

# Masterthesis

Patrick-Alexander Mattich

Dynamische Spannungsversorgung für  
Schrittmotoren für die Integration im  
Treiber-IC

Patrick-Alexander Mattich  
Dynamische Spannungsversorgung für  
Schrittmotoren für die Integration im  
Treiber-IC

Masterthesis eingereicht im Rahmen der Masterprüfung  
im gemeinsamen Studiengang Master Mikroelektronische Systeme  
am Department Informations- und Elektrotechnik  
der Fakultät Technik und Informatik  
der Hochschule für Angewandte Wissenschaften Hamburg  
und  
am Fachbereich Technik  
der Fachhochschule Westküste

Betreuender Prüfer: Prof. Dr.-Ing. Karl-Ragmar Riemschneider  
Zweitgutachter: Prof. Dr.-Ing. Reiner Johannes Schütt

Abgegeben am 10. Januar 2019

**Patrick-Alexander Mattich**

**Thema der Masterthesis**

Dynamische Spannungsversorgung für Schrittmotoren für die Integration im Treiber-IC

**Stichworte**

Schrittmotortreiber für Kleinleistung, Schrittmotor-Drehmoment, Dynamische Spannungsversorgung eines Schrittmotors, Verbesserte Drehmomentkennlinie, Regelungsstrategien für Schrittmotoren

**Kurzzusammenfassung**

Die vorliegende Masterarbeit beschäftigt sich mit der Entwicklung einer dynamischen Erhöhung der Versorgungsspannung eines Schrittmotors, basierend auf einem Aufwärtswandler, zur Aufrechterhaltung des Drehmoments über einen größeren Drehzahlbereich. Aufgrund der gegen-elektromotorischen Kraft sinkt das Drehmoment von Schrittmotoren bei Verwendung einer niedrigen Versorgungsspannung mit steigender Drehzahl früher als bei Verwendung einer hohen Versorgungsspannung. Diese Arbeit sollte beantworten, inwiefern die Implementierung in einen Schrittmotortreiber-IC eine lohnende Entwicklung wäre. Dazu wurde ein Testaufbau basierend auf einem FPGA entwickelt und auf einem Motorteststand mit herkömmlichen Schrittmotortreibern verglichen. Der neue Ansatz verbessert das Drehmomentverhalten im Vergleich zu einer herkömmlichen Schrittmotorsteuerung signifikant. Dies wird durch eine dynamisch angepasste Versorgungsspannung erreicht.

**Patrick-Alexander Mattich**

**Title of the paper**

Dynamic supply control for stepper motors for integration in the driver IC

**Keywords**

Control strategies for motor drives, low power drives, torque increase of stepper motors, dynamic voltage supply for stepper motors, improved torque characteristics

**Abstract**

The maximum torque stepper motors can produce decreases with the increasing speed of the motor because of the back electromotive force. This means that a low supply voltage leads to a decrease of the torque at a lower speed than a higher supply voltage would achieve. The goal of this thesis is to develop a motion system including a dynamic boost converter for increasing the supply voltage of a stepper motor to maintain high torque over a wider speed range. The results should indicate whether the integration of a dynamic voltage supply in a stepper motor driver IC would be a beneficial development. To this end a test setup based on a FPGA was developed and compared to a regular stepper motor driver using a motor torque tester. The results show that using the dynamic voltage supply results in significantly better torque at higher speeds than the regular stepper driver, while using the same low external voltage supply.

# Danksagung

Ich möchte sehr herzlich Herrn Prof. Dr.-Ing. Karl-Ragmar Riemschneider für die Betreuung dieser Arbeit danken sowie für all die Hilfe, die er mir hat zukommen lassen. Weiterhin möchte ich mich sehr herzlich bei Herrn Prof. Dr.-Ing. Reiner Johannes Schütt dafür bedanken, dass er der Zweitgutachter dieser Arbeit ist. Ein großer Dank gilt auch Herr Müller, Herr Schütthe, Herr Roscher, Herr Rittweger und Herr Modrzynski, die mir immer mit guten Tipps ausgeholfen haben, wenn ich mal nicht weiterwusste.

In der Firma TRINAMIC gilt mein Dank meinem Betreuer Herrn Dwersteg für die viele Geduld und Hilfe sowie Herrn Dressler für den entscheidenden Tipp bezüglich des Regelalgorithmus. Auch meinem Mitstudenten Herrn Ebel möchte ich meinen Dank für immer einen netten Plausch und Hilfe aussprechen.

Einen sehr großen Dank geht auch an meine Freunde, die mich in dem für mich nicht einfachen Jahr 2018 häufig aufgefangen und aufgebaut haben.

Ebenso gilt ein sehr großer Dank meinen beiden Schwestern, die mir mental immer zur Seite gestanden haben.

Abschließend gilt mein wohl größter Dank meinen beiden Eltern, denn ohne ihre Hilfe hätte ich diese Zeit und Arbeit wahrscheinlich niemals aufbringen können.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>4</b>
1.1	Aufbau dieser Arbeit . . . . .	4
1.2	Aktueller Stand . . . . .	5
1.3	Anforderungsanalyse . . . . .	6
1.3.1	Funktionale Anforderungen . . . . .	6
1.3.2	Nichtfunktionale Anforderungen . . . . .	6
1.3.3	Anforderungen an die technische Realisierung . . . . .	7
<b>2</b>	<b>Grundlagen</b>	<b>8</b>
2.1	Schrittmotoren . . . . .	8
2.1.1	Aufbau von Schrittmotoren . . . . .	8
2.1.2	Eigenschaften . . . . .	11
2.1.3	Ansteuerung . . . . .	13
2.1.4	Betriebsarten . . . . .	16
2.1.5	Gegen-elektromotorische Kraft (BEMF) . . . . .	18
2.1.6	Motorteststand . . . . .	19
2.2	Gleichspannungswandler . . . . .	20
2.2.1	Aufbau eines Aufwärtswandlers . . . . .	20
2.2.2	Schaltzustände . . . . .	21
2.2.3	Continuous- und Discontinuous-Conduction Mode . . . . .	23
2.2.4	Arten der Regelung . . . . .	26
2.2.5	Komponentenauswahl . . . . .	28
<b>3</b>	<b>Entwicklung der Hardware</b>	<b>30</b>
3.1	Vorgaben und Konzept . . . . .	30
3.2	Simulation des Aufwärtswandlers . . . . .	31
3.3	Entwicklung von Funktionsmustern . . . . .	35
3.4	Entwicklung der Platine . . . . .	37
3.4.1	Schrittmotortreiber . . . . .	38
3.4.2	Kontrolleinheit . . . . .	40
3.4.3	Leistungsteil des Aufwärtswandlers . . . . .	42
3.4.4	Räumliche Aufteilung und Layout der Platine . . . . .	43
3.4.5	Adapter für das <i>StepStick</i> -Modul Pinout . . . . .	45

---

3.5	Inbetriebnahme . . . . .	45
<b>4</b>	<b>Entwicklung der Firmware</b>	<b>49</b>
4.1	Regelung des Aufwärtswandlers . . . . .	50
4.1.1	Auswertung des ADC . . . . .	51
4.1.2	Hysteresse-Spannungsregelung . . . . .	53
4.2	Auswertung des Schrittsignals . . . . .	56
4.2.1	Vorbereitung der Rechenparameter . . . . .	56
4.2.2	Generierung des Sollwertes . . . . .	60
4.3	Parallelisierung der Zustandsautomaten . . . . .	64
4.4	Sanftanlauf . . . . .	67
<b>5</b>	<b>Ergebnisse</b>	<b>70</b>
5.1	Charakterisierung des Gleichspannungswandlers . . . . .	70
5.1.1	Wirkungsgrad des Aufwärtswandlers . . . . .	70
5.1.2	Störungen in der Ausgangsspannung . . . . .	74
5.1.3	Lastregulierung . . . . .	78
5.1.4	Spannungssprung an resistiver Last . . . . .	79
5.1.5	Spannungssprung an einem Schrittmotor . . . . .	83
5.2	Verhalten des Moduls auf dem Motorteststand . . . . .	88
5.2.1	Betriebsart: SpreadCycle . . . . .	89
5.2.2	Betriebsart: StealthChop . . . . .	95
5.2.3	Vergleich der Messungen . . . . .	101
<b>6</b>	<b>Schlussfolgerung</b>	<b>102</b>
6.1	Diskussion der Ergebnisse . . . . .	102
6.2	Ausblick . . . . .	104
6.2.1	Mögliche Verbesserungen und Weiterentwicklungen . . . . .	104
6.2.2	Integration in einen Chip . . . . .	107
6.2.3	Entwicklung eines skalierbaren Moduls . . . . .	108
6.3	Zusammenfassung und Fazit . . . . .	108
	<b>Literaturverzeichnis</b>	<b>109</b>
	<b>Tabellenverzeichnis</b>	<b>116</b>
	<b>Abbildungsverzeichnis</b>	<b>117</b>
	<b>Listings</b>	<b>121</b>
	<b>Anhang</b>	<b>121</b>
<b>A</b>	<b>Aufbau der Funktionsmuster</b>	<b>122</b>

---

A.1	Funktionsmuster Schaltpläne . . . . .	124
<b>B</b>	<b>Aufbau der entwickelten Platine</b>	<b>128</b>
B.1	Schaltplan der Platine . . . . .	128
B.2	Layout der Platine . . . . .	132
B.3	Aufbau des Adapters . . . . .	133
<b>C</b>	<b>Simulation der Zustandsautomaten</b>	<b>134</b>
<b>D</b>	<b>Firmware VHDL-Code</b>	<b>136</b>
<b>E</b>	<b>Inhalt der CD</b>	<b>146</b>

# 1 Einleitung

Im niedrigpreisigen Bereich der 3D-Drucker für den Heimbedarf wird meistens ein 12 V Netzteil als Versorgung für die gesamte Elektronik (Mikrocontroller, Display, Heizbett, Schrittmotortreiber, etc.) verwendet. Durch den Preisdruck werden dazu meistens sehr preiswerte und drehmomentschwache Schrittmotoren verwendet. Aus dieser Kombination resultiert eine Begrenzung der maximalen Verfahrgeschwindigkeit, bevor die Motoren anfangen aufgrund des sinkenden Drehmoments Schritte zu verlieren. Dadurch wird der laufende 3D-Druck in den meisten Fällen unbrauchbar. Eine Lösung wäre es, das Netzteil gegen eins mit einer höheren Spannung auszutauschen. Allerdings muss dann die Elektronik des Druckers an die höhere Spannung angepasst werden. Eine weitere Möglichkeit wäre zwei Netzteile zu verwenden, eins für die Elektronik und eins für die Schrittmotoren, was jedoch teuer wäre. Zusätzlich wird die höhere Spannung nicht konstant benötigt, sondern nur, wenn ein Schrittmotor sich so schnell dreht, dass sein Drehmoment bei der aktuellen Versorgungsspannung so stark abfallen würde, dass der Motor Schritte verlieren würde. In der Zwischenzeit, in der die niedrige Versorgungsspannung ausreicht, wäre die Verwendung einer höheren Spannung unnötig. Eine Möglichkeit dieses Problem zu lösen wäre es, die Spannung der einzelnen Schrittmotoren jeweils dynamisch an den aktuellen Bedarf anzupassen. Dies könnte mithilfe von Gleichspannungswandlern erfolgen, welche jeweils einen Schrittmotortreiber versorgen und ihre generierte Ausgangsspannung an den Bedarf des Treibers anpassen. Ziel dieser Arbeit war es daher, eine Lösung zu erforschen und zu entwickeln, die das Verhalten des Schrittmotors auswertet und dessen Versorgungsspannung mithilfe eines Gleichspannungswandlers so anpasst, dass das Drehmoment über einen weiten Drehzahlbereich aufrechterhalten werden kann.

## 1.1 Aufbau dieser Arbeit

Diese Arbeit beginnt mit dem aktuellen Stand. Anschließend werden die, für das Verständnis dieser Arbeit notwendigen, Grundlagen von Schrittmotoren und Aufwärtswandlern erklärt. Danach wird die Entwicklung der Hardware sowie die dafür durchgeführten Simulationen und aufgebauten Funktionsmustern erklärt und

die Inbetriebnahme beschrieben. Auf Basis der entwickelten Hardware wird darauf folgend die Entwicklung der Firmware beschrieben, welche die Auswertung der Schrittmotorsignale sowie die Regelung des Spannungswandlers enthält. Anschließend folgt der Ergebnisteil, worin einerseits das Einzelverhalten des entwickelten Spannungswandlers beschrieben wird und das Gesamtverhalten des entwickelten Moduls auf einem Motorteststand. Die Ergebnisse werden daraufhin diskutiert. Es folgt schließlich ein Ausblick auf mögliche Weiterentwicklungen dieser Arbeit und darauf aufbauender Projekte sowie eine abschließende Zusammenfassung und ein Fazit.

## 1.2 Aktueller Stand

Das Unternehmen *TRINAMIC Motion Control GmbH & CO. KG* (im Folgenden als TRINAMIC bezeichnet) ist eine sogenannte *Fabless Company*. Fabless, aus dem Englischen für *fabricationless*, bedeutet im Deutschen soviel wie *fabriklos*. Damit werden hauptsächlich Firmen aus der Halbleiterindustrie bezeichnet, welche ICs und andere Halbleiter entwickeln, jedoch keine eigenen Fertigungsstätten besitzen. TRINAMIC ist spezialisiert auf die Entwicklung von Schrittmotor- und BLDC<sup>1</sup>-Motortreiber-ICs und auf den Treiber-ICs basierende Module und komplette Motion-Control<sup>2</sup>-Systeme. Durch den in den letzten Jahren stark wachsenden Markt für 3D-Drucker im Heimbereich, hat sich dort ein besonders preiseempfindlicher Markt gebildet. TRINAMIC hat bereits einen großen Marktanteil bei mittelpreisigen Prosumer<sup>3</sup> Geräten. Die aktuellen Schrittmotortreiber von TRINAMIC bieten Vorteile, wie einen besonders leisen Motorlauf und das Erkennen eines Endanschlags ohne einen extra Endschalter durch Analyse der Motorsignale. An dem Motordrehmoment in Abhängigkeit der Drehzahl, welches hauptsächlich von dem Motorstrom, der Motorspannung sowie teilweise von dem Betriebsmodus abhängt, konnte bisher durch den Motortreiber nichts verändert werden, da die Schrittmotortreiber eine konstante Versorgungsspannung nutzen.

---

<sup>1</sup>Brushless DC Motor, englisch für bürstenloser Gleichstrommotor

<sup>2</sup>Englisch für Bewegungsregelung

<sup>3</sup>Kofferwort aus den englischen Wörtern *professional* und *consumer*. Es beschreibt einen Endverbraucher, der erhöhte bzw. professionelle Anforderungen an ein Produkt stellt

## 1.3 Anforderungsanalyse

Die durch TRINAMIC gestellten Anforderungen an die Entwicklung lassen sich in funktionale Anforderungen, nichtfunktionale Anforderungen sowie Anforderungen an die technische Realisierung aufteilen, welche im Folgenden beschrieben werden.

### 1.3.1 Funktionale Anforderungen

Folgende funktionale Anforderungen wurden von TRINAMIC vorgegeben:

- Das Drehmoment eines Schrittmotors soll durch Verwendung des zu entwickelnden Moduls über einen größeren Drehzahlbereich aufrechterhalten werden, als es die bloße Verwendung der Eingangsspannung des Moduls ermöglichen würde
- Es soll einem bestehenden 3D-Drucker durch einfaches Austauschen des vorhandenen Motortreibers mit dem entwickelten Modul ermöglichen, dass durch das hinzugewonnene Drehmoment schneller gedruckt werden kann
- Es soll nur ein Aufwärtswandeln der Eingangsspannung stattfinden, kein Abwärtswandeln
- Das zu entwickelnde Modul soll einfach an verschiedene Motoren anpassbar sein

### 1.3.2 Nichtfunktionale Anforderungen

Folgende nichtfunktionale Anforderungen wurden von TRINAMIC vorgegeben:

- Das zu entwickelnde Modul soll pin- und funktionskompatibel zu dem bestehenden *SilentStepStick* Schrittmotortreiber sein und in gängigen 3D-Drucker-Steuerungen verwendet werden können, ohne dessen Hard- oder Software anpassen zu müssen
- Das zu entwickelnde Modul soll auf dem *TMC2130* Schrittmotortreiber basieren
- An die Entwicklung anschließend, soll das Verhalten und die Charakteristika des Moduls ermittelt werden, unter anderem mithilfe eines Motorteststands
- Es ist das Erreichen eines möglichst hohen Wirkungsgrades gewünscht

### 1.3.3 Anforderungen an die technische Realisierung

Folgende Anforderungen an die technische Realisierung wurden von TRINAMIC vorgegeben:

- Entwicklung, Aufbau und Inbetriebnahme einer oder mehrerer Platinen
- Die zu generierende Ausgangsspannung soll zwischen  $12\text{ V}$  und  $36\text{ V}$  einstellbar sein
- Das zu entwickelnde Modul soll eine Ausgangsleistung von  $20\text{ W}$  besitzen
- Als Spannungswandler Topographie soll ein Aufwärtswandler mit einem aktiven und einem passiven Schalter verwendet werden

## 2 Grundlagen

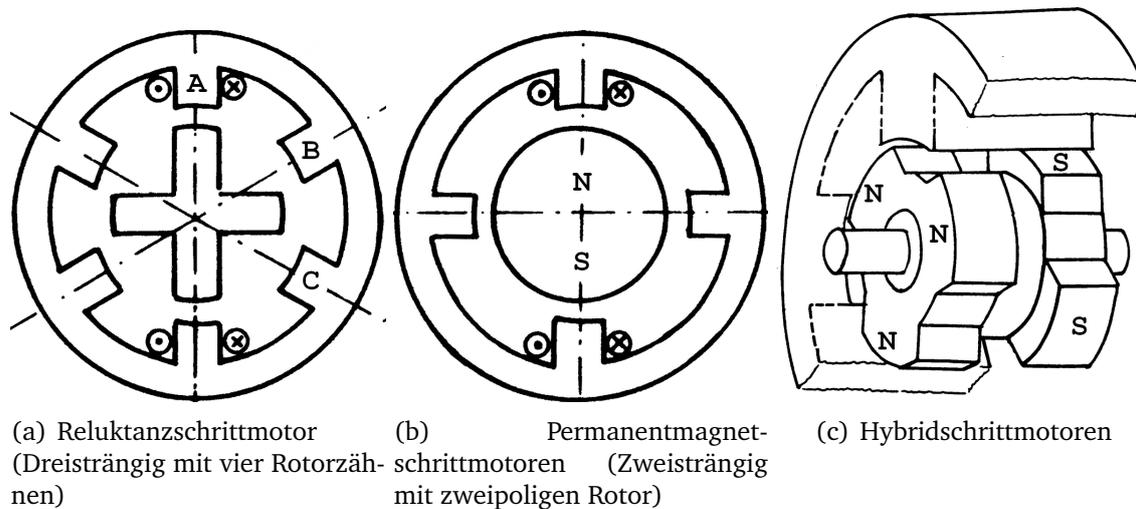
In diesem Kapitel sollen die Grundlagen vermittelt werden, auf denen diese Arbeit aufbaut. Dabei wird auf die Funktionsweise und Eigenschaften von Schrittmotoren und Aufwärtswandlern eingegangen und die Merkmale sowie Eigenschaften, denen diese Arbeit zu Grunde liegt, erläutert.

### 2.1 Schrittmotoren

Schrittmotoren sind eine Sonderform der Reluktanzmotoren, welche wiederum eine Unterkategorie der Synchronmotoren bildet [51]. Da sich diese Arbeit ausschließlich auf Schrittmotoren bezieht und Eigenschaften von Schrittmotoren genutzt werden, wird im Folgenden nicht weiter auf andere Motorentypen eingegangen. Es sei dazu auf entsprechende Fachliteratur verwiesen [51]. Schrittmotoren werden dort eingesetzt, wo eine genaue Positionierung und ein hohes Haltemoment notwendig ist. Anwendungsgebiete sind Drucker, 3D-Drucker, Roboter sowie in Fertigungsanlagen und automatisierte Prozessen in der Industrie .

#### 2.1.1 Aufbau von Schrittmotoren

Es gibt drei gängige Bauarten von Schrittmotoren, die Reluktanzschrittmotoren, die Permanentmagnetschrittmotoren und die Hybridschrittmotoren. Dabei ist der Hybridschrittmotor in den beschriebenen Anwendungen, der am häufigsten eingesetzte Schrittmotortyp [3]. Im Folgenden soll die Funktionsweise der verschiedenen Bauarten kurz erläutert werden.



**Abbildung 2.1** Schematische Darstellung der unterschiedlichen Schrittmotorbauarten. Die Polzahl ist für eine bessere Anschaulichkeit stark reduziert. Bildquelle [66, S. 5-6]

### Reluktanzschrittmotor

Abbildung 2.1a zeigt den schematischen Aufbau eines dreisträngigen Reluktanzschrittmotors. Die drei Strangwicklungen A, B und C sind auf dem Stator aufgebracht. Der Rotor besteht aus einem weichmagnetischen Material. Wenn, wie in Abbildung 2.1a, der Strang A stromdurchflossen ist, richtet sich der Rotor wie gezeigt aus. In dieser Position ist der magnetische Widerstand (Reluktanz) des Rotors für den erzeugten magnetischen Fluss am geringsten. Wenn danach die Windung A ausgeschaltet wird und die Windung C stromdurchflossen ist, dreht sich der Rotor im Uhrzeigersinn, bis die Reluktanz wieder minimal ist. In dem Beispiel dreht sich der Rotor dafür um  $30^\circ$ . Im Reluktanzschrittmotor wirkt im stromlosen Zustand kein Moment auf den Rotor, er besitzt kein Rastmoment. Durch die Anzahl der Stränge, und vor allem der Rotorzähne, wird der Schrittinkel des Schrittmotors bestimmt [66, S. 42-44].

### Permanentmagnetschrittmotor

In Abbildung 2.1b ist der schematische Aufbau eines zweisträngigen Permanentmagnetschrittmotors dargestellt. Der Rotor besteht, im Gegensatz zu den Reluktanzschrittmotoren, aus einem Permanentmagneten. Dadurch besitzt der Permanentmagnetschrittmotor auch im stromlosen Zustand ein Rastmoment. Beim Einschalten von Strang A richtet sich, je nach angelegter Stromrichtung, der Rotor aus. Wird Strang A abgeschaltet und Strang B eingeschaltet, dreht sich der Rotor, je nach angelegter Stromrichtung, um  $90^\circ$  im oder gegen den Uhrzeigersinn. Durch die Anzahl der Stränge (Phasen) und durch die Anzahl der magnetischen Polpaare des Rotors wird der Schrittinkel bestimmt [66, S. 68-69]. Eine häufig verwendete Bauform des Permanentmagnetschrittmotors sind Klauenpolschrittmotoren.

### Hybridschrittmotor

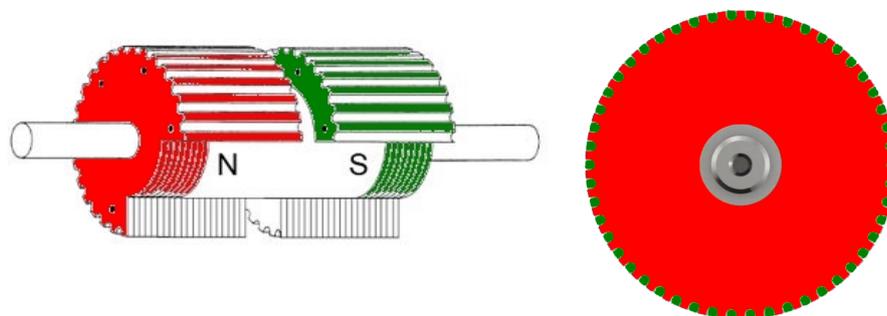
Hybridschrittmotoren sind eine Kombination aus den beschriebenen Reluktanzschrittmotoren und den Permanentmagnetschrittmotoren. Es kommt zum einen ein Permanentmagnet zum Einsatz, was ein hohes Rast- und Haltemoment ermöglicht, zum anderen wird ein gezahnter Rotor sowie Stator, wie bei einem Reluktanzschrittmotor, verwendet, um eine hohe Schrittauflösung zu erreichen. Hybridschrittmotoren sind, vor allem in anspruchsvolleren Anwendungen (Roboter, 3D-Drucker, Fertigungsanlagen), die am häufigsten verwendete Schrittmotorbauart [3]. Der Rotor ist wie in Abbildung 2.2 dargestellt aufgebaut. Es wird ein Permanentmagnet längs der Motorachse verwendet, auf dessen Nord- und Südpol jeweils eine Zahnscheibe aufgebracht ist. Dabei sind die beiden Zahnscheiben so angeordnet, dass sie um einen halben Zahn zueinander versetzt sind (siehe Abbildung 2.2b). Das verwendete Beispiel besitzt 50 Polpaare und kommt damit auf einen Winkel von  $7.2^\circ$  zwischen den Zähnen, der entsprechende Stator besitzt 48 Zähne. Durch diesen Versatz zwischen Rotor und Stator kann durch ein angelegtes, sich änderndes Magnetfeld, eine Rotation des Rotors entstehen, ähnlich wie bei dem Reluktanzschrittmotor. Die Anzahl der Stränge, beziehungsweise Phasen, hat Einfluss auf die Anzahl der Statorsegmente, der in Abbildung 2.3 gezeigte Stator ist für zwei Phasen ausgelegt [66, S. 100-110]. Die Anzahl der Schritte pro Umdrehung  $z$  hängt von der Polzahl  $p$  und dem Faktor  $k$  ab, dabei ist  $k$  für den Vollschrittbetrieb  $k = 2 \cdot m_s$  und für den Halbschrittbetrieb  $k = 4 \cdot m_s$ , wobei  $m_s$  für die Anzahl der Phasen steht [66, S. 109-110].

$$z = k \cdot p \quad (2.1)$$

Der Schrittwinkel eines Hybridschrittmotors kann dann über die folgende Formel bestimmt werden:

$$\varphi = \frac{360^\circ}{z} \quad (2.2)$$

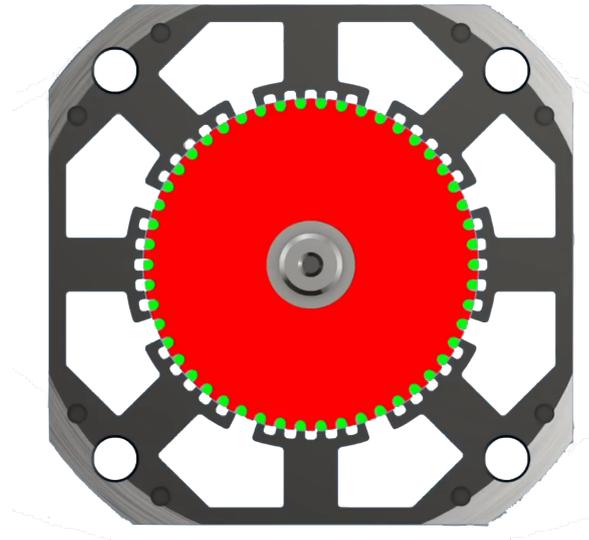
Für das oben genannte Beispiel mit  $p = 50$  und einem zweiphasigen Motor ergeben sich somit 200 Vollschritte pro Umdrehung beziehungsweise einen Vollschrittwinkel von  $\varphi = 1.8^\circ$ .



(a) Schnitt durch den Rotor  
Bild modifiziert nach [53]

(b) Draufsicht  
Bild modifiziert nach [39]

**Abbildung 2.2** Detailansicht eines Hybridschrittmotorrotors. Die Zähne mit Nord- und Südmagnetisierung sind jeweils versetzt, um eine hohe Auflösung zu erreichen.



**Abbildung 2.3** Detailansicht eines Hybridschrittmotorstators  
Bild modifiziert nach [39]

In den folgenden Kapiteln wird sich hauptsächlich auf zweiphasige Hybridschrittmotoren bezogen, da diese in den anspruchsvolleren Anwendungen fast ausschließlich zum Einsatz kommen [3].

### 2.1.2 Eigenschaften

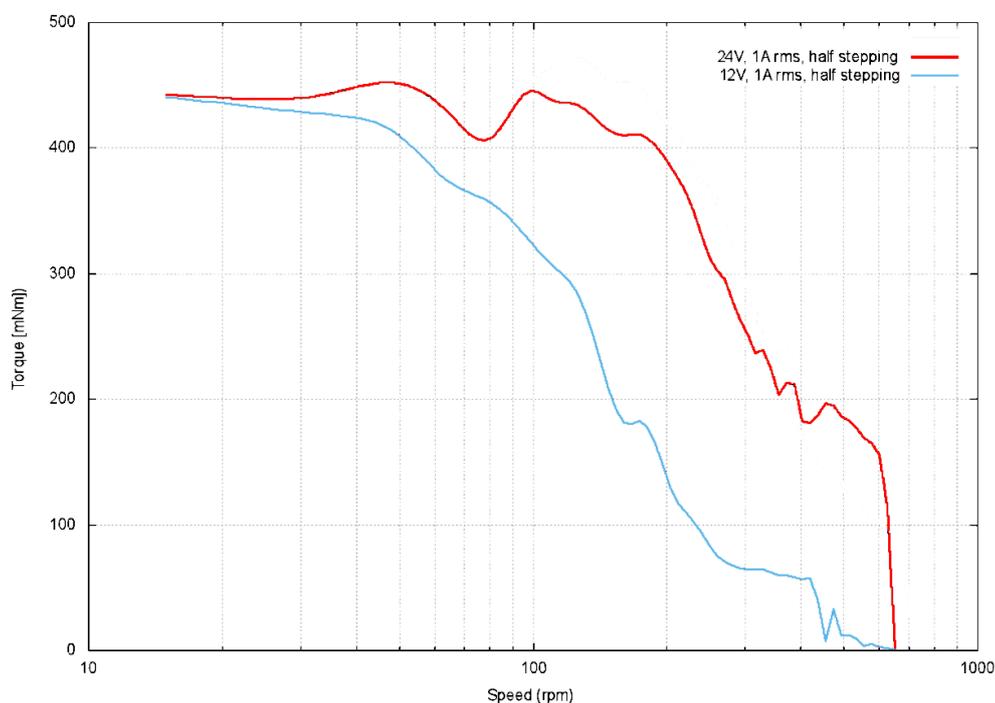
Schrittmotoren haben Eigenschaften, die sie von anderen Motortypen unterscheidet und für viele Anwendungen, insbesondere für solche mit einer hohen Anforderung an die Positionierungsgenauigkeit und das Dreh- und Haltemoment, prädestiniert. Diese Eigenschaften werden im Folgenden kurz beschrieben.

#### Schrittwinkel

Der mechanische, also Vollschrittwinkel eines Hybridschrittmotors hängt von der Anzahl der Rotor-Polpaare und der Anzahl der Phasen ab (siehe Kapitel 2.1.1). Eine in 3D-Druckern häufig verwendete Schrittauflösung sind 200 Vollschritte pro Umdrehung, was einem Winkel von  $\varphi = 1.8^\circ$  entspricht.

### Drehmoment

In Datenblättern von Schrittmotoren gibt es meistens zwei Angaben, zum einen das Haltemoment als eine Zahl für einen festen Motorstrom und zum anderen den Drehmomentverlauf als Graphen, in Abhängigkeit von der Drehzahl. Dieser Drehmomentverlauf hängt von der verwendeten Motorspannung, dem eingestellten Motorstrom und der Betriebsart ab. In Abbildung 2.4 ist beispielhaft ein Drehmomentverlauf des *PD42-2-1070* von TRINAMIC für unterschiedliche Versorgungsspannungen und Betriebsarten dargestellt. Je niedriger die Versorgungsspannung, desto früher (bezogen auf die Drehzahl) bricht das Drehmoment ein. Dies liegt an der gegen-elektromotorischen Kraft (siehe Kapitel 2.1.5).



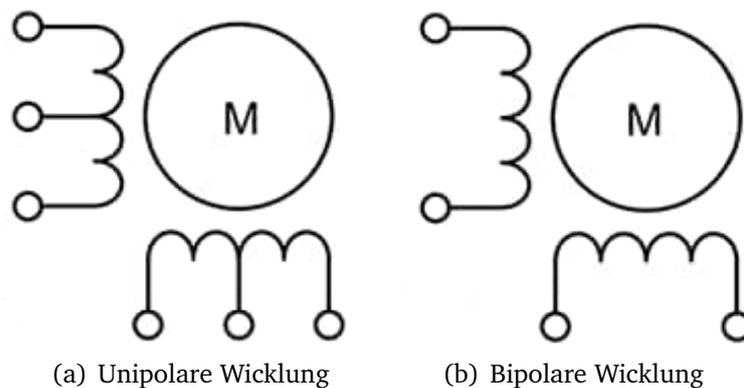
**Abbildung 2.4** Drehmomentkurve aus dem Datenblatt des *PD42-2-1070* von TRINAMIC [42] (farblich modifiziert). Sie zeigt die Auswirkung, die die Versorgungsspannung auf das Drehmoment hat. Die blaue Kennlinie wurde bei 12 V aufgenommen, die rote bei 24 V.

### Bauformen

Hybridschrittmotoren werden häufig in sogenannten NEMA Größen angegeben. NEMA steht dabei für *National Electrical Manufacturers Association* einem Zusammenschluss von Wirtschaftsvertretern der US-amerikanischen elektrotechnischen Industrie. Die Nummer gibt in Zoll die Breite des Motors an. Ein NEMA17 Motor hat somit eine Breite von  $1.7 \cdot 25.4 \text{ mm} = 43.18 \text{ mm}$ . Die Größe NEMA17 ist die in 3D-Druckern am häufigsten verwendete Größe. Große Schrittmotoren, zum Beispiel im NEMA42 Format, haben zum Teil ein Haltemoment von über  $10 \text{ Nm}$ .

### Wicklungsarten

Bei Reluktanzschrittmotoren hat die Polung des erzeugten Magnetfeldes keinen Einfluss auf die Richtungsänderung des Drehmomentes, da sich der Rotor über die minimale Reluktanz ausrichtet. Bei Permanentmagnet- und Hybridschrittmotoren hängt, aufgrund des permanentmagnetischen Rotors, die Richtung des Drehmoments von der Polung des angelegten Magnetfeldes ab. Eine Möglichkeit ist dafür die Verwendung einer unipolaren Wicklung (siehe Abbildung 2.5a). Dabei werden die beiden Mittelanzapfungen jeweils auf Masse gelegt und die vier Spulen jeweils ein- oder ausgeschaltet. Es sind dazu nur vier Schalter nötig, dafür ist der Wirkungsgrad jedoch geringer, da immer nur eine Hälfte der Spule bestromt wird [66, S. 79-81]. Bei der Verwendung einer bipolaren Wicklung (siehe Abbildung 2.5b) muss die Stromrichtung in den Spulen jeweils komplett umgekehrt werden, wofür eine aufwendigere Schaltung mit acht Schaltern (siehe Abbildung 2.6) notwendig ist. Dafür ist der Wirkungsgrad aber auch höher, da die gesamte Spule stromdurchflossen ist [66, S. 79-81].

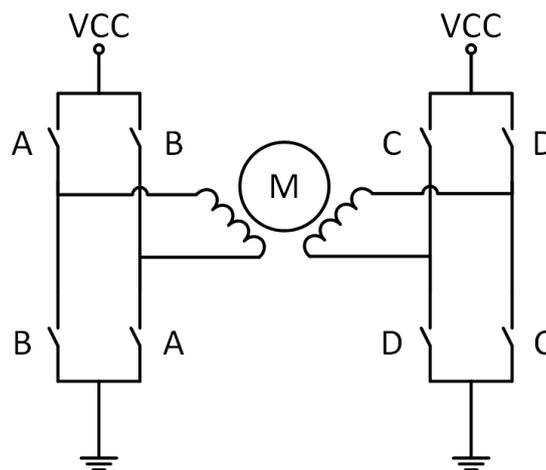


**Abbildung 2.5** Unterschiedliche Wicklungsarten, welche bei Schrittmotoren Anwendung finden  
 Bildquelle [68]

### 2.1.3 Ansteuerung

Wie in Kapitel 2.1.2 unter Wicklungsarten beschrieben, muss die Stromrichtung in den Motorphasen umkehrbar sein, um die Polung des erzeugten Magnetfeldes wechseln zu können, was bei Permanentmagnet- und Hybridschrittmotoren für eine Richtungsänderung des Drehmoments sorgt. Dafür nutzt man zumeist H-Brückenschaltungen, auch Vollbrückenschaltung genannt. Da für jede Phase des Schrittmotors die Stromrichtung geändert werden muss, ist für jede Phase eine einzelne H-Brücke notwendig. In Abbildung 2.6 ist der Aufbau eines Treibers für zwei Phasen dargestellt. Je nach Treiber können die Transistoren dabei alle in einem IC

implementiert sein oder extern an einen IC angeschlossen werden. Ein sehr verbreitetes Steuerprotokoll ist das Step/Direction<sup>1</sup> Protokoll. Dafür werden zwei Signale benötigt, einmal das Richtungssignal, dessen Polarität die Drehrichtung vorgibt, und das Schrittsignal. Das Schrittsignal ist ein Rechtecksignal, für dessen aufsteigende Flanke der Schrittmotor jeweils um einen Schritt weitergedreht wird. Der signalgenerierende Controller muss dabei jeden einzelnen Schritt ausgeben und für Beschleunigungsrampen und Zieldrehzahlen die Berechnungen durchführen.

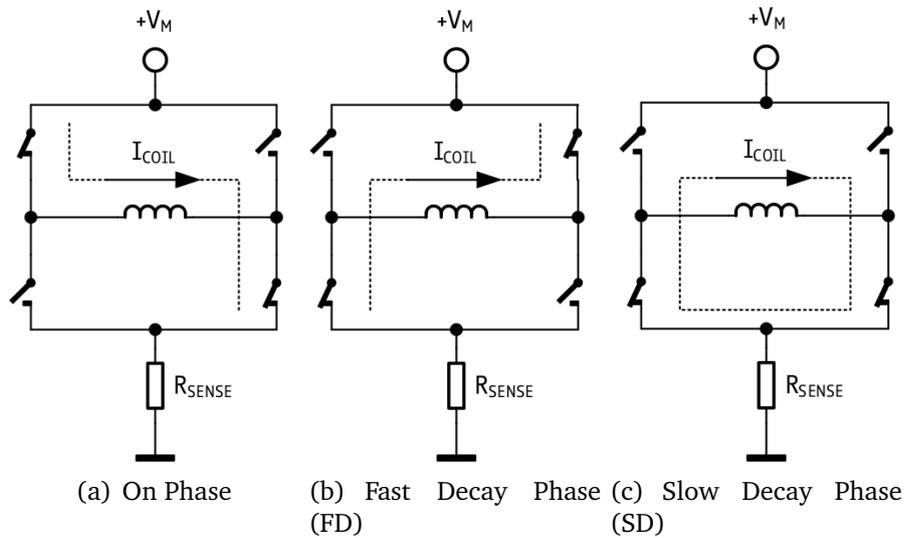


**Abbildung 2.6** Aufbau eines Schrittmotortreibers mit zwei H-Brückenschaltungen für einen zweiphasigen Schrittmotor mit bipolarer Wicklung

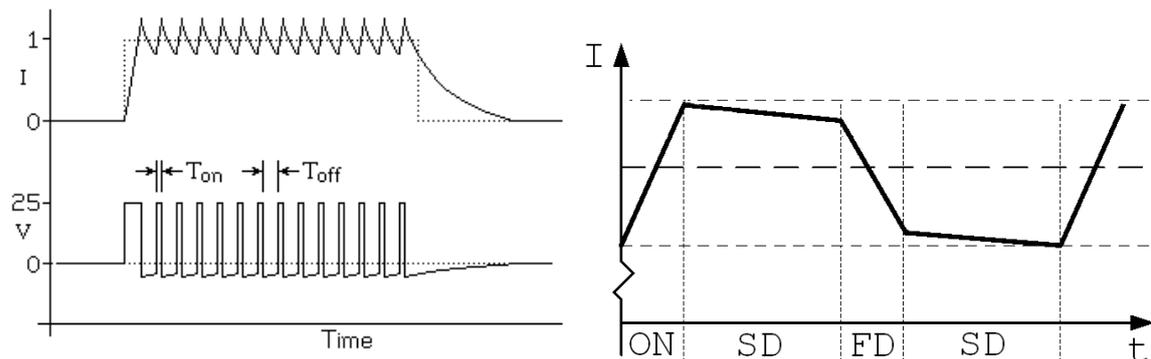
Viele Schrittmotortreiber steuern den Motorstrom, der durch die Motorspulen fließt, mithilfe eines sogenannten Choppers<sup>2</sup>. Dafür wird über einen Messwiderstand im Strompfad der Motorspule (in Abbildung 2.7  $R_{SENSE}$ ) der aktuelle Strom gemessen. Ist der Strom kleiner als der eingestellte Sollstrom, wird weitere Spannung an die Spule angelegt (siehe Abbildung 2.7a). Wurde ein größerer Motorstrom als der Sollstrom gemessen, wird mithilfe der H-Brücke entweder die anliegende Spannung so gepolt, dass sie dem Spulenstrom entgegen wirkt (siehe Abbildung 2.7b), was ein schnelles Abfallen des Stroms zur Folge hat, oder es wird keine Spannung mehr angelegt und der Motorstrom kann durch die beiden unteren Schalter im Kreis laufen und über den Kupferwiderstand und den Drain-Source Widerstand langsam absinken (siehe Abbildung 2.7c). Je nach Treiber kann sich durch das Zusammensetzen dieser Zustände ein in Abbildung 2.8 dargestellter Stromverlauf ergeben.

<sup>1</sup>Englisch für Schritt/Richtung

<sup>2</sup>Englisch für Zerhacker



**Abbildung 2.7** Schaltzustände einer H-Brücke während der Chopper-Stromregelung  
Bildquelle [48, S. 32]



(a) Stromverlauf bei Verwendung einer Chopper Regelung. Bildquelle [34]      (b) Detailaufnahme des Stromverlaufs der Chopper Regelung. Es sind die in Abbildung 2.7 gezeigten Schaltphasen zu erkennen. Bild modifiziert nach [48]

**Abbildung 2.8** Stromregelung einer Schrittmotorphase

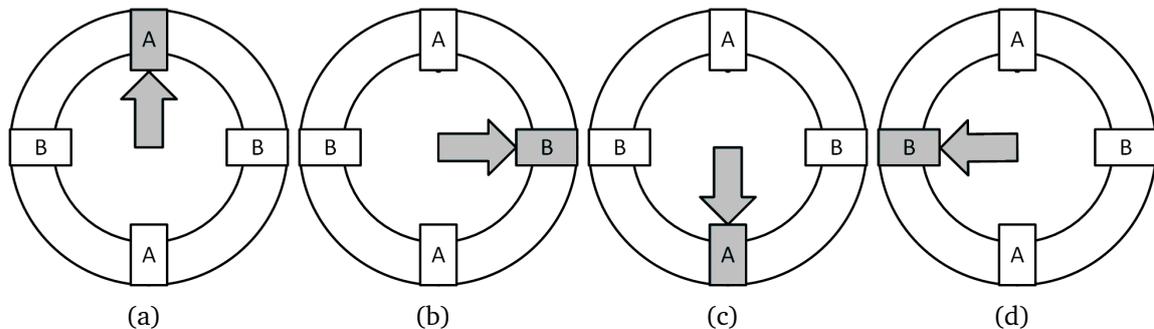
Eine Besonderheit ist dabei die Betriebsart *StealthChop* von TRINAMIC, da hier zusätzlich noch die am Motor anliegende Spannung moduliert wird, um möglichst geringe Stromschwankungen in der Spule zu erreichen, was einen leiseren Motorlauf zur Folge hat [46].

### 2.1.4 Betriebsarten

Da bei Schrittmotoren ein sich änderndes Magnetfeld erzeugt werden muss, um den Rotor in die gewünschte Bewegung zu versetzen, ist es notwendig, die Motorspulen in bestimmten Reihenfolgen zu schalten. Im Folgenden werden die verschiedenen Betriebsarten erläutert.

#### Vollschritt

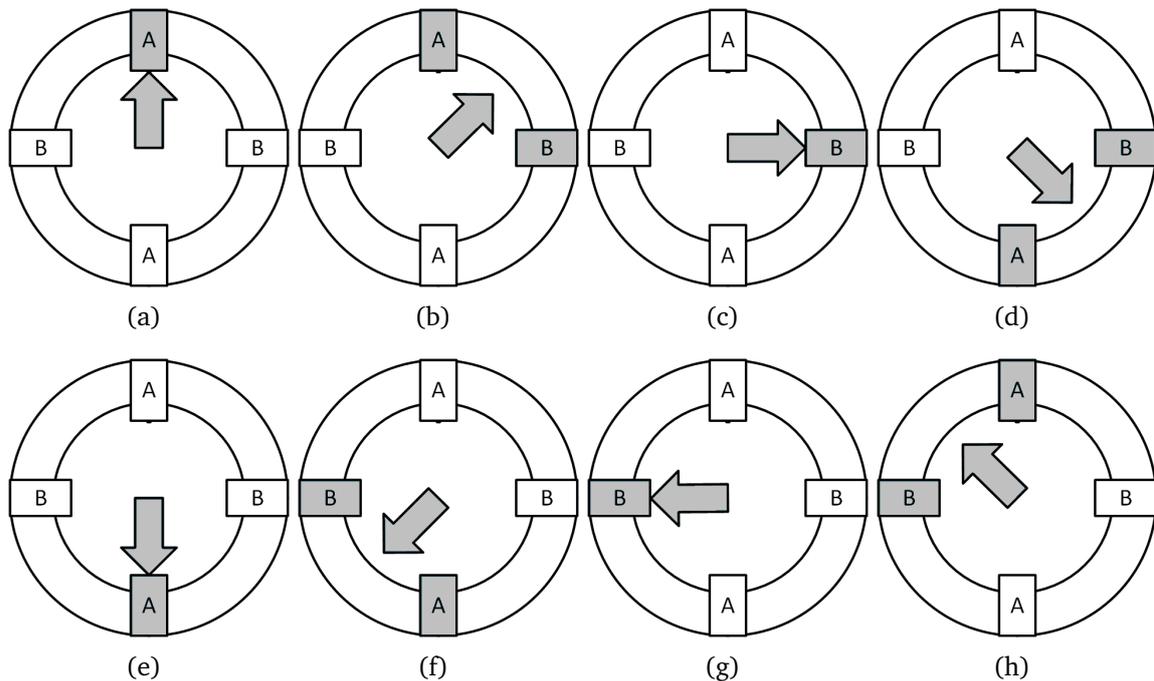
Für einen Vollschrittbetrieb werden die Motorspulen, wie in Abbildung 2.9 vereinfacht gezeigt, ein- beziehungsweise ausgeschaltet. Dadurch dreht sich der Rotor immer um einen festen, von der Konstruktion des Motors abhängigen Winkel  $\varphi$ . Bei einem zweipoligen Motor mit 50 Rotorzähnen sind das beispielsweise  $1.8^\circ$  (siehe Formel 2.2). Der Motor hat hier großes Drehmoment [66, S. 12, S.76-79].



**Abbildung 2.9** Verlauf des erzeugten magnetischen Feldes bei Verwendung des Vollschrittbetriebs. A und B repräsentieren dabei die beiden Motorphasen in einem bipolaren Schrittmotor.

#### Halbschritt

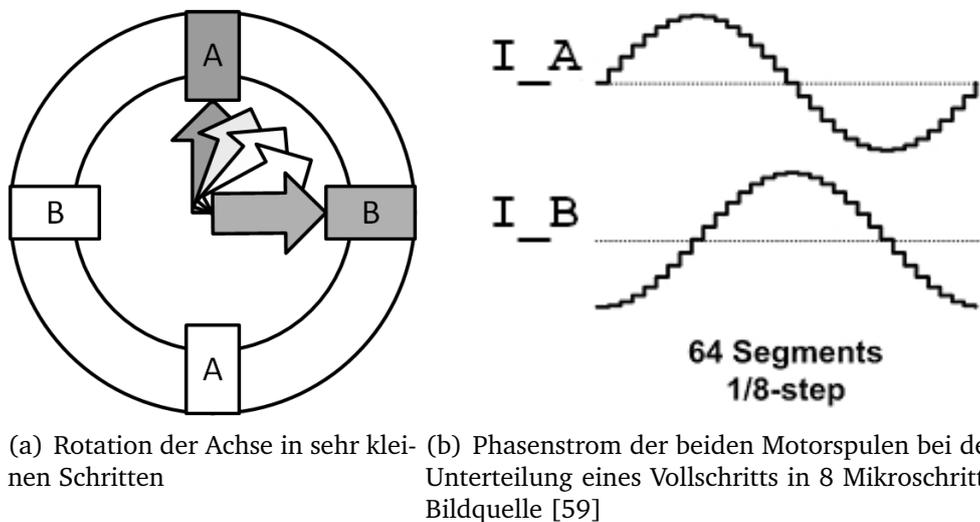
Für einen Halbschrittbetrieb werden die Motorspulen, wie in Abbildung 2.10 vereinfacht gezeigt, geschaltet. Im Gegensatz zum Vollschrittbetrieb gibt es beim Halbschrittbetrieb Phasen, in denen nur eine Motorspule bestromt ist. Das führt in diesen Phasen zu einem geringeren Drehmoment. Manche Schrittmotortreiber gleichen das aus, indem sie in diesen Phasen mehr Strom durch die einzelne Motorspule fließen lassen. Durch diese Betriebsart kann die Schrittauflösung verdoppelt werden, ein zweipoliger Motor mit 50 Rotorzähnen, also 200 Vollschritten pro Umdrehung, erreicht dadurch einen Schrittwinkel von  $0.9^\circ$  [66, S. 12, S.76-79].



**Abbildung 2.10** Verlauf des erzeugten magnetischen Feldes bei Verwendung des Halbschrittbetriebs. A und B repräsentieren dabei die beiden Motorphasen in einem bipolaren Schrittmotor.

### Mikroschritt

Mikroschrittbetrieb steht für eine Vielzahl an Betrieben, die die Vollschriffe des Motors weiter aufteilen. Der Mikroschrittbetrieb ist dadurch die Weiterentwicklung des Halbschrittbetriebs. Der Halbschrittbetrieb teilt jeden Vollschriff in zwei Halbschritte, es ist aber auch möglich, jeden Vollschriff in vier, acht oder mehr Teilschritte zu unterteilen. Dadurch steigt zwar einerseits der Aufwand, da nicht mehr jede Spule nur ein- oder ausgeschaltet werden muss, sondern je nach Rotorstand der einzelnen Motorspule ein bestimmter Strom eingepreßt werden muss. Andererseits wird dadurch eine höhere Schrittauflösung gewonnen. Ein weiterer Effekt ist, dass durch die kleineren Schritte der Schrittmotor leiser läuft [40].



**Abbildung 2.11** Mikroschrittauflösung. Im Gegensatz zum Vollschritt, werden beim Mikroschritt die Motorspulen nicht nur an- und ausgeschaltet, sondern mit einem bestimmten Strom durchflossen. Dieser Strom soll einer Sinuskurve entsprechen, siehe b). Dabei nähert sich der Stromverlauf einer idealen Sinuskurve immer näher an, je mehr Zwischenschritte genutzt werden.

### 2.1.5 Gegen-elektromotorische Kraft (BEMF)

Die gegen-elektromotorische Kraft (im englischen *back electromotive force (BEMF)*) ist die Kraft, die durch die Rotation des Motors generierte Spannung hervorgerufen wird und der Rotation des Motors entgegenwirkt. Sie beruht auf der Lenz'schen Regel, die besagt, dass wenn durch ein sich veränderndes magnetisches Feld in einer Leiterschleife eine Spannung induziert wird, fließt der daraus resultierende Strom in die Richtung, sodass das dadurch erzeugte Magnetfeld dem induzierenden Magnetfeld entgegen wirkt [52]. Daraus resultiert, dass mit zunehmender Drehzahl die induzierte Spannung im Schrittmotor steigt, diese Spannung der angelegten Versorgungsspannung entgegen wirkt und dadurch der Stromfluss in dem Schrittmotor behindert wird [66, S. 225]. Dadurch bricht bei Schrittmotoren das Drehmoment mit zunehmender Drehzahl ein. Eine Möglichkeit, dem entgegenzuwirken und das Drehmoment länger aufrechtzuerhalten, ist es, die Versorgungsspannung am Schrittmotor zu erhöhen, um über einen größeren Drehzahlbereich der induzierten Spannung entgegenwirken zu können [66, S. 225] [19]. Ein beispielhafter Drehmomentverlauf bei unterschiedlichen Versorgungsspannungen ist in Abbildung 2.4 dargestellt. Dieses Prinzip greift diese Masterarbeit auf.

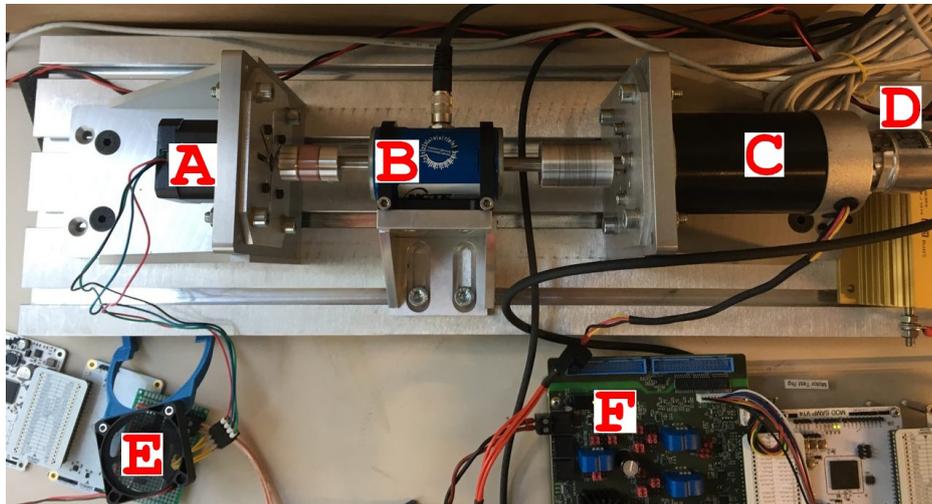
### 2.1.6 Motorteststand

Um eine Drehmomentkennlinie wie in Abbildung 2.4 aufnehmen zu können, muss der entsprechende Schrittmotor mithilfe eines Motorteststands vermessen werden. In Abbildung 2.12 ist der Motorteststand abgebildet, der zur Aufnahme der Kennlinien in Kapitel 5.2 verwendet wurde. Er wurde im Rahmen einer Masterarbeit bei TRINAMIC entwickelt. Der Motorteststand besteht aus folgenden Komponenten:

- **Testmotor:**  
Der Testmotor (in Abbildung 2.12 links im Bild) ist der Motor, dessen Drehmomentverlauf aufgezeichnet werden soll.
- **Drehmomentsensor:**  
Der Drehmomentsensor (in Abbildung 2.12 mittig im Bild) sitzt zwischen dem Testmotor und dem Lastmotor und misst das erzeugte Drehmoment zwischen den beiden Motoren.
- **Lastmotor mit Encoder:**  
Der Lastmotor (in Abbildung 2.12 rechts im Bild) wird zur Belastung des Testmotors eingesetzt, indem er so eingestellt wird, dass er langsamer laufen soll, als die vom Testmotor zu erreichende Drehzahl. Zur Überprüfung, ob der Testmotor seine Soll Drehzahl erreicht oder ob der Lastmotor es schafft, den Testmotor zu verlangsamen, wird die Drehzahl mithilfe eines Encoders gemessen.

Um das dynamische, drehzahlabhängige Drehmoment messen zu können, wird der Testmotor zuerst ohne Last auf die gewünschte Drehzahl beschleunigt. Nachdem diese erreicht wurde, beginnt der Lastmotor langsam den Testmotor abzubremesen. Der Drehmomentsensor misst dabei das entstehende Drehmoment. Wird mithilfe des Encoders eine Verringerung der Drehzahl festgestellt, der Testmotor kann die gewünschte Drehzahl unter der Last also nicht mehr halten, wird das dabei erreichte Drehmoment als maximal erreichbares Drehmoment für diese Drehzahl festgestellt. Da es durch den mechanischen Aufbau sowie der Eigenschaften der Motoren zu Resonanzen kommen kann, werden meistens mehrere Messungen für eine Drehzahl durchgeführt, um Ausreißer zu erkennen. Dieser Vorgang wird dann für verschiedenen Drehzahlen innerhalb des gewünschten Drehzahlbereiches durchgeführt. Die in Kapitel 5.2 durchgeführten Messungen wurden beispielsweise mit jeweils fünf Wiederholungen pro Drehzahl für insgesamt 25 Drehzahlen zwischen  $50 \text{ min}^{-1}$  und  $2000 \text{ min}^{-1}$  durchgeführt. Da die Beschleunigungsrate des Testmotors und die Entschleunigungsrate durch den Lastmotor einen größeren Einfluss auf die Messungen haben, sollten diese relativ niedrig gewählt werden, um ein vorzeitiges Einbrechen der Drehzahl zu ermöglichen. Dadurch dauern die Messungen für einen größeren Drehzahlbereich teilweise recht lange. Die in Kapitel 5.2 durchgeführten Messungen

dauerten beispielsweise zum Teil zwei Stunden pro Kennlinie. Weiterhin erzeugt der Drehmomentsensor einen Widerstand, den der Testmotor überwinden muss. Dies ist gerade bei sehr kleinen, drehmomentschwachen Motoren zu beachten.



**Abbildung 2.12** Der verwendete Motorteststand zur Aufnahme der Kennlinien in Kapitel 5.2. **A** ist der zu testende Motor, **B** ist der Drehmomentsensor, **C** ist der Lastmotor, **D** ist der Encoder, **E** ist die Steuerelektronik des zu testenden Motors, **F** ist die Elektronik, die die Lastmaschine ansteuert, sowie den Encoder und den Drehmomentsensor auswertet.

## 2.2 Gleichspannungswandler

Die Gruppe der Gleichspannungswandler umfasst viele verschiedenen Schaltungen, die eine Gleichspannung in eine höhere, niedrigere oder invertierte Gleichspannung wandeln können. Da in dieser Arbeit ausschließlich Aufwärtswandler entwickelt und verwendet werden, wird auf andere Gleichspannungswandler nicht weiter eingegangen und auf entsprechende Fachliteratur verwiesen [65].

### 2.2.1 Aufbau eines Aufwärtswandlers

Der grundsätzliche Aufbau eines Aufwärtswandlers (siehe Abbildung 2.13) besteht aus einer Speicherinduktivität, die über einen periodisch schließenden Schalter geladen wird und sich, während der Schalter geöffnet ist, durch einen zweiten Schalter, der sich, entgegengesetzt zu dem ersten Schalter, öffnet und schließt (häufig wird statt eines aktiv gesteuerten Schalters eine Diode verwendet), in einen Kondensator

entlädt. Dabei wird der Kondensator mit einer höheren Spannung geladen, als am Eingang des Aufwärtswandlers anliegt. Der genaue Ablauf der einzelnen Schaltphasen wird im Folgenden erklärt.

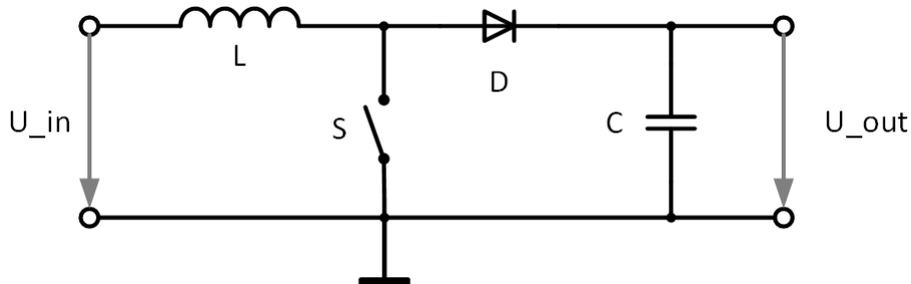


Abbildung 2.13 Grundlegender Aufbau eines Aufwärtswandlers

### 2.2.2 Schaltzustände

In einem Aufwärtswandler kann der Schalter, zum Beispiel ein MOSFET, zwei Zustände einnehmen, offen oder geschlossen. Je nach Schalterstellung wird dabei die Induktivität geladen oder entladen. Beide Phasen werden im Folgenden erläutert.

#### Entladephase

In der Entladephase ist der Schalter geöffnet (siehe Abbildung 2.14b). Das abrupte Öffnen des Schalters beendet den Stromfluss durch die Induktivität. Diese erzeugt dann durch ihre gespeicherte magnetische Energie eine Spannung, die entgegen der angelegten Spannung gepolt ist (Lenz'sche Regel [52]):

$$u_L = L \cdot \frac{di}{dt} \quad (2.3)$$

Die erzeugte Spannung  $u_L$  liegt in Reihe mit der Eingangsspannung  $U_{IN}$  und lädt durch die Diode den Ausgangskondensator  $C_{OUT}$  auf, da der Weg durch den geöffneten Schalter nicht möglich ist, und versorgt gleichzeitig die Last am Ausgang. Die generierte Ausgangsspannung ist dabei wie folgt:

$$U_{OUT} = U_{IN} + u_L - U_D \quad (2.4)$$

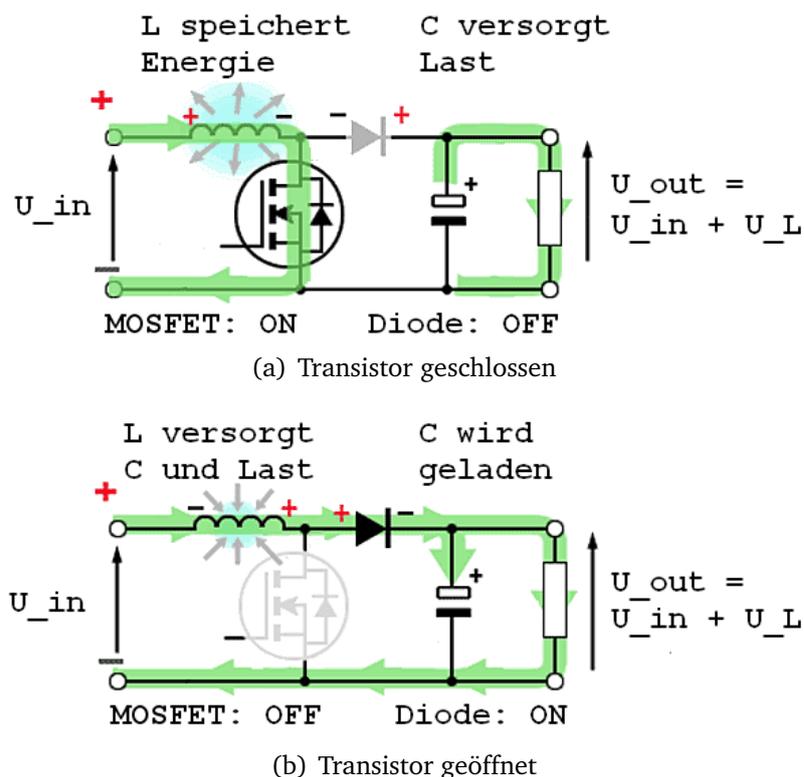
Dabei ist  $U_D$  der Spannungsabfall an der Diode.

### Ladephase

In der Ladephase ist der Schalter geschlossen (siehe Abbildung 2.14a). Die Eingangsspannung  $U_{IN}$  liegt durch den geschlossenen Schalter an der Induktivität an. Da die Eingangsspannung eine Gleichspannung ist, steigt der Strom durch die Spule linear mit der Zeit an:

$$\frac{di}{dt} = \frac{u_L}{L} \quad (2.5)$$

Gleichzeitig verhindert die Diode ein Entladen der in dem Ausgangskondensator  $C_{OUT}$  gespeicherten Ladung, da die an  $C_{OUT}$  anliegende Spannung  $U_{OUT} = U_{IN} + u_L - U_D$  höher ist, als die an dem geschlossenen Schalter. Sie entlädt sich dadurch über die Last am Ausgang. Aus diesem Grund ist bei Aufwärtswandlern eine minimale Grundlast am Ausgang für einen sicheren Betrieb erforderlich. Ohne diese könnte sich der Ausgangskondensator nicht entladen, beziehungsweise nur über parasitäre Widerstände anderer Komponenten. Die Spannung könnte im Leerlauf im schlimmsten Fall so groß werden, dass verwendete Komponenten zerstört werden können. Da die Spannung an  $C_{OUT}$  mit abnehmender Ladung sinkt, muss der Ausgangskondensator durch einen erneuten Entladevorgang die Induktivität geladen werden.



**Abbildung 2.14** Schaltzustände des Aufwärtswandlers. In a) ist der Ladevorgang dargestellt, in b) der Entladevorgang. Bilder modifiziert nach [13]

### 2.2.3 Continuous- und Discontinuous-Conduction Mode

Wie in Kapitel 2.2.2 beschrieben, gibt es zwei Zustände, die der Schalter annehmen kann. Je nach Schaltfrequenz, der Größe der Induktivität sowie der Last, ergeben sich daraus zwei mögliche Betriebsarten, der Continuous-Conduction Mode (CCM)<sup>3</sup> und der Discontinuous-Conduction Mode (DCM)<sup>4</sup>. Beide werden im Folgenden, unter Annahme von idealen Komponenten, beschrieben.

#### Continuous-Conduction Mode (CCM)

Im CCM gibt es zwei Phasen, einmal die Phase mit geschlossenem Schalter (Ladephase) und die des geöffneten Schalters (Entladephase). In der Ladephase steigt der Strom durch die Induktivität  $i_L$  an und es wird Energie in Form eines Magnetfeldes gespeichert. Durch den Schalter fließt in dieser Phase  $i_{SW} = i_L$ . Das ist in Abbildung 2.15 zu erkennen. Gleichzeitig fällt an dem Schalter keine Spannung ab und die Diode leitet nicht, führt also keinen Strom. Während dieser Zeit fällt über die Induktivität die Eingangsspannung  $U_{IN}$  ab, über die Diode die Ausgangsspannung  $U_{OUT}$ . In der Entladephase, bei geöffnetem Schalter, entlädt sich die Induktivität durch die Diode in den Ausgangskondensator und die Last, es fließt kein Strom durch den Schalter. Der Strom  $i_L$  sinkt dabei ab, wobei der Strom durch die Diode  $i_D = i_L$  ist (siehe Abbildung 2.15). Die Spannung über der Induktivität ist dabei die Differenz von Ausgangs- und Eingangsspannung und die Spannung über dem geöffneten Schalter ist die Ausgangsspannung. Das Merkmal des CCM ist, dass der Strom durch die Induktivität  $i_L$  nicht auf Null sinkt, sondern durchgehend, auch zum Ende der Entladephase, eine Restenergie vorhanden ist, wodurch die Spannung über den Schalter  $U_{OUT}$  entspricht und damit größer als  $U_{IN}$  ist. Dadurch leitet die Diode die ganze Entladephase durch und es wird während der gesamten Phase Energie von der Induktivität in den Ausgangskondensator transferiert. Daher kommt auch der Name, da kontinuierlich entweder der Schalter oder die Diode leitet. Die beiden Steigungen von  $i_L$  ergeben sich wie folgt:

$$m_{i_L|ON} = \frac{U_{IN}}{L} \quad (2.6)$$

$$m_{i_L|OFF} = \frac{U_{IN} - U_{OUT}}{L} \quad (2.7)$$

Wenn man davon ausgeht, dass der magnetische Fluss ( $V \cdot s$ ) bei geschlossenem Schalter gleich dem bei geöffnetem Schalter ist, ergibt sich folgende Gleichung:

$$\frac{U_{in}}{L} \cdot D \cdot T_S = \frac{U_{in} - U_{OUT}}{L} \cdot (1 - D) \cdot T_S \quad (2.8)$$

<sup>3</sup>Englisch für kontinuierlich leitender Modus

<sup>4</sup>Englisch für diskontinuierlich leitender Modus

Wenn man die Gleichung nach dem Duty-cycle  $D$  umstellt ergibt sich folgende Gleichung:

$$D_{CCM} = 1 - \frac{U_{IN}}{U_{OUT}} \quad (2.9)$$

Hiermit kann der Duty-cycle eines idealen Aufwärtswandlers für eine gewünschte Ausgangsspannung berechnet werden [56, S. 2-4].

Der CCM hat folgende Vorteile:

- Der Eingangsstrom, der dem versorgenden Netzteil entnommen wird, pulsiert nur relativ schwach (siehe Abbildung 2.15 bei  $i_L$ )
- Der Ripplestrom auf  $i_L$  ist geringer als der Mittelwert von  $i_L$
- Es kann ein besserer Wirkungsgrad erzielt werden als mit dem DCM [56, S. 5]

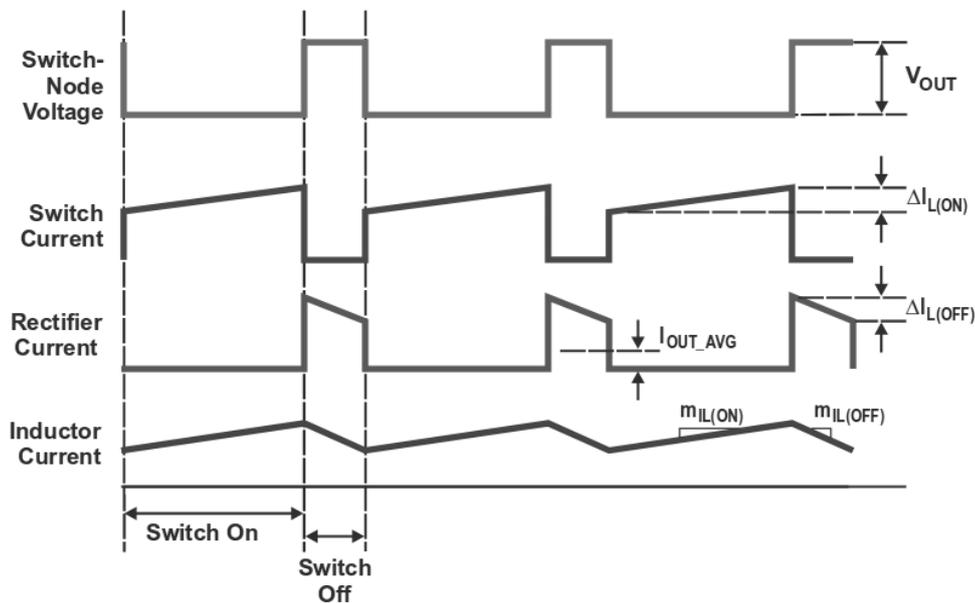


Abbildung 2.15 Strom- und Spannungsverlauf im CCM  
Bildquelle [56, S. 3]

### Discontinuous-Conduction Mode (DCM)

Im DCM gibt es drei Phasen, eine zusätzlich zu den beiden aus dem CCM. Die ersten beiden Phasen sind die gleichen wie bei dem CCM, die Lade- und die Entladephase. Die dritte Phase beginnt, wenn die Energie, die in der Induktivität gespeichert wurde, aufgebraucht ist und der Strom  $i_L$  auf Null abgesunken ist (siehe Abbildung 2.16). Während dieser Phase sinkt die Spannung über die Induktivität auf null ab,

was zur Folge hat, dass die Spannung über dem geöffneten Schalter auf  $U_{IN}$  absinkt. Dadurch leitet die Diode nicht mehr. Daher kommt auch hier der Name, da Schalter und Diode diskontinuierlich leiten. Während dieser dritten Phase wird die Last allein durch den Ausgangskondensator versorgt. Um die gleiche Energie in der gleichen Zeit im DCM zu übertragen wie im CCM, müssen die Spitzenströme durch die Diode, die Induktivität und den Schalter höher sein, da in der dritten Phase kein Energietransfer stattfinden kann. Wie der Duty cycle für eine gewünschte Ausgangsspannung im DCM zu berechnen ist, wird im Folgenden beschrieben. Zuerst wird die Zeit  $t_{fall}$  berechnet, die benötigt wird, um die Induktivität zu entladen, wenn sie in der Lade phase maximal geladen wurde:

$$t_{fall} = \frac{m_{i_L}|_{ON}}{-m_{i_L}|_{OFF}} \cdot D_{DCM} \cdot T_S \quad (2.10)$$

Wenn  $t_{fall}$  kleiner ist, als die Zeit, in der der Schalter geöffnet ist, arbeitet der Aufwärtswandler im DCM. Um den Duty cycle  $D_{fall}$  für die Entladezeit  $t_{fall}$  zu berechnen wird folgende Formel verwendet:

$$D_{fall} = \frac{t_{fall}}{T_S} \quad (2.11)$$

Durch Einsetzen von (2.10) ergibt sich folgende Gleichung:

$$D_{fall} = \frac{m_{i_L}|_{ON}}{-m_{i_L}|_{OFF}} \cdot D_{DCM} \quad (2.12)$$

Durch weiteres Einsetzen von (2.6) und (2.7) ergibt sich:

$$D_{fall} = \frac{U_{IN}}{U_{OUT} - U_{IN}} \cdot D_{DCM} \quad (2.13)$$

Es wird angenommen, dass bei einem stetigen Betrieb der durchschnittliche Ausgangsstrom  $I_{OUT}$  gleich dem durchschnittlichen Diodenstrom ist. Dieser durchschnittliche Ausgangsstrom entspricht dem Spitzenstrom, verteilt über die Schaltzeit:

$$I_{OUT} = \frac{U_{IN} \cdot T_S}{2 \cdot L} \cdot D_{DCM} \cdot D_{fall} \quad (2.14)$$

Umgestellt nach  $D_{DCM}$  ergibt sich folgende Gleichung [56, S. 4-5]:

$$D_{DCM} = \frac{1}{U_{IN}} \cdot \sqrt{\frac{2 \cdot L \cdot (U_{OUT} - U_{IN})}{T_S} \cdot I_{OUT}} \quad (2.15)$$

Der DCM hat folgende Vorteile:

- Der Schalter schaltet zu einem Zeitpunkt ein, an dem kein Strom mehr durch die Induktivität fließt, wodurch Schaltverluste beim Einschalten verringert werden
- Die Größe der Induktivität kann im Vergleich zu dem CCM reduziert werden, da diese in Phase drei keine Leistung bereitstellen muss

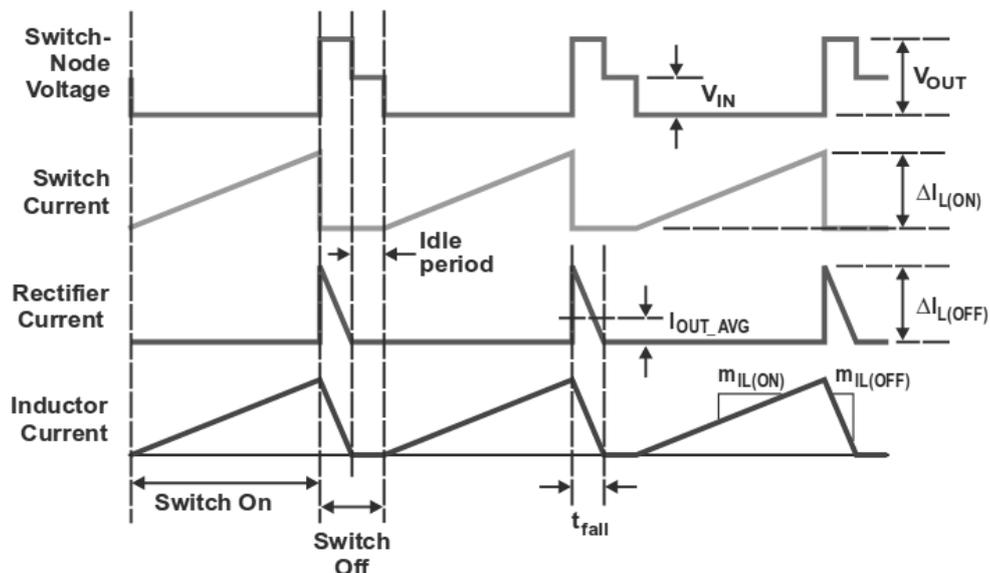


Abbildung 2.16 Strom- und Spannungsverlauf im DCM  
Bildquelle [56, S. 4]

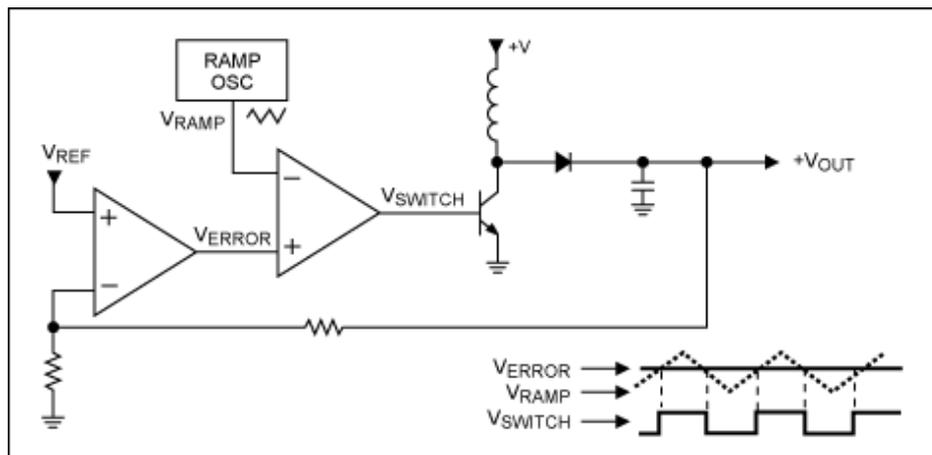
## 2.2.4 Arten der Regelung

Um die gewünschte Sollspannung zu generieren, muss eine Regelung der Schaltzeiten stattfinden. Dafür kommen mehrere Arten in Frage, welche im Folgenden beschrieben werden.

### Pulsweitenmodulation (PWM)

Eine gängige Regelungsart ist die Verwendung einer Pulsweitenmodulation des Schalttransistors. Dabei erfolgt die Auswertung und Generierung der Schaltimpulse in vielen entsprechenden ICs in analoger Form. Eine Variante ist dabei die spannungsgesteuerte PWM (*voltage-mode controller (VMC)*), deren prinzipieller Aufbau in Abbildung 2.17 dargestellt ist. Dabei wird die erzeugte Ausgangsspannung über

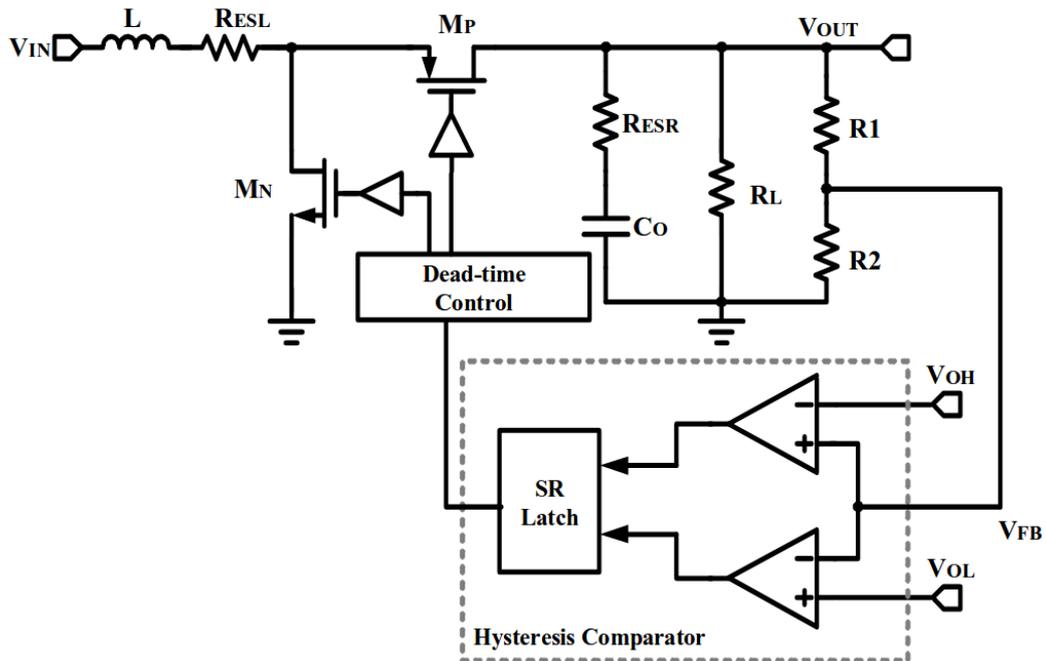
einen Spannungsteiler von einer Referenzspannung subtrahiert. Die Differenzspannung oder Fehlerspannung wird mit einer oszillierenden Rampenspannung verglichen und erzeugt so ein Rechtecksignal. Dabei hängt das Tastverhältnis von der Fehlerspannung ab. Dadurch wird wiederum die Ausgangsspannung geändert. Diese Regelung gibt konstant ein Rechtecksignal auf den Schalter, wenn auch teilweise mit einem Tastverhältnis gegen null. Der Aufwärtswandler wird also mit einer konstanten Schaltfrequenz betrieben [31] [37].



**Abbildung 2.17** Schematischer analoger Aufbau der Regelung eines Aufwärtswandlers mithilfe der Pulsweitenmodulation. Bildquelle [31]

### Hysterese-Regelung

Eine weitere Art der Regelung ist die Hysterese-Regelung. Hier wird die Ausgangsspannung über einen Spannungsteiler mit einer Referenzspannung verglichen (siehe Abbildung 2.18). Wenn die erzeugte und geteilte Ausgangsspannung  $U_{FB}$  größer als die Referenzspannung ist, wird der Transistor nicht eingeschaltet, die Induktivität wird nicht geladen. Das passiert so lange, bis sich die Ausgangsspannung durch die angelegte Last auf etwas unter die Sollspannung abgesenkt hat. Dann ist  $U_{FB}$  kleiner als die Referenzspannung und es wird ein einzelner Rechteckimpuls mit einer festen Einschaltzeit und einer festen Ausschaltzeit auf den Transistor gegeben. Danach erfolgt wieder ein Vergleich von  $U_{FB}$  mit der Referenzspannung und je nach Ergebnis wird wieder ein einzelner Rechteckimpuls auf den Transistor gegeben. Die feste Ausschaltzeit ist wichtig, damit der Transistor nicht für eine längere Zeit konstant eingeschaltet ist und dadurch einen Kurzschluss verursacht. Diese Regelung lädt die Induktivität nur auf, wenn Energie am Ausgang fehlt. Der Aufwärtswandler wird also mit einer variablen Schaltfrequenz betrieben [31] [38] [27] [22].



**Abbildung 2.18** Schematischer Aufbau der Regelung eines Aufwärtswandlers bei Verwendung der Hysterese-Regelung mit konstanter Einschaltzeit. Bildquelle [22, S. 67]

## 2.2.5 Komponentenauswahl

Je nach Leistung, Ein- und Ausgangsspannung sowie Anforderungen an Restwelligkeit werden unterschiedliche Bauelemente für den Aufwärtswandler ausgewählt. Wie die Bauteile bestimmt werden, wird im Folgenden beschrieben.

### Auswahl der Induktivität

Für die Auswahl der Induktivität wird nach [20, S. 3] folgende Formel empfohlen:

$$L = \frac{U_{IN} \cdot (U_{OUT} - U_{IN})}{\Delta I_L \cdot f_s \cdot U_{OUT}} \quad (2.16)$$

Dabei steht  $f_s$  für die minimale Schaltfrequenz des Transistors.  $\Delta I_L$  steht für den Ripplestrom auf dem Strom durch die Induktivität  $i_L$ . Er kann über folgende Formel abgeschätzt werden, wobei  $I_{OUT|MAX}$  für den maximalen Strom, der aus dem Aufwärtswandler entnommen werden soll, steht:

$$\Delta I_L = (0.2 \dots 0.4) \cdot \max(I_{OUT}) \cdot \frac{U_{OUT}}{U_{IN}} \quad (2.17)$$

### Auswahl der Diode

Da die Verluste durch die Diode neben dem Laststrom von der Schwellspannung der Diode abhängt, ist eine Diode mit möglichst niedriger Schwellspannung  $U_D$ , zum Beispiel eine Schottky-Diode, zu wählen. Die gewählte Diode sollte für folgende Leistung ausgelegt werden [20, S. 3]:

$$P_D = U_D \cdot \max(I_{OUT}) \quad (2.18)$$

### Auswahl der Ausgangskapazität

Da der Ausgangskondensator konstant geladen und entladen wird, sollte eine Bauart mit möglichst geringem äquivalenten Serienwiderstand (ESR) gewählt werden, um Verluste zu verringern. Die minimale Kapazität des Kondensators kann über folgende Gleichung bestimmt werden, wobei  $\Delta U_{OUT}$  für die gewünschte Rippelspannung steht [20, S. 5]:

$$C_{MIN} = \frac{\max(I_{OUT}) \cdot D}{f_S \cdot \Delta U_{OUT}} \quad (2.19)$$

Eine zusätzliche Rippelspannung  $\Delta U_{ESR}$  wird durch den ESR des Ausgangskondensators hervorgerufen [20, S. 5]:

$$\Delta U_{ESR} = ESR \cdot \left( \frac{\max(I_{OUT})}{1 - D} + \frac{\Delta I_L}{2} \right) \quad (2.20)$$

# 3 Entwicklung der Hardware

In diesem Kapitel soll die Herangehensweise sowie die Umsetzung der Entwicklung der Hardware beschrieben werden.

## 3.1 Vorgaben und Konzept

Für die Entwicklung des Moduls wurden seitens TRINAMIC folgende Vorgaben gefordert:

- Das Modul soll pin- und funktionskompatibel mit bestehenden Schrittmotortreibermodulen [64] sein, die im 3D-Druckermarkt verbreitet sind (siehe auch Kapitel 3.4).
- Es soll möglichst kein bestehender Gleichspannungswandler-IC verwendet werden, da das Interesse besteht, mithilfe des in dieser Masterarbeit gesammelten Wissens, einen neuen Schrittmotortreiber mit integriertem Spannungswandler zu entwickeln
- Die Leistungsfähigkeit des Spannungswandlers soll möglichst skalierbar sein

Als auszuwertendes Signal stand neben dem Schrittsignal noch das Tastverhältnis der Pulsweitenmodulation an den beiden H-Brücken der Schrittmotortreiber (siehe Kapitel 2.1.3) sowie die internen Geschwindigkeitsregister des Schrittmotortreibers zur Auswahl. Dabei wurde sich firmenseitig, für diese Abschlussarbeit, für die Auswertung des Schrittsignals entschieden, da es auf die meisten Schrittmotortreiber anwendbar ist. Die Auswertung der Pulsweite an den H-Brücken setzt dagegen voraus, dass der entsprechende Schrittmotortreiber mit extern angeschlossenen Schalttransistoren arbeitet, was eher im höheren Leistungsbereich und nicht im 3D-Druckermarkt verwendet wird. Für die Auswertung der internen Geschwindigkeitsregister müsste auf dieses schnell genug und unabhängig von der Ansteuerung zugegriffen werden können. Weiterhin unterscheidet sich dieses Register zum Teil je nach Treiber. Dadurch ergab sich das in Abbildung 3.1 dargestellte Konzept. Dabei fügt sich das Modul wie in Abbildung 3.2 gezeigt in ein Antriebssystem ein.

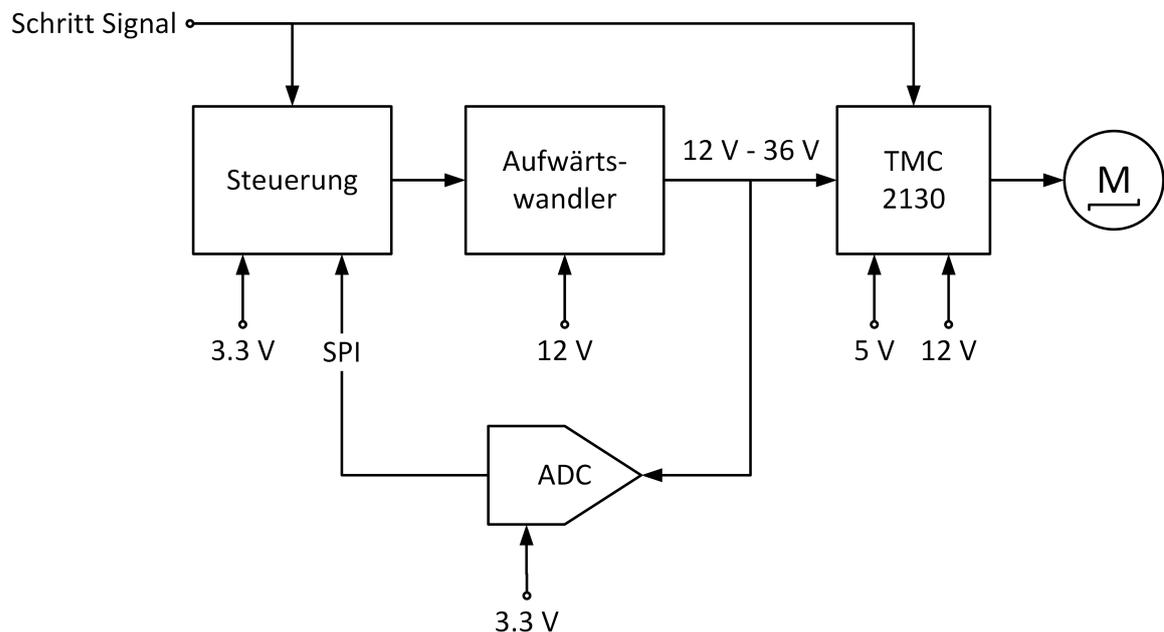


Abbildung 3.1 Konzept für die Entwicklung des Moduls

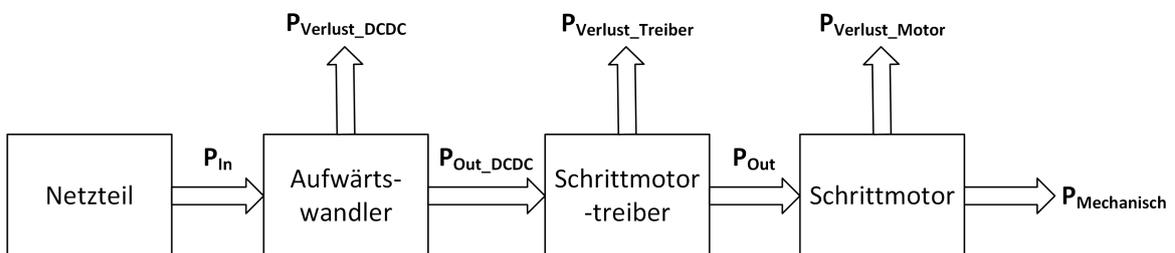


Abbildung 3.2 Energieflussdiagramm des Moduls inklusive der Verlustleistungen und der nutzbaren Ausgangsleistung

## 3.2 Simulation des Aufwärtswandlers

Bevor eine Hardware entwickelt wurde, wurden vorbereitende Schaltungssimulationen durchgeführt, um zu überprüfen, ob die Entwicklung eines diskreten Gleichspannungswandlers mit den gewünschten Anforderungen grundsätzlich funktionieren könnte. Dafür wurden zuerst die nötigen Parameter seitens TRINAMIC festgelegt:

- Eingangsspannung: 12 V
- Ausgangsspannung: 12 V bis 36 V
- Ausgangsleistung: max. 20 W

Als Gleichspannungswandler-Topologie wurde der Aufwärtswandler gewählt (siehe Kapitel 2.2), da die Ausgangsspannung nicht kleiner werden soll als die Eingangsspannung. Als Simulationssoftware wurde *LTSpice* von *Analog Devices Inc.* verwendet [23]. In Abbildung 3.3 ist die simulierte Schaltung dargestellt.

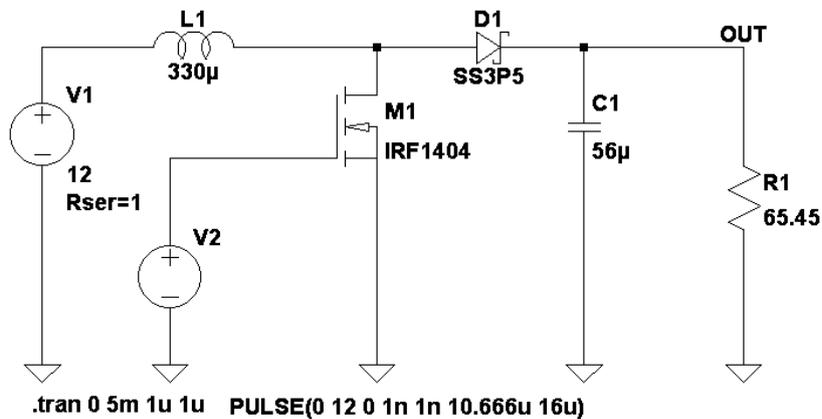


Abbildung 3.3 Schaltplan des simulierten Aufwärtswandlers in *LTSpice*

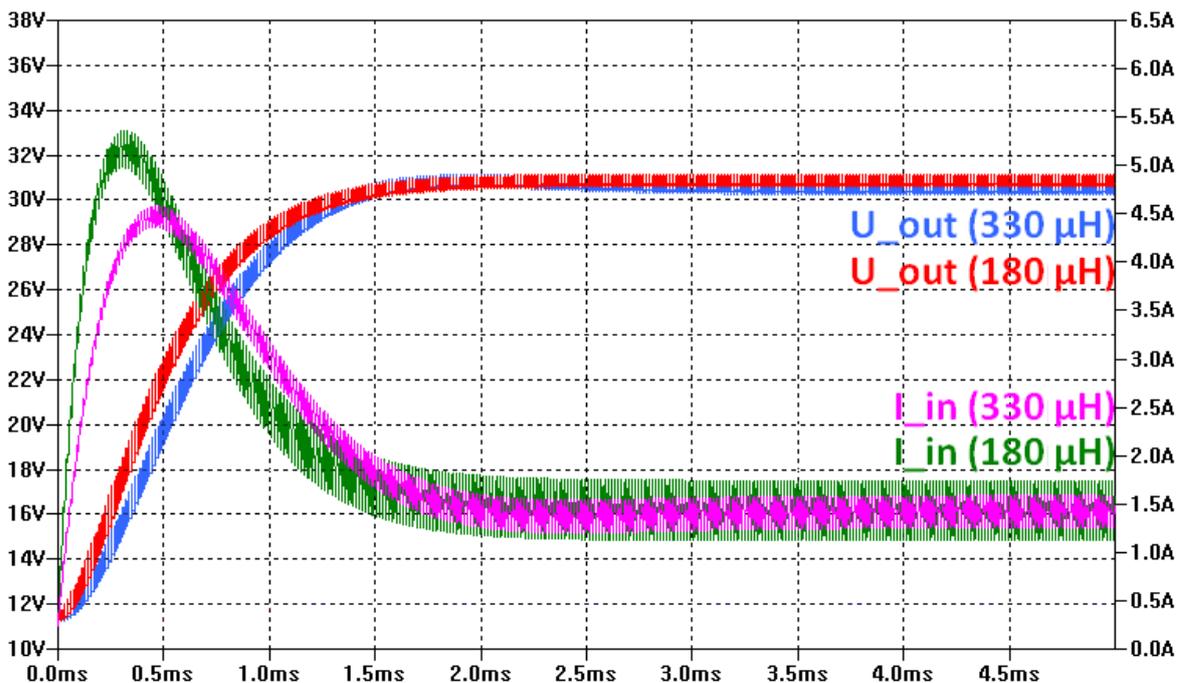
Da zu Beginn der Entwicklung noch nicht feststand, welche Art von Steuer- und Regellogik verwendet werden soll, war die letztendlich verwendbare maximale Schaltfrequenz noch nicht bekannt. Um jedoch einen Vergleich zwischen der Simulation und dem ersten Funktionsmuster (siehe Kapitel 3.3) zu ermöglichen wurde für einen Teil der Simulationen die maximale Frequenz verwendet, die ein zur Verfügung stehender *ATmega2560* von *Microchip Technology Inc.*, bei Nutzung einer PWM mit 8 Bit-Auflösung, liefern kann [26]. Das sind  $62.5 \text{ kHz}$ . Für diese Frequenz und die genannten Parameter wurden mithilfe der in Kapitel 2.2.5 beschriebenen Formeln die Induktivität sowie die Ausgangskapazität bestimmt. Weiterhin stand noch nicht fest, welche Art der Regelung verwendet werden soll, weshalb in der Simulation nur ein konstantes Rechtecksignal mit einem eingestellten Tastverhältnis an dem Transistor anliegt, es erfolgt keine Regelung anhand des erzeugten Ausgangssignals. Für eine Ausgangsspannung von  $36 \text{ V}$  bei einer Ausgangsleistung von  $20 \text{ W}$  und einer Rippelspannung von  $100 \text{ mV}$  ergeben sich folgende Werte:

- Ausgangsstrom  $I_{OUT} = 0.55 \text{ A}$
- Lastwiderstand  $R_{Last} = 65.45 \Omega$
- Duty cycle  $D = 0.66$
- Induktivität  $L = 193 \mu\text{H}$  (für  $\Delta I_L = 0.66 \text{ A}$ ) bis  $387 \mu\text{H}$  (für  $\Delta I_L = 0.33 \text{ A}$ )
- Ausgangskapazität  $C_{OUT} = 58.6 \mu\text{F}$

Als Werte wurden naheliegende Werte aus der E12-Reihe [2] gewählt. Um den Einfluss des äquivalenten Serienwiderstands (ESR) der Kapazität und des Gleichspannungswiderstands der Induktivität nicht auszuschließen, wurde sich an jeweils reale Bauelemente orientiert:

- $L_1$ :  $L=330 \mu\text{H}$ ,  $R_{L(DC)}=180 \text{ m}\Omega$ ,
- $L_2$ :  $L=180 \mu\text{H}$ ,  $R_{L(DC)}=82 \text{ m}\Omega$ ,
- $C_{OUT}$ :  $C=56 \mu\text{F}$ ,  $R_{ESR}=250 \text{ m}\Omega$

In Abbildung 3.4 ist das Einschalten der Schaltung zu sehen. Es ist zu erkennen, dass die vorher berechnete Ausgangsspannung von  $36 \text{ V}$  nicht erreicht wird, der eingeschwungene Zustand wird nach  $1.5 \text{ ms}$  erreicht und schwingt dann mit der Schaltfrequenz um  $31.5 \text{ V}$ . Weiterhin ist der Einschaltstromsprung zu Beginn zu erkennen, der Strom steigt kurzzeitig auf  $4.5 \text{ A}$  an und schwingt dann bei  $1.4 \text{ A}$  ein.



**Abbildung 3.4** Einschaltvorgang des simulierten, unregulierten Aufwärtswandlers. Simulationsparameter:  $f_S = 62.5 \text{ kHz}$ ,  $L = 330 \mu\text{H}$ ,  $R_{L(DC)} = 180 \text{ m}\Omega$ ,  $C = 56 \mu\text{F}$ ,  $R_{ESR} = 250 \text{ m}\Omega$ . Es ergibt sich eine Anstiegsdauer der Spannung von  $1.5 \text{ ms}$  und eine Rippelspannung von  $0.7 \text{ V}$  für beide Induktivitäten.

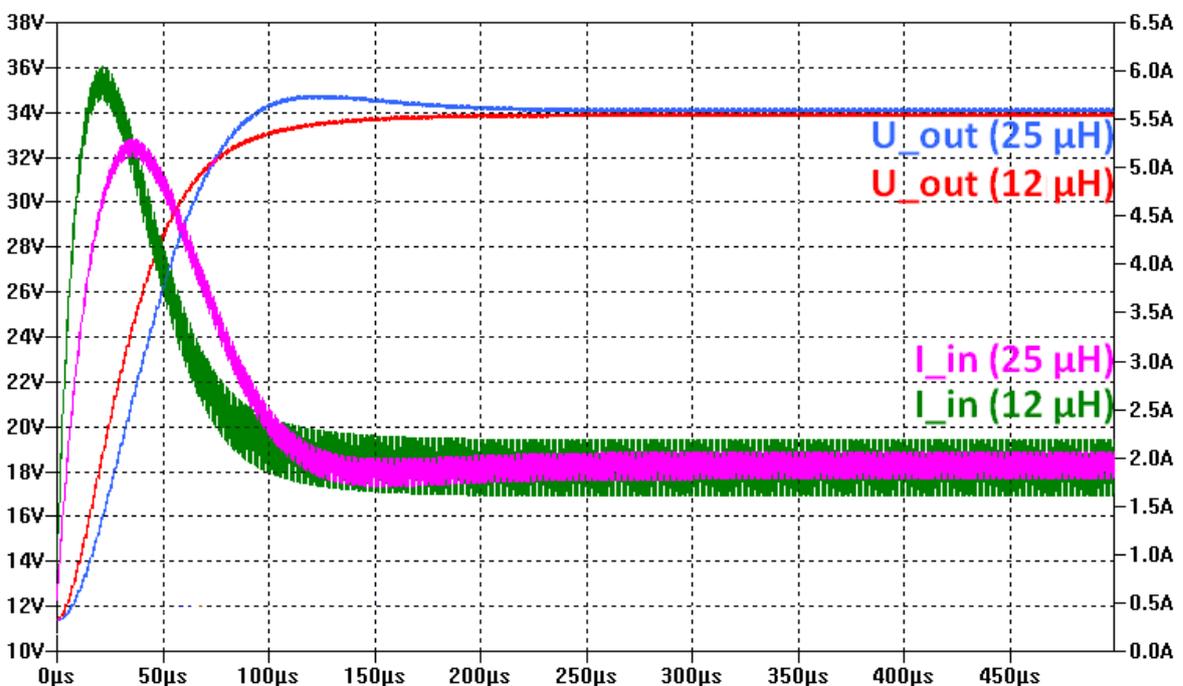
Um zu überprüfen, wie sich der Aufwärtswandler bei einer wesentlich höheren Schaltfrequenz von  $1\text{ MHz}$  verhält und ob dies irgendwelche Vorteile erzeugen würde, wurden die dafür nötigen Bauelementwerte berechnet und simuliert. Die berechneten Bauelementwerte sind wie folgt:

- Induktivität  $L = 12.12\ \mu\text{H}$  (für  $\Delta I_L = 0.66\text{ A}$ ) bis  $24.24\ \mu\text{H}$  (für  $\Delta I_L = 0.33\text{ A}$ )
- Ausgangskapazität  $C_{OUT} = 3.66\ \mu\text{F}$

Als Werte wurden naheliegende Werte aus der E12-Reihe [2] gewählt. Es wurde sich wieder an realen Bauelementen orientiert, um den Einfluss der äquivalenten Serienwiderstände (ESR) der Kapazität und des Gleichspannungswiderstands der Induktivität zu berücksichtigen:

- $L_1$ :  $L = 12\ \mu\text{H}$ ,  $R_{L(DC)} = 23\ \text{m}\Omega$ ,
- $L_2$ :  $L = 25\ \mu\text{H}$ ,  $R_{L(DC)} = 14\ \text{m}\Omega$ ,
- $C_{OUT}$ :  $C = 3.3\ \mu\text{F}$ ,  $R_{ESR} = 10\ \text{m}\Omega$

Das Ergebnis ist in Abbildung 3.5 dargestellt. Es ist zu erkennen, dass die Ausgangsspannung ihren eingeschwungenen Zustand schon nach  $150\ \mu\text{s}$  erreicht. Jedoch wird wieder nicht die berechnete Ausgangsspannung von  $36\text{ V}$  erreicht sondern  $34\text{ V}$ . Der Einschaltstrompuls ist ebenfalls deutlich kürzer als bei  $f_S = 62.5\ \text{kHz}$  und erreicht maximal  $6\text{ A}$ , danach schwingt er bei  $1.9\text{ A}$  ein.



**Abbildung 3.5** Einschaltvorgang des simulierten, unregulierten Aufwärtswandlers. Simulationsparameter:  $f_S = 1\text{ MHz}$ ,  $L = 12\ \mu\text{H}$ ,  $R_{L(DC)} = 23\ \text{m}\Omega$ ,  $C = 3.3\ \mu\text{F}$ ,  $R_{ESR} = 10\ \text{m}\Omega$ . Es ergibt sich eine Anstiegsdauer der Spannung von  $150\ \mu\text{s}$  und eine Rippelspannung von  $< 0.1\text{ V}$  für beide Induktivitäten. Damit ist die Anstiegszeit, im Vergleich zu der Simulation mit  $f_S = 62.5\ \text{kHz}$ , um den Faktor zehn gesunken. Ebenso ist die Rippelspannung von  $0.7\text{ V}$  auf weniger als  $0.1\text{ V}$  gesunken.

Aufgrund der Ergebnisse wurde entschieden, dass der *ATmega2560* wegen seiner langsamen maximalen PWM-Frequenz nicht verwendet wird. Stattdessen wurde sich für einen FPGA<sup>1</sup> entschieden, da ein höherer Grundtakt verwendet werden kann als die 20 MHz des *ATmega2560* sowie eine Anpassung der PWM-Auflösung möglich ist, was zu einer höheren maximalen PWM-Frequenz führt.

### 3.3 Entwicklung von Funktionsmustern

Um das Verhalten der Schaltung unter realen Bedingungen beobachten zu können, wurde zum Aufbau von Funktionsmustern übergegangen. Dabei wurden insgesamt drei verschiedene Testaufbauten angefertigt und ausgewertet. Die Ergebnisse werden im den folgenden Tabellen 3.1, 3.2 und 3.3 zusammengefasst dargelegt. Für eine ausführlichere Beschreibung der einzelnen Aufbauten sei auf den Anhang A verwiesen.

**Tabelle 3.1** Ergebnisse und Erkenntnisse des 1. Funktionsmuster

<b>1. Funktionsmuster</b>	
Aufbau	THT-Bauteile <sup>2</sup> auf Lochrasterplatine. Steuerung durch externen <i>ATmega2560</i> Mikrocontroller. Keine Messung der Ausgangsspannung mit dem Mikrocontroller, keine Regelung.
Vorteile	Induktivität und Ausgangskondensator durch Steckverbinder einfach für Testzwecke austauschbar.
Nachteile	Durch suboptimale Masseführung und elektromagnetischen Kopplungen über die vorhandenen Lötaugen sehr stark verrauschtes Ausgangssignal, Schaltvorgänge des MOSFET deutlich sichtbar.
Erkenntnisse	Bei Verwendung der in Abbildung 3.4 genutzten Bauteilwerte und Parameter, wurde ein Verhalten festgestellt, dass mit der Simulation vergleichbar ist. Allerdings ist die Ausgangsspannung stark verrauscht und sehr unsauber.

<sup>1</sup>Field Programmable Gate Array

<sup>2</sup>THT = Through Hole Technology, englisch für Durchsteckmontage

**Tabelle 3.2** Ergebnisse und Erkenntnisse des 2. Funktionsmuster

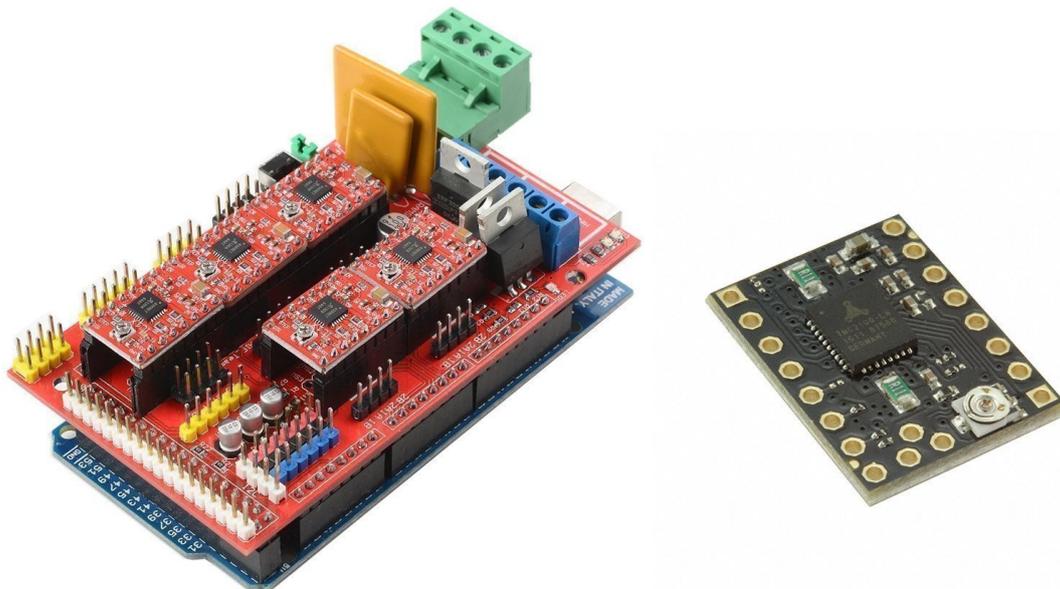
<b>2. Funktionsmuster</b>	
Aufbau	Frei verdrahtete SMD-Komponenten mit möglichst kurzen Masseverbindungen und sternförmiger Masseführung. Steuerung durch ein externes FPGA-Board [70] mit einer Schaltfrequenz von 1 MHz.
Vorteile	Saubereres Ausgangssignal durch verbesserte Masseführung, geringere Rippelspannung durch höhere Schaltfrequenz.
Nachteile	Austausch von Komponenten für Testzwecke sehr umständlich, keine Wärmeabführung durch eine Platine möglich.
Erkenntnisse	Masse- und Signalführung hat großen Einfluss auf Störungen des Ausgangssignals, hochfrequentes Schalten erzeugt geringere Rippelspannung, der FPGA in Verbindung mit dem verwendeten MOSFET-Treiber kann den Leistungs-MOSFET schnell genug schalten.

**Tabelle 3.3** Ergebnisse und Erkenntnisse des 3. Funktionsmuster

<b>3. Funktionsmuster</b>	
Aufbau	SMD-Komponenten auf Lochraster mit möglichst kurzen Masse- und Signalleitungen, ADC zur Auswertung der Ausgangsspannung, Auswertung des ADCs und Regelung des Spannungswandlers mithilfe eines externen FPGA-Boards [70].
Vorteile	Auswertung der Ausgangsspannung und daraus resultierende Regelung des Spannungswandlers möglich, dadurch ist die Entwicklung einer Regelung möglich.
Nachteile	Elektromagnetische Kopplung über die Lötungen der Platine, Austausch von Komponenten für Testzwecke umständlich.
Erkenntnisse	Auswertung der Ausgangsspannung mithilfe des ADCs und des FPGAs funktioniert, dadurch Regelung der Ausgangsspannung auf einen festen Sollwert möglich.

### 3.4 Entwicklung der Platine

Bei der Entwicklung der Platine, wurde das in Abbildung 3.1 dargestellte Baugruppenkonzept als Grundlage verwendet. Da das zu entwickelnde Modul mit vorhandenen Schrittmotortreibermodulen aus dem 3D-Druckermarkt pinkompatibel sein soll, wurde das Modul *SilentStepStick TMC2130* [11] von *Watterott electronic GmbH* als Grundlage genommen (siehe Abbildung 3.6b). Dieses basiert auf dem *TMC2130* [47] Schrittmotortreiber von TRINAMIC, der Motorstrom wird über ein Potentiometer eingestellt und das Modul kann entweder mit dem, im 3D-Druckermarkt verbreiteten, Step/Direction Interface oder per SPI angesteuert werden. Je ein Modul pro Achse des 3D-Druckers sowie pro Extruder wird dabei in eine Hauptplatine eingesteckt, die die weitere Elektronik für den 3D-Drucker enthält. in Abbildung 3.6a ist beispielhaft ein *RAMPS-Board* [63] mit fünf Schrittmotortreibermodulen abgebildet.



(a) Bestücktes *RAMPS-3D-Drucker Board*  
Bildquelle [61]

(b) *SilentStepStick Schrittmotormodul*  
Modul von *Watterott electronic GmbH*  
Bildquelle [12]

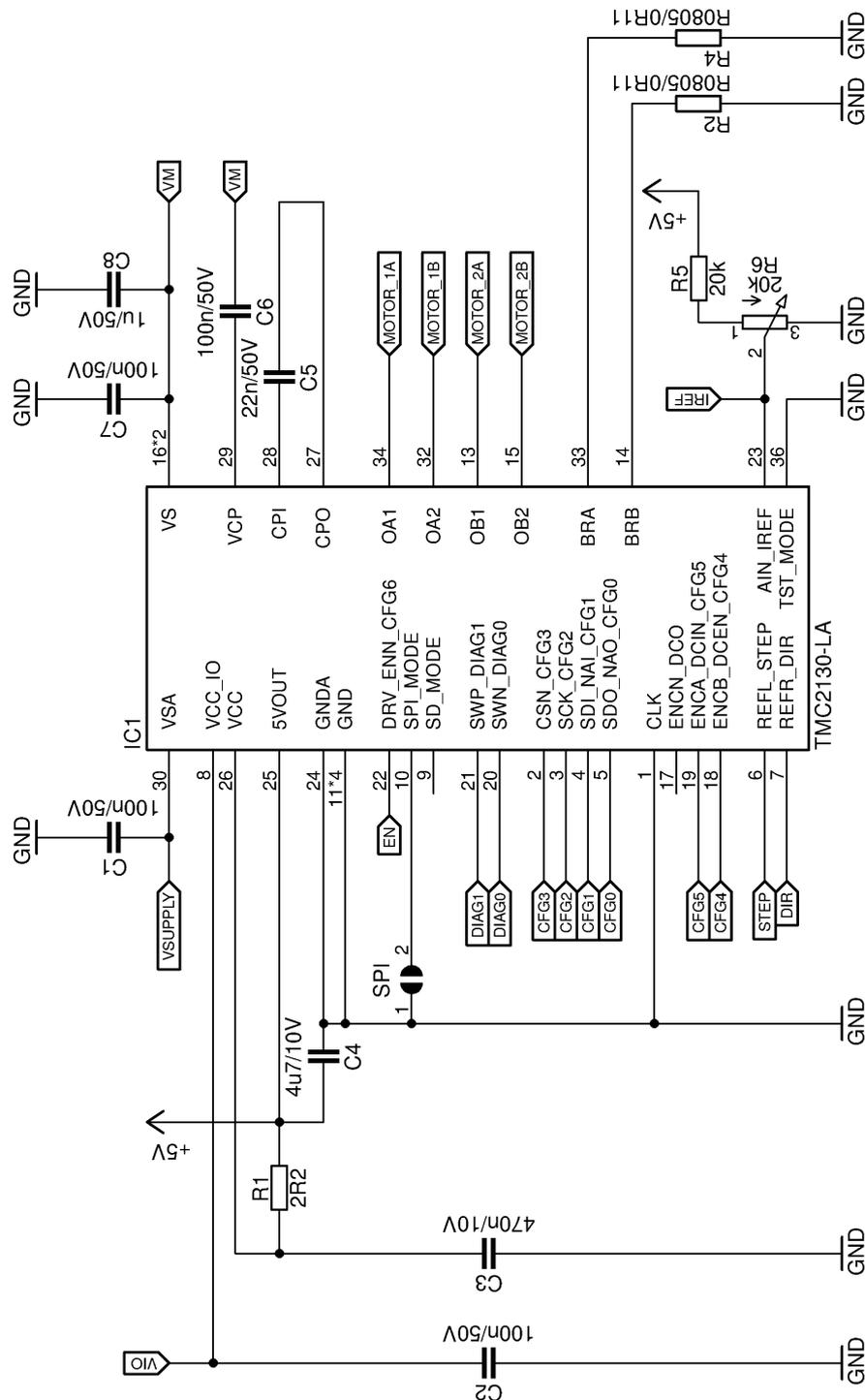
**Abbildung 3.6** Häufig verwendete Elektronik in 3D-Druckern

Diese Treibermodule haben eine Größe von  $1,524\text{ cm} \times 2,032\text{ cm}$  ( $0.6'' \times 0.8''$ ), wodurch neben dem Treiber-IC sowie einigen passiven Bauelementen und den Stiftleisten nicht viel Platz bleibt. Da neben diesen Komponenten noch der Aufwärtswandler inklusive des Logikteils und der Leistungselektronik auf dieser Fläche Platz finden

müsste, wurde nach anderen Möglichkeiten gesucht, das Modul mit genügend Fläche zu entwerfen und trotzdem mit den vorhandenen Schrittmotortreibermodulen pinkompatibel zu sein. Daraus wurde die Idee entwickelt, die Platine senkrecht auf die Grundfläche zu stellen. So ergibt sich eine Platine, die  $2,032\text{ cm}$  breit ist und in der Länge beziehungsweise Höhe keine festgelegte Begrenzung hat. Da die meisten 3D-Druckerboards mit  $5\text{ V}$  Logikspannung arbeiten, muss eine Spannungsanpassung für den FPGA vorhanden sein. Eine weitverbreitete Versorgungsspannung von 3D-Druckern ist  $12\text{ V}$ , weshalb das Modul darauf ausgelegt wird. Eine Übersicht der unterschiedlichen Versorgungsspannungen der einzelnen Baugruppen ist in Abbildung 3.1 dargestellt. Um eventuelle Fehlersuche und Messungen zu vereinfachen, wurden viele Signale über Testpunkte herausgeführt.

### 3.4.1 Schrittmotortreiber

Da, wie in Kapitel 3.4 beschrieben, das zu entwickelnde Modul auf dem *SilentStepStick TMC2130* von *Watterott electronic GmbH* basiert, wurde für diesen Teil der Platine deren Schaltplan als Vorlage genommen [11]. Es wurde der Großteil der Schaltung übernommen, da die Kompatibilität des zu entwickelten Moduls mit den bestehenden gewahrt werden sollte. Ein vereinfachter Schaltplan für den Schrittmotortreiberteil ist in Abbildung 3.7 gezeigt, der tatsächlich verwendete Schaltplan enthält zusätzliche Lötjumper zur Konfigurationen sowie Testpunkte und ist im Anhang unter Abbildung B.1 einsehbar. Angepasst wurde jedoch die Versorgung des VSA-Pins, da dieser die Versorgungsspannung für den chipinternen  $5\text{ V}$  linearen Spannungsregler bereitstellt. In der Schaltung des *SilentStepSticks* wird der Spannungsregler von derselben Spannung versorgt, die auch den Motortreiber über Pin VS versorgt. Da die Verlustleistung eines linearen Spannungsreglers direkt von der Differenz der Eingangsspannung zu der Ausgangsspannung abhängt [54], wird anstatt der Motorversorgungsspannung, welche von  $12\text{ V}$  bis  $36\text{ V}$  reichen kann, direkt die Versorgungsspannung von konstanten  $12\text{ V}$  verwendet. Dadurch werden unnötige Verluste und die daraus resultierende Wärmezeugung vermieden. Ansonsten entspricht die Originalschaltung von *Watterott electronic GmbH* zum größten Teil der Beispielschaltung aus dem Datenblatt des *TMC2130*. Der Motorstrom kann über das Potentiometer eingestellt werden. Dabei ist durch die verwendeten Strommesswiderstände von  $0.18\ \Omega$  ein maximaler Strom von  $0.8734\text{ A}$  möglich [47, S. 55]. Weiterhin ist über einen Lötjumper die Wahl des Interface möglich, es kann zwischen dem, im 3D-Druckermarkt weit verbreitet, Step/Direction Modus oder einem SPI-Modus gewählt werden. Im Step/Direction Modus können über nach außen geführte Pins verschiedenen Schrittauflösungen und Betriebsarten ausgewählt werden.



**Abbildung 3.7** Vereinfachter Schaltplan des Schrittmotortreibers. Der tatsächlich verwendete Schaltplan enthält zusätzliche Lötjumper zur Konfigurationen sowie Testpunkte und ist im Anhang unter Abbildung B.1 einsehbar. Diese Schaltung ist eine Modifikation der *SilentStepStick TMC2130* Schaltung von *Watterott electronic GmbH* [11].

### 3.4.2 Kontrolleinheit

Als steuerndes Element wurde, wie in Kapitel 3.3 beschrieben, ein FPGA verwendet. Da für die Entwicklung der Funktionsmuster das *TinyFPGA A2* [70] Modul auf Grund der geringen Baugröße und einfachen Beschaltung des FPGAs ausgewählt wurde, konnte der dort verbaute FPGA ebenfalls für die Platine verwendet werden. Bei dem FPGA handelt es sich um einen *XO2-1200* [6] von *Lattice Semiconductor* in einem QFN-32<sup>3</sup> Gehäuse. Dieser FPGA hat den Vorteil, dass er in einer sehr kompakten Bauform erhältlich ist ( $5 \times 5 \text{ mm}$ ), welche sich noch von Hand gut löten lässt. Weiterhin benötigt dieser FPGA nur sehr wenige externe Komponenten. Bei diesen handelt es sich Pufferkondensatoren für die Versorgungsspannung und einen Widerstand für das JTAG Programmierinterface. Es wird keine externe Taktquelle benötigt, der FPGA enthält einen Oszillator, der bis auf  $133 \text{ MHz}$  eingestellt werden kann. Weiterhin enthält der FPGA einen internen Flash Speicher zur Speicherung der Konfigurationsdatei. Dadurch entfällt ein externer Speicher.

Da der FPGA mit  $3.3 \text{ V}$  betrieben wird, muss das Schrittsignal, welches von einem Mikrocontroller mit  $5 \text{ V}$  Betriebsspannung generiert wird, durch einen Spannungsteiler auf  $3.3 \text{ V}$  geteilt werden.

Als ADC wurde der während der Funktionsmusterentwicklung verwendete *MAX1165* [33] von *Maxim Integrated* in die Platine implementiert. Dieser bietet eine maximale Abtastrate von  $500 \text{ kSPS}$  bei einer Auflösung von 12 Bit und wird über SPI ausgelesen. Später wurde dieser durch den pin- und protokollkompatiblen *MAX11116* [32] ersetzt, welcher eine Abtastrate von  $3 \text{ MSPS}$  bei einer Auflösung von 8 Bit hat (siehe Kapitel 4.3).

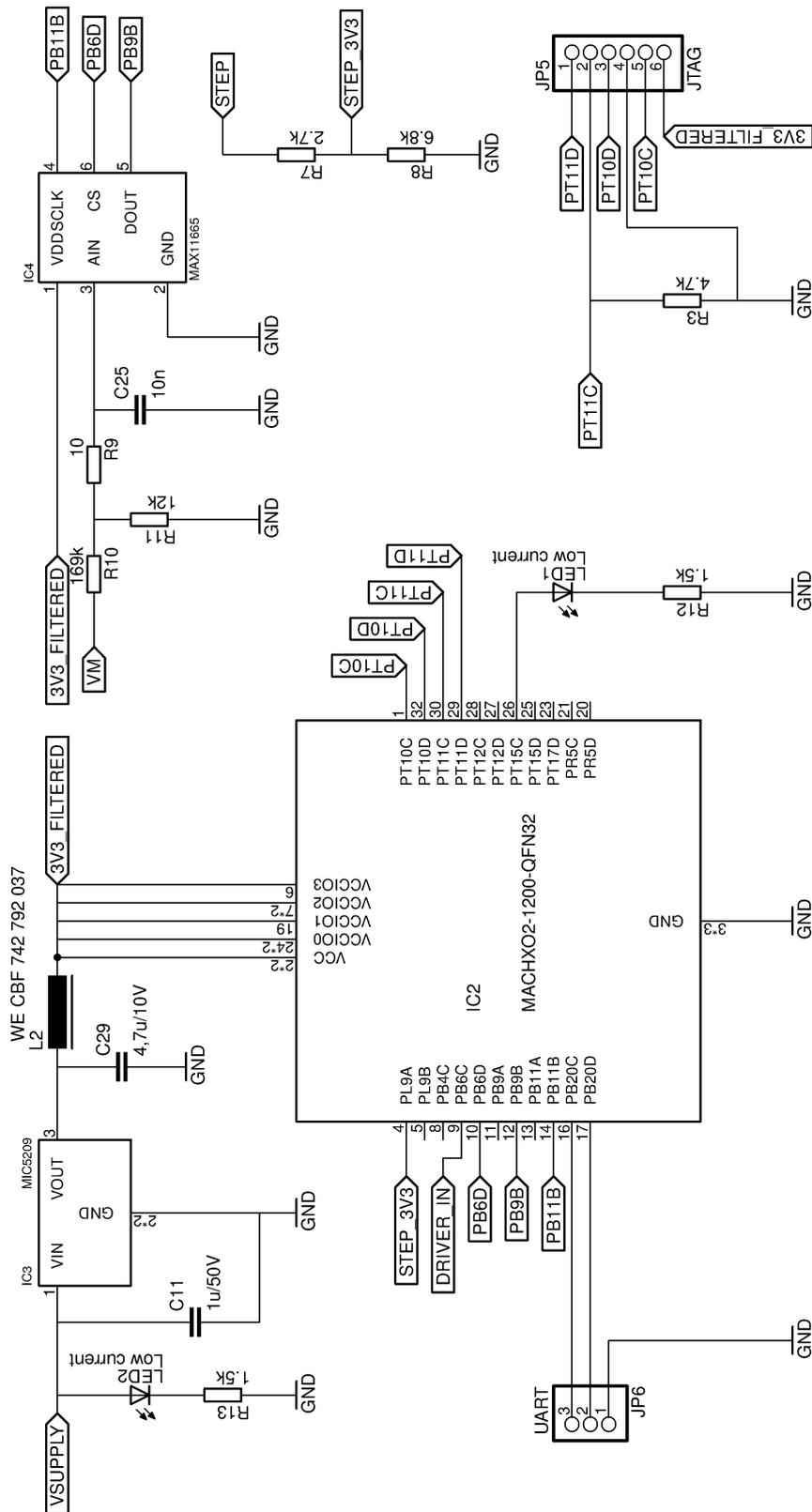
Die Ausgangsspannung wird für die Messung mit dem ADC mithilfe eines Spannungsteilers von  $0 \text{ V}$  bis  $50 \text{ V}$  auf  $0 \text{ V}$  bis  $3.3 \text{ V}$  geteilt.

Zusätzlich ist dem Eingang des ADC ein RC-Tiefpass mit einer Grenzfrequenz von  $1.59 \text{ MHz}$  vorgeschaltet, der das Rauschen und hochfrequente Störungen des Signals herausfiltert.

Als Versorgung für den FPGA und den ADC und damit auch als Referenzspannung für den ADC, wird ein linearer Spannungsregler für  $3.3 \text{ V}$  verwendet. Ein vereinfachter Schaltplan des FPGA- und des ADC-Teils ist in Abbildung 3.8 einsehbar, der tatsächlich verwendete Schaltplan enthält zusätzliche Stützkondensatoren sowie Testpunkte und ist im Anhang unter Abbildung B.2 einsehbar.

---

<sup>3</sup>QFN = Quad Flat No Leads Package ist eine Gehäusebauform für SMD-ICs



**Abbildung 3.8** Vereinfachter Schaltplan des FPGA- und des ADC-Teils. Der tatsächlich verwendete Schaltplan enthält zusätzliche Stützkondensatoren sowie Testpunkte und ist im Anhang unter Abbildung B.2 einsehbar.

### 3.4.3 Leistungsteil des Aufwärtswandlers

Der Aufbau des Leistungsteils entspricht grundsätzlich einem klassischen Aufwärtswandler (siehe Kapitel 2.2). Als Überspannungsschutz wurde eine Suppressordiode vom Typ *SMBJ33A* [25] von *Littelfuse, Inc.* verwendet, die ab einer Spannung von  $36.7\text{ V}$  in den leitenden Bereich übergeht.

Als Schalttransistor wurde ein N-Kanal MOSFET vom Typ *STD40NF10* [60] von *STMicroelectronics* verwendet. Er verträgt als maximale  $V_{DS}$   $100\text{ V}$ , als  $I_D$  kontinuierliche  $50\text{ A}$  bei  $25^\circ\text{ C}$  und bis zu  $300\text{ A}$  als Pulse. Die benötigte Gate-Ladung ist mit maximal  $62\text{ nC}$  angegeben beziehungsweise einer Eingangskapazität von  $2180\text{ pF}$ , der maximale Drain-Source-Widerstand mit höchstens  $28\text{ m}\Omega$ . Diese beiden Faktoren sind für ein möglichst schnelles und verlustfreies Schalten des MOSFETs relevant.

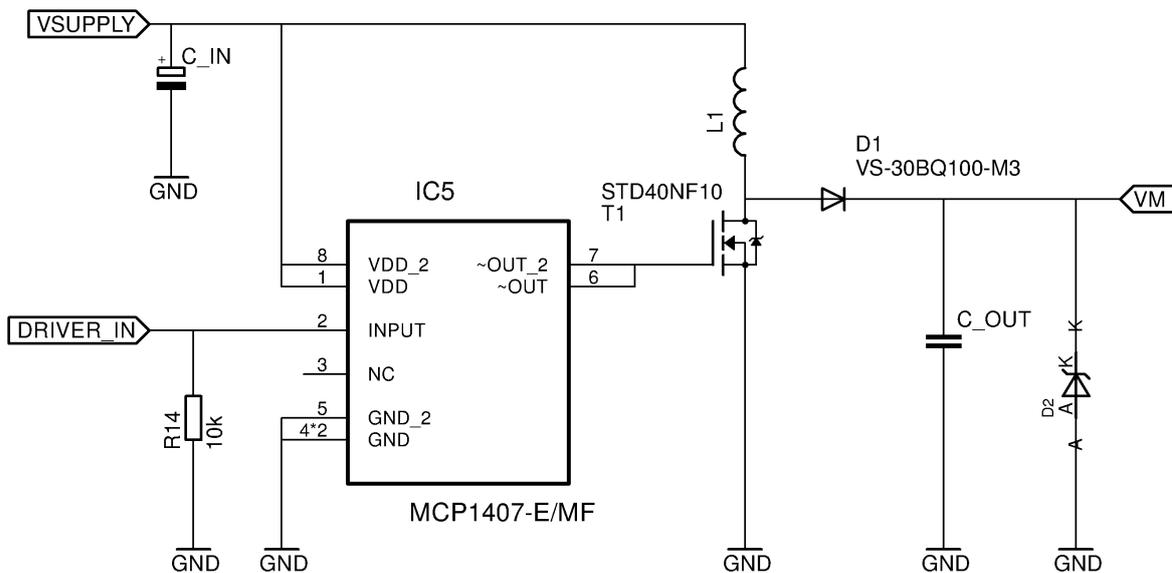
Da der *STD40NF10* erst ab einer  $V_{GS}$  von  $6\text{ V}$  größer als  $30\text{ A}$  schalten kann, und um ein möglichst schnelles Umladen der Gate-Kapazität zu gewähren, wurde zwischen dem FPGA und dem MOSFET ein MOSFET-Treiber geschaltet. Es handelt sich dabei um den *MCP1407* [28] von *Microchip Technology*, welcher von der  $3.3\text{ V}$  Logikspannung des FPGAs geschaltet werden kann. Somit kann das Gate mit den zur Verfügung stehenden  $12\text{ V}$  geschaltet werden. Der Treiber ermöglicht es, bei einer Versorgung mit  $12\text{ V}$  und einem Eingangsstrom von  $45\text{ mA}$ , die Eingangskapazität des MOSFETs von  $2180\text{ pF}$  mit bis zu  $1\text{ MHz}$  zu schalten. Da die letztendlich verwendete Regel- und Schaltfrequenz  $831.250\text{ kHz}$  beträgt, ist das ausreichend. Der Eingang des Treibers ist weiterhin mit einem Pull-Down Widerstand an Masse angeschlossen, um ein ungewolltes Einschalten, zum Beispiel während der FPGA noch konfiguriert wird, zu verhindern.

Als Gleichrichterdiode wurde eine *VS-30BQ100-M3* [29] Schottky-Diode von *Vishay Intertechnology, Inc.* mit einem maximalen Durchgangsstrom von  $3\text{ A}$  verwendet.

Als Speicherdrossel wurde die *WE-PD* [49] Baureihe von *Würth Elektronik* verwendet. Dabei wurden die Löt pads für die Induktivität so dimensioniert, dass verschiedene Varianten dieser Baureihe aufgelötet werden können, um sie auf ihre Belastbarkeit und Verhalten testen zu können.

Als Eingangskondensatoren (C27, C28, C15) wurden ein  $470\text{ }\mu\text{F}$  Elektrolytkondensator parallel zu einem  $4.7\text{ }\mu\text{F}$  und einem  $100\text{ nF}$  Keramikkondensator geschaltet, um das versorgende Netzteil zu entlasten und Spannungseinbrüche zu minimieren.

Für die Ausgangskondensatoren wurden mehrere verschiedenen Bauformen in paralleler Verschaltung vorgesehen, um verschiedene Bauarten sowie Kombinationen und Kapazitäten erproben zu können. Vorgesehen sind zwei Keramikkondensatoren im *1210*-Gehäuse, ein Keramikkondensator im *1206*-Gehäuse sowie ein Anschluss für einen THT-Elektrolytkondensator mit  $10\text{ mm}$  Pinabstand. Ein vereinfachter Schaltplan des Leistungsteils ist in Abbildung 3.9 dargestellt, der tatsächlich verwendete Schaltplan enthält zusätzliche Eingangs- und Ausgangskondensatoren sowie Stützkondensatoren für den Treiber-IC und Testpunkte und ist im Anhang unter Abbildung B.3 einsehbar.



**Abbildung 3.9** Vereinfachter Schaltplan des Leistungsteils. Der tatsächlich verwendete Schaltplan enthält zusätzliche Eingangs- und Ausgangskondensatoren sowie Stützkondensatoren für den Treiber-IC und Testpunkte und ist im Anhang unter Abbildung B.3 einsehbar.

### 3.4.4 Räumliche Aufteilung und Layout der Platine

Die Platine besteht funktionsbedingt aus drei Sektionen, dem Schrittmotortreiber, dem Kontrollteil und dem Leistungsteil. Diese wurden auch in dem Layout räumlich getrennt (siehe Abbildung 3.11), um lange Leitungen, die viel Leistung oder empfindliche Analogsignale übertragen, zu vermeiden. Durch den geringen verfügbaren Platz und der Menge an zu verdrahtenden Signalen reicht eine Platine mit zwei Lagen nicht aus, stattdessen wurde eine Platine mit vier Lagen verwendet (Die vier einzelnen Lagen sind im Anhang in Abbildung B.4 einzusehen). Da die vier Anschlüsse für den Schrittmotor über die Steckverbinder zurück auf die Steuerplatine geleitet werden, wurde der *TMC2130* möglichst nahe an den Steckverbindern platziert, um lange Motorleitungen zu vermeiden.

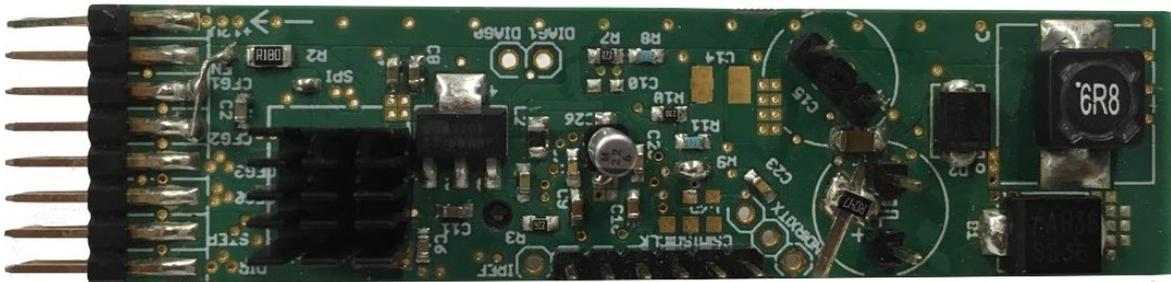
Dazu kommt, dass der Schrittmotortreiber viel Abwärme produziert, welche nur zum Teil über den Kühlkörper abtransportiert wird. Ein Teil der Wärme wird auch über die Platine abgeleitet.

Der andere Funktionsblock, der viel Abwärme erzeugt, ist der Leistungsteil, insbesondere die Induktivität und der Leistungstransistor. Um eine bessere Wärmeverteilung auf der gesamten Platine zu erreichen, wurde dieser Teil möglichst weit entfernt von dem *TMC2130* positioniert, am oberen Ende der Platine.

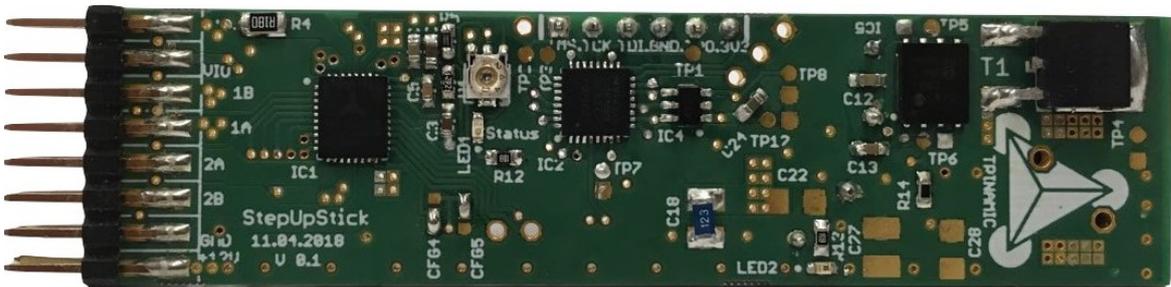
Auch wurde darauf geachtet, die versorgenden und abführenden Leitungen des Leistungsteils ausreichend zu dimensionieren, um einerseits Verluste durch den Kupfer-

widerstand zu reduzieren, und um die Wärmezeugung zu vermindern. Die zuführende Leitung für die 12 V Versorgungsspannung ist 2.7 mm breit und auf zwei Lagen vorhanden, womit sich, bei einer Kupferdicke von 35  $\mu\text{m}$ , ein maximaler Gesamtstrom von insgesamt 8.5 A bei einer Erwärmung um 10 °C ergibt [17].

Die Leitung zwischen dem Aufwärtswandler und dem Schrittmotortreiber wurde mit 6.35 mm Breite ausgelegt, was für einen maximalen Strom von 8 A bei einer Erwärmung um 10 °C reicht [17]. Dabei sind die Schwachstellen die beiden Verjüngungen auf die Pinbreite an dem IC selbst. Bei einem Kurzschluss durch den Schrittmotortreiber sind an dieser Stelle die Leiterbahnen durchgebrannt.

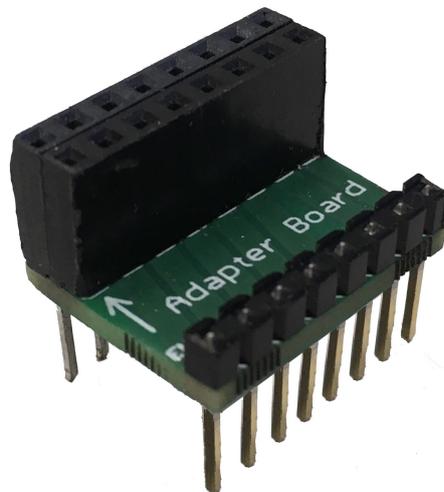


(a) Oberseite der entwickelten Platine



### 3.4.5 Adapter für das *StepStick*-Modul Pinout

Um die vorgegebene Pinompatibilität zu *StepStick*-Modulen zu erlauben, muss die längliche Platine mithilfe eines Adapters senkrecht stehend verbaut werden (siehe Abbildung 3.14). Der Adapter ist rein passiv (siehe dazu Abbildung B.5 und B.6 im Anhang) und leitet die Signale, die von dem Modul auf einer zweireihigen Steckerleiste ausgegeben werden, auf die vorgegebenen Pins, entsprechend eines *StepSticks*. Der aufgebaute Adapter ist in Abbildung 3.12 dargestellt und hat die gleiche Breite und Tiefe wie ein *StepStick*-Modul.



**Abbildung 3.12** Benötigter Adapter, um das entwickelte Modul in gängigen 3D-Drucker-Boards, wie zum Beispiel dem RAMPS-Board, betreiben zu können

## 3.5 Inbetriebnahme

Die Inbetriebnahme der Platine erfolgte in mehreren Ausbaustufen, um die Fehlfunktion einzelner Funktionsblöcke auszuschließen.

Eine Platine wurde nur mit den notwendigen Komponenten für die Funktion des FPGAs bestückt, um zu überprüfen, ob er sich programmieren lässt, die Spannungsversorgung funktioniert und alle Pins korrekt angesteuert werden. Zusätzlich wurde das Auslesen des ADC getestet. Diese teilbestückten Testaufbauten funktionierten dabei alle so wie geplant.

Eine Platine wurde nur mit den Komponenten des Schrittmotortreibers bestückt, es handelt sich dabei quasi um einen *SilentStepStick TMC2130* mit einem anderen Layout. Bei der Inbetriebnahme dieser Bestückung stellte sich eine Besonderheit der

Tri-State Pins des *TMC2130* heraus. Die beiden Pins zum Konfigurieren der Schrittart (Pin *CFG1* und *CFG2*) können jeweils an Masse oder an die Logikversorgungsspannung angeschlossen werden oder für einen dritten Zustand offen gelassen werden. Für die Betriebsart, die für die meisten 3D-Drucker genutzt wird, nämlich die Interpolation von  $1/16$  Mikroschritte auf  $1/256$  Mikroschritte mit aktivem *stealthChop* Modus, müssen beide Pins offen gelassen werden. Dies führte allerdings dazu, dass die Interpolation zufällig aktiviert und deaktiviert wurde. Die Überprüfung auf einen offenen Pin erfolgt innerhalb des *TMC2130*, indem der Pin mit einem Rechtecksignal angesteuert wird. Kann der Pin diesem Signal folgen, ist er nicht an ein definiertes angeschlossen, also offen. Liegt er jedoch auf Masse oder der Logikversorgungsspannung, ist er nicht offen. Die Lösung war es, die entsprechenden Pins mit einer Parallelschaltung aus einem  $2\text{ M}\Omega$  Widerstand mit einem  $10\text{ pF}$  Kondensator gegen Masse abzuschließen.

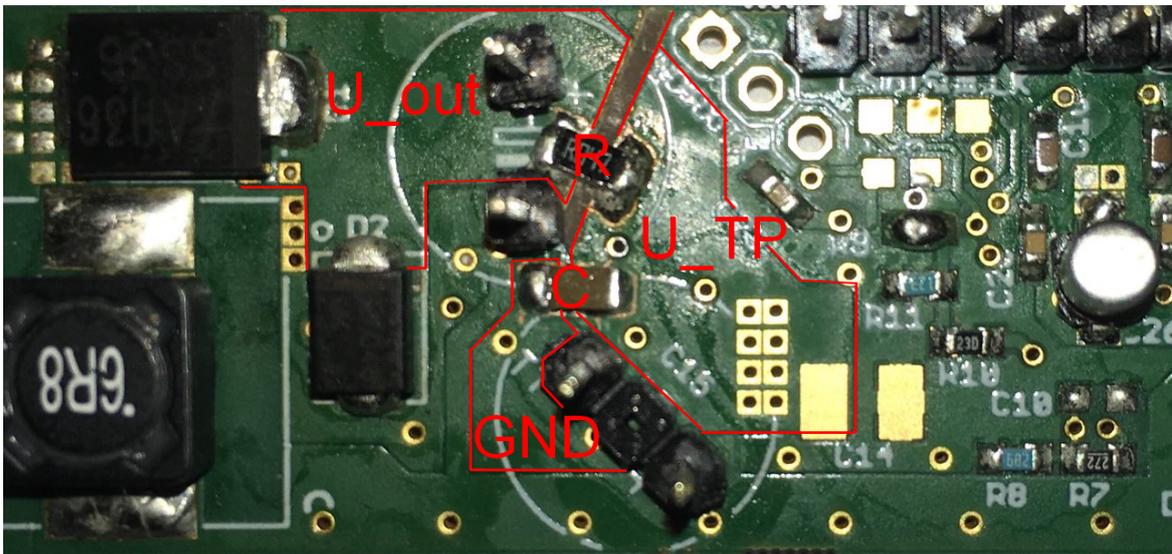
Für den Wärmeabtransport des Schrittmotortreibers musste, wie bei dem *SilentStep-Stick TMC2130* auch, ein Kühlkörper angebracht werden.

Nachdem sich der FPGA- und ADC-Teil der Platine als funktionsfähig herausgestellt hatten, wurde diese Platine mit den Komponenten für den Aufwärtswandler bestückt, wobei die erzeugte Ausgangsspannung dabei über Kabel extern abgeleitet wurde. Damit entspricht dieser Bestückungszustand ungefähr dem dritten Funktionsmuster (siehe Anhang A). Als dieser Bestückungszustand das erste Mal ohne extern angeschlossene Last getestet wurde, stellte sich heraus, dass versäumt wurde, eine Grundlast in die Schaltung zu integrieren. Ohne diese kann die erzeugte Spannung nur über die parasitären Innenwiderstände der Ausgangskondensatoren abfallen, wodurch sich innerhalb weniger Schaltzyklen des Transistors eine so große Spannung aufgebaut hat, dass die Suppressordiode in den leitenden Zustand übergeht und den Spannungswandler damit kurzschließt. Um das zu verhindern, wurde auf den Ausgangskondensator *C18* im 1206-Gehäuse ein  $12\text{ k}\Omega$  Widerstand im 0805-Gehäuse aufgelötet. Dieser erzeugt bei der Maximalspannung von  $36\text{ V}$  eine Grundlast von  $3\text{ mA}$ , was ausreicht, um eine stabile Regelung ohne Überspannung zu ermöglichen.

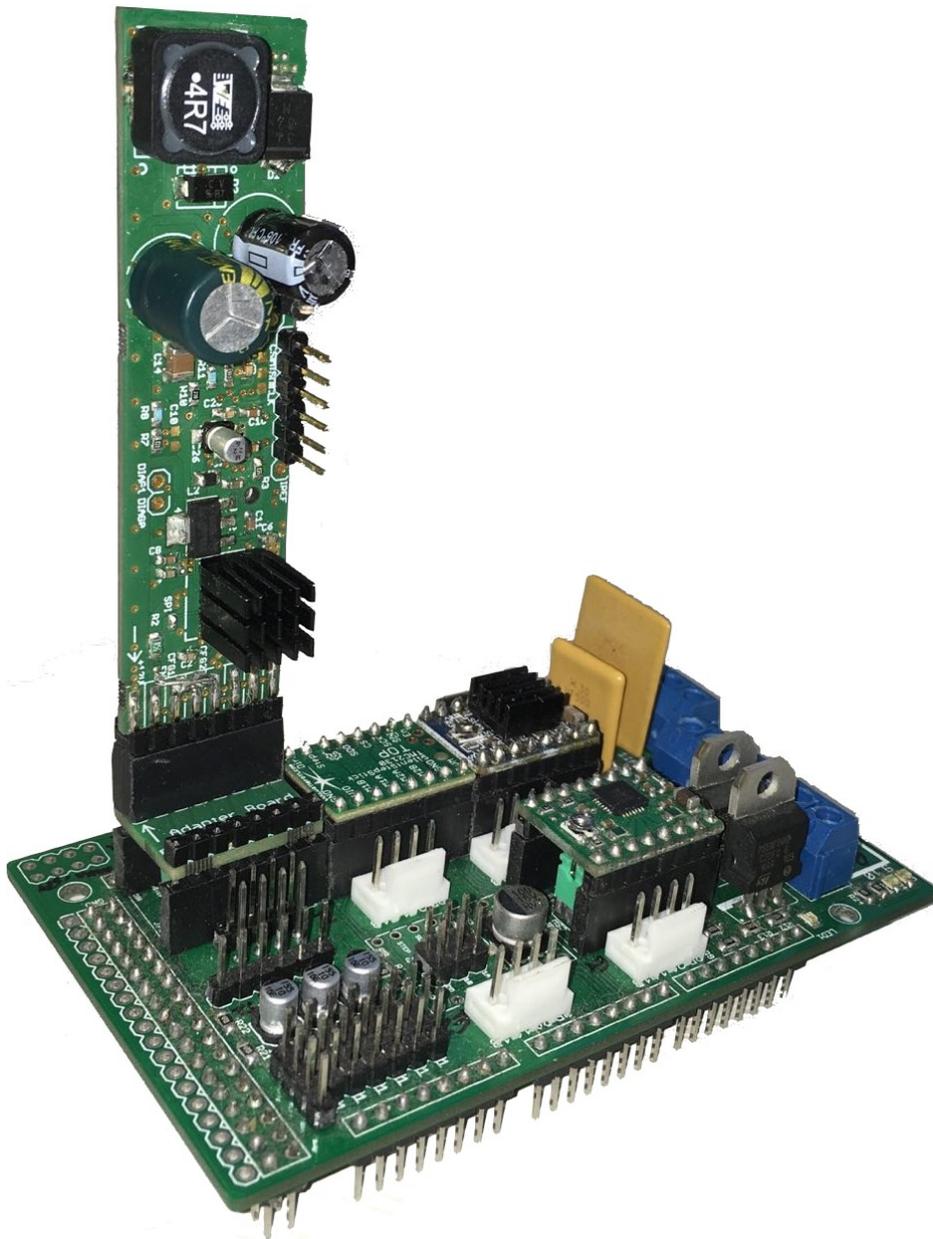
Danach wurde eine Platine komplett bestückt und die beschriebenen Modifikationen mit eingebaut. Danach funktionierte die Platine grundsätzlich, jedoch war trotz Anpassung der Regelung sowie Implementierung des Sanftanlaufs immer noch eine zu große Rippelspannung auf der erzeugten Ausgangsspannung. Als Lösung wurde schließlich ein RC-Tiefpass mit einer Grenzfrequenz von  $498.234\text{ kHz}$  ( $R = 0.47\ \Omega$ ,  $C = 680\text{ nF}$ ) zwischen dem Ausgang des Auswärtswandlers und dem *TMC2130* gesetzt. Da das in dem Platinendesign nicht vorgesehen war, musste die Leiterbahn zwischen Treiber und Auswärtswandler an einer Stelle durchtrennt sowie an mehreren Stellen der Lötstopplack entfernt werden, um dort den Widerstand und den Kondensator einlöten zu können. Diese Modifikation ist in Abbildung 3.13 zu erkennen. Danach blieb die Rippelspannung in einem akzeptablen Rahmen (siehe Kapitel

5.1.2) und die Platine verhielt sich wie geplant. Auf solchen, so modifizierten Platinen, wurde der größte Teil der Firmware und der Regelung entwickelt, was in dem folgenden Kapitel 4 beschrieben ist.

Das so funktionierende Modul wurde mithilfe des Adapters in ein RAMPS-3D-Drucker-Board eingesetzt (siehe Abbildung 3.14), um zu überprüfen, ob die geplante Pin- und Funktionskompatibilität zu einem *StepStick*-Modul erfüllt wird. Bei Ansteuerung des RAMPS-Board bei Verwendung der unveränderten 3D-Drucker-Firmware [1], konnte das entwickelte Modul die Schrittmotoren genau so ansteuern, wie ein zum Vergleich verwendeter *SilentStepStick TMC2130*.



**Abbildung 3.13** Nachträglich hinzugefügter RC-Tiefpass. Gut zu erkennen ist, wo die Leiterbahn, die die erzeugte Spannung an den *TMC2130* leitet, durchtrennt wurde, um nachträglich den Widerstand einzulöten.

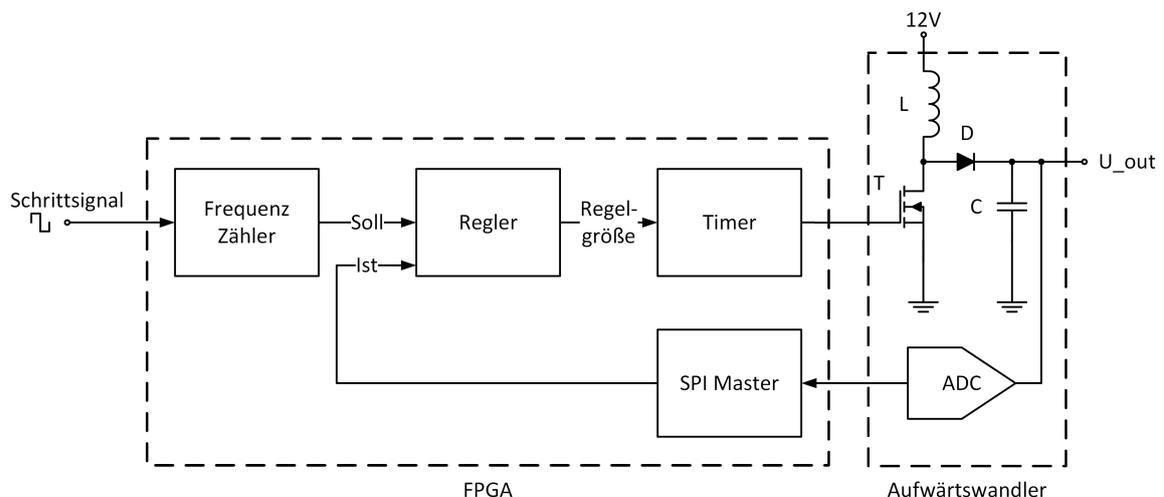


**Abbildung 3.14** Diese Abbildung zeigt, wie das entwickelte Modul mithilfe des Adapters in einem RAMPS 3D-Drucker Board eingesteckt wird [63]. Die drei weiteren Module sind verschiedenen Varianten der *StepStick*-Schrittmotortreiber [64]. In diesem Bild sind auch die räumlichen Begrenzungen zu erkennen, die zu dem schmalen sowie länglichem Design der Platine und der senkrechten Montage mittels Adapter geführt haben.

## 4 Entwicklung der Firmware

Die Entwicklung der Firmware für den FPGA erfolgte zu großen Teilen auf der fertig entworfenen und aufgebauten Platine. Grund hierfür waren die in Kapitel 3.3 beschriebenen Probleme der Funktionsmuster bezüglich der elektromagnetischen Kopplungsmechanismen. Weiterhin ist die Wärmeabgabe des *TMC2130* Schrittmotortreibers sowie des Aufwärtswandlers nicht zu unterschätzen, weshalb das gemeinsame Verhalten auf einer Platine während der Entwicklungs- und Testphase ein zu beachtender Aspekt war. Dadurch waren allerdings während der Entwicklung der Firmware keine Änderungen an der Hardware mehr möglich, wodurch sich Limitationen für das Messen von Parametern ergaben (siehe Kapitel 4.4). Aufgabe der Firmware ist es zum einen, den Aufwärtswandler auf einen gewünschten Sollwert zu regeln und zum anderen, aus den Steuersignalen für den Schrittmotortreiber einen Spannungssollwert zu generieren. Wie in Kapitel 3.1 dargelegt, wird als auszuwertender Parameter das Schrittsignal gewählt, das dem Schrittmotortreiber zugeführt wird.

Das vereinfachte Konzept der Firmware ist in Abbildung 4.1 dargestellt. Es sind die beiden Funktionsblöcke erkennbar, welche die Generierung des Sollwertes (*Frequenzzähler*) und die Regelung des Aufwärtswandlers (*Regler*) enthalten. Weiterhin ist vereinfacht die Anbindung des FPGAs an den Aufwärtswandler dargestellt. An den Funktionsblöcken stehen die Namen der Prozesse aus dem VHDL-Code. Der entwickelte VHDL-Code ist im Anhang unter D einsehbar.



**Abbildung 4.1** Vereinfachte Darstellung des entwickelten Aufwärtswandlers. Die Funktionsblöcke stellen die schematischen Funktion der Firmware des FPGAs dar.

Der Aufbau und die Strukturierung des entwickelten VHDL-Codes ist in Abbildung 4.2 dargestellt. Der Übersicht halber wurde der Code für den SPI-Master, die Teileinheit und den Schrittzähler als Komponenten ausgegliedert.

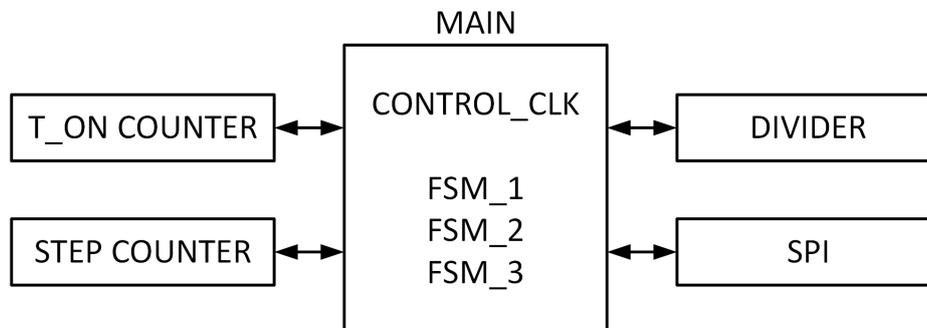


Abbildung 4.2 Aufbau der Firmware

Die grundsätzliche Funktionsweise der Firmware ist die periodische Messung der erzeugten Ausgangsspannung des Aufwärtswandlers sowie die Messung der aktuellen Schrittfrequenz, also der aktuellen Soll Drehzahl des Schrittmotors. Mit der gleichen Periodizität vergleicht ein Regler die beiden Messwerte (Sollwert und Istwert) um die Schaltzeiten des Transistors des Aufwärtswandlers entsprechend anzupassen. Dadurch teilt sich der entwickelte Code in mehrere Funktionsblöcke auf (siehe Abbildung 4.1). Im Folgenden soll die Funktion der einzelnen Blöcke erklärt werden. Wie die einzelnen Komponenten zusammen funktionieren wird danach beschrieben.

## 4.1 Regelung des Aufwärtswandlers

Die Regelung des Aufwärtswandlers geschieht in zeitlich konstanten Abständen, es handelt sich also um eine zeitdiskrete und nicht um eine zeitkontinuierliche Regelung. Die Abtastrate wird mithilfe eines dekrementierenden Auto-Reload-Timers eingestellt, also ein Timer der bei Erreichen seines Zielwertes sich automatisch neu stellt und damit ein einstellbares periodisches Signal erzeugt, dessen Zustand von mehreren Prozessen als Auslöser verwendet wird. Diese zeitlich gekoppelte Ausführung der Prozesse ist besonders wichtig für die entwickelte Parallelisierung der Prozesse zum Erreichen einer möglichst hohen Abtastfrequenz. Auf die Parallelisierung wird in Kapitel 4.3 eingegangen. Die Regelung setzt sich aus der Erfassung der aktuellen, erzeugten Ausgangsspannung mithilfe eines ADC und dem Regelalgorithmus, der diesen gemessenen Istwert mit einem Sollwert vergleicht, zusammen. Auf die Auswertung des ADC sowie auf die Regelung soll in den nächsten beiden Kapiteln eingegangen werden. Auf die Erzeugung des Sollwertes wird in Kapitel 4.2.1 eingegangen.

### 4.1.1 Auswertung des ADC

Der verwendete ADC wird über eine SPI-Schnittstelle angesteuert. Als SPI-Master wurde ein fertiger Soft-IP-Core verwendet [58]. Da die Übertragung der Daten vom ADC in den FPGA eine nicht vernachlässigbare Zeit dauert, wurde für die Steuerung des SPI-Masters ein Zustandsautomat entwickelt. Das Ablaufdiagramm dieses Zustandsautomaten ist in Abbildung 4.3 zu sehen. Hauptaufgabe des Zustandsautomaten ist es, die SPI-Kommunikation zum Abtastzeitpunkt, ausgelöst durch einen globalen Timer (`CONTROL_CLK_COUNTER`), zu starten und die empfangenen Daten kurz vor dem nächsten Abtastvorgang wegzuschreiben. Das Wegschreiben geschieht dabei deswegen so kurz vor dem nächsten Abtasten, weil während der laufenden SPI-Kommunikation die Daten aus dem vorherigen Lesezyklus verarbeitet werden (siehe Kapitel 4.1.2). Auf das Signal (`ADC_DATA`) darf deshalb nicht während der Verarbeitung geschrieben werden. Der Code für den SPI-Master ist im Anhang unter D.3 einsehbar, der Code für den Zustandsautomaten, der die SPI-Kommunikation auswertet, ist im Anhang unter D.1 einsehbar.

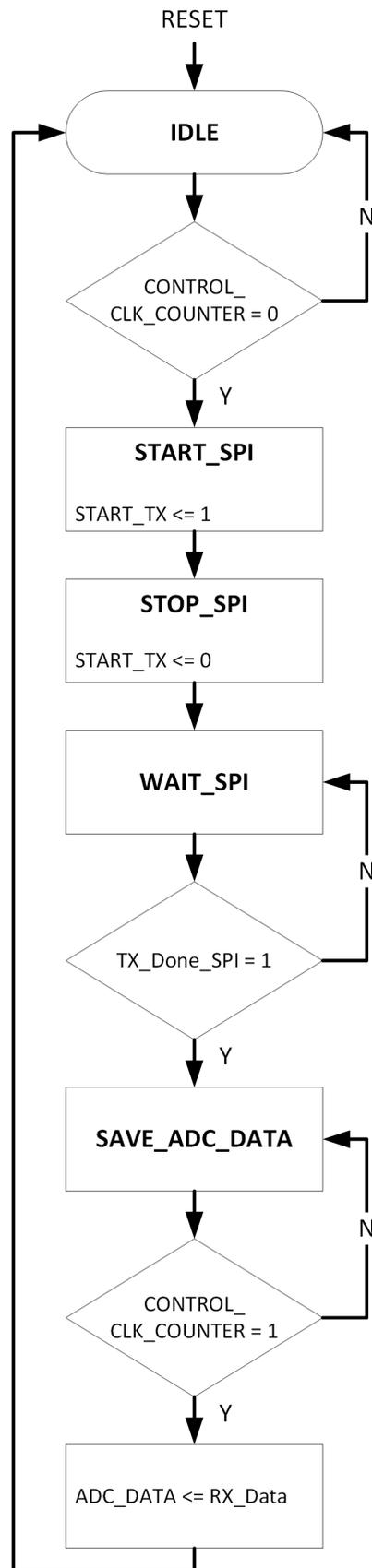


Abbildung 4.3 Zustandsautomat FSM\_1 zur Steuerung des SPI-Masters

### 4.1.2 Hysterese-Spannungsregelung

Die verwendete Regelungsart ist die in Kapitel 2.2.4 beschriebene Hysterese-Regelung. Diese wurde aufgrund ihrer simplen Implementierbarkeit gewählt sowie wegen ihrer sehr schnellen Reaktion auf veränderte Sollwerte der Regelung [55]. Da in der beabsichtigten Anwendung sich die benötigte Sollspannung durch die Be- und Entschleunigung des Schrittmotors stetig ändert, ist diese Eigenschaft ein sehr wichtiges Kriterium der Regelung. Der Ablauf der Regelung ist in Abbildung 4.1.2 dargestellt und soll im Folgenden erklärt werden. Der verwendete Code dafür ist im Anhang unter D.1 einsehbar.

Gestartet wird der Regelzyklus durch einen globalen Timer (`CONTROL_CLK_COUNTER`), woraufhin der gemessene ADC Wert (`ADC_DATA`) mit dem zu erzeugenden Sollwert (`ADC_SOLLWERT`) verglichen wird. Ist der Messwert beziehungsweise Istwert kleiner als der Sollwert, muss der Aufwärtswandler aktiviert werden, um die Spannung zu generieren, anderenfalls wird der MOSFET nicht eingeschaltet. Dafür wird der Pin (`PWM_OUT_PIN`) auf *HIGH* gesetzt, dadurch wird der MOSFET eingeschaltet und die Induktivität geladen.

Gleichzeitig wird der Timer (`T_ON_COUNTER_PROCESS`) geladen und gestartet. Der Ladewert des Timers bestimmt die Einschaltzeit des Transistors. Der Ladewert hängt dabei von der gemessenen Spannung (`ADC_DATA`) ab, der entsprechende Ladewert wird dabei aus einer Lookup-Tabelle abgelesen.

Bei der verwendeten Hysterese-Regelung wird eigentlich eine feste Einschaltdauer verwendet (siehe Kapitel 2.2.4), weswegen diese Regelungsart auch *Constant On-Time Control* genannt wird. Diese Einschaltdauer wurde zu Beginn der Entwicklung im Verhältnis zu der maximal möglichen Einschaltdauer relativ hoch gewählt, entsprechend einem Tastverhältnis von 0.8. Das hatte den Grund, dass bei anfänglich getesteten Lastsprüngen von  $0 \text{ min}^{-1}$  auf  $2000 \text{ min}^{-1}$  beziehungsweise 0 V bei geringem Strombedarf auf 36 V bei hohem Strombedarf, ein möglichst schnelles Ansteigen der Ausgangsspannung gesucht war. Dies ließ sich durch das hohe Tastverhältnis erreichen.

Als jedoch die Eingangsströme während dieser Lastsprünge gemessen wurden, stellten diese sich als unakzeptabel hoch heraus (siehe Kapitel 5.1.4). Daraufhin wurden niedrigere Tastverhältnisse erprobt, welche aber wiederum die maximale Ausgangsspannung von 36 V unter Last nicht erreichen konnten. Insbesondere bei kleineren Spannungsdifferenzen zwischen Eingangsspannung und Ausgangsspannung des Aufwärtswandlers, zum Beispiel 6 V Differenz, und gleichzeitig niedrigem Strombedarf der Last und größeren Spannungsdifferenzen zwischen Eingangsspannung und Ausgangsspannung des Aufwärtswandlers, zum Beispiel 20 V Differenz, und gleichzeitig großem Strombedarf der Last, ließ sich kein Tastverhältnis finden, welches bei beiden Anforderungen zufriedenstellende Ergebnisse lieferte. Gerade aber dieser Anwendungsfall, der Wechsel von niedriger Ausgangsspannung mit niedri-

gem Strombedarf zu hoher Ausgangsspannung bei hohem Strombedarf, kommt bei der Be- und Entschleunigung des Schrittmotors ständig vor.

Als Lösung wurde ein Sanftanlauf entwickelt, der je nach Anforderung eine andere Einschaltzeit erzeugt. Die genaue Funktion des Sanftanlaufs ist in Kapitel 4.4 beschrieben.

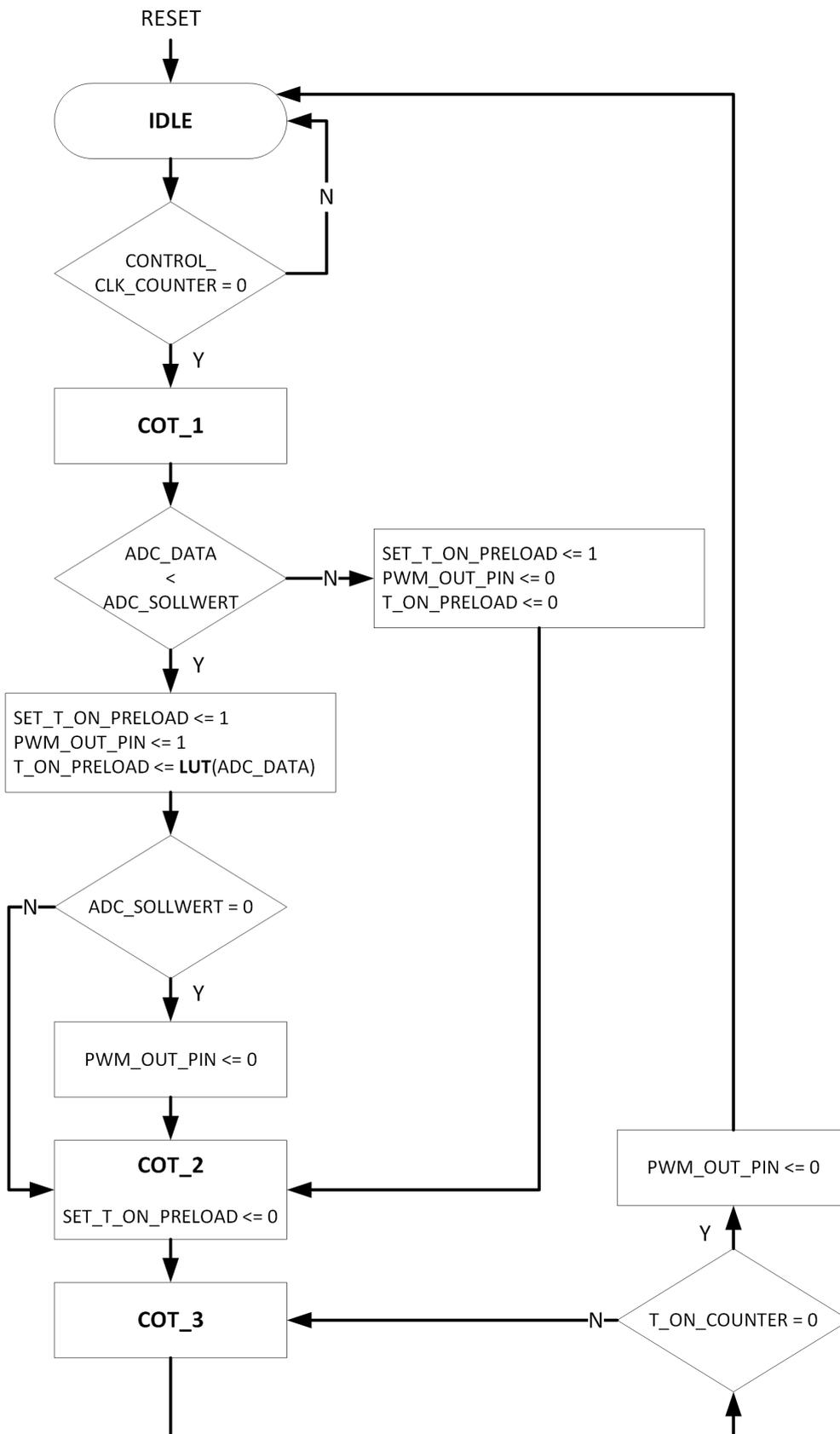


Abbildung 4.4 Zustandsautomat FSM\_3 für die Regelung und Festlegung der Einschaltdauer des Transistors

## 4.2 Auswertung des Schrittsignals

Bei dem am Schrittmotortreiber ankommenden Schrittsignal handelt es sich um ein Rechtecksignal, dessen Pulsweite im Verhältnis zu der Periodendauer in der Regel sehr kurz ist [10]. Bei jeder vom Schrittmotortreiber erkannten positiven Flanke, dreht dieser den Schrittmotor um einen Schritt weiter. Der Schrittwinkel hängt dabei zum einen von dem mechanischem Schrittwinkel des Motors ab und von der Einstellung des Schrittmotortreibers. Die Generierung der Schrittsignale muss an die Einstellung des Schrittmotortreibers angepasst sein, damit der Motor sich so verhält, wie der steuernde Computer/Mikrocontroller es vorgibt. Wenn der steuernde Controller zum Beispiel für eine gewünschte Drehzahl Schrittsignale in Vollschrittauflösung (siehe Kapitel 2.1.4) generiert und der Schrittmotortreiber auf  $1/16$  Mikroschritte eingestellt ist, dreht sich der Motor mit einem  $1/16$  der eingestellten Drehzahl. Um dieses Signal auszuwerten, wurde ein Frequenzzähler entworfen (siehe Code D.4), der den Abstand zwischen zwei positiven Flanken in der Periodendauer der Taktfrequenz des FPGAs zählt. Es wird eine maximale Zeit von  $0.0125$  s gezählt, was einer Frequenz von  $80$  Hz entspricht. Bei einer eingestellten Mikroschrittauflösung von  $1/16$  und einem mechanischem Schrittwinkel von  $1.8^\circ$  entspricht das einer Drehzahl von  $1.5 \text{ min}^{-1}$ . Diese Drehzahl ist so gering, das ein Aufwärtswandeln der Spannung keinen Unterschied bezüglich des Drehmoments macht (siehe Abbildung 4.5).

### 4.2.1 Vorbereitung der Rechenparameter

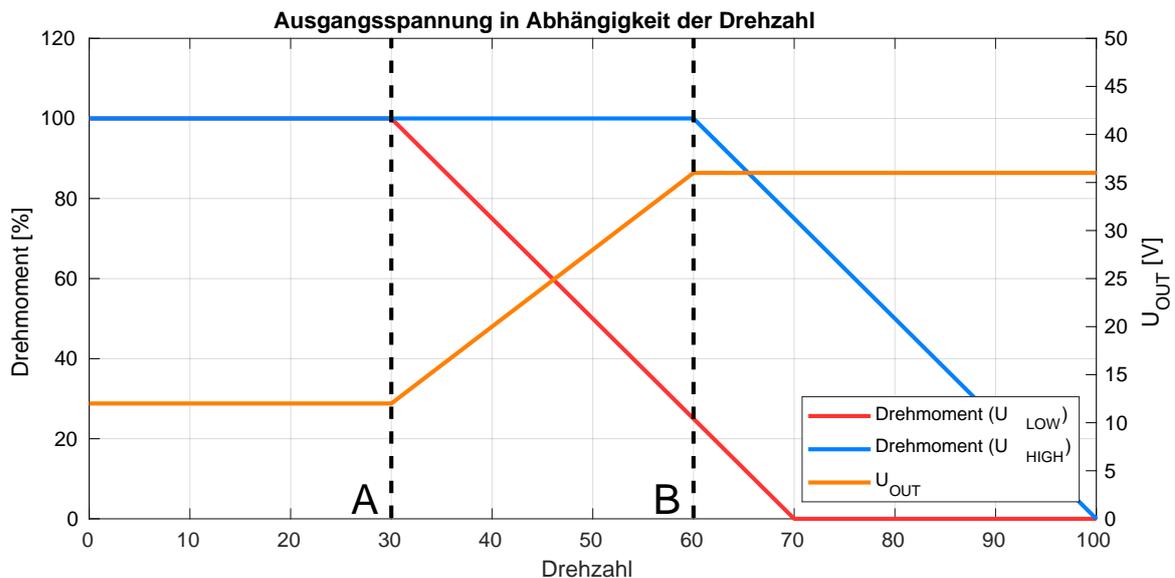
Der Sollwert des Aufwärtswandlers hängt von dem Schrittsignal, also der Drehzahl des Schrittmotors ab. Jedoch musste festgelegt werden, in welcher Art und Weise diese Abhängigkeit besteht. Da die Idee dieser Entwicklung war, das Drehmoment eines Schrittmotors über einen größeren Drehzahlbereich konstant hoch zu halten, wurde das Verhalten der Drehmomentkurve bei verschiedenen Versorgungsspannungen untersucht.

Es zeigt sich, dass, wie in Kapitel 2.1.2 erklärt, das Drehmoment umso länger konstant hoch bleibt, desto höher die Versorgungsspannung des Schrittmotors, beziehungsweise des Schrittmotortreibers ist. In Abbildung 4.5 sind zwei sinnbildliche Kennlinien eines Schrittmotors abgebildet. Eine Kennlinie beschreibt den Drehmomentverlauf bei einer Versorgungsspannung von  $12$  V und einen bei  $36$  V. Wie zu erkennen ist, starten beide Kennlinien bei dem gleichen Drehmoment, die Kurve mit der niedrigeren Spannung bricht jedoch früher ein. Ziel ist es, die gleiche Kennlinie wie die der höheren Spannung zu erreichen, indem die niedrige Versorgungsspannung von  $12$  V aufwärts gewandelt wird.

Da die Kennlinie der niedrigen Spannung bei der Drehzahl **A** einzubrechen beginnt,

wird die Spannung ab dieser Drehzahl aufwärts gewandelt. Darunter macht es, bezogen auf das maximale Drehmoment, keinen Unterschied, ob eine Spannung von 12 V oder größer verwendet wird. Ab der Drehzahl bei Punkt **B** beginnt auch die Kennlinie von 36 V einzubrechen. Ab diesem Punkt ist ein Aufrechterhalten des Drehmoments nur noch durch eine höhere Spannung möglich. Durch die Eingangsspannung und die verwendeten Komponenten des Aufwärtswandlers sowie der maximalen Eingangsspannung des Schrittmotortreibers ist der mögliche Spannungsbereich festgelegt. Bei dem entwickelten Modul geht dieser Bereich von 12 V bis 36 V. Das bedeutet, dass mithilfe des Aufwärtswandlers maximal der Drehmomentverlauf von 36 V erzeugt werden kann. Durch diese Beschränkungen ergibt sich der in Abbildung 4.5 gezeigte Spannungsverlauf. Unterhalb von Drehzahl **A** wird der Aufwärtswandler deaktiviert und die 12 V an den Schrittmotortreiber durchgeleitet. Ab Drehzahl **B** wird eine Spannung von 36 V erzeugt. Dazwischen steigt die Spannung linear mit der Drehzahl an.

Theoretisch sind auch andere Formen des Anstiegs möglich, die sich durch eine Lookup-Tabelle oder zum Beispiel durch ein Polynom erzeugen ließe. Jedoch besitzt der verwendete FPGA nicht genügend Kapazitäten, um rechenaufwendigere mathematischen Funktionen, wie zum Beispiel Potenzierung zu implementieren.



**Abbildung 4.5** Abhängigkeit des zu regelnden Sollwerts von der gemessenen Schrittfrequenz. Um die Abhängigkeit der zu erzeugenden Ausgangsspannung von der Motordrehzahl richtig einzustellen, sind zwei Kennlinien des verwendeten Schrittmotors nötig. Zum einen wird die Kennlinie bei der Eingangsspannung des Aufwärtswandlers benötigt ( $U_{LOW}$ , in diesem Beispiel 12 V), zum anderen die Kennlinie bei der maximal möglichen Versorgungsspannung des Schrittmotortreibers ( $U_{HIGH}$ , in diesem Beispiel 36 V). Beide Drehmomentverläufe brechen ab einer bestimmten Drehzahl ein, dabei bricht die Kurve bei der höheren Spannung später ein. Der Drehzahlbereich zwischen diesen beiden Drehzahlen liegt in der Abbildung zwischen **A** und **B**. Vor der Drehzahl **A** muss die Spannung nicht erhöht werden, da das maximale Drehmoment auch mit der niedrigen Versorgungsspannung erreicht wird. Nach der Drehzahl **B** wäre ein Erhöhen der Motorspannung über den maximal zulässigen Wert des Motortreibers nötig, um das Drehmoment weiter aufrecht zu erhalten. Daher wird die Spannung zwischen Drehzahl **A** und **B** linear gesteigert.

Daher wurde auf eine Geradengleichung zurückgegriffen, die als mathematische Operationen nur eine Addition und eine Multiplikation benötigt. Beide sind in VHDL einfach nutzbar, im Gegensatz zu einer Division oder Potenzierung.

$$y = x \cdot m + n \quad (4.1)$$

In der Gleichung 4.1 stellt dabei  $x$  die gemessene Schrittfrequenz dar und  $y$  den Sollwert. Die Steigung wird dabei über die Parameter  $m$  und  $n$  eingestellt. Im Folgenden wird erläutert, wie die Parameter  $m$  und  $n$  sowie die Schrittfrequenzen berechnet und für eine Verarbeitung im FPGA vorbereitet werden:

1. Es muss ein Drehzahlbereich ausgewählt werden:  
Dieser hängt hauptsächlich von dem verwendeten Motor und teilweise den Einstellungen des Schrittmotortreibers ab. Idealerweise stehen für den verwendeten Schrittmotor Kennlinien für die Eingangsspannung des Aufwärtswandlers und für die maximal erzeugbare Ausgangsspannung in der gewünschten Betriebsart zu Verfügung. Diese könnten ähnlich aussehen wie die Kennlinien in Abbildung 4.5. Der zu wählende Drehzahlbereich liegt zwischen dem Einbrechen des Drehmoments bei der niedrigen Spannung (in Abbildung 4.5 Drehzahl **A**) und dem Einbrechen des Drehmoments bei der maximalen Spannung (in Abbildung 4.5 Drehzahl **B**).
2. Die beiden Drehzahlen in der Einheit Umdrehung pro Minute müssen in die, von dem FPGA gemessene, Schrittfrequenz umgewandelt werden:  
Dafür sind neben der Drehzahl noch zwei weitere Faktoren notwendig, zum einen die mechanische Auflösung des Schrittmotors (siehe Kapitel 2.1.1) und zum anderen die Schritart, mit der der Schrittmotor betrieben werden soll (siehe Kapitel 2.1.4). Daraus lässt sich wie folgt die Schrittfrequenz berechnen:

$$f = \frac{n}{60 \text{ s}} \cdot \frac{360^\circ}{\varphi_0} \cdot \frac{1}{m} \quad (4.2)$$

Dabei steht  $f$  für die Schrittfrequenz in Hz,  $n$  ist die Drehzahl in Umdrehungen pro Minute,  $\varphi_0$  ist der mechanische Schrittwinkel des Motors in Grad und  $m$  ist die Mikroschrittauflösung (1, 1/2, 1/4, 1/16...). Als Beispiel soll im Folgenden für die Drehzahl  $40 \text{ min}^{-1}$  die Schrittfrequenz berechnet werden. Der Schrittmotor habe dabei einen Schrittwinkel von  $1.8^\circ$  und soll mit einer Mikroschrittauflösung von 1/16 betrieben werden:

$$f = \frac{40 \text{ min}^{-1}}{60 \text{ s}} \cdot \frac{360^\circ}{1.8^\circ} \cdot \frac{1}{1/16} = 2133,3\bar{3} \text{ Hz} \quad (4.3)$$

Der FPGA würde also in diesem Fall eine Frequenz von  $2133.33 \text{ Hz}$  messen.

3. Die ADC Werte der minimalen und maximalen Spannungen müssen bekannt sein. Der Spannungsteiler vor dem ADC wurde so ausgelegt, dass er den Spannungsbereich von  $0\text{ V} - 50\text{ V}$  auf  $0\text{ V} - 3.3\text{ V}$  teilt. Je nach Auflösung des verwendeten ADC ergeben sich daraus die gesuchten Werte für die beiden Spannungen. Als Beispiel sei der Wert für  $36\text{ V}$  gesucht, bei Verwendung eines 12 Bit-ADC:

$$ADC\ Wert = \frac{2^{12}}{50\text{ V}} \cdot 36\text{ V} = 2949.12 \quad (4.4)$$

Da die von dem ADC generierten Werte quantisiert werden, ergibt sich 2949 als Wert.

4. Die beiden Parameter  $m$  und  $n$  für die Geradengleichung (Gleichung 4.1) müssen berechnet werden. Durch die beiden Punkte **A** und **B** (siehe Abbildung 4.5) ergibt sich eine Gerade in der Zweipunktform. Diese muss dann in die Geradengleichung in der Normalform umgewandelt werden. Folgendes Beispiel sei hierfür gegeben:

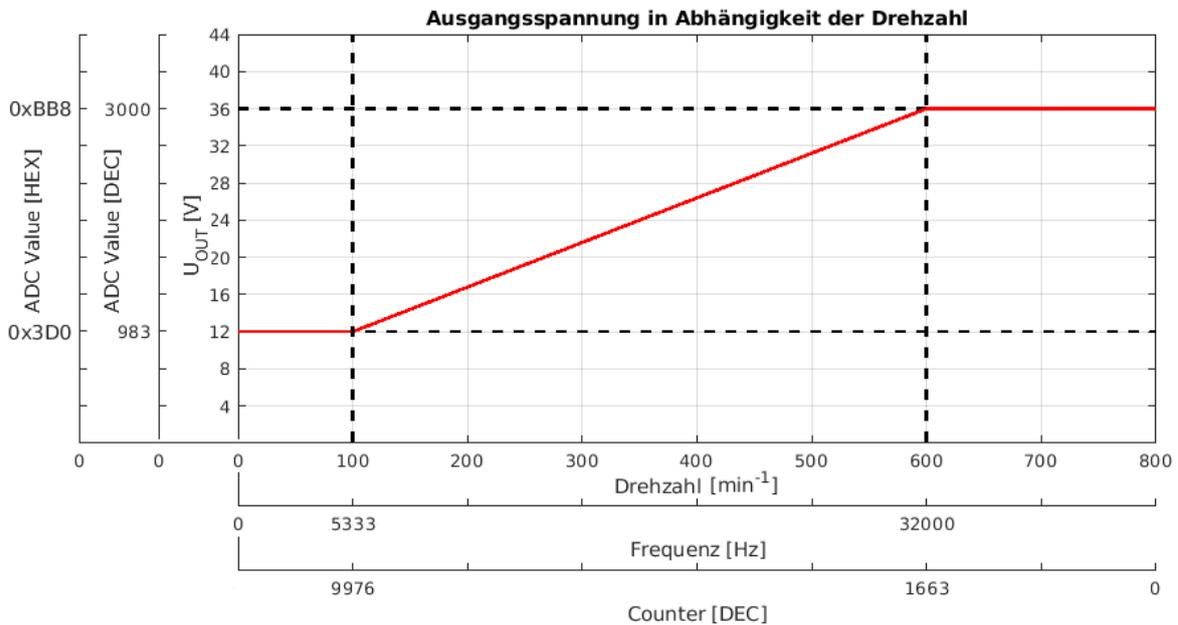
$$\begin{array}{l} \mathbf{A}(40\text{ min}^{-1}, 12\text{ V}) \\ \mathbf{B}(400\text{ min}^{-1}, 36\text{ V}) \end{array} \implies \begin{array}{l} \mathbf{A}(2133, \overline{33}\text{ Hz}, 0x3D7^1) \\ \mathbf{B}(21333, \overline{33}\text{ Hz}, 0xB85) \end{array} \quad (4.5)$$

$$\implies \begin{array}{l} m = 0.102396 \\ n = 764.58613 \end{array}$$

5. Skalierung der Parameter  $m$  und  $n$  zur einfacheren Verarbeitung im FPGA:  
Da die berechneten Parameter  $m$  und  $n$  reelle Zahlen sein können und insbesondere der Parameter  $m$  einen Wert  $< |1|$  annehmen kann, müssen diese für die Verarbeitung im FPGA vorbereitet werden. Da mit dem verwendeten *Mach XO2-1200* ein FPGA mit geringer Anzahl an Lookup-Tabellen gewählt wurde (der verwendete FPGA besitzt 1280 Lookup-Tabellen), ist für die Verwendung von Gleitkommazahlen keine Kapazität vorhanden. Um dennoch bei Werten  $< |1|$  ein verwertbares Ergebnis zu bekommen, werden die beiden Parameter jeweils mit  $2^{10} = 1024$  multipliziert. Dadurch werden die drei ersten Nachkommastellen in der Berechnung berücksichtigt. Durch das Assoziativgesetz kann das Ergebnis  $y$  der Geradengleichung durch 1024 geteilt werden, um den gesuchten Wert zu erhalten. Die Zahl  $2^{10} = 1024$  wurde deshalb als Multiplikator gewählt, da eine entsprechende Division sich durch eine einfache Verschiebung der Bits um 10 Stellen nach rechts erreichen lässt (siehe Code D.1 Zeile 368). Dies ist im FPGA ohne zusätzliche Divisionseinheit möglich.

<sup>1</sup>Die beiden Werte in hexadezimaler Schreibweise sind die ADC-Messwerte der beiden Spannungen von A und B

Mit den so generierten Parametern kann die Auswertung des Schrittsignals und die daraus resultierende Generierung des Sollwerts erfolgen. Zusammenfassend ist in Abbildung 4.6 der Verlauf der Spannungsgenerierung in den unterschiedlichen Einheiten dargestellt.



**Abbildung 4.6** Abhängigkeit des zu regelnden Sollwerts von der gemessenen Schrittfrequenz in den im FPGA verwendeten Einheiten. Die X-Achse ist in drei Einheiten abgebildet, in Umdrehungen pro Minute, in Hertz für die Frequenz der Schrittsignale und als Zählerwert des Zählers im FPGA, welcher mit einer Abtastrate von  $53.2 \text{ MHz}$  das ankommende Schrittsignal abtastet und die Zeit zwischen zwei aufsteigenden Flanken zählt. Die Y-Achse ist ebenfalls in drei Einheiten abgebildet. Einmal ist die zu generierende Ausgangsspannung in Volt dargestellt. Weiterhin ist der, der Ausgangsspannung entsprechende, 12 Bit-ADC-Messwert in dezimaler und hexadezimaler Schreibweise dargestellt.

## 4.2.2 Generierung des Sollwertes

Die Generierung des Sollwerts erfolgt periodisch zur Laufzeit auf dem FPGA. Sie erfolgt dabei zeitgleich und in derselben Periodizität wie das Auswerten des ADC sowie der Generierung des Schaltsignals für den Transistor (siehe Kapitel 4.3). Dafür wird aus der gemessenen Zeit zwischen zwei Schrittpulsen und den, in Kapitel 4.2.1 beschriebenen, Parametern ein Sollwert errechnet, der in dem Zustandsautomaten der Regelung mit dem Istwert des ADC verglichen und verarbeitet wird (siehe Kapitel 4.1.2). Da für die Geradengleichung zur Erzeugung des Sollwertes die Schrittfrequenz benötigt wird und nicht der Zählerzustand der Schrittzähler, muss dieser

Zählerwert vorher in die entsprechende Frequenz umgerechnet werden. Da der Frequenzzähler mit einer Abtastrate von  $53.2\text{ MHz}$ , also der eingestellten Taktfrequenz  $f_{clk}$  des FPGAs, zählt, muss diese Abtastrate durch den Zählerstand  $n$  geteilt werden, um die entsprechende Schrittfrequenz  $f_{step}$  zu erhalten:

$$f_{step}[\text{Hz}] = \frac{f_{clk}[\text{Hz}]}{n} \quad (4.6)$$

Da sich der Wert  $n$  des Zählers theoretisch in einem Bereich von  $n = 0$  bis  $n = 664000$  bewegen kann, was einer Frequenz von  $> f_{clk}$  bis  $80\text{ Hz}$ , beziehungsweise  $997500\text{ min}^{-1}$  bis  $1.5\text{ min}^{-1}$  entspricht, der Divisor also keinen festen Wert hat, muss eine Divisionseinheit in den FPGA implementiert werden. Dafür wurde ein fertiges Code-Beispiel verwendet (siehe Code D.2 [57]), welches eine Integer-Division mithilfe des Radix-2 Algorithmus [15] ermöglicht. Dieser Algorithmus benötigt einen Rechenschritt pro Bit, wodurch die Rechendauer vorhersagbar wird, beziehungsweise sich bei gleichbleibender Bitlänge nicht ändert. Diese Eigenschaft wird für die Parallelisierung der Zustandsautomaten benötigt (siehe Kapitel 4.3).

Im Folgenden wird der Ablauf des entsprechenden Zustandsautomaten erläutert, welcher in Abbildung 4.7 dargestellt ist:

1. Der Start des Zustandsautomaten wird durch den globalen Zähler ausgelöst, der unter anderem die Abtastrate des ADC bestimmt.
2. Es wird überprüft, ob die Divisionseinheit arbeitet, oder ob sie mit neuen Daten geladen und gestartet werden kann.
3. Es wird überprüft, ob der Schrittzählerwert innerhalb des in Kapitel 4.2.1 ausgewählten Drehzahlbereiches liegt, oder ob er größer, beziehungsweise kleiner ist.
  - Größer:  
Liegt der gemessene Wert über dem maximalen Zählerwert (`f_min_counter`), ist die Sollzahl also kleiner als die minimale Drehzahl, ab der aufwärts gewandelt werden soll (Drehzahl **A** in Abbildung 4.5), wird ein Statusregister (`CALC_FLAG`) gesetzt und der Wert, der der Eingangsspannung entspricht, (`MIN_ADC_VALUE`) als Sollwert verwendet. Es soll also nicht aufwärts gewandelt werden. Weiterhin wird die Divisionseinheit mit einem Platzhaltewert (`DUMMY_VALUE`) und der Taktfrequenz (`f_clk`) geladen und gestartet.

- Kleiner:  
Liegt der gemessene Wert unter dem minimalen Zählerwert (`f_max_counter`), ist die Solldrehzahl also größer als die maximale Drehzahl, bis zu der die Ausgangsspannung linear von der Drehzahl abhängig sein soll (Drehzahl **B** in Abbildung 4.5), wird ein Statusregister (`CALC_FLAG`) gesetzt und der Wert, der der maximal erzeugbaren Ausgangsspannung entspricht, (`MAX_ADC_VALUE`) als Sollwert verwendet. Weiterhin wird die Divisionseinheit mit einem Platzhaltewert (`DUMMY_VALUE`) und der Taktfrequenz (`f_clk`) geladen und gestartet.
  - Innerhalb des Bereichs:  
Liegt der gemessene Wert innerhalb des gewählten Bereiches, also zwischen der minimalen und maximalen Drehzahl (zwischen Punkt **A** und **B** in Abbildung 4.5), wird das Statusregister (`CALC_FLAG`) nicht gesetzt. Die Divisionseinheit wird mit dem Zählerwert des Frequenzzählers (`RESULT`) als Divisor und der Taktfrequenz (`f_clk`) als Dividend geladen und gestartet.
4. Das Bit zum Starten der Division wird wieder zurückgesetzt und solange gewartet, bis die Division beendet wurde.
  5. Es erfolgt die Berechnung der Geradengleichung mithilfe der in Kapitel 4.2.1 ermittelten Parameter  $m$  in  $n$  sowie die, durch die Division berechnete, Schritt-frequenz.
  6. Mithilfe des Statusregister (`CALC_FLAG`) wird überprüft, ob der durch die Geradengleichung berechnete Wert ( $FX$ ) verwendet werden soll oder ob der Frequenzzählerwert sich außerhalb des auszuwertenden Drehzahlbereiches befand. Befindet er sich innerhalb des auszuwertenden Drehzahlbereiches, wird der berechnete Wert ( $FX$ ) durch 1024 dividiert, indem eine zehnmalige bitweise Verschiebung nach rechts erfolgt (siehe Kapitel 4.2.1). Das Ergebnis wird dann als neuer Sollwert verwendet. Befindet sich der Frequenzzählerwert außerhalb des auszuwertenden Drehzahlbereiches, wurde der Sollwert schon am Anfang dieses Zustandsautomaten zugewiesen.
  7. Es erfolgt abschließend eine Überprüfung, ob der neue Sollwert sich innerhalb des erzeugbaren Spannungsbereichs befindet. Das ist eine Sicherheitsfunktion, die verhindern soll, dass falsch berechnete Sollwerte, die außerhalb des erzeugbaren Spannungsbereichs liegen, begrenzt werden, um eine Beschädigung des Aufwärtswandlers und des Schrittmotortreibers zu vermeiden.

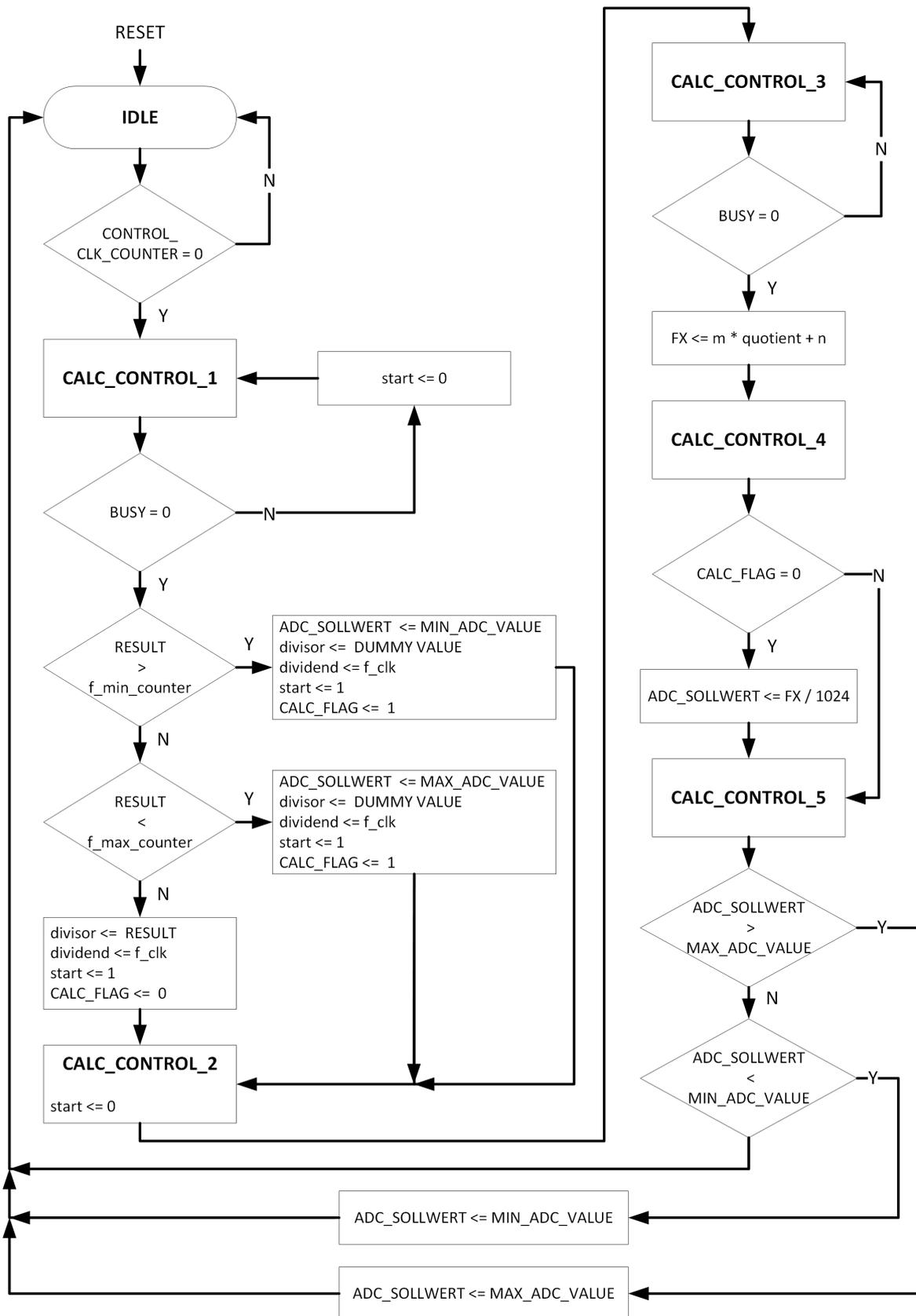


Abbildung 4.7 Ablauf des Zustandsautomaten FSM\_2 zur Berechnung des Sollwerts

### 4.3 Parallelisierung der Zustandsautomaten

Eine Verkürzung der Gesamtlaufzeit des Regelkreises (siehe Abbildung 4.1), also eingeschlossen der Auswertung des ADC sowie der der Schaltsignalgenerierung, hätte folgende Vorteile:

- Durch eine höhere Abtastrate des ADC würden Änderungen der Spannung schneller erkannt werden, was eine schnellere Reaktion auf Über- oder Unterschwinger ermöglichen würde [9, S. 112].
- Durch eine Reduzierung der Einschaltdauer des Transistors bei gleichzeitiger Erhöhung der Schaltfrequenz kann der Energietransfer zwischen der Spule und dem Ausgangskondensator in kürzeren Abständen mit dabei jeweils kleinerer Energiemenge stattfinden. Dadurch erfolgt der Transfer gleichmäßiger und weniger in großen Pulsen, was ein schnelleres Einschwingverhalten und schnellere Reaktionen auf Änderungen erlaubt [35] [7].

Um eine kürzere Gesamtlaufzeit zu ermöglichen, wurde die bis dahin entwickelte Firmware untersucht. Bis zu diesem Zeitpunkt erfolgte der Ablauf der, in Kapitel 4.1.1, 4.1.2 und 4.2 beschriebenen, Zustandsautomaten seriell. Es erfolgte zuerst die SPI-Kommunikation mit dem ADC, um den Istwert zu erhalten, dann wurde der Sollwert aus dem Schrittsignal berechnet und danach wurden beide Werte in dem Regelungsautomaten ausgewertet und der Transistor entsprechend eingeschaltet. In Abbildung 4.8 ist dieser zeitliche Ablauf als *FSM* dargestellt.

Wie gezeigt ist, dauert jeder Durchlauf  $6000\text{ ns}$ . Davon entfällt der größte zeitliche Anteil auf die SPI-Kommunikation, welche auf den zu der Zeit verwendeten ADC *MAX11665* zurückzuführen ist. Dieser hat eine maximale SPI-Taktfrequenz von  $8\text{ MHz}$  und überträgt bei jeder Kommunikation 16 Bit [33]. Da der verwendete SPI-Master den Systemtakt des FPGAs von  $53.2\text{ MHz}$  nur durch Zweierpotenzen teilen kann, war die maximal verwendbare SPI-Taktfrequenz  $53.2\text{ MHz} / 8 = 6.65\text{ MHz}$ . Die Übertragung der 16 Bit dauert dadurch insgesamt  $2406\text{ ns}$ . Dazu kommen noch  $188\text{ ns}$  durch die Wartezeit zwischen dem Schalten des Chipselect und dem Beginn der Übertragung. Das addiert sich zu  $2594\text{ ns}$ , was mehr als die Hälfte der gesamten Durchlaufzeit ist. Für die anschließende Berechnung des Sollwertes durch die Division wurden  $600\text{ ns}$  gemessen, diese ist in der Abbildung 4.8 als *calc* bezeichnet. Anschließend wird der Transistor, je nach Ergebnis der Regelung, eingeschaltet. Während der SPI-Kommunikation und der Berechnung ist der Transistor immer ausgeschaltet. Dadurch ist das Tastverhältnis nur in geringen Maßen veränderbar, dies ist insbesondere für den Sanftanlauf von Nachteil (siehe Kapitel 4.4).

Als erster Ansatz zur Verringerung der Gesamtdurchlaufzeit wurde die SPI-Kommunikation in einem eigenen Prozess geschrieben. Dadurch kann diese parallel zu der Berechnung und der Regelung sowie der Schaltdauer des Transistors laufen.

Dadurch steigt der Durchsatz um das 2.31-fache. Die Latenzzeit sinkt dagegen nicht, da vom Startzeitpunkt der SPI-Kommunikation bis zu dem daraus resultierendem Schalten des Transistors die gleiche Zeit vergeht. In Abbildung 4.8 ist diese Variante die mittlere mit *FSM\_1* und *FSM\_2*. Wie zu sehen ist, beträgt die Durchlaufzeit dadurch nur noch 2594 ns.

Um den Durchsatz weiter zu erhöhen und die Latenz zu senken, muss die Dauer der SPI-Kommunikation verringert werden. Da der verwendete *MAX11665* auf eine maximale SPI-Taktfrequenz von 8 MHz begrenzt ist, wurde ein anderer ADC gewählt. Der neue ADC ist der *MAX11116* von Maxim Integrated. Dieser hat zum einen den Vorteil, dass er pin- und protokollkompatibel mit dem *MAX11665* ist, und zum anderen, dass seine maximale SPI-Taktfrequenz 48 MHz beträgt [32]. Ein Nachteil ist, dass er statt der 12 Bit-Auflösung des *MAX11665* ADC nur eine Auflösung von 8 Bit hat. Dadurch sinkt die Auflösung (bezogen auf 50 V) von 0,0122 V auf 0,1953 V, was sich bezogen auf die letztendlich erreichten minimalen Rippelspannung (siehe Kapitel 5.1.2) als ausreichend genau herausgestellt hat. Durch diesen ADC kann die verwendete SPI-Taktfrequenz auf  $53.2 \text{ MHz} / 2 = 26.6 \text{ MHz}$  steigen. Dadurch dauert die Übertragung der 16 Bit insgesamt nur noch 602 ns. Mit den zusätzlichen 188 ns durch das vorzeitige Chipselect ergibt sich eine Gesamtdauer von 790 ns, was 30 % der Übertragungsdauer des *MAX11665* entspricht. Durch diese Verkürzung ergab sich die Möglichkeit, neben der SPI-Kommunikation, die Berechnung des Sollwerts und das Schalten des Transistors parallel zu schalten. In der Theorie ließe sich damit eine Regelfrequenz von ebendiesen 790 ns erreichen.

Allerdings stellte sich durch Versuche heraus, dass die daraus resultierende Einschaltdauer des Transistors zu kurz war, um die gewünschte maximale Spannung von 36 V zu generieren. Deshalb wurde eine Regelfrequenz von 1203 ns gewählt, womit sich auch die maximale Ausgangsspannung im Lastfall erreichen lässt. Dadurch konnte die Latenz von 3194 ns auf 1203 ns verringert werden und der Durchsatz konnte im Vergleich zu der ersten Variante ohne Parallelisierung um das 4.99-fache gesteigert werden.

Weiterhin konnte das Schwingen der Ausgangsspannung stark reduziert werden, zu sehen in Abbildung 4.9. Im Anhang unter Abbildung C.1 ist der zeitliche Verlauf der drei parallelen Prozesse sowie der SPI-Kommunikation während einer Simulation dargestellt. Sie zeigt die Abläufe der internen Registerzustände, die einzelnen Phasen der Zustandsautomaten sowie die Berechnung des Sollwertes inklusive der entsprechenden Zwischenwerte.

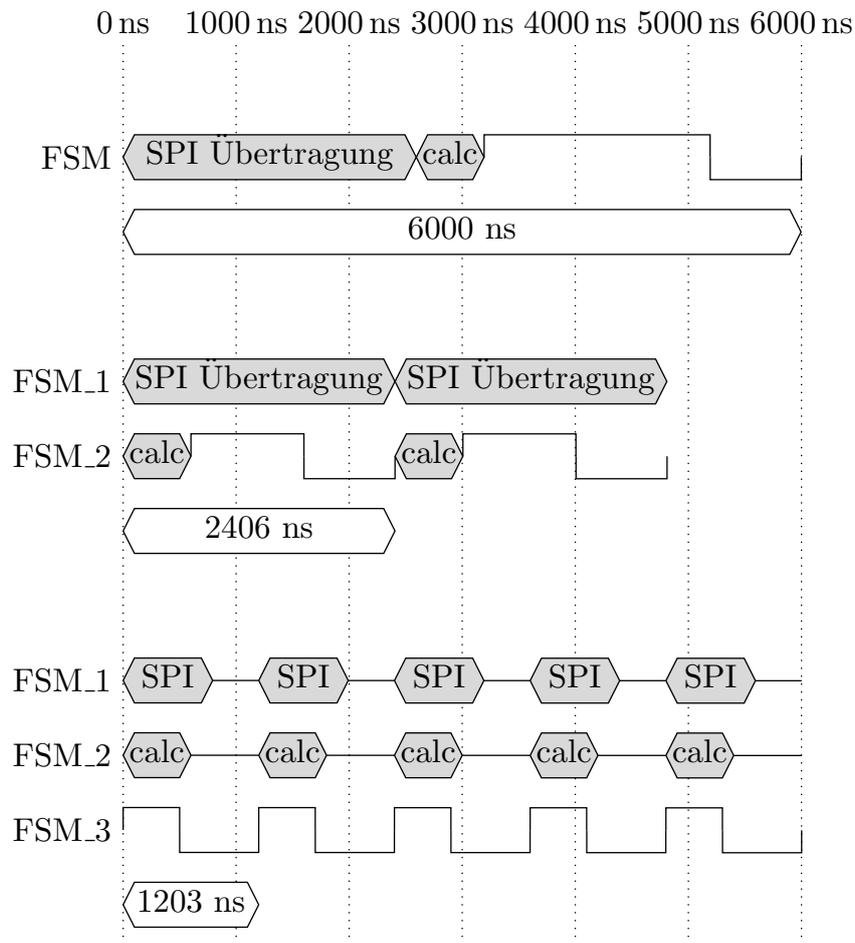
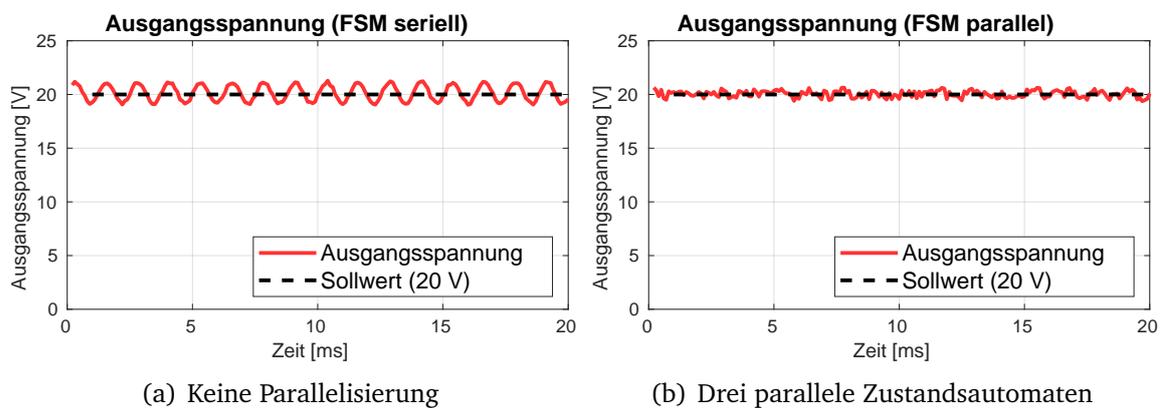


Abbildung 4.8 Zeitliche Verläufe der Zustandsautomaten bei keiner Parallelisierung, bei zwei parallelen Prozessen und bei drei parallelen Prozessen



(a) Keine Parallelisierung

(b) Drei parallele Zustandsautomaten

Abbildung 4.9 Beispielhafte Darstellung der erzeugten Schwingungen durch verschiedene Gesamtlauftzeiten des Regelkreises. Bei rein seriellen Zustandsautomaten (Zyklusdauer 6000 ns) ergeben sich deutlich sichtbare Schwingungen um den Sollwert. Bei Verwendung von parallelisierten Zustandsautomaten (Zyklusdauer 1203 ns) sinken die Spannungsabweichungen stark ab, es sind keine deutlichen Schwingungen zu erkennen.

## 4.4 Sanftanlauf

Die in Kapitel 2.2.4 erläuterte Hysterese-Regelung des Aufwärtswandlers hat die Eigenschaft, dass, wie der Name schon sagt, die Einschaltdauer des Transistors immer gleich lang ist, unabhängig vom Strombedarf des, an den Aufwärtswandler angeschlossenen, Verbrauchers oder der zu erzeugenden Ausgangsspannung. Dadurch wird mit jedem Puls die gleiche Energie von der Induktivität in den Ausgangskondensator des Auswärtswandlers transportiert. Das führt in folgenden Situationen zu unerwünschtem Verhalten:

1. Die zu erzeugende Ausgangsspannung ist nur geringfügig größer als die Eingangsspannung, zum Beispiel wenn von  $12\text{ V}$  auf  $14\text{ V}$  gewandelt werden soll. Insbesondere bei geringem Stromverbrauchs des Verbrauchers werden die Pulse nur mit großem zeitlichem Abstand ausgelöst. Durch den geringen Stromverbrauch steigt die Spannung am Ausgang zuerst stark über den gewünschten Sollwert, danach sinkt die Spannung nur sehr langsam wieder unter den Sollwert, wodurch ein neuer Puls ausgelöst wird. Das Ergebnis ist eine Ausgangsspannung mit großer Rippelspannung.
2. Wenn es zu einem positiven Sprung des Sollwerts kommt, zum Beispiel wenn die Drehzahl eines Schrittmotors aus dem Stillstand ohne Beschleunigungsphase auf eine Geschwindigkeit steigt. Das passiert zum Beispiel in 3D-Druckern bei den sogenannten Jerks<sup>2</sup>. Dadurch wird sehr plötzlich viel Leistung benötigt, der Transistor wird eine Zeit lang bei jedem Regeldurchgang geschaltet. Dadurch steigt die Ausgangsspannung zwar schnell auf den geforderten Wert, allerdings steigt die Stromentnahme des Aufwärtswandlers aus dem versorgenden Netzteil massiv an (siehe Abbildung 5.10a und b). Dadurch bricht die Spannung an dem Netzteil ein, viele digital geregelte Netzteile schalten sogar als Sicherheitsfunktion komplett ab<sup>3</sup>. Das ist in den meisten Anwendungen nicht akzeptabel.

In den meisten Spannungswandlern ist als vorbeugende Maßnahme eine sogenannte Softstart<sup>4</sup> Funktion eingebaut. Diese funktionieren zumeist entweder über eine Strommessung des Primärstroms in die Induktivität oder mithilfe eines externen Kondensators, der über eine interne Stromquelle langsam geladen wird. Die langsam ansteigende Spannung an dem Kondensator gibt dann die Steigrate der zu erzeugenden Ausgangsspannung vor [36]. Da bei der Entwicklung der Platine versäumt

<sup>2</sup>Ein Jerk, englisch für Ruck, bedeutet bei 3D-Druckern den Sprung aus dem Stillstand auf eine Geschwindigkeit ohne Beschleunigungsphase, von der aus der Motor mit einer Beschleunigungsphase eine höhere Geschwindigkeit erreicht

<sup>3</sup>Unterspannungsabschaltung

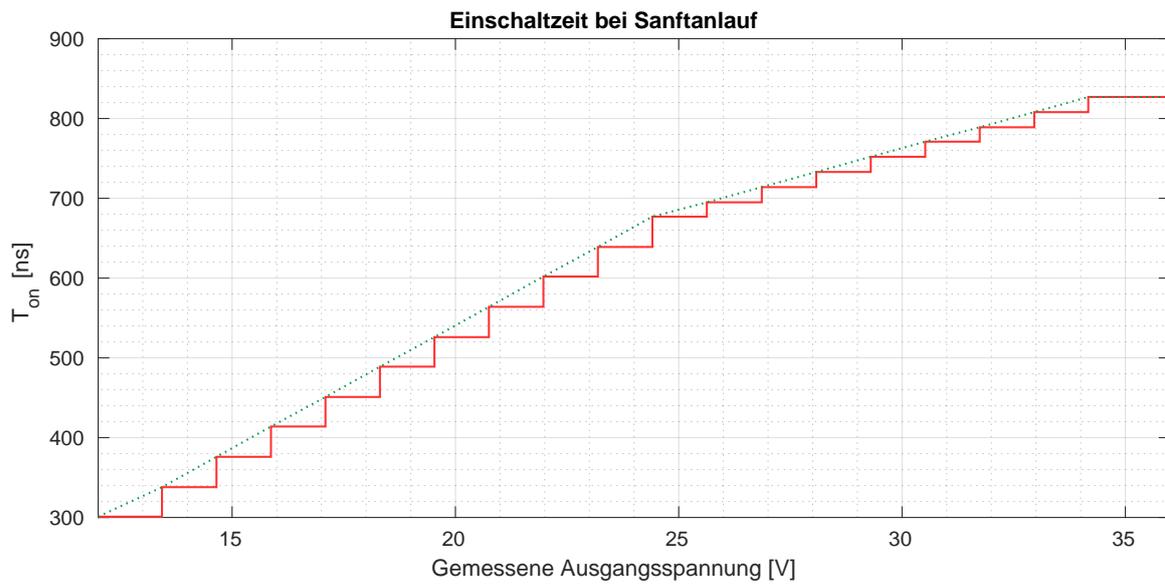
<sup>4</sup>Sanftanlauf

wurde, eine Möglichkeit der Primärstrommessung einzubauen, musste eine andere Möglichkeit gefunden werden, den Primärstrom zu begrenzen.

Eine häufig verwendete Methode, diese Strompulse vor dem Netzteil abzufangen, ist es, Kondensatoren mit einer hohen Kapazität ( $470 \mu F$  bis mehrere  $1000 \mu F$ ) parallel zu dem Netzteil vor die zu versorgende Schaltung zu schalten. Allerdings sind entsprechende Kondensatoren zum einen, aus Platzgründen, sehr groß und ein weiteres Bauelement, welches zusätzliche Kosten erzeugt. Zum anderen können die Kondensatoren nicht alle Strompulse dämpfen, wenn diese zu lange dauern oder zu groß werden. Dem entwickelten Modul wurde, zu Testzwecken, ein  $10000 \mu F$  Elektrolytkondensator vorgeschaltet. Das versorgende Netzteil aktivierte trotzdem die Unterspannungsabschaltung, weswegen die Entwicklung und Implementierung eines Sanftanlaufs nötig war.

Durch empirische Versuche stellte sich heraus, dass eine von der Ausgangsspannung abhängige Einschaltzeit eine Reduktion des Strompulses ermöglicht. Das bedeutet, dass bei einer, im Vergleich zur Eingangsspannung, niedrigen Spannung, nur kurze Pulse nötig sind, um die Zielspannung zu erreichen. Hohe Spannungen lassen sich jedoch durch kurze Pulse nicht erreichen, weswegen bei höheren Spannungen die Länge der Pulse erhöht wird. Es wurden mehrere Arten der Abhängigkeiten der Pulsbreite von der Spannung getestet und die in Abbildung 4.10 gezeigte Abhängigkeit stellte sich als gut funktionierend heraus.

Implementiert wurde dieser Sanftanlauf dabei über eine Lookup-Tabelle, wobei nicht jedem, von dem ADC diskret gemessenem, Spannungswert ein eigener Wert zugeordnet wurde. Die Zuordnung der Pulsdauer erfolgt für Spannungsbereiche. Einerseits nimmt die Pulsbreite mit steigender Spannung so schnell zu, dass hohe Spannungen erreicht werden können, andererseits konnte die Höhe der Strompulse um bis zu  $58.28 \%$  reduziert werden (siehe Abbildung 5.10 und Kapitel 5.1.4). Eine Auswirkung ist jedoch, dass die Anstiegszeit der Ausgangsspannung vergrößert wird. Dabei wurde eine Verlängerung der Anstiegszeit von bis zu  $409.9 \%$  gemessen (siehe Abbildung 5.9). Durch die Verwendung von kürzeren Pulsen im niedrigeren Spannungsbereich, dauert das Erreichen der Sollspannung länger als bei der Verwendung von längeren Pulsen.



**Abbildung 4.10** Die Einschaltdauer  $T_{on}$  des Transistors hängt bei Verwendung des Sanftanlaufs von der aktuell erzeugten Ausgangsspannung ab. Je niedriger die Spannung ist, desto kürzer sind die Schaltpulse.

# 5 Ergebnisse

## 5.1 Charakterisierung des Gleichspannungswandlers

Um das Verhalten des entwickelten Aufwärtswandlers zu charakterisieren, wurde eine Modulplatine nur mit den Bauelementen für den Aufwärtswandler bestückt und die erzeugte Spannung herausgeführt. Weiterhin wurden die SMD-Ausgangskondensatoren nicht bestückt und der THT-Ausgangskondensator mittels Steckkontakten austauschbar gehalten. Dadurch können die Auswirkungen einzelner Kondensatoren mit unterschiedlichen Kapazitäten und Bauarten gemessen werden.

### 5.1.1 Wirkungsgrad des Aufwärtswandlers

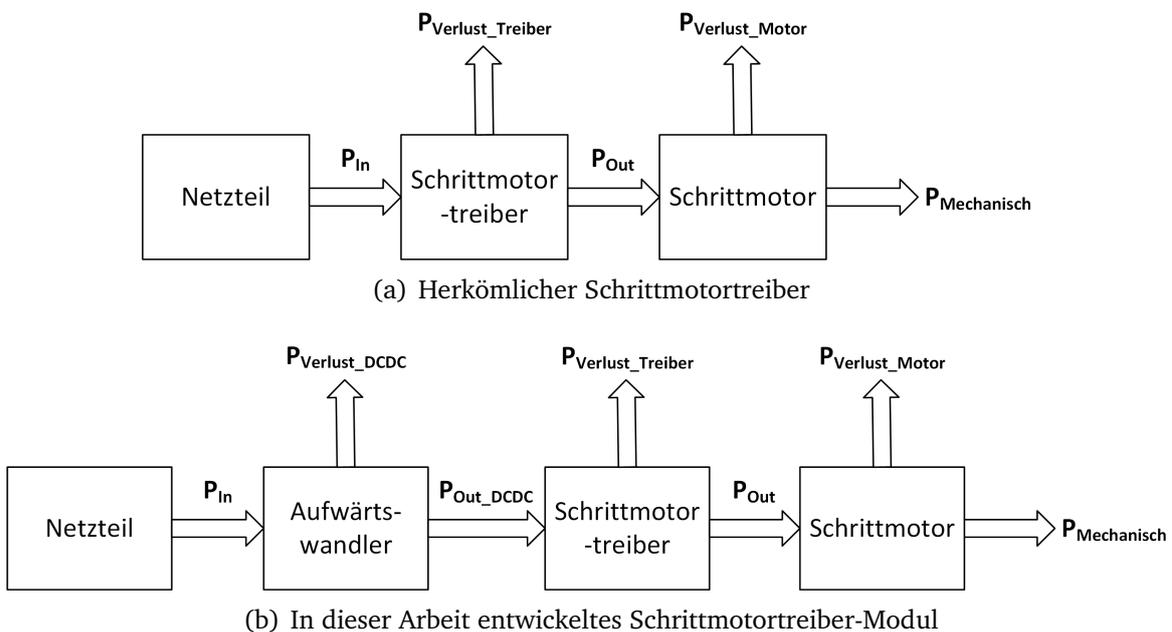
Eine hohe Effizienz eines Spannungswandlers ist ein häufiges Verkaufsargument und Voraussetzung für viele Anwendungen. Da es das Ziel dieser Entwicklung ist, bestehende Schrittmotortreiber ersetzen zu können, ist ein Vergleich des Wirkungsgrads zwischen dem bestehenden Schrittmotortreiber und dem Entwickelten von Interesse. Insbesondere für Anwendungen, die batterie- oder akkubetrieben sind oder die sehr lange Laufzeiten haben (zum Beispiel Überwachungskameras), ist ein möglichst hoher Wirkungsgrad interessant. Elektrotechnische Systeme geben die verlorene Energie zumeist in Wärme ab. In Gleichspannungswandlern entsteht diese hauptsächlich in der Leistungsinduktivität und dem Leistungstransistor durch Schaltverluste. Durch den Diodenstrom und dem Spannungsabfall an der Diode entstehen ebenfalls Verluste, wobei diese sich durch die Wahl von Dioden mit niedriger Schwellenspannung und schnellerer Schaltfähigkeit (zum Beispiel Schottky Dioden) reduzieren lassen. Der verwendete Ausgangskondensator hat ebenfalls einen Einfluss auf die Verluste. Da er in einem Gleichspannungswandler permanent be- und entladen wird fällt an dem ESR des Kondensators eine entsprechende Leistung ab. Insbesondere Elektrolytkondensatoren haben bauartbedingt einen höheren äquivalenten Serienwiderstand als zum Beispiel Keramikkondensatoren [18, S. 4]. Zusätzlich ist gerade bei Elektrolytkondensatoren eine Erwärmung sehr schädlich, da das verwendete Elektrolyt durch die Wärme schneller austrocknet und sie schneller altern. Nach

[4, S. 9] führt das Absenken der Betriebstemperatur um  $10\text{ K}$  zu einer Verdoppelung der Lebensdauer.

Die Energiekette eines herkömmlichen Schrittmotortreibers ist in Abbildung 5.1a dargestellt. Sie besteht, vereinfacht, aus dem versorgenden Netzteil, dem Schrittmotortreiber und dem Schrittmotor. Dabei gibt es zwei vorherrschende Energieverluste, zum einen die des Motortreibers und die des Motors. Bei dem Schrittmotortreiber sind die Verluste hauptsächlich Schaltverluste der H-Brücken. Bei den Schrittmotoren ist der Widerstand der Kupferwicklung sowie die Wirbelstromverluste in den Statorblechen Hauptursache für Wärmeverluste [24]. Bei dem, in dieser Arbeit entwickelten System, kommt dazu noch der Aufwärtswandler inklusive seinen Verlusten (siehe Abbildung 5.1b). Verluste durch Leitungen sowie des Netzteils wurden dabei vernachlässigt, da diese unabhängig von dem Schrittmotortreiber und Schrittmotor sind.

Der Wirkungsgrad wird aus dem Verhältnis der benötigten Eingangs- und der erzeugten Ausgangsleistung berechnet:

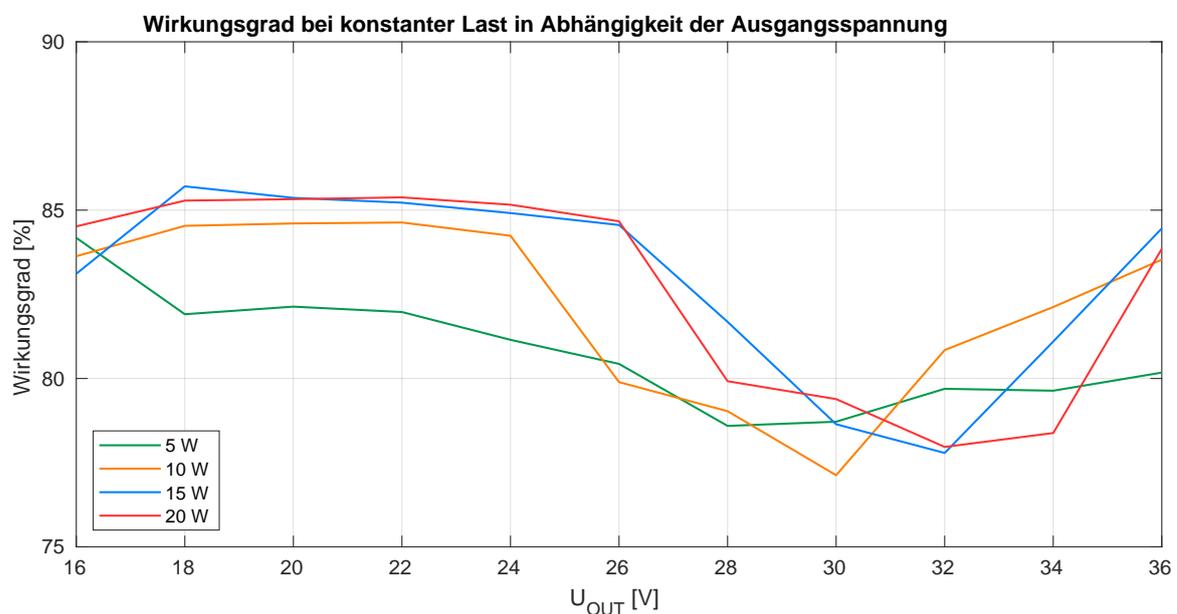
$$\eta = \frac{P_{out}}{P_{in}} \quad (5.1)$$



**Abbildung 5.1** Energiefluss des kompletten Testaufbaus. Dabei sind die jeweiligen Verlustleistungen sowie die Nutzleistungen die abgegeben werden dargestellt.

Um den absoluten Wirkungsgrad eines Treiber-Motor-Systems zu bestimmen, müsste die mechanische Ausgangsleistung mit der elektrischen Eingangsleistung verglichen werden. Allerdings war durch mangelnde Messmöglichkeiten eine genaue Messung der mechanischen Ausgangsleistung nicht möglich. Weiterhin ist der verwendete Schrittmotortreiber *TMC2130* unabhängig von dem entwickelten Aufwärtswandler-System, sein Wirkungsgrad soll nicht in die Messungen mit einfließen. Es soll nur der Wirkungsgrad des Aufwärtswandlers gemessen werden. Deshalb wurde anstatt der Treiber-Motor-Last eine einstellbare ohmsche Last verwendet und direkt von dem Aufwärtswandler versorgt. Dafür wurde eine Platine nur zum Teil bestückt, die Schrittmotortreiber-Komponenten wurde weggelassen und die erzeugte Ausgangsspannung herausgeführt. Dabei wurden zwei Arten von Messungen durchgeführt. Für die erste Messung wurde bei einer fest eingestellten Last (5 W, 10 W, 15 W, 20 W) der generierbare Ausgangsspannungsbereich von 16 V bis 36 V durchgeföhrt und dabei die Ein- und Ausgangsströme gemessen. Mithilfe der konstanten Eingangsspannung von 12 V und der eingestellten Ausgangsspannung wurden Ein- und Ausgangsleistung und daraus der Wirkungsgrad bestimmt.

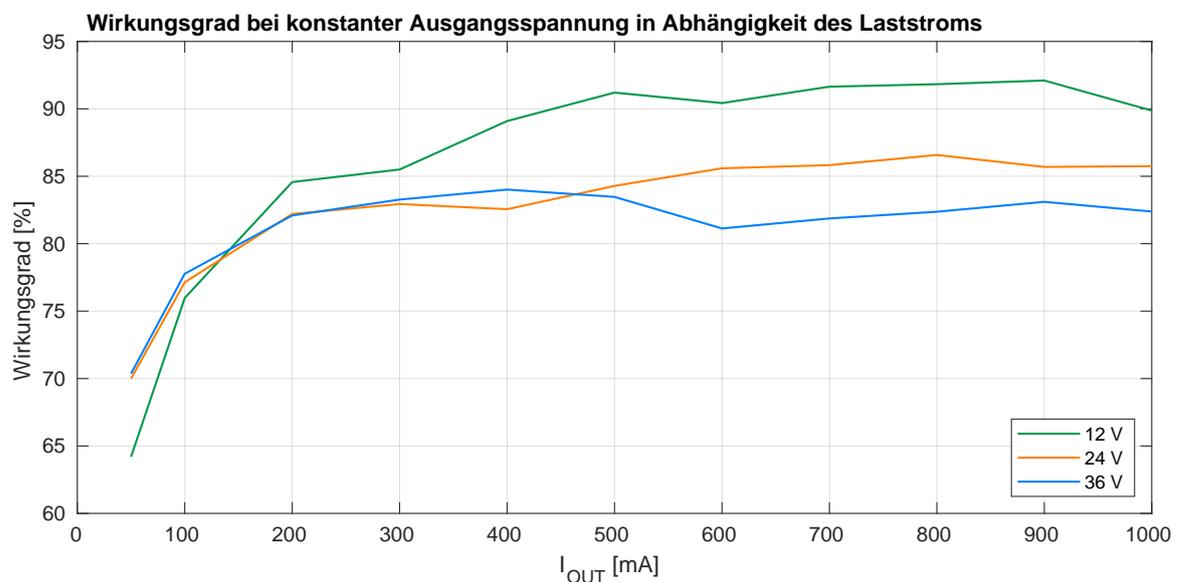
Die Ergebnisse sind in Abbildung 5.2 dargestellt. Der Wirkungsgrad ist relativ unabhängig von der generierten Ausgangsspannung und der entnommenen Leistung. Sie befindet sich durchgehend in dem Bereich um 80 %. Dabei ist festzustellen, dass der Wirkungsgrad in dem Spannungsbereich von 16 V - 24 V etwas höher ist als in dem Spannungsbereich 24 V - 34 V. Weiterhin führt eine höhere Last im Schnitt zu einem etwas höherem Wirkungsgrad.



**Abbildung 5.2** Wirkungsgrad über Spannungsbereich bei unterschiedlichen Ausgangsleistungen

Für die zweite Messung wurde für fest eingestellte Ausgangsspannungen (12 V, 24 V, 36 V) der Ausgangsstrom in einem Bereich von 50 mA bis 1000 mA variiert. Dabei wurde der Ein- und Ausgangsstrom sowie die Ausgangsspannung gemessen. Mithilfe der konstanten Eingangsspannung von 12 V und des eingestellten Ausgangsstroms wurden Ein- und Ausgangsleistung und daraus der Wirkungsgrad bestimmt. Dabei ist zu beachten, dass für die 12 V Ausgangsspannung der Aufwärtswandler deaktiviert wurde, das heißt, der Schalttransistor war konstant geöffnet. Dadurch gibt die Kennlinie bei 12 V die Verluste, die unabhängig von den Schaltverlusten auftreten, wieder. Das sind der Spannungsabfall über die Diode, der Energiebedarf von den ICs und dem 3.3 V Spannungsregler sowie die Verluste an den Widerständen und Kondensatoren.

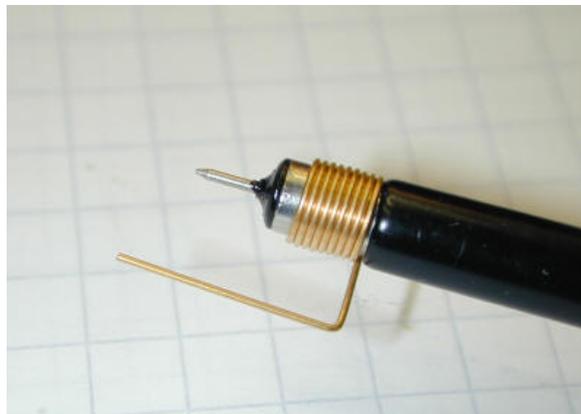
Die gemessenen Ergebnisse sind in Abbildung 5.3 dargestellt. Der deaktivierte Aufwärtswandler (12 V Ausgangsspannung) hat den höchsten Wirkungsgrad, da keine Schaltverluste auftreten. Dieser Wirkungsgrad könnte, bei den derzeitigen verwendeten Komponenten, auch bei weiterer Optimierung der Regelung und des Schaltvorgangs nicht überschritten werden. Weiterhin ist bei einer Ausgangsspannung von 36 V und einem Ausgangsstrom von 1 A das Maximum an Ausgangsleistung erreicht, die dieser Aufwärtswandler kurzzeitig aushalten kann. Dabei wurden an der Induktivität Temperaturen von über 100 °C gemessen. Allerdings befindet sich diese Ausgangsleistung von 36 W auch außerhalb des benötigten Leistungsbereichs kleinerer Schrittmotoren, für die dieses Modul entwickelt wurde. Der Wirkungsgrad bleibt unabhängig von der Ausgangsspannung ab einem Laststrom von 150 mA relativ konstant zwischen 80 % und 85 %.



**Abbildung 5.3** Wirkungsgrad in Abhängigkeit des Ausgangsstroms bei unterschiedlichen Ausgangsspannungen

### 5.1.2 Störungen in der Ausgangsspannung

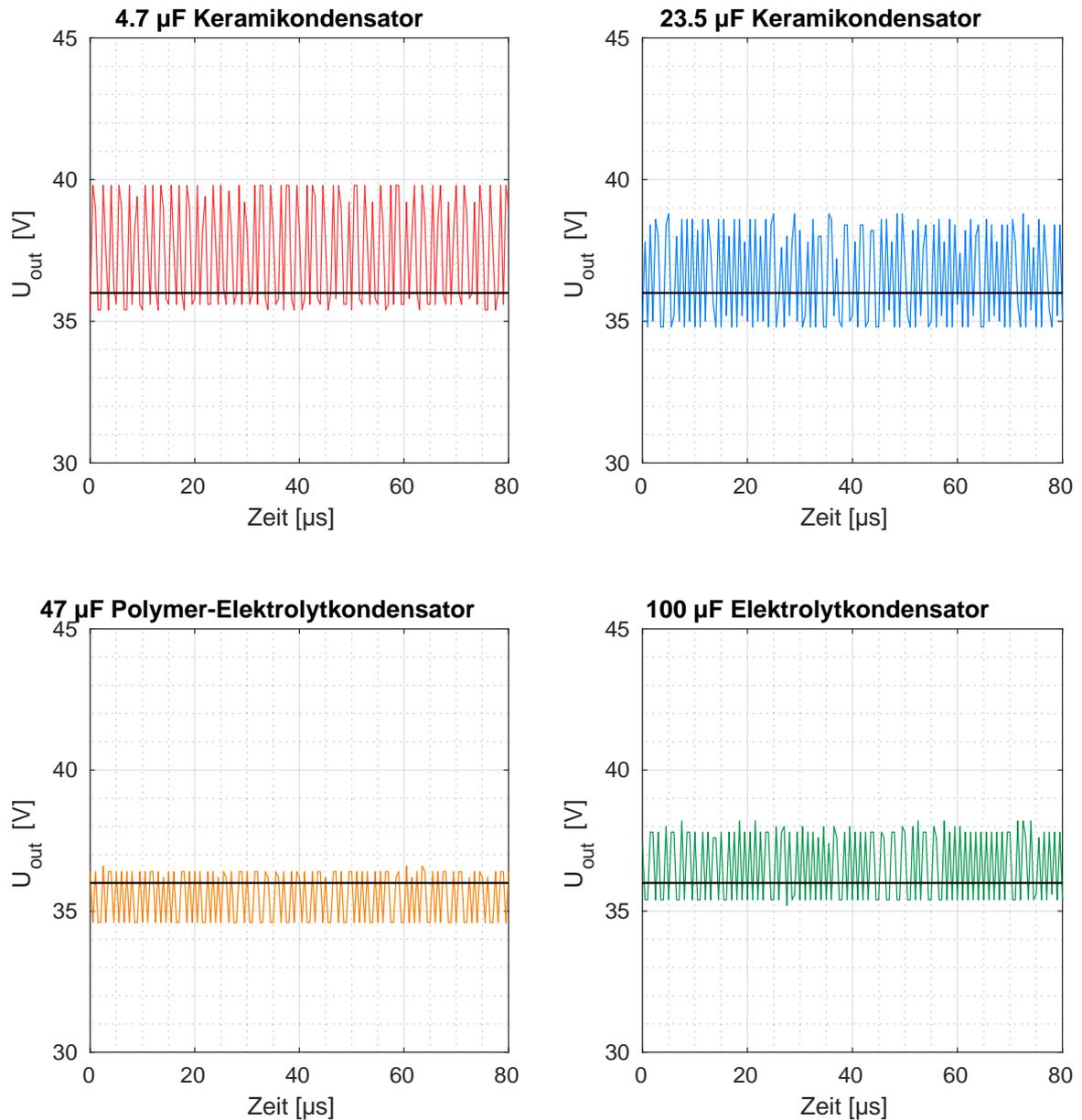
Die erzeugte Ausgangsspannung des Aufwärtswandlers ist keine ideale Gleichspannung, sondern enthält zusätzliche, die Sollspannung überlagernde, hochfrequente Wechselstromanteile [67]. Diese setzen sich aus zwei Teilen zusammen, der Rippelspannung und den hochfrequenten Schaltspitzen des Transistors [62]. Die Rippelspannung entsteht durch die periodischen Ladungen und Entladungen der Energiespeicher des Aufwärtswandlers (Ausgangskondensator und Induktivität). Die Rippelspannung macht zumeist nur wenige Prozent der Ausgangsspannung aus [67] und kann durch Verwendung von größeren Ausgangskondensatoren und Induktivitäten verringert werden. Die hochfrequenten Schaltspitzen, oft auch *Noise*<sup>1</sup> genannt, entsteht durch das schnelle Ein- und Ausschalten des Transistors und resultieren in hochfrequenten harmonischen Oberschwingungen [62]. Um diese Störungen sicher messen zu können, muss darauf geachtet werden, möglichst nahe am Ausgang des Aufwärtswandlers zu messen und lange Messleitungen oder Schleifen zu vermeiden. Dafür kann bei Tastköpfen eine Massefeder (siehe Abbildung 5.4) anstatt der sonst üblichen Masseklemme verwendet werden. Durch die hochfrequenten Schaltvorgänge des Aufwärtswandlers können über lange ungeschirmte Leitungen am hochimpedanten Oszilloskop durch das entstehende magnetische oder elektrische Feldsignale eingekoppelt werden, die in Wirklichkeit gar nicht in der Ausgangsspannung zu finden sind [67].



**Abbildung 5.4** Tastkopf eines Oszilloskops mit aufgesetzter Massefeder  
Bildquelle [14]

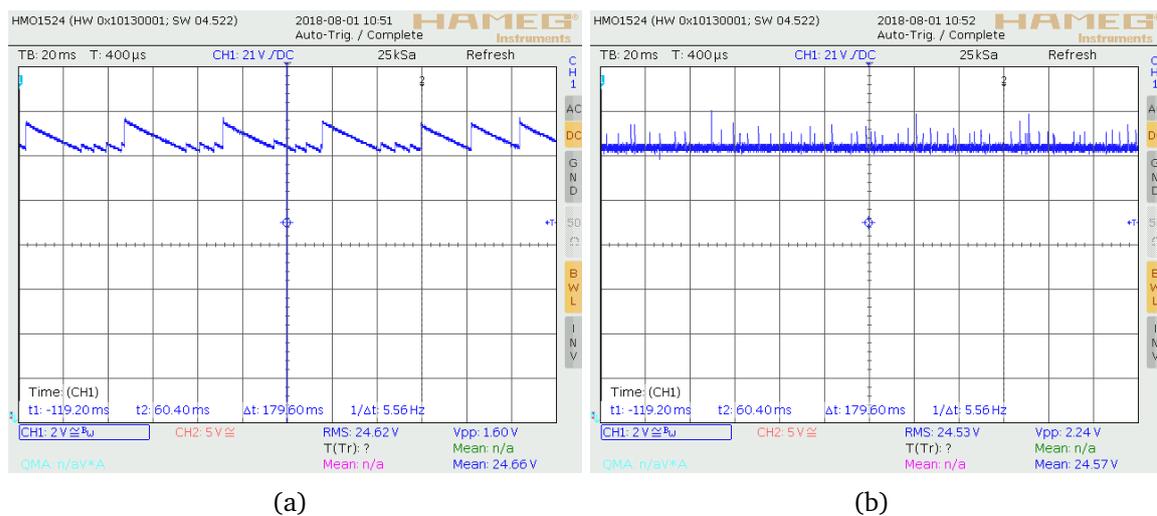
Zunächst wurde der Einfluss der verschiedenen Bauarten von Kondensatoren untersucht (siehe Abbildung 5.5). Dabei stellte sich heraus, dass der Polymer-Elektrolytkondensator sowie der Elektrolytkondensator die geringsten Störungen erzeugen. Da allerdings firmenseitiges Interesse bestand, den Auswärtswandler, aufgrund der räumlichen Größe sowie der Lebensdauer, mit Keramikkondensatoren zu verwenden, wurde für die weiteren Messungen zusätzlich zu dem  $47 \mu F$  Polymer-Elektrolytkondensator ein  $23.5 \mu F$  Keramikkondensator verwendet (siehe Abbildung 5.7).

<sup>1</sup>Englisch für Rauschen



**Abbildung 5.5** Ausgangsspannung bei konstanter resistiver Last mit verschiedenen Ausgangskondensatoren. Die Sollspannung beträgt 36 V. Keine Verwendung des Sanftanlaufs. Es zeigt sich deutlich, dass die Verwendung des 47 μF Polymer-Elektrolytkondensators die geringsten Spannungsabweichungen erzeugt, weswegen dieser für die weiteren Tests verwendet wird. Da Keramikkondensatoren den Vorteil haben, durch eine erhöhte Betriebstemperatur weniger stark zu altern als Elektrolytkondensatoren [8], wird zusätzlich der 23.5 μF Keramikkondensator für weitere Tests verwendet.

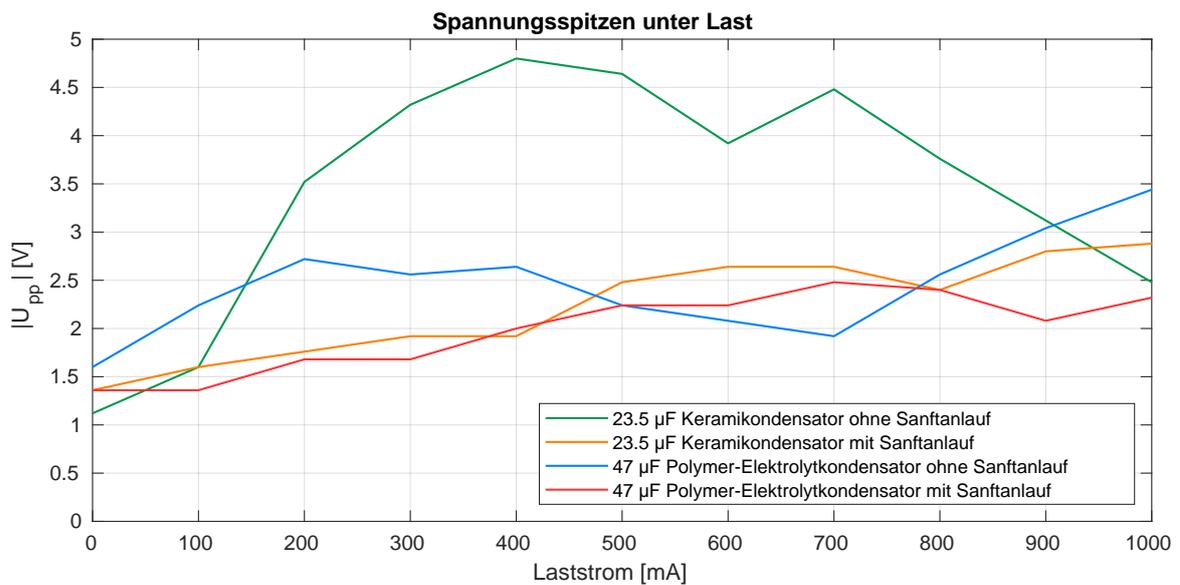
Weiterhin benötigt der Aufwärtswandler eine Grundlast am Ausgang, damit die Regelung funktionieren kann (siehe Kapitel 2.2.2). Ist keine Grundlast oder nur eine zu geringe vorhanden, kann die Energie, die im Ausgangskondensator vorhanden ist, nicht schnell genug verbraucht werden und es kommt zu einer sägezahnartigen Signalform (siehe Abbildung 5.6a). Ist eine Mindestlast vorhanden, in der die gespeicherte Energie des Ausgangskondensators verbraucht werden kann, so ergibt sich die Ausgangsspannung wie in Abbildung 5.6b.



**Abbildung 5.6** Vergleich der Rippelspannung der Ausgangsspannung bei Leerlauf mit der Grundlast von  $12\text{ k}\Omega$  (a) und  $200\ \Omega$  Last (b). Als Ausgangskondensator wurde ein  $47\ \mu\text{F}$  Polymer-Elektrolytkondensator verwendet.

Für die Messung der Störungen wurde eine einstellbare resistive Last verwendet, um die Abhängigkeit der Störungen von dem Laststrom messen zu können, und um den Einfluss einer sich zeitlich ändernden Last auszuschließen. Als Messinstrument wurde dabei das *HMO1524* Oszilloskop von *ROHDE & SCHWARZ GmbH & Co. KG* benutzt unter Verwendung der  $V_{PP}$  Messfunktion. Diese misst die Spannungsdifferenz zwischen positiver und negativer Spitzenspannung. Dabei wurde die Messung mit einem  $47\ \mu\text{F}$  Polymer-Elektrolytkondensator und einem  $23.5\ \mu\text{F}$  Keramik Kondensator, jeweils mit und ohne Sanftanlauf, durchgeführt.

Die Ergebnisse sind in Abbildung 5.7 dargestellt. Es ist festzustellen, dass die Verwendung des Sanftanlaufs bei einem Keramikcondensator die Rippelspannung um bis zu 60 % reduziert. Bei Verwendung des Polymer-Elektrolytkondensators ließ sich die Rippelspannung um bis zu 33 % durch die Verwendung des Sanftanlaufs reduzieren. Eine weitere Beobachtung war, dass, bei Verwendung des Schrittmotortreibers und Schrittmotor als Last, die Rippelspannung zum Teil akustisch hörbar ist. Diese Geräusche sind relativ leise, aber bei Verwendung des ruhigen *StealthChop* Betriebsmodus im Vergleich zum Betrieb an einem Labornetzteil gut wahrnehmbar.

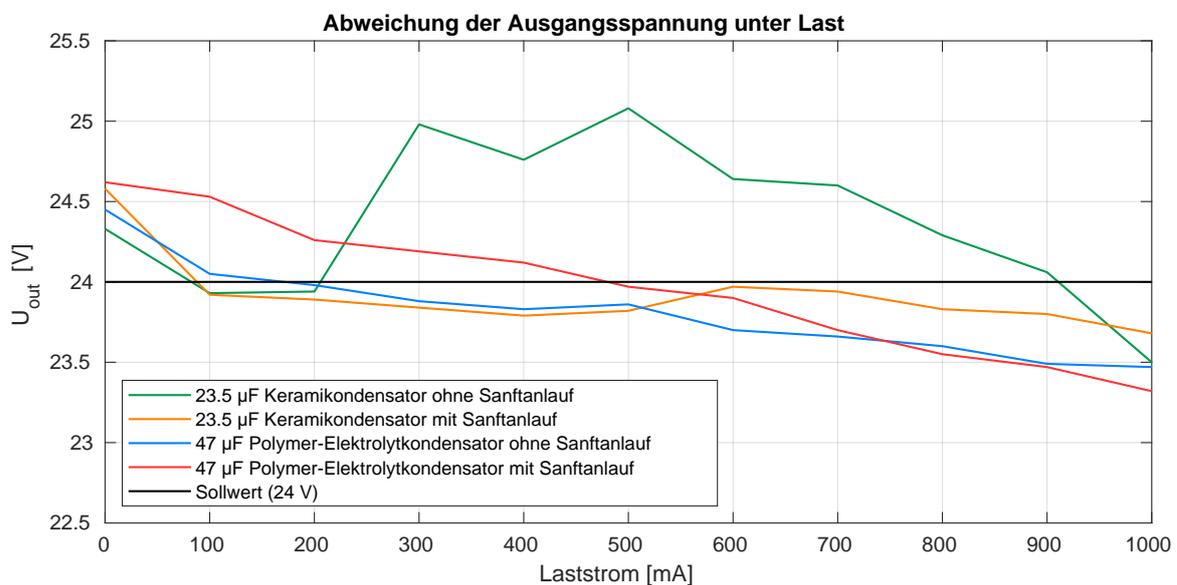


**Abbildung 5.7** Amplitude der Spannungsspitzen auf einer konstanten Ausgangsspannung von 24 V in Abhängigkeit von dem Laststrom bei verschiedenen Kondensator-Bauarten. Jeweils mit und ohne Sanftanlauf.

### 5.1.3 Lastregulierung

Ein weiteres Merkmal eines Spannungswandlers sind die auftretenden Spannungsabweichungen von der eigentlichen Sollspannung, weswegen deren Amplituden unter Last untersucht wurden. Dafür wird mithilfe einer einstellbaren resistiven Last der entnommene Ausgangsstrom stufenweise erhöht und die Spannungsabweichungen gemessen. Da sich in Kapitel 5.1.2 gezeigt hat, dass die Verwendung eines Polymer-Elektrolytkondensators die niedrigste Rippelspannung erreicht, wird die folgende Messung mit einem  $47 \mu\text{F}$  Polymer-Elektrolytkondensators durchgeführt. Weiterhin wird die Messung mit einem  $23.5 \mu\text{F}$  Keramikkondensator durchgeführt, da daran firmenseitiges Interesse besteht. Zusätzlich wird der Einfluss der entwickelten Sanftanlauf-Funktion auf die Spannungsabweichungen untersucht.

Die Messergebnisse in Abbildung 5.8 zeigen, dass ohne den Sanftanlauf die Spannungsabweichungen von dem  $24 \text{ V}$  Sollwert, unter Verwendung des  $23.5 \mu\text{F}$  Keramikkondensators, mit  $1.05 \text{ V}$  bei  $500 \text{ mA}$  Laststrom am größten sind. Mithilfe des Sanftanlaufs lässt sich die Spannungsabweichung bei einer angeschlossenen Last auf  $0.3 \text{ V}$  bei  $1000 \text{ mA}$  Laststrom reduzieren. Die Spannungsabweichungen unter Verwendung des  $47 \mu\text{F}$  Polymer-Elektrolytkondensators sind ohne Sanftanlauf niedriger als mit Sanftanlauf. Die maximale Spannungsabweichungen bei einer angeschlossenen Last unter Verwendung des Sanftanlaufs beträgt  $0.7 \text{ V}$  bei  $1000 \text{ mA}$  Laststrom, ohne den Sanftanlauf beträgt die maximale Spannungsabweichung  $0.55 \text{ V}$  bei  $1000 \text{ mA}$  Laststrom.



**Abbildung 5.8** Spannungsabweichung von der konstant eingestellten Ausgangsspannung ( $24 \text{ V}$ ), abhängig von einem Laststrom einer resistiven Last bei verschiedenen Kondensator-Bauarten. Jeweils mit und ohne Sanftanlauf.

### 5.1.4 Spannungssprung an resistiver Last

Eine wichtige Eigenschaft des Aufwärtswandlers ist die Zeit, die benötigt wird, um von einem deaktivierten Auswärtswandler, die Ausgangsspannung des Aufwärtswandlers ist gleich der Eingangsspannung, auf die maximale Ausgangsspannung zu steigen. Das ist für dynamische Systeme, in dem das entwickelte Modul eingesetzt werden soll wichtig, da ein plötzlicher Sprung der Drehzahl des Schrittmotors aus dem Ruhezustand je nach Anwendung häufig vorkommen kann. Idealerweise würde mit dem Sprung der Drehzahl auch die generierte Spannung sprunghaft steigen. Wenn der Aufwärtswandler jedoch länger braucht, um die gewünschte Spannung zu erreichen, der Schrittmotor sich aber schon mit einer hohen Solldrehzahl dreht, kann es je nach notwendigem Drehmoment sein, dass es aufgrund der generierten gegen-elektromagnetische Kraft (siehe Kapitel 2.1.5) zu Schrittverlusten kommt. In einem Open-Loop-System, bei dem keine Erkennung der Schrittverluste möglich ist, wäre es ab jetzt nicht mehr möglich, die Position des Schrittmotors zu erkennen. Je nach System kann das zu schweren Fehlern führen. Ein wichtiger Aspekt des Spannungssprunges ist jedoch zu beachten, der Einschaltstrom. Dadurch steigt der benötigte Primärstrom des Aufwärtswandlers für kurze Zeit auf das Vielfache des Wertes des Stroms im eingeschwungenen Zustand an. Dies kann, je nach Leistung des verwendeten Netzteils, dazu führen, dass die anliegende Versorgungsspannung einbricht und sich eventuell, auf Grund eines Unterspannungsschutzes, abschaltet. Dies zieht eine Reihe von Problemen nach sich. Durch den Spannungseinbruch kann es zu einem Abschalten der von dem Netzteil versorgten ICs kommen, dem sogenannten *Brown Out*, was ein Defekt der entsprechenden Schaltung zur Folge haben kann. Das Netzteil oder andere, den Strompuls leitende Komponenten, können durch den Strompuls beschädigt werden. Um dies zu verhindern, wurde ein Sanftanlauf entwickelt (siehe Kapitel 4.4), der diesen Strompuls reduziert. Für die Erzeugung des Spannungssprunges wurde ein Schrittssignal generiert, welches von  $0\text{ Hz}$  (keine Aufwärtswandlung) auf  $23\text{ kHz}$  (maximale Ausgangsspannung) springt. Als Last wurde nicht der Schrittmotortreiber zusammen mit einem Schrittmotor verwendet (siehe Kapitel 5.1.5), sondern ein  $100\ \Omega$  Leistungswiderstand. Um einen Einfluss der Strombegrenzung durch das Netzteil auszuschließen, wurde parallel zum Netzteil und dem zu testenden Modul ein  $10000\ \mu\text{F}$  Elektrolytkondensator parallel geschaltet, der den kurzzeitigen Strombedarf des Moduls versorgt. Um den Einfluss des verwendeten Ausgangskondensators zu ermitteln, wurden verschiedenen Bauarten mit unterschiedlichen Kapazitäten verwendet.

In Abbildung 5.9a) und b) sind die Auswirkungen der Sanftanlauf-Funktion auf einen Spannungssprung im Vergleich mit einem Spannungssprung ohne Sanftanlauf-Funktion für unterschiedliche Ausgangskondensatoren dargestellt. Festzustellen ist, dass unabhängig von dem verwendeten Ausgangskondensator, die Anstiegszeit bei Verwendung der Sanftanlauf-Funktion größer ist, als bei einem Spannungssprung

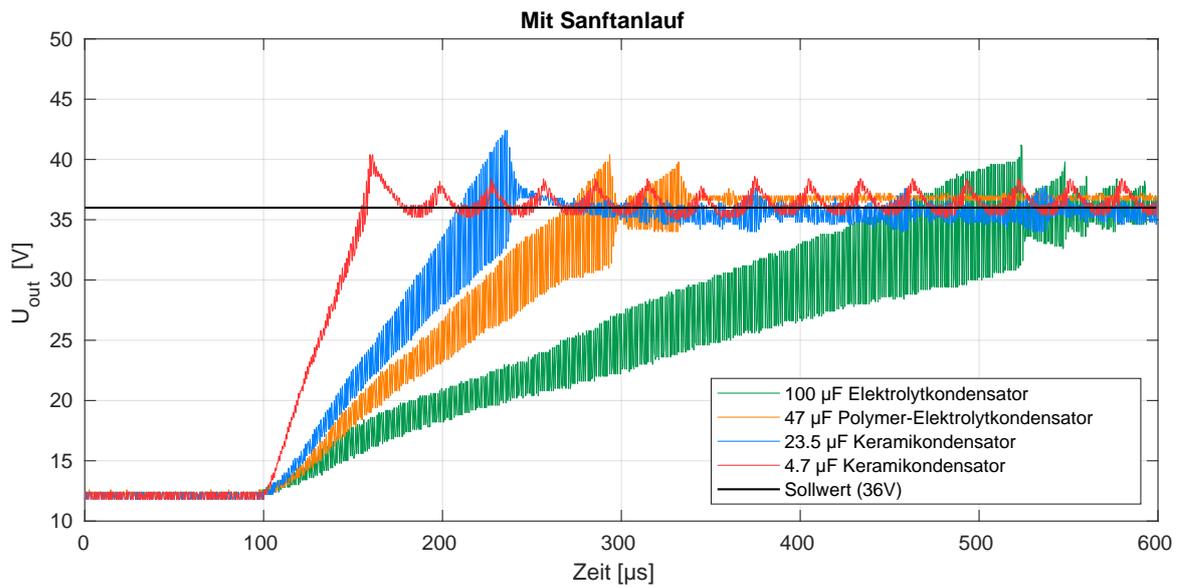
ohne Sanftanlauf. Im Folgenden ist aufgelistet, wie viel länger der Anstieg mit Sanftanlauf im Vergleich zum Anstieg ohne Sanftanlauf dauert:

- 100  $\mu F$  Elektrolytkondensator um 63.71 %
- 47  $\mu F$  Polymer-Elektrolytkondensator um 85.98 %
- 23.5  $\mu F$  Keramikkondensator um 187.76 %
- 4.7  $\mu F$  Keramikkondensator um 409.90 %

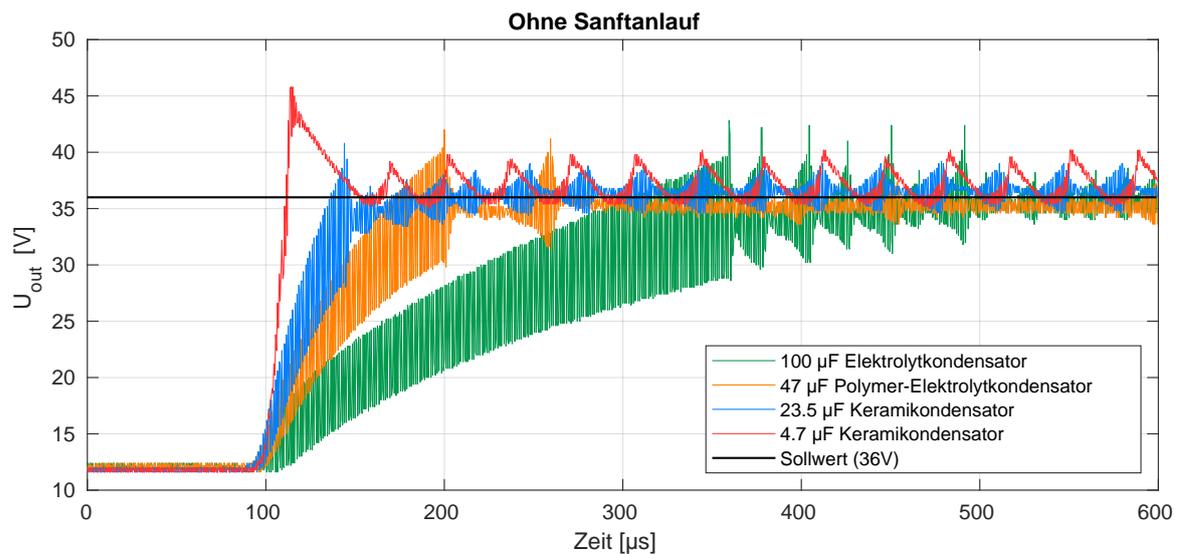
Weiterhin zeigt sich, dass die Sanftanlauf-Funktion auch unabhängig eines Sollwert-sprungs bei der Regelung auf einen konstanten Sollwert die Abweichungen der Ausgangsspannung von dem Sollwert reduziert. Dies zeigt sich gut in Abbildung 5.8. Allgemein ist festzustellen, dass je größer die Kapazität des Ausgangskondensators ist, desto länger ist die Anstiegszeit. In Abbildung 5.10 wird der Einschaltstrom der jeweiligen Spannungssprünge dargestellt. Der Vergleich von Abbildung 5.10a (mit Sanftanlauf) mit Abbildung 5.10b (ohne Sanftanlauf) zeigt deutlich, welchen Unterschied der Sanftanlauf macht. Die Stromspitzen konnten durch den Sanftanlauf wie folgt reduziert werden:

- 100  $\mu F$  Elektrolytkondensator um 54.45 %
- 47  $\mu F$  Polymer-Elektrolytkondensator um 52.84 %
- 23.5  $\mu F$  Keramikkondensator um 44.96 %
- 4.7  $\mu F$  Keramikkondensator um 58.28 %

Im Durchschnitt ermöglicht der Sanftanlauf in diesem Testlauf eine Reduktion der Stromspitzen des Einschaltstroms um 52.6 %.

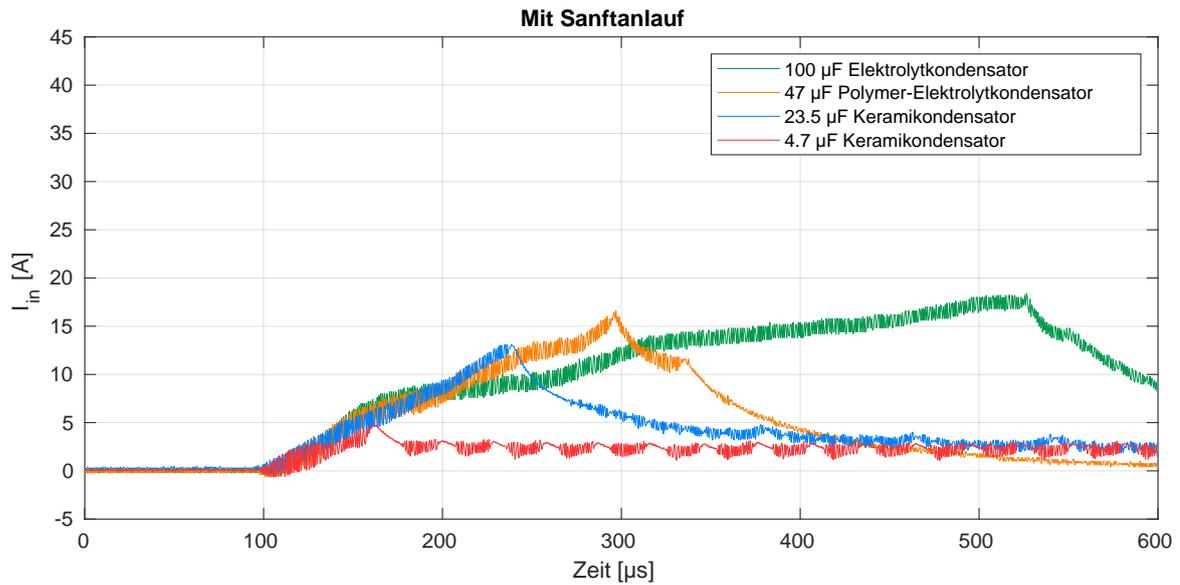


(a) Ausgangsspannung (Mit Sanftanlauf)

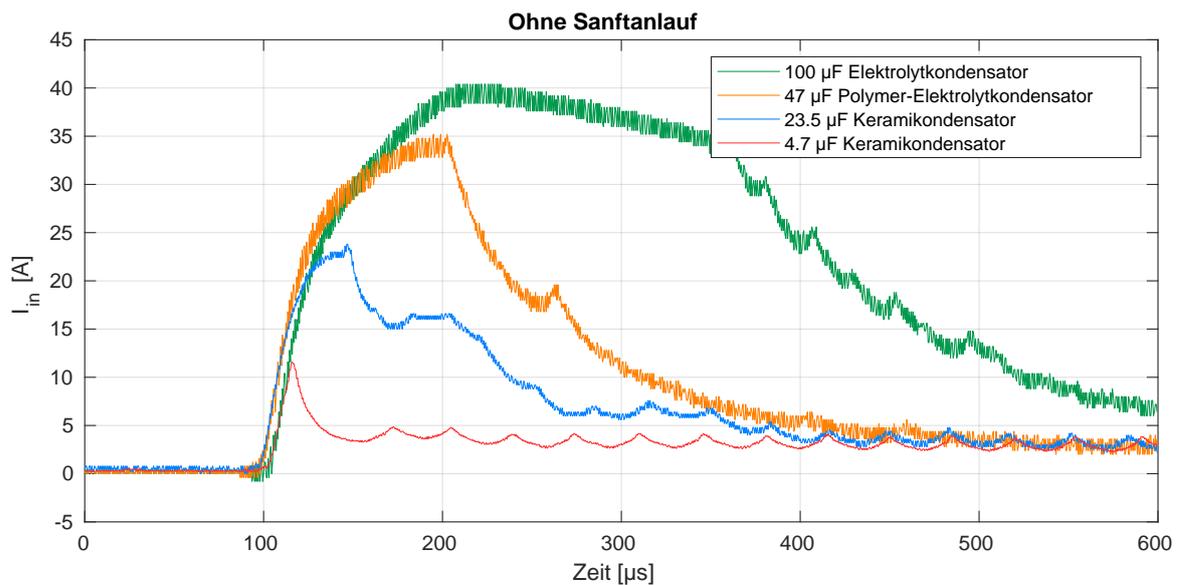


(b) Ausgangsspannung (Ohne Sanftanlauf)

**Abbildung 5.9** Verlauf eines Spannungssprungs von 12 V auf 36 V an einem 100  $\Omega$  Leistungswiderstand mit unterschiedlichen Ausgangskondensatoren. Verwendete Induktivität  $L = 6.8 \mu$ H.



(a) Eingangsstrom (Mit Sanftanlauf)



(b) Eingangsstrom (Ohne Sanftanlauf)

**Abbildung 5.10** Verlauf des Eingangsstrom bei einem Spannungssprung von 12V auf 36V an einem  $100\Omega$  Leistungswiderstand mit unterschiedlichen Ausgangskondensatoren. Verwendete Induktivität  $L = 6.8 \mu H$ .

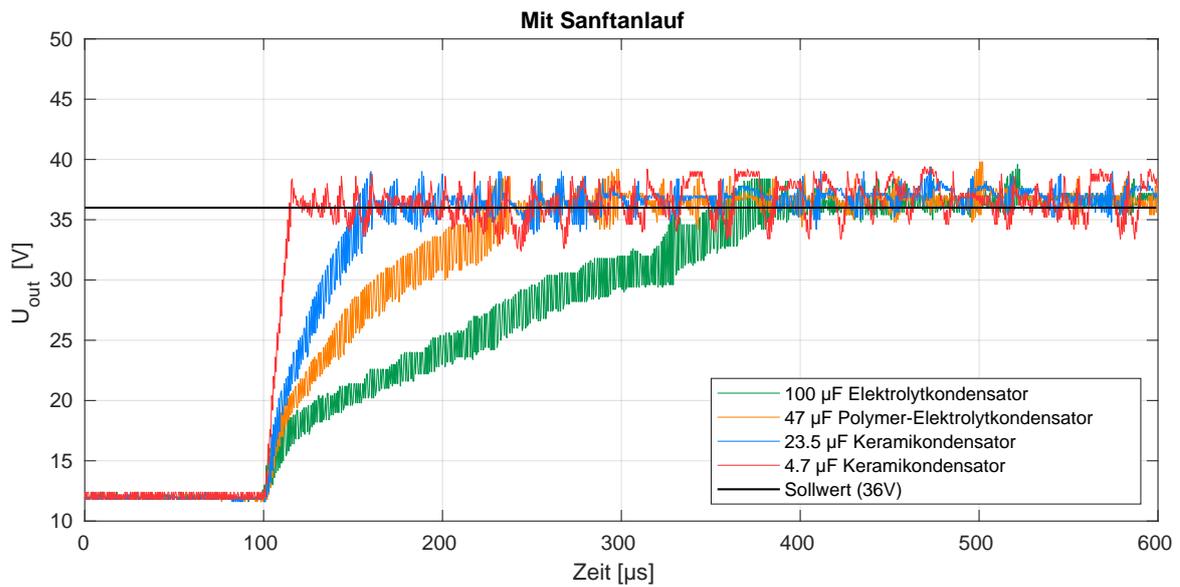
### 5.1.5 Spannungssprung an einem Schrittmotor

Nach dem Spannungssprung an einer resistiven Last wurden die gleichen Messungen mit dem, auf dem Modul implementierten, Schrittmotortreiber *TMC2130* und einem unbelasteten Schrittmotor (Typ *QSH4218-51* von TRINAMIC [43]) als Last durchgeführt. Dabei wurde der Schrittmotortreiber auf eine  $1/16$  Mikroschrittauflösung und *StealthChop* eingestellt. Der Sollstrom wurde auf  $0.87\text{ A}$  eingestellt. Zur Erzeugung des Spannungs- und Geschwindigkeitssprungs wurde ein Schrittsignal generiert, welches von  $0\text{ Hz}$  (keine Aufwärtswandlung und  $0\text{ min}^{-1}$ ) auf  $23\text{ kHz}$  (maximale Ausgangsspannung und  $431.25\text{ min}^{-1}$ ) springt. Ein Anwendungsfall hierfür ist der 3D-Druck, da trotz der Verwendung von Beschleunigungsrampen häufig ein Drehzahlsprung von  $0\text{ min}^{-1}$  auf eine festgelegte Drehzahl stattfindet, ab der aus beschleunigt wird. Dieser Drehzahlsprung wird auch als *Jerk* bezeichnet. Ein weiterer Grund für diese Messungen war, das Verhalten der Regelung der Ausgangsspannung bei einer dynamischen, nicht resistiven Last zu beobachten.

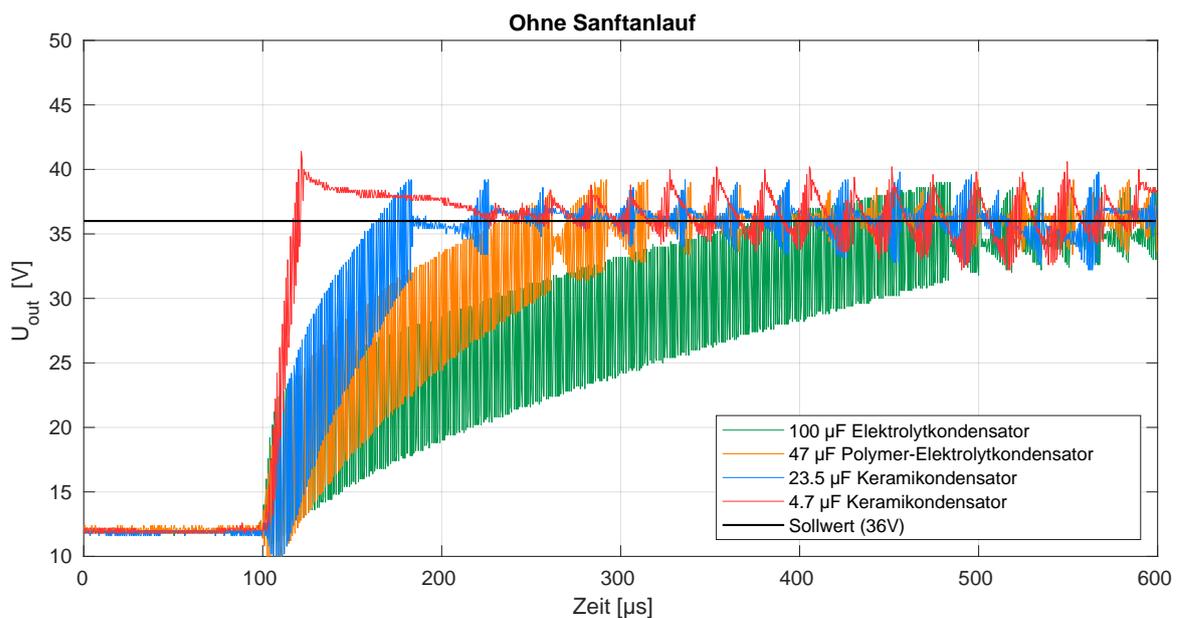
In Abbildung 5.11a) und b) sind die Auswirkungen der Sanftanlauf-Funktion auf einen Spannungssprung, im Vergleich mit einem Spannungssprung ohne Sanftanlauf-Funktion, für unterschiedliche Ausgangskondensatoren dargestellt. Im Gegensatz zu dem Spannungssprung an einer resistiven Last (siehe Abbildung 5.9), hat der Sanftanlauf, bei der Verwendung eines Schrittmotors als Last, einen geringeren Einfluss auf die Anstiegszeit der Ausgangsspannung. Bei allen Messungen ist die Anstiegszeit mit Sanftanlauf sogar kürzer, als die ohne. Im Folgenden ist aufgelistet, wie viel kürzer die Anstiegszeit mit Sanftanlauf im Vergleich zum Anstieg ohne Sanftanlauf ist:

- $100\ \mu\text{F}$  Elektrolytkondensator um  $10.68\%$
- $47\ \mu\text{F}$  Polymer-Elektrolytkondensator um  $11.39\%$
- $23.5\ \mu\text{F}$  Keramikkondensator um  $21.97\%$
- $4.7\ \mu\text{F}$  Keramikkondensator um  $14.71\%$

Ein weiterer Unterschied zu den Messungen an der resistiven Last ist, dass das Rauschen während des Spannungsanstiegs und zum Teil auch danach bei Verwendung des Sanftanlaufs geringer ist, zu sehen in Abbildung 5.11.



(a) Ausgangsspannung (Mit Sanftanlauf)



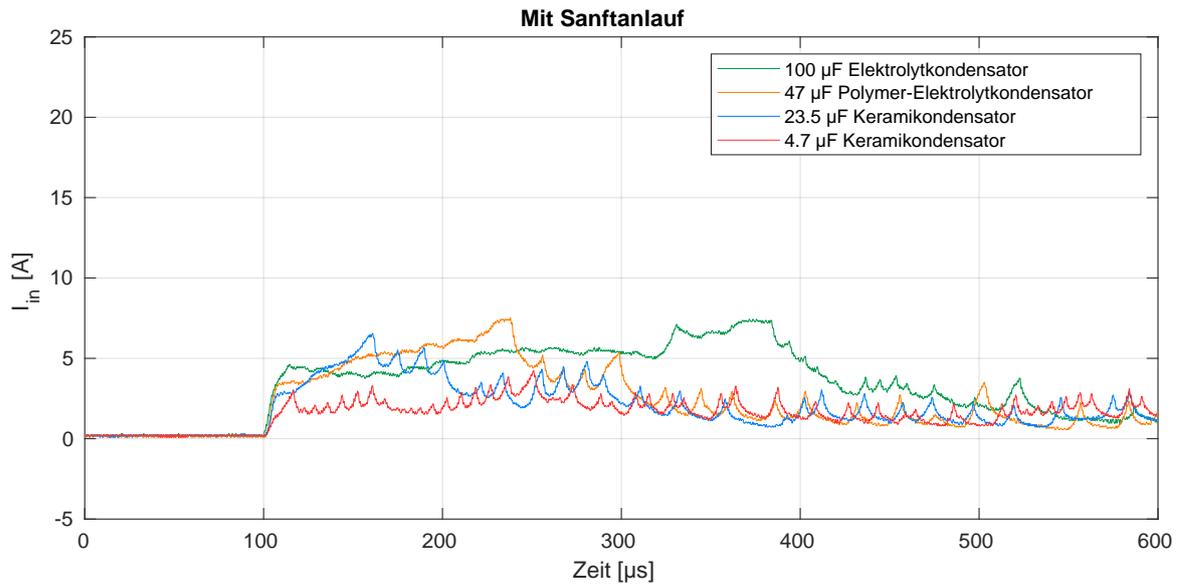
(b) Ausgangsspannung (Ohne Sanftanlauf)

**Abbildung 5.11** Verlauf der Ausgangsspannung bei einem Spannungssprung an einem Schrittmotor (QSH4218-51) mit unterschiedlichen Ausgangskondensatoren. Verwendete Induktivität  $L = 6.8 \mu\text{H}$ .

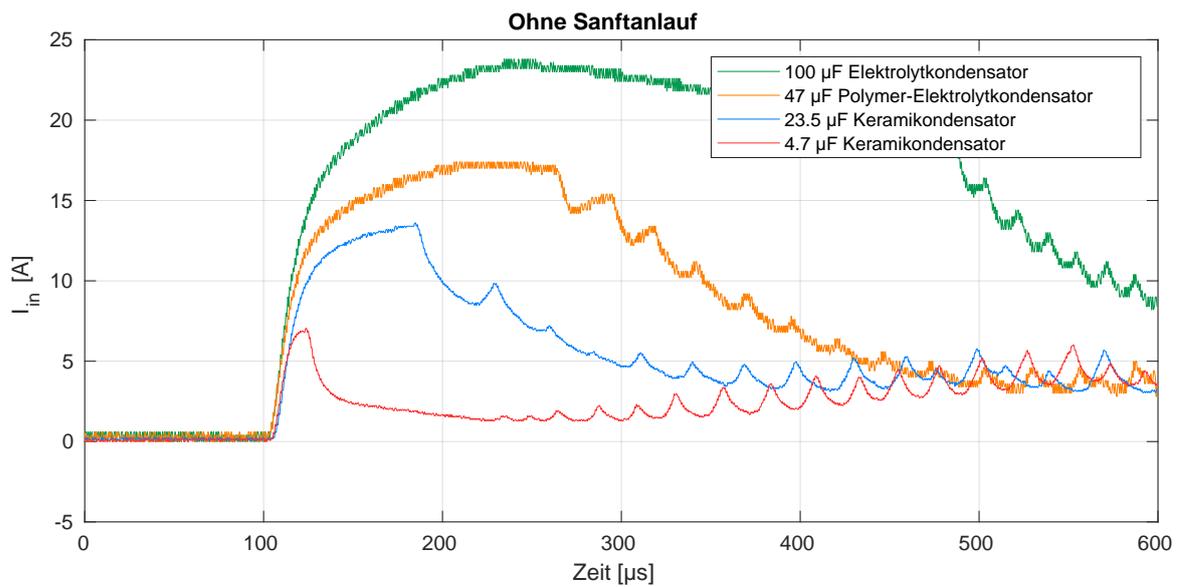
In Abbildung 5.12 wird der Einschaltstrom der jeweiligen Spannungssprünge dargestellt. Der Vergleich von Abbildung 5.12a (mit Sanftanlauf) mit Abbildung 5.12b (ohne Sanftanlauf) zeigt deutlich, welchen Unterschied der Sanftanlauf macht. Die Stromspitzen konnten durch den Sanftanlauf wie folgt reduziert werden:

- 100  $\mu F$  Elektrolytkondensator um 68.73 %
- 47  $\mu F$  Polymer-Elektrolytkondensator um 56.78 %
- 23.5  $\mu F$  Keramikkondensator um 51.76 %
- 4.7  $\mu F$  Keramikkondensator um 39.77 %

Im Durchschnitt ermöglicht der Sanftanlauf in diesem Testlauf eine Reduktion der Stromspitzen des Einschaltstroms um 54.26 %.



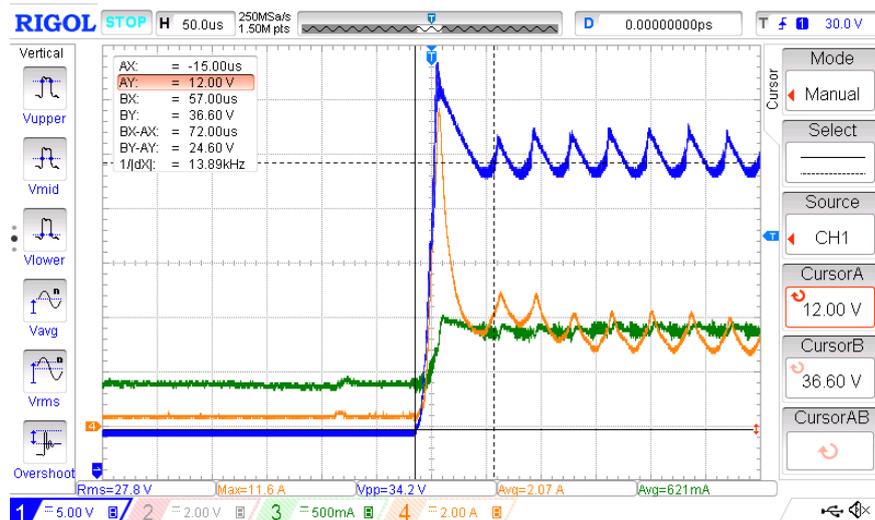
(a) Eingangsstrom (Mit Sanftanlauf)



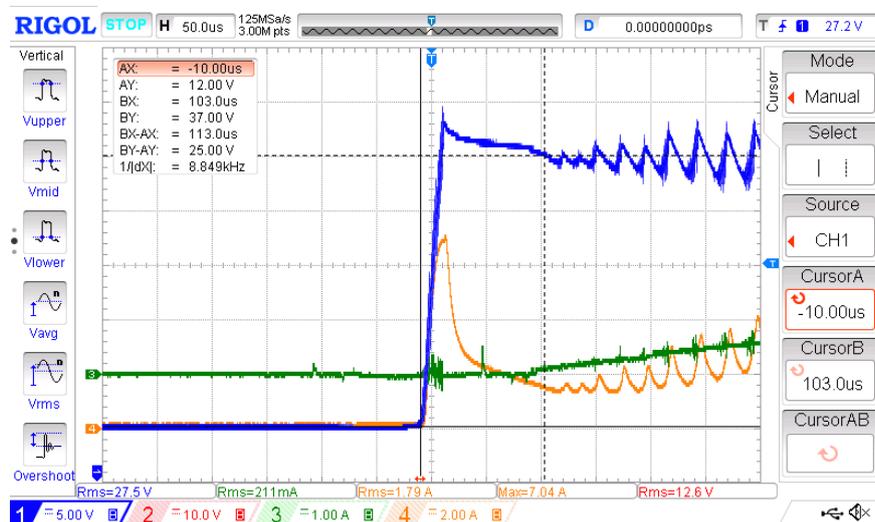
(b) Eingangsstrom (Ohne Sanftanlauf)

**Abbildung 5.12** Verlauf des Eingangsstroms bei einem Spannungssprung an einem Schrittmotor (QSH4218-51) mit unterschiedlichen Ausgangskondensatoren. Verwendete Induktivität  $L = 6.8 \mu\text{H}$ .

Ein zu beachtender Unterschied bezüglich der Stromentnahme ist, dass der Motorstrom durch den Schrittmotortreiber geregelt wird, der Strom durch den Leistungswiderstand jedoch nur durch den Widerstand begrenzt wird. Der Strom am Widerstand steigt dadurch nicht langsam an. Das hat Auswirkungen auf den benötigten Ausgangsstrom, den der Aufwärtswandler bereitstellen muss. In Abbildung 5.13 ist der Unterschied des Ausgangsstroms während des Spannungssprungs zu erkennen.



(a) Spannungssprung an resistiver Last



(b) Spannungssprung an stromgeregeltem Schrittmotor

**Abbildung 5.13** Vergleich des Anstiegs des Ausgangsstroms (grüner Kanal) bei verschiedenen Lasten. Der blaue Kanal ist die Ausgangsspannung, der orange Kanal ist der Eingangsstrom des Aufwärtswandlers und der grüne Kanal ist der Ausgangsstrom. Es ist deutlich die aktive Stromregelung des Motortreibers in b) zu erkennen, die den Motorstrom langsam steigert. Im Gegensatz dazu ist in a) ein Sprung des Ausgangsstroms an der unregulierten resistiven Last zu erkennen.

## 5.2 Verhalten des Moduls auf dem Motorteststand

Da die Kernidee dieser Arbeit ist, das maximale Drehmoment eines Schrittmotors bei gleicher Versorgungsspannung über einen größeren Drehzahlbereich aufrechtzuerhalten, wurden die Drehmomentkurven von unterschiedlichen Schrittmotoren bei verschiedenen Betriebsarten und Versorgungsspannungen aufgezeichnet. Dafür wurde ein von TRINAMIC entwickelter Motorteststand verwendet, der die eingestellten Messpunkte automatisch abarbeitet (siehe Kapitel 2.1.6). Um die, durch die entwickelte dynamische Spannungsanpassung, erzeugte Drehmomentkurve einordnen zu können, wurden mit den selben Schrittmotortreiber Einstellungen die Messungen mit der Eingangsspannung des Aufwärtswandlers (12 V) und der maximal erzeugbaren Ausgangsspannung des Aufwärtswandlers (36 V) wiederholt. Dabei wurde die Versorgungsspannung durch ein Labornetzteil bereitgestellt. Es lässt sich erkennen, wie ähnlich die durch die dynamische Spannungsanpassung generierten Drehmomentkurven den Drehmomentkurven bei einer Versorgung mit 36 V aus einem Netzteil sind. Als weitere Versorgungsoption wurde mithilfe des entwickelten Moduls konstant 36 V erzeugt, unabhängig davon wie schnell sich der Schrittmotor dreht. In der Theorie sollte bei Verwendung des Aufwärtswandlers die gleichen Drehmomentverläufe erzeugen werden, wie bei Verwendung des Labornetzteils.

Als Schrittmotortreiber wurde, je nach Spannungsversorgung, entweder ein voll bestücktes Modul verwendet oder eines, bei dem nur die Komponenten des Schrittmotortreibers bestückt waren. Dies soll verhindern, dass unterschiedliche Platinen die Verlustleistung des TMC2130 verschieden gut ableiten können. Während der gesamten Testläufe wurden die verwendeten Module aktiv durch einen Lüfter gekühlt. Der Sollstrom beträgt bei allen Test jeweils 0.87 A, die verwendete Mikroschrittauflösung beträgt 1/16 und die chipinterne Interpolation der Mikroschritte interpoliert die 1/16-Schritte auf 1/256-Schritte.

Insgesamt wurden drei verschiedene Schrittmotoren vermessen, wobei die Messungen jeweils in dem *SpreadCycle*-Modus und dem *StealthChop*-Modus durchgeführt wurden. Diese beiden Betriebsmodi kommen in 3D-Druckern mit TRINAMIC-Schrittmotortreibern zum Einsatz und sind deshalb von Interesse. Alle drei verwendeten Motoren sind bipolare Hybridschrittmotoren mit zwei Phasen und einem mechanischen Schrittwinkel von  $1.8^\circ$ .

Um eine einfachere Beschreibung der ermittelten Messkurven zu ermöglichen wird im Folgenden auf die hier eingeführten Abkürzungen zurückgegriffen:

- 12V-NT: Versorgung des Motortreibers mit 12 V aus einem Labornetzteil
- 36V-NT: Versorgung des Motortreibers mit 36 V aus einem Labornetzteil
- 36V-DCDC: Versorgung des Motortreibers mit 36 V die durch den Aufwärtswandler erzeugt wurden
- $U_{VAR}$ : Versorgung des Motortreibers mit einer variablen, von der Drehzahl abhängigen, Spannung die durch den Aufwärtswandler generiert wurde

### 5.2.1 Betriebsart: SpreadCycle

Die von TRINAMIC entwickelte Betriebsart *SpreadCycle* ist eine Weiterentwicklung der klassischen Strom-Chopper-Regelung (siehe Kapitel 2.1.3). Sie beinhaltet eine Verbesserung der Motorstromregelung [45]. Die Schrittmotoren erzeugen mit dieser Betriebsart Geräusche, je nach Drehzahl.

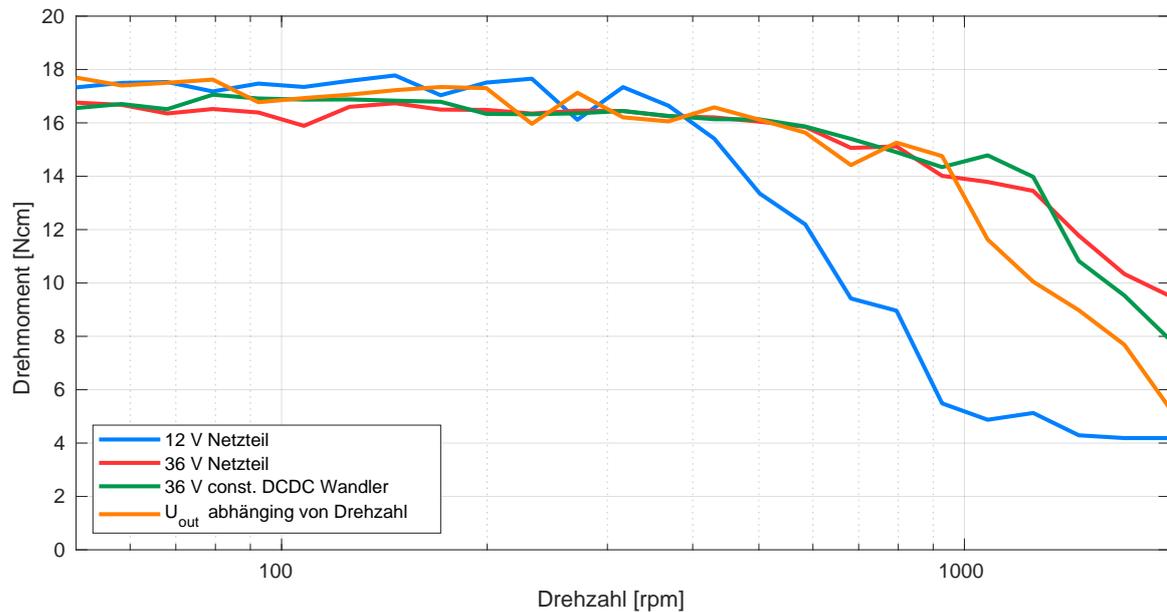
#### MOONS MS17HD4P4150

Der *MS17HD4P4150* [30] von *Moons' Industries* wurde für die Messungen ausgewählt, da er ein gutes Beispiel für einen Schrittmotor ist, wie er in vielen 3D-Druckern verbaut wird. Er hat die Baugröße NEMA17 und sein vom Hersteller angegebenes Haltemoment beträgt  $0.32 \text{ Nm}$  bei einem Motorstrom von  $1.5 \text{ A}$ .

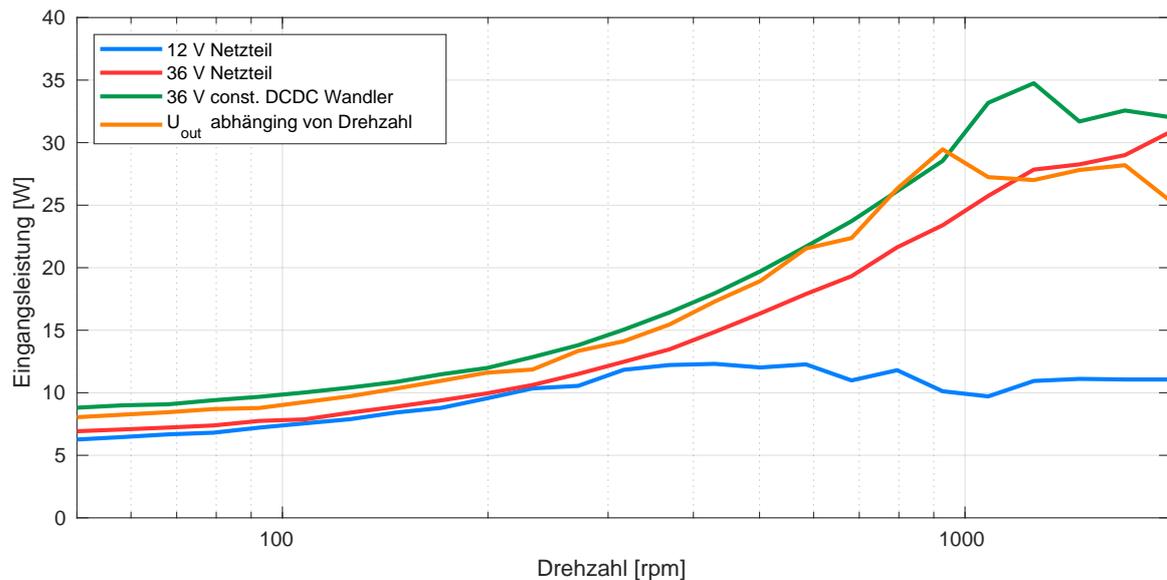
Die Messergebnisse sind in Abbildung 5.14 dargestellt, dabei ist in Abbildung 5.14a der Drehmomentverlauf in Abhängigkeit von der Drehzahl für die verschiedenen Versorgungsspannungen dargestellt und in Abbildung 5.14b die dafür benötigte Eingangsleistung in Abhängigkeit von der Drehzahl.

Es ist gut zu erkennen, dass bis  $300 \text{ min}^{-1}$  das Drehmoment unabhängig von der Spannung konstant um  $17 \text{ Nm}$  liegt. Ab  $300 \text{ min}^{-1}$  sinkt das Drehmoment bei der 12 V-NT-Kurve deutlich ab. Das Drehmoment bei den anderen Spannungen sinkt ab  $300 \text{ min}^{-1}$  leicht. Ab  $1000 \text{ min}^{-1}$  knickt das Drehmoment der  $U_{VAR}$ -Kurve ab. Die Kurven von 36V-NT und 36V-DCDC knicken bei  $1000 \text{ min}^{-1}$  nicht so deutlich ein, sinken aber auch stärker.

Die Abbildung 5.14b zeigt die für die Drehmomentverläufe in Abbildung 5.14a benötigten Eingangsleistungen. Die 12V-NT-Kurve ähnelt der 36V-NT-Kurve bis  $300 \text{ min}^{-1}$ , ab der das Drehmoment der 12V-NT-Kurve abfällt. Die anderen drei Leistungsaufnahmen steigen mit zunehmender Drehzahl an. Dabei benötigt die 36V-DCDC-Kurve aufgrund des Aufwärtswandlers durchgehend mehr Eingangsleistung als die 36V-NT-Kurve. Diese Differenz beginnt mit  $3 \text{ W}$  bei  $50 \text{ min}^{-1}$  und steigt auf  $7 \text{ W}$  bei  $1000 \text{ min}^{-1}$ . Es zeigt sich, dass trotz eines sehr ähnlichen Drehmomentverlaufs, die Verwendung eines Labornetzteils bei  $36 \text{ V}$  weniger Eingangsleistung benötigt, als die konstante Erzeugung von  $36 \text{ V}$  mithilfe des Aufwärtswandlers. Der Energiebedarf bei der dynamischen Spannungserzeugung deckt sich bis  $900 \text{ min}^{-1}$  größtenteils mit dem Bedarf der 36V-DCDC-Kurve.



(a) Drehmoment abhängig von der Drehzahl



(b) Für a) benötigte Eingangsleistung, abhängig von der Drehzahl

**Abbildung 5.14** Messergebnisse des Motors *MS17HD4P4150* von *Moons' Industries* im Betriebsmodus *SpreadCycle*. In a) ist zu erkennen, dass die 12V-NT-Kurve im Vergleich zu den anderen Kurven früher einbricht. Die drei anderen Kurven sind recht nahe beieinander, die  $U_{VAR}$ -Kurve bricht ab  $1000 \text{ min}^{-1}$  etwas mehr ein als 36V-NT und 36V-DCDC. Bei der Leistungsaufnahme in b) zeigt sich die zusätzliche Verlustleistung im Vergleich mit der 36V-NT-Kurve, die der Spannungswandler zusätzlich verbraucht.

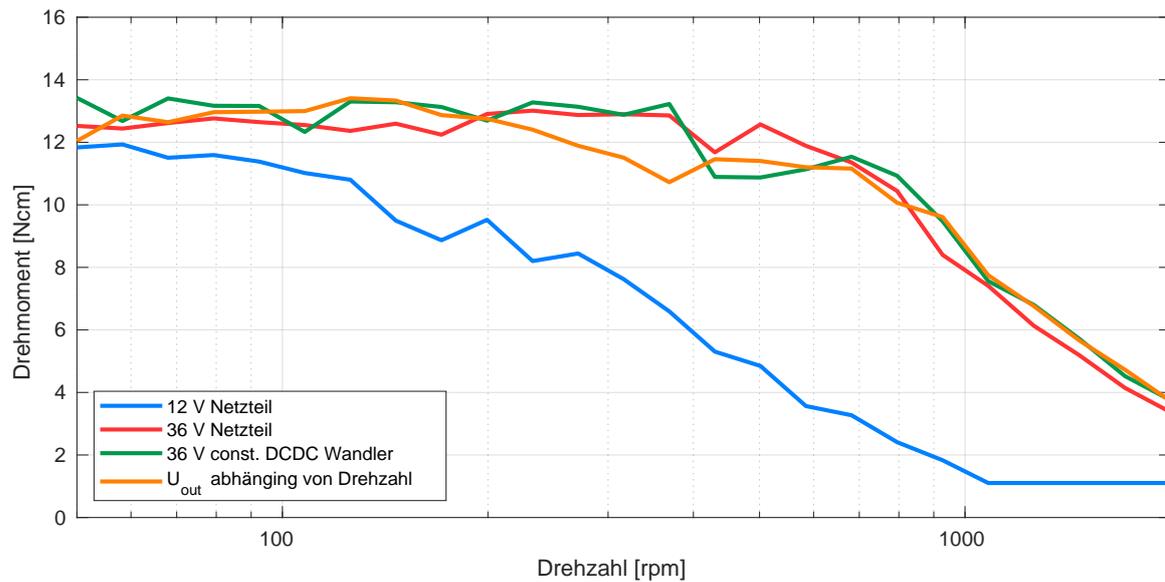
**TRINAMIC QSH2818-51**

Der *QSH2818-51* von TRINAMIC ist mit seiner NEMA11-Baugröße und seinem Haltemoment von  $0.12\text{ Nm}$  einer der kleinsten Motoren die von TRINAMIC vertrieben werden. Er wurde ausgewählt, da solch kleine Schrittmotoren auch in Anwendungen mit wenig Platz Verwendung finden. Gerade in solchen Anwendungen steht zumeist keine höhere Versorgungsspannung zur Verfügung, sondern typischerweise  $5\text{ V}$  oder  $12\text{ V}$ . Dafür wäre eine kompakte Lösung interessant, die aus der vorhandenen niedrigen Versorgungsspannung und dem kleinen Schrittmotor das maximale Drehmoment herausholen kann.

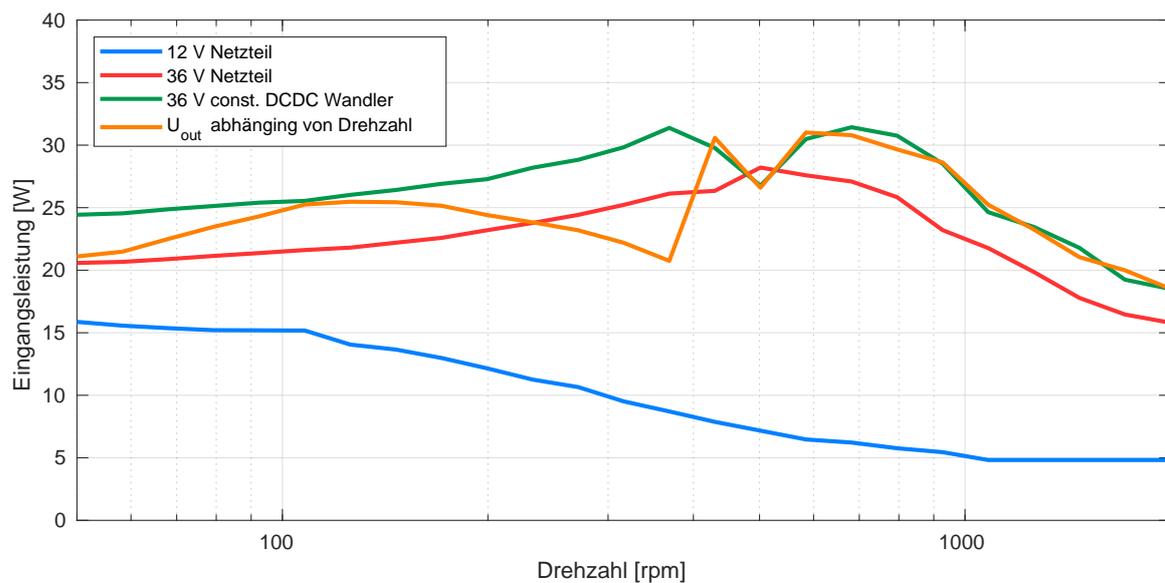
Die Messergebnisse sind in Abbildung 5.15 dargestellt. Dabei ist in Abbildung 5.15a der Drehmomentverlauf in Abhängigkeit von der Drehzahl für die verschiedenen Versorgungsspannungen dargestellt und in Abbildung 5.15b die dafür benötigte Eingangsleistung in Abhängigkeit von der Drehzahl.

Es zeigt sich (siehe Abbildung 5.15a), dass das Drehmoment der 12V-NT-Kurve direkt ab  $50\text{ min}^{-1}$  niedriger ist, als die restlichen drei Drehmomentverläufe. Die Drehmomentkurve von 36V-NT fällt ab ungefähr  $400\text{ min}^{-1}$  langsam ab. Deswegen wurde für die dynamische Spannungsanpassung ein Bereich von  $40\text{ min}^{-1}$  bis  $400\text{ min}^{-1}$  gewählt. Bis  $200\text{ min}^{-1}$  sind die drei Drehmomentkurven, abgesehen von 12V-NT, sehr ähnlich, danach bricht die Kurve von  $U_{VAR}$  etwas ein, bis sie ab  $700\text{ min}^{-1}$  wieder einen ähnlichen Verlauf wie die 36V-NT- und 36V-DCDC-Kurve hat. Die 36V-DCDC-Drehmomentkurve bricht ab  $400\text{ min}^{-1}$  ein wenig stärker ein als die 36V-NT-Kurve, ist ihr aber sonst sehr ähnlich.

Die benötigte Eingangsleistung in Abbildung 5.15b zeigt wieder deutlich die Differenz zwischen 36V-NT und 36V-DCDC, was auf die Verluste des Aufwärtswandler zurückzuführen ist. Sie bewegt sich beinahe durchgehend zwischen  $5\text{ W}$  und  $6\text{ W}$ , mit einem Einbruch bei  $400\text{ min}^{-1}$ . Bei dieser Drehzahl ist auch das Drehmoment von 36V-DCDC niedriger als von 36V-NT. Die Eingangsleistung der  $U_{VAR}$ -Kurve bewegt sich bis  $200\text{ min}^{-1}$  zwischen der 36V-DCDC- und 36V-NT-Kurve und sinkt danach bis  $368\text{ min}^{-1}$  ab. Ab  $430\text{ min}^{-1}$  ähnelt die  $U_{VAR}$ -Kurve sehr der 36V-DCDC-Kurve.



(a) Drehmoment abhängig von der Drehzahl



(b) Für a) benötigte Eingangsleistung, abhängig von der Drehzahl

**Abbildung 5.15** Messergebnisse des Motors *QSH2818-51* von TRINAMIC im Betriebsmodus *Spread-Cycle*. Die Drehmomentkurven in a) sind mit Ausnahme von der 12V-NT-Kurve sehr nahe beieinander, die  $U_{VAR}$ -Kurve bricht bei  $368 \text{ min}^{-1}$  etwas ein. Die Leistungsaufnahme in b) zeigt wieder die deutliche Verlustleistung des Spannungswandlers, wenn 36V-NT und 36V-DCDC verglichen werden. Zu erkennen ist, dass die  $U_{VAR}$ -Kurve bis  $400 \text{ min}^{-1}$  eine niedrigere Leistungsaufnahme hat, als die 36V-DCDC-Kurve, teilweise sogar niedriger, als die 36V-NT-Kurve. Ab  $400 \text{ min}^{-1}$  sind die Leistungsaufnahmen von 36V-DCDC und  $U_{VAR}$  beinahe identisch. Bei  $500 \text{ min}^{-1}$  bricht die Leistungsaufnahme von 36V-DCDC und  $U_{VAR}$  kurz ein.

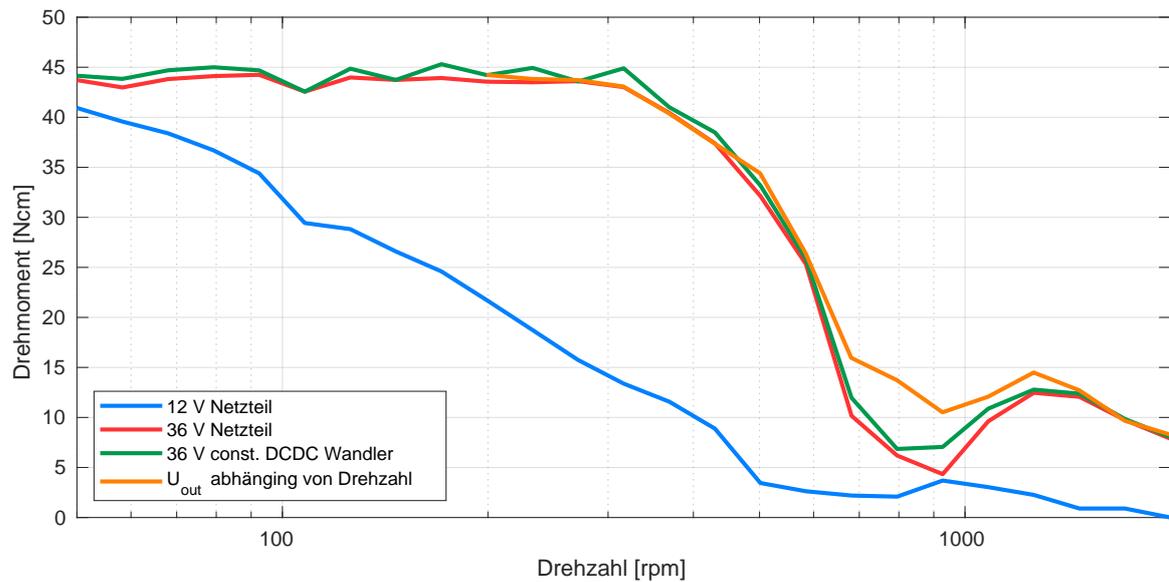
**TRINAMIC QSH4218-51**

Der *QSH4218-51* ist der meistverkaufte Schrittmotor von TRINAMIC. Er hat mit einem Haltemoment von  $0.49 \text{ Nm}$  bei einem Motorstrom von  $1 \text{ A}$  für seine NEMA17-Baugröße ein hohes Haltemoment.

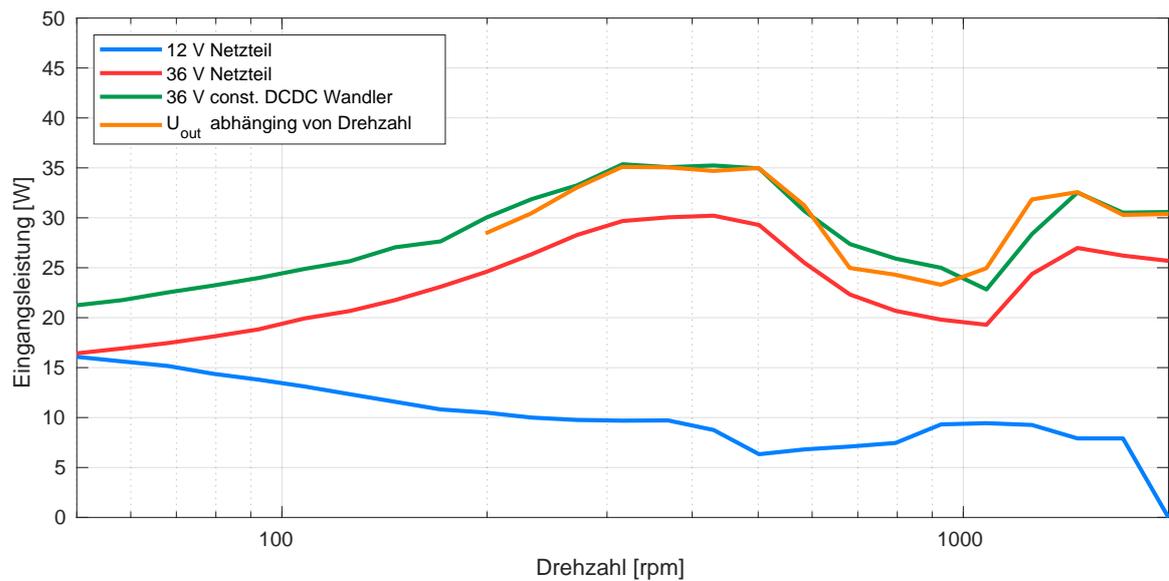
In dem Drehmomentverlauf in Abbildung 5.16a sieht man deutlich, dass die Verwendung des entwickelten Moduls eine Verbesserung des Drehmomentverlaufs bei höheren Drehzahlen ermöglicht.

Die Drehmomentkurven für 36V-NT, 36V-DCDC und  $U_{VAR}$  sind bis  $600 \text{ min}^{-1}$  beinahe identisch, von  $600 \text{ min}^{-1}$  bis  $1300 \text{ min}^{-1}$  ist das Drehmoment von  $U_{VAR}$  ein wenig höher, als von 36V-NT und 36V-DCDC, welche recht ähnlich zueinander sind. Danach, bis zum Ende der Messung bei  $2000 \text{ min}^{-1}$ , sind die Verläufe wieder sehr ähnlich. Der Verlauf der 12V-NT-Kurve liegt während des gesamten Drehzahlbereichs unter den anderen Kurven, sie bricht direkt von Beginn bei  $50 \text{ min}^{-1}$  ein.

Die Leistungsaufnahme über den Drehzahlbereich ist in Abbildung 5.16b dargestellt. Wie bei den Leistungsaufnahmen der anderen Motoren, ist hier eine konstante Differenz zwischen der Leistungsaufnahme von der 36V-NT-Kurve und den Kurven des Spannungswandlers zu erkennen. Das Modul mit dem Aufwärtswandler benötigt zur Erzeugung des gleichen Drehmoments recht konstant  $5 \text{ W}$  mehr als bei Verwendung des Netzteils mit  $36 \text{ V}$ . Dabei ist die Leistungsaufnahme bei Verwendung des entwickelten Moduls im konstanten  $36 \text{ V}$  und im  $U_{VAR}$ -Modus sehr ähnlich zueinander.



(a) Drehmoment abhängig von der Drehzahl



(b) Für a) benötigte Eingangsleistung, abhängig von der Drehzahl

**Abbildung 5.16** Messergebnisse des Motors *QSH4218-51* von TRINAMIC im Betriebsmodus *SpreadCycle*. Die Drehmomentkurven in a) sind, mit Ausnahme der 12V-NT-Kurve, sehr dicht beieinander. Bei der Leistungsaufnahme in b) zeigt sich die zusätzliche Leistungsaufnahme des Aufwärtswandlers von 5 - 6 W im Vergleich von 36V-DCDC zu 36V-NT.

### 5.2.2 Betriebsart: StealthChop

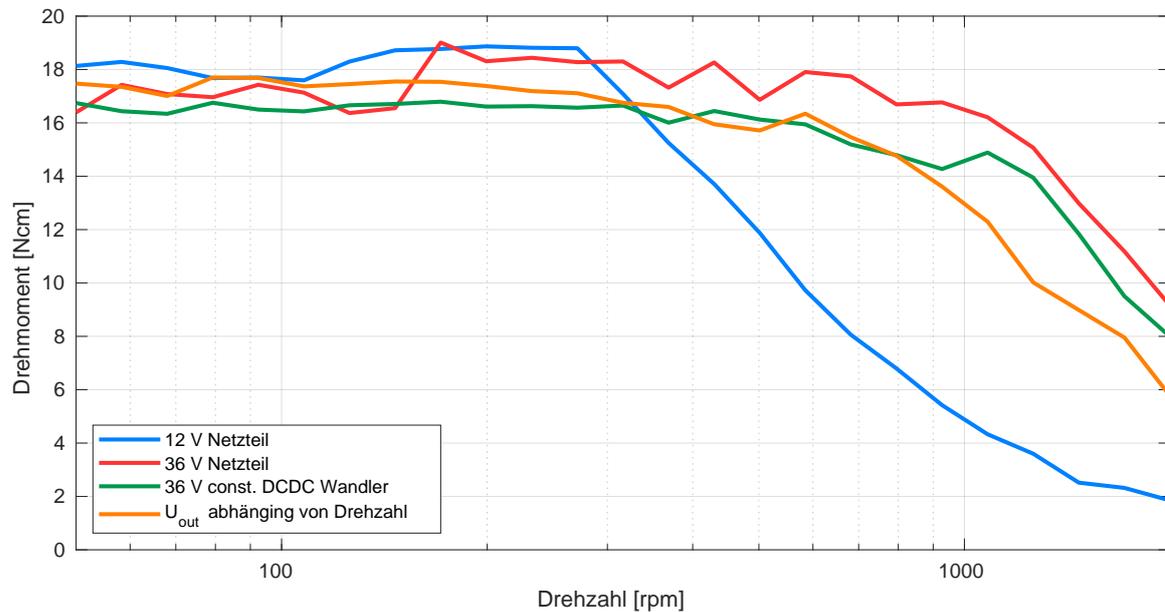
Die von TRINAMIC entwickelte Betriebsart *StealthChop* verwendet anstelle der herkömmlichen Strom-Chopper-Steuerung eine Spannungs-Chopper-Steuerung. Dadurch werden Motorgeräusche reduziert, die aufgrund von Schwingungen in der Regelfrequenz des Strom-Choppers entstehen würden [46]. Diese Betriebsart wird häufig in 3D-Druckern verwendet, da diese dadurch wesentlich weniger Betriebsgeräusche verursachen, wodurch der 3D-Drucker auch in Wohnräumen genutzt werden kann, ohne zu stören.

#### MOONS MS17HD4P4150

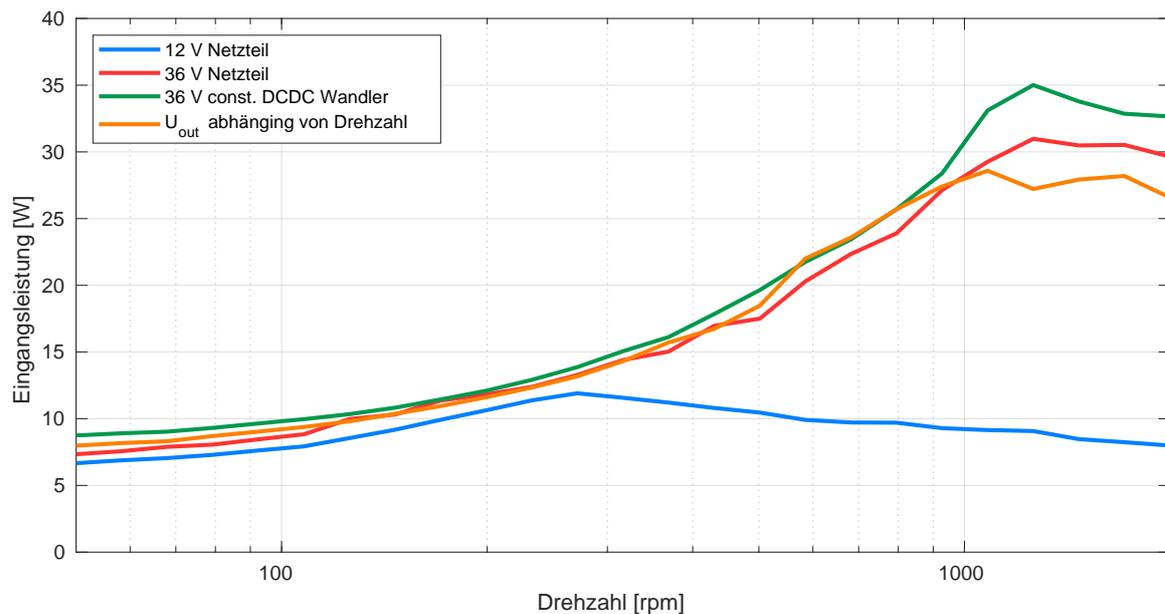
Die Messergebnisse sind in Abbildung 5.17 dargestellt, dabei ist in Abbildung 5.17a der Drehmomentverlauf in Abhängigkeit von der Drehzahl für die verschiedenen Versorgungsspannungen dargestellt und in Abbildung 5.17b die dafür benötigte Eingangsleistung in Abhängigkeit von der Drehzahl.

In Abbildung 5.17a bricht die 12V-NT-Drehmomentkurve ab  $171 \text{ min}^{-1}$  stark ein. Die 36V-NT-Kurve bricht ab  $600 \text{ min}^{-1}$  ein. Deswegen wurde der Drehzahlbereich für die dynamische Spannungsanpassung auf  $100 \text{ min}^{-1}$  bis  $600 \text{ min}^{-1}$  festgelegt. Die Drehmomentkurve von  $U_{VAR}$  ähnelt der 36V-DCDC-Kurve bis  $800 \text{ min}^{-1}$ , danach sinkt sie ab. Die 36V-DCDC-Kurve bricht ab  $1050 \text{ min}^{-1}$  ein.

Die Eingangsleistung (siehe Abbildung 5.17b) der 12V-NT-Kurve sinkt, wie das Drehmoment, ab  $171 \text{ min}^{-1}$  ab. Die restlichen drei Eingangsleistungskurven sind bis  $1000 \text{ min}^{-1}$  vergleichbar. Darüber benötigt die 36V-DCDC-Kurve am meisten Eingangsleistung, gefolgt von der 36V-NT- und  $U_{VAR}$ -Kurve.



(a) Drehmoment abhängig von der Drehzahl



(b) Für a) benötigte Eingangsleistung, abhängig von der Drehzahl

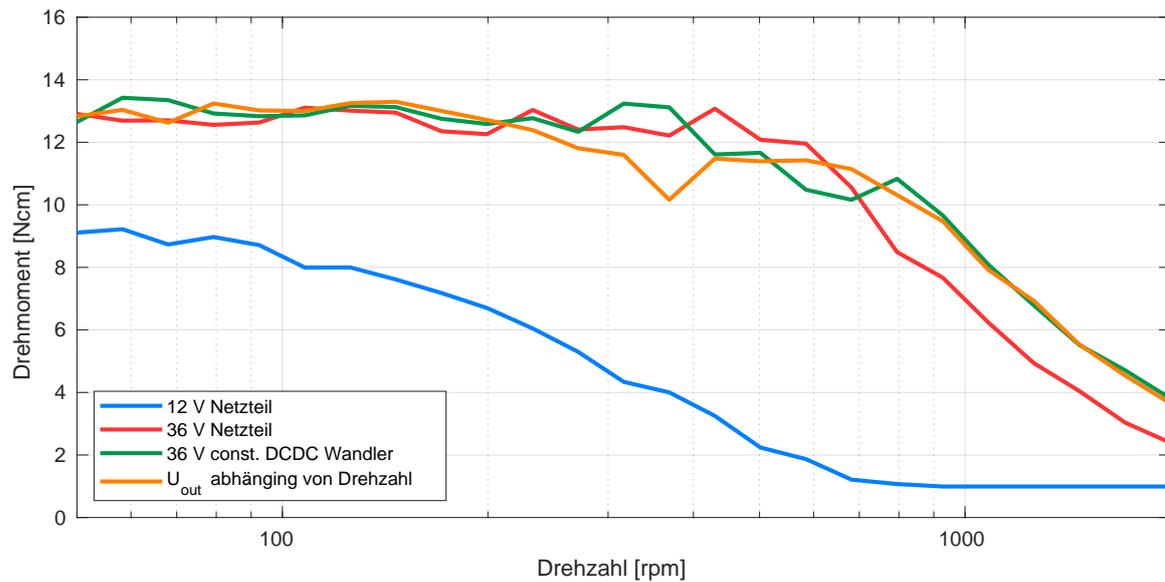
**Abbildung 5.17** Messergebnisse des Motors *MS17HD4P4150* von *Moons' Industries* im Betriebsmodus *StealthChop*. In a) ist zu erkennen, dass die 12V-NT-Kurve im Vergleich zu den anderen Kurven früher einbricht. Die beiden Spannungswandler-Kurven sind recht nahe beieinander, die 36V-DCDC-Kurve kann ab  $1000 \text{ min}^{-1}$  das Drehmoment etwas besser aufrechterhalten, als die  $U_{VAR}$ -Kurve. Bei der Leistungsaufnahme in b) zeigt sich die zusätzliche Verlustleistung im Vergleich mit der 36V-NT-Kurve, die der Spannungswandler zusätzlich verbraucht, etwas weniger als bei der *SpreadCycle*-Messung.

**TRINAMIC QSH2818-51**

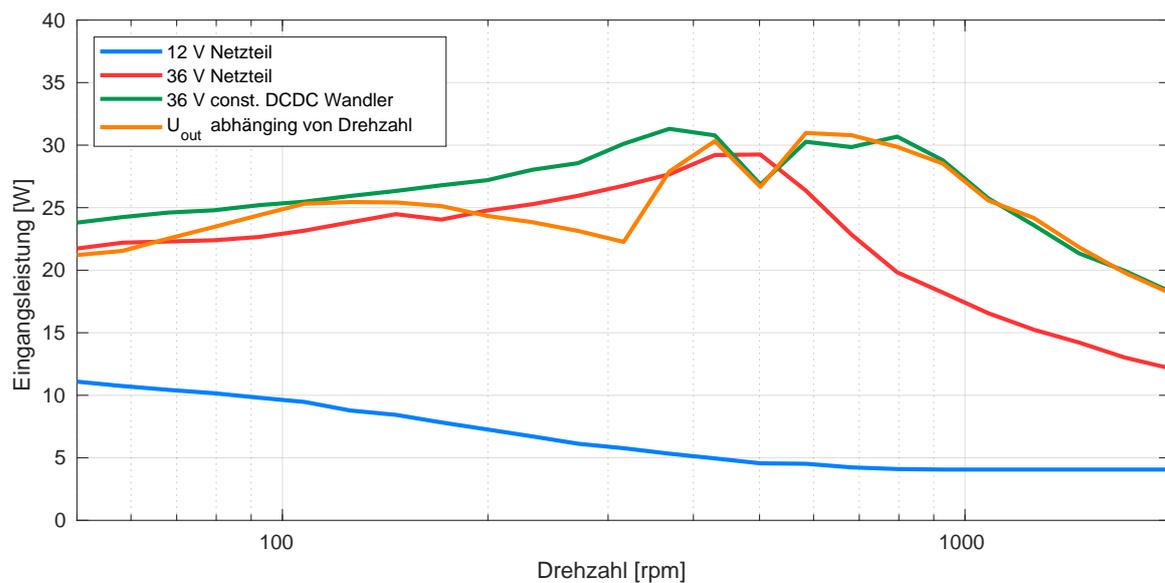
Die Messergebnisse sind in Abbildung 5.18 dargestellt, dabei ist in Abbildung 5.18a der Drehmomentverlauf in Abhängigkeit von der Drehzahl für die verschiedenen Versorgungsspannungen dargestellt und in Abbildung 5.18b die dafür benötigte Eingangsleistung in Abhängigkeit von der Drehzahl.

Die 12V-NT-Drehmomentkurve beginnt bei  $50 \text{ min}^{-1}$  mit nur  $9 \text{ Nm}$ , bei Verwendung des SpreadCycle Betriebsmodus in Abbildung 5.15a lag die 12V-NT-Kurve bei  $50 \text{ min}^{-1}$  noch bei  $12 \text{ Nm}$ . Die 36V-NT-Kurve bricht ab  $500 \text{ min}^{-1}$  ein, weswegen für die dynamische Spannungsanpassung ein Drehzahlbereich von  $40 \text{ min}^{-1}$  bis  $500 \text{ min}^{-1}$  gewählt wurde. Der Drehmomentverlauf bei  $U_{VAR}$  verhält sich zum Großteil wie die 36V-DCDC-Kurve, mit einem Einbruch des Drehmoments bei  $365 \text{ min}^{-1}$ .

Die Eingangsleistung, dargestellt in Abbildung 5.18b, ist relativ ähnlich zu der Eingangsleistung bei der SpreadCycle Betriebsart (siehe Abbildung 5.15b), inklusive des Einbruchs der 36V-DCDC-Kurve bei  $500 \text{ min}^{-1}$ .



(a) Drehmoment abhängig von der Drehzahl



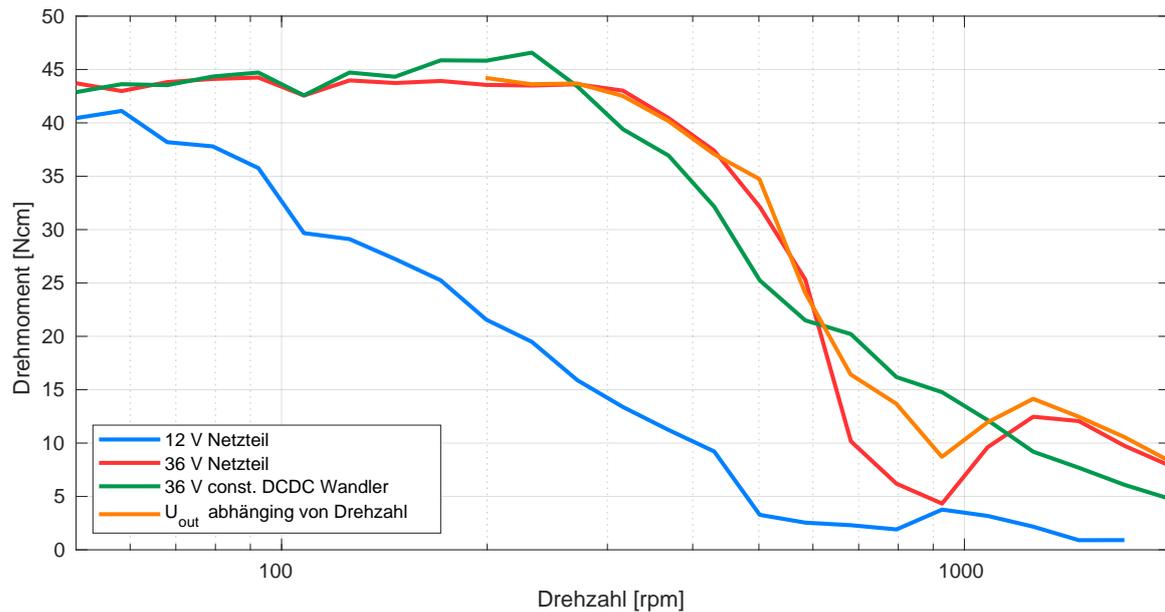
(b) Für a) benötigte Eingangsleistung, abhängig von der Drehzahl

**Abbildung 5.18** Messergebnisse des Motors *QSH2818-51* im Betriebsmodus *StealthChop*. Die Drehmomentkurven in a) sind mit Ausnahme von der 12V-NT-Kurve sehr nahe beieinander, die 12V-NT-Kurve erreicht bei keiner Drehzahl das Drehmoment der drei anderen Spannungsversorgungen. Die Leistungsaufnahme in b) zeigt die Verlustleistung des Spannungswandlers, wenn 36V-NT und 36V-DCDC verglichen werden. Zu erkennen ist, dass die  $U_{VAR}$ -Kurve bis  $400 \text{ min}^{-1}$  eine niedrigere Leistungsaufnahme hat, als die 36V-DCDC-Kurve, teilweise sogar niedriger, als die 36V-NT-Kurve. Ab  $500 \text{ min}^{-1}$  sind die Leistungsaufnahmen von 36V-DCDC und  $U_{VAR}$  beinahe identisch, die Differenz zur der Leistungsaufnahme von 36V-NT steigt jedoch.

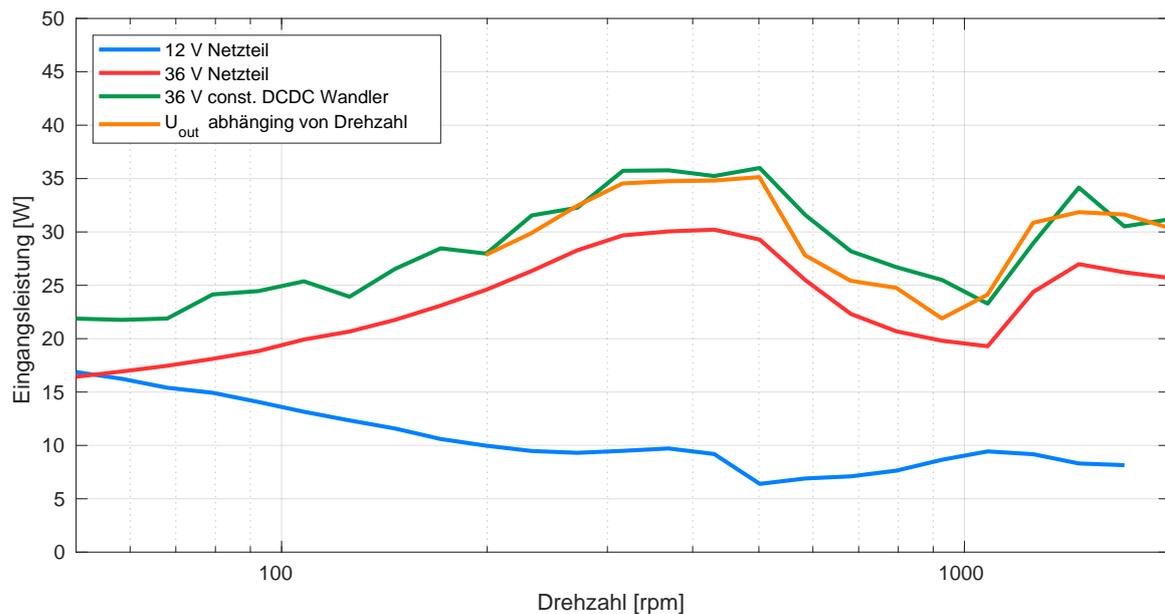
**TRINAMIC QSH4218-51**

Die Verläufe der 36V-NT und der  $U_{VAR}$  Drehmomentkurven in Abbildung 5.19a sind sehr ähnlich zueinander, von  $600 \text{ min}^{-1}$  bis  $1300 \text{ min}^{-1}$  ist das Drehmoment von  $U_{VAR}$  etwas größer. Der Drehmomentverlauf von 36V-DCDC weicht ein wenig von den beiden Kurven ab, er steigt bei  $230 \text{ min}^{-1}$  auf  $48 \text{ Nm}$  an und fällt danach kontinuierlich ab. Dabei liegt das erreichte Drehmoment zwischen  $250 \text{ min}^{-1}$  und  $600 \text{ min}^{-1}$  unter den Drehmomentverläufen von 36V-NT und  $U_{VAR}$ , zwischen  $600 \text{ min}^{-1}$  und  $1100 \text{ min}^{-1}$  darüber und danach bis  $2000 \text{ min}^{-1}$  wieder darunter.

Die Leistungsaufnahme in Abbildung 5.19b zeigt wieder, dass der Aufwärtswandler zum Erreichen des gleichen Drehmoments der 36V-NT-Kurve konstant 4 - 5 W mehr verbraucht. Dabei verbraucht  $U_{VAR}$  in dem Drehzahlbereich von  $500 \text{ min}^{-1}$  bis  $1020 \text{ min}^{-1}$  etwas weniger Leistung als 36V-DCDC.



(a) Drehmoment abhängig von der Drehzahl



(b) Für a) benötigte Eingangsleistung, abhängig von der Drehzahl

**Abbildung 5.19** Messergebnisse des Motors *QSH4218-51* im Betriebsmodus *StealthChop*. Die Drehmomentkurven in a) sind nicht so nah beieinander wie im *SpreadCycle* Modus. Die 12V-NT-Kurve bricht direkt ab  $50 \text{ min}^{-1}$  ein, 36V-NT und  $U_{VAR}$  sind sich recht ähnlich. Die 36V-DCDC-Kurve sinkt ab  $230 \text{ min}^{-1}$  recht gerade ab. Bei der Leistungsaufnahme in b) zeigt sich die zusätzliche Leistungsaufnahme des Aufwärtswandlers von 5 - 6 W im Vergleich von 36V-DCDC zu 36V-NT. Bei der  $U_{VAR}$ -Kurve liegt die Leistungsaufnahme etwas niedriger als bei der 36V-DCDC-Kurve.

### 5.2.3 Vergleich der Messungen

Die Messungen zeigen, dass bezogen auf die Drehmomentkennlinie und die Leistungsaufnahme sich die Betriebsarten *Spreadcycle* und *StealthChop* sehr ähnlich verhalten, bei Verwendung des selben Schrittmotors. Eine Ausnahme davon ist der TRINAMIC QSH2818-51, welcher im *SpreadCycle*-Modus bei der 12V-NT Versorgung ein höheres Drehmoment erzeugen kann, als im *StealthChop*-Modus. Zwischen den drei getesteten Schrittmotoren ergaben sich jedoch teilweise große Unterschiede bezüglich des hinzugewonnenen Drehmoments durch die dynamische Spannungsanpassung. Diese sollen in der folgenden Tabelle 5.1 kurz zusammengefasst werden.

**Tabelle 5.1** Zusammenfassung der Ergebnisse von den Messungen der drei verschiedenen Schrittmotoren

	Auswirkungen	Erkenntnisse
MOONS MS17HD4P4150	Drehmoment bis $350 \text{ min}^{-1}$ bei allen Versorgungsarten ähnlich. Danach bricht das Drehmoment von 12V-NT ein, die anderen Versorgungsarten brechen ab $1000 \text{ min}^{-1}$ stärker ein.	Eine erhöhte Spannungsversorgung erzeugt erst ab einer Drehzahl von $350 \text{ min}^{-1}$ Vorteile bzgl. des Drehmoments. Bei langsameren Anwendungen keine Vorteile, jedoch erhöhte Leistungsaufnahme bei höheren Versorgungsspannungen.
TRINAMIC QSH2818-51	Drehmoment von 12V-NT über gesamten Drehzahlbereich geringer, dafür auch durchgehend geringere Leistungsaufnahme, wobei im <i>SpreadCycle</i> -Modus zu Beginn ein höheres Drehmoment erzeugt werden kann. Die anderen Versorgungsarten liegen sehr nahe beieinander.	Bei diesem Schrittmotor ist eine signifikante Verbesserung des Drehmoments über den gesamten Drehzahlbereich durch das entwickelte Modul zu erreichen.
TRINAMIC QSH4218-51	Drehmoment von 12V-NT über gesamten Drehzahlbereich signifikant geringer, dafür auch durchgehend geringere Leistungsaufnahme. 36V-DCDC sowie $U_{VAR}$ erzeugen ähnlichen Drehmomentverlauf wie 36V-NT, bei 5 - 6 W höherer Leistungsaufnahme.	Bei diesem Schrittmotor ist eine signifikante Verbesserung des Drehmoments über den gesamten Drehzahlbereich durch das entwickelte Modul zu erreichen.

# 6 Schlussfolgerung

Im Folgenden sollen die gemessenen Ergebnisse diskutiert sowie bewertet werden, in einem Ausblick auf mögliche Weiterentwicklungen und Verbesserungen eingegangen werden und diese Arbeit in einem abschließenden Fazit enden.

## 6.1 Diskussion der Ergebnisse

Die Ergebnisse und Erkenntnisse dieser Arbeit und daraus resultierende Schlussfolgerungen und Bewertungen sollen im Folgenden kurz dargelegt werden.

Durch die Größenvorgaben an die Platine, aufgrund der Vorgabe das zu entwickelnde Modul in 3D-Druckerboards verwenden zu können, war die Leistungsfähigkeit und Belastbarkeit der räumlich größeren Bauelemente (Induktivität, Ausgangskondensator, MOSFET) stark eingeschränkt. Daraus resultierte eine stärkere Erwärmung der räumlich kleinen Induktivität unter größerer Last, weshalb insbesondere bei den Aufnahmen der Motorkennlinien im Teststand aktiv gekühlt werden musste. Durch die kleine Platine konnte auch nur wenig Wärme durch die Platine selber aufgenommen und abgeleitet werden, was durch die starke Wärmeentwicklung des *TMC2130* zusätzlich erschwert wurde.

Da der zu entwickelnde Aufwärtswandler eine sich ständig ändernde Ausgangsspannung generieren muss, war eine Optimierung der Komponenten und Regelung auf eine konstante Sollspannung nicht möglich. Dementsprechend wurde kein besonders hoher Wirkungsgrad erreicht, was sich besonders auf dem Motorteststand gezeigt hat. Dort war die Leistungsaufnahme des entwickelten Moduls im Schnitt 5 W höher als bei der Verwendung des Labornetzteils mit einer hohen Spannung, bei ähnlichem Drehmomentverlauf.

Weiterhin konnte durch die verwendete Hysterese-Regelung, die für die ständige Änderung der Sollspannung notwendig ist, keine konstante Schaltfrequenz verwendet werden. Das macht es zum schwieriger, einen Tiefpass am Ausgang zur Filterung der Ausgangsspannung entsprechend auszulegen. Zusätzlich verursacht eine Schaltfrequenz im hörbaren Frequenzbereich unter 20 kHz bei Verwendung von Keramik kondensatoren als Ausgangskondensator durch den piezoelektrischen Effekt störende Geräusche.

Die ursprüngliche Idee, durch Einsetzen der entwickelten Module einen fertigen, kommerziellen 3D-Drucker bei ansonsten unveränderter Elektronik schneller drucken zu lassen, konnte nicht erfolgreich umgesetzt werden, da in einem 3D-Drucker hauptsächlich die Mechanik und die Abkühlgeschwindigkeit des geschmolzenen Plastiks die maximale Geschwindigkeit des Druckens bestimmen.

Die Ergebnisse des Motorteststands zeigen, dass es bezüglich des hinzugewonnenen Drehmoments kaum einen Unterschied macht, ob die Spannung dynamisch an die Drehzahl angepasst wird oder ob der Aufwärtswandler durchgehend die maximal mögliche Spannung generiert. Allerdings ist eine herkömmlicher Aufwärtswandler, der eine konstante Ausgangsspannung generiert, preiswerter, bereits erfolgreich entwickelt und erprobt sowie effizienter als die hier entwickelte Variante.

Da allerdings von TRINAMIC das Interesse besteht, einen eigenen Spannungswandler in einen neuen Chip zu integrieren, kann nicht auf ein fertiges Produkt zurückgegriffen werden. Es könnte ein Spannungswandler in den zu entwickelnden Chip integriert werden, der eine konstante Ausgangsspannung generiert, was voraussichtlich einfacher und schneller zu entwickeln wäre.

Die erreichte Restwelligkeit und Rauschen der Ausgangsspannung ist trotz des nachträglich eingebauten Tiefpasses nicht optimal und erzeugt teilweise zusätzliche Motorgeräusche, die insbesondere bei Verwendung des *Stealthchop*-Betriebsmodus gut hörbar sind.

Durch die Verwendung des Aufwärtswandlers konnte bei gleicher Versorgungsspannung ein höheres Drehmoment bei hohen Drehzahlen erreicht werden, ohne dass ein zusätzliches Netzteil notwendig ist. Durch diese Kombination aus Schrittmotortreiber und Aufwärtswandler kann in bereits vorhandenen Systemen bei gleichbleibenden Schrittmotor und Versorgung eine Vergrößerung, beziehungsweise längeres Aufrechterhalten des Drehmoments, erreicht werden.

Dadurch wäre es auch theoretisch möglich, in preiseempfindlichen Anwendungen einen vorhandenen teuren Schrittmotor mit herkömmlicher Ansteuerung durch einen kleineren, günstigeren Schrittmotor mit dynamischer Spannungsanpassung zu ersetzen, bei gleichem Drehmoment. Weiterhin lassen sich die Herstellungskosten von ICs durch eine Massenproduktion stärker reduzieren, als die Kosten für mechanische Systeme, wie Schrittmotoren.

Das Konzept der dynamischen Spannungsanpassung ist gut skalierbar, es können auch Module entwickelt werden, die eine wesentlich höhere Leistung bereitstellen können, zum Beispiel für große NEMA32-Schrittmotoren. Durch ein Austauschen der Leistungskomponenten sowie des Schrittmotortreiber ist dies einfach möglich.

## 6.2 Ausblick

Die Aufgabe dieser Masterarbeit war es herauszufinden, ob eine dynamische Spannungsanpassung für Schrittmotortreiber die gewünschten Ergebnisse erzeugen kann. Da es sich herausgestellt hat, dass die Spannungsanpassung das Drehmoment für einen größeren Drehzahlbereich aufrecht erhalten kann, lohnt es sich in dieser Richtung weiter zu entwickeln und zu forschen. Eine Patentanmeldung durch TRINAMIC für die dynamische Spannungsanpassung in Schrittmotortreibern ist bereits erfolgt. Im Folgenden sollen weitere, noch zu erprobende Verbesserungen und Weiterentwicklungen erläutert werden. Zusätzlich besteht seitens TRINAMIC das Interesse, einen Schrittmotortreiber mit integrierten Spannungswandler sowie ein skalierbaren Modul auf Basis dieser Technologie zu entwickeln.

### 6.2.1 Mögliche Verbesserungen und Weiterentwicklungen

Der erreichte Zustand des entwickelten Moduls ermöglicht eine bessere Ausnutzung einer vorhandenen niedrigen Spannungsversorgung und ermöglicht es, das Drehmoment eines Schrittmotors über einen größeren Drehzahlbereich aufrecht zu erhalten. Weiterhin wären viele Verbesserungen möglich, um die Effizienz und andere Eigenschaften des Moduls zu verbessern. Im Folgenden sind deshalb mögliche Verbesserungen und Weiterentwicklungen aufgelistet:

#### **Zusätzliche Primärstrommessung**

Die jetzige Art des Sanftanlaufs, dessen Aufgabe die Reduzierung des Einschaltstroms ist, ist eine Steuerung, die nur die aktuell erzeugte Ausgangsspannung auswertet. Durch eine zusätzliche Messung des Eingangsstroms des Aufwärtswandlers wäre eine Regelung möglich, die den benötigten Eingangsstrom auswerten könnte. Dadurch könnten Stromspitzen verhindert werden, die eventuell weitere, an das selbe Netzteil angeschlossene Komponenten, beschädigen. Weiterhin könnte dadurch der maximal zulässige Strom eingestellt werden, den das versorgende Netzteil liefern kann. Mit dem Wissen des maximal entnehmbaren Stroms könnte, bei Verwendung eines leistungsfähigen Netzteils, die Leistungsaufnahme so angepasst werden, dass die benötigte Spannung schneller erreicht wird. Gleichzeitig könnte bei Verwendung eines schwächeren Netzteils dieses durch eine entsprechend langsamere Steigrate der Spannung berücksichtigt werden.

### **Einfachere Einstellung der Grenzggeschwindigkeiten und Grenzspannungen**

In dem aktuellem Zustand der FPGA-Firmware muss für eine Anpassung der Parameter, die die Grenzggeschwindigkeiten und Grenzspannungen des Moduls bestimmen (siehe Abbildung 4.5), die Firmware selber geändert und der FPGA neu programmiert werden. Abgesehen davon, dass dadurch bei falschen Parametern das Modul irreparabel beschädigt werden könnte, ist auch ein spezieller Programmieradapter notwendig. Eine bessere Lösung wäre es, nur die benötigten Parameter über ein bekanntes Protokoll, wie zum Beispiel eine serielle Schnittstelle oder SPI, zu übertragen. Dadurch könnten andere Teile der Firmware nicht versehentlich verändert werden. Weiterhin könnte diese Konfigurierung zum Beispiel auch von einem vorhandenem Mikrocontroller durchgeführt werden, der den Schrittmotortreiber ansteuert.

### **Software zur Berechnung der benötigten Grenzparameter**

Um die notwendigen Rechenparameter für die Grenzggeschwindigkeiten und Grenzspannungen zu berechnen, müssen bis jetzt die in Kapitel 4.2.1 beschriebenen Rechenschritte manuell durchgeführt werden, was ein Fehlerpotential bietet sowie die Konfigurierung des Moduls umständlich macht. Eine Software mit graphischer Benutzeroberfläche, in die nur die beiden Grenzggeschwindigkeiten des Motors in Umdrehungen pro Minute oder Hertz sowie die maximale Ausgangsspannung in Volt eingegeben werden müssen, um daraus die notwendigen Parameter zu berechnen, würde die Konfiguration stark vereinfachen. Zusätzlich könnte, wie weiter oben beschrieben, das Schreiben der Parameter mittels serieller Schnittstelle oder anderen Protokollen direkt aus der Software erfolgen. In der Software könnten auch gut funktionierende Voreinstellung, beziehungsweise Beispielkonfigurationen für TRINAMIC Schrittmotoren, eingespeichert sein. Zusätzlich könnte eine automatische Überprüfung erfolgen, ob die eingegebenen sowie errechneten Werte überhaupt verwendet werden können, zum Beispiel eine zu große gewünschte Ausgangsspannung oder ein zu steiler Anstieg der Sollspannung in Abhängigkeit der Drehzahl.

### **Messung des Tastverhältnis der Motor-PWM**

Die implementierte Auswertung des Schrittsignals, also der aktuellen Drehzahl des Schrittmotors, bringt gewisse Nachteile mit sich. Es muss von dem Benutzer eine einmalige Einstellung der Grenzggeschwindigkeiten erfolgen. Um den verwendeten Motor möglichst effizient nutzen zu können, muss wie in Kapitel 4.2.1 beschrieben, die Kennlinie des Motors bei der Eingangsspannung des Aufwärtswandlers sowie bei der maximalen gewünschten Ausgangsspannung des Aufwärtswandlers in dem gewünschten Betriebsmodus bekannt sein. Dies ist häufig nicht möglich, da in vielen Datenblättern von Schrittmotoren nur der Vollschrittmodus für Kennlinien benutzt wird und die meisten Anwender keinen geeigneten Motorteststand besitzen, um diese selber aufzunehmen. Weiterhin lässt sich mit der verwendeten Auswertungsmethode nicht nachprüfen, ob die erzeugte Spannung ausreicht, um

trotz der gegen-elektromotorischen Kraft den Sollstrom in den Schrittmotor einzuprägen. Eine Möglichkeit wäre es, das Tastverhältnis der PWM des Strom-Choppers auszuwerten (siehe Kapitel 2.1.3). Wenn das Tastverhältnis sehr klein wird, der Sollstrom also in sehr kurzer Zeit einpräglich ist, kann die Spannung reduziert werden, um keine unnötigen Verluste durch den Aufwärtswandler zu erzeugen. Wenn das Tastverhältnis sehr groß ist, bedeutet das, dass der Sollstrom, mit der aktuellen Spannung, sehr lange braucht, um eingepreßt zu werden. Wenn das Tastverhältnis der PWM maximal ist, bedeutet das, dass es dem Strom-Chopper nicht möglich ist, den eingestellten Sollstrom in die Motorspulen einzuprägen. Das kann dann zum Einbrechen des Drehmoments und zu Schrittverlusten führen. Um dem vorzubeugen sollte deshalb bei einem größeren Tastverhältnis die Spannung durch den Aufwärtswandler erhöht werden. Durch diese Betriebsart müsste nur die maximale Ausgangsspannung eingestellt werden und die Spannungsanpassung würde sich unabhängig von Motor und Betriebsart automatisch einstellen.

### **Einsparen des ADC**

Durch die Verwendung der Hysterese-Regelung ist die genaue Messung der absoluten Ausgangsspannung nicht notwendig. Es muss nur bekannt sein, ob die erzeugte Ausgangsspannung größer oder kleiner als der Sollwert ist. Das könnte mit einem Komparator und einem simplen Digital-Analog-Wandler, zum Beispiel einem PWM-DAC [5], erfolgen. Diese Art von Digital-Analog-Wandler benötigt nur ein generiertes PWM-Signal, welches von dem FPGA erzeugt werden könnte, und einen RC-Tiefpass. Ein zusätzlicher analoger Komparator ist günstiger, als der aktuell verwendete ADC, wodurch die Produktionskosten gesenkt werden könnten, ohne die Funktion zu beeinträchtigen.

### **Konstante Schaltfrequenz**

Durch die Verwendung der Hysterese-Regelung ist keine konstante Schaltfrequenz des Transistors möglich, woraus sich gewisse Nachteile ergeben. Zum einen sinkt die Schaltfrequenz zeitweise in den Bereich unter  $20\text{ kHz}$  ab, was bei Verwendung von Keramik Kondensatoren als Ausgangskondensator durch den piezoelektrischen Effekt hörbare hochfrequente Geräusche hervorruft [71], die in ruhiger Umgebung stören könnten. Zum anderen würde eine konstante Schaltfrequenz das Auswählen einer Induktivität ermöglichen, die für diese Schaltfrequenz optimiert ist, was in einen höheren Wirkungsgrad resultieren würde. Weiterhin wäre eine Kombination aus der Hysterese-Regelung und PWM-Regelung möglich, bei der die Hysterese-Regelung während des Anstiegs und Abfalls der Sollspannung verwendet wird, zum Beispiel bei Be- und Entschleunigungsphasen des Schrittmotors, um ein möglichst dynamisches Verhalten zu ermöglichen. Die PWM-Regelung könnte dann verwendet werden, wenn eine Sollspannung eine Zeit konstant gehalten werden soll, zum Beispiel wenn der Schrittmotor sich mit einer konstanten Drehzahl dreht. Sie könnte die konstante Sollspannung feiner regeln und so eine gleichmäßigere Ausgangsspannung erzeugen.

### **Nichtlineare Abhängigkeit der Ausgangsspannung von der Drehzahl**

In der entwickelten Firmware wird die Abhängigkeit der Ausgangsspannung von der Drehzahl mithilfe einer Geradengleichung beschrieben. Anstelle der Geradengleichung könnte jedoch auch ein anderer mathematischer Zusammenhang genutzt werden, die zum Beispiel einer invertierten Schrittmotorkennlinie ähnelt. Dadurch könnte eventuell eine bessere Ausnutzung des möglich Drehmoments erreicht werden.

### **Verwendung eines zweiten Schalters anstelle einer Diode**

In der aktuellen Version wird nur ein aktiv schaltendes Element, der Leistungsmosfet, verwendet. Der zweite Schalter, die Diode, kann nicht aktiv geschaltet werden, sondern schaltet von alleine durch die angelegte Spannungsdifferenz. Dadurch kommt es zu einem Überschneiden der Einschaltzeit beider Komponenten, in denen der Mosfet und die Diode gleichzeitig leiten. Es kommt zu Schaltverlusten und Spannungsspitzen, da der Mosfet unter Last schaltet. Weiterhin fällt über die Diode aufgrund ihrer Schwellenspannung immer eine Spannung ab, was den Wirkungsgrad mindert. Durch ein Ersetzen der Diode durch einen zweiten aktiv geschalteten Mosfet, wird zum einen der Verlust durch die Schwellenspannung verhindert, zum anderen können beide Schalter so angesteuert werden, dass zwischen ihren Einschaltzeiten kurze Perioden sind, in denen kein Schalter aktiv ist. Dies wird als spannungsloses Schalten beziehungsweise Zero Voltage Switching (ZVS) bezeichnet und reduziert Schaltverluste [72] [21].

## **6.2.2 Integration in einen Chip**

Das hauptsächliche Interesse seitens TRINAMIC an dieser Masterarbeit war es herauszufinden, ob sich durch Verwendung einer dynamischen Spannungsanpassung aus einer vorhandenen niedrigen Versorgungsspannung ein verbesserter Drehmomentverlauf über den Drehzahlbereich erreichen lässt. Abhängig von den Ergebnissen, von unter anderem dieser Arbeit, wird dann entschieden, ob ein neuer Chip entwickelt werden soll, der neben einem Schrittmotortreiber auch einen Spannungswandler beinhaltet. Dabei sollen so viele Komponenten wie möglich in den Chip integriert werden, die bei dem entwickelten Modul noch diskret sind. Idealerweise müssten nur noch eine Induktivität, ein Leistungsschalter, eine Leistungsdiode und die Ausgangskapazität an den IC angeschlossen werden. Ein großer Vorteil wäre, dass aufgrund der direkten Integration auf dem selben Chip wie der Schrittmotortreiber, die aktuellen Geschwindigkeits- und Beschleunigungswerte direkt aus den Motortreiber-Registern ausgelesen werden könnten. Eine weitere Möglichkeit wäre es, das Tastverhältnis der PWM der beiden H-Brücken, welches intern in dem

Schrittmotortreiber als digitales Signal vorliegt, als auszuwertendes Signal für den Aufwärtswandler zu verwenden. Dadurch müsste die PWM nicht erst durch Messungen ausgewertet werden. Wahrscheinlich würde allerdings der Großteil der Regelung des Aufwärtswandlers nicht auf dem in dieser Arbeit entwickelten VHDL-Code basieren, sondern analog aufgebaut werden.

### 6.2.3 Entwicklung eines skalierbaren Moduls

Aufgrund der Ergebnisse dieser Arbeit ergab sich firmenseitiges Interesse ein Modul zu entwickeln, welches nicht an die Größenvorgaben durch den 3D-Druckermarkt gebunden ist (siehe Kapitel 3.4). Dieses Modul soll sich an der Größe und dem Aufbau der TRINAMIC-Evaluation-Boards [41] orientieren und in der Leistungsfähigkeit skalierbar sein, indem unter anderem leistungsfähigere Komponenten verbaut werden.

## 6.3 Zusammenfassung und Fazit

Grundsätzlich lässt sich feststellen, dass die Annahme, die Spannung des Schrittmotors in Abhängigkeit der Drehzahl zu erhöhen, um über einen größeren Drehzahlbereich ein hohes Drehmoment zu erhalten, bestätigt werden konnte. Ein auf dieser Idee basierendes Modul, welches einen Schrittmotortreiber sowie den Aufwärtswandler inklusive Auswertung des Motorsteuersignals enthält, wurde entwickelt und erfolgreich getestet. Die Verwendung des Schrittsignals zur Bestimmung der Drehzahl, in Verbindung mit der linearen Abhängigkeit der erzeugten Ausgangsspannung von dieser Drehzahl, hat gute Ergebnisse geliefert und das Drehmoment über einen weiten Drehzahlbereich aufrechterhalten können. Der erreichte Wirkungsgrad des entwickelten Aufwärtswandlers in Verbindung mit dem Schrittmotortreiber ist nicht so hoch wie erhofft. Weiterhin musste festgestellt werden, dass handelsübliche 3D-Drucker für den Heimanwender nicht unbedingt schneller drucken können, nur weil der Schrittmotor bei höheren Drehzahlen keine Schritte verliert. Zusätzlich haben sich viele weitere noch zu erforschende Fragen ergeben. Dennoch würde diese Technologie eine gute Erweiterung für den Schrittmotortreiber-Markt darstellen, da eine signifikante Verbesserung des Drehmomentverlaufs in höheren Drehzahlen erreicht werden kann.

# Literaturverzeichnis

- [1] Marlin: Open Source 3D Printer Firmware. <http://marlinfw.org/> (Eingesehen am 28.11.2018).
- [2] academics GmbH. *Widerstandsreihe*. <http://deacademic.com/dic.nsf/dewiki/1506382> (Eingesehen am 21.11.2018).
- [3] Harmonic Drive AG. *Schrittmotor - Definition und Funktionsweise*. <https://harmonicdrive.de/de/service-downloads/wissen/glossar/begriff/schrittmotor/> (Eingesehen am 19.11.2018).
- [4] Dr. Arne Albertsen. *Elko-Lebensdauerabschätzung*. Jianghai Europe Electronic Components GmbH. [http://jianghai-europe.com/wp-content/uploads/JIANGHAI\\_Elko\\_Lebensdauer\\_AAL.pdf](http://jianghai-europe.com/wp-content/uploads/JIANGHAI_Elko_Lebensdauer_AAL.pdf) (Eingesehen am 21.11.2018).
- [5] David M. Alter. *Using PWM Output as a Digital-to-Analog Converter on a TMS320F280x Digital Signal Controller*. Texas Instruments Incorporated. <http://www.ti.com/lit/an/spraa88a/spraa88a.pdf> (Eingesehen am 21.11.2018).
- [6] Lattice Semiconductor Corporation. *LCMXO2-1200HC Datenblatt*. [https://www.latticesemi.com/view\\_document?document\\_id=38834](https://www.latticesemi.com/view_document?document_id=38834) (Eingesehen am 21.11.2018).
- [7] ON Semiconductor Corporation. *Effects of High Switching Frequency on Buck Regulators*. <https://www.onsemi.com/pub/Collateral/TND388-D.PDF> (Eingesehen am 21.11.2018).
- [8] TDK Corporation. *Guide to Replacing an Electrolytic Capacitor with an MLCC*. <https://product.tdk.com/info/en/products/capacitor/ceramic/mlcc/technote/solution/mlcc03/index.html> (Eingesehen am 19.12.2018).
- [9] Prof. Niklaus Degunda. *Regelungstechnik*. [https://ndegunda.ch/wp-content/uploads/2016/02/script\\_rtGL\\_12.pdf](https://ndegunda.ch/wp-content/uploads/2016/02/script_rtGL_12.pdf) (Eingesehen am 21.11.2018).

- [10] Strong Edge Dynamics. *Stepping Motion Profiles in Realtime*. [http://picprog.strongedge.net/step\\_prof/step-profile.html](http://picprog.strongedge.net/step_prof/step-profile.html) (Eingesehen am 21.11.2018).
- [11] Watterott electronic GmbH. *SilentStepStick*. <https://learn.watterott.com/silentstepstick/> (Eingesehen am 21.11.2018).
- [12] Digi-Key Electronics. *TMC SILENTSTEPSTICK: STEPPER MOTOR DRIVER BRD TMC2100*. <https://www.digikey.de/product-detail/de/trinamic-motion-control-gmbh/TMC-SILENTSTEPSTICK/1460-1159-ND/5724190> (Eingesehen am 2.12.2018).
- [13] Learnabout Electronics. *Module 3.2: Boost Converters*. <http://www.learnabout-electronics.org/PSU/psu32.php> (Eingesehen am 1.12.2018).
- [14] Stack Exchange. *TIP-Barrel test of oscilloscope*. <https://electronics.stackexchange.com/questions/221292/tip-barrel-test-of-oscilloscope> (Eingesehen am 9.12.2018).
- [15] Pham Galal. *Division Algorithms and Hardware Implementations*. [https://www.seas.ucla.edu/ingrid/ee213a/lectures/division\\_presentV2.pdf](https://www.seas.ucla.edu/ingrid/ee213a/lectures/division_presentV2.pdf) (Eingesehen am 21.11.2018).
- [16] Beckhoff Automation GmbH. *Application Note DK9221-0210-0014 Schrittmotor*. [http://download.beckhoff.com/download/document/Application\\_Notes/DK9221-0210-0014.pdf](http://download.beckhoff.com/download/document/Application_Notes/DK9221-0210-0014.pdf) (Eingesehen am 20.11.2018).
- [17] Beta LAYOUT GmbH. *Strombelastbarkeit von Kupfer (CU)-Leiterbahnen auf Basismaterial*. [http://www.pcb-pool.com/download/spezifikation/deu\\_cmso001\\_strombelastbarkeit.pdf](http://www.pcb-pool.com/download/spezifikation/deu_cmso001_strombelastbarkeit.pdf) (Eingesehen am 21.11.2018).
- [18] Renesas Electronics Europe GmbH. *Choosing and Using Bypass Capacitors*. <https://www.intersil.com/content/dam/Intersil/documents/an13/an1325.pdf> (Eingesehen am 19.11.2018).
- [19] Peter Goetz. *EMK eines sich drehenden Motors*. <http://www.goetz-automation.de/Schrittmotor/SchrittmotorEMK.htm> (Eingesehen am 19.11.2018).
- [20] Brigitte Hauke. *Basic Calculation of a Boost Converter's Power Stage*. Texas Instruments Incorporated. <http://www.ti.com/lit/an/slva372c/slva372c.pdf> (Eingesehen am 19.11.2018).
- [21] Sanjay Havanur. *Beware of Zero Voltage Switching*. Vishay Intertechnology Inc. [https://www.mouser.com/pdfdocs/Vishay\\_Zero\\_Voltage\\_Switching.pdf](https://www.mouser.com/pdfdocs/Vishay_Zero_Voltage_Switching.pdf) (Eingesehen am 21.11.2018).

- [22] C. Hsu, T. Chang, and C. Wey. A voltage-mode hysteretic boost dc-dc converter with dual control modes. In *2012 Asia Pacific Conference on Postgraduate Research in Microelectronics and Electronics*, pages 67–71, Dec 2012.
- [23] Analog Devices Inc. *LTspice*. <https://www.analog.com/en/design-center/design-tools-and-calculators/ltspice-simulator.html> (Eingesehen am 21.11.2018).
- [24] Geckodrive Inc. *Step Motor Basics*. <https://www.geckodrive.com/support/step-motor-basics.html> (Eingesehen am 21.11.2018).
- [25] Littelfuse Inc. *SMBJ33A Datenblatt*. <https://www.littelfuse.de/products/tvs-diodes/surface-mount/smbj/smbj33a.aspx> (Eingesehen am 21.11.2018).
- [26] Microchip Technology Inc. *ATmega2560 Datenblatt*. [http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561\\_datasheet.pdf](http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561_datasheet.pdf) (Eingesehen am 21.11.2018).
- [27] Microchip Technology Inc. *HV9150 Datenblatt*. <http://ww1.microchip.com/downloads/en/DeviceDoc/20005689A.pdf> (Eingesehen am 19.11.2018).
- [28] Microchip Technology Inc. *MCP1407 Datenblatt*. <http://ww1.microchip.com/downloads/en/devicedoc/20002019c.pdf> (Eingesehen am 21.11.2018).
- [29] Vishay Intertechnology Inc. *VS-30BQ100-M3 Datenblatt*. <https://www.vishay.com/docs/93360/vs-30bq100m.pdf> (Eingesehen am 21.11.2018).
- [30] MOONS' Industries. *MS17HD4P4150 Datenblatt*. <https://www.moonsindustries.com/p/nema-17-standard-hybrid-stepper-motors/ms17hd4p4150-000004611110008897> (Eingesehen am 21.11.2018).
- [31] Maxim Integrated. *DC-DC Converter Tutorial*. <https://www.maximintegrated.com/en/app-notes/index.mvp/id/2031> (Eingesehen am 19.11.2018).
- [32] Maxim Integrated. *MAX11116 Datenblatt*. <https://datasheets.maximintegrated.com/en/ds/MAX11102-MAX11117.pdf> (Eingesehen am 21.11.2018).
- [33] Maxim Integrated. *MAX11665 Datenblatt*. <https://datasheets.maximintegrated.com/en/ds/MAX11661-MAX11666.pdf> (Eingesehen am 21.11.2018).

- [34] Douglas W. Jones. *Current Limiting for Stepping Motors*. 1995. <http://homepage.divms.uiowa.edu/~jones/step/current.html> (Eingesehen am 1.12.2018).
- [35] Steven Keeping. *Design Trade-offs when Selecting a High-Frequency Switching Regulator*. Electronic Products. <https://www.digikey.com/en/articles/techzone/2015/feb/design-trade-offs-when-selecting-a-high-frequency-switching-regulator> (Eingesehen am 21.11.2018).
- [36] Steven Keeping. *Managing Converter In-Rush Current*. <https://www.digikey.com/en/articles/techzone/2012/jun/managing-converter-in-rush-current> (Eingesehen am 21.11.2018).
- [37] Steven Keeping. *Voltage- and Current-Mode Control for PWM Signal Generation in DC-to-DC Switching Regulators*. Digi-Key Electronics, Oct 2014. <https://www.digikey.com/en/articles/techzone/2014/oct/voltage-and-current-mode-control-for-pwm-signal-generation-in-dc-to-dc-switching-regulators> (Eingesehen am 19.11.2018).
- [38] N. Keskar and G. A. Rincon-Mora. Self-stabilizing, integrated, hysteretic boost dc-dc converter. In *30th Annual Conference of IEEE Industrial Electronics Society, 2004. IECON 2004*, volume 1, pages 586–591 Vol. 1, Nov 2004.
- [39] Nanotec Electronic GmbH & Co. KG. *Funktionsweise eines Schrittmotors*. <https://de.nanotec.com/support/knowledge-base-pages/funktionsweise-eines-schrittmotors/> (Eingesehen am 19.11.2018).
- [40] TRINAMIC Motion Control GmbH & CO. KG. *Microstepping*. <https://www.trinamic.com/technology/std-technologies/microstepping/> (Eingesehen am 19.11.2018).
- [41] TRINAMIC Motion Control GmbH & CO. KG. *Modular Evaluation System*. <https://www.trinamic.com/support/eval-kits/> (Eingesehen am 21.11.2018).
- [42] TRINAMIC Motion Control GmbH & CO. KG. *PANdrive PD-42-2-1070 Datenblatt*. <https://www.trinamic.com/products/drives/pandrive-details/pd42-x-1070/> (Eingesehen am 19.11.2018).
- [43] TRINAMIC Motion Control GmbH & CO. KG. *QSH2818-51 Datenblatt*. [https://www.trinamic.com/fileadmin/assets/Products/Motors\\_Documents/QSH2818\\_manual.pdf/](https://www.trinamic.com/fileadmin/assets/Products/Motors_Documents/QSH2818_manual.pdf/) (Eingesehen am 19.11.2018).
- [44] TRINAMIC Motion Control GmbH & CO. KG. *QSH4218-51 Datenblatt*. [https://www.trinamic.com/fileadmin/assets/Products/Motors\\_Documents/QSH4218\\_manual.pdf](https://www.trinamic.com/fileadmin/assets/Products/Motors_Documents/QSH4218_manual.pdf) (Eingesehen am 19.11.2018).

- [45] TRINAMIC Motion Control GmbH & CO. KG. *SpreadCycle*. <https://www.trinamic.com/technology/adv-technologies/spreadcycle/> (Eingesehen am 19.11.2018).
- [46] TRINAMIC Motion Control GmbH & CO. KG. *StealthChop*. <https://www.trinamic.com/technology/adv-technologies/stealthchop/> (Eingesehen am 19.11.2018).
- [47] TRINAMIC Motion Control GmbH & CO. KG. *TMC2130 Datenblatt*. [https://www.trinamic.com/fileadmin/assets/Products/ICs\\_Documents/TMC2130\\_datasheet.pdf](https://www.trinamic.com/fileadmin/assets/Products/ICs_Documents/TMC2130_datasheet.pdf) (Eingesehen am 21.11.2018).
- [48] TRINAMIC Motion Control GmbH & CO. KG. *TMC262 / TMC260A & TMC261 Datenblatt*. [https://www.trinamic.com/fileadmin/assets/Products/ICs\\_Documents/TMC261\\_datasheet.pdf](https://www.trinamic.com/fileadmin/assets/Products/ICs_Documents/TMC261_datasheet.pdf) (Eingesehen am 19.11.2018).
- [49] Würth Elektronik GmbH & Co. KG. *WE-PD SMT-Speicherdrossel*. <http://katalog.we-online.de/de/pbs/WE-PD> (Eingesehen am 21.11.2018).
- [50] Kreuth, Ebert, Jufer, Kohler, Müller, Richter, and Traeger. *Elektrische Schrittmotoren, Kontakt & Studium Bd. 128: Elektrotechnik*. Expert-Verlag GmbH, 1985. ISBN: 3-8169-0042-9.
- [51] Fritz Kümmel. *Elektrische Antriebstechnik, Teil 1: Maschinen*. VDE-VERLAG GmbH, 1986. ISBN: 3-8007-1366-7.
- [52] Emil Lenz. Über die bestimmung der richtung der durch elektrodynamische vertheilung erregten galvanischen ströme. *Annalen der Physik*, 107(31):483 – 494, 1834.
- [53] Oriental Motor Co. Ltd. *Hybrid Stepper Motors and AlphaStep Hybrid Control Systems*. <https://www.orientalmotor.com/stepper-motors/technology/hybrid-stepper-motors-v-hybrid-control.html> (Eingesehen am 10.12.2018).
- [54] ROHM Co. Ltd. *Linear Regulator IC Series: Basics of Linear Regulators*. [https://www.mouser.com/pdfdocs/linear\\_reg\\_basic\\_appli-e.pdf](https://www.mouser.com/pdfdocs/linear_reg_basic_appli-e.pdf) (Eingesehen am 21.11.2018).
- [55] ROHM Co. Ltd. *Switching Regulator Basics: Control Methods (Voltage Mode, Current Mode, Hysteresis Control)*. <https://micro.rohm.com/en/techweb/knowledge/dcdc/s-dcdc/02-s-dcdc/97> (Eingesehen am 21.11.2018).
- [56] Brian T. Lynch. *Under the Hood of a DC/DC Boost Converter*. Texas Instruments Incorporated. [http://www.ti.com/download/trng/docs/seminar/Topic\\_3\\_Lynch.pdf](http://www.ti.com/download/trng/docs/seminar/Topic_3_Lynch.pdf) (Eingesehen am 19.11.2018).

- [57] Lothar Miller. *Division in VHDL*. <http://www.lothar-miller.de/s9y/archives/51-Konfigurierbarer-SPI-Master.html> (Eingesehen am 21.11.2018).
- [58] Lothar Miller. *Konfigurierbarer SPI-Master*. <http://www.lothar-miller.de/s9y/archives/29-Division-in-VHDL.html> (Eingesehen am 21.11.2018).
- [59] Pete Millett. *Improving current control for better stepper motor motion quality*. 2016. <http://www.eenewsanalog.com/content/improving-current-control-better-stepper-motor-motion-quality> (Eingesehen am 1.12.2018).
- [60] STMicroelectronics N.V. *STD40NF10 Datenblatt*. <https://www.st.com/resource/en/datasheet/std40nf10.pdf> (Eingesehen am 21.11.2018).
- [61] Osoyoo. *OSOYOO 3D Printer Kit with RAMPS 1.4 Controller + Mega 2560 board + 5pcs A4988 Stepper Motor Driver with Heatsink + LCD 12864 Graphic Smart Display Controller with Adapter For Arduino RepRap*. <https://www.amazon.com/OSOYOO-Printer-Controller-Stepper-Heatsink/dp/B0111ZSS2O> (Eingesehen am 2.12.2018).
- [62] Thomas Rechlin. *Elektromagnetische Störungen durch DC/DC Wandler-Filter beheben: Befilterung von DC/DC-Wandlern*. RECOM Power GmbH. <https://www.all-electronics.de/befilterung-von-dcdc-wandlern/> (Eingesehen am 19.11.2018).
- [63] RepRap. *RAMPS 1.4*. [https://reprap.org/wiki/RAMPS\\_1.4](https://reprap.org/wiki/RAMPS_1.4) (Eingesehen am 21.11.2018).
- [64] RepRap. *Stepper motor driver*. [https://reprap.org/wiki/Stepper\\_motor\\_driver](https://reprap.org/wiki/Stepper_motor_driver) (Eingesehen am 21.11.2018).
- [65] Steve Roberts. *DC/DC BOOK OF KNOWLEDGE: Praktische Tipps für Anwender*. RECOM Engineering GmbH & Co KG, 3. edition, 2017. [http://www.recom-power.com/fileadmin/EMEA/Downloads/Book\\_of\\_Knowledge/BookOfKnowledge\\_DE\\_WEB.pdf](http://www.recom-power.com/fileadmin/EMEA/Downloads/Book_of_Knowledge/BookOfKnowledge_DE_WEB.pdf) (Eingesehen am 19.12.2018).
- [66] Rummich, Ebert, Gfrörer, and Traeger. *Elektrische Schrittmotoren und -antriebe: Funktionsprinzip - Betriebseigenschaften - Messtechnik*. Expert-Verlag GmbH, 2007. 4. Auflage.
- [67] Thomas Schulz. *Betriebsverhalten der DC-DC Wandler*. Querom Elektronik GmbH. <https://www.querom.de/betriebsverhalten-der-dc-dc-wandler/> (Eingesehen am 19.11.2018).

- [68] Bill Schweber. *Stepper Motors Make the Right Moves with Precision, Ease and Smarter Drivers*. Mouser Electronics Inc. <https://www.mouser.de/applications/stepper-motors-smart-drivers/> (Eingesehen am 10.12.2018).
- [69] Frank De Stasi. *Working with Boost Converters*. Texas Instruments Incorporated. <http://www.ti.com/lit/an/snva731/snva731.pdf> (Eingesehen am 19.11.2018).
- [70] Luke Valenty. *TinyFPGA A2*. <https://store.tinyfpga.com/products/tinyfpga-a2> (Eingesehen am 21.11.2018).
- [71] Mingyue Zhao. *How to reduce acoustic noise of MLC-Cs in power applications*. Texas Instruments Incorporated. [https://e2e.ti.com/blogs\\_/b/powerhouse/archive/2016/08/09/how-to-reduce-acoustic-noise-of-mlccs-in-power-applications](https://e2e.ti.com/blogs_/b/powerhouse/archive/2016/08/09/how-to-reduce-acoustic-noise-of-mlccs-in-power-applications) (Eingesehen am 21.11.2018).
- [72] P. Zuk and S. Havanur. *Zero-Voltage Switching Full-Bridge Converter: Operation, FOM, and Guidelines for MOSFET Selection*. Vishay Intertechnology Inc. <https://www.vishay.com/docs/90936/an847.pdf> (Eingesehen am 21.11.2018).

# Tabellenverzeichnis

3.1	Ergebnisse und Erkenntnisse des 1. Funktionsmuster . . . . .	35
3.2	Ergebnisse und Erkenntnisse des 2. Funktionsmuster . . . . .	36
3.3	Ergebnisse und Erkenntnisse des 3. Funktionsmuster . . . . .	36
5.1	Zusammenfassung der Ergebnisse von den Messungen der drei verschiedenen Schrittmotoren . . . . .	101

# Abbildungsverzeichnis

2.1	Schematische Darstellung der unterschiedlichen Schrittmotorbauarten. Die Polzahl ist für eine bessere Anschaulichkeit stark reduziert. Bildquelle [66, S. 5-6] . . . . .	9
2.2	Detailansicht eines Hybridschrittmotorrotors. Die Zähne mit Nord- und Südmagnetisierung sind jeweils versetzt, um eine hohe Auflösung zu erreichen. . . . .	10
2.3	Detailansicht eines Hybridschrittmotorstators Bild modifiziert nach [39] . . . . .	11
2.4	Drehmomentkurve aus dem Datenblatt des <i>PD42-2-1070</i> von TRINAMIC [42] . . . . .	12
2.5	Unterschiedliche Wicklungsarten, welche bei Schrittmotoren Anwendung finden Bildquelle [68] . . . . .	13
2.6	Aufbau eines Schrittmotortreibers mit zwei H-Brückenschaltungen für einen zweiphasigen Schrittmotor mit bipolarer Wicklung . . . . .	14
2.7	Schaltzustände einer H-Brücke während der Chopper-Stromregelung Bildquelle [48, S. 32] . . . . .	15
2.8	Stromregelung einer Schrittmotorphase . . . . .	15
2.9	Verlauf des erzeugten magnetischen Feldes bei Verwendung des Vollschrittbetriebs . . . . .	16
2.10	Verlauf des erzeugten magnetischen Feldes bei Verwendung des Halbschrittbetriebs . . . . .	17
2.11	Mikroschrittauflösung . . . . .	18
2.12	Der verwendete Motorteststand zur Aufnahme der Kennlinien . . . . .	20
2.13	Grundlegender Aufbau eines Aufwärtswandlers . . . . .	21
2.14	Schaltzustände des Aufwärtswandlers. In a) ist der Ladevorgang dargestellt, in b) der Entladevorgang. Bilder modifiziert nach [13] . . . . .	22
2.15	Strom- und Spannungsverlauf im CCM Bildquelle [56, S. 3] . . . . .	24
2.16	Strom- und Spannungsverlauf im DCM Bildquelle [56, S. 4] . . . . .	26
2.17	Schematischer analoger Aufbau der Regelung eines Aufwärtswandlers mithilfe der Pulsweitenmodulation. Bildquelle [31] . . . . .	27
2.18	Schematischer Aufbau Hysterese-Regelung . . . . .	28

3.1	Konzept für die Entwicklung des Moduls . . . . .	31
3.2	Energieflussdiagramm des Moduls inklusive der Verlustleistungen und der nutzbaren Ausgangsleistung . . . . .	31
3.3	Schaltplan des simulierten Aufwärtswandlers in <i>LTSpice</i> . . . . .	32
3.4	Einschaltvorgang des simulierten, unregulierten Aufwärtswandlers (62.5 kHz) . . . . .	33
3.5	Einschaltvorgang des simulierten, unregulierten Aufwärtswandlers (1 MHz) . . . . .	34
3.6	Häufig verwendete Elektronik in 3D-Druckern . . . . .	37
3.7	Vereinfachter Schaltplan des Schrittmotortreibers. Der tatsächlich verwendete Schaltplan enthält zusätzliche Lötjumper zur Konfigurationen sowie Testpunkte und ist im Anhang unter Abbildung B.1 einsehbar. Diese Schaltung ist eine Modifikation der <i>SilentStepStick TMC2130</i> Schaltung von <i>Watterott electronic GmbH</i> [11]. . . . .	39
3.8	Vereinfachter Schaltplan des FPGA- und des ADC-Teils. Der tatsächlich verwendete Schaltplan enthält zusätzliche Stützkondensatoren sowie Testpunkte und ist im Anhang unter Abbildung B.2 einsehbar. . . . .	41
3.9	Vereinfachter Schaltplan des Leistungsteils. Der tatsächlich verwendete Schaltplan enthält zusätzliche Eingangs- und Ausgangskondensatoren sowie Stützkondensatoren für den Treiber-IC und Testpunkte und ist im Anhang unter Abbildung B.3 einsehbar. . . . .	43
3.10	Entwickelte und bestückte Platine . . . . .	44
3.11	Die entwickelte Platine wurde räumlich in die Kategorien Schrittmotortreiber, Regelungslogik und Leistungsteil eingeteilt . . . . .	44
3.12	Adapter Platine . . . . .	45
3.13	Nachträglich hinzugefügter RC-Tiefpass . . . . .	47
3.14	Entwickeltes Modul in RAMPS-Board . . . . .	48
4.1	Vereinfachte Darstellung des entwickelten Aufwärtswandlers . . . . .	49
4.2	Aufbau der Firmware . . . . .	50
4.3	Zustandsautomat FSM_1 zur Steuerung des SPI-Masters . . . . .	52
4.4	Zustandsautomat FSM_3 für die Regelung und Festlegung der Einschaltdauer des Transistors . . . . .	55
4.5	Abhängigkeit des zu regelnden Sollwerts von der gemessenen Schrittfrequenz . . . . .	57
4.6	Abhängigkeit des zu regelnden Sollwerts von der gemessenen Schrittfrequenz in den im FPGA verwendeten Einheiten . . . . .	60
4.7	Ablauf des Zustandsautomaten FSM_2 zur Berechnung des Sollwerts . . . . .	63
4.8	Zeitliche Verläufe der Zustandsautomaten bei keiner Parallelisierung, bei zwei parallelen Prozessen und bei drei parallelen Prozessen . . . . .	66
4.9	Beispielhafte Darstellung der erzeugten Schwingungen durch verschiedene Gesamtlauzeiten des Regelkreises . . . . .	66

---

4.10	Einschaltdauer $T_{on}$ des Transistors beim Sanftanlauf . . . . .	69
5.1	Energiefluss des kompletten Testaufbaus . . . . .	71
5.2	Wirkungsgrad über Spannungsbereich bei unterschiedlichen Ausgangsleistungen . . . . .	72
5.3	Wirkungsgrad in Abhängigkeit des Ausgangsstroms bei unterschiedlichen Ausgangsspannungen . . . . .	73
5.4	Tastkopf eines Oszilloskops mit aufgesetzter Massefeder Bildquelle [14] . . . . .	74
5.5	Ausgangsspannung bei konstanter resistiver Last mit verschiedenen Ausgangskondensatoren . . . . .	75
5.6	Vergleich der Rippelspannung . . . . .	76
5.7	Amplitude der Spannungsspitzen . . . . .	77
5.8	Spannungsabweichung von der konstant eingestellten Ausgangsspannung . . . . .	78
5.9	Verlauf eines Spannungssprungs von 12 V auf 36 V an einem 100 $\Omega$ Leistungswiderstand mit unterschiedlichen Ausgangskondensatoren. Verwendete Induktivität $L = 6.8 \mu H$ . . . . .	81
5.10	Verlauf des Eingangsstroms bei einem Spannungssprung von 12V auf 36V an einem 100 $\Omega$ Leistungswiderstand mit unterschiedlichen Ausgangskondensatoren. Verwendete Induktivität $L = 6.8 \mu H$ . . . . .	82
5.11	Verlauf der Ausgangsspannung bei einem Spannungssprung an einem Schrittmotor (QSH4218-51) mit unterschiedlichen Ausgangskondensatoren. Verwendete Induktivität $L = 6.8 \mu H$ . . . . .	84
5.12	Verlauf des Eingangsstroms bei einem Spannungssprung an einem Schrittmotor (QSH4218-51) mit unterschiedlichen Ausgangskondensatoren. Verwendete Induktivität $L = 6.8 \mu H$ . . . . .	86
5.13	Vergleich des Anstiegs des Ausgangsstroms bei verschiedenen Lasten	87
5.14	Messergebnisse des Motors MS17HD4P4150 von Moons' Industries im Betriebsmodus <i>SpreadCycle</i> . . . . .	90
5.15	Messergebnisse des Motors QSH2818-51 von TRINAMIC im Betriebsmodus <i>SpreadCycle</i> . . . . .	92
5.16	Messergebnisse des Motors QSH4218-51 von TRINAMIC im Betriebsmodus <i>SpreadCycle</i> . . . . .	94
5.17	Messergebnisse des Motors MS17HD4P4150 von Moons' Industries im Betriebsmodus <i>StealthChop</i> . . . . .	96
5.18	Messergebnisse des Motors QSH2818-51 im Betriebsmodus <i>StealthChop</i>	98
5.19	Messergebnisse des Motors QSH4218-51 im Betriebsmodus <i>StealthChop</i>	100
A.1	Das erste Funktionsmuster . . . . .	122
A.2	Das zweite Funktionsmuster . . . . .	123
A.3	Das dritte Funktionsmuster . . . . .	124

---

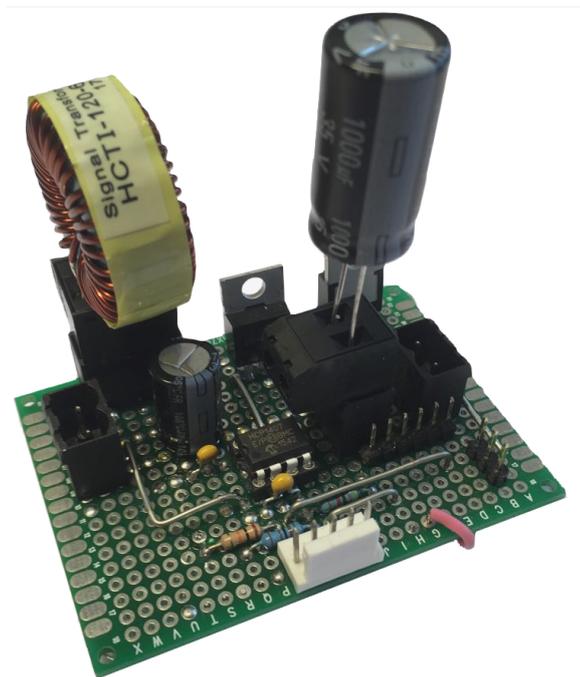
A.4	Schaltplan des ersten Funktionsmusters . . . . .	125
A.5	Schaltplan des zweiten Funktionsmusters . . . . .	126
A.6	Schaltplan des dritten Funktionsmusters . . . . .	127
B.1	Kompletter Schaltplan des Schrittmotortreibers . . . . .	129
B.2	Kompletter Schaltplan des FPGA- und ADC-Teils . . . . .	130
B.3	Kompletter Schaltplan des Leistungsteils . . . . .	131
B.4	Die entwickelte Platine besteht aus vier Lagen, die hier von Lage 1 bis 4 dargestellt sind . . . . .	132
B.5	Schaltplan des Adapters . . . . .	133
B.6	Beide Lagen die für den Adapter verwendet wurden . . . . .	133
C.1	Simulation der drei parallelisierten Zustandsautomaten . . . . .	135

# Listings

D.1	VHDL-Code der übergeordneten Main-Entity sowie der Zustandsautomaten und Zähler . . . . .	137
D.2	VHDL-Code der Teilereinheit . . . . .	141
D.3	VHDL-Code des SPI-Masters . . . . .	142
D.4	VHDL-Code des Frequenzzählers . . . . .	144

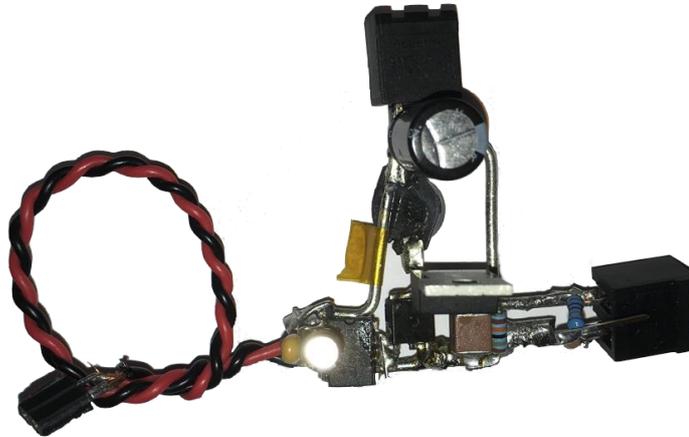
# A Aufbau der Funktionsmuster

## Funktionsmuster 1:



**Abbildung A.1** Das erste Funktionsmuster. Zu erkennen sind die auswechselbare Ausgangskapazität und die auswechselbare Induktivität.

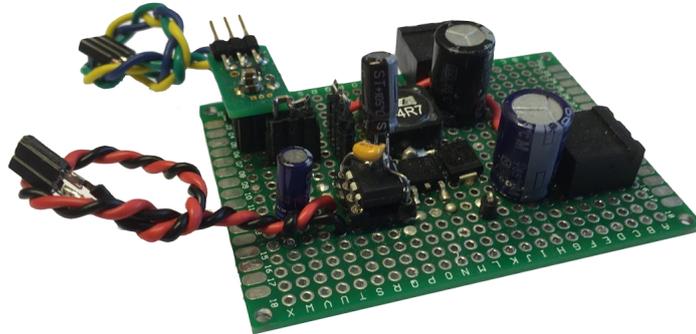
Das erste Funktionsmuster wurde auf einer Lochrasterplatine komplett aus THT-Bauteilen aufgebaut, wobei die verwendete Induktivität und der Ausgangskondensator steckbar gestaltet wurden. Die Komponenten, vor allem die Induktivität, sind bezogen auf die Leistung stark überdimensioniert, um Probleme, wie zum Beispiel einen thermischen Defekt der Induktivität zu verhindern. Es wird wie in der Simulation eine reine Steuerung verwendet, weswegen kein ADC zur Messung verwendet wird. Die Erzeugung des PWM-Signals erfolgt mit einem extern angeschlossenen Mikrocontroller. Es zeigt sich, dass bei gleicher Last und dem gleichen Schaltsignal wie in der Simulation, eine Spannung erzeugt wird, die vom Mittelwert nahe an der Simulation ist. Allerdings sind sehr starke Spannungsspitzen auf der Ausgangsspannung, was auf das Layout mittels der Lochrasterplatine zurückzuführen ist. Hieraus ergab sich eine unzureichende Masseführung sowie starkes kapazitives und induktives Überkoppeln.

**Funktionsmuster 2:**

**Abbildung A.2** Das zweite Funktionsmuster. Zu erkennen ist der fliegende Aufbau in Kombination mit SMD-Komponenten, um möglichst kurze Leitungslängen zu ermöglichen.

Für das zweite Funktionsmuster wurden keine Lochrasterplatine und keine THT-Bauteile verwendet, sondern es wurden SMD-Komponenten direkt miteinander verdrahtet (siehe Abbildung A.2a). So konnten wesentlich kürzere Strecken zwischen den Bauteilen erreicht werden, was induktives und kapazitives Koppeln reduziert. Weiterhin wurden mit den SMD-Komponenten Bauteilgrößen verwendet, die auch später auf der fertigen Platine Verwendung finden sollen. Das PWM-Signal wird von einem FPGA auf einem externen Board generiert [70]. In diesem Funktionsmuster ist ein Treiber-IC dem MOSFET vorgeschaltet, um ein möglichst schnelles Schalten zu ermöglichen (siehe Kapitel 3.4.3). Es konnten die Spannungsspitzen deutlich reduziert werden, allerdings wäre ein Austauschen der Komponenten im Falle eines Defekts aufgrund der freien Verdrahtung relativ umständlich.

### Funktionsmuster 3:



**Abbildung A.3** Das dritte Funktionsmuster. Dieser wurde wieder auf einer Lochrasterplatine aufgebaut, allerdings mit SMD-Komponenten und unter Berücksichtigung der Masseführung. Weiterhin wurde ein ADC eingebaut, um eine Messung und dadurch eine Regelung mithilfe des FPGAs zu ermöglichen.

Das dritte Funktionsmuster wurde wieder auf einer Lochrasterplatine aufgebaut, allerdings mit größtenteils SMD-Komponenten für den Leistungsteil sowie unter Berücksichtigung der Leitungslängen und möglichst sternförmiger Masseführung. Die Schaltung wurde um einen ADC erweitert, der über einen Spannungsteiler und einen Tiefpass die Ausgangsspannung misst, um eine Regelung zu ermöglichen. Zur Spannungsversorgung für den ADC wurde ein 3.3 V Spannungsregler eingebaut. Als Überspannungsschutz wurde an den Ausgang des Spannungswandlers eine Suppressordiode gesetzt. In diesem Funktionsmuster wurden auch mehrere Ausgangskondensatoren unterschiedlicher Bauarten parallel geschaltet, um das Verhalten zu untersuchen. Mithilfe dieses Funktionsmusters wurde erstmals ein externer Schrittmotortreiber mit einer aufwärts gewandelten Spannung versorgt, um die Reaktion des Aufwärtswandlers auf nicht rein resistive Lasten zu erproben. Mit diesem Funktionsmuster wurde die Regelung auf einen festen Sollwert mithilfe des ADC und des FPGAs getestet. Das dritte Funktionsmuster ist zu einem Großteil schon die Schaltung, die auf der Platine realisiert wird, zusätzlich kommen noch der FPGA und der Schrittmotortreiber inklusive der zum Betrieb notwendigen Bauteile hinzu.

## A.1 Funktionsmuster Schaltpläne

Die im Folgenden gezeigten Schaltpläne wurden mit *Autodesk EAGLE* erstellt und entsprechen den beschriebenen Funktionsmustern.

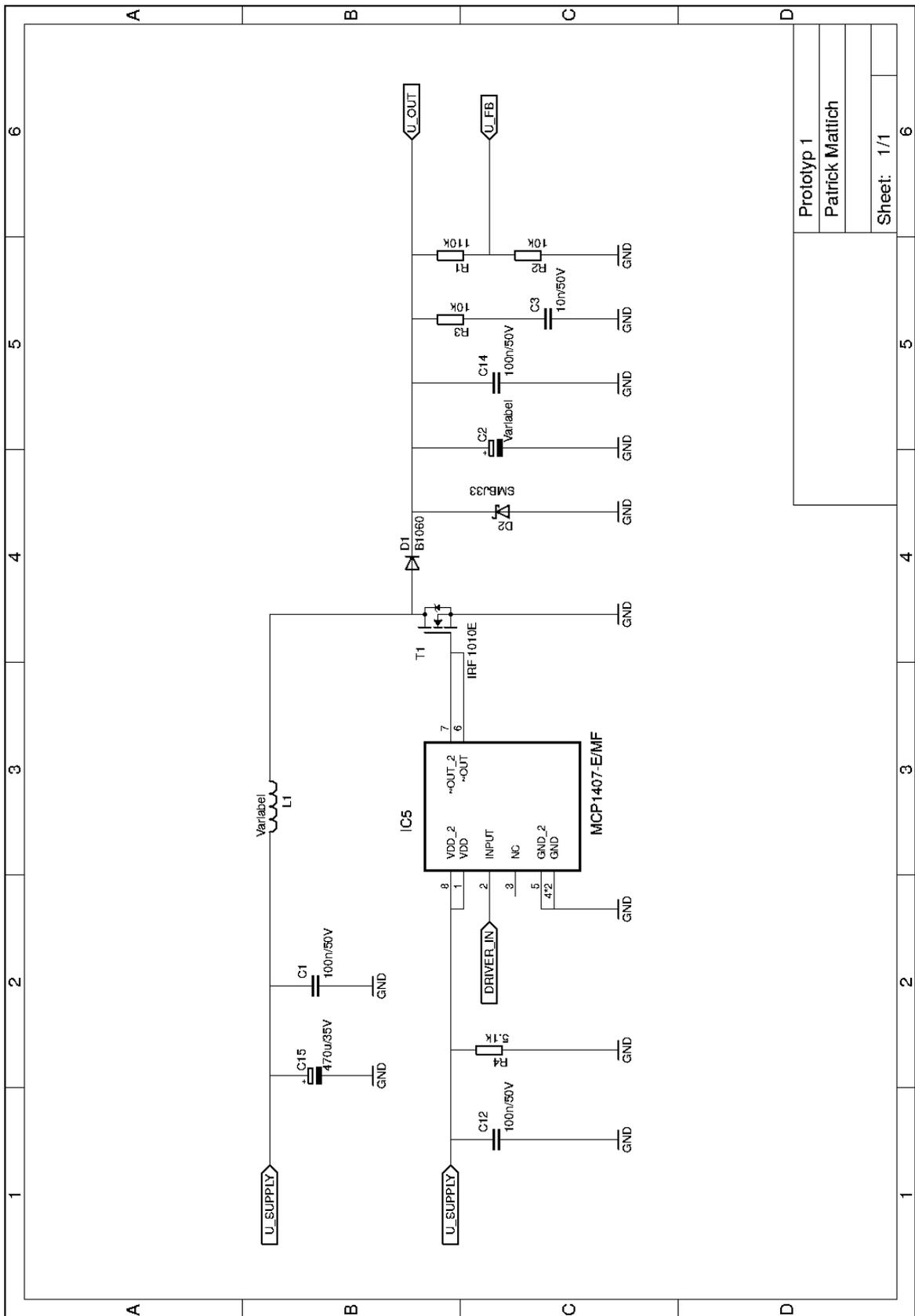


Abbildung A.4 Schaltplan des ersten Funktionsmusters

Prototyp 1	6
Patrick Matlich	6
Sheet: 1/1	6

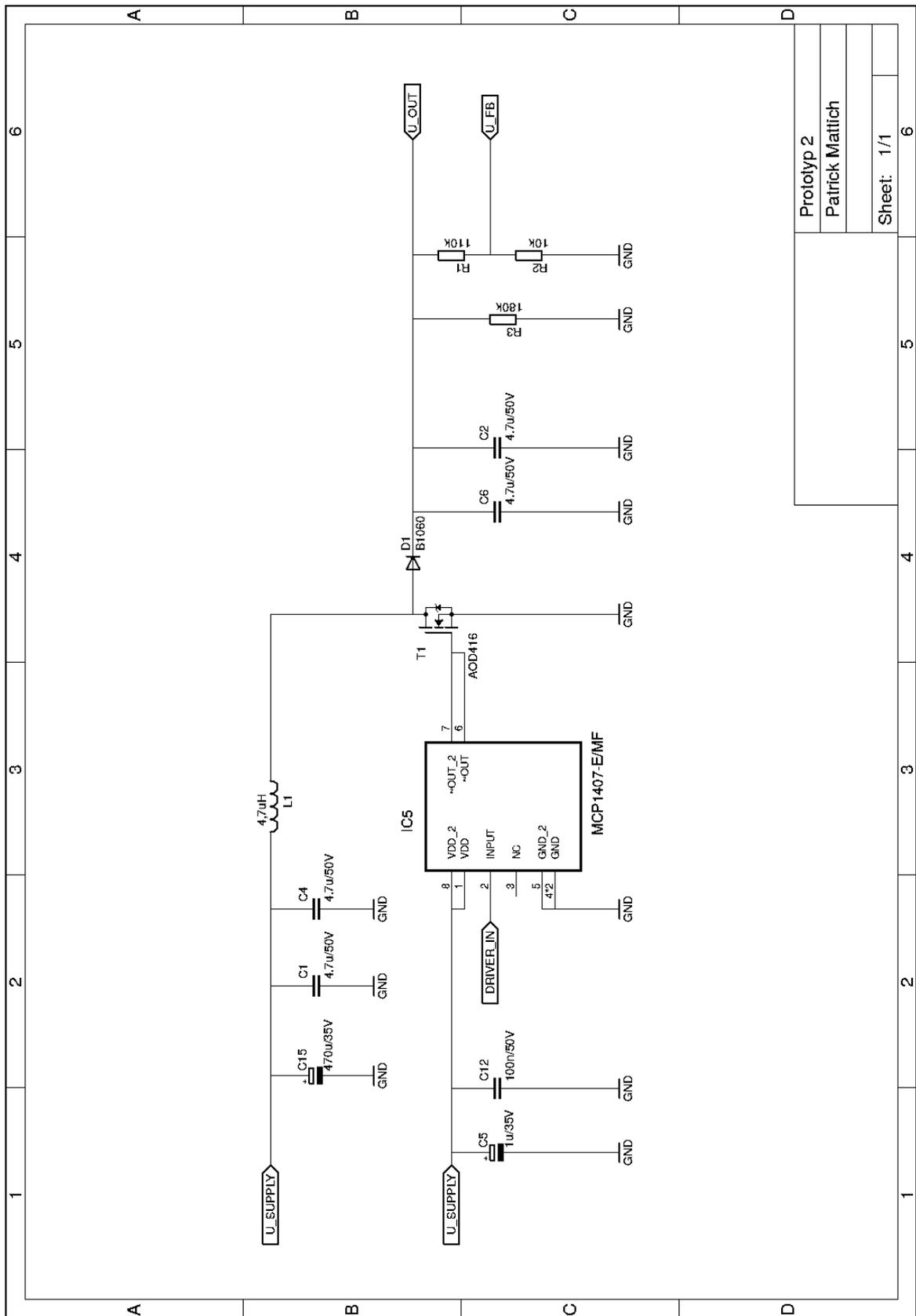


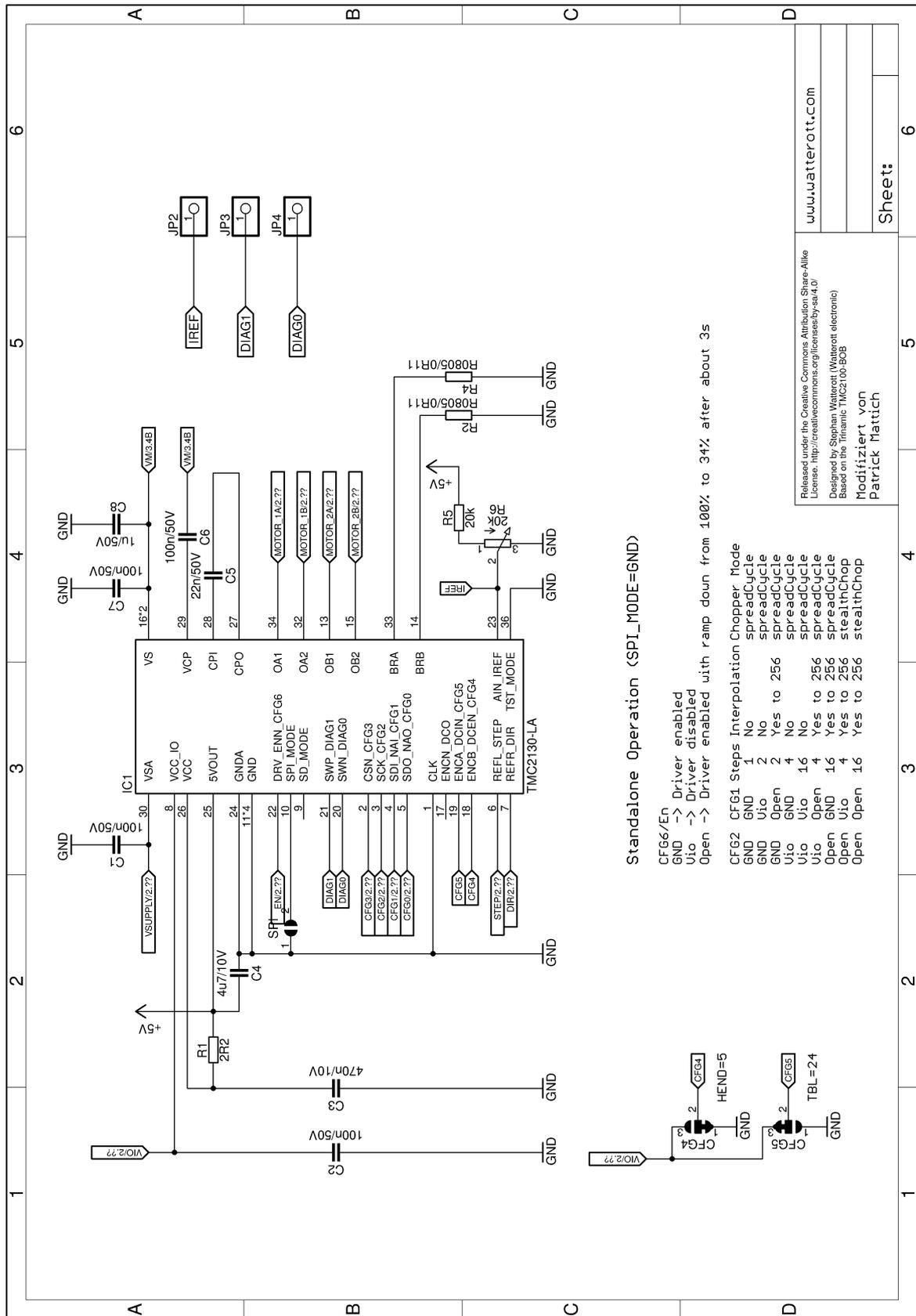
Abbildung A.5 Schaltplan des zweiten Funktionsmusters



# **B Aufbau der entwickelten Platine**

## **B.1 Schaltplan der Platine**

Die im Folgenden gezeigten Schaltpläne wurden mit *Autodesk EAGLE* erstellt und enthalten die Schaltungen des entwickelten Moduls.



Released under the Creative Commons Attribution Share-Alike License. <http://creativecommons.org/licenses/by-sa/4.0/>  
 Designed by Stephan Watterott (Watterott electronic)  
 Based on the Trinamic TMC2100-BOB  
 Modifiziert von Patrick Mattich  
[www.watterott.com](http://www.watterott.com)  
 Sheet: 6

Abbildung B.1 Kompletter Schaltplan des Schrittmotortreibers

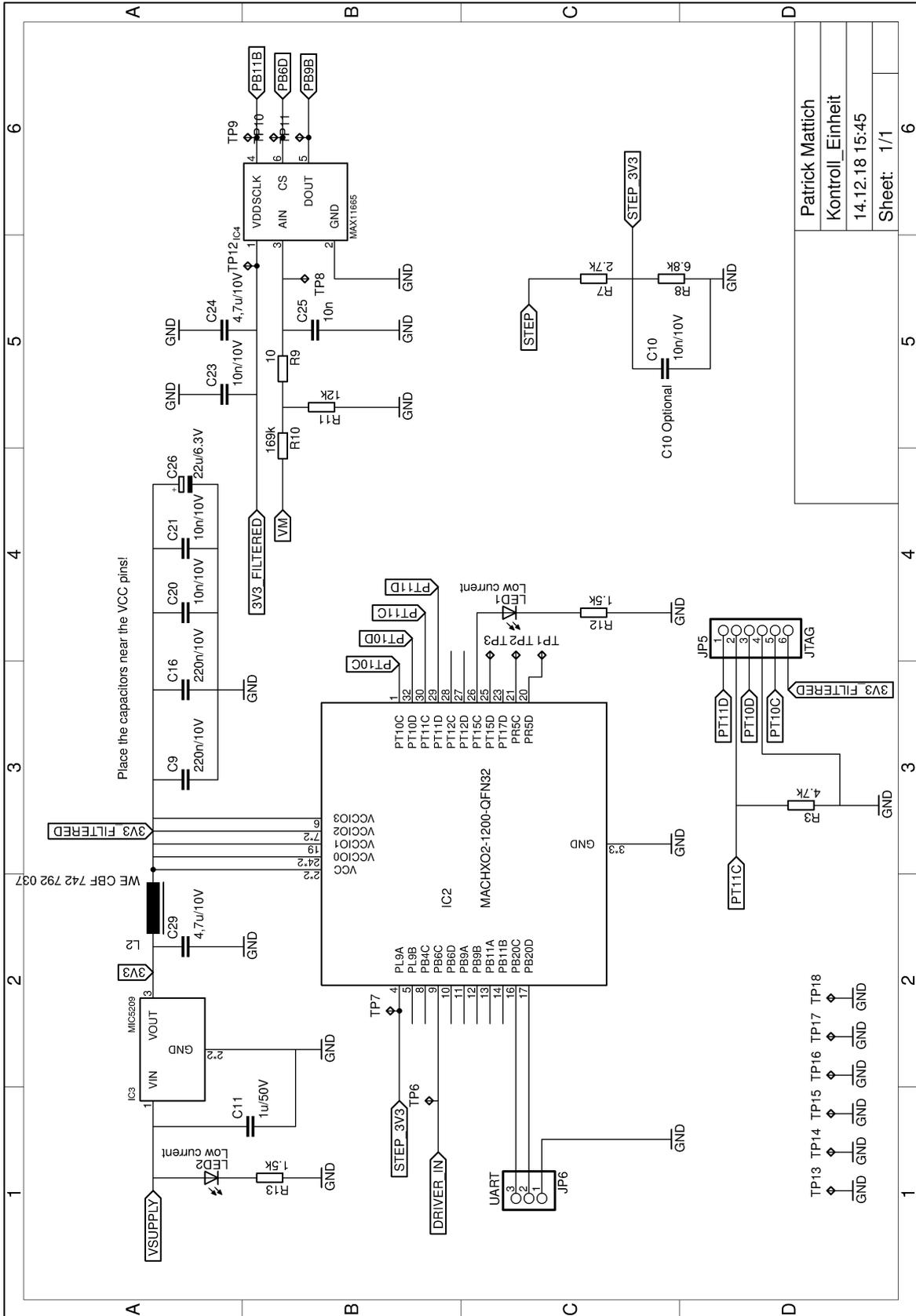


Abbildung B.2 Kompletter Schaltplan des FPGA- und ADC-Teils

Patrick Mattich  
 Kontroll\_Einheit  
 14.12.18 15:45  
 Sheet: 1/1

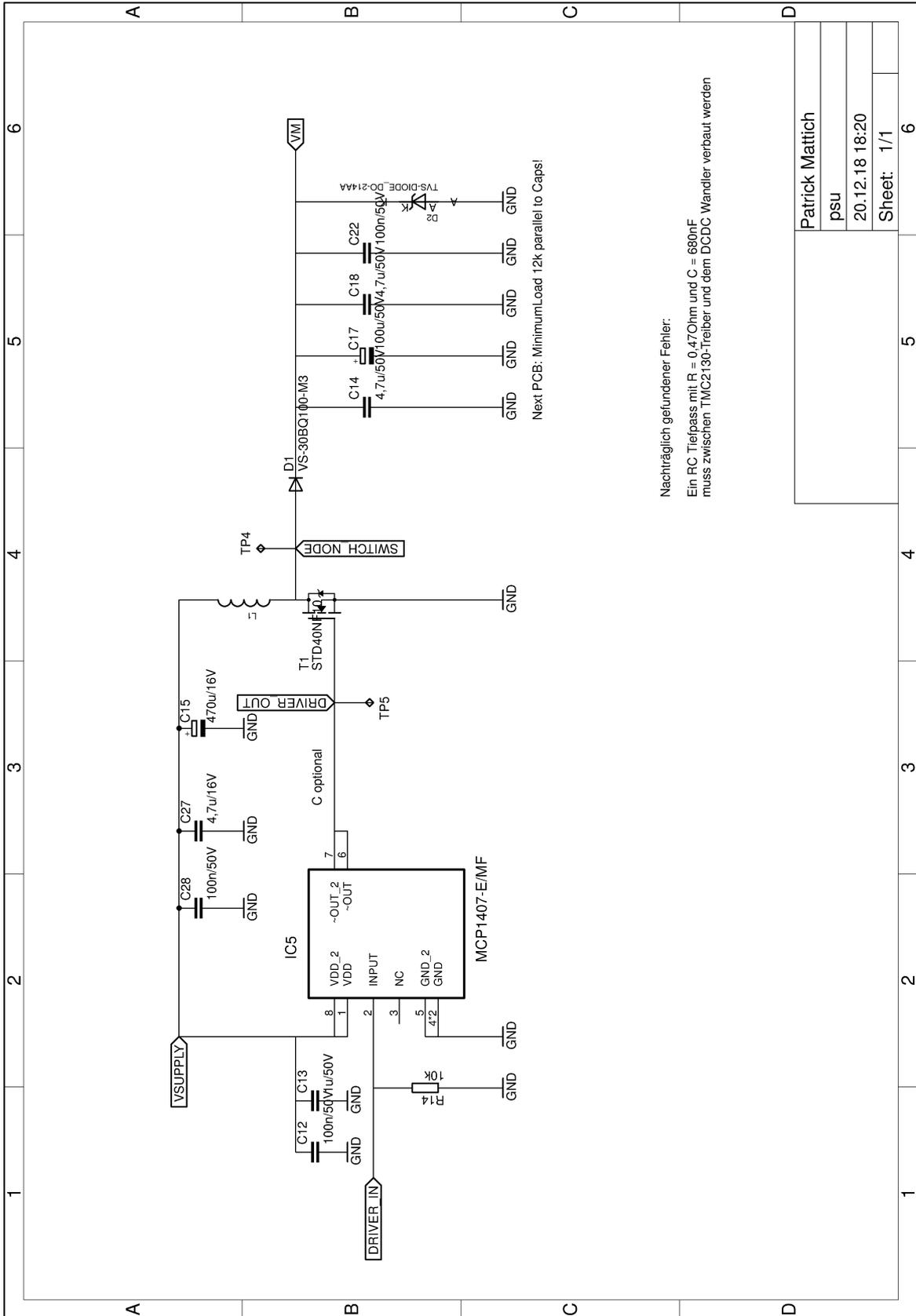
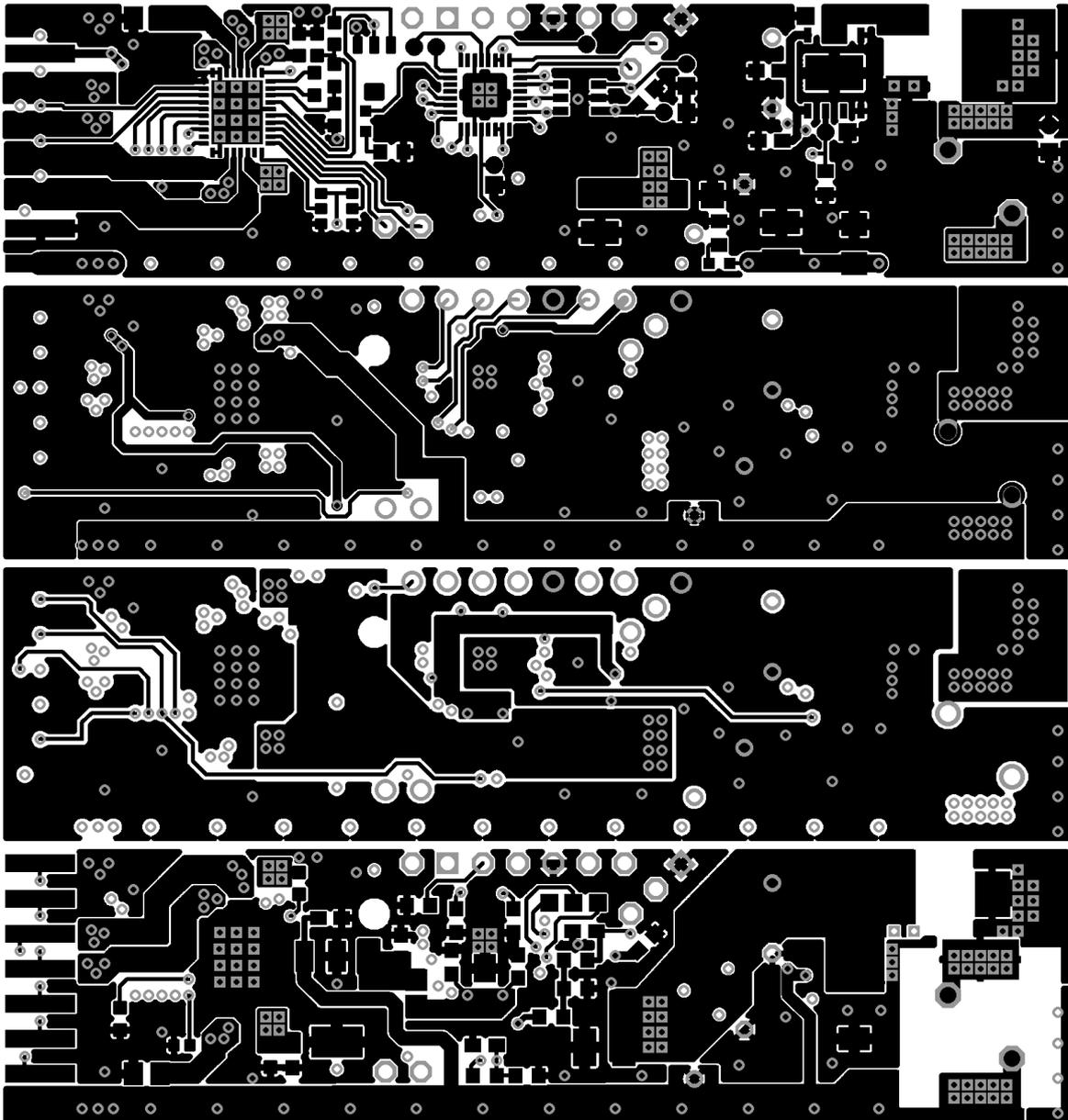


Abbildung B.3 Kompletter Schaltplan des Leistungsteils

## B.2 Layout der Platine

Da aufgrund der räumlichen Beschränkung zwei Lagen nicht ausgereicht haben, wurde eine vierlagige Platine entworfen. In Abbildung B.4 sind die vier Lagen einzeln dargestellt.



**Abbildung B.4** Die entwickelte Platine besteht aus vier Lagen, die hier von Lage 1 bis 4 dargestellt sind

## B.3 Aufbau des Adapters

Der Adapter ist nötig, um die längliche Platine in *StepStick*-Steckplätzen nutzen zu können. In Abbildung B.5 ist der Schaltplan des Adapters dargestellt, der eine rein passive Weiterleitung der Signale ermöglicht. In Abbildung B.6a und b sind die beiden Lagen der Platine dargestellt.

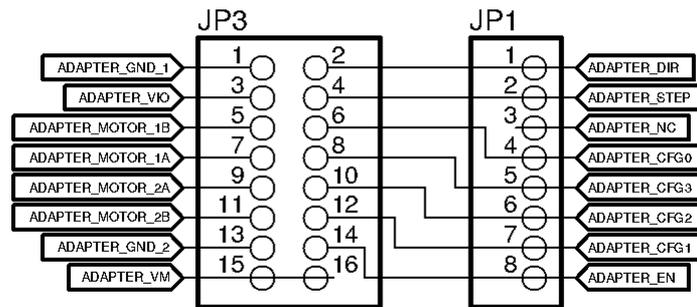
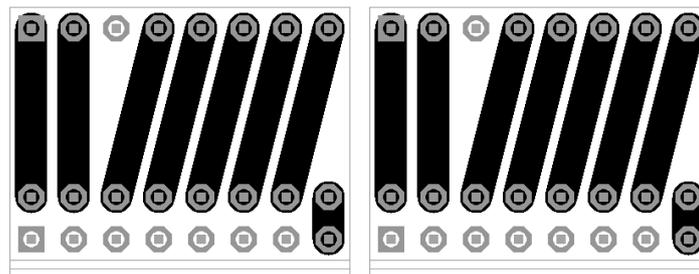


Abbildung B.5 Schaltplan des Adapters



(a) Top-Layer

(b) Bottom-Layer

Abbildung B.6 Beide Lagen die für den Adapter verwendet wurden. Die Leiterbahnen wurden auf beiden Seiten gedoppelt, um einen größeren Strom übertragen zu können.

## C Simulation der Zustandsautomaten

Abbildung C.1 zeigt den zeitlichen Ablauf der drei Zustandsautomaten während einer Simulation mit *ModelSim* von *Mentor Graphics*. Es sind zwei komplette Zyklen abgebildet, unter anderem erkennbar an den beiden Markierungen an der linken Seite. Durch die grünen Pfeile wird die Übernahme des ADC-Sollwerts (`ADC_SOLLWERT`) und des gemessenen ADC-Istwerts (`ADC_DATA`) aus dem ersten Zyklus in den zweiten Zyklus markiert. Diese beiden, in dem ersten Zyklus erzeugten, Werte werden während des zweiten Zyklus ausgewertet und bestimmen dadurch die Einschaltdauer des Transistors des Aufwärtswandlers. In der Abbildung sind weiterhin gut die einzelnen Zustände der drei Zustandsautomaten (`FSM_1_CURRENT_STATE`, `FSM_2_CURRENT_STATE`, `FSM_3_CURRENT_STATE`) zu erkennen sowie die Zeitpunkte, an denen die Zustände wechseln.

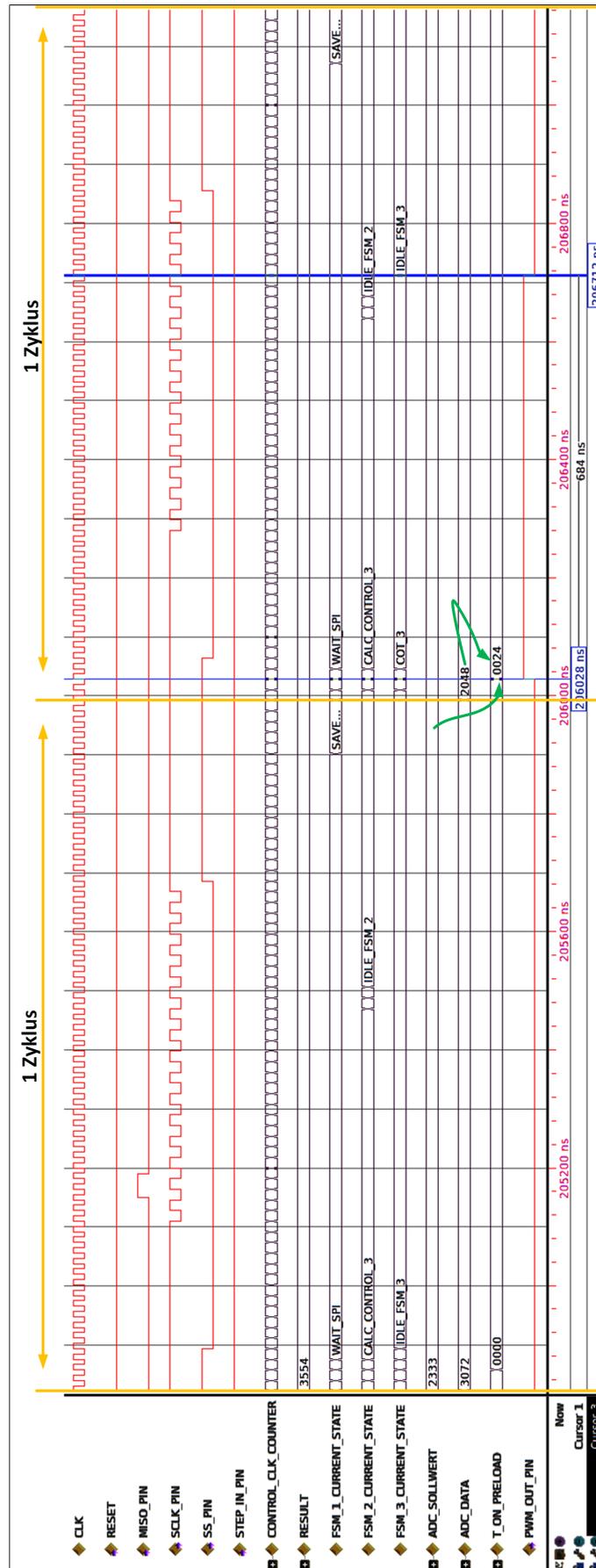


Abbildung C.1 Simulation der drei parallelisierten Zustandsautomaten

## D Firmware VHDL-Code

Der im Folgenden dargestellte VHDL-Code enthält die gesamte entwickelte Firmware. Der Code wurde mithilfe der *Lattice Diamond* Software von *Lattice Semiconductor Corporation* entwickelt und für den *XO2-1200* FPGA synthetisiert.

## Listing D.1 VHDL-Code der übergeordneten Main-Entity sowie der Zustandsautomaten und Zähler

```

1 library ieee;
2 use ieee.std_logic_1164.all;
3 use ieee.numeric_std.all;
4
5 ENTITY MAIN IS
6 PORT(
7   RESET      : in  std_logic;
8   MOSI_PIN   : out std_logic;
9   MISO_PIN   : in  std_logic;
10  SCLK_PIN   : out std_logic;
11  SS_PIN     : out std_logic;
12  UART_TX_PIN : out std_logic;
13  UART_RX_PIN : in  std_logic;
14  PWM_OUT_PIN : out std_logic;
15  ADC_MEASURE_PIN : out std_logic;
16  LED_PIN    : out std_logic;
17  STEP_IN_PIN : in  std_logic;
18  TEST_PIN_1 : out std_logic;
19  TEST_PIN_2 : in  std_logic;
20  TEST_PIN_3 : in  std_logic
21 );
22 END MAIN;
23
24 ARCHITECTURE ARCH of MAIN is
25
26 -- Konstante Parameter, die unabhängig von dem verwendeten Motor oder dem
27 -- Verhältnis von Spannung zu Drehzahl ist.
28 -- Diese Parameter sind für die entwickelte Hardware und die verwendeten
   Bauelemente
29 -- eingestellt
30 -----
31 constant ADC_SOLLWERT_FIX : unsigned(11 downto 0) := X"bb8";--must not be
   higher than MAX_ADC_VALUE!
32 constant T_SAMPLE : unsigned(19 downto 0) := X"00040";--every 1.20300752us
33 constant f_clk : unsigned(27 downto 0) := X"32bc480";-- 53.2 MHz
34 constant MAX_ADC_VALUE : unsigned(11 downto 0) := X"bb8";-- 3000 ca 36.62V
35 constant MIN_ADC_VALUE : unsigned(11 downto 0) := X"3d0";-- 983 ca 12V
36 -----
37
38 -- Die folgenden Parameter bestimmen von welcher minimal Drehzahl bis zu
   welcher
39 -- maximal Drehzahl die Spannung aufwärtsgewandelt wird.
40 -- Dafür sind zum einen die aus der Taktfrequenz und der entsprechenden
   Stepfrequenz
41 -- bestimmten Counter Werte wichtig. Zum anderen die beiden Parameter m und n
42 -- für die Geradengleichung um die lineare Abhängigkeit der Ausgangsspannung
43 -- von der gemessenen Stepfrequenz zu bestimmen.
44 -----
45 -- 1920 Hz bis 21331 Hz (40 rpm bis 400 rpm)
46 constant m : unsigned(31 downto 0) := X"0000006a";-- 106
47 constant n : unsigned(31 downto 0) := X"000c3df8";-- 802296
48 constant f_min_counter : unsigned(31 downto 0) := (X"00006c3c" );--f_clk/f_min
   53.2MHz/1920Hz = 27708
49 constant f_max_counter : unsigned(31 downto 0) := X"000009be";--f_clk/f_max
   53.2MHz/21331kHz = 2494
50 -----
51
52 -- Einstellung fuer die interne Takterzeugung des Lattice-FPGA
53 -----
54 COMPONENT OSCH
55 -- synthesis translate_off
56 GENERIC (NOM_FREQ : string := "53.20");--53.20MHz
57 -- synthesis translate_on
58 PORT (
59   STDBY_IN : in  std_logic;
60   OSC_OUT  : out std_logic;
61   SEDSTDBY_OUT : out std_logic);
62 END COMPONENT;
63 attribute NOM_FREQ : string;
64 attribute NOM_FREQ of OSCinst0 : label is "53.20";
65 signal CLK : std_logic;
66 signal STDBY_INT : STD_LOGIC := '0';
67 signal SEDSTDBY_INT : STD_LOGIC := '0';
68 -----
69
70 COMPONENT STEP_COUNTER IS
71 PORT(
72   CLK      : in  std_logic;
73   RESET   : in  std_logic;
74   STEP_IN  : in  std_logic;
75   RESULT_OUT : out std_logic_vector(19 downto 0);
76   FULL     : out std_logic;
77   FULL_OLD : out std_logic
78 );
79 END COMPONENT STEP_COUNTER;
80
81 signal RESULT, RESULT_OLD : std_logic_vector(19 downto 0);
82 signal STEP_COUNTER_FULL, STEP_COUNTER_FULL_OLD : std_logic;
83 -----
84
85 COMPONENT Divider is
86 Generic ( b : natural range 4 to 32 := 28 ); -- Breite
87 Port ( start : in  STD_LOGIC;
88   divisor : in  STD_LOGIC_VECTOR (b-1 downto 0);
89   dividend : in  STD_LOGIC_VECTOR (b-1 downto 0);
90   quotient : out STD_LOGIC_VECTOR (b-1 downto 0);
91   remainder : out STD_LOGIC_VECTOR (b-1 downto 0);
92   busy : out STD_LOGIC;
93   CLK : in  STD_LOGIC;
94   RESET : in  STD_LOGIC);
95 end COMPONENT Divider;
96
97 signal start, busy : std_logic := '0';
98 signal divisor, dividend, quotient, remainder : STD_LOGIC_VECTOR(27 downto 0)
   := (others => '0');
99 -----
100
101 component SPI_Master is
102 Generic ( Quartz_Taktfrequenz : integer := 53200000; -- Hertz
103   SPI_Taktfrequenz : integer := 25000000; -- Hertz / zur Berechnung
   des Reload-Werts fuer Takteiler
104   Pre_Delay : integer := 1; -- us / Zeit nach
   Aktivieren von CS bis Beginn der Übertragung
105   Post_Delay : integer := 1; -- us / Zeit nach
   Beenden der Übertragung bis Deaktivieren CS
106   Laenge : integer := 16 -- Anzahl der zu
   übertragenden Bits
107 );
108 Port ( TX_Data : in  STD_LOGIC_VECTOR (Laenge-1 downto 0); -- Sendedaten

```

```

109         RX_Data : out STD_LOGIC_VECTOR (Laenge-1 downto 0); -- Empfangsdaten
110         CPHA   : in  STD_LOGIC; -- Clock Phase
111         CPOL   : in  STD_LOGIC; -- Clock
112             Polarity
113         MOSI   : out STD_LOGIC;
114         MISO   : in  STD_LOGIC;
115         SCLK   : out STD_LOGIC;
116         SS     : out STD_LOGIC;
117         Start_TX : in  STD_LOGIC;
118         TX_Done : out STD_LOGIC;
119         CLK     : in  STD_LOGIC;
120         RESET  : in  STD_LOGIC;
121         BUSY_FLAG: out STD_LOGIC
122     );
123 end component SPI_Master;
124 signal TX_Data_SPI: STD_LOGIC_VECTOR (16-1 downto 0) := "1010100110101001";--
    dummy value
125 signal RX_Data: STD_LOGIC_VECTOR (16-1 downto 0);
126 signal Start_TX: std_logic;
127 signal TX_Done_SPI: std_logic;
128 signal CPHA: std_logic := '1';
129 signal CPOL: std_logic := '1';
130 -----
131 type FSM_1_STATES is ( IDLE_FSM_1,
132     START_SPI,
133     STOP_SPI,
134     WAIT_SPI,
135     SAVE_ADC_DATA
136 );
137
138 signal FSM_1_CURRENT_STATE : FSM_1_STATES;
139 -----
140 type FSM_2_STATES is ( IDLE_FSM_2,
141     CALC_CONTROL_1,
142     CALC_CONTROL_2,
143     CALC_CONTROL_3,
144     CALC_CONTROL_4,
145     CALC_CONTROL_5
146 );
147
148 signal FSM_2_CURRENT_STATE : FSM_2_STATES;
149 -----
150 type FSM_3_STATES is ( IDLE_FSM_3,
151     COT_1,
152     COT_2,
153     COT_3
154 );
155
156 signal FSM_3_CURRENT_STATE : FSM_3_STATES;
157 -----
158
159 signal CONTROL_CLK_COUNTER: unsigned(19 downto 0);
160 signal ADC_DATA: unsigned(11 downto 0);
161 signal ADC_SOLLWERT, ADC_SOLLWERT_FSM_3: unsigned(11 downto 0);
162 signal FX: unsigned(59 downto 0);
163 signal SET_T_ON_PRELOAD: std_logic;
164 signal T_ON_PRELOAD, T_ON_COUNTER: unsigned ( 15 downto 0);
165 signal CALC_FLAG: std_logic;
166 signal LED: std_logic;

```

```

168 -----
169 begin
170 -----
171 OSCInst0: OSCH
172 -- synthesis translate_off
173 GENERIC MAP( NOM_FREQ => "53.20" )
174 -- synthesis translate_on
175 PORT MAP (
176     STDBY=> STDBY_INT,
177     -- OSC => CLK,
178     SEDSTDBY => SEDSTDBY_INT
179 );
180 -----
181
182 SPI_Master_1: SPI_Master PORT MAP (
183     TX_Data => TX_Data_SPI,
184     RX_Data => RX_Data,
185     CPHA   => CPHA,
186     CPOL   => CPOL,
187     MOSI   => MOSI_PIN,
188     MISO   => MISO_PIN,
189     SCLK   => SCLK_PIN,
190     SS     => SS_PIN,
191     Start_TX => Start_TX,
192     TX_Done => TX_Done_SPI,
193     CLK     => CLK,
194     RESET  => RESET,
195     BUSY_FLAG => ADC_MEASURE_PIN
196 );
197 -----
198
199 STEP_COUNTER_1: STEP_COUNTER PORT MAP (
200     CLK => CLK,
201     reset => RESET,
202     STEP_IN => STEP_IN_PIN,
203     RESULT_OUT => RESULT,
204     FULL => STEP_COUNTER_FULL,
205     FULL_OLD => STEP_COUNTER_FULL_OLD
206 );
207 -----
208
209 Divider_1: Divider PORT MAP (
210     start => start,
211     divisor => divisor,
212     dividend => dividend,
213     quotient => quotient,
214     remainder => remainder,
215     busy => busy,
216     CLK => CLK,
217     RESET => RESET
218 );
219 -----
220
221 -- Die CONTROL_CLK bestimmt die Sample Rate und ist der Ausloser
222 -- fur die drei Zustandsautomaten die zusammen arbeiten
223 -----
224 CONTROL_CLK: process (CLK,RESET) --every 1.2us
225 begin
226     if RESET = '1' then
227         CONTROL_CLK_COUNTER<= T_SAMPLE;
228     elsif CLK = '1' and CLK'event then

```

```

229     if CONTROL_CLK_COUNTER = 0 then
230         CONTROL_CLK_COUNTER<= T_SAMPLE;--preload
231     else
232         CONTROL_CLK_COUNTER <= CONTROL_CLK_COUNTER - 1;
233     end if;
234 end if;
235 end process CONTROL_CLK;
236 -----
237
238 -- Der Zustandsautomat FSM_1 steuert den Ablauf der SPI Kommunikation
239 -- mit dem ADC. Der neue Messwert wird am Ende des CONTROL_CLK_COUNTER
240 -- freigegeben damit dieser für den nächsten Zyklus zur Verfügung steht.
241 -----
242 FSM_1: process(RESET, CLK)
243 begin
244     if RESET = '1' then
245         FSM_1_CURRENT_STATE <= IDLE_FSM_1;
246         Start_TX <= '0';
247
248     elsif CLK = '1' and CLK'event then
249
250         case FSM_1_CURRENT_STATE is
251             when IDLE_FSM_1 =>
252                 if CONTROL_CLK_COUNTER = 0 then
253                     FSM_1_CURRENT_STATE <= START_SPI;
254                 end if;
255             -----
256             when START_SPI =>
257                 Start_TX <= '1';
258                 FSM_1_CURRENT_STATE <= STOP_SPI;
259             -----
260             when STOP_SPI =>
261                 Start_TX <= '0';
262                 FSM_1_CURRENT_STATE <= WAIT_SPI;
263             -----
264             when WAIT_SPI =>
265                 if TX_Done_SPI = '1' then --adc ausgelesen
266                     FSM_1_CURRENT_STATE <= SAVE_ADC_DATA;
267                 end if;
268             -----
269             when SAVE_ADC_DATA =>
270                 if CONTROL_CLK_COUNTER = 1 then --cycle nearly complete
271
272                     --ADC_DATA <= unsigned(RX_Data(14 downto 3));-- For MAX11665 (12Bit
273                     ADC 500ks)
274                     ADC_DATA <= unsigned( RX_Data(13 downto 6)  & "0000");--For MAX11116
275                     (8Bit ADC 3MS)
276
277                     FSM_1_CURRENT_STATE <= IDLE_FSM_1;
278                 end if;
279             -----
280             when others =>
281                 end case;
282         end if;
283     end process FSM_1;
284 -----
285
286 -- Der Zustandsautomat FSM_2 steuert die Berechnung des ADC Sollwertes
287 -- aus den gemessenen Step Frequenzen.
288 -- In Zustand CALC_CONTROL_5 kann durch einkommentieren eingestellt werden,
289 -- ob anstatt einer Drehzahlabhangigen Sollwertberechnung ein fixer

```

```

288 -- Sollwert genutzt werden soll. Dadurch wurde der DCDC Wandler
289 -- eine konstante Ausgangsspannung erzeugen, unabhängig von der
290 -- Drehzahl des Motors.
291 -----
292 FSM_2: process(RESET, CLK)
293 begin
294     if RESET = '1' then
295         FSM_2_CURRENT_STATE <= IDLE_FSM_2;
296         ADC_SOLLWERT <= X"008";
297         divisor <= (others=>'0');
298         dividend <= (others=>'0');
299         start <= '0';
300         FX <= (others=>'0');
301         CALC_FLAG <= '0';
302
303     elsif CLK = '1' and CLK'event then
304
305         case FSM_2_CURRENT_STATE is
306             when IDLE_FSM_2 =>
307
308                 if CONTROL_CLK_COUNTER = 0 then
309                     FSM_2_CURRENT_STATE <= CALC_CONTROL_1;
310                 end if;
311             -----
312             when CALC_CONTROL_1 =>
313
314                 if busy = '0' then
315                     if ("000000000000" & unsigned(RESULT)) > f_min_counter then
316                         --OR (STEP_COUNTER_FULL_OLD = '1') OR (STEP_COUNTER_FULL = '1')
317                             then
318                                 ADC_SOLLWERT <= (MIN_ADC_VALUE-100);
319                                 divisor <= "00000000" & x"1111";--dummy value
320                                 dividend <= STD_LOGIC_VECTOR(f_clk);
321                                 start <= '1';
322                                 LED <= '0';
323
324                                 CALC_FLAG <= '1';
325                                 FSM_2_CURRENT_STATE <= CALC_CONTROL_2;
326
327                             elsif ("000000000000" & unsigned(RESULT)) < f_max_counter then
328                                 ADC_SOLLWERT <= MAX_ADC_VALUE;
329                                 divisor <= "00000000" & x"fff0";--dummy value
330                                 dividend <= STD_LOGIC_VECTOR(f_clk);
331                                 start <= '1';
332                                 LED <= '1';
333
334                                 CALC_FLAG <= '1';
335                                 FSM_2_CURRENT_STATE <= CALC_CONTROL_2;
336
337                             else
338                                 divisor <= "00000000" & RESULT;
339                                 dividend <= STD_LOGIC_VECTOR(f_clk);
340                                 start <= '1';
341                                 LED <= '1';
342
343                                 CALC_FLAG <= '0';
344                                 FSM_2_CURRENT_STATE <= CALC_CONTROL_2;
345
346                             end if;
347                         else

```

```

348     start <= '0';
349     end if;
350 -----
351 when CALC_CONTROL_2 =>
352
353     start <= '0';
354     FSM_2_CURRENT_STATE <= CALC_CONTROL_3;
355 -----
356 when CALC_CONTROL_3 =>
357
358     if busy = '0' then
359         --Geradengleichung:
360         FX <= m*unsigned(quotient) - n;
361
362         FSM_2_CURRENT_STATE <= CALC_CONTROL_4;
363     end if;
364 -----
365 when CALC_CONTROL_4 =>
366
367     if CALC_FLAG = '0' then
368         ADC_SOLLWERT <= FX(21 downto 10);-- division by 2^10
369     else
370         ADC_SOLLWERT <= ADC_SOLLWERT;
371     end if;
372     FSM_2_CURRENT_STATE <= CALC_CONTROL_5;
373 -----
374 when CALC_CONTROL_5 =>
375
376     if ADC_SOLLWERT >= MAX_ADC_VALUE then
377         ADC_SOLLWERT <= MAX_ADC_VALUE;
378     elsif ADC_SOLLWERT <= MIN_ADC_VALUE then
379
380         ADC_SOLLWERT <= (MIN_ADC_VALUE-100);
381     else
382         ADC_SOLLWERT <= ADC_SOLLWERT;
383     end if;
384
385     --FIXED MODE:
386     -- if TEST_PIN_2 = '1' then
387     -- --ADC_SOLLWERT <= x"000";
388     -- --ADC_SOLLWERT <= ADC_SOLLWERT_FIX;
389     -- else
390     -- -- ADC_SOLLWERT <= ADC_SOLLWERT;
391     -- end if;
392     FSM_2_CURRENT_STATE <= IDLE_FSM_2;
393 -----
394
395     when others =>
396         end case;
397     end if;
398 end process FSM_2;
399 -----
400
401 -- Der Zustandsautomat FSM_3 steuert die Schaltzeiten des
402 -- Transistors des DCDC Wandlers.
403 -- Er beinhaltet den Constant-On-Time Algorithmus sowie
404 -- eine Softstart Funktion.
405 -- Die Softstart Funktion kann deaktiviert werden, indem
406 -- die entsprechende Sektion auskommentiert wird und
407 -- Nicht-Softstart Sektion daruber einkommentiert wird.
408 -----

```

```

409 FSM_3: process(RESET, CLK)
410 begin
411     if RESET = '1' then
412         FSM_3_CURRENT_STATE <= IDLE_FSM_3;
413
414     elsif CLK = '1' and CLK'event then
415
416         case FSM_3_CURRENT_STATE is
417             when IDLE_FSM_3 =>
418
419                 if CONTROL_CLK_COUNTER = 0 then
420                     ADC_SOLLWERT_FSM_3 <= ADC_SOLLWERT;
421                     FSM_3_CURRENT_STATE <= COT_1;
422                 end if;
423
424             -----
425             when COT_1 =>
426                 -- Der COT Algorithmus arbeitet so, dass der Transistor nur
427                 -- dann geschaltet wird, wenn die Erzeugte Spannung unter der
428                 -- Sollspannung liegt
429                 if ADC_DATA < ADC_SOLLWERT_FSM_3 then
430
431                     SET_T_ON_PRELOAD <= '1';
432                     PWM_OUT_PIN <= '1';
433
434                     -----
435                     --Kein Softstart:
436                     --T_ON_PRELOAD <= x"0030"; -- 902ns [green]
437                     -----
438
439                     -----
440                     --Softstart:
441                     if ADC_DATA > 2800 then -- 34.17V
442                         T_ON_PRELOAD <= x"002C"; -- 827ns
443
444                     elsif ADC_DATA > 2700 then -- 32.96V
445                         T_ON_PRELOAD <= x"002B"; -- 808ns
446
447                     elsif ADC_DATA > 2600 then --31.74V
448                         T_ON_PRELOAD <= x"002A"; -- 789ns
449
450                     elsif ADC_DATA > 2500 then -- 30.52V
451                         T_ON_PRELOAD <= x"0029"; -- 771ns
452
453                     elsif ADC_DATA > 2400 then -- 29.30V
454                         T_ON_PRELOAD <= x"0028"; -- 752ns
455
456                     elsif ADC_DATA > 2300 then -- 28.08V
457                         T_ON_PRELOAD <= x"0027"; -- 733ns
458
459                     elsif ADC_DATA > 2200 then -- 26.86V
460                         T_ON_PRELOAD <= x"0026"; -- 714ns
461
462                     elsif ADC_DATA > 2100 then -- 25.63V
463                         T_ON_PRELOAD <= x"0025"; -- 695ns
464
465                     elsif ADC_DATA > 2000 then -- 24.41V
466                         T_ON_PRELOAD <= x"0024"; -- 677ns
467
468                     elsif ADC_DATA > 1900 then -- 23.19V
469

```

```

470         T_ON_PRELOAD <= X"0022"; -- 639ns
471
472     elsif ADC_DATA > 1800 then -- 21.97V
473         T_ON_PRELOAD <= X"0020"; -- 602ns
474
475     elsif ADC_DATA > 1700 then -- 20.75V
476         T_ON_PRELOAD <= X"001E"; -- 564ns
477
478     elsif ADC_DATA > 1600 then -- 19.53V
479         T_ON_PRELOAD <= X"001C"; -- 526ns
480
481     elsif ADC_DATA > 1500 then -- 18.31V
482         T_ON_PRELOAD <= X"001A"; -- 489ns
483
484     elsif ADC_DATA > 1400 then -- 17.09V
485         T_ON_PRELOAD <= X"0018"; -- 451ns
486
487     elsif ADC_DATA > 1300 then -- 15.87V
488         T_ON_PRELOAD <= X"0016"; -- 414ns
489
490     elsif ADC_DATA > 1200 then -- 14.65V
491         T_ON_PRELOAD <= X"0014"; -- 376ns
492
493     elsif ADC_DATA > 1100 then -- 13.43V
494         T_ON_PRELOAD <= X"0012"; -- 338ns
495
496     else -- < 13.43V
497         T_ON_PRELOAD <= X"0010"; -- 301ns
498
499     end if;
500     -----
501
502     if ADC_SOLLWERT_FSM_3 = X"000" then
503         PWM_OUT_PIN <= '0';
504     end if;
505
506     -- Wenn der gemessene Spannungswert über dem Sollwert liegt,
507     -- wird der Transistor nicht geschaltet
508     else
509         PWM_OUT_PIN <= '0';
510         T_ON_PRELOAD <= X"0000";
511         SET_T_ON_PRELOAD <= '1';
512     end if;
513
514     FSM_3_CURRENT_STATE <= COT_2;
515     -----
516     when COT_2 =>
517
518         SET_T_ON_PRELOAD <= '0';
519         FSM_3_CURRENT_STATE <= COT_3;
520     -----
521     when COT_3 =>
522
523         if T_ON_COUNTER = 0 then
524             PWM_OUT_PIN <= '0';
525             FSM_3_CURRENT_STATE <= IDLE_FSM_3;
526         end if;
527     -----
528     when others =>
529         end case;
530 end if;

```

```

531 end process FSM_3;
532 -----
533
534 -- Der T_ON_COUNTER_PROCESS wird von dem Zustandsautomaten FSM_3
535 -- eingestellt und gestartet.
536 -- Er bemisst die Zeit, die der Transistor des DCDC Wandlers
537 -- eingeschaltet wird.
538 -----
539 T_ON_COUNTER_PROCESS: process(CLK,RESET)
540 begin
541     if RESET = '1' then
542         T_ON_COUNTER <= x"0000";
543     elsif CLK = '1' and CLK'event then
544
545         if SET_T_ON_PRELOAD = '1' then
546             T_ON_COUNTER <= T_ON_PRELOAD;
547
548         elsif T_ON_COUNTER = 0 then
549
550             else
551                 T_ON_COUNTER <= T_ON_COUNTER - 1;
552             end if;
553         end if;
554     end process T_ON_COUNTER_PROCESS;
555     -----
556     LED_PIN <= LED;
557     -----
558 END ARCHITECTURE ARCH;

```

### Listing D.2 VHDL-Code der Teilereinheit

```

1 -- Den VHDL Code für die Division wurde von Lothar Miller geschrieben
2 -- und auf seiner Homepage veröffentlicht:
3 -- http://www.lothar-miller.de/s9y/archives/29-Division-in-VHDL.html
4 -- Veröffentlicht am 5. Februar 2009
5
6
7 library IEEE;
8 use IEEE.STD_LOGIC_1164.ALL;
9 use IEEE.NUMERIC_STD.ALL;
10
11 entity Divider is
12     Generic ( b : natural range 4 to 32 := 32 ); -- Breite
13     Port ( start : in STD_LOGIC;
14           divisor : in STD_LOGIC_VECTOR (b-1 downto 0);
15           dividend : in STD_LOGIC_VECTOR (b-1 downto 0);
16           quotient : out STD_LOGIC_VECTOR (b-1 downto 0);
17           remainder : out STD_LOGIC_VECTOR (b-1 downto 0);
18           busy : out STD_LOGIC;
19           clk : in STD_LOGIC;
20           RESET : in STD_LOGIC
21         );
22 end Divider;
23
24 architecture Behave_Unsigned of Divider is
25     signal dd : unsigned(b-1 downto 0); -- dividend
26     signal dr : unsigned(b-1 downto 0); -- divisor
27     signal q : unsigned(b-1 downto 0); -- quotient
28     signal r : unsigned(b-1 downto 0); -- remainder

```

```

29 signal bits : integer range b downto 0;
30 type zustaende is (idle, prepare, shift, sub, done);
31 signal z : zustaende;
32
33 begin
34
35     process(RESET, clk)
36     variable diff : unsigned(b-1 downto 0);
37     begin
38         if RESET = '1' then
39             dd <= (others=>'0');
40             dr <= (others=>'0');
41             q <= (others=>'0');
42             r <= (others=>'0');
43             bits <= 0;
44             z <= idle;
45             busy <= '0';
46
47         elsif clk = '1' and clk'event then
48             case z is
49                 when idle =>
50                     if (start='1') then
51                         z <= prepare;
52                         busy <= '1';
53                     else
54                         busy <= '0';
55                     end if;
56                     dd <= unsigned(dividend);
57                     dr <= unsigned(divisor);
58
59                     when prepare =>
60                         q <= (others=>'0');
61                         r <= (others=>'0');
62                         z <= shift;
63                         bits <= b;
64                         -- Sonderfall: Division durch Null
65                         if (dr=0) then
66                             q <= (others=>'1');
67                             r <= (others=>'1');
68                             z <= done;
69                         -- Sonderfall: Divisor groesser als Dividend
70                         elsif (dr>dd) then
71                             r <= dd;
72                             z <= done;
73                         -- Sonderfall: Divisor gleich Dividend
74                         elsif (dr=dd) then
75                             q <= to_unsigned(1,b);
76                             z <= done;
77                         end if;
78
79                     when shift =>
80                         -- erst mal die beiden Operanden
81                         -- fuer die Subtraktion zurechtrucken
82                         if ( r(b-2 downto 0)&dd(b-1) < dr ) then
83                             bits <= bits-1;
84                             r <= r(b-2 downto 0)&dd(b-1);
85                             dd <= dd(b-2 downto 0)&'0';
86                         else
87                             z <= sub;
88                         end if;
89

```

```

90         when sub =>
91             if (bits>0) then
92                 r <= r(b-2 downto 0)&dd(b-1);
93                 dd <= dd(b-2 downto 0)&'0';
94                 -- Rest minus Divisor
95                 diff := (r(b-2 downto 0)&dd(b-1)) - dr;
96                 if (diff(b-1)='0') then
97                     -- wenn kein Unterlauf
98                     --> Divisor passt noch rein
99                     --> MSB=0 --> 1 in Ergebnis einschieben
100                    q <= q(b-2 downto 0) & '1';
101                    r <= diff;
102                else
103                    -- wenn Unterlauf
104                    --> 0 einschieben, mit altem Wert weiterrechnen
105                    q <= q(b-2 downto 0) & '0';
106                end if;
107                bits <= bits-1;
108            else
109                z <= done;
110            end if;
111
112         when done =>
113             busy <= '0';
114             -- Handshake: wenn noetig warten, bis start='0'
115             if (start='0') then
116                 z <= idle;
117             end if;
118         end case;
119     end if;
120 end process;
121
122 quotient <= std_logic_vector(q);
123 remainder <= std_logic_vector(r);
124 end;

```

### Listing D.3 VHDL-Code des SPI-Masters

```

1 -- Den VHDL Code fur einen Konfigurierbarer SPI-Master wurde von Lothar Miller
   geschrieben
2 -- und auf seiner Homepage veröffentlicht:
3 -- http://www.lothar-miller.de/s9y/archives/51-Konfigurierbarer-SPI-Master.html
4 -- Veröffentlicht am 12. Juli 2009
5 --
6 -- Er wurde um eine asynchronen Reset erweitert.
7 -- Weiterhin wurde ein BUSY Flag hinzugefügt, wodurch erkennbar ist,
8 -- ob die Übertragung noch aktiv ist, oder ob die Daten angekommen und
9 -- auswertbar sind.
10
11 library IEEE;
12 use IEEE.STD_LOGIC_1164.ALL;
13 use IEEE.NUMERIC_STD.ALL;
14
15 entity SPI_Master is
16     Generic ( Quartz_Taktfrequenz : integer := 1000000; -- Hertz
17             SPI_Taktfrequenz : integer := 1000000; -- Hertz / zur
18                 Berechnung des Reload-Werts fuer Taktteiler
19             Pre_Delay : integer := 1; -- us / Zeit nach
20                 Aktivieren von CS bis Beginn der uebertragung

```

```

19         Post_Delay      : integer := 1;    -- us / Zeit nach
                Beenden der uebertragung bis Deaktivieren CS
20         Laenge         : integer := 8    -- Anzahl der zu
                uebertragenden Bits
21     );
22     Port ( TX_Data : in STD_LOGIC_VECTOR (Laenge-1 downto 0); -- Sendedaten
23           RX_Data : out STD_LOGIC_VECTOR (Laenge-1 downto 0); -- Empfangsdaten
24           CPHA    : in STD_LOGIC;        -- Clock
                Phase
25           CPOL    : in STD_LOGIC;        -- Clock
                Polarity
26           MOSI    : out STD_LOGIC;
27           MISO    : in STD_LOGIC;
28           SCLK    : out STD_LOGIC;
29           SS      : out STD_LOGIC;
30           Start_TX : in STD_LOGIC;
31           TX_Done  : out STD_LOGIC;
32           clk     : in STD_LOGIC;
33           RESET   : in STD_LOGIC;
34           BUSY_FLAG : out STD_LOGIC
35     );
36 end SPI_Master;
37
38 architecture Behavioral of SPI_Master is
39     signal delay      : integer range 0 to (Quarz_Taktfrequenz/(2*
                SPI_Taktfrequenz));
40     signal delay_pre  : integer range 0 to 10;
41     signal delay_post : integer range 0 to 10;
42     constant clock_delay : integer := (Quarz_Taktfrequenz/(2*SPI_Taktfrequenz))-1;
43
44     type spitx_states is (spi_stx,del_pre,spi_txactive,del_post,spi_etx);
45     signal spitxstate : spitx_states := spi_stx;
46
47     type spi_clkstates is (shift,sample);
48     signal spiclkstate : spi_clkstates;
49
50     signal bitcounter : integer range 0 to Laenge; -- wenn bitcounter = Laenge
                --> alle Bits uebertragen
51     signal tx_reg     : std_logic_vector(Laenge-1 downto 0) := (others=>'0');
52     signal rx_reg     : std_logic_vector(Laenge-1 downto 0) := (others=>'0');
53
54 begin
55     ----- Verwaltung -----
56     process(CLK, RESET)
57     begin
58
59         if RESET = '1' then
60             --TX_Done <= '1';
61             spitxstate <= spi_stx;
62
63         elsif clk = '1' and clk'event then
64
65             MOSI <= tx_reg(tx_reg'left);
66             delay_post <= 10; -- POST-Delay wg OPTO-Koppler
67             delay_pre <= 10; -- Initial-Delay wg OPTO-Koppler
68             --
69             -- delay_post <= (Post_Delay*(Quarz_Taktfrequenz/1000))/1000; -- POST-
                Delay wg OPTO-Koppler
70             -- delay_pre <= (Pre_Delay *(Quarz_Taktfrequenz/1000))/1000; -- Initial-
                Delay wg OPTO-Koppler
71             if(delay>0) then delay <= delay-1;

```

```

72         else
73             delay <= clock_delay;
74         end if;
75     case spitxstate is
76     when spi_stx =>
77         SS <= '1'; -- slave select disabled
78         TX_Done <= '0';
79         bitcounter <= Laenge;
80         SCLK <= CPOL;
81         if(Start_TX = '1') then
82             --TX_Done <= '0';
83             BUSY_FLAG <= '1';
84             spitxstate <= del_pre;
85         else
86             BUSY_FLAG <= '0';
87         end if;
88
89         when del_pre =>
90             -- SS aktivieren und Zeit fuer
                Optokoppler abwarten
91             SS <= '0';
92             SCLK <= CPOL;
93             if (CPHA='0') then spiclkstate <= sample; -- sample at odd SCLK-
                edge (1st, 3rd, 5th...)
94             else
95                 spiclkstate <= shift; -- sample at even SCLK-
                edge (2nd, 4th, 6th...)
96             end if;
97             delay <= 0;
98             if (delay_pre>0) then
99                 delay_pre <= delay_pre-1;
100             else
101                 spitxstate <= spi_txactive;
102             end if;
103
104         when spi_txactive => -- Daten aus tx_reg uebertragen
                ----- SPI-Takt generieren
105         -----
106         case spiclkstate is
107         when sample =>
108             SCLK <= (CPOL xor CPHA);
109             if (delay=0) then -- sample
110                 spiclkstate <= shift;
111                 if(CPHA='1') then bitcounter <= bitcounter-1; end if;
112             end if;
113
114         when shift =>
115             SCLK <= not (CPOL xor CPHA);
116             if (delay=0) then -- shift
117                 spiclkstate <= sample;
118                 if(CPHA='0') then bitcounter <= bitcounter-1; end if;
119             end if;
120         end case;
121         if (delay=0 and bitcounter=0) then -- alle Bits uebertragen ->
                deselektieren
122             SCLK <= CPOL;
123             spitxstate <= del_post;
124             end if;
125         ----- SPI-Takt fertig
126         -----
127         when del_post =>
128             SS <= '1'; -- disable Slave Select
129             if (delay_post>0) then
130                 delay_post <= delay_post-1;

```

```

127         else
128             spitxstate <= spi_etx;
129         end if;
130
131         when spi_etx =>
132             TX_Done <= '1';
133             BUSY_FLAG <= '0';
134             if (Start_TX = '0') then -- Handshake: warten, bis Start-Flag
135                 geloescht
136                 spitxstate <= spi_stx;
137             end if;
138         end case;
139     end if;
140 end process;
141
142 ---- Schieberegister in eigenem Prozess ist ressourcensparend -----
143 process begin
144     wait until rising_edge(clk);
145 --   if (spitxstate=del_pre) then -- Initialisierung weglassen spart
146 --       Ressourcen: 1 Mux = 10 Slices
147 --       rx_reg <= (others=>'0');
148 --   end if;
149   if (spitxstate=spi_txactive and spiclstate=sample and delay=0 and
150       bitcounter/=0) then
151       rx_reg <= rx_reg(rx_reg'left-1 downto 0) & MISO;
152   end if;
153   if (spitxstate=spi_stx) then
154       tx_reg <= TX_Data;
155   end if;
156   if (spitxstate=spi_txactive and spiclstate=shift and delay=0 and (cpha='0'
157       or bitcounter/=Laenge)) then
158       tx_reg <= tx_reg(tx_reg'left-1 downto 0) & tx_reg(0);
159   end if;
160 end process;
161
162 RX_Data <= rx_reg;
163 end Behavioral;

```

Listing D.4 VHDL-Code des Frequenzzählers

```

1 library ieee;
2 use ieee.std_logic_1164.all;
3 use ieee.numeric_std.all;
4
5 -- Der STEP_COUNTER zahlt, in CLK Zyklen, wie weit
6 -- zwei Step Signale auseinander liegen.
7 -- Daraus kann dann die Frequenz des Signals berechnet
8 -- werden. Diese wiederum ergibt dann die zu generierende
9 -- Ausgangsspannung.
10 ENTITY STEP_COUNTER IS
11 PORT
12 (
13     CLK          : in std_logic;
14     RESET        : in std_logic;
15     STEP_IN      : in std_logic;
16     RESULT_OUT   : out std_logic_vector(19 downto 0);

```

```

17     FULL          : out std_logic;
18     FULL_OLD      : out std_logic
19 );
20 END STEP_COUNTER;
21
22 ARCHITECTURE ARCH of STEP_COUNTER is
23
24     signal COUNTER: unsigned(19 downto 0);
25     signal RESTART_COUNTER: std_logic;
26     signal STEP_IN_INTERNAL: std_logic;
27     signal RESULT: unsigned(19 downto 0);
28     signal SIGNAL_NEW, SIGNAL_OLD, SIGNAL_RISING: std_logic;
29     signal FULL_INTERN: std_logic;
30
31 begin
32
33     STEP_IN_INTERNAL <= STEP_IN;
34
35     process (CLK, RESET)
36     begin
37         if RESET = '1' then
38             COUNTER <= (others => '0');
39             FULL <= '0';
40             RESULT_OUT <= (others => '0');
41             FULL_INTERN <= '0';
42             FULL_OLD <= '0';
43
44         elsif CLK = '1' and CLK'event then
45
46             -- Counter Register hat einen bestimmten Wert erreicht,
47             -- der einer minimalen Step Signal Frequenz entspricht,
48             -- unterhalb der die Spannung nicht aufwartsgewandelt werden soll.
49             -- Ist dieser Zustand erreicht, wird der aktuelle Counter Wert
50             -- sowie das FULL Flag gesetzt.
51             if COUNTER = X"a21c0" then --664000
52                 -- Bei einer Taktfrequenz von 53.2MHz entspricht
53                 -- ein Zahlerstand von 664000 einer Frequenz von
54                 -- 80Hz
55                 -- Wenn der Schrittmotor so langsam drehen sollte,
56                 -- wird das FULL Flag gesetzt.
57                 RESULT_OUT <= std_logic_vector( COUNTER );
58                 FULL <= '1';
59                 FULL_INTERN <= '1';
60
61             -- Hat der Counter den eingestellten maximal Wert noch nicht erreicht,
62             -- wird er einfach weiter hochgezahlt und das FULL Flag auf
63             -- LOW gesetzt.
64         else
65             COUNTER <= COUNTER + 1;
66             FULL <= '0';
67             FULL_INTERN <= '0';
68         end if;
69
70         -- Sobald eine neue Flanke eines Step Signals erkannt wird,
71         -- wird der Counter zurueckgesetzt und der erreichte Counter Wert
72         -- ausgegeben.
73         if SIGNAL_RISING = '1' then
74
75             FULL_OLD <= FULL_INTERN;
76             RESULT_OUT <= std_logic_vector( COUNTER );
77             COUNTER <= (others => '0');

```

```
78     FULL <= '0';
79     FULL_INTERN <= '0';
80     end if;
81
82     end if;
83     end process;
84
85     -- Der folgende Prozess sowie die nebenlaufige Zuweisung
86     -- ueberpruefen das anliegende Step Signal auf steigende
87     -- Flanken um den Counter auswerten und neu starten
88     -- zu koennen
89     SIGNAL_RISING <= SIGNAL_NEW and not SIGNAL_OLD;
90     process (CLK)
```

```
91     begin
92         if RESET = '1' then
93             SIGNAL_NEW <= '0';
94             SIGNAL_OLD <= '0';
95         elsif rising_edge(CLK) then
96
97             SIGNAL_OLD <= SIGNAL_NEW;
98             SIGNAL_NEW <= STEP_IN_INTERNAL;
99         end if;
100     end process;
101
102 END ARCHITECTURE ARCH;
```

# E Inhalt der CD

**Auf der beigefügten CD befindet sich:**

- Eine PDF-Datei dieser Arbeit
- Der VHDL-Quellcode
- Verwendete Datenblätter und andere Quellen als PDF-Dateien
- Funktionsmuster Schaltpläne
- Modul Schaltplan und Boarddesign
- Adapter Schaltplan und Boarddesign
- Messergebnisse in tabellarischer Form

# Versicherung über die Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §22(6) PO\_Ma\_MeS ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen habe ich unter Angabe der Quellen kenntlich gemacht.

Hamburg, 10. Januar 2019

---

Ort, Datum

---

Unterschrift