

mumble
*vom Voice-Chat zum
einfachen Remotesender*

11. Juni 2020

mumble

- Die Software realisiert ein klassisches Client-Server-System.
- Der Client „Mumble“ stellt eine grafische Oberfläche für Unterhaltungen und zur Administration bereit, der Server „Murmur“ realisiert das Back-End, über das die Gespräche laufen.

mumble

- Der offizielle Client läuft unter [Windows](#), [Linux](#) und [macOS](#). Für [Android](#) und [iOS](#) gibt es mehrere alternative Clients (z. B. *Plumble*).
- Der Server „Murmur“ lässt sich auf beinahe beliebigen Systemen kompilieren und ausführen.

Name	Platform	Programming language	Comment	License	Weblink
Plumble	Android	Java	<i>Unmaintained now.</i> Voice activation (with customizable threshold) and push-to-talk, dual-pane channel and chat with gestures, draggable overlay, Bluetooth headset support, customizable theming, proximity sensor utilizing "Voice Call" mode, push to talk 'hot corners' toggleable in settings, hardware push-to-talk key support, chat notifications and more!	GPLv3	GitHub
Mumla	Android	Java	Fork (aka continuation) of Plumble.	GPLv3	GitLab
Mumblefy	iOS	Objective C	Appears to be discontinued, last update was 2016.	Proprietary	Website
mumble-web	Web	HTML5 and JavaScript	Quite a few features are still missing, most noticeably voice activity detection and all administrative functionality.	ISC	GitHub
barnard	Linux	Go	barnard is a terminal-based client with a curses-like UI. Built on the Gumble library.	GPLv2	GitHub
talkKconnect	Linux	Go	talkKconnect is a Linux CLI Headless Self Contained Mumble Client For Raspberry Pi with LCD, Channel Control and Granular XML Config. Using GPIOs you can interface with RF radios or other network radio technologies.	MPL	talkkconnect

mumble

- Für was lässt sich also Mumble nutzen?
 - Klassische Kommunikation von PC zu PC
 - ähnlich Teamspeak
 - jedoch auf vielen Plattformen lauffähig → Raspberry Pi
 - Kommunikation mit dem Raspberry Pi oder Smartphone
 - Einfache Intercom-Lösung für den Heimanwender

talkKconnect

- talkKconnect ist ein
 - ▣ Headless Self Contained Mumble Client
 - basiert auf mumble, keine GUI, Terminalsoftware
 - ▣ mit LCD-Ansteuerung
 - ▣ Kanalwahl

- talkKconnect ist also ein Raspberry Pi Walkie Talkie, Intercom, Funkgerät, IP-Radio, Gateway

„Remote-Sender“ Klippeneck

- Der P10 Mai-OV-Abend war quasi der 1. Test für den 2m FM-Remote Sender auf der 144.625 auf dem Klippeneck
- Hier kommt derzeit ein Kenwood TK-7180 Betriebsfunkgerät zum Einsatz
[~20W, X-5000, 1000m üNN]
- Hier können auch weitere Funkgeräte hinzugefügt werden oder online eingesprochen werden (mumble)

talkKconnect

- Wie kann ich funken?
 - ▣ mit dem PC / Smartphone auf den Server verbinden
 - Passwortgeschützt, somit Zugang geregelt
 - Rechteverteilung via mumble
- „OV-Telefon“
 - ▣ wenn sowieso ein Raspberry Pi mitläuft:
 - kleiner Lautsprecher dran → immer empfangsbereit
 - Senden über Mikrofon, z. B. Handfunkgeräte-Mikrofon
 - ▣ Sender mit erhöhter Reichweite / Teilnahme an OV-Runden bei z. B. beruflicher Abwesenheit, Urlaub

talkKconnect

Mumble -- 1.3.0

Server Selbst Konfiguration Hilfe

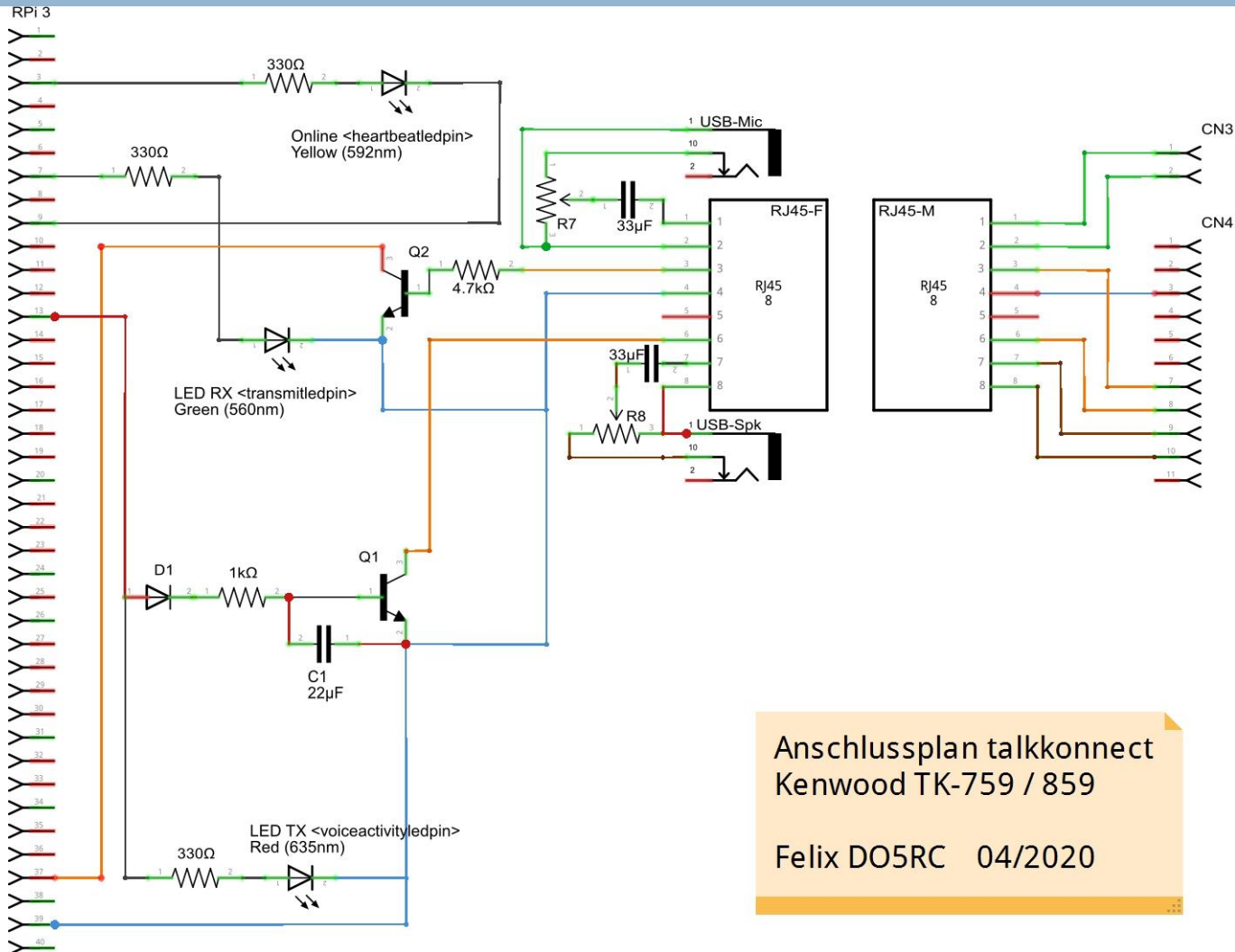
fehlgeschlagen: Der entfernte Rechner hat die Verbindung geschlossen.
[16:47:06] Neuverbindung.
[16:47:07] Verbunden.
[16:47:07] Willkommensnachricht:
Welcome to this server running **Murmur**.
Enjoy your stay!
[16:47:07] Maximale Netzwerkbandbreite des Servers beträgt nur 72 kbit/s. Audioqualität wurde automatisch auf 57 kbit/s (20

Nachricht an den Kanal 'DB0KB 439425' hier eingeben

Root

- DL1TOB_Shack2
- Repeater
- Simplex Link
- ▲ Klippeneck 144625
 - DL1TOB ✓
 - DL1TOB_pi_WoMo ✓
 - TRX_2m
- ▲ **DB0KB 439425**
 - DL1TOB_2 ✓
 - Felix_DO5RC** ✓
- ▲ RX TRO
 - DL1TOB_p ✓
 - RX_TRO ✓

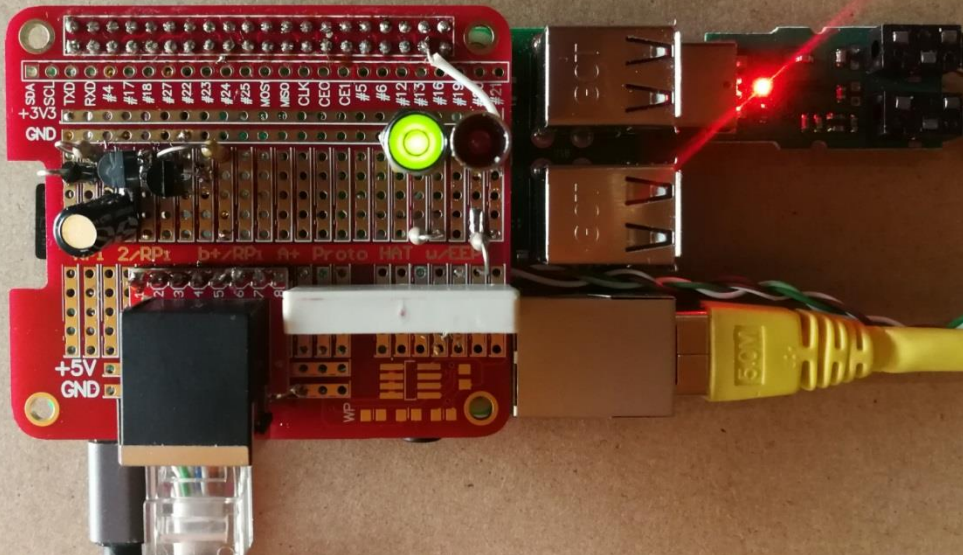
„Remote-Sender“ Klippeneck



Anschlussplan talkkonnnect
Kenwood TK-759 / 859

Felix DO5RC 04/2020

„Remote-Sender“ Klippeneck



1. Entwurf auf
Raspberry Proto-Shield



benötigte Hardware

- Raspberry Pi (ab Version 3)
 - ▣ Raspbian / Raspberry Pi OS
 - einfachste Version reicht, GUI nicht erforderlich
 - ▣ Speicherkarte, 4GB ausreichend
- USB Soundkarte
- Aufsteckplatine für den Pi (Proto HAT)
- RJ45 Buchse ohne LEDs
- RJ45 Stecker mit Kabel
- Kleinteile

talkKconnect / mumble

- Remote-TRX
 - ▣ zu Hause (Funkgerät an den Pi anschließen, von überall nutzen)
 - ▣ im Verein (als „Empfänger“ für die OV-Welle)
 - ▣ auf dem Berg als Remote-Station mit hoher Reichweite
 - ▣ unterwegs – nicht im Funkbereich, aber trotzdem „dabei“
 - ▣ Camper / Leihwagen / Firmenwagen (Antennenverbot)



talkKconnect / mumble

□ Repeater

- simple Methode um kurzfristig variable Repeater zu bauen

- NOTFUNK
- Fieldday
- Veranstaltung

□ Crossband

- Camper / Anhänger – einfaches HFuG mitnehmen und im Fahrzeug das Funkgerät laufen lassen (und 2 Pis)

□ Rundspruch

- eine Software und beliebig viele Endgeräte
- Sprecher(un)abhängig
- Vernetzung über HAMNET / AREDN
 - bessere Qualität als VoIP-Telefonie, weniger Daten

talkKconnect / mumble

□ Digitalfunk?

□ Ja, es lassen sich „Brücken“ bauen

- Allerdings entfallen dann die Systemfunktionen wie Nachrichten, Talkgroups, mehrere Zeitschlitz
- Audio wird von Digital zu Analog gewandelt und im nächsten System wieder digitalisiert → Qualität leidet
- Nützlich? Ja, um Systeme zu testen oder Nutzern die Möglichkeit zu geben, auf DMR / C4FM / D-Star etc. hereinzuhören ohne ein Gerät kaufen zu müssen.

talkKconnect / mumble



talkKconnect / mumble

