



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorthesis

Felix Karsten

Entwicklung eines intelligenten
Energiemanagementsystems für medizinische
Notfallgeräte

Felix Karsten
Entwicklung eines intelligenten
Energiemanagementsystems für medizinische
Notfallgeräte

Bachelorthesis eingereicht im Rahmen der Bachelorprüfung
im Studiengang Informations- und Elektrotechnik
am Department Informations- und Elektrotechnik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. -Ing. Karl-Ragmar Riemschneider
Zweitgutachter: Prof. Dr. -Ing. Jörg Dahlkemper

Abgegeben am 20. September 2016

Felix Karsten

Thema der Bachelorthesis

Entwicklung eines intelligenten Energiemanagementsystems für medizinische Notfallgeräte.

Stichworte

Elektrisches Energiemanagement, mikrocontrollergesteuerte DC-Wandler, Regelung von Energieflüssen, Energierückgewinnung eines Verbrauchers, Stromregelung von einem Akkulader .

Kurzzusammenfassung

Diese Arbeit befasst sich mit der Entwicklung eines softwaregesteuerten Energiemanagementsystems für medizinische Geräte. Diese reicht von der Konzeptionierung bis hin zu einer Machbarkeitsstudie an einer Versuchshardware. Es wird sich mit der Energieflussteuerung zwischen einer zentralen Versorgungsquelle, einem Akkumulator und den Verbrauchern befasst.

Felix Karsten

Title of the paper

Development of an intelligent Energymanagementsystem for medical devices.

Keywords

Electrical Energymanagement, microprocessor controlled DC-Converter, measurement and control of energy flow, energy recovery of a consumer, current control of a battery charger .

Abstract

This report deals with the construction of a software controlled Energymanagementsystem for medical devices. It ranges from conception to feasibility study of experimental hardware. It is dealt with the energy flow control between a central supply source , an battery and consumers

Danksagung

Die vorliegende Bachelorarbeit wurde in der Firma Weinmann Emergency Medical Technology GmbH + Co. KG erstellt. Ich möchte mich bei der Firma für die Möglichkeit bedanken. Meinem Betreuer bei Weinmann, Billy Schwalbe, gilt mein besonderer Dank. Ich wusste die stets hilfsbereite Unterstützung sehr zu schätzen.

Herrn Prof. Dr.-Ing Karl-Ragnar Riemschneider, spreche ich meinen herzlichen Dank für die Betreuung und die Erstkorrektur aus.

Meinen Eltern, Bärbel Karsten und Stefan Karsten sowie meiner Großmutter, Karin Finnern möchte ich mich für die ständige Unterstützung im Studium bedanken.

Meinen Korrekturlesern möchte ich herzlich Danken.

Inhaltsverzeichnis

Tabellenverzeichnis	7
Abbildungsverzeichnis	8
1 Einleitung	10
2 Grundlagen	13
2.1 Beatmung	13
2.2 Ideale Dioden	15
2.3 Schaltregler	15
2.4 Verlustleistung - Wirkungsgrad	17
2.5 PI - Regler	19
3 Anforderungen	22
4 Analyse	25
4.1 Stand der Technik	25
4.2 Herangehensweise	30
4.3 Ansätze	32
4.4 Gegenüberstellung	40
4.5 Konzeptentscheidung	41
5 Softwareentwicklung	42
5.1 Systemanforderung	42
5.2 Regelstrategie	43
5.3 Modellbildung	46
5.4 Simulation	51
5.5 Implementierung	52
6 Hardwareentwicklung	53
6.1 Dimensionierung	53
6.2 Umsetzung	59

7	Verifikation und Diskussion	61
7.1	Versorgungsquellen	61
7.2	Konstante Energieversorgung des Beatmungsgebläses	68
7.3	Akku Laden und Mischbetrieb	70
7.4	Speichern von zurückgewonnener Bremsenergie	77
7.5	Einschaltverhalten	79
7.6	Prozessorauslastung des Reglersystems	83
7.7	Wirkungsgrad	84
7.8	Leckstrom	88
7.9	Diskussion	89
8	Bewertung	91
8.1	Fazit	91
8.2	Ausblick	92
	Literaturverzeichnis	93
A	Simulation	94
A.1	Modell	94
A.2	Reglermodell	96
A.3	Simulationsergebnis	96
B	Hardware	98
C	Quellcode	104
C.1	main	104
	Glossar	125

Tabellenverzeichnis

3.1	Anforderungen	24
4.1	Gegenüberstellung der Hardwarekonzepte	40
7.1	Abschaltschwellen der Spannungsquellen	62
7.2	Auswertung der Anforderungen	90

Abbildungsverzeichnis

1.1	Energiefluss und Signalfluss des softwaregesteuerten Energiemanagementsystems für ein Beatmungsgerät	11
2.1	Lastsprung der Gebläseelektronik	14
2.2	Grundschialtung des synchronen Boost-Wandlers	16
2.3	Geschlossenes Regelungssystem eines PI-Reglers	21
4.1	Vorhandenes Energiemanagementsystem	25
4.2	Energieflussdiagramm der Beatmungselektronik	28
4.3	V-Modell für die Hardwareentwicklung des Energiemanagementsystems	30
4.4	V-Modell für die Softwareentwicklung	31
4.5	Konzept 1 - Mischbetrieb mit Akkulader ohne geschaltete Quellenversorgung	34
4.6	Konzept 2 - Buck-Boost-Wandler für den Eingang des Bordnetzes. Akku parallel zum Ausgang des Buck-Boost-Wandlers	36
4.7	Konzept 3 - Buck-Boost-Wandler für den Eingang des Akkus. Bordnetz parallel zum Ausgang des Buck-Boost-Wandlers	37
4.8	Konzept 4 - Boost-Wandler für die Eingangsquellen mit geregelter Zwischen- spannung	38
5.1	Reglerkonzept	44
5.2	Modell der Regelstrecke	47
5.3	Simulink-Modell der Eingangskapazität eines Boost-Wandlers	48
5.4	Simulink-Modell der Drosselspule eines Boost-Wandlers	49
5.5	Simulink-Modell der Ausgangskapazität eines Boost-Wandlers	49
5.6	Simulink-Modell des idealen Boost-Wandlers	50
5.7	Simulink Modell des Idealen Boost Wandlers	52
6.1	Induktivität des Boost-Wandlers über dem Ripplestrom bei veränderlichen Ausgangsspannungen	55
6.2	Einschaltelektronik für die Strombegrenzung der Elektrolytkondensatoren	57
6.3	Versuchshardware	60
7.1	Blockschaltbild des Messaufbaus	62
7.2	Blockschaltbild zur Messung der Strombegrenzung	63

7.3	Strombegrenzung des Bordnetzes	64
7.4	Blockschaltbild zu den Messungen des Quellenverhaltens im Zusammenspiel	65
7.5	Wechsel vom Akkubetrieb in den Mischbetrieb	66
7.6	Wechsel vom Mischbetrieb in den Akkubetrieb	67
7.7	Blockschaltbild zu den Messungen der Lastsprünge	68
7.8	Lastsprung bei der kleinsten Akku-Eingangleistung	69
7.9	Blockschaltbild zu den Messungen - Akkuladen und Mischbetrieb	70
7.10	Untere Anschaltgrenze des Akkuladers	71
7.11	Abschaltgrenze bei Erreichen der Ladeschlussspannung	72
7.12	Regelgrenze 2 Ampere laden	74
7.13	Spannungsgrenze 0 Ampere laden	75
7.14	Spannungsgrenze 2 Ampere unterstützen	76
7.15	Blockschaltbild zu den Messungen - Energierückgewinnung	77
7.16	Spannungssprung bei 10 Joule Energierückgewinnung	78
7.17	Blockschaltbild zu den Messungen - Einschaltverhalten	79
7.18	Einschaltverhalten mit Bordnetz und Akku	80
7.19	Einschaltverhalten im Akkubetrieb	81
7.20	Ausgleichströme	82
7.21	Prozessorauslastung der Regelschleife	83
7.22	Blockschaltbild zur Messung des Wirkungsgrades eines Boost-Wandlers	84
7.23	Wirkungsgrad eines Boost-Wandlers	85
7.24	Blockschaltbild zur Messung des Wirkungsgrades für den Akkuladepfad	86
7.25	Wirkungsgrad des Akkuladepfades über 2 Boost-Wandler	87
7.26	Blockschaltbild zur Messung des Leckstromes aus dem Akku während des ausgeschalteten Zustandes	88
A.1	Matlab-Simulink-Modell der Regelstrecke und des Reglers	95
A.2	Matlab-Simulink-Modell des implementierten Reglers	96
A.3	Simulation der Regelstrecke und des Reglers	97
B.1	Schaltplan - Sheet Entry	99
B.2	Schaltplan Leistungselektronik	100
B.3	Schaltplan Boost-Wandler	101
B.4	Schaltplan Kleinspannungsversorgung/Mikrocontroller	102
B.5	Platinenlayout	103

1 Einleitung

Medizinische Notfallgeräte spielen eine große Rolle in der Erstversorgung am Unfallort. Entscheidend ist die Betriebsbereitschaft der Geräte und damit verbunden ein sorgfältiger und effizienter Umgang mit den elektrischen Ressourcen.

Als Beispiel seien hier Beatmungsgeräte genannt. Stand der Technik bei mobilen Beatmungsgeräten sind heutzutage druckgasbetriebene Geräte mit elektronisch gestützten Komponenten. Aufgrund der hohen Anforderungen an die Mobilität und einer nötigen Infrastruktur für die Versorgung mit Sauerstoff geht der Trend in Richtung rein elektrisch betriebener Beatmung. Hierbei wird der Beatmungsdruck nicht der Sauerstoffflasche entnommen, sondern durch eine elektrisch getriebene Druckquelle zur Verfügung gestellt. Sauerstoff dient somit nicht mehr zusätzlich als Energielieferant, sondern nur noch als Medikament und kann beigemischt werden.

Die Energieversorgung übernimmt, wie in Abbildung 1.1 zu sehen, entweder das Bordnetz eines Krankenwagens, eines Rettungshubschraubers oder ein Akkumulator (im Folgenden kurz Akku genannt). Aufgrund des erhöhten elektrischen Energiebedarfs gilt dem Energiemanagementsystem ein großes Augenmerk. Es muss die Energie zwischen Bordnetz, Akku und dem Beatmungsgebläse geeignet aufteilen. Diese Aufgaben werden üblicherweise in Analogtechnik zuverlässig gelöst, jedoch lässt dies das Energiemanagementsystem schnell unflexibel und teuer werden.

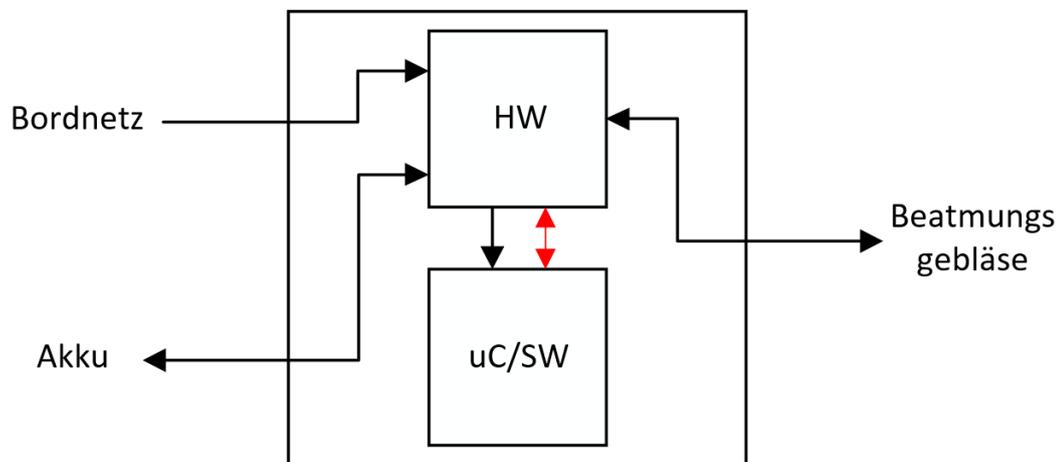


Abbildung 1.1: Energiefluss und Signalfluss des softwaregesteuerten Energiemanagementsystems für ein Beatmungsgerät (Rot: Signalfluss, Schwarz: Energiefluss)

Aufgabenstellung

Ziel dieser Arbeit ist es, eine Machbarkeitsstudie eines softwaregesteuerten Energiemanagementsystems durchzuführen. Es wird eine Kozeptionierung für die Hardware und Software erarbeitet. Darauf aufbauend wird eine Versuchshardware entwickelt.

Es wird überprüft, ob sich ein softwaregesteuertes Energiemanagementsystem mit vertretbarem Aufwand in Bezug auf Regelgeschwindigkeit und Hardware realisieren lässt. Zu den Aufgaben des Energiemanagements zählen das Aufteilen der Energie zwischen Akku und Bordnetz, das Laden des Akkus zu ermöglichen, eine Kleinspannungsversorgung der Controller und Sensoren sowie das Bereitstellen von Energie für einen Leistungsverbraucher. Die Versorgung des Aktors (Antriebs-elemente) am Beispiel eines Beatmungsgebläses stellt eine besondere Herausforderung dar, da die benötigte Energie starken Schwankungen ausgesetzt ist. Die Beatmungselektronik kennzeichnet sich durch einen hohen Leistungsverbrauch und einer Energierückgewinnung aus.

Ein softwaregesteuertes Energiemanagementsystem verspricht eine Optimierung an Komplexität und Flexibilität. Durch die Zentralisierung der Steuerung soll eine Reduktion der Bauteile erreicht werden sowie eine Erhöhung der Flexibilität in Bezug auf die Übertragbarkeit des Energiemanagements auf andere Geräte. Als Anwendungsfälle kommen Beatmungsgeräte wie auch Defibrillatoren in Frage.

Struktur

Die Grundlagen der in dieser Arbeit behandelten Themen werden in Kapitel 2 erarbeitet und vorgestellt.

In Kapitel 3 wird für die Aufgabenstellung ein konkretes Anforderungsprofil an das Energiemanagement vorgestellt.

Kapitel 4 beschreibt die Analyse . Es wird der Stand der Technik behandelt und die Herangehensweise an die Entwicklung vorgestellt. Darauf aufbauend werden Ansätze für ein Lösungskonzept aufgezeigt. Nach einer Gegenüberstellung wird das vielversprechendste Konzept ausgewählt.

Kapitel 5 beschreibt die Softwareentwicklung für die Steuerung/Regelung der Hardware. Es wird eine Regelstrategie beschrieben und ein Modell der Hardware in Matlab-Simulink abgebildet. Zudem wird das entwickelte C-Programm vorgestellt.

In Kapitel 6 wird die Hardwareentwicklung beschrieben. Auf eine Dimensionierung der Bauteile folgt die Erstellung eines Schaltplans und ein Platinenlayout.

In Kapitel 7 wird die Verifikation und Diskussion des Versuchsaufbaus behandelt.

Die Bewertung der Arbeit wird in Kapitel 8 vorgenommen, indem ein Fazit gezogen und ein Ausblick gegeben wird.

2 Grundlagen

2.1 Beatmung

Das Energiemanagement soll am Beispiel eines Notfallbeatmungsgerätes entwickelt werden. Es müssen dementsprechend die Eigenschaften einer Beatmungselektronik als Verbraucher berücksichtigt werden.

Die Beatmungselektronik steuert ein Radialgebläse an, welches in der Lunge eines Patienten einen Umgebungsdruck erzeugt. Die Beatmung setzt sich aus 4 Phasen zusammen, welche im Folgenden näher beschrieben werden:

1. Einleitung der Inspiration (Einatmen). Das Gebläse wird auf hohe Leistung eingeregelt, damit der gewünschte Beatmungsdruck schnell erreicht werden kann.
2. Inspiration: Der gewünschte Druck wurde eingeregelt und die Lunge füllt sich mit Atemluft. Das Gebläse arbeitet auf mittlerer Leistung, um den Beatmungsdruck aufrecht erhalten zu können.
3. Einleitung der Expiration (Ausatmen): Das Gebläse wird mit hoher Bremsleistung betrieben, um den geräteausgangsseitigen Druck abzubauen und eine Expiration durch das Expirationsventil zuzulassen.
4. Expiration: Das Gebläse wird auf einen Teillastbetrieb eingeregelt. Es wird ein sogenannter PEEP (Positive End Expiratory Pressure) eingestellt, um einen minimalen Beatmungsdruck aufrechtzuerhalten.

Bei näherem Interesse in der Beatmungstechnik sei auf die Quelle [1] verwiesen.

Abbildung 2.1 zeigt zwei Beatmungsphasen mit den zugehörigen Leistungsparametern der Gebläseelektronik.

Für das zu entwickelnde Energiemanagementsystem ist die Leistung (powerMot), welche die Beatmungselektronik dem System abverlangt, entscheidend. Denn die Beatmungselektronik regelt die Spannung für das Gebläse selbstständig ein.

Die Phasen der Beatmung aus energetischer Sicht stellen sich wie folgt dar:

1. 40 Watt - kurzfristig abverlangte Leistung
2. 33 Watt - konstante abverlangte Leistung in der Inspirationsphase
3. -45 Watt - Energierückgewinnung während des Bremsvorganges ¹
4. 10 Watt - kontinuierlich abverlangte Leistung in der Expirationsphase

Es wird davon ausgegangen, dass die kurzfristig abverlangte Energie in der Einleitung der Inspirationsphase der Bremsenergie entspricht. Mittels Energierückgewinnung kann die Bremsenergie im Dauerbetrieb die Energie für die Einleitung der Inspirationsphase liefern.

Das Energiemanagementsystem muss dementsprechend eine Energierückgewinnung bieten und eine konstante Leistung von 33 Watt liefern können.

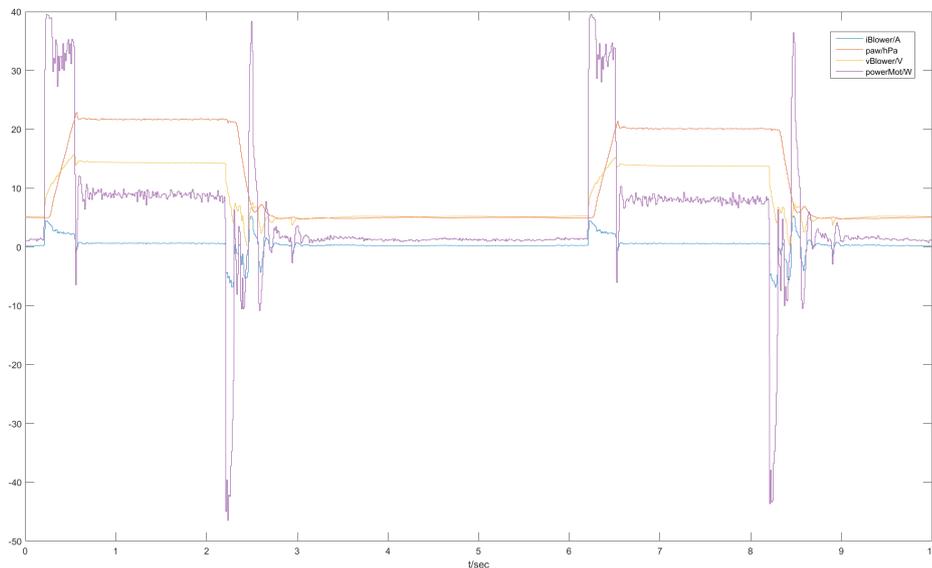


Abbildung 2.1: Lastsprung der Gebläseelektronik

¹ Nach [2] beträgt die aufgenommene Energie eines Bremsvorganges 10 Joule

2.2 Ideale Dioden

Ideale Dioden bestehen aus einem MOSFET (metal-oxid-semiconductor field-effect transistor) und einem IC (integrated circuit), welche das Verhalten von Dioden nachbilden. Bei Dioden fällt im leitenden Zustand zwischen Anode und Kathode die Flussspannung $U_F = 0,5V \dots 0,7V$ ab. Dies hat den Nachteil, dass bei hoher Strombelastung die Diode mit einer Verlustleistung von $P_{Diode} = U_F \cdot I$ den Wirkungsgrad des Systems entscheidend verschlechtert.

Setzt man eine ideale Diode ein, fließt der Strom der Leistungselektronik über den Widerstand eines MOSFETs R_{DSon} , welcher sich meist im Milliohm-Bereich bewegt. Das hat zur Folge, dass die Verlustleistung $P_{Diode} = I \cdot R_{DSon}$ erheblich verringert werden kann. Die Schaltverluste des MOSFETs und des ICs sind minimal und wurden vernachlässigt.

Der IC bildet das Verhalten einer Diode nach, indem dieses die Drain- und Sourcespannung eines MOSFETs misst und das Gate dementsprechend ansteuert.

2.3 Schaltregler

Synchroner Buck-Boost-Wandler

Der synchrone Buck-Boost-Wandler kann im Prinzip jedes beliebige Spannungsniveau auf ein höheres als auch niedrigeres Potenzial transformieren. Dieser ist aus zwei kaskadierten Buck- und Boost-Wandlern aufgebaut, benötigt allerdings nur eine Speicherdrossel. Ein Stromfluss kann in beliebiger Richtung fließen.

Synchroner Boost-Wandler

Der synchrone Boost-Wandler, oder auch Aufwärtswandler genannt, ist ein Schaltwandler. Er kann seine Ausgangsspannung auf ein höheres Potenzial als seine Eingangsspannung transformieren.

Aus schaltungstechnischer Sicht unterscheiden sich Buck- und Boost-Wandler nicht. Lediglich die Stromflussrichtung und das eingestellte Tastverhältnis des Ansteuersignals bestimmen das Übertragungsverhalten.

Abbildung 2.2 zeigt den grundsätzlichen Aufbau des Aufwärtswandlers. Die Schalter S1 und S2 dienen der Veranschaulichung und werden in der Praxis durch geeignete Transistoren ersetzt und mittels PWM (Pulse-Width-Modulation) angesteuert.

Die Schalter S1 und S2 werden zeitversetzt angesteuert. Wenn der eine leitet, sperrt der andere.

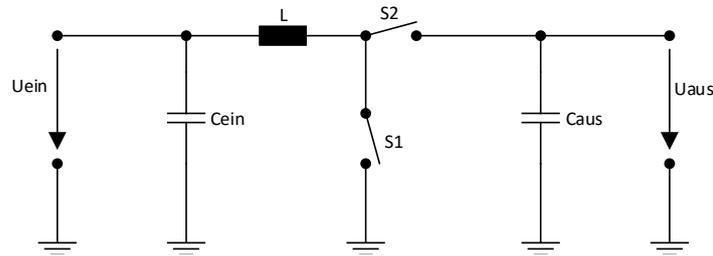


Abbildung 2.2: Grundsaltung des synchronen Boost-Wandlers

Die Einschaltdauer von $S1$ im Verhältnis zu $S2$ bestimmt hierbei das Übertragungsverhältnis.

Die Funktionalität des Boost-Wandlers kann in zwei Phasen aufgeteilt werden.

Phase 1 - $S1$ leitend, $S2$ geöffnet:

Der Strom in die Drosselspule L steigt an und lädt diese auf.

Phase 2 - $S1$ geöffnet, $S2$ leitend:

Die in der Drosselspule L gespeicherte Energie wird an den Ausgang über $S2$ weiter geschaltet.

Das Verhältnis von Ausgangsspannung und Eingangsspannung ergibt sich laut [3] wie folgt:

$$\frac{U_{aus}}{U_{ein}} = \frac{1}{1 - \frac{t_{ein}}{T}} \quad (2.1)$$

Dabei entspricht t_{ein} der Zeit, in der $S1$ leitend ist und T die Zeit, in der $S1$ einmal leitend und sperrend war.

Geht man von idealen Bauteilen aus und damit von einem Wirkungsgrad von 100%, folgt für das Verhältnis der Ströme:

$$\frac{I_{aus}}{I_{ein}} = 1 - \frac{t_{ein}}{T} \quad (2.2)$$

Speicherdrossel

Nach Quelle [3][2,S.72] wird die Drosselspule L ausgelegt nach

$$L = \frac{U_{ein}(U_{aus} - U_{ein})}{\Delta I \cdot f \cdot U_{aus}} \quad (2.3)$$

f beschreibt hierbei die Schaltfrequenz der Schalter $S1$ und $S2$. ΔI ist der Stromripple durch die Spule. Der Stromripple ist der zu- und abnehmende Strom während der Ladevorgänge und pendelt um den Eingangsstrom des Boost-Wandlers.

Die Induktivität ist umgekehrt proportional zum Ripplestrom und der Schaltfrequenz f . Um diese möglichst kompakt auszulegen, wird die Frequenz f möglichst hoch gewählt. Umgekehrt muss die Drosselspule so groß gewählt werden, dass der Ripplestrom möglichst klein gehalten wird, denn Spitzenströme durch die Spule sind maßgeblich an den Verlusten beteiligt.

Ausgangskapazität

Die Ausgangskapazität C_{aus} speichert die Energie der Drosselspule, während $S1$ geschlossen ist. Wird die Induktivität geladen, kann die gespeicherte Energie an den Ausgang weitergegeben werden. Damit wird die Ausgangsspannung geglättet.

Die Potenzialunterschiede ΔU am Kondensator C_{aus} können bedingt durch die Lade- und Entladevorgänge je nach Anforderung nach [3] durch einen geeigneten Kondensator ausgelegt werden:

$$C_{aus} = I_{aus} \cdot \frac{t_{ein}}{\Delta U} \quad (2.4)$$

I_{aus} beschreibt hierbei den maximalen Laststrom.

2.4 Verlustleistung - Wirkungsgrad

Um den Wirkungsgrad eines Schaltwandlers zu ermitteln, müssen die auftretenden Verluste berechnet werden. Die größten Verluste treten in den Umladeverlusten der MOSFETs und der Spule auf. Für den Wirkungsgrad gilt:

$$\eta = \frac{P_{aus}}{P_{ein}}. \quad (2.5)$$

Kernverluste

Kernverluste beschreiben Verlustleistungen, die im Kernmaterial der Speicherdrossel auftreten. Diese treten nur auf bei Spulen, die einen Kern besitzen. Es wird magnetisch, aber nicht elektrisch leitfähiges Material eingesetzt (Ferrit), um eine höhere Induktivität bei kleinerer Bauform zu erzielen.

Die Verluste des Kerns setzen sich nach [4] aus Wirbelstrom- und Ummagnetisierungsverlusten zusammen. Wirbelstromverluste treten durch die Leitfähigkeit des Kerns auf und können

durch geeignete Verfahren gering gehalten werden. Die Ummagnetisierungsverluste treten aufgrund der Hysterese von Induktivitäten in den Lade- und Entladevorgängen auf.

Die zu erwartenden Verluste sind kernmaterial- und frequenzabhängig. Sie können meist durch herstellerepezifische Angaben berechnet werden.

Die Sättigung der Speicherdrossel sollte möglichst vermieden werden, da die Induktivität in diesem Fall sinkt und die Verluste zunehmen. Beispielsweise gibt die Firma Würth Elektronik einen Sättigungsstrom an, bei dem die Spule eine Induktivität von 10% einbüßt.

Verluste im MOSFET

Ein MOSFET ist ein spannungsgesteuerter Leistungsschalter. Im sperrenden Zustand wird bis auf einen Leckstrom keine Leistung umgesetzt und kann somit vernachlässigt werden. Ist dieser voll durchgesteuert, trägt nur der Kanalwiderstand $R_{DS(on)}$ zur Verlustleistung bei, welcher sich im $m\Omega$ -Bereich bewegt. Wird der MOSFET mit einer hohen Frequenz umgeschaltet, tragen die Umschaltverluste den größten Teil zu Verlustleistung bei.

Beim Übergang vom sperrenden in leitenden Zustand muss die Gate-Source-Spannung, bei einem n-Kanal-MOSFET auf eine Spannung oberhalb der Threshold-Spannung $U_{GS(th)}$ geladen werden. In dieser Zeit durchläuft der Kanalwiderstand den linearen Bereich, wodurch Verlustleistung in Form von Wärme über dem Kanalwiderstand abgegeben wird, bevor dieser niederohmig wird. Um von dem leitenden in den sperrenden Zustand zu gelangen, muss die Spannung $U_{GS} < U_{GS(th)}$ betragen. Die Zeit, in der der lineare Bereich durchschritten wird, ist durch die Gatekapazität des MOSFETs bestimmt. Je schneller umgeladen werden kann, desto schneller kann der lineare Bereich verlassen werden.

Die Umschaltverluste können klein gehalten werden, indem Gatetreiber mit einem hohen Ausgangsstrom eingesetzt werden.

Bei dem Einschaltvorgang wird die Gatekapazität aufgeladen, zum Ausschalten muss diese wieder entladen werden. Die gespeicherte Energie in dem Gate des MOSFETs wandelt der Gatetreiber in Wärme um und trägt zur Verlustleistung bei.

Üblicherweise liegt die Ladung der Gatekapazität im nC-Bereich. Diese lässt sich folgendermaßen berechnen:

$$Q_g = \frac{C_g}{U_g} \quad (2.6)$$

Über die im Gate gespeicherte Energie

$$E_g = \frac{1}{2} \cdot U_g^2 \cdot C_g = \frac{1}{2} \cdot U_g \cdot Q_g \quad (2.7)$$

lässt sich mithilfe der Frequenz f auf die Verlustleistung schließen:

$$P_{Gate} = E_g \cdot f \quad (2.8)$$

Ohmsche Verluste treten überall dort auf, wo aufgrund von Ohmschen Widerständen, Wärme erzeugt wird.

$$P_{\Omega} = I^2 \cdot R \quad (2.9)$$

Ohmsche Widerstände treten in MOSFETs als Kanalwiderstand, in der Induktivität als Kupferwiderstand sowie in Kondensatoren auf und werden dort als ESR (Ersatzreihenwiderstand) zusammengefasst. Da der ESR frequenzabhängig ist, ist meist ein $\tan\delta$ in den Datenblättern angegeben und der ESR berechnet sich zu

$$ESR = \frac{\tan\delta}{\omega \cdot C}. \quad (2.10)$$

Berechnung des Wirkungsgrad

Die insgesamt auftretenden Verluste eines Boost-Wandlers lassen sich zusammenfassen als:

$$P_{ab} = P_{Gate} + P_{\Omega} + P_{Kern}, \quad (2.11)$$

wobei die Ausgangsleistung mit

$$P_{aus} = P_{ein} - P_{ab}, \quad (2.12)$$

ermittelt werden kann. Daraus ergibt sich für den Wirkungsgrad:

$$\eta = \frac{P_{aus}}{P_{ein}} = \frac{P_{aus}}{P_{aus} + P_{ab}} \quad (2.13)$$

2.5 PI - Regler

Damit die Störgrößen, welche auf das Energiemanagementsystem einwirken, ausgeregelt werden können, werden in dieser Arbeit PI-Regler verwendet. Abbildung 2.3 zeigt das Regelungssystem eines PI-Reglers.

Das Regelungssystem bildet die Differenz aus einem Sollwert und dem Istwert der Regelstrecke. Die sich daraus ergebende Regelabweichung wird dem PI-Regler überführt, welcher gemäß den Regelparametern eine Stellgröße auf die Regelstrecke gibt. Dieses ist nötig, damit der Störgröße, welche auf die Regelstrecke wirkt, entgegengewirkt werden kann.

Der P-Anteil des Reglers wirkt mit dem proportionalen Anteil der Verstärkung K_p . Mit seiner Stellgröße ist er proportional zu der Regelabweichung. Dieser ermöglicht ein schnelles Ausregeln der Stellgröße, hat allerdings den Nachteil einer bleibenden Regelabweichung. Deshalb wird zusätzlich ein I-Anteil verwendet.

Der I-Anteil (integrierender Anteil) des Reglers wirkt durch zeitliche Integration der Regelabweichung auf die Stellgröße mit der Gewichtung durch die Nachstellzeit T_n . Der I-Anteil wirkt langsamer als ein P-Anteil auf eine Regelabweichung, eliminiert allerdings eine Regelabweichung.

Für eine genauere Betrachtung von Reglersystemen sei auf [5] verwiesen.

Kann der Regler die Störgröße nicht ausregeln, da beispielsweise zu wenig Energie zur Verfügung steht, integriert der I-Anteil den Fehler durch die bleibende Regelabweichung weiter auf. Das hat zur Folge, dass möglicherweise eine fehlerhaft überhöhte Stellgröße auf die Regelstrecke wirkt, wenn wieder genügend Energie zur Verfügung steht. Dadurch kann es zu einem starken Überschwingen kommen oder zu einer Instabilität des Reglers, da der hohe I-Anteil zunächst wieder abgebaut werden muss. Aus diesem Grund wird zusätzlich ein "Anti-Wind-Up" System verwendet, welches dem überhöhten Integrieren des Reglers entgegenwirkt, wenn der Istwert sich nicht mehr ändert.

Das Reglersystem wird auf einem Mikrocontroller ausgeführt. Dieses kann nicht stetig auf die Stellgröße einwirken, sondern wird als Programm in einer gewissen Frequenz f ausgeführt. Die Frequenz f muss dementsprechend so groß gewählt werden, dass auf eine Änderung der Störgröße entsprechend schnell reagiert werden kann. Zudem muss der Istwert mittels ADC ausreichend schnell zur Verfügung stehen.

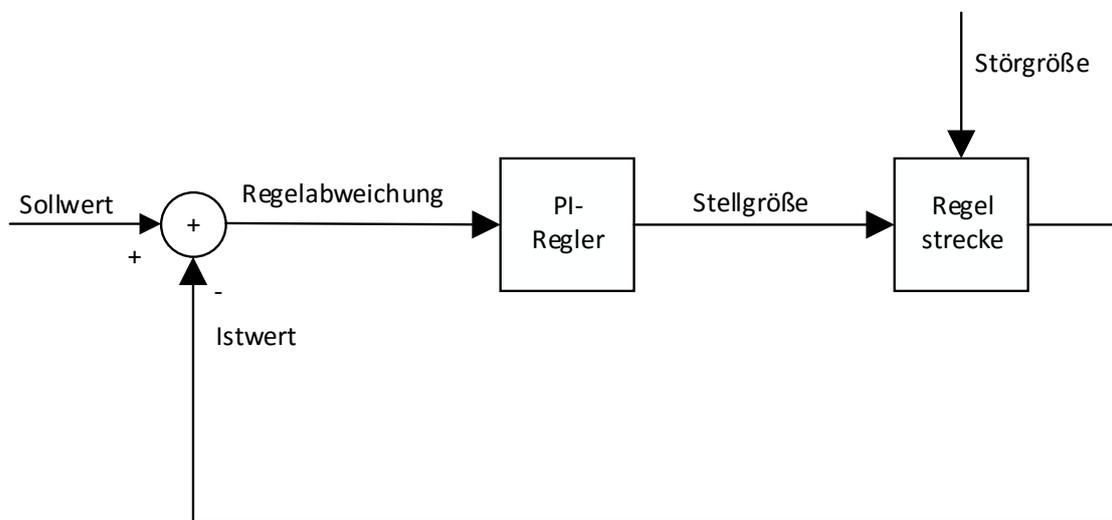


Abbildung 2.3: Geschlossenes Regelungssystem eines PI-Reglers

3 Anforderungen

In diesem Kapitel wird ein detailliertes Anforderungsprofil für das Energiemanagementsystem erstellt. Das System soll am Beispiel eines elektrischen Beatmungsgerätes entwickelt werden, welches in Kapitel 4 näher beleuchtet wird. Daraus ergeben sich Anforderungen an die Funktionalität, elektrische Parameter, an die Software und die Energieeffizienz. Aus diesen Rahmenbedingungen lässt sich ein Konzept entwickeln, welches anhand einer Versuchshardware auf Realisierbarkeit überprüft werden kann.

Eine mögliche Versorgungsquelle der Leistungselektronik ist ein 12-Volt-Bordnetz eines Rettungswagen oder eines Rettungshubschraubers. Das 12V-Bordnetz hat einen Toleranzbereich von -15% bis $+25\%$. Für diesen Spannungsbereich kann das Gerät die Versorgung des Bordnetzes nutzen. Von der Firma Weinmann wurde festgelegt, dass bei einem unzulässigen Spannungsbereich außerhalb von 10V bis 16V die Spannungsversorgung über das Bordnetzes nicht mehr verwendet wird. Die zweite Versorgungsquelle steht als 3-Zellen-Lithium-Ionen-Akku zur Verfügung. Deren Spannungsbereich reicht von der Entladespannung 9 Volt bis zur Ladeschlussspannung von 12,6 Volt. Dieser Spannungsbereich muss exakt eingehalten werden, da der Akku sonst zu Schaden kommen kann.

Für die Strombelastbarkeit der Quellen wurde festgelegt, dass diese nicht mehr als 4 Ampere betragen darf.

Für das Energiemanagementsystem soll gelten, dass das Bordnetz als Versorgungsquelle verwendet wird, wenn dieses zur Verfügung steht. Der Akku hingegen soll nur verwendet werden, wenn das Bordnetz nicht zur Verfügung steht. Damit unterliegt die Quellenauswahl einer Priorisierung mit:

1. Bordnetz
2. Akku

Zusätzlich kann es nützlich sein einen Mischbetrieb einzuführen. Kann dem Bordnetz nicht die benötigte Leistung entnommen werden, könnte der Akku unterstützen, um den Betrieb des Gerätes aufrecht zu erhalten. Soll das Energiemanagementsystem auf andere Geräte übertragen werden, kann ein Mischbetrieb zudem einen entscheidenden Vorteil bringen.

Für ein Akku-betriebenes Gerät muss eine Ladeelektronik zur Verfügung stehen. Es soll gezeigt werden, dass ein entwickeltes Konzept in der Lage ist, den Strom in den Akku zu

regeln. Es wird davon abgesehen, einen vollständigen Ladealgorithmus zu implementieren, da nur die Möglichkeit aufgezeigt werden soll. Zudem muss gewährleistet sein, dass das Gerät sich nicht entlädt, wenn dieses nicht verwendet wird und damit ausgeschaltet ist.

Die Hauptaufgabe eines Energiemanagementsystems ist es, die ihm zugeführte Energie sinnvoll auf die Verbraucher aufzuteilen. Die Verbraucher stellen sich aus Kleinspannungsversorgung, Versorgung des Aktors (Beatmungselektronik) sowie dem Akkulader zusammen. Der Aktor stellt im Vergleich zur Kleinspannungsversorgung die größte Herausforderung an die Leistungselektronik. Dieser soll der Beatmungselektronik eine mittlere Leistungsaufnahme von 33 Watt zur Verfügung stellen können. Damit die Regelung für das Energiemanagement betrieben werden kann, muss die Kleinspannung versorgt sein. Der Akku kann geladen werden, wenn genügend Energie zur Verfügung steht, darf allerdings nicht die Beatmung beeinträchtigen. Daraus ergibt sich eine Priorisierung für die Verbraucher mit:

1. Kleinspannungsversorgung
2. Aktor
3. Akkulader

Das Energiemanagementsystem soll mikrocontrollergesteuert realisiert werden. Davon ausgehend muss eine geeignete Software entwickelt werden, welche die geforderten Funktionen umsetzt. Für ein Reglersystem, welches auf Änderungen der Ströme und Spannungen reagieren muss, soll ein Multi- oder Einratensystem entworfen werden, welches mit einer Reglerfrequenz von mindestens 4kHz arbeitet. Die Reglerfrequenz begründet sich mit der Abtastfrequenz des Beatmungsalgorithmus.

Abgesehen von der Funktionalität muss ein Energiemanagementsystem mit den ihm zur Verfügung stehenden Ressourcen entsprechend haushalten. Damit ein Vorteil gegenüber dem vorhandenen System mit dezentralen Funktionsgruppen entsteht, sollen die Verluste in dem Leistungspfad verringert werden. Es wird ein Wirkungsgrad $> 90\%$ angestrebt.

Der Firma Weinmann ist es aus wirtschaftlichen Gründen zudem wichtig, die Anzahl an Bauteilen für eine verringerte Komplexität zu reduzieren.

In Tabelle 3.1 sind die erarbeiteten Anforderungen, welche das Konzept erfüllen muss, übersichtlich zusammengefasst.

Bereich:	Beschreibung	Wert	Typ
Funktion	Primäre Versorgungsquelle: Bordnetz	10V-16V / max. 4A	Anforderungen
	Sekundäre Versorgungsquelle: Akku	9V-12,6V / max. 4A	
	Sekundäre Last: Konstante Energieversorgung der Leistungselektronik (Motorgebläse)	min. 25V / 33W	
	Steuerung/Regelung der Leistungselektronik durch Mikrocontroller		
	Tertiäre Last: Ein Laden des Akkus durch das Bordnetz ermöglichen	max. 2A	
	Speichern von zurückgewonnener Bremsenergie des Gebläses	10 Joule	
	Unterstützen durch Akku bei schwachem Bordnetz (Mischen der Quellen)		Wunsch
	Primäre Last: Energieversorgung der Kleinspannungselektronik		Anforderungen
Kein Entladen des Akkus bei System-Aus (kein Nennenswerter Leckstrom)			
Software	Multi- oder Einratensystem	min. 4kHz Abtastrate	
Energieeffizienz	Verluste im Leistungspfad minimieren gegenüber dem vorhandenen System		
	Wirkungsgrad	>90%	
Wirtschaftlichkeit	Möglichst wenig Bauteile, Komplexität		Wunsch

Tabelle 3.1: Anforderungen

4 Analyse

In diesem Kapitel wird das vorhandene Energiemanagementsystem für ein Beatmungsgerät der Firma Weinmann vorgestellt und analysiert. Es wird auf die einzelnen Funktionsgruppen der Leistungselektronik und deren Aufgaben eingegangen. Am Ende wird das System als Gesamtheit betrachtet.

4.1 Stand der Technik

Wie in Abbildung 4.1 zu sehen, stellt das System aus einem Bordnetz und einem Akku die Spannungen für die Kleinspannungsversorgung und die Beatmungselektronik zur Verfügung. Sowie die Möglichkeit, den Akku aus dem Bordnetz zu laden.

Grundsätzlich ist das Energiemanagementsystem dezentral aufgebaut, welche in Funktionsblöcken beschrieben werden. Das Kleinspannungsnetzteil liefert die Spannung für die Peripherie. Diese setzt sich im Allgemeinen aus Beatmungscontroller, Embedded PC, Display und Sensoren zusammen. Im Vergleich zur Leistungselektronik stellt die Peripherie mit 3 Watt allerdings nur einen geringen Anteil der Leistungsaufnahme dar. Dagegen stellt die Beatmungselektronik mit einer Leistungsaufnahme von maximal 33 Watt die eigentliche Herausforderung an die Leistungselektronik.

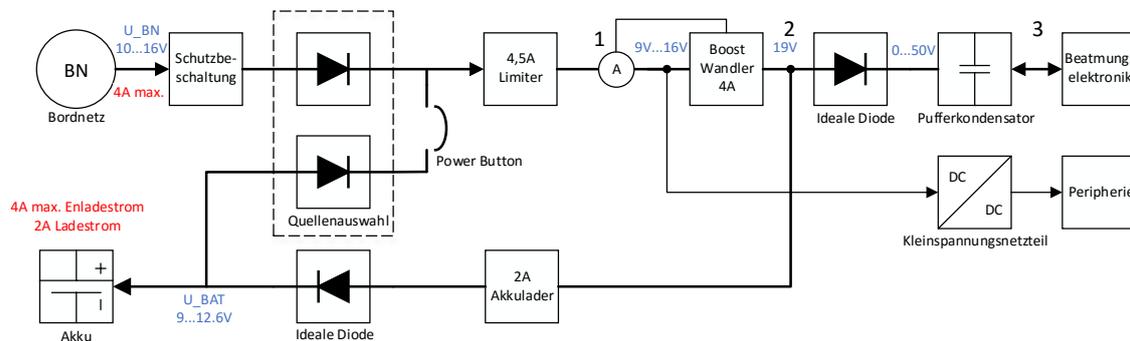


Abbildung 4.1: Vorhandenes Energiemanagementsystem

Bordnetz:

Die Zentralversorgung wird in aller Regel durch ein Kfz-Bordnetz zur Verfügung gestellt. Ist dieses angeschlossen, ist seine Aufgabe, die komplette Energieversorgung des Gerätes sicherzustellen und wenn möglich den Akku nachzuladen. Da die Firma Weinmann in Notfallfahrzeugen meist 2 Geräte verbaut, wurde die maximale Stromaufnahme auf 4 Ampere in einem Arbeitsbereich von 10-16V ($12V - 15\%$) bis ($12V + 25\%$) festgelegt. Für EMV- (Elektromagnetische Verträglichkeit) und Überspannungsmaßnahmen sind Schutzbeschaltungen vorgesehen.

Akku:

Als alternative Spannungsquelle ist ein 3 Zellen Lithium-Ionen Akku mit 10,9V; 4,5Ah; 46,8Wh mit einem eigenen Controller, welcher über eine SMBus Schnittstelle (System-Management-Bus) ausgelesen wird. Seine Entladespannung beträgt 9V und seine Ladeschlussspannung 12,6V.

Da Weinmann-Geräte für einen Temperaturbereich zwischen -20° bis 50° vorgesehen sind, wurde die maximale Strombelastung aufgrund von internen Tests der Temperaturbelastbarkeit auf 4 Ampere festgelegt. Das Laden des Akkus übernimmt ein 2A-Buck-Laderegler. Wird zudem der Akku mit über 2 Ampere geladen, schaltet der interne Akkucontroller die Batterie von der Versorgung ab.

Der Akkulader lädt den Lithium-Ionen Akku mit konstantem Ladestrom bis zu seiner Ladeschlussspannung. Bei Erreichen der Ladeschlussspannung schaltet dieser auf Konstantspannungsregelung um. Regelt man auf die Ladeschlussspannung, nimmt der Strom exponentiell ab und erreicht seinen maximalen Ladezustand, wenn der Strom in den Akku gegen ein Minimum geht. Das Umschalten bei Ladeschlussspannung ist notwendig, damit keine Überspannung an dem Akku anliegt.

Priorisierte Verknüpfung der Quellen:

Eine Anforderung an das Energiemanagement ist die Energiequellen zu priorisieren. Wenn ein Bordnetz angeschlossen ist und eine stabile Spannung anliegt (10-16V), soll das Bordnetz als Quelle verwendet werden. Die **Quellenauswahl** übernimmt ein IC, welches sich wie zwei Ideale-Dioden, mit Priorisierung der Quelle verhält.

In dem Fall, dass die Bordnetzspannung geringer als die Akkuspannung ist, kann keine zuverlässige Priorisierung mittels idealer-Dioden erfolgen, da ein MOSFET nur in eine Richtung sperren kann. Liegt eine Drain-Source-Spannung < 0 an, fließt ein Strom über die interne Body-Diode des MOSFETs. Damit die Energiequellen priorisiert werden können, steuert die Quellenauswahl einen zusätzlichen MOSFET an. Dieser kann die Akkuspannung von dem Leistungspfad trennen. Zusätzlich verhindert eine analoge Schaltung, das Aufschwingen der

Quellen, wenn deren Potenzial nahe einer Ein- bzw. Abschaltgrenze liegt. Um dem entgegenzuwirken, verhindert eine Hysterese und eine zeitliche Begrenzung das Umschalten der Spannungsquellen.

Da die Spannungsversorgung in einer geschalteten Verknüpfung realisiert wurde, lässt dieses keinen Mischbetrieb zu.

Spannungsebenen

Die Spannungsebenen der Eingangsspannung reichen von 9V-Entladespannung des Akkus bis zu 16V maximale Eingangsspannung des Bordnetzes. Diese speist die 5V-Kleinspannungsversorgung, welche ein Buck-Wandler übernimmt. Aus diesen 5V werden alle Spannungsebenen auf Kleinspannungsebene erzeugt, welche für aktive Bauelemente, Controller, Sensoren, Display und Embedded-PC benötigt werden.

Auf Leistungsspannungsebene erzeugt ein Boost-Wandler eine 19V Zwischenkreisspannung. Diese wird einem Akkulader und dem Beatmungsgebläsesystem zur Verfügung gestellt. An Punkt 1 in Abbildung 4.1 befindet sich die Strommessung für den 4A-Boost-Wandler. Durch die Strommessung erfolgt die Priorisierung der Lasten.

Die Spannung an Punkt 2 kann nur dann ausgeregelt werden, wenn aus den Versorgungsquellen ein Strom $< 4A$ fließt. Somit steht der Kleinspannungsversorgung immer genügend Energie zur Verfügung, dem Akkulader und der Beatmungselektronik jedoch nur bei Einhaltung der maximalen Leistungsaufnahme des Gerätes.

Der Akkulader kann von einem Mikrocontroller aktiviert werden, wenn genügend Energie zur Verfügung steht, bzw. deaktiviert werden, wenn das Beatmungssystem diese benötigt.

Daraus lässt sich die Priorisierung der Verbraucher ableiten:

1. Kleinspannungsversorgung
2. Beatmungselektronik
3. Akkulader

Punkt 3 stellt den Ausgang des Energiemanagementsystems dar. Die Beatmungselektronik ist der Aktor, welcher mit konstanter Energie versorgt werden muss. Hier stützt der Pufferkondensator bei Lastsprüngen des Aktors. Die Größe des Kondensators ist auf die Beatmungselektronik abgestimmt. Diese liefert während des Bremsvorganges eine Energie von 10 Joule, welche der Kondensator aufnimmt. Die Spannung an dem Kondensator steigt daraufhin auf bis zu 50V. Damit der Rest des Systems vor Überspannung geschützt ist, sperrt eine Ideale Diode den Energiefluss von der Beatmungselektronik in die Leistungselektronik.

Beatmungselektronik

Die Beatmungselektronik wird von einem ein Buck-Boost-Wandler und einem BLDC-Motor (Brushless-Direct-Current-Motor) mit angeflanschem Radialgebläse gebildet (Abbildung: 4.2). Dieser wird durch eine Vollbrückenschaltung (Motor-Brücke) kommutiert. Die Drehzahl des Motors wird von einem Beatmungscontroller über einen synchronen Buck-Boost-Wandler eingeregelt. Je nachdem, welche Spannung am Ausgang des DC-DC-Wandlers eingestellt wird, stellt sich in dem Gebläse eine Drehzahl und somit ein Beatmungsdruck ein.

Die Energie liefert das Energiemanagementsystem (hier EMS abgekürzt).

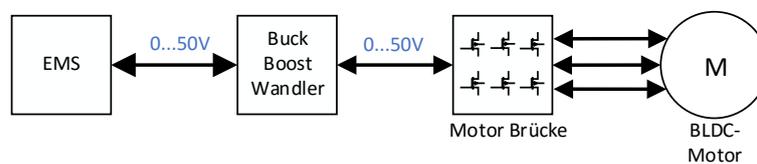


Abbildung 4.2: Energieflussdiagramm der Beatmungselektronik

Gesamtsystem

Das Energiemanagementsystem wurde speziell für das Gerät entwickelt. Es besteht aus mehreren dezentralen Funktionsgruppen. Hinzu kommen noch Sicherheitsschaltungen, wie ein 4,5A-Limiter-IC (integrated Circuit) um den hohen Anforderungen der Medizintechnik gerecht zu werden. Der 4,5A-Limiter begrenzt Stromspitzen in das Bordnetz seitens des 4A-Buck-Boost-Wandlers, welcher die Spannung für die Leistungselektronik bereitstellt. Dieser ist mit seiner 4A Begrenzung für die maximale Leistungsaufnahme des Gesamtsystems zuständig.

In dem Buck-Boost Wandler sind weitere Sicherheitsschaltungen für die Motoransteuerung integriert, um bei Teilsystemausfall den BLCD-Motor abzuschalten. Oberste Priorität hat die Verhinderung von Patientengefährdungen, denn im Fehlerfall kann die Energie ungesteuert in die Gebläseelektronik fließen und es kann zu einer akuten Lebensbedrohung kommen.

Bis auf das Gebläsesystem ist das Energiemanagementkonzept aus einzelnen ICs aufgebaut. Diese ICs können von einem Mikrocontroller nur ein- bzw. ausgeschaltet werden. Das Energiemanagement ist somit sehr statisch und unflexibel. Eine Änderung einer Anforderung an die Energieversorgung geht damit immer mit einer Hardwareanpassung einher.

Ein weiteres Problem stellen die Verluste der vielen Bauteilgruppen dar. Als Beispiel sei hier die Akkuladung angeführt. Um den Akku zu laden, muss die Energie durch mehrere verlustbehaftete Bauteilgruppen geführt werden, wobei der Buck-Boost-Wandler sowie der Akkulader die größten Verluste verursachen. Da die Energie aus dem Bordnetz begrenzt ist, muss diese Effizient verarbeitet werden. Auch der Leistungspfad zum Gebläsemotor trägt mit zwei Buck-Boost-Wandlern einen großen Teil zur Geräteeffizienz bei. Um den Akku im Betrieb effizient zu laden und einen ausreichend langen Akkubetrieb zu ermöglichen, ist das Energiemanagement ein entscheidender Faktor.

Ein für jedes Gerät spezielles Energiemanagementsystem erhöht die Entwicklungskosten. Hinzu kommen durch die dezentralen Funktionsgruppen geschuldete Bauteilvielfalt erhöhte Produktionskosten.

Das System funktioniert sicher, bietet allerdings keinen Mischbetrieb und muss daher in jeder Situation mit einer Versorgungsquelle auskommen.

4.2 Herangehensweise

Als Herangehensweise für die Entwicklung dient das V-Modell (Vorgehensmodell). Es kann sowohl für die Softwareentwicklung auf Mikrocontrollerebene als auch für die Hardwareentwicklung verwendet werden.

Nach dem V-Modell für die Hardware in Abbildung 4.3 werden aus den Anforderungen für das System mehrere Konzepte entwickelt und verglichen und das geeignetste ausgewählt. Das Konzept auf Blockschaltbild-Ebene wird im folgenden Schritt feinspezifiziert und dimensioniert. Daraus folgt die Umsetzung mit dem Erstellen eines Schaltplans, eines Platinen-Layouts und die Bestückung. Nach dem V-Modell werden die einzelnen Entwicklungsschritte getestet und können gegebenenfalls angepasst werden.

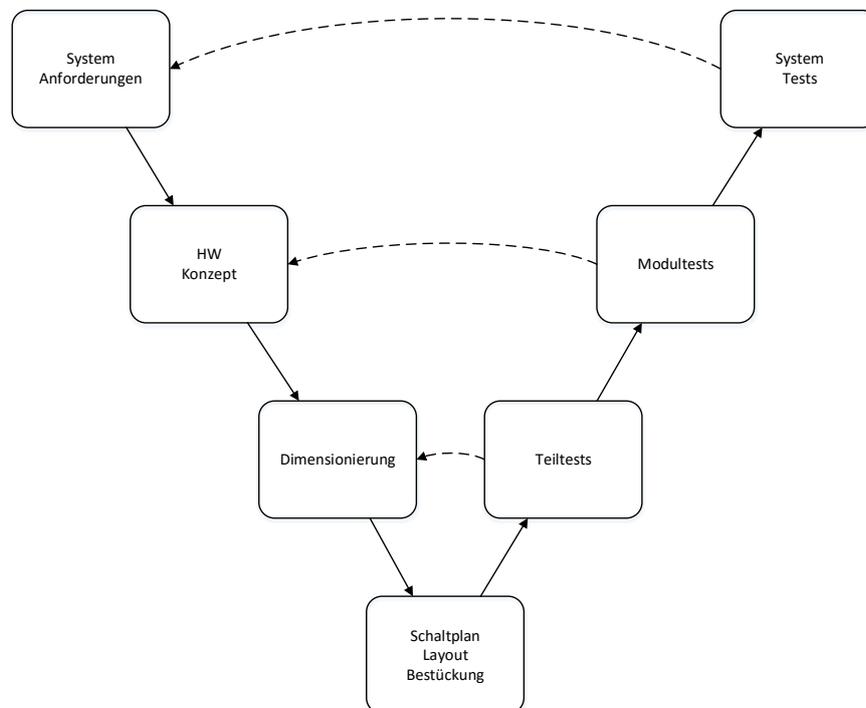


Abbildung 4.3: V-Modell für die Hardwareentwicklung des Energiemanagementsystems

Die Controller Software wird nach dem V-Modell in Abbildung 4.4 entwickelt, wobei das V-Modell die Verbundenheit zu dem V-Modell der Hardware nicht abbildet. Aus den Anforderungen für die Hardware wird ein Software-Konzept entwickelt. Als nächster Schritt wird ein Modell der Regelstrecke und des Reglers aufgebaut, wobei dies auch schon Teil einer Implementierung ist, die im V-Modell als letzter Entwicklungsschritt vorgesehen ist. Die Implementierung beinhaltet die Code-Generierung aus dem Matlab-Simulink-Modell und dem Programmieren der benötigten C-Treiber. Die Validierung beinhaltet das Sicherstellen der entwickelten Funktionalitäten nach dem V-Modell, wobei der System-Test sich mit dem des Hardware V-Modell überschneidet.

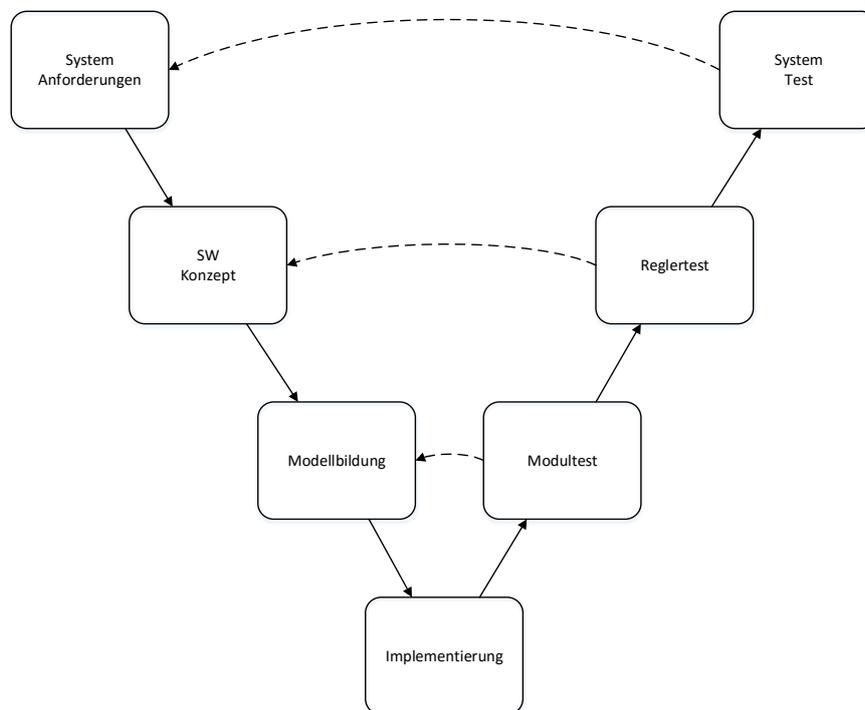


Abbildung 4.4: V-Modell für die Softwareentwicklung

4.3 Ansätze

In diesem Kapitel werden die entwickelten Konzepte vorgestellt. Grundlage bietet das bei Weinmann vorhandene Energiemanagementsystem. Es werden Möglichkeiten erörtert, um den gestellten Anforderungen aus Tabelle 3.1 gerecht zu werden. Für die Ansätze als Hilfreich erwies sich [6]

Als erster Ansatz wurde versucht, die geschaltete Quellenauswahl zu eliminieren und durch softwaregesteuerte Elemente zu ersetzen. Hier fiel die Wahl schnell auf DC/DC Wandler die es erlauben, benötigte Spannungsniveaus mit Strombegrenzungen einzustellen.

Im Folgenden werden die Konzepte, vorgestellt bewertet und verglichen.

Konzept 1

Wie in Abbildung 4.5 dargestellt, eliminiert dieses Konzept die geschaltete Verknüpfung der Quellen und soll einen gesteuerten Mischbetrieb ermöglichen. Dieses ist bei mobilen Geräten wie Laptops eine gängige Methode, sein Netzteil bei Spitzenlast zu entlasten.

Die Bezeichner (1),(2),(3),(4) beziehen sich im Folgenden, auf die Bezeichner in Abbildung 4.5.

Ist kein Bordnetz angeschlossen, wird das Gerät über den Power-Button gestartet. Der Akku versorgt die Kleinspannungselektronik über die Oring-Diode mit Energie. Die Beatmungselektronik ist über die ideale Diode (3) mit dem Akku verbunden.

Das Bordnetz ist über einen Buck-Boost-Wandler (1) mit der Ausgangsspannung verbunden. Die ideale Diode im Bordnetzeingang soll verhindern, dass Energie in das Bordnetz fließt. Die ideale Diode (4) sperrt den Energiefluss in das System während eines Bremsvorganges. Damit der Akku zuverlässig geladen werden kann, hebt der Buck-Boost-Wandler (1) die Zwischenkreisspannung U_{ZK} auf 15V. Das ermöglicht dem 2A-Akkulader (2) (Buck-Wandler), die Ladespannungen für den Akku zu erzeugen.

Während des Mischbetriebes ist das Bordnetz wie gefordert priorisiert. Die Zwischenkreisspannung liegt im Regelfall auf einem höheren Spannungsniveau als der Akku. Somit sperrt die Ideale Diode (3) ungewollten Stromfluss in den Akku.

Steht dem Akkulader (2) genügend Energie zur Verfügung, kann dieser via Software eingeschaltet werden. Steht im umgekehrten Fall der Beatmungselektronik zu wenig Energie zur Verfügung, wird der Akkulader wieder abgeschaltet.

Für den Fall, dass der Eingangsstrom I_{BN} durch den Buck-Boost-Wandler (1) begrenzt wird, sinkt die Zwischenkreisspannung. Unterhalb der Akkuspannung, kann der Akku über die Ideale-Diode (3) das Bordnetz unterstützen, um das weitere Einbrechen der Zwischenkreisspannung zu verhindern.

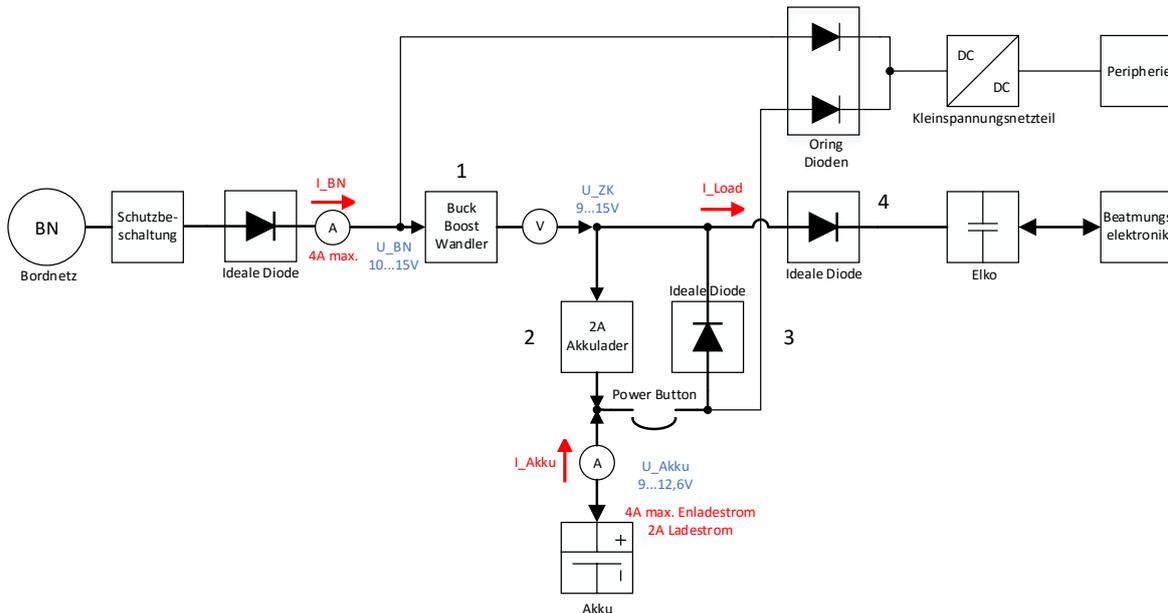


Abbildung 4.5: Konzept 1 - Mischbetrieb mit Akkulader ohne geschaltete Quellenversorgung

Das Konzept 1 hat den Vorteil, mit geringer Änderung einen Mischbetrieb einzuführen und die geschaltete Quellenauswahl abzulösen. Zudem kann es die geforderten elektrischen Parameter einhalten.

Nachteilig ist, dass keine Reduktion der Bauteilvielfalt erzielt wurde. Die Funktionsgruppen sind weiterhin dezentral ausgeführt.

Bedenklich ist zudem, dass der Strom I_{Akku} in den Elektrolytkondensator und in die Beatmungselektronik nicht gesteuert werden kann.

Konzept 2

Abbildung 4.6 zeigt den Aufbau von Konzept 2. Das Konzept reduziert die Bauteilelemente und verzichtet auf einen dezentralen Akkulader.

Der Bezeichner (1) bezieht sich im Folgenden, auf die Bezeichner in Abbildung 4.6.

Wie in Konzept 1 bleibt der Buck-Boost-Wandler am Eingang des Bordnetzes bestehen. Der Schaltwandler (1) übernimmt die Funktion des Akkuladers und regelt die Akkuladespannung U_{Akku} .

Der Buck-Boost-Wandler muss seinen Eingangsstrom I_{BN} sowie den I_{Akku} begrenzen. Sind die maximalen Strombegrenzungen eingehalten, kann der Schaltwandler die Akkuspannung auf 12,6V regeln. Läuft der Buck-Boost-Wandler in die 4-Ampere-Eingangsbegrenzung, sinkt die Zwischenkreisspannung auf die Akkuspannung und der Akku wirkt unterstützend.

Für den Fall, dass die Gebläseelektronik mehr Strom benötigt, als das Bordnetz liefern kann, während der Akku geladen wird, wird der Akkuladestrom reduziert. Die Ausgangsspannung des Schaltwandlers (1) sinkt auf die Akkuspannung ab und der Ladestrom stellt sich auf den Laststrom I_{Load} ein.

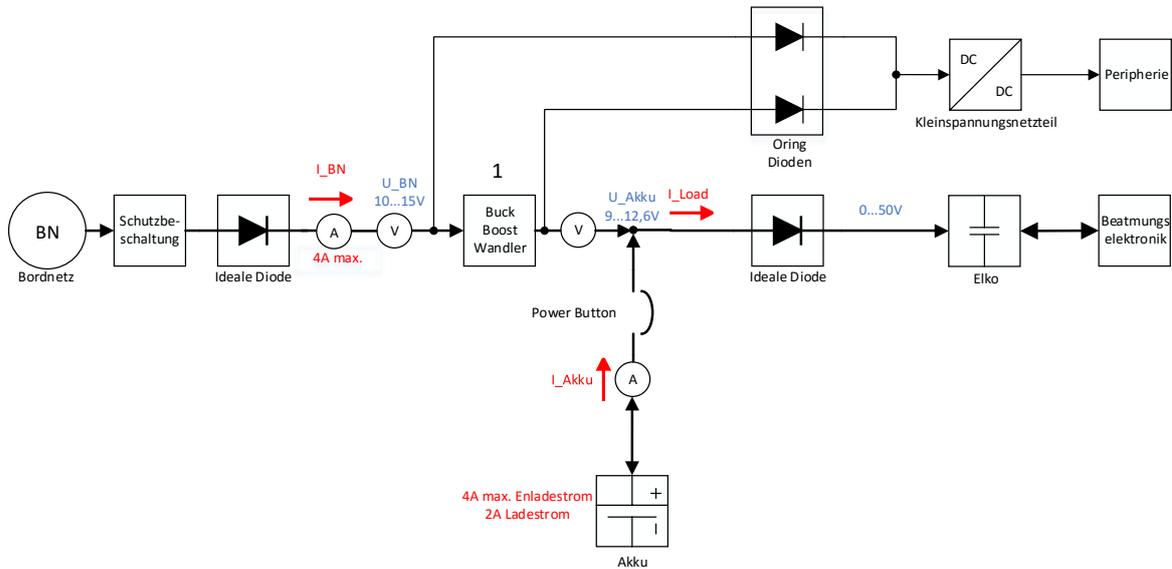


Abbildung 4.6: Konzept 2 - Buck-Boost-Wandler für den Eingang des Bordnetzes. Akku parallel zum Ausgang des Buck-Boost-Wandlers

Das Konzept stellt eine einfache Variante dar, das Energiemanagement zentral zu steuern/regeln. Der Akkuladestrom kann mithilfe eines Schaltwandlers geregelt werden. Zudem stellt sich der Akkuladestrom auf den Laststrom ein. Das hat den Vorteil, dass die Energie aus dem Bordnetz besser genutzt werden kann.

Nachteilig ist zu bemerken, dass der Strom aus dem Akku nicht begrenzt werden kann. Für einen Kurzschlussfall kann dies ein Problem darstellen.

Der Buck-Boost-Wandler muss so schnell auf Laständerungen reagieren können, dass keine ungewollten Ströme aus dem beziehungsweise in den Akku fließen.

Konzept 3

Konzept 3 ist in Abbildung 4.7 dargestellt. Der Bezeichner (1) bezieht sich im Folgenden, auf die Bezeichner in Abbildung 4.7.

Es beinhaltet im Vergleich zu Konzept 2 einen Buck-Boost-Wandler im Akkupfad (1) und keinen am Eingang des Bordnetzes. Das hat den Vorteil, dass der Strom aus dem Akku kontrolliert werden kann. Der Buck-Boost-Wandler ist synchron ausgeführt und kann auf die Akkuspannung U_{Akku} oder auf die Bordnetzspannung U_{BN} geregelt werden.

Der Eingangsstrom I_{BN} wird begrenzt, indem bei Erreichen der 4-Ampere-Grenze der Buck-Boost-Wandler auf die Bordnetzspannung U_{BN} regelt. Dies erlaubt über Potentialunterschiede die Ströme einzustellen.

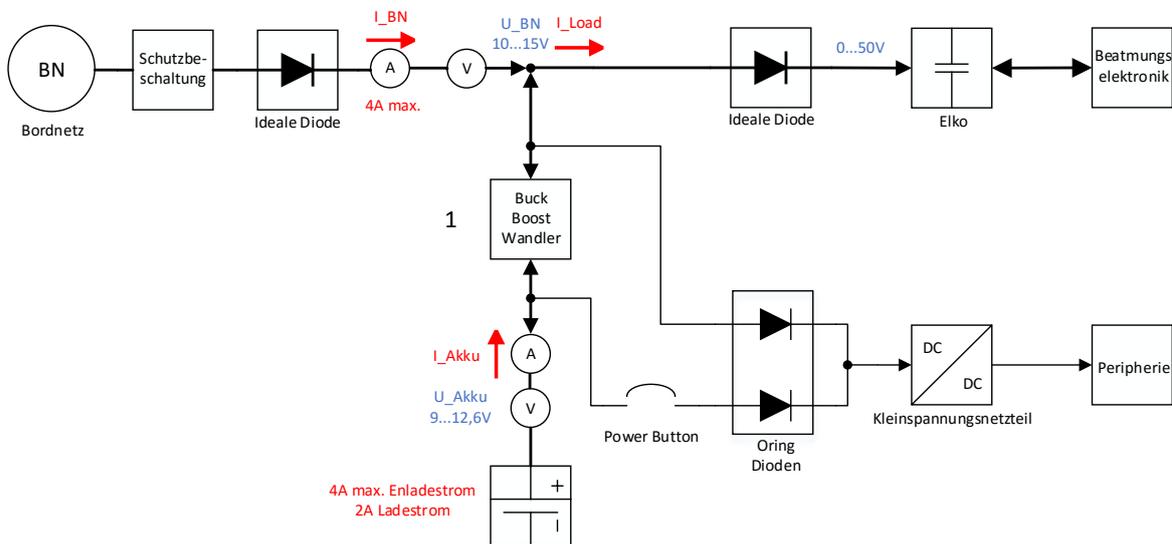


Abbildung 4.7: Konzept 3 - Buck-Boost-Wandler für den Eingang des Akkus. Bordnetz parallel zum Ausgang des Buck-Boost-Wandlers

In einem Kurzschlussfall kann der Bordnetz-Eingangsstrom nicht kontrolliert werden.

Konzept 4

Konzept 4 vereint die Überlegungen aus Konzept 2 und 3. Es ist in Abbildung 4.8 dargestellt. Die Bezeichner (1),(2) beziehen sich im Folgenden, auf die Bezeichner in Abbildung 4.5.

In diesem Konzept regeln die Boost-Wandler (1) und (2) auf dieselbe Zwischenkreisspannung U_{ZK} . Für eine Stromregelung der Eingangsströme kann ein ΔU zwischen der Ausgangsspannung der Boost-Wandler und der Zwischenkreisspannung U_{ZK} eingestellt werden.

Für eine kontrollierbare Regelung muss gelten: $U_{BN} < U_{ZK}$ und $U_{Akku} < U_{ZK}$.

Die Akkuladung wird durch Boost-Wandler (2) ermöglicht. Die Schaltausführung entspricht einem Synchronwandler (siehe Grundlagen) und lässt damit einen bidirektionalen Stromfluss zu.

Es kann beim Ladevorgang auf die Akkuspannung U_{Akku} geregelt werden. In der Betrachtungsweise entspricht der Synchronwandler (2) einem Buck-Wandler.

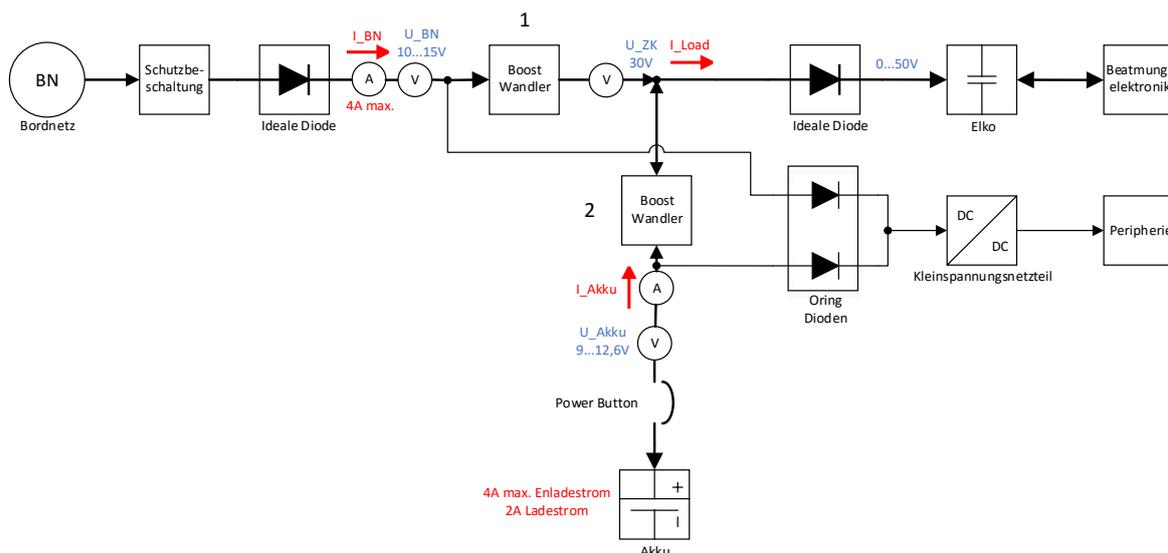


Abbildung 4.8: Konzept 4 - Boost-Wandler für die Eingangsquellen mit geregelter Zwischenspannung

Durch die Wahl einer hohen Zwischenkreisspannung können nachfolgende Leistungsstufen mit Abwärtswandlern realisiert werden, was einen Komplexitätsvorteil gegenüber den anderen Konzepten darstellt.

Die Eingangsströme können kontrolliert und begrenzt werden. Allerdings muss die Regelung auf die Zwischenkreisspannung, genau genug erfolgen, dass keine Ausgleichsströme zwischen den Versorgungsquellen auftreten. Dieses ist vor allem während des Einschaltmomentes kritisch.

4.4 Gegenüberstellung

Für die Konzeptionierung der Leistungselektronik wurden in diesem Kapitel 4 Konzepte auf der Grundlage von des Stands der Technik vorgestellt. Diese lösen das Umschalten der Versorgungsquellen und lassen eine zentrale Steuerung/Regelung des Energiemanagements zu.

Die Konzepte werden in Tabelle 4.1 gegenübergestellt, damit qualitativ eine Entscheidung für das vielversprechendste gefällt werden kann.

Tabelle 4.1: Gegenüberstellung der Hardwarekonzepte

Parameter	Konzept 1	Konzept 2	Konzept 3	Konzept 4
Leistungsverteilung	kann gesteuert werden	stellt sich ein	kann geregelt werden	kann aktiv geregelt werden
Strombegrenzung	Akkuentladung nicht steuerbar	Akkuentladung nicht steuerbar	Bordnetzstrom nicht steuerbar	kann begrenzt werden
Einschaltmoment	für Akku kritisch	für Bordnetz kritisch	für Akku kritisch	kritisch für Ausgleichsströme
DC-DC Topologie	2 Buck-Boost, 1 Charger	2 Buck-Boost	2 Buck-Boost	2 Boost 1 Buck
Anzahl Drossel-spulen	3	2	2	3
Anzahl Transistoren	10	8	8	6
Anzahl ideale Dioden	2	2	2	2
Anforderung an die Steuerung	simpel	simpel	Regelung	komplexe Regelung

4.5 Konzeptentscheidung

Die Entscheidung für eines der 4 Konzepte wurde anhand der vorgestellten Kriterien aus Tabelle 4.1 getroffen.

Die Konzepte wurden auf die geforderte Realisierbarkeit der Strombegrenzung überprüft. Diese ist nicht in jedem Konzept gegeben und kann im Fehlerfall fatale Folgen haben. Für die Qualitätssicherung in der Medizintechnik kann dies ein Ausschlusskriterium sein. Da alle Konzepte die Leistungsverteilung ermöglichen, wurde noch der Aufwand der Steuerung betrachtet. Zudem wurden die Schaltungstopologien mit ihrem Bauteileaufwand betrachtet, welche maßgeblich für die Verluste im Leistungspfad verantwortlich sind.

In Abstimmung mit dem Betreuer habe ich mich aus mehreren Punkten für das Konzept 4 entschieden. Entscheidend dabei waren einerseits die Möglichkeit den Strom zu begrenzen, andererseits der geringe Bauteileaufwand und damit eine reduzierte Komplexität. Die größere Anforderung an die Steuerung wurde in Kauf genommen, da dieser Nachteil nicht die vielen weiteren Vorteile des Konzepts überwog.

Das Konzept stellt eine neuartige Entwicklung in der Notfallmedizin dar. Es wird allen gestellten Anforderungen gerecht, um ein dezentrales Energiemanagementsystem abzulösen. In dem folgenden Praxisteil erfolgt eine nähere Untersuchung des gewählten Ansatzes.

5 Softwareentwicklung

In diesem Kapitel werden die Schritte der Softwareentwicklung beschrieben. Es wird auf die spezifischen Anforderungen eingegangen, ein Reglerkonzept entworfen und ein Modell der Regelstrecke erstellt.

5.1 Systemanforderung

Für die Systemanforderungen gelten die allgemeinen Anforderungen an das Energiemanagementsystem aus Tabelle 3.1. Hinzu kommen spezifische Anforderungen an den Mikrocontroller und das Reglersystem. Diese sind zum einen von dem Konzept der Hardware abhängig als auch von dem Regler selber.

Aufgabe der Bachelorarbeit war, die Hardware mit einem STM32-Mikrocontroller zu steuern. Der Konzeptentwurf stellt als Anforderung an den Mikrocontroller eine schnelle PWM-Ansteuerung, damit die Schaltwandler mit einem hohen Wirkungsgrad betrieben werden können. Denn der Rippelstrom durch die Drosselspule sinkt bei größerer PWM-Frequenz und damit die Verlustleistung. Je höher die PWM-Frequenz bei konstanter Timertaktung eingestellt wird, desto geringer wird die PWM-Auflösung. Hier gilt es, den schnellstmöglichen Timer zu finden, damit ein guter Kompromiss aus Regelgenauigkeit und Verluste im Leistungspfad geschlossen werden kann.

Die Auflösung der PWM ist abhängig von der Auflösung der Strom und Spannungsmessung, sprich die Auflösung der ADC(Analog-Digital-Converter)-Messwerte.

Bei internen Gesprächen hat sich herausgestellt, dass der schnellste Beatmungsalgorithmus in Bestandsgeräten mit einer Taktfrequenz von 4kHz arbeitet. Dieses lässt darauf schließen, dass ein Regelalgorithmus in dem Energiemanagement mindestens genauso schnell arbeiten muss.

Weitere Anforderungen an die Mikrocontroller-Hardware sind die gängigen Schnittstellen SPI (Serial-Peripheral Interface) für die Stromsensoren sowie genügend ADCs für die Spannungsmessungen. Desweiteren wird die SMBus-Schnittstelle, welche auf I²C (Inter-Integrated Circuit) basiert, für spätere Erweiterungen des Systems bedacht. Diese ermöglicht es, den Akkucontroller auszulesen.

5.2 Regelstrategie

Damit ein Mikrocontroller die von dem System geforderten Spannungen und Ströme einstellen kann, die sich aus den Betriebszuständen des Energiemanagementsystems ergeben, wird schnell klar, dass hier eine Steuerung nicht mehr ausreicht, da Spannungsunterschiede in der Zwischenkreisspannung zu Ausgleichsströmen in den Quellen führen. Zudem ist die Zwischenkreisspannung abhängig von der Eingangsspannung. Dementsprechend muss die Ausgangsspannung der Boost-Wandler entsprechend geregelt werden, damit eine konstante Ausgangsspannung erzielt werden kann.

Es muss eine Spannungsregelung entworfen werden, welche eine zuverlässige Energieversorgung der angeschlossenen Leistungsverbraucher sicherstellt. Zudem können durch falsche Ansteuerung der synchronen Boost-Wandler große Ausgleichsströme in den Akku entstehen. Da für die Energiequellen eine Strombegrenzung gefordert ist und der Akku nicht mit mehr als 2A geladen werden darf, muss zusätzlich der Strom geregelt werden.

Das entwickelte Reglerkonzept ist in Abbildung 5.1 dargestellt. An dieser Stelle sei auf [5] und [6] verwiesen.

Eine gängige Methode für eine Spannungsregelung mit Strombegrenzung ist eine Kaskade aus Spannungsregler und Stromregler (quelle einfügen). Bei dieser Methode erzeugt der PI-Spannungsregler aus der Differenz einer Referenzspannung U_{ZKRef} und der gemessenen Spannung U_{ZK} als Stellwert einen Stromsollwert I_{soll} .

Der Stromregler erzeugt daraufhin aus der Differenz des Sollwertes $I_{BATsoll}$ und dem gemessenen Strom I_{BAT} den Stellwert für das Tastverhältnis des Boost-Konverters D_{BAT} . Folgt man diesem Prinzip, kann dem Motorgebläse eine geregelte Spannung mit einer Eingangstrombegrenzung zur Verfügung gestellt werden, wenn nur eine Energiequelle angeschlossen ist.

Für den Fall, dass beide Energiequellen angeschlossen sind, wird zusätzlich ein zweiter Stromregler benötigt, wobei der Stellwert des Spannungsreglers für die Spannung U_{ZK} sinnvoll auf die Stromregler für die jeweiligen Boost-Konverter aufgeteilt werden muss. Dieses übernimmt der Funktionsblock F. Die Idee ist, dass, wenn die Energie zwischen Bordnetz und Akku aufgeteilt werden soll, das Bordnetz die Spannungs-Strom-Kaskade abarbeitet. Je nach Betriebszustand wird dem Akku dann ein Stromsollwert zugewiesen. Das hat zur Folge, dass der Akkustrom sich einstellt, indem die Ausgangsspannung des Boost-Wandlers relativ zur Zwischenkreisspannung eingestellt wird. Der Spannungsunterschied zwischen Ausgangsspannung und Zwischenkreis ruft einen Stromfluss hervor, welcher vom Innenwiderstand der Quelle abhängt.

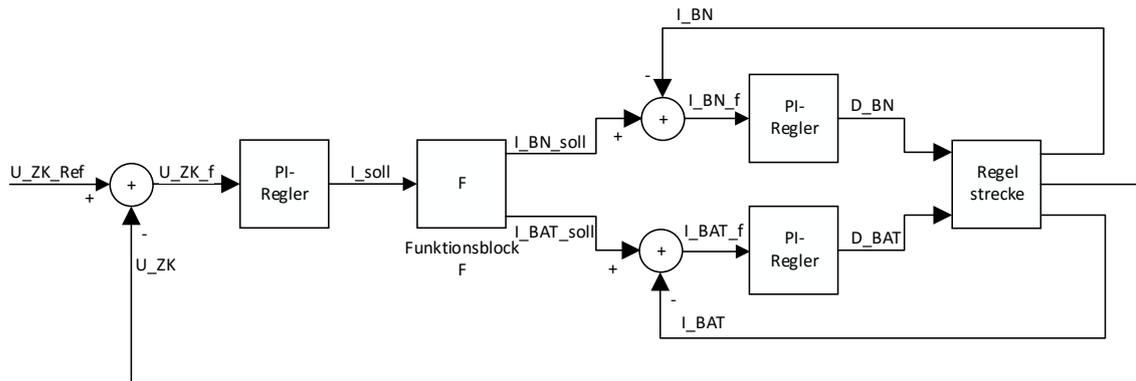


Abbildung 5.1: Reglerkonzept

Damit die geforderten Funktionen erfüllt werden und den Stromsollwert sinnvoll aufteilen, wurden für jeden Betriebsfall Zustände definiert, welche beim Abarbeiten der Regelschleife berücksichtigt werden müssen.

In den folgenden Betriebszuständen gilt: $U_{ZKRef} = 30V$

Bordnetzbetrieb

Das System befindet sich im Bordnetzbetrieb, wenn sich die Akkuspannung U_{BAT} außerhalb- und die Bordnetzspannung U_{BN} innerhalb der erlaubten Grenzen bewegt.

für

$$10V < U_{BN} < 15V \wedge U_{Bat} \leq 9V \implies I_{BNsoll} = I_{soll} \wedge I_{BATsoll} = 0A \quad (5.1)$$

Akkubetrieb

Das Pendant zum Bordnetzbetrieb bei ungültigen Bordnetzspannungen und gültigen Akkuspannungen.

$$9V < U_{BAT} < 12,6V \wedge U_{BN} < 10V \implies I_{BATsoll} = I_{soll} \wedge I_{BNsoll} = 0A \quad (5.2)$$

Ladevorgang des Akkus / Mischbetrieb

Der Mischbetrieb ist deutlich komplexer, wobei der Akku entweder geladen oder unterstützend wirken kann. Da zwei Regler in diesem Fall parallel arbeiten, muss sichergestellt werden, dass diese nicht gegeneinander arbeiten und sich aufschwingen. Von einer theoretischen Stabilitätsbetrachtung wird in dieser Arbeit allerdings abgesehen, da der Fokus der Arbeit ein anderer ist.

Beträgt der Eingangsstrom des Bordnetzes $I_{BN} < 4A$, kann der Akku mit 2 Ampere geladen werden. Erzeugt der Spannungsregler allerdings einen größeren Stromsollwert als 4 Ampere, tritt die Strombegrenzung in Kraft und der Regler kann die Zwischenkreisspannung nicht mehr aufrecht erhalten. Dieses kann man sich zu Nutze machen. Damit die Energieversorgung der Gebläseelektronik auch während des Ladens gewährleistet ist, wird diese von der Zwischenkreisspannung abhängig gemacht.

Der Ladestrom wird abhängig von der Spannung U_{ZK} linear reduziert, gemäß Formel 5.4. Dieses geschieht in den Grenzen von $28V < U_{ZK} < 29V$, in denen der Akkustrom von -2 Ampere (Stromflussrichtung: negativer Akkustrom entspricht Akkuladen) auf 0 Ampere reduziert wird. Fällt die Spannung noch weiter ab, tritt der Mischbetrieb in Kraft. In den Grenzen $26V < U_{ZK} < 28V$ wird Strom aus dem Akku von 0 Ampere auf bis zu 4 Ampere erhöht. Dies hat zur Folge, dass dem Motorgebläse keine konstante Spannung von 30V mehr zur Verfügung steht. Trotzdem steht ihm durch den unterstützenden Betrieb immer genügend Leistung zur Verfügung, auch wenn das Bordnetz nicht die volle Leistung liefern kann.

$$10V < U_{BN} < 15V \wedge 9V < U_{BAT} < 12,6V$$

$$\implies I_{BNsoll} = I_{soll} \wedge \begin{cases} 29V < U_{ZK} \implies I_{BATsoll} = -2A \\ U_{ZK} \leq 29V \implies I_{BATsoll} = f_{I_{BAT}}(U_{ZK}) \end{cases} \quad (5.3)$$

Die Funktion $f_{I_{Bat}}(U_{ZK})$ reduziert den Strom in den Akku linear, abhängig von der Spannung U_{ZK} . Die signifikanten Punkte der Funktion sind:

$$\begin{aligned} f_{I_{BAT}}(29V) &= -2A \\ f_{I_{BAT}}(28V) &= 0A \\ f_{I_{BAT}}(27V) &= 2A \\ f_{I_{BAT}}(26V) &= 4A \end{aligned}$$

$$f_{I_{BAT}}(U_{ZK}) = -2 \cdot \frac{1}{\Omega} \cdot U_{ZK} + 56A \quad (5.4)$$

Mischbetrieb bei Ladeschlussspannung

Erreicht der Akku die Ladeschlussspannung, darf dieser nicht mehr mit maximalem Lade-
strom geladen werden. Diese Arbeit fordert nur das Aufzeigen der Möglichkeit, den Akkulade-
strom zu regeln. Deshalb wird davon abgesehen, einen vollständigen Ladealgorithmus zu im-
plementieren. Bei Erreichen der Ladeschlussspannung wird der Stromsollwert $I_{BATsoll} = 0$
gesetzt, um eine Überspannung an dem Akku zu vermeiden. Reicht die Energie aus dem
Bordnetz nicht aus, um das Motorgebläse zu versorgen, tritt der Mischbetriebsfall in Kraft.
Für diesen Fall muss der Sollwert I_{soll} leistungsbehafet aufgeteilt werden.

Für die Eingangsleistung im Bordnetz gilt:

$$P_{Soll} = U_{BN} \cdot I_{soll} \wedge P_{ist} = U_{BN} \cdot I_{BN} \quad (5.5)$$

Daraus folgt bei einer 4 Ampere-Strombegrenzung für die fehlende Leistung mit:

$$P_{fel} = (I_{soll} - 4A) \cdot U_{BN} \quad (5.6)$$

Wird die noch fehlende Leistung P_{fel} positiv, muss der Akku diese zur Verfügung stellen, um
den Beatmungsbetrieb aufrecht erhalten zu können.

$$P_{fel} = P_{BATsoll} \quad (5.7)$$

Damit folgt für den Stromsollwert, wenn der Bordnetzregler in die Begrenzung läuft:

$$I_{BATsoll} = \frac{P_{fel}}{U_{BAT}} = \frac{(I_{soll} - 4A) \cdot U_{BN}}{U_{BAT}} \quad (5.8)$$

Der Mischbetriebsfall bei Ladeschlussspannung kann wie folgt definiert werden als:

$$\begin{aligned} &10V < U_{BN} < 15V \wedge U_{BAT} = 12,6V \\ \implies I_{BNsoll} = I_{soll} \wedge &\begin{cases} I_{soll} < 4A \implies I_{BATsoll} = 0A \\ I_{soll} \geq 4A \implies I_{BATsoll} = \frac{(I_{soll}-4A) \cdot U_{BN}}{U_{BAT}} \end{cases} \end{aligned} \quad (5.9)$$

5.3 Modellbildung

An dieser Stelle wird ein Modell der Regelstrecke in Matlab-Simulink aufgebaut. Es dient da-
zu, den im vorigen Kapitel entwickelten Regler zu testen und die zu erwartenden Betriebsfäl-
le zu simulieren. Matlab-Simulink bietet zudem die Möglichkeit, aus einem Simulink-Modell
mithilfe des Embedded-Coders C-Code erzeugen zu lassen. Aus diesem Grund wird das
Modell mit einer diskreten Abtaststrategie implementiert.

Abbildung 5.2 zeigt das Modell der Regelstrecke. Das Modell bildet die zwei Boost-Konverter, die Quellen sowie die Last ab. Die idealen Dioden werden nicht nachgebildet. Die Energiequellen sind als Konstanten mit einem Innenwiderstand nachgebildet. Die Last wurde als großer Elektrolytkondensator mit Innenwiderstand und dazu parallel ein variabler Lastwiderstand umgesetzt. Die Boost-Konverter sind mit allen passiven Bauelementen sowie einem idealen Boost-Wandler dargestellt, bei welchem die Ausgangsspannung von einem Tastverhältnis $D = \frac{t_{ein}}{T}$ abhängt. Zu den passiven Bauelementen gehört u.a die Eingangskapazität mit Innenwiderstand. Der Eingangspfad ist mit einer Induktivität und einem Kupferwiderstand nachgebildet. Zudem wurde ein Leitungswiderstand nachgebildet.

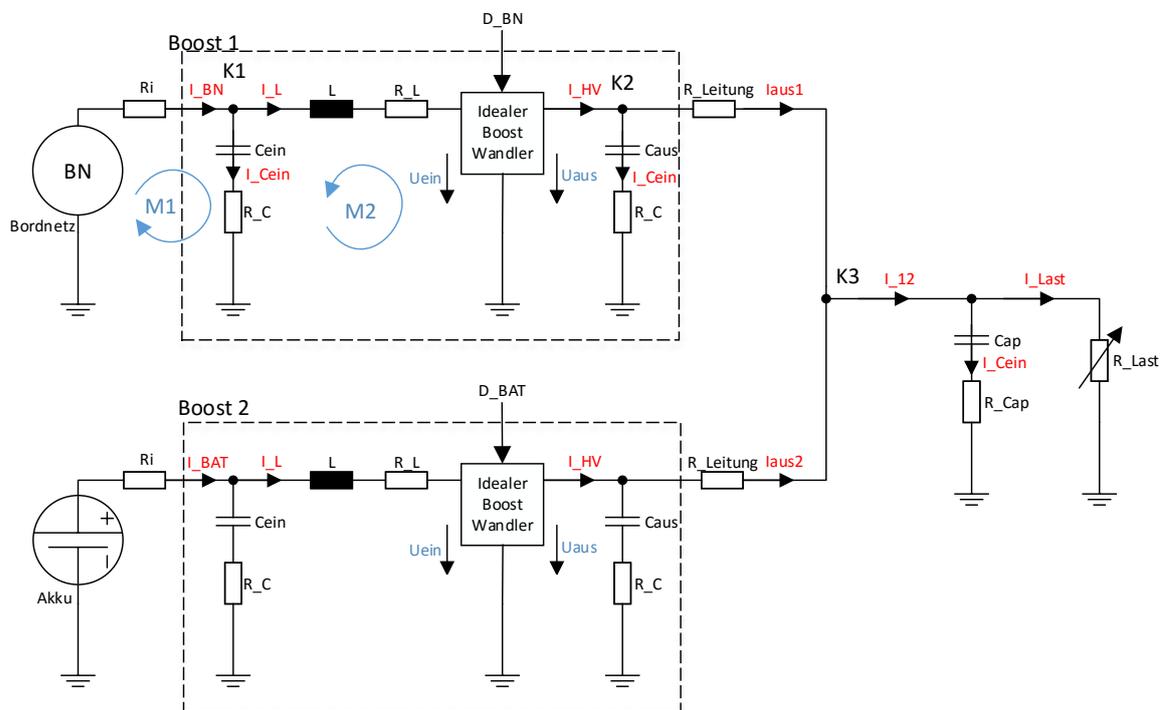


Abbildung 5.2: Modell der Regelstrecke

Damit eine Abbildung des Modells in Matlab-Simulink erfolgen kann, werden im Folgenden die Maschen und Knoten mathematisch beschrieben und auf Simulink-Blöcke reduziert. Hier sei auf [7] verwiesen.

Die idealen Boost-Wandler mit dem Übertragungsverhältnis für Strom und Spannung:

$$U_{aus} = \frac{U_{ein}}{1 - D} \quad (5.10)$$

$$I_{aus} = I_{ein} \cdot (1 - D) \quad (5.11)$$

Für den Strom am Kondensator gilt:

$$I = C \cdot \frac{dU}{dt} \implies U = \left(\frac{I}{C}\right)^2 \quad (5.12)$$

und für eine Induktivität gilt:

$$U = L \cdot \frac{dI}{dt} \implies I = \left(\frac{U}{L}\right)^2 \quad (5.13)$$

Daraus lässt sich nun mithilfe von diskreten Integratoren in Simulink ein Modell aufbauen. Die hierfür nötigen Maschen- und Knotengleichungen sind in Abbildung 5.2 zu sehen.

Masche 1:

$$U_{BN} = I_{ein} \cdot R_i + \left(\frac{I_{ein} - I_L}{C}\right)^2 + \left(\frac{I_{ein} - I_L}{R_C}\right) \quad (5.14)$$

Knoten 1:

$$I_C = I_{ein} - I_L \quad (5.15)$$

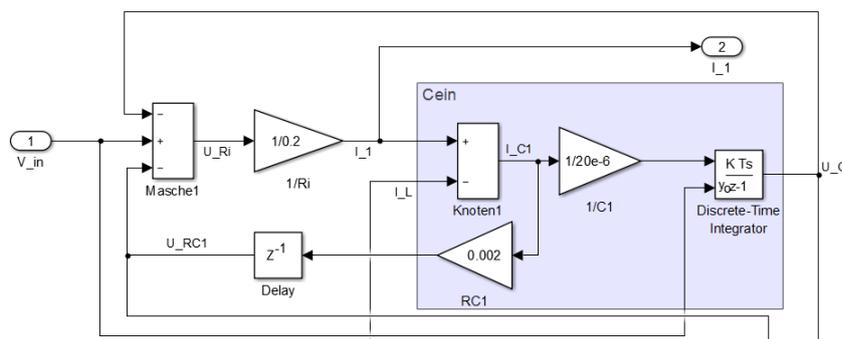


Abbildung 5.3: Simulink-Modell der Eingangskapazität eines Boost-Wandlers

Masche 2:

$$U_L = U_{RC} + U_C - I_L \cdot R_L - U_{ein} \quad (5.16)$$

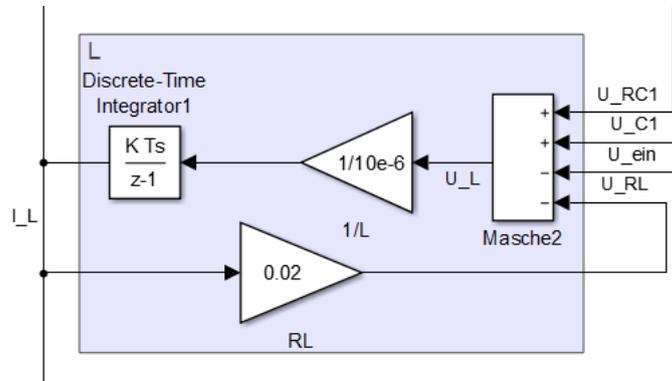


Abbildung 5.4: Simulink-Modell der Drosselspule eines Boost-Wandlers

Nach dem gleichen Prinzip wurde die Ausgangskapazität modelliert:

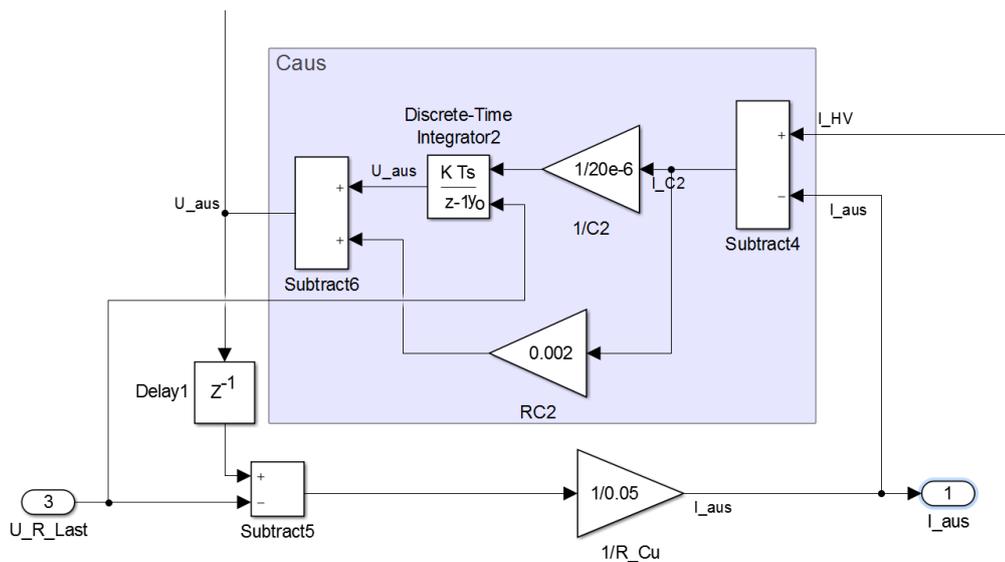


Abbildung 5.5: Simulink-Modell der Ausgangskapazität eines Boost-Wandlers

Sowie der ideale Boost-Wandler:

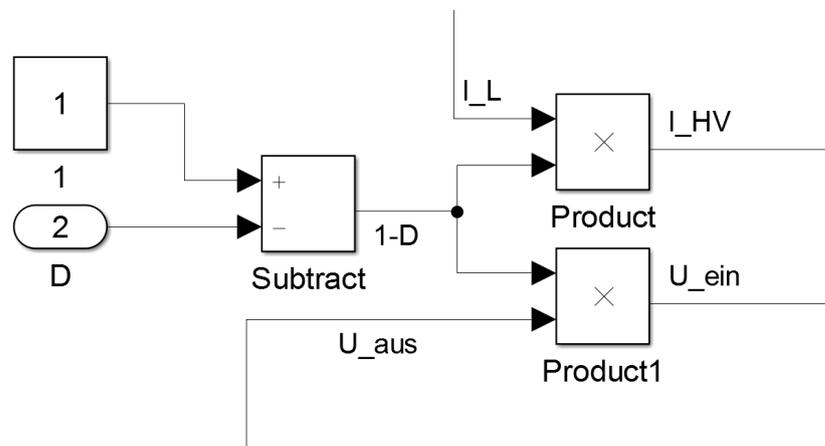


Abbildung 5.6: Simulink-Modell des idealen Boost-Wandlers

5.4 Simulation

Das Modell der Regelstrecke wird mit dem entworfenen Regler in Matlab-Simulink simuliert. Im Idealfall kann mit der Simulation das Verhalten des Energiemanagementsystems realitätsnah abgebildet werden. Zudem kann die Einstellung der Reglerparameter erfolgen und optimiert werden.

Es wurden alle Zustände des Systems auf Funktionalität überprüft. Zudem konnte ein korrektes Regelverhalten der Ströme und Spannungen erzielt werden.

Die folgenden Zustände des Systems konnten überprüft und bestätigt werden.

1. Bordnetz und Akkubetrieb erfolgt nur in ihren erlaubten Spannungsgrenzen.
3. Abschalten und Anschalten einer beliebigen Quelle während des Mischbetriebes.
3. Lastsprung auf 33 Watt und Lastsprung auf 10 Watt.
4. Akkuladen (2 Ampere): Abschalten und Einschalten in den Spannungsgrenzen des Akkus.
5. Mischbetrieb und Akkuladen: Der Akkuladestrom konnte abhängig von der Zwischenkreisspannung reduziert werden. Zudem konnte ein Mischbetrieb bei zu großer Last bestätigt werden.

Die besten Ergebnisse für die Einstellungen der Reglerparameter ergab die Simulation bei:

30V-Spannungsregler

Verstärkungsfaktor: $K_p = 5,75$

Integrationsfaktor: $I = 3,5$

Beide Stromregler

Verstärkungsfaktor: $K_p = 0,005$

Verstärkungsfaktor: $I = 10$

Für detaillierte Ergebnisse der Simulation sei auf Anhang A.3 verwiesen

5.5 Implementierung

Matlab stellt mit dem Embedded-Coder ein Tool zur Verfügung, welches erlaubt aus einem Matlab-Simulink-Modell C-Code zu erzeugen. Abbildung 5.7 zeigt den Aufbau des Simulink-Modells, aus welchem der C-Code erzeugt wurde.

Das Reglersystem und der Voltage Watchdog wurden mit dem Modell der Regelstrecke simuliert und optimiert. Matlab-Simulink erzeugte aus dem Modell einen C-Code und stellte diesen als Funktion „Step()“ zur Verfügung. Die Ein- und Ausgänge des Simulink-Modells stehen als globale Variablen zur Verfügung.

Es konnte zudem die Umrechnung der ADC-Werte sowie die Messwerte des SPI-Stromsensors in das Simulink-Modell ausgelagert werden.

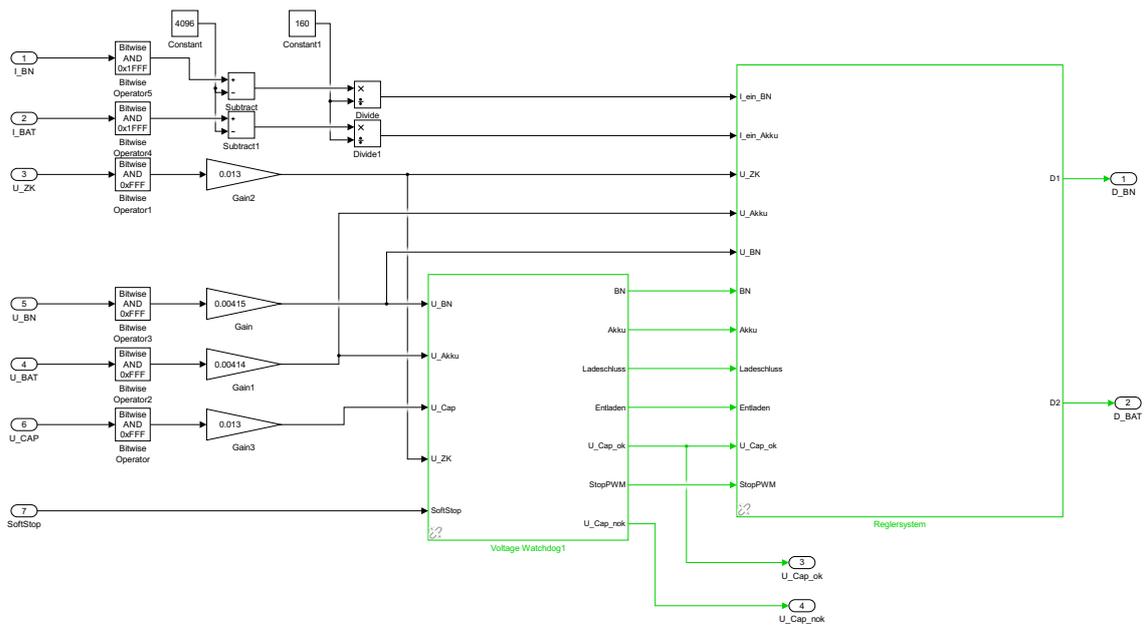


Abbildung 5.7: Simulink Modell des Idealen Boost Wandlers

Der Mikrocontroller (STM32F334) wurde mithilfe der HAL-Library programmiert, welche die Firma STMicroelectronics zur Verfügung stellt. Zudem wurde das Tool „MXCube“ verwendet, welches erlaubt, den C-Code für die Initialisierung der Peripherie zu erzeugen.

6 Hardwareentwicklung

In diesem Kapitel wird das entwickelte Hardwarekonzept spezifiziert und umgesetzt. Neben der Leistungselektronik wird für das Messen und Steuern, Peripherie und eine Kleinspannungsversorgung benötigt, welche hier ausgelegt und dimensioniert wird.

Zudem wird für den Einschaltmoment eine Elektronik entworfen, die den Strom begrenzt.

Im Anschluss erfolgt die Realisierung mit einem Schaltplanentwurf, dem Platinen-Layout und der Bestückung.

Die Hardware umfasst folgende Punkte:

- Aufwärtswandler mit Gateansteuerung
- Spannungsmessung
- Strommessung
- Ideale Dioden
- Energierückgewinnung/Speichercondensator
- Softstart-Elektronik
- Power On Elektronik
- Kleinspannungsversorgung

6.1 Dimensionierung

Aufwärtswandler mit Gatensteuerung

Damit ein hoher Wirkungsgrad erzielt werden kann, muss der Aufwärtswandler sorgfältig ausgelegt werden. Zudem wird versucht, das Layout so kompakt wie möglich zu halten. Geschaltete Ströme können schnell auf andere Funktionsgruppen einkoppeln und deren Funktionalität stören. Als Beispiel sei hier eine verfälschte Spannungsmessung genannt. Wie im

Grundlagenkapitel 2.3 angeführt, tragen vor allem die Speicherdrossel sowie die Transistoren maßgeblich zu den Verlusten bei. Der Dimensionierung der Speicherdrossel wird somit eine große Aufmerksamkeit geschenkt.

Dimensionierung der Speicherdrossel

Wie in Kapitel 2.6 beschrieben, kann die Speicherdrossel nach der Formel

$$L = \frac{U_{ein} \cdot (U_{aus} - U_{ein})}{\Delta I \cdot f \cdot U_{aus}} \quad (6.1)$$

ausgelegt werden. Die Induktivität ist somit von der Ein- und Ausgangsspannung sowie von der Schaltfrequenz und dem Ripplestrom abhängig.

Die Berechnung erfolgt mit den Werten für den größten Spannungshub zwischen Ein- und Ausgangsspannung, da für diesen Fall der Sättigungsstrom der Spule nicht überschritten werden darf.

Der Sättigungsstrom wird beispielsweise von der Firma Würth Elektronik mit einer um 10% verminderten Induktivität angegeben. Sinkt die Induktivität, muss im Gegenzug der Ripplestrom ansteigen, um die Ladung der Speicherdrossel auf die Ausgangskondensatoren umzuladen. Ein größerer Strom durch die Spule verursacht somit eine größere Verlustleistung in den ohmschen Widerständen.

Der größte Spannungshub liegt für die Eingangsspannung des Akkus bei 9V. Seine Ausgangsspannung pendelt um den Arbeitspunkt bei 30V. Die Frequenz f wurde auf 250kHz festgelegt, welches mit dem gewählten Mikrocontroller erlaubt, die Spannungsmessung (12 bit) auf eine PWM-Auflösung mit 12 bit umzusetzen.

Der Strom durch den Boost-Wandler setzt an dem Ersatzreihenwiderstand der Spule und an den Transistoren Verlustleistung um. Der quadratische Zusammenhang aus Strom und Widerstand macht deutlich, dass es Sinn macht den Ripplestrom zu optimieren, anstatt den Innenwiderstand der Drosselspule. Der Ripplestrom ist ein AC-Strom, welcher sich auf den Gleichstrom durch die Spule aufaddiert.

Neben der Induktivität, ist der Ripplestrom die unbekannteste Größe. Abbildung 6.1 zeigt einen Matlab-Plot des Ripplestroms über der Induktivität, bei Ausgangsspannungen um den Arbeitspunkt.

Bei der Auswahl einer Drosselspule ist es ratsam die Induktivität nicht zu groß zu wählen, da für eine größere Induktivität schnell die Baugröße und die Kosten steigen. Zudem muss für eine größere Induktivität mehr Draht aufgewickelt werden, was einen größeren Innenwiderstand bedeutet. Nach Angabe von [3] und [4] sollte der Drosselstrom auf maximal 50% des Stromes durch die Spule ausgelegt werden. Bei einem maximalen Eingangsstrom von 4A sollte der Ripplestrom demnach nicht größer als 2A werden.

Die Firma Würth Elektronik hat spezielle Drosselspulen in Ihrem Sortiment, welche geeignete Kernmaterialien aufweisen, damit ein geringer Bauraum bei ausreichend großer Induktivität erzielt werden kann. Anhand des Graphen aus Abbildung 6.1 wurde sich für eine $22\mu H$ mit 6A Sättigungsstrom und einem Serienwiderstand von $24,7m\Omega$, des Typs HCI-SMD entschieden. Diese bietet bei einem Spulenstrom von 5A ($4A + \frac{\Delta I}{2}$) noch eine Induktivität von $16\mu H$ und ist damit geeignet.

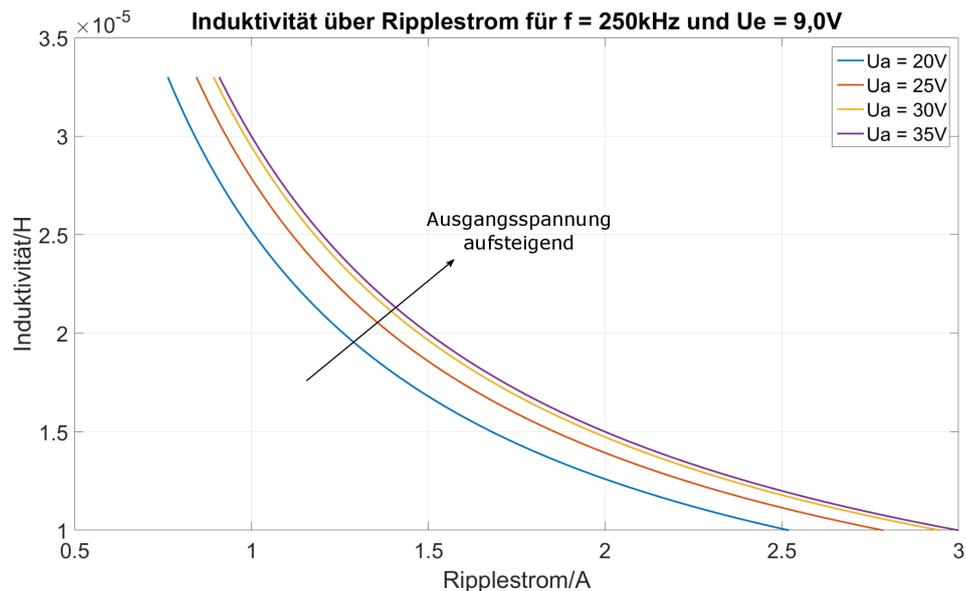


Abbildung 6.1: Induktivität des Boost-Wandlers über dem Ripplestrom bei veränderlichen Ausgangsspannungen

Spannungsmessung

Die Spannungsmessung erfolgt über die internen ADCs des STM32F334. Die Spannungen werden über einen Spannungsteiler und einen passiven Tiefpass 1. Ordnung mit einer Grenzfrequenz von 4kHz dem Mikrocontroller zugeführt.

Die Platine sieht zudem noch Impedanzfolgerschaltungen mit Operationsverstärkern vor, welche allerdings nicht bestückt wurden.

Strommessung

Für die Strommessung wurden SPI-Hall-Sensoren verwendet. Dies hat den Vorteil, dass kein Shunt zur Strommessung verwendet werden muss und damit weniger Verlustleistung auftritt.

Zudem lässt die SPI-Schnittstelle eine einfachere Inbetriebnahme zu. Der Sensor hat eine Auflösung von 6,25mA und eine Bandbreite von 18kHz. Da der Regler mit 8kHz läuft und

damit ein abgetastetes System ist, ist das Nyquist-Kriterium verletzt und es muss die Bandbreite mit einem digitalen Filter reduziert werden.

Ideale Dioden

Ideale Dioden wurden nach dem gewählten Konzept eingebaut. Im Endeffekt konnte aber auf die ideale Diode für den Bordnetzpfad verzichtet werden, da sich die Regelung als schnell genug erwiesen hat, um die Ströme einzustellen.

Energierückgewinnung/Speicherkondensator

Damit die Energierückgewinnung aus dem Leistungspfad gewährleistet werden kann, wurden Pufferkondensatoren eingesetzt. Wird Energie in einen Kondensator geladen, steigt seine Spannung. Damit 10 Joule gespeichert werden können, muss der Pufferkondensator ausreichend groß dimensioniert werden. Die Kondensatorspannung, nachdem dieser 10 Joule aufgenommen hat, muss zudem von der vorgeschalteten Diode ausgehalten werden.

Die Energie am Kondensator berechnet sich mit

$$W = \frac{1}{2} \cdot C \cdot U^2. \quad (6.2)$$

Die Entscheidung fiel auf 2 Elektrolytkondensatoren mit jeweils $4700\mu F$ parallel. Diese sind günstig, haben einen vergleichsweise kleinen ESR und sind kurzfristig auf bis zu 100V spannungsfest. Im Normalbetrieb sind die Kondensatoren auf 30V vorgeladen und damit auf

$$W_N = \frac{1}{2} \cdot 9400\mu F \cdot 30V^2 = 4,23J. \quad (6.3)$$

Beim Motor-Bremsvorgang fließt der Strom in den Kondensator und die Energie erhöht sich auf

$$W_{max} = W_N + W_{Brems} = 4,23J + 10J = 14,23J \quad (6.4)$$

und damit die Spannung am Kondensator auf

$$\Rightarrow U_{max} = \sqrt{\frac{W_{max} \cdot 2}{C}} = \sqrt{\frac{14,23J \cdot 2}{9400\mu F}} = 55,02V. \quad (6.5)$$

Somit ist die Spannungsfestigkeit bei Aufnahme der Bremsenergie gewährleistet.

Einen weiteren Vorteil bietet das Speicherelement beim Lastverhalten. Stromspitzen, welche größer als die mittlere Leistungsabgabe des Energiemanagements sind, werden geglättet. Zudem hat die Regelung mehr Zeit, auf Spannungsänderungen zu reagieren.

Softstart-Elektronik

Für den Einschaltmoment stellen die Elektrolytkondensatoren eine kritische Last dar. Die Kondensatoren sind im ersten Moment nicht geladen und würden einen unzulässigen Eingangstrom der Quellen verursachen.

Damit der Strom im Einschaltmoment begrenzt werden kann, wurde eine Schaltung entwickelt, welche es mikrocontrollergesteuert erlaubt, die Elektrolytkondensatoren zu überladen.

Die Schaltung ist in Abbildung 6.2 dargestellt. M1 sperrt im Einschaltmoment, damit der Strom in die Kondensatoren über $R_{Start} = 200\Omega$ fließen muss.

Schaltet der Mikrocontroller über das Signal „Keep Bat Alive“, den MOSFET M1 ein, entlädt sich das Gate von M1 und dieser wird leitend.

D1 ist eine Zener-Diode, welche verhindert, dass an M1 eine Spannung $U_{GS} < -20V$ anliegt. Dieses verhindert eine Überspannung des MOSFETs.

Die Bauteile R1, R2 und D2 wurden so gewählt, dass keine zu großen Gateströme auftreten und der MOSFET M1 sein Gate schnell genug aufladen kann, um einen Strom in die Kondensatoren zu sperren.

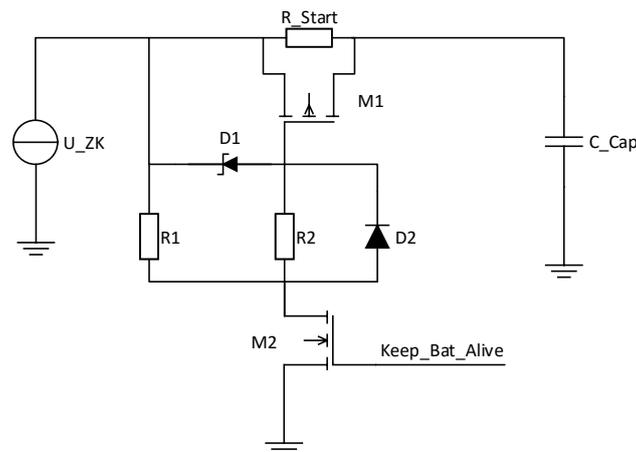


Abbildung 6.2: Einschaltelctronik für die Strombegrenzung der Elektrolytkondensatoren

Power-On-Elektronik

Es wurde eine Power-On-Elektronik entwickelt, welche es ermöglicht, mittels Taster das System im Akkubetrieb zu starten. Dies hat den Vorteil, dass der Akku sich nicht entlädt, wenn das System aus ist.

Wird das Bordnetz eingeschaltet, fährt das System ohne Tasterbetätigung hoch, damit der Akku geladen werden kann.

Kleinspannungsversorgung

Die Kleinspannungsversorgung besteht aus einem 5V-Buck-Wandler und einem 12V-Boost-Wandler, welche als ICs ausgeführt wurden.

Der 5V-Buck-Wandler wird aus dem Akku oder dem Bordnetz gespeist und versorgt den Mikrocontroller und den 12V-Buck-Wandler. Über einen Jumper kann die 5V-Spannung auch über eine USB-Buchse zur Verfügung gestellt werden.

Die 12V stellt die Spannung für die Gate-Treiber zur Verfügung.

Die Versorgung der SPI-Stromsensoren übernimmt der 3,3V Spannungsregler des Mikrocontrollers.

6.2 Umsetzung

Die bestückte Platine mit dem aufgesteckten STM32F334 Nucleo Development Board ist in Abbildung 6.3 zu sehen.

Der Schaltplan wurde mit Altium Designer entworfen. Dieser ist in Anhang B.1 eingefügt. Für eine detaillierte Betrachtung des Schaltplans, sei an dieser Stelle auf die digitale Form der Arbeit verwiesen, welche eine bessere Betrachtung ermöglicht.

Das Platinenlayout wurde mit dem Programm: Circuit Studio entworfen. Da die Hardware einem Laboraufbau entspricht, lagen keine Anforderungen an die Platinenmaße vor. Der Entwurf wurde in einem 4-lagigen Layout umgesetzt. Hier sei auf Anhang B.2 verwiesen.

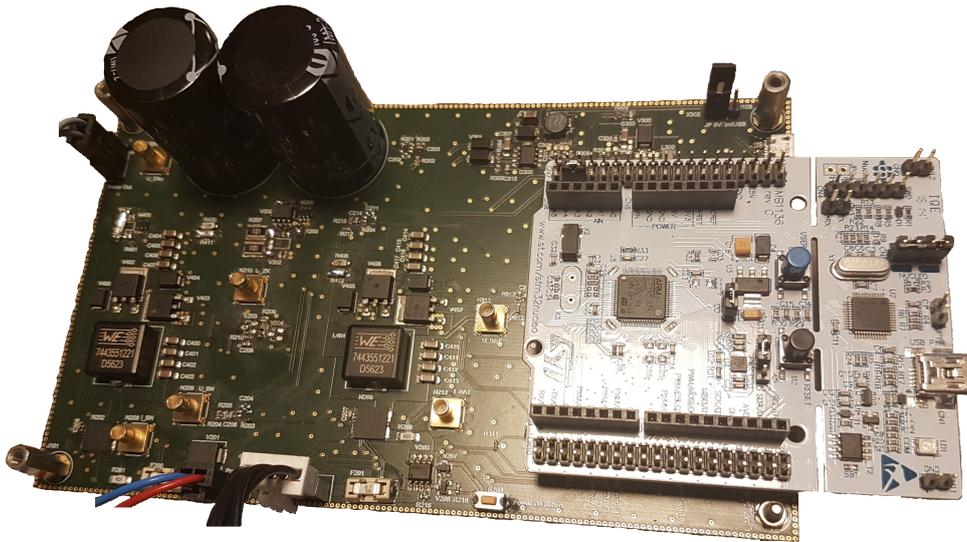


Abbildung 6.3: Versuchshardware

7 Verifikation und Diskussion

In diesem Kapitel wird die Inbetriebnahme und die Verifikation der Leistungselektronik dokumentiert. Es werden die anforderungsspezifischen Parameter überprüft und das Verhalten der Regelung beschrieben.

Alle Messungen werden mit einer Bandbegrenzung von 20kHz aufgenommen, damit das Verhalten des Systems nicht von den Störungen der Schaltwandler überlagert wird. Die Reglerfrequenz und damit das Regelverhalten, wird durch die Bandbegrenzung nicht beeinträchtigt.

7.1 Versorgungsquellen

Laut den Anforderungen aus Tabelle [3.1](#) sind die Versorgungsquellen für einen bestimmten Spannungsbereich und einen maximalen Strom definiert. An dieser Stelle wird die Einhaltung der geforderten Parameter überprüft.

Eingangsspannungsbereich

Der Eingangsspannungsbereich wird nach dem Aufbau in Abbildung 7.1 gemessen.

Das korrekte Systemverhalten der Eingangsspannungen kann überprüft werden, indem die Regelung der Zwischenkreisspannung U_{ZK} betrachtet wird. Beträgt der Eingangsstrom der Quelle < 4 Ampere und die Zwischenkreisspannung konstant 30V, wird die Eingangsspannung vom System als zulässig erachtet. Umgekehrt muss die Zwischenkreisspannung einbrechen, wenn der gültige Eingangsspannungsbereich verlassen wird.

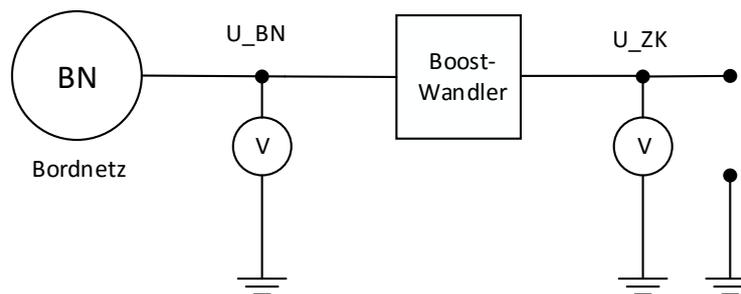


Abbildung 7.1: Blockschaltbild des Messaufbau

Die Eingangsspannungen der Quellen wurden mittels Multimeter gemessen und sind in Tabelle 7.1 dargestellt.

Der Zeitpunkt zur Aufnahme des Messwertes war der Zeitpunkt, bei dem die Zwischenkreisspannung U_{ZK} einbricht und nicht mehr auf 30V geregelt wurde.

Die Werte entsprachen den Anforderungen der in Matlab-Simulink hinterlegten Schaltschwellen. Laut Anforderung hat der Akku eine Ladeschlussspannung von 12,6V. Damit beim Erreichen der Ladeschlussspannung nicht in den Bordnetzbetrieb gewechselt wird, wurde die obere Schaltschwelle auf 15V gesetzt und der Zustand „Mischbetrieb bei Ladeschlussspannung“ wird aktiv.

Das erlaubt einen Mischbetrieb, wenn der Akku seine maximale Spannung erreicht und nicht mehr von der Zwischenkreisspannung abhängt.

Energiequelle:	untere Schaltschwelle	obere Schaltschwelle
Bordnetz	9,99V	15,89V
Akku	9,05V	14,94

Tabelle 7.1: Abschaltschwellen der Spannungsquellen

Strombegrenzung:

Der maximale Eingangsstrom für beide Quellen ist auf 4 Ampere festgelegt worden. Abbildung 7.2 zeigt den prinzipiellen Messaufbau für das Bordnetz.

Gemessen wurde: U_{BN} , I_{BN} , U_{Cap}

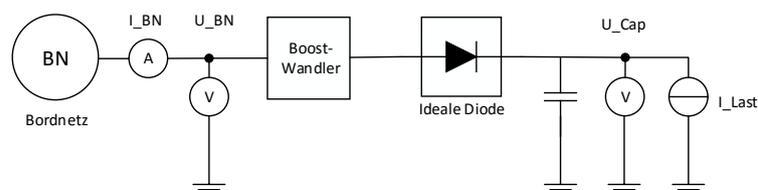


Abbildung 7.2: Blockschaltbild zur Messung der Strombegrenzung

Da beide Aufwärtswandler symmetrisch aufgebaut sind, reicht die Messung an einem Boost-Wandler. Abbildung 7.3 zeigt exemplarisch das Bordnetz mit einem Lastsprung, welcher eine Eingangs-Strombegrenzung hervorruft. Hierfür wird ein Strom mit einer elektronischen Last eingestellt, welcher das System in die Strombegrenzung zwingt:

Einstellungen für die Eingangsspannung und den Lastsprung:

Eingangsspannung: $U_{BN} = 12V$

Lastsprung: $I_{Last1} = 0,37A(10W) \implies I_{Last2} = 1,51A(40,3W)$

In dem Moment des Lastsprungs ist zu erkennen, dass der Strom sprunghaft auf 4 Ampere steigt und die Ausgangsspannung langsam einbricht, da die in den Elektrolytkondensatoren gespeicherte Energie die Versorgung der Last stützt. Die Strombegrenzung arbeitet korrekt und hat zur Folge, dass die Ausgangsspannung bei zu großer Last nicht mehr aufrecht erhalten werden kann.

Ferner lässt sich ein Spannungseinbruch an der Eingangsspannung U_{BN} zum Zeitpunkt des Lastsprungs erkennen. Der Grund hierfür, lässt sich mit den ohmschen Verlusten über dem Versorgungskabel des Netzeils und dem Strommesswiderstand auf der Platine erklären. Fließt ein größerer Strom, muss über den ohmschen Widerständen auch eine größere Spannung abfallen.

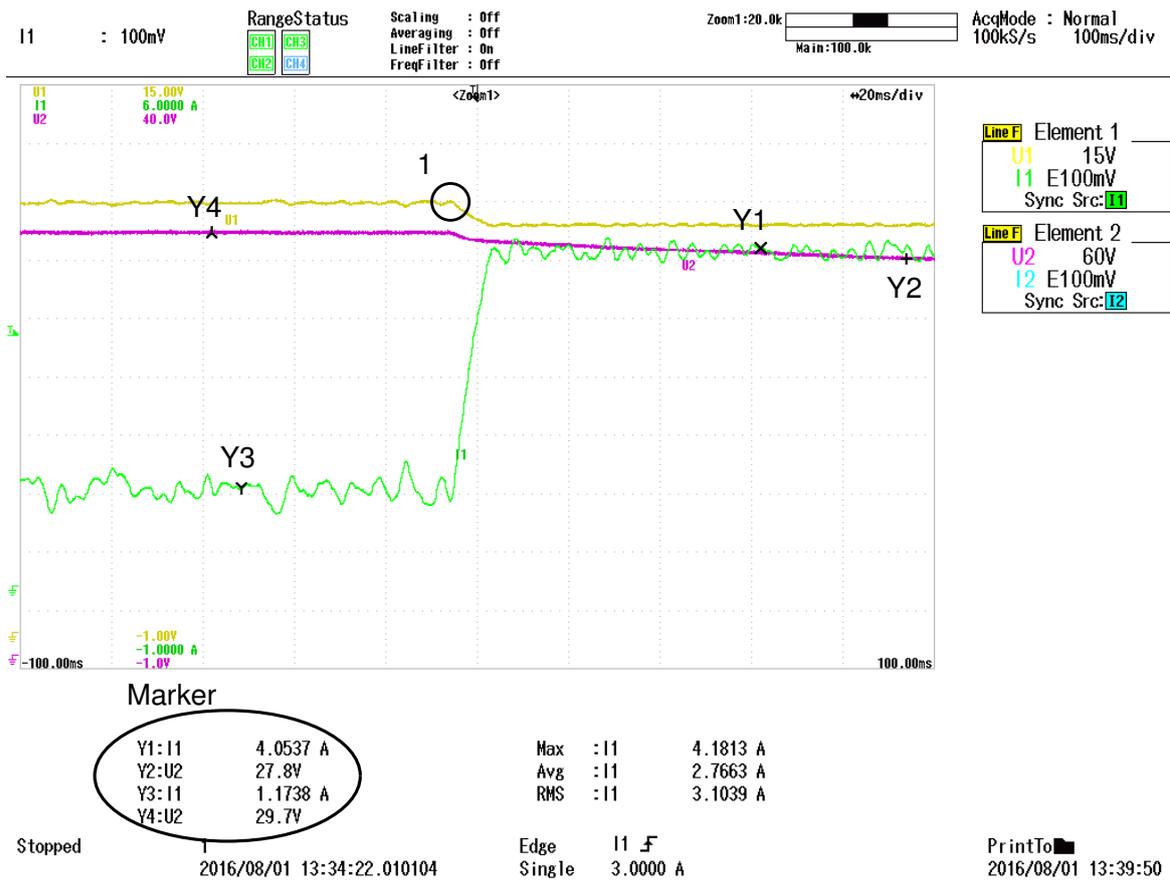


Abbildung 7.3: Strombegrenzung des Bordnetzes

$U1 = U_{ZK}$ (Gelb), $U2 = U_{Cap}$ (Lila), $I1 = I_{BN}$ (Grün)

Marker Y1-Y4 beziehen sich auf die in der Grafik angegebenen Spannungs- und Stromverläufe

Primäre Sekundäre Versorgungsquelle

Das Verhalten der Versorgungsquellen wird priorisiert gesteuert. Die primäre Versorgungsquelle (Bordnetz) übernimmt, wenn möglich, die Versorgung des gesamten Systems. Die sekundäre Energiequelle (Akku) versorgt nur für den Fall, dass kein Bordnetz zur Verfügung steht oder das Bordnetz Unterstützung bei der Versorgung der Last benötigt.

Abbildung 7.4 zeigt den prinzipiellen Messaufbau.

Gemessen wurde: U_{ZK} , I_{BN} , I_{BAT} .

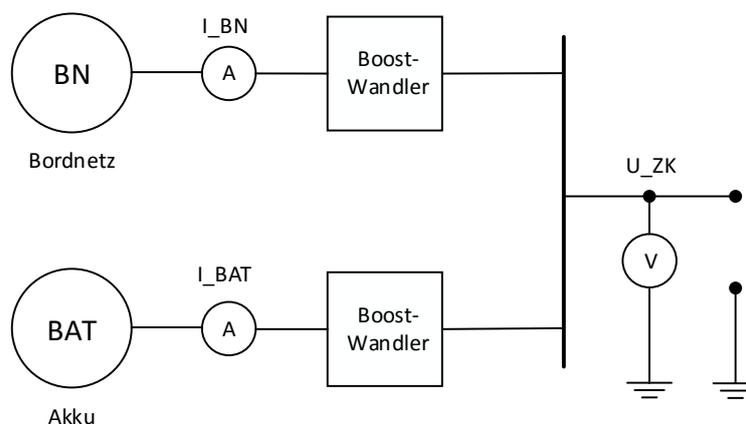


Abbildung 7.4: Blockschaltbild zu den Messungen des Quellenverhaltens im Zusammenspiel

Steht kein Bordnetz zur Verfügung und der Akku übernimmt die Last, wird die Leistungselektronik von der sekundären Energiequelle versorgt. Abbildung 7.5, zeigt das Verhalten, wenn das Bordnetz eingeschaltet wird. Zudem wurde die Messung ohne Last durchgeführt, da dies den kritischsten Fall darstellt. Die Regelung muss ausreichend schnell den Zustand umschalten, damit die Zwischenkreisspannung aufrecht erhalten werden kann und keine großen Ausgleichsströme fließen können.

Es gilt:

$$U_{BN} = 0 \implies U_{BN} = 12V \text{ und } U_{BAT} = 12V$$

Während des Einschaltvorgangs von I_{BN} ist ein Einschaltimpuls bei $I_{BN}=4,3712A$ zu erkennen. Dieser übersteigt für 2ms die Spezifikation. Der Strom und die Spannung benötigen 30ms, um sich einzuschwingen. Der Akkustrom I_{BAT} schwingt entgegengesetzt zu dem Bordnetzstrom. Die Zwischenkreisspannung bleibt während des Schaltmomentes bei $\pm 1V$ konstant.

Wird das Bordnetz eingeschaltet, wird auf den Zustand Mischbetrieb umgeschaltet und der Akku wird mit $-2A$ geladen. Das Bordnetz ist die primäre Energiequelle.

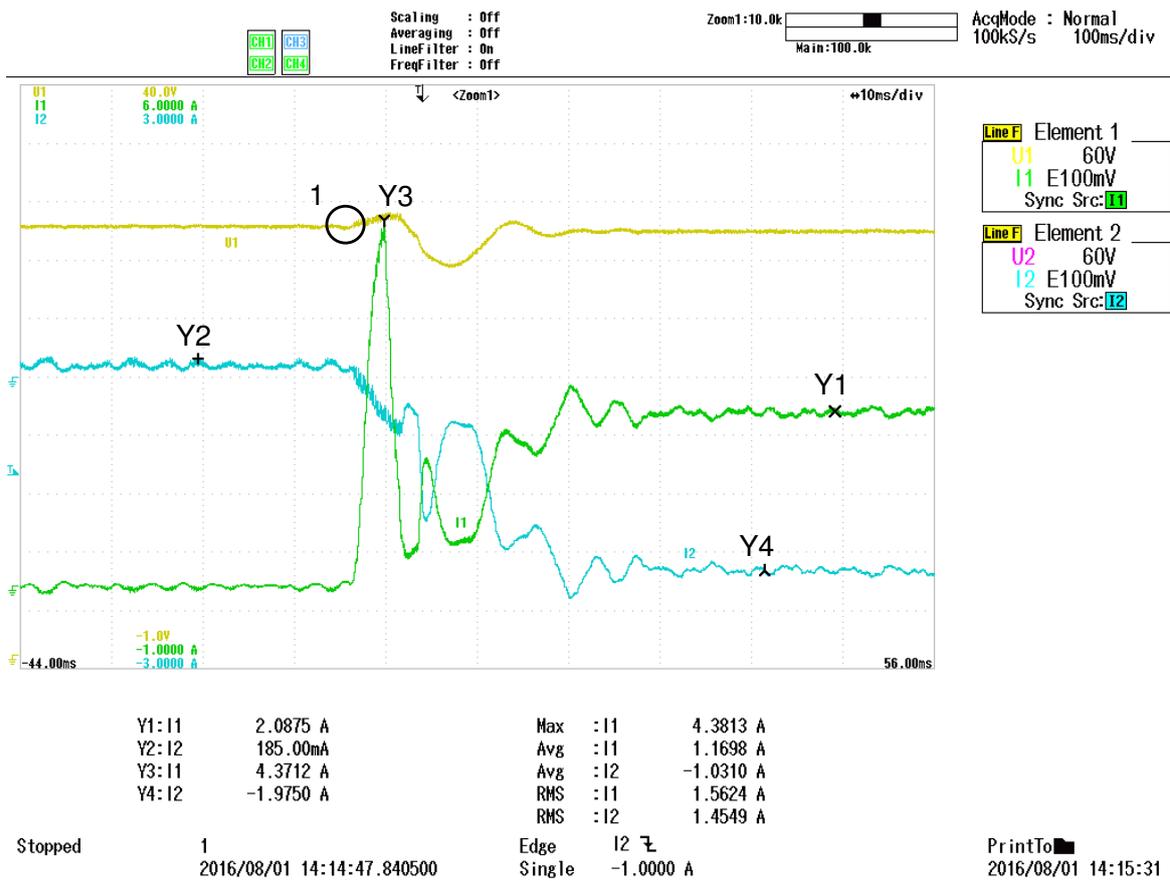


Abbildung 7.5: Wechsel vom Akkubetrieb in den Mischbetrieb

$U1 = U_{ZK}$ (Gelb), $U2 = U_{Cap}$ (Lila), $I1 = I_{BN}$ (Grün), $I2 = I_{BAT}$ (Türkis)

Marker Y1-Y4 beziehen sich auf die in der Grafik angegebenen Spannungs- und Stromverläufe

Abbildung 7.6 zeigt das Abschalten des Bordnetzes während des Mischbetriebs. Der Akku übernimmt die Versorgung der Leistungselektronik.

Im Schaltmoment tritt eine Stromspitze bei Y1 auf, welche mit 2,1 Ampere kleiner als der maximal erlaubte Strom ist.

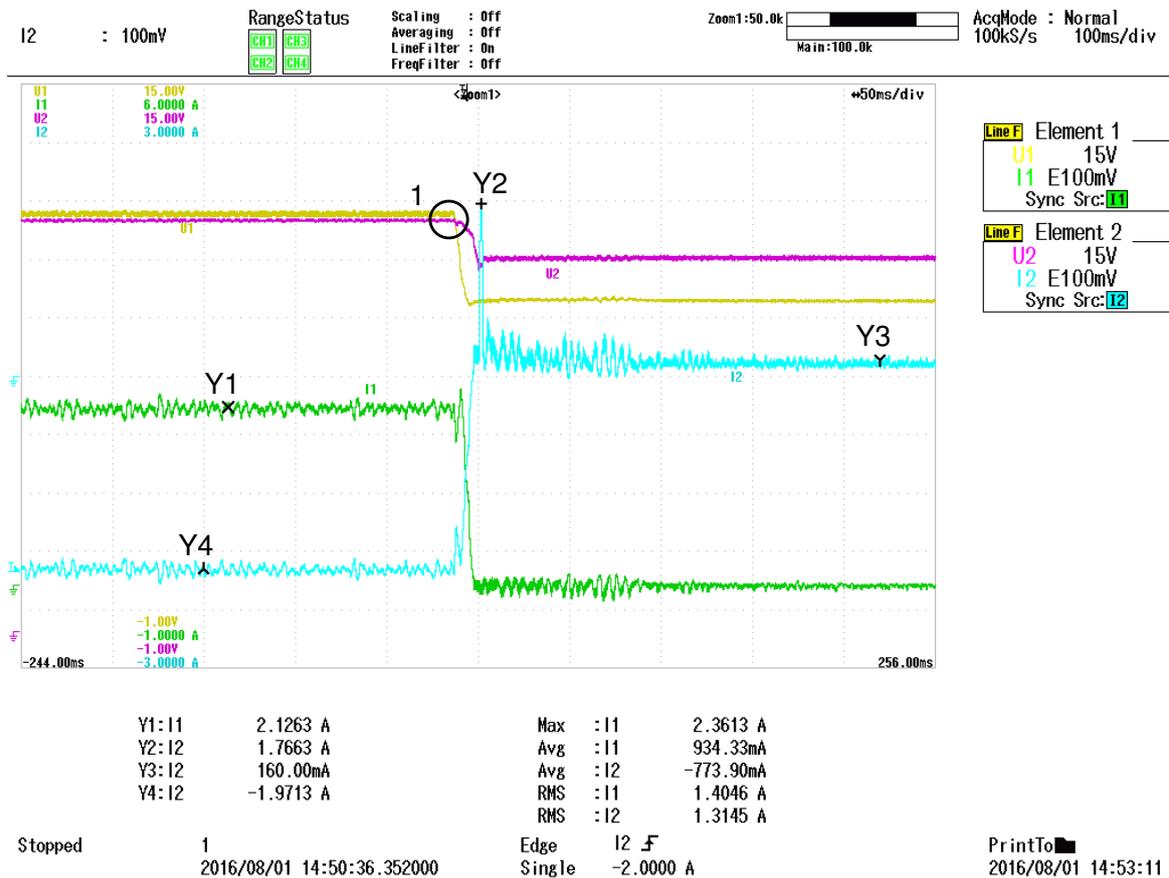


Abbildung 7.6: Wechsel vom Mischbetrieb in den Akkubetrieb

$U1 = U_{ZK}$ (Gelb), $U2 = U_{Cap}$ (Lila), $I1 = I_{BN}$ (Grün), $I2 = I_{BAT}$ (Türkis)

Marker Y1-Y4 beziehen sich auf die in der Grafik angegebenen Spannungs- und Stromverläufe

7.2 Konstante Energieversorgung des Beatmungsgebläses

Die konstante Versorgung des Beatmungsgebläses ist die Hauptaufgabe des Energiemanagementsystems. Sie soll einen Lastsprung von einer Grundlast (10W) auf eine Vollast (33 Watt) ausregeln können. Dieses muss bei der kleinsten Eingangsleistung gewährleistet sein. Die kleinste Eingangsleistung steht zur Verfügung, kurz bevor der Akku seine Entladespannung erreicht, mit $U_{BAT} = 9V$. Abbildung 7.7 zeigt den Messaufbau als Blockschaltbild.

Gemessen wurde: U_{BAT} , I_{BAT} , U_{Cap}

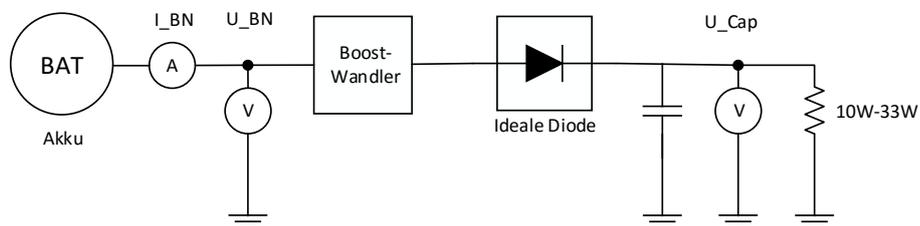


Abbildung 7.7: Blockschaltbild zu den Messungen der Lastsprünge

Abbildung 7.8 zeigt den Lastsprung auf 33,3 Watt. Die Versorgung erfolgt an der Leistungsgrenze und kann nicht lange aufrecht erhalten werden, da der Akku kurz vor dem Erreichen der Entladespannung steht.

Hiermit kann gezeigt werden, dass die Versorgung des Motorgebläses über den kompletten zulässigen Eingangsspannungsbereich gewährleistet ist.

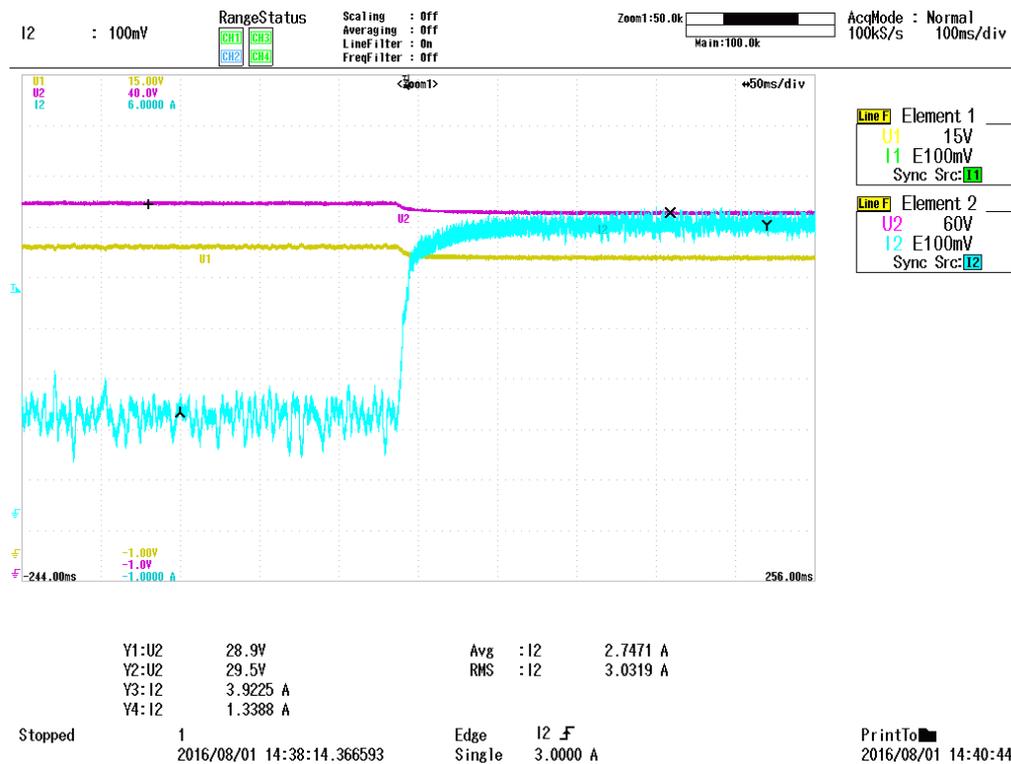


Abbildung 7.8: Lastsprung bei der kleinsten Akku-Eingangsleistung

$U1 = U_{BN}$ (Gelb), $U2 = U_{Cap}$ (Lila), $I2 = I_{BAT}$ (Türkis)

Marker Y1-Y4 beziehen sich auf die in der Grafik angegebenen Spannungs- und Stromverläufe

7.3 Akku Laden und Mischbetrieb

An dieser Stelle wird gezeigt, dass der Strom vom Energiemanagementsystem über den gültigen Spannungsbereich eines Akkus geregelt werden kann.

Zunächst wird gezeigt, dass bei Erreichen der Ladeschlussspannung sowie der Entladespannung der Strom in dem Akku von seinem maximalen Ladestrom auf seinen minimalen Ladestrom reduziert werden kann. Zudem wird gezeigt, dass die Akkuladung als sekundäre Last implementiert ist und die primäre Last (Motorgebläse) trotz des Akkuladens immer versorgt werden kann. Dazu wird die Zwischenkreisspannung aufgenommen und auf korrektes Verhalten zum Reduzieren des Akkuladestroms überprüft.

Der zweite Teil der Messungen bezieht sich auf den Mischbetrieb. Das Reduzieren des Akkuladestroms wird abhängig von der Spannung U_{ZK} so weit reduziert, dass ein Mischbetrieb die Folge ist.

Der Messaufbau erfolgt nach Abbildung 7.9.

Gemessen wurde: U_{BN} , I_{BN} , U_{BAT} , I_{BAT}

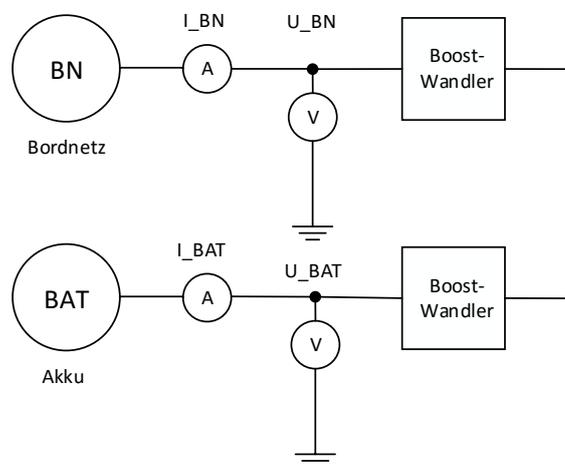


Abbildung 7.9: Blockschaltbild zu den Messungen - Akku Laden und Mischbetrieb

Es gilt: $U_{BN} = 12V$

Akku Laden über den Akkuspannungsbereich:

Abbildung 7.10 zeigt das Anschalten des Ladevorganges bei Überschreiten der Entladespannung $U_{BAT} = 9V$.

Die Akkuspannung wird von der Spannung Marker Y1 = 9,01V auf Y2 = 9,34V erhöht. Die Folge ist, dass ein Akku als gültig erkannt wird und das Bordnetz beginnt zu laden. Die Bordnetzeingangsspannung bricht ein, weil ein erhöhter Stromfluss über das Versorgungskabel fließt und über dem Strommesswiderstand Spannung abfällt.

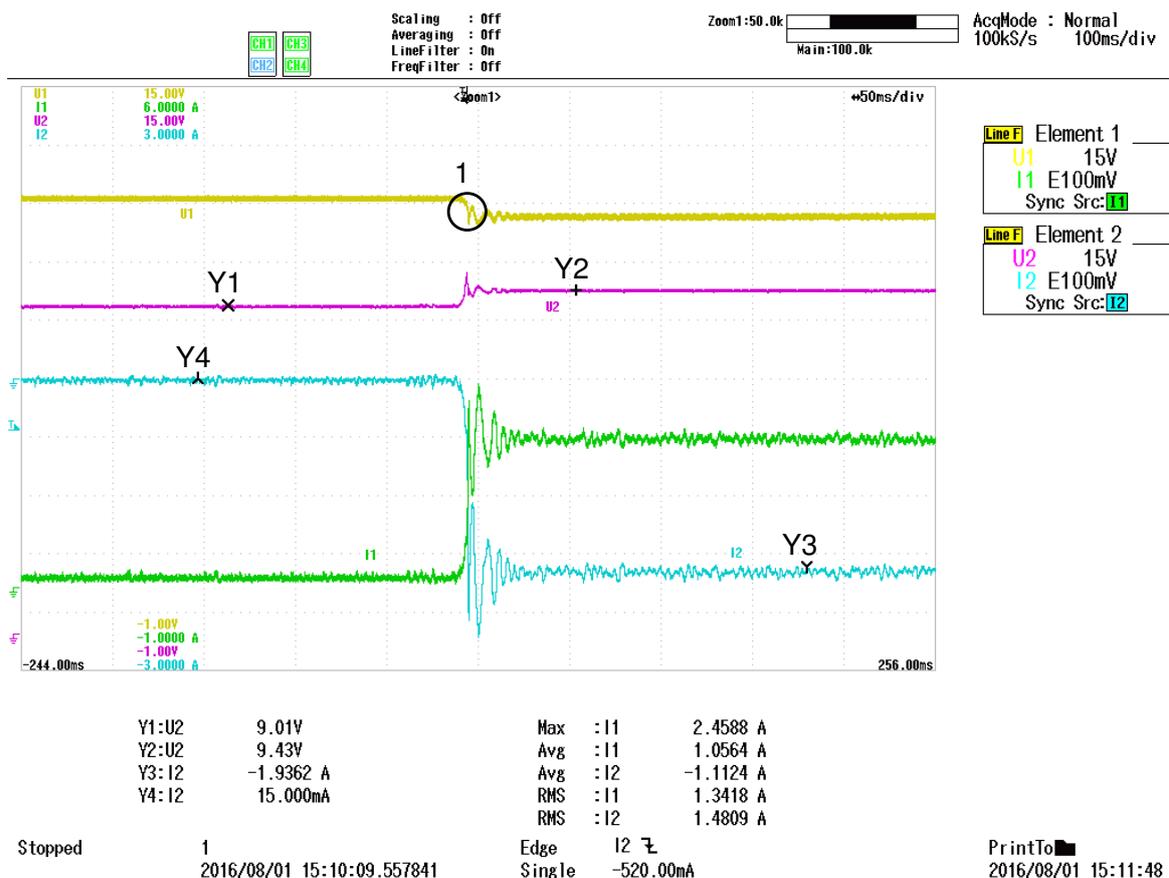


Abbildung 7.10: Untere Anschlaggrenze des Akkuladers

$U1 = U_{BN}$ (Gelb), $U2 = U_{BAT}$ (Lila), $I1 = I_{BN}$ (Grün), $I2 = I_{BAT}$ (Türkis)

Marker Y1-Y4 beziehen sich auf die in der Grafik angegebenen Spannungs- und Stromverläufe

Abbildung 7.11 zeigt das Abschalten des Akkuladestromes bei Erreichen der Ladeschlussspannung, indem die Akkuspannung erhöht wird. Das Verhalten ist zuverlässig. Allerdings ist ein Schwingvorgang zwischen den beiden Strömen zu beobachten. Durch Implementieren eines Akkuladealgorithmus kann an dieser Stelle ein optimiertes Verhalten erzielt werden. Hier kann der spannungsbegrenzte Stromregler für das Verhalten ausgemacht werden. Durch das Reduzieren des Akkuladestroms fällt über dem Innenwiderstand des Akkus keine Spannung mehr ab und die gemessene Spannung U_{BAT} fällt unterhalb der Ladeschlussspannung.

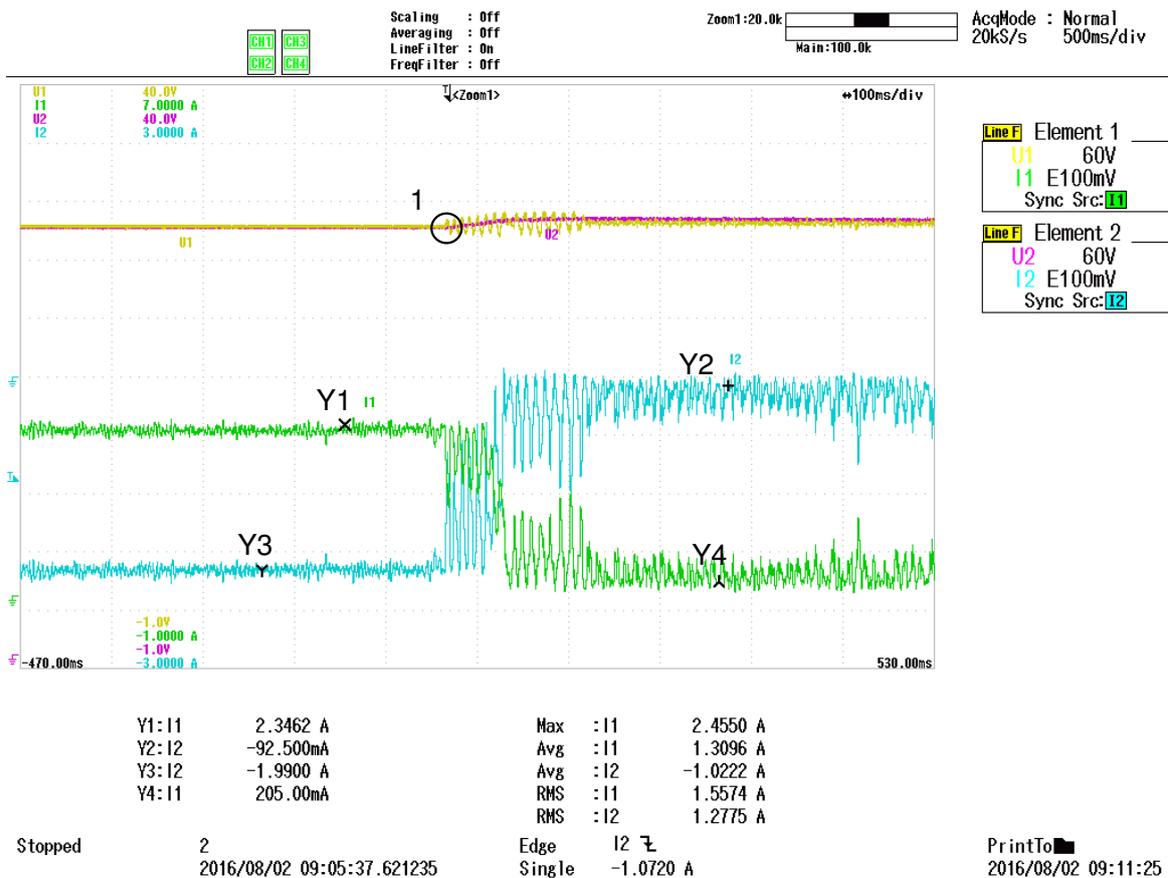


Abbildung 7.11: Abschaltgrenze bei Erreichen der Ladeschlussspannung

$U1 = U_{BN}$ (Gelb), $U2 = U_{BAT}$ (Lila), $I1 = I_{BN}$ (Grün), $I2 = I_{BAT}$ (Türkis)

Marker Y1-Y4 beziehen sich auf die in der Grafik angegebenen Spannungs- und Stromverläufe

Akku Laden und Lastverhalten:

Hier wird das Laden des Akkus während des Lastverhaltens untersucht. Der Akkuladestrom wird linear reduziert, wenn die Zwischenkreisspannung einbricht.

Es werden die signifikanten Punkte der Regelung aufgezeigt:

$$U_{ZK} = 29V \implies I_{Bat} = -2A \quad (7.1)$$

$$U_{ZK} = 28V \implies I_{Bat} = 0A \quad (7.2)$$

$$U_{ZK} = 27V \implies I_{Bat} = 2A \quad (7.3)$$

Der Strom wird linear gedrosselt.

Abbildung 7.12 und Abbildung 7.13 zeigt das Verhalten des Akkuladens nach den Annahmen.

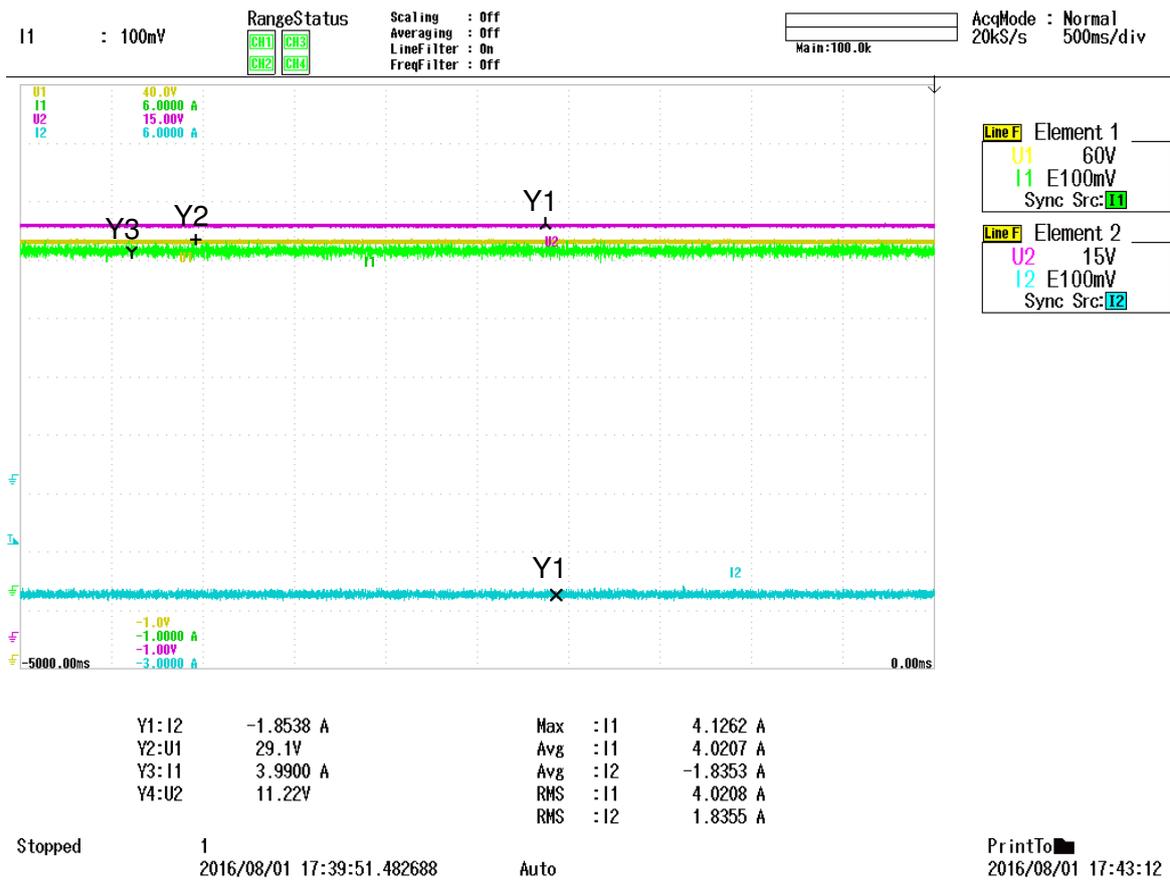


Abbildung 7.12: Regelgrenze 2 Ampere laden

$U1 = U_{ZK}$ (Gelb), $U2 = U_{Cap}$ (Lila), $I1 = I_{BN}$ (Grün), $I2 = I_{BAT}$ (Türkis)

Marker Y1-Y4 beziehen sich auf die in der Grafik angegebenen Spannungs- und Stromverläufe

Mischbetrieb

Der Mischbetrieb tritt ab einer Spannung $U_{ZK} < 28V$ in Kraft und ist in Abbildung 7.14 dargestellt. Auffällig ist, dass bei kleineren Ausgangsspannungen des Boost-Wandlers das Rauschen auf den Eingangsströmen größer wird.

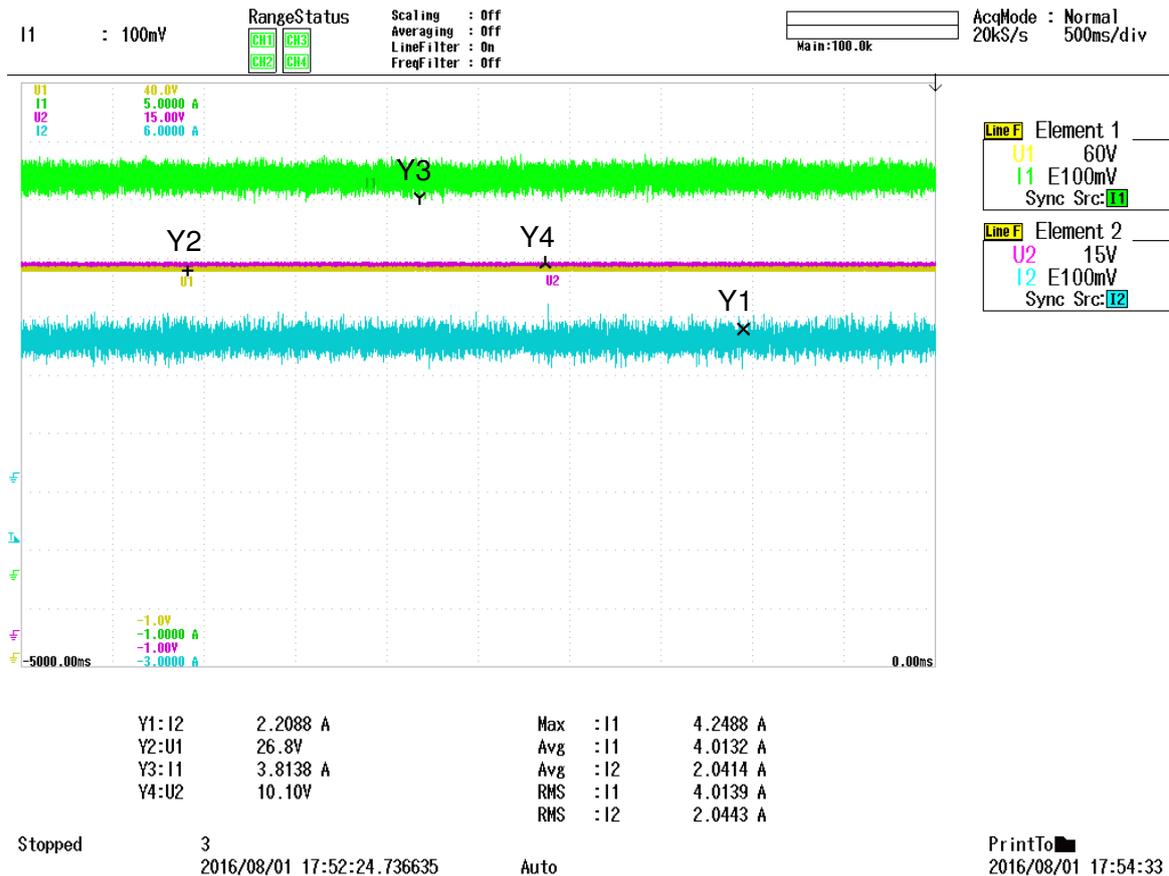


Abbildung 7.14: Spannungsgrenze 2 Ampere unterstützen

$U1 = U_{ZK}$ (Gelb), $U2 = U_{Cap}$ (Lila), $I1 = I_{BN}$ (Grün), $I2 = I_{BAT}$ (Türkis)

Marker Y1-Y4 beziehen sich auf die in der Grafik angegebenen Spannungs- und Stromverläufe

7.4 Speichern von zurückgewonnener Bremsenergie

An dieser Stelle wird die Leistungselektronik auf die Fähigkeit der Energierückgewinnung überprüft. Die geforderte Rückgewinnungsenergie der Motorelektronik (10 Joule) ruft eine Spannungserhöhung an den Elektrolytkondensatoren ($30V \Rightarrow 55V$) hervor.

Der Messaufbau erfolgt nach Abbildung 7.15

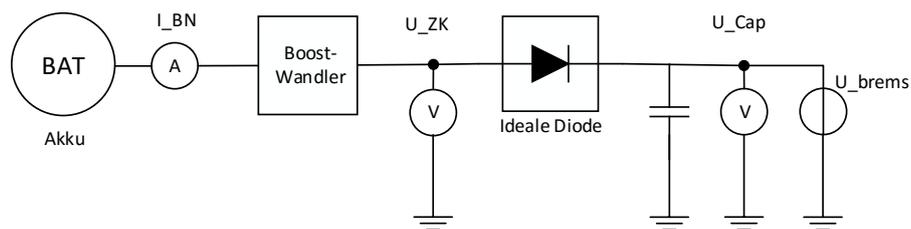


Abbildung 7.15: Blockschaltbild zu den Messungen - Energierückgewinnung

In Abbildung 7.16 wird ein Spannungssprung (55V) auf den Ausgang des Systems gegeben. Es ist zu sehen, dass dies auf die Regelung der Zwischenkreisspannung keinen Einfluss hat. Dadurch ist die Steifflankigkeit des negativen Lastsprunges unerheblich.

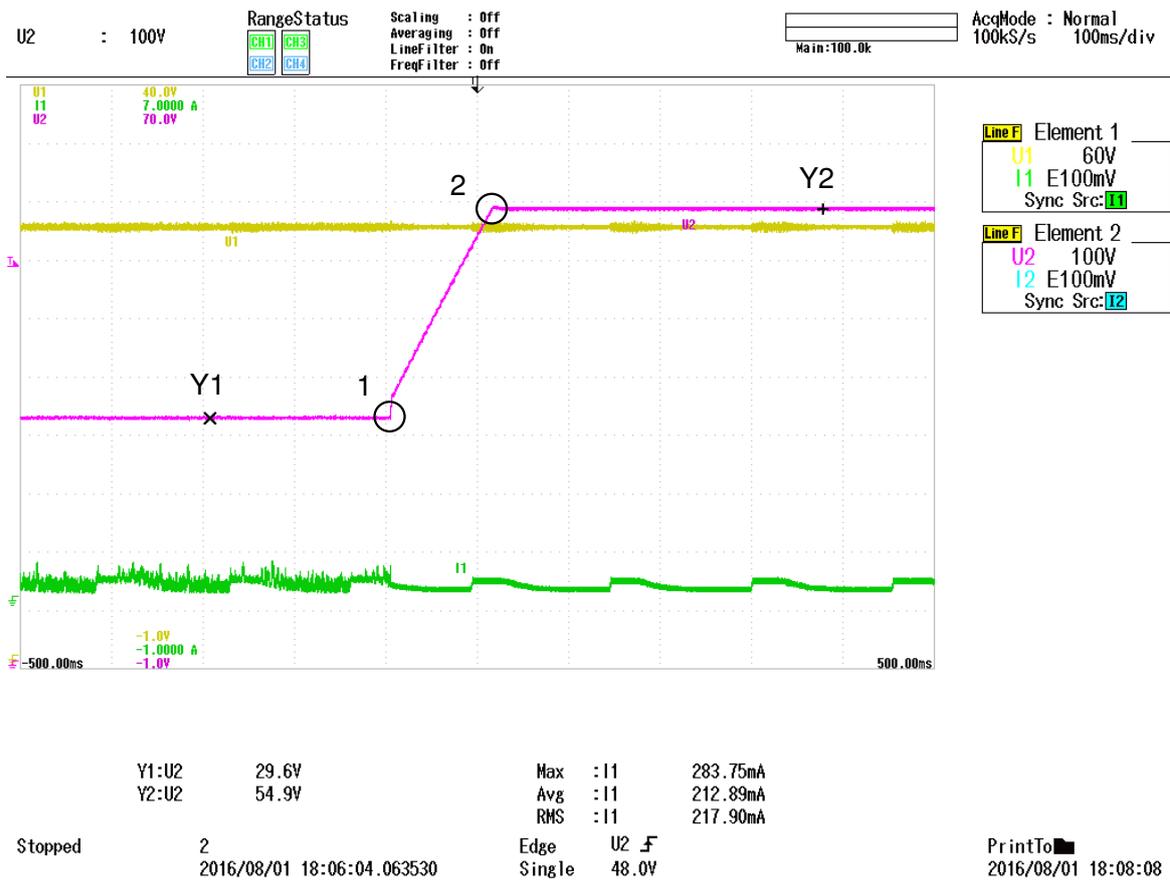


Abbildung 7.16: Spannungssprung bei 10 Joule Energierückgewinnung
 10 Joule entspricht nach Kapitel 6.1 einem Spannungssprung von 30V auf 55V
 $U1 = U_{ZK}$ (Gelb), $U2 = U_{Cap}$ (Lila), $I1 = I_{BN}$ (Grün)
 Marker Y1-Y4 beziehen sich auf die in der Grafik angegebenen Spannungs- und Stromverläufe

7.5 Einschaltverhalten

Aufgrund der großen Elektrolytkondensatoren ist der Einschaltmoment kritisch, was die Eingangsströme betrifft. Dazu wurde eine Elektronik entwickelt, welche die im Einschaltmoment kapazitive Last über 200Ω lädt. Zudem können Spannungsunterschiede in den Versorgungsquellen im Einschaltmoment Ausgleichsströme hervorrufen. Im Folgenden ist das Einschaltverhalten der Leistungselektronik dokumentiert.

Der Messaufbau erfolgt nach Abbildung 7.17

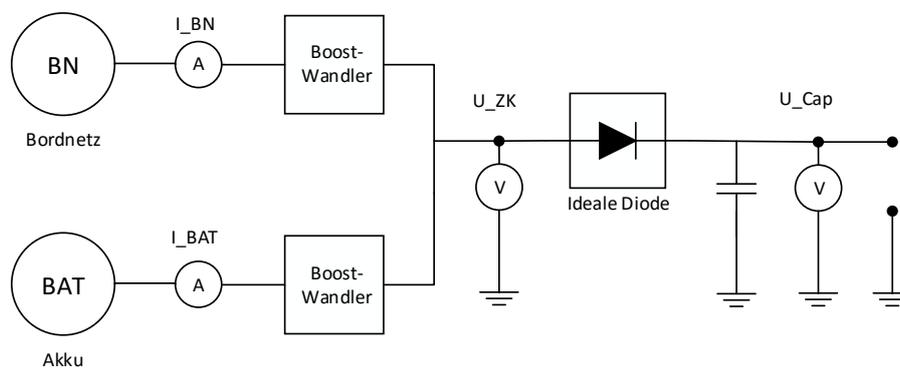


Abbildung 7.17: Blockschaltbild zu den Messungen - Einschaltverhalten

Abbildung 7.18 zeigt das Einschaltverhalten mit Akku und Bordnetz. In Punkt 1 wird das Bordnetz eingeschaltet und der Mikrocontroller wird mit Strom versorgt. Es kann kein Strom in die Quellen fließen, da die Boost-Wandler ausgeschaltet sind. Zum Zeitpunkt 2 werden die Boost-Wandler eingeschaltet. Aus dem Bordnetz fließt ein Stromimpuls ($I_2 = 1,2A$). Dieser fließt als Ausgleichstrom in den Akku. Nach dem Initialen Spannungsimpuls in 2 steigt die Zwischenkreisspannung linear an, bis diese die Regelspannung von U_4 erreicht. Die Spannung an den Elektrolytkondensatoren wird über 200Ω geladen. Dieser Widerstand wird in 3 über einen MOSFET abgeschaltet, wodurch die Elektrolytkondensatoren über die Stromimpulse in I_3 geladen werden und sich der Zwischenkreisspannung annähern.

Die Stromspitzen sind unkritisch und überschreiten nicht die maximale Stromaufnahme der Leistungselektronik.

Abbildung 7.19 zeigt das Verhalten der Leistungselektronik im Akkubetrieb. In Punkt 1 wird der Power-On-Button betätigt und der Akku versorgt den Mikrocontroller mit Spannung. Die Zwischenkreisspannung lädt sich auf und erfährt durch die Induktivität des Boost-Wandlers einen Spannungssprung.

In Punkt 2 wird der Boost-Wandler angeschaltet. Es entsteht ein großer Strompeak aus dem Akku in das Bordnetz.

Im weiteren Verlauf verhält sich das Einschaltverhalten wie in Abbildung 7.18. Zum Zeitpunkt 3 muss der Akku den Ladestrom für die Elektrolytkondensatoren direkt liefern.

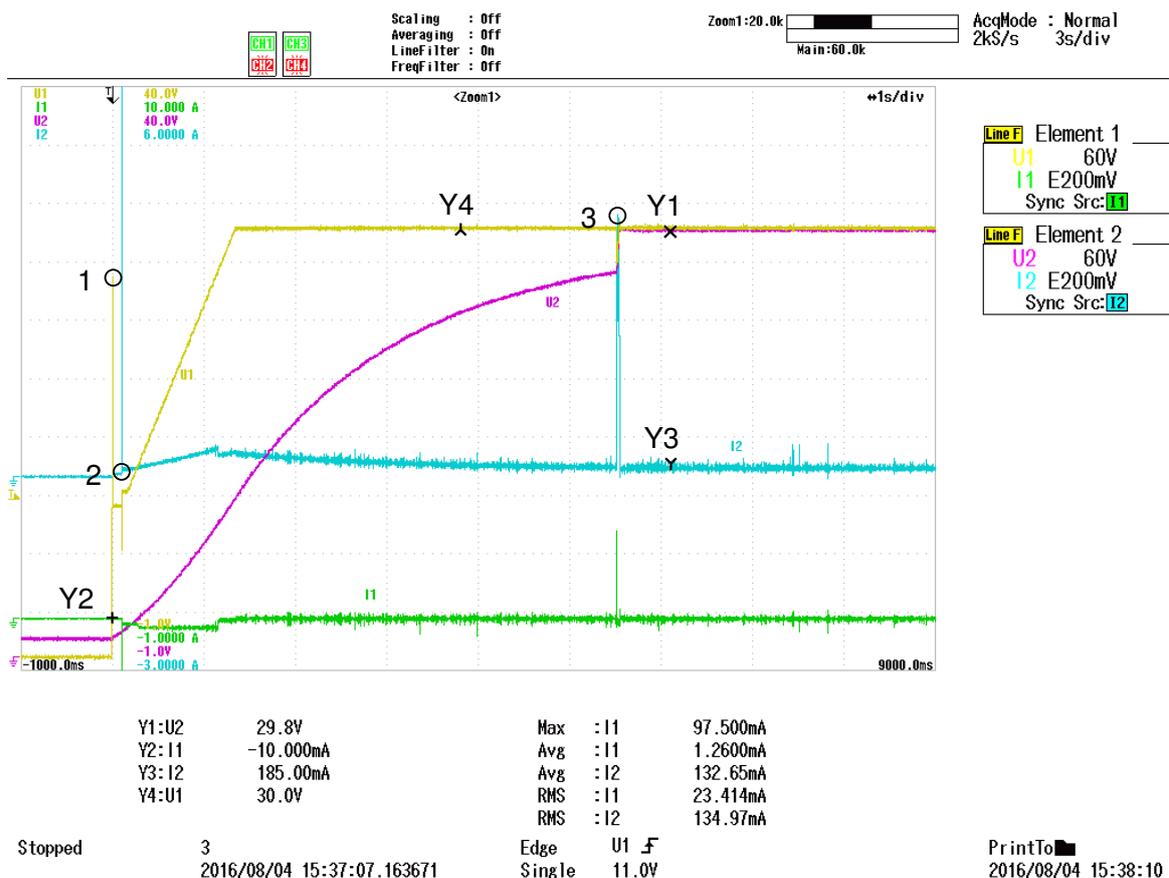


Abbildung 7.19: Einschaltverhalten im Akkubetrieb

$$U1 = U_{ZK}(\text{Gelb}), U2 = U_{Cap}(\text{Lila}), I1 = I_{BN}(\text{Grün}), I2 = I_{BAT}(\text{Türkis})$$

Marker Y1-Y4 beziehen sich auf die in der Grafik angegebenen Spannungs- und Stromverläufe

Marker Y1-Y4 beziehen sich auf die in der Grafik angegebenen Spannungs- und Stromverläufe

Abbildung 7.20 zeigt das Verhalten aus 7.19 zum Zeitpunkt 2. Die Ströme sind Ausgleichsströme, da Y4 ungefähr Y2 entspricht. Der Unterschied kann auch eine Messungenauigkeit in der Abtastung sein.

Zum Zeitpunkt 1 werden die Boost-Konverter angesteuert und die Eingangskapazitäten am Bordnetz werden durch den Stromimpuls geladen.

Es fließen bis zu 17,5 Ampere (Y2) über 2ms. Dieser Wert ist kritisch und bedarf einer genaueren Betrachtung.

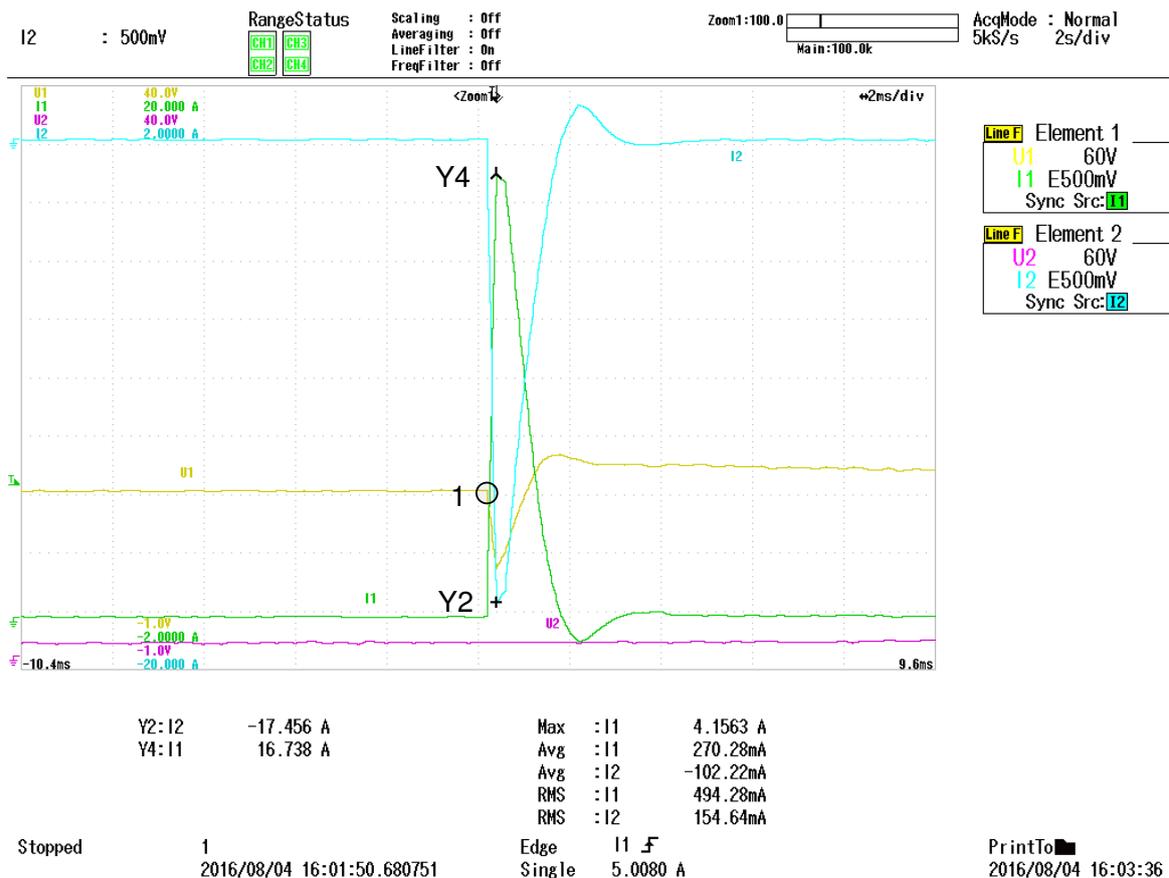


Abbildung 7.20: Ausgleichströme

$U1 = U_{ZK}$ (Gelb), $U2 = U_{Cap}$ (Lila), $I1 = I_{BN}$ (Grün), $I2 = I_{BAT}$ (Türkis)

Marker Y1-Y4 beziehen sich auf die in der Grafik angegebenen Spannungs- und Stromverläufe

7.6 Prozessorauslastung des Reglersystems

Abbildung 7.21 zeigt die Auslastung des Prozessors durch die Regelschleife. Die Periode entspricht dem Kehrwert der Reglerfrequenz mit $f = 8\text{kHz} = \frac{1}{T} \implies T = 128\mu\text{s}$.

Damit ergibt sich die Prozessorauslastung mit

$$\frac{67,2\mu\text{s}}{128\mu\text{s}} \cdot 100\% = 52,5\%. \quad (7.4)$$

Neben der Regelschleife belegen SPI-Interrupts für die Strommessung den Prozessor. Wird der Regler auf ein Fixed-Point-Datenformat geändert, könnte die Regelschleife mindestens doppelt so schnell ausgeführt werden.

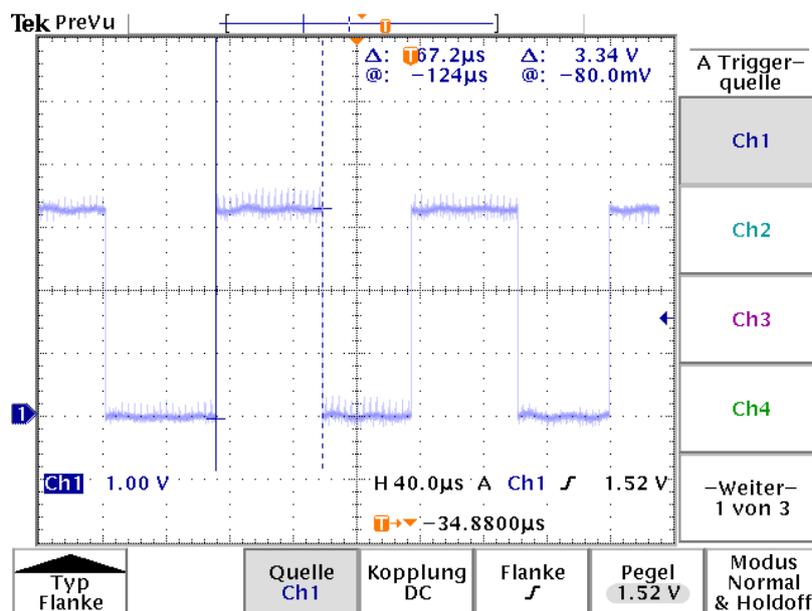


Abbildung 7.21: Prozessorauslastung der Regelschleife

7.7 Wirkungsgrad

In diesem Teil wird der Wirkungsgrad des Energiemanagementsystems überprüft. Die signifikanten Bauteile, welche den Wirkungsgrad maßgeblich beeinflussen, sind die Boost-Wandler.

Da beide Boost-Wandler identisch dimensioniert sind, wird exemplarisch ein Schaltwandler vermessen. Nach Abbildung 7.22 erfolgt der Messaufbau.

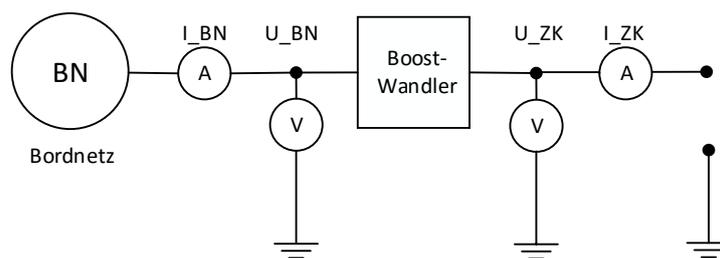


Abbildung 7.22: Blockschaltbild zur Messung des Wirkungsgrades eines Boost-Wandlers

Abbildung 7.23 zeigt den Wirkungsgrad über der Ausgangsleistung des Boost-Wandlers. Die Ausgangsspannung $U_{ZK} = 30V$ regelt das Energiemanagementsystem aus.

Der Ausgangsstrom I_{ZK} wurde mit einer elektronischen Last eingestellt. Der Ausgangsstrom wurde so eingestellt, dass sich der Eingangsstrom $I_{BN} = 200mA \dots 4A$ eingestellt hat.

Die Messung des Wirkungsgrades wurde mit den Eingangsspannungen $U_{BN} = 10V$ und $U_{BN} = 15V$ aufgenommen. Diese stellen die minimalen und maximalen Spannungsniveaus des Bordnetzes dar.

Die Schaltverluste des Boost-Wandlers werden nicht aufgenommen, diese fließen mit in die Grundlast der Kleinspannungsversorgung ein. Diese betragen

$$P_{Gate} = \frac{1}{2} \cdot U_g \cdot Q_g \cdot f \cdot 2 = 0,5V \cdot 12V \cdot 18nC \cdot 250kHz \cdot 2 = 54mW. \quad (7.5)$$

Es ist zu erwarten, dass der Boost-Wandler bei kleinen Ausgangsleistungen einen schlechteren Wirkungsgrad aufweist, da hier unvermeidliche Verluste in der Drosselspule und den Transistoren auftreten. Weiterhin wird für eine größere Eingangsspannung ein besserer Wirkungsgrad erwartet als für eine kleinere, da der Spannungshub auf die Zwischenkreisspannung dort geringer ist.

Wie zu erwarten, steigt der Wirkungsgrad mit einer größeren Ausgangsleistung. Bei einer Ausgangsleistung von etwa 6 Watt erreicht der Boost-Wandler einen Wirkungsgrad von

> 90%. Sein Maximum von 95,4% für eine Eingangsspannung von 15 Volt erreicht der Schaltwandler bei etwa 30 Watt. Das Maximum für eine Eingangsspannung von 10 Volt wird bei einer Ausgangsleistung von 20 Watt mit 94% erreicht. Damit erreicht die kleinste Eingangsspannung des Bordnetzes immer noch einen guten Wirkungsgrad.

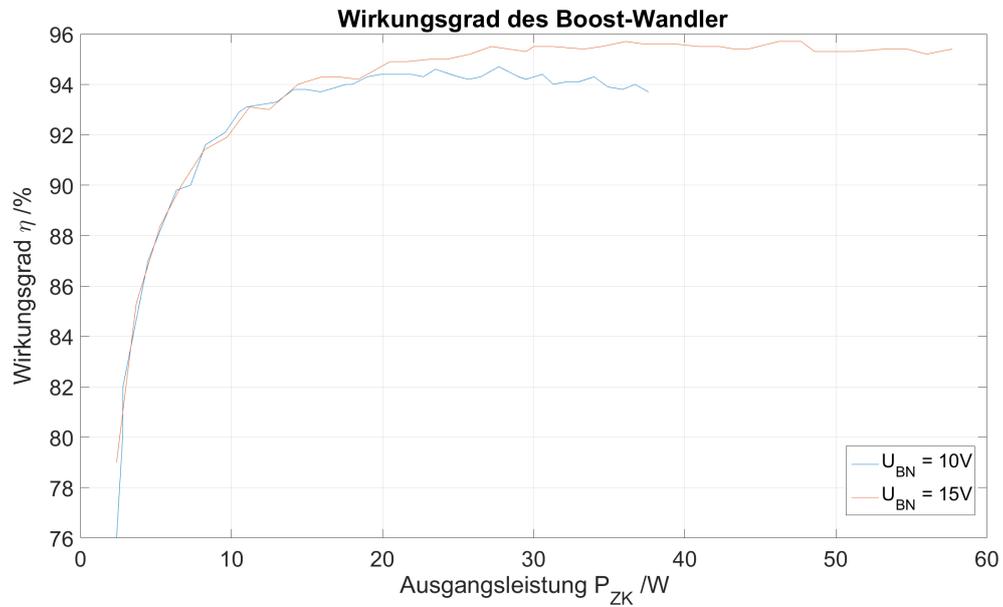


Abbildung 7.23: Wirkungsgrad eines Boost-Wandlers - $U_{ZK} = 30V$

Für das Zusammenspiel der beiden Boost-Wandler wird der Fall Akkuladen betrachtet. Der Wirkungsgrad kann aufgenommen werden, indem der Akku geladen wird und die Leistungen der Quellen betrachtet werden.

Der Messaufbau erfolgt nach Abbildung 7.24.

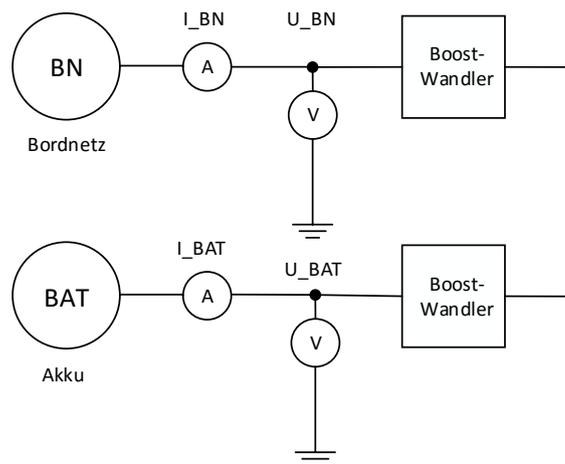


Abbildung 7.24: Blockschaltbild zur Messung des Wirkungsgrades für den Akkuladepfad

Die Eingangsspannung des Bordnetzes muss auf die Zwischenkreisspannung transformiert werden und von dem Aufwärtswandler des Akkus wieder auf eine kleinere Spannung geregelt werden, damit der Akkuladestrom eingestellt werden kann. Der Wirkungsgrad über zwei Schaltwandler fällt entsprechend geringer aus und beschreibt die signifikanten Verluste des Systems.

Es wurden Messungen für die Eingangsspannungen U_{BN} 10V, 12,5V und 15V durchgeführt.

Die Akkuspannung U_{BAT} wurde bei konstantem Akkuladestrom von 2 Ampere, von der Entladespannung 9.6V bis zu seiner Ladeschlussspannung eingestellt.

Abbildung 7.25 zeigt den aufgenommenen Wirkungsgrad über der Ausgangsleistung des Akkuladers (U_{BAT}).

Es ist zu erkennen, dass sich bei größerer Eingangsspannung ein verbesserter Wirkungsgrad einstellt. Dies lässt sich mit dem geringeren Spannungshub von U_{BN} auf $U_{ZK} = 30V$ erklären.

Weiterhin steigt der Wirkungsgrad bei zunehmender Akkuspannung, denn bei größerer Akkuspannung ist der Spannungshub, den der Schaltwandler $U_{ZK} \Rightarrow U_{BAT}$ transformieren muss, geringer.

Festzuhalten ist, dass der Wirkungsgrad über dem gesamten Spannungsbereich $> 90\%$ beträgt.

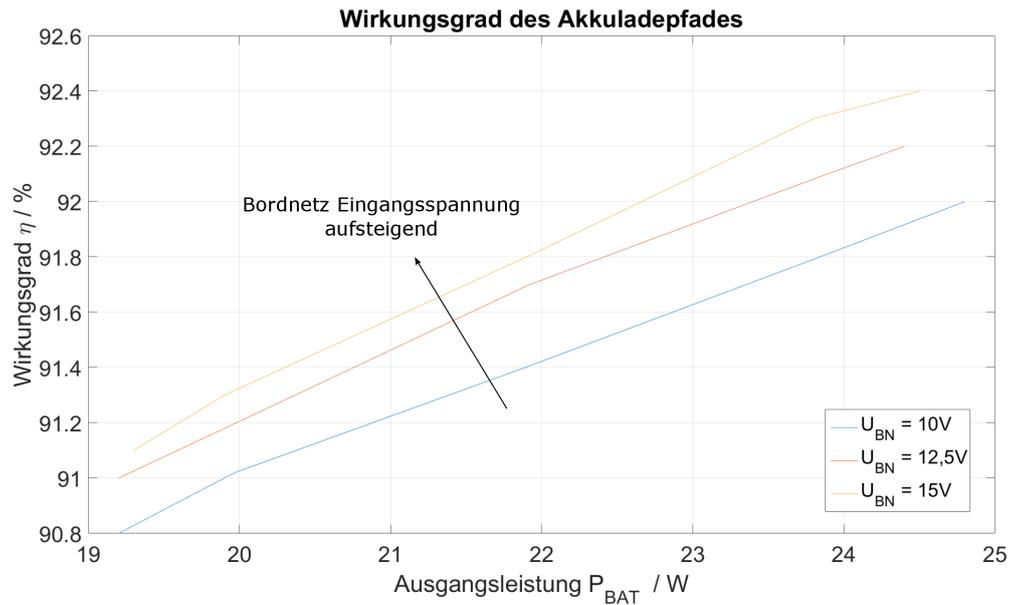


Abbildung 7.25: Wirkungsgrad des Akkuladepfades über 2 Boost-Wandler
Akkuspannung: $U_{BAT} = 9.7V \dots 12.5V$, $U_{ZK} = 30V$

7.8 Leckstrom

Um zu gewährleisten, dass sich der Akku im ausgeschalteten Zustand des Gerätes nicht entlädt, wird der Strom aus dem Akku überprüft.

Der Akkustrom darf nur einen minimalen Leckstrom in die Leistungselektronik aufweisen, um lange Standby-Zeiten des Beatmungsgerätes zu erreichen.

Wie in Abbildung 7.26 schematisch gezeigt, wird über einen $1\text{ k}\Omega$ Widerstand der Strom gemessen, damit eine gute Auflösung des geringen Leckstromes gegeben ist.

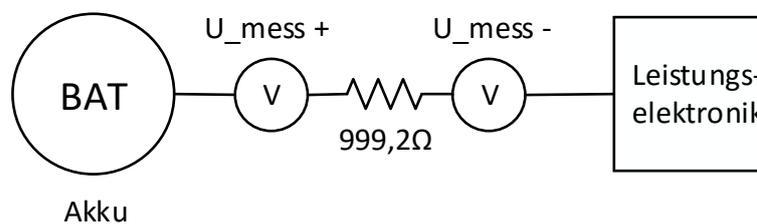


Abbildung 7.26: Blockschaltbild zur Messung des Leckstromes aus dem Akku während des ausgeschalteten Zustandes

Die Spannungsmessung über den beiden Messpunkten ergab einen Spannungsabfall über dem Widerstand von

$$U_{mess} = 10,3\text{ mV}. \quad (7.6)$$

Damit berechnet sich der Leckstrom aus dem Akku wie folgt:

$$I_{Leck} = \frac{U_{mess}}{R_{mess}} = \frac{10,3\text{ mV}}{999,2\Omega} = 10,3\mu\text{ A} \quad (7.7)$$

Der Leckstrom aus dem Akku ist minimal und im Vergleich zur Selbstentladung des Akkus zu vernachlässigen.

7.9 Diskussion

Die Versuchshardware konnte vollständig in Betrieb genommen werden und konnte wie geplant betrieben werden. Der Leistungspfad eines Boost-Wandlers zeigte einen Wirkungsgrad $> 93\%$ bei einem Eingangsstrom $I_{ein} > 20mA$ und einen maximalen Wirkungsgrad von 95% in den geforderten Spannungsbereichen. Der Leistungspfad über zwei Boost-Wandler vom Bordnetz in den Akku zeigte dabei einen Wirkungsgrad von $> 91\%$. Die Werte belegen eine geeignete Dimensionierung der Schaltwandler und damit ein System mit einem hohen Wirkungsgrad. Die Kleinspannungsversorgung nimmt eine dauerhafte Wirkleistung von $1,5$ Watt auf. Dieser Wert ist mit dem eines Notfallgerätes nicht zu vergleichen, da dort eine größere Zahl an Kleinspannungsverbrauchern verbaut ist. Eine größere Last der Kleinspannungsverbraucher erhöht den Grundenergiebedarf des Systems.

Eine Energierückgewinnung war uneingeschränkt möglich und hatte keinen Einfluss auf die Regelung.

Die Regelschleife, welche auf einem STM32f334 mit $8kHz$ ausgeführt wurde, erwies sich als ausreichend schnell. Diese belegte eine durchschnittliche Prozessorauslastung von 60% . Mittels Optimierung der Datentypen von „single“ auf einen „Fixed-Point“-Datentyp bietet die Prozessorauslastung sogar noch Optimierungspotenzial.

Die Funktionalität des Energiemanagements konnte mithilfe der entwickelten Software überprüft werden. Die Eingangsspannungsbereiche werden richtig erkannt und entsprechend verarbeitet. Die Eingangsströme können geregelt und begrenzt werden und überschreiten nicht deren zulässige Grenzen.

Es konnte gezeigt werden, dass der Akkuladestrom kontrolliert geregelt werden kann. Die Spannungsbegrenzung des Stromreglers bei Ladeschlussspannung führt zu keinem sauberen Abschaltverhalten und stellt keinen Akkuladealgorithmus dar. Es konnte aber bewiesen werden, dass der Akkustrom ohne Einschränkungen beherrschbar ist.

Die Zwischenkreisspannung konnte in allen Systemzuständen zuverlässig ausgeregelt werden. Bei kritischem Systemverhalten wie dem Einschalten der Bordnetzspannung wurden keine kritischen Überschwinger festgestellt.

Das Lastverhalten konnte über dem vollständigen Eingangsspannungsbereich zuverlässig ausgeregelt werden. Das Akkuladen konnte parallel zum Lastbetrieb betrieben werden, wobei der Akkuladestrom dem Laststrom entsprechend reduziert werden konnte. Zudem konnte ein Mischbetrieb eingeführt werden, welcher einen größeren Leistungsverbrauch ermöglichen würde, bzw. ein schwaches Bordnetz unterstützen kann.

Die Anforderungen gelten, für die in dieser Arbeit durchgeführte erste Entwicklungsrunde. Für eine zweite Entwicklungsrunde, muss auf die noch nicht berücksichtigten, sowie im

Nachhinein festgestellten Probleme eingegangen werden. Die entwickelte Schaltung für das Laden der Pufferkondensatoren funktionierte einwandfrei. Der Einschaltmoment ruft Ausgleichsströme zwischen den Quellenanschlüssen hervor, was einer genaueren Betrachtung bedarf.

In Tabelle 7.2 werden die Anforderungen aus Tabelle 3.1 mit den Messergebnissen aus Kapitel 7 gegenübergestellt.

Tabelle 7.2: Auswertung der Anforderungen

Anforderung	Wert	Typ	Ergebnis
Primäre Versorgungsquelle: Bordnetz	10V-15V / max. 4A	Anforderungen	✓
Sekundäre Energiequelle: Akku	9V-12,6V / max. 4A		✓
Primäre Last: Konstante Energieversorgung der Leistungselektronik (Motorgebläse)	min. 25V / 35W		✓
Steuerung/Regelung der Leistungselektronik durch Mikrocontroller			✓
Sekundäre Last: Ein Laden des Akkus durch das Bordnetz ermöglichen	max. 2A		✓
Speichern von zurückgewonnener Bremsenergie des Gebläses	10 Joule		✓
Unterstützen durch Akku bei schwachem Bordnetz (Mischen der Quellen)		Wunsch	✓
Energieversorgung der Kleinspannungselektronik		Anforderungen	✓
Kein Entladen des Akkus bei System-Aus (kein Nennenswerter Leckstrom)			✓
Multi- oder Einratensystem	min. 4kHz Abtastrate		✓
Verluste im Leistungspfad minimieren			✓
Wirkungsgrad	>90%		✓
Möglichst wenig Bauteile, Komplexität		Wunsch	✓

8 Bewertung

Mit diesem Kapitel wird die Bachelorarbeit abgeschlossen. Es wird ein Fazit aus dem erarbeiteten Energiemanagementsystem gezogen und mit einem Ausblick auf Erweiterungen und weiterführende Untersuchungen abgeschlossen.

8.1 Fazit

Das entwickelte Konzept eines softwaregesteuerten Energiemanagementsystems konnte mit einer Versuchshardware auf Machbarkeit überprüft werden. Die gesetzten Ziele wurden vollständig erreicht und das Konzept erscheint vielversprechend für medizinische Notfallgeräte mit erhöhtem elektrischen Energiebedarf. Es kann mit einem reduzierten Bauteilaufwand ein Energiemanagementsystem mit einem hohen Wirkungsgrad erzielt werden.

Das Softwarekonzept wurde umgesetzt und bestätigt, dass die Funktionalität eines dezentralen Energiemanagementsystems auch in der Software abgebildet werden kann. Die Regelgeschwindigkeit stellt mit 8kHz bei gängigen Mikrocontrollern keine großen Anforderungen an die Hardware. Zumal die Regelung in einem Fließkommatdatenformat umgesetzt wurde und damit Optimierungspotenzial bietet.

Es wurde bewiesen, dass der Strom aus den Quellen begrenzt und der Strom des Akkus geregelt werden kann. Außerdem wurde gezeigt, dass eine zuverlässige Versorgung der Last gegeben ist.

Die Priorisierung der Quellen sowie der Verbraucher konnte definiert werden. Es konnte zudem ein Mischbetrieb eingeführt werden, welcher die Flexibilität einer Softwaresteuerung nutzt.

Die Flexibilität ist durch Softwareanpassungen gegeben. Der Ausgang zu dem Leistungsverbraucher wurde für eine Gebläseelektronik ausgelegt und von dem Rest des Systems abgegrenzt, sodass für andere Aktoren ein minimaler Hardwareaufwand anfallen würde. Die Übertragung auf andere Geräte beschränkt sich auf eine Skalierung der Hardware.

Die Versuchshardware ist weit davon entfernt, in Serienstatus zu gehen. Damit dies geschehen kann, wird im Folgenden ein Ausblick auf weiterführende Untersuchungen des Hardware- und Softwarekonzeptes gegeben.

8.2 Ausblick

In diesem Abschnitt werden Erweiterungen vorgestellt, die für das entwickelte Konzept sinnvoll erscheinen.

Als wichtiger nächster Entwicklungsschritt wird das Implementieren eines vollständigen Akkuladealgorithmus angesehen. Der Strom kann geregelt werden. Für eine Spannungsregelung beim Erreichen der Ladeschlussspannung steht die gemessene Akkuspannung zur Verfügung. Für weitere Überwachungen des Akkuzustandes wurde die SMBus-Schnittstelle auf der Versuchshardware vorgesehen. Diese kann implementiert werden, damit alle wichtigen Informationen des Akkus ausgelesen werden können.

Die entwickelte Regelung wurde für ein Matlab-Simulink-Modell optimiert und auf einen Mikrocontroller übertragen. Der Rahmen dieser Arbeit ließ keine theoretische Betrachtung der Reglerstabilität zu. Als weiterführende Untersuchung kann die mathematische Betrachtung des Reglersystems zudem neue Erkenntnisse bringen.

Der Regelalgorithmus wird im Datenformat „single“ ausgeführt. Eine Umstellung auf ein „Fixed-Point“-Datenformat würde die Prozessorauslastung verringern, welches die Möglichkeit bieten würde, die Ausführung der Regelschleife zu beschleunigen.

Da die Stromsensorwerte eine zu große Bandbreite aufwiesen, wurde eine Mittelung der Sensorwerte vorgenommen. Für eine bessere Behandlung der abgetasteten Werte sollten geeignete digitale Filter entworfen werden. Diese bieten zudem eine größere Steilflankigkeit an der Grenzfrequenz.

Zur Sicherung der Schaltung wurde für die Ansteuerung der Boost-Wandler eine obere Grenze des Tastverhältnisses festgelegt. Das hat zur Folge, dass bei kleinen Eingangsspannungen, der Boost-Wandler am Ausgang nicht mehr die Zwischenkreisspannung erzeugen kann. Daher kann es zu ungewollten Stromflüssen an den Quellen kommen, da der Regler nicht mehr auf 0 Ampere regeln kann. Es sollte darüber nachgedacht werden, bei kleinen Eingangsspannungen die Schaltwandler via Software auszuschalten, um einem Stromfluss aus dem System entgegenzuwirken.

Es wäre denkbar, Teile des entwickelten Hardwarekonzeptes zu standardisieren und zur Verringerung des Entwicklungsaufwands auf andere Geräteklassen der medizinischen Notfallgeräte zu übertragen.

Literaturverzeichnis

- [1] Andler H. Oczenski, W. and A. Webra, editors. *Atmen - Atemhilfen*. Georg Thieme Verlag, 2006
- [2] Johannes Kreuzer. Fachliches Gespräch über das Lastverhalten und die energierückgewinnung der Gebläseelektronik. *bei Weinmann Emergency Medical Technology GmbH AND Co. KG*, 2016
- [3] Ulrich Schlienz, editor. *Schaltnetzteile und ihre Peripherie*. Springer Vieweg, 2012
- [4] Gerfer A. Rall B. Brander, D. and H. Zenker, editors. *TRILOGIE der induktiven Bauteile*. Swiridof Verlag, 2008
- [5] Jan Lunze, editor. *Regelungstechnik 1: Systemtheoretisch Grundlagen, Analyse und Entwurf einschleifiger Regelungen*. Springer Vieweg, 2012
- [6] Olivier Garcia. Dc/dc-wandler für die Leistungsverteilung in einem Elektrofahrzeug mit Brennstoffzellen und Superkondensatoren. Dissertation, Eidgenössischen Technischen Hochschule Zürich, 2002
- [7] Christophe Millet Christophe Batard, Frédéric Poitiers and Nicolas Ginot. Simulation of power converters using matlab-simulink, 2012
- [8] Deepu Jose Reshma Mary Thomas. Droop control method for parallel dc converters used in standalone pv-wind power generating systems. *International Journal of Research in Engineering and Technology*, (4.10):297–302, October 2015
- [9] Alexander Winter. Regelung von dc netzen. *Internationale Energiewirtschaftstagung an der TU Wien*, 2015

A Simulation

In diesem Teil des Anhanges ist das Matlab-Simulink-Modell in Anhang A.1 eingefügt. Für eine genauere Ansicht sei auf die beiliegende DVD verwiesen, welcher die Matlab-Modelle beiliegen.

In Anhang A.2 ist das implementierte Reglermodell abgebildet, aus welchem der C-Code für die SysTick-Interrupt-Routine erzeugt wurde.

Anhang A.3 legt beispielhaft eine vollständige Simulation des Matlab-Simulink-Modells bei, in welcher die Zustände des Systems abgebildet wurden. Für weitere Bilder zu den Simulationsergebnissen sei auf die digitale Version der Bachelorarbeit verwiesen.

A.1 Modell

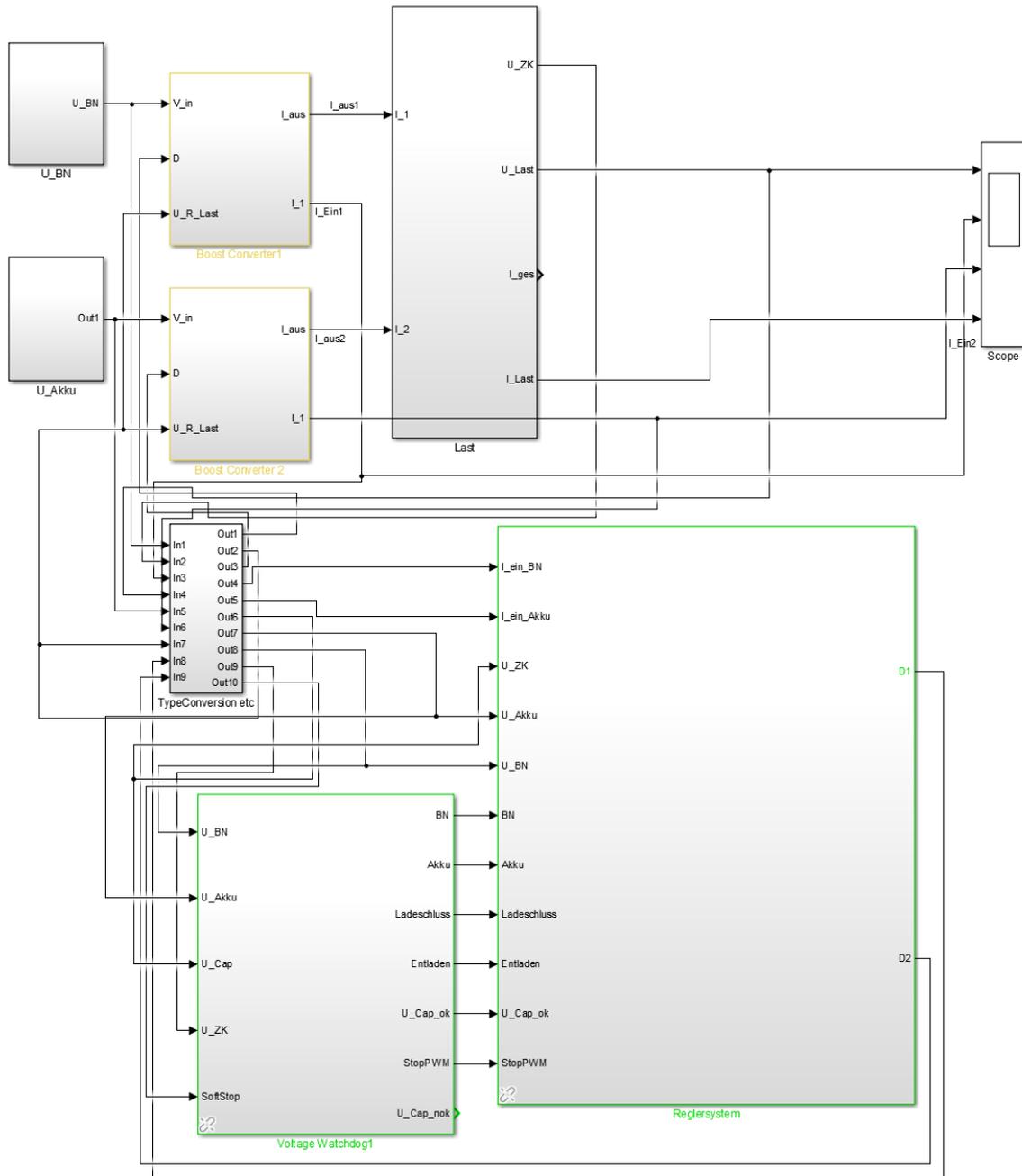


Abbildung A.1: Matlab-Simulink-Modell der Regelstrecke und des Reglers

A.2 Reglermodell

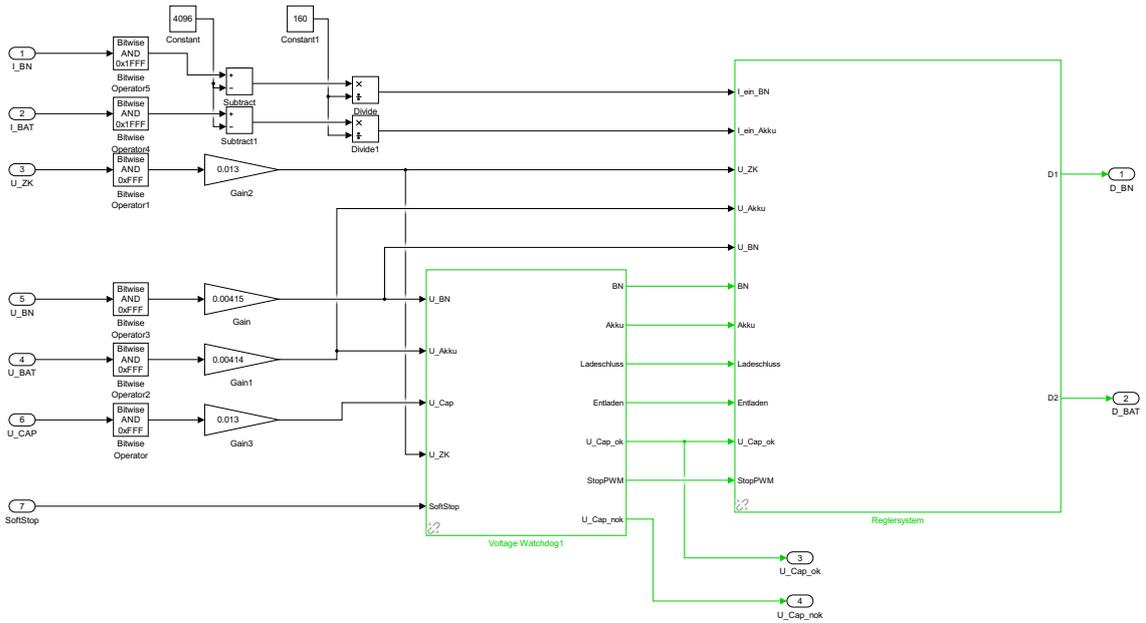


Abbildung A.2: Matlab-Simulink-Modell des implementierten Reglers

A.3 Simulationsergebnis

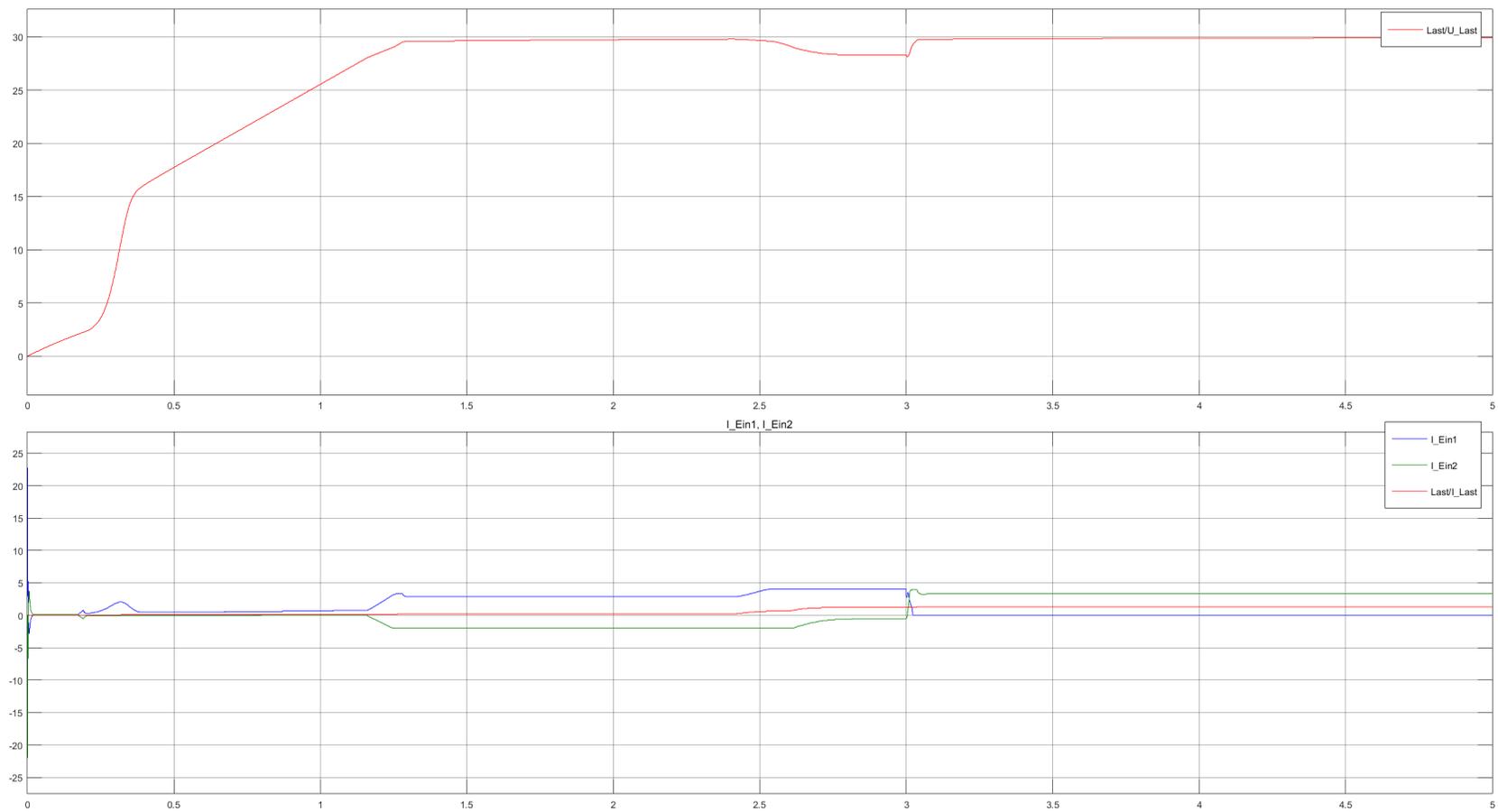


Abbildung A.3: Simulation der Regelstrecke und des Reglers

B Hardware

In diesem Teil des Anhangs ist der Schaltplan und das Platinenlayout beigelegt. Für eine detailliertere Betrachtung sei entweder auf die digitale Version der Bachelorarbeit verwiesen, oder auf die beiliegende DVD, in welcher die Altium Designer / Circuit Studio Dateien beiliegen.

Das Platinenlayout ist ein 4-lagiges Layout, welches

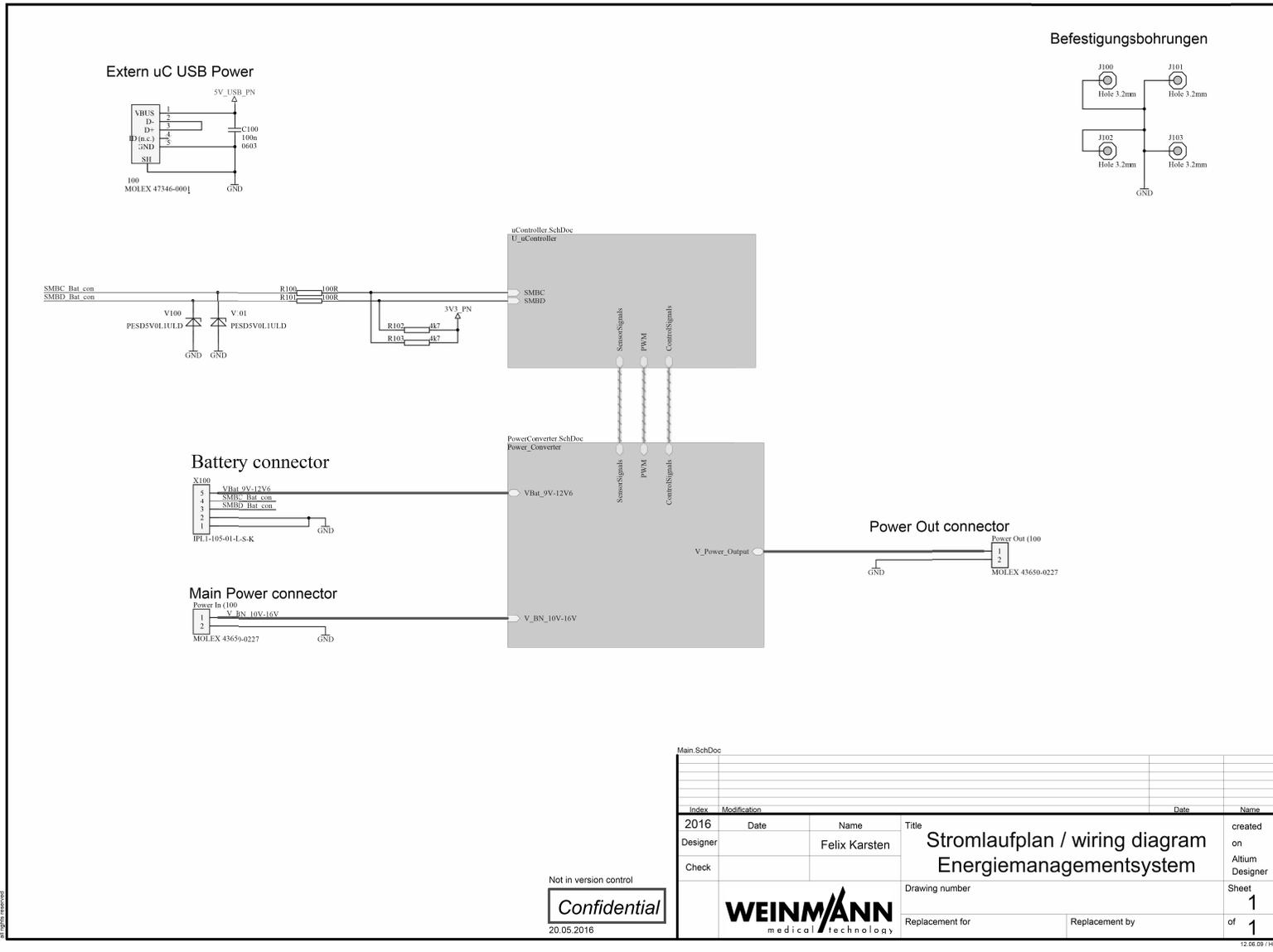
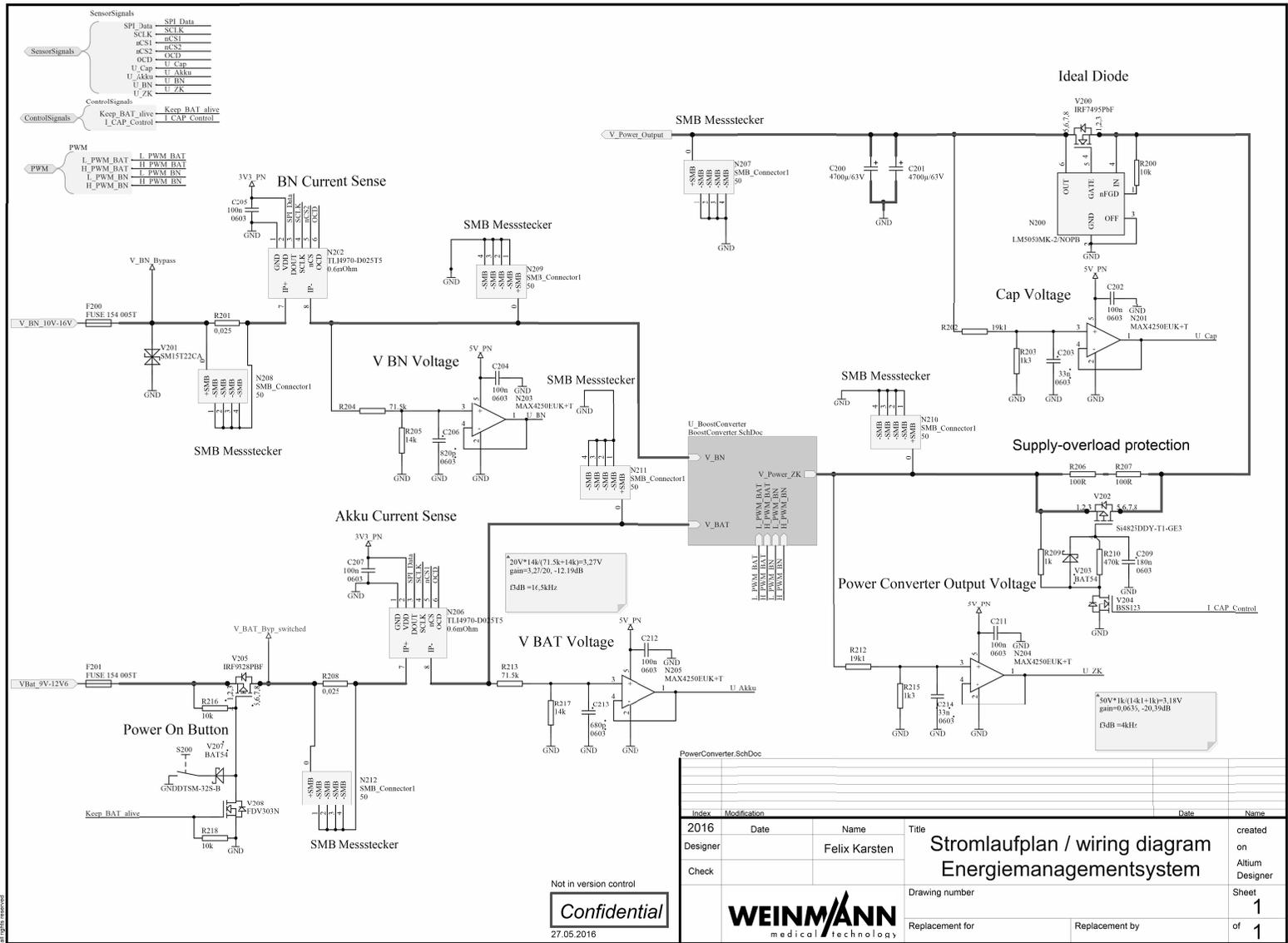


Abbildung B.1: Schaltplan - Sheet Entry



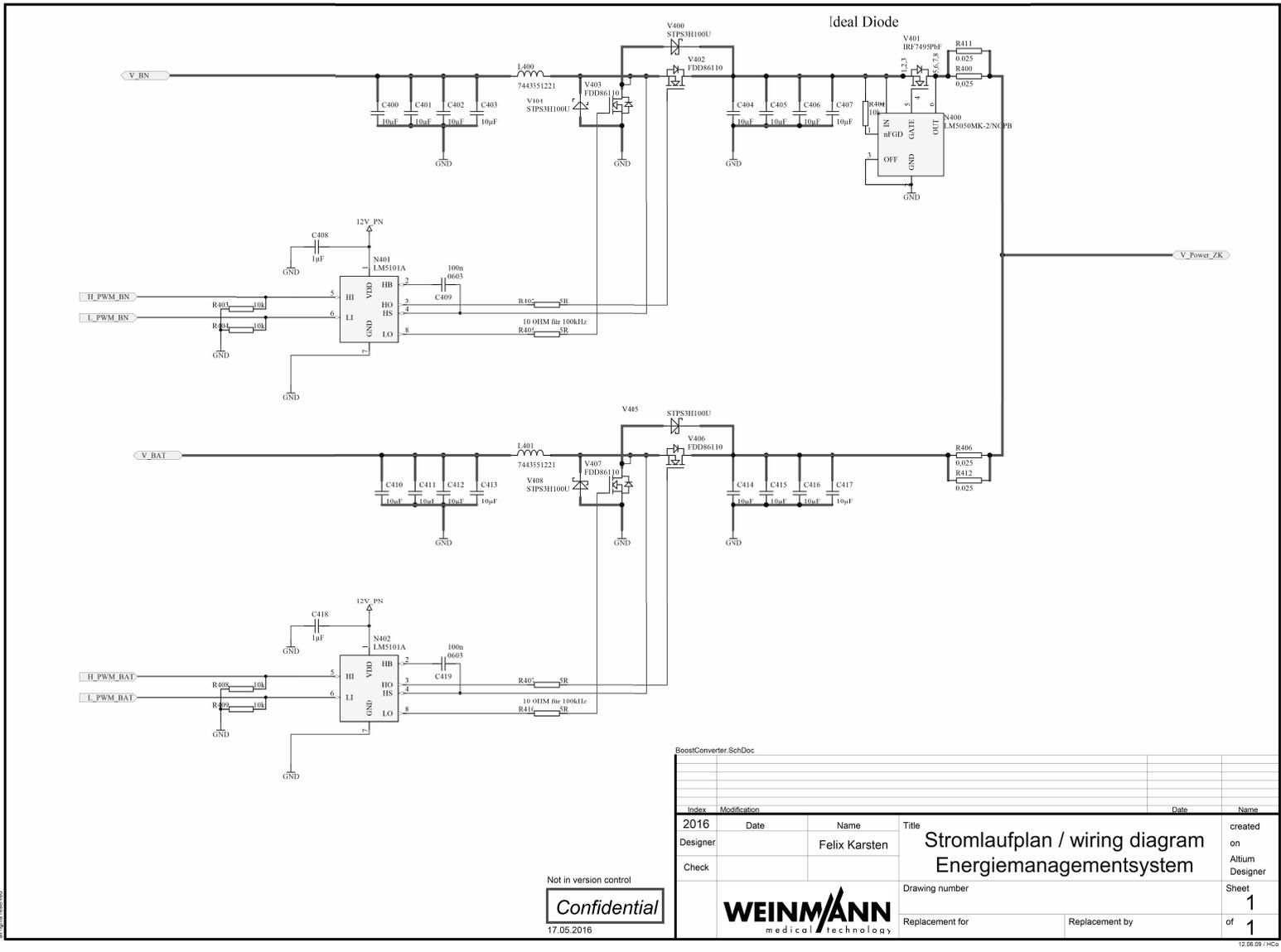
all rights reserved

Not in version control
Confidential
 27.05.2016

Index	Modification	Date	Name
2016			
Designer			Felix Karsten
Check			

Title		created on
Stromlaufplan / wiring diagram		Altium Designer
Energienagementsystem		Sheet 1
Drawing number		of 1
Replacement for		Replacement by

Abbildung B.2: Schaltplan Leistungselektronik



Index	Modification	Date	Name
2016			

Designer	2016	Name	Felix Karsten	Title	Stromlaufplan / wiring diagram	created on	Altium Designer
Check				Drawing number		Sheet	1
				Replacement for		of	1

Not in version control
Confidential
 17.05.2016



Abbildung B.3: Schaltplan Boost-Wandler

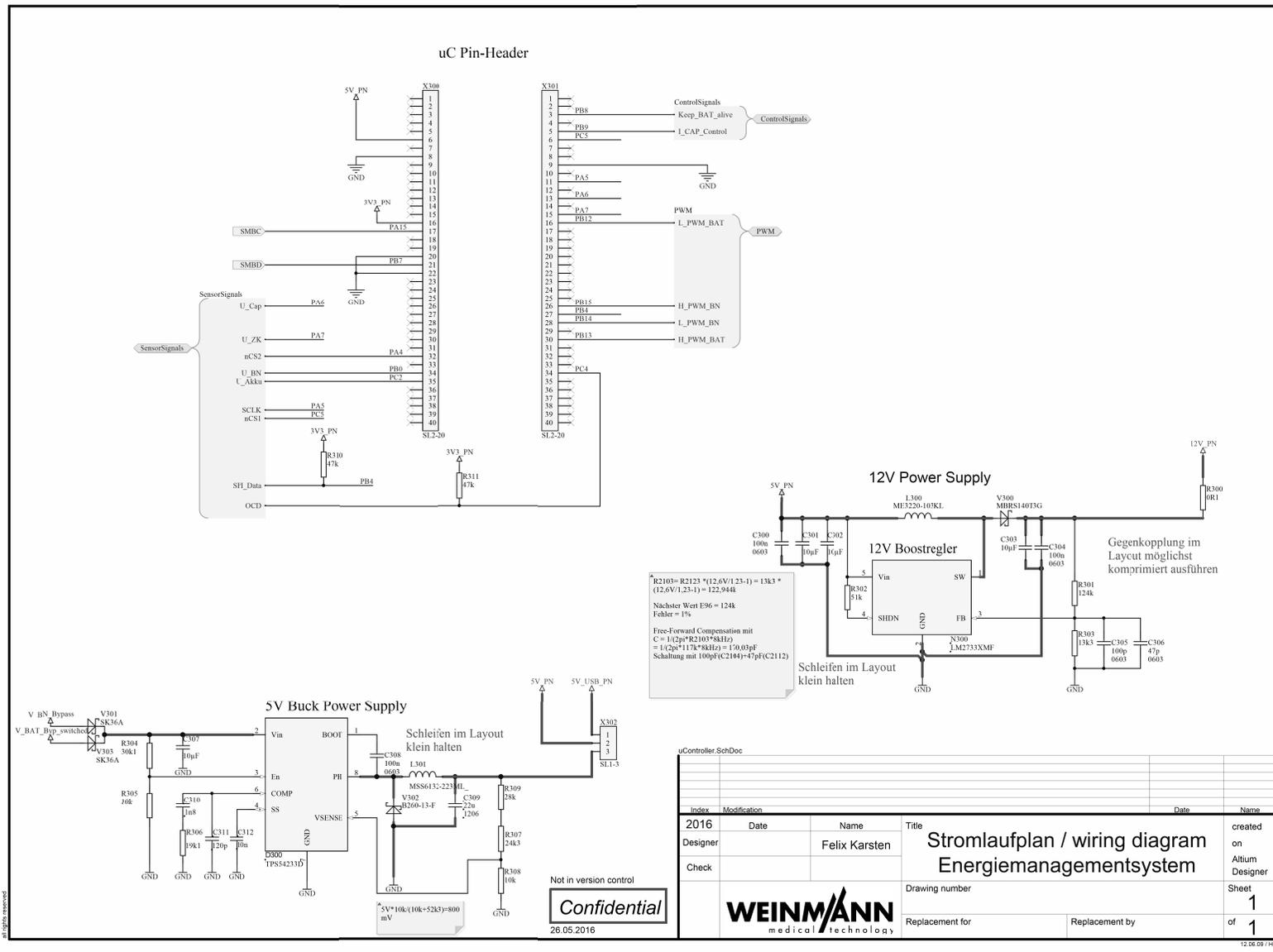


Abbildung B.4: Schaltplan Kleinspannungsversorgung/Mikrocontroller

C Quellcode

In diesem Teil des Anhangs ist der Quellcode der main-Funktion beigelegt, welche die Funktionen des Mikrocontroller abbildet. Der DVD liegt das gesamte TrueStudio Projekt bei.

C.1 main

```
/**
 * File Name      : main.c
 *-----*/
#include "stm32f3xx_hal.h"

#include <stddef.h>
#include <stdio.h>
        /* Model's header file */
#include "MatlabRegler_fptool.h"
#include "rtwtypes.h"

/* Private variables -----*/
ADC_HandleTypeDef hadc2;
DMA_HandleTypeDef hdma_adc2;

HRTIM_HandleTypeDef hrtim1;
DMA_HandleTypeDef hdma_hrtim1_c;
DMA_HandleTypeDef hdma_hrtim1_d;

SMBUS_HandleTypeDef hsmbus1;

SPI_HandleTypeDef hspi1;
DMA_HandleTypeDef hdma_spil_rx;

//hilfsvariablen
```

```
volatile int i;
volatile int a = 0;
volatile int b = 0;
//volatile uint32_t ADC1values[50];
volatile uint32_t ADC2values[4]; //Variablen für die 4 ADC Kanäle

// Variablen für die Spannungswerte
// (Übergabeparamter für die Matlab Funktion)
volatile float U_BN=0;
volatile float U_BAT=0;
volatile float U_ZK=0;
volatile float U_CAP=0;

//PWM Tastverhältnisvariablen (0...1)
volatile double D_BAT = 0;
volatile double D_BN = 0;

//vorgesehen für den voltage watchdog
// (nicht verwendet in letzter Fassung)
volatile int outofrange = 0;
volatile int outofrange2 = 0;
volatile int outofrange3 = 0;

//variable für die SPI Daten
volatile uint16_t pSPIData = 0;

//Variable für den Bordnetz Strom
volatile float I_BN = 0.0;

//Daten des SPI Sensors
volatile uint8_t pData[4]={0};

//empfangene Stromwerte
volatile uint16_t I_BNdata = 0;
volatile uint16_t I_BATdata = 0;

//variable für den Akkustrom
volatile float I_BAT = 0.0;
```

```
//chip select variablen
volatile      int ncs1 = 1;
volatile      int ncs2 = 0;

//eingangsvariable stop für die Matlab Funktion
volatile uint16_t SoftStop = 1;

//Zustände des Matlab Modells Voltage Watchdog
volatile boolean_T Akku = 1;
volatile boolean_T      BN;
volatile boolean_T      Entladen;
volatile boolean_T      Ladeschluss;
volatile boolean_T      U_CAP_OK ;

//Tastverhältnis der DC-Wandler
volatile uint32_t tein_BN = 0;
volatile uint32_t tein_BAT = 0;

//Hilfsvariablen für das Mitteln der Stromwerte
volatile uint16_t I_BN_sample[10];
volatile uint16_t I_BN_mittel;
volatile uint16_t I_BAT_sample[10];
volatile uint16_t I_BAT_mittel;

/* Init Functions -----*/
void SystemClock_Config(void);
void Error_Handler(void);
static void MX_GPIO_Init(void);
static void MX_DMA_Init(void);
static void MX_ADC2_Init(void);
static void MX_HRTIM1_Init(void);
static void MX_I2C1_SMBUS_Init(void);
static void MX_SPI1_Init(void);

void HAL_HRTIM_MspPostInit(HRTIM_HandleTypeDef *hhrtim);
```

```
/* Private function prototypes -----*/
void getVoltage(void);
void getCurrent(void);
void ReglerStep(void);

int main(void) {

    /* MCU Configuration-----*/
    HAL_Init();
    /* Reset of all peripherals, Initializes the
    Flash interface and the SysTick. */

    /* Configure the system clock */
    SystemClock_Config();

    /* Initialize all configured peripherals */
    MX_GPIO_Init();
    // Elko ist über 2000hm verbunden
    HAL_GPIO_WritePin(GPIOB, I_CAP_Control_Pin, GPIO_PIN_RESET);
    // Akku verbunden wenn uC läuft
    HAL_GPIO_WritePin(GPIOB, Keep_BAT_alive_Pin, GPIO_PIN_SET);
    //nCS1 I_BAT inactive
    HAL_GPIO_WritePin(GPIOC, GPIO_PIN_5, GPIO_PIN_SET);
    //nCS2 I_BN inactive
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_SET);

    //Intialisiert den DMA
    MX_DMA_Init();
    //Intialisiert den ADC2
    MX_ADC2_Init();

    //SMBus Init
    MX_I2C1_SMBUS_Init();
    //SPI Init
    MX_SPI1_Init();

    //ADC Start
    HAL_ADC_Start_DMA(&hadc2, (uint32_t*)&ADC2values, 4);
```

```
/*Regler startet erst wenn der Zwischenkreis
vorgeladen ist*/
getVoltage();
double U_ZK_help =0;
while(U_ZK_help < 8.0)
{
    getVoltage();
    //spannungsteiler gain
    U_ZK_help = U_ZK*0.013;
}

MX_HRTIM1_Init();

//PWM Start
HAL_HRTIM_WaveformOutputStart(&hhrtim1, HRTIM_OUTPUT_TD1 |
    HRTIM_OUTPUT_TD2 | HRTIM_OUTPUT_TC1 | HRTIM_OUTPUT_TC2);
HAL_HRTIM_WaveformCounterStart(&hhrtim1,
    HRTIM_TIMERID_TIMER_D | HRTIM_TIMERID_TIMER_C);

    //Matlab-Regler initialisieren
    MatlabRegler_fptool_initialize();

    //Regler-Modell aktivieren
    SoftStop = 0;

/* Infinite loop */
while (1)
{

}

}
/*8kHz Systick Interrupt*/
void HAL_SYSTICK_Callback()
{
    // Spannungswerte aus dem DMA holen
    getVoltage();
}
```

```
// Stromwerte aus dem DMA holen
getCurrent();

// Matlab Regler mit Übergabeparamtern
ReglerStep();

//Tastverhältnis D_BN * PWM Timer Period
tein_BN = (uint32_t)(D_BN*18432);
//Tastverhältnis D_BAT * PWM Timer Period
tein_BAT = (uint32_t)(D_BAT*18432);

/* PWM Capture Compare Register schreiben */
HRTIM1->sTimerxRegs[HRTIM_TIMERINDEX_TIMER_D].CMP1xR=tein_BN;
HRTIM1->sTimerxRegs[HRTIM_TIMERINDEX_TIMER_C].CMP1xR=tein_BAT;

}

/* Matlab Regler Werte übergeben und ausführen*/
void ReglerStep(void)
{
    // Eingangsparameter übergeben
    MatlabRegler_fptool_U.I_BAT = I_BAT;
    MatlabRegler_fptool_U.I_BN = I_BN;
    MatlabRegler_fptool_U.U_BAT = U_BAT;
    MatlabRegler_fptool_U.U_BN = U_BN;
    MatlabRegler_fptool_U.U_CAP = U_CAP;
    MatlabRegler_fptool_U.U_ZK = U_ZK;
    MatlabRegler_fptool_U.SoftStop = SoftStop;

    /* Elko wird Niederimpedant verbunden
    wenn Zwischenkreisspannung U_ZK > 20 */
    if(MatlabRegler_fptool_Y.U_Cap_ok) {
        HAL_GPIO_WritePin(GPIOB, I_CAP_Control_Pin,
        GPIO_PIN_SET);
    }

    // Matlab Regler Code ausführen
    MatlabRegler_fptool_step();
}
```

```
// Regler Ausgänge übernehmen
D_BAT =MatlabRegler_fptool_Y.D_BAT;
D_BN = MatlabRegler_fptool_Y.D_BN;
}

/*SPI Stromsensor Receive Interrupt*/
void HAL_SPI_RxCpltCallback(SPI_HandleTypeDef *hspi)
{
    //SPI Sensor Auswahl
    if(!ncs1)
    {
        //Chip Select zurücksetzen
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4,
        GPIO_PIN_SET);
        /*Prüfen ob Nachricht mit Werten
        angekommen
        ist und ob Werte vorhanden sind */
        if((pData[1]>>7)!=1 && (pData[1]&
        pData[0])!=0)
        {
            //Werte aus SPI Nachricht
            //auslesen
            I_BNdata=((uint16_t) (pData[1]
            & 0x1F))<<8
            |((uint16_t) (pData[0]));
        }
    }
    else
    {
        HAL_GPIO_WritePin(GPIOC, GPIO_PIN_5,
        GPIO_PIN_SET);

        if((pData[1]>>7)!=1
        && (pData[3]
        &pData[2])!=0)
        {
            I_BATdata=((uint16_t)
            (pData[3] & 0x1F))<<8
            |((uint16_t) (pData[2]))
        }
    }
}
```

```
    }  
}  
  
void getVoltage(void)  
{  
  
    //      Konvertierung erfolgt in Matlab  
    //      Vref = 3,3V  
    //      R1 = 19100Ohm  
    //      R22 = 33000Ohm  
    //      R21 = 13000Ohm  
    //      U = ADCvalue * 3,3V * (R1+R2)/(4096 * R)  
    //      G1 = 15.7; %R1+R2/R1  
    //      G2 = 5.78  
  
    //static int a=0;  
    //DMA Register auslesen  
  
    U_CAP = ((uint16_t)ADC2values[0]);  
    U_ZK = ((uint16_t)ADC2values[1]);  
    U_BN = ((uint16_t)ADC2values[2]);  
    U_BAT = ((uint16_t)ADC2values[3]);  
  
}  
/*Strom Werte mitteln und SPI Sensoren ansteuern */  
void getCurrent(void) {  
  
    //static int b=0;  
    //Sind Werte vorhanden  
    if(I_BNdata > 0 && I_BATdata > 0) {  
        I_BN_sample[b] = I_BNdata;  
        I_BAT_sample[b] = I_BATdata;  
        //Werte Mitteln  
        I_BN_mittel = (uint16_t)((I_BN_sample[0]  
        + I_BN_sample[1]  
        + I_BN_sample[2] + I_BN_sample[3] + I_BN_sample[4]
```

```
+ I_BN_sample[5] + I_BN_sample[6] + I_BN_sample[7]
+ I_BN_sample[8] + I_BN_sample[9])/10));
I_BAT_mittel = (uint16_t)(((I_BAT_sample[0]
+ I_BAT_sample[1] + I_BAT_sample[2] + I_BAT_sample[3]
+ I_BAT_sample[4] + I_BAT_sample[5] + I_BAT_sample[6]
+ I_BAT_sample[7] + I_BAT_sample[8] + I_BAT_sample[9])
/10));
b = (b+1)%10;

I_BN= I_BN_mittel;
I_BAT = I_BAT_mittel;
}
//Chip select wechseln
ncs1 = !ncs1;

//Chip select setzen und SPI Receive
//über DMA aufrufen
if(!ncs1)
{

    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4,
    GPIO_PIN_RESET);
    HAL_SPI_Receive_DMA(&hspi1, &pData[0],1);

}
//Chip select setzen und SPI Receive
//über DMA aufrufen
else
{

    HAL_GPIO_WritePin(GPIOC, GPIO_PIN_5,
    GPIO_PIN_RESET);
    HAL_SPI_Receive_DMA(&hspi1, &pData[2],1);

}

}
```

```
void SystemClock_Config(void)
{

    RCC_OscInitTypeDef RCC_OscInitStruct;
    RCC_ClkInitTypeDef RCC_ClkInitStruct;
    RCC_PeriphCLKInitTypeDef PeriphClkInit;

    RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSI |
    RCC_OSCILLATORTYPE_HSE;
    RCC_OscInitStruct.HSEState = RCC_HSE_ON;
    RCC_OscInitStruct.HSEPredivValue = RCC_HSE_PREDIV_DIV1;
    RCC_OscInitStruct.HSIState = RCC_HSI_ON;
    RCC_OscInitStruct.HSICalibrationValue = 16;
    RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
    RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSE;
    RCC_OscInitStruct.PLL.PLLMUL = RCC_PLL_MUL9;
    if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
    {
        Error_Handler();
    }

    RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK
    |RCC_CLOCKTYPE_SYSCLK|RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
    RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
    RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
    RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV2;
    RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV1;
    if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_2)
        != HAL_OK)
    {
        Error_Handler();
    }

    PeriphClkInit.PeriphClockSelection = RCC_PERIPHCLK_HRTIM1
    |RCC_PERIPHCLK_I2C1;
    PeriphClkInit.I2c1ClockSelection = RCC_I2C1CLKSOURCE_HSI;
    PeriphClkInit.Hrtim1ClockSelection = RCC_HRTIM1CLK_PLLCLK;
    if (HAL_RCCEx_PeriphCLKConfig(&PeriphClkInit) != HAL_OK)
    {
        Error_Handler();
    }
}
```

```
}

HAL_RCC_MCOConfig(RCC_MCO, RCC_MCO1SOURCE_HSE, RCC_MCODIV_1);

HAL_SYSTICK_Config(HAL_RCC_GetHCLKFreq()/8000);

HAL_SYSTICK_CLKSourceConfig(SYSTICK_CLKSOURCE_HCLK);

/* SysTick_IRQn interrupt configuration */
HAL_NVIC_SetPriority(SysTick_IRQn, 0, 0);
}

/* ADC2 init function */
static void MX_ADC2_Init(void)
{

    ADC_AnalogWDGConfTypeDef AnalogWDGConfig;
    ADC_ChannelConfTypeDef sConfig;

    /**Common config
     */
    hadc2.Instance = ADC2;
    hadc2.Init.ClockPrescaler = ADC_CLOCK_SYNC_PCLK_DIV1;
    hadc2.Init.Resolution = ADC_RESOLUTION_12B;
    hadc2.Init.ScanConvMode = ADC_SCAN_ENABLE;
    hadc2.Init.ContinuousConvMode = ENABLE;
    hadc2.Init.DiscontinuousConvMode = DISABLE;
    hadc2.Init.ExternalTrigConvEdge =
    ADC_EXTERNALTRIGCONVEDGE_NONE;
    hadc2.Init.DataAlign = ADC_DATAALIGN_RIGHT;
    hadc2.Init.NbrOfConversion = 4;
    hadc2.Init.DMAContinuousRequests = ENABLE;
    hadc2.Init.EOCSelection = ADC_EOC_SEQ_CONV;
    hadc2.Init.LowPowerAutoWait = DISABLE;
    hadc2.Init.Overrun = ADC_OVR_DATA_OVERWRITTEN;
    if (HAL_ADC_Init(&hadc2) != HAL_OK)
    {
        Error_Handler();
    }
}
```

```
    /**Configure Regular Channel
    */
    sConfig.Channel = ADC_CHANNEL_3;
    sConfig.Rank = 1;
    sConfig.SingleDiff = ADC_SINGLE_ENDED;
    sConfig.SamplingTime = ADC_SAMPLETIME_601CYCLES_5;
    sConfig.OffsetNumber = ADC_OFFSET_NONE;
    sConfig.Offset = 0;
    if (HAL_ADC_ConfigChannel(&hadc2, &sConfig) != HAL_OK)
    {
        Error_Handler();
    }

    /**Configure Regular Channel
    */
    sConfig.Channel = ADC_CHANNEL_4;
    sConfig.Rank = 2;
    if (HAL_ADC_ConfigChannel(&hadc2, &sConfig) != HAL_OK)
    {
        Error_Handler();
    }

    /**Configure Regular Channel
    */
    sConfig.Channel = ADC_CHANNEL_7;
    sConfig.Rank = 3;
    if (HAL_ADC_ConfigChannel(&hadc2, &sConfig) != HAL_OK)
    {
        Error_Handler();
    }

    /**Configure Regular Channel
    */
    sConfig.Channel = ADC_CHANNEL_8;
    sConfig.Rank = 4;
```

```
    if (HAL_ADC_ConfigChannel(&hadc2, &sConfig) != HAL_OK)
    {
        Error_Handler();
    }

    /**Configure Analog WatchDog 1 (wird nicht verwendet)
    */
    AnalogWDGConfig.WatchdogNumber = ADC_ANALOGWATCHDOG_1;
    AnalogWDGConfig.WatchdogMode = ADC_ANALOGWATCHDOG_SINGLE_REG;
    AnalogWDGConfig.HighThreshold = 4095;
    AnalogWDGConfig.LowThreshold = 1150;
    AnalogWDGConfig.Channel = ADC_CHANNEL_3;
    AnalogWDGConfig.ITMode = ENABLE;
    //if (HAL_ADC_AnalogWDGConfig(&hadc2, &AnalogWDGConfig)
        //!= HAL_OK)
    //{
    //    Error_Handler();
    //}

    AnalogWDGConfig.WatchdogNumber = ADC_ANALOGWATCHDOG_2;
    AnalogWDGConfig.WatchdogMode = ADC_ANALOGWATCHDOG_SINGLE_REG;
    AnalogWDGConfig.HighThreshold = 3221;
    AnalogWDGConfig.LowThreshold = 2177;
    AnalogWDGConfig.Channel = ADC_CHANNEL_7;
    AnalogWDGConfig.ITMode = ENABLE;
    //if (HAL_ADC_AnalogWDGConfig(&hadc2, &AnalogWDGConfig) != HAL_OK)
    //{
    //    Error_Handler();
    //}

    AnalogWDGConfig.WatchdogNumber = ADC_ANALOGWATCHDOG_3;
    AnalogWDGConfig.WatchdogMode = ADC_ANALOGWATCHDOG_SINGLE_REG;
    AnalogWDGConfig.HighThreshold = 3048;
    AnalogWDGConfig.LowThreshold = 2322;
    AnalogWDGConfig.Channel = ADC_CHANNEL_8;
    AnalogWDGConfig.ITMode = ENABLE;
    //if (HAL_ADC_AnalogWDGConfig(&hadc2, &AnalogWDGConfig)
        //!= HAL_OK)
    //{
    //    Error_Handler();
    //}
```

```
//}  
//  
//  
}  
  
/* HRTIM1 init function */  
static void MX_HRTIM1_Init(void)  
{  
  
    HRTIM_TimeBaseCfgTypeDef pTimeBaseCfg;  
    HRTIM_TimerCfgTypeDef pTimerCfg;  
    HRTIM_CompareCfgTypeDef pCompareCfg;  
    HRTIM_OutputCfgTypeDef pOutputCfg;  
    HRTIM_DeadTimeCfgTypeDef pDeadTimeCfg;  
  
    hhrtim1.Instance = HRTIM1;  
    hhrtim1.Init.HRTIMInterruptResquests = HRTIM_IT_NONE;  
    hhrtim1.Init.SyncOptions = HRTIM_SYNCOPTION_NONE;  
    if (HAL_HRTIM_Init(&hhrtim1) != HAL_OK)  
    {  
        Error_Handler();  
    }  
  
    if (HAL_HRTIM_DLLCalibrationStart(&hhrtim1,  
    HRTIM_CALIBRATIONRATE_7300)  
        != HAL_OK)  
    {  
        Error_Handler();  
    }  
  
    if (HAL_HRTIM_PollForDLLCalibration(&hhrtim1, 10) != HAL_OK)  
    {  
        Error_Handler();  
    }  
  
    pTimeBaseCfg.Period = 18432;  
    pTimeBaseCfg.RepetitionCounter = 0x00;  
    pTimeBaseCfg.PrescalerRatio = HRTIM_PRESCALERRATIO_MUL32;  
    pTimeBaseCfg.Mode = HRTIM_MODE_CONTINUOUS;  
    if (HAL_HRTIM_TimeBaseConfig(&hhrtim1, HRTIM_TIMERINDEX_TIMER_C
```

```
, &pTimeBaseCfg) != HAL_OK)
{
    Error_Handler();
}

pTimerCfg.InterruptRequests = HRTIM_TIM_IT_NONE;
pTimerCfg.DMARequests = HRTIM_TIM_DMA_NONE;
pTimerCfg.DMASrcAddress = 0x0000;
pTimerCfg.DMADstAddress = 0x0000;
pTimerCfg.DMASize = 0x1;
pTimerCfg.HalfModeEnable = HRTIM_HALFMODE_DISABLED;
pTimerCfg.StartOnSync = HRTIM_SYNCSTART_ENABLED;
pTimerCfg.ResetOnSync = HRTIM_SYNCRESET_DISABLED;
pTimerCfg.DACSynchro = HRTIM_DACSYNC_NONE;
pTimerCfg.PreloadEnable = HRTIM_PRELOAD_ENABLED;
pTimerCfg.UpdateGating = HRTIM_UPDATEGATING_INDEPENDENT;
pTimerCfg.BurstMode = HRTIM_TIMERBURSTMODE_MAINTAINCLOCK;
pTimerCfg.RepetitionUpdate = HRTIM_UPDATEONREPETITION_ENABLED;
pTimerCfg.PushPull = HRTIM_TIMPUSHPULLMODE_DISABLED;
pTimerCfg.FaultEnable = HRTIM_TIMFAULTENABLE_NONE;
pTimerCfg.FaultLock = HRTIM_TIMFAULTLOCK_READWRITE;
pTimerCfg.DeadTimeInsertion = HRTIM_TIMDEADTIMEINSERTION_ENABLED;
pTimerCfg.DelayedProtectionMode
= HRTIM_TIMER_A_B_C_DELAYEDPROTECTION_DISABLED;
pTimerCfg.UpdateTrigger = HRTIM_TIMUPDATETRIGGER_NONE;
pTimerCfg.ResetTrigger = HRTIM_TIMRESETTRIGGER_NONE;
pTimerCfg.ResetUpdate = HRTIM_TIMUPDATEONRESET_DISABLED;
if (HAL_HRTIM_WaveformTimerConfig(&hhrtim1,
HRTIM_TIMERINDEX_TIMER_C, &pTimerCfg) != HAL_OK)
{
    Error_Handler();
}

pTimerCfg.DelayedProtectionMode =
HRTIM_TIMER_D_E_DELAYEDPROTECTION_DISABLED;
if (HAL_HRTIM_WaveformTimerConfig(&hhrtim1,
HRTIM_TIMERINDEX_TIMER_D, &pTimerCfg) != HAL_OK)
{
    Error_Handler();
}
```

```
pCompareCfg.CompareValue = 400; //min PWM Wert
if (HAL_HRTIM_WaveformCompareConfig(&hhrtim1,
HRTIM_TIMERINDEX_TIMER_C, HRTIM_COMPAREUNIT_1,
&pCompareCfg) != HAL_OK)
{
    Error_Handler();
}

if (HAL_HRTIM_WaveformCompareConfig(&hhrtim1,
HRTIM_TIMERINDEX_TIMER_D, HRTIM_COMPAREUNIT_1,
&pCompareCfg) != HAL_OK)
{
    Error_Handler();
}

pOutputCfg.Polarity = HRTIM_OUTPUTPOLARITY_HIGH;
pOutputCfg.SetSource = HRTIM_OUTPUTRESET_TIMPER;
pOutputCfg.ResetSource = HRTIM_OUTPUTSET_TIMCMP1 ;
pOutputCfg.IdleMode = HRTIM_OUTPUTIDLEMODE_NONE;
pOutputCfg.IdleLevel = HRTIM_OUTPUTIDLELEVEL_INACTIVE;
pOutputCfg.FaultLevel = HRTIM_OUTPUTFAULTLEVEL_NONE;
pOutputCfg.ChopperModeEnable = HRTIM_OUTPUTCHOPPERMODE_DISABLED;
pOutputCfg.BurstModeEntryDelayed
= HRTIM_OUTPUTBURSTMODEENTRY_REGULAR;
if (HAL_HRTIM_WaveformOutputConfig(&hhrtim1,
HRTIM_TIMERINDEX_TIMER_C, HRTIM_OUTPUT_TC1, &pOutputCfg)
!= HAL_OK)
{
    Error_Handler();
}

if (HAL_HRTIM_WaveformOutputConfig(&hhrtim1,
HRTIM_TIMERINDEX_TIMER_D, HRTIM_OUTPUT_TD1, &pOutputCfg)
!= HAL_OK)
{
    Error_Handler();
}

pOutputCfg.SetSource = HRTIM_OUTPUTSET_NONE;
```

```
pOutputCfg.ResetSource = HRTIM_OUTPUTRESET_NONE;
if (HAL_HRTIM_WaveformOutputConfig(&hhrtim1,
HRTIM_TIMERINDEX_TIMER_C, HRTIM_OUTPUT_TC2, &pOutputCfg)
!= HAL_OK)
{
    Error_Handler();
}

if (HAL_HRTIM_WaveformOutputConfig(&hhrtim1,
HRTIM_TIMERINDEX_TIMER_D, HRTIM_OUTPUT_TD2, &pOutputCfg)
!= HAL_OK)
{
    Error_Handler();
}

pDeadTimeCfg.Prescaler = HRTIM_TIMDEADTIME_PRESCALERRATIO_MUL8;
pDeadTimeCfg.RisingValue = 50;
pDeadTimeCfg.RisingSign = HRTIM_TIMDEADTIME_RISINGSIGN_POSITIVE;
pDeadTimeCfg.RisingLock = HRTIM_TIMDEADTIME_RISINGLOCK_WRITE;
pDeadTimeCfg.RisingSignLock =
HRTIM_TIMDEADTIME_RISINGSIGNLOCK_READONLY;
pDeadTimeCfg.FallingValue = 50;
pDeadTimeCfg.FallingSign = HRTIM_TIMDEADTIME_RISINGSIGN_POSITIVE;
pDeadTimeCfg.FallingLock = HRTIM_TIMDEADTIME_FALLINGLOCK_WRITE;
pDeadTimeCfg.FallingSignLock =
HRTIM_TIMDEADTIME_FALLINGSIGNLOCK_WRITE;
if (HAL_HRTIM_DeadTimeConfig(&hhrtim1, HRTIM_TIMERINDEX_TIMER_C,
&pDeadTimeCfg) != HAL_OK)
{
    Error_Handler();
}

if (HAL_HRTIM_DeadTimeConfig(&hhrtim1, HRTIM_TIMERINDEX_TIMER_D,
&pDeadTimeCfg) != HAL_OK)
{
    Error_Handler();
}

if (HAL_HRTIM_TimeBaseConfig(&hhrtim1, HRTIM_TIMERINDEX_TIMER_D,
&pTimeBaseCfg) != HAL_OK)
```

```
{
    Error_Handler();
}

HAL_HRTIM_MspPostInit(&hhrtim1);

}

/* I2C1 init function */
static void MX_I2C1_SMBUS_Init(void)
{
    hsmbus1.Instance = I2C1;
    hsmbus1.Init.Timing = 0x2000090E;
    hsmbus1.Init.AnalogFilter = SMBUS_ANALOGFILTER_ENABLE;
    hsmbus1.Init.OwnAddress1 = 0;
    hsmbus1.Init.AddressingMode = SMBUS_ADDRESSINGMODE_7BIT;
    hsmbus1.Init.DualAddressMode = SMBUS_DUALADDRESS_DISABLE;
    hsmbus1.Init.OwnAddress2 = 0;
    hsmbus1.Init.OwnAddress2Masks = SMBUS_OA2_NOMASK;
    hsmbus1.Init.GeneralCallMode = SMBUS_GENERALCALL_DISABLE;
    hsmbus1.Init.NoStretchMode = SMBUS_NOSTRETCH_DISABLE;
    hsmbus1.Init.PacketErrorCheckMode = SMBUS_PEC_DISABLE;
    hsmbus1.Init.PeripheralMode =
    SMBUS_PERIPHERAL_MODE_SMBUS_SLAVE;
    hsmbus1.Init.SMBusTimeout = 0x00008061;
    if (HAL_SMBUS_Init(&hsmbus1) != HAL_OK)
    {
        Error_Handler();
    }
}

/* SPI1 init function */
static void MX_SPI1_Init(void)
{
    hspi1.Instance = SPI1;
    hspi1.Init.Mode = SPI_MODE_MASTER;
    hspi1.Init.Direction = SPI_DIRECTION_2LINES_RXONLY;
```

```
    hspi1.Init.DataSize = SPI_DATASIZE_16BIT;
    hspi1.Init.CLKPolarity = SPI_POLARITY_LOW;
    hspi1.Init.CLKPhase = SPI_PHASE_2EDGE;
    hspi1.Init.NSS = SPI_NSS_SOFT;
    hspi1.Init.BaudRatePrescaler = SPI_BAUDRATEPRESCALER_16;
    hspi1.Init.FirstBit = SPI_FIRSTBIT_MSB;
    hspi1.Init.TIMode = SPI_TIMODE_DISABLE;
    hspi1.Init.CRCCalculation = SPI_CRCCALCULATION_DISABLE;
    hspi1.Init.CRCPolynomial = 7;
    hspi1.Init.CRCLength = SPI_CRC_LENGTH_DATASIZE;
    hspi1.Init.NSSPMode = SPI_NSS_PULSE_ENABLE;
    if (HAL_SPI_Init(&hspi1) != HAL_OK)
    {
        Error_Handler();
    }
}

/**
 * Enable DMA controller clock
 */
static void MX_DMA_Init(void)
{
    /* DMA controller clock enable */
    __HAL_RCC_DMA1_CLK_ENABLE();

    /* DMA interrupt init */
    /* DMA1_Channel2_IRQn interrupt configuration */
    HAL_NVIC_SetPriority(DMA1_Channel2_IRQn, 0, 0);
    HAL_NVIC_EnableIRQ(DMA1_Channel2_IRQn);
    /* DMA1_Channel4_IRQn interrupt configuration */
    HAL_NVIC_SetPriority(DMA1_Channel4_IRQn, 0, 0);
    HAL_NVIC_EnableIRQ(DMA1_Channel4_IRQn);
    /* DMA1_Channel5_IRQn interrupt configuration */
    HAL_NVIC_SetPriority(DMA1_Channel5_IRQn, 0, 0);
    HAL_NVIC_EnableIRQ(DMA1_Channel5_IRQn);
    /* DMA1_Channel6_IRQn interrupt configuration */
    HAL_NVIC_SetPriority(DMA1_Channel6_IRQn, 0, 0);
    HAL_NVIC_EnableIRQ(DMA1_Channel6_IRQn);
}
```

```
}

/** Configure pins as
    * Analog
    * Input
    * Output
    * EVENT_OUT
    * EXTI
    PA2  -----> USART2_TX
    PA3  -----> USART2_RX
    PA8  -----> RCC_MCO
*/
static void MX_GPIO_Init (void)
{

    GPIO_InitTypeDef GPIO_InitStructure;

    /* GPIO Ports Clock Enable */
    __HAL_RCC_GPIOC_CLK_ENABLE();
    __HAL_RCC_GPIOF_CLK_ENABLE();
    __HAL_RCC_GPIOA_CLK_ENABLE();
    __HAL_RCC_GPIOB_CLK_ENABLE();

    /*Configure GPIO pins : PA2 USART_RX_Pin */
    GPIO_InitStructure.Pin = GPIO_PIN_2|USART_RX_Pin;
    GPIO_InitStructure.Mode = GPIO_MODE_AF_PP;
    GPIO_InitStructure.Pull = GPIO_NOPULL;
    GPIO_InitStructure.Speed = GPIO_SPEED_FREQ_HIGH;
    GPIO_InitStructure.Alternate = GPIO_AF7_USART2;
    HAL_GPIO_Init(GPIOA, &GPIO_InitStructure);

    /*Configure GPIO pin : nCS_I_BN_Sensor_Pin */
    GPIO_InitStructure.Pin = nCS_I_BN_Sensor_Pin;
    GPIO_InitStructure.Mode = GPIO_MODE_OUTPUT_PP;
    GPIO_InitStructure.Pull = GPIO_NOPULL;
    GPIO_InitStructure.Speed = GPIO_SPEED_FREQ_HIGH;
    HAL_GPIO_Init(nCS_I_BN_Sensor_GPIO_Port, &GPIO_InitStructure);

    /*Configure GPIO pin : nCS_I_BAT_Sensor_Pin */
    GPIO_InitStructure.Pin = nCS_I_BAT_Sensor_Pin;
```

```
GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
GPIO_InitStruct.Pull = GPIO_NOPULL;
GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_HIGH;
HAL_GPIO_Init(nCS_I_BAT_Sensor_GPIO_Port, &GPIO_InitStruct);

/*Configure GPIO pin : PA8 */
GPIO_InitStruct.Pin = GPIO_PIN_8;
GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
GPIO_InitStruct.Pull = GPIO_NOPULL;
GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
GPIO_InitStruct.Alternate = GPIO_AF0_MCO;
HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);

/*Configure GPIO pins : Keep_BAT_alive_Pin I_CAP_Control_Pin */
GPIO_InitStruct.Pin = Keep_BAT_alive_Pin;
GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
GPIO_InitStruct.Pull = GPIO_NOPULL;
GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);

/*Configure GPIO pins : Keep_BAT_alive_Pin I_CAP_Control_Pin */
GPIO_InitStruct.Pin = I_CAP_Control_Pin;
GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
GPIO_InitStruct.Pull = GPIO_NOPULL;
GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);

/*Configure GPIO pin Output Level */
HAL_GPIO_WritePin(nCS_I_BN_Sensor_GPIO_Port,
nCS_I_BN_Sensor_Pin, GPIO_PIN_RESET);

/*Configure GPIO pin Output Level */
HAL_GPIO_WritePin(nCS_I_BAT_Sensor_GPIO_Port,
nCS_I_BAT_Sensor_Pin, GPIO_PIN_RESET);

/*Configure GPIO pin Output Level */
HAL_GPIO_WritePin(GPIOB, I_CAP_Control_Pin, GPIO_PIN_RESET);
HAL_GPIO_WritePin(GPIOB, Keep_BAT_alive_Pin, GPIO_PIN_SET);
}
```

Glossar

ADC Analog-Digital-Converter ist ein Bauelement zur Umsetzung von analogen Signalen in digitale Datenströme.

BLDC Brushless-Direct-Current-Motor oder Bürstenloser Gleichstrommotor ist ein Motor, welcher durch ein angelegtes drehendes Magnetfeld ein Drehmoment auf einen Permanentmagnet ausübt.

Boost-Wandler Der Boost-Wandler oder auch Boost-Konverter ist eine Schaltwandlertopologie, welche ihre Eingangsspannung auf eine höhere Ausgangsspannung transformieren kann.

Buck-Boost-Wandler Der Buck-Boost-Wandler ist eine Kombination aus Buck- und Boost-Wandler und kann seine Eingangsspannung auf eine höhere wie auch geringere Ausgangsspannung transformieren.

Buck-Wandler Der Buck-Wandler, oder auch Buck-Konverter, ist eine Schaltwandlertopologie, welche ihre Eingangsspannung auf eine kleinere Ausgangsspannung transformieren kann.

EMV Die elektromagnetische Verträglichkeit ist ein Zeichen für die Störbeeinflussung zwischen Geräten und beruht auf ungewollten elektrischen sowie elektromagnetischen Effekten.

ESR Mit dem Equivalent-Series-Resistance, oder auch Ersatzreihenwiderstand, werden die inneren Verluste eines Kondensators beschrieben.

I2C Der Inter-Integrated-Circuit ist ein serieller Datenbus für die Kommunikation zwischen digitalen Schaltungen.

IC Integrated-Circuit, oder integrierte Schaltung, ist eine elektronische Schaltung, zusammengefasst auf einem Chip.

MOSFET Der Metal-Oxid-Semiconductor-Field-Effect-Transistor ist ein Halbleiterschaltetelement.

PEEP Positive-End-Expiration Pressure ist eine Beatmungstherapie, bei der beim Ausatmen ein Restdruck in der Lunge eingestellt wird.

Rdson Ist der Widerstand über der Drain-Source-Spannung im leitenden Betrieb eines MOSFET.

SMBus Der System-Management-Bus ist ein Zweileiter-Bus für die Kommunikation zwischen Baugruppen und baut technisch auf I2C auf.

SPI Das Serial-Peripheral-Interface ist ein Datenbus für digitale Schaltungen nach dem Master-Slave-Prinzip.

Versicherung über die Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §16(5) APSO-TI-BM 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, 20. September 2016

Ort, Datum

Unterschrift