

Peter Lindner

# Dual Lock-in Verstärker für ein Nahinfrarotspektroskopiesystem

Masterarbeit



Institut für Semantische Datenanalyse / Knowledge Discovery  
Laboratory of Brain-Computer Interfaces  
Technische Universität Graz  
Inffeldgasse 13/IV, A - 8010 Graz  
Vorstand: Assoc. Prof. Dipl.-Ing. Dr.techn. Gernot Müller-Putz

Betreuer:

Assoc. Prof. Dipl.-Ing. Dr.techn. Gernot Müller-Putz

Begutachter:

Assoc. Prof. Dipl.-Ing. Dr.techn. Gernot Müller-Putz

Graz, Austria, April 2013

# Kurzfassung

Diese Arbeit beschreibt die Entwicklung eines Lock-in Verstärkers, welcher innerhalb eines Nahinfrarotspektroskopiesystems in einem Brain-Computer-Interface eingesetzt werden kann. Ziel dabei ist es, ein kostengünstiges, portables System zu schaffen, welches die notwendigen Algorithmen mit Hilfe eines digitalen Signalprozessors umsetzt. Zusätzlich zum eigentlichen Messsystem für Nahinfrarotsignale soll dieser Messverstärker auch einen Signalgenerator beinhalten, welcher Referenzsignale verschiedener Frequenzen zur Verfügung stellt. Nach einer theoretischen Einführung in die Thematik wird innerhalb dieser Arbeit gleichermaßen auf die hardwaretechnischen, wie auch auf die softwaretechnischen Aspekte dieser Entwicklung eingegangen. Schlussendlich wird der entwickelte Prototyp vorgestellt, dessen Bedienung erläutert und aufgezeigt, dass dieser Lock-in Verstärker sämtliche Komponenten enthält, die zum Betrieb eines Nahinfrarotspektroskopiesystems notwendig sind.

**Schlüsselwörter.** Lock-in Verstärker, Demodulator, Nahinfrarotspektroskopie, Brain-Computer-Interface, Signalgenerator, Signalprozessor, uCLinux

# Abstract

This masterthesis describes the developement of a lock-in amplifier, which can be established in nearinfraredspectroscopy systems used in brain-computer-interfaces. The goal was to develop a cost-effective, portable system that implies a digital signal processor for calculating the essential algorithms. In addition to the measurement system for nearinfrared signals this amplifier should also contain a signal generator that provides the needed reference signals. First this thesis gives a theoretical introduction into the topic followed by the hard- and softwaredescription of the device. Finally the developed prototype is introduced, the handling of the hard- and software is sketched and it is demonstrated that this device contains all components needed for the operation of a nearinfraredspectroscopy system.

**Keywords.** Lock-in amplifier, demodulator, nearinfraredspectroscopy, brain-computer-interface, signal generator, signal processor, uCLinux

### Eidesstattliche Erklärung

*Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.*

### Statutory Declaration

*I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.*

.....  
Datum / Date

.....  
Peter Lindner

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Motivation und Aufgabenstellung . . . . .	1
1.2	Aufbau dieser Diplomarbeit . . . . .	3
<b>2</b>	<b>Grundlagen</b>	<b>4</b>
2.1	NIRS - Nahinfrarotspektroskopie . . . . .	4
2.2	Lock-in Verstärker . . . . .	9
2.3	Betriebssystem uClinux . . . . .	14
2.4	Bootloader U-Boot . . . . .	16
<b>3</b>	<b>Methode</b>	<b>18</b>
3.1	Hardwarerealisierung . . . . .	18
3.1.1	Prozessormodul . . . . .	20
3.1.2	Signalgenerator . . . . .	25
3.1.3	Messsignalpfad . . . . .	33
3.1.4	Analoger Ausgangssignalpfad . . . . .	40
3.1.5	Spannungsversorgung . . . . .	42
3.1.6	Modulationsplatinen . . . . .	47
3.1.7	Leiterplattendesign . . . . .	48
3.2	Softwarerealisierung . . . . .	51
3.2.1	Kernelspace: Treibermodule für SPORT0 und SPORT1 . . . . .	53
3.2.2	Userspace: Das Hauptprogramm . . . . .	66
3.3	Entwickeln von Software für das Prozessormodul . . . . .	77

---

<b>4</b>	<b>Bedienung</b>	<b>80</b>
4.1	Vorderseite des Verstärkers . . . . .	81
4.2	Rückseite des Verstärkers . . . . .	82
4.3	Bedienung der Software . . . . .	83
<b>5</b>	<b>Ergebnisse</b>	<b>85</b>
5.1	Platinenaufbau . . . . .	85
5.2	Gehäuseaufbau . . . . .	88
5.3	Testmessungen . . . . .	88
<b>6</b>	<b>Diskussion, Zusammenfassung und Ausblick</b>	<b>95</b>
6.1	Diskussion der Ergebnisse . . . . .	95
6.1.1	Signalgenerator . . . . .	95
6.1.2	Spannungsversorgung . . . . .	96
6.1.3	Ausgangssignalpfad . . . . .	97
6.1.4	Software . . . . .	98
6.2	Zusammenfassung und Ausblick . . . . .	99
	<b>Literaturverzeichnis</b>	<b>100</b>
<b>A</b>	<b>Abkürzungsverzeichnis</b>	<b>103</b>
<b>B</b>	<b>Schaltpläne und Layout</b>	<b>104</b>
<b>C</b>	<b>Stückliste</b>	<b>115</b>
C.1	Farnell . . . . .	115
C.2	RS Components . . . . .	117
C.3	Digikey . . . . .	117
C.4	Würth . . . . .	117
C.5	Lasercomponents . . . . .	117
C.6	Hamamatsu . . . . .	117
C.7	Pactec Enclosures . . . . .	118
C.8	Samtec . . . . .	118

# Kapitel 1

## Einleitung

### Inhalt

---

1.1	Motivation und Aufgabenstellung . . . . .	1
1.2	Aufbau dieser Diplomarbeit . . . . .	3

---

### 1.1 Motivation und Aufgabenstellung

Brain Computer Interfaces (BCIs) sind künstlich geschaffene Schnittstellen zwischen dem Gehirn und einem Computer. Sie schaffen, wie Wolpaw et al. in [35] definieren, einen neuen Kommunikationskanal über welchen Kommandos oder Nachrichten, die ein Individuum sendet, abseits der normalen Übertragungswege (periphere Nerven und Muskeln) an die Außenwelt gelangen können. Um solch ein System zu realisieren wird die Gehirnaktivität innerhalb einer bestimmten Region gemessen und an einen Computer übertragen. Nach geeigneter Auswertung der Messdaten ist der Computer in der Lage, z.B. mit zusätzlicher Hard- oder Software, Steuer- oder Kommunikationsaufgaben auszuführen. Denkbar sind hier z.B. Steuerung von Prothesen, Stimulation von Muskelpartien oder Benutzung einer virtuellen Tastatur am Computer [35]. Häufig wird in BCI-Systemen die Gehirnaktivität mit Hilfe der Elektroenzephalografie (EEG) gemessen, da diese, wie in [35] erläutert wird, kurze Reaktionszeiten aufweist sowie nur relativ einfaches und kostengünstiges Equipment erfordert. Neben der Aufzeich-

nung des EEGs existieren noch weitere Methoden die Gehirnaktivität zu bestimmen. Hierbei werden die Magnetoenzephalografie (MEG), die Positronenemissionstomografie (PET), die funktionelle Magnetresonanztomographie (fMRT) sowie andere bildgebende Verfahren verwendet (siehe [35]), welche jedoch den Nachteil erhöhten technischen sowie finanziellen Aufwands mit sich bringen. Eine weitere vielversprechende Methode ist die Nahinfrarotspektroskopie (NIRS), bei der mit Hilfe von Licht die Konzentration von Oxy- bzw. Deoxyhämoglobin (sauerstoffangereichertes bzw. sauerstoffarmes Hämoglobin) bestimmt wird.

Wie im Laufe dieser Arbeit beschrieben wird, müssen bei der Nahinfrarotspektroskopie für die Messung der Konzentration von Oxy- sowie Deoxyhämoglobin zwei Lichtquellen zum Einsatz kommen, dessen Leuchtintensitäten zum Zweck der Unterscheidbarkeit sinusförmig moduliert werden. Das Licht wird über Lichtwellenleiter in den Schädel eingekoppelt. Beim Durchtritt durch das Gehirngewebe werden diese modulierten Lichtsignale unter anderem auch, abhängig von der Konzentration, von Oxy- bzw. Deoxyhämoglobin abgeschwächt. Es entstehen somit amplitudenmodulierte Lichtsignale, welche in der Nähe der Einkoppelstelle wieder aus dem Schädel austreten, gemessen und in Spannungssignale umgewandelt werden. Nun müssen diese Amplitudenmodulierten Signale demoduliert werden, um die eigentliche Information zu extrahieren. Da die Leuchtintensität der beiden Lichtquellen beim Durchtritt durch Knochen und Gewebe stark abgeschwächt und die ursprünglichen Signale zusätzlich von Störsignalen verschiedener Ursachen überlagert werden, muss besonderes Augenmerk auf die Demodulation gelegt werden. Aufgrund der bauartbedingten Sensitivität für bestimmte bekannte Frequenzen eignen sich Lock-in Verstärker hervorragend als Demodulator für die Anwendung in einem Nahinfrarotspektroskopiesystem.

Da jedoch die bisher am Markt verfügbaren Geräte, wie z.B. der von Bauernfeind in seiner Arbeit [12] verwendete Lock-in Verstärker 7265 von Signal Recovery, sehr kostspielig und verhältnismäßig groß sind, wurde im Zuge dieser Diplomarbeit nach einer Lösung gesucht, welche portabler und damit einfacher in einem Nahinfrarotspektroskopiesystem einsetzbar ist. Des Weiteren sollte die neue Lösung einfacher in der Handhabung und kostengünstiger sein. Außerdem sollte der zu entwickelnde Lock-in Verstärker sämtliche Hardware integriert haben, welche zur NIRS-Messung in einem BCI notwendig ist. So sollen auch der Signalgenerator zur Generierung der Referenzsignale, sowie die Messeinheit, welche die Lichtsignale wieder aufnimmt, in einem Gehäuse vereint sein, wobei die Möglichkeit bestehen sollte, bis zu vier Referenzfrequenzen zu



generieren und mit zwei Messeinheiten Lichtintensitäten zu messen. Die gewünschten Referenzfrequenzen sind 2 kHz, 3 kHz, 5 kHz und 7 kHz. Der Eingangsspannungsbereich sollte zwischen 1 mV und 100 mV einstellbar sein. Die Ausgabe der Messwerte sollte einerseits auf analogem Wege, andererseits auch digital erfolgen, um bei der Auswertung flexibel zu sein. Eine weitere Vorgabe war, dass nur die Spannungswerte ausgegeben werden. Die Phasenverschiebung zwischen Mess- und Referenzsignal kann zwar von einem Dual-Phase Lock-in Verstärker gemessen werden, ist jedoch bei der Verwendung in einem BCI nicht relevant und wird aus diesem Grund auch innerhalb dieser Diplomarbeit nicht behandelt. Um die Flexibilität weiter zu steigern, sollte als Recheneinheit ein Digitaler Signalprozessor (DSP) zum Einsatz kommen.

## 1.2 Aufbau dieser Diplomarbeit

Diese Arbeit ist in sechs Kapitel unterteilt

**Kapitel 2** befasst sich mit den Grundlagen, die zum Verständnis hilfreich sind. Es wird darauf eingegangen, wie ein Nahinfrarotspektroskopiesystem funktioniert. Außerdem wird die prinzipielle Arbeitsweise eines Lock-in Verstärkers erläutert.

**Kapitel 3** schließt die Besprechung der verwendeten Methoden ein. Hierzu gehört die Behandlung der Hardwaredimensionierung in Abschnitt 3.1 sowie der Softwareprogrammierung in Abschnitt 3.2. Anschließend wird in Abschnitt 3.3 auf die Vorgehensweise bei der Programmierung von Software für den Lock-in Verstärker näher eingegangen.

**Kapitel 4** gibt eine Übersicht über die Front- und Rückseite des entwickelten Lock-in Verstärkers. Anschließend wird die Bedienung, wie sie sich für den Benutzer ergibt, beschrieben.

**Kapitel 5** beschreibt das Endergebnis dieser Diplomarbeit.

**Kapitel 6** liefert eine Diskussion der Ergebnisse und bietet einen Zukunftsausblick inklusive Erweiterungsmöglichkeiten.

# Kapitel 2

## Grundlagen

### Inhalt

---

<b>2.1</b>	<b>NIRS - Nahinfrarotspektroskopie . . . . .</b>	<b>4</b>
<b>2.2</b>	<b>Lock-in Verstärker . . . . .</b>	<b>9</b>
<b>2.3</b>	<b>Betriebssystem uClinux . . . . .</b>	<b>14</b>
<b>2.4</b>	<b>Bootloader U-Boot . . . . .</b>	<b>16</b>

---

### 2.1 NIRS - Nahinfrarotspektroskopie

Nahinfrarotspektroskopie ist ein spektroskopisches Messverfahren, welches auf der Absorption von Licht im Nahinfrarotbereich basiert. Neben einigen technischen Anwendungen z.B. in der Materialforschung, findet dieses Verfahren auch in der Medizin sehr erfolgreich seine Verwendung. Mittels Nahinfrarotspektroskopie ist es unter anderem möglich, nicht invasiv die Konzentrationsänderung von oxygeniertem sowie deoxygeniertem Hämoglobin im Blut zu bestimmen. Da im Gehirn ein Zusammenhang zwischen Gehirnaktivität und der Durchblutung durch das aktive Areal besteht, kann, wie Bauernfeind in seiner Arbeit [12] beschreibt, mit einem Nahinfrarotspektroskopiesystem durch Messung der Oxyhämoglobinkonzentration, sowie der Deoxyhämoglobinkonzentration im Gehirn, auch ein Brain Computer Interface (BCI) realisiert werden.

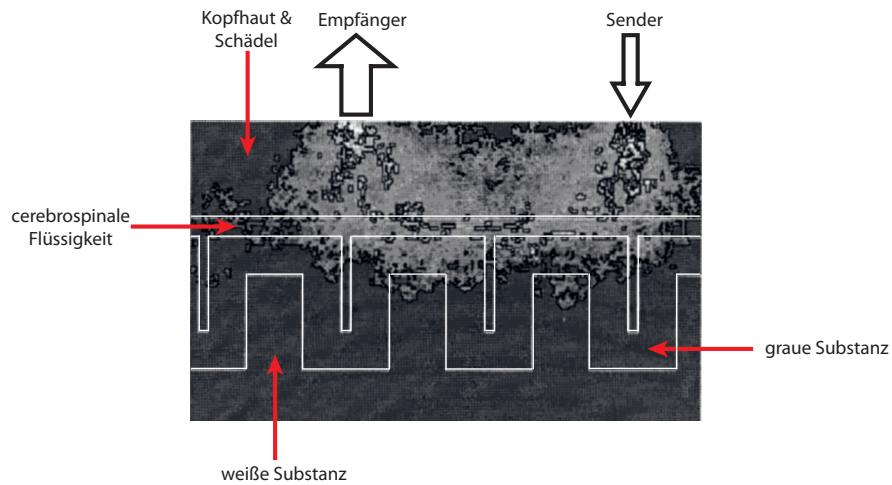
Der grundlegende Effekt, welcher in einem NIRS System ausgenutzt wird, ist der, dass Licht beim Durchtritt durch ein Medium (z.B. Gewebe) entweder gestreut oder absorbiert wird, oder eben dieses Gewebe ungehindert passieren kann. Dies bedeutet, dass Licht, welches mit einer bestimmten Intensität in den Kopf eingestrahlt wird, einen zufälligen Weg durch dieses Gewebe nimmt. Einige der Photonen treten Aufgrund mehrfacher Streuung wieder aus dem Kopf aus und können somit in kurzer Entfernung vom Sender gemessen werden. Die Austrittsintensität hängt von der Einstrahlstärke und von den Gewebeeigenschaften, aber auch vom Abstand von Sender und Empfänger ab [16]. So ist es möglich, mit Hilfe des modifizierten Lambert-Beer'schen Gesetzes die Konzentrationsänderung  $A$  des gewünschten Chromophors (z.B. Oxy- oder Deoxyhämoglobin) zu bestimmen [12].

$$A = \log \left( \frac{I_0}{I_1} \right) = \alpha \cdot c \cdot l \cdot x + K \quad (2.1)$$

Hierbei entspricht  $I_0$  der Intensität des eingestrahlt Lichtes und  $I_1$  die Intensität des austretenden Lichtes.  $\alpha$  stellt den dekadischen molaren Extinktionskoeffizienten,  $c$  die Konzentration des Absorbers,  $l$  den Abstand von Sender und Empfänger und  $x$  den differentiellen Weglängenfaktor (DPF) dar. Zusätzlich werden über  $K$  noch Gewebeerluste berücksichtigt. Okada et al. untersuchten in ihrer Studie [26] den Ausbreitungsweg, den eingestrahlt Nahinfrarotlicht im Kopf eines erwachsenen Menschen nimmt. Der DPF ist ein alters-, gewebs- und wellenlängenabhängiger Faktor, welcher ein Maß für die mittlere Weglänge ist, welche ein Photon durch Streuung innerhalb eines Gewebes nimmt. Wie Bauernfeind in [12] zusammenfasst, beschreiben Okada et al. den Zusammenhang zwischen Eindringtiefe, DPF und dem Abstand zwischen Sender und Empfänger wie folgt:

- Für einen konstanten DPF muss der Abstand von Sender und Empfänger mindestens 25 mm betragen.
- Nur bei einem Abstand von ca. 20-60 mm erreichen die Photonen den Kortex (Eindringtiefe ca. 25 mm). Die maximale Eindringtiefe liegt bei der Hälfte des Abstands zwischen Sender und Empfänger. Bei Abständen  $< 20$  mm erreichen die Photonen den Kortex nicht.
- Eine Messung der weißen Substanz ist auch bei Abständen  $> 60$  mm nicht möglich

Die Studie von Okada et al. verdeutlicht außerdem den Ausbreitungsweg des Lichtes, welcher stets bogenförmig verläuft (siehe Abbildung 2.1).

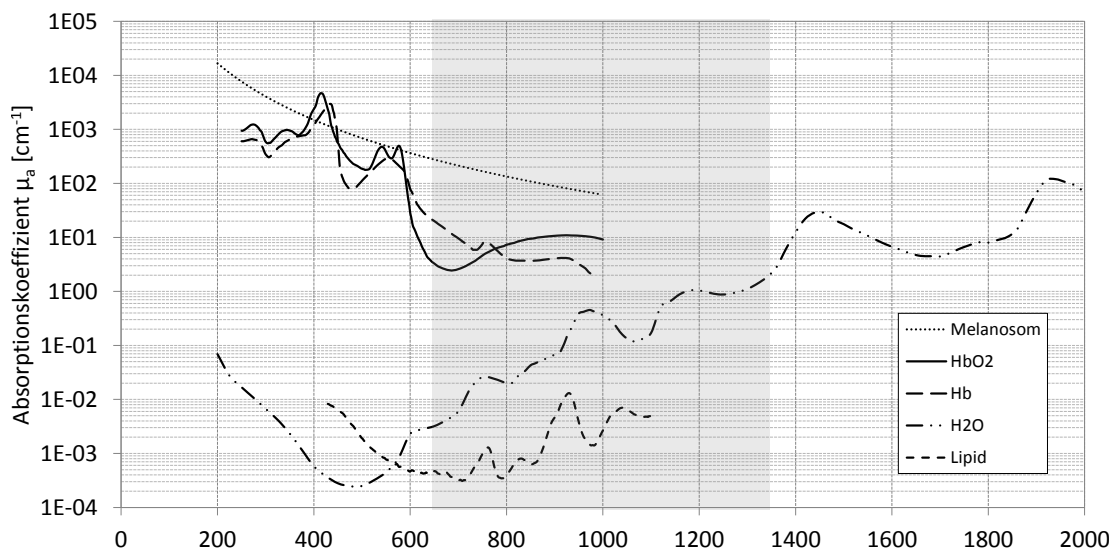


**Abbildung 2.1:** Bogenförmige Ausbreitung der Photonen in einer Gehirnsimulation (entnommen und modifiziert aus [26]). Der Abstand zwischen Sender und Empfänger beträgt hier 30 mm

Für die Realisierung eines Nahinfrarotspektroskopiesystems bieten sich, wie Bauernfeind in [12] näher ausführt, drei unterschiedliche Methoden an. Der zeitaufgelöste Ansatz (time of flight), der frequenz aufgelöste Ansatz sowie der Continuous-Wave-Ansatz. Jeder dieser Ansätze besitzt seine Vor- und Nachteile. Jedoch wird der letztgenannte Ansatz aufgrund seiner relativ leichten und kostengünstigen technischen Realisierbarkeit am häufigsten in NIRS-Systemen eingesetzt. Hierbei wird das in das zu beobachtende Gewebe eingestrahlte Licht mittels einer Photodiode wieder aufgezeichnet, wobei, im Gegensatz zu den anderen beiden Methoden, die Lichtintensität den einzigen gemessenen Parameter darstellt. Da mit diesem Ansatz der DPF nicht genau gemessen werden kann, ist, wie Bauernfeind beschreibt, die Quantifizierung der einzelnen Absorberkonzentrationen ungenau, eine Detektion von Konzentrationsänderungen ist jedoch genau feststellbar, was für den Einsatz eines NIRS in einem BCI System ausreichend ist. Beim Continuous-Wave-Ansatz werden so viele Lichtquellen unterschiedlicher Wellenlänge benötigt, wie Chromophore differenziert werden sollen. Zur näheren Erläuterung der beiden anderen Ansätze sei an dieser Stelle auf die Arbeit von Bauernfeind verwiesen [12].

Nun stellt sich noch die Frage nach der zu verwendenden Wellenlänge des Lichts. In Geweben befinden sich verschiedenste Absorber, welche alle ein spezifisches, wellenlängenabhängiges Absorptionsspektrum besitzen. Die wichtigsten Absorberarten sind Melanin, Hämoglobin, Lipide und Wasser. Abbildung 2.2 zeigt die

Absorptionsspektren der genannten Absorber in Abhängigkeit von der Wellenlänge des eingestrahlten Lichts. Es ist zu erkennen, dass ein sogenanntes optisches Fenster existiert, welches von ca. 630 nm bis ca. 1350 nm reicht. Dieses Fenster wird nach unten hin durch die Absorption von Licht durch das Hämoglobin in der Haut und nach oben hin durch die Absorption, verursacht durch Wasser im Gewebe, begrenzt. Des Weiteren zeigt Abbildung 2.2, dass nur das Absorptionsspektrum von Melanin über dem Absorptionsspektrum von Oxy- und Deoxyhämoglobin liegt, was jedoch kein Problem darstellt, da die Melaninkonzentration während einer Messung konstant bleibt und somit die Messung nicht beeinflusst. Für die Wahl der Wellenlänge des einzustrahlenden Lichts gilt nun, dass diese innerhalb des besprochenen optischen Fensters liegen muss und die Lichtquelle ein möglichst schmalbandiges Emissionsspektrum besitzen muss. Befinden sich mehrere Absorber (z.B. Oxy- und Deoxyhämoglobin) im beobachteten Gewebe, so ergibt sich die Gesamttextinktion aus der Summe der Einzelextinktionen.



**Abbildung 2.2:** Absorptionsspektren der für ein NIRS System entscheidenden Chromophore (Datenpunkte entnommen aus [28])

Das modifizierte Lambert-Beer'sche Gesetz aus Gleichung 2.1 ändert sich nun zu

$$A = \log \left( \frac{I_0}{I_1} \right) = (\alpha_{HbO_2} \cdot c_{HbO_2} + \alpha_{Hb} \cdot c_{Hb}) \cdot l \cdot x + K \quad (2.2)$$

Wobei hier  $\alpha_{HbO_2}$  und  $c_{HbO_2}$  den Extinktionskoeffizienten und die Konzentration

von Oxyhämoglobin und  $\alpha_{Hb}$  sowie  $c_{Hb}$  den Extinktionskoeffizienten und die Konzentration von Deoxyhämoglobin bezeichnen.

Durch Betrachtung von Gleichung 2.2 wird somit klar, dass für die Bestimmung der beiden unbekannt Konzentrationen  $c_{HbO_2}$  und  $c_{Hb}$  ein Gleichungssystem zweiter Ordnung notwendig ist. Aus diesem Grund wird eine zweite Lichtquelle unterschiedlicher Wellenlänge und, um eine gegenseitige Beeinflussung zu verhindern, einem unterschiedlichen Emissionsspektrum, benötigt, um eine zweite Gleichung zu erhalten. Wie Bauernfeind in [12] erläutert, erhält man somit durch Differenzierung und geeignete Umformung von Gleichung 2.2 für eine einzelne Wellenlänge die Gleichung 2.3

$$\frac{\Delta A_\lambda}{x_\lambda \cdot l} = (\alpha_{HbO_2} \cdot \Delta c_{HbO_2} + \alpha_{Hb} \cdot \Delta c_{Hb}) \quad (2.3)$$

Wobei  $\Delta A_\lambda$  der Differenz der Extinktion der Leuchtintensität einer Wellenlänge zu zwei verschiedenen Messzeitpunkten entspricht. Für zwei Wellenlängen ergibt sich somit das Gleichungssystem 2.4 und 2.5 mit Hilfe dessen die zwei gewünschten Konzentrationsänderungen berechnen lassen.

$$\frac{\Delta A_{\lambda_1}}{x_{\lambda_1} \cdot l} = (\alpha_{\lambda_1, HbO_2} \cdot \Delta c_{HbO_2} + \alpha_{\lambda_1, Hb} \cdot \Delta c_{Hb}) \quad (2.4)$$

$$\frac{\Delta A_{\lambda_2}}{x_{\lambda_2} \cdot l} = (\alpha_{\lambda_2, HbO_2} \cdot \Delta c_{HbO_2} + \alpha_{\lambda_2, Hb} \cdot \Delta c_{Hb}) \quad (2.5)$$

Bauernfeind wählt in seiner Arbeit [12] zwei Leuchtdioden (LEDs), welche Licht mit den Wellenlängen  $690 \pm 20$  nm und  $830 \pm 20$  nm emittieren. Diese LEDs werden auch in dieser Masterarbeit zum Einsatz kommen.

Abbildung 2.3 zeigt den schematischen Aufbau eines Nahinfrarotspektroskopiesystems. Zu erkennen sind hier die zwei LEDs unterschiedlicher Wellenlänge mit Hilfe derer Licht in den Schädel eingekoppelt wird. Die Lichtintensität des vom Schädel wieder austretenden Lichts enthält nun Anteile von beiden Leuchtdioden. Sie kann über eine Lawinenfotodiode (APD) gemessen und in eine zur Intensität proportionale Spannung umgewandelt werden. Das Problem besteht nun darin, die Intensitätsanteile der beiden LEDs wieder voneinander zu trennen. Hierbei hilft ein Signalgenerator, welcher den Strom, der durch die LEDs fließt und damit die Leuchtintensität, die in den Kopf eingestrahlt wird, mit einer bestimmten Frequenz (Trägerfrequenz) sinusförmig moduliert. Hierbei muss darauf geachtet werden, dass die Trägerfrequenz für jede LED

unterschiedlich und unabhängig von den anderen Trägerfrequenzen ist. Durch die Absorption im Kopf wird nun die Intensität dieses modulierten Lichtes verändert. Damit ergibt sich am Austrittsort des Lichtes ein Signal mit amplitudenmodulierter Intensität welches über die Lawinenfotodiode aufgenommen wird. Schlussendlich können mit einem geeigneten Demodulator (z.B. Lock-in Verstärker) die Signalanteile jeder LED aus der Ausgangsspannung der Fotodiode extrahiert und einem PC zur Berechnung der Konzentrationen von Oxy- und Deoxyhämoglobin zugeführt werden.

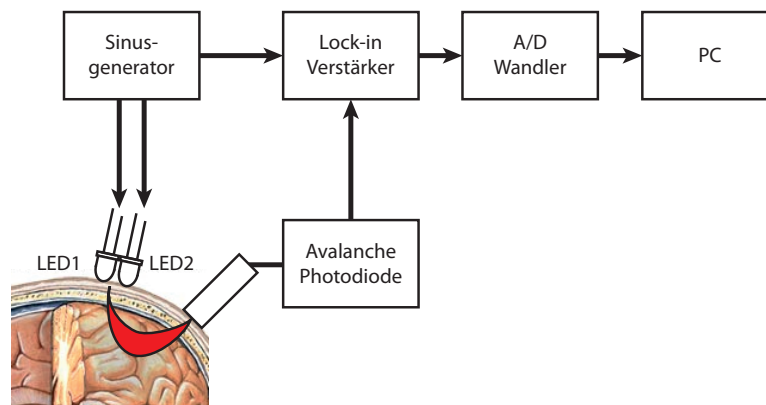


Abbildung 2.3: Blockschaltbild eines Zweikanal-NIRS Systems

## 2.2 Lock-in Verstärker

Im Folgenden soll nun die Funktionsweise eines Lock-in Verstärkers näher erläutert werden. Wie de Marcellis und Ferri in [25] beschreiben, ist ein Lock-in Verstärker ein äußerst mächtiges Werkzeug um Amplitudenmodulierte Signale zu messen, selbst wenn diese von Rauschsignalen überdeckt werden. Diese Rauschsignale gelangen im NIRS-Kontext durch verschiedenste Quellen wie z.B. Einstreuungen von Tageslicht, Computerbildschirmen oder Leuchtstoffröhren in das optische System des Messaufbaus und verschlechtern das SNR des Messsignals drastisch. Auch physiologische Effekte wie z.B. Pulsschlag oder Atmung beeinflussen das Messsignal. Das Kernstück eines Lock-in Verstärkers ist ein phasensensitiver Detektor, welcher mathematisch gesehen eine simple Multiplikation zweier Signale darstellt. Wie in Abbildung 2.5 zu sehen ist, benötigt jeder Lock-in Verstärker eine Referenzspannung mit der selben Frequenz wie die zu messende Spannung. Diese Referenzspannung kann intern generiert und nach außen geführt, oder auch extern zugeführt werden.

Angenommen, es existieren zwei sinusförmige Signale,  $R(t)$  und  $M(t)$ , wobei  $R(t)$  ein Referenzsignal und  $M(t)$  ein gemessenes Signal darstellt. Beide Signale besitzen eine bestimmte Frequenz ( $\omega_R, \omega_M$ ), sowie eine Amplitude ( $A_R, A_M$ ). Das gemessene Signal weist einen bestimmten Phasenbezug  $\varphi$  zum Referenzsignal auf.

$$\begin{aligned} R(t) &= A_R \cdot \sin(\omega_R t) \\ M(t) &= A_M \cdot \sin(\omega_M t + \varphi) \end{aligned} \quad (2.6)$$

Werden diese beiden Signale multipliziert, so ergibt sich:

$$\begin{aligned} U_{PSD}(t) &= R(t) \cdot M(t) = A_R \cdot A_M \cdot \sin(\omega_M t + \varphi) \cdot \sin(\omega_R t) = \\ &= \frac{A_R A_M}{2} \cdot [\cos(\omega_M t + \varphi - \omega_R t) - \cos(\omega_M t + \varphi + \omega_R t)] = \\ &= \frac{A_R A_M}{2} \cdot [\cos((\omega_M - \omega_R)t + \varphi) - \cos((\omega_M + \omega_R)t + \varphi)] \end{aligned} \quad (2.7)$$

Bei näherer Betrachtung dieses Ergebnisses erkennt man, dass sich, falls die Signalfrequenz von Referenzsignal und Messsignal gleich groß sind ( $\omega_R = \omega_M = \omega$ ), folgender Term ergibt:

$$U_{PSD1}(t) = \frac{A_R A_M}{2} \cdot [\cos(\varphi) - \cos(2\omega t + \varphi)] \quad (2.8)$$

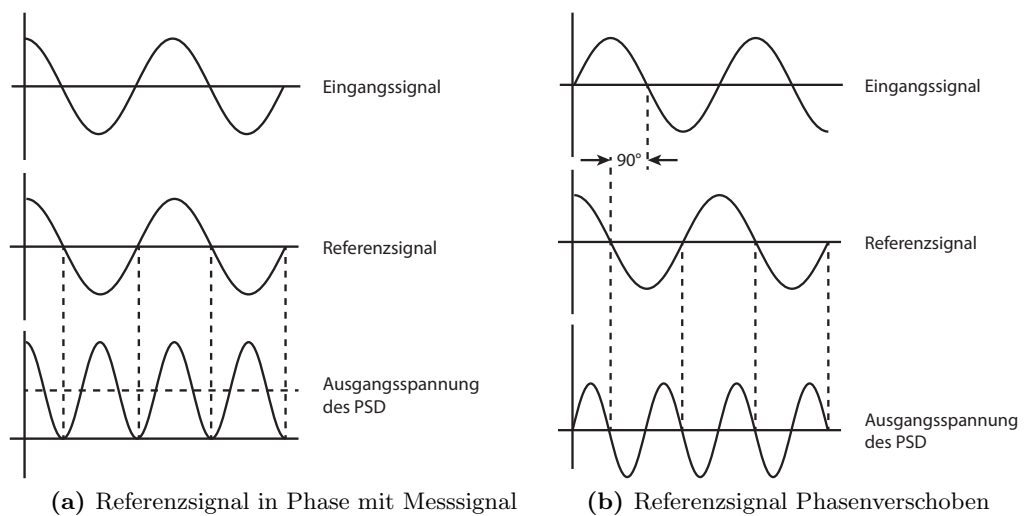
Dies bedeutet nun, dass sich am Ausgang des phasensensitiven Detektors eine sinusförmige Spannung einstellt, dessen Frequenz der doppelten Signal- bzw. Referenzfrequenz entspricht. Die Ausgangsspannung besitzt jedoch einen Gleichspannungsanteil ( $\cos(\varphi)$ ), dessen Größe vom Phasenwinkel  $\varphi$  abhängt, d.h. das Ausgangssignal wird maximal, wenn die Phasenverschiebung gleich null ist. Abbildung 2.4 zeigt diesen Zusammenhang.

Somit kann nun zusammenfassend festgehalten werden, dass sich, wenn die Amplitude des Referenzsignals konstant ist, das Ausgangssignal des PSD wie folgt einstellt [29]:

- Es ist mit  $2\omega t$  moduliert
- Es ist proportional zur Signalamplitude  $A_M$
- Es ist proportional zum Phasenwinkel zwischen Referenz- und Messsignal

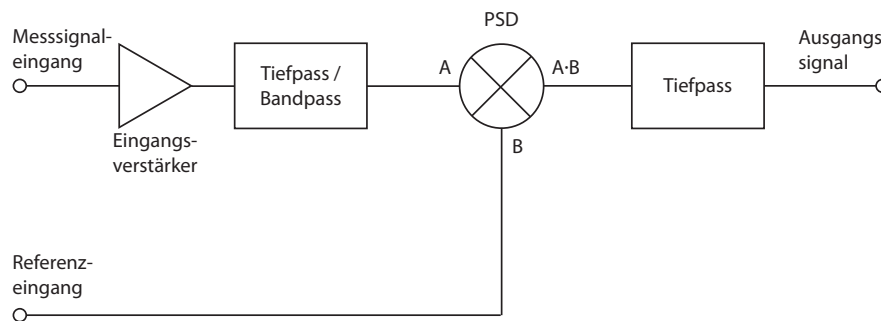
Bevor die Ausgangsspannung ausgegeben wird, wird mittels eines Tiefpassfilters der Wechselspannungsanteil weggefiltert, sodass nur mehr der Gleichspannungsanteil übrig





**Abbildung 2.4:** Einfluss des Phasenwinkels auf das Ausgangssignal des phasensensitiven Detektors

bleibt. Somit arbeitet ein Lock-in Verstärker wie ein sehr steiler Bandpassfilter dessen Mittenfrequenz der Referenzfrequenz entspricht. Das SNR ist proportional zur Zeitkonstante des Tiefpassfilters. Abbildung 2.5 verdeutlicht nocheinmal grafisch die Funktionsweise eines Single-Phase Lock-in Verstärkers.



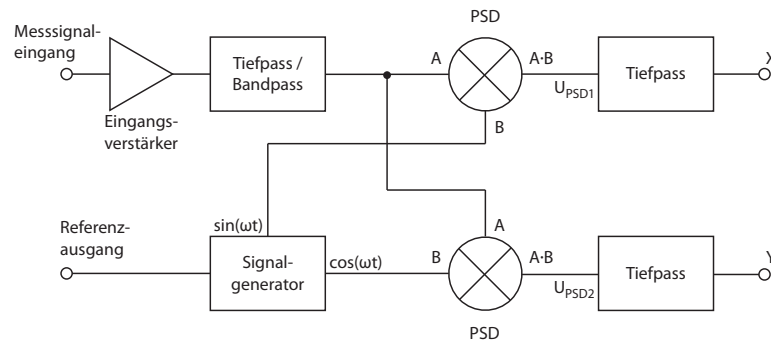
**Abbildung 2.5:** Blockschaltbild eines Single-Phase Lock-in Verstärkers (entnommen und modifiziert aus [29])

Bisher wurde nur ein rauschfreies Signal betrachtet. Rauschsignale, also Signale ohne fixen Frequenz- bzw. Phasenbezug zum Referenzsignal, werden ebenfalls mit dem Referenzsignal multipliziert und modulieren somit den Gleichspannungsanteil. Jedoch bei geeigneter Wahl des Tiefpassfilters werden diese Signalanteile ebenfalls weggefiltert und am Ausgang des Lock-in Verstärkers kann der gewünschte Gleichspannungsanteil

gemessen werden [29].

Aufgrund der Abhängigkeit des Ausgangssignals von der Phasenverschiebung, muss vor jeder Messung mittels des Phasenschiebers diese Phasenverschiebung ausgeglichen werden, um das Ausgangssignal zu maximieren. Ändert sich die Phasenverschiebung, so muss wiederum abgeglichen werden. Um dieses Abgleichen zu Beginn und während der Messung zu vermeiden, wird das Prinzip des Dual-Phase Lock-in Verstärkers angewendet, dessen Funktionsweise in Abbildung 2.6 dargestellt ist. Wie der Single-Phase Lock-in Verstärker benötigt auch der Dual-Phase Lock-in Verstärker ein Referenzsignal. Idealerweise produziert der Verstärker dieses Referenzsignal selbst und stellt diese dem internen Algorithmus sowie dem externen Messaufbau zur Verfügung. Des Weiteren beinhaltet ein solcher Verstärker zwei PSDs, welche das Messsignal einmal mit dem Referenzsignal ( $U_{R1}$ ) und einmal mit dem um  $90^\circ$  phasenverschobenen Referenzsignal ( $U_{R2}$ ) multiplizieren.

$$\begin{aligned} U_{R1}(t) &= A_R \cdot \sin(\omega_{R1}t) \\ U_{R2}(t) &= A_R \cdot \cos(\omega_{R2}t) \\ U_M(t) &= A_M \cdot \sin(\omega_Mt + \varphi) \end{aligned} \quad (2.9)$$



**Abbildung 2.6:** Blockschaltbild eines Dual-Phase Lock-in Verstärkers

Das Ergebnis der Multiplikation von  $U_{R1}(t)$  mit  $U_M(t)$  ist dasselbe wie bei der Multiplikation, welche der PSD im Single-Phase Lock-in Verstärker durchführt, und ist Gleichung 2.8 zu entnehmen. Die zweite Multiplikation geschieht wie nachfolgend

erklärt.

$$\begin{aligned}
 U_{PSD2}(t) &= U_{R2}(t) \cdot M(t) = A_R \cdot A_M \cdot \sin(\omega_M t + \varphi) \cdot \cos(\omega_{R2} t) = \\
 &= \frac{A_R A_M}{2} \cdot [\sin(\omega_M t + \varphi - \omega_{R2} t) + \sin(\omega_M t + \varphi + \omega_{R2} t)] = \quad (2.10) \\
 &= \frac{A_R A_M}{2} \cdot [\sin((\omega_M - \omega_{R2})t + \varphi) + \sin((\omega_M + \omega_{R2})t + \varphi)]
 \end{aligned}$$

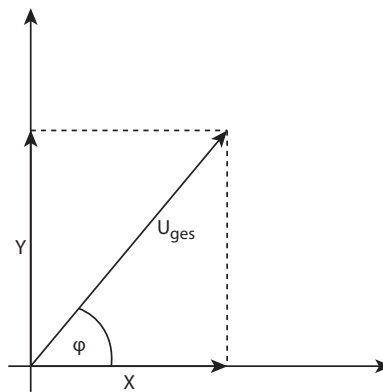
Wenn nun die Signalfrequenz des Messsignals und die des Referenzsignals gleich groß ist ( $\omega_{R1} = \omega_{R2} = \omega_M = \omega$ ), so ändert sich das Ergebnis der Gleichung 2.10 wie folgt:

$$U_{PSD2}(t) = \frac{A_R A_M}{2} \cdot [\sin(\varphi) + \sin(2\omega t + \varphi)] \quad (2.11)$$

Im Anschluss an die Multiplikation werden  $U_{PSD1}(t)$  und  $U_{PSD2}(t)$  tiefpassgefiltert (siehe Abbildung 2.6). Damit fallen die  $2\omega t$  Frequenzanteile weg und es ergeben sich die beiden Ausgangssignale X und Y wie sie in Gleichung 2.12 dargestellt sind.

$$\begin{aligned}
 X &= \frac{A_R A_M}{2} \cdot \cos(\varphi) \\
 Y &= \frac{A_R A_M}{2} \cdot \sin(\varphi)
 \end{aligned} \quad (2.12)$$

X und Y können nun, wie Burdett in [29] beschreibt, als zwei Vektoren angesehen werden, welche orthogonal zueinander stehen. Wie mit Hilfe von Abbildung 2.7 nachzuvollziehen ist, können in weiterer Folge aus diesen zwei Komponenten der Betrag und der Phasenwinkel des Gesamtausgangssignales berechnet werden (siehe Gleichungen 2.13 und 2.14).



**Abbildung 2.7:** Die zwei Ausgangsspannungen des Dual-Phase Lock-in Verstärkers als Vektoren interpretiert

$$A_{Uges} = \sqrt{X^2 + Y^2} = \sqrt{\frac{A_R^2 A_M^2}{2} \cos^2(\varphi) + \frac{A_R^2 A_M^2}{2} \sin^2(\varphi)} = \frac{A_R A_M}{2} \quad (2.13)$$

$$\varphi = \arctan\left(\frac{Y}{X}\right) = \arctan\left(\frac{\frac{A_R A_M}{2} \sin(\varphi)}{\frac{A_R A_M}{2} \cos(\varphi)}\right) = \arctan\left(\frac{\sin(\varphi)}{\cos(\varphi)}\right) \quad (2.14)$$

Somit fällt, wie in Gleichung 2.13 sehr gut zu erkennen ist, der beim Single-Phase Lock-in Verstärker besprochene Phasenbezug und damit das Ausgleichen der Phasenverschiebung zur Maximierung des Ausgangssignals weg. Dies ist einer der Gründe, weshalb Dual-Phase Lock-in Verstärker die meist verbreitetste Art an Lock-in Verstärkern darstellen [29].

## 2.3 Betriebssystem uClinux

**Allgemeines:** uClinux ist eine Linuxdistribution, welche speziell für Systeme ohne Memory-Management-Unit (MMU) angepasst wurde. Die Vorteile dieser Distribution liegen einerseits in der großen Stabilität, andererseits in ihrer Flexibilität und Portabilität. So wurde uClinux auf diverse Architekturen wie z.B. ARM, Freescale m68k, Intel i960, Xilinx MicroBlaze und Analog Devices Blackfin [33] portiert und läuft aus diesem Grund mittlerweile schon auf einer ganzen Reihe von Produkten wie Routern, WebCams, Telefonen und DVD-Playern. Auch die Tatsache, dass unter uClinux die komplette Linux API zur Verfügung steht, dass diese Distribution schon von Haus aus eine Vielzahl von Treibern für diverseste Hardware mitbringt, aber auch dass uClinux komplett quelloffen und damit kostenlos ist, sprechen für eine Verwendung von uClinux. Dem gegenüber stehen jedoch einige Schwächen bzw. Limitierungen von uClinux. So findet sich z.B. sehr viel weniger Onlinedokumentation im Internet wieder, als bei herkömmlichen Linuxdistributionen. Auch die Community ist bei uClinux verbreitungsbedingt kleiner. Des Weiteren ist es möglich, dass, durch das Fehlen von MMU und Memory-Protection-Unit (MPU), Userspaceprozesse direkt auf den Kernelspace Speicher zugreifen können.

**uClinux auf dem Blackfin DSP:** Für einige Prozessoren der Blackfin Prozessorfamilie von Analog Devices, so auch für den in dieser Arbeit eingesetzten digitalen Signalprozessor ADSP-BF527 [1], gibt es einen Port von uClinux, der von einer durchaus aktiven Community betreut wird. Auch Analog Devices hat die große Nachfrage

erkannt und fördert offiziell das Open Source Projekt mit Foren [2], [10] und einer Wiki-Seite [3]. Blackfin uClinux unterstützt mittlerweile sämtliche Hardware dieser Prozessoren. Zusätzlich existieren schon eine Menge an Treibern für externe Bauelemente, wie z.B. Audiocodecs, digitale Potentiometer, Digital-Analog-Wandler oder Analog-Digital-Wandler, welche je nach Bedarf in der Kernelkonfiguration aktiviert oder deaktiviert werden können. Dies bedeutet, dass im Regelfall eine vom Entwickler speziell auf die verwendete Hardware angepasste Kernelkonfiguration zum Einsatz kommt, in welcher lediglich diejenigen Komponenten aktiviert sind, welche auch wirklich benötigt werden. Dies reduziert die Größe des Betriebssystemimages auf ein Minimum.

**Aufbau einer Entwicklungsumgebung für uClinux:** Damit der Entwickler Software für uClinux erstellen kann, muss zuerst eine Entwicklungsumgebung aufgebaut werden, welche, wie Brinker et al. in [14] beschreiben, prinzipiell aus zwei Hauptkomponenten besteht, dem Hostsystem, worauf die Programme geschrieben und kompiliert werden, und dem Targetsystem, auf dem das Programm ausgeführt wird. Diese beiden Systeme sind per UART und/oder Ethernet miteinander verbunden. Das Hostsystem ist ein PC auf welchem idealerweise ebenfalls ein Linuxsystem läuft, weil viele der benötigten Tools schon in einem Standardlinuxsystem vorhanden sind. Um Sourcecode zu schreiben ist prinzipiell lediglich ein einfacher Editor, wie z.B. emacs oder gedit, notwendig. Allerdings muss nach dem Schreiben des Codes ebendieser in ein für das Targetsystem ausführbares Programm umgewandelt werden. Dies geschieht mit Hilfe eines sogenannten Crosscompilers, welcher zusätzlich installiert werden muss. Für Blackfin Prozessoren ist dies die Blackfin-Toolchain. Zum Debuggen der Software auf dem Targetsystem empfiehlt sich die Verwendung eines Debuggers, wie z.B. GDB, in Verbindung mit einem JTAG-Adapter (z.B. gnICE). Um dem Entwickler die Arbeit zu vereinfachen, wird in [14] empfohlen, integrierte Softwareumgebungen, wie Eclipse, welche diese drei (und noch mehr) Tools unter einer grafischen Oberfläche vereinen zu verwenden. Um die Software auf das Targetsystem zu kopieren, bzw. zu installieren, werden Tools wie ftp oder scp dringend benötigt. Zum Compilieren des uClinux Kernels an sich, werden der GNU-Compiler GCC sowie weitere Tools benötigt (für genauere Anleitungen siehe[3]). Des Weiteren ist eine Installation eines TFTP-Servers sehr hilfreich. Somit kann sich der Bootloader des Targetsystems das kompilierte uClinux Imagefile automatisch vom Hostsystem laden und dieses anschließend booten [14].

**Alternativen zu uClinux für den Blackfin DSP:** Alternativ zu uClinux können noch einige andere Betriebssysteme auf Blackfin Prozessoren genutzt werden [34]. Die wohl schärfste Konkurrenz bildet der VisualDSP++ Kernel (VDK). Dies ist ein real-time Kernel von Analog Devices welcher in Verbindung mit Analog Devices' Entwicklungsumgebung VisualDSP++ ausgeliefert wird. Diese Entwicklungsumgebung inkl. Debugger ist jedoch sehr kostspielig, wodurch sie für eine Prototypenentwicklung, wie sie in dieser Arbeit stattfindet, nicht in Frage kommt. Mit  $\mu C/OS - II$ , ThreadX, Nucleus oder INTEGRITY stehen noch weitere Alternativen zur Verfügung, welche jedoch allesamt entweder sehr kostspielig sind, mangelnde Hardwareunterstützung aufweisen, unflexibel sind, oder schlicht und einfach wenig Dokumentation und Hilfestellungen bieten. Aus diesen Gründen wurde bei der Entwicklung des Lock-in Verstärkers auf uClinux als Betriebssystem zurückgegriffen.

## 2.4 Bootloader U-Boot

Der Bootloader ist ein kleines Stück Software, welches sofort nach dem Start der Hardware zu laufen beginnt. Seine Hauptaufgabe besteht darin, das uClinux Image zu laden, zu entpacken und anschließend auszuführen. Ohne einen Bootloader kann uClinux nicht booten. Der erste Schritt, nachdem der Bootloader aktiv geworden ist, besteht darin, lowlevel-Initialisierungen durchzuführen. So werden gleich zu Beginn der Prozessor (Clock, PLL, interne Register, etc.) sowie der Speichercontroller initialisiert. An dieser Stelle können auch weitere Geräte und Schnittstellen durch den Bootloader initialisiert werden, was jedoch im Normalfall nicht nötig ist, weil dies vom Linux Kernel übernommen wird. Anschließend überprüft der Bootloader diverse Bootoptionen, wie z.B. Speicherort des uClinux-Images oder die Eingabe über eine externe Konsole, lädt das uClinux Image vom Flashspeicher in den RAM, entpackt es und startet es nach der Übergabe von Bootparametern [14]. Zusätzlich zu diesen Aufgaben können, wie Brune in [15] aufzählt, die meisten Bootloader zusätzliche Aufgaben, wie Lesen und Schreiben beliebiger Speicherzellen, Laden von neuen Binärimages in den RAM der Hardware über UART bzw. Ethernet sowie Kopieren von Binärimages vom RAM in den Flashspeicher der Hardware, durchführen. Ein sehr beliebter, weil quelloffener und somit kostenloser, Bootloader ist „Das U-Boot“. U-Boot wurde mittlerweile schon auf hunderten von embedded-Boards portiert und wird von einer großen Community betreut. So unterstützt dieser Bootloader auch den in diesem Projekt verwendeten BF527

---

reibunglos. Analog Devices hat sogar ein eigenes Forum [2] zur Betreuung von U-Boot auf Blackfin Signalprozessoren eingerichtet. U-Boot besitzt den großen Vorteil, sehr flexibel konfigurierbar zu sein. So hat der Entwickler die Möglichkeit, einen optimalen Konsens zwischen Funktionalität und Größe des binären Images zu erreichen. Es gibt jedoch auch schon vorkonfigurierte und vorkompilierte Binärimages, welche nur heruntergeladen und in den Flashspeicher kopiert werden müssen. Dies geschieht, falls noch kein Bootloader auf dem Targetsystem installiert ist, am einfachsten mit Hilfe eines JTAG-Adapters und, im Falle eines Blackfin Signalprozessors, mit Hilfe der Blackfin Toolchain. Ist z.B. schon U-Boot auf dem Targetsystem installiert, so kann sich U-Boot selbst flashen. In diesem Fall ist kein JTAG Adapter notwendig.

# Kapitel 3

## Methode

### Inhalt

---

<b>3.1</b>	<b>Hardwarerealisierung</b>	<b>18</b>
<b>3.2</b>	<b>Softwarerealisierung</b>	<b>51</b>
<b>3.3</b>	<b>Entwickeln von Software für das Prozessormodul</b>	<b>77</b>

---

### 3.1 Hardwarerealisierung

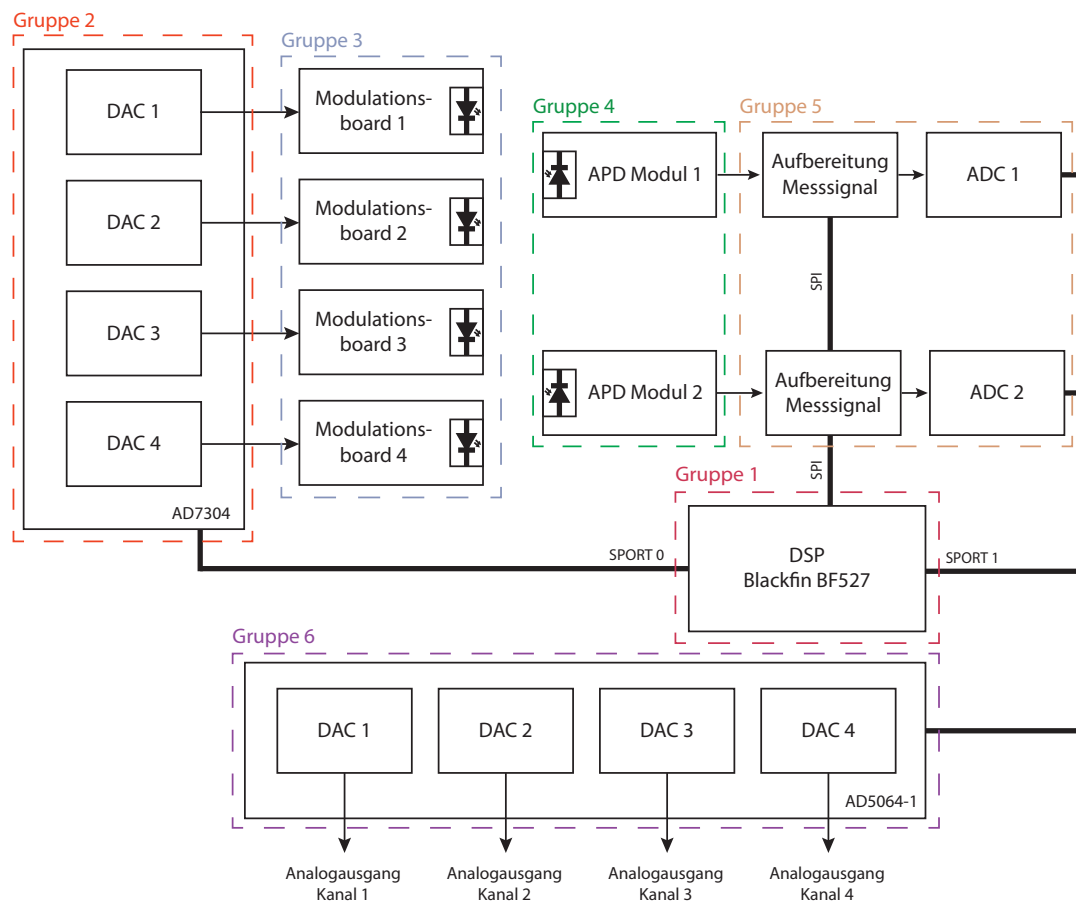
Im Wesentlichen kann die Hardware des im Zuge dieser Diplomarbeit entwickelten Lock-in Verstärkers in sechs Baugruppen unterteilt werden. In Abbildung 3.1 wird diese Aufteilung verdeutlicht. Herzstück des Verstärkers ist der Blackfin Signalprozessor ADSP-BF527 [1] von Analog Devices. Seine Aufgaben bestehen einerseits darin die Sinussignale zu generieren, andererseits die Messsignale aufzunehmen und weiterzuarbeiten (Baugruppe 1).

Die zweite Baugruppe stellen die Bauelemente für die Sinusausgabe dar. Mit Hilfe des vierfach-DACs AD7304 von Analog Devices [6] können hiermit bis zu vier, per Software in der Frequenz einstellbare, Sinussignale erzeugt werden.

Diese Sinussignale werden dazu verwendet, mittels aufsteckbarer Modulationsplatten (Gruppe 2) die Intensitäten der Nahinfrarot LEDs zu modulieren.



Die dritte Baugruppe wandelt die per Lichtwellenleiter aufgenommene Lichtintensität in eine proportionale Spannung um. Als Fotodetektor wurde im Zuge dieses Masterprojekts das selbe APD-Modul gewählt, das auch Bauernfeind [12] in seiner Arbeit zur Detektion der NIRS-Signale verwendet. Es handelt sich hierbei um ein Modul C5460-01 von Hamamatsu (Hamamatsu Photonics K.K., Solid State Division, Japan). Der große Vorteil dieses Produktes liegt in seiner Integriertheit. Es vereint Photodiode, Strom-Spannungs-Wandler und Temperaturkompensationsschaltung in einem kompakten und relativ leicht integrierbaren Modul. Für genauere technische Daten sei an dieser Stelle auf das Datenblatt verwiesen [20].



**Abbildung 3.1:** Blockschaltbild des entwickelten Lock-in Verstärkers

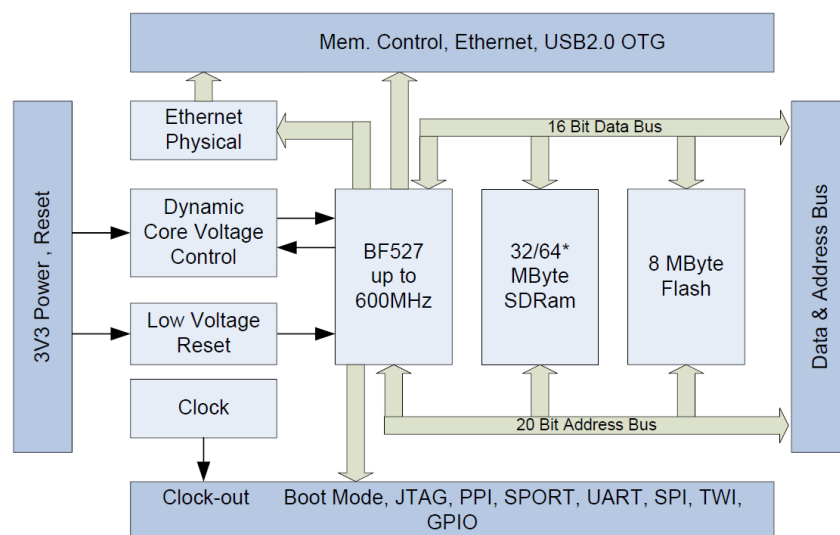
Das Spannungssignal, welches das APD-Modul ausgibt, durchwandert nun die nächste Baugruppe, den Messsignalpfad (Baugruppe 4). Hier werden zu Beginn diverse Signalanpassungen, wie Entfernen des Gleichspannungsanteils oder Signalverstärkung,

durchgeführt. Anschließend folgt in Baugruppe 5 die für die Analog-Digital-Wandlung obligate Antialiasingfilterung mit Hilfe eines aktiven Tiefpasses. Bevor das Messsignal nun den ADC AD7688 erreicht, wird es zunächst noch in ein differenzielles Signal aufgespaltet, welches der Wandler verarbeiten kann.

Die letzte Baugruppe (Baugruppe 6) stellt die Generierung der Ausgangsspannungen dar. Hierfür stehen mehrere Möglichkeiten zur Verfügung. Zum Einen ist dies die digitale Ausgabe per USB oder Ethernet, zum Anderen die analoge Ausgabe, welche wiederum mit Hilfe eines DACs (AD5064-1) mit nachgeschaltetem Rekonstruktionsfilter erfolgt.

### 3.1.1 Prozessormodul

Der in Abschnitt 3.1 erwähnte digitale Signalprozessor ADSP-BF527 [1] wird in diesem Lock-in Verstärker nicht direkt verbaut. Stattdessen kommt ein sogenanntes Coremodul (CM-BF527 Version 1.1.4) von Bluetechnix [13] zum Einsatz. Diese Firma integriert den bereits erwähnten Signalprozessor, RAM, Flash-Speicher, Ethernet PHY und die Spannungsversorgung in ein sehr kompaktes Modul (siehe funktionelles Blockschaltbild in Abbildung 3.2). Somit stehen dem Entwickler ein Mikroprozessor mit bis zu 600MHz

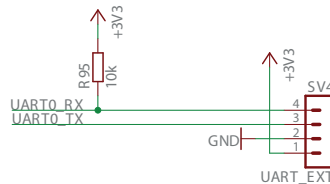


**Abbildung 3.2:** Funktionelles Blockschaltbild des CM-BF527 von Bluetechnix (entnommen aus [13])

Taktrate (eine Kombination aus einem 32-Bit RISC Prozessor und einem 16-Bit Fest-

komma DSP), 32MByte SDRAM, 8 MByte Flash-Speicher sowie sämtliche gängige Schnittstellen wie UART, SPI, SPORT, GPIO, Ethernet und USB2.0 zur Verfügung. Wie diese Schnittstellen im Zuge dieser Masterarbeit verwendet werden, verdeutlicht nachfolgende Aufstellung.

**Universal Asynchron Receiver Transmitter (UART)** Der ADSP-BF527 besitzt zwei serielle full-duplex Schnittstellen, wovon jedoch nur eine verwendet wird. Sie dient nur zu Entwicklungszwecken und ist deshalb nicht auf die Gehäuseaußenseite geführt. Die UART-Spannungspegel des ADSP-BF527 sind 3.3 V kompatibel, weshalb ein Pegelwandler nachgeschaltet werden muss, um für die Verwendung mit einem PC kompatible Pegel zu erreichen. Dieser ist jedoch nicht Teil dieser Arbeit und aus diesem Grund auch nicht in der Schaltung realisiert. Über die UART Schnittstelle erfolgt die Konsolenein- und -ausgabe zur Steuerung des Prozessormoduls während der Programmentwicklung (näheres zum Entwicklungsprozess ist in Abschnitt 3.3 erläutert). In Abbildung 3.3 ist die Beschaltung der UART Schnittstelle dargestellt. Der Widerstand R95 dient dazu, dem Prozessor am Eingang definierte 3.3 V Pegel zu liefern.



**Abbildung 3.3:** Beschaltung der UART-Schnittstelle

**Serial Peripheral Interface (SPI)** Mit Hilfe der SPI werden die beiden digitalen Potentiometer angesteuert, welche die Verstärkungen der Instrumentierungsverstärker im Messsignalpfad bestimmen (Siehe Abschnitt 3.1.3).

**Serial Port (SPORT)** Die beiden SPORTs des ADSP-BF527 sind, wie die SPI, ebenfalls serielle full-duplex Schnittstellen. Sie besitzen je zwei synchrone Ein- und Ausgangssignale mit eigenen Receive- und Transmittclocksignalen. Des Weiteren kann der SPORT Controller des ADSP-BF527 Framesyncsignale mit bestimmter Frequenz erzeugen. Somit ist der SPORT prädestiniert dazu, automatisch per DMA, die zwei

ADCs (IC9 sowie IC18) auszulesen und gleichzeitig Werte an den am selben SPORT (SPORT0) angeschlossenen DAC IC23 auszugeben. Hierzu sind die beiden ADCs jeweils mit einer der beiden Datenleitungen verbunden (SP0\_DRPRI und SP0\_DRSEC). Um eine funktionierende Kommunikation über diese Schnittstelle zu erreichen, benötigen diese Bauteile jedoch noch eine Taktleitung sowie eine Framesyncleitung, welche die Funktion des Chipselects innehat. Diese Leitungen (SP1\_RSCLK, SP1\_RFS) sind für beide ADCs die Gleichen, was bedeutet, dass beide Bauteile synchron zueinander arbeiten.

Der DAC IC23 kommuniziert über die drei Leitungen SP0\_DTPRI, SP0\_TSCLK und SP0\_TFS mit dem Prozessor. Somit können Transmit- und Receivekanäle vollkommen unabhängig voneinander betrieben werden, was hier beim Lock-in Verstärker bedeutet, dass das Auslesen und das Ausgeben mit verschiedenen Taktraten erfolgen kann.

Über den zweiten SPORT, den SPORT1, erfolgt die Übermittlung der Sinuswerte an den DAC IC10. Wiederum geschieht das über eine Datenleitung (SP1\_DTPRI), eine Clockleitung (SP1\_TSCLK) und eine Framesyncleitung (SP1\_TFS).

**Joint Test Action Group Debugschnittstelle (JTAG)** Die JTAG Schnittstelle dient, wie die UART, nur dem Entwicklungsprozess. In Verbindung mit einem JTAG-Adapter (z.B. gnICE) lassen sich über diese Schnittstelle z.B. neue Programme in verschiedene Speicherbereiche laden. Außerdem besteht die Möglichkeit, unter Zuhilfenahme eines Debuggers Fehlerquellen in Programmen zu suchen.

**Ethernet** Die Ethernetschnittstelle bildet ein wichtiges Kommunikationselement des Lock-in Verstärkers. Im laufenden Betrieb erfolgt über diese Schnittstelle einerseits die Steuerung, andererseits kann zusätzlich darüber die digitale Ausgabe der Messwerte erfolgen. Im Entwicklungsprozess wird über die Ethernetschnittstelle das Betriebssystemimage per Trivial File Transfer Protocol (TFTP) an den ADSP-BF527 übermittelt. Die Beschaltung ist in Abbildung 3.4 ersichtlich.

D6 ist ein Schutzelement, das sämtliche Leitungen, inkl. VA25, der Ethernetschnittstelle vor ESD Impulsen schützt. Die Buchse CONN1 (mit eingebautem Pulswandler) realisiert die RJ45 Verbindung in Richtung Ethernet. Diese Buchse hat zwei LEDs eingebaut, über die die Verbindungsgeschwindigkeit sowie die Netzwerkaktivität angezeigt

werden. Der Prozessor zieht bei entsprechender Aktivität die jeweiligen Ausgangsleitungen (ETH\_LED\_ACT bzw. ETH\_LED\_SPD) auf LOW-Pegel. R98 sowie R99 begrenzen bei 2.2 V Vorwärtsspannung der LEDs den Strom auf 5mA.

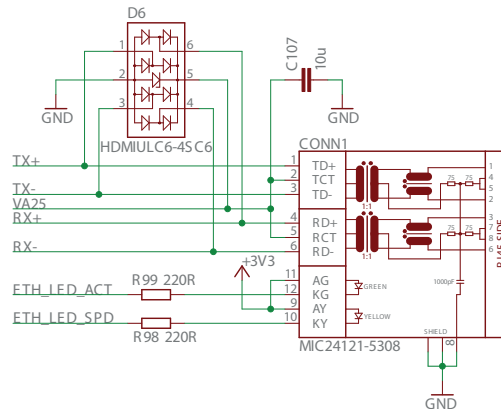


Abbildung 3.4: Beschaltung der Ethernetschnittstelle

**Universal Serial Bus (USB)** Der USB-Controller des ADSP-BF527 bietet die Möglichkeit, den Prozessor nicht nur als Peripherie Device, sondern auch als USB OTG Host Device laufen zu lassen. Im Lock-in Verstärker wird der Prozessor nur im Peripherie-Modus genutzt, was schon alleine an der USB Buchse (Buchse USB-Typ B) zu erkennen ist. Abbildung 3.5 zeigt, wie die restliche Schaltung des Lock-in Verstärkers von Störeinflüssen, die über die USB Schnittstelle kommen, abgeschirmt wird. Wie schon bei der Ethernetschnittstelle, wird auch hier mit Hilfe von D13 die Prozessorseite der Schnittstelle gegen schnelle ESD-Impulse auf den Datenleitungen USB\_D+ sowie USB\_D- geschützt. ESD Impulse auf der USB\_VBus Leitung blockt der Varistor VR1 ab. In diesem Fall begrenzt R97 den Strom.

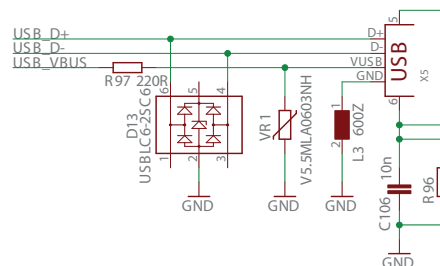
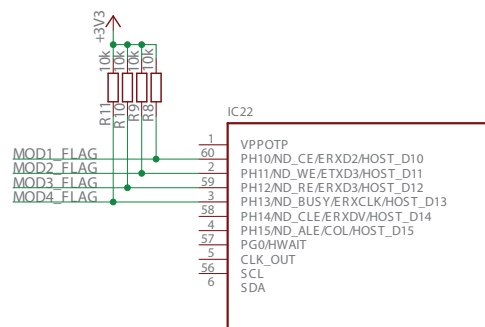


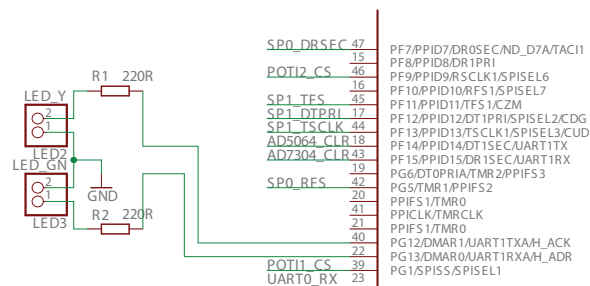
Abbildung 3.5: Beschaltung der USB-Schnittstelle

**General Purpose Input/Output (GPIO)** Neben den zuvor genannten Schnittstellen werden auch einige GPIO PINs verwendet. Diese PINs dienen einem Prozessor als allgemeine Ein- bzw. Ausgabeschnittstellen. Im Fall des Lock-in Verstärkers kann über die als Eingänge konfigurierten PINs PH10 bis PH13 festgestellt werden, ob und welche Modulationsplatten auf das Motherboard aufgesteckt sind. R8 bis R11 (siehe Abbildung 3.6) sind in diesem Fall Pullup Widerstände, die die PINs im Fall eines nicht eingesteckten Modulationsboards auf +3.3V legen. Ist eine Platine aufgesteckt, wird der entsprechende PIN des Prozessors (PH10, PH11, PH12 oder PH13) durch das Layout des Modulationsboards auf Masse gezogen. Dies kann die Software zur Detektion verwenden.



**Abbildung 3.6:** Detektierung der Modulationsplatten (Auszug aus dem Schaltplan)

PG12 und PG13 sind dafür vorgesehen zwei LEDs anzusteuern, mit Hilfe derer der Status des Gerätes angezeigt wird. Die Widerstände R1 und R2 begrenzen den Strom durch die beiden LEDs und werden laut Gleichung 3.1 berechnet.



**Abbildung 3.7:** Schaltung der Status - LEDs (Auszug aus dem Schaltplan)

$U_{ges}$  ... maximale Ausgangsspannung, die an einem PIN anliegen kann (siehe Datenblatt [1])

$U_f$  ... Vorwärtsspannung der LED (siehe Datenblatt [23])

$I_{soll}$  ... Strom, der durch die LED fließen soll (gewählt)

$$U_{ges} = 3.3V$$

$$U_f = 2.2V$$

$$I_{soll} = 5mA$$

$$R_{vor} = \frac{U_{ges} - U_f}{I_{soll}} = \frac{3.3V - 2.2V}{0.005A} = 220\Omega \quad (3.1)$$

Mittels der Signalleitungen AD5064\_CLR und AD7304\_CLR können per Software die internen Schieberegister der beiden Digital-Analog-Wandler IC10 und IC23 gelöscht werden.

**Einstellung des Bootmodus** Der ADSP-BF527 bietet die Möglichkeit, von verschiedensten Quellen, wie z.B. externer Flashspeicher, UART oder SPI Schnittstelle, zu booten. Diese Bootmodi werden über die PINs BMODE0 bis BMODE3 ausgewählt. Tabelle 3.1 listet die möglichen Bootmodi auf, die durch die Schalterstellungen von Schalter S5 bestimmt werden. Da das U-Boot (der Bootloader des Lock-in Verstärkers) im Flashspeicher liegt, muss hier Bootmode 1 gewählt werden.

Aus Tabelle 3.2 ist die Pinbelegung und die Funktion der verwendeten PORT Leitungen des CM-BF527 zu entnehmen.

### 3.1.2 Signalgenerator

Kernstück der Sinuserzeugung ist IC10, ein vierfach-DAC von Analog Devices, welcher, wie schon etwas weiter vorne erwähnt wurde, über den SPORT1 an das Prozessormodul angebunden ist. Beim AD7304 handelt es sich um einen Baustein mit vier 8-Bit Wandlern, der aufgrund seiner Settlingtime von  $2\mu s$  mit einer Ausgaberate von bis zu 500 kSps betrieben werden kann. Die Ausgaberate muss, um, wie Oppenheim und Schäfer in [27] erläutern, das Nyquist-Abtasttheorem zu befolgen, mindestens doppelt so groß sein, wie die höchste von Null verschiedene Signalfrequenz (Nyquistfrequenz  $\Omega_N$ ). Ausgehend von einer maximalen erwünschten Modulationsfrequenz von 7 kHz müsste somit die Ausgabe der Sinuswerte mit mindestens 14 kHz erfolgen. Um jedoch

**Tabelle 3.1:** Einstellungsmöglichkeiten von BMODE0, BMODE1, BMODE2 und BMODE3 zur Auswahl des Bootmodus

PIN Konfig. (BM3, BM2, BM1, BM0)	Bootmode	Beschreibung
0000	0	(default) Idle - No boot Reserved
0001	1	Boot from 8- or 16-Bit external flash memory
0010	2	Boot from 16-Bit asynchronous FIFO.
0011	3	Boot from serial SPI memory (EEPROM or flash)
0100	4	Boot from SPI host device
0101	5	Boot from serial TWI memory (EEPROM/flash)
0110	6	Boot from TWI host
0111	7	Boot from UART0 Host
1000	8	Boot from UART1 Host
1001	9	Reserved
1010	10	Boot from SDRAM
1011	11	Boot from OTP memory
1100	12	Boot from 8-Bit NAND flash via NFC using PORTF data pins
1101	13	Boot from 8-Bit NAND flash via NFC using PORTH data pins
1110	14	Boot from 16-Bit Host DMA
1111	15	Boot from 8-Bit Host DMA

später eventuell andere Frequenzen wählen zu können, wurde die Ausgabefrequenz mit 125 kHz fixiert. Die maximale Ausgangsspannung der einzelnen DACs wird durch die Referenzspannung (siehe Abschnitt 3.1.5) von 4.096 V definiert.

Der AD7304 besitzt die zwei zusätzlichen Steuerleitungen  $\overline{\text{CLR}}$  (PIN 8) und  $\overline{\text{LDAC}}$  (PIN 7).  $\overline{\text{CLR}}$  löscht alle DAC Register. Diese Leitung ist mit einem GPIO PIN des Prozessors verbunden, um ein Löschen per Software über einen LOW-Impuls zu ermöglichen. Ebenfalls mit einem LOW-Impuls, diesmal auf der zweiten Steuerleitung ( $\overline{\text{LDAC}}$ ), werden die Inhalte aller vier Eingangsregister an die entsprechenden Ausgangsregister weitergeleitet. Diese Leitung ist per Platinenlayout auf Masse gelegt, was dazu führt, dass Werte, die an IC10 gesendet werden, sofort im entsprechenden Ausgaberegister geladen und somit ausgegeben werden. Um die Quantisierungsstufen, die jeder DAC produziert, zu glätten, muss ein Rekonstruktionsfilter eingesetzt werden. Dieser besteht, wie Abbildung 3.8 zeigt, aus einem aktiven Tiefpassfilter zweiter Ordnung mit einer Verstärkung von 1.121 um bei einer Referenzspannung von 4.096 V die geforderte



Tabelle 3.2: Pinbelegung des CM-BF527

Port	PIN	Signal	Funktion	Beschreibung
PH10	60	MOD1_FLAG	GPIO Input	Erkennung Modul 1
PH11	2	MOD2_FLAG	GPIO Input	Erkennung Modul 2
PH12	59	MOD3_FLAG	GPIO Input	Erkennung Modul 3
PH11	3	MOD4_FLAG	GPIO Input	Erkennung Modul 4
PF0	11	SP0_DRPRI	DR0PRI	Prim. Empfangskanal SPORT0
PF1	50	SP0_RFS	RFS0	Receive Framesync SPORT0
PF2	12	SP0_RSCLK	RSCLK0	Receive Clock SPORT0
PF3	49	SP0_DTPRI	DT0PRI	Prim. Sendekanal SPORT0
PF4	13	SP0_TFS	TFS0	Transmit Framesync SPORT0
PF5	48	SP0_TSCLK	TSCLK0	Transmit Clock SPORT0
PF7	47	SP0_DRSEC	DR0SEC	Sek. Empfangskanal SPORT0
PF9	46	POTI2_CS	SPISEL6	SPI Chipselect Digipot IC16
PF11	45	SP1_TFS	TFS1	Transmit Framesync SPORT1
PF12	17	SP1_DTPRI	DT1PRI	Prim. Sendekanal SPORT1
PF13	44	SP1_TSCLK	TSCLK1	Transmit Clock SPORT1
PF14	18	AD5064_CLR	GPIO Output	Löschleitung für DAC IC23
PF15	43	AD7304_CLR	GPIO Output	Löschleitung für DAC IC10
PG5	42	SP0_RFS	TMR1	Steuerung RFS0 per Timer
PG12	40		GPIO Output	Ansteuerung LED gelb (LED2)
PG13	22		GPIO Output	Ansteuerung LED grün (LED1)
PG1	39	POTI1_CS	SPISEL1	SPI Chipselect Digipot IC12
PG8	23	UART0_RX	UART0RX	Empfangsleitung UART
PG7	38	UART0_TX	UART0TX	Sendeleitung UART
PG4	24	SPL_MOSI	SPIMOSI	SPI MOSI
PG3	37	SPL_MISO	SPIMISO	SPI MISO
PG2	25	SPL_SCK	SPISCK	SPI SCK

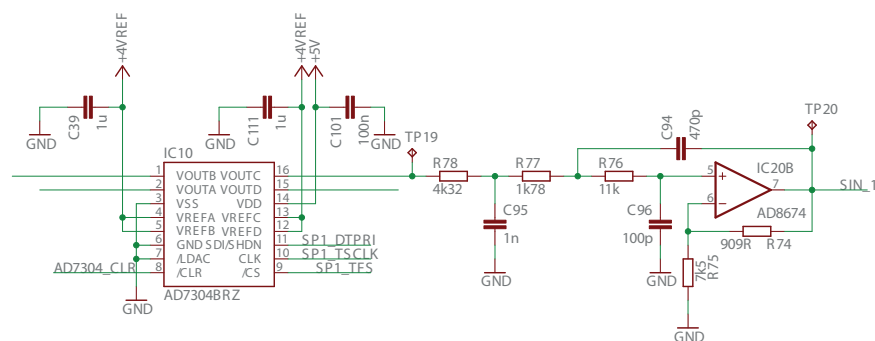


Abbildung 3.8: Hardwarerealisierung des Sinusgenerators (nur ein Kanal dargestellt)

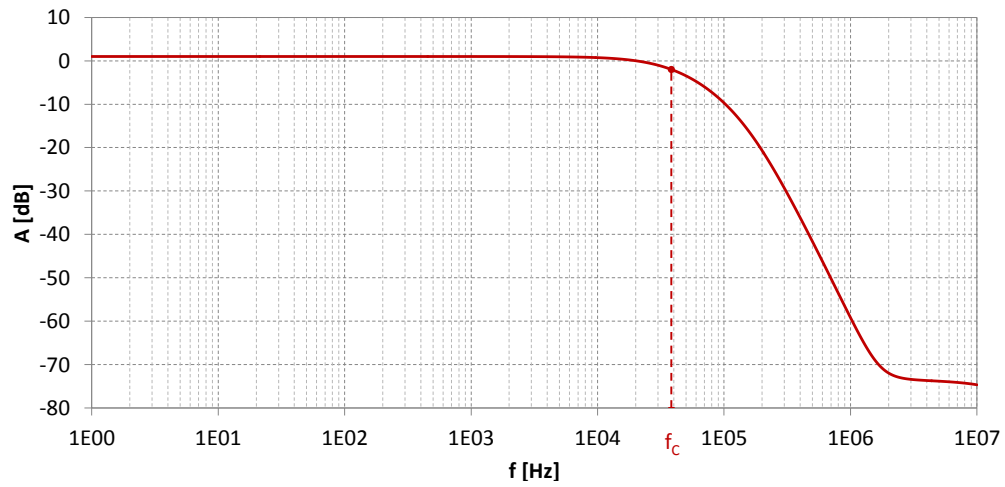
Ausgangsspannung von 4.592 V. Dem vorgeschaltet ist mit R78 und C95 ein passiver Tiefpass erster Ordnung, was im Gesamten einen Tiefpassfilter dritter Ordnung ergibt. Oppenheim und Schäfer zeigen in [27] wie die Grenzfrequenz  $f_C$  des Rekonstruktionsfilters gewählt wird. Sie beweisen, dass  $f_C$  zwischen  $\Omega_N$  (Nyquistfrequenz) und  $(\Omega_S - \Omega_N)$  ( $\Omega_S$  entspricht der Samplefrequenz), also hier zwischen 7 kHz und  $(125 \text{ kHz} - 7 \text{ kHz}) = 118 \text{ kHz}$ , liegen muss, um das gesampelte Signal wieder vollständig rekonstruieren zu können. Aufgrund dieser Vorgaben wurde die Grenzfrequenz mit 40 kHz festgelegt. Die Filterschaltung, die hier Anwendung findet, wurde, wie Hartl et al. in [21] beschreiben, dimensioniert. Hierzu wurden die Widerstands- und Kondensatorwerte eines Dimensionierungsbeispiels verwendet, den Anforderungen entsprechend skaliert und anschließend mit Hilfe des freien Softwaretools LTSpice von Linear Technologies [24] simuliert. Das Ergebnis der AC-Analyse ist in Abbildung 3.9 zu sehen. Da bei diesem Lock-in Verstärker nur die Amplitude, nicht jedoch die Phasenverschiebung zwischen Referenzsignal und Messsignal gemessen wird, ist es auch nicht von Bedeutung, welche Phasenverschiebung die einzelnen Filter verursachen. Aus diesem Grund ist in Abbildung 3.9 nur der Amplitudengang dargestellt.

Als Operationsverstärker (IC20) dient in der Filterschaltung ein AD8761 von Analog Devices. Er wurde aufgrund seiner Bandbreite von 10 MHz, seiner sehr geringen Spannungsrauschdichte von maximal  $3.8 \text{ nV}/\sqrt{\text{Hz}}$  bei 1 kHz und seiner geringen Offsetspannung von  $75 \text{ }\mu\text{V}$  ausgewählt.

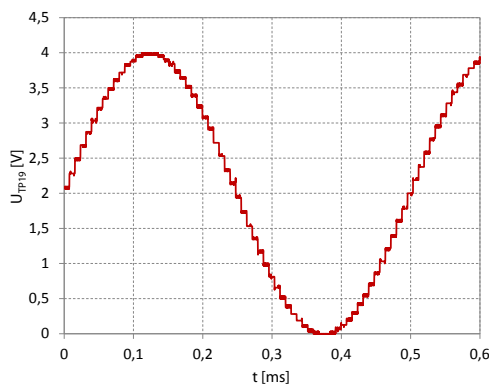
$$U_{TP20} = A \cdot U_{TP19} = 1.121 \cdot 4.096 \text{ V} = 4.592 \text{ V} \quad (3.2)$$

Zusammenfassend beschrieben kann der Lock-in Verstärker mit Hilfe der gerade erläuterten Beschaltung maximal vier Referenzspannungen mit einer Ausgangsspannung von je 4.592 V (siehe Gleichung 3.2) erzeugen.

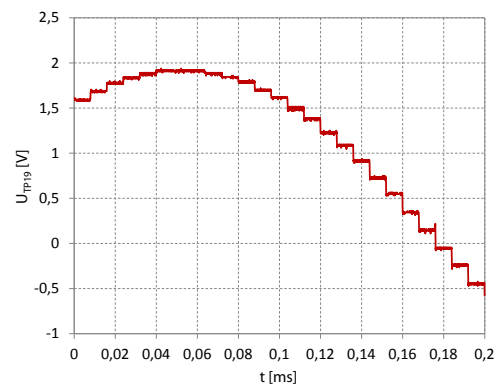
Zum Abschluss dieses Abschnitts über den Signalgenerator soll nun noch die Funktionsfähigkeit des dimensionierten Filters veranschaulicht werden. Abbildungen 3.10 bis 3.13 zeigen die Signalformen der Ausgangsspannungen der einzelnen Kanäle, welche mit Frequenzen von 2 kHz, 3 kHz, 5 kHz und 7 kHz betrieben werden. Es sind jeweils die Spannungsverläufe vor und nach dem Rekonstruktionsfilter inklusive Detailausschnitten dargestellt. Für genauere Details zur softwareseitigen Generierung der Sinussignale sei an dieser Stelle auf Abschnitt 3.2 verwiesen.



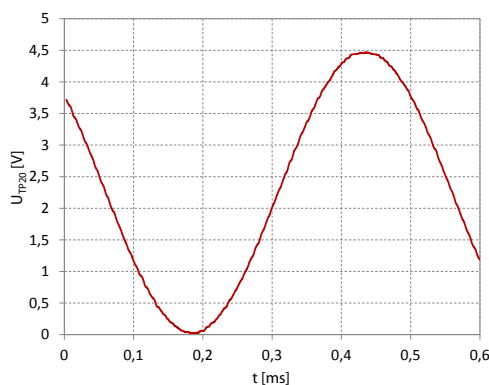
**Abbildung 3.9:** Amplitudengang des Rekonstruktionsfilters im Sinusgenerator



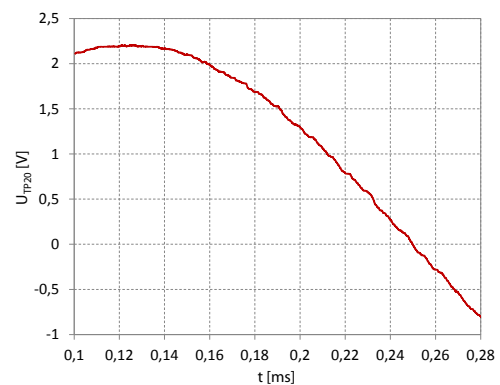
(a) DAC Ausgangsspannung  $U_{TP19}$



(b) Detail aus  $U_{TP19}$  (AC-Mode)

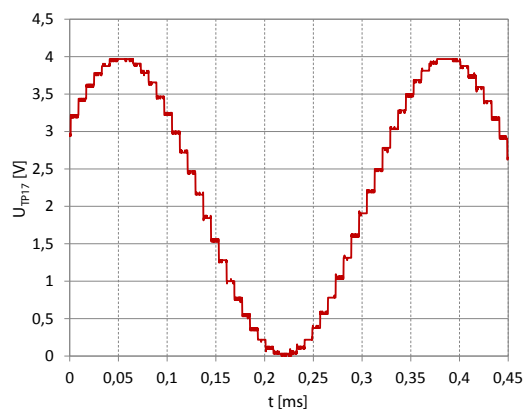
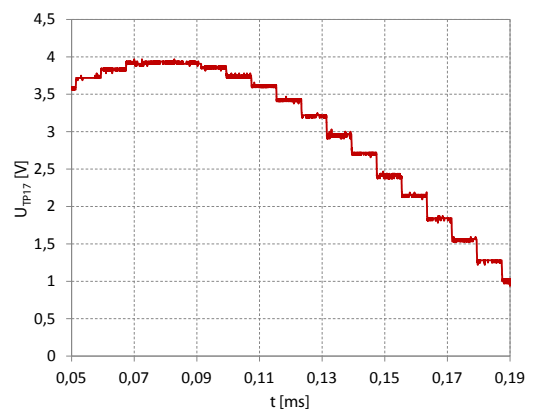
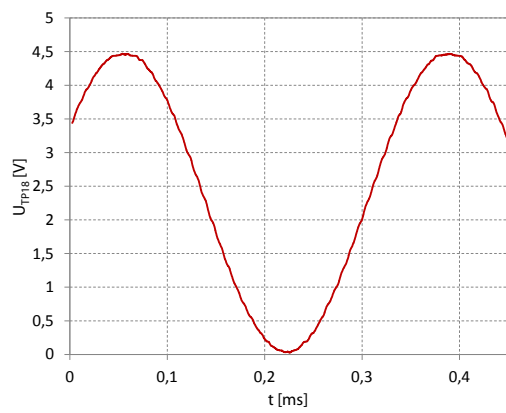
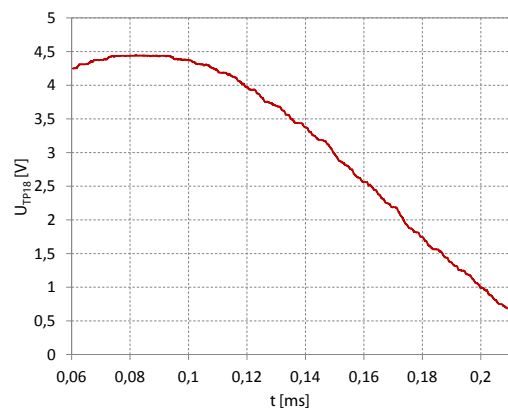


(c) Ausgangsspannung Filter ( $U_{TP20}$ )

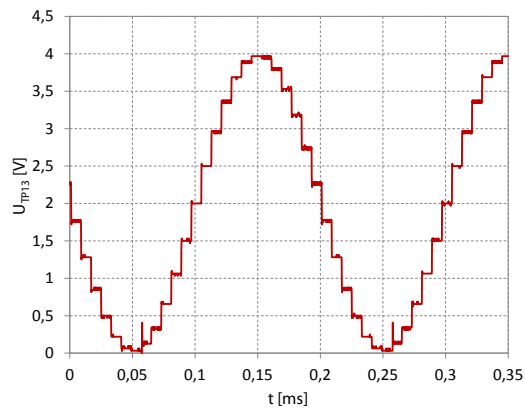
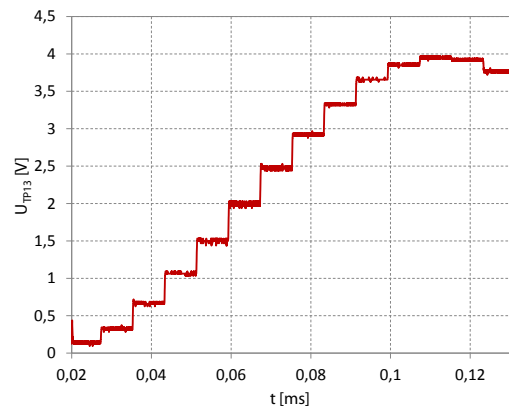
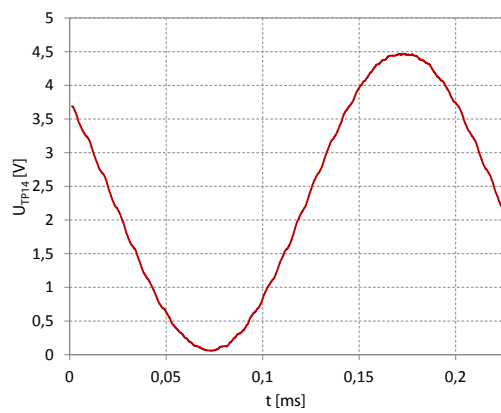
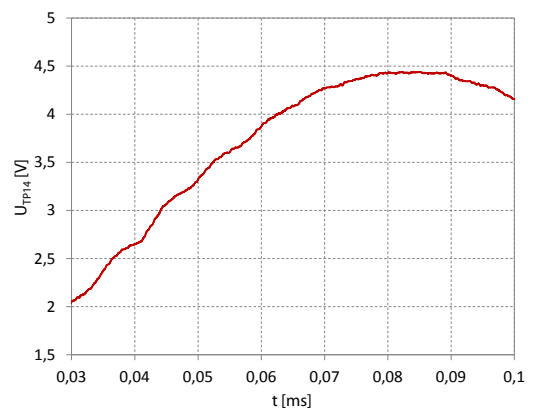


(d) Detail aus  $U_{TP20}$  (AC-Mode)

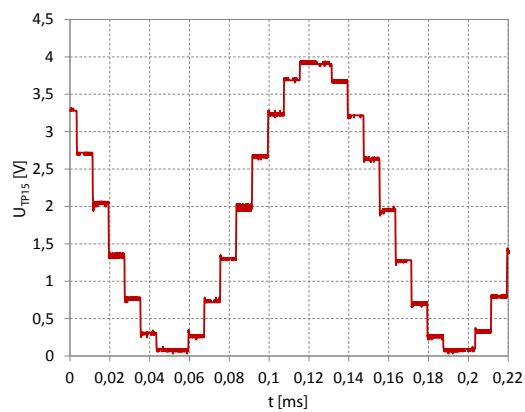
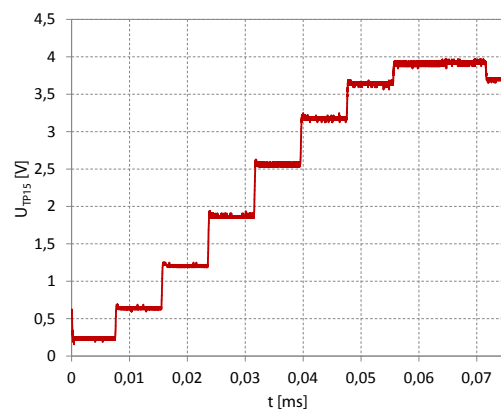
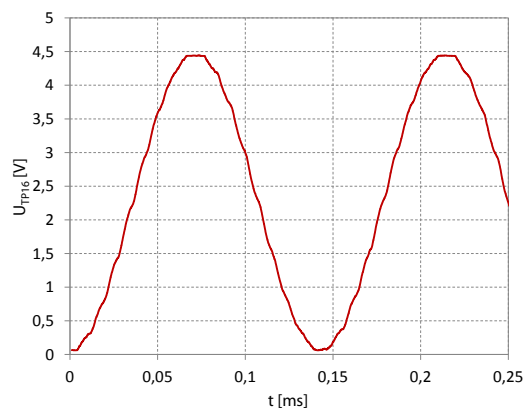
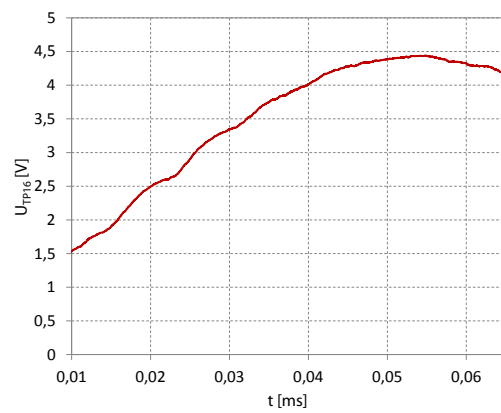
**Abbildung 3.10:** 2 kHz Ausgangsspannung vor (a, b) und nach (c, d) der Filterung

(a) DAC Ausgangsspannung  $U_{TP17}$ (b) Detail aus  $U_{TP17}$ (c) Ausgangsspannung Filter ( $U_{TP18}$ )(d) Detail aus  $U_{TP18}$ 

**Abbildung 3.11:** 3 kHz Ausgangsspannung vor (a, b) und nach (c, d) der Filterung

(a) DAC Ausgangsspannung  $U_{TP13}$ (b) Detail aus  $U_{TP13}$ (c) Ausgangsspannung Filter ( $U_{TP14}$ )(d) Detail aus  $U_{TP14}$ 

**Abbildung 3.12:** 5 kHz Ausgangsspannung vor (a, b) und nach (c, d) der Filterung

(a) DAC Ausgangsspannung  $U_{TP15}$ (b) Detail aus  $U_{TP15}$ (c) Ausgangsspannung Filter ( $U_{TP16}$ )(d) Detail aus  $U_{TP16}$ 

**Abbildung 3.13:** 7 kHz Ausgangsspannung vor (a, b) und nach (c, d) der Filterung

### 3.1.3 Messsignalpfad

Im folgenden Abschnitt wird nun die Schaltungsdimensionierung eines der beiden Messsignalpfade erläutert. Da beide Messsignalpfade identisch im Aufbau sind, gilt diese Beschreibung genauso für den zweiten Pfad.

Das in der Einführung dieses Kapitels besprochene APD-Modul liefert eine zur Lichtintensität proportionale, negative Ausgangsspannung. Bei einer Bestrahlung mit einer sinusförmig modulierten Lichtintensität stellt sich somit eine sinusförmige Ausgangsspannung mit einem der Amplitude proportionalen negativen Offset ein. Bei einer Signalverstärkung im Messsignalpfad würde dieser Offset ebenfalls mitverstärkt werden, was zur Folge hätte, dass die Operationsverstärker des Instrumentierungsverstärkers sehr schnell in ihre Sättigung gingen und kein Signal mehr gemessen werden könnte. Aus diesem Grund wird gleich am Eingang des Messsignals ein Hochpass mit einer Grenzfrequenz von  $f_g = 10\text{ Hz}$  geschaltet. Abbildung 3.14 zeigt, wie R33, R34 sowie C42 und C43 diesen passiven Bipolarfilter bilden. R34 bzw. R35 sind mit  $1\text{ M}\Omega$  sehr groß gewählt, um die Ausgangsstufe des APD-Moduls so wenig wie möglich zu belasten.

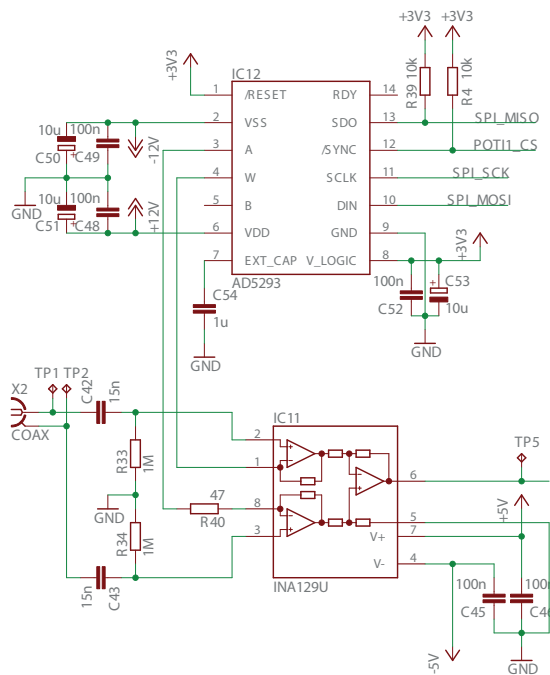


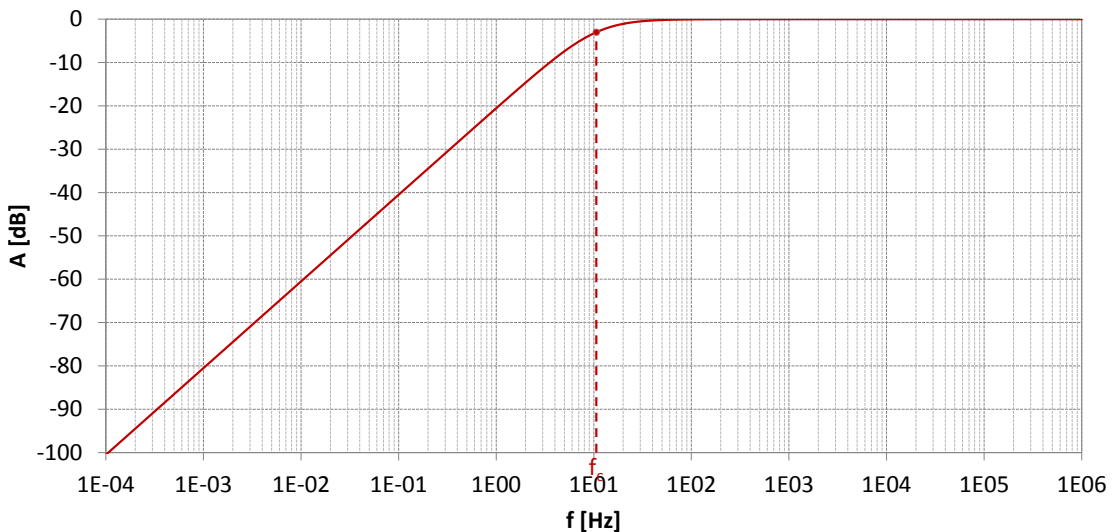
Abbildung 3.14: Eingangsstufe des Messsignalpfades

Sind diese Widerstandswerte festgelegt, so können mit Hilfe der Gleichung 3.4 durch

Umformung die Kapazitätswerte für C42 und C43 bestimmt werden (siehe Gleichung 3.4).

$$f_g = \frac{1}{2\pi R_{33} C_{42}} \quad (3.3)$$

$$C_{42} = \frac{1}{2\pi R_{33} f_g} = \frac{1}{2 \cdot 3.142 \cdot 1 \text{ M}\Omega \cdot 10 \text{ Hz}} = 15.915 \text{ nF} \quad (3.4)$$



**Abbildung 3.15:** Amplitudengang des Hochpassfilters am Eingang des Messsignalpfades

Die AC-Analyse mit LTSpice, welche in Abbildung 3.15 zu sehen ist, ergibt, dass der Eingangsfiler korrekt dimensioniert wurde und mit den errechneten Bauteilwerten wie erwünscht arbeitet.

Nun ist das Signal bereit zur Verstärkung. Gemeinsam mit IC12 bildet der Instrumentierungsverstärker INA129U von Texas Instruments (IC11) die erste Verstärkerstufe. IC12 ist ein digitales 10-Bit 50 k $\Omega$  Potentiometer, dessen Schleiferposition per Software über die SPI-Schnittstelle verändert werden kann. Die Verstärkung des Instrumentierungsverstärkers lässt sich mit Hilfe der Gleichung 3.5 berechnen, wobei  $R_{pot}$  den Widerstandswert des digitalen Potentiometers und  $R_G$  den gesamten Verstärkungswiderstand darstellen. In der Nullposition hat das Potentiometer einen Restwiderstand von  $R_{WB,min} = 120 \Omega$ . Dies bedeutet, dass sich in Verbindung mit dem fix verbauten Vorwiderstand R40 von 47  $\Omega$  eine minimale Verstärkung von  $G_{instr,min} = 1.989$  (siehe Gleichung 3.6) und eine maximale



Verstärkung von  $G_{instr,max} = 1051.063$  (siehe Gleichung 3.7) ergibt.

$$G_{instr} = 1 + \frac{49.4 \text{ k}\Omega}{R_G} \quad (3.5)$$

$$\begin{aligned} G_{instr,min} &= 1 + \frac{49.4 \text{ k}\Omega}{R_{G,max}} = 1 + \frac{49.4 \text{ k}\Omega}{R_{AB} - R_{WB,min} + R_{40}} = \\ &= 1 + \frac{49.4 \text{ k}\Omega}{50 \text{ k}\Omega - 120 \Omega + 47 \Omega} = \\ &= 1.989 \end{aligned} \quad (3.6)$$

$$\begin{aligned} G_{instr,max} &= 1 + \frac{49.4 \text{ k}\Omega}{R_{G,min}} = 1 + \frac{49.4 \text{ k}\Omega}{R_{AB} - R_{WB,max} + R_{40}} = \\ &= 1 + \frac{49.4 \text{ k}\Omega}{50 \text{ k}\Omega - 50 \text{ k}\Omega + 47 \Omega} = \\ &= 1051.063 \end{aligned} \quad (3.7)$$

Die zweite Stufe im Messsignalpfad ist der Antialiasingfilter. Abbildung 3.16 zeigt im Detail wie dieser Filter aufgebaut ist. Es handelt sich hierbei um einen nicht verstärkenden aktiven Butterworthtiefpass sechster Ordnung, welcher aus drei aktiven Tiefpassfiltern zweiter Ordnung in Sallen-Key Topologie zusammengesetzt ist. Mit Hilfe des freien Softwaretools FilterPro Desktop von Texas Instruments ist es möglich, solch komplexe Filter relativ komfortabel zu dimensionieren. Oppenheim und Schäfer beschreiben in [27], dass die Eckfrequenz  $f_C$  eines Antialiasingfilters bei maximal der halben Abtastfrequenz liegen muss. Bei der Dimensionierung dieses Filters wurde absichtlich die Eckfrequenz möglichst tief gelegt, um etwaige Störungen schon vor der analog-digital Wandlung herauszufiltern. Da die maximal zu erwartete Modulationsfrequenz bei 7 kHz liegt, wurde somit  $f_C$  bei 15 kHz festgelegt. Als Operationsverstärker IC14 dient hier ein vierfach-OPV von Linear Technologies. Der LT1125CSW vereint eine sehr geringe Spannungsrauschdichte von  $5.5 \text{ nV}/\sqrt{\text{Hz}}$  bei 1 kHz, ein hohes Verstärkungs-Bandbreite-Produkt von 12.5 MHz sowie eine geringe Offsetspannung von maximal  $140 \mu\text{V}$ , was ihn für die Verwendung in aktiven Filtern sehr empfiehlt. Da das Eingangssignal um den Nullpunkt herum schwingt, müssen die Operationsverstärker symmetrisch, d.h. positiv und negativ (in diesem Fall mit  $\pm 5 \text{ V}$ ) versorgt werden.

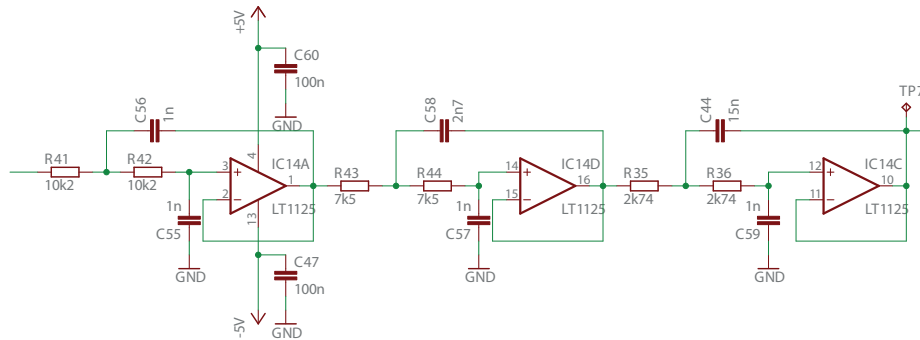


Abbildung 3.16: Antialiasingfilter im Messsignalpfad

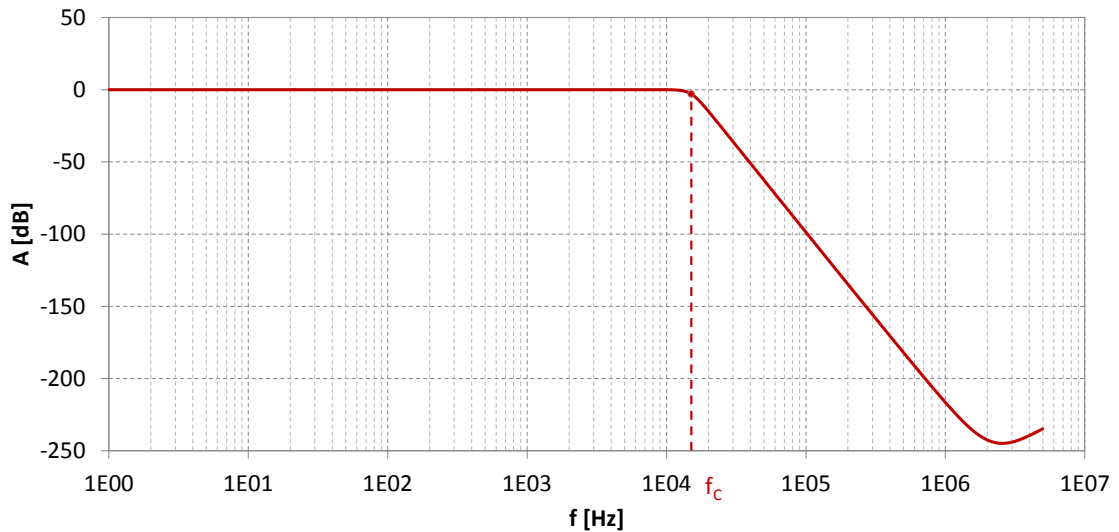


Abbildung 3.17: Amplitudengang des Antialiasingfilters

Nachdem das Messsignal nun durch den Antialiasingfilter in seiner Bandbreite beschnitten wurde, trifft es auf die nächste Stufe des Messsignalpfades, den ADC-Treiber (IC13). Dieser Baustein von Analog Devices hat einerseits die Aufgabe, das ankommende Signal noch einmal zu verstärken und andererseits die Aufgabe, aus einem massebezogenen Signal ein differenzielles Signal zu bilden, um dem ADC die Möglichkeit zu bieten, seine hervorragenden Eigenschaften in punkto Genauigkeit auszuspielen. Abbildung 3.18 veranschaulicht die Beschaltung des AD8139. Die Widerstände  $R_{38}$  und  $R_{46}$  sowie  $R_{37}$  und  $R_{45}$  bestimmen die Spannungsverstärkung der Schaltung. Sie lässt sich mit Hilfe von Gleichung 3.8 (siehe Datenblatt AD8139 [8]) bestimmen. Um die gewünschte Spannungsverstärkung von fünf zu erreichen, werden die Widerstände mit  $R_{45} = 1 \text{ k}\Omega$  und  $R_{37} = 200 \Omega$  gewählt. Die nächstgelegenen verfügbaren Widerstands-



voltage“ eingestellt werden. Hierbei handelt es sich um eine Offsetspannung, die zu jedem der beiden Ausgangssignalen hinzuaddiert wird. Da der verwendete ADC mit einer Referenzspannung von 2.5 V arbeitet, wird nun diese Offsetspannung über den Spannungsteiler, den die Widerstände R47 sowie R48 bilden, auf die halbe Referenzspannung, also 1.25 V, gesetzt. C64 dient zur Stabilisierung dieser Spannung.

Wie das Datenblatt des AD8139 [8] empfiehlt, muss bei der Dimensionierung überprüft werden, ob die Spannung  $V_{ACM}$  (Common Mode Voltage, die direkt an den Eingangspins des Treibers anliegt) innerhalb der Grenzen des Input Common Mode Voltage Bereiches liegt. Dieser Bereich lässt sich aus der Spezifikationstabelle des Datenblattes ablesen und reicht von  $-4\text{ V}$  bis  $4\text{ V}$ . Der Spannungsbereich von  $V_{ACM}$  wird durch Gleichung 3.12 bestimmt.

#### Rahmenbedingungen:

$$R_{38} = 205\ \Omega$$

$$R_{46} = 1\ \text{k}\Omega$$

$$V_{OCM} = 1.25\ \text{V}$$

*Output Common Mode Voltage*

$$V_{IN} = 0\ \text{V}$$

*Spannung am invertierenden Eingang*

$$V_{IP,max} = 5\ \text{V}$$

*maximale Spannung am nichtinvertierenden Eingang*

$$V_{IP,min} = -5\ \text{V}$$

*minimale Spannung am nichtinvertierenden Eingang*

Mit Hilfe der beiden Widerstandswerte aus den Rahmenbedingungen und Gleichung 3.10 kann nun der Rückkopplungsfaktor  $\beta$  bestimmt werden.

$$\beta = \frac{R_{38}}{R_{46} + R_{38}} = \frac{205\ \Omega}{1\ \text{k}\Omega + 205\ \Omega} = 0.170 \quad (3.10)$$

Gleichung 3.11 beschreibt, wie der Bereich berechnet wird, in dem sich die Input Common Mode Spannung bewegt. Da  $V_{IN}$  auf Masse liegt, fließt sie mit  $0\ \text{V}$  in diese Gleichung ein.  $V_{IP,min}$  und  $V_{IP,max}$  ist der Spannungsbereich, der aufgrund der symmetrischen Versorgungsspannung der Operationsverstärker des Antialiasingfilters von  $\pm 5\ \text{V}$  am nichtinvertierenden Eingang des AD8139 anliegen kann.

$$V_{ICM,min} = \frac{V_{IP,min} + V_{IN}}{2} = \frac{-5\ \text{V} + 0\ \text{V}}{2} = -2.5\ \text{V} \quad (3.11a)$$

$$V_{ICM,max} = \frac{V_{IP,max} + V_{IN}}{2} = \frac{5\ \text{V} + 0\ \text{V}}{2} = 2.5\ \text{V} \quad (3.11b)$$

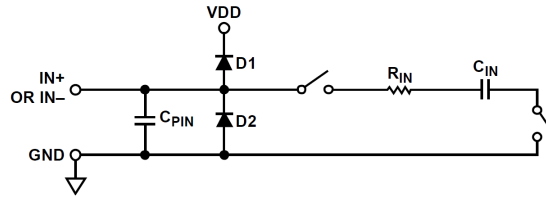
Mit den aus Gleichung 3.10 und 3.11 erhaltenen Werten kann nun schlussendlich der Wertebereich von  $V_{ACM}$  bestimmt werden.

$$\begin{aligned} V_{ACM,min} &= V_{ICM,min} + \beta(V_{OCM} - V_{ICM,min}) = \\ &= -2.5 \text{ V} + 0.170(1.25 \text{ V} + 2.5 \text{ V}) = -1.862 \text{ V} \end{aligned} \quad (3.12a)$$

$$\begin{aligned} V_{ACM,max} &= V_{ICM,max} + \beta(V_{OCM} - V_{ICM,max}) = \\ &= 2.5 \text{ V} + 0.170(1.25 \text{ V} - 2.5 \text{ V}) = 2.287 \text{ V} \end{aligned} \quad (3.12b)$$

Somit zeigt sich dass  $V_{ACM,min}$  und  $V_{ACM,max}$  innerhalb des geforderten Bereiches der Input Common Mode Spannung liegen.

Wie der Schaltplan in Abbildung 3.18 zeigt, befindet sich in jedem der beiden Signalpfade des AD8139 ein Tiefpass erster Ordnung (R67 und C90 bzw. R68 und C91). Dieser Tiefpass hat zwei Funktionen. Laut Datenblatt [8] würde ein direktes Treiben kapazitiver Lasten hochfrequente Schwingungen in der Sprungantwort verursachen. Die Eingänge des ADCs stellen, wie das Ersatzschaltbild eines solchen Einganges in Abbildung 3.19 zeigt, genau solche kapazitiven Lasten dar. Diese Schwingungen können mit Hilfe des Tiefpasses, dessen Grenzfrequenz bei 1.591 MHz liegt, gefiltert werden. Wie aus Abbildung 3.19 ebenso ersichtlich ist, besitzt jeder Eingang mit D1 und D2 eine ESD Schutzbeschaltung. Das bedeutet nun, wenn die am Eingang des ADCs angelegte Spannung größer als  $V_{DD} + 0.3 \text{ V}$  (in diesem Fall größer als 5.3 V) bzw. kleiner als  $-0.3 \text{ V}$  ist, eine dieser Dioden durchschaltet und somit einen Kurzschluss verursacht. Diese Bedingungen können hier im Normalbetrieb auftreten, weil die Operationsverstärker des Antialiasingfilters und der ADC-Treiber mit  $\pm 5 \text{ V}$  Versorgungsspannung betrieben werden und die Offsetspannung der Ausgänge des ADC-Treibers auf 1.25 V liegt. Somit kann es sein, dass bei Übersteuerung der OPVs, wenn z.B. die Verstärkung des Instrumentierungsverstärkers zu hoch eingestellt ist, die Eingangsspannung am ADC auf 6.25 V ansteigt, bzw. auf  $-3.75 \text{ V}$  absinkt. In so einem Fall muss der Strom, der durch die Dioden D1 bzw. D2 fließt, auf maximal 130 mA begrenzt werden. Dies ist die zweite Aufgabe von R49 bzw. R50, welche mit ihrem Wert von  $100 \Omega$  den Strom auf 62.5 mA bzw. 37.5 mA begrenzen. Die Letzte Stufe des Messsignalpfades ist die Konvertierung des Messsignals von einer Analogspannung in einen digitalen Wert. Dies erledigt IC9, ein 16-Bit ADC, mit bis zu 500 kHz und einer maximalen integralen Nichtlinearität (INL) von  $\pm 1.5 \text{ LSB}$ . Dies, die hohe SNR, sowie die Tatsache, dass dieser Wandler ein SPI kompatibles Interface besitzt, machen den AD7688 zum ADC der Wahl für den



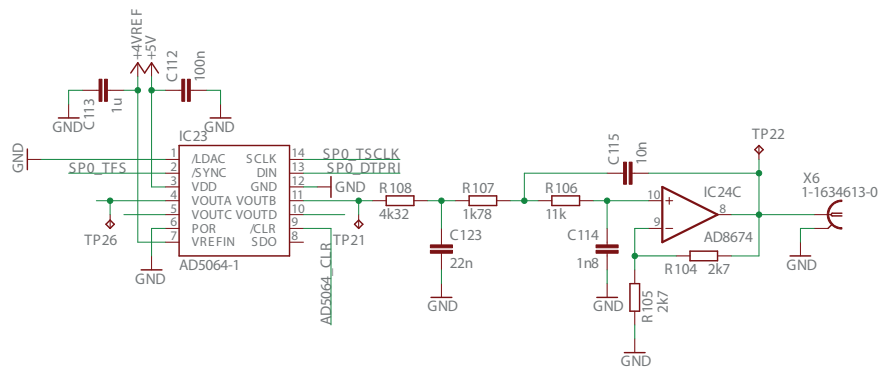
**Abbildung 3.19:** Ersatzschaltbild eines Einganges des ADCs AD7688 (Entnommen aus [9])

Einsatz in diesem Lock-in Verstärker. Um die benötigten Verstärkungen in Grenzen zu halten, wird der ADC mit einer Referenzspannung von 2.5 V betrieben. Aufgrund der hohen Auflösung sind somit bei maximaler Verstärkung rein rechnerisch Spannungsunterschiede von 7.440 nV feststellbar (siehe Gleichungen 3.1.3 und 3.1.3).

$$\Delta V_{X2} = \frac{V_{X2,min}}{2^{16}} = \frac{0.488 \text{ mV}}{65536} = 7.440 \text{ nV} \quad (3.13)$$

### 3.1.4 Analoges Ausgangssignalpfad

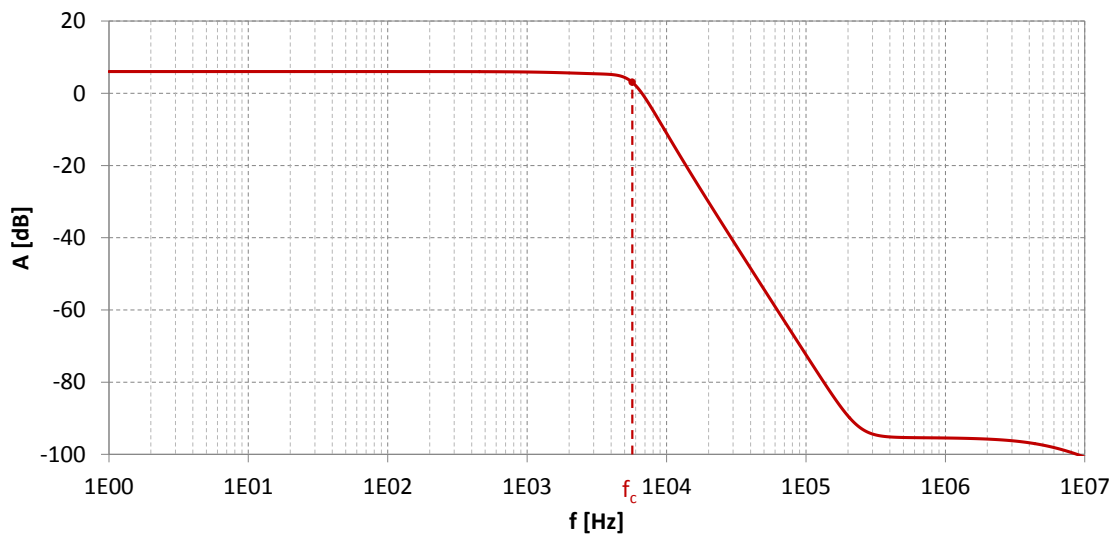
Das Kernstück des analogen Ausgangssignalpfades ist der Digital Analog Wandler IC23 von Analog Devices (siehe Abbildung 3.20). Wie bei den ADCs in den beiden Messsignalpfaden handelt es sich beim AD5064-1 um einen 16-Bit Wandler. Diese hohe Auflösung wurde gewählt, um eine möglichst feine Änderungen der Lichtintensität messen und auch wieder ausgeben zu können. Im Gehäuse des AD5064-1 befinden sich vier einzelne DACs welche alle an die selbe Referenzspannung von 4.096 V angeschlossen sind.



**Abbildung 3.20:** Detailschaltung eines der vier analogen Ausgangskanäle

Wie schon in Abschnitt 3.1.2 erklärt, muss auch hinter diesen DAC ein Rekon-

struktionsfilter geschaltet werden. Dieser Filter ist, genau wie die Rekonstruktionsfilter des Signalgenerators, ein Tiefpassfilter dritter Ordnung, welcher aus einem passiven Tiefpass erster Ordnung und einem aktiven Tiefpass zweiter Ordnung in Sallen-Key Topologie zusammengesetzt ist. Der Gesamtfiter hat eine Spannungsverstärkung von zwei, wodurch eine maximale analoge Ausgangsspannung von  $4.096 \text{ V} \cdot 2 = 8.192 \text{ V}$  erreicht werden kann. Zur Dimensionierung dieses Filters wurde ein Dimensionierungsbeispiel von Hartl et al. [21] herangezogen, welches für eine Eckfrequenz von  $f_c = 1 \text{ kHz}$  und eine Verstärkung von  $G = 2$  berechnet wurde. Da hier eine Grenzfrequenz von  $f_c = 5 \text{ kHz}$  erwünscht ist, werden die Werte für die Kondensatoren durch 5 dividiert. Abschnitt 2.1 beschreibt, dass ein NIRS Signal eine Frequenz im niederen Hertzbereich besitzt. Aus diesem Grund kann die Grenzfrequenz relativ niedrig angesetzt werden. Eine Grenzfrequenz von  $f_c = 5 \text{ kHz}$  bedeutet nun, dass die Signalfrequenz im Bereich  $0 \text{ Hz} \leq \Omega_N < 5 \text{ kHz}$  und die Samplefrequenz im Bereich  $5 \text{ kHz} > \Omega_S \geq 5 \text{ kHz} + \Omega_N$  liegen kann. Abbildung 3.21 zeigt das Simulationsergebnis mit LTSpice [24]. Wie zu erkennen ist, befindet sich die Grenzfrequenz  $f_c$  etwas oberhalb von  $5 \text{ kHz}$ . Diese Verschiebung ergibt sich durch die Verwendung realer Bauteilwerte.



**Abbildung 3.21:** Amplitudengang des Rekonstruktionsfilters am Analogausgang

### 3.1.5 Spannungsversorgung

Im Lock-in Verstärker finden sieben verschiedene Spannungsversorgungspegel ihre Anwendung. Neben den beiden Referenzspannungen von 2.5 V sowie 4.096 V für die ADCs (IC8, IC19) und DACs (IC10, IC23) werden noch 3.3 V,  $\pm 5$  V und  $\pm 12$  V verwendet.

**3.3V Spannungsversorgung:** Dieser Spannungspegel wird vornehmlich dazu verwendet, das Prozessormodul und dessen Peripherie zu versorgen. So sind sämtliche Pullup Widerstände und die LEDs der Ethernetbuchse an diesen Spannungspegel gebunden. Auch die Versorgungspins für die digitalen Ausgänge der beiden ADCs und der beiden digitalen Potentiometer sind auf 3.3 V gelegt, um mit den Spannungspegeln des Prozessormoduls kompatibel zu sein. Tabelle 3.3 bietet eine Abschätzung des Stromverbrauches bei 3.3 V, welche mit Hilfe der Datenblätter der einzelnen Bauteile erstellt wurde. Da der Stromverbrauch des Prozessormoduls stark von der Aktivität auf der Ethernetschnittstelle abhängt, wurde hier von Bluetechnix [13] nur ein Wert für den durchschnittlichen Verbrauch angegeben, welcher hier in der Abschätzung verwendet wird.

**Tabelle 3.3:** Abschätzung Stromverbrauch bei 3.3 V

Beschreibung	Anzahl	Strom einzel ca. [mA]	Strom gesamt ca. [mA]
Coremodul (IC22)	1	350	350
Bmode Auswahl	4	15	60
LED Ethernet	2	5	10
LED Status	2	5	10
2,5V ADR441 (IC25)	1	4	4
<b>Gesamtverbrauch ca.</b>			440

Nun kann mit Gleichung 3.14 die Leistung berechnet werden, welche die 3.3 V Spannungsquelle mindestens liefern muss.

$$P = U \cdot I \tag{3.14}$$

$$P_{3V} = 3 \text{ V} \cdot I_{3V} = 3 \text{ V} \cdot 440 \text{ mA} = 1.32 \text{ W}$$

Als Spannungsquelle dient hier ein Linearregler von Texas Instruments (siehe [30]). Der TPS73733 (IC29) wurde deshalb ausgewählt, weil er einen maximalen Ausgangsstrom von 1.6 A liefern kann, was der Abschätzung plus einiges an Reserven entspricht.



**$\pm 5\text{V}$  Spannungsversorgung:** Neben den  $3.3\text{V}$  wird im Lock-in Verstärker auch eine  $\pm 5\text{V}$  Versorgung verwendet. Dies hat den Grund, dass einerseits die verwendeten ADCs (IC8, IC19) und DACs (IC10, IC23) mit  $5\text{V}$  betrieben werden müssen und andererseits, wenn die im Messsignalpfad verwendeten Operationsverstärker symmetrisch mit  $\pm 5\text{V}$  versorgt werden, der Gesamtpegel nie größer als  $\pm 5\text{V}$  werden kann und somit die Eingänge der ADCs vor zu hohen Spannungspegeln geschützt werden. IC1, die  $4.096\text{V}$  Referenzspannungsquelle, wird ebenfalls mit  $5\text{V}$  versorgt, um eine zu hohe Verlustleistung, welche zwangsläufig beim herunterregeln einer Spannung entsteht, zu vermeiden. Tabelle 3.4 gibt eine Abschätzung über den Stromverbrauch für die  $\pm 5\text{V}$  Spannungsversorgung.

**Tabelle 3.4:** Abschätzung Stromverbrauch bei  $\pm 5\text{V}$

Beschreibung	Anzahl	Strom einzel ca. [mA]	Strom gesamt ca. [mA]
ADC Treiber (IC13,IC19)	2	30	60
OPV Antialiasing (IC14,IC17)	8	3	24
DAC AD7305 (IC10)	1	12	12
ADC AD7688 (IC9,IC18)	2	5	10
DAC AD5064 (IC23)	1	6	6
4V REF ADR444 (IC1)	1	4	4
Instr. Verst. (IC11,IC15)	2	1	2
<b>Gesamtverbrauch ca.</b>			<b>118</b>

Mit Hilfe dieser Abschätzung, die über die entsprechenden Datenblätter erfolgte, und Gleichung 3.15 kann die Leistung errechnet werden, die die  $5\text{V}$  Spannungsquelle mindestens liefern muss.

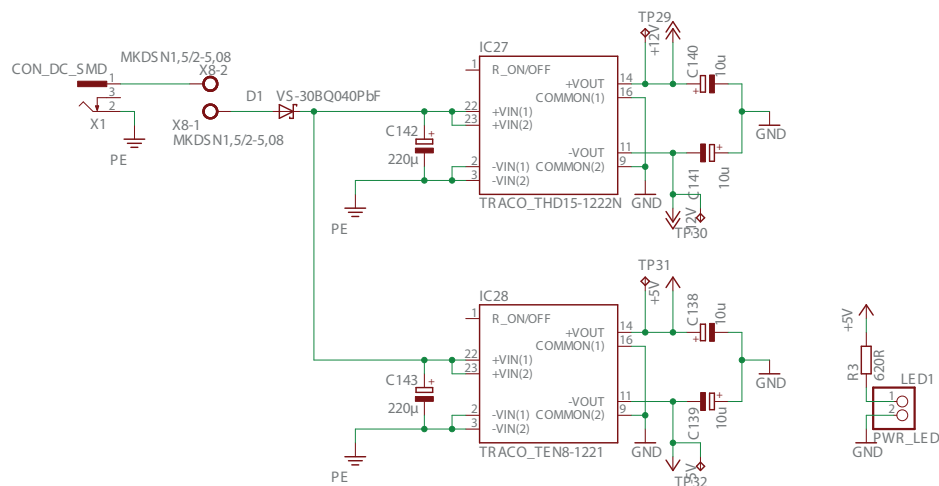
$$P = U \cdot I \quad (3.15)$$

$$P_{5V} = 5\text{V} \cdot I_{5V} = 5\text{V} \cdot 118\text{mA} = 0.59\text{W}$$

Da diese Spannungsquelle auch den  $3.3\text{V}$  Spannungsregler bedient, müssen auch dessen Leistung, plus die Verlustleistung des  $3.3\text{V}$  Reglers, hinzugezählt werden. Somit ergibt sich, wie Gleichung 3.16 zeigt, eine minimale Leistung von  $2.658\text{W}$ .

$$\begin{aligned} P_{5V,ges} &= P_{5V} + P_{3V} + P_{3V,verlust} = \\ &= 0.59\text{W} + 1.32\text{W} + (5\text{V} - 3.3\text{V}) \cdot 440\text{mA} = \\ &= 0.59\text{W} + 1.32\text{W} + 0.748\text{W} = 2.658\text{W} \end{aligned} \quad (3.16)$$

Die Realisierung der Spannungsquelle erfolgt durch einen DC/DC Wandler TEN8-1221 von Traco Power (IC28), der, wie Abbildung 3.22 zeigt, unsymmetrisch mit 12 V betrieben wird. Der Wandler generiert nun aus dieser Eingangsspannung die beiden benötigten Spannungen von  $\pm 5$  V mit einem Ausgangsstrom von bis zu jeweils 800 mA, was einer Gesamtleistung von 8 W entspricht. Die Verwendung eines DC/DC Wandlers hat den großen Vorteil, dass, ganz im Gegenteil zur Verwendung eines Linearreglers, viel weniger Verlustleistung bei der Spannungswandlung anfällt. Laut Datenblatt [31] schafft das der TEN8-1221 mit einem Wirkungsgrad von bis zu 83 %, was aufgrund der Spannungsdifferenz zwischen Ein- und Ausgangsspannung von großer Wichtigkeit ist. Des Weiteren ist der verwendete Wandler in einem kompakten Metallgehäuse gekapselt. Somit wird die empfindliche Messschaltung des Lock-in Verstärkers vor EMV Strahlung, die durch die Schaltvorgänge im Wandler entstehen, optimal geschützt.



**Abbildung 3.22:** Detailschaltung der  $\pm 5$  V und  $\pm 12$  V Spannungsversorgung

Die Eingangsbeschaltung von IC28 besteht aus einem Kondensator (C143) um, wie im Datenblatt [31] erwähnt wird, EN61000-4-4 und EN61000-4-5 zu erfüllen. Die Beschaltung der Ausgänge mit C138 und C139, wie sie Abbildung 3.22 zeigt, dient der Glättung der Restwelligkeit, die eine charakteristische Eigenschaft jedes DC/DC Wandlers ist.

**$\pm 12$  V Spannungsversorgung:** Diese beiden Spannungen sind notwendig, um die beiden verwendeten APD Module sowie die Modulationsplatten symmetrisch versor-

gen zu können. Außerdem benötigen die beiden digitalen Potentiometer IC12 und IC16 (Abbildung 3.14) eine Versorgungsspannung zwischen  $\pm 9\text{ V}$  und  $\pm 16.5\text{ V}$ , weswegen auch diese an das  $\pm 12\text{ V}$  Netz angeschlossen sind. Tabelle 3.5 gibt eine Abschätzung des Stromverbrauchs auf der  $\pm 12\text{ V}$  Schiene. Mit dieser Abschätzung kann über Gleichung 3.17 die Leistung errechnet werden, welche die Spannungsversorgung auf diesen Pegeln bringen muss.

**Tabelle 3.5:** Abschätzung Stromverbrauch bei  $\pm 12\text{ V}$

Beschreibung	Anzahl	Strom einzel ca. [mA]	Strom gesamt ca. [mA]
Modulationsplatinen	4	80	320
APD Module	2	50	100
OPV Sinusgenerator (IC20)	4	4	16
OPV Ausgang (IC24)	4	4	16
<b>Gesamtverbrauch ca.</b>			452

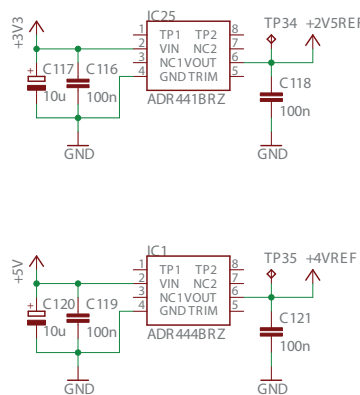
$$P = U \cdot I \tag{3.17}$$

$$P_{12V} = 12\text{ V} \cdot I_{12V} = 12\text{ V} \cdot 452\text{ mA} = 5.424\text{ W}$$

Für die Generierung der  $\pm 12\text{ V}$  wird, wie schon für die Generierung der  $\pm 5\text{ V}$  Versorgung, ein fertiger DC/DC Wandler von Traco Power verwendet. Der THD15-1222N ist ebenfalls mit  $12\text{ V}$  versorgbar und mit einer Effizienz von bis zu  $90\%$  viel effektiver als es ein Linearregler wäre [32]. Aus dieser Spannung generiert nun der Wandler (IC27) die benötigten  $\pm 12\text{ V}$ . Der THD15-1222N bietet mit  $15\text{ W}$  (je  $7.5\text{ W}$  auf der positiven und auf der negativen Schiene) mehr als genug Leistung, um die Abschätzung zu erfüllen sowie noch einiges an Reserven zu haben. Auch er ist komplett in einem DIP-24 Metallgehäuse gekapselt. Die Beschaltung ist aufgrund der großen Integriertheit dieses Bauteils sehr einfach und besteht, wie Abbildung 3.22 zeigt, lediglich aus einem Eingangskondensator, um, wie im Datenblatt [32] erwähnt wird, EN61000-4-4 und EN61000-4-5 zu erfüllen, sowie zwei Ausgangskondensatoren C140 und C141, um die Ausgangsspannung zu glätten.

**Referenzspannungsquellen:** Die ADCs sowie die DACs benötigen beide Referenzspannungsquellen, auf die sie Ihre Ein- bzw. Ausgangsspannungen beziehen können.

Diese Referenzspannungsquellen müssen hoch präzise sein, um ein optimales Ergebnis bei der Wandlung zu erzielen. Für die ADCs wird eine Spannung von 2.5 V verwendet. Sie ist aus den 3.3 V mit Hilfe der Referenzspannungsquelle ADR441 von Analog Devices generierbar. Die Spannung von 2.5 V wurde deshalb gewählt, da bei dieser Spannung die Verstärkungsfaktoren für den gewünschten Messbereich nicht zu groß werden. Für die DACs wird eine Referenzspannung von  $\pm 4.096$  V verwendet, welche von einem AD444 bereitgestellt wird. Die Beschaltung der beiden Referenzspannungsquellen ist in Abbildung 3.23 zu sehen und entspricht den Vorschlägen aus dem Datenblatt der beiden Bauteile [7].



**Abbildung 3.23:** Detailschaltung der beiden Referenzspannungsquellen ADR441 und ADR444

Da nun sämtliche Leistungen bekannt sind, kann nun errechnet werden, welche Gesamtleistung das Netzteil haben muss, um den gesamten Lock-in Verstärker zu versorgen. Die Maximalwertabschätzung erfolgt auf Basis der Leistungen der beiden DC/DC Wandler, zu welchen die Verlustleistung hinzugezählt wird. Gleichung 3.18 veranschaulicht diesen Rechenweg.

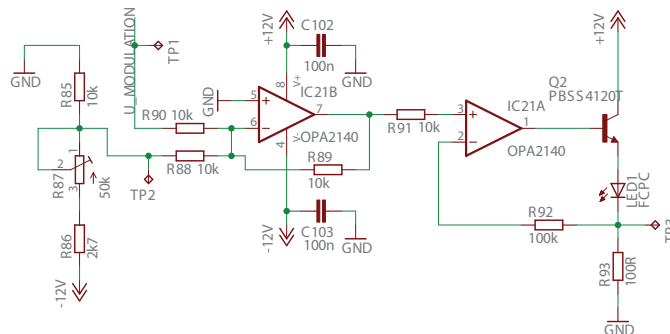
$$\begin{aligned}
 P_{gesamt} &= \frac{100 \cdot P_{IC28}}{83} + \frac{100 \cdot P_{IC27}}{90} = \\
 &= \frac{100 \cdot 8 \text{ W}}{83} + \frac{100 \cdot 15 \text{ W}}{90} = \\
 &= 9.639 \text{ W} + 16.667 \text{ W} = 26.306 \text{ W}
 \end{aligned}
 \tag{3.18}$$

Ein entsprechendes Netzteil wurde mit dem PDM30US12 gefunden. Dieses Externe Desktopnetzteil bietet bei einer Ausgangsspannung von 12 V einen Ausgangsstrom von 2.5 A, was einer Ausgangsleistung von 30 W entspricht. Es wurde genau dieses Netzteil

ausgewählt, weil es laut Datenblatt [36] nach EN60601-1 medizingeräte zertifiziert ist.

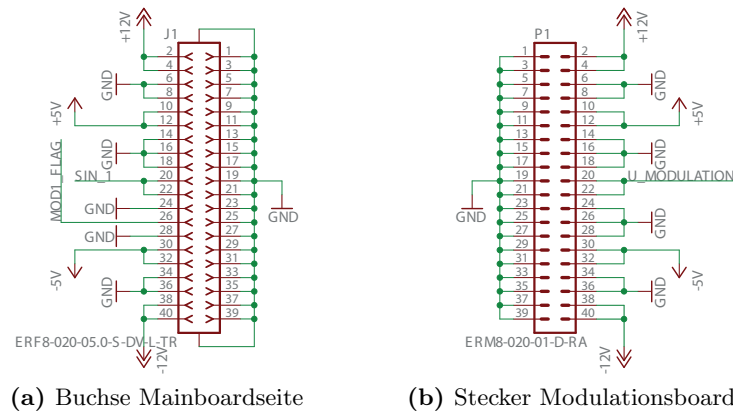
### 3.1.6 Modulationsplatinen

Die Modulationsplatinen stellen die Ausgabehardware des Sinusgenerators dar. Das bedeutet, dass sie mit Hilfe der Ausgangsspannungen von IC10 den Strom der Nahinfrarot LEDs und somit deren Leuchtintensität modulieren. Auf diesen Platinen befinden sich auch die Adapter, über die die Lichtwellenleiter das Licht der Nahinfrarot LEDs aufnehmen können. Die Schaltung, welche in Abbildung 3.24 abgebildet ist, wurde von Bauernfeind [12] übernommen. Aus diesem Grund wird für die Diskussion dieser Schaltung dorthin verwiesen. Die Modulationsplatinen werden senkrecht in die dafür



**Abbildung 3.24:** Schaltung der Modulationsplatinen

vorgesehenen Steckplätze eingebaut, wobei die Reihenfolge keine Rolle spielt. Es wurde eine Modularbauweise gewählt, um einerseits den Platz im Gehäuse optimal nutzen zu können und um andererseits bei einem Defekt der empfindlichen Leuchtdioden einfach ein defektes Modul tauschen zu können. Als Stecker-Buchse Kombination wurde die Verbindung ERF8/ERM8 von Samtec gewählt, welche eine ansprechende mechanische Belastbarkeit mitbringt. Aus diesem Grund wurde auch die PIN Anzahl von 20 ausgesucht. Gemeinsam mit der Durchführung der FCPC-Buchse (Lichtwellenleiteranschluss) durch die Gehäusefrontplatte, wird somit eine gewisse mechanische Beanspruchbarkeit sichergestellt. Abbildung 3.25 zeigt die Beschaltung des Steckers bzw. der Buchse der Steckverbindung. Aus Gründen der Flexibilität und der eventuellen späteren Erweiterbarkeit, wird neben der  $\pm 12\text{ V}$  Spannung auch die  $\pm 5\text{ V}$  Versorgungsspannung mitgeführt. Mit Hilfe von PIN26 wird der jeweilige GPIO PIN des Prozessormoduls (PIN60, PIN2, PIN59, PIN3; siehe Abbildung 3.6) auf Massepotential gelegt, um der Software zu signalisieren, welcher der vier Steckplätze belegt ist.



**Abbildung 3.25:** Steckverbindung zwischen Mainboard (a) und Modulationsboard (b)

### 3.1.7 Leiterplattendesign

Da innerhalb des entwickelten Lock-in Verstärkers (im Messsignalpfad) sehr kleine analoge Signale, sowie schnelle digitale Signale (SPORTs) existieren, muss besonderes Augenmerk auf das Platinenlayout gelegt werden, um zu verhindern, dass elektromagnetische Störungen der digitalen Spannungen in den analogen Signalpfad eingekoppelt werden. So wurde als erste Maßnahme das Mainboard als vierlagige Leiterplatte ausgeführt. Auf den außenliegenden beiden Layern werden Signalleitungen geführt, wobei darauf geachtet wurde, dass analoge und digitale Signale nicht parallel geführt werden und außerdem, falls sie nahe beieinander liegen, auf unterschiedlichen Layern geführt werden. Die beiden mittleren Layer sind eine Massefläche, sowie eine Schicht auf der nur Versorgungsspannungen geführt werden. Diese große Massefläche hat einerseits schirmende Wirkung, andererseits den Vorteil, dass sich keine Leiterschleifen zwischen hin- und rückführenden Leitern bilden können. Zudem wurden sämtliche Flächen auf der Leiterplatte, welche nicht von Leiterbahnen bedeckt sind, als Massefläche ausgeführt.

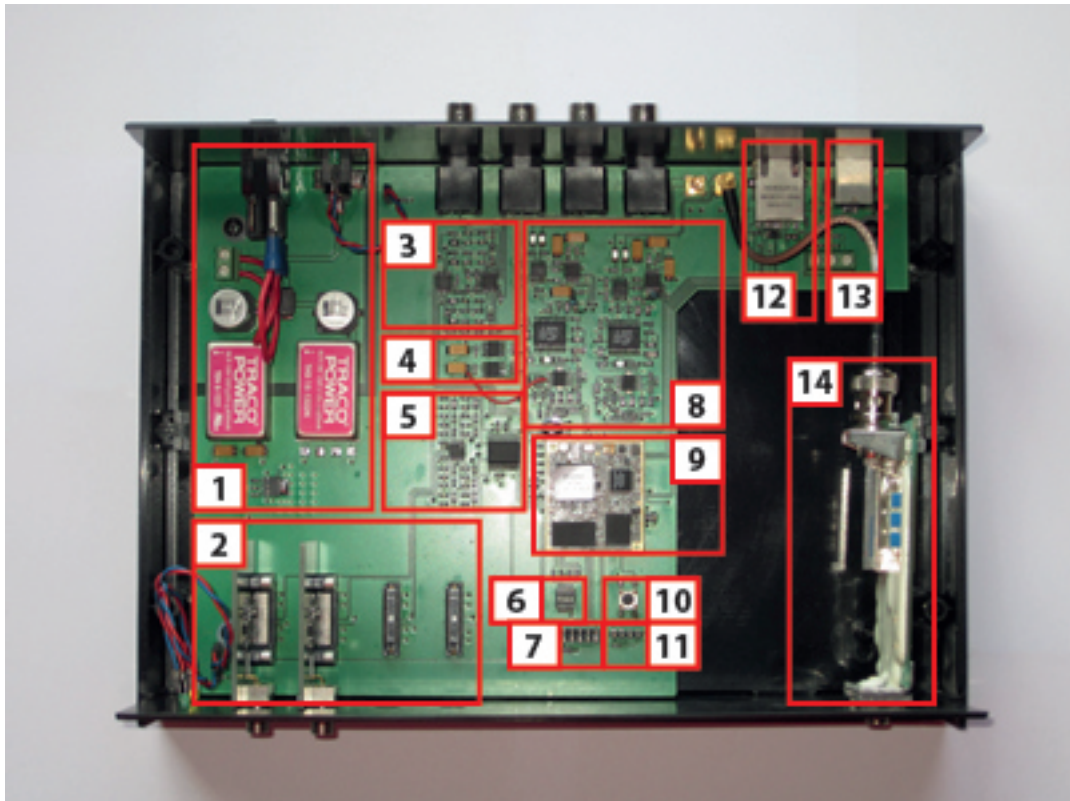
Des Weiteren wurden potentiell störende Bauelemente räumlich von empfindlichen Baugruppen auf der Platine getrennt. So wurden die DC/DC-Wandler an den Rand der Platine und der analoge Schaltungsteil mittig mit etwas Distanz zum Spannungsversorgungsteil auf der Platine platziert. Auch das Prozessormodul, welches aufgrund der Taktfrequenz von 600 MHz als potentiell strahlende Baugruppe gilt, wurde eher an den vorderen Rand der Platine platziert, während der analoge Schaltungsteil eher an den rückwärtigen Rand gesetzt wurde.

Der Lock-in Verstärker besitzt intern zwei verschiedene Massen. Eine, welche von außen, vom Desktopnetzteil stammt und eine geräteinterne Masse, welche das Bezugspotential für die DC/DC-Wandler darstellt. Diese zwei Massebereiche müssen ebenfalls räumlich voneinander getrennt sein und zwar mit einer Distanz, die mindestens für die Isolationsspannung der DC/DC-Wandler ausgelegt ist. In [18] wird für eine Spannung von mehr als 500 V ein Wert von 0,12 mil pro Volt angegeben, was bei 1500 V einen Abstand von 180 mil entspricht.

Auch der Abstand zwischen den einzelnen Modulationsplatinen untereinander sowie der Abstand von den Modulationsplatinen zu den APD-Modulen wurde maximiert, um zu verhindern, dass durch EMV-Strahlung intern eine Rückkopplung zwischen Aus- und Eingängen auftritt.

Die Modulationsplatinen wurden zweilagig, mit einer unteren Massefläche, sowie einer oberen Signal- und Versorgungsfläche, ausgeführt. Auch hier wurden nicht benutzte Flächen mit Masseflächen ausgefüllt.

Um die Verlustleistung, die der 3.3 V Spannungsregler (IC29) produziert, abführen zu können, wurden rund um dessen PIN6, über welchen laut Datenblatt [30] die Hauptabfuhr der Wärme geschieht, einige Durchkontaktierungen zwischen den Masseflächen gesetzt. Abbildung 3.26 veranschaulicht die Platzierung der einzelnen Baugruppen auf der Hauptplatine. Sehr gut ist auch zu erkennen, wie das Projekt in Hauptplatine und Modulationsplatinen aufgeteilt ist.



**Abbildung 3.26:** Platzierung der einzelnen Baugruppen auf der Hauptplatine

- |   |                            |
|---|----------------------------|
| 1. Spannungsversorgung                    | 8. Messsignalfad           |
| 2. Steckplätze für die Modulationsboards  | 9. Prozessormodul          |
| 3. Ausgangssignalfad                      | 10. Resettaster            |
| 4. Referenzspannungsquellen               | 11. UART Schnittstelle     |
| 5. Signalgenerator                        | 12. Ethernet Schnittstelle |
| 6. Schalter zur Einstellung des Bootmodes | 13. USB Schnittstelle      |
| 7. JTAG Schnittstelle                     | 14. APD Modul              |



## 3.2 Softwarerealisierung

Wie im Kapitel 2 über die Grundlagen zu dieser Arbeit bereits erwähnt wurde, kommt in diesem Lock-in Verstärker uClinux als Betriebssystem zum Einsatz. In einem ersten Schritt wurde die Einbindung der benötigten Hardwareressourcen in das Grundsystem untersucht und festgestellt, welche Treiber bereits zur Verfügung stehen, welche hiervon verwendet werden können, bzw. wo mit der Programmierarbeit angesetzt werden muss. Im zweiten Schritt folgte die Implementierung der benötigten Treibermodule sowie des Userprogramms, mit welchem der Benutzer den Lock-in Verstärker bedienen kann.

**Die Auswahl geeigneter Gerätetreiber:** Da in einem Linuxsystem der Hardwarezugriff stets über Treiberfunktionen erfolgen sollte, wurde bei der Softwarerealisierung zuerst nach entweder direkt von Analog Devices oder von der uClinux Community geschaffenen Treibern gesucht. Benötigt werden Treiber für die SPORT Schnittstellen, zur Ansteuerung der DACs und ADCs, sowie Treiber für die SPI, an welche die digitalen Potentiometer angeschlossen sind. Außerdem sind noch Treiber für die GPIO-PINs sowie für Ethernet und USB nötig. Während der Recherche stellte sich heraus, dass die Unterstützung von USB, Ethernet und den GPIO-PINs sehr gut funktioniert. Für die Ansteuerung der beiden digitalen Potentiometer gibt es zwei Möglichkeiten. Entweder die Verwendung des allgemeinen Treibers für die SPI, oder die Verwendung des Treibers, welcher von der Community speziell für das verwendete Potentiometer AD5293 entwickelt wurde. Im Zuge dieser Arbeit wurde der spezielle Treiber gewählt, da sich dadurch die Ansteuerung stark vereinfacht.

Die Ansteuerung der DACs und der ADCs hingegen gestaltete sich jedoch etwas schwieriger. Es existiert zwar ein Treiber für den DAC AD5064-1, allerdings steuert dieser Treiber den Baustein über die SPI Schnittstelle an. Da der AD5064-1 jedoch an einem SPORT hängt, ist dieser Treiber leider nicht zu gebrauchen. Unglücklicher Weise existieren für den DAC AD7304 sowie die ADCs AD7688 überhaupt keine Treiber. Nun könnte man auf den Treiber für die SPORT Schnittstellen ausweichen. Jedoch stellte sich bei den Tests heraus, dass auch dieser Treiber nicht brauchbar ist, da mit diesen Treibern immer nur eine bestimmte Byteanzahl auf einmal gesendet werden kann. Nachdem diese Bytes gesendet wurden, wird die komplette SPORT Schnittstelle vom Treiber deaktiviert. Somit ist jedes mal, wenn Daten über die SPORTs gesendet werden sollen, eine CPU Interaktion inklusive Kontextwechseln zwischen Userspace und Kernspace

notwendig. Dies ist für den Lock-in Verstärker inakzeptabel, da z.B. beim Sinusgenerator laufend Daten ohne Verzögerung gesendet werden müssen und dies am besten ohne, bzw. mit möglichst wenig CPU-Beteiligung. Auch für das Einlesen der Messwerte sowie das Ausgeben der berechneten Werte ist der vorhandene Treiber unbrauchbar. Die Berechnungen des Algorithmus müssen im Userspace ablaufen, da nur dort die Signalverarbeitungsbibliothek (libbfdsp) von Analog Devices verwendet werden kann. Diese Berechnungen sowie das Einlesen und Ausgeben der Werte müssen synchron zueinander geschehen. Somit sind spezielle Kommunikationsmechanismen zwischen Userspace und Kernspace notwendig, welche der standard SPORT Treiber nicht unterstützt. Deshalb ist es notwendig eine Lösung mit selbst entwickelten Treibern zu erarbeiten.

**Die Wahl der Samplingfrequenzen:** Bei der Anforderungsspezifikation an die Treibermodule sieht man sich unweigerlich mit der Frage der Wahl der Samplingfrequenzen konfrontiert. Einerseits Diejenige, mit welcher die Werte für die Modulation der Leuchtintensitäten ausgegeben werden, und andererseits diejenigen Samplefrequenzen, mit denen die Messsignale, welche die APD Module liefern, aufgezeichnet und die berechneten Signale wieder ausgegeben werden.

Im ersten Fall wird die Samplingfrequenz mit 125 kHz festgelegt, um saubere Sinussignale ausgeben zu können. Im zweiten Fall gilt es zunächst die Frequenz zu wählen, mit der die einkommenden Signale von den APD Modulen abgetastet werden. Diese Frequenz wurde möglichst gering gewählt, um weniger Samples aufzuzeichnen und damit bei der Verarbeitung Zeit zu sparen. Der nach unten hin limitierende Faktor ist in diesem Fall der hardwaremäßig ausgeführte Antialiasingfilter im Messsignalpfad, dessen Grenzfrequenz bei 15 kHz liegt. Dies bedeutet, dass die Samplingfrequenz laut Oppenheim und Schäfer [27] im optimalen Fall mindestens doppelt so hoch sein sollte. Hier wurde die Eingangssamplingfrequenz mit 48 kHz mehr als doppelt so groß gewählt, um nicht mit der Samplingfrequenz im Grenzbereich zu liegen. Bei der Wahl der Ausgangssamplingfrequenz gilt es mehrere Bedingungen zu beachten. Im Zuge der Berechnung des Lock-in Algorithmus kommt ein digitaler Tiefpassfilter zum Einsatz, dessen Grenzfrequenz im Hertzbereich liegt. Um diese Filtercharakteristik zu erreichen muss auch die Samplingfrequenz des Filters entsprechend niedrig gewählt sein. Die Eingangssamplingfrequenz ist hier eindeutig zu hoch, weswegen unbedingt downgesampelt werden muss. Um diesen Downsamplingvorgang zu vereinfachen wird mit sechs ein ganzzahliger Divisor gewählt. Somit ändert sich auch die Ausgangsamplere-

quenz um den Divisor sechs. Da das Spektrum des zu erwartenden Nutzsignals sich im niederen Hertzbereich befindet, ist die Ausgangssamplefrequenz von 8 kHz ausreichend hoch genug. Diese Frequenz kann auch mit den zur Verfügung stehenden Teilern optimal von der Systemclock abgeleitet werden. Der hardwaremäßig ausgeführte Rekonstruktionsfilter im Ausgangssignalfeld weist eine Grenzfrequenz von 5 kHz auf. Die Ausgangssamplingfrequenz liegt somit mit 8 kHz nicht im optimalen Bereich, dieser Umstand ist jedoch tolerierbar.

### 3.2.1 Kernspace: Treibermodule für SPORT0 und SPORT1

**SPORT1 - Der Sinusgenerator (Treibermodul AD7304mod):** Das Treibermodul AD7304mod, welches als Character Device Driver ausgelegt ist, regelt die Sinusausgabe über den SPORT1. Dies sind diejenigen Signale, welche über Lichtwellenleiter in das zu beobachtende Gewebe geleitet werden. Das Treibermodul wurde so konzipiert, dass die Signalausgabe mit Hilfe des DMA Controllers erfolgt. Das heißt, dass bei der Planung besonders Wert darauf gelegt wurde, dass im laufenden Betrieb nur ein minimaler Eingriff der CPU notwendig ist, um die benötigten Sinuswerte auszugeben.

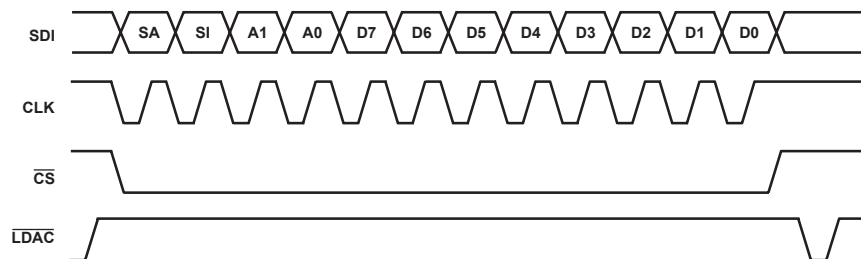
Beim Initialisieren des Moduls, also wenn es vom Betriebssystem geladen wird, wird automatisch die Routine `sport_init_module` aufgerufen. Innerhalb dieser Routine wird ein Linux Character Device angelegt. Öffnet ein Userspaceprogramm dieses Device, so erfolgt über die Fileoperation `sport_open` zunächst die Reservierung der benötigten Ressourcen wie die SPORT Leitungen, der DMA Kanal und der GPIO PIN PF15, welcher den CLR Eingang des DACs AD7304 (IC10) steuert. Im Anschluss daran wird der SPORT konfiguriert. Hierbei wird das Konfigurationsregister `SPORT_TCR1` so belegt, dass die entstehende Signalform der im Datenblatt des AD7304 [6] geforderten entspricht:

- `TCKFE=1`: Daten und interne Framesyncs werden mit der steigenden Flanke von `TSCLK` ausgegeben
- `LATFS=1`: Late Framesyncs werden verwendet
- `LTFS=1`: Active LOW Transmit Framesync
- `DITFS=1`: Framesyncs werden unabhängig von anliegenden Daten generiert (notwendig, weil Framesyncs mit fester Frequenz ausgegeben werden sollen)

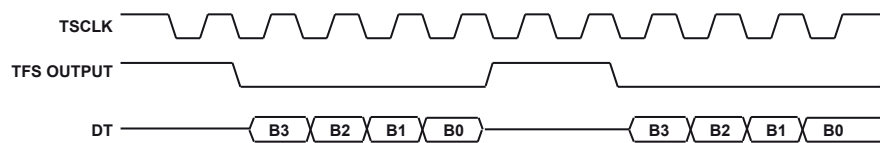
- TFSR=1: Framesync wird für jedes Datenwort benötigt
- ITFS=1: Framesyncsignale werden intern generiert
- TLSBIT=0: MSB wird zuerst übertragen
- TDTYPE=00: Keine Kompression der Daten
- ITCLK=1: Das Clocksignal wird intern generiert
- TSPEN=0: Der SPORT1 Transmitkanal ist zum Konfigurationszeitpunkt noch abgeschaltet

Das SPORT\_TCR2 wird mit 0x000B konfiguriert, was bedeutet, dass der sekundäre Transmitkanal des SPORT1 abgeschaltet ist und dass eine Wortlänge von 12 Bit verwendet wird.

Zur Veranschaulichung des geforderten Timings vom AD7304 bzw. des Timings, welches der SPORT1 Controller nach erfolgter Konfiguration ausgibt, ist dieser Zusammenhang in Abbildung 3.27 dargestellt. Des Weiteren wird innerhalb dieser Routine



(a) Timingdiagramm des AD7304 (entnommen und modifiziert aus [6])



(b) Timingdiagramm des konfigurierten SPORT Controllers (entnommen und modifiziert aus [1])

**Abbildung 3.27:** Kommunikationstimings zwischen AD7304 und dem SPORT1 Controller

die Frequenz der Transmittclock sowie der Framesyncimpulse eingestellt. Dies geschieht über die beiden Divider SPORT\_TCLKDIV und SPORT\_TFSDIV. Sie sind voneinander abhängig und werden von der Systemclock SCLK abgeleitet. SCLK wird im

Bootloader mit 120 MHz festgelegt. Das Gesamtkonzept sieht vor, dass die Sinuswerte mit 125 kHz ausgegeben werden. Da vier Sinusschwingungen erzeugt und die Werte seriell gesendet werden, müssen diese folglich mit der vierfachen Geschwindigkeit, also mit 500 kHz, übertragen werden. Aus diesen Angaben lässt sich nun ermitteln, dass eine TSCLK Frequenz von 7.5 MHz und ein SPORT\_TFSDIV Teiler von 14 notwendig sind, um die gewünschte Framesyncfrequenz zu erreichen. Die beiden Gleichungen 3.19 und 3.20 helfen dabei, die notwendigen Teiler zu berechnen.

$$SPORT\_TCLKDIV = \frac{SCLK}{2 \cdot (TSCLK\text{Frequenz})} - 1 = \frac{120\text{ MHz}}{2 \cdot 7.5\text{ MHz}} - 1 = 7 \quad (3.19)$$

Die maximale Clockfrequenz wird durch die minimale Periodendauer von 110 ns im Datenblatt des AD7304 [6] bestimmt. Dies bedeutet, dass TSCLK maximal 9.091 MHz betragen darf. Ein TSCLK von 7.5 MHz liegt somit voll im Rahmen des Datenblattes.

$$SPORT\_TFSDIV = \frac{TSCLK\text{Frequenz}}{(TFSFrequenz)} - 1 = \frac{7.5\text{ MHz}}{500\text{ kHz}} - 1 = 14 \quad (3.20)$$

Das Userspaceprogramm hat nun zwei Möglichkeiten die Funktionsweise des Treibers zu beeinflussen. Diese zwei Möglichkeiten sind das Aktivieren sowie das Deaktivieren der Sinusausgabe. Beide Befehle sind über IOCTL-Kommandos realisiert, welche in der Funktion `sport_ioctl` über eine Switch-Anweisung interpretiert werden. Zum Einschalten der Sinusausgabe muss das Userprogramm dem Treiber innerhalb des IOCTL-Befehls ein Struct vom Typ `sine_container` (siehe Codeabschnitt 3.1) übergeben. Dieses Struct enthält ein Array mit den Sinuswerten, welche ausgegeben werden sollen, sowie die Länge ebendieses Arrays. Das Array `sinetable` muss so aufgebaut sein, dass jeweils die Werte für eine Millisekunde eines Sinussignals hintereinander stehen. Diesen Aufbau verdeutlicht Abbildung 3.28.

**Codeabschnitt 3.1:** Form der Struktur `sine_container` in welcher die Sinuslookuptabelle gespeichert wird

---

```

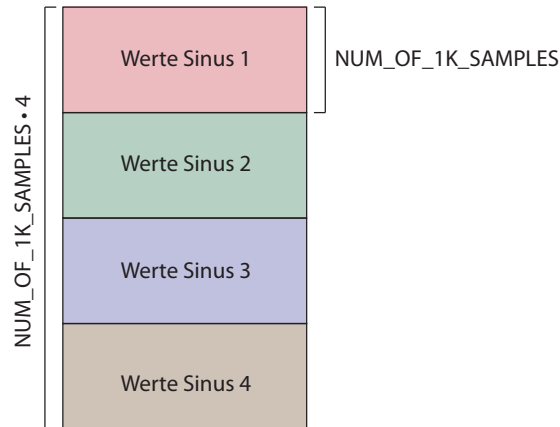
1 struct sine_container {
2     size_t num_of_sin_bytes;
3     uint16_t *sinetable;
4 };
5 \caption{}

```

---

Wird der Befehl zum Einschalten der Sinusausgabe, das IOCTL Kommando

`AD7304_IOC_ENABLE` vom Userspace an den Treiber gesendet, so reserviert dieser zunächst genügend Kernspeicher, kopiert anschließend die Tabelle mit den Sinuswerten in diesen Speicherbereich und startet über die Funktion `sport_enable` die Ausgabe der Sinuswerte.



**Abbildung 3.28:** Format der Sinus Lookuptabelle

`sport_enable` konfiguriert zu Beginn den DMA Controller so, dass eine vollautomatische Ausgabe mit minimaler CPU-Intervention erfolgt. Hierfür wird die 2D-Fähigkeit des DMA Controllers im Autobuffermodus ausgenutzt, was im Prinzip bedeutet, dass das übergebene Sinusarray mit Hilfe von zwei ineinander verschachtelten Schleifen abgetastet und die Werte an den entsprechenden Stellen des Arrays vom DMA Controller an den SPORT Controller gesendet werden, welcher diese dann letztendlich ausgibt. Die innere Schleife hat die Schrittweite (`xmodify`) von einem Viertel der Gesamtlänge des Arrays, also der Anzahl der Werte eines Sinussignals. So zeigt, aufgrund der Startadresse, die bei der Konfiguration dem DMA Controller übergeben wurde, der Adresszeiger des DMA Controllers zunächst auf die Speicherzelle des ersten Werts des Arrays, d.h. der erste Wert des ersten Sinus wird ausgegeben. Anschließend wird der Adresszeiger um `xmodify` erhöht, er zeigt somit auf die Speicherzelle, welche den ersten Wert des zweiten Sinus enthält. Dies wird vier mal (`xcount`) ausgeführt, sodass jeweils ein Wert jedes Sinus ausgegeben wird. Nun wird der Adresszeiger mit Hilfe der zweiten Schleife wieder zurückgesetzt. Und zwar auf die Speicherzelle, welche den zweiten Wert des ersten Sinus beinhaltet (`ymodify`). Dieses Rücksetzen geschieht so oft, bis sämtliche Werte eines Sinus abgearbeitet wurden, also ein Viertel der Länge des Gesamtarrays. Durch Aktivierung des Autobuffermodus beginnt nun der DMA Controller automatisch diese Schritte erneut von vorne auszuführen.

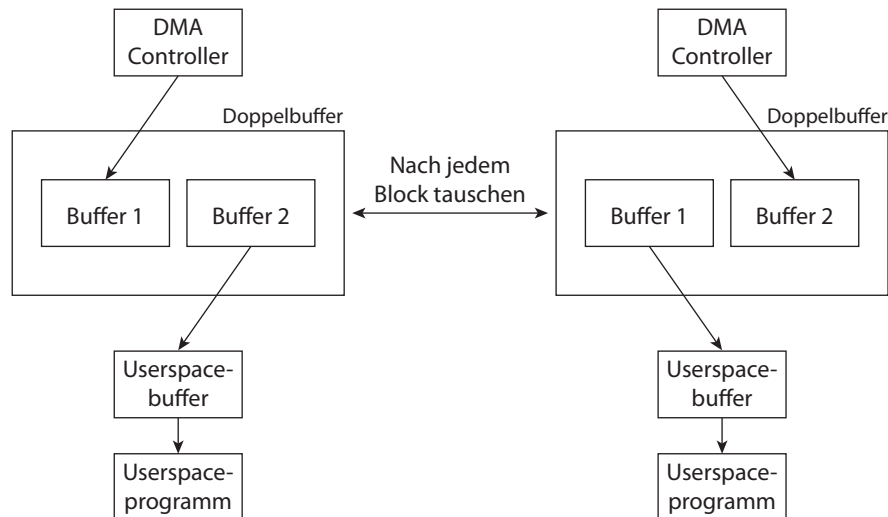
Nach der Konfiguration werden DMA- und SPORT-Ausgabe aktiviert, wobei laut Datenblatt des Signalprozessors darauf geachtet werden muss, erst den DMA Controller und dann den SPORT zu starten.

Mit Hilfe des IOCTL-Kommandos *AD7304\_IOC\_DISABLE* kann ein Userspaceprogramm die Sinusausgabe wieder stoppen. Treiberintern wird hier zunächst die SPORT Schnittstelle und anschließend der DMA Controller abgeschaltet.

### **SPORT0 - Messung und Ausgabe der Signale (Treibermodul AD7688mod):**

Das zweite Treibermodul, welches entwickelt wurde, steuert das Einlesen der Werte von den beiden ADCs (IC9 und IC18), sowie das Ausgeben der berechneten Werte über den DAC AD5064-1 (IC23). Alle der genannten Bauteile hängen, wie schon in der Hardwarebeschreibung besprochen, am SPORT0, weshalb die Funktionalität des SPORT0 in ein eigenes Treibermodul, mit dem Namen AD7688mod, gekapselt wurde. Da für die Berechnung des Lock-in Algorithmus bestimmte Mathematikbibliotheken wie z.B. die Signalprocessing Bibliothek *libbfdsp* von Analog Devices benötigt werden, diese jedoch nicht im Kernspace funktionieren, muss die Berechnung des Algorithmus im Userspace ablaufen. Aus diesem Grund sind besondere Mechanismen notwendig, welche eine Kommunikation und den schnellen Datenaustausch zwischen Kernspace und Userspace ermöglichen, ohne dass durch die Verzögerungen, welche beim Kontextwechsel entstehen, eine Beeinträchtigung der Messung erfolgt. Dieses Problem wurde letztendlich mit Hilfe blockweiser Datenverarbeitung, sowie durch die Verwendung von Doppelbuffern und Signalen gelöst. Durch die Blockweise Verarbeitung werden immer eine gewisse Anzahl an Inputsamples gespeichert und dem Userspaceprogramm als Block zur Verarbeitung überlassen. Dieses wendet den Lock-in Algorithmus immer auf den ganzen Datenblock an und liefert den Ergebnisdatenblock wieder an das Kernelmodul zurück. Dieses gibt abschließend die erhaltenen Daten aus. So sind Kontextwechsel (zwischen Userspace und Kernspace) nur zu Beginn und am Ende der Verarbeitung jedes Datenblocks notwendig. Bei sampleweiser Verarbeitung hingegen, wären bei jedem Sample zwei Kontextwechsel nötig. Doppelbufferung, wie sie Gan und Kuo in [19] beschreiben, macht die verzögerungsfreie blockweise Verarbeitung erst richtig möglich. Grundprinzip ist hierbei, dass ein Buffer verwendet wird, der doppelt so groß ist, wie der zu verarbeitende Block (siehe Abbildung 3.29). Während das Kernelmodul in die eine Hälfte schreibt, kann das Userspaceprogramm gleichzeitig aus der anderen Hälfte lesen. Ist die eine

Hälfte vollgeschrieben, werden die beiden Bufferhälften gewechselt. Das Userspaceprogramm liest aus der ersten Hälfte und das Kernelmodul schreibt in die zweite Hälfte.



**Abbildung 3.29:** Prinzip des Ping-Pong Bufferings (Doppelbufferung)

Nun muss noch eine Lösung gefunden werden, wie der Kerneltreiber dem Userspaceprogramm mitteilen kann, dass neue Daten im Doppelbuffer liegen, bzw. dass die Doppelbufferhälften gewechselt werden müssen. Hier bietet sich das Konzept der „Asynchronous Notification“ mit Hilfe von Signalen an. Signale sind, wie Kerrisk in [22] erläutert, eine Möglichkeit von Inter-Prozess-Kommunikation. Sie arbeiten wie Software Interrupts. Jedoch nicht nur Prozesse können sich untereinander Signale senden. Auch Treibermodule können hiermit ein bestimmtes Userspaceprogramm über Aktionen benachrichtigen. Der Standardablauf des Userspaceprogramms wird sofort unterbrochen und ein Interrupthandler ausgeführt. Sobald dieser abgearbeitet ist, läuft das Programm wieder normal weiter.

Wie das vorher besprochene Modul AD7304mod legt auch AD7688mod bei der Initialisierung durch das Betriebssystem ein Characterdevice an, welches wiederum vom Userspaceprogramm mit IOCTLs gesteuert werden kann. Die Möglichkeiten die AD7688mod zur Verfügung stellt sind folgende:

- *AD7688\_ENABLE\_READING*: Startet das Einlesen von den ADCs. Gleichzeitig wird die Ausgabe über den DAC IC23 aktiviert.
- *AD7688\_DISABLE\_READING*: Beendet das Einlesen und Ausgeben der Daten.



Zuvor muss jedoch dieses Characterdevice vom Userprogramm geöffnet werden. Im Treibermodul wird in diesem Fall die entsprechende Fileoperation **sport\_open** aufgerufen. Da das Einlesen von den ADCs bzw. das Ausgeben über den DAC mit möglichst wenig CPU-Belastung erfolgen soll, werden auch in diesem Modul die Fähigkeiten des DMA Controllers genutzt. Im Gegensatz zum Treibermodul AD7304mod wird im AD7688mod Modul ein DMA-Interrupt verwendet, welcher jedes Mal ausgelöst wird, wenn ein Block an Daten fertig von den ADCs ausgelesen wurde. Damit DMA eingesetzt werden kann, müssen zunächst die benötigten Ressourcen, also SPORT0-RX und -TX DMA-Kanal, sowie der DMA-RX-Interrupt, wie das nachfolgende Codelisting (Codeabschnitt 3.2) zeigt, beim Betriebssystem angefordert werden.

**Codeabschnitt 3.2:** Anforderung der DMA Ressourcen beim Betriebssystem uClinux

---

```

1 // Request tx dma
2 ret = request_dma(dev->dma_tx_chan, "sport_tx_dma_chan");
3 if (ret) {
4     pr_err("unable to request sport tx dma channel\n");
5     return ret;
6 }
7
8 // Request rx dma
9 ret = request_dma(dev->dma_rx_chan, "sport_rx_dma_chan");
10 if (ret) {
11     pr_err("unable to request sport rx dma channel\n");
12     return ret;
13 }
14
15 //set rx dma irq handler
16 set_dma_callback(dev->dma_rx_chan, dma_rx_irq_handler, dev);
17
18 ret = peripheral_request_list(sport_req, DRVNAME);
19 if (ret) {
20     pr_err("requesting peripherals failed\n");
21 }

```

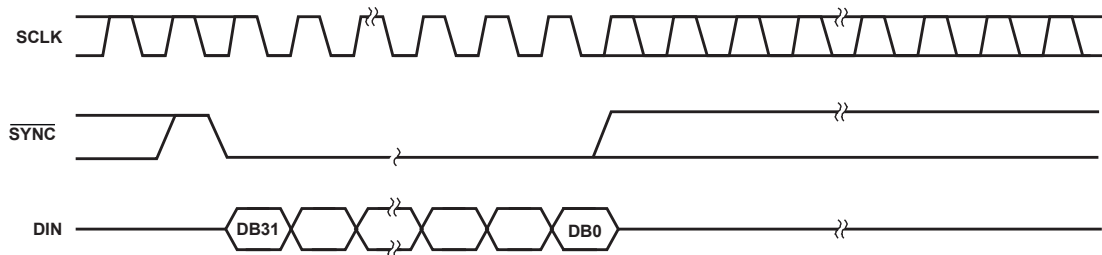
---

Des Weiteren wird der GPIO PIN PF14 beim Betriebssystem angefordert, welcher den CLR PIN des DACs IC23 steuert. Anschließend erfolgt die Konfiguration der SPORT0 Schnittstelle, damit die Ausgabetimes der Schnittstelle an die geforderten Timings

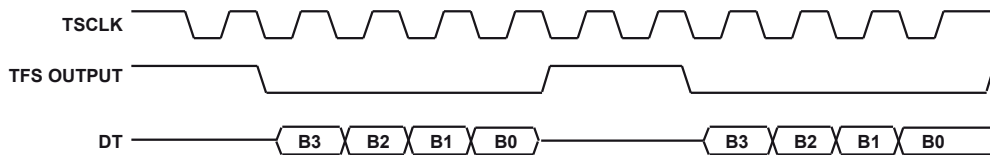
vom DAC angepasst werden (siehe Abbildung 3.30). Hierfür wird das Konfigurationsregister `SPORT_TCR1` wie folgt belegt:

- `TCKFE=0`: Daten und interne Framesyncs werden mit der fallenden Flanke von `TSCLK` ausgegeben
- `LATFS=1`: Late Framesyncs werden verwendet
- `LTFS=1`: Active LOW Transmit Framesync
- `DITFS=1`: Framesyncs werden unabhängig von anliegenden Daten generiert (notwendig, weil Framesyncs mit fester Frequenz ausgegeben werden sollen)
- `TFSR=1`: Framesync wird für jedes Datenwort benötigt
- `ITFS=1`: Framesyncsignale werden intern generiert
- `TLSEBIT=0`: MSB wird zuerst übertragen
- `TDTYPE=00`: Normale Operation; Keine Komprimierung der Daten
- `ITCLK=1`: Das Clocksignal wird intern generiert
- `TSPEN=0`: Der Transmittkanal des `SPORT0` ist zum Konfigurationszeitpunkt noch abgeschaltet

Das `SPORT_TCR2` Konfigurationsregister erhält den Wert `0x001F`, was bedeutet, dass der sekundäre Transmittkanal abgeschaltet ist und eine Wortlänge von 32 Bit verwendet wird, da nur ein `AD5064-1` vorhanden ist, welcher 32 Bit lange Wörter annimmt. Nun werden die Frequenz der Transmittclock (`TCLK`) und der Ausgaberate, mit der das Framesyncsignal, also das Chipselect für den DAC, ausgegeben wird, berechnet und eingestellt. Die Formeln dafür unterscheiden sich nicht von der Berechnung, welche in den beiden Gleichungen 3.19 und 3.20 angestellt wurden. Da sich jedoch die Ausgangsparameter verändert haben, ändern sich auch die berechneten Werte dementsprechend. Die Ausgabefrequenz pro Kanal wurde mit 8 kHz festgelegt, was für die Schnittstelle schlussendlich eine Framesyncfrequenz (TFS Frequenz) von  $4 \cdot 8 \text{ kHz} = 32 \text{ kHz}$  bedeutet. Somit sind eine Transmittclockfrequenz von 2.4 MHz, ein `SPORT_TCLKDIV` von 24 und ein `TFS_DIV` von 74 notwendig, um eine Framesyncfrequenz von 32 kHz zu



(a) Timingdiagramm des AD5064-1 (entnommen und modifiziert aus [4])



(b) Timingdiagramm des konfigurierten SPORT0 Controllers (entnommen und modifiziert aus [1])

**Abbildung 3.30:** Kommunikationstimmings zwischen AD5064-1 und dem SPORT0 Controller

erreichen. Mit Gleichungen 3.21 sowie 3.22 lassen sich die benötigten Frequenzteiler berechnen.

$$SPORT\_TCLKDIV = \frac{SCLK}{2 \cdot (TSCLK\text{Frequenz})} - 1 = \frac{120\text{ MHz}}{2 \cdot 2.4\text{ MHz}} - 1 = 24 \quad (3.21)$$

$$SPORT\_TFSDIV = \frac{TSCLK\text{Frequenz}}{(TFS\text{Frequenz})} - 1 = \frac{2.4\text{ MHz}}{100\text{ kHz}} - 1 = 74 \quad (3.22)$$

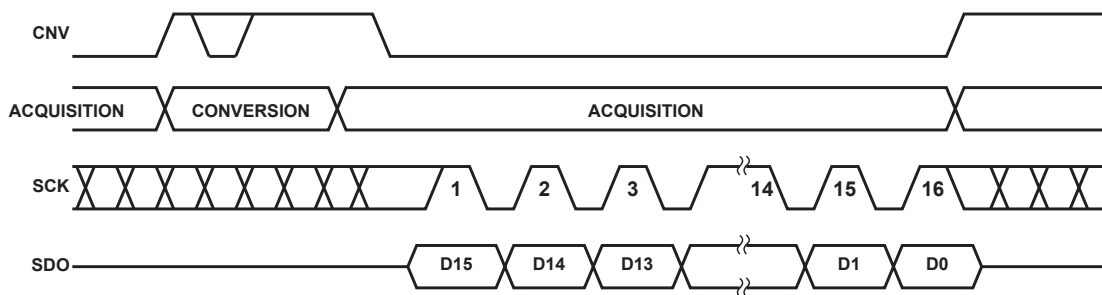
Laut Datenblatt des AD5064-1 [4] darf die Clockfrequenz maximal 50 MHz betragen. Mit 2.4 MHz liegt die TSCLK Frequenz somit im korrekten Frequenzbereich.

Zusätzlich zum Transmitkanal müssen bei diesem Treibermodul auch noch der Recievekanal des SPORT0 konfiguriert werden, da hierüber die Werte von den beiden ADCs eingelesen werden. Die entsprechenden Register sind SPORT\_RCR1 sowie SPORT\_RCR2. Das Register SPORT\_RCR1 wird wie folgt belegt:

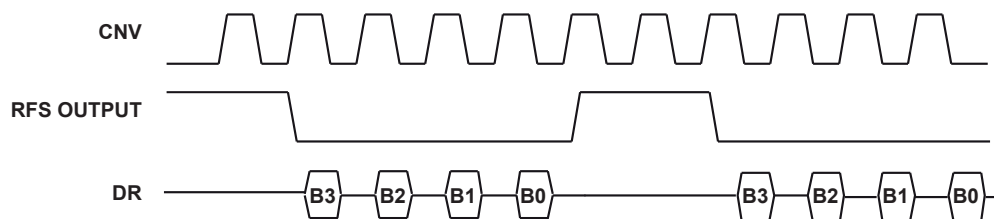
- RCKFE=1: Interne Framesyncs werden mit der fallenden Flanke von RSCLK ausgegeben, Daten werden mit der steigenden Flanke von RSCLK gesampelt
- LARFS=1: Late Framesyncs werden verwendet
- LRFS=1: Active LOW Transmit Framesync

- RFSR=1: Framesync wird für jedes Datenwort benötigt
- IRFS=1: Framesyncsignale werden intern generiert
- RLSBIT=0: MSB wird von den ADCs zuerst übertragen also vom SPORT0 zuerst empfangen
- RDTYPE=00: Normale Operation; Keine Komprimierung der Daten
- IRCLK=1: Das Clocksignal wird intern generiert
- RSPEN=0: Der Receivekanal des SPORT0 ist zum Konfigurationszeitpunkt noch abgeschaltet

Das Register SPORT\_RCR2 wird mit 0x010F initialisiert, was bedeutet, dass beide Receivekanäle aktiv sind und die Wortlänge 16 Bit beträgt, d.h. es können 16 Bit lange Werte von beiden ADCs empfangen werden. Abbildung 3.31 zeigt, wie die Signaltimings, welche der AD7688 sendet, aussehen und wie der SPORT0 Controller konfiguriert ist, um die Werte zu empfangen.



(a) Timingdiagramm des AD7688 (entnommen und modifiziert aus [9])



(b) Timingdiagramm des konfigurierten SPORT0 Controllers (entnommen und modifiziert aus [1])

**Abbildung 3.31:** Kommunikationstimmings zwischen den AD7688 und dem SPORT0 Controller

Abschließend wird nun noch die Frequenz der Receiveclock sowie des Receive Framesyncsignals eingestellt. Die Samplefrequenz für die beiden ADCs wurde während des Designs mit 48 kHz festgelegt. Mit diesen Angaben lässt sich nun die Receiveclockfrequenz (RSCLK) sowie die beiden Teiler für die Frequenz der Receiveclock (SPORT\_RCLKDIV) und für die Framesync Frequenz (SPORT\_RFSDIV) bestimmen, die benötigt werden, um eine Framesyncrate von den gewünschten 48 kHz zu erreichen. So ergibt sich, dass bei einer Systemclockfrequenz von 120 MHz ein SPORT\_RCLKDIV von 9, somit eine Receiveclockfrequenz von 6 MHz und ein SPORT\_RFSDIV Teiler von 124 notwendig ist, um mit 48 kHz sampeln zu können. Die beiden Gleichungen 3.23 und 3.23 beschreiben die Berechnung dieser Frequenzteiler.

$$SPORT\_RCLKDIV = \frac{SCLK}{2 \cdot (RSCLKFrequenz)} - 1 = \frac{120 \text{ MHz}}{2 \cdot 6 \text{ MHz}} - 1 = 9 \quad (3.23)$$

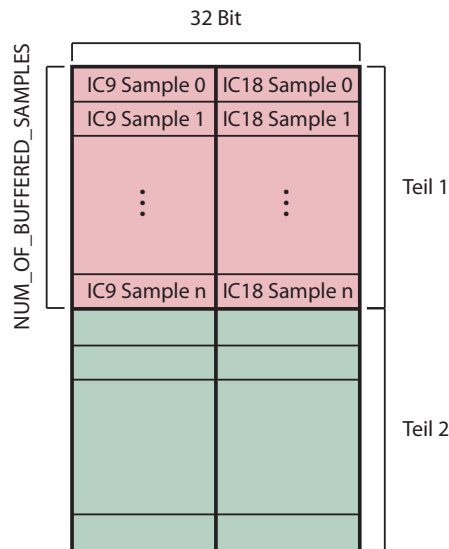
$$SPORT\_RFSDIV = \frac{RSCLKFrequenz}{(RFSFrequenz)} - 1 = \frac{6 \text{ MHz}}{48 \text{ kHz}} - 1 = 124 \quad (3.24)$$

Die maximale Clockfrequenz, die der AD7688 fähig ist zu verarbeiten, ist im Datenblatt [9] mit 66.667 MHz angegeben, was wiederum bedeutet, dass sich die RSCLK Frequenz mit 6 MHz genau im geforderten Bereich befindet.

Nachdem nun die SPORT0 Schnittstelle fertig konfiguriert ist, wird als nächstes in der Fileoperation **sport\_open** die Konfiguration der Doppelbuffer durchgeführt. Hierfür werden zwei Speicherbereiche, einer für das Einlesen der Daten von den ADCs und einer für das Ausgeben der Daten an den DAC IC23, im nicht gecachten DMA Speicherbereich reserviert. Wie in Abbildung 3.32 zu sehen ist, gliedert sich der Eingangsdoppelbuffer wieder in zwei Teile, zwischen denen immer gewechselt wird, wenn NUM\_OF\_BUFFERED\_SAMPLES Samples eingelesen wurden. Jeder dieser Teile ist NUM\_OF\_BUFFERED\_SAMPLES lang und 32 Bit breit. Dies hat seine Begründung darin, dass im SPORT Eingangsbuffer die 16 Bit langen Daten vom primären und sekundären Receivekanal genau hintereinander stehen. Somit können mit einer 32 Bit langen DMA Abfrage die Daten von beiden ADCs auf einmal abgeholt und in den Eingangsdoppelbuffer geschrieben werden.

Abbildung 3.33 verdeutlicht die Struktur des Ausgangsdoppelbuffers. Er gliedert sich auch in zwei Teile, welche jeweils vier mal NUM\_OF\_DAC\_SAMPLES lang sind. Innerhalb jedes dieser zwei Teile müssen die Werte der einzelnen DACs in abwechselnder

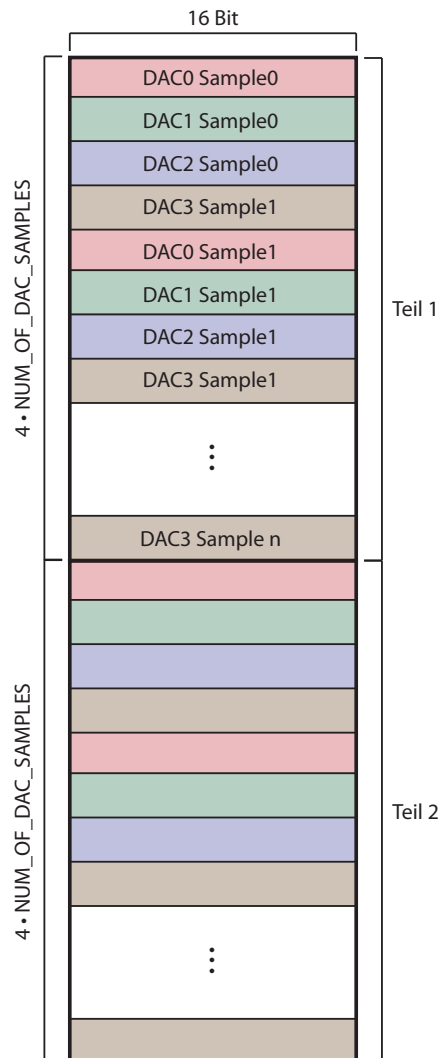
Reihenfolge stehen. Somit ist eine einfache Ausgabe mit Hilfe des DMA Controllers möglich.



**Abbildung 3.32:** Speicherbelegung des Eingangsdoppelbuffers

Als letzter Konfigurationsschritt wird nun noch der DMA Controller für das korrekte Einlesen und Ausgeben der Daten eingestellt. Beim Einlesen wird der zweidimensionale Autoflow Modus verwendet, was dafür sorgt, dass der DMA Controller `NUM_OF_BUFFERED_SAMPLES` Werte in den ersten Teil des Eingangsdoppelbuffers einliest (über `x_count` und `x_modify` konfiguriert), anschließend auf den zweiten Teil des Eingangsdoppelbuffers umschaltet (über `y_count` und `y_modify` konfiguriert) und wiederum `NUM_OF_BUFFERED_SAMPLES` einliest. Nachdem der zweite Teil des Eingangsdoppelbuffers vollgeschrieben ist, beginnt der DMA Controller aufgrund des konfigurierten Autoflow Modus, automatisch wieder an der Startadresse Werte zu schreiben. Weiters ist der DMA Controller über `WDSIZE_32` für das Einlesen von Werten so konfiguriert, dass er immer 32 Bit lange Wörter vom SPORT Controller abholt. Somit werden mit einem Einlesevorgang gleich die Werte von beiden ADCs eingelesen. Mit Hilfe der beiden Konfigurationsoptionen `DI_SEL` sowie `DI_LEN` wird erreicht, dass nach jedem Wechsel der Doppelbufferhälften ein Interrupt ausgelöst wird, was erforderlich ist, um die blockweise Verarbeitung der Daten zu starten. Auch beim Ausgeben von Werten an die Ausgangs DACs wird der zweidimensionale Autoflow Modus verwendet. Hier werden jedoch vier mal `NUM_OF_DAC_SAMPLES` Werte ausgegeben, bevor ein Wechsel der Ausgangsdoppelbufferhälften erfolgt. Bei der Ausgabe der Daten werden

keine Interrupts benötigt, weswegen bei der Konfiguration dieses DMA Kanals `DI_SEL` und `DI_EN` nicht gesetzt sind.



**Abbildung 3.33:** Speicherbelegung des Ausgangsdoppelbuffers

Über die Funktion `ad7688_ioctl` werden die IOCTL Befehle, welche das User-spaceprogramm sendet, ausgewertet. Möchte das Userprogramm über das Kommando `AD7688_ENABLE_READING` den Messvorgang starten, so werden zunächst die DMA Kanäle für SPORT0 transmit und receive gestartet und anschließend erst der SPORT0 Controller. Beim Stoppen des Messvorgangs mit `AD7688_DISABLE_READING` wird umgekehrt vorgegangen. Also zuerst der SPORT0 Controller und anschließend die entsprechenden DMA Kanäle deaktiviert.

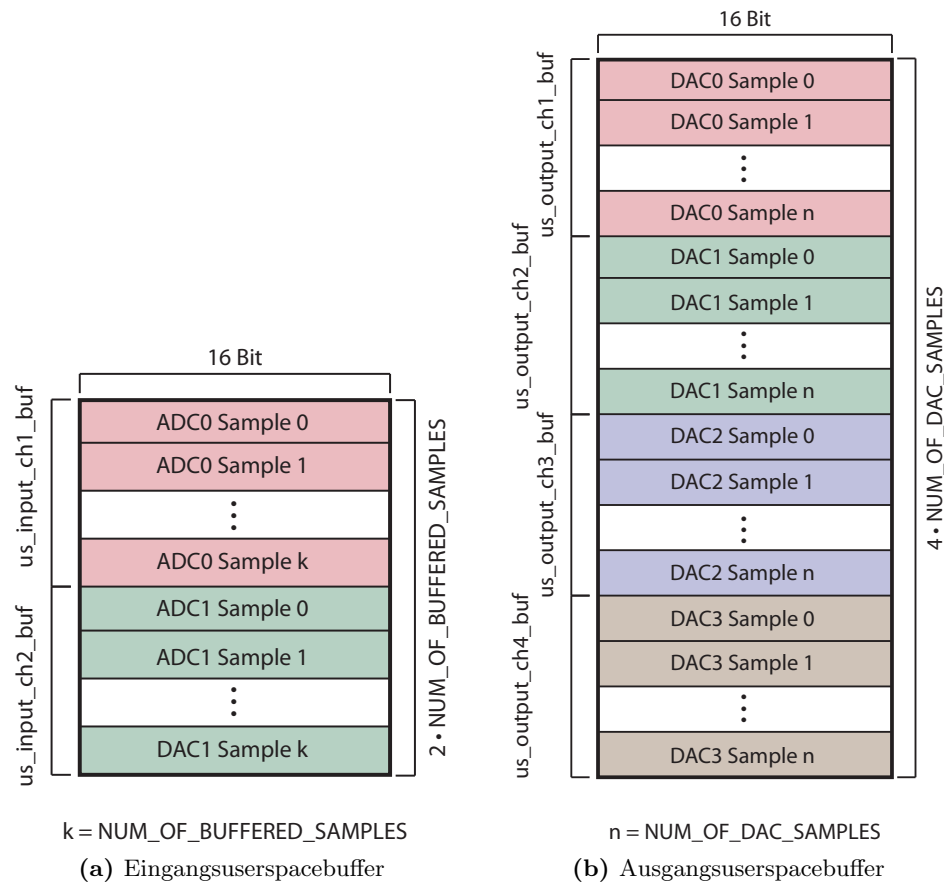
Zusätzlich zu Ein- und Ausgangsdoppelbuffer, welche im Kernspacebereich liegen, werden, wie Abbildung 3.29 zeigt, in diesem Modul noch zwei weitere Buffer verwendet. Diese beiden Userspacebuffer liegen in einem Speicherbereich nach **ramend\_**, welcher nicht vom Kernel verwaltet wird. Dieser Speicherbereich kann mit Hilfe eines Eintrags in den Bootparametern über die Kernelkonfiguration eingestellt werden. Die beiden Userspacebuffer werden mit Hilfe von Memorymapping dem Userspaceprogramm sowie dem Treibermodul zugänglich gemacht. Somit können beide auf dieselben Speicherzellen zugreifen. Dies hat den großen Vorteil, dass es nicht mehr notwendig ist, die Werte vor jeder Blockverarbeitung vom Kernspace in den Userspacespeicherbereich und umgekehrt zu kopieren, was wiederum die Verarbeitung ungemein beschleunigt. Der Einganguserspacebuffer ist zwei mal `NUM_OF_BUFFERED_SAMPLES` mal 16 Bit und der Ausganguserspacebuffer vier mal `NUM_OF_DAC_SAMPLES` mal 32 Bit lang. Die beiden Konstanten `NUM_OF_BUFFERED_SAMPLES` und `NUM_OF_DAC_SAMPLES`, welche im gesamten Treibermodul verwendet werden, sind im Headerfile `AD7688.h` definiert.

Die DMA-RX-Interrupts, welche beim Einlesen der Werte vom DMA Controller ausgelöst werden, werden in der Interruptserviceroutine **`dma_rx_irq_handler`** behandelt. Hier speichert die Routine die eingelesenen Werte vom Eingangsdoppelbuffer in den Einganguserspacebuffer und vom Ausganguserspacebuffer in den Ausgangsdoppelbuffer um, wobei die Reihenfolge der Werte verändert wird (vergleiche Abbildungen 3.32, 3.33 sowie 3.34) Dies ist notwendig, um im Userspace die blockweise Verarbeitung zu vereinfachen, bzw. um die korrekte Ausgabe der Werte per DMA zu ermöglichen. Abschließend wird in der Interruptserviceroutine das Signal `SIGIO` an das Userspaceprogramm gesendet um zu signalisieren, dass Werte zur Verarbeitung im Einganguserspacebuffer liegen.

### 3.2.2 Userspace: Das Hauptprogramm

Das Userspaceprogramm liegt in der Datei `lockin.c`. Die zugehörige Headerdatei ist `lockin.h`. Das Userspaceprogramm bietet dem Anwender eine einfache Konsolenumgebung, über welche der Lock-in Verstärker gesteuert werden kann. Möglich sind hier die Verstärkungseinstellung der beiden Eingangsverstärker sowie das Starten und Stoppen des Messvorganges. Eine nähere Beschreibung der Softwarebedienung erfolgt in Abschnitt 4.3.





**Abbildung 3.34:** Speicherbelegung des Ein- und des Ausgangsuserspacebuffers

Nach dem Start des Programms, wird in der **main** Routine zunächst festgelegt, welche Frequenzen (Trägerfrequenzen) der Lock-in Verstärker generieren soll. Dies geschieht über das Array **freq**. Es können hier Frequenzwerte von 1 kHz weg in 1 kHz Schritten eingegeben werden. Nach oben hin ist der mögliche Frequenzbereich durch die Grenzfrequenz des Rekonstruktionsfilters (40 kHz) begrenzt, weil, wie in Abschnitt 3.1.2 erläutert ist, die Bedingung  $\Omega_N < \Omega_C < (\Omega_S - \Omega_N)$  erfüllt bleiben muss. Der Wert muss in kHz eingegeben werden.

Anschließend werden diverse Initialisierungen und Einstellungen durchgeführt. So werden hier z.B. die IIR Filter, welche für die Berechnung des Lock-in Algorithmus benötigt werden, initialisiert, die SPORT Characterdevices geöffnet, das Memorymapping für die Userspacebuffer angestoßen, sowie der Schleiferzustand der beiden digitalen Potentiometer ausgelesen.

Nun folgt ein essentieller Konfigurationsabschnitt. Mit Hilfe der nachfolgenden Programmsequenz (Codeabschnitt 3.3) werden die benötigten Signale konfiguriert und für das Userspaceprogramm die asynchrone Benachrichtigung aktiviert.

**Codeabschnitt 3.3:** Konfiguration für das Handling der beiden Signale  
**ioact** und **stpact**

---

```

1 //configuration of Signals
2 sigemptyset(&ioact.sa_mask);
3 sigemptyset(&stpact.sa_mask);
4
5 ioact.sa_flags =0;
6 ioact.sa_handler = sigc;
7
8 stpact.sa_flags =0;
9 stpact.sa_handler = sigc1;
10
11 sigaction(SIGIO, &ioact, NULL);
12 sigaction(SIGTSTP, &stpact, NULL);
13
14 fcntl(input_channels[0].sport_fd, F_SETOWN, getpid());
15 int oflags = fcntl(input_channels[0].sport_fd, F_GETFL);
16 fcntl(input_channels[0].sport_fd, F_SETFL, oflags | FASYNC);

```

---

Wie in den Zeilen 11 und 12 des Listings zu sehen ist, werden hier zweierlei Signale verwendet. Das erste Signal ist SIGIO. Über dieses Signal benachrichtigt das AD7688mod Treibermodul das Userprogramm, dass ein neuer Datenblock zur Verarbeitung zur Verfügung steht. Der Handler für dieses Signal, die Funktion **sigc**, wird in Zeile sechs des Listings festgelegt. Das zweite Signal ist SIGTSTP, welches vom Betriebssystem gesendet wird, wenn der Benutzer die Tastenkombination „ctrl+z“ drückt. Dieses Signal wird dazu verwendet, den Messvorgang zu beenden. Der Handler für SIGTSTP, die Funktion **sigc1** wird in Zeile neun konfiguriert.

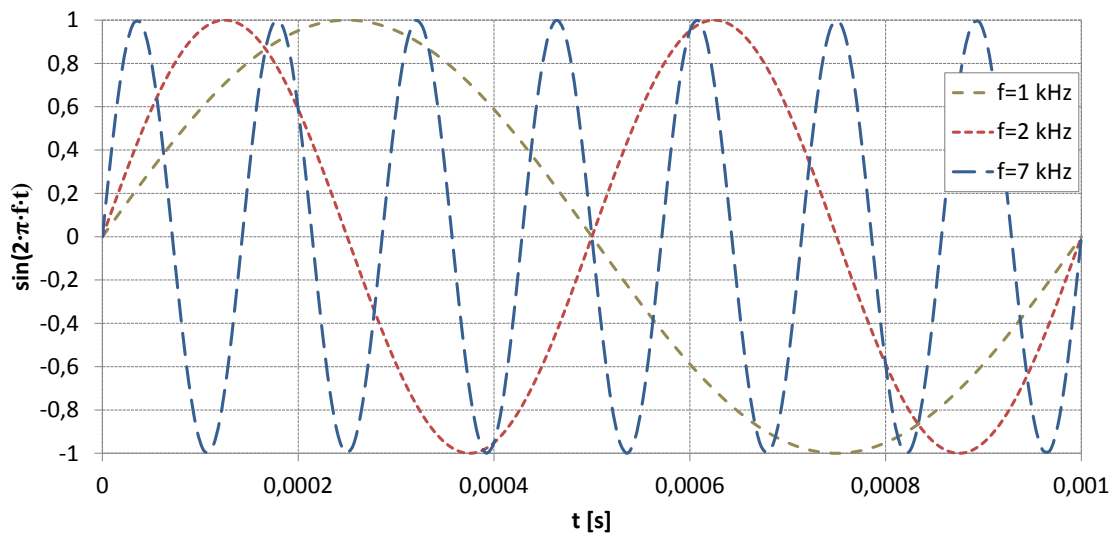
Die erste für den Benutzer sichtbare Aktion, welche das Userspaceprogramm tätigt, ist die Anzeige der Eingabeaufforderung für die Lock-in Shell, was bedeutet, dass auf eine Usereingabe gewartet wird. Um diese Eingabe zu interpretieren dient die Funktion **dispatch**. Wurde nun vom Benutzer mittels des Buchstabens „e“ und anschließendem Bestätigen per ENTER-Taste der Messvorgang aktiviert, wird zunächst mit Hilfe des IOCTL-Befehls *AD7304\_IOC\_ENABLE* dem SPORT1

Treibermodul mitgeteilt, die Sinusausgabe zu starten, und anschließend über den IOCTL-Befehl `AD7688_ENABLE_READING` dem SPORT0 Treibermodul signalisiert, dass Werte von den ADCs eingelesen und die berechneten Messwerte über die entsprechenden DACs ausgegeben werden sollen. Des Weiteren wird hier das Flag `measurement_` gesetzt, welches dazu da ist, um anzuzeigen, ob der Messvorgang zu starten oder zu stoppen ist. Anschließend begibt sich das Hauptprogramm in eine Warteschleife und wartet, laufend die Flags `measurement_` und `data_arrived_` pollend, darauf, dass ein Datenblock verarbeitet werden kann. Ist nun ein vollständiger Datenblock vorhanden, sendet das AD7688mod Treibermodul das Signal SIGIO. Dieses wird vom Userspaceprogramm gefangen und der entsprechende Handler, `sigc`, aufgerufen. Innerhalb dieses Handlers wird das Flag `data_arrived_` auf eins gesetzt, was wiederum das Hauptprogramm dazu veranlasst die Datenverarbeitungsroutine `processData` aufzurufen. Diese Routine berechnet nun über den gesamten Datenblock den Lock-in Algorithmus, speichert die berechneten Werte in den Ausgangsuserspacebuffer und setzt das Flag `data_arrived_` wieder auf null. Dies zwingt das Hauptprogramm erneut in die Warteschleife zu verfallen und auf den nächsten Datenblock zu warten. Diese Warteschleife kann nur durch Drücken der Tastenkombination „ctrl+z“ durchbrochen werden. Durch diese Benutzeraktion wird der Handler des Signals SIGTSTP aufgerufen, welcher das Flag `measurement_` rücksetzt und somit dem Hauptprogramm das Zeichen gibt, die Warteschleife zu verlassen und wiederum die Eingabeaufforderung der Lock-in Shell anzuzeigen.

**Generierung der Werte für die Modulationssignale:** Wie die eigentliche Ausgabe der Sinussignale, das Zusammenspiel zwischen Hard- und Software, funktioniert wurde schon bei der Besprechung für das Treibermodul AD7304mod behandelt. Bevor jedoch ein Sinussignal ausgegeben werden kann, müssen erst die entsprechenden Sinuswerte generiert werden. Um bei der Auswahl der Modulationsfrequenzen der Sinussignale flexibel zu sein, werden die Sinuswerte aller vier Modulationssignale beim Start des Hauptprogramms von einer 1 kHz Lookuptabelle abgeleitet. Diese lässt sich vorab mit Hilfe von Matlab und dem Skript `lookup_1k_external.m` berechnen und über das Headerfile `sine_1k_external.h` in das Hauptprogramm laden. Das erwähnte Matlabskript passt zuerst das 1 kHz Signal für die direkte Ausgabe über den DAC an, sampelt eine Millisekunde des Signals mit einer Samplerate von 125 kHz und quantisiert diese Werte mit 256 Quantisierungsstufen. Dies entspricht der acht Bit hohen Auflösung des AD-

Cs AD7304. Im Headerfile `sine_1k_external.h` stehen nun die Werte des Sinussignals in einem Array. Zusätzlich wird noch die Länge des Arrays in dieser Datei gespeichert.

Um nun die Werte für die Sinussignale mit der gewünschten Modulationsfrequenz zu erhalten wird jeweils der  $n$ -te Wert aus dem Array `sin1k_output_` ausgewählt und zwar immer Werte für 1 ms des Signals, wobei  $n$  der Frequenz in kHz entspricht. Wie Abbildung 3.35 veranschaulicht, treffen sich die ganzzahligen Sinusschwingungen nach 1 ms und  $n$  Perioden wieder bei 0 und können dadurch optimal Ausgegeben werden. Um ein 2 kHz Sinussignal zu erhalten wird somit jeder zweite Wert verwendet und das über zwei Perioden hinweg. Über einen zirkularen Index werden die entsprechenden Werte aus der 1 kHz Tabelle `sin1k_output_` ausgewählt.



**Abbildung 3.35:** Sinusschwingung mit  $f = 1$  kHz,  $f = 2$  kHz und  $f = 7$  kHz. Zu erkennen ist hier, wie sich die drei Schwingungen jeweils nach einer, zwei und sieben Perioden wieder bei 0 treffen.

Beim Start der Sinusausgabe mit dem IOCTL Kommando `AD7304_IOC_ENABLE` wird nun ein Array dem Kernelmodul `AD7304mod` übergeben, welches die Lookuptabellen der vier modulierten Signale enthält, die aus der DAC Lookuptabelle des Matlabskriptes abgeleitet wurden (siehe Abbildung 3.28).

**Generierung der Werte für die Sinussignale für interne Berechnungen:** Da sich die Samplerate, mit der die Modulationssignale ausgegeben werden (125 kHz), und die Samplerate, mit der das Signal von den APD-Modulen abgetastet wird (48 kHz), voneinander unterscheiden, müssen für die Berechnung des Lock-in Algorithmus Loo-

kuptabellen generiert werden, welche mit einer Abtastrate von 48 kHz gesampelt wurden. Dies erledigt das Matlabskript `lookup_1k_internal.m`. Innerhalb dieses Skripts werden Werte für eine Millisekunde eines Sinus- und eines Cosinussignals der vier gewünschten Modulationsfrequenzen, also 2 kHz, 3 kHz, 5 kHz und 7 kHz, gesampelt. Da die Berechnung des Lock-in Algorithmus in Floatingpointarithmetik erfolgt, werden auch die Sinuswerte mit „single precision floatingpoint,“ Genauigkeit quantisiert. Die entsprechenden Werte werden in Arrays im Headerfile `sine_internal.h` gespeichert, welches anschließend im Hauptprogramm über das Headerfile `lockin.h` eingebunden wird. Somit stehen die Lookuptabellen dem Lock-in Algorithmus zur Verfügung.

**Einstellen der Eingangsverstärkung:** Wie schon in Abschnitt 3.1.3 erwähnt, findet die Einstellung der Eingangsverstärkung durch Veränderung der Schleiferposition eines digitalen Potentiometers statt. Softwareseitig geschieht diese Einstellung über den bereits in den Kernelquellen enthaltenen Treiber für digitale Potentiometer `dpot` (siehe [11]). Um diesen jedoch verwenden zu können, muss zuerst das Boardfile in den Kernelquellen angepasst werden. Dies erfolgt durch Einfügen folgender Zeilen aus Codeabschnitt 3.4 in die Datei `linux-2.6.x/arch/blackfin/mach-bf527/boards/cm_bf527.c`

---

**Codeabschnitt 3.4:** Änderungen im Boardfile des `cm-bf527` - Teil I

---

```
1 #if defined(CONFIG_AD525X_DPOT) ||
2     defined(CONFIG_AD525X_DPOT_MODULE)
3     static struct bfin5xx_spi_chip spi_ad_dpot_chip_info = {
4         .enable_dma = 0,
5     };
6 #endif
```

---

Des Weiteren muss in der gleichen Datei in das Struct `bfin_spi_board_info[]` die Konfiguration in Codeabschnitt 3.5 eingefügt werden.

---

**Codeabschnitt 3.5:** Änderung im Struct `bfin_spi_board_info[]` im Boardfile des `cm-bf527`

---

```
1 #if defined(CONFIG_AD525X_DPOT) || defined(CONFIG_AD525X_DPOT_MODULE)
2 {
3     .modalias = "ad_dpot",
4     .platform_data = "ad5293",
5     .max_speed_hz = 1000000,
6     .bus_num = 0,
```

---

```

7   .chip_select = 1,
8   .controller_data = &spi_ad_dpot_chip_info ,
9   .mode = SPI_MODE_1,
10  },
11  {
12   .modalias = "ad_dpot" ,
13   .platform_data = "ad5293" ,
14   .max_speed_hz = 1000000,
15   .bus_num = 0,
16   .chip_select = 6,
17   .controller_data = &spi_ad_dpot_chip_info ,
18   .mode = SPI_MODE_1,
19  },
20 #endif

```

---

Hiermit wird dem Treiber die SPI Taktrate, die verwendete Chipselectleitung sowie der SPI-Modus mitgeteilt. Damit der dpot Treiber auch vom Kernel verwendet wird, muss nun in der Kernelkonfiguration dieser Treiber aktiviert und der Kernel neu kompiliert werden. Erst jetzt kann das Userprogramm über den dpot Treiber auf die digitalen Potentiometer zugreifen. Zu diesem Zweck gibt es die beiden Routinen **getWiperPosition** und **setWiperPosition**. Beide Routinen verwenden das Struct **channel** (siehe Codeabschnitt 3.6 um den aktuellen Zustand des Eingangskanals auszulesen oder zu speichern.

**Codeabschnitt 3.6:** Struktur in der der aktuelle Zustand eines Eingangskanals gespeichert wird

---

```

1  struct channel {
2   int active;
3   int sport_fd;
4   int wiper_pos;
5   int gain;
6   int resistor;
7   const char *spi_dev_path;
8  };

```

---

Für jeden der beiden Messsignalpfade gibt es ein solches Struct. Hier wird festgehalten, ob der aktuelle Kanal aktiv ist (wird momentan nicht ausgewertet), welchen Filedescriptor das SPORT Characterdevice hat, in welcher Position sich der Schleifer des

entsprechenden digitalen Potentiometers gerade befindet, welche Werte sich für die Eingangsverstärkung und den Widerstandswert aus der Schleiferposition ergeben, sowie den Pfad für das SPI Device, über welches das digitale Potentiometer angesprochen werden kann. Diese beiden Pfade ergeben sich über die Chipselectleitung, welche im Boardfile angegeben wurde und lauten `/sys/bus/spi/devices/spi0.1/rdac0` für IC12 sowie `/sys/bus/spi/devices/spi0.6/rdac0` für IC16.

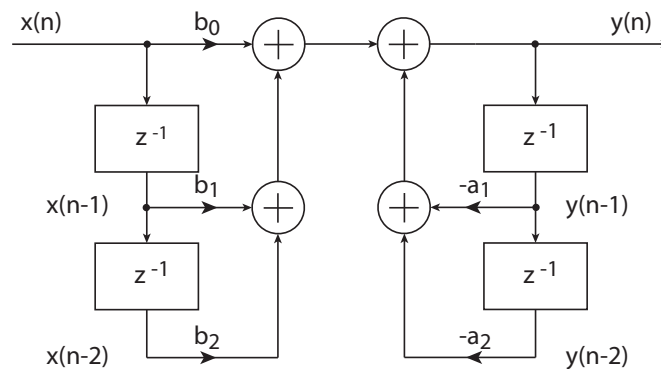
Mit der Routine **getWiperPosition** wird die Schleiferposition des übergebenen Potentiometers ausgelesen. Hierfür wird zunächst das entsprechende Device geöffnet und versucht die Schleiferposition zu ermitteln. Hat das Auslesen funktioniert, so werden die Werte im zugehörigen channel-Struct aktualisiert. Abschließend wird das geöffnete SPI Device wieder geschlossen.

Mit **setWiperPosition** hingegen wird der Schleifer auf eine neue Position gesetzt und somit die Verstärkung des Instrumentierungsverstärkers geändert. Hierfür wird wieder zuerst das entsprechende SPI Device geöffnet und anschließend versucht, den neuen Wert an das digitale Potentiometer zu senden. Um zu überprüfen, ob das Setzen der neuen Schleiferposition funktioniert hat, wird mit **getWiperPosition** der aktuelle Wert ausgelesen und eine Fehlermeldung ausgegeben, falls der Sollwert nicht mit dem ausgelesenen Istwert übereinstimmt.

**Berechnung des Lock-in Algorithmus:** Die eigentliche Berechnung des Lock-in Algorithmus erfolgt in der Routine **processData**. Da in der ersten Ausbaustufe das Mainboard nur mit zwei der vier Modulationsplatten und einem APD-Modul bestückt wurde, ist auch der Algorithmus nur für zwei der vier Trägerfrequenzen und einem Eingangskanal ausgeführt. Zur Veranschaulichung der Arbeitsweise des dual-Phase Lock-in Algorithmus sei an dieser Stelle noch einmal auf Abbildung 2.6 verwiesen. Es ist zu erkennen, dass zur Berechnung der Ausgangswerte Tiefpassfilter notwendig sind. Da der Lock-in Algorithmus per Software berechnet wird, sind auch diese Filter digital ausgeführt. Im Zuge des Entwicklungsprozesses wurde festgelegt, dass für sämtliche digitale Filter des Lock-in Algorithmus kaskadierte IIR-Filter zum Einsatz kommen sollen. Dies bedeutet, dass Filter höherer Ordnung durch Serienschaltung von Filterelementen zweiter Ordnung realisiert werden. Die Verwendung eines IIR Filters hat den Vorteil, dass hier eine wesentlich geringere Filterordnung nötig ist, als es der Fall wäre, würde hier ein FIR Filter verwendet werden. Der Nachteil eines IIR Filters besteht jedoch darin, dass er, im Gegensatz zu einem FIR Filter, instabil sein kann. Ein weiterer Nachteil ist,

dass sich bei einem IIR Filter Quantisierungsfehler der Koeffizienten und Rundungsfehler, Überlauf oder Sättigung im Filteralgorithmus wesentlich stärker auswirken, als bei einem FIR Filter. So kann es passieren, dass z.B. ein stabiler IIR Filter mit double precision Floatingpointkoeffizienten nach dem Quantisieren der Koeffizienten plötzlich mit Fixedpointkoeffizienten nicht mehr stabil ist. Dies kann z.B. dann passieren, wenn Polstellen in der Nähe des Einheitskreises liegen und durch die Koeffizientenquantifizierung außerhalb des Einheitskreises zu liegen kommen.

Als Implementierungsstruktur wurde die Direktform I gewählt, wie sie in Abbildung 3.36 dargestellt ist. Diese Struktur lässt sich in Gleichung 3.25 umschreiben und in Sourcecode implementieren.



**Abbildung 3.36:** IIR Filter zweiter Ordnung in Direktform I (entnommen und modifiziert aus [19])

$$y(n) = b_0x(n) + b_1x(n-1) + b_2x(n-2) - a_1y(n-1) - a_2y(n-2) \quad (3.25)$$

In der Funktion **iirFloat** ist diese Gleichung schlussendlich umgesetzt. Als Übergabeparameter nimmt diese Funktion einen Pointer auf den Buffer, welcher die Ausgangswerte beinhaltet, einen Pointer auf den Buffer, welcher die gefilterten Werte enthalten soll, die Größe der beiden Buffer sowie die Anzahl der Filterstufen. Diese Filterroutine wird für sämtliche digitalen Filter innerhalb des Lock-in Algorithmus verwendet. Zum Betrieb solch eines digitalen Filters sind einige zusätzliche Konfigurationen zu treffen. Hierfür steht das Struct **iir\_state\_float** (siehe Codeabschnitt 3.7) zur Verfügung.



**Codeabschnitt 3.7:** Struktur um die Konfiguration des IIR Filters zu speichern

---

```

1  typedef struct
2  {
3      float *c;           // coefficients array
4      float *d;           // delay line
5      short num_of_stages; // number of stages
6  } iir_state_float;

```

---

`c` hält den Zeiger auf das Koeffizientenarray. Die Koeffizienten in diesem Array müssen in der Reihenfolge  $b_{01}, b_{11}, b_{21}, a_{11}, a_{12}, b_{02}, b_{12}, \dots$  gespeichert werden, wobei die zweite Ziffer des Index die Stage bezeichnet. Die  $a_{0k}$  Koeffizienten sind immer 1. Die Berechnung der Filterkoeffizienten für sämtliche verwendeten digitalen Filter erfolgt in Matlab mit Hilfe des Skriptes **filterdim.m**. Dieses Skript generiert mit Hilfe der Funktion **coeff\_2\_c\_header** zwei C-Headerfiles, welche die Anzahl der Filterstufen sowie die Filterkoeffizienten in der richtigen Reihenfolge beinhalten. **aa\_lockin\_def.h** enthält die Koeffizienten für die Antialiasingfilter, **lp\_lockin\_def\_8000.h** die Koeffizienten für die Ausgangstiefpassfilter. Diese Headerfiles werden im Headerfile **lockin.h** ins Hauptprogramm eingebunden.

Da, wie schon etwas weiter vorne in dieser Arbeit erwähnt, die Datenverarbeitung blockweise erfolgt, müssen immer die letzten zwei Ein- und Ausgangswerte ( $x$ - und  $y$ -Werte) jeder Filterstufe gespeichert werden, um für die Verarbeitung des nächsten Datenblocks die jeweiligen  $x(n-1)$ ,  $x(n-2)$  sowie  $y(n-1)$  und  $y(n-2)$  Werte zur Verfügung zu haben und somit ein Einschwingen des Filters zu Beginn eines jeden Datenblocks zu verhindern. Diese Ausgangswerte werden in der Delayline **d** gespeichert. Für jeden verwendeten Filter muss vor dessen Verwendung mit Hilfe des folgenden Makros (Codeabschnitt 3.8) ein solches Struct initialisiert werden.

**Codeabschnitt 3.8:** Initialisierung einer Filterkonfiguration

---

```

1  #define iir_init_float(state, coeffs, delay, stages) \
2      (state).c = (coeffs); \
3      (state).d = (delay); \
4      (state).num_of_stages = (stages)

```

---

Nun zum eigentlichen Ablauf der Datenblockverarbeitung in **dataProcessing**. Zunächst wird in einer Schleife der aktuelle Datenblock von Kanal 1, welcher sich im

Userspace Eingangsbuffer `indata_dbl_buf_1_` befindet, durchlaufen und dabei jedes Sample dieses Buffers einmal mit einem Sample der internen 3 kHz Sinustabelle und einmal mit einem Sample der internen 3 kHz Cosinustabelle multipliziert. Das Selbe geschieht mit den beiden 7 kHz Tabellen. Über den global definierten zirkularen Index `lockin_sin_idx_`, welcher in der Funktion `circindex` realisiert ist, werden die entsprechenden Samples aus den internen Lookuptabellen bestimmt. Somit ergeben sich insgesamt vier neue Buffer. Nun erfolgt die erste digitale Filterung, bei der diese vier Buffer je einen Tiefpassfilter durchlaufen. Sie ist notwendig, um Antialiasing beim Downsampeln zu vermeiden. Es wird ein Chebychev Tiefpassfilter zweiter Ordnung eingesetzt, dessen Grenzfrequenz mit 500 Hz festgelegt wurde. Die Samplefrequenz für welche dieser Filter dimensioniert wurde, ist die gleiche, wie die, mit der die Eingangswerte gesampelt wurden, also 48 kHz. Diese Grenzfrequenz wurde gewählt, um einerseits von Beginn an Störsignale zu dämpfen, andererseits um die Samplerate des Ausgangstiefpassfilters möglichst niedrig wählen zu können.

Der nächste Schritt im Algorithmus ist das Downsampeln. Hier werden nun die Samples der vier Ausgangssignale von den Antialiasingfiltern in einer Schleife durchlaufen und jeder n-te Wert jedes dieser vier Arrays in je einen weiteren Buffer gespeichert, wobei n der Downsamplingfaktor ist, welcher durch die Konstante `DS_FACTOR` im Headerfile `lockin.h` festgelegt ist. Bei einer Eingangssamplerate von 48 kHz und einer Ausgabesamplerate von 8 kHz ergibt sich somit ein `DS_FACTOR` von

$$\frac{48 \text{ kHz}}{8 \text{ kHz}} = 6 \quad (3.26)$$

Hier hilft wiederum ein zirkularer Index, um die entsprechenden Samples aus den Ausgangsbuffern von den Antialiasingfiltern auszuwählen.

Nun kann die Ausgangstiefpassfilterung erfolgen. Hierbei wird ein Butterworth Tiefpassfilter zweiter Ordnung verwendet, dessen Grenzfrequenz bei 2 Hz liegt und der für eine Samplingfrequenz von 8 kHz berechnet wurde.

Den nächsten Schritt in der Datenverarbeitungsroutine stellt die Absolutwertberechnung dar. Wie in Gleichung 2.13 erläutert, wird nun wiederum in einer Schleife aus den Samples der zwei entsprechenden Buffer jeder Trägerfrequenz ein Amplitudenwert berechnet. Eine Multiplikation mit dem Faktor zwei nimmt die Anpassung an den Signalebereich der Ausgangs DACs vor. Anschließend werden die berechneten Amplitudenwerte einem der vier DACs zugeordnet, indem sie mit der entsprechenden Adresse

ODER-verknüpft wird. Wie das Datenblatt des DAC zeigt [4] sind hier die Adressen 0x03000000, 0x03100000, 0x03200000 und 0x03300000 möglich. Sie setzen sich aus dem Kommando „Write to and update DAC Channel n“ (0x03000000) und der entsprechenden Adresse des DACs 0x00000000, 0x00100000, 0x00200000 oder 0x00300000 zusammen.

Zum Abschluss wird noch das Flag `data_arrived_` rückgesetzt, um dem Hauptprogramm zu signalisieren, dass die Datenverarbeitung abgeschlossen ist und ein neuer Block verarbeitet werden kann.

### 3.3 Entwickeln von Software für das Prozessormodul

Für die Softwareentwicklung für den ADSP-BF527 wurde in diesem Projekt ein Linux Hostsystem gewählt. Zum Einsatz kam hier der Einfachheit halber Linux Mint 12 (Lisa). Zusätzlich wurde noch die Blackfin Toolchain in der Version ADI-2011R1-RC3 installiert, welche kostenlos über die Webseite [5] heruntergeladen werden kann. Über die Toolchain werden das Hauptprogramm sowie die Treibermodule kompiliert. Auch die Software zum Flashen des U-Boot Images bzw. des uCLinux Images sowie der Debugging Sever GDB befinden sich im Toolchainpaket. Für diese drei Aufgaben wird zusätzlich der JTAG-Adapter gnICE benötigt. Hier ist anzumerken, dass mit der Toolchainversion ADI-2011R1-RC3 das Flashen des U-Boot Images nicht funktioniert hat. Aus diesem Grund wurden für diese Aufgabe die JTAG Tools einer 2009er Toolchainversion verwendet.

Als Entwicklungsumgebung diente die 32-Bit Variante der „Eclipse IDE for C/C++ Developers“ in der **Helios** Version. Sie kann kostenfrei unter der URL [17] heruntergeladen werden. Es ist auch möglich, das Eclipse Paket aus den Linux Mint Paketquellen zu installieren, jedoch ergaben sich während der Konfiguration einige Schwierigkeiten, weswegen auf die selbst heruntergeladene Version zurückgegriffen wurde. Das heruntergeladene Archiv lässt sich einfach in ein Verzeichnis entpacken und von dort aus konfigurieren und starten. Es ist noch darauf zu achten, dass innerhalb von Eclipse die beiden Plugins „Memory Mapped Registers View Feature“ sowie „Blackfin GNU Toolchain Feature“ installiert werden müssen. Aus diesem Grund ist hier die **Helios** Version von Eclipse zu bevorzugen, da zum Zeitpunkt der Entstehung dieser Arbeit diese Plugins nicht mit neueren Eclipse Versionen funktionierten.

Erste Anlaufstellen für Fragen und Hilfestellungen bei Installation, Konfiguration und Programmierung von U-Boot, uCLinux, Treibern und Anwendungen sind die Analog Devices Wiki Seiten unter [3] sowie die Foren der „Analog Devices Engineer Zone“ unter [2] und [10].

Um den Entwicklungsprozess zu vereinfachen, wurde mit **tftp-hpa** ein TFTP-Server installiert. Hiermit kann dem Bootloader U-Boot automatisch das aktuelle Betriebssystemimage per Ethernetverbindung zur Verfügung gestellt werden. Somit erspart sich der Entwickler jedes Mal das Kernelimage zu flashen, nachdem eine Konfigurationseinstellung der Kernelkonfiguration geändert und der Kernel neu kompiliert wurde. Stattdessen muss dieses Image lediglich in das konfigurierte Uploadverzeichnis des TFTP Servers kopiert werden. Ist U-Boot entsprechend konfiguriert, lädt sich U-Boot automatisch das Betriebssystemimage und bootet dieses. Um unvorhersehbare Probleme bei der Netzwerkübertragung zu vermeiden, wird empfohlen, der Ethernetschnittstelle des Hostsystems eine statische IP Adresse (z.B. 10.0.0.1) zuzuweisen.

Als Bootloader kommt in dieser Arbeit „Das U-Boot“ in der Version ADI-2012R1 zum Einsatz.

Für die uCLinux Distribution wurde die Version 3.0.8-ADI-2011R1-RC3 gewählt. Die für den Betrieb des Lock-in Verstärker benötigten Dateien sind einerseits die beiden Treibermodule **AD7688mod** sowie **AD7304mod** und andererseits die Hauptprogrammdatei **lockin**. Die beiden Treibermodule befinden sich in „blackfin-uclinux-dist/linux-2.6.x/drivers/char/lockin\_amplifier“ und werden beim Kompilieren des Kernels automatisch mitkompiliert, in den Kernel eingefügt und beim Booten geladen. Die Hauptprogrammdatei **lockin** muss händisch kompiliert und vor dem Kompilieren des Kernels in das Verzeichnis „blackfin-uclinux-dist/romfs/bin“ kopiert werden. Nach dem Kompilieren des Kernels befindet sich nun die Hauptprogrammdatei in „/usr/bin“ und kann nach dem Booten des Kernels von dort aus gestartet werden.

Um effizient Software für das Coremodul zu entwickeln wird empfohlen, die Hauptplatine über einen USB - UART Adapter mit Pegelwandler (optimaler Weise basierend auf einem cp2102 Wandler von Silabs), mit dem Hostsystem zu verbinden. Mit dem Terminalprogramm PUTTY lässt sich nun bequem auf der U-Boot oder uCLinux Konsole des CM-BF527 arbeiten. Wird an den Treibermoduldateien entwickelt, so empfiehlt es sich, im Makefile „blackfin-uclinux-dist/linux-2.6.x/drivers/char/Makefile“ die Zeile in Codeabschnitt 3.9 auszukommentieren, den Kernel neu zu kompilieren und zu booten.

---

**Codeabschnitt 3.9:** Zeile im Makefile mit dem das Compilieren der Lockin Files ein- oder ausgeschaltet werden kann

---

```
1 obj-y += lockin_amplifier/
```

---

Nun werden die Treibermodule für SPORT0 und SPORT1 nicht mehr automatisch geladen. Das kompilierte Treibermodul (\*.ko Datei) kann nun mittels RCP auf das Coremodul geladen werden und von dort über die Konsole mit Hilfe des *insmod* Befehls geladen bzw. mit Hilfe des Befehls *rmmod* entladen werden. Wird hingegen am Hauptprogramm entwickelt, so ist nur das Kopieren der kompilierten Programmdatei per RCP notwendig. Über den Befehl *./lockin* kann das Programm in der Konsole gestartet werden.

Auf dem dieser Arbeit beiliegenden Datenträger befinden sich neben dem Sourcecode für die Treibermodule und dem Hauptprogramm auch die verwendeten Versionen der Toolchain, der uCLinux Distribution inkl. konfigurierbarem Kernel, von Eclipse und Putty. Auch die Matlabskripte, welche zum Berechnen der Filterkoeffizienten sowie der Sinuslookuptabellen verwendet wurden, liegen dieser Arbeit bei.

# Kapitel 4

## Bedienung

### Inhalt

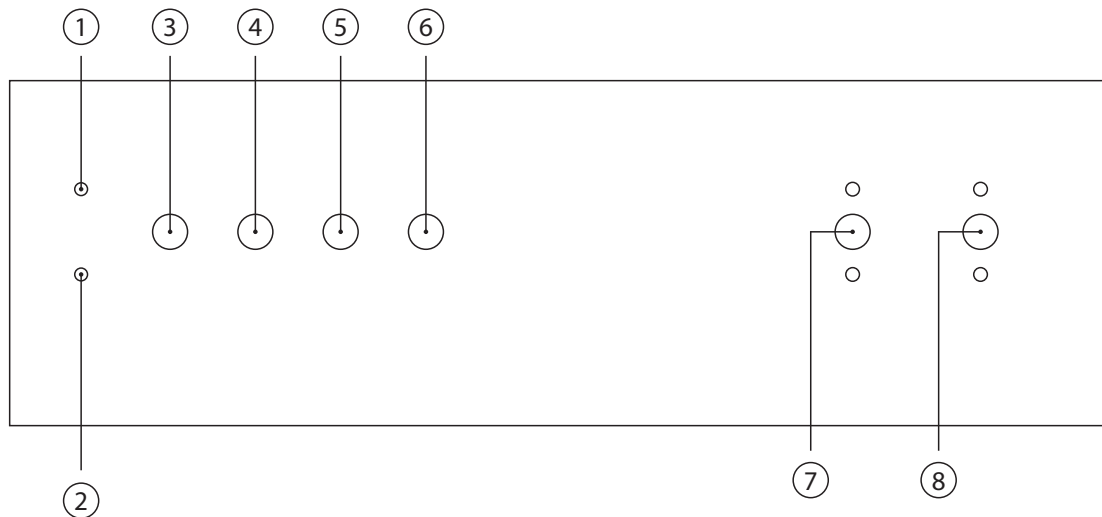
---

<b>4.1</b>	<b>Vorderseite des Verstärkers</b>	<b>81</b>
<b>4.2</b>	<b>Rückseite des Verstärkers</b>	<b>82</b>
<b>4.3</b>	<b>Bedienung der Software</b>	<b>83</b>

---

Der entwickelte Lock-in Verstärker besitzt Bedien- und Anschlusselemente auf der Vorder- sowie auf der Rückseite des Gerätes. Die Vorderseite ist die dem Probanden zugewandte Seite. Auf Ihr befinden sich diejenigen Elemente, welche für die Messung am Probanden relevant sind. Beim Gehäusedesign wurde speziell darauf geachtet, dass auf der Probandenseite keine mit Masse verbundenen, berührbaren Metallteile vorhanden sind. Sämtliche metallische Anschlussbuchsen wurden von der Gerätemasse abgekoppelt. Die Anschlussbuchsen auf der Geräterückseite, also der dem Probanden abgewandten Seite, besitzen jedoch Masseverbindung. Auf dieser Geräteseite sind die Elemente angebracht, an denen die analogen bzw. die digitalen Ausgangssignale abgegriffen werden können. Auch die Verbindung zur Stromversorgung, sowie der Ein/Aus Schalter befinden sich auf dieser Geräteseite. Im Folgenden werden nun sowohl die genaue Lage und Bedeutung der Bedienelemente, als auch die softwareseitige Bedienung des Lock-in Verstärkers beschrieben.

## 4.1 Vorderseite des Verstärkers



**Abbildung 4.1:** Die Vorderseite des Verstärkers

- ① Status LED 1: Anzeige des allgemeinen Betriebszustands (Programm gestartet / Messung aktiv)
- ② Status LED 2: Anzeige Bootvorgang
- ③ Ausgang Modulationssignal 1: 2 kHz - FCPC Buchse
- ④ Ausgang Modulationssignal 2: 3 kHz - FCPC Buchse
- ⑤ Ausgang Modulationssignal 3: 5 kHz - FCPC Buchse
- ⑥ Ausgang Modulationssignal 4: 7 kHz - FCPC Buchse
- ⑦ Eingang Kanal 1 - FCPC Buchse
- ⑧ Eingang Kanal 2 - FCPC Buchse

Um die Modulationssignale zum Einstrahlort zu führen bzw. vom Messort abführen zu können, werden zwei Glasfaserkabel mit FCPC Steckern, wie sie Bauernfeind in [12] beschreibt, benötigt. Ein Y-Kabel sowie ein 1:1 durchverbundenes. Das Y-Kabel wird an zwei von den vier FCPC Buchsen ③ - ⑥ angeschlossen, um zwei gewünschte Modulationsfrequenzen in den Einstrahlort einzukoppeln. Das Durchverbundene Kabel wird mit der Buchse des gewünschten Eingangskanals ⑦ oder ⑧ verbunden.

## 4.2 Rückseite des Verstärkers

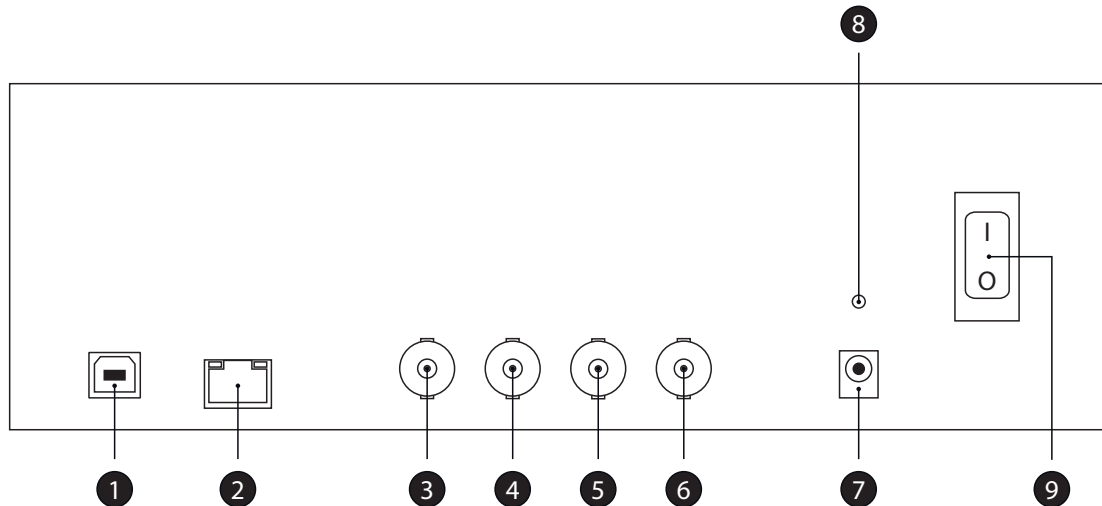


Abbildung 4.2: Die Rückseite des Verstärkers

- ❶ USB Anschluss: Zur digitalen Ausgabe der Messwerte
- ❷ Ethernetanschluss: Zur Steuerung des Lock-in Verstärkers bzw. zur digitalen Ausgabe der Messwerte
- ❸ Analogausgang 1: BNC Buchse
- ❹ Analogausgang 2: BNC Buchse
- ❺ Analogausgang 3: BNC Buchse
- ❻ Analogausgang 4: BNC Buchse
- ❼ Eingang Spannungsversorgung DC 12V
- ❽ Kontrollleuchte Spannungsversorgung
- ❾ Ein/Aus Schalter

Zum Betrieb des Lock-in Verstärkers wird ein externes Netzteil mit einer Ausgangsleistung von mindestens 30 W bei einer Ausgangsspannung von 12 V DC benötigt. Dies entspricht einem Ausgangsstrom von 2.5 A. Idealerweise weist das verwendete Netzteil eine Medizingeräte-zertifizierung auf. Der Klinkenstecker muss die Maße von 5.5 mm x



2.5 mm x 12 mm sowie eine Polarität, wie sie in Abbildung 4.3 dargestellt ist, aufweisen und wird mit Buchse ⑦ verbunden. Wird das Gerät mit dem Ein/Ausschalter ⑨ eingeschaltet, so beginnt die Spannungskontrollleuchte ⑧ grün zu leuchten. Buchse ① dient



**Abbildung 4.3:** Die Polarität des Anschlusses für die Spannungsversorgung

dazu, den Lock-in Verstärker über USB mit einem PC zu verbinden, um eine digitale Übertragung der Messwerte zu ermöglichen. Auch über ② können je nach Konfiguration digitale Messwerte übertragen werden. Sie dient in erster Linie jedoch zur Steuerung des Verstärkers.

Über die Anschlüsse ③ bis ⑥ erfolgt die analoge Ausgabe der Messwerte, wobei hier ein maximaler Spannungspegel von 8.192 V möglich ist.

### 4.3 Bedienung der Software

Um den Lock-in Verstärker bedienen zu können, muss er zunächst per Netzkabel mit einem PC verbunden werden. Auf dem PC wird ein SSH Client, wie z.B. Putty benötigt. Nach dem Einschalten des Gerätes über den Netzschalter ⑨ (siehe Abschnitt 4.2) wird der Lock-in Verstärker automatisch gebootet. Der laufende Bootvorgang wird durch das durchgehende Leuchten der orangen LED ② angezeigt. Ist dieser Vorgang abgeschlossen, erlischt die orange LED ② und das Gerät ist per SSH auf der IP Adresse **10.0.0.3** erreichbar. Der Benutzer kann sich nun auf der Konsole mit dem Standard uClinux Benutzernamen **root** sowie dem casesensitiven Passwort *uClinux* an der Linux Konsole des Lock-in Verstärkers anmelden. Im Verzeichnis „/usr/bin“ liegt nun die Programmdatei **lockin** welche mit dem Befehl

```
/usr/bin/lockin
```

ausgeführt werden kann. Umgehend erscheint nun die Eingabeaufforderung „LockIn Shell>“, des Programmes und die grüne LED ① beginnt durchgehend zu leuchten um den betriebsbereiten Zustand des Gerätes zu signalisieren. An dieser Stelle stehen dem Benutzer nun folgende Möglichkeiten offen, den Programmablauf zu steuern

- *g*: Einstellen der Eingangsverstärkung eines der beiden Eingangskanäle

- *e*: Starten des Messvorgangs
- *q*: Beenden des Programmes
- *h, ?*: Anzeige eines Hilfetextes mit den möglichen Steuerbefehlen

***g* - Einstellen der Eingangsverstärkung:** Hier erscheint zunächst die Aufforderung, den Eingangskanal zu wählen. Korrekte Eingabemöglichkeiten sind hier *1* oder *2*. Bei einer anderen Eingabe wird die Aufforderung zur Kanalwahl nocheinmal angezeigt. Wurde die Eingabe als Korrekt erkannt, so wird nun der Benutzer mit der Eingabeaufforderung „Enter gain value:“ aufgefordert, den gewünschten Verstärkungswert einzugeben. Der Wertebereich, welcher eingestellt werden kann, reicht hier von *2* bis *1050*. Nachdem die Verstärkung eingegeben und die Entertaste betätigt wurde, wird im Programm der nächst mögliche Verstärkungswert berechnet, die neue Eingangsverstärkung eingestellt und das Programm zeigt die neue Verstärkung an. Abschließend erscheint wiederum die Eingabeaufforderung der Lock-in Shell.

***e* - Starten des Messvorgangs:** Mit dem Kommando *e* werden zwei essenzielle Aktionen gestartet. Als erstes beginnt die Ausgabe aller vier Modulationssignale, gefolgt vom Start des Einlesevorgang von den ADCs mit gleichzeitiger Berechnung des Lock-in Algorithmus sowie der Ausgabe der berechneten Amplitudenwerte. Wurde alles korrekt gestartet, erscheint am Bildschirm die Bestätigung „Measurement in progress... Press ctrl+z to stop“. Ist diese Meldung zu sehen, so läuft der Messvorgang. Dieser Betriebszustand wird gleichzeitig mit dem Blinken der grünen LED ① angezeigt. In diesem Betriebszustand wird vom Programm keine andere Eingabe als *ctrl+z* angenommen, d.h. es ist auch keine Verstärkungsanpassung im laufenden Betrieb möglich. Drückt der Benutzer nun diese Tastenkombination, so wird der Messvorgang gestoppt, die grüne LED ① geht wieder in den stetig leuchtenden Zustand über und die Lock-in Shell wird wieder angezeigt.

***q* - Beenden des Programms:** Nach Eingabe dieses Kommandos und anschließender Bestätigung über die *Enter*-Taste wird das Programm beendet und die grüne LED ① erlischt. Anschließend erscheint wieder die Standard Linuxkonsole im SSH-Client Putty.

# Kapitel 5

## Ergebnisse

### Inhalt

---

<b>5.1</b>	<b>Platinenaufbau</b>	<b>85</b>
<b>5.2</b>	<b>Gehäuseaufbau</b>	<b>88</b>
<b>5.3</b>	<b>Testmessungen</b>	<b>88</b>

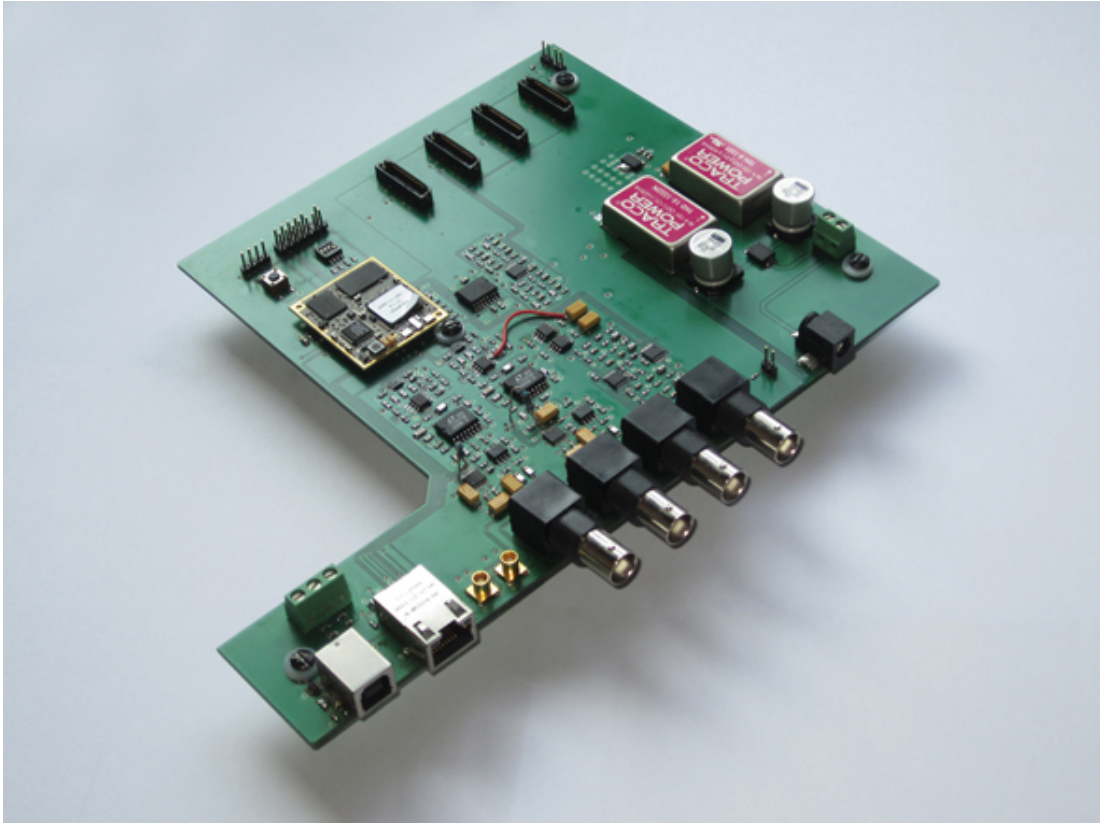
---

Dieses Kapitel gibt einen Überblick über die Ergebnisse, die in dieser Diplomarbeit erreicht wurden. Die ersten beiden Abschnitte befassen sich mit dem Hardwareaufbau, also des Mainboards und der Modulationsplatinen, sowie dem Gehäuse. Im zweiten Abschnitt werden Testmessungen beschrieben, welche die korrekte Funktionsweise des entwickelten Lock-in Verstärkers demonstrieren.

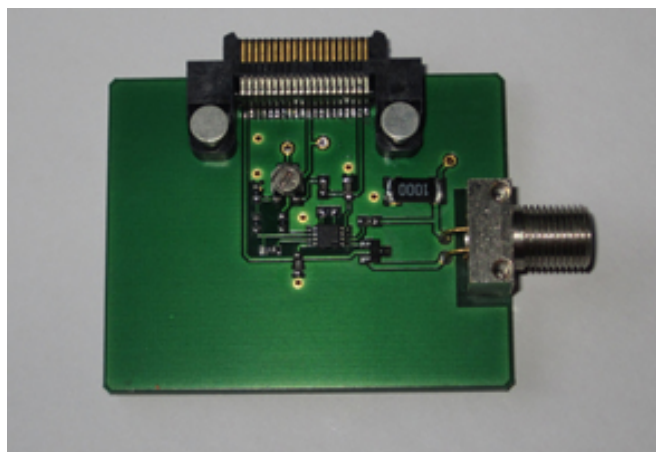
### 5.1 Platinenaufbau

Abbildung 5.1 zeigt den Prototypenaufbau des Mainboards. Gut zu erkennen ist das DSP Modul, die beiden DC/DC Wandler sowie die vier Steckplätze, welche die Modulationsboards aufnehmen können. Abbildung 5.2 zeigt eine der Modulationsplatinen. Am oberen Rand der Platine ist deutlich der verwendete robuste Stecker zu sehen, welcher den Modulationsplatinen optimalen Halt auf der Hauptplatine gibt. Des Weiteren ist hier die FCPC Buchse zu erkennen, an die das Glasfaserkabel angeschlossen wird, welches das von der LED ausgestrahlte Licht an den Messort transportiert. In Abbildung

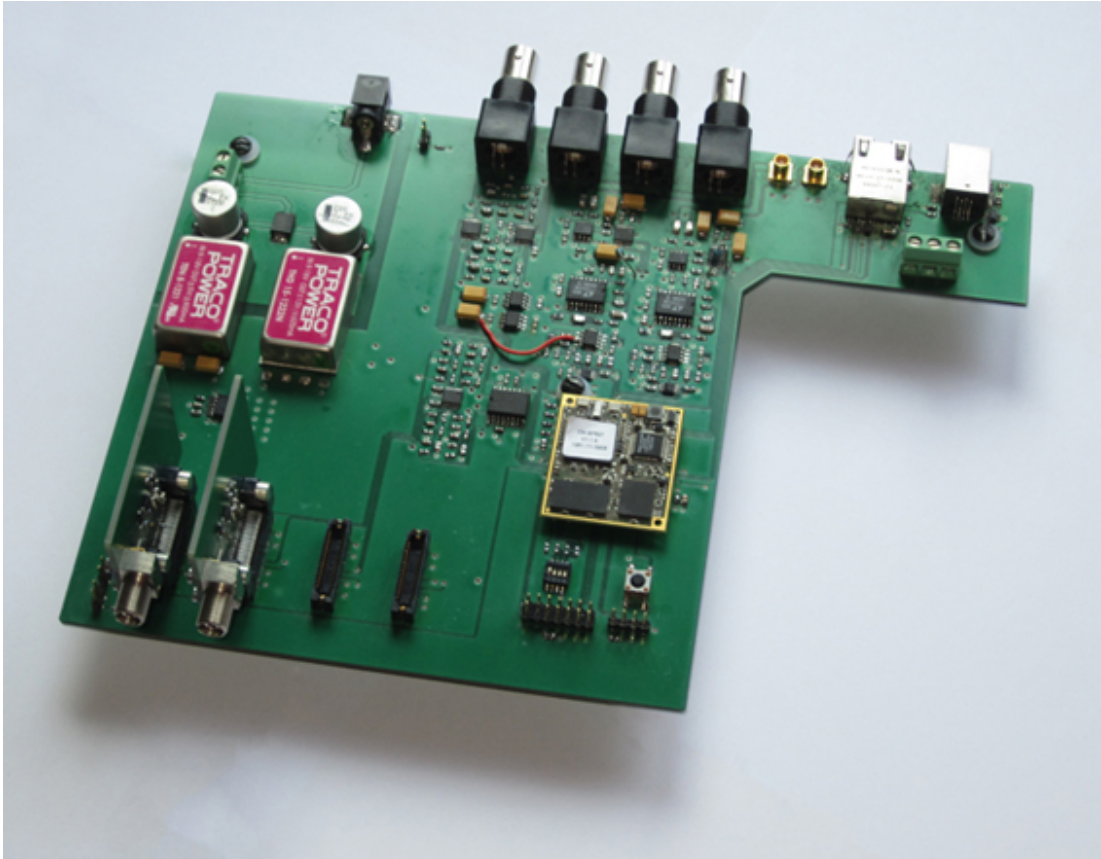
5.3 ist nocheinmal die Hauptplatine zu sehen, jedoch diesmal mit zwei aufgesteckten Modulationsplatinen. Sämtliche Hardwareaufbauten wurden händisch in Eigenarbeit ausgeführt.



**Abbildung 5.1:** Mainboard des entwickelten Lock-in Verstärkers



**Abbildung 5.2:** Fertig bestückte Modulationsplatine



**Abbildung 5.3:** Mainboard des entwickelten Lock-in Verstärkers mit zwei aufgesteckten Modulationsplatinen

## 5.2 Gehäuseaufbau



**Abbildung 5.4:** Die dem Probanden zugewandte Seite des Lock-in Verstärkers - die Gehäusefront



**Abbildung 5.5:** Die dem Probanden abgewandte Seite des Lock-in Verstärkers - die Gehäuserückseite

## 5.3 Testmessungen

Für sämtliche Testmessungen wurde eine mit zwei Modulationsboards versehene Mainboardkonfiguration verwendet. Ein Modulationsboard, das mit einer 890 nm LED bestückt worden ist, wurde im Steckplatz J2 (siehe ④ in Abbildung 4.1) platziert, ein zweites, dieses war mit einer 670 nm LED versehen, steckte in J4 (siehe ⑥ in

Abbildung 4.1). So ergab sich für das erste Modul eine Trägerfrequenz von 3 kHz und für das zweite Modul eine Trägerfrequenz von 7 kHz.

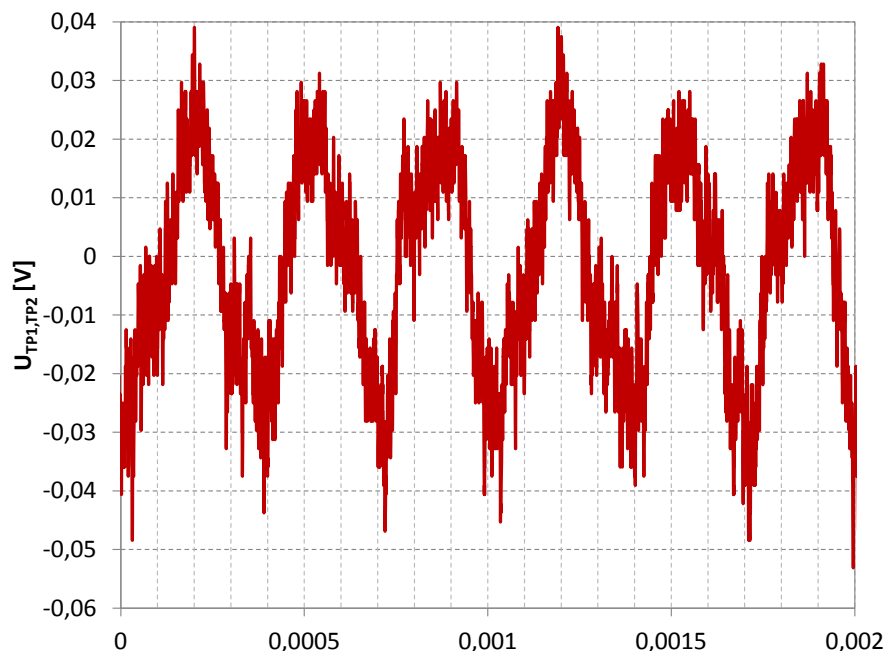
Abbildung 5.6 zeigt die bei allen Testmessungen verwendete Anordnung der Optoden. Für die Einkopplung der Lichtsignale in den Kopf wurde ein Y-Glasfaserkabel verwendet, dessen einzelnes Ende, dasjenige mit dem Winkelkopf, in die Kopfhalterung (siehe Abbildung 5.6) gesteckt wurde. Die zwei Enden mit den FCPC Steckern wurden mit den FCPC Buchsen der Modulationsboards verbunden. Während der Messung wurde der Messraum abgedunkelt, um Störungen zu minimieren, welche durch Streulicht hervorgerufen werden.



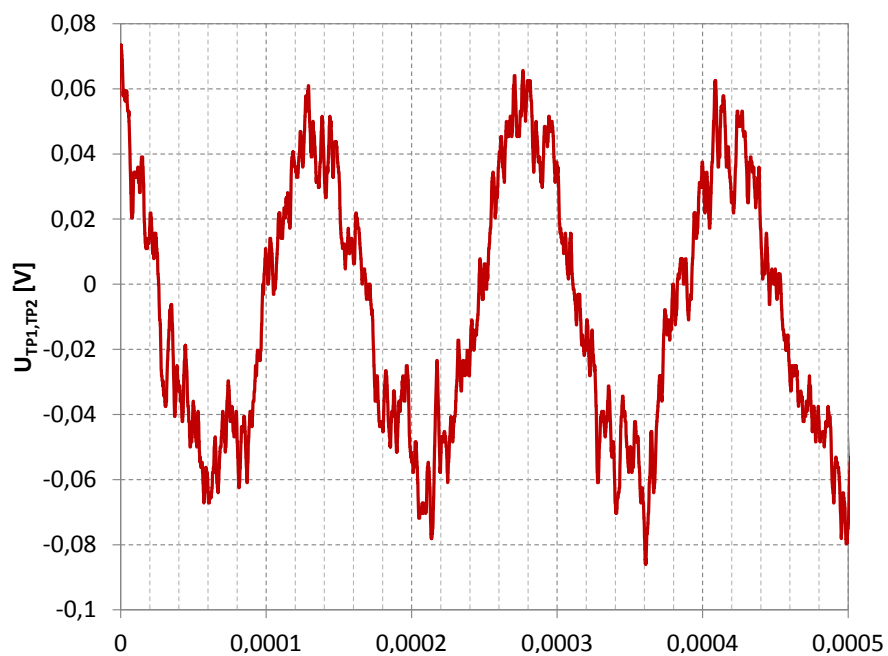
**Abbildung 5.6:** Position der Optoden für die Messungen

Zunächst wurden zur Kontrolle nur die während der Messungen verwendeten Modulationsfrequenzen aufgenommen. Abbildung 5.7 zeigt das Oszilloskopbild des Spannungssignals, welches vom APD Modul ausgegeben wird, wenn nur das mit 3 kHz modulierte Lichtsignal in den Kopf eingekoppelt und in einem Abstand von ca. 3 cm wieder aufgenommen wird.

In Abbildung 5.8 ist das Oszilloskopbild des Ausgangssignals des APD Moduls dargestellt, wenn nur das 7 kHz Signal verwendet wird. Nun wurden beide Enden des Y-Glasfaserkabels an die beiden Modulationsplatten angeschlossen. Es erfolgt somit innerhalb des Kabels eine optische Überlagerung des 3 kHz und des 7 kHz Signals, dessen Ergebnis am Ausgang des APD Moduls, wie in Abbildung 5.9 dargestellt ist, aussieht. Somit ist sichergestellt, dass ein korrektes Signal in den Kopf eingekoppelt wird.

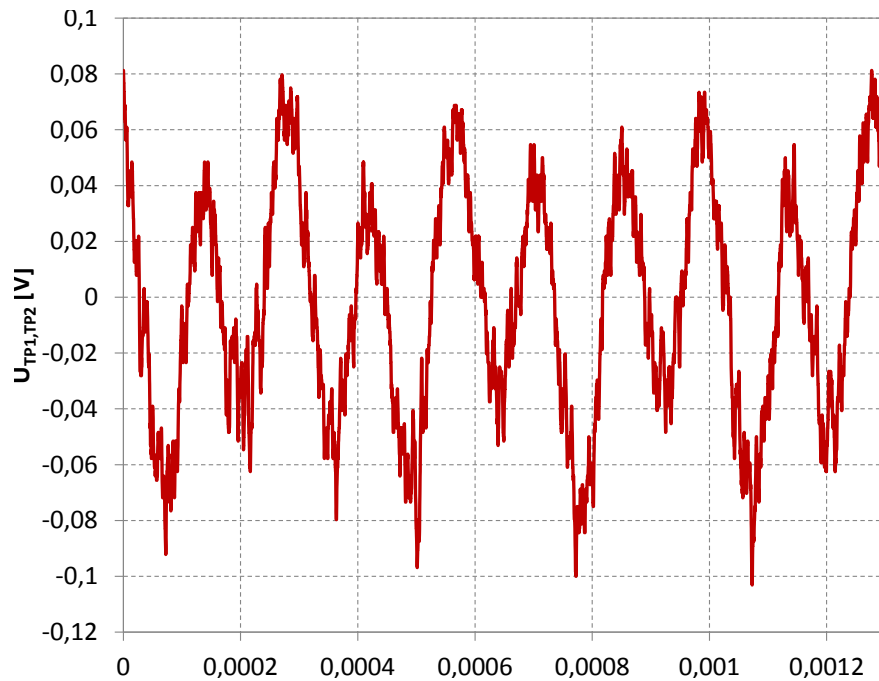


**Abbildung 5.7:** Oszilloskopbild des Ausgangssignals des APD Modules für das 3 kHz Modulationssignal



**Abbildung 5.8:** Oszilloskopbild des Ausgangssignals des APD Modules für das 7 kHz Modulationssignal



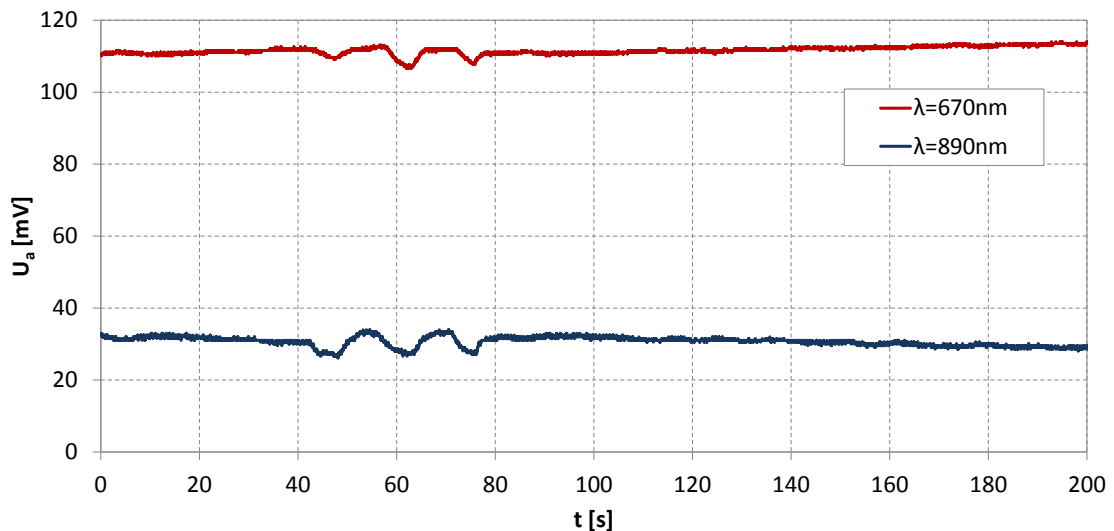


**Abbildung 5.9:** Oszilloskopbild des Ausgangssignals des APD Modules wenn beide Modulationssignale (3 kHz und 7 kHz) eingekoppelt werden

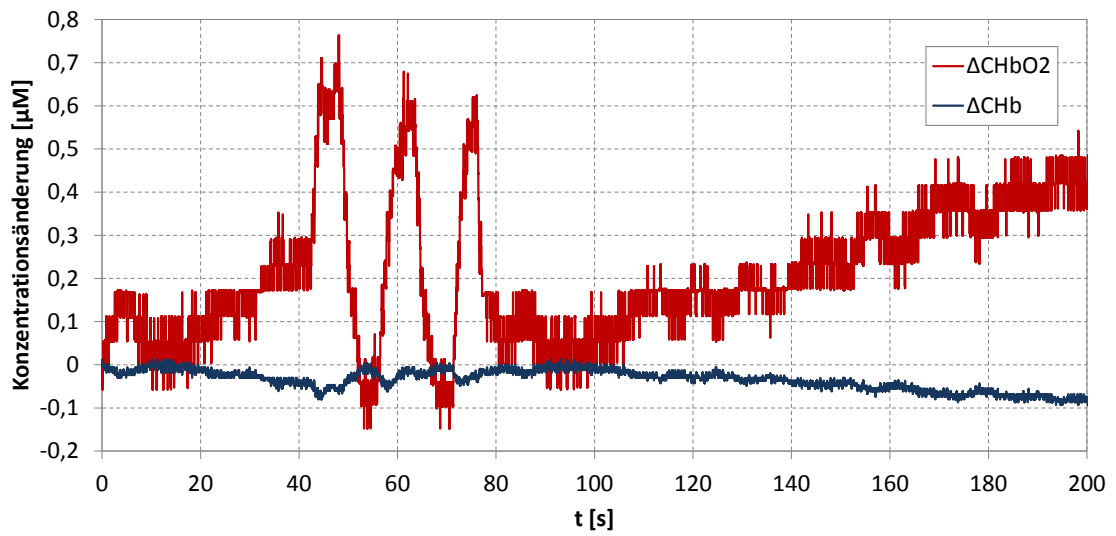
Nach dieser ersten Überprüfung wurden nun zwei Tasks gemessen. Zum Einen ein Kopfbewegungstask und zum Anderen ein Rückatmungstask. Es wurden hier bewusst zwei starke Tasks, d.h. die Signalveränderungen sind relativ groß, gewählt, um zu Beginn die Signalanalyse zu vereinfachen. Für die Messung der beiden Tasks wurde nun das APD Modul mit einem der beiden Messsignalpfade verbunden. Die Ausgangsspannungen an den entsprechenden Ausgangsbuchsen X7 und X3 (siehe ③ und ⑤ in Abbildung 4.2) des Lock-in Verstärkers wurden mit Hilfe einer DAQ Karte und eines Laptops in Matlab aufgezeichnet und ausgewertet. Die Samplerate für die Aufzeichnung über die DAQ Karte lag bei 100 Hz. Der erste Task, welcher ausgeführt wurde, war langsames, weites Kopfnicken. Hierdurch verändert sich die Lage des Gehirns innerhalb des Schädels sowie die des Volumens des Liquors, Absorption und Streuung des Lichtes werden durch die Veränderung der Dicke der Flüssigkeitsschicht verändert und somit ergibt sich ein Messsignal, welches in Abbildung 5.10 dargestellt ist.

Eine Berechnung der Konzentrationsänderungen von Oxy- bzw. Deoxyhämoglobin aus diesen Messwerten mit Matlab ergibt das Diagramm, welches in Abbildung 5.11 zu sehen ist.

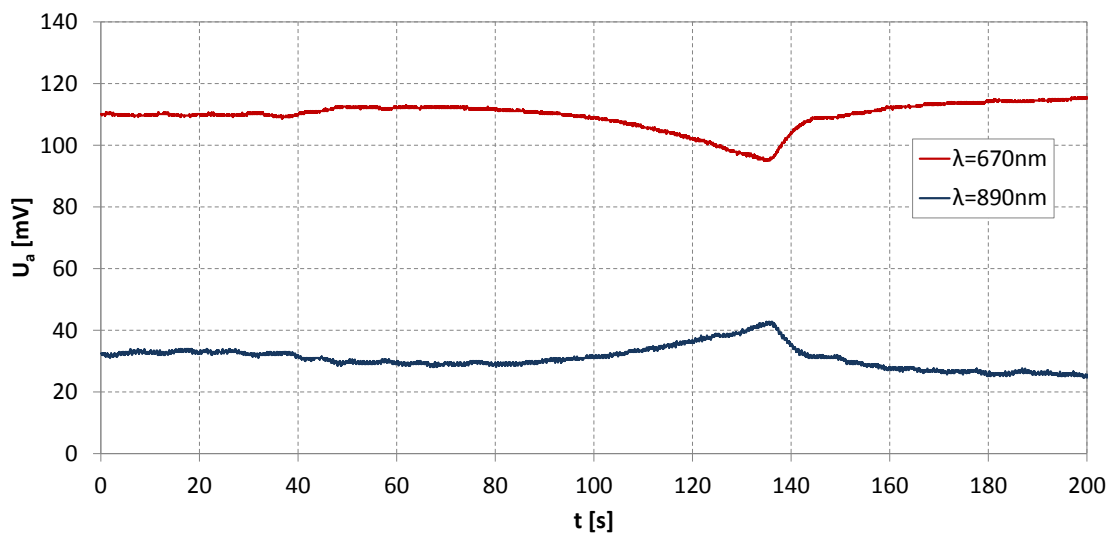
Ein zweiter Task, der aufgenommen wurde, ist ein Rückatmungstask. Hierbei atmet der Proband während der Messung in einen Kunststoffbeutel. Es entstehen hier typische Signalverläufe, welche in Abbildung 5.12 (Ausgangsspannungen des Lock-in Verstärkers) sowie 5.13 (Konzentrationsänderungen) dargestellt sind. Vorwiegender Effekt ist hierbei, dass durch das Atmen in den Beutel langsam der vorhandene Sauerstoff verbraucht wird. Der Körper verlangt nach mehr Sauerstoff, als eigentlich aufgrund der niedrigen Sauerstoffsättigung im Volumen des Beutels vorhanden ist. Aus diesem Grund sinkt die  $HbO_2$  Konzentration und die  $Hb$  Konzentration steigt, der Körper versucht dies durch Erhöhung der Vasodilatation sowie durch erhöhte Atemfrequenz zu kompensieren, was jedoch aufgrund des geschlossenen Volumens nicht zum gewünschten Ergebnis führt. Nach Beenden der Rückatmung wird das  $CO_2$  im Blut wieder abgeatmet, frischer Sauerstoff wird aufgenommen und das Ausgangsniveau der  $Hb$  sowie der  $HbO_2$  Konzentration normalisiert sich wieder.



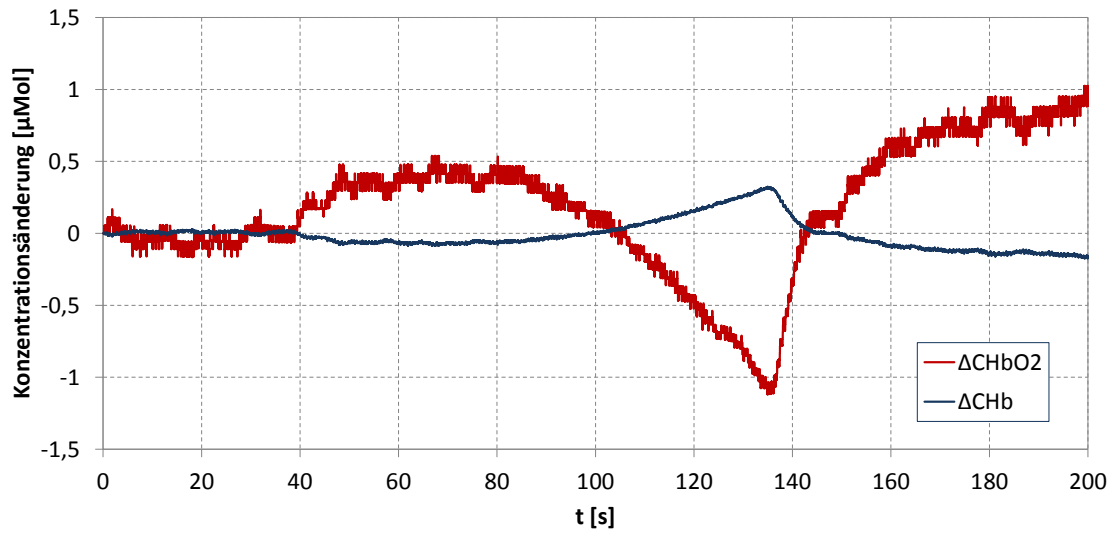
**Abbildung 5.10:** Ausgangssignale des Lock-in Verstärkers während des Kopfbewegungstasks



**Abbildung 5.11:** Darstellung der Konzentrationsänderungen bei Kopfbewegung



**Abbildung 5.12:** Darstellung der Ausgangsspannungen des Lock-in Verstärkers während des Rückatmungstasks



**Abbildung 5.13:** Darstellung der Konzentrationsänderungen während des Rückatmungstasks

# Kapitel 6

## Diskussion, Zusammenfassung und Ausblick

### Inhalt

---

6.1	Diskussion der Ergebnisse . . . . .	95
6.2	Zusammenfassung und Ausblick . . . . .	99

---

### 6.1 Diskussion der Ergebnisse

#### 6.1.1 Signalgenerator

Wie in den Abbildungen 3.10, 3.11, 3.12 und 3.13 zu sehen ist, funktionierten die Rekonstruktionsfilter für die gewünschten Frequenzen von 2 kHz, 3 kHz, 5 kHz sowie 7 kHz sehr gut. Jedoch ist gerade bei den beiden höheren Frequenzen (5 kHz bzw. 7 kHz) eine gewisse Restwelligkeit zu erkennen. Dies ist darauf zurückzuführen, dass, da für alle Sinussschwingungen die Lookuptabelle gleich lang ist, bei einer Samplingrate von 125 kHz beim 5 kHz Sinus nur noch 25 Samples und beim 7 kHz Sinus gar nur mehr 17.8 Samples pro Periode übrigbleiben. Um diese Restwelligkeit noch weiter zu glätten, müsste hier die Grenzfrequenz des Rekonstruktionsfilters etwas tiefer gelegt und zusätzlich die Ordnung, d.h. die Steilheit der Filter, erhöht werden.

Wie ebenfalls in den erwähnten Diagrammen zu sehen ist, erfahren die Sinusschwingungen am positiven Amplitudenmaximum eine Abflachung. Dies entsteht dadurch, dass softwareintern mit Fixkommazahlen im 1.15 Format gearbeitet wird, d.h. der minimalste negative Wert einer solchen Zahl ist -1, der maximalste positive Wert 0.999969482. Deshalb kann hier nicht der volle Sinusbereich von -1 bis +1 ausgeschöpft werden, was jedoch aufgrund der Präzision des 1.15 Formats (0.000305176) vernachlässigt werden kann.

Wie der Arbeit von Bauernfeind [12] zu entnehmen ist, kann jede der LEDs auf den Modulationsplatinen bis zu 80 mA an Strom ziehen. Diesen Strom muss der Spannungsregler IC27 schnell genug bereitstellen. Wie sich herausstellte wird aufgrund der sinusförmigen Stromaufnahme, das gesamte 12 V Versorgungsnetz der Hauptplatine mit ca.  $\pm 20$  mV moduliert. Trotz Pufferung mittels Stützkondensatoren setzt sich diese Modulation bis zu den APD-Modulen fort. Dies hat nun zur Folge, dass die Ausgangsspannung der APD-Module schon mit ca. 20 mV sinusförmig moduliert ist, obwohl noch kein Lichtsignal aufgenommen wird. Im Endeffekt führt diese Tatsache zu einer gewissen Offsetspannung im berechneten Ausgangssignal des Lock-in Verstärkers. Durch Filterung der Versorgungsspannung oder durch Aufbau eigener Spannungsversorgungen mittels Linearregler auf jeder der Modulationsplatinen könnte diesem Problem entgegengewirkt werden.

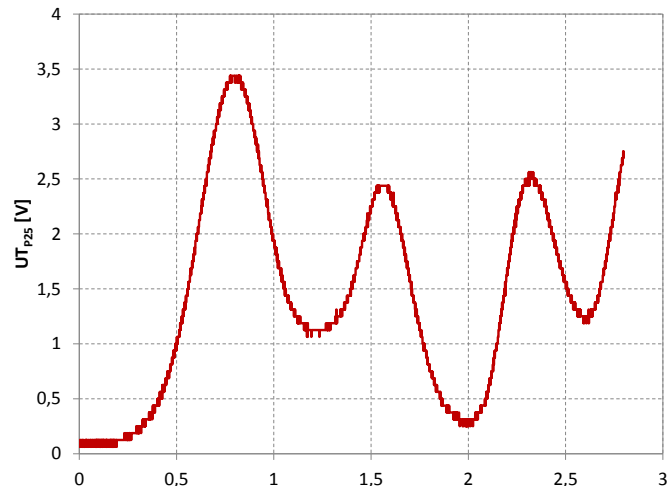
### 6.1.2 Spannungsversorgung

Bei der Auslegung der Spannungsversorgung wurde speziell darauf geachtet überall im sicheren Bereich zu sein. Als besonders schwierig stellten sich, aufgrund der unterschiedlichsten Herstellerangaben in den Datenblättern, die Stromverbrauchsabschätzungen heraus. Deshalb wurde überall etwas an Sicherheitspolster einberechnet, was wiederum zu einer leichten Überdimensionierung der Spannungsversorgung führt. Auch aufgrund der fehlenden Verfügbarkeit bestimmter Bauelemente mussten an dieser Stelle Kompromisse eingegangen werden. Hier könnte durch Messungen am Prototypen bzw. durch etwas kleinere Sicherheitspölster die Auslegung der Spannungsversorgung etwas effizienter gestaltet werden.

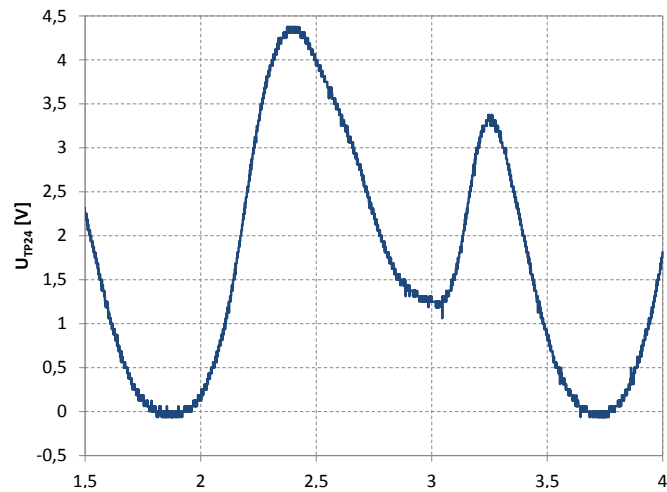
### 6.1.3 Ausgangssignalpfad

Bei der Beschreibung des analogen Ausgangssignalpfades in Abschnitt 3.1.4 wurde erwähnt, dass dieser Schaltungsteil für eine maximale Ausgangsspannung von ca. 8 V ausgelegt wurde. Wie sich im Zuge der Testmessungen jedoch herausstellte, und wie es auch in den Diagrammen in Abbildung 5.10 und 5.12 zu erkennen ist, ist nur eine maximale Spannung von ca. 110 mV möglich. Eine weitere Erhöhung der Eingangsverstärkung führt in diesem Fall auch nicht zu einer höheren Ausgangsspannung, weil hier schon die OPVs des Messsignalpfades an ihren Aussteuerungsgrenzen sind und übersteuern würden. Es liegt nun also die Vermutung nahe, dass hier die Dimensionierung des digitalen Tiefpassfilters für den Lock-in Algorithmus etwas unglücklich gewählt wurde und somit Probleme bereitet.

In einem weiteren Arbeitsschritt wurde der digitale Tiefpassfilter redimensioniert und die vorliegenden Präzisionsfehler ausgebessert. Es wurde wiederum eine Testmessung in einem abgedunkelten Raum durchgeführt. Hierbei wurde nur die direkte Übertragung zwischen Sende- und Empfangskopf (d.h. ohne Gewebe dazwischen) gemessen, indem Sende- und Empfangskopf in einem Abstand von einem Meter gegenüber aufgebaut wurden. Der Sendekopf wurde nun um seine vertikale Achse verdreht, sodass der Empfangskopf mit unterschiedlichen Beleuchtungsstärken angestrahlt wurde. Es zeigte sich, dass nun die maximale Ausgangsspannung weitaus höher liegt, als im vorherigen Fall. Die beiden Abbildungen 6.1 und 6.2 veranschaulichen das Ergebnis dieser Messungen. Somit kann davon ausgegangen werden, dass auch im Fall einer Messung durch Gewebe hindurch eine höhere Ausgangsspannung zu erwarten ist.



**Abbildung 6.1:** Oszilloskopbild des Ausgangssignals am analogen Ausgang 7 ( $\lambda = 670$  nm)



**Abbildung 6.2:** Oszilloskopbild des Ausgangssignals am analogen Ausgang 5 ( $\lambda = 890$  nm)

#### 6.1.4 Software

Der gesamte Algorithmus, also auch die Tiefpassfilter, sind in Floatingpointarithmetik ausgeführt. Tests haben ergeben, dass, Aufgrund der Samplerate und der benötigten niederen Grenzfrequenz der Tiefpassfilter, Festkommaarithmetik, auch mit 32 Bit Genauigkeit, nicht zu gebrauchen ist, weil die Quantisierungs- und Rundungsfehler zu viel Einfluss auf die Filtercharakteristik haben. Der Vorteil der größeren Genauigkeit durch



die Verwendung von Floatingpointarithmetik wird jedoch durch erhöhten Rechenaufwand erkauft. Da die Blackfin-Prozessoren keine Floatingpoint-Unit besitzen, müssen Floatingpointoperationen in Software emuliert werden, was zu einer höheren Rechenzeit führt. Im Prototypen des Lock-in Verstärkers werden innerhalb des Algorithmus nur zwei der vier möglichen Messfrequenzen demoduliert. Aufgrund der Floatingpointimplementierung könnte es bei der Erweiterung um die zwei fehlenden Frequenzen zu Problemen mit der Rechenleistung des Prozessors kommen. Eine Lösung dieses Problems bestünde in der Erhöhung der Blockgröße der zu Verarbeitenden Datenblöcke, eine andere Lösung wäre eine Geschwindigkeitsoptimierung des Softwaredesigns.

## 6.2 Zusammenfassung und Ausblick

Wie diese Dokumentation darlegt, konnte im Laufe dieser Masterarbeit ein äußerst praktikables, modulares Lock-in Messsystem zur Verwendung in einem BCI entwickelt werden. Mit diesem Prototypen können vier modulierte Lichtsignale ausgegeben werden. Die implementierten Frequenzen sind 2 kHz, 3 kHz, 5 kHz und 7 kHz. Des Weiteren wurde der Lock-in Verstärker mit zwei voneinander unabhängigen Messkanälen ausgestattet. Softwareseitig kann der Prototyp des Lock-in Verstärkers Messsignale mit den zwei Trägerfrequenzen 3 kHz und 7 kHz demodulieren, wobei besonders auf eine für den Benutzer einfache Bedienung Wert gelegt wurde. Diese erfolgt über Ethernet und SSH. Die berechneten Messwerte werden analog über vier BNC Buchsen ausgegeben.

Zukünftige Weiterentwicklungen des Gerätes könnten soft- sowie auch hardwaremäßige Änderungen beinhalten. Auf der Softwareseite ist hier allem voran die Implementierung der zwei fehlenden Trägerfrequenzen im Lock-in Algorithmus zu nennen. Weiters ist es denkbar, auch die Ausgangsverstärkung der einzelnen Kanäle per Software einstellbar zu machen. Auch die Erkennung, welcher der Steckplätze für die Modulationsplatinen nun belegt ist, wurde bei der Prototypenentwicklung ausgelassen. Eine weitere Evolutionsstufe dieses Verstärkers würde die Entwicklung einer grafischen Bedienoberfläche, eventuell in Kombination mit Matlab, darstellen, welche auch die Implementierung der digitalen Messwertausgabe beinhaltet.

Bei einer Revision der Hardware könnte die in der Diskussion angesprochene Problematik der schwankenden Spannungsversorgung auf den Modulationsplatinen Ziel der Weiterentwicklung sein.

## Literaturverzeichnis

- [1] ANALOG DEVICES. ADSP-BF52x Blackfin® Processor Hardware Reference.
- [2] ANALOG DEVICES. Analog Devices-Blackfin uClinux wiki. <http://docs.blackfin.uclinux.org>, 03.2012. Abgerufen am: 18.03.2012.
- [3] ANALOG DEVICES. Engineer Zone - Das U-Boot for Blackfin. <http://ez.analog.com/community/dsp/software-and-development-tools/uboot-blackfin>, 03.2012. Abgerufen am: 19.03.2012.
- [4] ANALOG DEVICES. Fully Accurate, 12-/14-/16-Bit VOUT nanoDAC, Quad, SPI Interface, 4.5 V to 5.5 V in TSSOP; AD5024/AD5044/AD5064 Datasheet: Datasheet, 05/2011.
- [5] ANALOG DEVICES. Analog Devices Open Source Koop - Blackfin Toolchain. <http://blackfin.uclinux.org/gf/project/toolchain/frs>, 05.2012. Abgerufen am: 27.05.2012.
- [6] ANALOG DEVICES. V/5 V, Rail-to-Rail Quad, 8-Bit DAC AD7304/AD7305: Datasheet, 11.2004.
- [7] ANALOG DEVICES. Ultralow Noise, LDO XFET Voltage References with Current Sink and Source, ADR440/ADR441/ADR443/ADR444/ADR445 Datasheet: Datasheet, 11.2010.
- [8] ANALOG DEVICES. Low Noise, Rail-to-Rail, Differential ADC Driver AD8139 Datasheet: Datasheet, 2007.
- [9] ANALOG DEVICES. 16-Bit, 1.5 LSB INL, 500 kSPS PulSAR™ Differential ADC in MSOP/QFN: AD7688 Datasheet: Datasheet, 2011.
- [10] ANALOG DEVICES. Digital Potentiometer Linux Driver. <http://wiki.analog.com/resources/tools-software/linux-drivers/misc/dpot>, 2012. Abgerufen am: 11.04.2012.
- [11] ANALOG DEVICES. EngineerZone - Linux Distribution for Blackfin. <http://ez.analog.com/community/dsp/open-source/linux-blackfin>, 2012. Abgerufen am: 18.03.2012.

- [12] BAUERNFEIND, G. *Nah-Infrarot-Spektroskopie am Menschen: Entwicklung eines Nah-Infrarot-Spektroskopie Systems (NIRS-System) für eine zukünftige Verwendung als optisches Brain-Computer Interface*. VDM Verlag Dr. Müller, Saarbrücken, 2008.
- [13] BLUETECHNIX. Hardware User Manual CM-BF527 V2.x: Datasheet, 18.10.2011.
- [14] BRINKER, T., KUPRIS, G., AND DEGENHARDT, H. *Embedded Linux: Praktische Umsetzung mit uClinux*. VDE Verl, Berlin [u.a.], 2007.
- [15] BRUNE, C. Introduction to Das U-Boot, the universal open source bootloader. <http://www.linuxfordevices.com/c/a/Linux-For-Devices-Articles/Introduction-to-Das-UBoot-the-universal-open-source-bootloader>, 08.2004. Abgerufen am: 19.03.2012.
- [16] COYLE, S. M., WARD, T. E., AND MARKHAM, C. M. Brain-computer interface using a simplified functional near-infrared spectroscopy system. *Journal of Neural Engineering* 4, 3 (2007), 219–226.
- [17] ECLIPSE. Eclipse Helios Sr2 Packages. <http://www.eclipse.org/downloads/packages/release/helios/sr2>, 2010. Abgerufen am: 27.05.2012.
- [18] FECHNER-SCHULTE LEITERPLATTENTECHNIK GMBH. Layout-Tipps – Elektrische Bemessungsrichtlinien. <http://www.fs-leiterplatten.de/technik/layout-tipps/elektrische-bemessungsrichtlinien/>, 2012. Abgerufen am: 17.03.2012.
- [19] GAN, W.-S., AND KUO, S. M. *Embedded signal processing with the micro signal architecture*. Wiley-Interscience, Hoboken and N.J., 2007.
- [20] HAMAMATSU PHOTONICS. APD module C5460 series Datasheet: Datasheet, 2010.
- [21] HARTL, H. *Elektronische Schaltungstechnik: Mit Beispielen in PSpice*. Pearson Studium, München [u.a.], 2008.
- [22] KERRISK, M. *The Linux programming interface: A Linux and UNIX system programming handbook*. No Starch Press, San Francisco, 2010.
- [23] KINGBRIGHT. T-1 (3mm) CYLINDRICAL LED LAMP Datasheet: Datasheet, 15.09.2010.

- [24] LINEAR TECHNOLOGIES. LTSpice IV, 2012.
- [25] MARCELLIS, A. D., AND FERRI, G. *Analog circuits and systems for voltage-mode and current-mode sensor interfacing applications*. Springer, Dordrecht [Netherlands], 2011.
- [26] OKADA, E., FIRBANK, M., SCHWEIGER, M., ARRIDGE, S. R., COPE, M., AND DELPY, D. T. Theoretical and experimental investigation of near-infrared light propagation in a model of the adult head. *Applied optics* 36, 1 (1997), 21–31.
- [27] OPPENHEIM, A. V., AND SCHAFFER, R. W. *Zeitdiskrete Signalverarbeitung: Mit 19 Tabellen, 112 Beispielen und 403 Aufgaben*, 3 ed. Oldenbourg, München and Wien, 1999.
- [28] PRAHL, S. Optical Properties Spectra. <http://omlc.ogi.edu/spectra/>, 2001. Abgerufen am: 16.03.2012.
- [29] SYDENHAM, P., AND THORN, R., Eds. *Handbook of measuring system design*. Wiley, Chichester, 2005.
- [30] TEXAS INSTRUMENTS. 1A Low-Dropout Regulator with Reverse Current Protection Datasheet: Datasheet, 06.2012.
- [31] TRACO POWER. TEN 8 Series, 8Watt, DC/DC Converters Datasheet: Datasheet, 05.2011.
- [32] TRACO POWER. THD 15N Series, 15 Watt DC/DC Converters Datasheet: Datasheet, 07.2010.
- [33] WIKIPEDIA. Blackfin. <http://en.wikipedia.org/w/index.php?title=Blackfin&oldid=479733341>, 03.2012. Abgerufen am: 19.03.2012.
- [34] WIKIPEDIA. MClinux. <http://en.wikipedia.org/wiki/%CE%9Clinux>, 03.2012. Abgerufen am: 18.03.2012.
- [35] WOLPAW, J. R., BIRBAUMER, N., MCFARLAND, D. J., PFURTSCHELLER, G., AND VAUGHAN, T. M. Brain-computer interfaces for communication and control. *Clinical neurophysiology : official journal of the International Federation of Clinical Neurophysiology* 113, 6 (2002), 767–791.
- [36] XPPower. PDM30 Series, 30 Watts AC-DC Datasheet: Datasheet, 31.01.2012.

# Anhang A

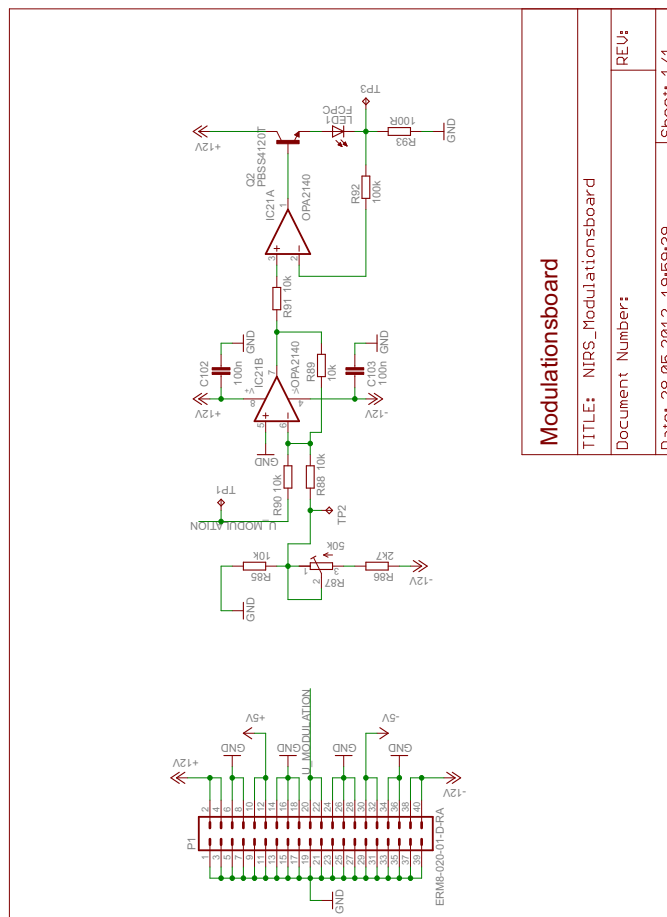
## Abkürzungsverzeichnis

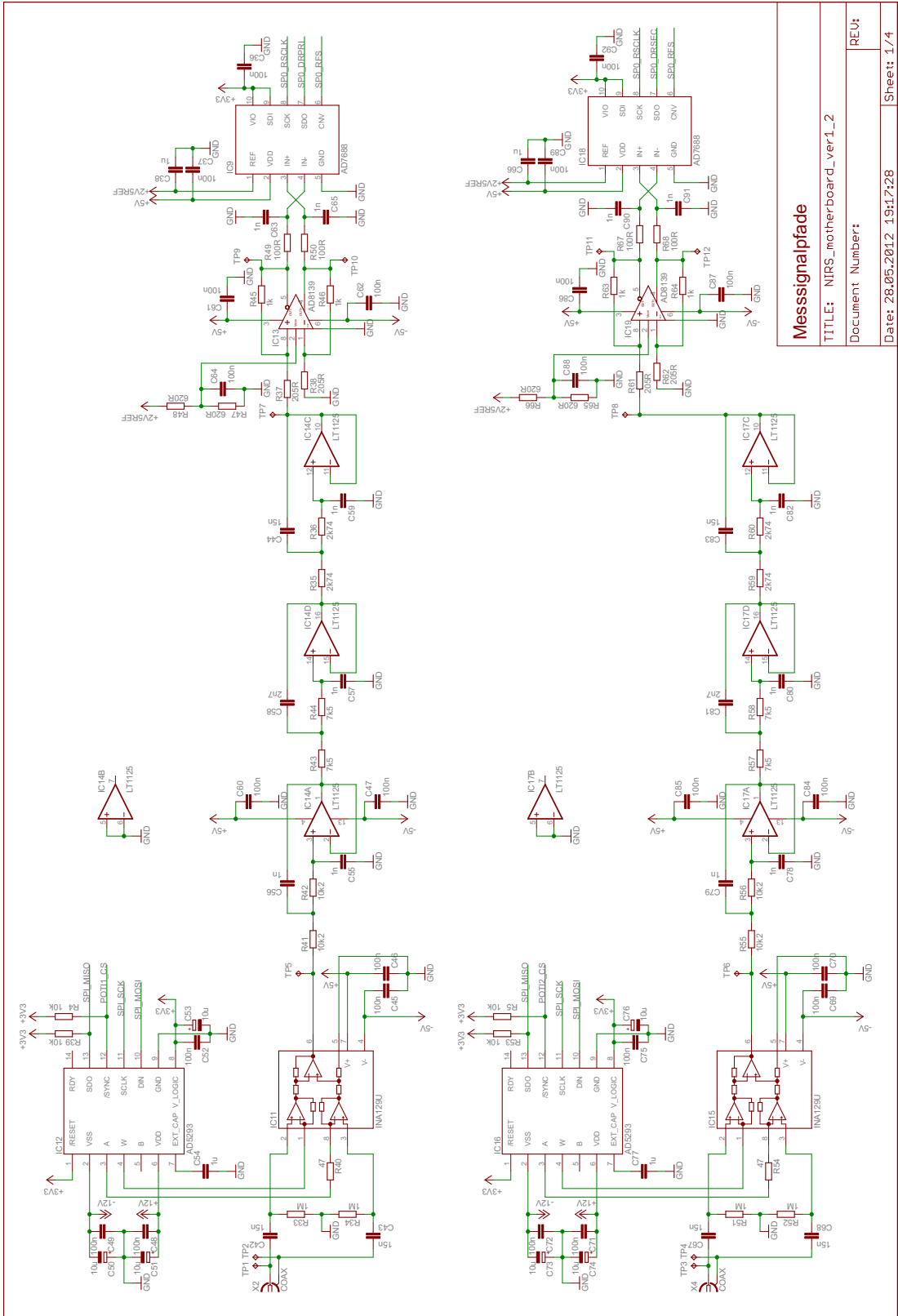
<b>ADC</b>	Analog-Digital-Wandler (Analog-Digital-Converter)
<b>APD</b>	Lawinenfotodiode (Avalanche Photo Diode)
<b>BCI</b>	Brain Computer Interface
<b>DAC</b>	Digital-Analog-Wandler (Digital-Analog-Converter)
<b>DMA</b>	Direct Memory Access
<b>DPF</b>	Differential Pathlength Factor
<b>GPIO</b>	General Purpose Input Output
<b>INL</b>	Integraler Nichtlinearitätsfehler (Integral Nonlinearity Error)
<b>JTAG</b>	Joint Test Action Group
<b>kSps</b>	Kilosamples pro Sekunde
<b>MMU</b>	Memory-Management-Unit
<b>MPU</b>	Memory-Protection-Unit
<b>NIRS</b>	Nahinfrarotspektroskopie
<b>SNR</b>	Signal-Rausch-Verhältnis (Signal-To-Noise Ratio)
<b>SPI</b>	Serial Peripheral Interface
<b>SPORT</b>	Serial Port
<b>TFTP</b>	Trivial File Transfer Protocol
<b>UART</b>	Universal Asynchronous Receiver Transmitter
<b>USB OTG</b>	USB On The Go
<b>OPV</b>	Operationsverstärker
<b>PSD</b>	phasensensitiver Detektor

# Anhang B

## Schaltpläne und Layout

Achtung, die Platinenlayouts wurden für die Darstellung dem Papierformat dieser Arbeit entsprechend skaliert!





Messsignalpfade

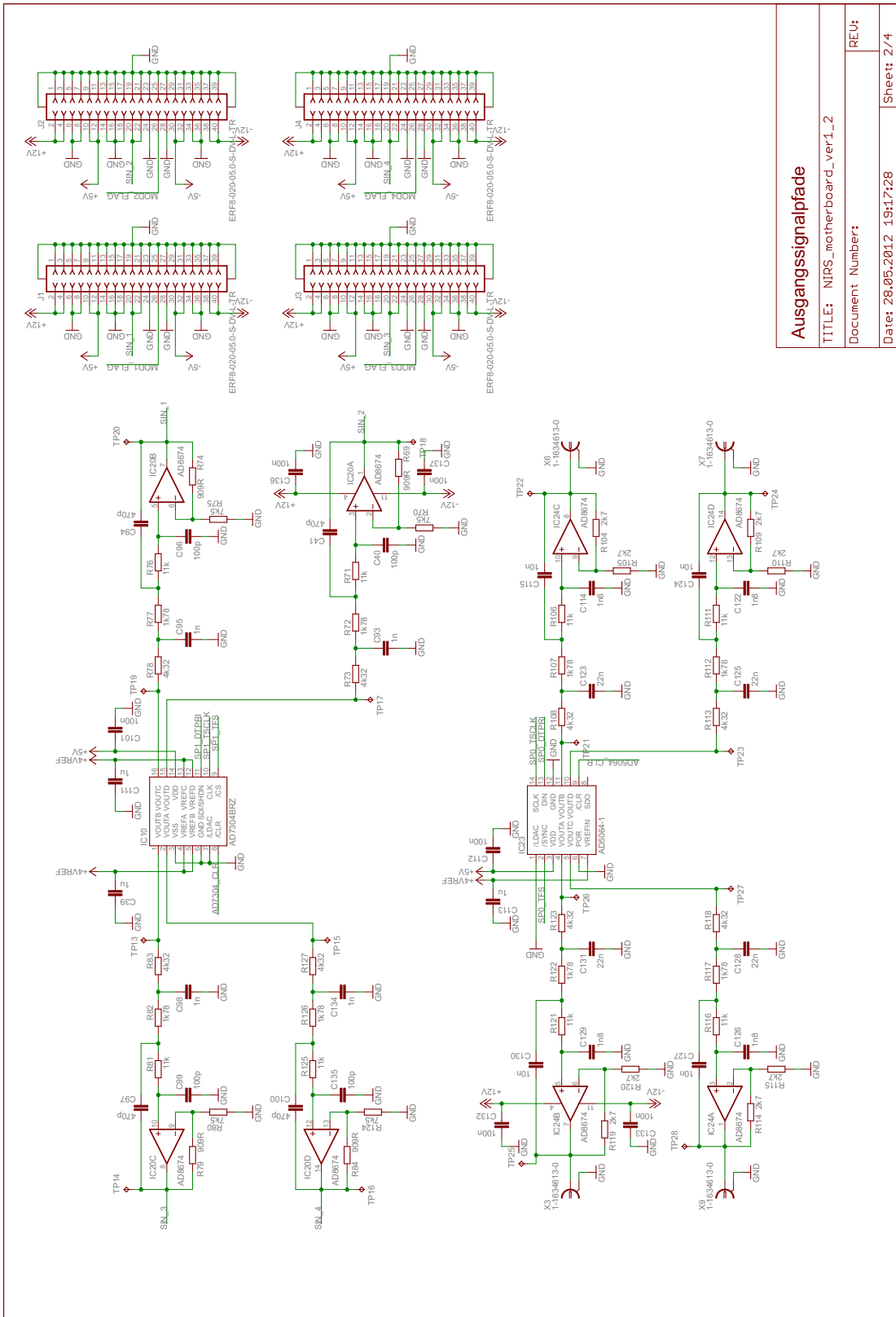
TITLE: NIRS\_motherboard\_ver1\_2

Document Number:

Date: 28.05.2012 19:17:28

REV:

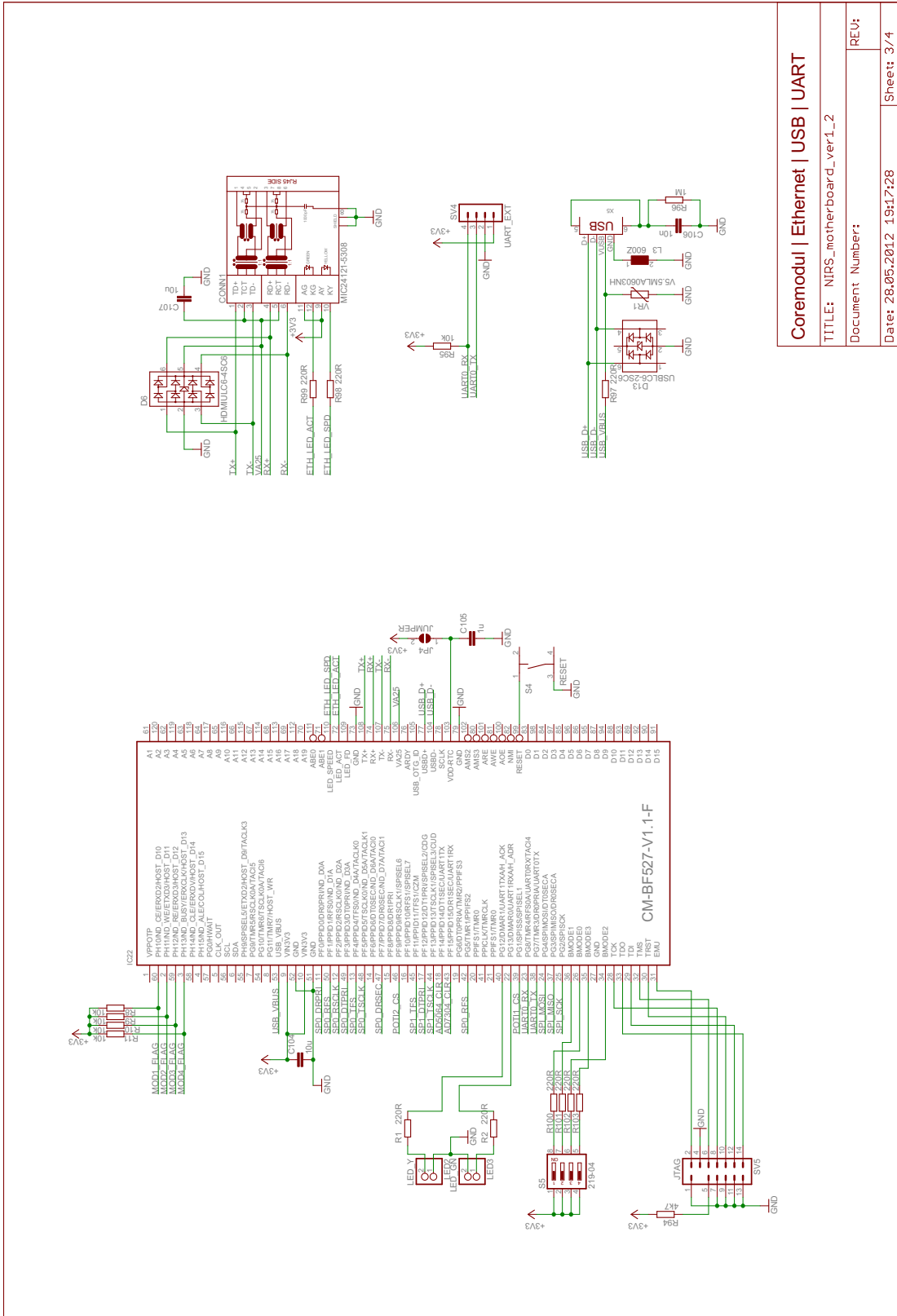
Sheet: 1/4



**Ausgangssignalfade**  
TITLE: NIRS\_motherboard\_ver1\_2  
Document Number:  
Date: 28.05.2012 19:17:28  
Sheet: 2/4

REU:





Coremodul | Ethernet | USB | UART

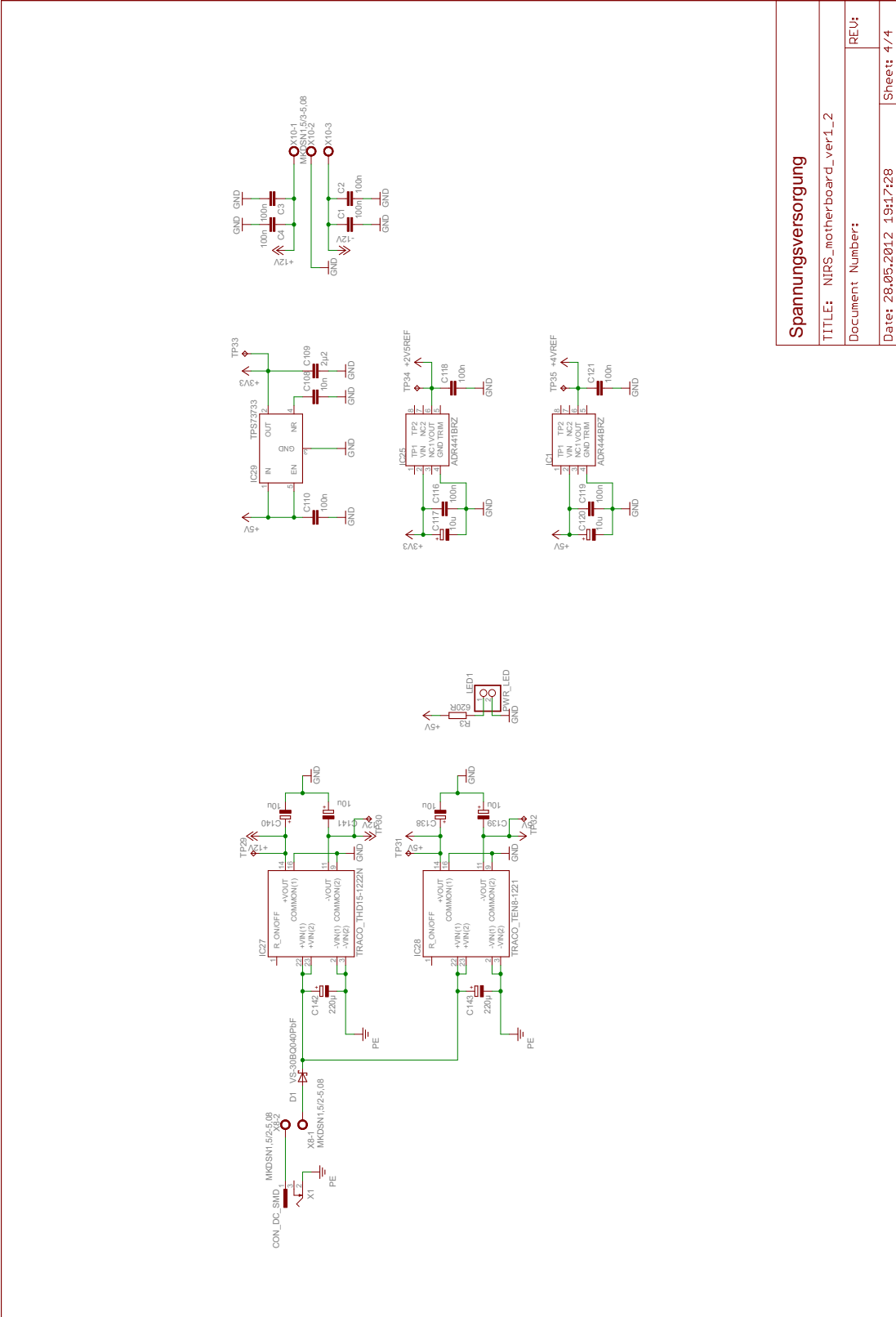
TITLE: NIRS\_motherboard\_ver1\_2

Document Number:

Date: 28.05.2012 19:17:28

Sheet: 3 / 4

REV:



**Spannungsversorgung**

TITLE: NIRS\_motherboard\_ver1\_2

Document Number:

Date: 28.05.2012 19:17:28

REU:

Sheet: 4/4

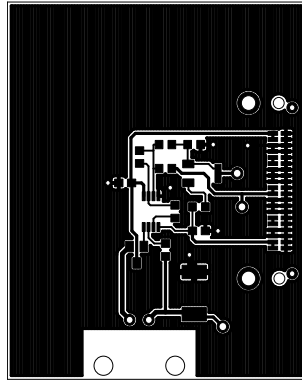


Abbildung B.1: Modulationsboard Top Layer (Maßstab 1:1)

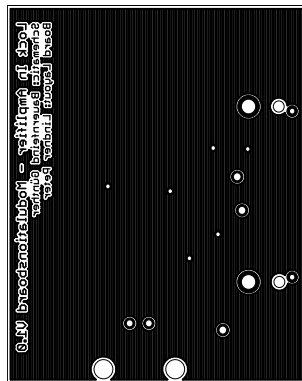


Abbildung B.2: Modulationsboard Bottom Layer (Maßstab 1:1)

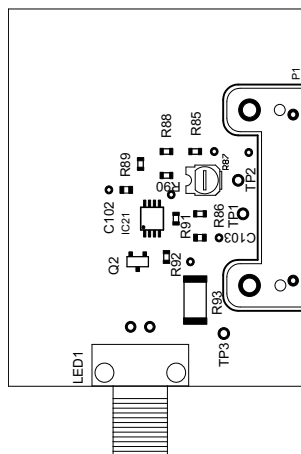


Abbildung B.3: Modulationsboard Bestückungsplan (Maßstab 1:1)

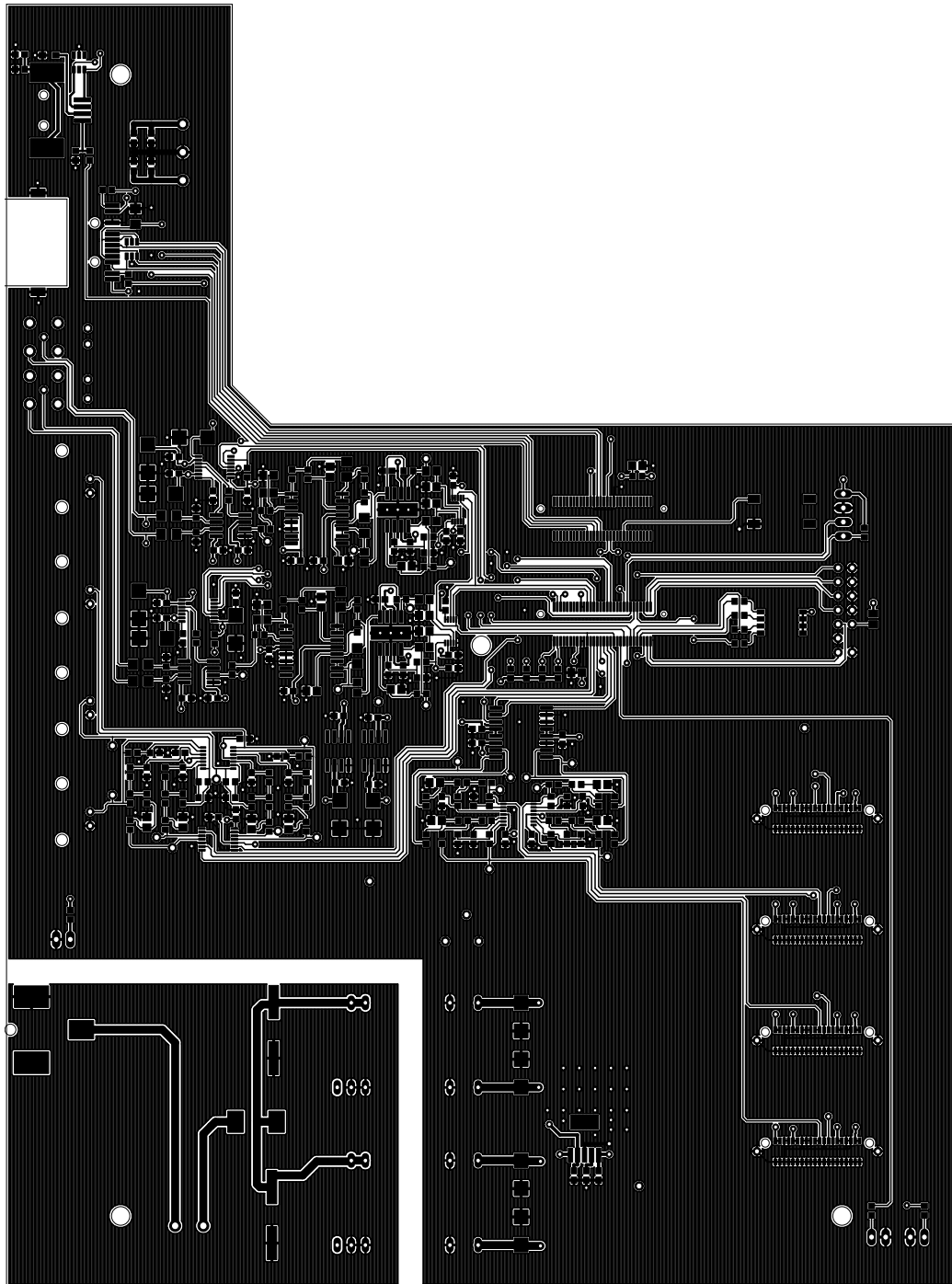


Abbildung B.4: Mainboard Ver: 1.0 Top Layer (Maßstab 1:1,25)

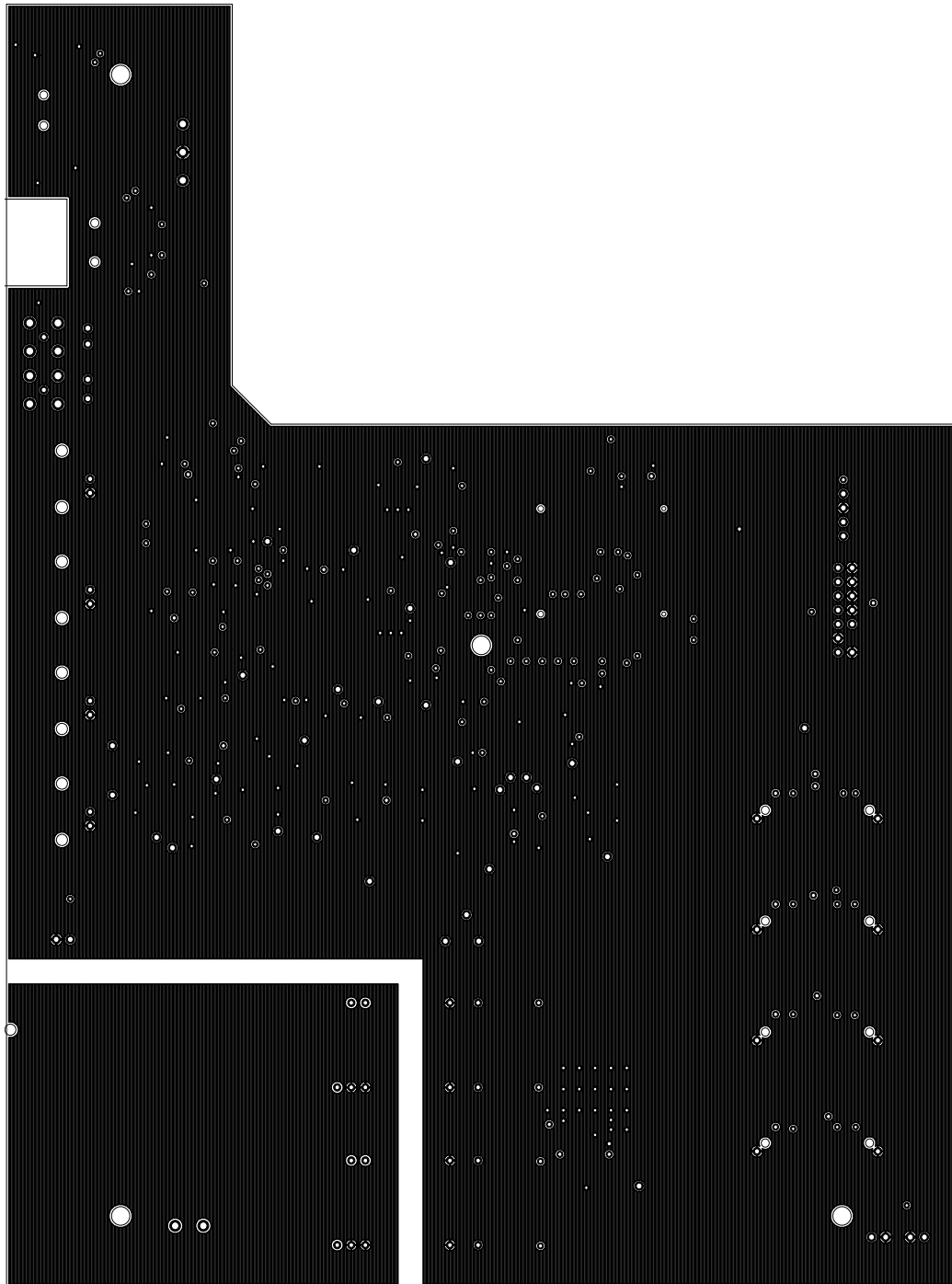


Abbildung B.5: Mainboard Ver: 1.0 GND Layer (Maßstab 1:1,25)

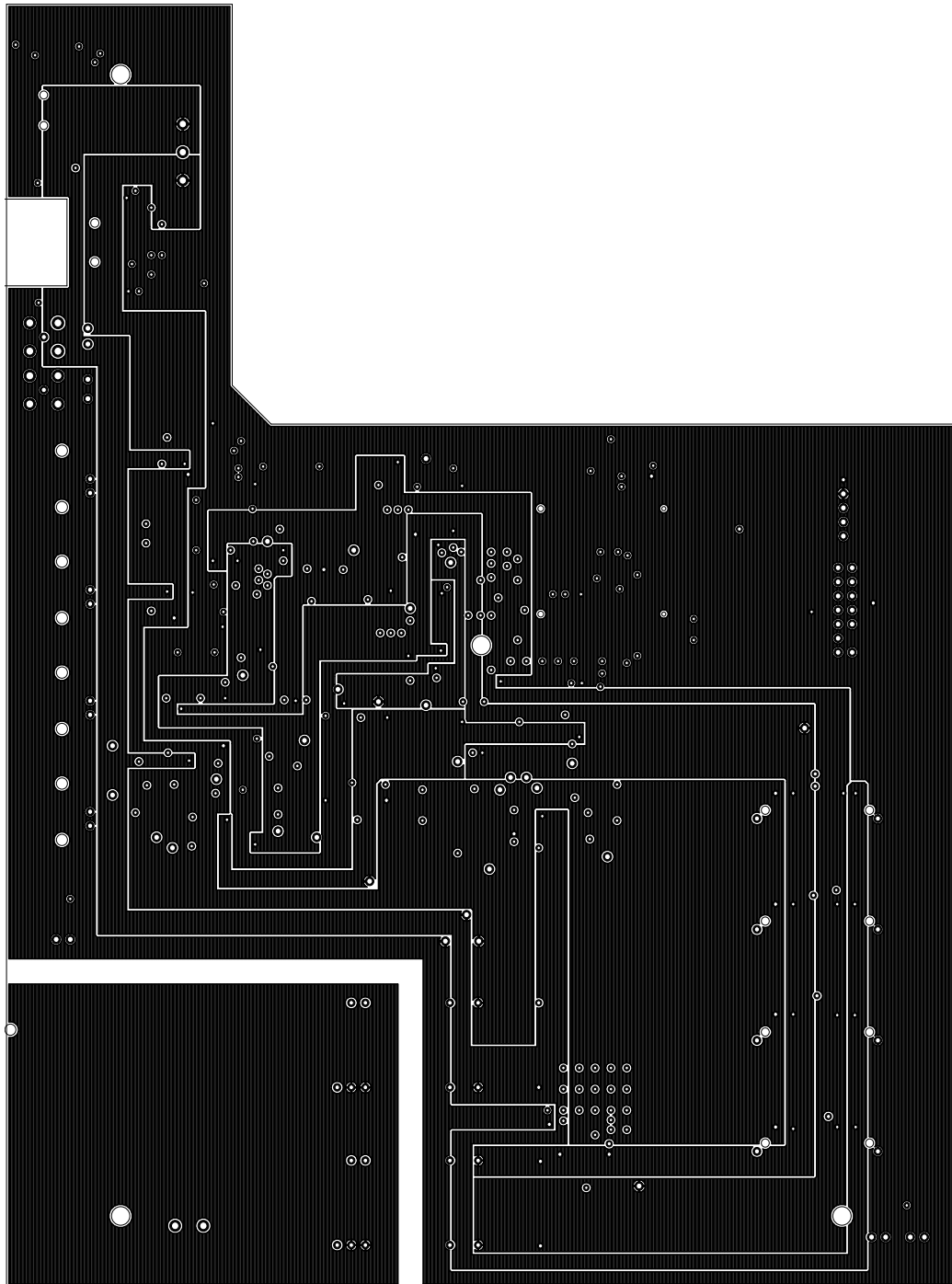


Abbildung B.6: Mainboard Ver: 1.0 VCC Layer (Maßstab 1:1,25)

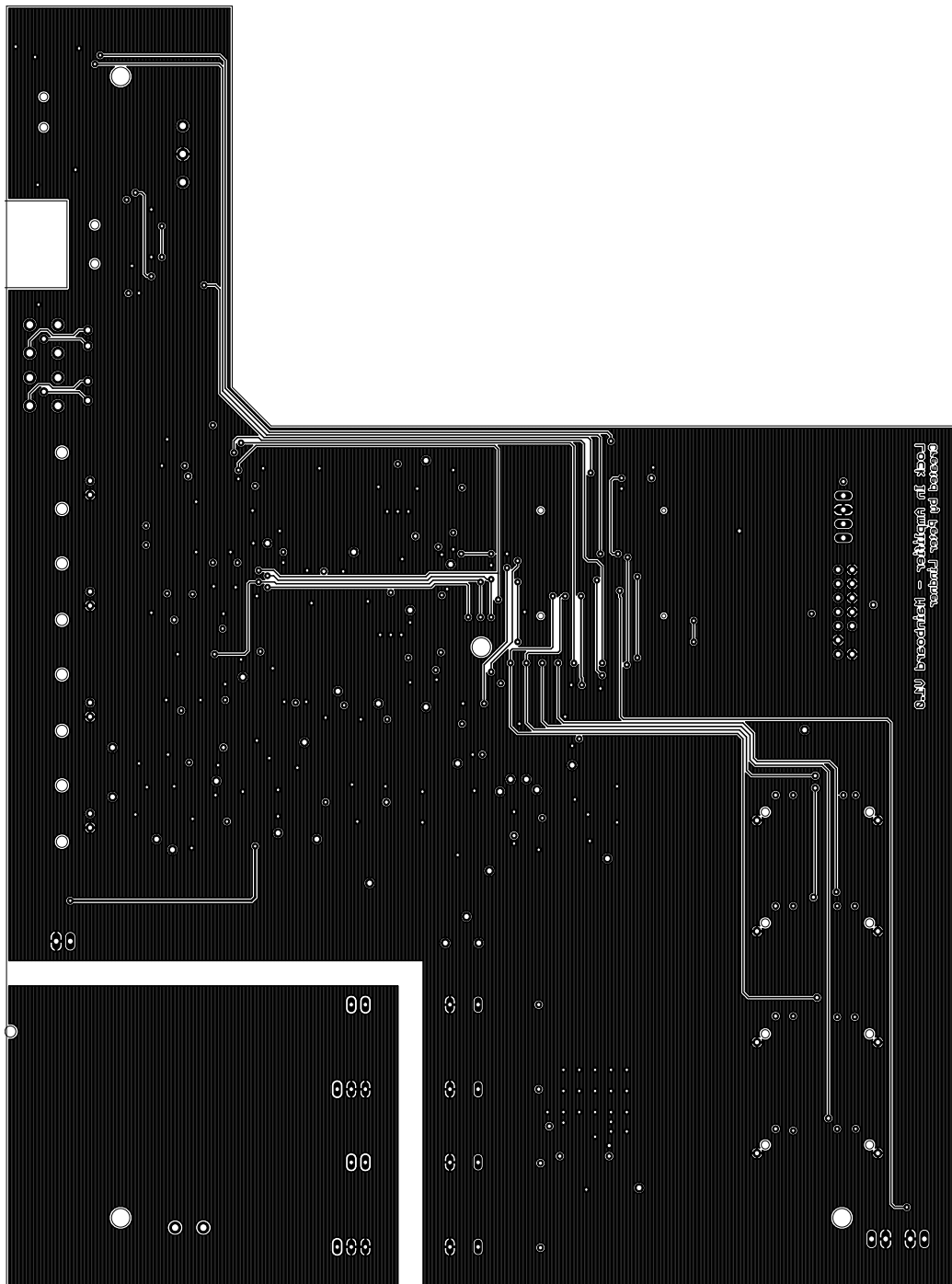


Abbildung B.7: Mainboard Ver: 1.0 Bottom Layer (Maßstab 1:1,25)

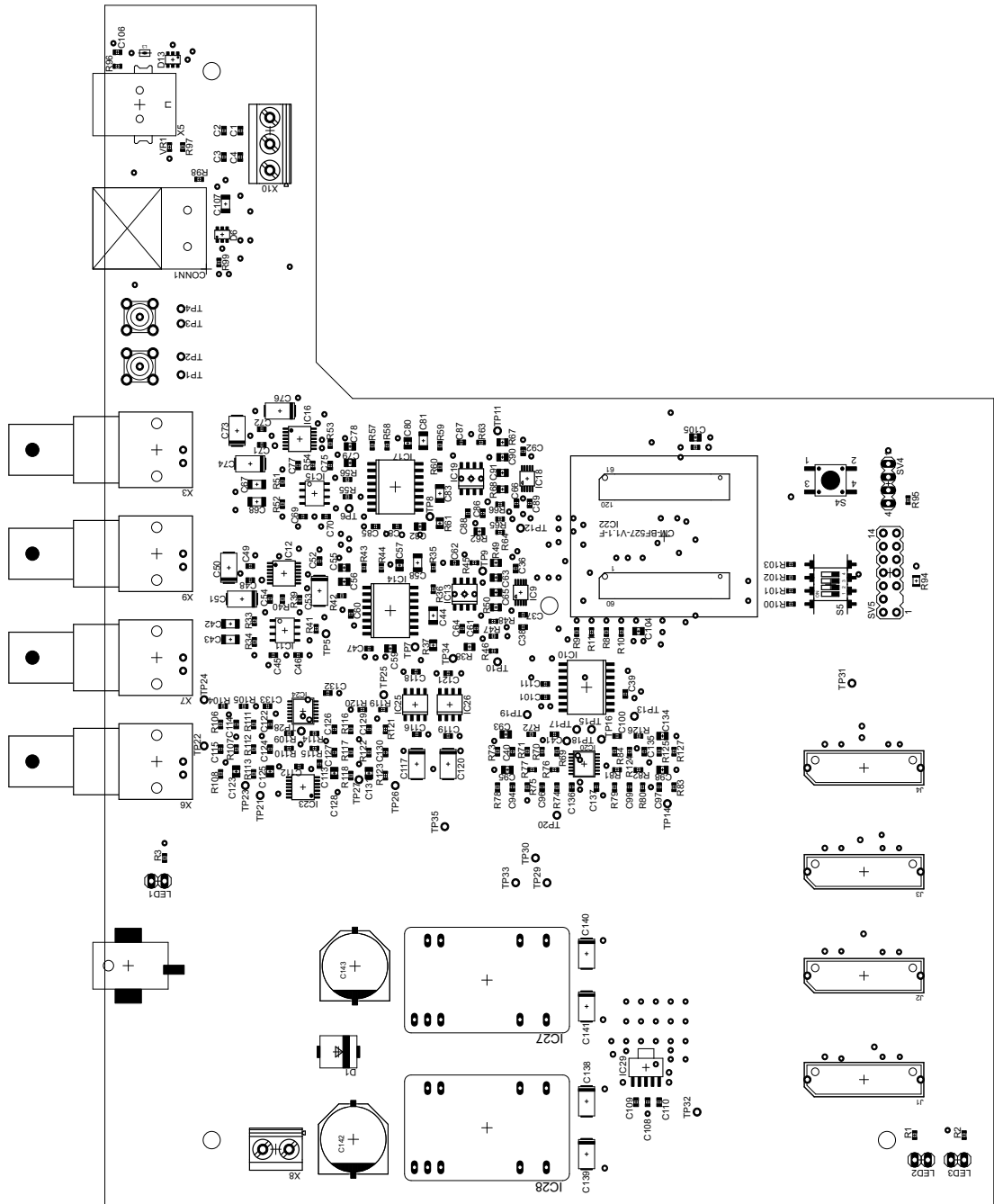


Abbildung B.8: Mainboard Ver: 1.0 Bestückungsplan (verkleinert)



# Anhang C

## Stückliste

### C.1 Farnell

www.farnell.at

Stk	Beschreibung	Bauteil	Best-Nr.	Daten-Blatt
12	Kond. tantal 10 $\mu$ F, 16 V, 10%, 2312	C53, C50, C51, C73, C74, C76, C117, C120, C138, C139, C140, C141	197270	D1
47	Kond. X7R, 100 nF, 50 V, 10%, 0603	C45, C46, C48, C49, C52, C47, C60, C61, C62, C64, C37, C36, C71, C72, C75, C69, C70, C84, C85, C88, C86, C87, C89, C92, C101, 4 x C102, 4 x C103, C112, C116, C118, C119, C121, C132, C133, C136, C137, C110, C1, C2, C3, C4	1288255	D2
1	Kond. X5R, 10 $\mu$ F, 16 V, 10%, 1206	C107	1611956	D3
29	Wid. 10 k $\Omega$ , 0.1 W, 1%, 0603	R39, R53, 4 x R85, 4 x R88, 4 x R89, 4 x R90, 4 x R91, R95, R4, R5, R8, R9, R10, R11	1469748	D4
5	Wid. 1 M $\Omega$ , 0.125 W, 1%, 0603	R33, R34, R51, R52, R96	1469746	D5
7	Kond. X5R, 1 $\mu$ F, 16 V, 10%, 0603	C54, C77, C38, C66, C39, C111, C113	1611954	D6
2	Wid. 47 $\Omega$ , 0.250 W, 1%, 0603	R40, R54	1738887	D7
2	Kond. Keramik 2.7 nF, 100 V, 1%, 1206	C58, C81	1692303	D8
16	Kond. Keramik 1 nF, 50 V, 1%, 0805	C55, C56, C57, C59, C63, C65, C79, C78, C80, C82, C90, C91, C93, C95, C98, C134	1740641	D9
6	Kond. Film 15 nF, 16 V, 2%, 1206	C42, C43, C44, C67, C68, C83	9694900	D10
4	Wid. 10.2 k $\Omega$ , 0.063 W, 0.1%, 0603	R41, R42, R55, R56	3989434	D11
8	Wid. 7.5 k $\Omega$ , 0.063 W, 0.1%, 0603	R43, R44, R57, R58, R70, R75, R80, R124	1506021	D12
4	Wid. 2.74 k $\Omega$ , 0.063 W, 0.1%, 0603	R35, R36, R59, R60	1160336	D13
4	Wid. 205 $\Omega$ , 0.125 W, 1%, 0805	R37, R38, R61, R62	1652952	D14
4	Wid. 1 k $\Omega$ , 0.063 W, 0.1%, 0603	R45, R46, R63, R64	1653252	D15
5	Wid. 620 $\Omega$ , 0.100 W, 0.1%, 0603	R47, R48, R65, R66, R3	1670173	D16
4	Wid. 100 $\Omega$ , 0.330 W, 1%, 0603	R49, R50, R67, R68	1738946	D17
4	Wid. 909 $\Omega$ , 0.063 W, 0.1%, 0603	R69, R74, R79, R84	1353195	D18
4	Kond. Keramik 100 pF, 250 V, 2%, 0603	C40, C96, C99, C135	1650746	D19
4	Kond. Keramik 470 pF, 200 V, 5%, 0603	C41, C94, C97, C100	1284117	D20
12	Wid. 2.7 k $\Omega$ , 0.100 W, 1%, 0603	4 x R86, R104, R105, R109, R110, R114, R115, R119, R120	1500617	D21
4	Pot. 50 k $\Omega$ , 0.150 W, 25%	4 x R87	1174182	D22
4	Wid. 100 k $\Omega$ , 0.063 W, 1%, 0603	4 x R92	1653254	D23
4	Wid. 100 $\Omega$ , 1 W, 1%, 2512	4 x R93	1653199	D24

8	Wid. 4.32 k $\Omega$ , 0.063 W, 0,1%, 0603	R108, R113, R118, R123, R73, R78, R83, R127	1160343	D25
8	Wid. 1.78 k $\Omega$ , 0.063 W, 0,1%, 0603	R107, R112, R117, R122, R72, R77, R82, R126	1160330	D26
8	Wid. 11 k $\Omega$ , 0.100 W, 0,1%, 0603	R106, R111, R116, R121, R71, R76, R81, R125	1400594	D27
4	Kond. Keramik 22 nF, 50 V, 5%, 0805	C123, C125, C128, C131	1828943	D28
4	Kond. Keramik 1.8 nF, 100 V, 5%, 0603	C114, C122, C126, C129	1740629	D29
5	Kond. Keramik 10 nF, 50 V, 5%, 0603	C115, C124, C127, C130, C108	1833871	D30
9	Wid. 220 $\Omega$ , 0.250 W, 1%, 0603	R98, R99, R97, R100, R101, R102, R103, R1, R2	1738895	D31
1	TVS Schutzdiode LAN, HDMILC6-4SC6, SOT-23	D6	1751962	D32
1	Varistor USB, 5.5 V, 0603	VR1	1816927	D33
1	TVS Schutzdiode USB, USBLC6-2SC6, SOT-23	D13	1269406	D34
1	Kond. Keramik 10 nF, 100 V, 10%, 0603	C106	1284119	D35
1	Ferrit 100 $\Omega$ , 500 mA, 0805	L3	1800363	D36
2	LED, Flachkopf, gn, 3 mm	LED1, LED3	1142496	D37
1	LED, Flachkopf, gb, 3 mm	LED2	1142500	D38
4	BNC Buchse, schwarz	X3, X6, X7, X9	1020956	D39
1	USB Buchse B, SMD	X5	1321918	D40
1	Kond. Keramik 2.2 $\mu$ F, 10 V, 10%, 0603	C109	1611957	D41
2	Kond. Aluminium El. 220 $\mu$ F, 63 V, 20%, SMD	C142, C143	9696059	D42
1	Schottkydiode 40 V, 3 A	D1	8647879	D43
1	Wid. 4.7 k $\Omega$ , 0.063 W, 1%, 0603	R94	9331247	D44
2	Instrumentierungsverst. INA129U, SOIC8	IC11, IC15	1212415	D45
2	Quad Opamp LT1125, SOIC16	IC14, IC17	1330582	D46
2	ADC Treiber AD8139ARDZ, SOIC8	IC13, IC19	8397864	D47
2	ADC AD7688BRMZ, MSOP10	IC9, IC18	1117897	D48
2	Quad Opamp AD8674ARZ, SOIC14	IC20, IC24	1581959	D49
4	NPN Transistor PBSS4120T, SOT-23	4 x Q2	8736162	D50
1	Spannungsreferenz 2.5 V ADR441BRZ, SOIC8	IC25	1274158	D51
1	Spannungsreferenz 4.096 V ADR444BRZ, SOIC8	IC1	1274151	D52
1	Spannungsregler 3.3 V TPS73733DCQ, SOT-223	IC29	1703384	D53
4	Dual Opamp OPA2140, MSOP8	4 x IC21	1855120	D54
1	Quad DAC AD7304YRZ, SOIC16	IC10	1438404	D55
1	Quad DAC AD5065BRUZ-1, TSSOP14	IC23	2067838	D56
1	DC/DC Wandler 8 W, $\pm$ 5 V, $\pm$ 800 mA, TEN8-1221	IC28	1205019	D57
1	Schraubklemme 2-polig	X8	3041440	D58
1	Schraubklemme 3-polig	X10	3041451	D59
1	Wippschalter 16 A		7892527	D60
2	Buchsen Coremodul, Hirose FX8-60S-SV(21)		1324656	D61
4	Buchsen Modulationsplatinen, Samtec ERF8-020-05.0-S-DV	J1, J2, J3, J4	1667633	D62
1	Dipswitch 4-fach	S5	1123953	D63
1	Stiftleiste	SV4	1022262	D64
2	Coax Stecker APD-Module	X2, X4	1753815	D65
1	Desktopnetzteil Medical 30 W		1625687	D66
1	Coremodul CM-BF527	IC22	2079856	D67
1	Stiftleiste zweireihig	SV5	1022242	D68
1	Resettaster	S4	3801287	D69

## C.2 RS Components

[www.rs-components.at](http://www.rs-components.at)

Stk	Beschreibung	Bauteil	Best-Nr.	Daten-Blatt
1	DC/DC Wandler 15 W, $\pm 12$ V, $\pm 625$ mA, THD 15-1222N	IC27	706-5720	D70
1	DC Buchse 2.5 mm, SMD	X1	476-157	D71

## C.3 Digikey

[www.digikey.at](http://www.digikey.at)

Stk	Beschreibung	Bauteil	Best-Nr.	Daten-Blatt
2	Digitales Potentiometer AD5293BRUZ-50, TSSOP14	IC12, IC16	AD5293BRUZ-50-ND	D72
1	Spannungsreferenz 4.096 V, SOIC-8	IC1	ADR444BRZ-ND	D73

## C.4 Würth

[www.we-online.de](http://www.we-online.de)

Stk	Beschreibung	Bauteil	Best-Nr.	Daten-Blatt
1	Ethernet Buchse, MIC24121-5308	CONN1	7498011122R	D74

## C.5 Lasercomponents

[www.lasercomponents.com](http://www.lasercomponents.com)

Stk	Beschreibung	Bauteil	Best-Nr.	Daten-Blatt
4	FCPC Buchse		74733004	

## C.6 Hamamatsu

[www.hamamatsu.com](http://www.hamamatsu.com)

Stk	Beschreibung	Bauteil	Best-Nr.	Daten-Blatt
2	LED L2656-03 890 nm			D75
2	LED L6112-01 670 nm			D76

## C.7 Pactec Enclosures

[www.pactecenclosures.com](http://www.pactecenclosures.com)

Stk	Beschreibung	Bauteil	Best-Nr.	Daten-Blatt
1	Kunststoffgehäuse BM-3 Kit schwarz		75419-510-000	D77

## C.8 Samtec

[www.samtec.com](http://www.samtec.com)

Stk	Beschreibung	Bauteil	Best-Nr.	Daten-Blatt
4	Stecker Modulationsplatinen, Samtec ERM8-020-01-D-RA		ERM8-020-01-D-RA	D78

