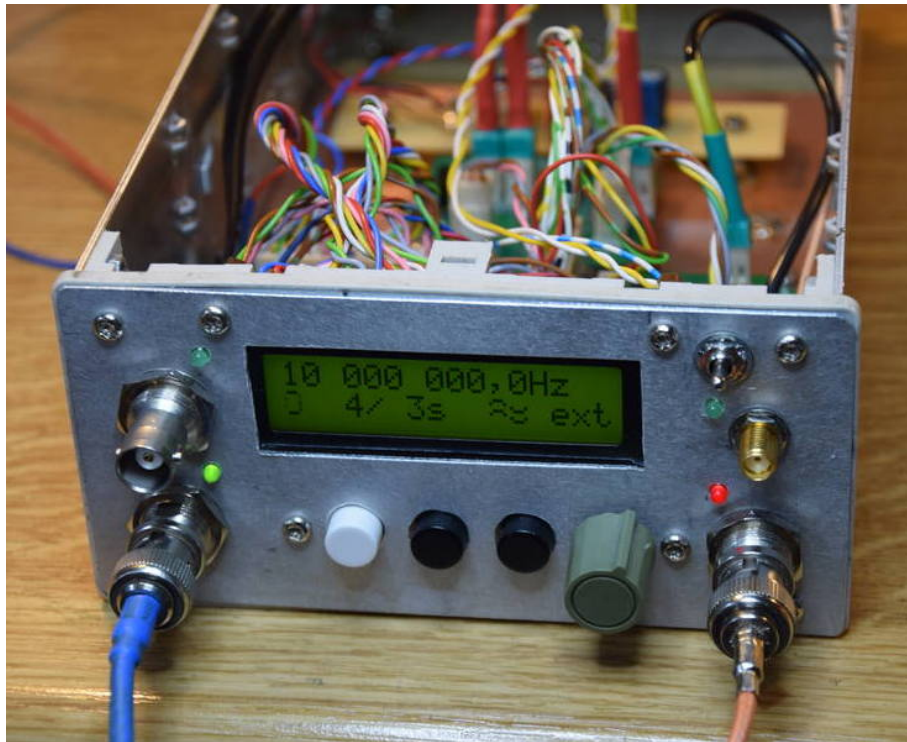


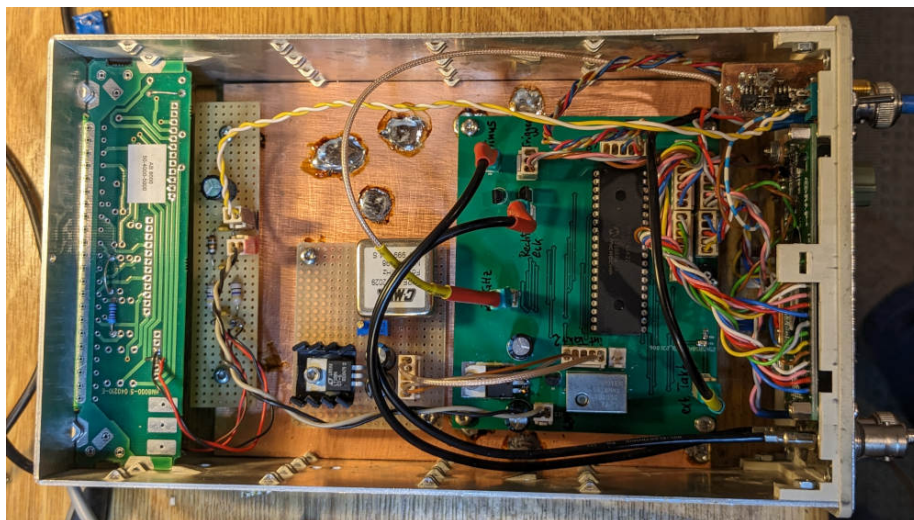
Frequenzmesser  
nach dem Reziprok-Verfahren  
Hardware 2.01  
Aufbau HW und  
Beschreibung SW

(c) DL4JAL

17. Januar 2024



Ich habe den Frequenzmesser in einen leeren Einschub von *Hameg Modular System 8000* gebaut. Ich kann aber auch den Einschub separat betreiben mit einem 12 Volt Netzteil. Das Kabel rechts kommt vom externen Takt (GPS Frequenznormal 10,000 MHz). Das linke Kabel kommt vom DDS VFO, ein Thermostat besitzt. Eingestellt sind genau 10,0 MHz.



Die neue Hauptplatine. Links habe ich noch einen OCXO 10MHz eingebaut. Der geht etwas genauer als der TXCO 12,8 MHz.

# Inhaltsverzeichnis

<b>1</b>	<b>Vorwort</b>	<b>3</b>
<b>2</b>	<b>Hardware</b>	<b>4</b>
2.1	Platine „hauptplatine“ . . . . .	4
2.1.1	Die ICs der „hauptplatine“ . . . . .	6
2.1.2	Die Stecker auf der „hauptplatine“ . . . . .	9
2.2	Platine „vv_2_8_ghz“ . . . . .	12
2.2.1	ICs und Lötunkte . . . . .	12
2.3	Platine „tasten_poti“ . . . . .	14
2.4	Platinen für die serielle Ausgabe, RS232 . . . . .	15
<b>3</b>	<b>Schaltungsbeschreibung des Frequenzzählers</b>	<b>17</b>
<b>4</b>	<b>Hilfe bei der Fehlersuche</b>	<b>20</b>
4.0.1	Weg der Taktfrequenz intern . . . . .	20
4.0.2	Weg der Messfrequenz, Trigger . . . . .	20
4.0.3	Torsignal und CLR-Signal 74HC393 . . . . .	21
<b>5</b>	<b>Software</b>	<b>22</b>
5.1	Version der Firmware . . . . .	22
5.2	Tastenfunktionen . . . . .	22
5.2.1	Normalbetrieb . . . . .	22
5.3	SETUP (Taste 1 lang) . . . . .	23
5.3.1	Kalib. Takt int. . . . .	24
5.3.2	Takt-Switch auto . . . . .	28
5.3.3	Edit Takt extern . . . . .	28
5.4	Messbeispiele . . . . .	29
5.5	Serielle Ausgabe der Messwerte . . . . .	30
<b>6</b>	<b>Schlusswort</b>	<b>31</b>
<b>7</b>	<b>Zusätzliche Bilder</b>	<b>33</b>
<b>8</b>	<b>Angehängte Dateien</b>	<b>39</b>

# Kapitel 1

## Vorwort

**Dieses Projekt darf nicht kommerziell vermarktet oder genutzt werden. Alle Rechte liegen bei DL4JAL (Andreas Lindenau).**

Ich wollte mir schon seit längerer Zeit eine Frequenzmesser bauen, der etwas genauer geht als die sonstigen üblichen Konzepte. Ich hatte mir vorgenommen möglichst eine Messgenauigkeit kleiner 1Hz im Kurzwellenbereich zu erreichen. Das klingt zwar etwas übertrieben, aber für die Kontrolle geringster Frequenzabweichungen, wenn man zum Beispiel die Funktion eines Thermostats überprüfen will, ist das ein nicht zu hoch gestecktes Ziel. Ich habe mein gestecktes Ziel erreicht und bin mit meiner Konstruktion zufrieden.

In der HW 2.01 habe ich noch einen Stecker J14 eingebaut für die Ausgabe der Messwerte mit einem zusätzlichen Zeitstempel.

## Kapitel 2

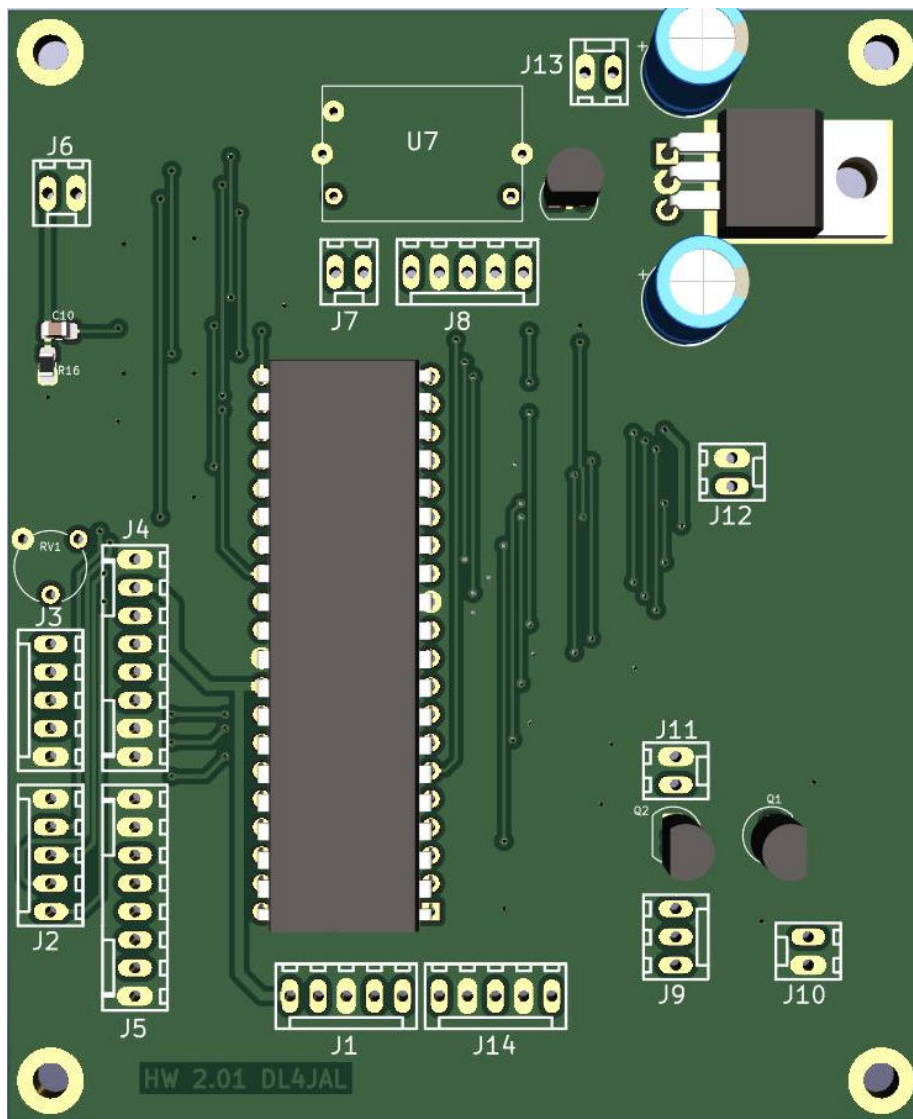
# Hardware

Die HW Version 2.0 besteht aus einer Hauptplatine und nicht mehr aus 3 Einzelplatinen. Die ganzen Verbindungsstecker fallen weg.

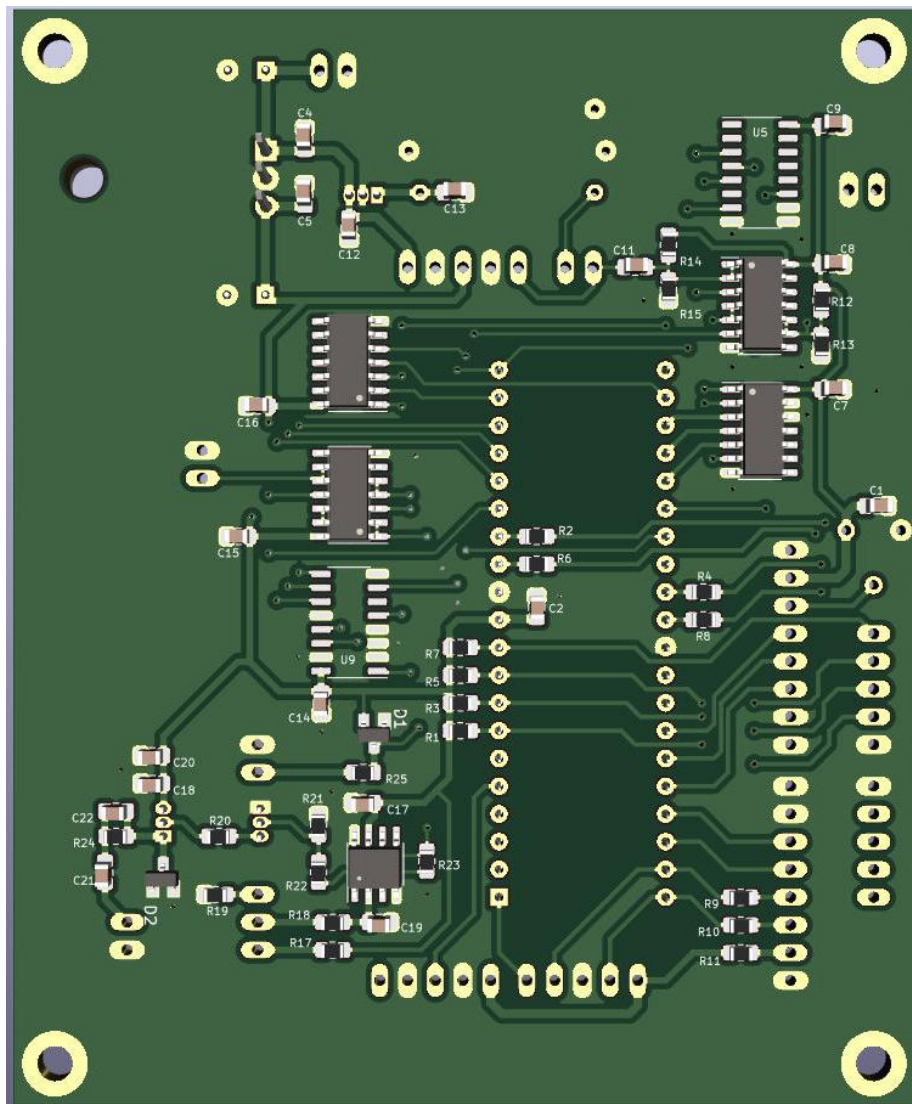
Davon habe ich wieder Platinen fertigen lassen. Nach dem Bestücken und Testen funktionierte alles tadellos. Ein bisschen knifflig ist die Auswahl der Widerstände am Poti für die Triggereinstellung. Die Schaltbilder befinden sich am Ende der PDF. Die habe ich mit einem Linux-PDF-Tool an die diese PDF-Datei hinten angehängen.

### 2.1 Platine „hauptplatine“

Die Hautplatine habe ich mit Kicad 7 entworfen und Leiterplatten fertigen lassen.



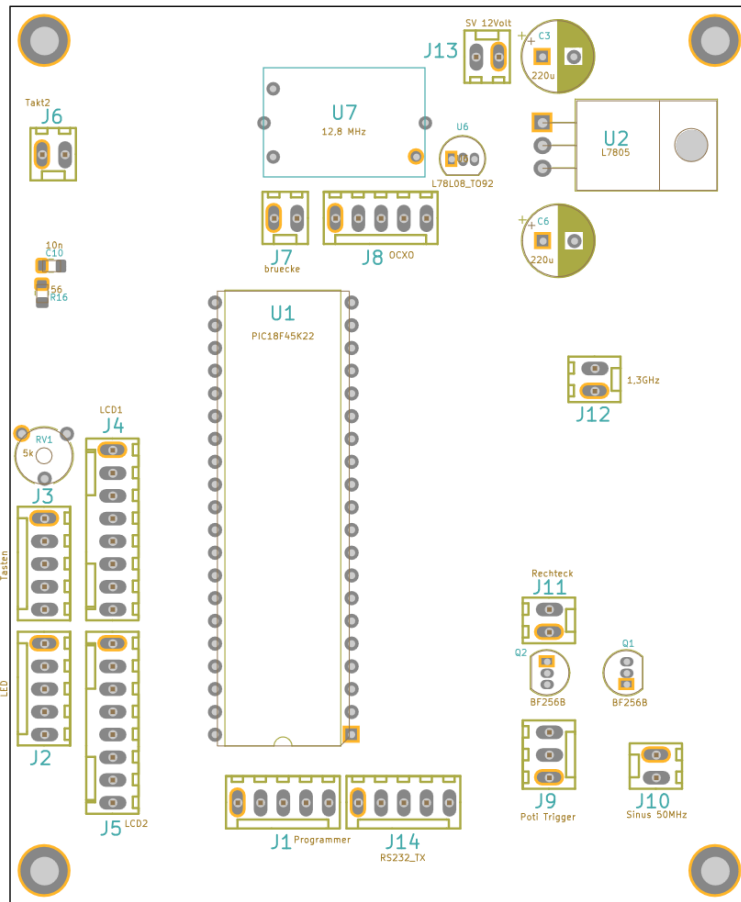
Ansicht der Platine von Oben. Wir sehen alle Steckplätze und U1 den PIC18F45K22. Zusätzlich J14 neben J1 für die RS232 Ausgabe.



Auf der Unterseite sind fast alle SMD-Bauteile und ICs untergebracht.

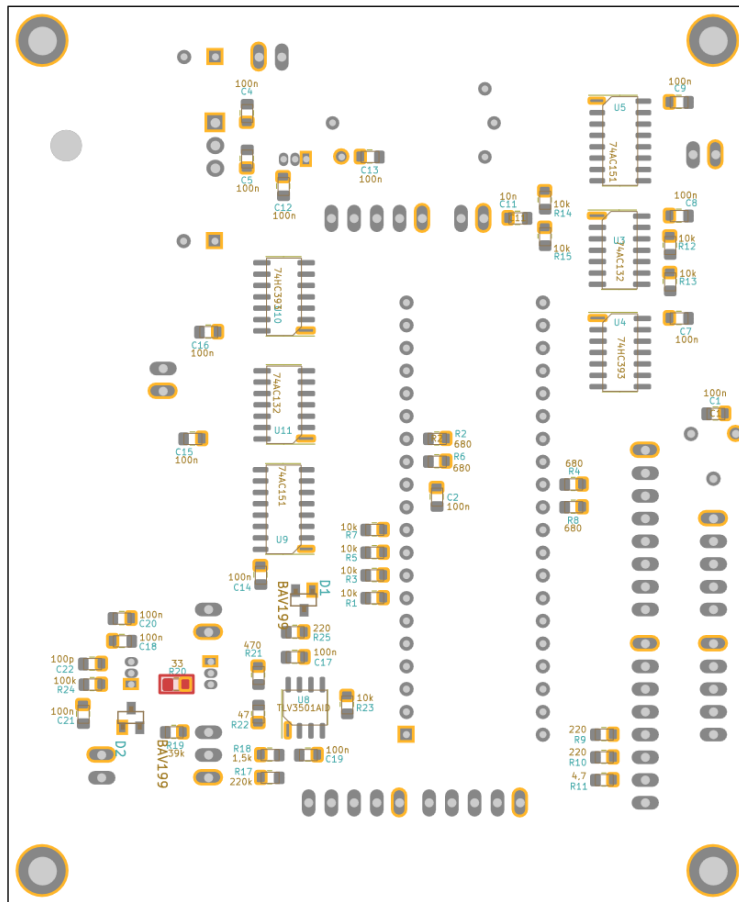
### 2.1.1 Die ICs der „hauptplatine“

In Kicad 7 ist es möglich eine HTML Bestückungshilfe zu generieren. Es folgen 2 Bilder aus dieser Generierung.



Bei allen Footprints ist Pin1 markiert. Hier zu sehen ist die Oberseite der Platine. Neu ist der Stecker J14 für die serielle Ausgabe der Messdaten.





Auch sehen wir von allen Bauelementen Pin1 als Markierung. Die rote Markierung resultiert aus der Position in der Bestückungstabelle der HTML.

#### Die ICs oben auf der Platine

**U1 PIC18F45K22** Der Mikrocontroller, der alles steuert.

**U2 L7805** Der Spannungsregler 5 Volt für fast alle ICs.

**U6 L7808** Der Spannungsregler 8 Volt für den TCXO 12,8 MHz.

#### Die ICs unten auf der Platine

**U3 74AC132** Taktfrequenz Triggertaster

**U4 74HC393** Taktfrequenz Schneller binärer Zähler (70MHz).

**U5 74AC151** Taktfrequenz Umschalter.

**U7 TCXO 12,8 MHz** Taktgenerator intern.

**U8 TLV3501AID** sehr schneller Komparator für die Messfrequenz.

**U9 74AC151** Messfrequenz Umschalter.

**U10 74HC393** Messfrequenz Schneller binärer Zähler (70MHz).

**U11 74AC132** Messfrequenz Triggertaster

### 2.1.2 Die Stecker auf der „hauptplatine“

Es folgt die Beschreibung der Steckverbinder auf der „Hauptplatine“.

#### **J1, 5 polig, Programmer, SV VV-2,8GHz**

Diese Steckverbindung hat 2 Funktionen. Einmal dient er der Programmierung des PIC18F45K22 und die zweite Funktion ist die Stromversorgung der Vorteilplatine 2,8GHz.

PIN	Programmer	SV VV 2,8GHz
Pin 1	5 Volt	5 Volt VV-Platine
Pin 2	RB7	frei
Pin 3	GND	GND VV-Platine
Pin 4	RB6	frei
Pin 5	MCLP	frei

#### **J2, 5 polig, LEDs der Koaxbuchsen**

In diesem Stecker werden die LEDs auf der Frontplatte angeschlossen. Die LEDs dienen der Information, welcher Koaxeingang gerade aktiv ist.

PIN	Beschreibung
Pin 1	GND
Pin 2	RA6, LED Sinus Eingang
Pin 3	RD6, LED Rechteck Eingang
Pin 4	RA7, LED VV-2,8GHz Eingang
Pin 5	RD7, LED externer Takt 10 MHz

#### **J3, 5 polig, Tasten der Bedienung**

Hier werden die Tasten 1 bis 3 angeschlossen.

PIN	PORT	zur BG	Beschreibung
1	RE2	Frontplatte	Taste 3
2	RE1	Frontplatte	Taste 2
3	RE0	Frontplatte	Taste 1
4	RA5	bleibt frei	frei
5	GND	Frontplatte	Masse

#### **J4, 8 polig, LCD1 Display**

Hier wird das LCD-Display angeschlossen PIN 1 bis PIN 8. Die Drähte werden direkt ans Display gelötet.

#### **J5, 8 polig, LCD2 Display**

Hier wird das LCD-Display angeschlossen PIN 9 bis PIN 16. Die Drähte werden direkt ans Display gelötet.

### J6, 2 polig, externer Takt 10MHz

Verbindung zur Koaxbuchse *externer Takt*, mit RG174 oder RG178.

PIN	Funktion	Beschreibung
Pin 1	Eingang	externer Takt
Pin 2	Eingang	GND

### J7, 2 polig, bruecke

Das ist nur für einen Brückenstecker. Beide PINs müssen verbunden werden, wenn der TXCO 12,8MHz benutzt wird.

### J8, 5 polig, OCXO

Verwenden wir für den Frequenzzähler eine andere Taktquelle, wird diese hier angeschlossen. Ich habe ein OCXO 10,0 MHz hier angeschlossen. Der geht natürlich noch etwas genauer als der TXCO 12,8 MHz. Die verwendete Frequenz wird im SETUP eingestellt.

PIN	Funktion	Beschreibung
Pin 1	HF	Takt vom OCXO 10MHz
Pin 2	HF-GND	Takt vom OCXO 10MHz GND
Pin 3	SV	+5V
Pin 4	SV	GND
Pin 5	SV	+12V

### J9, 3 polig, Poti für Triggerpunkt

Hier wird ein Poti angeschlossen zur Regelung des Triggerpunktes vom 50 MHz Sinus Eingang.

PIN	Richtung	Beschreibung
Pin 1	Eingang	Poti Kalt
Pin 2	Eingang	Poti Schleifer
Pin 3	Eingang	Poti Heiß

### J10, 2 polig, Messeingang Sinus

Zählereingang bis 50 MHz Sinus. Durch die Verwendung eines schnelleren Komparators geht der Eingang jetzt bis etwa 70 MHz bei 0 dBm.

PIN	Richtung	Beschreibung
Pin 1	Eingang	Zählen 50 MHz
Pin 2	Eingang	GND

### J11, 2 polig, Messeingang Rechteck

Zählereingang bis 50 MHz Rechteck.

PIN	Richtung	Beschreibung
Pin 1	Eingang	Zählen 50 MHz
Pin 2	Eingang	GND

### J12, 2 polig, Messeingang vom Vorteiler 2,8GHz

Der Ausgang der Vorteiler-Platine-2,8GHz geht auf diesen Eingang. Auf dem Vorteiler wird die hohe Frequenz durch 64 geteilt.

PIN	Richtung	Beschreibung
Pin 1	Eingang	HF vom Vorteiler
Pin 2	Eingang	GND

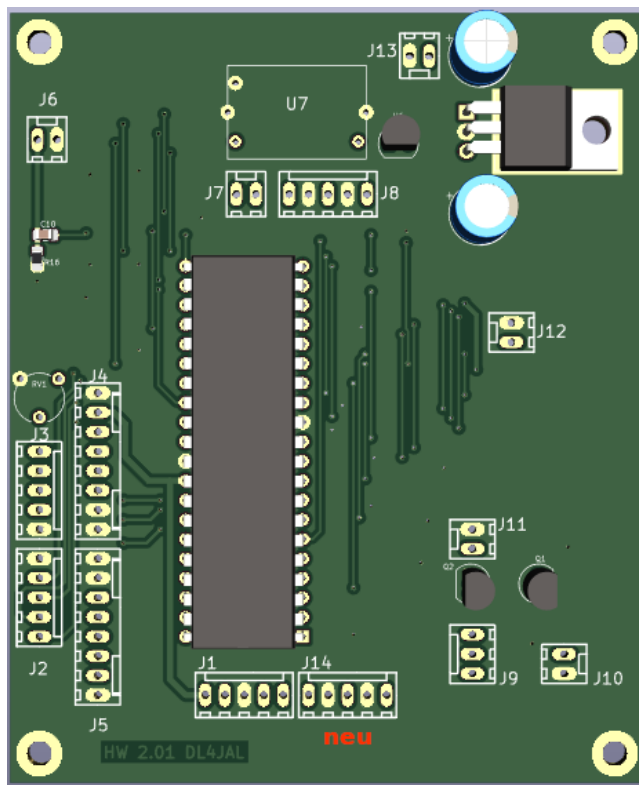
### J13, 2 polig, Stromversorgung 12 Volt

Die Versorgungsspannung des Frequenzzählers wird hier angeschlossen.

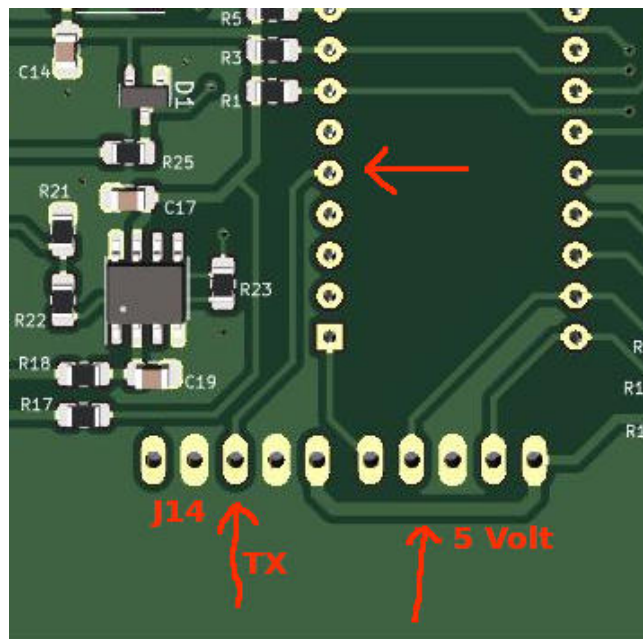
PIN	Richtung	Beschreibung
Pin 1	Eingang	12 Volt
Pin 2	Eingang	GND

### J14, 5 polig, RS232 Ausgabe

Die RS232-Ausgabe ist eine HW-Erweiterung auf der Platine. J14 wird neben J1 platziert.



J14 wird neben J1 platziert.



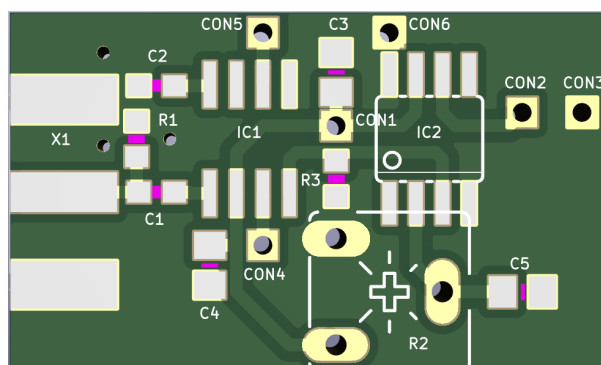
Ein Verbindung 5Volt wird an J14 mit einem Draht gelötet und ein Draht für die Verbindung *RS232 TX* von PIC, PIN5 zu PIN3 J14.

PIN	Beschreibung
1	5 Volt SV für Module
2	GND
3	RS232 Sendesignal
4	GND
5	bleibt frei

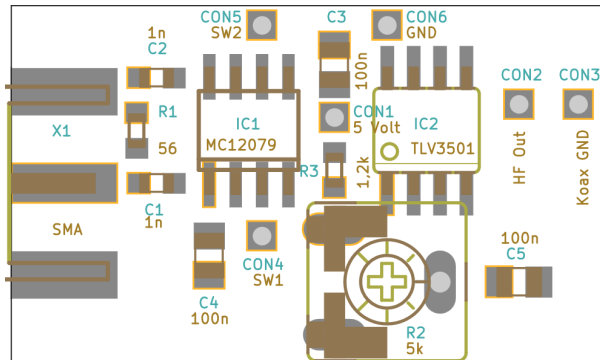
## 2.2 Platine „vv\_2.8\_ghz“

Die Platine „vv\_2.8\_ghz“ ist für Frequenzmessungen über 70 MHz vorgesehen.

### 2.2.1 ICs und Lötunkte



3D Darstellung aus Kicad 7.



Und die Darstellung aus der HTML Bestückungshilfe.

**IC1 MC12079** Vorteiler 2,8 GHz (3,2 GHz).

**IC2 TLV3501** schneller Komparator.

Lötunkte auf der Platine.

**CON1** SV 5 Volt

**CON2** HF-Out Koax RG178

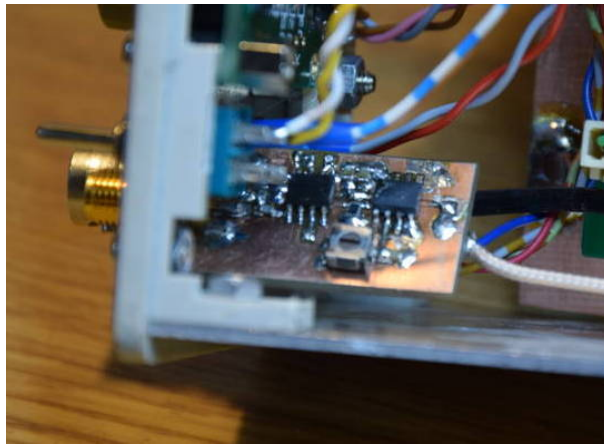
**CON3** HF-Out-GND Koax RG178

**CON4** SW1 an +5 Volt legen (Teiler 64:1)

**CON5** SW2 an +5 Volt legen (Teiler 64:1)

**CON6** SV GND

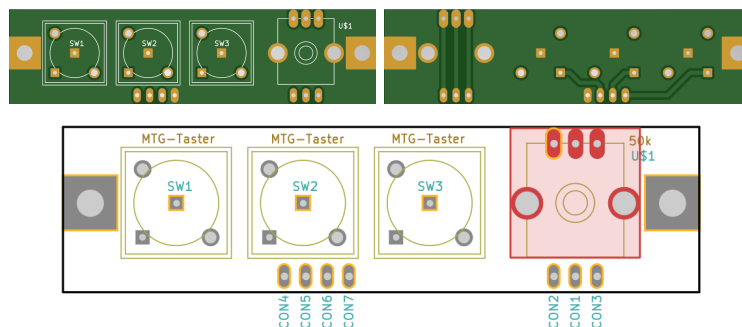
Die Platine wird mit der SMA-Buchse direkt in die Frontplatte geschraubt.



Meine Testplatine Vorteiler bis 3,2 GHz. Links der MC12079 und rechts der schnelle Komparator der aus dem ECL Pegel einen TTL Pegel macht.

## 2.3 Platine „tasten\_poti“

Diese kleine Platine ist für die Bedienung des Frequenzzählers. Links sind 3 Tasten und rechts das Poti für die richtige Trigger-Einstellung.



### Die Anschlüsse auf der Platine

Hier noch die Beschreibung der einzelnen Pin's auf der Platine. Alle Litzen werden direkt angelötet.

**CON1** geht auf den Schleifer des Potis 50k, auf der Hauptplatine J9/2

**CON2** geht auf Poti 50k, auf der Hauptplatine J9/1

**CON3** geht auf Poti 50k, auf der Hauptplatine J9/3

**CON4** Taste 3, J3/3

**CON5** Taste 3, J3/2

**CON6** Taste 3, J3/1

**CON7** GND, J3/5

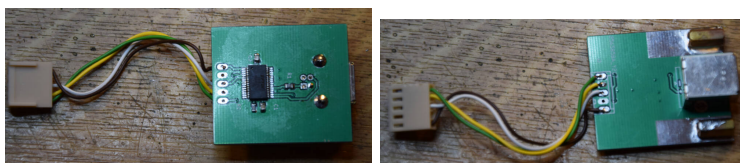
Die Platine wird an der Frontseite des Frequenzzählers mit Abstandshaltern befestigt.

## 2.4 Platinen für die serielle Ausgabe, RS232

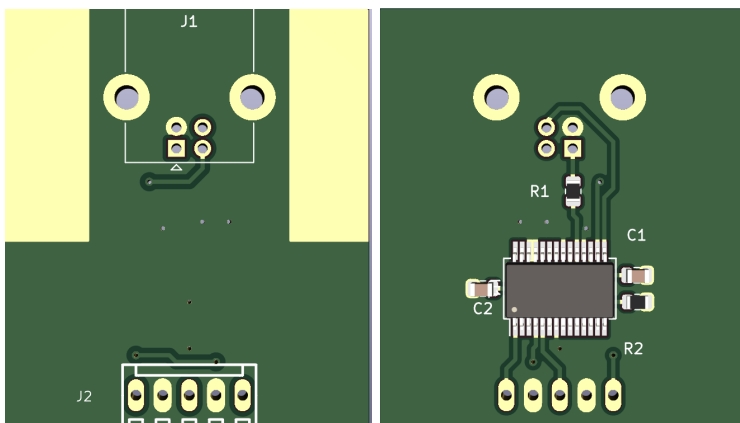
Wird die serielle Ausgabe der Messdaten genutzt ist es notwendig eine HW-Anpassung an die entsprechende Schnittstelle vorzunehmen. Dies kleinen Platine werden auf der Hauptplatine an J14 angesteckt. An J14 liegt nur ein TTL-Pegel an. Das Ausgabeformat ist 9600,8,N,1 (9600 Baud, 8Bit, Parity NO, 1 Stopbit).

### USB-Platine

Die eine Möglichkeit ist die Wandlung der seriellen TTL-Ausgabe in ein USB-Signal.



Fotos vom Modul



3D Ausgabe von Kicad 7.

Das USB-Modul enthält einen FT232RL. Dieser IC wandelt das TTL-Signal in ein USB-Signal, passend zum Betriebssystem Linux oder Windows. Mit einem Terminal-Programm können die Ausgaben gelesen werden und je nach Programm auch als Textdatei gespeichert werden.

### J2, 5 polig, RS232 Verbindung

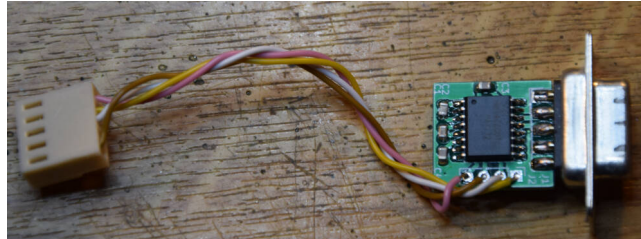
Die Steckerbelegung von J2.

PIN	Hauptplatine J14	Beschreibung
1	PIN1	5 Volt SV für Module
2	PIN2	GND
3	PIN3	RS232 Sendesignal
4	PIN4	GND
5	PIN5	bleibt frei



## RS232-Platine

Eine weitere Möglichkeit ist die Pegel-Umwandlung in eine richtige RS232 Schnittstelle.



Allerdings ist diese Variante ein Auslauf-Modell. Die wenigsten PC's habe noch eine echte serielle Schnittstelle.

Das RS232-Modul enthält einen MAX232. Dieser IC wandelt das TTL-Signal in ein RS232-Signal, passend zum Betriebssystem Linux oder Windows. Mit einem Terminal-Programm können die Ausgaben gelesen werden und je nach Programm auch als Textdatei gespeichert werden.

### J2, 4 polig, RS232 Verbindung

Die Steckerbelegung von J2. An Stecker J2 wird nur PIN1, PIN2 und PIN4 benutzt.

PIN	Hauptplatine J14	Beschreibung
4	PIN1	5 Volt SV für Module
3	PIN5	bleibt frei
2	PIN3	RS232 Sendesignal
1	PIN4	GND

## Kapitel 3

# Schaltungsbeschreibung des Frequenzzählers

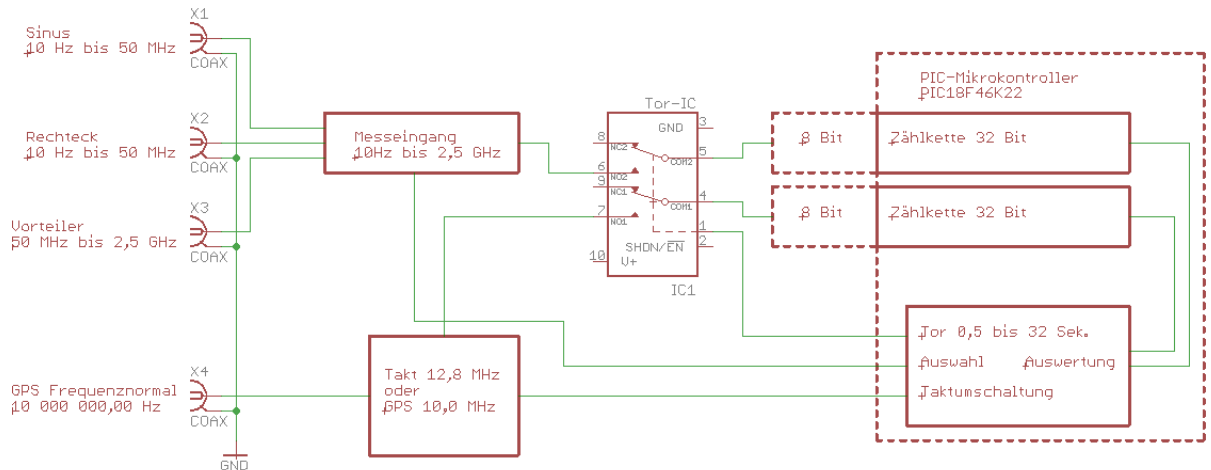
Es gibt verschiedene Möglichkeiten die Frequenz zu messen. Ich zähle mal die mir bekannten Verfahren auf:

**Frequenzzähler mit genauen Toröffnungszeiten:** Normale Frequenzzähler arbeiten nach dem Prinzip der Öffnung eines Tores wo die Schwingungen der Sinuswellen gezählt werden. Öffnet man zum Beispiel das Tor für genau eine Sekunde, ist das Zählergebnis genau die Frequenz in Hz. 1 zehntel Sekunde ergibt ein Zählergebnis auf 10 Hz genau, usw..

**Frequenzzähler mit Messung der Periode einer Sinuswelle:** Diese Frequenzzähler arbeiten nach dem Prinzip der Periodendauer einer Sinus-schwingung.

**Frequenzmesser nach der Reziproken-Verfahren** *So arbeitet mein Frequenz-messer.* Mein Frequenzmesser hat 2 identische Zählketten. In einer Zählkette wird die zu messende Frequenz gezählt und in der zweiten Zählkette eine genaue Referenzfrequenz. Wichtig bei diesem Verfahren ist das gleichzeitige Öffnen und Schließen der Tore der Zählketten. Es kommt nicht auf die genaue Zeit der Toröffnung an. Der Rest ist Mathematik.

## Blockschaltbild Frequenzzähler Reziprok



Eine Frequenzmessung hat folgenden Ablauf:

1. Beide Tore bleiben gesperrt. Beide Zählketten löschen.
2. Beide Tore gleichzeitig für eine definierte Zeit öffnen (Bereich 1..32 Sekunden). Die Öffnungszeit braucht nicht so genau stimmen. Die Öffnungszeit wird über die Taktfrequenz des PIC18F45K22 gesteuert.
3. Beide Tore werden gleichzeitig geschlossen.
4. Beide Zählketten werden getrennt ausgelesen und abgespeichert. Einmal der Vorteiler 74HC393 und die Register im PIC18F45K22.
5. Mit der Formel

$$FreqMessung = \frac{FreqZaehlkette1}{FreqZaehlkette2} * ReferenzTaktfrequenz$$

wird die zu messende Frequenz errechnet.

6. Wurde die Frequenz mit dem Vorteiler 64:1 gemessen (Messfrequenz > 70MHz), ist das Ergebnis noch mit 64 zu multiplizieren.

$$Frequenz = Frequenz * 64$$

7. Ist die RS232 Ausgabe aktiv, wird jetzt ein Zeitstempel in Sek und die gemessene Frequenz auf der seriellen Schnittstelle ausgegeben.
8. Der Messzyklus ist beendet. Die Messung beginnt wieder von vorn.

Je nach Dauer der Toröffnung errechnet sich folgende Messgenauigkeit. Diese mathematische Betrachtung gilt nur, wenn der Referenztakt eine sehr genaue bekannte Frequenz ist. Wie zum Beispiel ein GPS-Frequenznormal oder ein GPSDO (Siehe [1]). So einen GPSDO kann ich nur empfehlen. Ich habe einen

GPSDO im Einsatz für den QO100-TRX Adalm-Pluto als genauen Referenz-  
takt 40 MHz. In der neuen HW 2.0 habe ich einen OCXO 10,0 MHz eingebaut.

**Errechnete Messgenauigkeit:**

- Torzeit 1 Sek: +/- 1 Hz. Mit Vorteiler (DIV 64) +/- 64 Hz.
- Torzeit 2 Sek: +/- 0,5 Hz. Mit Vorteiler (DIV 64) +/- 32 Hz.
- Torzeit 4 Sek: +/- 0,25 Hz. Mit Vorteiler (DIV 64) +/- 16 Hz.
- Torzeit 8 Sek: +/- 0,125 Hz. Mit Vorteiler (DIV 64) +/- 8 Hz.
- Torzeit 16 Sek: +/- 0,063 Hz. Mit Vorteiler (DIV 64) +/- 4 Hz.
- Torzeit 32 Sek: +/- 0,03 Hz. Mit Vorteiler (DIV 64) +/- 2 Hz.

Diese Werte sehen doch nicht schlecht aus.

## Kapitel 4

# Hilfe bei der Fehlersuche

Es ist eventuell sinnvoll wenn ich den Weg der beiden Zählketten beschreibe.

### 4.0.1 Weg der Taktfrequenz intern

Angenommen die Taktfrequenz beträgt 12,8MHz. Wir Messen mit dem Oszi der Reihe nach oder auch rückwärts vom PIC18F46K22.

#### Auflistung der Messpunkte

- An Pin 1/2 von U3 74AC132 liegt 12,8MHz an.
- An Pin 3 von U3 74AC132 liegt 12,8MHz an.
- An Pin 3 von U3 74AC151 liegt 12,8MHz an.
- An Pin 5 von U3 74AC151 liegt 12,8MHz an.
- An Pin 1 von U4 74HC393 liegt 12,8MHz an.
- An Pin 6 von U4 74HC393 liegt 800kHz an. (12,8MHZ / 16)
- An Pin 4/5 von U3 74AC132 liegt 800kHz an.
- An Pin 6 von U3 74AC132 liegt 800kHz an.
- An Pin 6 PIC18F46K22 Port A4 liegt 800kHz an.

### 4.0.2 Weg der Messfrequenz, Trigger

Angenommen die Messfrequenz beträgt 10MHz. Wir Messen mit dem Oszi der Reihe nach oder auch rückwärts vom PIC18F46K22.

#### Auflistung der Messpunkte

- An Pin 1/2 von U11 74AC132 liegt 10MHz an.
- An Pin 3 von U11 74AC132 liegt 10MHz an.
- An Pin 1 von U9 74AC151 liegt 10MHz an.

- An Pin 5 von U9 74AC151 liegt 10MHz an.
- An Pin 1 von U10 74HC393 liegt 10MHz an.
- An Pin 6 von U10 74HC393 liegt 625kHz an. (10MHZ / 16)
- An Pin 12/13 von U11 74AC132 liegt 625kHz an.
- An Pin 11 von U11 74AC132 liegt 625kHz an.
- An Pin 15 PIC18F46K22 Port C0 liegt 625kHz an.

### 4.0.3 Torsignal und CLR-Signal 74HC393

Die Vorteiler 74HC393 müssen am Beginn der Zählung zurück gesetzt werden.

#### CLR 74HC393

- Pin 2 von U4 74HC393.
- Pin 28 PIC18F46K22 Port D5.
- Pin 2 von U10 74HC393.
- Pin 19 PIC18F46K22 Port D0.

Beide Zählketten müssen zu gleicher Zeit geöffnet und geschlossen werden. Während das Tor geschlossen ist, werden beide Frequenzen ausgewertet und die Messfrequenz berechnet.

#### Torsignal 74AC151

- Pin 2 von U4 74HC393.
- Pin 2 von U10 74HC393.
- Pin 20 PIC18F46K22 Port D1.

# Kapitel 5

## Software

Die Hauptarbeit des Projektes steckt in der Entwicklung der Software. Ich habe alles mit Assembler geschrieben und Kompiliert. Für die mathematischen Berechnungen nutze ich eine selbst geschriebenes Gleitpunkt-Paket mit erhöhter Genauigkeit. Die Stellenzahl reicht für die Darstellung von MHz mit 2 Nachkommastellen. Das würde mit einfachen Float 32 Bit nicht gehen.

### 5.1 Version der Firmware

Im Moment ist die Version 2.04 aktuell. Ich arbeite weiter an der Verbesserung der Funktionen.

Zusätzlich wurde in der Version 2.04 noch eine Ausgabe der Messwerte auf eine serielle Schnittstelle programmiert. Das ist eine virtuelle Nachbildung der RS232-TX, da die Pin's beiden echten RS232-Schnittstellen mit anderen Funktionen belegt sind.

### 5.2 Tastenfunktionen

Der Frequenzmesser wird mit 3 Tasten bedient. Die Tasten haben eine Doppelfunktion die ich in einer Tabelle aufliste.

#### 5.2.1 Normalbetrieb

**Taste kurz gedrückt** Die Taste wird nicht länger als 1 Sekunden gedrückt.

**Taste lang gedrückt** Die Taste wird länger als 1 Sekunden gedrückt. Im Display kommt eine Kurzinfo.

Taste	Taste kurz gedrückt	Taste lang gedrückt
1	Time-Tor	SETUP
2	Quelle	RS232 <i>ein</i> oder <i>aus</i>
3	Takt	RS232 Zeitstempel wird wieder 0 Sekunden

**Time-Tor** In dieser Funktion wird die Torzeit eingestellt. Mit jedem Tastendruck erhöht sich die Zeit. Es beginnt mit 1 Sekunde. Dann folgt 2, 4, 8, 16 und 32 Sekunden. Nach 32 Sekunden beginnt die Torzeit wieder mit 1

Sekunde. Je nach Torzeit und Messfrequenz ändert sich die Anzeige des Messergebnisses. Bei langen Torzeiten werden auch die Nachkommastellen mit angezeigt. Ist die Messfrequenz dazu noch niedrig erhöhen sich noch einmal die Nachkommastellen passend.

**Quelle** Die Eingangs-Quelle wird umgeschaltet. Der Frequenzmesser hat 3 Messingänge, Sinus, Rechteck und Eingang mit Vorteiler für hohe Frequenzen.

**Takt** Der Frequenzmesser hat intern einen TCXO 12,8 MHz. Wer einen noch genaueren Takt verwenden will kann ihn extern einspeisen, an Koax-Stecker rechts unten. Sehr gut eignet sich ein GPSDO, den man in der Frequenz einstellen kann (Siehe [1]). Aber ein Eigenbau GPS-Frequenznormal funktioniert genau so gut (Siehe [2]).

**SETUP** Der *lange Tastendruck* führt in das SETUP. Das beschreibe ich im Kapitel 5.3 auf Seite 23.

**RS232 ein/aus** Ab FW 2.04 ist es möglich die Messung auf der RS232 auszugeben. Ein „USB-Symbol“ in LCD Zeile 2 zeigt an ob die RS232 aktiv ist.



RS232 ist aktiv.

**Zeitstempel=0** Diese Funktion setzt den Zeitstempel der RS232-Ausgabe zurück.

## 5.3 SETUP (Taste 1 lang)

Mit der Taste 1 lange gedrückt kommen wir in das SETUP. Das SETUP hat 4 Punkte.

**Kalib. Takt int.** Takt intern kann einmal der TCXO 12,8 MHz sein oder es wird ein Takt am Stecker „OCXO“ angeschlossen. Der eingebaute TCXO 12,8 MHz schwingt natürlich nicht genau auf 12,800000 MHz, sondern etwas daneben. Die genaue Referenz-Takt-Frequenz wird hier eingestellt. Es gibt die Möglichkeit das per Hand zu erledigen oder mit *Taste 2 lang* die neue Referenz-Takt-Frequenz automatisch berechnen zu lassen und anschließend per Hand noch einen Feinabgleich zu machen. Ich beschreibe das noch weiter unten.

**Takt-Switch auto** Der Referenztakt des Frequenzmessers kann auf den intern Takt 12,8 MHz, oder auf den externen Takt geschaltet werden. Ist das BIT „Takt-Switch auto“ aktiv, wird am Eingang „externer Takt“ geprüft ob der Takt anliegt und vor Beginn der neuen Messung auf den externen Takt umgeschaltet.



**Edit Takt extern** In dieser Funktion wird die Frequenz des externen Taktes eingestellt. Ich habe den externen Takt auf 10,000000 MHz festgelegt. Die Frequenz kann aber verändert werden.

**Edit Takt intern** In dieser Funktion wird die Frequenz des internen Referenz-Taktes eingestellt. Default steht die Frequenz auf 12,8 MHz.

### 5.3.1 Kalib. Takt int.



*Referenztakt intern kalibrieren.*

In dieser Funktion wird die exakte Taktfrequenz der internen Referenztaktquelle eingestellt. Entweder die Frequenz des TXCO 12,8 MHz oder eine Taktquelle die am Stecker „OCXO“ eingespeist wird. Dazu wird am Messeingang eine sehr genaue Frequenz angeschlossen, von 10,000 000 MHz. Bei mir ist das mein GPS-Frequenznormal mit genau 10 000 000,00 Hz.

Am Anfang wird nach der Torzeit gefragt. Wir können auswählen 8, 16 oder 32 Sekunden. Wurde die Auswahl getroffen beginnt der Messzyklus. **Im ersten Messzyklus sind die Taste 1 bis 3 gesperrt.** Damit wird verhindert, dass mit „Taste 2 lang (Referenz-Frequenz automatisch neu berechnen)“ eine falsche Berechnung gestartet wird. Wir haben die Möglichkeit die interne Referenz-Frequenz per Hand zu verstellen oder erst einmal mit der *Taste 2 lang* die neue Referenz-Takt-Frequenz vorberechnen zu lassen. Anschließend kann man mit der Hand noch einen Feinabgleich durchführen.

Es folgt die Tabelle der Tastenbelegung:

Taste	Taste kurz gedrückt	Taste lang gedrückt
1	Kursor weiter schalten	interne Referenz-Frequenz speichern
2	Stelle am Kursor +	<b>Referenz-Frequenz automatisch neu berechnen</b>
3	Stelle am Kursor –	Funktion abbrechen

Die Berechnung der internen Takt-Frequenz geschieht mit folgender Formel:

$$int.Ref.Freq.NEU = \frac{10000000Hz}{gemesseneFrequenz} * int.Ref.Freq.ALT$$

Die interne Referenz-Takt-Frequenz kann beliebig sein. Wird der TXCO auf der Platine verwendet sind das etwa 12,8 MHz. Aber es ist auch eine andere Frequenz möglich, zum Beispiel 16,384 MHz. Die Frequenz sollte größer oder gleich 10 MHz sein. Das Ergebnis wird sofort im Display angezeigt. Bin ich damit zufrieden wird mit „Taste 1 lang“ die neue interne Referenz-Takt-Frequenz im Eeprom abgespeichert.

#### Beispiel für die Kalibrierung des Referenztaktes

Ich habe mein GPS-Frequenznormal an der Frequenzzähler angeschlossen.



Ich habe das Bild mit den Tasten-Nummern ergänzt.

Die Referenz-Takt-Frequenz habe ich genau auf 12 800 000 Hz eingestellt. Das Messergebnis zeigt aber, dass meine Messung etwa 6 Hz daneben liegt. Wir müssen den Referenztakt neu kalibrieren. Mit *Taste 1 lang* gehen wir in das SETUP.



Mit OK starten wir die Funktion.

Zuerst müssen wir die Torzeit der Messungen festlegen.



Meine Auswahl ist 16 Sekunden Torzeit. Das müsste ausreichend genau werden.

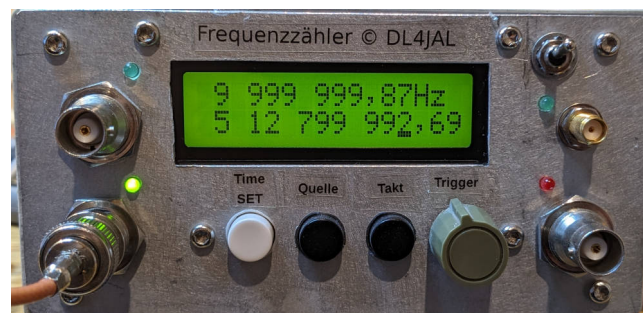
Der erste Messdurchlauf startet anschließend. Damit keine Falschbedienung erfolgt sind beim ersten Messdurchlauf die 3 Tasten gesperrt.



Der Referenztakt steht genau auf 12,8 MHz und müsste wahrscheinlich korrigiert werden. Wir warten die erste Messung ab.



Im Messergebnis sehen wir auch die Abweichung von etwa 6 Hz. In Zeile 2 steht die Referenz-Takt-Frequenz von 12 800 000 Hz. Als erstes Starten wir jetzt mit *Taste 2 lang* die automatische Berechnung der neuen Referenz-Takt-Frequenz.



Die automatische Berechnung ergibt eine neue Referenz-Takt-Frequenz von

12 799 992,69 Hz

Mit dieser neuen Referenz-Takt-Frequenz warten wir einen Messdurchlauf ab.



Die Messfrequenz ist jetzt 9 999 999,94 Hz. Wir liegt jetzt nur noch 0,06Hz daneben. Die Referenz-Takt-Frequenz muss ganz leicht erhöht werden.

Mit *Taste 1 kurz* bewegen wir den Cursor auf die hundertstel Stelle.

Mit *Taste 2 kurz* wird die *Feinabstimmung Referenz-Takt-Frequenz erhöhen* durchgeführt.

Das Messergebnis wird sofort durch Berechnung ungefähr angepasst. **Man muss aber anschließend immer eine Messung abwarten, ob die Änderung auch richtig ist.**



Ich habe die Taktfrequenz so lange in 0,01 Hz Schritten erhöht bis die gemessene Frequenz den 10 000 000,00 Hz sehr Nahe kommt. Diese neue Einstellung wird mit *Taste 1 lang* im Eeprom abgespeichert.



Die Neukalibrierung war erfolgreich. Die Messung mit Torzeit von 4 Sekunden liefert ein ziemlich genaues Messergebnis.

**Ich hoffe dieses ausführliches Beispiel hilft bei der Kalibrierung des Frequenzzählers.**

### 5.3.2 Takt-Switch auto



Diese Funktion aktiviert oder sperrt das automatische Umschalten der Taktreferenz intern oder extern. Ist das Bit gesetzt, wird nach jedem Messdurchlauf getestet ob der externe Takt 10 000 000 Hz angeschlossen wurde. Wenn ja, wird auf diese genauere Referenztaktquelle umgeschaltet.

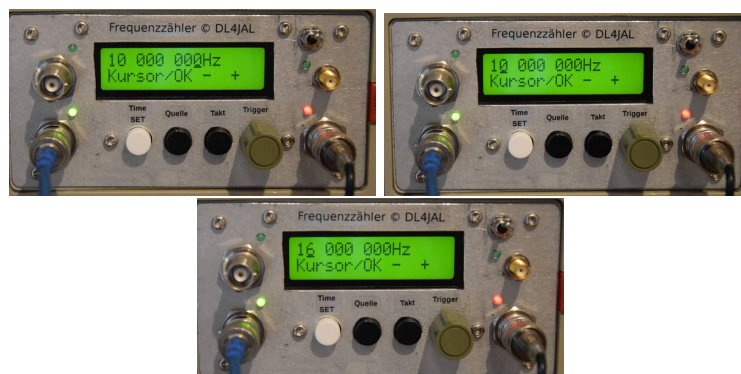


Auswahl BIT ein/aus.

### 5.3.3 Edit Takt extern



In dieser Funktion wird die Frequenz der externen Taktreferenz eingestellt. Mit „Taste 1 lang“ wird wieder der Wert im Eeprom abgespeichert.



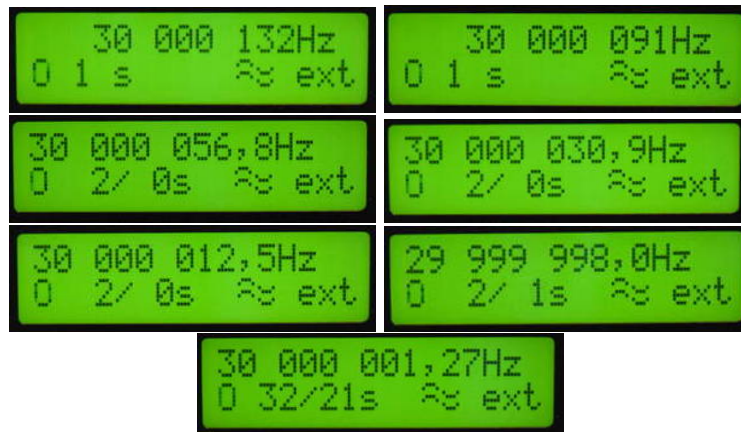
Kursor unter 1 Hz, Kursor unter 1 MHz; Externen Takt auf 16 MHz eingestellt.

## 5.4 Messbeispiele

Es folgen noch ein paar Messbeispiele mit dem Frequenzmesser.

### DDS-DoppelVFO, Frequenz nach Power ON

Ich habe mal meinen DDS-DoppelVFO überprüft, wie der Frequenzverlauf nach PowerON bei 30,000000 MHz ist. Ich habe den 400 MHz DDS-Taktgenerator in eine Thermostat eingebaut.

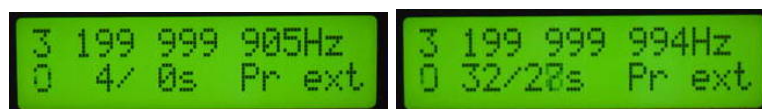


Bilder der Reihe nach PowerON, 15, 30, 45 Sekunden, 1, 2, 5 Minuten. Ich habe mit verschiedenen Torzeiten gemessen 1, 2, 32 Sekunden.

Nach etwa 5 Minuten nach dem Einschalten stimmt die Frequenz bis fast aufs Herz genau.

### Spektrumanalyser SIGLENT Tracking Generator

Ich habe einen Spektrumanalyser. Den Tracking Generator kann man als VFO benutzen. „Zero SPAN“ muss dabei eingestellt werden.



3,2 GHz mit verschiedenen Torzeiten gemessen



2,4 GHz mit verschiedenen Torzeiten gemessen

Der SIGLENT hat eine Anschluss für externen Takt 10,0 MHz. Da habe ich den Takt meines Frequenznormales eingespeist.



2,4 GHz ohne TG externer Takt und mit externen Takt. Mit gleicher Torzeit gemessen.

### Überprüfung der Netzfrequenz 50 Hz



Bei den Messungen mit Torzeit 32 Sekunden überwiegen die Messungen größer 50 Hz.

### 5.5 Serielle Ausgabe der Messwerte

## Kapitel 6

# Schlusswort

**Dieses Projekt darf nicht kommerziell vermarktet oder genutzt werden. Alle Rechte liegen bei DL4JAL (Andreas Lindenau).**

Ich wünsche viel Spaß beim Basteln.

vy 73 Andreas DL4JAL

✉ [dl4jal@t-online.de](mailto:dl4jal@t-online.de)

☎ 037291-68873

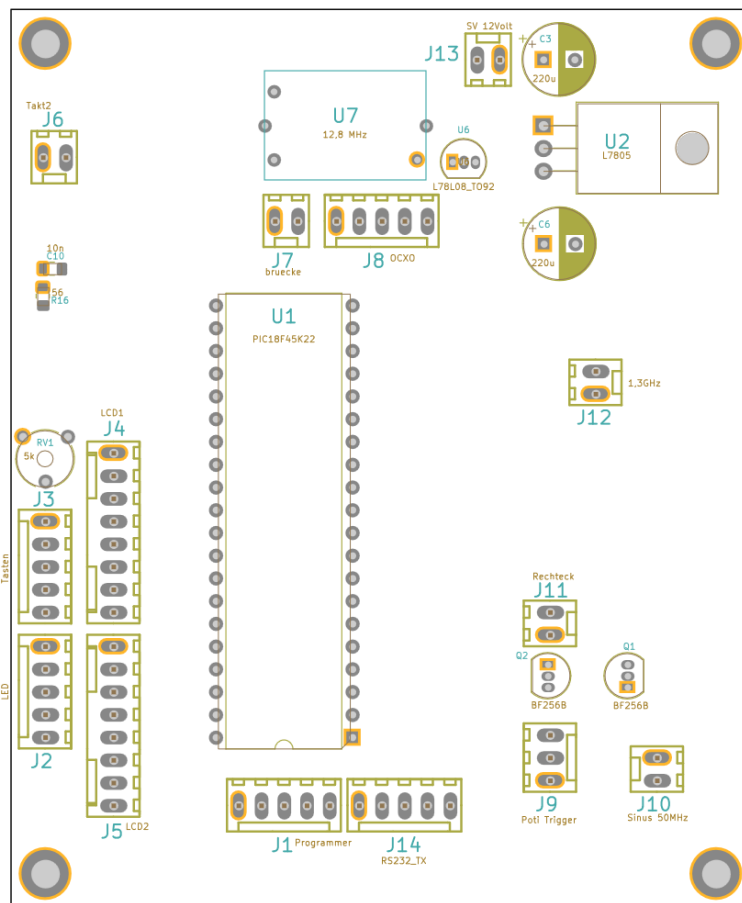


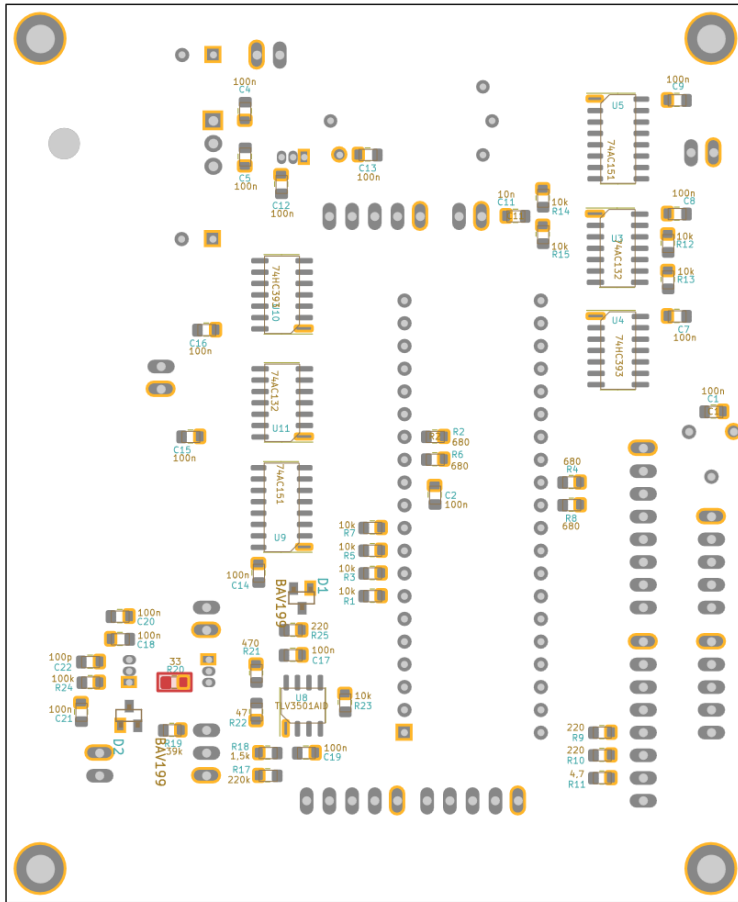
# Literaturverzeichnis

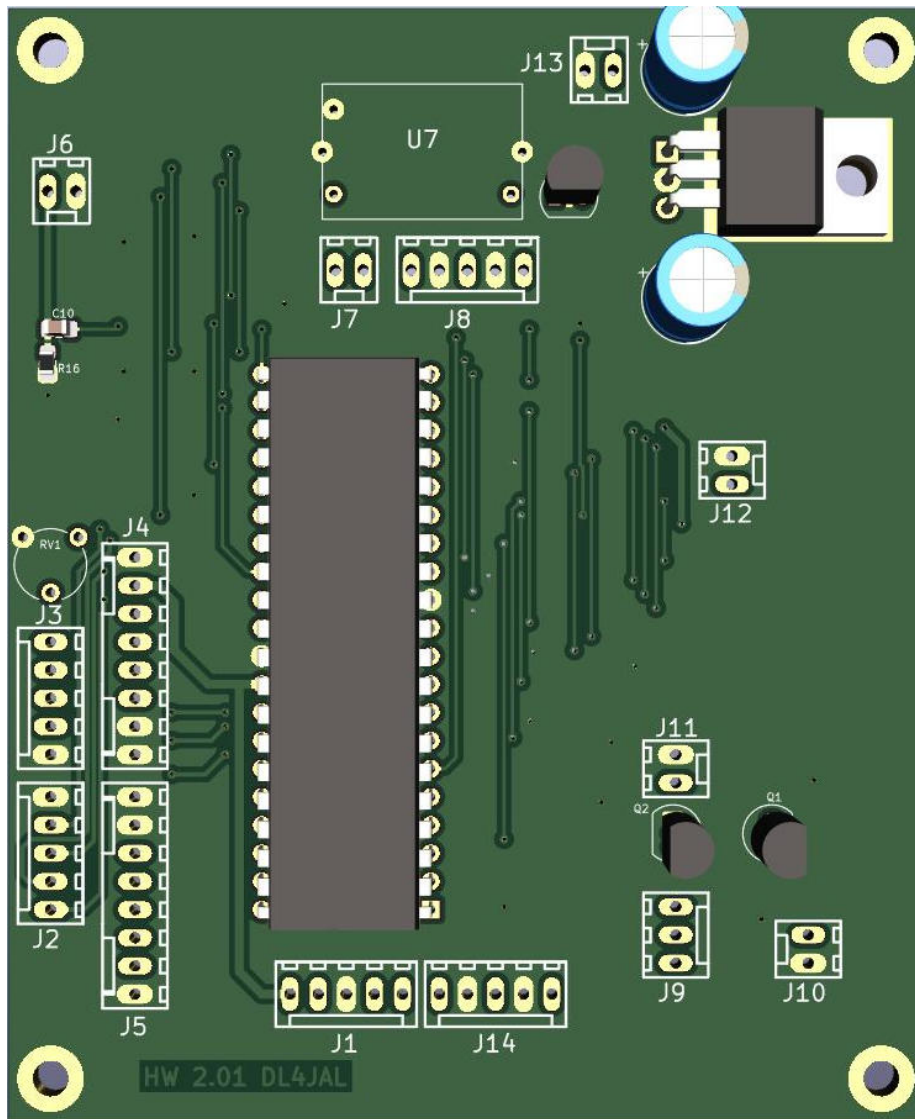
- [1] <http://www.leobodnar.com/shop/>
- [2] <http://www.dl4jal.eu/fnormal/fnormal.html>

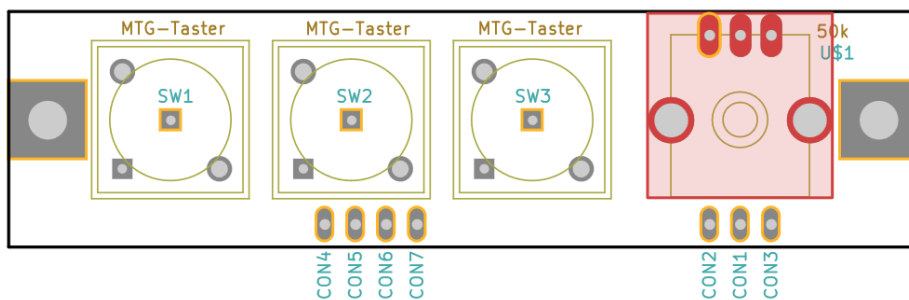
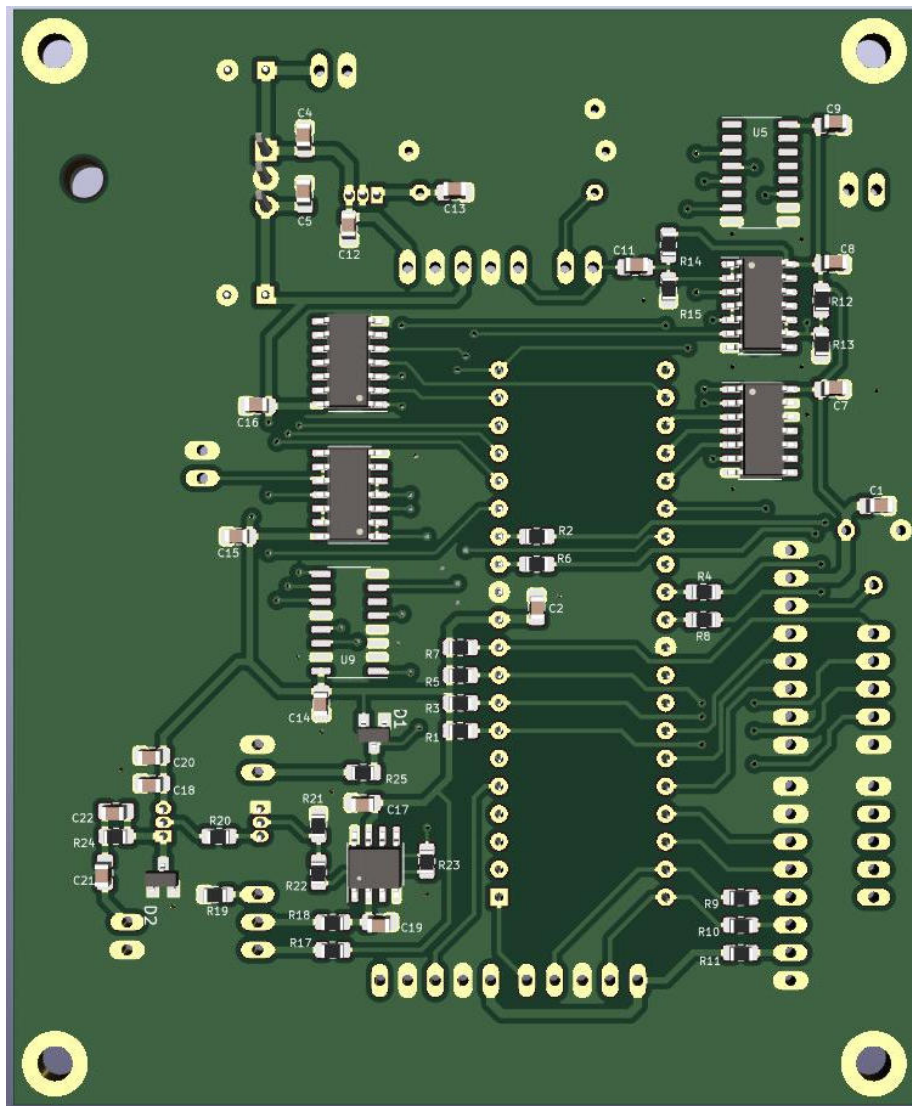
# Kapitel 7

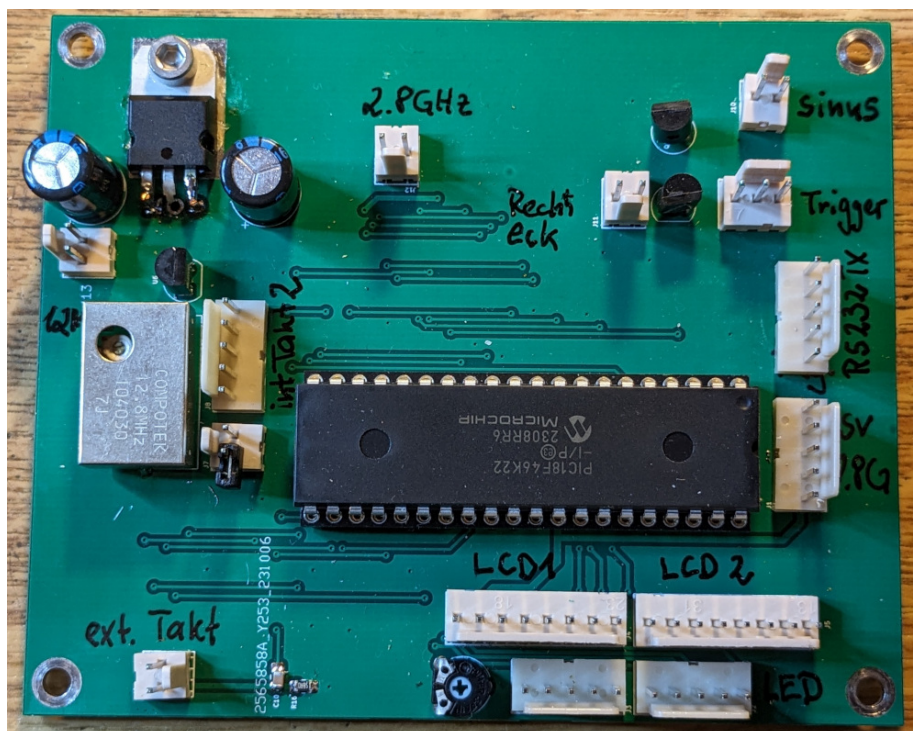
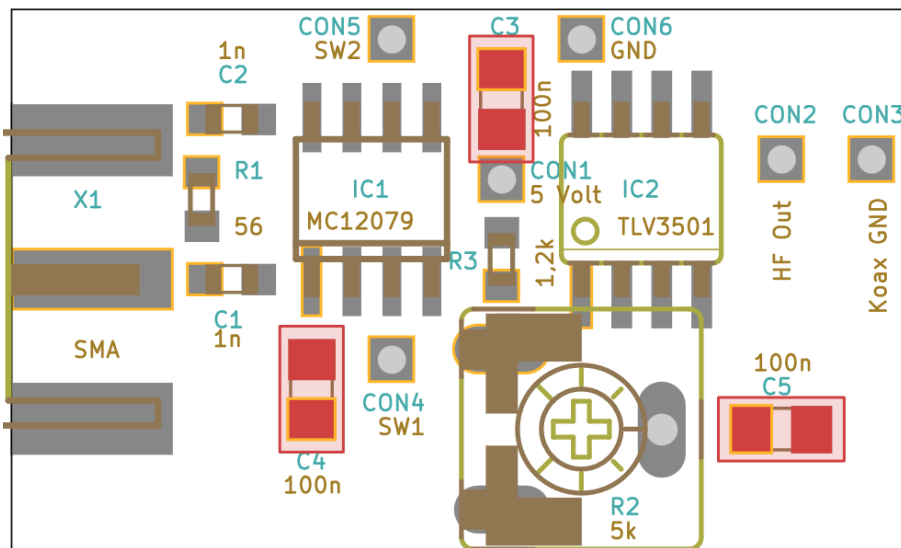
## Zusätzliche Bilder

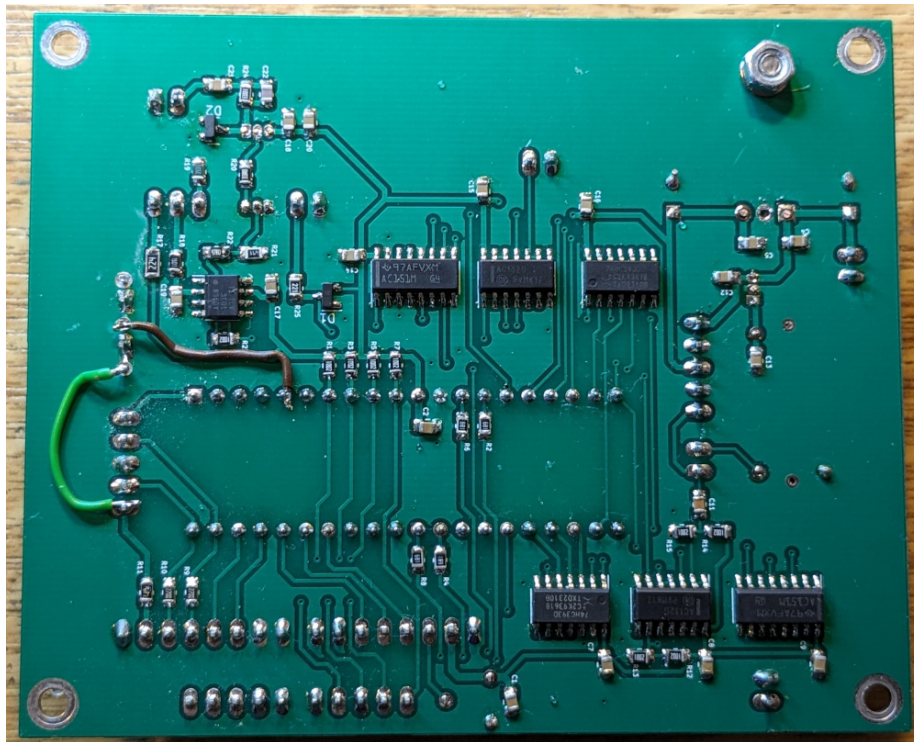










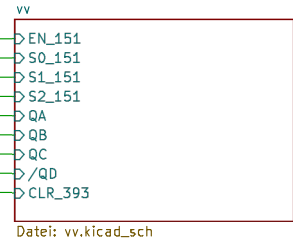
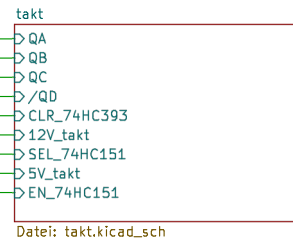
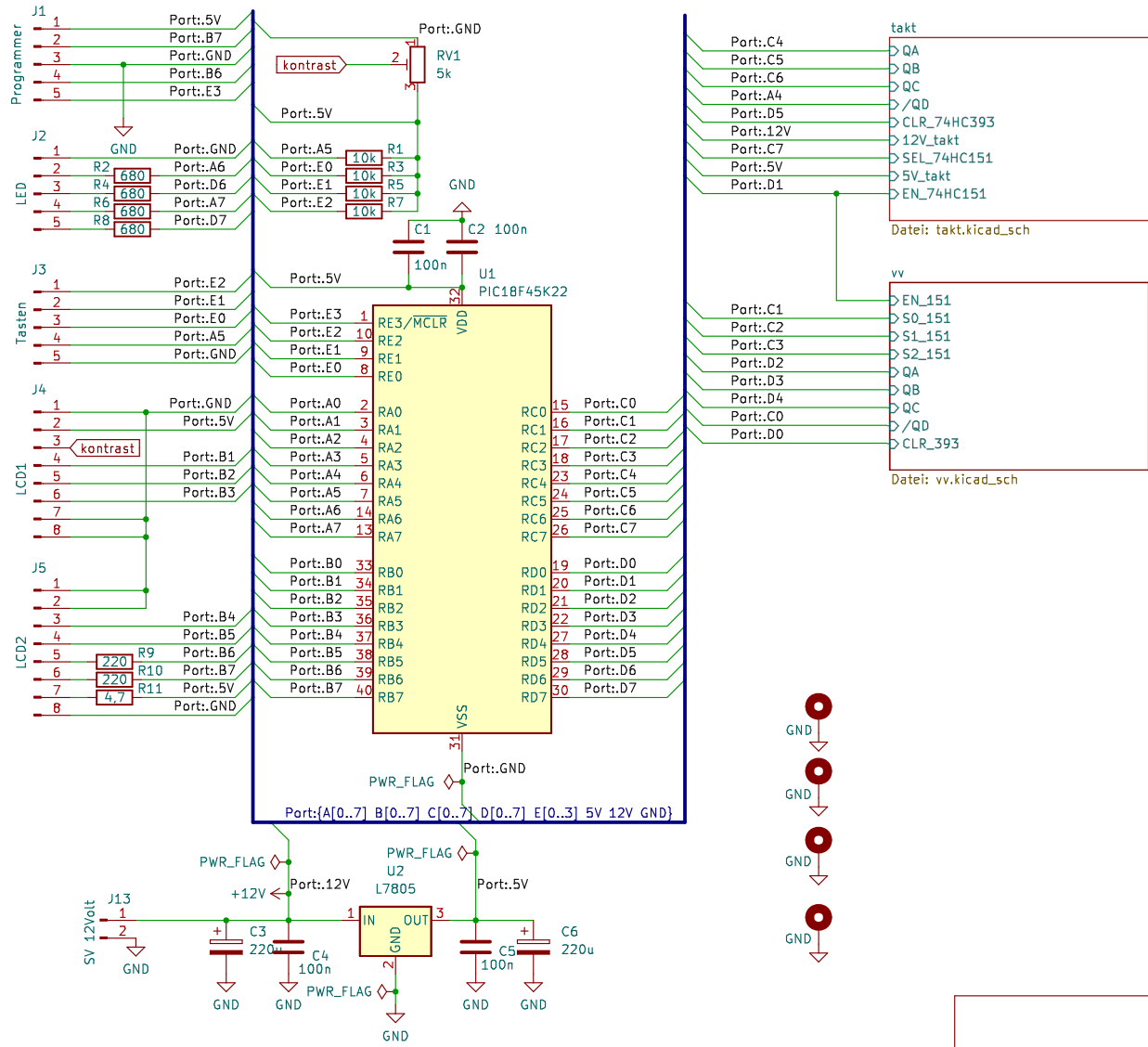


## Kapitel 8

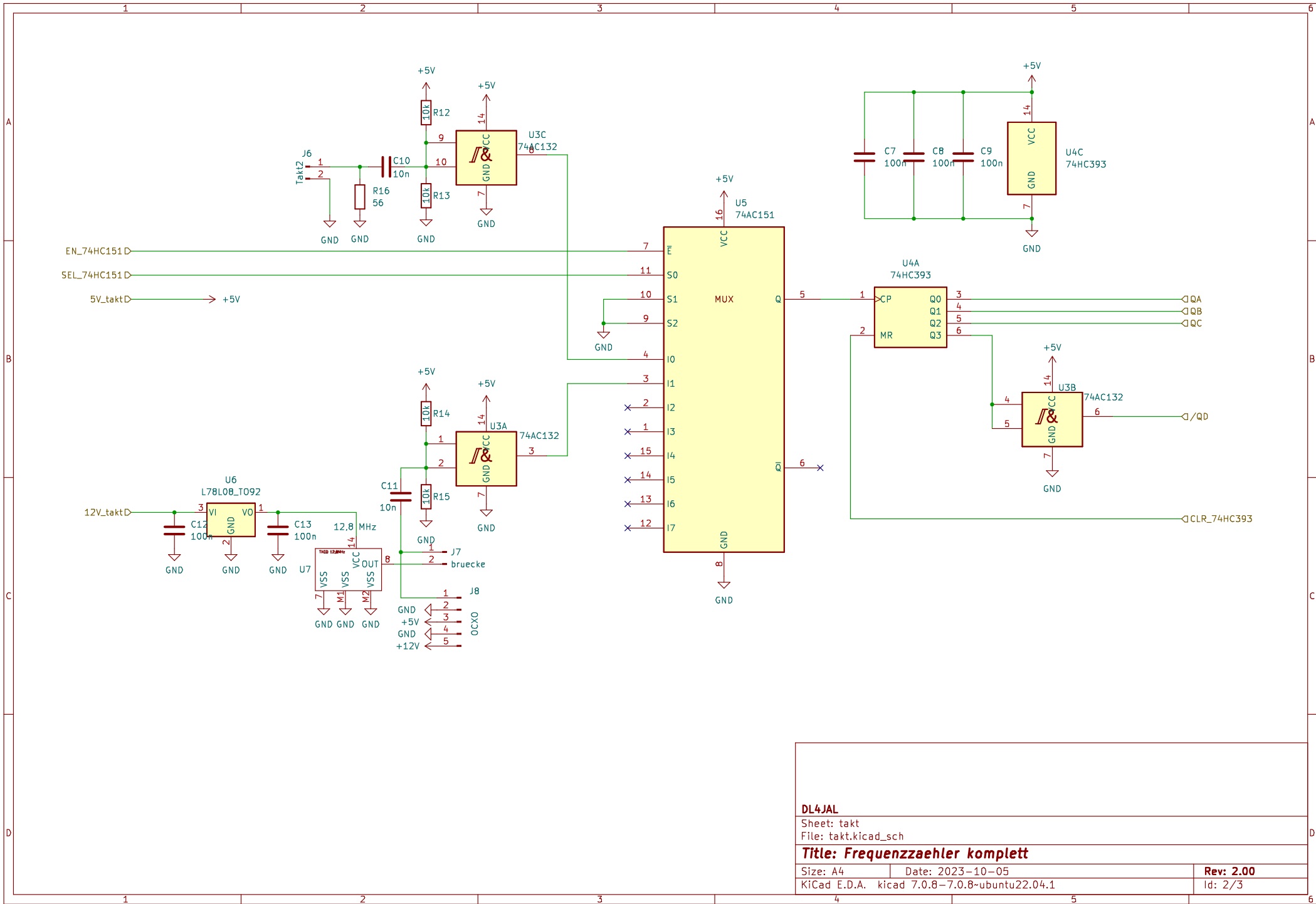
# Angehängte Dateien

Es folgen noch Dateien, die ich mit einem PDF-Tool angehängt habe.

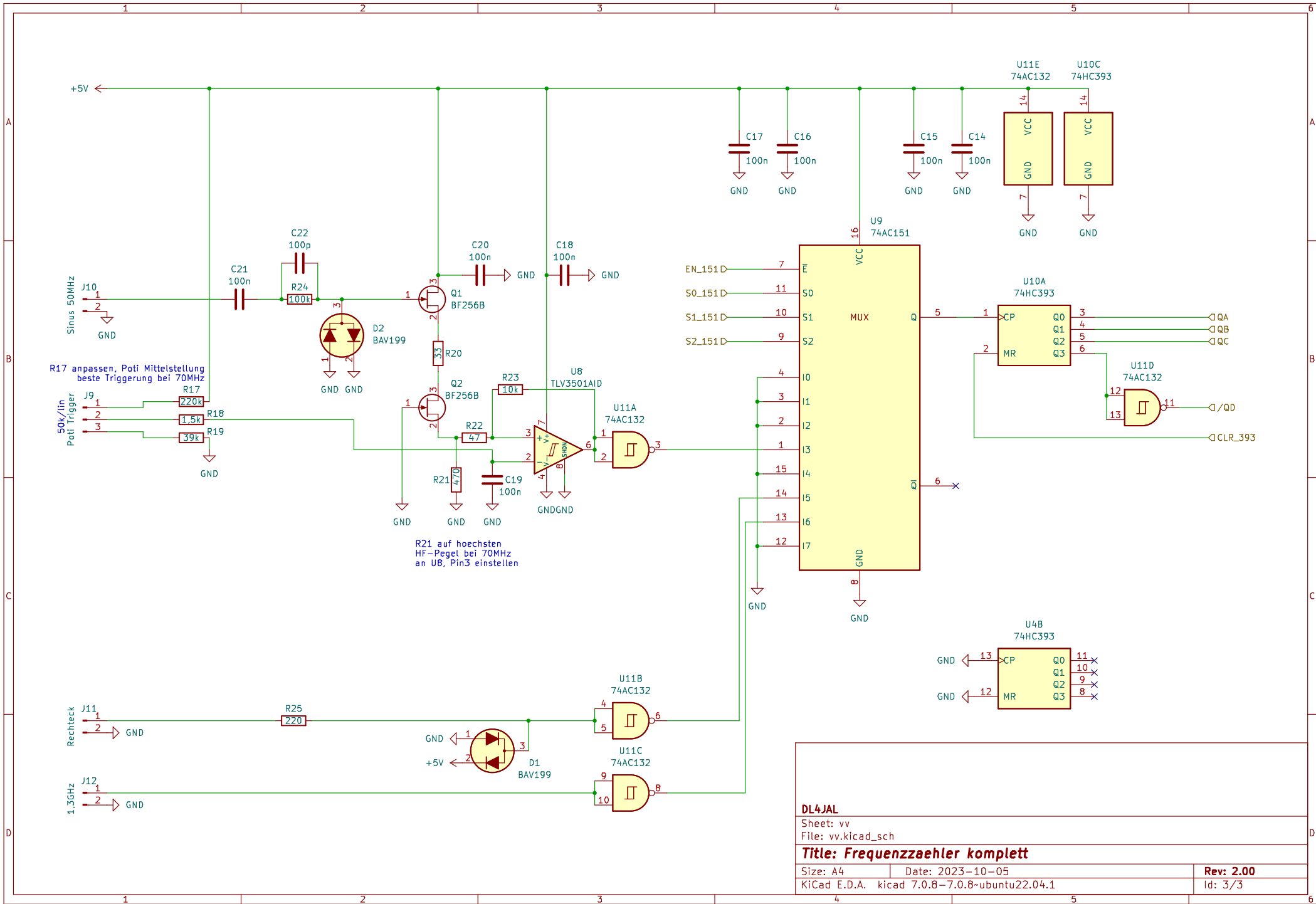




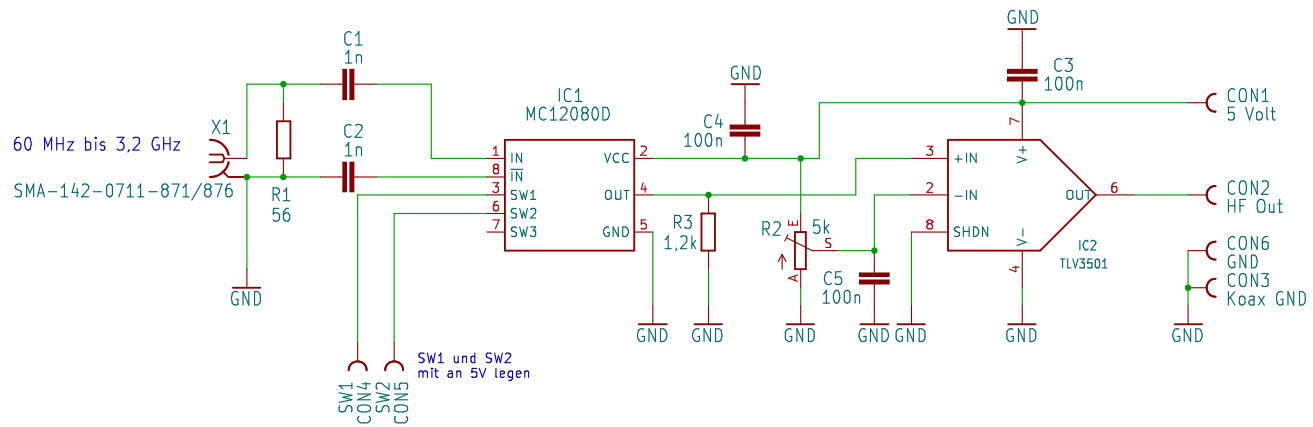
<b>DL4JAL</b>	
Sheet:	
File: freq_zaehl_rezi_kicad.kicad_sch	
<b>Title: Frequenzzaeher komplett</b>	
Size: A4	Date: 2023-10-05
KiCad E.D.A. kicad 7.0.8-7.0.8-ubuntu22.04.1	Rev: <b>2.00</b>
	Id: 1/3



<b>DL4JAL</b>	
Sheet: takt	
File: takt.kicad_sch	
<b>Title: Frequenzzaehler komplett</b>	
Size: A4	Date: 2023-10-05
KiCad E.D.A. kicad 7.0.8-7.0.8-ubuntu22.04.1	Rev: 2.00
	Id: 2/3



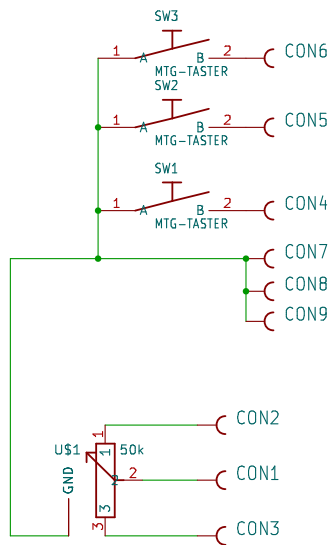
<b>DL4JAL</b>	
Sheet: vv	
File: vv.kicad_sch	
<b>Title: Frequenzzaeher komplett</b>	
Size: A4	Date: 2023-10-05
KiCad E.D.A. kicad 7.0.8-7.0.8-ubuntu22.04.1	Rev: 2.00
	Id: 3/3



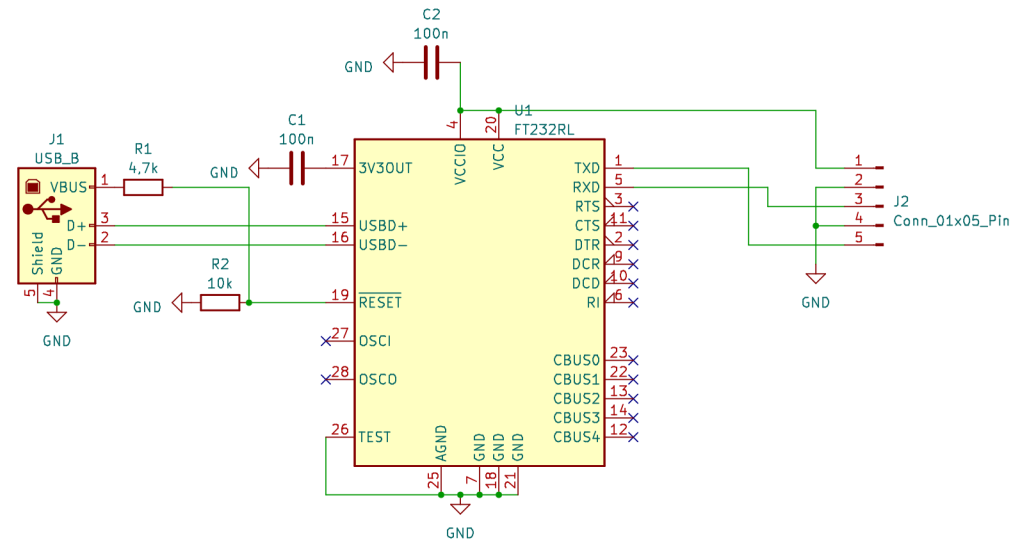
SW1 und SW2  
mit an 5V legen

SW1=High, SW2=High Teiler 64  
 SW1=High, SW2=Low Teiler 128  
 SW1=Low, SW2=High Teiler 128  
 SW1=Low, SW2=Low Teiler 256

TITLE: \${PROJECTNAME}	
Document Number:	REV:
Date: \${CURRENT_DATE}	Sheet: \${#}/\${##}



TITLE: \${PROJECTNAME}	
Document Number:	REV:
Date: \${CURRENT_DATE}	Sheet: \${#}/\${##}



zum Anschließen an die RemotePlatine 3.0

**DL4JAL**

Sheet: /

File: usb\_modul.kicad\_sch

**Title: USB Modul mit FT232RL**

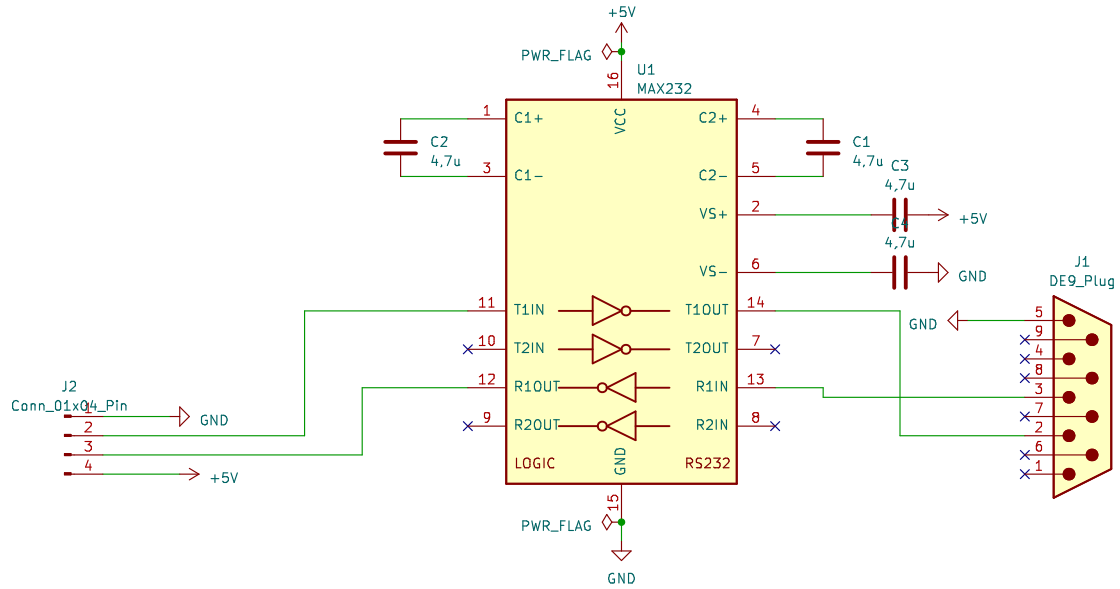
Size: A4

Date: 2023-03-12

Rev: 1.0

KiCad E.D.A. kicad 7.0.1-3b83917a11-171-ubuntu22.04.1

Id: 1/1



**DL4JAL**

Sheet: /  
 File: rs232modul.kicad\_sch

**Title: RS232 Modul Pegelwandlung**

Size: A5	Date: 2023-03-11	<b>Rev: 1.00</b>
KiCad E.D.A. kicad 7.0.0-da2b9df05c-171-ubuntu22.04.1		Id: 1/1