

# **Saba Webradio Hardware**

Autor: Martin Wagner, DL2WAG  
Dokumentart: Schaltungsbeschreibung, Messungen  
Datum: 8. August 2022 16:46

Version	Datum	Bemerkung
1	05.06.2022	Neu

# Inhaltsverzeichnis

<b>1. Saba Webradio</b>	<b>3</b>
<b>2. Empfohlene Modifikationen</b>	<b>4</b>
2.1. Magisches Auge . . . . .	4
2.2. Erkennung TA aktiv . . . . .	4
<b>3. Aufbau</b>	<b>7</b>
3.1. Material . . . . .	8
3.2. Anschlussstecker Radio . . . . .	9
3.3. Steuerung . . . . .	11
3.4. Montage . . . . .	13
3.5. Inbetriebnahme / Test . . . . .	14
3.6. Programmierung ESP32 . . . . .	22
<b>4. Bedienung</b>	<b>24</b>
4.1. W-Lan Anmeldung . . . . .	24
4.2. Webradio . . . . .	26
4.3. Fernbedienung . . . . .	28
4.4. Einstellungen . . . . .	28
4.5. Weitere Funktionen . . . . .	29
4.6. OTA Firmwareupdate . . . . .	29
<b>5. Schaltungsbeschreibung</b>	<b>30</b>
5.1. Datenblätter . . . . .	30
5.2. Microcontroller . . . . .	31
5.3. Audioschaltung . . . . .	31
5.4. Steuerung Radio . . . . .	32
5.5. Steuerung Webradio . . . . .	33
<b>6. Untersuchungen</b>	<b>34</b>
6.1. Probleme . . . . .	34
6.2. Ausgangsübertrager . . . . .	36
6.3. W-Lan Antenne . . . . .	42
<b>A. Ausgabe md5sum Webserver</b>	<b>43</b>
<b>B. Ausgabe esptool.py</b>	<b>44</b>
<b>C. Ausgabe serielle Schnittstelle beim Boot</b>	<b>45</b>

## 1. Saba Webradio

Das Saba Webradio erweitert ein Saba Automatic Radio um die folgenden Funktionen:

- W-Lan Webradio
- Steuerung von Automatik und Lautstärke per Webinterface, MQTT, IR Fernbedienung ...
- Steuerung des Webradio per Radio Automatiktasten

Es kann direkt auf die Anschlussbuchse für die Fernbedienung aufgesteckt werden. Kompatibel sind mind. das Meersburg und Freiburg 6-3D. Am Gerät selbst sind keine Modifikationen notwendig (aber empfehlen). Der Audioausgang des Webradio wird mit dem TA Eingang des Radios verbunden und kann durch Wahl des zugehörigen Eingangs am Radio verwendet werden.

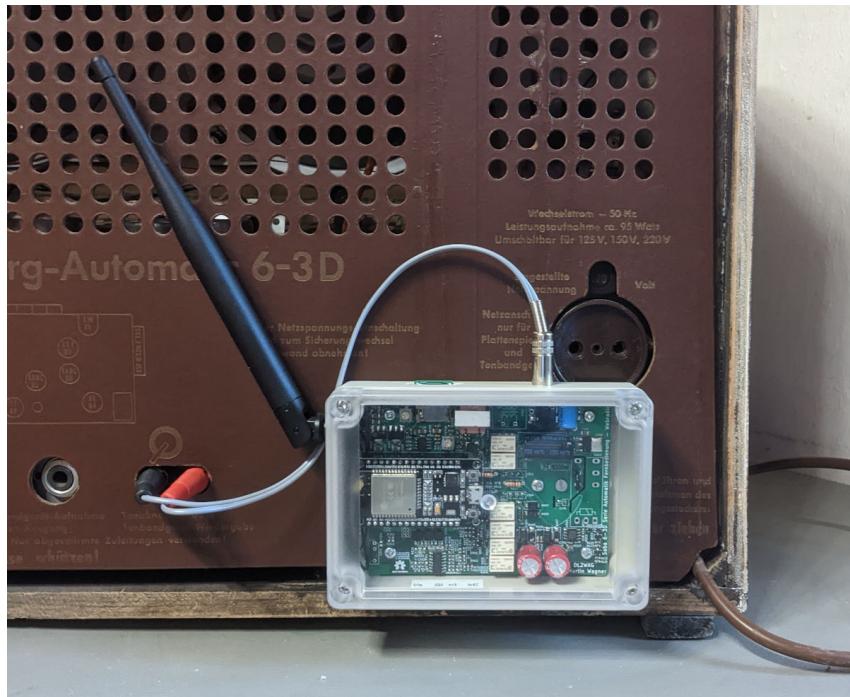


Abbildung 1: Fertig montiertes Saba Webradio

Das Webradio basiert auf dem ESP32-Radio Projekt von Ed Smallenburg: <https://github.com/Edzelf/ESP32-Radio>.

**Hardware:** <https://github.com/martin-wagner/saba-6-3d-fernbedienung-hw>  
**Software:** <https://github.com/martin-wagner/ESP32-Radio>

## 2. Empfohlene Modifikationen

### 2.1. Magisches Auge

Wird häufig die Webradiofunktion verwendet, sollte das magische Auge im TA Betrieb deaktiviert werden. Bei neueren Saba Geräten ist dies ab Werk der Fall, bei älteren wie dem 6-3D wie folgt modifizieren:

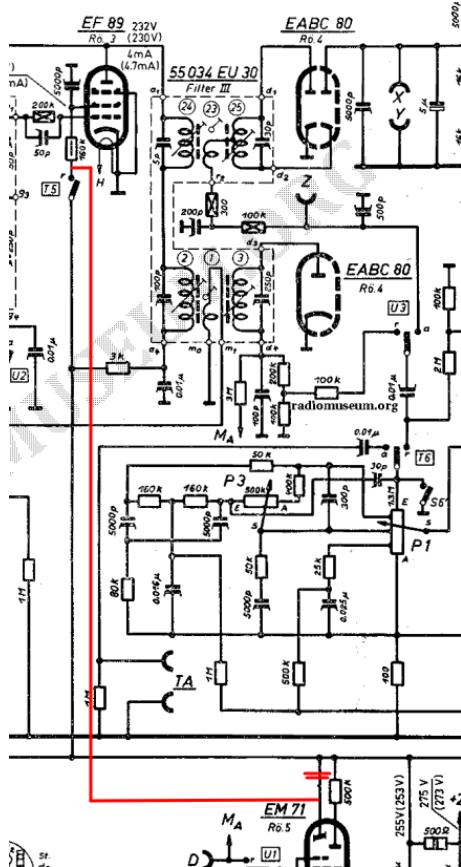


Abbildung 2: Schaltungsvorschlag

Um ein ggf. neu gekauftes Auge zu schonen kann bei diesem Umbau auch direkt ein Poti 1 W 50 kOhm in Serie eingeschleift werden. Je niedriger die Helligkeit eingestellt wird desto höher ist die Lebensdauer des Auges. Ohne Widerstand ist bei der verwendeten EM71 ein Abnutzungseffekt bereits nach wenigen Betriebsstunden (!!!) sichtbar.

### 2.2. Erkennung TA aktiv

Das Webradio hat keine Möglichkeit zu erkennen ob gerade normales Radio oder Webradio aktiv ist. Der Webradio Stream ist daher immer eingeschalten. Abgesehen von unnützer Date-

nübertragung führt die W-Lan Kommunikation zu Störungen im Radioempfang bei schwächeren Sendern. Soll dies vermieden werden kann das „TA Aktiv“ Signal von der Glühlampe der TA Taste abgegriffen und an die Fernbedienungsbuchse verdrahtet werden. Hierfür wird der freie Kontakt „11“ verwendet.

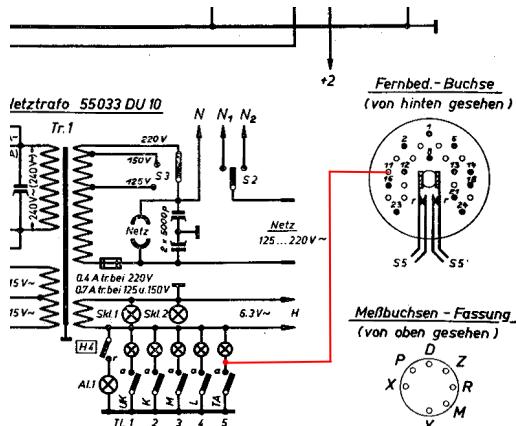


Abbildung 3: Schaltungsvorschlag

Unbenutzte Kontakte sind leer, daher muss ein Kontaktstift aus einem Schlachtgerät entfernt werden. Alternativ kann die 230 V Wechselschaltung geopfert und einer der zugehörigen Kontaktstifte versetzt werden.

- Leitung ablöten



Abbildung 4: Kontakt ohne Leitung

- Lötose reinigen und geradebiegen



Abbildung 5: Kontakt gerade

- Feder zur Geräte Außenseite herausschieben



Abbildung 6: Kontaktstift

- Umsetzen und Schaltung nach Vorgabe herstellen



Abbildung 7: Kontakt an neuer Position



Abbildung 8: Anschluss an Lampenfassung

### 3. Aufbau

Die Schaltung ist auf drei Platinen aufgebaut:

- Anschlussstecker Radio

- Webradio Steuerplatine
- ESP32 Aufsteckplatine

Der Anschlussstecker wird über eine 2,54 mm Stiftleiste mit der Steuerplatine verbunden. Dies ermöglicht ein Anpassen des Steckers auf Gerätetypen mit anderer Buchse. Das ESP32 wird einfach auf die Basisplatine aufgesteckt und kann bei Bedarf entfernt werden.

### 3.1. Material

Deutsche Distris versenden nur an Kunden mit Steuernummer (Firma). Internationale Distris machen keine Unterscheidung. Ich habe daher alles was *Reichelt* nicht hat bei Digikey in den USA bestellt. Alle Bauteile mit Herstellervorgabe, z.B. *WE*, können von einem beliebigen Hersteller kommen, solange die Kenndaten eingehalten werden.

Folgendes Material wird benötigt:

- Bauteile nach „stueckliste.csv“ für Anschlussstecker und Basisplatine. Bauteile mit Markierung *NP* (No Placement) werden i.d.R. nicht benötigt (siehe Rest der Anleitung). Die erste Hälfte der Liste sind alle Bauteile einzeln mit Index, die zweite Hälfte die Summe der Bauteile.
- Jeweils eine Platine Anschlussstecker und Steuerung. Anschlussstecker – 2lagig, Steuerung – 4lagig. Bestellen z.B. bei *JLC-PCB*, die Platten sind kleiner als 10 cm x 10 cm. Falls man Reflow Löten will den Stencil gleich mitbestellen. Auf der Unterseite sind keine Bauteile.
- 2x ESP32 DOIT-DEVKITV1-ESP32-WROOM-32. Bestellen z.B. bei Ebay. Es gibt ESP32 Boards auch mit zwei Pins mehr, daher Pinbelegung vergleichen! Der zweite ist notwendig falls man vermutet den ersten zerstört zu haben.
- VLSI VS1053B. Bestellen z.B. bei Digikey (teuer) oder Ebay direkt aus China (Risiko gefälschte Chips zu bekommen). Ich habe bei Ebay bestellt und meine Lieferung war OK.
- 2x Buchsenleiste 2,54mm 15pol, für ESP32
- 6x Stiftleiste 2,54mm 4pol,  $l \geq 20mm$ , für Verbindung der beiden Platten
- Ausgangstrafo 600 Ohm, 1:1. Ebay, Suche nach *transformer EI14*. Falls nicht zu bekommen kann auch ein anderer verwendet werden, siehe 6.2.
- Klinkenstecker 3,5 mm + Audiokabel + 2x Bananenstecker.
- Gehäuse Hammond RP1125C. Das exakt gleiche Gehäuse gibt es auch von anderen Herstellern.
- Kunststoff- / Holzstab rund,  $d = 6 mm$ , ca. 5 cm lang. Ein Bleistift funktioniert auch, bricht aber leicht ab.

- Messingstab 3 mm x 1,5 mm, mind. ca. 30 cm. Wenn man es bekommt ist 3 mm x 1,2 mm besser, ggf. geht auch 3 mm x 1,0 mm.
- Ein Set SMD Widerstände 0603, wenn noch nicht vorhanden. Notwendig falls ein Widerstandswert angepasst werden muss. Ebay, Suche nach *resistor kit 0603*.
- Ggf. eine W-Lan Antenne.
- Gut gefüllte Teilekiste für alles was ich vergessen habe.

### 3.2. Anschlussstecker Radio

Wir brauchen:

- Platine
- Messingstab
- Holzstab

Aus dem Messingstab 13 Stifte mit je 15 mm Länge herstellen. Vorne zuspitzen so dass die Stifte in die Öffnung des Federkontakte passen (siehe Bild 6). Mit dem 1,5 mm Material werden die Federn etwas aufgebogen, 1,2 mm sollten genau passen. Ggf. reicht auch 1,0 mm aus, habe ich nicht probiert.



Abbildung 9: Kontaktstift

Die Stifte in die Kontaktfedern drücken und danach die Platine aufsetzen. Ggf. Stifte etwas abrunden damit diese besser in die Platine passen. Stift 11 hat ggf. keine Feder, siehe 2.2. Falls Modifikation nicht durchgeführt wird diesen Stift weglassen.



Abbildung 10: Stecker

Stifte in dieser Position verlöten. Nur sehr kurz und mit hoher Leistung, sonst schmilzt der Kunststoff der Buchse! Mit Widerstandsmessgerät prüfen ob alle Stifte guten Kontakt zum jeweiligen Kontakt in der Buchse haben, ggf. nacharbeiten.

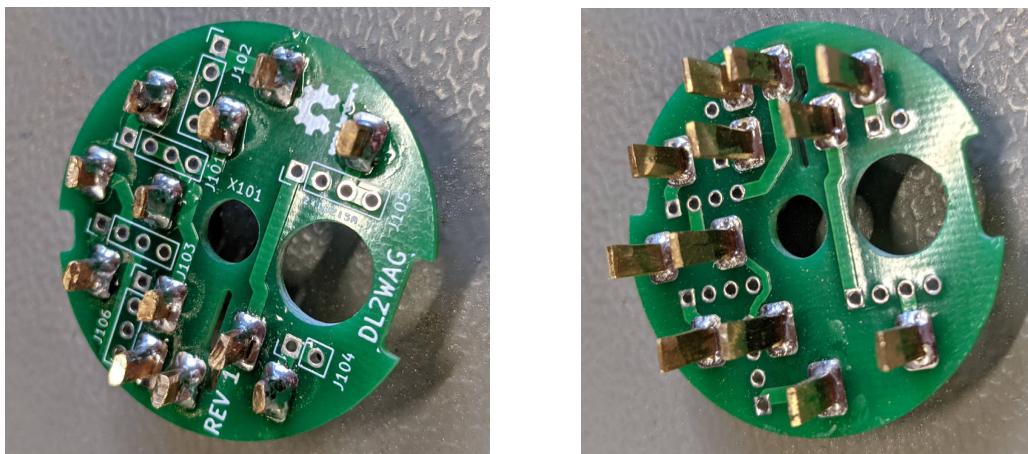


Abbildung 11: Stecker gelötet

Stifteleisten mit Platine verlöten. Kunststoff so verschieben das die Stifte auf der Unterseite bündig mit der Platine abschließen (und der Stecker bündig auf dem Kunststoff der Buchse aufsitzt).

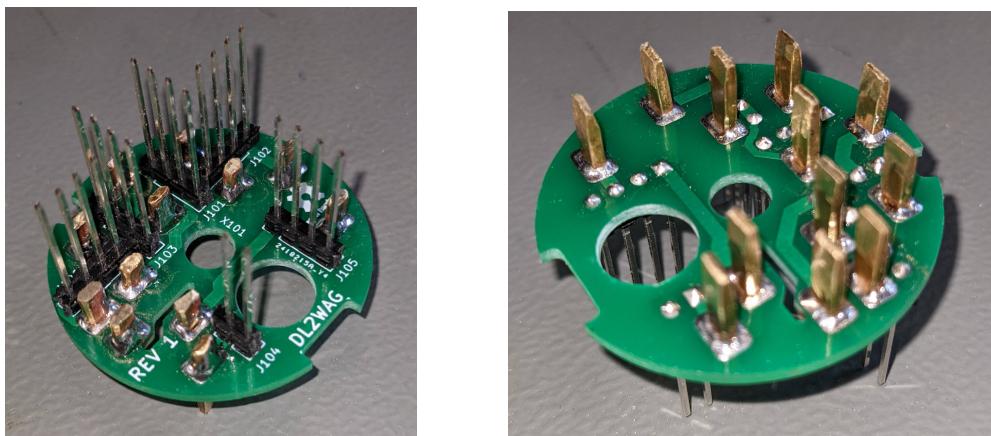


Abbildung 12: Stecker fertig

Holzstab auf ca. 4,5 cm kürzen und vorne mit Bleistiftpitzer anspitzen. Hinten Loch für Schraube bohren.



Abbildung 13: Zentrierstift

Der Stift muss durch das Loch in der Mitte des Steckers passen und wird auf die Steuerplatine aufgeschraubt.

### 3.3. Steuerung

Wir brauchen:

- Platine
- Bauteile

Zuerst nicht benötigte herausstehende Stücke der Platine absägen (USB und ggf. SD Karte). Dann kleine Bauteile auflöten (IC und ggf. SD Karte).

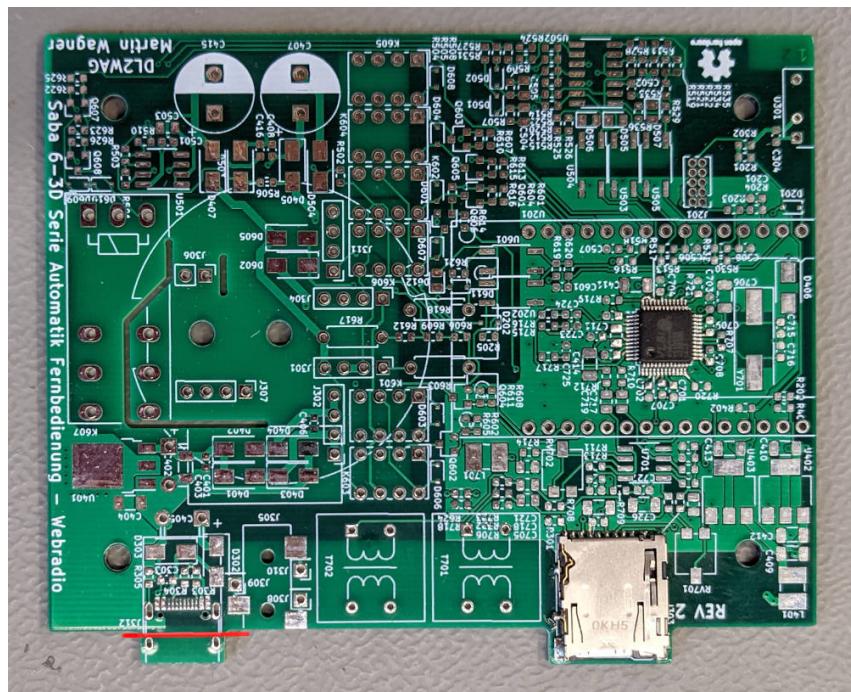


Abbildung 14: Bestückung kleine SMD Bauteile

Den Rest der Bauteile nacheinander bestücken, beginnend um IC herum. „NP“ Bauteile beachten (nicht bestücken!). Kontakt 230 V AC auf N1 oder N2 brücken, siehe 5.4.

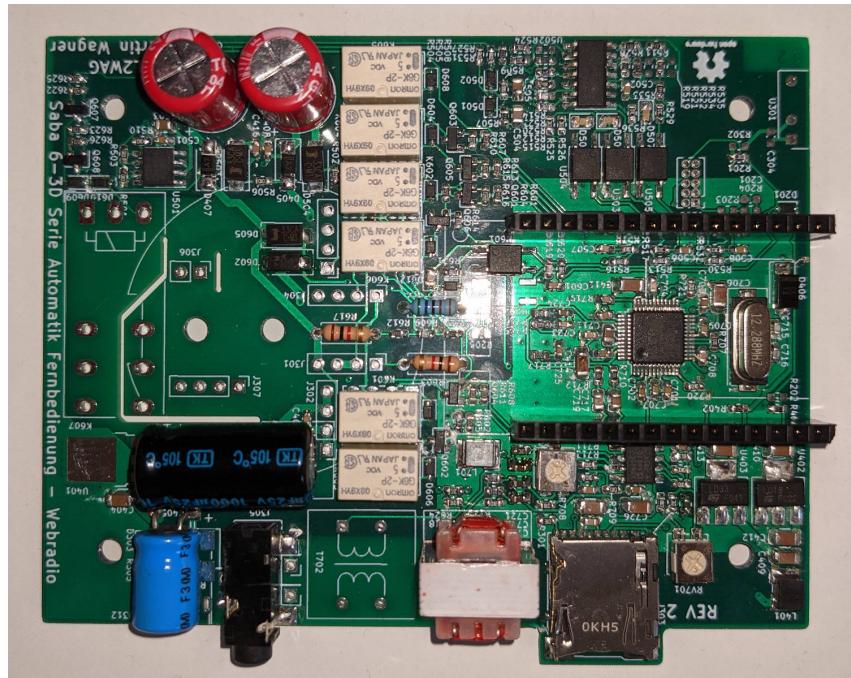


Abbildung 15: Bestückung fertig

### 3.4. Montage

Gehäuse für Montage der Platine und Stecker vorbereiten. Der Mittelpunkt des Stecker Bohrlochs kann durch das Loch für den Zentrierstift auf der Steuerplatine angezeichnet werden. Das Loch sollte so groß sein das der Stecker durchpasst (um die Baugruppe im späteren Verlauf herausnehmen zu können).



Abbildung 16: Gehäuse

Stecker in Buchse einstecken, Gehäuse mit montierter Steuerplatine bündig aufsetzen. Stifte einrasten und verlöten.

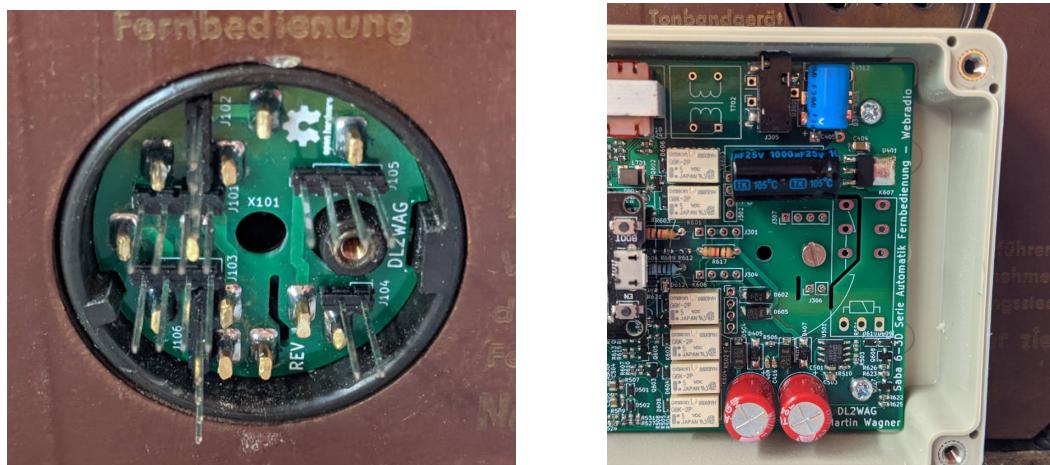


Abbildung 17: Stecker fertig

Zuletzt wird der Zentrierstift auf der Steuerplatine verschraubt.

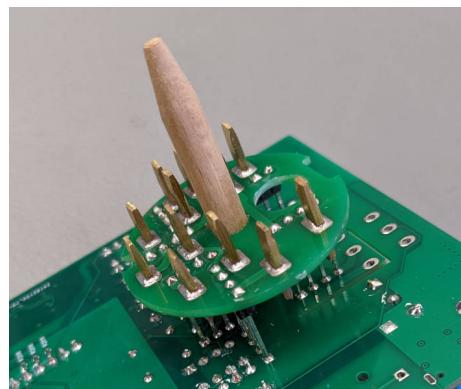


Abbildung 18: Zentrierstift im Stecker

Die gesamte Baugruppe kann jetzt nach dem Einstecken auf der Anschlussbuchse festgeschraubt werden. Vorsichtig, da Platine nicht auf der Buchse aufliegt.

### 3.5. Inbetriebnahme / Test

Die Steuerplatine sollte vor der Montage auf korrekte Funktion geprüft werden. Die einzelnen Schaltungsteile werden unabhängig voneinander betrachtet.

#### Spannungsversorgung Mikrocontroller

Die Versorgung wird aus der Heizspannung 6,3 V durch Vollweggleichrichtung + darauffolgend 5 V Linearregler gewonnen. Folgende Spannungen stehen zur Verfügung<sup>1</sup>:

- ca. 8 V
- 5,0 V
- 3,3 V Digital (werden vom ESP32 ausgegeben)
- 1,8 V Digital
- 3,3 V Analog

#### Spannungsversorgung Suchlauf

Die Versorgung wird aus der 2x15 V Spannungsversorgung für den Suchlauf durch Einweggleichrichtung gewonnen. Folgende Spannungen stehen zur Verfügung:

- ca. +/- 20 V

---

<sup>1</sup>Siehe Schaltplan Seite 3

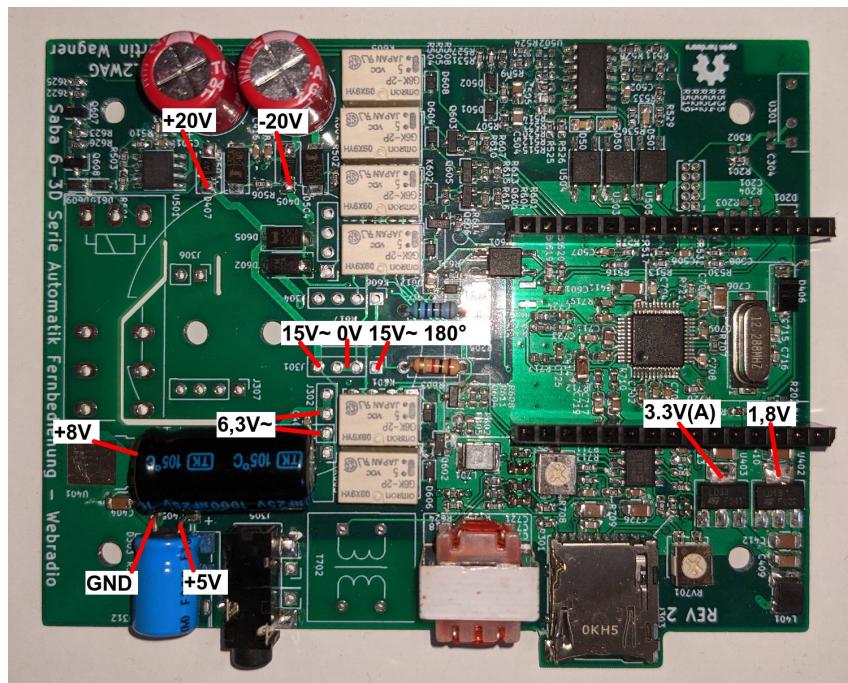


Abbildung 19: DC Spannungen Messpunkte

### Suchlauf Operationsverstärker / Komparatoren

Versorgungsspannung 2x15 V AC – +/-20 V. Die Comparatoren sind auf eine Versorgungsspannung von 22 V berechnet. Da in der Schaltung Spannungsverhältnisse verglichen werden sind die berechneten Schaltschwellen nur bei dieser Spannung gültig. Bei anderen Versorgungsspannungen verschieben sich die Schaltpunkte proportional zur Versorgungsspannung.

Alle Messungen mit Bezug auf die Mittelanzapfung der 15 V Wicklung, nicht Masse des Radios! Beim Messen aufpassen das nicht versehentlich die verschiedenen Massenpotentiale verbunden werden! Suchlauftaste am Gerät betätigen (Suchlauf aus) oder Suchlaufschaltung auf Breadboard nachbauen.

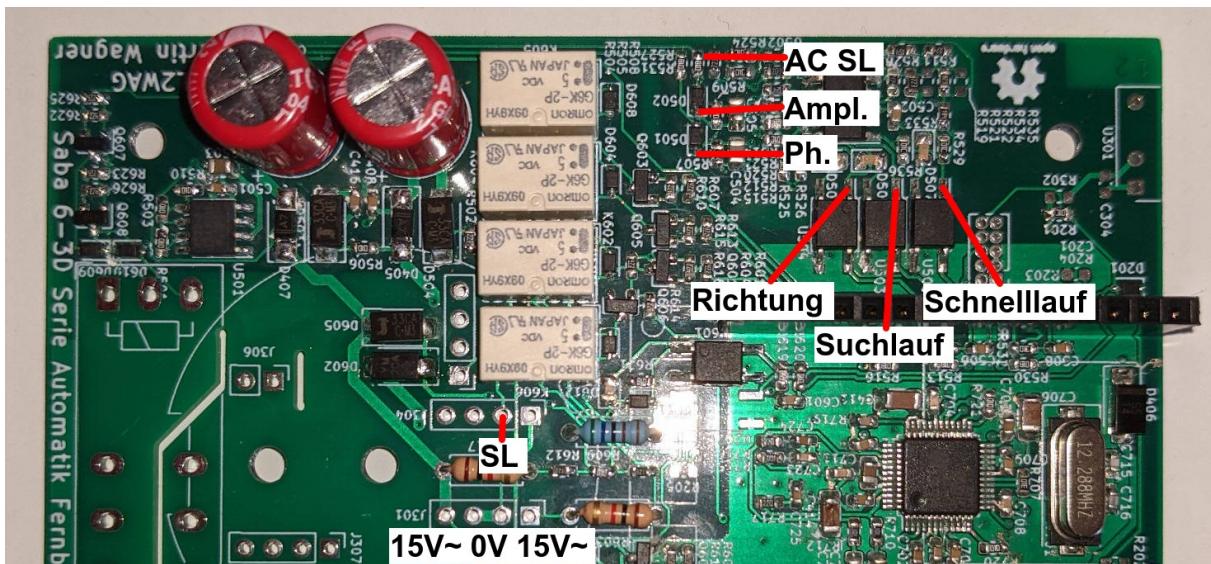


Abbildung 20: AC Messpunkte

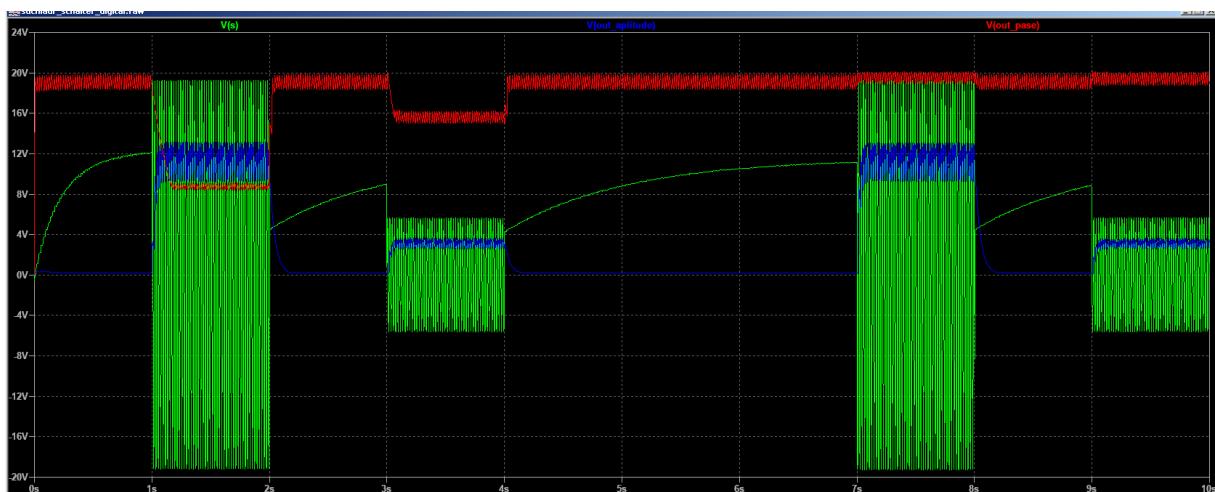


Abbildung 21: Spannungen in Simulation

### Messungen Signal Laufrichtung

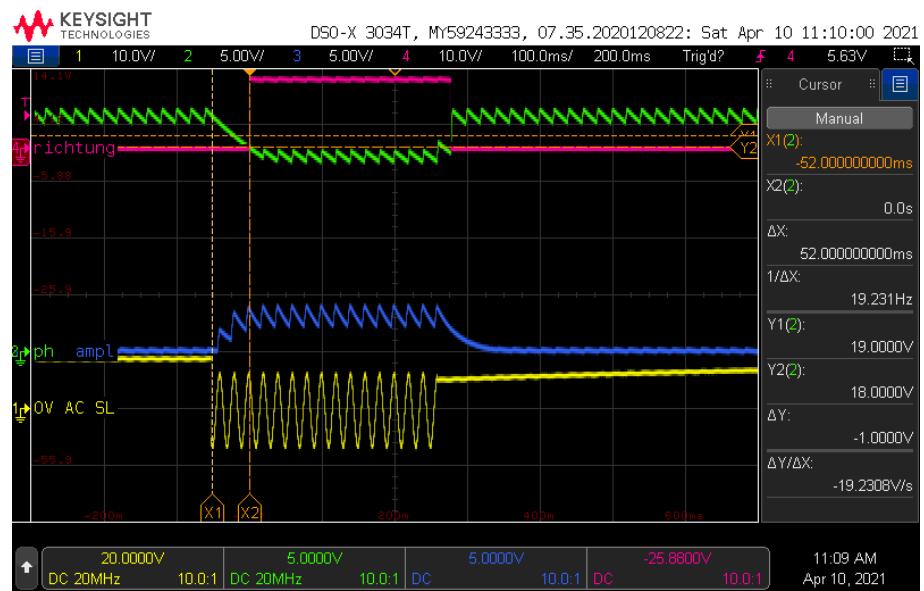


Abbildung 22: Laufrichtung 1, suchen

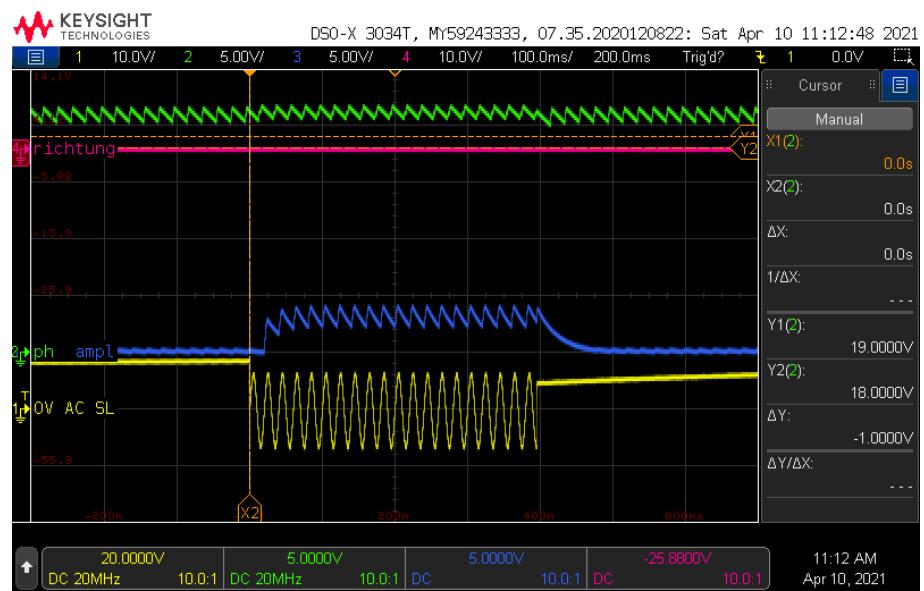


Abbildung 23: Laufrichtung 2, suchen

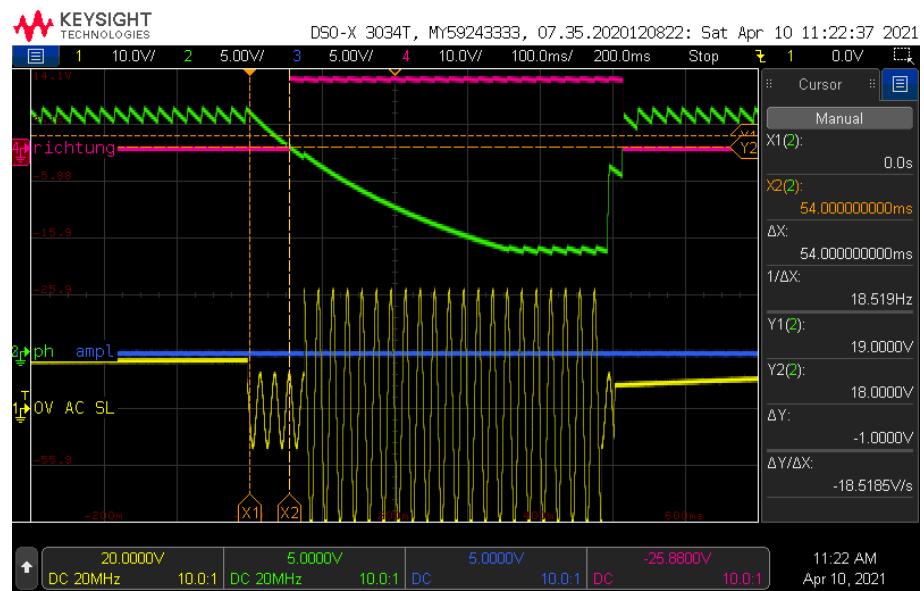


Abbildung 24: Laufrichtung 1, schnell

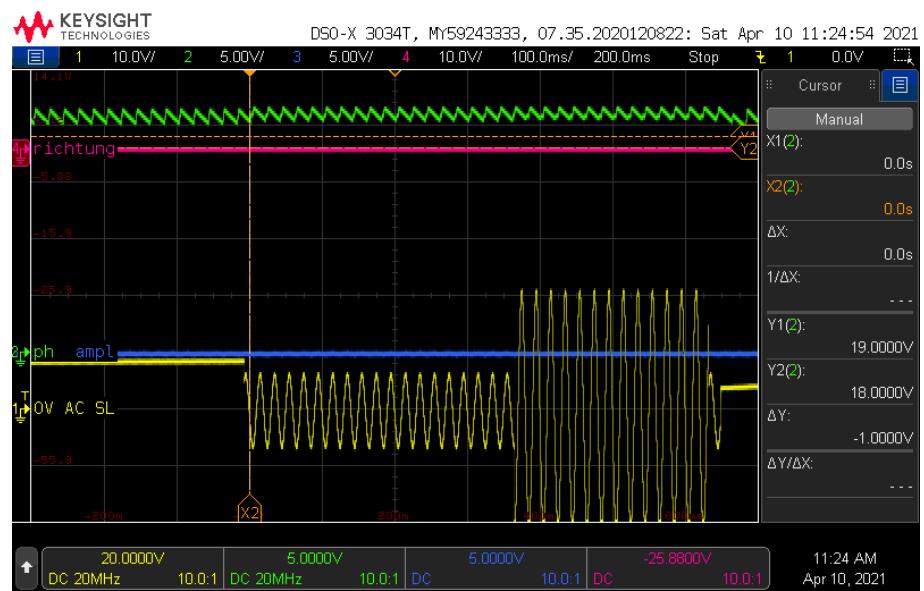


Abbildung 25: Laufrichtung 2, schnell

Die Laufrichtung muss korrekt erkannt werden. Die Verzögerung muss im Bereich bis max. ca 50 ms sein. Beim Schnelllauf wird zuerst der Kontakt für den Suchlauf geschlossen, bereits dieser führt zur Erkennung.

### Signal Laufrichtung im Bereich der zulässigen Versorgungsspannung

Die Versorgungsspannung für den Schaltungsteil ist nicht stabilisiert und schwankt daher in Abhängigkeit der Netzspannung.

Sweep Versorgungsspannung +/- 10V ... 18V AC.

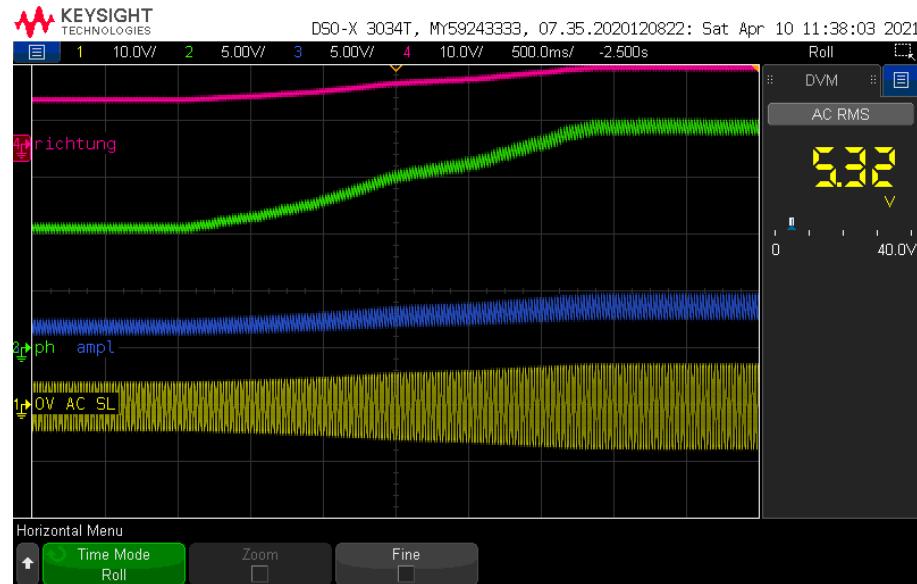


Abbildung 26: Laufrichtung 1, suchen

Das digitale Signal steht im gesamten Versorgungsspannungsbereich korrekt an.

### Messungen Signal Schnellauf

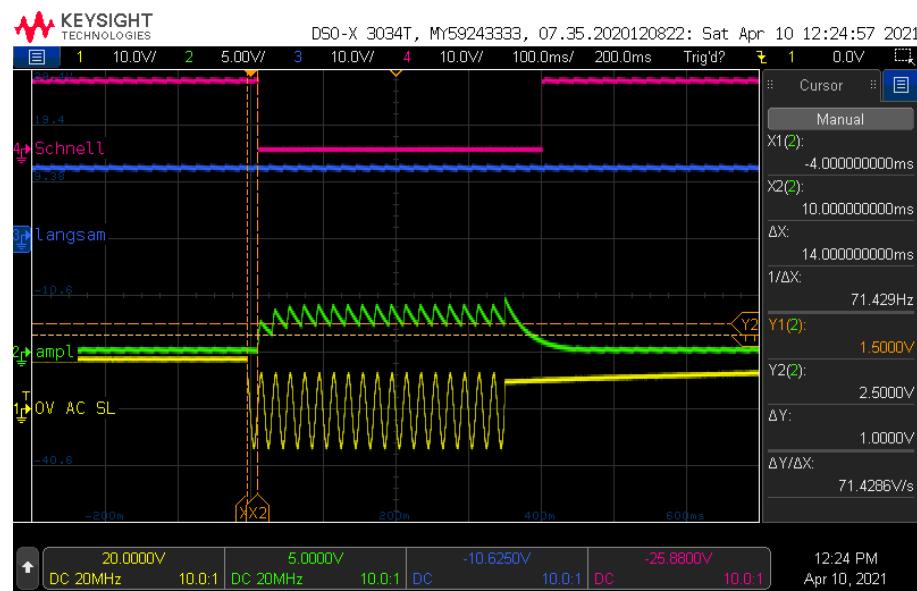


Abbildung 27: suchen

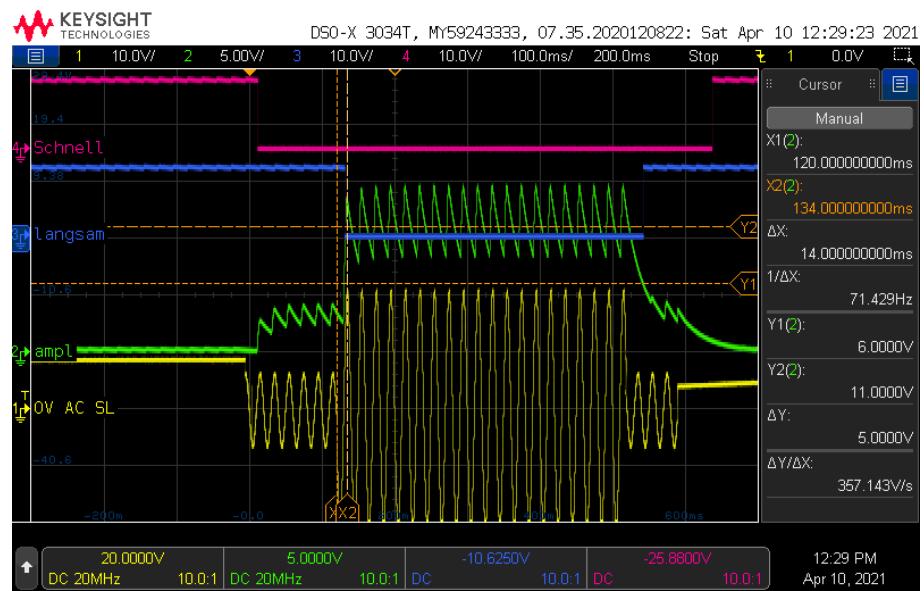


Abbildung 28: schnelllauf

Der Schnelllauf muss korrekt erkannt werden. Die Verzögerung muss im Bereich bis max. ca 50 ms sein.

### Signal Schnelllauf im Bereich der zulässigen Versorgungsspannung

Sweep Versorgungsspannung +/- 10 V ... 18 V AC.

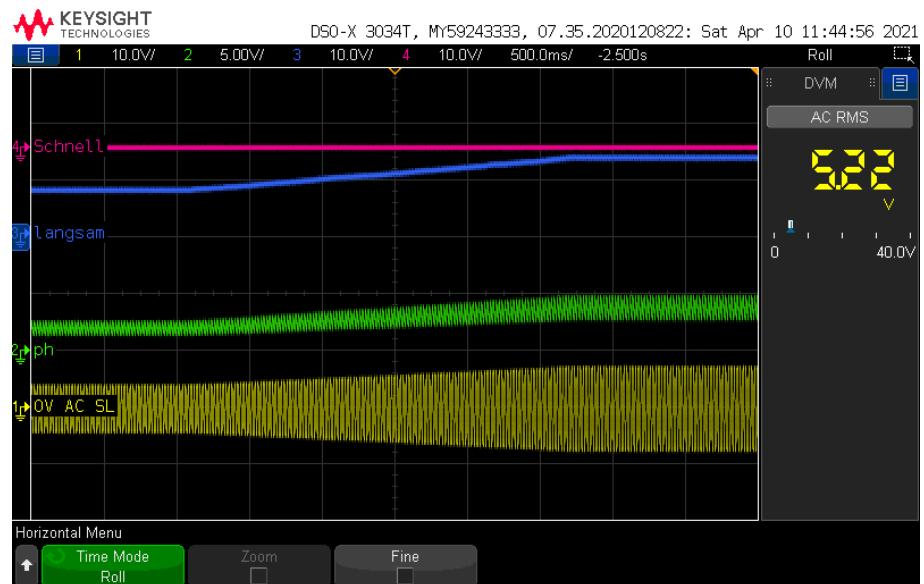


Abbildung 29: suchen

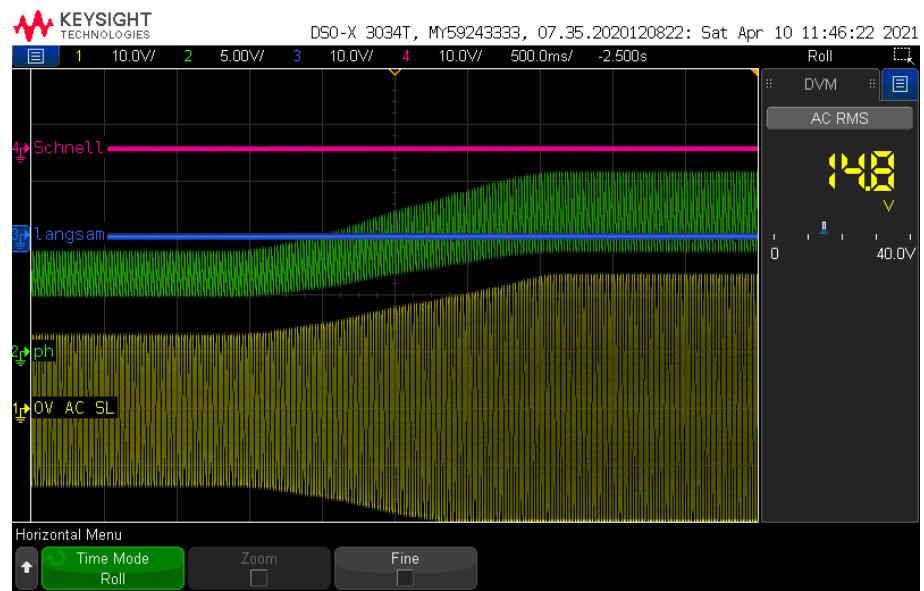


Abbildung 30: schnelllauf

Das digitale Signal steht im gesamten Versorgungsspannungsbereich korrekt an.

### Haltespule

Die Funktion der „Haltespule“ kann durch das Anlegen von ca. 35 V AC getestet werden.

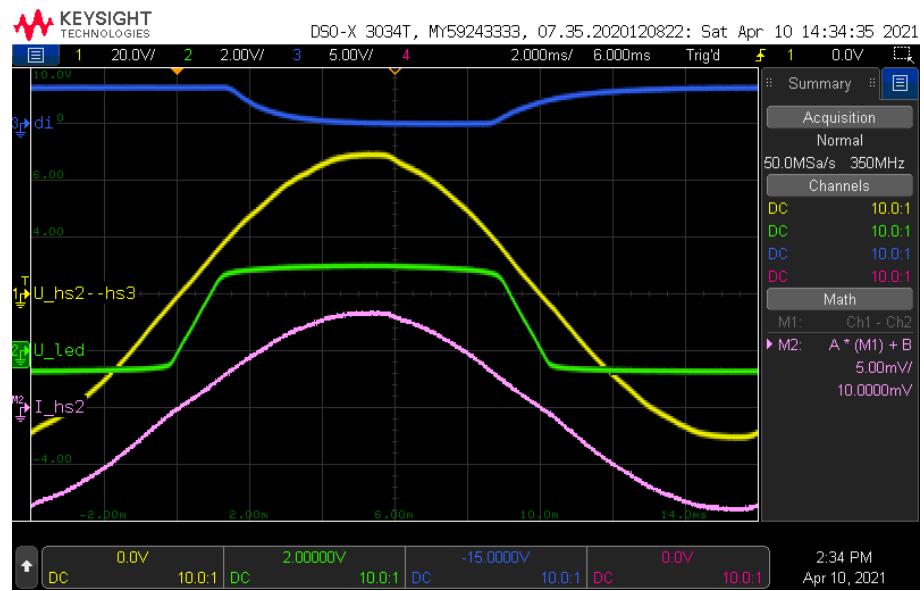


Abbildung 31: Haltespule

Der digitale Eingang muss bei ca. 30 V / 5 mA umschalten.

### Digitale Signale Mikrocontroller

Die Signale am ESP32 können auf zwei Möglichkeiten getestet werden:

- Spannungsquelle (Relais) + Multimeter (Eingänge) anstelle des ESP32 an der Stifteleiste anschließen
- ESP32 + Arduino Testsoftware<sup>2</sup>

Signale nacheinander anlegen, es muss der zugehörige Eingang umschalten bzw. das zugehörige Relais schalten.

### Webradio / Audio Mikrocontroller

Nicht vorab möglich. Nachdem ESP32 programmiert ist kann das Webradio wie in 4. beschrieben benutzt werden, an die Klinkenbuchse kann ein Kopfhörer, PC Lautsperecher o.ä. angeschlossen werden.

## 3.6. Programmierung ESP32

Der ESP32 Controller kann sowohl auserhalb wie auch in der Schaltung programmiert werden. Im letzten Fall ist **penibelst** darauf zu achten dass das Gerätechassis nicht geerdet ist. Grund: Der Schaltungs GND entspricht nicht dem Radio GND (Chassis), siehe Steuerplatine<sup>3</sup>. Werden beide über das USB Programmierkabel verbunden entsteht ein Kurzschluss.

### Möglichkeit 1: Kommandozeile, Python – esptool.py

Funktionierende Pythoninstallation mit *pip* wird vorausgesetzt. Sollte dann mit allen Betriebssystemen etwa gleich funktionieren. Normalerweise erkennt *esptool.py* automatisch die richtige Schnittstelle, falls nicht kann man nachhelfen mit *--port /dev/ttyUSBx* (Linux) oder *--port COMx* (Windows).

Vorcompilierte Binaries sind im Verzeichnis „software/bin“ abgelegt.

- \$ pip install esptool
- ESP Board mit PC USB verbinden
- Verbindung testen und ggf. vorhandene Altlasten entfernen:  
\$ esptool.py erase\_flash
- Bootloader und Firmware programmieren:  
\$ esptool.py -chip esp32 -baud 921600 -before default\_reset -after hard\_reset write\_flash -z -flash\_mode dio -flash\_freq 80m -flash\_size detect 0xe000 boot\_app0.bin 0x1000 bootloader\_dio\_80m.bin 0x10000  
ESP32\_Radio.bin 0x8000 ESP32\_Radio.partitions.bin

---

<sup>2</sup>Verzeichnis „software/test“, aufspielen wie in 3.6 „Arduino“

<sup>3</sup>Seite 4, Spannungsversorgung 5 V

Ein Firmwareupdate kann einfach durch erneutes Aufrufen des Programmierkommandos durchgeführt werden. Die Einstellungen bleiben dann erhalten.

*esptool.py Dokumentation*. Die Kommandozeilenbefehle werden beim Compilieren / Hochladen durch die Arduino Umgebung ausgegeben.

### Möglichkeit 2: Arduino IDE

Ich empfehle die Installation als Portable App (<https://docs.arduino.cc/software/ide-v1/tutorials/PortableIDE>). Diese Installation wird dann nur für dieses eine Projekt verwendet.

- Kann einfach auf anderen Rechner kopiert werden
- Keine Probleme durch Updates der IDE / Bibliotheken / Toolchain.

Getestet mit Arduino V1.8.12 Linux.

- Arduino IDE herunterladen, entpacken
- Ordner *portable* anlegen, wie in Anleitung beschrieben
- Anleitung von *Espressif* folgen. Stable verwenden.
- Sourcecode von der in 1. „Software“ angegebenen URL herunterladen.
- Datei „ESP32-Radio.ino“ laden. Da die Arduino IDE nicht für die Entwicklung verwendet wird ist diese Datei fast leer.
- Board „DOIT ESP32 Devkit V1“ auswählen
- Bibliotheken hinzufügen (PubSubClient, Ch376msc, TFT, SD, ... ohne Garantie auf Vollständigkeit) solange bis alle Abhängigkeiten erfüllt sind
- „Upload“ drücken

### Möglichkeit 3: Eclipse IDE

...das verwende ich. Falls man ernsthaft etwas am Sourcecode ändern will empfehle ich ausdrücklich eine richtige IDE zu verwenden (Eclipse, VS-Code, ...).

Startpunkte:

- <https://duckduckgo.com/?q=eclipse+sloebert+esp32>
- <https://eclipse.baeyens.it>
- <https://www.youtube.com/watch?v=UuYy4YJhToo>

Nach erfolgreicher Installation und Import des Projekts kann wie bei der Arduino IDE „Upload“ gedrückt werden.

## 4. Bedienung

Webradio hinten auf Anschlusstecker aufstecken und mit Schraube festschrauben. Gehäusedeckel aufschrauben. Audiokabel mit TA Buchsen verbinden.

### 4.1. W-Lan Anmeldung

Bei der ersten Inbetriebnahme, oder falls das eingestellte W-Lan nicht erreichbar ist, öffnet der ESP32 ein offenes W-Lan Netz. Darin einloggen, IP 192.168.4.1 im Webbrowser öffnen.

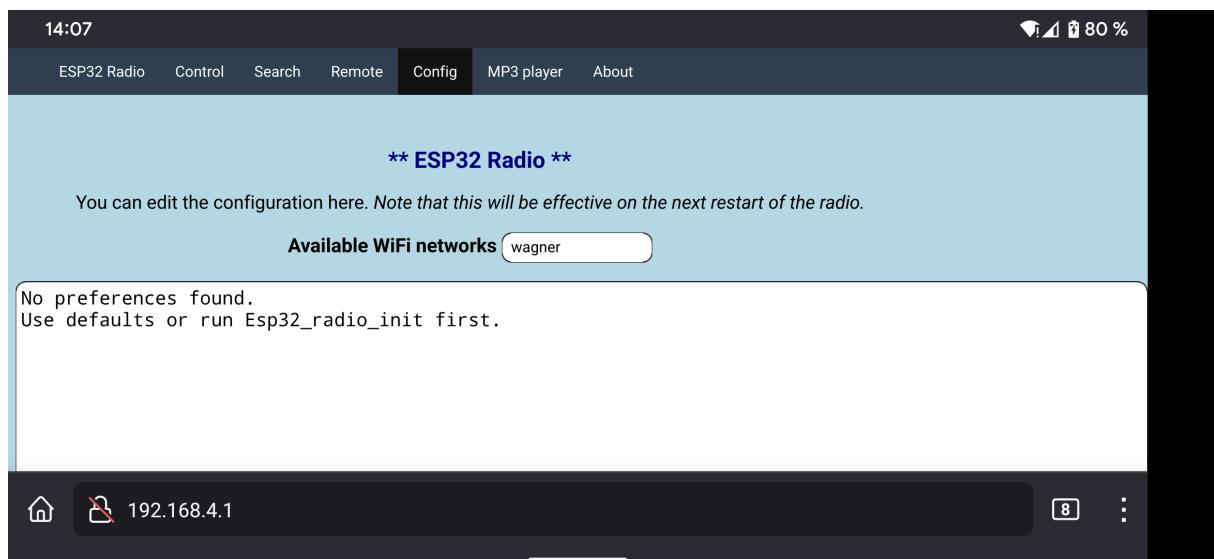


Abbildung 32: Startseite

Bis zum Ende scrollen, Knopf „Default“ drücken.

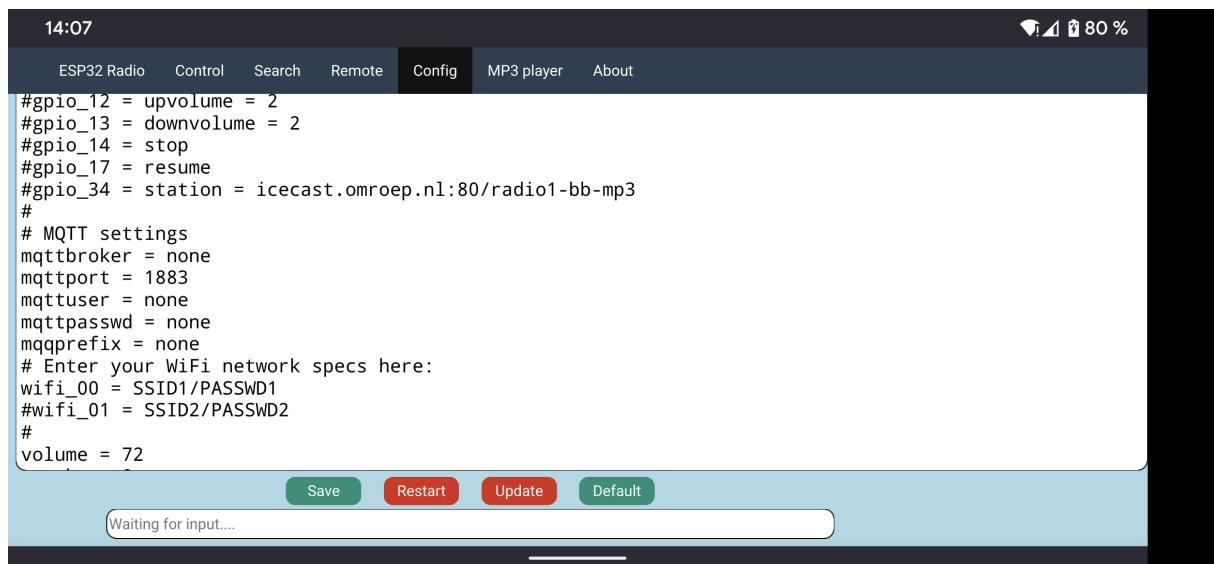


Abbildung 33: Defaults

Feld `wifi_00` W-Lan Namen und Passwort eingeben. „Save“, „Restart“ drücken. Das Passwort wird auf dem ESP32 unverschlüsselt gespeichert!

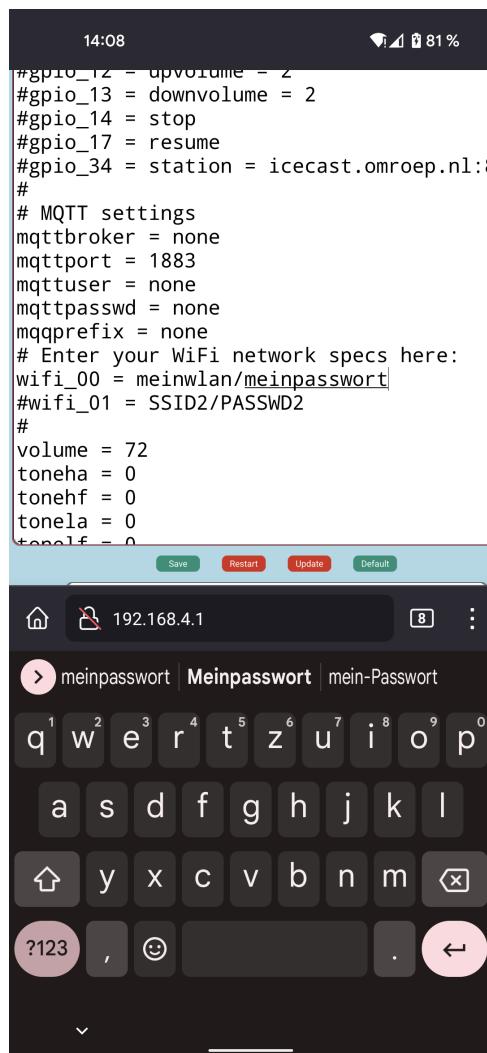


Abbildung 34: Passwort

Danach sollte sich der ESP32 im vorgegebenen W-Lan anmelden. Falls nicht, Terminalprogramm auf der USB seriellen Schnittstelle öffnen und Debugausgaben auswerten. Website aufrufen mit IP des ESP32 oder z.B. <http://esp32-arduino.fritz.box> falls man eine Fritzbox hat. Das W-Lan wird immer geöffnet wenn sich der ESP32 nicht in eines der vorgegebenen Netze einloggen kann.

## 4.2. Webradio

Taste **TA** am Radio drücken.

### Bedienung am Radio

Über den Wippschalter der Automatik können am Webradio die Stationen ausgewählt werden, wie im normalen Radiobetrieb. Die Automatiktaste kann betätigt (→ Automatik Aus) sein.

- Wippschalter ganz links → Station „1“ auswählen
- Wippschalter links tippen → vorherige Station
- Wippschalter rechts tippen → nächste Station
- Wippschalter ganz rechts → **todo** nicht belegt

## Bedienung Webinterface

Nach dem Öffnen der Webseite stehen alle Grundfunktionen zur Verfügung.

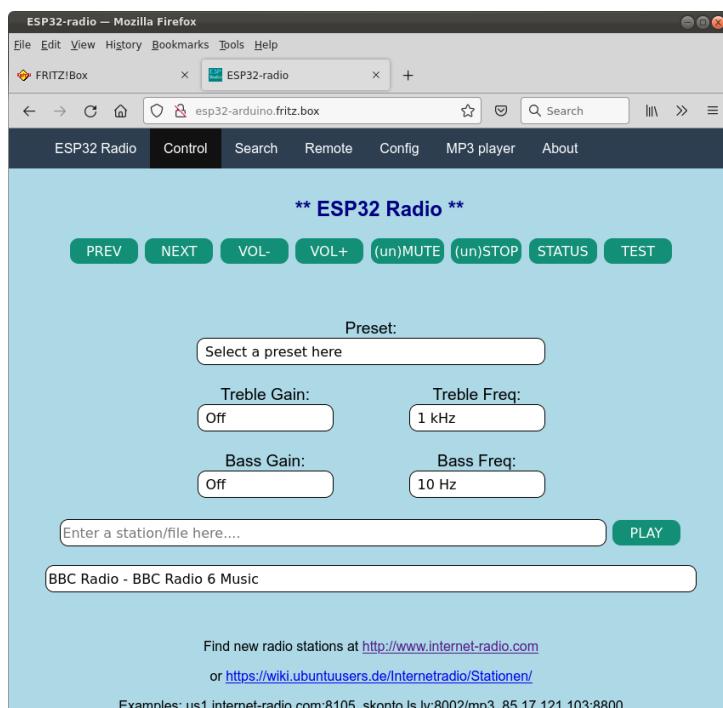


Abbildung 35: Webradio

Im Feld „Enter a station/file here...“ kann die URL einer Radiostation angegeben und mit *Play* gestartet werden. Geht nur ohne Protokollangabe (`http://` der URL weglassen). Es können nur unverschlüsselte Stationen wiedergegeben werden (kein `https://`). Die meisten Stationen unterstützen unverschlüsselte Übertragung, auch wenn es nicht explizit angegeben ist.

Über den Reiter *Search* steht eine nach Musikrichtung sortierte Auswahl an Radiostationen zur Verfügung. Auswählen durch Anklicken, danach wird der Stream gestartet. Nicht alle funktionieren immer.

### 4.3. Fernbedienung

Rundfunkband am Radio auswählen. Webseite Reiter *Remote* auswählen.

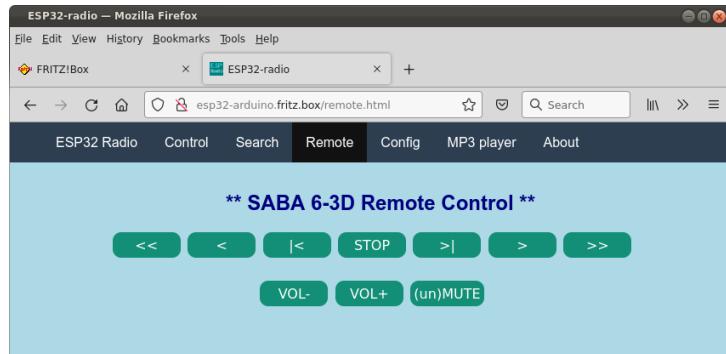


Abbildung 36: Fernbedienung

Bedienung entspricht dem Wippschalter am Gerät. |< und >| starten einen Suchlauf. Falls aktueller Sender nicht verlassen wird die Taste gedrückt halten bis Sender verlassen wurde (so wie am Gerät auch...).

### 4.4. Einstellungen

Webseite Reiter *Config* auswählen.

Es können die folgenden Einstellungen geändert werden:

- Uhrzeit / NTP
- MQTT IoT
- Voreingestellte Radiosender. Diese Liste wird mit dem Wippschalter / dem *Control* Reiter zur Verfügung gestellt
- W-Lan Nutzer / Passwort

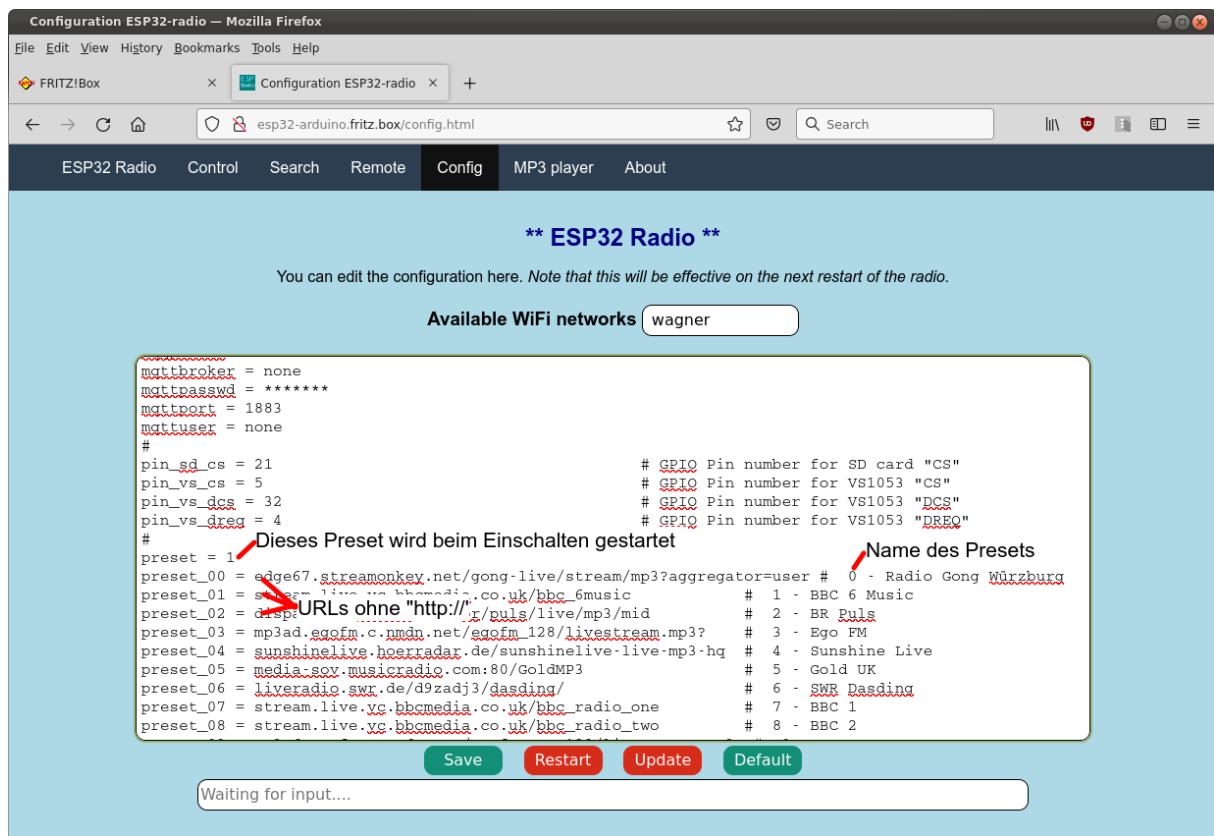


Abbildung 37: Konfiguration

Einstellungen übernehmen mit „Save“, „Restart“.

## 4.5. Weitere Funktionen

Weitere Funktionen sind in der Dokumentation des ESP32-radio Projekts beschrieben.

## 4.6. OTA Firmwareupdate

Webseite Reiter *Config* auswählen. *Update* drücken, siehe Bild 37. Es wird die Datei `http://martinwag.atwebpages.com/ESP32_radio/firmware.bin` geprüft, heruntergeladen und ins Flash geschrieben<sup>4</sup>. Das Kommando wird im Textfeld bestätigt, eine Erfolgsmeldung gibt es nicht. Spätestens nach ein paar Minuten sollte das Update fertig sein, im Reiter *About* sollte eine neuen Builddatum auftauchen (oder auch nicht falls kein Update verfügbar ist).

Die Einstellungen werden beim Update beibehalten.

MD5 Version „Build: 2022-08-08 – 1“: c14b029a3c16b14e4aa896f9e22090a8.

<sup>4</sup>Download per HTTP. Keine Verschlüsselung, keine Authentifizierung, etc. .

## **5. Schaltungsbeschreibung**

Die Schaltung besteht aus folgenden Blöcken:

- Microcontroller ESP32 auf Daughterboard
- Audioschaltung
- Radio steuern per Webradio
- Webradio steuern per Radio
- Spannungsversorgung
- Schnittstellen

In den folgenden Kapiteln wird Design, Funktion und Zusammenwirken beschrieben.

### **5.1. Datenblätter**

Folgende Datenblätter und Application Notes wurden verwendet (u.A):

- ADA4522-1/ADA4522-2/ADA4522-4 Rev. F
- AD8655/AD8656 Rev. E
- VS1053b Datasheet Version: 1.31, 2017-11-17
- VS10XX AppNote Rev 1.25
- vs1033example\_rev07
- Layout Considerations VS10XX AppNote Rev 1.0
- DOIT ESP32 Devkit V1
- ESP32WROOM32D & ESP32WROOM32U Datasheet Version 2.0 2020
- TI LM139, LM239, LM339, LM139A LM239A, LM339A, LM2901, LM2901AV, LM2901V SLCS006U – OCTOBER 1979 – REVISED NOVEMBER 2018

## 5.2. Microcontroller

Der ESP32 wird verwendet da für diesen mit dem ESP32-Radio ein passendes Softwareprojekt bereits zur Verfügung steht. Ebenfalls möglich wäre ein Raspberry Pi o.ä. gewesen, dieser hätte aber erheblich höhere Anforderungen an die Leistung des Netzteils gestellt.

Es wird ein vorgefertigtes Board verwendet.

- Sichere Funktion durch vorgefertigte Platine, inkl. Netzteil, Resetbeschaltung und Programmierschaltung. Leichtes Prüfen da tauschbar.
- ESP32-WROOM-32 Auflötmodul hat Pad auf Unterseite, somit nur umständlich handlötbar.
- Belegt weniger Platz auf Baseboard.
- Programmieren ohne Baseboard möglich.
- Vernachlässigbarer Aufpreis.

## 5.3. Audioschaltung

### Codec

Die Audioschaltung basiert auf einem VLSI VS1053 MP3 Hardware Codec. Wir nutzen nur den Decoder. Der VS1053 wird vom ESP32-Radio Projekt vorgegeben. Schaltung und Layout sind nach Application Note von VLSI.

### Verstärker

Das vom VS1053 ausgegebene Audiosignal wird von einem Audioverstärker verstärkt. Dies ermöglicht ein Anpassen des Audiopegels an die Empfindlichkeit des TA Eingangs.

Der OP ist als *nichtinvertierender Wechselspannungsverstärker mit einfacher Versorgungsspannung* verschaltet. Die Schaltung ist dimensioniert um einen hochohmigen Eingang zu treiben. Der OP muss folgende Bedingungen erfüllen:

- Betrieb an 5 V einfacher Versorgung
- Linear im Audiobereich
- Ausgangsspannung bis ca. +/- 0,5 V an Versorgungsspannungen

Es sind somit praktisch alle für niedrige Versorgungsspannungen optimierten OPs geeignet.

### **Masseführung**

Die Analogschaltung verwendet die allgemeine Massefläche (keine Sternanbindung). Die Schaltung ist nicht mit der Masse des Radios verbunden.

### **Ausgangstrafo**

Um Brummen zu vermeiden muss die Audioschaltung am GND Anschluss TA mit dem Radio verbunden werden. Gleichzeitig muss aber der Versorgungsstrom der Mikrocontrollerschaltung über den Fernbedienungsanschluss fließen. Diese Aufteilung ist nur möglich wenn entweder die Versorgungsseite oder die Audioseite potentialgetrennt werden. Schaltung Rev. A hat einen DC/DC Wandler vorgesehen, dieser hat den Radioempfang gestört. Schaltung Rev. B verwendet daher einen Ausgangstrafo.

### **Stereo**

Die Audioschaltung ist zweikanalig ausgeführt. Bei der Platinenbestückung kann ausgewählt werden ob am Ausgang beide Signale zu einem Monosignal gemischt werden oder Stereo ausgegeben wird. Dies ermöglicht eine einfache Anpassung zur Verwendung an Stereogeräten.

## **5.4. Steuerung Radio**

Der Steuerteil ersetzt 1:1 die kabelgebundene SABA Fernbedienung. Die Schaltung basiert auf der IR Fernbedienung Saba Freiburg 125 von H. Krebs. Schaltung und Bauteilwerte wurde an die 6-3D Serie angepasst.

### **Automatik Steuerung**

Bildet die Schaltung in der Fernbedienung nach. Die Widerstände 1 kOhm sind leistungstechnisch so ausgelegt das sie bei Softwarefehler (alle drei Relais geschaltet) nicht zerstört werden.

### **Haltespule**

In der SABA Fernbedienung entspricht die Haltespule einem Relais mit Selbsthaltung. Wenn ein Sender gefunden ist fließt kein Strom mehr und die Spule fällt ab. Die Schaltung und Software muss dieses Verhalten nachbilden:

- Das Radio muss weiterhin einen Widerstand ca. 6 kOhm „sehen“<sup>5</sup> da die im Radio verbaute Haltespule sonst nicht mehr funktioniert
- Die Software muss Abfall des Stromflusses erkennen („Sender gefunden“) und Suchlaufrelais abschalten.

---

<sup>5</sup>Siehe Schaltschema SABA - Fernbed. Anschl.

Es muss ein Widerstand mind. 0,25 W verwendet werden.

### Lautstärke, Stummschaltung

Im Gegensatz zur Schaltung von H. Krebs verwenden wir Relais. Die zusätzlichen Kontakte werden zur Verriegelung verwendet. Dies verhindert Zerstörung des Radio Netztrafos durch Kurzschluss bei Softwarefehler (Relais laut und leise gleichzeitig geschaltet). Es verhindert auch gleichzeitigen Betrieb von Suchlauf und Stummschaltung.

### Netzschalter

Der Schaltungsteil Netzschalter ist optional. Er ermöglicht ein Ausschalten (und bei USB Versorgung auch Einschalten) des Radios per Webradio. Falls nicht benötigt, Spannung 230 V AC auf N1 oder N2 brücken. Alternativ kann auch die eine Hälfte des Zentrierstiftes aus 3.2 abgeschliffen werden um den Schaltkontakt im Radio dauerhaft geschlossen zu halten. Die zugehörigen Kontaktstifte können dann entfallen.

Der Wechsler im Relais bildet in Verbindung mit dem Netzschalter im Radio eine Wechselschaltung. Um den Zustand auch ohne Versorgungsspannung zu halten wird ein bistabiles Relais verwendet.

**todo** nicht in Software implementiert.

### USB Versorgung

Optional, nur Sinnvoll in Verbindung mit Netzschalter, ermöglicht Standby / Einschalten per Netzwerk. Der Schaltungsteil „Spannungsversorgung 5 V“<sup>6</sup> und der Ausgangstrafo kann entfallen.

### Infrarotfernbedienung

Wird vom ESP32-radio Projekt unterstützt, nicht weiter getestet. Funktion wird in der Doku des ESP32-radio detailliert beschrieben.

## 5.5. Steuerung Webradio

Der Schaltungsteil Steuerung Webradio ermöglicht eine Bedienung des Webradios mit der Suchlaufwippe am Radio. Diese hat fünf Schalterstellungen:

- Nicht betätigt
- Suchlauf jeweils links und rechts.
- Schnelllauf jeweils links und rechts

Abhängig von der Schalterposition wird das Signal auf der Suchlaufleitung verändert:

---

<sup>6</sup>Siehe Schaltplan Seite 3

- Keine Betätigung - DC Vorspannung der Triode EABC80 Rö. 4.
- Betätigung Suchlauf - DC Spannung wird negativ verschoben (Stummschaltung), AC wird aufaddiert. AC Phase abh. von der Laufrichtung.
- Betätigung Schnelllauf - Wie Suchlauf, aber AC Amplitude höher.

Die Schalterposition kann folglich anhand der aufaddierten Wechselspannung ermittelt werden. Der Verlauf wird an der Simulation sichtbar:

- 1 s ... 2 s - links Schnelllauf
- 3 s ... 4 s - links Suchlauf
- 7 s ... 8 s - rechts Schnelllauf
- 9 s ... 10 s - rechts Suchlauf
- restliche Zeit - nicht betätigt.

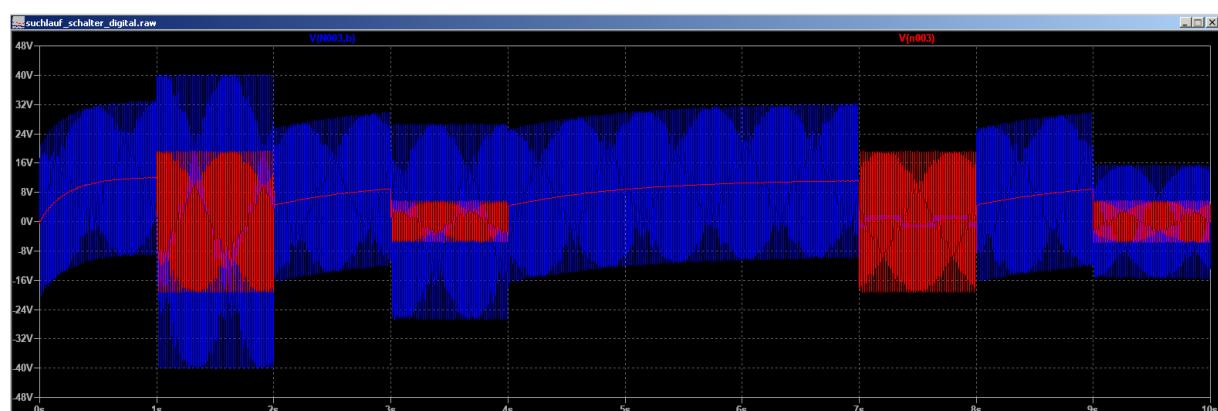


Abbildung 38: Spannung Rot: Suchlaufleitung – AC 0V, Blau: Suchlaufleitung – AC 15V -90°

Messung mit Einweggleichrichtung und Siebung der Wechselspannung. Umsetzung in digitale Signale mit Komparatoren. Die Spannungen beziehen sich auf die Mittelanzapfung des Trafos. Dieser Punkt ist **nicht** mit der Masse des Radios verbunden. Daher ist für die digitalen Signale Potentialtrennung erforderlich.

## 6. Untersuchungen

### 6.1. Probleme

Probleme und Lösungsvorschläge.

## In Software funktioniert xyz nicht

- ESP32 mit PC USB verbinden, Kapitel 3.6 beachten. Der ESP32 verhält sich wie ein USB Seriell Wandler.
- Serielles Terminal verbinden, 115200 8-N-1. Linux z.B. minicom, Windows z.B. Putty oder HyperTerminal
- Funktion starten, Ausgabe auswerten.
- Im Terminal können auch Kommandos eingegeben werden (z.B. *reset*). Der ESP32 selbst macht kein „Echo“, d.H. abhängig von der Terminalkonfiguration sieht man die eigene Eingabe nicht. „Enter“ führt Eingabe aus. Liste der Kommandos ist im Sourcecode in „main.c“, Funktion „analyzeCmd“ enthalten.

## Erkennung TA Eingang und LEDs in den Tasten

Die Glühlampen für die Taster werden nicht mehr hergestellt und sind auch anderweitig nicht auftriebbar. Werden diese durch LED + Gleichrichter ersetzt funktioniert die Schaltung zur Erkennung des TA nicht mehr. Diese kann wie folgt angepasst werden

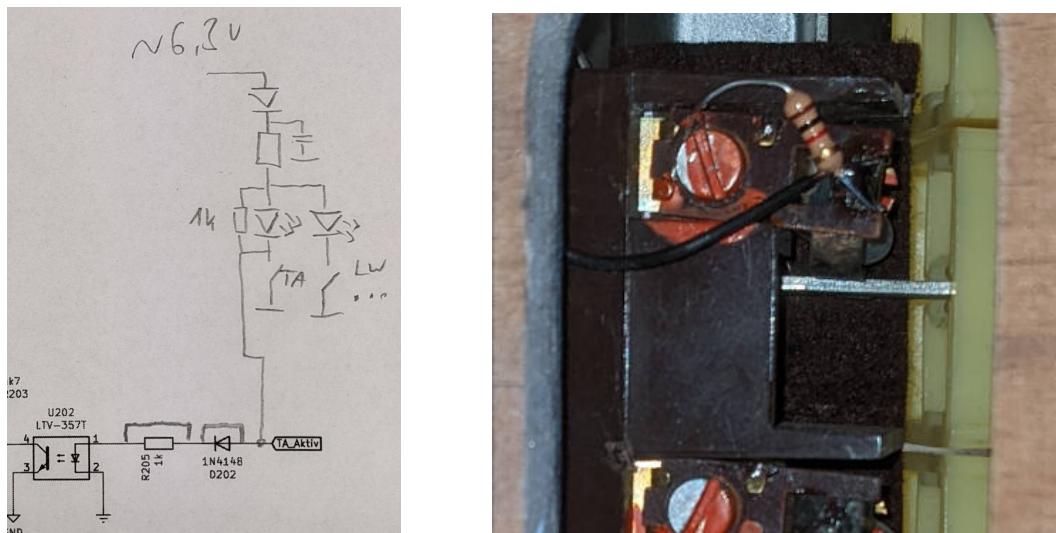


Abbildung 39: Anpassung LEDs

## Webinterface reagiert langsam

Webseite aufrufen mit der IP Adresse statt URL. Scheinbar bringen div. Router da etwas durcheinander wenn mehrere ESP32 am Netz sind.

## Weitere

**todo** weitere ...

## 6.2. Ausgangsübertrager

Wir haben einen relativ niederohmigen Ausgang (Opamp) und eine sehr hochohmige Last (Radio Phonoeingang).

Es wurden verschiedene Ausgangsübertrager untersucht um ein geeignetes Bauteil zu finden.

Von links nach rechts:

- Ground Loop Isolator Sinus Live GL-205, 2 x RCA (ca. 15 Euro)
- Telefonübertrager 600:600 Ohm EI14 (Ebay China ca. 4 Euro / 10 Stk.)
- CMC 47 mH Talema CAF 0,5-47 (ca. 3 Euro)
- NFU 1-1 EI19 50 Hz ... 10 kHz, 34 mH (ca. 3 Euro)
- FeinTech ATG00101 Audio Masse-Trennglied, Klinke 3,5 mm (ca. 15 Euro)

Wir brauchen ca. 100 Hz ... 15 kHz.

Versuchsaufbau nach *Vorschlag Keysight, how to create bode plots on an oscilloscope*.

Sweep 0 ... 20 kHz, 1 V<sub>SS</sub>, 50 Ohm Ausgangswiderstand Generator, 100 kOhm Lastwiderstand Sekundärseite. Scheinbar hat der Generator hier einen Bug, es kommen 2 V<sub>SS</sub> raus?



Abbildung 40: getestete Übertrager

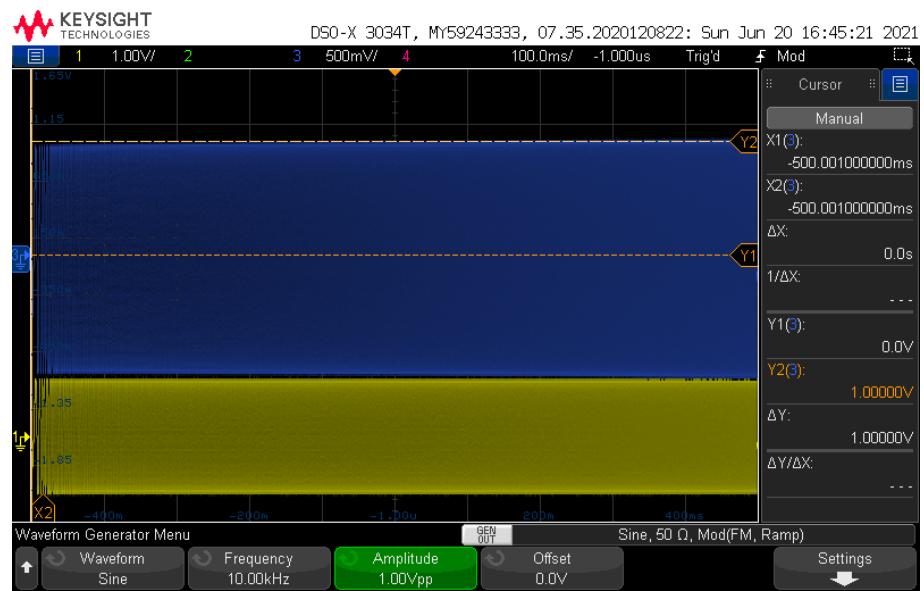


Abbildung 41: GL-205

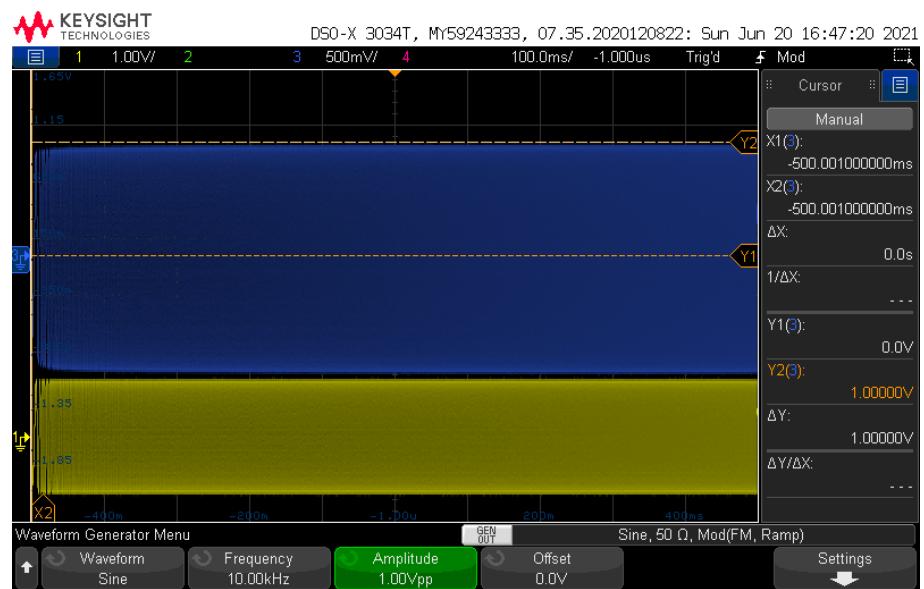


Abbildung 42: China 600:600 Ohm

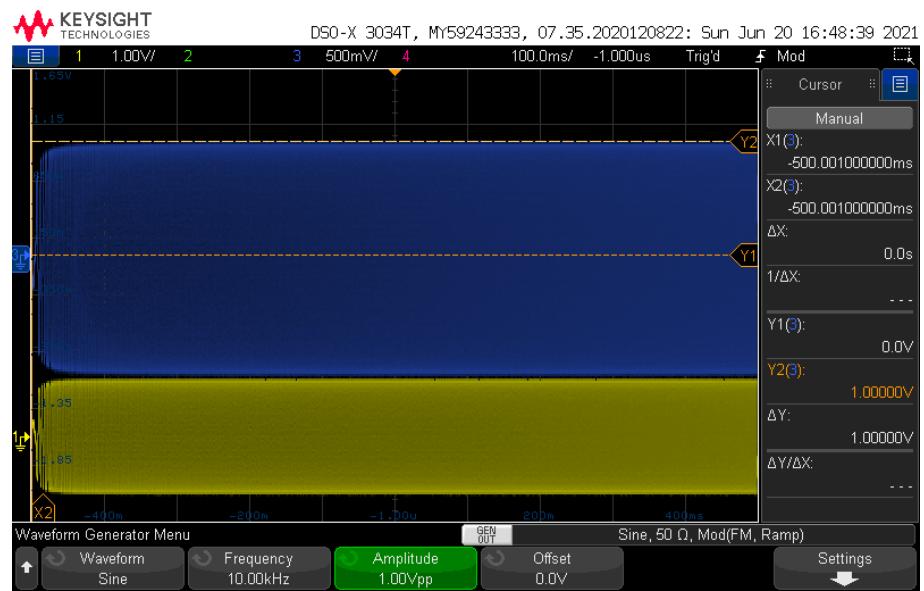


Abbildung 43: CMC 47 mH

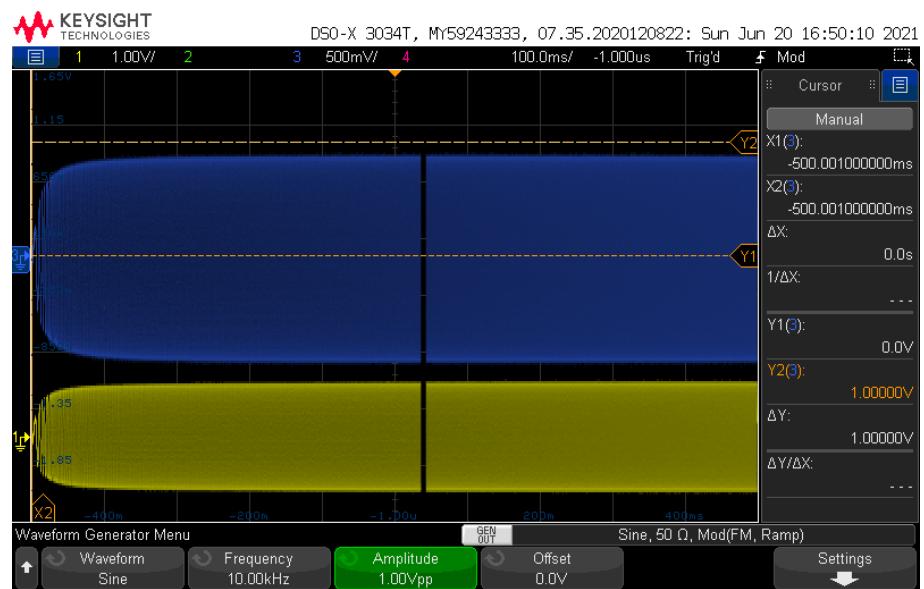


Abbildung 44: NFU 1-1

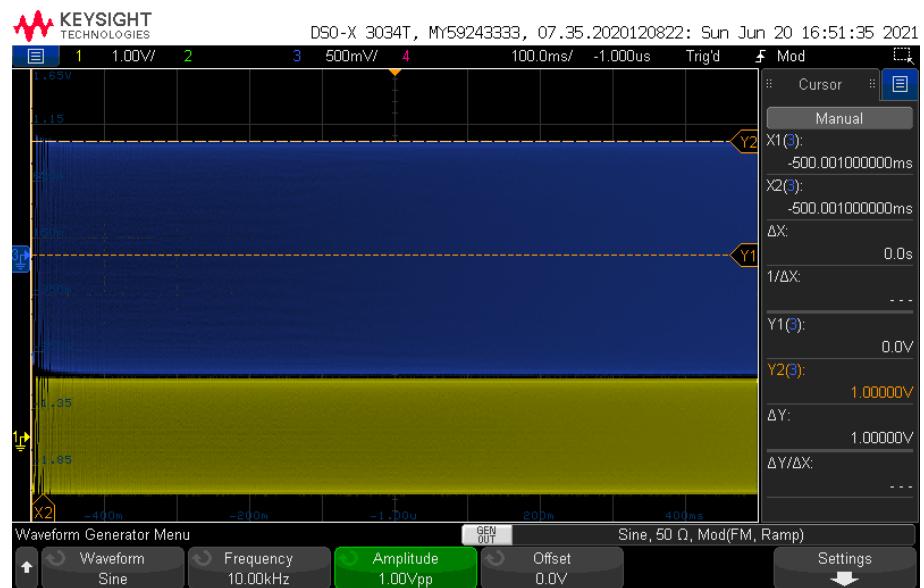


Abbildung 45: ATG00101

Im niedrigen Frequenzbereich sind Unterschiede sichtbar. Daher nochmal Detailmessung 0 ... 2 kHz.

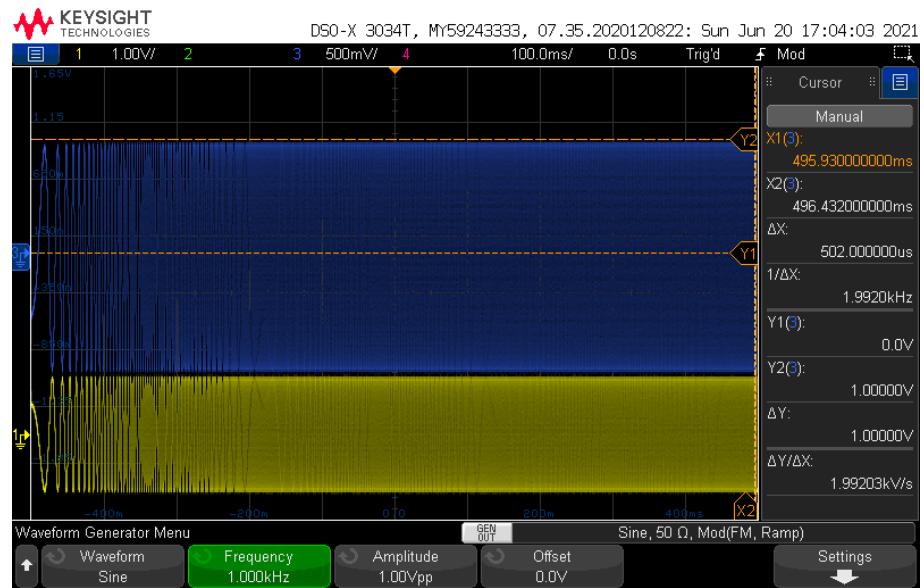


Abbildung 46: GL-205

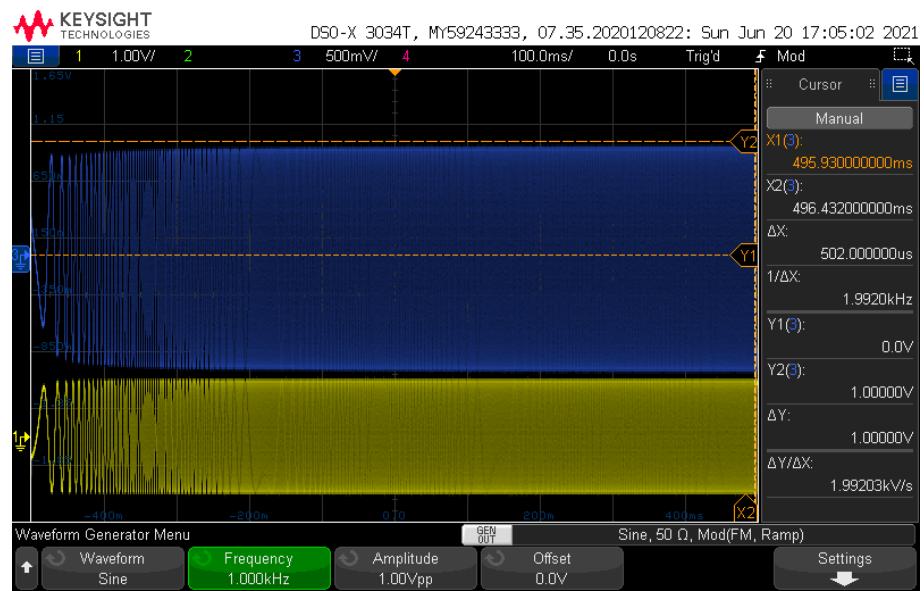


Abbildung 47: China 600:600 Ohm

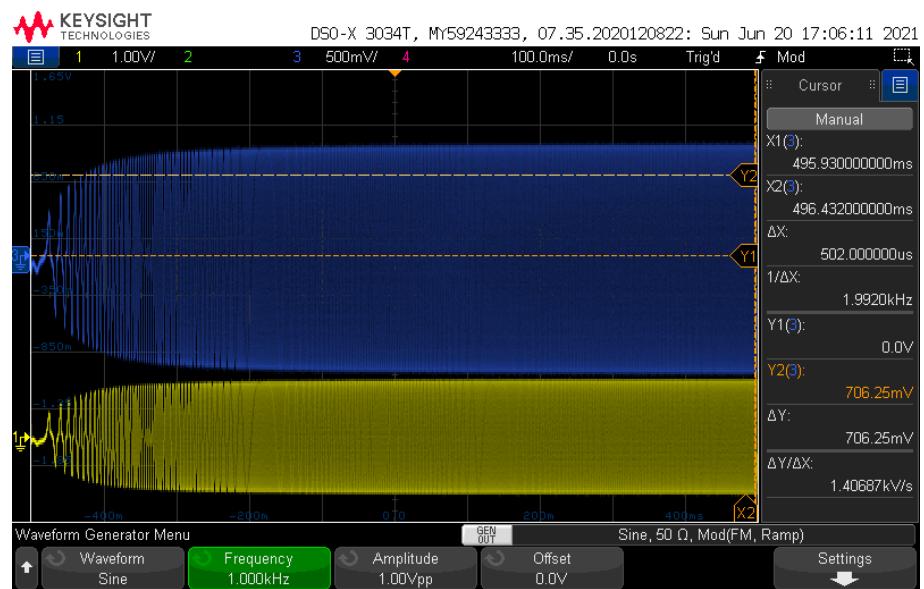


Abbildung 48: CMC 47 mH

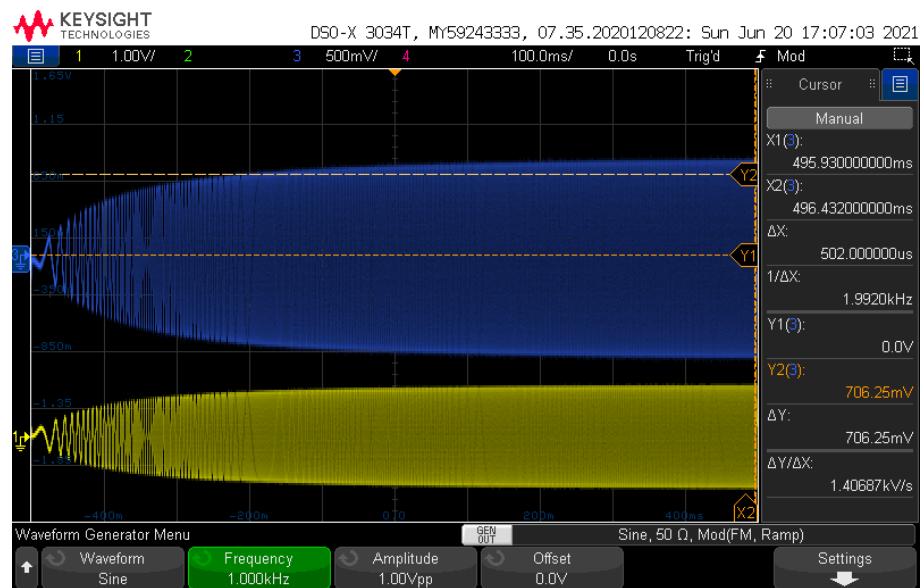


Abbildung 49: NFU 1-1

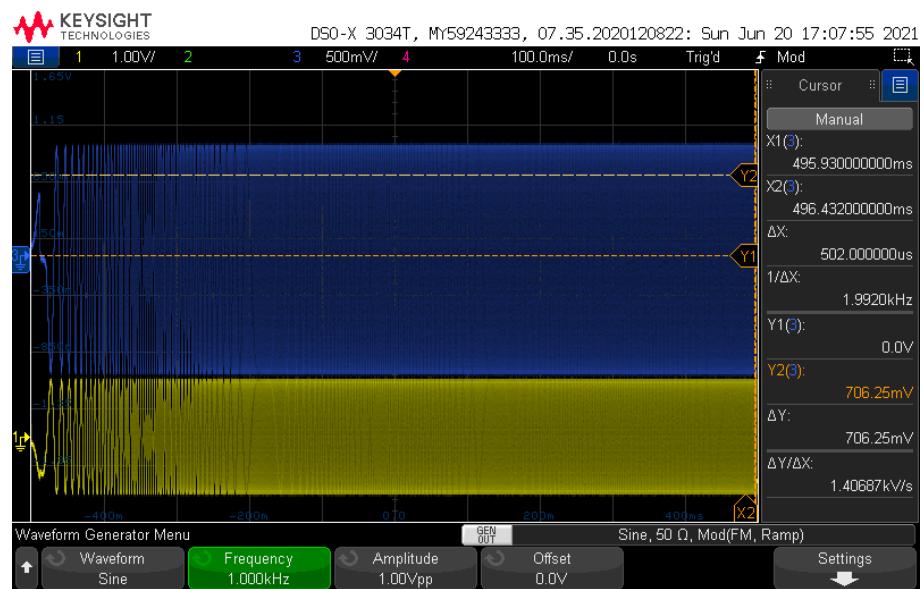


Abbildung 50: ATG00101

### Ergebnis:

- Die beiden kommerziell erhältlichen Massetrenner zeigen keinen nennenswerten Unterschied zwischen dem Spannungspegel an Ein- und Ausgang.
- Bei niedrigen Ausgangsfrequenzen wird ggf.  $X_L$  zu niedrig? → Stromfluss.
- Die CMC funktioniert überraschend gut – in diesem Testaufbau sogar besser als der Audioübertrager.

- Der 600:600 Ohm Telefonübertrager ist mehr als ausreichend gut, billig, leicht zu bekommen und klein.

### 6.3. W-Lan Antenne

Es kann eine 2,4 GHz W-Lan Antenne angeschlossen werden<sup>7</sup>.

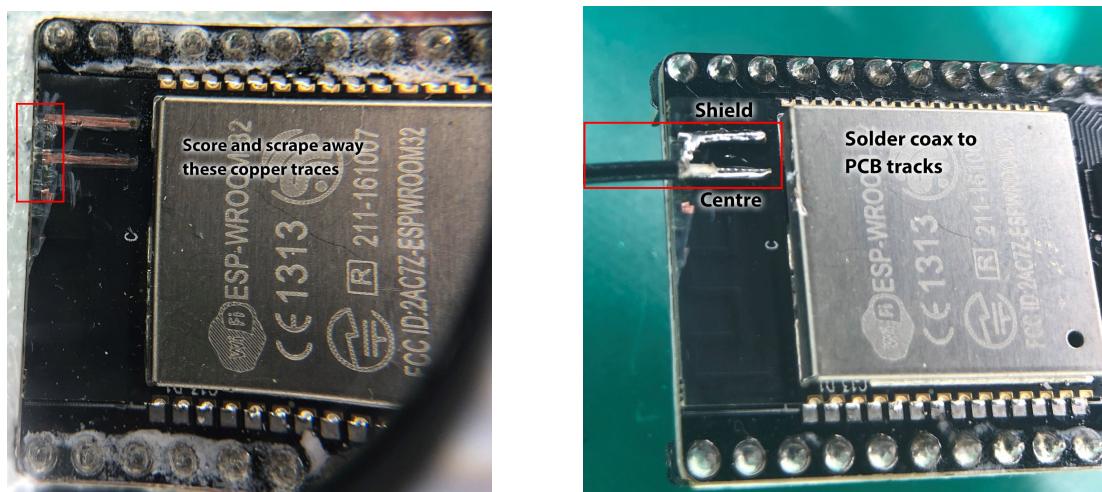


Abbildung 51: Nach Anleitung

Anstelle eines Kabels kann auch eine Buchse montiert werden.

---

<sup>7</sup><https://community.home-assistant.io/t/how-to-add-an-external-antenna-to-an-esp-board/131601>

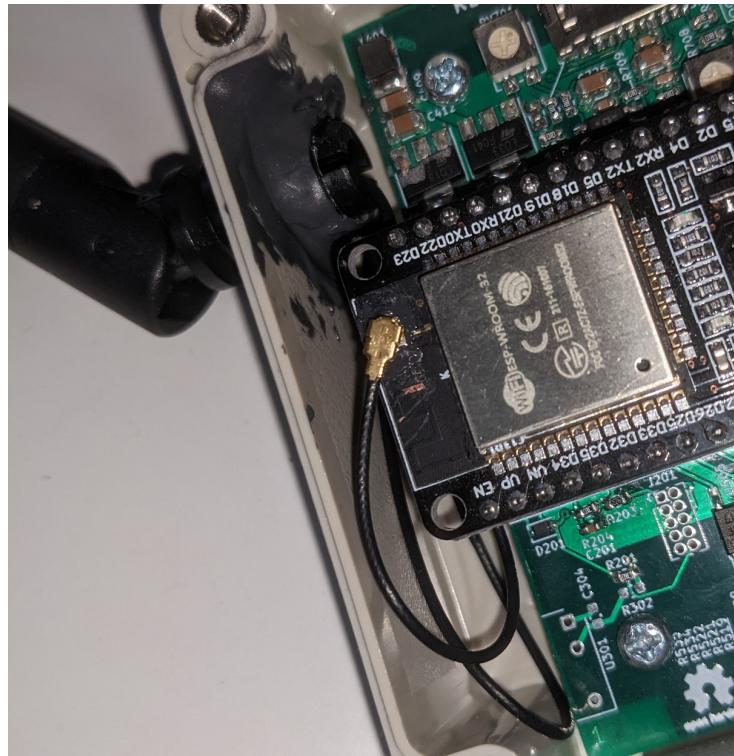


Abbildung 52: Mit Buchse / Stecker

Im Versuch wurde das Signal ca. 6 ... 10 dB stärker.

## A. Ausgabe md5sum Webserver

Version „Build: 2022-08-08 – 1“

```
martin@mint ~/workspace/esp32_radio/ESP32-Radio/Release $ wget http://martinwag.
atwebpages.com/ESP32_radio/firmware.bin
--2022-08-08 16:10:38-- http://martinwag.atwebpages.com/ESP32_radio/firmware.bin
Resolving martinwag.atwebpages.com (martinwag.atwebpages.com) ... 185.176.43.53
Connecting to martinwag.atwebpages.com (martinwag.atwebpages.com)
| 185.176.43.53|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 937040 (915K) [application/octet-stream]
Saving to: firmware.bin.1

firmware.bin.1
100%[=====] 915,08K
3,88MB/s   in 0,2s

2022-08-08 16:10:38 (3,88 MB/s) - firmware.bin.1 saved [937040/937040]

martin@mint ~/workspace/esp32_radio/ESP32-Radio/Release $ md5sum firmware.bin*
c14b029a3c16b14e4aa896f9e22090a8  firmware.bin
```

## B. Ausgabe esptool.py

```
$ esptool.py erase_flash
esptool.py v4.1
Found 2 serial ports
Serial port /dev/ttyUSB0
Connecting.....
Detecting chip type... Unsupported detection protocol, switching and trying again...
Connecting.....
Detecting chip type... ESP32
Chip is ESP32-D0WDQ6 (revision 1)
Features: WiFi, BT, Dual Core, 240MHz, VRef calibration in efuse, Coding Scheme None
Crystal is 40MHz
MAC: ac:67:b2:36:b3:b4
Uploading stub...
Running stub...
Stub running...
Erasing flash (this may take a while)...
Chip erase completed successfully in 2.3s
Hard resetting via RTS pin...

$ esptool.py --chip esp32 --baud 921600 --before default_reset --after hard_reset
  write_flash -z --flash_mode dio --flash_freq 80m --flash_size detect 0xe000 /opt
  /eclipse/arduinoPlugin/packages/esp32/hardware/esp32/1.0.6/tools/partitions/
  boot_app0.bin 0x1000 /opt/eclipse/arduinoPlugin/packages/esp32/hardware/esp32
  /1.0.6/tools/sdk/bin/bootloader_dio_80m.bin 0x10000 /home/martin/workspace/
  esp32_radio/ESP32-Radio/Release/ESP32_Radio.bin 0x8000 /home/martin/workspace/
  esp32_radio/ESP32-Radio/Release/ESP32_Radio.partitions.bin
esptool.py v4.1
Found 2 serial ports
Serial port /dev/ttyUSB0
Connecting.....
Chip is ESP32-D0WDQ6 (revision 1)
Features: WiFi, BT, Dual Core, 240MHz, VRef calibration in efuse, Coding Scheme None
Crystal is 40MHz
MAC: ac:67:b2:36:b3:b4
Uploading stub...
Running stub...
Stub running...
Changing baud rate to 921600
Changed.
Configuring flash size...
Auto-detected Flash size: 4MB
Flash will be erased from 0x0000e000 to 0x0000ffff...
Flash will be erased from 0x00001000 to 0x00005fff...
Flash will be erased from 0x00010000 to 0x000f4fff...
Flash will be erased from 0x00008000 to 0x00008fff...
Compressed 8192 bytes to 47...
```

```

Wrote 8192 bytes (47 compressed) at 0x0000e000 in 0.1 seconds (effective 1005.8 kbit
/s)...
Hash of data verified.
Compressed 17120 bytes to 11164...
Wrote 17120 bytes (11164 compressed) at 0x00001000 in 0.3 seconds (effective 460.0
kbit/s)...
Hash of data verified.
Compressed 936896 bytes to 530440...
Wrote 936896 bytes (530440 compressed) at 0x00010000 in 7.1 seconds (effective
1051.7 kbit/s)...
Hash of data verified.
Compressed 3072 bytes to 128...
Wrote 3072 bytes (128 compressed) at 0x00008000 in 0.1 seconds (effective 446.9 kbit
/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...

```

---

## C. Ausgabe serielle Schnittstelle beim Boot

Boot, Webradio aktiv (TA Modus), keine SD Karte.

```

D: Starting ESP32-radio running on CPU 1 at 240 MHz. Version Mon, 28 Jun 2021
    12:40:00 GMT. Free memory 273896
D: Display type is DUMMYTFT
D: Partition nvs found, 20480 bytes
D: Read 36 keys from NVS
D: pin_ir set to -1
D: pin_enc_clk set to -1
D: pin_enc_dt set to -1
D: pin_enc_sw set to -1
D: pin_tft_cs set to -1
D: pin_tft_dc set to -1
D: pin_tft_scl set to -1
D: pin_tft_sda set to -1
D: pin_tft_bl set to -1
D: pin_tft_blx set to -1
D: pin_sd_cs set to 21
D: pin_ch376_cs set to -1
D: pin_ch376_int set to -1
D: pin_vs_cs set to 5
D: pin_vs_dcs set to 32
D: pin_vs_dreq set to 4
D: pin_shutdown set to -1
D: pin_shutdownx set to -1
D: pin_spi_sck set to 18
D: pin_spi_miso set to 19
D: pin_spi_mosi set to 23
D: pin_saba_power1 set to -1
D: pin_saba_power2 set to -1
D: pin_saba_power_fb set to -1
D: pin_saba_vol_up set to 15

```

```

D: pin_saba_vol_down set to 2
D: pin_saba_vol_mute set to 16
D: pin_saba_move_left set to 13
D: pin_saba_move_right set to 14
D: pin_saba_move_fast set to 12
D: pin_saba_move_dir set to 27
D: pin_saba_move_is_fast set to 34
D: pin_saba_move_is_slow set to 33
D: pin_saba_search_hold set to 36
D: pin_saba_pickup_active set to 39
D: GPIO0 is HIGH
D: GPIO2 is LOW, probably no PULL-UP
D: GPIO4 is HIGH
D: GPIO5 is HIGH
D: GPIO12 is LOW, probably no PULL-UP
D: GPIO13 is LOW, probably no PULL-UP
D: GPIO14 is LOW, probably no PULL-UP
D: GPIO15 is LOW, probably no PULL-UP
D: GPIO16 is LOW, probably no PULL-UP
D: GPIO17 is HIGH
D: GPIO18 is LOW, probably no PULL-UP
D: GPIO19 is HIGH
D: GPIO21 is HIGH
D: GPIO22 is HIGH
D: GPIO23 is HIGH
D: GPIO25 is LOW, probably no PULL-UP
D: GPIO26 is LOW, probably no PULL-UP
D: GPIO27 is HIGH
D: GPIO32 is HIGH
D: GPIO33 is HIGH
D: GPIO34 is HIGH
D: GPIO35 is LOW, probably no PULL-UP
D: GPIO36 is HIGH
D: GPIO39 is HIGH
[E][sd_diskio.cpp:123] sdSelectCard(): Select Failed
[E][sd_diskio.cpp:775] sdcard_mount(): f_mount failed: (3) The physical drive cannot
    work
[E][sd_diskio.cpp:123] sdSelectCard(): Select Failed
D: SD Card Mount Failed!
D: Create list with acceptable WiFi networks
D: Added wlan-meins to list of networks
D: End adding networks
D: Scan Networks
D: Scan completed
D: Number of available networks: 2
D: 1 - wlan-meins           Signal: -59 dBm, Encryption WPA2_PSK, Acceptable
D: 2 - WLAN-xxxxxx          Signal: -93 dBm, Encryption WPA2_PSK,
D: End of list
D: Command: clk_dst with parameter 1
D: Command: clk_offset with parameter 1
D: Command: clk_server with parameter pool.ntp.org
D: Command: lstmods with parameter Sun, 19 Jun 2022 13:10:24 GMT
D: Command: mqqprefix with parameter none
D: Command: mqttbroker with parameter none

```

```

D: Command: mqttpasswd with parameter ******
D: Command: mqttport with parameter 1883
D: Command: mqttuser with parameter none
D: Command: pin_sd_cs with parameter 21
D: Command: pin_vs_cs with parameter 5
D: Command: pin_vs_dcs with parameter 32
D: Command: pin_vs_dreq with parameter 4
D: Command: preset with parameter 9
D: Command: preset_00 with parameter edge67.streammonkey.net/gong-live/stream/mp3?
    aggregator=user
D: Command: preset_01 with parameter stream.live.vc.bbcmedia.co.uk/bbc_6music
D: Command: preset_02 with parameter dispatcher.rndfnk.com/br/puls/live/mp3/mid
D: Command: preset_03 with parameter mp3ad.egofm.c.nmdn.net/egofm_128/livestream.mp3
    ?
D: Command: preset_04 with parameter sunshinelive.hoerradar.de/sunshinelive-live-mp3
    -hq
D: Command: preset_05 with parameter media-sov.musicradio.com:80/GoldMP3
D: Command: preset_06 with parameter liveradio.swr.de/d9zadj3/dasding/
D: Command: preset_07 with parameter stream.live.vc.bbcmedia.co.uk/bbc_radio_one
D: Command: preset_08 with parameter stream.live.vc.bbcmedia.co.uk/bbc_radio_two
D: Command: preset_09 with parameter mp3ad.egofm.c.nmdn.net/egofmpure_128/livestream
    .mp3?
D: Command: preset_10 with parameter sunshinelive.hoerradar.de/sunshinelive-house-
    mp3-hq
D: Command: preset_11 with parameter sunshinelive.hoerradar.de/sunshinelive-classics
    -mp3-hq
D: Command: preset_12 with parameter airspectrum.cdnstream1.com:8114/1648_128
D: Command: preset_13 with parameter airspectrum.cdnstream1.com:8000/1261_192
D: Command: preset_14 with parameter icecast.omroep.nl:80/radiol-bb-mp3
D: Command: preset_15 with parameter stream.srg-ssr.ch/drsvirus/mp3_128.m3u
D: Command: toneha with parameter 0
D: Command: tonehf with parameter 0
D: Command: tonela with parameter 0
D: Command: tonelf with parameter 0
D: Command: volume with parameter 100
D: Slow SPI, Testing VS1053 read/write registers...
D: Fast SPI, Testing VS1053 read/write registers again...
D: endFillByte is 0
D: Connect to WiFi
D: Try WiFi wlan-meins
D: Connected to wlan-meins
D: IP = 192.168.1.86
D: Start server for commands
D: Network found. Starting mqtt and OTA
D: MDNS responder started
D: Rotary encoder is disabled (-1/-1/-1)
D: STOP requested
D: New preset/file requested (9/0) from mp3ad.egofm.c.nmdn.net/egofmpure_128/
    livestream.mp3?
D: New station request
D: Connect to new host mp3ad.egofm.c.nmdn.net/egofmpure_128/livestream.mp3?
D: Connect to mp3ad.egofm.c.nmdn.net on port 80, extension /egofmpure_128/livestream
    .mp3?
D: Song stopped correctly after 0 msec

```

```

D: Connected to server
D: switch to pickup (direct)
D: station active
D: Switch to HEADER
D: Headerline: Server: nginx
D: Headerline: Content-Length: 0
D: Headerline: Connection: close
D: Headerline: Cache-Control: no-cache, no-store, must-revalidate
D: Headerline: Expires: Mon, 26 Jul 1997 05:00:00 GMT
D: Headerline: Access-Control-Allow-Origin: *
D: Headerline: Access-Control-Allow-Headers: Origin, Accept, X-Requested-With,
    Content-Type
D: l-Allow-Headers: Origin, Accept, X-Requested-With, Content-Type seen.
D: Headerline: Access-Control-Allow-Methods: GET, OPTIONS, HEAD
D: Headerline: Location: http://egofm--di--nacs-ais-lgc--0c--cdn.cast.addradio.de/
    egofm/pure/mp3/high/stream.mp3?_art=dj0yJmlwPTkzLjIzMy4xNi4yNDkmaWQ9aW
D: Switch to DATA, bitrate is 0, metaint is 0
D: Duration mp3loop 52
D: New station request
D: Connect to new host egofm--di--nacs-ais-lgc--0c--cdn.cast.addradio.de/egofm/pure/
    mp3/high/stream.mp3?_art=dj0yJmlwPTkzLjIzMy4xNi4yNDkmaWQ9aWNzY3hsLWN
D: Connect to egofm--di--nacs-ais-lgc--0c--cdn.cast.addradio.de on port 80,
    extension /egofm/pure/mp3/high/stream.mp3?_art=dj0yJmlwPTkzLjIzMy4xNi4yNDkma
D: Connected to server
D: Command: resume with parameter 0
D: Command accepted
D: Switch to HEADER
D: Headerline: Cache-Control: no-cache
D: Headerline: Pragma: no-cache
D: Headerline: Expires: Mon, 26 Jul 1997 05:00:00 GMT
D: Headerline: icy-br: 128
D: Headerline: Instance-id: 93bd887a54781410ccf2dd678482d06d
D: Headerline: Server: nacs-ais-lgc_fra-01-eco_edge_765330e391dc9d6ac20249d04f9ce671
    9.0.6
D: Headerline: icy-genre: egoFM - pure
D: Headerline: icy-metaint: 16000
D: Headerline: Content-Type: audio/mpeg
D: audio/mpeg seen.
D: Headerline: icy-name: egoFM - pure
D: Headerline: Access-Control-Allow-Origin: *
D: Headerline: Connection: close
D: Headerline: icy-url:
D: Headerline: icy-audio-info: ice-samplerate=44100;ice-bitrate=128;ice-channels=2
D: Headerline: icy-pub: 0
D: Headerline: icy-description: egoFM - pure
D: Headerline: X-Loudness: -10.097031
D: Headerline: Set-Cookie: AISSessionId=62509951
    cb194777_6196743_KpUkaTto__0000001qoJk; Path=/; Domain=egofm--di--nacs-ais-lgc
    --0c--cdn.cast.addradio.de
D: Switch to DATA, bitrate is 128, metaint is 16000
D: Duration mp3loop 82
D: New station request
D: Stopping client

```

```
D: Connect to new host egofm--di--nacs-ais-lgc--0c--cdn.cast.addradio.de/egofm/pure/
  mp3/high/stream.mp3?_art=dj0yJmlwPTkzLjIzMy4xNi4yNDkmaWQ9aWNzY3hsLWN
D: Connect to egofm--di--nacs-ais-lgc--0c--cdn.cast.addradio.de on port 80,
  extension /egofm/pure/mp3/high/stream.mp3?_art=dj0yJmlwPTkzLjIzMy4xNi4yNDkma
D: Connected to server
D: Switch to HEADER
D: Headerline: Cache-Control: no-cache
D: Headerline: Pragma: no-cache
D: Headerline: Expires: Mon, 26 Jul 1997 05:00:00 GMT
D: Headerline: icy-br: 128
D: Headerline: Instance-id: 93bd887a54781410ccf2dd678482d06d
D: Headerline: Server: nacs-ais-lgc_fra-01-eco_edge_765330e391dc9d6ac20249d04f9ce671
  9.0.6
D: Headerline: icy-genre: egoFM - pure
D: Headerline: icy-metaint: 16000
D: Headerline: Content-Type: audio/mpeg
D: audio/mpeg seen.
D: Headerline: icy-name: egoFM - pure
D: Headerline: Access-Control-Allow-Origin: *
D: Headerline: Connection: close
D: Headerline: icy-url:
D: Headerline: icy-audio-info: ice-samplerate=44100;ice-bitrate=128;ice-channels=2
D: Headerline: icy-pub: 0
D: Headerline: icy-description: egoFM - pure
D: Headerline: X-Loudness: -10.097031
D: Headerline: Set-Cookie: AISSessionId=62509951
  cb194777_6196744_moKPC10f__0000001qoJm; Path=/; Domain=egofm--di--nacs-ais-lgc
  --0c--cdn.cast.addradio.de
D: Switch to DATA, bitrate is 128, metaint is 16000
D: Metadata block 112 bytes
D: Streamtitle found, 107 bytes
D: StreamTitle='';StreamUrl='';adw_ad='true';durationMilliseconds='30537';adId
  ='7567';insertionType='preroll';
```

---