
Hardware-beschleunigte Drehmoment-regelung Ultraschallmotors

eines adaptiven
Piezo-

Diplomarbeit von Paul Hildebrand, TU Darmstadt
Dezember 2013



TECHNISCHE
UNIVERSITÄT
DARMSTADT



Embedded Systems & Applications

Paul Hildebrand
Matrikelnummer: 1314070
Studiengang: Automatisierungstechnik

Diplomarbeit:
Hardware-beschleunigte adaptive Drehmomentregelung eines Piezo-
Ultraschallmotors
Eingereicht: 20. Dezember 2013

Prüfer: Prof. Dr.-Ing. Andreas Koch
Betreuer: Dipl.-Inform. Andreas Engel

FG Eingebettete Systeme und ihre Anwendungen
Fachbereich Informatik
Technische Universität Darmstadt
Hochschulstr. 10
64289 Darmstadt

Co-Betreuer: Dr. rer. nat. Peter P. Pott

Institut für Elektromechanische Konstruktion
Fachbereich Elektro- und Informationstechnik
Technische Universität Darmstadt
Merckstr. 25
64283 Darmstadt

Ehrenwörtliche Erklärung

Hiermit versichere ich, die vorliegende Diplomarbeit ohne Hilfe Dritter nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die den Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Darmstadt, 20. Dezember 2013

Paul Hildebrand

Kurzfassung

Im Rahmen der vorliegenden Arbeit wurde eine Platine mit einem FPGA als Reglerplattform entwickelt. Diese Platine soll zwei bereits bestehende voneinander unabhängige analoge Platinen (CONBOX und Signalkonditionierungsplatine) ansteuern. Für die CONBOX sollen Test-Module implementiert werden. Diese sind aus Zeitgründen entfallen.

Die zu entwerfende Platine beinhaltet neben dem FPGA die Spannungsversorgung, einen AD-Wandler, mehrere Speicher, einen UART zu USB Wandler, einen Voltage Monitor, einen Oszillator, mehrere LEDs und Schalter zur Benutzerinteraktion sowie Steckverbinder, um die Verbindung zu den analogen Platinen herzustellen. Gefertigt und bestückt wurde sie von einem kommerziellen Anbieter.

Die Signalkonditionierungsplatine wird im Zusammenhang mit einem Piezomotor verwendet. Als Anwendung für die Signalkonditionierungsplatine und den Piezomotor sollen VHDL-Module implementiert werden, die eine virtuelle Rasterung simulieren. Das anliegende Drehmoment soll über eine Messbrücke auf der Signalkonditionierungsplatine ermittelt und eine Drehmomentregelung für eine virtuelle Rasterung implementiert werden. Mit der Messbrücke war das Messen der proportional zum Drehmoment ausgegebenen Spannung nicht möglich. Die Ermittlung des Drehmoments erfolgte daraufhin mit einem externen Drehmomentsensor und es wurde festgestellt, dass das vom Piezomotor erzeugte Drehmoment nicht proportional zu der angelegten Motorspannung ist. Stattdessen wurde als einfachere Alternative eine Drehmomentsteuerung implementiert.

Als weitere Anwendung wurde eine Drehzahlregelung implementiert. Drei verschiedene Regler wurden implementiert, wobei lediglich zwei getestet wurden.

Zur Überprüfung der Wirksamkeit der Regler wurden Messungen durchgeführt. Der Piezomotor wurde mit einem Bremsmoment beaufschlagt und die Regelabweichung ermittelt. Sie betrug beim reinen P-Regler 62,5% bei einer Sollzahl von 108 rpm. Der P-Regler wurde modifiziert indem die Stellgröße mit einem Offset addiert wurde. Der Offset beträgt die Motorspannung, bei der der Motor mit der Sollzahl dreht. Außerdem wurde die Stellgrößenbeschränkung modifiziert. Die Regelabweichung bei der Sollzahl 108 rpm beträgt lediglich noch 18 %.

Die Parametrierung des PI-Reglers ist aus Zeitgründen entfallen.

Inhaltsverzeichnis

1	Einleitung.....	1
2	Klärung der Aufgabenstellung.....	5
2.1	PCB-Entwurf einer FPGA-basierten Reglerplattform.....	5
2.2	Hardware-Beschreibung der Regleranwendungen	7
3	Verwandte Arbeiten	9
4	Aufbau der Leiterplatte.....	11
4.1	FPGA	11
4.2	Spannungsversorgung	13
4.3	AD-Wandler.....	15
4.4	Speicher	17
4.5	USB.....	17
4.6	Voltage Monitor / Power on Reset.....	18
4.7	Oszillator	19
4.8	Steckverbinder, LEDs, Taster	21
5	FPGA Hardware-Design	24
5.1	Haptisches Feedback.....	24
5.2	Testmodule für CONBOX.....	39
6	Ergebnisse und Diskussion der Arbeit	41
6.1	Inbetriebnahme	41
6.2	Messungen mit Drehmomentsensor.....	45
6.3	Steuerung für virtuelle Rasterung.....	47
6.4	Messungen mit Drehzahlregelung.....	48
6.5	Ergebnisse der Hardware-Synthese.....	57
7	Zusammenfassung und Ausblick	59
7.1	Platinen	59
7.2	FPGA Hardware-Design.....	60
A	Literaturverzeichnis.....	62
B	Schaltplan und Platinenlayout.....	63
C	Steckerbelegungen/Positionen	71
D	Pinbelegung FPGA	76

E	Messungen Drehzahlregelung	79
F	Abbildungsverzeichnis	84
G	Tabellenverzeichnis	87

Abkürzungsverzeichnis

AdRIA	Adaptronik-Research, Innovation, Application
AD-Wandler	Analog/Digital-Wandler
ADC	Analog Digital Converter
BGA	Ball Grid Array
CAD	computer-aided design
CCL	CMOS configurable latch
CMOS	Complementary Metal Oxide Semiconductor
CONBOX	Controller Box
DATAV_AV	Data available
DA-Wandler	Digital/Analog-Wandler
DDS	Direct Digital Synthesis
DIP	Dual in-line package
DSP	Digitaler Signalprozessor
EMK	Institut für Elektromechanische Konstruktionen
ESA	Eingebettete Systeme und ihre Anwendungen
ESD	electrostatic discharge
HACON	Hardware-Accelerated Controller
IO	input/output
ISE	Integrated Software Environment
LED	light-emitting diode
LOEWE	Landes-Offensive zur Entwicklung Wissenschaftlich- ökonomischer Exzellenz
LUT	Lookup-Tabelle
PC	Personal Computer
PCB	Printed Circuit Board
PI-Regler	Proportional-Integral-Regler
ppr	pulses per round
PWM	Pulsweitenmodulation
REFIO	Reference input output
rpm	rounds per minute
SiKo-Platine	Signalkonditionierungsplatine
SMD	Surface-mounted device
SNS	Sense
SRAM	Static random-access memory
SW	Switch
UART	Universal Asynchronous Receiver Transmitter
USB	Universal Serial Bus
VHDL	Very High Speed Integrated Circuit Hardware Description Language

Symbolverzeichnis

C	Kondensator
c_0	Lichtgeschwindigkeit
C_p	Piezomotorkapazität
e	Regelabweichung
e_i	summierte Regelabweichung für I-Anteil
e_k	Regelabweichung für P-Anteil
ϵ_r	Relative Permittivität
I	Strom
K_i	Integrierbeiwert
K_p	Proportionalbeiwert
L	Induktivität
l_{krit}	kritische Leitungslänge
P	Steckverbinder
R	Widerstand
T_a	Abtastzeit
t_{rise}	Anstiegszeit
U	Spannung
$U_{0,AC}$	Wechselspannung
$U_{0,DC}$	Gleichspannung
u_1	Stellgröße nach Regelung
u_2	Stellgröße nach DA-Wandler
u_3	Stellgröße nach Verstärker
u_{max}	maximale Spannung
u_{min}	minimale Spannung
$u_{min,dez}$	minimaler dezimaler Spannungswert
v	Geschwindigkeit
V_1	Potential an Stelle 1
V_{Banken}	Spannung der I/O-Bänke
V_{ccaux}	Zusatzspannung
V_{ccint}	Spannung des Prozessorkerns
w	Führungsgröße
ω	Drehzahl
x	aktuelle Drehzahl
y	Regelgröße, Drehzahl des Piezomotors
z	Störgröße

1 Einleitung

Das Institut ESA des Fachbereichs Informatik und das Institut EMK des Fachbereichs Elektro- und Informationstechnik arbeiten im LOEWE-Zentrum AdRIA am Einsatz von Piezomotoren für adaptronische Anwendungen. Ein Ziel dieser Zusammenarbeit ist es eingebettete Systeme mit Aktorik und Ansteuerungselektronik zu realisieren. Im Rahmen dieser Zusammenarbeit wurden bereits die CONBOX und eine Signalkonditionierungsplatine (SiKo-Platine) entwickelt.

Die CONBOX-Platine arbeitet analoge Signale auf und wird mit einem TMS320F28335 eZdsp Starter Kit, auf dem sich ein DSP befindet, verwendet. Das Starter Kit dient als Reglerplattform. Ein Vergleich mit einem FPGA als Reglerplattform soll in Zukunft erfolgen.

Die SiKo-Platine wurde in einer vorhergehenden Arbeit [1] entwickelt und nicht vollständig in Betrieb genommen. Die SiKo-Platine wird verwendet, um die analogen Signale bei der Ansteuerung eines Piezomotors zu konditionieren und für eine Reglerplattform bereitzustellen.

In Abbildung 1.1 ist der Signalverlauf dargestellt. Der DA-Wandler wandelt das von einer Reglerplattform vorgegebene Signal ($U_{M, \text{digital}}$) in eine analoge Spannung ($U_{M, \text{analog}}$) um. Diese Spannung wird im Piezoverstärker verstärkt und wieder auf die Platine geleitet, um die Höhe der Spannung (U_{eff}) zu bestimmen und den Strom (I_{eff}) zu messen. Die Phasenlage zwischen Motorstrom und Motorspannung (U_{Phase}) wird ebenfalls ermittelt. Über eine Messbrücke wird das Drehmoment ($U_{\text{Drehmoment}}$) bestimmt. Diese analogen Werte werden vom AD-Wandler gemessen und für eine Reglerplattform bereitgestellt. Die Reglerplattform kann über die Ansteuerung eines Relais die Drehrichtung des Motors verändern. Über den Encoder wird der gedrehte Winkel des Motors bestimmt. Eine detaillierte Beschreibung der Signalkonditionierung auf der Platine befindet sich in [1]. Der AD- und der DA-Wandler sowie die Messbrücke werden im Rahmen dieser Arbeit verwendet, wurden jedoch in der Vorarbeit nicht in Betrieb genommen. Die Messung der Effektivwerte und der Phasenlage wird nicht benötigt.

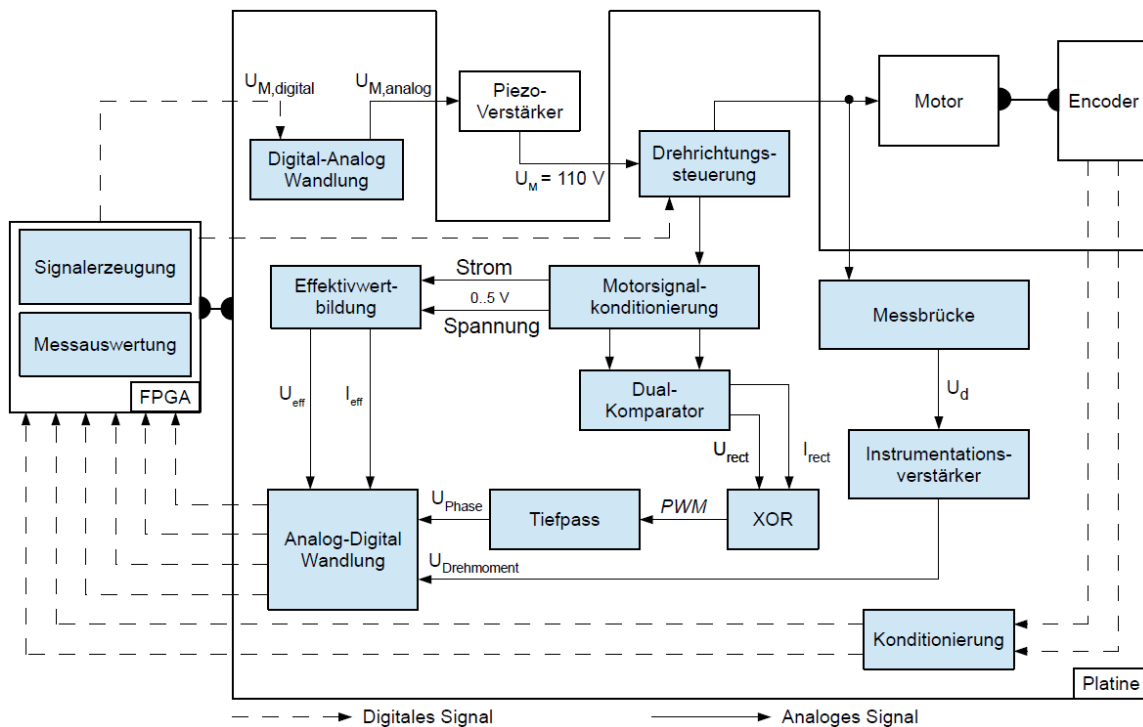


Abbildung 1.1: Signalverlauf zur Ansteuerung des Piezomotors auf der SiKo-Platine [1]

Eine mögliche Anwendung für die SiKo-Platine und den Einsatz eines Piezomotors im Zusammenhang mit einem FPGA als Reglerplattform ist die Realisierung einer virtuellen Rasterung. In Abbildung 1.2 ist der Aufbau einer mechanischen Rasterung abgebildet. Wenn der Drehknopf am Zahnrad gedreht wird, steigt das Drehmoment an, weil die Feder zusammengedrückt wird und eine höhere Kraft benötigt wird, um weiterzudrehen. Sobald die Spitze eines Zahnes überschritten wird, springt der Stift wieder in die Lücke. Falls die Spitze noch nicht überschritten wurde und der Drehknopf losgelassen wird, springt der Drehknopf wieder in die letzte Rasterposition zurück. Der Nachteil dieser mechanischen Rasterung ist die fest vorgegebene Anzahl der Rastpunkte. Für eine Änderung der Anzahl der Rastpunkte bzw. ein anderes Drehmoment, müssen die mechanischen Bauteile gewechselt werden.

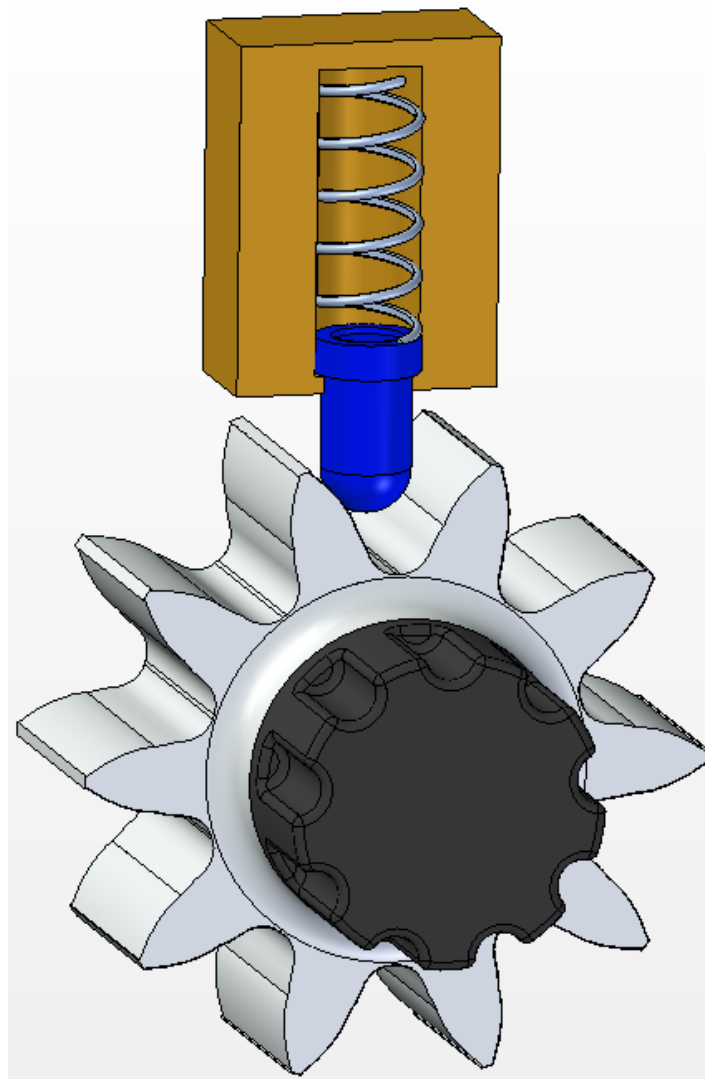


Abbildung 1.2: Mechanisches Rastsystem. Die Feder drückt den Stift nach unten. Beim Drehen wird der Stift hochgedrückt und das Drehmoment steigt. Nach Überschreiten der Spitze eines Zahnrads springt der Stift in die nächste Rasterung. Wenn der Drehknopf vorher losgelassen wird, dreht sich der Drehknopf zurück und der Stift springt in die letzte Rasterung.

Piezomotoren eignen sich für die eingebettete Realisierung aktorischer Systeme mit integrierter Ansteuerungselektronik. Sie besitzen einen kompakten Aufbau (Werte für den in der Arbeit verwendeten Piezomotor U-164.11 von Physikinstrumente GmbH & Co. KG 44 x 24 x 9 mm) und bieten hohe Werte für Beschleunigung (bis zu 5 g), Geschwindigkeit (500 mm/s) und Präzision (kleinste Schrittweite 0,05 μm) [2]. Ein weiterer Vorteil für einen kompakten Aufbau mit einer Ansterelektronik ergibt sich aus der Wirkweise des Piezomotors. Der Piezomotor erzeugt keine elektromagnetischen Felder und wird von diesen auch nicht beeinflusst [3].

Als Reglerplattform bietet sich der Einsatz eines FPGAs an. Neben der reinen Berechnung lassen sich bei FPGAs die Ansteuerung der Sensorik und Aktorik gut parallelisieren.

Der Bericht gliedert sich wie folgt: Kapitel 1 bildet die Einleitung mit einer Motivation und der Vorstellung bisher erstellter Platinen. In Kapitel 2 wird die Aufgabenstellung detailliert beschrieben. Dazu gehören die konkreten Anforderungen sowie die Rahmenbedingungen für die zu entwerfende Platine. Ferner werden die zu implementierenden VHDL-Module erläutert. Kapitel 3 behandelt eine Literaturrecherche, in der verwandte Arbeiten vorgestellt werden, in denen FPGAs zur Ansteuerung von Piezomotoren verwendet werden. Kapitel 4 befasst sich ausführlich mit dem Aufbau der Platine. Dabei wird auf Besonderheiten beim Platinenentwurf eingegangen und die Verschaltung der einzelnen Komponenten erläutert. In Kapitel 5 werden die einzelnen Module und die Implementierung der Zielanwendungen beschrieben. Kapitel 6 befasst sich mit der Inbetriebnahme der verwendeten Platinen und den Ergebnissen der Arbeit, die anschließend diskutiert werden. Es wird u. a. erläutert, wieso eine Implementierung zur Regelung eines Drehmoments mit diesem Piezomotor nicht möglich war. Abschließend gibt es eine Zusammenfassung und einen Ausblick in Kapitel 7.

2 Klärung der Aufgabenstellung

In diesem Kapitel wird die Aufgabenstellung detailliert beschrieben. Dabei wird eine Gesamtübersicht über die verwendeten Geräte gegeben und die Funktion der bereits vorhandenen analogen Platinen erläutert. Außerdem werden die Rahmenbedingungen für die zu entwerfende Platine genannt.

Die Aufgabenstellung gliedert sich in zwei Abschnitte. Ziel des ersten Teils ist der Entwurf, die Realisierung und die Evaluation einer gedruckten Schaltung, welche einen Xilinx Spartan-6 FPGA für den Einsatz als Reglerplattform (im Folgenden HACON-Platine genannt) bereitstellt. Der zweite Teil befasst sich mit der Implementierung von Hardware-Modulen für das FPGA, die die analogen Platinen ansteuern.

2.1 PCB-Entwurf einer FPGA-basierten Reglerplattform

Konkrete Anforderungen an den Entwurf der HACON-Platine ergeben sich aus zwei Anwendungen, der CONBOX-Platine und der SiKo-Platine.

2.1.1 CONBOX

Die CONBOX-Platine wurde in früheren Arbeiten mit einem TMS320F28335 eZdsp Starter Kit als Reglerplattform betrieben. Das Starter Kit enthält ein DSP. Die zu entwerfende HACON-Platine muss diese Reglerplattform ersetzen. Dazu muss sie auf die CONBOX-Platine aufsteckbar sein. Die Steckverbinder der beiden Platinen müssen sich deshalb auf derselben Position befinden. Die Pinbelegung der gemeinsam verwendeten Steckverbinder ist vorgegeben. Durch die Position der Steckverbinder sind ebenfalls die minimalen Abmessungen der zu entwerfenden HACON-Platine vorgegeben.

Des Weiteren muss die zu entwerfende HACON-Platine alle Aufgaben des Starter Kits übernehmen. Der DSP auf dem Starter Kit enthält einen internen AD-Wandler. Da das FPGA keinen internen AD-Wandler besitzt, ist ein externer AD-Wandler notwendig.

2.1.2 SiKo-Platine

Die SiKo-Platine wird verwendet um die Signale für eine Steuerung bzw. Regelung des Piezomotors zu konditionieren und für eine Reglerplattform bereitzustellen. Sie wird mit einem 40 pin Flachbandkabel mit der zu entwerfenden HACON-Platine verbunden. Die Position des Steckverbinders ist frei wählbar.

Eine Inbetriebnahme des DA- und AD-Wandlers ist zu erfolgen, ebenso ist die Messbrücke zur Ermittlung des Drehmoments in Betrieb zu nehmen.

Um mit der SiKo-Platine eine Anwendung zu entwickeln, die einen Piezomotor ansteuert, sind folgende Komponenten vorgegeben:

Der Piezomotor (Physikinstrumente GmbH, U-164.11) und der zugehörige Piezoverstärker LE 150/100 EBW von Piezomechanik GmbH, ein Encoder zur Drehwinkelerfassung (HEDS-5600 von Avago Technologies) sowie eine Hysteresebremse zur künstlichen Erzeugung eines Drehzahlunabhängigen Drehmoments am Piezomotor (HB-10M-2DS von mobac).

Die Auswahl der zu verwendenden Komponenten für die HACON-Platine war nicht Teil der Aufgabenstellung. Folgende Komponenten wurden in der Aufgabenstellung vorgegeben:

- XC6SLX150-3FGG484C (FPGA),
- CP2102-GM (USB-UART-bridge),
- M25P64-VMF6P (64 Mbit Konfigurationsspeicher),
- BS62LV1600EIP55 (16 Mbit SRAM), ersetzt durch LY62W20488ML-55LLI,
- M29W160ET70N6E (16 Mbit Flash),
- LTC6905CS5-133 (Oszillator),
- LM2853MH-3.3, LM2853MH-1.2 (Spannungswandler),
- MAX11049ETN+ (AD-Wandler) und
- DS1834AS+ (Voltage Monitor).

Der vorgegebene SRAM-Speicher war beim Distributor nicht mehr verfügbar und wurde durch einen gleichartigen Speicher ersetzt.

Die Pins des FPGA sind auf vier Bänke verteilt. Es sollen möglichst wenige Bänke verwendet werden. Durch das Abschalten einer unbenutzten Bank wird der Stromverbrauch gesenkt.

Das FPGA gibt es in diversen Varianten, die sich in der Anzahl der verfügbaren Speicher- und Logikeinheiten sowie der verwendbaren Pins unterscheiden. Aus diesem Grund gibt es bei der Verdrahtung die Forderung lediglich die Pins zu verwenden, die ebenfalls in der kostengünstigsten Variante (XC6SLX25) vorhanden sind.

Die HACON-Platine wird mit dem CAD-Programm Altium Designer entworfen. Eine Einarbeitung in die umfangreichen Funktionen der Software ist notwendig.

Mit der Software werden Schaltpläne und die dazugehörige Platine entworfen. Die Software ist besonders gut geeignet für den Platinenentwurf mit mehr als zwei Lagen. Das Routen von Leiterbahnen sowie die Bearbeitung von vielen parallelen Leiterbahnen wird unterstützt.

Die Fertigung und Bestückung der HACON-Platine erfolgt durch den kommerziellen Anbieter PCB-Pool. Dessen Vorgaben (design rules) sind einzuhalten.

2.2 Hardware-Beschreibung der Regleranwendungen

Das FPGA auf der zu entwerfenden HACON-Platine wird in der Hardwarebeschreibungssprache VHDL programmiert. Eine Einarbeitung in diese Programmiersprache sowie in die Entwicklungsumgebung Xilinx ISE V14.6 ist notwendig.

2.2.1 Testmodule für CONBOX

Um die Funktionsfähigkeit und Verbindung der CONBOX mit der zu entwerfenden HACON-Platine zu testen, soll ein Testmodul entwickelt werden. Das Testmodul soll den AD-Wandler auf der HACON-Platine ansteuern und einen AD-Kanal abtasten. Das Ergebnis dieser Wandlung soll per UART und als analoges Signal wieder ausgegeben werden. Um das analoge Signal zu realisieren, wird der gemessene Wert über eine PWM ausgegeben. In der CONBOX wird daraus über einen Tiefpassfilter ein analoges Signal generiert.

2.2.2 Haptisches Feedback per Piezomotor

Für die Realisierung eines haptischen Feedbacks ist eine Drehwinkelabhängige Drehmomentregelung für einen Piezomotor auf dem FPGA zu implementieren. Als Anwendung soll eine Rasterung, wie in Abbildung 1.2 dargestellt, simuliert werden. Der Piezomotor soll derart angesteuert werden, dass sich das Drehmoment in Abhängigkeit des gedrehten Winkels erhöht.

Der AD-Wandler soll angesteuert werden, um die proportional zum Drehmoment ausgegebene Spannung zu messen. Die Signale des Encoders sind für die Bestimmung des Drehwinkels auszuwerten. Diese Informationen werden benötigt, um eine Drehmomentregelung zu implementieren. Dabei soll die Drehwinkel-Drehmoment-Charakteristik zur Laufzeit veränderbar sein, d. h. das entstehende Drehmoment sowie die Anzahl der Rastschritte sollen im Betrieb verändert werden können. In Abbildung 2.1 sind beispielhaft zwei Charakteristiken einer drehwinkelabhängigen Drehmomentregelung dargestellt.

Die schwarzen Linien zeigen eine grobe Rasterung und ein geringeres maximales Drehmoment, die grünen Linien zeigen eine feinere Rasterung und ein höheres maximales Drehmoment. Diese Verläufe simulieren die Rasterung mit einem Zahnrad, das auf einer Seite einen Anstieg hat und auf der anderen Seite steil abfällt. Nach Überschreiten der Spitze des Zahnrads, springt der Stift unverzüglich in den nächsten Rastpunkt.

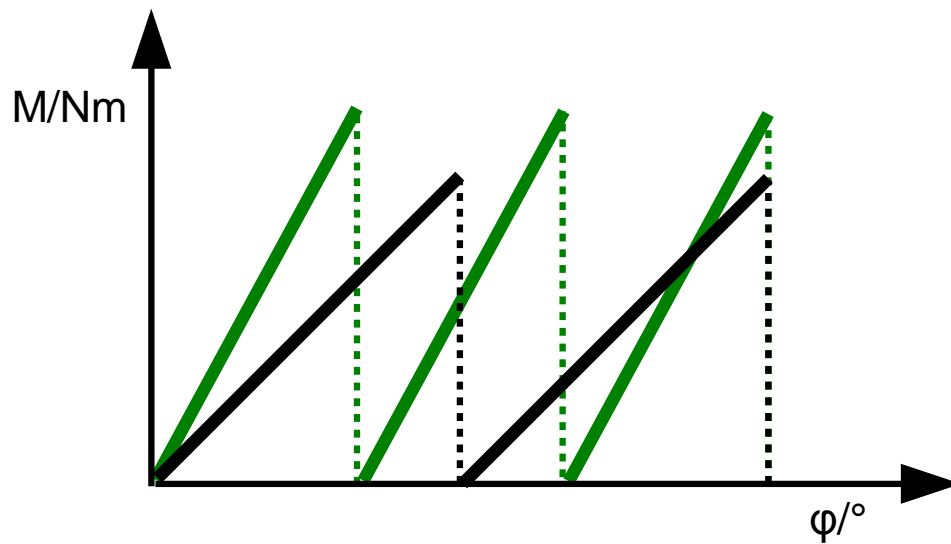


Abbildung 2.1: Zwei mögliche Charakteristiken mit verschiedenen feinen Rasterungen und verschiedenen maximalen Drehmomenten

3 Verwandte Arbeiten

Dieses Kapitel gibt einen Überblick über Arbeiten, in denen eine Drehmoment- bzw. Drehzahlregelung für Piezomotoren entwickelt wurde bei denen FPGAs als Reglerplattform zum Einsatz kommen. Eine Vielzahl an Arbeiten beschäftigt sich mit einer Positionsregelung, da in der Industrie ein großer Bedarf danach besteht.

Da sich Eigenschaften wie beispielsweise die optimale Anregungsfrequenz bei Piezomotoren im Laufe der Zeit durch reibungsbedingte Abnutzung verändern, wird in [4] eine Regelung der Läuferposition mit einem neuronalen Netz in einem FPGA implementiert. Das neuronale Netz nimmt Veränderungen wahr und ändert die Regel- und Ansteuerungsparameter. Allerdings ist das Ziel der vorliegenden Arbeit nicht die Adaption langfristiger variierender Betriebsparameter. Ein neuronales Netz wird daher nicht implementiert.

In [5] wird ein FPGA zur Signalerzeugung und Regelung der Geschwindigkeit verwendet. Das FPGA befindet sich auf einem Entwicklungsboard. Ziel der vorliegenden Arbeit ist allerdings die Entwicklung einer Platine mit einem FPGA als Reglerplattform, die die bereits vorhandenen analogen Platinen ansteuern kann. Da die Belegung der Steckverbinder und teilweise sogar die Position fest vorgegeben ist, kann das Entwicklungsboard nicht verwendet werden.

Die Ansteuerung eines Roboterarms mit mehreren Piezomotoren wird in [6] beschrieben. Dazu werden u. a. ein PC und ein Control Board verwendet. Auf dem Control Board befinden sich u. a. ein FPGA und ein Mikrocontroller. Am PC können die Vorgaben über LabView zum Controller Board gesendet werden. Die Vorgaben werden vom Mikrocontroller empfangen. Die Regelung wird ebenfalls mit dem Mikrocontroller durchgeführt. Die Berechneten Stellgrößen werden an das FPGA übergeben. Das FPGA wird lediglich dafür verwendet, um die einzelnen Piezomotoren über einen „High Speed Serial Databus“ anzusteuern.

In [7] wird die Ansteuerung eines Piezoaktors mit mehreren Freiheitsgraden erläutert. Für die Signalsynthese wird ein FPGA verwendet. Im FPGA ist ein Array von DDSs (Direct Digital Synthesizer) implementiert. Die einzelnen DDSs arbeiten unabhängig voneinander. Die Amplitude und die Frequenz können für alle Bewegungsrichtungen separat eingestellt werden. Allerdings wird in der vorliegenden Arbeit ein Piezomotor verwendet, der lediglich einen Freiheitsgrad besitzt. Aufgrund der begrenzten Bandbreite des Piezoverstärkers von 70 kHz

wird keine DDS benötigt. Außerdem ist das Ziel der Arbeit die Implementierung einer Regelung und nicht einer Steuerung.

4 Aufbau der Leiterplatte

In diesem Kapitel wird der Aufbau der HACON-Platine beschrieben. Diese umfasst das FPGA, die Spannungsversorgung, einen AD-Wandler, diverse Speicher sowie den Oszillator und weitere periphere Komponenten wie LEDs, Steckverbinder und Schalter. Die Schaltpläne und das Layout befinden sich in Anhang B.

Die Leiterplatte ist vierlagig. Innen befinden sich die beiden Versorgungslagen mit dem Groundpotential und der Versorgungsspannung von 3,3 V. Auf den beiden äußeren Lagen liegen die Signalleitungen. Die Anordnung der Lagen ist in Abbildung 4.1 visualisiert.

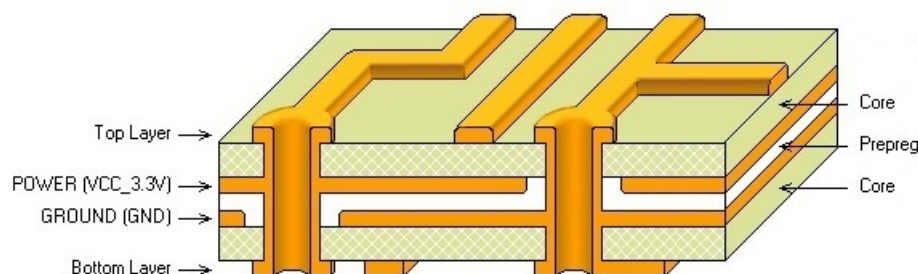


Abbildung 4.1: Anordnung der Lagen der vierlagigen HACON-Platine. In der Mitte sind die beiden Versorgungslagen 3,3 V und Ground. Außen befinden sich die Signalleitungen. [8]

4.1 FPGA

Das verwendete FPGA ist das XC6SLX150 der Spartan-6-Familie von Xilinx. Das FPGA besitzt 484 Pins mit einem 1 mm-Raster in einem BGA-Gehäuse [9]. Die Pins der Bank 0 bleiben vollständig unbeschaltet.

Das FPGA ist CCL-basiert (CMOS configurable latch). Es muss daher nach jedem Ausschalten neu konfiguriert werden. Als Konfigurationsmodus wurde der „Master Serial/SPI configuration mode“ gewählt. In diesem Modus werden die Konfigurationsdaten nach jedem Einschalten aus einem Konfigurationsspeicher ausgelesen. Als Konfigurationsspeicher wurde der in [9] empfohlene M25P64-VMF6P von Micron [10] gewählt. Um diesen Modus auszuwählen, muss der Pin M0 an die Versorgungsspannung und der Pin M1 an das Groundpotential angeschlossen werden.

Der Pin PROGRAM_B ist mit einem Taster verbunden. Durch Drücken des Tasters wird dieser Pin mit dem Groundpotential verbunden. Sobald an dem Pin für mindestens 500 ns ein Low-Pegel und anschließend wieder einen High-Pegel anliegt, werden die Konfigurationsdaten erneut aus dem Konfigurationsspeicher ausgelesen.

INIT_B wurde mit dem Voltage Monitor verbunden (siehe Kapitel 4.6).

4.2 Spannungsversorgung

Die Spannungsversorgung der Platine erfolgt über ein 5 V-Netzteil, das bis zu 4 A [11] bereitstellt. Über die HACON-Platine werden ebenfalls die angeschlossenen Platinen (CONBOX, SiKo-Platine) versorgt.

4.2.1 Spannungswandler

a) Verschaltung

Das FPGA benötigt zwei Versorgungsspannungen. Für die I/O-Bänke werden 3,3 V benötigt, für den Prozessorkern 1,2 V. Um die verschiedenen Spannungen zu realisieren, werden die Schaltregler LM2853MH-1.2 und LM2853MH-3.3 von Texas Instruments eingesetzt, die als Stepdown-Wandler arbeiten. Stepdown-Wandler wandeln die Spannung von einer höheren in eine tiefere Spannung herunter.

In Abbildung 4.3 ist die Verschaltung eines Stepdown-Wandlers für die Konvertierung von 5 V in 3,3 V veranschaulicht. Die Verschaltung für die Erzeugung von 1,2 V ist identisch aufgebaut.

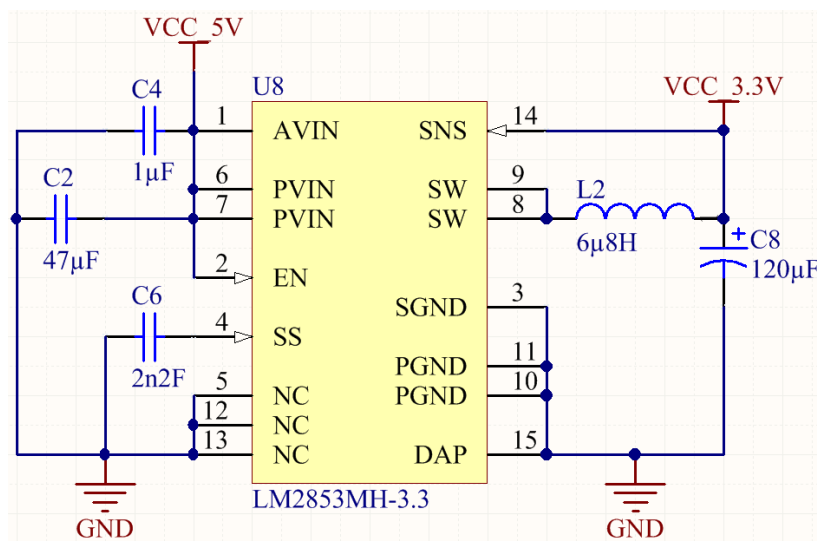


Abbildung 4.3: Verschaltung der Schaltregler für die Spannungsversorgung

Die Eingangspins (1, 6 und 7) sind mit zwei Kondensatoren verbunden (C2 und C4). Als Stützkondensator für den Spannungseingang (C2) wird ein 47 μF Keramikkondensator gewählt. Dadurch wird eine konstante Eingangsspannung garantiert. Ein hierzu parallel verschalteter 1 μF -Kondensator glättet die hochfrequenten Lastspitzen der digitalen Komponenten.

Der Kondensator C6 erzeugt einen „Soft-Start“. Damit wird die endgültige Spannung um einige Mikrosekunden verzögert erreicht. Die Dimensionierung der Spule (L2) und des Ausgangskondensators (C8) wurde dem Datenblatt [12] entnommen.

b) Layout

Der Stepdown-Wandler ist ein Regelkreis. Aus diesem Grund ist darauf zu achten, die Leiterbahnen möglichst kurz zu halten, um einen Spannungseinbruch möglichst schnell ausregeln zu können. In Abbildung 4.4 ist die Anordnung der einzelnen Komponenten zu sehen. Die Spule (L2) muss mit ihren beiden Anschlüssen möglichst nah an die Pins 8 und 9 („SW“, Switch) und Pin 14 („SNS“, Sense) des Spannungsreglers (U8). Der Ausgangskondensator (C8) muss möglichst nah an die Spule platziert werden.

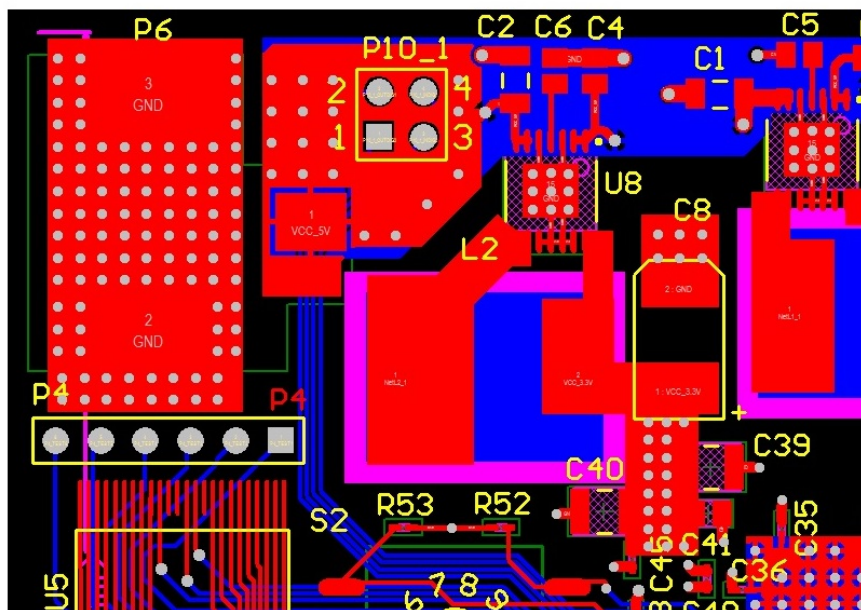


Abbildung 4.4: Anordnung der Komponenten für die Erzeugung der Versorgungsspannung

Die Spulen erzeugen ein Magnetfeld. Um das Induzieren von Strömen und Störungen zu verhindern, wird unter die Spulen auf der Unterseite und in den Versorgungslagen eine Sperrfläche gelegt. Damit können dort keine Leitungen verlegt werden.

4.2.2 Entkoppelung des FPGAs

Um eine konstante Spannungsversorgung zu garantieren, werden Entkoppelkondensatoren eingesetzt. Diese dienen dazu bei hochfrequenten Schaltvorgängen von ICs sofort ausreichend Energie bereitzustellen und damit von der Spannungsversorgung der anderen Komponenten zu entkoppeln bzw. andere ICs nicht zu beeinflussen.

Bei FPGAs mit hohen Schaltfrequenzen ist dies besonders wichtig. Deshalb wird die Anzahl der zu verwendenden Kondensatoren vom Hersteller empfohlen. In Tabelle 4.1 stehen die vom Hersteller empfohlenen und die verwendeten Werte und Mengen [13].

Anschluss\ Kondensator(Bauform)	100 μ F (1206)		4,7 μ F (0805)		470 nF (0402)	
	empf.	verw.	empf.	verw.	empf.	verw.
V_{ccint} (1,2 V)	2x100 μ F	1x220 μ F	3x4,7 μ F	1x10 μ F + 1x4,7 μ F	6x470 nF	6x470 nF
$V_{ccaux} + V_{Banken}$ (3,3 V)	5x100 μ F	2x220 μ F + 2x100 μ F	6x4,7 μ F	4x10 μ F	15x470 nF	13x470 nF

Tabelle 4.1: Übersicht über die benötigten und verwendeten Stütz- bzw. Entkoppelungskondensatoren

Die großen Kondensatoren (100 μ F bzw. 220 μ F) werden als Stützkondensatoren verwendet. Die kleineren Kondensatoren werden benötigt, um die Stromversorgung bei hochfrequenten Schaltvorgängen zu garantieren. Laut Datenblatt [13] dürfen zwei 100 μ F- zu einem 220 μ F- bzw. zwei 4,7 μ F- zu einem 10 μ F-Kondensator zusammengefasst werden. Bei den 470 nF-Kondensatoren wurden lediglich 13 Kondensatoren verwendet, weil Bank 0 nicht beschaltet ist.

4.3 AD-Wandler

Um analoge Signale zu digitalisieren, wird der AD-Wandler MAX11049 von Maxim Integrated verwendet. Der AD-Wandler hat eine Auflösung von 16 Bit und besitzt acht separate Eingänge für analoge Signale [14].

a) Verschaltung

In der Konfiguration auf der HACON-Platine verwendet der AD-Wandler die interne Referenzspannung. Aus diesem Grund wird an den Pin REFIO ein 100 nF-Kondensator angebracht.

Die RDC-Pins werden alle miteinander verbunden. Laut Datenblatt sollen die RDC-Pins mit Bypass-Kondensatoren verbunden werden, die insgesamt eine minimale Kapazität von 80 μF haben [14]. Für die fünf RDC-Pins wurde jeweils ein 22 μF -Kondensator verwendet.

An die Spannungsversorgungseingänge wird jeweils ein 100 nF Entkoppelkondensator angebracht sowie ein Stützkondensator mit 10 μF . Zwischen analogem und digitalem Ground wird nicht unterschieden. Die beiden Potentiale werden verbunden.

Sechs der acht Kanäle sind reserviert für analoge Leitungen der CONBOX-Platine. Zwei dieser Leitungen dienen als Referenzspannungen auf der CONBOX-Platine. Diese Signale werden direkt an den AD-Wandler angeschlossen. Die vier anderen Signale werden über einen Tiefpass an den AD-Wandler geleitet. Der Widerstand beträgt 0 Ohm. Der Kondensator hat auf Grund der minimalen Abtastrate den im Datenblatt empfohlenen Wert von 1 nF. Die beiden verbliebenen Kanäle werden zu Testzwecken mit einem Steckverbinder verbunden. Die angelegten Spannungen können nach dem Tausch der Widerstände des Spannungsteilers in den messbaren Bereich des AD-Wandlers verringert werden. Anschließend gibt es noch die Möglichkeit die Spannung über einen Tiefpass zu führen.

Der AD-Wandler wird im Rahmen der Arbeit zur Ansteuerung des Piezomotors nicht verwendet.

b) Layout

Das RDC-Potential gilt als Referenzspannung. Das Signal muss deshalb stabil sein. Um das zu erreichen, wird für die Spannung eine eigene Fläche eingefügt, die mit den Kondensatoren verbunden ist. Abbildung 4.5 zeigt die Anordnung des AD-Wandlers, der Tiefpässe sowie der RDC-Fläche.

Die Leitungen der analogen Signale werden kurz gehalten und die Tiefpässe nah an den AD-Wandler angebracht. Damit wird eine mögliche Störung auf den Leitungen minimiert.

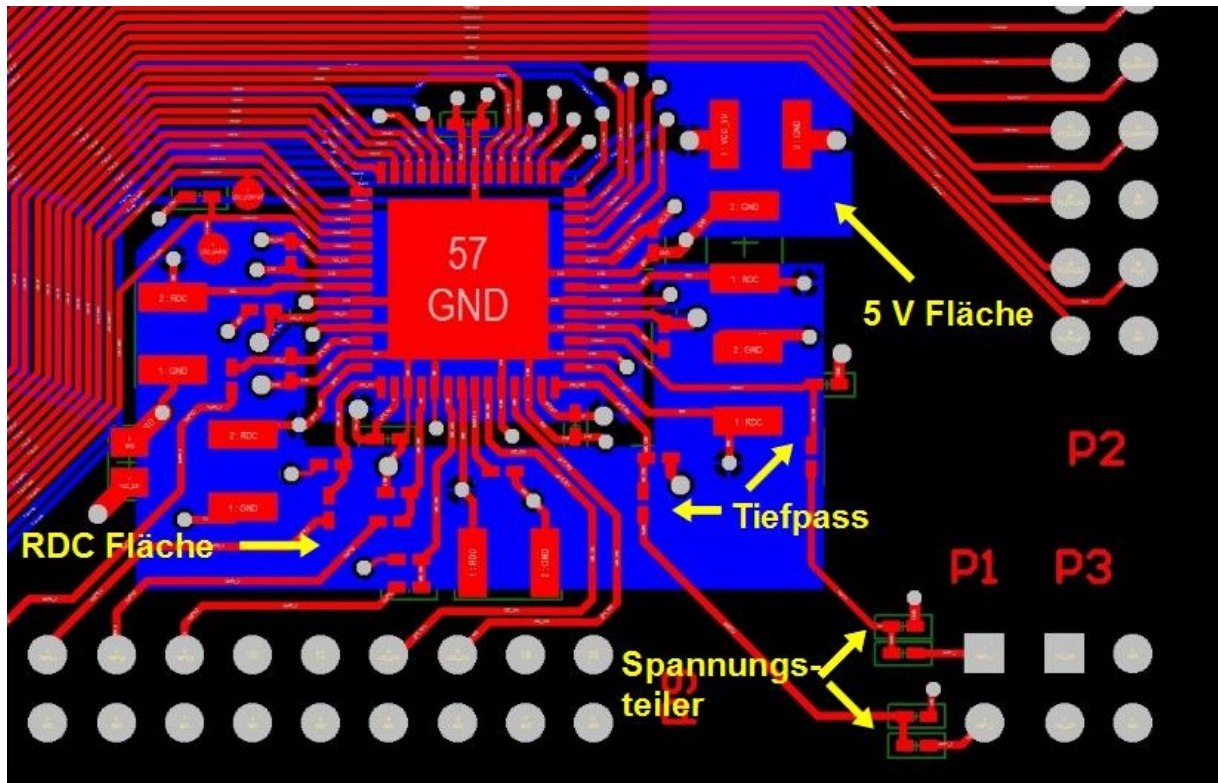


Abbildung 4.5: Anordnung der Komponenten für die AD-Wandlung

4.4 Speicher

Auf der Platine befinden sich außer dem Speicher für die Konfigurationsdaten zwei weitere Speicher, ein CMOS SRAM mit flüchtigem Speicher sowie ein persistenter Flash-Speicher.

Um die Routing-Komplexität gering zu halten, teilen sich die beiden Speicher die Adress- und Datenleitungen. Die Auswahl des aktiven Speichers erfolgt über separate Steuersignale. Für die Datenzugriffsgranularität von 8 Bit wurde der "Byte" Pin des Flash-Speichers mit dem Groundpotential verbunden [15].

4.5 USB

Um eine Kommunikation zwischen FPGA und einem PC herzustellen, wird der USB zu UART Wandler CP2102 von Silicon Labs verwendet. Die Verschaltung ist Abbildung 4.6 zu entnehmen. Der Wandler wird im Self-Powered Mode betrieben. Dazu wird der Pin VBUS mit dem entsprechenden Signal des USB-Steckverbinders verbunden [16]. An die Pins Vdd und REGIN wird die Betriebsspannung 3,3 V angeschlossen. Die Datenleitungen D+ und D- sowie GND werden ebenfalls an die entsprechenden Pins des USB-Steckverbinders

angeschlossen. Der Pin RST wird über einen Pullup-Widerstand an die Betriebsspannung angeschlossen. Auf einen FPGA gesteuerten Reset wird verzichtet.

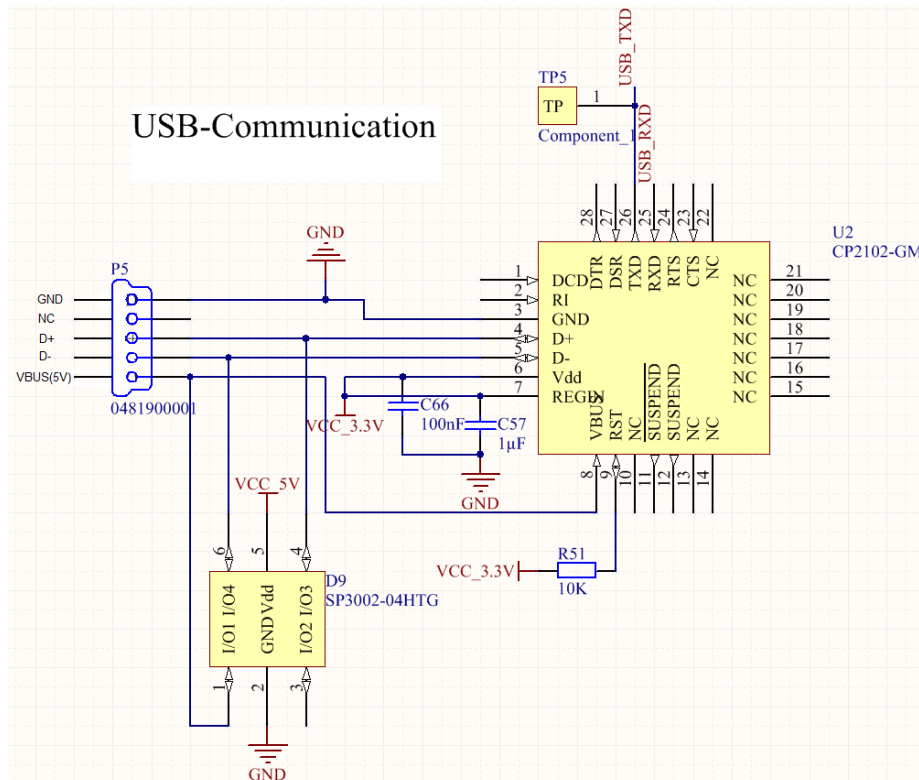


Abbildung 4.6: Verschaltung des USB zu UART Wandlers

Um den Wandler vor zu hoher Spannungen zu schützen, wird eine ESD-Schutzdiode eingesetzt. Diese Diode leitet den Strom bei einer zu hohen Spannung an die 5 V-Leitung weiter.

4.6 Voltage Monitor / Power on Reset

Der Voltage-Monitor realisiert einen Power-On-Reset. Er erzeugt folglich nach dem Anlegen der Versorgungsspannung oder dem aktivieren seines PBRST# Eingangs ein RST# Signal, das noch etwa 350ms auf 0 bleibt und danach automatisch auf 1 geht [17]. Das wird für zwei Sachen verwendet. Bei der Konfiguration nach einem Power-Cycling wird über INIT_B = RST# sichergestellt, dass der Konfigurationsspeicher bereits eine stabile Versorgungsspannung hat, bevor das FPGA die Konfigurationsdaten ausliest. Während der Laufzeit (nach der Konfiguration) führt das Betätigen des RESET Tasters dazu, dass der RESET# Eingang des FPGAs für 350ms auf 0 gehalten wird. In diesem Fall wird der Voltage Monitor zum Entprellen des Tasters verwendet.

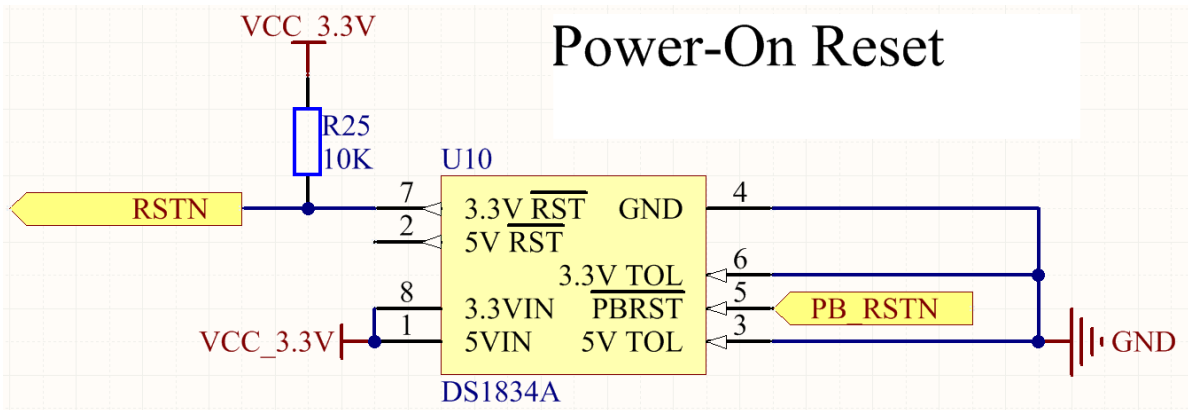


Abbildung 4.7: Verschaltung des Voltage Monitors

4.7 Oszillator

a) Verschaltung

Als Taktgenerator für das FPGA wird der Oszillator LTC6905CS5-133 von Linear Technology verwendet. Die Verschaltung ist Abbildung 4.8 zu entnehmen. An die Versorgungseingänge wird ein Bypass-Kondensator angebracht. Der Pin OE wird über einen Pullup-Widerstand mit der Betriebsspannung verbunden. Außerdem wird der Pin an das FPGA angeschlossen. Damit kann der Oszillator, beispielsweise in einem Ruhemodus, ausgeschaltet werden.

Um die höchstmögliche Frequenz von 133 MHz verwenden zu können, wurde der Pin „DIV“ auf der HACON-Platine mit der Betriebsspannung verbunden [18]. Der Ausgang des Oszillators wird mit einem „GCLK“-Pin verbunden. Diese Eingänge sind globale Takteingänge und verteilen das Taktsignal mit möglichst geringem Aufwand an alle benötigten Stellen.

133 MHz OSC

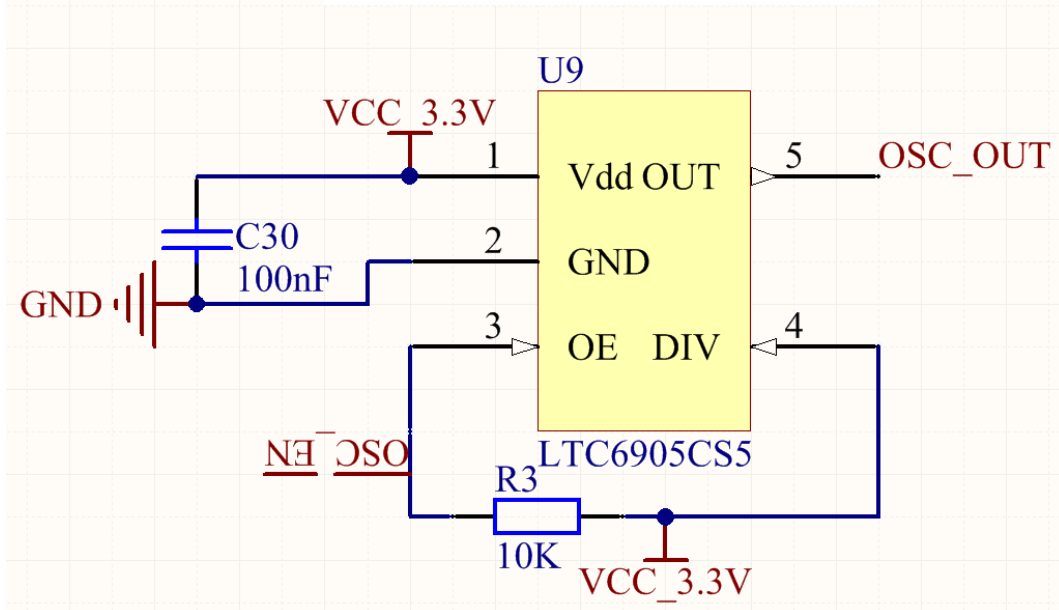


Abbildung 4.8: Verschaltung des Oszillators

b) Layout

In Abbildung 4.9 sind der Oszillator und das FPGA zu sehen. Die Leitung zwischen FPGA und Oszillator ist möglichst kurz zu wählen. Die Ausgänge von ICs haben sehr kurze Anstiegs- und Fallzeiten. Um Reflexionen auf den Datenleitungen zu vermeiden, darf die Leitungslänge eine kritische Länge nicht überschreiten. Für Rechteckimpulse gilt als Faustregel, dass eine Leitung als „lang“ gilt, wenn die einfache Laufzeit über die Leitung größer als die Hälfte der minimalen Anstiegszeit t_{rise} der Signale ist [19].

Die Ausbreitungsgeschwindigkeit auf den Leiterbahnen der Platine beträgt mit der Lichtgeschwindigkeit $c_0 \approx 30 \text{ cm/ns}$ und der relativen Dielektrizitätskonstante des Platinensubstrats $\epsilon_r = 5$ [19]

$$v = \frac{c_0}{\sqrt{\epsilon_r}} = \frac{30 \text{ cm/ns}}{\sqrt{5}} = 13,4 \text{ cm/ns}. \quad (4.1)$$

Die Anstiegszeit des Oszillatorpins beträgt $t_{\text{rise}} = 0,5 \text{ ns}$ [18]. Somit ergibt sich mit obiger Faustregel eine kritische Länge von

$$l_{\text{krit}} = \frac{1}{2} \cdot t_{\text{rise}} \cdot v = \frac{1}{2} \cdot 0,5 \text{ ns} \cdot 13,4 \text{ cm/ns} = 3,35 \text{ cm}. \quad (4.2)$$

Die Leitung vom Oszillator zum FPGA hat eine Länge von ca. 4,5 mm. Sie liegt folglich deutlich unter der kritischen Länge.
Der Bypass-Kondensator ist ebenfalls möglichst nah an die Versorgungspins anzubringen.

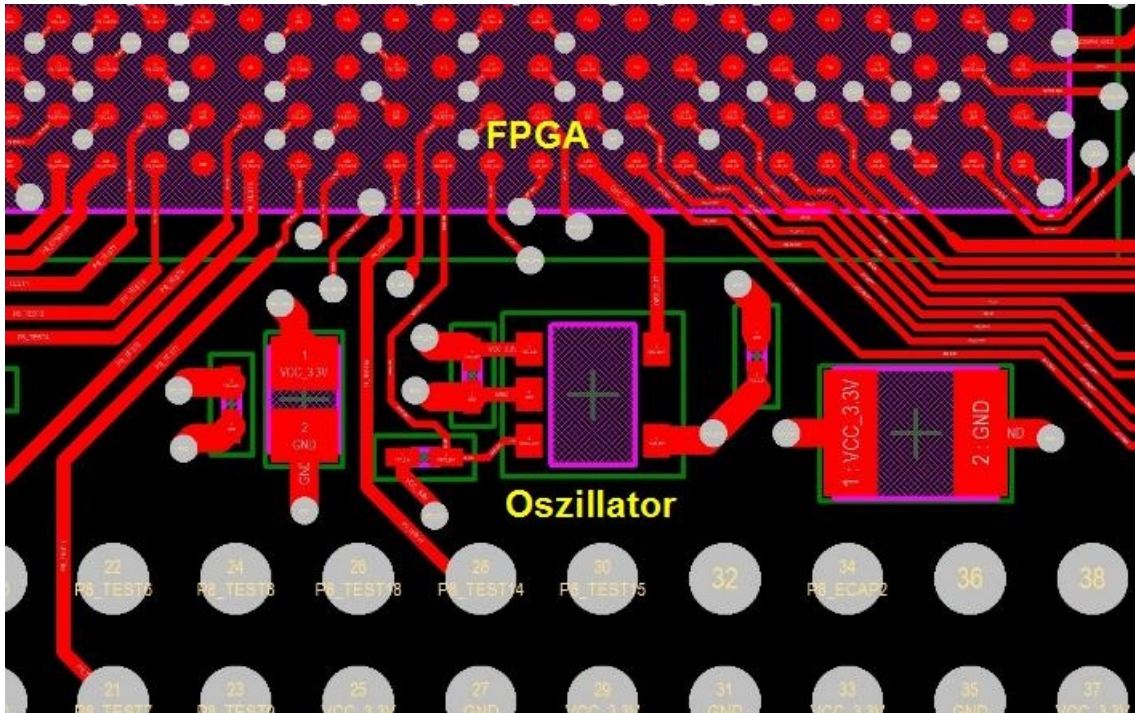


Abbildung 4.9: Anordnung des Oszillators. Die Leiterbahn zwischen Oszillator und FPGA ist möglichst kurz zu wählen.

4.8 Steckverbinder, LEDs, Taster

a) Steckverbinder

Auf der Platine befinden sich Steckverbinder um verschiedene Funktionen zu erfüllen. Die Bezeichnung und die Funktion sind Tabelle 4.2 zu entnehmen. Die Pinbelegung der einzelnen Steckverbinder befindet sich im Anhang C.

Die CONBOX-Platine wird über die Steckverbinder P8, P9, P10_1 und P10_2 mit der HACON-Platine verbunden. Die Steckverbinder müssen sich auf beiden Platinen auf derselben Position befinden. An den Steckverbindern P8 und P10_1 liegen digitale Signale an, die direkt mit dem FPGA verbunden werden. Der Steckverbinder P10_2 dient als mechanische Stütze. Die Pins sind elektrisch nicht verbunden. An Steckverbinder P9 liegen analoge Signale an,

die mit dem AD-Wandler verbunden werden. Die genauen Steckverbinderpositionen befinden sich in Anhang C.

Steckverbinder	Funktion
P1	Analoge Eingänge für AD-Wandler
P2	Verbindung zur SiKo-Platine
P3	3,3 V bzw. GND, als Referenzspannungen für Potentiometer
P4	Signalleitungen für Testzwecke mit FPGA
P5	USB Mini Buchse
P6	Buchse für Versorgungsspannung
P7	Programmierheader
P8, P9, P10_1, P10_2	Verbindung zur CONBOX-Platine

Tabelle 4.2: Überblick und Funktion der Steckverbinder auf der HACON-Platine

b) LEDs

Zu Testzwecken wurden auf der Platine LEDs vorgesehen. Die verwendeten LEDs haben eine Durchlassspannung von 2 V. Als Vorwiderstand wird ein Widerstand mit 130 Ω gewählt. Somit ergibt sich folgender Strom:

$$I = \frac{U}{R} = \frac{3,3\text{ V} - 2\text{ V}}{130\ \Omega} = 10\text{ mA}, \quad (4.3)$$

was der Hälfte des maximalen Durchlassstroms entspricht.

Die LEDs sind über die Kathode mit dem FPGA verbunden. Somit muss der Pin am FPGA einen High-Pegel haben, um die LED auszuschalten, und einen Low-Pegel, um die LED einzuschalten. Diese Konfiguration wurde gewählt, damit das FPGA nicht den gesamten Strom bereitstellen muss.

Eine Ausnahme bildet dabei LED D10. Sie ist mit dem Pin „AWAKE“ verbunden. Dieser kann so eingestellt werden, dass er nach der Konfiguration einen High-Pegel hat und die LED somit einschaltet.

c) Schalter

Für die Benutzerinteraktion zur Laufzeit, sind zwei Schalter auf der Platine vorgesehen. In Abbildung 4.10 ist die Verschaltung eines DIP-Schalters abgebildet. Bei offenem Schalter wird eine 1 von den FPGA Pins erkannt, bei geschlossenem eine 0. Die Längswiderstände dienen als zusätzlicher ESD-Schutz zu den Pullup-Widerständen.

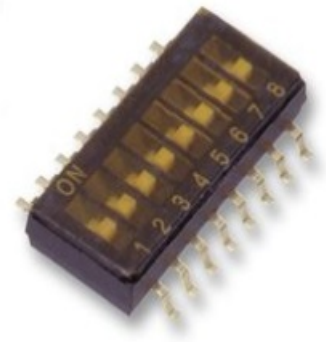
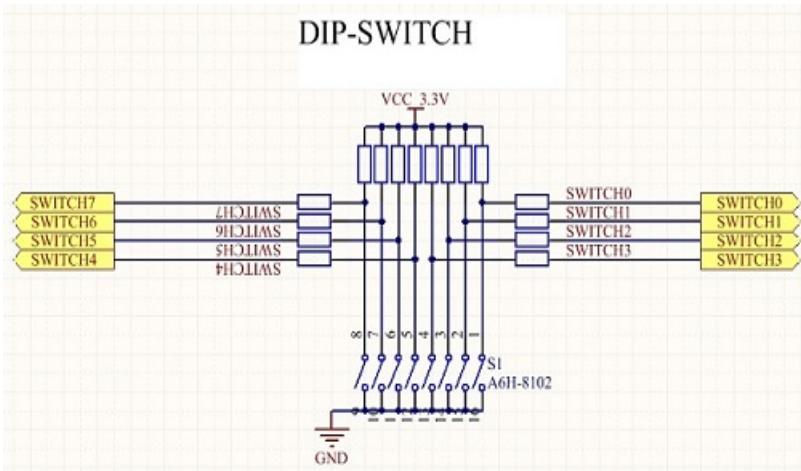


Abbildung 4.10: Verschaltung und Abbildung eines DIP-Schalters [26]

Der zweite Schalter ist ein zehnstufiger Drehschalter. In Abbildung 4.11 sind die Verschaltung sowie der Drehschalter abgebildet. An Pin zwei und fünf liegen 3,3 V an. Die restlichen vier Pins werden je nach Schalterstellung binär codiert und vom FPGA ausgewertet.

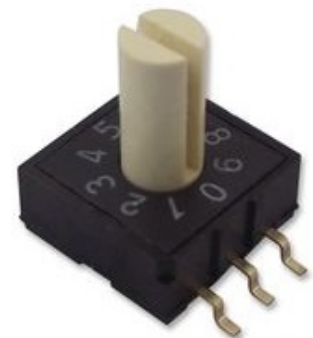
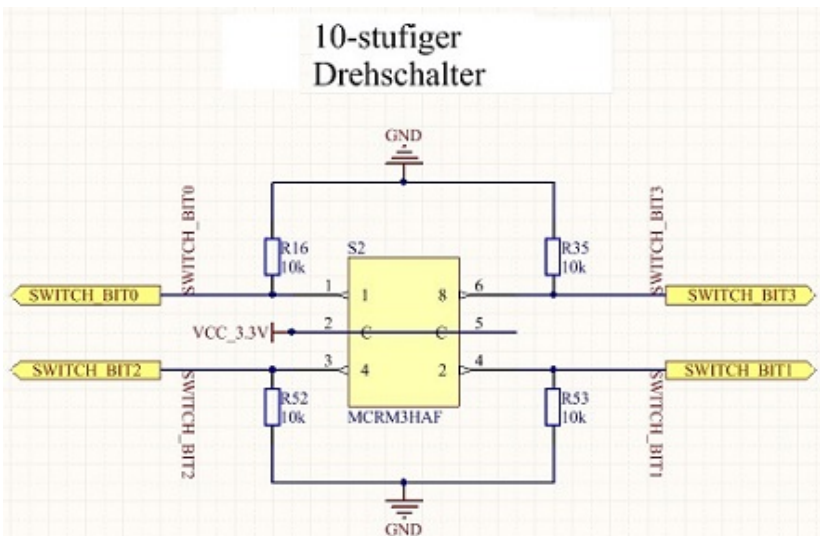


Abbildung 4.11: Verschaltung und Abbildung eines Drehschalters [27]

5 FPGA Hardware-Design

Die Hardware-Module wurden in der Hardwarebeschreibungssprache VHDL geschrieben. Für die Programmierung wurde die Entwicklungsumgebung Xilinx ISE V14.6 verwendet.

5.1 Haptisches Feedback

Abbildung 5.1 enthält eine Übersicht der VHDL-Module für die in den folgenden Unterkapiteln beschriebenen Steuerungen bzw. Regelungen. Der AD-Wandler misst eine proportional zum Drehmoment ausgegebene Spannung. Der Encoder ermittelt den gedrehten Winkel. Diese Informationen werden für die Regelung bzw. Steuerung benötigt. Diese berechnet die Stellgröße und übergibt diese an die Ansteuerung des DA-Wandlers.

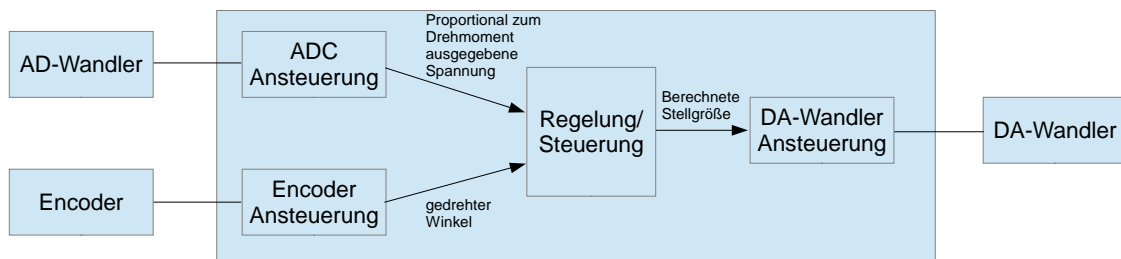


Abbildung 5.1: Modulübersicht mit Schnittstellen zum AD-Wandler, Encoder und DA-Wandler

5.1.1 Ansteuerung ADC

a) Konfiguration des AD-Wandlers

Der AD-Wandler muss konfiguriert werden, da die Standardeinstellungen nicht den gewünschten Einstellungen entsprechen. Einige Einstellungen sind durch die Verschaltung des AD-Wandlers auf der SiKo-Platine vorgegeben sind. Der Ablauf der Konfiguration läuft folgendermaßen ab: Um die Konfiguration zu starten, wird das Reset Bit im Konfigurationsregister 1 gesetzt und wieder zurückgesetzt. Danach werden die Konfigurationsregister 0 und 1 beschrieben.

Der AD-Wandler wird so konfiguriert, dass die interne Referenzspannung gewählt wird und der AD-Wandler im „Continuous conversion mode“ arbeitet. Ein im FPGA erzeugter Takt startet die Konvertierung. Der zu messende Kanal wird eingestellt. Zum Konfigurieren muss das Reset Bit auf ‚0‘ gesetzt werden. Das Triggerlevel wird auf ‚1‘ gesetzt. Dadurch wird ein Signal zum Auslesen der Daten sofort gesetzt, nachdem die konvertierten Daten verfügbar sind. Sobald eine Konvertierung beendet ist und die Daten ausgelesen werden können, wird

der AD-Wandler Pin DATA_AV auf ‚0‘ gesetzt. Der AD-Wandler Pin R/\overline{W} wird zum Schreiben bzw. Lesen verwendet. Zum Schreiben wird eine ‚0‘ angelegt, beim Lesen eine ‚1‘. Der AD-Wandler Pin \overline{RD} wird abgeschaltet. Die Ausgabe der Daten im binären Format wird eingestellt. Hierzu werden die beiden Konfigurationsregister mit den Werten „0000011000“ und „0011000000“ beschrieben.

b) Schreibvorgang

Die Pins des FPGA, die die Daten des AD-Wandlers lesen bzw. ins Konfigurationsregister schreiben, sind als „inout“ Pins deklariert. Sie können damit als Ein- und Ausgänge verwendet werden.

In Abbildung 5.2 ist der zeitliche Ablauf des Schreibvorgangs zum Konfigurieren des AD-Wandlers dargestellt. Der Ablauf erfolgt in einem Zustandsautomaten. Der Pin \overline{WR} (R/\overline{W}) muss während des Schreibvorgangs auf ‚0‘ gesetzt werden. Der Pin $\overline{CS0}$ wird beim Start des Schreibvorgangs auf ‚0‘ gesetzt. Danach werden die Daten an die Datenpins gelegt und im Anschluss wird der Pin $\overline{CS0}$ wieder auf ‚1‘ gesetzt. Diese Prozedur wird zum Konfigurieren vier Mal wiederholt. Die Wartezeiten $t_{su(R/\overline{W})}$, t_h , t_{su} , $t_w(CS)$ und $t_h(R/\overline{W})$ werden realisiert, indem Wartezustände eingefügt werden. Bei Wartezeiten unter einer Taktdauer (7,519 ns) wird in den Wartezuständen lediglich der nächste Zustand aufgerufen. Bei Wartezeiten, die länger als einen Takt dauern, wird ein Zähler hochgezählt, bis die Wartezeit abgelaufen ist. Die Wartezeiten können [20] entnommen werden.

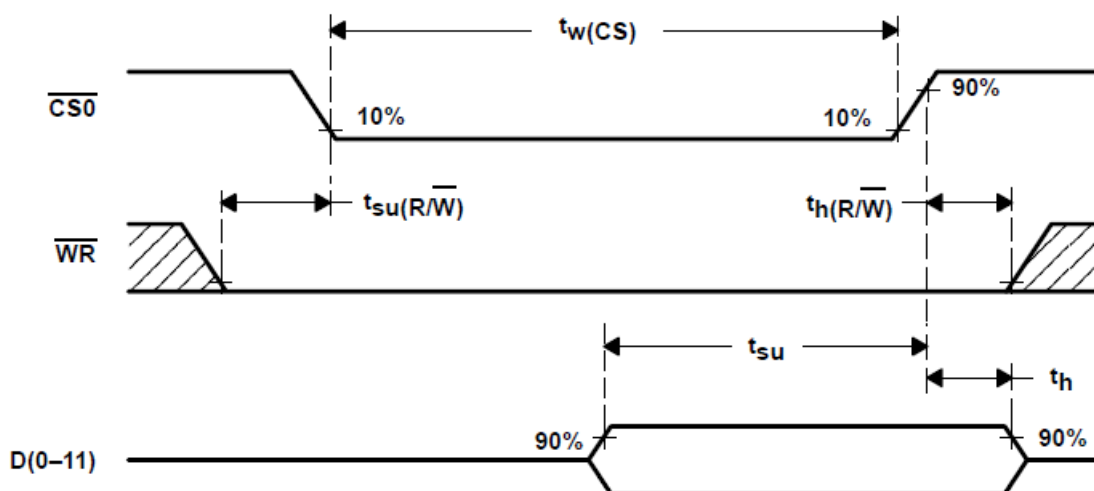


Abbildung 5.2: Ablauf des Schreibvorgangs [20]

c) Lesevorgang

Während des Konfigurierens fungieren die Datenpins als Ausgänge. Um die Pins als Eingänge verwenden zu können, muss an die Pins der Wert ‚Z‘ angelegt werden. Damit besitzen die Pins einen hohen Ausgangswiderstand und sind imstande die angelegte Spannung als High- oder Low-Pegel zu erfassen.

Der Lesevorgang und die Wartezeiten werden in einem Zustandsautomaten identisch zum Schreibvorgang realisiert. Beim Lesen wird lediglich der Pin $\overline{WR} (R/\overline{W})$ auf ‚1‘ gesetzt. In Abbildung 5.3 ist der zeitliche Ablauf dargestellt. Der Pin $\overline{DATA_AV}$ zeigt an, dass die Konvertierung fertiggestellt und die Daten bereitgestellt sind. In Abbildung 5.3 ist $\overline{DATA_AV}$ high-active. Im Hardware-Modul ist der Pin low-active.

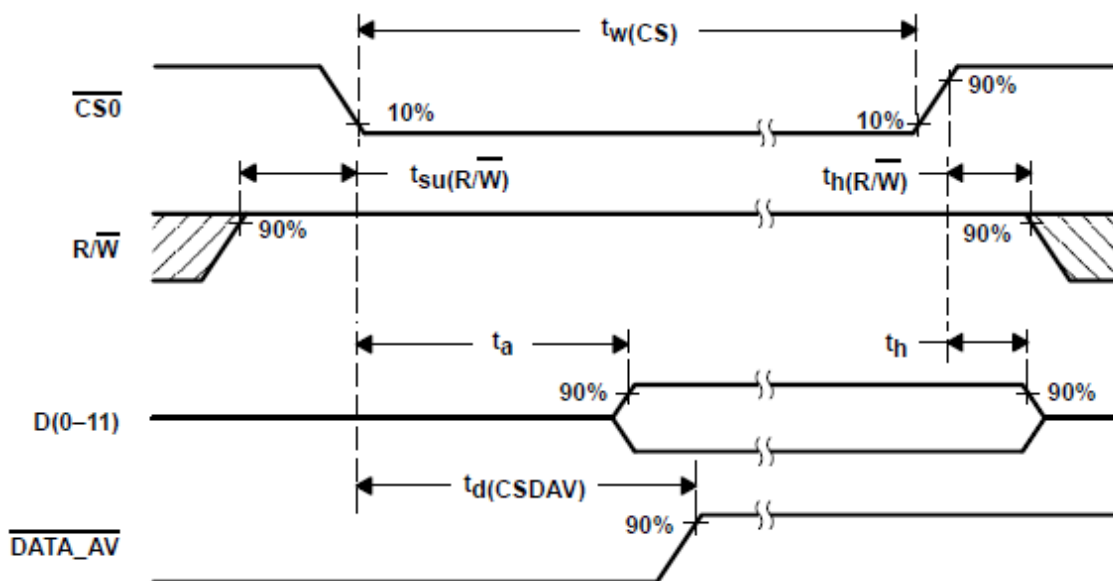


Abbildung 5.3: Ablauf des Lesevorgangs [20]

Sobald am Pin $\overline{DATA_AV}$ ein Übergang von ‚1‘ auf ‚0‘ festgestellt wird, wird der konvertierte Wert gespeichert.

d) Konvertierungstakt

Der AD-Wandler wird im „Continuous conversion mode“ betrieben. Ein Prozess erzeugt einen Takt, der über einen Pin des FPGA ausgegeben wird und die Konvertierung kontinuierlich startet.

Die Taktfrequenz wird erzeugt, indem ein Zähler hochgezählt wird. Ein Überlauf erfolgt bei

$$CONV_CLK_CNT = \frac{CLOCK_FREQUENCY}{CONV_CLK_FREQUENCY * 2} = \frac{133 \text{ MHz}}{6 \text{ MHz} * 2} = 12. \quad (5.1)$$

Bei der Division entstehen nur ganzzahlige Werte. Der tatsächliche Wert beträgt 11,083. Dabei entsteht ein Fehler 7,6%. Für den Ablauf ist es unerheblich, dass die am AD-Wandler anliegende Spannung nicht mit der maximalen Frequenz abgetastet wird.

Nach dem Überlauf wird der Zähler zurückgesetzt und der Pin für den Takt umgeschaltet.

5.1.2 Ansteuerung des Piezomotors

a) Drehrichtung

Das FPGA kann die Drehrichtung über ein Relais auf der SiKo-Platine verändern. Dazu muss am FPGA Pin „DIRECTION_LR“ eine ‚1‘ oder eine ‚0‘ angelegt werden.

b) Motorspannung

Der Piezomotor wird mit einer Frequenz von 160 kHz angesteuert. Als Signalform für die Motorspannung wurde die Rechteckspannung gewählt, da diese mit dem geringsten Aufwand implementiert werden kann, und da die Bandbreite von 70 kHz [21] des Piezoverstärkers das Signal in allen Fällen nicht anders als verzerrten Sinus ausgeben wird. Der Maximalwert des Rechtecksignals wird in anderen Prozessen ermittelt und an den Prozess für die Ansteuerung des DA-Wandlers übergeben. Als DA-Wandler auf der Siko-Platine wird der CA3338 von intersil Corporation verwendet.

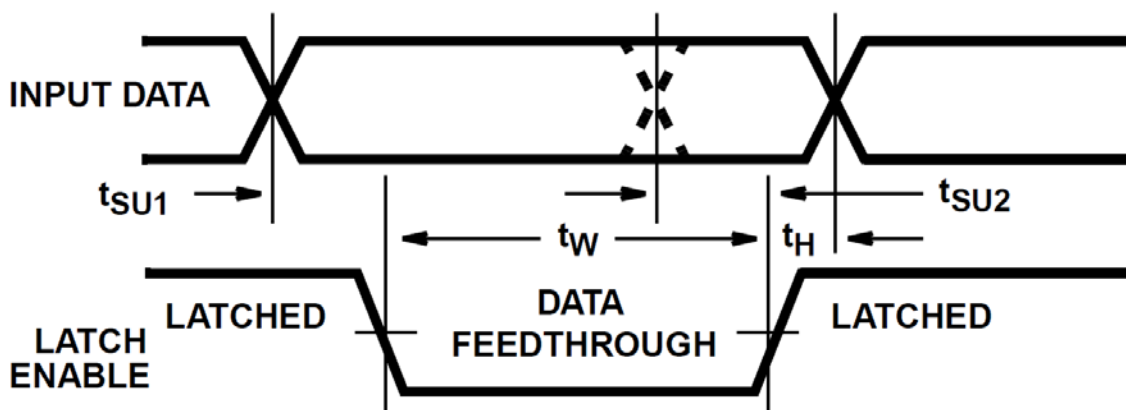


Abbildung 5.4: Zeitlicher Ablauf der Ansteuerung des DA-Wandlers [22]

In Abbildung 5.4 ist der zeitliche Ablauf der Ansteuerung des DA-Wandlers dargestellt. Um die Rechteckspannung anzulegen, wird ein Zustandsautomat mit sechs Zuständen benötigt. Folgender Code wurde implementiert:

```

1  if(piezoCnt = PIEZO_CNT_MAX - 1) then
2      piezoCnt <= 0;
3  else
4      piezoCnt <= piezoCnt+1;
5      if(piezoCnt = PIEZO_CNT_MAX * 0 / 8) then
6          P2_DAC_Data <= "00000000";
7      elsif(piezoCnt = PIEZO_CNT_MAX * 1 / 8) then
8          P2_DAC_LE <= '0';
9      elsif(piezoCnt = PIEZO_CNT_MAX * 3 / 8) then
10         P2_DAC_LE <= '1';
11         elsif(piezoCnt = PIEZO_CNT_MAX * 4 / 8) then
12             P2_DAC_Data <= DAC_Daten;
13         elsif(piezoCnt = PIEZO_CNT_MAX * 5 / 8) then
14             P2_DAC_LE <= '0';
15         elsif(piezoCnt = PIEZO_CNT_MAX * 7 / 8) then
16             P2_DAC_LE <= '1';
17         end if;
18     end if;

```

Die Konstante PIEZO_CNT_MAX wird folgendermaßen berechnet:

$$PIEZO_CNT_MAX = \frac{CLOCK_FREQUENCY}{MOTOR_FREQUENCY} = \frac{133\text{ MHz}}{160\text{ KHz}} = 832. \quad (5.2)$$

PIEZO_CNT_MAX Takte werden bei einer Taktfrequenz von 133 MHz benötigt für eine Periode der Frequenz 160 kHz. Die Variable piezoCNT wird in jedem Takt inkrementiert. In Abhängigkeit ihres Wertes werden die Daten angelegt und der Steuerpin P2_DAC_LE gesetzt bzw. zurückgesetzt. Wenn piezoCNT den Wert PIEZO_CNT_MAX – 1 erreicht hat, wird der Zähler wieder zurückgesetzt.

Die Wartezeiten t_{su1} , t_H , t_{su2} und t_w werden eingehalten, da sie deutlich kürzer sind als die Übergänge zwischen den einzelnen Zuständen.

Wenn der Piezomotor sich nicht drehen soll, wird der Wert „00000000“ übergeben. Somit liegt am Piezomotor keine Spannung an.

Die Auflösung des DA-Wandlers beträgt 8 Bit [22]. Der DA-Wandler kann Spannungen zwischen 0 V und 5 V erzeugen.

5.1.3 Ansteuerung Encoder

An dem Piezomotor ist der Encoder HEDS-5600 angeschlossen. Encoder werden verwendet, um den gedrehten Winkel bzw. die Drehzahl von drehenden Achsen zu ermitteln.

a) Pegelerfassung der Encodersignale

Der Encoder verwendet zwei Datenleitungen, auf denen ein Gray-Code erzeugt wird. Es ändert sich immer nur eine Spur auf einmal. In Abbildung 5.5 sind die Spuren der beiden Datenleitungen abgebildet. Im ersten Teil dreht sich die Achse in eine Richtung und ändert dann im zweiten Teil die Drehrichtung.

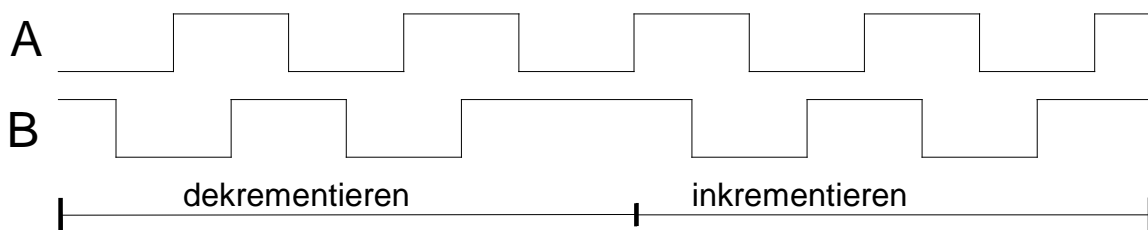


Abbildung 5.5: Spur A und B eines Encoders

Die Signale des Encoders liegen am FPGA an. In einem Prozess werden die Pegel der beiden Spuren erfasst. Der Pegelstatus wird im nächsten Takt von einem anderen Prozess übernommen. In dem Zustandsautomaten in Abbildung 5.6 wird der Pegelstatus verarbeitet und je nach Zustandswechsel der Encodersignale wird eine Variable zum In- bzw. Dekrementieren gesetzt. In Abhängigkeit des gewählten Betriebszustands wird der Zähler verarbeitet.

Die Zustände in Abbildung 5.6 sind nach dem Zustand der beiden Spuren A und B benannt. Die Implementierung der Pegelermittlung und des Zustandsautomaten wurde [23] entnommen.

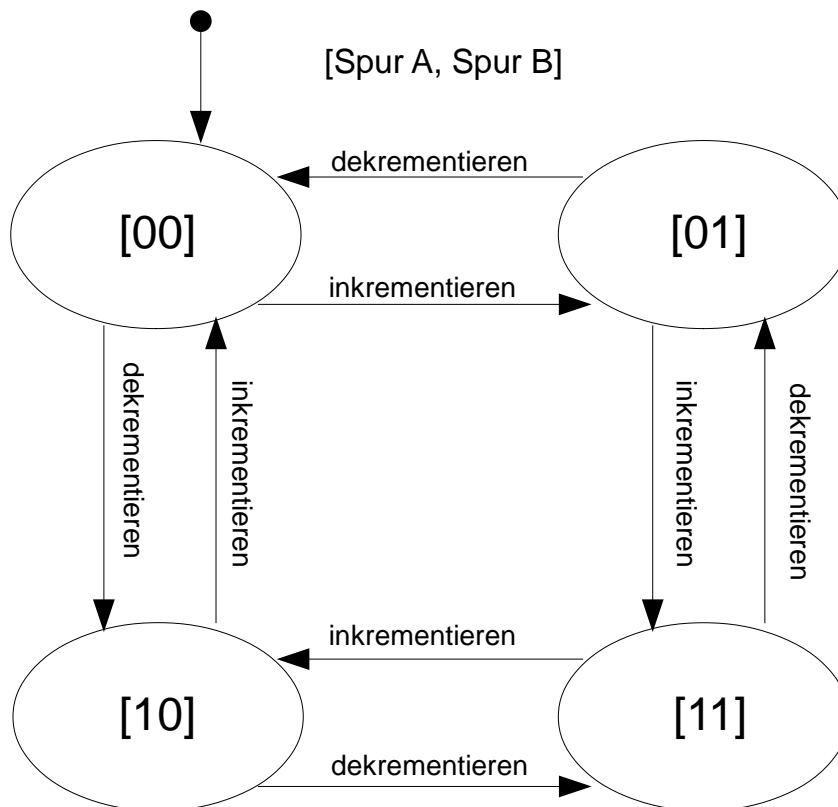


Abbildung 5.6: Zustandsautomat zur Bestimmung des gedrehten Winkels bzw. Drehzahl des Piezomotors

b) Berechnung der maximalen Drehzahl

Der Encoder hat eine Auflösung von 500 ppr [24], d. h. bei einer Drehung um 360° muss der Zustandsautomat 500 Mal durchlaufen werden. Der inkrementierte Zähler hat somit den Wert 2000.

Das FPGA kann in jedem Taktzyklus eine Zustandsänderung erkennen, somit sind das bei einer Taktfrequenz von 133 MHz 133 Millionen Zustandsänderungen. Bei 2000 Zustandsänderungen pro Umdrehung ergibt das eine maximale Drehzahl von

$$\omega = \frac{133000000}{2000} = 66500 \text{ s}^{-1}, \quad (5.3)$$

was deutlich über der maximalen Drehzahl des Encoders liegt (30.000 rpm) [24].

5.1.4 Drehmoment Regelung

Der Ablauf in Abbildung 5.7 wurde bis auf den Regler komplett implementiert. Auf die einzelnen Prozesse wird nicht detailliert eingegangen. Lediglich der Ablauf wird beschrieben, da die Regelung des Drehmoments nicht in Betrieb genommen werden konnte. Das vom Piezomotor erzeugte Drehmoment ist nicht proportional zu der angelegten Spannung, folglich ist eine Regelung nicht möglich. Die genaue Begründung befindet sich in Kapitel 6.2.

Die proportional zum Drehmoment ausgegebene Spannung eines Drehmomentsensors wird auf der SiKo-Platine gemessen. Sie liefert Informationen über die Drehrichtung sowie, ob ein Schwellwert für das Drehmoment überschritten wurde. Der Schwellwert beträgt ca. 150 mV und wurde experimentell ermittelt. Wenn der Drehknopf losgelassen wird und in den letzten Rastpunkt dreht, liegt ein Drehmoment am Piezomotor an. Dieses muss kleiner als der Schwellwert sein, sonst ermittelt der Zustandsautomat einen falschen Zustand. In Abhängigkeit der Drehrichtung wird das Relais gesteuert. Der gedrehte Winkel wird über die Encodereingänge ermittelt. In Abhängigkeit des gedrehten Winkels wird in einer LUT die Führungsgröße ermittelt. Auf diese Weise können verschiedene Drehmomentcharakteristika erstellt werden. Durch die Führungsgröße kann bestimmt werden, ob sich der Drehknopf in der Nulllage befindet.

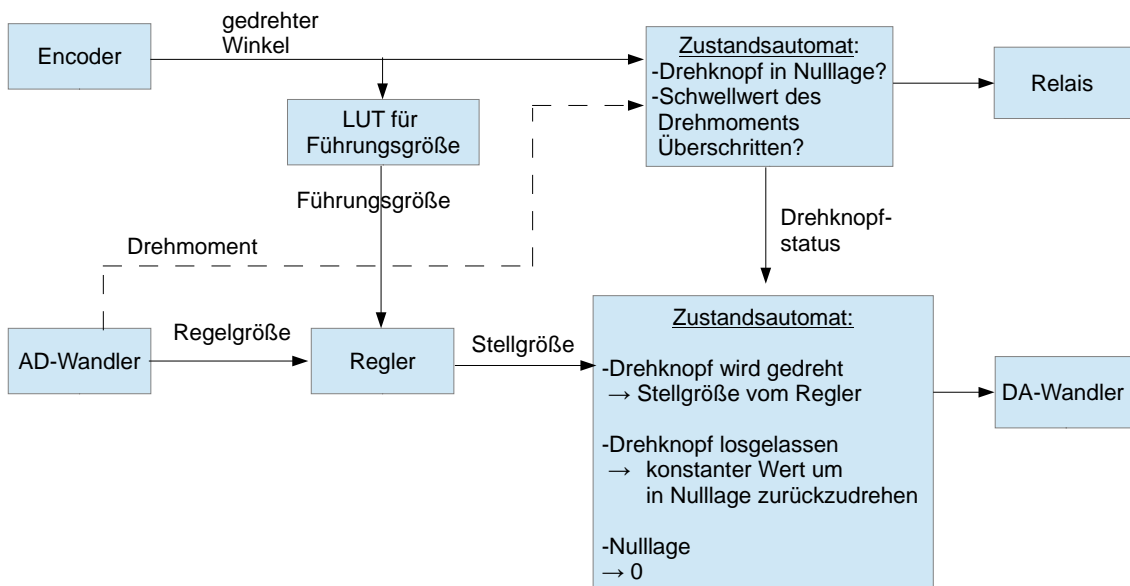


Abbildung 5.7: Ablauf der Drehmomentregelung

Die Informationen über Nulllage (genaue Erläuterung in 5.1.5 b) und anliegendem Drehmoment entscheiden, wie die Spannung ermittelt wird, die über den DA-Wandler und den Piezoverstärker an den Piezomotor angelegt wird. Folgende drei Zustände sind möglich:

- Nulllage: Wenn sich der Drehknopf in der Nulllage befindet, wird unabhängig vom anliegenden Drehmoment der Wert Null an den DA-Wandler übergeben. Der Piezomotor dreht sich folglich nicht.
- Drehknopf wird gedreht: Der Drehknopf wird gedreht und hat die Nulllage verlassen. In diesem Fall berechnet die Regelung in Abhängigkeit der Führungsgröße und des ermittelten Drehmoments den neuen Wert für die Spannung. Dadurch wird ein Drehmoment erzeugt, welches dem manuell eingeleiteten Drehmoment entgegen wirkt.
- Drehknopf wurde losgelassen: Der Drehknopf wurde nicht bis zur nächsten Nulllage gedreht, sondern vorher losgelassen. In diesem Fall liegt kein Drehmoment an und es wird ein konstanter Wert an den DA-Wandler übergeben, um den Drehknopf in die letzte Rasterung/Nulllage zu drehen.

Für die Messung in Kapitel 6.2 wurde keine Regelung implementiert. Stattdessen wird die an den DA-Wandler übergebene Spannung proportional zum gedrehten Winkel erhöht.

5.1.5 Drehmoment Steuerung für virtuelle Rasterung

Als einfachere Alternative zur Drehmomentregelung wurde eine Drehmomentsteuerung implementiert. Der Drehmomentsensor und der AD-Wandler werden dafür nicht mehr benötigt.

Abbildung 5.8 zeigt den deutlich vereinfachten Ablauf der Drehmomentsteuerung. Die einzelnen Prozesse werden im Folgenden erläutert.

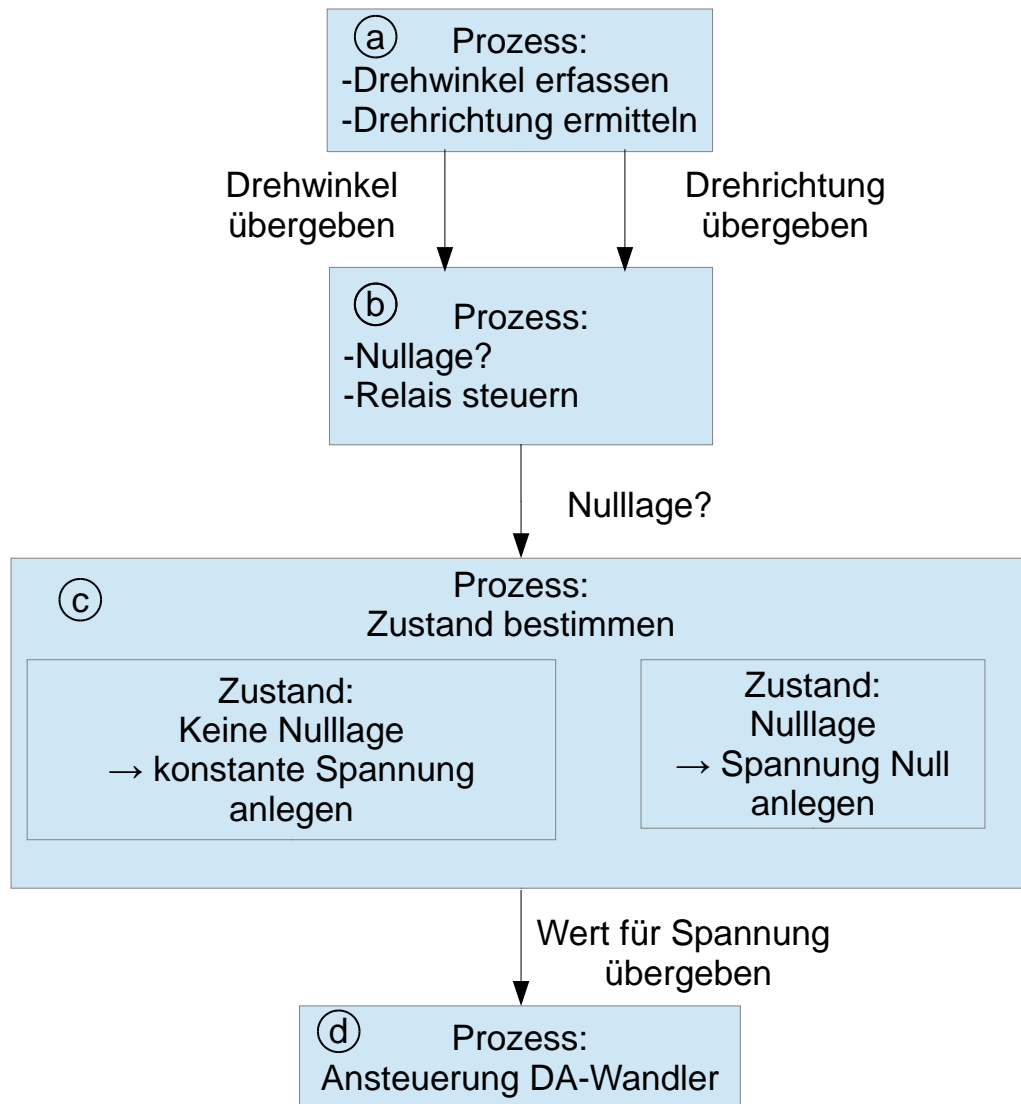


Abbildung 5.8: Ablauf der Drehmomentsteuerung

a) Drehwinkelermittlung mit Encoder

Bei der Drehmomentsteuerung wird mit dem Encoder der gedrehte Winkel des Piezomotors ermittelt. Der Zustandsautomat in Kapitel 5.1.3 wertet die Pegel des Encoders aus und setzt Variablen zum In- bzw. Dekrementieren eines Modulo-ANGLE-Zählers. „ANGLE“ ist ein Array der zur Compile-Zeit festgelegten Grenzwerte, die die Anzahl der Rastpunkte festlegen. Diese können über den Drehschalter zur Laufzeit verändert werden.

b) Überprüfung auf Nulllage und Ansteuerung des Relais

Mit Nulllage wird ein virtueller Rastpunkt bezeichnet, vergleichbar mit der stabilen Position zwischen zwei Zähnen des mechanischen Zahnrads.

Die Nulllage entspricht einem Winkel von $1,8^\circ$. Dieser Wert wurde experimentell ermittelt. Wird der Wert zu klein gewählt, kann es passieren, dass die Motorspannung zu spät ausgeschaltet wird und der Drehknopf eine Rasterung überspringt.

In dem Takt, in dem die Nulllage verlassen wird, wird das Relais für die Drehrichtung des Piezomotors derart geschaltet, dass sich der Piezomotor versucht in die entgegengesetzte Richtung zu drehen. Dafür wird die Information „Drehrichtung“ verwendet, die beim In- bzw. Dekrementieren gesetzt wird. Dadurch wird ein Drehmoment erzeugt, welches dem manuell eingeleiteten Drehmoment entgegen wirkt.

c) Auswahl der zu übergebenden Spannung

Befindet sich der Drehknopf in der Nulllage, wird ein Wert übergeben, der eine Spannung von 0 V erzeugt. Der Piezomotor wird folglich abgeschaltet.

Wurde der Drehknopf manuell aus der Nulllage bewegt wird, wird der binäre Wert „10111111“ an den DA-Wandler übergeben. Das entspricht einer Spannung von 3,7 V. Wenn der Drehknopf losgelassen wird, dreht er sich nicht mit maximaler Geschwindigkeit in die letzte Nulllage zurück. Es könnte sonst der Fall eintreten, dass sich der Piezomotor zu schnell zurückdreht und so einen Rastpunkt überspringt.

d) Ansteuerung DA-Wandler

Der vom Prozess in c) übergebene binäre Wert wird, wie in Kapitel 5.1.2 beschrieben, an den Prozess für die Ansteuerung des DA-Wandlers übergeben.

5.1.6 Drehzahl Regelung

Abbildung 5.9 zeigt den Ablauf der Drehzahlregelung. Die einzelnen Prozesse werden im Folgenden erläutert. Für die Drehzahlregelung wurden ein P-Regler und ein PI-Regler implementiert. Der P-Regler wurde anschließend modifiziert, um ein besseres Regelverhalten zu erzeugen.

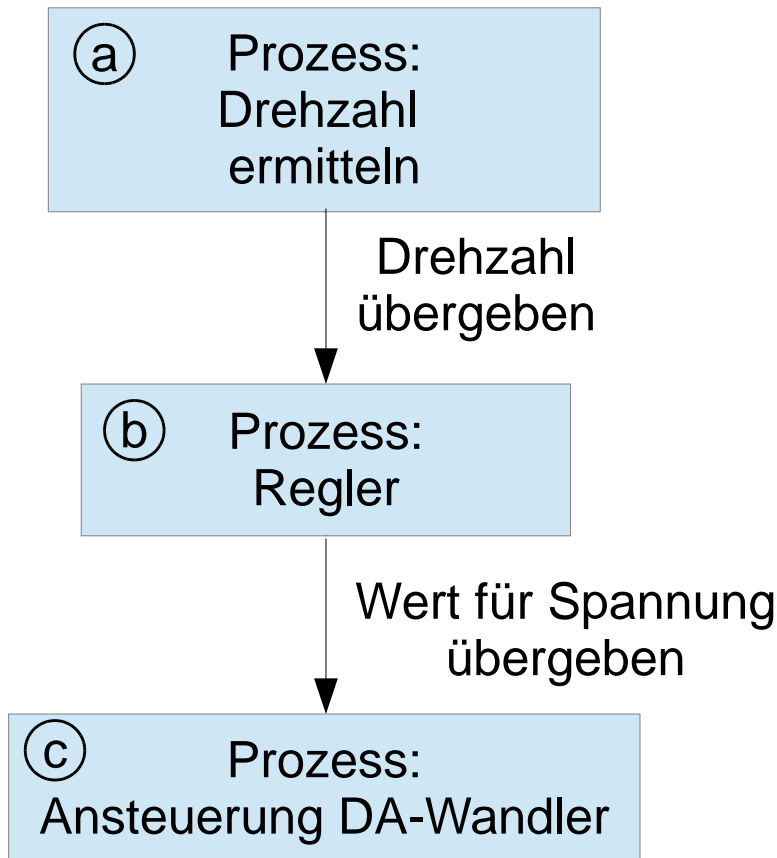


Abbildung 5.9: Ablauf der Drehzahlregelung

a) Drehzahl ermitteln

Bei der Drehzahlregelung wird über den Encoder die Drehzahl des Piezomotors ermittelt. Der Zustandsautomat in Kapitel 5.1.3 wertet die Pegel des Encoders aus und setzt Variablen zum In- bzw. Dekrementieren. Ein Zähler wird dabei 10 ms lang in- bzw. dekrementiert. Nach 10 ms wird das Signal S_10msOver für die abgelaufene Zeit gesetzt und die Anzahl der Impulse pro 10 ms an den Regelprozess übergeben.

Das Zeitfenster von 10 ms wurde aufgrund der minimalen Drehzahl von 51 rpm gewählt, was 17 Impulsen pro 10 ms entspricht. Diese Drehzahl wurde ermittelt, indem eine Spannung angelegt wurde, bei der sich der Piezomotor dreht, ohne durch Reibung anzuhalten. Ein größeres Zeitfenster verlangsamt die Dynamik der Regelung unnötig. Bei einem kleineren Zeitfenster ist die Anzahl der Impulse zu gering.

b) Regler

P- und PI-Regler:

In Abbildung 5.10 ist die Struktur der Regelung dargestellt.

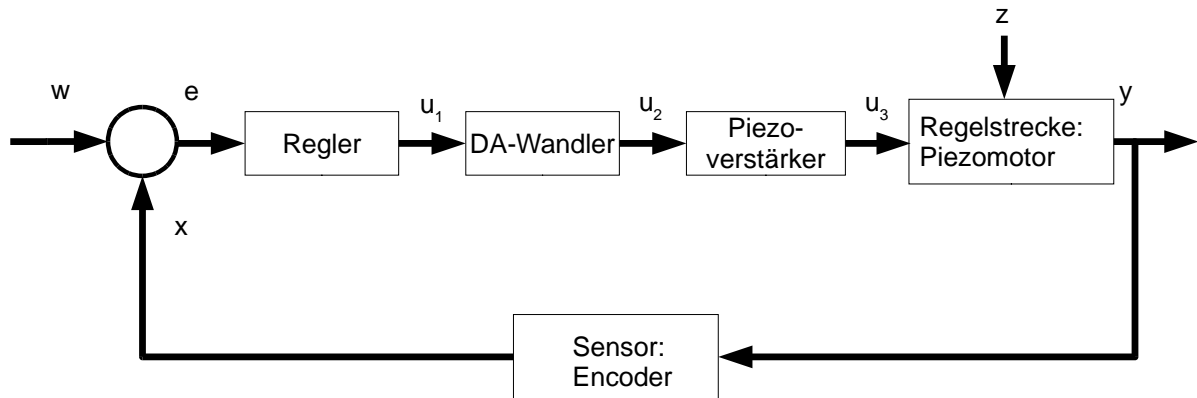


Abbildung 5.10: Struktur der Drehzahlregelung

Die Bezeichnungen bedeuten (Name des Signals im Programm):

w := Führungsgröße (SPEED)

e := Regelabweichung (Regelabweichung)

x := aktuelle Drehzahl (act_speed_in_tp10ms)

u₁ := Stellgröße nach Regelung

u₂ := Stellgröße nach DA-Wandler

u₃ := Stellgröße nach Piezoverstärker

y := Regelgröße, Drehzahl des Piezomotors

z := Störgröße, z. B. Anlegen eines Bremsmoments

Der PI-Regler besteht aus einem Proportionalanteil und einem Integrationsanteil. Soll lediglich ein P-Regler verwendet werden, muss die Konstante K_i Null gesetzt werden. Die Formel [24] für den PI-Regler ist folgende:

$$u_1 = K_p * e(t) + K_i * \int_0^t e(\tau) d\tau. \quad (5.4)$$

Regler in digitalen Systemen arbeiten zeitdiskret. Die Drehzahl wird nicht kontinuierlich ermittelt. Der Wert der neuen Drehzahl steht im Abstand von T_a=10 ms zur Verfügung. Die zeitdiskrete Formel [24] für den PI-Regler lautet:

$$u_k = K_p * e_k + K_i T_a * \sum_{i=0}^k e_i. \quad (5.5)$$

Die Konstanten K_i und T_a werden zusammengefasst zur Konstante $K_{iDrehzahl}$. Damit muss die Multiplikation dieser Konstanten nicht in jedem Zyklus neu berechnet werden.

Zusammen mit der Struktur in Abbildung 5.10 ergibt sich folgender Code:

```
1  -- Regelabweichung berechnen
2  Regelabweichung := SPEED - act_speed_in_tp10ms;
3  esum := esum + Regelabweichung;          -- Integration des I-Anteils
4  -- Begrenzung des I-Anteils, Anti Wind-Up
5  if esum <= - I_ANTEIL_BEGRENZUNG then
6      esum := -I_ANTEIL_BEGRENZUNG;
7  elsif esum >= I_ANTEIL_BEGRENZUNG then
8      esum := I_ANTEIL_BEGRENZUNG;
9  end if;
10 -- Berechnung des P-Anteils
11 sum1 := KpDrehzahl * Regelabweichung ;
12 -- Berechnung des I-Anteils
13 sum2 := KiDrehzahl * esum;
14 tmp_Stellgroesse := sum1 + sum2;          -- Summanden addieren
15 -- Stellgrößenbegrenzung
16 if tmp_Stellgroesse >= 255 then
17     tmp_Stellgroesse := 255;
18 elsif(tmp_Stellgroesse <= 0) then
19     tmp_Stellgroesse:= 0;
20 end if;
```

In Zeile 2 wird die Regelabweichung (e) berechnet. Die Regelabweichung ist die Differenz der vorgegebenen Drehzahl (w) und der vom Encoder ermittelten Drehzahl (x). Diese Abweichung wird fortwährend für den I-Anteil aufsummiert (Zeile 3). In den Zeilen 5 bis 9 wird die Summe begrenzt. Hinter dieser Saturierung verbirgt sich der Anti-Windup-Algorithmus. Eine Saturierung von $esum$ muss erfolgen, um bei blockiertem Piezomotor einen undefinierten Überlauf zu vermeiden.

In den Zeilen 11 bis 14 wird die Regelabweichung mit dem Proportionalbeiwert $K_{pDrehzahl}$ und die Summe der Regelabweichungen mit dem Integrierbeiwert $K_{iDrehzahl}$ multipliziert sowie die beiden Produkte addiert. Das Ergebnis ist die neue Stellgröße. Diese muss noch begrenzt werden (Zeile 16 bis 19). Der

minimale Wert ist Null. Der maximale Wert ist durch die Auflösung von 8 Bit vorgegeben und beträgt 255.

Modifizierter P-Regler:

Der P-Regler wurde folgendermaßen modifiziert:

Zu der in Zeile 14 berechneten Stellgröße wird der Offset SPEED_VOLTAGE addiert. SPEED_VOLTAGE ist die von der Solldrehzahl abhängige Spannung, die in einer Voruntersuchung (s. Kapitel 6.4.1) ermittelt wurde.

Außerdem wurde die Stellgrößenbeschränkung verändert. Eine sinnvolle untere Schranke ist die minimale Spannung, bei der sich der Piezomotor dreht, ohne durch Reibung oder andere Störungen angehalten zu werden. Diese Spannung wurde experimentell ermittelt und beträgt ca. $u_{\min} = 1,6 \text{ V}$. Der verwendete DA-Wandler besitzt eine Auflösung von 8 Bit [22]. Somit ergibt sich für die untere Schranke mit $u_{\max} = 5 \text{ V}$ als dezimal-Wert

$$u_{\min,dez} = (2^8 - 1) * \frac{u_{\min}}{u_{\max}} = 255 * \frac{1,6 \text{ V}}{5 \text{ V}} = 81. \quad (5.6)$$

Die Zeilen 1 bis 14 bleiben unverändert.

```
1    ... --Zeile 1 bis 14 unverändert
15   tmp_Stellgroesse := tmp_Stellgroesse + SPEED_VOLTAGE;
16   -- Stellgrößenbegrenzung
17   if tmp_Stellgroesse >= 255 then
18       tmp_Stellgroesse := 255;
19   elsif(tmp_Stellgroesse <= 81) then
20       tmp_Stellgroesse:= 81;
21   end if;
```

c) Ansteuerung DA-Wandler

Der vom Prozess in b) übergebene binäre Wert wird, wie in Kapitel 5.1.2 beschrieben, an den DA-Wandler übergeben.

5.1.7 Konfiguration zur Compile-Zeit / Laufzeit (Benutzerschnittstelle)

a) Compile-Zeit Konfiguration

In der Datei settings.vhd sind alle Konstanten deklariert. Hier befinden sich die Parameter für die Regelung. Die Solldrehzahl und der dazugehörige Spannungswert sind in diese Datei einzutragen.

Die Systemtaktfrequenz sowie alle für die Komponenten verwendeten Frequenzen. sind in der Datei deklariert.

Die Anzahl der Rastpunkte wird hier ebenfalls eingetragen. Vor dem Synthetisieren werden die Grenzwerte in das Integerarray „RASTERSCHRITTE“ in die Datei settings.vhd eingetragen. Der Grenzwert für eine gewünschte Anzahl an Rastpunkten pro Umdrehung berechnet sich folgendermaßen:

$$Grenzwert = \frac{2000}{Anzahl_{Rastpunkte}}. \quad (5.7)$$

b) Laufzeit Konfiguration

Zur Laufzeit wird die Auswahl der Anzahl der Rastpunkte über den Drehschalter gewählt.

Drehzahlregelung mit modifiziertem P-Regler und virtuelle Drehmomentsteuerung sind zwei Betriebsmodi, die im FPGA implementiert sind. Zwischen den beiden Betriebsmodi kann mit Schalter Null des DIP-Schalters umgeschaltet werden.

c) LED Anzeigen

Um dem Benutzer anzuzeigen, dass die HACON-Platine betriebsbereit ist, wurde ein Prozess implementiert, der die LED PowerLED blinken lässt. Sie blinkt für die beiden Betriebsmodi mit verschiedenen Frequenzen. Wird der Betriebsmodus Rasterung gewählt, blinkt die LED mit 2 Hz, während der Drehzahlregelung mit 1 Hz.

Auf der HACON-Platine befinden sich fünf weitere LEDs. Je nach Betriebsmodus zeigen sie verschiedene Informationen an.

Im Betriebsmodus Rasterung zeigen die LEDs an, wie weit der Drehknopf noch gedreht werden muss, bis der nächste Rastpunkt erreicht ist. Eine LED entspricht etwa 20 % des Restwegs.

Während der Regelung wird die Drehzahl des Piezomotors auf die LEDs abgebildet. LED1 leuchtet ab einer Drehzahl von 51 rpm. LED5 leuchtet ab einer Drehzahl von 177 rpm. Die Drehzahlen dazwischen wurden gleichmäßig verteilt.

5.2 Testmodule für CONBOX

Die Implementierung der Testmodule für die CONBOX wurde aus Zeitgründen nicht durchgeführt.

6 Ergebnisse und Diskussion der Arbeit

6.1 Inbetriebnahme

Eine Übersicht des gesamten Systems mit HACON-Platine, SiKo-Platine sowie der dazugehörigen Peripherie zur Ansteuerung des Piezomotors ist Abbildung 6.1 und Tabelle 6.1 zu entnehmen.

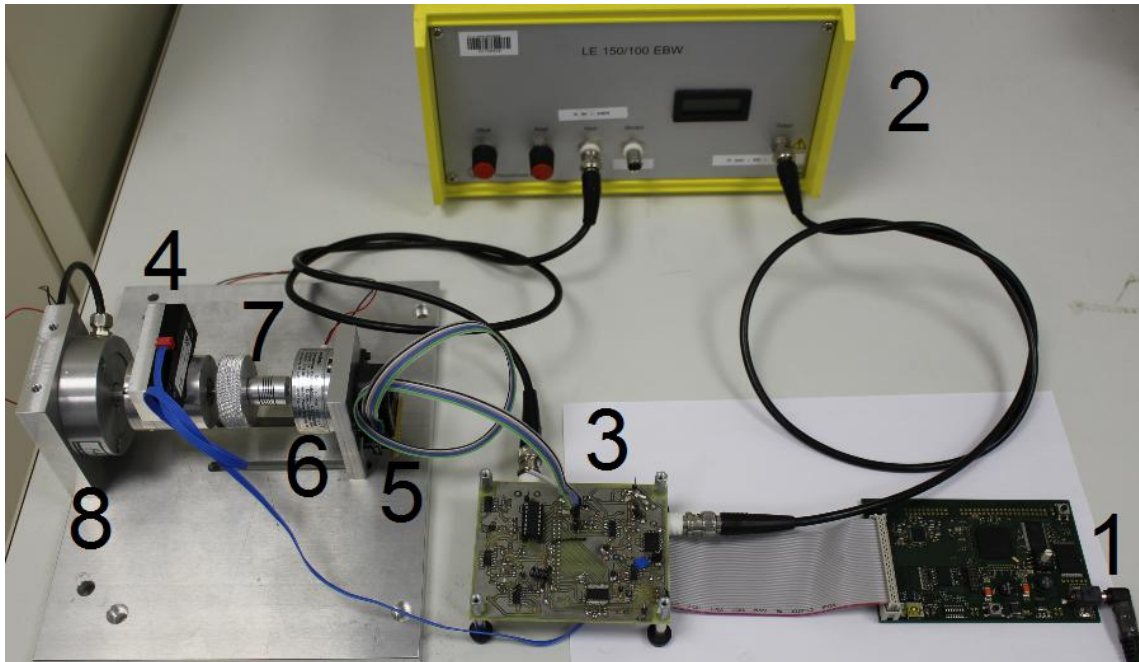


Abbildung 6.1: Gesamtübersicht über die verwendeten Platinen sowie der Peripherie zur Ansteuerung des Piezomotors

Nr.	Bezeichnung	Funktion
1	HACON-Platine	Reglerplattform mit FPGA
2	Piezoverstärker LE 150/100 EBW von Piezomechanik GmbH	Verstärkt die vom DA-Wandler erzeugte Spannung
3	SiKo-Platine	Platine um die analogen Signale der Motoransteuerung zu messen, konditionieren und für das Auslesen durch die HACON- Platine bereitzustellen
4	Piezomotor U-164.11 von Physikinstrumente GmbH & Co. KG	Motor für Regelung, bzw. Simulation einer Rasterung
5	Encoder HEDS-5600 von Avago Technologies	Erfassung des gedrehten Winkels bzw. der Drehzahl des Piezomotors
6	Hysteresebremse HB-10M- 2DS von mobac	Erzeugt Bremsmoment
7	Drehknopf	Benutzerschnittstelle für haptisches Feedback
8	Drehmomentsensor ATF315 von Althen	Messung der proportional zum Drehmoment ausgegebenen Spannung

Tabelle 6.1: Bezeichnung und Funktion der verwendeten Geräte zur Ansteuerung des Piezomotors

6.1.1 HACON-Platine

In Abbildung 6.2 ist die bestückte Platine mit den einzelnen in Kapitel 4 beschriebenen Baugruppen zu sehen. Der Voltage-Monitor und der Oszillator sind aufgrund der kleinen ICs nicht separat markiert.

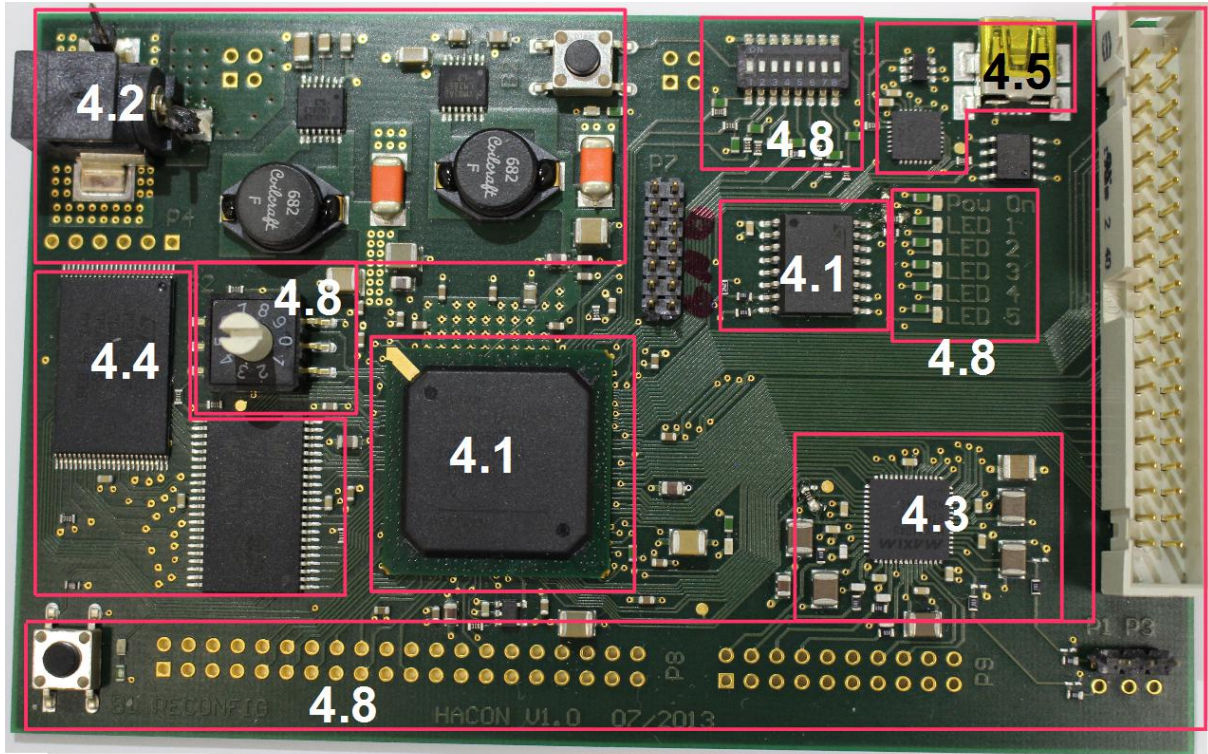


Abbildung 6.2: Bestückte HACON-Platine. Die Nummerierung entspricht den Unterkapiteln in Abschnitt 4. Voltage-Monitor und Oszillator sind nicht nummeriert.

Nach der Verifikation der Spannungs- und Taktversorgung per Oszilloskop wurde das FPGA mit einem Testdesign zum Ansteuern der LEDs programmiert. Das Laden des Bitstreams aus dem Konfigurationsspeicher und das Zurücksetzen per Taster B1 bzw. B2 wurden ebenfalls erfolgreich getestet. AD-Wandler, USB zu UART Wandler, SRAM-Speicher und Flash-Speicher wurden nicht in Betrieb genommen, da sie in der vorliegenden Arbeit nicht verwendet wurden.

6.1.2 SiKo-Platine

Die SiKo-Platine wurde im Rahmen einer vorhergehenden Arbeit [1] nicht vollständig in Betrieb genommen. Der DA-Wandler und der AD-Wandler sowie einige andere Komponenten wurden bestückt und in Betrieb genommen. Der AD-Wandler ersetzt den Spannungsteiler R33 / R35 zur Generierung der 2,5 V Referenzspannung.

Die Ansteuerung des Relais zur Änderung der Drehrichtung des Piezomotors hat nicht funktioniert. Am Steckverbinder, der die Platine mit dem Piezomotor verbindet, wurden zwei Anschlüsse vertauscht.

Auf der SiKo-Platine befindet sich eine kapazitive Messbrücke zur intrinsischen Messung des Drehmoments. Die Messbrücke besteht, wie in Abbildung 6.3 zu sehen, aus einer Reihenschaltung der Piezomotorkapazität C_p und einem Kondensator $C=33\text{ nF}$. Parallel dazu befindet sich eine weitere Reihenschaltung aus einem Kondensator $C_x \approx C_p$ mit und einem weiteren Kondensator C .

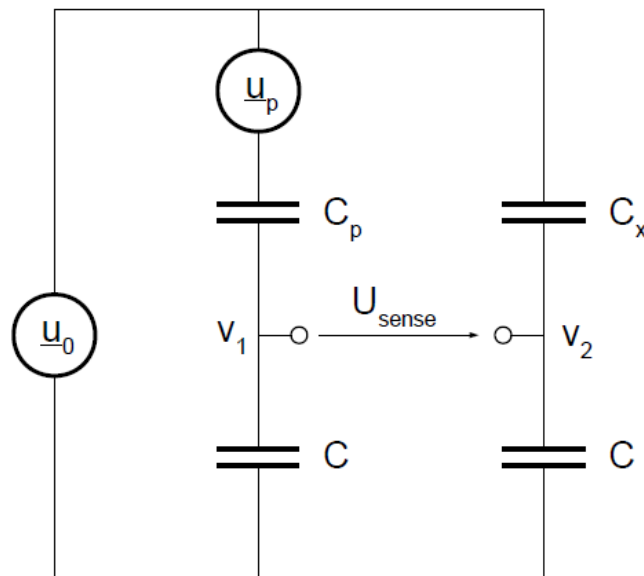


Abbildung 6.3: Messbrücke zur intrinsischen Drehmomentmessung

Die Potentiale V_1 und V_2 an den Mittelabgriffen der beiden kapazitiven Spannungsteiler liegen laut [1] bei nichtanliegendem Drehmoment bei ca. 30 V. Gemessen wurde bei $U_0 \approx 150\text{ V}$ mit 155 kHz allerdings für $V_1 \approx 60\text{ V}$ und für $V_2 \approx 1\text{ V}$.

Die erste Vermutung war eine fehlerhafte Messung der Motorkapazität. Eine erneute Messung mit einem LCR-Messgerät und einem Spektrumanalysator ergab einen Wert von $C_p \approx 470\text{ pF}$, was identisch mit dem Ergebnis in [1] ist.

Um mögliche unbekannte Fehlerquellen auf der Platine auszuschließen, wurde der linke Zweig der Messbrücke separat auf einer Lochrasterplatine aufgebaut. Die Messungen ergaben dasselbe Verhalten wie auf der SiKo-Platine.

Anschließend wurden für den Kondensator C verschiedene Werte eingesetzt. Für $4,7\text{ nF} \leq C \leq 14,7\text{ }\mu\text{F}$ schwankte V_1 zwischen 55 V und 61 V. U_0 setzt sich zusammen aus einem Gleichspannungsanteil von $U_{0,DC}=75\text{ V}$ und einem Wechselspannungsanteil von $U_{0,AC}=150\text{ V}$. Die Kondensatoren bilden für den

Gleichspannungsanteil einen Spannungsteiler von zwei unendlichen Widerständen. Der Gleichspannungsanteil wird folglich halbiert. Der Wechselspannungsanteil berechnet sich mit der Formel für den kapazitiven Spannungsteiler [19]:

$$V_1 = U_{0,AC} \frac{c_p}{c + c_p}. \quad (6.1)$$

Folglich sollten für V_1 Spannungen zwischen 37,5 V und 51 V gemessen werden. Bis zum Abschluss der Arbeit konnte dieses Verhalten nicht geklärt werden. Eine mögliche Ursache könnte die begrenzte Bandbreite von 70 kHz [21] des Piezoverstärkers sein. Weitere Untersuchungen zum Ausschluss des Piezoverstärkers als Fehlerquelle sind zu erfolgen.

6.2 Messungen mit Drehmomentsensor

a) Durchführung

Zur Simulation der in Kapitel 2.2.2 beschriebenen Rasterung wird das am Piezomotor anliegende Drehmoment benötigt. Da die intrinsische Drehmomentmessung nicht in Betrieb genommen werden konnte, wurde der externe Drehmomentsensor ATF315 von Althen angeschlossen. Die Integration in den Messstand ist Abbildung 6.1 zu entnehmen. Die proportional zum Drehmoment ausgegebene Spannung wird mit dem Messverstärker KWS 3082 A von HBM verstärkt. Die verstärkte Spannung liegt im Bereich zwischen -3,5 V und +3,5 V. Der AD-Wandler auf der SiKo-Platine besitzt einen Spannungsmessbereich von 1,5 V bis 3,5 V. Um die verstärkte Spannung des Drehmomentsensors in den messbaren Bereich des AD-Wandlers zu konvertieren, wird ein invertierender Verstärker mit einem Verstärkungsfaktor von $A=-0,283$ und einem Levelshift um 2,5 V eingesetzt. Abbildung 6.4 zeigt die Verschaltung des verwendeten Operationsverstärkers AD8044 von Analog Devices. Die Versorgungsspannung beträgt 5 V. Die Verstärkerschaltung wurde auf einer Lochrasterplatine aufgebaut und mit Fädeldrähten mit der SiKo-Platine verbunden.

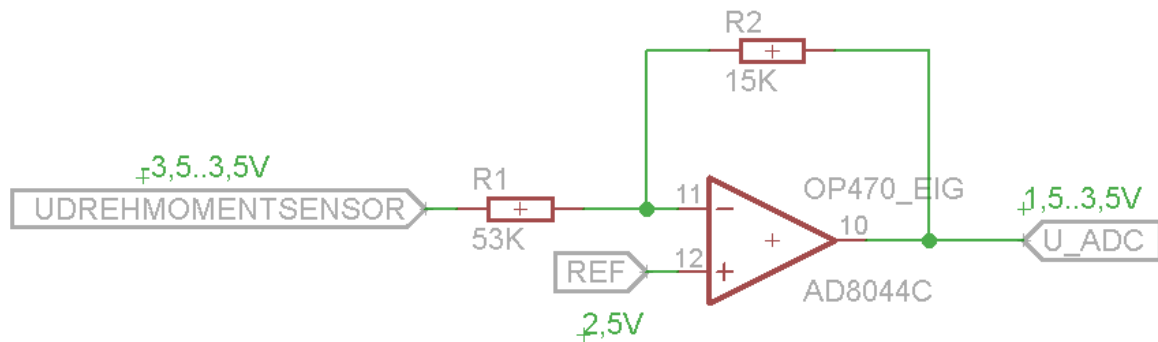


Abbildung 6.4: Verschaltung des invertierenden Operationsverstärkers, um die Spannung des Drehmomentsensors in den Messbereich des AD-Wandlers zu konvertieren

Als Vorversuch für die Drehmomentregelung soll zunächst der Einfluss der Motorspannung auf das Drehmoment untersucht werden. Der Drehknopf wird bis zum nächsten Rastpunkt gedreht. Dabei wird in Abhängigkeit des gedrehten Winkels die Motorspannung gesteigert. Die proportional zum Drehmoment ausgegebene Spannung wird aufgenommen.

b) Ergebnisse

In Abbildung 6.5 ist der Einfluss der Motorspannung auf das entstehende Drehmoment zu sehen. Die blaue Linie C_3 zeigt den Verlauf der vom DA-Wandler erzeugten Motorspannung, Kanal C_4 (grün) zeigt den Verlauf der dem Drehmoment am Piezomotor proportionalen Spannung. Bei Punkt 1 beginnt die Krafteinwirkung auf den Drehknopf. Das Drehmoment steigt an, bis das Reibmoment in Punkt 2 überschritten ist und die Drehung beginnt. Die Motorspannung wird danach proportional zum gedrehten Winkel bis zum Maximum erhöht. In Punkt 3 ist der nächste Rastpunkt erreicht.

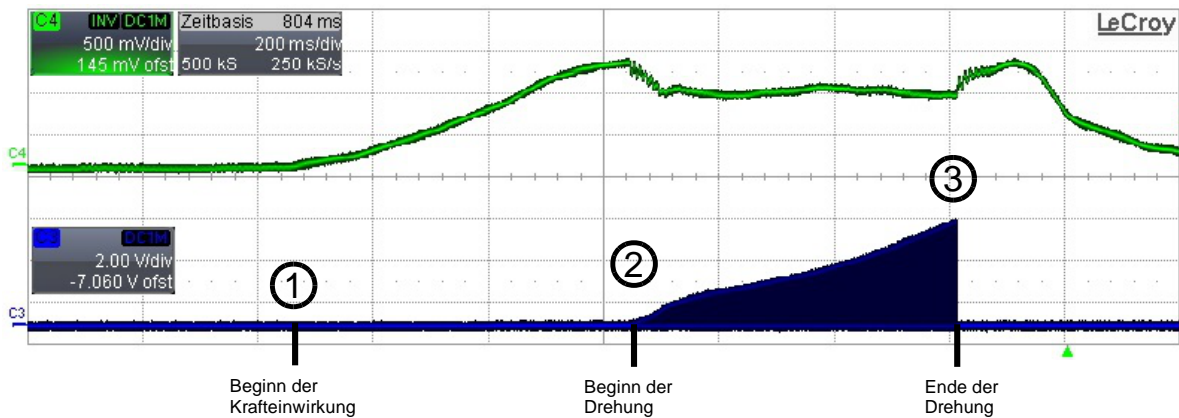


Abbildung 6.5: Einfluss der Motorspannung auf das Drehmoment. Kanal C₃ stellt den Verlauf der Motorspannung dar, die am DA-Wandler anliegt. Kanal C₄ stellt den Verlauf der zum Drehmoment proportionalen Spannung des Drehmomentsensors dar. Bei 1 beginnt die Kräfteinwirkung auf den Drehknopf. Punkt 2 zeigt den Beginn der Drehung, bei 3 ist die maximale Motorspannung erreicht. Die Motorspannung wird dann abgeschaltet.

c) Diskussion

Der Piezomotor ist selbsthemmend. Aus diesem Grund wird ein bestimmtes Drehmoment benötigt, um den Drehknopf zu drehen. In Punkt 2 beginnt die Drehung und die Motorspannung steigt proportional zum gedrehten Winkel. Zwischen den Punkten 2 und 3 ist kein Anstieg des Drehmoments zu erkennen. Das entstehende Drehmoment ist folglich nicht proportional zur angelegten Motorspannung. Aus diesem Grund ist eine Regelung des Drehmoments nicht möglich.

6.3 Steuerung für virtuelle Rasterung

Statt einer Drehmomentregelung mit virtueller Rasterung wurde eine Steuerung erfolgreich implementiert. Diese Steuerung simuliert die in Kapitel 2.2.2 beschriebene Rasterung. Die Rasterung, welche in beide Richtungen funktioniert, ist beim Drehen des Drehknopfs deutlich spürbar. Wenn der Drehknopf losgelassen wird, bevor der nächste Rastpunkt erreicht ist, dreht sich der Drehknopf in die Position des letzten Rastpunkts zurück. Beim Zurückdrehen wird nicht die maximale Spannung an den Piezomotor angelegt, weil der Piezomotor nicht rechtzeitig abbremst und einen Rastpunkt überspringen könnte. Die Anzahl der Rastpunkte ist zur Laufzeit veränderbar. In Abhängigkeit der Schalterstellung können 4, 5, 8, 16, 20 oder 40 Rastpunkte pro Umdrehung gewählt werden. Die Konfigurationsmöglichkeiten können zur Compile-Zeit frei definiert werden.

6.4 Messungen mit Drehzahlregelung

Für die Drehzahlregelung wurde zuerst eine Voruntersuchung durchgeführt, bei der geklärt werden soll, ob die Drehzahl abhängig von der angelegten Spannung ist. Anschließend wurde ein reiner P-Regler implementiert und Messungen durchgeführt. Die Messungen sollen den Drehzahlverlauf bei einer konstanten Spannung und die Reaktion auf ein angelegtes Bremsmoment zeigen. Der Einfluss der Regelung auf den zeitlichen Verlauf der Motordrehzahl wird untersucht. Anschließend wurde der P-Regler modifiziert, und die Messung des Drehzahlverlaufs erneut aufgenommen. Die Erläuterung zur genauen Implementierung befindet sich in Kapitel 5.1.6.

6.4.1 Voruntersuchung

a) Durchführung

Für die Voruntersuchung wurde die Drehzahl bei den vier Motorspannungen 2 V, 3 V, 4 V und 4,5 V ermittelt. Diese Motorspannung wird vom DA-Wandler erzeugt, vom Piezoverstärker mit dem Faktor 30 verstärkt und an den Piezomotor angelegt. Die Anzahl der Umdrehungen nach 20 s wurde gezählt und mit drei multipliziert. Der Drehzahlverlauf wurde aufgezeichnet.

b) Ergebnisse

Abbildung 6.6 zeigt die gemessenen Drehzahlen sowie die Abhängigkeit der Drehzahl von der angelegten Spannung.

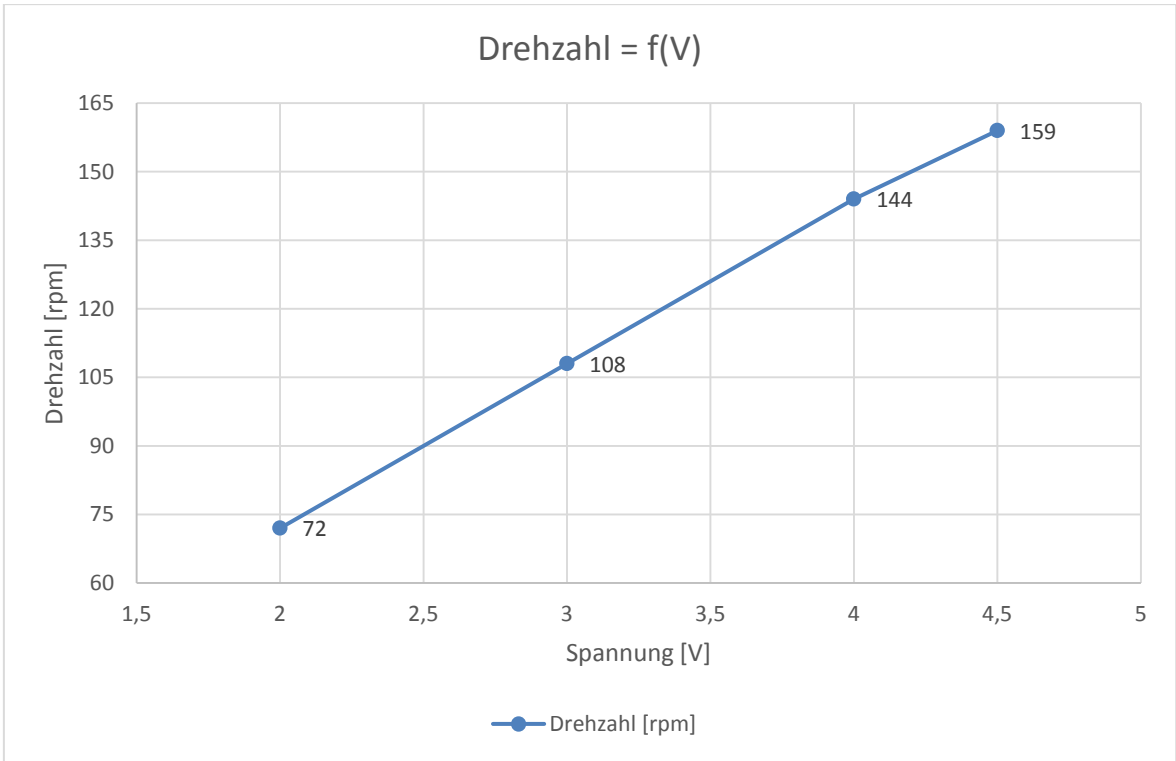


Abbildung 6.6: Abhängigkeit der Drehzahl von der angelegten Spannung

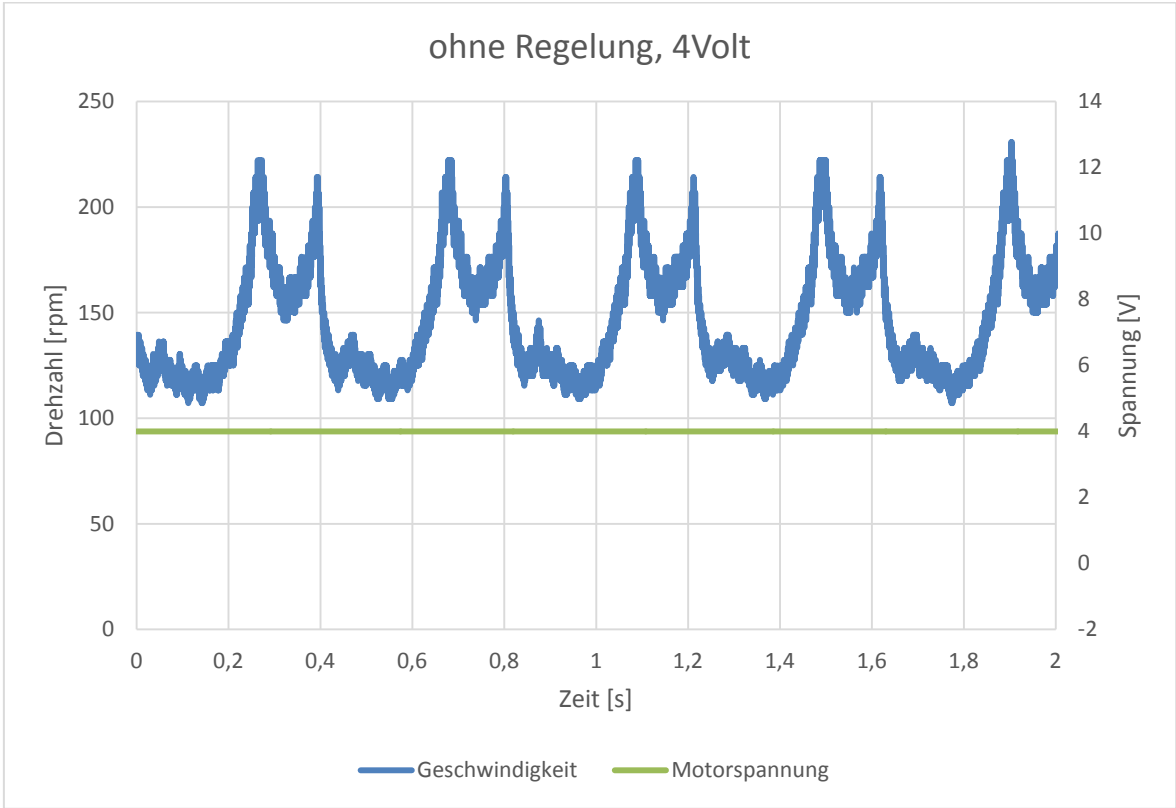


Abbildung 6.7: Verlauf der Drehzahl bei konstanter Motorspannung von 4 V

Der Drehzahlverlauf bei 4 V ist in Abbildung 6.7 zu sehen. Der Piezomotor besitzt einen unrunder Verlauf. Die Drehzahl schwankt bei der Messung mit 4 V im Verlauf einer Drehung zwischen 107 rpm und 222 rpm.

Die Diagramme der weiteren Messungen befinden sich in Anhang D.

c) Diskussion

Die Drehzahl ist abhängig von der angelegten Motorspannung. Ein linearer Zusammenhang ist hierbei zu erkennen. Somit kann eine Drehzahlregelung implementiert werden.

6.4.2 Messungen mit P-Regler

a) Durchführung

Ein P-Regler zur Regelung der Drehzahl des Piezomotors wurde implementiert. Der Proportionalbeiwert sowie die vorgegebene Drehzahl wurden variiert. Die vorgegebenen Drehzahlen sind die in Kapitel 6.4.1 ermittelten Drehzahlen. Außerdem wurde eine Bremsspannung (14 V) an die Hysteresebremse angelegt, um den Piezomotor zu belasten und eine Störung zu induzieren. Der Verlauf der Drehzahl vor und nach dem Anlegen des Bremsmoments wurde aufgenommen.

b) Ergebnisse

In Abbildung 6.8 und Abbildung 6.9 sind die Messungen bei der Solldrehzahl 108 rpm zu sehen. Der Proportionalbeiwert wurde variiert. In Abbildung 6.10 wurde die Solldrehzahl auf 144 rpm erhöht.

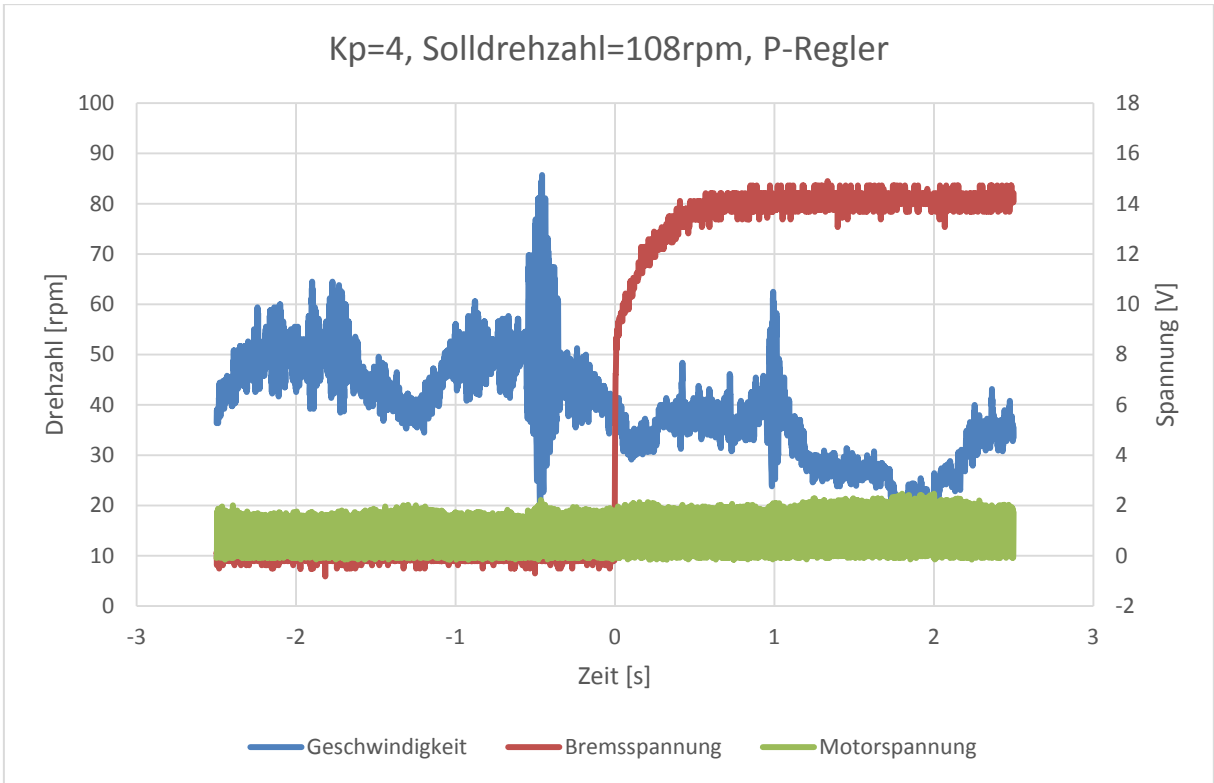


Abbildung 6.8: Drehzahlverlauf der Drehzahlregelung mit $K_p=4$ und Solldrehzahl=108rpm

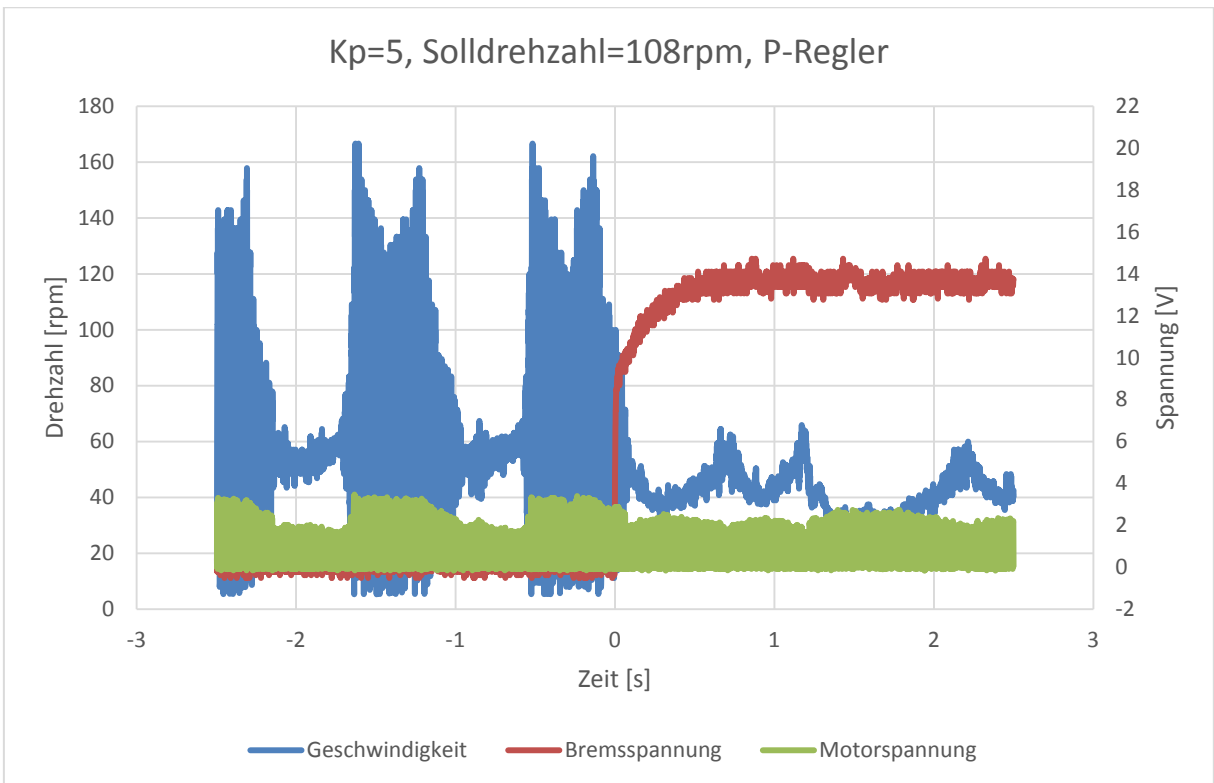


Abbildung 6.9: Drehzahlverlauf der Drehzahlregelung mit $K_p=5$ und Solldrehzahl=108rpm

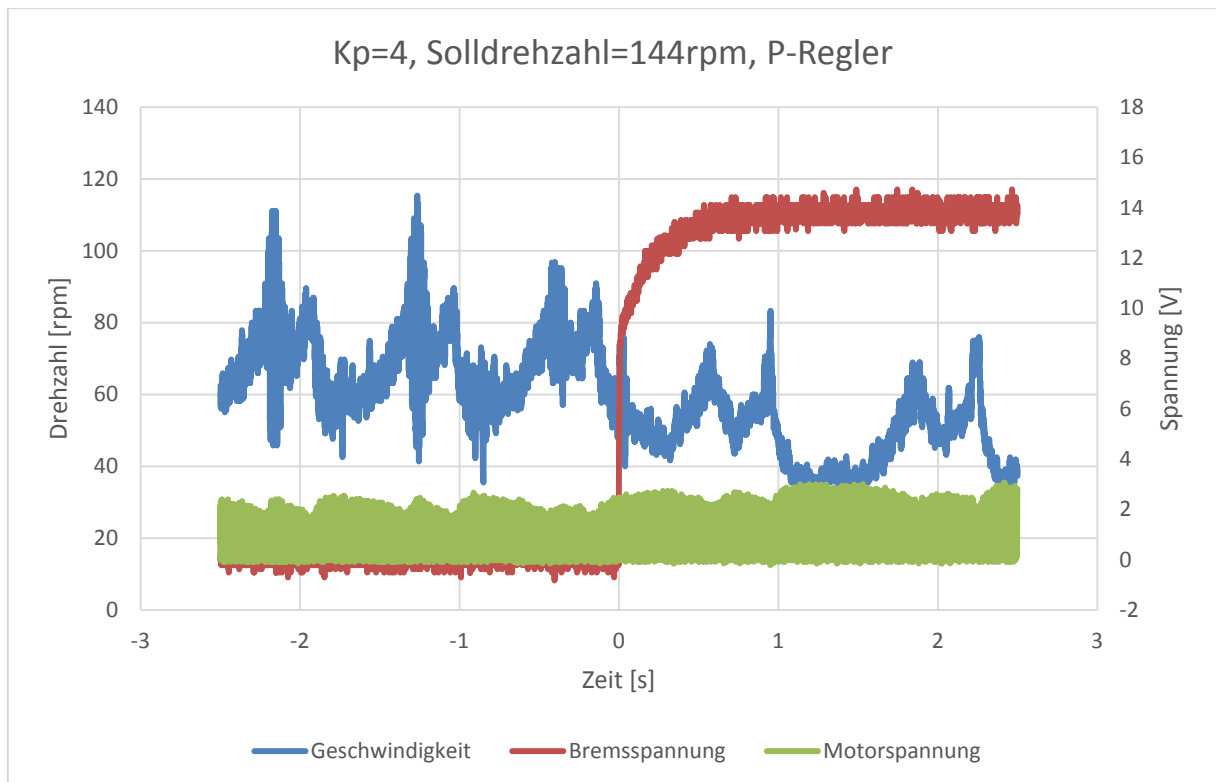


Abbildung 6.10: Drehzahlverlauf der Drehzahlregelung mit $K_p=4$ und Solldrehzahl=144rpm (ermittelt bei 4 V)

c) Diskussion

In Abbildung 6.9 ist zu erkennen, dass die Drehzahl zwischen minimaler und maximaler Drehzahl alterniert. Die Regelung ist bei der Messung mit $K_p=5$ und 108 rpm nicht stabil. Der Piezomotor hat angefangen zu vibrieren. Mit $K_p=4$ (Abbildung 6.8) zeigt sich ein stabileres Verhalten. Die Drehzahl gerät nur einmal kurz vor dem Anlegen des Bremsmoments ins Schwingen. Nach dem Anlegen des Bremsmoments ist die Regelung allerdings stabil. Das liegt wahrscheinlich an der geminderten Drehzahlschwankung.

Die Drehzahlregelung mit der Solldrehzahl 144 rpm (ermittelt bei 4 V) und $K_p=4$ ist stabil. Der positive Einfluss der Regelung ist die geringere Drehzahlschwankung. Sie schwankt vor dem Anlegen des Bremsmoments zwischen 35 rpm und 115 rpm (zum Vergleich ohne Regelung: 107 rpm und 222 rpm).

Die mittlere Drehzahl nach dem Anlegen des Bremsmoments beträgt 54 rpm. Mit dem P-Regler ergibt sich demzufolge eine bleibende Regelabweichung von 90 rpm, was einem relativen Fehler von 62,5% entspricht. Vor dem Anlegen des Bremsmoments erreicht die Drehzahl nie die vorgegebene Drehzahl. In Abbildung 6.7 ist zu sehen, dass sich die Drehzahl ca. 0,2 s über der Solldrehzahl befindet. Der P-Regler ermittelt dafür eine negative Regelabweichung und nach der Stellgrößenbegrenzung wird als Stellgröße eine

Spannung von 0 V ausgegeben. Das bremst den Motor derart stark ab, dass die vorgegebene Drehzahl nicht erreicht wird.

Die Drehzahlregelung mit reinem P-Regler liefert keine zufriedenstellenden Ergebnisse. Die berechneten Stellgrößen werden auf Werte zwischen 0 V und 5 V beschränkt. Da der Piezomotor sich erst ab einer Spannung von 1,6 V dreht ohne durch Reibung zu stoppen, ist eine Stellgröße unter diesem Wert nicht sinnvoll. Die Drehzahl verläuft zwischen 2 V und 4,5 V proportional zur angelegten Spannung. Zwischen 1,6 V und 2 V ist die Linearität durch den höheren Einfluss der Reibung nicht mehr gegeben. Der Piezomotor stoppt bei einer Spannung unter 1,6 V abrupt ab. Aus diesem Grund sollte die untere Schranke der Stellgröße 1,6 V betragen.

Um die Regelabweichung zu minimieren, wird ein Offset zu der berechneten Stellgröße addiert. Als Offset wird für die gewählte Solldrehzahl der entsprechende Wert aus Abbildung 6.6 gewählt.

6.4.3 Messungen mit modifiziertem P-Regler

a) Durchführung

Der P-Regler wurde wie in Kapitel 6.4.2 in der Diskussion beschrieben modifiziert. Der Proportionalbeiwert wurde mit $K_p=4$ gewählt.

b) Ergebnisse

Abbildung 6.11 zeigt den unregelmäßigen Verlauf der Drehzahl bei einer konstanten Spannung von 3 V. In Abbildung 6.12 sind die Messergebnisse für eine Drehzahlregelung mit einer vorgegebenen Drehzahl von 108 rpm zu sehen. Die Diagramme der weiteren Messungen befinden sich im Anhang D. Die Schwankung der Drehzahl ist auf den unregelmäßigen Verlauf in der Drehung des Piezomotors zurückzuführen.

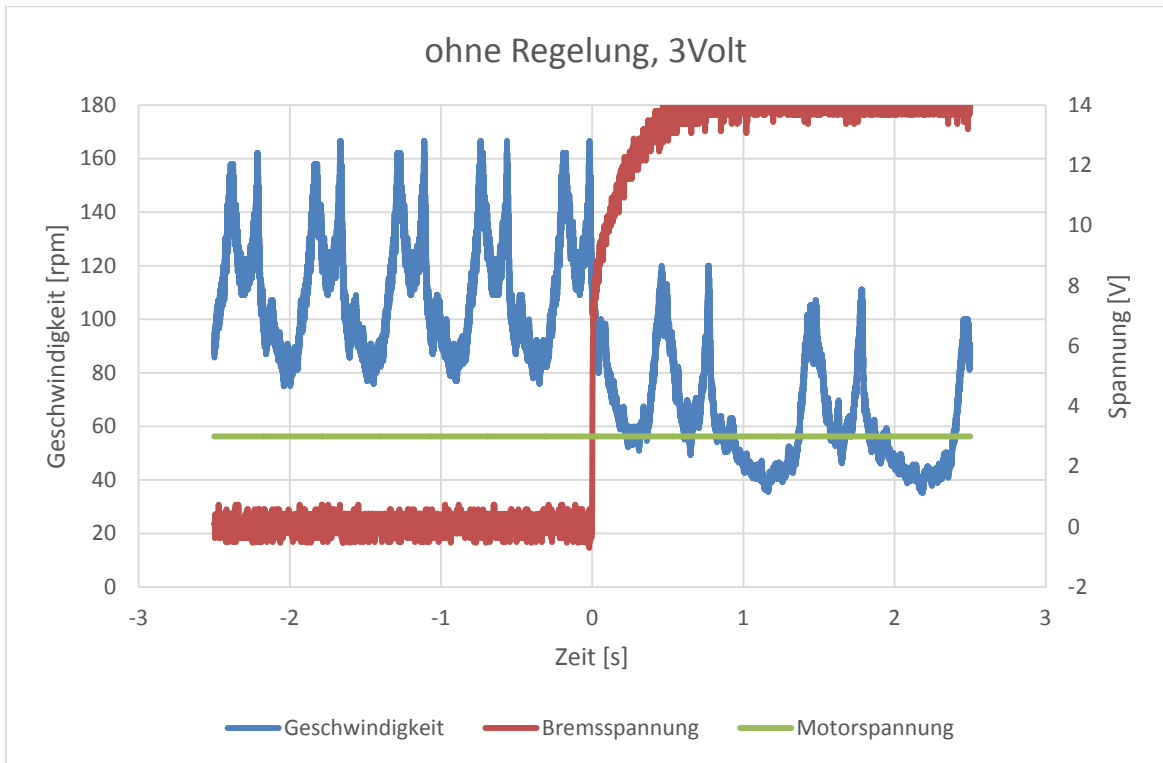


Abbildung 6.11: Drehzahl des Piezomotors bei konstanter Spannung. Zum Zeitpunkt 0 s wird eine Spannung an die Hysteresebremse angelegt.

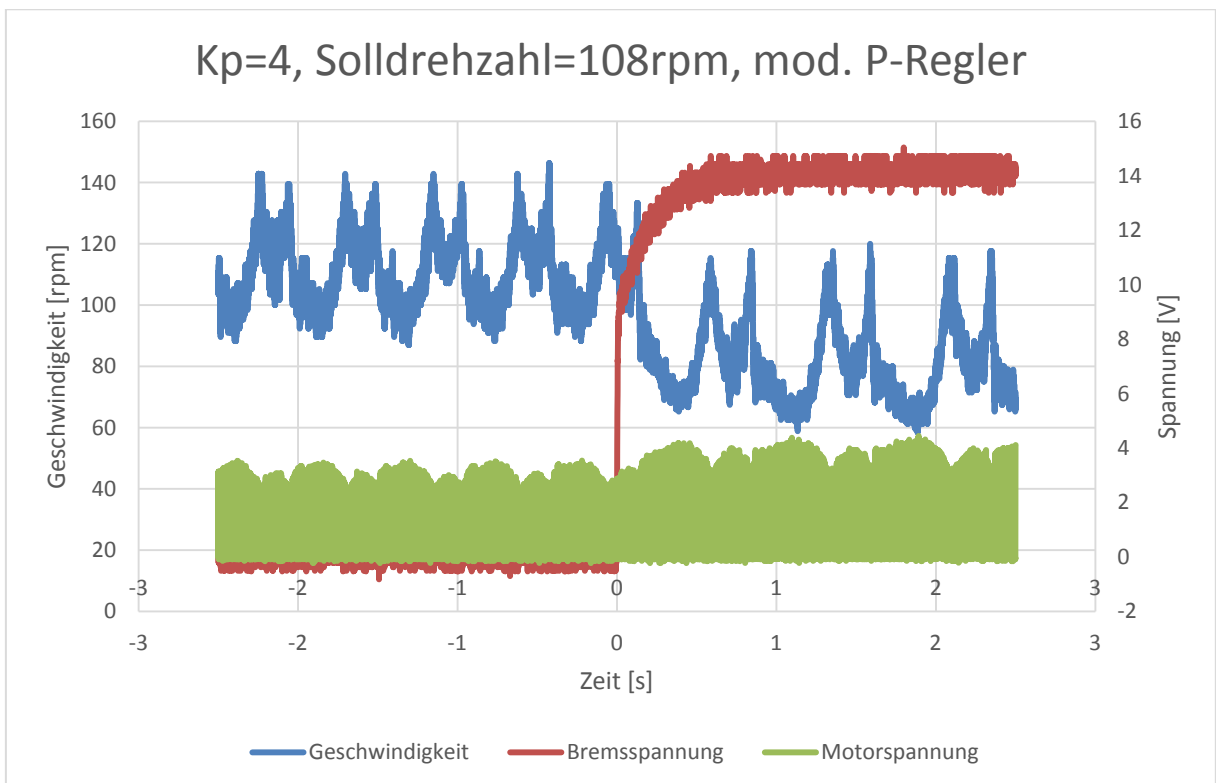


Abbildung 6.12: Drehzahlverlauf der Drehzahlregelung mit $K_p=4$ und Söldrehzahl=108rpm (ermittelt bei 3 V). Zum Zeitpunkt 0 s wird eine Spannung an die Hysteresebremse angelegt.

In Tabelle 6.2 sind die Ergebnisse der Messungen dargestellt. Die minimale und maximale Drehzahl vor dem Anlegen der Bremsspannung wurde ermittelt. Nach dem Anlegen der Bremsspannung wurde ebenfalls die minimale und maximale Drehzahl nach einer Einschwingphase ermittelt.

ohne Regelung (Einheiten in rpm)						
Spannung/ vorgegebene Drehzahl	Min. v. B.	Max. v. B.	Differenz	Min. n. B.	Max. n. B.	Differenz
2 V/72 rpm	48	113	66	20	85	65
3 V/108 rpm	75	167	92	35	120	85
4 V/144 rpm	107	222	115	56	150	94
4.5 V/159 rpm	122	250	128	68	188	119
mit Regelung						
2 V/72 rpm	56	105	50	35	79	44
3 V/108 rpm	87	143	56	58	120	62
4 V/144 rpm	111	200	89	72	158	86
4.5 V/159 rpm	125	214	89	75	194	119

Tabelle 6.2: Ergebnisse der Messungen mit verschiedenen Spannungen. Die minimale und maximale Drehzahl vor (v. B.) und nach (n. B.) dem Anlegen der Bremsspannung wurde ermittelt und die Differenz berechnet.

Tabelle 6.3 beinhaltet die Mittelwerte nach dem Anlegen der Bremsspannung und die Regelabweichung. Die Regelabweichung wird aus der Differenz zwischen der vorgegebenen Drehzahl und dem Mittelwert gebildet.

Spannung / vorgegebene Drehzahl	ohne Regelung (Einheiten in rpm)		mit Regelung	
	Mittelwert n. B.	Regelabweichung	Mittelwert n. B.	Regelabweichung
2 V/72 rpm	53	19	59	13
3 V/108 rpm	80	28	89	19
4 V/144 rpm	112	32	119	25
4.5 V/159 rpm	128	31	129	30

Tabelle 6.3: Mittelwerte der Drehzahl nach (n. B.) dem Anlegen der Bremsspannung. Die Regelabweichung ist die Differenz zwischen vorgegebener Drehzahl und dem Mittelwert.

c) Diskussion

Die Drehzahlregelung hat zwei positive Einflüsse auf den Betrieb des Piezomotors. Durch die Drehzahlregelung wird der unrunde Verlauf teilweise ausgeglichen. Die Differenz zwischen maximaler und minimaler Drehzahl sinkt. In Abbildung 6.13 ist die Differenz aus Tabelle 6.2 vor dem Anlegen der Bremsspannung grafisch dargestellt. Bei der Messung mit

Solldrehzahl=108 rpm schwankt die Drehzahl um 92 rpm, mit Regelung lediglich noch um 56 rpm.

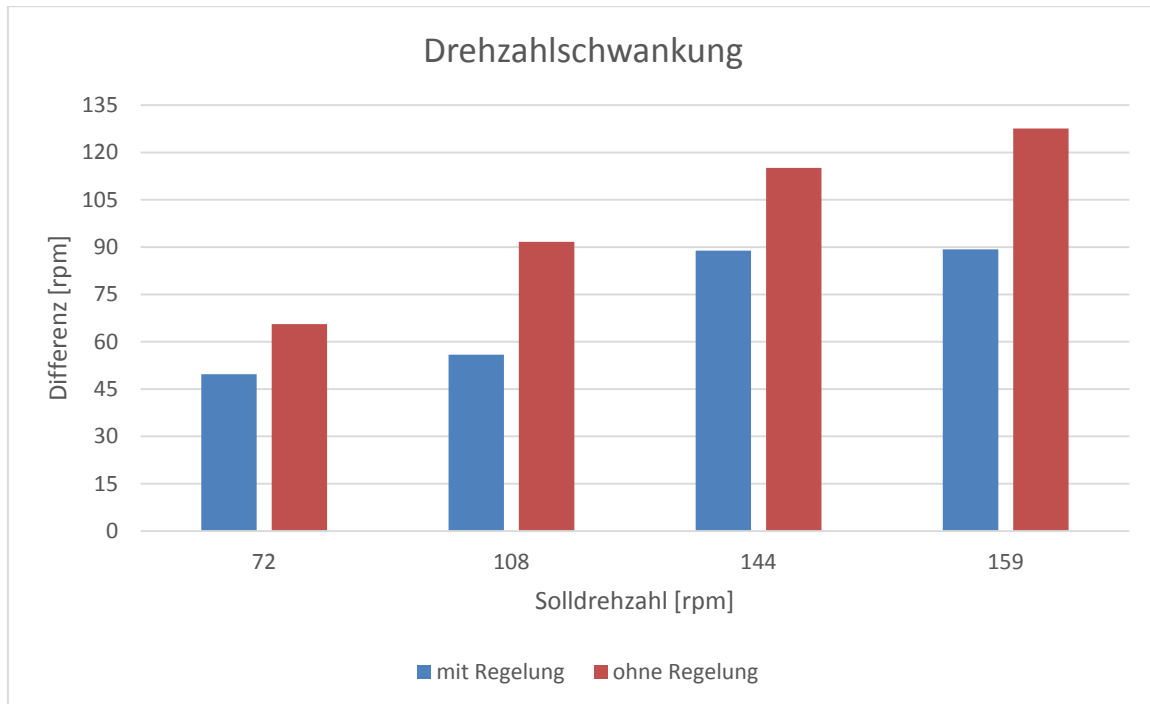


Abbildung 6.13: Drehzahlschwankung, Differenz zwischen maximaler und minimaler Drehzahl

Der zweite positive Einfluss der Drehzahlregelung ist die verringerte Regelabweichung. Um die Regelabweichung zu bestimmen, wurde der Mittelwert der Drehzahl nach dem Anlegen der Bremsspannung ermittelt. Die Regelabweichung ist die Differenz zwischen dem Sollwert und der erreichten mittleren Drehzahl. In Abbildung 6.14 ist die relative Regelabweichung für die einzelnen Messungen dargestellt. Bei der Messung mit Solldrehzahl=108 rpm beträgt die Regelabweichung ohne Regelung 26 %, mit Regelung lediglich 18 %. Im Vergleich dazu lag der Fehler beim reinen P-Regler bei derselben Solldrehzahl bei 62,5 %.

Die Annahme, dass die Regelung bei 4,5 V ihre Grenz erreicht hat, liegt nahe. Der Fehler der Regelabweichung hat nahezu den gleichen Wert wie ohne Regelung. Das liegt daran, dass die Stellgröße (siehe Abbildung E.9 in Anhang D) sich nahezu durchgehend in der oberen Stellgrößenbegrenzung befindet.

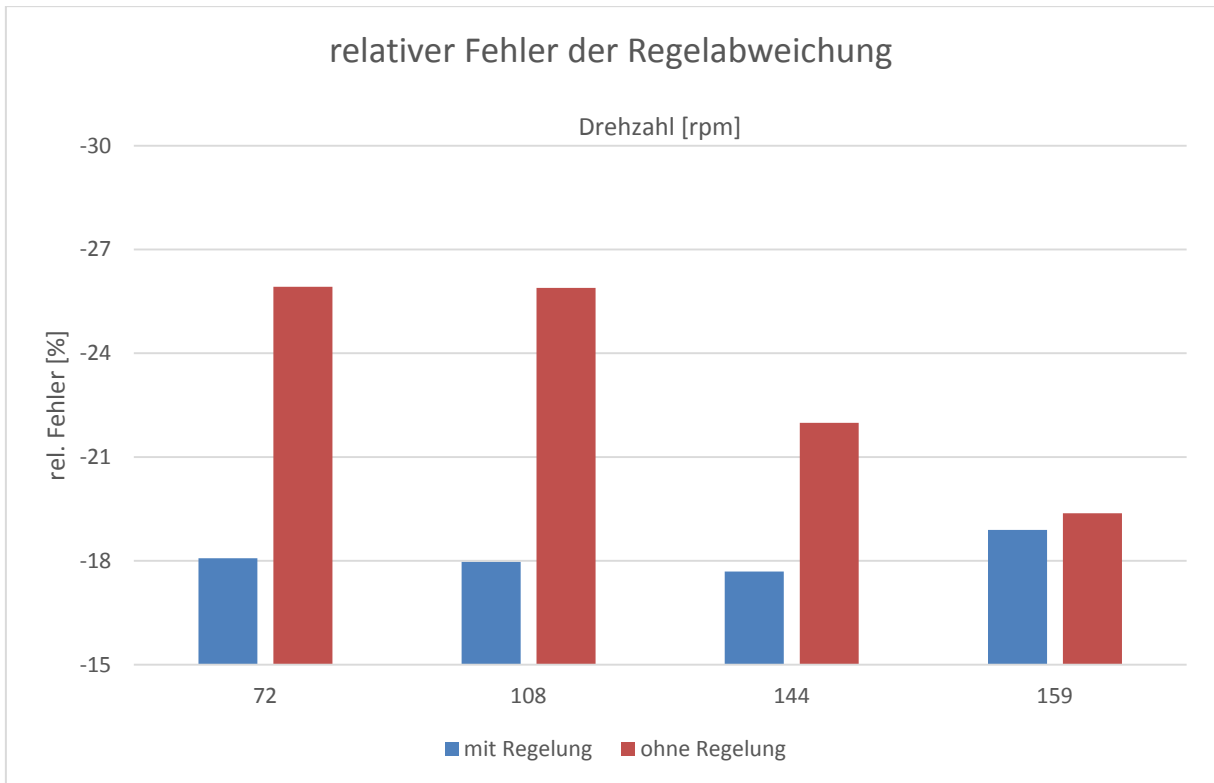


Abbildung 6.14: Vergleich der Regelabweichungen mit und ohne Regelung nach Anlegen der Bremsspannung

6.4.4 Messungen mit PI-Regler

Die Parameter-Auslegung des PI-Reglers musste aus Zeitgründen entfallen.

6.5 Ergebnisse der Hardware-Synthese

Die Ergebnisse der Taktfrequenz und des Speicherplatzes wurden mit ISE 14.6 ermittelt. Als FPGA kam der XC6SLX150 mit Speed-Grade -3 zum Einsatz.

a) maximale Taktfrequenz

Bei Verwendung des P-Reglers ist der Prozess zur Ermittlung des gedrehten Winkels bzw. der Drehzahl der kritische Pfad der Schaltung. Dieser Prozess beschränkt die maximale Taktfrequenz auf 168 MHz. Mit PI-Regler wird die maximale Taktfrequenz auf 88 MHz beschränkt.

b) Speicherplatz

Für das Programm werden insgesamt weniger als 1 % der verfügbaren Speicher- und Logikeinheiten verwendet.

c) Verwendete Pins

Es wurden lediglich Pins verwendet, die ebenfalls in der kostengünstigsten Variante (XC6SLX25) des FPGA vorhanden sind. Bank 0 bleibt komplett unbeschaltet.

Aufgrund der geringen verwendeten Speicher- und Logikeinheiten und der verwendeten Pins, ist ein Einsatz des kostengünstigeren XC6SLX25 für die nächste Platinenversion möglich

7 Zusammenfassung und Ausblick

7.1 Platinen

7.1.1 HACON

a) Zusammenfassung

Eine Platine mit einem FPGA als Reglerplattform wurde entwickelt und teilweise in Betrieb genommen. Das FPGA mit Konfigurationsspeicher, die Spannungsversorgung, der Oszillator, der Voltage Monitor sowie die LEDs und Steckverbinder mit Verbindung zur SiKo-Platine wurden in Betrieb genommen. Die Inbetriebnahme des AD-Wandlers, des UART zu USB Wandlers, der SRAM- und Flash-Speicher sowie die Ansteuerung der CONBOX-Platine wurde nicht durchgeführt.

Die entworfene Platine besitzt eine Schnittstelle zur Ansteuerung der analogen SiKo-Platine. Diese beiden Platinen werden verwendet, um einen Piezomotor anzusteuern.

b) Ausblick

Einige Mängel sind beim Einsatz der Platine aufgefallen und sollten bei der nächsten Platinenversion behoben werden. Der RST#-Pin des Voltage-Monitors ist mit zwei Pins des FPGAs verbunden. Die Verbindung zu Pin K19 ist zu entfernen.

Bei der ersten Version der Platine wurden die meisten SMD-Bauteile in der Bauform 0603 bestückt. Der aus einer Standard-Bibliothek verwendete Footprint dieser Bauteile ist jedoch zu klein für die Bauform 0603. Die künftigen Generationen der Platine sind deshalb mit Bauteilen der Bauform 0402 zu bestücken.

In den Ecken der Platine sind Bohrlöcher für Halterungen vorzusehen.

Die Steckverbinder P1 und P3 können zusammengefasst werden. In der ersten Version der Platine ist der Abstand zwischen beiden Steckverbindern etwas zu groß.

Die Beschriftung „RESET“ ist an Taster B2 vorzusehen.

Die in Kapitel 4.1 beschriebenen 0 Ω -Widerstände können entfernt werden. Das Auslesen der Konfigurationsdaten aus dem Konfigurationsspeicher war erfolgreich.

7.1.2 SiKo-Platine

a) Zusammenfassung

Der AD- und DA-Wandler auf der SiKo-Platine sowie die Ansteuerung des Relais zur Drehrichtungsänderung des Piezomotors wurden in Betrieb genommen. Die intrinsische Drehmomentmessung wurde nicht in Betrieb

genommen. Die Spannungen der Effektivwerte und der Phase zwischen Spannung und Strom befinden sich im messbaren Bereich des AD-Wandlers. Eine weitere Untersuchung dieser Werte fand nicht statt.

b) Ausblick

Bei der Inbetriebnahme der SiKo-Platine wurden einige Mängel festgestellt, die in der nächsten Version behoben werden sollten. An allen ICs sind Entkoppelkondensatoren (100 nF) an die Versorgungspins anzubringen. Ebenso sind am Steckverbinder P2 jeweils ein Elektrolytkondensator (10 – 100 μ F) für die Versorgungsspannungen 3,3 V und 5 V vorzusehen.

Die Pins 9, 11, 13, 15, 17, 19 und 21 des Steckverbinders P2 sind mit dem Groundpotential zu verbinden.

Beim Steckverbinder zum Piezomotor wurden einige Pins vertauscht. Pin 1 ist mit Pin 10 des Relais zu verbinden. Die Pins 2 und 3 des Steckverbinders sind miteinander zu verbinden (Signal IM, siehe [1]). Pin 4 ist mit Pin 12 des Relais zu verbinden.

Die nichtkontaktierten Masseflächen sind zu entfernen oder zu kontaktieren sowie die Lötflächen um die Halterungen der BNC-Stecker zu vergrößern.

7.1.3 CONBOX-Platine

Die Ansteuerung der CONBOX-Platine wurde aus Zeitgründen nicht durchgeführt.

7.2 FPGA Hardware-Design

Für das FPGA wurden zwei Betriebsmodi implementiert, die zur Laufzeit über einen Schalter gewechselt werden können.

7.2.1 Haptische Feedback

a) Zusammenfassung

Der erste Modus ist eine Steuerung, die eine virtuelle Rasterung simuliert. An der vom Piezomotor angetriebenen Achse ist ein Drehknopf angebracht. Beim Drehen des Drehknopfs ist die Rasterung spürbar. Die Anzahl der Rastpunkte ist zur Laufzeit veränderbar. Eine Drehmomentregelung wurde nicht implementiert, da das vom Piezomotor erzeugte Drehmoment nicht proportional zu der angelegten Motorspannung ist.

b) Ausblick

Eine Vibration des Piezomotors bei Überschreiten eines kritischen Bereichs an Rastpunkten ist noch zu implementieren. Dabei soll das Ansteuersignal mit einem niederfrequenten Sinus-Signal überlagert werden.

7.2.2 Drehzahlregelung

a) Zusammenfassung

Der zweite Betriebsmodus ist eine Drehzahlregelung. Drei Regler wurden implementiert und Messungen durchgeführt, die zur Überprüfung der Wirksamkeit der Regler dienen.

Die Drehzahl des Piezomotors wird per Encoder ermittelt und daraus mit einem Regler die neue Stellgröße berechnet. Die Ergebnisse mit einem reinen P-Regler ergaben für die Regelabweichung nach dem Anbringen eines Bremsmoments bei einer vorgegebenen Drehzahl von 108 rpm einen Fehler von 62,5 %. Mit dem modifizierten P-Regler ergab sich eine Regelabweichung von 18 %.

Die Drehzahl schwankt bei einer Umdrehung sehr stark. Im unregelmäßigen Verlauf hat sie bei einer Drehzahl von 108 rpm zwischen 75 und 167 rpm geschwankt, was einer Differenz von 92 rpm entspricht. Mit dem modifizierten P-Regler schwankt sie lediglich zwischen 87 und 143 rpm, was einer Differenz von 56 rpm entspricht.

Die Parameter-Auslegung des PI-Reglers musste aus Zeitgründen entfallen.

b) Ausblick

Für weitere Versuche kann der Messaufbau verändert werden, damit der Piezomotor einen runderen Verlauf besitzt und die Drehzahl nicht mehr derart extrem schwankt. Die Parameter des PI-Reglers sind noch zu ermitteln und Messungen durchzuführen.

Um die Dynamik der Regelung zu verbessern, kann die Ermittlung der Geschwindigkeit verändert werden. Als Alternative kann die Zeit zwischen zwei Pegeländerungen der Encodersignale gemessen und daraus die Geschwindigkeit ermittelt werden.

A Literaturverzeichnis

- [1] Wick, Konja. Aufbau einer FPGA-Schaltung für einen Piezo-Ultraschallmotor, BA
- [2] Piezo Motoren, Aktoren und Linearantriebe, Katalog, Physik instrumente
- [3] <http://www.physikinstrumente.de/de/produkte/prdetail.php?sortnr=1000100>
(Stand: 20.12.2013)
- [4] Faa-Jeng Lin, Senior Member, IEEE, Ying-Chih Hung, and Syuan-Yi Chen, FPGA-Based Computed Force Control System Using Elman Neural Network for Linear Ultrasonic Motor, Paper
- [5] Jian-Shiang Chen , In-Dar Lin, Toward the implementation of an ultrasonic motor servo drive using FPGA, Paper
- [6] Moksoon Jang, Francis Dawson, George Bailak, Control System for Multiple Joint, Paper Robotic Arm Powered by Ultrasonic Motor
- [7] Minghui Zhang, Mantian Li and Lining Sun, A Multi-DOF Ultrasonic Motor Using In-plane Deformation of PZT Elements and Its Driving Circuit
- [8] Altium Designer
- [9] User Guide 380, Spartan-6 FPGA Configuration, 2013
- [10] Nymonix, M25P64, 64 Mbit, low voltage, Serial Flash memory with 75 MHz SPI bus interface, Datenblatt, 2009
- [11] Ideal Power , Ideal Power Ltd Product Specification Document, Datenblatt, Rev.A3
- [12] Texas Instruments, LM2853 3A 550 kHz Synchronous SIMPLE SWITCHER® Buck Regulator, Datenblatt, 2006
- [13] User Guide 393, Spartan-6 FPGA PCB Design and Pin Planning Guide, 2012
- [14] Maxim Integrated, 4-/6-/8-Channel, 16-/14-Bit, Simultaneous-Sampling ADCs, Datenblatt, 2011
- [15] Nymonix, M29W160ET, M29W160EB, Datenblatt. 2008
- [16] Silicon Labs, SINGLE-CHIP USB TO UART BRIDGE, Datenblatt, Rev 1.4
- [17] Dallas Semiconductor, DS1834/A/D, Dual EconoReset with Pushbutton, Datenblatt
- [18] Linear Technology, LTC6905-XXX Series Fixed Frequency SOT-23 Oscillator, Datenblatt, 2005
- [19] Tietze, Schenk. Halbleiter-Schaltungstechnik, 13. Auflage. 2009.
- [20] Texas Instruments, 12-BIT, 4 ANALOG INPUT, 6 MSPS, SIMULTANEOUS SAMPLINGANALOG-TO-DIGITAL CONVERTERS, Datenblatt, 1999
- [21] Piezomechanik GmbH, Betriebsanleitung für Verstärker LE 150/100 EBW, Datenblatt
- [22] intersil, CA3338, CA3338A, CMOS Video Speed, 8-Bit, 50 MSPS, R2R D/A Converters, Datenblatt, 2004
- [23] <http://www.lothar-miller.de/s9y/categories/46-Encoderz> (Stand: 20.12.2013)
- [24] Avago, HEDM-55xx/560x & HEDS-55xx/56xx, Datenblatt, 2012
- [25] <http://www.rn-wissen.de/index.php/Regelungstechnik> (Stand: 10.03.2013)
- [26] <http://de.farnell.com/multicomp/mcrm3haf-10r-v-b/drehschalter-smd-10x10-10-pos-3x3/dp/2082999?Ntt=2082999> (Stand: 20.12.2013)
- [27] <http://de.farnell.com/omron-electronic-components/a6h-8102/schalter-dip-1-2-raster-smd-8-kont/dp/1829492?Ntt=1829492> (Stand: 20.12.2013)

B Schaltplan und Platinenlayout

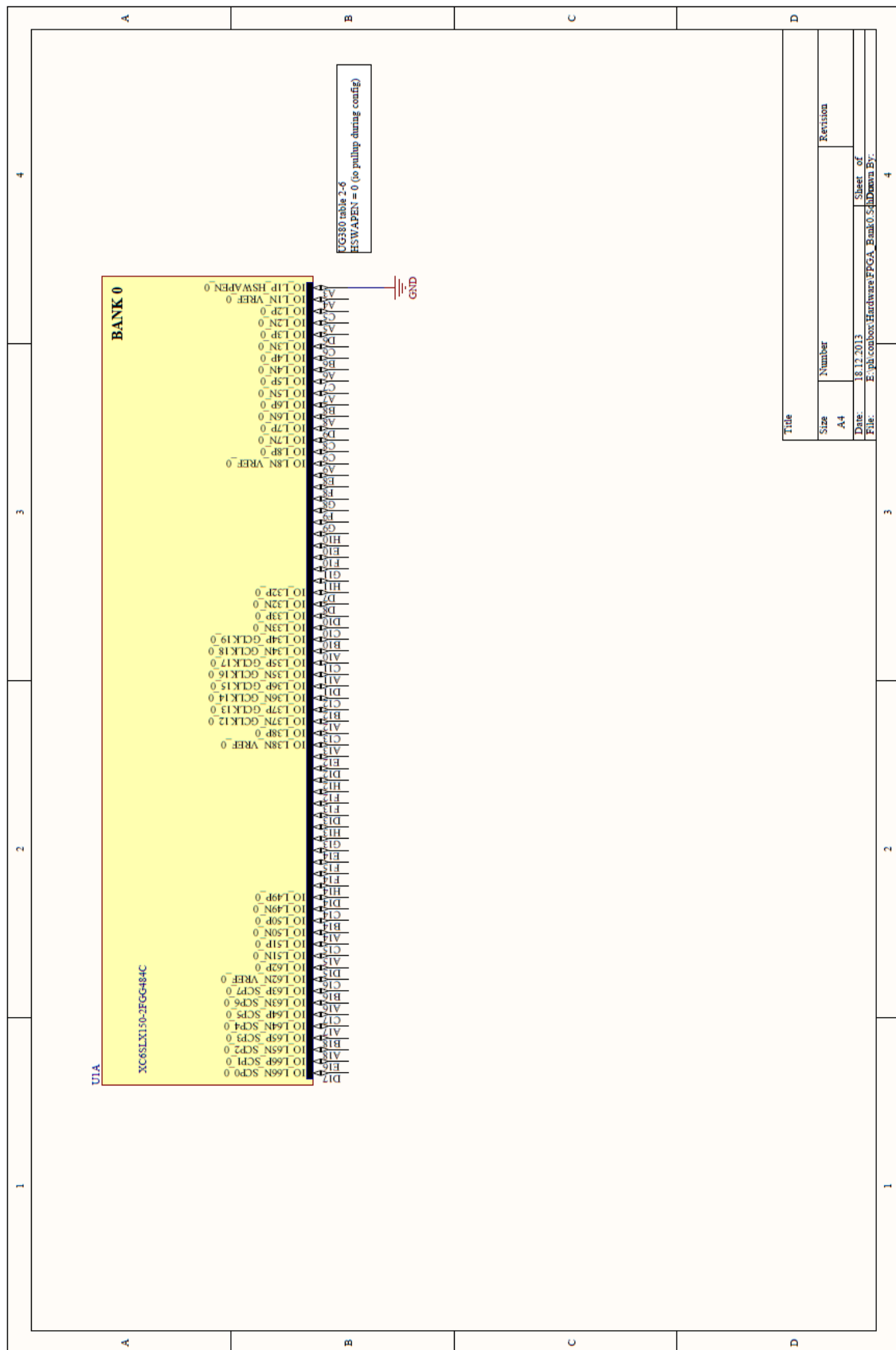


Abbildung B.1: Bank 0 des FPGA

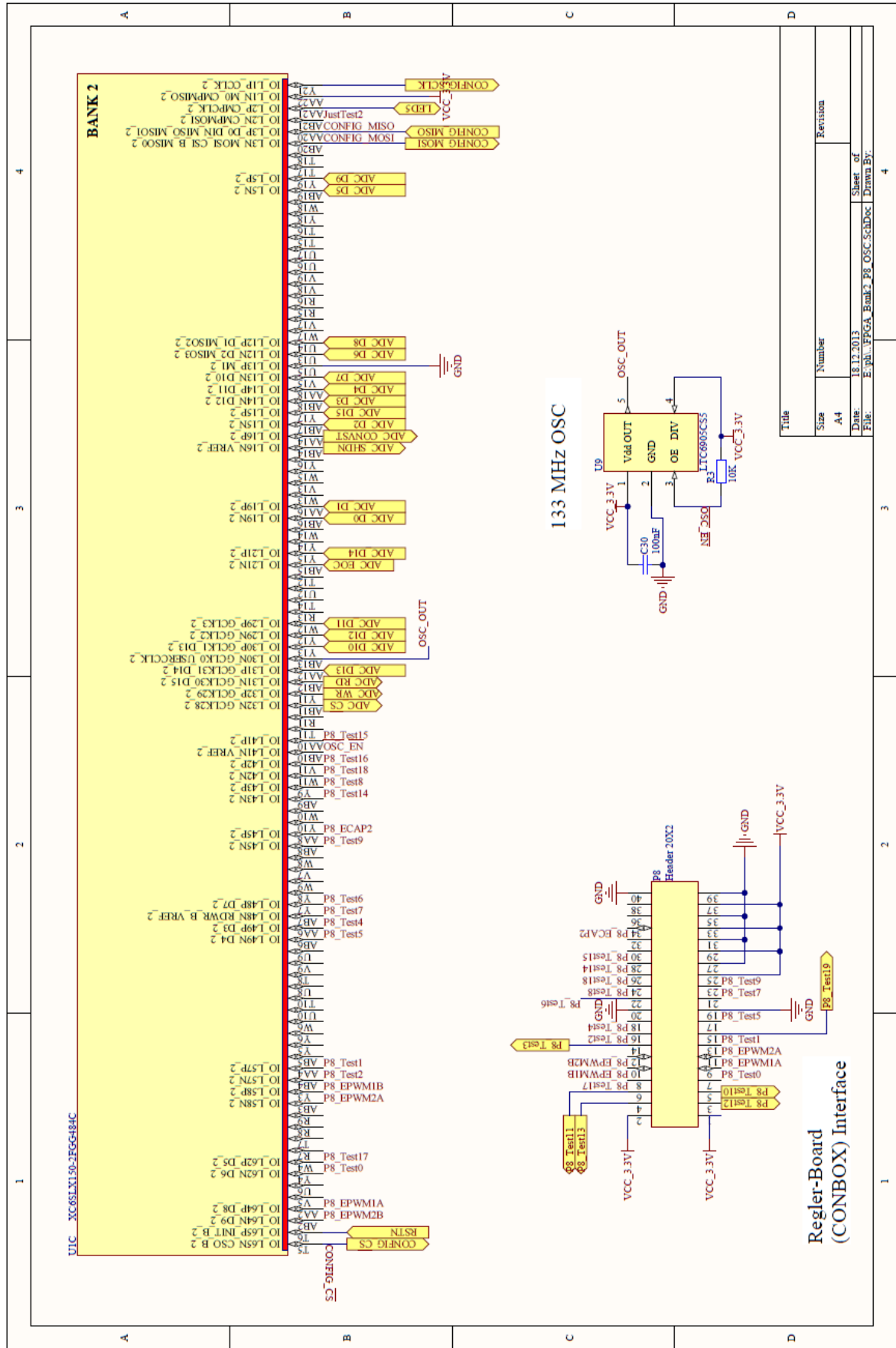


Abbildung B.3: Bank 2 des FPGA und Oszillator

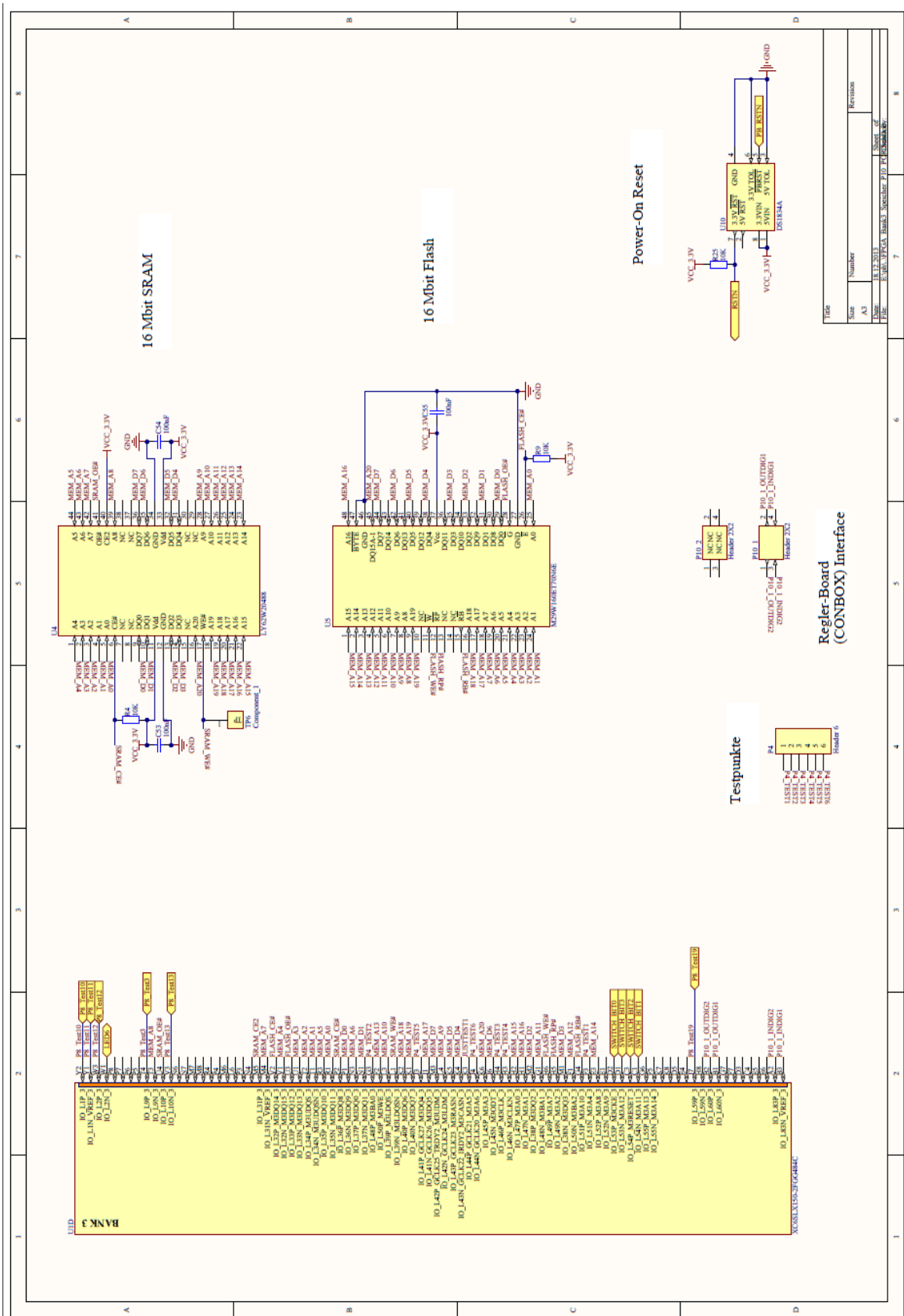


Abbildung B.4: Bank 3 des FPGA, SRAM- und Flashspeicher und Voltage Monitor

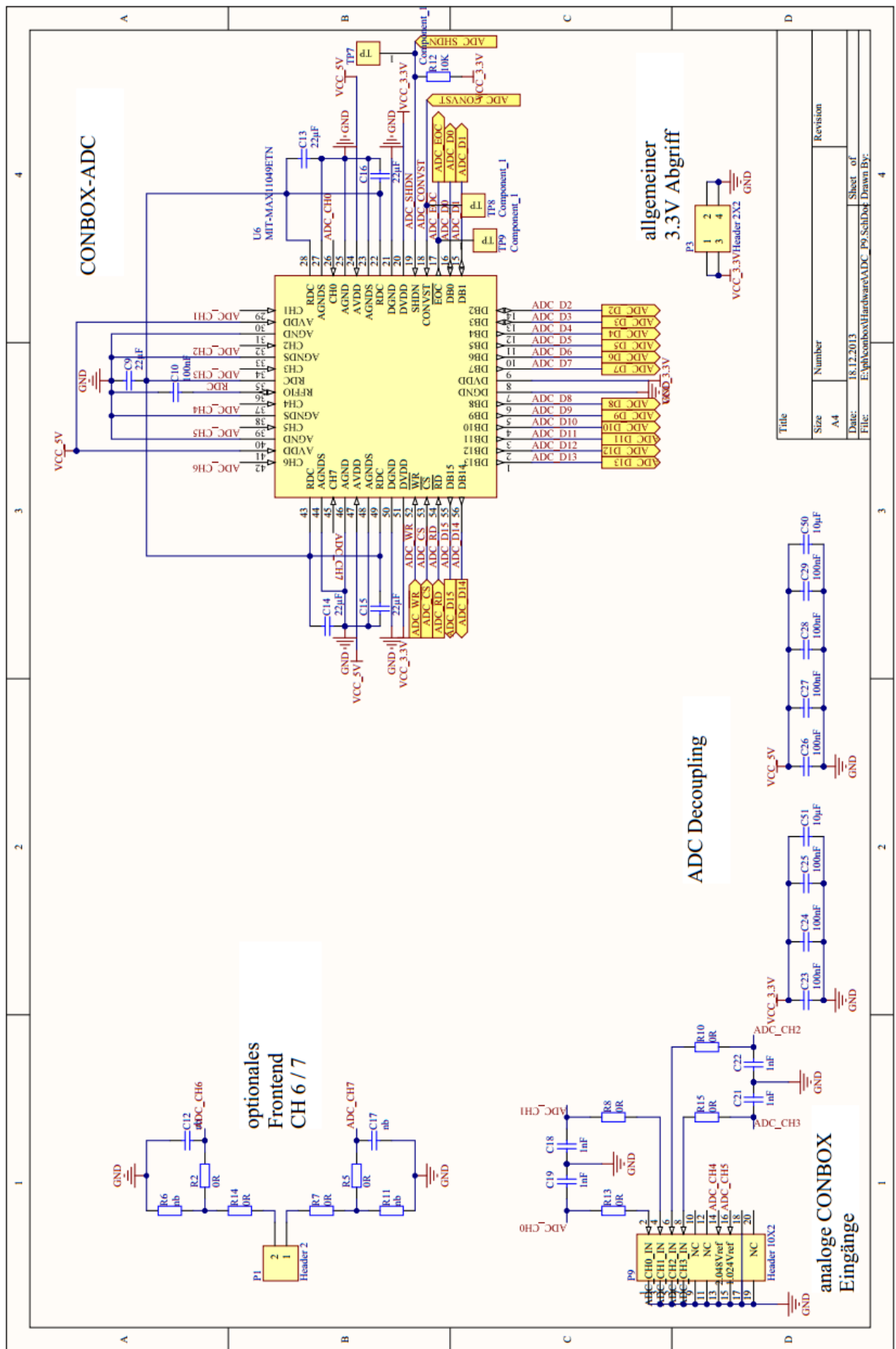
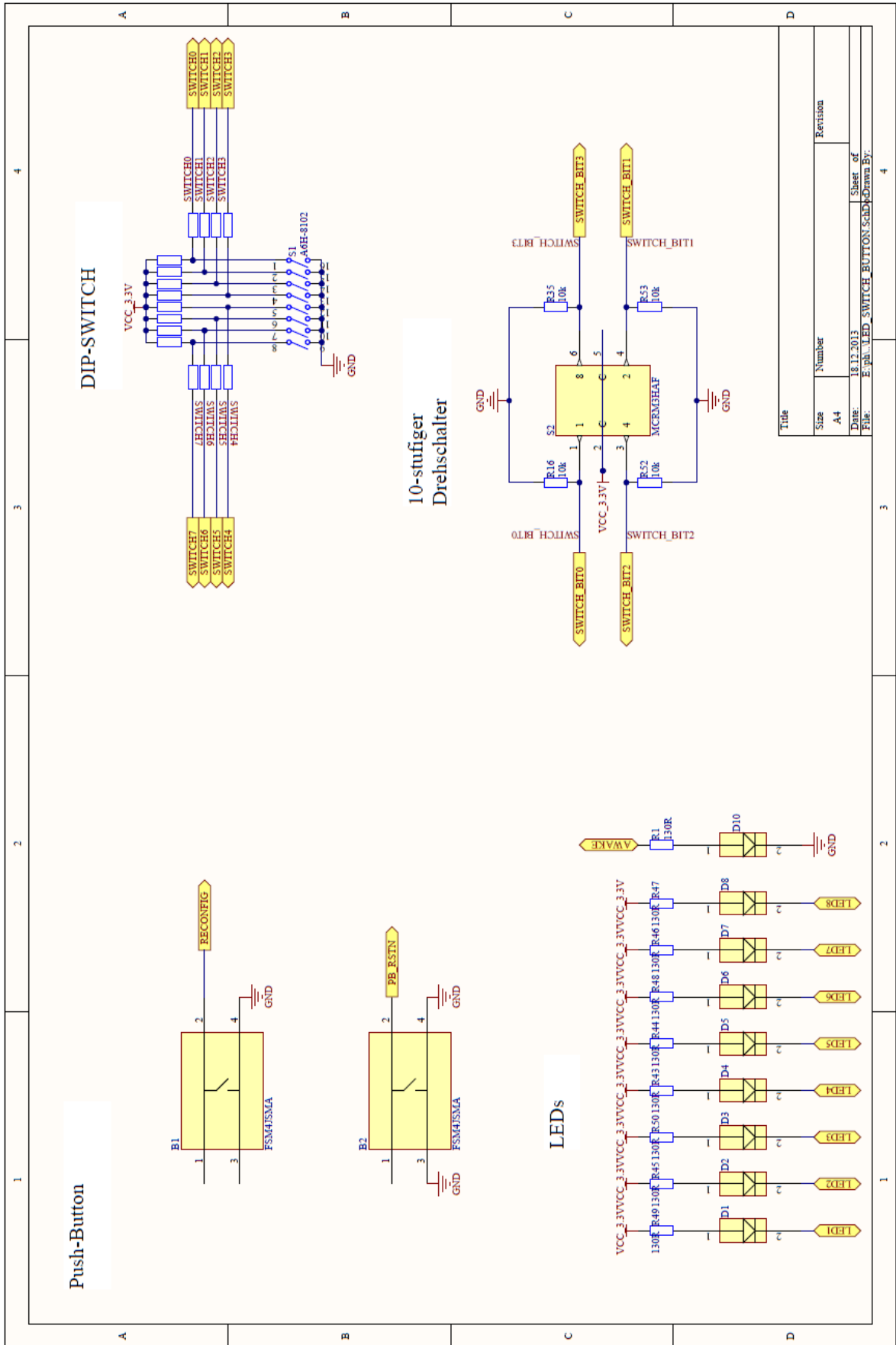


Abbildung B.5: AD-Wandler



Title	Size	Number	Revision
	A4		
Date:	18.12.2013	Sheet of	
File:	Exp1_LED_SWITCH_BUTTON_SchD	Gamma Br.	

Abbildung B.6: LEDs, Dreheswitcher, DIP-Schalter und Taster

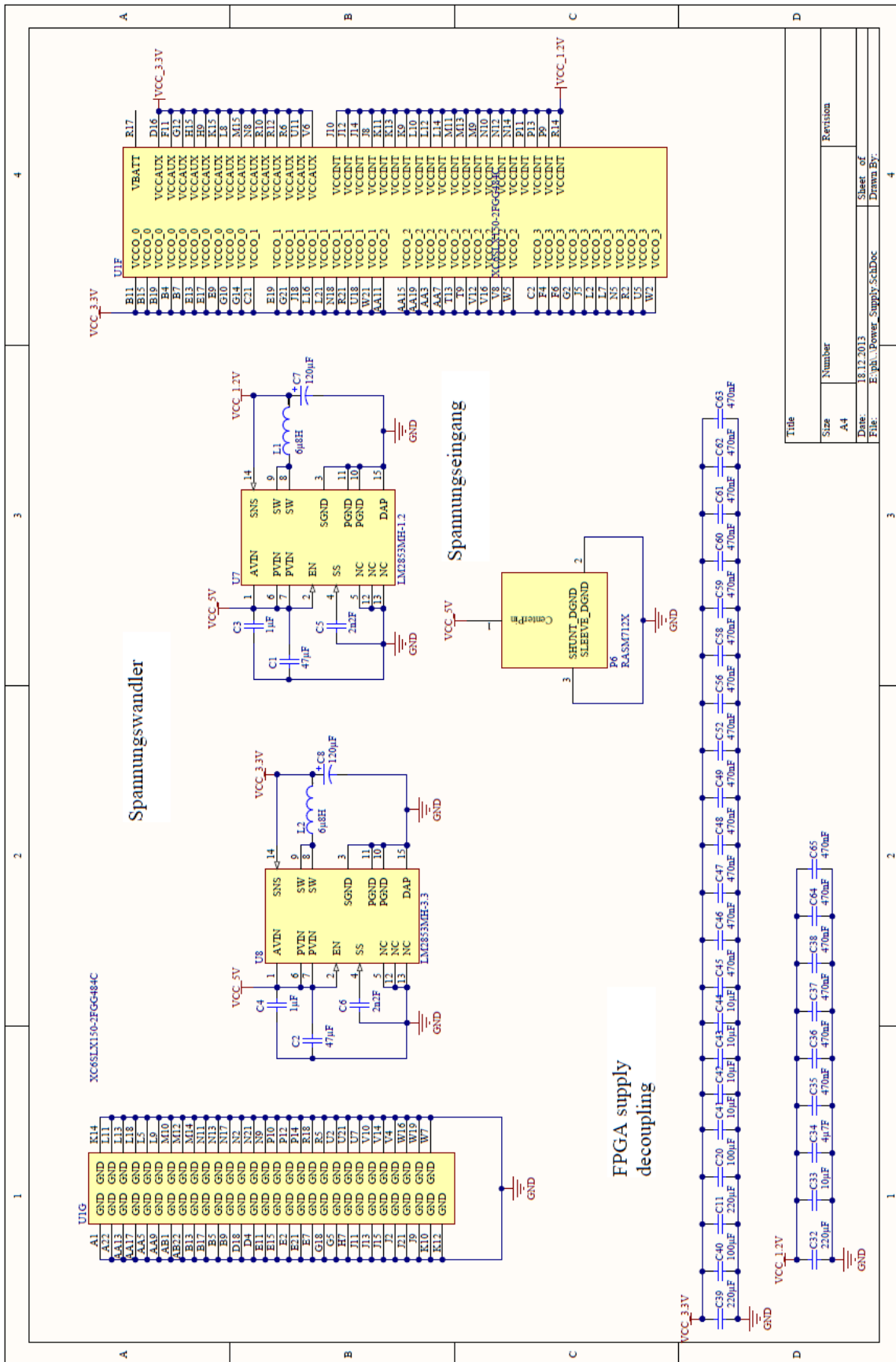


Abbildung B.7: Spannungsversorgung und Entkoppelungskondensatoren

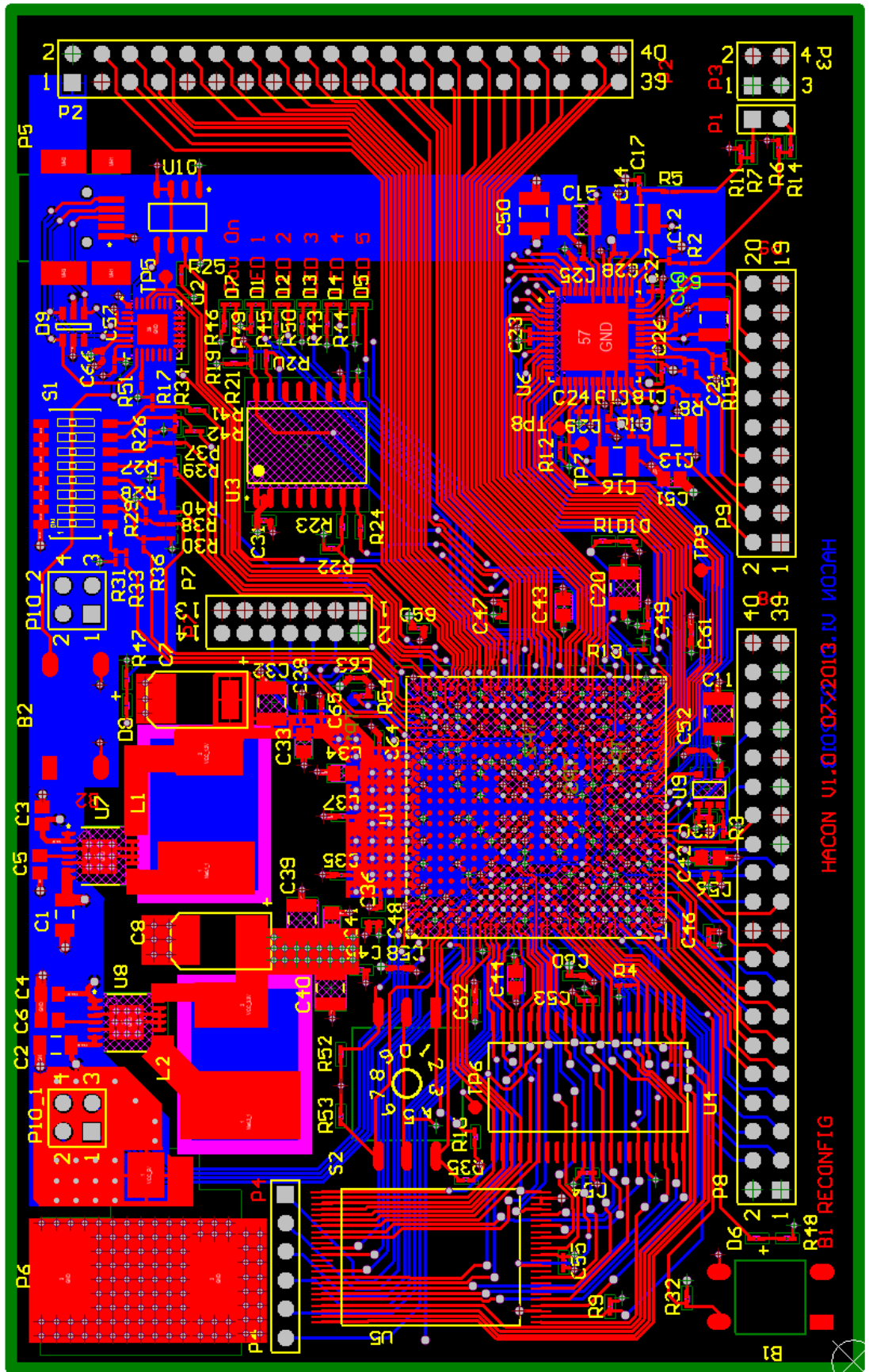


Abbildung B.8: Layout der HACON-Platine

C Steckerbelegungen/Positionen

Steckverbinder	Eagle (x-/y-Achse) [mm]	Altium (x-/y-Achse) [mm]
P8	39,37/7,62	15,24/6,35
P9	83,82/7,62	72,39/6,35
P10_1	20,32/69,85	20,32/67,31
P10_2	66,04/69,85	66,04/67,31

Tabelle C.1: Positionen der Steckverbinder mit Verbindung zur CONBOX-Platine

Die CONBOX-Platine lag nur als Eagle-Datei vor. Die Positionen der Steckverbinder sind in den CAD-Programmen Eagle und Altium Designer verschieden definiert. Bei Eagle ist der Ursprung der Mittelpunkt des Steckverbinders, bei Altium Designer ist es die Position des ersten Pins. Der Koordinatenursprung ist die linke untere Ecke der Platine.

Pinnummer	Signalname	Richtung
P1.1	ADC_CH7	in
P1.2	ADC_CH6	in

Tabelle C.2: Pinbelegung des Steckverbinders P1

Pinnummer	Signalname	Richtung
P3.1	DVDD	
P3.2	GND	
P3.3	DVDD	
P3.4	GND	

Tabelle C.3: Pinbelegung des Steckverbinders P3

Pinnummer	Signalname	Richtung
P4.1	P4_TEST1	
P4.2	P4_TEST2	
P4.3	P4_TEST3	
P4.4	P4_TEST4	
P4.5	P4_TEST5	
P4.6	P4_TEST6	

Tabelle C.4: Pinbelegung des Steckverbinders P4

Pinnummer	Signalname	Richtung
P2.1	Vcc (5Volt)	
P2.2	DVDD	
P2.3	GND	
P2.4	ADC_DATA_AV	in
P2.5	ADC_WR(R/W)	out
P2.6	ADC_CONV_CLK	out
P2.7	ADC_CS0	out
P2.8	ADC_D11	in
P2.9	GND	
P2.10	ADC_D10	in
P2.11	GND	
P2.12	ADC_D9	in
P2.13	GND	
P2.14	ADC_D8	in
P2.15	GND	
P2.16	ADC_D7	in
P2.17	GND	
P2.18	ADC_D6	in
P2.19	GND	
P2.20	ADC_D5	in
P2.21	GND	
P2.22	ADC_D4	in
P2.23	DAC_LE	out
P2.24	ADC_D3	in
P2.25	DAC_B8	out
P2.26	ADC_D2	in
P2.27	DAC_B7	out
P2.28	ADC_D1	in
P2.29	DAC_B6	out
P2.30	ADC_D0	in
P2.31	DAC_B5	out
P2.32	U-ENC-IN-1	in
P2.33	DAC_B4	out
P2.34	U-ENC-IN-2	in
P2.35	DAC_B3	out
P2.36	GND	
P2.37	DAC_B2	out
P2.38	LR	out
P2.39	DAC_B1	out
P2.40	GND	

Tabelle C.5: Pinbelegung des Steckverbinders P2

Pinnummer	Signalname	Richtung
P8.1	3.3Vdig	out
P8.2	3.3Vdig	out
P8.3	P8_Test12	
P8.4	P8_Test13	
P8.5	P8_Test10	
P8.6	P8_Test11	
P8.7	P8_Test0	
P8.8	P8_Test17	
P8.9	EPWM1A	out
P8.10	EPWM1B	out
P8.11	EPWM2A	out
P8.12	EPWM2B	out
P8.13	P8_Test1	
P8.14	P8_Test0	
P8.15	P8_Test3	
P8.16	P8_Test2	
P8.17	P8_Test5	
P8.18	P8_Test4	
P8.19	P8_Test7	
P8.20	DGND	
P8.21	DGND	
P8.22	P8_Test6	
P8.23	P8_Test9	
P8.24	P8_Test8	
P8.25	3.3Vdig	
P8.26	P8_Test18	
P8.27	DGND	
P8.28	P8_Test14	
P8.29	3.3Vdig	
P8.30	P8_Test15	
P8.31	DGND	
P8.32	nc	
P8.33	3.3Vdig	
P8.34	ECAP2	in
P8.35	DGND	
P8.36	nc	
P8.37	3.3Vdig	
P8.38	nc	
P8.39	DGND	
P8.40	DGND	

Tabelle C.6: Pinbelegung des Steckverbinders P8

Pinnummer	Signalname	Richtung
P5.1	USB_VBUS	in
P5.2	USB_D-	inout
P5.3	USB_D+	inout
P5.4	nc	
P5.5	DGND	

Tabelle C.7: Pinbelegung des Steckverbinders P5

Pinnummer	Signalname	Richtung
P6.CENTER	5Vdig	in
P6.SLEEVE	DGND	
P6.SHUNT	DGND	

Tabelle C.8: Pinbelegung des Steckverbinders P6

Pinnummer	Signalname	Richtung
P7.1	DGND	
P7.2	CONFIG_VREF	out
P7.3	DGND	
P7.4	CONFIG_TMS	in
P7.5	DGND	
P7.6	CONFIG_TCK	in
P7.7	DGND	
P7.8	CONFIG_TDO	out
P7.9	DGND	
P7.10	CONFIG_TDI	in
P7.11	DGND	
P7.12	nc	
P7.13	DGND	
P7.14	nc	

Tabelle C.9: Pinbelegung des Steckverbinders P7

Pinnummer	Signalname	Richtung
P10_1.1	OUTDIG2	out
P10_1.2	OUTDIG1	out
P10_1.3	INDIG2	in
P10_1.4	INDIG1	in
P10_2.1	nc	
P10_2.2	nc	
P10_2.3	nc	
P10_2.4	nc	

Tabelle C.10: Pinbelegung des Steckverbinders P10

Pinnummer	Signalname	Richtung
P9.1	AGND	
P9.2	ADC_CH0	in
P9.3	AGND	
P9.4	ADC_CH1	in
P9.5	AGND	
P9.6	ADC_CH2	in
P9.7	AGND	
P9.8	ADC_CH3	in
P9.9	AGND	
P9.10	nc	
P9.11	AGND	
P9.12	nc	
P9.13	AGND	
P9.14	2.048Vref	in
P9.15	AGND	
P9.16	1.024Vref	in
P9.17	AGND	
P9.18	AGND	
P9.19	AGND	
P9.20	nc	

Tabelle C.11: Pinbelegung des Steckverbinders P9

D Pinbelegung FPGA

Pin Nr.	Signalname	Richtung	Pin Nr.	Signalname	Richtung
A20	SWITCH0	INPUT	J22	P2_ADC_D4	INPUT
A21	SWITCH2	INPUT	K19	SRESET	INPUT
B20	SWITCH1	INPUT	K20	LEDs<0>	OUTPUT
B21	SWITCH3	INPUT	K21	P2_DAC_LE	OUTPUT
B22	SWITCH4	INPUT	K22	P2_ADC_D3	INPUT
C19	SWITCH6	INPUT	L19	P2_DAC_Data<6>	OUTPUT
C20	USB_RXD	INPUT	L20	P2_DAC_Data<7>	OUTPUT
C22	USB_RESET	OUTPUT	L22	P2_ADC_D2	INPUT
D19	SWITCH5	INPUT	M20	LEDs<1>	OUTPUT
D20	USB_TXD	OUTPUT	M21	P2_DAC_Data<5>	OUTPUT
D21	P2_ADC_W\R\R(W)	OUTPUT	M22	P2_ADC_D1	INPUT
D22	P2_ADC_DATA_AV	INPUT	N20	P2_DAC_Data<4>	OUTPUT
E20	P2_ADC_C\S\0	OUTPUT	N22	P2_ADC_D0	INPUT
F18	LED_Button2	OUTPUT	P19	LEDs<2>	OUTPUT
F19	SWITCH7	INPUT	P21	P2_DAC_Data<3>	OUTPUT
F21	P2_ADC_CONV_CLK	OUTPUT	P22	P2_U_ENC_IN_1	INPUT
F22	P2_ADC_D11	INPUT	R20	P2_DAC_Data<2>	OUTPUT
G20	P2_ADC_D10	INPUT	R22	P2_U_ENC_IN_2	INPUT
G22	P2_ADC_D9	INPUT	T19	AWAKE	OUTPUT
H20	P2_ADC_D7	INPUT	T20	LEDs<3>	OUTPUT
H21	P2_ADC_D8	INPUT	T21	DIRECTION_LR	OUTPUT
H22	P2_ADC_D6	INPUT	T22	P2_DAC_Data<1>	OUTPUT
J19	PowerLED	OUTPUT	U20	P2_DAC_Data<0>	OUTPUT
J20	P2_ADC_D5	INPUT			

Tabelle D.1: Pinbelegung FPGA, Bank 1

Pin Nr.	Signalname	Richtung	Pin Nr.	Signalname	Richtung
AA10	P8_Test15	INOUT	AB4	P8_Test2	INOUT
AA12	ADC_D13	INPUT	AB6	P8_Test5	INOUT
AA14	ADC_CONVST	OUTPUT	AB7	P8_Test7	INOUT
AA16	ADC_D1	INPUT	AB8	P8_Test9	INOUT
AA18	ADC_D4	INPUT	AB9	P8_Test14	INOUT
AA2	P8_EPWM1A	OUTPUT	T5	CONFIG_C\S\	OUTPUT
AA20	CONFIG_MISO	INPUT	T6	RESETBUTTON	INPUT
AA21	LEDs<4>	OUTPUT	U13	ADC_D6	INPUT
AA22	VCC_3.3V		U14	ADC_D8	INPUT
AA4	P8_Test1	INOUT	U15	GND	
AA6	P8_Test4	INOUT	V11	P8_Test16	INOUT
AA8	P8_ECAP2		V15	ADC_D7	INPUT
AB10	O\S\C\E\N\	OUTPUT	W11	P8_Test18	INOUT
AB11	ADC_C\S\	OUTPUT	W12	ADC_D11	INPUT
AB12	ADC_R\D\	OUTPUT	W4	P8_Test17	INOUT
AB13	clk	INPUT	Y11	ADC_W\R\	OUTPUT
AB14	ADC_SHDN	OUTPUT	Y12	ADC_D12	INPUT
AB15	ADC_E\O\C\	INPUT	Y13	ADC_D10	INPUT
AB16	ADC_D0	INPUT	Y15	ADC_D14	INPUT
AB17	ADC_D2	INPUT	Y17	ADC_D15	INPUT
AB18	ADC_D3	INPUT	Y19	ADC_D9	INPUT
AB19	ADC_D5	INPUT	Y21	CONFIG_SCLK	OUTPUT
AB2	P8_EPWM2B	OUTPUT	Y3	P8_EPWM1B	OUTPUT
AB20	CONFIG_MOSI	OUTPUT	Y4	P8_Test0	INOUT
AB21	JustTest2	INOUT	Y7	P8_Test6	INOUT
AB3	P8_EPWM2A	OUTPUT	Y9	P8_Test8	INOUT

Tabelle D.2: Pinbelegung FPGA, Bank 2

Pin Nr.	Signalname	Richtung	Pin Nr.	Signalname	Richtung
A2	P10_1_INDIG2	INPUT	K5	MEM_D5	INOUT
B1	P10_1_OUTDIG1	OUTPUT	K6	MEM_A20	INOUT
B2	P10_1_OUTDIG2	OUTPUT	L1	SRAM_WE#	OUTPUT
B3	P10_1_INDIG1	INPUT	L3	MEM_A10	INOUT
C1	DREHSWITCH<1>	INPUT	L4	MEM_A9	INOUT
C3	DREHSWITCH<2>	INPUT	M1	MEM_D3	INOUT
D1	DREHSWITCH<3>	INPUT	M2	MEM_D2	INOUT
D2	DREHSWITCH<0>	INPUT	M3	MEM_D7	INOUT
E3	MEM_A14	INOUT	M4	MEM_A7	INOUT
F1	MEM_A12	INOUT	M5	SRAM_CE2	OUTPUT
F2	MEM_A13	INOUT	N1	MEM_D1	INOUT
F3	P4_TEST1	INOUT	N3	MEM_A6	INOUT
G1	MEM_A11	INOUT	P1	MEM_D0	INOUT
G3	P4_TEST2	INOUT	P2	SRAM_CE#	OUTPUT
G4	FLASH_RB#	OUTPUT	R1	MEM_A0	INOUT
H1	MEM_A16	INOUT	R3	MEM_A5	INOUT
H2	MEM_A15	INOUT	T1	MEM_A1	INOUT
H3	P4_TEST4	INOUT	T2	MEM_A2	INOUT
H4	P4_TEST3	INOUT	T3	MEM_A8	INOUT
H5	FLASH_RP#	OUTPUT	T4	P8_Test3	INOUT
H6	FLASH_WE#	OUTPUT	U1	MEM_A3	INOUT
J1	MEM_A17	INOUT	U3	FLASH_OE#	OUTPUT
J3	P4_TEST5	INOUT	U4	SRAM_OE#	OUTPUT
J4	P4_TEST6	INOUT	V1	MEM_A4	INOUT
J6	MEM_D6	INOUT	V2	FLASH_CE#	OUTPUT
J7	P8_Test19	INOUT	V3	P8_Test13	INOUT
K1	MEM_A19	INOUT	W1	LED_Button1	OUTPUT
K2	MEM_A18	INOUT	W3	Testpin	OUTPUT
K3	JUSTTEST1	INOUT	Y1	P8_Test11	INOUT
K4	MEM_D4	INOUT	Y2	P8_Test10	INOUT

Tabelle D.3: Pinbelegung FPGA, Bank 3

E Messungen Drehzahlregelung

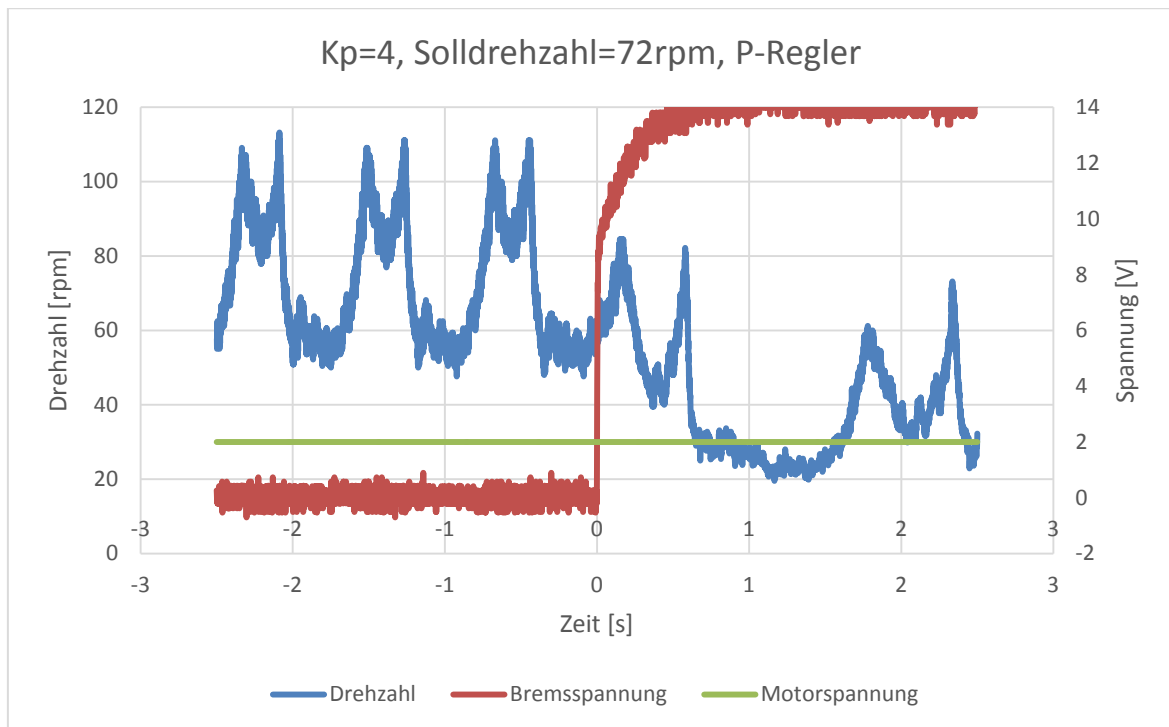


Abbildung E.1: Verlauf der Drehzahl bei konstanter Motorspannung von 2 V

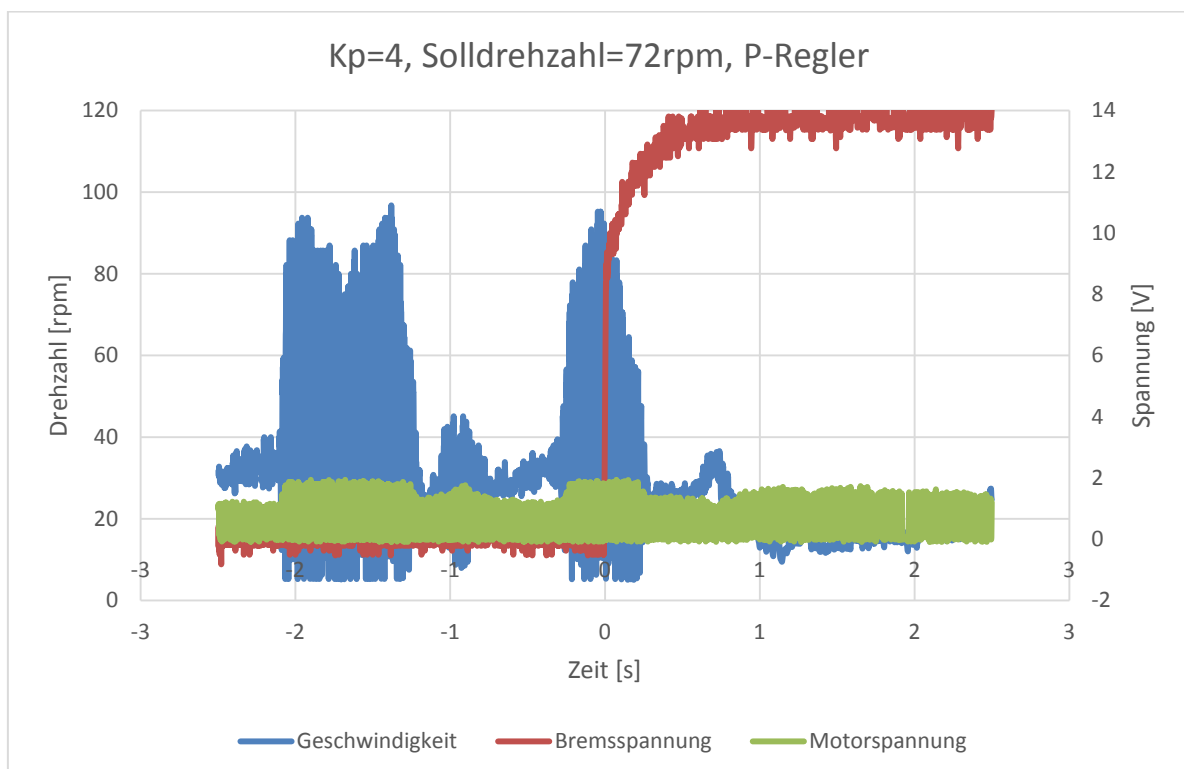


Abbildung E.2: Drehzahlverlauf der Drehzahlregelung mit $K_p=4$ und Solldrehzahl=72rpm

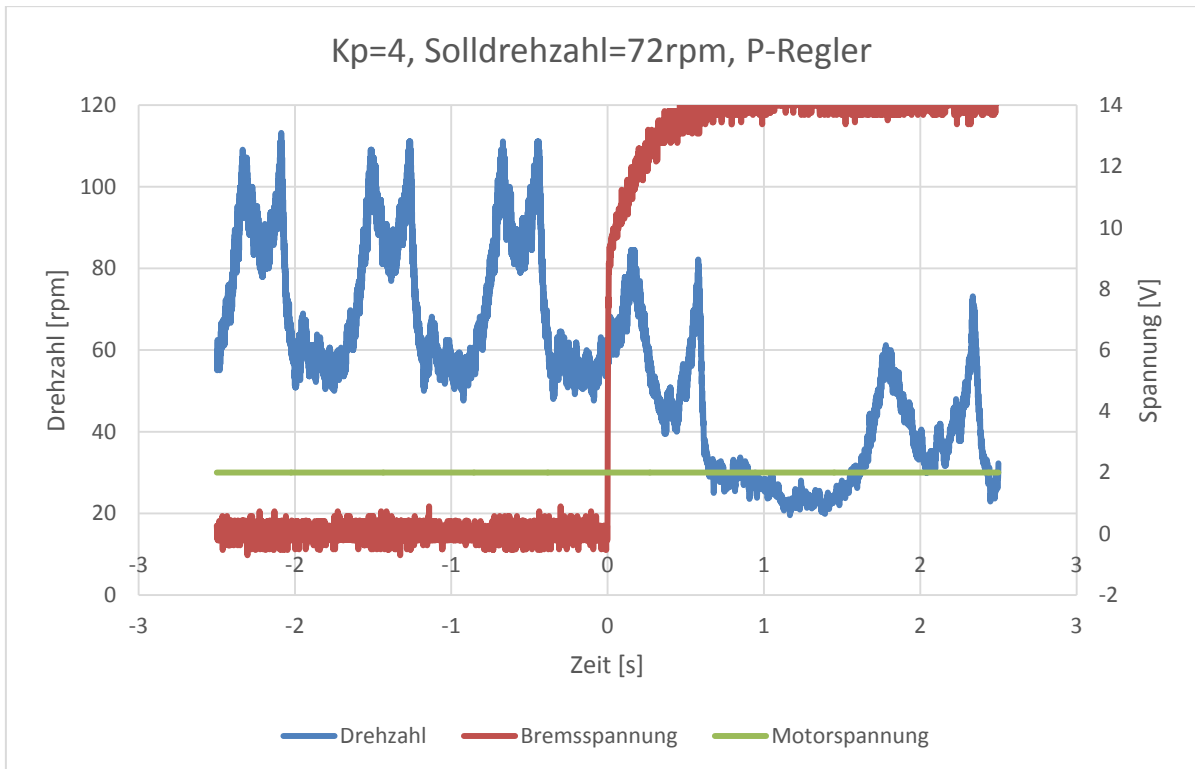


Abbildung E.3: Drehzahlverlauf der Drehzahlregelung mit $K_p=4$ und Solldrehzahl=72rpm

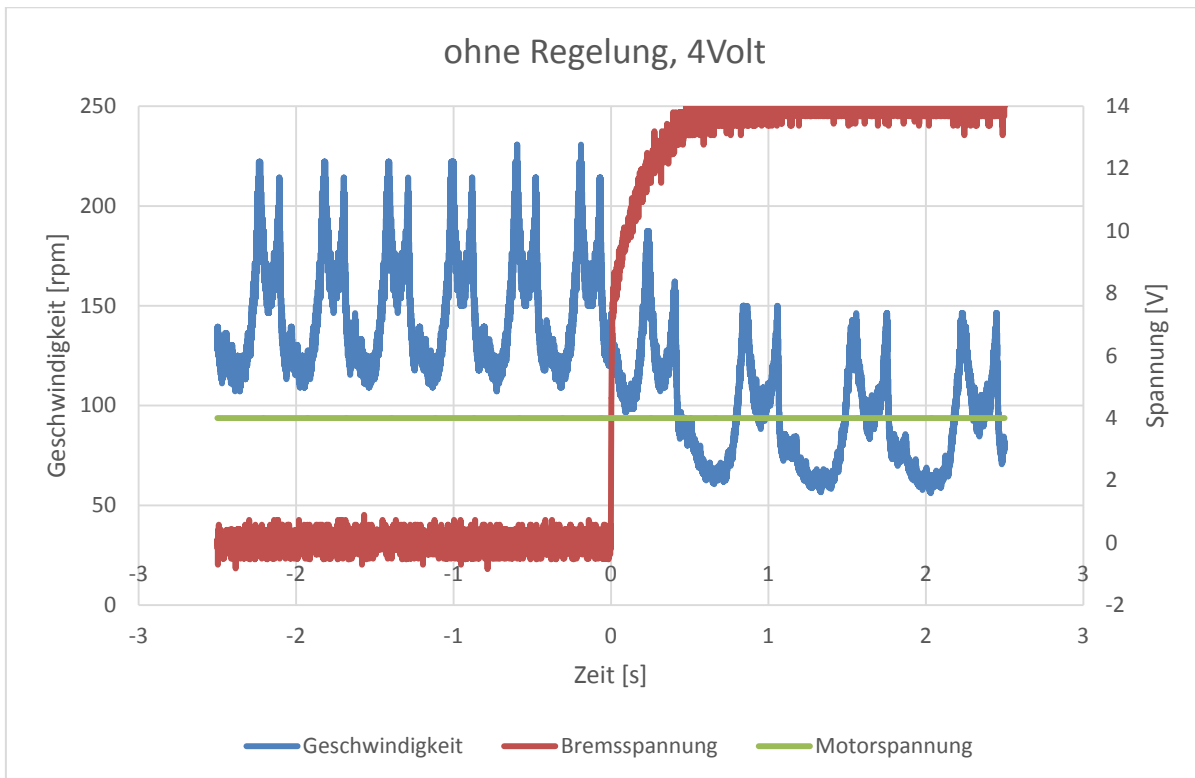


Abbildung E.4: Verlauf der Drehzahl bei konstanter Motorspannung von 4 V

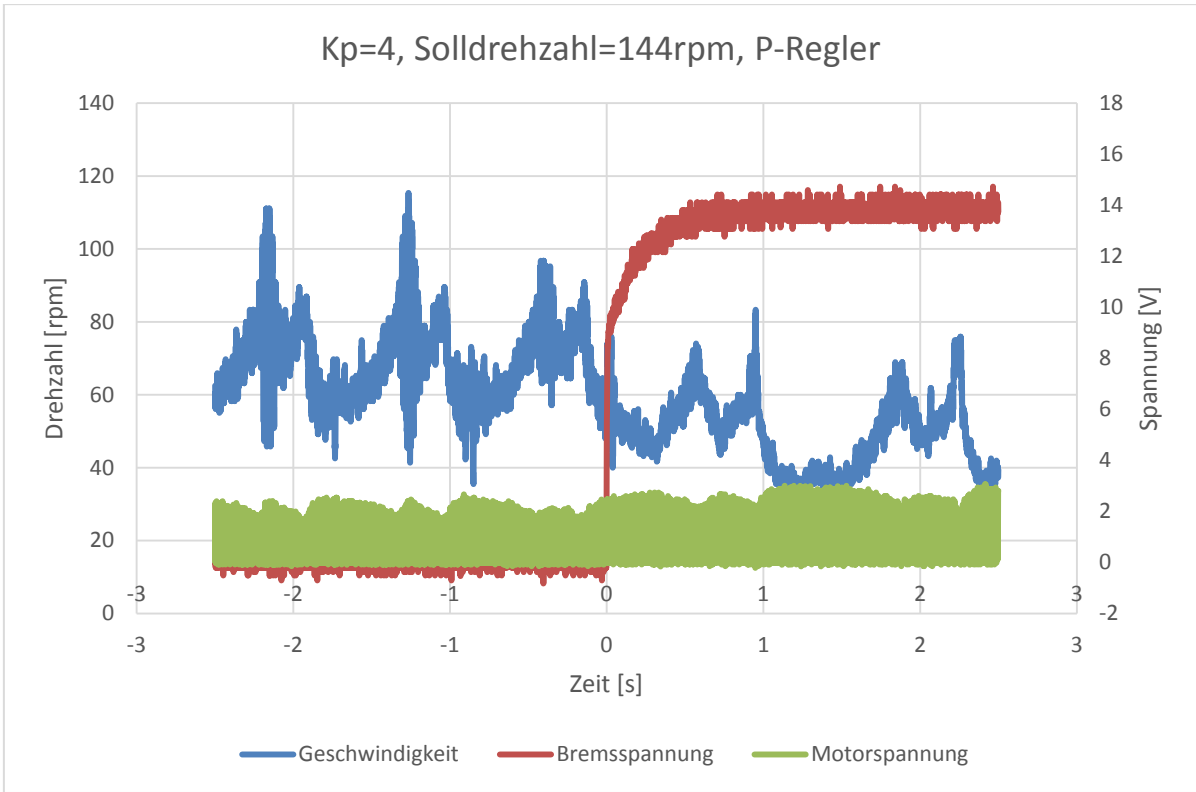


Abbildung E.5: Drehzahlverlauf der Drehzahlregelung mit $K_p=4$ und Solldrehzahl=144rpm

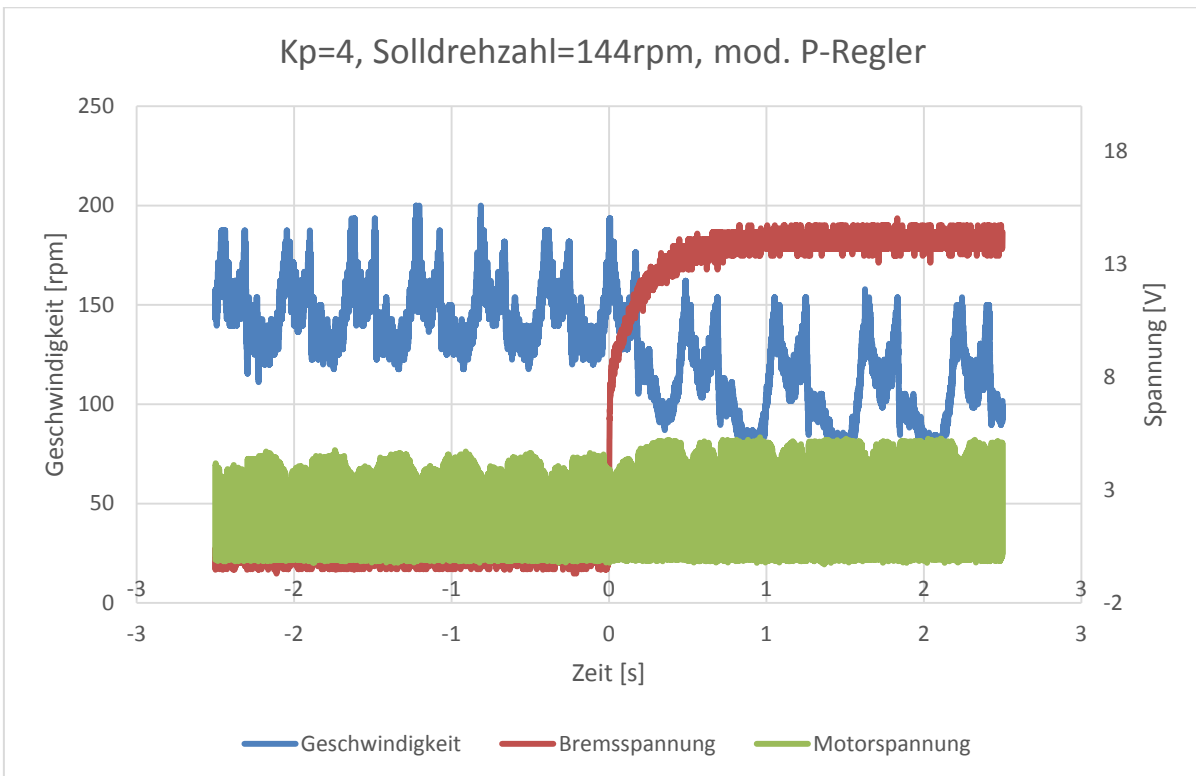


Abbildung E.6: Drehzahlverlauf der Drehzahlregelung mit $K_p=4$ und Solldrehzahl=144rpm

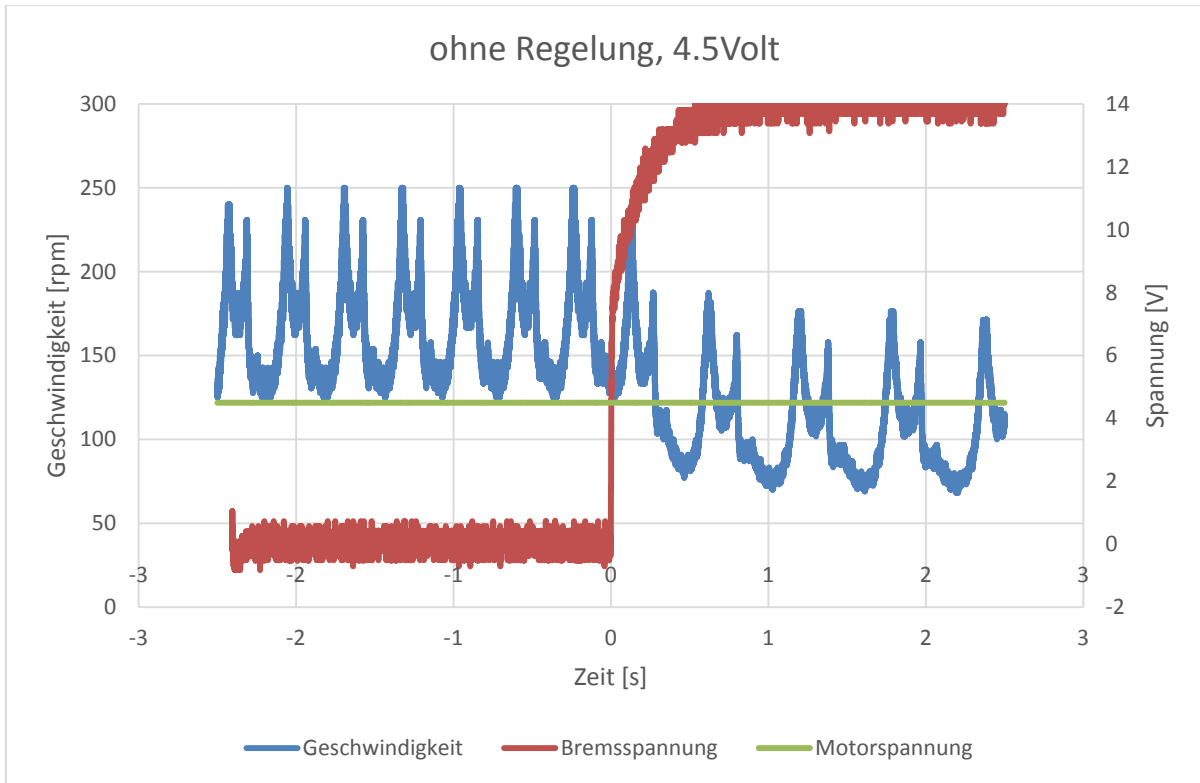


Abbildung E.7: Verlauf der Drehzahl bei konstanter Motorspannung von 4,5 V

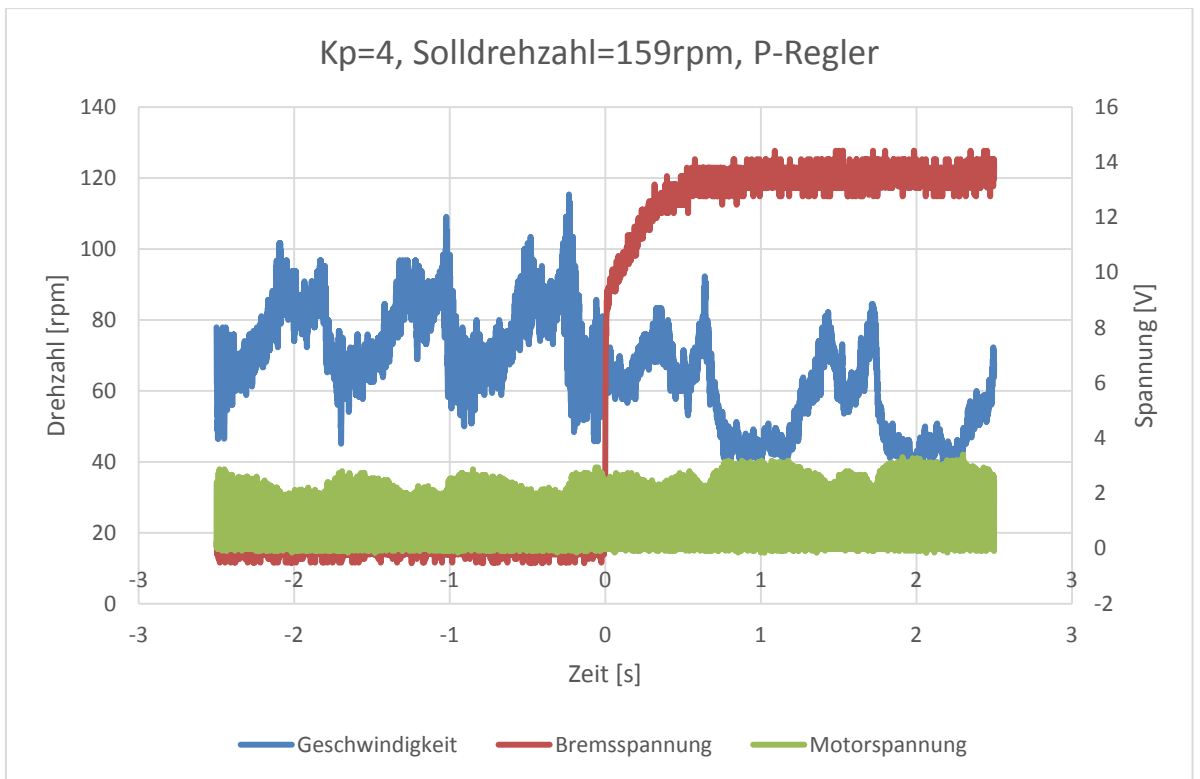


Abbildung E.8: Drehzahlverlauf der Drehzahlregelung mit $K_p=4$ und Solldrehzahl=159rpm

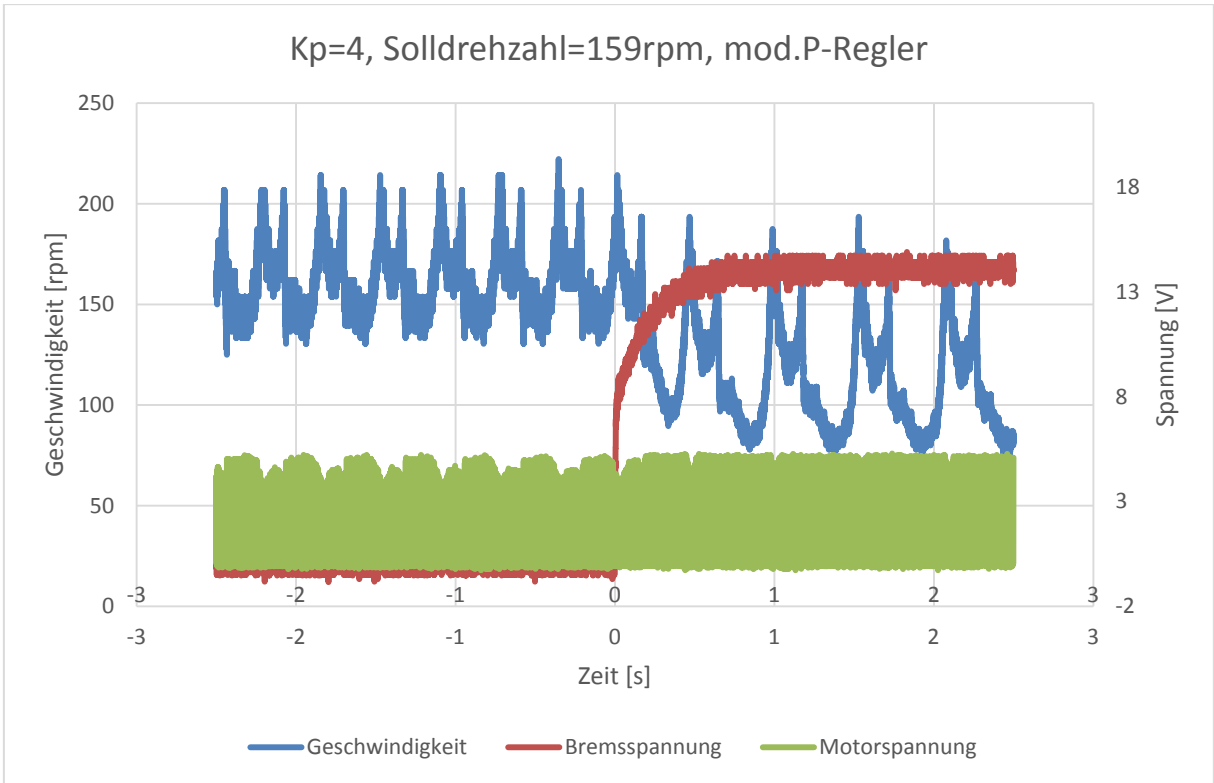


Abbildung E.9: Drehzahlverlauf der Drehzahlregelung mit $K_p=4$ und Solldrehzahl=159rpm

F Abbildungsverzeichnis

Abbildung 1.1: Signalverlauf zur Ansteuerung des Piezomotors auf der SiKo-Platine [1]	2
Abbildung 1.2: Mechanisches Rastsystem. Die Feder drückt den Stift nach unten. Beim Drehen wird der Stift hochgedrückt und das Drehmoment steigt. Nach Überschreiten der Spitze eines Zahnrads springt der Stift in die nächste Rasterung. Wenn der Drehknopf vorher losgelassen wird, dreht sich der Drehknopf zurück und der Stift springt in die letzte Rasterung.	3
Abbildung 2.1: Zwei mögliche Charakteristiken mit verschiedenen feinen Rasterungen und verschiedenen maximalen Drehmomenten.....	8
Abbildung 4.1: Anordnung der Lagen der vierlagigen HACON-Platine. In der Mitte sind die beiden Versorgungslagen 3,3 V und Ground. Außen befinden sich die Signalleitungen. [8].....	11
Abbildung 4.2: Verschaltung für die Konfiguration im „Master Serial/SPI configuration mode“ [9].....	12
Abbildung 4.3: Verschaltung der Schaltregler für die Spannungsversorgung	13
Abbildung 4.4: Anordnung der Komponenten für die Erzeugung der Versorgungsspannung	14
Abbildung 4.5: Anordnung der Komponenten für die AD-Wandlung.....	17
Abbildung 4.6: Verschaltung des USB zu UART Wandlers	18
Abbildung 4.7: Verschaltung des Voltage Monitors.....	19
Abbildung 4.8: Verschaltung des Oszillators	20
Abbildung 4.9: Anordnung des Oszillators. Die Leiterbahn zwischen Oszillator und FPGA ist möglichst kurz zu wählen.....	21
Abbildung 4.10: Verschaltung und Abbildung eines DIP-Schalters [26]	23
Abbildung 4.11: Verschaltung und Abbildung eines Drehschalters [27]	23
Abbildung 5.1: Modulübersicht mit Schnittstellen zum AD-Wandler, Encoder und DA-Wandler	24
Abbildung 5.2: Ablauf des Schreibvorgangs [20].....	25
Abbildung 5.3: Ablauf des Lesevorgangs [20]	26
Abbildung 5.4: Zeitlicher Ablauf der Ansteuerung des DA-Wandlers [22]	27
Abbildung 5.5: Spur A und B eines Encoders	29
Abbildung 5.6: Zustandsautomat zur Bestimmung des gedrehten Winkels bzw. Drehzahl des Piezomotors	30
Abbildung 5.7: Ablauf der Drehmomentregelung.....	31
Abbildung 5.8: Ablauf der Drehmomentsteuerung	33
Abbildung 5.9: Ablauf der Drehzahlregelung	35

Abbildung 5.10: Struktur der Drehzahlregelung	36
Abbildung 6.1: Gesamtübersicht über die verwendeten Platinen sowie der Peripherie zur Ansteuerung des Piezomotors	41
Abbildung 6.2: Bestückte HACON-Platine. Die Nummerierung entspricht den Unterkapiteln in Abschnitt 4. Voltage-Monitor und Oszillator sind nicht nummeriert.....	43
Abbildung 6.3: Messbrücke zur intrinsischen Drehmomentmessung	44
Abbildung 6.4: Verschaltung des invertierenden Operationsverstärkers, um die Spannung des Drehmomentsensors in den Messbereich des AD-Wandlers zu konvertieren	46
Abbildung 6.5: Einfluss der Motorspannung auf das Drehmoment. Kanal C_3 stellt den Verlauf der Motorspannung dar, die am DA-Wandler anliegt. Kanal C_4 stellt den Verlauf der zum Drehmoment proportionalen Spannung des Drehmomentsensors dar. Bei 1 beginnt die Krafteinwirkung auf den Drehknopf. Punkt 2 zeigt den Beginn der Drehung, bei 3 ist die maximale Motorspannung erreicht. Die Motorspannung wird dann abgeschaltet.	47
Abbildung 6.6: Abhängigkeit der Drehzahl von der angelegten Spannung	49
Abbildung 6.7: Verlauf der Drehzahl bei konstanter Motorspannung von 4 V	49
Abbildung 6.8: Drehzahlverlauf der Drehzahlregelung mit $K_p=4$ und Solldrehzahl=108rpm.....	51
Abbildung 6.9: Drehzahlverlauf der Drehzahlregelung mit $K_p=5$ und Solldrehzahl=108rpm.....	51
Abbildung 6.10: Drehzahlverlauf der Drehzahlregelung mit $K_p=4$ und Solldrehzahl=144rpm (ermittelt bei 4 V).....	52
Abbildung 6.11: Drehzahl des Piezomotors bei konstanter Spannung. Zum Zeitpunkt 0 s wird eine Spannung an die Hysteresebremse angelegt.....	54
Abbildung 6.12: Drehzahlverlauf der Drehzahlregelung mit $K_p=4$ und Solldrehzahl=108rpm (ermittelt bei 3 V). Zum Zeitpunkt 0 s wird eine Spannung an die Hysteresebremse angelegt.	54
Abbildung 6.13: Drehzahlschwankung, Differenz zwischen maximaler und minimaler Drehzahl.....	56
Abbildung 6.14: Vergleich der Regelabweichungen mit und ohne Regelung nach Anlegen der Bremsspannung	57
Abbildung B.1: Bank 0 des FPGA	63
Abbildung B.2: Bank 1 des FPGA, Konfigurationsspeicher und UART zu USB-Wandler.....	64
Abbildung B.3: Bank 2 des FPGA und Oszillator	65
Abbildung B.4: Bank 3 des FPGA, SRAM- und Flashspeicher und Voltage Monitor	66
Abbildung B.5: AD-Wandler.....	67

Abbildung B.6: LEDS, Drehschalter, DIP-Schalter und Taster.....	68
Abbildung B.7: Spannungsversorgung und Entkoppelungskondensatoren.....	69
Abbildung B.8: Layout der HACON-Platine.....	70
Abbildung E.1: Verlauf der Drehzahl bei konstanter Motorspannung von 2 V.....	79
Abbildung E.2: Drehzahlverlauf der Drehzahlregelung mit $K_p=4$ und Solldrehzahl=72rpm.....	79
Abbildung E.3: Drehzahlverlauf der Drehzahlregelung mit $K_p=4$ und Solldrehzahl=72rpm.....	80
Abbildung E.4: Verlauf der Drehzahl bei konstanter Motorspannung von 4 V.....	80
Abbildung E.5: Drehzahlverlauf der Drehzahlregelung mit $K_p=4$ und Solldrehzahl=144rpm.....	81
Abbildung E.6: Drehzahlverlauf der Drehzahlregelung mit $K_p=4$ und Solldrehzahl=144rpm.....	81
Abbildung E.7: Verlauf der Drehzahl bei konstanter Motorspannung von 4,5 V	82
Abbildung E.8: Drehzahlverlauf der Drehzahlregelung mit $K_p=4$ und Solldrehzahl=159rpm.....	82
Abbildung E.9: Drehzahlverlauf der Drehzahlregelung mit $K_p=4$ und Solldrehzahl=159rpm.....	83

G Tabellenverzeichnis

Tabelle 4.1: Übersicht über die benötigten und verwendeten Stütz- bzw. Entkoppelungskondensatoren	15
Tabelle 4.2: Überblick und Funktion der Stechverbinder auf der HACON-Platine	22
Tabelle 6.1: Bezeichnung und Funktion der verwendeten Geräte zur Ansteuerung des Piezomotors.....	42
Tabelle 6.2: Ergebnisse der Messungen mit verschiedenen Spannungen. Die minimale und maximale Drehzahl vor (v. B.) und nach (n. B.) dem Anlegen der Bremsspannung wurde ermittelt und die Differenz berechnet.....	55
Tabelle 6.3: Mittelwerte der Drehzahl nach (n. B.) dem Anlegen der Bremsspannung. Die Regelabweichung ist die Differenz zwischen vorgegebener Drehzahl und dem Mittelwert.....	55
Tabelle C.1: Positionen der Steckverbinder mit Verbindung zur CONBOX-Platine	71
Tabelle C.2: Pinbelegung des Steckverbinders P1.....	71
Tabelle C.3: Pinbelegung des Steckverbinders P3.....	71
Tabelle C.4: Pinbelegung des Steckverbinders P4.....	71
Tabelle C.5: Pinbelegung des Steckverbinders P2.....	72
Tabelle C.6: Pinbelegung des Steckverbinders P8.....	73
Tabelle C.7: Pinbelegung des Steckverbinders P5.....	74
Tabelle C.8: Pinbelegung des Steckverbinders P6.....	74
Tabelle C.9: Pinbelegung des Steckverbinders P7.....	74
Tabelle C.10: Pinbelegung des Steckverbinders P10.....	74
Tabelle C.11: Pinbelegung des Steckverbinders P9.....	75
Tabelle D.1: Pinbelegung FPGA, Bank 1	76
Tabelle D.2: Pinbelegung FPGA, Bank 2	77
Tabelle D.3: Pinbelegung FPGA, Bank 3	78