

DOK R17

Willich



im März 2021

R17 Remote Station

von DH5HS, Heiko Streeck

These

Mit dem kleinstcomputer Raspberry PI ist es möglich mit frei verfügbarer Software eine Remote Station zu betreiben.

Hintergrund

Ich besitze einen ICOM IC7300 und kann diesen mit der ICOM Software *RS-BA* oder der Software *Win4ICOMSuite* remote betreiben. Leider ist hierfür bei beiden Produkten ein Windows Rechner als Station Server nötig welcher im Shack bei Nichtbenutzung vor sich hinrauscht.

Eine Möglichkeit einen Raspberry als StationServer zu verwenden bietet m.W. aktuell nur MFJ mit dem RigPI Station Server MSJ-1234 als kommerzielle Lösung an.

Da ich einen Raspberry 4 bereits im Dauerbetrieb als Webserver, IOBroker, Datenbankserver, NEXTCloud Server und HAMclock im Shack betrieb bot es sich anVersuche zu unternehmen ob und wie der Raspberry als StationServer dienen kann.

Einleitung

Dieses Dokument soll meine Versuche, eingesetzte Software und Detaillösungen festhalten welche letztlich zu einem, das kann ich vorwegnehmen, positiven Ergebnis geführt haben welches als Ausgangslage für weitere Versuche dienen kann. Es ist keine Step-by-Step Anleitung und setzt Kenntnisse im Umgang mit Linux, Windows und dem Netzwerkrouter voraus.

Hauptstruktur

Meinen Versuch habe ich in 2 Bereiche unterteilt:

- a. Transceiversteuerung / CAT Schnittstelle
- b. Audio Übertragung RX/TX

Teil A - Transceiversteuerung

Die Transceiversteuerung erfolgt mittels der Software FLRIG auf dem entfernten Windows Notebook/PC über Netzwerk/Internet. Lokal am Transceiver (hier IC7300) ist lediglich der Raspberry angeschlossen. Es gilt also die serielle Kommunikation vom TRX an einem PC im Netzwerk/Internet zu leiten.

Überblick über die Konfiguration

1. TRX IC-7300 (oder jeder andere) per USB an dem
2. Raspberry 4 angeschlossen. Mit der
3. Software *SER2NET* wird die serielle Schnittstelle (COM Port) über das Netzwerk geleitet
4. Mit einem Windows Notebook/PC oder auch einem Linux System(nicht getestet) und der Software FLRIG wird der Transceiver ferngesteuert.

Details zur Konfiguration - USB Port ermitteln

Der IC7300 ist per USP mit dem Raspberry verbunden. Die Ermittlung des USB Ports im Raspberry erfolgt mit:

- `dmesg | grep tty`

In der Ausgabe ist der COM Port zu erkennen:

- ...
- `[12658.342468] usb 1-1.1.1: cp210x converter now attached to ttyUSB0`
- `[17123.662243] cp210x ttyUSB0: cp210x converter now disconnected from ttyUSB0`
- `[25795.531498] usb 1-1.1.1: cp210x converter now attached to ttyUSB0`

hier in diesem Fall **ttyUSB0**. Diesen Port notieren.

Raspberry 4 Konfiguration durchführen/ermitteln

Ich setze ein Raspberry PI OS voraus. Wir benötigen die IP Adresse – notieren. Die Software SER2NET mit

-
- `sudo apt-get install ser2net`

Die Konfigurationsdatei `/etc/ser2net.conf` editieren wie folgt (hier den notierten COM Port verwenden falls abweichend):

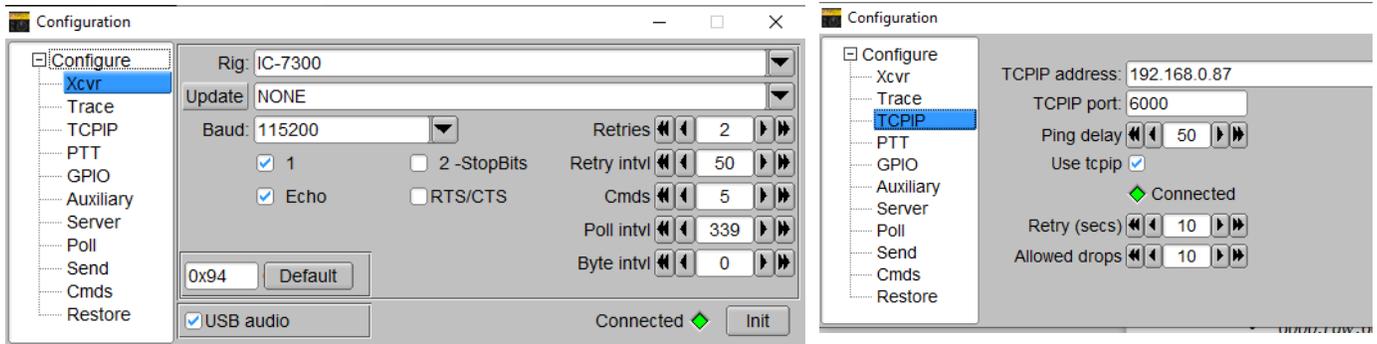
-
- `#ICOM IC7300`
- `6000:raw:600:/dev/ttyUSB0:115200 8DATABITS NONE 1STOPBIT`

Alle anderen Zeilen mit einem # am Anfang versehen.

Die Baudrate, hier 115200 Bits, muss entsprechend dem am Funkgerät eingestellter Baudrate eingestellt werden. Bei abweichenden Daten, Soppbits, Datenbits und Parity sind anzupassen. Das war es auf der Linux Seite.

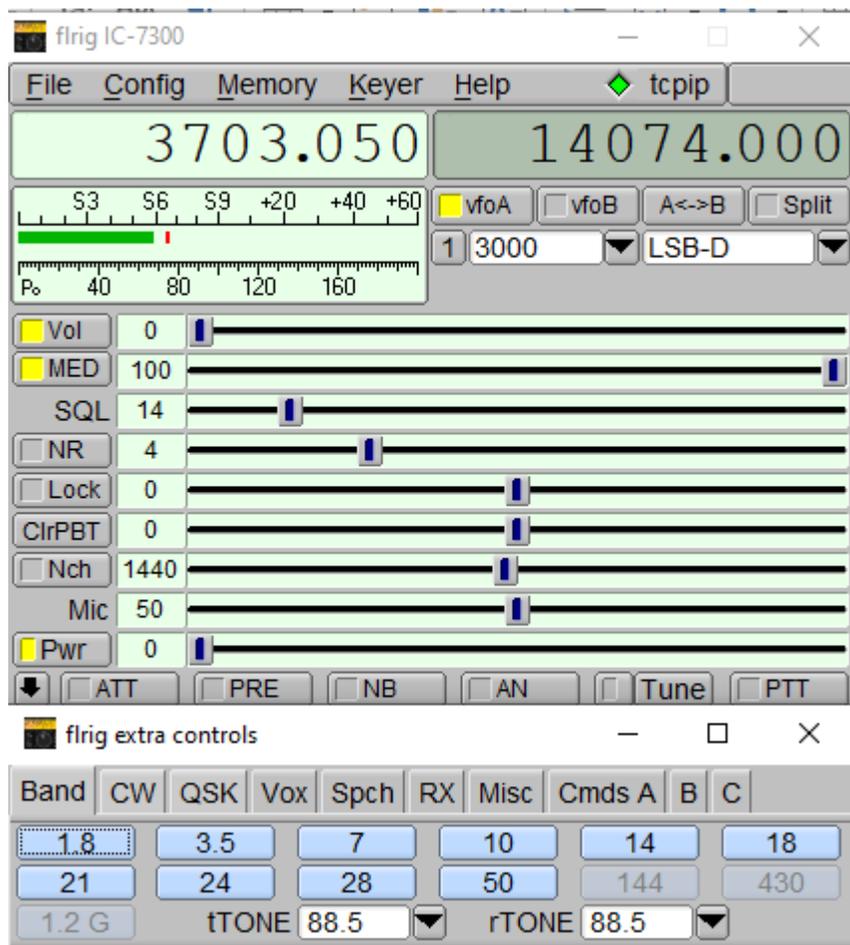
Windows Notebook/PC konfigurieren

-
- Download und Installation der Software FLRIG zB auf <http://www.w1hky.com/> (hier die notierte IP Adresse des Raspberries verwenden)



Das war es für den Windows Notebook/PC auch schon.

Es dürfte nun die remote Transceiversteuerung im lokalen Netz funktionieren. Für eine Steuerung über das Internet sind auf dem Router die entsprechenden Ports freizuschalten. *SER2NET* verwenden in der gezeigten Konfiguration den Port 6000.



Teil B – Audio

Um Audio auf dem Remote PC wiederzugeben ist eine Audioübermittlung über das Netzwerk/Internet nötig. Hier kommt es auf eine geringe Latenz an. Ich habe mich hier für die Software *MUMBLE* entschieden welche auch von Spielern, welche extrem viel Wert auf geringe Latenzen legen, genutzt wird. Mumble Clients gibt es für alle gängigen Betriebssysteme. Einen eigenen Mumble Server (Murmur) habe ich auf dem Raspberry eingerichtet. Auch MFJ verwendet für sein kommerzielles Produkt *RigPI Station Server* Mumble zur Audioübertragung.

1. Mumble Server auf Raspberry installieren

Die Installation des Mumble Servers starten wir wie folgt:

- `apt-get install mumble-server`

Ist die Installation abgeschlossen, können wir den Mumble Server konfigurieren

- `sudo dpkg-reconfigure mumble-server`

Innerhalb der Konfiguration können wir folgende Einstellungen vornehmen:

- Autostart: Mumble Server beim Booten Starten? Yes
- Der Mumble Server darf eine höhere Priorität nutzen? Yes
- Passwort für den Super-User: beliebiges Passwort

Anschließend bearbeiten wir mit einem Editor

- `sudo nano /etc/mumble-server.ini`

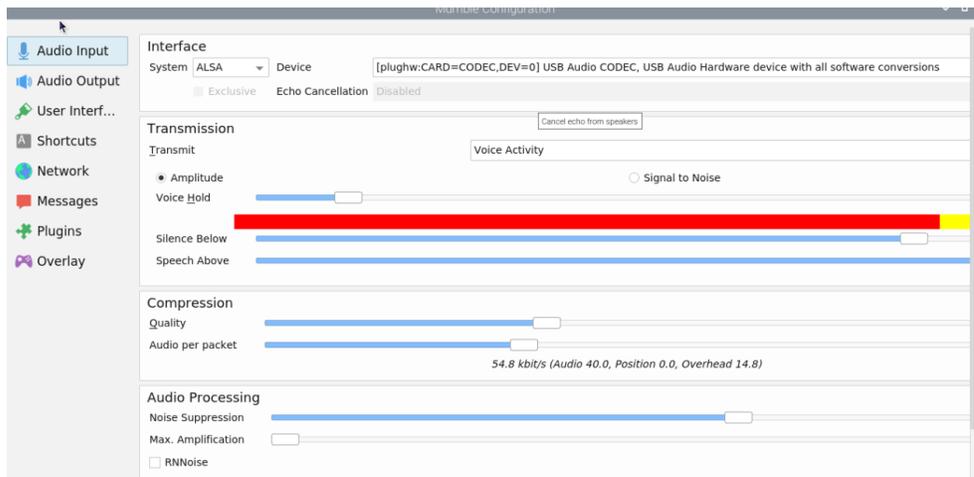
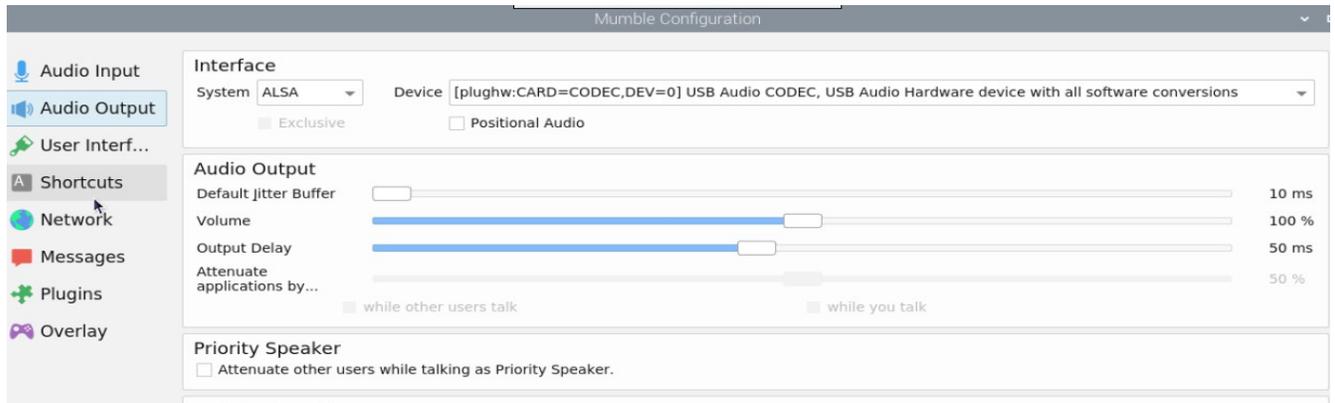
und passen diese an (Welcome Banner, Password, ...).

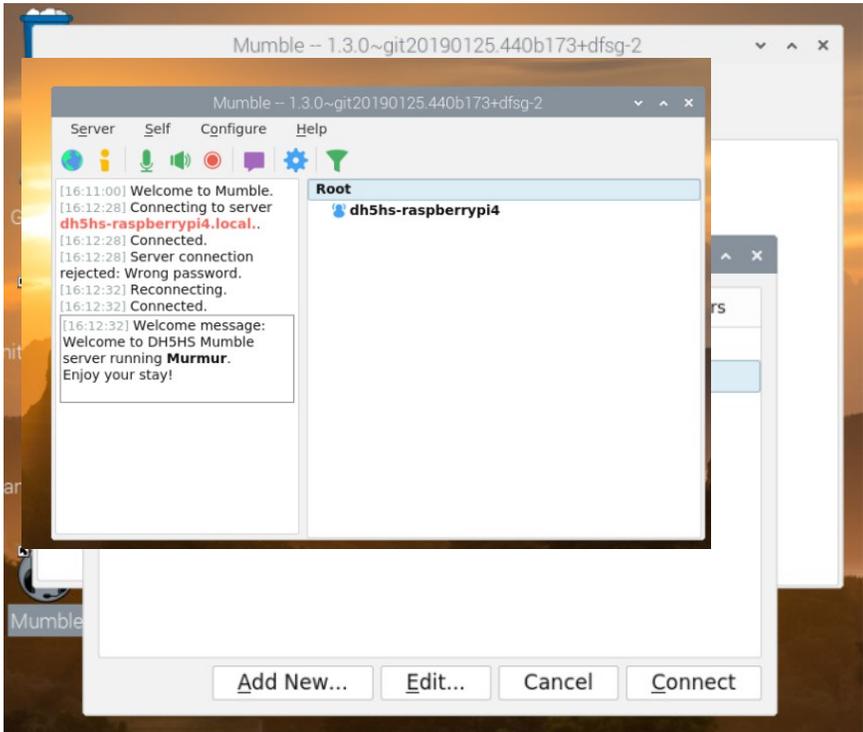
2. Mumble Client auf Raspberry installieren

Der Client wird über add/remove Software auf der grafischen Oberfläche installiert.

Nun kann der Client gestartet werden und der Server hinzugefügt werden. Audio im Client konfigurieren. Dann verbinden.



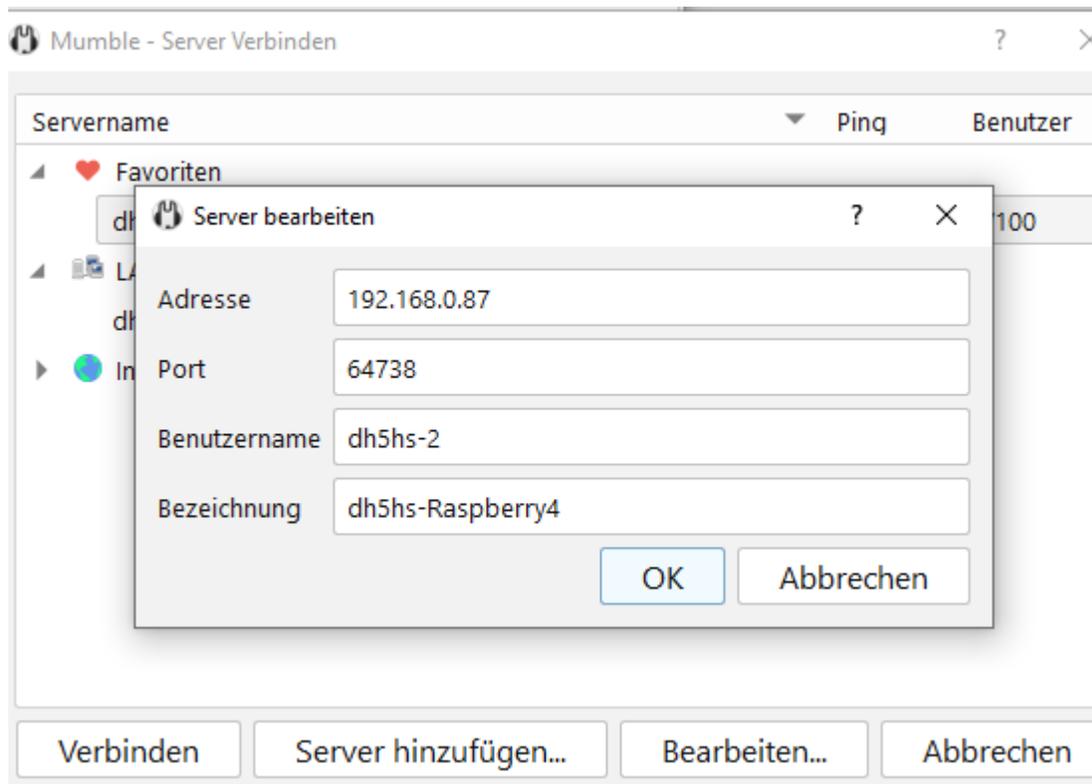




3. Mumble Client auf Windows Notebook/PC installieren

Client herunterladen und installieren.

Server hinzufügen wählen – dann verbinden – hier im LAN – für Internet Router konfigurieren und ggf. DYNDNS Provider.



Extra Teil 1: Scripte

Mumble Server restart

- `sudo /etc/init.d/mumble-server restart`

Mumble Client von aus dem Terminal starten und stoppen für den Server *dh5hs-raspberry4* und den User *dh5hs-raspberry4*

- `ssh pi@dh5hs-raspberrypi4 "DISPLAY=:0 nohup mumble mumble://dh5hs-raspberrypi4:[password]@127.0.0.1:64738/LAN/dh5hs-raspberrypi4/Y"`
- `killall -v mumble mumble`

VNC Server starten

- `vncserver -geometry 1360x768 -depth 24`

VNC Server stoppen

- `echo '*** stopping VNC displays ***'`
- `vncserver -kill :1`
- `vncserver -kill :2`
- `vncserver -kill :3`
- `vncserver -kill :4`
- `vncserver -kill :5`
- `vncserver -kill :6`
- `vncserver -kill :7`
- `vncserver -kill :8`
- `vncserver -kill :9`

Zugriffprobleme COM Port

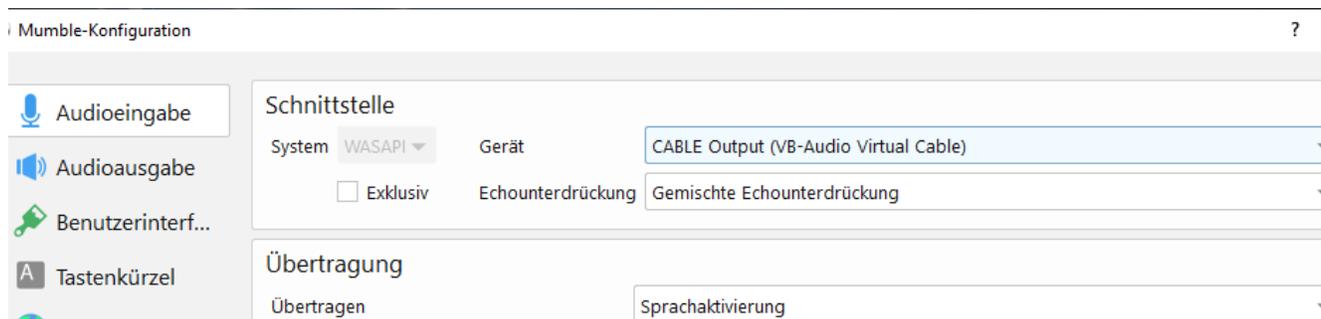
- `sudo chown pi /dev/ttyUSB0`
- `sudo chmod 777 /dev/ttyUSB0`

Extra Teil 2 – Konfiguration WSJTX (FT8) für Remotebetrieb in dieser Konfiguration

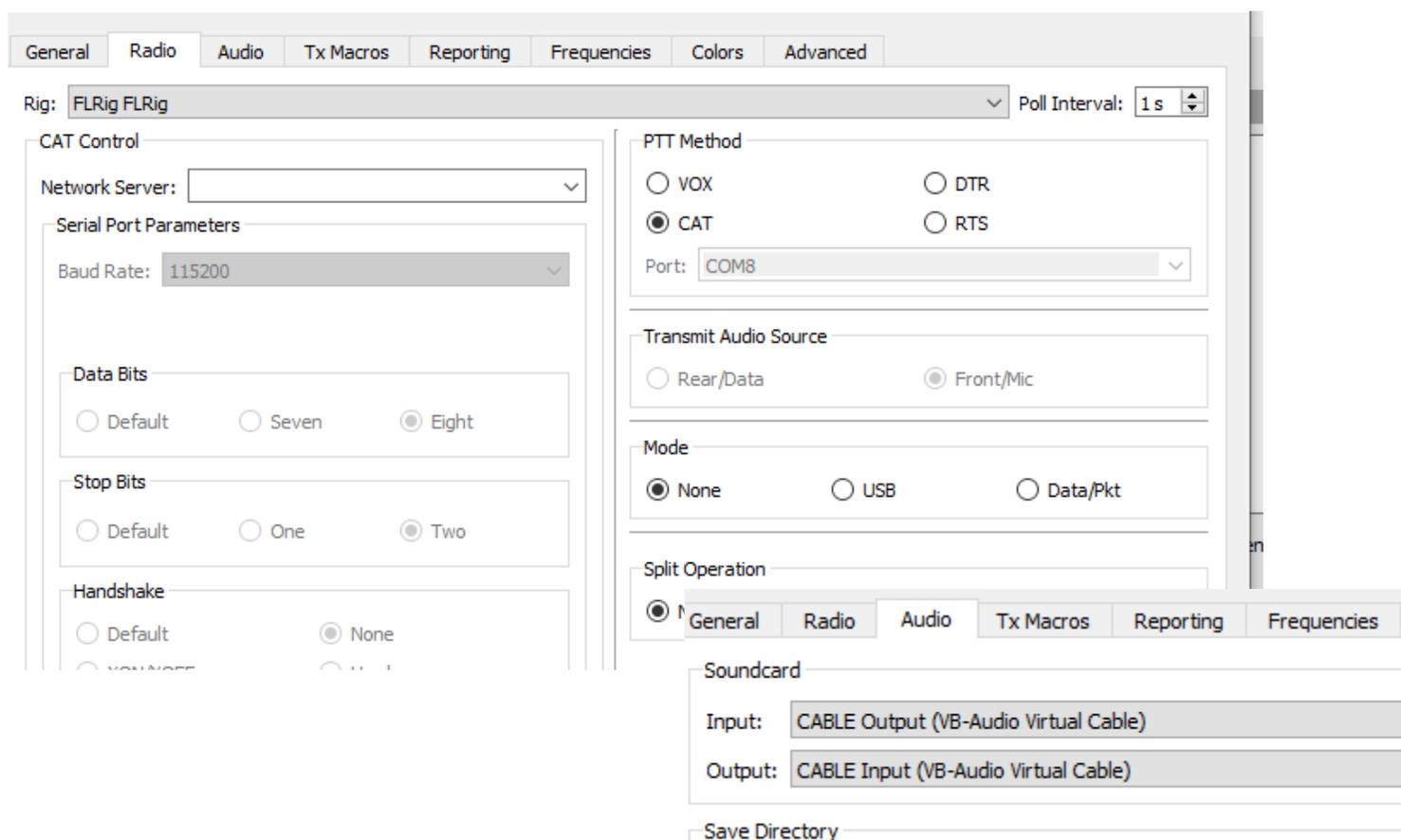
Um WSJTX zu betreiben sind lediglich auf dem Windows Notebook/PC Einstellungen vorzunehmen.

Es wird ein virtuelles Audio Kabel benötigt. Ich nutze hier die Software VB-Audio Cable von <https://vb-audio.com/Cable/>.

Im Mumble Client (Windows) die Audioeingabe auf *CABLE Output* und die Audioausgabe auf *CABLE Input* konfigurieren.



WSJTX konfigurieren:



WSJT-X v2.3.0 by K1JT, G4WJS, and K9AN

File Configurations View Mode Decode Save Tools Help

Band Activity

UTC	dB	DT	Freq	Message	UTC	dB	DT	Freq	Message
154515	-19	0.4	2122	~ CQ UA3VE1 KO95 EU Russia					
154515	-3	0.4	624	~ J68HZ YO4W2 KN44					
154515	-18	0.3	2236	~ DB2QM UA3QUP -06					
154515	-13	-0.1	2755	~ AB1NS RU3DF -20					
154515	-12	0.6	1787	~ NL7XT R6DOB KN95					
154515	-15	0.4	503	~ DL9YEH SV3SPD RR73					
154515	-15	0.5	1623	~ OH3OPW OH7OE -12					
154515	-16	0.4	792	~ CQ VE7ON CN89 Canada					
154515	-12	0.6	1592	~ SP6QM RD4D LO32					
154515	-18	0.5	419	~ A71CT W7YED -03					
154515	-17	0.4	888	~ CQ RN6LBT LN17 EU Russia					
154515	-24	0.4	1481	~ DL5XAH UA3DHR R-06					
154515	-21	0.4	905	~ F1RUK FD0MBU JO21					
154515	-24	0.4	878	~ DL4HY R2DTR -10					

CQ only Log QSO Stop Monitor Erase Decode Enable Tx Halt Tx Tune Menu

20m 14,074 000 Tx 1481 Hz Hold Tx Freq Rx 1163 Hz

DX Call: 7X2ARA DX Grid: JM16 Az: 191 1644 km Report: -15 Auto Seq Call List

2021 Mrz 16 15:45:39

Generate Std Msgs Next Now Pwr

7X2ARA DH9SH JO31	<input type="radio"/>	Tx 1
7X2ARA DH9SH -15	<input type="radio"/>	Tx 2
7X2ARA DH9SH R-15	<input type="radio"/>	Tx 3
7X2ARA DH9SH RR73	<input type="radio"/>	Tx 4
7X2ARA DH9SH 73	<input type="radio"/>	Tx 5
CQ DH9SH JO31	<input checked="" type="radio"/>	Tx 6

Receiving FT8 18 9/15 WD-3m

WSJT-X - Wide Graph

Controls 600 800 1000 1200 1400 1600 1800 2000 2200 2400 2600

flrig IC-7300

File Config Memory Keyer Help tcpip

14074.000 14074.000

S3 S6 S9 +20 +40 +60 VfoA VfoB A<->B Split
 Ps 40 80 120 160 1 3000 USB-D

Vol 0 MED 100 SQL 14 NR 4 Lock 0 ChPBT 0 Nch 1440 Mic 50 Pwr 5

ATT PRE NB AN Tune PTT

Mumble -- 1.3.1

Server Selbst Konfiguration Hilfe

Root dh5hs-2 dh5hs-raspberrypi4

[16:15:33] Willkommen in Mumble.
 [16:15:34] Version 1.3.4 ist jetzt verfügbar. Die neue Version kannst du dir hier herunterladen: [our server](#)

[16:29:02] Zu Server 192.168.0.87 verbinden.
 [16:29:07] Verbunden.
 [16:29:07] Willkommensnachricht: Welcome to DH5HS Mumble server running **Murmur**. Enjoy your stay!

Nachricht an den Kanal 'Root' hier einsehen