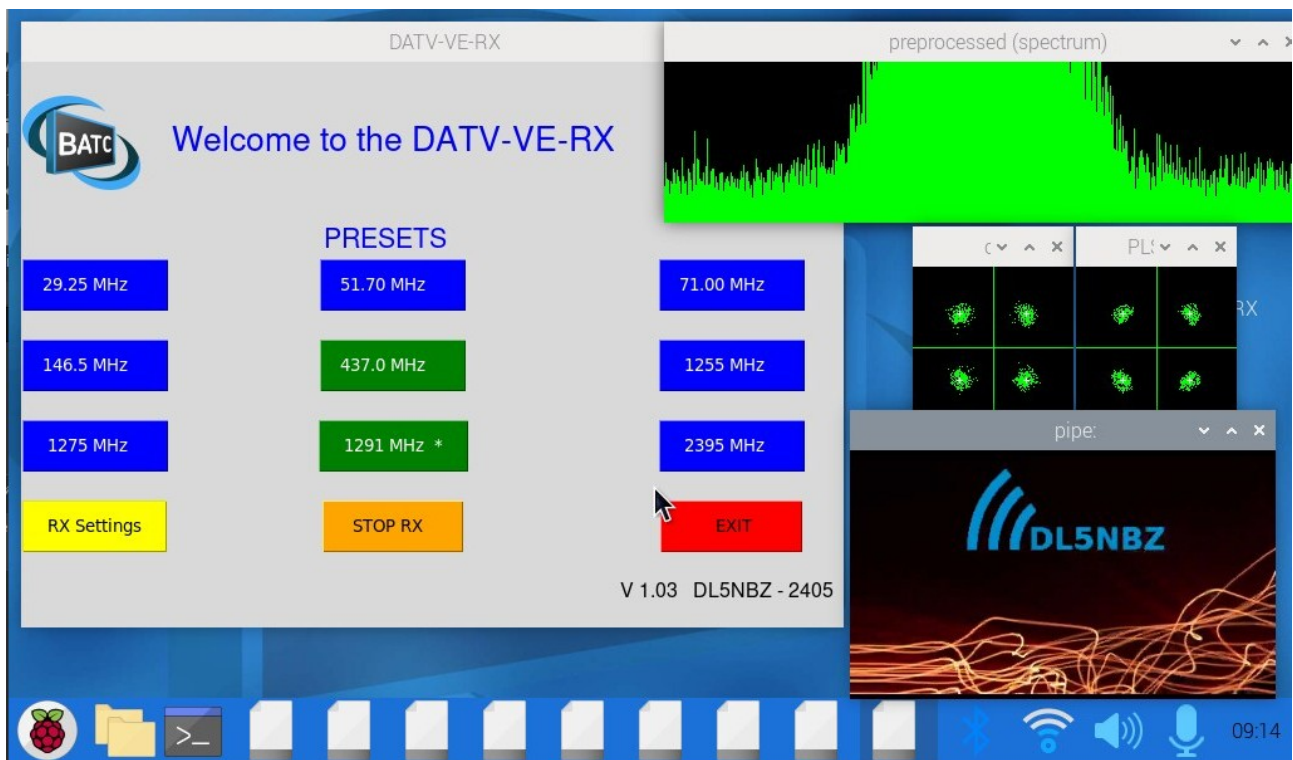




DATV-VE-RX



DL5NBZ - 2405 - V1.03

Inhaltsverzeichnis

Vorwort.....	3
Konzept.....	4
Aufbau und Baugruppen.....	5
Funktionsbeschreibung.....	6
Technische Daten.....	7
Hinweise zur Software.....	8
Bestandteile des System und Lizenzen.....	8
Beschreibung der Software.....	9
1. datv-ve-rx.py.....	9
2. settings.py.....	11
SDR-Parameter.....	11
RX-Parameter.....	12
RX-OPTIONS.....	12
Edit Preset-No.....	13
3. presets.csv.....	14
Noch zu erledigen:.....	15
VNC-Server.....	16
Installation der Software auf dem Raspi 5.....	16
Entwicklungsfortschritt.....	17
QO-100 Empfang.....	18
Wie geht es weiter?.....	19
.....	19
Appendix.....	20
Unterstützte RTL-SDR Hardware.....	20

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

Nach drei Jahren Erfahrung mit dem digitalen ATV bin ich zu der Erkenntnis gekommen, dass DATV zu senden wesentlich einfacher ist als DATV zu empfangen.

Meine ersten Empfangsversuche habe ich mit dem Portsdown2020 gemacht, mit dem man mit einem RTL-SDR-Stick DVB-S empfangen konnte.

Leider wurde das Konzept von der BATC nicht weiter verfolgt. Die BATC hat auf den MiniTiouner gesetzt und diesen in ihren Projekten eingesetzt. Die Empfänger mit dem Minitiouner sind sehr gute Geräte, die keine Wünsche offen lassen.

Im Laufe der letzten Jahre hat sich die Welt etwas weiter gedreht und durch QO-100 fand ein Wechsel von DVB-S zu DVB-S2 im digitalen Amateurfunk-Fernsehen statt. Damit war auch der Empfänger im Portsdown 2020 mit dem RTL-SDR-Stick hinfällig.

Um mehr Funkamateure von DATV zu begeistern war ein einfacher Einstieg in die Empfangstechnik nötig. Es musste also ein neues Empfangskonzept her, dass auch preislich deutlich unter dem Preis des Minitiouner liegt. Da die Verfügbarkeit des Minitiouner (Serrit Tuner) dem Ende entgegen geht ist ein weiterer Anreiz für ein einfaches Konzept gegeben.

Auf der Suche nach einer Lösung habe ich mich auf den harten und steinigen Weg der SDR-Technik zu einem einfachen DATV-Empfänger gemacht.

Nach einigen Monaten und mehreren Fehlschlägen, mehreren Gedanken an Kapitulation gibt es jetzt ein funktionierendes Gerät. Es ist sicher nicht der beste DATV-Empfänger der Welt, aber kostengünstig, ohne Abgleich und Löten aufzubauen.

Mein Arbeitsname für das Projekt ist „Volksempfänger“, weil das Gerät das Potenzial hat, ein neues Medium der breiteren Masse zugänglich zu machen. Daher auch nun die offizielle Bezeichnung DATV-VE-RX.

Der Empfänger ist ein reines Software- und SDR-Projekt und wird deshalb wahrscheinlich nie ganz fertig. Die Version V0.75 des Projekts ist aber inzwischen einsatzfähig (Mai 2024), nur die Benutzeroberfläche ist noch stark verbesserungsfähig.

Jede gute Lösung ist einfach

Im Weiteren wird der **DATV-VE-RX** im Detail beschrieben.

Konzept

In Deutschland ist das niedrigste, nutzbare Frequenzband für DATV 435 MHz (70cm), als weiteres Frequenzband bietet sich 1275 MHz (23cm) an. Höhere Frequenzen erfordern spezielle Antennen und waren meiner Meinung nach für das Projekt nicht im Fokus.

Diskrete Empfänger für 70cm und 23cm erfordern besondere Leiterplatten und SMD-Bauteile, die ein gewisses Können beim Lötten 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 einem Fernseher oder Monitor mit HDMI-Eingang erfolgen. Ein Display am Raspi über die DSI-Schnittstelle ist auch möglich.

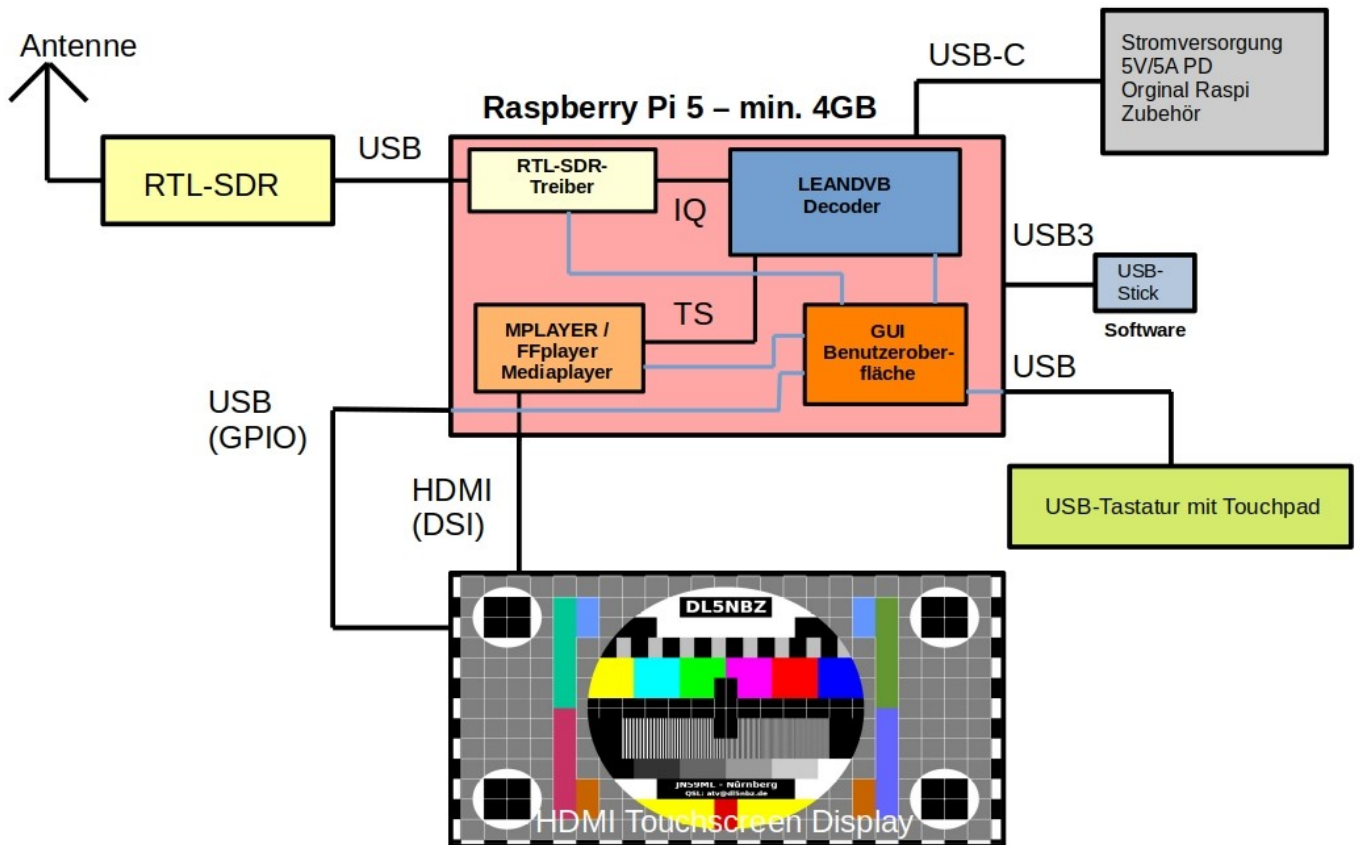
Folgende Komponenten sind nötig:

- 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 derzeit noch nicht richtig unterstützt. Das Original hat ein Metallgehäuse und schützt besser vor Einstrahlungen des Rechners.
- ein **Raspberry Pi5** (min 4GB) - da das Konzept mehrere Jahre Bestand haben sollte, habe ich mich für das aktuelle Modell mit dem aktuellen Betriebssystem „Debian Bookworm“ (Debian 12) entschieden.
- ein **Display** - min. 1024x600 Pixel, gerne auch als TouchScreen, alternativ ein Fernseher oder Monitor mit HDMI-Eingang, dann aber mit Maus und Tastatur.
- Eine geeignete **Stromversorgung** für den Raspi5 (**Achtung:** 5V/27W mit Power-Management)
- Ausreichende **Kühlung** des Raspi, 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

Der DATV-VE-RX wird aus fertigen Baugruppen zusammen gesteckt. Für die Verbindung der einzelnen Baugruppen werden handelsübliche Kabel verwendet.

Die USB-Tastatur mit Touchpad ist aktuell noch nötig. Die Bedienung soll in der endgültigen Version ausschließlich über den Touch-Screen möglich sein. Bei Verwendung eines Fernsehers oder Monitors am HDMI-Anschluß kommt man



DATV-VE-RX – Baugruppen und Verschaltung

um die USB-Tastatur mit Touchpad oder Maus nicht herum. Den Zusammenbau und 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 **mplayer bzw. ffplay** zur Umwandlung des Transportstreams in einen MPG2, MPG4, 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.

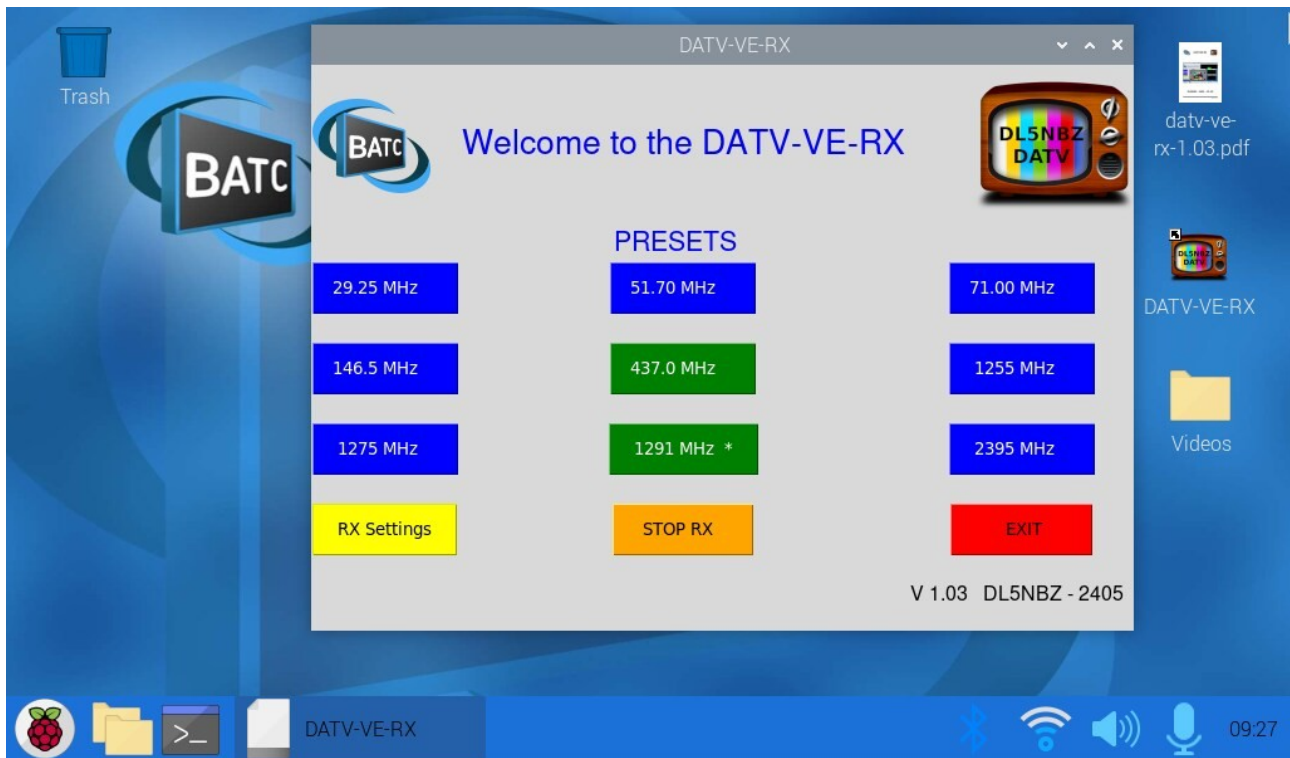
rtl_sdr bekommt die Abtastrate, die Verstärkung und die Empfangsfrequenz übergeben und wird gestartet, danach steht ein IQ-Signal bereit.

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 mplayer)** 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 sind durch grüne Buttons gekennzeichnet.

Über den gelben Button „RX-Settings“ wird ein weiteres Fenster geöffnet, in dem man 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: 18kS/s – 2000kS/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, dass 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 hoffe dadurch eine einfachere Inbetriebnahme des Empfängers. Die Zeit die so am Support einsparen kann, kann ich sinnvoller in die Weiterentwicklung stecken.

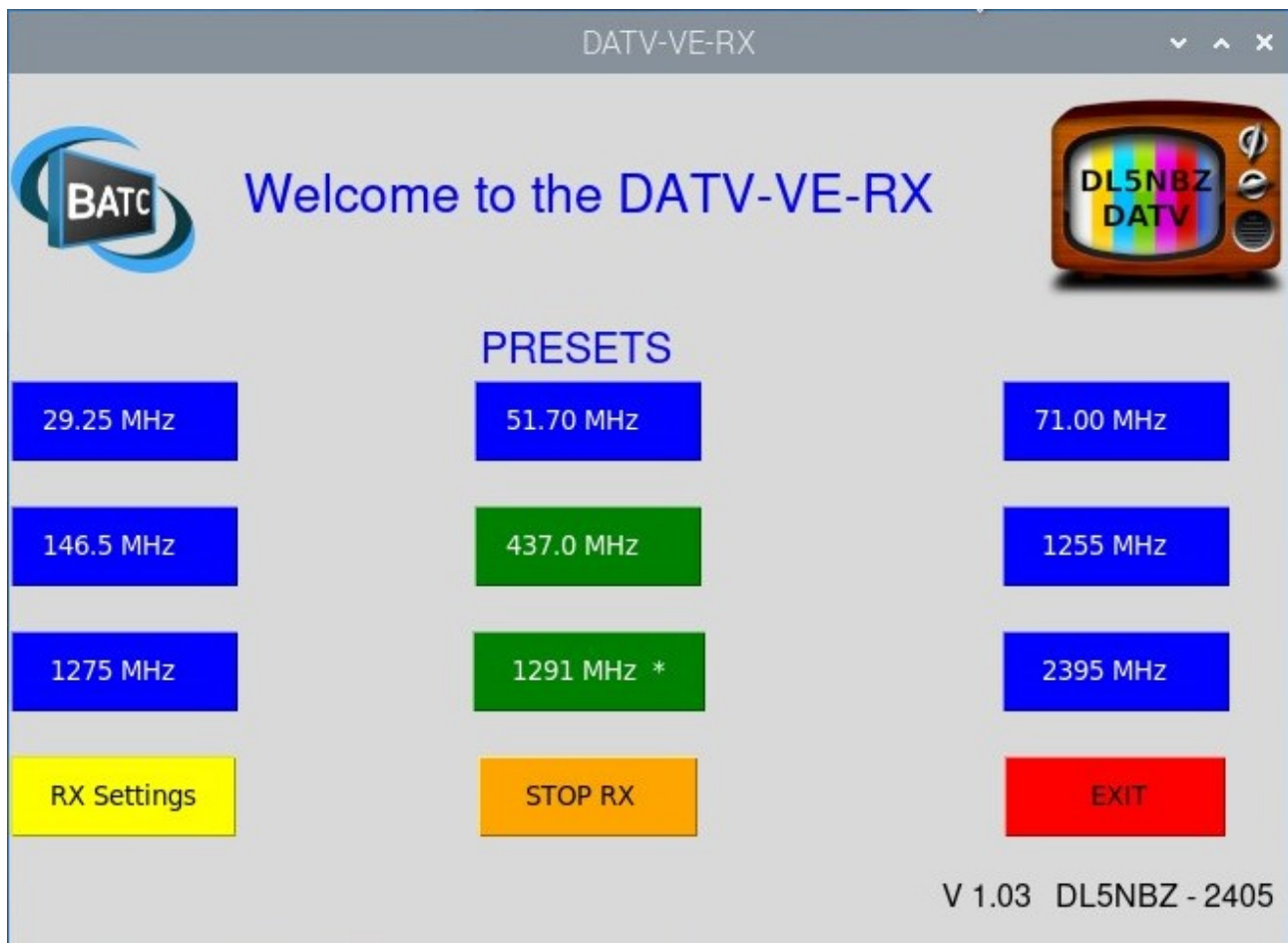
Grundsätzlich ist es auch möglich den Empfänger auch auf einem Laptop zu installieren. Voraussetzung ist auch hier ein Betriebssystem auf der Basis von Debian 12 Bookworm (i.e. Ubuntu, Linux Mint, usw). Es ist nicht auszuschließen, dass ich irgendwann dafür ein Installationsskript erstelle. Dieses Projekt 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

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:
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.

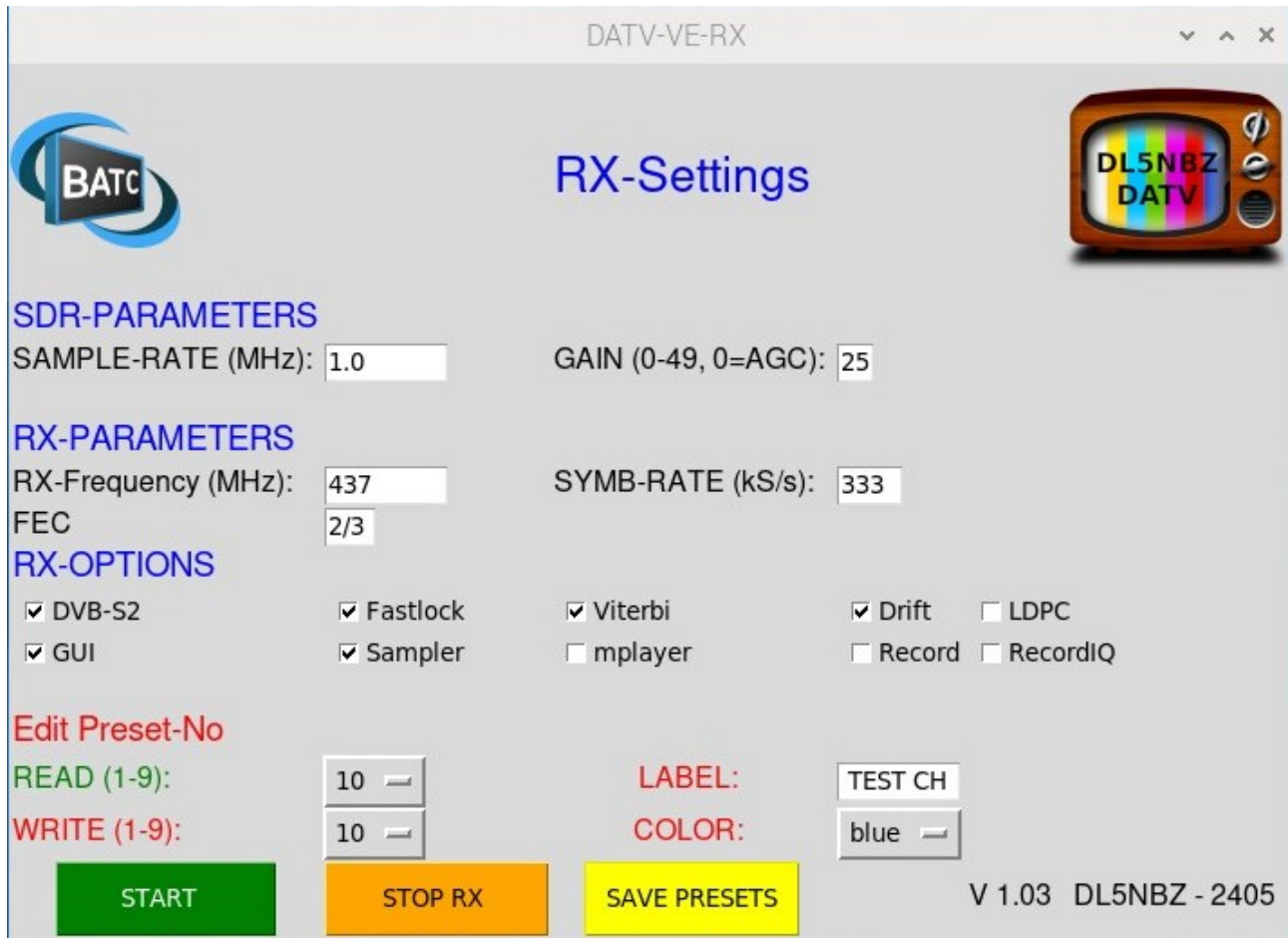
Die Presets kann man auch mit einem Texteditor direkt in der Datei **presets.csv** bearbeiten. Das setzt aber genaue Kenntnis der Syntax der Programme **rtl-sdr**, **leandvb** und **ffplay** voraus. Empfohlen wird die Bearbeitung mit dem Programm **settings.py**.

Dieser Programm-Teil ist mit einem Touch-Screen bedienbar.



Schaubild 1: Verschiedene RTL-SDR-Sticks

2. settings.py



Das Bild zeigt die Version 1.03 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 werden die Sample-Rate und die Verstärkung des Empfangsteils eingestellt.

Sample-Rate: Für die Sample-Rate empfehlen sich 2.4MHz, 1.8MHz oder 1.0MHz. Auf den niedrigen Bändern (29MHz, 51MHz und 71MHz) kann eine niedrigere Sample-Rate hilfreich sein, da die Sample-Rate die Eingangsbandbreite des Empfängers bestimmt. Mit einer niedrigeren Sample-Rate lassen sich „Störer“ links und rechts des Empfangskanals ausblenden. Die Sample-Rate wird auch an den Decoder (leandvb) weitergegeben.

GAIN: Die Grundeinstellung für GAIN ist 0. Mit dieser Einstellung übernimmt die automatische Verstärkungsregelung (AGC). Sollte das Grundrauschen (Noise Floor) zu stark sein, kann man die Verstärkung zu kleineren Werten ändern (1-49). Damit sollte das Rauschen geringer werden und den

Empfangskanal nicht mehr so stark stören. Die beiden Spektrum-Anzeigen nach Start des Empfänger 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 kommen eigentlich nur ganze Zahlen zur Anwendung. Gängige Werte sind 125kS/s, 250kS/s, 333kS/s, 500kS/s und 1000kS/s.

Die **Fehlerkorrektur (FEC)** wird als Bruch eingegeben, also z.B 8/9. Der Zähler gibt die Anzahl der Datenpakete an, der Nenner die Gesamtpaketzahl. Also in dem Beispiel sind von 9 Paketen 8 Datenpakete und ein Paket mit Korrekturdaten.

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 1.03 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 tickt 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 V1.03 noch ohne Funktion.

GUI: Nach dem Starten des Empfänger 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. Ab der Version 1.02 sind diese Anzeigen auf kleine Bildschirme optimiert.

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

MPLAYER: 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 MPLAYER. Durch ticken dieser Box wird mit dem MPLAYER angezeigt und nicht mit FFPLAY.

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.**

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, das Programm **settings.py** beendet und das Programm **datv-ve-rx.py** gestartet.

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 eingegebenen Parametern. Mit **STOP RX** beendet man den Empfang und kehrt in das Programm **datv-ve-rx.py** zurück. 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 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.

3. 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 die Überschriften der Spalten.

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 mit Default-Daten vom Programm angelegt:

No	Label	Color	OS Command
1	29.25 MHz	blue	rtl_sdr -f 29.25e6 -g 0 -s 6e4 - ./leandvb --gui -f 6e4 --sr 33e3 --cr 1/2 --standard DVB-S2 --sampler rrc --rrc-rej 30 --fastlock --viterbi ffmpeg -x 512 -y 300 -
2	51.70 MHz	blue	rtl_sdr -f 51.7e6 -g 25 -s 8e5 - ./leandvb --gui -f 8e5 --sr 125e3 --cr 2/3 --standard DVB-S2 --sampler rrc --rrc-rej 30 --fastlock --viterbi ffmpeg -x 512 -y 300 -
3	71.00 MHz	blue	rtl_sdr -f 71e6 -g 25 -s 8e5 - ./leandvb --gui -f 8e5 --sr 125e3 --cr 2/3 --standard DVB-S2 --sampler rrc --rrc-rej 30 --fastlock --viterbi ffmpeg -x 512 -y 300 -
4	146.5 MHz	blue	rtl_sdr -f 146.5e6 -g 25 -s 1.8e6 - ./leandvb --gui -f 1.8e6 --sr 125e3 --cr 2/3 --standard DVB-S2 --sampler rrc --rrc-rej 30 --fastlock --viterbi ffmpeg -x 512 -y 300 -
5	437.0 MHz	green	rtl_sdr -f 437e6 -g 25 -s 1.8e6 - ./leandvb --gui -f 1.8e6 --sr 333e3 --cr 2/3 --standard DVB-S2 --sampler rrc --rrc-rej 30 --fastlock --viterbi ffmpeg -x 512 -y 300 -
6	1255 MHz	blue	rtl_sdr -f 1255e6 -g 25 -s 1.8e6 - ./leandvb --gui -f 1.8e6 --sr 333e3 --cr 2/3 --standard DVB-S2 --sampler rrc --rrc-rej 30 --fastlock --viterbi ffmpeg -x 512 -y 300 -
7	1275 MHz	blue	rtl_sdr -f 1275e6 -g 25 -s 1.8e6 - ./leandvb --gui -f 1.8e6 --sr 333e3 --cr 2/3 --standard DVB-S2 --sampler rrc --rrc-rej 30 --fastlock --viterbi ffmpeg -x 512 -y 300 -
8	1291 MHz *	green	rtl_sdr -f 1291e6 -g 25 -s 1.8e6 - ./leandvb --gui -f 1.8e6 --sr 333e3 --cr 2/3 --standard DVB-S2 --sampler rrc --rrc-rej 30 --fastlock ffmpeg -x 512 -y 300 -
9	2395 MHz	blue	rtl_sdr -f 1255e6 -g 25 -s 2.4e6 - ./leandvb --gui -f 2.4e6 --sr 333e3 --cr 2/3 --standard DVB-S2 --sampler rrc --rrc-rej 30 --fastlock --viterbi ffmpeg -x 512 -y 300 -
10	TEST CH	black	rtl_sdr -f 437e6 -g 0 -s 2.4e6 - ./leandvb --gui -f 2.4e6 --sr 333e3 --cr 8/9 --standard DVB-S2 --sampler rrc --rrc-rej 30 --fastlock --viterbi ffmpeg -x 512 -y 300 -

Noch zu erledigen:

- geplante Funktionen testen und freischalten
- ~~eingestellte Kanäle in die neun presets schreiben, um die manuelle, fehlerträchtige Bearbeitung der presets.csv unnötig zu machen.~~
- Benutzeroberfläche verbessern
- ~~eine weniger rechenintensive Signalanzeige einbauen~~
- ~~eingeebene Daten sollen beim nächsten Start von settings.py wieder in den Eingabe-Feldern stehen~~
- Fehlerkorrekturen
- Prüfung der eingegebenen Daten auf Plausibilität
- Ungültige Kombinationen von Einstellungen und Optionen erkennen und automatisch korrigieren

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 beliebige 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: **raspberrry**

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.

Installation der Software auf dem Raspi 5

1. Das vollständige Image **datv-rx-103.img** (11,4GB) ist herunterzuladen und mit einem Image-Writer auf eine MicroSD-Karte oder einen (USB3) Stick zu schreiben. Das Medium sollte 32GB groß sein. Ich empfehle das Programm **Etcher** (etcher.io). Etcher ist für Windows, MacOS und Linux verfügbar. 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 Datei-System 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. Dann wird das RaspianOS mit auch mit Updates versorgt.

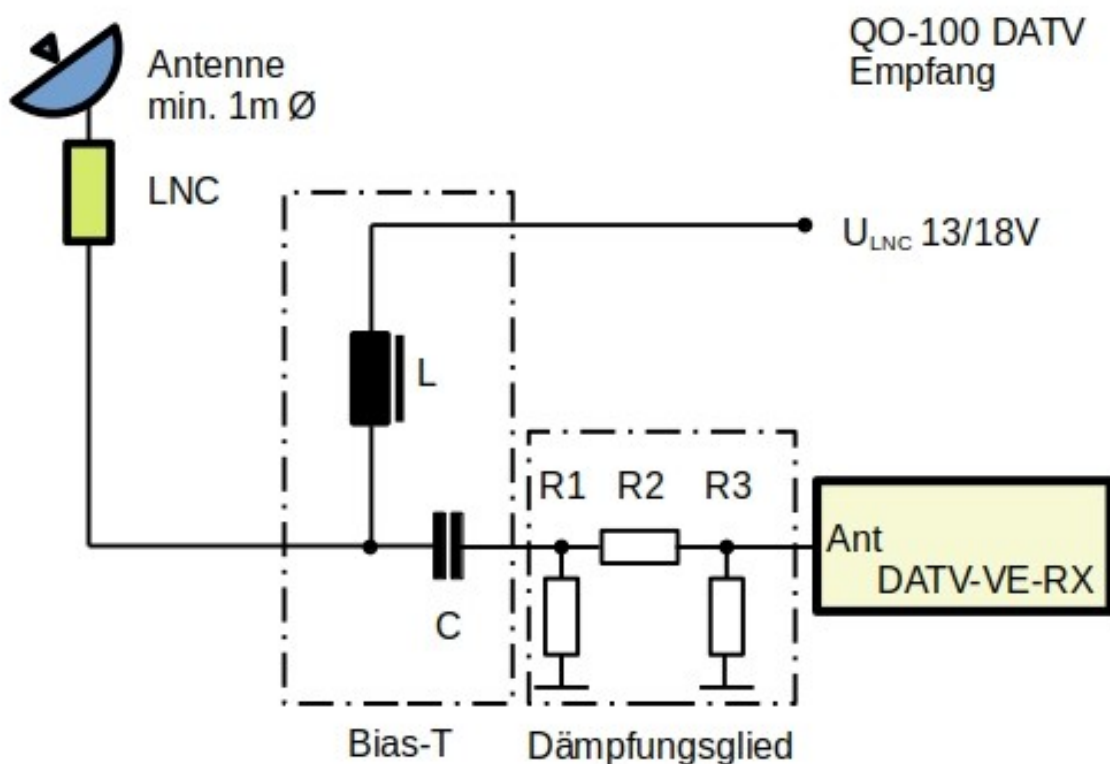
5. Sollte man zur Eingabe eines Benutzernamens und eines Passworts aufgefordert werden, gelten die Default-Einstellungen von RaspianOS:
Benutzername: **pi** Passwort: **raspberrry**

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 vollem, 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 vollem, 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.

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 zu sehen ist. 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 1m Durchmesser nötig. Für den Empfang der Bake mit 1500 kS/s muss eine Samplerate von 2.0 oder 2.4 MHz eingestellt sein.

Wie geht es weiter?

Der DATV-VE-RX ist ein Software-Projekt und wird deshalb wahrscheinlich nie ganz fertig. Die vorliegende Version 1.03 ist hinreichend stabil und bietet einen einfachen Einstieg ins terrestrische DATV, um die lokalen DATV-Stationen auf 70cm und 23cm zu beobachten. Bei guten Bedingungen ist es auch möglich britische Stationen auf 51,7MHz, 71MHz oder sogar auf 146,5MHz zu sehen. Es wird sicher in absehbarer Zeit eine neuere Version geben, die auch den QO-100 Empfang komfortabler machen wird.

Es werden in die neue Version dann auch die Erfahrungen meiner „Beta-Tester“ einfließen.

Aus heutiger Sicht wird das aber nicht vor Anfang 2025 sein.

Appendix

Unterstützte RTL-SDR Hardware

Außer den bereits erwähnten original 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