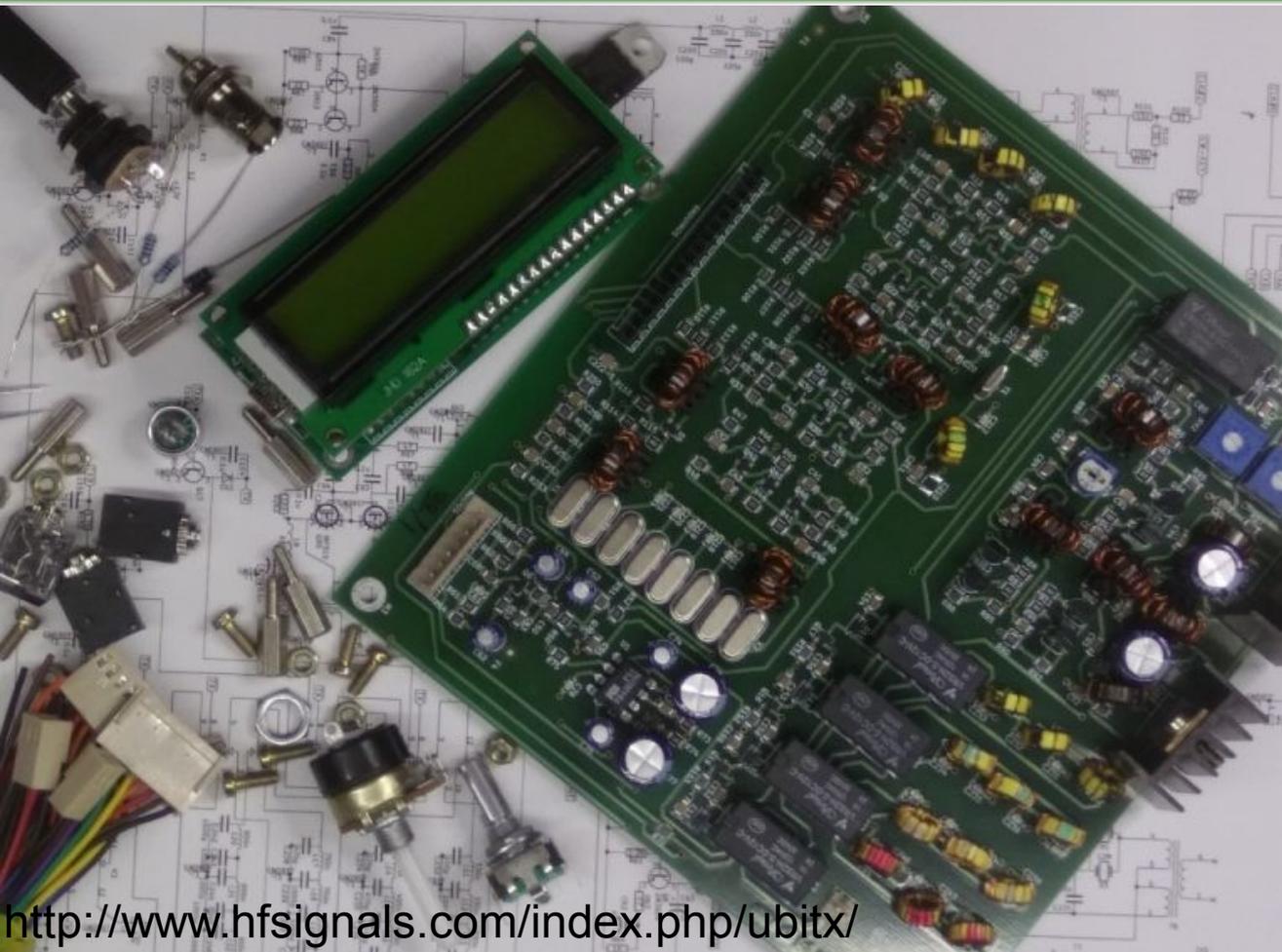


- uBITX -

# Aufbau, Erfahrungen und erforderliche Modifikationen



## DARC OV G25

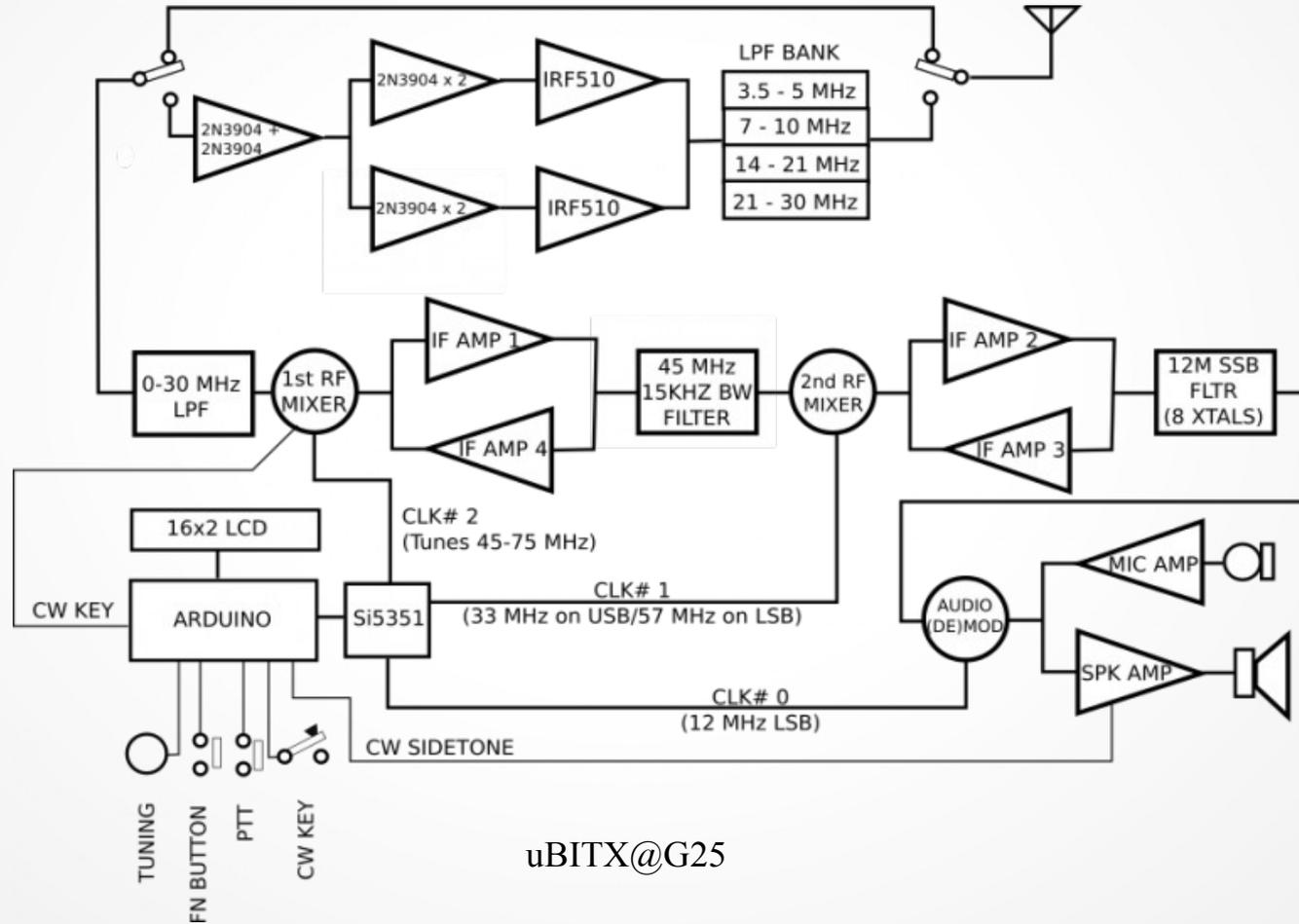
Königswinter, 15.02.2018

Erhard Härtel, DF3FY

# - uBITX - Inhalt

- Schaltung und Frequenzaufbereitung
- Befunde und Maßnahmen
- Modifikation und Abgleich der Endstufe
- ALC-Nachrüstung
- Einstellung des NF-Teils
- Nachrüstung eines Sprachprozessors und CW-Filters
- Weitere notwendige Verbesserungen
- Erreichte technische Daten nach durchgeführten Modifikationen
- Update der Firmware inkl. der dazu erforderlichen Software-Umgebung
- Einbau in ein HF-taugliches Gehäuse

# - uBITX - Blockschaltbild

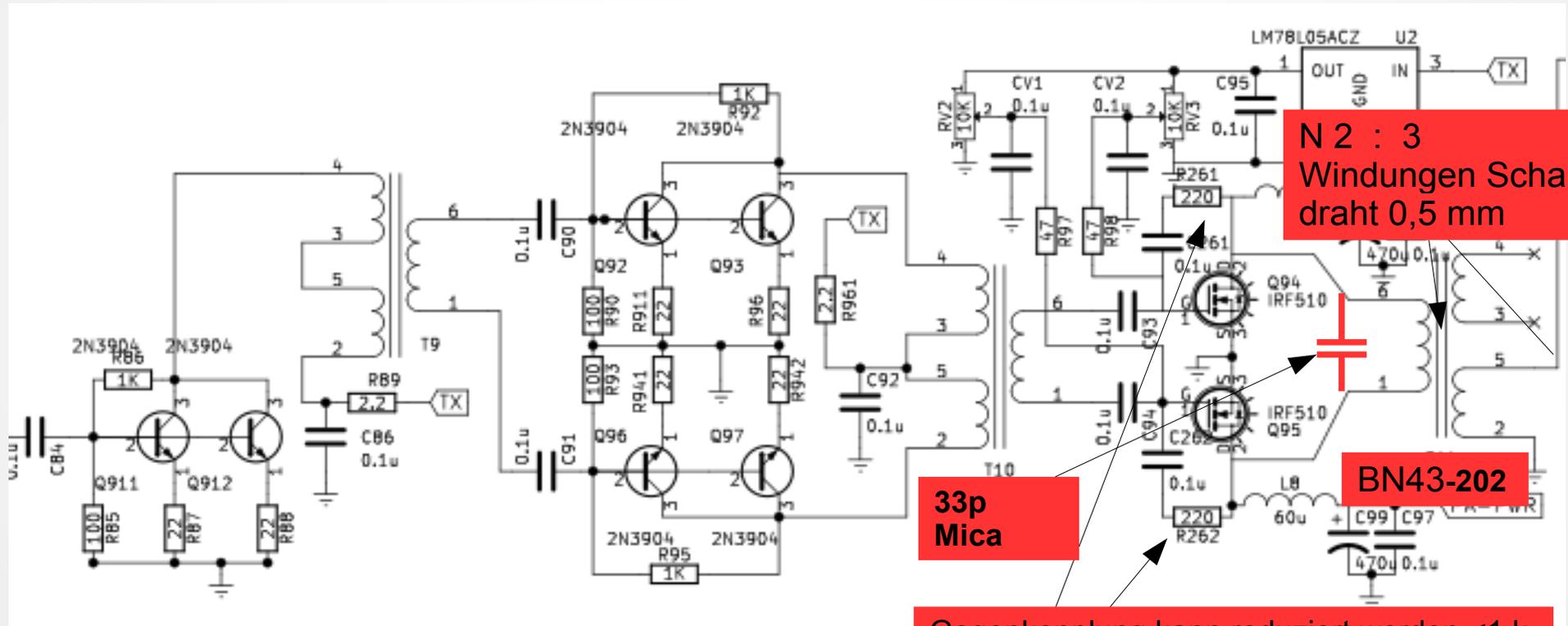


# - uBITX -

## Befunde und Maßnahmen

Befund	Maßnahme
PA bringt viel zu wenig Leistung, nur 2 W auf 10m, IRF 510 ungeeignet, Fehlanpassung	Tausch der PA-Transistoren in RD16HFF1, Ausgangstrafo neu BN43-202, 2:3 Wdgn
Keine AGC – unzumutbare Lautstärkeunterschiede, kein Ohrhörerbetrieb möglich	Regelung mit PIN-Diode in 2. ZF-Stufe, Regelungsumfang ca. 25 dB,
Stark rauschender NF-Verstärker-IC, 39 dB-Leerlaufverstärkung	NF-Tiefpass, Reduzierung der Verstärkung durch Gegenkopplung
Extrem starkes Umschaltknacken, rx-tx pop, CW-Sidetone zu laut, SSB-Geräusch im LS	Stummschaltung des NF-Eingangs, Versorgung NF-IC mit Vrx, Sideton in LS
Mikrofon muss relativ laut besprochen werden	Einbau des Mikrofon-VV mit Sprachprozessor von ELV

# - uBITX - Schaltplan - PA



N 2 : 3  
Windungen Schalt-  
draht 0,5 mm

BN43-202

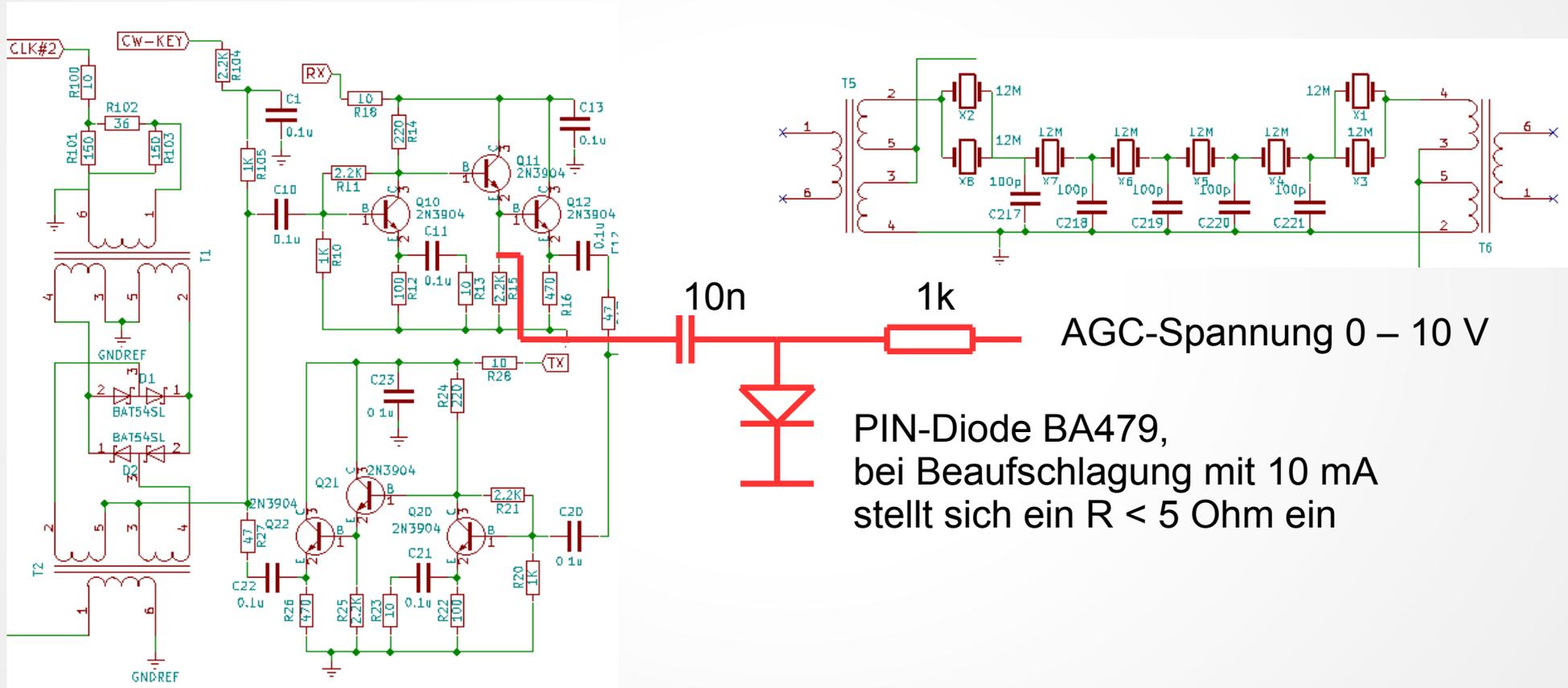
33p  
Mica

Gegenkopplung kann reduziert werden <1 k,  
Ruhestrom je FET 250mA

# - uBITX - Modifizierte PA



# - uBITX - Schaltplan - Auszüge

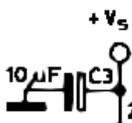


# - uBITX -

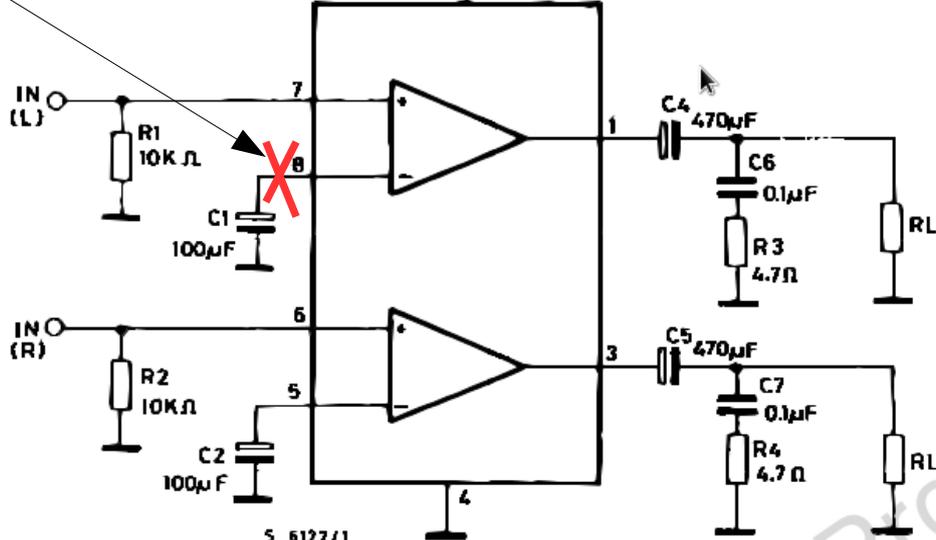
## NF-Stufe - Einsatz des TDA2822 ist problematisch

Figure 1 : Test Circuit (Stereo)

zwecks Stummschaltung und  
Gegenkopplung unterbrechen



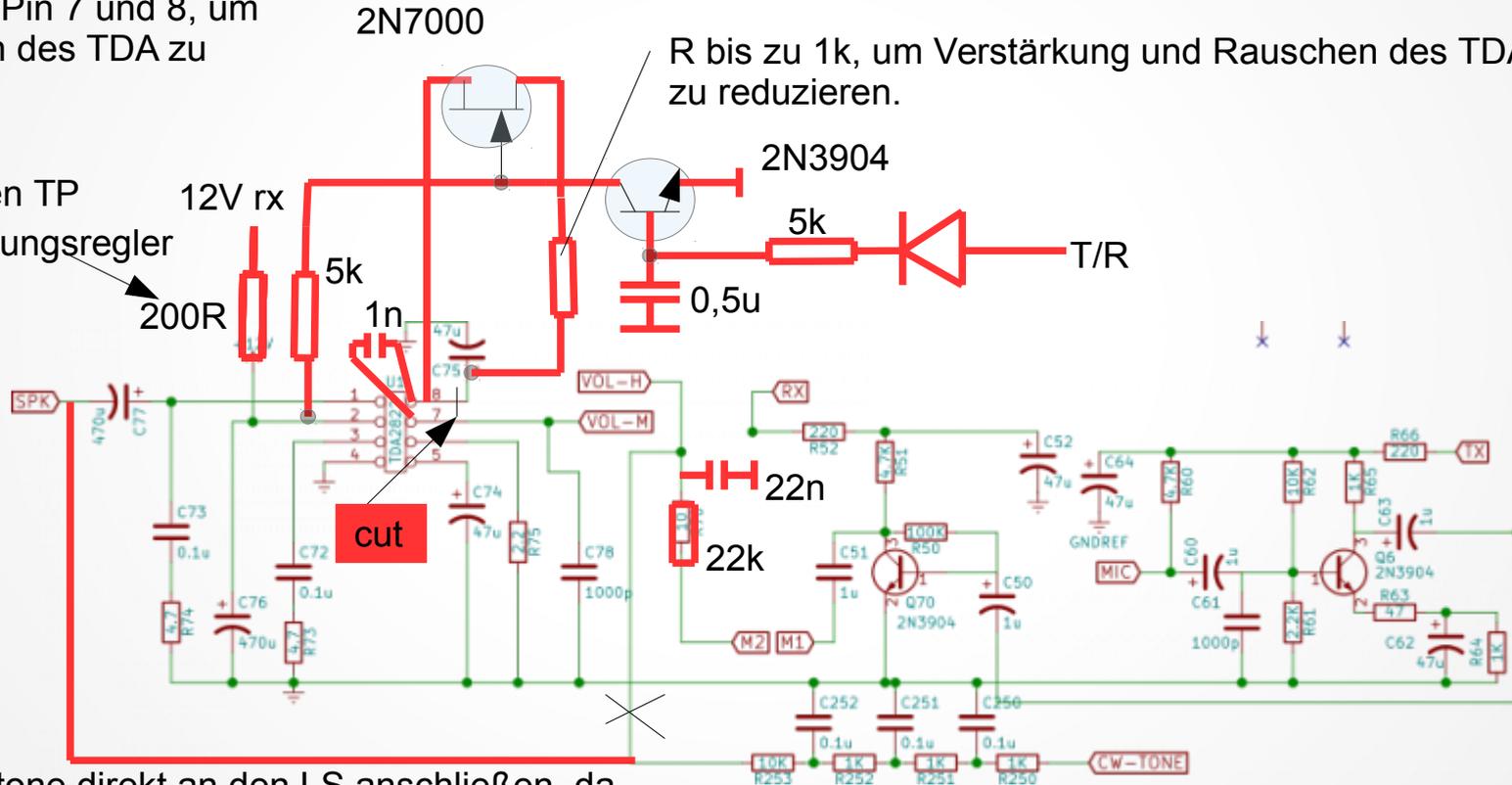
TDA2822 hat  
39 dB Spannungsverstärkung !



# - uBITX - rx-tx-pop fix and AF LPF

1nF zwischen Pin 7 und 8, um HF-Schwingen des TDA zu Verhindern.

Alternativ aktiven TP oder Festspannungsregler



R bis zu 1k, um Verstärkung und Rauschen des TDA 2822 zu reduzieren.

zu lauten Sidetone direkt an den LS anschließen, da der TDA nun ohne Spannung bei tx ist – andernfalls brabbelt es bei SSB-Sendung

uBITX@G25

# - uBITX - ELV Sprachprozessor

ELV-Kompressor SMV5  
Anpassungen für Funkgeräte

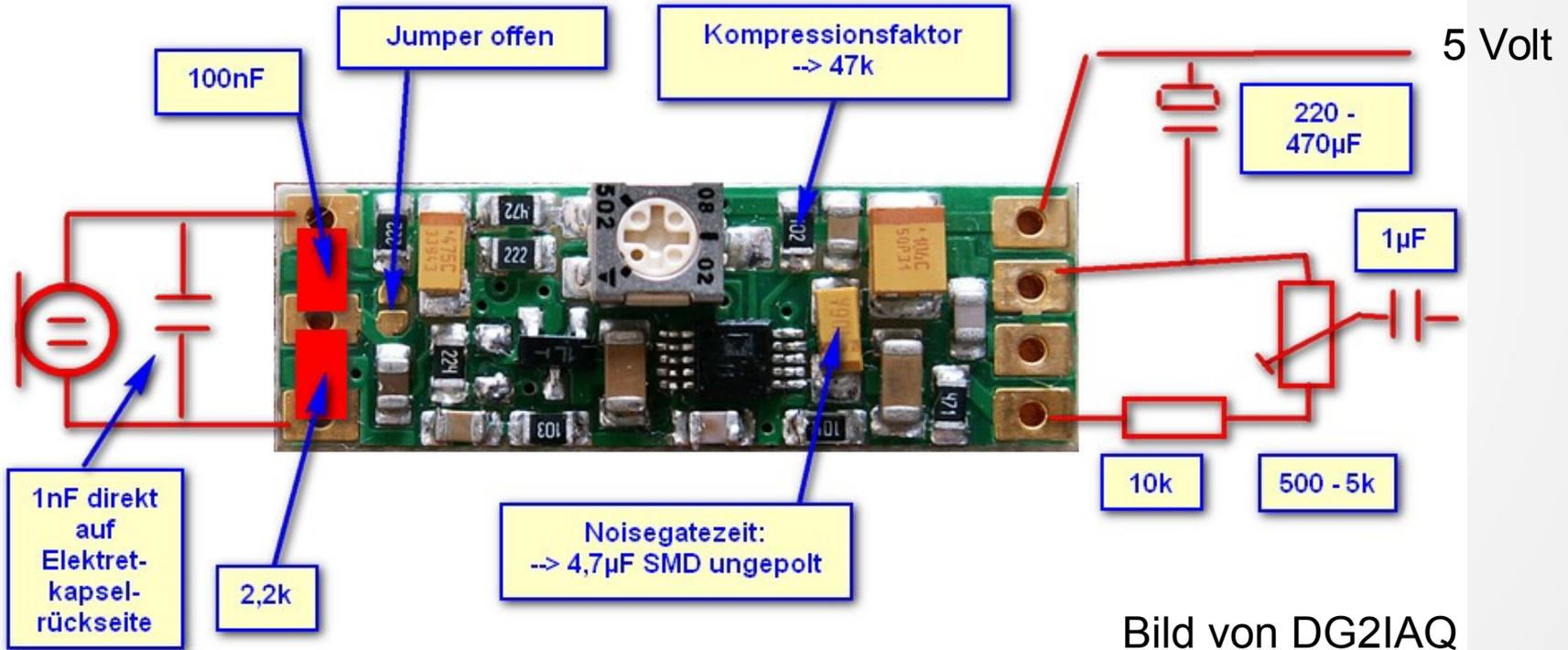


Bild von DG2IAQ

# - uBITX - Sprachprozessor Schaltplan mit Änderungen

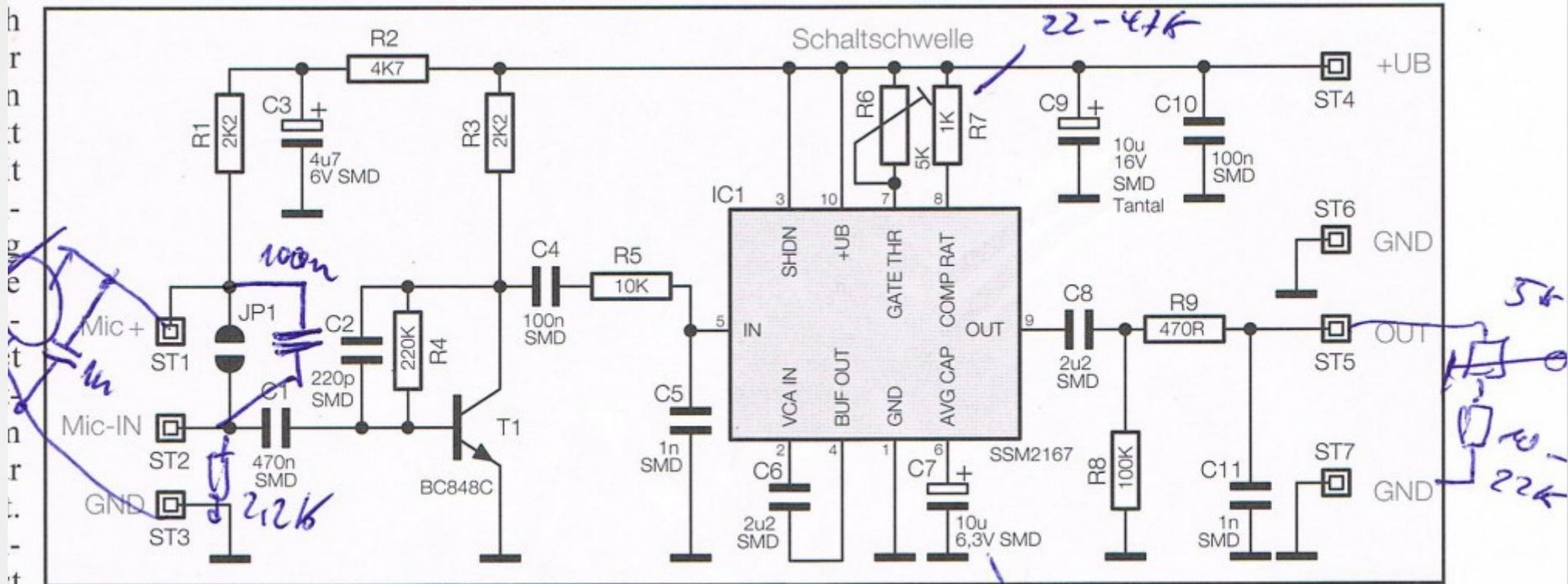
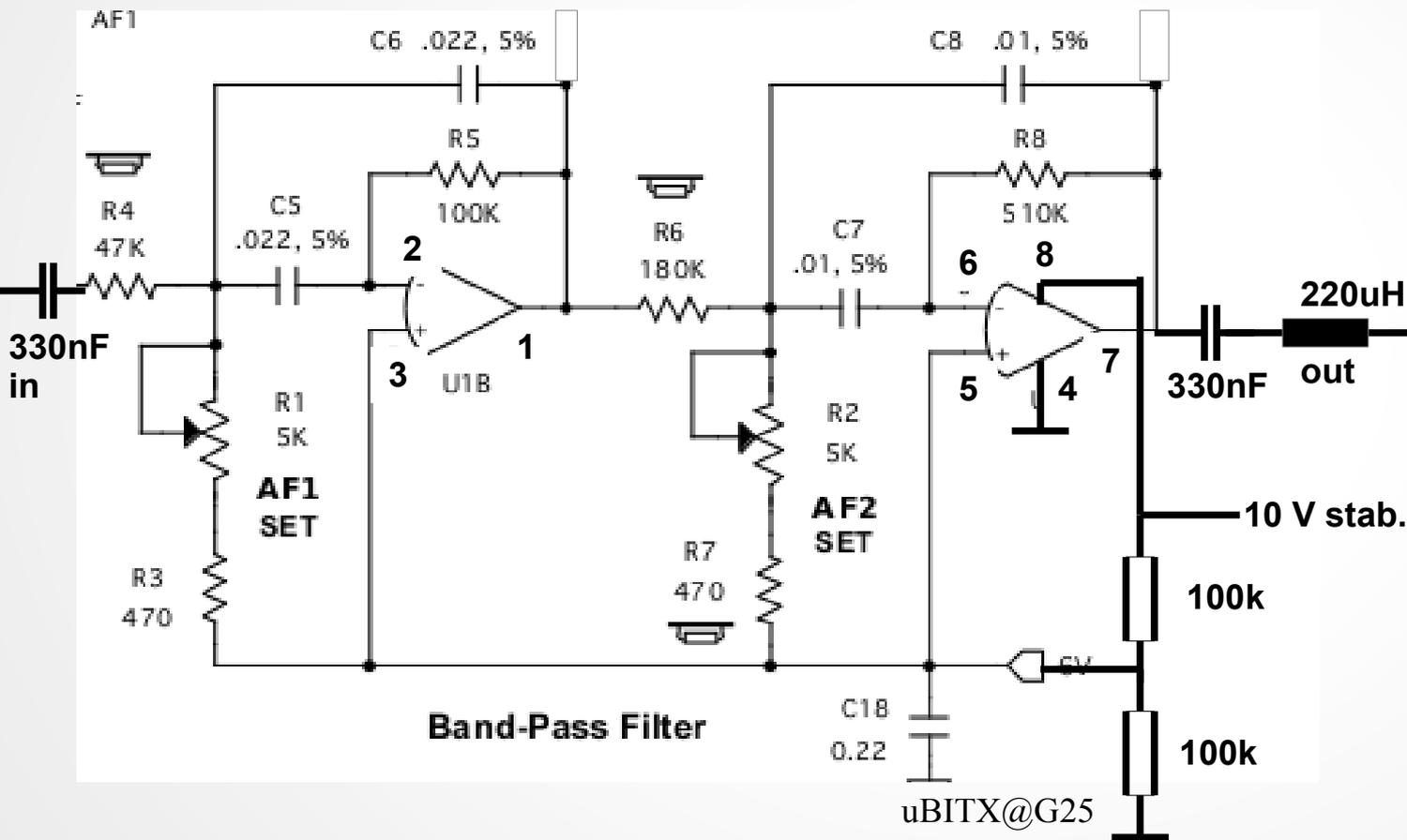


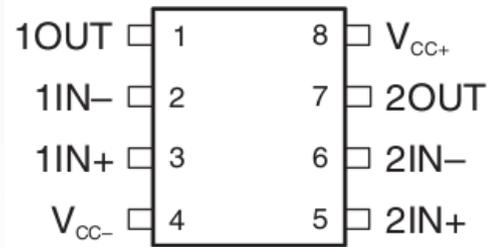
Bild 1: Schaltbild des Mikrofonverstärkers von ELV

# - uBITX -

## Beispiel für 2-stufiges NF-CW-Filter (aus Elecraft K2)



(TOP VIEW)



rauscharmer  
Dual-Op-Amp NE5532

NF-CW-Filter vor dem  
Lautstärkepoti einfügen,  
zum Ausschalten überbrücken

# - uBITX -

## Erreichte technische Daten (teils nach Modifikation)

Parameter	Wert
Empfindlichkeit Trennschärfe	mehr als ausreichend - 0,2 uV ca. 10 dbS/N Ladder Quarzfilter mit 8 Quarzen, B=2,4 kHz, optimal für SSB
Großsignalfestigkeit Pfeifstellen	völlig ausreichend, gut angepasste Schottky-Dioden-Ringmischer, ZF Verstärker haben je $v=16$ dB bei IP3 20 dBm, gesamt ca. 6 dBm, 3 Pfeifstellen in den Amateurbändern, Rest ist irrelevant
Sendeleistung Frequenzbereich	nach Modifikation 15 – 18 W, 12 W auf 10m, Trägerleistung CW 100 kHz – 30 MHz, Achtung: kein wirksames TPF auf 160m
Mikrofonempfindlichkeit	gut, Electret-Kapsel kann leise besprochen werden, Vollaussteuerung wird erreicht, Kompression > 1:4
Offene Punkte	Software-Anpassung bei CW mit ubitx_V2_01R von W0EB/W2CTX erfolgt

# - uBITX Firmware Update - Arduino IDE SW (Vorgehen bei Linux)

- Download der gepackten Datei „arduino-1.8.5-linux64.tar.xz“  
<https://www.arduino.cc/en/main/software>
- `tar -xf arduino-1.8.5-linux64.tar.xz` (Befehl entpackt Datei im Ordner)
- `./install.sh` (als normaler User installieren)
- `usermod -a -G dialout <user>` (als root, <user> darf über USB „nach außen gehen“)
- `chmod 666 /dev/ttyS0` (Freigabe der seriellen Schnittstelle, als root durchführen)
- Start des Programms und Installation der SI5351 Library über „Sketch, Bibliothek einbinden, Bibliotheken verwalten ...“
- Download `ubitx_V2_01R.zip`, (tnx @ W2CTX & W0EB) entpacken und die Datei `ubitx_V2_01R.ino` mit der Arduino-IDE-SW öffnen
- Radiuno mit USB verbinden, SW kompilieren und hochladen – der Raduino wird dabei über USB gespeist
- (Bei Windows ist der Gerätemanager zu bemühen und ggfs. der Treiber noch zu installieren – dieser ist bei Linux schon im Kernel)

# - uBITX - Arduino IDE SW



ubitx\_V1\_00R | Arduino 1.8.5

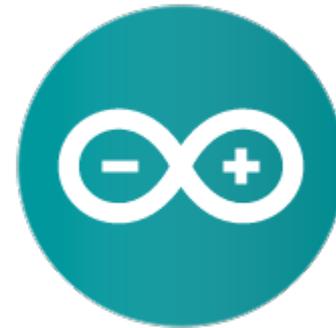


Datei Bearbeiten Sketch Werkzeuge Hilfe



ubitx\_V1\_00R   ubitx.h   ubitx\_EEPROM.cpp   ubitx\_factory\_alignment.cpp   ubitx\_keyer.cpp   ubitx\_menu.cpp   ubitx\_si5351.cpp   ubitx\_ui.cpp

```
/**
 * File name ubitx_V1_00R.ino
 * This source file is under General Public License version 3.
 *
 * This version uses a built-in Si5351 library
 * Most source code are meant to be understood by the compilers and the computers.
 * Code that has to be hackable needs to be well understood and properly documented.
 * Donald Knuth coined the term Literate Programming to indicate code that is written be
 * easily read and understood.
 *
 * The Raduino is a small board that includes the Arduino Nano, a 16x2 LCD display and
 * an Si5351a frequency synthesizer. This board is manufactured by Paradigm Ecomm Pvt Ltd
 *
 * To learn more about Arduino you may visit www.arduino.cc.
 *
 * The Arduino works by starts executing the code in a function called setup() and then it
 * repeatedly keeps calling loop() forever. All the initialization code is kept in setup()
 * and code to continuously sense the tuning knob, the function button, transmit/receive,
 * etc is all in the loop() function. If you wish to study the code top down, then scroll
 * to the bottom of this file and read your way up.
 *
 * Below are the libraries to be included for building the Raduino
 * The EEPROM library is used to store settings like the frequency memory, calibration data,
 * call sign etc .
 *
 * The main chip which generates up to three oscillators of various frequencies in the
 * Raduino is the Si5351a. To learn more about Si5351a you can download the datasheet
 * from www.silabs.com although, strictly speaking it is not a requirement to understand this code.
 * Instead, you can look up the Si5351 library written by xxx, yyy. You can download and
 * install it from www.url.com to compile this file.
```



- uBITX -

# Aufbau, Erfahrungen und erforderliche Modifikationen

