

Masterthesis

Entwicklung eines Mixed-Signal-Frontends für die
Validierung von integrierten Halbleitern im Automobil-
Temperaturbereich

Development of a mixed-signal front-end for the
validation of integrated semiconductors in the
automotive temperature range

Vorgelegt an der FH Dortmund
Erstprüfer: Herr Prof. Dr. Karagounis
Zweitprüfer: Herr Walsemann

Verfasser: Edis Salkovic
Dortmund, 30.01.2023

Inhaltsverzeichnis

Abstract.....	4
1. Einleitung.....	5
2. Gliederung	6
3. Grundlagen	7
3.1 Blockdiagram der Analog-Frontend Platine	7
3.2 Serial Peripheral Interface (SPI).....	9
3.3 Prüfung der Fahrzeugelektronik.....	11
3.3.1 Temperaturbereich im Automotive Bereich	12
3.4 Gründe für Messabweichungen	14
4. Umsetzung der ersten Platinenversion.....	19
4.1 Tiefsetzsteller für die 6V-Spannungsversorgung.....	19
4.2 LDO für die 5V Spannungsversorgung.....	20
4.3 LDOs mit variabler Ausgangsspannung	22
4.4 Spannungsgenerierung durch Operationsverstärkerschaltungen	27
4.5 Instrumentenverstärkerschaltungen zur Strommessung.....	30
4.5.1 Strommessung mit Hilfe von Shunt-Widerständen	30
4.6 Pegelumsetzer zur Angleichung unterschiedlicher Logik-Pegel.....	32
4.8 ADS8688-Baustein	34
4.9 AMC7823 ADC/DAC Baustein.....	36
5. Registeroperationen und Befehlsformate.....	38
5.1 Steuerung des ADS8688 ADC Bausteins	38
5.1.1 Programm-Register Lese/Schreibe Operation.....	38
5.1.2 Einstellung des Messbereiches beim ADS8688 ADC Baustein.....	40
5.1.3 Registersatz des ADS8688 ADC Bausteins	43
5.1.3.1 Beschreibung des Command Registers des ADS8688 ADC Bausteins	43
5.1.3.2 Beschreibung des Program Registers des ADS8688 ADC Bausteins.....	44
5.1.4 Betriebsmodi des ADS8688 ADC Bausteins	45
5.1.5 Kanalauswahl des ADS8688 ADC Bausteins	47
5.2 Kommunikation und Steuerung des AMC7823 ADC/DAC-Bausteins.....	49
5.2.1 Command Format des AMC7823 ADC/DAC-Bausteins	49
5.2.2 Lese- und Schreibvorgang im AMC 7823 ADC/DAC Baustein.....	51
6. Im Projekt verwendete Hardware Hardware	59
6.1 Rekonfigurierbare Rechnerplattform auf Basis eines Zedboards.....	59

6.2	Bereitstellung und Messung von Versorgungsspannungen und Signalen	60
7.	Software	61
7.1	Software für Leiterplattenentwurf	61
7.2	Software für den Schaltungsentwurf und Softwareentwicklung	61
7.2.1	Vivado und Vitis	61
7.2.1	Projekterstellung Vitis	62
8	Programmierung der Bausteine	67
8.1	Programmierung des ADC-Bausteins	67
8.1.1	Erstellung der Funktionen zur Auslesung und Steuerung des ADC-Bausteins	67
8.1.2	Funktion für die SPI Datenübertragung	69
8.1.3	Funktionen zum Lesen und Schreiben von Registern	70
8.1.4	Funktion zum Empfang von ADC Spannungswerten	71
8.1.5	Funktion zur Definition der Messbereiche	74
8.1.6	Funktion zur Erfassung von Strömen	76
8.2	Programmierung des AMC7823 ADC/DAC Bausteins	77
8.2.1	Erstellte Funktionen	77
8.2.2	Funktionen zur Konfiguration des AMC7823-Bausteins	79
8.2.3	Funktionen zur Einstellung der DAC-Spannungen	80
8.3	Main()-Funktion zum Test der Funktionalität aller Bausteine	82
9.	Inbetriebnahme der ersten Leiterplattenversion	84
10.	Inbetriebnahme der zweiten Leiterplattenversion	89
10.1	ADS8688 ADC Baustein (1)	89
10.2	ADS8688 ADC Baustein (2) und Instrumentenverstärkerschaltungen	90
10.3	Validierung des ADC/DAC-Bausteins	92
10.4	Layout der zweiten Platinenversion	93
11.	Messabweichung und Korrektur	95
11.1	Projektbezogene Grundlagen	95
11.2	Messungenauigkeit bei der Spannungsmessung des ADS8688-Bausteins (1)	97
11.3	Messungenauigkeit bei der Strommessung am ADS8688-Baustein (2)	98
11.4	Messabweichung am AMC7823-Baustein	100
10.	Fazit	106
	Abbildungsverzeichnis	107
	Digitaler Anhang auf der angehängten CD/USB-Stick	108
	Quellenverzeichnis	108

Abstract

Im Rahmen dieser Masterthesis soll die bereits im Rahmen meiner Masterstudienarbeit entwickelte Frontend-Platine bestückt und im Zusammenspiel mit einem Zedboard in Betrieb genommen werden. Das Zedboard ist mit einem Baustein von Xilinx bestückt, der sowohl einen FPGA als auch einen ARM-Mikrocontroller beinhaltet. Der FPGA-Mikrocontroller wurde bereits so konfiguriert, dass SPI Schnittstellen implementiert sind, die für die Ansteuerung der ADCs und DACs verwendet werden können. Die Aufgabe dieser Masterthesis besteht darin die Software für den ARM-Mikrocontroller unter Petalinux zu schreiben, mit der die SPI Schnittstellen gelesen und beschrieben werden können. Der Softwareteil, welcher wesentlicher Bestandteil der Thesis ist, konnte zufriedenstellend gelöst werden, sodass alle gewünschten Funktionen enthalten sind. Die Frontendplatine aus der Masterstudienarbeit wurde überarbeitet und eine zweite Version angefertigt, welche bis auf einige kleine Fehler gut funktioniert.

Abstract

As part of this master's thesis, the front-end circuit which is already equipped as part of my previous project has to be put into operation in conjunction with a Zedboard. The Zedboard is populated with a Xilinx device that includes both an FPGA and an ARM microcontroller. The FPGA microcontroller has already been configured to implement SPI-Interfaces that can be used to drive the ADCs and DACs. The task of this master's thesis is to write the software for the ARM microcontroller under Petalinux, with which the SPI interfaces can be read and written. The software part, which is an essential part of the thesis, can be solved satisfactorily so that all desired functions are included. The frontend circuit board from the master's thesis was revised and a second version was made, which works well except with a few small errors.

1. Einleitung

„Seit Anfang der 1970er Jahre ist die Entwicklung von Kraftfahrzeugen geprägt von einem rasanten Anstieg des Einsatzes von Elektronik und Software. Dieser Trend hält bis heute an und wird getrieben von steigenden Kunden- und Umweltaforderungen. Nahezu alle Funktionen des Fahrzeugs werden inzwischen elektronisch gesteuert, geregelt oder überwacht“ [1].

Dieser Trend zum vermehrten Einsatz von Automobilen und damit auch des Einsatzes der Elektronik hält an und wird durch die Elektromobilität weiter angetrieben. Die sichere Funktion von Fahrzeugen kann aber nur dann gewährleistet werden, wenn die Ausfallsicherheit und der korrekte Betrieb der Automobilelektronik sichergestellt wird. Hierzu sind eine Reihe von Validierungsschritten nötig, welche Tests in einem weiten Temperaturbereich, beschleunigte Alterung durch Bestrahlung und Prüfung der funktionalen Sicherheit durch Stimulation von Bitfehlern (engl. Single Event Upsets, SEU) durch Bestrahlung mit Photonen und Schwerionen einschließt.

Diese Vorgänge setzen die leichte Installation und Überwachung der zu testenden Automobilelektronik in Großgeräten und Laboren an verschiedenen Standorten voraus. Diese Thesis setzt an dieser Stelle an und hat zum Ziel, durch die Entwicklung eines Front-End-Boards eine Vielzahl von Netzteilen, Multimetern und anderen Geräten am aktuellen Testsystem zu ersetzen und dadurch den Transport zu vereinfachen.

Das Board verfügt über einige integrierte Schaltkreise und bietet viele Möglichkeiten zur Überwachung und Steuerung. Es kommen insbesondere Analog zu Digital-Umsetzer (engl. Analog to Digital Converter, ADC) als auch Digital zu Analog Umsetzer (engl. Digital to Analog Converter, DAC) zum Einsatz. ADCs werden verwendet, um analoge Signale z.B. Versorgungsspannungen und Lastströme zu digitalisieren und im Testsystem durch die ausgeführte Software zu überwachen. DACs werden genutzt um z.B. aus der Software Referenzspannungen zu generieren.

Es sind jeweils acht Kanäle zur Spannungs- und Strommessung ausgelegt. Darüber hinaus sind acht Spannungsversorgungen für unterschiedliche Laststrombereiche und einem besonderen Verfahren für die Bereitstellung und Überwachung derselben vorhanden. Zudem sind zwei feste Stromquellen für Thermistoren vorgesehen.

Die zu entwerfende Software soll Funktionen für das Auslesen und Setzen von Spannungen und Strömen besitzen. Darüber hinaus soll auch die Einstellung des gewünschten Messbereiches und auch eine Offset-Korrektur der Kanäle vorhanden sein. Die Priorität bei der Programmierung liegt darin, eine einfache Nutzung der Funktionen zu ermöglichen und die Bauteile so vorzukonfigurieren, dass der größte Teil der Funktionalität ohne Eingriff des Nutzers ablaufen kann.

2. Gliederung

Nach Vorstellung einiger Projektgrundlagen zur Funktion der Front-End-Platine werden Testverfahren zur Qualifikation der Fahrzeugelektronik, die SPI-Kommunikation und die Behandlung von Messabweichungen vorgestellt. Anschließend wird die Umsetzung der ersten Front-End Platine und die Registeroperationen und Formate mit welchen die eingesetzten Bausteine AMC7823 und ADS8688 angesprochen vorgestellt. Nachdem die verwendete Software und Hardware kurz beschrieben wurde, wird die Entwicklung der für den Betrieb der Platine entwickelten Software erklärt und die Inbetriebnahme der Platine vorgestellt. Die dabei entdeckten Fehler und deren Korrektur zur Erstellung der zweiten Version der Platine werden auch beschrieben, gefolgt von der Inbetriebnahme der zweiten Version der Platine und der Vorgehensweise zur Korrektur vorhandener Messabweichungen an bestimmten Kanälen. Die Vorstellung des Layouts der zweiten Platinenversion und ein Fazit runden die Thesis ab.

3. Grundlagen

3.1 Blockdiagramm der Analog-Frontend Platine

Anhand des Blockschaltbildes (Abbildung 1) soll der Aufbau und die Funktion der Analog-Frontend Platine veranschaulicht werden. Die Analog Platine wird über ein externes Netzteil mit einer konstanten Gleichspannung von 12V versorgt, welche über einen Stecker zum Versorgungspin des Tiefsetzstellers geführt und dort auf 6V heruntergesetzt wird. Dies ist in der oberen linken Ecke der Abbildung zu sehen. Es wurde darauf geachtet, dass der Tiefsetzsteller einen Laststrom liefern kann, der groß genug ist, um alle angeschlossenen Bauteile bei ihrer Maximalauslastung zu versorgen.

Diese 6V wird von vier Low Drop Out (LDO) Spannungsreglern, welche mit dem Analog Monitoring Circuit (AMC) verbunden sind, zur Spannungsversorgung genutzt. Die LDO-Regler sind dem Tiefsetzsteller nachgeschaltet um die 6V Ausgangsspannung des Tiefsetzstellers auf eine 5V Gleichspannung herunterzuregulieren. Der Vorteil dieses Ansatzes besteht darin, dass der LDO Regler die Spannungs-Ripple, welche der Tiefsetzsteller bei der Spannungsumsetzung erzeugt, herausreguliert und damit den angeschlossenen Bauteilen eine geglättete Versorgungsspannung zur Verfügung stellen kann.

Darüber hinaus sollen zwei feste Stromquellen für die Versorgung von Thermistoren integriert werden. Die Analog-Platine verfügt über zwei ADCs vom Typ ADS8688. Ein ADC besitzt 8 Spannungseingänge (0 bis 5V). Diese Kanäle werden für Monitoring von Spannungen bereitgestellt, welche diesem über einen Stecker von außen zugeführt werden können.

Der andere ADC verfügt ebenfalls über 8 Eingänge für die niederohmige Strommessung. Diese Messungen werden über jeweils 4 niederohmige Shunt Widerstände realisiert, deren Spannungsabfall über Instrumentenverstärker den ADC Kanälen zugeführt werden. Die ersten vier Schaltungen sind für die Messung kleiner Ströme von 0 bis 20mA und die anderen vier Schaltungen für die Messung größerer

Ströme von 0 bis 1A ausgelegt. Weiterhin enthält die Schaltung einen AMC vom Typ AMC7823, welcher 8 DAC Ausgänge besitzt. Vier davon werden für die Ansteuerung der Ausgangsspannung von vier LDOs verwendet, welche Spannungen von 0,5 bis 5V bei Lastströmen von bis zu 1A generieren können. Die anderen vier Ausgänge sind mit Operationsverstärkerschaltungen verbunden, welche ebenfalls Spannungen von 0 bis 5V bei kleinen Lastströmen von 0-20mA liefern. Die Versorgungsspannung wird über Senseleitungen direkt am Verbraucher abgegriffen und dem AMC zur Überwachung über die 8 ADC-Kanäle übermittelt, um die Spannungsverluste durch Kabelwiderstände ausgleichen zu können. Zudem werden noch Pegelumsetzer verwendet, damit die Logikspannung von externen Signalen auf die 3,3V-Logikspannung der ADCs und des AMC angepasst wird. Schließlich sind noch zwei 80-Pin Stecker für jeweils analoge und digitale Signale vorgesehen.

Sowohl der ADS8688 als auch der AMC7823 kommunizieren über eine SPI-Schnittstelle mit dem Entwicklungsboard (Zedboard).

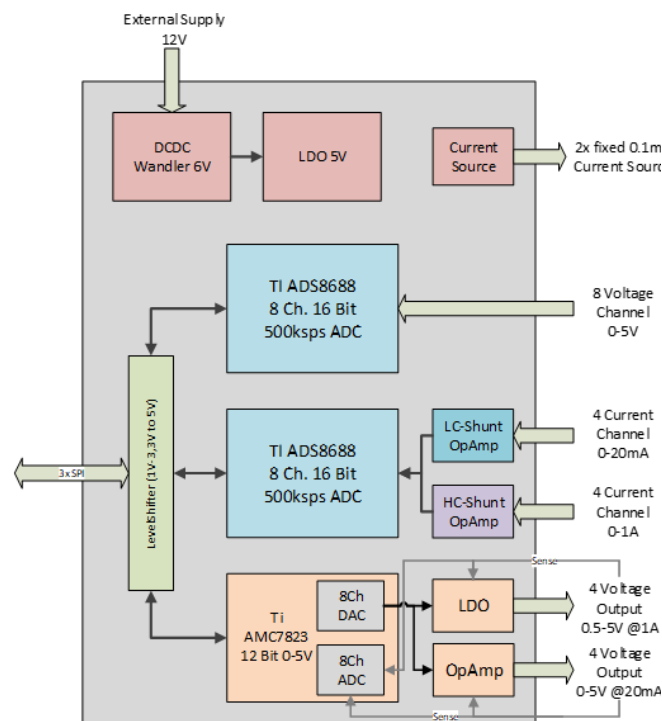


Abbildung 1: Blockdiagramm der Mess- und Versorgungselektronik [2]

3.2 Serial Peripheral Interface (SPI)

Das Zedboard besitzt drei SPI-Schnittstellen, über welche es mit den zwei ADS8688 und dem AMC7823 Bausteinen kommunizieren kann. In diesem Fall bilden sie die Slaves während der ARM Prozessor des Zedboards die Rolle des Master einnimmt. Für die SPI-Kommunikation werden drei Leitungen benötigt. Über die SDO-Leitung (Slave Data Out) antwortet der Slave auf Anfragen des Masters, welche er über die SDI-Leitung (Slave-Data-In) an den Slave sendet. Zudem gibt es noch die SCK-Leitung (Serial Clock für den Schiebetakt) für die Synchronisation der Kommunikation. Zusätzlich hat jeder Slave eine SS bzw. CS (Chip Select) Leitung, welche aktiv Low ist. Diese Signale werden für die Auswahl des Slaves verwendet, mit welchem kommuniziert werden soll, da es mehrere Slaves geben kann.

Es gibt mehrere Übertragungs-Modi mit denen die SPI Kommunikation ablaufen kann. CPOL (Clock Polarity) und CPHA (Clock Phase) nach Bedarf entsprechend der in der Tabelle 1 dargestellten Möglichkeiten im Register des Controllers eingestellt. Dabei bestimmt CPOL ob das SCLK im Ruhezustand auf dem Low-Pegel, oder dem High-Pegel liegt. CPHA bestimmt, ob die Daten bei der ersten steigenden Flanke des SCLK-Signals übernommen werden, oder bei der fallenden Flanke.

Im Projekt wird der Mode 1, also die Einstellung CPOL=0 und CPHA=1 gewählt. Laut Datenblatt des AMC7823-Bausteins ist der Mode 1 zu wählen ([3], Seite 20). Im Datenblatt des ADS8688-Bausteins wird kein Mode vorgeschlagen, jedoch befindet sich das SCLK-Signal im Ruhezustand auf Low und die Übertragung startet bei der zweiten (fallenden) Flanke des SCLK-Signals, somit ist der Mode 1 auch für diesen Baustein die richtige Wahl ([4], Seite 35). Chips entsprechend des Datenblattes für die Kommunikation in diesem Mode ausgelegt sind. Abbildung 2 zeigt den Unterschied zwischen dem Mode 0 und dem Mode 1. Das SCLK Signal liegt bei bei CPOL=0 im Ruhezustand auf dem Low-Pegel. Die blauen Striche in der Abbildung zeigen wie das Sampling des Signals an der SDO-Leitung an zweiten fallenden Flanke mittig im Bit stattfindet, während es im MODE 0 (rote Striche) bereits eine Flanke früher passiert. Indem die CS-Leitung auf den High-Pegel gezogen wird, endet die Kommunikation.

SPI-Mode	CPOL	CPHA
0	0	0
1	0	1
2	1	0
3	1	1

Tabelle 1: SPI Übertragungs-Modi

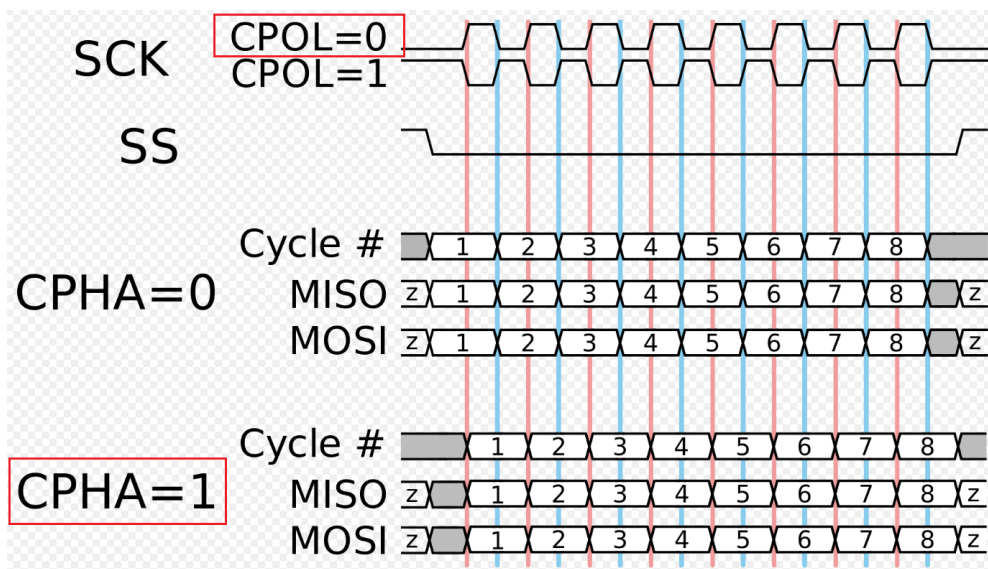


Abbildung 2: SPI-Kommunikation im Mode0 und Mode1 [5]

3.3 Prüfung der Fahrzeugelektronik

Fahrzeugelektronik wird in sogenannten DUT (Device under Test) -Testverfahren geprüft, indem diese in speziellen Kammern gelegt und deren Verhalten bei gewünschten Temperaturen bewertet wird. Dafür sind eine Ansteuerung und Auslesung der Bauteile von außerhalb der Kammer notwendig. Diesen Zweck erfüllt die Front-End-Platine, welche im Rahmen dieser Thesis entwickelt worden ist. Für die Temperaturregelung wird der Thermostream (Abbildung 3) von Temptonic verwendet, welcher sehr schnelle Temperaturänderungen von 40°C/min bei Temperaturen von -100°C bis 300°C ermöglicht. Zudem ist dieser mit +-1°C Offsettemperatur sehr präzise und verhindert durch einen ständigen Strom trockener Luft die Bildung von Kondenswasser, welche die Elektronik während des Tests schädigen könnte.

Die Bedienung erfolgt entweder über ein integriertes Panel, oder eine Software-Schnittstelle. Der eigentliche Wärme- bzw. Kältestrahl verläuft über eine flexible Rohrleitung zum Arbeitstisch, deren Ende mechanisch zum Versuchsaufbau geführt werden kann. Das Testobjekt befindet sich in der abgebildeten Box in der zwei Löcher vorhanden sind. In das Loch an der Oberseite wird die Düse eingeführt, aus welcher der temperaturregulierte Luftstrom des Thermostreams austritt. Aus dem Loch an der linken Seite der Box tritt die Luft wieder aus. Die Kabel, welche für den Anschluss des Testobjekts benötigt werden, werden von einer weiteren mit Schaumstoff isolierten Öffnung an der rechten Seite der Box zugeführt. Die Steuerungs- und Versorgungseinheit des Testobjekts liegt also außerhalb des temperaturgeregelten Volumens und muss dementsprechend nicht die gleichen Anforderungen wie das Testobjekt in Bezug auf den Temperaturbereich erfüllen.



Abbildung 3: ThermoStream-Aufbau

3.3.1 Temperaturbereich im Automotive Bereich

Laut der AEC-Q100 Regelung des Automotive Electronics Council (AEC) existieren 4 definierte Temperaturbereiche (Tabelle 2) in denen ICs im Automotivebereich als Device under Test getestet werden [6].

Grad	Operativer Temperaturbereich in °C
0	-40 bis +150
1	-40 bis +125
2	-40 bis +105
3	-40 bis +85

Tabelle 2: Operativer Temperaturbereich für verschiedene Grade

Die in diesem Projekt verwendeten ICs haben die für den operativen Betrieb zulässigen Temperaturbereiche entsprechend der Tabelle 3.

Bauteil	Temperaturbereich in °C
ADC	-40 bis +125
AMC	-40 bis +105
DCDC	-40 bis +125
LDO	-40 bis +150
Operationsverstärker (OPA)	-40 bis +150
Instrumentenverstärker (INV)	-40 bis +85
LEVELSHIFTER	-40 bis +125
Inverter	-40 bis +125

Tabelle 3: Operativer Temperaturbereich der verwendeten integrierten Schaltkreise

Dadurch entspricht das Analog-Frontend dem Grad 3. Durch Verwendung eines alternativen Instrumentenverstärkers mit einem Temperaturbereich bis +105°C könnte der Grad 2 erreicht werden. Bei Verwendung eines Instrumentenverstärkers mit einem Temperaturbereich bis +125°C und eines AMC mit einem Temperaturbereich bis +125°C könnte sogar Grad 1 erreicht werden, was durch Recherche in vielen Datenblättern und Dokumentationen als gängiger Temperaturbereich im Automotivbereich erkannt wurde. Allgemein ist es ratsam Bauteile mit einem möglichst weitem Temperaturbereich zu wählen, um die Lebenszeit des Systems zu verlängern. Die Alterung von elektronischen Bauteilen nimmt bei hohen Temperaturen zu. Bei 100°C hat beispielsweise ein IC, welcher bis 105°C ausgelegt ist, eine kürzere Lebenserwartung als ein IC, welcher bis 150°C zugelassen ist.

Für die Testplatine, welche nur im Labor eingesetzt wird, ist der Temperaturbereich bis 85°C völlig ausreichend. Bei der Entwicklung von Systemen für die Nutzung im Fahrzeug müsste mindestens der Grad 1 erreicht werden.

3.4 Gründe für Messabweichungen

Unter idealen Umständen gäbe es keine Abweichung von Messwerten zu den tatsächlichen Werten. Unter realen Umständen gibt es jedoch Messabweichungen, welche eine Vielzahl von Gründen haben können, die hier teilweise aufgeführt werden. Für die Messabweichung ist zunächst der Begriff LSB (Least Significant Bit), also das niedrigwertigste Bit bzw. der niedrigste darstellbare Wert zu erläutern, mit welchem die Messabweichungen im Kontext der Analog/Digital und Digital/Analog Wandlung beschrieben werden. Ein LSB berechnet sich durch die Übertragungsfunktion eines ADC mit Hilfe von Formel 6.1. Dabei entspricht FS dem Full Scale, also dem Höchstwert, welcher typischerweise den Wert der Referenzspannung besitzt. FSR (Full Scale Range) steht für den gesamten Wertebereich. Beim AMC7823 mit 12-Bit Auflösung und einem Maximalwert von 5V ergibt sich nach Formel 6.1 ein LSB von 1.22mV und für den ADS8688 mit einem Maximalwert von 5,12V nach Formel 6.2 ein LSB von 0,08mV [7].

$$LSB_{amc} = \frac{FS}{2^n - 1} = \frac{5V}{2^{12} - 1} = 1.22mV \quad 6.1$$

$$LSB_{ads} = \frac{FS}{2^n - 1} = \frac{5,12V}{2^{16} - 1} = 0.08mV \quad 6.2$$

Bei der Digitalisierung des analogen Werts muss dieser einem passenden diskreten Wert (1 LSB ; 2LSB etc.) zugeordnet werden. Dabei muss entweder aufgerundet, oder abgerundet werden. Somit ergibt sich ein Digitalisierungsfehler in der Praxis von +- 1LSB.

Weiterhin kann sich auch ein Offset-Fehler einstellen. Dabei geht es beim A/D-Wandler darum, bei welcher Eingangsspannung noch ein digitaler Wert von 0 ausgegeben wird. Im Falle des ADS8688 wird bei einer Eingangsspannung von +/- 0.5-1mV noch ein digitaler Wert von 0 ausgegeben. Beim D/A-Wandler beschreibt der Offsetfehler, welche Spannung sich bei einem digitalen Wert von 0 einstellt. Beim AMC7823 sind es +/-2LSB.

Zudem sind oft auch Verstärkungsfehler zu beobachten. Im Gegensatz zum Offsetfehler, welcher den Beginn der ADC Kennlinie bewertet, beschreibt der Verstärkungsfehler die Steigung der ADC Kennlinie. Bei der ADC-Wandlung ist dabei der Unterschied zwischen dem Nennwert und dem tatsächlichen Wert der Eingangsspannung beim höchsten digitalen Wert gemeint. Beim ADS8688 beträgt dieser Fehler typischerweise $\pm 0.002 \cdot \text{FSR}$, jedoch maximal $\pm 0.005 \cdot \text{FSR}$. Bei einem Wertebereich von +5V kann sich also ein Fehler von 25mV einstellen. Bei der ADC-Wandlung des AMC7823 ist ein Fehler von maximal $\pm 6 \text{LSB}$ zu erwarten, was bei einem LSB von 1,22mV einen Wert von $\pm 7,32 \text{mV}$ entspricht. Bei der DAC-Wandlung ist mit dem Verstärkungsfehler die Differenz zwischen Ausgangsspannung und Nennwert beim höchsten Digitalwert gemeint und beträgt beim verwendeten Baustein AMC7823 max. 1% von FS, was bei FS=5V etwa 5mV entspricht. Dabei wird ein Offset-Fehler von 0mV angenommen.

Des Weiteren gibt es die Differentielle Nichtlinearität (DNL) und die Integrale Nichtlinearität (INL), welche Metriken für die Abweichung von der idealen Übertragungsfunktion darstellen. Die DNL beschreibt bei der AD-Wandlung die Abweichung von den idealen Treppenstufen. Abbildung 4 veranschaulicht eine zu kleine und eine zu große Treppenstufe. Beim Baustein ADS8688 ist eine maximale DNL von 1.5 LSB und beim Baustein AMC7823 eine maximale DNL von 1.25 LSB spezifiziert. Bei der DA-Wandlung ist mit DNL die maximale Abweichung zwischen den aufeinanderfolgenden diskreten Spannungen gemeint, welche beim AMC7823 maximal $\pm 1 \text{LSB}$ beträgt. Auch dieser Fehler ist in Abbildung 5 veranschaulicht, wo der Sprung von 001 auf 010 um $\frac{1}{4} \text{LSB}$ zu groß und um $-\frac{1}{4} \text{LSB}$ zu gering ist [7].

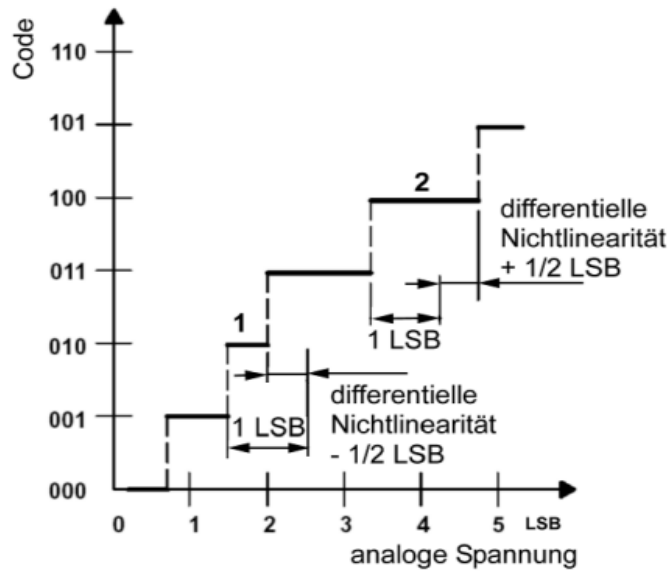


Abbildung 5: DNL-D/A-Wandlung [7]

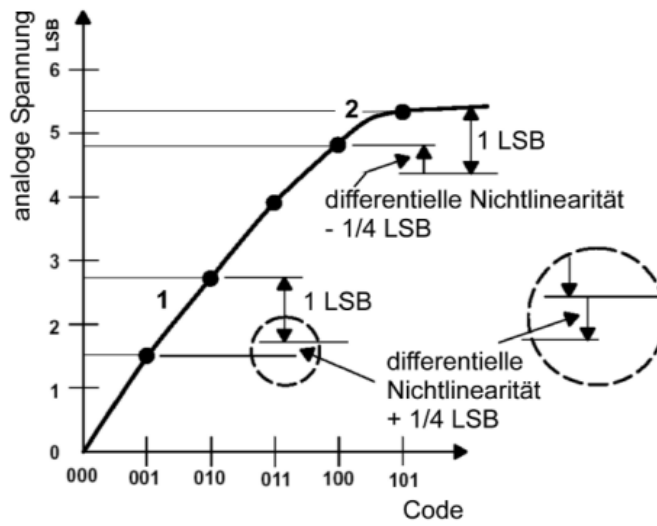


Abbildung 4: DNL A/D-Wandlung [7]

Die Integrale Nichtlinearität (INL) beschreibt beim ADC die Abweichung beim Übergang von einer Treppenstufe zur nächsten. In Abbildung 6 ist diese Situation veranschaulicht, wo sowohl der reale Übergang als auch der ideale Übergang dargestellt sind. Beim ADS8688 beträgt der INL Fehler maximal 2 LSB und beim AMS7823 maximal 1,25 LSB. Beim DA-Wandler wird mit INL die Abweichung der einzelnen diskreten Spannungen beschrieben, was in Abbildung 7 dargestellt wird. Beim AMC7823 beträgt der maximale INL Fehler +/-8 LSB, wobei +/-2 LSB typisch sind [7].

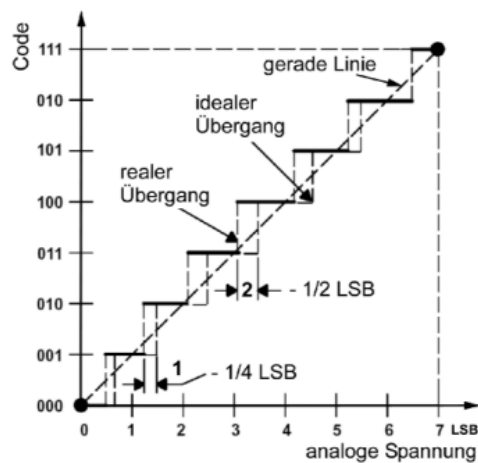


Abbildung 6: INL A/D-Wandlung[7]

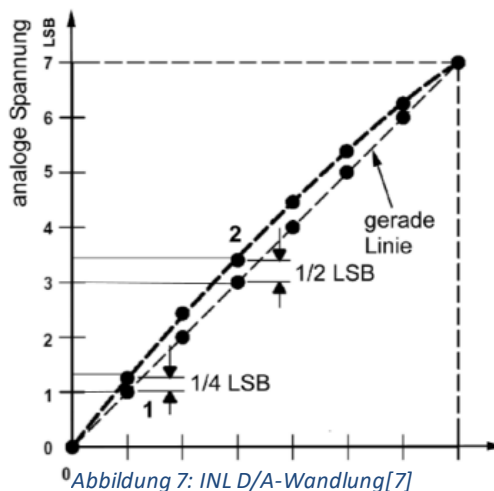


Abbildung 7: INL D/A-Wandlung[7]

Die Zusammenfassung der genannten Fehler spiegelt sich im Total/Absolute Error wider. Es gibt noch weitere Fehler, welche hauptsächlich die AD-Wandlung betreffen. Das wäre der Quantisierungsfehler, Endwertfehler, Hysterese Fehler und die sogenannte Monotonizität [7].

Diese Vielzahl an Ursachen verfälscht in Folge die Ergebnisse. Um trotzdem eine hohe Genauigkeit zu erreichen, ist es möglich durch eine Kalibrierung mit hochpräzisen Messgeräten eine Korrektur der Abweichung durchzuführen.

4. Umsetzung der ersten Platinenversion

4.1 Tiefsetzsteller für die 6V-Spannungsversorgung

Die Schaltung des Tiefsetzstellers wurde mit dem Webench Tool von Texas Instruments dimensioniert. Dabei werden die benötigten Parameter wie z.B. der Wandlungsfaktor und der maximale Laststrom angegeben, während das Tool die benötigte Induktivität, Eingangs- und Ausgangskapazitäten, als auch die Feedback-Widerstände und den Typ der Schottky Diode ermittelt. In diesem Fall wurde eine Eingangsspannung von 12V, eine Ausgangsspannung von 6V und eine maximale Stromstärke von 5A vorgegeben. Ein Laststrom von 5 A wurde gewählt, da die 4 LDOs im Stande sein müssen bis 1A zu liefern. Die vier Operationsverstärker liefern zusammen maximal 80mA. Damit liegt der maximal benötigte Laststrom des DCDC-Wandlers auch bei Vollaustattung unter 5A. An Hand der generierten Ergebnisse wurde die in Abbildung 8 dargestellte Schaltung entworfen, welche bis auf eine leicht erhöhte Eingangsimpedanz von 10uF, statt der empfohlenen 4,7uF für eine noch bessere Glättung des Eingangssignals im Schaltungsentwurf von Altium Designer so übernommen wurde.

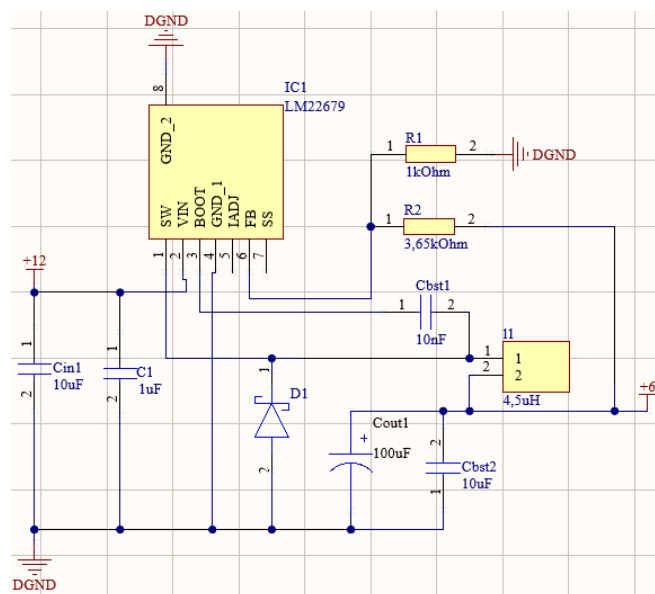


Abbildung 8: Beschaltung des Tiefsetzstellers

4.2 LDO für die 5V Spannungsversorgung

Für die 5V Versorgungsspannung wurde ein LDO vom Typ TPS746-Q1 mit einer einstellbaren Ausgangsspannung zwischen 0.50V bis 5.5V gewählt. Die vereinfachte Darstellung aus dem Datenblatt [8], auf welcher der Schaltplan beruht, ist in Abbildung 9 zu sehen.

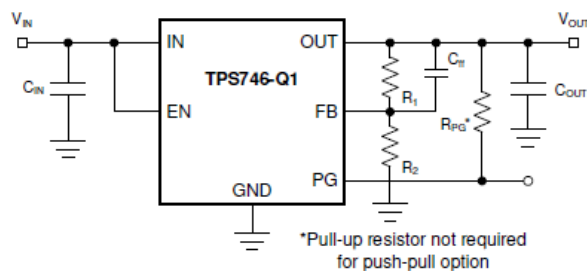


Abbildung 9: Vereinfachtes Schaltbild vom LDO mit fester Ausgangsspannung [8]

Um die Widerstände R1 (R3 in Abb. 10) und R2 (R4 in Abb. 10) des Spannungsteilers zu berechnen, müssen die folgenden Gleichungen aus dem Datenblatt [8] verwendet werden, wobei V_{OUT} für die generierte Ausgangsspannung steht, V_{FB} für die Referenzspannung und I_{FB} für den Referenzstrom:

$$V_{OUT} = V_{FB} \times (1 + R_1 / R_2) \quad (4.1)$$

$$R_1 + R_2 \leq V_{OUT} / (I_{FB} \times 100) \quad (4.2)$$

Dabei ist laut Datenblatt [8] (Seite 18) $V_{FB}=0,55V$ (typ.) und $I_{FB} = 10nA$ (typ.).

$$V_{OUT} = V_{FB} \times (1 + R_1 / R_2) \quad (4.3)$$

$$R_1 = R_2 \times \left(\frac{V_{OUT}}{V_{FB}} - 1 \right) \quad (4.4)$$

Beispielsweise ergibt sich für den Widerstand R2 aus der Gleichung 4.5 der Wert für R1 aus Gleichung 4.7.

$$\underline{R_2 = 4,32k\Omega} \quad (4.5)$$

$$R_1 = 4,32k\Omega \times \left(\frac{5V}{0,55V} - 1 \right) \quad (4.6)$$

$$\underline{R_1 = 34,8k\Omega} \quad (4.7)$$

Damit ist laut Datenblatt die Bedingung erfüllt um den Feedback Pin Stromfehler ignorieren zu können:

$$R_1 + R_2 \leq V_{OUT} / (I_{FB} \times 100) \quad (4.8)$$

$$34,8k\Omega + 4,32k\Omega \leq 5V / (10nA \times 100) \quad (4.9)$$

$$39,12k\Omega \leq 5M\Omega \quad (4.10)$$

Entsprechend der Empfehlung für den gewählten LDO mit einstellbarer Ausgangsspannung im Datenblatt wurde ein Feed-Forward-Capacitor, welcher im Schaltbild als Cff1 bezeichnet wird (Abb. 10) vom Pin Out der Ausgangsspannung zum Rückführungs-PIN FB geführt. Dieser Kondensator soll für eine bessere Transienten-, Rausch- und PSRR Performance sorgen (S.22, Datenblatt [8]).

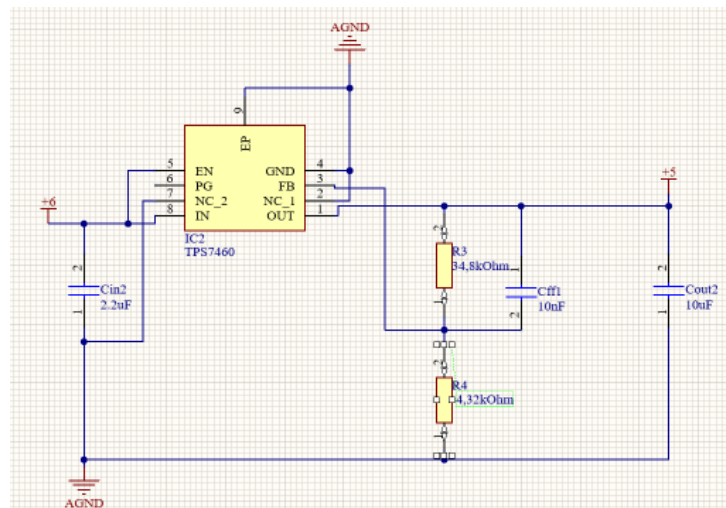


Abbildung 10: Beschaltung des LDOs mit fester Ausgangsspannung

4.3 LDOs mit variabler Ausgangsspannung

Während ein LDO eine konstante Ausgangsspannung von 5V generieren soll, werden 4 LDOs für die Erzeugung von variablen Ausgangsspannungen eingesetzt.

Üblicherweise kann ein LDO aufgrund seines voreingestellten Spannungsteilers mit fest eingestelltem Spannungsteilerverhältniss in seinem Rückkopplungsweig nur eine einzige Spannung erzeugen. Durch die besondere Verschaltung der vier LDOs ist es aber möglich, mit Hilfe von DAC-Kanälen die Ausgangsspannung der LDOs zu verändern. In Abbildung 11 wird dies veranschaulicht. Jeweils ein DAC- wird einem FB-Pin eines LDOs zugeführt und übt dort einen entsprechenden Einfluss aus.

Die auf dem FB Pin anliegende Spannung wird mit einer internen Spannungsreferenz verglichen und ein regelbares Element meist ein PMOS oder PNP Transistor zwischen Eingang und Ausgang des LDOs so eingestellt, dass sich eine Ausgangsspannung ergibt, die nach der Spannungsteilung dem Spannungswert der internen Referenz entspricht. Die LDO Ausgangsspannung folgt dann aus dem Spannungsteiler Verhältnis und der internen Referenzspannung des LDOs. Die DAC Spannung beeinflusst die Rückführspannung am Spannungsteiler des LDOs in dem Sie einen Offset einführt, der die Ausgangsspannung des LDOs je nach Vorzeichen anhebt oder reduziert.

Um die LDOs über die Enable Pins ein- und ausschalten zu können, wird jeweils ein GPIO Signal des AMC verwendet. Das Enable-Signal bleibt Low, solange die gewünschte Spannung des AMC, welche die Ausgangsspannung des LDO beeinflusst nicht am LDO anliegt. Erst dann wird der GPIO Pin auf High geschaltet und damit der LDO eingeschaltet. Die generierte Ausgangsspannung wird abgegriffen, überwacht und der Spannungsabfall durch eventuelle Kabelverluste am Verbraucher durch Sensing ausgeglichen. Der Sense- Widerstand ist in der Schaltung (siehe Abb. 12) als R5 bezeichnet und besitzt einen Wert von 220Ohm.

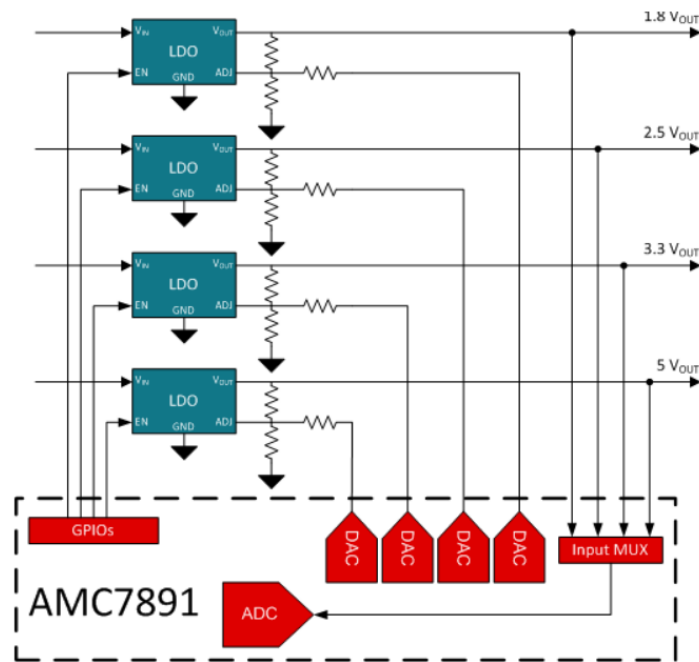


Abbildung 11: Spannungsregelung- und Überwachung der vier LDOs mit variabler Ausgangsspannung [9]

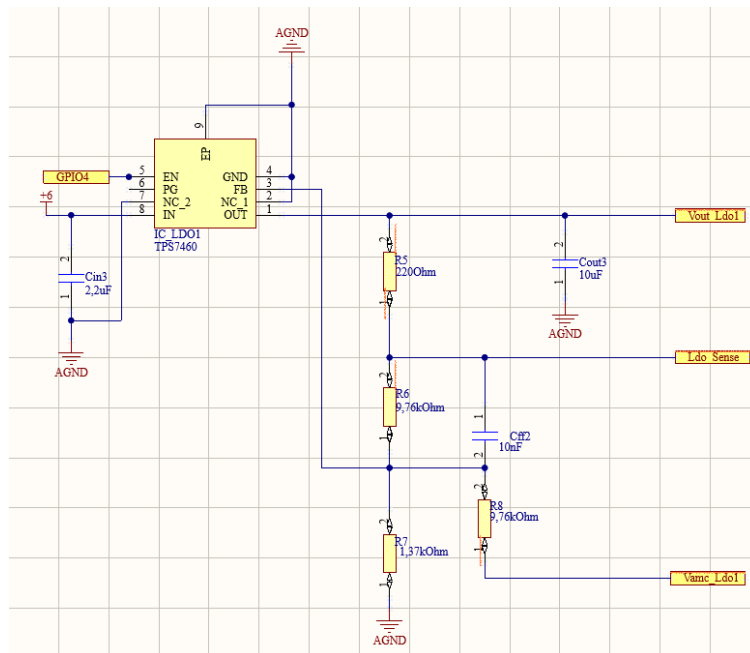


Abbildung 12: Beschaltung der vier LDOs mit variabler Ausgangsspannung

Für die Dimensionierung des Spannungsteilers und des Vorwiderstandes über den der DAC Port eingekoppelt wird, wurde zunächst eine vereinfachte Zeichnung angefertigt (Abbildung 13) und die elektrischen Zusammenhänge mathematisch ausgedrückt. Hierbei ist zu beachten, dass die Widerstände R6 und R7 des Spannungsteilers aus Abbildung 12 in Abbildung 13 mit R1 und R2 bezeichnet werden.

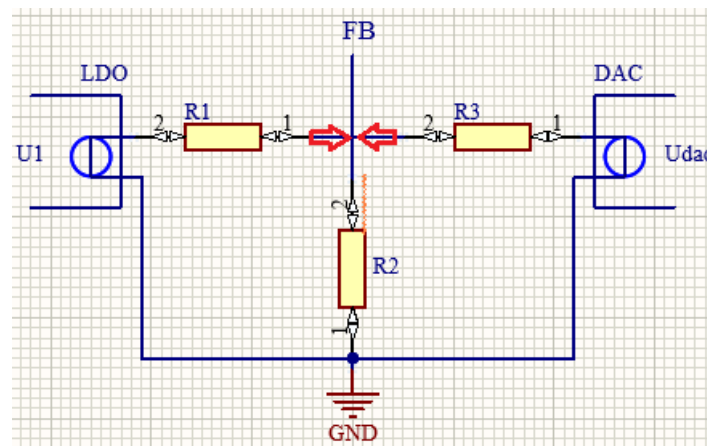


Abbildung 13: Spannungsteiler für LDOs mit variabler Ausgangsspannung

Durch Anwendung der Knotenregel auf den Knoten FB ergibt sich:

$$\left(\frac{U1-U_{FB}}{R1}\right) + \left(\frac{U_{DAC}-U_{FB}}{R3}\right) - \left(\frac{U_{FB}}{R2}\right) = 0 \quad (4.11)$$

Mit U1 ist hierbei die Ausgangsspannung des LDO, mit UDAC die Ausgangsspannung des DACs und mit UFB die Mittenspannung des Spannungsteilers gemeint. Bei aktiver Regelung des LDO wird U1 so eingestellt, dass die Spannung UFB gleich der Spannung der internen Referenz UBG ist.

$$U_{FB} = U_{BG} \quad (4.12)$$

$$\left(\frac{U1-U_{BG}}{R1}\right) + \left(\frac{U_{DAC}-U_{BG}}{R3}\right) - \left(\frac{U_{BG}}{R2}\right) = 0 \quad (4.13)$$

Wird die Gleichung auf beiden Seiten mit R1 multipliziert ergibt sich:

$$U1 - U_{BG} = R1 * \left(\frac{-U_{DAC}+U_{BG}}{R3}\right) + \frac{U_{BG}*R1}{R2} \quad (4.14)$$

Wird die Gleichung auf beiden Seiten mit UBG addiert, folgt:

$$U1 = UBG + R1 * \left(\frac{UBG - UDAC}{R3} + \frac{UBG}{R2} \right) \quad (4.15)$$

Ausklammern von UBG auf der rechten Seite der Gleichung führt zu:

$$U1 = UBG * \left(1 + \frac{R1}{R3} + \frac{R1}{R2} \right) - \frac{UDAC}{R3} * R1 \quad (4.16)$$

Zunächst soll sichergestellt werden, dass sich die maximale LDO-Ausgangsspannung von 5V einstellt, wenn die DAC Spannung UDAC auf null gesetzt wird.

Mit $R1=R3=9,76k\Omega$ und $UBG=0,55V$ ergibt sich:

$$U1 = UBG * \left(2 + \frac{R1}{R2} \right) \quad (4.17)$$

Laut Datenblatt gilt für die interne Referenz $UBG=0,55V$

$$U1 = 0,55V \left(2 + \frac{9,76k\Omega}{R2} \right) \quad (4.18)$$

Nach Auflösung der Klammern ergibt sich:

$$U1 = 1,1V + \frac{0,55 * 9,76k\Omega}{R2} \quad (4.19)$$

Für U1 wird die gewünschte LDO-Ausgangsspannung von 5V angesetzt.

$$5V - 1,1V = \frac{0,55 * 9,76k\Omega * V}{R2} \quad (4.20)$$

Durch Multiplikation mit R2 ergibt sich:

$$3,9V * R2 = 0,55 * 9,76k\Omega \quad (4.21)$$

Nach Auflösung der Gleichung nach R2 ergibt sich:

$$R2 = \frac{0,55 * 9,76k\Omega}{3,9V} \quad (4.22)$$

Für den zweiten Widerstand des Spannungsteilers muss also gelten:

$$R2 = 1,376k\Omega \quad (4.23)$$

In dieser Konstellation mit $UDAC=0V$ muss der DAC Kanal den Strom aufnehmen, der durch den Widerstand R3 fließt. Dieser Strom berechnet sich zu:

$$I = \frac{U_3}{R_3} = \frac{0,55V}{9,76k\Omega} = 56,3\mu A \quad (4.23)$$

Wenn der DAC Kanal die maximal mögliche Spannung von 5V generiert, muss der DAC folgenden Strom liefern können.

$$I = \frac{U_{DAC} - U_{BG}}{R_3} = \frac{5V - 0,55V}{9,76k\Omega} = 0,46mA \quad (4.25)$$

Wenn der DAC die maximale Spannung von 5V generiert, wird entsprechend Gleichung 4.1.6 die Spannung nach 0V geregelt. Wenn der DAC, wie es bei obiger Rechnung für die Ermittlung des Widerstands R2 der Fall war, eine Spannung von 0V ausgibt, wird die maximale Spannung von 5V vom LDO bereitgestellt.

4.4 Spannungsgenerierung durch Operationsverstärkerschaltungen

Vier Operationsverstärker werden für die Generierung variabler Ausgangsspannungen zwischen 0 bis 5V verwendet. Die Ausgangsspannungen können wie bei den vier LDOs über die DAC-Kanäle des AMC (Vamc-Port, Abbildung 15) gesteuert werden. Die maximalen Ströme sind deutlich geringer als bei den LDOs und auf 20mA begrenzt. Wie bei den LDOs sind auch Sense-Leitungen vorgesehen. Die Schaltung funktioniert als sogenannter Voltage Buffer. Die Ausgangsspannung folgt dabei der Eingangsspannung $V_{out}=V_{in}$ mit dem Verstärkungsfaktor 1. Der Vorteil dieser Schaltung besteht darin, dass die vom AMC vorgegebene Spannung von der Last entkoppelt wird und dadurch nicht durch Belastung verändert wird oder zusammenbricht.

Die Eingangsspannung, welche vom AMC generiert wird, wird zum nicht invertierenden Eingang (+) geführt und die Ausgangsspannung wird auf den invertierenden Eingang (-) zurückgekoppelt (siehe Abb. 14). Wenn nach einem Power-Up die Ausgangsspannung des Verstärkers auf 0V liegt, erkennt der OPA eine höhere Spannung am nicht-invertierenden Eingang als am mit dem Ausgang verbundenem invertierenden Eingang ($V_{+}>V_{-}$), sodass sich die Ausgangsspannung des Verstärkers erhöht, bis die Spannungen an den Eingängen angleichen [10].

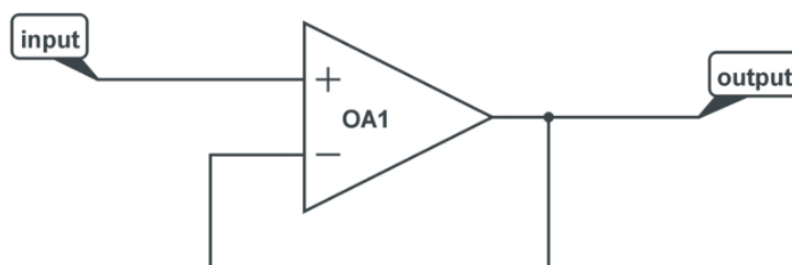


Abbildung 14: Vereinfachte Voltage-Buffer Schaltung [10]

Für die Ausgangsspannung des Operationsverstärkers gilt:

$$V_{out} = A * (V - (V-)) \quad (4.26)$$

wobei A der differentiellen Leerlaufverstärkung des OPV entspricht und üblicherweise sehr große Werte wie z.B. 120dB erreichen kann. Wird $V_+ = V_{in}$ und $V_- = V_{out}$ ersetzt ergibt sich:

$$V_{out} = A(V_{in} - V_{out}) \quad (4.27)$$

Nach V_{out} aufgelöst ergibt sich:

$$V_{out} = \left(\frac{A}{1+A} \right) * V_{in} \quad (4.28)$$

Mit der Vorstellung eines idealen OPA mit unendlich großer Leerlaufverstärkung $A \rightarrow \infty$ ergibt sich:

$$V_{out} = V_{in} \quad (4.29)$$

Der invertierende und nichtinvertierende Eingang des OPVs nehmen also die gleiche Spannung an, obwohl sie nicht direkt miteinander verbunden sind. Diese Konstellation wird auch als virtueller Kurzschluss bezeichnet und stellt sich immer bei OPV Schaltungen mit negativer Rückkopplung und hoher Leerlaufverstärkung ein. In diesem Projekt wird ein RC-Tiefpass-Filter zur Reduzierung des Rauschens in den invertierenden Pfad integriert (siehe Abb. 15).

Die Cutt-Off-Frequenz f_c eines RC-Glieds kann wie folgt berechnet werden:

$$f_c = \frac{1}{2\pi * \tau} \quad (4.30)$$

wobei τ der Zeitkonstante des RC-Filters entspricht und wie folgt berechnet werden kann:

$$\tau = R * C \quad (4.31)$$

Für die Anstiegszeit eines RC Gliedes mit der Anstiegszeit τ gilt:

$$trise = 2,2 * \tau \quad (4.32)$$

Löst man die obige Gleichung nach der Anstiegszeit auf und setzt sie in die Gleichung für die Cut-Off Frequenz ein, ergibt sich folgender Ausdruck:

$$f_c = \frac{2,2}{2\pi * trise} \quad (4.33)$$

Für eine Anstiegszeit von $trise=1ms$ ergibt sich eine Cut-Off Frequenz von :

$$f_c = \frac{2,2}{2\pi * 1ms} \quad (4.34)$$

$$f_c = 350Hz \quad (4.35)$$

Zur Einstellung der Anstiegszeit wird Formel 4.32 für die Anstiegszeit zunächst nach dem Widerstand R aufgelöst.

$$R = \frac{trise}{2,2 * C} \quad (4.36)$$

Wird der Kondensator, wie im Datenblatt empfohlen, mit $C=1nF$ gewählt ergibt sich für den Widerstand:

$$R = \frac{1ms}{2,2 * 1nF} \quad (4.37)$$

$$R \approx 500k\Omega \quad (4.48)$$

Wie bei den LDOs ist auch bei den OPA Schaltungen ein Widerstand im Sense-Pfad vorgesehen. Wenn keine Sense-Leitung angeschlossen ist, wird über den Widerstand die Spannung direkt am OPA Ausgang statt an der Last abgegriffen.

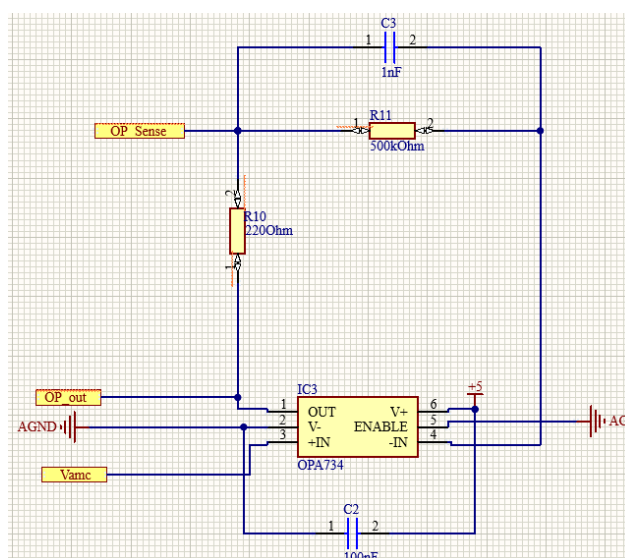


Abbildung 15: Beschaltung der OPAs

4.5 Instrumentenverstärkerschaltungen zur Strommessung

Mit Hilfe der Instrumentenverstärkerschaltung sollen acht präzise Strommessungen über die analogen Kanäle des zweiten ADC-Bausteins ermöglicht werden. Vier Ströme sollen hierbei einen maximalen Wert von bis zu 20mA und die anderen vier Ströme einen Wert von bis zu 1A erreichen können.

4.5.1 Strommessung mit Hilfe von Shunt-Widerständen

Für die Strommessung wird ein Präzisionswiderstand (Shunt) in den Strompfad platziert [11]. Die Spannung, welche an dem Widerstand abfällt, ist proportional zum durchfließenden Strom und wird gemessen. Um den Stromfluss nicht bzw. kaum zu beeinträchtigen, wird der Shunt Widerstand sehr klein gewählt. Das führt allerdings auch dazu, dass die abgefallene Spannung sehr klein ist und deswegen durch einen Instrumentenverstärker verstärkt werden muss, bevor Sie mit Hilfe eines ADC gewandelt werden kann. Als Instrumentenverstärker wird im Projekt der Typ AD8553 gewählt. Die Strommessung kann z.B. verwendet werden, um die Einhaltung eines voreingestellten Grenzwertes zu überwachen. Für die Schaltung (Abbildung 17) wird als Basis die Beispielschaltung (Abbildung 16) des Datenblattes (Seite 14) gewählt:

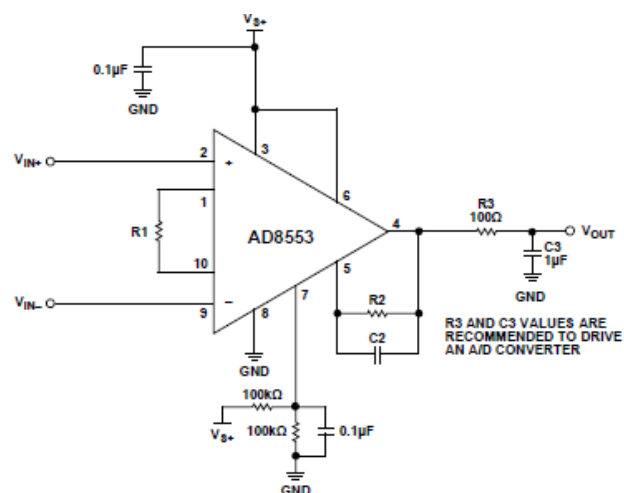


Abbildung 16: Schaltbild des AD8553-Bausteins aus dem Datenblatt [12]

Bis auf den Widerstand R1 und den RC-Filter werden die im Datenblatt vorgeschlagenen Werte übernommen. Zudem kommt noch die Dimensionierung der

Shunt Widerstände. Dieser befindet sich zwischen den Ports Ua und Ub. Der Ausgang Uout wird zum ADC geleitet. Laut Angaben des Datenblatts muss der Widerstand R1 (R1 in Abbildung 17) mindestens den Wert 3,92kΩ erhalten. Diese Empfehlung wird berücksichtigt. Als Shunt Widerstand für die hohen Ströme bis 1 A wird gewählt :

$$R_{Hshunt} = 10m\Omega \quad (4.39)$$

Dies ergibt einen maximalen Spannungsabfall über dem Widerstand von:

$$U_{shunt} = 10m\Omega * 1A = 10mV \quad (4.40)$$

Um den kompletten dynamischen Bereich des ADCs von 5V auszunutzen wird der folgende Verstärkungsfaktor benötigt:

$$A = \frac{U_{max}}{U_{shunt}} = \frac{5V}{10mV} = 500 \quad (4.41)$$

$$A = 2 * \frac{R_2}{R_1} \quad (4.42)$$

Wird die Formel nach R2 (Ric1 in Abbildung 17), aufgelöst ergibt sich bei Verwendung eines Widerstands R1=3,92kΩ ein Wert von :

$$R_2 = A * \frac{R_1}{2} = 250 * 3,92k\Omega \approx 980k\Omega \quad (4.43)$$

Für die Messung der kleinen Ströme bis 20mA werden die anderen vier Shunt Widerstände wie folgt gewählt :

$$R_{Lshunt} = \frac{U}{I} = \frac{10mV}{20mA} = 0,5 \Omega \quad (4.44)$$

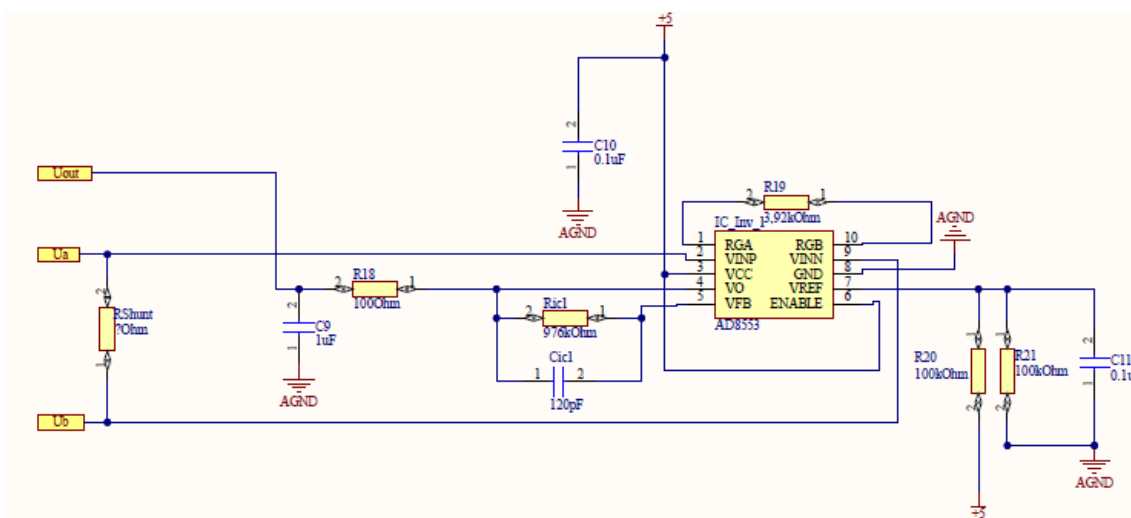


Abbildung 17: Instrumentenverstärker Schaltung

4.6 Pegelumsetzer zur Angleichung unterschiedlicher Logik-Pegel

Pegelumsetzer werden verwendet, um die unterschiedlichen Signalpegel verschiedener Logikfamilien einander anzupassen [13]. Im Rahmen der Arbeit wird auf zwei ICs vom Typ 74AVC zurückgegriffen (siehe Abb.18 und 19).

Es ist jeweils ein 16-Bit Pegelumsetzer für die Inputs der ADCs und des AMC und ein 4-Bit Baustein für die Outputs vorgesehen. VCC(a) und VCC(b) sind die Versorgungsspannungen zwischen denen eine Pegelanpassung erfolgt. In diesem Projekt wird an VCC(a), die 3,3V Logikspannung und VCC(b) die Logikspannung Vadj, mit welcher die Informationen von außen ankommen, angeschlossen.

Da der IC in beiden Richtungen umsetzt, muss die Richtung über eine entsprechende Belegung der DIR Ports und OE Ports eingestellt werden. An den „A“ und „B“ Ports liegen jeweils die Inputs/Outputs an, je nachdem wie die DIR-Ports und OE-Ports beschaltet sind. An der rechten „B“-Seite des 16-Bit Umsetzers liegen die von außen ankommenden digitalen Inputs der ADCs und des AMC mit dem falschen Signalpegel an und werden nach der Anpassung mit 3,3V-Pegel zur linken „A“-Seite durchgeschaltet. Für die Konfiguration dieser Schaltrichtung müssen nach Informationen des Datenblattes die DIR und OE Kanäle auf GND gezogen werden.

Anders verhält es sich bei dem 4-Bit Umsetzer. Dieser überträgt die Daten in die entgegengesetzte Richtung. Die auf der Platine generierten Signale mit 3.3V Pegel werden mit der A-Seite des Umsetzers verbunden und dann von der B-Seite mit angepassten Pegeln an den Stecker geführt.

An jedem nicht verwendeten Kanal wurde eine LED mit einem Vorwiderstand vorgesehen. Diese können später bei eventuellen Fehlersuchen behilflich sein.

Für die Berechnung des Vorwiderstandes wird ein Durchlass-Strom von $I_f=1\text{mA}$ gewählt, da keine hohe Leuchtkraft gewünscht ist. Die Durchlass-Spannung beträgt $U_d=1,9\text{V}$. Die Versorgungsspannung liegt bei 3,3V.

Die am Vorwiderstand abfallende Spannung ergibt sich zu:

$$U_{rv} = U_{ges} - U_d = 3,3V - 1,9V = 1,4V \quad (4.45)$$

Für den Vorwiderstand gilt dann :

$$R_v = \frac{U_{rv}}{I_f} = \frac{1,4V}{1mA} = 1,4k\Omega \quad (4.46)$$

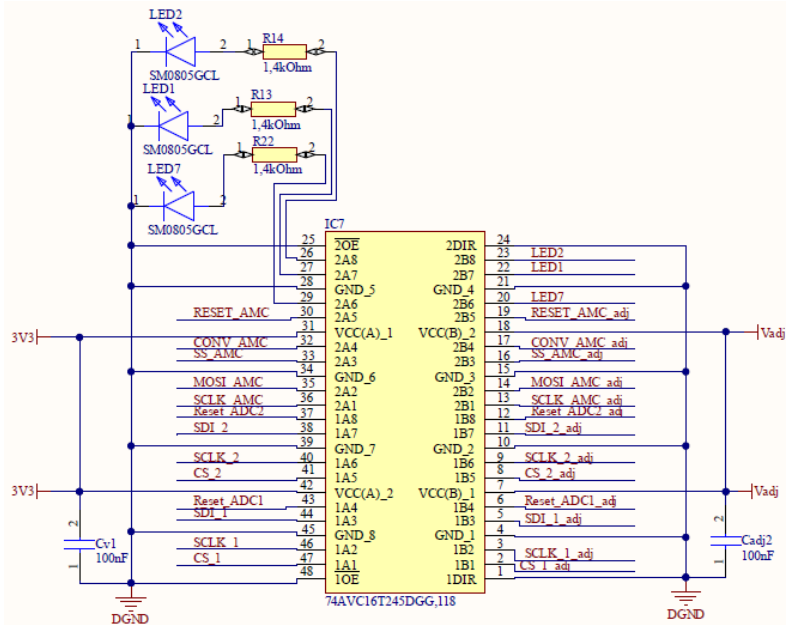


Abbildung 18: Beschaltung des Pegelumsetzers (16-Bit) für Digitale Inputs

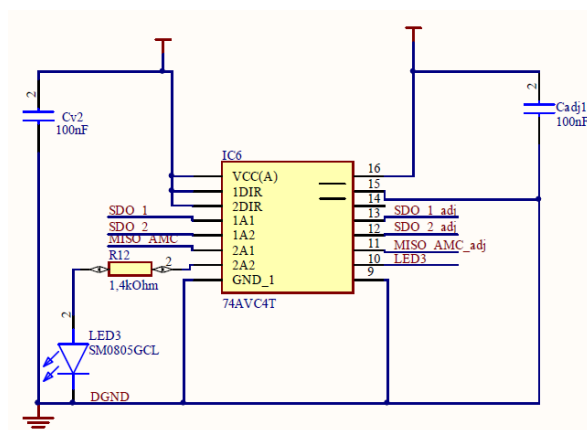


Abbildung 19: Beschaltung des Pegelumsetzers (4-Bit) für digitale Outputs

4.8 ADS8688-Baustein

In diesem Projekt werden zwei ADCs des Herstellers Texas Instruments vom Typ ADS8688 verwendet. Es sind 8 Kanäle mit einer Auflösung von 12 Bit integriert. Es ist ein 8-Kanal Multiplexer mit automatischem und manuellem Abtastmodus und eine integrierte 4,096V Referenzspannungsschaltung vorhanden [4].

Bei der Schaltplanerstellung (Abbildung 20) und Beschaltung des ADCs wurden bei der Dimensionierung der Bauteile die Hinweise im Datenblatt [4] beachtet. Insbesondere wurde bei den acht analogen Eingängen ein Tiefpassfilter eingesetzt, welcher hohe Frequenzen filtern und Aliasing minimieren soll.

Für die Berechnung des Kondensators und der Widerstände wird die gleiche Cut-Off-Frequenz wie beim Operationsverstärker, nämlich 350Hz angenommen.

Der verwendete RC-Filter hat die Besonderheit aus zwei gleichen Widerständen zu bestehen. Aus diesem Grund wird die allgemeine Formel für die Berechnung der Cut-Off-Frequency angepasst zu [14]:

$$f = \frac{1}{2\pi \cdot 2R \cdot C} \quad (4.47)$$

Die Auflösung nach C ergibt:

$$C = \frac{1}{2\pi \cdot 2R \cdot f} \quad (4.48)$$

Bei der Dimensionierung der Widerstände wurde eine von Texas Instruments veröffentlichten Beschaltung des ADCs berücksichtigt und 1kΩ gewählt [15].

Dadurch ergibt sich folgender Wert für den Kondensator des Filters:

$$C = \frac{1}{2\pi \cdot 1k\Omega \cdot 350\text{Hz}} = 0,455\mu\text{F} \quad (4.49)$$

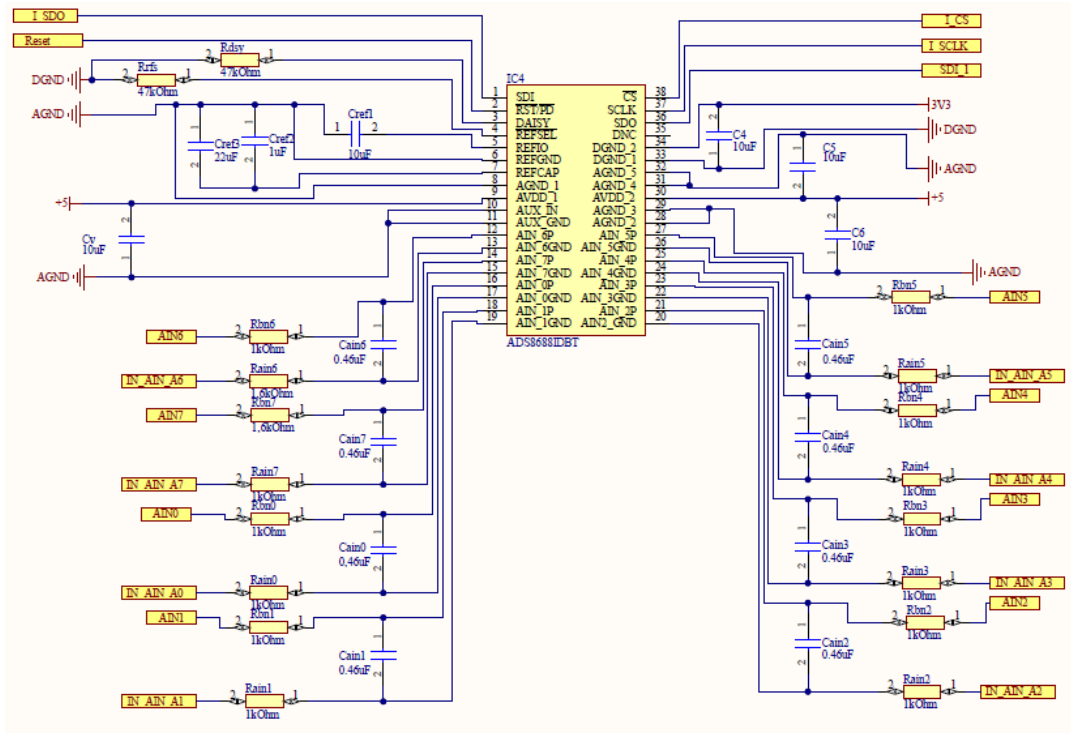


Abbildung 20: Beschriftung des ADS8688 ADC -Bausteins

Die Pins AIN0-AIN7 bezeichnen die analogen Kanäleingänge und IN_AIN0-IN_AIN7 die dazugehörigen Masseanschlüsse. Mit Hilfe der digitalen Inputs Reset, CS, SCLK, SDI kann der ADC nach Anpassung der Signalpegel über den Pegelumsetzer angesteuert werden. Der SDO Port steht für den Daten-Output und wird ebenfalls über weitere Pegelumsetzer zum vorgesehenen Stecker geleitet.

4.9 AMC7823 ADC/DAC Baustein

Der AMC7823 ist eine Überwachungs- und Steuerschaltung mit jeweils acht 12-Bit ADC- und DAC Kanälen. Darüber hinaus verfügt er über vier „out of range“ Alarmer für die Überwachung analoger Signale und sechs GPIOs zur Steuerung externer Geräte. Wahlweise kann eine interne programmierbare Referenzspannung von +2,5V und +1,25V, oder eine externe Referenzspannung genutzt werden. Der Bereich der analogen Eingangsspannung des ADC liegt zwischen 0 und 5V und der Bereich der analogen Ausgangsspannung des DAC je nach gewählter Referenzspannung zwischen 0 bis +2,5V bzw. +5V. Die Steuerung erfolgt über ein SPI Interface [3].

Der AMC wird in diesem Projekt eingesetzt, um acht unterschiedliche Referenzspannungen für die Generierung von Versorgungsspannungen bereitzustellen. Jeweils vier dieser Referenzspannungen werden an die LDOs und die anderen vier Referenzspannungen an die Operationsverstärker geführt. Die Spannungen werden über Sense Leitungen überwacht. Die Sense Leitungen werden dem AMC über die ADC Kanäle zugeführt. Bei der Dimensionierung der Beschaltung wurden entsprechend der Guidelines von Texas Instruments Kapazitäten von 150pF an die Kanal-Eingänge eingesetzt. Die übrigen Bauteile wurden mit Erfahrungswerten dimensioniert [16]. Die Schaltung ist in Abbildung 21 zu sehen. Die Ports AMC_AIN0 bis AMC_AIN7 stehen für die analogen Eingänge. Die DAC_0_OUT bis DAC_7_OUT Ports stehen für die analogen Ausgänge der DAC Kanäle. Die Signale SCLK, SS, MOSI und MISO Ports, sind digitale Eingänge der seriellen Schnittstelle. Der Signal (DAC_CONV in Abb. 21) ist ein digitaler logischer Eingang zur Ansteuerung des AMC. Sowohl die digitalen Eingänge als auch Ausgänge werden über Pegelumsetzer zum AMC geführt bzw. vom AMC weitergeleitet. Der Precision Output entspricht einer Präzisionsstromquelle zur Ansteuerung eines Thermistors und der Reset Port löst einen Reset des AMC aus. Der GPIO4 Pin wird zur Ansteuerung der 4 LDOs und 4 OPAs verwendet. Mit dem AMC7823 wurde das letzte Bauteil der Platine behandelt. Nun soll die Vorstellung der Software und Inbetriebnahme folgen.

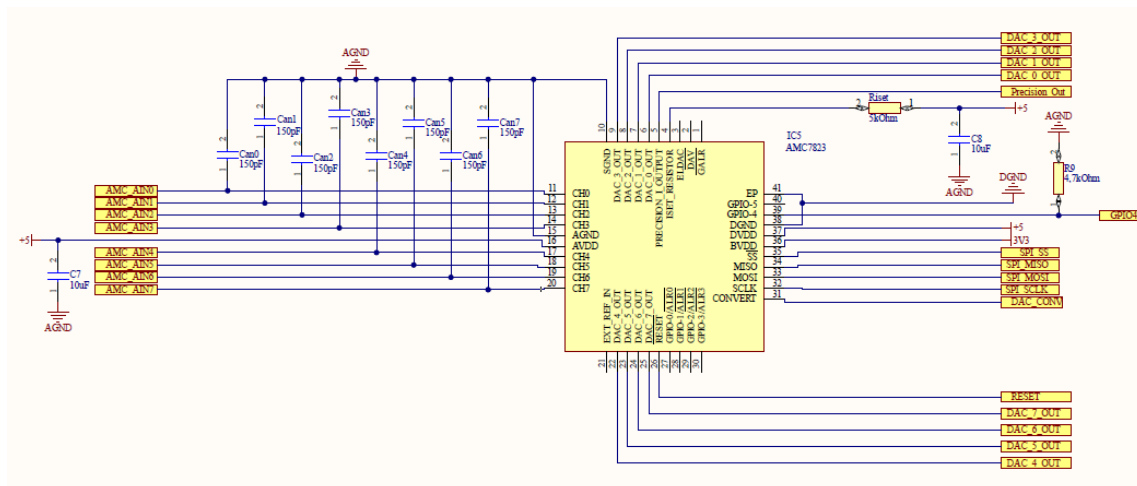


Abbildung 21: Beschaltung des AMC7823 ADC/DAC-Bausteins

5. Registeroperationen und Befehlsformate

5.1 Steuerung des ADS8688 ADC Bausteins

5.1.1 Programm-Register Lese/Schreibe Operation

Der ADS8688-Baustein verfügt über ein 16-Bit Lese/Schreib-Register. Für jegliche Lese- und Schreibvorgänge werden mindestens 24 SCLK Zyklen nach einer fallenden CS Flanke benötigt. Wenn CS auf den Low-Pegel übergeht, geht die SDO Leitung ebenfalls auf Low. Der ADC erhält einen Befehl entsprechend des Command Formats aus der Tabelle 4 über die SDI Leitung, wo die ersten sieben Bits (15-9) die Registeradresse repräsentieren. Das achte Bit (8) ist die Lese- oder Schreibweisung, dabei ist RD aktiv Low und WR aktiv High, wodurch eine 1 den Schreibvorgang einleitet.

Für den Schreibprozess, welcher in der Abbildung 22 veranschaulicht wird sind die nächsten acht Bits (7-0) an der SDI Leitung die gewünschten Daten für das adressierte Register. Zusätzlich gibt das Bauteil während der nächsten 8 SCLK-Zyklen die 8-Bits über die SDO-Leitung zurück, welche in das Register geschrieben wurden. Mit dieser Datenrückgabe kann überprüft werden, ob diese korrekt im Register abgelegt wurden.

Pin	Register Address	W/R (active low)	Data
SDI	ADDR[6:0]	1	DIN[7:0]

Tabelle 4: Command Format / Befehlsformat

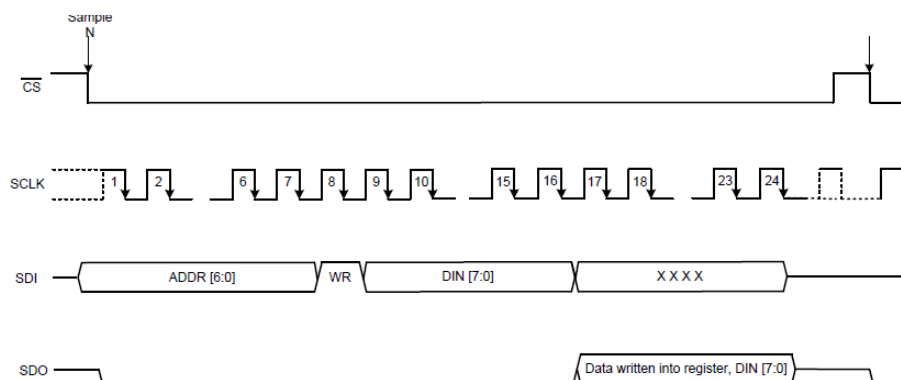


Abbildung 22: Schreibprozess ins Programmregister des ADS8688[4]

Beim Lesezyklus, welcher in der Abbildung 23 veranschaulicht wird und dessen Command Format in Tabelle 5 zu finden ist, ist nun die Adresse der Daten welche gelesen werden sollen zu verwenden. Anschließend das Bit um den Lesevorgang zu definieren. Die nächsten acht Bits (7-0) an der SDI Leitung sind „don't care“ Bits und SDO bleibt Low. Ab der 16. fallenden Flanke von SCLK wird das zuvor adressierte Register während der nächsten 8 SCLK Zyklen ausgegeben. Dabei wird das MSB (Most Significant Bit) zu erst übertragen.

PIN	REGISTER ADDRESS(Bits 15-9)	WR/RD(active low)(Bit 8)	DATA(Bits 7-0)
SDI	ADDR[6:0]	0	XXXX
SDO	0000 000	0	DOUT[7:0]

Tabelle 5: Schreibenweisung Befehlsformat

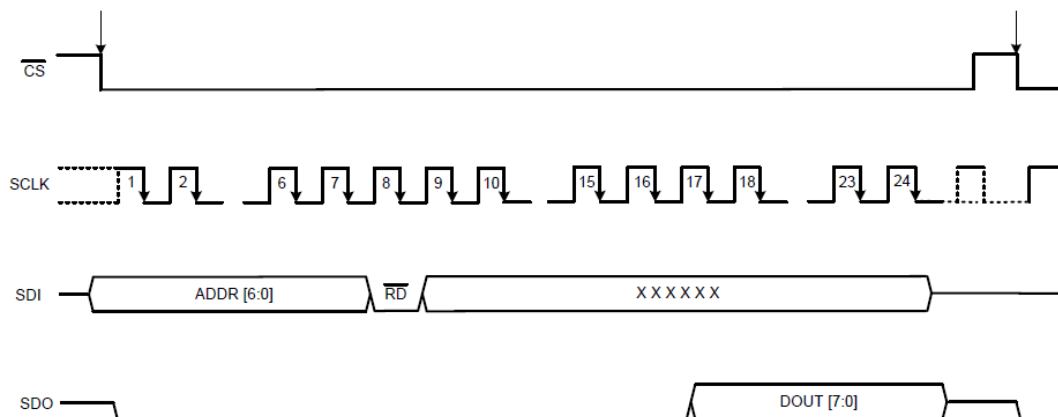


Abbildung 23: Programm Register Leseprozess [4]

5.1.2 Einstellung des Messbereiches beim ADS8688 ADC Baustein

Als erster Konfigurationsschritt muss der Messbereich der ADC-Kanäle übermittelt werden. Die unterschiedlichen Messbereiche sind in Tabelle 6 dargestellt. Die Spalte „Input Range“ enthält die Vorfaktoren, mit welchen die Referenzspannung von 4.096V multipliziert wird. Dies ergibt die ebenfalls in der Tabelle enthaltenen positiven (Positive Full Scale) und negativen (Negative Full Scale) Wertebereiche und den Betrag des gesamten Wertebereiches (Full-Scale Range), in dem der ADC betrieben werden kann. Schließlich ist noch in der letzten Spalte der Tabelle 6 das LSB (Least Significant Bit) in μV dargestellt. Das ist der kleinste Wert, welcher vom ADC bei der Quantisierung des Signals dargestellt werden kann.

INPUT RANGE	POSITIVE FULL SCALE	NEGATIVE FULL SCALE	FULL-SCALE RANGE	LSB (μV)
$\pm 2.5 \times V_{\text{REF}}$	10.24 V	-10.24 V	20.48 V	312.50
$\pm 1.25 \times V_{\text{REF}}$	5.12 V	-5.12 V	10.24 V	156.25
$\pm 0.625 \times V_{\text{REF}}$	2.56 V	-2.56 V	5.12 V	78.125
0 to $2.5 \times V_{\text{REF}}$	10.24 V	0 V	10.24 V	156.25
0 to $1.25 \times V_{\text{REF}}$	5.12 V	0 V	5.12 V	78.125

Tabelle 6: ADS8688 Messbereich und Quantisierungsfehler[4]

Es ist möglich allen Kanälen den gleichen Messbereich, oder jedem Kanal einen unterschiedlichen Bereich zuzuweisen. Ist beispielsweise bereits im Vorfeld bekannt, dass mit Kanal 1 nur Spannungen bis 2V gemessen werden, ist es nicht ratsam einen hohen Messbereich zu wählen, da die Genauigkeit der Messung durch einen höheren LSB beeinträchtigt wird (siehe Tabelle 6, LSB). Der Befehl zur Konfiguration des Messbereiches setzt sich entsprechend Tabelle 5 wie folgt zusammen. Für die Adresse sind 7 Bits vorgesehen. Die Adresse von Kanal 1 lautet 05h, binär 0000101. Das achte Bit legt die write/read Funktion fest. Da es sich in diesem Beispiel um eine Write-Anweisung handelt ist das achte Bit eine „1“. Damit ist die erste Hälfte der Anweisung (Tabelle 7) fertig.

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
0	0	0	0	1	0	1	1

Tabelle 7: Bitfolge mit Adresse von Kanal 1 des ADS8688 Bausteins

Als Messbereich soll $\pm 0.625 \cdot V_{ref}$ eingestellt werden. Die Bitabfolge dafür lautet 0010. Diese und weitere Bitfolgen der übrigen Verstärkungen sind Tabelle 9 zu entnehmen. Auch diese Information muss mit Vier Nullen erweitert werden, sodass sich die Information laut Tabelle 8 ergibt und auch der zweite Teil der Anweisung mit 8- Bit vollständig ist.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	0	0	0	1	0

Tabelle 8: Zweiter Teil der 16-Bitabfolge zur Einstellung des Messbereiches beim ADS8688 Bausteins

Bitfolge	Meßbereich
0000	$\pm 2.5 \cdot V_{ref}$
0001	$\pm 1.25 \cdot V_{ref}$
0010	$\pm 0.625 \cdot V_{ref}$
0101	0 to $2.5 \cdot V_{ref}$
0110	0 to $1.25 \cdot V_{ref}$

Tabelle 9: Meßbereiche des ADS8688 Bausteins

Der vollständige 16-Bit Befehl für die Übermittlung des genannten Messbereiches an Kanal 1 ist in Tabelle 10 in binärer Schreibweise dargestellt und kann in dieser Form übermittelt werden. Der Hexadezimalwert lautet 0XB02.

Bit 15 MSB	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	0	1	0	1	1	0	0	0	0	0	0	1	0

Tabelle 10: 16-Bit Befehl für Übermittlung des Meßbereiches $\pm 2.5 \cdot V_{ref}$ an Kanal 1 des ADS8688 Bausteins

Nach Auslesung der Daten von Kanal 1 müssen die von der SDO-Leitung empfangenen Rohdaten in logische Spannungswerte umgerechnet werden. Dabei wird entsprechend Formel 5.1 der ermittelte Wert des ADC mit der maximalen Spannung des Messbereiches (z.B $2 \cdot 2.5 \cdot V_{ref}$) multipliziert und das Ergebnis mit der Auflösung dividiert, welche sich aus 16^2 zu 65535, oder in Hexadezimal zu 0xFFFF berechnet. Da der Messbereich sich sowohl ins positive als auch ins negative gleichermaßen erstreckt, ist ein Faktor „2“ in die Gleichung einzubringen. Ein realistisches Beispiel für die Auslesung des Kanals 1 wäre der Wert 0x18FF. Mit dem erwähnten Messbereich und der genannten Auflösung ergibt sich die Spannung am Kanal 1 gemäß Gleichung 5.2 zu $U_{k1} = 2V$. Nach dem gleichen Prinzip werden die

Spannungen und Ströme beim 12-Bit AMC7823 gemessen, mit dem Unterschied, dass die Auflösung (0Xfff) geringer ist. Unter Umständen kann es zu Messabweichungen kommen, welche im weiteren Verlauf der Thesis behandelt werden.

$$\text{Spannung} = \text{WERT}_{adc} * \frac{\text{Maximale Spannung}}{\text{Auflösung}} \quad 5.1$$

$$U_{k1} = 0x18FF * \frac{2*2,5*4,096}{0xFFFF} = 2V \quad 5.2$$

Beim ADC 2 mit dessen Hilfe die Ströme auf Basis eines Shunt-Widerstandes gemessen werden, werden alle Kanäle auf einen Messbereich von $1,25V * V_{ref} \approx 5V$ eingestellt. Die vom ADC erhaltenen Werte werden ebenfalls wie in Formel 5.1 nach der Formel 5.3 berechnet, um zunächst einen Spannungswert zu erhalten. Erst anschließend kann eine Umrechnung nach Formel 5.4 in einen Strom erfolgen. Da die Frontend-Platine keine negative Spannungsversorgung besitzt, sind zusätzliche Maßnahmen nötig um negative Ströme messen zu können. Der Instrumentenverstärker welcher mit 5V versorgt wird, ist so beschaltet, dass sich bei einer Differenzspannung am Eingang von 0V eine Ausgangsspannung von 2.5V einstellt. Der Ausgangsspannungsbereich des Instrumentenverstärkers von 2,5V bis 5V wird dadurch für positive Spannungen und Ströme verwendet und der Spannungsbereich von 2.5V bis 0V für negative Spannungen und Ströme. Die 2.5V am Ausgang der Instrumentenverstärker bedeuten demnach 0mA und 5V bedeuten 20mA. Dies kann durch Einsetzen in die Formel 5.4 leicht nachgeprüft werden. Eine Beispielrechnung soll für einen Spannungswert von 3V an Kanal 1 erfolgen. Die ersten 4 Instrumentenverstärker an den ersten vier Kanälen sind für Ströme bis 20mA ausgelegt.

$$I = \frac{I_{max}}{(\text{Messbereich} - 2,5V)} * (\text{WERT}_{adc} - 2,5V) \quad 5.3$$

$$I = \frac{0,02A}{(5V - 2,5V)} * (3V - 2,5V) = 4mA \quad 5.4$$

5.1.3 Registersatz des ADS8688 ADC Bausteins

Die internen Register des ADC sind in zwei Kategorien gegliedert. Befehlsregister (Command Registers) und Programmregister (Programm Registers). Die Befehlsregister werden genutzt, um einen automatischen (Auto_RST) oder manuellen (MAN_CH) Sequenzierungsmodus zu wählen und das Bauteil in den Standby Modus, oder den Power-Down Modus (PWR_DN) zu versetzen. Darüber hinaus kann das Befehlsregister verwendet werden, um durch Aktivierung des (RST) die RegisterEinstellungen zurückzusetzen.

Die Programmregister werden benutzt, um die Reihenfolge der Kanäle im AUTO_RST Modus einzustellen und den Eingangsbereich für bestimmte Kanäle einzustellen.

5.1.3.1 Beschreibung des Command Registers des ADS8688 ADC Bausteins

Die Befehle für die erwähnten Einstellungen des Befehlsregisters sind in der Registermap (Tabelle 11) aufgeführt. Nach dem Einschaltvorgang des ADC-Bausteins wird auf einen Befehl gewartet, um in einen gültigen Operationsmodus zu gelangen.

REGISTER	MSB BYTE								LSB BYTE	COMMAND (Hex)	OPERATION IN NEXT FRAME
	B15	B14	B13	B12	B11	B10	B9	B8	B[7:0]		
Continued Operation (NO_OP)	0	0	0	0	0	0	0	0	0000 0000	0000h	Continue operation in previous mode
Standby (STDBY)	1	0	0	0	0	0	1	0	0000 0000	8200h	Device is placed into standby mode
Power Down (PWR_DN)	1	0	0	0	0	0	1	1	0000 0000	8300h	Device is powered down
Reset program registers (RST)	1	0	0	0	0	1	0	1	0000 0000	8500h	Program register is reset to default
Auto Ch. Sequence with Reset (AUTO_RST)	1	0	1	0	0	0	0	0	0000 0000	A000h	Auto mode enabled following a reset
Manual Ch 0 Selection (MAN_Ch_0)	1	1	0	0	0	0	0	0	0000 0000	C000h	Channel 0 input is selected
Manual Ch 1 Selection (MAN_Ch_1)	1	1	0	0	0	1	0	0	0000 0000	C400h	Channel 1 input is selected
Manual Ch 2 Selection (MAN_Ch_2)	1	1	0	0	1	0	0	0	0000 0000	C800h	Channel 2 input is selected
Manual Ch 3 Selection (MAN_Ch_3)	1	1	0	0	1	1	0	0	0000 0000	CC00h	Channel 3 input is selected
Manual Ch 4 Selection (MAN_Ch_4) ⁽¹⁾	1	1	0	1	0	0	0	0	0000 0000	D000h	Channel 4 input is selected
Manual Ch 5 Selection (MAN_Ch_5)	1	1	0	1	0	1	0	0	0000 0000	D400h	Channel 5 input is selected
Manual Ch 6 Selection (MAN_Ch_6)	1	1	0	1	1	0	0	0	0000 0000	D800h	Channel 6 input is selected
Manual Ch 7 Selection (MAN_Ch_7)	1	1	0	1	1	1	0	0	0000 0000	DC00h	Channel 7 input is selected
Manual AUX Selection (MAN_AUX)	1	1	1	0	0	0	0	0	0000 0000	E000h	AUX channel input is selected

Tabelle 11: Comand Register Map des ADS8688 ADC Bausteins [4]

5.1.3.2 Beschreibung des Program Registers des ADS8688 ADC Bausteins

Die Einstellungen in diesem Register werden genutzt, um die Kanalabfolge im AUTO_RST Modus einzustellen, die Bauteil-ID im Daisy Chain Mode zu konfigurieren, das SDO Ausgangsformat zu wählen und den Eingangsbereich für bestimmte Kanäle zu definieren. Alle möglichen Einstellungen des Programm-Registers sind in Tabelle 12 gelistet. Nach einem Power-Up, oder Reset werden die unterschiedlichen Programmregister des Bauteils mit Standartwerten initialisiert und das Bauteil wartet auf Befehle, um in einen Operationsmodus zu gelangen.

REGISTER	REGISTER ADDRESS BITS[15:9]	DEFAULT VALUE ⁽¹⁾	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
AUTO SCAN SEQUENCING CONTROL										
AUTO_SEQ_EN	01h	FFh	CH7_EN ⁽²⁾	CH6_EN	CH5_EN	CH4_EN	CH3_EN	CH2_EN	CH1_EN	CH0_EN
Channel Power Down	02h	00h	CH7_PD	CH6_PD	CH5_PD	CH4_PD	CH3_PD	CH2_PD	CH1_PD	CH0_PD
DEVICE FEATURES SELECTION CONTROL										
Feature Select	03h	00h	DEV[1:0]			0	0	0	SDO [2:0]	
RANGE SELECT REGISTERS										
Channel 0 Input Range	05h	00h	0	0	0	0	Range Select Channel 0 [3:0]			
Channel 1 Input Range	06h	00h	0	0	0	0	Range Select Channel 1 [3:0]			
Channel 2 Input Range	07h	00h	0	0	0	0	Range Select Channel 2 [3:0]			
Channel 3 Input Range	08h	00h	0	0	0	0	Range Select Channel 3 [3:0]			
Channel 4 Input Range	09h	00h	0	0	0	0	Range Select Channel 4 [3:0]			
Channel 5 Input Range	0Ah	00h	0	0	0	0	Range Select Channel 5 [3:0]			
Channel 6 Input Range	0Bh	00h	0	0	0	0	Range Select Channel 6 [3:0]			
Channel 7 Input Range	0Ch	00h	0	0	0	0	Range Select Channel 7 [3:0]			
COMMAND READ BACK (Read-Only)										
Command Read Back	3Fh	00h	COMMAND_WORD [7:0]							

Tabelle 12: 12 Programm Register Map des ADS8688 ADC Bausteins [4]

Für die Nutzung mehrerer ICs, welche parallel an einer SPI-Schnittstelle kommunizieren sollen, existieren sogenannten Chain- und Star-Topologien, allerdings werden diese im Falle der Front-End Platine nicht benötigt, da jeder Slave eine eigene SPI-Schnittstelle besitzt. Wenn jedoch die Daten der Slaves nacheinander abgefragt werden, so wäre auch eine Implementierung mit einer einzigen SPI-Schnittstelle ausreichend. Der ADC besitzt unterschiedliche Zustände (Modes), welche bei entsprechender Programmierung eingenommen werden können. Nach dem Einschaltvorgang des ADC geht dieser in den „IDLE Mode“ über, in welchem er keine Aktionen durchführt und auf Anweisungen des Nutzers wartet. Dabei wird das Register mit Standartwerten initialisiert und erfordert entsprechende Konfigurationseinstellungen bevor der Wandlungsprozess starten kann.

5.1.4 Betriebsmodi des ADS8688 ADC Bausteins

In Abbildung 24 sind die unterschiedlichen Anweisungen zum Betreten der unterschiedlichen Modi des ADC dargestellt. Die entsprechenden Befehle sind der Command Register Map (Tabelle 11) zu entnehmen.

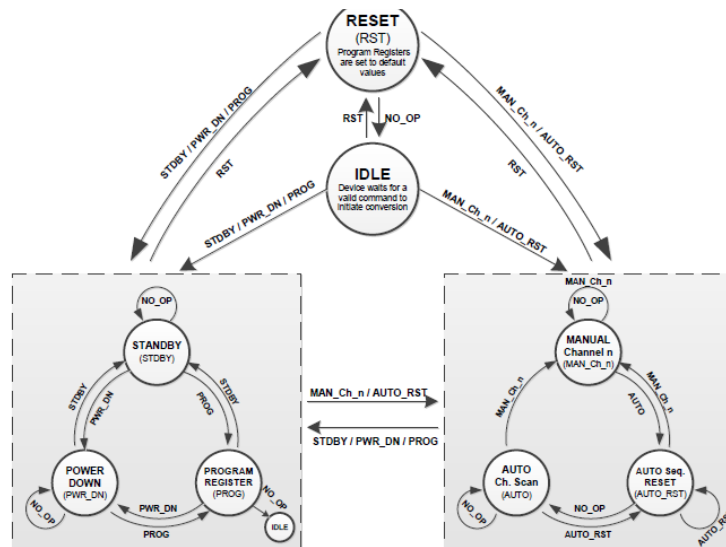


Abbildung 24: Zustandsdiagramm des ADS8688 Bausteins [4]

Wenn die SDI-Leitung für 16 Clock-Zyklen auf Low-Pegel gehalten wird (alle 16 Bits auf Low), dann startet das Bauteil im zuletzt operierenden Modus, also mit den in den Programmregistern bereits konfigurierten Einstellungen. Während die 16 Bits auf der SDI Line gelesen werden, muss CS auf Low bleiben, ansonsten geht der ADC in den sogenannten INVALID Zustand, sodass die gelesenen DATEN ungültig werden.

Es wird ein Low-Power-Standby Mode unterstützt, wodurch nur einige Teile des ADC in den Standby-Modus versetzt und das Device durch Beendigung des Standby-Modus schnell (in 20us) gestartet werden kann. Dabei werden die Programmregister nicht zurückgesetzt. Zudem verfügt der ADC über einen softwareseitigen als auch hardwareseitigen Power-Down Mode (PWR_DN), bei welchem der komplette ADC inkl. der internen Referenzspannung und dem Buffer ausgeschaltet wird. Nachdem der Mode beendet wird, werden mindestens 15ns benötigt, bis der ADC eingeschaltet ist und den ausgewählten analogen Eingangskanal im Internal Reference Mode

(REFSEL=0) konvertiert. Für den Hardware Power-Down Mode muss der RST-Pin auf Low geschaltet werden. Der Unterschied zum Software Power-Down Mode ist, dass die Voreinstellungen des Programmregisters nicht beibehalten werden.

Der ADC unterstützt sowohl einen Reset über Hardware als auch über Software, in welchem alle Programmregister auf ihre Standartwerte zurückgesetzt werden. Für den Hardware Reset ist der RST/PD (Input, active low) zu nutzen. Ein RST Befehl von 8500h führt den Software Reset entweder sofort, oder nach Beendigung des aktuellen Sample-Vorgangs an der steigenden Flanke von CS durch. Solange kein gültiger Befehl erkannt wird (AUTO_RST oder MAN_Ch_n), verbleibt der ADC im Reset Modus und SDI bleibt Low während der nachfolgenden Datenströme.

5.1.5 Kanalauswahl des ADS8688 ADC Bausteins

In der für die Frontend-Platine entwickelten Software werden die Kanäle manuell gelesen. Es ist jedoch möglich den ADC so zu programmieren, dass alle analogen Eingangssignale automatisch gelesen werden, indem eine gültige Auto-Channel Abfolge mit einem Reset- Befehl (AUTO_RST, A000h) im Befehlsregister ausgeführt wird. In diesem Modus iteriert der ADC durch die gewählten Kanäle in aufsteigender Abfolge, beginnend mit dem niedrigsten Kanal und führt eine Konversion durch. Dies wiederholt sich bei der nächsten Abfolge. Dabei kann der Spannungseingangsbereich in der Auto-Scan Abfolge konfiguriert werden, indem das „range select register“ eingestellt wird. Das Bauteil verbleibt im AUTO_RST Mode, solange bis kein neuer Befehl ausgeführt wird und SDI Low bleibt. Falls der AUTO_RST Befehl erneut ausgeführt werden würde, so würde die Abfolge zurückgesetzt werden und vom niedrigsten Kanal an neu beginnen.

Im Manual Channel Mode (MAN_Ch_n) werden die Kanäle einzeln abgetastet. Der entsprechende Befehl ist gemäß der Command Memory Map in Tabelle 43 ins Command Register abzusetzen. Damit das Eingangssignal am nächsten Kanal genau erfasst und umgewandelt wird, sind 32 SCLK Zyklen notwendig. In Abbildung 25 wird die Konfiguration des MANUAL Channel Modes veranschaulicht. Die seriellen Daten auf dem SDI Kanal entsprechen dem MAN_CH_n_Command (rotes Rechteck). Der grüne Pfeil zeigt wie das CS-Signal nach dem Man_Ch Befehl den aktuellen Lesevorgang (grünes Rechteck) abwartet, bevor es auf High geht. Der rote Pfeil zeigt, wie das CS-Signal optional, direkt nach Ausführung des Man-Ch Befehls auf High geht und der Befehl auf der steigenden Flanke ausgeführt wird. An Hand des SCLK-Signals ist ebenfalls die optionale Unterbrechung des aktuellen Lesevorgangs zu sehen, falls CS sofort nach Ausführung des MAN_ch_n Mode Befehls auf High geht. Ebenfalls sind die geforderten 32 SCLK Zyklen dargestellt und die anschließende Lesung eines weiteren Samples (gelbes Rechteck).

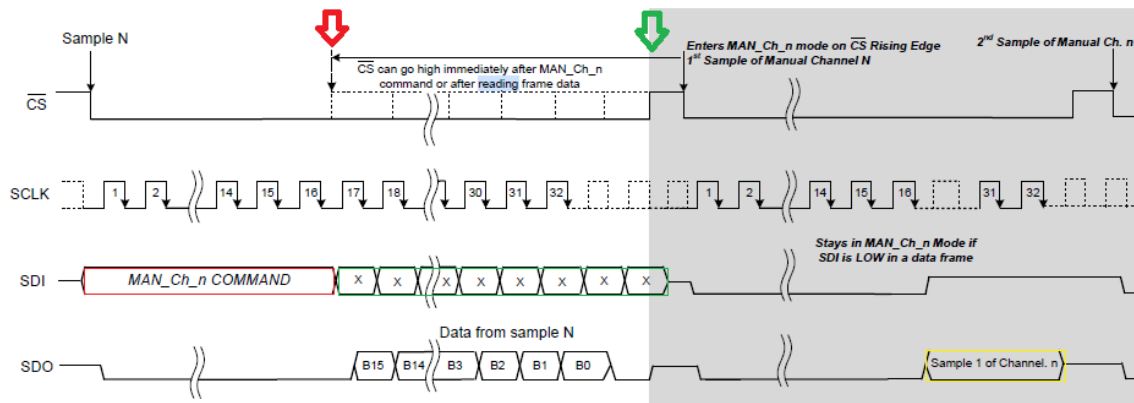


Abbildung 25: Ausführung des Manual Channel Modes des ADS8688 ADC-Bausteins [4]

Abbildung 26 ist zu entnehmen, dass solange der gleiche Kanal abgetastet (hier CH1) wird, bis der Befehl erfolgt, einen anderen Kanal abzutasten. Hierfür wird das CS Signal auf Low-Zustand gehalten, und ständig 16-Bit Nullen gesendet. Erfolgt nun der Befehl einen anderen Kanal (hier Kanal 3) abzutasten, so wird dies am Ende des aktuellen Lese/Schreib-Vorgangs mit der fallenden Flanke von CS durchgeführt.

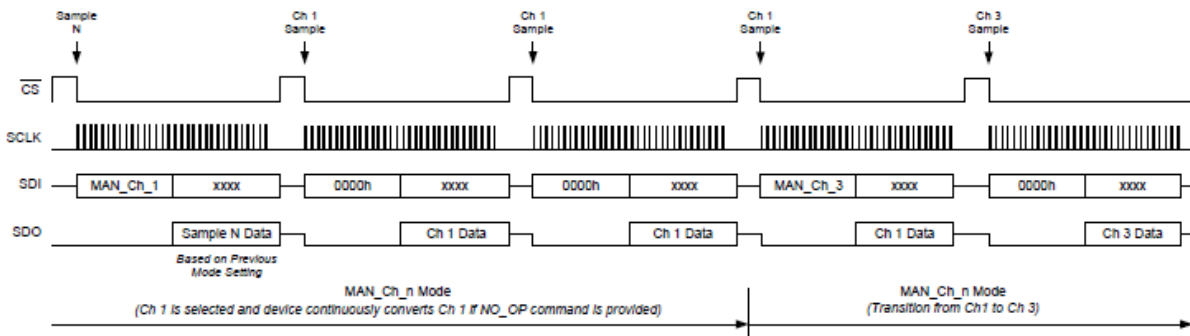


Abbildung 26: Verlauf Lese/Schreib-Vorgang im Manual Channel Mode [4]

5.2 Kommunikation und Steuerung des AMC7823 ADC/DAC-Bausteins

5.2.1 Command Format des AMC7823 ADC/DAC-Bausteins

Das Command Format, in welchem mit dem AMC7823 kommuniziert wird, ist in Tabelle 12 dargestellt und 16-Bit lang. Das MSB Bit 15 entspricht dem Read/Write (active low) Bit. Demnach steht eine „1“ an dieser Stelle für eine Read-Operation und eine „0“ für eine Write-Operation. Das Register mit den Anweisungen ist die sogenannte Memory Map (Abbildung 27), die in zwei Seiten organisiert ist, welche durch Bit 13 und 12 bestimmt werden. Seite "0" entspricht dabei dem Data/Status Register „0“ und Seite "1" dem Control/Setting Register. Die Bits 6 bis 10 stehen für die Startadresse des ersten zu lesenden Kanals und die Bits 0 bis 4 für die Endadresse des zuletzt zu lesenden Kanals. Die anderen mit „X“ gekennzeichneten Bits sind don't care Bits und können beispielsweise mit 0 vorbelegt werden. Mit diesem 16-Bit Format wird jedes der im weiteren Verlauf der Arbeit beschriebenen Register angesprochen, worauf weitere 16-Bit Daten mit Befehlen folgen.

Bit 15 MSB	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
R/W(active low)	X	PG1	PG0	X	SADR4	SADR3	SADR2	SADR1	SADR0	X	EADR4	EADR3	EADR2	EADR1	EADR0

Tabelle 13: Command Format des AMC7823 ADC/DAC-Bausteins

Page 0: Data/Status Registers				Page 1: Control/Setting Registers			
Address	R/W	Default	Register Name	Address	R/W	Default	Register Name
00	R	0x0000	ADC-0 Data	00	R/W	0x0000	DAC-0 Data
01	R	0x0000	ADC-1 Data	01	R/W	0x0000	DAC-1 Data
02	R	0x0000	ADC-2 Data	02	R/W	0x0000	DAC-2 Data
03	R	0x0000	ADC-3 Data	03	R/W	0x0000	DAC-3 Data
04	R	0x0000	ADC-4 Data	04	R/W	0x0000	DAC-4 Data
05	R	0x0000	ADC-5 Data	05	R/W	0x0000	DAC-5 Data
06	R	0x0000	ADC-6 Data	06	R/W	0x0000	DAC-6 Data
07	R	0x0000	ADC-7 Data	07	R/W	0x0000	DAC-7 Data
08	R	0x0000	ADC-8 Data	08	R/W	0x0000	LOAD DAC
09	R	0x0000	ALR Register	09	R/W	0x0000	DAC Configuration
0A	R/W	0xFFFF	GPIO Register	0A	R/W	0x4000	AMC Status/Configuration
0B			Reserved	0B	R/W	0x0000	ADC Control
0C			Reserved	0C	R/W	0x0000	RESET
0D			Reserved	0D	R/W	0x0000	Power-Down
0E			Reserved	0E	R/W	0x0FFF	Threshold-Hi-0
0F			Reserved	0F	R/W	0x0000	Threshold-Low-0
10			Reserved	10	R/W	0x0FFF	Threshold-Hi-1
11			Reserved	11	R/W	0x0000	Threshold-Low-1
12			Reserved	12	R/W	0x0FFF	Threshold-Hi-2
13			Reserved	13	R/W	0x0000	Threshold-Low-2
14			Reserved	14	R/W	0x0FFF	Threshold-Hi-3
15			Reserved	15	R/W	0x0000	Threshold-Low-3
16			Reserved	16			Reserved
17			Reserved	17			Reserved
18			Reserved	18			Reserved
19			Reserved	19			Reserved
1A			Reserved	1A			Reserved
1B			Reserved	1B			Reserved
1C			Reserved	1C			Reserved
1D			Reserved	1D			Reserved
1E			Reserved	1E	R	0xE000	Part Revision Number
1F			Reserved	1F			Reserved

Abbildung 27: Memory Map des AMC7823 ADC/DAC-Bausteins [3]

5.2.2 Lese- und Schreibvorgang im AMC 7823 ADC/DAC Baustein

Der Schreibvorgang in AMC 7823 Register ist im oberen Teil von Abbildung 28 zu sehen. Nachdem das Chip Select bzw. Slave Select Signal nach Masse gezogen wurde, wird die Schreibanweisung, welche die Adresse des anzusprechenden Registers enthält, mit der nächsten steigenden Taktflanke über 16 Takte auf der MOSI-Leitung übermittelt. Anschließend folgen alle Daten, die in das angesprochene Register zu schreiben sind. Nach der fallenden Flanke des 16ten Bits der letzten Daten endet die Übertragung mit dem High-Pegel am Chip Select Signal. Es wird kein Rückgabewert erwartet, daher werden über die MISO-Leitung auch keine Daten übertragen.

Der Lesevorgang ist im unteren Teil von Abbildung 28 zu sehen und startet nach gleichem Prinzip. Über die MOSI-Leitung wird lediglich die Leseanweisung (Read Command) ans Register gesendet um z.B den ADC-Kanal 0 auszulesen. Während der 16 Takte in denen die 16-Bit Information gesendet wird, werden keine Daten empfangen. Erst nachdem die Information vollständig, also auch das letzte Bit gesendet ist, kann der AMC (Slave) den erwarteten Rückgabewert liefern, welcher im Beispielfall den Daten des Kanals 0 entsprechen würde. Nachdem die letzten Daten über die MISO-Leitung übertragen worden sind, wird das CS Signal auf High-Pegel gesetzt.

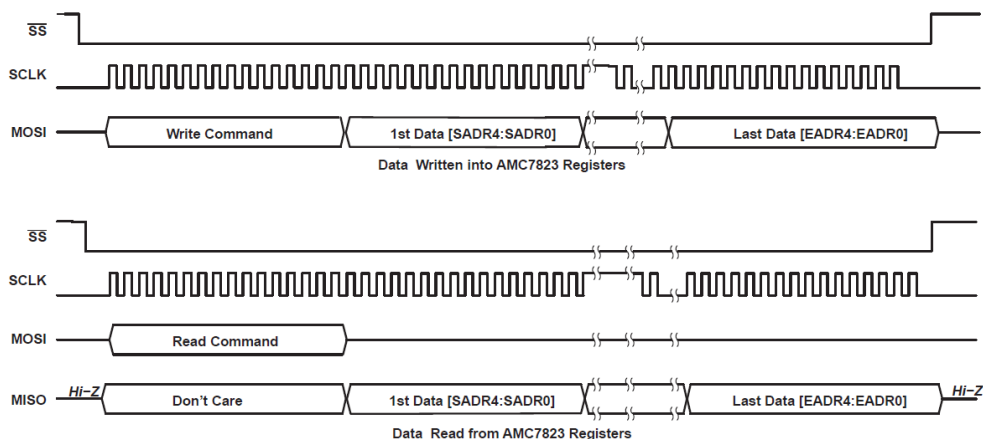


Abbildung 28: Schreib -und Leseprozess des AMC7823 ADC/DAC Bausteins [3]

5.2.2.1 Power-Down-Register des AMC7823 ADC/DAC-Bausteins

Der AMC7823 ist vor der eigentlichen Nutzung zu konfigurieren. Dazu ist eine Reihe von Einstellungen notwendig, welche dem Register jeweils in Form eines 16-Bit Befehls übermittelt werden. Zunächst ist das Power-Down-Register zu konfigurieren. Das Command Format dafür ist in Tabelle 14 dargestellt. Die Bits 4-0 stellen don't care Bits dar. Das Bit 5 (PREF) definiert den Status des Reference Buffers und wird auf High gesetzt, damit dieser aktiviert wird. Das Bit 6 (PTS) definiert den Betriebsmodus der Präzisionsstromquelle, welche im normalen Betriebsmodus aktiviert sein soll und deshalb auf High gesetzt wird. Ebenso werden die Bits 14-7 und Bit 15 auf High-Pegel gesetzt, damit die DACs und ADCs im normalen Betriebsmodus arbeiten. Der vollständige Befehl ist in Tabelle 15 dargestellt.

Bit 15 MSB	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PADC	PDAC7	PDAC6	PDAC5	PDAC4	PDAC3	PDAC2	PDAC1	PDAC0	PTS	PREFB	EADR4	EADR3	EADR2	EADR1	EADR0

Tabelle 14: Power Down Register Format des AMC7823 ADC/DAC Bausteins

Bit 15 MSB	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0

Tabelle 15: Power Down Register Befehl des AMC7823 ADC/DAC Bausteins

5.2.2.2 Status / Configuration Register des AMC7823 ADC/DAC Bausteins

Anschließend ist das AMC Status/Configuration Register entsprechend dem Command Format aus Tabelle 17 zu konfigurieren. Das Bit 5 (ECNVT) definiert, ob die Kanäle im Direkt-Mode oder im Auto-Conversion-Mode betrieben werden sollen. In diesem Projekt wird das ECNVT-Bit auf High gesetzt, sodass der Direkt-Mode eingestellt wird und die ADC-Kanäle manuell ausgelesen werden. Das Bit 6 (GREF) wird ebenfalls auf High gesetzt, um die Referenzspannung von 2,5V einzustellen und damit einen Messbereich von 0-5V zu erhalten, welcher an allen ADC-Kanälen gleichermaßen erwünscht ist. Alternativ ist eine Referenzspannung von 1,25V möglich, welche mit einem Low-Pegel am Bit 6 eingestellt werden würde. Durch das Bit 7 wird die interne Referenzspannung der externen vorgezogen, wenn es auf Low gesetzt wird. Die Vorbelegung des Bit 13 (DAVF) führt zu keiner Veränderung, da es nur für eine Leseanweisung gedacht ist und zur Prüfung verwendet wird, ob die aktuelle A/D-Wandlung zu Ende ist und eine neue gestartet werden kann. Dies muss vor jedem Befehl überprüft werden. Schließlich wird das Bit 14 (RSTC) auf Low gesetzt, da kein Reset gewünscht ist. Es wird zunächst ein Befehl entsprechend der Tabelle 16 gesendet, mit welchem das ADC Status/Config Register angesprochen wird. Bit 15 ist auf Low-Pegel gesetzt, da es sich bei dem Befehl um eine Write-Anweisung handelt. Die Adresse des AMC Status Registers befindet sich auf Seite 0 der Register Map, daher ist das Bit 13 Low und Bit 12 High. Die Adresse lautet 0x0A (1010 binär) und wird sowohl als Start als auch als End-Adresse eingegeben. Anschließend wird der Befehl mit der Konfiguration des AMC Status /Konfiguration-Registers aus Tabelle 18 gesendet.

Bit 15 MSB	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	1	0	0	1	0	1	0	0	0	1	0	1	0

Tabelle 16: Write Befehl des AMC7823 ADC/DAC Bausteins zur Ansprache der ADC Status/Config Registers

Bit 15 MSB	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
X	RSTC	DAVF	0	X	X	X	X	SREF	GREF	ECNVT	X	X	X	X	X

Tabelle 17: ADC Status / Configuration Register Command Format des AMC7823 ADC/DAC Bausteins

Bit 15 MSB	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Tabelle 18: Konfiguration des ADC Status/Config Registers des AMC7823 ADC/DAC Bausteins im Format der Tabelle 6

5.2.2.3 Konfiguration des AMC7823 AMC/DAC Bausteins

Das DAC-Register ist ebenfalls entsprechend Tabelle 19 zu konfigurieren. Die Bits 15-8 (SLDA) sind für die Aktivierung der asynchronen Ladungsfunktion der Kanäle auf Low-Pegel zu setzen, da nicht mehrere Kanäle gleichzeitig gelesen werden sollen. Für synchrones Laden müsste der Befehl 0xBB00 anschließend ins LOAD-DAC-Register geschrieben werden, um das interne ILDAC Signal zu generieren. Bit 7-0 (GDAC) sind auf High-Pegel zu setzen, damit alle acht ADC-Kanäle einen Output-Range von 0-5V besitzen. Durch eine andere Vorbelegung der GDAC-Bits in Kombination mit einer anderen Belegung der GREF Bits im AMC-Status-Konfigurations-Register können auch weitere Messbereiche umgesetzt werden, welche jedoch im Rahmen dieser Arbeit nicht benötigt werden. Somit setzt sich der verwendete Befehl aus dem 16-Bit Command Format mit der 8-Bit Adresse (0x09) des DAC-Registers um dieses anzusprechen und den eigentlichen 16-Bit Daten (0Xff, siehe Tabelle 20) zusammen.

Bit 15 MSB	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SLDA7	SLDA6	SLDA5	SLDA4	SLDA3	SLDA2	SLDA1	SLDA0	GDAC7	GDAC6	GDAC5	GDAC4	GDAC3	GDAC2	GDAC1	GDAC0

Tabelle 19: Register Comand Format des AMC7823 ADC/DAC Bausteins für DAC-Register Konfiguration

Bit 15 MSB	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Tabelle 20: Befehl mit Daten zur DAC-Register Konfiguration des AMC7823 ADC/DAC Bausteins

5.2.2.4 ADC-Control-Register

Das MSB-Bit 15 des Command Formats (Tabelle 21) sagt aus, ob die Kanäle im Direct-mode, oder im Auto-Mode gelesen werden sollen. Mit den Bits 11-8 wird die Startadresse des ersten zu lesenden Kanals und in den Bits 7-4 die Endadresse des letzten zu lesenden Signals entsprechend Tabelle 23 eingetragen. Projektbezogen wird in diesem Register die Startadresse und die Endadresse gleich gewählt, da pro Befehl nur ein Kanal ausgelesen werden soll. Somit wurden in Tabelle 22 die Bits der Adresse gleich benannt. Die Ausführung des Befehls startet den Übertragungsprozess, also die Auslesung des gewählten Kanals. Zunächst muss jedoch das Register mit der Adresse 0x0B angesprochen werden. Ausschaltvorgänge und Resets leeren den Registerspeicher.

Bit 15 MSB	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CMODE	X	X	X	SA3	SA2	SA1	SA0	EA3	EA2	EA1	EA0	X	X	X	X

Tabelle 21: ADC Control Register Command Format des AMC7823 ADC/DAC Bausteins

Bit 15 MSB	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	0	A3	A2	A1	A0	A3	A2	A1	A0	0	0	0	0

Tabelle 22: Konfiguration des ADC Control Registers des AMC7823 ADC/DAC Bausteins

ICH3	ICH2	ICH1	ICH0	ANALOG INPUT
0	0	0	0	CH0
0	0	0	1	CH1
0	0	1	0	CH2
0	0	1	1	CH3
0	1	0	0	CH4
0	1	0	1	CH5
0	1	1	0	CH6
0	1	1	1	CH7
1	0	0	0	CH8

Tabelle 23: Analog Input Channel Map des AMC7823 ADC/DAC Bausteins [3]

5.2.2.5 ADC-Data-n-Register des AMC7823 ADC/DAC Bausteins

Nun soll der Befehl für die Auslesung der in den ADC Data-n Registern gespeicherten Daten beschrieben werden. Insgesamt existieren neun ADC Data-n Register. Das neunte Register ist für die Auslesung des im Board integrierten Temperatursensors vorgesehen. Ausschaltvorgänge und Resets leeren den Registerspeicher.

Das ADC Daten-n Register Command Format ist in Tabelle 24 dargestellt. Die ersten 12-Bits (Bit 0-11) enthalten den zurückgegebenen Wert der Kanalauslesung. Die Bits 12 bis 15 bestimmen den Kanal, dessen Daten aus dem ADC-Data-Register gelesen werden sollen und werden entsprechend Tabelle 24 belegt.

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ICH3	ICH2	ICH1	ICH0	ADC11	ADC10	ADC9	ADC8	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0

Tabelle 24: ADC Data-n Register Command Format des AMC7823 ADC/DAC Bausteins

ICH3	ICH2	ICH1	ICH0	ANALOG INPUT
0	0	0	0	CH0
0	0	0	1	CH1
0	0	1	0	CH2
0	0	1	1	CH3
0	1	0	0	CH4
0	1	0	1	CH5
0	1	1	0	CH6
0	1	1	1	CH7
1	0	0	0	CH8

Tabelle 25: ADC Data-n Register des AMC7823 ADC/DAC Bausteins [3]

5.2.2.6 DAC-n Data Register des AMC7823 ADC/DAC Bausteins

Im DAC-n Data Register werden Daten gespeichert, welche an die DAC-Kanäle übermittelt werden, sobald der Befehl gesendet wird. Wurden die Daten übertragen, so werden sie auch dem Latch-Speicher übergeben, wodurch der Speicher im Register für neue Daten frei wird. Wird die Auslesung der DAC-Kanäle durchgeführt, so wird der Wert des Latch-Speichers und nicht des Registers weitergereicht. Ausschaltvorgänge und Resets leeren den Registerspeicher.

Entsprechend des Commando Formats in Tabelle 26 sind die Daten in den ersten 12-Bits enthalten. Wie in Tabelle 27 gezeigt, entsprechen die Bits 12-14 der Adresse des zu den Daten zugehörigen DAC-Kanals. Das Bit 15 ist ein don't care Bit.

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
x	OCH2	OCH1	OCH0	DAC11	DAC10	DAC9	DAC8	DAC7	DAC6	DAC5	DAC4	DAC3	DAC2	DAC1	DAC0

Tabelle 26: DAC-n Command Format des AMC7823 ADC/DAC Bausteins

OCH2	OCH1	OCH0	ANALOG OUTPUT
0	0	0	DAC0
0	0	1	DAC 1
0	1	0	DAC 2
0	1	1	DAC 3
1	0	0	DAC 4
1	0	1	DAC 5
1	1	0	DAC 6
1	1	1	DAC 7

Tabelle 27: DAC-n Data Registers des AMC7823 ADC/DAC Bausteins [3]

5.2.2.7 GPIO Register des AMC7823 ADC/DAC Bausteins

Der AMC besitzt zwei bidirektionale digitale I/O Kanäle (GPIO-4 und GPIO-5). Zusätzlich enthält der Chip vier Dual-Purpose Pins, welche als ALR (Out of Range) Signale programmiert werden können, um den Wertebereich bestimmter Kanäle überwachen zu können. Dazu gibt es auch ein entsprechendes ALR Register. Beides wird jedoch im Rahmen dieses Projekts nicht benötigt, weshalb nicht näher drauf eingegangen wird. Die vier Dual-Purpose Pins können ebenfalls als GPIO Signale fungieren und werden in diesem Projekt zur Ansteuerung der vier LDOs der zweiten Platinenversion benutzt, welche am AMC angeschlossen sind.

Der Befehl zur Konfiguration des GPIO Registers ist in Tabelle 28 dargestellt. Die Bits 5-0 (IOST) sind Status-Bits zur Aktivierung/Deaktivierung der I/O-Kanäle. Mit den Bits 8-11(IOMOD) wird festgelegt, in welchem Mode die Dual-Purpose Pins konfiguriert werden sollen.

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	1	1	1	IOMOD3	IOMOD2	IOMOD1	IOMOD0	1	1	IOST5	IOST4	IOST3	IOST2	IOST1	IOST0

Tabelle 28: GPIO-Register Command Format des AMC7823 ADC/DAC Bausteins

6. Im Projekt verwendete Hardware Hardware

6.1 Rekonfigurierbare Rechnerplattform auf Basis eines Zedboards

In diesem Projekt wird das Zedboard des Herstellers Avnet als Entwicklungsboard verwendet, welches über viele Ressourcen verfügt und eine Vielzahl an Anschlüssen bietet (Abbildung 29). „Das ZedBoard ist ein kostengünstiges Entwicklungsboard für die Xilinx Zynq-7000 Extensible Processing Platform (EPP). Dieses Board stellt alle notwendigen Komponenten bereit um ein Linux, Android, Windows oder anderes OS/RTOS-basiertes Design auszuführen. Darüber hinaus stellen mehrere Erweiterungssteckverbinder das Verarbeitungssystem und die programmierbaren Logik-E/A für einen einfachen Benutzerzugriff bereit“ [17]. Einige wichtige Eigenschaften sind [17]:

- Dual-Core ARM Cortex-A9
- Speicher:
 - 512 MB DDR3
 - 256 MB Quad-SPI Flash
 - 4 GB SD-Karte
- Onboard USB-JTAG Programmierung
- 10/100/1000 Ethernet
- USB OTG 2.0 und USB-UART
- PS & PL I/O Erweiterung (FMC, Pmod, XADC)
- Verschiedene Displays (1080p HDMI, 8-Bit VGA, 128 x 32 OLED)

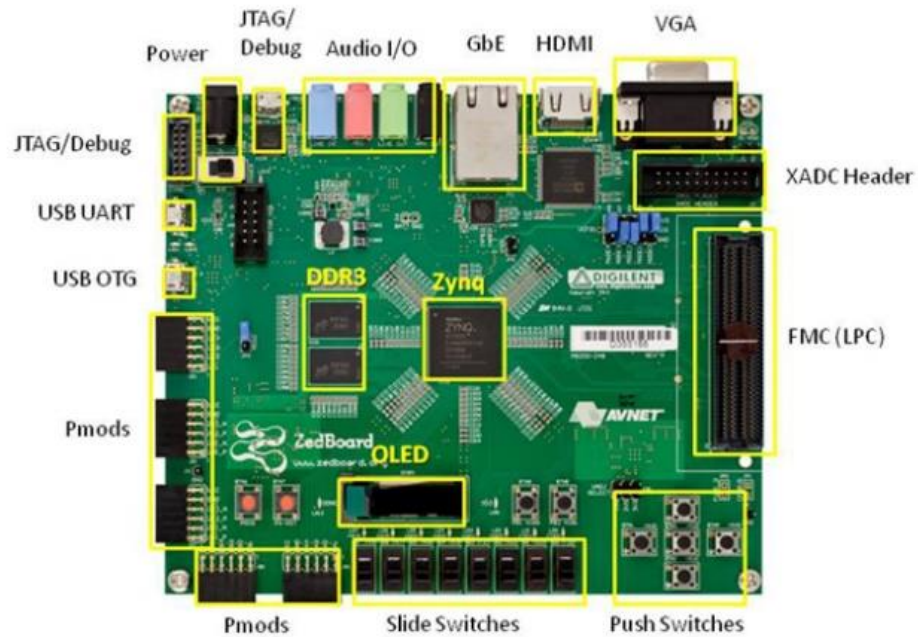


Abbildung 29: Zedboard [17]

6.2 Bereitstellung und Messung von Versorgungsspannungen und Signalen

Als Netzteil wurde das Modell HMC 8043 von Rohde & Schwarz gewählt, welches über drei Kanäle verfügt, die eine Spannung von bis zu 32V liefern und mit bis zu 3A belastet werden können. Der erste Kanal wurde für die Versorgungsspannung von 12V genutzt. Der zweite Kanal wurde für die Generierung von Spannungen zu Testzwecken genutzt und der dritte Kanal für die digitale Versorgungsspannung von 3,3V verwendet.

Als Messgerät diente das DMM6500 von Keithley aufgrund seiner hohen Messgenauigkeit von 100 nV, 10 pA und 1 μ Ohm im entsprechenden Messmodus. Zudem hat es insgesamt 6 Kanäle und 15 Funktionen. Es wurde für die Kalibrierung der Offsets und insbesondere für die Messreihe zur Prüfung der Ist-Spannung genutzt.

Als Oszilloskop wurde das Modell RTB2004 von Rohde & Schwarz genutzt. Es ist ein modernes 4-Kanal Oszilloskop mit hochauflösendem Touchscreen-Display, hoher Bandbreite von 300Mhz und weiteren zeitgemäßen Eigenschaften. Mit diesem Gerät wurden die SPI-Signale dargestellt und geprüft.

7. Software

7.1 Software für Leiterplattenentwurf

Altium Designer ist eine professionelle PCB-Design-Software für Wissenschaftler und Ingenieure für die Erstellung von Schaltplänen und Leiterplatten-Layouts. Altium besitzt eine Vielzahl von integrierten Bibliotheken, welche sich auch durch zufügen weiterer Bibliotheken erweitern lassen. Zudem können Tools zur Einbindung von Bauteilen eingefügt werden. Bei der Erstellung von Leiterplatten-Layouts ist die Umschaltung zwischen 2D- und 3D Körpern möglich. Zudem können Bauteile aus früheren Projekten eingebunden werden. Die Erstellung von Bauteillisten und die Kostenberechnungen sind ebenfalls möglich.

7.2 Software für den Schaltungsentwurf und Softwareentwicklung

7.2.1 Vivado und Vitis

Xilinx ist ein Hersteller von programmierbaren integrierten Schaltkreisen bzw. Field Programmable Gate Arrays (FPGAs). Petalinux ist für die Entwicklung von Linux-Anwendungen mit Xilinx-Produkten ausgelegt und beinhaltet alle nötigen Tools für eine reibungslose Funktion. Xilinx stellt zudem auch die Programme Vivado und Vitis für die Embedded Software-Entwicklung zur Verfügung. Dabei wird Vivado für das Hardwaredesign und Vitis für die Softwareentwicklung verwendet. Alle in diesem Projekt erstellten Quellcodes wurden unter Vitis (Version 21.2.0) mit der Programmiersprache C geschrieben und für die Ausführung auf dem Zedboard ausgelegt. Das Anlegen eines Vitis-Projekts soll mit Hilfe von Screenshots veranschaulicht werden. Zuvor müssen die Hardwareeinstellungen des Zedboards konfiguriert und der Hardwareentwurf im FPGA Teil des Xilinx Bausteins fertiggestellt worden sein, was bereits im Vorfeld durch Mitarbeiter des Labors für integrierten Schaltungsentwurf durchgeführt wurde. Im Rahmen dieses Projekts werden die bereitgestellten Konfigurationsdateien des Xilinx-Bausteins nur dem Projekt hinzugefügt, ohne das darauf weiter eingegangen wird. Deshalb wird auch das Programm Vivado nicht detailliert behandelt.

7.2.1 Projekterstellung Vitis

Nachdem das Vitis-Programm geöffnet wurde, wird das Platform Project mit dem Befehl File->New->Platform Project erzeugt, welches die Hardwareinformationen und Softwareumgebung durch Einbindung der FPGA-Konfigurationsdateien erhält. Im nächsten Fenster (Abbildung 30) ist der gewünschte Name des Projekts einzutragen. Anschließend erscheint das Fenster (Abbildung 31) in welchem das XSA-File der FPGA-Konfiguration einzubinden ist. Des Weiteren wird als „Operating System“ Linux gewählt und mit „Finish“ bestätigt. Dieser Vorgang kann eine Minute dauern.

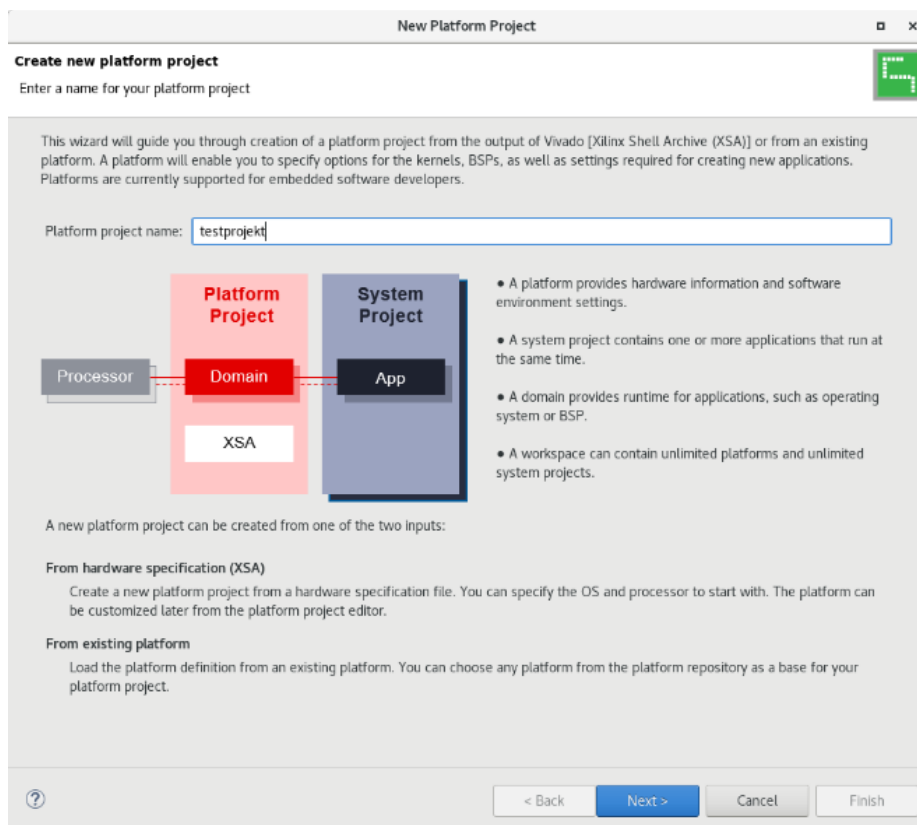


Abbildung 30: Maske zur Eintragung des Projektnamens

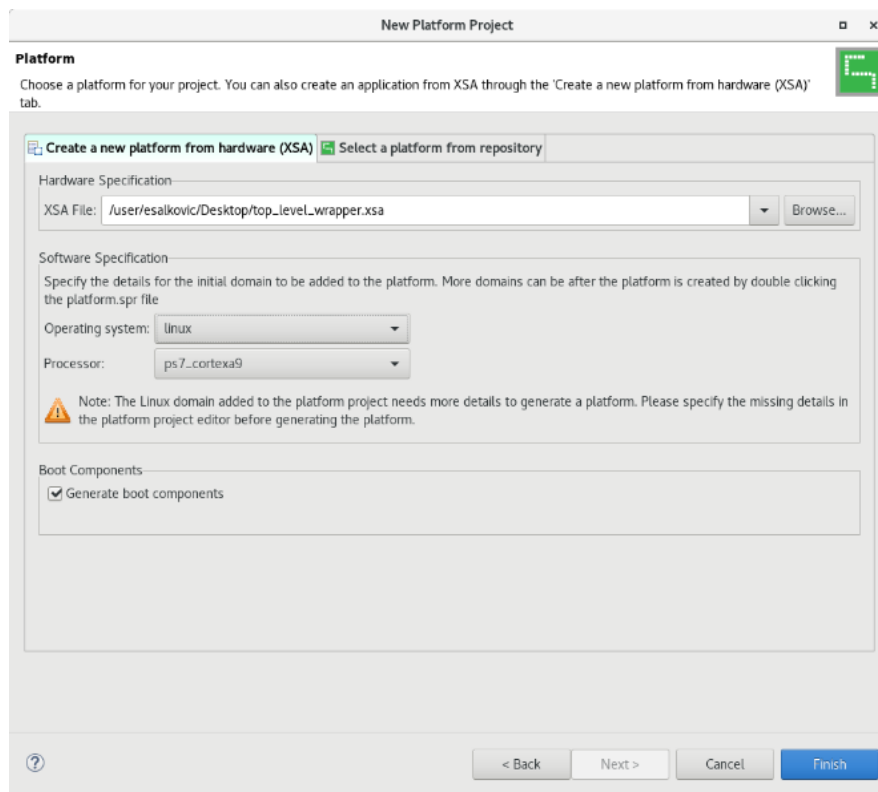


Abbildung 31: Eingabe des Pfads vom XSA-File

Nachdem die Plattform erzeugt wurde, klickt man links im Projektordner auf den Reiter `linux_on_ps7_cortex_9` entsprechend der Abbildung 32. Dort wird das Verzeichnis der Sysroot Directory Datei aus dem Petalinux SDK mit dem Namen „cortexa9t2hf-neon-xilinx-linux-gnueabi“ eingebunden. Anschließend wird das Application Projekt erzeugt, welchem man einen Namen zuweisen muss (Abbildung 33).

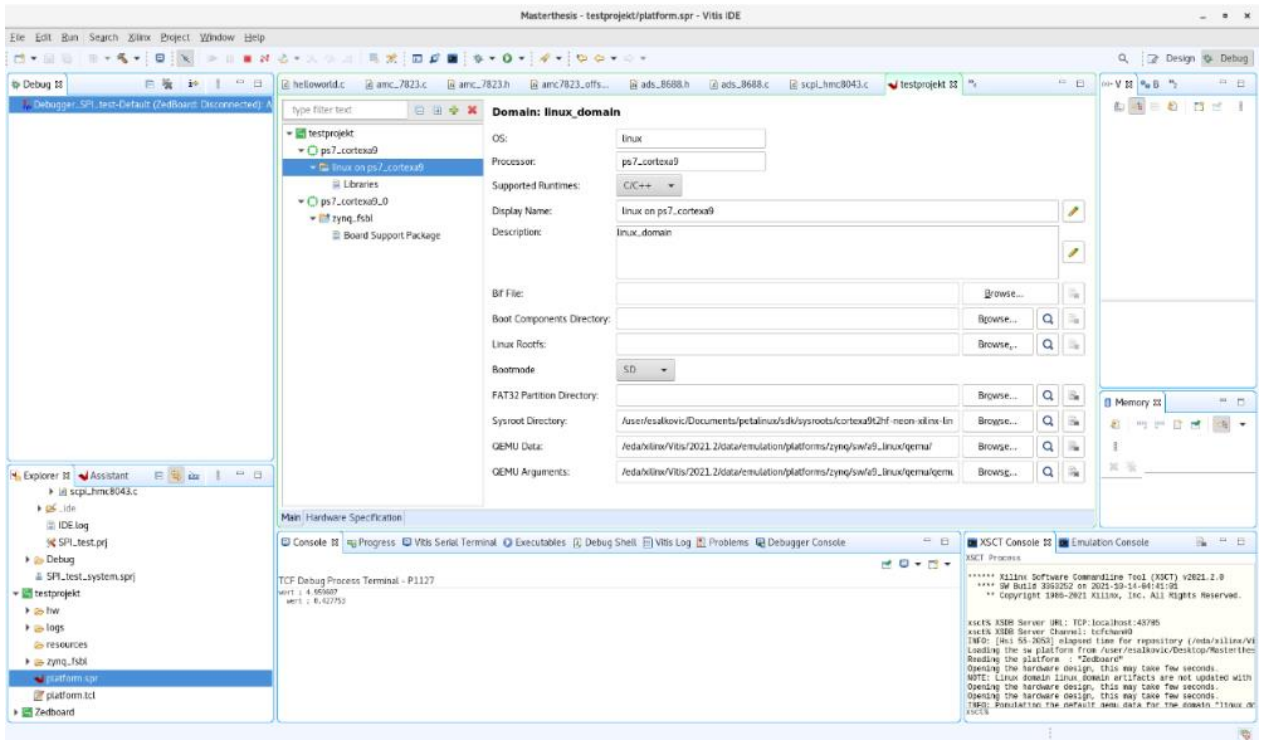


Abbildung 32: Einbindung der FPGA-Konfigurationsverzeichnisse

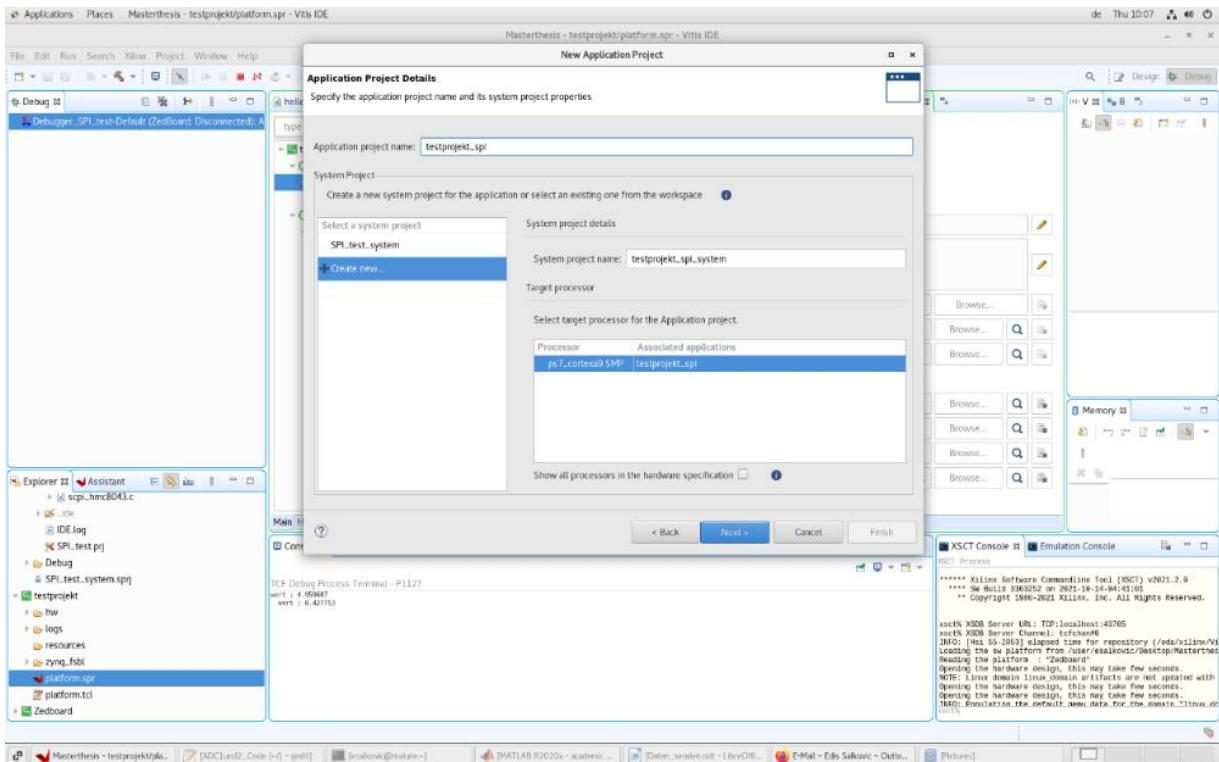


Abbildung 33: Maske zur Eingabe des Applikationsprojektnamens

Im nächsten Fenster (Abbildung 34) ist der Sysroot Path (cortexa9t2hf-neon-xilinx-linux-gnueabi) erneut einzubinden. Darüber hinaus das Root FS (rootfs.ext4) und das Kernel Image (image.ub).

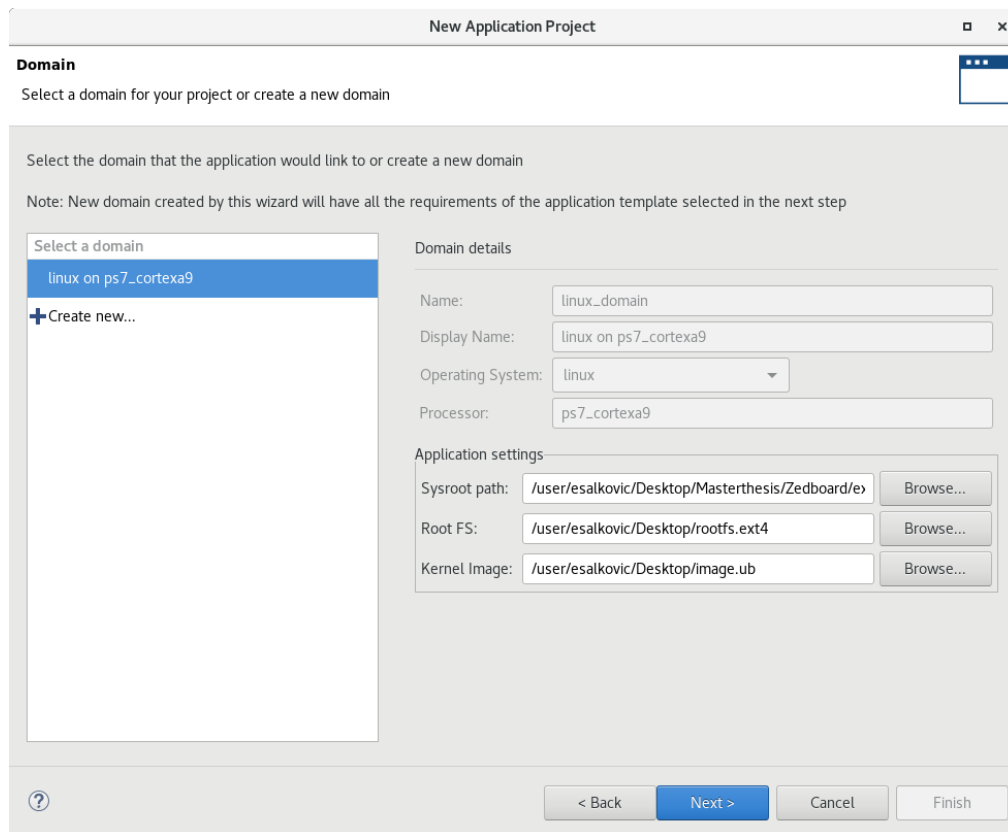


Abbildung 34: Einbindung weiterer Konfigurationsdateien

Nun folgt die Anlegung des Application Projects indem man auf den Reiter File->New->Application Project klickt, „Linux Hello World“ auswählt und bestätigt. Links im Projektverzeichniss ist ein „Chip-Symbol“ dargestellt (Abbildung 35), wobei der Name des Projekts neben dem Symbol angezeigt wird. Mit Rechtsklick->Build Project wird das Application Project kompiliert. Die Vollendung des Kompilationsvorgangs wird in der Console durch „Build finished“ angezeigt. Nun folgt erneut ein Rechtsklick auf das Chip-Symbol und anschließend „Debug As->Debug Configurations->Single Application Debug“(Abbildung 36). Nun wird unter Project der Projektname und unter Connection „Linux Agent“ angezeigt. Der Button „New“ wird betätigt, was ein Fenster mit „New Target Connection“ (Abbildung 35) erscheinen lässt, welches für die Verbindung über Ethernet mit dem Zedboard notwendig ist. Der Target Name, welcher

8 Programmierung der Bausteine

Die Programmierung des ARM-Mikrocontrollers teilt sich in drei Abschnitte. Im ersten Abschnitt wird die Steuerung und die Auslesung der ADS8688 ICs programmiert während im zweiten Abschnitt die Funktionen des AMC7823-Bausteins implementiert werden. Der dritte Abschnitt entspricht einem Algorithmus zur Extraktion von Messabweichungen. Die im dritten Programmabschnitt ermittelten Messdaten, werden mit Hilfe eines Matlab-Skripts ausgewertet und zur Korrektur der Messabweichungen verwendet. Jeder Softwareteil wird in eine eigene .c – Datei hinterlegt und die vom Nutzer benötigten Funktionen in eine .h -Datei gelistet, wodurch die Nutzung der Funktionen in der Main-Funktion erleichtert wird.

Die Programmfunktionen sollen mit geringem Hintergrundwissen verwendbar sein, daher werden möglichst viele Konfigurationsschritte im Code abgedeckt, so dass in der Main-Datei die Funktionen direkt ausgeführt werden können. Eine ausführliche Auskommentierung steigert die Lesbarkeit des Quellcodes und soll dem besseren Verständnis der Funktionen dienen.

8.1 Programmierung des ADC-Bausteins

8.1.1 Erstellung der Funktionen zur Auslesung und Steuerung des ADC-Bausteins

Der erste Teil der Programmierung enthält die in der Tabelle 27 aufgeführten Funktionen. Die transfer-Funktion ist hierbei eine Funktion aus der Bibliothek spidev.h. Die Funktionen 2 bis 7 sind Funktionen, welche dem Nutzer zur Verfügung stehen und die restlichen Funktionen sind Unterfunktionen. Alle Funktionen werden ausführlich erklärt und dessen Quellcodes erläutert. Quellcodes mit ähnlichem Inhalt werden nicht gezeigt, um den Umfang der Arbeit nicht zu sprengen, jedoch dem Anhang kommentiert beigelegt.

	Name der Funktion	Funktion
1.	transfer	Transfer des Befehle an Mikrocontroller
2.	Ads8688_read_channel	Spannungswert auslesen
3.	Set_gain	Meßbereich einstellen
4.	Set_all_gain	Gl. Meßbereich an allen Kanälen einstellen
5.	Read_gain	Befehl Meßbereich einlesen
6.	Print_gain	Befehl Meßbereich ausgeben
7.	Read_current	Strom vom ADC 2 ausgeben
8.	write_register	Bereitet Befehl für Schreibvorgang vor
9.	read_register	Bereitet Befehl für Lesevorgang vor
10.	Read_channel_raw	Unterfunktion von read_channel, empfängt Rohdaten vom ADC
11.	Conv_raw_value	Unterfunktion von read_channel, rechnet Rohdaten in logische Spannungswerte um

Tabelle 29: Funktionen für die Ansteuerung des ADC-Bausteins

8.1.2 Funktion für die SPI Datenübertragung

In der Datei ads_8688.c befinden sich die Quellcodes für den ersten Teil, also für die Programmierung der Funktionen zur Ansteuerung der ADC-Bausteine vom Typ ADS8688. Als erstes wird die die Transferfunktion aus der Bibliothek spidev.h eingefügt. Diese enthält als Parameter einen Zeiger auf den tx_buffer für Sendedaten, einen Zeiger auf den rx_buffer für empfangene Daten, die Länge der zu sendenden Daten und den Filedescriptor fd, welcher für den Systemaufruf IOCTL notwendig ist. In der spidev.h Bibliothek sind auch die unterschiedlichen SPI-Modi definiert, welche in der Main Datei bestimmt werden können. Die Eigenschaften der Daten werden im struct spi_ioc_transfer beschrieben und außerhalb der Funktion bestimmt. Mit dem IOCTL Systemaufruf wird die eigentliche Übertragung. Der IOCTL-Aufruf bildet sozusagen die Schnittstelle zum Zedboard und enthält den File-Descriptor, welcher in der Main-Funktion beim Öffnen der SPI-Schnittstelle des Zedboards definiert wird. Sind die gesendeten Daten der bestimmten Datenlänge übertragen worden, so wird die erfolgreiche Übertragung mit einem „return 0,“ bestätigt .

```
29 static int transfer(int fd, uint8_t *tx, uint8_t *rx, int length)
30 {
31     int ret;
32     struct spi_ioc_transfer tr = {
33         .tx_buf = (unsigned long)tx, // Sendedaten
34         .rx_buf = (unsigned long)rx, // Empfangsdaten
35         .len = length, // Länge Daten
36         .delay_usecs = delay, // Verzögerung
37         .speed_hz = speed, // Geschwindigkeit
38         .bits_per_word = bits, // Bitgröße pro wort
39     };
40     ret = ioctl(fd, SPI_IOC_MESSAGE(1), &tr); // Systemaufruf
41     if (ret == length )
42     {
43         return 0;
44     }
45     else
46     {
47         return -1;
48     }
49 }
```

Listing 1: Code der Transferfunktion für die Verbindung mit der SPI-Schnittstelle

8.1.3 Funktionen zum Lesen und Schreiben von Registern

Die `write_register()`-Funktion sendet Daten und enthält drei Übergabeparameter (Zeile 58). Wie jede andere Datei erhält sie den Filedescriptor `fd` zum Zugriff auf die korrekte SPI-Schnittstelle. Zudem erhält sie die Adresse, an der die gewünschten Daten gesendet werden sollen und die Daten selbst. Weiterhin werden die `tx` und `rx` Buffer mit dem Datentyp `uint_8_t` definiert, welche $3 \cdot 8 = 24$ Bits speichern können, die entweder gesendet werden sollen, oder empfangen worden sind (Zeile 58). Dabei enthält `tx[0]` die Adresse (7 Bits) und die Write-Anweisung (1 Bit), `tx[1]` enthält die Daten (8 Bit), während die letzten 8 Bits von `tx[2]` don't care Bits sind. Während dieser 8 Takte werden die zuvor gesendeten 8 Bits an Daten in `rx[2]` empfangen, um anschließend prüfen zu können, ob diese korrekt versendet worden sind (Zeile 72). Damit das Write-Bit an seiner richtigen Stelle positioniert wird, müssen die ersten 8-Bits der zu sendenden Daten (`tx[0]`) um ein Bit nach links verschoben und das Write-Bit an der ersten Stelle mit einer „oder“-Operation angehängt werden. Somit wird die Adresse nicht verändert und es ergibt sich der gewünschte High-Pegel am Bit 8 entsprechend des Write Cycle Command Words der Write-Anweisung.

Die `read_register()`-Funktion empfängt Daten und ist nach gleichem Prinzip entworfen, wobei diese als Übergabewert einen Pointer bekommt, welcher auf eine Variable zeigt, in der die empfangenen Daten gespeichert werden. Zudem sind die in den Buffern enthaltenen Informationen nun andere, da `tx` nur die Anweisung zur Auslesung der gewünschten Daten ans Register sendet und keine empfangenen Daten enthält.

Die durch die Read-Anweisung empfangenen Rohdaten müssen einen Umwandlungsprozess durchlaufen, um sie für den Nutzer zur besseren Handhabung aufzubereiten.

```
56 int write_register(int fd, uint8_t address, uint8_t data)
57 { uint8_t tx[3], rx[3]; // 24 SCLK, 7 Adress und 1 W/R, 8 DATA, 8 NULL
58   tx[0]=(address<<1) | 0x01;
59 //Erste 7Bits enthalten Adresse und werden nach links geschoben
60 // Das eine Write-Bit wird an der ersten Stelle mit Hilfe einer
61 // "oder" Operation angehängt
62   tx[1]=data; // 8 Bit Daten
63   tx[2]=0x00; // 8 dont care bits
64   int length=3; //Länge der gesamten Übertragung immer 3*8bit
65   transfer(fd,tx,rx,length);
66   if (tx[1]==rx[2]){ //Empfangene Daten mit gesendeten verglichen
67     return 0;
68   } else {
69     return -2;
70   }
71 }
```

Listing 2: Code der write-register Funktion zum Senden von Daten an Register

8.1.4 Funktion zum Empfang von ADC Spannungswerten

Mit der Funktion `ads8688_read_channel()` wird die Spannung am gewünschten Kanal des ADCs ausgelesen. Diese enthält als Parameter neben dem File-Descriptor (`fd`) den gewünschten Kanal bei gewünschter Verstärkung und eine Pointer-Variable „value“ zur Speicherung des Spannungswertes. Für die unterschiedlichen Verstärkungen wurde zuvor eine Enum-Aufzählung mit dem Namen `ads8688_gain` erstellt. Der erste Schritt zur Auslesung der Spannung ist der Aufruf der Funktion `read_channel_raw()`, (Zeile 289) welche in einer If-Anweisung geschachtelt wird. Wenn ein Fehler-Code innerhalb der Funktion auftritt, wird ein Fehler ausgegeben (Zeile 291), ansonsten wird die Funktion `conv_raw_value()` ausgeführt und deren Return-Wert der Pointer-Variable „value“ übergeben (Zeile 294).

Die `read_channel_raw()`- Funktion enthält ebenfalls den ADC-Kanal als Parameter und „value“ als Pointer-Variable. Nachdem durch eine Switch-Anweisung die Adresse des gewünschten Kanals bestimmt wird, kann der Befehl zum Lesen gesendet werden. Der Rohwert wird in der Variable „value“ von der `read_channel_raw()`- Funktion gespeichert, jedoch anschließend in die Variable `adc_value` der Funktion `ads8688_read_channel()` kopiert.

Die Funktion `conv_raw_value()` ist dafür gedacht, die von der Funktion `adc_value()` erhaltenen Rohdaten in logische Spannungswerte umzurechnen. Sie enthält die

Verstärkung des ADC-Kanals und eine Pointer-Variable „value“ für den Rückgabewert. Zudem enthält die Funktion einen Offset, welcher für die Berechnung der Messbereiche notwendig ist, die auch negative Werte annehmen können. Der Wert wird dann der ursprünglichen Funktion `8688_read_channel()` in der `Main()`-Funktion in „value“ übergeben. Die Berechnung erfolgt an Hand der Formel 8.1, die bereits im Kapitel DAC erläutert wurde. Hierbei muss jedoch beachtet werden, dass bei einer Berechnung negativer Vorzeichen der `Offset=0x8000` übergeben werden muss:

$$\text{Spannung} = (\text{WERT}_{\text{adc}} - \text{Offset}) * \frac{\text{Maximale Spannung}}{\text{Auflösung}} \quad 8.1$$

Für den Fall, dass die ADC Kanäle Messabweichungen aufweisen würden, die über den gesamten Messbereich konstant sind, so könnten diese in der Formel berücksichtigt werden. Bei variablen Fehlern ist es effektiver die Fehler individuell über den gesamten Messbereich zu erfassen und in einer Look-Up Table zu speichern. Die Vorgehensweise wird beispielsweise auf die ADC-Kanäle des ADC/DAC-Bausteins AMC7823 angewendet. Die so ermittelten Korrekturfaktoren werden in der Funktion `conv_raw_value()` eingebunden und bei Bedarf angewendet. Eine Formel zur Umrechnung ist in diesem Fall überflüssig, da die Datensätze bereits den Spannungswert enthalten.

```
283 int ads8688_read_channel(int fd, int channel, enum ads8688_gain
284 gain, float * value)
285
286 {
287     uint16_t adc_value;
288
289     if ((read_channel_raw(fd,channel, &adc_value))<0) //if anweisung
290 und ausführung
291     {
292         return -7;
293     }
294     *value=conv_raw_value(adc_value, gain); // returnwert von
295 conv_raw_vewlue ist value;
296     return 0;
297 }
```

Listing 3: Code der `ads8688_read_channel()`-Funktion zum Auslesen der Daten aus ADC-Kanälen der ADC-Bausteine


```
255 float conv_raw_value(uint16_t raw_value, enum ads8688_gain gain){
256 float verstaerkung;
257 float recievedata;
258 uint16_t offset;
259
260 switch( gain){
261
262 case AC10V:verstaerkung=2*2.5*4.096;
263         offset=0x8000; break;
264 case AC5V: verstaerkung=2*1.25*4.096;
265         offset=0x8000;break;
266 case AC2V5: verstaerkung=2*0.625*4.096;
267         offset=0x8000;break;
268
269 case DC10V: verstaerkung=2.5*4.096;
270         offset=0x0000;break;
271 case DC5V: verstaerkung=1.25*4.096;
272         offset=0x0000;break;
273 }
274 //formel zur umrechnung der empfangenen Daten
275 recievedata=(raw_value - offset)*(verstaerkung)/(0xFFFF);
276 return recievedata;
277 }
```

Listing 4: Code der conv_raw_value()-Funktion zur Umrechnung der empfangenen Rohwerte der ADC-Kanäle

```
279 int read_channel_raw(int fd, int channel, uint16_t * value)
280 {
281     // Kanalabfrage ADC1 Kanal 1-8
282     uint8_t command_1[4];
283     uint8_t command_2[4];
284
285     switch(channel){
286
287     case 0:     command_1[0]=0xC0; //Adresse ch1 gemäß comand
288                //register map seite 45
289                break;
290     case 1:     command_1[0]=0xC4;
291                break;
292     case 2:     command_1[0]=0xC8;
293                break;
294     case 3:     command_1[0]=0xCC;
295                break;
296     case 4:     command_1[0]=0xD0;
297                break;
298     case 5:     command_1[0]=0xD4;
299                break;
300     case 6:     command_1[0]=0xD8;
301                break;
302     case 7:     command_1[0]=0xDC;
303                break;
304
305     }
306     command_1[1]=0x00; // Teil der Adresse
307     command_1[2]=0x00; //dont care
308 }
```

```
309     command_1[3]=0x00; //dont care
310
311     command_2[0]=0x00; //continue operation
312     command_2[1]=0x00; //continue operation
313     command_2[2]=0x00; //dont care
314     command_2[3]=0x00; //dont care
315     uint8_t rx[4];
316
317     if(transfer(fd,command_1,rx,4)!=0) //if Anweisung und zugleich
318     ausführung der fkt
319     {
320         //length=4 Rx von ersten
321     Tranfer uninteressant
322         return -5;
323     }
324     //erst die daten vom zweiten rx interessant
325     // daher können die ersten überschrieben werden
326     if (transfer(fd,command_2,rx,4)!=0)
327     {
328         return -6;
329     }
330     *value=(rx[2]<<8)|rx[3]; //value 16bit, daher rx[2] und rx[3]
331     zusammensetzen
332
333     return 0;
334 }
335
```

Listing 5: Code der Unterfunktion `read_channel_raw()`, welche den gewünschten Kanal selektiert und die Daten abspeichert

8.1.5 Funktion zur Definition der Messbereiche

Mit den erstellten Funktionen ist es möglich, den Messbereich einzelner Kanäle mit `set_gain()` und aller Kanäle mit `set_all_gain()` zu bestimmen. Darüber hinaus kann die Verstärkung mit `read_gain()` gelesen und mit `print_gain()` ausgegeben werden.

Die Funktion `set_gain()` enthält als Parameter den gewünschten Kanal (`channel`) und die gewünschte Verstärkung (`gain`). Innerhalb der Funktion selektiert eine `switch`-Anweisung die Adresse des gewünschten Kanals (Listing 5, Zeile 89-98) und speichert diese in der Variable „`data`“ (Zeile 99). Da „`data`“ vom Typ `uint8_t` ist, die Adresse jedoch nur 4 Bits groß ist, werden die 4 fehlenden Bits durch eine UND-Verknüpfung mit dem Wert `0x0F` mit Nullen gefüllt, die Adresse jedoch nicht verändert (`1&1=1` und `1&0=0`). Schließlich wird die `write_register()` Funktion mit den ausgewählten Daten und der Adresse gekoppelt an eine `if`-Anweisung ausgeführt (Zeile 102). Somit würde ein Fehler ausgegeben werden, wenn die Funktion nicht ordnungsgemäß ausgeführt werden kann.

```
86 int set_gain(int fd, int channel, enum ads8688_gain gain)
87 {
88     uint8_t address;
89     switch (channel) {
90         case 0: address=0x05; break;
91         case 1: address=0x06; break;
92         case 2: address=0x07; break;
93         case 3: address=0x08; break;
94         case 4: address=0x09; break;
95         case 5: address=0x0A; break;
96         case 6: address=0x0B; break;
97         case 7: address=0x0C; break;
98     }
99     uint8_t data=gain & 0x0F;
100     //da gain 4 bit, data jedoch 8 bits, zufügen von 4 unrelevanten
101     bits nötig
102     if (write_register(fd,address,data)){ // Ausführung und
103     Prüfung ob write_register Funktion ok
104         return 0;
105     }
106     else {
107         return -3;
108     }
109 }
```

Listing 6: Code der set_gain()-Funktion zur einstellung des Messbereiches beim ADC-Baustein

Die Funktion set_all_gain() enthält als Parameter lediglich die gewünschte Verstärkung an allen Kanälen. Dabei wird die set_gain() Funktion in einer Schleife ausgeführt, um die Verstärkung bei allen Kanälen einzustellen.

Mit der Funktion read_gain(), welche als Parameter den angefragten Kanal und eine Pointer-Variable für den Rückgabewert enthält, erfolgt durch eine Switch-Anweisung die Selektion der Adresse und Ausführung der read_register()-Funktion.

Die Funktion print_gain() kann in die Funktion read_gain() eingefügt werden und gibt lediglich den Wertebereich der entsprechenden Verstärkungen aus. Bei einer Verstärkung von DC5V gemäß der Enum-Aufzählung würde die Ausgabe „Gain: DC5V=+5V“ lauten.

8.1.6 Funktion zur Erfassung von Strömen

Mit der Funktion `read_current()` ist es möglich Ströme über die Kanäle des zweiten ADS8688-Bausteins zu bestimmen, indem die an den analogen Kanälen des ADC anliegende Spannung gelesen und mit Hilfe der Werte der Shunt-Widerstände und der Verstärkung der Instrumentenverstärkung in Stromwerte umgerechnet wird.

Außerdem wird ein Pointer übergeben, der auf die Variable zeigt, in welcher der ausgelesene Wert gespeichert wird. Es wird als erstes mit `ads8688_read_channel()` und einem festen Spannungsbereich von 0-5V der angefragte Kanal eingelesen. Der positive Spannungsbereich liegt im Bereich von 2,5V-5V. Demnach bedeuten 2,5V an allen 8 Kanälen, dass kein Strom fließt. 5 V bedeuten, dass der maximale Strom fließt. Der Bereich von 0-2,5V ist negativen Ströme vorbehalten. Die ersten vier Kanäle sind für die kleinen Ströme bis 20mA und die letzten vier Kanäle für großen Ströme bis 1A ausgelegt. Jeder Kanal besitzt eine individuelle Umrechnung der gemessenen Spannungen in den entsprechenden Strömen, da der Offsetfehler, welcher in der Funktion berücksichtigt wird, sich von Kanal zu Kanal unterscheidet. Daher muss entweder eine „If-else“-Anweisung, oder eine switch-Anweisung verwendet werden, um zum jeweiligen Kanal den zugehörigen Datensatz auszuwählen.

8.2 Programmierung des AMC7823 ADC/DAC Bausteins

8.2.1 Erstellte Funktionen

Für den AMC7823-Baustein wurde eine Reihe von Funktionen geschrieben, welche in Tabelle 30 dargestellt sind. Wie der ADS8688 verfügt auch der AMC7823-Baustein über acht ADC-Kanäle, deren Spannung nach gleichem Prinzip, jedoch durch unterschiedliche Befehle ausgelesen werden können. Das gleiche Prinzip der Registeransprache- und Auslesung findet sich auch in den write und read-Register Funktionen wieder. Der AMC7823-Baustein verfügt über eine zweiseitige Register Map (Page 0 und 1), welche im Command-Format berücksichtigt werden muss. Der ADS8688-Baustein besitzt dagegen über zwei einseitige Register. Der AMC7823-Baustein verfügt zudem 8 DAC-Kanäle, mit denen die gewünschte Spannung der 4 LDOs und 4 OPAs eingestellt werden können. Die ADS8688-Bausteine dagegen sind reine ADC-Bausteine. Dazu wurden die Funktionen `set_voltage_opa()` und `set_ldo_voltage()` entworfen. Zudem hat der AMC mehrere GPIOs, wovon vier zur Ansteuerung der LDOs, benutzt und über die Funktion `gpio_control()` gesetzt werden können. Schließlich benötigt der AMC noch eine Reihe an Konfigurationen für seine vielen Register, welche die Arbeitsweise des AMC7823 bestimmen. Dafür wurden die Funktionen 4-9 entworfen und in der Main- Funktion am Anfang eingebunden. Auf die Funktionen, welche sich von denen des ADC-Bausteins unterscheiden, wird im Folgenden etwas näher eingegangen.

	Name der Funktion	Funktion
1.	transfer	Transfer des Befehle an Mikrocontroller
2.	write_register	Bereitet Befehl für Schreibvorgang vor
3.	read_register	Bereitet Befehl für Lesevorgang vor
4.	amc_status_config	Konfiguration AMC Status Register
5.	dac_config	Konfiguration DAC Register
6.	power_down_register	Konfiguration Power Down Register
7.	adc_control_register	Konfiguration ADC control Register
8.	gpio_control	Ansteuerung LDO
9.	load_dac_register	Laden DAC Register
10.	set_voltage_opa	Setze gewünschte Spannung an OPA
11.	set_ldo_voltage	Setze gewünschte Spannung an LDO
12.	Amc7823_read_channel	Ablesung ADC-Kanäle
13.	Read_temperature	Ablesung des neunten ADC-Kanals (ADC8)
14.	read_adc_channel_raw	Ablesung Rohdaten in AMC7823_read_channel
15.	Conv_raw_value	Verarbeitung Roh-Daten bzw. Einbindung von Daten nach Korrektur der Messabweichung
16.	Check_conv	Abfrage ob aktuelle conversion zu Ende ist

Tabelle 30: Funktionen zur Auslesung und Steuerung des AMC7823-Bausteins

8.2.2 Funktionen zur Konfiguration des AMC7823-Bausteins

Die Konfiguration des AMC7823-Bausteins ist auf den Verwendungszweck im Rahmen der Masterthesis ausgerichtet, kann jedoch im Quellcode nach Bedarf verändert werden. Der Inhalt der Daten wird im Quelltext ausführlich erklärt und elektronisch angehängen, sodass eine zusätzliche Erklärung in der Thesis überflüssig ist. In der Funktion `amc_status_config()` wird das AMC-Status-Register konfiguriert, mit dessen Hilfe der Messbereich und die Nutzung des ADC-Triggers festgelegt werden kann. Hier findet sich auch das DAVF Bit, welches mit einer gesonderten Funktion `check_conv()` gelesen wird und anzeigt, ob die aktuelle Datenübertragung zu Ende ist, oder noch andauert. In der `dac_config()` Funktion wird die Konfiguration des DAC-Registers vorgenommen. An dieser Stelle wird die asynchrone Lade-Funktion bei allen Kanälen aktiviert, welche stets nacheinander und nie parallel eingelesen werden. Zusätzlich wird ein Output-Range von 0-5V eingestellt.

Die `power_down_register()`-Funktion wird ebenfalls am Anfang der Main-Funktion aufgerufen und beinhaltet einige Einstellungen zur Funktionsweise des ICs. Es werden die ADC-Kanäle, DAC-Kanäle und Präzisionsstromquelle auf den normalen operierenden Modus eingestellt und der Reference Buffer eingeschaltet.

Die Funktion `adc_control_register()` unterscheidet sich von den anderen in diesem Abschnitt behandelten Registerfunktionen, da sie den Parameter `channel` enthält, welcher ihr übergeben werden muss. Diese Funktion muss in der Read-Anweisung verwendet werden, da sie die Information enthält, welcher Kanal ausgelesen werden soll und in welchem Mode die Auslesung stattfinden soll. Im Rahmen dieses Projekts wird der Direct-Mode verwendet, damit jeder Kanal einzeln abgefragt werden kann. Die Auslesung kann jedoch auch automatisch über mehrere Kanäle erfolgen, da in diesem Command-Format sowohl die Anfangsadresse als auch Endadresse einzufügen sind. Bei der Auslesung eines einzigen Kanals ist die Anfangs- und Endadresse gleich.

Eine weitere Besonderheit des AMC7823 ist, dass der aktuelle Status der A/D-Wandlung abgefragt werden kann, was bereits bei der Beschreibung der

AMC_status_configuration()-Funktion erläutert wurde. Dies geschieht mit Hilfe der Funktion check_conv(), welche das Bit 13 DAVF aus dem 16-Bit Datensatz (Tabelle 17) mit Hilfe einer Schiebeoperation extrahiert und in die Variable data zur weiteren Verwendung abspeichert.

8.2.3 Funktionen zur Einstellung der DAC-Spannungen

Die DAC-Kanäle werden für die Einstellung der Ausgangsspannung der OPAs und LDOs nach den Wünschen des Nutzers verwendet. Dazu sind zwei separate Funktionen erstellt worden.

Die Funktion set_voltage_opa() ist für die Einstellung der Spannung an den OPAs zuständig und bekommt als Parameter den gewünschten OPA und die einzustellende Spannung zugewiesen. Eine switch-Anweisung selektiert die gewünschte Adresse und rechnet die angegebene gewünschte Spannung aus. Zudem wird eine kanalabhängige Offset-Korrektur eingeführt, nachdem der Offset durch eine Messreihe bestimmt wurde. Diese Informationen d.h. Kanaladresse und gegebenenfalls korrigierter DAC-Wert werden schließlich mit der Ausführung der write_register() Funktion ins entsprechende Register geschrieben.

Die Funktion set_ldo_voltage() funktioniert nach gleichem Prinzip, wobei der gewünschte LDO erst aktiviert werden muss. Hierfür wird die Funktion gpio_control() verwendet, indem das korrespondierende Bit verändert wird, um den LDO entweder einzuschalten, oder auszuschalten. Da immer ein kompletter Datensatz gesendet wird und nicht einzeln auf das Bit zugegriffen werden kann, muss der aktuelle Datensatz mit dem neuen Datensatz verknüpft werden. Zum Einschalten wird der entsprechende LDO ausgewählt und mit Set=0 aktiviert. Innerhalb der Funktion wird der aktuelle Datensatz des GPIO-Registers ausgelesen und mit einer Maske UND-verknüpft. Dabei enthält die Maske überall Einsen, außer an der Stelle des korrespondierenden GPIOs, wo sich eine logische „0“ befindet. Somit wird nur dieses Bit verändert. An den übrigen Stellen, wo zuvor eine „0“ vorzufinden war, bleibt durch „0&1“ weiterhin eine „0“ erhalten und wo vorher eine 1 war, bleibt durch 1&1 weiterhin die „1“ bestehen. Für

die Deaktivierung des LDO wird der vorhandene Datensatz mit einer Maske ODER-verknüpft verknüpft und die korrespondierende Stelle der Maske mit „1“ belegt.

Mit der Funktion `amc7823_read_channel()` werden die ADC-Kanäle des AMC7823 ausgelesen. Das Prinzip gleicht der Vorgehensweise beim ADS8688 jedoch mit dem Unterschied, dass die Daten, welche durch Messabweichungskorrektur entstehen, für jeden Kanal in eine Headerdatei „`amc_7823_offset.h`“ eingepflegt werden. Anstatt, dass empfangene Rohdaten in der Funktion `conv_raw()` umgerechnet werden, wird der korrigierte Spannungswert nach Auswahl des entsprechenden Kanals über eine `switch`-Anweisung ausgegeben.

8.3 Main()-Funktion zum Test der Funktionalität aller Bausteine

In der Main Funktion werden zunächst gewünschte Variablen deklariert und ihnen Werte zugewiesen (Zeile 6-10). Mit der Variable „channel“ werden die gewünschten analogen Kanäle der ADC-Bausteine und des ADC/DAC-Bausteins gewählt. Mit den Variablen „opa“ und „ldo“ wird der gewünschte OPA und LDO selektiert und der gewünschte Zustand des LDOs mit der Variablen „set“ bestimmt, welche in der Funktion gpio_control zur Ansteuerung der LDOs benötigt wird. Anschließend wird der File-Descriptor deklariert, mit der Variablen „voltage“ wird die gewünschte Spannung vorgegeben, welche sich an den LDOs und OPAs des ADC/DAC-Bausteins einstellen soll. In die Variable „wert“ werden die von den Bausteinen zurückgegebenen Werte abgespeichert, um dann ausgegeben werden zu können. Anschließend wird die SPI-Schnittstelle geöffnet und der File-Descriptor zugewiesen. Zusätzlich wird der Mode, in dem die SPI Schnittstelle betrieben wird, festgelegt (Zeile 12-13). Anschließend folgt die Konfiguration des AMC-Registers mit den im vorherigen Kapitel genannten Funktionen (Zeile 15-19). Diese Programmzeile können auskommentiert werden, wenn der AMC nicht angesprochen werden soll. In Zeile 20 ist die Funktion zur Ansteuerung der LDOs zu finden. Mit der Variablen „ldo“ wird der gewünschte LDO gewählt und mit der Variable „set“ wird der gewählte ldo über die Inverterschaltung eingeschaltet (set=0) und ausgeschaltet (set=1). Die Funktion set_all_gain() in Zeile 22 stellt den Messbereich für den ADC-Baustein ein, mit welchem die SPI-Kommunikation erfolgt, für alle Kanäle auf 0-5V fest. Schließlich werden einige Funktionen ausgeführt und deren Ergebnisse ausgegeben. In Zeile 24 wird der Kanal 5, welcher in der Variable „channel“ des AMC7823 gespeichert ist, ausgelesen und in der Variable „wert“ gespeichert. Dieser Wert kann mit einer printf()-Funktion ausgegeben werden. Mit den Funktionen set_voltage_opa() und set_voltage_ldo() in den Zeilen 27-28 wird die gewünschte Spannung an den LDOs und OPAs eingestellt, welche zuvor in der Variablen „voltage“ festgelegt worden ist. Die Funktion zur Auslesung des Kanal 5 des ADS8688-Baustein befindet sich in Zeile 30. Zuletzt wird am Ende der Main-Funktion die SPI-Schnittstelle mit close(fd) geschlossen.

Für die Korrektur der Messabweichung der analogen Kanälen des ADC müssen einige Funktionen zur Verbindung mit dem Netzteil und dem Multimeter eingebunden werden, welche von Mitarbeitern des Labors für integrierte Schaltungen entworfen worden sind. Aufgrund des Umfangs werden diese Funktionsbereiche nicht in der Thesis behandelt, jedoch als elektronischer Anhang der Thesis beigefügt. Dafür werden zwei Bereiche „1“ und „2“ in der Main()-Funktion erstellt, welche bei Bedarf genutzt werden, oder auskommentiert werden können. Zusätzlich muss beim erstellen der Messreihe am Ende der Main-Funktion die Datei „Offset.txt“ in welche die Werte der Messreihe gespeichert wurden mit dem Befehl „fclosefp()“ geschlossen werden.

```
4  int main(int argc, char *argv[])
5  {
6      int channel=5;
7      int opa=3,ldo=1,set=1;
8      int fd;
9      float wert;
10     float voltage=1;
11     fd = open("/dev/spidev2.0", O_RDWR); // o. device deklarieren
12     uint32_t mode = 0x0 | SPI_CPHA; //mode
13     int ret = ioctl(fd, SPI_IOC_WR_MODE32,&mode); //Mode CPL=0 CPHA=1;
14     // Konfiguration register amc7823
15     power_down_register(fd);
16     amc_status_config(fd);
17     dac_config(fd);
18     load_dac_register(fd);
19     adc_control_register(fd,channel);
20     gpio_control(fd,ldo,set);
21     // Setzen aller Kanäle auf Messbereich +5V
22     set_all_gain(fd,DC5V);
23     // Ablesen ADC_Kanal vom AMC7823 (5) und Speichern in wert
24     amc7823_read_channel(fd,channel,&wert);
25     printf("%f",wert); // Ausgabe Wert
26     //Setze Spannung an LDO und OPA
27     set_ldo_voltage( fd, ldo,voltage);
28     set_voltage_opa(fd,opa,voltage);
29     // Ablesen vom Kanal (5) des ADS8688
30     ads8688_read_channel(fd,channel,gain,&wert);
31     printf("%f",wert); // Ausgabe Wert
32     //lesen Strom an Kanal (5) und speichern in wert
33     read_current(fd,channel,&wert);
34     printf("wert ; %f \n ", wert); // Ausgabe Wert
35     close(fd);
36     return 0;
37 }
```

Listing 7: Main()-Funktion

9. Inbetriebnahme der ersten Leiterplattenversion

Bei der Inbetriebnahme der Frontendplatine wird sowohl die entwickelte Software, als auch die Platine selbst auf korrekte Funktion geprüft. Die Inbetriebnahme erfolgte in zwei Schritten, wobei im ersten Schritt nur Funktionen der Frontendplatine geprüft worden sind, die ohne Programmierung der Bauteile verwendet werden können. Bei der Inbetriebnahme wurden Fehler identifiziert, die in einer zweiten Version korrigiert worden sind.

Als erstes wurde die Oberseite der Frontend-Platine bestückt und mit einem Reflow-Verfahren gelötet. Die Stecker auf der Unterseite der Platine mussten manuell verlötet werden. Aufgrund der geringen Größe der Stecker-Pins konnte der Lötvorgang nur unter Zuhilfenahme eines Heißluftföhns durchgeführt werden, welcher jedoch einen der Stecker leicht verformte.

Da sich die Trägerplatine noch in Entwicklung befindet, musste anschließend eine provisorische Platine entworfen werden, mit welcher die Pins der Stecker zu Stiftleisten für die digitalen und analogen Signale geführt und somit einfacher abgegriffen werden können. Das entsprechende Layout ist in Abbildung 37 zu sehen. Auch diese beiden Platinen wurden mit den benötigten Steckern und Stiftleisten verlötet.

Nach der Zusammenführung der beiden „Steckerplatinen“ mit dem Analog- und Digitalstecker der Frontend-Platine wurde festgestellt, dass drei Pins des Digitalsteckers aufgrund der thermisch bedingten Verformung des Steckers nicht mit den Stiftleisten verbunden sind (Pin 1,3,5 unten links). Zwei Pins gehören zur 12V Spannungsversorgung und ein Pin entspricht dem DGND-Massepotential. Dies hatte jedoch für die Testung der Platine keine Konsequenzen, jedoch zeigt diese Problematik, dass bei der Verlotung des unterseitigen Steckers besondere Vorsicht geboten ist.

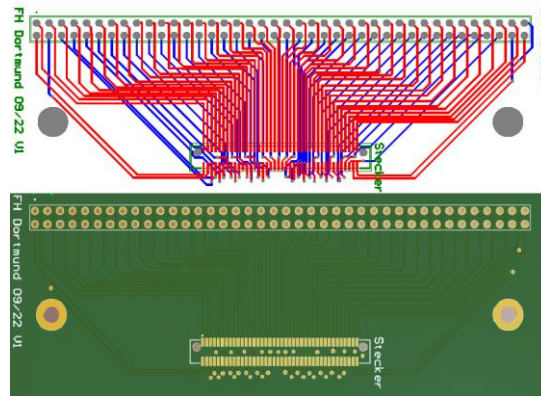


Abbildung 37: Layout Platine Steckleiste zum vereinfachten Abgriff von Signalen

Im weiteren Verlauf der Inbetriebnahme wurde festgestellt, dass die Anode- und Kathode der Diode am Tiefsetzsteller vertauscht wurden, wodurch die Eingangsspannung des Tiefsetzsteller nach Masse gezogen wurde. Die Diode konnte provisorisch gedreht werden um den Test fortzusetzen. Entsprechend wurde die Änderung im Schaltplan für der neuen Platine vorgenommen (Abbildung 38, roter Rahmen).

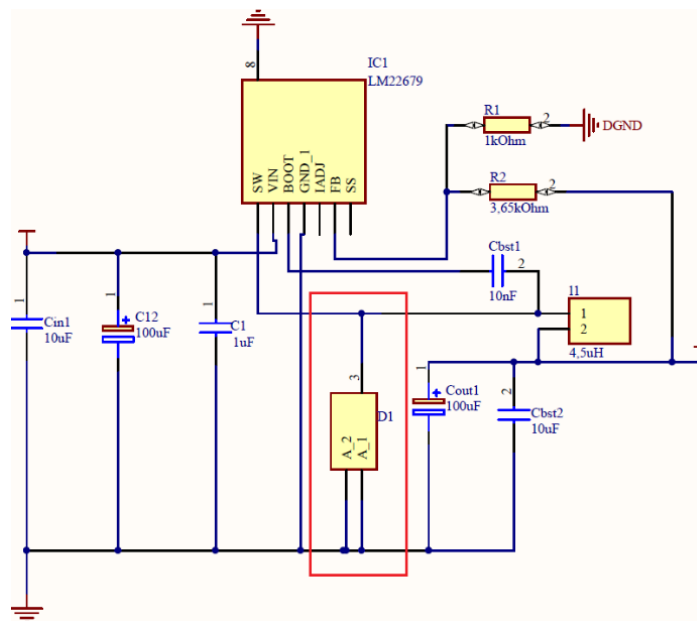


Abbildung 38: Korrektur der Beschaltung der Diode am Tiefsetzsteller

Des Weiteren stellte sich heraus, dass die SDO- und SDI-Leitungen der SPI-Schnittstelle des ADS8688-Bausteins vertauscht wurde. Ein softwareseitiger Tausch der Pinne ist nicht möglich, da der SDI-Eingang über den Pegelumsetzer für Inputs

und der SDO-Ausgang über den Pegelumsetzer für Outputs geführt werden muss. Daher musste eine manuelle Umlötung, erfolgen um die Software zur Ansteuerung der ICs testen zu können. Der ADS8688(1), welcher für die Messung von 8 Spannungen vorgesehen ist, wurde getestet, indem dem Bauteil von einem Netzteil 8 Spannungen zugeführt wurden und schließlich präzise gemessen werden konnten.

Anschließend wurde der ADS8688-Baustein(2) getestet, indem Ströme in die Signalpfade, die den 8 ADC-Kanälen zugeordnet sind, injiziert worden sind. Die 4 Instrumentenverstärker, welche für geringe Ströme bis 20mA ausgelegt worden sind, können nur Ströme bis maximal 10mA messen, während die 4 anderen Instrumentenverstärkern, welche für höhere Ströme bis 1A ausgelegt sind, nur Ströme bis ca. 406mA messen können. Die Instrumentenverstärker generieren am im Ruhezustand, also ohne eine messbare Spannungsdifferenz über den Shuntwiderständen eine Ausgangsspannung von 2,5V. Dadurch ist es möglich sowohl negative Ströme (0-2,5V) als auch positive Ströme (2,5-5V) zu messen. Dies wurde bei der Dimensionierung der Shuntwiderstände nicht berücksichtigt und wurde bei der zweiten Platine korrigiert. Gemäß der Formel zur Berechnung des Verstärkungsfaktors des Instrumentenverstärkers ergibt sich eine neue theoretische Verstärkung von 250 :

$$A = \frac{U_{max}}{U_{shunt}} = \frac{2,5V}{10mV} = 250 \quad 9.1$$

Mit der alten Berechnung und der Verstärkung von A=500 wäre trotz der falschen Annahme, dass der Ruhestrom des Instrumentenverstärkers 0V beträgt und nicht mit 2,5V angenommen wurde, ein maximaler Strom von 500mA zu erwarten gewesen. Dieser liegt jedoch bei etwa 400mA. Daher wird als neuer Verstärkungsfaktor nicht der neu errechnete Wert von A=250 sondern A=200 gewählt.

Der neue Wert für den Widerstand R2 ergibt sich gemäß Formel 9.2 zu:

$$R2 = A * \frac{R1}{2} = 100 * 3,92k\Omega = 392k\Omega \quad 9.2$$

Schließlich wurde der AMC7823-Baustein in Betrieb genommen. Die ADC-Kanäle konnten erfolgreich getestet werden, hatten jedoch Messabweichungen, welche zu korrigieren sind.

Die Operationsverstärker konnten erst angesteuert werden, nachdem die Beschaltung der Enable-Pins aufgetrennt wurde. Diese waren fälschlicherweise mit Masse verbunden, sollten jedoch unbeschaltet gelassen werden. Die zu erwartende Spannung konnte anschließend an den 4 Pins aller 4 OPAs erfolgreich gemessen werden.

Die LDOs, welche höhere Ströme bis 1A liefern können, sollten über einen GPIO des AMC7823-Bausteins an- und ausgeschaltet werden können. Dadurch, dass der Pull-Up Widerstand des GPIOs mit Masse statt mit einer 3,3V Spannung verbunden war, konnten die LDOs jedoch nicht eingeschaltet werden. Eine Spannung von 1V am Enable-Pin ist notwendig um den LDO anzuschalten. Daher wurden diese provisorisch direkt mit einer 5V-Spannung verbunden und konnten dadurch angeschaltet werden. Dabei wurde festgestellt, dass der volle Ausgangsspannungsbereich nicht realisiert werden konnte. Durch den Einbau eines Potentiometers an der Stelle des Vorwiderstandes (Abbildung 12, R8) in der Leitung des AMC zu den LDOs konnte die Auswirkung des Widerstandswertes betrachtet werden. Die Messung lieferte mit einem kleineren Widerstand von 8,62kOhm statt der verbauten 11,5kOhm bessere Ergebnisse. Aus diesem Grund wurde dieser Widerstand bei der zweiten Platinenversion vorgesehen. Die endgültige Platine wurde zudem um 4 GPIO Ausgänge erweitert, welche über einen Inverter mit den vier LDOs verbunden sind, damit diese einzeln angesteuert werden können (Abbildung 39). Der Inverter kann laut Datenblatt mit einer Spannung von 4,5V bis 5,5V versorgt werden. Daher wird die 5V Spannungsversorgung verwendet. Die Ansteuerung ist sinnvoll, da dadurch die LDOs erst nach der Einstellung der Spannung, welche der AMC vorgibt, angeschaltet werden können und somit ein Vollausschlag und die damit einhergehende Beschädigung von Verbrauchern verhindert werden kann. Indem das GPIO-Signal auf „0“ gesetzt wird, wird der EN-Pin des jeweiligen LDOs auf die Spannungsversorgung des Inverters gelegt und somit eingeschaltet. Entsprechend umgekehrt schaltet der Inverter den EN-

Pin des Inverter auf Masse, wenn ein High-Pegel des GPIOs erkannt wird. Es wird erwartet, dass ein Strom vom 10mA bis 20mA über die Pull-Up Widerstände R_{iv} fließen wird. Weswegen für R_{iv} nach den Formeln 9.3 und 9.4 Widerstandswerte zwischen 25kOhm und 50kOhm errechnet werden. In diesem Fall werden 33kOhm Widerstände verwendet.

$$R_{iv} = \frac{U}{I} = \frac{5V}{10mA} = 50kOhm \quad 9.3$$

$$R_{iv} = \frac{U}{I} = \frac{5V}{20mA} = 25kOhm \quad 9.4$$

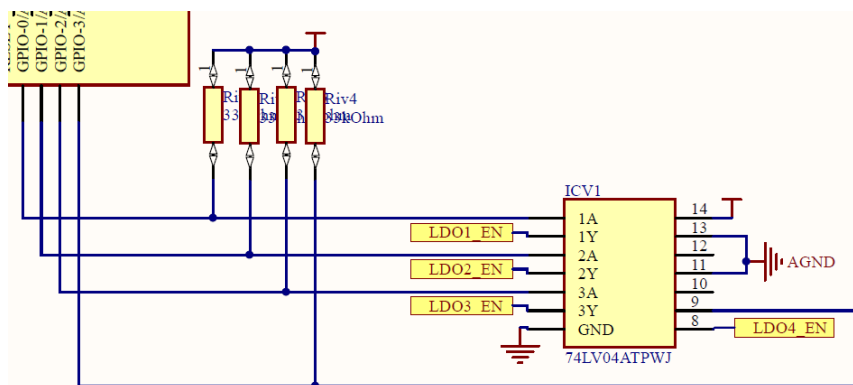


Abbildung 39: Ergänzung der GPIO-Ansteuerung in der Beschaltung des AMC7823-Bausteins

10. Inbetriebnahme der zweiten Leiterplattenversion

10.1 ADS8688 ADC Baustein (1)

Es wurde zunächst der ADC-Baustein (1) getestet, welcher 8 Spannungen messen kann. Vorgesehen ist ein Messbereich von 0-5V, allerdings wurde wie bereits erklärt die Möglichkeit geschaffen den Messbereich zu verändern. Mit dem gewünschten Messbereich „DC5V“ war das Verhalten des ADC1 auf allen Kanälen gleichermaßen nicht wie erwartet. Der Spannungsbereich von 2,6V-5V wurde korrekt gemessen. Der Spannungsbereich von 1,1V-2,6V konnte nur gemessen werden, nachdem bereits einmalig eine Spannung über 2,6V am Kanal anlag. Nach Überschreitung der Spannung von 2,6V kann der ADC zusätzlich den Spannungsbereich von 1,1V-5V korrekt messen. Den Bereich unter 1,1V konnte der ADC jedoch fehlerfrei auflösen. Durch die Änderung des Messbereiches in andere Spannungsbereiche wie etwa „AC5V“, welcher für den Messbereich -5V bis +5V ausgelegt ist, funktioniert der ADC jedoch fehlerfrei. Zwar steigt dadurch der Quantisierungsfehler, jedoch erreicht die Genauigkeit des ADCs Werte unter 1mV.

Eine weitere Platine gleichen Typs zeigte das gleiche Phänomen, wodurch ein Bestückungsfehler ausgeschlossen werden kann. Da der zweite ADC, dessen Kanäle mit den gleichen Funktionen ausgelesen wird, fehlerfrei funktioniert, können Softwarefehler auch ausgeschlossen werden. Bei Anwendung der aktuellen Software-Version auf die Platine der Version 1 mit der Verstärkung „DC5V“ ist die gleiche Einschränkung zu beobachten, was andererseits einen Software-Fehler wahrscheinlich macht, jedoch nicht endgültig geklärt werden kann. Im Vergleich zur ursprünglichen ersten Platinenversion wurden lediglich die SDO und SDI Leitungen vertauscht, was jedoch gleichermaßen sowohl beim ADC1 als auch beim ADC2 durchgeführt wurde. Die Einschränkung bei Messung im DC5V Messbereich betrifft jedoch nur den ADC1.

10.2 ADS8688 ADC Baustein (2) und Instrumentenverstärkerschaltungen

Der zweite ADC misst vier kleine (bis 20mA) und vier große (bis 1A) Ströme über Shuntwiderstände, deren Spannung über Instrumentenverstärker verstärkt werden. Bei einem Strom von 0A und damit einem Spannungsabfall von 0V über dem Shunt-Widerstand stellt sich eine Spannung von 2,5V am Ausgang des Instrumentenverstärkers und damit am ADC-Eingang ein. Injiziert man negative Ströme in den Shunt-Widerstand und erzeugt damit negative Spannungen, so reduziert sich die Ausgangsspannung des Instrumentenverstärkers. Nach diesem Prinzip wurde geprüft ob der ADC 2 mit dem voreingestellten Messbereich „DC5V“ funktioniert. Die Messung kleiner Ströme funktioniert fehlerfrei, wobei die Kanäle die Messung von Strömen bis 25mA ermöglichen. Der maximale Auslesewert des ADC-Kanals entspricht einer analogen Spannung von 5V, bei welcher eine Strömstärke von 25mA erreicht wird. Entsprechend muss die Formel zur Umrechnung der erfassten Spannung in den Strom angepasst werden. Drei der vier Kanäle für die Messung großer Ströme sind ungenau und der zweite Kanal kann nur Ströme bis maximal 700mA messen.

Ursächlich für die Einschränkung des Messbereiches ist ein Designfehler im Layout. Für die Veranschaulichung des Fehlers ist ein vereinfachtes Layout der Instrumentenverstärkerschaltung mit der Stromleitung (gelb bepunktet) und dem Shuntwiderstand (Hsht) in Abbildung 40 dargestellt. Der Spannungsabfall über dem Widerstand wird nicht direkt an den Enden des Widerstandes abgegriffen, wodurch sich die Spannungsdifferenz durch den Widerstand der Leiterbahn erhöht. Die Leiterbahnstücke, die zusätzlich zu berücksichtigen sind, sind in Abbildung 40 mit I1 und I2 grün beschriftet und bemaßt.

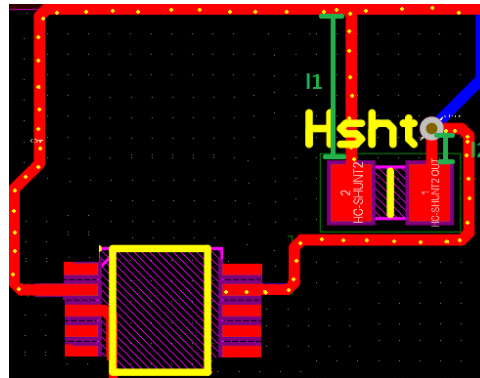


Abbildung 40: Spannungsabfallmessung über Shunt

Unter Berücksichtigung dieser Annahme wurde die Spannungsdifferenz U_{k2} am Kanal 2 bei einem anliegenden Strom von $I=600\text{mA}$ gemessen und ein Betrag von 10.66V ermittelt, so dass gilt:

$$U_{k2} = 10,66\text{mV} \quad 10.1$$

Mit Hilfe von Formel 10.2 errechnet sich nach dem Ohm'schen Gesetz ein Widerstand im Stromkreis von $17,76\text{m}\Omega$ und beträgt damit nahezu $8\text{m}\Omega$ mehr als gewünscht.

$$R_{k2} = \frac{U_{k2}}{I} = \frac{10,66\text{mV}}{600\text{mA}} = 17,76\text{m}\Omega \quad 10.2$$

Anschließend werden die überflüssigen Leiterlängen im Layout gemessen und der resultierende Leitungswiderstand (RL_{k1}) gemäß Formel 10.5 berechnet. Die Konstante „ p “ entspricht dabei dem spezifischen Kupferwiderstand. Die Variable „ l “ entspricht der gesamten Leiterlänge, welche sich aus den Leiterlängen l_1 ($3,7\text{mm}$) und l_2 ($0,8\text{mm}$) zusammensetzt. Der Querschnitt der Leitung ist „ A “ und berechnet sich aus der Breite ($b=0.3\text{mm}$) und Höhe ($h=35\mu\text{m}$) der Leiterbahn.

$$p = 0.017 * \frac{\Omega * \text{mm}^2}{\text{m}} \quad 10.3$$

$$A = b * h = 0,3\text{mm} * 0,035\text{mm} = 0,0105\text{mm}^2 \quad 10.4$$

$$RL_{k1} = \frac{p * l_{k1}}{A} = \frac{0.017 * \frac{\Omega * \text{mm}^2}{\text{m}} * 4.5\text{mm}}{0,0105\text{mm}^2} = 7,29\text{m}\Omega \quad 10.5$$

Der errechnete Widerstand beträgt $7,29\text{m}\Omega$ und gleicht in etwa dem gemessenen Wert, was die Annahme bezüglich des Layoutfehlers bestätigt. Der Fehler wurde im Layout korrigiert und unter der Version 3 gespeichert. Bei den Instrumentenverstärkerschaltungen für geringe Ströme hat der zusätzliche Leitungswiderstand im Milliohm-Bereich aufgrund des vergleichsweise großen Shuntwiderstandes von $0,5\Omega$ keinen Einfluss. Alle Kanäle haben leichte Messabweichungen im mA-Bereich, welche gleichbleibend sind.

10.3 Validierung des ADC/DAC-Bausteins

Der AMC wurde mit den Registerfunktionen konfiguriert und getestet. Die analogen Kanäle für die Strommessung haben eine schwankende Messungsgenauigkeit von 5-20mA, funktionieren ansonsten jedoch wunschgemäß.

Alle vier Operationsverstärker, über welche Spannungen von 0-5V und Ströme bis 20mA generiert werden sollten, liefern die gleichen Werte. Bestimmt durch die 5V Versorgungsspannung kann laut Datenblatt eine theoretisch mögliche Ausgangsspannung von 0.1-3.5V erreicht werden. Diese Angabe ist der „Common-Mode Voltage Range VCM (V-) – 0.1 (V+) – 1.5 V“ im Datenblatt zu entnehmen. In der Praxis lassen sich Spannungen von 20mV bis 4.3V einstellen. Daher ist der gewählte OPA für die gewünschte Ausgangsspannung von 5V nicht geeignet. Die OPAs müssten theoretisch mit 6,5V betrieben werden um eine Ausgangsspannung von 5V liefern zu können. Praktisch sollte die 6V Versorgung ausreichen, da durch die jetzige 5V-Versorgung bereits eine maximale Spannung von 4,3V erzielt wurde. Die Messwerte der OPAs sind bis 300mV auf ca. 5 mV genau. Die Messungsgenauigkeit sinkt anschließend bis auf 1mV bei 900mV und beträgt anschließend nahezu 0mV. Alle 4 LDOs, welche für die Generierung der Spannung 0.5-5V und 1A gedacht sind, funktionieren fehlerfrei und der angestrebte Messbereich wird erreicht. Unter 600mV sind alle Werte sehr ungenau. Insbesondere ohne Sensing entstehen Messabweichungen von bis zu 50mV. Mit Sensing reduzieren sich die Messabweichungen auf maximal 25mV.

10.4 Layout der zweiten Platinenversion

Das korrigierte Layout der Leiterplatte mit der Version 2 ist in Abbildung 41 dargestellt. Bei der Umsetzung wurde der Teil der leistungselektronischen Schaltung bestehend aus dem Tiefsetzsteller und den 1A-LDOs oben rechts positioniert. Es wurde versucht alle funktional zusammenhängende Bauteilgruppen in örtlicher Nähe zu positionieren. Außerdem wurde versucht, die Bauteile so anzuordnen, dass möglichst kurze Leitungen und wenig Vias notwendig sind. Der digitale Teil wurde im linken Bereich, der Platine platziert und der analoge Teil im rechten Bereich um eine Trennung der Massefläche zu ermöglichen.

Dementsprechend wurden auch die Stecker mit den digitalen und analogen Signalen jeweils links und rechts positioniert:

Position	Stecker	Position der Mitte
X-Achse/ Y-Achse	Digital	35/75 mm
X-Achse/ Y-Achse	Analog	115/75 mm

Tabelle 31: Position der Stecker an der Platinenunterseite

Die beiden Masseflächen, DGND für die digitalen und AGND für die analogen Masse-Anschlüsse wurden durch Polygonzüge links für DGND und rechts für AGND auf der Ober- und Unterseite erstellt und durch Stiching verbunden.

Für die mechanische Befestigung sind Gewindeschrauben vorgesehen, deren Bohrlöcherposition in der folgenden Tabelle angegeben wird:

	Oben links	Oben rechts	Unten links	Unten rechts
X/Y in mm	29/121	121/121	29/29	121/29

Tabelle 32: Bohrlöcherposition für Gewindeschrauben zur mechanischen Befestigung der Platine

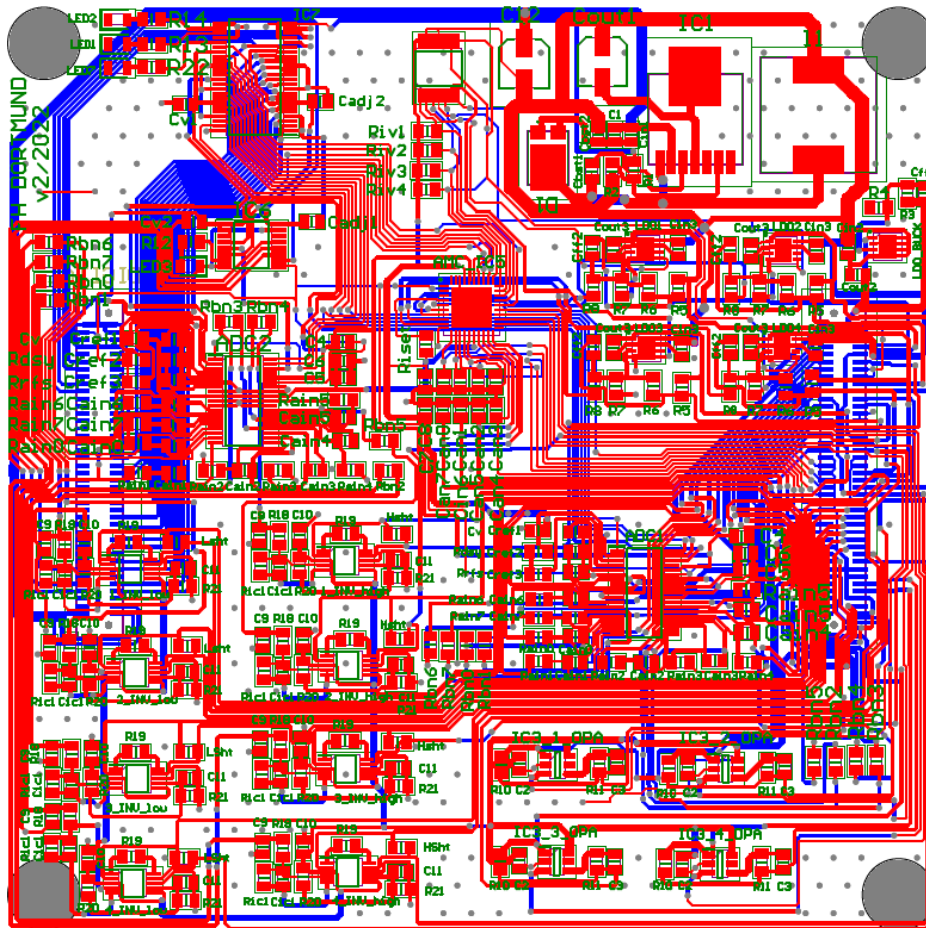


Abbildung 41: Layout der zweiten Leiterplattenversion

11. Messabweichung und Korrektur

11.1 Projektbezogene Grundlagen

Bereits während der Inbetriebnahme der ersten Version der Platine und Prüfung der Genauigkeit der Ergebnisse wurde, Messabweichungen festgestellt. Das betrifft die Strommessungen durch die Instrumentenverstärkerschaltungen am ADS8688(2), die vorgegebenen Ströme an den LDO- und OPA-Schaltungen am AMC7823-Baustein und die analoge Kanäle des AMC7823-Bausteins. Wie bereits in den Grundlagen der Messabweichungen erläutert, können Messungenauigkeiten eine Vielzahl von Gründen haben und unter Umständen sehr groß werden. Deswegen wurde eine Vorgehensweise entwickelt um die Messgenauigkeit zu erhöhen.

Bei den Instrumentenverstärkern, LDOs und OPAs konnten Messabweichungen in den Funktionen zur Auslesung der Werte korrigiert werden, indem ein zuvor ermittelter Offset-Wert abgezogen wird. Für die Ermittlung der Offset-Fehler wurden den Instrumentenverstärkern präzise Ströme zugeführt und die Messungenauigkeiten ermittelt. Bei den OPAs und LDOs wurden mithilfe der entwickelten Funktion gewünschte Spannungen eingestellt und die tatsächlichen Spannungen gemessen. Die Messabweichungen blieben konstant und konnten deswegen auch durch konstante Faktoren korrigiert werden. Die Ergebnisse der Messungen werden in den nächsten Unterkapiteln dargestellt.

Bei ADC-Kanälen des AMC ist die Fehlerkorrektur durch Offset-Kompensation nicht möglich, da die Werte stark schwanken. Für eine Korrektur der Messabweichungen wird zunächst ein Datensatz der Messungen benötigt. Dafür wurde eine fertige Funktion zur Ansteuerung eines Labornetzteils verwendet, welche in die Main-Funktion eingebunden wird. Das Netzteil wird so angesteuert, dass es Spannungen im Bereich von 0V-5.1V selbstständig in Schritten von 1mV bei einer Zeitverzögerung von 1ms generiert. Zusätzlich wird die Spannung des Netzteils mit einem sehr genauen Messgerät erfasst und als Referenzwert bei der ADC Charakterisierung verwendet. Zudem wird der dazugehörige ADC-Wert abgelesen. Pro Messung ergeben sich somit

drei Werte. Die Messungen wurden bis zu einer maximalen Spannung von 5.1V durchgeführt, da nicht genau bekannt ist, wie hoch die Messabweichung ist, wie ungenau das Netzteil ist und bei welcher Netzteileneinstellung sich die eigentlich gewünschte Spannung von 5V einstellt. Alle Werte werden in eine Textdatei mit drei Spalten gespeichert. Bei der Messung ergeben sich durch die eingestellte Schrittweite von 1mV für den Bereich von 0-5,1V 5100 Werte, wobei die 5V am ADC erkannt wird. Der 12-Bit-ADC hat einen Wertebereich von $2^{12}=4096$ Konvertierungsschritten. Bei $5,5V / 4096$ ergibt sich eine Schrittweite von ca. 1,3mV. Ein bereits bestehendes Matlab-Skript liest die erstellten Textdateien aller Kanäle ein und erstellt jeweils ein Array mit 4096 diskreten Werten, welche in eine Header-Datei eingefügt werden. Zudem wird jeweils ein Diagramm mit der Messabweichung vor der Korrektur und nach der Korrektur erstellt. Die Korrektur besteht darin, jedem der 4096 Werte den naheliegendsten Ist-Wert zuzuweisen.

Auch diese Ergebnisse werden in den nächsten Unterkapiteln veranschaulicht. Das Matlab-Skript wird aufgrund seiner Größe nicht in der Thesis eingefügt, jedoch vollständigshalber als Textdatei in elektronischer Form der Thesis beigefügt. Das Matlab-Skript benötigt zur Ausführung Informationen wie z.B die Bitzahl des Bauteils zur Berechnung der Größe des Arrays, den Wertebereich von 0-5V und den Namen der Textdatei, welche sich im gleichen Ordner wie das Matlab-Skript befindet muss (siehe im Quellcode Input Settings und Read File). Die Dateigrößen werden ebenfalls berechnet.

11.2 Messungenaugigkeit bei der Spannungsmessung des ADS8688-Bausteins (1)

Der ADC-Baustein, welcher für die Messung von acht Spannungen vorgesehen ist, hat eine sehr hohe Messgenauigkeit mit einer durchschnittlichen Messabweichung der IST-Werte von 0,4mV zu den SOLL-Werten. Daher wird an diesem Baustein keine Messkorrektur der Kanäle durchgeführt. Der Durchschnitt wird aus den Beträgen einzelner Messabweichungen gebildet, da ansonsten das Ergebnis verfälscht werden würde. Würde der Durchschnitt zweier Messabweichung von -1mV und 1mV gebildet, würde das 0mV betragen, was falsch ist, da der Durchschnitt 1mV beträgt. Die Abbildung 42 veranschaulicht die Messabweichung des Kanals 1, welcher alle Kanäle gleichermaßen repräsentiert. Der blaue Graph zeigt den Verlauf der Messabweichung über die auf der X-Achse dargestellte Eingangsspannung von 0-5000mV. Die Messabweichung ist zunächst positiv und wird mit zunehmender Spannung negativ. Insbesondere durch die geringe Skalierung auf der Y-Achse wird die kleine Messabweichung verdeutlicht

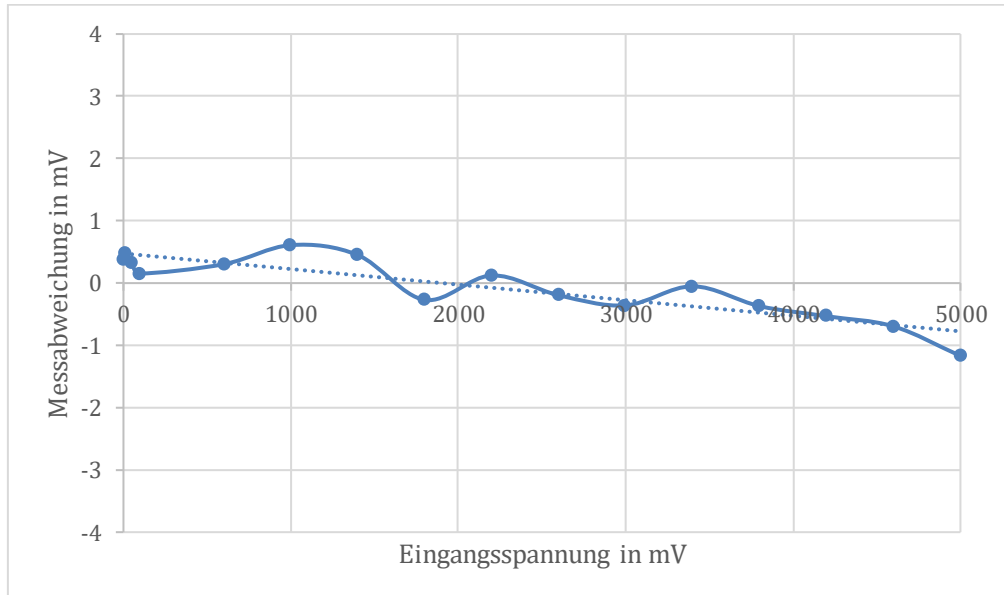


Abbildung 42: Messabweichung bei der Spannungsmessung an Kanal 1 am ADC-Baustein (1)

11.3 Messungenaugigkeit bei der Strommessung am ADS8688-Baustein (2)

Bei der Strommessung an den acht Kanälen des zweiten ADC-Bausteins entstehen ebenfalls kleine Messungenaugigkeiten, welche bei den Instrumentenverstärkerschaltungen für Ströme bis 20mA im Durchschnitt bei 0,4mA liegen und für Ströme bis 1A etwas unter 0,4mA betragen. Deshalb wurde auch bei diesen Kanälen keine Messkorrektur der Kanäle durchgeführt.

Die Abbildung 42 veranschaulicht die Messabweichung des vierten analogen Kanals. Insbesondere verdeutlicht die Durchschnittslinie die geringe Messabweichung. Alle Kanäle können bis 25mA messen, weshalb die Messreihe bis 25mA aufgenommen wurde, wobei die Messung ab 24mA deutlich ungenauer wird, was auch in Abbildung 43 am ende der X-Achse zu erkennen ist. Für höhere Ströme empfehlen sich die Kanäle, welche für Messungen bis 1A ausgelegt sind.

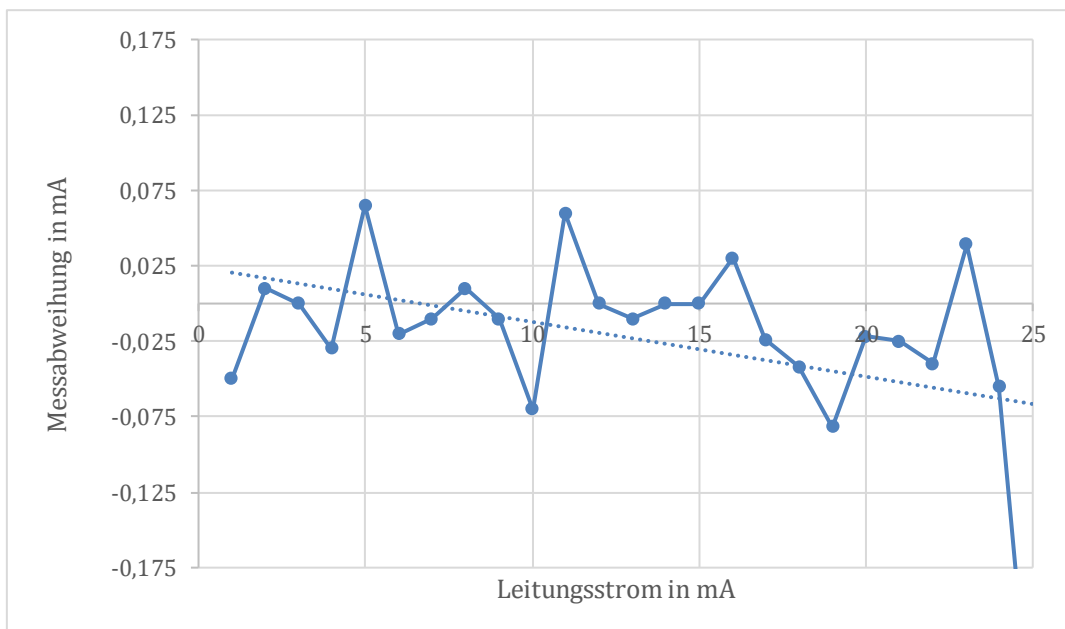


Abbildung 43: Messabweichung bei der Strommessung des vierten Kanals (bis 20mA)

Abbildung 44 zeigt die Messabweichung am Kanal vier. Der Ausschlag des Graphen am Anfang der Messung in negative Richtung hängt damit zusammen, dass die Messreihe ab 1mA geführt wurde und die Kanäle aufgrund der höheren Auslegung bis 1A bei kleinen Strömen ungenau sind. Die Werte schwanken innerhalb eines geringen Bereiches.

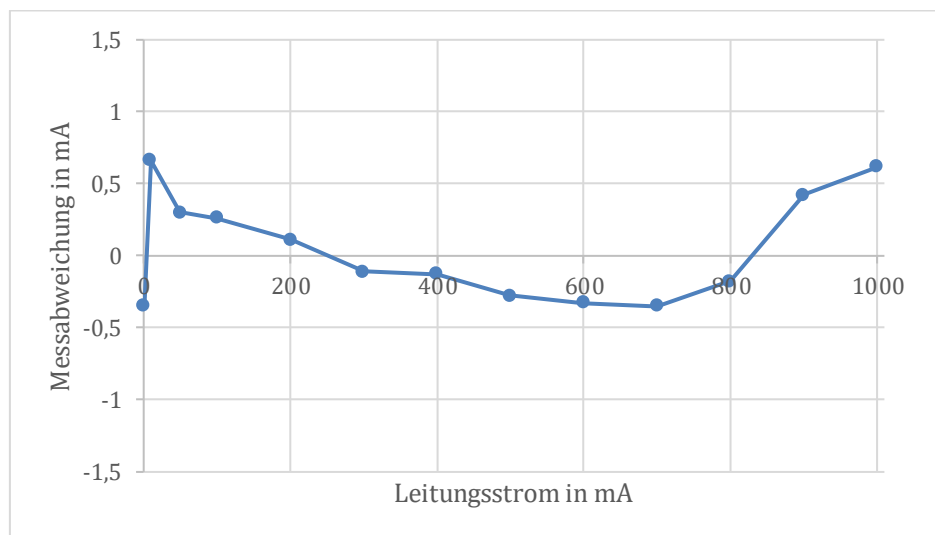


Abbildung 44: Messabweichung bei der Strommessung des vierten Kanals (bis 1000mA)

11.4 Messabweichung am AMC7823-Baustein

Der ADC/DAC-Baustein steuert über seine acht DAC-Kanäle die Ausgangsspannungen der vier Operationsverstärker und vier LDOs. Zudem verfügt er über acht analoge Kanäle. Alle Operationsverstärker sind in ihrem Verhalten nahezu gleich und verursachen ähnliche Messungenauigkeiten, wobei ein unbekannter Anteil der Messungenauigkeit durch den AMC-Baustein selbst verursacht wird. Für die Veranschaulichung der Messabweichungen dient Abbildung 45. Der Operationsverstärker 3 hat eine durchschnittliche Messabweichung von 3,94mV, welche durch den blauen Graphen dargestellt ist. Aufgrund der Bauweise beträgt der theoretische Ausgangsspannungsbereich aller Operationsverstärker 0,1-3,5V. In der Praxis lässt sich der Spannungsbereich 0-4,3V erzielen, welcher im Diagramm an der X-Achse dargestellt ist. Nachdem der Offset innerhalb des Quellcodes abgezogen wird, stellt sich eine durchschnittliche Messabweichung von 1,78mV ein. Dieser ist durch den roten Graphen, welcher die Messdaten nach der Offsetkorrektur beinhaltet gut nachvollziehbar.

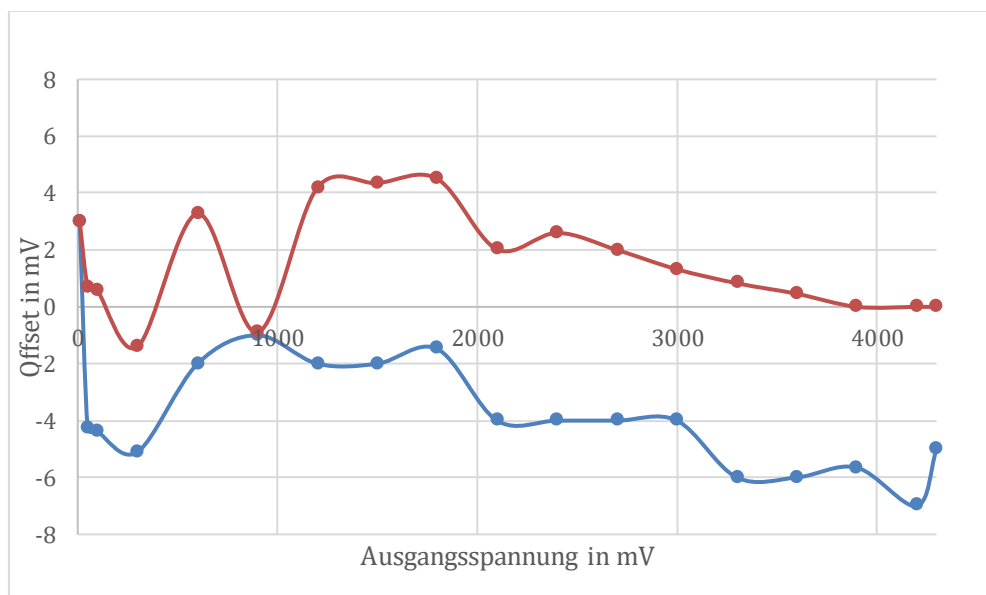


Abbildung 45: Messabweichung bei der Spannungsgenerierung am OPA (3) vor und nach der Korrektur

Abbildung 46 zeigt die Messabweichungen von LDO 1 des AMC7823-Bausteins vor und nach der Messabweichungskorrektur. Alle vier LDOs besitzen einen theoretischen Ausgangsspannungsbereich von 0,5-5V. In der Praxis wurde festgestellt, dass die Messungsgenauigkeit bis zu einer LDO-Ausgangsspannung von 600mV sehr hoch ist, was auch in der Abbildung zu sehen ist. Die Messungsgenauigkeit des verwendeten LDOs (blauer Graph) liegt bei etwa 21mV und ist damit deutlich höher als die der Operationsverstärker. Die Korrektur der Messabweichungen in der Software führt zu einer durchschnittlichen Messabweichung von etwa 3mV, welche durch den roten Graphen dargestellt wird.

In der Software kann jeder Kanal einzeln angepasst werden. Wenn die Messabweichungen nicht über den gesamten Messbereich konstant sind, können die gewünschten Messbereiche, welche sich unterscheiden, aufgetrennt und separat korrigiert werden. Wenn die anschließend resultierende Messgenauigkeit nicht genau genug ist, kann eine präzisere Offsetkorrektur die Messabweichung senken, indem mehr Messdaten generiert werden und mehr Messbereiche aufgetrennt und angepasst werden.

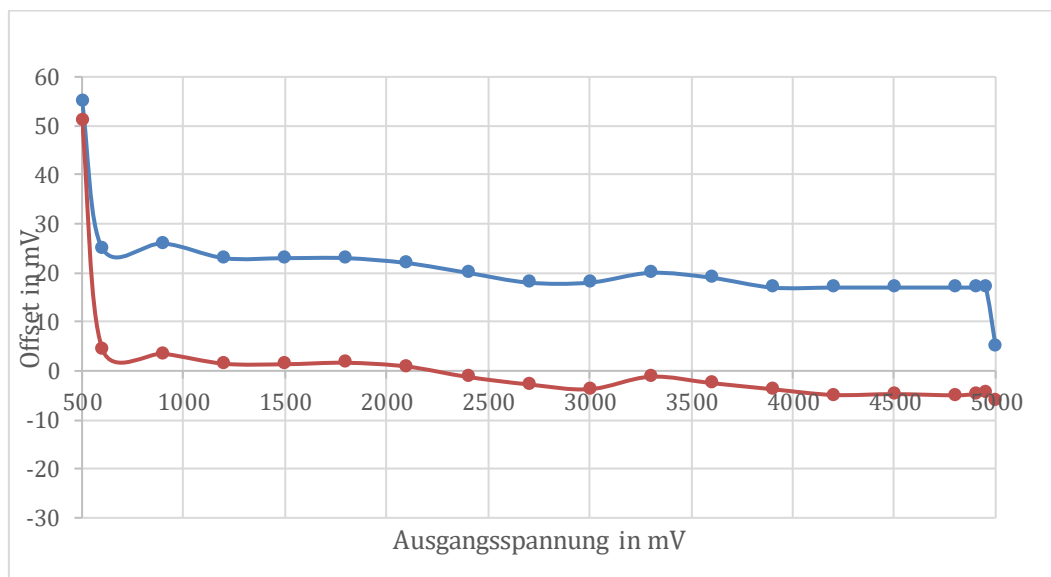


Abbildung 46: Messabweichungen bei der Spannungsgenerierung am LDO(1)

Im Folgenden soll die Korrekturprozedur am ADC Kanal 6 des AMC7823-Bausteins gezeigt werden. Das Diagramm in Abbildung 47 zeigt die Messabweichung vor und nach der Korrektur. Dieses wurde mit einem Matlab-Skript erzeugt, welche den Datensatz aus einer zuvor durchgeführten Messreihe am Kanal 6 nutzt und korrigierte Werte zur Verfügung stellt. Auf der Y-Achse ist die Höhe des Offsets mit einer Skalierung von +30mV bis -40mV dargestellt. Auf der X-Achse sind 4096 Werte des 12-Bit Wandlers dargestellt, welche 0-5V repräsentieren. Der violette Graph stellt die Messabweichung vor der Korrektur dar. Am Anfang der Messung stellt sich eine Messabweichung von +15mV bis -15mV ein, welche sich mit zunehmender Eingangsspannung in die negative Richtung bewegt und schließlich zu einer Messabweichung bis -35mV führt. Der blaue Graph zeigt die Messabweichung nach der Korrektur. Der Fehler bleibt bei -15mV bis +15mV, ohne sich bei höheren Spannungen in negativer Richtung zu erhöhen.

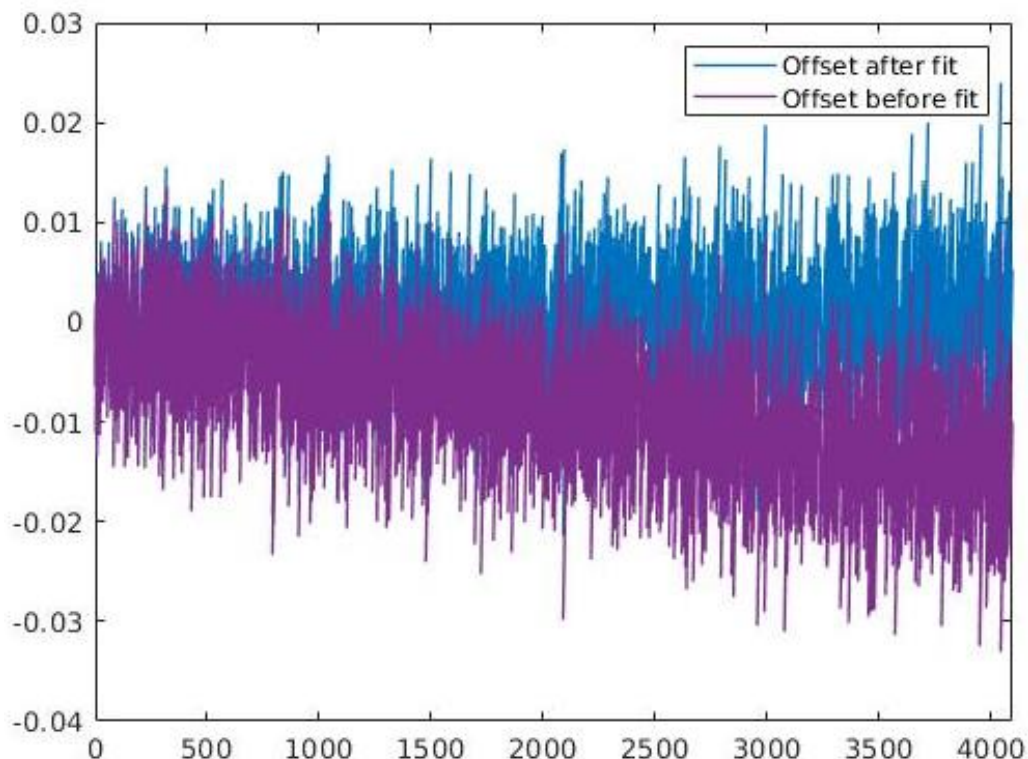


Abbildung 47: Messabweichung Daten vor Korrektur (lila) und nach Korrektur (blau)

Bei einer erneuten Messreihe am Kanal 6 wurde im Vergleich zur Messung aus Abbildung 47 nur der Wert des ADC-Kanals aus dem Mittelwert von 10 Messungen ermittelt um den hohen Schwankungen des Bausteins entgegenzuwirken. Das dazugehörige Diagramm ist in Abbildung 48 dargestellt. Vernachlässigt man den Peak bei X-Wert von 2600, würde eine Skalierung von 20mV bis -20mV an der Y-Achse reichen um den Offset darzustellen. Bereits beim unkorrigiertem Graphen (violett) sind die deutlich kleinerem Messunterschiede zu erkennen, obwohl sich der Offset wie bereits in Abbildung 47 zu höheren Spannungen in negative Richtung erhöht.

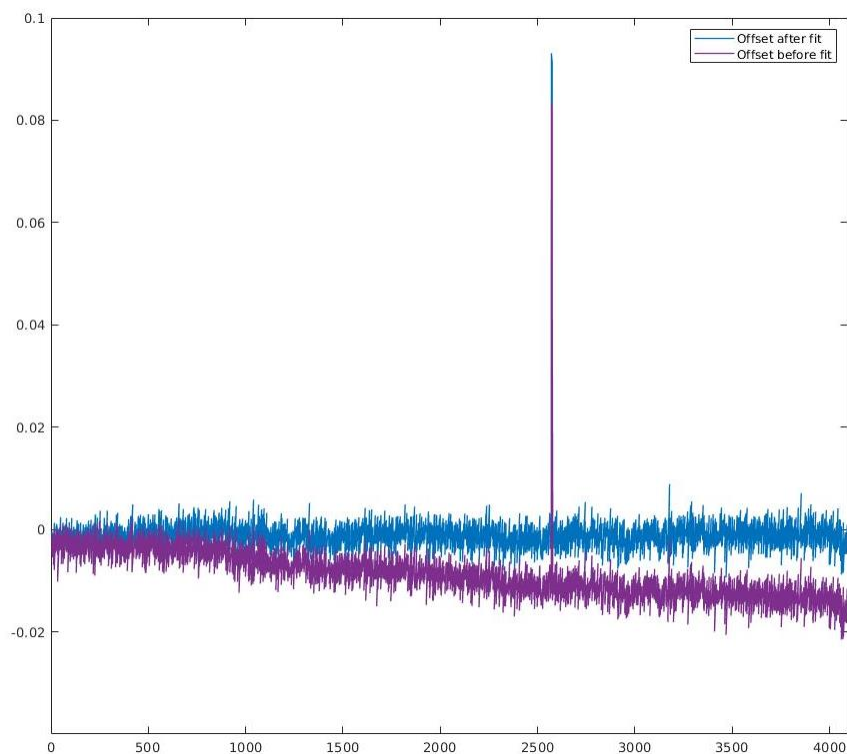


Abbildung 48: Messabweichung des ADC Kanal 6 des AMC7823-Bausteins mit Mittelwerten

Bei einer weiteren Messreihe wurde im Vergleich zur Messung aus Abbildung 48 die Geschwindigkeit der Messung von 1ms pro Messwert auf 2ms pro Messwert verringert. Abbildung 44 zeigt die Messabweichung des Kanals 6 mit den gemittelten ADC-Werten bei verlangsamer Messung. Betrachtet man den Graphen der Messabweichungen vor der Korrektur, fallen Messabweichungen von zunächst 5mV bis -5mV auf, welche sich ebenfalls mit höherer Spannung in negative Richtung bis zu einer Messabweichung von maximal -18mV bewegen. Insbesondere die korrigierten Werte sind mit einem Offset von +5mV bis -5mV im Vergleich zur Abbildung 48 sehr zufriedenstellend. Daher wurde diese Geschwindigkeit für alle weiteren Messreihen beibehalten .

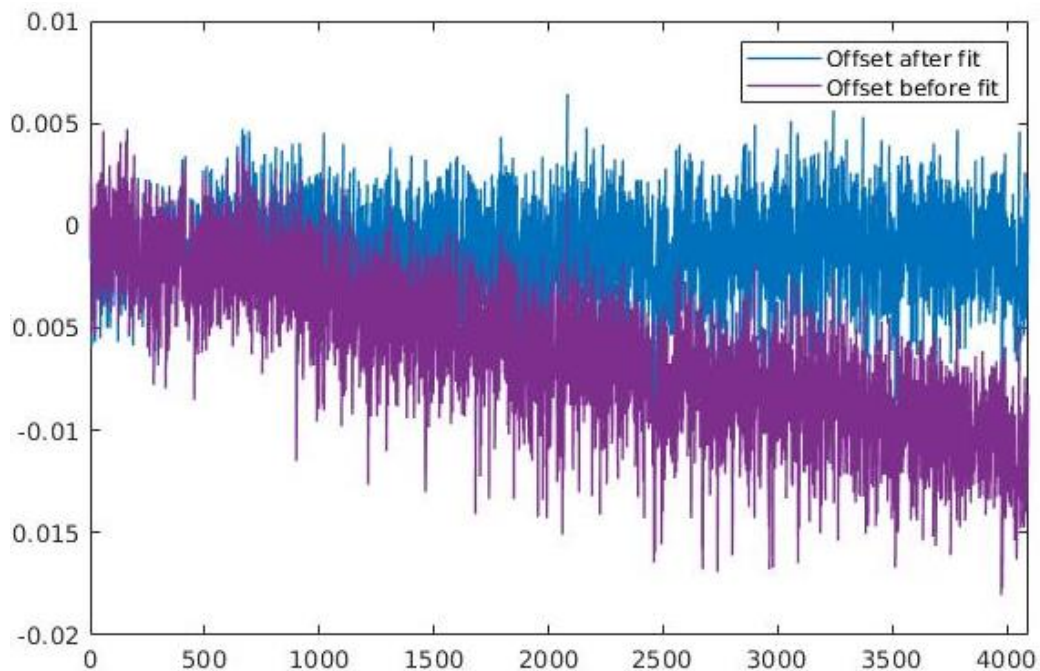


Abbildung 49: Kanal 6 mit gemitteltem ADC-Wert und verringerter Messgeschwindigkeit

Eine abschließende Messreihe wurde durchgeführt, welche die korrigierten Werte nutzt und eine zweite Korrektur durchführt, um zu prüfen ob die Messung dadurch genauer wird. Dabei werden auch in diesem Fall die ADC-Werte gemittelt und die reduzierte Messgeschwindigkeit beibehalten.

In Abbildung 50 ist das entsprechende Diagramm dargestellt. Der violette Graph zeigt den Offset vor der zweiten Korrektur und der blaue Graph zeigt den Offset nach der zweiten Korrektur. Der Offset beträgt beim unkorrigierten Graphen zwischen -8mV bis 8mV und beim korrigierten Graphen 4mV bis -10mV . Eine erneute Offsetkorrektur führt daher zu keinem kleineren Offset. Die Mittelung der ADC-Werte in Kombination mit einer geringen Messgeschwindigkeit und einer einzigen Offsetkorrektur reicht demnach für zufriedenstellende Ergebnisse aus.

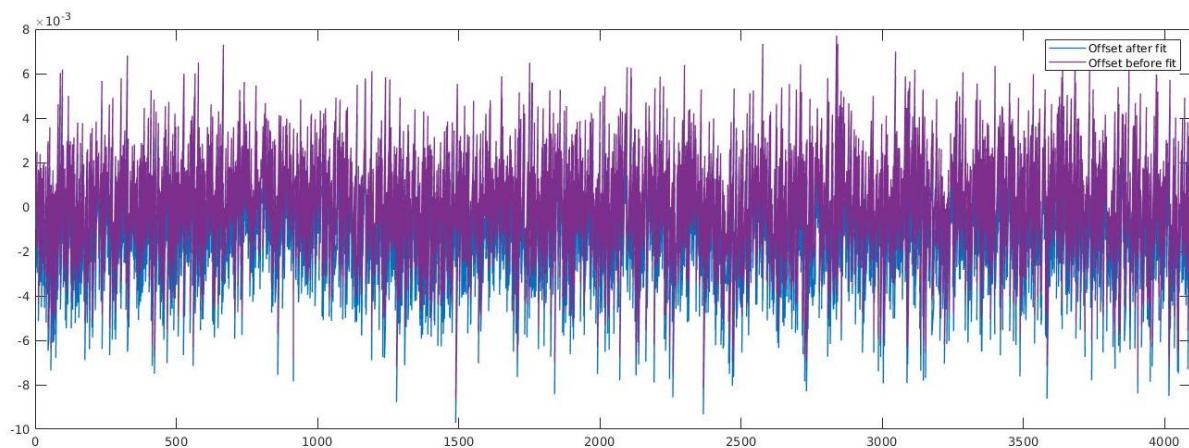


Abbildung 50: Kanal 6 mit gemittelten ADC- Werten und 2-facher Korrektur

10. Fazit

Die Aufgabe, welche im Rahmen der Masterthesis bearbeitet worden ist, bestand darin, die Software für einen ARM-Prozessor in einem ZYNQ-Baustein von Xilinx zu entwerfen. Diese Software soll zur Ansteuerung einer bestehenden Frontend-Platine für ein Testsystem zur Validierung von integrierten Halbleitern nach automotive Temperaturstandard verwendet werden. Die Frontend-Platine besitzt unterschiedliche Ein- und Ausgänge zur Spannungs- und Stromüberwachung, aber auch zur Versorgung der integrierten Halbleiterschaltungen.

Eine Herausforderung bestand zunächst darin, die unterschiedlichen Fehler der vorliegenden Platine zu identifizieren. Eine weitere Herausforderung bestand darin, sich in die Ansteuerung der Bauteile über eine Vielzahl an Befehlsformaten und Registerstrukturen einzuarbeiten. Die Anforderungen an die Software wurden erfüllt, zudem wurde eine zweite Version der Platine mit einer Korrektur der entdeckten Fehler entwickelt. Diese Platine funktioniert zufriedenstellend.

Die Projekterstellung in Vitis wurde dokumentiert, um anderen Studenten den Einstieg zu erleichtern. Auch wenn das entworfene Testsystem nicht alle möglichen und nötigen Testszenarien für die Validierung von Automobilelektronik abdecken kann, so ist trotzdem ein wesentlicher Beitrag geleistet worden, um Tests nach Automotive Standard im Labor für integrierten Schaltungsentwurf an der Fachhochschule Dortmund durchführen zu können.

Da auf Grund der Ressourcenknappheit, der Umweltbelastung durch den Kraftverkehr und der zunehmenden Elektromobilität ein großer Bedarf nach innovativer Elektronik vorhanden ist, kann das Testsystem von großem Nutzen bei zukünftigen Forschungs- und Entwicklungsprojekten sein, welche die Steigerung der Ressourceneffizienz, des Komforts und der Sicherheit von Kraftfahrzeugen zum Ziel haben.

Abbildungsverzeichnis

Abbildung 1: Blockdiagramm der Mess- und Versorgungselektronik [2]	8
Abbildung 2: SPI- Kommunikation im Mode0 und Mode1 [5]	10
Abbildung 3: ThermoStream-Aufbau.....	12
Abbildung 4: DNL A/D-Wandlung [7]	16
Abbildung 5: DNL-D/A-Wandlung [7].....	16
Abbildung 6: INL A/D-Wandlung[7]	17
Abbildung 7: INL D/A-Wandlung[7]	17
Abbildung 8: Beschaltung des Tiefsetzstellers.....	19
Abbildung 9: Vereinfachtes Schaltbild vom LDO mit fester Ausgangsspannung [8].....	20
Abbildung 10: Beschaltung des LDOs mit fester Ausgangsspannung.....	21
Abbildung 11: Spannungsregelung- und Überwachung der vier LDOs mit variabler Ausgangsspannung [9].....	23
Abbildung 12: Beschaltung der vier LDOs mit variabler Ausgangsspannung	23
Abbildung 13: Spannungsteiler für LDOs mit variabler Ausgangsspannung	24
Abbildung 14: Vereinfachte Voltage-Buffer Schaltung [10].....	27
Abbildung 15: Beschaltung der OPAs.....	29
Abbildung 16: Schaltbild des AD8553-Bausteins aus dem Datenblatt [12]	30
Abbildung 17: Instrumentenverstärker Schaltung	31
Abbildung 18: Beschaltung des Pegelumsetzers (16-Bit) für Digitale Inputs	33
Abbildung 19: Beschaltung des Pegelumsetzers (4-Bit) für digitale Outputs	33
Abbildung 20: Beschaltung des ADS8688 ADC -Bausteins.....	35
Abbildung 21: Beschaltung des AMC7823 ADC/DAC-Bausteins.....	37
Abbildung 22: Schreibprozess ins Programmregister des ADS8688[4].....	38
Abbildung 23: Programm Register Leseprozess [4].....	39
Abbildung 24: Zustandsdiagramm des ADS8688 Bausteins [4]	45
Abbildung 25: Ausführung des Manual Channel Modes des ADS8688 ADC-Bausteins [4]	48
Abbildung 26: Verlauf Lese/Schreib-Vorgang im Manual Channel Mode[4]	48
Abbildung 27: Memory Map des AMC7823 ADC/DAC-Bausteins [3]	50
Abbildung 28: Schreib -und Leseprozess des AMC7823 ADC/DAC Bausteins [3]	51
Abbildung 29: Zedboard [17].....	60
Abbildung 30: Maske zur Eintragung des Projektnamens	62
Abbildung 31: Eingabe des Pfads vom XSA-File	63
Abbildung 32: Einbindung der FPGA-Konfigurationsverzeichnisse	64
Abbildung 33: Maske zur Eingabe des Applikationsprojektnamens.....	64
Abbildung 34: Einbindung weiterer Konfigurationsdateien	65
Abbildung 35: New Target Connection with Zedboard	66
Abbildung 36: Auswahl des Targetnamens	66
Abbildung 37: Layout Platine Steckleiste zum vereinfachten Abgriff von Signalen.....	85
Abbildung 38: Korrektur der Beschaltung der Diode am Tiefsetzsteller.....	85
Abbildung 39: Ergänzung der GPIO-Ansteuerung in der Beschaltung des AMC7823-Bausteins	88
Abbildung 40: Spannungsabfallmessung über Shunt.....	91
Abbildung 41: Layout der zweiten Leiterplattenversion	94
Abbildung 42: Messabweichung bei der Spannungsmessung an Kanal 1 am ADC-Baustein (1)	97
Abbildung 43: Messabweichung bei der Strommessung des vierten Kanals (bis 20mA)	98
Abbildung 44: Messabweichung bei der Strommessung des vierten Kanals (bis 1000mA)	99
Abbildung 45: Messabweichung bei der Spannungsgenerierung am OPA (3) vor und nach der Korrektur	100
Abbildung 46: Messabweichungen bei der Spannungsgenerierung am LDO(1)	101
Abbildung 47: Messabweichung Daten vor Korrektur (lila) und nach Korrektur (blau)	102
Abbildung 48: Messabweichung des ADC Kanal 6 des AMC7823-Bausteins mit Mittelwerten.....	103
Abbildung 49: Kanal 6 mit gemitteltem ADC-Wert und verringerter Messgeschwindigkeit.....	104
Abbildung 50: Kanal 6 mit gemittelten ADC- Werten und 2-facher Korrektur.....	105

Digitaler Anhang auf der angehängten CD/USB-Stick

Quellcodes zur Ansteuerung der ADC- und ADC/DAC-Bausteine

Quellen als PDF-Dateien

Altium-Dateien inkl. PDF mit Schaltungen der Komponenten

Datenblätter der verwendeten Bauteile

Masterthesis in digitaler Form

Quellenverzeichnis

- [1] Scheuffele, J. (2003). *Automotive Software Engineering*.
- [2] Kampkötter, J. (2021). *Analogboard Präsentation*.
- [3] AMC7823-Datenblatt:
https://www.ti.com/lit/ds/symlink/amc7823.pdf?ts=1672592903917&ref_url=https%253A%252F%252Fwww.google.de%252F (Zugriff: 01.01.2022)
- [4] TI. (31. 12 2022). Datenblatt ADC. <https://www.ti.com/lit/ds/symlink/ads8688.pdf>.
- [5] Wikipedia [01.01.2023], https://en.wikipedia.org/wiki/Serial_Peripheral_Interface
- [6] AEC. (04. 08 2022). AEC.
http://www.aecouncil.com/Documents/AEC_Q100_Rev_H_Base_Document.pdf
- [7] Elektronik for Embedded Systems (01.01.2023)
http://www.controllersandpcs.de/lehrarchiv/pdfs/elektronik/ees05_02.pdf
- [8] Datenblatt LDO: https://www.ti.com/lit/ds/symlink/tps746-q1.pdf?ts=1674544286399&ref_url=https%253A%252F%252Fwww.ti.com%252Fpower-management%252Flinear-regulators-ldo%252Fproducts.html
- [9] Poole, M. (2016). https://e2e.ti.com/blogs_/archives/b/precisionhub/posts/how-to-build-a-monitor-and-control-solution-for-voltage-regulators.
- [10] Robbins, M. F. (31. 12 2022). <https://ultimateelectronicsbook.com/op-amp-voltage-buffer/>.
- [11] Leibson, S. (2018). <https://www.digikey.de/de/articles/fundamentals-of-current-measurement-part-1-current-sense-resistors>.
- [12] <https://www.mouser.de/datasheet/2/609/AD8553-1502247.pdf> (31.12.2022)
- [13] Neufang, O. (2013). *Lexikon der Elektrotechnik*.
https://www.google.de/books/edition/Lexikon_der_Elektronik/QBwmBgAAQBAJ?hl=de&gbpv=1&dq=pegelumsetzer&pg=PA369&printsec=frontcover
- [14] Andrews, R. (2015). https://e2e.ti.com/blogs_/archives/b/precisionhub/posts/three-guidelines-for-designing-anti-aliasing-filters (Zugriff: 31.12.2022)
- [15] <https://www.ti.com/lit/zip/tidrsj3> (Zugriff: 31.12.2022)
- [16] AMC Evaluation Module: <https://www.ti.com/lit/pdf/slau171> (Zugriff: 31.12.2022)
- [17] <https://shop.trenz-electronic.de/de/24539-ZedBoard-Zynq-7000-ARM/FPGA-SoC-Development-Board-Akademisch> (Zugriff: 31.01.2022)

Selbstständigkeitserklärung

Hiermit versichere ich, dass die von mir vorgelegte Prüfungsleistung selbständig und ohne unzulässige fremde Hilfe erstellt worden ist. Alle verwendeten Quellen sind in der Arbeit so aufgeführt, dass Art und Umfang der Verwendung nachvollziehbar sind.



Dortmund, 29.01.2023 , Unterschrift : Edis Salkovic