
DIPLOMARBEIT

Herr
Axel Schneider

**Entwicklung einer updatefähigen
Embedded-Linux-Hardwareplattform zum
Einsatz in einer speziellen Gerätesteuerung**

Mittweida, 2012

DIPLOMARBEIT

Entwicklung einer updatefähigen Embedded-Linux-Hardwareplattform zum Einsatz in einer speziellen Gerätesteuerung

Autor:

Herr Axel Schneider

Studiengang:

Elektrotechnik

Schwerpunkt Energiesystemtechnik

Seminargruppe:

ET07wE-D

Erstprüfer:

Prof. Dr.-Ing. Thomas Beierlein

Zweitprüfer:

Dipl.-Ing. (FH) Jan Färber

Einreichung:

Mittweida, 17.08.2012

Bibliografische Angaben:

Schneider, Axel: Entwicklung einer updatefähigen Embedded-Linux-Hardwareplattform für den Einsatz in einer speziellen Gerätesteuerung - 2012 – 74 Seiten, 43

Abbildungen, 13 Tabellen, 3 Anlagen , Mittweida, Hochschule Mittweida (FH),

University of Applied Sciences, Fakultät Elektro- und Informationstechnik

Diplomarbeit, 2012

Referat:

Das Projekt „Pfeifen-Orgel mit dynamischer Stimmung“ ist ein Steuerungssystem zur Verbesserung der Klangqualität einer Orgel. Das System besteht aus dezentralen Elementen und einer zentralen Steuerung.

Diese Arbeit befasst sich mit der Entwicklung der zentralen Einheit, der Zentralen Akteur-Steuerung. Ihre Aufgabe umfasst grundlegend die Mikrocontroller gestützte Datenverarbeitung und Kommunikation über spezielle, im Projekt benötigte Peripherie. Für die an diese Arbeit angrenzende Entwicklung der Steuerungssoftware, verfügt die Hardwareplattform über ein angepasstes Embedded Linux.

Inhaltsverzeichnis

Inhaltsverzeichnis	I
Tabellenverzeichnis	III
Abkürzungsverzeichnis	IV
1 Einleitung	1
1.1 Motivation	1
1.2 Aufbau der Arbeit.....	2
2 Rolle der ZAS im Gesamtprojekt	3
2.1 Projektübersicht.....	3
2.2 Motivation des Projekts.....	4
2.3 Theorie und mechanischer Ansatz.....	5
2.4 Spezifikation der Zentralen Aktor Steuerung.....	5
3 Grundlagen und Stand der Technik	7
3.1 Konzept der Elektronikentwicklung.....	7
3.2 EMV gerechter Leiterplattenentwurf.....	10
3.2.1 Feldgebundene EMV.....	11
3.2.2 Leitungsgebundene EMV.....	12
3.2.3 Signalreflexionen.....	14
3.2.4 Signalterminierung.....	15
3.2.4.1 Serienterminierung.....	15
3.2.4.2 Parallelterminierung.....	16
3.3 CAN-Bus.....	16
3.4 Der AT91SAM9G45 Mikrocontroller.....	18
3.4.1 SPI - Serial Peripheral Interface.....	19
3.4.2 I ² C-Bus - Inter-Integrated Circuit Datenbus.....	20
3.4.3 Ethernet.....	20
3.4.4 USB.....	21
3.4.5 UART.....	21
3.4.6 Echtzeituhr.....	22
3.5 Das Betriebssystem Linux in eingebetteten Systemen.....	22
3.5.1 Der Linux Kernel.....	23
3.5.2 Treiber.....	24
3.5.3 In-Circuit Buildroot.....	24
3.6 Entwicklungsumgebung.....	24
4 Hardwareentwicklung	26
4.1 Entwurf der Baugruppen.....	26
4.1.1 Bedienschnittstelle.....	26
4.1.1.1 Tastenmatrix.....	26

4.1.1.2 Drehgeber	27
4.1.1.3 Display.....	27
4.1.2 Energieversorgung.....	30
4.1.2.1 Bestimmung der Stromaufnahme aller Komponenten...31	
4.1.2.2 Dimensionierung der Gleichspannungsregler.....	33
4.1.2.3 Display Beleuchtung.....	34
4.1.2.4 Dimensionierung des Boardnetzteils.....	35
4.1.3 Steuerung der Displaybeleuchtung und des Relais.....	39
4.1.4 Umsetzung des potentialfreien Kontakts.....	41
4.1.4.1 Steuerstromkreis.....	41
4.1.4.2 Lastseitige Umsetzung.....	41
4.1.5 Umsetzung der Schnittstellen.....	43
4.1.5.1 UART Schnittstelle.....	43
4.1.5.2 Mikrofoneingang.....	43
4.1.5.3 CAN-Bus.....	44
4.1.5.4 Ethernet.....	45
4.1.5.5 MIDI.....	45
4.1.5.6 USB Host Ports.....	46
4.1.5.7 RS232.....	46
4.1.5.8 Real Time Clock M41T82R.....	47
4.1.5.9 LCD-Steckverbinder.....	48
4.1.5.10 Zusätzliche Peripherie.....	48
4.2 Leiterplattenentwurf.....	50
4.2.1 Planung.....	51
4.2.2 Umsetzung von EMV Designrichtlinien.....	52
4.2.2.1 Allgemeines EMV Design.....	52
4.2.2.2 VersorgungsspannungsfILTER.....	53
4.2.3 PCB-Design des Ethernets.....	53
4.2.4 PCB-Design des Mikrofoneingangs.....	56
4.3 Hardwareprototyp Übersicht	57
4.3.1 Eigenschaften.....	59
4.3.2 Hardwareinbetriebnahme und Funktionstest.....	60
4.4 EMV-Test.....	60
4.4.1 Feldgekoppelte Störeinwirkung.....	60
4.4.2 Feldgekoppelte Störaussendung nach DIN EN 55022.....	61
4.4.3 Vorbetrachtung.....	61
4.4.4 Messung.....	61
4.4.5 Auswertung.....	63
4.4.6 Ausblick.....	63
5 Softwarekonfiguration.....	64
5.1 Spezifikation	64
5.2 Boardspezifische Implementation und Tests.....	65
5.2.1 Relaisansteuerung.....	65

5.2.2 UART und Terminalkommunikation.....	65
5.2.3 Eingabeelemente.....	66
5.2.4 Display Ansteuerung.....	66
5.2.5 Ethernet.....	67
5.2.6 USB.....	67
5.2.7 Echtzeituhr.....	67
5.2.8 MIDI.....	68
5.2.9 SD-Speicherkarteninterface.....	69
5.2.10 CAN Implementation.....	69
6 Zusammenfassung und Ausblick.....	73
6.1 Zusammenfassung.....	73
6.2 Ausblick.....	74
Literaturverzeichnis.....	75
Anlagen.....	i
Anlage A – AT91SAM9G45 Pinbelegung der ZAS.....	i
Anlage B – ZAS-Schaltplanauszug: Spannungsversorgung.....	v
Anlage C – Top Layer Ansicht des PCB-Designs der ZAS.....	vi
Eidesstattliche Erklärung.....	vii

Tabellenverzeichnis

Tabelle 1: Verwendung der AT91SAM9G45 Peripherie.....	19
Tabelle 2: Verwendete Messgeräte.....	25
Tabelle 3: Gray-Code im Vergleich zum Binär-Code.....	27
Tabelle 4: Zusammenfassung Displayvergleich.....	29
Tabelle 5: Stromaufnahme der ZAS Komponenten.....	31
Tabelle 6: Strombelastung der Spannungsquellen auf der ZAS.....	36
Tabelle 7: Pinbelegung des CAN-Bus Steckers.....	44
Tabelle 8: Steckerbelegung des RS232.....	47
Tabelle 9: Prioritäten in der Platzierung und Routing.....	51
Tabelle 10: Maße des ZAS Prototyps.....	59
Tabelle 11: Betriebstemperatur-Bereich des ZAS V1.0 Prototypen.....	59
Tabelle 12: Trägerfrequenzen der Peripherie.....	61
Tabelle 13: Kritische Frequenzen feldgekoppelter Störaussendung.....	62

Abkürzungsverzeichnis

AC97	Audio Codec 97
CAN	Controller Area Network
CAS	Client Aktor-Steuerung
CoM	Computer-on-Module
DMA	Direct Memory Access
EDA	Electronic Design Automation
EMV	Elektromagnetische Verträglichkeit
GND	Ground
GPIO	General Purpose Input / Output
I/O	Input / Output
I ² C	Inter-Integrated Circuit
IC	Integrated Circuit
LCD	Liquid Crystal Display
LED	Light Emitting Diode
HMT	Hermoder Tuning
MAC	Media Access Control
MIDI	Musical Instrument Digital Interface
OS	Operating System
RAM	Random Access Memory
RTC	Real Time Clock
SPI	Serial Peripheral Interface
STN	Super Twisted Nematic
TFT	Thin-film transistor
TWI	Two Wire Interface
UART	Universal Asynchronous Receiver Transmitter
USB	Universal Serial Bus
VCC	Versorgungsspannung
ZAS	Zentrale Aktor-Steuerung

1 Einleitung

Eingebettete Systeme finden immer mehr Einzug in unseren Alltag. Ob als Verbesserer der Verkehrssicherheit im Auto, als Steuerung eines intelligenten Hauses, oder als integrierter Decoder im Flachbildfernseher. So ist es nicht verwunderlich, dass sie auch in einer Orgel dafür sorgen können, die Qualität der gespielten Musik zu verbessern, ohne dabei ihren klanglichen Charakter zu verschlechtern.

Beim gleichzeitigen Spielen mehrerer Töne, ergeben sich bei genauerem Hinhören nicht die rein harmonischen Klangkombinationen, die erwartet werden. Das Projekt „Pfeifen-Orgel mit dynamischer Stimmung“, gefördert durch: Bundesministerium für Wirtschaft und Technologie aufgrund eines Beschlusses des Deutschen Bundestags, widmet sich der Anpassung dieser Töne, indem Pfeifen dynamisch während des Spielens mit einem speziellen System umgestimmt werden. Die Steuerung dieses Stimmprozesses soll zentral, durch ein Gerät auf Embedded Linux Basis erfolgen.

Diese Arbeit befasst sich mit Entwurf und Entwicklung dieses Geräts, der Zentralen Aktor-Steuerung (ZAS). Ihre Aufgabe umfasst grundlegend die Mikrocontroller-gestützte Kommunikation und Datenverarbeitung spezieller im Projekt benötigter Peripherie, die teilweise extern angeschlossen, teilweise im Gerät integriert sind. Die Entwicklung der ZAS unterteilt sich in die Hardwareplattform inklusive Betriebssystem und die Entwicklung der ZAS Steuerungssoftware. Die Steuerungssoftware wird getrennt von dieser Diplomarbeit entwickelt.

1.1 Motivation

Ziel dieser Arbeit ist die Neuentwicklung der ZAS mit Methoden des Hardware-Designs und Software-Engineerings von einer vorgegebenen Spezifikation bis zum verwendbaren Prototypen durchzuführen. Dem Nutzer sollen maßgebende Entscheidungen, die im Laufe dieses Prozesses getroffen wurden, aufgezeigt werden. Dazu gehören unter anderem die Dimensionierung von Bauelementen, die elektromagnetische Verträglichkeit und Konfiguration eines Embedded Linux Betriebssystems.

Die aus dieser Diplomarbeit resultierende Hardwareplattform ist eine Basis zur Programmierung von Steuerungssoftware für ein komplexes verteiltes System in einer Orgel. Eine Herausforderung war, die zu entwickelnde Gerätesteuerung in einem Umfeld bereits vorhandener Hard- und Software zu integrieren und dabei als deren Basis zu dienen. Der Nutzen dieser Arbeit äußert sich im entstandenen Prototypen. So ließ sich eine Umsetzung der beschriebenen Methoden in die Praxis darstellen und es konnte ein Nachweis der Funktion der Baugruppen auf Basis der Elektronikentwicklung erarbeitet werden. Weiterhin ermöglicht dieses Projekt eine Vielzahl unterschiedlicher Funktionen auf einer Leiterplatte.

Die entstandene Hardwareplattform bietet gemäß ihren Anforderungen auch Schnittstellen und Funktionen, die mit auf dem Markt befindlichen Geräten nicht abgedeckt werden können.

1.2 Aufbau der Arbeit

Zu Beginn erfolgt im Kapitel 2 eine Einführung in das Gesamtprojekt. Sie dient der Einordnung dieser Arbeit und beschreibt relevante Schnittstellen zu anderen Elementen im Projekt.

In Kapitel 3 werden die zu Grunde liegenden theoretischen Aspekte der Entwicklung eines eingebetteten, elektronischen Geräts betrachtet. Sie dienen als Bezug für angewandte Methoden und Schnittstellen in den darauf folgenden Kapiteln.

Kapitel 4 beschäftigt sich mit der Hardwareentwicklung der Zentralen-Aktor-Steuerung von der Spezifikation bis zum betriebsbereiten Prototypen.

Um die Grundfunktionen des Prototypen sicher zu stellen, wird in Kapitel 5 die Implementation des angepassten Embedded Linux Betriebssystems behandelt.

In Kapitel 6 erfolgt eine Zusammenfassung dieser Arbeit und den entwickelten Prototypen ZAS V1.0, sowie ein Ausblick über Ansätze zur Weiterentwicklung.

2 Rolle der ZAS im Gesamtprojekt

Dieses Kapitel soll das dieser Arbeit zu Grunde liegende Projekt umreißen, um eine klare Abgrenzung zu anderen Bestandteilen des Projekts zu erreichen. Gleichzeitig stellt es die Schnittstellen zu anderen Elementen im System heraus, mit denen es interagiert.

2.1 Projektübersicht

Der Aufbau des Orgelprototyps wird in Abbildung 1 dargestellt. Für die reine Funktion der Orgel ohne Umstimmsteuerung werden zunächst per Tastendruck auf den Bedienelementen, bestehend aus Manual und Pedal, die zu aktivierenden Töne durch eine orgelinterne Elektronik, die Orgelsteuerung, ausgewertet. Je nach gedrückter Taste wird nun ein von der Orgelsteuerung bestimmtes Elektroventil aktiviert und der entsprechende Ton dessen zugehöriger Pfeife erzeugt.

Neben diesem Normalbetrieb, ist auch eine Funktion der Orgel ohne eingeschaltete Umstimmsteuerung erlaubt.

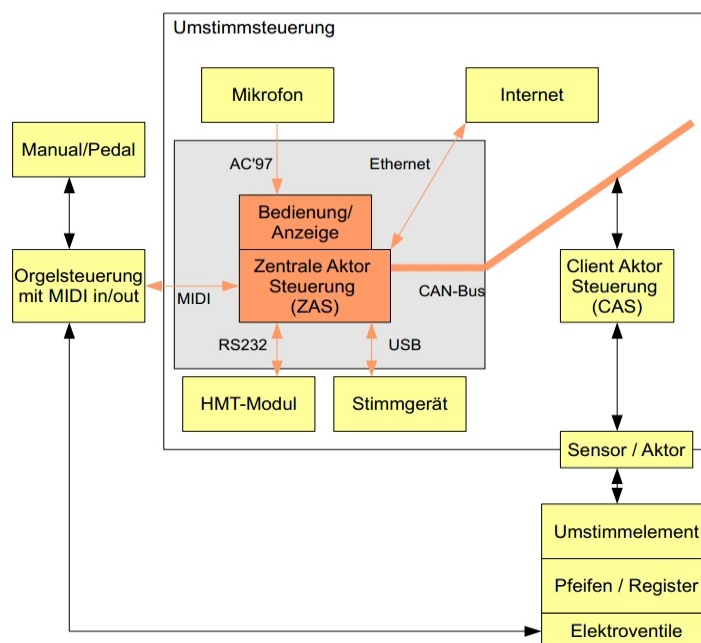


Abbildung 1: Gesamtprojekt Übersicht

Zieht man nun die Umstimmsteuerung hinzu, wird diese an die MIDI-Schnittstelle der Orgelsteuerung angeschlossen. Bei MIDI handelt es sich um eine digitale Schnittstelle für Datenübertragung in der Musik, auf welche in Kapitel 3 näher eingegangen wird. Über diese Schnittstelle erhält die Umstimmsteuerung unter anderem die Tonhöhe der aktuell gespielten Töne. Die ZAS nimmt die Töne per MIDI entgegen und leitet die Toninformationen weiter an ein Hermode Tuning Modul (HMT-Modul). Das HMT-Modul ist ein Blackbox-Modul, was über einen patentierten Algorithmus die aktuell benötigte Umstimmung berechnet und die Ergebnisse an die ZAS zurück sendet. Diese Kommunikation findet über eine serielle RS232 Schnittstelle statt. Dabei bereitet die ZAS die Informationen über gespielte Töne und notwendige Umstimmung auf und verteilt diese über den CAN-Bus an die Client-Aktor-Steuerungen (CAS). Sie setzen das Anfahren der Aktorpositionen zur Ton-Umstimmung um. Somit ergibt sich ein Verteiltes System. Die zentrale Rolle verbleibt bei der ZAS.

Bevor der beschriebene Umstimmprozess stattfinden kann, werden die CAS durch die ZAS mit Hilfe eines Stimmgeräts parametrieren. Beide, Aktorstellwerte und Tonhöhe, verhalten sich nicht linear zueinander. Nur durch das Parametrieren können die Aktoren entsprechend gestellt werden, um den gespielten Ton korrekt anzupassen.

2.2 Motivation des Projekts

Die Pfeifenorgel als Tasteninstrument besteht aus einer variablen Zahl an Pfeifen. Die Stimmung der Orgelpfeifen in diesem Projekt orientiert sich am Tonsystem der westlichen Musik, was dazu führt, dass beim Orgelbau Kompromisse eingegangen werden müssen: „Unser Tonsystem mit seinen zwölf Tonschritten kann bei Tasteninstrumenten leider nicht in allen Tonarten mit den reinen Quinten und Terzen der Naturtonreihe übereinstimmen, da sonst viele zusätzliche Tasten notwendig wären.“ [1]. Beim gleichzeitigen Spielen mehrerer Töne, kann man diese Diskrepanz also hören. Da der Mensch, je nach Tonhöhe einige Zeit benötigt, um einen Ton bewusst aufzunehmen, kann innerhalb dieses Zeitfensters der Ton manipuliert werden. Letzteres hat den Hintergrund, dass einerseits hörbare Tonveränderungen als unangenehm empfunden werden, und andererseits eine elektronisch gestützte Manipulation innerhalb einer Orgel nicht von jeder Audienz akzeptiert wird.

Zusammenfassend ist das Ziel des Projekts „Pfeifen-Orgel mit dynamischer Stimmung“ die Verstimmung in den Quinten und Terzen während des Spielens innerhalb eines definierten Zeitfensters zu korrigieren.

2.3 Theorie und mechanischer Ansatz

Orgelpfeifen sind komplex geformte Hohlkörper aus Holz oder einer Legierung aus Metall. Ihr Aufbau ergibt sich aus einem definierten Ton und der angestrebten Klangfarbe. Werden sie von Luft durchströmt, wird diese in Schwingung versetzt. Für die Frequenz des entstehenden Grundtons spielt besonders die Länge der Pfeife, aber auch die innere Form eine Rolle.

Zur Manipulation der Tonfrequenz ergeben sich für das Projekt drei Mechanismen: Die Variation der Klangkörperlänge durch Herein- und Herausfahren eines Körpers in die Pfeife, die Veränderung des Ausströmverhaltens der Luft durch zwei bewegliche Metallklappen, und bei kleineren Pfeifen das Überdecken mit einer Metallklappe. Die drei Mechanismen verändern die effektive Länge der Pfeife im Bezug auf die Tonerzeugung. Sie ermöglichen eine kontrollierbare Frequenzänderung des Tons, um die Diskrepanzen der Tonreihen auszugleichen.

Die musikalische Fachbezeichnung für feine Unterscheidungen auch innerhalb des Definitionsbereichs eines Tons werden in Cent angegeben. Cent ist eine logarithmische Maßeinheit für Intervalle in der Musik. Dies erleichtert die Bestimmung der Intensität der Umstimmung.

2.4 Spezifikation der Zentralen Aktor Steuerung

Die ZAS ist ein Knotenpunkt im verteilten System zur zentralen Datenverarbeitung und Verteilung durch an sie angeschlossene, spezielle Hardware. Sie realisiert eine Bedienschnittstelle für Organist, Orgelbauer und Entwickler um Prozesse der Umstimmsteuerung beobachten zu können. Die Entwicklung basiert auf einem Computer-on-Module von In-Circuit mit einem Atmel AT91SAM9G45 Mikrocontroller.

Der Prototyp verfügt über folgende Schnittstellen:

- Zwei USB-Host Schnittstellen zum Anschluss eines Stimmgeräts und zum Anschluss weiterer USB Geräte, wie eine QWERTZ-Tastatur.
- Eine RS232 Schnittstelle wird für die Kommunikation mit dem HMT-Modul benötigt.
- Über eine MIDI-Schnittstelle mit MIDI IN und MIDI OUT werden Noteninformationen von der Orgelsteuerung angenommen.
- Ein CAN-Bus Interface zur Kommunikation mit den CAS, mit Stromversorgung für 78 CAN-Bus Teilnehmer.
- Das Ethernet nach IEEE 802.3 ermöglicht eine Fernwartung.
- Damit die ZAS gegebenenfalls die gesamte Umstimmsteuerung deaktivieren kann, soll sie über einen potentialfreien Kontakt für 230 Volt Wechselspannung mit mindestens 0,5 Ampere Belastbarkeit verfügen.
- Über ein Mikrofon mit 3,5 mm Klinke Anschluss soll in Zukunft auf die Verwendung eines externen Stimmgeräts verzichtet werden können.
- Für das Speichern von Logfiles und zum einfachen Booten des Embedded Linux von einer Speicherkarte, ist die Verwendung eines Speicherkarteninterfaces angedacht.
- Damit zeitbezogene Anwendungen sinnvoll genutzt werden können, ist die Integration einer Batterie gestützten Echtzeituhr vorgesehen.
- Damit eine Eingabe in das Embedded Linux Betriebssystem erfolgen kann, soll eine UART an Stifteleisten herausgeführt werden.

3 Grundlagen und Stand der Technik

Die folgenden Unterpunkte befassen sich mit theoretischen Grundlagen der Hardwareentwicklung des eingebetteten Systems Zentrale Aktor-Steuerung.

3.1 Konzept der Elektronikentwicklung

Im Laufe einer Geräteentwicklung müssen Entscheidungen getroffen werden, welche Einfluss auf spätere Entwicklungsstadien haben. Fehler im Entwurf, Schaltplan, oder im Leiterplattendesign können im Prototyp schwierig oder nicht korrigierbare Fehler zur Folge haben und komplette Geräte unbrauchbar machen, denn je früher ein Fehler in der Entwicklung auftritt, desto größer sind die Folgen im Prototyp und in der späteren möglichen Serienproduktion. Deshalb wird in diesem Entwurf großen Wert auf die Dokumentation aufgetretener Probleme gelegt, um gegebenenfalls in neuen Hardwarerevisionen auf diese Einfluss zu nehmen.

Der Entwurf erfolgt nach dem Top-Down-Bottom-Up-Entwurfsprinzip.

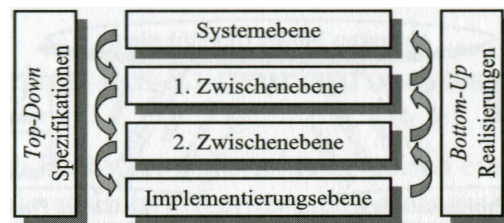


Abbildung 2: Top-Down-Bottom-Up Entwurf [2] S. 59

Die Spezifikationsphase erfolgt über die in Kapitel 2 beschriebene Systemebene und der Einordnung der ZAS, über die in diesem Abschnitt folgenden Abstraktionsebenen, bis hin zur Implementationsebene in Kapitel 3. Die Realisierung beginnt in Kapitel 4 mit der Umsetzung der Hardware der ZAS entsprechend der Abstraktionsebenen. Kapitel 5 enthält neben der Realisierung der Betriebssystemkonfiguration zugleich die Prüfung der Hardware mit Hilfe des Betriebssystems.

Dazu wird zunächst ein abstraktes Modell im Sinne der in Kapitel 2 erfolgten Analyse und Spezifikation des Geräts, seines Umfeldes und der benötigten Hardware erstellt.

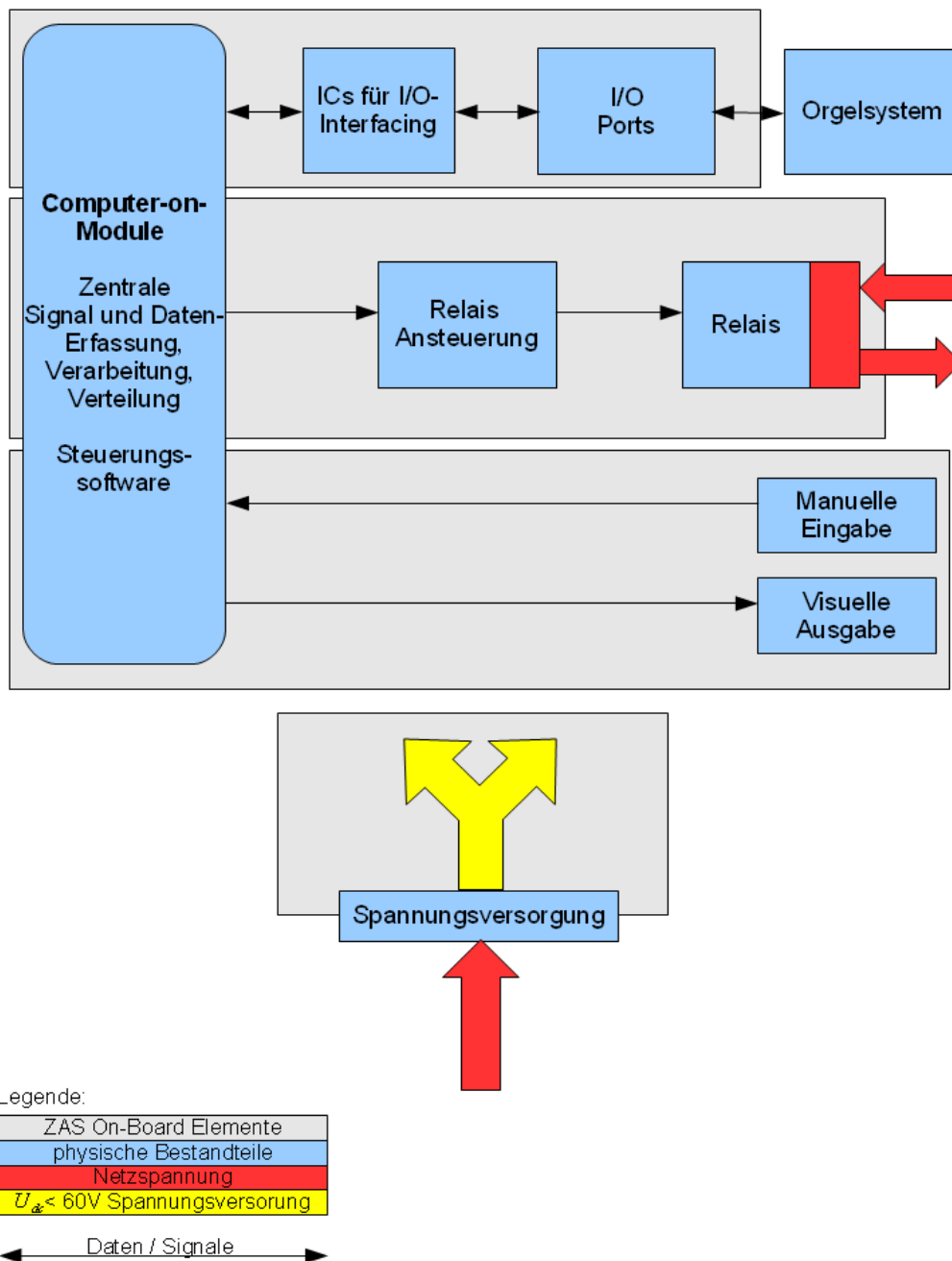
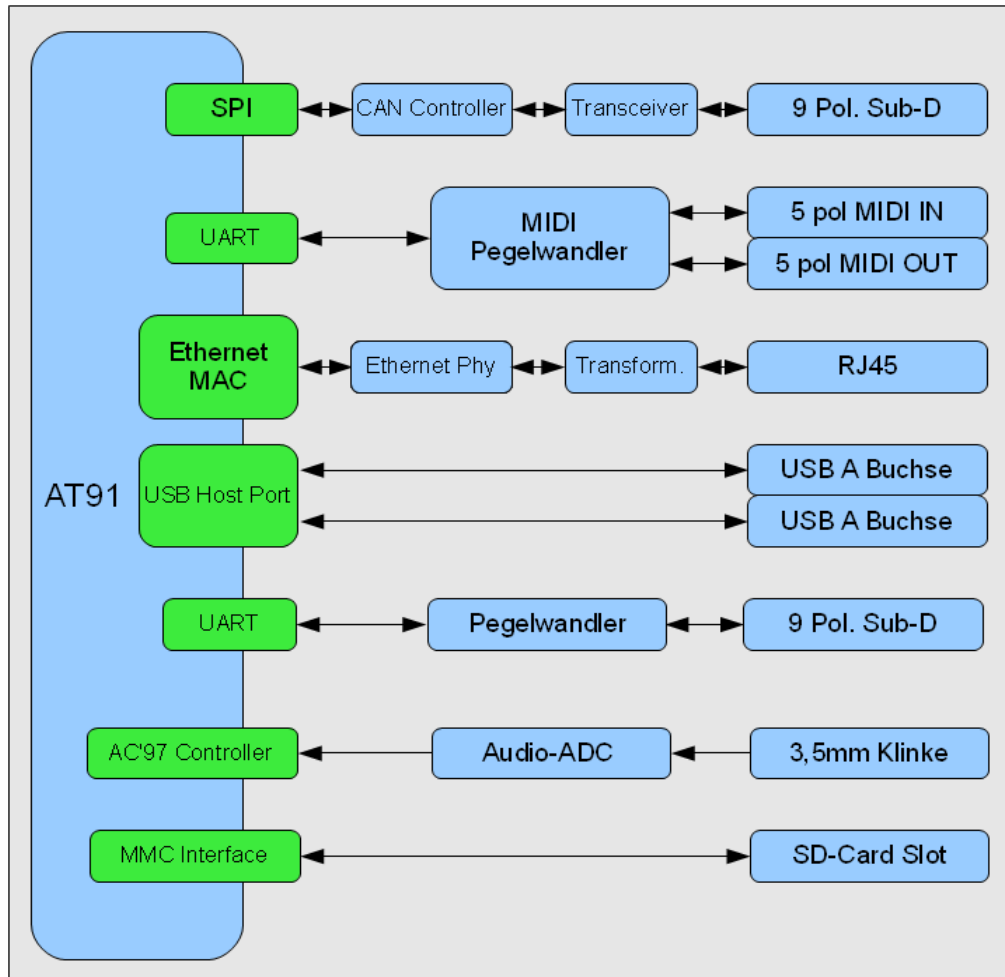


Abbildung 3: 1. Zwischenebene: Abstraktes Modell der ZAS Hardware

Die Zentrale Einheit ist ein Computer-on-Module, von dem aus die zentrale Kommunikation mit dem restlichen System erfolgt. Das Relais stellt in der Abbildung 3 den schaltbaren Kontakt für Netzspannung dar, der zum Ein- und Ausschalten von externen Geräten dienen soll. Er ist losgelöst von der sonstigen Peripherie aufgeführt, weil es das einzige Element im System ist, das Netzspannung führt und gesondert behandelt werden muss. Ausgehend von dieser Abstraktion, können nun in die grau unterteilten Bereiche genauer ausgearbeitet werden und in eine weniger abstrahierte

Ebene überführt werden. So hat sich herausgestellt, welche Peripherie des AT91-SAM9G45 für welchen Zweck geeignet ist. Über die Spezifikation ist somit die Funktionalität und die Anschlüsse der Hardwareschnittstellen genauer, jedoch immer noch abstrakt definiert worden.



Legende:

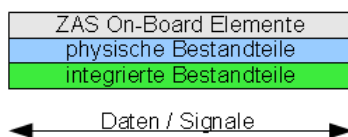


Abbildung 4: 2 Zwischenebene: Abstraktion der ZAS Hardwareschnittstellen

Die Umsetzung der Benutzerschnittstelle wurde, bis auf das Display, zu großen Teilen über GPIOs umgesetzt. Das Display und eine Status-LED dienen zur visuellen

Ausgabe, während die Tastenmatrix und der Drehgeber als Eingabegeräte ausgeführt sind. Abbildung 5 stellt die Benutzerschnittstelle übersichtlich dar.

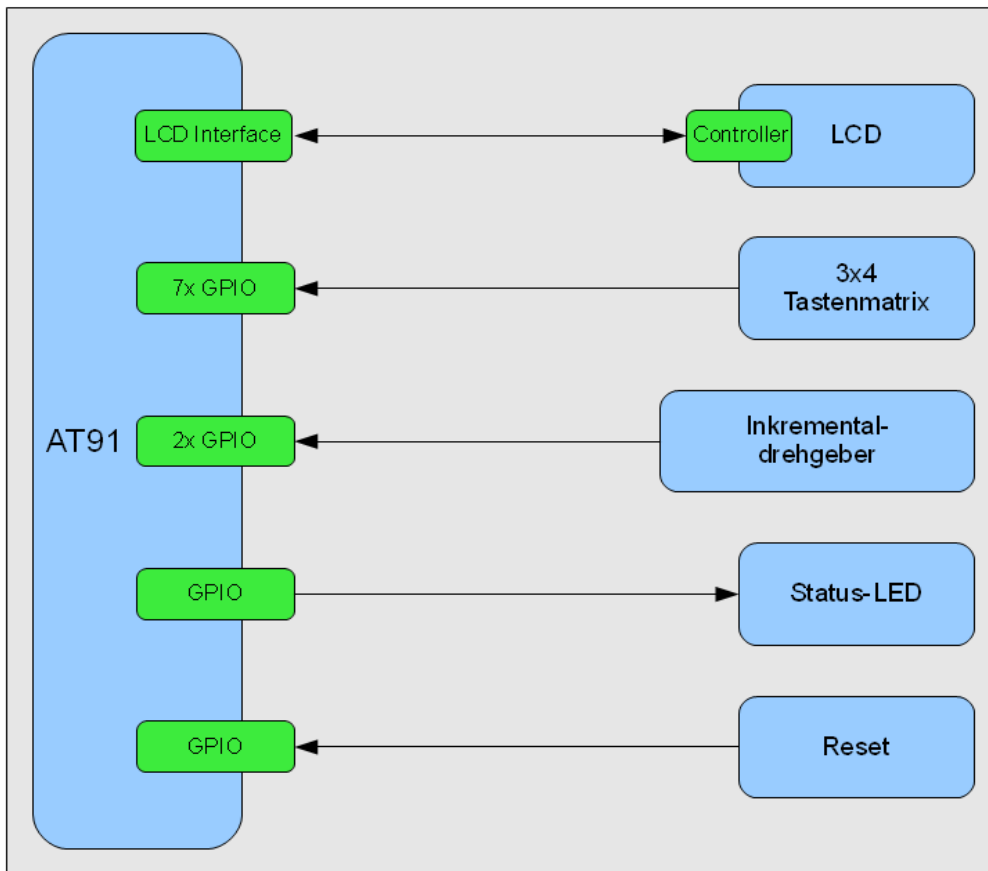


Abbildung 5: 2. Zwischenebene: Abstraktion der ZAS Benutzerschnittstelle

Die Legende von Abbildung 5 ist äquivalent zu Abbildung 4. Auf Basis dieses Modells kann die Auswahl und Dimensionierung der Bauelemente durchgeführt werden.

3.2 EMV gerechter Leiterplattenentwurf

Mit zunehmender Integrationsdichte und kleiner werdenden Schaltzeiten, nimmt die elektromagnetische Verträglichkeit eine feste Größe in der Entwicklung digitaler Schaltungen ein. Die Auswirkungen von vernachlässigten EMV-Designrichtlinien zeigen sich erst nach der Entwurfsphase durch Messungen an der Leiterplatte oder im Prototyping [3] S. 48. Elektromagnetische Emissionen und Immissionen können zu Fehlfunktionen der eigenen oder externer Hardware führen, z.B. galvanisch eingekoppelte Oberwellen des Netzteils, oder Verlust der Signalintegrität bei empfindlichen Baugruppen durch Übersprechen innerhalb des Layouts. Die EMV ist ein eigenes

Fachgebiet mit teilweise schwer kontrollierbaren Einflüssen. Deshalb ist ein EMV-gerechter Leiterplattenentwurf bereits in der Planungsphase zu berücksichtigen und konsequent im PCB-Design umzusetzen, um potentielle Problemstellen kontrollieren oder gar beseitigen zu können.

In der Theorie wird eine ideale Umsetzung des Hardwaredesigns angestrebt. Im Kapitel drei wird sich aber herausstellen, dass projektinterne Absprachen zu Details der Umsetzung des Hardwaredesigns die Umsetzung in der Praxis beeinflussen werden. Der Prototyp wird nach EN 55022 zur Einhaltung der Grenzwerte für gestrahlte Störaussendungen und mit einem Bestrahlungsgerät zur Bestimmung der Störfestigkeit gegen hochfrequente elektromagnetische Felder getestet.

Die folgenden Abschnitte befassen sich mit den theoretischen Grundlagen zur Erfüllung dieser Anforderungen.

3.2.1 Feldgebundene EMV

Das Leiterplattendesign ist ein maßgebender Faktor für feldgebundene Ein- und Abstrahlung eines elektronischen Geräts. Der Leitsatz, „Leitungen so kurz wie möglich halten“, [3] S. 561 gilt für nahezu jede Übertragungsstrecke, denn feldbezogene Kopplungen sind proportional zur Leiterbahnlänge. Demnach sind besonders lange Leiterbahnen zu vermeiden.

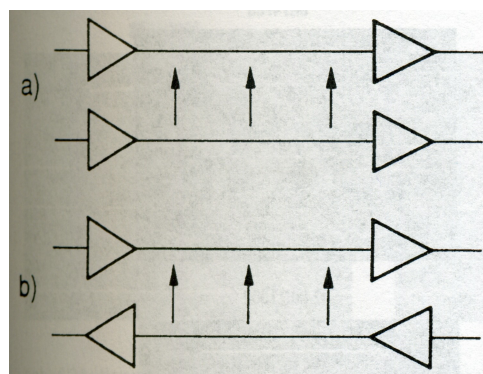


Abbildung 6: a) paralleles Übersprechen b) antiparalleles Übersprechen [3] S.529

Während des Leiterbahnroutings ist beim parallelen Verlegen von Signal-Leiterbahnen darauf zu achten, dass sie nur so dicht wie nötig beieinander liegen und antiparallele Signale bestenfalls nicht nebeneinander geroutet werden: „Während beim parallelen Übersprechen zwischen dem Ausgang der Quelle und dem Eingang der Sen-

ke die Summe von Leitungs- und Kopplungsimpedanz wirkt, wird beim antiparallelen Übersprechen der Ausgang der Quelle mit dem Eingang der Senke im wesentlichen nur die Kopplungsimpedanz entscheidend.“ [3] S.529

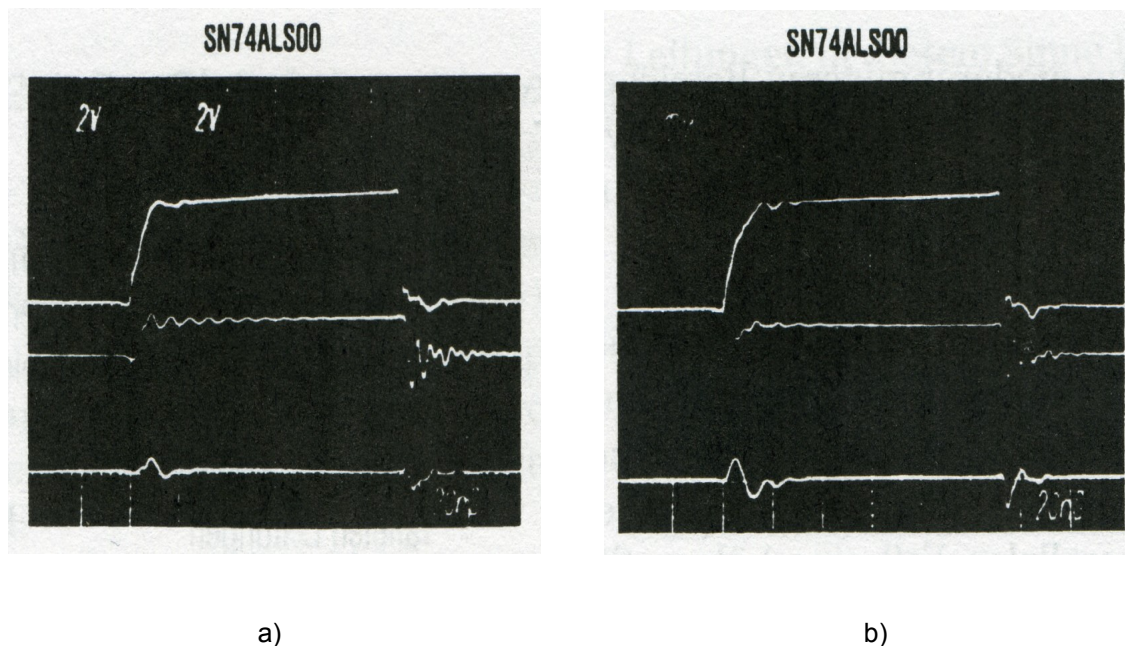


Abbildung 7: a) Paralleles und b) Antiparalleles Übersprechen schnell schaltender Logikschaltkreise [3] S. 530

Durch Messungen bei Texas Instruments wurde festgestellt, dass antiparalleles Übersprechen für größere Störspannungen verantwortlich sein kann [3] S. 529. Der in Abbildung 7 dargestellte Impuls verursacht an der unten dargestellten parallel gerouteten Leiterbahn kapazitiv gekoppelte Störspannungen bei betragsmäßig großen Spannungsanstiegen. Auffällig ist, dass die Darstellung der fallenden Flanke kaum erkennbar ist, da die negative Flankenanstiegszeit kleiner ist, als die positive Anstiegszeit. Dadurch sind die feldgebundenen Effekte nach der fallenden Flanke größer als nach der steigenden. In besonderen Fällen kann das Einfügen einer GND-Potential Leiterbahn einen Teil der Emissionen absorbieren und das Ausmaß des Übersprechens vermindern.

3.2.2 Leitungsgebundene EMV

Elektronische Geräte verfügen häufig über Schaltregler, um die benötigte Spannungsversorgung der Hardware zu realisieren. Sie besitzen einen hohen Wirkungsgrad, erzielen diesen jedoch durch kurze Schaltzeiten in Kombination mit induktiven

Bauelementen. Diese beziehen entsprechend ihrer Auslegung impulsartig Strom aus ihrer Spannungsquelle. Resultat sind Spannungseinbrüche in der Spannungsversorgung und gegebenenfalls zusätzlich feldgekoppelte Störaussendungen. Maßnahmen dagegen sind große Pufferkondensatoren, damit die Spannung nicht unter ein gewisses Minimum fällt. Da die Impulsstrombelastung mit Frequenzen über einhundert Kilohertz, Oberwellen auf der Versorgungsspannung zur Folge hat, kommt ein passiver analoger Tiefpassfilter zum Unterdrücken hochfrequenter Schwingungsanteile zum Einsatz:

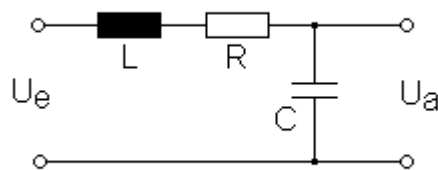


Abbildung 8: Tiefpassfilter 2. Ordnung [4]

Die Übertragungsfunktion des Tiefpass 2. Ordnung beschreibt die frequenzabhängige Dämpfung höherfrequenter Signale ab einer definierten Grenzfrequenz.

$$\underline{H}(\omega) = \frac{jX_c}{R + j(X_L + X_C)}$$

Gleichung 1: Übertragungsfunktion Tiefpass 2. Ordnung [4]

Zur Dimensionierung eines Siebgliebes für hochfrequente Oberwellen, wird der Amplitudengang berechnet und die Grenzfrequenz bei einer Dämpfung der Amplitude um -3dB bestimmt.

$$H(\omega) = \frac{U_a}{U_e} = \frac{1}{\sqrt{\omega^4 L^2 C^2 + \omega^2 R^2 C^2 - 2\omega^2 LC + 1}}$$

Gleichung 2: Amplitudengang des Siebgliebes [4]

In der Verstärkertechnik markiert die Grenzfrequenz den Wert, bei dem die Verstärkung von Strom und Spannung auf 70,7% der maximalen Verstärkung gesunken ist und die an einen ohmschen Lastwiderstand abgegebene Leistung genau halbiert. Somit lassen sich hochfrequente Wechselspannungsüberlagerungen, wie sie durch Schaltregler entstehen, effizient dämpfen.

3.2.3 Signalreflexionen

Eine galvanisch gekoppelte Signalübertragungsstrecke besteht in der Regel aus einer niederohmigen Quelle und einer hochohmigen Senke, die durch einen elektrischen Leiter miteinander verbunden sind. Betrachtet man einen infinitesimal kleinen Abschnitt einer Leitung, so besitzt diese eine charakteristische Impedanz, bestehend aus einer in Reihe geschalteten Induktivität und parallelen Kapazität, unabhängig der Länge der Leitung. Wird am Anfang der Leitung eine Spannung angelegt, werden die Kapazitäten der in Reihe geschalteten LC-Schaltungen nacheinander geladen. Bei diesem transienten Vorgang wirkt die Induktivität gegen das Aufladen der Kapazitäten, wodurch sich letztendlich eine Verzögerung ergibt, bis die Spannung am Ende der Leitung anliegt.

Das Auftreten von Signalreflexionen hängt primär von der Flankenanstiegszeit ab. So wird eine Signalübertragungsstrecke als elektrisch lang bezeichnet, sobald „[...] die Durchlaufzeit einer Signalflanke [...] länger als ihre Anstiegszeit andauert.“ [3] S. 508.

Die Ausbreitungsgeschwindigkeit von elektromagnetischen Feldern beträgt ungefähr zwei Drittel der Lichtgeschwindigkeit:

$$c_{\text{Leiterbahn}} = 0,2 \frac{m}{ns}$$

Gleichung 3: Ausbreitungsgeschwindigkeit des Elektrischen Feldes in einer Leiterbahn [3] S.508

Die Wellenlänge eines Signals ergibt sich aus ihrer Flankenanstiegszeit und der Ausbreitungsgeschwindigkeit im Medium:

$$\lambda = c_{\text{Leiterbahn}} * t_{\text{rise/fall}}$$

Gleichung 4: Berechnung der Wellenlänge der Flanken eines digitalen Signals

Somit kann nun die Grenzbedingung für reflexionsgefährdete Signale bestimmt werden:

$$l_{\text{krit}} > \frac{\lambda}{10}$$

Gleichung 5: Kritische Leitungslänge in Abhängigkeit der Wellenlänge einer Signalflanke [3] S.508

Somit wird bei der Auswahl der Bauelemente darauf geachtet, dass die Flanken angemessenen Wert entsprechend ihrer Trägerfrequenz besitzen, denn während bei hochtaktenden Schaltungen kleine Anstiegs und Abfallzeiten Voraussetzung sind, ist es nicht ratsam, diese auch bei Schaltungen weniger hoher Anforderungen, zu verwenden, da trotz niedrigerer Trägerfrequenz die Flankenform und damit ihre Anforderungen an das Design bestehen bleiben. Reflexionen entstehen außerdem an Stellen, an denen eine Inhomogenität des Wellenwiderstands auftritt, also an Knicken, Abgriffen, Signalsenken und Durchkontaktierungen (Vias). Signalreflexionen sind deshalb bei jeder Leitung zu bedenken, weil reflektierte Signale sich mit einem neuen eintreffenden Signal überlagern. Resultat sind Spannungs-Overshoots und -Undershoots, die die Störfestigkeit beeinträchtigen und bei extremen Fällen Bauelemente beschädigen können.

3.2.4 Signalterminierung

Ist die Impedanz der Signalstrecke bekannt, können Terminierungswiderstände mit exakt dem Wert des Betrags der Impedanz eingefügt werden. Dies wird als Impedanzanpassung bezeichnet und kann besonders bei impedanzkontrolliertem Leiterbahnrouting für Reflexionsfreie Signale sorgen. Je nach Anwendung unterscheidet man dabei in Serienterminierung und Parallelterminierung.

3.2.4.1 Serienterminierung

Serienterminierung findet Anwendung bei unidirektionalen Punkt zu Punkt Signalverbindungen. Hier wird ein Widerstand nah bei der Signalquelle in Reihe geschaltet. Der ohmsche Widerstand sorgt dafür, dass die abgegebene Energiemenge an die Signalstrecke genau so hoch ist, dass die Kapazität der Signalstrecke ausreichend schnell aufgeladen wird. Wird sie langsamer aufgeladen, ergeben sich zu lange Flankenanstiegszeiten. Ist sie zu klein, verringert sich die Effizienz der Terminierung. Die Platzierung findet deshalb nah bei der Quelle statt, weil eine entfernte Platzierung wiederum für Reflexionen am Terminierungswiderstand führen würde. Die Signallaufzeit bis zum Widerstand ist deshalb klein zu halten. Der Vorteil gegenüber der Parallelterminierung, ist die geringe Strombelastung der Signalquelle, jedoch ist Serienterminierung nur bei Punkt-zu-Punkt Signalverbindungen anwendbar.

3.2.4.2 Parallelterminierung

Parallelterminierung führt Störspannungen gegen ein Bezugspotential ab. Je nach Anwendung kann sie bei unidirektionalen Signalstrecken als ohmsche Parallelterminierung an einer Seite, oder bei bidirektionalen an der Quelle und Senke der Signalstrecke platziert werden. Bei der Terminierung hochfrequenter Signale im Megahertz Bereich, wird auf eine Parallelterminierung mit in Reihe geschaltetem Kondensator zurückgegriffen. Dadurch werden nur hochfrequente Störspannungen abgeführt. Dies ist eine weniger effiziente Terminierung, als die rein ohmsche Parallelterminierung, belastet jedoch insbesondere Bussysteme weniger stark. Anwendung findet diese Art der Terminierung bei Ethernetsignalen vom Phy bis zum Transformator und beim CAN-Bus zwischen den differentiellen CAN Pegeln.

3.3 CAN-Bus

Der Controller Area Network Bus ist ein asynchrones Bussystem, das ursprünglich als Kommunikationsprotokoll im KFZ-Bereich verwendet, heute in verschiedensten verteilten Systemen Anwendung findet. Seine Robustheit bei Datenraten von bis zu 1 MBit/s bei Leitungslängen von bis zu 40m macht es zum idealen Bussystem für das Projekt. Das Bussystem ist nach dem OSI-Modell in zwei Schichten definiert, dem Layer 1 „Physical Layer“, und dem Layer 2 „Data Link Layer“. Zu beiden Layern existiert ein IEEE Standard: ISO DIS 11519 für Layer 2 und den oberen Teil des Layer 1, sowie ISO DIS 11898-2 für den unteren Teil des Layer 1. Der Physical Layer bestimmt, neben den mechanischen Eigenschaften der Verbindung, die elektrischen Eigenschaften der Bitcodierung und der Steuersignale [17] S. 22. So beschreibt er die Verwendung eines differentiellen Übertragungsmechanismus, wie auch die Bustermi- nierung, auf die im zweiten Absatz eingegangen wird. Im Data Link Layer wird zwischen Sender und Empfänger eine störungsfreie Übertragung sichergestellt. Im sogenannten CAN-Frame wird ein Rahmen um die Dateninformation gesetzt, um die Information zielgerichtet seinem Empfänger zukommen zu lassen und eine Fehlererkennung beim Nachrichtenempfang vom asynchronen Bus zu ermöglichen.

Die differentielle Datenübertragung im CAN-Bus sorgt für eine hohe Störsicherheit auch bei großen Leitungslängen. Nach ISO11898-2 kommen Twisted-Pair-Kabel mit

einem Wellenwiderstand rund 120 Ohm zum Einsatz. Passend zu diesem Wert, um Signalreflexionen zu unterdrücken, wird der CAN-Bus an beiden Enden mit jeweils einem 120 Ohm Widerstand zwischen dem differentiellen Signalpaar parallel terminiert. Für den Treiber des Teilnehmers, der Nachrichten auf den Bus senden muss, ergibt das eine Strombelastung von:

$$I_{CAN-Drive} = \frac{U_{CAN-H} - U_{CAN-L}}{\left(\frac{R_{Term} * R_{Term}}{2 * R_{Term}}\right)}$$

Gleichung 6: Strombelastung des Treibers beim Senden auf den CAN-Bus

Mit seinem, in diesem Projekt zur Anwendung kommenden, Lineartopologie, zeichnet sich der CAN-Bus im Vergleich zu alternativen Bussystemen, wie Ethernet, durch eine deutlich preisgünstigere Umsetzung bei einer Vielzahl von Teilnehmern aus. Die Busteilnehmer werden parallel an den Bus gekoppelt.

In diesem verteilten System muss die ZAS jede CAS einzeln ansprechen können, während CAS nur bei Anfrage durch die ZAS, Nachrichten an die ZAS senden. Bei den CAS findet untereinander keine Kommunikation statt.

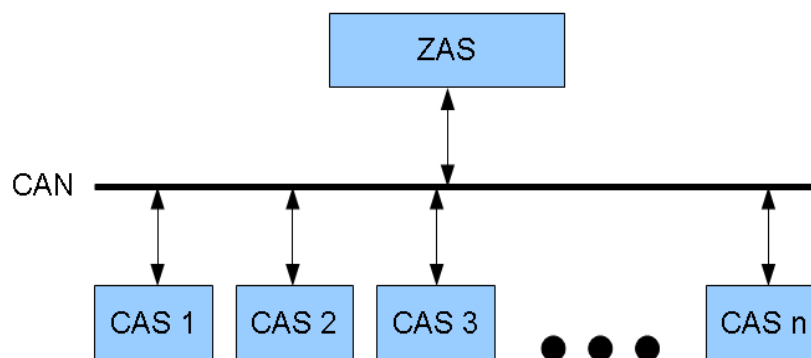


Abbildung 9: Verteiltes System aus ZAS und mehreren CAS

Für das Projekt wurde die Verwendung des CAN-Bus bereits vor der Entwicklung der ZAS festgelegt. Die Steckerbelegung des CAN-Bus ist in einer projektinternen Spezifikation [19] festgelegt.

3.4 Der AT91SAM9G45 Mikrocontroller

Der Atmel AT91SAM9G45 ist ein Mikrocontroller auf Basis des 32 Bit ARM926EJ-S Mikroprozessorkerns. Er stellt sowohl funktionell, als auch räumlich den Mittelpunkt der ZAS dar. Zur Datenerfassung und Ausgabe verfügt der AT91SAM9G45 über eine Vielzahl integrierter, universeller I/O, aber auch spezielle Peripherie zur Realisierung der für das Projekt nötigen Schnittstellen mit externen Geräten und zur Kommunikation. Trotz der großen Anzahl verwendeter Peripherie für die festgelegten Aufgaben in der Spezifikation, verfügt das Computer-on-Module mit diesem Mikrocontroller über ausreichend Reserven zur flexiblen Einbringung zusätzlicher Funktionen für die ZAS, wie die Unterbringung einer Tastaturmatrix an sieben GPIO. Darüber hinaus ist es sinnvoll einige GPIO auf zusätzliche Stiftleisten zur modularen Erweiterung herauszuführen. Damit die Spezifikation der ZAS erfüllt werden kann, muss zunächst die Verwendung der AT91SAM9G45 internen Peripherie definiert werden. In der folgenden Tabelle wird ihre Verwendung veranschaulicht:

Peripherie	Direkte Anbindung	Verbindungsbeschreibung	Vorgesehene Funktion
AC97 Interface 0	CS4299 Audio Codec 97	Synchrone, serielle Schnittstelle speziell für Audio Codec 97 Mixed Signal Controller	Mikrofon Input
DBGU	Nicht herausgeführt, Abgriff bei Bedarf an Hirose Buchsen	Asynchrone, serielle Schnittstelle	Serielle Debugschnittstelle, Zugang zum Bootloader, Terminalkommunikation
Ethernet MII	KSZ5011MLL Ethernet Physical Layer Transceiver	Media Independent Interface mit MDIO Interface	Ethernetschnittstelle
JTAG/ICE	Breakout Pins	JTAG/ICE Interface für Atmel 10-Pin Steckerbelegung	Atmel Debugschnittstelle, direktes Lesen und Schreiben in Register
LCD Controller	Molex Steckverbinder	24-bit RGB Interface mit zusätzlichen Steuersignalen	TFT Display
MCI 0	SD-Steckkartenverbinder	4-bit synchrones Steckkarteninterface mit Kartenerkennung	SD-Speicherkarten lesen und beschreiben
SPI 0	MCP2515 CAN Controller	Synchroner, serieller Datenbus	CAN

Touch Screen ADC Controller	Molex Steckverbinder	Analog-Digitalkonversion von Spannungswerten	Resistives Touch Screen für Display
TWI 0	M41T82R	Synchrone, serielle Schnitt- stelle	Batteriegestützte Echtzeituhr
UART 0	Breakout Pins	Asynchrone, serielle Schnitt- stelle	Linux Ein- und Ausgabe- schnittstelle
UART 1	MAX3232 Pegelwand- ler	Asynchrone, serielle Schnitt- stelle	RS232 mit RTS und CTS
UART 2	74HC00D NAND-Gate und ACPL-M60L-500E Optokoppler	Asynchrone, serielle Schnitt- stelle	MIDI Out, MIDI In
UART 3	Breakout Pins	Asynchrone, serielle Schnitt- stelle	Freie Verwendung, bei Be- darf als GPIO
USB Host 1	Dual USB A Buchse	Serieller Bus mit einem dif- ferentiellen Signalpaar	USB Host Port
USB Host / Device	Dual USB A Buchse	Serieller Bus mit einem dif- ferentiellen Signalpaar	USB Host Port

Tabelle 1: Verwendung der AT91SAM9G45 Peripherie

Die genaue Pinbelegung der Peripherie am Mikrocontroller befindet sich im Anhang [Anlage A].

3.4.1 SPI - Serial Peripheral Interface

Das Serial Peripheral Interface ist ein von Motorola entwickeltes, synchrones Bus-System, das auf dem Prinzip des Schieberegisters aufbaut und seriell Daten in das Datenregister eines weiteren SPI überträgt.

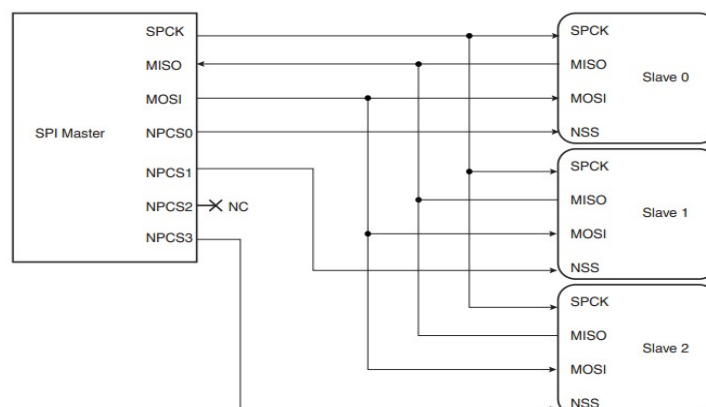


Abbildung 10: SPI-Bus Aufbau [6] S. 419

Die Kommunikationshierarchie basiert auf einem Master-/Slave-Prinzip. Während der Master an mehrere Slaves gleichzeitig senden kann, ist es nur einem der Slaves möglich, durch den Master kontrolliert, Daten zu senden. Die Leistungsfähigkeit dieses Bussystems äußert sich darin, dass es mit einer Taktleitung, zwei Datenleitungen und einer Chip Select Leitung pro Slave, Taktraten bis zu 10 Mhz erreichen kann. Der SPI Bus stellt somit entsprechende Anforderungen an das PCB-Design. Der AT91SAM9G45 Mikrocontroller verfügt über 2 unabhängige SPI.

3.4.2 I²C-Bus - Inter-Integrated Circuit Datenbus

Der von Philips entwickelte, zweiadrige Datenbus I²C, avancierte nach dem Auslauf des Patents im Jahr 2006 zu einem beliebten Datenbus für kurze Signalwege innerhalb eines Geräts. Ein Vorteil des I²C ist die Verwendung von nur zwei Pins am IC. Eine Übertragungsrate von 100KBit/s bis 3,4 MBit/s ermöglicht die Anbindung diverser Peripherie, wie Echtzeituhr-ICs.

HI-Pegel für Takt- und Datenleitung werden passiv mittels pull-up Widerständen hergestellt, was den Bus anfällig für äußere Störeinflüsse macht, im Gegenzug jedoch die bidirektionale Realisierung der Datenleitung über eine einfache Open-Collector Schaltung innerhalb der Peripherie zulässt. Legen Master- und alle Slave-Teilnehmer auf der Datenleitung gleichzeitig LO-Pegel an, fließt somit in der Summe ein Maximalstrom von $I_{max} = U / R_{pullup}$. Um eine ausreichende Flankensteilheit und Störfestigkeit herzustellen, muss der Dimensionierung der pull-up Widerstände somit erhöhte Aufmerksamkeit zugewendet werden, denn mit Verkleinerung der Flankenanstiegszeit steigt proportional die Strombelastung der Leiterbahnen respektive Verlustleistung in den Open Collector Eingängen der integrierten Schaltungen.

Der I²C-Bus eignet sich hervorragend zur Kommunikation zwischen Mikrocontrollern und Peripherie, welche durch ihre Funktionalität über wenige Pins verfügen, oder aufgrund begrenzter Anzahl von Pins nur auf zwei Datenleitungen zurückgreifen können. I²C ist bei Mikrocontrollern von Atmel in der Regel im „Two Wire Interface“ implementiert, das sich von diesem in den wesentlichen Aspekten des Protokolls, nicht unterscheidet [25].

3.4.3 Ethernet

Ethernet ist ein asynchroner Datenbus mit komplexem Kommunikationsprotokoll. An dieser Stelle sollen nur für die hardwareseitige Umsetzung relevante Aspekte erläu-

tert werden. Für das Projekt soll der Fast Ethernet Standard IEEE 802.3 implementiert werden, der Datenraten bis 100 MBit/s ermöglicht. Wie der CAN-Bus, beschreibt Ethernet die OSI Schichten 1 und 2 zur Sicherstellung der Datenflusskontrolle und Zugriffskontrolle. In Schicht 2 wird die Verwendung von Media Access Control (MAC) festgelegt. Der AT91SAM9G45 Mikrocontroller verfügt über einen integrierten Ethernet MAC. Er sorgt für die Kommunikation zwischen Mikroprozessor und Ethernet nach IEEE 802.3 [6] S.683. Zur Umsetzung der Layer 1 Spezifikation, wird ein Physical Layer Transceiver (Phy) für das Ethernet benötigt. Dieser kommuniziert über ein Media Independent Interface, MII, welches eine synchrone parallele Schnittstelle speziell für die universelle Kommunikation zwischen MAC und verschiedenen Ethernet Phy darstellt. Der Ethernet Phy besitzt eine galvanisch getrennte Anbindung an den Ethernet Bus, das elektrisch mit zwei differentiellen Leitungspaaren arbeitet.

3.4.4 USB

Der Universal Serial Bus (USB) ist ein asynchroner serieller Bus, der je nach Standard, bis zu 255 Busteilnehmer zulässt. Die Datenübertragung erfolgt über zwei Leitungen, die ein differentielles Signalpaar bilden. Die Datenrate kann bei EHCI, je nach Anbindung an den Mikrocontroller, bis zu 480 MBit/s erreichen. Zusätzlich kann ein USB Host seine Teilnehmer mit 5 Volt Spannung und einem durch das Hostsystem definierten Strom von bis zu einigen hundert Milliampere versorgen, was für USB Geräte, wie Eingabegeräten und Speichermedien mit geringer Stromaufnahme, ausreichend ist.

3.4.5 UART

Die universelle asynchrone Empfangs- und Sendeschnittstelle ist eine betagte Kommunikationsschnittstelle, die durch ihre einfache und robuste Benutzung bis heute häufig Verwendung in eingebetteten Systemen findet. Der AT91SAM9G45 verfügt über 4 dieser seriellen Schnittstellen. Die asynchrone Datenübertragung kann nur bei einer vorher definierten Baudrate jeweils bei Sender und Empfänger sichergestellt werden.

Die UART-Schnittstelle kann vielseitig genutzt werden: eine Kommunikationsschnittstelle zur Linux Konsole, RS232 und MIDI sollen hier als im Projekt angewandte Beispiele dienen.

3.4.6 Echtzeituhr

Die Echtzeituhr bezeichnet einen integrierten Schaltkreis zur Messung der physikalischen Zeit. Echtzeituhren messen die Zeit über eine Taktquelle, in der Regel auf der Basis von Schaltungen mit Schwingquarzen, die der Integrierte Schaltkreis selbst anregt, oder über fremde Taktquellen. Dabei ist die Zeitmessung genau so präzise, wie die Taktquelle für die sie ausgelegt wurde. Die Frequenz der Taktquelle wird so oft geteilt, bis die gewünschte Zeiteinheit erreicht ist. Zur Anwendungen kommen häufig Schwingquarze mit einer Eigenfrequenz von 32.768 Hz. Diese Frequenz lässt sich mit einer einfachen 15-bit Timerschaltung in Hardware auf 1 Hz teilen, was dem Basiswert von einer Sekunde entspricht. Ausgehend von der Basiszeit, der Unix-Epoche, die auf den 1. Januar 1970 00:00 Uhr Weltzeit festgelegt ist [7], zählt die Echtzeituhr bitweise Sekunde für Sekunde aufwärts und füllt damit ihren flüchtigen Speicher. Sollte es zu einer Unterbrechung der Stromversorgung kommen, werden alle Bits im flüchtigen Speicher auf 0 gesetzt. Die Echtzeituhr wird beim nächsten aktivieren bei der Unix-Zeit von vorn beginnen zu zählen. Je nach Bedarf kann eine Echtzeituhr mit Pufferbatterie ausgeführt werden, um Stromunterbrechungen zu überbrücken und einem Zurücksetzen der Zeit zu entgehen.

Bei integrierten Echtzeituhr-Schaltkreisen wird die Zeitmessung häufig durch das Zählen der Sekunden ab der letzten Unix-Epoche realisiert. Für die Ausgabe an den Benutzer kann nun mit der Unix-Epoche als Offset-Wert das Datum und die seit dem vergangene Zeit auf die Sekunde genau berechnet werden.

3.5 Das Betriebssystem Linux in eingebetteten Systemen

Linux ist ein Open Source Betriebssystem, das von seiner Gemeinschaft seit 1991 kontinuierlich weiterentwickelt wird. Ausgehend vom Buildroot, dem Verzeichnis mit einer Ansammlung von breit gefächerten modularen Quelldateien für das Kompilieren eines speziell konfigurierten Linux Build, bietet es außerdem einen völlig frei konfigurierbaren Betriebssystemkern, den Linux Kernel. Der Kernel ist unter anderem für die Verwaltung der Kernkomponenten, wie CPU, RAM und I/Os zuständig. Wenn der Kernel für spezielle Funktionen eines Mikrocontrollers der von einer speziellen Hardwareumgebung umgeben ist, kompiliert wird, spricht man von einem Embedded Li-

nux. Für diese Funktionen hält das aktuell vorliegende Buildroot von In-Circuit eine große Treiberbibliothek für gängige integrierte Schaltkreise der Peripherie bereit. Linux unterstützt ARM basierte Mikroprozessorkerne wie dem ARM926EJ-S. Dies erlaubt das Konfigurieren und Kompilieren des Kernels mit einer einfach zu beziehenden Entwicklungsumgebung, wie Ubuntu Linux.

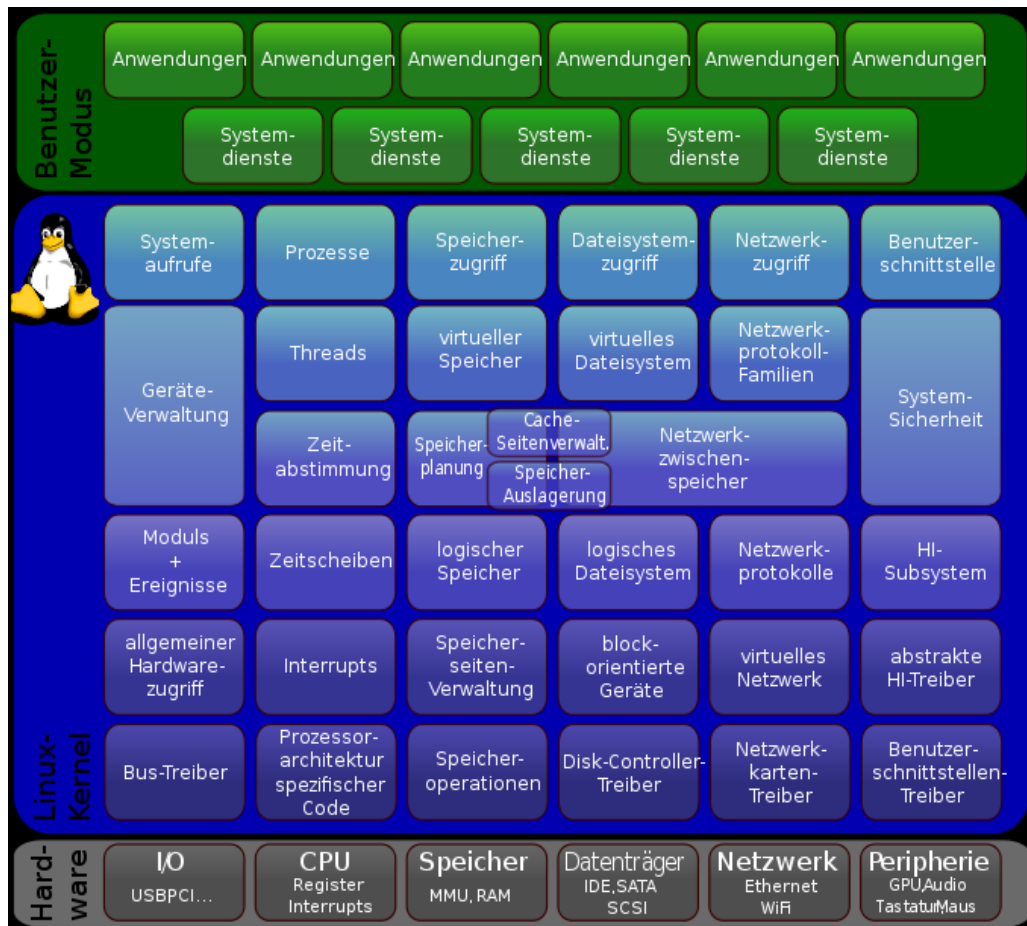


Abbildung 11: Linux Kernel Struktur [9]

Die klassische Linux Benutzerschnittstelle ist die Kommandozeile. Für eingebettete Systeme findet BusyBox als Programmpaket mit direktem Zugriff auf die Kernel-schnittstellen häufig Verwendung. Es stellt eine konfigurierbare und sehr ressourcenschonende Erweiterung zur Kommandozeile dar.

3.5.1 Der Linux Kernel

Als Zentrum des Betriebssystems ist der Kernel das Bindeglied zwischen Hardwareebene und Anwenderebene. Er erledigt eine Vielzahl von Funktionen zum grundlegenden Systembetrieb ohne Zutun des Anwenders, wie Hardware und Prozessverwaltung. Als monolithischer Kernel sind diverse Funktionen und Gerätetreiber fest in

ihm verankert, oder modular einbindbar: „[...] if a driver is configured as a loadable module, its code is not included in the kernel proper, but rather exists as a stand-alone object module, a loadable module, to be inserted into the running kernel after boot.“ [10] S. 80. Vorteil der modularen Treibereinbindung ist die Ein- und Ausbindbarkeit im Betriebssystem, sowie die Minimierung der Kernelgröße im Speicher.

3.5.2 Treiber

Um Programmierern in ihren Anwendungen eine wiedererkennbare Schnittstelle bereitzustellen, verwendet man auf ein Gerät zugeschnittene Treiber, welche die Kommunikation übernehmen und die Hardwareebene abstrahieren. Der Hintergrund ist der, dass ein Programmierer nicht zu wissen braucht, mit welchen Bytes und welchem Timing die Register eines Geräts zu beschreiben sind, um eine gewohnte Funktionalität herzustellen, wenn nur die erwartete Funktionalität von Bedeutung ist. Einer der Gründe, warum Linux als Betriebssystem für eingebettete Systeme so populär ist, ist die breit gefächerte Treiber-Bibliothek, welche viele Geräte und ICs mit wenig Aufwand implementieren lässt.

3.5.3 In-Circuit Buildroot

In-Circuit bietet zum SAM9G45OEM ein Open Source Softwarepaket an. Dieses enthält unter anderem ein Linux Buildroot, welches auf ihre Evaluation Boards und CoM angepasst wurde. Nach einem Vergleich der Quelldateien für Evaluation Boards von In-Circuit und Atmel konnte festgestellt werden, dass die In-Circuit Quelldateien ihrerseits auf den Atmel Quelldateien aufbauen. Für die Treibereinbindung wurden die Quelldateien vom In-Circuit Evaluation Board ADB1000 als Basis verwendet.

3.6 Entwicklungsumgebung

Schaltplan- und PCB-Entwurf wurden mit der Elektronischen Designsoftware „Altium Designer 6“ durchgeführt.

Für die Messungen in den folgenden Kapiteln kommen folgende Geräte zum Einsatz:

Gerät	Funktion	Verwendung
Tektronix TDS 2024B	Digitales Speicheroszilloskop	Zeit- und Frequenzabhängige Messung von Spannung
Metrix MX 55C	True RMS Multimeter	Effektivwertmessung von Strom und Spannung, Durchgangsprüfung, Widerstandsmessung
Kaleas Infrarot Temperatur Scanner 23020	Infrarot Messgerät zur Temperaturmessung	Berührungslose Temperaturbestimmung von Bauelementen

Tabelle 2: Verwendete Messgeräte

4 Hardwareentwicklung

4.1 Entwurf der Baugruppen

Folgende Unterpunkte befassen sich mit der Auswahl und Dimensionierung der Bestandteile der ZAS Hardware. Sofern nicht anders beschrieben, gehen Informationen zur Pin- und Steckerbelegung immer von der Sicht auf den Top-Layer aus, sodass der Schriftzug „TOP“ auf der Leiterplatte lesbar ist.

4.1.1 Bedienschnittstelle

Als zentraler Knotenpunkt im Gesamtsystem, findet die Ein- und Ausgabe an den Benutzer über die ZAS statt. Direkte Eingabe und Ausgabe von Systemparametern und Rückkopplung von Daten ohne zusätzlichen PC sind für die Entwicklung von Software im Frühstadium gleichsam von Vorteil, wie für die Inbetriebnahme der kompletten Anlage im Feld. So ist es möglich, die ZAS über USB Anschluss mit QWERTZ-Tastatur und Maus per Ausgabe über das Display zu bedienen. Die Hauptaufgabe liegt jedoch darin, dem Orgelbauer die Möglichkeit zu geben, Anlagenparameter zu überwachen und bei Bedarf auf diese Einfluss zu nehmen. Aber auch während eines Konzertes ist es von Bedeutung, dem Organisten mit einem einzigen kurzen Blick die einwandfreie Funktion des Systems zu signalisieren. Im Extremfall kann das dynamische Stimmsystem deaktiviert und die Orgel ohne dieses genutzt werden.

Zu den Eingabeelementen existierten vorab bereits Absprachen, wonach eine Tastenmatrix mit typischem Telefonlayout zur Eingabe und ein Drehgeber zur Menünavigation gefordert sind. Die Ausgabe soll primär über ein Display erfolgen. Zur schnellen Zustandskontrolle kann auf die Status-LED zurückgegriffen werden.

4.1.1.1 Tastenmatrix

Eine klassische Telefontastatur mit 3 Spalten und 4 Zeilen bietet dem Bediener eine übersichtliche und vertrauliche Eingabeschnittstelle. Zum Zeitpunkt der Entwicklung des Prototyps ist die Unterbringung der direkten Eingabeelemente in der Orgel nicht festgelegt. Deshalb wurde die Tastenmatrix zusammen mit dem Drehgeber für den

ZAS Prototyp auf einer eigenständigen Eingabeplatine untergebracht. Das ermöglicht das freie Platzieren der Bedienelemente nach Bedarf.

4.1.1.2 Drehgeber

Die finale ZAS Software für den Benutzer soll über ein Menü verfügen. Ein Inkrementaldrehgeber sorgt dabei für eine schnelle und intuitive Navigation. Er ist mit einem integrierten Taster ausgestattet, der zur Bestätigung der Menüauswahl genutzt werden kann. Der verwendete Drehgeber arbeitet mit 2-Bit Graycode, einem Codierungsverfahren, dessen Hamming-Distanz aller benachbarter Codewörter 1 beträgt:

Gray-Code	Binär-Code
0 0	0 0
0 1	0 1
1 1	1 0
1 0	1 1

Tabelle 3: Gray-Code im Vergleich zum Binär-Code

Der Gray-Code findet bei Inkrementaldrehgebern häufig Verwendung. Er verhindert Sprünge durch Prellen oder Übertragungsfehler, weil maximal ein Bit falsch erfasst werden kann. Zur Verminderung der elektromagnetischen Abstrahlung, wurden für Drehgeber und enthaltenen Taster Entprellungsschaltungen in Form von RC-Gliedern eingefügt. Die Beschaltung des Drehgebers mit Eingangswiderständen zum Schutz des Mikrocontrollers, pull-up Widerständen und Hardwareentprellung erfolgt auf dem Motherboard der ZAS, damit bei Verpolung oder Kurzschluss der Mikrocontroller nicht beschädigt wird.

4.1.1.3 Display

Die ZAS wird im Normalfall im Abstand von etwa 80cm vom Organisten angebracht. Aus dieser Entfernung sollen alle Elemente auf der Anzeige ohne Aufwand ablesbar sein. Die Visualisierung wird sich aus vier Elementen zusammensetzen: aktueller Menüpunkt, gewählte Grundstimmung, Intensität der Stimmkorrektur und auftreten von Fehlern. Demnach benötigt das Display mindestens vier Zeilen. Die eindeutige Anzeige der auftretenden Fehler wird hierbei voraussichtlich den meisten Platz einnehmen. Während die Displaytechnologie für den Organisten nicht von unmittelbarer

Bedeutung ist, ist es jedoch für die Implementation im Betriebssystem von Vorteil, wenn das Display mit dem TFT/STN Interface des AT91SAM9G45 kommunizieren kann, da für dieses Interface die Einbindung im Linux Kernel nur mit moderatem Aufwand verbunden ist.

Technologie	480x272 TFT LCD 16M Farben	128x64 Grafik LCD Monochrom	4-Zeilen, 16 Zeichen Dot-Matrix LCD Monochrom
Beispielgerät	NHD-4.3-480272MF-AT-XI#-1	LCD 128AM DIP	TC1604A-04
Hintergrundbeleuchtung (integriert)	LED	LED	LED
Kommunikation mit Mikrocontroller	TFT/STN Controller Interface 24 bit CMOS, DMA Nutzbar	8- oder 4-bit parallel an freien GPIO, ggf. auch 24 bit RGB Interface	8- oder 4-bit parallel an freien GPIO
Sichtbarer Bereich	95x54mm	56x33mm	56x21mm
Kernelimplementa-tion	Einfach bis moderat je nach Controller und Displaytyp, Treiber für Interface vorhanden	Einfach bis schwierig je nach Controller und Displaytyp	moderater Aufwand über Anpassung von dritten Open Source Treibern
Hardware Modulari-tät	moderat über Adapter	moderat über Adapter	gut
Software Modulari-tät	Gut Linux Kernel verfügt über breite Unterstützung von TFT Displays unterschiedliche Displaygröße muss angepasst werden	moderat bis gut Geräte mit gleichem / vergleichbarem Controller unterschiedliche Displaygröße muss angepasst werden	moderat bis gut
Besonderheiten	Flachbandkabel muss bei Layout beachtet werden 105,5 x 67,3mm		
Bemerkungen	Touchpanel Nachrüstbar bzw. Austausch gegen gleiches Display mit Touchpanel vorbereitet	Mehr Anzeigemöglichkeiten als Dotmatrix LCD, jedoch weniger als TFT	Erfüllt Minimum an Anzeigemöglichkeiten
Kosten	Ab 29,00 €	Ab 34,00€	Ab 28,00 €

Pro	freie Gestaltung der Visualisierung in Farbe groß Kompatibilität mit Atmel LCD-Controller optische Qualität Touchpanel optional	passable Visualisierungsmöglichkeiten	Auslaufmodelle preiswert beziehbar
Contra	komplexes Leiterbahnrouting	Aufwand der Kernell-plementation schwer abschätzbar	klein keine grafischen Gestaltungsmöglichkeiten

Tabelle 4: Zusammenfassung Displayvergleich

Nach Abwägen aller Vor- und Nachteile der Displaytechnologien, fiel die Entscheidung zu Gunsten des Fabrikats NHD-4.3-480272MF-ATXI#-1 des Herstellers New Haven Display. Größe, Implementierbarkeit und modulare Touch Screen Funktion sprechen für die Verwendung dieses TFT Displays. Das hochauflösende, farbige Display verschafft der ZAS letztendlich auch einen hochwertigen optischen Eindruck und freie Hand in der Gestaltung der Benutzeroberfläche. In der Entwicklungsphase kann das Display die Konsole des Embedded Linux ausgeben und somit die ZAS völlig eigenständig mit einer Tastatur bedient und bedingt konfiguriert werden.

4.1.2 Energieversorgung

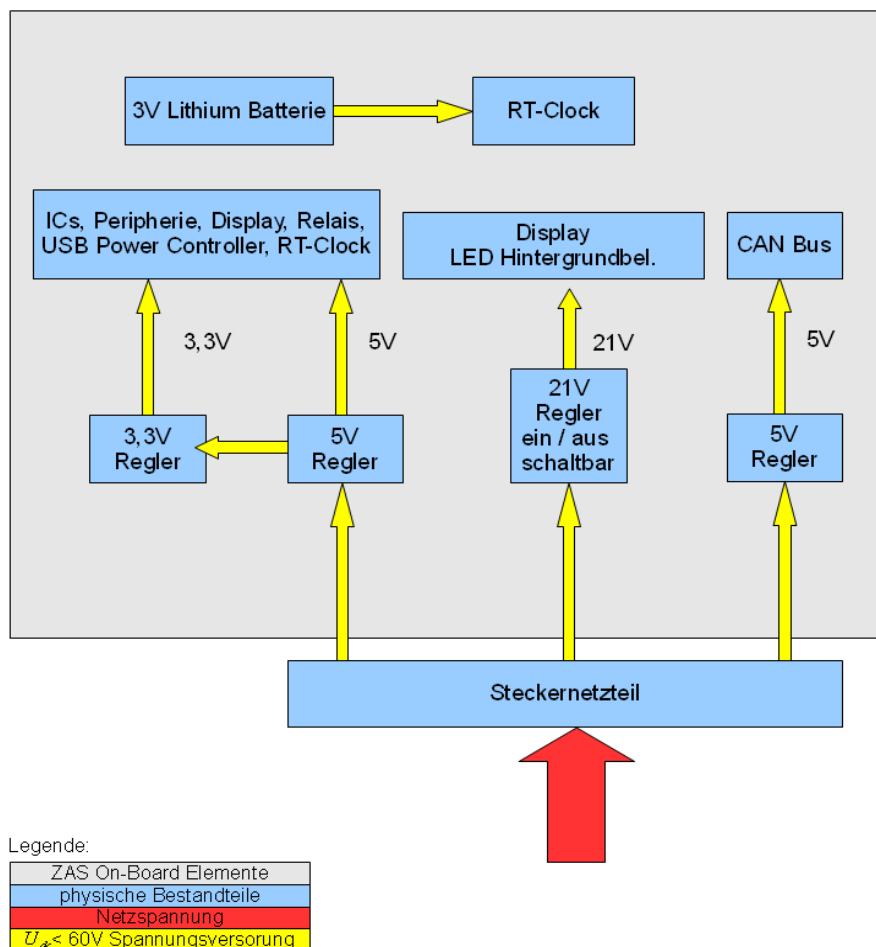


Abbildung 12: Übersicht der Spannungsversorgung der ZAS

Das ZAS Motherboard benötigt für das Computer-on-Module, die Peripherie und I/O Versorgung eine 5 Volt und eine 3,3 Volt Versorgungsspannung. In der ZAS Spezifikation wurde außerdem vorausgesetzt, dass die CAN-Bus Transceiver der Busteilnehmer, welche galvanisch von ihrer Trägerplatine getrennt sind, ihre busseitige 5 Volt Versorgungsspannung vom ZAS Motherboard erhalten sollen. Zur Sicherung der ZAS Versorgung bekommt der CAN-Bus eine gesonderte Spannungsversorgung, um im Fehlerfall auf dem Bus den Betrieb der ZAS zu garantieren. Das verwendete TFT Display arbeitet mit einer LED Hintergrundbeleuchtung, welche aus einer Reihenschaltung von LEDs besteht. In dessen Datenblatt ist ein Versorgungsstrom von $I_{21V}=32mA$ (siehe [11], S. 4) angegeben.

4.1.2.1 Bestimmung der Stromaufnahme aller Komponenten

Zur Dimensionierung der Spannungsversorgung muss anhand der Datenblätter der verwendeten Bauelemente die Stromaufnahme des Boards bestimmt werden. Alle Werte sind aus den Datenblättern entnommen und orientieren sich an den Maximalstromaufnahmen im Betrieb, insofern angegeben. Einige ICs benötigen verschiedene Versorgungsspannungen für den Betrieb. Folgende Tabelle stellt die Stromaufnahme aller relevanten Komponenten auf dem Board zusammen:

Bauelement	$I_{3,3V}/mA$	I_{5V}/mA
ICnova SAM9G45OEM		200
MCP2515	10	
TPS2046B		250
CS4299	24,5	43,4
ISO1050DUBR	3	73
KSZ8051MLL	49	
74HC00D		1
M41T82R	0,06	
G6C-2117P		40
MAX3232	1	
SDCA7A-0200	75	
ACPL-M60L	10	
NHD4.3-480272	5	
MIDI OUT		23
Zwischensumme	178	631

Tabelle 5: Stromaufnahme der ZAS Komponenten

Einige Baugruppen weisen Besonderheiten in der Stromaufnahme auf: Der IC TPS2046B dient der Strombegrenzung an den USB Anschlüssen. Er erlaubt in der Summe einen abgegebenen Strom von 250mA für beide USB Hosts. Somit können einem einzelnen USB Gerät bis zu 250 mA zur Verfügung stehen, oder der Wert auf zwei USB Geräte aufgeteilt werden. Bei Überschreiten dieses Wertes wird der Strom begrenzt und das Betreiben des Geräts ist nicht möglich. Dies wird dem AT91-

SAM9G45 über den Overcurrent Detection Pin vermittelt. Beim ICnova-SAM9G45OEM CoM ergibt sich je nach Prozessorauslastung und Beschaltung eine stark variierende Stromaufnahme. In-Circuit gibt im Datenblatt eine Leistungsaufnahme von maximal einem Watt an [27]. Das entspricht bei 5 Volt Versorgungsspannung einem Strom von:

$$I_{CoM} = \frac{P}{U} = \frac{1W}{5V} = 200mA$$

Formel 7: Stromaufnahme des ICnovaSAM9G45OEM

Die effektive Stromaufnahme des Relais G6C-2117P bezieht sich auf 5 Volt Steuerungsspannung im Steuerstromkreis zum Schließen des Arbeitskontakts:

$$I_{Relais} = \frac{U_{Spule}}{R_{Spule}} = \frac{5V}{125\Omega} = 40mA$$

Formel 8: Stromaufnahme des Relais G6C-2117P

Bei der Berechnung der Stromaufnahme aller CAN-Bus Teilnehmer wird eine Normalverteilung der Stromaufnahme angenommen. Da sich im System 78 CAS befinden werden, von denen nur eine dominant agiert, kann für die restlichen 77 rezessiven Busteilnehmer der Wert für die typische Stromaufnahme anstatt der maximalen herangezogen werden. Alle CAN Transceiver im Projekt sind ISO1050DUBR mit einer typischen Stromaufnahme von 8 mA [20] S. 2. Nur ein Teilnehmer kann auf den CAN-Bus senden. Die Stromabgabe beim Senden orientiert sich an der Terminierung des CAN-Bus nach Abschnitt 3.3:

$$I_{CAN-Drive} = \frac{5V}{120 * 120 / (120 + 120)} = 83 mA$$

Gleichung 9: Theoretische CAN-Bus Strombelastung eines Treibers beim Senden.

Durch den Ausgangswiderstand der Treiberstufe im IC ergibt sich beim Senden laut Datenblatt ein Maximalstrom von 73 mA. Somit berechnet sich die Summe der CAN-Bus Teilnehmer wie folgt:

$$I_{5V,CAN} = 73 mA + 77 * 8 mA = 689$$

Gleichung 10: Theoretische CAN-Bus Strombelastung eines Treibers beim Senden.

Diese Berechnungen dienen der Grundlage zur Dimensionierung der Spannungsversorgungen auf dem Board.

4.1.2.2 Dimensionierung der Gleichspannungsregler

Die eigentliche Spannungsversorgung der Bauelemente auf dem Board unterteilt sich in eine 3,3 Volt und eine 5V Spannungsversorgung. Die 5V Spannungsversorgung wird mit Strömen von mehreren hundert Milliampere belastet, wofür sich ein Step-Down Schaltregler ideal eignet, um die Verlustleistung zu minimieren. Zur einfachen Wartung und Wahrung der EMV bietet sich hier die Verwendung eines Traco Power TSR-1 2450 an. Diese integrierten Schaltregler sind mit bis zu einem Ampere belastbar bei einem Eingangsspannungsbereich von bis zu 36 Volt. Die Dimensionierung der ZAS Spannungsregler erfolgte nicht rein nach Vorgabe durch Bauelemente. Das ICnovaSAM9G45OEM verfügt selbst über Schaltregler. Es kann an 5 Volt betrieben werden. So lässt sich die Belastung auf die Spannungsregler der ZAS nach Bedarf verteilen. Bauelemente, wie das Relais, wurden nach verbesserter Auslastung des 5 Volt Schaltreglers ausgewählt. So konnte die 5 Volt Spannungsebene so weit in ihrem Rahmen belastet werden, dass für die 3,3 Volt Spannungsversorgung keine hohe Strombelastung vorhanden ist und demnach nicht zwangsläufig ein teurer Schaltregler verwendet werden muss. Ein LM317 Linearregler erfüllt diese Aufgabe mit moderater Verlustleistung, wenn seine Eingangsspannung nicht zu stark von der Ausgangsspannung differiert. In Tabelle 6 ist zu entnehmen, dass der 3,3 Volt Nennstrom $I_{3,3V}=178\text{ mA}$ beträgt und die Strombelastung der 5 Volt Spannungsversorgung etwa 81% der Maximalbelastung von einem Ampere beträgt. Der neue Wert der Strombelastung ergibt sich nun mit:

$$I_{5V}=631\text{ mA}+178\text{ mA}=809\text{ mA}$$

Gleichung 11: Berechnung der Belastung der 5 Volt Spannungsversorgung

Für den LM317 Linearregler wird deshalb die 5V Spannung als Eingangsspannung verwendet und damit der TSR-1 in seinem Rahmen zusätzlich belastet. Die Verlustleistung des LM317 ergibt sich aus folgender Gleichung:

$$P_{LM317} = (U_e - U_a) * I_{3,3V} = (5V - 3,3V) * 0,178A = 0,303W$$

Gleichung 12: maximale Verlustleistung des LM317 Linearreglers

Sie ist proportional zur Differenz aus Eingangs- und Ausgangsspannung. Entsprechend dem Datenblatt des LM317 [12] S. 2 erwärmt sich der IC mit SOT223-Gehäuse um $\Theta_{JA} = 53 K/W$ abhängig von der Verlustleistung. Es ergibt sich im Nennbetrieb eine Temperaturerhöhung von:

$$T_{LM317} = \Theta_{JA} * P_{LM317} = 53 K/W * 0,303 W = 16,06 K$$

Gleichung 13: Temperaturerhöhung des LM317 im Nennbetrieb gegenüber der Umgebungstemperatur

welche zur Umgebungstemperatur des LM317 addiert wird. Messungen mit einem Infrarot-Thermometer bei 25 °C Raumtemperatur haben Temperaturwerte zwischen 38 °C und 42 °C ergeben.

4.1.2.3 Display Beleuchtung

Das Display benötigt für seine LED Hintergrundbeleuchtung einen konstanten Strom von 32 Milliampere. Da es sich um eine integrierte Reihenschaltung von LED handelt, summiert sich die Strangspannung auf 20 bis 22 Volt, siehe [11] S. 4. Das entspricht einem mehr als vierfachen der sonst auf dem Board verwendeten Spannungen. Step-Up Schaltregler können diese Spannung erzeugen, sind jedoch aus EMV Sicht äußerst kritisch zu betrachten. Hochfrequente und energiehaltige Impulse können in der sonstigen empfindlichen Hardwareumgebung Störungen verursachen. Da die verwendeten TSR-1 Schaltregler für die 5 Volt Spannungsversorgung einen hohen Eingangsspannungsbereich von bis zu 36 Volt garantieren [13] S.1, fiel die Entscheidung zu gunsten einer Board Versorgungsspannung von 24 Volt in Verbindung mit einem Linearregler als Konstantstromquelle zur Bereitstellung der 32 Milliampere LED Strom. Der LM317 kann mit Hilfe eines Shunts als Konstantstromquelle genutzt werden. Die Spannungsregelung im LM317 regelt auf 1,25V Referenzspannung. Diese bezieht sich immer auf die Differenz zwischen Ausgangsspannung und der Spannung am Adj-Pin. [12] S. 7 Die Referenzspannung, auf die der LM317 regelt, bewegt sich in einem Bereich von 1,2 Volt bis 1,3 Volt:

$$\Delta R_{82} = \frac{\Delta U_{ref}}{I_{LED}} = \frac{\pm 0,05 V}{32 mA} = \pm 1,56 \Omega$$

$$R_{82} = \frac{U_{ref}}{I_{LED}} = \frac{1,25 V}{32 mA} = 39,07 (\pm 1,56) \Omega$$

Gleichung 14: Dimensionierung des Shunts für die Konstantstromquelle

Verwendung findet ein 40,2 Ohm Shunt. Dadurch wird der LED Strom zwar auf 31 mA dimensioniert, jedoch wird die LED Beleuchtung vor Zerstörung durch Überhitzung des P-N-Übergangs geschützt, falls die Herstellertoleranz der Referenzspannung hoch ausfällt.

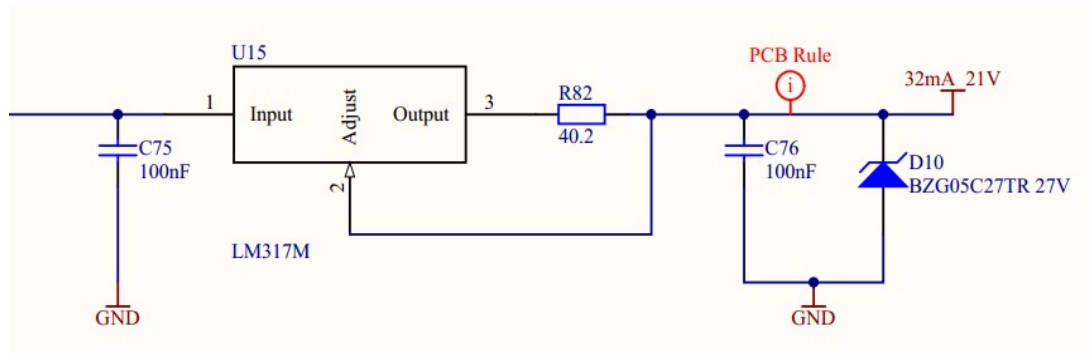


Abbildung 13: Beschaltung der LM317 Display-LED Stromquelle (siehe [Anlage B])

Zum Schutz der LED Hintergrundbeleuchtung gegen übermäßige Eingangsspannungen wurde eine Zenerdiode mit 27 Volt Durchbruchspannung verwendet. Die abgebildete Schaltung wurde zusammen mit der Transistorschaltung zur Einschaltsteuerung wegen Platzmangels auf der BOT Seite unterhalb des Displaysteckverbinders platziert.

4.1.2.4 Dimensionierung des Boardnetzteils

Das Netzteil als Steckernetzteil auszuführen hat mehrere Gründe: Die bessere Handhabung während der Inbetriebnahme und Softwareentwicklung, sowie Platzeinsparung auf der Leiterplatte. So kann bei Nichtverwendung des Relais, die gehäuselose ZAS ohne weitere Vorsichtsmaßnahmen offen betrieben und berührt werden. Über die Stromaufnahme aller Spannungsebenen lässt sich die Anforderung an die 24 Volt Spannungsversorgung bestimmen:

	$I_{3,3V}/mA$	I_{5V}/mA	$I_{5V,CAN}/mA$	I_{21V}/mA
Strombelastung	178	809	689	32
Versorgung	ICs, SD-Karte	3.3V, CoM, ICs, USB, Relais	CAN-Bus	Displaybeleuchtung

Tabelle 6: Strombelastung der Spannungsquellen auf der ZAS

TRACO Power, der Hersteller der TSR-1 2450 Step-Down Schaltregler, veröffentlicht keine Kennlinien zur exakten Bestimmung des Wirkungsgrades seiner ICs. Die grobe prozentuale Angabe im Datenblatt reicht von 84% bis 94% ohne weitere Parameter [13] S. 2. Die Tatsache, dass Schaltregler im oberen Lastbereich ihrer Maximallast betrieben werden, um höchste Effizienz zu erreichen, spricht für einen Wirkungsgrad über 0,9. Für die folgenden Berechnungen wird dennoch vom Mindestwirkungsgrad von $\eta_{TSR}=0,84$ ausgegangen.

$$I_{24V} = \frac{5V}{24V} * \frac{(I_{5V} + I_{5V,CAN})}{\eta_{TSR}} + I_{21V} = \frac{5}{24} * \frac{(809mA + 631mA)}{0,84} + 32mA$$

$$I_{24V} = 332mA$$

Gleichung 15: Benötigter Effektivstrom der 24 Volt Versorgungsspannung durch das Steckernetzteil

Es handelt sich hierbei um den einen Effektivwert, der die maximale Dauerbelastung der 24 Volt Versorgung darstellt. Das Heruntersetzen der 24 Volt Spannung auf 5 Volt bei bis zu einem Ampere Ausgangsstrom hat zur Folge, dass die Schaltregler jeweils für die ZAS Komponenten, als auch für die CAN-Bus Versorgungsspannung, mit einem kleinen Tastverhältnis kurze Impulse bei entsprechend hohen Strömen aus der 24 Volt Spannungsebene beziehen.

$$d = \frac{U_{aus}}{U_{ein}} = \frac{5V}{24V} \approx 0.21$$

Gleichung 16: Berechnung des Tastverhältnis der Step-Down Schaltregler

Demnach muss der 24 Volt Eingang über entsprechende Pufferkondensatoren und einer Impulsstrombegrenzung verfügen, um eine übermäßige Strombelastung des Steckernetzteils zu verhindern. Dies wird mittels des Siebgliedes realisiert:

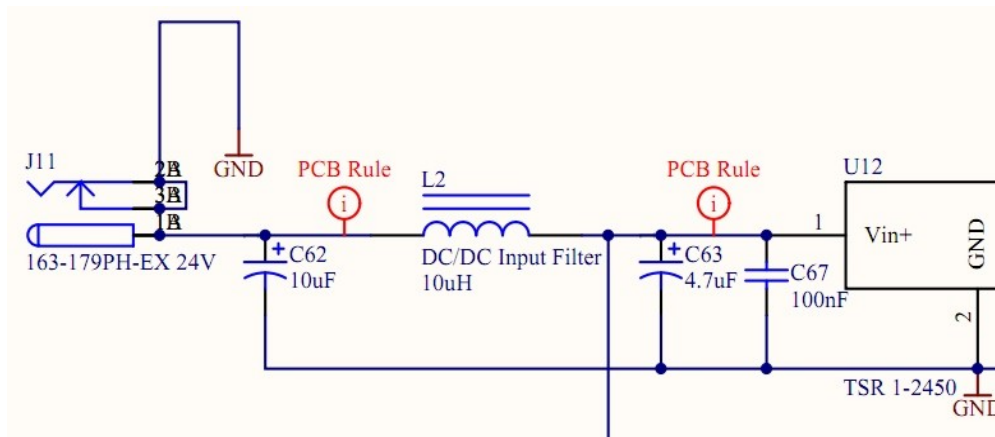


Abbildung 14: Siebglied der 24 Volt Board-Versorgungsspannung (siehe [Anlage B])

Gleichzeitig dient es der Dämpfung der netzteilseitigen Ripplespannung. Der Betrag der Übertragungsfunktion für die relevante Frequenz $f = 500 \text{ KHz}$ berechnet sich über Gleichung 2 der EMV Grundlagen:

$$H(\omega) = \frac{1}{\sqrt{\omega^4 L_2^2 (C_{63} + C_{67})^2 + \omega^2 R_{L_2}^2 (C_{63} + C_{67})^2 - 2\omega^2 L_2 (C_{63} + C_{67}) + 1}}; \omega = 2\pi f$$

$$f = 500 \text{ KHz}; L_2 = 10 \mu\text{H}; C_{63} = 4,7 \mu\text{F}; C_{67} = 0,1 \mu\text{F}; R_{L_2} = 0,15 \Omega$$

$$H(\omega) = 0,00211$$

Gleichung 17: Berechnung des Betrags der Übertragungsfunktion des Siebgliedes bei $f = 500 \text{ KHz}$

Dies ist der Dämpfungsgrad bei besagter Frequenz. Die hochfrequenten Stromimpulse werden so auf ein Maß unter 20 mV minimiert. Die ZAS kann nun mit einem gängigen 24 Volt Steckernetzteil mit einem Ausgangsstrom von 500 mA versorgt werden.

Mit der Simulationssoftware LTSpice, äquivalent zur Pspice, lässt sich das Siebgliedverhalten simulieren:

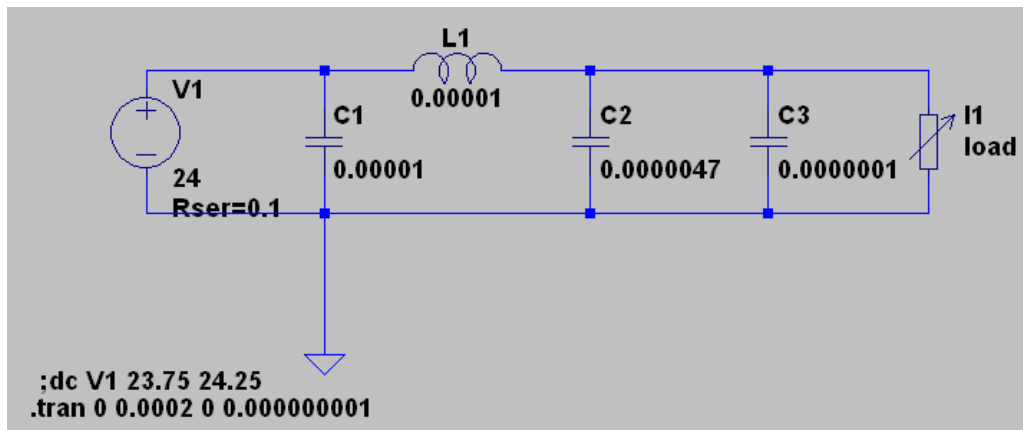


Abbildung 15: Darstellung der Schaltungssimulation des Siebgliebes in LTSpice

Die 24V Gleichspannungsversorgung wird von der Spannungsquelle V1 simuliert. I1 simuliert einen Schaltregler mit 810 mA Impulsstromaufnahme als Spitzenwert bei einem Tastverhältnis von $d \approx 0,21$. Über den Kondensator C1 mit 10 µF Kapazität wird die Eingangsspannung gepuffert und reflektierte Spannungen durch die Schaltregler gedämpft. C2 stellt den Pufferkondensator eines Schaltreglers dar. Dieser wird über die Spule L1, die hochfrequenten Impulsströme entgegenwirkt, geladen. Somit entsteht ein Filtereffekt, der die Momentanstrombelastung des Steckernetzteils reduziert. C3 dient als zusätzlicher Abblockkondensator am Schaltreglereingang.

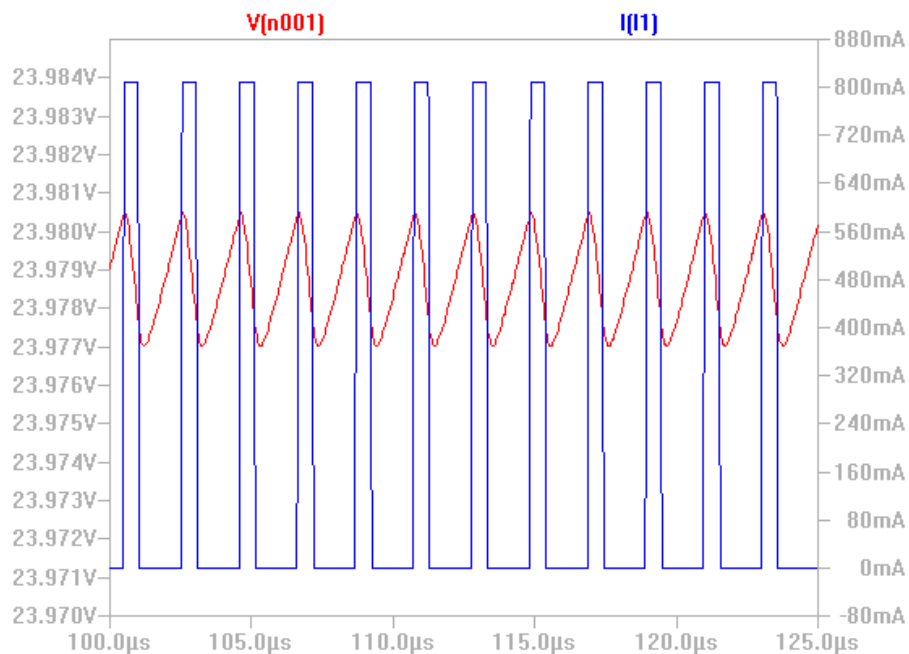


Abbildung 16: Simulation des Siebgliebes

Zu erkennen ist der impulsweise bezogene Strom des simulierten Schaltreglers I1. Die resultierende Spannung am 24V Ausgang des Steckernetzteils wird rot dargestellt. Die Ripplespannung beträgt $U_{SS} = 3,5mV$.

4.1.3 Steuerung der Displaybeleuchtung und des Relais

Damit die Display Hintergrundbeleuchtung und das Relais über den Mikrocontroller mit 3,3 Volt ein- und abgeschaltet werden können, kommt eine Transistorschaltung mit einem logic level Texas Instruments BSS138 Metall-Oxid-Halbleiter-Feldeffekttransistor (MOSFET) zum Einsatz. Der N-Kanal enhancement MOSFET ist für die Ansteuerung mittels logischer Spannungen ab 2,5 Volt Gate-Source-Spannung geeignet [21] S. 3. Das Schalten des Relais und der LED-Hintergrundbeleuchtung bedarf zwar keiner hohen Anforderungen an Schaltzeiten, jedoch sind die beiden Baugruppen im Betrieb der ZAS erwartungsgemäß jederzeit eingeschaltet. MOSFET als spannungsgesteuerte Schalter besitzen im Durchlasszustand nahezu keine Steuerstromaufnahme und einen niedrigen Durchlasswiderstand ($R_{DS(on)}$), was zu einer geringen Verlustleistung führt. Der $R_{DS(on)}$ der Transistoren bestimmt sich über die Kennlinien im Datenblatt bei 3 Volt Gate-Source-Spannung. 3 Volt Gate-Source Spannung:

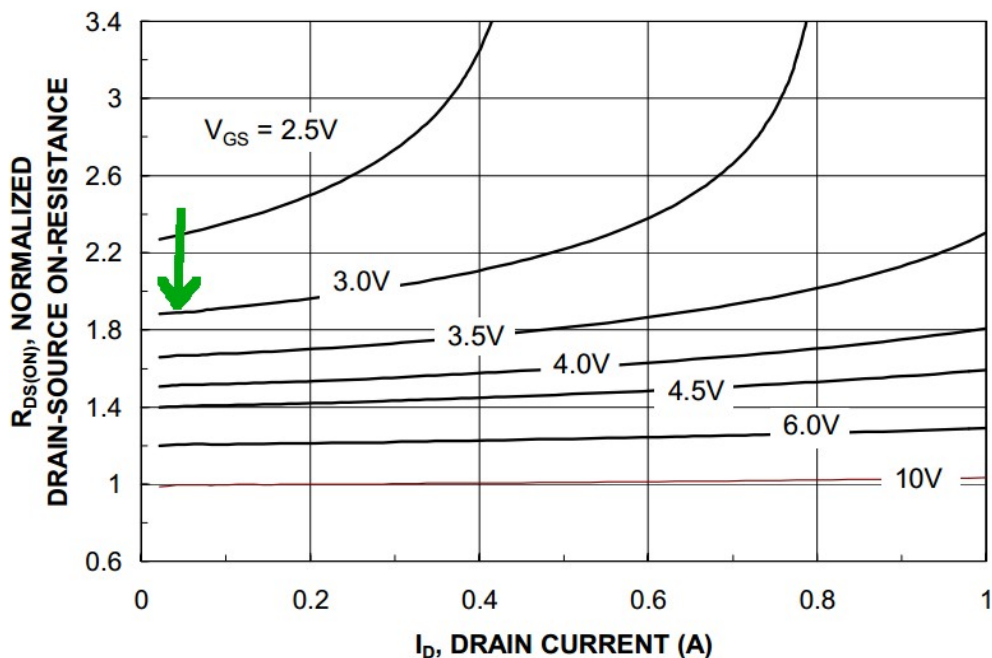


Abbildung 17: Kennlinie des normalisierten Durchlasswiderstands des BSS138 in Abhängigkeit vom Drainstrom und der Gate-Source-Spannung [21] S. 3

Die relevante Ablesestelle für die beiden ZAS Komponenten ist grün markiert. Sie zeigt auf einen Drainstrom Wert von etwa 40 mA. Der Faktor für die Berechnung des $R_{DS(on)}$ ist somit etwa 1,9. Der Ausgangswert des $R_{DS(on)}$ lässt sich auf der gleichen Seite im Datenblatt von Kennlinie „Figure 2“ ablesen, nach der der Basiswert temperaturabhängig bis zu 2,1 Ohm bei 125°C Chiptemperatur beträgt. Damit berechnet sich der Wert des Durchlasswiderstandes wie folgt:

$$R_{DS(on)} = 1,9 * 2,1 \Omega = 3,99 \Omega \approx 4 \Omega$$

Gleichung 18: Berechnung des Durchlasswiderstandes des BSS138

Bei bis zu 40 mA Strombelastung beläuft sich der Spannungsfall in den Transistoren auf maximal 0,16 Volt, was weniger als 4 % Spannungsfall im Relais Stromkreis und weniger als 1 % beim Stromkreis der Display-Hintergrundbeleuchtung ausmacht. Ihren Betrieb beeinträchtigt er nicht. Die Verlustleistung beträgt beim Relais Stromkreis etwa 7 mW und bei der Hintergrundbeleuchtung etwa 4 mW.

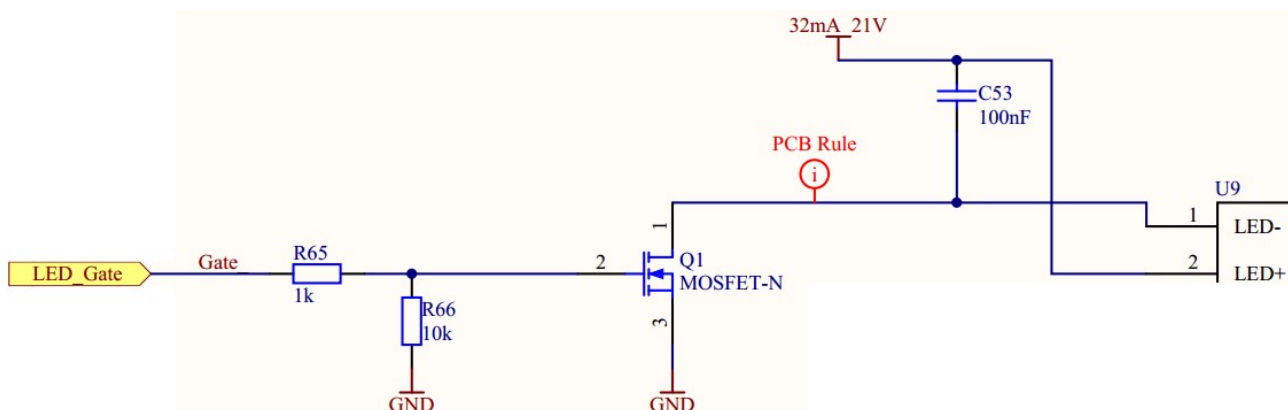


Abbildung 18: LO-Side Transistorschaltung zur Steuerung der LED-Hintergrundbeleuchtung

Durch das Laden über einen Widerstand und einem GPIO mit 3,3 Volt HI-Pegel wird die Gate-Source-Kapazität über den Spannungsteiler der Widerstandsschaltung auf circa 3 Volt geladen und der MOSFET leitfähig. Sobald der GPIO auf LO-Pegel gesetzt wird, entlädt sich die Gate-Source-Kapazität über den pull down Widerstand und den GPIO auf LO-Pegel gegen Masse. Der pull down Widerstand dient zur sicheren Abschaltung des Relais, falls der GPIO am Mikrocontroller keinen definierten Pegel annimmt.

4.1.4 Umsetzung des potentialfreien Kontakts

Der potentialfreie Kontakt soll bei 230 Volt effektiver Wechselspannung einen Strom von mindestens 0,5 Ampere schalten und vom Mikrocontroller gesteuert werden.

Als potentialfreier Kontakt kommt das Relais G6C von Omron zum Einsatz. Dieses kann mit einer Wechselspannung von 250 Volt bei bis zu 10 Ampere Dauerstrombelastung betrieben werden.

4.1.4.1 Steuerstromkreis

Die Spule besitzt einen Kupferwiderstand von 125 Ohm bei einer Steuerspannung von 5 Volt, was einem Steuerstrom von 40 mA entspricht. Um das Relais anzusteuern, kommt die Schaltung aus 4.1.3 zum Einsatz und wird durch eine Relais-Schaltkompensation ergänzt. An einem induktiven Bauelement, wie der Steuerspule, bewirkt ein Abriss des Stromflusses, dass das Magnetfeld um die Induktivität zusammenbricht und dabei eine Spannung induziert. Dies kann zu extremen Spannungsspitzen führen, die das schaltende Element zerstören oder enorme elektromagnetische Abstrahlung zur Folge haben. Deshalb wird eine Schaltkompensation hinzugefügt, die die Abschaltspannung absorbiert. Sie besteht aus einer zum Stromfluss in der Steuerspannung antiparallel geschalteten Freilaufdiode. Sie hat alleinstehend jedoch den Nachteil, dass sie eine Verzögerung des Abschaltvorgangs verursacht und damit die Arbeitskontakte des Relais unnötig belastet. Deshalb sorgt eine Zenerdiode erst ab einer Spannung von 5,1 Volt für die Absorption der Relais-Abschaltspannung.

4.1.4.2 Lastseitige Umsetzung

Lastseitig ist auf die Einhaltung der Mindestabstände auf dem PCB zu achten. 230 Volt sinusförmige Effektivspannung hat einen Scheitelwert von 326 Volt. Zur Definition der Mindestabstände auf dem PCB wird von den Normen IEC 60038, DIN EN50178 und VDE 0160 ausgegangen. Dazu liegt ein auf diesen Normen basierendes Dokument vor, das die relevanten Werte übersichtlich darstellt [22]:

- Als Gerät, das an ein festes Niederspannungsnetz ohne Maßnahmen zum Schutz gegen transiente Spannungen entworfen wurde, ist die ZAS in die Überspannungskategorie II einzuordnen. Somit beträgt die Bemessungs-Stoßspannung 2,5 KV.

- Die ZAS ist für den Betrieb in Räumlichkeiten mit nur trockener, nichtleitender Verschmutzung gegebenenfalls leichter Betauung vorgesehen, deswegen entspricht sie dem Verschmutzungsgrad II. Zur Bemessung der Mindestluftstrecke kann nun in Tabelle „Mindestluftstrecken für die Isolationskoordination“ mit der Bemessungsstoßspannung 1,5 KV und der Unterstellung eines inhomogenen elektrischen Feldes, wovon aus Toleranzgründen ausgegangen wird, eine Mindestluftstrecke von 1,5 mm abgelesen werden.
- Nach Angaben des Leiterplattenherstellers verfügt das verwendete Basismaterial „standard FR4“ über eine Kriechstromfestigkeit CTI > 175 V [23]. Damit fällt es in die Kategorie Isolierstoffgruppe III a. Die Kriechstreckendimensionierung ergibt sich anhand der Tabelle „Mindestkriechstrecken für Betriebsmittel mit langzeitiger Spannungsbeanspruchung“ für gedruckte Schaltung der Isolierstoffgruppe III a und ergibt 1mm.

Da die ZAS in der Entwicklung und gegebenenfalls im Feld im Betrieb berührt werden soll, wird die Isolation auf Schutzkleinspannung ausgelegt, was eine Verdoppelung der Isolationsabstände zur Folge hat.

Somit ergibt sich der Fall, der auf der Bemessungsseite als Beispiel 1c angeführt ist: Eine Mindestluftstrecke von 3 mm bei einer Mindestkriechstrecke von 3,6mm und einer Spannungsfestigkeit von 3000 Volt effektiver Wechselspannung.

Im PCB wurde den Netzspannung führenden Kontakten, sowie der Schutzkleinspannung ein Abstand von 4mm umgesetzt.

Ein Test der Isolation und der Spannungsfestigkeit des potentialfreien Kontakts steht noch aus. Er muss in einer speziellen Umgebung mit geeigneten Sicherheitsmaßnahmen und Prüfmethoden durchgeführt werden. Dabei muss die Isolation der Relais-Kontakte gegen die Board-Schutzkleinspannung auf 3000V Stoßspannung getestet werden. Die Verwendung des Relais mit Netzspannung kann deshalb erst nach einer erfolgreichen Prüfung erfolgen.

4.1.5 Umsetzung der Schnittstellen

Dieser Abschnitt befasst sich mit der Umsetzung der Hardwareschnittstellen der ZAS.

4.1.5.1 UART Schnittstelle

Eine UART Schnittstelle wird für die Ein- und Ausgabe der Linux Konsole benötigt. Dazu wird die nach dem AT91SAM9G45 Datenblatt definierte UART0 verwendet. Damit sie kompatibel zur vorhandenen USB-UART Bridge von In-Circuit ist, entspricht sie der gleichen Pinbelegung:



Abbildung 19: Pinbelegung der UART 0

4.1.5.2 Mikrofoneingang

Um die Audioaufnahme über ein Mikrofon zu realisieren, gibt es beim Atmel AT91-SAM9G45 Mikrocontroller zwei Möglichkeiten: Die Audiosignale über einen Analog-Digitalumsetzer direkt am Mikrocontroller aufnehmen, oder das Audio Codec 97 Interface nutzen. Ersteres hat jedoch eine lange analoge Signalstrecke in einer sonst von digitalen Schaltungen dominierten Umgebung zur Folge. Es ist dann mit Störungen im analogen Signal zu rechnen, was bei dessen späterer Verwendung als Audiomessgerät zur Tonfrequenzbestimmung nicht tragbar wäre.

Der Mikrocontroller verfügt über ein Audio Codec 97 Interface, das speziell zur Kommunikation mit einem AC97 kompatiblen Audio Controller konzipiert wurde. Für den analogen Signalweg bedeutet das, dass er nur vom Mikrofonanschluss bis zum AC97 Controller führt, demnach sehr kurz ist und abgeschirmt von digitalen Signalen an einer Boardseite gesetzt werden kann.

Zur Verwendung kommt ein Cirrus Logic 4299 Audio Codec '97 Mixed Signal Controller mit einer Vielzahl von weiteren Ein- und Ausgängen. Diese wurden bereits im Schaltplan ihrer Spezifikation entsprechend so beschaltet, dass sie nicht benutzt werden können, um keine EMV Probleme, oder zusätzliche Stromaufnahme zu verursachen. Ausschlaggebend für die Wahl des CS4299 ist die gute Treiberunterstüt-

zung für den Linux Kernel und der gute Preis im Vergleich zu Produkten von Mitbewerbern, wie dem WM9707SCFT von Wolfson Microelectronics. Die Umsetzung des PCB Designs befindet sich im Abschnitt „PCB-Design des Mikrofoneingangs“.

4.1.5.3 CAN-Bus

Die ZAS versorgt CAN-Bus Teilnehmer mit einer 5 Volt Spannungsversorgung mit bis zu einem Ampere Ausgangsstrom. Der CAN-Bus ist demnach galvanisch mit der ZAS Massepotential verbunden. Zur Schirmung der Buskabel ist CAN Shield an der ZAS verbunden. Die Steckerbelegung des CAN-Bus erfolgt nach der im Projekt verwendeten CAN Spezifikation [19] komplementär an einem Sub-D 9-Pin Stecker:

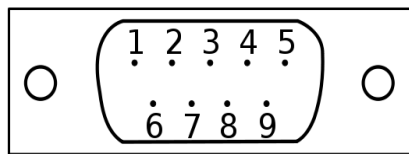


Abbildung 20: Sub-D 9-Pin Steckerbelegung

1	2	3	4	5	6	7	8	9
-	CAN_L	GND	-	-	GND	CAN_H	-	+5V _{CAN}

Tabelle 7: Pinbelegung des CAN-Bus Steckers

Die CAN Implementation erfolgt bei der ZAS über einen Standalone CAN Controller mit Anbindung an SPI Interface des AT91SAM9G45. Der SPI 0 dient auf der ZAS als Medium zur Übertragung der CAN-Bus Inhalte. Da der CAN-Bus für eine Datenrate von 1 MBit/s ausgelegt ist, verfügt der SPI Bus über die zehnfache Datenrate, um eine latenzarme Übertragung der CAN Nachrichten zu ermöglichen.

Aufgrund der hohen Dichte von Leiterbahnen in der Nähe der Hirose Buchsen musste beim SPI auf Terminierung verzichtet werden. Statt dessen ist der MCP2515 CAN Controller sehr nah am CoM platziert. Zwischen beiden Seiten des SPI Taktsignals wurde eine GND Leitung eingefügt, die Übersprechen vermindert.

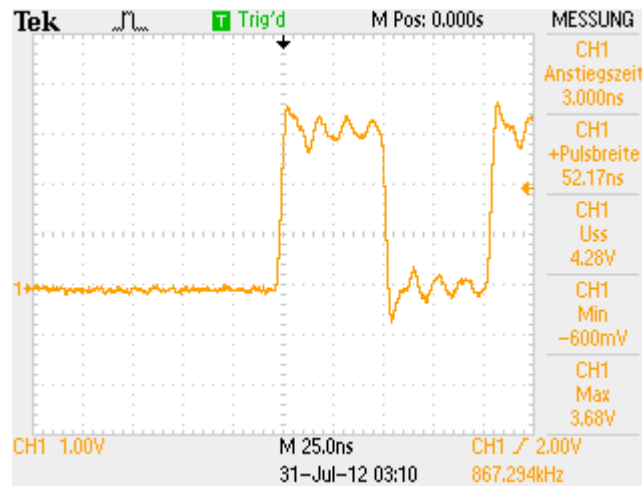


Abbildung 21: SPI-Bus SCLK Flankenbewertung

Die Signaleingänge des MCP2515 sind für Spannungen „ $-0,6V$ to $V_{DD}+1V$ “ [14] S.69 ausgelegt, was bei $V_{DD}=3,3V$ einer Spannungstoleranz von $-0,6V$ bis $4,3V$ entspricht. Die Overshoot und Undershoot Werte bewegen sich in vertretbaren Grenzen (siehe [14], S. 69).

4.1.5.4 Ethernet

Mit einer Trägerfrequenz von 25 Mhz ist das Ethernet eines der kritischen Elemente in diesem Design. Wegen der Komplexität der Umsetzung ist das Ethernet maßgebend für die Realisierung aller anderen Schnittstellen. Ihre Umsetzung wird auf der Basis dieses Abschnitts durchgeführt.

Bei einer Frequenz von 25 MHz sind folglich die Anstiegszeiten der Signalfanken sehr klein. Im Datenblatt des verwendeten Ethernet Phy KSZ5011MLL [15] S. 38 sind Mindestanstiegszeiten von 3 ns angegeben. Daraus ergibt sich die Forderung, den Ethernet Phy so nah wie möglich an die CoM Buchsen zu platzieren, um eine gute Voraussetzung für das Leiterbahnrouting der Signalwege zu haben. Auf die konkrete Umsetzung des Routings nehmen diverse Parameter Einfluss: Von Design Hinweisen zur Umsetzung von Ethernetbaugruppen [18], bis hin zu den EMV-Grundlagen „Feldgebundene EMV“.

Näheres zur konkreten Umsetzung befindet sich unter „PCB-Design des Ethernets“.

4.1.5.5 MIDI

Die Schaltung der MIDI Ein- und Ausgangsanbindung an den Mikrocontroller basiert auf dem TTL – MIDI Interface von Pete Brown [24]. Diese wurde auf 3,3 Volt Span-

nungspegel angepasst. Grundlage für die Implementation dieser Schaltung ist eine UART Schnittstelle und das Baud Rate Division Register im AT91SAM9G45, das die Generierung der für MIDI nötigen Baudrate von 31,25 KBit/s ermöglicht. MIDI wird elektrisch durch eine Stromschleife realisiert, die über Widerstände am Sender und Empfänger zu einem Spannungssignal gewandelt wird. Der MIDI Eingang muss galvanisch getrennt sein, um Masseschleifen zu vermeiden. Er wird mit einem Optokoppler realisiert, der gleichzeitig den eintreffenden 5 Volt Pegel in einen 3,3V Pegel wandelt. Dieser wird an UART2 RxD gelegt. Der MIDI Ausgang muss mit einem Strom von 23 mA bei 5 Volt belastbar sein. Ermöglicht wird dies durch eine doppelte Invertierung des UART2 TxD Signals.

4.1.5.6 USB Host Ports

Ein USB Host Port ist als Anbindung für das Stimmgerät und ein weiterer für ein beliebiges Zusatzgerät, wie eine Tastatur, bestimmt. Der im AT91SAM9G45 integrierte USB Host Controller gibt direkt das differentielle USB Signal aus. Dieses muss mit differentiellem Leiterbahnrouting bis zu der Dual USB-A Buchse geführt werden. Die als Standard angewendete Serienterminierung von 22 Ohm je Leiterbahn wurde in der Nähe der CoM Anschlussbuchsen platziert. Nach einem Telefonat mit einem Mitarbeiter von In-Circuit am 21.04.2012, stellte sich heraus, dass der Mikrocontroller zwar über drei USB Host Ports und einen USB Host-Device Port verfügt, jedoch zwei der Host Ports auf dem ICnovaSAM9G45OEM nicht zugänglich sind, wodurch auf die Verwendung des Host-Device Ports zurückgegriffen werden musste. Dieser befindet sich auf gegenüberliegender Seite der CoM Anschlussbuchsen und hat einen entsprechend langen Weg bis zur Dual USB-A Buchse. Zum Schutz der USB Leiterbahnen vor elektrostatischer Entladung wurden Pulse Guard Schutzdioden parallel geschaltet.

4.1.5.7 RS232

Das Hermode Tuning Modul kommuniziert über RS232 über RxD und TxD. Auf der ZAS wurde das RS232 mit CTS und RTS Leitungen umgesetzt, da am verwendeten Pegelwandler MAX2323 ohnehin zwei Sende- und zwei Empfangskanäle vorhanden sind.

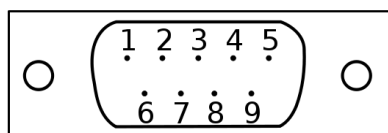


Abbildung 22: Sub-D 9-Pin Stecker

Die Steckerbelegung an der ZAS wurde als „Data Terminal Equipment“ (DTE) an einem Sub-D 9-Pin Stecker umgesetzt:

1	2	3	4	5	6	7	8	9
-	RxD	TxD	-	GND	-	RTS	CTS	-

Tabelle 8: Steckerbelegung des RS232

4.1.5.8 Real Time Clock M41T82R

Bezug nehmend auf die Grundlagen in Kapitel 2 benötigen Echtzeituhren für ihren Zweck ausreichend genaue Taktquellen, um eine vertretbare zeitliche Drift aufzuweisen. Der AT91SAM9G45 Mikrocontroller verfügt über eine integrierte, nicht gepufferte Echtzeituhr, welche nach jedem Trennen vom Netz zurückgesetzt wird. Zum Anlegen von Logfiles muss das ZAS System über eine verlässliche, nicht volatile Uhrzeitangabe verfügen, um Vorgänge und Fehler im System zeitbezogen zurückverfolgen zu können. Deswegen wird auf einen integrierten Echtzeituhr-Schaltkreis zurückgegriffen.

Für die Verwendung des M41T82R von ST sprechen einige Fakten: Die Echtzeituhr arbeitet intern mit einzelnen Registern für Jahrhunderte, Jahre, Monate, Wochen, Tage, Stunden, Minuten, sowie hunderstel, zehntel und ganzen Sekunden. Das macht ihn für ein Betreiben über das Jahr 2038 hinweg beständig.

Zur Kommunikation mit dem Mikrocontroller, besitzt die Echtzeituhr ein I²C Interface. Der AT91SAM9G45 besitzt in seiner verwendeten Ausführung jedoch einen unzuverlässigen TWI Controller. Dieser verursacht Überläufe und Unterläufe bei hoher Busauslastung. Im von Atmel zur Verfügung gestellten Quellcode wird auf die Verwendung des GPIO Code hingewiesen, um diesen Problemen aus dem Weg zu gehen [16] Z. 584-588.

4.1.5.9 LCD-Steckverbinder

Das Display NHD-4.3-480272MF-ATXI-1 mit einer Trägerfrequenz von 9,2 Mhz in das Design einzupflegen bedarf lediglich einer, jedoch zwingenden Voraussetzung: Leiterbahnen so kurz wie möglich zu halten. Aus Signalintegritätsgründen ist dies Pflicht, da eine zusätzliche Terminierung der Signale auf der Leiterplatte nicht in Frage kommt, denn die elektrischen Eigenschaften des Flachbandkabels sind nicht bekannt und bei knapp 30 Leiterbahnen eine Serienterminierung vorzunehmen ließe, ähnlich wie beim Ethernet, die Struktur des Routings ausufern und hätte möglicherweise sogar negative Folgen. Das einzige Problem, was dagegen spricht, den Steckverbinder für das LCD direkt neben die Hirose Buchsen des CoM zu platzieren, ist die Tatsache, dass das Display über ein relativ starres Flachbandkabel verfügt, das die örtliche Unterbringung des Displays stark von der Platzierung des Steckverbinders abhängig macht.

4.1.5.10 Zusätzliche Peripherie

Das Bedieninterface ist auf einer gesonderten Leiterplatte untergebracht. Die Anschlüsse von Tastenmatrix und Drehgeber befinden sich auf der BOT Seite. Folgende Abbildung definiert den linken Pin, beim Drehgeber links unten, als Bezugspin zum Anschluss der Kabel. Die verwendeten Kabel verfügen zusätzlich über eine rote Markierung, die zur Orientierung dienen.

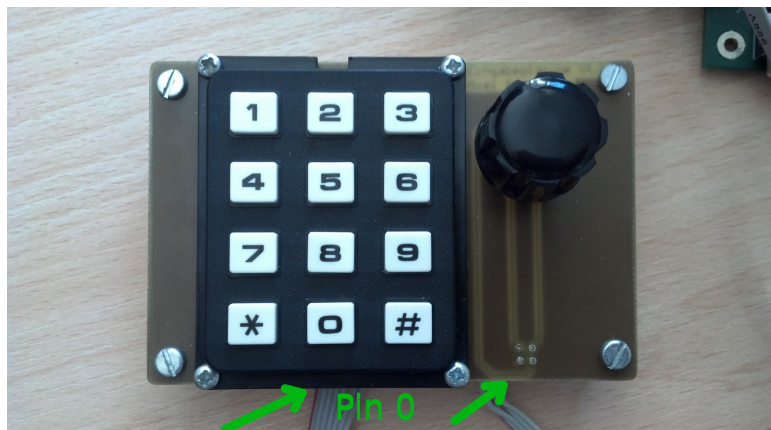


Abbildung 23: ZAS Eingabeplatine mit Markierung eines Bezugspins

Die zur Verbindung mit der Eingabeplatine verwendeten GPIO sind auf der ZAS als Stiftleisten herausgeführt.

- Die Tastenmatrix-Steckverbindung S 2 als 1 * 7 Stiftleiste mit 2,54 mm Raster:

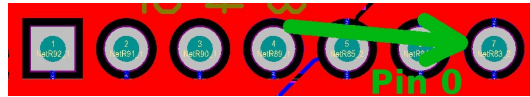


Abbildung 24: Markierung an S 2 zur eindeutigen Verbindung der Tastenmatrix

- Die Drehgeber-Steckverbindung U 3 als 2 * 2 Stiftleiste mit 2,54 mm Raster:

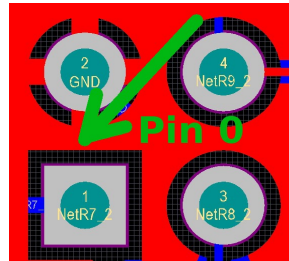


Abbildung 25: Markierung an U 3 zur eindeutigen Verbindung des Drehgebers

Der AT91SAM9G45 verfügt über ein JTAG Interface, dessen Pins kompatibel mit dem bei Atmel häufig verwendeten JTAG/ICE sind. Diese wurden auf eine 2 * 5 Stiftleiste im 2,54 mm Raster herausgeführt:

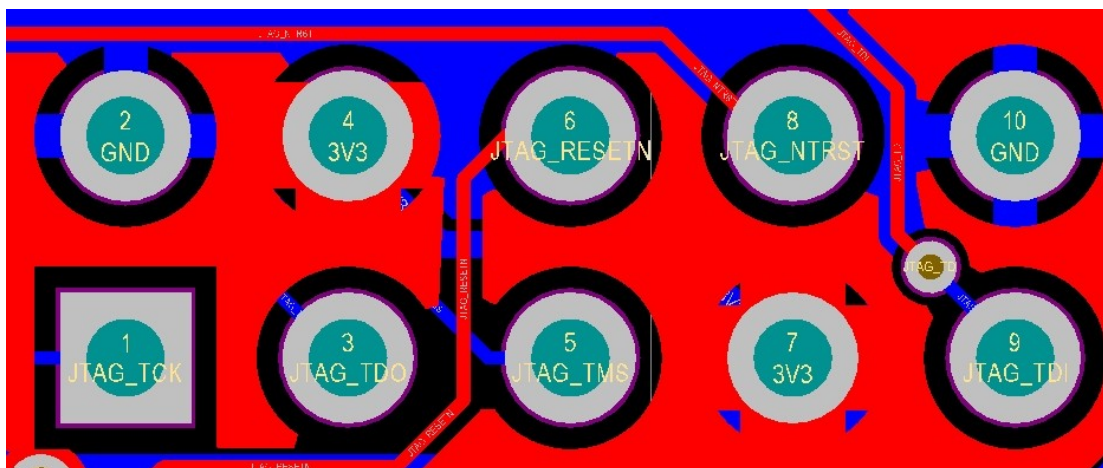


Abbildung 26: JTAG Pinbelegung, kompatibel mit Atmel JTAG/ICE 10-Pin Stecker

Der ZAS Prototyp der Version 1.0 verfügt außerdem über einige zusätzlich herausgeführte GPIO als Stiftleisten, welche über die Anforderungen der Hardwarespezifikation von Punkt 3.1 hinaus gehen. Sie dienen der universellen Erweiterbarkeit der ZAS.

- P 1 ist eine 2 * 3 Stiftleiste im 2,54 mm Raster mit optionaler SPI 1 Funktionalität:

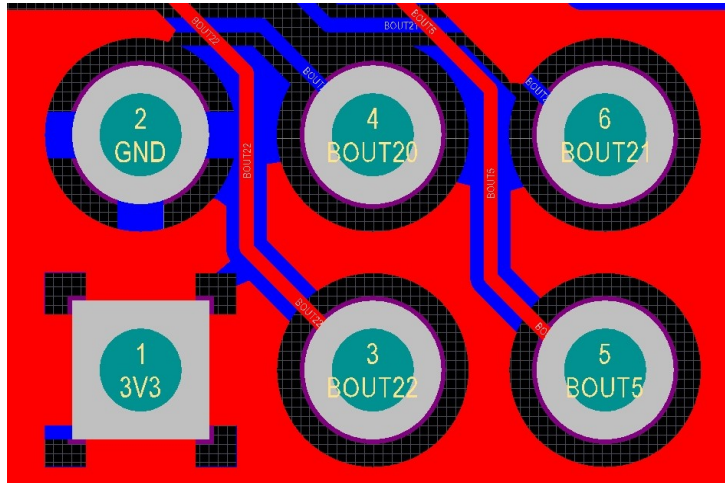


Abbildung 27: SPI 1 Pinbelegung, 3 – SCK, 4 – MOSI, 5 – MISO, 6 – NPCS0

- P 3 ist eine 1 * 4 Stiftleiste im 2,54 mm Raster mit optionaler UART 3 Funktionalität. Die Pinbelegung ist äquivalent zu UART 0.

Wird die Funktion von SPI 1 und UART 3 nicht benötigt, stehen auf der Platine somit 6 freie GPIO zur Verfügung, über die bei Bedarf weitere Funktionen ermöglicht werden können.

Die restlichen freien GPIO des ICnovaSAM9G45OEM wurden nicht herausgeführt, weil sie nur über sehr lange Wege und Durchkontaktierungen geroutet werden könnten. Das hätte zur Folge, dass Stromführende Polygonflächen zusätzlich verschachtelt und die Signalintegrität anderer Peripherie gefährden würden.

4.2 Leiterplattenentwurf

Der Leiterplattenentwurf erfolgt auf Basis des Schaltplans der ZAS. Dieser verfügt in seinem kompletten Umfang über 19 Seiten und kann auf der Anlagen-CD unter „ZAS.pdf“ betrachtet werden. Eine Übersicht über den PCB Top Layer befindet sich im Anhang [Anlage C], um Einblick in die Platzierung und das grundlegende Routing haben zu können. Bei genauen Analysen von spezielle Baugruppen sollte die Ansicht im Programm Altium Designer vorgezogen werden.

In den folgenden Unterpunkten wird auf die besondere Designumsetzungen eingegangen.

4.2.1 Planung

Vor dem Beginn des PCB Designs müssen diverse Einschätzungen gemacht und Entscheidungen getroffen werden, die den Verlauf des Leiterplattenentwurfs nachhaltig beeinflussen werden. Größe und Form, sowie Anzahl der Lagen der Leiterplatte sind die maßgebendsten. Während Größe und Form für dieses Projekt eine untergeordnete Rolle spielen, ist die Frage der Anzahl der Lagen vor allem eine finanzielle. Vergleicht man bei Multi-Circuit-Boards die Preise, so fällt schnell auf, dass zweilagige Leiterplatten mit etwa fünfzig Euro zu Buche schlagen, während bei vierlagigen Platinen mindestens mit dem fünffachen gerechnet werden muss. Auf der anderen Seite ergeben sich durch die Verwendung von nur zwei Lagen, aufgrund von Anschlussvorgaben und mangelndem Platz, unerwünschte Umwege über längere Leiterbahnstrecken, oder Durchkontaktierungen von Signalen und Bezugspotentialen.

Beim Entwurf einer zweilagigen Leiterplatte mit teilweise empfindlichen Baugruppen ist es also sinnvoll, die Leiterplatte zunächst in analoge und digitale Baugruppen zu unterteilen:

Priorität	hoch	mittel	niedrig
Signalintegrität digital	Ethernet, USB, AC97, MMC, Display	TWI, UART, SPI	Tastenmatrix, Drehgeber, Buttons, MOSFET Gatespannung
Signalintegrität analog	Mikrofoneingang		Relais

Tabelle 9: Prioritäten in der Platzierung und Routing

Die Unterteilung ergibt sich aus den Flankenanstiegszeiten der Signale und ihre Trägerfrequenz im Hinblick auf die elektromagnetische Beeinflussung anderer Baugruppen

Das ZAS Motherboard soll zur einfachen Bestückung so weit wie möglich in SMD-Technik realisiert werden. Durchsteckelemente an den Anschlüssen sind für die mechanische Stabilität dennoch sinnvoll. Ein zweilagiges Leiterplattendesign ist aus finanzieller Sicht anzustreben. Da es keine Vorgaben zur örtlichen Platzierung der Anschlüsse gibt, können die Baugruppen so platziert werden, dass kritische Signalwege mit wenig Durchkontaktierungen realisiert werden können.

4.2.2 Umsetzung von EMV Designrichtlinien

4.2.2.1 Allgemeines EMV Design

Bei Gleichspannungsreglern ist bei einem großen Verhältnis von Eingangs- zu Ausgangsspannung und Strömen jenseits der hundert Milliampere die Verwendung von Schaltreglern für die Energieeffizienz von Vorteil. Diese Form der Gleichspannungswandlung birgt hohe Anforderungen an das PCB-Design, um Störaussendungen zu vermeiden. Schaltregler nach dem Step-Down Prinzip arbeiten mit hochfrequenten Stromimpulsen mit mehreren hundert Kilohertz, um die Größe der Speicherelemente minimieren zu können. Diese Stromimpulse in den Induktivitäten und Leiterbahnen können zu großen Strahlungsemissionen führen. Folglich ist bei Schaltreglern auf Verwendung korrekten Kernmaterials für Induktivitäten zu achten. Design-Hinweise in den Datenblättern geben in der Regel Hinweise auf das zu verwendende Kernmaterial für die Induktivitäten, als auch empfohlene Anordnungen der Bauelemente des Step-Down Reglers. Diese beruhen in der Regel auf den Grundlagen, möglichst kurze Leitungswege zu nutzen, um ohm'sche Verluste und parasitäre Blindkomponenten zu verringern, als auch die Emissionen nicht abgeschirmter Leiterbahnen zu vermindern.

Eine Leistungsfaktorkorrektur der Energieversorgung ist nicht vorgesehen. Das ZAS Motherboard als eigenständiges Gerät mit eigenem Netzanschluss, wird nach Dimensionierung der Energieversorgung eine Wirkleistungsaufnahme von $P_{Gesamt} = 15W$ nicht überschreiten. Da Oberwellen-bedingte Blindleistungsanteile vorab nicht eindeutig erfasst werden können, lässt sich eine Abschätzung der resultierenden Scheinleistung nur grob durchführen. Insgesamt wird die ZAS eine Scheinleistung von $S = 20W$ jedoch nicht überschreiten. Nach DIN EN 61000-3-2 ist eine Leistungsfaktorkorrektur bei elektrischen Geräten erst ab einer Scheinleistung $S = 75W$ Pflicht. Grund dafür ist die überhand nehmende Anzahl an kapazitiven Lastenflüssen auf das Niederspannungsnetz, welche durch Schaltnetzteile in elektronischen Geräten verursacht werden. Folgen für das Niederspannungsnetz sind nicht mehr rein sinusförmige Spannungsverläufe durch hohe kapazitive Blindleistungsbelastung der Energieversorgungsnetze und erhöhte Oberwellenanteile.

Abblockkondensatoren mit $C = 100nF$ werden nah bei Bauelementen platziert. Dies sorgt für die Bereitstellung von Impulsströmen, welche somit nur in unmittelbarer

Nähe des Bauelements auftreten [3] S.629 und verhindert großflächige Störaussendungen durch Impulsströme auf der Spannungsversorgungs- und Masseleitung. Diese verursachen ohne Abblockkondensatoren einen transienten Spannungsfall mit erhöhten Strahlungsemissionen für angrenzende Leiter, sowie andere Bauelemente.

Ein Grund, warum sich die dual USB Buchse mit Leistungsschalter in unmittelbarer Nähe des 5 Volt Schaltreglers befindet, ist die hohe Stromaufnahme von bis zu 250 mA. Dies entspricht der größten Strombelastung der 5 Volt Spannungsversorgung im Design. Die Anschlüsse der USB Ausgänge am CoM liegen weit auseinander, wodurch die Platzierung der USB A-Dual Buchse letztendlich zu Gunsten der Spannungsversorgung in der Nähe der Schaltregler stattfindet.

4.2.2.2 VersorgungsspannungsfILTER

Der EingangsspannungsfILTER liegt direkt am 24 Volt Anschluss. Eine Messung der Ripplespannung am Siebglied bestätigt die in Abschnitt 4.1.2.4 simulierten Werte:

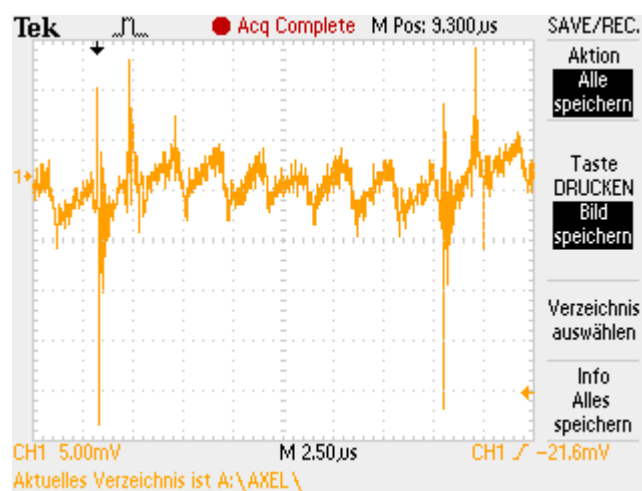


Abbildung 28: Messung der 24 Volt Ripplespannung mit Siebglied

Die Ripplespannung beträgt $U_{SS} = 10\text{mV}$. Die größeren Peaks entstehen durch Überlagerung der Impulsströme der 5 Volt Boardspannungsversorgung im Nennbetrieb und der CAN-Bus 5 Volt Spannungsversorgung bei etwa einem Ampere Ausgangsstrom.

4.2.3 PCB-Design des Ethernets

Der Ethernet Phy wurde mit höchster örtlicher Priorität in der Nähe der Anschlussbuchsen des CoM untergebracht, um möglichst kurze Signalwege zu garantieren und bei den vom CoM zum Phy laufenden MII-Signalen auf eine Terminierung verzichten

zu können. Grund hierfür ist der sehr begrenzte Platz an den Hirose Buchsen unter dem CoM, wobei zusätzlich einige Vias gesetzt werden müssen, um die Leiterbahnkreuzungen umzusetzen, die zusätzliche Inhomogenität hervorrufen, die die Vorteile einer Terminierung aufheben.

Die Signale unterteilen sich funktional in MII, zuständig für die reine Datenübertragung der Ethernetpakete, und das MDIO Interface, was das Lesen und Schreiben der Phy Command-Register und damit dessen Steuerung ermöglicht. MII unterteilt sich weiterhin in Tx und Rx Signalwege. Tx ist getrennt von Rx geroutet, um antiparalleles Übersprechen von Takt- und Datenleitungen zu vermeiden. Bei der Leitungslänge getakteter paralleler Signale muss auf ein möglichst gleichzeitiges Eintreffen beim Empfänger geachtet werden. Setzt man voraus, dass die Leitungsimpedanz und Flankenform aller Signale gleich ist, ergibt sich somit die Forderung, alle Leiterbahnen in ihrer Länge anzugleichen. In der Praxis haben sich aufgrund von Störeinwirkungen und inhomogener Leitungsimpedanzen kleine Toleranzen ergeben. [18] Richtwerte bei 3 ns Flankenanstiegszeit sind 9 mm zwischen kürzester und längster Leitung einer Signalgruppe. Der Takt beider Signalrichtungen ist mit einem erhöhten Abstand zu den Datenleitungen bedacht. Sämtliche Ethernetsignale sind mit möglichst wenigen Vias versehen. Vias führen zu schwer kontrollierbaren Reflexionen und erhöhten elektromagnetischen Emissionen und Immisionen durch Dipoleigenschaften. Durch Anschlussvorgaben, kreuzende und ebenfalls kritische Signale, wie USB-Signale, kann dennoch nicht auf die Verwendung von Vias verzichtet werden. So werden Vias bei kritischen Signalwegen möglichst dicht bei der Signalquelle platziert. Niederohmige Potentialflächen sorgen auch bei Vias für Absorption elektromagnetischer Strahlung.

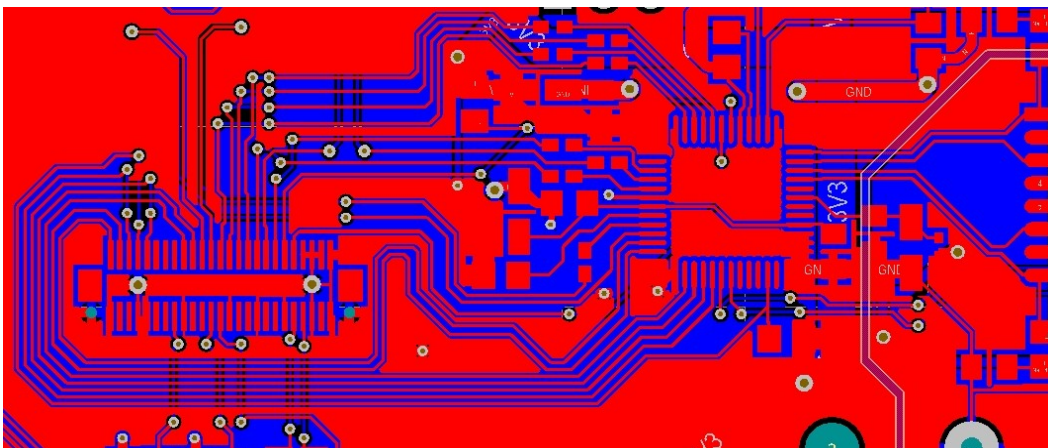


Abbildung 29: Routing der Ethernet MII Signale

Links im Bild befindet sich der eigens angelegte Footprint einer Hirose Buchse für das CoM. Bei dem rechten QFP-48 Footprint handelt es sich um den Ethernet Phy KSZ5011MLL. Die Signale, die beide verbinden, sind die MII- und die MDIO-Interface Signale. Die MII Rx Signalgruppe, sowie die COL, CRS und Tx-Clock, die vom Phy ausgehen, sind mit 100 Ohm terminiert. Die rechts am Phy gerouteten Signale bis zum rechten Bildrand sind die differentell gerouteten Rx und Tx Signale, die dort am Ethernet Trafo angebunden sind. Als induktives Bauelement widerspricht der Ethernet Transformator den Richtlinien, Signalwege kurz zu halten, denn auch bei nahezu idealer induktiver Kopplung im Transformator ist der Signalübertrager als Quelle hochfrequenter elektromagnetischer Störungen zu sehen. Als Kompromiss empfiehlt sich ein Abstand von etwa 2,5 cm zum Phy. Weiterhin lässt sich auf dieser Abbildung die Schwäche eines zweilagigen Designs erkennen: die Massefläche unter den MII Signalen wird durch kreuzende Leiterbahnen unterbrochen, was ein impedanzkontrolliertes Routing sehr schwierig gestaltet.

Dennoch sind Signale mit kurzen Anstiegszeiten als terminierungspflichtig zu betrachten. Zu diesen gehören in erster Instanz die ausgehenden Signale des Ethernet Phy KSZ5011MLL mit $t_{rise/fall} = 3\text{ ns}$. Aus den Gleichungen in Abschnitt 3.2.3 ergibt sich die kritische Leitungslänge bei diesen Anstiegszeiten bei einer Länge ab 6 cm, die von allen MII Signalen überschritten wird. Bei den Verbindungen zwischen Mikrocontroller und den genannten ICs, handelt es sich um Punkt-zu-Punkt Signalverbindungen. Bei dieser Art ermöglicht die Serienterminierung eine effiziente Verminderung von Reflexionen:

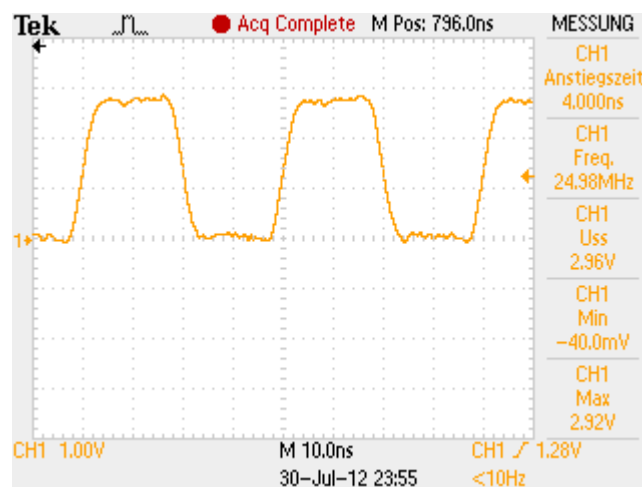
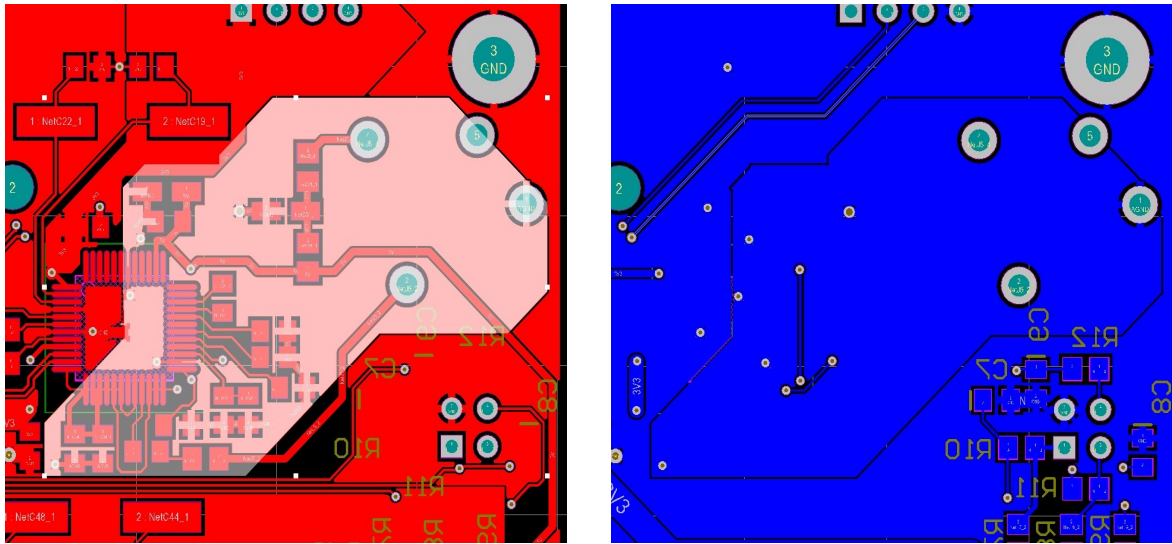


Abbildung 30: Serienterminiertes RxC Signal des MII Interface

Signalgruppen werden idealer Weise mit einer durchgängigen Bezugsfläche für rücklaufende Signale ausgeführt, sodass sich Feldeffekte hinlaufender und rücklaufender Signale gegenseitig aufheben. Dies würde einer Sternpunktanordnung der Masseverbindung bedürfen. Auf einer zweilagigen Leiterplatte wirkt sich eine Sternpunktanordnung der Masseflächen bei einem komplexeren Design mit kreuzenden Signalwegen jedoch negativ auf die Signalintegrität aus. Der Vorteil durch sich aufhebende Felder bei Feldkopplung von hin- und rückläufigen Signalwegen geht verloren, wenn das Massepotential über Vias verlegt werden muss. Die durch Vias entstehende Inhomogenität kann bei ungünstiger Anordnung die Signale so verschleppen, dass die Summe der Abstrahlung beider Leiterbahnen größer ausfällt, als ohne Sternpunktanordnung. So wird an dieser Stelle ein Kompromiss eingegangen und die Entscheidung getroffen, den gesamten Bottom Layer als Massefläche auszulegen. Nur bei unumgänglichen Kreuzungen von Signalwegen, wie den MII Signalen und USB auf dem Top Layer, werden Signale über den Bottom Layer geführt. Der Schwingquarz mit Kondensatoren für die Takterzeugung des Phy wurde so nah wie möglich beim Phy platziert.

4.2.4 PCB-Design des Mikrofoneingangs

Der CS4299 Audio Controller bedarf bei seiner Umsetzung im PCB der gleichen Sorgfalt wie das Ethernet. Er hat jedoch noch eine Besonderheit in Hinblick auf den analogen Mikrofoneingang. Analoge und digitale Signale sind idealerweise immer voneinander geschirmt zu führen, so ist für den Mikrofoneingang und die analoge Signalseite des CS4299 eine zusätzliche Massefläche vorhanden. Sie schirmt die analogen Signale des Mikrofons ab. Für die analogen Signale verfügt der CS4299 außerdem über Analog-Spannungseingänge, die mit einem Ferrit zur galvanischen Störrentkopplung von der digitalen Spannungsversorgung versehen sind.



a)

b)

Abbildung 31: a) Top Layer b) BOT Layer des analogen Mikrofoneingangs mit separater Massefläche

Die Verbindung der analogen Massefläche erfolgt nach den Design Vorgaben des Datenblatts des CS4299 unterhalb des ICs mit einem dünnen Verbindungssteg zur digitalen Masse.

4.3 Hardwareprototyp Übersicht

Die folgende Abbildung stellt das ZAS Motherboard mit aufgestecktem ICnova-SAM9G45OEM nach der Bestückung dar.

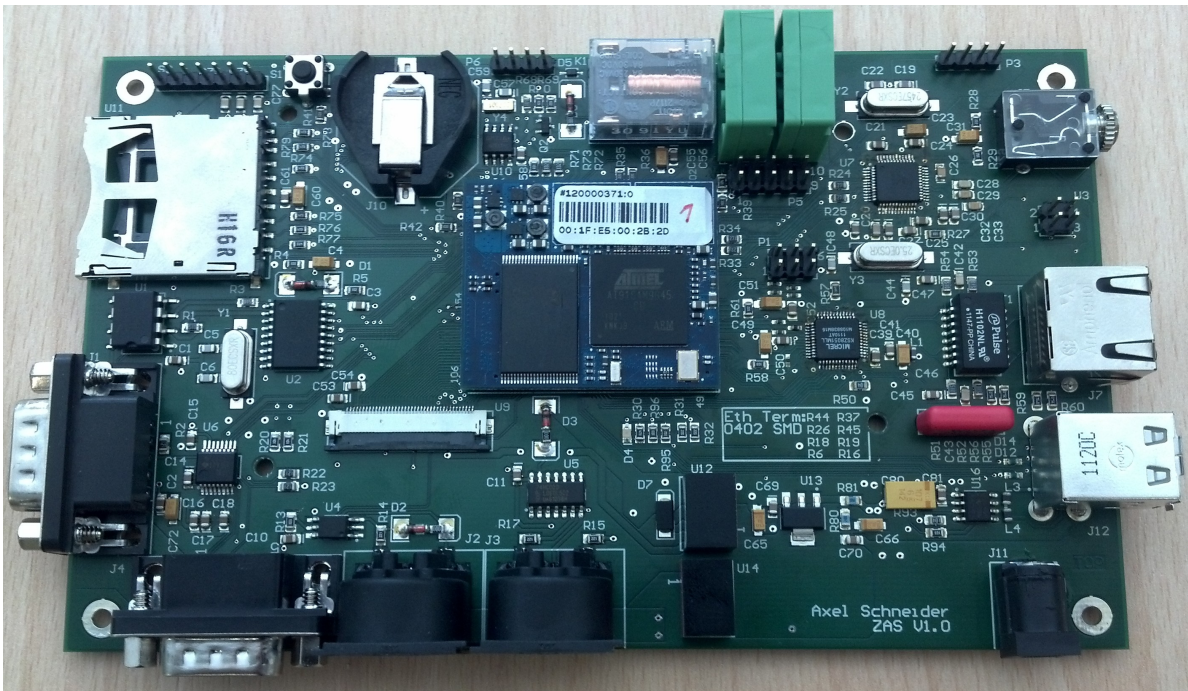


Abbildung 32: ZAS Prototyp Betrachtung der Top Seite mit IcnovaSAM9G45OEM

Zur Übersicht über die Verteilung der Baugruppen und ihren Anschlüssen dient eine bearbeitete Übersicht aus dem PCB Entwurf:

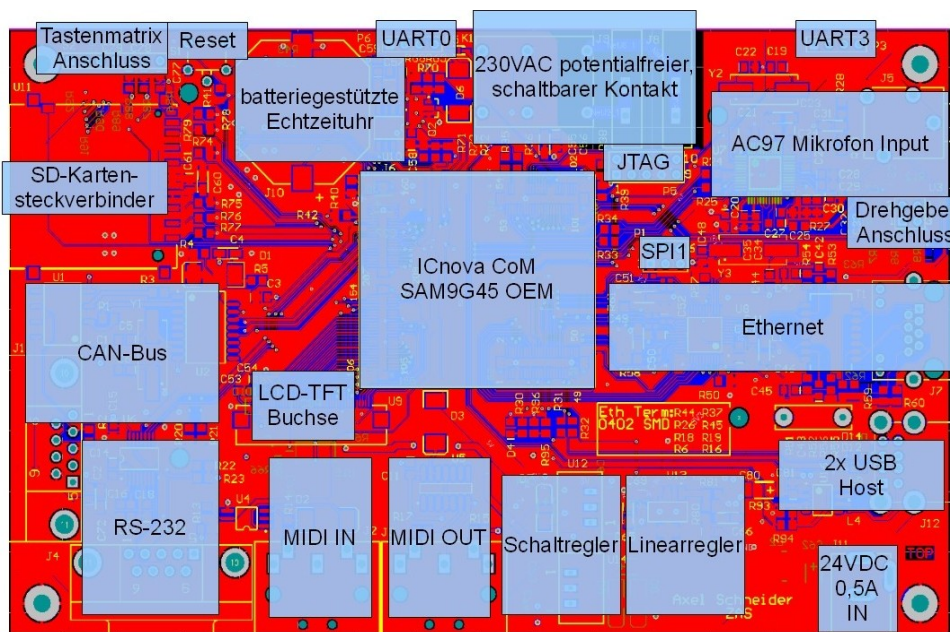


Abbildung 33: Übersicht über die Verteilung der Baugruppen auf der Leiterplatte

Die vollständige ZAS mit Display, Eingabeplatine und diversen Schnittstellen wird bereits zur Softwareentwicklung eingesetzt:



Abbildung 34: Vollständige ZAS mit Bedienschnittstelle bei der Entwicklung der Steuerungssoftware

4.3.1 Eigenschaften

Mit Display und Abstandsbolzen hat das ZAS Motherboard folgende Maße:

Länge	182mm
Breite	104mm
Höhe	45mm

Tabelle 10: Maße des ZAS Prototyps

Je nach Verwendung von verschiedenen Abstandsbolzen und Montagematerial kann die Höhe der Gesamtanordnung variieren.

ZAS	0 bis +70 °C
Eingabeplatine	0 bis +40 °C

Tabelle 11: Betriebstemperatur-Bereich des ZAS V1.0 Prototypen

Bei den in der Tabelle angegebenen Temperaturwerten handelt es sich um theoretische Werte, die durch die Datenblätter der verwendeten Bauelemente vorgegeben werden.

4.3.2 Hardwareinbetriebnahme und Funktionstest

Zur Inbetriebnahme der Hardware wird der ZAS Prototyp inklusive ICnova-SAM9G45OEM und Display, das 24 Volt Steckernetzteil und eine USB Tastatur benötigt. Das Einschalten der ZAS funktioniert über einfaches Anstecken des Steckernetzteils. Ist dies getan, vergehen etwa fünf Sekunden, bis das Display initialisiert wird. Das erfolgreiche Booten des Kernels ist an dem „LED-Heartbeat“ zu erkennen, der mit zwei kurz aufeinander folgenden Lichtimpulsen einen Herzschlag simuliert. Der erfolgreiche Bootvorgang ist auf dem Display oder ein Terminal mit dem Schriftzug „Welcome to In-Circuit ICnova“ zu erkennen, dass die Eingabe des Benutzernamens zum Zugang zur Konsole fordert.

4.4 EMV-Test

Der Test zur elektromagnetischen Verträglichkeit soll die Störfestigkeit des Geräts selbst und seine Störaussendung, die auf andere Geräte einwirken kann, prüfen.

Ein EMV Test ist in der Regel nur dann sinnvoll, wenn ein Gerät in seiner finalen Funktionalität getestet werden kann. Das bedeutet, dass die Steuerungssoftware der ZAS mit ihren Grundfunktionen und allen relevanten Geräten kommunizieren sollte. Bei diesem EMV Test wird dies so gut wie möglich durch ein permanentes Senden über CAN-Bus, aktives Ethernet im Hochschulnetzwerk, Anbringung des Displays und Verwendung einer USB Tastatur simuliert. Damit sind alle potentiell störenden Elemente abgedeckt.

4.4.1 Feldgekoppelte Störeinwirkung

Die ZAS wurde hierzu in einem Gerät zur aktiven elektromagnetischen Bestrahlung getestet. Sie erfährt dabei eine Bestrahlung in einem Frequenzspektrum mit bis zu 1 GHz.

Das Board hat den Störeinwirkungen stand gehalten, bis auf eine Besonderheit: Bei 500 MHz Frequenz eingekoppelter elektromagnetischer Wellen gibt der Ethernet Phy eine Art Warnung aus, was andeutet, dass feldgekoppelte Störeinwirkungen das Ethernet stören können. Die Link LED Blinkt dabei mit etwa 4 Hz. Der Betrieb ist dennoch weiterhin möglich.

4.4.2 Feldgekoppelte Störaussendung nach DIN EN 55022

Der Prototyp wurde auf feldgekoppelte Störaussendungen nach DIN EN 55022 getestet. Dazu wurde die ZAS in einem EMV-Labor in einer elektromagnetisch strahlungsneutralen Kammer auf einen Tisch gestellt und die verwendete Peripherie verbunden. Während der Messung dreht sich dieser Tisch um Emissionen von allen Seiten des Boards erfassen zu können. Die Antenne erfasst elektromagnetische Störaussendungen bis zu einer Frequenz von 3 GHz.

4.4.3 Vorbetrachtung

Folgende Tabelle bezeichnet die Grundfrequenzen bei denen Spitzenwerte im Frequenzspektrum eines Emissionstests für die ZAS auftreten können:

Baugruppe	ICnova-SAM9G450 EM	Schaltregler	Ethernet	AC97	TWI	LCD	SPI	MIDI
f_{Peak}	12 MHz	500 KHz	25 MHz	24,58 MHz	400 KHz	9,2 MHz	10 MHz	31,25 KHz

Tabelle 12: Trägerfrequenzen der Peripherie

Sie ergeben sich aus den verschiedenen Trägerfrequenzen der ZAS Hardware.

4.4.4 Messung

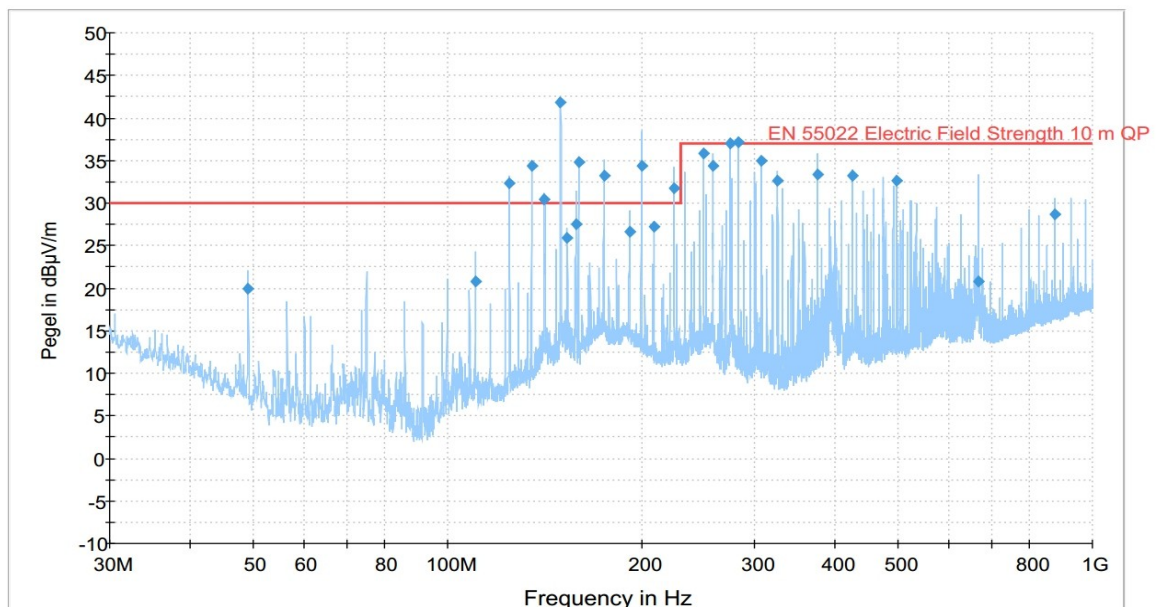


Abbildung 35: EMV-Test: Feldgekoppelte Störaussendung des ZAS Prototypen

Die rote Markierung stellt die Pegel-Grenzwerte für eine erfolgreiche Prüfung nach DIN EN 55022 dar. Die besonders herausragenden QuasiPeak Pegel sind mit blauen Punkten markiert. Die kritischen Frequenzen, die deutlich überhöhte Pegel bewirken, sind:

Fre- quenz (MHz)	Quasi- Peak (dB μ V/ m)	Grenz- wert (dB μ V/ m)
125	32,4	30
135,2	34,3	30
141,7	30,4	30
150	41,8	30
159,8	34,9	30
175	33,3	30
200	34,3	30
225	31,8	30
275	37	37
282,7	37,1	37

Tabelle 13: Kritische Frequenzen feldgekoppelter Störaussendung

Die Messanordnung besitzt eine Messgenauigkeit von $\Delta f = 0,1 \text{ MHz}$. Aus den Tabellen der Messung und der Vorbetrachtung lassen sich die Grundfrequenzen der Störaussendungen ermitteln. 125 MHz, 150 MHz, 175 MHz, 200 MHz und 225 MHz besitzen 25 MHz als Grundfrequenz. Sie sind also Oberwellen 5., 6., 7., 8. und 9. Ordnung. Diese stammt von der Ethernet Trägerfrequenz. 135,2 MHz, 159,8 MHz, 282,7 MHz entsprechen der Trägerfrequenz des AC97. Sie beträgt laut Datenblatt 12,288 MHz. Sie ergeben somit Oberwellen 11., 13. und 23. Ordnung. Für 141,7 MHz konnte keine Grundfrequenz ermittelt werden. Möglicherweise stammt sie ebenfalls vom AC97, die durch eine zusätzliche Frequenzteilung der Trägerfrequenz entsteht.

4.4.5 Auswertung

Die Störaussendungen stammen nach der erfolgten Analyse nahezu vollständig von den Baugruppen AC97 und Ethernet. Während das AC97 bei Nichtverwendung einfach deaktiviert werden kann, ist das Ethernet als kritisch zu betrachten. Sollte die ZAS über den Prototypenstatus hinaus kommerziell vertrieben werden, müssen zunächst die Störquellen identifiziert und falls möglich, abgestellt werden. Ansätze zur Verbesserung der Störaussendung können nicht deaktivierte Pins der Schnittstellen, oder gegebenenfalls Layoutkorrekturen sein.

4.4.6 Ausblick

Für zukünftige Leiterplattenlayouts kann auf Basis einer vierlagigen Leiterplatte durch einen Inselaufbau der Masseflächen mit kapazitiver Kopplung der Signalwege über das Basismaterial, ein impedanzkontrolliertes Routing mit Serienterminierung erfolgen. Mit einer Sternverbindung aller Masseflächen an einem zentralen Punkt verhindert man somit feldgekoppelte und galvanisch gekoppelte Wechselwirkungen durch Masseflächen, aufgrund von Stromimpulsen anderer Baugruppen. In der aktuellen zweilagigen Leiterplatte ist eine zuverlässige Impedanzkontrolle wegen der Einschränkungen vieler kreuzender Signale nicht möglich.

5 Softwarekonfiguration

In diesem Kapitel wird die Konfiguration des Embedded Linux Betriebssystems behandelt, auf dessen Basis die ZAS Hardware getestet wird.

Für die Software-Inbetriebnahme wurde ein ZAS-Motherboard Wiki (siehe [31]) in der Labor Embedded Control Wissensdatenbank mit diversen How-To's erstellt.

5.1 Spezifikation

Die Zentrale Aktor-Steuerung verfügt über ein Embedded Linux Betriebssystem. Es ist Grundlage für die Software zur Prozesssteuerung und damit eine ideale Plattform zur Überprüfung der Funktion des Hardwareprototypen. Das Betriebssystem übernimmt mit entsprechenden Treibern die Ansteuerung der Peripherie, um eine leicht anzusprechende Hardwareplattform zu bieten. Auf dieser Plattform aus Hardware und Betriebssystem baut die Zentrale-Aktor-Steuerungssoftware auf. Diese wird unabhängig von dieser Arbeit entwickelt und letztendlich auf der ZAS Plattform betrieben. Folgende Abbildung soll die Einordnung der in dieser Arbeit vorgenommenen Softwarekonfiguration verdeutlichen:

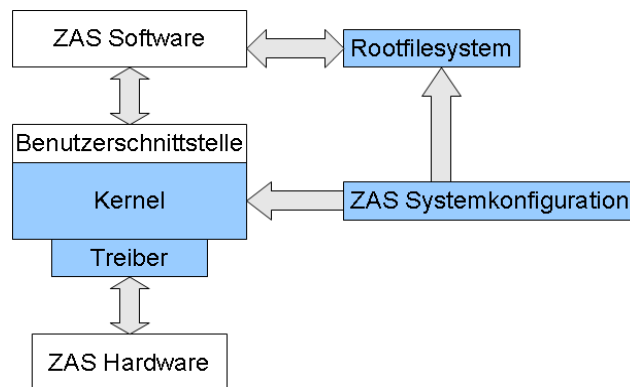


Abbildung 36: Mit eigener Konfiguration erstellte Bestandteile des Embedded Linux

Auf die blau dargestellten Elemente wird in der Linux Konfiguration Einfluss genommen. Die Aktualisierung des Betriebssystems wird über Ethernet-Dateidownload über HTTP oder FTP, oder den SD-Kartenslot ermöglicht.

5.2 Boardspezifische Implementation und Tests

Für die Erstellung eines auf die ZAS Anforderungen zugeschnittenen Kernels, wurde eine individuelle Boardkonfiguration angelegt, die `board-icnova_zas.c`. Diese befindet sich auf der Anlagen-CD. Sie ist das Bindeglied zwischen den eingebundenen Treibern im Kernel und der tatsächlich verwendeten Geräte. Sie definiert Treiber-Parameter der verwendeten Peripherie am AT91SAM9G45, damit diese beim Booten des Kernels korrekt angesteuert werden.

Die erstellte `board-icnova_zas.c` basiert auf der Konfiguration der Evaluationsplattform ADB1000, „`board-icnova_adb1000.c`“ von In-Circuit. Dabei wurden wesentliche Anpassungen an die ZAS in Form von zusätzlicher Treiberimplementation von Drehgeber, Tastenmatrix, Display und CAN-Bus vorgenommen. Vorhandene Treiberimplementationen der restlichen Peripherie wurden an die ZAS Pinbelegung angepasst.

5.2.1 Relaisansteuerung

Das Relais wird HI-aktiv über einen GPIO gemäß der Pinbelegungsliste angesteuert.

Zum Test wird es in der Boardkonfiguration einmalig beim Booten angesteuert. Dies erfolgt zeitgleich mit der Displayinitialisierung und ist hörbar. Es ist so implementiert, dass das monostabile Relais bei ausgeschaltetem Board die Kontakte öffnet. So können gegebenenfalls von der ZAS abhängige Geräte über dieses Relais geschaltet werden.

5.2.2 UART und Terminalkommunikation

Die auf der ZAS als Standard festgelegte, serielle Schnittstelle zur Terminalkommunikation, ist UART 0 mit einer Baudrate von 115200 Baud. Über eine USB-UART-Bridge kann so ein Zugriff auf die Linux Konsole erfolgen.

UART 1 und 2 sind für RS232 und MIDI reserviert. UART 3 ist, ebenso wie UART 0, herausgeführt. Sie ist keiner genauen Anwendung zugeteilt und kann somit frei verwendet werden.

Über UART 0 ist kein Zugang zum Bootloader möglich. Die `icnovaSAM9G45OEM` werden von In-Circuit mit einem vorkonfigurierten Bootloader ausgeliefert, der die serielle Schnittstelle zur Terminalkommunikation auf die serielle Atmel-Debugschnittstelle (DBGU) festgelegt hat, über die auch ein Zugriff auf den Bootloader erfolgen

kann. Auf der ZAS ist diese Schnittstelle nicht vorhanden. Sie kann jedoch auf UART 0 oder UART 3 gelegt werden, insofern der Bootloader neu konfiguriert wird. Die ZAS wurde so konfiguriert, dass ein Zugriff auf den Bootloader nicht notwendig ist.

5.2.3 Eingabeelemente

Der Drehgeber wird über den Open Source Treiber `rotary_encoder.c` implementiert. In der Boardkonfiguration werden dazu freie GPIO als LO-aktive Inputs definiert und die Eigenschaften des verwendeten Inkrementaldrehgebers eingepflegt.

Der Drehgeber ist als event-Input im Linux hinterlegt. Liest man diesen Input aus, werden bei Drehbewegungen binäre Zeichen in der Konsole ausgegeben.

Bei der Tastenmatrix kommt ein Treiber zum Einsatz, der die gedrückten Tasten nach Reihe und Spalte der Tastenanordnung auswertet. Dies hat den Vorteil, dass nur die Summe an Reihen und Spalten als GPIO benötigt werden und nicht die tatsächliche Anzahl an Tasten.

Die Tastenmatrix ist als event-Input im Linux hinterlegt. Über sie kann eine direkte Eingabe in die Konsole erfolgen.

5.2.4 Display Ansteuerung

Das NHD-4.3-480272MF-ATXI-1 Display basiert hardwareseitig auf dem Displaytreiber OTA5180A. Zur korrekten Ansteuerung des Hardwaretreibers, muss eine Anpassung der vertikalen und horizontalen Displaytimings, sowie der Aktualisierungsrate in der softwareseitigen Ansteuerung durch den Mikrocontroller vorgenommen werden. Diese werden über die Werte im OTA5180A Datenblatt (siehe [30] S. 25) bestimmt.

Das Display wird nach dieser Konfiguration erfolgreich zu Beginn des Kernel-Bootvorgangs initialisiert.

Zur zukünftigen Verwendung eines Touchpanels auf dem Display, ist der Treiber bereits aktiv in den Kernel eingebunden wurden. Es wurde so eingepflegt, dass ein zum aktuell verwendeten Display äquivalentes NHD-4.3-480272MF-ATXI-T-1 mit resistivem Touchpanel angesteckt werden kann. Im Betriebssystem ist das Touchpanel als „event0“ Input hinterlegt.

5.2.5 Ethernet

Der Test des Ethernet erfolgt über einen Dateidownload im Hochschulinternen Netzwerk. Dazu wurde eine Datei, in diesem Fall ein in den NAND-Flash zu schreibendes Rootfilesystem heruntergeladen.

```
Connecting to 141.55.66.147:8000... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 14,307,476 (14M) [application/octet-stream]  
  
100%[=====>] 14,307,476  3.22M/s  ETA 00:00  
  
17:06:02 (3.21 MB/s) - `rootfs.arm.jffs2' saved [14307476/14307476]
```

Abbildung 37: erfolgreicher Download einer Datei über Ethernet in den RAM des Icnova-SAM9G45OEM

In der Abbildung ist zu erkennen, dass das Ethernet die Datei mit einer Datenrate von 3,22 MB / s heruntergeladen hat. Dies entspricht einer Datenrate der 100 Mbit/s IEEE 802.3 Spezifikation. Auf diese Weise können Kernel und Rootfilesystem-Images bezogen und in eine NAND-Flash Partition geschrieben werden.

5.2.6 USB

Die USB Anschlüsse können mit USB Geräten bis zu einer Stromaufnahme von bis zu 250 mA betrieben werden. Bei überschreiten dieses Werts, verhindert ein IC übermäßige Stromaufnahme. Der Anschluss des im Projekt verwendeten Stimmgeräts ergibt folgende Ausgabe:

```
usb 2-2.5: FTDI USB Serial Device converter now attached to ttyUSB1
```

Abbildung 38: erfolgreiche Initialisierung eines USB-Geräts

5.2.7 Echtzeituhr

Die mittels I²C angesteuerte Echtzeituhr äußert ihre korrekte Funktion durch das Setzen der Systemzeit beim Booten des Kernels:

```
rtc-m41t80 0-0068: setting system clock to 2012-08-13 12:27:16 UTC (1344860836)
```

Abbildung 39: erfolgreiche Initialisierung der Echtzeituhr

5.2.8 MIDI

Zur Konfiguration des MIDI kommt ein Linux Hack zum Einsatz. Bei den Baudraten der seriellen Schnittstellen handelt es sich im Linux um tabellarische Werte, hinter denen die tatsächlichen Werte für die Baudrate variieren können. Dazu wird die Baudrate von 38400 Baud so manipuliert, dass sie der MIDI Baudrate von 31250 Baud entspricht.

Die Baudrate wird prinzipiell über die Peripheral Clock von 8333333 Hz und einem festgelegten Teiler bestimmt. Das Open Source Programm setserial, was auch in zukünftige Software einfließen kann, kann direkt auf den Treiber der seriellen Schnittstellen zugreifen und einen Custom Divisor für die Baudratenerzeugung setzen. Dieser Custom Divisor berechnet sich aus dem Quotient aus Peripheral Clock und gewünschter Baudrate:

$$y = \frac{8333333 \frac{1}{s}}{31250 \frac{1}{s}} = 266,66 \approx 267$$

Gleichung 19: Berechnung des Custom Divisor für den MIDI Output

Über das Programm setserial kann nun der Custom Divisor gesetzt werden. Verbindet man nun MIDI OUT und MIDI IN mit einem MIDI Kabel, kann man über die entsprechende UART 2 über MIDI Nachrichten ausgeben und einlesen.

Als Nachweis über die erzeugte MIDI Baudrate soll eine Messung am MIDI Out dienen.

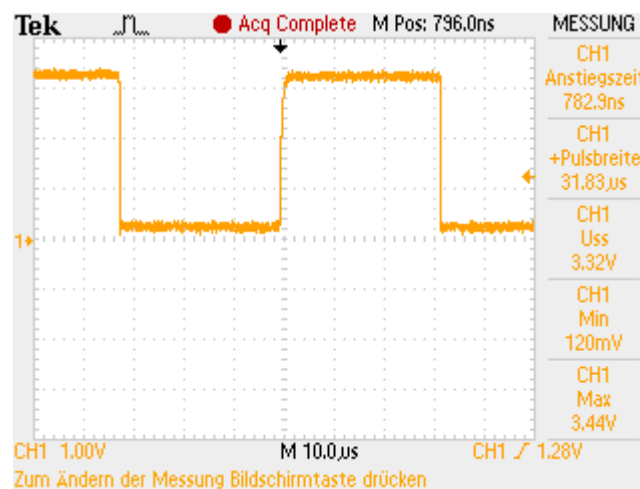


Abbildung 40: Messung der MIDI Pulsbreite

Die gemessene Pulsbreite beträgt 31,83 µs. Damit ergibt sich eine Baudrate von 31416 Baud, was annähernd der MIDI Baudrate entspricht.

5.2.9 SD-Speicherkarteninterface

Die Pinbelegung für das Speicherkarteninterface wurde gemäß der ZAS Pinbelegung [Anlage A] angepasst. Die frei definierbaren Write-Protection und Card-Detection Pins sind unabhängig von der Kommunikation des Karteninterface und wurden auf freie GPIO gelegt.

Das Speicherkarteninterface kann geprüft werden, indem ein Kernelimage als ulmage in das Stammverzeichnis einer SD-Karte abgelegt wird. Nun muss die SD-Karte eingesteckt und die ZAS eingeschaltet werden. Sie bootet nun das Kernelimage von der SD-Karte.

5.2.10 CAN Implementation

Die Betriebsbereitschaft des CAN wird mit folgender Nachricht beim booten signalisiert:

```
mcp251x spi0.0: CNF: 0x00 0x92 0x02
```

Abbildung 41: erfolgreiche Initialisierung des CAN

Das Senden und Empfangen über CAN mit 1 MBit/s Datenrate funktioniert prinzipiell. Beim Test des Empfangs von CAN Nachrichten hat sich jedoch herausgestellt, dass bei hoher Busauslastung, wenn die Dauer des Auslesens der MCP2515 Pufferregister durch den AT91SAM9G45 länger ist, als die Zeit zwischen zwei empfangenen CAN Nachrichten, der Empfang nicht zuverlässig erfolgt. Daraus resultiert der Verlust ganzer Nachrichten und das Verändern der Reihenfolge empfangener Nachrichten. Grund hierfür ist die große Zeitspanne des Linux Betriebssystems und Gerätetreibers beim Auslesen der CAN Puffer im MCP2515. Der CAN Controller fordert über einen separaten Interrupt Pin an den Mikrocontroller das Auslesen der Nachrichten via SPI Bus an. Dies erfolgt sofort nachdem eine Nachricht in einen der beiden Puffer geschrieben wurde. Damit teilt er dem Mikrocontroller mit, dass eine Nachricht eingetroffen ist und bereit ist, abgeholt zu werden. Der Mikrocontroller liest nun über SPI die Statusbits der Puffer aus, um festzustellen, welche Puffer voll sind. Ist einer oder sind beide der Puffer beschrieben, liest er einen Puffer aus, setzt dann das Status Bit

zurück und liest den zweiten Puffer aus. Die Messung in Abbildung 30 stellt dar, wie viel Zeit das Auslesen erfordert. Es ist die Zeit zwischen dem Setzen des Interrupts auf LO-Pegel, lila dargestellt, und der letzten SPI Aktivität, SPI Takt in blau dargestellt. Die grüne Messung stellt die SPI Informationen vom MCP2515 an den AT91SAM9G45 dar. Die letzte SPI Aktivität ist das Zurücksetzen der MCP2515 internen Puffer Interrupt Bits, damit die Puffer neu beschrieben werden können. Erst dann kann ohne Datenverlust eine neue CAN Nachricht empfangen werden:

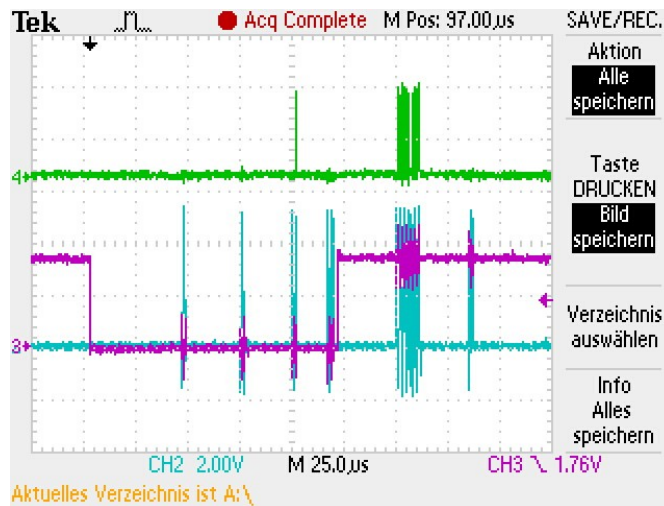


Abbildung 42: Dauer des Auslesens eines einzelnen MCP2515 Pufferregisters

Schlussfolgernd aus der Messung entspricht die Zeit, die dem AT91SAM9G45 in der aktuellen Konfiguration eingeräumt werden muss, um rechtzeitig die CAN Nachrichten auszulesen, mindestens:

$$t_{Read} = 180 \mu s \quad .$$

Gleichung 20: Dauer der Ausleseroutine des MCP2515 Puffer

Betrachtet man die Pausen zwischen dem SPI Takt, fällt auf, dass die SPI Aktivität in der Relation zur gesamten Ausleseroutine klein ist und insgesamt nicht länger als 50µs dauert.

Für zukünftige Revisionen der ZAS-Kernelkonfiguration kann eine preemptive Taskverwaltung für eine Verbesserung der Auslesegeschwindigkeit sorgen. Da die Abholung der CAN Nachrichten aus dem MCP2515 Puffer aufgrund ihrer Dauer und Komplexität kein Bestandteil der Interrupt Service Routine ist, müsste für den

MCP2515 Treiber ein preemptives Treibermodul erstellt werden, damit der CAN Nachrichtenempfang innerhalb einer Deadline erfolgen kann. Momentan ist der MCP2515 Treiber so implementiert, dass nach jeder abgeschlossenen Kommunikation mit dem MCP2515, unabhängig davon, ob Lesen oder Schreiben, neue Prozessorzeit angefordert wird und diese von der Taskverwaltung im Kernel neu bewertet wird. Dadurch entstehen für jede Anforderung Latenzzeiten, die in einer unannehmbaren Gesamtzeit von mindestens 180 Mikrosekunden resultieren. Für die Implementation eines preemptiven Kernels muss gegebenenfalls der Treiber des MCP2515 angepasst werden, oder seine Bestandteile in einen mit einer Deadline versehenen Rahmen eingepflegt werden.

Ein möglicher Workaround ist, CAS-seitig eine Verzögerung zwischen den CAN Nachrichten zu implementieren, sodass zwischen zwei CAN Nachrichten ausreichend Zeit für den MCP2515 besteht, dem Mikrocontroller zu signalisieren, dass der Puffer voll ist und gelesen werden muss. Über SPI liest der Mikrocontroller die Puffer aus, wodurch neue Nachrichten ohne Verluste in die Puffer geschrieben werden können. Für das Projekt ist eine Verzögerungszeit zwischen den CAN Nachrichten implementierbar, da der Empfang vieler aufeinanderfolgender CAN Nachrichten nur bei Wartung während der Abfrage von Kennliniendaten auftritt und nicht zeitkritisch ist. In der aktuellen CAS Firmware senden die CAS mit minimal möglicher Verzögerung zwischen zwei CAN Nachrichten von etwa zwanzig Mikrosekunden:

$$t_{mindelay} = 20 \mu s$$

Gleichung 21: Zeit zwischen zwei eintreffenden CAN Nachrichten von einer CAS

Für den fehlerfreien Empfang durch die ZAS, muss diese Verzögerung auf mindestens 100 Mikrosekunden verlängert werden. Der Wert ist abhängig von der Zeit, welche die kernelinterne Taskverwaltung bestimmt, Prozessorzeit für die Treiberfunktionen einzuräumen und kann mit steigender Prozessorlast größer werden oder durch Verwendung der Preemptivfunktionen niedriger werden. Eine Verzögerung ließe sich in der CAS Firmware mit einer Delay Funktion nach jedem Senden realisieren. Dazu muss noch die Dauer des Sendens eines CAN Frames bekannt sein:

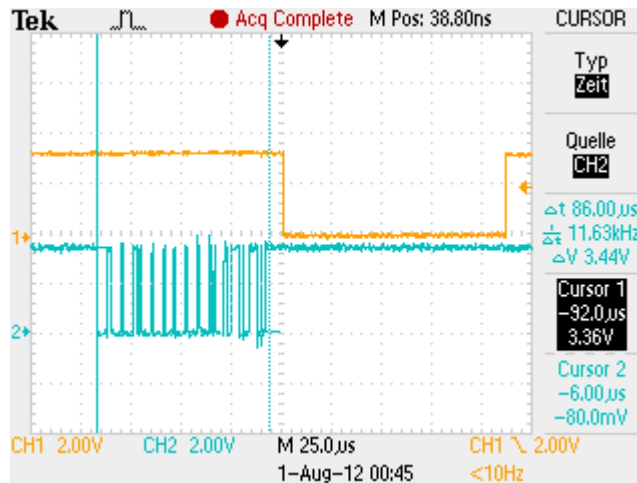


Abbildung 43: Dauer des Sendens eines CAN Frames durch eine CAS

Das hier blau dargestellte CAN RxD am MCP2515, zeigt dass ein CAN Frame in 86 Mikrosekunden übertragen wird.

$$t_{CANframe} = 86 \mu s$$

Gleichung 22: CAN Frame Übertragungsdauer

Mit etwas Toleranz ergibt sich aus den oben genannten Zeitwerten die Empfehlung für die Verzögerung nach dem Senden einer Nachricht durch eine CAS mit:

$$t_{delay} = t_{pread} - t_{CANframe} - t_{cansend} - t_{mindelay} = 180 \mu s - 90 \mu s - 20 \mu s = 70 \mu s$$

Gleichung 23: Empfohlene Verzögerungszeit zwischen zwei CAN Frames

6 Zusammenfassung und Ausblick

Das abschließende Kapitel dient der Zusammenfassung gewonnener Erkenntnisse und Ergebnisse zur Bewertung der eigenen Arbeit. Mögliches Potential für eine Weiterentwicklung werden im Ausblick aufgezeigt.

6.1 Zusammenfassung

Das Ziel der Diplomarbeit „Entwicklung einer updatefähigen Embedded-Linux-Hardwareplattform für den Einsatz in einer speziellen Gerätesteuerung“ war die Entwicklung der ZAS von der Spezifikation bis zum Prototypen. Das Resultat ist eine Embedded Linux Hardwareplattform, die ihre Funktion im dynamischen Stimmsystem erfüllt und für die Softwareentwicklung der ZAS Steuerungssoftware vorbereitet ist. Sie verfügt über die spezifizierten Schnittstellen und ein Bedieninterface, das für den Einsatz im Feld gemäß der Spezifikation optimiert wurde. Die Updatefähigkeit der Hardwareplattform ist durch mehrere Wege gewährleistet. So kann die ZAS-Software mit Hilfe einer SD-Karte, oder über Ethernet aktualisiert werden.

Bei der Einarbeitung in die Hardwareentwicklung der ZAS wurde klar, dass der Entwurf dieser Hardware mit einer zweilagigen Leiterplatte komplex ist und mit Beachtung aller kontrollierbaren Parameter durchgeführt werden muss. Dies führte dazu, dass die Arbeit wie folgt unterteilt wurde.

Das Kapitel 2 vermittelt das Grundverständnis der Hardware, die mit der ZAS in Verbindung stehen und gibt einen Einblick in die Umgebung des dynamischen Stimmsystems.

In Kapitel 3 wurden die Entwicklungsschritte von der Spezifikation ausgehend, in eine abstrakte Ebene überführt, auf dessen Basis die Umsetzung der Hardware stattfinden kann. Damit die Umsetzung in der Praxis gelingen konnte, wurde hier auf die Grundlagen der Elektronikentwicklung, sowie der verwendeten Peripherie eingegangen.

Kapitel 4 beschäftigt sich mit der Realisierung des ZAS Entwurfs. Zunächst erfolgte die Umsetzung der Benutzerschnittstelle und der Peripherie, bevor auf die tiefgreifende Dimensionierung der Baugruppen eingegangen wurde. Auf Basis der gewonnenen Erkenntnisse konnte der Schaltplangentwurf und das PCB-Design erfolgen. Das Resultat ist der ZAS Prototyp in der Version 1.0.

Damit eine Grundlage für die Softwareentwicklung vorhanden ist und die entwickelte Hardware getestet werden kann, erfolgte in Kapitel 5 die Konfiguration und Software-Implementation der ZAS-Peripherie in einem Embedded Linux.

6.2 Ausblick

Als erster Prototyp der ZAS, ist die Entwicklung der Hardwareplattform mit der Version 1.0 noch nicht vollendet. Es gibt mögliche Ansätze zur Weiterentwicklung.

Der EMV-Test zur Störabstrahlung war nicht erfolgreich. Für einen kommerziellen Vertrieb der ZAS Hardwareplattform muss die gültige Norm zur Störabstrahlung eingehalten werden. Mit einer Überprüfung des Ethernet und anderer relevanter Baugruppen kann die elektromagnetische Verträglichkeit verbessert werden.

Des Weiteren konnte die Konfiguration des Mikrofoneingangs noch nicht vollendet werden. Der Treiber ist implementiert und der Audio Controller aktiv, doch wird dieser noch nicht korrekt angesprochen. Ein Ansatz sind zu groß dimensionierte Terminierungswiderstände, die einer Neuanpassung bedürfen.

Für einen Einsatz des Relais als Schalter für Netzspannungsgeräte, muss dieses noch auf Spannungsfestigkeit geprüft werden. Dies muss in einem geeigneten Hochspannungsprüflabor erfolgen.

Die Computer-on-Modules von In-Circuit verfügen über einen vorkonfigurierten Bootloader, der nur über ein JTAG-Interface mit angepasster Software neu geschrieben werden kann. Da ein Ändern des Bootloaders immer mit Risiken verbunden ist, kann für diesen Zweck ein sicherer Ablauf zum Schreiben eines eigenen Bootloaders erstellt werden. Dies würde die Inbetriebnahme der ICnovaSAM9G45OEM in Zukunft vereinfachen und beschleunigen.

Literaturverzeichnis

[1]

Reiner Janke: Unser Tonsystem und seine Temperierung. URL:
<http://www.orgel-info.de/tempe-te.htm>, 23.07.2012

[2]

Axel Sikora / Rolf Drechsler: Software Engineering und Hardware-Design,
Verlag Hanser, Wien 2002, ISBN 3-446-21861-0

[3]

Georg Durcansky: EMV-gerechtes Gerätedesign, Verlag Franzis', Poing 1999,
ISBN 3-7723-5388-6

[4]

Wikipedia: Tiefpass URL:
<http://upload.wikimedia.org/wikipedia/commons/b/be/LCR-Tiefpass.png>,
28.07.2012

[5]

Manfred Seifart: Analoge Schaltungen, Verlag Technik, Berlin 2003, ISBN 3-
341-01298-2

[6]

Atmel Corp: AT91SAM9G45 Datasheet Rev. 6438G 19.04.2011

[7]

Wikipedia: Unixzeit. URL: <http://de.wikipedia.org/wiki/Unixzeit>, 23.07.2012

[8]

Linux-Tutorial.info: DMA. URL: http://www.linux-tutorial.info/Linux_Tutorial/The_Computer_Itself/The_Expansion_Bus/dma.png, 20.07.2012

[9]

Wikipedia: Struktur des Linux Kernels. URL:
http://upload.wikimedia.org/wikipedia/commons/thumb/f/f8/Linux_Kernel_Stuktur.svg/440px-Linux_Kernel_Stuktur.svg.png, 23.07.2012

[10]

Christopher Hallinan: Embedded Linux Primer, Verlag Pearson Education,
Boston 2011, ISBN 0-137-01783-9

[11]

Newhaven Display: NHD-4.3-480272MF-ATXI-1 Datasheet, 23.07.2012

[12]

Texas Instruments: LM317M Datasheet, Rev. 01.04.2000

[13]

Traco Power: TSR-1 Series Datasheet, Rev. 01.06.2011

[14]

Microchip: MCP2515 Datasheet, Rev. 08.04.2010

[15]

Micrel: MSZ8051MLL Datasheet, Rev. Juli 2010

[16]

Atmel Corp.: On-Chip devices setup code for the AT91SAM9G45 family. URL:
http://lxr.free-electrons.com/source/arch/arm/mach-at91/at91sam9g45_devices.c, 23.07.2012

[17]

Wolfhard Lawrenz: CAN Controller Area Network, Verlag Hüthig, Heidelberg
1994, ISBN 3-7785-2263-7

[18]

TOPAS: Ethernet Design Techniken. URL:
[http://www.topas.de/newsletter1.html?
&no_cache=1&tx_ttnews\[tt_news\]=178&cHash=a823245e572e0f287a5d
3926016f1275](http://www.topas.de/newsletter1.html?&no_cache=1&tx_ttnews[tt_news]=178&cHash=a823245e572e0f287a5d3926016f1275), 23.07.2012

[19]

Thomas Wedemeyer: Grundlegende Informationen zum CAN-Bus. URL:
<http://www.thomas-wedemeyer.de/uploads/file/can.pdf>, Rev. 14.04.2006

[20]

Texas Instruments: ISO1050 Datasheet, Rev. Juli 2009

[21]

Omron: Power PCB Relay G6C Datasheet, Rev. September 2011

[22]

Schaltrelais.de. Bemessungsbeispiele URL:
<http://www.schaltrelais.de/terminologie/bemessung.htm>, 01.08.2012

[23]

Multi-Circuit-Boards.eu: Leiterplatten-Materialien. URL: <http://www.multi-circuit-boards.eu/leiterplatten-design-hilfe/leiterplatten-materialien.html>, 05.08.2012

[24]

Pete Brown: MIDI Module Assembly Instructions. URL: <http://10rem.net/blog/2012/03/31/midi-module-assembly-instructions>, 05.08.2012

[25]

Mikrocontroller.net: I2C. URL: <http://www.mikrocontroller.net/articles/I2C>, 05.08.2012

[26]

Mikrocontroller.net: DMA. URL: <http://www.mikrocontroller.net/articles/DMA>, 05.08.2012

[27]

In-Circuit: ICnovaSAM9G45OEM Datasheet, Rev. B Februar 2010

[28]

In-Circuit: Support CD. URL: http://www.ic42.de/Linux-Images/icnova_20110104.iso, 12.08.2012

[29]

Wikipedia: Tiefpass. URL: <http://de.wikipedia.org/wiki/Tiefpass>, 23.07.2012

[30]

Orise Tech: OTA5180A Datasheet. Preliminary Version 01.04.2008

[31]

Labor Embedded Control: ZAS Motherboard Dokumentation. URL: <http://mpt-www.eit.hs-mittweida.de/twiki/bin/view/Labweb/ZASMotherboard>, 15.08.2012

Anlagen

Anlage A – AT91SAM9G45 Pinbelegung der ZAS

PIN Hirose	Port/Pin AT91SAM9G45	ICnova SAM9G45 OEM	ZAS Motherboard Funktion
			Grundfunktion (- Pinbeschreibung)
Hirose 1	1	GND	GND
	2	PA10	ETH - TXD0
	3	PC0	USB3_PSWN/DQM2
	4	PA11	ETH - TXD1
	5	PC1	USB3_OCN/DQM3
	6	PA29	ETH - CRS
	7	GND	GND
	8	PA26	ETH_RESETh/MC11_DA3/TIOB2
	9	GND	
	10	PA30	ETH - COL
	11	PD31	TIOB1/PWM1
	12	PA25	ETH_PDWN/MC11_DA2/PWM3
	13	PD30	TIOB0/SCK2
	14	PC6	ETH_MDINTRn/A23
	15	PD28	SPI_NPCS1/TSADTRG
	16	PA27	PowerLED
	17	GND	
	18	PA6	ETH - TXD2
	19	PC0	USB2_PSWN/DQM2
	20	PA7	ETH - TXD3
	21	PC1	USB2_OCN/DQM3
	22	PA14	ETH - TXEN
	23	PC0	USB - PWSN (1&2)
	24	PA17	ETH - TXC
	25	PC1	USB - OCN (1&2)
	26	PA12	ETH - RXD0
	27	GND	
	28	PA13	ETH - RXD1
	29	USB-Host3 +	USB-Host3 +
	30	PA8	ETH - RXD2
	31	USB-Host3 -	USB-Host3 -
	32	PA9	ETH - RXD3
	33	GND	
	34	PA16	ETH - RXER
	35	USB-Host2 +	USB-Host2 +
	36	PA28	ETH - RXC
	37	USB-Host2 -	USB-Host2 -
	38	PA15	ETH - RXDV
	39	GND	
	40	PA19	ETH - MDIO
	41	USB-Host1 +	USBHost1 +
	42	PA18	ETH - MDC
	43	USB-Host1 -	USBHost1 -
	44	GND	
	45	GND	
	46	GND	
	47	PE31	PCK1/PWM2
	48	NC	
	49	GND	
	50	NC	
	51	NC	non plated hole
	52	NC	non plated hole

Hirose 2

PIN Hirose	Port/Pin AT91SAM9G45	ICnova SAM9G45 OEM	ZAS Motherboard Funktion
			Grundfunktion (- Pinbeschreibung)
53	PD8	AC97FS/TIOB5	AC97 - SYNC
54	GND		
55	PD7	AS97TX/TIOA5	AC97 - SDO
56	HHSDMB/DHSDM		USBHost2 -
57	PD9	AC97CK/TCLK5	AC97 - SCLK
58	HHSDPB/DHSDP		USBHost2 +
59	PD6	AC97RX	AC97 - SDI
60	GND		
61	PB14	SPI1_MISO	SPI1 - MISO
62	GND		
63	PB31	ISI_MCK/PCK1	
64	GND		
65	GND		
66	GND		
67	GND		
68	TDI		
69	PB11	ISI_DAT11/TWI1_SCL	
70	NTRST		JTAG - NTRST
71	PB10	ISI_D10/TWI1_SDA	
72	NC		
73	PB9	ISI_D9/UART3_RXD	UART3 - RXD
74	RESETn		JTAG - RESETn
75	PB8	ISI_D8/UART3_TXD	UART3 - TXD
76	TMS		JTAG - TMS
77	PB28	ISI_PCLK	Tasten - COL3
78	TDO		JTAG - TDO
79	PB29	ISI_VSYNC	Tasten - COL2
80	TCK		JTAG - TCK
81	PB30	ISI_HSYNC	Tasten - COL1
82	GND		
83	PB27	ISI_DAT7	Tasten - TOW4
84	GND		
85	PB26	ISI_DAT6	Tasten - ROW3
86	GND		
87	PB25	ISI_DAT5	Tasten - ROW2
88	PD3	RDO	Relais (active HI)
89	PB24	ISI_DAT4	Tasten - TOW1
90	PD2	TD0	
91	PB23	ISI_DAT3	Drehgeber - A (active LO)
92	PD1	TF0	
93	PB22	ISI_DAT2	Drehgeber - B (active LO)
94	GND		
95	PB21	ISI_DAT1	Drehgeber - SW (active LO)
96	PD0	TK0/PWM3	
97	PB20	ISI_DAT0	
98	PD4	RK0	
99	GND		
100	PD5	RF0	
101	GND		
102	GND		
103	NC	non plated hole	
104	NC	non plated hole	

Hirose 3

PIN Hirose	Port/Pin AT91SAM9G45	ICnova SAM9G45 OEM	ZAS Motherboard Funktion
			Grundfunktion (- Pinbeschreibung)
105	PE30	LCDD23	LCD - B7
106	GND	GND	
107	PE29	LCDD22	LCD - B6
108	PE23	LCDD16/LCDD22	LCD - B0
109	PE28	LCDD21	LCD - B5
110	PE22	LCDD15/LCDD21	LCD - G7
111	PE27	LCDD20	LCD - B4
112	PE21	LCDD14/LCDD20	LCD - G6
113	GND		
114	PE20	LCDD13/LCCD19	LCD - G5
115	PE26	LCDD19	LCD - B3
116	PE19	LCDD12/LCDD18	LCD - G4
117	PE25	LCDD18	LCD - B2
118	PE18	LCDD11/LCDD15	LCD - G3
119	PE24	LCDD17/LCDD23	LCD - B1
120	PE17	LCDD10/LCDD14	LCD - G2
121	GND		
122	PE16	LCDD9/LCDD13	LCD - G1
123	GND		
124	PE15	LCDD8/LCDD12	LCD - G0
125	PD27	GPAD7/SPI0_NPCS3/PCK1	
126	PE14	LCDD7/LCDD11	LCD - R7
127	PD26	PCK0/PWM2/GPAD6	
128	PE13	LCDD6/LCDD10	LCD - R6
129	GND		
130	PE12	LCDD5/LCDD7	LCD - R5
131	PD23	TCLK0/TSAD3	TS - YD
132	PE11	LCDD4/LCDD6	LCD - R4
133	PD22	TIOA2/TSAD2	TS - YU
134	PE10	LCDD3/LCDD5	LCD - R3
135	PD21	TIOA1/TSAD1	TS - XL
136	PE9	LCDD2/LCDD4	LCD - R2
137	GND		
138	PE8	LCDDAT1/LCDD3	LCD - R1
139	PD20	TIOA0/TSAD0	TS - XR
140	PE7	LCDD0/LCDD2	LCD - R0
141	PD19	SPI1_NPCS3/FIQ	
142	PE0	LCDPWR/PCK0	LCD - DISP
143	PD18	SPI1_NPCS2/IRQ	CAN - INT
144	PE1	LCDMOD	
145	GND		
146	PE6	LCDDEN/LCDDVAL	LCD - DE
147	PA24	MCI1_DA1/CTS3	
148	PE3	LCDSYNC	LCD - VSYNC
149	PA23	MCI1_DA0/RTS3	LCD - LED EN (active HI)
150	PE5	LCDDOTCK	LCD - PCLK
151	PA22	MCI1_CDA/SCK3	AC97 - RST
152	PE4	LCDHSYNC	LCD - HSYNC
153	GND		
154	PE2	LCDC	
155	NC	non plated hole	
156	NC	non plated hole	

Hirose 4

PIN Hirose	Port/Pin AT91SAM9G45	ICnova SAM9G45 OEM	ZAS Motherboard Funktion
			Grundfunktion (- Pinbeschreibung)
157	NC	non plated hole	
158	NC	non plated hole	
159	ARM9_EN		5V pullup
160	VCC	3,3V - 5V	5V
161	VCC	3,3V - 5V	5V
162	VCC	3,3V - 5V	5V
163	VCC	3,3V - 5V	5V
164	VCC	3,3V - 5V	5V
165	VCC	3,3V - 5V	5V
166	VCC	3,3V - 5V	5V
167	GND		
168	GND		
169	GND		
170	GND		
171	GND		
172	GND		
173	GND		
174	GND		
175	WKUP	WAKEN	3,3V pullup
176	GND		
177	NRST	RESET_OUT	rReset Button (active LO)
178	PB16	SPI1_SPCK/SCK0	SPI1 - SCK
179	PD29	TCLK1/SCK1/(Card detect)	
180	PB18	RXD0/SPI0_NPCS1	UART0 - RxD
181	PB5	RXD1	UART1 - RxD
182	PB19	TXD0/SPI0_NPCS2	UART0 - TxD
183	GND		
184	PB15	CTS0/SPI1_MOSI	SPI1 - MOSI
185	PB4	TXD1	UART1 - TxD
186	PB17	RTS0/SPI1_NPCS0	SPI1 - NPCS0
187	PD16	RTS1	UART1 - RTS
188	GND		
189	PD17	CTS1	UART1 - CTS
190	PB6	TXD2	UART2 - TxD2
191	GND		
192	PB7	RXD2	UART2 - RxD
193	PB12	DRXD	
194	PA20	TWDO	
195	PB13	DTXD	
196	PA21	TWCK0	
197	PB0	SPI0_MISO	SPI0 - MISO
198	PA0	MCIO_CK/TCLK3	MCIO - MMC2
199	PB1	SPI0_MOSI	SPI0 - MOSI
200	PA1	MCIO_CDA/TIOA3	MCIO - MMC1
201	PB2	SPI0_SCK	SPI0 - SCK
202	PA2	MCIO_DA0/TIOB3	MCIO - MMC3
203	PB3	SPI0_NPCS0	SPI0 - NPCS0
204	PA3	MCIO_DA1/TCLK4	MCIO - MMC4
205	PD24	SPI0_NPCS1/PWM0/GPAD4	MCIO - MMC_CD
206	PA4	MCIO_DA2/TIOA4	MCIO - MMC5
207	PD25	SPI0_NPCS2/PWM1/GPAD5	MCIO - MMC_WP
208	PA5	MCIO_DA3/TIOB4	MCIO - MMC6

Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch in keiner anderen Prüfungsbehörde vorgelegt.

Leipzig, den 16.08.2012

Axel Schneider