

# Universität Bonn

## Physikalisches Institut

### **Entwicklung einer Ansteuerelektronik für die Netzgeräte der ELSA-Hauptmagnete**

Thomas Perlitius

Physikalisches Institut der  
Universität Bonn  
Nussallee 12  
D-53115 Bonn



FEB 2014



# Universität Bonn

## Physikalisches Institut

### **Entwicklung einer Ansteuerelektronik für die Netzgeräte der ELSA-Hauptmagnete**

Thomas Perlitius

Dieser Forschungsbericht wurde als Diplomarbeit von der Mathematisch-Naturwissenschaftlichen Fakultät der Universität Bonn angenommen.

Angenommen am: 03.02.2014  
1. Gutachter: PD Dr. Wolfgang Hillert  
2. Gutachter: Prof. Dr. Reinhard Beck



---

# Inhaltsverzeichnis

---

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Elektronen-Stretcher-Anlage ELSA . . . . .	1
1.2	Anforderungen an die Magnetfeldrampen . . . . .	4
1.3	Netzgeräte der Hauptmagnete . . . . .	4
1.4	Ansteuerung der Magnete . . . . .	6
1.5	Zielsetzung dieser Arbeit . . . . .	7
<b>2</b>	<b>Konzeption</b>	<b>9</b>
2.1	Strukturierung . . . . .	9
2.2	HOLEC-Interface Card . . . . .	10
2.3	Verwendeter Mikrocontroller und Werkzeuge . . . . .	13
2.4	Sprachkonstrukte der Sprache C . . . . .	15
<b>3</b>	<b>Schaltungen und Firmware</b>	<b>17</b>
3.1	L4 – ELSA-Kontrollsystem . . . . .	17
3.2	C3 – Ethernet-Protokoll . . . . .	17
3.3	L3 – Kommunikations-Modul . . . . .	19
3.3.1	Übersicht . . . . .	19
3.3.2	C3 – Ethernet-Schaltungsteil . . . . .	19
3.3.3	C2 – SPI-Schaltungsteil . . . . .	20
3.3.4	Weitere Schaltungsteile . . . . .	20
3.3.5	C2 – C2pak-Protokoll . . . . .	21
3.3.6	L3 – Programmierung . . . . .	22
3.3.7	L3 – Platinen-Layout . . . . .	23
3.4	L2 – Prozess-Modul . . . . .	26
3.4.1	Übersicht . . . . .	26
3.4.2	SPI-Schaltungsteil . . . . .	26
3.4.3	Optokoppler-Schaltungsteil . . . . .	26
3.4.4	Gesamtschaltung . . . . .	26
3.4.5	L2 – Programmierung . . . . .	27
3.4.6	L2 – Platinen-Layout . . . . .	30
3.4.7	Unterschiede der Bridge-Select-Variante . . . . .	30
3.5	L1 – Trigger-Modul . . . . .	32
<b>4</b>	<b>Die L3-/L2-Befehle</b>	<b>35</b>
4.1	Allgemeines . . . . .	35

4.2	L2-Statusflags . . . . .	36
4.3	Aufbau und Funktion der Befehle . . . . .	37
<b>5</b>	<b>Entwicklung und Test der Ansteuerelektronik</b>	<b>43</b>
5.1	Entwicklung . . . . .	43
5.2	Tests der neuen Ansteuerung . . . . .	44
<b>6</b>	<b>Zusammenfassung</b>	<b>47</b>
<b>A</b>	<b>HOLEC-IFC</b>	<b>49</b>
<b>B</b>	<b>Schaltungsdokumentation</b>	<b>51</b>
	<b>Literatur</b>	<b>59</b>

---

## Einleitung

---

### 1.1 Elektronen-Stretcher-Anlage ELSA

Die Elektronen-Stretcher-Anlage ELSA ist ein dreistufiger Beschleuniger, welcher mehrere hadronenphysikalische Experimente mit Elektronen mit Energien von bis zu 3,2 GeV versorgt. Eine Übersicht der Beschleunigeranlage ist Abbildung 1.1 zu sehen. ELSA verfügt über zwei, als erste Vorbeschleunigungsstufe fungierende, Linearbeschleuniger (LINAC), von denen der LINAC 1 für den Hochstrombetrieb optimiert wurde. Im Standardbetrieb werden die Elektronen, welche wahlweise einer thermischen Quelle für unpolarisierte Elektronen oder einer Quelle für spinpolarisierte Elektronen entnommen werden, im LINAC 2 auf eine Energie von 26 MeV vorbeschleunigt.

Die zweite Beschleuniger-Stufe besteht aus einem netzsynchron betriebenen Booster-Synchrotron, welches keine längere Speicherung oder eine langsame Extraktion des Elektronenstrahls erlaubt. Nachdem die aus dem LINAC stammenden Elektronen mit dem Booster-Synchrotron auf eine Energie von bis zu 1,6 GeV beschleunigt wurden, werden sie extrahiert und gelangen über einen Transferkanal in den 164 m langen Stretcherring. Zur Akkumulation des Strahls wird dieser Vorgang mehrfach wiederholt.

Mit dem Stretcherring wird neben einer nochmaligen Energieerhöhung auf bis zu 3,2 GeV, besonders eine Verbesserung des makroskopischen Tastverhältnisses erzielt. Darunter wird das Verhältnis der Extraktionszeit, in welcher die Experimente mit Elektron versorgt werden, zur Gesamtdauer eines Zyklus verstanden. Die beim Stretcherring vorliegende Aufteilung der strahlablenkenden und strahlfokussierenden Funktion auf verschiedene Magnete definiert eine separate-function-Maschine.

Der Stretcherring kann in drei Modi betrieben werden: Der Speichermodus ist für die Nutzung des Synchrotronlichts konzipiert. Nach der Akkumulation und Beschleunigung des Strahles wird dieser nicht extrahiert, sondern über Zeiträume bis zu mehrere Stunden gespeichert. Die von dem im Stretcherring umlaufende Elektronenstrahl erzeugte Synchrotronstrahlung wird gegenwärtig nur für beschleunigerphysikalische Untersuchungen genutzt.

Die Funktion des Stretchermodus ist es, eine Erhöhung des makroskopischen Tastverhältnisses bei Beibehaltung der Energie zu erzielen. Hierzu werden bei jedem Zyklus des Booster-Synchrotrons der Elektronenstrahl aus diesem in den Stretcherring transferiert. Aus dem Stretcherring wird der gespeicherte Strahlstrom bis zur nächsten Injektion kontinuierlich zu den Experimenten extrahiert.

Der Nachbeschleunigungsmodus stellt besonders für den Experimentierbetrieb die wichtigste Betriebsart dar. Der zeitliche Verlauf der Größen Strahlstrom, Teilchenenergie und extrahierter Strom sind in Abbildung 1.2 aufgetragen. Ein solcher Zyklus besteht aus mehreren Phasen. Es werden zunächst





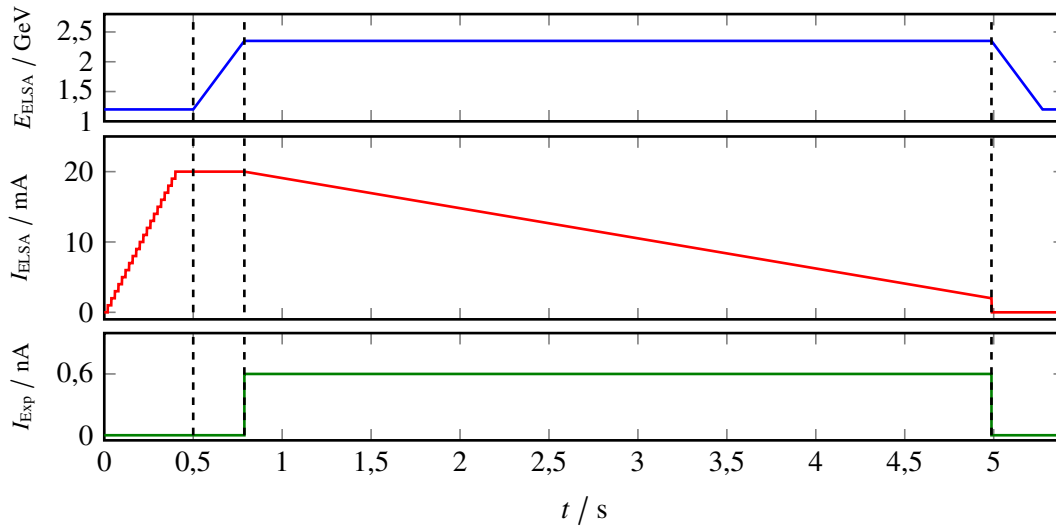


Abbildung 1.2: Darstellung des Nachbeschleunigungsmodus der Beschleunigeranlage ELSA.  $I_{\text{ELSA}}$  bezeichnet den im Stretcherring gespeicherten Strahlstrom (—),  $I_{\text{Exp}}$  den zu den Experimentierplätzen extrahierten Strom (—) und  $E_{\text{ELSA}}$  die Energie der im Stretcherring gespeicherten Elektronen (—). Es sind typische Werte aufgetragen. [Pro11]

typischerweise 21 Extraktionen aus dem Booster-Synchrotron im Stretcherring akkumuliert. Nach einer kurzen Dämpfungszeit können die Elektronen auf eine Energie von bis zu 3,2 GeV beschleunigt werden. Anschließend wird nach einer kurzen Präparationszeit mit der Extraktion über mehrere Sekunden bis hin zu einigen Minuten begonnen. Für die nächste Injektion muss die Maschine dann wieder auf die Injektions-Energie eingestellt werden. Zum Erzielen eines hohen makroskopischen Tastverhältnisses gilt es, neben den unvermeidbaren Dämpfungs- und Präparationsphasen besonders die Zeit zur Energieerhöhung kurz zu gestalten. Dazu wurde der Stretcherring für Energieanstiegszeiten von bis zu 7 GeV/s ausgelegt. Die Wahl des Betriebsmodus, der Endenergie und des Timings legt eine Energie-“Rampe“ fest. Rampe meint hier die Beschreibung des zeitlichen Verlaufs einer Größe (z.B. der Teilchenenergie) über einen Zyklus. Das Timing wird durch die Festlegung der Dauer der verschiedenen Phasen des Beschleunigerzyklus festgelegt.

Der extrahierte Elektronenstrahl wird auf ein Konverter-Target gelenkt, in welchem er durch Bremsstrahlung Photonen erzeugt. Indem die Energie der dabei gestreuten Elektronen gemessen wird, kann mit der bekannten Anfangsenergie, die Photonenenergie bestimmt werden. Mit den hierdurch energie-markierten Photonen werden an ELSA mit dem BGO-OD-Experiment photoproduzierte Endzustände mit geladenen und ungeladenen Mesonen untersucht. Mit dem Crystal-Barrel/TAPS-Experiment werden Anregungszustände von Nukleonen erforscht.

Der Stretcherring besitzt mehrere Magnetfamilien. Die wichtigsten von ihnen sind die Dipolmagnete zur Strahlführung, die fokussierenden und defokussierenden Quadrupolmagnete zur Strahlbündelung sowie die fokussierenden und defokussierenden Sextupolmagnete zur Kompensation von Chromatizitätseffekten. Für die oben erwähnte Resonanzextraktion werden Extraktions-Sextupolmagnete benutzt. Darüber hinaus gibt es weitere Magnete zur Orbit-Korrektur sowie zur Injektion und Extraktion.

Diese Arbeit umfasst die Erneuerung der Ansteuerung der Netzgeräte für die Dipol-, Quadrupol- und Sextupol-Magnete des Stretcherrings. Dies beinhaltet das Entwickeln mehrerer Schaltungen und die Anbindung an das Kontrollsystem. Um den Inhalt dieser Aufgabe genauer formulieren zu können, werden zunächst einige Informationen über die Netzgeräte und eine Beschreibung der bisherigen Ansteuerung

gegeben.

## 1.2 Anforderungen an die Magnetfeldrampen

Da die Elektronen im Stretcherring auf einer Teilchenbahn mit festem Ablenkradius  $R$  umlaufen sollen, ist es nötig die Magnetfeldstärke  $B$  an die gewünschte Teilchenenergie  $E$  anzupassen:

$$B(E) = \frac{E}{ecR}.$$

Somit ergibt sich aus der vorgegebenen Energierampe direkt eine Magnetfeldstärkenrampe für die Dipolmagnete. Für den Strom der felderzeugenden Spulen der Magnete lässt sich hieraus eine Stromrampe ableiten. Bei größeren Strömen wird der prinzipiell proportionale Zusammenhang durch Sättigungseffekte des Magnetfeldes verfälscht.

Elektronen mit einer Energieabweichung zur Sollenergie besitzen auch eine, auf das Sollteilchen bezogene, abweichende Umlaufzeit. In den Beschleunigungselementen (Hohlraumresonatoren) können die unterschiedlichen Umlaufzeiten genutzt werden, um der Energieabweichung entgegen zu wirken. Diese sogenannte longitudinale Phasenfokussierung bewirkt, dass die Energie der Elektronen über die Stärke des Dipolfeldes eingestellt werden kann. Somit ist eine hinreichende Genauigkeit bei der Ansteuerung des entsprechenden Netzgerätes nötig.

Die Stärken der Quadrupole und Sextupole legen die optischen Eigenschaften des Stretcherrings wie die Strahlfokussierung fest. Aufgrund dieser Fokussierung des Elektronenstrahls führen die Elektronen Schwingungen um die Sollbahn aus, die sogenannten Betatron-Schwingungen. Die Anzahl dieser Schwingungen pro Umlauf definiert den Arbeitspunkt  $Q$ .

Fehler der Magnetfelder stellen für den umlaufenden Strahl eine periodische Störung dar. Diese kann bei ungünstiger Wahl des Arbeitspunktes die Betatronschwingung resonant anregen und zu Strahlverlust führen, dies bezeichnet man als optische Resonanz. Bei Dipolfeldfehlern ist dies bei allen ganzzahligen Arbeitspunkten der Fall. Bei höheren Multipolfeldern trifft dies bei halbzahligen, drittelzahligen, usw. Arbeitspunkten zu. In den beiden zur Strahlrichtung transversalen Richtungen ( $x, z$ ) liegt im allgemeinen ein voneinander verschiedener Arbeitspunkt vor [Wil96].

Berücksichtigt man noch das Auftreten von Kopplung dieser Resonanzen, ergibt sich eine Vielzahl von optischen Resonanzen. Abbildung 1.3 zeigt ein  $Q_x$ - $Q_z$ -Diagramm mit eingezeichneten optischen Resonanzen. Um einen stabilen Strahlbetrieb zu erreichen, ist es notwendig, die Arbeitspunkte mit genügend Abstand zu diesen Resonanzen zu wählen. Für einen separate-function-Beschleuniger bedeutet dies, dass die Magnete sowohl die geforderten Magnetfeldstärken mit hinreichender Präzision erzeugen müssen als auch, dass die unterschiedlichen Magnetfamilien mit genau synchronisierten Stromrampen angesteuert werden müssen. Beides muss durch die Ansteuerung der Netzgeräte der Magnete sichergestellt werden.

Mit den Extraktions-Sextupolen wird gezielt eine drittelzahlige Resonanz angeregt, welche bei der Resonanzextraktion genutzt wird. Auch hierfür wird eine Stromrampe mit hinreichender Genauigkeit und Synchronität benötigt. Auch andere Systeme des Beschleunigers, wie zum Beispiel das Multi-Bunch-Feedback-System, benötigen einen stabilen Arbeitspunkt.

## 1.3 Netzgeräte der Hauptmagnete

Im Folgenden sollen die Eigenschaften der Netzgeräte der Hauptmagnete dargestellt werden. Eine Übersicht über die Kenndaten der Netzgeräte gibt Tabelle 1.1.

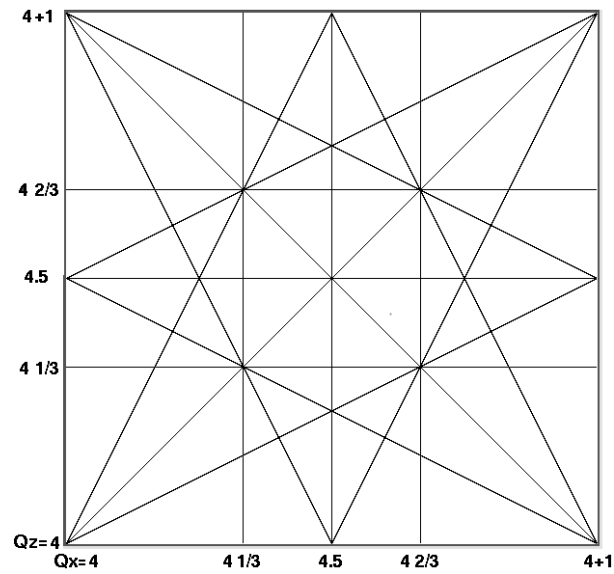


Abbildung 1.3:  $Q_x$ - $Q_z$ -Diagramm mit eingetragenen optischen Resonanzen

		Dipol	Quadrupol	Sextupol
Eingang	$U_{AC}$	10 kV	380 V	380 V
	$P_{nom}$	2,6 MVA	240 kVA	30 kVA
	$P_{max}$	3,6 MVA	262 kVA	33 kVA
Ausgang	$U_{DC}$	940 V	240 V	44 V
	$I_{nom}$	2,82 kA	830 A	510 A
	$I_{max}$	3,1 kA	910 A	560 A

Tabelle 1.1: Kenndaten der Netzgeräte der ELSA-Hauptmagnete

Wesentliche Anforderungen an die Stromversorgung der Hauptmagnete leiten sich aus den gewünschten Energierampen ab. Der Maximalstrom muss ausreichen, um die notwendigen Magnetfeldstärken für die Ablenkung und Fokussierung bei der größten gewünschten Teilchenenergie zu ermöglichen.

Besonders bei den Dipolmagneten ist die gewünschte kurze Anstiegs- und Abfallzeit hervorzuheben. Diese legen die Zeit fest, welche zum Hoch- und Herunterfahren der Magnetfelder benötigt wird und begrenzen so das erreichbare Tastverhältnis. Wegen der hauptsächlich induktiven Last, welche die (Elektro-) Magnete darstellen, werden zum schnellen Ändern der Magnetströme hohe Spannungen benötigt. In Phasen konstanter Magnetfeldstärke hingegen wird nur eine geringere Spannung gebraucht, da nur gegen die ohmschen Leitungswiderstände gearbeitet wird.

Bei den Netzgeräten der Dipolmagnete wurde dieser Sachverhalt berücksichtigt, indem nach dem Haupttransformator eine Aufspaltung in zwei Zweige vorgenommen wurde: Ein direkt gespeister High-Voltage-Zweig und ein Low-Voltage-Zweig, dessen Spannung nochmals heruntertransformiert wird. Die Zweige werden mittels separater Thyristor-Brücken<sup>1</sup> geregelt gleichgerichtet. Nachgeschaltet ist ein gemeinsamer, induktiv arbeitender, passiver Filter, um Wechselstromanteile zu unterdrücken. Zusätzlich kommen aktive Filter zum Einsatz, die dem primären Magnetstrom mittels induktiven Überträgern Wechselstromanteile kompensierende Ströme aufprägen. Die Steuerung zur Auswahl der Brücke (und damit der Zweige) geschieht über das sogenannte „Bridge-Select“-Signal. Einige Millisekunden vor einer Stromrampe wird auf die High-Voltage-Brücke umgeschaltet und einige Millisekunden nach der Rampenphase wieder auf die Low-Voltage-Brücke zurück gewechselt [Hol].

Die Netzgeräte der Quadrupole haben einen ähnlichen Aufbau. Der Haupttransformator definiert über eine Stern- und eine Dreieck-Anordnung der Sekundärwicklungen wieder zwei Zweige. Auch verfügen die Netzgeräte über einen ähnlichen aktiven Filtermechanismus. Bei den Sextupol-Netzgeräten werden über eine Phasenverschiebung zwei Zweige erzeugt. Hier werden keine aktiven Filter eingesetzt. Die beiden Zweige zielen auf eine möglichst rippelarme Gleichrichtung ab. Da immer beide Zweige genutzt werden, ist hier kein Bridge-Select-Signal nötig.

Die Netzgeräte der Hauptmagnete wurden von der Firma HOLEC hergestellt und werden daher häufig kurz HOLEC-Netzgeräte genannt. Zur Ansteuerung besitzen diese eine Schnittstellenkarte, die „HOLEC-InterFace Card“, welche im Folgenden mit „IFC“ abgekürzt wird. Diese erzeugt über einen DAC<sup>2</sup> eine zum Sollstrom proportionale Referenzspannung. Genaugenommen wird hier also eine digitale Referenzspannungs-Rampe benutzt, im Folgenden wird hierfür, falls keine Mehrdeutigkeiten bestehen, synonym der Begriff „Stromrampe“ bzw. einfach „Rampe“ benutzt. Die IFCs werden durch einen parallelen Bus angesteuert, welcher, um Störungen zu vermeiden, mit Optokopplern<sup>3</sup> entkoppelt wird.

### 1.4 Ansteuerung der Magnete

Die gesamte Elektronen-Stretcher-Anlage wird durch das ereignisbasierte Kontrollsystems gesteuert und kontrolliert. Dieses besteht aus mehreren Schichten:

Eine grafische Benutzer-Schnittstelle ermöglicht Zugriff auf die Beschleunigerparameter. Der Control-Layer verwaltet die Parameter und stellt über sogenannte Experten-Programme die physikalische Intelligenz des Kontrollsystem dar. Für die Hardwarekommunikation ist der Prozess-Layer, bestehend aus

---

<sup>1</sup> Ein Thyristor ist ein elektronisches Halbleiter-Bauelement. Es funktioniert im Wesentlichen wie ein Schalter, welcher mit einem Steueranschluss in den leitenden Zustand geschaltet werden kann.

<sup>2</sup> Digital-Analog-Wandler (Digital-to-Analog-Converter: DAC)

<sup>3</sup> Bei Optokopplern handelt es sich um optoelektronische Bauelemente, welche zum Übertragen eines Signals zwischen galvanisch entkoppelten Stromkreisen eingesetzt werden. Sie benutzen in der Regel eine Lumineszenz-Diode, um die Information in einen Lichtstrom umzusetzen, welcher auf der Empfangsseite mit einem Fototransistor wieder in ein elektrisches Signal zurückgewandelt wird.

mehreren VME-Rechnern<sup>4</sup>/Intel-Rechnern zuständig [Pic95]. An diese schließen sich der Fieldbus- und der Device-Layer an. Diese kommunizieren direkt oder mittels Schnittstellenschaltungen mit den anzusteuernden Geräten.

Für diese Arbeit relevant sind hier die MACSe (Microprocessor aided Accelerator Control System). Hierbei handelt es sich um eine im Jahr 1983 im physikalischen Institut entwickelte Prozessorkarte, die mit verschiedenen IO-Modulen kombiniert werden kann [Gö95]. Diese steuern auch die oben genannten IFCs der HOLEC-Netzgeräte.

Den IFCs wird der Sollwert für den Strom der Netzgeräte in digitaler Form übergeben. Um eine Stromrampe abzuspielen, also den gewünschten zeitlichen Verlauf des Sollstroms vorzugeben, wird die Stromrampe in zeitdiskrete Stützstellen zerlegt. Der IFC werden in entsprechenden Zeitabständen die Stützstellenwerte als Sollwert vorgegeben.

## 1.5 Zielsetzung dieser Arbeit

Sowohl die MACSe als auch die diese ansteuernden VME-Rechner (VM30) sind technisch überholt und haben an Zuverlässigkeit eingebüßt. Auch gestaltet es sich zunehmend problematisch Ersatzkomponenten für diese zu beziehen.

Die Zielsetzung dieser Arbeit ist die Ansteuerung der entsprechenden Netzgeräte neu aufzubauen. Gewünscht ist dabei die Kommunikation mit dem Kontrollsystem über Ethernet auszuführen, da dies eine vom Kontrollsystem und den hierfür eingesetzten Rechnern direkt unterstützte Schnittstelle ist und die zugehörige Infrastruktur bereits in weiten Teilen der Anlage zur Verfügung steht. Die Schnittstelle zu den HOLEC-Netzgeräten sollte weiterhin über die vom Hersteller der Netzgeräte eingebaute IFC realisiert werden. Auch werden einige funktionale Erweiterungen, besonders in Bezug auf die Steuerung der Rampenwiedergabe, gewünscht.

---

<sup>4</sup> VME bezeichnet primär einen Bus (Versa Module Eurocard-Bus), auf den vielfältige Prozessorkarten aufbauen.



---

# Konzeption

---

## 2.1 Strukturierung

Durch die in Abschnitt 1.5 genannten Forderungen sind die wesentlichen Rahmeneigenschaften für die weitere Entwicklung abgesteckt. Aufgrund der notwendigen Datenverarbeitungen und Steueraufgaben wurde sich für den Einsatz von Mikrocontrollern entschieden. Es wurde zunächst eine Strukturierung erstellt, welche das Problem in mehrere Hardware-Level (L4, . . . , L0) und diese verknüpfende Kommunikation-Schnittstellen (C3, . . . , C0) gliedert. Die in Abbildung 2.1 dargestellten einzelnen Level, sollen nun genauer erläutert werden.

Das Hardware-Level L4 entspricht den Rechnern des Kontrollsystems. Auf ihnen werden das Kontrollsystem selbst und die „Experten“ für physikalische Berechnungen ausgeführt. Die für die vorliegende Arbeit relevante Kommunikationsschnittstelle C3 ist Ethernet bis einschließlich der Protokollschicht UDP (siehe Abschnitt 3.2), über welches das Kontrollsystem in der Lage ist, Steuerbefehle zu versenden und Statusinformationen zu erhalten.

Die nächsten drei Hardware-Level sind einzelne Schaltungen, deren Platinen in einem gemeinsamen Gehäuse untergebracht wurden.

Die L3-Ebene ist eine Mikrocontroller-Schaltung, welche über die C3-Kommunikationsschnittstelle angeschlossen ist. Sie besitzt einen Programmcode, der unabhängig vom angeschlossenen Endgerät ist. Ihre Aufgabe ist die Filterung eingehender UDP-Pakete und gegebenenfalls die Weiterleitung der Informationen an die nächst tiefere L2-Ebene. Das L3-Modul bietet somit die Funktion eines Routers. Damit ist hier ein Netzwerkgerät gemeint, welches Netzwerkpakete zwischen Rechnernetzen, mit ggf. auch verschiedenen Protokollen/Schnittstellentypen, weiterleiten kann. Des Weiteren übernimmt es die Aufgabe der Überwachung der L2-Module. So kann es deren Status anzeigen und übermitteln. Auch kann es automatisch bei Ausfall eines L2-Moduls oder per Befehl angefordert einen Hardware-Reset des betreffenden L2-Modules durchführen. Für die C2-Schnittstelle wird das SPI (Serial Peripheral Interface) gewählt, da es eine einfache Ansteuerung mehrerer Geräte ermöglicht, ohne dass ein Busmanagement eingesetzt werden muss und damit verbundene Verzögerungen oder Paketverluste bei dem Datentransfer auftreten. Die L3-, L2- und L1-Module sind über einen Bus miteinander verknüpft. Auf diesen werden neben dem primären C2-SPI auch noch Reset- und Trigger-Signale sowie die Stromversorgung verteilt.

Die L2-Module übernehmen die softwarespezifische Ansteuerung der externen Geräte (L0). Auch bei diesen Modulen handelt es sich um Mikrocontroller-Schaltungen, welche über die C2-Schnittstelle Datenpakete empfangen und diese entsprechend ihrer, auf das Endgerät zugeschnittenen Programmierung

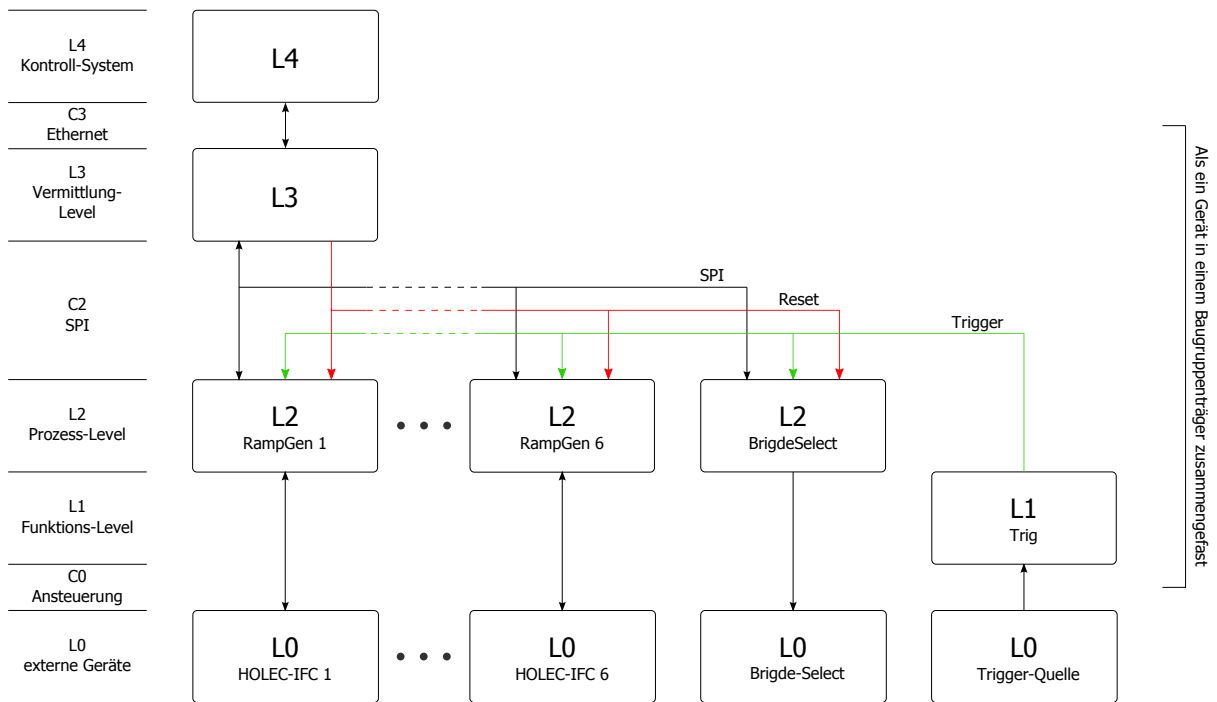


Abbildung 2.1: Strukturierung in Schaltungs- und Kommunikations-Level

abarbeiten oder regelmäßig Aufgaben ausführen. Die für die IFC-Ansteuerung programmierten Geräte können Rampenwerte berechnen, Steuerbefehle ausführen und Rückmeldungen über den Zustand an das L3-Modul senden.

Der L1-Ebene werden Schaltungen zugeordnet, die nicht programmiert, sondern nur konfiguriert werden können. Sie übernehmen die hardwarespezifische Ansteuerung der externen L0-Geräte. Sie bestehen z.B. aus Treiberschaltungen oder Konverterschaltungen. Zu Beginn der Entwicklungsphase waren in dieser Ebene Optokopplermodule (L1-Opto) zur Ansteuerung der IFC geplant, welche allerdings durch Integration in die L2-Module eingespart werden konnten. Es wurde letztlich nur ein L1-Trig-Modul gefertigt, welches Trigger-Signale an die L2-Module weiterleitet. Jedes L1-Trig-Modul besitzt zwei, mittels Optokopplern galvanisch getrennte Eingangskanäle. Sie werden über das Setzen von Steckbrücken (Jumpers) konfiguriert und besitzen eine optische Anzeige für Triggerereignisse.

In der L0-Ebene befinden sich die anzusteuern oder signalgebenden externen Geräte, wie die IFC (HOLEC-Netzgerät) und die Quellen der Trigger-Signale.

Für die technische Ausführung erwies es sich als sinnvoll die einzelnen Module der L3- bis L1-Ebene in einen 19-Zoll-Baugruppenträger zu integrieren.

## 2.2 HOLEC-Interface Card

Zu Beginn der Konzeptionierungsphase mussten zunächst Kenntnisse über die genaue Ansteuerung der HOLEC-IFC gewonnen werden. Die zur Verfügung stehende Dokumentation zu den IFC entspricht leider teilweise nicht dem aktuellen Stand, unter anderem weil Modifikationen an der Schaltung vorgenommen wurden. Eine weitere mögliche Informationsquelle stellt der Programm-Code der MAC-Prozessoren dar, welcher allerdings als Assembler-Code vorliegt und daher nur schwer lesbar ist. Daher wurde das Verhalten und die korrekte Ansteuerung aus dem modifizierten Schaltplan abgeleitet und



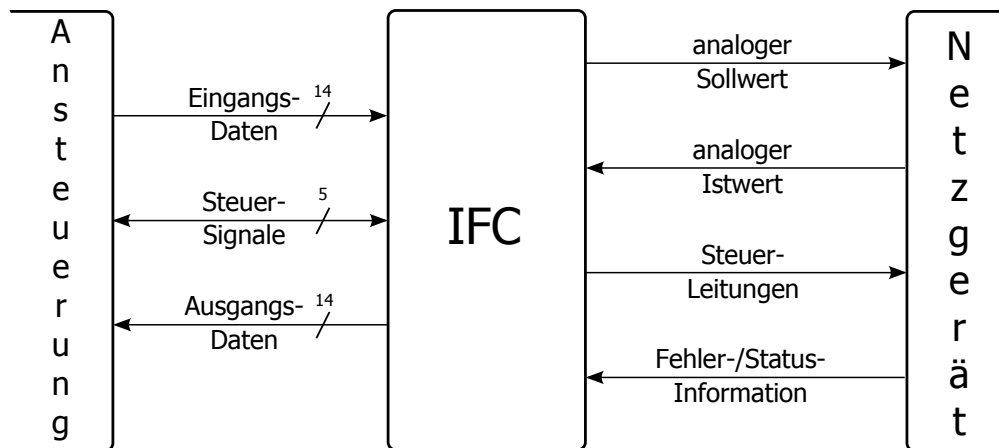


Abbildung 2.2: Ein- und ausgehende Signalleitungen der IFC

während der Entwicklung an einer nicht eingebauten IFC verifiziert.

Eine erste Übersicht über die ein- und ausgehenden Signale der IFC zeigt Abbildung 2.2. Die wesentlichen Aufgaben dieser Einheit sind die Vorgabe eines Strom-Sollwertes über einen Digital-Analog-Konverter (DAC) und das Auslesen des Strom-Istwertes mittels eines Analog-Digital-Konverters (ADC). Zusätzlich kann sie Zustandsinformationen ausgeben und Steuerbefehle weiterleiten.

Die Funktionsweise wird mit Hilfe der Abbildung 2.3 erklärt werden. Die HOLEC-IFC besitzt ein paralleles Interface, welches ursprünglich direkt auf der Platine durch Optokoppler galvanisch getrennt wurde. Da aber bereits die Ansteuerung über eine MACS-Optokopplerkarte geschieht, wurden die integrierten Optokoppler überbrückt. Die IFC hat 16 Eingangsleitungen, von denen zwei ausschließlich als Steuerleitungen dienen, so dass noch 14 Bit als Datenbreite für den DAC übrig bleiben. Von den 16 Ausgangsleitungen werden zwei für Statusindikation der IFC selbst („Interface Available“ und „ADC Valid“) benutzt. Es bleiben wiederum 14 Bit für den Datenaustausch übrig. Zur Taktung der Schaltung gibt es einen weiteren Eingang, der im Folgenden mit „Strobe“ bezeichnet wird.

Über die Dekodierung der Steuerleitungen wird festgelegt, welche Operation ausgeführt werden soll. Als Operationen stehen das Setzen einer Steuerleitung, das Setzen eines DAC-Wertes mit gleichzeitigem Starten einer ADC-Konvertierung, der NOP-Mode (No-Operation-Mode) und das Auslesen von Informationen des HOLEC-Netzgerätes zur Verfügung. Bei letzterer kann über ein weiteres Bit der Eingangsleitung ausgewählt werden, ob die Statusbits oder die Fehlerbits ausgegeben werden sollen. Der Befehlsdekoder steuert sowohl die Weiterleitung des Strobe-Pulses (S) zu den entsprechenden Schaltungsteilen als auch den Buszugriff auf den Ausgangsbus der IFC. Die eingangs genannten Modifikationen erzielen hauptsächlich einen NOP-MODE, bei dem ein Takten der Strobe-Leitung oder eine Änderung der übrigen Daten-Eingangssignale keinerlei Aktion auslösen. Dazu wurde allerdings auf eine separate Ansteuerung von DAC und ADC verzichtet.

Die Steuerung der HOLEC-Netzgeräte umfasst sieben mögliche einzelne Steuerbefehle:

- AUX ON/OFF (Hilfsbetriebe)
- MCB ON/OFF (Hauptschalter)
- RESET (der Fehlermeldungen)
- RELEASE / (SPARE 1)
- STANDBY / (SPARE 2)

Nach Anwählen der Steuer-Operation, kann durch das Anlegen eines bestimmten Bitmusters an vier

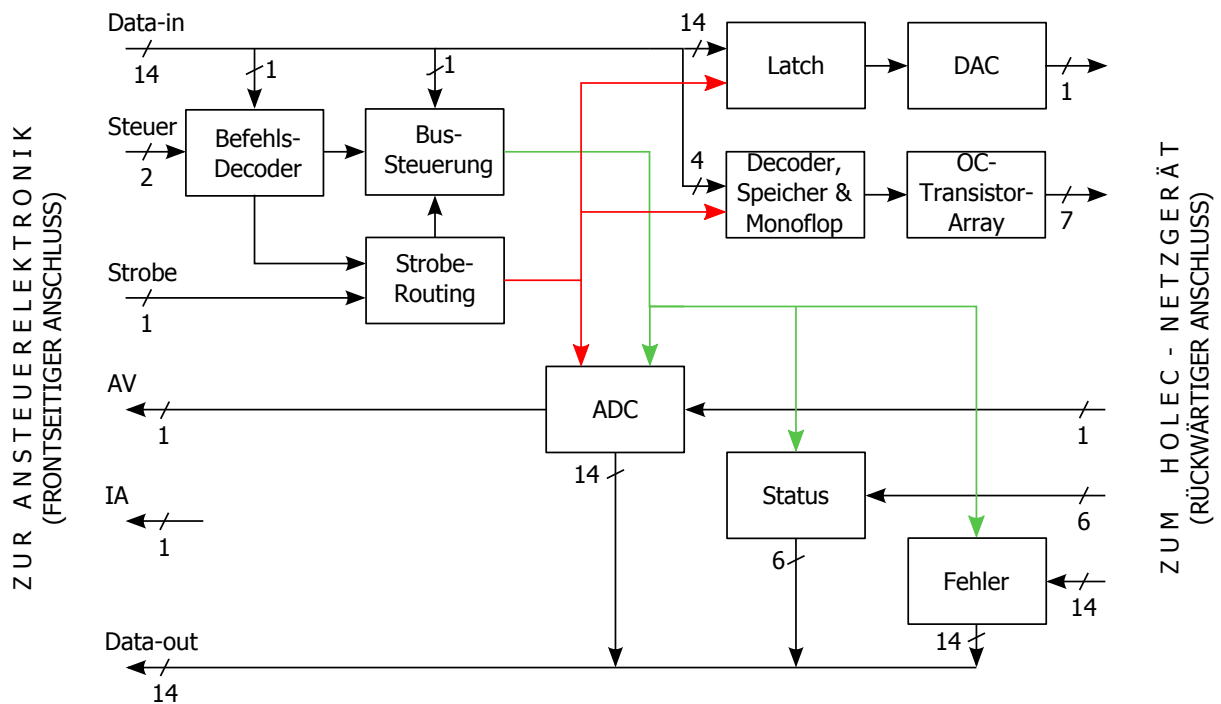


Abbildung 2.3: Blockschaltbild der Holec Interface Card (IFC)

der Dateneingangsleitungen einer der obigen Steuerbefehle ausgewählt werden. Bei einer steigenden Flanke des Strobesignals wird das (dekodierte) Bitmuster gespeichert und ein Monoflop<sup>1</sup> getriggert, welcher die gewählte Steuerleitung für ca. 100 ms über einen Transistor mit offenem Kollektor aussteuert. Die Schaltung stellt sicher, dass immer nur eine Steuerleitung aktiv ist. Weitere, während der Aussteuerzeit eingehende Steuerbefehle werden daher ignoriert.

Eine weitere Operation ermöglicht das Setzen des DACs und das Auslesen des ADCs. Die zugehörigen analogen Spannungen entsprechen den Soll- und Ist-Wertvorgaben des Stroms der Magnete. Diese Operationen sind leider nur gekoppelt möglich. Die Ausführung beginnt wiederum mit dem Anwählen der gewünschten Operation über die Steuerleitungen und Anlegen des gewünschten DAC-Wertes auf den Eingangsbus. Setzen der Strobeleitung auf low<sup>2</sup> schaltet das dem DAC vorgeschaltete Latch auf transparent, d.h. der Eingangs-Zustand wird direkt auf den Ausgang des Latches durchgeschaltet. Bei dem eingesetzten DAC (Type DAC72) handelt es sich um einen ungepufferten Parallel-Umsetzer, der keine Steuerung besitzt. Bei der steigenden Flanke des Strobesignals speichert das Latch den angelegten DAC-Wert. Gleichzeitig erhält der ADC (Type DAS1152) einen Trigger, der ein integriertes Sample-and-Hold-Glied<sup>3</sup> aktiviert und eine neue AD-Konvertierung startet. Das Ende einer AD-Umsetzung, welche typischerweise 35  $\mu$ s dauert, wird mit der Statusleitung „ADC Valid“ (AV) signalisiert. Ist dieses Signal high, liegt das Ergebnis der AD-Konvertierung auf den Ausgangsleitungen an. Wurde eine AD-Umsetzung gestartet, kann sie nicht unterbrochen oder neu gestartet werden. Allerdings führt ein weiterer Trigger-Puls zu einem Neusetzen des Sample-and-Hold-Gliedes, dies führt dann zu einem falschen Ergebnis der Konvertierung. Durch die Verknüpfung von DAC- und ADC-Operation wird die Auslese

<sup>1</sup> Elektronisches Schaltungselement, welches triggeregesteuert Pulse bestimmter Zeitdauer generieren kann.

<sup>2</sup> Mit „low“ und „high“ werden hier die technischen Realisierungen der Binärzustände bezeichnet:  $0 \hat{=} \text{low} \hat{=} L \hat{=} U < 1,5 \text{ V}$  und  $1 \hat{=} \text{high} \hat{=} H \hat{=} U > 3,5 \text{ V}$ . Die Spannungsniveaus sind durch die verwendete 5 V-CMOS-Logik gegeben.

<sup>3</sup> Eine Schaltung, die einen analogen (Spannungs-)Wert speichern kann.

von Ist-Werten in Phasen häufiger DAC-Werteänderung stark erschwert bzw. unmöglich gemacht.

Weiterhin bietet die HOLEC-IFC die Möglichkeit, Fehler und Statusinformationen auszulesen. Hierzu muss wieder mittels der Steuerleitungen die Auslese-Operation ausgewählt werden und darüber hinaus mit einer weiteren Eingangsleitung festgelegt werden, ob das Status-Wort<sup>4</sup> oder das Fehler-Wort ausgelesen werden soll. Bei einer fallenden Flanke des Strobe-Signals werden die Informationen auf die Ausgangsleitungen der IFC gelegt. Eine aus dem Schaltplan erstellte Logiktablette ist in Tabelle A.1 zu sehen.

## 2.3 Verwendeter Mikrocontroller und Werkzeuge

Ausgehend von den Überlegungen, welche Schnittstellen zum Einsatz kommen sollen und welche rechen- und steuerungstechnischen Aufgaben in den L3- und L2-Modulen ausgeführt werden sollen, ergaben sich mehrere Kriterien zur Auswahl eines geeigneten Mikrocontrollers. Eingesetzt wird der auf einer 32-Bit RISC-AVR-Architektur basierte AT32UC3A1256 der Firma Atmel [Atm12]. Abbildung 2.4 zeigt das Blockdiagramm des Mikrocontrollers. Er wird in den Schaltungen der L2- und L3-Module eingesetzt und weist folgende, für das Projekt relevante Eigenschaften auf:

- 32-Bit Datenbreite und eine Rechenleistung von 91 DMIPS @ 66 MHz
- integrierte 64 KB high-speed SRAM und 256 KB Flash-Speicher
- Ethernet MAC 10/100 Mbps mit RMII und DMA
- Multi-Slave Serial Peripheral Interface
- Peripheral DMA Channels
- Interrupt Controller und External Interrupt Controller
- Quarz-Treiber, PLL, Echtzeit- und Watchdog-Timer
- drei 16-Bit Timer/Counter-Kanäle
- JTAG-Schnittstelle (Debugging)
- 3,3 V Versorgungsspannung bei 5 V toleranten I/Os
- 100-Pin TQFP-Gehäuse mit 69 General Purpose Input/Output-Pins

Sowohl die Programme der Mikrocontroller (Firmware) als auch die Testprogramme, welche auf normalen Computern liefen, wurden alle in der Sprache C (C99-Standard) geschrieben. Als Entwicklungsumgebung wurde das von der Herstellerfirma bereitgestellte „Atmel Studio 6“ benutzt. Dieses bietet die Möglichkeiten der Projektverwaltung, des Hinzufügens von vorgefertigten Software-Modulen und, über den benutzten Debugger, des Programmierens des Mikrocontrollers und der Fehlersuche. Als Debugger wurde der JTAGICE3 benutzt. Mit einem JTAG-Interface können integrierte Schaltungen (IC), wie hier der Mikrocontroller, direkt in der Schaltung programmiert werden. Auch kann durch Einflussname auf den Programmablauf (mittels sogenannter Breakpoints) nach Fehlern im Programm gesucht werden. Leider erwies sich der Debugger nicht als ausreichendes Mittel zur Fehlersuche. Zum einen waren nicht alle Möglichkeiten, welche der Mikrocontroller bezüglich des Debuggings bot, implementiert. Zum anderen lässt sich der Debugger nur praktisch nutzen, wenn der Compiler<sup>5</sup> keine Optimierung des Codes vornimmt. Die Optimierung kann nicht deaktiviert werden, weil dann einige benutzte Treiber des Mikrocontroller-Herstellers nicht mehr korrekt funktionieren.

Als nützliche Hilfsmittel zum Testen und zur Fehlersuche bei der Mikrocontroller-Programmierung, erwiesen sich, neben einfachen LEDs, LCD-Anzeigen zur Textdarstellung und, besonders bei zeitkriti-

---

<sup>4</sup> Mit Wort ist hier die Gesamtheit aller Status-/Fehler-Bits gemeint

<sup>5</sup> Ein Programm (hier gcc), welches einen Programmcode (hier der in C geschriebene Code) in eine von dem gewünschten Prozessor ausführbare Form übersetzt.



schen Effekten (Kommunikation), der Einsatz eines Oszilloskops.

## 2.4 Sprachkonstrukte der Sprache C

Da im Folgenden nicht gänzlich auf Bezugnahmen zum Programmcode verzichtet werden kann, sollen nun die relevanten Konstrukte der Sprache C (C99-Standard) erklärt werden.

Die häufigsten Datentypen sind Ganzzahlen (integer) mit verschiedenem Bitumfang. Die vorzeichenlosen werden mit `uint8_t`, `uint16_t` und `uint32_t`, die Vorzeichenbehafteten mit `int8_t`, `int16_t` und `int32_t` bezeichnet. Typennamen werden konventionsgemäß meist mit einem „\_t“ am Namensende kenntlich gemacht. Nur bei der Berechnung der Werte für die Stromrampe werden Gleitkommazahlen (floating point number) mit der Typenbezeichnung „float“ eingesetzt. Da es sich als günstig erwiesen hat, wird für die Soll- und Istwerte der Datentyp `uint16_t` mit der Interpretation als U14Q2 benutzt<sup>6</sup>. D.h. die beiden niedrigwertigsten Bits werden als Nachkommastellen behandelt. Es handelt sich hierbei also um keinen neuen Datentyp.

Um Daten zu strukturieren, werden „structs“ (Daten-Verbünde) benutzt. Ein struct besteht aus einer Anordnung von einem oder mehreren Datentypen mit fester Reihenfolge. Structs können ebenfalls wieder einen Datentyp definieren. Die Definition eines neuen Datentyps wird mit dem Schlüsselwort „typedef“ eingeleitet.

Unter einem Array, wird eine Liste von Variablen eines einheitlichen Datentyps verstanden. Auf die Einträge der Liste kann mittels eines Indexes zugegriffen werden. Ein Array kann also als ein Vektor aufgefasst werden.

Die Variablen werden in dem Speicher des Mikrocontrollers abgelegt. Eine elegante Methode, um auf Speicherinhalte zuzugreifen, ist die Benutzung von Zeigern (pointer). Zeiger können als eine Variable aufgefasst werden, deren Wert der Speicheradresse einer (anderen) Variable von spezifischem Datentyp entspricht.

---

<sup>6</sup> U14Q2 entspricht einer vorzeichenlosen Festkommazahl mit 14 Bitstellen vor dem Komma und zwei hinter dem Komma



---

# Schaltungen und Firmware

---

Für die in Abschnitt 2.1 vorgestellte und in Abbildung 2.1 skizzierte Strukturierung, sollen im Folgenden die entwickelten Hardware-Level bezüglich Funktion und Realisierung beschrieben werden. Der Aufbau der Mikrocontroller-Programme (Firmware) und die benutzten Kommunikationsschnittstellen mit den zugehörigen Protokollen werden vorgestellt.

### 3.1 L4 – ELSA-Kontrollsystem

Die L4-Ebene wird durch das ELSA-Kontrollsystem dargestellt. Aus der Festlegung des Beschleunigermodus, der gewünschten Energie der Elektronen und des Timings berechnet das Kontrollsystem den zeitlichen Verlauf der Energie über einen Zyklus. Hieraus werden weiter die Magnetstärken der Dipol-, Quadrupol- und Sextupol-Magnete und die zugehörigen Stromrampen berechnet. Die Berechnungen berücksichtigen auch Sättigungseffekte der Magnete. Darüber hinaus kann das Kontrollsystem Steuerbefehle an die IFCs schicken und den Status der Netzgeräte anzeigen.

Für die Umstellung auf die neue Ansteuerung waren einige Anpassungen an dem Kontrollsystem nötig. So wurden die Routinen, welche aus der Vorgabe des Beschleuniger-Modus, des Timings und des Energieverlaufs die Stromrampen der Magnete berechnen, neu geschrieben. Auch die neu hinzugekommenen Steuermöglichkeiten der L2-Module mussten eingebunden und die geänderte Kommunikationstopologie musste berücksichtigt werden [Pro].

### 3.2 C3 – Ethernet-Protokoll

Ethernet spezifiziert ein Datennetz über die Festlegung von Protokollen auf der Softwareseite sowie der geforderten Hardwareeigenschaften der Übertragungselektronik. Als Ethernet-Medientyp wird das weitverbreitete, kabelgebundene 10BASE-T/100BASE-TX (IEEE 802.3), mit Übertragungsraten von 10 MBit/s bzw. 100 MBit/s benutzt. Diese Schnittstelle ist voll-duplexfähig (erlaubt gleichzeitigen Datenaustausch in beide Richtungen), asynchron und nutzt eine Basisbandübertragung. Sie ist eine der meistverwendeten Techniken für den paketorientierten Datenaustausch.

Das Ethernet-Protokoll legt den Aufbau eines Datenpakets (Ethernet-Paket), also eine bestimmte Abfolge von zu übermittelnden Informationen, fest (siehe Abbildung 3.1). Die Wichtigsten hiervon sind die Ziel- und Quell-MAC-Adresse<sup>1</sup>, ein Typenfeld, der Nutzdatenbereich und eine Prüfsumme (mit CRC

---

<sup>1</sup> MAC: Media Access Control; nicht zu verwechseln mit MACS

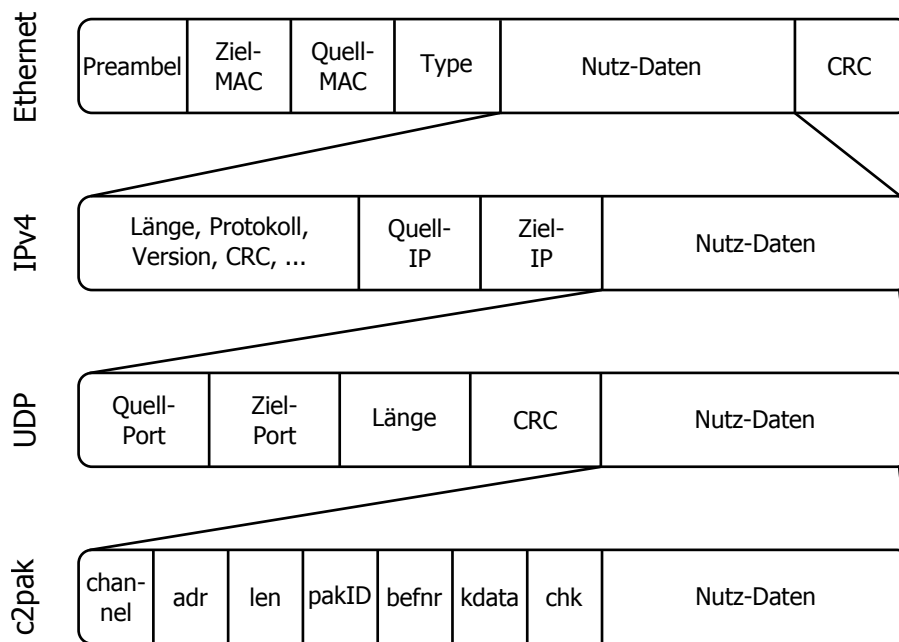


Abbildung 3.1: Aufbau der benutzten Protokollschichten

abgekürzt). Die MAC-Adresse stellt die Hardware-Adresse eines Netzwerkkartens (Netzwerkkarte) dar und wird für das Routing innerhalb des Datennetzes genutzt. Das Typenfeld gibt den Protokoll-Typ der nächsten Schicht an (s.u.), welches in den Nutzdatenbereich des Ethernetpakets eingebettet wird [Int]. Die Prüfsumme ermöglicht das Feststellen von Bitfehlern, die bei der Übertragung entstanden sein können.

Üblicherweise werden beim Datentransfer mehrere Protokolle ineinander verschachtelt, welche jeweils verschiedene Aufgaben der Netzwerkkommunikation übernehmen. So deckt das oben dargestellte Ethernet-Protokoll die physikalische Bitübertragung, Regelung des Zugriffs auf das Übertragungsmedium und die Fehlererkennung ab. Darauf aufbauend wird das IPv4-Protokoll benutzt, um die Vermittlung, also das korrekte Weiterleiten der Datenpakete an Verbindungszwischenstellen, zu realisieren. Das hierauf aufsetzende User Datagram Protocol (UDP) stellt die Zuordnung der versendeten Daten zu einer bestimmten Anwendung mittels sogenannter Portnummern sicher. Alle diese Protokolle sind transportorientiert, hierauf folgt eine letzte anwendungsorientierte Protokollschicht: das c2pak-Protokoll. Dieses wurde selbst entwickelt und konnte so optimal an die gegebene Problemstellung angepasst werden. Eine genauere Beschreibung hierzu findet sich in Abschnitt 3.3.5. Diese Protokollschicht ist für die Ethernet-Kommunikation nicht relevant und entspricht für diese nur den Nutzdaten.



### 3.3 L3 – Kommunikations-Modul

#### 3.3.1 Übersicht

Der Hardware-Level L3 hat primär die Aufgabe, die Kommunikation zwischen dem Kontrollsystem (L4) und den eigentlichen Rampen-Generatoren (L2) zu vermitteln (Router-Prinzip). Sie detektiert ob ein L2-Modul angeschlossen ist, ob dieses korrekt kommuniziert und führt Resetoperationen aus. Zunächst soll ein Überblick über das Modul anhand von Abbildung 3.2 gegeben werden, bevor näher auf die Programmierung und die Details der Schaltung eingegangen wird.

Kernstück der Schaltung ist der Mikrocontroller, der mittels eines weiteren ICs, des sogenannten PHY, über Ethernet mit der L4-Ebene kommunizieren kann (näheres s.u.). Für die C2-Schnittstelle wird das interne SPI-Modul des Mikrocontrollers genutzt. Dieses Modul bietet multiple Adressierungs-Anschlüsse, wodurch über einen CS-Decoder (CS: Chip Select) insgesamt acht L2-Module adressiert werden können. Die Daten- und Takt-Signale werden über Treiber-ICs mit differentiellen Aus-/Eingängen auf den Bus gelegt. Der Zustand jeder der acht C2-Kanäle wird über eine dreifarbige LED angezeigt. Des Weiteren gibt es eine L3-Aktivitätsanzeige, welche signalisiert, dass das Modul ordnungsgemäß arbeitet. Ein Reset kann mittels Taster, Watchdog<sup>2</sup> oder C3-Befehl durch eine zusätzliche Schaltung ausgelöst werden. Für die Programmierung ist eine JTAG-Schnittstelle vorhanden. Die Stromversorgung geschieht über einen 3,3 V-Spannungsregler.

Ein detaillierter Schaltplan des L3-Moduls befindet sich im Anhang in den Abbildungen B.3 bis B.5.

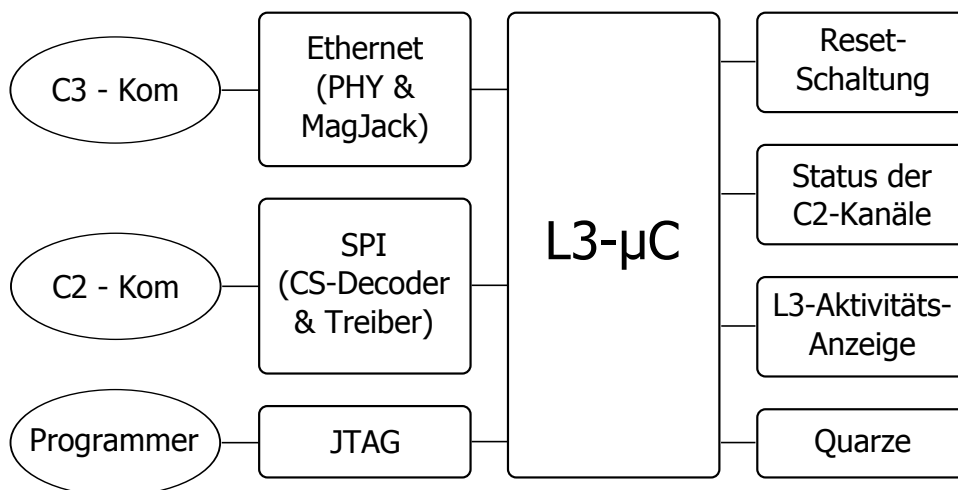


Abbildung 3.2: Blockschaltbild des L3-Moduls

#### 3.3.2 C3 – Ethernet-Schaltungsteil

Der verwendete Mikrocontroller besitzt für die Ethernet-Kommunikation ein sogenanntes „Ethernet MACB“-Modul. Mit MAC (Media Access Control) wird eine Steuerungseinheit für den Zugriff auf das verwendete Medium bezeichnet. Das MACB-Modul kann über Register-Einträge gesteuert und konfiguriert werden. Die zu übermittelnden Daten der Anwendungssoftware werden aus dem RAM des

<sup>2</sup> Hierbei handelt es sich um ein Mikrocontroller-Modul, welches ein Aufhängen der Programmausführung detektieren kann und dann ggf. einen Reset ausführt. Es basiert auf einem Timer-Modul, welches regelmäßig durch den Programm-Code zurückgesetzt werden muss

Mikrocontrollern über ein DMA<sup>3</sup>-Interface zum MACB-Modul übertragen, wo sie in FIFOs<sup>4</sup> gepuffert werden. Das MACB-Modul füllt ein Ethernet-Paket mit den Daten bzw. extrahiert diese aus einem solchen. Die empfangenen Pakete werden nach der MAC-Adresse gefiltert und die Checksumme wird überprüft, wobei alle ungültigen Pakete direkt verworfen werden. Umgekehrt wird für alle zu sendenden Pakete die zugehörige MAC-Adresse eingetragen und die Checksumme berechnet.

An das MACB-Modul ist ein spezieller integrierter Schaltkreis angeschlossen, der das physikalische Übertragungsmedium festlegt und deswegen kurz als „PHY“ bezeichnet wird. Der hier verwendete IC DP83848 [Nat08] tauscht über ein RMII (Reduced Media Independent Interface) Daten mit dem MACB-Modul des Mikrocontrollers aus und kann mittels einer weiteren seriellen Schnittstelle (MDIO) konfiguriert werden. Die zu sendenden Daten werden in ein analoges moduliertes Signal kodiert bzw. die empfangenen aus diesen heraus enkodiert. Die Kodierung stellt sicher, dass eine Taktrückgewinnung aus dem asynchron übertragenen Signal möglich ist und dieses einen möglichst geringen Gleichstromanteil aufweist. Der PHY besitzt einen differentiellen Eingang für das „Receive“-Signal (RX) und einen differentiellen Ausgang für das „Transmit“-Signal (TX). Diese Anschlüsse führen zu einem induktiven Überträger. Durch diesen wird eine galvanische Entkopplung<sup>5</sup> erzielt. Auch alle Gleichtaktstörungen werden so effektiv eliminiert. Der Überträger speist das Signal in das eigentliche Übertragungsmedium, bestehend aus zwei verdrehten Aderpaaren (sog. twisted-pair) eines „LAN-Kabels“. In dem hier beschriebenen Aufbau wird ein Mag-Jack (ST2) verwendet – eine Kombination aus Buchse, magnetischen Überträger und LEDs für Link- und Aktivitäts-Anzeige in einem. Für den Datentransfer über das RMII als auch zur De-/Enkodierung der Signale wird ein frequenzstabiler 50 MHz-Taktgenerator benötigt. Für diese Aufgabe wird ein integrierter Quarzoszillator verwendet.

#### 3.3.3 C2 – SPI-Schaltungsteil

SPI ist ein einfaches, protokollloses, serielles Bussystem, das auf dem Prinzip von Schieberegistern basiert. Es zählt zu den synchronen „Single-Master-Multi-Slave“-Schnittstellen. Der Bus umfasst zwei Datenleitungen (MOSI  $\hat{=}$  Master-Output-Slave-Input und MISO  $\hat{=}$  Master-Input-Slave-Output), einer Taktleitung (SCLK  $\hat{=}$  Slave-Clock) und einer oder mehrerer Steuerleitungen (SS  $\hat{=}$  Slave-Select oder CS  $\hat{=}$  Chip-Select), die jeweils genau einen Slave-Busteilnehmer aktiv schalten. Es gibt mehrere mögliche Bustopologien, von denen die hier benutzte mit separaten CS-Leitungen in Abbildung 3.3 skizziert ist. Der Master beginnt die Kommunikation, indem er mittels der CS-Leitung einen Slave auswählt. Synchron mit der Taktung der SCLK-Leitung werden die Daten sowohl von Master zu Slave als auch umgekehrt transferiert (zwangsvoll-duplex). Der Slave hat also keinerlei steuernden Einfluss auf die Kommunikation. Das L3-Modul stellt den SPI-Master dar. Das SPI-Modul des Mikrocontrollers besitzt die Fähigkeit einen externen Encoder anzusteuern, um somit eine größere Anzahl von SPI-Slaves und damit L2-Modulen anzusteuern. Um eine störsichere Kommunikation sicher zustellen werden die MOSI-, MISO- und SCLK-Signale über differentielle<sup>6</sup> Treiber in die Busleitung eingespeist [TSG10].

#### 3.3.4 Weitere Schaltungsteile

Um eine übersichtliche Indikation des Kommunikationsstatus von allen über die C2-Schnittstelle verbundenen L2-Modulen zu erhalten, wird jeder C2-Kanal mit einer Duo-LED dargestellt. Es ergeben sich vier Anzeigemodi:

---

<sup>3</sup> Datentransfer welcher nahezu ohne Belastung der CPU abläuft.

<sup>4</sup> FIFO: First In – First Out; Pufferspeicher der nach dem Prinzip einer Warteschlange arbeitet

<sup>5</sup> dies ist nur bei ungeschirmten Steckern sicher gestellt

<sup>6</sup> Differentielle bzw. Symmetrische Signalübertragung nutzt die (Spannungs-)Differenz zweier komplementär ausgesteuerter Signalleitungen, hierdurch werden u.a. alle Gleichtaktstörungen unterdrückt.

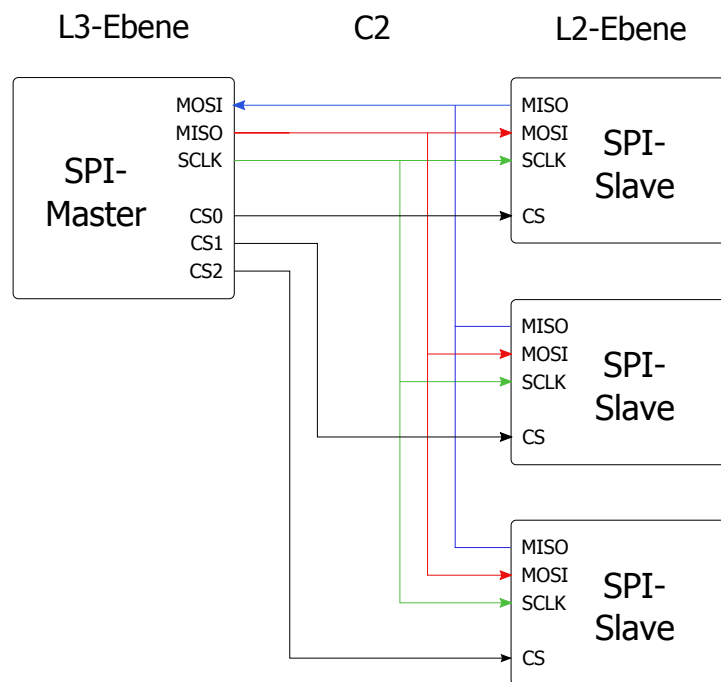


Abbildung 3.3: Benutzte Topologie der C2-Schnittstelle

- aus  $\hat{=}$  kein Modul angeschlossen
- grün  $\hat{=}$  ein korrekt kommunizierendes Modul ist angeschlossen
- gelb  $\hat{=}$  Nutzdaten werden transferiert
- rot  $\hat{=}$  eine Störung der Kommunikation liegt vor

Mit einem Monoflop-IC wird eine Aktivitäts-Statusanzeige erzeugt. Ein weiteres Monoflop kann einen, das L3-Modul resettenden Puls generieren. Die Schaltung kann des Weiteren auch über einen Taster zurückgesetzt werden. Für den Reset eines L2-Moduls wird ein Signal-Puls über einen freien Ausgang des SPI-Treibers auf die Busleitung gegeben, die Selektion des Gerätes geschieht über die CS-Leitungen der C2-SPI-Schnittstelle.

Ein Linear-Spannungsregler mit nachgeschaltetem Filter stellt eine störungsfreie Versorgungsspannung sicher.

### 3.3.5 C2 – C2pak-Protokoll

Das C2-Protokoll legt die bei der C2-Kommunikation anzuwendende Daten- und Informationsstruktur fest. Durch die Definition eines Paketheaders (eine „struct“-Deklaration) wird so das Routing in der L3-Ebene und die Zuordnung der auszuführenden Operation in der L2-Ebene („Befehl“ genannt) ermöglicht. Der Aufbau des C2-Paketheaders ist in Tabelle 3.1 und Abbildung 3.1 zu sehen. Die Gesamtlänge eines Paketes ist auf die, für die Mikrocontroller günstigen Vielfache von 32 Bit festgelegt. Auch der Paketheader wird in Blöcke dieser Größe unterteilt.

Das erste Byte des ersten Headerblocks beinhaltet die Angabe des sogenannten Channels, dies ermöglicht den L3-Modulen die Unterscheidung, welche der ihnen zur Verfügung stehenden Schnittstellen sie zur Weiterleitung der Daten benutzen sollen. Aktuell sind folgende Channels definiert: der C3-Channel für UDP über Ethernet, C2-Channel für SPI sowie ein L3-Channel zur Ausführung von Befehlen auf dem L3-Modul selbst. Das zweite Byte (adr) beinhaltet eine Unteradressierung zur Unterscheidung der

1.Byte	2.Byte	3.Byte	4.Byte	
channel (uint8_t)	adr (uint8_t)	len (uint16_t)		(1. Block)
pakID (uint16_t)		befnr (uint8_t)	kdata (uint8_t)	(2. Block)
chk (uint32_t)				(3. Block)
data (uint8_t)	...	...	...	(4. Block ...)

Tabelle 3.1: Struct-Aufbau des c2pak-Headers

eingebauten L2-Module. Da die übermittelten Pakete unterschiedliche Befehle beinhalten, ist ihre Länge nicht fix, sondern variiert deutlich. Somit ist es nötig die Information über die Gesamtlänge des Datenpaketes (in Byte) anzugeben, welche in dem 3. und 4. Byte (len) hinterlegt wird.

Der zweite Block beginnt mit einer ID-Nummer (pakID), welche vom Kontrollsystem generiert wird und zur Zuordnung von Quittierungs- und Antwort-Paketen benutzt wird. Mit der sich daran anschließenden ein Byte langen Befehlsnummer (befnr), wird festgelegt, welcher Befehl ausgeführt werden soll. Das Kurzdaten-Feld (kdata) wird befehlsabhängig mit Steuerdaten befüllt.

Der dritte 32-Bit-Block ist eine Prüfsumme des Paketes (chk). Mit ihrer Hilfe kann der Empfänger die Fehlerfreiheit des Paketes überprüfen. Sie berechnet sich aus dem negierten bitweisen Exklusiv-Oder der Paketdaten, unter Auslassung des Prüfsummenfeldes selbst.

Auf den Paketheader folgen gegebenenfalls die Nutzdaten mit variabler Länge.

### 3.3.6 L3 – Programmierung

Ein Überblick der L3-Programmierung ist als Flussdiagramm in Abbildung 3.4a zu sehen. Nach der Initialisierung des Mikrocontrollers und der benutzten integrierten Peripherie-Module, werden die C2- und C3-Schnittstellen initialisiert. In der sich daran anschließenden Hauptschleife wird geprüft, ob die L4- oder L2-Ebenen Datenpakete übermittelt haben, welche dann gegebenenfalls einer Subroutine übergeben werden, die das eigentliche Routing bewerkstelligt. Weitere Subroutinen aktualisieren regelmäßig die LED-Anzeige und setzen den Watchdog zurück.

Für die Implementierung des eigentlichen Ethernet-, IP- und UDP-Protokolls, wurde der freie IP-Stack „lwip“ (lightweight TCP/IP) eines Drittanbieters genutzt, der auf die reduzierten Ressourcen eines Mikrocontrollers optimiert ist. Der Mikrocontroller-Hersteller stellt einen Softwaretreiber bereit, welcher das Bindeglied zwischen dem MAC-Modul, einem externen PHY (Hardware) und dem „lwip“ IP-Stack (Anwendungs-Software) darstellt. Hierauf aufbauend wurden Funktionen zum Empfangen und Versenden von UDP-Paketen geschrieben (udp\_receive und udp\_send).

Die Benutzung von SPI bringt einige Besonderheiten mit sich, die bei der Programmierung der C2-Routine berücksichtigt werden mussten. Besonders relevant hierbei war die Eigenschaft des Zwangs-Vollduplex und die Tatsache, dass L2-Module (SPI-Slaves) keine direkte Möglichkeit haben, dem L3-Modul (SPI-Master) mitzuteilen, dass sie zu verschickende Daten haben. Dies führte zu der Notwendigkeit, dass das L3-Modul regelmäßig ein spezielles Paket (onlyread-pak) an die L2-Ebene schickt, welches diesem sagt, dass die Kommunikation funktioniert und gleichzeitig über die Eigenschaft des Zwangs-Vollduplex ein Paketanfang an das L3-Modul zurück transferiert. Ist dies ebenfalls ein onlyread-pak, weiß das L3-Modul seinerseits, dass die Kommunikation funktioniert und das L2-Modul bereit ist. Entspricht der Paketheader nicht dem eines onlyread-Paketes, werden entsprechend dem „len“-Feld des

Headers weiter Datenbytes gelesen und das so erhaltene Paket anschließend weiterverarbeitet.

Eine der Kern-Subroutinen ist diejenige zum Routen von Paketen, deren Flussdiagramm in Abbildung 3.4b gezeigt ist. Falls der Channel auf UDP gesetzt ist, werden die Daten an die Funktion `udp_send` übergeben. Komplizierter gestaltet sich der Ablauf für den C2-Channel, denn beim Übermitteln der Daten an die L2-Ebene kann gleichzeitig ein Paket empfangen werden. Ist dies der Fall, wird die Route-Subroutine nochmals aufgerufen. Lautet der Channel-Wert L3, wird das Paket einer weiteren Funktion übergeben, welche abhängig von der Befehlsnummer eine Reset-Operation oder eine Ping-Funktion ausführt. Vor Verlassen der Subroutine wird der Speicherbereich des verarbeiteten Pakets wieder freigegeben. Mit dieser Routing-Funktion ist es somit möglich Daten zwischen beliebigen Ebenen (L4, ..., L2) auszutauschen, wobei auch der der Datentransfer zwischen verschiedenen L2-Modulen möglich ist.

### 3.3.7 L3 – Platinen-Layout

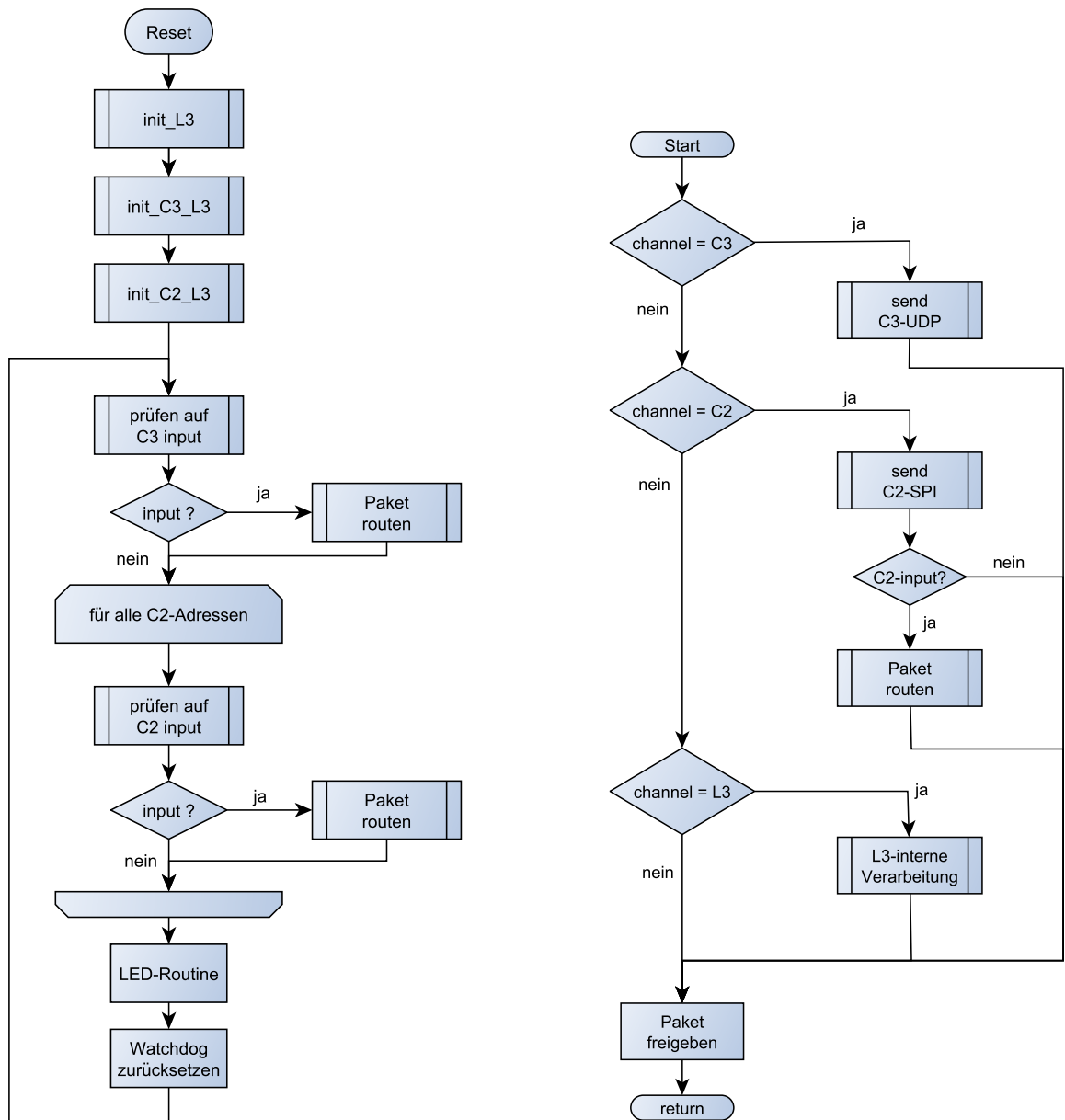
Für die Fertigung einer Platine für die entwickelte Schaltung, wird ein sogenanntes Platinen-Layout erstellt. Dabei handelt es sich um die Festlegung der geometrischen Position von Bauteilen und den Verlauf der diese verbindenden Leiterflächen auf der Leiterplatte. Bei dem Platinen-Layout wurde durch die Entscheidung, die Module in einem 3HE-Baugruppenträger<sup>7</sup> unterzubringen, eine Größenvorgabe einer „EURO“-Platine (160 mm × 100 mm) gemacht. Deren Fläche ist ausreichend groß, so dass die Zahl der Kupfer-Leiterbahnebenen (Layer) auf zwei beschränkt werden konnte. Der obere Layer ist primär für die Signalleitung vorgesehen, wohingegen der untere Layer flächendeckend als Massenfläche genutzt wird und nur kurze Kreuzungs-Verbindungen auf diesem verlegt sind. Alle Bauelemente der Frontplatte, konnten auf einer Breite von 5 TE<sup>8</sup> untergebracht werden und so direkt auf der Platine aufgelötet werden. Der rückwärtige Anschluss für die Verbindung der Module untereinander wurde mittels der für Baugruppenträger üblichen vielpoligen Steckverbinder nach DIN 41612 realisiert.

Für das Erstellen des Platinenplanes wurde der Layout-Editor „EAGLE“ benutzt. Mit diesem kann ein Schaltplan erstellt und dabei den Bauelementen neben elektrischen Werten auch Gehäuse-Parameter zugeordnet werden. In dem Layout-Editor werden dann die Footprints (Darstellungen der Kupferflächen zum Auflöten der Bauelemente) und symbolisch die notwendigen Verbindungen gezeigt. Durch Positionieren der Footprints und Verlegen der Leiterbahnen wurde das in Abbildung 3.5 gezeigte Layout entwickelt. In rot sind die Kupferflächen der Oberseite und in blau die der Unterseite dargestellt.

---

<sup>7</sup> 1 HE = 44,45 mm (Höheneinheit)

<sup>8</sup> 1 TE = 5,08 mm (Teilungseinheit)



(a) Flussdiagramm der L3-mainroutine

(b) Flussdiagramm der Funktion zum Routen von C2pak-Paketen

Abbildung 3.4: Flussdiagramme zum Programm der L3-Module

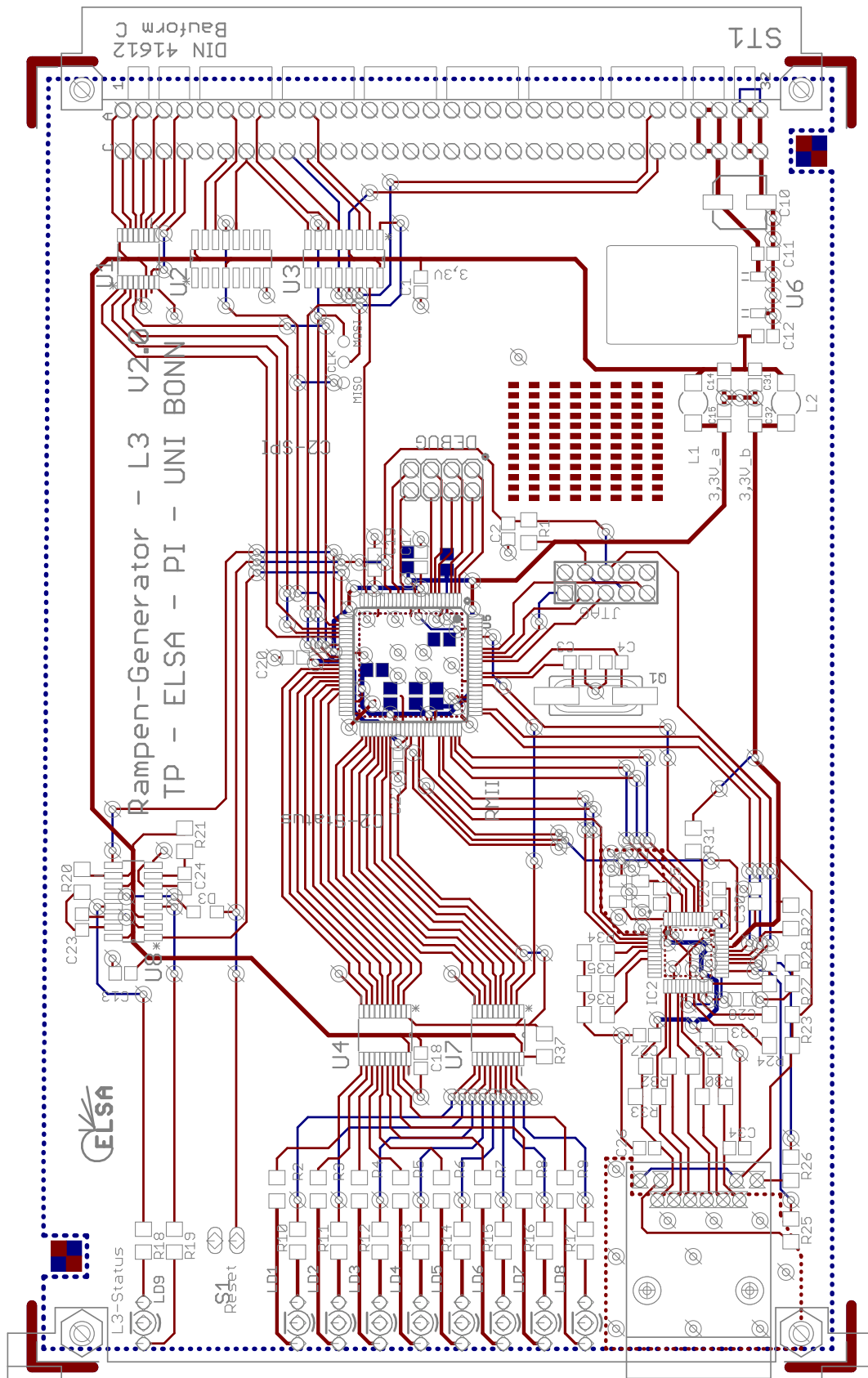


Abbildung 3.5: Platinen-Layout des L3-Moduls

## 3.4 L2 – Prozess-Modul

### 3.4.1 Übersicht

In der L2-Ebene findet die eigentliche Verarbeitung der Pakete statt. Der Mikrocontroller der L2-Ebene wird spezifisch auf das anzusteuernende Gerät (L0) und die auszuführenden Aufgaben programmiert. Im Rahmen dieser Arbeit wurden zwei leicht verschiedene L2-Module gefertigt: eine Standardvariante zur Ansteuerung der HOLEC-IFC und eine abgewandelte Ausführung zur Generierung eines Bridge-Select-Signals.

In diesem Abschnitt soll zunächst nur die Standardvariante beschrieben werden und zum Schluss auf die Abweichungen der anderen Variante eingegangen werden. Wie auch bei der L3-Ebene ist das Kernstück der L2-Schaltung ein Mikrocontroller vom Typ AT32U3A1256. Vorhanden ist eine SPI-Schnittstelle für C2-Kommunikation. Die Anbindung zur nächst tieferen Ebene findet über eine Reihe von Optokopplern statt. Zur Statusindikation werden mehrere LEDs eingesetzt.

### 3.4.2 SPI-Schaltungsteil

Die Signale des SPI werden wiederum über einen differentiellen Treiber zwischen Bus und Mikrocontroller ausgetauscht. Da die L2-Ebene den SPI-Slave darstellt, muss ihm eine Adresse (eine der CS-Leitungen) zugeordnet werden. Dazu dient ein Kodierschalter (8-poliger DIP-Schalter) auf der Platine, mit dem einer der acht Chip-Select-Signale, welche auf dem Bus liegen, ausgewählt wird. Jedes dieser Signale entspricht einer anderen Adresse. Im Kontrollsystem muss die Zuordnung eines L0-Gerätes und der eingestellten C2-SPI-Adresse fest vorgegeben werden.

### 3.4.3 Optokoppler-Schaltungsteil

Durch die Verwendung von Optokopplern (Typ HCPL0638) in einem kleinen SMD-Gehäuse war es möglich, diese zusätzlich auf der L2-Platine unterzubringen. Somit konnte hier die L1-Ebene eingespart werden. Für die Ansteuerung der IFC werden 17 Ausgänge (14 Daten-, 2 Steuer-Signale und 1 Strobe-Signal) und 16 Eingänge (14 Daten- und 2 Status-Signale) benötigt. Siehe auch Abbildung 2.3.

Bei der Auswahl der Optokoppler wurde sich wesentlich an der Schaltung der MACS-Optokopplerkarten orientiert, welche bisher zur Ansteuerung genutzt wurden. In diesen werden schnelle Optokoppler, basierend auf einer Infrarot-Lumineszenzdiode und einer in Sperrrichtung betriebenen Fotodiode, die bei Lichteinfall den Basisstrom für einen Transistor liefert, eingesetzt. Der offene Kollektor des Transistors stellt den Ausgang dar. Der Optokoppler kann somit seinen Ausgangsanschluss aktiv nur auf Masse ziehen. Um den high-Zustand einzunehmen, werden Widerstände eingesetzt, die den Ausgang auf positives Versorgungsspannungsniveau ziehen. Für die Ausgangsrichtung sind diese nicht auf der L2-Schaltung, sondern in der IFC eingebaut und weisen einen Wert von 2 k $\Omega$  auf. Die Optokoppler benötigen auf der Empfangsseite eine Versorgungsspannung von 5 V. Diese liefert die IFC und wird im Datenkabel mitgeführt.

### 3.4.4 Gesamtschaltung

Für einen Überblick der L2-Schaltung sei zunächst auf Abbildung 3.6 hingewiesen. Neben den bereits beschriebenen Teilen für die Optokoppler und das C2-SPI, werden drei Mehrfarb-LEDs zur Statusanzeige benutzt, deren Bedeutung in Tabelle 3.2 aufgeschlüsselt ist. Um das Abarbeiten einer Rampe durch ein externes Signal zu starten, wurde in der Schaltung eine Triggerfunktionalität realisiert.



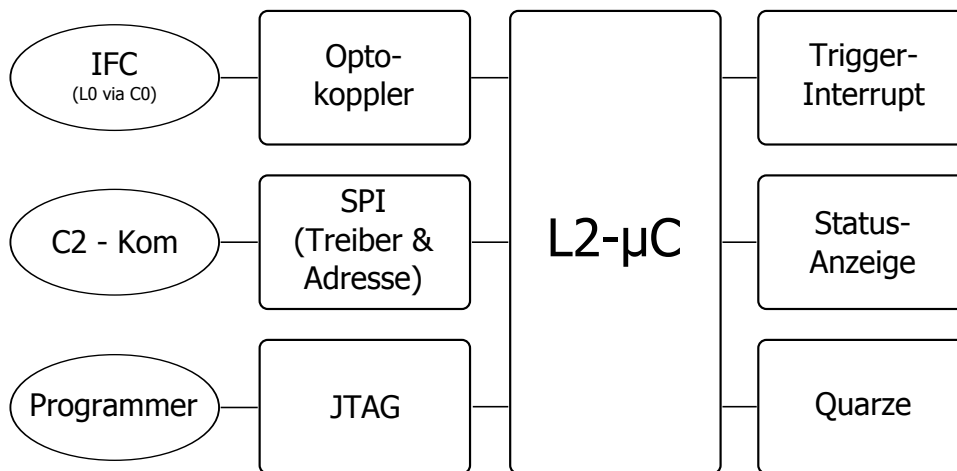


Abbildung 3.6: Blockschaltbild des L2-Modul

LED	Anzeige	Bedeutung
L2-Status	grün gelb rot	Modul arbeitet einwandfrei C2-Kommunikation hat statt gefunden eine Störung liegt vor
Rampen-Status	grün gelb rot aus	Rampe wird abgefahren eine Rampe ist geladen und es wird auf ein Triggerereignis gewartet keine gültige Rampe geladen es wurde ein fixer DAC-Wert gesetzt
Holec-Status	grün rot	IFC bereit IFC nicht bereit

Tabelle 3.2: Bedeutung der Statusanzeige des L2-Modules

Auf den rückwärtigen Bus können zwei mit einem L1-Trig-Modul erzeugte Triggersignale gelegt werden. Auf der Leiterplatte des L2-Moduls kann hiervon eines, durch Setzen einer Steckbrücke, ausgewählt werden. Das Signal wird dem EIC-Modul (External Interrupt Controller) des Mikrocontrollers zugeführt. Um verschiedene Programmteile in regelmäßigen Zeitintervallen auszuführen, kommt das RTC-Modul (Real Time Clock) des Mikrocontrollers zum Einsatz. Für dieses ist neben dem primären 12 MHz-Quarz, ein „Uhren“-Quarz mit einer Frequenz von 32,768 kHz eingebaut. Zur Programmierung wurde eine Steckverbindung für das JTAG-Interface vorgesehen. Das Verbindungskabel zu den IFC wird an einem 44-poligen Sub-D-Stecker angeschlossen. Die Schaltung benutzt neben der 5 V-Grundversorgungsspannung auch eine über einen Linearspannungsregler erzeugte 3,3 V Spannung.

Der vollständige Schaltplan ist in den Abbildungen B.1 und B.2 dargestellt.

### 3.4.5 L2 – Programmierung

Die wesentliche Struktur des Programms ist im Flussdiagramm 3.7 verdeutlicht. Nach einem Reset werden zunächst das L2-Modul selbst und die C2-Kommunikation initialisiert. Daran schließen sich Initialisierungsroutinen für die Optokopplertreiber und die IFCs an. In der Hauptschleife gibt es im wesentlichen drei Auslöser für Operationen.

#### Von der L4-Ebene angeforderte Operationen

Zum ersten können Befehle durch ein von der L4-Ebene stammendes C2-Paket aufgerufen werden. Dazu wird das Feld „befnr“ des C2pak\_t ausgewertet. Hierfür gibt es eine indizierte Liste (Array) mit Funktionszeigern. Mit den Befehlsnummern kann somit die zugehörige Funktion, welche das empfangene Paket weiterverarbeitet, aufgerufen werden. Bezüglich der IFC gibt es Funktionen für: DAC- und ADC-Operationen, Setzen der Steuerbits und Auslesen der Status-/Fehler-Bits. Für die L2-Ebene gibt es Funktionen zur Berechnung von Rampen-Werten, Generierung eines Software-Triggers zum Starten einer Rampe sowie die Möglichkeit die L2-Module zu steuern und zu konfigurieren. Um die Erreichbarkeit eines L2-Moduls zu testen, wurde eine „Ping“-Funktion implementiert. Eine ausführliche Beschreibung des Paketaufbaus zur Befehlsübermittlung und der genauen Funktion der Befehle ist in Kapitel 4 zu finden.

#### Zeitunkritische regelmäßige Operationen

Zum zweiten gibt es Funktionen, die in regelmäßigen Zeitabständen ausgeführt und im Folgenden als „Task“ bezeichnet werden. Da diese nicht zeitkritisch sind, wird auf die Zeitangabe des RTC-Moduls (Real-Time-Clock) zurückgegriffen. Jedem Task werden Variablen zugeordnet, welche den Zeitpunkt der nächsten gewünschten Ausführung („*nexttime*“) und das Intervall zwischen den Ausführungen („*intervall*“) beinhalten. In der Hauptschleife wird nun immer wieder geprüft, ob die Zeitvariable der RTC größer als die *nexttime* eines Tasks ist. Falls dies zutrifft, wird die Routine des Tasks ausgeführt und anschließend die *nexttime* um den Wert *intervall* erhöht. Es gibt insgesamt sechs Tasks mit den folgenden Aufgaben:

- C2-Kommunikation
- LED-Anzeige aktualisieren
- den ADC der IFC auslesen
- den Status prüfen und bei relevanten Änderungen an die L4-Ebene senden
- prüfen ob eine IFC angeschlossen ist und bereit ist
- die RTC vor einem Werteüberlaufen zurücksetzen

#### Zeitkritische regelmäßige Operationen

Für die Abarbeitung zeitkritischer Operationen werden Interrupts genutzt. Durch ein externes oder internes, also vom Mikrocontroller selbst erzeugtes, Signal wird die Programmausführung unterbrochen und eine spezielle, dem jeweiligen Interrupt zugeordnete Programmroutine ausgeführt. Nach Abarbeitung dieser Routine wird die vorherige Programmausführung an der Stelle der Unterbrechung wieder fortgesetzt.

Hat ein L2-Modul eine gültige Rampe berechnet und ist entsprechend konfiguriert, wird durch Auslösen eines Trigger-Ereignisses (also eines externen Interrupts) eine Interruptroutine aufgerufen, welche das Abfahren der Rampe startet. Dazu aktiviert sie einen Hardware-Timer mit entsprechenden Werten, welcher wiederum in regelmäßigen Zeitabständen eine zweite Interruptroutine aufruft. So wird der DAC-Wert der IFC regelmäßig aktualisiert und die gewünschte Rampe abgefahren. Die Zeitbasis mit welcher der Timer getaktet wird ist 10  $\mu$ s. Alle Zeitvariablen verstehen sich daher in Einheiten von 10  $\mu$ s. Zur Speicherung der Rampenwerte wird die folgende Form genutzt.



#### 3.4.6 L2 – Platinen-Layout

Für das Layout der L2-Platine gelten die gleichen Vorgaben wie für die L3-Platine. Die Schaltung wurde wieder auf einer Euro-Platine mit zwei Kupferlagen aufgebaut. Zur Erhöhung der Fehlersicherheit wurde auf eine saubere Trennung, der durch die Optokoppler galvanisch getrennten Schaltungsteile, geachtet. Die Ausgangsoptokoppler werden durch Treiber-ICs gepuffert. Durch die Verwendung eines „high-density“-Steckkontaktes, können Stecker und LEDs wieder auf einer gemeinsamen 5 TE breiten Frontplatte montiert werden. Eine Abbildung des Platinen-Layouts ist in Abbildung 3.8 zusehen.

#### 3.4.7 Unterschiede der Bridge-Select-Variante

Neben dem Ansteuern der IFC muss darüber hinaus auch ein Signal zum Umschalten zwischen den beiden Brücken-Konfigurationen erzeugt werden. Dazu lässt sich nicht die oben geschilderte Schaltung benutzen, da ein anderer Optokoppler-Ausgang benötigt wird. So musste ein abgewandeltes L2-Modul erstellt werden. Dieses nutzt das Layout der Standard-L2-Platine, jedoch ohne die normalen Optokoppler und deren Treiber-IC. Stattdessen wurde ein einzelner Optokoppler vom Typ 4N25 eingebaut und sein Signal über eine „LEMO“-Buchse herausgeführt. Dieses Modul wurde außerdem anders programmiert. Die Grundstruktur des Programms wurde beibehalten, nur alle für das Holec-Netzgerät spezifischen Funktionalitäten sind nicht einprogrammiert worden. Die Auflösung der Rampe wurde auf ein Bit modifiziert, das heißt auf die Werte 0 oder 1.

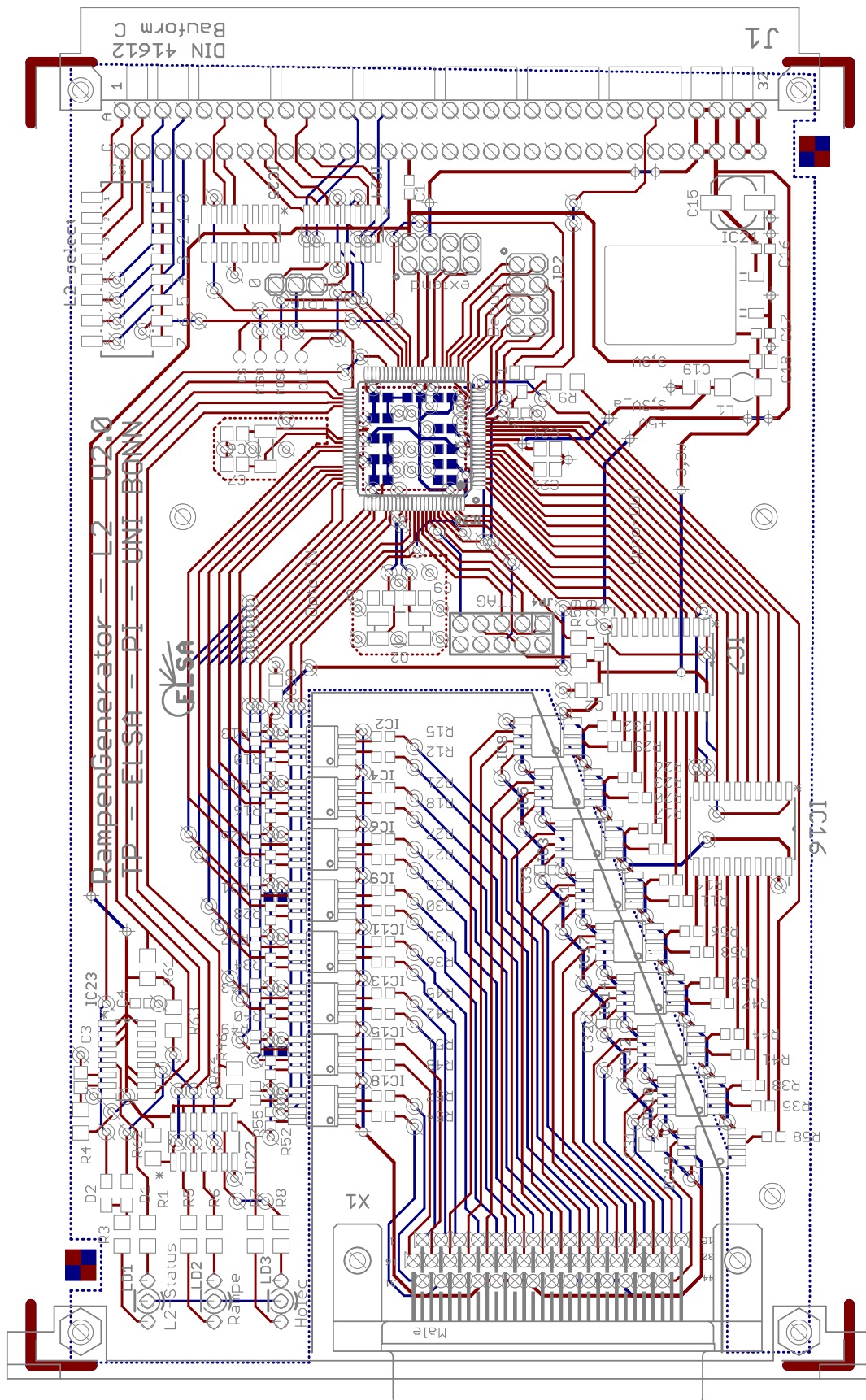


Abbildung 3.8: Platinen-Layout des L2-Moduls

### 3.5 L1 – Trigger-Modul

Wie bereits in Abschnitt 3.4.5 beschrieben, wird das Starten der Rampenabarbeitung von den L2-Modulen durch ein externes Interrupt-Signal ausgelöst. Dieses liegt mit auf dem rückwärtigen Bus. Als Quelle hierfür dienen zwei externe Trigger-Signale: der „Zyklusstart“-Trigger (für das Bridge-Select-Modul) und der „Rampstart“-Trigger (für das Ramp-Gen-Modul). Um die Triggersignale in den Bus einzuspeisen, wird die in Abbildung 3.9 wiedergegebene Schaltung benutzt.

Das Eingangssignal wird über einen Optokoppler (HCPL2602) galvanisch getrennt. Dieser ist verpolungs- und überspannungsgeschützt und liefert ein digitales Ausgangssignal. Zur Buseinspeisung wird ein Treiber-IC mit differentiellem Ausgang benutzt. Da dieser gleich vier Treiberstufen beinhaltet, kann er zusätzlich zur Erzeugung eines invertierten Signals genutzt werden. Durch Setzen einer Steckbrücke kann ausgewählt werden, bei welcher Flanke des Eingangssignals ein Interrupt ausgelöst werden soll. Das Auftreten eines solchen optisch über LEDs signalisiert.

Auch für diese Schaltung wurde ein Layout erstellt und eine Platine gefertigt.

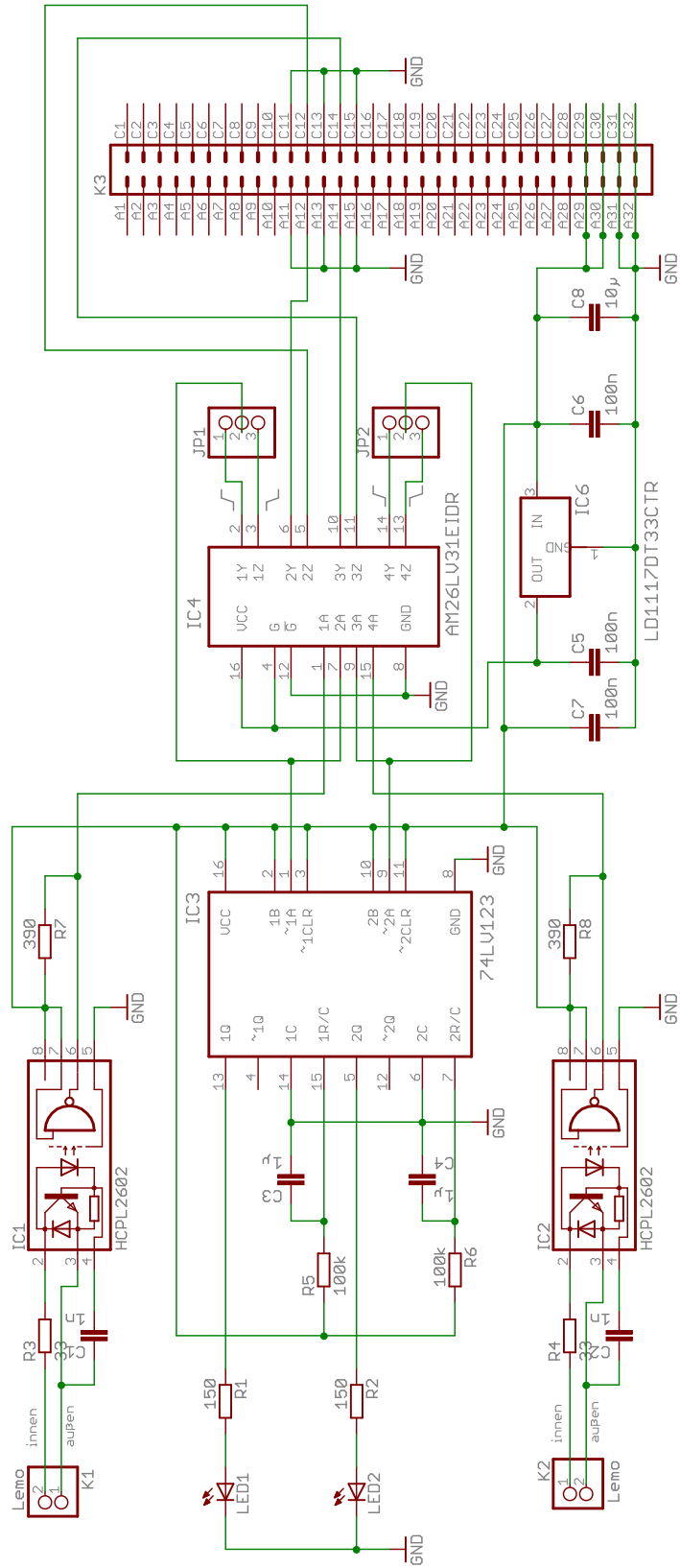


Abbildung 3.9: Schaltplan des L1-Trig-Modules





---

## Die L3-/L2-Befehle

---

### 4.1 Allgemeines

Die Befehle für die L3- und L2-Ebene werden als Datenpaket mit der durch die Typendefinition „`c2pak_t`“ festgelegten Struktur übermittelt. Für Referenzierungen sei auch auf die Dateien „`C2_pak.h`“, „`C2_pak.c`“ sowie „`L2_bef.h`“ und „`L2_bef.c`“ verwiesen.

```
typedef struct{
    //1. 32bit-Wort
    uint8_t  channel; //Routing-Information für die L3-Ebene
    uint8_t  adr;     //Empfänger-Adresse des L2-Moduls
    uint16_t len;     //Länge der Nutzdaten in Byte (Vielfache von 4)
    //2. 32bit-Wort
    uint16_t pakID;  //ID-Nummer des Paketes (von L4-Ebene vergeben)
    uint8_t  befnr;  //Nummer des Befehls (Index des Bef_list-Arrays)
    uint8_t  kdata;  //Kurzdaten-Feld, welche vom Befehl benutzt werden
    //3. 32bit-Wort
    uint32_t chk;    //Checksumme (neg. ex-or)
    //4. .. (len/4 + 3) 32bit-Wort
    uint8_t  data;   //Anfang des Nutzdaten-Bereiches
} c2pak_t;
```

Um ein solches Paket zu erstellen, steht die Funktion „`c2pak_build`“ zur Verfügung. Diese erweitert gegebenenfalls den reservierten Speicher (mit `realloc`) oder, falls dem Paket noch kein Speicher zugewiesen worden ist, reserviert diese einen neuen Speicherbereich (mit `malloc`). Zum Füllen des Nutzdatenbereiches kopiert sie die mit „`len`“ angegebene Anzahl von Bytes von der mit `data` übergebenen Adresse. Die Checksumme `chk` wird entsprechend gesetzt und ein Zeiger auf das Paket zurückgegeben. Soll die L2-Ebene adressiert werden, ist `channel` auf „`CHANNEL_SPI`“ und `adr` auf den Adresswert (von 0 bis 7) zu setzen. Zur Adressierung der L3-Ebene ist als `channel` „`CHANNEL_L3`“ zu benutzen. Das Adressfeld wird in diesem Fall nicht verwendet. Der „`CHANNEL_UDP`“ ist primär für Pakete, welche von der L3-Ebene ausgehen, konzipiert. Alle von einem L2-Modul gesendeten Pakete werden in der L3-Ebene mit der entsprechenden Adresse ergänzt (die L2-Module sind sich selbst ihrer Adresse nicht bewusst).

```
c2pak_t * c2pak_build(c2pak_t *pak, uint8_t channel, uint8_t adr, uint16_t len,
                    uint16_t pakID, uint8_t befnr, uint8_t kdata, void *data);
```

Alle empfangenen Pakete werden zunächst mit einem Quittierungspaket beantwortet und danach bearbeitet. Abhängig vom Befehl wird zuletzt ggf. ein Antwortpaket verschickt. Sollen zurück gesendete Pakete zuordnen können, ist desweiteren der Wert der *pakID* zu setzen. Diese wird in der L3- bzw. L2-Ebene nicht verändert, sondern nur zurückgespiegelt.

### 4.2 L2-Statusflags

Im Programm der L2-Module wird eine 16 Bit-Variable benutzt, um verschiedene Status- und Steuerinformationen mit der Größe von jeweils einem Bit, sogenannte „Flags“ (Flaggen), zu hinterlegen. Tabelle 4.1 gibt einen Überblick über alle definierten Flags. Es wird nur der hintere Teil der Flagnamen angegeben, um den vollständigen Namen zu erhalten ist diesem noch ein „L2STATUS\_FLAG\_“ voranzustellen (siehe auch „L2\_bef.h“).

Neben den Flags, die nur einen Status widerspiegeln, gibt es auch Flags mit zusätzlich steuernder Wirkung. Diese haben vornehmlich auf die Wiedergabe der Stromrampe Einfluss. Diese Unterscheidung ist in der Spalte „Art“ kenntlich gemacht. Ob Flags durch den L2-Steuer-Befehl modifiziert werden können, ist für jedes Bit einzeln festgelegt. Dies ist in den Spalten „set“ (setzbar) und „clr“ (löschar) eingetragen. Des Weiteren gibt es Flags, bei deren Änderung ein Status-Paket an die L4-Ebene geschickt wird (Spalte „cha“).

Hat der Mikrocontroller aus den ihm zugesendeten Paketen eine gültige Rampe berechnet, wird das RAMPEN\_DATEN-Flag gesetzt. Sobald ein Fehler bei der Berechnung auftritt, wird das RAMPEN\_FEHLER-Flag gesetzt und alle weiteren zu dieser Rampe gehörenden Pakete verworfen. Da Zahldarstellungen auf Prozessoren endlich sind, kann es bei Berechnungen zu Werten kommen, die zu groß bzw. zu klein sind, um in den gewünschten Datentyp abgespeichert werden zu können. Findet dies bei der Rampenberechnung statt werden die größten bzw. kleinsten noch darstellbaren Werte abgespeichert, dies bezeichnet man als Sättigung. Das RAMPEN\_SAETTIGUNG-Flag weist auf das Auftreten einer solchen Über-/Unterlauf-Sättigung während der Berechnung hin. Dies stellt keinen Fehler dar, sondern weist nur darauf hin, dass die berechnete Rampe eventuell von dem gewünschten Verlauf abweicht.

Wurde das Flag RAMPEN\_MONOZYK gesetzt, wird die Rampe nur ein einziges Mal, ungeachtet weiterer Triggerereignisse, wiedergegeben. Mit dem RAMPEN\_PERIODISCH-Flag lässt sich eine, besonders für Testzwecke nützliche, periodische Rampenwiedergabe erreichen, d. h. wenn das Ende der definierten Rampenwerte erreicht wurde, wird wieder an den Anfang gesprungen. Das Starten einer Rampenwiedergabe durch einen externen Interrupt findet nur bei gesetztem TRIGGER\_ENABLE-Flag statt. Dieses Flag wird beim Setzen eines fixen DAC-Wertes oder beim Starten einer monozyklischen Rampe zu 0 gesetzt.

Ob bereits ein Trigger-Ereignis stattgefunden hat, wird mit dem TRIGGER-Flag signalisiert. Wird aktuell eine Rampe wiedergegeben, lässt sich dies am RAMPE\_AKTIV-Flag erkennen. Bei gesetztem DAC\_OFFSET-Flag wird zum DAC-Wert aus der Rampendefinition ein zusätzlicher Offset-Wert hinzu addiert. Wurde ein fester Strom-Sollwert gesetzt, wird dies mit dem FIXDAC-Flag kenntlich gemacht. Das C2AUSFALL-Flag ist nur für progamminterne Zwecke zugänglich, so verhindert es zum Beispiel die Rampenwiedergabe bei Ausfall der Kommunikation.

Nach einem Reset müssen in den L2-Modulen einige Parameter für die DAC-Ansteuerung festgelegt werden, um eine unzulässige Ansteuerung zu verhindern. Dazu zählen der Minimalwert und die maximale Steigungsrate des Sollwertes. Diese hängen von der angeschlossenen L0-Ebene ab und können somit nicht fest einprogrammiert werden. Das L2-Modul signalisiert mit dem RESET-Flag einen erfolgten Reset und unterbindet die weitere Ansteuerung der L0-Ebene. Wurden alle oben genannten Werte initialisiert, wird auch das RESET-Flag gelöscht.

Bezeichnung	Art	set	clr	cha	Erläuterung
RAMPEN_DATEN	Status	nein	ja	nein	Es liegen gültige Rampenwerte vor
RAMPEN_FEHLER	Status	nein	nein	nein	Bei der Berechnung der Rampenwerte ist ein Fehler aufgetreten
RAMPEN_SAETTIGUNG	Status	nein	nein	nein	Bei der Berechnung der Rampenwerte sind diese in Sättigung geraten
RAMPEN_MONOZYK	Steuer	ja	ja	nein	Rampe wird nur einmal durchlaufen
RAMPEN_PERIODISCH	Steuer	ja	ja	nein	Rampenwerte werden periodisch wiedergegeben
TRIGGER_ENABLE	Steuer	ja	ja	ja	Bei eingehendem Trigger wird die Rampe wiedergegeben
TRIGGER	Status	nein	ja	ja	Ein Triggerereignis hat stattgefunden
RAMPE_AKTIVE	Status	nein	nein	nein	Die Rampe wird aktuell wiedergegeben
DAC_OFFSET	Steuer	ja	ja	nein	Aktiviert den OFFSET-Wert für den DAC
FIXDAC	Status	nein	nein	ja	DAC-Wert wurde als Einzelwert gesetzt (keine Rampe)
C2AUSFALL	Status	nein	nein	nein	C2_L2_quality unterschreitet unteren Grenzwert
RESET	Status	nein	nein	ja	Win Reset hat statt gefunden und es wurde noch kein L2_steuer-Paket empfangen
ADCAUTO	Steuer	ja	ja	nein	Der HOLEC-ADC wird regelmäßig ausgelesen und an L4 gesendet

Tabelle 4.1: Aufschlüsselung der L2-Status-Flags (Erläuterung siehe Text)

Sollen regelmäßig Angaben der vom ADC gemessenen Istwerte erfolgen, ist das Flag ADCAUTO zu setzen. Die automatische Werteübermittlung ist mit weiteren Steuerparametern konfigurierbar (siehe Abschnitt 4.3).

### 4.3 Aufbau und Funktion der Befehle

Im Folgenden wird eine Beschreibung der Funktion der einzelnen Befehle gegeben und dargestellt, wie die zugehörigen C2-Pakete aufgebaut werden. Da es vom Hardware-Level und der Variante des L2-Modules abhängt, welche Befehle implementiert sind, ist dies jeweils vermerkt. Zusätzlich wird der Name der C-Routine genannt. Diese finden sich für die L3-Ebene in der zugehörigen „main.c“-Datei und bei Routinen der L2-Ebene in der „bef.c“-Datei.

#### **Ping-Befehl (alle L2-/L3-Module; L3\_intern() bzw. bef\_fkt\_ping())**

Dieser Befehl stellt ein Diagnosewerkzeug zum Testen und zur Qualitätsbeurteilung der Kommunikationswege dar. Er kann an die L3-Ebene geschickt werden, um nur die UDP-Verbindung zu testen, oder zur Überprüfung der C2-Verbindung an die L2-Module gesendet werden. Als Erkennungstext steht in

dem Datenfeld des Antwortpaketes der Text „pong“. Überdies hinaus kann auch der normale ICMP<sup>1</sup>-Ping genutzt werden.

struct-Feld	Befehl	Antwort
channel	CHANNEL_SPI / CHANNEL_L3	CHANNEL_UDP
len	0 .. C2PAK_MAX_DATA_SIZE	4
befnr	BEFNR_PING	ANTNR_PONG
kdata	irrelevant	kdata
data	irrelevant	0x504F4E47 = „pong“ (ASCII)

### Reset-Befehl (nur L3-Modul; L3\_intern())

Dieser Befehl generiert auf den entsprechenden Leveln ein Reset. Es werden keine Antwortpakete generiert, aber von der L2-Ebene werden automatisch Status-Informationen gesendet. Es sei angemerkt, dass es sich um keinen Software-Reset, sondern um einen echten Hardware-Reset handelt.

struct-Feld	Wert	
channel	CHANNEL_L3	
len	0	
befnr	BEFNR_L2RESET	BEFNR_L3RESET
kdata	Adresse des L2-Moduls	bel.
data	-	

### L2-Steuer-Befehl (Alle-L2-Module; bef\_holec\_L2steuer())

Dieser Befehl erlaubt die L2-Module zu konfigurieren und die Rampendarstellung zu steuern. Es können die in Tabelle 4.1 angegebenen Flags verändert werden. Dazu gibt es eine Bit-Maske zum Setzen und eine zum Löschen der Flags. Für Erfassung von Istwerten mit dem ADC, kann die Zahl der zu mitteln- den Messungen und das Intervall selbiger vorgegeben werden. Darüber hinaus können auch die bereits erwähnten Variablen des Minimalwertes und der maximalen Steigungsrate des Sollwertes (DAC-Wert) mit diesem Befehl konfiguriert werden.

struct-Feld	Befehl
channel	CHANNEL_SPI
len	sizeof(bef_L2_steuer_t)
befnr	BEFNR_HOLEC_L2STEUER
kdata	irrelevant
data	gemäß bef_L2_steuer_t

```
typedef struct {
    uint16_t L2_status_flags_set;           //Bit-Maske der Flags die gesetzt werden
                                           //sollen
    uint16_t L2_status_flags_clear;       //Bit-Maske der Flags die gelöscht werden
                                           //sollen
    uint16_t timed_holecadc_mittelung;     //Wert über wie viele Messungen gemittelt
                                           //werden soll
    uint16_t xdummy;                       //wg. alignment
}
```

<sup>1</sup> Internet Control Message Protocol; Bestandteil vom IPv4-Protokoll

```

uint32_t timed_holecadc_intervall; //Zeitintervall der Istwert-Erfassung
uint16_t holec_dac_standby;       //Default-Wert des DACs (U14Q2)
uint16_t holec_slew_rate;         //Maximale Steigungsrate der Sollwerte
                                   //pro 1/ms (U14Q2)
} bef_L2_steuer_t;

```

Die Implementierung dieser Funktion auf dem Bridgeselect-Modul ist ähnlich. Aufgrund der einfachen Gestalt der Rampe genügt es, die relevanten Flags modifizieren zu können.

### Status-Befehl (Alle-L2-Module; bef\_holec\_status())

Dieser Befehl fordert die Status-Informationen des L2-Moduls und der an dieses angeschlossene IFC an. Sie werden auch automatisch an die L4-Ebene gesendet, falls sich bestimmte Flags des L2-Status ändern (siehe Tabelle 4.1). Die Daten sind gemäß des „L2\_status\_t“-struct angeordnet. Nach den L2-Flags folgt der aktuell gesetzte DAC-Wert. Die nächsten beiden relevanten Einträge beinhalten das Status- und Fehler-Wort der IFC. Als letzter Parameter wird die Qualität der C2-Kommunikation übermittelt (Für Informationen zu der genauen Bedeutung dieses Parameters, sei auf den Programmcode der „C2\_L2\_get“-Routine in Datei C2\_L2L3\_d.c verwiesen).

struct-Feld	Befehl	Antwort
channel	CHANNEL_SPI	CHANNEL_UDP
len	0	sizeof(L2_status_t)
befnr	BEFNR_HOLEC_STATUS	ANTNR_HOLEC_STATUS
kdata	irrelevant	0
data	–	gemäß L2_status_t

```

typedef struct {
uint16_t L2_flags;           //status-flags des L2-Modules
uint16_t holec_dac;         //aktuell gesetzter DAC-Wert (U14Q2)
uint16_t holec_dacoffset;   //z.Z. nicht implementiert
uint16_t holec_status;     //Status-Wort der IFC
uint16_t holec_fehler;     //Fehler-Wort der IFC
uint16_t C2_L2_quality;    //Qualität der C2-Verbindung
} L2_status_t;

```

Das Modul der Bridgeselect-Variante antwortet in gleicher Form; nur werden alle nicht relevanten Felder auf 0 gesetzt.

### HOLEC-Steuerungs-Befehl (HOLEC-L2; bef\_holec\_steuer())

Mit diesem Befehl lassen sich die Steuerungsleitungen der IFC aktivieren. Hierzu werden die den Steueraktionen entsprechenden Bitmuster in das Kurzdatenfeld des Paketes geschrieben. Die gültigen vordefinierten Bitmuster lauten holec\_steuer\_auxon/auxoff (Hilfsbetriebe), holec\_steuer\_mcbon/mcboff (Hauptschalter), holec\_steuer\_resetfaults, holec\_steuer\_release und holec\_steuer\_standby. Alle Pakete mit von diesen abweichenden Bitmustern werden ignoriert.

struct-Feld	Befehl
channel	CHANNEL_SPI
len	0
befnr	BEFNR_HOLEC_STEUER
kdata	holec_steuer_??
data	-

**SuperDACADC-Befehl (HOLEC-L2; bef\_holec\_superdacadc())**

Alle Aktionen bezüglich DAC und ADC lassen sich mit diesem Befehl realisieren. Um die auszuführenden Aktionen auszuwählen, können verschiedene Flags angegeben werden. Die BEF\_HOLEC\_SUPERDACADC-Flags werden in das kdata-Feld des Paketes eingetragen.

Ist das \_DACSET-Flag gesetzt, wird der dac\_value Eintrag aus dem Daten-Feld als neuer Sollwert genutzt. Eventuell aktive Rampen werden unterbrochen. Stattdessen wird entsprechend des Werts von „holec\_slew\_rate“, eine lineare Rampe bis zum Sollwert gefahren. Entsprechend werden die L2STATUS\_FLAG-Einträge aktualisiert: das Flag \_FIXDAC wird gesetzt und die Flags \_RAMPE\_AKTIVE, \_TRIGGER\_ENABLE und \_TRIGGER werden gelöscht.

Hierbei wird auch der ADC getriggert. Soll kein neuer DAC-Wert gesetzt werden, aber trotzdem eine neue AD-Konvertierung gestartet werden, muss das \_ADCTRIG-Flag gesetzt werden.

Durch Setzen des \_ADCREAD-Flag wird der ADC-Istwert angefordert. Dieser ist nur aktuell, wenn das \_ADCTRIG- oder \_DACSET-Flag gesetzt wurde. Der so ermittelte Ist-Wert wird in dem adc\_value-Eintrag zugeordnet. Der aktuelle Sollwert findet sich im dac\_value. Falls der ADC-Wert nicht ermittelt werden konnte, wird im kdata-Feld des Antwort-Paketes das FEHLER\_HOLEC\_SUPERDACADC\_ADCREADFEHLER-Flag gesetzt.

Falls der ADC nicht gelesen werden soll, kann der Sollwert durch Setzen des \_DACVARIABLE-Flag anfordert werden.

struct-Feld	Befehl	Antwort
channel	CHANNEL_SPI	CHANNEL_UDP
len	sizeof(bef_holec_dacadc_t)	sizeof(bef_holec_dacadc_t)
befnr	BEFNR_HOLEC_SUPERDACADC	ANTNR_HOLEC_SUPERDACADC
kdata	(Steuer-Flags)	(Fehler-Flag)
data	gemäß bef_holec_dacadc_t	gemäß bef_holec_dacadc_t

```
typedef struct {
    uint16_t dac_value;
    uint16_t adc_value;
} bef_holec_dacadc_t;
```

**Rampenberechnung-Befehl (Alle L2-Module; bef\_holec\_rampenberechnung\_2())**

Der Befehl zur Berechnung von Rampen ist der einzige, der aus mehreren C2-Paketen bestehen kann. Um die geforderten Rampenverläufe günstig beschreiben zu können, werden diese in einzelne Abschnitte unterteilt. Jeder Abschnitt wird durch einen sogenannten Rampendefinitionsblock dargestellt. Für diese gibt es wiederum drei Typen, welche durch verschiedene „structs“ beschrieben werden.

struct-Feld	Befehl	Antwort
channel	CHANNEL_SPI	CHANNEL_UDP
len	(variabel)	2
befnr	BEFNR_HOLEC_RAMPE	ANTNR_HOLEC_RAMPE
kdata	(siehe Text)	0xFE bzw. 0xFD
data	(variabel)	L2_status.L2_flags

Alle Rampendefinitionsblock-structs beginnen gleich:

```
typedef struct { // allgemeiner Header aller Rampendefinitionen
    uint8_t    type;
    uint8_t    option;
    uint16_t   n;
    ...
}
```

Der type-Eintrag gibt den Rampendefinitionsblock-Typ an und  $n$  die Anzahl der durch ihn definierten Stützstellen der Rampe. An dem `_ERSTESPAK`-Flag im Option-Feld wird erkannt, ob es sich um das erste Paket einer neuen Rampendefinition handelt. Nur dann werden die beiden Flags `_AKTIVE`, `_MONOZYK` und `_PERIODISCH`, welche die korrespondierenden Flags in der `L2_status.L2_flags`-Variable festlegen, ausgewertet (alle Flagnamen beginnen mit `BEF_HOLEC_RAMPE_OPT`).

Um sicherzustellen, dass alle Blöcke in der richtigen Reihenfolge verarbeitet werden, muss das `kdata`-Feld des ersten Paketes die Gesamtanzahl der Blöcke beinhalten. Alle folgenden Pakete geben in diesem Feld an, wie viele Rampendefinitionsblöcke vor ihnen verarbeitet worden sein müssen. Beim Empfang des ersten Paketes wird die Rampenwiedergabe gestoppt und der Standbywert angefahren.

Bei der eigentlichen Berechnung werden aus den Parametern der Rampendefinitionsblöcke Wertereihen, wie in Abschnitt 3.4.5 beschrieben, in einen dafür reservierten Speicherbereich geschrieben. Im Folgenden werden die drei implementierten Rampendefinitionsblocktypen vorgestellt.

**TYPE\_KONST:** Mit Rampendefinitionsblöcken dieses Typs können Abschnitte mit konstantem Wert erzeugt werden. Durch Werte  $n > 1$  kann ein regelmäßiges, mehrfaches Setzen dieses Wertes erzwungen werden. Der Parameter `c0` entspricht direkt dem gewünschten Sollwert.

```
typedef struct { // konstanter Wert
    uint8_t    type;
    uint8_t    option;
    uint16_t   n;
    uint16_t   dt;
    uint16_t   c0; //U14Q2
} bef_holec_rampe_konst_t;
```

**TYPE\_POLY:** Dieser Blocktype ermöglicht den Sollstrom-Verlauf mit einem Polynom dritten Grades nach folgender Gleichung zu beschreiben:

$$w_{i,j} = \sum_{k=0}^3 c_{i,k} (t_{\text{abs},i} - t_0 + j\Delta t_i)^k .$$

Wobei  $i$  die Blocknummer angibt und  $j$  den zu berechnenden Wert innerhalb des Blockes indiziert.  $t_{\text{abs}}$  berechnet sich aus den zuvor verarbeiteten Rampendefinitionsblöcken und gibt die Zeit relativ zum Anfang der Rampe, für den jeweils zu berechnenden Rampenblock, an. Alle Zeitparameter sind in Einheiten von  $10 \mu\text{s}$  angegeben, was gleichzeitig der feinsten Auflösung entspricht. Falls die Koeffizienten  $c_3 = 0$  bzw.  $c_3 = c_2 = 0$  sind, wird die Berechnung entsprechend abgekürzt, um Rechenzeit zu sparen.

Während der Berechnung wird für die Variable  $w_{i,j}$  eine 32-Bit-Gleitkommazahl benutzt. Diese muss anschließend in eine 16-Bit-Ganzzahl (genauer ein U14Q2) umgewandelt werden<sup>2</sup>. Ist der berechnete Wert größer als das zulässige Maximum, wird das Status-Flag „Sättigung“ gesetzt und der Wert auf den Maximalwert gesetzt. Analog wird bei Unterläufen verfahren.

```
typedef struct { //Polynom 3.Grades
    uint8_t    type;
    uint8_t    option;
    uint16_t   n;
    int32_t    t0;
    uint16_t   dt;
    uint16_t   xdummy1;
    int32_t    c0;
    float      c1;
    float      c2;
    float      c3;
} bef_holec_rampe_polynom_t;
```

**TYPE\_DIREKT:** Um jegliche Gestalt der Rampen zu ermöglichen, bietet dieser Typ die Möglichkeit, die Stützpunktwerte direkt anzugeben. Da die Anzahl der Stützstellen pro Block variabel ist, wurde nur für den minimalen Rampendefinitionsblock mit einem Wert ein struct angelegt. Auf den Eintrag `value_data` folgen weitere  $(n - 1)$  Werte gleichen Datentyps.

```
typedef struct { // Direkte Stützpunkt-Werte
    uint8_t    type;
    uint8_t    option;
    uint16_t   n;
    uint16_t   dt;
    uint16_t   value_data; //U14Q2
} bef_holec_rampe_direkt_t;
```

Auf dem Bridgeselect-L2-Modul wurde diese Funktion in abgewandelter Art implementiert. Die berechneten Werte der Rampe sind entweder 0 oder 1. Als Rampendefinitionsblocktyp steht nur `_HOLEC_RAMPE_TYPE_DIREKT` zur Verfügung.

### Softwaretrigger-Befehl (HOLEC-L2; `bef_holec_softwaretrigger()`)

Hiermit lässt sich unabhängig von den anliegenden Triggersignalen ein externer Interrupt generieren und so eine Rampe starten.

struct-Feld	Befehl
channel	CHANNEL_SPI
len	0
befnr	BEFNR_HOLEC_SOFTWARETRIGGER
kdata	irrelevant
data	–

<sup>2</sup> Der Wert der Float-Variable entspricht dem vierfachen des eigentlichen Rampenwertes. Dieser Faktor versteckt sich in der Zuweisungskette: `float` → `uint16_t` → `U14Q2`.



---

# Entwicklung und Test der Ansteuerelektronik

---

## 5.1 Entwicklung

Die Komplexität der Gesamtschaltung macht es nötig, die Software- und Schaltungsentwicklung parallel auszuführen.

Im ersten Stadium wurde die L3-Ebene zusammen mit der C3-Schnittstelle entwickelt. Die benutzte Schaltung wurde hierzu auf einer Steckplatine (Breadboard) aufgebaut, so dass Modifikationen einfach vorgenommen werden konnten. Da es zu Problemen mit dem hochfrequent getakteten PHY-Schaltungsteil kam, wurde dieser separat auf einer Streifenrasterplatine aufgelötet. Die Ethernet-Implementierung stellte besonders beim Einbinden des IP-Stacks LWIP, welcher für die Protokollverarbeitung zuständig ist, Probleme dar. Es zeigte sich, dass der MACB-Treiber des Mikrocontroller-Herstellers, welcher den hardware-spezifischen Aufgabenteil übernimmt, nicht alle Funktionalitäten des benutzten IP-Stacks unterstützt, so dass letzterer erst passend konfiguriert werden musste.

Um die Kommunikation zu testen und auch später Steuerbefehle senden zu können, wurde ein weiteres C-Programm geschrieben. Dieses läuft auf einem gewöhnlichen PC, der es auch ermöglichte, die Protokollheader und den Dateninhalt der Netzwerkpakete mit der Software „Wireshark“ zu analysieren. Das C-Programm wurde immer wieder erweitert, um weitere Teile der Ansteuerung der IFC testen zu können.

Nachdem die C3-Kommunikation funktionierte, wurde mit der Entwicklung des L2-Moduls fortgefahren. Auch dieses wurde zunächst auf Steckplatinen durchgeführt. Die C2-Schnittstelle wurde mit mehreren L2-Modulen ausprobiert. Zuletzt wurde eine separate IFC an ein L2-Modul angeschlossen und so einzeln die verschiedenen Ansteuerbefehle und die Rampenwiedergabe getestet.

In einem zweiten Entwicklungsstadium wurden zu den erstellten Schaltungen komplette Schaltpläne gezeichnet und daraus Platinenpläne (Layouts) erstellt. Hierzu wurde der Layout-Editor „EAGLE“ benutzt, wobei alle Schritte des Routings manuell ausgeführt wurden. Die von einer externen Firma gefertigten Platinen wurden anschließend bestückt und auf ihre Funktion getestet. Ergänzungen und einige kleine Fehler wurden durch Manipulation auf diesen Platinen korrigiert. Alle diese Arbeitsschritte wurden wiederholt und so eine zweite überarbeitete Version der Platinen erstellt.

## 5.2 Tests der neuen Ansteuerung

Bei weiteren Tests der Ansteuerung der IFC fiel zunächst auf, dass bei der Wiedergabe einer Rampe einige falsche DAC-Werte gesetzt wurden. Als Fehlerquelle wurde die steil fallende Flanke der Optokoppler-Ausgänge identifiziert. Diese wirkt auf die Versorgungsspannung und auch direkt auf die anderen Signalpegel zurück. Letzteres kommt durch das Übersprechen in der Kabelstrecke (beim Test wurde ein Kabel von rund 10 m Länge zur Verbindung von L2 und L0 benutzt) und einer Mitkopplung in den „pull-up“-Widerstandsnetzen der IFC zustande. Dieses Problem konnte durch verschiedene Entstörmaßnahmen nicht vollständig eliminiert werden. Durch Software-Anpassungen konnte zwar erzielt werden, dass keine ungültigen DAC-Werte gesetzt werden. Hiermit konnte allerdings nicht das ungewünschte Setzen von Steuerbits, die z.B. das Schalten des MCBs (10 kV-Hauptschalter) hervorrufen, unterbunden werden. Durch den Austausch eines der Optokoppler gegen einen Typ, dessen Anstiegs- und Abfallzeiten etwa 10-mal länger waren, konnte die Fehlerquelle verifiziert werden. Die Beeinflussung anderer Signale lag nun deutlich unter 100 mV statt einiger Volt bei dem schnellen Typ. Um die Störungen zu beseitigen, empfiehlt es sich die Optokoppler durch langsamere Typen zu ersetzen.

Trotz dieser Probleme wurden einige Tests der neuen Ansteuerung durchgeführt. Hierbei wurde durch herausnehmen des Transistor-Array-IC der IFC, welches die Steuerleitungen treibt, verhindert, dass diese ungewollt Schaltoperationen hervorrufen. Die Operationen zum Ansprechen der Steuerleitungen wurde ohne angeschlossenes Netzgerät getestet.

Für den Test wurden die erstellten Platinen in einem 19-Zoll-Baugruppenträger eingebaut. Als ein erster Schritt wurde die Kommunikation und korrekte Ansteuerung mittels des Kontrollsystems getestet. Es mussten verschiedene Anpassungen in den Programmen der Mikrocontroller und des Kontrollsystems gemacht werden. Nachdem die Kommunikation mit dem und die Anzeige im Kontrollsystem zufriedenstellend funktionierte, wurden vergleichende Messungen an dem Netzgerät für die D-Sextupole gemacht. Dabei wurde die Fähigkeit des Kontrollsystems genutzt, die alte und neue Netzgerätesteuerung parallel nutzen zu können.

### Testen der Steuerfunktion und des Auslesen der Statusinformationen

Bis auf die oben genannte Einschränkung wurde das Ausführen der Steueroperation positiv getestet. Auch das Auslesen der Fehler- und Statusbits funktionierte nach anfänglichen Zuordnungsproblemen einwandfrei.

### Messung verschiedener Zeitparameter

Es wurde untersucht, inwieweit die Triggersignale und dadurch der Startzeitpunkt der Rampenwiedergabe durch die neue Schaltung verzögert werden.

Die Signallaufzeit des Triggersignals in dem L1-Trig-Modul beträgt ca. 70 ns, bezogen auf die steigende Flanke des Eingangssignals. Durch dieses Modul entstehen also keine nennenswerten Verzögerungen.

Die effektive Gesamtverzögerung des Starts der Rampenwiedergabe wurde bestimmt, indem die Zeit zwischen eingehendem Triggersignal und dem Setzen des ersten DAC-Wertes der Rampe gemessen wurde. Für letzteres wurde die steigende Flanke des Strobe-Pulses auf der IFC genutzt. Es ergab sich eine Verzögerung von ca. 7  $\mu$ s, wovon alleine 3,1  $\mu$ s der Routine zum Setzen des DAC der IFC zugeordnet werden können. Entsprechende Messungen mit der alten Ansteuerung ergaben einen Wert von über 140  $\mu$ s.

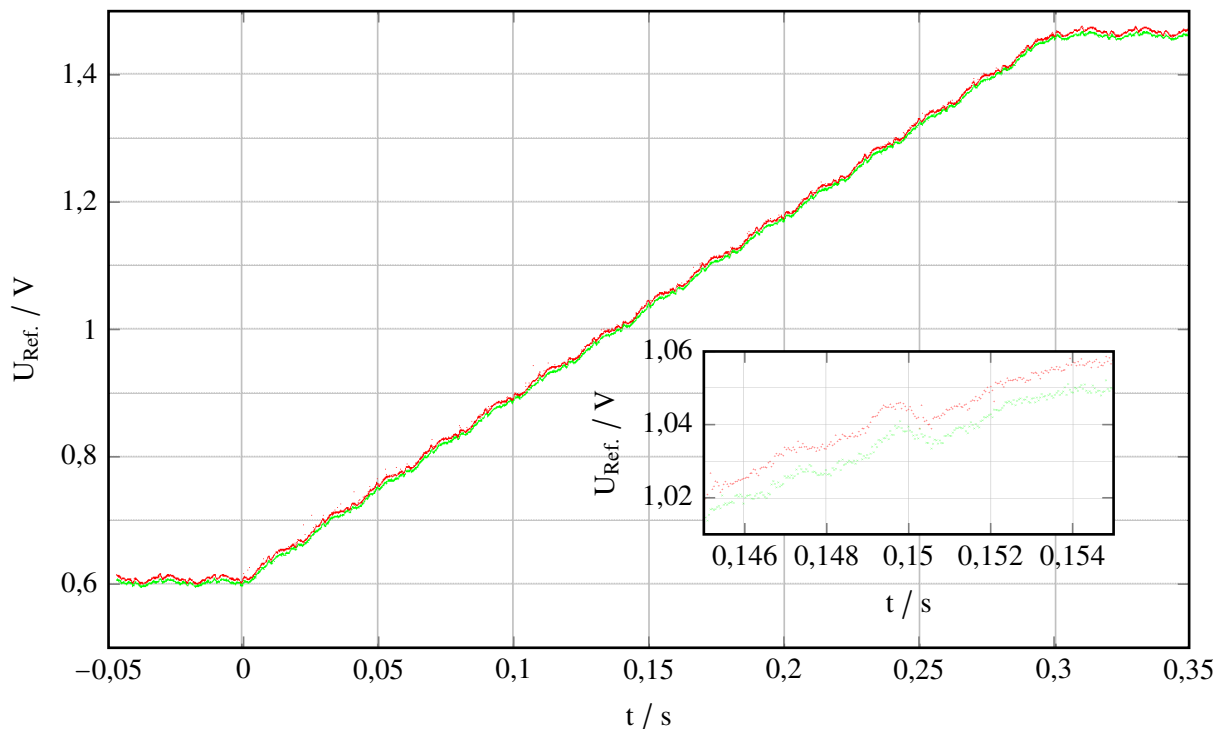


Abbildung 5.1: Die von dem DAC der IFC ausgegebene Referenzspannungen im Bereich der steigenden Rampe; gemessen mit der neuen (—) und alten (—) Ansteuerung. In dem vergrößerten Ausschnitt ist der gleichbleibende Offset zu erkennen.

Entsprechende Messungen wurde auch bezüglich dem Ende der Rampenwiedergabe gemacht. Bei der neuen Ansteuerung wurde der letzte DAC-Wert  $72\ \mu\text{s}$  vor dem eigentlichen Ende registriert. Die bei der Messung benutzte Rampe hatte eine Zyklusdauer von  $4,7\ \text{s}$ . Unter Berücksichtigung der Anfangsverzögerung ergibt sich somit eine Zeitabweichung von  $\frac{4,7\ \text{s}}{72\ \mu\text{s}} = 15,3\ \text{ppm}$ . Diese Abweichung ist mit der Grundtoleranz<sup>1</sup> des Oszillatorquarzes von  $\pm 20\ \text{ppm}$  erklärbar. Bei Benutzung der alten Ansteuerung wird eine Abweichung von rund  $2,7\ \text{ms}$  beobachtet. Dies entspricht einem relativen Fehler von  $574\ \text{ppm}$ . Bei der Interpretation dieser Messergebnisse ist zu berücksichtigen, dass der Zeitbasisfehler des zu diesen Messungen benutzten Oszilloskops unbekannt ist.

### Messungen des durch den DAC erzeugten Rampenverlauf

Ausgehend von einem gegebenen Energie-Verlauf wurden mit dem Kontrollsystem Stromrampen für die alte und die neue Ansteuerung berechnet. Durch Abspielen dieser Rampen mit beiden Systemen wurde nun nach Abweichungen der beiden Verläufe gesucht. Diese können sowohl durch Unterschiede bei der Berechnung durch das Kontrollsystem als auch durch die Weiterverarbeitung oder Wiedergabe der Werte im Mikrocontroller verursacht werden. Es wurde für beide Messung die selbe IFC benutzt um Abweichungen z.B. durch den DAC oder dessen Spannungsreferenz zu vermeiden.

Damit zeitliche Abweichungen eine möglichst geringe Rolle spielen, wurde der zu Beginn des Zyklus stattfindende steigende Teil der Rampe untersucht. Die so gewonnenen Messwerte sind in Abbildung 5.1 dargestellt. Der DAC der IFC gibt den Sollstrom in Form einer Referenzspannung aus, hierbei entspricht

<sup>1</sup> Diese gilt bei  $25\ ^\circ\text{C}$  und ohne Berücksichtigung von Alterungseffekten.

eine Spannung von 10 V dem maximalen Strom des jeweiligen Netzgerätes. Auch diese Messung wurde an dem Netzgerät der defokussierenden Sextupole durchgeführt.

Die beiden Kurven verlaufen bis auf einen kleinen Offset von ca. 6 mV gleich. Dies zeigt, dass die Berechnung und Wiedergabe der Rampe keine Fehler aufweist. Der Offset kann auf eine temperaturbedingte Drift der analogen Schaltungselemente der IFC zurückgeführt werden. Auch hat es sich gezeigt, dass das Oszilloskop, welches für die Messungen benutzt wurde, eine schlechte Amplitudenstabilität aufweist.

Dem Signal ist eine 50 Hz-Störung überlagert. Diese ist trotz des zeitlichen Abstands der beiden Messungen synchron, da der die Messungen startende Triggerpuls netzsynchron war (Rampstart-Trigger). Überraschend ist, dass auch der wesentliche Teil der restlichen Störungen bei beiden Messungen synchron verläuft. Dies lässt sich in dem vergrößerten Ausschnitt von Abbildung 5.1 erkennen. Eine Ursache hierfür könnte die Nichtlinearität<sup>2</sup> des DAC sein. Genauere Abschätzung hierüber ist aus den im Datenblatt gegebenen Daten leider nicht möglich.

Hier könnte eine nicht netzsynchrone gestartete Messung Klarheit verschaffen. Auch könnte eine präzisere Messung der Übertragungskennlinie, also der Ausgangsspannung als Funktion des angelegten Digitalwertes, beurteilen lassen, ob die mit der IFC erreichbare Genauigkeit noch zufriedenstellend ist.

---

<sup>2</sup> Die integrale (INL) bzw. differentielle (DNL) Nichtlinearität eines DAC, gibt an, wie groß Abweichungen von der idealen (streng linear) verlaufenden Übertragungskennlinie ist.

---

# Zusammenfassung

---

Ziel dieser Arbeit war es, eine neue Ansteuerung für die Netzgeräte der Hauptmagnetfamilien des Stretcherrings von ELSA zu entwickeln. Für diese Aufgabenstellung wurde eine Konzept erstellt und die darin abgegrenzten Aufgaben auf verschiedene Hardware- und Kommunikations-Ebenen aufgeteilt. Entsprechend dieser Strukturierung wurden mehrere Schaltungen entwickelt.

Für die eingesetzten Mikrocontroller wurde umfangreiche Software geschrieben. Hier wurde neben der Kommunikation über Ethernet und SPI auch das Vermitteln von, auf einem eigens entworfenen Protokoll basierenden, Datenpaketen implementiert. Die Prozessebene wurde für die Berechnung der Rampenstützstellen und die Ansteuerung der IFC programmiert.

Um ein Testen der so aufgebauten Ansteuerung zu ermöglichen, wurde eine weitere Software geschrieben. Auch waren einige Anpassungen am Kontrollsystem der Beschleunigeranlage notwendig.

Mit einer Layout-Software wurden aus den Schaltplänen der einzelnen Module Platinenlayouts entworfen. Insgesamt zehn Platinen wurden mit Bauelementen bestückt und somit fertig gestellt.

Bei ersten Tests der so gefertigten Module wurden durch die eingesetzten schnellen Optokoppler erzeugte Störpulse festgestellt, welche zur fehlerhaften Ansteuerung der IFCs führen können. Dennoch wurden einige Messungen an einem der Netzgeräte durchgeführt. Die Ergebnisse bezüglich Timing und Rampenwiedergabe sind sehr zufriedenstellend.

Um das Auftreten der Störpulse abzustellen, empfiehlt es sich die Optokoppler gegen einen langsameren Typ auszutauschen. Auch sind vor dem dauerhaften Einsatz noch weitere Tests und Messungen angebracht.

Durch Erweiterung der Programmierung der Prozessmodule, könnte in Zukunft die Aufgabe der Luftquadrupole ganz oder teilweise von den regulären Quadrupolen übernommen werden. Auch wäre mit einer Hardware-Erweiterung der Einsatz an anderen Magnetfamilien, wie zum Beispiel den Korrektormagneten, möglich.



---

## HOLEC-IFC

---

Bit	Fehler	Status
0	Ext intl II	-
1	Ext int I	-
2	E.O.	-
3	Öl-Level	-
4	DCCT sat	-
5	Tür	-
6	$U_{\text{Rippel}}$	-
7	Sicherung	-
8	Buchholz	Remote
9	Thyristor Temp	Aux on
10	Trafo. Temp	Ready
11	$I_{\text{DC RMS}}$	MCB on
12	$I_{\text{AC}}$	Spare
13	Phase/Sync	Spare
14	ADC valid	ADC valid
15	Interface available	Interface available

Tabelle A.1: Aufschlüsselung der Fehler- und Statusbits (Kann nach je nach Magnettyp variieren. Gezeigt ist die ursprüngliche Bedeutung.)

		IFC-Input										IFC-Output			
		Strobe	Steuer			Daten							IA <sup>1)</sup>	AV <sup>2)</sup>	Daten
		S	15	14	13	12	11	10	9	8	7..0	15	14	13..0	
Fehler auswählen	•	↑	X	L	L	X	X	X	X	X	X	L	AV	FEHER-Wort	
Fehler auswählen		L	↑	L	L	X	X	X	X	X	X	L	AV	FEHER-Wort	
Status auswählen	•	↑	X	L	H	X	X	X	X	X	X	L	AV	STATUS-Wort	
Status auswählen		L	↑	L	H	X	X	X	X	X	X	L	AV	STATUS-Wort	
(DAC setzen)		L	L	H		<u>DAC-Wert</u>					L	AV	Q'		
ADC trig. + DAC schr.	•	↑	L	H		<u>DAC-Wert</u>					L	AV	<u>ADC-Wert</u> <sup>3)</sup>		
ADC trig. + DAC schr.		L	↑	H		<u>DAC-Wert</u>					L	AV	<u>ADC-Wert</u> <sup>3)</sup>		
ADC trig. + DAC schr.		L	L	↓		<u>DAC-Wert</u>					L	AV	<u>ADC-Wert</u> <sup>3)</sup>		
NOP-Mode	•	H	X	X	X	X	X	X	X	X	L	AV	Q'		
NOP-Mode	•	X	H	H	X	X	X	X	X	X	L	AV	Q'		
Steuerbits schreiben	•	↓	H	L	X	X	A	B	C	D	X	L	AV	Q'	
Steuerbits schreiben		L	↑	L	X	X	A	B	C	D	X	L	AV	Q'	
Steuerbits schreiben		L	H	↓	X	X	A	B	C	D	X	L	AV	Q'	
(Steuerbits schreiben) <sup>4)</sup>		L	H	L	X	X	A	B	C	D	X	L	AV	Q'	

Input-Daten				Steuerleitungen						
A	B	C	D	AUX on	AUX off	MCB on	MCB off	reset faults	release	standby
H	L	H	L	LP	H	H	H	H	H	H
L	H	H	L	H	LP	H	H	H	H	H
H	L	L	H	H	H	LP	H	H	H	H
L	H	L	H	H	H	H	LP	H	H	H
H	H	L	L	H	H	H	H	LP	H	H
L	L	H	H	H	H	H	H	H	LP	H
L	L	L	L	H	H	H	H	H	H	LP
X	X	X	X	H	H	H	H	H	H	H

- H** : high-Zustand (logisch 1)
- L** : low-Zustand (logisch 0)
- ↑ : low-high-Flanke (steigende Flanke)
- ↓ : high-low-Flanke (fallende Flanke)
- X** : irrelevant / indefinit
- Q'** : Zustand der letzten Ausgabe
- : vorzuziehende Modi
- LP** : 100 ms-low-Puls

- 1) Interface-Available : Falls die Versorgungsspannung anliegt = L
- 2) ADC-Valid : Wenn die AD-Konvertierung abgeschlossen ist = H
- 3) Nur gültig, wenn AV = H
- 4) Wirkt erst 100 ms nach der letzter Steuer-Operation

Abbildung A.1: Logiktable der IFC



---

## Schaltungsdokumentation

---

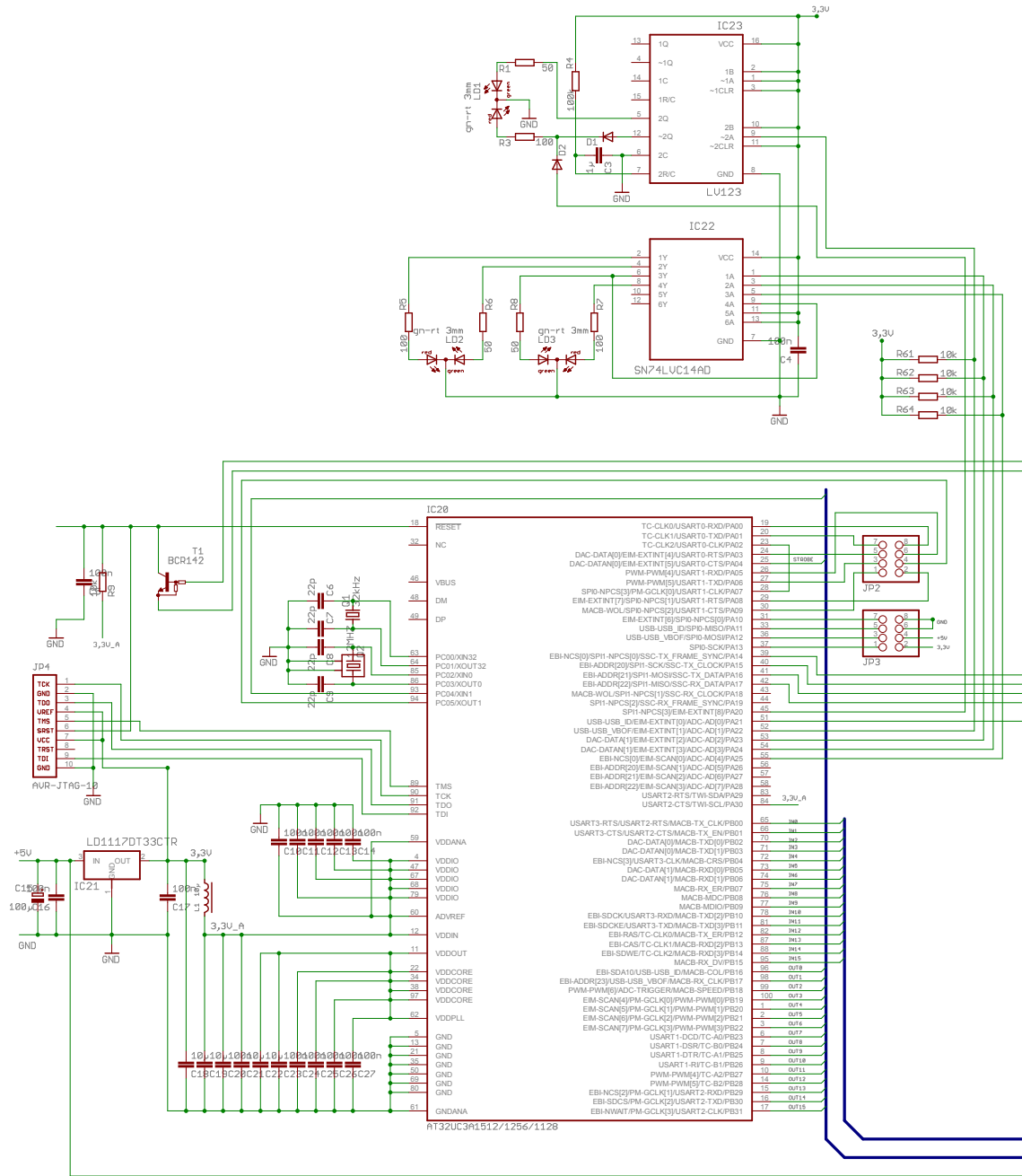


Abbildung B.1: Schaltplan L2 (Teil 1)

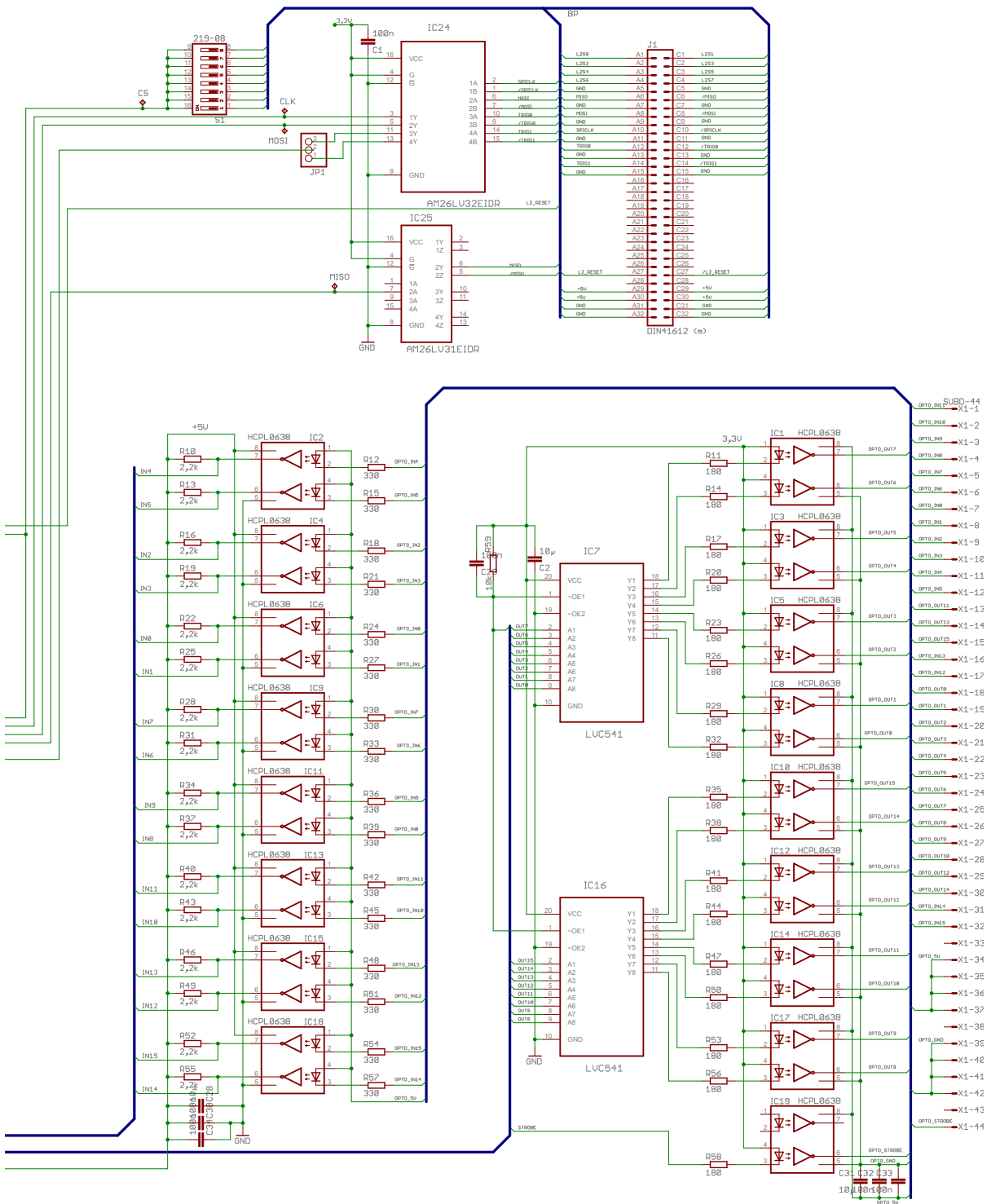


Abbildung B.2: Schaltplan L2 (Teil 2)

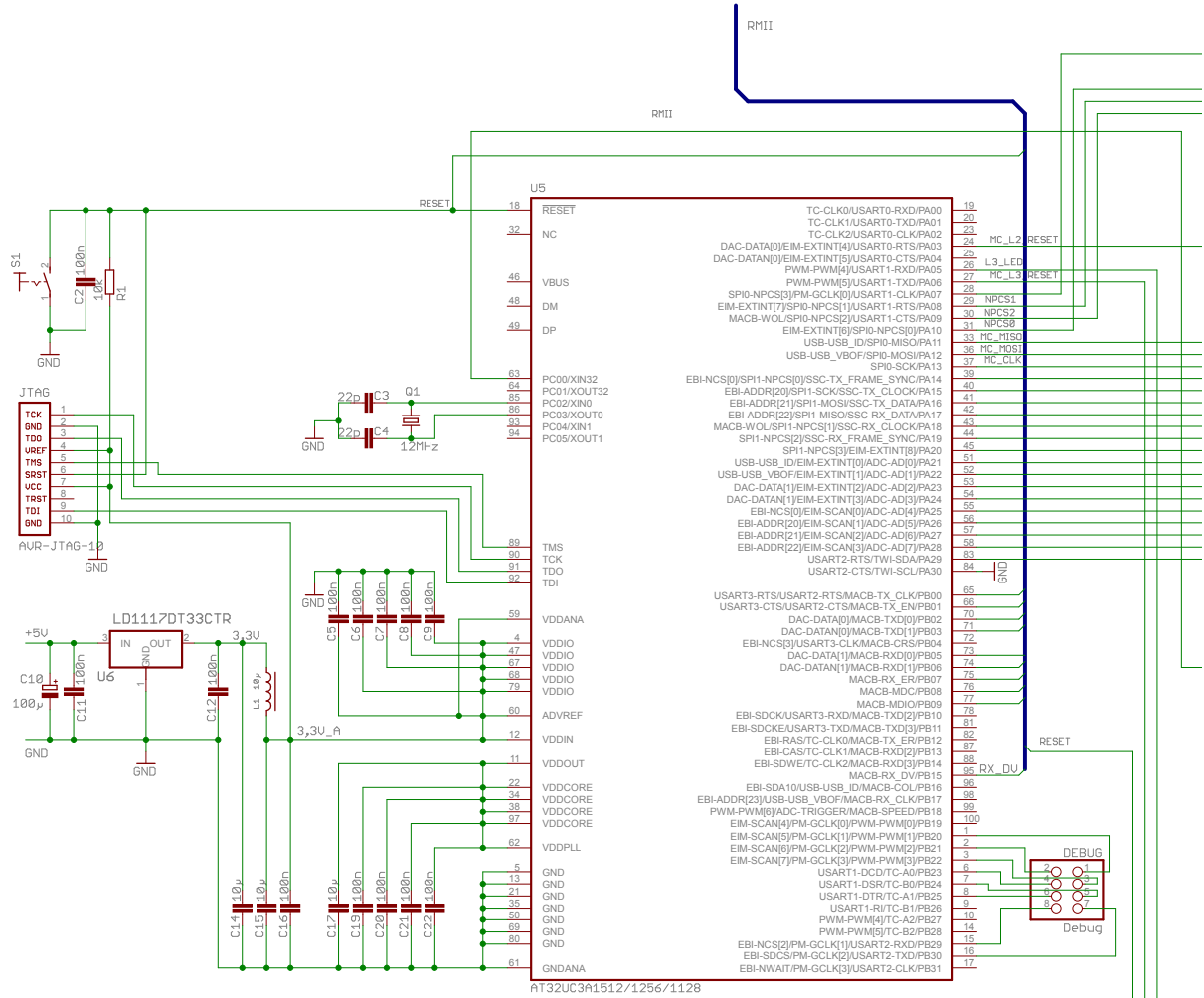


Abbildung B.3: Schaltplan L3 (Teil 1)

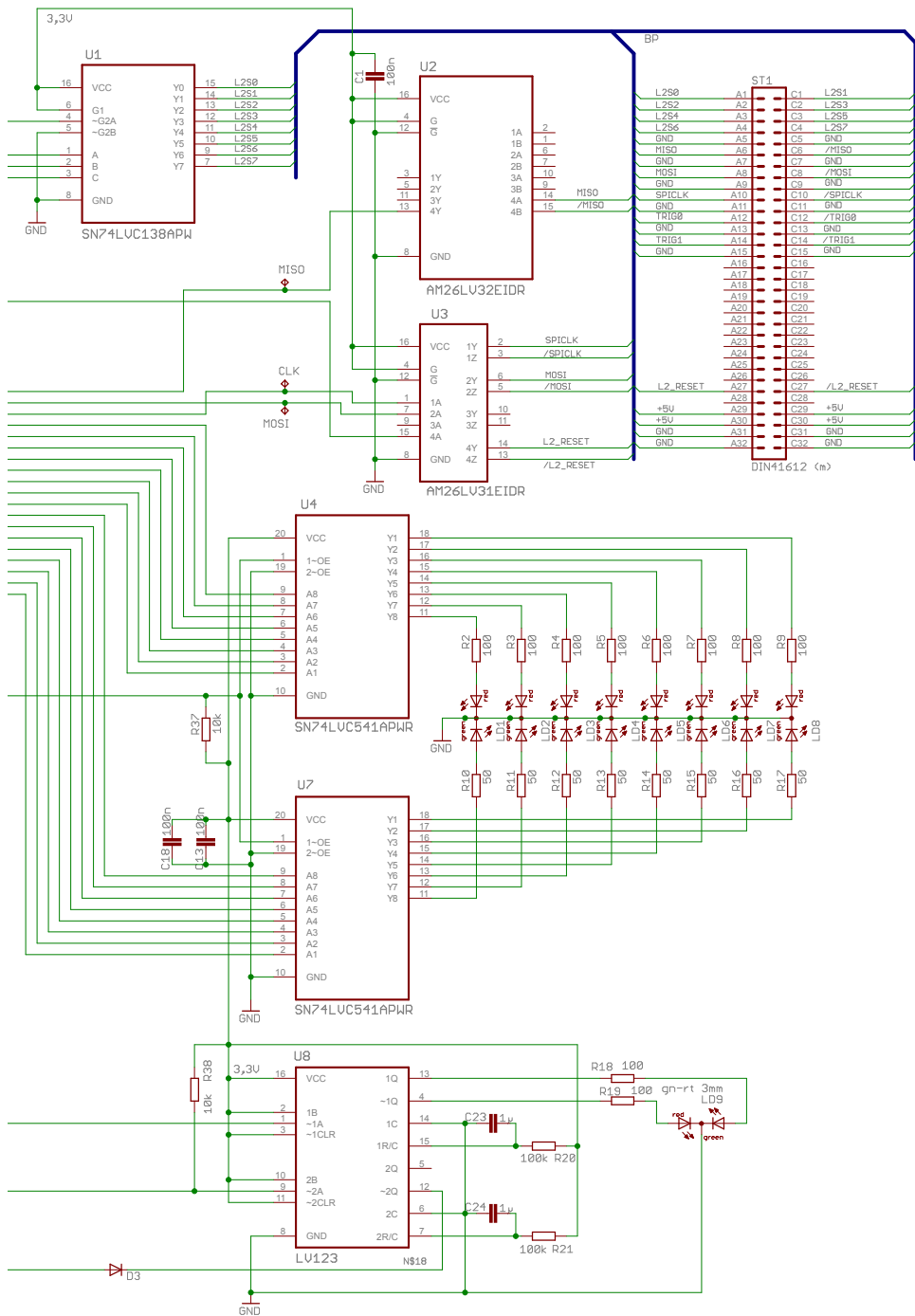


Abbildung B.4: Schaltplan L3 (Teil 2)

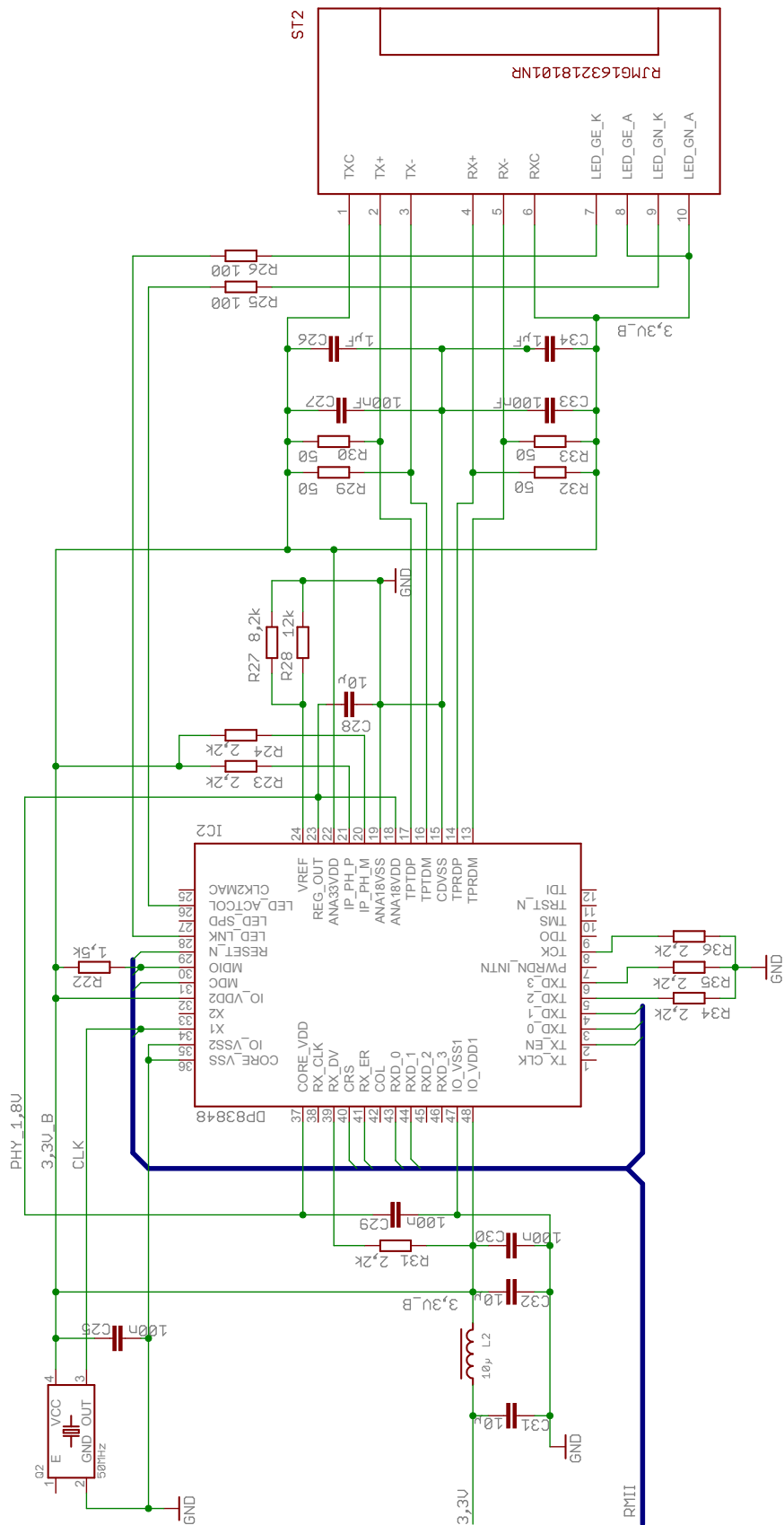


Abbildung B.5: Schaltplan L3 (Teil 3)

Pin	Reihe a	Reihe c
1	CS_0	CS_1
2	CS_2	CS_3
3	CS_4	CS_5
4	CS_6	CS_7
5	Masse	Masse
6	MISO	$\overline{\text{MISO}}$
7	Masse	Masse
8	MOSI	$\overline{\text{MOSI}}$
9	Masse	Masse
10	CLK	$\overline{\text{CLK}}$
11	Masse	Masse
12	TRIGGER_0	$\overline{\text{TRIGGER}_0}$
13	Masse	Masse
14	TRIGGER_1	$\overline{\text{TRIGGER}_1}$
15	Masse	Masse
16	NC	NC
17	NC	NC
18	NC	NC
19	NC	NC
20	NC	NC
21	NC	NC
22	NC	NC
23	NC	NC
24	NC	NC
25	NC	NC
26	NC	NC
27	RESET	$\overline{\text{RESET}}$
28	NC	NC
29	+5 V	+5 V
30	+5 V	+5 V
31	Masse	Masse
32	Masse	Masse

Tabelle B.1: Pinbelegung des Modulsteckers (DIN 41612) (NC = Not connected)

Anschluss IFC	Sub-D 37 (Stecker)	Sub-D 44 (Buchse)	Anschluss L2
STROBE	1	44	opto_Strobe
DAC BIT 8	2	26	opto_out8
DAC BIT 10	3	28	opto_out10
DAC BIT 12	4	29	opto_out12
DAC BIT 14	5	30	opto_out14
DAC BIT 7	6	25	opto_out7
DAC BIT 6	7	24	opto_out6
DAC BIT 5	8	23	opto_out5
DAC BIT 4	9	22	opto_out4
ADC BIT 8	10	4	opto_in8
ADC BIT 10	11	2	opto_in10
ADC BIT 12	12	17	opto_in12
ADC DATA VALID	13	31	opto_in14
ADC BIT 7	14	5	opto_in7
ADC BIT 6	15	6	opto_in6
ADC BIT 5	16	12	opto_in5
ADC BIT 4	17	11	opto_in4
GND	18	39, 40, 41, 42	opto_GND
+5 V, OPTO	19	34, 35, 36, 37	opto_5V
DAC BIT 9	20	27	opto_out9
DAC BIT 11	21	13	opto_out11
DAC BIT 13	22	14	opto_out13
DAC BIT 15 MSB	23	15	opto_out15
DAC BIT 0 LSB	24	18	opto_out0
DAC BIT 1	25	19	opto_out1
DAC BIT 2	26	20	opto_out2
DAC BIT 3	27	21	opto_out3
ADC BIT 9	28	3	opto_in9
ADC BIT 11	29	1	opto_in11
ADC BIT 13 MSB	30	16	opto_in13
INTERFACE AVAILABLE	31	32	opto_in15
ADC BIT 0 LSB	32	7	opto_in0
ADC BIT 1	33	8	opto_in1
ADC BIT 2	34	9	opto_in2
ADC BIT 3	35	10	opto_in3
GND	36	39, 40, 41, 42	opto_GND
+5 V, OPTO	37	-	-

Tabelle B.2: Belegung des Kabel (HOLEC IFC – L2-Modul)



---

## Literatur

---

- [Pro11] D. Proft, Diplomarbeit: Rheinische Friedrich-Wilhelms-Universität Bonn, 2011.
- [Wil96] K. Wille, *Physik der Teilchenbeschleuniger und Synchrotronstrahlungsquellen*, Bd. 2, 1996, ISBN: 9783519130871.
- [Hol] *Dokumentation der HOLEC-Netzgeräte.*
- [Pic95] M. Picard, „Entwurf, Entwicklung und Inbetriebnahme eines verteilten Rechnerkontrollsystems zur Steuerung der Elektronen-Stretcher-Anlage ELSA, unter besonderer Berücksichtigung der Extraktion im Nachbeschleunigungsbetrieb bis 3.5 GeV“, BONN-IR-95-01, PhD Thesis: Rheinische Friedrich-Wilhelms-Universität Bonn, 1995.
- [Gö95] T. Götz, „Entwicklung und Inbetriebnahme eines verteilten Rechnerkontrollsystems zur Steuerung der Elektronen-Stretcher-Anlage ELSA, unter besonderer Berücksichtigung der Anforderungen des Nachbeschleunigungsbetriebes bis 3.5 GeV“, BONN-IR-95-03, PhD Thesis: Rheinische Friedrich-Wilhelms-Universität Bonn, 1995.
- [Atm12] Atmel, *32-Bit Atmel AVR Microcontroller AT32UC3A0512 AT32UC3A0256 AT32UC3A0128 AT32UC3A1512 AT32UC3A1256 AT32UC3A1128*, datasheet, Version Rev. K – 01/12, 2012.
- [Pro] D. Proft, Dissertation in Vorbereitung: Rheinische Friedrich-Wilhelms-Universität Bonn.
- [Int] *Internetworking with TCP/IP*, Pearson Prentice Hall, 1991, ISBN: 0134685059.
- [Nat08] National Semiconductor, *DP83848I PHYTER® - Industrial Temperature, Single Port 10/100 Mb/s Ethernet Physical Layer Transceiver*, datasheet, 2008.
- [TSG10] U. Tietze, C. Schenk und E. Gamm, *Halbleiter-Schaltungstechnik*, Bd. 13, 2010, ISBN: 9783642016219.



---

# Danksagung

---

An dieser Stelle möchte ich mich für die Hilfsbereitschaft und Unterstützung bei der Erstellung dieser Arbeit bedanken.

Herrn Priv.-Doz. Dr. Wolfgang Hillert möchte ich für die Vergabe des interessanten Themas sowie die Betreuung meiner Arbeit danken. Ich möchte Herrn Prof. Dr. Reinhard Beck für die Übernahme des Koreferats danken.

Besonderen Dank möchte ich allen Mitgliedern der ELSA-Arbeitsgruppe für die freundliche Aufnahme in die Gruppe und die ungemene Hilfsbereitschaft aussprechen.

Ich danke Jens-Peter Thiry und Dennis Proft für die viele Hilfe, die sie mir während meiner ganzen Arbeit boten. Besonders bei Problemen mit der Programmierung in C haben sie mir häufig eine Lösung aufgezeigt. Dennis Proft hat darüber hinaus die Anbindung der von mir entwickelten Hardware an das Kontrollsystem ermöglicht.

Herrn Dr. Frank Frommberger danke ich für die Hilfe bei verschiedensten Fragen zu meiner Arbeit.

Für das Anfertigen der Verbindungskabel danke ich Herrn Andreas Offermanns herzlich.

Ich danke Jan Schmidt, Dennis Sauerland, Nikolas Heurich, Jens-Peter Thiry, Dennis Proft und Manuel Schedler für das Korrekturlesen meiner Arbeit.

Meinem Zimmerkollegen Nikolas Heurich danke ich für die vielen aufbauenden Gespräche.

Zum Schluss möchte ich mich bei meiner Familie, insbesondere bei meinen Eltern, für die Unterstützung während meines Studiums bedanken.