

Entwicklung einer Messeinrichtung zur Erfassung, Auswertung,
Speicherung und Visualisierung der Spannungsqualität

Abgabedatum: 25.05.2021

Ich versichere, dass ich die vorliegende Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe, und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Ort, Datum

Unterschrift

Inhaltsverzeichnis

Abbildungsverzeichnis	V
Tabellenverzeichnis	VII
Abstract	VIII
1 Einleitung	1
2 Grundlagen der Netzfrequenz	3
2.1 Notwendigkeit der Überwachung der Spannungsqualität	3
2.1.1 Netzfrequenz zur Indikation der Stabilität eines Netzes	3
2.2 Erlaubte Grenzwerte der DIN-Norm EN 50160	5
2.2.1 Kurvenform	5
2.2.2 Spannungshöhe	7
2.2.3 Frequenz	7
3 Entwicklung einer Messeinrichtung	8
3.1 Problematik des Transformators zur Trennung der Netzspannung	8
3.1.1 Besondere Anforderungen für berührungssicheres Design	14
3.2 Notwendige Bit-Tiefe der Datenauswertung	20
3.3 Notwendige Abtastrate	21
3.4 Analoge Filter zur Verbesserung der Genauigkeit der Auswertung	23
3.5 Genauigkeit des Zeitstempels	33
3.6 Entwicklung einer HAT-Platine für den Raspberry Pi	34
3.6.1 Spannungsversorgung	35
3.6.2 Analog/Digital-Umsetzung	38
3.6.3 Analogfilter	39
3.6.4 GPS	39
4 Programmierung der Software und Validierung	44
4.1 Programmierung	44
4.1.1 Auswertung der GPS Daten durch gpsd	44
4.1.2 Algorithmus zur Bestimmung der Frequenz	48

4.1.3	Oberschwingungsmessung	50
4.1.4	Abfragen der Spannungswerte des A/D-Umsetzers	51
4.2	Validieren der Messergebnisse	61
4.2.1	Prüfprotokoll zur Einhaltung der Anforderungen	61
4.2.2	Validieren des analogen Tiefpasses	61
4.2.3	Überprüfen des Rauschens am Eingang des Analog/Digital-Umsetzers	63
4.2.4	Genauigkeit des Zeitstempels	64
4.2.5	Korrektheit der Amplitude der Oberschwingungen	65
4.2.6	Auswertung der Netzfrequenz	66
4.2.7	Verbreitung der gemessenen Netzfrequenz durch das Internet	71
5	Zusammenfassung	73
A	C-Code des Raspberry Pi	74
B	Matlab Code	84
B.1	SNR-Bit-Tiefe-Simulation	84
B.2	Datenauswertung DT9816	86
B.3	Wrapper für die DT9816 zum Benutzen der func analyze	87
B.4	Funktion zur Detektion von Nullstellen mit Auswertung der Frequenz und Oberschwingungen	89
C	CAD-Zeichnungen	92
C.1	Schaltpläne	92
C.1.1	Isolationsboard	92
C.1.2	HAT-Platine	93
C.2	Platinenentflechtungen	95
C.2.1	Isolationsboard	95
C.2.2	HAT-Platine	96
D	Prüfprotokolle	98
D.1	DIN EN 61010 SK2,SK3	99
D.2	DIN VDE 0701-0702	101
D.3	DIN EN 60335 SK2,SK3	102
	Literatur	104

Abbildungsverzeichnis

2.1	Aussehen einer 50Hz Schwingung mit den maximal erlaubten Oberschwingungen	6
3.1	Ersatzschaltbild eines realen Transformators mit Kapazitäten	9
3.2	Grafische Darstellung eines Ausschnitts der Werte aus Tabelle 3.1	11
3.3	Hysteresekennlinie eines Transformators	12
3.4	FFT des Transformators bei einer Eingangsfrequenz von 50 Hz	13
3.5	Gegenüberstellung der Spannungsverläufe zwischen der Messung durch einen Tastkopf und einem Transformator	14
3.6	Schematische Kennlinien des SMD-Kondensatoren Herstellers Vishay für die SMD Größe 1603 [14]	17
3.7	Abbildung der Ober- und Unterseite der Isolationsplatine	19
3.8	Entflechtung der Isolationsplatine mit Bemaßung	19
3.9	Aufbau eines SAR-ADCs	23
3.10	Schematischer Aufbau einer Sallen-Key Stage	25
3.11	Gesamtbeschaltung des Vierfach-Operationsverstärkers	32
3.12	Abbildung der bestückten HAT-Platine	35
3.13	Schaltplan um den Connector des Raspberry Pi mit Glättungskondensatoren	36
3.14	Schaltplan der Beschaltung des LTC3265 Chip	38
3.15	Aufbau des Schaltkreises um den Analog/Digital-Umsetzers ADS8681	39
3.16	Fehlangepasste Leitung	42
3.17	Schaltplan der Beschaltung des GPS-RF Chips NEO-M8M der U-blox AG	43
4.1	Datenausgabe der Messung von zwei Perioden der Netzfrequenz durch den ADS8681	58
4.2	Nulldurchgang des Messinstruments DT9816-S bei 750 kHz Abtastrate	59
4.3	Nulldurchgang des Messinstruments DT9816-S bei 50 kHz Abtastrate	59
4.4	Übertragungsfunktion des Tiefpassfilters	62

4.5	Rauschspannung am Eingang des Analog/Digital-Umsetzers	63
4.6	Ausgabe des Status des NTP-Services durch <i>ntpq -p</i>	64
4.7	Frequenzauswertung bei einer steigenden Frequenz von 44 Hz bis 56 Hz, bei der jede Sekunde um 10 mHz gesteigert wird	67
4.8	Frequenzauswertung bei einer steigenden Frequenz von 49,75 Hz bis 50,25 Hz, bei der alle drei Sekunden um 1 mHz gesteigert wird	68
4.9	Aufzeichnung von Frequenzen bei benutzen eines Frequenzgenerators nach über drei Stunden Frequenzaufzeichnung	69
4.10	Histogramm der Messaufzeichnung	70
4.11	Skala der Netzfrequenzmessung mit einer Frequenz von 50,034 Hz	72
4.12	Visualisierung des Verlaufs der Netzfrequenz über 6 Stunden des Morgens des 29. März	72

Tabellenverzeichnis

2.1	Ausschnitt der Tabelle 1 der DIN EN 50160 Norm	6
3.1	Tabelle der Werte der Ausgangsspannung $A(f)$ des Transformators	10
4.1	Amplituden der Harmonischen der Netzspannung	66

Abstract

In this document, it is described how to measure, process, save and display the actual grid voltage waveform. The use of high quality interfaces, using no transformer makes it possible to analyse the frequency in a linear, low noise operation. Combined with a GPS-Timestamp, the frequency and harmonics will be saved. In this project, a single board computer *Raspberry Pi 4* is used as the computing and evaluation device. The focus of this work is designing, engineering and using an analog front end connected to an isolation amplifier that feeds several processing stages. This signal finally gets digitally converted and feeded as raw signal data into the *Raspberry Pi*. This isolational amplifier will dereference the mains voltage and makes sure that no living being can be harmed by the output of the isolational amplifier board. This amplifier board will be connected to another printed circuit board that holds several devices such as a GPS-Chip, an Analog-Digital-Converter and power supplies for the used operational amplifier for analog modification of the signal. The frequency and harmonics will be evaluated, and turned into a logging system to make this data visually available on a website. Using only the Raspberry Pi will lead to an eventual frequency accuracy loss that is not recoverable. Therefore a USB driven datarecorder is used as the digital evaluation device in order to ensure a valid measurement in a non-stop operation. This data is published online via a database connection to a webserver that caches the data and feeds all connected clients via HTTPS.

1 Einleitung

Die Netzfrequenz stellt in einem Gesamtnetzverbund wie dem europäischen Verbundsystem eine Größe dar, die in jedem Bereich und jedem Leistungserzeuger aufgrund ihres Verbundes, bis auf geringfügige lokale Differenzen, gleich groß ist. Dadurch, dass elektrische Leistung, die in einem Bereich des Netzes entnommen wird, an diesem oder an einem anderen Bereich erzeugt werden muss, wird in einem Netz keine Leistung gespeichert. Die Nachfrage der Leistung muss genau so groß sein, wie die aktuelle Erzeugung der Leistung in Kraftwerken. In dem Europäischen Verbundnetz ist, wie in den meisten Verbundnetzen eine Drehstrom-Hochspannungs-Übertragung dafür zuständig, Leistung über das Netz zu transportieren. Die Synchronmaschinen, die dort in dieses Netz ihre Energie einspeisen, laufen mit der aktuellen Netzfrequenz, da sie synchron zur Netzfrequenz drehen. Wird nun dem Netz mehr Leistung entnommen, fällt die Netzfrequenz. Wird zu wenig Leistung entnommen, steigt die Netzfrequenz bei bleibender Einspeisung. In dem Verbundnetz Kontinentaleuropas ist die nominale Frequenz 50Hz. Eine Abweichung dieser Netzfrequenz gibt in sehr guter Näherung an, wie viel Leistung gerade im Netz durch Regelleistung nachgeführt werden muss. Dies kann sowohl positive Regelleistung, also erhöhte Erzeugung von elektrischer Leistung aus Energiequellen, oder negativer, also die Verminderung von Erzeugung elektrischer Leistung, sein. Am 8. Januar 2021 ergab sich durch Abschaltens aufgrund von Überstrom eines Umspannwerks in Ernestinovo, Kroatien, eine Änderung der Lastflüsse im Gesamtnetz. In weniger als 20 Sekunden später lösten mehrere andere Überlastschutzsysteme aus und sorgten dafür, dass das Netz voneinander getrennt wird. Als Resultat war ein Netzsplit, womit die Nord-West-Seite von der Süd-Ost-Seite des Verbundnetzes voneinander getrennt wurden. Somit fiel die Netzfrequenz des Nord-Westlichen Teils aufgrund der Unterproduktion auf insgesamt 49,75Hz. Die Süd-Ost-Seite erreichte eine Netzfrequenz von 50,6Hz. Nach fast einer Stunde wurden diese zwei Teilnetze miteinander synchronisiert und wieder angekoppelt [3]. Durch Aufzeichnen der Netzfrequenz lassen sich nicht nur Fehler, sondern auch verschiedene zyklische Vorgänge im Netz anschaulich darstellen. In dieser Arbeit wird gezeigt, wie sich mit dem Einplatinencomputer *Raspberry Pi 4* durch Entwickeln von zwei Platinen eine Messschaltung erstellen lässt. In der einen Platine wird ein Isolationsverstärker im-

plementiert, der die Netzspannung in ein sicheres, verwendbares Maß herunterskaliert, die andere Platine modifiziert dieses Signal für die Verwendung eines Analog/Digital-Umsetzers. Durch diesen wird die Netzspannung als digitales Signal im *Raspberry Pi 4* zur Verfügung gestellt und eine Frequenzmessung und eine Amplitudenmessung, sowohl der Hauptamplitude, in dem europäischen Netz 230V Soll-Effektivwert, als auch die der ersten 25 Oberschwingungen, wird vorgenommen. Durch einen GPS-Zeitgeber werden diese Messdaten mit einem sehr genauen Zeitstempel sowohl auf einem festen Medium gespeichert, als auch in einer Datenbank für die Online-Auswertung hochgeladen und durch eine zeitliche Darstellung an verschiedene Clients via HTTPS visualisiert.

2 Grundlagen der Netzfrequenz

2.1 Notwendigkeit der Überwachung der Spannungsqualität

2.1.1 Netzfrequenz zur Indikation der Stabilität eines Netzes

Die Netzfrequenz ist in einem Verbundstromnetz wie dem Bereich des Netzverbands ENTSO-E, dessen gesamter Raum sich von 36 Ländern mit 15 EU-Ländern sowie 21 nicht-EU-Ländern erstreckt, weitgehend ein konstanter Wert. Dies bedeutet, dass kein geografischer Unterschied der Netzfrequenz existiert. Die Netzfrequenz ist eine Größe, die im idealen Falle immer konstant ist, da die Rotationsgeschwindigkeit von Synchrongeneratoren, wie derer in verschiedenen Kraftwerken, mit der aktuellen Frequenz mitlaufen. Dieses Übertragungsnetz ist hierbei jedoch kein Energiespeicher. Wenn an einem Ort Leistung benötigt wird, und so das Netz belastet wird, muss an einem anderen Ort im Netzraum die Leistung hierfür bereitgestellt werden. Durch die Erhöhung der Netzlast wird dementsprechend die wirkliche Frequenz des Netzes herabgesenkt. Durch das Herabsenken bei fehlender und Steigen der Frequenz bei einem Überschuss an Leistung im Netz sollte die Netzfrequenz durch eine Regelung stabilisiert werden. Hierfür existieren verschiedene Regelsysteme, die von den European Network of Transmission System Operators for Electricity (ENTSO-E) öffentlich gemacht werden. Die ENTSO-E ist ein Verband der Übertragungsnetzbetreiber, bei der eine Mitgliedschaft für jeden Betreiber Pflicht ist. Bei Netzen, die aus einem Verbund, im Vergleich zu einem Inselnetz bestehen, ist das Netz eher als starr anzunehmen, da deutlich mehr schwingende Massen und Leistungserzeuger verfügbar sind. Dies bedeutet, dass sich selbst bei stark steigendem Leistungsbedarf die Frequenz nur leicht ändert. Bei dem Verband wird von den Teilnehmern erwartet, wie diese Netzfrequenz zu stabilisieren ist. Hierbei sind insgesamt vier verschiedene Vorkehrungen nötig. In einem Teilbereich des Netzes ist die Charakteris-

tik des Leistungs-Frequenz-Verhältnisses als ein Quotient λ gegeben. Dies ist durch die folgende Gleichung definiert:

$$\lambda = \frac{\Delta P}{\Delta f} \quad \text{in} \quad \frac{\text{MW}}{\text{Hz}} \quad (2.1)$$

Hierbei beschreibt Δf den Unterschied zur gewollten Frequenz, also meist 50,000 Hz. Es kann manuell eine Änderung der Zielfrequenz vorgenommen werden. Der Faktor ΔP bezeichnet dann die fehlende Leistung bzw. den Überschuss der Leistung. Ist die Differenz zur Sollfrequenz größer als 20 mHz, wird durch die Regelung zuerst Primärregelleistung durch die Generatoren der Energieerzeuger bereitgestellt. Dies geschieht nach spätestens einigen Sekunden, wobei die Sekundärregelleistung nach etwa 15 Sekunden einzusetzen beginnt. Diese Sekundärregelung hat zur Aufgabe, innerhalb des geografischen Bereichs die Leistungsverteilung so zu regeln, dass sie wieder dem nominalen Wert, für die das Netz ausgelegt ist, entspricht. Dadurch soll die Balance der Teilbereiche des Gesamtnetzes ausgeglichen werden und die Primärregelleistung kann, wie zuvor ausgelegt, benutzt werden. Die Änderung der Leistungsnachfrage eines Teilbereichs ist im normalen Betrieb vorgesehen, da selbst trotz Vorhersagen nie davon ausgegangen werden kann, dass der Leistungsfluss von einem Teilbereich des Gesamtnetzes zu den nächsten anliegenden Bereichen konstant bleibt. Die Minutenreserve, dementsprechend die Tertiärregelung hat die Aufgabe, durch Vorhersagen und Simulieren die Sekundärregelleistung so zu dimensionieren, dass sie die richtige Leistung bereitstellen kann. Sie sorgt für ein langfristiges Bereitstellen von Leistung über einen größeren Zeitraum. Die letzte Regeleinheit beschreibt die der Zeitkontrolle. Durch das Netz kann die aktuelle Zeit bestimmt werden, da Geräte wie z.B. Küchengeräte, die keine Zeitgeber intern eingebaut haben, durch die Anzahl der Nulldurchgänge die Dauer seit dem letzten Einstellen der Zeit zählen können. Ist die Netzfrequenz im Schnitt über einen gewissen Zeitraum nicht im Mittelwert genau 50,000 Hz, so läuft die Netzzeit, die durch die Nulldurchgänge abgeleitet wird, nach oder vor. Diese Zeitdifferenz zur echten Zeit gibt eine gute Übersicht der Qualität der Netzfrequenz des Gesamtnetzes an. Falls diese Netzzeit mehr als 30 Sekunden der echten Zeit nach- oder vorläuft, wird die Nennfrequenz des Gesamtnetzes auf 50,01 Hz respektive 49,99 Hz gesetzt. Ist die Netzzeit vorlaufend, wird die Netzfrequenz reduziert, ist sie nachlaufend, so wird sie vom nominalen Punkt erhöht. Sollte die Frequenz einmal unter 49,8 Hz fallen, so sollten alle schnell startenden Kraftwerke an das Netz angekoppelt werden. Sinkt die Netzfrequenz unter 47,5 Hz oder steigt sie über 52,5 Hz werden alle Kraftwerke von dem Netz instantan abgekoppelt und somit wird ein Ausfall des Stromnetzes unab-

dingbar. Um diesen Zustand gut vermeiden zu können wird bei einer Unterfrequenz ab 49,0 Hz ein Lastabwurf eingeleitet. Dieser Lastabwurf nimmt bestimmte Bereiche oder große Verbraucher vom Netz und sorgt somit dafür, dass die entnommene Leistung im Netz geringer wird. Dadurch soll die Netzfrequenz wieder normalisiert werden [2].

2.2 Erlaubte Grenzwerte der DIN-Norm EN 50160

Unabhängig der ENTSO-E werden verschiedene Bedingungen für die Auslieferung von elektrischer Energie durch das Stromnetz an Endkunden festgelegt. Die Norm DIN EN 50160:2011-02 mit dem Titel *Merkmale der Spannung in öffentlichen Elektrizitätsversorgungsnetzen* schreibt vor, welche Merkmale die von dem Netzbetreiber gelieferte Netzspannung an der Übergabestelle, also z.B. dem Hausanschluss, aufweisen darf. Diese Merkmale beschreiben den Normalfall, spezifizieren die Ausnahmesituationen wie Umweltkatastrophen oder Störungen durch Dritte nicht. Sie gelten aber dennoch bei Lastschwankungen und jeglichem normalen Betriebszeitpunkt. Die Norm bezeichnet die Merkmale der Kurvenform, Frequenz, und Höhe der Netzspannung, die der Nutzer an Nieder- oder Mittelspannung zu erwarten hat. Diese Merkmale unterscheiden sich bei beiden Spannungsniveaus. In der Norm ist die Symmetrie der Außenleiter zueinander als Kriterium mit aufgenommen. In dieser Arbeit beläuft sich jedoch der Aspekt auf das einphasige Niederspannungsnetz, sodass man die Messung im Allgemeinen in jeglichem Ort in Europa durchführen kann. Daher wird die Symmetrie des Netzes nicht mit untersucht.

2.2.1 Kurvenform

Oberschwingungen

Die Größe der Oberschwingungen, auch Harmonische Spannungen, ist durch die Tabelle 1 der DIN EN 50160 bestimmt. Diese wurde in Tabelle 2.1 in diesem Dokument vereinfacht übernommen. Sie beschreibt im ersten Schritt einen maximalen Gesamtoberschwingungsgehalt THD von unter 8% der ersten vierzig Oberschwingungen. Folgend wird definiert, wie groß die Amplitude der ersten 25 Oberschwingungen, also von 100 Hz bis 1250 Hz maximal sein darf im Vergleich zur Grundschwingung der Netzspannung. In der Norm

des Jahres 2011 stand aktuell zur Diskussion, wie mit Zwischenharmonischen verfahren wird, da diese durch Frequenzumrichter und ähnliche Geräte steigen. Dazu existiert jedoch noch keine normative Vorgabe.

Frequenz in Hz	100	150	200	250	300	350	400	450
Amplitude in %	2,0	5,0	1,0	6,0	0,5	5,0	0,5	1,5
Frequenz in Hz	500	550	600	650	700	750	800	850
Amplitude in %	0,5	3,5	0,5	3,0	0,5	0,5	0,5	2,0
Frequenz in Hz	900	950	1050	1100	1150	1200	1250	
Amplitude in %	1,5	0,5	0,5	0,5	1,5	0,5	1,5	

Tabelle 2.1: Ausschnitt der Tabelle 1 der DIN EN 50160 Norm

Würde man nun alle diese maximalen Oberschwingungen auf einen Sinus mit der Amplitude 1 addieren und jede Oberschwingung ohne Phasenversatz betrachten, dann sieht die Kurvenform des Signals mit allen Oberschwingungsgehalten wie in Abbildung 2.1 aus. Eine Addition von Oberschwingungen auf eine Grundschwingung ist jedoch immer periodisch und auch symmetrisch. Die echte Netzform kann sich aufgrund ihrer variablen Natur ändern und ist nicht fest an eine Symmetrie oder Periodizität gebunden.

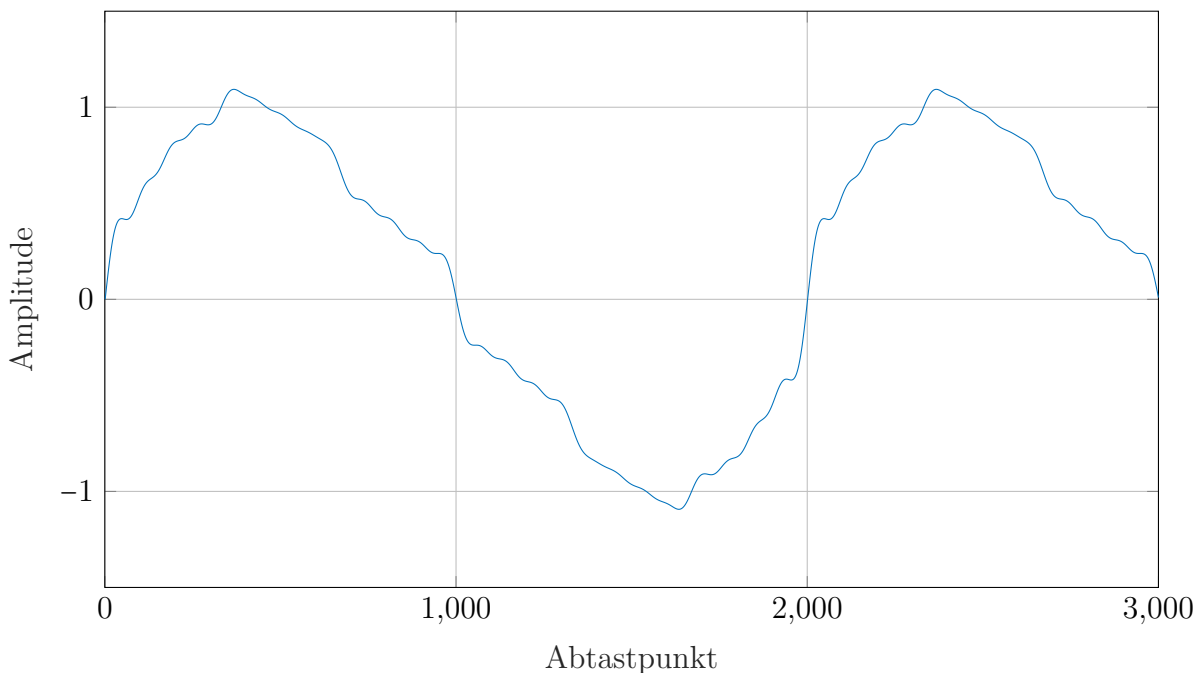


Abbildung 2.1: Aussehen einer 50Hz Schwingung mit den maximal erlaubten Oberschwingungen

2.2.2 Spannungshöhe

Amplitude der Netzspannung

Die Netzspannung muss sehr genau in einem Intervall liegen. Es müssen zum einen 95% der 10-Minuten-Mittelwerte des Effektivwertes U_n jedes Wochenintervalls nicht den Bereich von $U_n \pm 10\%$ über- oder unterschreiten. Zum anderen müssen alle 10-Minuten-Mittelwerte innerhalb von $U_n + 10\% / - 15\%$ liegen. In dem Fall der einphasigen Niederspannung ist U_n exakt 230 V zwischen dem Außenleiter und Neutralleiter, falls vier Leiter benutzt werden. Die Überspannung ist bei einer zehnpromzentigen Erhöhung der Fall, Unterspannung bei einem Abfall der Spannung um 10%. Es wird von Unterbrechung der Versorgung gesprochen, wenn weniger als 5% der Bezugsspannung an dem Übergabepunkt zum Kunden verfügbar sind.

Flicker

Das Flimmern, der sogenannte Flicker, beschreibt kurzzeitige Änderung der Netzspannung, bei der schnell-schaltende Leuchten ein Flimmern, also ein störendes, flackerndes, visuelles Signal ausgeben. Dies kann für einige Personen als störend erachtet werden, zumal es durch Wachsen der Amplituden-Änderung rapid störender wird. Die Flickerstärke wird in zwei Bereiche unterteilt: einmal der Kurzzeit- und einmal der Langzeit-Flickerstärke. Diese werden in dem Intervall von zehn bzw. zwei Stunden ausgewertet. Die Langzeit-Flickerstärke ergibt sich aus den letzten zwölf Werten der Kurzzeit-Flickerstärke. Die Flickerstärke lässt sich durch Betrachten der Amplitude ermitteln.

2.2.3 Frequenz

Die Frequenz des Netzes wird durch zwei Bedingungen eingegrenzt. Zum einen muss zu 100% der Zeit die Netzfrequenz zwischen 47 bis 52 Hz sein. Zudem muss zu 99,5% der Zeit eines Jahres eine Netzfrequenz von $50\text{Hz} \pm 1\%$ gelten. Dies ist jedoch rein die normative Begrenzung der Netzfrequenz. Bei einer Inselanbindung des Netzes sind diese Intervalle, in der sich die Netzfrequenz befinden muss, mit einer größeren Breite versehen.

3 Entwicklung einer Messeinrichtung

3.1 Problematik des Transformators zur Trennung der Netzspannung

Um die Netzspannung zu analysieren braucht es zu Beginn eine Möglichkeit, die Spannung für übliche Analysationsmöglichkeiten zugänglich zu machen. In dieser Arbeit findet vornehmlich eine digitale Auswertung der Netzspannung statt. Somit muss dafür gesorgt werden, die Netzspannung mit einer Amplitude von 325 V zu einem Wert von ca. 10 V linear und sicher umzuwandeln. Sicher bedeutet, dass die Schutzklasse 3, somit eine verstärkte Isolierung, eingehalten werden muss. Es darf niemals eine Möglichkeit existieren, die Netzspannung, selbst im Fehlerfall, auf eine Person oder Gegenstände einzukoppeln. Ein Trenntransformator ist ein gutes Mittel, eine Spannung durch ein magnetisches Feld, zu einer anderen Spannung umzusetzen. Durch die verwendeten Windungszahlen der Primärspule (n_1) und der Sekundärspule (n_2) ist eine Veränderung der Spannung durch das Verhältnis $\ddot{u}$ bei einem idealen Transformator wie folgt gegeben:

$$\frac{U_1}{U_2} = \frac{n_1}{n_2} = \ddot{u} \quad (3.1)$$

$$U_2 = \frac{n_2}{n_1} \cdot U_1 \quad (3.2)$$

Somit lässt sich durch die Variation der Windungszahlen die Spannung linear auf ein andere Amplitude skalieren, so wie es ein Widerstandsteiler vollbringt, jedoch mit der Garantie, dass bei einem Defekt eines Widerstands, die Spannung am Ende niemals der Netzspannung entspricht und keine Gefahr für Leben entsteht. Bei Benutzen eines Transformator kann, im Gegensatz zum Spannungsteiler mit Widerständen, kein Gleichspannungsanteil mit übertragen werden. Dadurch, dass Primär- und Sekundärspule keine direkte Verbindung miteinander eingehen, ist eine Isolation von der Netzseite und der

Ausgangsseite gegeben. Solange dieser Transformator kein sog. Spartransformator ist, findet somit eine einfache Isolation statt. Ist dieser Transformator durch zwei Schichten isoliert, sodass insgesamt zweifache Isolation existiert und damit die Anforderungen eines Sicherheitstransformators nach DIN EN 61558-2-6 erfüllt, so ist dieser nach der Schutzklasse 3 zertifiziert. Das Benutzen eines Transformators der Schutzklasse 3, wie sie im Einzelhandel ohne Hürde erhältlich sind, genügt den Sicherheitsanforderungen, dass eine Berührung in keinem Falle zu einer Gefahr für den Menschen enden kann. Dies gilt aber nur, wenn die darauf folgende Messkette diese Sicherheit nicht stört. Trotz der erhöhten Sicherheit gibt es aufgrund der Benutzung eines Transformators verschiedene ungewollte Verhalten, wie ein Tiefpassverhalten, was aus der Übertragung durch Spulen innerhalb des Kreises entsteht und dazu existiert eine Nichtlinearität, die durch den Transformator kern eintritt. Das T-Ersatzschaltbild eines Transformators wie in Abbildung 3.1 bietet sich an, um die Wirkung des Tiefpasses zu erläutern. Dieses Ersatzschaltbild versucht alle Nichtidealitäten eines Transformators, bis auf die Nichtlinearität, in Einklang mit einem idealen Übertrager zu bringen.

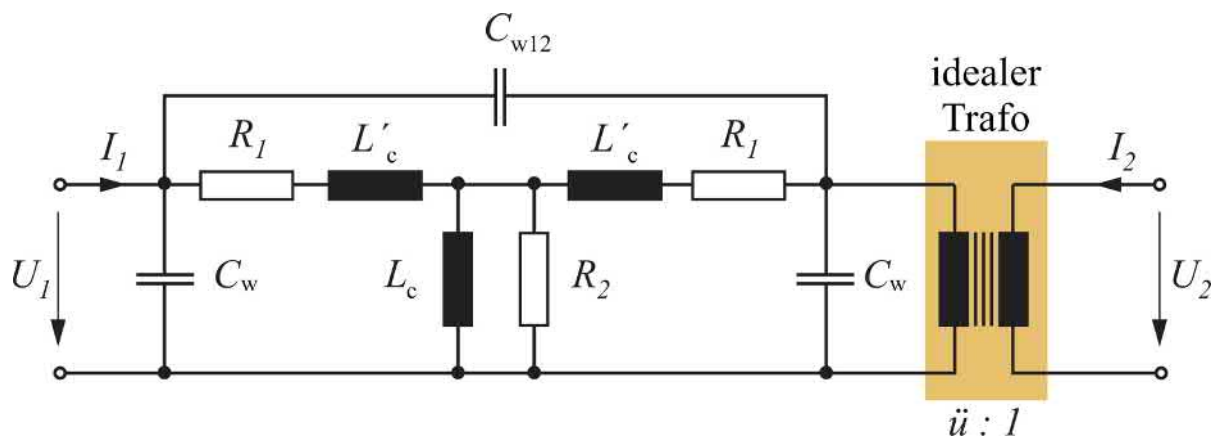


Abbildung 3.1: Ersatzschaltbild eines realen Transformators mit Kapazitäten [15]

- R_1 : Verluste in der Drahtwicklung
- R_2 : Kernverluste
- L_c : Hauptinduktivität
- C_w : Wicklungskapazität
- C_{w12} : kapazitive Verkopplung der Wicklungen

Ein realer Transformator besitzt aufgrund seiner Eigenschaften verschiedene Nichtidealitäten. Jede Spule eines Transformators ist aus einem Metall, meist Kupfer, gewickelt. Das hat zur Folge, dass dort ein Widerstand R_1 anliegt. An den Eingängen und Ausgängen des Transformators findet sich eine Kapazität C_w , die zwischen den beiden Anschlüssen des Eingangs bzw. Ausgangs, sowie eine Kapazität C_{w12} , die zwischen Eingang und Ausgang liegt. Dazwischen wirkt bei der Primärseite und bei der Sekundärseite wie gewollt eine

Spule, die die Wicklung um den Transformator Kern bezeichnet. Bei diesem Ersatzschaltbild wird deutlich, dass sich der Transformator wie ein Tiefpass zwischen Eingangs- und Ausgangsseite verhält. Es existieren aufgrund der Spulen in Reihe zu einem Widerstand ein R-L-Glied, welches ein frequenzabhängiges Element ist. Durch die Spule, deren Impedanz linear mit der Frequenz steigt, wird die Spannung bei höheren Frequenzen gedämpft. Durch eine Rücktransformation an die Eingänge des idealen Transformators des Ersatzschaltbilds wird durch das Verhältnis $\ddot{u}$ ein Widerstand $R_{\text{sekundär}}$ von der Sekundärseite des Trafos an die Primärseite durch die Bedingung des idealen Transformators transformiert. Somit erhält man den Widerstand zur Bestimmung der Übertragungsfunktion an der Primärseite und es kann eine gesamte Übertragungsfunktion des Transformators gebildet werden. Hierbei ist jedoch zu beachten, dass die Rückwirkungsfreiheit nicht gegeben ist, was bedeutet, dass man hierbei nicht das ESB in zwei Teile aufteilen kann und die Gesamtübertragungsfunktion des Übertragers durch Multiplizieren erlangen kann.

$$R_{\text{primär}} = \ddot{u}^2 \cdot R_{\text{sekundär}}$$

Das Messsignal durchläuft damit zwei Tiefpässe und es kann nicht mehr davon ausgegangen werden, dass das Messsignal, obwohl es einen sehr tiefen Frequenzbereich besitzt, fehlerfrei übertragen werden kann.

Ein Beispiel eines Trenntransformators ist der handliche Transformator der Firma Conrad Electronic mit der Modellnummer EA5420120167G. Dieser ist für eine Primärspannung von 230 V ausgelegt und gibt sekundärseitig eine Spannung von 12 V AC aus. Durch seine Schutzklasse eignet sich dieser Transformator auch als Trennelement. Um bei diesem das Tiefpassverhalten zu bestimmen wird nun auf dessen Primärseite ein Funktionsgenerator 33611A der Firma Keysight angelegt. Dieser gibt eine AC-Amplitude von maximal 5 V aus, von welcher die Frequenz verändert werden kann. Durch ein Messen der Amplitude der Sekundärseite des Transformators lässt sich so das Übertragungsverhalten ermitteln. In Tabelle 3.1 sind die Messwerte erfasst und werden danach in der Abbildung 3.2 ausgegeben.

f	50 Hz	500 Hz	5000 Hz	50000 Hz
$A(f)$	419 mV	416 mV	406 mV	382 mV
f	25 Hz	250 Hz	2500 Hz	25000 Hz
$A(f)$	416 mV	418 mV	410 mV	388 mV
f	12,5 Hz	125 Hz	1250 Hz	12500 Hz
$A(f)$	309 mV	424 mV	413 mV	395 mV

Tabelle 3.1: Tabelle der Werte der Ausgangsspannung $A(f)$ des Transformators

In diesen Messreihen wurde die Frequenz so gewählt, dass immer um eine Dekade erhöht wird, das bedeutet, dass immer die Frequenz nachfolgend um den Faktor 10 erhöht wird. Die Anfangswerte wurden von einem ganzzahligen Teiler der Frequenz 50Hz bezogen.

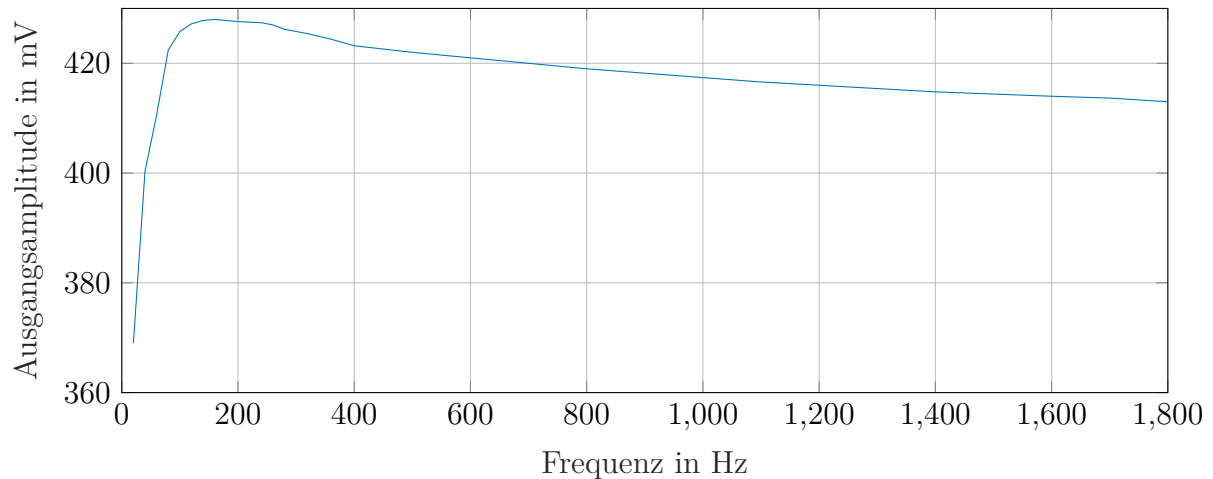


Abbildung 3.2: Grafische Darstellung eines Ausschnitts der Werte aus Tabelle 3.1

Durch diesen Transformator wird die oberste gemessene Schwingung, also in diesem Fall die 25. Harmonische, um den Faktor $1 - \frac{A(f=1250\text{Hz})}{A(f=50\text{Hz})} = 1 - \frac{413\text{mV}}{419\text{mV}} = 0,0143$, also um 1,43% reduziert. Dies bedeutet, dass aufgrund des Tiefpassverhaltens die Messung danach zu stark verändert wird, um die Messung mit einem Transformator gut zu vollziehen. Dies ist jedoch eine Betrachtung bei einer Linearisierung des Arbeitspunktes eines Transformators. Ein Transformator ist jedoch keine günstige Wahl für eine lineare Übertragung, da dieser resultierend aus der Hysteresekennlinie eines Übertragers eine nicht-lineare Übertragung darstellt. Die Hysterese ist eine Kennlinie, die die Magnetisierung des Eisenkerns in Verbindung der anliegenden magnetischen Feldstärke vereint. Hierbei ist diese Kennlinie nicht linear, das heißt mit einer Steigung, die sich mit der Veränderung der magnetischen Feldstärke nicht ändert. Diese Hysterese bezeichnet einen Prozess, der immer einem speziellen Verlauf nachgeht. Hier lassen sich keine Sprünge zeigen, sondern es werden immer feste Punkte auf der Kennlinie nachvollzogen. Wie in Abbildung 3.3 leicht zu erkennen ist, ist diese Kennlinie nicht linear, bzw. nur bei einer kleinen Auslenkung fast linear. Selbst bei einer geringen Auslenkung ist auch keine lineare Übertragung möglich [8].

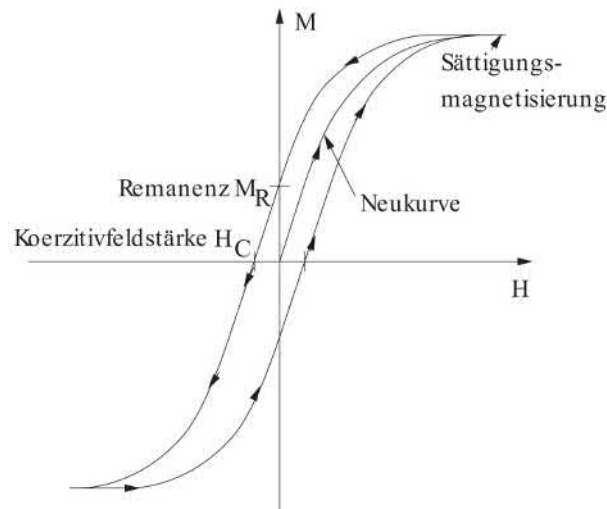


Abbildung 3.3: Schematische Hysteresekennlinie eines Transformators mit einem Eisenkern [13]

M: Sättigung des Eisenkerns
H: Anliegendes Magnetfeld

Ein lineares Zweitor hätte die Eigenschaft, die Wechselspannung, die dort in den Eingang des Zweitors angelegt wird, rein in ihrer Phasenlage und Amplitude zu verändern und schließlich am Ausgang auszugeben. Auch eine Mischung aus mehreren Wechselspannungen wird lediglich wie eine Überlagerung von mehreren Frequenzen behandelt. Die Veränderung der Frequenz ist hierbei nicht erlaubt, da sonst die linearen Gleichungen, die dort für die Definition benutzt werden, nicht gelöst werden können. Durch ein nicht-lineares Zweitor wird jedoch die Möglichkeit gegeben, jeden Parameter zu ändern. Das beste Beispiel hierbei wäre eine Diode. Die Diode verläuft in ihrem idealisierten Arbeitspunkt wie ein Ventil, das entweder durchlässig ist (positive Spannung) oder sperrt (negative Spannung). Dieses Sperren ist jedoch nicht durch ein lineares Zweitor beschreibbar, da es sich eben nicht-linear in ihrer Kennlinie verhält. In Analogie dazu verhält sich der Transformator so, dass er die Hysteresekennlinie nachläuft. Dieses Nachlaufen aufgrund der Netzspannung in Verbindung des Eisenkerns hat zur Folge, dass harmonische Oberschwingungen innerhalb des Ausgangs induziert werden. In diesem Beispiel, mit dem vorher beschriebenen 12-V-Transformator, werden laut FFT eines Oszilloskop DSO-X 3014A des Herstellers Agilent Technologies die Oberschwingungen bei Frequenzen 150 Hz, 250 Hz und 350 Hz stark induziert, wenn eine 50 Hz Wechselspannung am Eingang angelegt wird. Der Abstand der Grundwelle zur 3. Oberschwingung ist dabei gerade einmal 50,625 dB. Der allgemeine Signal-zu-Rausch-Abstand bei Betrachtung von einer Bandbreite von 10 kHz beträgt hierbei ca. 70 dB. Dieses SNR ist jedoch keine absolute

Angabe, sondern beschränkt sich auf den ausgewerteten Zeitraum von zwei Sekunden. Betrachtet man nun hierbei, dass die zu messende Netzspannung am Ausgang an sich verschiedene Harmonische und die Grundschiwingung enthält, wird klar, dass sich dadurch ein Unterschied zu einer linearen Messung aufmacht. Für jede Oberschwingung, die in der Netzspannung enthalten ist, werden für diese eine Schwingung drei neue Wechselspannungen mit dem Faktor der Frequenz 3,5 und 7 stark erzeugt. Dies bedeutet, dass Intermodulationen aufgrund der Nichtlinearität entstehen. Zu sehen ist dies in der Abbildung 3.4. Hierbei wurde eine reine sinusförmige 50-Hz-Schwingung auf den Eingang des Transformators gelegt.

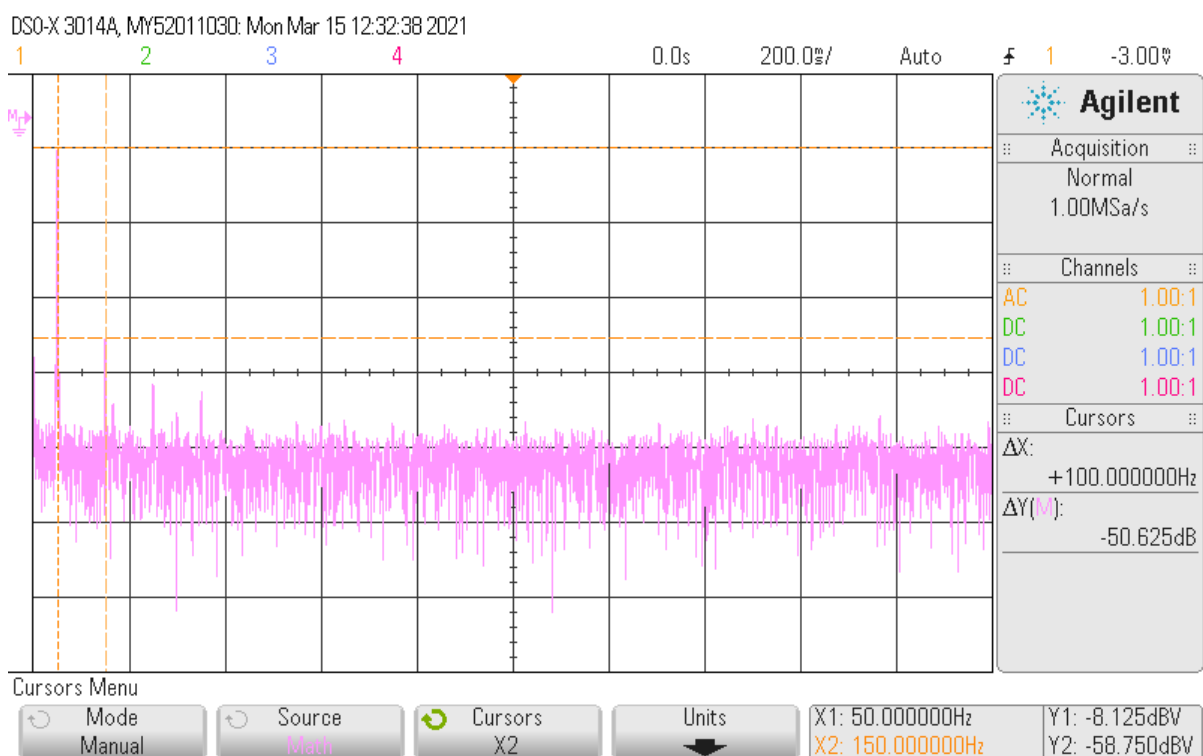


Abbildung 3.4: FFT des Transformators bei einer Eingangsfrequenz von 50 Hz

In einer Messung ist eine Mischung nicht gewollt und sorgt dafür, dass eine deutlich komplexere Störung innerhalb der Messung auftritt, die nicht reversibel ist. Das originale Signal kann dadurch nicht fehlerfrei extrahiert werden. Durch die finale Überlagerung von allen auftretenden Intermodulationsfrequenzen sowie der Eingangsoberschwingungen der echten Netzspannung wird ein Bild deutlich, das von der real auftretenden Netzspannung arg abweicht. In Abbildung 3.5 ist eine deutliche Abweichung ersichtlich, welche stark verdeutlicht, warum dieser explizite Transformator nicht für eine valide Messung der Oberschwingungsgehalte in Frage kommt. Anzumerken ist jedoch, dass die Frequenz aufgrund des Transformators nicht abweicht. Die Grundfrequenz muss trotz des Trans-

formators immer identisch sein, da die Überlagerung von Oberschwingungen nicht die Änderung des Nullpunktdurchgangs zum nächsten Nullpunkt ändern kann.

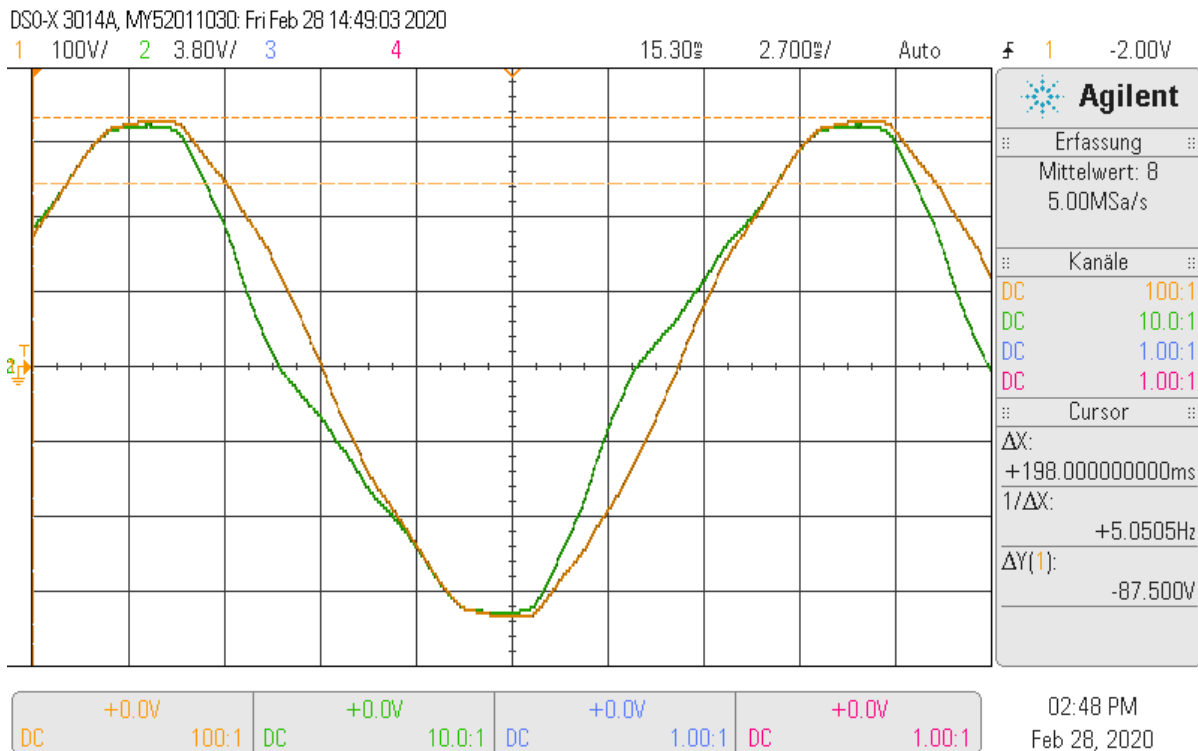


Abbildung 3.5: Gegenüberstellung der Spannungsverläufe zwischen der Messung durch einen Tastkopf und einem Transformator
 Orange Linie: Wellenform des Netzes durch Benutzen eines geeigneten Tastkopfes
 Grüne Linie: Wellenform bei Benutzen eines Transformators

3.1.1 Besondere Anforderungen für berührungssicheres Design

Aufgrund der Nachteile der Verwendung eines Transformators wird ein Entwickeln eines potentialtrennenden Elements nötig. Durch die Potentialtrennung wird sichergestellt, dass das lebensgefährliche Potential der Netzspannungsseite nicht in Richtung des auswertenden Elements weitergeführt werden kann.

Definition der Schutzklasse 3

Elektronische Geräte sind im Europäischen Raum durch die VDE Normen VDE 0140-1 und ihrem europäischen Pendant DIN EN 61140 eingeteilt. Hierbei sind Geräte in vier Klassen von null bis drei einzuteilen. Verschiedene Klassen können für verschiedene

Einsatzzwecke entwickelt werden. Hierbei stellen die Klassen aufsteigend eine höhere Sicherheit gegenüber dem Menschen dar. Die Schutzklasse 0 wird nicht mehr in der EU für zugelassen erklärt, da diese Klasse nicht nach einer Norm, sondern nur durch Nichteinhalten einer Norm erreicht wird. Mit Steigen der Schutzklasse werden größere Hürden für den Entwickler gestellt, um die Anlagen sicherer zu gestalten. Dabei werden z.B. die nötigen Abstände der Leiterplatten-Verbindungen vergrößert und die bemessene Spannung bei der Stückprüfung erhöht. Die Stückprüfung bezeichnet den Prozess vor Marktreife eines Produkts, bei dem ein einzelnes Stück der Normprüfung unterzogen wird. Getestet wird z.B. eine Überspannungsprüfung, bei der die vorgeschriebene Überspannung der Klasse an die Eingänge der Schaltung gelegt werden. Durch eine Sonde wird auf einen Stromfluss an den Ausgängen bzw. dem Gehäuse des Gerätes geprüft. Findet diesbezüglich ein Stromfluss oberhalb der Norm oder ein Durchschlag der Spannung statt, so ist das Gerät nicht der Klasse entsprechend. Nebenher kann z.B. ein Isolationswiderstand R_{iso} gemessen werden, der den Widerstand zwischen dem Erdpotential und den Ausgängen misst. Falls ein Schutzleiter angebracht wäre, würde hier der Schutzleiterwiderstand mit gemessen, um davon ausgehen zu können, dass an dem Stück die Norm eingehalten wird.

Isolationsboard zur galvanischen Trennung der Systeme

Zum Erreichen der Schutzklasse 3 müssen mehrere Kriterien erfüllt werden. Als erste Schutzbarriere wird die Netzseite mit einem eigenen Board von der Auswertung von dem Raspberry Pi, dem Gerät, das schlussendlich die analoge Messung in digitale Werte umsetzt, getrennt. Solange das sogenannte Isolationsboard die Schutzklasse 3 erreicht, ist die nachfolgende Strecke sicher, da das Pi aus einem eigenen Netzteil der entsprechenden Schutzklasse gespeist wird. Bei dem Entwickeln der Platine ist besondere Vorsicht auf die Bauteilauswahl und den Orten der Bauteile zu nehmen. Bei dem benutzten Trennverstärker AMC1301 vom US-amerikanischen Bauelementehersteller Texas Instruments ist dieser nach DIN VDE V 0884-11 und UL1577 unter dem Zertifikat 40040142 bis zu einer Spannung von bis zu $7000 V_{\text{peak}}$ ausgelegt. V_{peak} bedeutet jedoch lediglich eine sehr kurzzeitige Spannung. Eine dauerhafte Spannung von 1500 V-DC kann jedoch laut VDE Zertifizierung immer isoliert werden. Am Eingang dieses Trennverstärkers wird dieser von einem Widerstands-Spannungsteiler, der die Netzspannung von den nominalen 325 V auf das 0.8-fache des linearen Bereichs des AMC1301, somit 200 mV linear heruntersetzt. Durch Benutzen eines 200Ω Widerstands am Eingang des Trennverstärkers wird sichergestellt, dass keine Skalierung die Messung beeinflusst, da der interne Wider-

stand des Trennverstärkers einen Wert von ca. $22\text{ k}\Omega$ besitzt. Als Sicherheitsmaßnahme wird dieser Spannungsteiler mit 1 mA Stromstärke konstant gespeist, welcher jedoch zusätzlich mit einer 4 mA Polyfuse abgesichert ist. Es werden insgesamt vier Widerstände in dem Spannungsteiler verbaut, wodurch dann die maximale Spannung über einem Widerstand von normativ 400 V nicht überschritten werden kann. Somit ist selbst bei einer ungewollten Berührung durch eine Person der maximale Strom begrenzt. Der AMC1301 bietet eine Delta-Sigma-Modulation zwischen dem Eingang und dem Ausgang und trennt die Netzseite von der isolierten Seite durch eine kapazitive Kopplung. Der Verstärker muss durch zwei Spannungen versorgt werden, einmal mit 5 V bis $3,3\text{ V}$ an der unsicheren Netzseite und einmal mit 5 V bis $3,3\text{ V}$ an der schutzisolierten Seite. Die Netzseite wird mit einem Mean-Well-Netzteil gespeist, welches die Schutzklasse 2 aufweist. Da die sekundäre Seite des Verstärkers auch eine Spannungsquelle braucht, ist es nötig, die 5 V Spannung der Netzseite auf die sichere Seite zu wandeln. Hier wird ein RECOM DC/DC-Wandler eingesetzt, der nach der IEC-EN61010-1 zertifiziert ist. Dieser ist mit einer Sekunde von $8000V_{\text{peak}}$ Überspannung der Netzseite getestet und stellt sicher, dass die sichere Seite immer galvanisch von der Netzseite getrennt ist. Im Zeitraum von einer Minute sind AC-Spannungen von bis zu $4000V_{\text{peak}}$ sichergestellt, dass das Bauelement diese Spannung sicher trennt. Beide Bauelemente bieten eine große Strecke zwischen dem Input und dem Output, wobei kein leitfähiges Material unterhalb der Bauelemente gebracht werden darf. Somit ist unter den Bauelementen keine Metallisierung auf das Printed Circuit Board (PCB) aufgebracht und die Kriechstrecken zwischen dem Ein- und Ausgang sollen so groß wie möglich gehalten werden. Kriechstrecken bezeichnen die kürzesten Abstände zwischen zwei leitfähigen Elementen. Hier beschreibt das den kürzesten Abstand der potentialfreien und der Netzseite, zwischen derer kein leitfähiges Material sitzt. Bei Benutzen eines Isolationslacks sollten laut Tabelle 7 der DIN VDE Norm 0411-1 mindestens 3 mm Abstand ohne leitendes Material sein, wenn eine Spannung von 230 V existiert. In diesem Beispiel liegen die kleinsten Kriechstrecken, also die geringsten Abstände bei $5,35\text{ mm}$. Somit sind die normativen Werte deutlich eingehalten. Unabhängig von der Hochspannungssicherheit ist dennoch viel Wert auf die Genauigkeit der Messdaten zu legen. Laut Empfehlung des Herstellers des Trennverstärkers sollten Glättungskondensatoren verbaut werden. Hierbei sind, um die verschiedenen Frequenzbereiche des Verstärkers abzudecken mehrere Kondensatoren mit verschiedenen Kapazitätswerten verbaut. Um eine gute Entstörung und Glättung der Versorgungsspannung zu erreichen, sollte der Kondensator mit der geringsten Kapazität, hier 100 nF , am nächsten am Bauelement verbaut werden, da dieser die höherfrequenten Störungen durch die geringe Impedanz bei dieser Frequenz unterdrückt. Um breitbandig die Störungen zu unterdrücken, werden insgesamt vier lokale Glättungskondensatoren

von 100 nF bis 10 μF für den Verstärker verbaut. Der Grund für das Parallelschalten von Kondensatoren verschiedener Kapazität liegt darin, dass diese Kondensatoren eine unterschiedliche Impedanzkennlinie über der Frequenz aufweisen. Hierbei hat bei gleichen äußeren Ausmaßen des Bauteils der Kondensator mit der geringeren Kapazität einen späteren Tiefpunkt der Impedanz. Somit entkoppeln Kondensatoren mit hoher Kapazität kleine Frequenzen und Kondensatoren mit kleinen Kapazitäten hohe Frequenzen. Impedanzverläufe von Kondensatoren verschieden großer Kapazität sind in der Abbildung 3.6 zu sehen. Durch eine Parallelschaltung wird sichergestellt, dass Störfrequenzen über ein breites Band sicher unterdrückt werden.

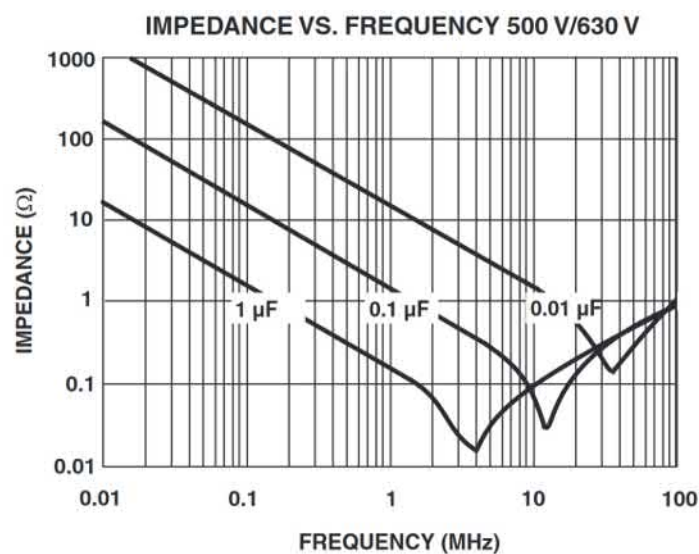


Abbildung 3.6: Schematische Kennlinien des SMD-Kondensatoren Herstellers Vishay für die SMD Größe 1603 [14]

Der DC/DC-Wandler von RECOM sowie das Netzteil benötigen beide zwischen ihren Gleichspannungsausgängen Kondensatoren, um die Funktionalität der Spannungsversorgung zu gewährleisten. Als letzter Schritt muss noch eine Verbindung zwischen der analogen Auswertung und dem Trennverstärker hergestellt werden. Anzumerken ist, dass der Trennverstärker einen differentiellen Ausgang besitzt. Differenziell bedeutet, im Gegensatz zu einer unsymmetrischer Signalübertragung, dass kein gemeinsamer Massepunkt zwischen den beiden Signalen existiert. Differenzielle Übertragungen haben bei einer Gleichtaktstörung von außen eine sehr gute Charakteristik, die bei unsymmetrischer Übertragung, also ein analoger Spannungswert bezüglich eines Massepunkts, nicht vorhanden ist. Differenzielle Signale müssen im Allgemeinen bevor sie gemessen werden können von sich selbst subtrahiert werden. Bei einer Übertragung mit einem Kabel können sie durch Fern- und Nahfeldstörung durch Elektromagnetische Felder direkt gestört

werden. Dadurch wird beispielhaft eine Spannung von 50 mV auf das Signal addiert, welche jedoch im normalen Falle gaußverteilt ist. Dies bedeutet, dass das ursprüngliche Signal nicht mehr ohne Umstände rekonstruiert werden kann, der Signal-zu-Rauschabstand (SNR) wird verschlechtert. Wird aber differenzielle Übertragung benutzt, wird durch die Störung auf beide Leitungen das gleiche Rauschen gleichzeitig addiert. Wird nun am Ende der Übertragung ein Differenzverstärker, also ein Verstärker, der die Differenz aus beiden Signalen verstärkt, eingesetzt, dann werden beide Störungen voneinander subtrahiert und das ursprüngliche Signal wird mit wenig Verlust rekonstruiert. Bei der Übertragung wird ein wenig verbreiteter RJ12 Stecker mit einer 6 poligen Verbindung eingesetzt. Durch die Benutzung dieses Steckers wird zum einen eine Verpolung ausgeschlossen und zum anderen wird mit einfachen Mittel sichergestellt, dass kein falsches Kabel, wie ein Ethernet-Kabel oder USB-Kabel angeschlossen wird. Es werden zwar nur zwei der sechs Adern des Kabels benutzt, aber dafür wird mit einfachen Mittel eine Sicherung von anderen Systemen eingebaut, da diese nicht in Konfrontation mit den Abgängen des Trennverstärkers in Berührung kommt. In Abbildung 3.7 ist eine bestückte Platine dargestellt. Somit ist die Schaltung für die Isolierung aus schaltungstechnischer Sicht abgeschlossen und weitere Sicherheitsvorkehrungen werden erläutert. Ein einfaches, auch hier benutztes Mittel zur Erhöhung der Isolation ist es, die gesamte Platine mit einem spezifizierten Isolationslack zu benetzen. Nach dem fertigen Verbauen und Anlöten der Bauteile wird auf das fertige PCB ein Isolationslack der Firma Kontakt Chemie namens PLASTIK 70 verwendet. Hierbei werden mehrere Lackschichten nach Trocknen der vorherigen Schichten aufgebracht. Dieser Lack ist bis zu einer Durchschlagfestigkeit von über $110 \pm 10 \text{ kV}_{\text{eff}}$ pro Millimeter geprüft. Durch eine solche Maßnahme wird eine allgemeine Schutzmaßnahme gegenüber Berührungen erfüllt, da dieser Lack Berührungen gegenüber der Netzspannung unterbinden kann. Zusätzlich dazu wird die Kriechstrecke deutlich verlängert, da die Oberfläche, an der Überspannungsentladungen stattfinden können, weit über der Platinenoberfläche liegt. Die Leiterplattenentflechtung mit der Bemaßung der essentiellen Kriechstrecke ist unter Abbildung 3.8 zu sehen. Da jedoch eine weitere Schutzmaßnahme empfohlen bzw. benötigt wird, ist ein Gehäuse um die Platine von Nöten. Zurückgegriffen wurde auf ein Plastik auf Polylactidbasis. Dieses wurde durch eine Additive Fertigung, umgangssprachlich aus einem 3D Drucker, gefertigt. Dieses Gehäuse wurde mit dem Logo der Friedrich-Alexander-Universität auf der oberen Gehäusenhälfte versehen und fasst die Platine plan ein. Somit sitzt die Buchse, in die der Gerätestecker für den Netzanschluss angeschlossen wird, in dem Gehäuse und es können die Anschlüsse für den Außenleiter nicht mehr berührt werden. Der Gerätestecker ist von dem Typ IEC C8 und darf eine maximale Stromstärke von 2,5 A führen. Dieser bietet insgesamt zwei Anschlüsse, den Außenleiter und den Neutraleiter. Die beiden

Gehäusehälften werden nach dem Einsetzen der Platine ineinander gesteckt und durch Clips dazwischen fixiert. Eine Klebnaht sorgt dafür, dass sich das Gehäuse nur durch starke Gewalteinwirkung trennen lässt. Somit sind viele Vorkehrungen zur Erhöhung der Sicherheit vorgenommen.

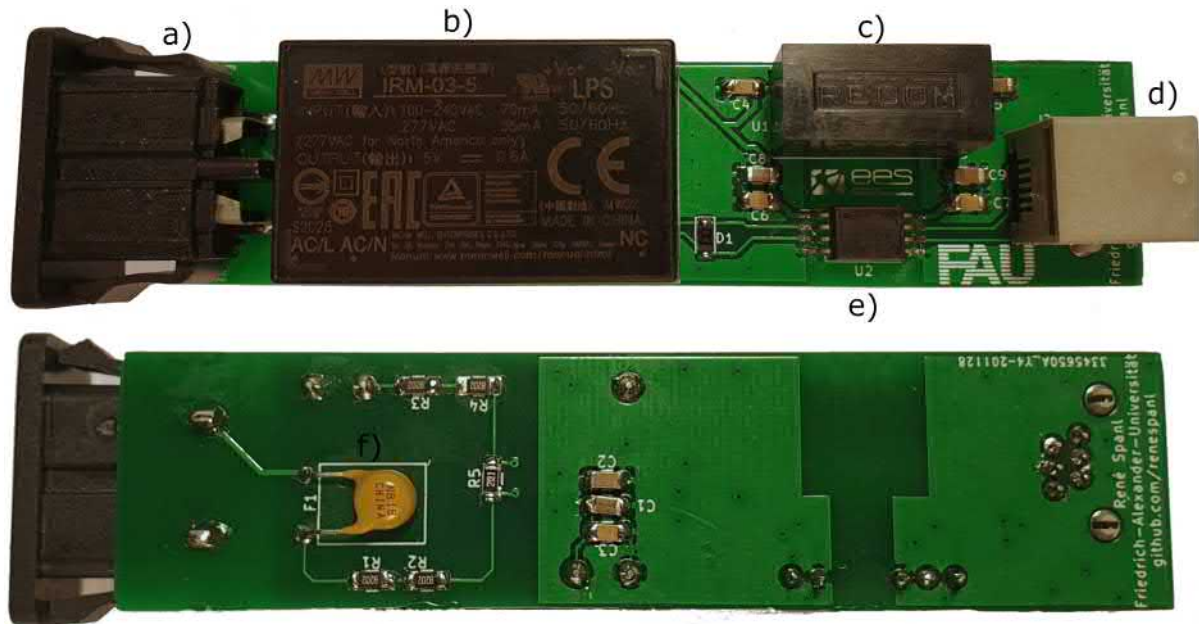


Abbildung 3.7: Abbildung der Ober- und Unterseite der Isolationsplatine ohne Isolationslack sowie Gehäuse
a) IEC C8 Buchse, b) Mean-Well Netzteil c) RECOM DC/DC-Wandler
d) RJ12 Buchse e) AMC1301 Trennverstärker f) Polyfuse

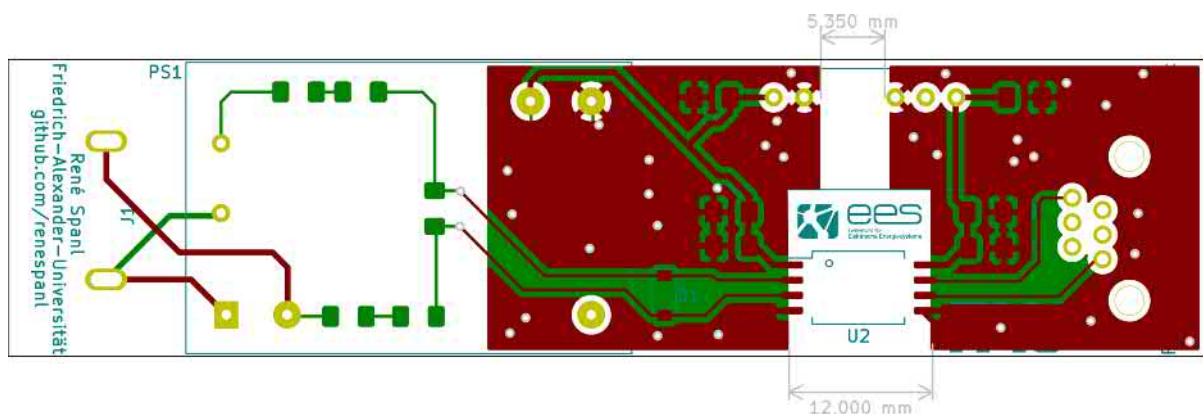


Abbildung 3.8: Entflechtung der Isolationsplatine mit Bemaßung

3.2 Notwendige Bit-Tiefe der Datenauswertung

Die notwendige Bit-Tiefe des auswertenden Glieds in der Messkette ist durch viele Eigenschaften eingeschränkt. Zum einen ist eine hohe Bit-Tiefe mit höheren Kosten verbunden, somit sollte darauf geachtet werden, dass die Bit-Tiefe so gering wie möglich gehalten wird. Zum anderen sind Faktoren wie der Spannungsbereich, der den A/D-Umsetzer auslenkt wichtig. Hierbei sollte der Spannungsbereich des Umsetzers so gut wie möglich ausgenutzt werden, damit das Signal-zu-Rauschverhältnis am höchsten ist. Das Eingangssignal sollte so skaliert werden, dass zwar ein kleiner Sicherheitsrahmen für Überspannungsspitzen vorhanden ist, aber so wenig, dass das Signal so gut wie möglich digital rekonstruiert werden kann. Hier wird davon ausgegangen, dass das gesamte Signal eine maximale Amplitude von $\pm 9\text{ V}$ einnehmen kann. Dadurch wird ein Eingangsbereich des A/D-Umsetzers von $\pm 10\text{ V}$ benötigt. Dies ist jedoch ein weit verbreiteter Standard, da Referenzspannungen meist in diesen Bereichen arbeiten können. Um zu bestimmen, welche Bit-Tiefe überhaupt erforderlich ist, sollte man berechnen, welchen Abstand das Signal zu dem anderen Teil des Signal hat. In diesem Falle ist gemeint, wie groß das Verhältnis aus der Amplitude des größten Signals, also hier der Grundschiwingung und der kleinsten zu messenden Oberschiwingung ist. Laut der Tabelle 2.1 sind teilweise Schwiungen unterhalb von 0,5% der Amplitude der Grundschiwingung nötig zu überwachen. Somit wird dafür angesetzt, dass eine Genauigkeit von 0,1% erreicht werden soll. Um diese 0,1% Genauigkeit auch wirklich erreichen zu können sollte hierbei genug Dynamik des A/D-Umsetzers offen sein, dass vier verschiedene Spannungswerte für eine Oberwelle im Bereich von 0,1% der Grundwelle angesetzt sind. Die notwendige Bit-Tiefe (NOB für Number-Of-Bits) ist hierbei wie folgt auszurechnen:

$$\text{NOB} = \lceil \log_2 \left(\frac{9\text{V} \cdot 2}{0,1\% \cdot 4} \right) \rceil = \lceil 12,13\text{bit} \rceil \implies \text{NOB} = 13\text{bit} \quad (3.3)$$

Somit wären für die aktuelle Berechnung eine Bit-Tiefe von 13 Bit in dem A/D-Umsetzer nötig. Jedoch existiert dazu eine andere Problematik, die aus dem Trennverstärker AMC1301 resultiert. Dieser Umsetzer gibt ein Signal mit einem hohen SNR aus, welches laut Datenblatt einen Wert von 84 dB besitzt. Entsprechend der Formel, die sich aus der Quantisierung eines Sinus-förmigen Signals ergibt, lautet die wirklich effektive Nummer an benutzbaren Bits(ENOB) bei:

$$\text{ENOB} = \left\lceil \frac{\text{SINAD}_{\text{MEASURED}} - 1,76\text{dB} + 20\log\left(\frac{A_{\text{Umsetzer}}}{A_{\text{Signal}}}\right)}{6,02\text{dB}} \right\rceil \quad (3.4)$$

$$= \left\lceil \frac{84\text{dB} - 1,76\text{dB} + 20\log_{10}\left(\frac{10V}{9V}\right)}{6,02\text{dB}} \right\rceil = \lceil 13,813\text{-bit} \rceil = 14\text{-bit} \quad (3.5)$$

Durch diese Beschränkung ist eine Dynamik von mindestens 14-bit am Eingang des A/D-Umsetzers nötig, um den AMC1301 voll auszusteuern und die gesamte Informationstiefe zu erhalten [6]. Durch Benutzen von 14-bit ist eine Amplitudentreue für vier Stellen der Amplitude bei Umstellen der Gleichung wie folgt zu berechnen:

$$\text{Accuracy} = \frac{9V \cdot 2}{4 \cdot 2^{13,83}} = 309 \times 10^{-6} \approx 0,031\% \quad (3.6)$$

Durch benutzen eines 14-bit A/D-Umsetzers ist also eine sehr gute Genauigkeit zu erreichen, die die gewollte Dynamik, die ein 13-bit A/D-Umsetzer besitzt, deutlich übersteigt. In diesem Falle ist es jedoch so, dass eine höhere Bit-Tiefe keinen Nachteil erzeugt, da der Rechenaufwand für einen 14-bit Umsetzer nicht geringer ausfällt als für einen 16-bit Umsetzer. Dies entsteht dadurch, da die meisten Computer eine minimale Wortbreite von 8-Bit, also einem Byte in ihrem Prozessor enthalten. Somit ist es für die Verarbeitungszeit für einen Wert irrelevant, ob dieser nun 9- oder 16-bit tief ist, da zwei Byte für Speicher dafür alloziert werden und diese zwei Byte für die Berechnung verwendet werden.

3.3 Notwendige Abtastrate

Die erste Bedingung, um die nötige Abtastrate zu beschreiben, ist es, dass das Shannon-Nyquist-Abtasttheorem eingehalten werden muss. Dieses Theorem besagt, dass die Abtastfrequenz mindestens doppelt so groß sein muss, wie die Frequenz des zu messenden Signals. Da in diesem Fall jedoch nicht nur die 50-Hz-Schwingung gemessen wird, sondern dazu die Oberschwingungen betrachtet werden sollen, sind also mindestens 2500 Hz Abtastrate nötig, um eine Aussage über das Signal zu treffen. Für den zu implementierenden Algorithmus, der im Kapitel 4.1.2 näher erläutert wird, ist es jedoch von Wichtigkeit, viele Stützstellen der Spannungswerte zu haben, da eine Interpolation fehlerbehaftet ist.

Somit sollten die Zwischenstellen des Nulldurchgangs, also die Stelle bei der ein Punkt an der Stelle i positiv ist oder 0 entspricht, und an der darauffolgenden Stelle ein negatives Vorzeichen besitzt, so nah wie möglich beieinander liegen. Hierbei gibt es bei verschiedenen A/D-Umsetzer-Arten Vor- und Nachteile. Bei der Schaltung muss aber davon ausgegangen werden, dass der A/D-Umsetzer im gesamten Frequenzbereich eine Linearität besitzt. Dies ist bei einigen für den Audibereich entwickelten A/D-Umsetzern nicht gegeben, da diese teils die Netzfrequenz aufgrund ihrer störenden Weise aus dem Frequenzbereich herausfiltern. Somit müssen All-Purpose-ADC benutzt werden, die garantieren, im gesamten Frequenzbereich zu funktionieren. Es existieren drei größere verfügbare Typen eines Analog/Digital-Umsetzers: einmal einen Delta-Sigma-Konverter, einen Successive Approximation-Umsetzer (SAR) und einen Pipeline-Umsetzer. Diese bieten jeweils verschiedene Vorteile. Ein Delta-Sigma-Umsetzer hat die Problematik, zwar eine gute Genauigkeit zu besitzen, aber jedoch eine geringe Abtastfrequenz aufzuweisen. Die Pipelined-ADC sind hierbei das genaue Gegenteil, sie besitzen eine sehr hohe Abtastfrequenz, jedoch nur eine bit-Tiefe von unter 14-bit. Diese verbrauchen dazu noch recht viel Chipfläche im Vergleich zu den anderen Typen von Umsetzern. Diese Pipelining-ADC sind oft in Verwendung bei einem digitalen Oszilloskop, da diese eine sehr hohe Abtastrate besitzen müssen, um das Signal vollständig abzutasten. Hierbei sind 2 GigaSamples pro Sekunde und höher möglich. Ein Successive-Approximation Umsetzer, auch SAR-ADC, stellt hier die genaue Mitte zwischen beiden Extremen dar, da dieser eine Abtastrate von 10 kS/s bis hin zu 20 MS/s besitzt. Diese Typen von AD-Umsetzer sind zudem in der Anschaffung relativ preiswert und bieten den Vorteil, dass sie eine Bit-Tiefe von bis zu 18-bit bieten können. Ein SAR-ADC beinhaltet vereinfacht Logik zur Kontrolle des Chips, einen Komparator und einen Digital/Analog-Umsetzer, also das genaue Gegenstück zu einem A/D-Umsetzer. Ein schematisches Blockdiagramm ist unter Abbildung 3.9 dargestellt. Wird ein Spannungswert abgefragt, so wird das aktuelle Signal in einem Sample-and-Hold-Block starr fixiert, sodass dieses über die Dauer der Konvertierung konstant bleibt. Dieser statische Wert wird in den Eingang eines Komparators gelegt, der mit dem Ausgang des internen DACs zum starren Signal vergleicht und entweder eine logische Eins oder logische Null ausgibt. Zu Beginn ist dieser DAC auf Null gesetzt, sodass der Startwert der DAC-Spannung 0 V beträgt. Durch einen Suchalgorithmus in der Logik des Chips wird die Spannung des D/A-Umsetzers Bit für Bit näher an den Spannungswert des eingehenden Signals angeglichen. Dieser Wert, den der DAC als Parameter zuletzt gesetzt hat, ist schlussendlich der Ausgabewert des A/D-Umsetzers, da dieser dem Wert des eingehenden Spannungssignal in der spezifizierten Genauigkeit, abgesehen von geringen Fehlern, die im Datenblatt beschrieben sind, identisch ist.

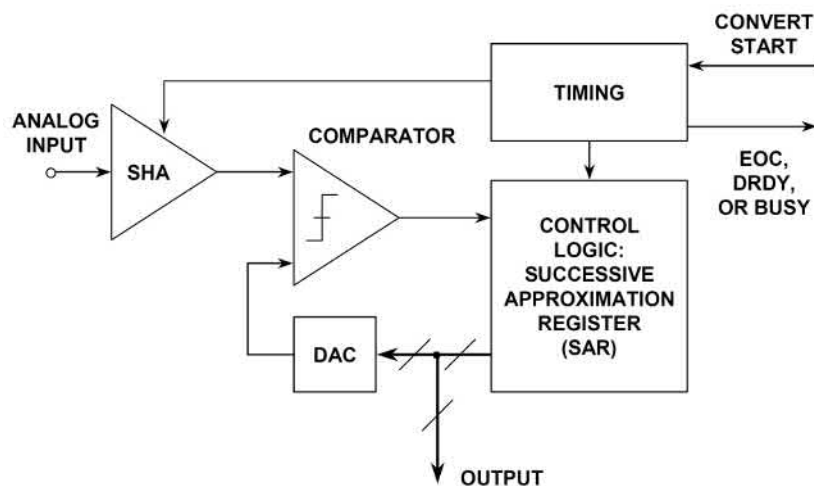


Abbildung 3.9: Schematischer Aufbau eines Successive-Approximation Analog/Digital-Umsetzers [6]

3.4 Analoge Filter zur Verbesserung der Genauigkeit der Auswertung

Rauschleistung als Problem der Messdatenerfassung

Die Betrachtung eines idealen System weicht von der Realität ab, weil Vereinfachungen vorgenommen werden. Im realen System findet unter anderem ein Addieren eines Rauschens auf das vorher rauschfreie System statt. Durch diese Addition ist es nötig, das betrachtete Signal in einem Band zu begrenzen, welches zwischen 0 Hz und 1250 Hz liegt. Hier wird ein Tiefpass nötig, da aufgrund der Simplizität von jeder nötigen Stützstelle ($n * 50 \text{ Hz}$ für $n = 0 \dots 25$) ein Notchfilter zu aufwendig wäre. Bei den Betrachtungen reicht ein Filter, dessen Band von 0 Hz bis 1250 Hz die Amplitude und Phase der Frequenz nicht sonderlich verändert. Bei der Berechnung der hierfür benötigten Filter wird von einer maximalen Amplitudenänderung von 1% ausgegangen. Damit ist sicher zu stellen, dass das originale Signal nicht verfälscht wird. Es kann durch den Tiefpass eine Phasenverschiebung entstehen, die aber alleinig die Form verzerrt. Da das zu untersuchende Frequenzspektrum im Voraus bekannt ist, bietet sich die Anwendung eines Tiefpassfilters an. Bei einer digitalen Messdatenerfassung muss immer eine Tiefpassfilterung vorgenommen werden, um dem Aliasing, also das Auftreten von zu hohen Frequenzen, die oberhalb der Shannon-Nyquist-Frequenz liegen, zu unterdrücken. Diese hohen Frequenzen können sonst eine Schwebung innerhalb des Signals erzeugen. Dieser Anti-Alias-Filter lässt dennoch eine sehr große Bandbreite in einen Analog-Digital-Umsetzer. Dieses Band lässt

sich hier stark begrenzen, sodass die Energie im Band, also die Amplitude des Signals bei nicht gebrauchten Frequenzen so gering wie möglich ist. Die Korrelation des Signals im Zeit- bzw. Frequenzbereich wird durch das Theorem von Parseval in Zusammenhang gebracht. Dieses besagt, dass die Energie E eines Zeitsignals auch im Betragsspektrum ausgerechnet werden kann. Durch die Gleichung zur Fourier Transformation wird das Zeitsignal $s(t)$ in den Kontext des Spektrums $S(f)$ des Signals $s(t)$ gebracht [10].

$$E_s = \int_{-\infty}^{+\infty} |s(t)|^2 dt = \int_{-\infty}^{+\infty} |S(f)|^2 df \quad (3.7)$$

Unter Annahme, dass die Störung des Messsignals durch weißes, gaußsches Rauschen entsteht, lässt sich durch Begrenzung des Bandes das Rauschen deutlich reduzieren. Dieses additive weiße gaußsches Rauschen ist unkorreliert und ein reines probabilistisches Signal, welches ein Zufallssignal darstellt und kann dementsprechend nicht reversibel gemacht werden. Die einzige Abhilfemaßnahme stellt dementsprechend eine Verkleinerung des Frequenzbandes dar.

Sallen-Key-Schaltung zur Optimierung der Simplität der Schaltungstechnik

Zur Realisierung verschiedener analoger Filter wird oft die Sallen-Key-Schaltung der US-Amerikanischen Wissenschaftler R. P. Sallen und E. L. Key des Massachusetts Institute of Technology (MIT) aus dem Jahre 1955 gebraucht. Diese Filterstruktur zeichnet sich dadurch aus, dass für jede Stufe (engl. stage) ein Operationsverstärker und vier passive Bauelemente gebraucht werden. Durch diese Anordnung lassen sich pro Stufe eine Filterstrecke zweiter Ordnung erstellen. Aufgrund der Symmetrie lassen sich durch eine Hoch/Tiefpass-Transformation und Vertauschen der Bauelemente die Funktion des Tiefpasses mit dem eines Hochpasses tauschen. Die typische Beschaltung des Operationsverstärkers für einen Tiefpass ist unter Abbildung 3.10 zu sehen. Da in Realität eine Polstelle in der Darstellung der s-Transformation in einem Abfallen der Verstärkung von -10 dB pro Dekade bedeutet, hat es einen enormen Vorteil, mehrere Tiefpässe hintereinander zu kaskadieren. Diese individuellen Tiefpässe müssen jedoch eine unterschiedliche Grenzfrequenz besitzen, da sie sich durch die Kaskade zu einer gemeinsamen Grenzfrequenzen f_g addieren. In diesem Szenario werden drei der vier vorhandenen Operationsverstärker zu einem Tiefpass zusammengeschaltet. Der übrige Operationsverstärker stellt einen Differenzverstärker dar, da das Signal des Trennverstärkers AMC1301 von Texas Instruments einen differenziellen Ausgang bietet. Dieser differenzielle Ausgang muss, um von Analogtechnik leicht zu manipuliert zu werden, auf das gemeinsame Massepotential der Analogplatine umgesetzt werden. Somit wird eine symmetrische Übertragung zu

einer asymmetrischen mit gemeinsamen Potenzial umgesetzt. Die Struktur einer Sallen-Key-Filterstufe hat die folgende Form:

$$\frac{U_{\text{aus}}(s_n)}{U_{\text{in}}(s_n)} = A(s_n) = \frac{A_0}{1 + s_n w_g C_1 (R_1 + R_2) + s_n^2 w_g^2 R_1 R_2 C_1 C_2} \quad (3.8)$$

$$A(s_n) = \frac{A_0}{1 + a_1 s_n + b_1 s_n^2} \quad (3.9)$$

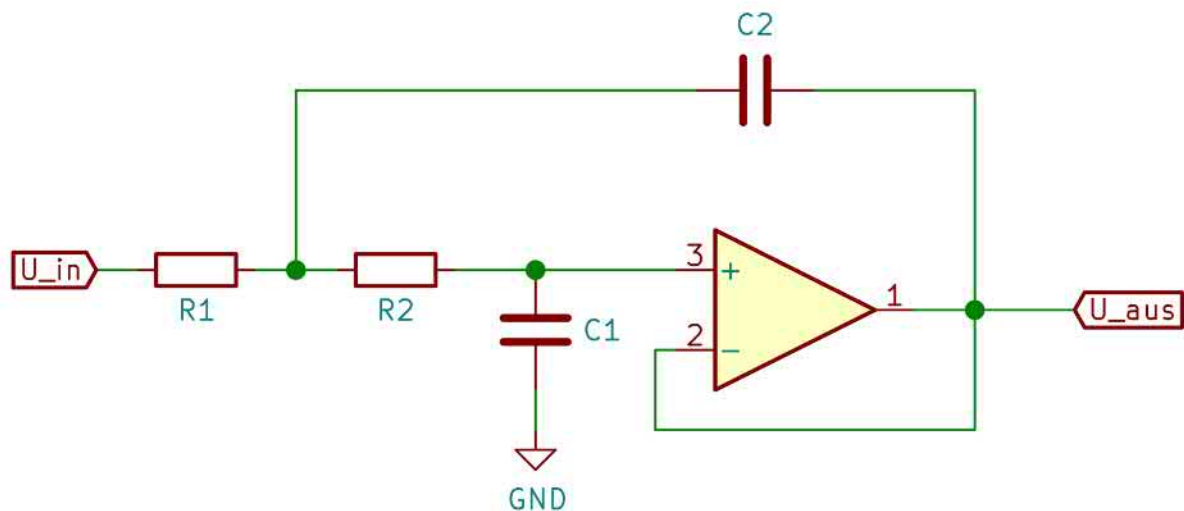


Abbildung 3.10: Schematischer Aufbau einer Sallen-Key Stage

Hierbei entspricht der lineare Verstärkungsfaktor $A_0=1$, da eine Verstärkung innerhalb der Tiefpassfilter abzulehnen ist, weil eine Erhöhung der Verstärkung eine lineare Zunahme des *Gain-bandwidth product* mit sich trägt, wovon der jeweilige Operationsverstärker nur eine endliche Zahl aufweist, was auf den Grundaufbau eines Differenzverstärkers innerhalb des OPVs zurückzuführen ist. Das GBWP beschreibt die maximale Frequenz bei dem der Verstärker die Verstärkung von eins besitzt. Die Form eines Biquads, also einer Filterfunktion zweiten Grades ohne jeweilige Bauteilbestimmung ist in Gleichung 3.9 gegeben. Hierbei sind alleinig die Filterkoeffizienten in a_i und b_i gegeben und können je nach Typ des gewählten Filters bestimmt werden. Durch einen Koeffizientenvergleich mit ausgewählten Werten a_i und b_i können die Bauelementewerte ausgerechnet werden.

Filterkoeffizienten für verschiedene Tiefpassverhalten

Ein sehr großer Vorteil ist die vorhergehende Berechenbarkeit von Filterkoeffizienten. Hierbei gibt es verschiedene Familien der Koeffizienten, unter anderem Bessel,

Butterworth, Cauer- und Tschebyscheff-Filterstrukturen. Der Besselfilter hat hierbei den großen Vorteil der geringen Unterschiede der Sprungantwort und damit verbunden auch eine exakt konstante Gruppenlaufzeit im Durchlassbereich sowie keine Phasenverschiebung der Frequenzen derer. Dieser Typ von Filterkoeffizient hat den Nachteil, dass der Übergang zwischen Durchlassbereich und Sperrbereich sehr weich ist, das bedeutet, man hat weniger Dämpfung im Sperrbereich. Eine andere Option wäre eine Butterworth-Filterstruktur. Hierbei werden die Koeffizienten so gewählt, dass der Durchlassbereich so lange wie möglich bis zur Grenzfrequenz gerade, also ohne Veränderung der Amplitude, gehalten wird. Bei diesem Filter hat man jedoch ein Überspringen in der Sprungantwort, also entwickelt man dadurch einen nicht exakt linearen Phasenverlauf. Elliptische Filter, auch Cauer-Filter genannt, sind dahin optimiert, einen sehr steilen Übergang zwischen Durchlass- und Sperrbereich zu erzielen. Um dieses Ziel zu erreichen ist es von Nöten, dass im Durchlassbereich und Sperrbereich die Übertragungsfunktion des Filters oszilliert. Dieses Schwingen sorgt dafür, dass im Durchlassbereich die Amplitude nicht korrekt dargestellt wird. Zumal wird die Phasenlage durch diese Eigenschaft sehr stark verzerrt und nicht mehr korrekt am Ende dargestellt. Der Tschebyscheff'sche Filter hat eine ähnliche Eigenschaft, sodass der Durchlassbereich sehr stark schwingt und keine korrekte Amplitudendarstellung hat. Hierbei gibt es zusätzlich die Möglichkeit einen Typ-2-Tschebyscheff-Filter zu benutzen, der die Oszillation dafür im Sperrbereich hat. Um die eigentlichen Spannungswerte korrekt und ohne Verzerrung zu erhalten, stellt sich heraus, dass der Butterworth-Filter eine deutliche Verbesserung der Eigenschaften im Vergleich zu den anderen Filtertypen hat, da dieser Filter eine konstante Amplitude aufweist. Hierbei ist es für eine repräsentative Darstellung der Netzfrequenz am wichtigsten, die richtige netzseitige Spannung anzuzeigen, ohne die Oberschwingungsgehalte durch Amplitudenänderung der Oberschwingung zu verzerren. Bei der Betrachtung sollte bei der Bestimmung von f_g der kaskadierten Filterstruktur eine Änderung von maximal 1% der Amplitude der höchsten Oberschwingung erlauben. Bei dem Entwickeln einer Butterworth-Filterstruktur werden die Koeffizienten a_i und b_i , die in Gleichung 3.9 benutzt werden durch folgende Gleichungen beschrieben [12]:

Ordnung N gerade:

$$a_i = 2 \cos \frac{(2i-1)\pi}{2N} \quad \text{für } i = 1 \dots \frac{N}{2} \quad (3.10)$$

$$b_i = 1 \quad (3.11)$$

Ordnung N ungerade:

$$a_1 = 1 \quad (3.12)$$

$$a_i = 2 \cos \frac{(i-1)\pi}{N} \quad \text{für } i = 2 \dots \frac{N+1}{2} \quad (3.13)$$

$$b_i = 1 \quad (3.14)$$

Die Koeffizienten sind somit durch verschiedene Cosinus-Werte definiert, jedoch sind die Koeffizienten der quadratischen Teile b_i jedoch immer einer eins entsprechend. Ein Benutzen der Tabelle 12.21 aus [12] ergibt mit einem Filtergrad von sechs folgende Werte:

$$\begin{array}{ll} a_1 = 1,9319 & b_1 = 1 \\ a_2 = 1,4142 & b_2 = 1 \\ a_3 = 0,5176 & b_3 = 1 \end{array}$$

Durch Einsetzen der Koeffizienten in die Gleichung 3.9 lautet die Gleichung unter der Bedingung, dass die Gleichspannungsverstärkung A_0 eins entspricht wie folgt:

$$A(s_n) = \frac{1}{(1 + 1,9319s_n + s_n^2)(1 + 1,4142s_n + s_n^2)(1 + 0,5176s_n + s_n^2)} \quad (3.15)$$

Wenn man diese Gleichung ausmultipliziert und nach Ordnungen aufreihet, ergibt sich eine Übertragungsfunktion, die keine Klammer enthält. Hierbei ist ersichtlich, dass der höchste Grad des s_n insgesamt sechs entsprechen muss, da sechs Polstellen für den Tiefpass gewollt sind. Es ergibt folgende Form:

$$A(s_n) = \frac{1}{s_n^6 + 3,8637s_n^5 + 7,464s_n^4 + 9,1422s_n^3 + 7,464s_n^2 + 3,8637s_n + 1} \quad (3.16)$$

Um jedoch hierbei die Amplitude zu berechnen, benötigen wir noch die Definition von s_n , bei dem jedoch nur die imaginäre Einheit genommen wird:

$$s_n = \frac{s}{w_g} = j \frac{w}{w_g} = j \frac{f}{f_g} \quad (3.17)$$

Das Ziel ist es, die Grenzfrequenz f_g so zu bestimmen, dass nur noch maximal 1% der Amplitude der obersten Oberschwingung überhaupt verändert wird. Somit wird der Betrag der Übertragungsfunktion gewertet, da dieser sonst normalerweise einer komplexen Größe entspricht. Die Bedingung zur Lösung lautet nun:

$$|A(1250\text{Hz})| \stackrel{!}{=} 1 - 1\% = 0,99 \quad (3.18)$$

$$= \left| \frac{1}{s_n^6 + 3,8637s_n^5 + 7,464s_n^4 + 9,1422s_n^3 + 7,464s_n^2 + 3,8637s_n + 1} \right| \quad (3.19)$$

Durch eine Lösung des Problems durch einen Solver wie WolframAlpha ergibt sich ein Verhältnis von $\frac{f}{f_g}$ von 0,722997. Durch Umsetzen ergibt sich eine Filtereckfrequenz von 1736,1 Hz, die für Einhaltung der Bedingung der geringen Amplitudenänderung von einem Prozent für die höchste gemessene Frequenz nötig ist. Um jedoch vor allem im analogen Bereich Spielraum für Ungenauigkeiten zu haben, wird diese Eckfrequenz auf insgesamt 3 kHz aufgerundet. Somit wird sichergestellt, dass Toleranzen keine Auswirkungen auf den Durchlassbereich haben können. Dies hat jedoch in einem fest aufgebauten System ohne Bauelementtoleranzen keine Anwendung, sondern ist nur für die leichtere Möglichkeit der Fehlereingrenzung implementiert. Dadurch wird im Allgemeinen eine Filterstruktur für $f_g = 3000$ Hz Eckfrequenz ausgelegt.

Auslegen der Filterstufe 1:

Eine Sallen-Key Struktur ergibt zu viele Freiheiten bei der Auslegung von Filtern. Hierbei muss ein Bauelementwert vorgegeben sein. Als Empfehlung lässt sich der Kondensator C_1 für diese und alle nachfolgenden Stufen auf 10 nF setzen. Den Bauelementwert für C_2 lässt sich schließlich aus der Bedingung für reale Widerstände errechnen. Diese lautet wie folgt :

$$\frac{C_2}{C_1} \geq \frac{4b_1}{a_1^2} \implies C_2 \geq \frac{4 \cdot 1,0}{1,931^2} \cdot C_1 \geq 1,0727 \cdot C_1 \quad (3.20)$$

Die E24-Reihe für Kondensatoren sieht vor, dass dort sowohl 10 nF als auch 11 nF möglich sind. Somit wird C_1 auf 10 nF und C_2 auf 11 nF gesetzt, die Widerstände R_1 und R_2 werden im nächsten Schritt durch die Koeffizienten definiert. Diese Berechnung erfolgt ähnlich einem Gleichungssystem, welches zwei Variablen besitzt. Durch einen Koeffizientenvergleich der Definition der Sallen-Key Struktur unter Gleichung 3.8 zur Auslegung der Filterstruktur eines Biquads in Gleichung 3.9 lassen sich die Werte für

Bauelemente bestimmen.

$$A(s_n) = \frac{1}{1 + s_n w_g C_1 (R_1 + R_2) + s_n^2 w_g^2 R_1 R_2 C_1 C_2} \quad (3.21)$$

$$A(s_n) = \frac{1}{1 + a_1 s_n + b_1 s_n^2} \quad (3.22)$$

Die Definition von der Grenzkreisfrequenz zur Grenzwinkelgeschwindigkeit $w_g = 2\pi f_g$ ist jedoch bei der Berechnung wichtig. Durch die vorhergehende Tabellenwerte stellt sich für a_1 ein Wert von 1,9319 ein, somit lässt sich durch den Koeffizientenvergleich die Gleichung für den Koeffizienten a_1 lösen:

$$\begin{aligned} a_1 &= w_g C_1 (R_1 + R_2) = 1,9319 \\ \implies (R_1 + R_2) &= \frac{1,9319}{C_1 w_g} \\ &= \frac{1,9319}{C_1 2\pi f_g} \\ \text{I)} \quad &= 10,249\text{k}\Omega \end{aligned} \quad (3.23)$$

Bei der Berechnung des Koeffizienten b_1 , welcher den Wert von 1 laut der Tabelle hat, wird dann schließlich für die Lösung der ersten Gleichung 3.23 benötigt.

$$\begin{aligned} b_1 &= w_g^2 C_1 C_2 R_1 R_2 = 1,0 \\ \implies R_1 R_2 &= \frac{1,0}{(2\pi f_g)^2 C_1 C_2} \\ &= \frac{1}{(2\pi f_g)^2 C_1 C_2} \\ \text{II)} \quad &= 25,586\text{M}\Omega^2 \end{aligned} \quad (3.24)$$

Somit zur Lösung des Gleichungssystems:

$$\text{aus I): } R_1 = 10,249\text{k}\Omega - R_2 \quad (3.25)$$

$$\text{in II): } R_2 \cdot (10,249\text{k}\Omega - R_2) = 25,586\text{M}\Omega^2 \quad (3.26)$$

$$\implies -R_2^2 + R_2 \cdot 10,249\text{k}\Omega = 25,586\text{M}\Omega^2 \quad (3.27)$$

$$(3.28)$$

$$\implies \text{durch P-Q-Formel ergibt } R_2 = 4,303\text{k}\Omega \quad (3.29)$$

$$\implies \text{in I): ergibt } R_1 = 5,946\text{k}\Omega \quad (3.30)$$

Die beiden nächsten Widerstände der E-Reihe wären hierbei 5,9kΩ und 4,32kΩ, was eine

sehr gute Näherung mit einem nominalen Fehler unter 1% suggeriert. Da die verwendeten E-Reihen jedoch jeweils eine Toleranz von $\pm 2\%$ besitzen, sind diese Werte für diese Zwecke ohne negative Aspekte benutzbar.

Bei der Berechnung der Komponenten der zweiten Filterstufe wird der Wert des Kondensators C_1 wie bei der letzten Stufe auf 10 nF gesetzt. Die Berechnung des Kondensators C_2 sieht diesmal wie folgt aus:

$$\frac{C_2}{C_1} \geq \frac{4b_2}{a_2^2} \implies C_2 \geq \frac{4 \cdot 1,0}{1,4142^2} \cdot C_1 \geq 2,0 \cdot C_1 \quad (3.31)$$

Dadurch wird der Kondensator C_2 dem Wert 20,5 nF zugewiesen, welcher ab der E48-Reihe verfügbar ist. Die Lösung der Berechnung wird nun analog zu der ersten Stufe des kaskadierten Filters vollzogen.

$$\begin{aligned} a_2 &= w_g C_1 (R_1 + R_2) = 1,4142 \\ \implies (R_1 + R_2) &= \frac{1,9319}{C_1 w_g} \\ &= \frac{1,4142}{C_1 2\pi f_g} \\ \text{I)} \quad &= 7,502 \text{ k}\Omega \end{aligned} \quad (3.32)$$

Die Lösung des zweiten Teils des Gleichungssystems:

$$\begin{aligned} b_2 &= w_g^2 C_1 C_2 R_1 R_2 = 1,0 \\ \implies R_1 R_2 &= \frac{1,0}{(2\pi f_g)^2 C_1 C_2} \\ &= \frac{1}{(2\pi f_g)^2 C_1 C_2} \\ \text{II)} \quad &= 13,73 \text{ M}\Omega^2 \end{aligned} \quad (3.33)$$

$$\text{aus I): } R_1 = 7,502 \text{ k}\Omega - R_2 \quad (3.34)$$

$$\text{in II): } R_2 \cdot (10,249 \text{ k}\Omega - R_2) = 13,73 \text{ M}\Omega^2 \quad (3.35)$$

$$\implies -R_2^2 + R_2 \cdot 10,249 \text{ k}\Omega = 13,73 \text{ M}\Omega^2 \quad (3.36)$$

$$(3.37)$$

$$\implies \text{durch P-Q-Formel ergibt } R_2 = 4,303 \text{ k}\Omega \quad (3.38)$$

$$\implies \text{in I): ergibt } R_1 = 3,19 \text{ k}\Omega \quad (3.39)$$

Bei der dritten Stufe wird der Wert von C_1 wieder auf 10 nF gesetzt. Durch Berechnung

des Verhältnisses durch die Gleichung 3.20 muss der Kondensator C_2 mindestens um den Faktor 14,93 größer sein. Um den nächsthöheren Wert der Reihe zu erlangen, wird der Kondensator C_2 auf den Wert 154 nF gesetzt.

$$\begin{aligned}
 a_3 &= w_g C_1 (R_1 + R_2) = 0,5176 \\
 \implies (R_1 + R_2) &= \frac{0,5176}{C_1 w_g} \\
 &= \frac{0,5176}{C_1 2\pi f_g} \\
 \text{I)} \quad &= 2,745\text{k}\Omega
 \end{aligned} \tag{3.40}$$

Die Lösung des zweiten Teils des Gleichungssystem:

$$\begin{aligned}
 b_3 &= w_g^2 C_1 C_2 R_1 R_2 = 1,0 \\
 \implies R_1 R_2 &= \frac{1,0}{(2\pi f_g)^2 C_1 C_2} \\
 &= \frac{1}{(2\pi f_g)^2 C_1 C_2} \\
 \text{II)} \quad &= 1,8275\text{M}\Omega^2
 \end{aligned} \tag{3.41}$$

$$\text{aus I): } R_1 = 2,745\text{k}\Omega - R_2 \tag{3.42}$$

$$\text{in II): } R_2 \cdot (2,745\text{k}\Omega - R_2) = 1,8275\text{M}\Omega^2 \tag{3.43}$$

$$\implies -R_2^2 + R_2 \cdot 10,249\text{k}\Omega = 1,8275\text{M}\Omega^2 \tag{3.44}$$

$$\tag{3.45}$$

$$\implies \text{durch P-Q-Formel ergibt } R_2 = 1,13\text{k}\Omega \tag{3.46}$$

$$\implies \text{in I): ergibt } R_1 = 1,62\text{k}\Omega \tag{3.47}$$

Somit sind alle drei Stufen der Sallen-Key-Filterstruktur definiert und drei der vier Operationsverstärker sind in Benutzung.

Das letzte Element der Filterstrecke wird für den Eingangsverstärker benutzt. Es wird eine klassische Differenzverstärkerschaltung umgesetzt. Hierbei gibt es bei der Entwicklung einige Dinge zu beachten, die aufgrund der analogen Charakteristik zurückzuführen ist. Zuerst ist es so, dass ein Operationsverstärker nur einen begrenzten linearen Bereich seiner Ausgangsspannung besitzt. Laut Datenblatt wird bei einer einzelnen Versorgungsspannung von insgesamt 30 V eine maximale Ausgangsspannung von 28V erreicht. Im Umkehrschluss sollte, um einen linearen Bereich zu garantieren, bei einer symmetrischen

Spannungsversorgung mit $\pm 10\text{ V}$ eine maximale Ausgangsspannung von $\pm 9\text{ V}$ erreicht werden. Durch die Definition des vorhergehenden AMC1301 Differenzverstärker, der nach Datenblatt eine maximale Ausgangsspannung von $0,25\text{ V} \cdot 8,1 = 2,025\text{ V}$ ausgibt, bleibt nach einer Rundung des Wertes für die nötige Verstärkung von $A = \frac{9\text{ V}}{2,025\text{ V}} = 4,444 \approx 4,5$. Dies wird durch eine Anschaltung eines Differenzverstärkers durch den einen, nicht benutzten, Operationsverstärker innerhalb des LM2902-Chips, umgesetzt. Hierbei wird ein Eingangswiderstand der beiden differenziellen Eingangssignale mit einem Wert von $5\text{ k}\Omega$ seriell vor den Eingängen des Operationsverstärkers geschaltet. Eine Verbindung mit Widerständen mit dem Faktor 4,5 werden einmal zu dem Ausgang und bei dem anderen Eingang zur Masse gekoppelt. Durch dieses wird dementsprechend ein differenzielles Signal zu einem asymmetrischen Signal mit einer Amplitude von maximal 9 V Aussteuerungsbereich ausgegeben. Die gesamte Schaltung um den vierfach Operationsverstärker ist in Abbildung 3.11 ersichtlich.

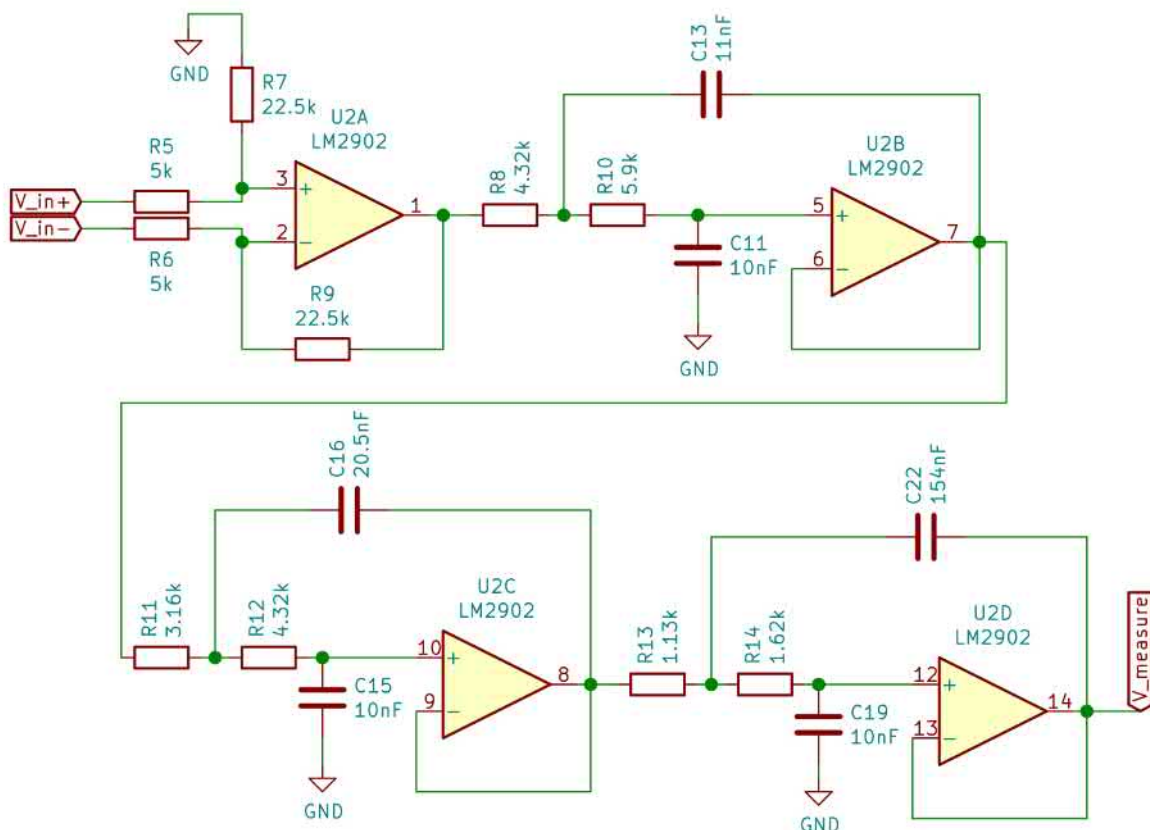


Abbildung 3.11: Gesamtbeschaltung des Vierfach-Operationsverstärkers

Würde man das Beispiel unter der Abbildung 2.1 auf einen nicht angepassten Tiefpassfilter legen, der eine zu geringe Grenzfrequenz aufweist, wird die Kurvenform stark verändert, da Oberschwingungen herausgefiltert werden.

3.5 Genauigkeit des Zeitstempels

Der Zeitstempel, der dafür benutzt wird, die gemessenen Daten langfristig abzuspeichern sollte so genau wie möglich sein. Dadurch, dass die Messschaltung mobil sein soll und diesbezüglich keine Internetverbindung garantiert werden kann, muss eine Möglichkeit der Zeitsynchronisation gegeben sein, die nicht durch einen anderen Computer, Server oder ähnliches stattfindet. Im Großraum Westeuropa gäbe es für diese Funktion zwei verschiedene Möglichkeiten, einmal durch die Nutzung eines DCF77 Zeitsignalempfängers und zum anderen durch einen GPS-Baustein. Der DCF77 ist ein Sendeturm aus Mainflingen, in der Nähe von Frankfurt, der mit einer langwelligen Frequenz von 77,5 kHz den aktuellen Zeit- bzw. Datumsstempel sendet. Die Trägerfrequenz wird durch eine Atomuhr abgeleitet und hat dementsprechend eine sehr geringe Abweichung von der nominalen Frequenz. Die Sendeleistung beträgt 50 Kilowatt, wovon ungefähr 30-35 Kilowatt wirklich abgestrahlt werden. Bis zu einer Entfernung von 2000 Kilometer kann dieses Signal durch einen empfindlichen Empfänger detektiert und genutzt werden. Funkuhren und kleinere Geräte benutzen dieses Signal um sich zu synchronisieren. Durch eine Amplitudenmodulation kann die aktuelle Zeit sowie das Datum ermittelt werden. Dadurch kann zwar auch der aktuelle Sekundenwert ermittelt werden, jedoch nur auf ca. 10 ms Genauigkeit. Um diese zeitliche Unsicherheit bei der Sekundenmarke zu verringern wurde eine pseudozufällige Bitfolge mit in das Signal eingebracht. Durch diese kann eine Kreuzkorrelation für eine deutlich genauere Detektion der Sekundenmarke genutzt werden, wobei dann die Genauigkeit der Sekundenmarke bei einigen Mikrosekunden liegt. Jedoch ist selbst dies ein Problem, da einerseits diese Genauigkeit nicht ein sicheres Limit darstellt, sondern über den Tages- und Jahreszeitraum fluktuieren kann. Andererseits, da nur ein Zeitsender existiert, ist bei Ausfällen oder Empfangsproblemen durch z.B. Überlagerung der Welle, der Zeitstempel nicht sicher [1]. Um dies zu umgehen kann ein Navigationssatellitensignal für die Erzeugung eines Zeitstempels benutzt werden. GNSS für *global navigation satellite system* ist der Überbegriff für ein Globales Navigationssatellitensystem. Hierbei befinden sich verschiedene, nicht geostationäre, Satelliten in dem Erdbereich in ungefähr zwanzigtausend Kilometer Abstand zur Oberfläche der Erde und senden jeweils ein Signal, in welchem deren Position und aktuelle Zeit, stammend aus einer Atomuhr, codiert ist. Dadurch, dass der Empfänger in den meisten geografischen

Regionen mehrere Satelliten gleichzeitig empfangen kann, lässt sich durch die Benutzung von vier Satelliten die GNSS-Zeit mit Genauigkeit bestimmen. Da in den normalen Regionen der Welt, außer den Polen, genug Satelliten den Punkt mit Signalen versorgen, ist der Empfang quasi überall garantiert. Je mehr Satelliten empfangen werden, desto genauer wird die rekonstruierte Positions-/Zeitinformation. Es existieren mehrere globales Navigationssatellitensysteme, unter anderem GPS des US-Amerikanischen Verteidigungsministeriums, GLONASS, das von dem Verteidigungsministerium der Russischen Föderation bereitgestellt wird, das neuartige Galileo der Europäischen Union und Beidou der Volksrepublik China [16]. Dadurch existieren viele verschiedene Satelliten in der Erdumlaufbahn, die für das Erlangen der Positions und Zeitdaten genutzt werden können. Somit wird durch verschiedene Systeme sichergestellt, dass äußere Einflüsse wie das Wetter weniger Störung auf die Erfassung nimmt [11]. Durch Benutzen von GNSS zum Erlangen eines Zeitstempels wird bei einem GPS-Empfänger für Endanwenderbereiche eine Genauigkeit von unter 60 ns im Verlauf von 99% der Zeit erreicht.

3.6 Entwicklung einer HAT-Platine für den Raspberry Pi

Der Single-Board-Computer *Raspberry Pi* der britischen Raspberry Pi Foundation besitzt verschiedene GPIO Pins (engl. General Purpose Input/Output Pins für Allzweck Anschlüsse für Ein- und Ausgabe), welche eine offene Schnittstelle für verschiedene Standards wie I2C, SPI, UART und digitale Abtastung von Pins bereitstellt. Diese Pins sind in einem Raster mit 2×20 Pins im Abstand von 2,54 mm angeordnet. Dieser Raspberry Pi, der nun zusätzlich zum Heimgebrauch, auch in der Industrie Verwendung findet, bietet aufgrund seines kleinen Ausmaßes von nur $85,6\text{mm} \times 53,9\text{mm}$ bei einer Höhe von 17 mm eine äußerst kompakte Größe. Verbaut ist in der vierten Generation ein Broadcom BCM2711 System-On-Chip Prozessor, der einer ARM-Cortex Architektur unterliegt. Als Betriebssystem wird meist auf eine Linux-Distribution zurückgegriffen, da die Kompatibilität zu anderen Betriebssystemen quasi nicht existiert. Hinzukommend zu den GPIO Pins sind standardisierte Anschlüsse wie USB, HDMI, Ethernet, Wi-Fi und Bluetooth verbaut. In dieser Arbeit ist vor allem die GPIO für die Akquise mit anschließender Verarbeitung der Daten sowie das Versenden über die Ethernet-Schnittstelle zur Außenwelt nötig. Es ist erstrebenswert, eine Platine mit verschiedenen Funktionen aufsteckbar auf das *Raspberry Pi* zu entwickeln. Diese Art von Platine nennt sich im Kreise der Raspberry Pi Foundation *Hardware Attached on Top* (Akronym HAT). Die verschiedenen benötig-

ten Bausteine werden auf einer Platine der Größe von $65,0\text{mm} \times 56,5\text{mm}$ aufgebracht und durch ein Printed Circuit Board miteinander verbunden. Hierbei wurde auf die Surface-Mount-Technology, also einer Oberflächenmontage, gesetzt, da somit deutlich weniger Platz durch Komponenten auf der Platine gebraucht wird, wie unter Abbildung 3.12 zu sehen ist. Das HAT, das in dieser Arbeit behandelt wird, wird in insgesamt vier Gruppen gegliedert.



Abbildung 3.12: Fotografie der bestückten HAT-Platine, Oberseite links, Unterseite rechts zu sehen

3.6.1 Spannungsversorgung

Die Spannungsversorgung muss hier bei dem HAT recht aufwendig gebaut werden, solange es andere Spannungen bereitstellen soll. Es sollen vor allem kleine Spannungsabfälle der Versorgungsspannung durch das 5-V-Schaltnetzteil besser ausgeglichen werden können. Das Pi an sich besitzt an dessen Eingang nur geringe Pufferkapazitäten. Dies stellt kein Problem dar, da der System-On-Chip durch einen Power-Management-Integrated-Circuit (PMIC) gespeist wird. Dieser PMIC, hier ein MxL7704 von MaxLinear versorgt den SoC mit seiner benötigten Spannung von ca. 3 V, abhängig von der aktuell benötigten Spannung, die bei verschiedenen hoher Taktrate des Chips höher bzw. niedriger ist. Der PMIC wird von den 5 V des Schaltnetzteils für das Raspberry Pi gespeist und setzt die Spannung dementsprechend tiefer. Durch das Tiefsetzen und präzise Schalten des PMICs werden Spannungsunterschiede recht schnell ausgeglichen. Da jedoch eine Operationsverstärkerstrecke auf der Platine implementiert wird, ist es von Nöten, andere Spannungen bereitzustellen. Es müssen die Spannungen 5 V, 3,3 V, +10 V und -10 V auf der Platine zur Verfügung gestellt werden. Sowohl die 5 V und die 3,3 V Spannung wird

vom Raspberry Pi bereitgestellt. Die 3,3 V stammen aus dem Ausgang des oben genannten PMIC, die 5 V direkt aus dessen Eingangsspannung, also durch das Schaltnetzteil. Um diese Spannung etwas abpuffern zu können, wird jeweils eine Kondensatorbank bei den Spannungen parallelgeschaltet. Diese hat ein besonders breites Frequenzband und sorgt dafür, dass die Spannung dennoch weitestgehend entstört wird. Dazu sind noch Elektrolytkondensatoren mit sehr hoher Kapazität zum Ausgleichen von Spannungsunregelmäßigkeiten verbaut. Hier sind dazu 1,5 mF Glättungskondensatoren eingesetzt. Dieser Wert ist der höchste, der in diesen Bemaßungen der Größe 10 mm x 10 mm erhältlich sind. Die gesamte Beschaltung ist in Abbildung 3.13 abgebildet.

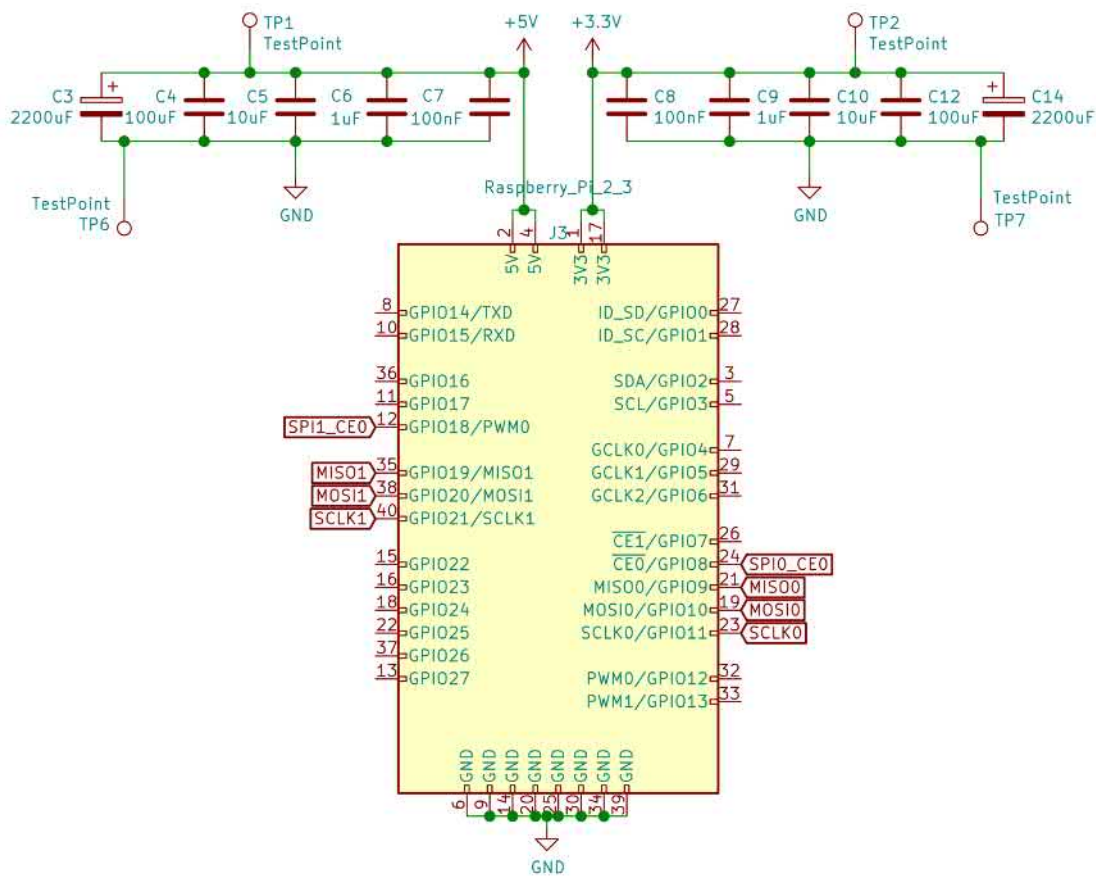


Abbildung 3.13: Schaltplan um den Connector des Raspberry Pi mit Glättungskondensatoren

Um die Versorgungsspannung des Operationsverstärkers zu generieren wird das Bauelement LTC3265 von Linear Technology eingesetzt. Dieser 21-beinige-Baustein beinhaltet zwei Ladungspumpen, womit Spannungen zum einen hochgesetzt und zum anderen invertiert werden können. Durch einen getakteten Kondensator wird dafür gesorgt, dass

ähnlich zu einer Dickson-Ladungspumpe der Kondensator geladen und entladen wird. Mit rapidem Schalten durch Transistoren und abgleichen durch eine Diodenschaltung wird erreicht, dass durch eine richtige Anordnung, entweder die Spannung vervielfacht, oder sogar invertiert werden kann. Durch Benutzen des in dem Chip verbauten Low-Drop-Längsregler, jeweils auf der invertierenden und erhöhenden Seite, werden die noch mit einer Welligkeit versehenen Spannungen stabil auf +10 V und -10 V geregelt. Diese Regelung ist einstellbar und deswegen muss zunächst ein Widerstandsverhältnis geschaffen werden. Laut Datenblatt des LTC3265 wird die positive Ausgangsspannung V_{LDO}^+ wie folgt berechnet:

$$V_{\text{LDO}}^+ = 1,2\text{V} \cdot \left(\frac{R_1}{R_2} + 1\right) = -V_{\text{LDO}}^- \quad (3.48)$$

Hierbei wurde der Wert des Widerstands R_2 dem Beispiel des Datenblatts entnommen, was 52,3 k Ω entspricht und der Widerstand R_1 durch Ausrechnen der Gleichung berechnet.

$$V_{\text{LDO}}^+ = 1,2\text{V} \cdot \left(\frac{R_1}{R_2} + 1\right) \quad (3.49)$$

$$\implies \frac{R_1}{R_2} = \frac{V_{\text{LDO}}^+}{1,2\text{V}} - 1 \quad (3.50)$$

$$\implies R_1 = R_2 \cdot \left(\frac{V_{\text{LDO}}^+}{1,2\text{V}} - 1\right) = 52,3\text{k}\Omega \cdot \left(\frac{10\text{V}}{1,2\text{V}} - 1\right) = 383\text{k}\Omega \approx 390\text{k}\Omega \quad (3.51)$$

Da die Berechnung des Widerstandsverhältnis von V_{LDO}^- exakt dem Schema folgt, ist hierbei der Widerstand R_1 exakt gleich dem des positiven Spannungszweigs, R_2 ebenfalls. Andere Komponenten, die laut Datenblatt des LTC3265 einen Vorteil bringen, um das Rauschen auf den Versorgungsspannungen zu minimieren, wurden vorgenommen. Die Beschaltung ist in Abbildung 3.6.1 gegeben. Bei dem Designprozess der Platine muss jedoch viel beachtet werden, da dieser Chip auf lediglich 4,4 mm breit und 6,5 mm lang ist und dabei 20 Anschlüsse auf die Platine bringt. Es muss zusätzlich noch ein Kontakt unterhalb des Chips getroffen werden. Dieser Chip besitzt aus Gründen der Temperaturabgabe eine Masseflächenkontaktierung, die fast 60% der Unterseite des Chips ausmacht. Dies stellt eine große Herausforderung dar, da mit alltäglichen Mitteln wie einem Lötkolben keine Verbindung unterhalb des Chips geschaffen werden kann. Somit muss auf eine Heißluft zurückgegriffen werden. Es wird also vorher eine Lotpaste aufgetragen, die aus mikroskopisch kleinen Kügelchen aus Sn60Pb38-Lot besteht, welche innerhalb eines Flussmittels aufgelöst sind. Danach wird der Chip genau auf die Position gesetzt und durch Heißluft mit einer hohen Temperatur von über 500 °C versetzt. Durch die Oberflächenspannung des flüssigen Lots beim Aufschmelzen des Lots wird der Chip

schlussendlich an die richtige Position versetzt und eine feste Verbindung der Anschlüsse mit der Platine entsteht. Somit stehen nun auf der Platine alle nötigen Spannungen zur Verfügung und es können die restlichen Komponenten mit aufgebracht werden.

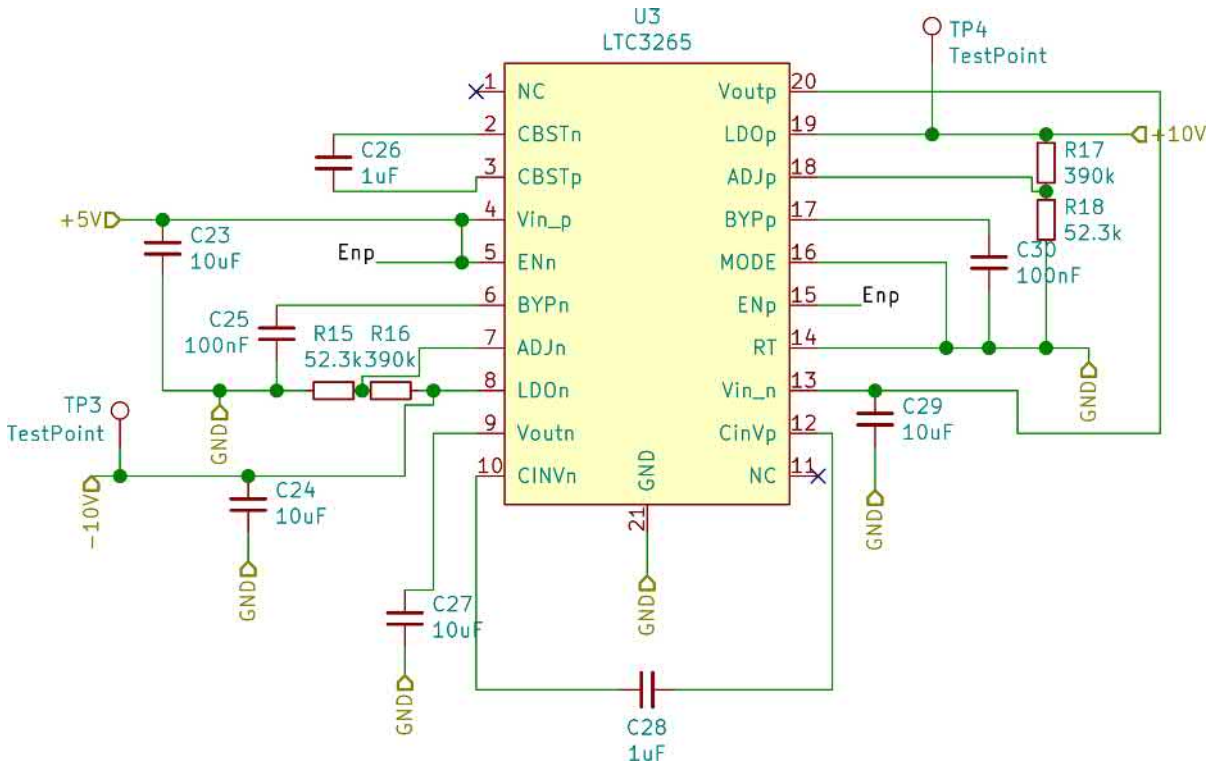


Abbildung 3.14: Schaltplan der Beschaltung des LTC3265 Chip

3.6.2 Analog/Digital-Umsetzung

Die Analog-zu-Digital-Umsetzung benötigt, wie in den Kapiteln zuvor, eine sehr hohe Abtastrate, die mit einer genug hohen Bittiefe versehen ist. Erhältlich war in dem Zusammenhang ein SAR-ADC von Texas Instruments mit der Teilenummer ADS8681. Dieser ADC besitzt einen Eingangskanal und erreicht eine Samplingrate von bis zu 1 MS/s, wobei der Eingangsbereich aufgrund eines programmierbaren Verstärkers am Eingang sich von $\pm 2.5\text{V}$ bis $\pm 12\text{V}$ strecken lässt. Hierbei eignet sich die Eingangskonfiguration von $\pm 10\text{V}$ für das System am besten, da der maximale Pegel, der ankommen könnte, genau bei $\pm 10\text{V}$ liegt. Die Signalfadlänge der SPI-Schnittstelle wird auf eine exakt gleiche Länge optimiert, da Frequenzen von über 50 MHz in den digitalen Schaltungen verlaufen. Somit wird sicher gestellt, dass keine Laufzeitunterschiede zwischen den Signalpfaden CLOCK und DATA entstehen. Die Beschaltung, die unter Abbildung 3.15 zu sehen ist,

wird aus den empfohlenen Bauelementewerten des Datenblatts unter Beachtung der Systemkonfiguration entwickelt.

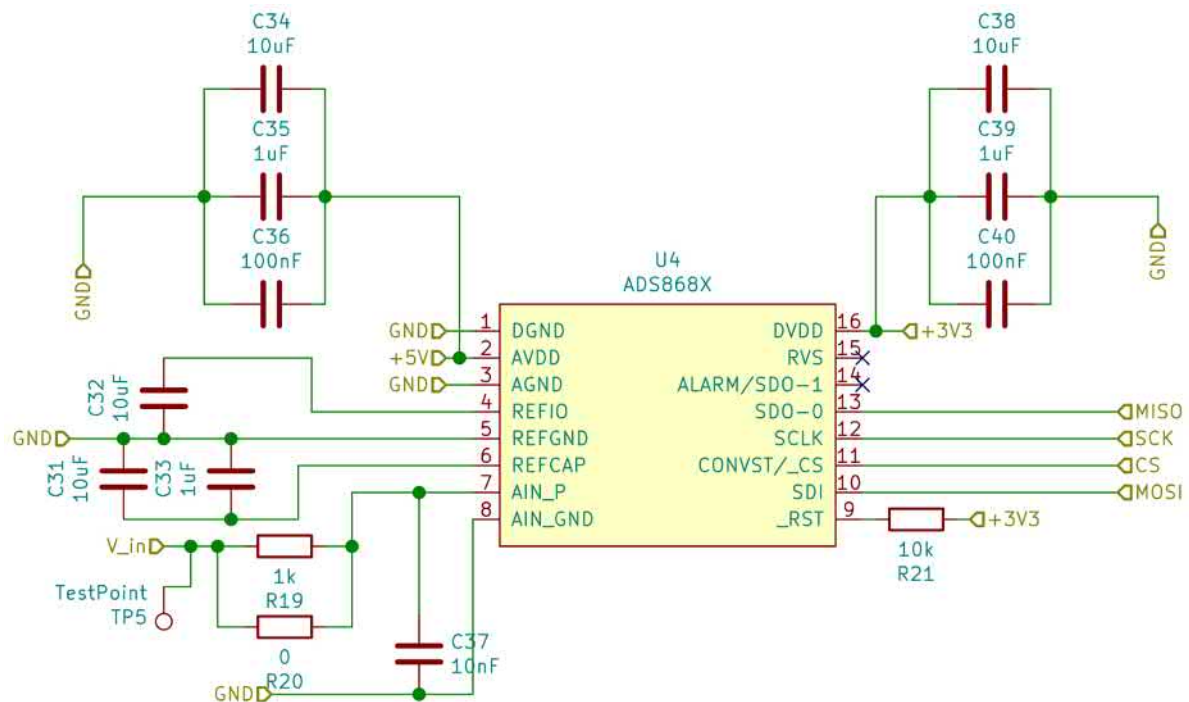


Abbildung 3.15: Aufbau des Schaltkreises um den Analog/Digital-Umsetzers ADS8681

3.6.3 Analogfilter

Bei der Umsetzung der analogen Filter sind im Vergleich zu den vorherigen Teilbereichen recht viele Freiheiten offen, jedoch sollte vor allem darauf geachtet werden, dass die Wege von den Eingängen des Operationsverstärkers immer gleich lang und kurz sein sollten. Durch die Einhaltung dieser Kriterien wird deutlich, dass eine Oberflächenmontage der einzelnen Bauelemente wie Widerstände und Kondensatoren mit einem kleinen geometrischen Ausmaß realisiert werden sollen. Das Benutzen des LM2902 in einer TSSOP-16 Form hat zur Folge, dass nur ein sehr geringer Platzbedarf nötig ist.

3.6.4 GPS

Ein, im Vergleich zu anderen Lösungen, recht einfach zu verwendender Baustein, der eine Funktionalität der Auswertung von GPS-Daten bereitstellt ist der NEO-M8M, der schweizerischen U-Blox AG, der sowohl für den Automotive- als auch industriellen Bereich entwickelt worden ist. Dieser Chip stellt verschiedene Interfaces zur Abfrage der Daten

bereit, zum einen ist eine serielle UART-Schnittstelle geteilt mit einem SPI-Interface verbaut, die eine solide Einbaumöglichkeit in eine Schaltung wie hier bereitstellen. Zum anderen wird auch eine USB-Schnittstelle bereitgestellt. Durch einen USB-Port, der auf der Platine montiert wird, können sowohl stationäre Geräte wie PCs und mobile Geräte wie Smartphones mit einem USB-Kabel auf die Werte des GPS-Chips zugreifen und diesen auslesen. Es wird zusätzlich noch ein DIP-Schalter auf der Platine verbaut, womit das Protokoll des GPS-Bausteins geändert wird. Durch Ändern der Schalterstellung lässt sich zwischen den Protokollen UART und SPI wechseln. Hierbei wird die Schalterstellung 1 benötigt, damit SPI aktiv ist. Es wird trotz der Anbindung aufgrund der Simplizität, den ein Dienst des Linux-Betriebssystems bietet, statt der SPI-Schnittstelle, ein USB-Kabel zur Verbindung des NEO-M8M und dem Pi benutzt.

Die Versorgungsspannung des RF-Chips beträgt 3,3 V, was im Gegensatz zu USB mit 5 V eine deutlich niedrigere Spannung ist. Die Schaltung um den USB-Port enthält einige Bauelemente, die dafür Sorge tragen, dass der RF-Baustein nicht Schaden von außen nehmen kann. Hierzu werden bei der Datenleitung zum Baustein $27\ \Omega$ Widerstände benutzt, die die Spannung der eingehender Datensignale auf ein richtiges Level setzt. Zusätzlich dazu wird eine Vorkehrung durch Suppressordioden getroffen, die die maximale Spannung, zum Beispiel durch elektrostatische Entladungen, begrenzen. Diese Suppressordioden haben eine symmetrische Charakteristik und sorgen dafür, dass transiente Vorgänge sehr gut abgehalten werden. Sie begrenzen die Spannung indem sie bei einer absoluten Spannung von über 5 V in einen leitenden Zustand gelangen. Somit kann davon ausgegangen werden, dass das Limit der USB-Datenspannung von 3,3 V nicht überschritten wird. Die gesamte Beschaltung ist unter Abbildung 3.17 zu sehen. Damit der Chip seiner Funktion der Auswertung von Signalen verschiedener Satelliten nachgehen kann, sollte, da dieser hinter einer Glasscheibe platziert wird, eine aktive Antenne bereitgestellt werden. Diese aktiven Antennen setzen einen Low-Noise-Amplifier in ihrem Gehäuse ein, der durch das Antennenkabel gespeist wird. Das Positionssignal, das durch den Anschluss durchgeleitet wird, beträgt zwischen 1575 MHz bis zu 1605 MHz, wobei die niedrigere Frequenz durch das US-Amerikanische GPS und die höhere Frequenz von dem russischen GLONASS-System stammt. Der Hersteller hat verschiedene Antennen im Angebot, welche sich rein durch den Anschluss unterscheiden. Somit ist eine breitbandige Übertragung nötig und das bei einer recht hohen Frequenz von über einem Gigahertz. Hierbei lässt sich zwischen einem Stecker mit dem Anschluss Sub-Miniature-A (SMA), einem Sub-Miniature-B (SMB) und Miniature Coax (MCX) auswählen. Aus Gründen der Verfügbarkeit wird auf einen SMB-Anschluss zurückgegangen, da dieser im Vergleich zu den anderen Typen eine sehr schnelle Verfügbarkeit bei der Beschaffung aufweist und durch einfache Mittel, wie Löten von Stecker und Buchsen an ein geeignetes HF-Kabel,

eine Verlängerung des Antennenkabels realisiert werden kann. Bei der Realisierung soll laut Hersteller bei allen Verbindungen und Leitungen ein Leitungswellenwiderstand Z_L von 50Ω benutzt werden. Eine nicht sachgemäße Installation kann dafür sorgen, dass Reflexionen innerhalb der Leitung auftreten. Diese sind bei einer Frequenz von 1,6 GHz angesiedelt und sehr prägnant. Um dies leichter zu veranschaulichen, lässt sich die Wellenlänge dieser elektromagnetischen Welle durch die Vakuum-Wellenlänge λ durch folgende Vereinfachung berechnen:

$$\lambda = \frac{c_0}{f_0} = \frac{299 \times 10^6 \frac{\text{m}}{\text{s}}}{1.6 \times 10^9 \text{ Hz}} = 18,6 \text{ cm} \quad (3.52)$$

Bei einer 18,6 cm langen Welle innerhalb der Übertragung ist auf viel zu achten. Der Leitungswellenwiderstand sollte so weit wie möglich angepasst werden. Eine geringe Fehlanpassung würde dort massiv zu Fehlern führen, wodurch die gesamte Übertragungsstrecke gestört werden kann. Hierbei stellt sich ein Problem, welches darauf basiert, dass die Platine, die in diesem Fall benutzt wird, nur zwei Kupferlagen besitzt. Dabei wird die eine Lage rein als Massefläche benutzt, die andere Lage führt das Signal. Diese Art der Signalübertragung ist eine sogenannte Microstrip-Transmission-Line, oder auch Mikrostreifenleitung genannt. Rein durch die geometrischen Ausmaße der Platine lässt sich der Wellenwiderstand vor der Fertigung der Platine berechnen. Dem Schaltungsentwickler wird jedoch die Dicke der dielektrischen Schicht aus dem Material FR-4, sowie die Dicke der Kupferschicht, die das Signal leiten soll, vorgegeben. Somit ist der einzig steuerbare Parameter bei der Entwicklung die Dicke des Signalpfads. Durch den Platinenhersteller **JLCPCB** werden die folgenden Parameter gegeben:

$$\varepsilon_{\text{FR-4}} = 4,6$$

$$d_{\text{FR-4}} = 1,6 \mu\text{m}$$

$$d_{\text{Cu}} = 35 \mu\text{m}$$

Durch Benutzen eines Mikrostrip-Calculator wie er in der Software KiCad enthalten ist, wird bei Eingeben der Werte und Setzen der Optimierungs-Bedingung auf einen Leitungswellenwiderstand von 50Ω bei einer Mittenfrequenz von 1,6 GHz die Breite des Mikrostrips auf 2,923 mm vorgegeben. Eine anschließende Verifikation durch eine online verfügbare Seite liefert die gleichen Ergebnisse [4]. Das Problem liegt jedoch darin, dass eine 3 mm breite Mikrostreifenleitung, also eine Bahn auf einer Platine, nicht möglich ist, da der benutzte Hochfrequenzstecker der Firma Molex nicht breit genug ist, ein 3 mm breite Leitung durchzuschleifen, da sonst eine Verbindung mit den außenliegenden Masseanschlüssen entstehen würde. Dazu ist es nicht möglich, eine 3 mm breite leitende

Schicht zu benutzen, da sonst die Anschlüsse 11, 12 und 10 des GPS-Chips miteinander kurzgeschlossen wären, da die Pinabstände des Bauelements sehr klein sind. Dadurch wird die Leitung auf den größtmöglichen Wert 1 mm gesetzt. In Abbildung 3.16 kann man diese Streifenleitung sehen. Diese Verkleinerung hat jedoch eine Fehlanpassung zur Folge. Eine Analyse durch den Mikrostrip-Calculator stellt somit einen Leitungswellenwiderstand von $84,55 \Omega$ fest.

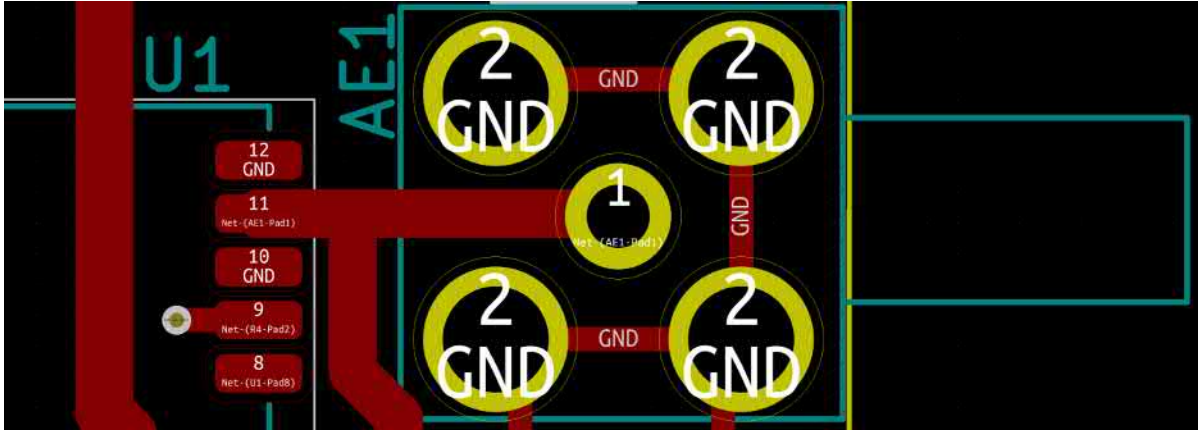


Abbildung 3.16: Die Verbindung des Pin 1 des Antennenanschlusses AE1 und Pin 11 des GPS-Chips mit der Bezeichnung U1 ist die fehlangepasste Leitung mit 1 mm Dicke

Daraus resultieren in dieser Leitung Reflexionen, welche bedeuten, dass zu einem Teil Wellen zum RF-Chip gelangen, und ein anderer Teil den Weg zur Antenne zurück reflektiert wird. Somit verschlechtert sich der allgemeine Signal-zu-Rauschabstand signifikant. Eine Berechnung des Reflexionsfaktors, also einer absoluten Angabe, wie stark das Signal beim Eingehen zur Mikrostreifenleitung reflektiert und transmittiert wird, erfolgt durch:

$$r(Z_L) = \frac{U_{\text{rück}}}{U_{\text{hin}}} = \frac{Z_L - Z_{\text{PCB}}}{Z_L + Z_{\text{PCB}}} = \frac{50\Omega - 84,55\Omega}{50\Omega + 84,55\Omega} = 0,2567 \quad (3.53)$$

Somit werden absolut gesehen 25,6% der Leistung der Welle wieder zurück zur Antenne reflektiert. Würde man ein vier-lagiges PCB verwenden, wäre eine absolut genaue Möglichkeit der Einstellung des Leitungswellenwiderstands möglich, da dort die Berechnung des Leitungswellenwiderstands aufgrund von zwei planaren Masseflächen ober- und unterhalb der Hochfrequenzleitung in ein kontrollierbares Maß fällt.

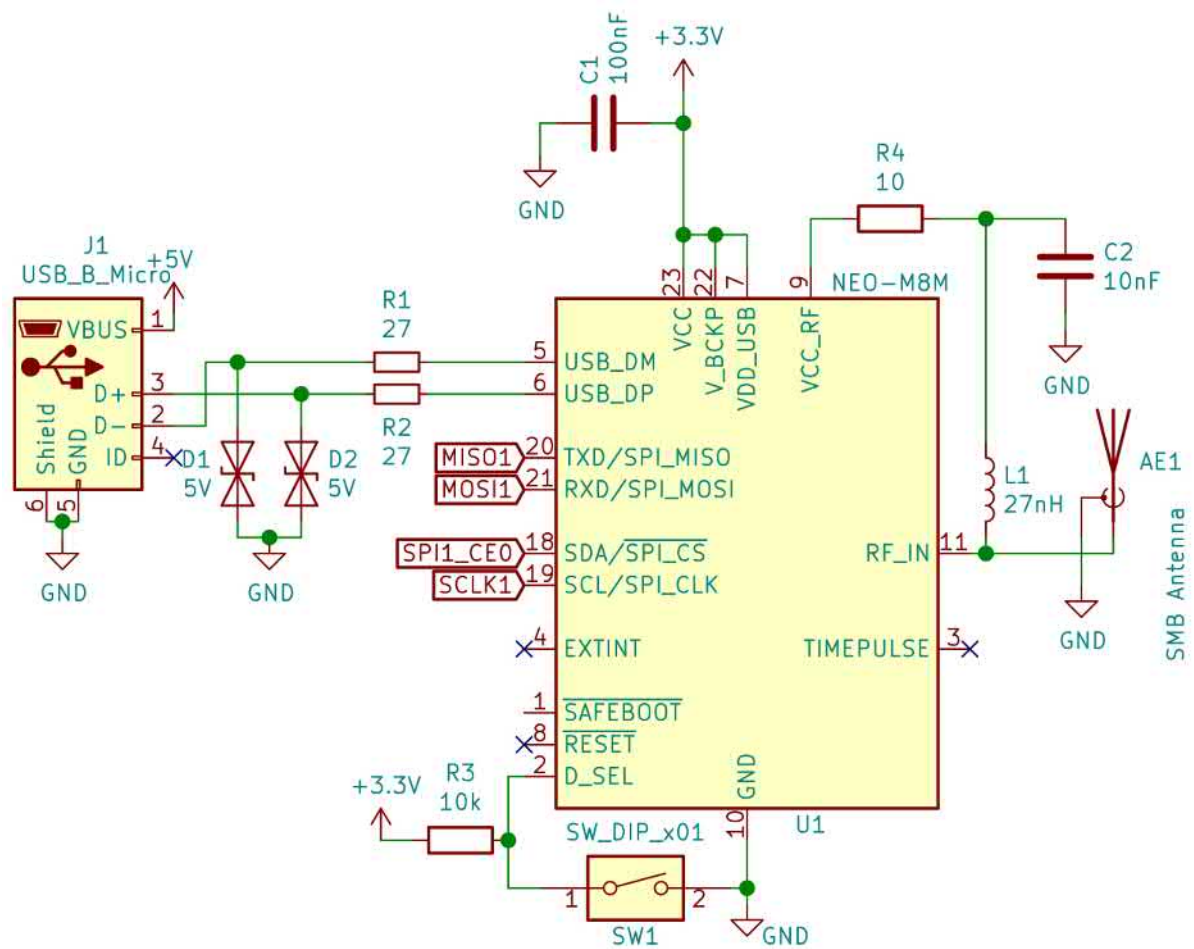


Abbildung 3.17: Schaltplan der Beschaltung des GPS-RF Chips NEO-M8M der U-blox AG

4 Programmierung der Software und Validierung

4.1 Programmierung

4.1.1 Auswertung der GPS Daten durch gpsd

Da bei der Implementierung des GPS-Empfängers auf einem Raspberry Pi das Betriebssystem Linux Platz einnimmt, lassen sich auf kurze Sicht die gesamte Funktionspalette des GPS-Empfängers per USB implementieren. Eine verbaute, aber nicht benutzte Art der Adressierung des GPS-Bausteins funktioniert über die SPI-Schnittstelle des Chips. Diese Implementation wird jedoch aufgrund der Simplizität des gpsd-Dienstes, der ohne starke Veränderung nur die USB-Schnittstelle ausliest, nicht benutzt. Stattdessen wird ein kurzes USB-Kabel benutzt, um den Micro-USB Port des HATs mit einem USB-Port des Raspberry Pi zu verbinden. Hier wird durch den GPS-Baustein eine virtuelle serielle Konsole, ähnlich einer RS232 bzw. UART-Schnittstelle emuliert. Bei einer Baudrate von 9600 Symbole pro Sekunde mit acht Bits und einem Stoppbit pro Symbol ergibt das eine 8b9b-Übertragung. Hierbei wird der Übertragungsstandard NMEA0183 der National Marine Electronics Association benutzt. Dieser Standard stellt die Symbolik der Übertragung dar, wie Daten über ein serielles Muster übertragen werden.

```
$GPGGA,HHMMSS.ss,BBBB.BBBB,b,LLLLL.LLLL,1,Q,NN,D.D,H.H,h,G.G,  
g,A.A,RRRR*PP
```

Dies stellt ein typisches Übertragungsmuster von dem GPS-Empfänger an den Rechner dar. Die Zeichenkette mit insgesamt 74 Stellen wird dafür nutzbar gemacht, alle nötigen Informationen, wie Datumsstempel, Position in Längen- sowie Breitengrad und Höhe und auch Informationen zur Übertragungsstrecke wie Anzahl der GPS-Satelliten, die zur

Auswertung benutzt werden und GPS-Qualität Aussage zu treffen. Zuletzt finden sich noch, um in dem NMEA-Protokoll wenig Fehler zu erzeugen, zwei Byte zur Prüfsumme Platz. Durch diese Ankündigung wie oben zu sehen werden danach die Platzhalter-Buchstaben durch die jeweiligen Werte ersetzt und somit existiert eine gewisse Übertragungssicherheit, weil diese von verschiedenen Protokollversionen benutzt werden kann, da das Protokoll vorgibt, welche Daten im Nachhinein wie folgen. Diese Nachrichten werden periodisch jede Sekunde übertragen und lassen somit Aufschluss auf die Beschleunigung sowie Bewegung im Raum schließen. Die Auswertung der periodisch an den Computer gesendeten seriellen Daten wird durch den Dienst `gpsd` deutlich vereinfacht. Dieser Dienst nimmt die virtuelle Konsole unter der Gerätedatei `/dev/ttyACM0` an sich, überwacht diesen und wertet diese in leserliche Form aus. Es werden die Positionsdaten, Zeit und GPS-spezifische Daten durch Bibliotheken leicht für verschiedene Clientprogramme zur Verfügung gestellt.

Gewinnung des Zeitstempels mit hoher Genauigkeit

Der Zeitstempel des Betriebssystems wird meist aus einem Protokoll namens NTP, für Network-Time-Protocol, generiert. Im Falle des Linux Betriebssystem übernimmt der Dienst `ntpd` diese Aufgabe. Dort lassen sich über die Konfiguration unter `/etc/ntp.conf` verschiedene Server zur Synchronisierung der Zeit benutzen. Dieser Dienst lässt sich auch so modifizieren, dass er die Daten des `gpsd`-Dienstes übernimmt und mit dem exakten Zeitstempel als systemweite Zeit überträgt. In einer Windows-Betriebsumgebung ist dieser Schritt deutlich schwerer, da bei der Programmierung von Linux Programmen zwischen den verschiedenen Zeitarten wie einer monotonischen Uhrzeit und einer echten Uhrzeit gewählt werden kann. Die monotonische Uhrzeit beschreibt eine Uhr, die immer monoton verläuft, das heißt, dass selbst nach einer Synchronisierung durch NTP eine Sekunde in der Laufzeit immer einer Sekunde Unterschied entspricht. Bei einer Echtzeit-Uhr innerhalb des Betriebssystems ist dieser Abstand nicht gegeben. Falls ein Wartezeitraum von 10 Sekunden gewollt ist und während diesem Zeitraum eine Synchronisierung stattfindet, welche im äußersten Fall mehrere Sekunden der Uhrzeit ändert, so ist der gewollte Unterschied von 10 Sekunden nicht zu erreichen, sondern um die Synchronisierungsdifferenz verschieden. Eine Benutzung des Befehls `ntpq` gibt die aktuell benutzte Zeitquelle aus. Der Dienst erkennt selbst, ob es zum Beispiel einen sog. Falseticker gibt. Hierbei wird von dem Dienst erkannt, ob die Zeitquelle valide Daten ausgibt. Invalide Daten könnten daraus resultieren, dass dieser Zeitstempel nicht mit dem rechnerinternen Zeitgeber übereinstimmt, also kein absoluter Zeitfehler, sondern ein relativer Zeitfehler, stattfindet.

Zudem ist die Erkennung eines absoluten Zeitfehlers möglich. Da der NTP-Dienst mehrere NTP-Zeitserver anspricht ist die Falschinformation eines oder mehrerer Zeitserver herausfilterbar und lässt sich als Falseticker markieren. Falseticker werden nicht für den Zeitstempel als Informationsquelle genommen. Bei Benutzen der `gpsd`-Zeitquelle wird jedoch ein großes Problem deutlich. Die Zeitgenauigkeit des `gpsd`-Clients wird hierbei auf über 10 und unter 20 μ s betitelt. Die Genauigkeit des Zeitstempels des NEO-M8M und des GPS-Protokolls sollten jedoch bei ca. 30 ns liegen. Dies ist dem Protokoll resultierend. Das NMEA-Protokoll über eine serielle Schnittstelle ist nicht echtzeitfähig, da der Computer nicht rekonstruieren kann, wann dieser Zeitstempel, den der Chip ausgibt, als Sekundenmarke zu definieren ist. Somit ist durch diese Ungewissheit keine genaue Möglichkeit gegeben, den Zeitstempel zu definieren.

PPS-Signal zur Gewinnung eines Taktes durch den Baustein NEO-M8M

Ein Zeitpuls-Signal, auch Pulse-Per-Second-Signal, kann diesen Nachteil des NMEA-Protokolls ausgleichen. Dieser Zeitpuls lässt sich bei der Konfiguration des NEO-M8M zwischen 0,25 Hz und 10 MHz einrichten. Im Linux Betriebssystem existieren speziell auf dem Raspberry Pi Möglichkeiten diesen Zeitpuls als Taktflanke zu detektieren und dadurch an den NTP-Dienst weiterzugeben. Hierbei muss jedoch eine spezielle Einstellung bei dem Kompilieren des Linux-Kernels gesetzt worden sein. Durch dieses PPS-Signal und nachfolgender Konfiguration innerhalb der NTP-Konfiguration lässt sich dieses als explizites Signal zur relativen Synchronisierung einstellen. Dieses Zeitsignal kommt dementsprechend immer im Bereich von unter 30 ns um den Nullpunkt des Sekundenwechsels, welches es leichter macht als eine Synchronisierung mit einer langsamen oder schnelleren Frequenz als 1 Hz. Bei einer nachfolgenden Untersuchung der Ausgabe von `ntpq` resultiert bei einer ausgegebenen Genauigkeit von unter 1 μ s, was dem erwartenden Ergebnis sehr nahe kommt. Hierbei liegt dieser sogenannte Jitter, also die Abweichung, die zwischen einem Puls liegt, unter einer Mikrosekunde, die der NTP-Client aufnimmt. Durch diese Möglichkeit der Synchronisierung der lokalen Rechnerzeit mit der Zeit aus dem PPS-Signal liegt eine deutliche Vereinfachung des Programmieraufwandes vor. Durch den Linux-Systemcall `clock_gettime` lässt sich diese Daten mit wenigen Nanosekunden Fehlertoleranz ausgeben und direkt in jedem Programm benutzen.

Möglichkeit der Nutzung des NTP-Servers der FAU

Das Regionale Rechenzentrum der Friedrich-Alexander-Universität bietet für Anwender insgesamt vier verschiedene NTP-Server an. Ein NTP-Server hat die Aufgabe eines Zeitserver, der seinen genauen Zeitstempel über das Network-Time-Protocol an verschiedene andere Clients überbringt. Dieser Zeitstempel selbst wird über verschiedene Quellen generiert. Das Rechenzentrum stellt zwei Zeitquellen des DCF77-Zeitsignals und zwei durch einen GPS-Empfänger bereit. Laut dem Rechenzentrum nutzt der Zeitserver `ntp0.fau.de` als Zeitquelle eine externe Uhr vom Typ GPS167 der Firma Meinberg, welche aus GPS gespeist wird. Hierbei stellt jedoch die Übertragungsstrecke ein relativ großes Problem dar, da diese eine Latenz aufweist. Benutzt man auf einer Platine einen Baustein, der sowohl den GPS Zeitstempel, als auch eine genaue Taktflanke (PPS) mit einer genauen Frequenz, ausgibt, lässt sich die Latenz, die zwischen dem Baustein, hier der U-Blox NEO-M8, und dem auswertenden Computer, dem Raspberry Pi, herrscht, auf beinahe Null Mikrosekunden vereinfachen. Der Ethernetanschluss und das Benutzen des NTP-Clients werfen auf das System eine Latenz, die in guter Näherung so groß wie der Round-Trip-Time eines Pings der ICMP Pakete ist. Diese beschreibt eine gute Näherung an die physikalisch auftretende Latenz, dass ein Netzwerkpaket von Server zu Client und wieder zurück zum Client benötigt. Hierbei ist jedoch nicht die Lichtgeschwindigkeit vorrangig der ausschlaggebende Faktor, sondern die Verzögerungen des Paketes durch Netzwerkhardware wie Switches und Netzwerkkarten. Im Universitätsnetzwerk liegt diese bei ca. 30 μ s. Das Precision-Time-Protocol ist hierbei eine Weiterentwicklung des im Jahre 1985 spezifizierten Protokoll NTP und bietet in solchen Ethernet-Netzen eine sehr hohe Genauigkeit und sorgt für eine Synchronisierung der Zeit zwischen Server und Client, die im garantierten Mikro- bis Nanosekundenbereich, je nach Implementierung, liegt. Hierbei verwendete Hardware ist jedoch deutlich teurer, da die durch den TCP/IP-Stack hervorgebrachte Latenz ausgeglichen werden muss. Somit sind spezielle Switches, die eine gleiche Zeit für Vor- und Rücklaufendes Paket garantieren, von Vorteil. Wird im Client eine spezielle Hardwarekarte für die Evaluierung der Zeit benutzt, lassen sich in Verbindung mit vorherig genannter Hardware eine Genauigkeit von unter 100 Nanosekunden erreichen [9]. Es lässt sich aber dennoch sagen, dass eine ausreichende Genauigkeit von unter 30 μ s rein durch das Benutzen des Zeitserver der FAU erreicht wird.

4.1.2 Algorithmus zur Bestimmung der Frequenz

Um die Frequenz zu bestimmen gibt es verschiedene Möglichkeiten, wie zum Beispiel einen Regelkreislauf, bei dem durch eine Regelung versucht wird, ein Cosinus-Signal auf exakt dieselbe Frequenz wie die Netzfrequenz mit einem Regelkreis zu stellen. Dieses ist in einem Simulink-System leicht implementierbar, jedoch bedeuten regelungstechnische Anwendungen im Vergleich zu Algorithmik eine deutliche Verschlechterung der Rechenzeit für ein System. Eine sogenannte Frequency-Locked-Loop findet z.B. bei einer Synchronisierung eines Wechselrichters mit dem Stromnetz statt. Hierbei wird entgegen dieser Messung der Phasenwinkel der Netzspannung zum internen Oszillator benötigt. In dieser Arbeit liegen jedoch andere Ziele, wodurch die Rechenzeit so gering wie möglich gehalten werden soll. Ein anderer Weg wäre es, eine Non-Uniform-DFT einzusetzen. Dies ist eine spezielle Form der Diskreten-Fourier-Analyse eines periodischen Signals. Hierbei ist jedoch ein großer Unterschied darin, dass die Frequenzabstände der Diskreten-Fourier-Transformation nicht äquivalenten Abstand besitzen. Somit kann an den gewünschten Punkten, vornehmlich um die 50 Hz Netzfrequenz eine Häufung der Stützstellen der Fourier-Transformation erzeugt werden und dadurch durch sehr viele Abtastpunkte das Maximum der Amplitude an einer Stützstelle gesucht werden. Umso mehr Stützstellen benutzt werden, desto größer ist die Genauigkeit der Frequenzanalyse. Die Bibliothek **NFFT** der Technischen Universität Chemnitz stellt für diesen Zweck eine Funktion bereit, die diese Non-Uniform-Discrete-Fourier-Transformation versucht in eine Fast-Fourier-Transformation zu verpacken, um die Rechenzeit deutlich zu verringern. Die FFT stellt eine kleine Optimierung der DFT dar, da die FFT durch gewisse Verknüpfungen die Rechenzeit immens einsparen kann, unter der Bedingung, dass der Eingangsvektor ein Vielfaches von der Zahl 2 ist. Das Benutzen der Bibliothek NFFT wäre eine sehr gute Maßnahme der Frequenzbestimmung, jedoch ist selbst dies auf einem kleinen Single-Board-Computer mit sehr hohen Rechenzeiten verbunden. Auf einem System mit mehr Rechenleistung wie einem x86-basierten Computer wäre dies eventuell ein Schritt, der die Frequenz besser approximieren kann. Um jedoch den Aufwand so gering wie möglich zu halten, ist es ratsam die Frequenz aufgrund der Nulldurchgänge zu bestimmen. Dies stellt eine ähnliche Vorgehensweise dar, wie sie auch auf reinen Hardwareschaltungen zu implementieren ist. Die Hardware setzt durch einen Hysterese-Durchgang bei dem Nulldurchgang einen Zähler hoch, der dann durch einen kleinen Controller ausgewertet werden kann, falls dieser in einer Sekunde diesen Zähler auswertet. Diese Hardware hat jedoch aufgrund der Genauigkeit eines Schwingquarzes einen deutlichen Nachteil, und zwar, dass es durch den verwendeten Quarz genauer bzw. ungenauer wird. Diese Schwingquarze können im Vergleich zu dieser Schaltung sehr teuer sein und zuletzt

bleibt der Nachteil, dass die Oberschwingungen nicht mitgemessen werden können. Somit müsste, um das zu erreichen, dennoch eine digitale Aufzeichnung des Signals folgen. In dem Fall der digitalen Aufnahme der Spannungswerte existieren in einer Sekunde so viele Abtastpunkte wie die Abtastrate F_s , das bedeutet, dass beim ADS8681 eine Million Werte der Spannung in einer Sekunde aufgezeichnet werden. Durch diese vielen Stützstellen, unter der Bedingung, dass die Abtastpunkte immer genau äquivalent verteilt sind, lässt sich die Frequenz durch den Abstand und die Anzahl der in diesem Abstand auftretenden Nulldurchgänge bestimmen, indem man die Definition der Frequenz umkehrt. Hierbei stellt die Variable T_{zc} die Zeit zwischen zwei direkt aufeinanderfolgenden Nulldurchgängen gleicher Art der Steigung dar:

$$f_{\text{Netz}} = \frac{1}{T_{zc}} = \frac{n}{T_s \cdot (x_{\text{Rechts}} - x_{\text{Links}})} = \frac{F_s \cdot n}{x_{\text{Rechts}} - x_{\text{Links}}} \quad (4.1)$$

Hierbei stellt n die Anzahl der jeweils positiv-steigenden Nulldurchgänge dar und der Wert x_{Rechts} den Ort des rechten Nulldurchgangs und x_{Links} dann den linken. Durch diese Berechnung kann die Frequenz schnell mit Programmierung bestimmt werden. Wird die Anzahl n erhöht, also pro Sekunde weniger Frequenzen erhoben, so ist die Genauigkeit der Messung höher, da Fehler aufgrund der langen Messzeit abnehmen. Es unterliegt jedoch dem Problem, dass die Präzision der Frequenz sehr gering ist, falls die Abtastfrequenz klein ist. Wenn die Abtastfrequenz zehnmal so hoch ist, ist der Nulldurchgang dann dadurch zehnmal genauer bestimmbar, da innerhalb der zwei Punkte bei der kleineren Abtastfrequenz, insgesamt zehn Punkte mehr dazwischen liegen. Eine Möglichkeit diese Ungenauigkeit zu verbessern ist eine Interpolation der Spannungswerte zu implementieren. Dadurch, dass die Spannung innerhalb dieses Nulldurchgangs sich im Vergleich zu den Hoch- bzw. Tiefpunkten sehr linear verhält, da im Nulldurchgang die Steigung am höchsten ist, ist die Möglichkeit der Linearisierung und Entwickeln der Nullstelle gut gegeben. Durch eine Interpolation lässt sich der genaue Nulldurchgang näher bestimmen. Hierbei gibt es meist verschiedene Methoden der Interpolation, unter anderem einer linearen Interpolation, ähnlich dem Newton-Verfahren oder einer komplexeren Interpolation durch kubische Splines, bei der eine Funktion dritten Grades durch den Verlauf der Spannungskurve am Nulldurchgang gelegt wird und damit der exakte Nulldurchgang approximiert wird. Durch eine Erhöhung der Abtastrate wäre jedoch ein exakteres Messen des Nulldurchgangs möglich.

4.1.3 Oberschwingungsmessung

Da die Netzspannung schon als digitale, diskretisierte Wellenform verfügbar ist, lässt sich hier eine Fast-Fourier-Transformation durchführen. Dieser FFT-Algorithmus nimmt die gesamten in der Abtastzeit T existierenden Stützpunktstellen des Signals und wandelt dies in ein Frequenzspektrum um, aus dem die Amplitude der Oberschwingung der jeweiligen Frequenz genommen werden kann. Jedoch ist dieses Spektrum kein kontinuierliches, sondern ein diskretisiertes Spektrum. Dies bedeutet, dass dieses Spektrum von der Frequenz 0 Hz bis hin zu $\frac{F_s}{2}$ in einzelne Frequenzstützpunkte im Abstand Δf unterteilt ist. Dadurch findet ein Amplitudenfehler statt, da die Amplitude meist nicht an den gewollten Frequenzpunkten gemessen wird. Die Linienzahl N ist dadurch definiert, wie viele Abtastpunkte für die Auswertung benutzt werden. Dies bedeutet, dass N für eine Sekunde Abtastung genau der Abtastfrequenz F_s entspricht. Somit sind für dieses Beispiel genau $\frac{F_s}{2}$ Amplitudenwerte mit dem Abstand $\frac{F_s}{N} = 1\text{Hz}$ verfügbar. Würde man die Aufnahmezeit erhöhen, ergibt sich ein geringerer Fehler [7]. Dadurch, dass hier jedoch wie bei der Frequenz ein Update pro Sekunde existiert, wird die letzte jeweilige Sekunde als Abtastzeitraum benutzt.

$$N = F_s \cdot T = \frac{F_s}{\Delta f} \quad (4.2)$$

Die FFT fordert aufgrund ihres Aufbaus einen Eingabevektor der Größe 2^N , wobei N einer ganzzahligen positiven Zahl entspricht. Durch diese Bedingung wird das Ende eines Eingabevektors mit Nullen aufgefüllt, sodass er diese Bedingung erreicht. Durch das sogenannte Zero-Padding werden so viele Nullen hinten angebracht, wie nötig sind. Dies hat zur Folge, dass der Eingabevektor größer ist, und somit wird der Ausgabevektor nun auch größer. Es sind dadurch mehr Frequenzstützstellen in der Auswertung verfügbar. Dies bedeutet jedoch nicht, dass die Genauigkeit höher wird, sondern es muss der Ausgabevektor nicht mehr interpoliert werden, um die Amplitude an einer bestimmten Frequenz auszuwerten. Eine Interpolation des Ausgabevektors kommt der Erhöhung der Auflösung gleich. Dadurch, dass die Rechenzeit der Fast-Fourier-Transformation so schnell ist, ist es nicht nötig, die Länge des Vektors zu reduzieren, sondern eine Verdopplung der Länge des Vektors kommt einer Verdopplung der Rechenzeit nahe.

$$\mathcal{O}(2N \cdot \log 2N) = \mathcal{O}(2N \cdot (\log 2 + \log N)) \approx 2 \cdot \mathcal{O}(N \log(N)) \quad \text{für } N \gg 2 \quad (4.3)$$

Die Komplexität einer Diskreten-Fourier-Transformation ist dabei stark verschieden, da diese alle mathematischen Operationen nacheinander folgend abarbeitet. Hierbei ist die

Komplexität $\mathcal{O}(N^2)$, was bei einer verdopplung des Eingangsvektors eine vervierfachung der Rechenzeit mit sich zieht. Selbst bei einem Single-Board-Computer mit geringer Leistung ist die Komplexität der FFT so gering, dass es möglich ist, die Größe des Eingabevektors künstlich zu verlängern. Somit ist die Genauigkeit der Amplitudenmessung deutlich besser, als es zuvor möglich war.

4.1.4 Abfragen der Spannungswerte des A/D-Umsetzers

Um die Daten des A/D-Umsetzers ADS8681 in digitaler Form in den Computer, hier den Raspberry Pi zu bekommen, wird hier eine SPI-Schnittstelle benutzt. Es gibt verschiedene Möglichkeiten der Ansteuerung. Der Linux Kernel gibt eine Möglichkeit der Nutzung der Gerätedatei unter `/dev/spidev1.0` preis. Diese Gerätedatei ist aus dem Userspace, also mit schlichten Benutzerrechten, ansprechbar und somit sind keine Bibliotheken für diese Art der Programmierung notwendig. Eine Möglichkeit wäre hier, mit der Sprache C/C++ echten Programmcode zu erstellen, der auf dem System läuft. Die nächste Möglichkeit bietet sich, den echten Linux-Treiber `spidev` zu benutzen, der mit der Header-Direktive `#include <linux/spi/spidev.h>` eingebunden werden kann. Dieser Treiber agiert auf einer Systemebene und kann somit nochmals eine deutliche Verschnellerung der Datenaufnahme erzielen. Für diesen Gerätetreiber existieren Bibliotheken für die Skriptsprache Python namens `spidev`, die diesen Gerätetreiber in der Sprache Python zur Verfügung stellen kann. Die letzte Möglichkeit wäre jedoch den Programmcode durch eine Bibliothek umzusetzen. Hierbei existiert von dem Programmierer Mike McCauley eine **Bibliothek**, die die exakten SPI-Zugriffe durch Benutzen von Low-Level-Hardwareoperationen umsetzt. Dies ist zwar die aufwendigste Möglichkeit den Programmcode umzusetzen, jedoch verspricht dies die beste Ausnutzung von CPU-Zeit bei gleichem Ergebnis, da diese Umsetzung ab vom Betriebssystem verschiedene Register des SoCs setzt und so die Daten des SPI-Slaves, also dem ADS8681-Chip, schneller auslesen kann.

Problematiken der genauen Abtastung: Systemnahe Programmiersprache

Der ADS8681 hat die Eigenschaft, eine Abtastfrequenz von bis zu einem Megahertz zur Verfügung zu stellen. Da dieser Chip ein SAR-ADC ist, bedeutet dies, dass eine Abtastung des Signals genau exakt den Kehrwert der maximal zur Verfügung gestellten Abtastrate darstellt. Hierbei entspricht die Zeit dementsprechend $1\ \mu\text{s}$, die für eine gesamte Umsetzung des analogen Spannungswertes in einen digitalen Wert dauert. Diese Mikrosekunden erfordern im Zeitbereich von Computern sehr viel Genauigkeit. Hierbei

darf es um keine Verzögerungen kommen, da sonst die gesamte Datenerfassung nichtig ist. Ist dieser Mikrosekunden-Abstand nicht gegeben, kann es passieren, dass die Frequenzmessung nicht akkurat ist. Somit muss sichergestellt werden, dass das Betriebssystem immer zur richtigen Zeit die Abtastung beginnt. Dadurch, dass Linux für die Kommunikation eine C-Schnittstelle durch Syscalls bereitstellt und die Programmiersprache C die hardwarenaheste Sprache bis auf Assemblersprachen ist, wird eindeutig klar, dass die Sprache C erforderlich ist.

Echtzeitverhalten: Realtime-Linux-Kernel

Ein erster Schritt zur fehlerfreien Abtastung ist es, einen sogenannten Realtime-Kernel, also ein Betriebssystem-Kern mit Unterstützung einer Echtzeitfähigkeit zu benutzen. Der Kernel eines Linux Betriebssystem muss direkt für die benutzte Hardware extra kompiliert und nachher benutzt werden. Hier muss der Quelltext des Kernel des Raspberry-Pi im ersten Schritt mit den richtigen Patches von der Plattform [Github](#) heruntergeladen werden. Dieser Quelltext beinhaltet alle Änderungen, die die Lauffähigkeit des Kernels für den Raspberry-Pi garantieren, wie zum Beispiel die Konfiguration für den SoC BCM2711 und Dateien für dessen Bootloading-Prozess. Durch den Preempt_RT-Patch der unter [Kernel.org](#) herunterladbar ist, wird die Echtzeitfähigkeit in den Quelltext eingepflegt. Dieser Patch hat den Vorteil, dass dieser die allgemeinen Funktionen des Linux-Betriebssystem für Nutzer lediglich erweitert, aber die Programmierung keine neuen Funktionen benötigt. Dieser Patch muss nun durch den Befehl *patch* innerhalb des Kernel-Codes eingebracht werden und kann danach mit dem Befehl *make menuconfig* konfiguriert werden. Die nötigen Konfigurationen sind:

```
HIGH_RES_TIMERS=y
CONFIG_PREEMPT_RT_FULL=y
CONFIG_HZ_1000=y
CONFIG_HZ=1000
```

Mit folgendem Kompilieren des Linux-Kernels mit allen nötigen Patches ist dieser Kernel startfähig. Durch setzen des richtigen Pfades unter */boot/config.txt* mit der Direktive *kernel=kernel_rt.img* wird beim nächsten Start der richtige Kernel geladen. Dieses ist durch den Befehl *uname -a* in der Linux-Shell zu kontrollieren. In dem richtigen Falle gibt der Kernel eine folgende Nachricht aus:

```
Linux raspberrypi 5.4.51-rt32-v8 #1 SMP PREEMPT_RT Thu Oct 15 13:22:10 CEST
2020 aarch64 GNU/Linux
```

Der Zusatz `PREEMPT_RT` gibt nun an, dass die Konfiguration vollständig echtzeitfähig ist und dementsprechend die Bedingung des Kernels so gut wie möglich getroffen werden kann.

Um ein Programm in dem Modus der Realtime-Fähigkeit laufen zu lassen sind jedoch zwei Bedingungen nötig. Zuerst muss der Start des Programms mit erhöhten Zugriffsrechten, also root-Rechten im Linux System gestartet werden. Daraufaufgehend wird eine Einstellung des sogenannten Scheduling Parameter erfordert. Diese Parameter geben an, welcher Echtzeitprozess zu einem Zeitpunkt die gewollte CPU-Zeit erhält. Als eingängiges Beispiel ist ein Audiotreiber, der wie auch ein Videotreiber in Echtzeit laufen muss. Dort ist es meist so, dass der Audiochip mit einer Frequenz von 96 kHz getaktet werden muss, wohingegen der Videotreiber mit 60 Hz den Bildschirm aktualisiert wird. Würde der Videotreiber vor dem Audiotreiber die CPU-Zeit zugewiesen bekommen, so würde die Audioausgabe stocken. Somit braucht es einen Echtzeit-Scheduler, der die CPU-Zeit auf den richtigen Prozess verteilt. In diesem Beispiel wird diese Affinität zur Echtzeit durch den `pthread_setschedparam()` übergeben. Hierbei wird die Affinität für diesen Prozess auf 99, also das höchstmögliche eingestellt, womit sichergestellt werden soll, dass so viel CPU-Zeit wie möglich dem Prozess beibehalten wird. Als Argument lässt sich noch die Art der Warteschleife einstellen, in der die Echtzeit-Prozesse abgearbeitet werden. Hierbei wird das Round-Robin-Verfahren gewählt, da somit wirklich die meiste Zeit an den programmierten Task übergeben wird.

Speicherverwaltung und Prozessoreinstellung

Die nächste Optimierung, die zu treffen ist, damit die Abtastung wirklich gut funktioniert, ist es, als aller erstes das Swapping zu deaktivieren. Swapping beschreibt den Fall, dass Teile eines Programms im RAM durch das Betriebssystem auf einen Bereich der SD-Karte geschrieben wird, um Platz im Arbeitsspeicher frei zu geben. Dies lässt sich zwar auch systemweit einstellen, jedoch ist es besser, um keine Abhängigkeiten zu zerstören, das Swapping ausschließlich dieses Programms auszuschalten. Mit dem Deaktivieren lassen sich teilweise Millisekunden Verzögerung verhindern. Mittels des Linux-Systemcall im Userspace `mlockall(MCL_CURRENT | MCL_FUTURE)` im C-Programm kann dafür gesorgt werden, dass dieses Programm, in dem das ausgeführt wird, rein im Arbeitsspeicher verweilt und keine Speicheroptimierung durchgeführt wird.

Da ein Betriebssystem meist die verfügbaren Prozessorkerne für alle verschiedenen anstehenden Programme aufteilt, sollte sichergestellt werden, dass ein Kern nicht vom Be-

triebssystem für das Scheduling, also das Aufteilen von allen laufenden Threads benutzt werden soll. Hierbei wird der letzte Kern, der vierte Kern des Raspberry Pis, durch die Direktive *isolcpus=3* in der Datei */boot/cmdline.txt* beim Starten des Kernels als nicht zu benutzen markiert und es werden keine Threads auf diesen Kern verteilt. Eine nachfolgende Überprüfung durch einen Task-Manager wie *htop* stellt eindeutig klar, dass auf diesem Kern als einziger eine ständige CPU-Last von 0,0% anliegt. Somit ist der CPU-Kern als isoliert zu betrachten.

Durch die Anweisung *CPU_SET()* im C-Code kann nun der freie Kern ausgewählt werden. Dadurch wird nun dieser Kern allein von dem Programm ausgenutzt und sollte von dem Betriebssystem nicht mehr benutzen werden. Somit sind dem Programmierer alle Freiheiten gegeben, einen gesamten CPU-Kern vollkommen auszunutzen.

Um die allgemeine Performance des Systems zu steigern lässt sich der SoC übertakten. Hierbei ist aber zu beachten, dass dieser Takt die SPI-Clock des Systems mit beeinträchtigt. Somit wird bei einer Erhöhung des CPU-Taktes gleichzeitig der SPI-Takt erhöht, da dieser lediglich mit einem Teiler heruntergetaktet wird. Also ist darauf zu achten, dass bei der Programmierung diese Erhöhung berücksichtigt wird, sodass der Teiler des SPI-Taktes im Rahmen des maximalen Taktes des ADC-Chips gewählt wird.

Programmstruktur

Das Serial Peripheral Interface-Protokoll (SPI-Protokoll) sieht vor, dass, um von dem SPI-Slave Daten zu erhalten, müssen zuerst die gleiche Anzahl an Daten an den SPI-Slave übertragen werden. Das bedeutet, dass, um einen 16-bit Wert zu erhalten, müssen zwei 8-bit Daten von dem Master an den Slave übertragen werden, da 8-bit das kleinste Symbol des SPI-Protokoll darstellt. Wie schon in 4.1.4 angesprochen, wird die Umsetzung des Programms zur Abfrage rein durch die Bibliothek BCM2835 umgesetzt. Um das Programm zu optimieren und allgemein schon im Voraus eine echte funktionierende Codebasis zu haben, ist es nötig Threading zu implementieren. Dieses Threading sorgt dafür, dass zwei Threads in dem Kontext des Programms laufen und somit kann ein Thread rein für das Polling der Spannungswerte genutzt werden, wohingegen der andere Thread dafür sorgt, die Werte zu extrahieren und durch den Algorithmus zur Frequenzmessung und zur Oberschwingungsmessung zu verarbeiten. Es werden zwei Threads erstellt, einmal einen Polling-Thread und einmal einen Evaluation-Thread. Der Polling Thread hat die Aufgabe der ständigen Aufnahme von Datenpunkten durch den ADC und der Evaluation-Thread nutzt dann diese Daten um durch einen Algorithmus die

Netzspannung auszuwerten. Dies benötigt zwei relativ große Speicherbereiche, die beide mindestens so groß wie die Datenmenge, die für eine Aufnahmeperiode, also eine Sekunde, anfällt. Da die Daten als Typ der Fließkomma am Ende der Auswertungskette benötigt werden, und deren Variable mindestens 32-bit weit ist, sind dafür also $1 \text{ MS/s} \cdot 32\text{-bit} = 32 \text{ Mbit/s} = 4 \text{ MB/s}$, zwei 4 MB große Arrays nötig. Ohne diese beiden Arrays wäre ein sogenannter Threadlock immer möglich. Der Thread, der in Echtzeit die Werte vom ADC nehmen soll, sollte immer nahtlos die Daten des ADC speichern und muss deswegen den Zielthread ändern. Somit nimmt der Polling-Thread nach einer Sekunde den anderen, gerade nicht beschriebenen Array als Zielspeicher und lässt den vorhergehenden Array durch einen sogenannten Mutex frei. Mutual exclusion (Mutex) sind Verfahren, die eine Variable von einem Thread nehmen lassen und aber von anderen Threads gesperrt werden. Somit wird immer sichergestellt, dass der Polling-Thread niemals die Zugriffsrechte auf einen Array verliert. Der Evaluation-Thread wartet darauf, dass der Polling-Thread eine Sekunde Datenaufnahme abgeschlossen hat und den Array wechselt. Hierbei wartet er auf die Freigabe des jeweiligen Mutex des vollen Arrays, nimmt diese Mutex an sich und kann dementsprechend den Algorithmus über die Daten im Array laufen lassen. Am Ende des Auswertens des Arrays wird dieser Array durch die Methode *memset()* effizient gesamt auf 0 gesetzt. Falls dieser Evaluation-Thread mehr als eine Sekunde Zeit für das Verarbeiten des kompletten Arrays in Anspruch nimmt, würde ein Threadlock passieren. Hierbei gibt der Polling-Thread den Mutex für den gerade gefüllten Array ab und möchte sich den Mutex des anderen Arrays annehmen, was jedoch verweigert wird, da der Evaluation-Thread diese Mutex noch nicht freigegeben ist. Dadurch gerät das Programm ins Stocken und die Daten des ADCs können nicht mehr valide gespeichert werden. Da jedoch der Evaluation-Thread durch effiziente Programmierung den gerade von ihm behandelten Array in einigen Millisekunden wieder freigibt, stellt dies kein Problem dar. Somit kann nach Einstellen der SPI-Commands der Bibliothek und nachfolgendem Einstellen des Verstärkers im ADC, sodass eine Eingangsdynamik von $\pm 10 \text{ V}$ erreicht wird, die eigentliche Abarbeitung des Programms beginnen. Zu Beginn des Polling-Threads wird ein sogenanntes Dummy-Load ausgeführt. Dies stellt einfach eine Rechenaufgabe dar, bei der eine Variable hochgezählt wird. Das hat zur Folge, dass, falls durch unbekannte Umstände ein Energiesparmodus des SoCs aktiviert ist, der Chip durch die Rechenaufgabe in den normalen Taktzustand zurückversetzt wird. Dies geschieht einmalig vor Beginnen der eigentlichen Ausführung des Abfragen der Werte aus dem ADC. Danach befindet sich eine Schleife, die bis zum Beenden des Programms fortgeführt ist. Zu Beginn der Schleife wird der jeweilige Array ausgewählt, der gerade nicht in Benutzung ist, um diesen als Zielspeicher auszuwählen. Erkennt es dabei einen Threadlock bricht das Programm sofortig ab, da keine reibungslose Aufnahme der Daten-

punkte garantiert werden kann. Nun wird genau vor Aufnahme des ersten Datenpunkts der Zeitstempel mit Nanosekunden Genauigkeit abgefragt und darauf anschließend sofort die Datenpunkte aus dem ADC extrahiert, indem zwei leere Byte an ihn gesendet werden. Dieser gibt zwei Byte zurück, wovon das erste Byte in eine Zwischenvariable kopiert wird, dann wird dieses um 8-bit nach links geschiftet, das Vorzeichen richtig gesetzt und schließlich das andere Byte in den freien Platz der Zwischenvariable kopiert. Somit ist dann der echte Spannungswert diskretisiert in 2^{15} Schritten binär kodiert. Dies muss noch mit einem Vorfaktor, der sich aus den gesamten Spannungsteilern und Verstärkern der Strecke ergibt und als echte Dezimalzahl mit Komma an den richtigen Platz des Arrays abgespeichert werden. Dieser Schritt wird so lange wiederholt bis exakt eine Sekunde vergangen ist, dann wird der der Mutex des Arrays freigegeben, der Array gewechselt und der neue mit den Daten gefüllt. In der Zwischenzeit benutzt der zweite Thread den gefüllten Array, durchläuft zuerst den Algorithmus der Frequenzmessung, wobei aber hier der Barycentric Rational Interpolation-Algorithmus der **Boost** C++ Bibliothek für die Interpolation und somit die Erkennung des genauen Nullpunktes benutzt wird. Die Berechnung der Fast-Fourier-Transformation wäre durch die Methode `fftw_plan_r2r_1d` der Bibliothek FFTW durchführbar, dies wird jedoch nicht implementiert, da das Störverhalten des Betriebssystems Linux bei der Frequenzmessung absehbar ist.

Störverhalten durch das Betriebssystem

Das Linux Betriebssystem sorgt für eine wirkungsverändernde Problematik, die nicht wieder rückgängig gemacht werden kann. Dadurch, dass der Polling-Thread 100% der CPU-Zeit des eigentlich freien Kerns in Anspruch nimmt, kommt das Betriebssystem in den Glauben, dass dieser Thread nicht mehr reagiert, das heißt, dass dieser einem Absturz zu Grunde geht. Selbst mit den höchsten Rechten sorgt es dafür, dass die Arbeitsspeicheroperationen dadurch nicht mehr ordentlich vollbracht werden, Speicher wieder freigegeben wird und somit entsteht ein sogenanntes Speicherleck. Durch eine Analyse des Programms **Valgrind** fallen keine vom Nutzer provozierten Speicherlecks auf. Diese Speicherlecks treten dadurch auf, dass keine CPU-Zeit dem Betriebssystem bereitgestellt wird, um die, von dem Programm als frei markierten Speicherbereiche freizugeben. Dies resultiert in einem Absturz nach einer gewissen Zeit, wenn zu viel Arbeitsspeicher eingenommen, aber nicht wieder frei gestellt wurde. Die Dauer dieses Prozesses liegt zwischen 20 Minuten und einer halben Stunde, was variabel ist, da andere Systemprozesse auch teils mehr oder weniger Arbeitsspeicher einnehmen. Der Linux-Kernel schließt in diesem Fall, wenn kein Arbeitsspeicher mehr frei ist den Prozess und die Datenaus-

wertung schlägt fehl. Bei Implementierung einer Pausenzeit, die zwischen den einzelnen Aufnahmen eine kleine Unterbrechung der Aufnahme der Spannungswerte einfügt, lassen sich die Problematiken der Speicherlecks und Abstürze vermeiden, jedoch kostet dies die Garantie der exakt äquivalent verteilten Stützstellen. Selbst ohne die Pausenzeit ist es jedoch nicht garantiert, dass die Stützstellen äquivalent verteilt sind. Dies ist selbst in der Echtzeit-Umgebung des Linux-Betriebssystems nicht vorgesehen. Die Zwischenzeiten von einer Mikrosekunde sind nicht in einer Auflösung der Echtzeit einhaltbar. Hierbei ist auf dieser Hardware mit einer Ungenauigkeit von ca. 20 μ s zu rechnen. Die Hardware-Timer Auflösung des Pi ist dafür zu gering, um diesen als Interrupt-Quelle für das Abfragen eines Wertes zu nehmen. Eine Funktion, die ein x86-basiertes System seit einiger Zeit bereitstellt, ist ein High-Precision-Event-Timer, der sehr genaue und präzise Events bereitstellt, die dann zu einer Abfrage eines Spannungswertes führen. Dies wäre dann eine Art der Interrupt-basierten Abfrage, anstatt wie in diesem Programm Polling-basiert [5]. Durch die nicht gleichverteilten Stützstellen der Spannungsabfrage kommen Ungenauigkeiten bei der Auswertung der Netzfrequenz hinein. Auch wenn der allgemeine Kurvenverlauf dem Auge der realen Netzfrequenz gleichkommt, lässt sich so selbst mit Interpolation keine richtige Aussage über die Netzfrequenz schließen, da für die Berechnung die Abtastdauer zwischen den beiden Nulldurchgängen exakt gleich sein muss. Dies ist hierbei jedoch nicht gegeben und es kommt so zu einer Auswertung, die bei einer Frequenz von 50 Hz einen Wert zwischen 47 Hz und 53 Hz ausgibt. Die Spannungswerte an sich zeigen keine negativen Muster auf, was darauf schließen lässt, dass der Algorithmus und die Kette zur Übertragung der binären Datenwerte des Analog/Digital-Umsetzers in Ordnung ist und die richtigen Werte mit der richtigen Skalierung an den Computer übertragen werden. In Abbildung 4.1 lässt sich die Netzspannung gut erkennen.

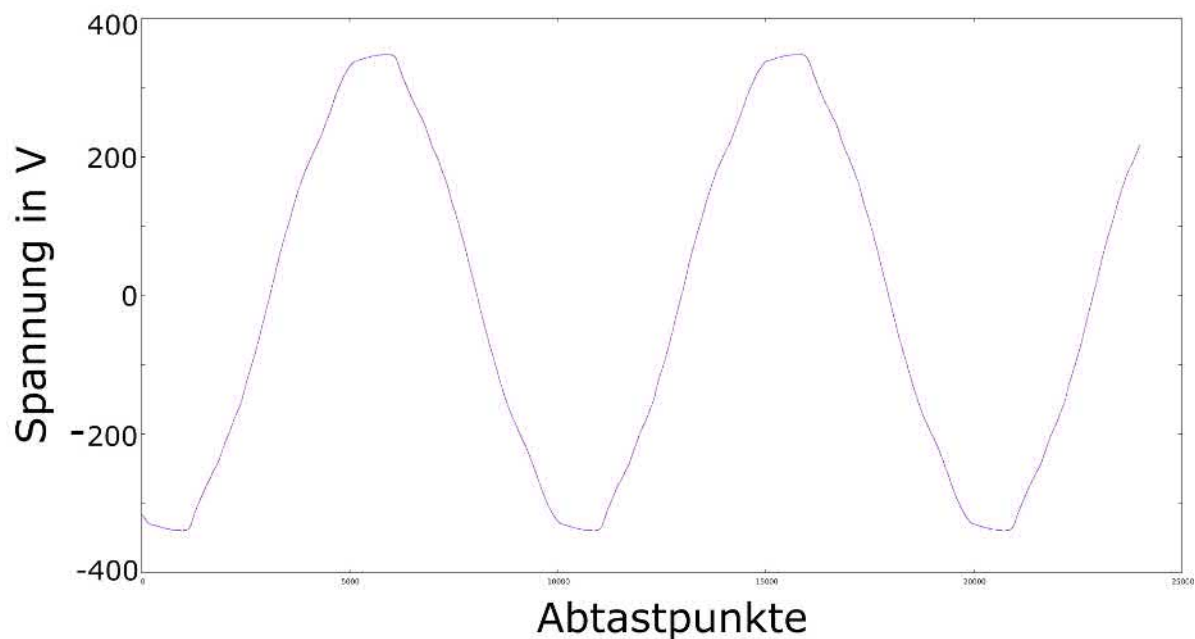


Abbildung 4.1: Datenausgabe der Messung von zwei Perioden der Netzfrequenz durch den ADS8681

Ein Mikrocontroller, der die Datenwerte mit einer Hardware-Uhr nach einer Taktflanke abfragt wäre hierbei eine deutlich bessere Option und sorgt dafür, dass die zeitkritischen Bauteile des Programms nicht per Software betrieben werden müssen. Hierbei kann ein Mikrocontroller diese Daten bis zu einer gewissen Zeit zwischenspeichern und gibt danach diese Werte in einem Block an den Computer ab. Dies ist dem Prinzip eines Datenloggers ähnlich, der Daten, die hier Spannungswerte darstellen, durch einen Prozessor abfragt und danach ohne Veränderung der Werte in einem bestimmten Rhythmus in einem Block an einen Computer oder einem Speichermedium übergibt. Somit wäre die Problematik des zeitkritischen Programms überkommen und es muss schließlich nur noch der Algorithmus zur Frequenz- und Amplitudenmessung der bereitgestellten Daten des Datenloggers ausgeführt werden.

Benutzen des DataTranslation DT9816-S Dataloggers mit MATLAB Integration

Dadurch, dass sich der Raspberry Pi aufgrund seiner Nicht-Echtzeitfähigkeit nicht für den Messaufbau eignet, ist die digitale Auswertung durch den ADS8681-Chip nicht möglich, jedoch ist der gesamte vorherige Aufbau, somit ist die Isolationsplatine als auch die Tiefpass-Sektion mit Spannungsversorgung immer noch relevant und für die Funktion nötig. Die Aufzeichnung der Spannungswerte wird jedoch nun mit einem Datenlogger DT9816-S des Unternehmens DataTranslation vollzogen. Der DataTranslation USB-

Datenlogger hat eine maximale Abtastrate von insgesamt 750 kHz, wenn jedoch nur ein von den sechs verfügbaren Kanälen genutzt wird, ansonsten wird diese Abtastfrequenz zwischen den Kanälen geteilt. Trotz der Benutzung von nur einem Kanal weist der Datenlogger eine Problematik auf. Wird die Abtastfrequenz über den Wert von 150 kHz eingestellt, finden Artefakte innerhalb des Messsignals statt. Dies hat bei kleineren Abtastraten unter 300 kHz ein Ringen und Aufschwingen bei der maximalen Auslenkung des Signals zur Folge, was aber bei einer Abtastrate von 750 kHz, wie in Abbildung 4.2 in einem Zerhacken des Signals endet. Somit wird die Abtastrate auf die übliche Rate von 50 kHz, wie in Abbildung 4.3, eingestellt, da dort die Artefakte minimal bzw. nicht vorhanden sind.

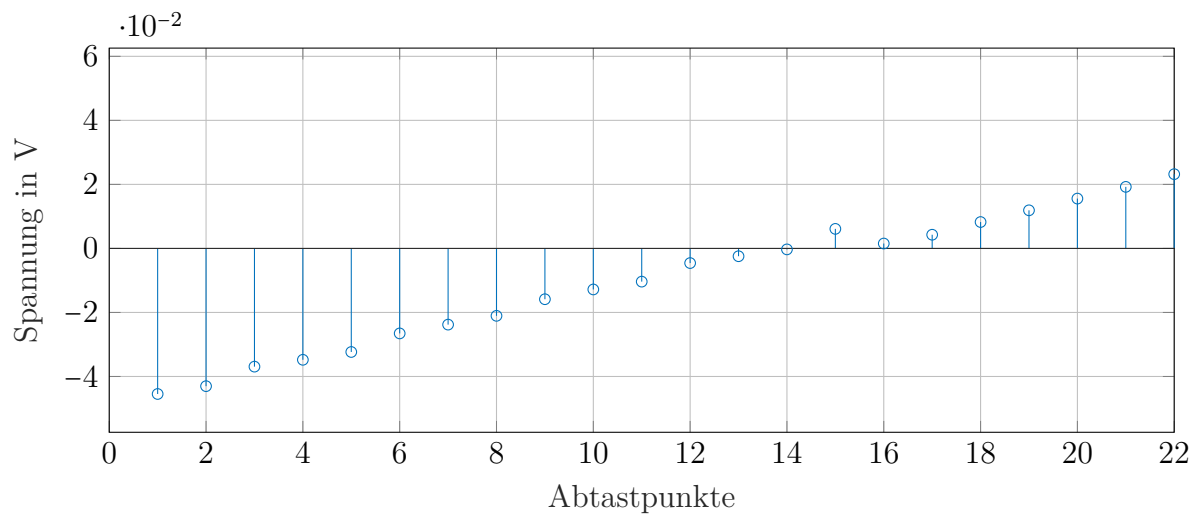


Abbildung 4.2: Nulldurchgang des Messinstruments DT9816-S bei 750 kHz Abtastrate

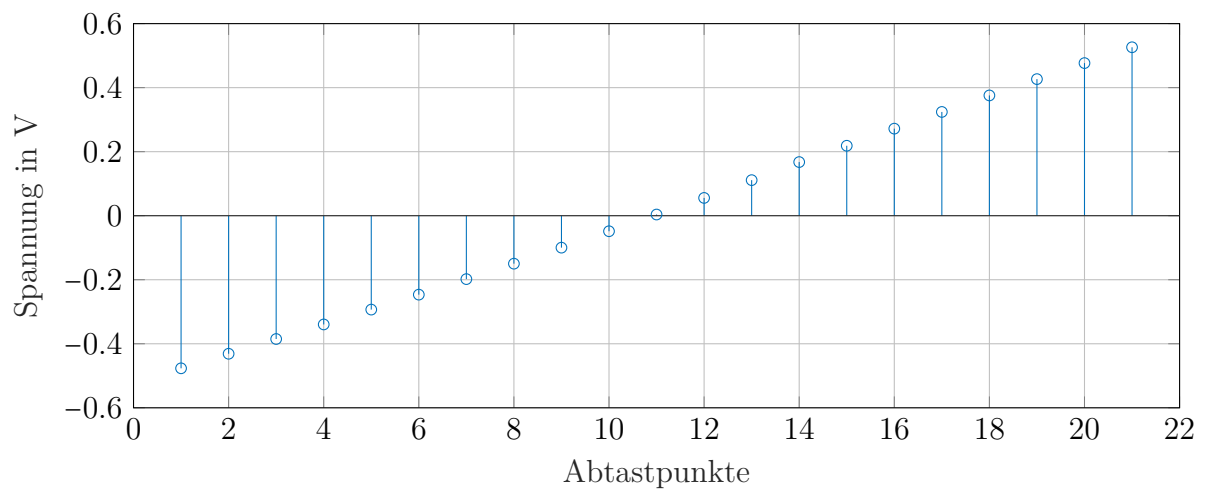


Abbildung 4.3: Nulldurchgang des Messinstruments DT9816-S bei 50 kHz Abtastrate

Aufbau und Programmierung Das USB-Oszilloskop DataTranslation DT9816-S wird per USB-Schnittstelle mit einem Windows-Computer verbunden. Hierbei sind verschiedene Software-Schnittstellen zur Akquise der Spannungswerte verfügbar. Jedoch wird in dieser Arbeit die MATLAB-Schnittstelle benutzt und die Auswertung ist in dieser Sprache geschrieben. Dadurch, dass simple Rechenoperationen wie FFT, Nullstellendetektion keine Echtzeitproblematik haben, ist die Rechenkapazität des Computers wenig ausschlaggebend. MATLAB bietet jedoch bei der Benutzung einige Funktionen, die die Programmierung einfach halten. Durch eine MATLAB-Executable Datei wird später das gesamte Programmpaket fertig gestellt und kann so genutzt werden. Dies ist jedoch nur auf einem Windows-Rechner mit installiertem Treiberpaket des Herstellers DataTranslation möglich. Wie vorher erwähnt wird die Abtastfrequenz auf 50 kHz reduziert, somit kann bei 16-bit Tiefe immer noch eine gute Amplitudendetektion stattfinden.

Implementiert wird der Algorithmus zur Frequenzmessung in einer anderen Weise wie in der Sprache C. Wie vorher wird der Eingangsvektor an der Netzspannung über eine Sekunde abgegriffen und wird hierfür benutzt. Durch den Datenlogger werden in MATLAB insgesamt 50000 Datenpunkte pro Sekunde übergeben, die jeweils dem 64-bit-Double Datentyp entsprechen. Diese müssen aufgrund der Eingangsspannung von ± 10 V durch das Isolationsboard auf ± 325 V skaliert werden. Nach der Skalierung können durch die Nullstellendetektion Vektoren aus der Spannungskurve extrahiert werden. Durch das Extrahieren kann der Interpolationsalgorithmus *interp1* mit der Variante *spline* benutzt werden. Somit wird der exakte Schnittpunkt des Nullpunkts mit positiver Flanke an der linken und an der rechten Stelle des Aufnahmeintervalls ermittelt werden. Die Spline-Interpolation erfordert vier Punkte zur Interpolation der Nullstelle, wobei die zwei linken und zwei rechten Abtastpunkte um den Nulldurchgang dafür verwendet werden. Durch die Gleichung unter 4.1 kann die Frequenz so bestimmt werden.

Danach wird durch eine FFT die Möglichkeit der Amplitudenmessung gegeben. Diese muss jedoch aufgrund der Bedingungen, die die Sprache MATLAB mitbringt, skaliert und verändert werden. Die Skalierung muss durch den Faktor der Länge des Eingabevektors herunterskaliert werden und danach durch die Funktion *fftshift* mit der richtigen Reihenfolge, als einseitiges Spektrum, zur Abfrage vorbereitet werden. Danach kann die Oberschwingung durch einfache Auswertung der Datenpunkte an den gewollten Frequenzpunkten durchgeführt werden. Eine Speicherung ist sowohl über eine Datenbank nach außen implementiert, sowie die Speicherung aller Messdaten über die Festplatte vollzogen. Hierbei fallen ca. 25 MB Speicherbedarf pro Tag an, der sich aus 25 Amplituden, einem Frequenzwert und einem Zeitstempel pro Sekunde zusammensetzt.

Sollte die Datenbankanbindung nicht funktionieren, da keine Verbindung aufgebaut wer-

den kann, so werden die Daten, die nicht gespeichert werden konnten, zwischengespeichert und nach Wiederaufbau einer Verbindung werden die Daten mit deren korrekten Zeitstempeln hochgeladen.

4.2 Validieren der Messergebnisse

4.2.1 Prüfprotokoll zur Einhaltung der Anforderungen

Das Prüfprotokoll der Norm DIN EN 61010 für Mess- bzw. Laborgeräte wurde erfolgreich ausgeführt. In dieser Stückprüfung werden verschiedene Parameter durch ein Messgerät an dem Stück geprüft. Zur Überprüfung wird hier ein SECUTEST SIII+ H von der Firma Gossen-Metrawatt GmbH benutzt. Dies kann das Einhalten der entsprechenden Normen in einem Prüfprotokoll ausgeben.

Die gesamten Prüfprotokolle findet sich im Anhang unter D.

4.2.2 Validieren des analogen Tiefpasses

Bei Validierung des Tiefpassfilters durch den Frequenzgenerator stellt sich die Realität im Vergleich zur Simulation heraus. Dieses Fehlverhalten ist durch Bauelemente-Fehler zu ergründen. Auf die Eingänge der Operationsverstärkerstrecke wurde ein Frequenzgenerator von Keysight angelegt, der eine saubere Eingangsspannung erzeugt. Der Ausgang der Operationsverstärkerstrecke wird dann als Ausgangsspannung angesehen und eine Übertragungsfunktion von $\frac{U_{\text{aus}}(f)}{U_{\text{ein}}(f)} = A(f)$ gebildet. Diese Übertragungsfunktion wird dann normiert und in die Einheit Dezibel umgerechnet.

Jedoch auch wichtig zu messen ist der DC-Offset, der am Ausgang erscheint. Bei einer Ausgangsamplitude von 18 V Spitze-Spitze liegt ein DC-Wert von ca. 16 mV überlagert an, was einer sehr guten Toleranz der Bauelemente im Differenzverstärker zu erkennen gibt. Ein DC-Offset von genau 0 V ist meist nicht zu erreichen, da der Operationsverstärker intern einen Offsetfehler besitzt und die Widerstände des Differenzverstärkers nicht genau identisch sind.

Wie an Abbildung 4.4 zu sehen ist, ist die Grenzfrequenz, also der Punkt der Kennlinie, der die -3 dB Linie schneidet lediglich bei 2913 Hz. Dies bedeutet, dass die Toleranzen der Bauelemente so groß sind, dass diese nicht wie in der vorhergehenden Berechnung

der Werte für Bauelemente folgen. Vor allem sind hierbei die Kapazitäten ein Problem, da diese selbst in der höchsten Spezifikation einen sehr großen thermischen Koeffizient besitzen und die Kapazitätswerte durch diesen und der allgemeinen Toleranz stark abweichen kann. Durch Benutzen von Klasse-1-Kondensatoren ließe sich dieses Problem verhindern und eine Schaltung kreieren, die selbst in verschiedenen Bedingungen fehlerfrei und nach Toleranz funktioniert. Diese erste Stufe wurde durch die Toleranz so stark verändert, dass die Grenzfrequenz dieses Teilfilters so stark verändert wurde, dass dieser die benötigten Frequenzen mit abschwächt. Dies ist daran zu sehen, dass bei einer Frequenz von 1250 Hz, also der letzten gewollten Oberschwingung eine Abnahme von $1 - \frac{14,00V}{14,10V} = 0,709\%$ stattfindet. Dies liegt zwar innerhalb der gewollten Spezifikation, wirkt sich jedoch schlecht auf das Ergebnis aus. Die erste Tiefpass-Stufe muss demnach neu ausgelegt werden, damit sich die kaskadierte Grenzfrequenz des gesamten Filters wieder auf 3000 Hz verlagert.

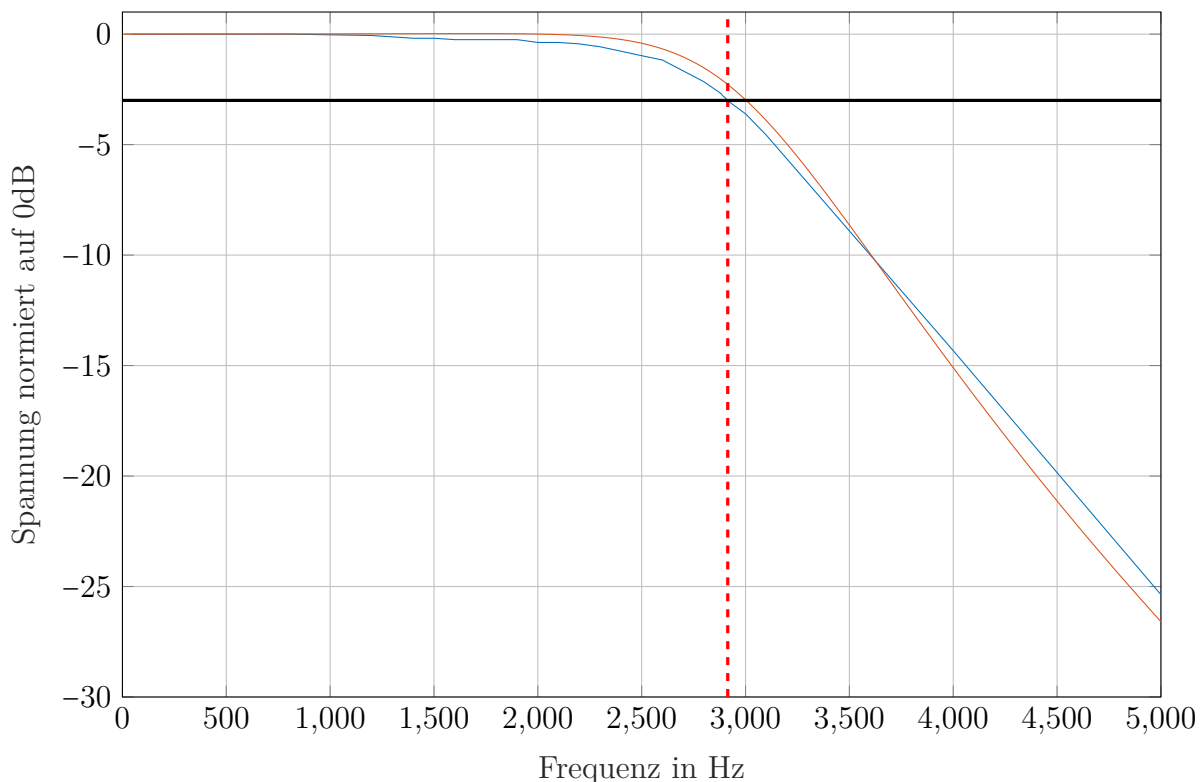


Abbildung 4.4: Übertragungsfunktion des Tiefpassfilters
 Hellblau: Gemessene Übertragungsfunktion
 Orange: Simulierte Übertragungsfunktion
 Schwarze Linie: 3dB-Dämpfungslinie
 Rote Linie: Schnittpunkt der Übertragungsfunktion mit der -3dB-Linie

4.2.3 Überprüfen des Rauschens am Eingang des Analog/Digital-Umsetzers

In Abbildung 4.5 wird eine Messung der Rauschspannung am Eingang des Analog/Digital-Umsetzers dargestellt. Dies stellt nach dem Schaltbild auch den Ausgang der Filterstrecke dar. Das Rauschen am Umsetzer hat bei Betrachtung von 10 Sekunden eine maximale Spitze-Spitze-Spannung von 18 mV, wobei jedoch die allgemeine AC-RMS-Rauschspannung wichtiger ist. Die 18 mV beschreiben das Maximum der Spannung, der RMS-Wert die kontinuierliche Spannung, die das Ergebnis verfälscht. Diese AC-RMS-Spannung ist durch die Messung des Keysight Oszilloskops auf 1,13 mV bestimmt worden. Dies suggeriert einen sehr kleinen Wert, der, wenn der A/D-Umsetzer die Werte diskretisiert, auch konstant bleibt. Somit wird keine Störung bei der Aufnahme der Spannungswerte in die Messung eingebracht. Um den DC-Offset getrennt zu messen, wird ein Hochpass des Oszilloskops zugeschaltet, womit die Nulllage des Signals in den Mittelpunkt versetzt wird. Dazu wird die Bandbreite des Oszilloskops durch die Funktion BW-Limit auf 20 MHz limitiert, obwohl selbst dieser Bereich weit außerhalb des messbaren Bereichs des ADCs liegt.

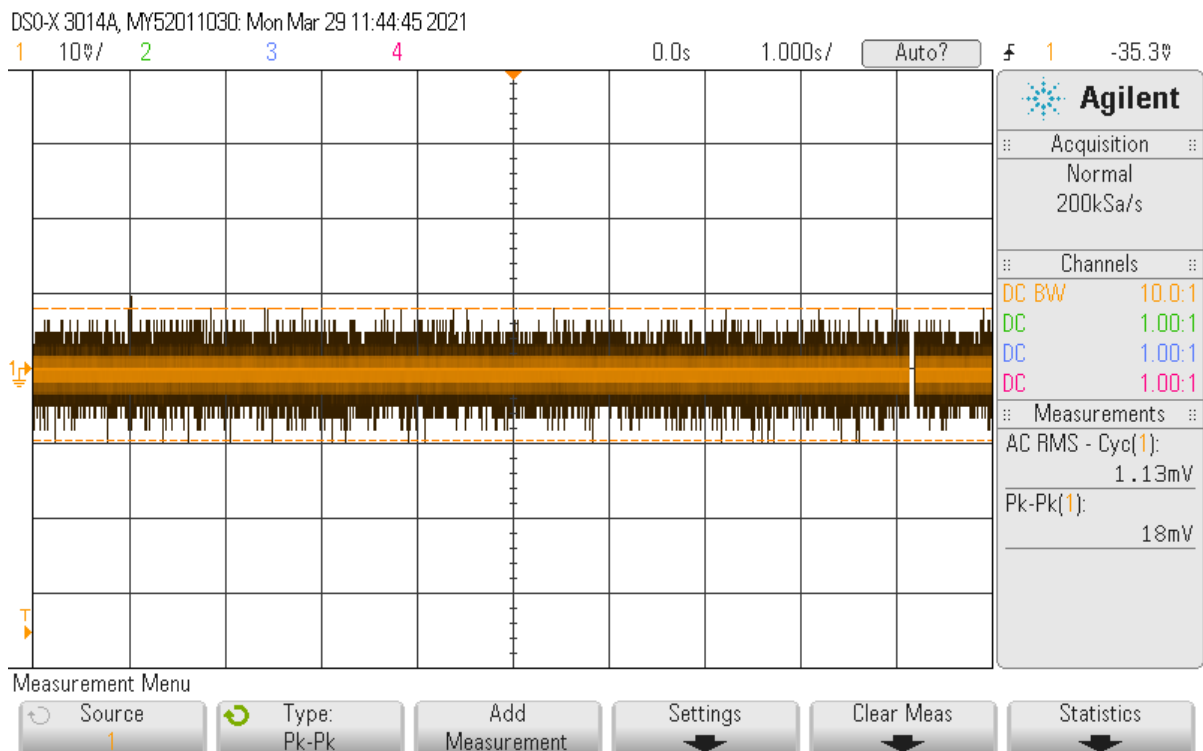


Abbildung 4.5: Rauschspannung am Eingang des Analog/Digital-Umsetzers

4.2.4 Genauigkeit des Zeitstempels

Durch das Einbauen der fünf meter langen Antennenverlängerung fand eine Dämpfung des GPS-Signals um 1 dB statt. Trotz dieser Verschlechterung waren im Schnitt 20 Satelliten mit der Antenne hinter Fensterglas für die Positionierung möglich. Der GPS-Chip erkennt eine Geschwindigkeit des Messortes von $0,1 \frac{\text{m}}{\text{s}}$ also $0,36 \frac{\text{km}}{\text{h}}$. Bei einer Betrachtung von einer Stunde existiert ein Distanzfehler des Ortes von 30 Metern, den der GPS-Baustein aufzeichnet. Für die Positions- und Zeitermittlung existieren viele Satelliten, da GLONASS, GPS und Beidou gleichzeitig benutzt werden. Dies kann von Uhrzeit und Standort abweichen. Der U-Blox Chip sorgt dafür, dass immer die meisten GNSS-Satelliten für die Messung in Betracht kommen. Um die Genauigkeit des Zeitstempels zu ermitteln, lässt sich die Möglichkeit des NTP-Services benutzen. Dieser kann durch den Befehl *ntpq -p* ausgelesen werden. In Abbildung 4.6 wird diese Ausgabe dargestellt. Dieser Dienst gibt aktuell aus, welche Zeitquellen der NTP-Client für seine Zeitbestimmung nutzt und gibt deren Genauigkeitswerte aus.

remote	refid	st	t	when	poll	reach	delay	offset	jitter
=====									
*SHM(0)	.SHM.	0	1	1	16	377	0.000	0.001	0.001
oPPS(0)	.PPS.	0	1	1	16	377	0.000	0.001	0.001
+ntp0.rrze.uni-e	.GPS.	1	u	62	64	377	0.358	0.013	0.009

Abbildung 4.6: Ausgabe des Status des NTP-Services durch *ntpq -p*

Die Quellen sind zeilenweise geordnet und werden nach ihrem Interface gruppiert, wovon die ersten beiden Zeilen die Schnittstelle zum GPS-Chip NEO-M8M bedeuten. Die letzte Zeile beschreibt den NTP-Zeitserver ntp0 des Rechenzentrums. Hierbei bedeuten die Präfixe jeweils: Ein Kreuz (+) für einen Kandidat, der zur Kombination genutzt wird. Ein Stern (*) bedeutet, dass diese Quelle als aktuelle Zeitquelle genutzt wird. Ein Kreis (0) bedeutet, dass dieser Pulse-Per-Second-Geber als Synchronisierungsquelle genutzt wird.

Der Buchstabe x bedeutet, dass dieser als Falseticker markiert wird und falsche Daten liefert und dementsprechend aussortiert ist.

Somit wird ersichtlich, dass sowohl die GPS-Zeitquelle als auch die GPS-PPS-Quelle zur Synchronisierung der Systemzeit des Linux-Betriebssystems und somit zum Zeitstempel des Auswertungsprogramms genutzt wird. Dadurch, dass beide als Stratum-0, also als die höchste Qualität von Zeitquellen in der Konfiguration markiert sind, werden diese

bevorzugt benutzt. Je höher der Stratum-Wert liegt, desto schlechter ist die eingestellte Genauigkeit der Zeitquelle. Diese Auswertung erfolgt für die Parameter *jitter*, *offset* und *delay* in Millisekunden. Natürlicherweise zeigt der Dienst an, dass die beiden GPS-Quellen einen *delay*, also Latenz von 0 besitzen, da es bei dieser Übertragung keinen Verzögerung gibt. Dieser Parameter ist für die Zeitserver, die via Ethernet angebunden sind, von Relevanz, da dort die Zeitverzögerung der Datenübertragung gemeint ist. Dies beschreibt die Round-Trip-Time. Der Offset-Parameter gibt an, wie viele Millisekunden die Zeit der echten Zeit vorangeschritten ist. Dieser Wert beträgt bei dem GPS-Chip eine Mikrosekunde und bei dem Zeitserver 13 Mikrosekunden, was bei beiden Zeiten ein gutes Ergebnis darstellt. Der Jitter ist der Faktor, bei dem der GPS-Chip durch sein PPS-Protokoll seine Stärke zeigen kann, da dieser bei einer Mikrosekunde liegt, im Gegensatz zu den neun Mikrosekunden, die der Zeitserver liefert. Der Wert *reach* ist eine Oktalzahl, die, falls die letzten acht Abfragen des Clients von der Zeitquelle beantwortet wurden, 377 anzeigt. Somit sind, wie jedoch zu erwarten war, alle Anfragen an die Zeitquellen korrekt. *Poll* bedeutet, in welchem Abstand in Sekunden die Werte zur Synchronisierung benutzt werden, also hier alle 16 Sekunden für den GPS-Chip und alle 64 Sekunden für den Zeitserver. *When* beschreibt, wann das letzte Paket den NTP-Clients gemessen in Sekunden erreicht hat. Die restlichen Parameter beschreiben lediglich Namen und id, sowie deren Stratum-Level und die Art der Übertragung (lokal, broadcast oder unicast).

4.2.5 Korrektheit der Amplitude der Oberschwingungen

Durch ein Prüfgerät CMC256 der Firma Omicron können unabhängig von der Grundschwingung verschiedene Oberschwingungen induziert werden. Die Spannung kann dabei im Niederspannungsbereich, also 230 V AC-RMS eingespeist werden. Diese noch reine Netzspannung wird als Spannungsquelle benutzt und dadurch lässt sich die reale Netzspannung sowie den gesamten Messaufbau, also Isolationsboard, HAT und DT9186 simulieren. Die Grundschwingung kann dabei in Frequenz und Amplitude modifiziert werden. Die Harmonischen werden entsprechend der Grundfrequenz variiert. Die Oberschwingungen können durch die Software individuell oder insgesamt induziert werden, in diesem Falle wurde nacheinander die jeweilige Oberschwingung getrennt eingeschleust, wodurch die Tabelle 4.1 entstanden ist. Diese Tabelle stellt den Unterschied dar, der durch die Messung gemessen wird zu dem was in Real in der Netzspannung durch das Prüfgerät eingeschleust wird. Die Amplituden A_{sol} , die den eingestellten Wert im Omicron Gerät darstellt, sowie A_{mess} , die aus der Messschaltung resultiert, sind in prozentualen Anga-

ben im Vergleich zur Grundschiwingung angegeben. Dies hat den Grund, da die in der Norm enthaltenen Grenzwerte in Prozent der Grundschiwingung angegeben sind. Es ist zwar möglich, dies sowohl in der Software des CMC, als auch in der Messung als Volt-Amplitude auszugeben, aber es wurde der prozentuale Wert aufgrund der Simplizität in der Auswertung präferiert.

f	100Hz	150Hz	200Hz	250Hz	300Hz	350Hz	400Hz	450Hz	500Hz
A_{soll}	10,0%	10,0%	10,0%	10,0%	10,0%	10,0%	10,0%	10,0%	10,0%
A_{mess}	9,99%	9,95%	9,95%	9,95%	9,92%	9,90%	9,87%	9,83%	9,81%

f	550Hz	600Hz	650Hz	700Hz	750Hz	800Hz	900Hz	950Hz	1000Hz
A_{soll}	10,0%	10,0%	10,0%	10,0%	10,0%	10,0%	10,0%	10,0%	10,0%
A_{mess}	9,78%	9,75%	9,72%	9,70%	9,67%	9,63%	9,6%	9,53%	9,51%

Tabelle 4.1: Amplituden der Harmonischen der Netzspannung

Es ergibt sich also ein maximaler Fehler von $1 - \frac{9,51}{10,0} = 4,9\%$ der bei der höchsten Harmonischen der Netzspannung entsteht. Die Gründe für die Fehler der Detektion sind unter anderem die unangepasste Grenzfrequenz des Tiefpasses, die dafür sorgt, dass die Oberschwingungen gedämpft werden und dadurch das Resultat fehlerbehaftet ist.

4.2.6 Auswertung der Netzfrequenz

Der Algorithmus der Netzfrequenzmessung basiert zwar rein auf dem physikalischen Grundsatz, dass die Schwingungsfrequenz durch die Periodendauer berechnet werden kann, jedoch muss diese ebenfalls aufgrund des Messaufbaus einer digitalen Abtastung validiert werden. Teilweise existieren keine genauen Teiler für die Abtastfrequenz und somit sind ungeradzahlige Abtastfrequenzen möglich und würden damit dann die gemessene Netzfrequenz mit einem Offset belasten. Dies lässt sich zum einen getrennt von dem Isolationsboard durch einen Funktionsgenerator realisieren, der eine höhere Ganggenauigkeit wie das Omicron CMC256 besitzt, und an dem Omicron Gerät in Verbindung des Trennelements.

Validierung mit dem Omicron CMC256

Durch das oben genannte OMICRON Prüfgerät lässt sich die Frequenz der Grundschiwingung von 230 V Effektivwert verändern. Diese Frequenz kann durch einen Sweep, also einer zeitlichen linearen Änderung, variieren. Somit ist eine Kennlinie des Messgeräts und

des Prüfgeräts entwickelbar. Ist diese Kennlinie linear, so ist die entwickelte Messschaltung und der Algorithmus zur Bestimmung der Netzfrequenz korrekt und kann damit validiert werden. Durch die Prüfung verschiedener eingestellter Frequenzpunkte kann ein zusätzlicher Schritt zum Erkennen eines Offset-Fehlers benutzt werden. Somit ist durch das Offset und der korrekten Steigung die Gerade definiert und bei einem Fehler kann es nur zu einem Ändern der Steigung kommen. Um den Aufbau zu prüfen, wird zu Beginn die Frequenz in ihrem gesamten Arbeitsbereich und weiter getestet. Hier wurde die Eingangsfrequenz von 44 Hz bis 56 Hz variiert. Da dies ein sehr großes Intervall ist, wird die Frequenz nicht im kleinsten Schritt, also ein Millihertz, sondern mit größeren Schritten, zehn mHz jede Sekunde, erhöht. Es lässt sich dadurch eine Eingangs-zu-Ausgangskennlinie der Frequenz entnehmen. Dadurch, da aber jede Sekunde die Frequenz erhöht wird, kann eine zeitliche Aufnahme ebenfalls genug Aussagekraft bringen, da die Steigung exakt mit derer, wenn die Ausgangsfrequenz des Messgeräts aufgezeichnet wird, übereinstimmt. In diesem Prüfaufbau wird ein Außenleiter in Verbindung des Neutralleiters an den C8-Stecker des Trennelements angebracht und die Frequenz der Spannung des Außenleiters variiert.

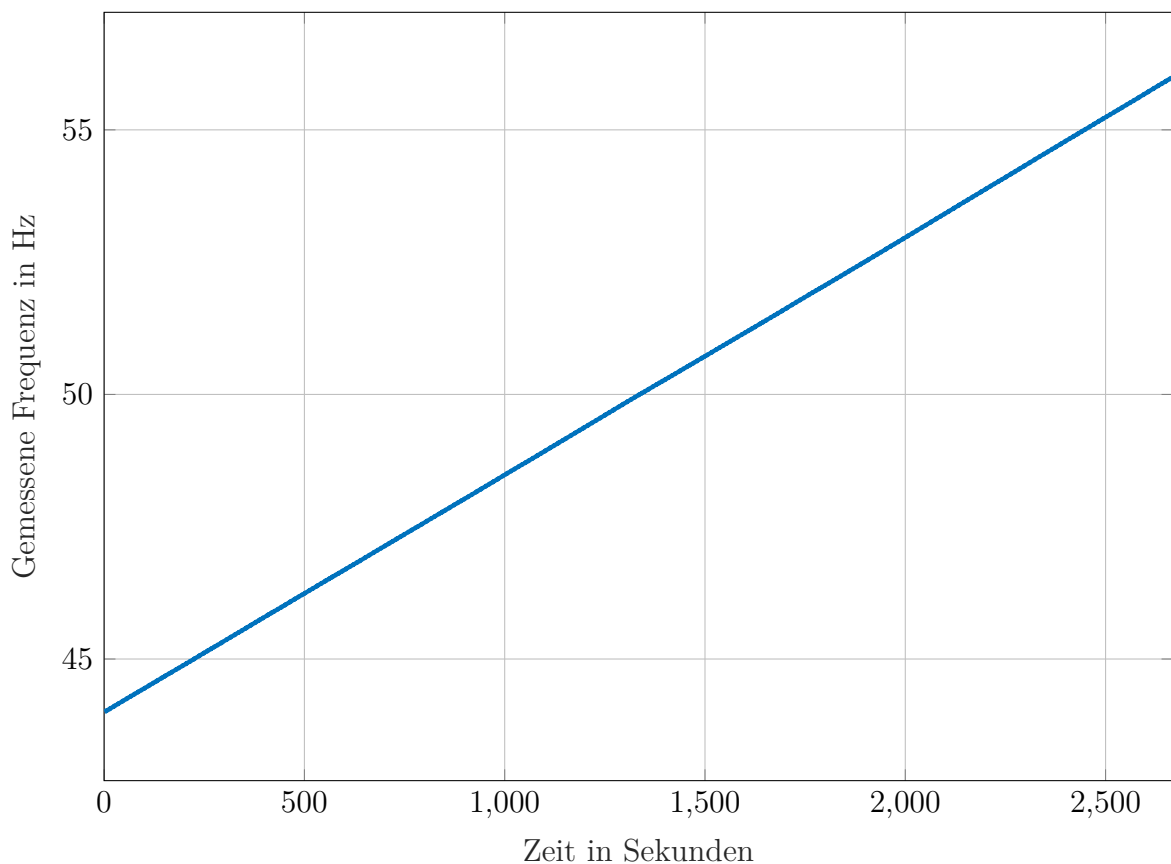


Abbildung 4.7: Frequenzauswertung bei einer steigenden Frequenz von 44 Hz bis 56 Hz, bei der jede Sekunde um 10 mHz gesteigert wird

Bei Betrachten des Verlaufs der Kurve unter Abbildung 4.7 lässt sich eine lineare Kennlinie erkennen. Durch diese lineare Kennlinie lässt sich ein systematischer Fehler der Messung ausschließen. Dadurch, da manuell verschiedene Punkte auf der Linie verifiziert wurden, lässt sich davon sprechen, dass die Messung im gesamten sehr gut die Frequenz linear aufnimmt. Um jedoch die feine Genauigkeit der Schaltung zu testen wird nun ein Frequenzsweep benutzt, der in der feinsten Genauigkeit pro Sekunde erhöht. Da dies im Arbeitspunkt der Messung gestartet wird, lässt sich eine Genauigkeitskennlinie vollziehen. Durch diese Kennlinie kann entdeckt werden, ob Schritte übersprungen werden und damit die Messung nicht exakt vollzogen wird. Da das Omicron CMC256 eine minimale Frequenzänderung von 1 mHz zulässt, wird nun alle drei Sekunden die Frequenz um 1 mHz erhöht, um der Messeinrichtung genug Zeit zu geben, die Detektion aus der letzten ganzen Sekunde zu nehmen. Hierbei sollten sich Treppenstufen zu erkennen geben, die 1 mHz-Schritte symbolisieren.

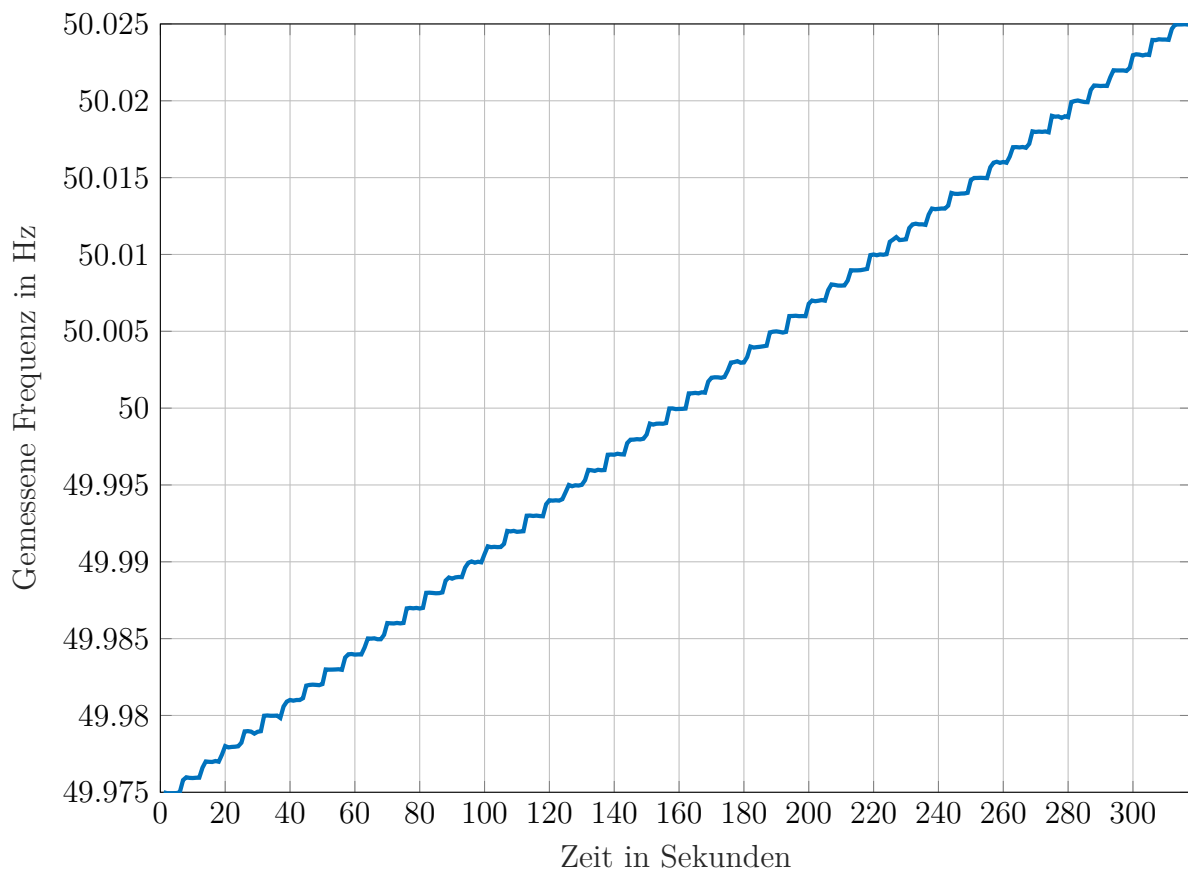


Abbildung 4.8: Frequenzauswertung bei einer steigenden Frequenz von 49,75 Hz bis 50,25 Hz, bei der alle drei Sekunden um 1 mHz gesteigert wird

An Abbildung 4.8 lassen sich diese Treppenstufen genau erkennen, und dank des Rasters des Graphen kann man davon stark ausgehen, dass sowohl das Omicron Gerät die Fre-

quenz genau erreicht und die Messung diese genau erkennt. Jeder Y-Achsen-Rasterpunkt beinhaltet einschließlich fünf Frequenzpunkte. Diese Abstufungen in jeweils ein Millihertz lassen sich genau erkennen, da nach jeder Steigung ein Plateau erreicht wird, die immer gleichen Abstand entsprechen. Somit kann davon ausgegangen werden, dass um den Arbeitspunkt 50,000 Hz eine genaue Messung der Frequenz möglich ist.

Validierung mit dem Keysight 33611A Funktionsgenerator

Da jedoch ein Funktionsgenerator eine deutlich höhere Ganggenauigkeit der Frequenz als das Omicron Gerät besitzt, lässt sich die analoge Filtereinrichtung und die Messung an sich getrennt untersuchen. Hier wird der Funktionsgenerator an die Eingänge der ersten Operationsverstärker-Stufe angeschlossen und die passende Amplitude von 1,6 V eingestellt. Es soll dadurch untersucht werden, wie stark die gemessene Frequenz sich mit der Zeit verändert. Es muss zu Beginn der Messung sowohl die Schaltung als auch der Funktionsgenerator auf Betriebstemperatur gebracht worden sein, damit davon ausgegangen werden kann, dass beide Geräte einer genaueren Aufnahme und Erzeugung von Spannung nachgehen können. Bei dieser Validierung wird eine 50 Hz-Frequenz auf die Eingänge der Messung gesetzt und der zeitliche Verlauf der gemessenen Netzfrequenz in einem Graphen ausgegeben. Ist diese Abweichung nach einer bestimmten Zeit zu hoch, so kann davon ausgegangen werden, dass die Messung auf langfristige Sicht keine richtigen Ergebnisse ausgibt. Die Abbildung 4.9 zeigt die Ausgabe der Frequenzmessung nach mehr als 3 Stunden Aufzeichnung der Spannung.

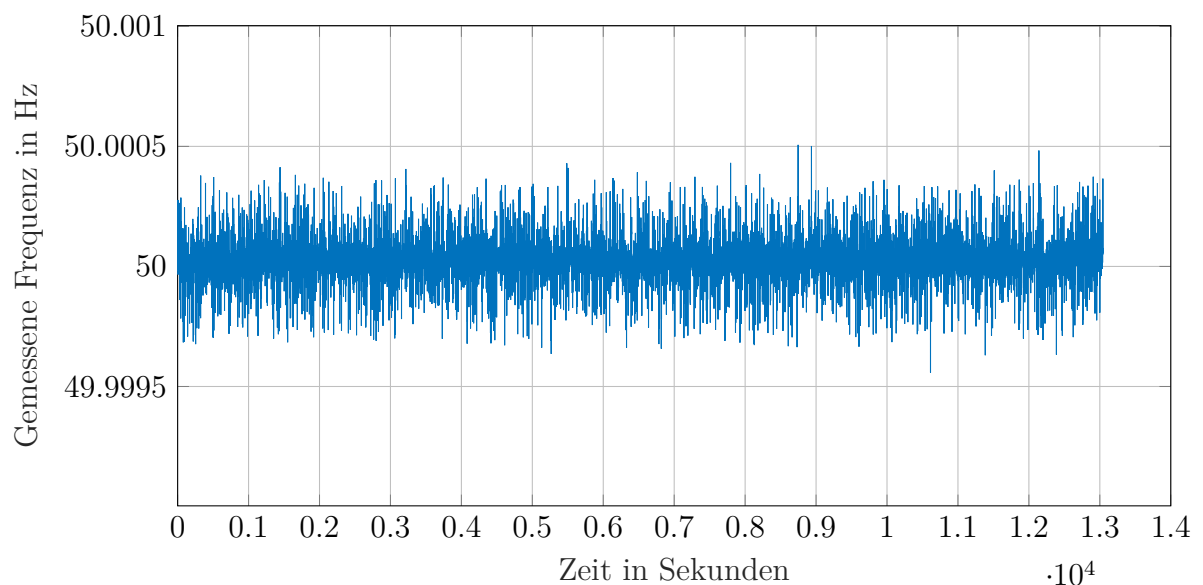


Abbildung 4.9: Aufzeichnung von Frequenzen bei benutzen eines Frequenzgenerators nach über drei Stunden Frequenzaufzeichnung

Als minimale Frequenz ist aus dem Graphen 49,9996 Hz und als maximale Frequenz 50,0005 Hz ersichtlich. Somit existiert ein geringer Messfehler in der Schaltung. Um damit einen sinnvolleren Eindruck der Messung an sich zu bekommen, lassen sich diese zeitlichen Punkte als Histogramm, also als Auftrittswahrscheinlichkeit gegenüber der jeweiligen Frequenz darstellen. Durch ein Histogramm lassen sich deutlich mehr Fehler in einer Messung erkennen.

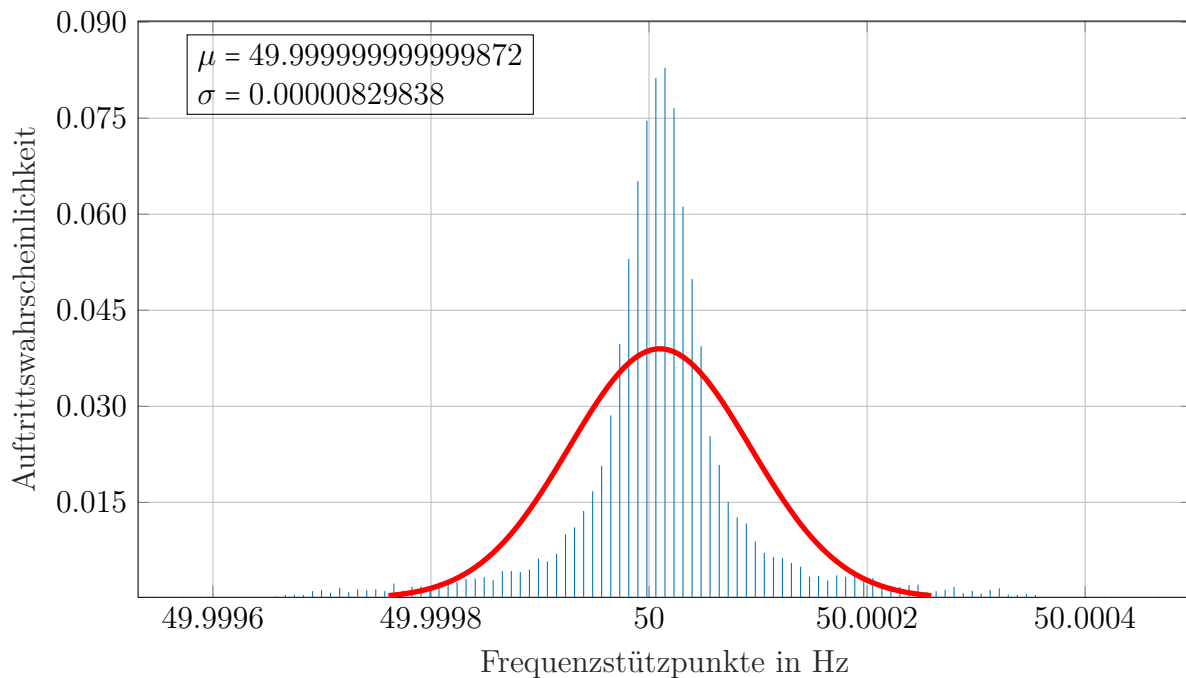


Abbildung 4.10: Auftrittswahrscheinlichkeiten von Frequenzen bei Benutzen eines Frequenzgenerators nach über drei Stunde Frequenzaufzeichnung
rote Linie: passende Gaußverteilung zu den Messdaten

Wie auf Abbildung 4.10 zu erkennen ist, der einer passenden Normalverteilung in roter Farbe hinterlegt wurde, lässt sich eine sehr geringfügige Verschiebung der Mittenfrequenz sehen. Dies bedeutet, dass im Durchschnitt die Frequenz richtig detektiert wird. Durch die Varianz, die die Funktion aufweist, also in diesem Fall 0,008 mHz, kann gesagt werden, dass durch Störungen die Frequenz schlechter detektiert wird. Jedoch ist diesem keine Verschiebung unterlegen, was auf einen systematischen Fehler der Messung deuten würde. Auch eine geringe Schiefe wird bei dieser Messung erzielt. Eine Schiefe wird dadurch erreicht werden, wenn die Verteilung nicht symmetrisch um einen Punkt wäre. Dies bedeutet, dass rein der zufällige Fehler die Messung beeinträchtigt, was durch bessere Bauelemente, höhere Abtastraten oder längeren Erfassungszeit der Messung verbessert werden kann.

4.2.7 Verbreitung der gemessenen Netzfrequenz durch das Internet

Eine Verbreitung der Netzfrequenz findet durch die Website <https://netz.miniwire.de> statt. Implementiert wurde dies durch eine Datenbankstruktur, welche die jeweiligen Werte der Frequenz zu dem genauen Zeitstempel aufbewahrt. Durch eine Kompression halten sich diese Datenmengen in Grenzen, da die Redundanz in den jeweiligen Datensätzen sehr hoch ist. Ohne Kompression beläuft sich eine Abspeicherung der Dateien auf ca. 25 MB pro Tag. Abgesendet wird dies durch eine Integration der Datenbank durch das MySQL Plugin von MathWorks in Matlab, nachdem die sekundliche Auswertung abgeschlossen ist. Übertragen wird der Zeitstempel nach der Auswertung, das bedeutet, dass die vorherige Sekunde an Daten dem aktuellen Zeitstempel entsprechen. Um die Daten an den Client, der die Webseite aufruft, weiterzugeben wird eine JavaScript Anbindung an einen PHP-Webserver gebraucht. Durch PHP, eine Skriptsprache für Webserver, wird sichergestellt, dass die Datenbankzugangsdaten nicht an die Clients weitergegeben wird, sondern nur die aktuellsten Einträge in der Datenbank herausgegeben werden. Durch JavaScript, dass der Browser des Endnutzers im Hintergrund ausführt, werden die anfänglichen statischen Werte sekundlich aktualisiert. Um Programmieraufwand zu reduzieren wird die kleine, quelloffene Bibliothek namens jQuery benutzt. Durch diese Verbindung aus PHP und JavaScript sind zwar die richtigen Daten im Arbeitsspeicher des Webbrowsers verfügbar, jedoch noch keine Darstellung dafür programmiert. Durch eine Cascaded Style Sheet, eine Beschreibungssprache für das Design und Aussehen einer Website, was auch wie PHP eine Empfehlung des World Wide Web Consortium, was als W3C auch bekannt ist, wird es möglich, die aktuelle Frequenz auf einem Strich darzustellen. Durch die Standards PHP, JavaScript und CSS wird auch leicht sichergestellt, dass ein Nutzer von seinem mobilen Endgerät das richtige Layout der Seite zu sehen bekommt. Hierzu wird dieser Rahmen farblich codiert, sodass die verschiedenen Zonen der Netzfrequenz leichter ersichtlich sind. In der Mitte wird das Optimum, also 50,000 Hz ausgewiesen, dieser Bereich wird mit der Farbe Grün markiert. Ab einer Abweichung von den 50 Hz von über 10 mHz wird die Farbe von grün zu gelb gewechselt, um den Totbereich der Primärregelung darzustellen. Ab diesen 10 mHz wird die Skala bis 200 mHz rot verlaufend, da dieser Zustand schon eine starke Unregelmäßigkeit des Netzes offenlegt, welcher nur in den seltensten Fällen erreicht wird. Zuletzt wird über dem aktuellen Zeiger der Frequenz die aktuelle Netzfrequenz angezeigt. Dieser Zeiger wird sekundlich aktualisiert und wie in Abbildung 4.11 im Browser angezeigt.

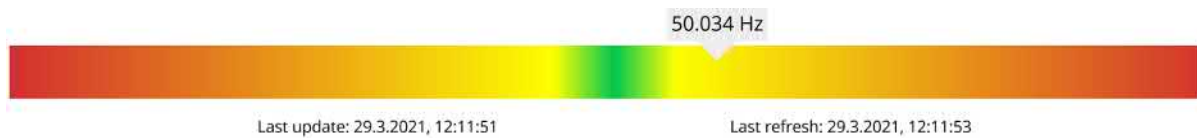


Abbildung 4.11: Skala der Netzfrequenzmessung mit einer Frequenz von 50,034 Hz



Abbildung 4.12: Visualisierung des Verlaufs der Netzfrequenz über 6 Stunden des Morgens des 29. März

Durch ein Visualisierungstool namens Grafana, unter Abbildung 4.12 sichtbar, ist es möglich, die Netzfrequenz in einer bestimmten Zeitdarstellung anzuzeigen, sodass ein Verlauf der Netzfrequenz über beispielsweise einen Tag, mehrere Stunden oder eine Woche ersichtlich wird, um besser sehen zu können, wie sich die Netzfrequenz zeitlich entwickelt. Dies wird in einer Diagrammdarstellung mit einer Abszisse der Zeit und einer Ordinate mit der Frequenz umgesetzt. Auch Histogramme und Heatmaps, die den Verlauf des Durchschnitts der Netzfrequenz in einer bestimmten Zeit darstellen sind dadurch möglich.

Diese Website wird nach Abschluss dieser Arbeit auf die Lehrstuhlseite zu finden sein. Die Datenbank als auch die Implementation in die Seite lässt sich durch das Regionale Rechenzentrum Erlangen der Friedrich-Alexander-Universität leicht umsetzen. Durch die standardisierten Programmiermethoden ist es schnell möglich, auf ein neues System zu portieren.

5 Zusammenfassung

In dieser Arbeit wurden zwei Platinen entwickelt, mit denen es möglich ist, die Spannung des Stromnetzes durch das Einstecken an eine Steckdose mit Niederspannung einphasig zu messen. Das Netzsignal wird zuerst potentialfrei getrennt, sodass keine Gefahr für Leib und Leben besteht, danach analog durch eine Kaskade an Operationsverstärkerschaltungen gefiltert und schließlich die Netzspannung als Signal digital umgesetzt. Die Platine zur Skalierung und potentialfreien Trennung der Netzspannung ist mit einem Gehäuse als Berührungsschutz versehen. Die andere Platine wurde zum Aufstecken an einen *Raspberry Pi 4* dimensioniert. Durch eine digitale Auswertung kann sowohl die aktuelle Netzfrequenz, als auch die Amplitude der aktuellen Netzspannung sowie deren Oberschwingungen von der zweiten bis zur 25. Harmonischen gemessen werden. Die Messung der Frequenz hat aufgrund von Störungen eine Genauigkeit von einem Millihertz, die Oberschwingungen werden jedoch durch einen einstelligen Fehler aufgrund von fehlerbehafteten Bauelementen gemessen. Aufgrund des Störverhalten des Betriebssystems lässt sich durch eine Analog/Digital-Umsetzung, bei der durch Software die Abtastung angestoßen wird, keine korrekte Messung der Frequenzdaten ausführen. Somit wurde ein USB-Datenlogger für die Aufnahme des Netzsignals an einem Windows-Computer eingesetzt. Diese gemessenen Daten werden danach über eine GPS-Zeit mit einem Zeitstempel versehen, der eine Genauigkeit von ungefähr einer Mikrosekunde aufweist. Diese Messdaten werden korrekt auf der Festplatte gespeichert und später in eine Datenbank übertragen, aus welcher die Daten für eine Online Visualisierung gespeist werden. Für zukünftige Entwicklungen wird empfohlen, dass man zur direkten Auswertung des Analog/Digital-Umsetzers einen Mikrocontroller zum Abrufen des SPI-Protokolls benutzen sollte, womit die zeitliche Verfälschung des Betriebssystems behoben und eine fehlerfreie Messung ermöglicht wird.

A C-Code des Raspberry Pi

```
1 // #define _GNU_SOURCE
2 #define ARMA_64BIT_WORD 1
3 #include <bcm2835.h>
4 #include <armadillo>
5 #include <stdio.h>
6 #include <unistd.h>
7 #include <sys/time.h>
8 #include <sys/resource.h>
9 #include <sched.h>
10 #include <sys/mman.h>
11 #include <vector>
12 #include <pthread.h>
13 #include <limits.h>
14 #include <string.h>
15 #include <errno.h>
16 #include </usr/include/boost/math/interpolators/cubic_b_spline.hpp>
17 #include </usr/include/boost/math/interpolators/barycentric_rational.hpp>
18 #include <boost/range/adaptors.hpp>
19 #include <boost/math/tools/roots.hpp>
20 extern "C"
21 {
22     #include "gnuplot_i/src/gnuplot_i.h"
23 }
24 using namespace arma;
25
26 gnuplot_ctrl *h;
27
28 int AVG_ERROR = 20;
29 char EMPTY_DATA[2] = {0x00, 0x00};
30 char VOLTAGE_READ[2];
31 int count = 0;
32 int voltage_level = 0;
33 int *VV_ptr;
34
35 struct timespec timespec_pre;
```

```

36 struct timespec timespec_post;
37 struct timespec timespec_inter;
38
39 struct timespec t1, t2;
40 double elapsedTime;
41 float curVoltage;
42 double VoltageValues1[1000000]; //Allocate 2MB, wont work in Windows, only
    Linux
43 int count_VV1;
44 int count_VV2;
45 double VoltageValues2[1000000]; //Allocate 2MB, wont work in Windows, only
    Linux. So around 4MB Stack Variables. wont use them fully. Have to test
    .
46
47 pthread_mutex_t mutex_VV1 = PTHREAD_MUTEX_INITIALIZER;
48 pthread_mutex_t mutex_VV2 = PTHREAD_MUTEX_INITIALIZER;
49 pthread_t DAQthread, Evalthread;
50
51 void dummyLoad(void)
52 {
53     int j;
54     printf("Using a dummy Load to make sure the CPU Clock is right(1 of 2)\n"
55         );
56     for (int i = 0; i < INT_MAX; i++)
57     {
58         j = i;
59     }
60     printf("Using a dummy Load to make sure the CPU Clock is right(2 of 2)\n"
61         );
62     for (int i = 0; i < INT_MAX; i++)
63     {
64         j = i;
65     }
66 }
67
68 std::vector<int> findpositiveZeroCorssings(std::vector<double> input)
69 {
70     std::vector<int> res;
71     for (int i = 0; i < input.size() - 1; i++)
72     {
73         if ((input.at(i) >= 0) && (input.at(i + 1) < 0))
74         {
75             res.push_back(i);
76         }
77     }
78 }

```

```

75     }
76     return res;
77 }
78
79 void assignToCore(int core_nmbr)
80 {
81     cpu_set_t mask;
82     CPU_ZERO(&mask);
83     CPU_SET(core_nmbr, &mask);
84     sched_setaffinity(0, sizeof(mask), &mask);
85 }
86
87 void sleep_until_thread_finished(struct timespec *pre, int delay_ns)
88 {
89     pre->tv_nsec += delay_ns;
90     if (pre->tv_nsec >= 1000 * 1000 * 1000)
91     {
92         pre->tv_nsec -= 1000 * 1000 * 1000;
93         pre->tv_sec++;
94     }
95     clock_nanosleep(CLOCK_MONOTONIC, TIMER_ABSTIME, pre, NULL);
96 }
97 void *evalthread(void *x)
98 {
99     double priv_Voltagelevels[1000000];
100     memset(priv_Voltagelevels, 0, 1000000);
101     int priv_count;
102     std::vector<double> left_timevec(6);
103     std::vector<double> right_timevec(6);
104     std::vector<double> rightvec(6);
105     std::vector<double> leftvec(6);
106     bool testing = true;
107     bool pdf_export = false;
108     assignToCore(2);
109     std::vector<double> std_vector(1000000);
110     std::fill(std_vector.begin(), std_vector.end(), 0);
111     printf("Here we go, Evaluation Thread started\n");
112     FILE *fp;
113
114     bool VV1 = true;
115     fp = fopen("/home/pi/Desktop/VoltageValuetest.txt", "a+");
116     if (fp == NULL)
117     {
118         printf("Cant open LogFile");

```

```

119 }
120 else
121 {
122     while (1)
123     {
124         //Take the AD-Values to private Stack
125         if (VV1 == true)
126         {
127             pthread_mutex_lock(&mutex_VV1);
128             priv_count = count_VV1;
129             memcpy(priv_Voltagelevels, VoltageValues1, 4 * 1000000);
130             memset(VoltageValues1, 0, 1000000); //Zero old array
131             pthread_mutex_unlock(&mutex_VV1); //give mutex back
132             VV1 = false;
133         }
134         else
135         {
136             pthread_mutex_lock(&mutex_VV2);
137             //printf("Eval got MUIEX for VV2\n");
138             priv_count = count_VV2;
139             memcpy(priv_Voltagelevels, VoltageValues2, 4 * 1000000);
140             memset(VoltageValues2, 0, 1000000); //Zero old array
141             pthread_mutex_unlock(&mutex_VV2); //give mutex back
142             VV1 = true;
143         }
144         //eval the private Stack
145         //fwrite(priv_Voltagelevels, sizeof(float), 1000000, fp);
146         if (testing)
147         {
148
149             if (pdf_export)
150             {
151                 gnuplot_setstyle(h, "points");
152                 gnuplot_set_xlabel(h, "Zeit");
153                 gnuplot_set_xlabel(h, "Spannung");
154                 gnuplot_cmd(h, "set style linespoints");
155                 gnuplot_cmd(h, "set terminal postscript portrait enhanced color
156                 dashed lw 1 12");
157                 gnuplot_cmd(h, "set output \"temp.ps\");
158                 gnuplot_cmd(h, "set yrange[-400:400]");
159                 gnuplot_plot_x(h, priv_Voltagelevels, 20000, "Netzspannung");
160                 gnuplot_close(h);
161             }
162         }
163     }
164 }
165 else

```

```

162     {
163
164         gnuplot_setstyle(h, "dots");
165         gnuplot_plot_x(h, priv_Voltagelevels, 100000, "Netzspannung");
166         gnuplot_close(h) ;
167     }
168     testing = false; //Only do it once.
169 }
170 copy(priv_Voltagelevels, priv_Voltagelevels + 1000000, std_vector.
begin());
171
172 rowvec arma_vec(std_vector);
173 cout << arma_vec.size() << endl;
174 printf("The exact size of VV right now is: %d\n", priv_count);
175 //gnuplot_plot_x(h, priv_Voltagelevels, 24000,"Netzspannung");
176 //uvec zC= find(arma_vec>0.0);
177 std::vector<int> zCVec = findpositiveZeroCorssings(std_vector);
178 int leftpoint = zCVec.front();
179 int rightpoint = zCVec.back();
180 leftvec = {std_vector.begin() + (leftpoint - 2), std_vector.begin() +
(leftpoint + 4)};
181 std::vector<double> left_timevec{leftpoint - 1, leftpoint, leftpoint
+ 1, leftpoint + 2, leftpoint + 3, leftpoint + 4};
182 rightvec = {std_vector.begin() + (rightpoint - 2), std_vector.begin()
+ (rightpoint + 4)};
183 std::vector<double> right_timevec{rightpoint - 1, rightpoint,
rightpoint + 1, rightpoint + 2, rightpoint + 3, rightpoint + 4};
184 std::map<double, double> leftvector;
185 std::map<double, double> rightvector; //make a time to voltage map.
It is like a pair
186 for (size_t i = 0; i < leftvec.size(); ++i)
187 {
188     leftvector[left_timevec[i]] = leftvec[i];
189 }
190 for (size_t i = 0; i < rightvector.size(); ++i)
191 {
192     rightvector[right_timevec[i]] = rightvec[i];
193 }
194
195 //boost::math::interpolators::cubic_b_spline<double> spline(
std_vector.begin(), std_vector.end(), 0.000000, 1);
196
197 cout << "We have" << zCVec.size() << ". Zero Crosses are at: " <<
endl;

```



```

198     for (int i = 0; i < zCVec.size(); i++)
199     {
200         cout << zCVec.at(i) << ";";
201     }
202     cout << endl;
203     cout << "Vector left" << endl;
204
205     for (int i = 0; i < leftvec.size(); i++)
206     {
207         cout << leftvec.at(i) << ",";
208     }
209     cout << endl;
210     cout << "Vector right" << endl;
211     for (int i = 0; i < rightvec.size(); i++)
212     {
213         cout << rightvec.at(i) << ",";
214     }
215
216     //BOOST LIBRARY SECTION: INTERPOLATE THE REAL CROSSING
217     boost::uintmax_t iterations = std::numeric_limits<boost::uintmax_t>::
max();
218     //boost::math::cubic_b_spline<double> interp_func(leftvec.begin(),
leftvec.end(),0,0.01);
219
220     //left vector
221     auto x_range_left = boost::adaptors::keys(leftvector);
222     auto y_range_left = boost::adaptors::values(leftvector);
223     boost::math::barycentric_rational<double> b_left(x_range_left.begin(),
x_range_left.end(), y_range_left.begin());
224
225     double abscissa_left = boost::math::tools::bracket_and_solve_root
([=](double x) { return b_left(x); }, 0.6, 1.2, false, boost::math::
tools::eps_tolerance<double>(), iterations).first;
226     */
227
228     //right vector
229     auto x_range_right = boost::adaptors::keys(rightvector);
230     auto y_range_right = boost::adaptors::values(rightvector);
231     boost::math::barycentric_rational<double> b_right(x_range_right.begin()
), x_range_right.end(), y_range_right.begin());
232     double abscissa_right = boost::math::tools::bracket_and_solve_root
([=](double x) { return b_right(x); }, 0.6, 1.2, false, boost::math::
tools::eps_tolerance<double>(), iterations).first;
233

```

```

234
235     cout << endl;
236     //cout << "The interpolated Crossing left: " << std::setprecision(4)
    << abscissa_left << " and right:" << "no Val" << endl;
237     std::fill(std_vector.begin(), std_vector.end(), 0);
238     memset(priv_Voltagelevels, 0, 1000000); //Zero private array
239
240     double Freq_unaccurate = (500000 * (elapsedTime / 1000000000)*49) / (
    (zCVec.back() - zCVec.front()));
241     cout << "Unaccurate Frequency estimation: " << Freq_unaccurate << endl;
242 }
243 fclose(fp);
244 }
245 }
246
247 void *pollingthread(void *x)
248 {
249     double *cur_ptr;
250     double oneSecTimer;
251     bool VV1 = true; //Assign the first Variable section for making sure that
    the first is always the first
252     assignToCore(3);
253     struct sched_param param;
254     param.sched_priority = 99;
255     pthread_setschedparam(pthread_self(), SCHED_RR, &param);
256     if (mlockall(MCL_CURRENT | MCL_FUTURE))
257     {
258         printf("Failed to set the Memory Lock, unknown behavior, swap is
    possible.\n");
259     };
260
261     dummyLoad();
262
263     printf("Here we go, DAQ Thread started\n");
264
265     while (1)
266     { //Here is our endless while Loop. Take care.
267         printf("New iteration of one DAQ Loop\n");
268         if (VV1 == true)
269         {
270             errno = pthread_mutex_trylock(&mutex_VV1);
271             if (errno)
272             {

```

```

273     printf("DAQ Thread was locked. I wont go further, reason: %s\n",
strerror(errno));
274     exit(-1);
275 }
276 cur_ptr = VoltageValues1;
277 }
278 else
279 {
280     errno = pthread_mutex_trylock(&mutex_VV2);
281     if (errno)
282     {
283         printf("DAQ Thread was locked. I wont go further, reason: %s\n",
strerror(errno));
284         exit(-1);
285     }
286     cur_ptr = VoltageValues2;
287 }
288
289 clock_gettime(CLOCK_MONOTONIC, &t1);
290 count = 0;
291 clock_gettime(CLOCK_MONOTONIC, &timespec_pre);
292 while (count < 500000)
293 {
294     bcm2835_spi_transfernb(EMPTY_DATA, VOLTAGE_READ, 2);
295     voltage_level = (VOLTAGE_READ[0] - 127);
296     voltage_level = voltage_level << 8;
297     voltage_level = voltage_level + VOLTAGE_READ[1] - AVG_ERROR;
298     cur_ptr[count] = 0.015 * voltage_level;
299     count++;
300 }
301
302 clock_gettime(CLOCK_MONOTONIC, &t2);
303 elapsedTime = (t2.tv_sec - t1.tv_sec) * 1000000000.0;
304 elapsedTime += (t2.tv_nsec - t1.tv_nsec);
305 printf("\nElapsed %dkS in %f ms \n", count / 1000, elapsedTime /
1000000);
306
307 clock_gettime(CLOCK_MONOTONIC, &timespec_post);
308 timespec_post.tv_nsec += 100000;
309 if (timespec_post.tv_nsec >= 1000000000)
310 {
311     timespec_post.tv_nsec -= 1000000000;
312     timespec_post.tv_sec++;
313 }

```

```
314
315     if (VV1 == true)
316     {
317         count_VV1 = elapsedTime;
318         if (pthread_mutex_unlock(&mutex_VV1))
319         {
320             printf("Severe Problem with unlocking VV1 mutex");
321         }
322         VV1 = false;
323     }
324     else
325     {
326         count_VV2 = elapsedTime;
327         if (pthread_mutex_unlock(&mutex_VV2))
328         {
329             printf("Severe Problem with unlocking VV2 mutex");
330         }
331         VV1 = true;
332     }
333
334     clock_nanosleep(CLOCK_MONOTONIC, TIMER_ABSTIME, &timespec_post, NULL);
335 }
336 }
337
338 int main(int argc, char **argv)
339 {
340     h = gnuplot_init();
341
342
343     if (mlockall(MCL_CURRENT | MCL_FUTURE))
344     {
345         printf("Failed to set the Memory Lock\n");
346     };
347
348     if (!bcm2835_init())
349     {
350         printf("BCM Library Error: Maybe Root Access failed\n");
351         return 1;
352     }
353
354     if (!bcm2835_spi_begin())
355     {
356         printf("BCM Library Error: Maybe Root Access failed\n");
357         return 1;
```

```

358 }
359
360
361 memset(VoltageValues1, 0, 1000000); //Zero private array
362 memset(VoltageValues2, 0, 1000000); //Zero private array
363
364 sleep(1);
365 bcm2835_spi_setBitOrder(BCM2835_SPI_BIT_ORDER_MSBFIRST); // The default
366 bcm2835_spi_setDataMode(BCM2835_SPI_MODE0); // The default
367 bcm2835_spi_setClockDivider(BCM2835_SPI_CLOCK_DIVIDER_32); // The default
368 bcm2835_spi_chipSelect(BCM2835_SPI_CS0); // The default
369 bcm2835_spi_setChipSelectPolarity(BCM2835_SPI_CS0, LOW); // the default
370
371 char PGA_DATA[4] = {0b11010000, 0b00010100, 0b00000000, 0b00000001};
372 char READ_PGA_DATA[4];
373 bcm2835_spi_transfernb(PGA_DATA, READ_PGA_DATA, 4);
374 printf("Sent to SPI: %d %d %d %d Read back from SPI: %d %d %d %d\n",
        PGA_DATA[0], PGA_DATA[1], PGA_DATA[2], PGA_DATA[3], READ_PGA_DATA[0],
        READ_PGA_DATA[1], READ_PGA_DATA[2], READ_PGA_DATA[3]);
375
376 pthread_create(&DAQthread, NULL, pollingthread, NULL);
377 pthread_create(&Evalthread, NULL, evalthread, NULL);
378 pthread_detach(DAQthread);
379 pthread_join(Evalthread, NULL);
380
381 //Close all connections to the ADS8681
382 bcm2835_spi_end();
383 bcm2835_close();
384 printf("I closed everything.");
385
386 return 0;
387 }

```

B Matlab Code

B.1 SNR-Bit-Tiefe-Simulation

```

1  % This file is for testing a specific Word and SNR Range for a given ADC
2  clc; clear all; close all;
3
4  bit_width=16; % This value is the bitlength of the ADC
5  SNR = 30; % This value is in dB, as AWCN
6  SPS = 1e5; % The Samples per second.
7  Ts=1/SPS; % The Timem per every sample
8
9  T = 1; % Time in seconds to simulate
10
11 N = T*SPS;
12 df = SPS/N;
13 tiledlayout(4,1);
14
15 f = (0:df:(N/2-1)*df);
16 Frequency_Points = [50,100,150,200,250,300,350,400,450,500,550,600
17 ,650,700,750,800,850,900,950,1000,1050,1100,1150,1200,1250];
18 Amplitude_Points = [100,20,5,1,6,0.5,5.0,0.5,0.5,
19 0.5,3.5,0.5,3.0,0.5,0.5,0.5,2.0,0.5,1.5,
20 0.5,0.5,0.5,1.5,0.5,1.5];
21 Amplitude_Points = Amplitude_Points/100;
22
23 t=0:Ts:T-Ts;
24 x=int64(0);
25 x = Amplitude_Points(1)*sin(2*pi*Frequency_Points(1)*t);
26 for i =2:length(Frequency_Points)
27     x = x+ Amplitude_Points(i)*sin(2*pi*Frequency_Points(i)*t);
28 end
29
30 %FFT the normal signal (single sided)
31 xa= fft(x)/length(x);
32 nexttile;

```

```

33 x_normalized = 2*abs(xa(1:length(x)/2));
34 %p_x_normalized = abs(xa(1:length(xa)/2)).^2;
35 plot(f,mag2db(x_normalized));
36 title('FFT des perfekt abgetastetes Signal');
37 axis([30 1300 -360 3]);
38 %Quantize signals
39
40 x_quantized = quant2c(x,bit_width,'f');
41 xq_a= fft(x_quantized)/length(x_quantized);
42 nexttile;
43 xq_normalized = 2*abs(xq_a(1:length(xq_a)/2));
44 %p_xq_normalized = abs(xq_a).^2;
45 plot(f,mag2db(xq_normalized));
46 title('FFT des quantisiertes Signals');
47 axis([30 1300 -360 3]);
48
49 %Add AWGN to the quantized signals
50 %output = x_quantized;
51 output= awgn(x_quantized,SNR);
52 output = output/max(output);%adding the noise floor
53
54 %FFT the new signal
55 nexttile;
56 output_f= fft(output)/length(output);
57 output_fft = 2*abs(output_f(1:length(output)/2));
58 plot(f,mag2db(output_fft));
59 title('FFT des Signals mit Addition des AWGNs');
60 axis([30 1300 -360 3]);
61
62 % Plot the difference between our former signal and the measured signal
63 nexttile;
64 difference = mag2db(output_fft)-mag2db(x_normalized);
65 plot(difference);
66 title('Differenz des Optimums zum reellen Signal');
67
68 axis([30 1300 -50 50]);
69 %
70
71 %% Section test
72 nexttile;
73 periodogram(x,kaiser(length(x),38),[],SPS,'power');
74 nexttile;
75 periodogram(x_quantized,kaiser(length(x),38),[],SPS,'power');

```

B.2 Datenauswertung DT9816

```

1
2
3 Fs = 50000;
4 BufferDivisor = 1;
5 transformer_ratio=15;
6 warning('off');
7 s = daq.createSession('dt');
8 s.Rate =Fs;
9 s.NotifyWhenDataAvailableExceeds = s.Rate/BufferDivisor;
10 ch1=addAnalogInputChannel(s,'DT9816-S(00)', '0', 'Voltage');
11 ch1.Range = [-10 10];
12
13
14 global fileID;
15 global pathStr;
16 global datasource;
17 datasource="MiniWire.de_MariaDB_Netzfrequenz";
18 global username;
19 username="netzfrequenz";
20 global password;
21 password="#####";
22 global tablename;
23 tablename="measurements";
24
25
26 mkdir(append('C:\Users\uj88oval\Desktop\50Hz_Log\',
27 datestr(now,'yyyy\mmm\')));
28 pathStr = append('C:\Users\uj88oval\Desktop\50Hz_Log\',
29 datestr(now,'yyyy\mmm\dd_'),'Log.txt');
30 fileID = fopen(pathStr,'a'); % append to file
31
32
33
34
35
36
37
38
39 lh = addlistener(s,'DataAvailable', @(src,evnt) func_analyzeDT9816(src,evnt));
40
41
42

```



```

43
44     Use this if you want the Data to be saved in RAM
45     global dataBuffer;
46     lh=addlistener(s,'DataAvailable', @(src, evnt) func_plotAndSaveDT9816(src, evnt));
47
48
49
50
51     s.IsContinuous = true;
52     startBackground(s);
53
54
55     s.stop();
56     fclose(fileID);
57     delete (lh);
58     delete (s);
59     %%

```

B.3 Wrapper für die DT9816 zum Benutzen der func analyze

```

1  function func_analyzedT9816(src, event)
2  %FUNC_ANALYZEDT9816 Summary of this function goes here
3  %
4  Fs = 50000;
5  buf = event.Data';
6  global fileID;
7  global pathStr;
8  global gauge;
9  global datasource;
10 global username;
11 global password;
12 global tablename;
13 connection=database(datasource, username, password);
14
15 curPathStr = append('C:\Users\uj88oval\Desktop\50Hz_Log\ ',
16 datestr(now, 'yyyy\ \mmm\ \dd_'), 'Log.txt ');
17 if pathStr == curPathStr
18 else
19     mkdir(append('C:\Users\uj88oval\Desktop\50Hz_Log\ ',
20 datestr(now, 'yyyy\ \mmm\ \ ')));
21     fclose(fileID);
22     fileID = fopen(curPathStr, 'a');

```

```

23     pathStr = curPathStr;% Date Changing right now. New day starts.
24 end
25
26 [public_Frequency,public_Harmonics] = func_analyze(buf,Fs);
27 % datestr(now,'yyyy.mm.dd HH:MM:SS,FFF')
28 %     Date Freq 50hz max   harm harm harm
29
30 fmt = '%s | %.6f | %f | %f | %f | %f | %f | %f | %f | %f | %f | %f | %f
31 | %f | %f | %f | %f | %f | %f | %f | %f | %f | %f | %f | %f | %f
32 %f | %f | %f | %f | %f | %f | %f | %f | %f | %f | %f | \r\n' ;
33 s(1).id=0;
34 s(1).timestamp=datestr(now,'yyyy-mm-dd HH:MM:SS,FFF');
35 s(1).freq=round(public_Frequency,3);
36 s(1).a50=public_Harmonics(1);
37 s(1).a100=public_Harmonics(2);
38 s(1).a150=public_Harmonics(3);
39 s(1).a200=public_Harmonics(4);
40 s(1).a250=public_Harmonics(5);
41 s(1).a300=public_Harmonics(6);
42 s(1).a350=public_Harmonics(7);
43 s(1).a400=public_Harmonics(8);
44 s(1).a450=public_Harmonics(9);
45 s(1).a500=public_Harmonics(10);
46 s(1).a550=public_Harmonics(11);
47 s(1).a600=public_Harmonics(12);
48 s(1).a650=public_Harmonics(13);
49 s(1).a700=public_Harmonics(14);
50 s(1).a750=public_Harmonics(15);
51 s(1).a800=public_Harmonics(16);
52 s(1).a850=public_Harmonics(17);
53 s(1).a900=public_Harmonics(18);
54 s(1).a950=public_Harmonics(19);
55 s(1).a1000=public_Harmonics(20);
56 s(1).a1050=public_Harmonics(21);
57 s(1).a1100=public_Harmonics(22);
58 s(1).a1150=public_Harmonics(23);
59 s(1).a1200=public_Harmonics(24);
60 s(1).a1250=public_Harmonics(25);
61 Struct=struct2table(s);
62
63 %gauge.Value = public_Frequency;
64
65
66 logstr = sprintf(fmt,datestr(now,'dd.mm.yyyy HH:MM:SS,FFF') ,

```

```
67 public_Frequency , public_Harmonics(1) , public_Harmonics(2) ,
68 public_Harmonics(3) , public_Harmonics(4) , public_Harmonics(5)
69 , public_Harmonics(6) , public_Harmonics(7) , public_Harmonics(8)
70 , public_Harmonics(9) , public_Harmonics(10) , public_Harmonics(11)
71 , public_Harmonics(12) , public_Harmonics(13) , public_Harmonics(14)
72 , public_Harmonics(15) , public_Harmonics(16) , public_Harmonics(17)
73 , public_Harmonics(18) , public_Harmonics(19) , public_Harmonics(20)
74 , public_Harmonics(21) , public_Harmonics(22) , public_Harmonics(23)
75 , public_Harmonics(24) , public_Harmonics(25))
76
77 sqlwrite(connection , tablename , Struct );
78 fprintf(fileID , logstr );
79 end
```

B.4 Funktion zur Detektion von Nullstellen mit Auswertung der Frequenz und Oberschwingungen

```
1 function [Frequencies , Harmonics] = func_analyze(inputBuffer , Fs)
2
3
4 Interpolation = 1;
5 Freq_Offset = 0;
6 inputBuffer = resample(inputBuffer , Interpolation , 1);
7
8 Finden der eigentlich steigenden/fallenden Flanken
9 rising_edge = find((inputBuffer(1:end-1) <= 0 & inputBuffer(2:end) > 0));
10 crossings = rising_edge;
11 crossings = crossings(2:end-1);
12 NumGridPoints = 3;
13 PeriodsForMedian = 'all';
14 if(PeriodsForMedian == 'all')
15
16     timevec_left = crossings(1)-NumGridPoints:crossings(1)+NumGridPoints;
17     timevec_right = crossings(end)-NumGridPoints:crossings(end)+NumGridPoints;
18
19     vector_left = inputBuffer(timevec_left);
20     vector_right = inputBuffer(timevec_right);
21
22     zero_cross_left = interp1(vector_left , timevec_left , 0.0 , 'spline');
23     zero_cross_right = interp1(vector_right , timevec_right , 0.0 , 'spline');
24
25     Frequencies = Fs*(length(crossings)-1)/((zero_cross_right - zero_cross_left)*
26     Interpolation);
```

```

27 else
28     for i = PeriodsForMedian:PeriodsForMedian:length(crossings)-PeriodsForMedian
29
30         timevec_left = crossings(i)-NumGridPoints:crossings(i)+
31             NumGridPoints+1;
32         timevec_right = crossings(i+PeriodsForMedian)-NumGridPoints:
33             crossings(i+PeriodsForMedian)+NumGridPoints+1;
34
35         vector_left = inputBuffer(timevec_left);
36         vector_right = inputBuffer(timevec_right);
37
38         zero_cross_left = interp1(vector_left,timevec_left,0.0000,'spline');
39         zero_cross_right = interp1(vector_right,timevec_right,0.0000,'spline');
40
41         Frequencies(i/PeriodsForMedian) = (PeriodsForMedian*Fs/
42             (zero_cross_right - zero_cross_left)*Interpolation)+Freq_Offset;
43     end
44 end
45
46 fft_vector = inputBuffer(rising_edge(1):rising_edge(end));
47 scaling_factor = length(fft_vector);
48 p = 2^nextpow2(length(fft_vector));
49 padding = p - length(fft_vector);
50 fft_vector = [fft_vector, zeros(1,padding)];
51 N = length(fft_vector);
52 dF = Fs/(N);
53 freq_grid = (-N/2*dF:dF:((N/2)-1)*dF);
54 fft_output = fftshift(fft(fft_vector))*2/scaling_factor;
55
56 plot(freq_grid, abs(fft_output));
57 semilogy(freq_grid, abs(fft_output))
58 Frequency_Points = [50,100,150,200,250,300,350,400,450,500,
59 550,600,650,700,750,800,850,900,950,1000,1050,1100,1150,1200,1250];
60 Point_Margin = 1;
61
62 Harmonics = zeros(1,2);
63
64
65 for i = 1:length(Frequency_Points)
66
67     Vector_Points(i) = max(find(freq_grid >
68         (Frequency_Points(i)-Point_Margin) & freq_grid <
69         (Frequency_Points(i) + Point_Margin)));
70

```

```
71     Harmonics(i) = max(abs(fft_output(Vector_Points(i)-3:Vector_Points(i)+3)));
72
73 end
74 end
```

C CAD-Zeichnungen

C.1 Schaltpläne

C.1.1 Isolationsboard

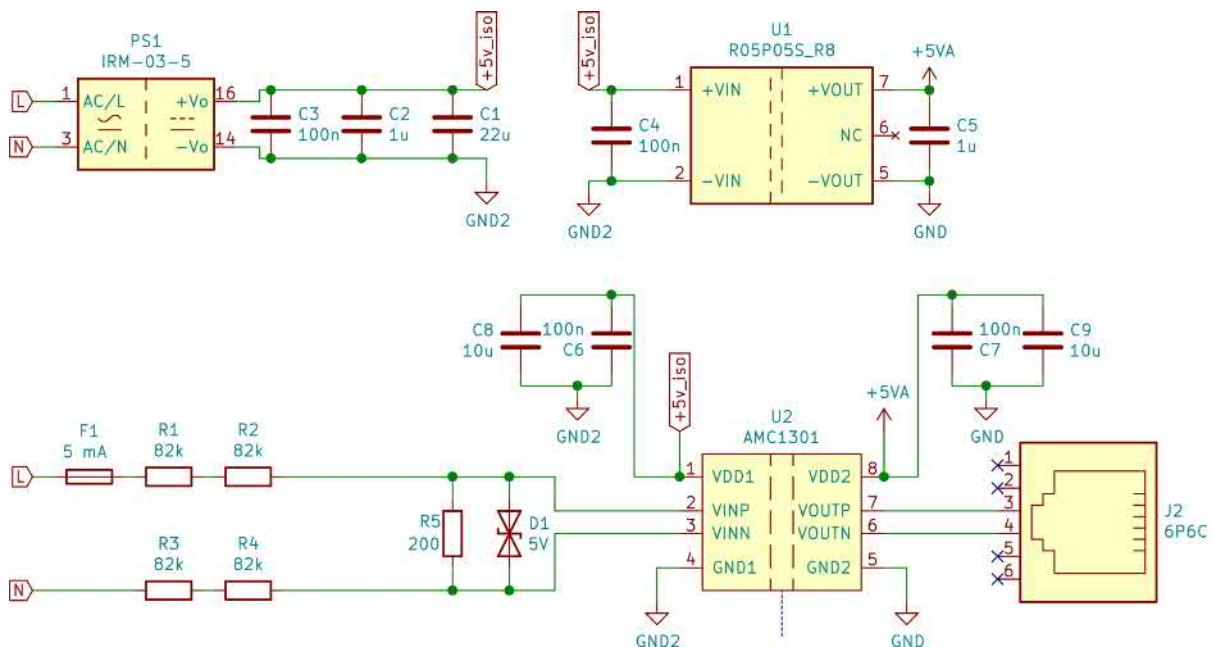


Abbildung C.1: Schaltung des Isolationsboards

C.1.2 HAT-Platine

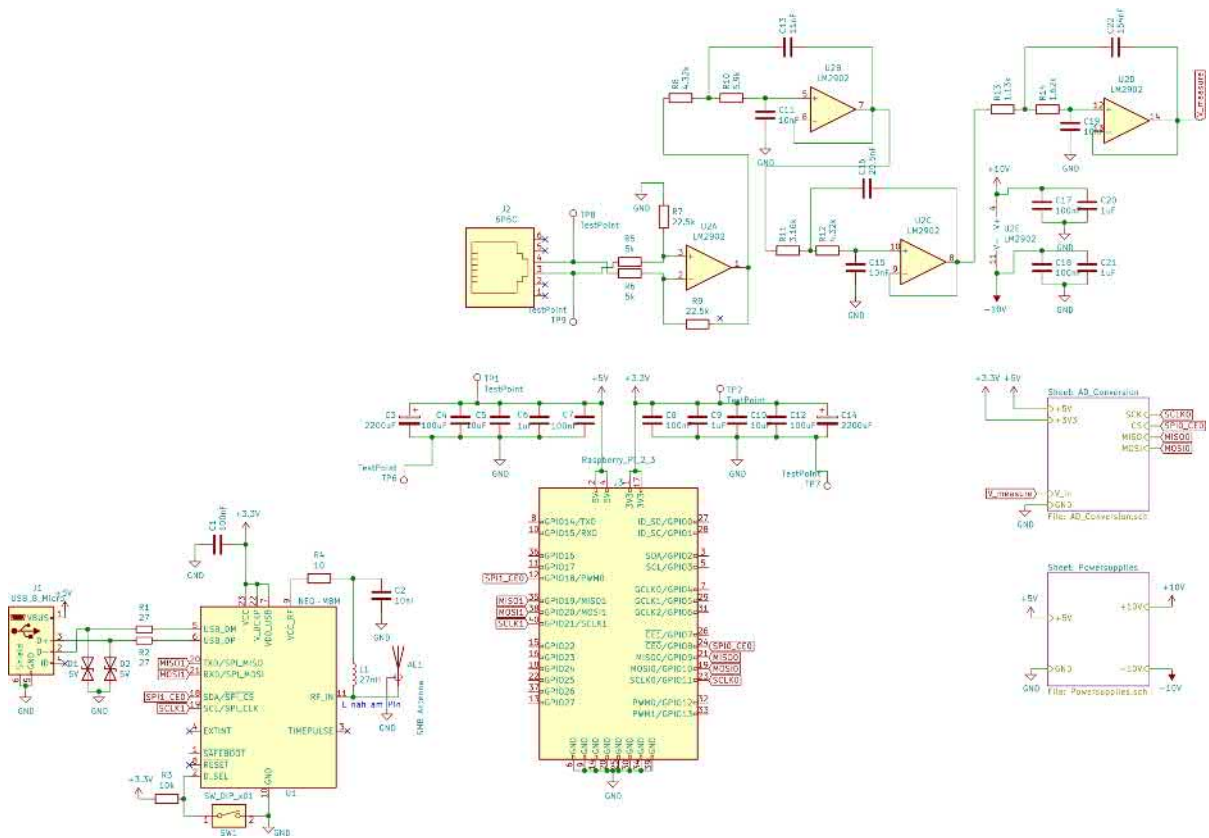


Abbildung C.2: Schaltung der HAT-Platine

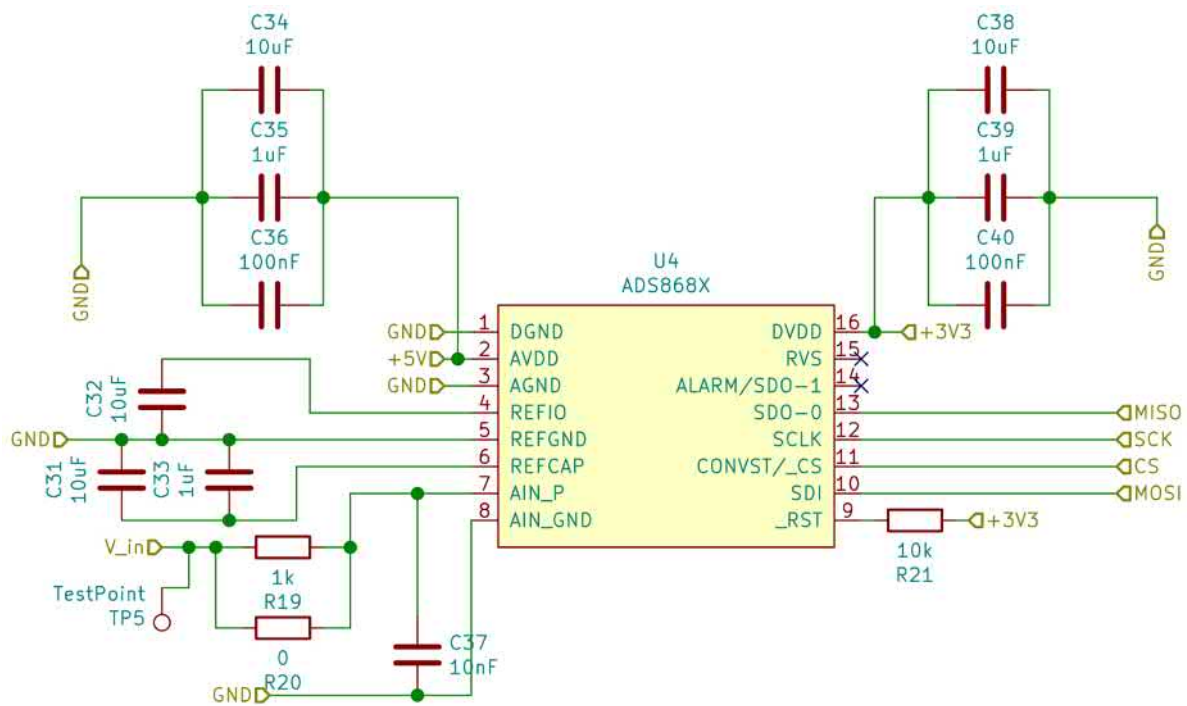


Abbildung C.3: Schaltung des Analog/Digital-Umsetzers

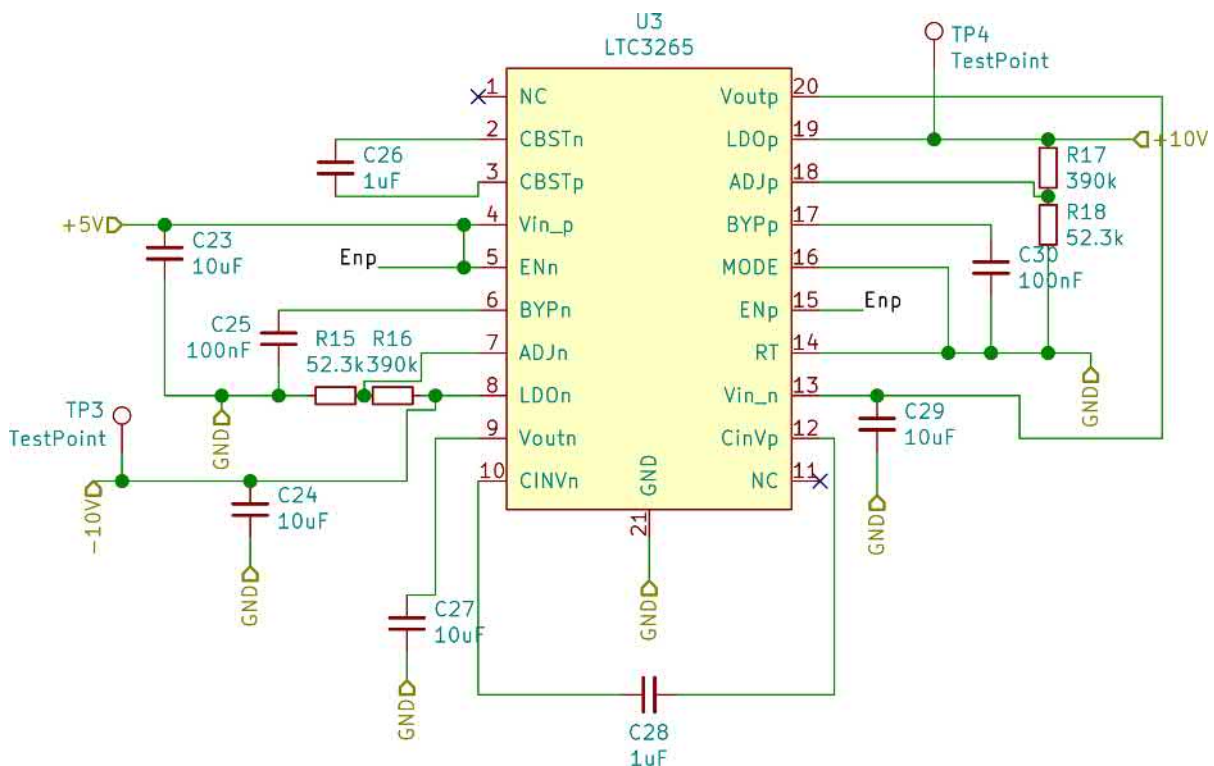


Abbildung C.4: Schaltung der LTC3265 Ladungspumpe

C.2 Platinentflechtungen

C.2.1 Isolationsboard

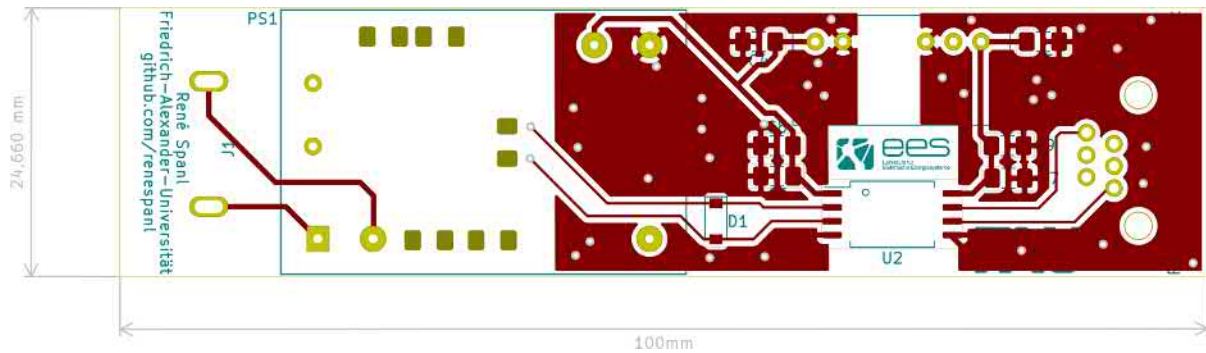


Abbildung C.5: Oberseite des Isolationsboards

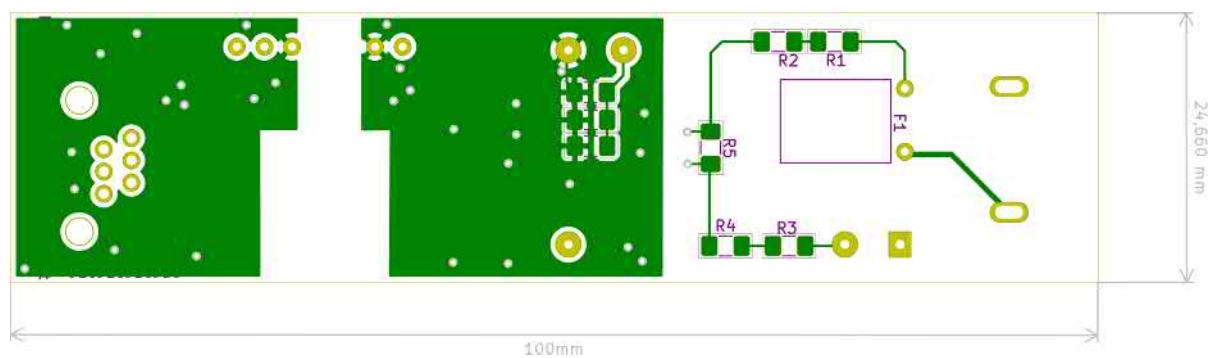


Abbildung C.6: Unterseite des Isolationsboards

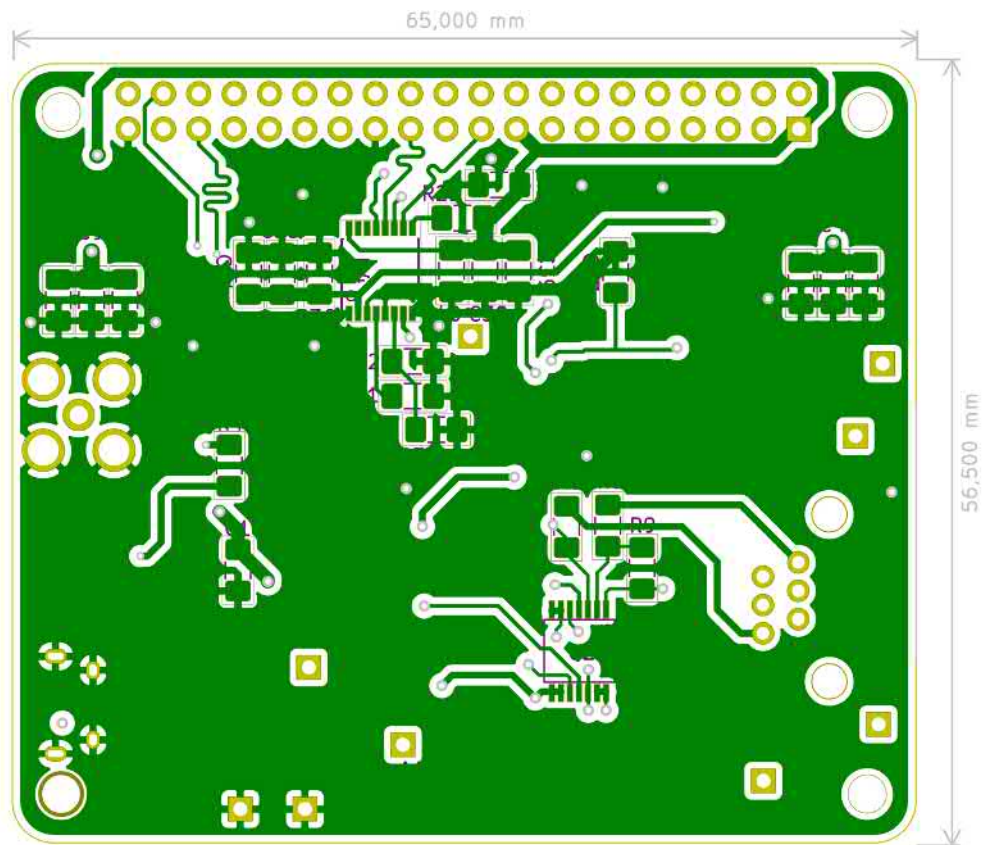


Abbildung C.8: Unterseite der HAT-Platine

D Prüfprotokolle

D.1 DIN EN 61010 SK2,SK3

Prüfprotokoll Überprüfung der elektrischen Sicherheit nach EN 61010

Auftrag Nummer:			
Auftraggeber: Erlangen		Auftragnehmer: Uni. Erlangen Cauerstr. 5 91058 Erlangen	
Geräteart:		Hersteller:	
Typenbezeichnung:		Schutzkl.: II	
Ident. Nr.: @1172			
Prüfung nach EN 61010		Prüfdatum: 30.03.21	
Besichtigung:			
[-] Schutzleiter in Ordnung (nur bei Schutzklasse I)			
[OK] Gehäuse und mechanische Teile in Ordnung			
[OK] Isolierteile in Ordnung			
[OK] Geräte-Anschlussleitungen einschließlich Steckvorrichtungen mängelfrei			
[OK] Aufschriften vorhanden bzw. vervollständigt, Typschild korrekt			
[OK] Sonstiges			
Messung	Messwert	Grenzwert	OK
Schutzleiterwiderstand [R _{SL}]			-
Isolationswiderstand [R _{ISO}]	>+310.0MΩ	>2.000MΩ	OK
Differenzstrom [I _{DIFF}]		<03.50mA	-
Gehäuseableitstrom [I _{G-NC}]	+001.4μA	<0.500mA	OK
Gehäuseableitstrom Single Fault Condition [I _{G-SFC}]		<03.50mA	
Hochspannungsprüfung [+05.37kV]			OK
Funktionstest			-
[OK] Funktions- und Sicherheitsprüfung mängelfrei			
Prüfzyklus 12 Monate Nächster Prüftermin : 30.03.2022			
Verwendete Messgeräte	Typ: SECUTEST 0751/601 / SIII	Fabrikat: GOSSEN-METRAWATT	
Unterschrift:			
Prüfer:			
Ort: _____ Datum 30.03.2021			

Abbildung D.1

Prüfprotokoll

Überprüfung der elektrischen Sicherheit nach EN 61010

Auftrag Nummer:			
Auftraggeber:		Auftragnehmer:	
Erlangen		Uni. Erlangen Cauerstr. 5 91058 Erlangen	
Geräteart:		Hersteller:	
Typenbezeichnung:		Schutzkl.: III	
Ident. Nr.: @1171			
Prüfung nach EN 61010		Prüfdatum: 30.03.21	
Besichtigung:			
☐ Schutzleiter in Ordnung (nur bei Schutzklasse I) ☒ Gehäuse und mechanische Teile in Ordnung ☒ Isolierteile in Ordnung ☒ Geräte-Anschlussleitungen einschließlich Steckvorrichtungen mängelfrei ☒ Aufschriften vorhanden bzw. vervollständigt, Typschild korrekt ☒ Sonstiges			
Messung	Messwert	Grenzwert	OK
Schutzleiterwiderstand [R_{SL}]			-
Isolationswiderstand [R_{ISO}]	>+310.0MΩ	>2.000MΩ	OK
Differenzstrom [I_{DIFF}]		<0.50mA	-
Gehäuseableitstrom [I_{G-NC}]	+001.2μA	<0.500mA	OK
Gehäuseableitstrom Single Fault Condition [I_{G-SFC}]		<0.50mA	
Hochspannungsprüfung [+05.37kV]			OK
Funktionstest			-
☒ Funktions- und Sicherheitsprüfung mängelfrei			
Prüfzyklus 12 Monate Nächster Prüftermin : 30.03.2022			
Verwendete Messgeräte	Typ:	Fabrikat:	
	SECUTEST 0751/601 / SIII	GOSSEN-METRAWATT	
Unterschrift:			
Prüfer:			
Ort:		Datum 30.03.2021	

Abbildung D.2

D.2 DIN VDE 0701-0702

Prüfprotokoll für die Sicherheitsprüfung elektrischer Geräte gemäß DIN VDE 0701-0702

Auftrag Nummer:			
Auftraggeber (Kunde):		Auftragnehmer:	
Erlangen		Uni. Erlangen Cauerstr. 5 91058 Erlangen	
Geräteart:	Hersteller:	Type:	
Ident. Nr.: @1180	Schutzkl.: II	Baujahr:	
Nennstr. (A):	Nennspg.(V):	Nennleistg.(W):	
Prüfung am: 30.03.21			
Prüfung nach DIN VDE 0701-0702			
Besichtigung:			
☐ Schutzleiter in Ordnung (nur bei Schutzklasse I) ☒ Gehäuse und mechanische Teile in Ordnung ☒ Isolierteile in Ordnung ☒ Geräte-Anschlussleitungen einschließlich Steckvorrichtungen mängelfrei ☒ Aufschriften vorhanden bzw. vervollständigt, Typschild korrekt ☒ Sonstiges			
Messung	Messwert	Grenzwert	OK
Schutzleiterwiderstand [Ω]			-
Differentieller Schutzleiterwiderstand [Ω]			-
Isolationswiderstand [$M\Omega$]	>+310.0M Ω m	>2.000M Ω m	OK
Schutzleiterstrom			
Ersatzableitstromverfahren [mA]			-
Differenzstromverfahren [mA]			-
Berührungstrom [mA]	+0.004mA	<0.500mA	OK
Hochspannungsprüfung []			-
Funktionstest			-
☒ Funktions- und Sicherheitsprüfung mängelfrei ☐ Das Gerät kann nicht mehr instandgesetzt werden Das Gerät hat erhebliche sicherheitstechnische Mängel, es besteht ☐ Brandgefahr ☐ Gefahr durch elektrischen Schlag ☐ mechanische Gefahr			
Gemäß Unfallverhütungsvorschrift BGV A3 (ehemals VBG4):			
Prüfzyklus 12 Monate Nächster Prüftermin : 30.03.2022			
Verwendete Messgeräte			
Fabrikat GOSSEN-METRAWATT		Typ SECUTEST ...	
Unterschriften			
Prüfer:		Verantwortlicher Unternehmer:	
Ort:	Datum: 30/03/2021	Ort:	Datum: 30/03/2021

Abbildung D.3

D.3 DIN EN 60335 SK2,SK3

Prüfprotokoll zur Überprüfung der elektrischen Sicherheit nach DIN VDE0700 / EN 60335

Auftrag Nummer: -			
Auftraggeber: Erlangen		Auftragnehmer: Uni. Erlangen Cauerstr. 5 91058 Erlangen	
Geräteart: Trennelement		Hersteller:	
Typenbezeichnung:		Schutzkl.: II	
Ident. Nr.: @1169			
Prüfung nach DIN VDE 0700 / EN 60335		Prüfdatum: 30.03.21	
Besichtigung: [-] Schutzleiter in Ordnung (nur bei Schutzklasse I) [OK] Gehäuse und mechanische Teile in Ordnung [OK] Isolierteile in Ordnung [OK] Geräte-Anschlussleitungen einschließlich Steckvorrichtungen mängelfrei [OK] Aufschriften vorhanden bzw. vervollständigt, Typschild korrekt [OK] Sonstiges			
Messung	Messwert	Grenzwert	OK
Schutzleiterwiderstand [R_{SL}]			-
Isolationswiderstand [R_{ISO}]	>+310.0MΩ	>2.000MΩ	OK
Differenzstrom [I_{DIFF}]	+0.027mA	<0.50mA	OK
Ersatzgeräteableitstrom [I_{EA-NC}]	+0.000mA	<0.250mA	OK
Hochspannungsprüfung [+05.37kV]			OK
Funktionstest			-
[OK] Funktions- und Sicherheitsprüfung mängelfrei			
Prüfzyklus 12 Monate Nächster Prüftermin : 30.03.2022			
Verwendete Messgeräte	Typ: SECUTEST 0751/601 / SIII	Fabrikat: GOSSEN-METRAWATT	
Unterschrift:			
Prüfer:			
Ort: Datum 30.03.2021			

Abbildung D.4

Prüfprotokoll zur Überprüfung der elektrischen Sicherheit nach DIN VDE0700 / EN 60335

Auftrag Nummer: -			
Auftraggeber: Erlangen		Auftragnehmer: Uni. Erlangen Cauerstr. 5 91058 Erlangen	
Geräteart:		Hersteller:	
Typenbezeichnung:		Schutzkl.: III	
Ident. Nr.: @1170			
Prüfung nach DIN VDE 0700 / EN 60335		Prüfdatum: 30.03.21	
Besichtigung:			
[-] Schutzleiter in Ordnung (nur bei Schutzklasse I)			
[OK] Gehäuse und mechanische Teile in Ordnung			
[OK] Isolierteile in Ordnung			
[OK] Geräte-Anschlussleitungen einschließlich Steckvorrichtungen mängelfrei			
[OK] Aufschriften vorhanden bzw. vervollständigt, Typschild korrekt			
[OK] Sonstiges			
Messung	Messwert	Grenzwert	OK
Schutzleiterwiderstand [R _{SL}]			-
Isolationswiderstand [R _{ISO}]	>+310.0MΩ	>0.250MΩ	OK
Differenzstrom [I _{DIFF}]	+0.010mA	<0.50mA	OK
Ersatzgeräteableitstrom [I _{EA-NC}]	+0.000mA	<0.500mA	OK
Hochspannungsprüfung [+05.37kV]			OK
Funktionstest			-
[OK] Funktions- und Sicherheitsprüfung mängelfrei			
Prüfzyklus 12 Monate Nächster Prüftermin : 30.03.2022			
Verwendete Messgeräte	Typ: SECUTEST 0751/601 / SIII	Fabrikat: GOSSEN-METRAWATT	
Unterschrift:			
Prüfer:			
Ort:	Datum 30.03.2021		

Abbildung D.5

Literatur

- [1] D. P. et al., „Zeit- und Normalfrequenzverbreitung mit DCF77,“ *PTB-Mitteilungen* 114 Heft 4, März 2004.
- [2] ENTSO-E, „Load-Frequency Control and Performance,“ März 2009.
- [3] ENTSO-E und F. R. et al., „Continental Europe Synchronous Area Separation on 8 January 2021- Interim Report,“ Feb. 2021.
- [4] EverythingRF, *Microstrip Impedance Calculator*. Adresse: <https://www.everythingrf.com/rf-calculators/microstrip-impedance-calculator>.
- [5] Intel, *IA-PC HPET (High Precision Event Timers) Specification*, 2004. Adresse: <https://www.intel.com/content/dam/www/public/us/en/documents/technical-specifications/software-developers-hpet-spec-1-0a.pdf>.
- [6] W. Kester, *The Data Conversion Handbook*, ISBN: 9780750678414. Adresse: <https://www.analog.com/en/education/education-library/data-conversion-handbook.html>.
- [7] T. Kuttner, „Verfahren und Beispiele zur Signalanalyse,“ in *Praxiswissen Schwingungsmesstechnik*. Wiesbaden: Springer Fachmedien Wiesbaden, 2015, S. 263–335, ISBN: 978-3-658-04638-5.
- [8] „Gerthsen Physik,“ in, D. Meschede, C. Gerthsen und H. Vogel, Hrsg., 25. Auflage, Ser. Springer-Lehrbuch. Berlin Heidelberg: Springer Spektrum, 2015, Kap. Elektrodynamik.
- [9] H. B. Messtechnik, *Precision Time Protocol bei Datenerfassung und Prüfung*. Adresse: <https://www.hbm.com/de/5143/precision-time-protokoll/>.

- [10] J. Ohm und H. D. Lüke, „Korrelationsfunktionen determinierter Signale,“ in *Signalübertragung: Grundlagen der digitalen und analogen Nachrichtenübertragungssysteme*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, S. 115–131, ISBN: 978-3-540-69258-4.
- [11] T. Schüttler, *Satellitenavigation*, 1. Aufl. Springer Vieweg, 2014.
- [12] U. Tietze, C. Schenk und E. Gamm, *Halbleiter-Schaltungstechnik*, 16. Aufl. 2019. Springer Berlin Heidelberg, 1969.
- [13] *Versuch 20: Transformator und Ferromagnetische Hysterese*. Adresse: <https://www.tu-braunschweig.de/index.php?eID=dumpFile&t=f&f=71297&token=54a83f751cc9bf43a1df997aa5e4dc8261bc2622>.
- [14] Vishay, *Surface-Mount Multilayer Ceramic Chip Capacitors for Non-Magnetic Applications*. Adresse: https://www.vishay.com/docs/49003/_non-magnetic_series_vmn-pl0441-1603.pdf.
- [15] P. D.-I. M. Vossiek, *Passive Bauelemente und deren HF-Verhalten - Skript zu Vorlesung und Übung*.
- [16] Z. Yao und M. Lu, *Next-Generation GNSS Signal Design*, 1. Aufl. Springer, Singapore, 2021.