Aufbau eines FM/analog-Repeaters auf Basis SVXLINK mit einem YAESU DR1XE

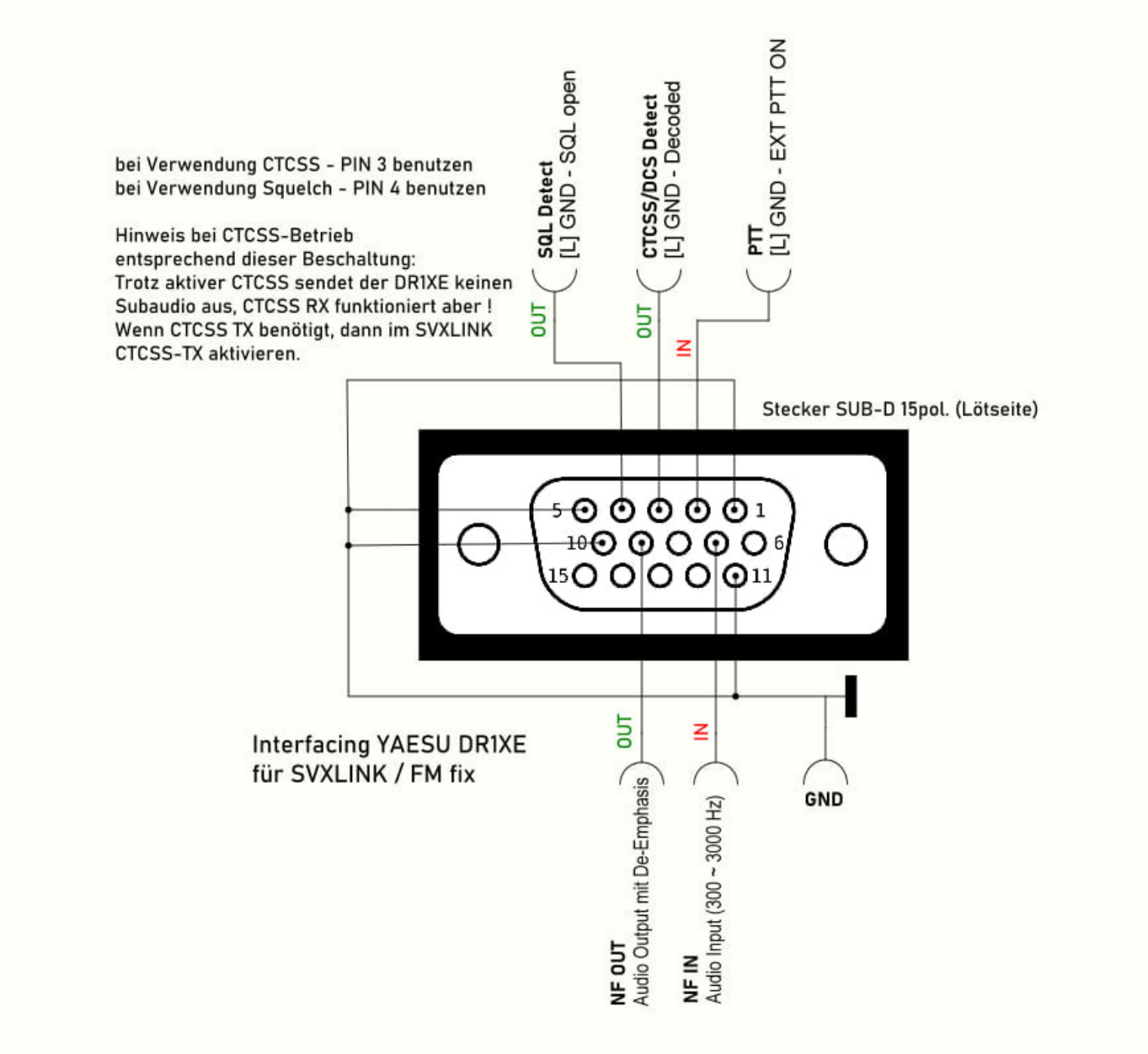
Einleitung

Bei dem YAESU DR1XE handelt es sich um ein von YAESU kommerziell gefertiges Relaissystem, was einerseits FM/analog und ebenfalls C4FM/digital beherrscht. Hier soll es nur um den Einsatz dieses Gerätes für FM/analog auf Basis der Open-Source-Software von Tobias/SM0SVX namens SVXLINK gehen, einer kompletten Steuerungslösung für FM/analog-Relais auf Softwarebasis, die ebenfalls zuätzliche Module wie ECHOLINK, ECHO-Funktion usw. enthält.
Inzwischen hat YAESU den DR1XE durch den DR2XE abgelöst, wir wollen uns hier aber ausschließlich mit dem Vorgänger DR1XE beschäftigen.
Ich selber habe als Sysop der Relaisfunkstellen DB0SPB, DB0OLL und DB0GRZ drei dieser DR1XE aktiv im Einsatz als öffentliche Relaisfunkstellen, kenne also diese Geräte sehr gut einschließlich ihrer „Tücken“.

Anschluß eines DR1XE mittels der 15pol. REMOTE-Buchse

Grundlagen einer Anschaltung von externen Zusatzgeräten am DR1XE entnehmen wir dem Handbuch(engl.) des DR1X, und dort speziell den Seiten 17/18 „Connection to an external controller“.

An der Rückfront des DR1XE befinden sich mehrere Anschlußbuchsen. Für unseren Anwendungsfall, FM/analog-Repeater mit SVXLINK, verwenden wir die 3-reihige, 15polige SUB-D-Anschlußbuchse. Das notwendige Anschlußkabel benötigt einen 15poligen, 3-reihigen SUB-D-Stecker, der bei den Elektronikhändlern problemlos beschaffbar sein sollte. Hier das notwendige Anschlußschema für diesen Stecker:



Hierbei folgende Erläuterungen:

- Wir benötigen die Betriebsart REMOTE und nicht, wie man vielleicht vermuten würde, die Betriebsart REPEATER. Die Betriebsart REPEATER ist dafür gedacht, wenn dieser Repeater stand-alone arbeitet, also ohne externe Interface-Anschaltungen. Dabei bleibt der RX und TX intern gekoppelt, der Audio wird also intern direkt zwischen RX und TX zusammengeschalten. Das wollen wir aber bei Verwendung einer externen Steuerung wie SVXLINK nicht. Wir möchten den RX und TX getrennt steuern, also auch dessen Audio getrennt verarbeiten, deswegen benötigen wir die Betriebsart REMOTE. Dazu legt man den PIN 1 auf Masse/GND.
- Weiterhin wollen wir den DR1X fix auf die Betriebsart FM stellen. Dazu muss ebenfalls PIN 11 auf Masse/GND gelegt werden, der PIN 12 muss auf H bleiben, was beim DR1X bedeutet, dieser wird nicht angeschlossen und bleibt frei. Die PIN 5 und 10 sind beide Masse/GND, wir benötigen also eine Brücke zwischen den PIN 1,5,10 und 11. Damit haben wir den Repeater im REMOTE-Mode und fix auf FM eingestellt.
- Zur Squelch-Auswertung haben wir zwei Optionen: Einen Ausgang am PIN 4, der die normale, eingebaute Squelch signalisiert, wenn diese durch ein Signal öffnet. Möchten wir stattdessen aber CTCSS oder DCS verwenden, liegt am PIN 3 eine Signalisierung vor, wenn der DR1XE eine gültige CTCSS- oder DCS-Decodierung an seinem RX erkennt. Beide Signale sind Low-Active

durch einen Open-Collector, schalten also gegen Masse/GND wenn Erkennung vorliegt. Es macht natürlich nur Sinn, eine der beiden Optionen zu verwenden, je nachdem, ob man einfach die normale Squelch verwenden möchte oder den CTCSS-Decoder des RX nutzen möchte. Wenn man CTCSS machen will/muss, empfehle ich grundsätzlich das dem RX des DR1X zu überlassen und nicht - wie mit SVXLINK ebenfalls möglich - das per Software auszuwerten. Die Zuverlässigkeit der CTCSS-Erkennung des DR1X ist um Längen besser als die Softwareerkennung im SVXLINK.
- die PTT des TX wird am PIN 2 geschalten, Sender EIN bedeutet, PIN 2 muss gegen Masse/GND geschalten werden.

Grundeinstellungen des DR1X am Touchdisplay

Folgende Einstellungen sollten für FM-Betrieb gemacht werden:

- AUDIO-Mode sollte für 1k2-Packetspeed bei FM-Betrieb gewählt werden, also kein Diskriminator-Audio: auf SETUP tippen, dann F tippen und solange auf die oben angezeigte Frequenz tippen, bis im Display Packetspeed 1200bps angezeigt wird
- DEVIATION sollte auf WIDE stehen
- bei MODE/REMOTE muss REMOTE auf ON stehen, die beiden anderen Optionen auf OFF
- bei ID/SET muss das Repeatercall eingetragen werden, das ist zwar eigentlich nur bei C4FM/digital erforderlich, ABER der DR1X arbeitet sonst auch in FM nicht richtig !
- bei TIMER die Option TOT auf OFF setzen, Sendezeitbegrenzung erledigen wir später direkt im SVXLINK. SQL TAIL LENGTH bitte auf 0 setzen, sonst ziehen wir eine Rauschschleppe mit, wenn die SQL oder das CTCSS schließt
- bei ID/ANNOUNCE stellen wir INTERVAL auf OFF, denn die Relaiskennung erledigt ebenfalls das SVXLINK für uns

Die Sendeleistung des DR1X NIEMALS höher als 20W (also MID) wählen, auch wenn der DR1X 50W kann, überlebt das die Endstufe nicht lange. Grund ist die unzureichende Kühlung des DR1X, die YAESU wirklich schlecht umgesetzt hat. Auch nicht „zum mal ausprobieren“ !!!

Besonderheiten im Mode REMOTE bei CTCSS-Betrieb (ebenfalls Hardware-SQL-Betrieb)

Im Mode REMOTE sendet der DR1X - auch wenn es am Gerät so eingestellt wird - KEINEN CTCSS-Subaudioton aus! Die CTCSS-Erkennung am RX funktioniert aber korrekt. Wenn man CTCSS-Subaudio aussenden will, muss man das im SVXLINK aktivieren und damit dem SVXLINK die Generierung des CTCSS-Subaudio übergeben. Dazu ist in der svxlink.conf folgendes zu machen:

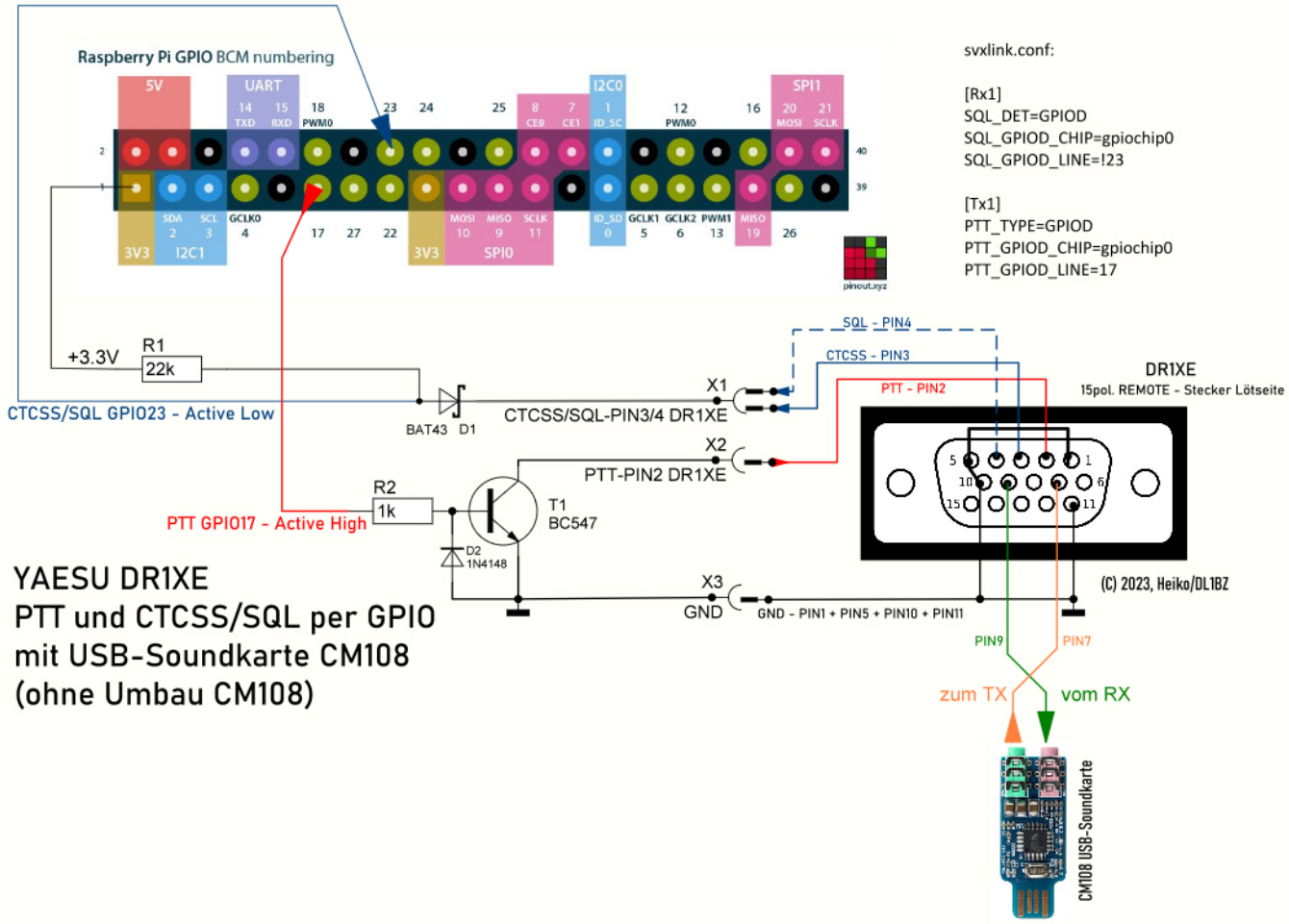
```
[RepeaterLogic]
# CTCSS 123Hz oder anpassen
REPORT_CTCSS=123.0
TX_CTCSS=ALWAYS

[Tx1]
# CTCSS 123Hz oder anpassen
```

```
CTCSS_FQ=123.0
CTCSS_LEVEL=9
```

Die CTCSS-Erkennung am Rx kann ganz normal unter SIGNALING mit der korrekten Frequenz eingestellt werden und unter SQL die Option RX SQL auf TONE stellen. TX SQL kann auf OFF bleiben, weil er es ja sowieso nicht sendet, wie ich das eben beschrieben habe. Wenn am RX CTCSS erkannt wird, meldet der PIN 3 das Schaltsignal korrekt.

Weiterhin muss man wissen, das sowohl am PIN 3 (CTCSS-Signal) als auch am PIN 4 (SQL-Signal) KEINE Spannung ausgegeben wird. Das ist ein reiner Open-Collector-Ausgang und fungiert nur als Schalter. Da dieser Ausgang Low-Active ist, also gegen Masse/GND schaltet, muss man ggf. am Auswerteinterface einen Pullup-Widerstand gegen max. +3.3V schalten, keinesfalls aber +5V ! Der max. Strom darf 10mA nicht überschreiten. Ich habe das immer mit einem R von 22kOhm als Pullup gemacht, dann liegen ca. +2.7V an, was zur Auswertung ausreicht, um sicher H- oder L-Level zu erkennen. Hier ein Beispiel an einem Raspberry Pi, wo der GPIO23 des Pi für die SQL-Auswertung verwendet wird (die Diode dient als zusätzlicher Schutz und ist zu empfehlen):



Damit der DR1X bei erkanntem CTCSS-Decoder-Signal sofort startet, müssen wir in der `svxlink.conf` folgendes einstellen:

```
[RepeaterLogic]
OPEN_SQL_FLANK=OPEN
```

Diese Schaltung lässt sich auch auf die normale Hardware-SQL-Signalisierung anwenden, die ebenfalls Low-Active arbeitet. In diesem Fall ist allerdings der PIN 4 der 15pol. REMOTE-Buchse zu verwenden anstatt der PIN 3 für CTCSS. Beide Signalleitungen - SQL und CTCSS - arbeiten unabhängig voneinander.

Der RX des DR1X liefert immer ein Signal vom RX - und zwar unabhängig davon, ob die SQL geschlossen ist oder CTCSS erkannt wird. Auch bei CTCSS-Betrieb muss man die eingebaute normale SQL so einstellen, das sie bei keinem Signal schließt, sonst *kann* es passieren, das der TX nicht angeht.

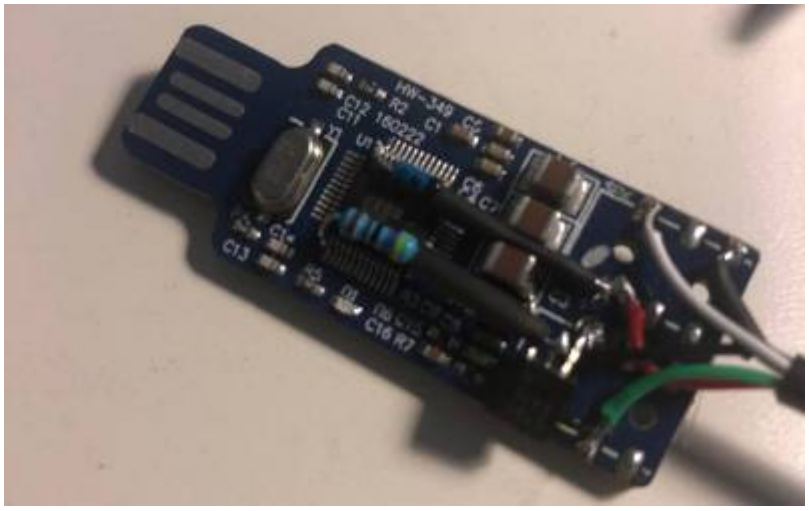
Soundkarten und der DR1XE - eine nicht ganz so einfache Sache

Bei Einsatz vom SVXLINK benötigen wir ja eine Soundkarte, um den RX-Audio und den TX-Audio verarbeiten zu können.
Leider beginnnen hier einige Probleme.
Die Fakten: Nicht jede Soundkarte schafft die notwendigen Audio-Pegel, die für den DR1XE erforderlich sind. Dazu gehört - leider - auch das inzwischen sehr verbreitete ELENATA-Board vom Andy/DK1LO. Die ELENATA verwendet den LINE-IN, schafft es aber bei selbst voll auf 100% aufgeregeltem Mixer nur auf 40% Input-Level auszusteuern. Das ist leider zu wenig, der Audio vom RX wäre zu leise.
Nun kann man zwar in der `svxlink.conf` nachhelfen:

```
[Rx1]
# wenn ELENATA Board verwendet wird sonst auf 0 lassen
PREAMP=10
```

ABER: Das hebt dummerweise auch den Audiostream in gleichem Maße mit an, der via Netzwerk vom SVXReflektor kommt. Als Notlösung ok, aber nicht unbedingt zu empfehlen. Auch die SQL-/CTCSS-Signalisierung funktioniert im Zusammenhang mit der ELENATA derzeit nicht bzw. erst nach einer Modifikation. Andy hatte das für MOTOROLAs gemacht, die weniger Audiopegel brauchen und deren SQL-Signale H-active sind, also mit +5V SQL EIN signalisieren, während der DR1X wie beschrieben ja Low-Active ist, wenn SQL EIN signalisiert wird. Es soll aber bald eine Überarbeitung der ELENATA geben, wo die „Eigenheiten“ eines DR1X berücksichtigt werden sollen.

Die Alternative wäre eine ziemlich verbreitete USB-Soundkarte auf Basis eines CM108-Soundchips. Diese lässt sich sogar umbauen, so dass PTT und SQL durch die Soundkarte selber ausgewertet und gesteuert werden können, denn dieser Soundchip verfügt über eigene GPIOs, die mittels HID_RAW_DEVICEs angesteuert werden können, was SVXLINK zum Glück unterstützt. Auch schafft dieser Soundchip die erforderlichen Pegel für den DR1X, so dass auf den Parameter `PREAMP` verzichtet werden kann bzw. der auf 0 gesetzt werden kann `PREAMP=0` .



Der Umbau ist nicht weiter schwierig, ein bisschen Lötarbeiten, ein paar Bauelemente (R1 4,7kOhm, D1 BAT43, T1 BC547C), schon hat man nach ca. 1h Bastelarbeit ein USB-Soundinterface mit SQL-Auswertung, PTT-Steuerung und Audio-IN/Audio-OUT.

Ich habe speziell zum Thema DR1XE und CM108-Soundkarten einen Wiki-Beitrag erstellt, wo ich genauer auf diese Option CM108-USB-Soundadapter am DR1X eingehe.

Die schon oft gehörte Meinung, das USB-Soundkarten mit SVXLINK Probleme machen, gehört definitiv ins Reich der Märchen und Sagen. Sie funktionieren ufb, auch im Dauereinsatz am Relais. Da hatte ich wesentlich mehr Stress bei HAT-Soundkarten, also die, die auf den Pi gesteckt werden. Meist muss man da erst irgendwelche Kernelmodule compilieren, die Konfiguration des Raspian anpassen und andere Zusatzarbeiten ausführen. Bei den meisten USB-Soundkarten ist das nur „Dranstecken“ und fertig. Manchmal sind die einfachen Lösungen eben doch die besseren und effektiveren.

Am PIN 7 des DR1X wird der Audio-Out der Soundkarte eingespeist, am PIN 9 kommt das RX-Audio raus (bereits gefiltert 300~3000Hz), was der Soundkarte an deren Input zugeführt werden muss. De-Emphasis und Pre-Emphasis sind in der `svxlink.conf` auf 0 zu setzen - weil wir ja den DR1X bereits auf Packet Speed 1200bps gestellt hatten und damit De-Emphasis und Pre-Emphasis bereits im DR1X umgesetzt werden:

```
[Rx1]
DEEMPHASIS=0

[Tx1]
PREEMPHASIS=0
```

Will man Audio auf Diskriminator-Ebene (entspricht Packet Speed 9600bps) wie bei den MMDVM-Modems nutzen, benötigt man mehr Audio-Pegel, was nur wenige Soundkarten schaffen.
Um zu prüfen, ob die Soundkarte genügend NF-IN (also Audio vom RX zur Soundkarte) liefert, können wir das überprüfen (zeigt ein VU-Level-Meter auf der Konsole an):

```
$ sudo systemctl stop svxlink
$ sudo arecord -D hw:1 -V mono -f S16_LE -c1 -r48000 /dev/null
```

Bei Soundkarten mit 2 Kanälen, also stereo, muss man folgenden Befehl verwenden:

```
$ sudo arecord -D hw:0 -V stereo -f S16_LE -c2 -r48000 /dev/null
```

Zu den Parametern:

- -D hw:1 | Nummer des ALSA-Audio-Devices, meist 0 oder 1, abhängig von der/den eingesetzten Soundkarte(n)
- -V mono | zeigt nur einen oder beide Kanäle an, wenn beide zu sehen sein sollen, dann -V stereo verwenden
- -c1 | Anzahl der Audio-Kanäle, bei Mono-Soundkarten -c1 verwenden, bei Stereo-Soundkarten -c2 verwenden


Liefert trotz voll aufgedrehten Capture-Device (ALSAMIXER Input = 100)*und*Rauschen vom RX (der RX des DR1X liefert IMMER ein Signal, auch bei geschlossener Hardware-SQL und/oder aktiviertem CTCSS) das VU-Level-Meter nur ca. 40% Aussteuerung, reicht die Verstärkung der Soundkarte für 9k6-Audio nicht aus, das Audio des RX wäre immer zu leise.In diesem Fall muss der DR1X auf 1200 Packetspeed gestellt werden und das 9k6-Audio kann nicht genutzt werden. Nur wenn man ca. 80-90% Aussteuerung erreichen kann, wäre 9k6-Audio an der Soundkarte nutzbar. Die „künstliche“ Verstärkung mittels PREAMP-Parameter (in db) in der [Rx1]-Sektion der svxlink.conf sollte nach Möglichkeit nur in Ausnahmefällen Anwendung finden.

Bisher konnte ich *nur* mit USB-Soundkarten auf Basis des CM108-Soundchips eine ausreichende Aussteuerung bei 9k6-Audio-Betrieb erreichen, alle anderen wie ELENATA etc. schafften das leider nicht.

DR2X vs. DR1X

Ohne jetzt auf alle Details bzw. die Unterschiede zwischen diesen beiden Versionen einzugehen, nur das Wichtigste:

- auch ein DR2X liesse sich für SVXLINK einsetzen
- was am DR2X nicht mehr geht, ist die Umschaltung Packet Speed 1200bps auf 9600bps. Man kommt also am DR2X nicht mehr an den Diskriminator-Ein-/Ausgang des Audios ran, was u.a. den Einsatz eines digitalen MMDVM-Modems nicht mehr möglich macht, also mal einen DR2X für DV-Multimode wie DMR, C4FM/YSF, D-STAR wie es der DR1X noch konnte, ist leider Geschichte. Da gibt es auch keinen „Work-around“. Ich sage, das hat YAESU bewusst unterbunden, denn 90% der eingesetzten DR1X liefen digital mit einem MMDVM-Modem und nicht mit dem YAESU-eigenen HRI200, was ja nur C4FM konnte und das auch nur an YAESUs eigenem WiresX-Netzwerk und auch noch einen WINDOWS-PC für deren komische Steuersoftware erforderlich machte, die es leider nur für WINDOWS gab.
- Für FM/analog ist diese Diskriminator-Geschichte aber nicht relevant, das ginge also auch mit einem DR2X

73 Heiko, DL1BZ
Sysop DB0SPB / DB0OLL / DB0GRZ

zurück zur Startseite