ADDR: Hier wird die IP-Adresse des Computers im Netzwerk eingegeben, auf dem der Transportstrom angeschaut werden soll. Der Default-Wert ist die eigene Adresse des Empfängers (127.0.0.1). Während des Streamens wird auf dem Empfänger selbst kein Bild dargestellt.

EDIT PRESET-No

Die Eingabemaske des Programms zeigt beim Programmstart nur die Daten des „Presets 10“, der vom Programm **datv-ve-rx.py** nicht erreichbar ist. Dieser Programmplatz 10 ist zum Testen, Einstellen und Bearbeiten der Parameter gedacht. Eingegebene Parameter können getestet werden. Dazu klickt man auf den Button **START** und der Empfänger startet mit den eingegeben Parametern. Mit **STOP RX** beendet man den Empfang. Dort kommt man mit **RX SETTINGS** zurück in die Eingabemaske der Empfangsparameter. Die vorher gemachten Eingaben wurden auf dem Programmplatz 10 gespeichert und sind als Default-Werte wieder in der Eingabemaske zu sehen.

Möchte man diese Parameter auf einen Button (Programmplatz) speichern, gibt man bei **LABEL** einen Namen für den PRESET Button ein, mit **COLOR** wird die Hintergrundfarbe des Buttons ausgewählt. Nun öffnet man die Auswahlliste unter **WRITE (1-9)** und wählt den Programmplatz (Button) aus, auf dem die Parameter gespeichert werden. Hat man alle Einstellungen noch einmal überprüft, klickt man auf **SAVE PRESETS**. Damit ist der Programmplatz neu belegt und datv-ve-rx.py wird wieder mit den neuen Daten gestartet.

Man kann auch Programmplätze bearbeiten und ändern. Dazu holt man sich den zu bearbeitenden Programmplatz auf den Programmplatz 10, indem man in der Auswahlliste **READ (1-9)** den gewünschten Programmplatz auswählt und anklickt. Die Eingabemaske wird neu aufgebaut und die Parameter des ausgewählten Programmplatzes können in der Maske bearbeitet werden. Das Abspeichern der Daten funktioniert wie bereits oben beschrieben.

Button START: Beim Drücken auf diesen Button werden die eingegebenen Daten zu einem Start-Kommando für den Empfänger verarbeitet. Dieses Start-Kommando wird an Position 10 der **presets.csv** geschrieben und der Empfänger gestartet.

Button STOP RX: Beim Drücken werden alle Daten in der **presets.csv** gesichert, alle Tasks beendet.

Button DATV-VE-RX: Beim Drücken kehrt man in das Programm DATV-VE-RX zurück.

Button BASIC_CONF: Beim drücken startet man das Programm BASIC_CONF

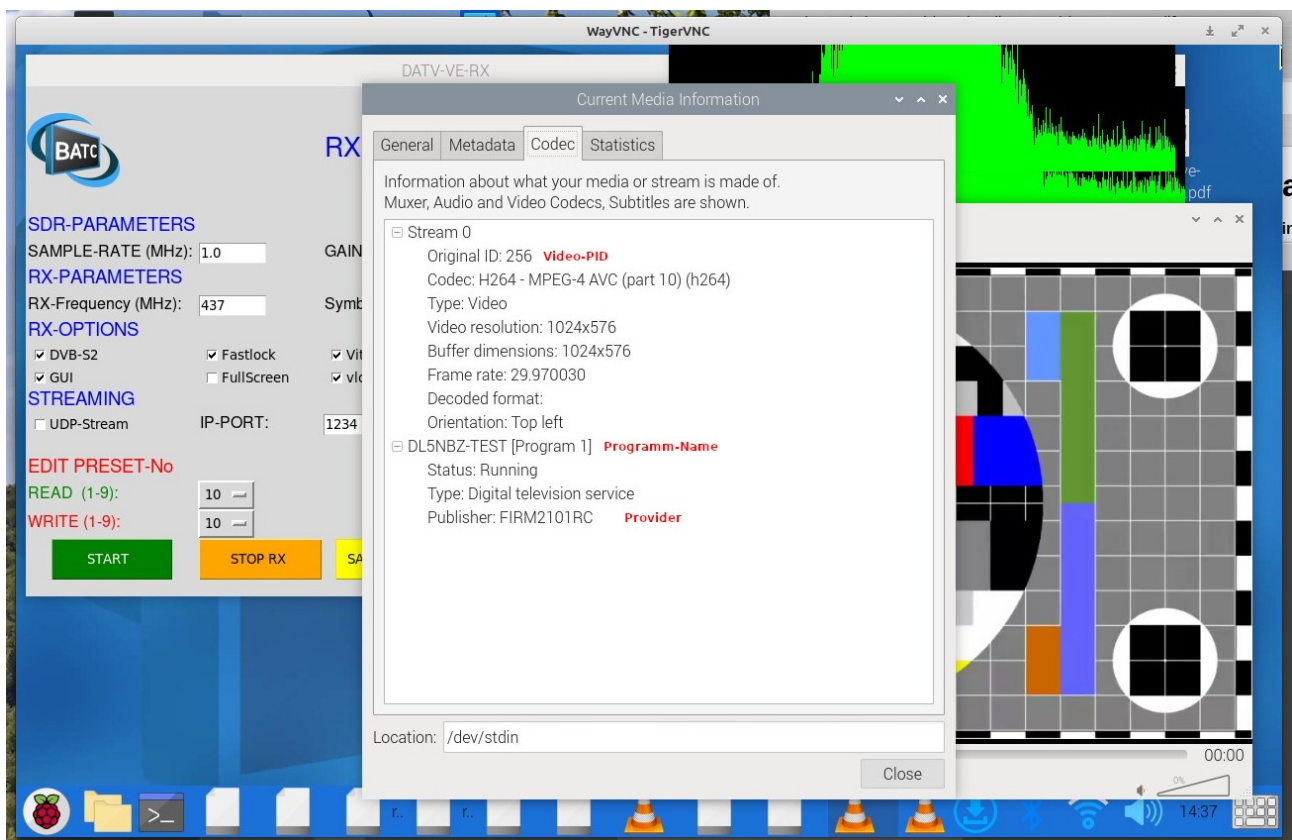


Schaubild 1: Verschiedene RTL-SDR-Sticks

Anzeigen der Daten aus der Service Description Table (SDT)

Bei einer Aussendung im DVB-Format wird bei richtiger Konfiguration des Senders das Rufzeichen und andere Informationen über verwendete Codecs in digitaler Form mit ausgestrahlt wird, kann man das Rufzeichen, bei fehlenden Rufzeichen im Bildinhalt, aus der **Service Description Table (SDT)** ermitteln.

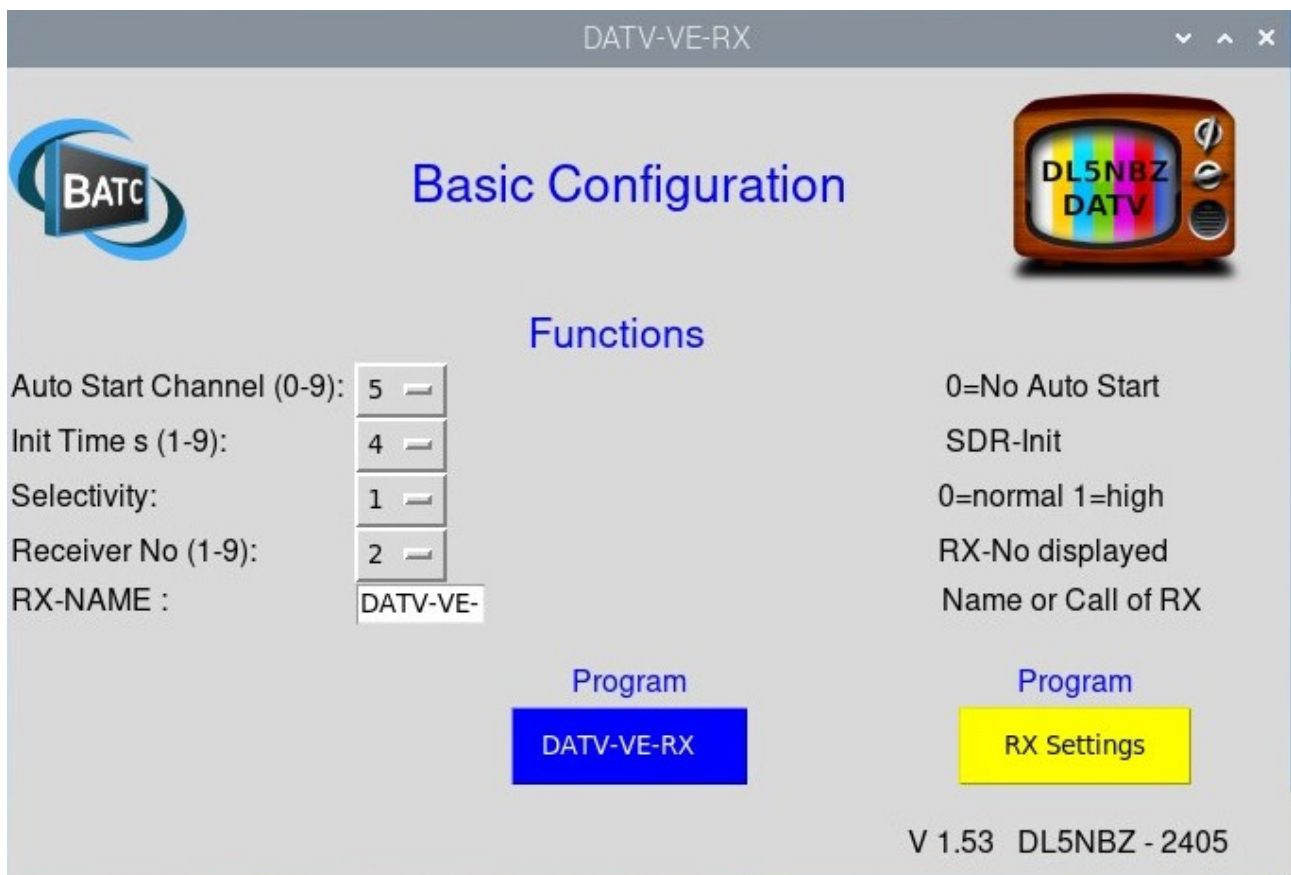
Um mir den Aufwand des Extrahierens aus dem Transportstrom zu sparen, nutze ich dafür die Funktion des **VLC**. Um den **VLC** als Media-Player zum Empfang zu nutzen, muss in den **SETTINGS** „**VLC**“ getickt sein. Beim Empfang wird die Sendung mit dem VLC wiedergeben. Zur Anzeige der SDT-Daten beim laufenden Empfang klickt man im VLC-Fenster auf „Tools“ und dann auf „Codec-Information“. Danach geht folgendes Fenster auf:



Im Reiter „Codec“ findet man dann die interessanten Daten.

Neu in Version V1.53

3. basic_conf.py



Auto Start Channel (0-9):

Hier kann einer der neun voreingestellten Kanäle als Auto-Start-Kanal ausgewählt werden. Die Zuordnung folgt einer Telefontastatur. Ist 0 ausgewählt findet kein Auto-Start statt.

1	2	3
4	5	6
7	8	9

Init Time s (1-9):

Hier kann die Zeitdauer der Initialisierung der RTL-SDR-Sticks in Sekunden eingestellt werden. Der Wert ist abhängig von der Hardware und muss durch probieren gefunden werden. Ein guter Startwert ist 4. Für die Zeitdauer der Initialisierung wird im auslösenden Programm oben ein roter Balken angezeigt.

Selectivity:

Hier kann der Empfänger zwischen Standard Selektivität (0) und hoher Selektivität (1) umgeschaltet werden. Für die Nutzung der hohen Selektivität ist die Initialisierung zwingend erforderlich.

Receiver No (1-9):

Hier kann eine Empfängernummer eingestellt werden bei Anzeige mit ffplay. Das ist Vorteilhaft, wenn man mehrere Empfänger parallel betreibt.

RX-NAME-:

Zusätzlich zur Empfängernummer kann ein hier eingetragener Name vor der Empfängernummer angezeigt werden. Ist dieses Feld leer, wird nur die Empfängernummer angezeigt.

Die Daten werden kodiert zum Datenaustausch mit den anderen Programmteilen in der ersten Zeile der **presets.csv** gespeichert.

Button DATV-VE-RX:

Durch drücken kommt man zum Programmteil DATV-VE-RX

Button RX-SETTINGS:

Durch drücken kommt man zum Programmteil RX-SETTING

Hinweis: Die Änderungen in **basic_conf.py** sind sofort in **settings.py** verfügbar und können dort sofort genutzt werden. Erst nach Abspeichern in die neuen Preset-Kanäle sind sie in **datv-ve-rx.py** wirksam.

4. presets.csv

In der Datei **resets.csv** werden alle zum Empfang benötigten Parameter gespeichert. Die Datei ist im **CSV-Format** erstellt und besteht aus 11 Zeilen mit 3 Spalten. Als Spalten-Trenner wird der Tabulator verwendet.

Die erste Zeile enthält Daten für die einzelnen Programmteile. Die erste Spalte enthält die **LABEL-Namen**, also die Beschriftung der Buttons.

Die zweite Spalte enthält die **Hintergrundfarbe** der Buttons.

In der dritten Spalte findet sich das **Betriebssystem-Kommando**, um den Empfänger, den Decoder und den Mediaplayer zu starten. Hier werden zahlreiche Parameter mit übergeben. Der Syntax ist umfangreich und komplex. Ohne genaues Wissen über diese Programmteile sollte man die Finger von Änderungen lassen. Das Programm **settings.py** erledigt das viel zuverlässiger. Sollte man die Datei **presets.csv** vermurkst haben, ist der sicherste Weg, die Datei zu löschen. Beim nächsten Programmstart wird sie wieder mit Default-Werten neu angelegt.

Die Datei presets.csv der Version 1.53 ist nicht mehr kompatibel zu früheren Versionen und ist nicht mehr für manuelle Bearbeitung geeignet.

5. Virtuelle Tastatur

Seit der Version V1.10 ist eine virtuelle Tastatur (On-Screen-Keyboard) aktiv. Damit ist nun eine vollständige Bedienung des Empfängers über den Touch-Screen möglich.

In der eigentlichen Empfänger-Benutzeroberfläche konnte man schon immer die Presets über den Touch-Screen starten und beenden. Die virtuelle Tastatur wird nur im Programm settings.py benötigt. Hier kann man in die weissen Eingabefelder durch Touch klicken. Dann öffnet man die virtuelle Tastatur durch Touch auf das Tastatur-Symbol ganz rechts in der Taskleiste. Nachdem man die Eingaben erledigt hat, klickt man auf die Enter-Taste und kann dann die virtuelle Tastatur wieder durch Klick auf das Tastatursymbol wieder ausschalten.

Für eine gute Bedienbarkeit empfehle ich Touch-Displays mit mindestens 10.1“ Größe.

6. VNC-Server

Auf dem DATV-VE-RX ist der VNC-Server standardmäßig aktiviert. Ist der DATV-VE-RX mit dem Heimnetzwerk verbunden, kann man sich mit einem VNC-Viewer von einem beliebigen PC über den VNC-Viewer mit dem DATV-VE-RX verbinden. Dazu benötigt man folgende Daten:

IP-Adresse des DATV-VE-RX, zu finden im eigenen Router oder dem Raspi
Benutzername: **pi**
Passwort: **raspberry**

Nach dem Verbinden mit dem heimischen PC öffnet sich die Benutzeroberfläche des DATV-VE-RX auf dem PC und man kann den Empfänger komplett über das Netzwerk bedienen.

7. FTP-Server

Ab der Version V1.53 läuft ein FTP-Server im Hintergrund des Empfängers mit. Über den FTP-Server kann man Dateien vom Empfänger herunterladen oder hochladen. So bringt man aufgezeichnete Transportstreams oder Snapshots auf den eigenen PC. Als FTP-Client kann man Filezilla einsetzen

IP-Adresse des DATV-VE-RX, zu finden im eigenen Router oder dem Raspi nach dem Start rechts oben in einem grauen Feld.
Benutzername: **pi**
Passwort: **raspberry**
Port: **21**

Nach dem Verbinden landet man automatisch im Verzeichnis **/home/pi/Desktop/Videos**, in dem der Empfänger aufgezeichnete Transportstreams und Snapshots speichert. In andere Verzeichnisse kann man aus Sicherheitsgründen nicht wechseln.

8. Installation der Software auf dem Raspi 5 oder Raspi4

1. Das vollständige Image **datv-rx-153.zip** (ca. 7GB) ist herunterzuladen und mit einem Image-Writer auf eine MicroSD-Karte oder einen (USB3) Stick zu schreiben. Das Medium sollte 32 GB groß sein. Ich empfehle das Programm **Etcher** (etcher.io). Etcher ist für Windows, MacOS und Linux verfügbar. Für das Schreiben des Image mit Etcher muss die Zip-Datei nicht entpackt werden. Nach dem Schreiben des Image auf das Medium sind auf dem Medium zwei Partitionen vorhanden: **bootfs** (FAT32) und **rootfs** (EXT4). Auf Windows-Systemen wird die Partition **rootfs** nicht angezeigt, Windows kann mit dem Dateisystem EXT4 nichts anfangen.

2. Das Medium mit dem Raspi verbinden, mindestens Bildschirm und RTL-SDR-Stick anschließen.

3. Raspi einschalten. Der Raspi bootet bis ins Betriebssystem und fährt sofort wieder herunter. Kurz danach startet er automatisch wieder. Bei diesem Vorgang expandiert die Software die Partition **rootfs** auf die volle verbleibende Größe des Mediums. Danach wird automatisch die Empfängersoftware gestartet. **Während dieses Vorgangs den Raspi keinesfalls ausschalten oder von der Stromversorgung trennen!!!**



4. Es empfiehlt sich, den Raspi mit dem heimischen Netzwerk entweder über LAN oder WLAN zu verbinden.

5. Sollte man zur Eingabe eines Benutzernamens und eines Passworts aufgefordert werden, gelten die Default-Einstellungen von RaspianOS:
Benutzername: **pi** Passwort: **raspberry**

Entwicklungsfortschritt

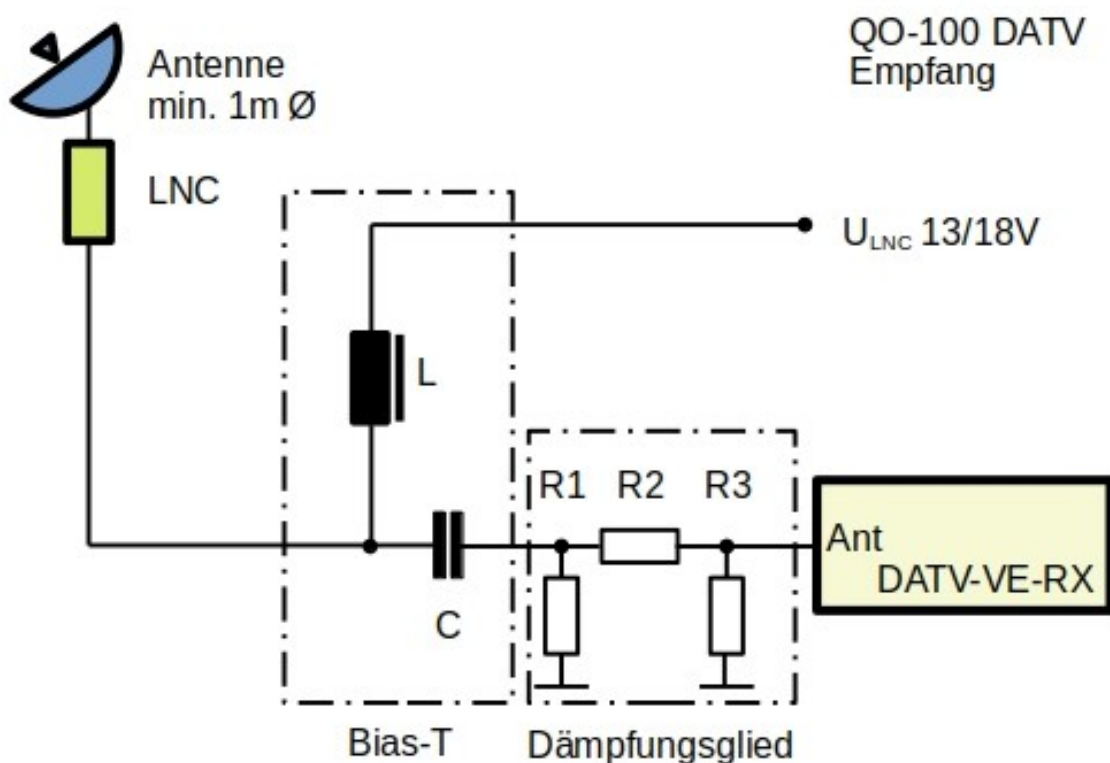
V 0.4	März 24	funktionierender RX, aber keine Anzeige im Terminal
V 0.5	April 24	erste funktionierende RX-Version im Terminal auf dem Desktop
V 0.7	April 24	erste grausame GUI auf dem Desktop
V 0.81	Mai 24	erste halbwegs benutzbare GUI für Presets
V 0.85	Mai 24	zuverlässiger Aufruf von rtl_sdr, leandvb and mplayer aus der Python3 GUI
V 0.90	Mai 24	Presets in einer Datei (presets.csv). Geplanter Funktionsumfang bei datv-ve-rx.py erreicht. Versuche mit dem ffplay sind erfolgversprechend. Mit dem ffplay wird das Video wesentlich schneller angezeigt als mit dem mplayer . Mit settings.py kann nun eine beliebige Frequenz mit beliebigen Parametern eingegeben und gestartet werden. Die Daten werden als zehntes Preset in presets.csv gespeichert. Eine Überprüfung der eingegebenen Daten erfolgt nicht.
V1.01	Mai 24	Programm-Teil datv-ve-rx.py V1.01 mit dem vollen geplanten Funktionsumfang. Prüft Vorhandensein der Datei presets.csv ab; wenn die Datei nicht vorhanden ist, wird sie mit Standard-Werten angelegt. Programm-Teil settings.py mit dem vollen geplanten Funktionsumfang. Einige Options (drift, record ldp) sind noch nicht freigeschaltet. Damit ist das Programm-Paket einsatzfähig.
V1.02	Juni 24	Spektrum- und IQ-Anzeige für kleine Bildschirme angepasst. Funktion drift ist nun freigeschaltet.
V1.03	Juli 24	Für kleine Bildschirme kann das Programm nun mit einem Parameter aufgerufen werden. Für einen 5" Bildschirm startet man das Programm mit ./datv-ve-rx.py 5 . Mit anderen Werten startet das Programm mit den Default-Werten für 7" oder größer. Die record -Funktion ist für Transportstreams freigeschaltet. Die Transportstreamdateien werden im Verzeichnis /home/pi/Desktop/Videos/ gespeichert. Das Verzeichnis ist direkt aus dem Desktop erreichbar. Eine recordiq -Funktion wurde neu hinzugefügt, um beim Empfang die IQ-Daten für eine spätere Bearbeitung aufzuzeichnen. Mit der neuen READ -Funktion im Settings-Programm kann man nun beliebige Programmplätze zum Bearbeiten laden.

V1.10	Jun 25	Virtuelle Tastatur eingeführt. Das decodierte Signal kann per UDP-Stream an andere Rechner gestreamt werden. Der MPLAYER wurde entfernt und VLC als zweiter Media-Player eingesetzt. Mit dem VLC lassen sich auch Screenshots machen. VLC zeigt das Bild auch schneller an. Es kann ein Symbol-Raten-Scan eingesetzt werden. Bei unbekannter Symbol-Rate werden die üblichen Symbol-Raten für DATV gescannt.
V1.53	Sep 25	Möglichkeit zwischen zwei Bandbreiten umzuschalten. Die Bandbreiten sind in Tabellen hinterlegt und können nicht manuell geändert werden. Die Benutzeroberfläche wurde etwas übersichtlicher gemacht. Ein neuer Programmteil wurde hinzugefügt um globale Einstellungen im Programm veränderbar zu machen. Ein FTP-Server wurde auf dem Empfänger aktiviert um Dateien vom Empfänger auf den PC zu laden oder vom PC auf dem Empfänger. Es wurde ein Autostart eingeführt, der nach dem Hochfahren des Empfängers einen der neun Kanäle aus DATV-VE-RX startet. Der RTL-SDR-Treiber wurde auf den neuesten Stand gebracht. Der RTL-SDR-V4 funktioniert nun vollständig mit dem Empfänger

QO-100-Empfang

Der Empfänger lebt von einem guten Signal/Rauschverhältnis. Deshalb sollte die Verstärkung des Empfängers (Gain) so eingestellt sein, dass im Spektrum ein möglichst niedriges Grundrauschen zu sehen ist und das Nutzsignal deutlich im Spektrum erscheint. Das vorgeschaltete LNC hat eine sehr hohe Durchgangsverstärkung und übersteuert den Empfänger besonders bei kurzen Kabellängen zwischen LNC und Empfänger. Die Stromversorgung des LNC wird durch ein Bias-T eingespeist.

Hier eine schematische Darstellung:



Hat das LNC eine zu große Verstärkung und der Empfänger lässt sich mit Verstärkungsregelung nicht weit genug abregeln, muss man ein zusätzliches Dämpfungsglied zwischen Bias-T und den Empfänger einfügen. Der Dämpfungswert richtet sich nach der Verstärkung des LNC.

Für den QO-100 Empfang ist ein Spiegel von mindestens 1 m Durchmesser nötig. Die für den Empfang der Bake nötige Symbolrate 1500 kS/s kann nun direkt ausgewählt werden. Der Empfänger sollte auf dem Transponder mit **SEL HIGH** betrieben werden.

Appendix

Unterstützte RTL-SDR Hardware

Außer den bereits erwähnten originalen RTL-SDR-Sticks sind folgende Sticks dafür bekannt, dass sie mit der RTL-SDR-Software funktionieren:

VID	PID	tuner	device name
0x0bda	0x2832	all of them	Generic RTL2832U (e.g. hama nano)
0x0bda	0x2838	E4000	ezcap USB 2.0 DVB-T/DAB/FM dongle
0x0ccd	0x00a9	FC0012	Terratec Cinergy T Stick Black (rev 1)
0x0ccd	0x00b3	FC0013	Terratec NOXON DAB/DAB+ USB dongle (rev 1)
0x0ccd	0x00d3	E4000	Terratec Cinergy T Stick RC (Rev.3)
0x0ccd	0x00e0	E4000	Terratec NOXON DAB/DAB+ USB dongle (rev 2)
0x185b	0x0620	E4000	Compro Videomate U620F
0x185b	0x0650	E4000	Compro Videomate U650F
0x1f4d	0xb803	FC0012	GTek T803
0x1f4d	0xc803	FC0012	Lifeview LV5TDeluxe
0x1b80	0xd3a4	FC0013	Twintech UT-40
0x1d19	0x1101	FC2580	Dexatek DK DVB-T Dongle (Logilink VG0002A)
0x1d19	0x1102	?	Dexatek DK DVB-T Dongle (MSI DigiVox mini II V3.0)
0x1d19	0x1103	FC2580	Dexatek Technology Ltd. DK 5217 DVB-T Dongle
0x0458	0x707f	?	Genius TVGo DVB-T03 USB dongle (Ver. B)
0x1b80	0xd393	FC0012	GIGABYTE GT-U7300
0x1b80	0xd394	?	DIKOM USB-DVBT HD
0x1b80	0xd395	FC0012	Peak 102569AGPK
0x1b80	0xd39d	FC0012	SVEON STV20 DVB-T USB & FM