

Entwicklung eines Datenerfassungssystems auf Raspberry Pi-Basis



Entwicklung und Evaluierung eines prototypischen Messdatenerfassungssystems basierend auf Raspberry Pi und Digilent MCC DAQ HATs mit oszilloskop- und netzähnlichen Funktionen

ohm Technische Hochschule Nürnberg

Projektarbeit im Masterstudiengang: Elektronische und Mechatronische Systeme (MSY)

Projektziel:	Mit Hilfe eines Raspberry Pi's und Digilent MCC DAQ HAT soll ein Messdaten-Erfassungssystem erstellt werden (ähnlich einem MyDAQ von NI, das über LabVIEW gesteuert wird).
Projektaufgabe:	<ul style="list-style-type: none">• Konzepterstellung• Programmierung mit Python/ Jupyter Notebooks• Standard-Messroutinen als Jupyter Notebooks• Analoge Eingänge für Messungen, Analoge Ausgänge als Spannungsversorgung (mit kleiner Leistung)• Realisierung der Peripherie• Konstruktion eines Gehäuses mit Planung aller Anschlüsse und Stecker
Projektbetreuer:	Prof. Dr. Sebastian Walter
Projektteilnehmer:	Atil, Erol MSY M.Eng Bao, Changlai MSY M.Eng Nguepeusi, Will MSY M.Eng Schmidt, Sven MSY M.Eng
Projektanmeldung:	14.03.2025
Projektende:	15.09.2025

Inhaltsverzeichnis

I	Abbildungsverzeichnis.....	5
II	Tabellenverzeichnis	8
III	Abkürzungsverzeichnis.....	9
1	Projekteinführung	10
1.1	Projektbeschreibung	10
1.2	Organigramm	12
2	Projektplanung.....	13
2.1	Kostenplanung und Logistik	13
2.2	Entstandene Kosten im Rahmen der Projektarbeit.....	14
2.3	Systemanforderungen.....	15
2.4	Zeitplanung im Projekt.....	16
2.5	Stand der Technik.....	17
2.5.1	Die Messvorrichtung MyDAQ von National Instruments [4] [5]	17
2.5.2	Raspberry Pi 5 [6] [7].....	19
2.5.3	Diligent MCC 118 DAQ HAT [8].....	22
3	Projektentwicklung	25
3.1	Technische Grundlagen	25
3.1.1	Strommessung mit Shunt.....	25
3.1.2	OPV - Grundschaltungen.....	26
3.1.3	GPIOs - Raspberry Pi	27
3.1.4	SPI-Schnittstelle.....	27
3.2	Konzeptionierung	28
3.2.1	Funktionsgliederung eines Labornetzteils	28
3.2.2	Funktionsgliederung eines Oszilloskops.....	30
3.2.3	Lösungskonzept.....	33
4	Hardwareentwicklung	34
4.1	Multimeter	34
4.1.1	Spannungsmessung.....	34
4.1.2	Strommessung	34
4.2	Oszilloskop	45
4.3	Netzteil.....	47
4.4	Das Funktionsgenerator Modul	51
4.4.1	Spezifikationen des AD9833 Signalgenerator-Modul.....	52
4.4.2	Konzeptentwicklung zur Anpassung des Spannungsbereiches.....	52

4.4.3	Schaltplan des Funktionsgenerator Moduls in LTspice.....	53
4.5	Funktionssimulation mit LTspice und Simulationsergebnisse	58
4.5.1	Simulationsergebnisse für den AD8479	58
4.5.2	Simulationsergebnisse für den LT1352	60
4.5.3	Anmerkung zum Funktionsgeneratormodul.....	63
4.6	Diodenkennlinie	64
4.6.1	Messgenauigkeit und Fehlerabschätzung	64
4.7	Filterkennlinie	67
5	Softwareentwicklung	70
5.1	Einführung in die Software.....	70
5.2	Software-Architektur.....	71
5.2.1	Auswahl der Programmiersprache: Warum Python?.....	71
5.2.2	Verwendete Bibliotheken und Tools	72
5.2.3	Warum Jupyter Notebooks?	73
5.2.4	Modulare Struktur und Integration.....	75
5.3	Implementierung der Funktionen.....	76
5.3.1	Multimeter	76
5.3.2	Funktionsgenerator.....	78
5.3.3	Oszilloskop.....	80
5.3.4	Netzteil	81
5.3.5	Kennlinienmessung	84
5.3.6	Integration mit Dashboard.....	89
5.4	Installation und Anleitung.....	91
6	Design - Schaltplan.....	94
7	Reale Messungen HW – Vergleich zu Simulation.....	97
7.1	Spannungsmessung	97
7.1.1	Gleichspannung.....	97
7.1.2	Wechselspannung	98
7.2	Netzteil.....	101
7.3	Oszilloskop	103
7.4	Funktionsgenerator.....	108
7.5	Diodenkennlinie	112
7.6	Filterkennlinie	115
8	Vergleich OurDAQ vs. NI myDAQ	119
9	Design - Leiterplatte	122
10	Konstruktion von Gehäuse und Komponenten.....	127

10.1	CAD-Modellierung.....	127
10.2	Gehäuse Konzept	128
10.3	Einzelteil- und Montagebeschreibung	132
10.3.1	Grundplatine	132
10.3.2	Gehäuse Deckel.....	134
11	Projektstand und Ausblick.....	137
12	Anhang	139
13	Literaturverzeichnis.....	142

I Abbildungsverzeichnis

Abbildung 1.1: Produktentwicklungsprozess nach VDI 2221 [2]	11
Abbildung 1.2: V-Modell XT	12
Abbildung 1.3: Organigramm der Projektmitglieder in Form einer Divisionsorganisation.....	12
Abbildung 2.1: Gant - Diagramm des Projektes	16
Abbildung 2.2: MyDAQ von National Instruments.....	17
Abbildung 2.3: Raspberry Pi 5.....	19
Abbildung 2.4: Digilent MCC 118 DAQ HAT.....	22
Abbildung 3.1: Strommessung - Shunt + OPV [9, 10]	25
Abbildung 3.2: Grundschaltung - Nichtinvertierender Verstärker [11].....	26
Abbildung 3.3: Grundschaltung - Impedanzwandler [11].....	27
Abbildung 3.4: SPI-Kommunikationsverfahren [13].....	28
Abbildung 3.5: Funktionsgliederung eines Labornetzteils.....	30
Abbildung 3.6: Funktionsgliederung eines Oszilloskops	32
Abbildung 4.1: Strommessung OPA551 mA-Bereich	35
Abbildung 4.2:Strommessung OPA551 A-Bereich.....	35
Abbildung 4.3: Shunt-Spannung mA-Bereich.....	36
Abbildung 4.4: Verstärkte Spannung mA-Bereich.....	36
Abbildung 4.5: Shunt-Spannung A-Bereich.....	37
Abbildung 4.6:Verstärkte Spannung A-Bereich	37
Abbildung 4.7: Strommessung INA190A1 A-Bereich	38
Abbildung 4.8: Strommessung INA190A4 mA-Bereich	39
Abbildung 4.9: Laststrom A-Bereich.....	40
Abbildung 4.10: Ausgangsspannung A-Bereich (Messspannung für HAT-Modul)	40
Abbildung 4.11: Laststrom mA-Bereich	41
Abbildung 4.12: Ausgangsspannung mA-Bereich (Messspannung für HAT-Modul)	41
Abbildung 4.13: Umschaltung Messbereiche mit ADG419.....	42
Abbildung 4.14: Aufbau Netzteil positiv	47
Abbildung 4.15: DAC-Spannung vs. Netzteil positiv Ausgangsspannung.....	47
Abbildung 4.16: Aufbau Netzteil negativ	49
Abbildung 4.17:DAC-Spannung vs. Netzteil negativ Ausgangsspannung.....	49
Abbildung 4.18: Strombegrenzung Netzteil positiv	51
Abbildung 4.19: Strombegrenzung Netzteil negativ	51
Abbildung 4.20: AD9833 Signalgenerator-Modul	52
Abbildung 4.21: Schaltplan des Funktionsgenerator-Moduls zur Simulation der Funktionsfähigkeit in LTspice	54
Abbildung 4.22: Differenzverstärker [8]	55
Abbildung 4.23: Anpassung des Spannungsbereiches mittels AD8479 Differenzverstärker und der Multiplexer ADG419 (Blockschatzbild 1 – 3 des Funktionsgenerator-Moduls)	56
Abbildung 4.24: Schaltplanabschnitt: 2-stufige Verstärker Schaltung (Blockschatzbild 4 der Funktionsgenerator Moduls)	57
Abbildung 4.25: Ausgangslage der LTspice Simulation mit Darstellung der 3 Signalformen mit den jeweiligen Parametern	58
Abbildung 4.26: Schalterlogik für den Multiplexer ADG419 von Analog Devices.....	58
Abbildung 4.27: Offset Sinussignal von AD8479	59
Abbildung 4.28: Offset Dreiecksignal von AD8479	59

Abbildung 4.29: Offset Rechtecksignal von AD8479 (Im Vergleich größer als bei den Signalformen Sinus und Dreieck)	59
Abbildung 4.30: Eingangssignal Sinus und dessen 2-stufige Verstärkung.....	60
Abbildung 4.31: Messung des Sinussignals.....	60
Abbildung 4.32: Quantitative Messung des Sinussignals (Diff: Cursor2 - Cursor1)	60
Abbildung 4.33: Eingangssignal Dreieck und dessen 2-stufige Verstärkung	61
Abbildung 4.34: Messung des Dreiecksignals.....	61
Abbildung 4.35: Quantitative Messung des Dreieckssignals (Diff: Cursor2 - Cursor1)	61
Abbildung 4.36: Eingangssignal Rechteck und dessen 2-stufige Verstärkung	62
Abbildung 4.37: 1. Messung des Rechtecksignals zur Bestimmung der Amplituden	62
Abbildung 4.38: Quantitative Messung des Rechtecksignals (Diff: Cursor2 - Cursor1)	62
Abbildung 4.39: Prototypischer Aufbau des Funktionsgenerator Moduls.....	63
Abbildung 5.1: DC-Spannung Messung.....	77
Abbildung 5.2: Datenaufzeichnung in DMM.....	77
Abbildung 5.3: Konfiguration für Messmodus in DMM	78
Abbildung 5.4: Konfiguration für Wellenform in DMM.....	78
Abbildung 5.5: Konfiguration für Funktionsgenerator	80
Abbildung 5.6: Konfiguration für Oszilloskop.....	81
Abbildung 5.7: Netzteilsteuerung für positive Seite	83
Abbildung 5.8: Netzteilsteuerung für negative Seite	84
Abbildung 5.9: Diodenkennlinie.....	86
Abbildung 5.10: Filterkennlinie	89
Abbildung 5.11: Dashboard-Oberfläche	90
Abbildung 6.1: Leiterplatte - Versorgungsspannungen.....	94
Abbildung 6.2: Leiterplatte - Erzeugung ± 12 V.....	95
Abbildung 6.3: Leiterplatte - Schutzbeschaltung Eingänge.....	95
Abbildung 6.4: Leiterplatte - Erzeugung Referenzspannung 2,5 V.....	96
Abbildung 6.5: Leiterplatte - Anschlüsse für Schnittstellen	96
Abbildung 7.1: OurDAQ-SW - Messung Sinus (Amplitude = 0,5 V, f = 1 kHz)	99
Abbildung 7.2: Messung einer 1 kHz Sinus-Spannung (symmetrisch).....	104
Abbildung 7.3: Messung einer 2 kHz Sinus-Spannung (asymmetrisch).....	104
Abbildung 7.4: Messung einer 5 kHz Sinus-Spannung (asymmetrisch).....	104
Abbildung 7.5: Messung einer 10 kHz Sinus-Spannung (symmetrisch).....	105
Abbildung 7.6: Messung einer 500 Hz Dreiecksspannung (asymmetrisch)	105
Abbildung 7.7: Messung einer 2 kHz Dreiecksspannung (asymmetrisch)	106
Abbildung 7.8: Messung einer 4 kHz Dreiecksspannung (asymmetrisch)	106
Abbildung 7.9: Messung einer 6 kHz Dreiecksspannung (asymmetrisch)	106
Abbildung 7.10: Messung einer 500 Hz Rechteckspannung (asymmetrisch)	107
Abbildung 7.11: Messung einer 2 kHz Rechteckspannung (asymmetrisch)	107
Abbildung 7.12: Messung einer 4 kHz Rechteckspannung (asymmetrisch)	108
Abbildung 7.13: Messung einer 6 kHz Rechteckspannung (asymmetrisch)	108
Abbildung 7.14: Funktionsgenerator - Sinus (2 kHz).....	109
Abbildung 7.15: Funktionsgenerator - Sinus (10 kHz).....	109
Abbildung 7.16: Funktionsgenerator - Dreieck (2 kHz)	110
Abbildung 7.17: Funktionsgenerator - Dreieck (7 kHz)	110
Abbildung 7.18: Funktionsgenerator - Rechteck (3 kHz).....	111
Abbildung 7.19: Funktionsgenerator - Rechteck (5 kHz).....	111
Abbildung 7.20: OurDAQ-SW - aufgenommene Diodenkennlinie (15 Punkte).....	114

Abbildung 7.21: OurDAQ-SW - aufgenommene Diodenkennlinie (30 Punkte).....	114
Abbildung 7.22: Messung Filterkennlinie mit Oszilloskop (1880 Hz)	116
Abbildung 7.23: Messung ideale Filterkennlinie mit LTspice	117
Abbildung 7.24: OurDAQ-SW – aufgenommene Filterkennlinie (12 Punkte)	118
Abbildung 9.1: Leiterplatte - Top-Layer.....	122
Abbildung 9.2: Leiterplatte - Power-Plane	123
Abbildung 9.3: Leiterplatte - Ground-Plane	123
Abbildung 9.4: Leiterplatte - Bottom-Layer.....	124
Abbildung 9.5: Leiterplatte - 3D-Ansicht oben.....	125
Abbildung 9.6: Leiterplatte - 3D-Ansicht seitlich	125
Abbildung 9.7: Leiterplatte - 3D-Ansicht unten	126
Abbildung 10.1: Design Concept "Das Messdatenerfassungssystem - OurDAQ"	128
Abbildung 10.2: Analoge und Digitale In- und Outputs vom Raspberry Pi 5 und Digilent MCC 118 DAQ HAT	129
Abbildung 10.3: Leiterplatine: "Header-Platine"	129
Abbildung 10.4: Schaltplan der Header-Platine.....	129
Abbildung 10.5: Oberseite des Gehäuses mit den Anschlüssen	130
Abbildung 10.6: Anordnung der Peripherie Anschlüsse an der großen Leiterplatine	131
Abbildung 10.7: Freie Stelle des Gehäuses für die Peripherie des Raspberry Pi 5	131
Abbildung 10.8: Grundplatte des Gehäuses	132
Abbildung 10.9: Graphische Montage der Grundplatte.....	133
Abbildung 10.10: Montage des Raspberry Pi 5 zusammen mit dem Digilent MCC 118 DAQ HAT	134
Abbildung 10.11: Graphische Montage des Gehäuse Deckels (1)	135
Abbildung 10.12: Graphische Montage des Gehäuse Deckels (2)	135
Abbildung 10.13: Montage des Gehäuse Deckels an der Grundplatte	136

II Tabellenverzeichnis

Tabelle 2.1: Bestellungsliste im Projekt.....	14
Tabelle 2.2: Anforderungsliste des Messdatenerfassungssystems	15
Tabelle 4.1: Strommessung - Zusammenfassung.....	45
Tabelle 4.2: Oszilloskop - Übersicht Abtastraten in Abhängigkeit der Signalform	45
Tabelle 4.3: Oszilloskop - Genauigkeit Sinus	46
Tabelle 4.4: Netzteil – Übersicht Kühlkörper-Berechnung n-Kanal-MOSFET	48
Tabelle 4.5: Netzteil – Übersicht Kühlkörper-Berechnung p-Kanal-MOSFET	49
Tabelle 4.6: Ausgewählte Parameter aus dem Datenblatt des MyDAQ von NI [5]	51
Tabelle 4.7: Diodenkennlinie - Genauigkeit mit 0,1% Serienwiderstand	66
Tabelle 4.8: Diodenkennlinie - Genauigkeit mit 1,0% Serienwiderstand	66
Tabelle 4.9: Filterkennlinie - Genauigkeit bei unterschiedlichen Amplituden	68
Tabelle 4.10: Filterkennlinie - maximale Phasendifferenz.....	69
Tabelle 5.1: Softwarekomponente	70
Tabelle 5.2: Vergleich Raspberry Pi 3 und Raspberry Pi.....	71
Tabelle 5.3: Vergleich Programmiersprachen	72
Tabelle 5.4: Verwendete Bibliotheken	73
Tabelle 5.5: Alternative zu Jupyter Notebooks und deren Eigenschaften	74
Tabelle 7.1: Multimeter - reale Messergebnisse DC	98
Tabelle 7.2: Multimeter - reale Messergebnisse AC (Sinus).....	99
Tabelle 7.3: Multimeter - reale Messergebnisse AC (Dreieck).....	100
Tabelle 7.4: Multimeter - reale Messergebnisse AC (Rechteck).....	100
Tabelle 7.5: Netzteil positiv - reale Messergebnisse (1 kΩ Last)	101
Tabelle 7.6: Netzteil positiv - reale Messergebnisse (10 kΩ Last)	102
Tabelle 7.7: Netzteil negativ - reale Messergebnisse (1 kΩ Last)	102
Tabelle 7.8: Netzteil negativ - reale Messergebnisse (10 kΩ Last).....	103
Tabelle 7.9: Diodenkennlinie - reale Messergebnisse (15 Punkte)	112
Tabelle 7.10: Diodenkennlinie - reale Messergebnisse (30 Punkte)	113
Tabelle 7.11: Filterkennlinie - reale Messergebnisse	115
Tabelle 7.12: Filterkennlinie - simulierte Messergebnisse	116
Tabelle 8.1: Vergleich OurDAQ vs. NI myDAQ.....	120

III Abkürzungsverzeichnis

AC	Alternating Current
CAD	Computer-Aided Design
CPHA	Clock Phase
CPOL	Clock Polarity
CS	Chip Select
CSV	Comma-Separated Values
CPU	Central Processing Unit
DAC	Digital-Analog-Converter
DAQ	Data Acquisition
DC	Direct Current
DDS	Direct Digital Synthesis
DMM	Digitalmultimeter
FMCLK	Master Clock Frequency
FSYNC	Frame Synchronization
GND	Ground
GPIO	General Purpose Input/Output
HAT	Hardware Attached on Top
IP	Internet Protocol
LGPIO	Linux General Purpose Input/Output
MCC	Measurement Computing Corporation
MOSFET	Metal-Oxide-Semiconductor Field-Effect Transistor
OPV	Operationsverstärker
OS	Operating System
PCB	Printed Circuit Board
RAM	Random Access Memory
RMS	Root Mean Square
RPI	Raspberry Pi
RSS	Root Sum Square
SPI	Serial Peripheral Interface
UI	User Interface

1 Projekteinführung

1.1 Projektbeschreibung

Im Rahmen dieser Projektarbeit wird ein Produkt aus dem Bereich der Messtechnik entwickelt. Das Ziel besteht darin, ein kostengünstiges Datenerfassungssystem auf Open-Source-Basis zu entwickeln. Dieses soll auf einem herkömmlichen **Raspberry Pi** und einem **Digilent MCC DAQ HAT** basieren und die Funktionsspektren eines Oszilloskops, eines Labornetzteils und eines digitalen Multimeters weitgehend abdecken.

Das System soll in Funktionsumfang und Bedienkonzept mit dem NI myDAQ vergleichbar sein, jedoch vollständig auf frei verfügbarer Hard- und Software basieren. Mithilfe des Raspberry Pi als Recheneinheit und frei zugänglicher Python-Bibliotheken wird eine flexible und erweiterbare Plattform geschaffen, die nicht nur für den Einsatz im Messtechnik-Praktikum des Bachelorstudiengangs Elektrotechnik und Informationstechnik geeignet ist, sondern auch eine kostengünstige und nachhaltige Alternative zu proprietären Lösungen darstellt.

Gerade im Bereich der ingenieurwissenschaftlichen Ausbildung spielen praxisorientierte Laborübungen eine zentrale Rolle. Besonders in der Messtechnik ist es entscheidend, dass Studierende nicht nur theoretische Konzepte erlernen, sondern diese auch anhand realer Messungen, Signalverarbeitungen und Systemsteuerungen praktisch nachvollziehen können. Hierfür werden häufig multifunktionale Messgeräte wie das NI myDAQ eingesetzt, das in kompakter Form Funktionen wie Multimeter, Oszilloskop, Funktionsgenerator und weitere Messmöglichkeiten vereint. [1]

Der Einsatz solcher kommerzieller Systeme ist jedoch mit Einschränkungen verbunden, da sie nicht frei zugänglich sind. Einerseits sind diese Geräte kostenintensiv, was die Ausstattung größerer Studierendengruppen oder die Nutzung in Heimübungen erschwert. Andererseits basiert die Softwareumgebung auf einer geschlossenen Plattform, wodurch Studierende in ihrer Arbeit eingeschränkt sind. Erweiterungen der Gerätefunktionen, die Implementierung eigener Schnittstellen oder die individuelle Anpassung der Messumgebung an spezifische Versuchsanordnungen sind daher nicht möglich.

Für unsere Produktrealisierung werden Konzeptentwürfe nach **VDI 2221 „Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte“** (vgl. Abbildung 1.1.1) generiert, um die möglichst effizienteste Produktlösung zu finden und diese umzusetzen. Dafür werden Anhand des Lastenheftes vom Auftraggeber jeweilige Lösungskonzepte generiert und bewertet wodurch das finale Konzept durch das Projektteam entworfen und ausgearbeitet wird. Mit der Anwendung der VDI 2221 wird dabei das Ziel verfolgt im Projekt systematisch vorzugehen.

Mit der Auswahl des finalen Konzeptes erfolgt eine Unterteilung in die Abteilungen Elektronik, Konstruktion und Softwareentwicklung mit dem Ziel, dass die Abteilungen eigenständig nach Lösungskonzepten und den Jour-Fix Treffen gebunden, simultan die Bausteine für das Datenerfassungssystem entwickeln.

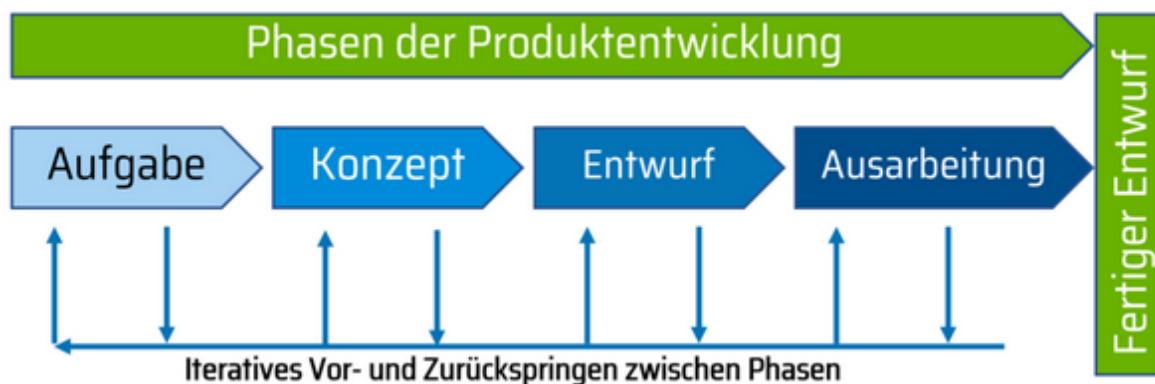


Abbildung 1.1: Produktentwicklungsprozess nach VDI 2221 [2]

Das Vorgehen der Softwareentwicklung für die Funktionsgenerierung wird wiederum nach dem **V-Modell XT** erfolgen, welches allgemein der Entwicklungsstandard für öffentliche Systementwicklungsprojekte der Bundesrepublik Deutschland ist. Dieses Modell ist der Nachfolger des V-Modells, wobei „XT“ für die Abkürzung „eXtreme Tailoring“ steht (siehe Abbildung 1.2).

Mit diesem Modell wird das Ziel verfolgt die Softwareentwicklung zum einen strukturiert vorzugehen und zum anderen immer noch flexibel zu sein, wenn es um Anpassungen und mögliche Ergänzungen geht.

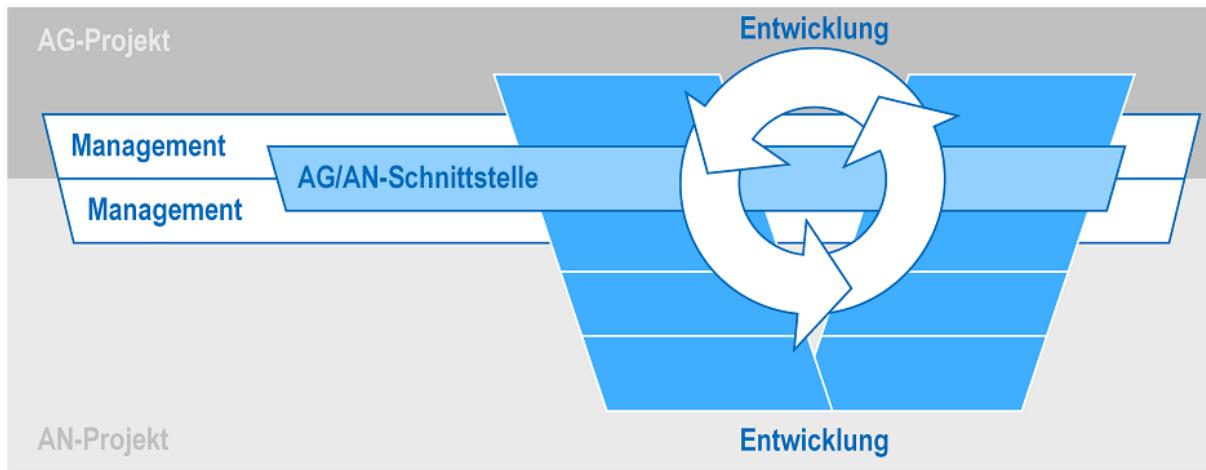


Abbildung 1.2: V-Modell XT

1.2 Organigramm

Im Rahmen des Projektes gibt es verschiedene Bereiche, an welche zeitgleich und effizient herangegangen werden muss. In der Phase der Produktentwicklung führen Missverständnisse in der Kommunikation zur massiven Stagnierung der Entwicklungskurve sowie zusätzlich zu massiven Unkosten. Damit die Team- und Kommunikationsfähigkeit gefördert werden kann, wird im Rahmen dieses Projektes eine Rollenverteilung erfolgen, sodass die Projektmitglieder an jeweils zwei Bereichen gleichzeitig tätig sein werden. Mit dieser Divisionsstruktur soll erreicht werden, dass die Kommunikation zwischen den Abteilungen aufrecht erhalten bleibt, um bei Anpassungen schneller agieren zu können und gleichzeitig eine Aufgabenvielfalt bereitstellen (Abbildung 1.3).

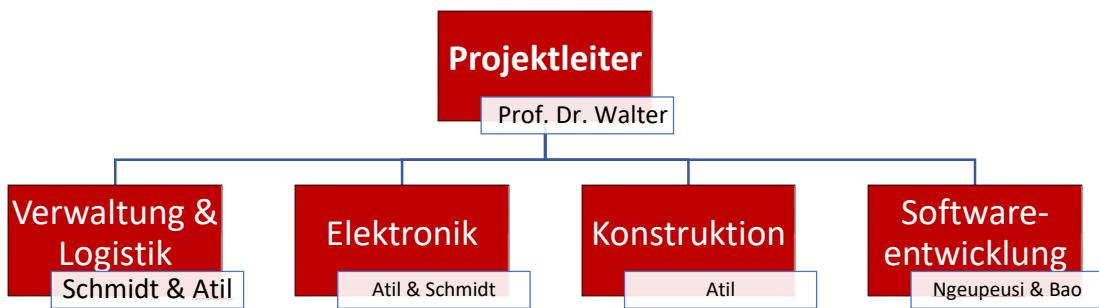


Abbildung 1.3: Organigramm der Projektmitglieder in Form einer Divisionsorganisation

2 Projektplanung

Das Projekt läuft vom 14.03.2025 bis zum 15.09.2025. Das entspricht eine Zusammenarbeit von ca. 6 Monaten. Für das Projekt wird den Studenten der Aufenthalt im Labor Mechatronische Komponente im Raum WB.110 gewährleistet, sowie die Nutzung der Messtechnischen Instrumente zur Verifikation der Funktionalität des zu entwickelnden Produktes: „OurDAQ“. Die Schließberechtigungen sind gültig vom 07.03.2025 bis zum 01.10.2025.

2.1 Kostenplanung und Logistik

Im Rahmen des Projektes wurde zu Beginn aus Seiten des Projektleiters ein Budget von 500€ als Startkapital festgelegt. In Abhängigkeit von der Projektentwicklung wurde ebenfalls vereinbart, dass das Budget von 500€ überschritten werden darf. I.d.R. entstehen selten Kosten bei der Softwareentwicklung, außer wenn Software-Lizenzen benötigt werden. Das Projekt-Team ist stets darum bemüht, auf Open-Source Produkte zurückzugreifen, sodass keine Kosten aus Seiten der Softwareentwicklung entstehen sollte.

Unvermeidlich ist die Kostenentstehung bei der Hardwareentwicklung aus Seiten der Elektronik und Konstruktion. Elektronische Bauteile, Schrauben und weitere mechanischen Werkstoffe können kostspielig sein. Die Kostenentwicklung steht unter ständiger Beobachtung und wird in Abhängigkeit der Projektentwicklung gesteuert, das bedeutet, dass das Projekt-Team das Controlling eigenständig steuert.

Die Bestellungen für die Hardware erfolgt ausschließlich durch ausgewählte Lieferanten wie Mouser oder Reichelt. In Ausnahme Fällen kann auch ein anderer Lieferant zum Einsatz kommen durch Absprache mit dem Projektleiter Prof. Dr. Walter.

2.2 Entstandene Kosten im Rahmen der Projektarbeit

Wie vermutet hat die Softwareentwicklung kein Budget für sich in Anspruch genommen. Die Konstruktion hat bis zum Schluss ebenfalls keine Kosten verursacht. Jedoch liegt das daran, dass ein Gehäuse nicht hergestellt worden ist. Somit sind alle Kosten nur durch die Elektronikabteilung entstanden. Von dem eingeplanten Budget wurden dabei nur 44,15% ausgegeben (vgl. Tabelle 2.1) Dabei entstanden knapp über 24% allein nur durch den Raspberry Pi 5 der neu bestellt werden musste, weil der ursprüngliche Raspberry Pi 3B die Anforderungen an das Messdatenerfassungssystem durch seine geringe Leitung nicht erfüllen konnte.

	Bauteil	Anzahl	Artikel-Nr.	Einzelpreis	Kosten	Lieferant
Raspberry Pi	Raspberry Pi 5 B, 4x2,4 GHz, 8 GB RAM, WLAN/ BT	1	RASPII 5 B8GB	91,20 €	91,20 €	Reichelt
	Raspberry Pi - Netzteil, 5,1 V, 5,0 A, USB Type-C, EU, schwarz	1	RPI PS 27WBKEU	12,40 €	12,40 €	Reichelt
	Raspberry Pi - Lüfter für Raspberry Pi 5	1	RASP ACTIVE COOL	6,25 €	6,25 €	Reichelt
	Raspberry Pi - Kabel, micro-HDMI (m) auf HDMI (f), 23cm, weiß	1	RPI M-HDMI HDMI5	3,90 €	3,90 €	Reichelt
	Raspberry Pi - Stacking-Header, 40-polig, RM2,54	1	RPI HEADER40XL	2,15 €	2,15 €	Reichelt
	Raspberry Pi - Stacking-Header, 40-polig, RM2,54	1	RPI HEADER40	1,60 €	1,60 €	Reichelt
	Raspberry Pi - Abstandsbolzen-Set, 30mm	1	RPI MOUNTINGKIT3	3,45 €	3,45 €	Reichelt
Teilsumme					120,95 €	
Spannungsversorgung	Bauteil	Anzahl	Artikel-Nr.	Einzelpreis	Kosten	Lieferant
	Shunt-Spannungsreferenz, fest, 5,0 V, ±0,5%, TO-92	2	LM4040 CIZ-5,0	0,99 €	1,98 €	Reichelt
	D/A-Wandler, 12-bit, 2-Kanal, SPI, 2,7 ... 5,5 V, DIP-14	1	MCP4922-E/P	3,75 €	3,75 €	Reichelt
	Widerstand, Metallschicht, 15 kOhm, 0207, 250 mW, 0,1%	2	ARC MRA0207 15K	0,52 €	1,04 €	Reichelt
	Widerstand, Metallschicht, 68 kOhm, 0207, 0,6 W, 0,1%	2	MPR68,0K	0,32 €	0,64 €	Reichelt
	Widerstand, Metallschicht, 100 Ohm, 0207, 0,6 W, 1%	10	METALL 100	0,07 €	0,70 €	Reichelt
	Eko radial, 330 uF, 35 V, 105°C, low ESR	3	FMA330U35B	0,35 €	1,05 €	Reichelt
	Keramik-Kondensator 10N	3	KERKO 10N	0,08 €	0,24 €	Reichelt
	Keramik-Kondensator, 100 nF, -20...+80 %, Y5V, 50/100 V, RM5	20	KERKO 100N	0,03 €	0,60 €	Reichelt
	Aufsteckkühlkörper für Gehäuse TO-220, 9KW	2	SK525 25 ST	4,90 €	9,80 €	Reichelt
	Spannungsreferenz, fest, 2,5 V, 1,5%, ± 80 ppm/°C, TO-92	2	LM285-ZZ,5	0,60 €	1,20 €	Reichelt
	D/A-Wandler, 12-bit, 1-Kanal, SPI, 2,7 ... 5,5 V, DIP-8	1	MCP4921-E/P	3,10 €	3,10 €	Reichelt
	Schottkydiode, 30 V, 1 A, DO-41	12	SB130	0,05 €	0,60 €	Reichelt
	Widerstand, Metallschicht, 1,0 kOhm, 0207, 0,6 W, 0,1%	14	MPR1,00K	0,28 €	3,92 €	Reichelt
	Eko radial, 100uF, 25V, 105°C, low ESR	3	FR-A100U25	0,14 €	0,42 €	Reichelt
	Widerstand, Metallschicht, 10 kOhm, 0207, 0,6 W, 0,1%	3	MPR10,0K	0,27 €	0,81 €	Reichelt
	Widerstand, Metallschicht, 1,3 kOhm, 0207, 0,6 W, 0,1%	1	MPR1,30K	0,18 €	0,18 €	Reichelt
	INA190A1IDCKR	2	595-INA190A1IDCKR	1,67 €	3,34 €	Mouser
	UT5-1RB3	1	652-UT5-1RB3	1,56 €	1,56 €	Mouser
	INA190A4QDCKRQ1	1	595-INA190A4QDCKRQ1	2,32 €	2,32 €	Mouser
	ADG419BNZ	1	584-ADG419BNZ	5,80 €	5,80 €	Mouser
Teilsumme					43,05 €	
Funktionsgenerator	Bauteil	Anzahl	Artikel-Nr.	Einzelpreis	Kosten	Lieferant
	Widerstand, Metallschicht, 10,0 kOhm, 0207, 0,6 W, 1%	10	METALL 10,0K	0,07 €	0,70 €	Reichelt
	Widerstand, Metallschicht, 1,0 kOhm, axial, 0,4 W, 1%	5	VISC001FC100	0,60 €	3,00 €	Reichelt
	Widerstand, Metallschicht, 5,6 kOhm, axial, 0,4 W, 1%	5	VISC5601FC100	0,39 €	1,95 €	Reichelt
	Widerstand, Metallschicht, 41,2 kOhm, 0207, 0,6 W, 1%	5	METALL 41,2K	0,07 €	0,35 €	Reichelt
	Widerstand, Metallschicht, 4,99 kOhm, 0207, 0,6 W, 1%	10	METALL 4,99K	0,07 €	0,70 €	Reichelt
	Trimmopotentiometer, 1 kOhm, linear	5	PIH PTC10LV1102A	0,79 €	3,95 €	Reichelt
	Präzisionspotentiometer, 25 Gänge, stehend, 10 kOhm	2	64Y-10K	1,80 €	3,60 €	Reichelt
	LT1352CN8#PBF	2	584-LT1352CN8#PBF	10,46 €	20,92 €	mouser
	AD8479ARZ-RL	1	584-AD8479ARZ-RL	9,12 €	9,12 €	mouser
	ADG419BNZ	1	584-ADG419BNZ	5,80 €	5,80 €	mouser
Teilsumme					56,73 €	
Bestellungen im Projekt SS 2025			Budget	Gesamtausgaben		
			500 €	220,73 €		

Tabelle 2.1: Bestellungsliste im Projekt

2.3 Systemanforderungen

Dem Projekt-Team wurden keine expliziten Anforderungslisten bzw. ein Lastenheft gegeben.

Aus diesem Grund wurde weitestgehend eine Systemanforderung zum Produkt MyDAQ von NI entwickelt, sodass ein Pflichtenheft entwickelt werden kann. Die Anforderungsliste setzt sich zusammen aus den Anforderungen von NI, Digilent und den eigenen Anforderungen des Projekt-Teams (Tabelle 2.2).

Nr.	Anforderung	Abteilung			Zahlenwert	Anforderungsart			Name	Datum
		Elektronik	Konstruktion	Software		Fest	Toleranz	Wunsch		
1	Analoge Eingänge (AI)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	≥2 Kanäle, diff./SE umstellbar	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NI	27.03.2025
2	AI Auflösung	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	≥16 Bit (effektiv ≥14 Bit)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NI	27.03.2025
3	AI Abtastrate	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	≥100 kS/s pro Board (Ziel 200 kS/s)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Digilent	27.03.2025
4	AI Eingangsbereiche	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	±10 V (weitere: ±5/±2/±1 V wählbar)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NI	27.03.2025
5	Analoge Ausgänge (AO)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	≥2 Kanäle	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NI	27.03.2025
6	AO Auflösung/Rate	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	≥12 Bit, ≥5 kS/s (Ziel 16 Bit, ≥20 kS/s)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Digilent	27.03.2025
7	Digital I/O	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	≥8 Leitungen, 3,3/5 V tolerant	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NI	27.03.2025
8	Counter/Timer / Trigger	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1 Zähler (≥32 Bit), ext. Trigger/Scan-Clock	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Digilent	27.03.2025
9	Integriertes DMM-Äquivalent	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	DC/AC V, R, Durchgang, Diotentest	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Projekt-Team	27.03.2025
10	Versorgungsschienen	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	±5 V, ±15 V Hilfsspannungen (≥100 mW)	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	NI	27.03.2025
11	Genauigkeit AI	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	≤0,1% FS @ 23°C (Ziel)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NI	27.03.2025
12	Rauschen / ENOB	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ENOB ≥ 13 Bit (AI)	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Projekt-Team	27.03.2025
13	Stew Rate AO	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	≥0,8 V/μs	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Projekt-Team	27.03.2025
14	Überspannungsschutz AI	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	±20 V min	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NI	27.03.2025
15	Kurzschluss-/Überlastschutz AO	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	gegen GND=V, thermisch	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NI	27.03.2025
16	ESD/EMV	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	IEC 61000-4-x orientiert	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Projekt-Team	27.03.2025
17	Galvanische Trennung	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	optional AI/AO/USB isoliert	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Projekt-Team	27.03.2025
18	Anschlussklemmen	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Schraub-/Federklemmen, Farbcodierung	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NI	27.03.2025
19	Buchsen für Audio I/O	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	3,5 mm (Stereo In/Out)	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Projekt-Team	27.03.2025
20	Gehäuse	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Kompakt, lüfterlos, beschriftet	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NI	27.03.2025
21	Statusanzeigen	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	LEDs: Power, Trigger, IO-Status	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Projekt-Team	27.03.2025
22	Raspberry Pi 5 Integration	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	HAT-Stack bis 3 Boards	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Projekt-Team	27.03.2025
23	Externe Trigger/Clock	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	BNC/SMB/Schraubklemme	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Projekt-Team	27.03.2025
24	Netzteil	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	USB-C 5V/5A stabilisiert	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NI	27.03.2025
25	Masseleitung/Abschirmung	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Sternpunkt/Analog-Digital-Trennung	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NI	27.03.2025
26	Python API	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	mcc-daq-hat & RPI.GPIO	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Projekt-Team	27.03.2025
27	Jupyter Notebooks	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Mess-/Steuer-Beispiele	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Projekt-Team	27.03.2025
28	Treiber/Install	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Einzelner-Setup-Skript	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Projekt-Team	27.03.2025
29	Datenlogging	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	CSV/Parquet, Zeitstempel, Metadaten	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Projekt-Team	27.03.2025
30	Live-Plotting	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Matplotlib/Plotly	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Projekt-Team	27.03.2025
31	Kalibrierung	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Werks-/Benutzerkalibrierung, Offset/Gain	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Projekt-Team	27.03.2025
32	Selbsttest/Loopback	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	AO→AI Test, DIO Loopback	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Projekt-Team	27.03.2025
33	Sicherheitsfunktionen	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Spannungs-/Stromlimits, Watchdog	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Projekt-Team	27.03.2025
34	Dokumentation	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Readme, Pinout, Beispiele	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Projekt-Team	27.03.2025
35	Akzeptanzkriterien	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Messfehler ≤0,1% FS @ DC	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Projekt-Team	27.03.2025
36		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
37		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
38		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
39		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
40		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
41		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
42		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
43		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
44		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
45		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
46		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
47		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
48		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
49		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
50		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

Tabelle 2.2: Anforderungsliste des Messdatenerfassungssystems

2.4 Zeitplanung im Projekt

Das Projekt-Team hat weitestgehend die Zeitplanung eingehalten. Auf Grund der Master-Prüfungen wurde während der Prüfungsphase das Projekt das erste Mal eingefroren. Am 12.08.2025 wurde das Projekt vom Projektleiter selbst endgültig eingefroren, sodass das Projekt als beendet gilt. Aus diesem Grund wurden Projektvorhaben ab der 33 KW nicht mehr praktisch, sondern theoretisch behandelt (siehe Abbildung 2.1).

Zeitplanung																													
Monat		März				April				Mai				Juni				Juli			August			September					
Kalenderwoche KW		12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	ABGABE	
Planung																													
Team	1 Personal-Zeit-Planung	X																											
	2 Formularentwurf	X																											
	3 Analyse Stand d. Technik	X	X																										
	4 Anforderungsliste	X	X																										
	Konzipierung																												
	1 Funktionsgliederung		X	X																									
	2 Lösungsprinzipien				X	X	X	X																					
	3 Konzepte (mind. 3)					X	X	X																					
	Entwurf & Ausarbeitung																												
	1 Dimensionierung der Elektronik								X	X	X																		
Elektronik	2 Bauelementenübersicht								X	X	X	X																	
	3 Simulation LTSpice								X	X	X	X																	
	4 Schaltplanerstellung								X	X	X	X	X																
	5 Hardware Verknüpfung								X	X	X	X																	
	1 Konzeptionierung												X	X	X	X													
	2 Bauelementenübersicht											X	X	X	X														
	3 Bestellung																												
	4 3D-Modelle: Gestaltung																		X	X	X								
Konstruktion	5 Einzelteilzeichnung																		X	X									
	6 Baugruppenzeichnung																		X	X									
	7 3D-Druck																				X								
	8 Elektronik integration																				X								
	Systemdesignphase																												
	1 Systemarchitektur		X	X	X																								
	2 Funktionspezifikation		X	X	X																								
	Implementierung																												
Software-entwicklung	1 Simulationsumgebung			X	X	X																							
	2 Funktionsrealisierung			X	X	X	X	X	X	X	X																		
	Validierung																												
	1 Einzelfunktionstest								X	X	X	X																	
Team	2 Funktionsintegrationstest											X	X																
	3 Softwaretest																		X	X	X	X							
	4 Abnahmetest																		X	X	X								
	Dokumentation																												
Team	1 Doku mit Quellenachweise	X	X	X	X	X	X	X	X	X	X	X	X	X	X														
	2 Anlagen					X	X	X																					
	3 Präsentation																					X	X						
	4 Dateien (CAD, Code, Schaltplan)																		X	X	X	X	X	X					
Jour-Fix		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		
Meeting mit dem Prof.		X		X		X		X		X		X		X		X		X		X		X		X		X		X	
Prüfungsphase																													
Monat		März				April				Mai				Juni				Juli			August			September					
Kalenderwoche KW		12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	ABGABE	

Abbildung 2.1: Gant - Diagramm des Projektes

2.5 Stand der Technik

Für die Projektrealisierung ist die Recherche aus dem Stand der Technik in der Planung ausschlaggebend für den Erfolg. Nach dem Pareto-Prinzip kann bereits mit 20% Zeitaufwand von einer Erfolgsquote aus 80% ausgegangen werden [3].

Die Auseinandersetzung mit dem Stand der Technik ermöglicht die Feststellung der Funktionsmöglichkeiten sowie die Auseinandersetzung der Realisierungsmöglichkeiten trotz den limitierenden Anforderungen an das System.

2.5.1 Die Messvorrichtung MyDAQ von National Instruments [4] [5]

Das **NI myDAQ** stellt ein vielseitiges und kompaktes Datenerfassungsgerät dar, das speziell für den Einsatz in Lehre und Ausbildung entwickelt wurde. Es vereint wesentliche messtechnische Funktionen in einem portablen Format und ermöglicht Studierenden wie Lehrenden den direkten Zugang zu realen Messungen sowie deren Analyse in Verbindung mit der Softwareumgebung **LabVIEW**. Aufgrund seines günstigen Preises und seiner breiten Funktionalität bietet es eine praxisnahe Plattform, um grundlegende Kenntnisse der Mess- und Elektrotechnik zu vermitteln und gleichzeitig experimentelle Anwendungen zu realisieren.



Abbildung 2.2: MyDAQ von National Instruments

2.5.1.1 Aufbau und Funktionsweise

Das Gerät verfügt über mehrere Schnittstellen und Funktionsblöcke, die eine große Bandbreite an Anwendungen abdecken. Dazu gehören **analoge Ein- und Ausgänge**, **digitale I/O-Anschlüsse**, ein integriertes **Digitalmultimeter**, verschiedene **Stromversorgungsausgänge** sowie **Audiofunktionen**. Die analogen Eingänge erlauben Spannungsmessungen im Bereich von ± 10 V oder – im Audiomodus – von ± 2 V und erreichen eine maximale Abtastrate von 200 kS/s. Diese Eingänge werden beispielsweise beim Einsatz von Oszilloskop-, Bode- oder

Signal-Analyser-Funktionen genutzt. Ergänzt wird dies durch zwei Analogausgänge, die ebenfalls eine hohe Flexibilität in der Signalbereitstellung bieten und simultan mit bis zu 200 kS/s betrieben werden können.

Neben der analogen Signalverarbeitung stellt das Gerät **acht digitale I/O-Kanäle** bereit, die mit 3,3 V LVTTL arbeiten und 5-V-tolerant sind. Fünf dieser Kanäle können zusätzlich als Zähler- und Timer-Eingänge fungieren, wodurch sich komplexere digitale Anwendungen umsetzen lassen. Für die Versorgung externer Schaltungen stehen drei Spannungsquellen (+15 V, -15 V und +5 V) bereit, deren Gesamtleistung jedoch auf 500 mW begrenzt ist. Das integrierte **Digitalmultimeter** erlaubt zudem die Messung grundlegender elektrischer Größen wie Spannungen, Ströme, Widerstände und Diodenkennwerte mit einer Auflösung von 3,5 Stellen.

2.5.1.2 Software-Ökosystem

Die Leistungsfähigkeit des NI myDAQ entfaltet sich insbesondere in Kombination mit der **National Instruments Softwareumgebung**. Über den **NI-ELVISmx-Treiber** wird eine Vielzahl von virtuellen Instrumenten bereitgestellt, die in **LabVIEW** als Soft-Frontpanels zur Verfügung stehen. Hierzu zählen unter anderem ein Oszilloskop, ein Funktions- und Arbiträr-Generator, ein Bode- und Frequenzanalysator sowie digitale Ein- und Ausgabemodule. Damit lassen sich sowohl klassische Messaufgaben als auch fortgeschrittene Analysen intuitiv umsetzen. Darüber hinaus ermöglicht die Integration mit **NI Multisim** den direkten Vergleich zwischen simulierten und realen Messdaten, was den didaktischen Wert des Geräts zusätzlich steigert.

Das NI myDAQ kann sowohl über LabVIEW-Express-VIs als auch über **NI-DAQmx** programmiert werden. Dies erlaubt es, individuell zugeschnittene Mess- und Steuerungsanwendungen zu entwickeln, die über den Bildungsbereich hinaus auch in prototypischen technischen Projekten eingesetzt werden können.

2.5.1.3 Handhabung und Sicherheit

Für den Einsatz im Lehrbetrieb ist die einfache Inbetriebnahme ein entscheidender Vorteil. Nach der Installation der erforderlichen Software und Treiber wird das Gerät über USB mit dem Rechner verbunden und automatisch erkannt. Dennoch sind bei der Nutzung sicherheitsrelevante Aspekte zu berücksichtigen: Die Anschlüsse sind empfindlich gegenüber elektrostatischer Entladung, weshalb entsprechende Schutzmaßnahmen einzuhalten sind. Ferner sind die zulässigen Spannungsgrenzen strikt zu beachten, da das Gerät nur für

Messungen im Bereich der **Messkategorie I** ausgelegt ist und somit nicht an Netzspannungen betrieben werden darf.

2.5.1.4 Physikalische Eigenschaften

Das NI myDAQ weist kompakte Abmessungen und ein geringes Gewicht auf, wodurch es sich ideal für mobile Anwendungen eignet. Es ist ausschließlich für den Einsatz in Innenräumen konzipiert und für Betriebstemperaturen zwischen 0 und 45 °C spezifiziert. Auch hinsichtlich elektromagnetischer Verträglichkeit erfüllt es die einschlägigen europäischen Richtlinien und ist entsprechend **CE-zertifiziert**.

Zusammenfassend lässt sich feststellen, dass das NI myDAQ ein didaktisch wertvolles und technisch vielseitiges Gerät ist, das in der Ausbildung der Elektrotechnik und Signalverarbeitung eine zentrale Rolle einnehmen kann. Es ermöglicht den praxisnahen Erwerb messtechnischer Kompetenzen, indem es reale Signale erfassbar, analysierbar und reproduzierbar macht. Durch seine enge Verzahnung mit Softwarelösungen wie LabVIEW und Multisim verbindet es theoretische Lerninhalte mit praktischen Anwendungen und leistet damit einen wichtigen Beitrag zur ingenieurwissenschaftlichen Ausbildung.

2.5.2 Raspberry Pi 5 [6] [7]

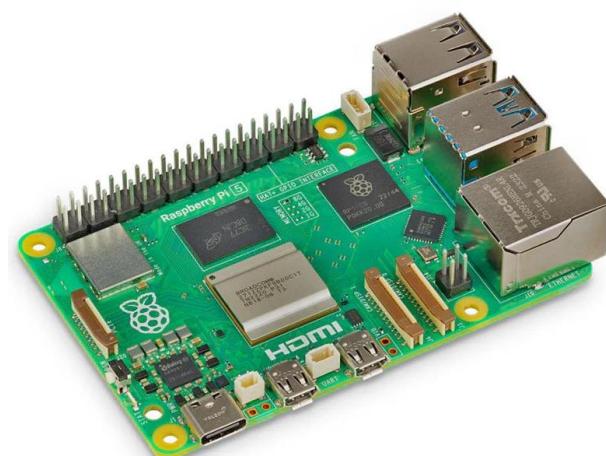


Abbildung 2.3: Raspberry Pi 5

Der **Raspberry Pi 5** repräsentiert die jüngste Entwicklungsstufe der Einplatinencomputer-Reihe der Raspberry Pi Foundation und markiert einen bedeutenden Fortschritt in der Mini-Computer-Technologie. Seit seiner Veröffentlichung im September 2023 wird er als „Alles-Computer“ bezeichnet, da er sowohl für den privaten als auch für den industriellen Einsatz vielseitige

Anwendungsmöglichkeiten bietet. Die Entwicklung des Raspberry Pi ist eng mit dem Ziel verbunden, den Rückgang der Informatikstudentenzahlen durch ein kostengünstiges, leicht zugängliches und programmierbares Gerät zu kompensieren. Während die Stiftung die Bildungsförderung verantwortet, erfolgt die Produktion und Vermarktung über die

Tochtergesellschaft Raspberry Pi Trading. Der Raspberry Pi 5 ist für eine lange Produktlebensdauer ausgelegt und soll mindestens bis 2035 verfügbar bleiben.

2.5.2.1 Architektur und technische Eigenschaften

Mit dem **Arm Cortex-A76 Quad-Core-Prozessor** (2,4 GHz, 64-bit) erreicht der Raspberry Pi 5 eine **2–3-fache CPU-Leistungssteigerung** gegenüber seinem Vorgänger, dem Raspberry Pi 4. Ergänzt wird diese durch eine **VideoCore VII GPU**, die moderne Grafikstandards wie OpenGL ES 3.1 und Vulkan 1.2 unterstützt. Die Arbeitsspeichervarianten reichen von **2 GB bis 8 GB LPDDR4X-4267 SDRAM**, wobei auch eine Version mit 16 GB diskutiert wird.

Die Schnittstellenausstattung unterstreicht die Vielseitigkeit des Systems: Zwei **microHDMI-Anschlüsse** ermöglichen Dual-4K-Ausgabe bei 60 Hz mit HDR, während ein integrierter 4Kp60-HEVC-Decoder für effiziente Videowiedergabe sorgt. Zusätzlich verfügt der Raspberry Pi 5 über **Dualband-WLAN (802.11ac)**, **Bluetooth 5.0/BLE**, zwei USB-3.0- sowie zwei USB-2.0-Anschlüsse und **Gigabit-Ethernet** mit optionaler **PoE+-Unterstützung**. Auch der **40-Pin-GPIO-Header** bleibt erhalten, was die Kompatibilität mit bestehenden Projekten sicherstellt.

Eine wesentliche Neuerung stellt die **PCIe-2.0-x1-Schnittstelle** dar, die erstmals eine Anbindung schneller Peripheriegeräte erlaubt. Darüber hinaus wurden die Kamera- und Display-Schnittstellen durch **zwei 4-Lane-MIPI-Transceiver** modernisiert, wodurch sich Bandbreite und Flexibilität erheblich erhöhen. Als Stromversorgung dient ein **USB-C-Anschluss mit 5V/5A und Power-Delivery-Unterstützung**, womit das Gerät höhere Leistungsreserven bietet. Ergänzend wurden eine **integrierte Echtzeituhr (RTC)**, ein **Power-Button** sowie ein Lüfteranschluss eingeführt – Funktionen, die von der Community lange gefordert wurden.

2.5.2.2 Anwendungsfelder

Die gesteigerte Leistung und erweiterte Ausstattung eröffnen ein breites Anwendungsspektrum. Im **Bildungsbereich** kann der Raspberry Pi 5 als leistungsfähige Plattform für Programmier- und Hardwareprojekte eingesetzt werden. Seine Rechenleistung erlaubt zudem den Einsatz als **Desktop-Computer** oder **Media Center**, etwa für

hochauflösendes Streaming. In der **Maker-Community** wird er häufig für die Entwicklung von Robotern, DIY-Spielkonsolen oder Sensornetzwerken genutzt.

Auch im industriellen Umfeld gewinnt der Raspberry Pi 5 an Bedeutung: Mit Gigabit-Ethernet, PCIe-Erweiterungen und stabiler Stromversorgung eignet er sich für Automatisierungsaufgaben, Steuerungssysteme oder Edge-Computing-Anwendungen. Damit schließt er die Lücke zwischen klassischem Lernwerkzeug und professioneller Embedded-Lösung.

2.5.2.3 Nutzung, Zubehör und Sicherheit

Die Inbetriebnahme erfolgt unkompliziert über das **Raspberry Pi OS**, wobei auch andere Linux-Distributionen und Windows 10 ARM64 unterstützt werden. Für den Betrieb sind neben einer **microSD-Karte** eine stabile Stromversorgung (5V/5A) sowie Monitor, Tastatur und Maus erforderlich. Zusatzkomponenten wie **aktive Kühler**, **Schutzgehäuse**, eine **RTC-Batterie** oder spezielle Kamera- und Displaykabel erweitern die Funktionalität und gewährleisten eine langfristige Zuverlässigkeit.

Hinsichtlich der **Sicherheit** ist auf eine gute Belüftung sowie den Schutz vor Feuchtigkeit, extremen Temperaturen und elektrostatischer Entladung zu achten. Insbesondere sollte das Gerät während des Betriebs nur an den Kanten berührt werden. Beschädigte oder überhitzte Platinen dürfen nicht weiterverwendet werden.

2.5.2.4 Zusammenfassung

Der Raspberry Pi 5 stellt einen bedeutenden Schritt in der Evolution der Raspberry-Pi-Serie dar. Mit seiner **deutlich gesteigerten Rechen- und Grafikleistung**, modernen Schnittstellen und erweiterten Sicherheits- sowie Komfortfunktionen verbindet er die kompakte Bauweise eines Einplatinencomputer mit den Ansprüchen moderner Anwendungsszenarien. Damit etabliert er sich als vielseitige Plattform, die gleichermaßen in **Bildung**, **Forschung**, **Hobbyprojekten** und **industriellen Anwendungen** zum Einsatz kommt und durch seine geplante Langzeitverfügbarkeit eine hohe Investitionssicherheit bietet.

2.5.3 Digilent MCC 118 DAQ HAT [8]

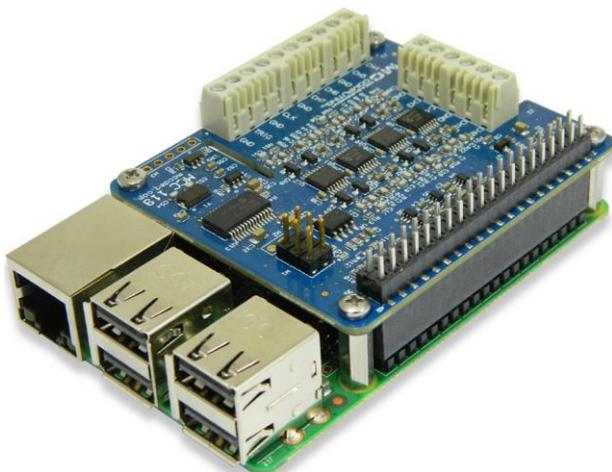


Abbildung 2.4: Digilent MCC 118 DAQ HAT

Der **MCC 118** stellt ein speziell für den Einsatz mit dem Raspberry Pi entwickeltes **DAQ HAT (Hardware Attached on Top)** dar, das als Erweiterungsmodul über den 40-poligen GPIO-Anschluss direkt aufgesteckt wird. Es ist mit allen Raspberry-Pi-Modellen kompatibel, die über diesen Anschluss verfügen, und erweitert deren Funktionalität erheblich im Bereich der **präzisen Spannungsmessung**. Mit seinen

acht 12-Bit-Single-Ended-Eingängen und einem festen Messbereich von **±10 V** eignet sich das Modul für eine Vielzahl an Anwendungsfeldern, insbesondere für die experimentelle Datenerfassung und die Sensordatenüberwachung. Die Möglichkeit, präzise elektrische Spannungen zu messen und digital zu verarbeiten, macht den MCC 118 zu einem vielseitigen Werkzeug für Lehre, Forschung und praxisorientierte Entwicklungsprojekte.

2.5.3.1 Zentrale Merkmale und technische Fähigkeiten

Das Gerät bietet eine Reihe von Eigenschaften, die es zu einem leistungsstarken Messinstrument machen. Dazu zählen acht Analogeingänge mit einer Eingangsimpedanz von **1 MΩ** sowie einer absoluten maximalen Eingangsspannung von **±25 V**. Die Abtastrate liegt bei **100 kS/s pro Platine**, wobei bei gestapeltem Betrieb von bis zu acht HATs eine aggregierte Rate von **320 kS/s** erreicht werden kann. Ein integrierter **FIFO-Speicher von 7.168 Mustern** ermöglicht die Pufferung von Hochgeschwindigkeitserfassungen.

Darüber hinaus erlaubt die Stapelbarkeit die **Kombination mehrerer MCC HATs**, einschließlich unterschiedlicher Modelle, wodurch komplexere Mess- und Steuerungssysteme aufgebaut werden können. Über externe Takt- und Trigger-Eingänge lassen sich mehrere Module synchronisieren, was präzise und flexible Messkonfigurationen ermöglicht. Ergänzend erlaubt der konfigurierbare **externe Scan-Takt** mit bis zu 100 kHz eine flexible Steuerung, während der digitale Trigger verschiedene Betriebsarten (Flanken- oder Pegelerkennung) unterstützt.

Die Stromversorgung erfolgt über die 3,3-V-Leitung des Raspberry Pi, wobei der Stromverbrauch mit durchschnittlich 35 mA sehr gering bleibt. Für den praktischen Einsatz sind Standardmodelle mit Schraubklemmen verfügbar, während die OEM-Version alternativ unbestückte Header-Anschlüsse bietet. Dank kompakter Abmessungen von $65 \times 56,5 \times 12$ mm lässt sich der MCC 118 auch in platzkritischen Projekten einsetzen. Ein Onboard-EEPROM unterstützt die automatische Konfiguration der GPIO-Pins beim Anschluss.

2.5.3.2 Softwareunterstützung und Programmierung

Die Softwareintegration erfolgt über die **MCC DAQ HAT Library**, eine frei verfügbare Open-Source-Bibliothek für **C/C++ und Python**, die den Betrieb unter Linux/Raspbian ermöglicht. Diese unterstützt den **gleichzeitigen Betrieb mehrerer HATs**, wodurch sich verteilte Datenerfassungen leicht implementieren lassen. Neben einer umfangreichen API-Dokumentation stehen zahlreiche Beispielprogramme zur Verfügung. Konsolenbasierte Beispiele demonstrieren kontinuierliche und endliche Scans sowie synchrone Erfassungen, während Programme mit grafischer Oberfläche wie der **DataLogger** Daten in Echtzeit visualisieren und aufzeichnen. Weiterhin existieren Webserver-Implementierungen zur browserbasierten Anzeige sowie Python-Skripte für die Integration in Cloud-Dienste wie Google Sheets über IFTTT, was insbesondere für Remote-Monitoring-Anwendungen interessant ist.

2.5.3.3 Technische Spezifikationen

Die Signalwandlung basiert auf einem **A/D-Wandler nach dem Prinzip der sukzessiven Approximation** mit 12-Bit-Auflösung. Die Genauigkeit liegt im Messbereich von ± 10 V bei einem maximalen Gain-Fehler von 0,098 % des Messwertes und einem Offset-Fehler von 11 mV. Temperaturkoeffizienten von 0,016 %/ $^{\circ}\text{C}$ (Gain) und 0,87 mV/ $^{\circ}\text{C}$ (Offset) unterstreichen die Stabilität unter variierenden Umgebungsbedingungen. Das Rauschniveau beträgt bei angeschlossenem AGND 0,76 LSBrms. Die Kommunikation mit dem Raspberry Pi erfolgt über die **SPI-Schnittstelle** mit bis zu 10 MHz. Neben dem FIFO-Speicher für die Eingangsdaten verfügt das Modul über 4 KB nichtflüchtigen Speicher für Identifikations- und Kalibrierungsdaten. Der zulässige Betriebsbereich erstreckt sich von 0 °C bis 55 °C, bei einer nicht kondensierenden relativen Luftfeuchtigkeit von bis zu 90 %.

2.5.3.4 Gesamtbewertung

Der MCC 118 kombiniert eine **hohe Messgenauigkeit, skalierbare Abtastraten und flexible Softwareunterstützung** mit einer kompakten Bauform. Seine Fähigkeit, mehrere Module zu stapeln und synchronisiert zu betreiben, macht ihn besonders interessant für Anwendungen, die eine große Kanalanzahl oder komplexe Messkonfigurationen erfordern. Dank offener Softwarebibliotheken und Beispielanwendungen lässt er sich schnell in Projekte integrieren und ermöglicht eine enge Verzahnung von Hardware- und Softwareentwicklung.

Insgesamt stellt der MCC 118 eine **leistungsfähige und zugleich kosteneffiziente Lösung** für die präzise Spannungsmessung mit Raspberry-Pi-Systemen dar und eignet sich sowohl für den didaktischen Einsatz als auch für industrielle und experimentelle Anwendungen im Bereich der Datenerfassung und Signalverarbeitung.

3 Projektentwicklung

3.1 Technische Grundlagen

In dem Kapitel Technische Grundlagen wird das für dieses Projekt notwendige Grundverständnis zu den relevanten Schaltungen, Messprinzipien und Kommunikationsprotokollen vermittelt.

Ziel ist es, eine einheitliche Wissensbasis zu schaffen, auf der die folgenden Abschnitte zur Hardware- und Softwareentwicklung aufbauen können. Neben grundlegenden elektronischen Schaltungen wie Operationsverstärker-Konfigurationen oder Strommessverfahren werden auch Schnittstellen und Protokolle wie GPIO und SPI sowie die eingesetzten Messmodule vorgestellt.

3.1.1 Strommessung mit Shunt

Ein Shunt ist ein niederohmiger Widerstand, der speziell zur Strommessung eingesetzt wird und deshalb auch als Strommess-Widerstand bezeichnet wird. Über den bekannten Widerstandswert lässt sich mithilfe des Ohm'schen Gesetzes die Stromstärke berechnen:

$$I = \frac{U}{R}$$

Um die Verlustleistung und damit auch die Wärmeentwicklung möglichst gering zu halten, werden Shunts in der Regel mit sehr kleinen Widerstandswerten ($1\text{ m}\Omega$ bis $10\text{ }\Omega$) ausgelegt.

Bei der Strommessung in Kombination mit einem Operationsverstärker (OPV) wird der Spannungsabfall über dem Shunt erfasst und anschließend verstärkt. Dadurch können selbst sehr kleine Spannungen, wie sie bei niederohmigen Shunts entstehen, präzise gemessen und weiterverarbeitet werden (vgl. Abbildung 3.1). [9, 10]

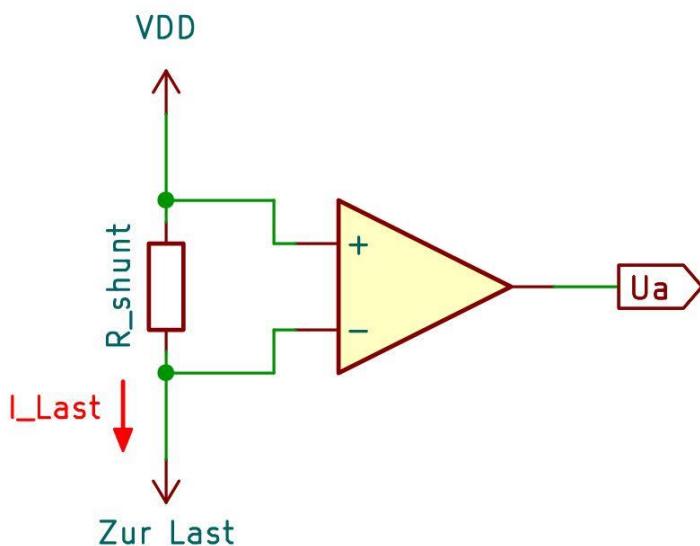


Abbildung 3.1: Strommessung - Shunt + OPV [9, 10]

3.1.2 OPV - Grundschatungen

3.1.2.1 Nichtinvertierender Verstärker

Beim nichtinvertierenden Verstärker wird die Eingangsspannung U_e direkt am Plus-Eingang des Operationsverstärkers angelegt. Die Gegenkopplung erfolgt über die beiden Widerstände R_2 und R_1 (vgl. Abbildung 3.2). Durch diese Beschaltung bleibt die Eingangsspannung phasengleich zur Ausgangsspannung, d.h. es findet keine Invertierung statt. [11]

Die Verstärkung V eines nichtinvertierenden Verstärkers berechnet sich nach folgender Formel:

$$V = 1 + \frac{R_2}{R_1}$$

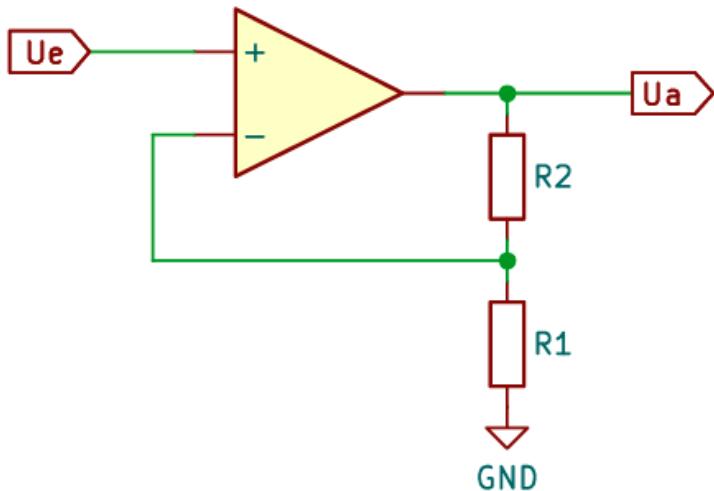


Abbildung 3.2: Grundschatzung - Nichtinvertierender Verstärker [11]

3.1.2.2 Impedanzwandler

Der Impedanzwandler (auch Spannungsfolger genannt), ist eine der einfachsten Operationsverstärker-Schaltungen. Dabei wird der Ausgang direkt mit dem Minus-Eingang des OPV verbunden (vollständige Gegenkopplung, siehe Abbildung 3.3). [11]

Eigenschaften:

- Verstärkung: $V = 1$
- Eingangswiderstand: sehr hoch
- Ausgangswiderstand: sehr niedrig
- Keine Phasendrehung (phasengleich)

Der Impedanzwandler dient zur Pufferung empfindlicher Signalquellen. Er verhindert, dass der Signalgeber durch den Eingangsstrom einer nachfolgenden Schaltung belastet wird.

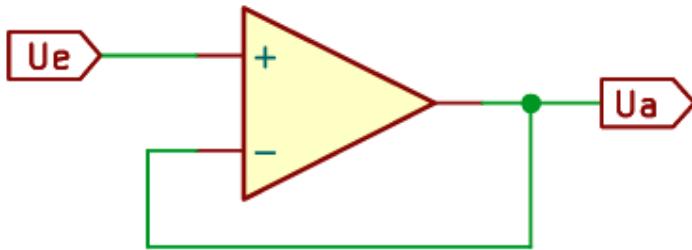


Abbildung 3.3: Grundschaltung - Impedanzwandler [11]

3.1.3 GPIOs - Raspberry Pi

Die Abkürzung GPIO steht für General Purpose Input/Output und bezeichnet die universell nutzbaren Pins auf dem Raspberry-Pi-Board. Mit ihnen lassen sich digitale Signale sowohl einlesen (Input) als auch ausgeben (Output).

Im Eingabemodus können die Pins beispielsweise erfassen, ob ein Signal anliegt, wie bei einem Taster oder Sensor. Im Ausgabemodus lassen sich damit Signale schalten, etwa zum Ansteuern von LEDs oder Relais.

Jeder GPIO-Pin kann per Software individuell konfiguriert werden und ausgewählte Pins können bei Bedarf auch Spezialfunktionen wie I²C, SPI, UART oder PWM realisieren. [12]

3.1.4 SPI-Schnittstelle

Die SPI-Schnittstelle (Serial Peripheral Interface) ist ein serielles Kommunikationsprotokoll, das vor allem für die schnelle Datenübertragung zwischen einem Master (z. B. Raspberry Pi) und einem oder mehreren Slaves (z. B. Sensoren, DACs, ADCs) verwendet wird. [13]

Die Kommunikation erfolgt über vier Hauptleitungen (vgl. Abbildung 3.4):

- SCLK (Serial Clock) – Takteleitung vom Master
- MOSI (Master Out, Slave In) – Daten vom Master zum Slave
- MISO (Master In, Slave Out) – Daten vom Slave zum Master
- SS (Slave Select) – Auswahl der aktiven Slave-Komponente

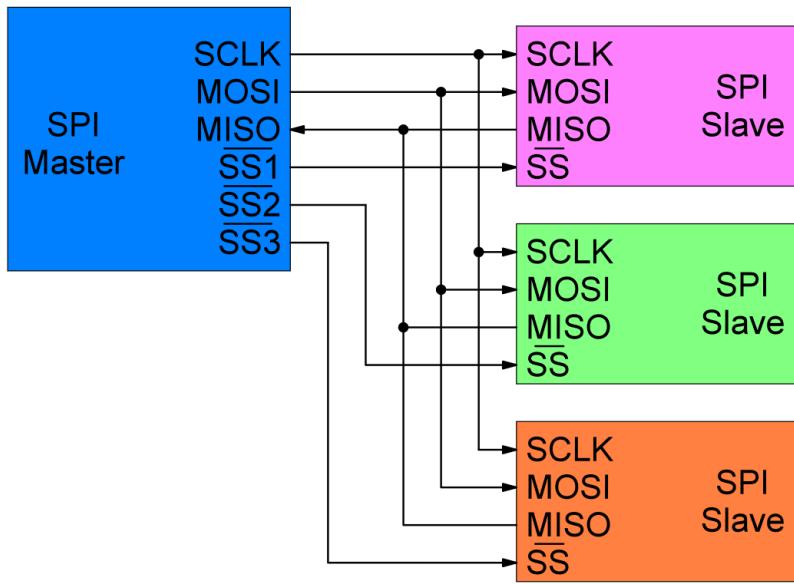


Abbildung 3.4: SPI-Kommunikationsverfahren [13]

3.2 Konzeptionierung

3.2.1 Funktionsgliederung eines Labornetzteils

Ein Labornetzteil ist ein spezielles Netzgerät, das in Laboren, Werkstätten und bei der Elektronikentwicklung eingesetzt wird. Im Gegensatz zu einfachen Steckernetzteilen oder Ladegeräten bietet es die Möglichkeit, Spannung und Strom präzise einzustellen und zu überwachen. Dadurch eignet es sich besonders für Experimente, Messungen, Reparaturen oder Entwicklungsaufgaben, bei denen verschiedene Spannungen benötigt werden oder empfindliche Schaltungen getestet werden (vgl. Abbildung 3.5).

Grundfunktionen eines Labornetzteils:

1. Spannungsversorgung

- Stellt eine einstellbare Gleichspannung (DC) bereit, oft in einem weiten Bereich (z. B. 0–30 V).
- Ermöglicht die präzise Vorgabe über Drehregler, Tasten oder digital.
- Häufig sind auch mehrere unabhängige Ausgänge vorhanden.

2. Strombegrenzung (Konstantstrombetrieb, CC)

- a. Schutzfunktion für Schaltungen: Wenn der eingestellte Maximalstrom erreicht wird, regelt das Netzteil automatisch auf Konstantstrom und verhindert Schäden durch Überlastung.
- b. Nützlich zum Testen von Bauteilen oder für das Laden von Akkus.

3. Konstantspannungsbetrieb (CV)

- a. Das Netzteil hält die eingestellte Spannung unabhängig von der Belastung stabil, solange der Maximalstrom nicht überschritten wird.

4. Anzeige- und Kontrollfunktionen

- a. Digitale Anzeigen für Spannung, Strom und oft auch Leistung.
- b. Teilweise mit Speicherplätzen für häufig verwendete Einstellungen.
- c. Moderne Geräte haben Schnittstellen (USB, LAN, RS232) zur PC-Steuerung.

5. Schutzfunktionen

- a. Überstromschutz (OCP)
- b. Überspannungsschutz (OVP)
- c. Übertemperaturschutz (OTP)
- d. Kurzschlussfestigkeit

6. Zusatzfunktionen

- a. Mehrkanalbetrieb (z. B. zwei oder drei Ausgänge, auch in Serie oder parallel schaltbar).
- b. Tracking-Funktion: mehrere Ausgänge lassen sich gekoppelt einstellen.
- c. Programmierbarkeit und Fernsteuerung (LabVIEW, SCPI-Befehle).
- d. Geräusch- und Restwelligkeitsminimierung für empfindliche Elektronik.

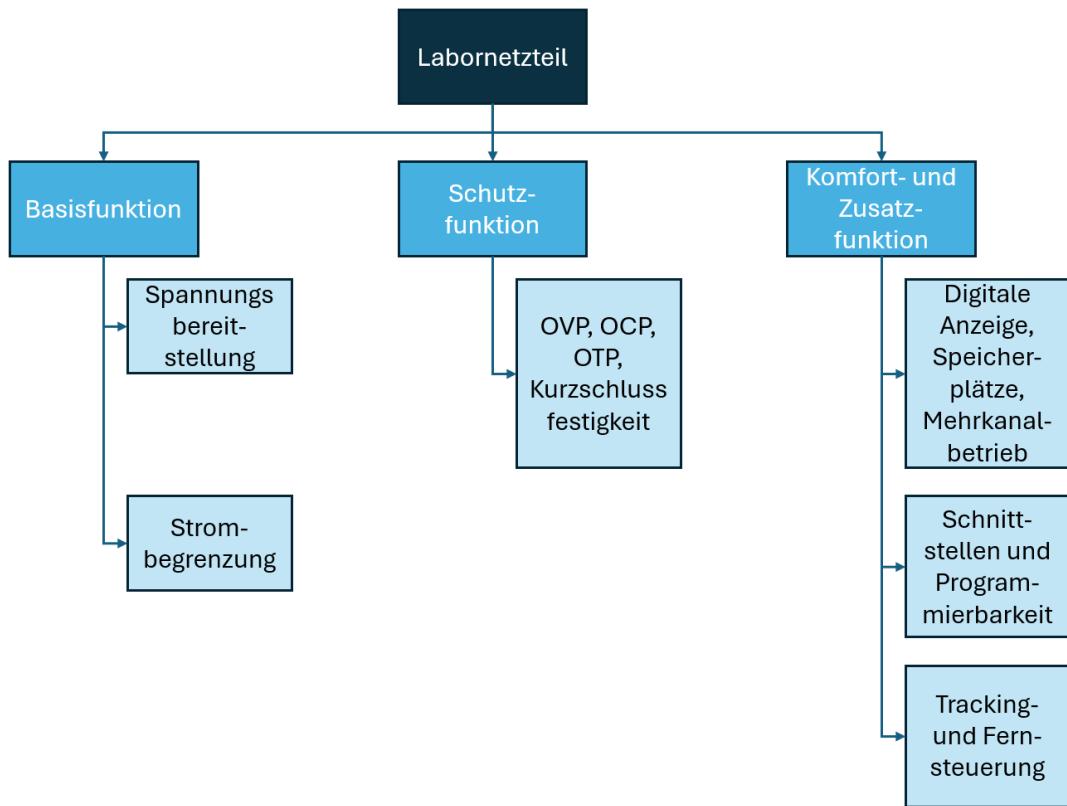


Abbildung 3.5: Funktionsgliederung eines Labornetzteils

3.2.2 Funktionsgliederung eines Oszilloskops

Ein Oszilloskop ist ein elektronisches Messgerät, das zeitabhängige elektrische Signale sichtbar macht. Es zeigt an, wie sich Spannung über die Zeit verändert, und ist damit eines der wichtigsten Werkzeuge in der Elektronik, Elektrotechnik und Physik. Im Gegensatz zu einem Multimeter, das nur Zahlenwerte (z. B. Spannung, Strom, Widerstand) liefert, stellt ein Oszilloskop die Signale grafisch als Kurvenverlauf dar (vgl. Abbildung 3.6).

Grundfunktionen eines Oszilloskops:

1. Signalaufnahme

- Eingangskanäle:** Anschlüsse für Tastköpfe (Probes), oft mehrere Kanäle (z. B. 2–4).
- Spannungsbereich:** Anpassung des Messbereichs (vertikale Verstärkung).
- Kopplung:** Auswahl zwischen DC-, AC- oder Massekopplung.

2. Signalverarbeitung

- Abtastrate:** Digitale Oszilloskope wandeln das Signal in Abtastwerte um.

- b. **Bandbreite:** Bestimmt, wie schnelle Signale zuverlässig erfasst werden.
- c. **Triggerung:** Startbedingung für die Darstellung, z. B. Flanke oder Pegel.

3. Signalanzeige

- a. **Zeitachse (horizontal):** Einstellbar zur Darstellung langsamer oder schneller Signale.
- b. **Spannungsachse (vertikal):** Darstellung der Amplitude.
- c. **Mehrkanaldarstellung:** Vergleich von mehreren Signalen gleichzeitig.

4. Mess- und Analysefunktionen

- a. **Grundmessungen:** Spannungsspitzen (U_{\max} , U_{\min}), Effektivwerte (RMS), Frequenz, Periodendauer.
- b. **Mathematische Funktionen:** FFT (Frequenzanalyse), Addition/Subtraktion von Kanälen, Integration/Differentiation.
- c. **Automatische Cursor-Messungen:** Manuelle oder automatische Marker für Zeit- und Spannungsdifferenzen.

5. Zusatzfunktionen

- a. **Speicherfunktion:** Signalaufzeichnungen speichern und wieder abrufen.
- b. **Schnittstellen:** USB, LAN, RS232 zur Datenübertragung.
- c. **Dekodierung von Bussystemen:** z. B. I²C, SPI, UART.
- d. **Mixed-Signal-Oszilloskop (MSO):** Kombination von analogen und digitalen Signalen.

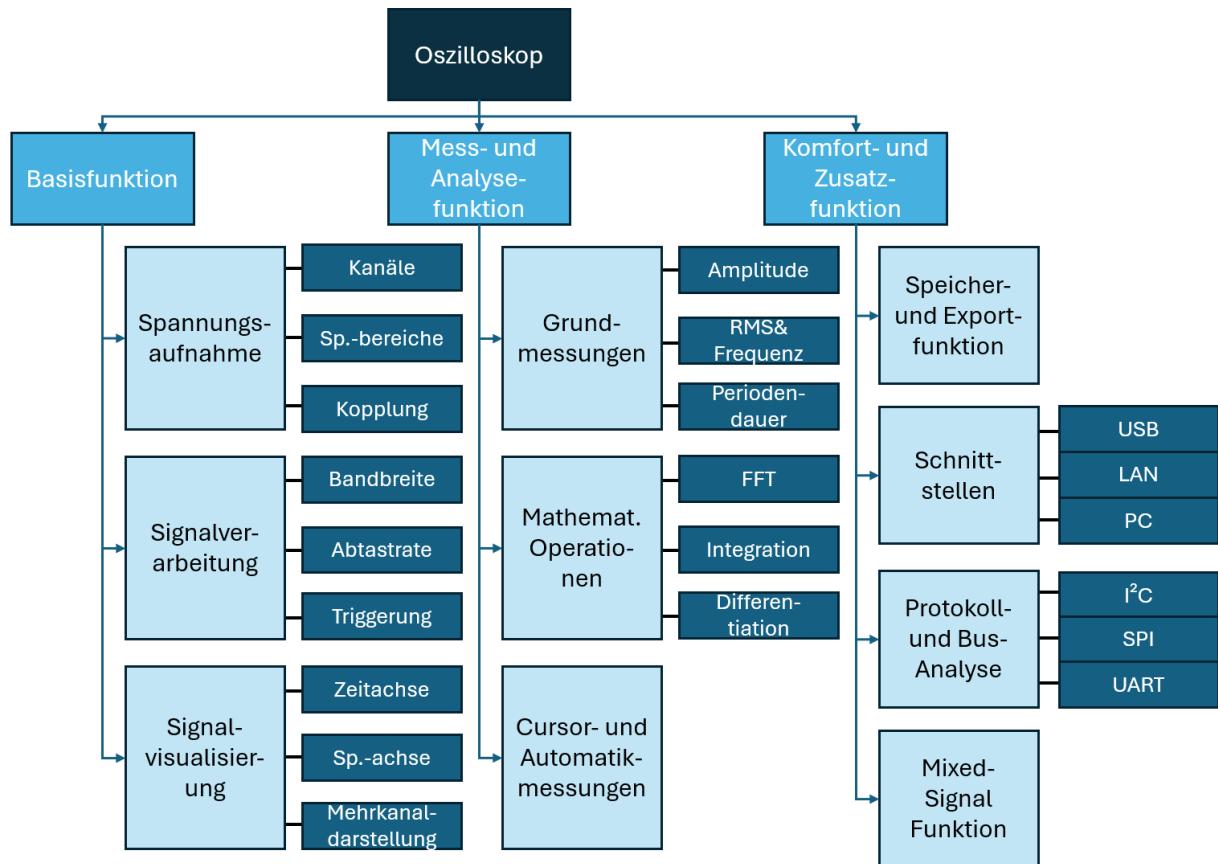


Abbildung 3.6: Funktionsgliederung eines Oszilloskops

3.2.3 Lösungskonzept

Das Lösungskonzept wurde schon vor dem Projektstart weitestgehend durch die Aufgabenstellung vorgegeben. Die Aufgabenstellung kann in der folgenden Infobox erneut eingesehen werden:

Die Aufgabenstellung:

Entwicklung und Evaluierung eines prototypischen Messdatenerfassungssystems basierend **auf Raspberry Pi und Digilent MCC DAQ HATs mit Oszilloskop-, Netzteil- und Multimeter-Funktionen.**

- Programmierung mit Python/ Jupyter Notebooks
- Standard-Messroutinen als Jupyter Notebooks
- Analoge Eingänge für Messungen, Analoge Ausgänge als Spannungsversorgung (mit kleiner Leistung)
- Realisierung der Peripherie
- Konstruktion eines Gehäuses mit Planung aller Anschlüsse und Stecker

Im Rahmen dieser Projektarbeit ist vorgesehen, dass das prototypische Messdatenerfassungssystem aus einem Raspberry Pi 5, einem Digilent MCC 118 HAT, einem Netzteil-Modul, einem Funktionsgenerator-Modul und einem Multimeter-Modul bestehen soll.

Die Teifunktionen des Oszilloskops sollen sowohl vom Digilent MCC 118 DAQ HAT als auch vom Raspberry Pi 5 übernommen werden. Das Digilent MCC 118 DAQ HAT übernimmt dabei die Aufgabe die Output-Signale einer beliebigen Schaltung aufzunehmen und dem Raspberry Pi 5 ordnungsgemäß zur Analyse und Auswertung weiterzugeben. Die Analyse, Visualisierung und Auswertung erfolgen, durch die in Python und Jupiter Notebooks entwickelte Software, welche wiederum mit den Raspberry Pi 5 gestartet und gesteuert werden soll.

Die Teifunktionen des Labornetzteils sollen durch ein Netzteil-Modul übernommen werden. Damit soll eine nahezu ähnliche Mobilität wie beim myDAQ von NI gewährleistet werden.

Das Funktionsgenerator-Modul wird entwickelt, um Schaltungen mit unterschiedlichen Eingangssignalen zu prüfen. Im Gegensatz zum Netzteil-Modul wird dabei das Funktionsgenerator-Modul nicht neu entwickelt, da bereits kostengünstige Funktionsgeneratoren für Mikrocontroller im Markt vorhanden sind. Demnach wird ein solcher Funktionsgenerator herangezogen und für unsere Anforderungen angepasst.

4 Hardwareentwicklung

In diesem Kapitel werden sämtliche entwickelten Schaltungen beschrieben, die zur Realisierung der einzelnen Funktionen des OurDAQ erforderlich sind. Die Darstellung umfasst sowohl die konzeptionelle Entwicklung als auch die praktische Umsetzung. Ergänzend werden die Simulationsergebnisse aus LTspice vorgestellt, die zur Verifikation und Optimierung der Schaltungsentwürfe vor dem Aufbau herangezogen wurden. [14]

4.1 Multimeter

4.1.1 Spannungsmessung

Die Spannungsmessung erfolgt direkt über das MCC 118 HAT-Modul, das einen Messbereich von ± 10 V sowohl für Gleich- (DC) als auch für Wechselspannungen (AC) bietet. Die Messwerte werden mit der Auflösung und Abtastrate des MCC 118 erfasst und anschließend in der Software weiterverarbeitet. [8]

Das MCC 118 gibt die Genauigkeit für diesen Messbereich wie folgt an:

- Offsetfehler: maximal ± 11 mV (fester Spannungsversatz unabhängig vom Messwert)
- Gain-Fehler (Verstärkungsfehler): maximal 0,098 % des Messwerts, wirkt proportional zur gemessenen Spannung
- Absolute Genauigkeit bei Vollaussteuerung (± 10 V): maximal $\pm 20,8$ mV

Damit entspricht der maximale Fehler im Vollbereich $\pm 0,208$ % des Messwertes, bei kleinen Spannungen wirkt sich vor allem der Offsetfehler stärker aus.

Da das MCC 118 die Spannungsmessung vollständig hardwareseitig unterstützt, war für diese Funktion keine zusätzliche Schaltung erforderlich. Entsprechend entfiel auch eine vorherige Simulation, sodass die Tests direkt mit dem HAT-Modul durchgeführt werden konnten.

4.1.2 Strommessung

Für die Strommessung wurde zunächst ein zweistufiger Aufbau mit dem Operationsverstärker OPA551 entwickelt und in LTspice simuliert. Dabei kamen zwei nichtinvertierende Verstärkerschaltungen zum Einsatz, um die Messbereiche ± 10 mA und $\pm 0,8$ A abzudecken (vgl. Abbildung 4.1 und 4.2). Im mA-Bereich wurde ein Shunt-Widerstand von $1\ \Omega$ mit einem Verstärkungsfaktor von 48 verwendet, im A-Bereich ein Shunt von $0,1\ \Omega$ mit einer Verstärkung von etwa 92.

.step param R_last_1 1k 100k 10k

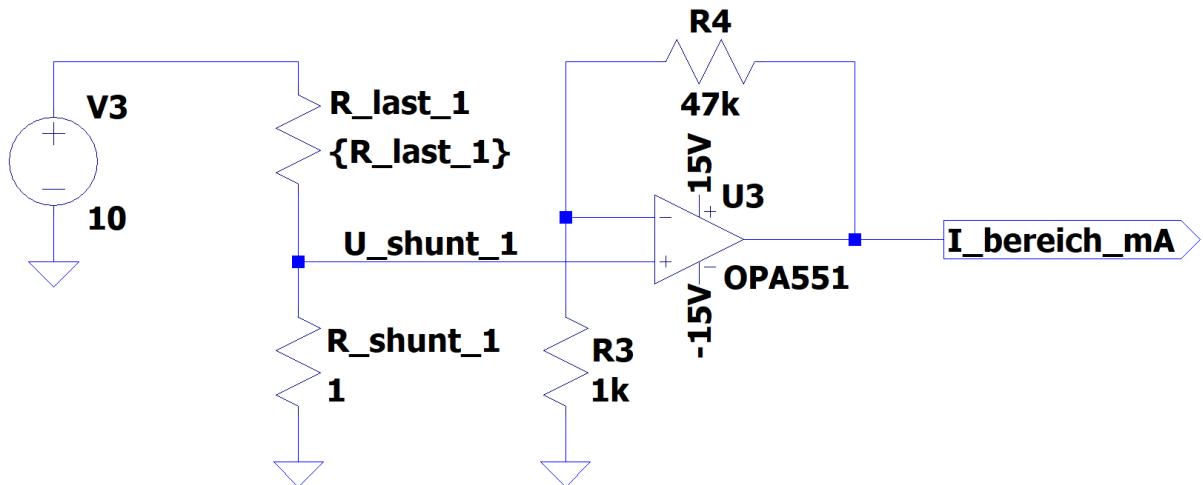


Abbildung 4.1: Strommessung OPA551 mA-Bereich

.step param R_last_2 10 900 100

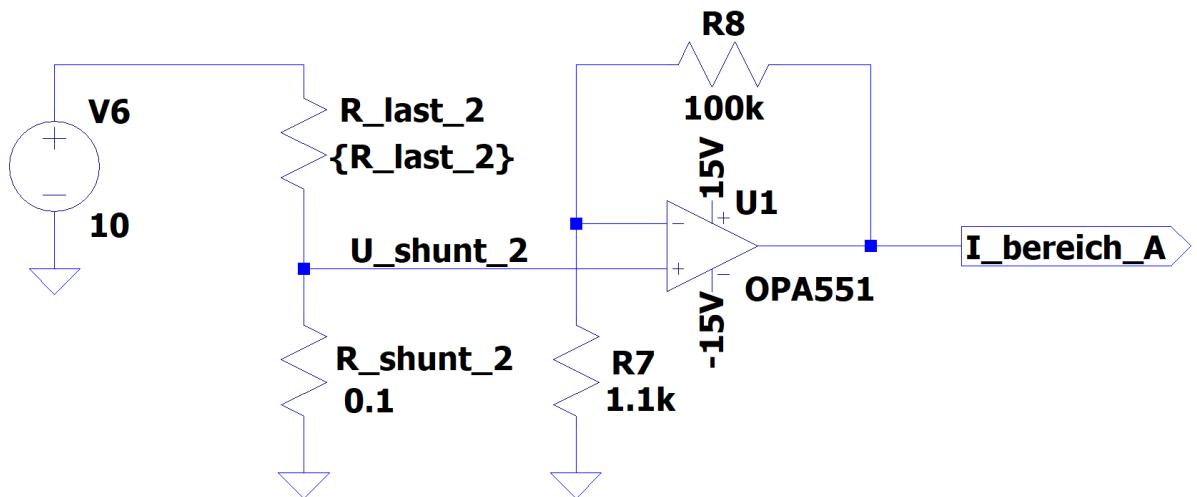


Abbildung 4.2: Strommessung OPA551 A-Bereich

Die Simulationsergebnisse (Abbildung 4.3 bis 4.6) zeigten in beiden Fällen eine korrekte Umsetzung. Sehr kleine Spannungen am Shunt (0–100 mV, U_shunt_1 und U_shunt_2) wurden präzise in entsprechend höhere Ausgangsspannungen (0 bis 9 V, I_bereich_mA und I_bereich_A) umgesetzt.

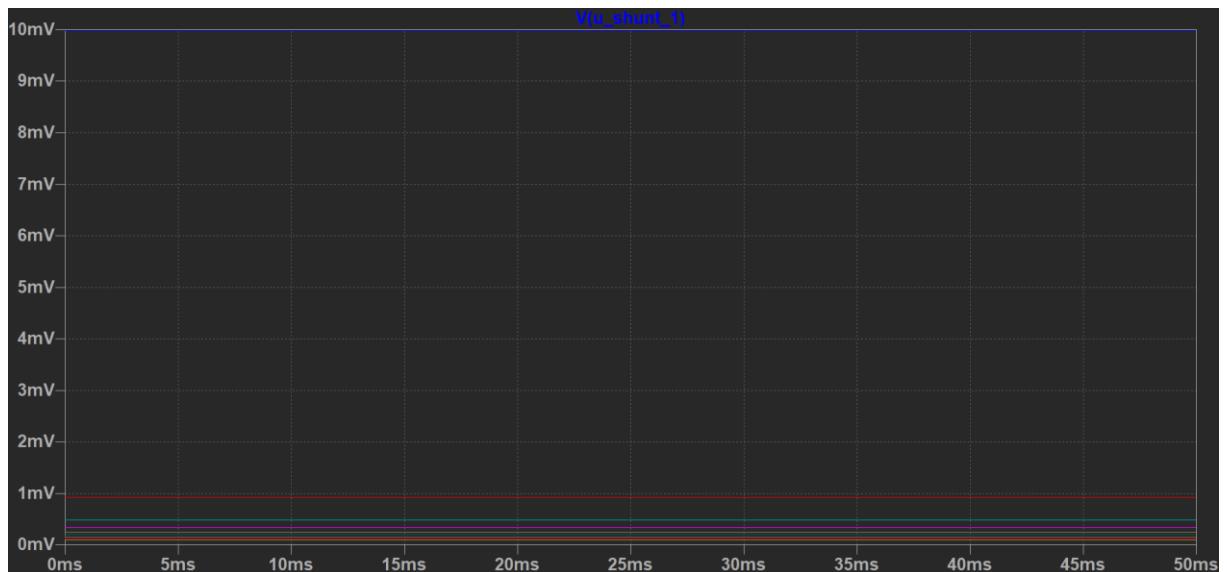


Abbildung 4.3: Shunt-Spannung mA-Bereich

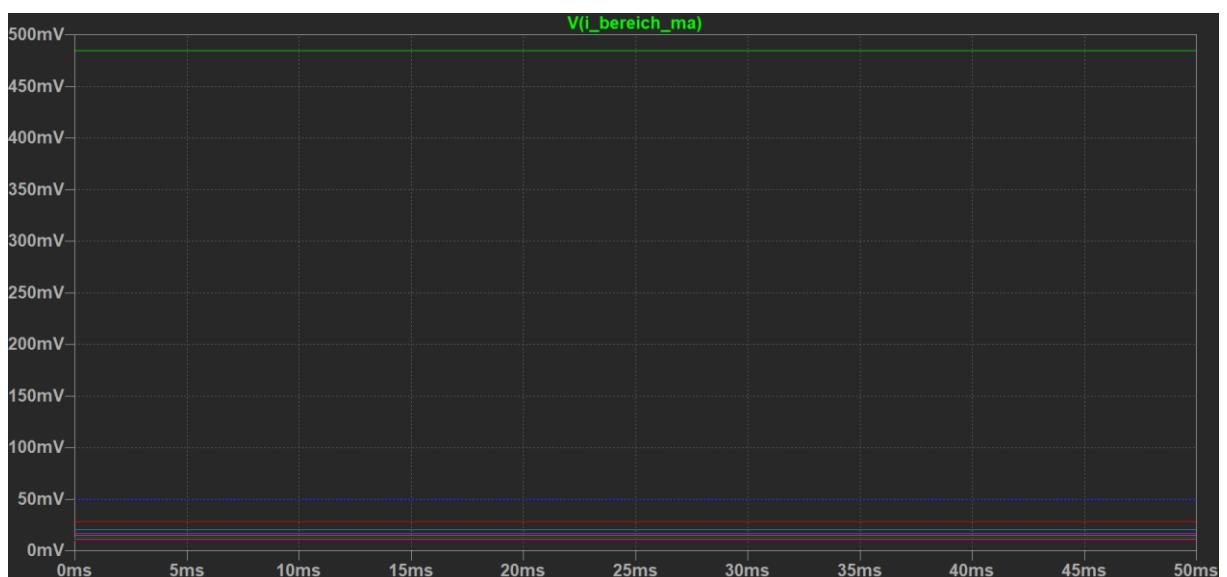


Abbildung 4.4: Verstärkte Spannung mA-Bereich

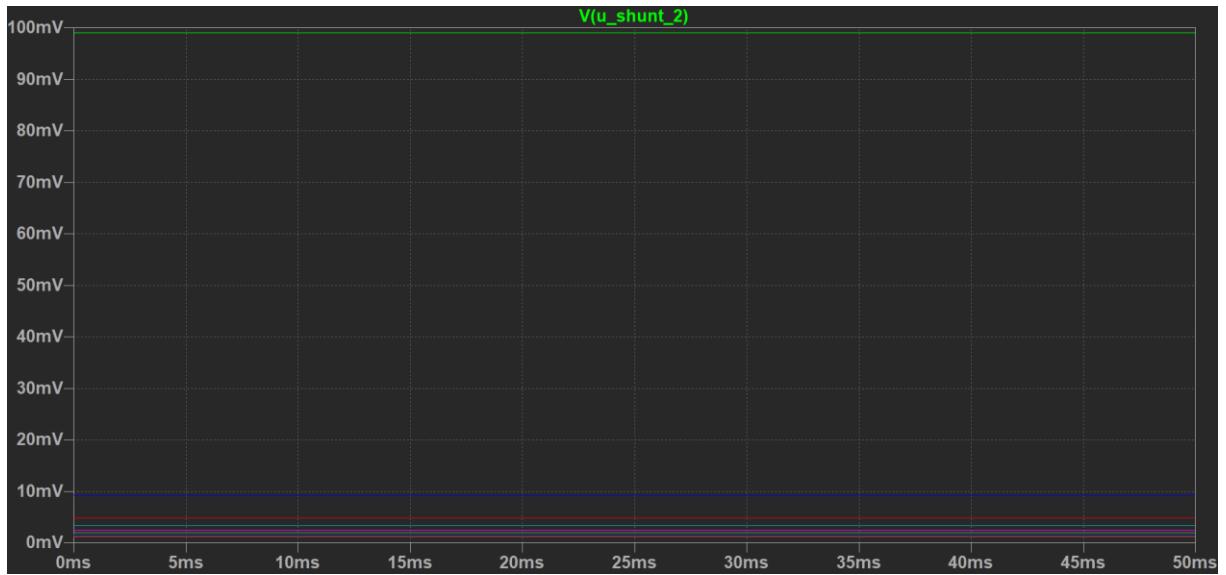


Abbildung 4.5: Shunt-Spannung A-Bereich



Abbildung 4.6: Verstärkte Spannung A-Bereich

Im praktischen Betrieb zeigte sich jedoch eine gravierende Einschränkung. Die Offset-Spannung des OPA551 beträgt laut Datenblatt bis zu ± 3 mV. [15]

Bei einer zu messenden Shunt-Spannung im Bereich weniger Millivolt führt dies zu erheblichen Messfehlern. Messungen mit realen Bauteilen bestätigten dies. So wurden bei einer Eingangsspannung von 2,44 mV wurden Ausgangsspannungen von 45,6 mV bzw. 87 mV gemessen (bei einer Verstärkung von 101), während rechnerisch etwa 246,4 mV zu erwarten gewesen wären.

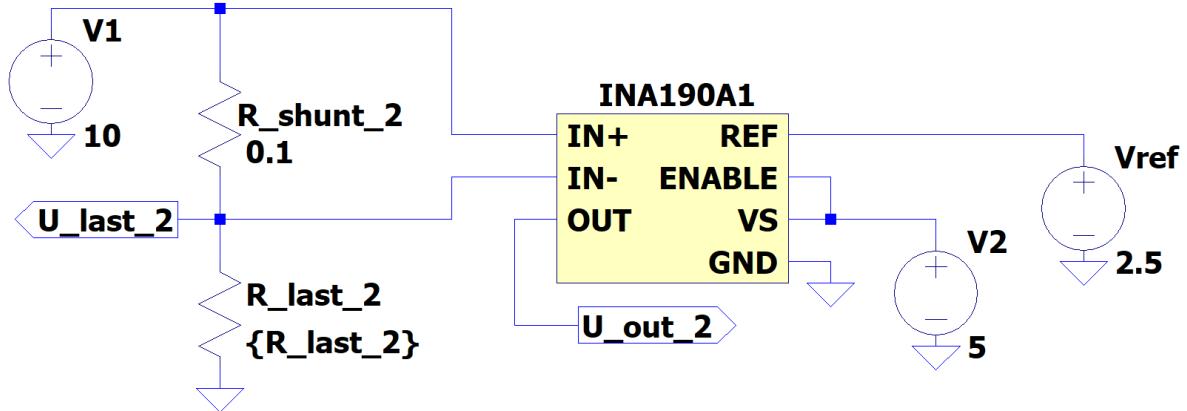
Diese Abweichungen machen deutlich, dass der OPA551 für diese präzise Strommessanwendung nicht geeignet ist. Die Ursache für die Diskrepanz zwischen Simulation und Realität liegt darin, dass Offset-Spannungen im verwendeten Spice-Modell nicht berücksichtigt wurden. Dies konnte im Nachgang durch eine LTspice-Simulation überprüft werden.

Der Plus-Eingang wurde hierbei auf Masse gelegt, der Minus-Eingang mit dem Ausgang verbunden. Ergibt die Simulation nahezu null am Ausgang, besitzt das Modell keinen eingebauten Offset. Bei dieser Simulation lag der Ausgang bei ca. 102 µV und damit weit unterhalb der im Datenblatt angegebenen ± 3 mV.

Auf Grundlage der zuvor gewonnenen Erkenntnisse wurde eine präzisere Strommessschaltung entwickelt, die zwei wählbare Messbereiche abdeckt: ± 10 mA und $\pm 0,8$ A. Hierfür kommen Messverstärker vom Typ INA190 in den Varianten A1 und A4 zum Einsatz, jeweils in Kombination mit geeigneten Shunt-Widerständen (vgl. Abbildung 4.7 und 4.8).

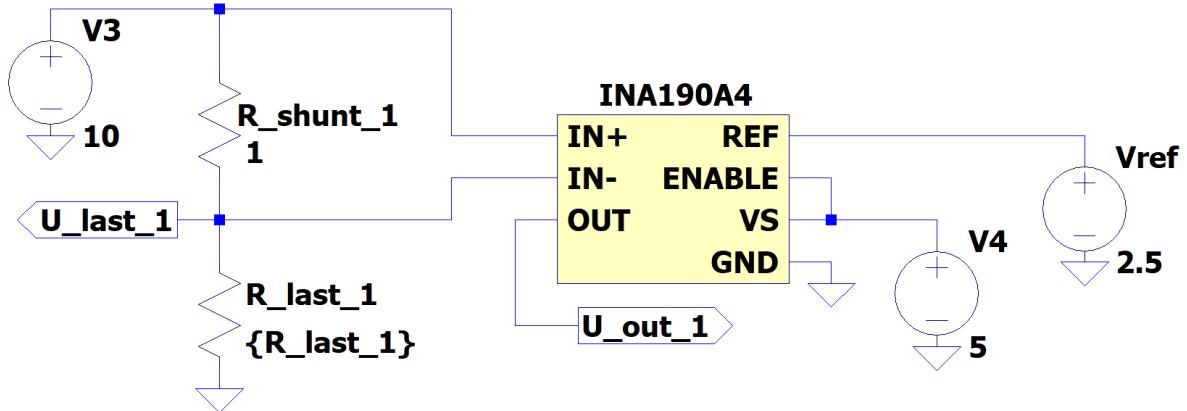
Die Schaltung ist als High-Side-Strommessung ausgeführt: [16]

- Die Last ist direkt mit Masse verbunden, was eine einfache Integration ermöglicht
- Kurzschlüsse zwischen Last und Masse lassen sich zuverlässig erfassen
- Es wird keine Massereferenz benötigt, wodurch eine flexible Einbindung möglich ist
- Nachteilig ist die Notwendigkeit einer hohen Gleichtaktunterdrückung, da die sehr kleine Shunt-Spannung knapp unterhalb der Versorgungsspannung der Last liegt



.step param R_last_2 13 600 80

Abbildung 4.7: Strommessung INA190A1 A-Bereich



.step param R_last_1 1k 10k 2k

Abbildung 4.8: Strommessung INA190A4 mA-Bereich

Für die beiden Messbereiche ergeben sich folgende Konfigurationen:

- Messverstärker INA190A1 (Verstärkung 25) mit Shunt 0,1 Ω für den ±0,8 A-Bereich
- Messverstärker INA190A4 (Verstärkung 200) mit Shunt 1 Ω für den ±10 mA-Bereich

Ein wesentlicher Vorteil dieser Lösung ist die sehr geringe Offset-Spannung von maximal ±15 µV, wodurch auch Spannungen im unteren Millivolt-Bereich nahezu unverfälscht erfasst werden. Die Referenzspannung wird auf 2,5 V gelegt, sodass ein symmetrischer Arbeitsbereich für positive wie auch negative Ströme entsteht. [17]

Die Ausgangsspannung des Messverstärkers wird schließlich über das MCC 118 eingelesen und in den zugehörigen Stromwert umgerechnet:

$$I = \frac{U_{out} - U_{ref}}{\text{Verstärkung} * R_{shunt}}$$

Die Abbildungen (4.9 bis 4.12) zeigen jeweils verschiedene Lastströme (R_{Last_2} und R_{Last_1}) sowie die dazugehörigen Ausgangsspannungen (U_{out_2} und U_{out_1}).



Abbildung 4.9: Laststrom A-Bereich

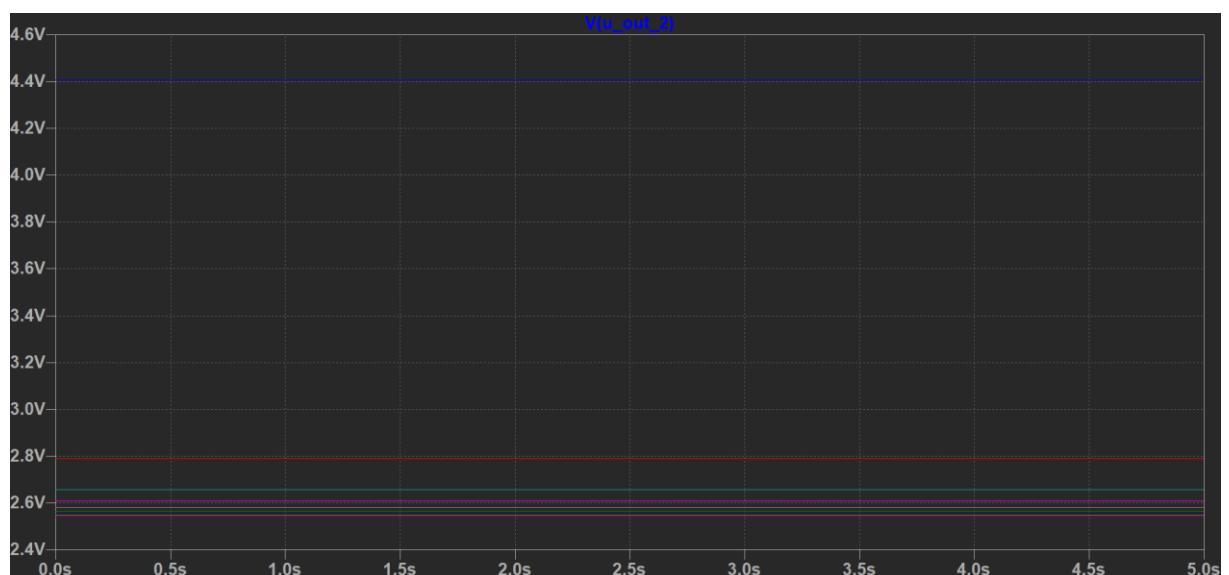


Abbildung 4.10: Ausgangsspannung A-Bereich (Messspannung für HAT-Modul)

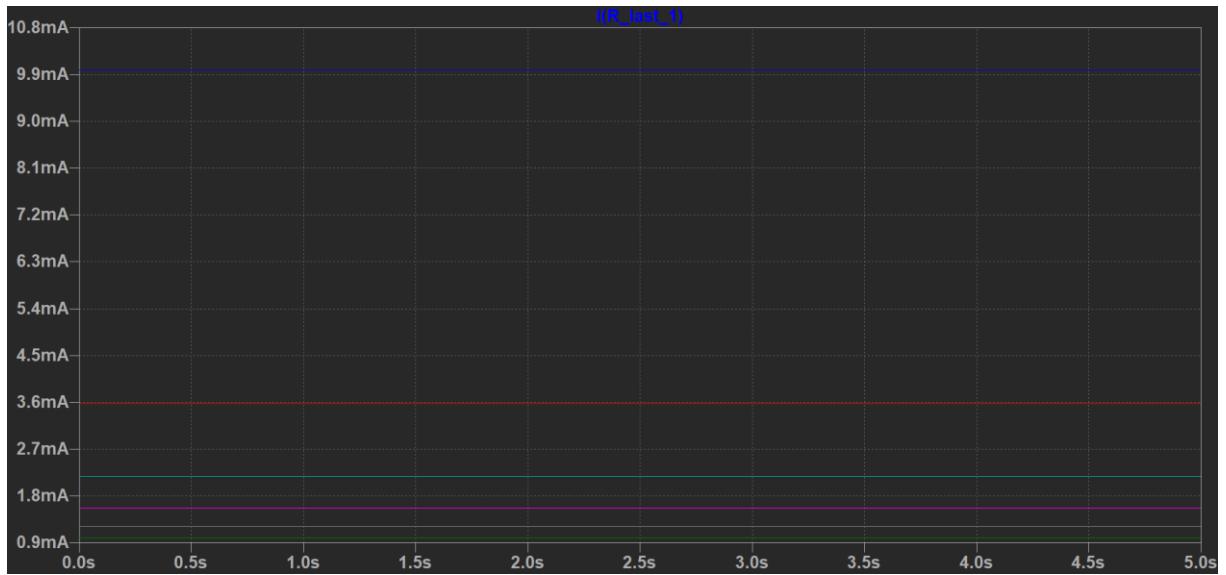


Abbildung 4.11: Laststrom mA-Bereich

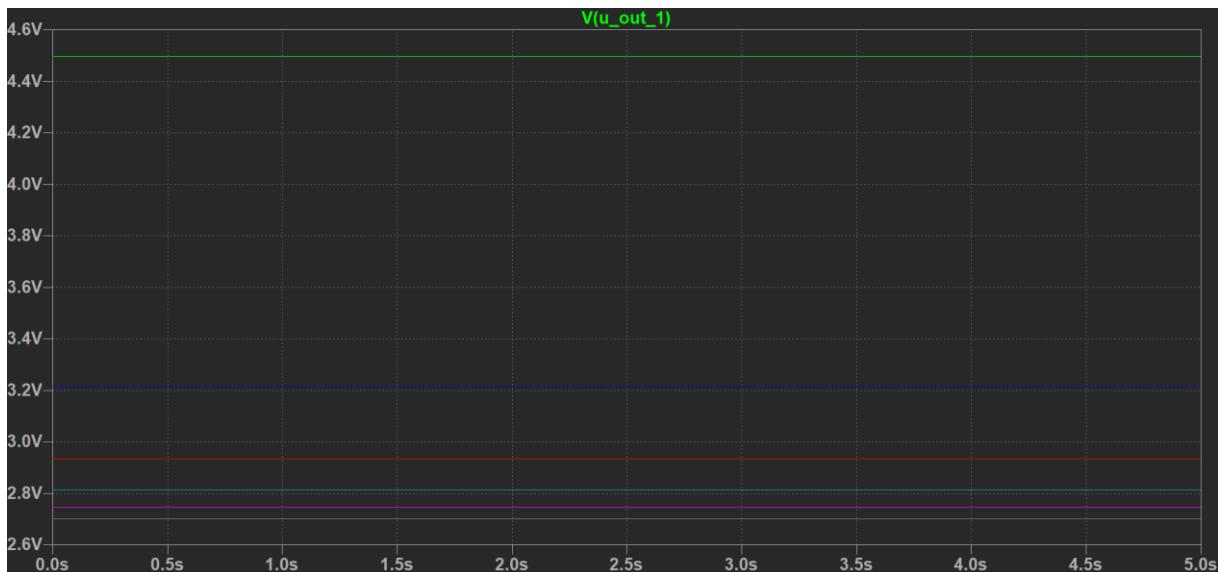


Abbildung 4.12: Ausgangsspannung mA-Bereich (Messspannung für HAT-Modul)

Die Umschaltung zwischen den beiden Messbereichen erfolgt softwaregesteuert über einen ADG419-Analogschalter, sodass lediglich ein Kanal des MCC 118 für die Strommessung erforderlich ist. Die Referenzspannung für die INA190 wird über eine Z-Diode (2,5 V) und einen Impedanzwandler (OPA 551) erzeugt (vgl. Abbildung 4.13).

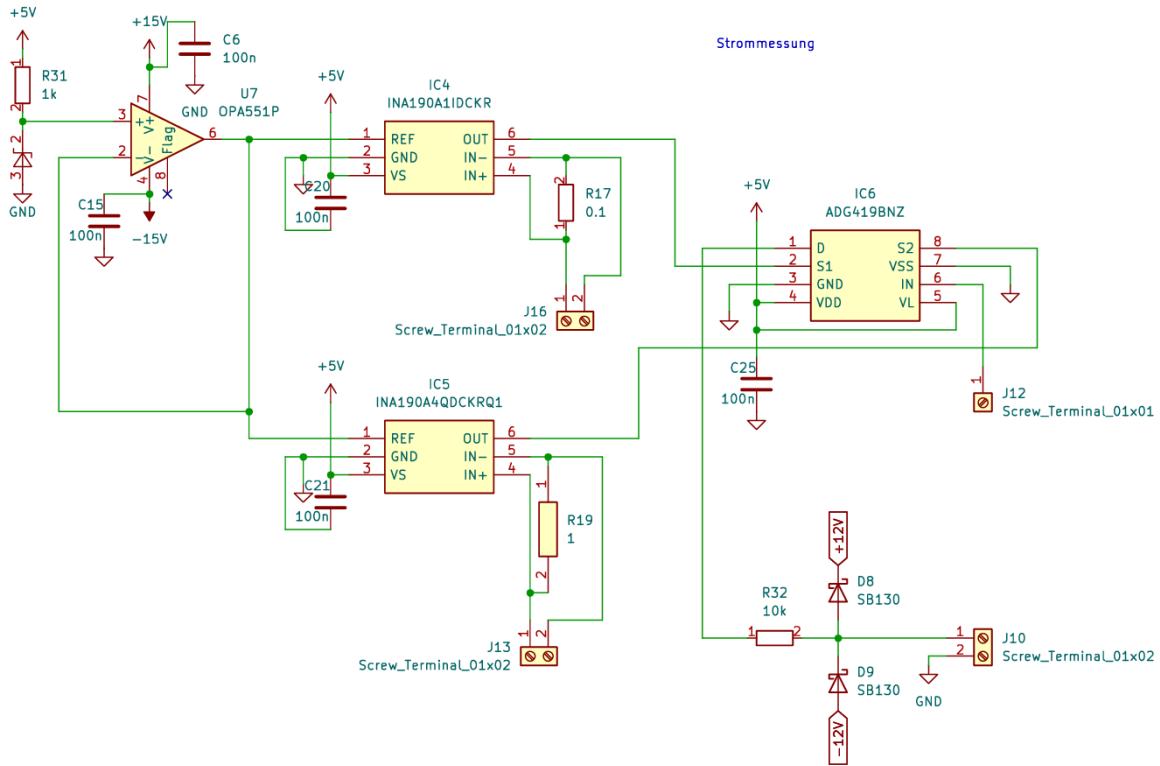


Abbildung 4.13: Umschaltung Messbereiche mit ADG419

Genauigkeit

Die Strommessung basiert auf

$$I = \frac{U_{Shunt}}{R_{Shunt}}$$

wobei U_{Shunt} durch den Messverstärker (INA190) verstärkt und dann vom MCC 118 digitalisiert wird. Fehler entstehen durch:

1. Shunt-Widerstand: Toleranz (z. B. $\pm 1\%$) -> direkter Einfluss, proportional zum Strom
2. INA190 maximaler Verstärkungsfehler:
 - $\pm 0,2\%$ (A1)
 - $\pm 0,3\%$ (A4)
3. INA190 Offsetspannung: maximal $\pm 15\text{ }\mu\text{V}$ -> umgerechnet über den Shunt und die Verstärkung in einen Fehlerstrom
4. MCC 118: Gain-Fehler (0,098 %) und Offset-Fehler (bis $\pm 11\text{ mV}$)

Genauigkeitsbetrachtung A-Bereich

Prozentualer Anteil: Dieser Anteil hängt von den Toleranzen der Messkette ab.

a) Worst-Case Addition

$$1\% + 0,2\% + 0,098\% \approx 1,3\%$$

-> sehr konservativ, für den Fall, dass alle Fehler in die gleiche Richtung wirken.

b) realistische RSS-Bildung (Wurzel-Summe der Quadrate): [18]

$$\left(\sqrt{1^2 + 0,2^2 + 0,098^2} \right) \% \approx 1,02\%$$

-> realistischer Fehler: ca. 1,0 % vom Messwert.

Konstanter Anteil: Das ist der Offset-Fehler, also ein fester Betrag unabhängig vom Strom.

Setzt sich zusammen aus:

- INA190 Offsetspannung: $\pm 15 \mu V$
- MCC 118 Offset-Fehler: $\pm 11 mV$

Umrechnung in Stromwert:

$$I = \frac{15 \mu V}{R_{Shunt} * G}$$

Hier folgt die Berechnung für den A-Bereich (Shunt = $0,1 \Omega$, Verstärkung G = 25)

MCC 118 Offset:

$$I_{MCC} = \frac{11 mV}{0,1 \Omega * 25} = 4,4 mA$$

INA190 Offset:

$$I_{INA} = \frac{15 \mu V}{0,1 \Omega * 25} = 6,0 \mu A$$

-> Summe $\approx 4,5$ mA Offset.

Der Offset wirkt besonders bei kleinen Strömen stark, daher ist ein Nullabgleich (stromlos messen und subtrahieren), sehr empfehlenswert.

So würde nur noch der Restdrift übrigbleiben (typischerweise wenige 100 µA). Auf Grund dessen folgt die Angabe ±0,5 mA nach dem Nullabgleich in Tabelle (4.1).

Genauigkeitsbetrachtung mA-Bereich

Prozentualer Anteil

a) Worst-Case Addition

$$0,1\% + 0,3\% + 0,098\% \approx 0,5\%$$

b) realistische RSS-Bildung (Wurzel-Summe der Quadrate):

$$\left(\sqrt{0,1^2 + 0,3^2 + 0,098^2} \right) \% \approx 0,33\%$$

-> realistischer Fehler: ca. 0,33 % vom Messwert.

Konstanter Anteil

MCC 118 Offset:

$$I_{MCC} = \frac{11 \text{ mV}}{1 \Omega * 200} \approx 0,06 \text{ mA}$$

INA190 Offset:

$$I_{INA} = \frac{15 \mu V}{1 \Omega * 200} = 0,075 \mu A \approx 0 \text{ mA}$$

Endformel für Messgenauigkeit

Fehler = ± (prozentualer Anteil vom Messwert + Offset)

In Tabelle 4.1 sind die Charakteristiken für die Strommessung zusammengefasst.

A-Bereich	mA-Bereich
Shunt: 0,1 Ω (±1 %)	Shunt: 1 Ω (±0,1 %)
Verstärker: INA190A1	Verstärker: INA190A4
Verstärkung: 25	Verstärkung: 200
Max. Strom: ±0,8 A	Max. Strom: ±10 mA

Messgenauigkeit:	Messgenauigkeit:
<ul style="list-style-type: none"> Nach Nullabgleich: $\pm (1,0\% \text{ vom Messwert} + 0,5 \text{ mA})$ Ohne Nullabgleich: $\pm (1,0\% \text{ vom Messwert} + 4,4 \text{ mA})$ 	<ul style="list-style-type: none"> Nach Nullabgleich: $\pm (0,33\% \text{ vom Messwert} + 0,0 \text{ mA})$ Ohne Nullabgleich: $\pm (0,33\% \text{ vom Messwert} + 0,07 \text{ mA})$

Tabelle 4.1: Strommessung - Zusammenfassung

Um die Genauigkeit deutlich zu verbessern, ist ein Nullabgleich in beiden Messbereichen sehr empfehlenswert. Eine mögliche Erweiterung wäre, mit einem zusätzlichen Kanal, falls dieser vorhanden wäre, den Offset kontinuierlich mitzuerfassen und softwareseitig herauszurechnen. Dadurch ließe sich die Genauigkeit noch weiter steigern.

4.2 Oszilloskop

Zur Signalerfassung im Oszilloskop-Modus können Rechteck-, Dreieck- und Sinusformen mit einer Amplitude von bis zu $\pm 10 \text{ V}$ direkt über das MCC 118 aufgezeichnet werden. Es stehen zwei Kanäle zur Verfügung, sodass zeitgleiche Messungen an unterschiedlichen Signalquellen möglich sind. Die erfassten Signale werden mit der Auflösung und Abtastrate des MCC 118 aufgenommen und in der Software dargestellt.

Da die Messung ausschließlich über das HAT-Modul erfolgt und keine zusätzlichen Messschaltungen notwendig sind, entfällt für den Oszilloskop-Modus eine Simulation in LTspice. Die Tests konnten direkt mit der realen Hardware durchgeführt werden.

In Tabelle 4.2 sind Richtwerte für die erforderliche Abtastrate in Abhängigkeit von der Signalform dargestellt. [19, 20]

Signalform	Oberwellen, Komplexität	Empfohlene Samples pro Periode	Begründung
Sinus	gering	10 – 20	Glatte Welle, kaum Oberwellen, ausreichend für visuelle Darstellung
Dreieck	mittel	15 – 25	Enthält ungerade Oberwellen -> mehr Punkte nötig für klare Darstellung
Rechteck	hoch	20 – 40	Viele ungerade Oberwellen, Oversampling (3-5x) empfohlen

Tabelle 4.2: Oszilloskop - Übersicht Abtastraten in Abhängigkeit der Signalform

Daraus ergibt sich bei der maximalen Abtastrate $f_s = 100 \text{ kS/s}$ die maximal sinnvoll darstellbare Frequenz:

$$f_{max} = \frac{f_s}{Samples \text{ per Period}}$$

- Sinus: 5 - 10 kHz
- Dreieck: 4 - 6,7 kHz
- Rechteck: 2,5 - 5 kHz

Werden beide Kanäle parallel betrieben, halbiert sich die Abtastrate pro Kanal ($f_s = 50 \text{ kS/s}$), und die Frequenzgrenzen verringern sich entsprechend.

Wird lediglich die Grundfrequenz benötigt (z. B. für Spektrumanalyse), reicht nach Nyquist bereits $f_s \geq 2 * f_{\text{signal}}$. Für eine visuell saubere Darstellung der Wellenform sind jedoch die höheren Faktoren aus der Tabelle empfehlenswert. [19]

Genauigkeit und Phasenfehler

Die Tabelle 4.3 zeigt die Genauigkeit eines einzelnen Kanals bei voller Abtastrate (100 kS/s) für ein Sinus-Signal.

$U_{ss} [\text{V}]$	$U_s [\text{V}]$	$\Delta U [\text{mV}]$	$\Delta\phi @ 1 \text{ kHz}$	$\Delta\phi @ 5 \text{ kHz}$	$\Delta\phi @ 10 \text{ kHz}$	Samples/Periode
2,0	1,0	$\pm 11,98$	$3,6^\circ$	18°	36°	10 - 20
4,0	2,0	$\pm 12,96$	$3,6^\circ$	18°	36°	10 - 20
6,0	3,0	$\pm 13,94$	$3,6^\circ$	18°	36°	10 - 20
8,0	4,0	$\pm 14,92$	$3,6^\circ$	18°	36°	10 - 20
10,0	5,0	$\pm 15,90$	$3,6^\circ$	18°	36°	10 - 20
20,0	10	$\pm 20,80$	$3,6^\circ$	18°	36°	10 - 20

Tabelle 4.3: Oszilloskop - Genauigkeit Sinus

Der Amplitude-Fehler ΔU wird durch den ADC Offset-Fehler (max. 11 mV) und den Gain-Fehler (0,098 % vom gelesenen Wert) bestimmt.

$$\Delta U = \pm (11 \text{ mV} + 0,00098 * U_s)$$

Der Phasenfehler $\Delta\phi$ entsteht, weil das MCC HAT nur diskrete Abtastungen durchführt. Bei einer Abtastrate von $f_s = 100 \text{ kS/s}$ beträgt der zeitlich Abstand zwischen zwei Messpunkten:

$$\Delta t = \frac{1}{f_s} = \frac{1}{100\,000} \frac{1}{\text{s}} = 10 \mu\text{s}$$

und somit der Phasenfehler $\Delta\phi$:

$$\Delta\varphi = f * \Delta t * 360^\circ$$

In der Praxis lässt sich der Phasenfehler durch geeignete mathematische Methoden in der Software größtenteils korrigieren, da es sich um einen deterministischen Effekt handelt, der berechnet und kompensiert werden kann. [21]

4.3 Netzteil

Der Aufbau für das Netzteil basiert auf einem DAC, einem Operationsverstärker und einem MOSFET (vgl. Abbildung 4.14). Die Sollspannung wird vom Raspberry Pi vorgegeben und über die SPI-Schnittstelle an einen MCP4922 DAC übertragen. Dieser wandelt digitale Werte zwischen 0 und 4095 (12 Bit) in eine analoge Spannung von 0 bis 5 V um. [22]

Die DAC-Ausgangsspannung (U_{dac_out}) wird mit einem OPA551 nichtinvertierend verstärkt und auf an das Gate eines IRLZ44N n-Kanal-MOSFET geführt. Durch die variable Gate-Source-Spannung kann der MOSFET die positive Ausgangsspannung des Netzteils stufenlos zwischen 0 und 10 V regeln (siehe Abbildung 4.15).

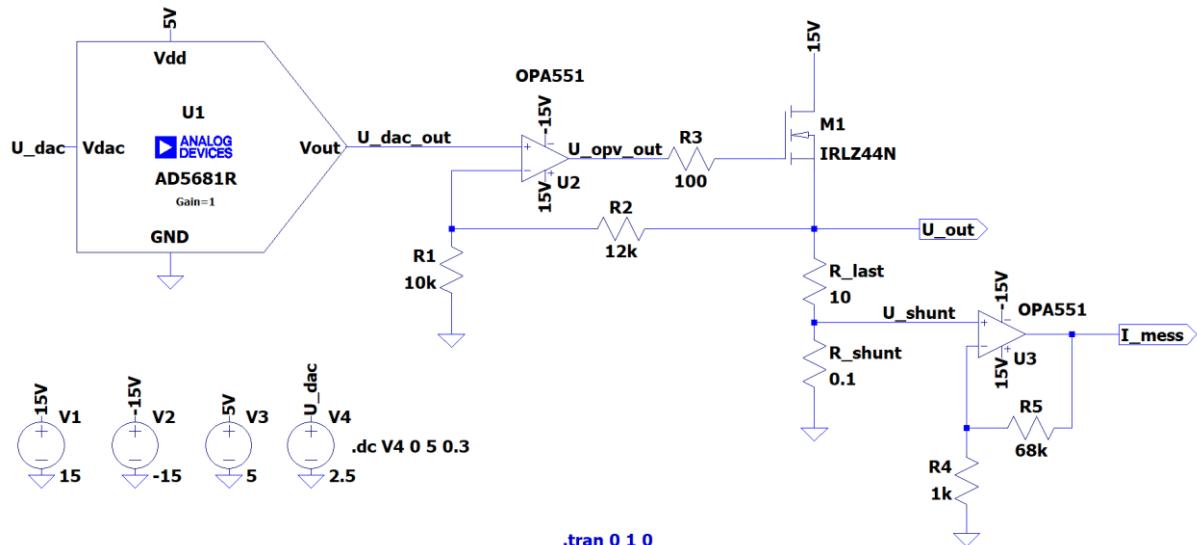


Abbildung 4.14: Aufbau Netzteil positiv



Abbildung 4.15: DAC-Spannung vs. Netzteil positiv Ausgangsspannung

Die Ausgangsspannung wird über eine Rückkopplung auf den OPA551 geführt, wodurch eine präzise Spannungsregelung möglich ist. Es können Ströme bis zu ca. 1 A bereitgestellt werden. Zur Wärmeableitung ist der MOSFET mit einem berechneten Kühlkörper ausgestattet, dessen Dimensionierung auf Basis der maximalen Verlustleistung erfolgt ist. Tabelle 4.4 enthält alle dazugehörigen Daten. [23]

Datenblatt IRLZ44N	Eigene Vorgaben (Simulation)
$R_{JA,max} = 62 \frac{^{\circ}C}{W}$	$P_{v,max} = 6 W$ (LTspice)
$R_{JC,max} = 1,4 \frac{^{\circ}C}{W}$	$\vartheta_{A,max} = 30 ^{\circ}C$
$R_{CS} = 0,5 \frac{^{\circ}C}{W}$	$\vartheta_{J,max} = 100 ^{\circ}C$

Tabelle 4.4: Netzteil – Übersicht Kühlkörper-Berechnung n-Kanal-MOSFET

Kühlkörperberechnung: [24]

$$R_{ges,max} = \frac{\vartheta_{J,max} - \vartheta_{A,max}}{P} = \frac{100 ^{\circ}C - 30 ^{\circ}C}{6 W} = 11,7 \frac{^{\circ}C}{W}$$

$$R_{ges} = R_{JC} + R_{CS} + R_{SA}$$

$$\rightarrow R_{SA,max} = 11,7 \frac{^{\circ}C}{W} - 1,4 \frac{^{\circ}C}{W} - 0,5 \frac{^{\circ}C}{W} = 9,8 \frac{^{\circ}C}{W}$$

Die praktische Wahl fiel somit auf einen Kühlkörper mit $9,0 \frac{^{\circ}C}{W}$.

Überprüfung der Sperrsichttemperatur:

$$\vartheta_J = \vartheta_A + P * (R_{JC} + R_{CS} + R_{SA}) = 30 ^{\circ}C + 6 W * \left(1,4 \frac{^{\circ}C}{W} + 0,5 \frac{^{\circ}C}{W} + 9 \frac{^{\circ}C}{W} \right) = 95,4 ^{\circ}C$$

à Der n-Kanal-MOSFET erwärmt sich mit dem verwendeten Kühlkörper ($9,0 \frac{^{\circ}C}{W}$) unter voller Belastung auf ca. $95,4 ^{\circ}C$.

Die Schaltung für die negative Ausgangsspannung ist im Wesentlichen identisch zur positiven Seite. Der Unterschied besteht darin, dass die vom DAC ausgegebene Spannung zunächst über einen weiteren OPA551 in den Bereich von 0 V bis -12 V skaliert wird. Diese skalierte Spannung steuert anschließend einen p-Kanal-MOSFET, der es ermöglicht, die negative Ausgangsspannung stufenlos zwischen 0 und -10 V einzustellen (vgl. Abbildung 4.16 und 4.17).

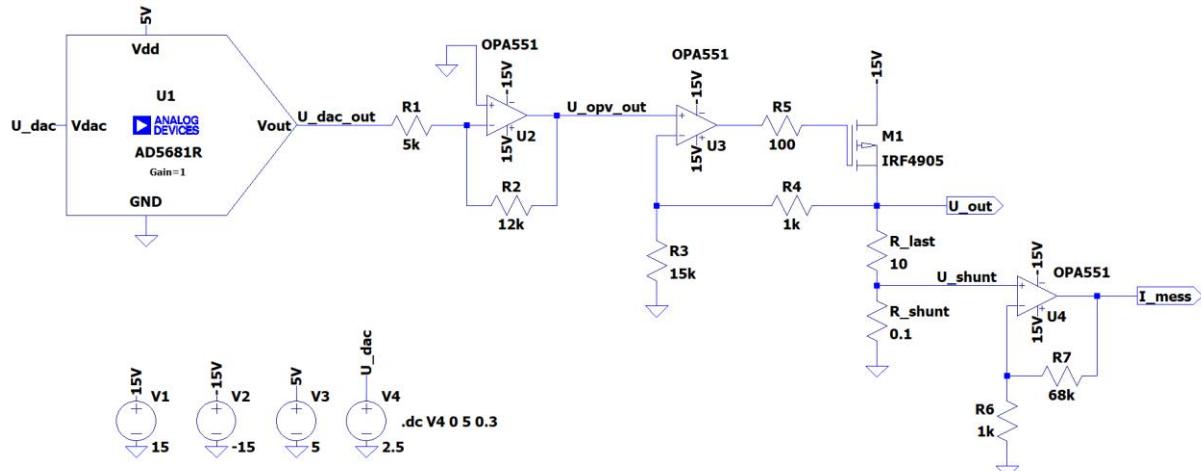


Abbildung 4.16: Aufbau Netzteil negativ

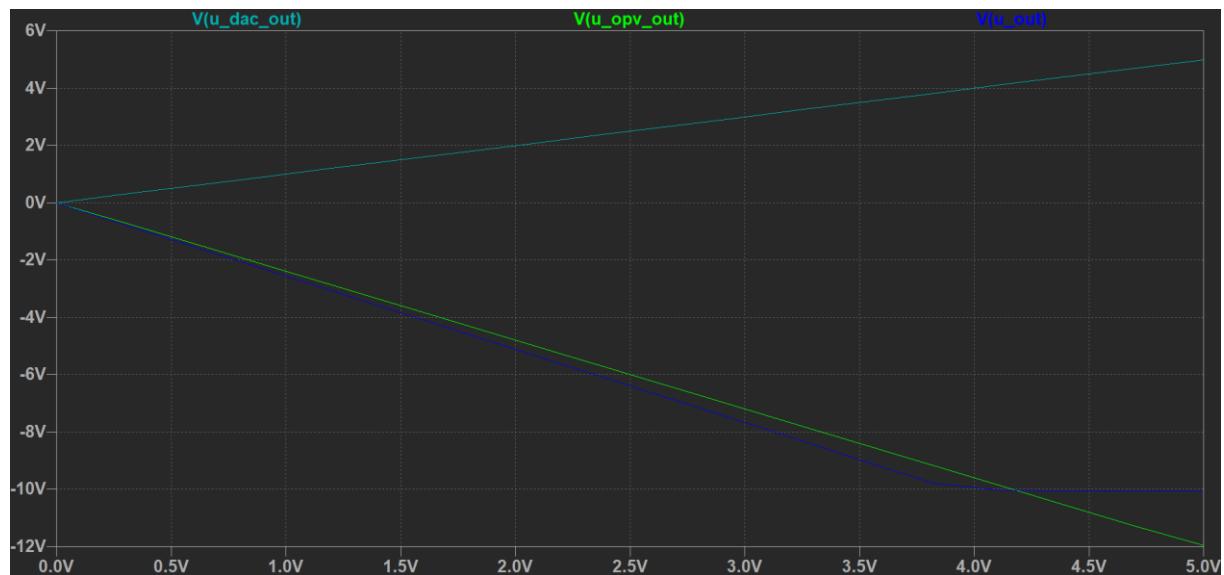


Abbildung 4.17:DAC-Spannung vs. Netzteil negativ Ausgangsspannung

Die Daten für die Dimensionierung für den Kühlkörper ist in Tabelle 4.5 dargestellt. [25]

Datenblatt IRF4905	Eigene Vorgaben (Simulation)
$R_{JA,max} = 62 \frac{^{\circ}C}{W}$	$P_{v,max} = 6 W$ (LTspice)
$R_{JC,max} = 0,75 \frac{^{\circ}C}{W}$	$\vartheta_{A,max} = 30 ^{\circ}C$
$R_{CS} = 0,5 \frac{^{\circ}C}{W}$	$\vartheta_{J,max} = 100 ^{\circ}C$

Tabelle 4.5: Netzteil – Übersicht Kühlkörper-Berechnung p-Kanal-MOSFET

Kühlkörperberechnung: [24]

$$R_{ges,max} = \frac{\vartheta_{J,max} - \vartheta_{A,max}}{P} = \frac{100\text{ }^{\circ}\text{C} - 30\text{ }^{\circ}\text{C}}{6\text{ W}} = 11,7\text{ }\frac{{}^{\circ}\text{C}}{\text{W}}$$

$$\begin{aligned} R_{ges} &= R_{JC} + R_{CS} + R_{SA} \\ \rightarrow R_{SA,max} &= 11,7\text{ }\frac{{}^{\circ}\text{C}}{\text{W}} - 0,75\text{ }\frac{{}^{\circ}\text{C}}{\text{W}} - 0,5\text{ }\frac{{}^{\circ}\text{C}}{\text{W}} = 10,45\text{ }\frac{{}^{\circ}\text{C}}{\text{W}} \end{aligned}$$

Die praktische Wahl fiel somit auf einen Kühlkörper mit $9,0\frac{{}^{\circ}\text{C}}{\text{W}}$.

Überprüfung der Sperrsichttemperatur:

$$\vartheta_J = \vartheta_A + P * (R_{JC} + R_{CS} + R_{SA}) = 30\text{ }^{\circ}\text{C} + 6\text{ W} * \left(0,75\text{ }\frac{{}^{\circ}\text{C}}{\text{W}} + 0,5\text{ }\frac{{}^{\circ}\text{C}}{\text{W}} + 9\text{ }\frac{{}^{\circ}\text{C}}{\text{W}} \right) = 91,5\text{ }^{\circ}\text{C}$$

à Der p-Kanal-MOSFET erwärmt sich mit dem verwendeten Kühlkörper ($9,0\frac{{}^{\circ}\text{C}}{\text{W}}$) unter voller Belastung auf ca. $91,5\text{ }^{\circ}\text{C}$.

Für die LTspice-Simulationen wurde auf beiden Netzteilseiten ein AD5681R verwendet, da für den im realen Aufbau eingesetzten MCP4922 kein passendes Simulationsmodell verfügbar ist. Der MCP4922 bietet zwei Kanäle in einem Gehäuse und ist in seinen elektrischen Eigenschaften dem AD5681R sehr ähnlich, sodass die Simulationsergebnisse auf die praktische Umsetzung übertragbar sind. [26, 27]

Zur Überstromsicherung wird der Ausgangsstrom beider Netzteilkanäle direkt auf der Lastseite erfasst. Hierzu sind jeweils Shunt-Widerstände mit einem Wert von $0,1\text{ }\Omega$ in Serie zum Ausgang geschaltet. Der Spannungsabfall über diesen Shunt wird über einen OPA551 in nichtinvertierender Grundschaltung mit einer Verstärkung von 69 verstärkt.

Die auf diese Weise aufbereiteten Messsignale werden über zwei separate Kanäle des MCC 118 eingelesen, ein Kanal für die positive und einer für die negative Versorgung. Wird ein zuvor definierter Grenzwert (maximal 1 A) erreicht oder überschritten, setzt die Steuerelektronik den DAC-Ausgang unmittelbar auf 0 V. Dadurch wird der jeweilige MOSFET geschlossen und der betroffene Netzteilkanal abgeschaltet. Die simulierte Strommessung ist für beide Kanäle in Abbildung 4.18 und 4.19 dargestellt. Hier kann jeweils die Spannung am Shunt-Widerstand (U_{shn}), der Strom durch die Last (R_{last}) sowie die verstärkte Spannung am Ausgang vom OPV (I_{mess}) erkannt werden.

Der OPA551 ist, wie bereits bei der Multimeterfunktion festgestellt, nicht optimal für die Messung sehr kleiner Ströme im mA-Bereich geeignet. Da es sich bei der Strombegrenzung jedoch um Ströme bis $\pm 0,8\text{ A}$ handelt, tritt diese Problematik hier nicht auf.

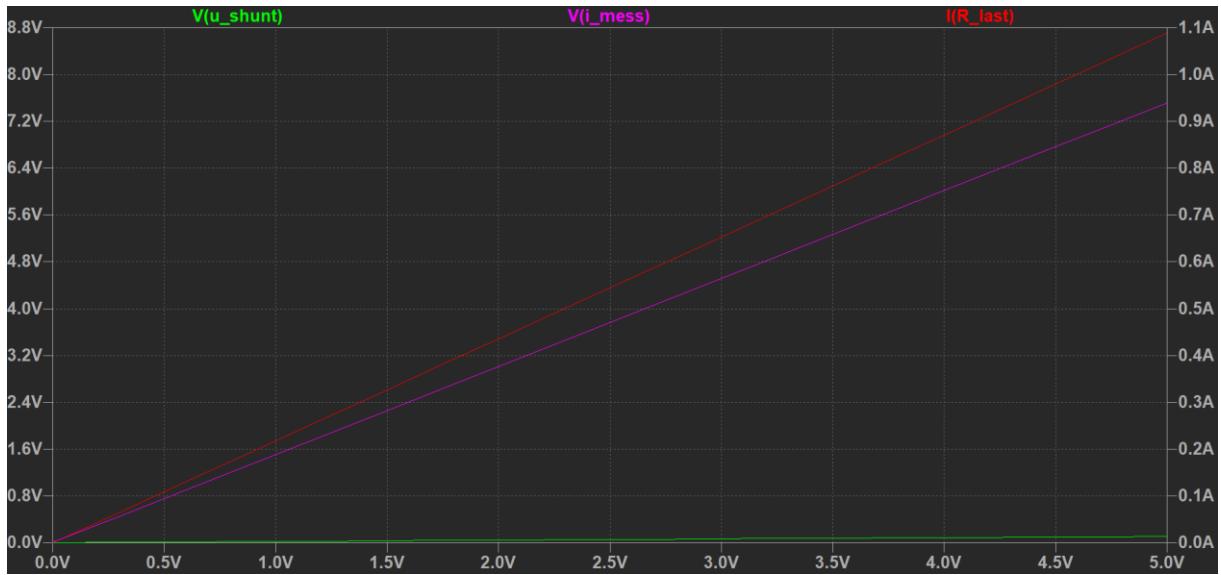


Abbildung 4.18: Strombegrenzung Netzteil positiv



Abbildung 4.19: Strombegrenzung Netzteil negativ

4.4 Das Funktionsgenerator Modul

Zur Realisierung der Teilfunktion „Funktionsgenerator“ des Projektes „OurDAQ“ werden ausgewählte Parameter aus dem Datenblatt des „MyDAQ“ von National Instruments herangezogen. [5]

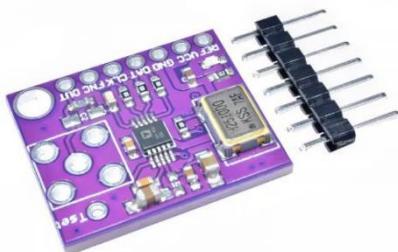
Maximum update rate	200 kS/s
Analog output	+/- 10V, +/- 2V, DC-coupled
Slew rate	4V/ μ s
Maximum output current per line	4mA
Internal base clocks	100MHz

Tabelle 4.6: Ausgewählte Parameter aus dem Datenblatt des MyDAQ von NI [5]

Diese Parameter sind Wunschanforderung, welche weitestgehend realisiert werden sollen, damit der OurDAQ dieselben Ergebnisse wie das MyDAQ nachweisen kann.

Aufgrund der Tatsache, dass im Bereich der Messtechnik Genauigkeit und Präzision erforderlich sind, wurde im Rahmen dieser Projektarbeit festgelegt, dass die Basis bzw. das Herzstück dieser Teifunktion ein „AD9833 Signalgenerator-Modul“ sein soll (vgl. Abbildung 4.20: AD9833 Signalgenerator-Modul). Die eigenständige Entwicklung eines Funktionsgenerators ist möglich, erfordert jedoch relativ viele elektronische Baukomponenten. Mit dieser Art von Entwicklung kommt noch einher, dass bei Manipulation der Frequenz, das Ausgangssignal darunter leidet. Letztendlich ist das eigenständige Entwickeln eines Funktionsgenerators dank fertigen Modulen wie dem „AD9833 Signalgenerator-Modul“ zum einen nicht notwendig und zum anderen auch nicht wirtschaftlich.

4.4.1 Spezifikationen des AD9833 Signalgenerator-Modul



Wie im vorherigen Kapitel angemerkt, wird in diesem Projekt das AD9833 Signalgenerator-Modul (vgl. Abbildung 4.20: AD9833 Signalgenerator-Modul) als Basis zur Teifunktionsentwicklung verwendet.

Abbildung 4.20: AD9833 Signalgenerator-Modul
Dieses programmierbare Modul liefert laut dem Datenblatt des Herstellers „Analog Devices“ ein Signal von 0V bis 0,65V bei maximal 12,5 MHz und das bei einer Betriebsspannung von 2,3V - 5,5V. Die Ansteuerung dieses Moduls wird im Rahmen dieses Projektes durch einen Raspberry Pi 5B durch dessen SPI-Schnittstelle erfolgen. Das Modul kann Sinus-, Dreieck- und Rechtecksignale generieren, jedoch mit dem Nachteil, dass V_{pp} nur für 0,038V-0,65V gilt, sodass nur positive Spannungen erzeugt werden. Ein weiterer Nachteil ist, dass das Rechtecksignal davon ausgeschlossen ist. Die Ausgangsspannung des Rechtecksignals ist genau so groß wie die Eingangsspannung des Moduls.

4.4.2 Konzeptentwicklung zur Anpassung des Spannungsbereiches

Die geringe Ausgangsspannung des Moduls soll für das OurDAQ in Abhängigkeit von der ausgewählten Signalform angepasst werden, sodass die Anforderungen aus (**Fehler! V erweisquelle konnte nicht gefunden werden.**) erfüllt werden können. Je nach Signalform soll

ein Multiplexer „ADG419“ die für den Output relevanten Strompfad wählen. Um aus einer Gleichspannung eine Wechselspannung zu generieren, soll das Modul zusätzlich um einen Differenzverstärker ergänzt werden.

Für den Differenzverstärker soll zum einen das Modul selbst als Eingang dienen und als zweiter Eingang ein Spannungsteiler, dessen Spannungsquelle wiederum der 3,3V Pin vom Raspberry Pi 5B ist. Der Funktionsgenerator soll ein Eingangssignal von $V_{pp}=0,6V$ bereitstellen, welches dann mit der Eingangsspannung des Spannungsteilers von 0,3V vom Differenzverstärker zu einer Ausgangsspannung von $+/- 0,3V$ angepasst werden soll.

Im Anschluss darauf soll das Ausgangssignal des Differenzverstärkers durch eine 2-stufige Verstärkung angepasst werden mit dem Ziel ein Ausgangssignal mit $V_{pp} +/- 10V$ zu erreichen. Der gesamte Verstärkungsfaktor wird in der folgenden Gleichung berechnet.

$$\frac{u_a}{u_e} = \frac{10V}{0.6V} = 16,67 \quad (1.0)$$

4.4.3 Schaltplan des Funktionsgenerator Moduls in LTspice

Bevor das Konzept an realen elektronischen Bauteilen realisiert wird, muss die Schaltung zunächst verifiziert werden. Eine Simulation in LTspice hilft dabei die Funktionalität der Schaltung zu überprüfen und die Widerstände zu dimensionieren.

Es wurde die Simulationsumgebung LTspice verwendet, weil der Herausgeber dieser Software derselbe ist, von welchem die Operationsverstärker hergestellt werden. Um genauer zu sein den AD8479 als Differenzierenden Operationsverstärker und zweimal den LT1352 um die 2-stufige Verstärkung zu realisieren.

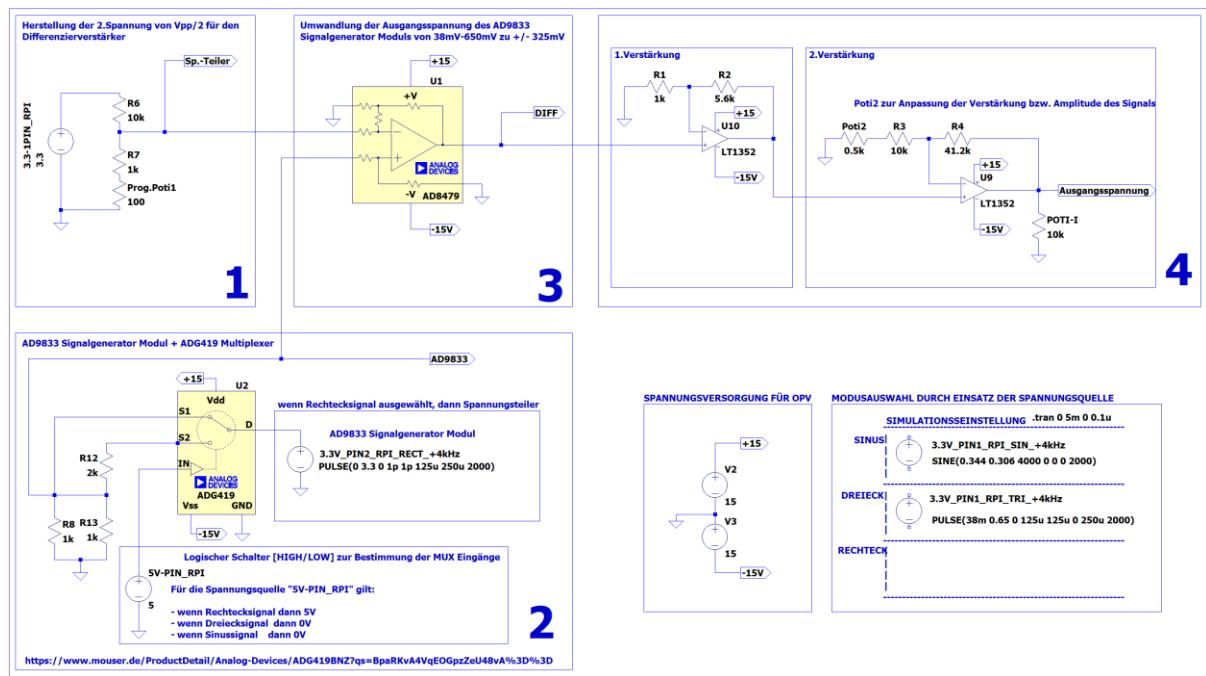


Abbildung 4.21: Schaltplan des Funktionsgenerator-Moduls zur Simulation der Funktionsfähigkeit in LTspice

Um den Schaltplan näher erläutern zu können werden Blockschaltbilder aus Abbildung 4.21: Schaltplan des Funktionsgenerator-Moduls zur Simulation der Funktionsfähigkeit in LTspice erneut vergrößert dargestellt. Die Simulationsergebnisse wiederum werden im Folgekapitel erläutert.

4.4.3.1 Anpassung des Spannungsbereiches durch den AD8479 Differenzverstärker und dem ADG419 Multiplexer

Es wurde bereits erwähnt, dass das Ausgangssignal des AD9833 Signalgenerator Moduls auf +/- 0,3V angepasst werden muss. In der Theorie kann man diese Anpassung mit folgender Schaltung aus Abbildung 4.22: Differenzverstärker realisieren.

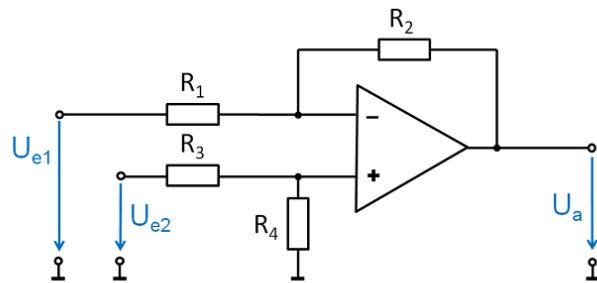


Abbildung 5.25: Subtrahierer

Abbildung 4.22: Differenzverstärker [8]

Das Verhalten des Differenzverstärkers kann durch die folgende Gleichung folgendermaßen beschrieben werden [8]:

$$U_a = U_{E2} \cdot \frac{R_1 + R_2}{R_1} \cdot \frac{R_4}{R_3 + R_4} - U_{E1} \cdot \frac{R_2}{R_1} \quad (2.0)$$

Um dieses Ausgangssignal anzupassen, wird der Differenzverstärker AD8479 von Analog Devices in der Simulation herangezogen. In Abbildung 4.23: Anpassung des Spannungsbereiches mittels AD8479 Differenzverstärker und der Multiplexer ADG419 (Blockschatzbild 1 – 3 des Funktionsgenerator-Moduls) wird dieser im gelben Fenster in Blockschatzbild 3 dargestellt. Die Spannungsquelle „3.3-1Pin_RPI“ mit 3.3V soll den ersten 3,3V-Pin des Raspberry Pis andeuten, von welchem nach einem Spannungsteiler nur 0,3V als Eingangsspannung in den Differenzverstärker eingespeist wird (siehe Blockschatzbild 1). Die zweite Spannungsquelle soll das Ausgangssignal des AD9833 Signalgenerator Moduls darstellen (siehe Blockschatzbild 2). Der Potentiometer „Prog.Poti1“ im 1. Blockschatzbild ist als Feinjustierung vorgesehen, welches das Signal in vertikaler Richtung verschieben soll.

Der Multiplexer ADG419 soll von S1 auf S2 schalten, wenn 5V anliegen. 5V sollen anliegen, wenn die Signalform „Rechteck“ ausgewählt wird. Das bedeutet, dass 5V = HIGH und 0V = LOW bedeutet (Logischer Schalter). Es ist dabei wichtig, dass der Signalgenerator mit einer

Eingangsspannung von 3,3V betrieben wird. Da wie bereits bekannt, die Ausgangsspannung des Rechtecksignals genauso hoch ist, wie die Eingangsspannung.

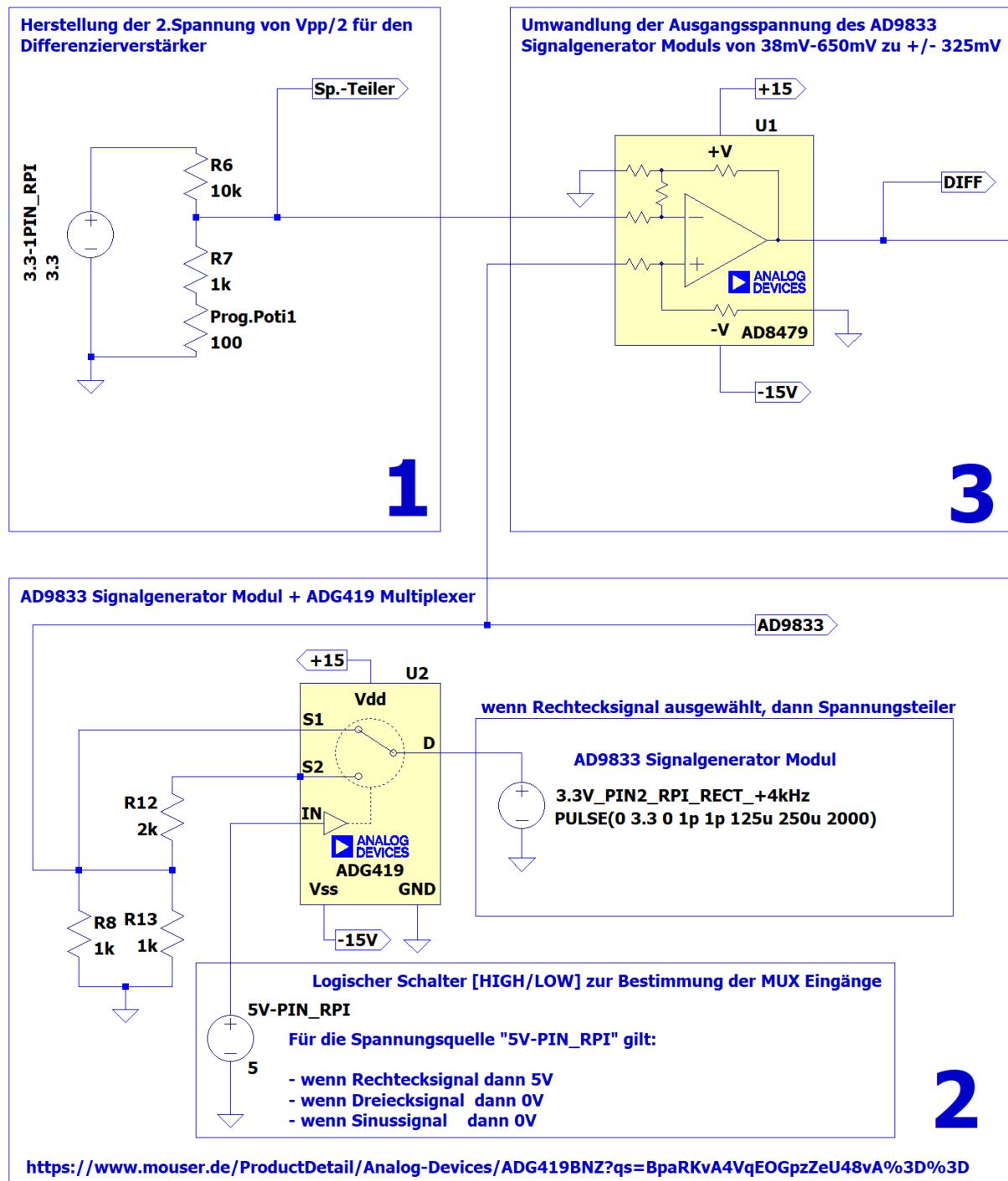


Abbildung 4.23: Anpassung des Spannungsbereiches mittels AD8479 Differenzverstärker und der Multiplexer ADG419
(Blockschaltbild 1 – 3 des Funktionsgenerator-Moduls)

4.4.3.2 Verstärkung des Ausgangssignals des Signalgenerator Moduls

Das 4. Blockschaltbild der Schaltung beschäftigt sich mit der Verstärkung des Ausgangssignals des Differenzverstärkers aus dem 3. Blockschaltbild. Im roten Fenster ist dabei die erste Verstärkung zu sehen, bei welchem das Ausgangssignal des Differenzverstärkers zum ersten Mal auf +/- 2V verstärkt wird. Im grünen Fenster wird dieses Ausgangssignal erneut auf +/- 10V verstärkt (2.Verstärkung).

Das bedeutet, dass nach (Gl.1.0) die erste Stufe das Signal um den Faktor 6,67 und die zweite Stufe das Signal erneut um den Faktor 5 verstärkt wird.

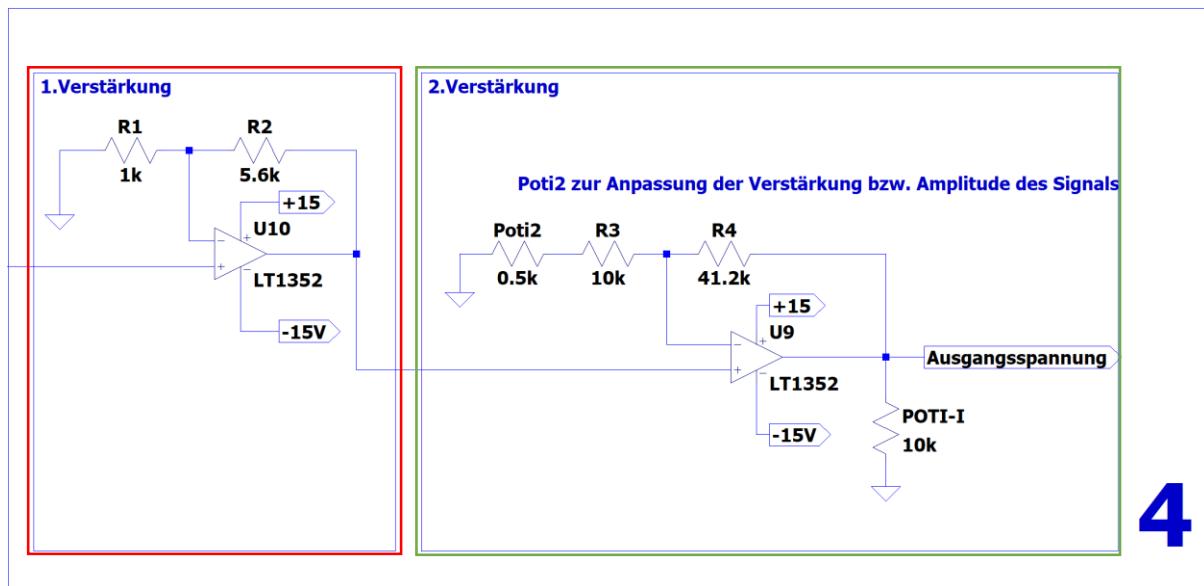


Abbildung 4.24: Schaltplanabschnitt: 2-stufige Verstärker Schaltung (Blockschaltbild 4 der Funktionsgenerator Moduls)

Die Verstärkungen erfolgen mit Hilfe von nicht-invertierenden U/U-Verstärkern LT1352. Diese erfordern als Versorgungsspannung +/- 15V und haben ein Verstärkungs-Bandbreitenprodukt (GBP) von 3MHz sowie eine Anstiegsrate (Slew-rate) von 200V/ μ s. [9]

„Poti2“ soll im Vergleich zu „Prog.Poti1“ aus Blockschaltbild 1 nicht weiterhin das Signal in vertikaler Richtung verschieben, sondern das Ausmaß der Amplitude manipulieren. Damit das Signal optimiert werden kann, müssen beide Potentiometer für den Benutzer am Gehäuse zugänglich gemacht werden.

4.5 Funktionssimulation mit LTspice und Simulationsergebnisse

Die Simulationslaufzeit beträgt 5ms wobei alle $0,01\mu\text{s}$ ein Wert erfasst wird um die unterschiedlichen Signalformen so präzise wie möglich darzustellen. Die Signale werden dabei mit einer Frequenz von 4kHz erzeugt. Betrachtet wird hier das Sinus-, Dreiecks- und Rechtecksignal (vgl. Abbildung 4.25).

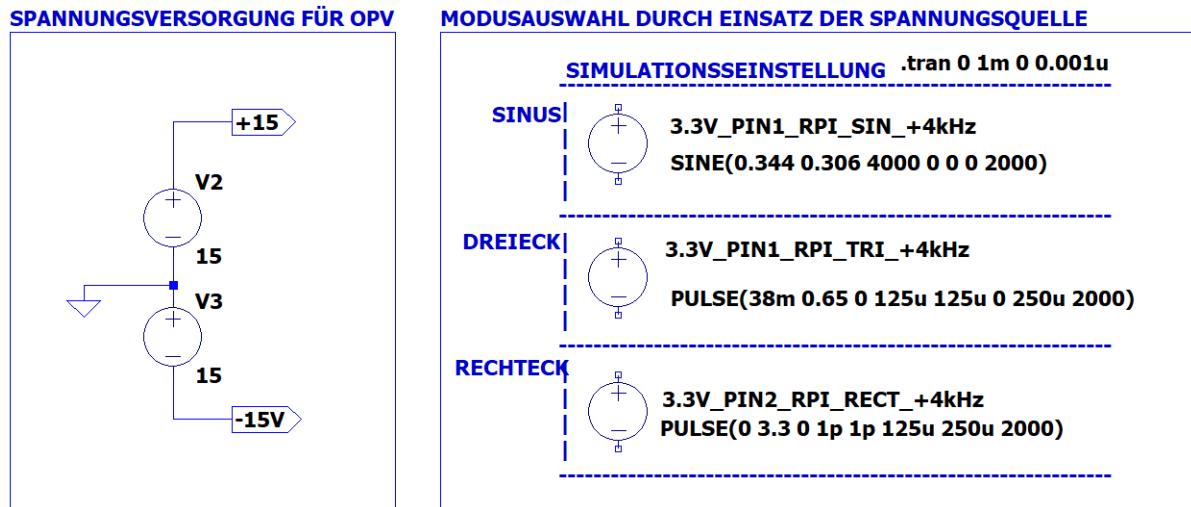


Abbildung 4.25: Ausgangslage der LTspice Simulation mit Darstellung der 3 Signalformen mit den jeweiligen Parametern

4.5.1 Simulationsergebnisse für den AD8479

In diesem Unterkapitel soll die Darstellung erfolgen, wie die unterschiedlichen Signale mittels dem Differenzverstärker ca. zu einer Spannung von +/-0,3V angepasst werden. Wichtig dabei ist, dass bei der Signalform Sinus und Dreieck die Schalter Logik LOW also 0V gewählt wird (siehe Abbildung 4.26).



Abbildung 4.26: Schalterlogik für den Multiplexer ADG419 von Analog Devices

In den folgenden drei Abbildungen ist dabei ersichtlich, dass der Spannungsbereich für jede Signalform relativ gut an den Wunschbereich angepasst werden kann. Wobei die die Spannungsbereiche vom Sinus- und Dreieckssignal unter +/- 0,3 V sind, jedoch der Spannungsbereich des Rechtecksignals über +/- 0,3V. Das liegt daran, dass im Vergleich zur

Ausgangsspannung vom Sinus- und Dreieckssignal, die Ausgangsspannung des Rechtecksignals keinen Offset von 38mV hat, sondern einen Spannungsbereich von 0-3,3V nachweist. **Die Anpassung des Output-Signals nach der zweifachen Verstärkung muss mittels der beiden Potentiometer fein eingestellt werden.**

4.5.1.1 Sinus

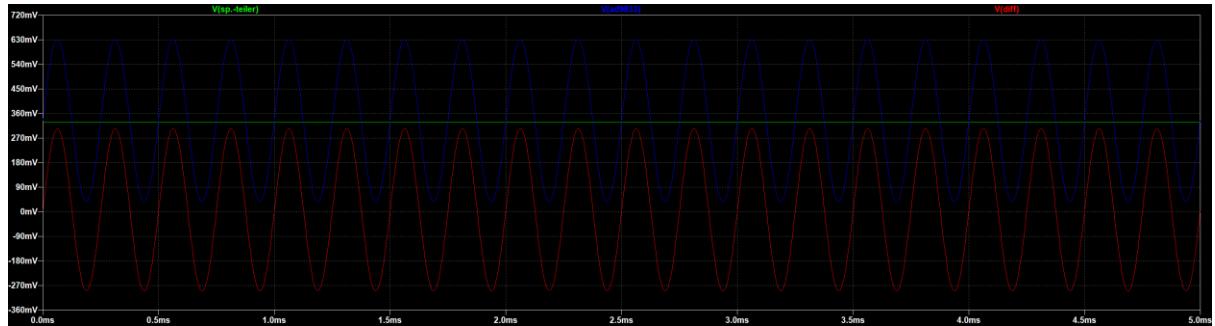


Abbildung 4.27: Offset Sinussignal von AD8479

4.5.1.2 Dreieck

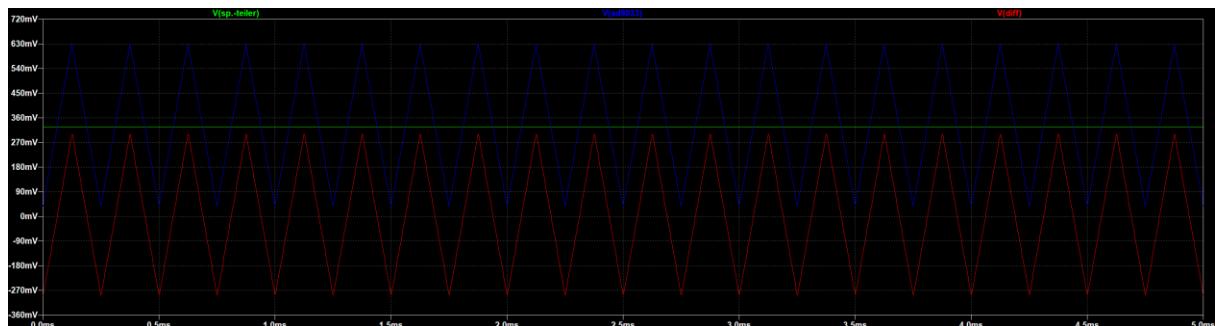


Abbildung 4.28: Offset Dreieckssignal von AD8479

4.5.1.3 Rechteck

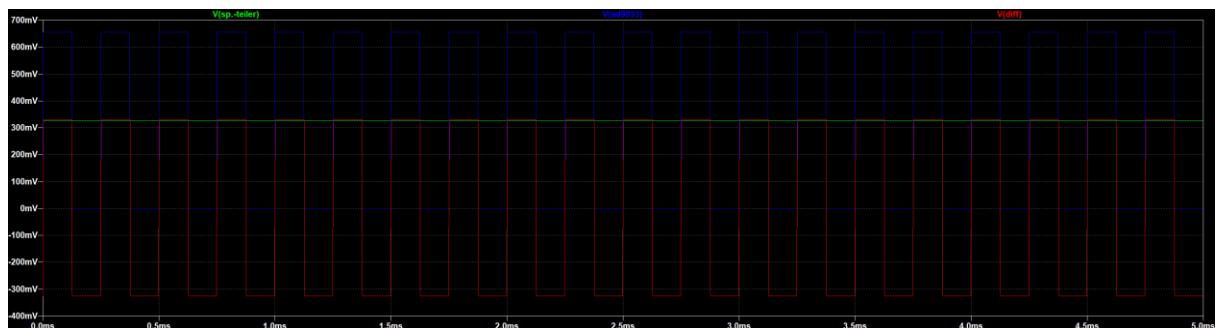


Abbildung 4.29: Offset Rechteckssignal von AD8479 (Im Vergleich größer als bei den Signalformen Sinus und Dreieck)

4.5.2 Simulationsergebnisse für den LT1352

In diesem Kapitel wird die Verstärkung des Differenzsignals als Eingangssignal verdeutlicht, welches durch eine 2-stufige Verstärkung auf +/- 10V angepasst werden soll. Zusätzlich wurden Messungen an den Amplituden.

4.5.2.1 Sinus

In Abbildung 4.30: Eingangssignal Sinus und dessen 2-stufige Verstärkung ist erkennlich, dass das Signal erfolgreich verstärkt worden ist.

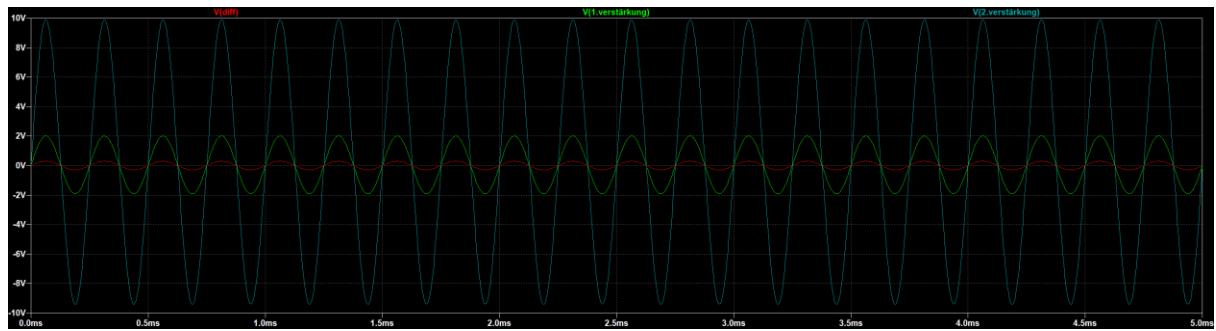


Abbildung 4.30: Eingangssignal Sinus und dessen 2-stufige Verstärkung

In Abbildung 4.31: Messung des Sinussignals ist ersichtlich, dass das Sinussignal nahezu ein $V_{pp} = 20V$ (genau: $V_{pp} = 19,287V$ vgl. Abbildung 4.32: Quantitative Messung des Sinussignals (Diff: Cursor2 - Cursor1)) darstellt. Diese Abweichung kann eliminiert werden, wenn eine Offsetkompensation erfolgen soll mittels der Potentiometer Einstellungen.

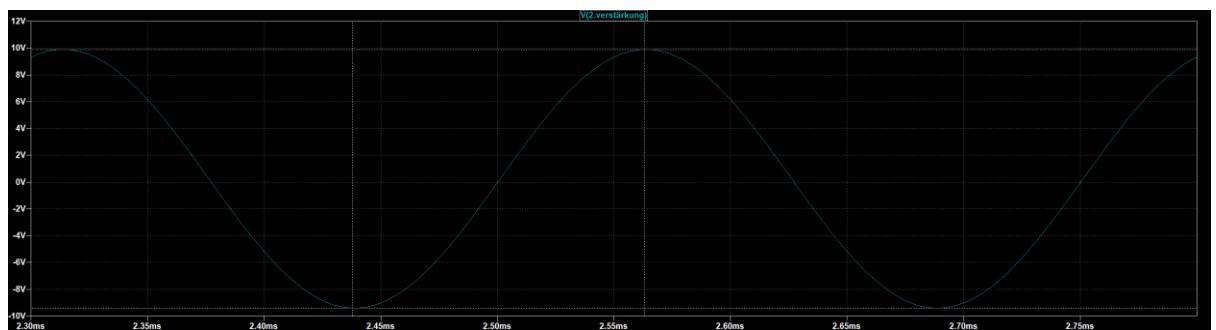


Abbildung 4.31: Messung des Sinussignals

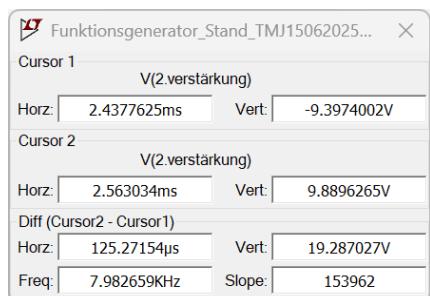


Abbildung 4.32: Quantitative Messung des Sinussignals (Diff: Cursor2 - Cursor1)

4.5.2.2 Dreieck

In Abbildung 4.33: Eingangssignal Dreieck und dessen 2-stufige Verstärkung ist erkenntlich, dass das Signal erfolgreich verstärkt worden ist.

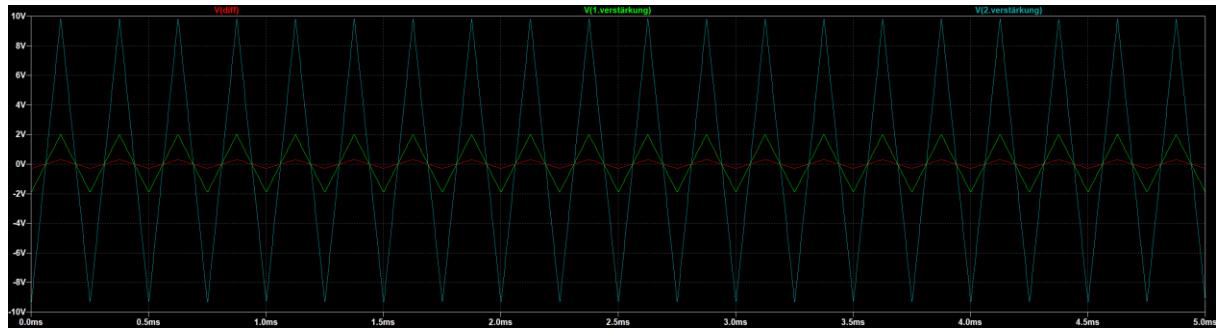


Abbildung 4.33: Eingangssignal Dreieck und dessen 2-stufige Verstärkung

In Abbildung 4.34: Messung des Dreiecksignals ist ersichtlich, dass das Dreiecksignal knapp unter $V_{pp} = 20V$ darstellt (genau: $V_{pp} = 19,156V$ vgl. Abbildung 4.35: Quantitative Messung des Dreieckssignals (Diff: Cursor2 - Cursor1)). Die Differenz sowie die Amplituden sind minimal größer im Vergleich zum Sinus- und Rechtecksignal. Diese Abweichung kann eliminiert werden, wenn eine Offsetkompensation erfolgen soll mittels der Potentiometer Einstellungen.

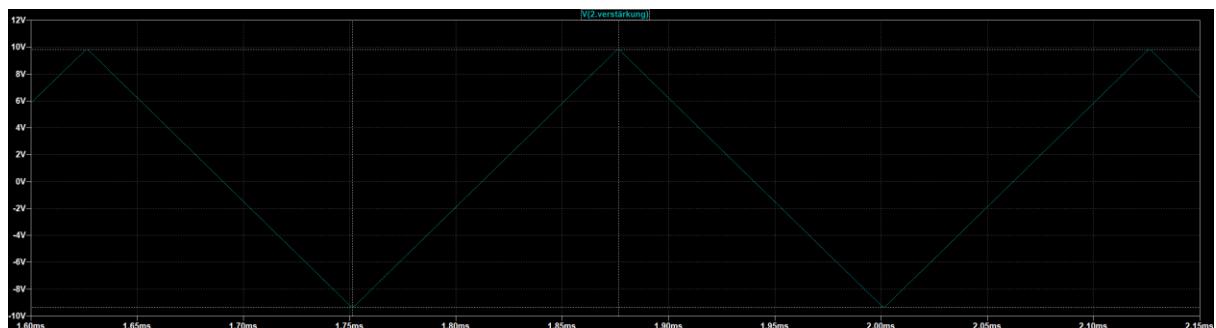


Abbildung 4.34: Messung des Dreiecksignals

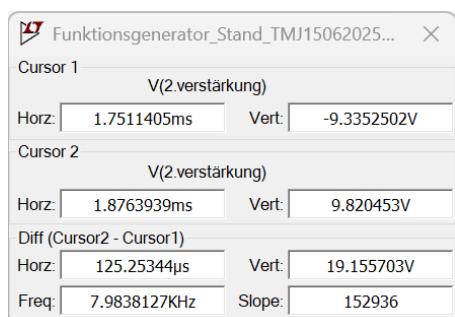


Abbildung 4.35: Quantitative Messung des Dreieckssignals (Diff: Cursor2 - Cursor1)

4.5.2.3 Rechteck

In Abbildung 4.36: Eingangssignal Rechteck und dessen 2-stufige Verstärkung ist erkenntlich, dass das Signal erfolgreich verstärkt worden ist.

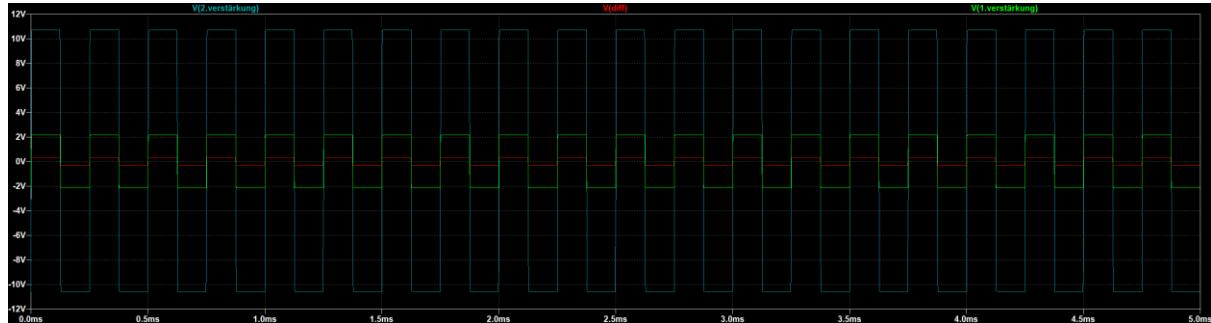


Abbildung 4.36: Eingangssignal Rechteck und dessen 2-stufige Verstärkung

In Abbildung 4.37: 1. Messung des Rechtecksignals zur Bestimmung der Amplituden ist ersichtlich, dass das Rechtecksignal größer als ein $V_{pp} = 20V$ darstellt (genau: $V_{pp} = 21,307V$ vgl. Abbildung 4.38: Quantitative Messung des Rechtecksignals (Diff: Cursor2 - Cursor1)). Die Differenz sowie die Amplituden sind minimal größer im Vergleich zum Sinussignal. Diese Abweichung kann eliminiert werden, wenn eine Offsetkompensation erfolgen soll mittels der Potentiometer Einstellungen.

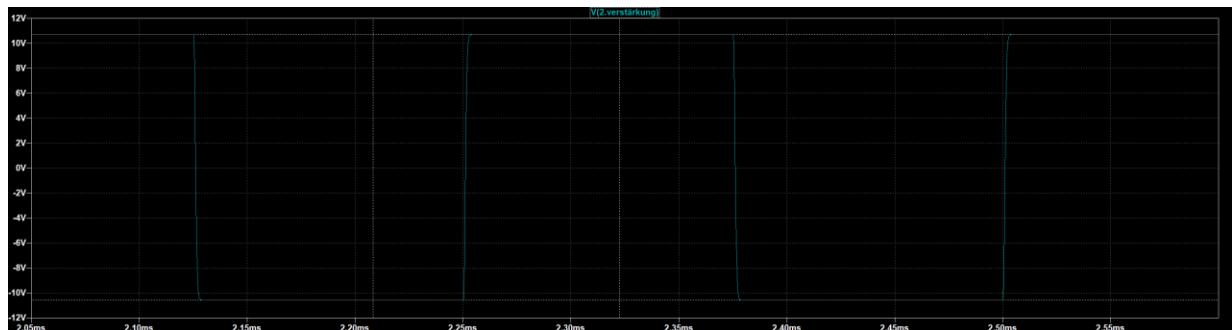


Abbildung 4.37: 1. Messung des Rechtecksignals zur Bestimmung der Amplituden

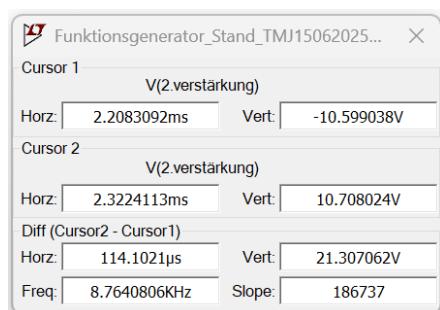


Abbildung 4.38: Quantitative Messung des Rechtecksignals (Diff: Cursor2 - Cursor1)

4.5.3 Anmerkung zum Funktionsgeneratormodul

Das Funktionsgenerator Modul wurde nicht nur in der Theorie, sondern auch in der Praxis erstellt und getestet. Die Ergebnisse für die jeweiligen Signalformen waren weitestgehend ähnlich zu den Ergebnissen aus den LTspice Simulationen.

Die Realisierung sowie die Verifikation des Moduls hat auf einem konventionellen Breadboard stattgefunden (vgl. **Fehler! Verweisquelle konnte nicht gefunden werden.**).

Auf Grund eines Fehlers sind beim An- und Abstecken von Jump-Wire-Kabeln ist dem Projekt-Team leider der OPV LT1352 abgeraucht. Auf Grund dieses Fehlers ist es dem Projekt-Team nicht mehr möglich, die realen Ergebnisse des Funktionsgenerators nachzuweisen.

Es wird jedoch versichert, dass wenn der Schaltplan des Funktionsgenerators strickt, verfolgt wird und nichts umgesteckt wird, dass dann der Funktionsgenerator funktionieren wird. Das Projekt-Team würde sich freuen, wenn es bei Fragen die nachfolgenden Projekt-Teams bei dieser Angelegenheit unterstützen dürfte.

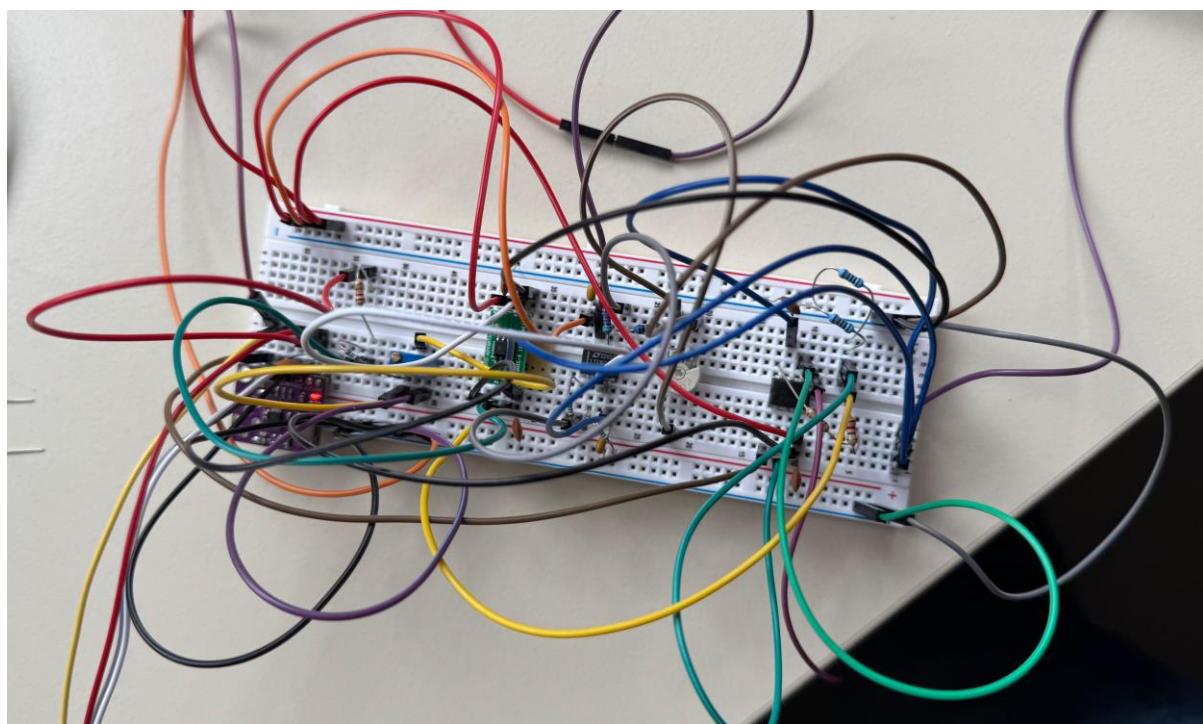


Abbildung 4.39: Prototypischer Aufbau des Funktionsgenerator Moduls

4.6 Diodenkennlinie

Für die Aufnahme der Diodenkennlinie wurde festgelegt, dass die Messschaltung (bestehend aus der Diode und einem Serienwiderstand) extern, beispielsweise auf einem Steckbrett, aufgebaut wird. Die Messobjekte sollen dabei über das interne Netzteil des OurDAQ mit der erforderlichen Spannung versorgt werden.

Zur Erfassung der relevanten Messgrößen kommen die Kanäle 6 und 7 des MCC 118 zum Einsatz. Kanal 6 misst die Gesamtspannung der Schaltung, während Kanal 7 die Spannung direkt über der Diode erfasst. Aus der gemessenen Diodenspannung und dem berechneten Strom durch den Serienwiderstand kann anschließend die Kennlinie der Diode grafisch dargestellt werden. [30]

Eine separate Simulation der Messschaltung ist hierbei nicht erforderlich, die Messungen können direkt mit dem HAT-Modul durchgeführt werden.

4.6.1 Messgenauigkeit und Fehlerabschätzung

Die Genauigkeit wird im Wesentlichen durch

1. die Spannungsmessung des MCC 118 und
2. die Toleranz des Serienwiderstands R bestimmt. Verwendet werden sollte ein Präzisionswiderstand mit 0,1 % bis 0,5 % Toleranz.

Die lineare (maximale) Fehlerfortpflanzung ist konservativ, dabei nimmt man die Beträge der partiellen Ableitungen und multipliziert diese mit den Fehlergrenzen: [31]

$$\Delta y \approx \sum_i \left| \frac{\partial f}{\partial x_i} \right| * \Delta x_i$$

Der Strom berechnet sich nach:

$$I = \frac{U_{ges} - U_D}{R}$$

Ableitungen:

$$\frac{\partial I}{\partial U_{ges}} = \frac{1}{R}$$

$$\frac{\partial I}{\partial U_D} = -\frac{1}{R}$$

$$\frac{\partial I}{\partial R} = -\frac{U_{ges} - U_D}{R^2}$$

einsetzen in lineare Abschätzung:

$$\Delta I \approx \left| \frac{\partial I}{\partial U_{ges}} \right| * \Delta U_{ges} + \left| \frac{\partial I}{\partial U_D} \right| * \Delta U_D + \left| \frac{\partial I}{\partial R} \right| * \Delta R = \frac{\Delta U_{ges}}{R} + \frac{\Delta U_D}{R} + \frac{U_{ges} - U_D}{R^2} * \Delta R$$

Für die aus zwei Spannungen und einem Widerstand berechnete Stromstärke ergibt sich (maximaler Fehler):

$$\Delta I \approx \frac{|\Delta U_{ges}| + |\Delta U_D|}{R} + \frac{|U_{ges} - U_D|}{R} * \frac{\Delta R}{R}$$

ΔU_{ges} , ΔU_D ... Messfehler der beiden Spannungen durch MCC 118,
 ΔR ... Widerstandstoleranz (z.B. 0,1% * R)

Für das MCC 118 (DC) kann ein einzelner Spannungsfehler grob abgeschätzt werden aus Gain-Fehler und Offset (Worst Case):

$$\Delta U \approx 0,00098 * U + 11 \text{ mV}$$

Beispielrechnung

- Gesamtspannung: $U_{ges} = 0,750 \text{ V}$
- Spannung über Widerstand: $U_R = 0,050 \text{ V}$
- Serienwiderstand $R = 100 \Omega$ (0,1% Toleranz)

$$I_{nominal} = \frac{U_R}{R} = \frac{0,050 \text{ V}}{100 \Omega} = 0,5 \text{ mA}$$

Spannungsfehler über Widerstand:

$$\Delta U_R = 0,00098 * U_R + 11 \text{ mV} = 0,00098 * 0,050 \text{ V} + 11 \text{ mV} \approx 11 \text{ mV}$$

Gesamter Spannungsfehler:

$$U_{ges} = 0,750 \text{ V} \rightarrow \Delta U_{ges} = 0,00098 * 0,750 \text{ V} + 0,011 \text{ V} \approx 11,735 \text{ mV}$$

$$\begin{aligned} U_D &= U_{ges} - U_R = 0,750 \text{ V} - 0,050 \text{ V} = 0,700 \text{ V} \rightarrow \Delta U_D \\ &= 0,00098 * 0,700 \text{ V} + 0,011 \text{ V} \approx 11,686 \text{ mV} \end{aligned}$$

$$\Delta U_{ges} + \Delta U_D \approx 11,735 \text{ mV} + 11,686 \text{ mV} \approx 23,421 \text{ mV}$$

Umrechnung auf Stromfehler:

$$\Delta I_{HAT} = \frac{|\Delta U_{ges}| + |\Delta U_D|}{R} = \frac{23,421 \text{ mV}}{100 \Omega} \approx 0,234 \text{ mA}$$

Fehler durch Widerstand:

$$\Delta I_R = I * \frac{\Delta R}{R} = 0,5 \text{ mA} * 0,001 = 0,0005 \text{ mA} \approx 0 \text{ mA}$$

Worst-Case Stromfehler:

$$\Delta I_{WC} = \Delta I_{HAT} + \Delta I_R \approx 0,234 \text{ mA} + 0 \text{ mA} = 0,234 \text{ mA}$$

Relativer Fehler:

$$\text{relativer Fehler} = \frac{\Delta I_{WC}}{\Delta I_{nominal}} * 100 \% = \frac{0,234 \text{ mA}}{0,5 \text{ mA}} * 100 \% \approx 46,8 \%$$

Dadurch können für einen Serienwiderstand mit 0,1 % die in Tabelle 4.7 dargestellten maximalen Toleranzangaben angegeben werden.

I _{nominal} [mA]	U _{ges} [V]	U _R [V]	ΔI _{HAT} [mA]	ΔI _R [mA]	ΔI _{WC} [mA]	relativer Fehler [%]
0,5	0,750	0,050	0,234	0,000	0,234	46,8
1,0	0,800	0,100	0,235	0,001	0,236	23,6
5,0	1,200	0,500	0,239	0,005	0,244	4,9
10,0	1,700	1,000	0,244	0,010	0,254	2,5
20,0	2,700	2,000	0,253	0,020	0,273	1,4
50,0	5,700	5,000	0,283	0,050	0,333	0,7

Tabelle 4.7: Diodenkennlinie - Genauigkeit mit 0,1% Serienwiderstand

Tabelle 4.8 zeigt die maximalen Toleranzangaben für einen Serienwiderstand mit 1,0 %.

I _{nominal} [mA]	U _{ges} [V]	U _R [V]	ΔI _{HAT} [mA]	ΔI _R [mA]	ΔI _{WC} [mA]	relativer Fehler [%]
0,5	0,750	0,050	0,234	0,005	0,239	47,8
1,0	0,800	0,100	0,235	0,010	0,245	24,5
5,0	1,200	0,500	0,239	0,050	0,289	5,8
10,0	1,700	1,000	0,244	0,100	0,344	3,4
20,0	2,700	2,000	0,253	0,200	0,453	2,3
50,0	5,700	5,000	0,283	0,500	0,783	1,6

Tabelle 4.8: Diodenkennlinie - Genauigkeit mit 1,0% Serienwiderstand

Legende:

- ΔI_{HAT} = Fehler durch Spannungsmessung MCC 118
- ΔI_R = Fehler durch Widerstandstoleranz
- ΔI_{WC} = Worst-Case-Summe
- Relativer Fehler = $\Delta I_{WC} / I_{nominal}$

Praxisempfehlungen:

- Präzisionswiderstand (0,1 % - 0,5 %) einsetzen
- Messbereich so wählen, dass $U_{ges} - U_D$ nicht zu klein wird (besseres SNR).
- Beide Spannungen aus demselben HAT und zeitnah erfassen (gleicher Gain/Offset, reduzierter Drift).

Die zuvor gezeigten Toleranzen und Fehlerwerte stellen **Maximalwerte** dar.

Insbesondere bei kleinen Strömen im mA-Bereich wirkt sich der **relative Fehler** stärker aus, da Offset und Rauschen im Verhältnis zum Strom größer sind.

Wie in Kapitel 7.5 zu sehen ist, tritt der **Worst-Case-Fehler** jedoch nicht auf, und die Messungen zeigen eine sehr gute Genauigkeit.

4.7 Filterkennlinie

Für die Messung der Filterkennlinie wird ein externer Funktionsgenerator verwendet, der das Eingangssignal bereitstellt. Als Testobjekt dient beispielweise ein einfacher Tiefpass aus einem Widerstand R und einem Kondensator C. [32]

Zur Erfassung der Signale werden die Kanäle 6 und 7 des MCC 118 genutzt. Kanal 6 misst die Eingangsspannung des Filters, während Kanal 7 die Ausgangsspannung direkt nach dem Filter erfasst.

Aus den gemessenen Spannungen kann die Frequenzgangkurve des Tiefpasses berechnet und grafisch dargestellt werden. Die Darstellung erfolgt dabei sowohl logarithmisch in dB für die Amplitude und auch der Phasenverlauf des Filters kann anschaulich visualisiert werden.

Da für die Filterkennlinienmessung ausschließlich Standardkomponenten und das vorhandene HAT-Modul verwendet werden, kann auf eine vorherige Simulation verzichtet werden und die Auswertung direkt in der realen Messumgebung erfolgen.

Samplerate und Frequenz

Nyquist:

$$f_{max} \approx \frac{\text{Samplerate}}{2} = 50 \text{ kHz}$$

- Die Sinusform muss dabei nicht perfekt wiedergegeben werden, da für die Berechnung der Amplitude, z. B. für RMS- oder Peak-Werte, die diskreten Abtastwerte ausreichen
- Praktisch sind Messungen bis ca. 30 kHz möglich

Fehlerquellen

Amplitude

- MCC 118 Offset-Fehler: $\pm 11 \text{ mV}$
- Gain-Fehler $\approx 0,098\%$ (kürzt sich bei Verhältnismessung fast raus)

$$U_{mess} = U_{out} + U_{offset}$$

$$A_{dB} = 20 * \log_{10} \left(\frac{U_{mess}}{U_{out}} \right) = 20 * \log_{10} \left(1 + \frac{U_{offset}}{U_{out}} \right)$$

Für die unterschiedlichen Amplituden am Eingang ergeben sich bei 0 dB, -20 dB und -40 dB folgende Werte (vgl. Tabelle 4.9).

$U_{ss} [\text{V}]$	$U_s [\text{V}]$	0 dB ($U_{out} = U_s$)	-20 dB ($U_{out} = 0,1 * U_s$)	-40 dB ($U_{out} = 0,01 * U_s$)
1,0	0,5	0,189 dB	1,727 dB	10,103 dB
2,0	1,0	0,095 dB	0,906 dB	6,444 dB
3,0	1,5	0,063 dB	0,615 dB	4,778 dB
4,0	2,0	0,048 dB	0,465 dB	3,807 dB
5,0	2,5	0,038 dB	0,374 dB	3,167 dB
6,0	3,0	0,032 dB	0,313 dB	2,713 dB
7,0	3,5	0,027 dB	0,269 dB	2,374 dB
8,0	4,0	0,024 dB	0,236 dB	2,110 dB
9,0	4,5	0,021 dB	0,210 dB	1,900 dB
10,0	5,0	0,019 dB	0,189 dB	1,727 dB

Tabelle 4.9: Filterkennlinie - Genauigkeit bei unterschiedlichen Amplituden

- In Durchlass (0 dB) ist der Offset-Einfluss sehr klein: z. B. ab $4 U_{ss} \approx 0,048 \text{ dB}$
- Bei -20 dB wird der Offset relevant, ab $2 U_{ss}$ ist der Fehler $< 1 \text{ dB}$
- Bei -40 dB wird der Fehler groß für kleine Signale (z. B. $1 U_{ss} \approx 10 \text{ dB}$). Bei höheren Pegeln ($4-10 U_{ss}$) ist der Fehler in der Spanne $\approx 1,7-3,8 \text{ dB}$

Phase [21]

Die Phase wird durch die Zeitdifferenz Δt zwischen Ein- und Ausgangssignal bestimmt:

$$\Delta\phi_{max} = 360^\circ * f * \Delta t$$

Da die Kanäle gemultiplexed werden, beträgt der Zeitversatz Δt zwischen den Kanal-Samples bei einer Abtastrate von 100 kS/s etwa 8 μs . [8]

In der Tabelle 4.10 ist für verschiedene Frequenzen der maximale Phasendifferenz-Fehler zwischen den Kanälen dargestellt.

f [kHz]	$\Delta\phi_{max}$
1,0	2,88°
2,0	5,76°
5,0	14,4°
10	28,8°
20	57,6°
40	115,2°

Tabelle 4.10: Filterkennlinie - maximale Phasendifferenz

Dieser Phasenbeitrag ist hauptsächlich systematisch und kann/ sollte kompensiert werden:

- Entweder die Zeitreihen softwaretechnisch zeitlich um Δt gegeneinander verschieben (Interpolation)
- oder beim Frequenzgang die Phase um $\Delta\phi_{max} = 360^\circ * f * \Delta t$ subtrahieren

Somit würde nur noch eine Restunsicherheit übrigbleiben, welche vom Jitter der Taktquelle und dem Quantisierungsrauschen abhängt.

Somit ist zu empfehlen Spannungen $U_{ss} \geq 4$ V zu nutzen, damit der Offset-Einfluss gering ist. Außerdem sollte unbedingt eine Phasendifferenz-Korrektur durchgeführt werden.

5 Softwareentwicklung

5.1 Einführung in die Software

Die Software steuert Hardware-Komponenten wie den MCC 118 (8-Kanal-ADC), den AD9833 (Funktionsgenerator) sowie den DAC (Netzteil) und ermöglicht die Analyse und Visualisierung von Daten. Die Kernfunktionen umfassen:

- **Multimeter**: Spannungs- und Strommessung
- **Funktionsgenerator**: Erzeugung von Sinus-, Dreieck- und Rechteckwellen
- **Oszilloskop**: Echtzeitmessung mehrerer Kanäle
- **Netzteil**: regelbare Spannungsversorgung mit Überstromschutz
- **Kennlinienmessung**: Analyse von Dioden- und Filterkennlinien
- Der Software-Stack basiert auf Python und umfasst drei Hauptkomponenten:
 - **Lokale Skripte**: direkte Hardware-Steuerung
 - **Jupyter Notebooks**: interaktive Messungen
 - **Dashboard-Web-Oberflächen**: Remote-Zugriff über Browser

Die Software ist reproduzierbar, modular aufgebaut und erweiterbar. Über das zentrale Dashboard **Dashboard_web.py** werden alle Module integriert und ein Remote-Zugriff über ein Netzwerk ermöglicht. Dies ist ideal für Labore oder Fernexperimente. Bei der Erstellung und Optimierung des Programmcodes wurde unterstützend auf KI-basierte Werkzeuge wie ChatGPT, Claude und Gemini zurückgegriffen, um die Effizienz der Entwicklung zu steigern und komplexe Algorithmen zu implementieren. Die folgende Tabelle 5.1 gibt einen kleinen Überblick:

Komponente	Zweck	Beispiel-Datei
Lokale Skripte	Hardware-Steuerung	DMM.py
Jupyter Notebooks	Interaktive Analysen	Filterkennlinie.ipynb
Web-Apps (Dash)	Remote-Zugriff und Visualisierung	Dashboard_web.py

Tabelle 5.1: Softwarekomponente

Die Entwicklung begann mit Python-Skripten, die direkt auf dem Raspberry Pi ausgeführt wurden. Jupyter Notebooks wurden hinzugefügt, um komplexe Analysen wie die Diodenkennlinie zu ermöglichen. Schließlich wurde eine Hauptweboberfläche eingeführt, um die Funktionalität über ein Netzwerk zugänglich zu machen und einen direkten Zugriff auf alle Funktionen zu ermöglichen.

Anfangs wurde der Raspberry Pi 3 verwendet, da er kostengünstig ist und über GPIO-Pins für Hardware-Interfacing verfügt. Aufgrund der begrenzten Rechenleistung (Quad-Core 1,2 GHz, 1 GB RAM) kam es jedoch zu Einschränkungen:

- **Multikanal-Messungen:** Instabile Echtzeit-Datenerfassung beim Oszilloskop
- **Web-Performance:** Langsame Rendering-Zeiten für Dash-Apps
- **GPIO-Steuerung:** Probleme mit der **RPi.GPIO**-Bibliothek bei parallelen Prozessen

Diese Probleme konnten durch den Wechsel zum Raspberry Pi 5 (Quad-Core 2,4 GHz, bis zu 8 GB RAM) gelöst werden. Die verbesserte GPIO-Unterstützung mit Igpio sowie die höhere CPU-Leistung ermöglichen stabile Multikanal-Messungen und parallele Web-Prozesse. Die folgende Tabelle 5.2 vergleicht beide Modelle:

Attribut	Raspberry Pi 3	Raspberry Pi 5
CPU	Quad-Core 1.2 GHz	Quad-Core 2.4 GHz
RAM	1 GB	4–8 GB
GPIO	40 Pins, RPi.GPIO	40 Pins, Iglio
Performance	begrenzt für Web	hoch für Echtzeit

Tabelle 5.2: Vergleich Raspberry Pi 3 und Raspberry Pi

5.2 Software-Architektur

5.2.1 Auswahl der Programmiersprache: Warum Python?

Python wurde aus folgenden Gründen als primäre Programmiersprache für dieses Projekt gewählt:

- **Breite Unterstützung für Hardware-Interfacing:** Python bietet gut dokumentierte Bibliotheken wie **daqhats** für den MCC 118-ADC, **spidev** für die SPI-Kommunikation und **Iglio** für GPIO-Steuerung auf dem Raspberry Pi. Diese Bibliotheken sind speziell für den Raspberry Pi optimiert und ermöglichen eine einfache Integration mit der Hardware.
- **Umfangreiches Ökosystem für Datenanalyse:** Bibliotheken wie **numpy**, **scipy** und **matplotlib** sind ideal für die Analyse und Visualisierung von Messdaten. Python ist in der wissenschaftlichen Community weit verbreitet, was die Wiederverwendbarkeit von Code erleichtert.
- **Web-Entwicklung:** Mit Dash bietet Python eine leistungsstarke Lösung für interaktive Web-Oberflächen, die für den Remote-Zugriff auf das Datenerfassungsmodul von entscheidender Bedeutung sind. Im Vergleich zu Alternativen wie JavaScript oder PHP

ermöglicht Python eine einheitliche Codebasis für Hardware-Steuerung, Analyse und Web-Apps.

- **Einfachheit und Lesbarkeit:** Die klare Syntax von Python reduziert die Einarbeitungszeit und erleichtert die Wartung. Dies war besonders wichtig für die Entwicklung in einem akademischen Kontext in dem die Lesbarkeit für Mitentwickler oder Nachnutzer entscheidend ist.
- **Gemeinschaft und Ressourcen:** Python verfügt über eine große Community und zahlreiche Tutorials, z.B. für Raspberry Pi und MCC 118. Das war sehr hilfreich bei der Fehlersuche.

Alternativen und deren Nachteile:

- **C/C++:** Bietet zwar eine höhere Performance, erfordert aber einen komplexeren Code für den Hardware-Zugriff und die Web-Integration. Die Entwicklungsdauer wäre länger gewesen.
- **Java:** Weniger geeignet für Raspberry Pi GPIO und wissenschaftliche Analysen, außerdem schwerfälliger als Python.
- **MATLAB:** Proprietär, teuer und nicht für Web-Oberflächen optimiert. Jupyter Notebooks bieten ähnliche Funktionalität kostenlos.

Die folgende Tabelle 5.3 vergleicht verschiedene Programmiersprachen:

Programmier-sprache	Vorteil	Nachteil	Warum nicht gewählt?
Python	breites Ökosystem, einfache Syntax	weniger performant als C++	Gewählt für Flexibilität und Community
C/C++	Hohe Performance	Komplexe Entwicklung	Zu aufwendig für schnelle Prototypen
Java	Plattformunabhängigkeit	Weniger GPIO-Unterstützung	Nicht ideal für Raspberry Pi
MATLAB	Starke Analyse-Tools	Proprietär, keine Web-Integration	Kosten und eingeschränkte Flexibilität

Tabelle 5.3: Vergleich Programmiersprachen

5.2.2 Verwendete Bibliotheken und Tools

Der Software-Stack nutzt spezialisierte Python-Bibliotheken, die für Hardware-Zugriff, Datenanalyse und Web-Entwicklung optimiert sind. Die wichtigsten Bibliotheken werden in folgende Tabelle 5.4 gezeigt:

Bibliothek	Zweck	Beispiel-Code	Verwendung
daqhats	MCC118-Steuerung	mcc118(address).a_in_read(channel)	Multimeter, Oszilloskop, Kennlinien
spidev	SPI-Kommunikation (z.B. AD9833)	spi.open(0, 0)	Funktionsgenerator
Igpio	GPIO-Steuerung auf Raspi 5	Igpio.gpio_claim_output(handle, pin)	Funktionsgenerator, Netzteil
Dash	Web-Oberflächen	@callback(Output('id'), Input('id'))	Dashboard, Multimeter, Oszilloskop, etc.
Jupyter	Interaktive Notebooks	from IPython.display import display	Diodenkennlinie, Filterkennlinie
numpy	Numerische Berechnungen	np.linspace(0, 5, 50)	Diodenkennlinie, Filterkennlinie
matplotlib	Visualisierung	plt.semilogx(frequencies, amplitudes)	Diodenkennlinie, Filterkennlinie
scipy.optimize	Kurvenanpassung	scipy.optimize.curve_fit(shockley, ...)	Diodenkennlinie
ad9833	AD9833-Funktionsgenerator-Steuerung	ad9833.set_frequency(freq, 'SINE')	Funktionsgenerator, Filterkennlinie
plotly.graph_objects	Interaktive Web-Visualisierung	go.Scatter(x=data, y=values)	Oszilloskop

Tabelle 5.4: Verwendete Bibliotheken

Ein typischer Import-Block aus **Oszilloskop_web.py** zeigt die Kombination dieser Bibliotheken:

```
import socket
import json
from time import sleep
from collections import deque
from dash import Dash, dcc, html, Input, Output, State, callback
import plotly.graph_objects as go
```

5.2.3 Warum Jupyter Notebooks?

Jupyter Notebooks wurden für die interaktiven Analyse- und Visualisierungskomponenten aus folgenden Gründen gewählt:

- **Interaktivität:** Notebooks ermöglichen Benutzereingaben (z.B. Spannungsbereich) und sofortige Visualisierung (z.B. Matplotlib-Plots für Diodenkennlinien). Dies ist ideal für experimentelle Anpassungen, z.B. bei der Aufnahme einer Diodenkennlinie.
- **Kombination von Code und Dokumentation:** Code, Erklärungen und Plots können in einer Datei kombiniert werden, was die Nachvollziehbarkeit erhöht.
- **Open Source und plattformunabhängig:** Im Gegensatz zu proprietären Alternativen wie MATLAB ist Jupyter kostenlos und lässt sich leicht auf dem Raspberry Pi installieren.
- **Wissenschaftliche Community:** Jupyter ist in der akademischen Welt Standard für Datenanalyse, was die Wiederverwendung von Code.
- **Integration mit Python:** Nahtlose Nutzung von **numpy**, **matplotlib** und **scipy**, die für die Analyse von Kennlinien essenziell sind.

Alternativen und deren Nachteile:

- **Standard-Python-Skripte:** Fehlen der Interaktivität und integrierten Visualisierung; ungeeignet für iterative Experimente
- **MATLAB:** Proprietär, teuer und nicht für den Raspberry Pi optimiert, keine Web-Integration
- **R Notebooks:** weniger verbreitet in der Elektronik-Community und schlechtere Unterstützung für Hardware-Bibliotheken wie **daqhats**
- **Google Colab:** Cloudbasiert, erfordert Internet und unterstützt keine direkte Hardware-Steuerung (z.B. MCC118)

Die folgende Tabelle 5.5 vergleicht verschiedene Tools:

Tool	Vorteil	Nachteil	Warum nicht gewählt?
Jupyter Notebooks	interaktiv, Open Source, Python-basiert	höherer Speicherbedarf	gewählt für Interaktivität
Standard-Skripte	einfach, leichtgewichtig	keine interaktive Visualisierung	ungeeignet für Analysen
MATLAB	starke Analyse-Tools	proprietär, keine Hardware-Steuerung	Kosten und eingeschränkte Flexibilität
Google Colab	Cloud-basiert, einfach zu teilen	Kein Hardware-Zugriff	Nicht für Raspberry Pi geeignet

Tabelle 5.5: Alternative zu Jupyter Notebooks und deren Eigenschaften

Ein Beispielsnippet aus **Diodenkennlinie.ipynb** veranschaulicht die Interaktivität:

```
import numpy as np
from IPython.display import display
```

```

import ipywidgets as widgets

# Interaktive Eingabe
start_voltage = widgets.FloatSlider(min=0, max=5, value=0, description='Start (V):')
end_voltage = widgets.FloatSlider(min=0, max=5, value=2, description='Ende (V):')
display(start_voltage, end_voltage)

```

5.2.4 Modulare Struktur und Integration

Die Software ist modular aufgebaut, um Wiederverwendbarkeit und Erweiterbarkeit zu gewährleisten. Das Dashboard dient als zentraler Einstiegspunkt und startet Submodule wie separate Prozesse. Die Integration erfolgt über die ProcessManager-Klasse, die Prozesse startet, überwacht und Ports zuweist.

Die Architektur umfasst:

- **Dashboard:** Zentrale Weboberfläche mit Links (Buttons) zu den Modulen
- **Module:** funktionenspezifische Apps
- **Jupyter Notebooks**

Der Entwicklungsprozess umfasste drei Phasen:

1. **Lokale Skripte:** Implementierung grundlegender Funktionen wie der Netzteil-Funktion **Netzteil_plus.py** und **Netzteil_minus.py** auf dem Raspberry Pi 5 mit **daqhats**
2. **Jupyter Notebooks:** Einführung interaktiver Analysen in **Diodenkennlinie.ipynb**. Benutzereingaben wie der Spannungsbereich werden unterstützt
3. **Web-Integration:** Entwicklung von Dash-Apps für den Remote-Zugriff auf den Raspberry Pi 5, gesteuert über das Dashboard

Die meisten Skripte unterstützen einen Simulationsmodus, mit dem sich Tests ohne Hardware durchführen lassen. Das ist nützlich für das Debugging und die Entwicklung auf Nicht-Raspberry-Pi-Systemen.

Ein Beispiel aus **Funktionsgenerator_web.py**:

```

SIMULATION_MODE = '--simulate' in sys.argv
if SIMULATION_MODE:
    logger.info("AD9833 simuliert initialisiert")

```

Der Simulationsmodus reduziert Abhängigkeiten und beschleunigt die Entwicklung, insbesondere bei der Arbeit an Weboberflächen oder bei Analysen ohne Hardware-Zugang.

5.3 Implementierung der Funktionen

5.3.1 Multimeter

Das Digitalmultimeter (DMM) des OurDAQ-Systems ist als vielseitige, webbasierte Anwendung konzipiert, die grundlegende Messungen von Spannung und Strom ermöglicht. Die Abbildung 5.1 zeigt die Oberfläche von DMM.

Das DMM basiert auf dem Python-Framework Dash, um eine reaktionsfähige und netzwerkfähige Benutzeroberfläche zu schaffen. Dank dieser Architektur können die Messungen von jedem beliebigen Gerät mit einem Webbrowser aus gesteuert und überwacht werden.

Um eine flüssige Benutzererfahrung zu gewährleisten, wird die kontinuierliche Datenerfassung in einem separaten Thread (`_measurement_loop`) ausgeführt. Dieser Thread liest in kurzen Intervallen (ca. 20 Hz) die Messwerte vom MCC 118 DAQ HAT aus und speichert sie in einem threadsicheren Zwischenspeicher (`display_cache`). Über periodische Callbacks greift die Weboberfläche auf diesen Zwischenspeicher zu, um die Anzeige zu aktualisieren, ohne die Messung selbst zu blockieren.

Das DMM-Modul unterstützt vier grundlegende Messmodi, die über ein Dropdown-Menü wie in Abbildung 5.3 ausgewählt werden können:

- **Gleichspannung** (DC-Spannung)
- **Wechselspannung** (AC-Spannung)
- **Gleichstrom** (DC-Strom)
- **Wechselstrom** (AC-Strom)

Eine Besonderheit ist die erweiterte Behandlung von Wechselspannung. Der Benutzer kann die erwartete Wellenform des AC-Signals (Sinus, Dreieck, symmetrisches Rechteck, asymmetrisches Rechteck) spezifizieren wie in Abbildung 5.4. Die Software passt daraufhin die Berechnung und Anzeige des Messwertes an. Beispielsweise wird für Sinus- und Dreieckwellen der Effektivwert (RMS) berechnet und angezeigt, während für Rechteckwellen der Spitzenwert relevanter ist. Mithilfe dieser Funktionalität können die Messwerte je nach Signalart genauer interpretiert werden.

Neben der reinen Anzeige des aktuellen Messwerts bietet das DMM-Modul auch Funktionen zur Datenaufzeichnung und Visualisierung wie in Abbildung 5.2:

- **Echtzeit-Diagramm:** Nach dem Start der Aufzeichnung werden die Messwerte in einem Zeitverlaufsdiagramm dargestellt, das mit **Plotly** realisiert wurde. Um die Performance zu optimieren, wird hierfür eine `deque` mit einer maximalen Punktzahl (`max_punkte`) verwendet, was zu einem flüssigen, rollenden Graphen führt
- **Aufzeichnungssteuerung:** Buttons für Start, Pause und Stop ermöglichen eine präzise Steuerung der Datenaufzeichnung

- **CSV-Export:** Aufgezeichnete Messreihen können per Button als CSV-Datei exportiert werden, um eine weiterführende Analyse in externen Programmen wie Microsoft Excel zu ermöglichen

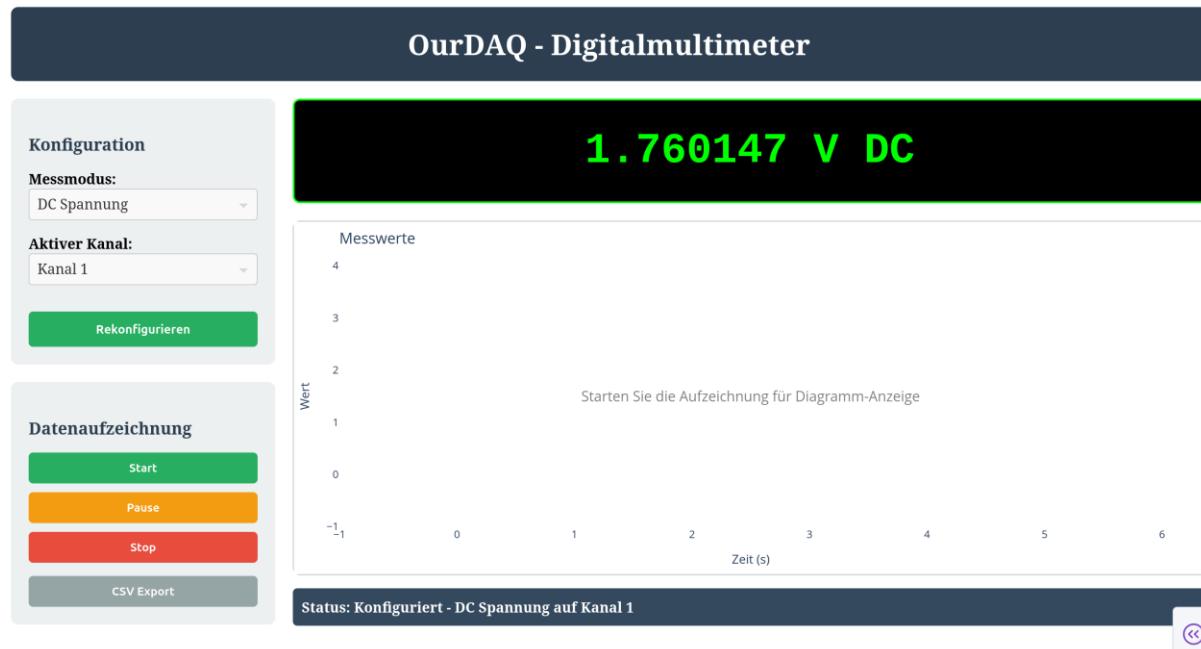


Abbildung 5.1: DC-Spannung Messung

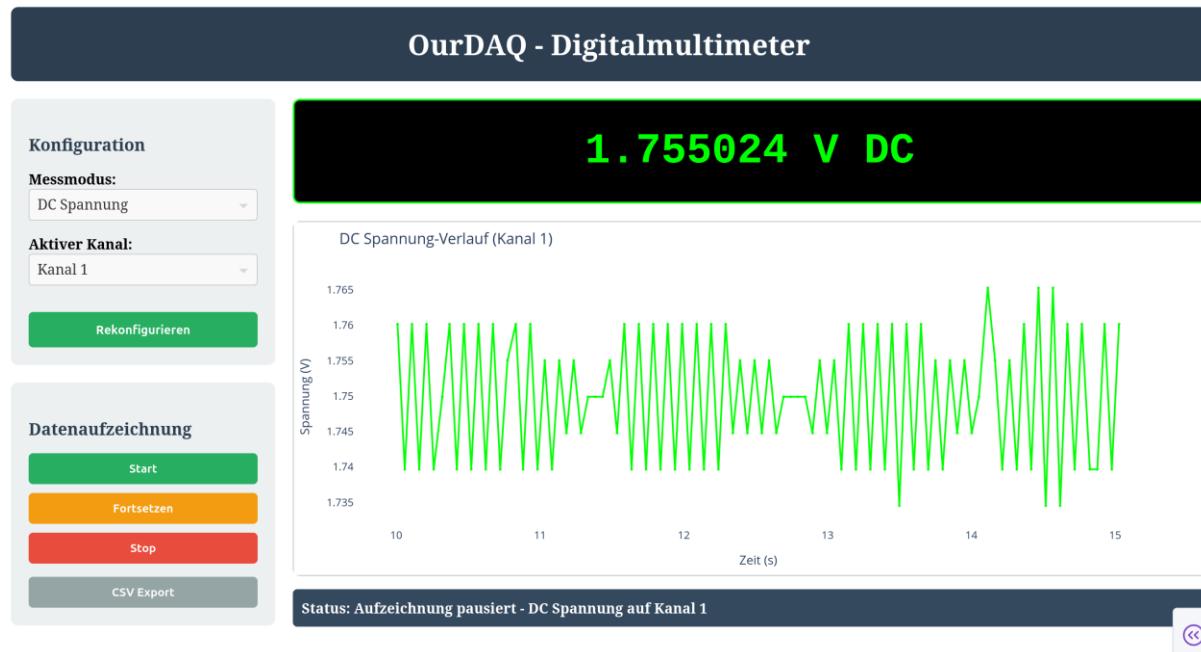


Abbildung 5.2: Datenaufzeichnung in DMM

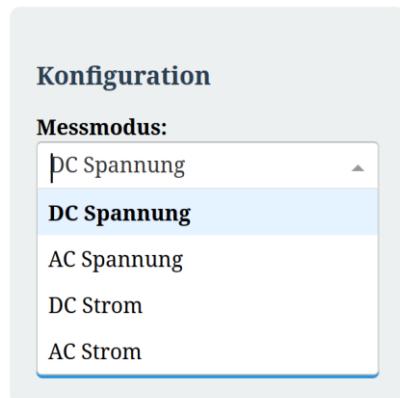


Abbildung 5.3: Konfiguration für Messmodus in DMM



Abbildung 5.4: Konfiguration für Wellenform in DMM

5.3.2 Funktionsgenerator

Der Funktionsgenerator ist eine wesentliche Komponente des OurDAQ-Systems und dient zur Erzeugung periodischer Signale für Test- und Messzwecke. Die softwareseitige Implementierung konzentriert sich auf die Ansteuerung eines AD9833-Signalgenerators über die SPI-Schnittstelle des Raspberry Pi 5. Die Abbildung 5.5 zeigt die Oberfläche von Funktionsgenerator.

Die Ansteuerung des AD9833 erfolgt über zwei primäre Schnittstellen des Raspberry Pi: SPI für die Datenübertragung und GPIO für das Chip-Select-Signal, das hier als **FSYNC** bezeichnet wird.

Zur Implementierung werden die Python-Bibliotheken **spidev** und **Igpio** verwendet. **spidev** ermöglicht die Kommunikation über den SPI-Bus, während **Igpio** den direkten Zugriff auf die GPIO-Pins für das Timing-kritische **FSYNC**-Signal bereitstellt.

Initialisierungsprozess:

- 1) GPIO-Setup: Der für **FSYNC** designierte GPIO-Pin (**Pin 25**) wird als Ausgang konfiguriert und initial auf HIGH gesetzt, um den AD9833 in einen inaktiven Zustand zu versetzen.
- 2) SPI-Konfiguration: Die SPI-Schnittstelle (Bus 0, Device 0) wird mit einer Taktfrequenz von 1 MHz und im SPI-Modus 2 (**CPOL=1, CPHA=0**) initialisiert, wie es das Datenblatt des AD9833 erfordert.
- 3) Initiales Reset: Nach der Konfiguration wird ein 16-Bit-Reset-Befehl (**0x2100**) an den AD9833 gesendet, um ihn in einen definierten Ausgangszustand zu versetzen. Eine kurze Pause von 0,1 Sekunden stellt sicher, dass der Reset-Vorgang abgeschlossen ist.
- 4) Die Erzeugung der Signale basiert auf dem DDS-Prinzip. Ein 28-Bit Frequenz-Steuerwort wird in das Frequenzregister des AD9833 geschrieben, um die Ausgangsfrequenz präzise einzustellen.

Frequenzeinstellung: `set_ad9833_frequency`

Die gewünschte Ausgangsfrequenz wird anhand der Master-Taktfrequenz (**FMCLK = 25 MHz**) in das digitale Frequenzwort umgerechnet: **freq_word = int(freq_hz * (2**28)) / FMCLK**.

Da das Frequenzwort 28 Bit breit ist, die SPI-Kommunikation aber in 16-Bit-Paketen erfolgt, muss das Wort aufgeteilt und in einer spezifischen, kritischen Sequenz an den AD9833 gesendet werden, um eine korrekte Funktion zu gewährleisten:

- 1) Senden des Reset-Befehls.
- 2) Senden der unteren 14 Bits des Frequenzwortes an das Frequenzregister 0 (**FREQ0_REG**).
- 3) Senden der oberen 14 Bits des Frequenzwortes an dasselbe Register.

Wellenform-Auswahl: `activate_waveform`

Nach der Frequenzeinstellung wird die gewünschte Wellenform durch das Senden eines Steuerbefehls an das Kontrollregister aktiviert. Dieser Befehl beendet gleichzeitig den Reset-Zustand und startet die Signalausgabe. Hierfür werden definierte Konstanten wie **SINE_WAVE (0x2000)**, **TRIANGLE_WAVE (0x2002)** und **SQUARE_WAVE (0x2028)** verwendet.

In einer optimierten Testversion **Funktionsgenerator_optimiert.py** wird zusätzlich ein separater GPIO-Pin (**Pin 24**) genutzt, um extern zwischen Rechteck- und anderen Wellenformen umzuschalten. Dadurch ist eine hybridere Hardware-Software-Steuerung möglich.

Um eine flexible Bedienung zu ermöglichen, wurden zwei verschiedene Benutzeroberflächen implementiert: eine kommandozeilenbasierte und eine webbasierte Schnittstelle.

Kommandozeilen-Interface: `Funktionsgenerator.py`

Die Basisimplementierung bietet eine einfache, menügeführte Steuerung über das Terminal. Der Benutzer wird schrittweise durch die Auswahl der Wellenform und die Eingabe der Frequenz geführt, woraufhin das Signal konfiguriert und ausgegeben wird.

Web-Interface: Funktionsgenerator_web.py

Für eine intuitivere und fernsteuerbare Bedienung wurde eine webbasierte Oberfläche mit dem Framework Dash entwickelt. Die Interaktion erfolgt über Dash-Callbacks, die bei Benutzeraktionen die entsprechenden Python-Funktionen zur Hardware-Ansteuerung `set_ad9833_frequency`, `activate_waveform` aufrufen.

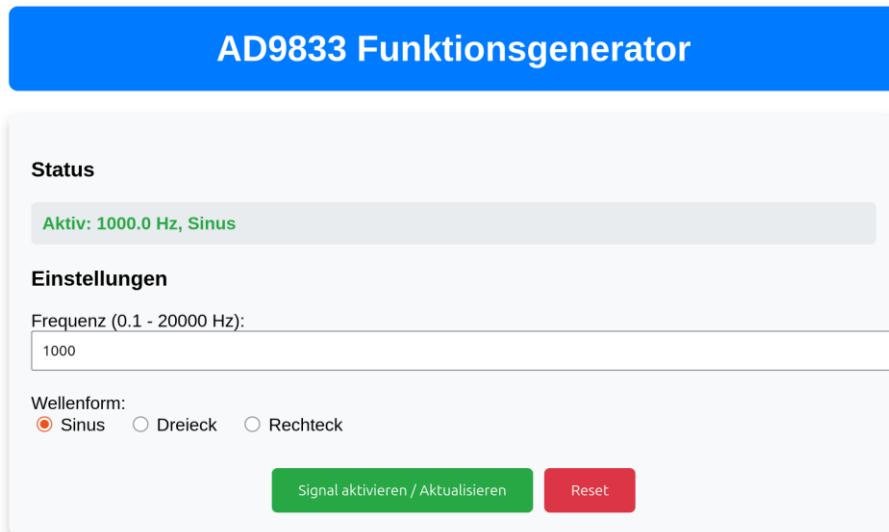


Abbildung 5.5: Konfiguration für Funktionsgenerator

5.3.3 Oszilloskop

Das Oszilloskop-Modul ist eine zentrale Funktion des OurDAQ-Systems und ermöglicht die Visualisierung von Spannungsverläufen in Echtzeit. Im Gegensatz zu traditionellen, kommandozeilenbasierten Ansätzen wurde für dieses Modul eine interaktive, webbasierte Benutzeroberfläche implementiert, um eine intuitive Bedienung und einen plattformunabhängigen Zugriff zu gewährleisten. Die Abbildung 5.6 zeigt die Oberfläche von Oszilloskop.

Die Benutzeroberfläche des Oszilloskops wurde mithilfe des Python-Frameworks Dash realisiert. Die Ansteuerung wird durch die `daqhats`-Bibliothek abstrahiert.

Kontinuierlicher Scan-Modus:

Für die Oszilloskop-Funktion ist eine kontinuierliche Abtastung der Eingangskanäle erforderlich. Dies wird durch die Funktion `hat.a_in_scan_start` mit dem Parameter `OptionFlags.CONINUOUS` realisiert. Die Messdaten werden dabei in einen internen Puffer geschrieben, aus dem sie in regelmäßigen Abständen ausgelesen werden.

Threading und Datenpufferung:

Um die Benutzeroberfläche reaktionsfähig zu halten, während kontinuierlich Daten erfasst werden, wird die Messschleife in einem separaten Thread (`_measurement_loop`) ausgeführt. Die erfassten Messwerte werden in einer `deque (double-ended queue)` mit einer festen maximalen Länge gespeichert. Dieses Vorgehen stellt sicher, dass immer nur die neuesten Datenpunkte für die Visualisierung vorgehalten werden, was die Performance verbessert und einen rollenden Graphen erzeugt.

Die Weboberfläche bietet dem Benutzer umfassende Kontrolle über die Messparameter:

- **HAT-Auswahl:** Falls mehrere MCC 118 HATs gestapelt sind, kann das gewünschte Gerät über ein Dropdown-Menü ausgewählt werden
- **Kanalauswahl:** Der Benutzer kann bis zu acht Eingangskanäle über Checkboxen aktivieren oder deaktivieren
- **Abtastrate:** Die Abtastrate kann in einem definierten Bereich eingestellt werden. Eine Validierungsfunktion (`validiere_abtastrate`) stellt sicher, dass die gewählte Rate die aggregierte maximale Abtastrate des MCC 118 von 100 kS/s, geteilt durch die Anzahl der aktiven Kanäle, nicht überschreitet
- **Anzeigedauer:** Die Anzahl der gleichzeitig im Diagramm dargestellten Samples kann konfiguriert werden

Die Konfiguration wird über den Button Konfigurieren abgeschlossen. Dieser startet die Messung und sperrt die Eingabefelder, um Änderungen während des Betriebs zu verhindern.

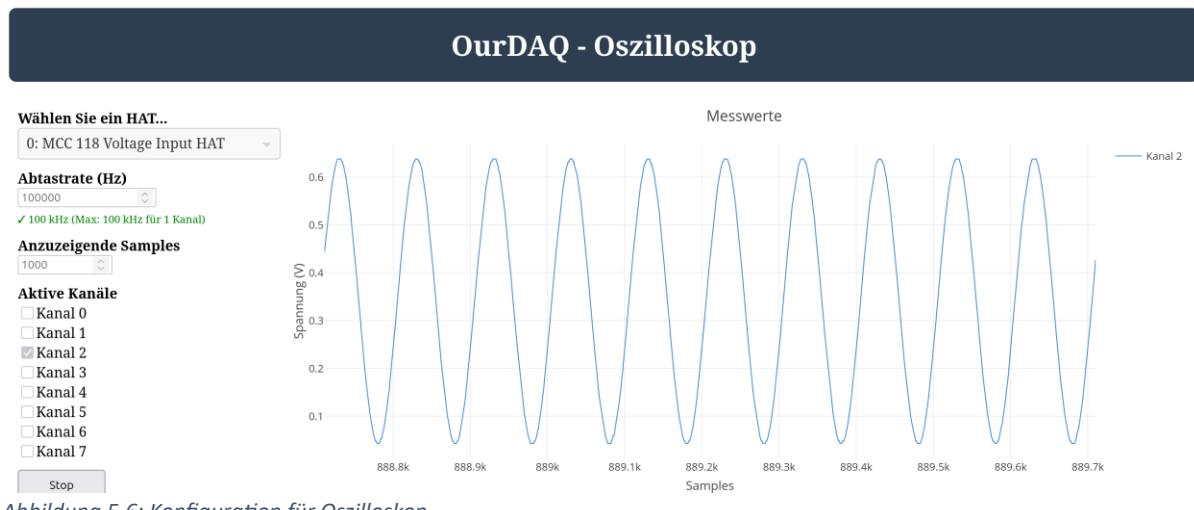


Abbildung 5.6: Konfiguration für Oszilloskop

5.3.4 Netzteil

Die Netzteilfunktionalität des OurDAQ-Systems dient der Bereitstellung von stabilen, einstellbaren Gleichspannungen zur Versorgung externer Schaltungen. Die Software realisiert zwei unabhängige, regelbare Spannungsquellen: eine für positive und eine für negative Spannungen. Beide Module basieren auf demselben Grundprinzip der digitalen Ansteuerung eines Digital-Analog-Wandlers (DAC) mit kontinuierlicher Rückmessung des Ausgangsstroms über das MCC 118 DAQ HAT.

Über die SPI-Schnittstelle werden zwei separate 12-Bit-DACs angesteuert, einer für die positive und einer für die negative Spannungsschiene. Sie werden über unterschiedliche Chip-Select-Pins (CS) adressiert, **Pin 22** für die positive und **Pin 27** für die negative Spannungsschiene. Die Abbildung 5.7 zeigt die Oberfläche für positive Spannung und die Abbildung 5.8 zeigt die Oberfläche für negative Spannung.

DAC-Ansteuerung: `write_dac`

Die Funktion `write_dac` sendet ein 16-Bit-Wort über SPI an den jeweiligen DAC. Dieses Wort enthält neben dem 12-Bit-Spannungswert (0-4095) auch Steuerbits. Um die entsprechenden DAC-Bausteine korrekt zu konfigurieren, werden für die positive und negative Spannungsversorgung unterschiedliche Steuerbits verwendet.

Da die Beziehung zwischen dem digitalen DAC-Wert und der analogen Ausgangsspannung aufgrund von Bauteiltoleranzen und nichtlinearen Effekten nicht perfekt linear ist, ist eine Kalibrierung für eine präzise Spannungseinstellung unerlässlich.

Kalibrierungsprozess: `kalibrieren`

Die Software führt beim Start eine automatische Kalibrierung durch. Dabei wird der DAC schrittweise über seinen gesamten Wertebereich (0 bis 4095) gefahren. Dabei wird bei jedem Schritt die resultierende Ausgangsspannung mit dem MCC 118 gemessen. Die Paare aus gemessener Spannung und DAC-Wert werden in einer Kalibriertabelle gespeichert. Bei der negativen Spannungsversorgung werden nur Messpunkte mit negativer Spannung berücksichtigt.

Lineare Interpolation: `spannung_zu_dac_interpoliert`

Wenn der Benutzer eine Zielspannung vorgibt, wird die Kalibriertabelle verwendet, um den passenden DAC-Wert zu ermitteln. Anstatt einen theoretischen Wert zu berechnen, wird mittels linearer Interpolation zwischen den nächstgelegenen Kalibrierpunkten der präziseste DAC-Wert gefunden.

Die kommandozeilenbasierten Skripte `Netzteil_plus.py` und `Netzteil_minus.py` bieten erweiterte Funktionalitäten zur Strommessung und zum Schutz der Hardware.

- **Kontinuierliche Strommessung:** Nach Einstellung der Spannung startet eine Dauerschleife `strom_ueberwachung`, die den Strom über einen Shunt-Widerstand und einen weiteren Kanal des MCC 118 (Kanal 4 für positiv, Kanal 5 für negativ) misst
- **Lineare Korrektur:** Um die Genauigkeit der Strommessung zu verbessern, wird eine lineare Korrekturformel $I_{true_mA} = corr_a + corr_b * I_{mcc_mA}$ auf die Rohdaten angewendet
- **Überstromschutz:** Überschreitet der gemessene Strom einen vordefinierten Schwellenwert `MAX_STROM_MA`, wird der DAC-Ausgang sofort auf 0 gesetzt, um die angeschlossene Schaltung und die Spannungsversorgung zu schützen

Die Skripte **Netzteil_plus_web.py** und **Netzteil_minus_web.py** stellen eine vereinfachte, aber benutzerfreundliche Weboberfläche mittels Dashs zur Verfügung. Diese fokussiert sich auf die Kernfunktionalitäten:

- Ein Button startet den Kalibrierungsvorgang, dessen Log direkt auf der Webseite angezeigt wird
- Ein Schieberegler (Slider) und ein synchronisiertes Eingabefeld ermöglichen die intuitive Einstellung der Zielspannung
- Im Gegensatz zur Kommandozeilen-Version wurde in der Web-Implementierung auf die Stromüberwachung und den Überstromschutz verzichtet, um die Komplexität zu reduzieren

Labornetzteil Steuerung (Positive Spannung)

1. Kalibrierung

Automatische Kalibrierung starten

```
Starte Kalibrierung (Positive Spannung)...
DAC 0 -> 0.00758 V
DAC 32 -> 0.07896 V
DAC 64 -> 0.166053 V
DAC 96 -> 0.24210 V
DAC 128 -> 0.31858 V
DAC 160 -> 0.40015 V
DAC 192 -> 0.48173 V
DAC 224 -> 0.56330 V
DAC 256 -> 0.63978 V
DAC 288 -> 0.72645 V
DAC 320 -> 0.80292 V
DAC 352 -> 0.88450 V
DAC 384 -> 0.96097 V
DAC 416 -> 1.01755 V
```

2. Spannung einstellen

Stellen Sie die gewünschte positive Spannung ein (nur nach Kalibrierung).



Spannung auf 10.000 V gesetzt (DAC=3988).

Abbildung 5.7: Netzteilsteuerung für positive Seite

Labornetzteil Steuerung (Negative Spannung)

1. Kalibrierung

Automatische Kalibrierung starten

```
Starte Kalibrierung (Negative Spannung)...
DAC 0 -> -0.00262 V
DAC 32 -> -0.09949 V
DAC 64 -> -0.20146 V
DAC 96 -> -0.29323 V
DAC 128 -> -0.38500 V
DAC 160 -> -0.47677 V
DAC 192 -> -0.56854 V
DAC 224 -> -0.66541 V
DAC 256 -> -0.75718 V
DAC 288 -> -0.85405 V
DAC 320 -> -0.94582 V
DAC 352 -> -1.03759 V
DAC 384 -> -1.13446 V
... 112 -> -1.22729 V
```

2. Spannung einstellen

Stellen Sie die gewünschte negative Spannung ein (nur nach erfolgreicher Kalibrierung möglich).



Spannung auf -10.000 V gesetzt (DAC=3424).

Abbildung 5.8: Netzteilsteuerung für negative Seite

5.3.5 Kennlinienmessung

Die Kennlinienmessung analysiert die elektrischen Eigenschaften von Bauelementen (z.B. Dioden) und Schaltungen (z.B. Filter) durch Messung von Strom-Spannungs- oder Frequenzgang-Kennlinien. Die Implementierung erfolgt hauptsächlich über Jupyter Notebooks **Diodenkennlinie.ipynb**, **Filterkennlinie.ipynb**, die interaktiven Eingaben und Visualisierungen ermöglichen. Diese Module nutzen den MCC118 (ADC) für Messungen und den AD9833 (Funktionsgenerator) oder DAC (Spannungsquelle) für Stimuli.

5.3.5.1 Diodenkennlinie

Das Ziel ist die Messung der Strom-Spannungs-Kennlinie einer Diode, um eine einfache U-I-Kennlinie (U-I-Diagramm) aufzunehmen und darzustellen. Die Messung erfolgt durch schrittweises Anlegen einer Spannung (via DAC) und Messen des Stroms (via MCC 118 und Shunt-Widerstand).

Der Aufbau umfasst:

- **DAC**: Erzeugt präzise Spannungen (z.B. 0–5 V) als Stimulus.
- **MCC118**: Misst Spannung über einen Shunt-Widerstand zur Strombestimmung.
- **Diode**: Z.B. 1N4148, in Reihe mit dem Shunt.
- **Raspberry Pi 5** mit GPIO-Erweiterung
- **SPI-Schnittstelle** für DAC-Kommunikation

Die Spannung wird in Schritten von 0,1 V angelegt, und der Strom wird über $I = U_{\text{Shunt}} / R_{\text{Shunt}}$ berechnet. Kalibrierung ist entscheidend, um Messfehler (z.B. Offset des MCC118) zu minimieren.

Messungen:

- Kanal 6: Spannung über der Diode
- Kanal 7: Gesamtspannung (Referenz/Kalibrierung)

Das Jupyter Notebook **Diodenkennlinie.ipynb** ermöglicht interaktive Eingaben (z.B. Spannungsbereich) und Visualisierungen. Ein Beispiel-Snippet für die Messschleife:

```
import numpy as np
from mcc118 import mcc118
from matplotlib import pyplot as plt

# Initialisierung
address = 0
mcc = mcc118(address)
voltages = np.linspace(0, 5, 50) # Spannung von 0 bis 5 V
currents = []

# Messschleife
for v in voltages:
    set_dac_voltage(v) # DAC-Funktion (angenommen)
    v_shunt = mcc.a_in_read(channel=0) # Spannung über Shunt
    current = v_shunt / R_SHUNT # Strom berechnen
    currents.append(current)

# Plot
plt.plot(voltages, currents, 'o-')
plt.xlabel('Spannung (V)')
plt.ylabel('Strom (mA)')
plt.title('Diodenkennlinie')
plt.show()
```

Das Notebook erstellt eine U-I-Kennlinie (U-I-Diagramm) der Diode basierend auf den gemessenen Werten von Diodenspannung und Strom. Die Ergebnisse werden als Plot (z.B. (I_D) vs. (V_D)) und Tabelle ausgegeben.

Die Messprozedur ist folgende:

1. Eingabe des Serienwiderstands-Wertes
2. Festlegung der Anzahl Messpunkte
3. Definition des Spannungsbereichs

4. Automatische Messwerterfassung
5. Echtzeit-Anzeige der Messwerte

Das Programm erzeugt zwei Diagramme wie in Abbildung 5.9:

- **Diodenkennlinie:** Strom vs. Spannung über der Diode
- **Systemkennlinie:** Strom vs. eingestellte DAC-Spannung

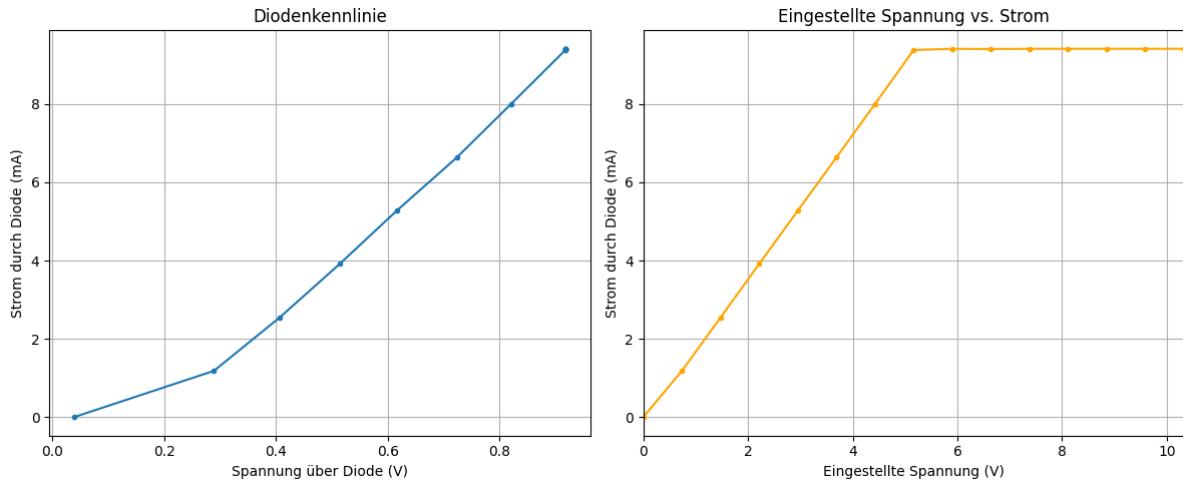


Abbildung 5.9: Diodenkennlinie

5.3.5.2 Filterkennlinie

Das Ziel ist die Messung des Frequenzgangs eines Filters (z.B. Tiefpass), dargestellt als Bode-Diagramm (Amplitudengang und Phasenverschiebung). Der Amplitudengang wird berechnet als:

$$A = 20 * \log_{10}(V_{out} / V_{in})$$

Die Phasenmessung ist ein Platzhalter, da sie in **Filterkennlinie.ipynb** implementiert ist.

- **AD9833:** Erzeugt Sinus-Signale (z.B. 10 Hz bis 10 kHz) über SPI
- **MCC118:** Misst die Eingangs- und Ausgangsspannung des Filters
- **Filter:** Z.B. RC-Tiefpass mit definierter Grenzfrequenz

Das System führt eine **semi-automatische Frequenzganganalyse** durch:

- **Kanal 6:** Eingangssignal vom Funktionsgenerator
- **Kanal 7:** Ausgangssignal des zu testenden Filters/Verstärkers
- **Simultane Zweikanal-Messung** für die präzise Bestimmung der Übertragungsfunktion

Das Notebook ermöglicht Eingaben wie den Frequenzbereich und misst den Amplitudengang. Beispiel-Snippet:

```
# Initialisierung
ad9833 = AD9833() # SPI-Interface
mcc = mcc118(address=0)
frequencies = np.logspace(1, 4, 50) # 10 Hz bis 10 kHz
amplitudes = []

# Messschleife
for freq in frequencies:
    ad9833.set_frequency(freq, 'SINE')
    v_in = mcc.a_in_read(channel=0) # Eingangsspannung
    v_out = mcc.a_in_read(channel=1) # Ausgangsspannung
    amplitude = 20 * np.log10(v_out / v_in)
    amplitudes.append(amplitude)

# Plot
plt.semilogx(frequencies, amplitudes)
plt.xlabel('Frequenz (Hz)')
plt.ylabel('Amplitudengang (dB)')
plt.title('Bode-Diagramm (Tiefpassfilter)')
plt.grid(True)
plt.show()
```

Das Notebook erstellt ein Bode-Diagramm mit Amplitudengang (Gain in dB) und Phasengang. Die Phasenmessung erfolgt via Lock-In mit Korrektur für den Zeitversatz (Multiplexer Delay). Zusätzlich wird ein Plot der Effektivwerte (Vrms) für Ein- und Ausgang erstellt. Ergebnisse werden als Plots ausgegeben.

AC-RMS Messung

```
def ac_rms(x):
    """AC-RMS nach Entfernen des DC-Anteils"""
    x_ac = x - np.mean(x)
    return np.sqrt(np.mean(x_ac * x_ac)), x_ac
```

- Entfernt den DC-Offset für korrekte Verstärkungsberechnung
- Berechnet den Effektivwert nur der AC-Komponente
- **Vorteil:** Unabhängig von Gleichspannungsanteilen

Lock-In Phasenmessung

```
def lockin_phase_deg(signal_ac, fs, f_hz):
    """Phase via I/Q-Demodulation"""
    # I-Komponente: In-Phase
    I = (2.0/N) * np.dot(signal_ac, cos(2πft))
    # Q-Komponente: Quadratur-Phase
    Q = (2.0/N) * np.dot(signal_ac, -sin(2πft))
    return arctan2(Q, I)
```

Multiplexer-Delay Korrektur

Das MCC118 tastet Kanäle **sequenziell** ab (nicht simultan):

```
# Zeitversatz zwischen Kanälen
dt = 8e-6 # 8μs (aus Datenblatt)
phi_delay = 360.0 * f * dt # Frequenzabhängige Phasenkorrektur
phi_corrected = phi_raw - phi_delay
```

- **Problem:** Kanal 7 wird 8 μs nach Kanal 6 gemessen
- **Auswirkung:** Phasenfehler von 2.88° pro kHz!
- **Lösung:** mathematische Korrektur der scheinbaren Phasenverzögerung

Die Messprozedur ist wie folgt:

1. Parameter eingeben: Startfrequenz (z.B. 10 Hz), Stoppfrequenz (z.B. 10000 Hz), Anzahl Schritte (z.B. 50)
2. Für jede Frequenz den Funktionsgenerator manuell einstellen
3. **Enter** drücken → automatische Messung
4. Anzeige der Ergebnisse

Das Programm zeigt drei Diagramme wie in Abbildung 5.10:

- **Amplitudengang:** Verstärkung in dB über der Frequenz
- **Phasengang:** Phasendifferenz in Grad (korrigiert)
- **RMS-Monitoring:** Kontrolle des Eingangspegels

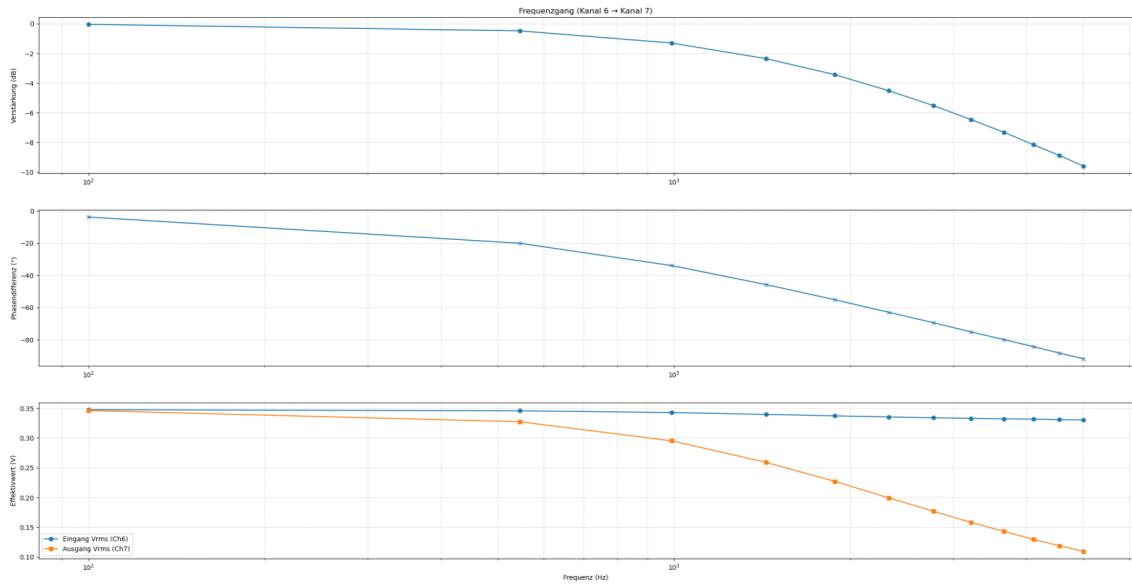


Abbildung 5.10: Filterkennlinie

5.3.6 Integration mit Dashboard

Das Dashboard (**Dashboard_web.py**) ist die zentrale Web-App, die alle Module integriert. Die Abbildung 5.11 zeigt die Oberfläche von Dashboard. Es bietet eine Benutzeroberfläche mit Links zu jedem Modul und startet diese als separate Prozesse. Die App nutzt Dash für die Web-Implementierung und läuft auf dem Raspberry Pi 5 unter einer IP-Adresse: **http://<IP-Adresse-raspberrypi>:8000**

Ein Beispiel-Layout:

```
from dash import Dash, html
app = Dash(__name__)
app.layout = html.Div([
    html.H1("Datenerfassungsmodul Dashboard"),
    html.A("Multimeter", href="/multimeter"),
    html.A("Oszilloskop", href="/oszilloskop"),
    html.A("Funktionsgenerator", href="/funktionsgenerator"),
    html.A("Netzteil", href="/netzteil")]
```



Abbildung 5.11: Dashboard-Oberfläche

Das Dashboard des OurDAQ Datenerfassungssystems dient als zentrale Benutzeroberfläche zur Überwachung und Steuerung des Systems. Es bietet eine übersichtliche Darstellung des Systemstatus und direkten Zugriff auf die verfügbaren Module und Funktionen. Die Oberfläche ist in einem modernen, klaren Design gehalten, mit einem dunklen Header und hellen, abgerundeten Karten für die Inhalte. Der Header enthält den Systemnamen "OurDAQ Datenerfassungssystem" sowie eine Statusanzeige, die die Anzahl aktiver Module, die IP-Adresse und die aktuelle Uhrzeit anzeigt. Im Folgenden wird das Dashboard detailliert beschrieben, unterteilt in seine Hauptabschnitte.

- **Netzwerk-Karte (links oben):**
 - **IP-Adresse:** Zeigt die aktuelle IP-Adresse des Systems an. Dies ermöglicht es dem Benutzer, das System im Netzwerk zu identifizieren und darauf zuzugreifen
 - **Laufzeit:** Gibt die Betriebszeit des Systems seit dem letzten Start an. Diese Information ist nützlich für die Überwachung der Systemstabilität
- **Hardware-Karte (rechts oben):**
 - **Simulation/Hardware:** Informiert darüber, ob das System im Simulationsmodus oder Hardwaremodus.
 - **Raspberry Pi:** Zeigt den Status des angeschlossenen Raspberry Pi an.
 - **Dashboard Port:** Gibt den Port an, auf dem das Dashboard erreichbar ist (**8000**). Dies ist relevant für die Konfiguration von Netzwerzkonfigurationen oder Firewalls

Die Hardware-Karte ist mit einer roten Umrandung versehen, was möglicherweise auf einen Warn- oder Fehlerstatus hinweist (z. B. aufgrund der fehlenden Raspberry Pi-Verbindung oder des Simulationsmodus).

Unter der Systemübersicht folgt der Abschnitt "Module & Funktionen", in dem die Kernfunktionalitäten des Systems als interaktive Buttons dargestellt sind. Diese farblich codierten Buttons sind horizontal angeordnet, um eine intuitive Navigation zu ermöglichen. Jeder Button führt zu einer dedizierten Unterseite oder einem Modul, in dem der Benutzer spezifische Messungen oder Konfigurationen vornehmen kann. Die verfügbaren Module sind:

- **Digitalmultimeter** (blauer Button): Dieses Modul ermöglicht die Messung elektrischer Größen wie Spannung, Stromstärke oder Widerstand. Es simuliert oder steuert ein digitales Multimeter und ist für präzise Einzelmessungen geeignet
- **Funktionsgenerator** (roter Button): Hier kann der Benutzer Signale generieren, z. B. Sinus-, Rechteck- oder Dreieckwellen mit einstellbarer Frequenz, Amplitude und Phase. Dies ist nützlich für Testsignale in elektronischen Schaltungen
- **Oszilloskop** (grüner Button): Bietet die Visualisierung von Signalen über die Zeit, inklusive Trigger-Funktionen, Zoom und Analysewerkzeugen. Es dient der Beobachtung dynamischer Signale und ist essenziell für Debugging in der Elektronik
- **Netzteil positiv** (orangener Button): Steuert eine positive Spannungsquelle, z. B. für die Versorgung von Schaltungen. Der Benutzer kann hier Spannung und Stromgrenzen einstellen
- **Netzteil negativ** (orangener Button): Ähnlich wie das positive Netzteil, aber für negative Spannungen. Dies ermöglicht symmetrische Versorgungen, die in vielen analogen Schaltungen benötigt werden

Am unteren Rand des Dashboards befindet sich ein Hinweis-Text. Dieser Hinweis weist auf erweiterte Funktionen hin, die eine Integration mit Jupyter Notebooks erfordern. Speziell für die Analyse von Dioden- oder Filterkennlinien (z. B. IV-Kurven oder Frequenzgänge) muss ein separater Jupyter-Server gestartet werden. Dies deutet auf eine hybride Architektur hin, bei der das Web-Dashboard mit Python-basierten Analyse-Tools (wie Jupyter) kombiniert wird.

5.4 Installation und Anleitung

Dieses Kapitel dient als praktische Anleitung zur Inbetriebnahme und Nutzung des OurDAQ-Systems. Es beschreibt die erforderlichen Systemvoraussetzungen, den Installationsprozess der Software sowie die Ausführung der einzelnen Mess- und Steuerungsapplikationen. Alle hier beschriebenen Schritte und Informationen basieren auf der Dokumentation des Projekts.

Auf der Softwareseite werden die folgenden Komponenten benötigt:

- **Betriebssystem:** Ein Linux-System wie **Raspberry Pi OS** oder **Ubuntu**.
- **Python-Version:** **Python 3.11** oder eine neuere Version.

- **Python-Paketmanager:** Der Python-Paketmanager **uv** wird hier für die Verwaltung der Abhängigkeiten verwendet.
- Die folgenden **Python-Bibliotheken** sind für die Ausführung der Software erforderlich:
 - **daqhats** >= 1.4.1.0
 - **dash** >= 3.2.0
 - **Igpio** >= 0.2.2.0
 - **matplotlib** >= 3.10.3
 - **notebook** >= 7.4.2
 - **numpy** >= 2.2.5
 - **pandas** >= 2.2.3
 - **pyqtgraph** >= 0.13.7
 - **spidev** >= 3.7

Zunächst wird das Betriebssystem auf den neuesten Stand gebracht, um Kompatibilität und Sicherheit zu gewährleisten. Dies geschieht mit den folgenden Befehlen:

```
sudo apt update
sudo apt full-upgrade
sudo reboot
```

Die herstellerspezifische Bibliothek für das MCC 118 DAQ HAT muss installiert werden. Dazu wird das offizielle Repository geklont und das mitgelieferte Installationsskript ausgeführt:

```
cd ~
git clone https://github.com/mccdaq/daqhats.git
cd ~/daqhats
sudo ./install.sh
```

Anschließend wird der Quellcode des OurDAQ-Projekts auf das lokale System kopiert:

```
cd ~
git clone https://github.com/MSY-Walter/OurDAQ.git
```

Nach der Installation der Systemabhängigkeiten wird die Python-Umgebung für das Projekt eingerichtet. Falls **uv** noch nicht auf dem System vorhanden ist, kann es mit dem folgenden Befehl installiert werden:

```
curl -LsSf https://astral.sh/uv/install.sh | sh
```

Das Projekt verwendet **uv** zur Verwaltung einer virtuellen Umgebung und der darin enthaltenen Pakete. Durch den Befehl **uv sync** werden alle in der **pyproject.toml**-Datei spezifizierten Abhängigkeiten in einer isolierten Umgebung installiert:

```
cd ~/OurDAQ
uv sync
```

Nach Abschluss dieser Schritte ist das OurDAQ-System vollständig installiert und betriebsbereit.

Die Funktionen des Systems können entweder einzeln oder über ein zentrales Dashboard gestartet werden. Das Dashboard dient als Hauptmenü und startet automatisch alle zugehörigen Web-Applikationen. Dies ist die empfohlene Methode zur Nutzung des Systems.

Starten des Dashboards:

```
uv run src/Dashboard_web.py
```

Öffnen Sie einen Webbrower (entweder auf dem Raspberry Pi selbst oder auf einem anderen Gerät im selben Netzwerk) und geben Sie die folgende Adresse ein: <http://<IP-Adresse-raspberrypi>:8000>

Es ist zu beachten, dass die Verwendung der Buttons in der aktuellen Version von **Dashboard_web.py** zum Starten des Funktionsgenerator-Programms dazu führt, dass der AD9833 nicht initialisiert werden kann. Dadurch wird die Steuerung seiner Wellenformausgabe verhindert. Wenn der Funktionsgenerator verwendet werden soll, wird daher empfohlen, das unabhängige Programm **Funktionsgenerator_web.py** auszuführen.

Jede Funktionalität kann auch als eigenständige Anwendung gestartet werden. Nach dem Start eines der folgenden Skripte kann die zugehörige Oberfläche über einen Webbrower unter der angegebenen Adresse aufgerufen werden.

Funktionalität	Skript	Startbefehl	Port
Digitalmultimeter	DMM_web.py	uv run src/DMM_web.py	8050
Funktionsgenerator	Funktionsgenerator_web.py	uv run src/Funktionsgenerator_web.py	8060
Netzteil (positiv)	Netzteil_plus_web.py	uv run src/Netzteil_plus_web.py	8071
Netzteil (negativ)	Netzteil_minus_web.py	uv run src/Netzteil_minus_web.py	8072
Oszilloskop	Oszilloskop_web.py	uv run src/Oszilloskop_web.py	8080

Für die Aufnahme von Dioden- und Filterkennlinien werden Jupyter Notebooks verwendet. Hierfür muss der Jupyter Lab Server gestartet werden.

Starten des Jupyter Servers:

```
uv run jupyter lab --ip=0.0.0.0 --port=8888
```

Navigieren Sie im Browser zu <http://<IP-Adresse-raspberrypi>:8888> und öffnen Sie die entsprechenden Notebook-Dateien (**Diodenkennlinie.ipynb**, **Filterkennlinie.ipynb**).

6 Design - Schaltplan

Der komplette Schaltplan des OurDAQ wurde in KiCad umgesetzt (siehe Anhang 4). Außerdem ist zu erkennen, dass für jedes integrierte Bauteil ein 100 nF Blockkondensator direkt an den Versorgungspins vorgesehen ist. [33, 34]

An den Versorgungsein- und -ausgängen sind Sicherungen und Verpolschutzdioden platziert, um das System vor fehlerhafter Verkabelung und Kurzschlägen zu schützen. Zusätzlich wurden je Versorgungsschiene Pufferkondensatoren von 100 μ F am Versorgungseingang platziert (+15 V, -15 V und 5 V), um Stabilität bei Lastspitzen zu gewährleisten (vgl. Abbildung 6.1).

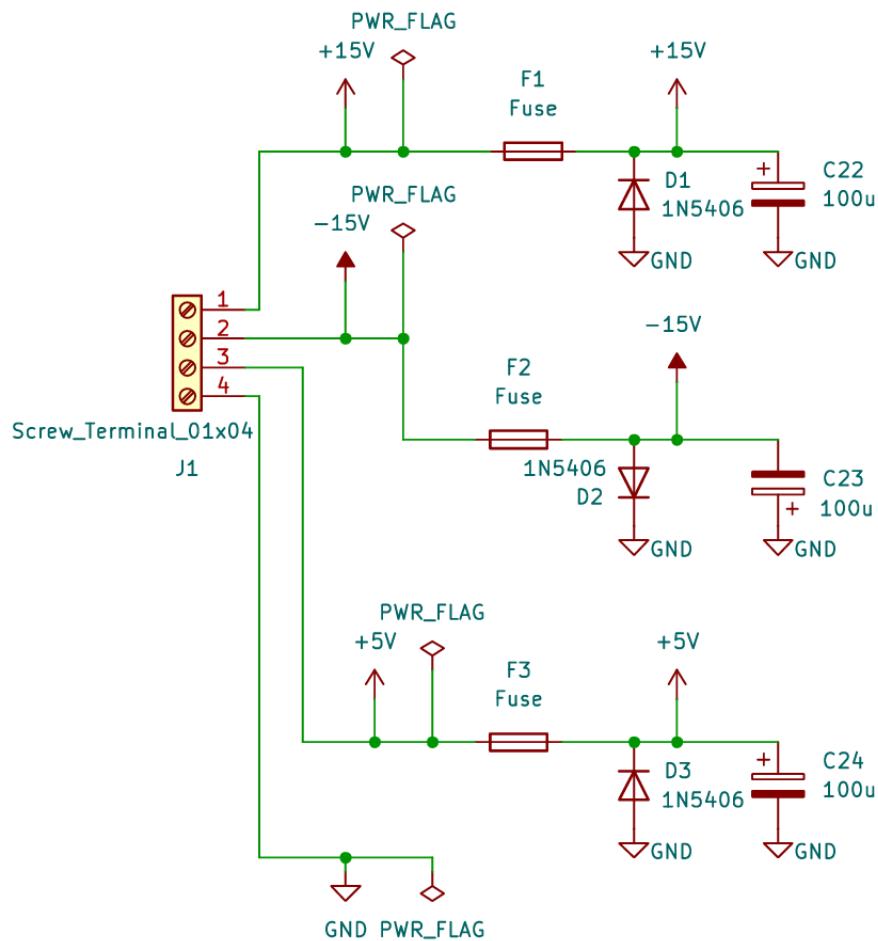


Abbildung 6.1: Leiterplatte - Versorgungsspannungen

An den Eingängen des MCC 118 HAT wurden Schottky-Dioden ergänzt, die bei Überspannung schnellen Ableitstrom ermöglichen und so die ADC-Eingänge schützen. [35]
Die für die Schottky-Beschaltung benötigte ± 12 -V-Schiene wird über zwei Linearregler aus der höheren Versorgung erzeugt (vgl. Abbildung 6.2 und 6.3).

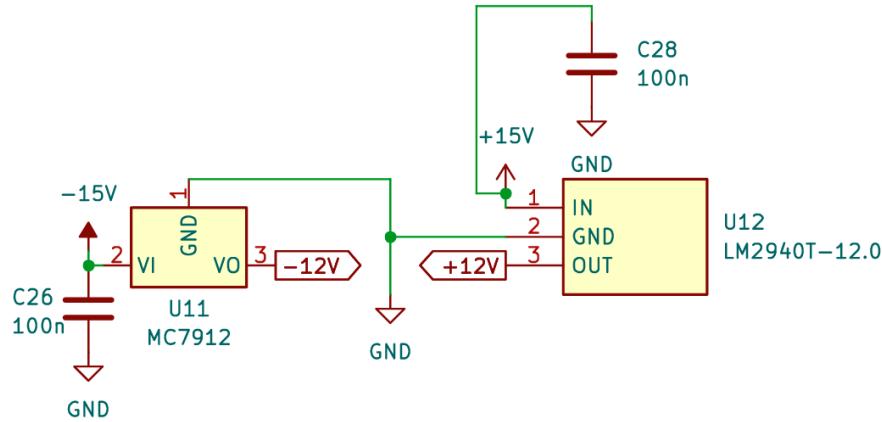


Abbildung 6.2: Leiterplatte - Erzeugung ± 12 V

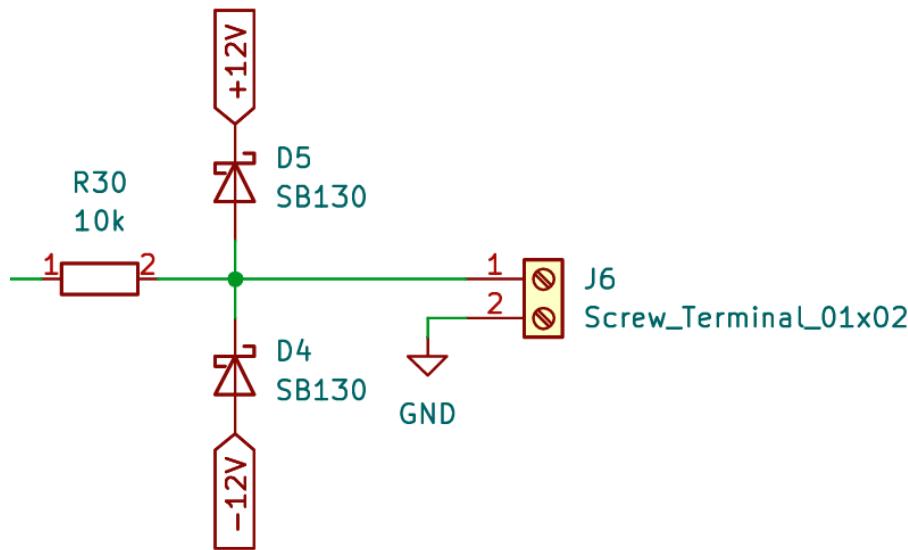


Abbildung 6.3: Leiterplatte - Schutzbeschaltung Eingänge

Die Referenzspannung für präzise AD-Wandlung beträgt 2,5 V und wird über ein dediziertes Referenz-IC erzeugt. Zur Eliminierung der Lastabhängigkeit wird diese Referenz über einen Impedanzwandler (OPA551) gepuffert, sodass die Referenzquelle niederohmig und stabil bleibt (siehe Abbildung 6.4). [36]

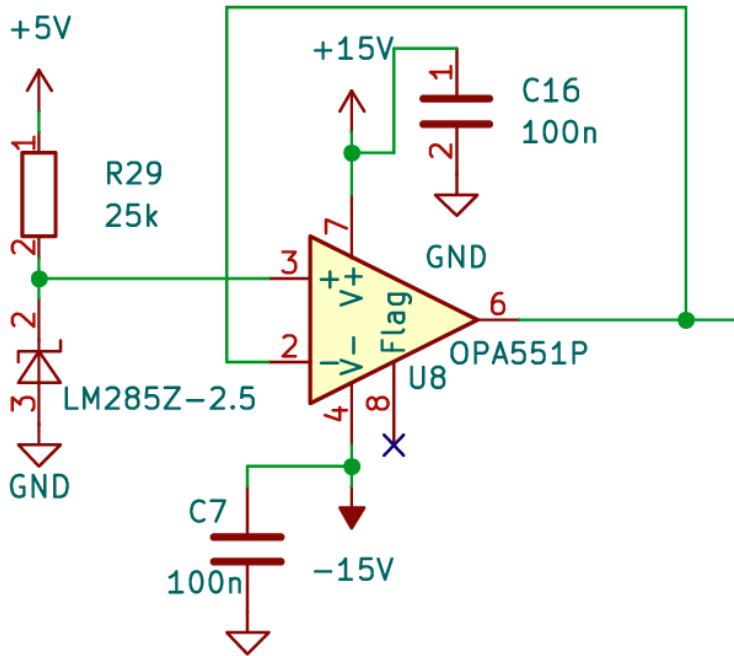


Abbildung 6.4: Leiterplatte - Erzeugung Referenzspannung 2,5 V

Sämtliche externen Anschlüsse wurden als robuste Schraubklemmen im Schaltplan vorgesehen, sowohl auf der Platine selbst als auch an den externen Anschlussstellen am Gehäuse (Abbildung 6.5).

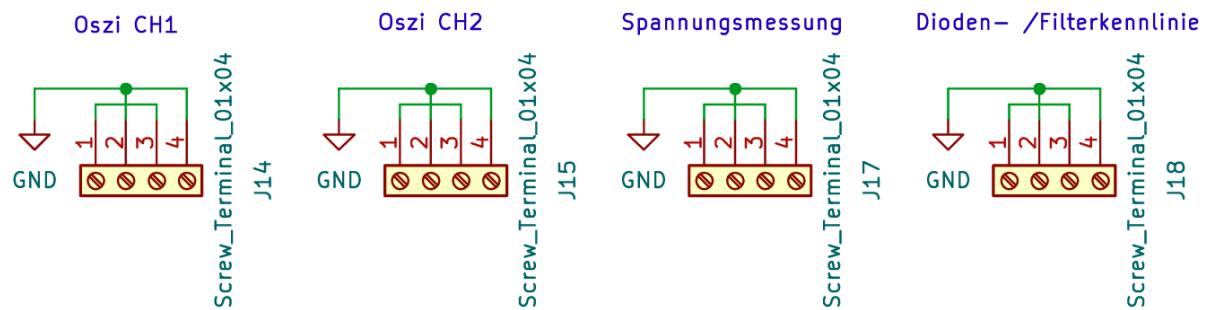


Abbildung 6.5: Leiterplatte - Anschlüsse für Schnittstellen

7 Reale Messungen HW – Vergleich zu Simulation

Reale Messungen konnten zeitlich für alle entwickelten Schaltung auf dem Steckbrett durchgeführt werden. Ausnahmen bilden nur die neue Strommessung vom Multimeter sowie die Verstärkerschaltung am Funktionsgenerator. Hier konnte nur das Funktionsgenerator-Modul mit dem AD9833 aufgezeichnet werden.

Die Ergebnisse der Messungen werden im Folgenden dargestellt.

7.1 Spannungsmessung

7.1.1 Gleichspannung

Zur Überprüfung der Genauigkeit des entwickelten OurDAQ-Systems wurde die gemessene Gleichspannung mit den Referenzwerten eines Labor-Multimeters (METRA Hit 16S) verglichen. Das Multimeter besitzt bei DC-Spannungsmessung in diesem Messbereich eine Grundgenauigkeit von typ. \pm (0,1 % vom Messwert + 1 Digit). [37]

Die nachfolgenden Tabelle 7.1 zeigt die Vergleichswerte für positive und negative Spannungen, jeweils eingestellt über ein Labornetzteil.

Multimeter [V]	OurDAQ [V]
0,068	0,064
0,878	0,874
1,725	1,710
2,052	2,041
03,22	03,22
04,28	04,29
05,59	05,60
05,96	05,97
06,90	06,91
08,05	08,06
08,46	08,46
09,51	09,51
-0,054	-0,058
-0,941	-0,941
-1,748	-1,751
-02,82	-02,83
-03,28	-03,29

-05,16	-05,15
-05,62	-05,62
-06,37	-06,38
-07,07	-07,07
-07,75	-07,75
-08,01	-08,01
-08,75	-08,74
-09,57	-09,57

Tabelle 7.1: Multimeter - reale Messergebnisse DC

- Die Abweichungen zwischen Multimeter und OurDAQ liegen im Bereich weniger Millivolt
- Beispiel: bei 1,725 V beträgt die Differenz 15 mV, das entspricht ca. 0,9 %
- Bei höheren Spannungen (z. B. 9,51 V) ist praktisch kein Unterschied mehr erkennbar (<1 mV)
- Für negative Spannungen zeigt sich ein vergleichbares Verhalten

Damit wird bestätigt, dass das OurDAQ-System im Spannungsbereich von ± 10 V sehr präzise DC-Messungen liefert.

7.1.2 Wechselspannung

Zur Überprüfung der Genauigkeit des OurDAQ-Systems wurde die Messung von Wechselspannungen mit einem Labor-Multimeter (METRA Hit 16S) verglichen.

Als Signalquelle diente ein Funktionsgenerator, der Sinus-, Dreieck- und Rechtecksignale mit einer Spitze-Spitze-Spannung von 1 V (Amplitude ca. 0,5 V) erzeugte.

Hinweis: In der OurDAQ-Software wird der Effektivwert berechnet, wodurch eine leichte Schwankung auftreten kann. Die Abbildung 7.1 zeigt einen typischen Verlauf bei einem Sinussignal mit 5 kHz.

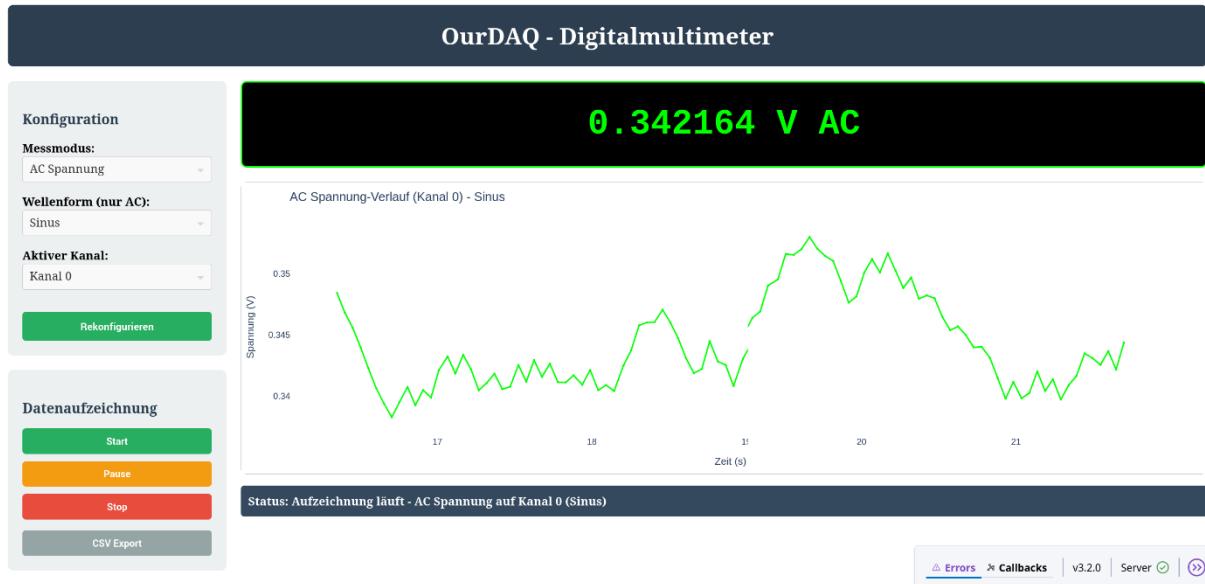


Abbildung 7.1: OurDAQ-SW - Messung Sinus (Amplitude = 0,5 V, f = 1 kHz)

Das Multimeter besitzt bei AC-Spannungsmessung in diesem Messbereich eine Grundgenauigkeit von typ. $\pm (0,75\% \text{ vom Messwert} + 3 \text{ Digit})$. Beim Multimeter ergeben sich durch die Genauigkeit folgende maximalen Messabweichungen: [37]

$$\Delta U = 0,0075 * U_{\text{Messwert}} + 0,003 V$$

Bei 0,347 V (10 kHz, Sinus) ergibt sich:

$$\Delta U = 0,0075 * 0,346 V + 0,003 V \approx 5,6 mV$$

Das heißt, Abweichungen bis ca. $\pm 6 \text{ mV}$ sind innerhalb der Spezifikation vom Multimeter. Somit sollte unser OurDAQ den Messwert vom Multimeter nicht um $\pm 6 \text{ mV}$ überschreiten.

Alle Messwerte sind in den Tabellen 7.2 bis 7.4 als Effektivwerte (RMS) dargestellt.

f [Hz]	Multimeter [V]	OurDAQ [V]	Differenz [mV]
10	0,346 V	0,350 V	4,0
500	0,348 V	0,351 V	3,0
1000	0,347 V	0,350 V	3,0
5000	0,343 V	0,345 V	2,0
10000	0,341 V	0,343 V	2,0
30000	0,339 V	0,337 V	-2,0
50000	0,334 V	0,334 V	0,0
100000	0,315 V	0,288 V	-27

Tabelle 7.2: Multimeter - reale Messergebnisse AC (Sinus)

f [Hz]	Multimeter [V]	OurDAQ [V]	Differenz [mV]
10	0,282 V	0,277 V	-5,0
500	0,285 V	0,284 V	-1,0
1000	0,285 V	0,290 V	5,0
5000	0,281 V	0,285 V	4,0
10000	0,280 V	0,276 V	-4,0
30000	0,278 V	0,275 V	-3,0
50000	0,278 V	0,270 V	-8,0
100000	0,256 V	0,222 V	-34

Tabelle 7.3: Multimeter - reale Messergebnisse AC (Dreieck)

f [Hz]	Multimeter [V]	OurDAQ [V]	Differenz [mV]
10	0,486 V	0,487 V	1,0
500	0,486 V	0,487 V	1,0
1000	0,484 V	0,484 V	0,0
5000	0,476 V	0,479 V	3,0
10000	0,471 V	0,472 V	1,0
30000	0,454 V	0,449 V	-5,0
50000	0,437 V	0,433 V	-4,0
100000	0,392 V	0,361 V	-31

Tabelle 7.4: Multimeter - reale Messergebnisse AC (Rechteck)

Fazit AC-Messungen

- OurDAQ stimmt in allen Signalformen zwischen 500 Hz und 30 kHz sehr gut mit dem METRA Hit 16S überein
- Niedrige Frequenzen (10 Hz): Leichte Abweichungen durch Software-RMS-Berechnung
- Hohe Frequenzen (>50 kHz): OurDAQ-Abtastrate limitiert die Genauigkeit, deutliche Differenzen
- Signalformabhängigkeit: Rechteck- und Dreiecksignale reagieren stärker auf Sampling-Grenzen
- Optimierungspotenzial: Software kann noch RMS-Algorithmen verbessern, um hohe Frequenzen präziser zu messen

Für den Arbeitsbereich 500 Hz bis 30 kHz kann für das OurDAQ-System eine konservative Worst-Case-Genauigkeitsangabe abgeleitet werden, die die Unsicherheit des Referenzmessgerätes (METRA Hit 16S) mit einbezieht. Aus den Messungen ergibt sich:

$$\Delta U_{OurDAQ,max} \approx \pm(max. 3,6\% vom Messwert + max. 11,4 mV)$$

7.2 Netzteil

Für die Überprüfung des Labornetzteils wurde die eingestellte Sollspannung (Softwarevorgabe) mit den realen Ausgangswerten des Netzteils verglichen. Als Referenzmessgerät diente das Multimeter METRA Hit 16S.

Zur Validierung wurden Messungen mit zwei unterschiedlichen Lastwiderständen durchgeführt.

In Tabelle 7.5 sind die Messwerte für das positive Netzteil bei einem Lastwiderstand von 1 kΩ zu sehen, in Tabelle 7.6 die entsprechenden Werte bei 10 kΩ.

Vorgabe Software [V]	Netzeil_plus + Multimeter [V]
00,00	0,001
01,00	1,002
02,00	2,001
03,00	3,001
04,00	04,00
05,00	04,99
06,00	05,99
07,00	06,99
08,00	07,99
09,00	09,00
10,00	10,00

Tabelle 7.5: Netzteil positiv - reale Messergebnisse (1 kΩ Last)

Vorgabe Software [V]	Netzeil_plus + Multimeter [V]
00,00	0,005
01,00	0,998
02,00	1,997
03,00	02,99
04,00	03,99
05,00	04,98
06,00	05,99

07,00	06,98
08,00	07,99
09,00	08,99
10,00	09,99

Tabelle 7.6: Netzteil positiv - reale Messergebnisse (10 kΩ Last)

- Die gemessenen Spannungen stimmen in beiden Fällen sehr gut mit den Sollwerten überein
- Die maximale Abweichung liegt im Bereich 1–10 mV
- Zwischen den Lastfällen (1 kΩ vs. 10 kΩ) zeigen sich wenig Unterschiede

Für das negative Netzteil wurden die gleichen Messungen durchgeführt. Auch hier wurde die eingestellte Sollspannung (Softwarevorgabe) mit den am Ausgang gemessenen Werten verglichen.

Als Last wurden ebenfalls 1 kΩ (Tabelle 7.7) sowie 10 kΩ (Tabelle 7.8) verwendet.

Vorgabe Software [V]	Netzeil_minus + Multimeter [V]
00,00	00,00
-01,00	-1,018
-02,00	-2,088
-03,00	-02,98
-04,00	-04,00
-05,00	-05,14
-06,00	-05,88
-07,00	-07,05
-08,00	-07,91
-09,00	-08,88
-10,00	-10,02

Tabelle 7.7: Netzteil negativ - reale Messergebnisse (1 kΩ Last)

Vorgabe Software [V]	Netzeil_minus + Multimeter [V]
00,00	-0,001
-01,00	-0,996

-02,00	-1,992
-03,00	-2,996
-04,00	-03,99
-05,00	-05,02
-06,00	-05,97
-07,00	-06,95
-08,00	-07,99
-09,00	-09,02
-10,00	-09,91

Tabelle 7.8: Netzteil negativ - reale Messergebnisse ($10 \text{ k}\Omega$ Last)

- Bei negativer Ausgangsspannung zeigen sich größere Abweichungen im Vergleich zur positiven Versorgung
- Im Bereich von -2 V bis -6 V treten Differenzen von bis zu ca. 140 mV auf
- In den höheren Bereichen (-8 V bis -10 V) liegt die Genauigkeit wieder ähnlich wie beim positiven Zweig, mit Abweichungen um etwa 100 mV
- Die Stabilität der Versorgung ist insgesamt ausreichend für die Messanwendungen, auch wenn die negative Spannung tendenziell eine etwas größere Toleranz aufweist

7.3 Oszilloskop

Die Abbildungen (7.2 bis 7.5) zeigen die Darstellung eines sinusförmigen Eingangssignals bei verschiedenen Frequenzen (1 kHz, 2 kHz, 5 kHz und 10 kHz).

Bis etwa 3 kHz ist der Signalverlauf in der Software nahezu ideal und ohne sichtbare Abweichungen.

Ab einer Frequenz von ca. 4–5 kHz treten erste leichte Verzerrungen auf. Diese werden mit steigender Frequenz deutlicher sichtbar, was auf die begrenzte Abtastrate des Systems sowie die internen Signalverarbeitungsroutinen zurückzuführen ist.

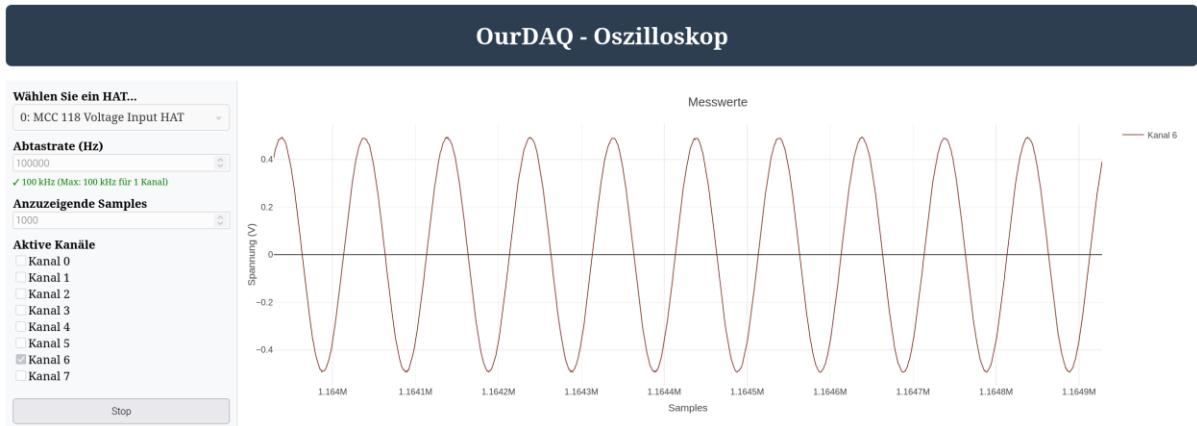


Abbildung 7.2: Messung einer 1 kHz Sinus-Spannung (symmetrisch)

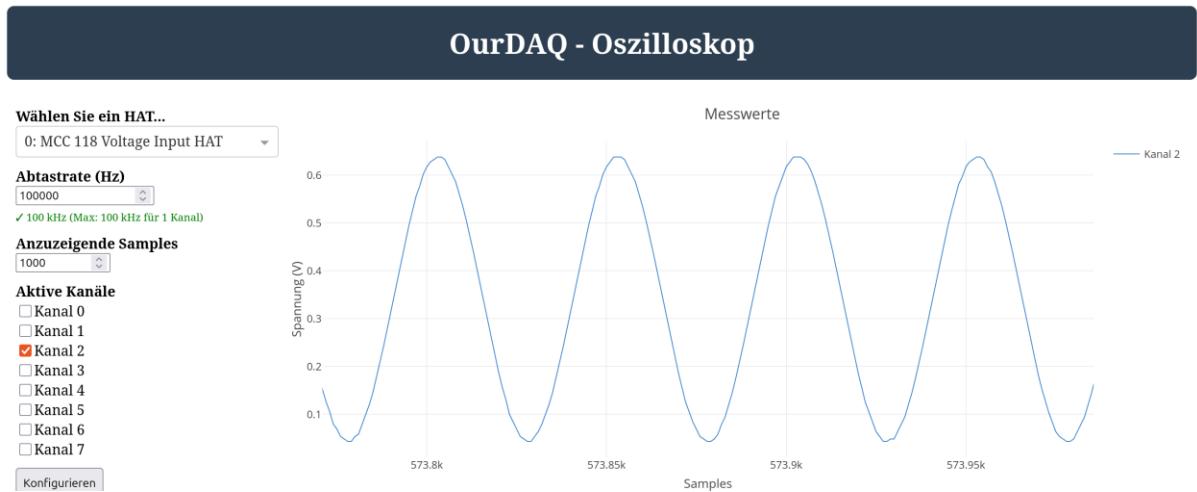


Abbildung 7.3: Messung einer 2 kHz Sinus-Spannung (asymmetrisch)

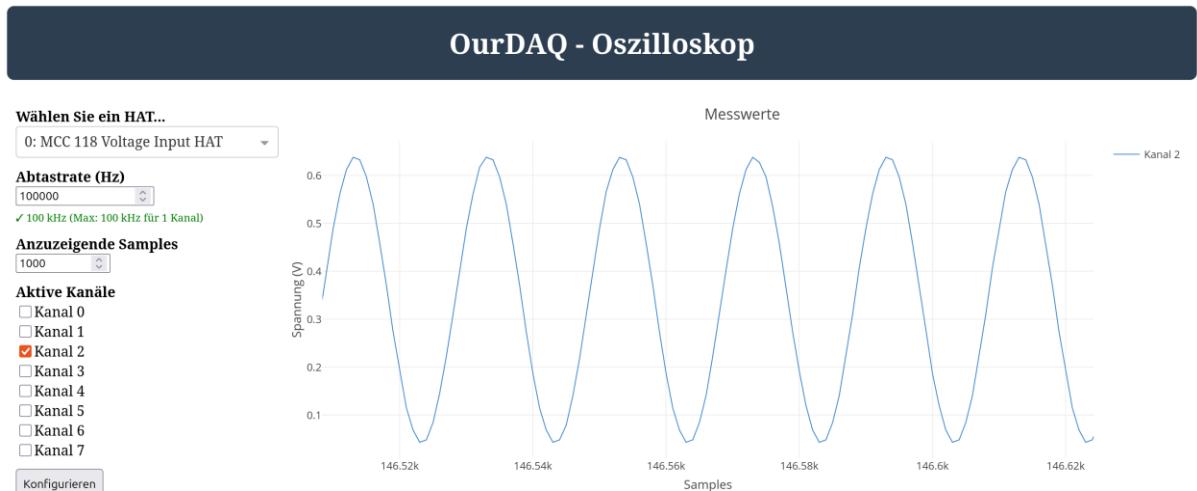


Abbildung 7.4: Messung einer 5 kHz Sinus-Spannung (asymmetrisch)

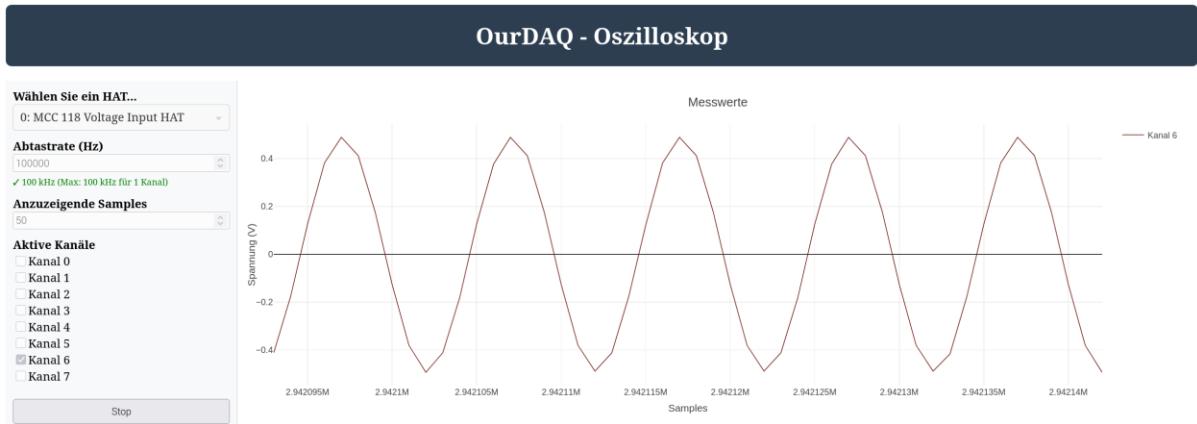


Abbildung 7.5: Messung einer 10 kHz Sinus-Spannung (symmetrisch)

Die Abbildungen (7.6 bis 7.9) zeigen die Darstellung eines Dreiecksignals bei verschiedenen Frequenzen (500 Hz, 2 kHz, 4 kHz und 6 kHz).

Bis zu einer Frequenz von etwa 1–2 kHz ist der Signalverlauf in der Software noch gut erkennbar und nähert sich einer idealen Dreiecksform an.

Ab einer Frequenz von ca. 3–4 kHz treten jedoch deutliche Verzerrungen auf. Die Spitzen des Dreiecks werden zunehmend abgerundet, und die linearen Anstiegs- und Abfallflanken verlieren ihre Geradlinigkeit.

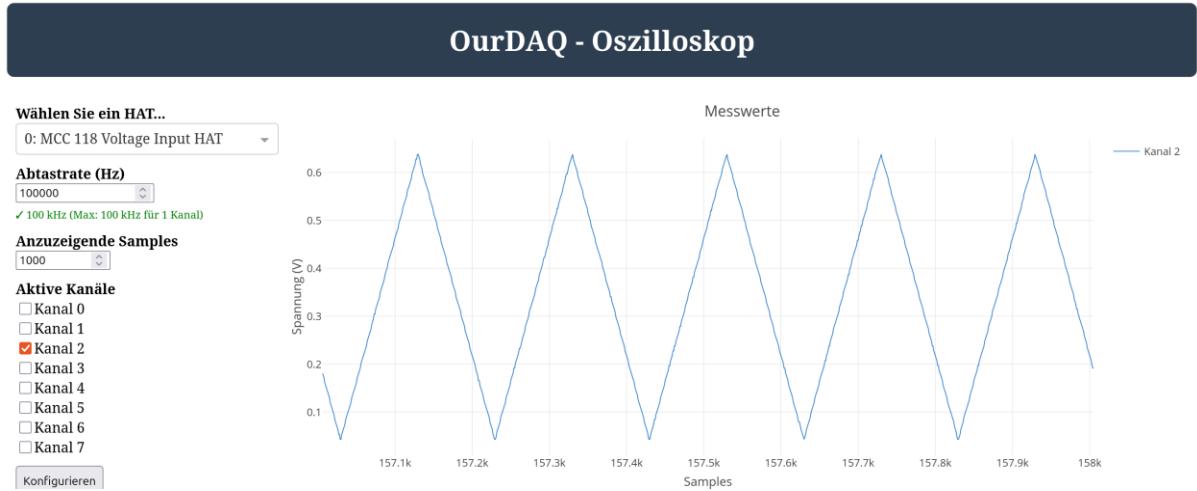


Abbildung 7.6: Messung einer 500 Hz Dreiecksspannung (asymmetrisch)

OurDAQ - Oszilloskop

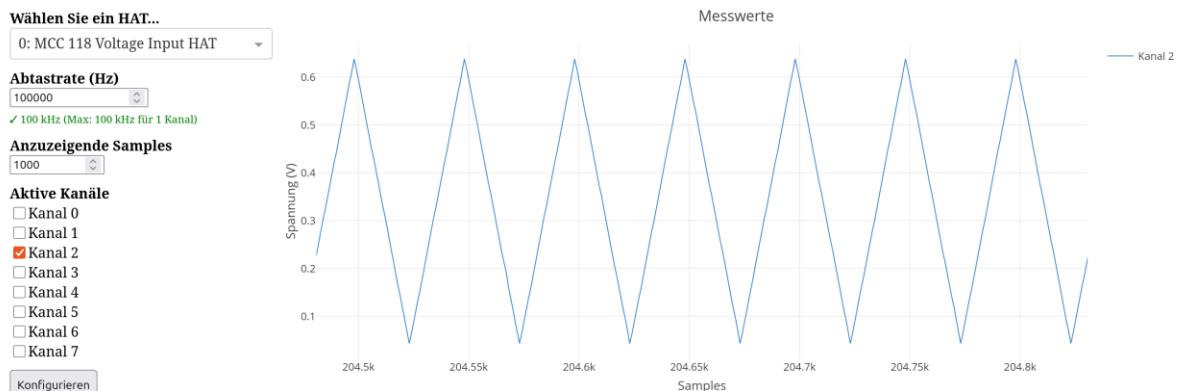


Abbildung 7.7: Messung einer 2 kHz Dreiecksspannung (asymmetrisch)

OurDAQ - Oszilloskop

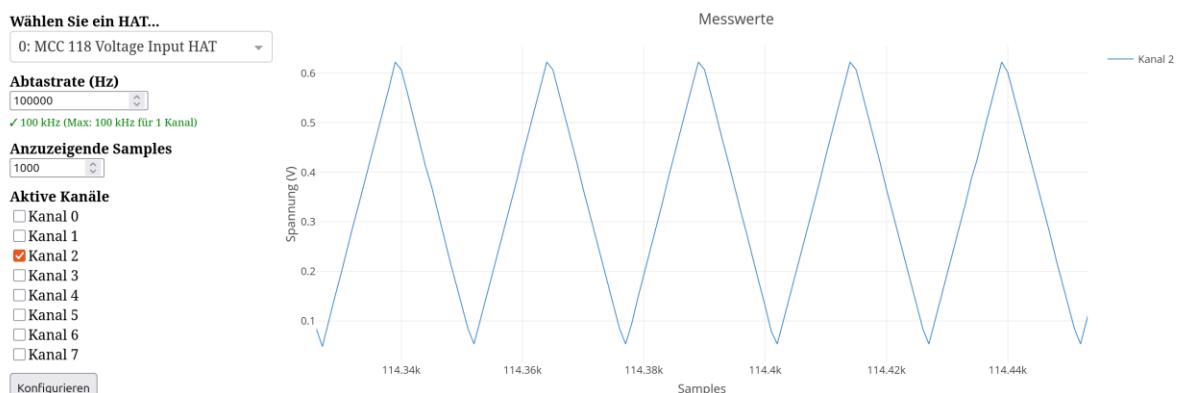


Abbildung 7.8: Messung einer 4 kHz Dreiecksspannung (asymmetrisch)

OurDAQ - Oszilloskop

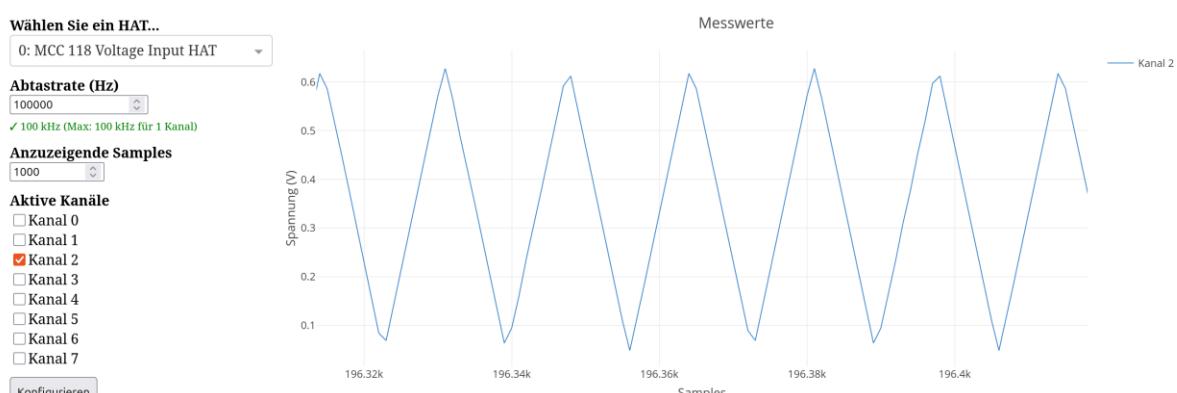


Abbildung 7.9: Messung einer 6 kHz Dreiecksspannung (asymmetrisch)

Die Abbildungen (7.10 bis 7.13) zeigen die Darstellung eines Rechtecksignals bei verschiedenen Frequenzen (500 Hz, 2 kHz, 4 kHz und 6 kHz).

Bereits bei niedrigen Frequenzen (z.B. 2 kHz) sind erste Artefakte sichtbar. Obwohl die Grundform erkennbar ist, zeigen sich an den steilen Flanken leichte Überschwinger und eine sichtbare Anstiegs- und Abfallzeit.

Diese Effekte werden mit steigender Frequenz rapide stärker. Die Anstiegs- und Abfallflanken werden deutlich flacher, und die Überschwinger an den Ecken nehmen zu. Ein Rechtecksignal benötigt theoretisch unendlich hohe Frequenzen, um seine perfekt senkrechten Flanken darzustellen. Aufgrund der begrenzten Abtastrate und der Bandbreite der Signalverarbeitung kann das System diese hohen Frequenzanteile nicht korrekt reproduzieren. Bei Frequenzen wie 6 kHz ist die ursprüngliche Rechteckform kaum noch zu erkennen.

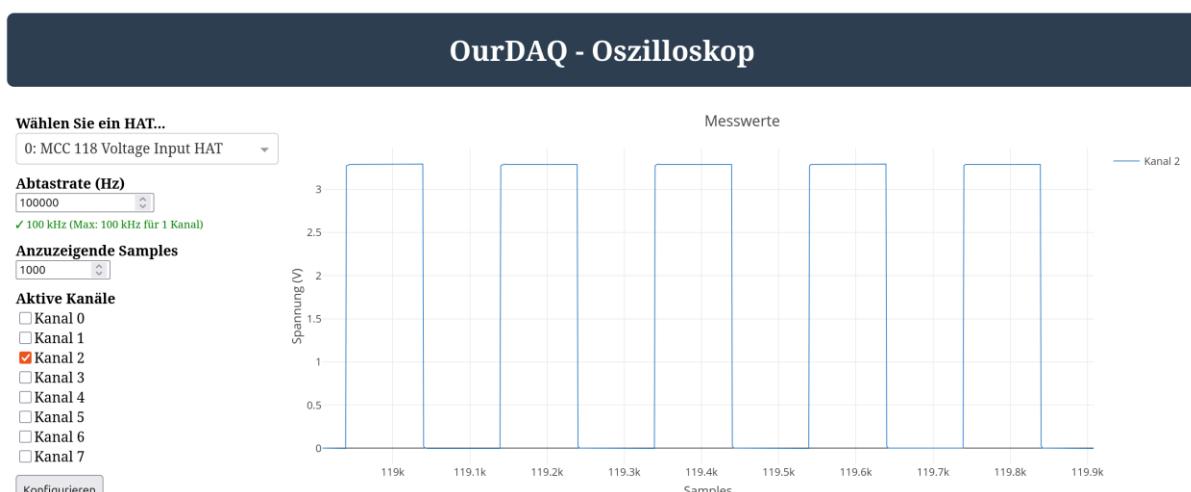


Abbildung 7.10: Messung einer 500 Hz Rechteckspannung (asymmetrisch)

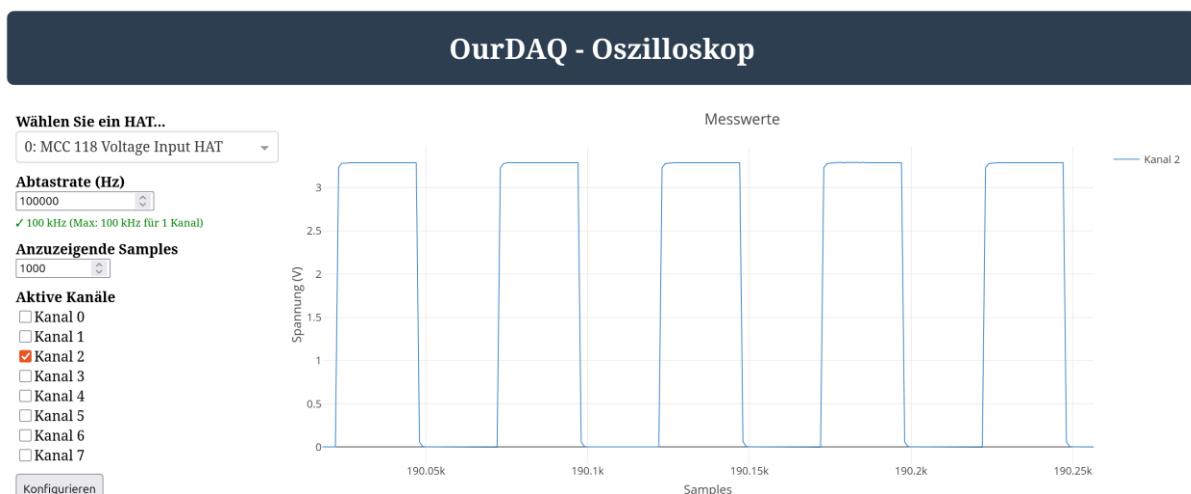


Abbildung 7.11: Messung einer 2 kHz Rechteckspannung (asymmetrisch)



Abbildung 7.12: Messung einer 4 kHz Rechteckspannung (asymmetrisch)

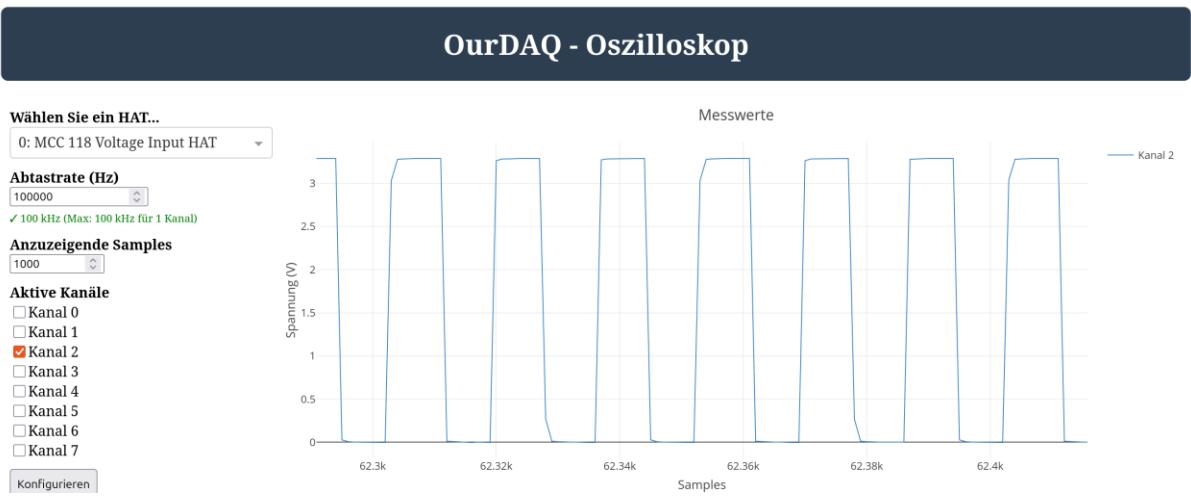


Abbildung 7.13: Messung einer 6 kHz Rechteckspannung (asymmetrisch)

7.4 Funktionsgenerator

Die Signalformen des verwendeten Funktionsgenerators-Moduls wurden am Oszilloskop überprüft.

- Sinus: Die Abbildungen 7.14 und 7.15 zeigen die aufgenommenen Sinus-Signale
- Dreieck: Die Abbildungen 7.16 und 7.17 zeigen die aufgenommenen Dreieck-Signale
- Rechteck: Die Abbildungen 7.18 und 7.19 zeigen die aufgenommenen Rechteck-Signale

Mit den am Oszilloskop verwendeten Cursorn wurde jeweils überprüft, ob die eingestellte Frequenz mit der gemessenen übereinstimmt.

Das Ergebnis zeigt, dass die Signale frequenzgenau und stabil dargestellt werden.

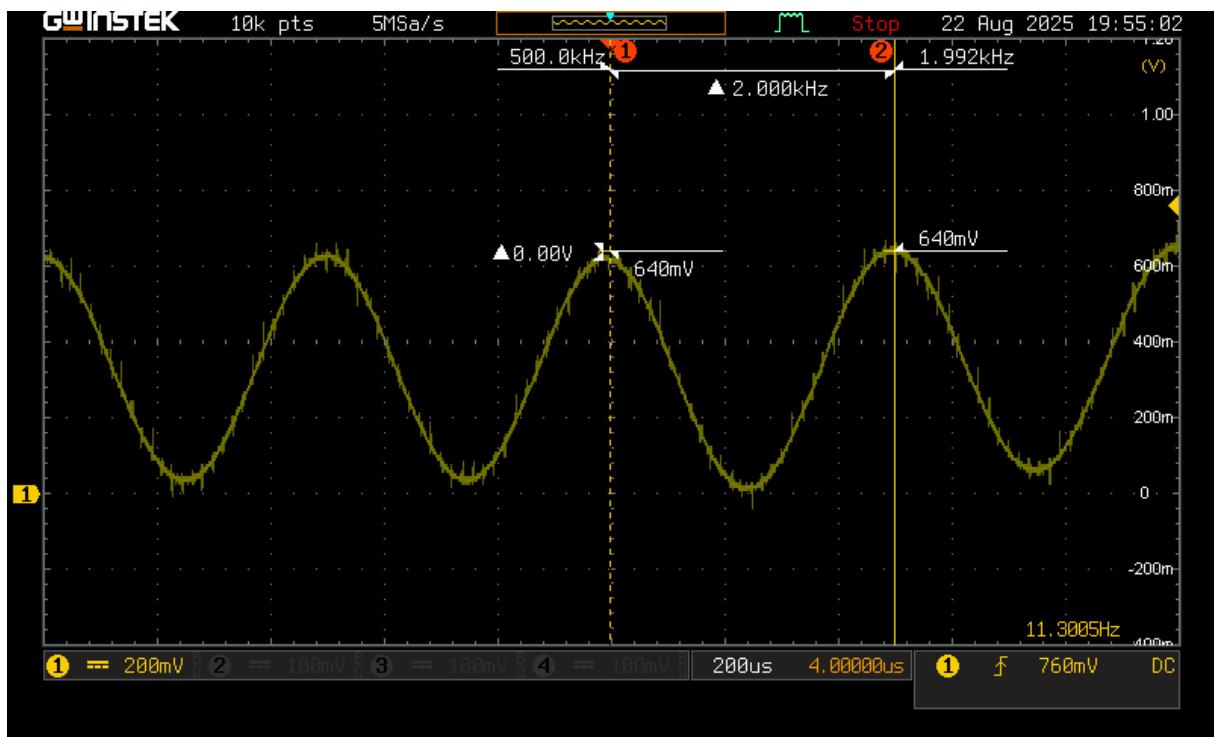


Abbildung 7.14: Funktionsgenerator - Sinus (2 kHz)

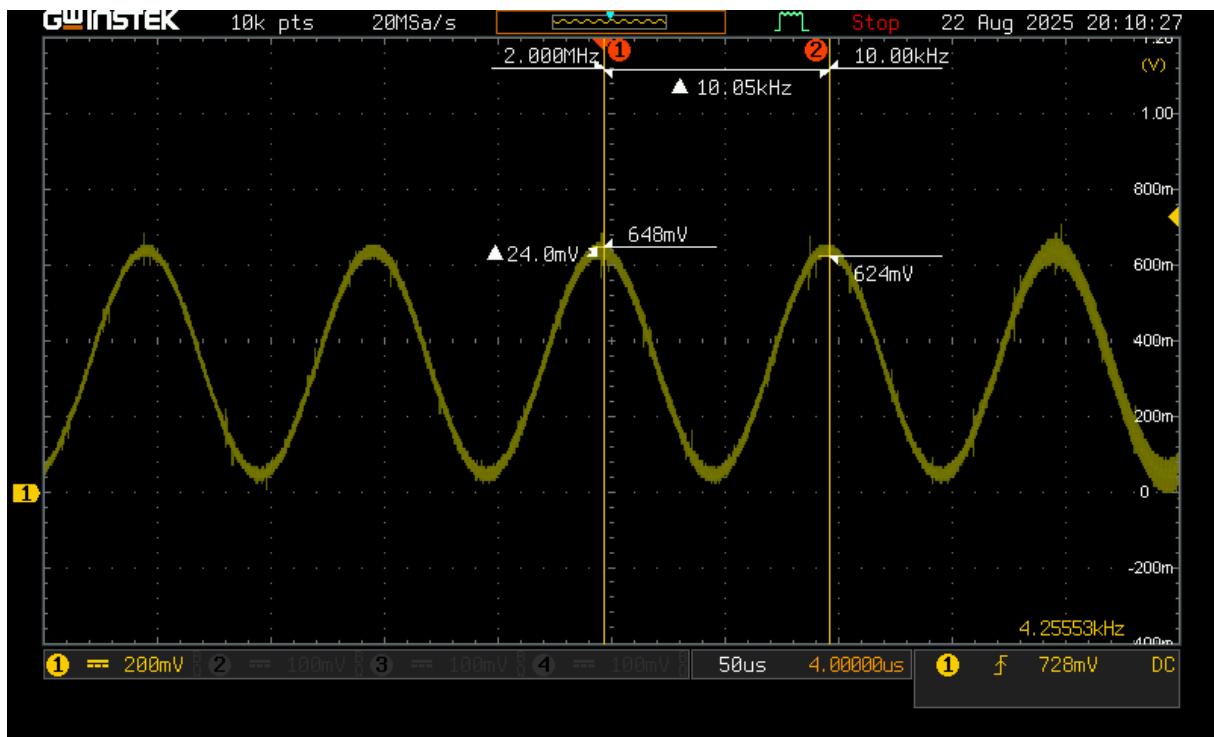


Abbildung 7.15: Funktionsgenerator - Sinus (10 kHz)

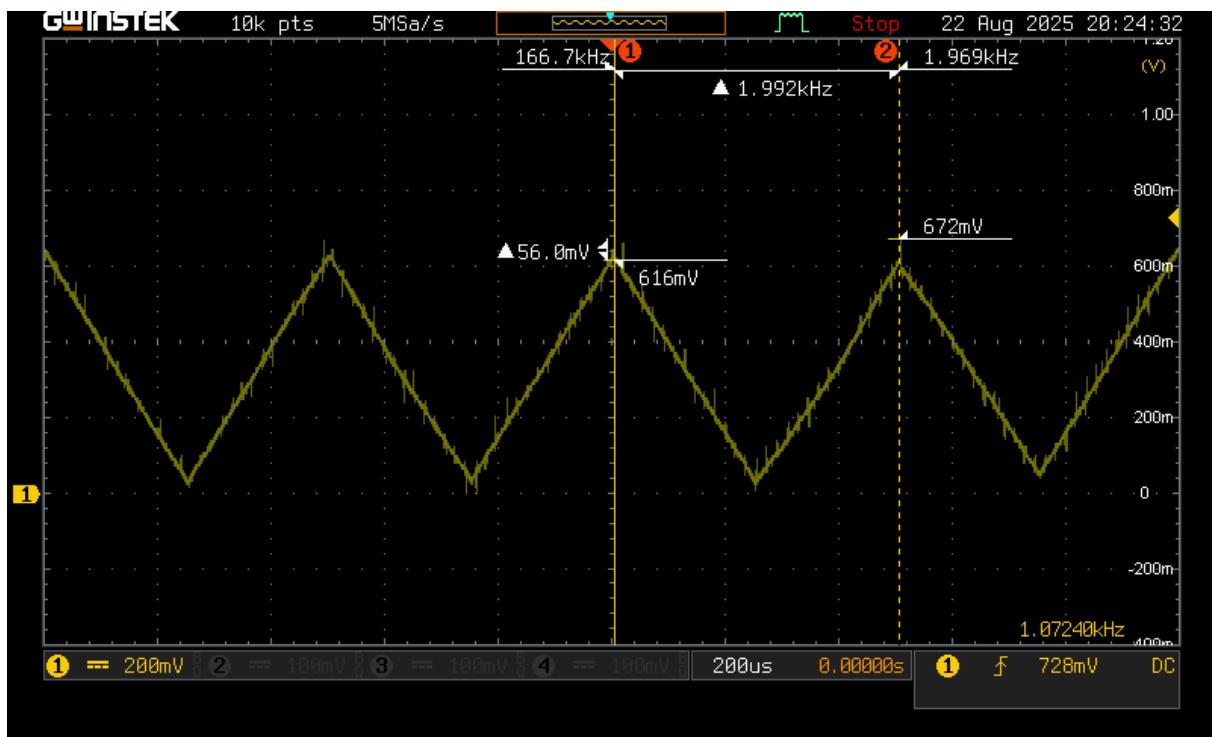


Abbildung 7.16: Funktionsgenerator - Dreieck (2 kHz)

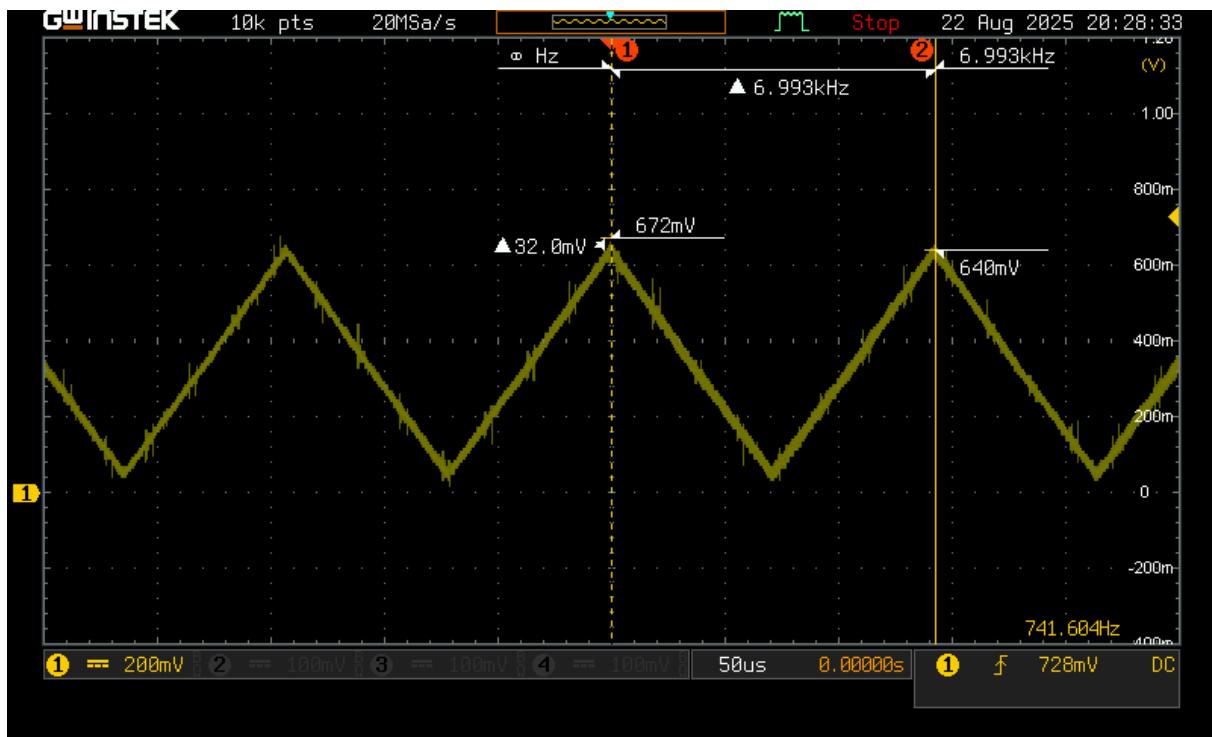


Abbildung 7.17: Funktionsgenerator - Dreieck (7 kHz)

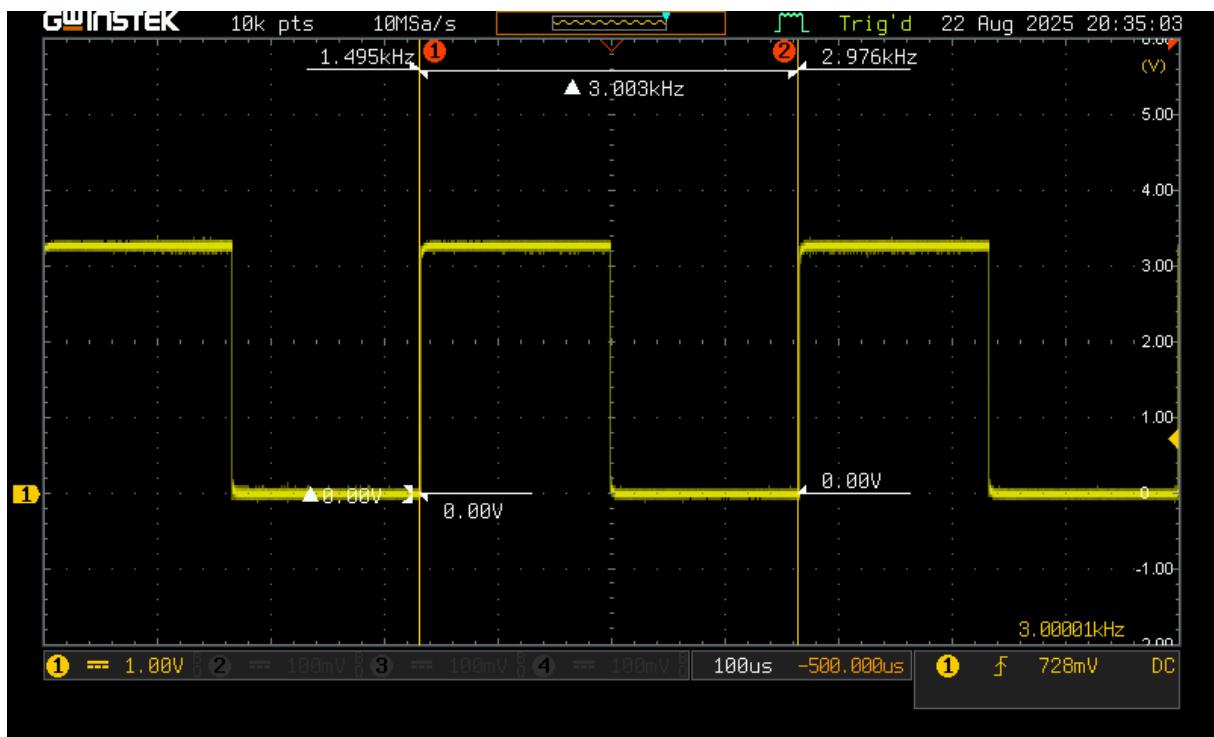


Abbildung 7.18: Funktionsgenerator - Rechteck (3 kHz)

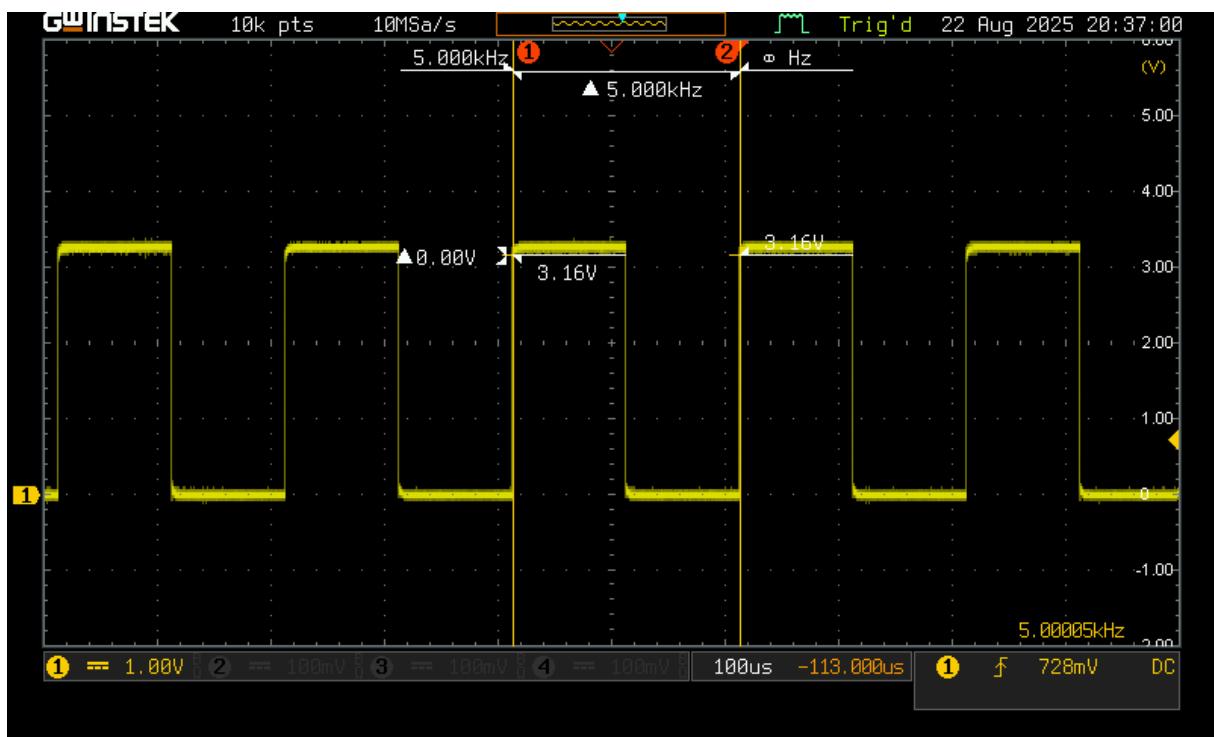


Abbildung 7.19: Funktionsgenerator - Rechteck (5 kHz)

Wie bereits erwähnt, war es nicht mehr möglich, Messergebnisse mit der Verstärkerschaltung für den Funktionsgenerator aufzunehmen.

7.5 Diodenkennlinie

Für die Charakterisierung der Schottky-Diode SR130 wurde eine Messschaltung mit einem Serienwiderstand von $1\text{ k}\Omega$ (Toleranz 0,1 %) verwendet. Ziel der Messung war die Aufnahme der Spannungs-Strom-Kennlinie der Diode über einen weiten Spannungsbereich.

Messaufbau und Vorgehensweise

- **Serienwiderstand:** $1\text{ k}\Omega$, 0,1 % Toleranz
- **Diode:** SR130 Schottky-Diode
- **Messpunkte:** Zwei Datensätze mit 15 bzw. 30 Messpunkten
- **Messgrößen:**
 - Spannung über die Diode, direkt mit Multimeter und Software erfasst
 - Strom durch die Diode, berechnet aus Spannung über Serienwiderstand bzw. direkt gemessen

Die Gesamtspannung U_{ges} wurde über das Labornetzteil eingestellt. Der Strom durch die Diode wurde nach folgender Formel berechnet:

$$I = \frac{U_{ges} - U_D}{R}$$

Die Messergebnisse sind in Tabelle 7.9 und 7.10 dargestellt.

Spannung Netzteil [V]	Spannung Diode Multimeter [V]	Strom Multimeter [mA]	Gemessene Spannung Diode SW [V]	Strom SW [mA]
0,04023	0,040	0,001	0,038	0,002
1,47338	0,292	1,187	0,289	1,184
2,95260	0,409	2,549	0,407	2,546
4,42699	0,541	3,900	0,514	3,913
5,89055	0,621	5,400	0,616	5,274
7,35441	0,729	6,600	0,724	6,631
8,81828	0,828	8,000	0,821	7,997
10,29237	0,924	9,300	0,918	9,374
10,32308	0,926	9,400	0,918	9,405
10,32308	0,926	9,400	0,918	9,405
10,32308	0,926	9,400	0,918	9,400
10,32308	0,926	9,400	0,918	9,405
10,32308	0,926	9,400	0,918	9,405
10,32308	0,926	9,400	0,918	9,405

Tabelle 7.9: Diodenkennlinie - reale Messergebnisse (15 Punkte)

Spannung Netzteil [V]	Spannung Diode Multimeter [V]	Strom Multimeter [mA]	Gemessene Spannung Diode SW [V]	Strom SW [mA]
0,04023	0,040	0,001	0,038	0,002
0,71586	0,220	0,497	0,217	0,498
1,42731	0,288	1,141	0,284	1,143
2,13365	0,346	1,794	0,345	1,788
2,85023	0,401	2,454	0,402	2,449
3,57192	0,456	3,100	0,453	3,119
4,27826	0,507	3,800	0,504	3,774
4,97948	0,558	4,400	0,555	4,424
5,69094	0,608	5,500	0,601	5,090
6,39216	0,658	5,700	0,652	5,740
7,10361	0,709	6,040	0,703	6,400
7,81507	0,764	7,000	0,754	7,061
8,51629	0,810	7,700	0,806	7,711
9,21751	0,858	8,300	0,852	8,366
9,92897	0,906	9,000	0,903	9,026
10,31285	0,933	9,400	0,923	9,390
10,31285	0,933	9,400	0,928	9,385
10,32308	0,933	9,400	0,928	9,395
10,31796	0,933	9,400	0,928	9,390
10,31796	0,933	9,400	0,923	9,395
10,31796	0,933	9,400	0,918	9,400
10,31796	0,933	9,400	0,918	9,400
10,32308	0,933	9,400	0,918	9,405
10,32308	0,933	9,400	0,913	9,410
10,31796	0,933	9,400	0,913	9,405
10,31796	0,933	9,400	0,913	9,405
10,32308	0,933	9,400	0,913	9,410
10,32308	0,933	9,400	0,913	9,410
10,31796	0,933	9,400	0,913	9,405

Tabelle 7.10: Diodenkennlinie - reale Messergebnisse (30 Punkte)

Die Messungen zeigen hierbei den typischen Verlauf einer Schottky-Diode (vgl. Abbildungen 7.20 und 7.21). Bei sehr kleinen Spannungen (<0,1 V) ist der Strom noch extrem gering und ab ca. 0,2-0,3 V steigt der Strom stark exponentiell an. Bei höheren Spannungen (>0,9 V) erreicht die Diode Ströme im Bereich von 9-10 mA, die sich nahezu konstant verhalten, da die Diode im Leitbereich ist.

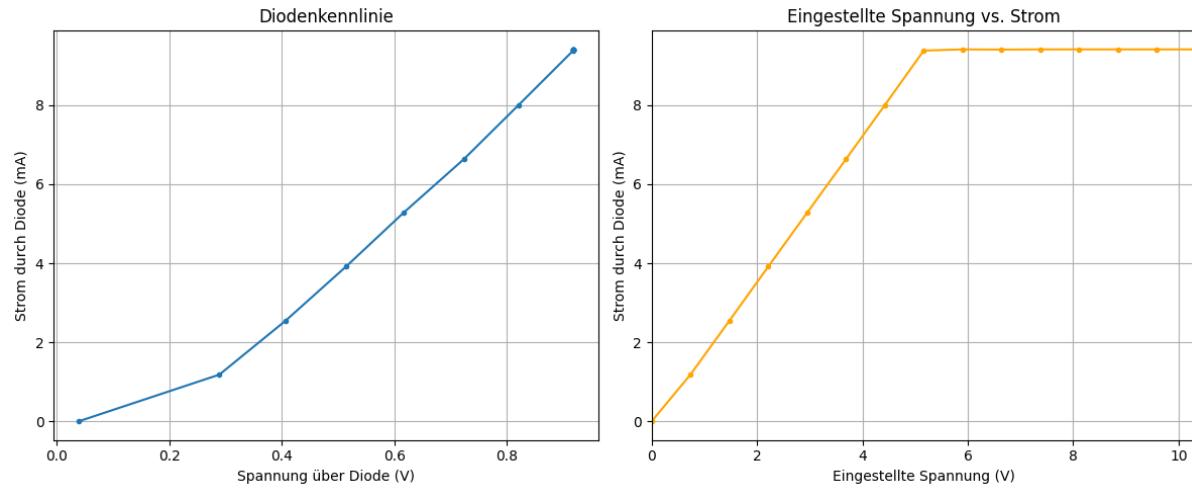


Abbildung 7.20: OurDAQ-SW - aufgenommene Diodenkennlinie (15 Punkte)

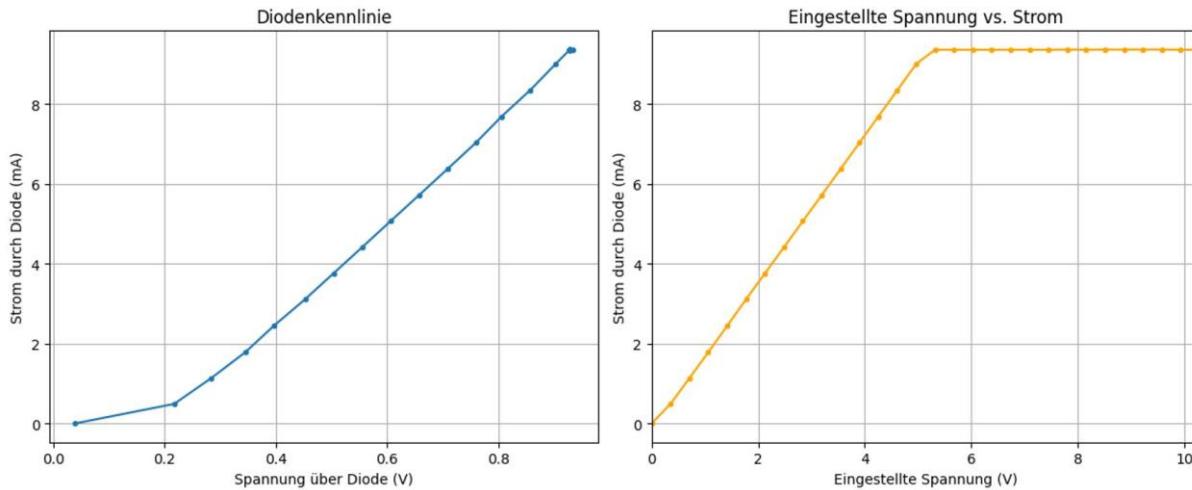


Abbildung 7.21: OurDAQ-SW - aufgenommene Diodenkennlinie (30 Punkte)

Vergleich Theorie und Praxis bei Diodenmessungen

Die realen Messungen der Diodenkennlinie zeigen eine sehr gute Übereinstimmung zwischen dem OurDAQ-System und dem Labor-Multimeter. Selbst bei kleinen Strömen im mA-Bereich weichen die Werte kaum voneinander ab.

Die zuvor im Kapitel 4.6.1 dargestellten hohen relativen Fehler bei sehr kleinen Strömen beziehen sich auf ein Worst-Case-Szenario, das alle möglichen Fehlerquellen (Spannungsfehler des MCC 118 und Serienwiderstandstoleranz) gleichzeitig in ungünstiger Weise kombiniert. In der Realität ist dieses Szenario extrem unwahrscheinlich.

Die eingesetzten Serienwiderstände besitzen nur 0,1 % Toleranz, was die Beiträge aus Widerstandstoleranz sehr klein hält. Das MCC 118 arbeitet mit präziser Spannungsmessung, Offset- und Gain-Fehler fallen in der Praxis deutlich geringer aus.

Damit müssten praktisch alle Fehler gleichzeitig maximal auftreten, um den theoretischen Worst-Case-Fehler zu erreichen - ein äußerst unwahrscheinliches Ereignis.

Fazit: Während die theoretische Abschätzung konservativ sehr hohe relative Fehler bei kleinen Strömen zeigt, bewahrheitet sich dies in den realen Messungen nicht. Die tatsächlichen Abweichungen zwischen OurDAQ und Multimeter sind klein, wodurch die Messergebnisse als zuverlässig und präzise gelten.

7.6 Filterkennlinie

Für die Untersuchung der RC-Filterkennlinie wurde ein Tiefpassfilter bestehend aus $R = 1 \text{ k}\Omega$ (0,1 % Toleranz) und $C = 100 \text{ nF}$ (10 % Toleranz) verwendet. Das Eingangssignal war ein Sinus mit 1 Vpp (Amplitude ca. 0,5 V).

Die Messwerte wurden sowohl mit Multimeter, Oszilloskop (Tabelle 7.11 und Abbildung 7.22) als auch über das OurDAQ-System aufgenommen. Zusätzlich wurde eine LTspice-Simulation (Tabelle 7.12) zur Vergleichbarkeit durchgeführt.

$f [\text{Hz}]$	U_{ein} Multim. [V]	U_{aus} Multim. [V]	$U_{\text{ein}} / U_{\text{aus}}$ Multim. [dB]	Oszi. Phase	U_{ein} OurDAQ [V]	U_{aus} OurDAQ [V]	$U_{\text{ein}} / U_{\text{aus}}$ OurDAQ [dB]	OurDAQ Phase
100	0,348	0,346	-0,05	-1,0°	0,3477	0,3463	-0,04	-03,34
550	0,349	0,328	-0,62	-16,5°	0,3459	0,3272	-0,48	-17,75
990	0,348	0,295	-1,44	-31,4°	0,3429	0,2954	-1,30	-29,67
1440	0,346	0,258	-2,54	-41,2°	0,3398	0,2592	-2,35	-39,56
1880	0,344	0,226	-3,69	-48,8°	0,3373	0,2273	-3,43	-47,00
2330	0,342	0,198	-4,74	-51,5°	0,3355	0,1995	-4,52	-52,94
2770	0,341	0,175	-5,77	-58,1°	0,3341	0,1771	-5,52	-57,45
3220	0,340	0,156	-6,78	-63,3°	0,330	0,1581	-6,47	-61,26
3660	0,339	0,141	-7,67	-66,0°	0,3322	0,1429	-7,33	-64,11
4110	0,338	0,128	-8,44	-67,2°	0,3318	0,1297	-8,16	-66,53
4550	0,338	0,117	-9,23	-69,1°	0,3309	0,1190	-8,89	-68,60
5000	0,338	0,108	-9,95	-70,7°	0,3305	0,1094	-9,60	-70,22

Tabelle 7.11: Filterkennlinie - reale Messergebnisse

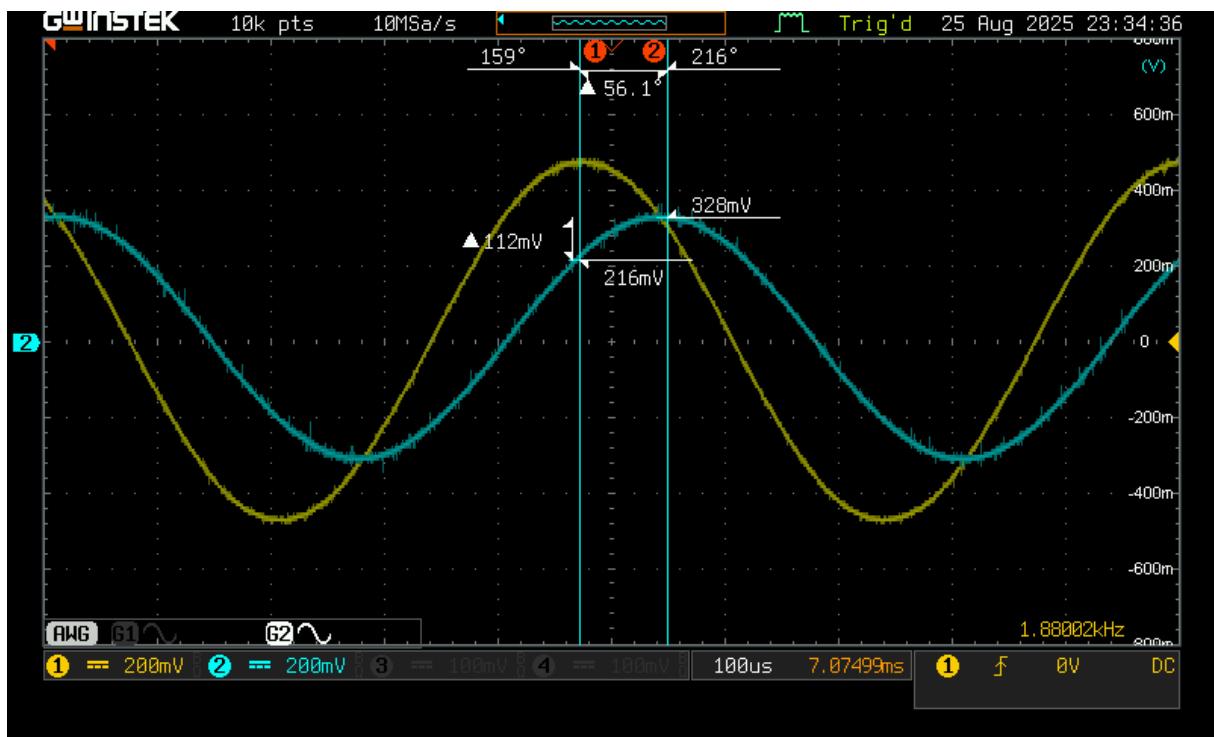


Abbildung 7.22: Messung Filterkennlinie mit Oszilloskop (1880 Hz)

f [Hz]	Verstärkung in dB	Phase
100	-0,017	-3,6°
550	-0,49	-19,1°
990	-1,42	-31,9°
1440	-2,60	-42,1°
1880	-3,79	-49,8°
2330	-4,97	-55,7°
2770	-6,05	-60,1°
3220	-7,07	-63,7°
3660	-7,99	-66,5°
4110	-8,85	-68,8°
4550	-9,63	-70,7
5000	-10,4	-72,3°

Tabelle 7.12: Filterkennlinie - simulierte Messergebnisse

Abbildung 7.23 zeigt den Frequenzgang und Phasenverlauf der Simulation.

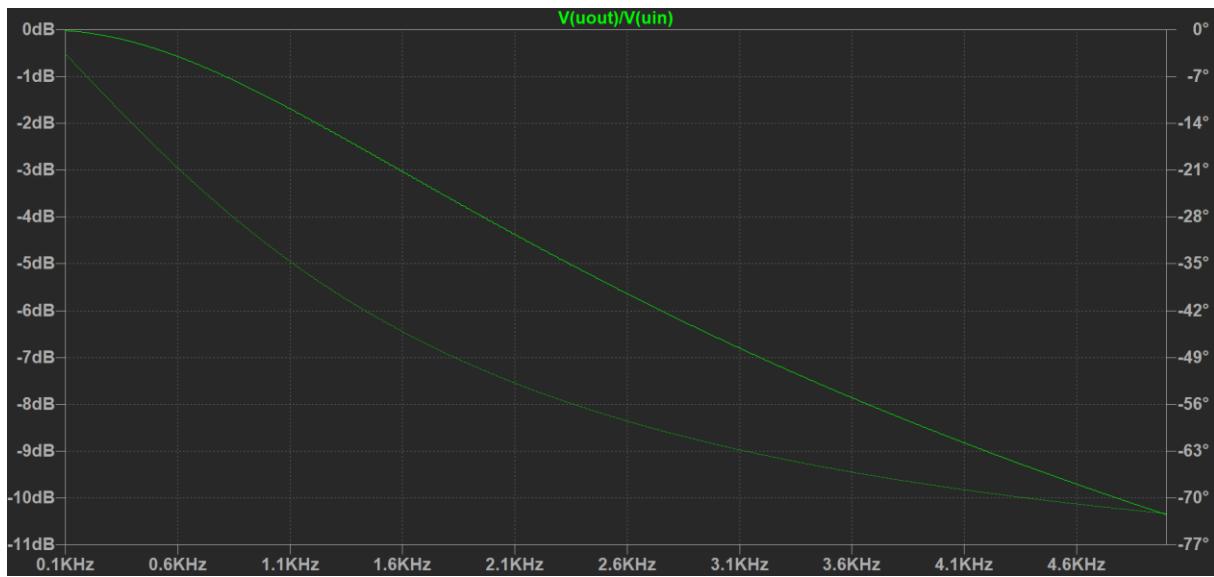


Abbildung 7.23: Messung ideale Filterkennlinie mit LTspice

Die Grenzfrequenz f_g beträgt

$$f_g = \frac{1}{2 * \pi * R * C} \approx 1592 \text{ Hz}$$

Aufgrund der Kondensator-Toleranz $\pm 10\%$ liegt die praktische Grenzfrequenz zwischen $\approx 1447 \text{ Hz}$ ($C = 110 \text{ nF}$) und $\approx 1769 \text{ Hz}$ ($C = 90 \text{ nF}$), was die leichten Verschiebungen gegenüber der Idealsimulation erklärt.

Vergleich Messung vs. Simulation

- Amplitude / Verstärkung:
 - Multimeter vs. LTspice: mittlere Abweichung ca. 0,23 dB
 - OurDAQ vs. LTspice: mittlere Abweichung ca. 0,44 dB
-> Die gemessene Verstärkung stimmt gut mit der Simulation überein
- Phase:
 - Die Phasenlagen stimmen sehr gut mit der LTspice-Simulation überein, es zeigen sich Abweichungen von maximal 2-3°
 - Die Simulation bildet ein ideales Verhalten ab (exakte Bauteilwerte, kein Rauschen, kein Kanalversatz). In der Realität wirken jedoch:
 - Bauteiltoleranzen (insbesondere die verwendeten Kondensatoren mit $\pm 10\%$ Abweichung),
 - geringes Rauschen und numerische Effekte in der Auswertung,
 - sowie der Kanal-zu-Kanal-Delay des MCC 118 ($\Delta t \approx 8 \mu\text{s}$).
 - Durch diese Einflüsse weichen die Messwerte geringfügig von der idealen Simulation ab. Insgesamt bleibt die Restunsicherheit mit etwa 2–3° jedoch klein und für die meisten praktischen Anwendungen vernachlässigbar

In Abbildung 7.24 ist noch das Ergebnis der OurDAQ-Software für die Aufnahme einer Filterkennlinie dargestellt.

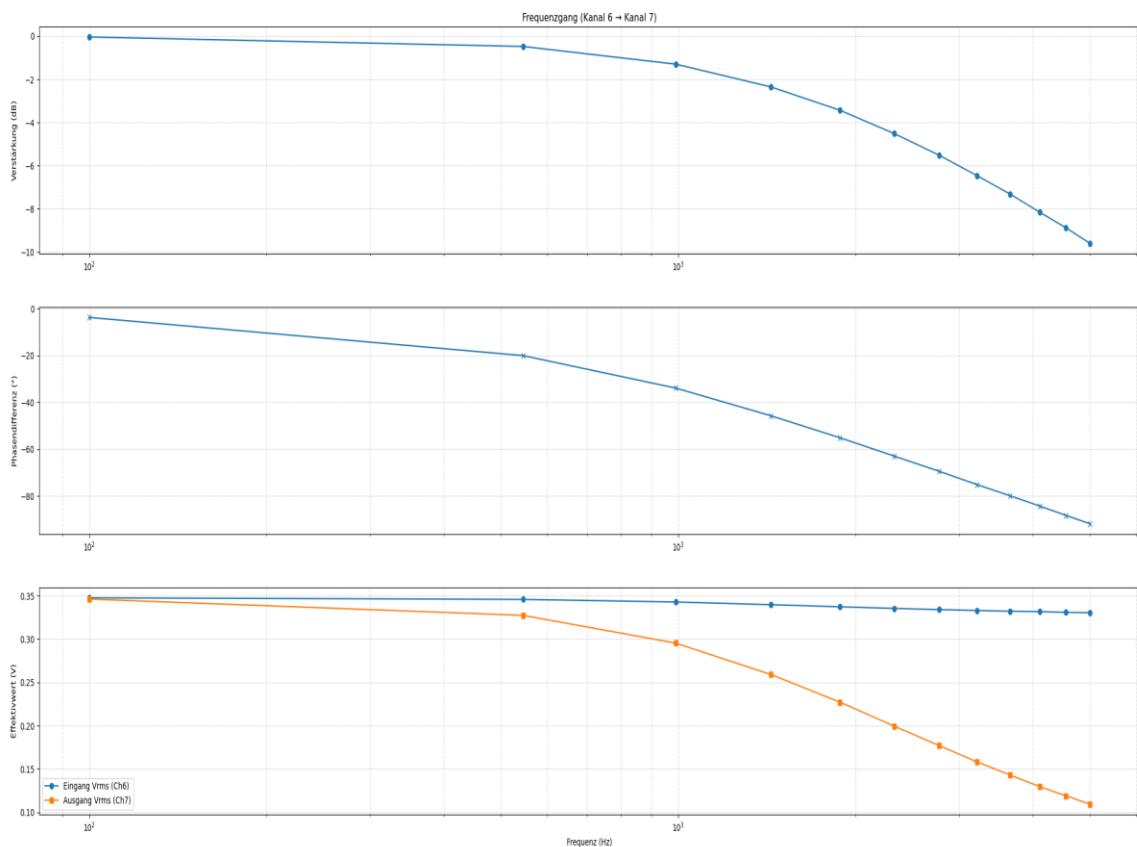


Abbildung 7.24: OurDAQ-SW – aufgenommene Filterkennlinie (12 Punkte)

8 Vergleich OurDAQ vs. NI myDAQ

In Tabelle 8.1 werden die wesentlichen Eigenschaften des entwickelten OurDAQ-Systems mit den Spezifikationen des kommerziellen NI myDAQ gegenübergestellt. Dabei liegt der Fokus auf den wichtigsten Funktionsblöcken wie Spannungs- und Strommessung, Netzteil, Funktionsgenerator und Oszilloskop. [5, 8]

Funktion	OurDAQ	NI myDAQ
Multimeter-DC-Spannung	±10 V Bereich Genauigkeit: ±(0,098 % vom Messwert + 11 mV Offset) ≈ ±20,8 mV bei 10 V	Bereiche: 200 mV, 2 V, 20 V, 60 V Genauigkeit: ±(0,5 % vom Messwert + Offset) z. B. bei 20 V: ±(0,5 % + 20 mV)
Multimeter-AC-Spannung	±10 V Bereich (500 Hz bis 30 kHz) Genauigkeit: ±(max. 3,6 % vom Messwert + max. 11,4 mV)	Bereiche: 200 mV, 2 V, 20 V Genauigkeit: ±(1,4 bis 5,5 % vom Messwert + Offset) z. B. bei 20 V und bis 2 kHz: ±(5,5 % + 50 mV)
Multimeter-DC-Strom	Bereich: ±0,8 A (±1,0 % + 0,5 mA nach Nullabgleich) Bereich: ±10 mA (±0,33 % + 0,07 mA) -> noch nicht getestet	Bereiche: 20 mA, 200 mA, 1 A Genauigkeit: ±(0,5 % vom Messwert + Offset) z. B. 20 mA: ±(0,5 % + 0,03 mA)
Labornetzteil	Bereich ±10 V, max. 1,0 A Abweichung bis ~140 mV (neg. Bereich)	Fest integriert: ±15 V, max. 32 mA nicht regelbar wie OurDAQ

Funktionsgenerator (Analogausgang)	Bereich ± 10 V, konnte zeitlich noch nicht genau spezifiziert werden	± 10 V oder ± 2 V Genauigkeit: $\pm(0,2\% \text{ bis } 0,4\% \text{ vom Wert} + \text{bis } 42,8 \text{ mV})$
Oszilloskop (Analogeingang)	12 Bit, 100 kS/s je Kanal, ± 10 V (Spitze-Spitze) Genauigkeit: $\pm(0,098\% + 11 \text{ mV})$	16 Bit, bis 200 kS/s, ± 10 V Genauigkeit: $\pm(22,8 \text{ mV bei } \pm 10 \text{ V})$
Diodenkennlinie	Spannung 0 bis 10 V, Fehler ab 5 mA $< 6\%$ (Worst Case), bei größeren Strömen ($< 0,7\%$)	Diodentest integriert (2 V Range), Genauigkeit vergleichbar zu Spannungsmessung (0,5 % + Offset)
Zusätzliche Funktionen	Messung Filterkennlinie, Dioden, Python/Jupyter Integration, GPIOs Raspberry Pi	Widerstände bis $20 \text{ M}\Omega$, Audioeingang

Tabelle 8.1: Vergleich OurDAQ vs. NI myDAQ

Fazit

- Genauigkeit:
 - Bei DC-Spannungen ist OurDAQ präziser als NI myDAQ ($\approx 0,1\%$ vs. $0,5\%$)
 - Bei AC-Spannungen (500 Hz – 30 kHz) erreicht OurDAQ mit max. $\pm 3,6\% + 11 \text{ mV}$ ebenfalls sehr gute Werte und liegt auf vergleichbarem Niveau wie das myDAQ ($\pm 5,5\% + \text{Offset bis } 2 \text{ kHz}$)
- Funktionalität:
 - OurDAQ bietet ein regelbares Netzteil (± 10 V) bis 1 A, was beim myDAQ nicht möglich ist (nur ± 15 V / 32 mA fix)

- Das Oszilloskop des OurDAQ ist ausreichend, während das NI myDAQ durch höhere Auflösung (16 Bit, doppelte Abtastrate) punktet
- Zusatzfunktionen sind unterschiedlich: OurDAQ hebt sich durch Jupyter/Python-Integration und Messungen an Dioden und Filtern ab, während NI myDAQ eine Audio-Schnittstelle und Widerstandsmessung bietet
- Gesamtbewertung:
 - OurDAQ zeigt sich besonders stark bei Präzision der DC-Messungen und durch das regelbare Netzteil
 - NI myDAQ hat Vorteile bei höherer Auflösung und Samplingrate im Oszilloskop sowie integrierter Widerstandsmessung
 - Für Lehrexperimente und praxisnahe Laborszenarien kann OurDAQ durch die Kombination von Präzision, Netzteil, Funktionsgenerator und Softwareintegration einen sehr flexiblen Ansatz bieten

9 Design - Leiterplatte

Aus dem vollständigen Schaltplan des OurDAQ wurde anschließend das Layout der Leiterplatte in KiCad erstellt. Die Platine wurde als 4-Lagen-PCB ausgeführt, wobei die Lagen wie folgt aufgeteilt wurden:

- Signalebene (Top-Layer, rot) für Mess- und Steuerleitungen
- Versorgungsspannungsebene (Power-Plane, orange)
- GND-Ebene (Ground-Plane, grün) für stabile Massebezüge
- Signalebene (Bottom-Layer, blau) für weitere Mess- und Steuerleitungen

Die dedizierte Masseebene sorgt für niedrige Impedanzpfade und reduziert elektronische Störfelder. Die Abbildungen 9.1 bis 9.4 zeigen den Aufbau der einzelnen Lagen. [38]

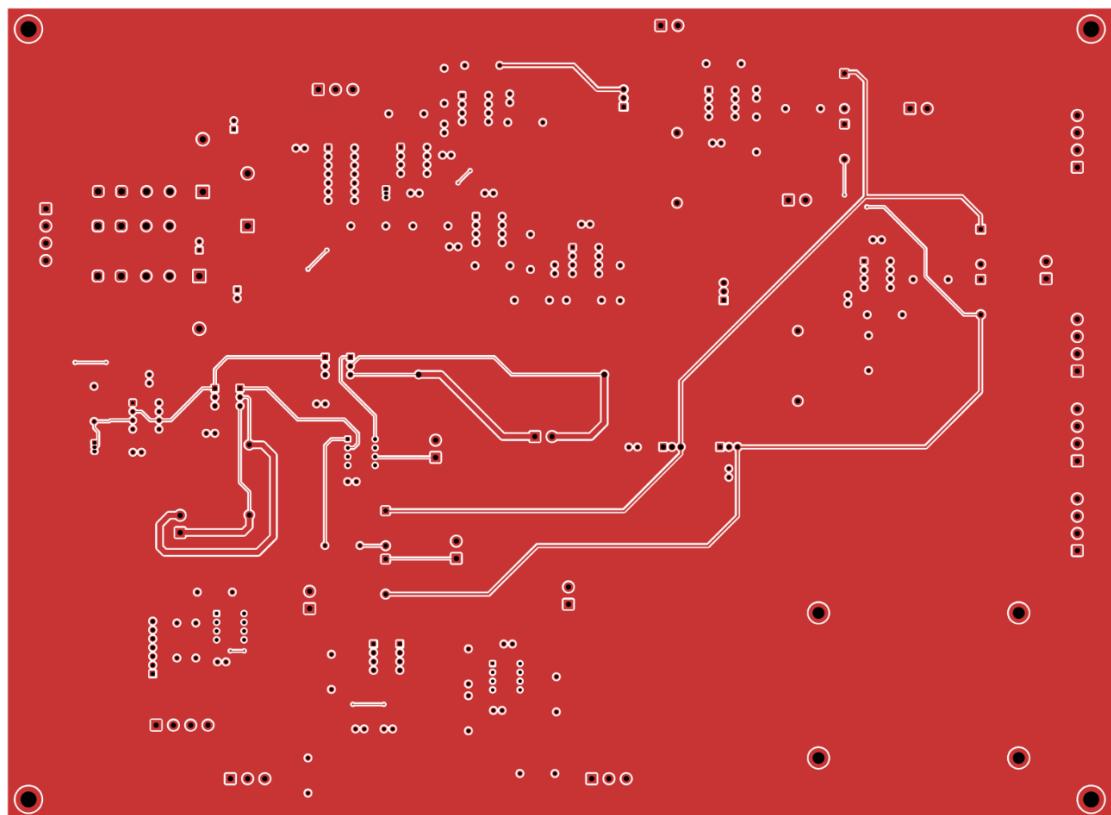


Abbildung 9.1: Leiterplatte - Top-Layer

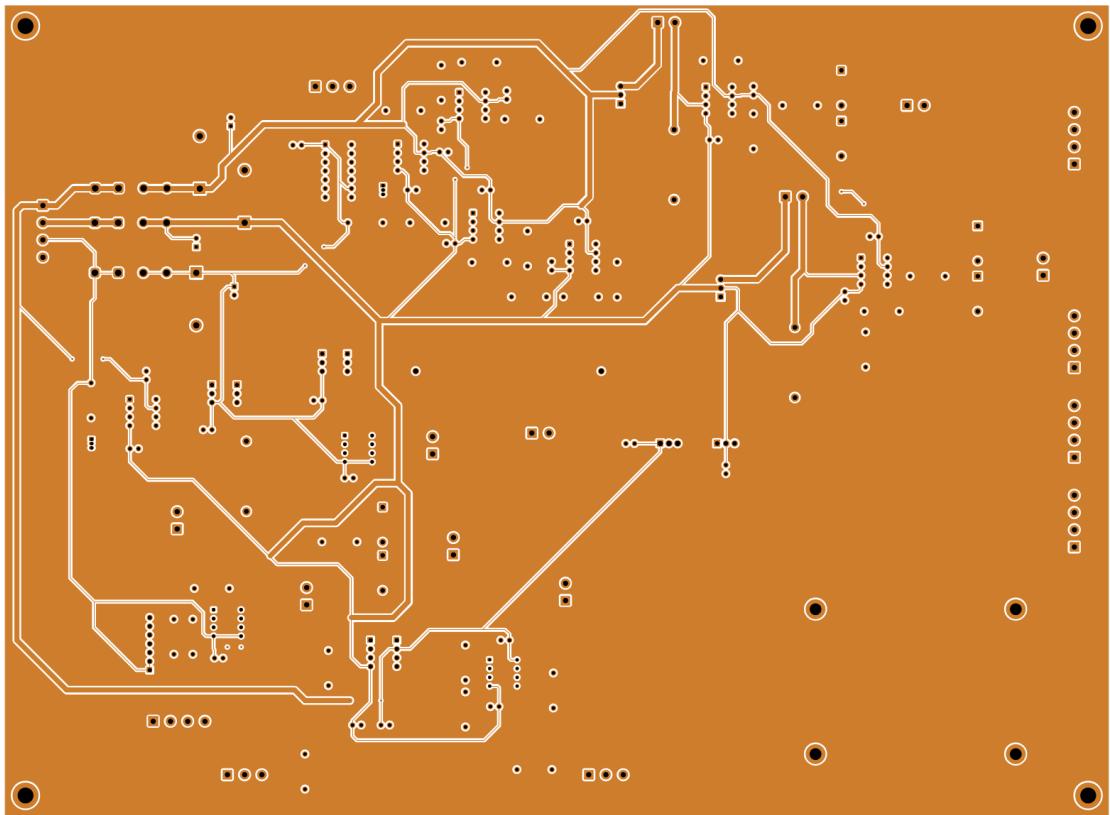


Abbildung 9.2: Leiterplatte - Power-Plane

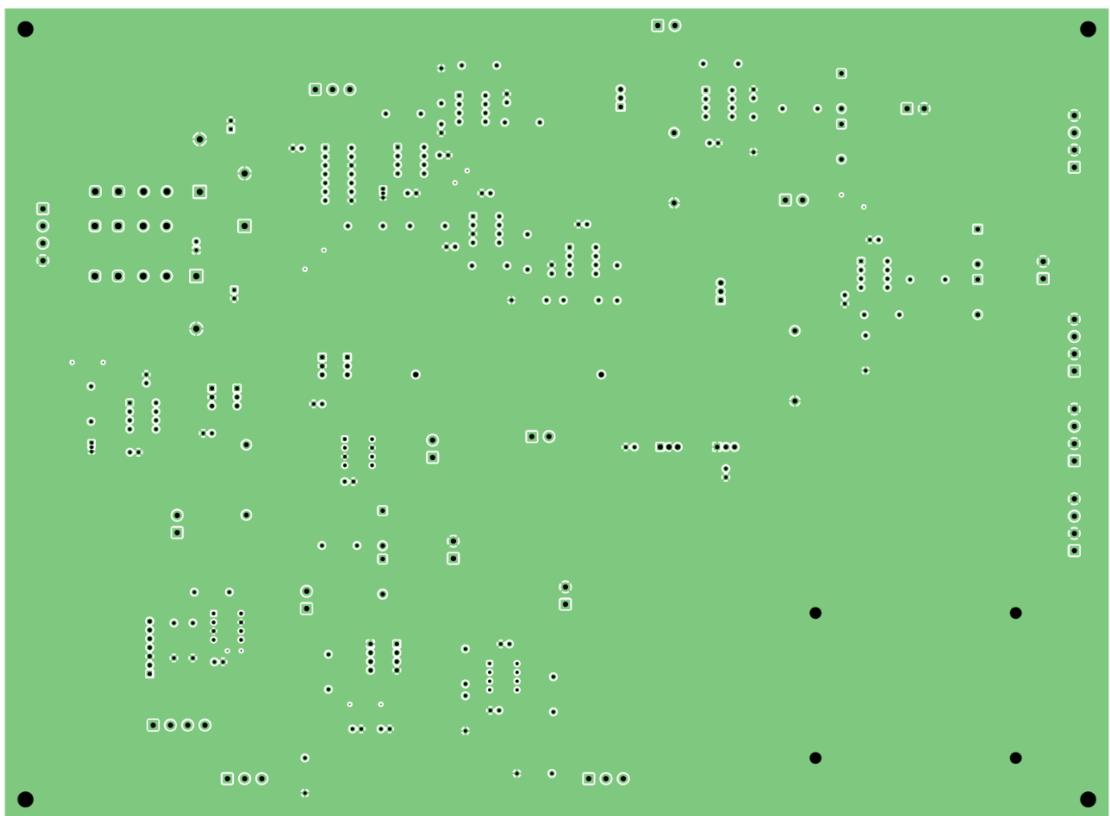


Abbildung 9.3: Leiterplatte - Ground-Plane

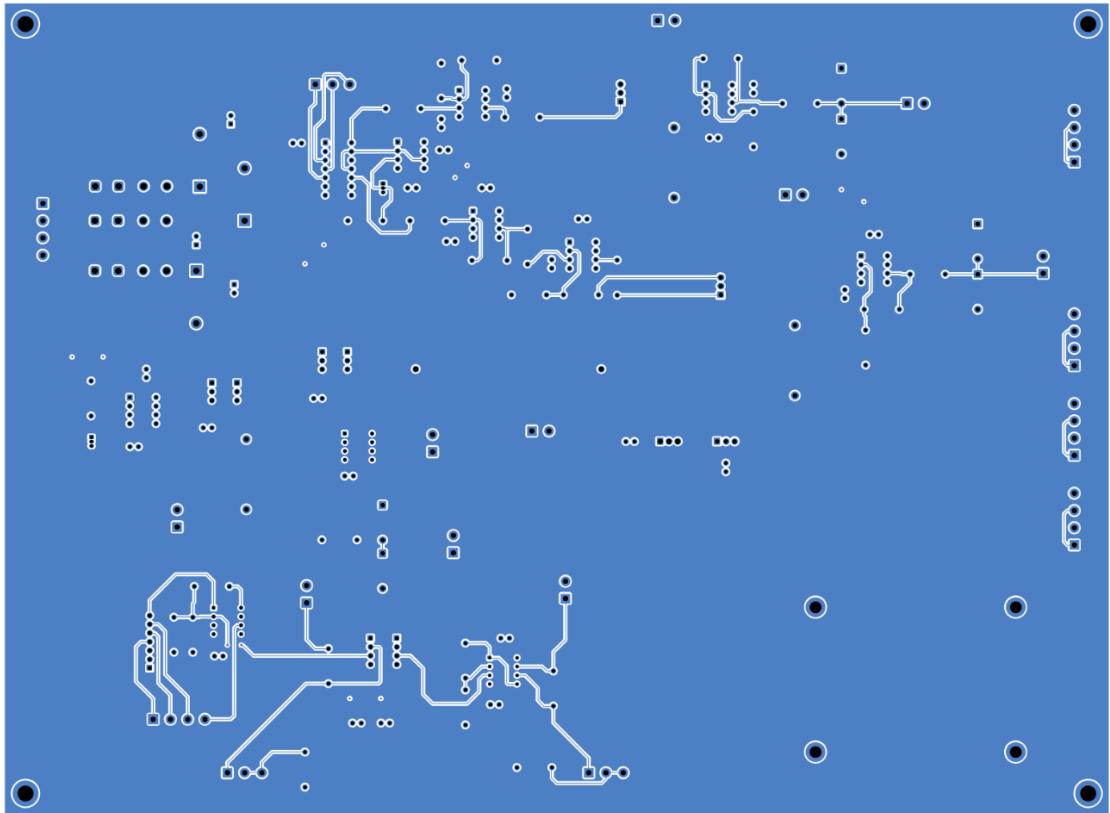


Abbildung 9.4: Leiterplatte - Bottom-Layer

Alle Bauteile wurden hier thematisch gruppiert und strategisch auf der Platinne platziert, um kurze Signalwege, geringe Störempfindlichkeit und einfache Fertigung zu gewährleisten. Die Verbindungsleitungen wurden anschließend auf den einzelnen Lagen gesetzt, wobei Designregeln wie die Vermeidung von schrägen Leitungsführungen direkt aus den Bauteilen heraus beachtet wurden.

Die Versorgungsleitungen, die für höhere Ströme ausgelegt sind, wurden mit einer Breite von 1,5 mm dimensioniert, um Spannungsverluste und Erwärmung zu minimieren.

Alle Anschlüsse, sowohl interne Verbindungen als auch für externe Schnittstellen, wurden in das Layout integriert und für die Montage mit Schraubklemmen vorgesehen. Dies erleichtert den Aufbau, die Handhabung während der Messungen sowie den Anschluss externer Geräte. Dies ist in den 3D-Modellen der Leiterplatte (vgl. Abbildung 9.5 bis 9.7) ersichtlich.

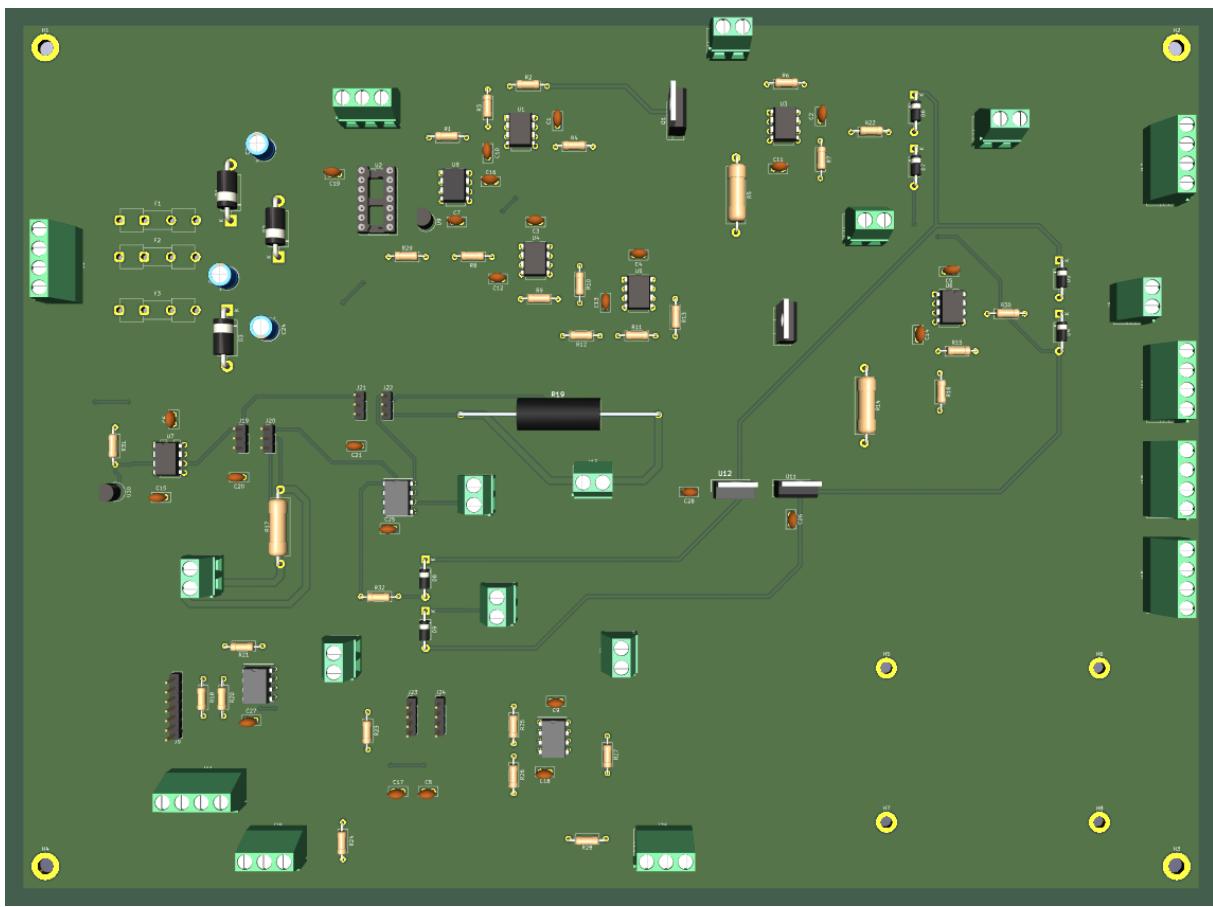


Abbildung 9.5: Leiterplatte - 3D-Ansicht oben

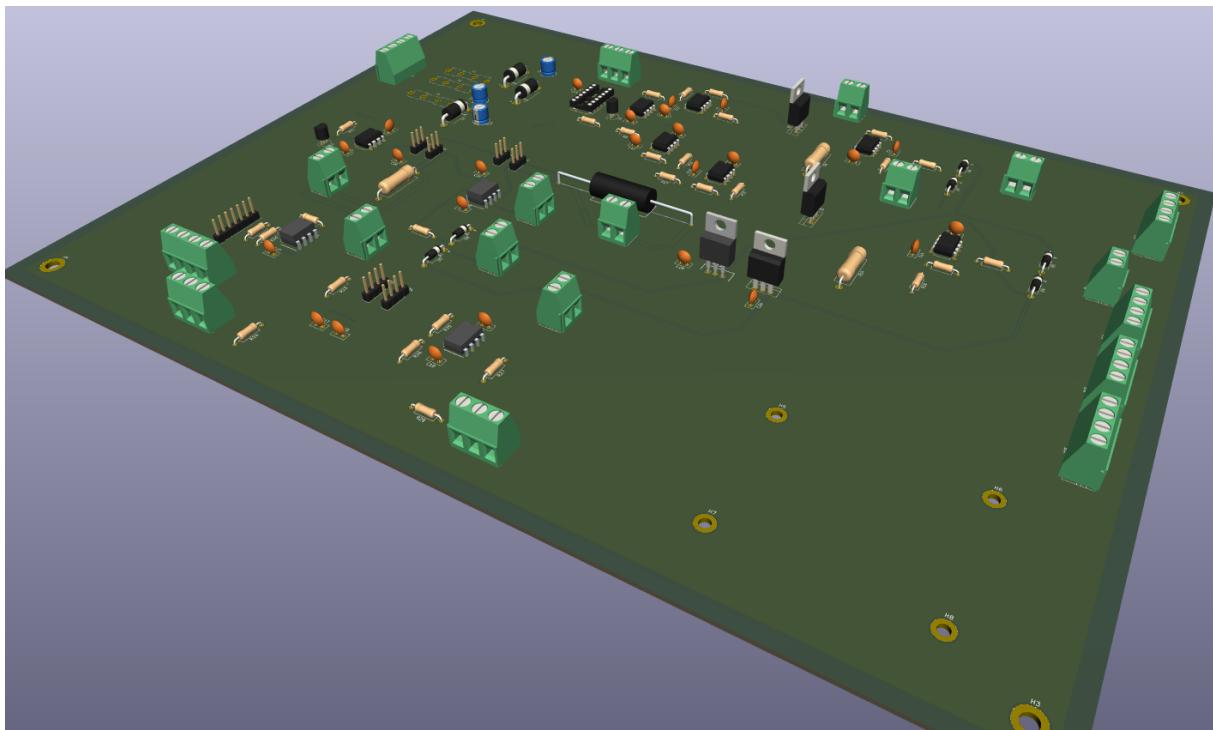


Abbildung 9.6: Leiterplatte - 3D-Ansicht seitlich

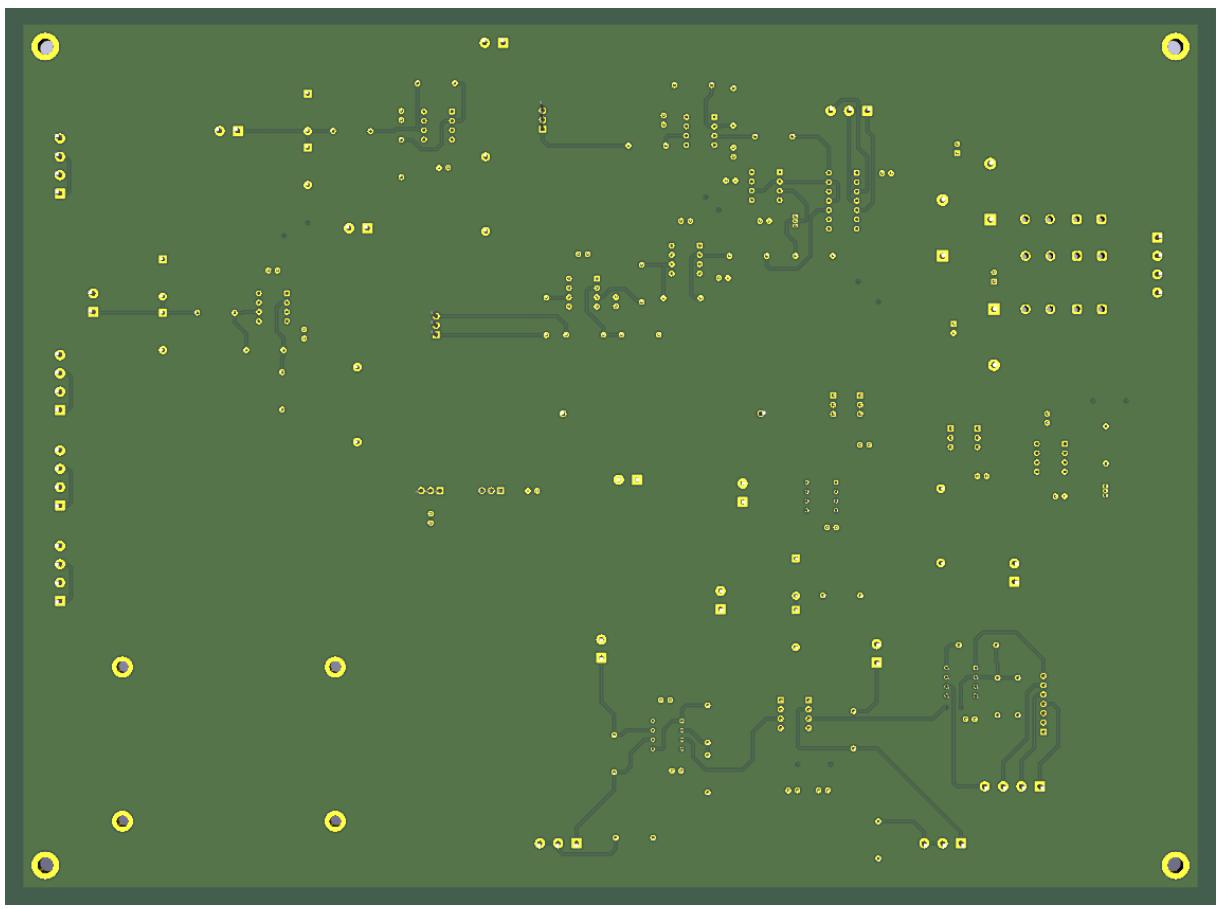


Abbildung 9.7: Leiterplatte - 3D-Ansicht unten

10 Konstruktion von Gehäuse und Komponenten

Im Rahmen des Projektes ist vorgesehen, dass für das Messdatenerfassungssystem ein Gehäuse zu konstruieren ist. Auf Grund der Tatsache, dass das Projekt vorzeitig eingefroren worden ist, ist eine Realisierung des Gehäuses auch nicht mehr notwendig. Mit der Hoffnung, dass das Projekt zukünftig weiterverfolgt wird, wurde aus Seiten des Projekt-Teams sich darauf geeinigt, dass trotzdem ein Konzept vorgestellt werden soll, dass einem nachfolgenden Team noch genug Spielraum zur kreativen Vielfalt ermöglicht.

Die Konstruktion hat sich gegen eine Konstruktion wie es in der Mechanik vorgesehen ist entschieden, da die Leiterplatte, welche die Module Funktionsgenerator und Netzteil vereint, viel zu groß ist und in seinen Dimensionen deutlich reduziert werden kann. Aus diesem Grund steht eher das Konzept im Vordergrund, welches wie gesagt viel Spielraum zur Anpassung liefert.

10.1 CAD-Modellierung

Die CAD-Modellierung erfolgt über Inventor Fusion 360, dessen Lizenz für Studierende kostenlos über die Technische Hochschule Nürnberg erwerbbar ist. Dieses CAD-Tool bietet sowohl die Möglichkeit mechanische Bauteile bzw. Baugruppen als auch Leiterplatten zu entwickeln. Mit einem breiten Spektrum an unterschiedlichen Bibliotheken, sind Norm-Teile sowie elektronische Bauelemente leichter zugänglich als durch das Importieren von .STEP-Dateien wie in anderen CAD-Tools.

10.2 Gehäuse Konzept

Das Konzept des Gehäuses ist so ausgelegt, dass sich dieses sowohl an der Leiterplatte orientiert, welches das Netzteil und den Funktionsgenerator als Modul vereint als auch an der Benutzerfreundlichkeit sowie der Transparenz. Weiterhin wurde darauf abgezielt, dass das eigentliche Gehäuse ohne die Plexiglasabdeckungen zu 100% durch einen 3D-Drucker hergestellt werden kann. In diesem Maße, wie in diesem Konzept vorgefunden, können die Gehäuse Komponenten auf bekannten 3D-Druckern wie Bambu-LAB nicht gedruckt werden, da die Druckfläche kleiner als die Hauptmaße des Gehäuses ist. Eine Reduzierung der großen Leiterplatine ist von daher wichtig!



Abbildung 10.1: Design Concept "Das Messdatenerfassungssystem - OurDAQ"

Die Bedienung des Messdatenerfassungssystems erfolgt nur auf der Vorder- und Oberseite. Auf der Vorderseite (vgl. Abbildung 10.2: Analoge und Digitale In- und Outputs vom Raspberry Pi 5 und Digilent MCC 118 DAQ HAT) hat man die Möglichkeit alle Ein- und Ausgänge vom Raspberry Pi 5 und Digilent MCC 118 DAQ HAT zu benutzen. Außerdem sind zwei Potentiometer integriert, um die potentiometrischen Widerstände des Funktionsgenerators zu bedienen, um eine feine Justierung zu erstellen (vgl. Abbildung 4.23: Anpassung des Spannungsbereiches mittels AD8479 Differenzverstärker und der Multiplexer ADG419 (Blockschaltbild 1 – 3 des Funktionsgenerator-Moduls)).

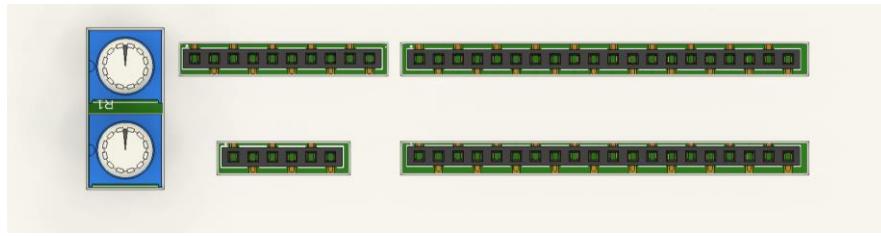


Abbildung 10.2: Analoge und Digitale In- und Outputs vom Raspberry Pi 5 und Digilent MCC 118 DAQ HAT

Dafür wurde überschlägig eine weitere Leiterplatte – kurz: „Header-Platine“ – entwickelt (vgl.

Abbildung 10.3: Leiterplatine: "Header-Platine"), die entweder bei konventionellen Herstellern zur Fertigung gegeben werden kann oder in Kooperation mit Prof. Dr. Reichenberger selbst entwickelt werden kann. Eine weitere Option wäre auch die Nutzung von Lochrasterplatinen. Mit Jump Wire Kabeln kann man GPIO-Anschlüsse mit dieser Leiterplatine verbinden, sodass auch vom Gehäuse aus das Raspberry Pi benutzt werden kann.

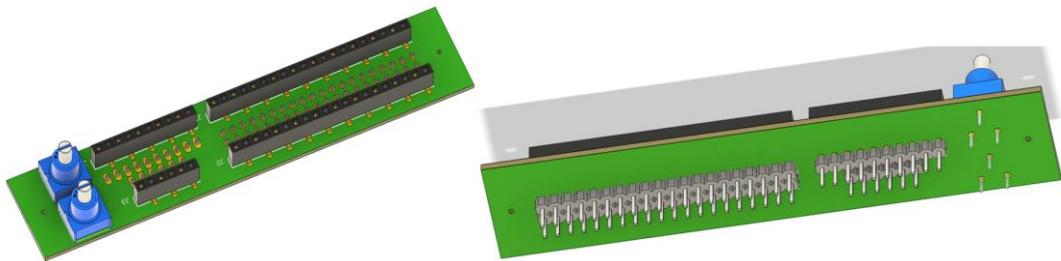


Abbildung 10.3: Leiterplatine: "Header-Platine"

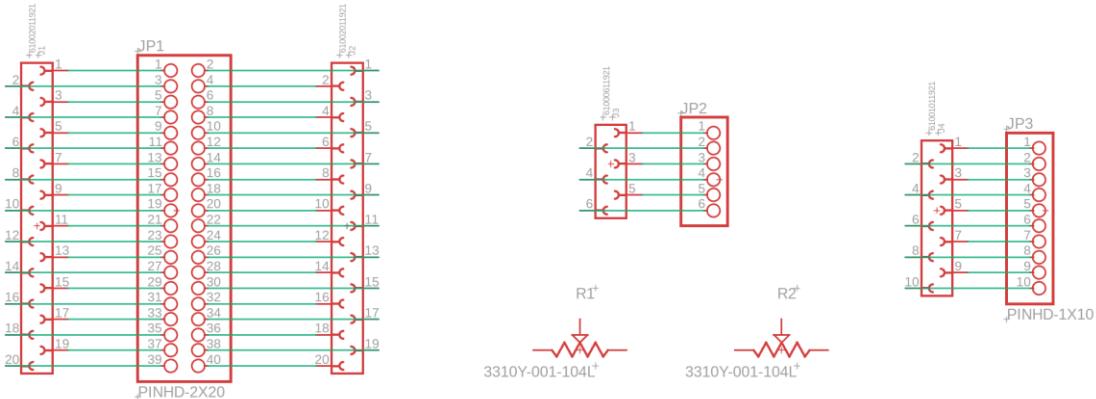


Abbildung 10.4: Schaltplan der Header-Platine

Die Anordnung der Peripherie Anschlüsse auf der Oberseite des Gehäuses (vgl. Abbildung 10.5: Oberseite des Gehäuses mit den Anschlässen) ergeben sich durch die große Leiterplatine (vgl. **Fehler! Verweisquelle konnte nicht gefunden werden.**) und sind auf dem Gehäusedeckel m

it einem angenehmen Abstand zueinander geordnet platziert, sodass mit Leichtigkeit Messungen vorgenommen werden können, ohne an der Orientierung des Messdatenerfassungssystems gebunden zu sein. Die Montage der Anschlüsse an den Seitenwänden entspricht dabei keinem ergonomischen Design, auch wenn alle Seiten des OurDAQs zugänglich wären. Lediglich die GPIO-Anschlüsse wurden abweichend von der vorgesehenen Platzierung auf der Vorderseite platziert. Diese Position bringt jedoch keine Nachteile mit sich (vgl. Abbildung 10.2: Analoge und Digitale In- und Outputs vom Raspberry Pi 5 und Digilent MCC 118 DAQ HAT).

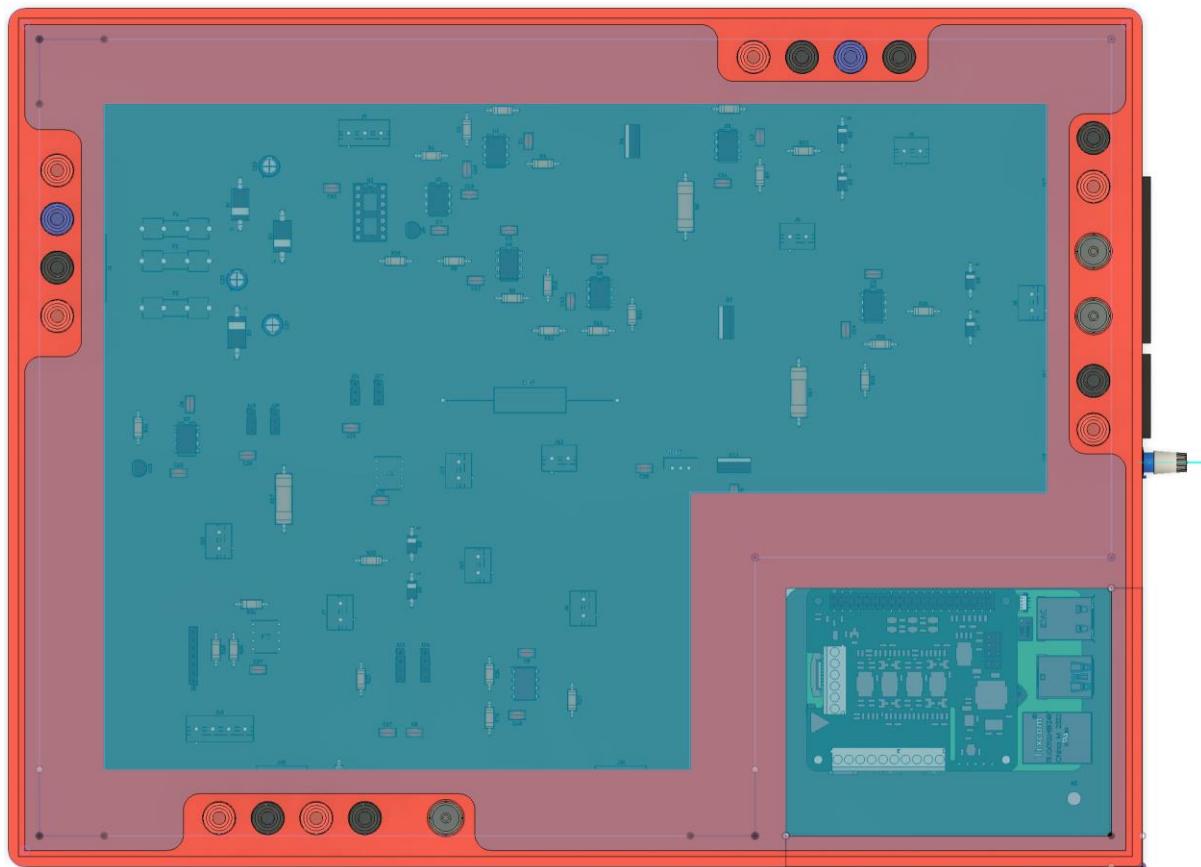


Abbildung 10.5: Oberseite des Gehäuses mit den Anschlüssen

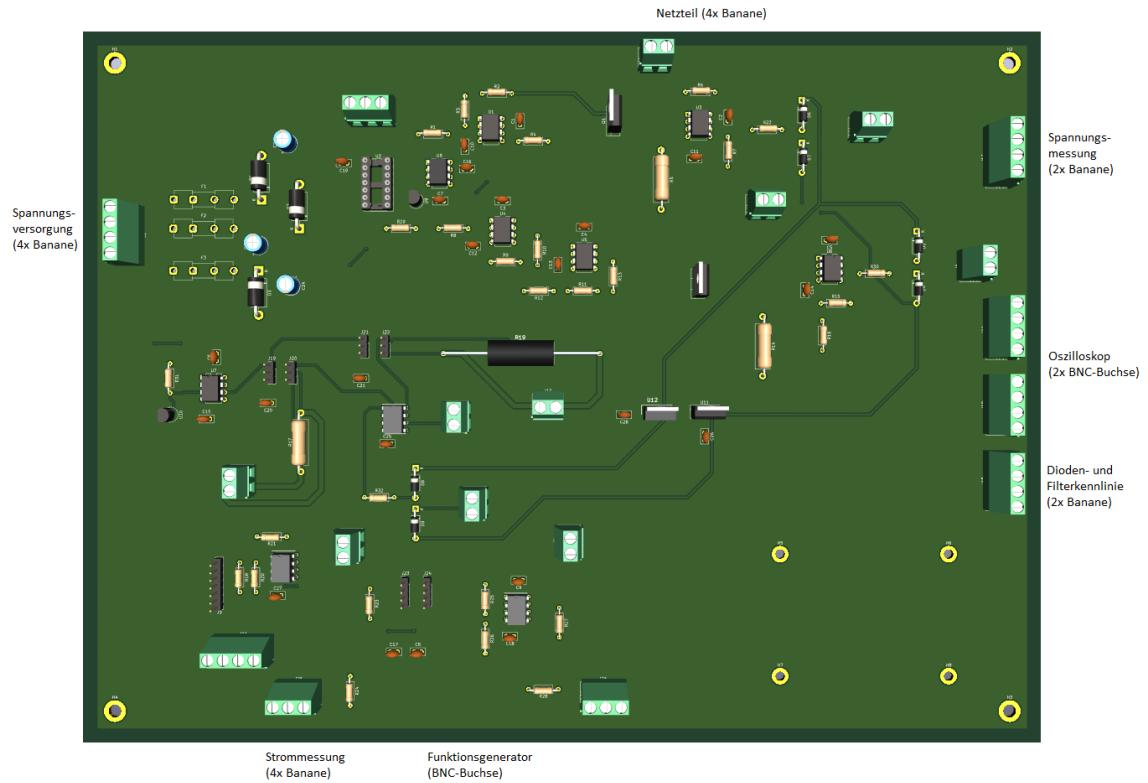


Abbildung 10.6: Anordnung der Peripherie Anschlüsse an der großen Leiterplatine

Um an Peripherie des Raspberry Pi 5 zu gelangen, wurde am Gehäuse weiterhin eine freie Stelle gelassen, sodass Tastatur, Maus, HDMI-Kabel etc. weiterhin zugänglich sind (vgl. Abbildung 10.7: Freie Stelle des Gehäuses für die Peripherie des Raspberry Pi 5). Man kann zwar für Maus und Tastatur auf Bluetooth USB-Receiver zurückgreifen, jedoch gibt es diese Alternativen nicht für den USB-C und Micro-HDMI Anschluss, sodass ein fester Platz für das OurDAQ weiterhin empfehlenswert ist.

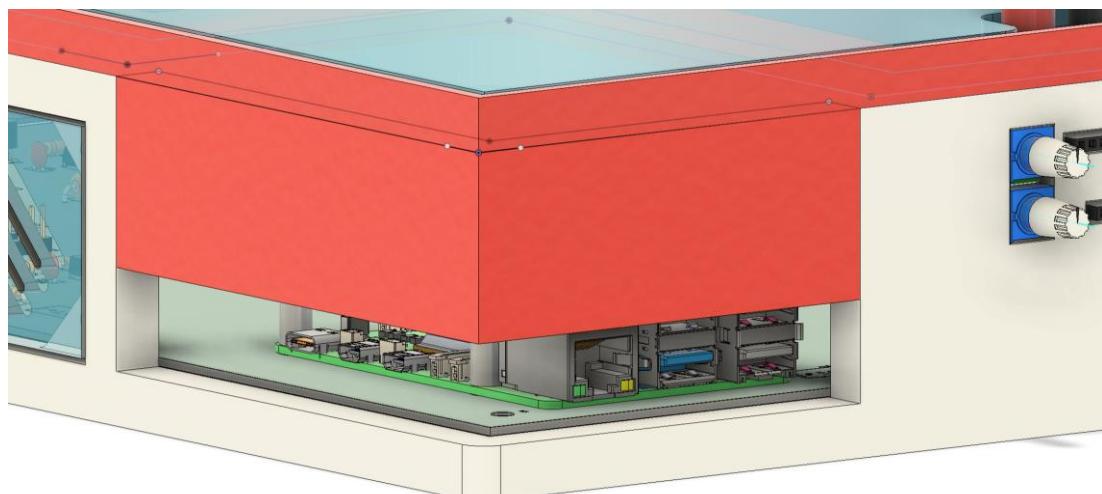


Abbildung 10.7: Freie Stelle des Gehäuses für die Peripherie des Raspberry Pi 5

10.3 Einzelteil- und Montagebeschreibung

10.3.1 Grundplatine

Die Basis des Gehäuses bildet die Grundplatte (vgl. Abbildung 10.8: Grundplatte des Gehäuses), welche 3D gedruckt schon mehr als 50% des Gehäuses ausmacht. Die Grundplatte enthält dabei alle Seitenwände, die entweder offengelassen werden können oder durch abdeckende Ebenen oder Plexiglas abgedeckt werden können. Auf Grund der Skelettbauweise des Gehäuses wurden an den Ecken der tragenden Balken verstärkende Fasen Elemente implementiert, um der Kerbwirkung entgegenzuwirken, da auf der Gehäusedecke durch das An- und Abstecken von Kabel Zug- und Druckkräfte entstehen.

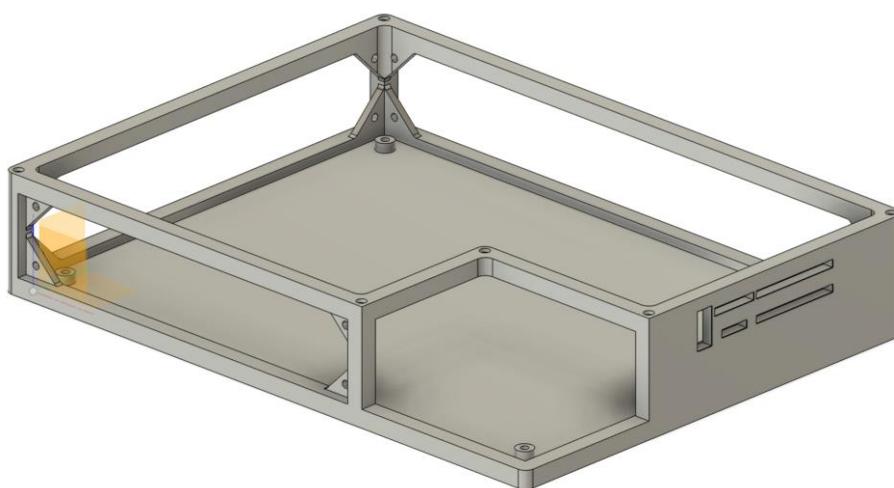


Abbildung 10.8: Grundplatte des Gehäuses

Auf der Rückseite von der Vorderseite der Grundplatine ist eine Aussparung vorgesehen, um die Header-Platine zu montieren. An den oberen Ecken wiederum sind weitere Aussparungen, um mit einem Kleber 5x2 mm kleine Scheibenmagnete zu kleben, die deckungsgleich mit den Scheibenmagneten des Gehäusedeckels liegen sollen (vgl. Abbildung 10.13: Montage des Gehäuse Deckels an der Grundplatte). Weiterhin sind am Grund der Grundplatine jeweils 4 Beine zur Montage der Leiterplatte mit jeweiligen Schrauben (vgl. Abbildung 10.9: Graphische Montage der Grundplatte).

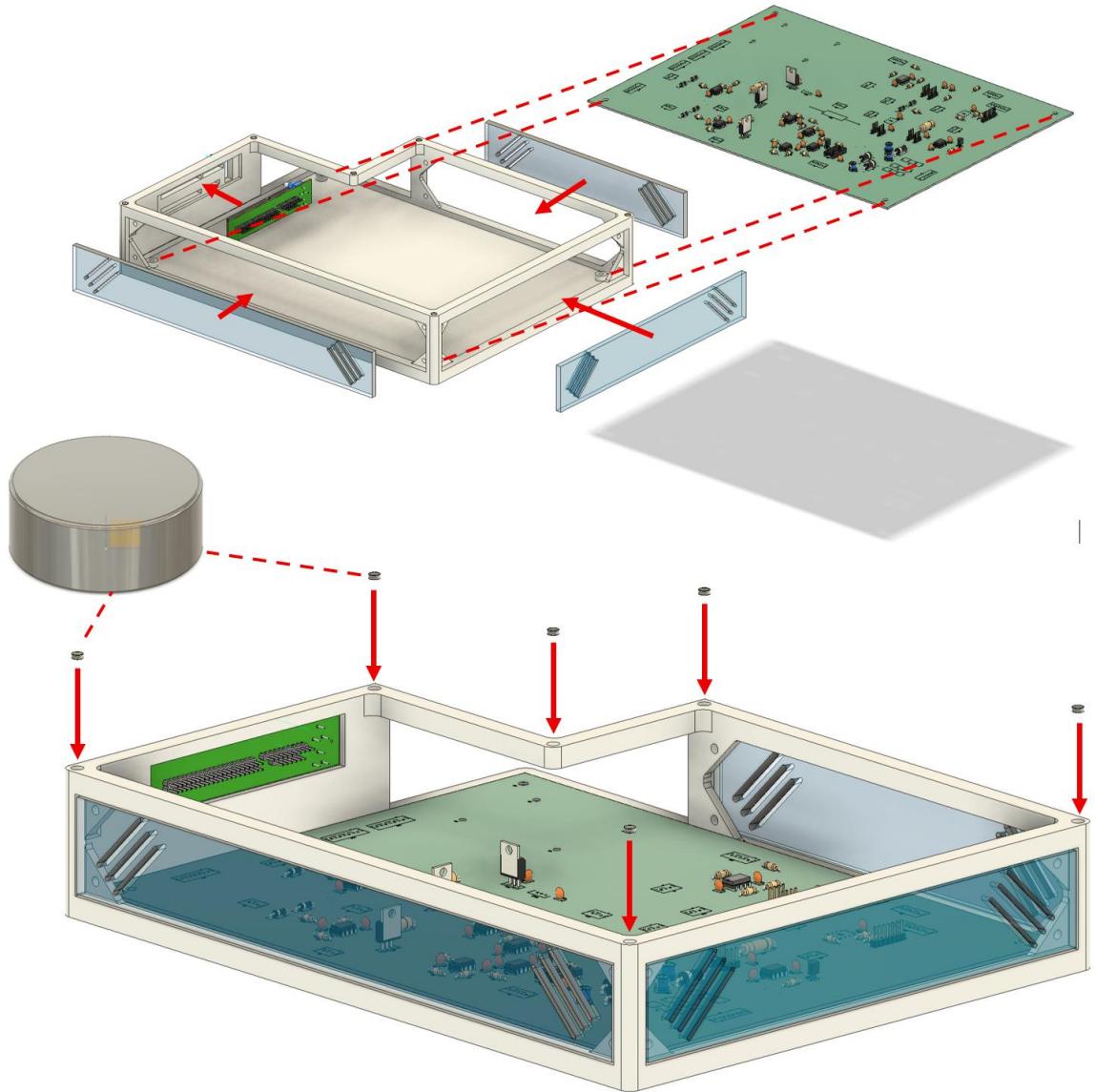


Abbildung 10.9: Graphische Montage der Grundplatte

Nachdem die große Leiterplatine sowie die Header Platine an der Grundplatine montiert sind, kann auch der Raspberry Pi 5 zusammen mit dem Digilent MCC 118 HAT an der großen Leiterplatine montiert werden, womit die Montage an der Grundplatine als beendet gilt (vgl. Abbildung 10.10: Montage des Raspberry Pi 5 zusammen mit dem Digilent MCC 118 DAQ HAT).

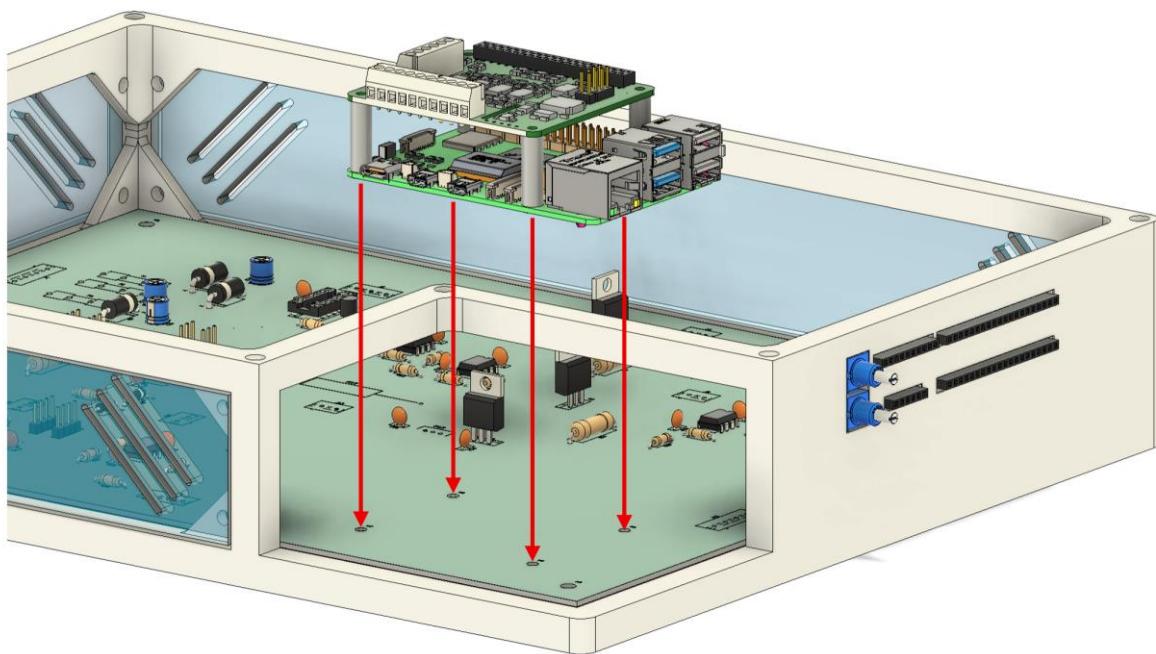


Abbildung 10.10: Montage des Raspberry Pi 5 zusammen mit dem Digilent MCC 118 DAQ HAT

10.3.2 Gehäuse Deckel

Auch der Gehäuse Deckel ist so konzipiert, dass dieser per 3D-Druck hergestellt werden soll. Der Gehäuse Deckel selbst soll magnetisch mit Scheibenmagneten an den von der Grundplatte vorhandenen Magneten leicht platziert werden und auf Wunsch wieder leicht entfernt werden können (vgl. Abbildung 10.13: Montage des Gehäuse Deckels an der Grundplatte). Auf dem Deckel sind 16 Bananenbuchsen und 3 BNC-Buchsen vorgesehen. Auch hier soll die Abdeckung durch das Auflegen eines Plexiglases erfolgen, sodass das Innenleben des OurDAQs gesehen werden kann (vgl. Abbildung 10.11: Graphische Montage des Gehäuse Deckels (1)).

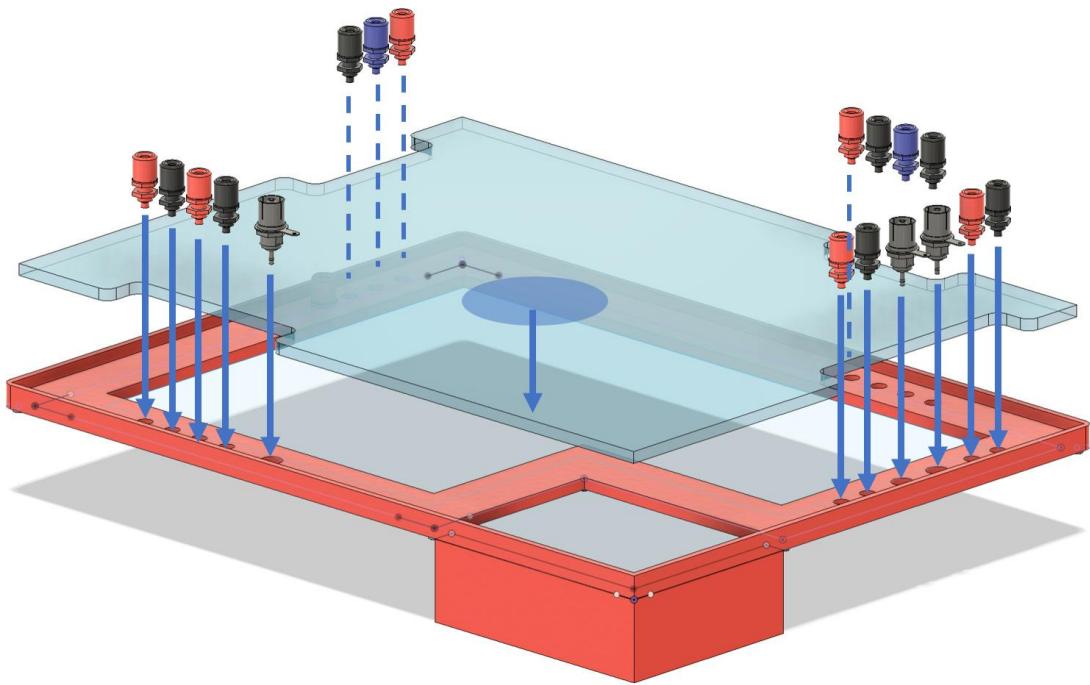


Abbildung 10.11: Graphische Montage des Gehäuse Deckels (1)

An den Ecken der Rückseite wird wie auch schon auf der Oberseite der Grundplatte wieder 5x2 mm Scheibenmagnete mit entgegengesetzter Polung aufgeklebt, sodass das Gehäuse auf der Grundplatte sitzt (vgl. Abbildung 10.12: Graphische Montage des Gehäuse Deckels (2)).

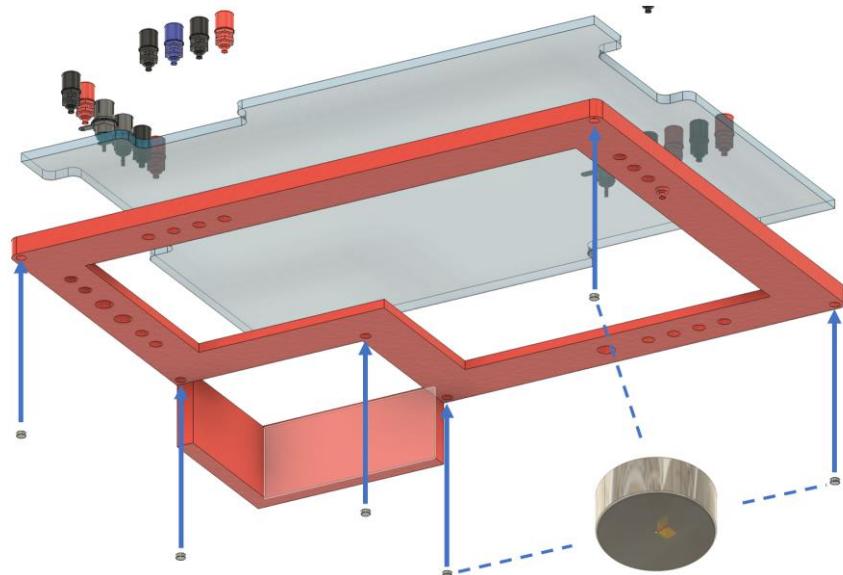


Abbildung 10.12: Graphische Montage des Gehäuse Deckels (2)

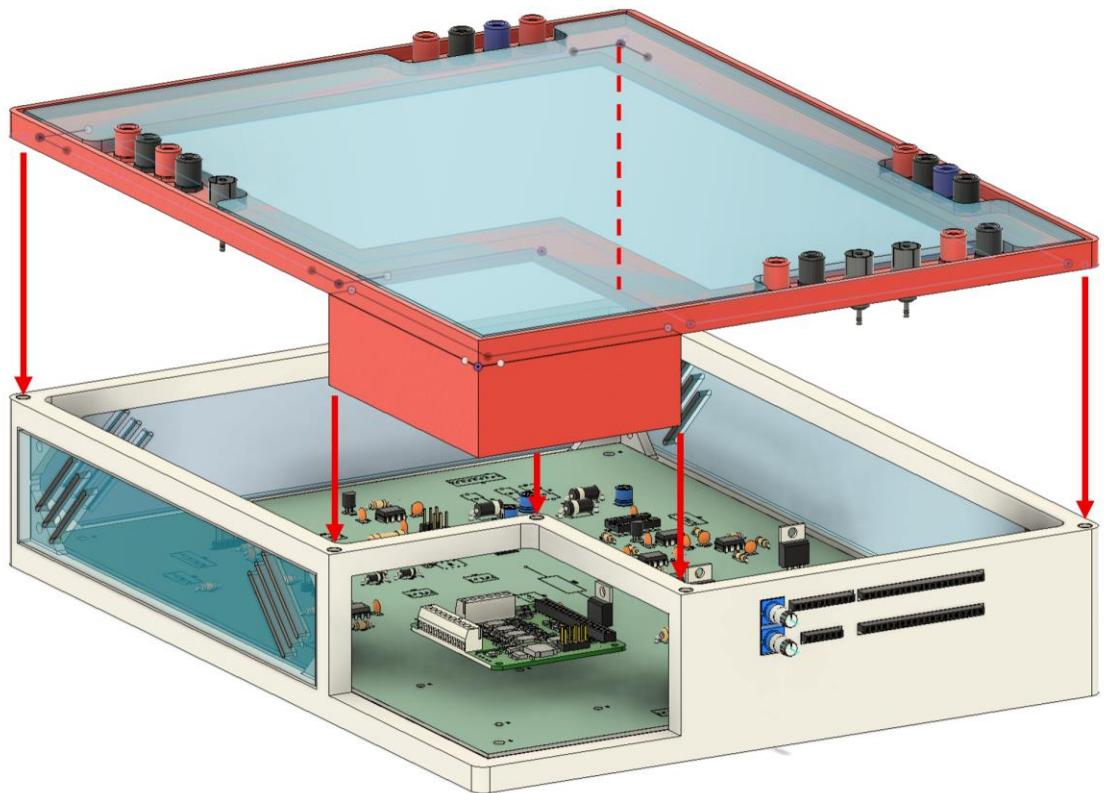


Abbildung 10.13: Montage des Gehäuse Deckels an der Grundplatte

11 Projektstand und Ausblick

Im Rahmen dieser Projektarbeit wurde ein modulares Datenerfassungssystem auf Basis des Raspberry Pi entwickelt. Es integriert wesentliche Laborfunktionen wie die eines Digitalmultimeters (DMM), eines Oszilloskops, eines Netzteils sowie eines Funktionsgenerators in einer webbasierten Benutzeroberfläche. Mit den entwickelten Schaltungen, einer gefertigten Leiterplatte und einem passenden Gehäuse liegt bereits ein funktionsfähiger Prototyp vor, der in der nächsten Projektphase vollständig bestückt, in Betrieb genommen und umfassend getestet werden kann.

Softwarestand

- Oszilloskop (**Oszilloskop_web.py**): Wellenformen können auf mehreren Kanälen zuverlässig dargestellt werden. Eine Optimierung der Samplingrate ist noch erforderlich.
- Netzteil (**Netzteil_plus_web.py / Netzteil_minus_web.py**): Die DAC-Ausgänge für positive und negative Spannungen funktionieren zuverlässig. Für eine höhere Benutzerfreundlichkeit sollten beide Module in einer gemeinsamen Oberfläche zusammengeführt werden.
- Funktionsgenerator (**Funktionsgenerator_web.py**): Wellenform und Frequenz des AD9833 lassen sich korrekt einstellen. Beim Start über das Dashboard (**Dashboard_web.py**) tritt jedoch ein Initialisierungsfehler auf, sodass das Modul aktuell nur separat nutzbar ist. Eine Fehleranalyse ist notwendig. Eine sinnvolle Erweiterung wäre eine Sweep-Funktion, um Frequenzen automatisch zu durchlaufen.
- Digitalmultimeter (**DMM_web.py**): Gleichspannungen werden korrekt gemessen. Die Strommessung konnte aufgrund einer beschädigten Schaltung noch nicht getestet werden. Für die AC-Messung existiert ein Grundgerüst (**DMM_test.py**), das weiterentwickelt werden muss.
- Kennlinienmessung (**Diodenkennlinie.ipynb / Filterkennlinie.ipynb**): Die Jupyter Notebooks erzeugen derzeit Fehler. In den zugehörigen Python-Skripten (**Diodenkennlinie_test.py, Filterkennlinie_test.py**) funktioniert die Logik bereits korrekt. Die Integration in die Notebook-Umgebung ist eine vordringliche Aufgabe für die nächste Version.

Verbesserungen Hardware

- Netzteil: Einführung einer kontinuierlichen Spannungsrückkopplung per ADC, wodurch eine präzisere Regelung möglich wird und eine separate Kalibrierung entfallen könnte
- Strommessung: Schaltung vorhanden, jedoch noch nicht praktisch getestet, Validierung der Genauigkeit steht aus

- Oszilloskop: Für zuverlässige AC-Messungen bei hohen Frequenzen sind höhere Abtastraten erforderlich. Dies kann durch zusätzliche MCC118-Module oder leistungsfähigere ADCs realisiert werden
- Kennlinienmessung: Die funktionierenden Python-Skripte müssen vollständig in die Jupyter-Umgebung überführt werden

Mit den genannten Maßnahmen kann das OurDAQ-System nicht nur funktional erweitert, sondern auch in Bezug auf Genauigkeit, Stabilität und Benutzerfreundlichkeit optimiert werden. Durch die Veröffentlichung als Open-Source-Projekt auf GitHub ist sichergestellt, dass sowohl zukünftige Studierende als auch die Community das System weiterentwickeln, anpassen und nachhaltig verbessern können.

12 Anhang

Datenblätter sowie Projektdaten wie LTspice – Simulationen und die Konstruktionsdaten für Inventor Fusion 360 sind zu finden in „10_Projektarbeit“.

1. **Die Simulation für den Funktionsgenerator:**

10_Projektarbeit → 04_Elektronik → 01_Funktionsgenerator →
01_LTspice_Modell → Funktionsgenerator_Final.asc

2. **Die Simulationen für das Netzteil:**

10_Projektarbeit → 04_Elektronik → 02_Netzteil → 01_LTspice_Modell →
Netzteil_10V.asc und Netztteil_minus10V.asc

3. **Die Simulationen für die Strommessung:**

10_Projektarbeit → 04_Elektronik → 03_Strommessung → 01_LTspice_Modell →
I_Shunt_Messung.asc und I_Shunt_Messung_INA.asc

4. **KiCad-Projekt und gesamter OurDAQ-Schaltplan:**

10_Projektarbeit → 04_Elektronik → 04_KiCad → ourDAQ_THT_4_Lagen

5. **Datenblätter in der Elektronik:**

10_Projektarbeit → 04_Elektronik → 05_Datenblätter in der Elektronik

- a. → 01_MyDAQ von NI
 - i. Bedienungsanleitung und Spezifikationen für NI MyDAQ.pdf
 - ii. mydaq_specifications_2025-02-05-12-09-06.pdf

- b. → 02_Raspberry Pi 5
 - i. Raspberry Pi 5.pdf
 - ii. raspberry-pi-4-datasheet.pdf

- c. → 03_Digilent MCC 118 HAT
 - i. DS-MCC-118.pdf

- d. → 04_Elektronik
 - i. AD8031_8032.pdf
 - ii. ad8479-3119573.pdf
 - iii. ad9833.pdf
 - iv. ADG419-3120308.pdf
 - v. Datenblatt_AD9833_Signalgenerator_Modul.pdf
 - vi. LTI1352.pdf

- vii. REN_ca3130_a_DST_20170525-1997427.pdf
- viii. AD5683R_5682R_5681R_5683-3118438.pdf
- ix. ina190.pdf
- x. Infineon_IRF4905_DataSheet_v01_01_EN-3362811.pdf
- xi. Infineon_IRLZ44N_DataSheet_v01_01_EN-3363447
- xii. opa552.pdf
- xiii. METRA 16S.pdf
- xiv. MCP4922.pdf

6. **Literatur:**

10_Projektarbeit → Literatur

- a. ABC_der_Oszilloskope.pdf
- b. Aktive elektronische Bauelemente.pdf
- c. BMF6+BMMF6_PS_Konstruktionsmethodik.pdf
- d. Das_Raspberry_Pi_Kompendium.pdf
- e. Essential Circuit Analysis Using LTspice.pdf
- f. fehlerrechnung.pdf
- g. High-Current Voltage Output Circuit Using a Precision.pdf
- h. How to Choose a Shunt Resistor.pdf
- i. Leiterplattendesign_skript_A5.pdf
- j. Lineare_Netzwerke_R_C_Filter.pdf
- k. Phase-Compensated, 8-Ch, Multiplexed Data Acquisition
- l. Schutzschaltung.pdf
- m. Sensornetzwerke_in_Theorie_und_Praxis.pdf
- n. signalverarbeitung_2019_06_26.pdf
- o. Tietze_Schenk.pdf
- p. TolerancingStat498B.pdf
- q. Understanding Current Sensing Applications.pdf
- r. Understanding Thermal Dissipation and Design of a Heatsink.pdf
- s. V-Modell-XT-Gesamt.pdf
- t. zickert-2023-leiterplatten.pdf
- u. Zwanger_Messtechnik_2024WS.pdf

7. **Inventor Fusion 360 Gehäuse Daten (Fusion 360 zwingend erforderlich!):**

10_Projektarbeit → 03_Mechanik → 01_Konzept Konstruktion

→ OurDAQ v24 Copy.f3z

13 Literaturverzeichnis

- [1] M. Levesley und P. Culmer. "University of Leeds Teaches First-Year Engineering Students With LabVIEW and NI myDAQ." Zugriff am: 28. August 2025. [Online.] Verfügbar: <https://www.ni.com/en/solutions/academic-research/case-studies/university-of-leeds-teaches-first-year-engineering-students-with-labview-and-ni-mydaq.html>
- [2] VDI Zentrum Ressourceneffizienz. "VDI Zentrum Ressourceneffizienz." Zugriff am: 22. Februar 2025. [Online.] Verfügbar: <https://www.ressource-deutschland.de/leitfaden-re/methoden/konstruktionsmethodik/>
- [3] Der Prozessmanager. "Der Prozessmanager." Zugriff am: 24. Februar 2025. [Online.] Verfügbar: <https://der-prozessmanager.de/aktuell/wissensdatenbank/pareto-prinzip>
- [4] National Instruments. "NI myDAQ - Bedienungsanleitung und Spezifikationen." Zugriff am: 3. September 2025. [Online.] Verfügbar: https://mil.ufl.edu/3111/docs/myDAQ/myDAQ_user_guide_and_specs.pdf
- [5] National Instruments. "myDAQ Specifications." Zugriff am: 24. August 2025. [Online.] Verfügbar: <https://www.ni.com/docs/de-DE/bundle/mydaq-specs/page/specs.html>
- [6] Raspberry Pi Ltd. "Raspberry Pi 5."
- [7] elektor. Zugriff am: 1. September 2025. [Online.] Verfügbar: <https://www.elektor.de/products/raspberry-pi-5-8-gb-ram?srsltid=AfmBOoppTrIJ55QHT5JpyHXS6OH0w10z4mLvWEOWBNmOa3idy7AY2V8>
- [8] MEASUREMENT COMPUTING. "MCC 118: Voltage Measurement DAQ HAT for Raspberry Pi®." Zugriff am: 24. August 2025. [Online.] Verfügbar: <https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=https://files.digilent.com/datasheets/DS-MCC-118.pdf>
- [9] I. Williams und R. Itarsiwala. "How to Choose a Shunt Resistor: TI Precision Labs – Current Sense Amplifiers." Zugriff am: 28. August 2025. [Online.] Verfügbar: <https://www.ti.com/content/dam/videos/external-videos/de-de/4/3816841626001/6076324596001.mp4/subassets/current-sense-amplifiers-how-to-choose-a-shunt-resistor-presentation-quiz.pdf>
- [10] M. Lis. "Understanding Current Sensing Applications & How to Choose the Right Device." Zugriff am: 28.28.2025. [Online.] Verfügbar: <https://www.ti.com/lit/pdf/slap178>
- [11] U. Tietze und C. Schenk, *Halbleiter-Schaltungstechnik* (Springer eBook Collection Computer Science and Engineering). Berlin, Heidelberg, s.l.: Springer Berlin Heidelberg, 1993.
- [12] R. Follmann, *Das Raspberry Pi Kompendium*, 2. Aufl. Berlin, Heidelberg: Springer Vieweg, 2018.
- [13] A. Meroth und P. Sora, *Sensornetzwerke in Theorie und Praxis: Embedded Systems-Projekte erfolgreich realisieren*, 2. Aufl. Wiesbaden, Heidelberg: Springer Vieweg, 2021.
- [14] F. Asadi, *Essential Circuit Analysis using LTspice®*, 1. Aufl. Cham: Springer International Publishing; Imprint Springer, 2023.
- [15] Texas Instruments. "OPA55x High-Voltage, High-Current Operational Amplifiers." Zugriff am: 28. August 2025. [Online.] Verfügbar: <https://www.ti.com/lit/gpn/OPA551>

- [16] S. Leibson. "Grundlagen der Strommessung: Teil 2 – Strommessverstärker." Zugriff am: 26. August 2025. [Online.] Verfügbar: https://www.digikey.de/de/articles/fundamentals-of-current-measurement-part-2-current-sense-amplifiers?srsltid=AfmBOorcmwEjU_o66xviKzbJ14VP6ofNc5w2lsk_ugl4AXSJbbBoPCNx
- [17] Texas Instruments. "INA190 Bidirectional, Low-Power, Zero-Drift, Wide Dynamic Range, Precision Current-Sense Amplifier With Enable." Zugriff am: 28. August 2025. [Online.] Verfügbar: <https://www.ti.com/lit/gpn/INA190>
- [18] F. Scholz, "Statistical Tolerancing," University of Washington STATISTICS, 2007. [Online]. Verfügbar unter: <http://faculty.washington.edu/fscholz/DATAFILES498B2008/TolerancingStat498B.pdf>
- [19] Tektronix. "ABC der Oszilloskope: Einführungshandbuch." Zugriff am: 2. September 2025. [Online.] Verfügbar: https://download.tek.com/document/03G_8605_6_MR_A4.pdf_14_12_2012.pdf
- [20] O. Nelles, "Signalverarbeitung," Universität Siegen, 2019. [Online]. Verfügbar unter: https://www.mb.uni-siegen.de/mrt/lehre/sv/signalverarbeitung_2019_06_26.pdf
- [21] Texas Instruments. "Phase-Compensated, 8-Ch, Multiplexed Data Acquisition System for Power Automation Reference Design." Zugriff am: 26. August 2025. [Online.] Verfügbar: www.ti.com/lit/ug/tidu427b/tidu427b.pdf
- [22] U. K. Sahu. "Analog Engineer's Circuit: High-Current Voltage Output Circuit Using a Precision DAC." [Online.] Verfügbar: <https://www.ti.com/lit/pdf/sbaa345>
- [23] International Rectifier. "IRLZ44NPbf: HEXFET® Power MOSFET." Zugriff am: 28. August 2025. [Online.] Verfügbar: https://www.infineon.com/dgdl/Infineon-IRLZ44N-DataSheet-v01_01-EN.pdf?fileId=5546d462533600a40153567217c32725
- [24] N. Seshasayee. "Understanding Thermal Dissipation and Design of a Heatsink." [Online.] Verfügbar: <https://www.ti.com/lit/pdf/slva462>
- [25] International Rectifier. "IRF4905PbF: HEXFET® Power MOSFET." Zugriff am: 28. August 2025. [Online.] Verfügbar: https://www.infineon.com/dgdl/Infineon-IRF4905-DataSheet-v01_01-EN.pdf?fileId=5546d462533600a4015355e329b1197e
- [26] ANALOG DEVICES. "Data Sheet AD5683R/AD5682R/AD5681R/AD5683." Zugriff am: 27. August 2025. [Online.] Verfügbar: https://www.analog.com/media/en/technical-documentation/data-sheets/AD5683R_5682R_5681R_5683.pdf
- [27] MICROCHIP. "MCP4902/4912/4922: 8/10/12-Bit Dual Voltage Output Digital-to-Analog Converter with SPI Interface." Zugriff am: 2. September 2025. [Online.] Verfügbar: <https://ww1.microchip.com/downloads/en/devicedoc/22250a.pdf>
- [28] M. Zwanger, "Elektrische Messtechnik," Technische Hochschule Nürnberg Georg Simon Ohm, 2024.
- [29] LINEAR TECHNOLOGY. "LT1352/LT1353: Dual and Quad 250µA, 3MHz, 200V/µs Operational Amplifiers." Zugriff am: 3. September 2025. [Online.] Verfügbar: https://www.analog.com/media/en/technical-documentation/data-sheets/1352_53fa.pdf
- [30] L. Stiny, *Aktive elektronische Bauelemente: Aufbau, Struktur, Wirkungsweise, Eigenschaften und praktischer Einsatz diskreter und integrierter Halbleiter-Bauteile*, 2. Aufl. Wiesbaden: Springer Vieweg, 2015. [Online]. Verfügbar unter: <http://gbv.eblib.com/patron/FullRecord.aspx?p=4068633>

- [31] Dr. Hempel, "Script zur Einführung in die Grundlagen der Fehlerrechnung," TU Braunschweig. [Online]. Verfügbar unter: <https://www.tu-braunschweig.de/index.php?eID=download&t=f&f=45992&token=b041478add5f7a3e136b69905b72bfa0e192d82b>
- [32] R. Scholz, "Lineare Netzwerke: R-C-Filter," 2012, Leibniz Universität Hannover. [Online]. Verfügbar unter: https://www.praktikumphysik.uni-hannover.de/fileadmin/praktikumphysik/Versuche/HF/C-Elektrizitaet/C11_HF.pdf
- [33] E. Bluoss, "Leiterplattendesign," Internes Dokument, Technische Hochschule Nürnberg Georg Simon Ohm, 2012.
- [34] W. Fritzsche, "Schutzschaltungen," Technische Universität Berlin - TU Berlin, 2015. [Online]. Verfügbar unter: <https://service.projektlabor.tu-berlin.de/wordpress/wp-content/uploads/sites/11/2015/06/Schutzschaltung.pdf>
- [35] Daniel Wiese. "Schutzdiode an Eingänge." [Online.] Verfügbar: <https://devxplained.eu/de/blog/schutzdiode-an-eingaengen>
- [36] M. F. Robbins. "Ultimate Electronics: Practical Circuit Design and Analysis." Zugriff am: 27. August 2025. [Online.] Verfügbar: <https://ultimateguidebook.com/op-amp-voltage-reference/>
- [37] METRA BLANSKO. "Metra 16S: Digital multimeter." Zugriff am: 28. August 2025. [Online.] Verfügbar: https://www.metra.cz/wp-content/uploads/2020/07/KL_multi16s_AJ-1.pdf
- [38] G. Zickert, *Leiterplatten: Stromlaufplan, Layout und Fertigung*, 3. Aufl. (Hanser eLibrary). München: Hanser, 2023.