



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorthesis

Frederic Alexander Hermann Adler

Entwicklung einer Evaluierungsplattform
für drahtlose und drahtgebundene
FlexRay-Datenkommunikation im Automobil

Frederic Alexander Hermann Adler
Entwicklung einer Evaluierungsplattform für
drahtlose und drahtgebundene
FlexRay-Datenkommunikation im Automobil

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

Betreuender Prüfer : Prof. Dr. Lutz Leutelt
Zweitgutachter : Prof. Dr. Ulrich Sauvagerd

Abgegeben am 12. April 2015

Frederic Alexander Hermann Adler

Thema der Bachelorthesis

Entwicklung einer Evaluierungsplattform für drahtlose und drahtgebundene FlexRay-Datenkommunikation im Automobil

Stichworte

Datenkommunikation im Automobil, FlexRay, Hardwareentwicklung, PCB, Mikrocontroller, TMS570LS3137, Deterministisches Bussystem

Kurzzusammenfassung

Diese Bachelorthesis beschreibt die Entwicklung und Verifikation einer Evaluierungsplattform für FlexRay-Teilnehmer. Die entwickelte Platine bietet sowohl FlexRay-Bustreiber für ein drahtgebundenes System als auch die Verbindung und Spannungsversorgung für ein FPGA-Modul und ein Modul für eine zukünftige drahtlose Kommunikation. Des Weiteren dient eine angepasste Mikrocontroller-Software der Verifizierung der FlexRay-Datenkommunikation.

Frederic Alexander Hermann Adler

Title of the paper

Development of an evaluation board for wireless and wired automotive FlexRay data transmission

Keywords

In-vehicle data communication, FlexRay, hardware development, PCB, microcontroller, TMS570LS3137, deterministic bus communication

Abstract

This bachelor thesis describes the development and verification of an evaluation board for a FlexRay node. The developed PCB offers bus transceivers for a wired system as well as connections and power supply for a FPGA board and a module for future wireless transmission. In addition a modified microcontroller software serves to verify the FlexRay data transmission.

Inhaltsverzeichnis

Tabellenverzeichnis	6
Abbildungsverzeichnis	7
1. Einleitung	9
1.1. Motivation	9
1.2. Kapitelübersicht	12
2. Grundlagen FlexRay	13
2.1. Aufbau eines FlexRay-Knotens	14
2.2. Buspegel	14
2.3. Topologien	16
2.4. Der FlexRay-Kommunikationszyklus	17
2.5. Framing einer FlexRay-Botschaft	19
2.6. FlexRay Zeithierarchie	20
3. Anforderungen an die Evaluierungsplattform	23
3.1. Anforderungsanalyse	23
3.2. Entscheidung für Mikrocontroller- und FPGA-Modul	26
4. Hardware	30
4.1. Abschätzung der Stromaufnahme	30
4.2. Spannungsversorgung	33
4.2.1. Eingangsschutzschaltung	34
4.2.2. Boardspannungsnetz für 5V und 3,3V	35
4.2.3. Spannungsnetz für das FPGA-Modul	38
4.3. FlexRay-Transceiver	41
4.4. Signalleitungen der Evaluierungsplattform	43
4.5. Layout der Platine	46
5. Mikrocontroller Software	50
5.1. Aufbau und Programmablauf der Software	50
5.2. Protokollzustandsmaschine des FlexRay-Controllers	54
5.3. Konfiguration des FlexRay-Controllers	56

6. Messergebnisse	57
6.1. Testmessung der Hardware-Komponenten	57
6.1.1. Verifikation der Spannungsversorgung	59
6.1.2. Stromaufnahme der Evaluierungsplattform	62
6.1.3. Laufzeitmessung der FlexRay-Signale über das FPGA-Modul	64
6.2. Verifizierung der Software-Funktionalität	65
6.2.1. Verifikation der Zeitdauer eines Kommunikationszyklus	65
6.2.2. Verifikation der SPI-Schnittstelle	67
6.2.3. Nutzdatenauswertung der FlexRay-Kommunikation	68
6.3. Optimierung der Evaluierungsplattform	72
7. Zusammenfassung und Ausblick	73
A. Unterlagen zur Evaluierungsplattform	74
A.1. Übersichtstabelle zu den Signalleitungen der Evaluierungsplattform	74
A.2. PCB Layout der Evaluierungsplattform	75
A.3. Schaltplan der Evaluierungsplattform	78
A.4. Schaltfrequenz des 5V Schaltreglers	84
A.5. Messbilder der FlexRay-Datenkommunikation	85
A.6. Reflow-Ofen	88
Literaturverzeichnis	89
Abkürzungsverzeichnis	96

Tabellenverzeichnis

3.1. Vergleich der wichtigsten Funktionen der beiden Mikrocontroller der Endauswahl [Tex15f], [Tex15d]	27
3.2. Features des Texas Instruments TMS570LS31x Hercules Development Kit [Tex13c]	28
3.3. Vergleich der beiden MicroZed-Platinen der Endauswahl [Xil14], [Avn15a] . .	29
4.1. Stromaufnahme der Komponenten	32
4.2. Signalleitungen der Node Transceiver (vgl. A.1)	43
4.3. Wichtigste Signalleitungen des Mikrocontroller-Moduls (vgl. A.1)	44
4.4. Signalleitungen für die Kommunikation zwischen FPGA- und Drahtlos-Modul (vgl. A.1)	45
4.5. Strombelastbarkeit von 35µm dicken Kupferleiterbahnen in Abhängigkeit der Leiterbahnbreite	47
4.6. Leiterbahnbreite der jeweiligen Signal- bzw. Spannungsnetze auf der realisierten Platine	47
4.7. Prozessablauf des Reflow-Lötvorgangs	48
5.1. Baudraten und resultierende relative Fehler	52
6.1. Übersicht der für die Messungen verwendeten Geräte	59
6.2. Messwerte Spannungen der Spannungsversorgung	60
6.3. Maximale Stromaufnahme der Hardware in verschiedenen Betriebsarten . .	63
6.4. Übersicht des Statusregisters der Node Transceiver [NXP10, S.22]	68
A.1. Übersicht aller Signalleitungen der Evaluationsplattform	75

Abbildungsverzeichnis

1.1. Diagramm des geplanten Clusters	11
2.1. Schematischer Aufbau eines FlexRay-Knotens	14
2.2. Darstellung der FlexRay-Buspegel [Vec15a]	15
2.3. Schematischer Aufbau verschiedener Topologien eines FlexRay-Clusters . .	17
2.4. Schematische Darstellung des FlexRay-Kommunikationszyklus nach Time Division Multiple Access (TDMA)-Verfahren [Wik15b]	18
2.5. Übersicht des Aufbaus einer FlexRay-Botschaft [Vec15c]	20
2.6. „Abhängigkeiten zwischen Bitrate und Microtick-Länge“ [Rau07, Bild 6.4] . . .	21
2.7. Schematische Darstellung der FlexRay Zeithierarchie [Tem]	22
4.1. Hierarchischer Aufbau der Spannungsversorgung auf der Evaluierungsplattform	33
4.2. Schaltplan der Schutzschaltung für die 12V Spannungsversorgung	34
4.3. Beschaltung des 5 V Abwärtswandlers	36
4.4. Beschaltung des 3,3V Abwärtswandlers und der Status-LEDs der Spannungsversorgung	37
4.5. Sequenzdiagramm der MicroZed-Spannungsversorgung während des Startvorgangs [Avn15c, S. 29]	38
4.6. Schaltplan für die zusätzliche Spannungsversorgung der FPGA-Aufsteckplatine	40
4.7. Ausschnitt zur Beschaltung der FlexRay-Transceiver TJA1082 (hier gezeigt wird Kanal A)	41
4.8. PCB Layout für die Sektion der 5 V und 3,3 V Spannungsversorgung	48
4.9. Die Oberseite der realisierten Platine (Revision 1.5, Prototyp 1)	49
5.1. Haupt-Programmkomponenten der Software für Node A	52
5.2. Programmablauf der Hauptprogramme von Node A (rechts) und Node B (links)	53
5.3. Protokollzustandsdiagramm des E-Ray FlexRay-Controllers [Rob09, S. 107] .	55
6.1. Foto des Messaufbaus für die drahtgebundene FlexRay-Datenkommunikation	58
6.2. Startvorgang der Spannungsversorgung für das 5 V und 3,3 V Spannungsnetz	60
6.3. Schematische Darstellung des zeitlichen Ablaufs der MicroZed-Spannungsversorgung während des Startvorgangs [Avn14b, S. 13]	61
6.4. Startvorgang der Spannungsversorgung für das MicroZed	62

6.5. Messung der Laufzeitdifferenz zwischen Signals FRAY_TX1 und MOD_DGQ_1	64
6.6. Messung der ersten beiden Cycle Counts	66
6.7. Screenshot des Terminalprogramms für die seriellen Kommunikation zwischen Mikrocontroller und Rechner	67
6.8. Beschaltung des Heißeiters auf dem TI Hercules Development Kit [Tex12, S.14]	69
6.9. Ausschnitt der PicoScope-Messung einer vollständigen FlexRay-Datenkommunikation (vgl. Abbildung A.11)	71
A.1. PCB Layout - Evaluierungsplattform Revision 1.5	76
A.2. PCB Layout - Evaluierungsplattform Revision 2	77
A.3. Schaltplan - Projektseite 1: Spannungsversorgung	78
A.4. Schaltplan - Projektseite 2: FlexRay Transceiver (beide Kanäle)	79
A.5. Schaltplan - Projektseite 3: Steckverbinder zum TI HDK	80
A.6. Schaltplan - Projektseite 4: Spannungsversorgung für MicroZed	81
A.7. Schaltplan - Projektseite 5: Steckverbinder zum MicroZed	82
A.8. Schaltplan - Projektseite 6: Steckverbinder zum Drahtlos-Modul	83
A.9. Screenshot PicoScope - Schaltfrequenz 5 V Schaltregler	84
A.10.Screenshot PicoScope - FlexRay-Botschaft	85
A.11.Screenshot PicoScope - FlexRay-Zyklus	86
A.12.Screenshot PicoScope - 3 statische FlexRay-Botschaften	87
A.13.Reflow-Ofen (links) mit temperaturgesteuertem Netzregler (rechts)	88

1. Einleitung

Diese Bachelorthesis beschreibt die Entwicklung und Verifikation einer Evaluierungsplattform für FlexRay-Datenkommunikation. Dabei legt sie den Grundstein für die Untersuchung möglicher Lösungen einer drahtlosen Datenkommunikation mit FlexRay und anderen Bussystemen im übergeordneten Urban Mobility Lab der HAW Hamburg.¹ Dieses Kapitel beschreibt zunächst die Motivation für eine drahtlose Datenkommunikation im Automobil und die Verwendung von FlexRay. Abschnitt 1.2 bietet anschließend eine Übersicht der Kapitel und beschreibt kurz deren Inhalt.

1.1. Motivation

Systeme zur effizienten Kraftstoffnutzung, komfortable Mediasysteme und Sicherheitsfunktionen wie ein Antiblockiersystem (ABS) sind mittlerweile im Automobil zur Normalität geworden. Die Anforderungen der Endkunden an ihr Automobil sind hoch und können nur noch mithilfe von elektronischen Systemen erfüllt werden. Insbesondere im Bereich der Fahrassistenzsysteme, welche den Endnutzer in kritischen Situationen warnen bzw. assistieren sollen, steigt die Akzeptanz solcher Systeme [Vec15b]. Nachteil dieser Systeme, die eine Vielzahl von elektrischen und elektronischen Komponenten beinhalten, die über das Automobil verteilt sind, ist jedoch die damit stetig steigende Komplexität der Vernetzung im Auto. Nicht nur die Materialkosten der Verkabelung spielen hier eine Rolle, sondern auch das Gewicht, welches zu erhöhtem Kraftstoffverbrauch führt. Hinzu kommt die konstruktive Herausforderung bei der Entwicklung und Verlegung der Kabelbäume. Mit der breiten Einführung der seriellen Bussysteme wie Controller Area Network (CAN) und Local Interconnect Network (LIN) in den 1990er Jahren, konnte die Komplexität der Verkabelung deutlich reduziert werden. Durch die vielen zusätzlichen Funktionen im Kfz stellt diese Verkabelung jedoch wieder eine große Herausforderung dar. Daher entsteht in der Automobilindustrie der Wunsch, zumindest einen Teil der Funktionen auf drahtlose Datenkommunikation umzustellen. Neben dem Einsparen von Kabeln ist es beispielsweise auch denkbar, die Informationsdaten für Service-Kräfte zukünftig kontaktlos zu übermitteln. Bei der eigentlichen Produktion des Fahrzeugs kann zudem der

¹Für nähere Informationen bezüglich des Lehrangebots siehe [HAW15]

aufwendige Einbau von Kabelbäumen vermieden werden, da die drahtlos-Module nur eine einmalige Planung in Bezug auf die Elektromagnetische Verträglichkeit (EMV) erfordern.

Diese Arbeit hat zum Ziel, die technische Basis bereitzustellen, um heutige Datenkommunikationssysteme im Auto durch drahtlose Systeme zu ersetzen und diese zu evaluieren. In einem ersten Schritt soll ein bestehender Standard auf der physikalischen Schicht (Physical Layer), von einer kabelgebunden auf eine kabellose Übertragung umgestellt und die Machbarkeit untersucht werden. Die Wahl fällt dabei auf das aktuelle Bussystem FlexRay [Wik15b]. Bereits 2007 zeigte die Firma BMW das Problem bis dato vorhandener Bussysteme am Beispiel des weit verbreiteten Feldbus CAN (vgl. [Ant07]) auf. Die steigende Komplexität der Systeme erforderte hohe Übertragungsraten, die CAN nicht ohne weiteres liefern kann. FlexRay bietet eine im Vergleich zu CAN zehnmal höhere Bitrate von 10 MBit/s und ist zudem prädestiniert für bestimmte Fahrerassistenzsysteme mit hohen Sicherheitsanforderungen, die beispielsweise Redundanz, garantierte Latenzzeiten sowie weitere Fehlerschutzmaßnahmen erfordern. Das hohe Sicherheitsrisiko bei der Anwendung von Brake-by-Wire (wörtlich: Bremsen per Draht) oder Steer-by-Wire (wörtlich: Lenken per Draht) kann auch FlexRay nicht gänzlich beheben, stellt aber einen großen Schritt in Richtung vollelektronischer Fahrerassistenzsysteme dar.

Um mögliche Funklösungen zu untersuchen und mit dem konventionellen drahtgebundenen Übertragungspfad zu vergleichen, wird ein geeignetes Testsystem benötigt. Im Rahmen dieser Arbeit wird dafür eine Evaluierungsplattform entwickelt, welche die geeigneten Hardware-Komponenten für diese Untersuchung bereitstellt und untereinander verbindet. Die realisierte Platine stellt zum einen die für die drahtgebundene Übertragung notwendigen Pegelumsetzer (FlexRay-Transceiver), sowie ein Mikrocontroller-Modul zur Verfügung, welches den eigentlichen FlexRay-Controller beinhaltet. Zum anderen dient ein leistungsfähiges Field Programmable Gate Array (FPGA)-Modul der digitalen Signalverarbeitung eines zukünftigen Modulationsverfahrens. Ein flexibler Steckplatz für ein mögliches Drahtlos-Modul bereitet zudem die Umstellung der drahtgebundenen Übertragung auf eine Funkübertragung vor.

Die Abbildung 1.1 bietet eine Übersicht des geplanten FlexRay-Testsystems. Eine solche Gruppe von Teilnehmern wird als Cluster bezeichnet, die Teilnehmer selbst als Nodes (engl. Knoten). Die Diagrammblocke *Teilnehmer* entsprechen jeweils einer Evaluierungsplattform. Die Diagrammblocke mit gestrichelter Umrandung repräsentieren die benötigte Hardware für die Feldsysteme CAN und LIN, welche zunächst nur optional zur Verfügung stehen sollen.

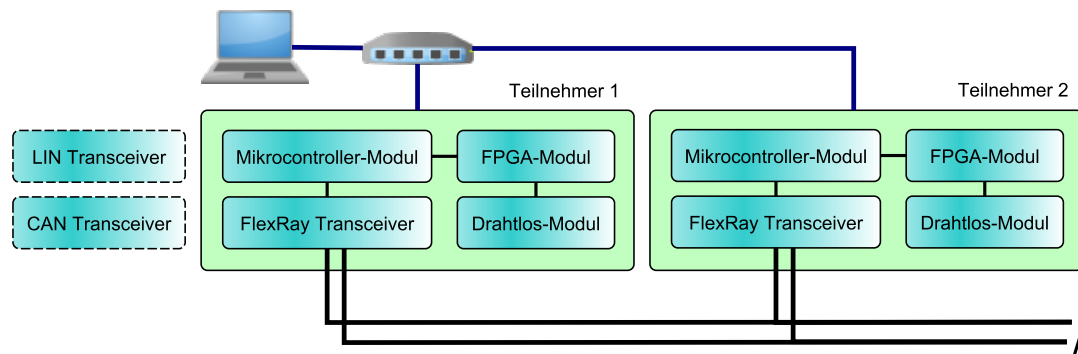


Abbildung 1.1.: Diagramm des geplanten Clusters

Die Verbindung der unterschiedlichen Busteilnehmer mit einem Computernetzwerk (Datenkommunikation über Ethernet) dient der Konfiguration von Parametern des Bussystems. Zukünftig kann dazu eine graphische Benutzeroberfläche entwickelt werden, um möglichst komfortabel verschiedene Konfigurationen zu laden und zu überwachen. Eine weitere Aufgabe ist die Überwachung des Datenverkehrs auf dem Bus durch einen oder mehrere Teilnehmer. Die hier gewonnenen Daten können dann über Ethernet an ein Display gesendet werden. Dadurch wird das sicherheitskritische FlexRay-System vollständig von den Komfort-Funktionen getrennt.

Insbesondere in der Automobilbranche oder der Luftfahrt gelten höchste Sicherheitsstandards, um das Risiko etwaiger Personenschäden zu minimieren. Eine Funkübertragung stellt gegenüber einer drahtgebundenen Übertragung eine größere technische Herausforderung dar. Es gilt also zu eruieren, ob und unter welchen Voraussetzungen ein Drahtlos-System überhaupt einsetzbar ist. Die Evaluierungsplattform bietet das benötigte Testumfeld, um die nötigen Untersuchungen durchzuführen.

1.2. Kapitelübersicht

Nach einer Einführung in diesem Kapitel, beschäftigt sich *das zweite Kapitel* mit den Grundlagen des Feldbussystems FlexRay. Dabei werden die für die Erstellung der Arbeit notwendigen Inhalte der FlexRay-Spezifikation vorgestellt.

Das dritte Kapitel gibt die Anforderungen an die Evaluierungsplattform wieder und zeigt den Prozess der Anforderungsanalyse zu Beginn der Arbeit. Außerdem werden die ausgewählten Hardware-Komponenten vorgestellt.

Das vierte Kapitel geht näher auf die Komponenten der Evaluierungsplattform ein, indem die Beschaltung dieser und das Routing der verschiedenen Signalleitungen erläutert werden. Am Ende des Kapitels wird das Layout des Printed Circuit Board (PCB) vorgestellt.

Das fünfte Kapitel behandelt den Aufbau der Software für das Mikrocontroller-Modul anhand von Flussdiagrammen. Weiterhin wird auf die Konfiguration des FlexRay-Treibers eingegangen.

Das sechste Kapitel stellt die Testmessungen der Hard- und Software vor und bewertet diese unter Berücksichtigung der Anforderungen. Zuletzt werden außerdem Optimierungsansätze der aktuellen Revision vorgestellt.

Im letzten Kapitel werden die Ergebnisse der Arbeit zusammengefasst und ein Ausblick für weitere Untersuchungen mit der Evaluierungsplattform gegeben.

2. Grundlagen FlexRay

Dieses Kapitel erläutert die Grundlagen des Bussystems FlexRay. Die Entwicklung der Spezifikation des Bussystems begann im Jahre 2000 mit der Gründung des FlexRay-Konsortiums, welches sich aus den Unternehmen BMW, Daimler AG, Motorola und Philips zusammensetzte. Nach zehn Jahren wurde die Entwicklung abgeschlossen und das Konsortium aufgelöst. Anschließend darauf wurde im Jahr 2013 zudem die Spezifikation in einen kostenpflichtigen ISO-Standard überführt ([Int13a] bis [Int13b]).

Aus der Zielsetzung der Entwicklung leiten sich die grundlegenden Eigenschaften des FlexRay-Systems ab [Rau07, S. 5 ff.]. Das System bietet eine Bitrate von maximal 10 MBit/s . Diese bezieht sich wie bei jedem Protokoll auf die Bruttodatenrate. Die Bitrate, mit der Nutzdaten übertragen werden, ist je nach Konfiguration der Teilnehmer, um einiges geringer. Im Vordergrund eines FlexRay-Systems steht zudem der Anspruch, eine möglichst sichere Übertragung zu gewährleisten. Dazu sind die Kommunikationskanäle redundant ausgelegt. Das bedeutet, dass jeder Kommunikationsteilnehmer zwei physikalische Übertragungsmedien unterstützt. Wird dieser Schutz nicht benötigt, können die Kanäle zur Bandbreitenerhöhung genutzt werden, indem auf den Kanälen verschiedene Daten übertragen werden. Der Buszugriff erfolgt über das TDMA-Verfahren. Jedem Teilnehmer wird hier ein bestimmter Zeitschlitz (engl.: slot) zugewiesen, in dem er exklusiven Zugriff auf den Bus erhält. Da die Abfolge der Zuweisung bekannt ist, kann eine maximale Reaktionszeit des Systems bestimmt werden. Dieses deterministische Bussystem eignet sich daher für Echtzeitanwendungen.

Die folgenden Abschnitte gehen zunächst näher auf den elektrischen Physical Layer der FlexRay-Spezifikation ein und beschreiben anschließend Grundlagen des Kommunikationsprotokolls.

2.1. Aufbau eines FlexRay-Knotens

Die Teilnehmer eines FlexRay-Netzwerks werden als Knoten (engl. node) bezeichnet und bestehen im wesentlichen aus einem Mikrocontroller (auch als Host bezeichnet), dem eigentlichen FlexRay-Controller und einem oder mehreren Bustreibern. Dabei wird bei den FlexRay-Controllern zwischen sogenannten stand-alone Controllern und integrierten Versionen, welche Host und FlexRay-Controller vereinen, unterschieden. Stand-alone FlexRay-Controller beinhalten nur den für die eigentliche FlexRay-Protokollierung zuständigen Schaltkreis und benötigen einen externen Host für den Datenaustausch des Anwendungsprogramms. Integrierte Controller beinhalten den FlexRay-Controller als Peripheriemodul im Gehäuse des Host-Controllers. Im Gegensatz zu den flexibel einsetzbaren stand-alone Controllern bietet die Variante mit integriertem FlexRay-Controller Kosteneinsparung und einen schnelleren Datendurchsatz, da die Speicherbereiche auf einem Halbleiterchip gebündelt werden. Die Bustreiber, auch FlexRay-Transceiver genannt, bringen die Datensignale auf den definierten Buspegel. [Rau07, S. 21] Abbildung 2.1 zeigt den schematischen Aufbau eines FlexRay-Knotens.

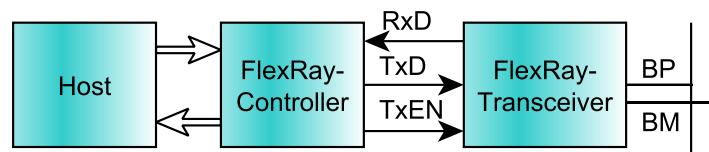


Abbildung 2.1.: Schematischer Aufbau eines FlexRay-Knotens

Die Kommunikation zwischen FlexRay-Controller und FlexRay Transceiver erfolgt über eine Empfangsdatenleitung (RxD engl. Receive Data) und eine Sendedatenleitung (TxD engl. Transmit Data). Über eine weitere Low-aktive Leitung, Transmit Data Enable Not (TxEN) initialisiert der FlexRay-Controller die Übertragung der anliegenden Daten vom FlexRay-Transceiver auf den Bus. [Rau07, S. 22]

2.2. Buspegel

Innerhalb eines FlexRay-Clusters findet die Übertragung von Signalen über den Austausch von Differenzsignalen statt [Vec15a]. Die physikalische Verbindung zwischen zwei Teilnehmern setzt sich daher aus zwei Leitungen zusammen, die als Bus Plus (BP) und Bus Minus (BM) bezeichnet werden. Durch die Auswertung der Spannungsdifferenz zwischen den

beiden Leitungen werden Gleichtaktstörungen, die auf beiden Adern gleichwertig auftreten, herausgefiltert.

Die FlexRay Spezifikation des Physical Layers definiert insgesamt vier zulässige Buszustände. Zum einen sind dies die dominanten Pegel, welche die Daten einer FlexRay-Botschaft übertragen. Wird eine logische Eins übertragen (Buspegel Data_1), steigt der Spannungspegel auf der BP-Leitung, während der Spannungspegel auf der BM-Leitung sinkt. Die Spannungsdifferenz zwischen den Leitungen muss folglich positiv sein. Bei der Übertragung einer logischen Null gilt genau das Gegenteil. Hier sinkt der Spannungspegel auf der BP-Leitung, während der Spannungspegel auf der BM-Leitung steigt. Die Differenzspannung ist negativ.

Werden keine Daten übertragen, stellt sich der rezessive Buszustand Idle ein. Die Differenzspannung der Leitungen beträgt hier 0 V, wobei die Spannungspegel sowohl auf der BP- als auch auf der BM-Leitung 2,5 V betragen. Befinden sich alle FlexRay-Transceiver im Low-Power-Modus wechselt der Buszustand in den Idle Low Power Zustand. Anders als im Idle Zustand beträgt hier sowohl die Differenzspannung, als auch der Spannungspegel auf den Leitungen 0 V. [Rau07, S. 117 f.]

Abbildung 2.2 zeigt die beschriebenen Buspegel und gibt die zulässigen Differenzspannungen der Empfangs- bzw. Sendeseite an.

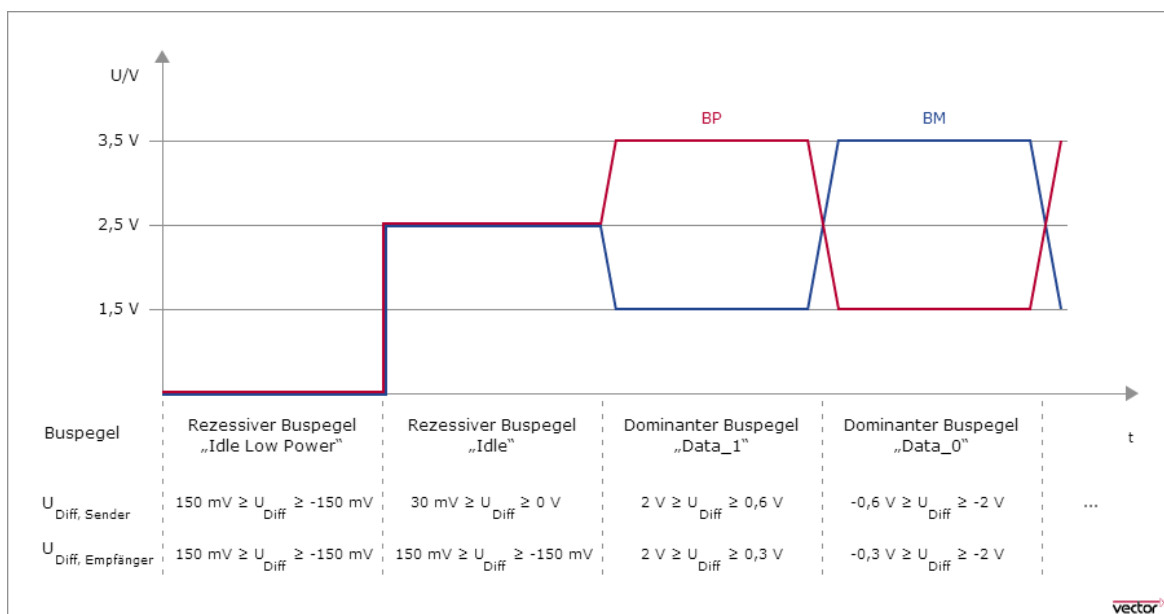


Abbildung 2.2.: Darstellung der FlexRay-Buspegel [Vec15a]

2.3. Topologien

Als Topologie wird prinzipiell die physikalische Anordnung der Teilnehmer eines Bussystems bezeichnet. FlexRay-Systeme können mit vier verschiedenen Topologien bzw. Mischformen dieser realisiert werden. Dabei werden die Topologien in passive und aktive unterteilt. Bei aktiven Topologien wird neben den Transceivern der Teilnehmer ein weiteres Bauteil, der aktive Sternkoppler, eingesetzt. Dieser verstärkt und verteilt die elektrischen Signale der Busverbindung und gewährleistet aufgrund idealer Busabschlüsse eine bessere Signalintegrität. Außerdem können fehlerhafte Kommunikationszweige getrennt werden um den Rest des Systems zu schützen. Passiven Topologien fehlen diese zusätzlichen Sternkoppler und die damit verbundenen Vorteile. Der Vorteil passiver Topologien ist jedoch der einfache und kostensparende Aufbau. [Vec15d], [Rau07]

Abbildung 2.3 zeigt die vier grundlegenden Topologien. Bei der Kommunikation zweier Teilnehmer wird eine Punkt-zu-Punkt-Verbindung aufgebaut. Laut Spezifikation darf die Länge der elektrischen Verbindung zwischen den Teilnehmern 24 m nicht überschreiten [Fle06a, S. 13]. Werden weitere Teilnehmer benötigt, können diese über Stichleitungen (sogenannte Stubs) an den Bus angeschlossen werden. Dies wird dann als Linientopologie bezeichnet. Auch in diesem Fall darf die maximale Kabellänge zwischen den zwei am weitesten voneinander entfernten Teilnehmern 24 m nicht überschreiten. Um eine ausreichend hohe Signalintegrität zu gewährleisten, sollten laut Spezifikation nicht mehr als 22 Stichleitungen eingesetzt werden. Dasselbe gilt für eine passive Sterntopologie. Die Topologie unten rechts in der Abbildung 2.3 zeigt ein System mit aktivem Sternkoppler, der wie bereits erwähnt, die elektrischen Signale verstärkt und verteilt. Die Distanz zwischen dem Sternkoppler und einem FlexRay-Knoten soll 24 m nicht überschreiten. Dadurch steigt die Distanz zwischen zwei Teilnehmer auf 48 m und kann durch das Hintereinanderschalten zweier aktiver Sternkoppler zudem auf insgesamt 72 m ausgedehnt werden.

Die Anordnung der Knoten innerhalb eines FlexRay-Clusters muss nicht zwangsläufig einheitlich sein, sondern kann auch aus einer Mischform der bereits beschriebenen Topologien bestehen. Beispielsweise können nah aneinander liegende Knoten mit einer günstigen, passiven Bustopologie versehen werden, während die Verbindung zu weiter entfernten Knoten über eine aktive Topologie erfolgt, um hier die Übertragungssicherheit zu verbessern. Außerdem ist es nicht notwendig, dass die zwei Kanäle eines FlexRay-Systems die gleiche Topologie besitzen. Dadurch wird eine größtmögliche Freiheit bei der Planung eines FlexRay-Systems geboten. [Rau07]

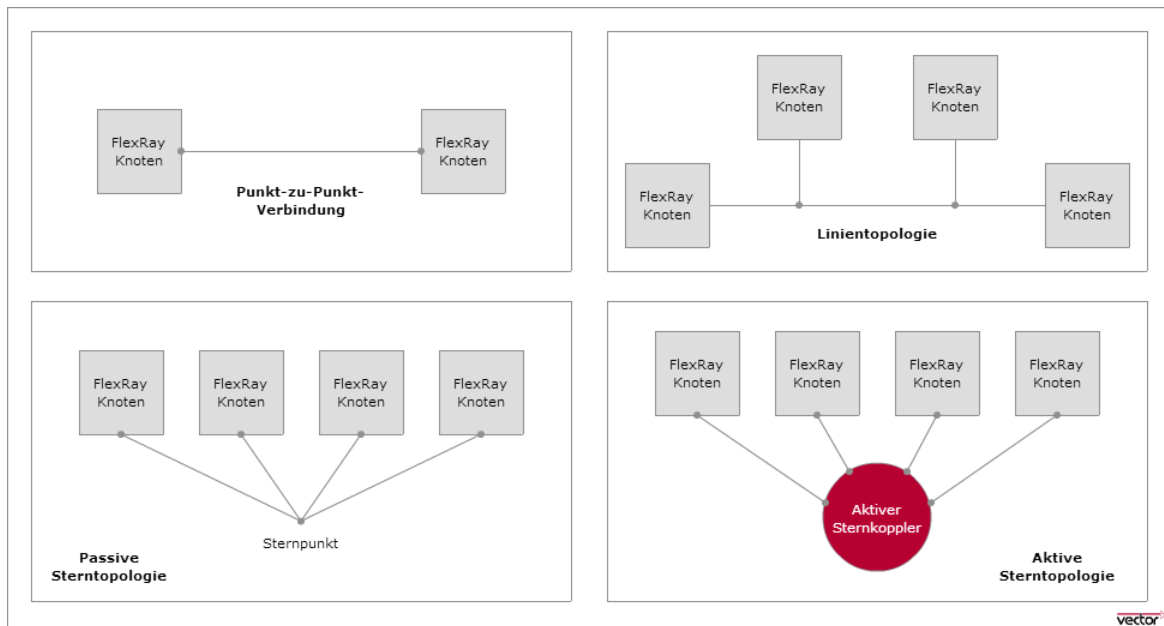


Abbildung 2.3.: Schematischer Aufbau verschiedener Topologien eines FlexRay-Clusters

2.4. Der FlexRay-Kommunikationszyklus

Die Kommunikation im FlexRay-Protokoll ist in sogenannte Kommunikationszyklen unterteilt. Dabei ist der zeitliche Ablauf dieser Zyklen klar definiert und allen Teilnehmern des Clusters bekannt. Die einzelnen Zyklen sind jeweils mit einem Zähler versehen, der bei null beginnt und bis 63 hochgezählt wird. Danach beginnt die Kommunikation wieder mit dem Zykluszähler (cycle count) null. Die vier Segmente eines Kommunikationszyklus, statisches Segment, dynamisches Segment, Symbol Window und Network Idle Time (NIT), werden in Abbildung 2.4 dargestellt.

Das erste Segment eines Kommunikationszyklus wird als statisches Segment bezeichnet. Die Zeitschlitze innerhalb des Segments, sogenannte Static-Slots, haben alle die gleiche vorab definierte Zeitdauer. Jedem Teilnehmer eines Clusters können ein oder mehrere dieser aufsteigend nummerierten Static-Slots zugewiesen werden. Die Nutznachrichten die innerhalb der Slots übertragen werden, haben ebenfalls eine definierte Länge, welche nur bei einem Neustart des Systems neu konfiguriert werden kann. Der Zugriff im statischen Segment erfolgt mittels TDMA-Verfahren.

Das dynamische Segment nutzt für den Buszugriff das sogenannte Minislot-Verfahren. Wie die Static-Slots besitzen auch die Minislots eine fest definierte Zeitdauer. Diese ist jedoch kleiner, als die der Zeitslitze des statischen Segments. Beginnt ein Teilnehmer mit der Übertragung innerhalb eines dynamischen Slots, wird dieser so lange um weitere Minislots erweitert, bis die Nachricht übermittelt wurde. Sollte keine Übertragung von Nutzdaten benötigt werden, beträgt die Größe des jeweiligen dynamischen Slots genau einen Minislot. Ziel dessen ist es, eine dynamische Ausnutzung der zur Verfügung stehenden Bandbreite zu gewährleisten.

Das Symbol Window dient zum Übertragen von Symbolen, welche für die Funktion des sogenannten Busguardians benötigt werden, auf den hier nicht weiter eingegangen werden soll. Der Aufbau des Symbol Windows selbst ist identisch mit dem eines statischen Slots. Dabei findet zwischen den Teilnehmern selbst keine Übertragung von Nutzdaten statt. Im Gegensatz zum statischen Segment und der Network Idle Time ist die Konfiguration des Symbol Windows und des dynamischen Segments optional. Werden diese Segmente nicht genutzt, treten sie in keinem Zyklus auf. Eine Änderung dieser Einstellung ist nur bei Systemreset und anschließender Neukonfigurierung aller Teilnehmer des Clusters möglich.

Das abschließende Segment eines Kommunikationszyklus wird als Network Idle Time (kurz NIT) bezeichnet und dient controllerinternen Prozessen, in denen die Teilnehmer ihre Uhrenkorrektur durchführen, um ein globales Zeitverständnis zu gewährleisten. In diesem Segment findet keine Kommunikation innerhalb des Clusters statt. Die NIT wird also ausschließlich für die Uhrenkorrektur reserviert. [Rau07]

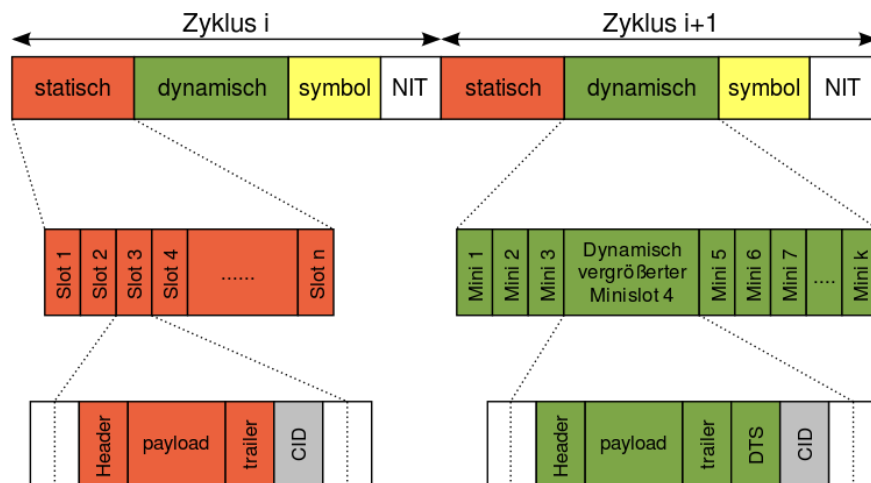


Abbildung 2.4.: Schematische Darstellung des FlexRay-Kommunikationszyklus nach TDMA-Verfahren [Wik15b]

2.5. Framing einer FlexRay-Botschaft

Für die Übertragung von Daten in den Zeit-Slots eines Zyklus übermitteln die Teilnehmer Botschaften, sogenannte Frames. Diese setzen sich aus den drei Teilen, dem Header, den eigentlichen Nutzdaten (engl. Payload) und dem Trailer zusammen (siehe Abbildung 2.5).

Im ersten Teil, dem 40 Bit breiten Header, werden zunächst Informationsdaten des FlexRay-Protokolls übertragen. Bei dem ersten Bit handelt es sich um einen für Protokollerweiterungen reservierten Indikator, welcher nicht vom Anwender genutzt werden kann. Der folgende Payload Preamble Indicator gibt an, ob das Payload-Segment reguläre Nutzdaten oder weitere Steuerinformationen beinhaltet. Enthält das Payload-Segment weder Steuerinformationen noch reguläre Nutzdaten, wird dies über den Null-Frame Indikator indiziert. Für die Uhrensynchronisation der FlexRay-Controller kann der Sync-Frame Indikator gesetzt werden. Dieser kennzeichnet dann den Frame als sogenannten Sync-Frame. Das letzte Indikator-Bit (Startup Frame Indicator) dient der Kennzeichnung eines Startup-Frames. Dieser initialisiert den Startvorgang des FlexRay-Cluster und darf nur von Teilnehmern gesendet werden, die speziell dafür vorgesehen sind (sogenannte Coldstart-Knoten).

Nach den ersten fünf Indikator-Bits folgt ein 11 Bit breites Feld für die ID (Identifikationsnummer) des Frames. Innerhalb eines Kommunikationszyklus ist diese eindeutig und kann Werte von 1 bis 2047 annehmen. Die Länge der Nutzdaten wird über das Payload Length Feld mitgeteilt. Dabei kann die Länge der Payload zwischen 0 und 254 Byte liegen. Das nächste 11 Bit breite Feld überträgt die Prüfsumme für die zyklische Redundanzprüfung (Cyclic Redundancy Check (CRC)), womit die bisherigen Daten auf eine fehlerfreie Übertragung geprüft werden. Der Header schließt mit dem 6 Bit breiten Cycle Count Feld, welches den aktuellen Zyklus kennzeichnet. Dieser wird von 0 bis 63 hochgezählt und startet dann von Neuem.

Dem Header folgen nun die eigentlichen Nutzdaten im Payload-Segment. Übertragen werden die Daten zu jeweils 16 Bit breiten Wörtern. Die Länge der Nutzdaten kann zwischen 0 und 127 Wörtern betragen, demzufolge also 0 bis 254 Bytes.

Abgeschlossen wird jeder Frame mit dem sogenannten Trailer, einem weiteren CRC, welcher sich aus dem gesamten Frame berechnet. Mit diesem 24 Bit breiten Segment können die empfangenen Framedaten auf Übertragungsfehler geprüft werden. [Vec15c], [Rau07]

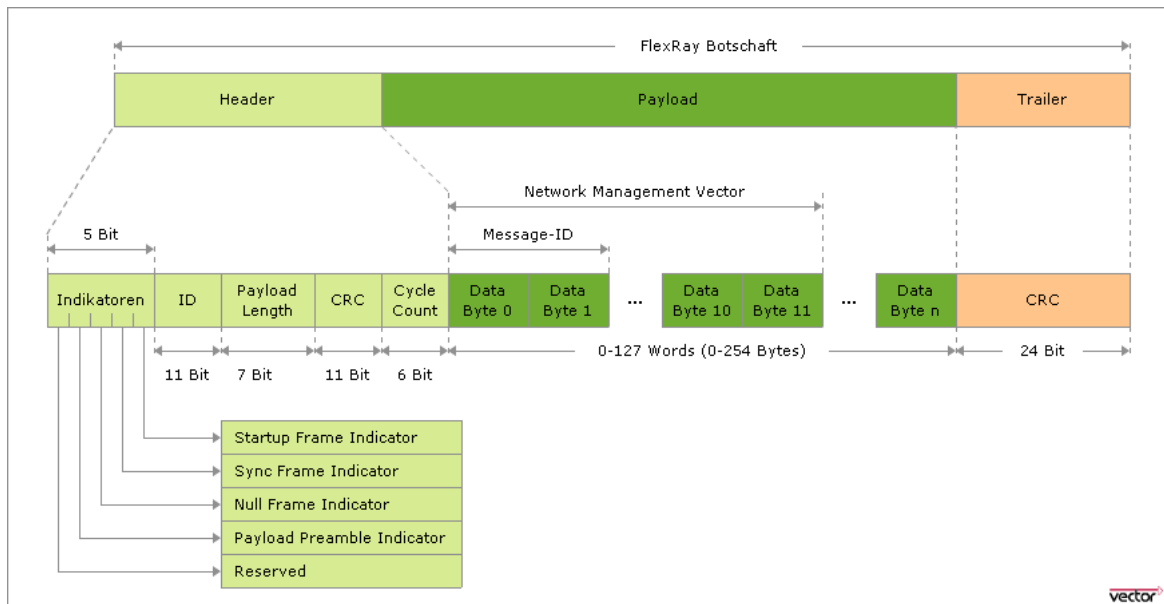


Abbildung 2.5.: Übersicht des Aufbaus einer FlexRay-Botschaft [Vec15c]

2.6. FlexRay Zeithierarchie

Bei FlexRay handelt es sich, wie bereits erwähnt um ein deterministisches Bussystem, d.h. jeder Teilnehmer bekommt einen definierten Zeitschlitz zugeteilt, in dem dieser exklusiv seine Botschaft senden darf. Kollisionen sind somit ausgeschlossen, sofern die korrekte Zeiteinteilung von allen Teilnehmern eingehalten wird. Dazu ist es notwendig, dass alle Kommunikationsteilnehmer das gleiche Zeitverständnis haben. Die FlexRay-Spezifikation sieht hierzu eine Zeithierarchie vor, die von jedem Teilnehmer eingehalten werden muss.

Die kleinste Zeiteinheit eines FlexRay-Controllers ist der Microtick. Prinzipiell kann dieser frei gewählt werden, wird in der Praxis jedoch von der Taktfrequenz des lokalen Oszillators und des gewählten Vorteilers des FlexRay-Controllers bestimmt. Abbildung 2.6 zeigt die Abhängigkeiten zwischen Bitrate (hier auch als Busgeschwindigkeit bezeichnet) und Microtick-Länge. Da jedes Bit achtmal abgetastet wird, ergibt sich die Zeitdauer eines Sampleticks zu einem Achtel der Bitzeit. Aus der Sampletick-Frequenz kann dann über ein Teilverhältnis von 1, 2 oder 4 der gewünschte Microtick erzeugt werden.

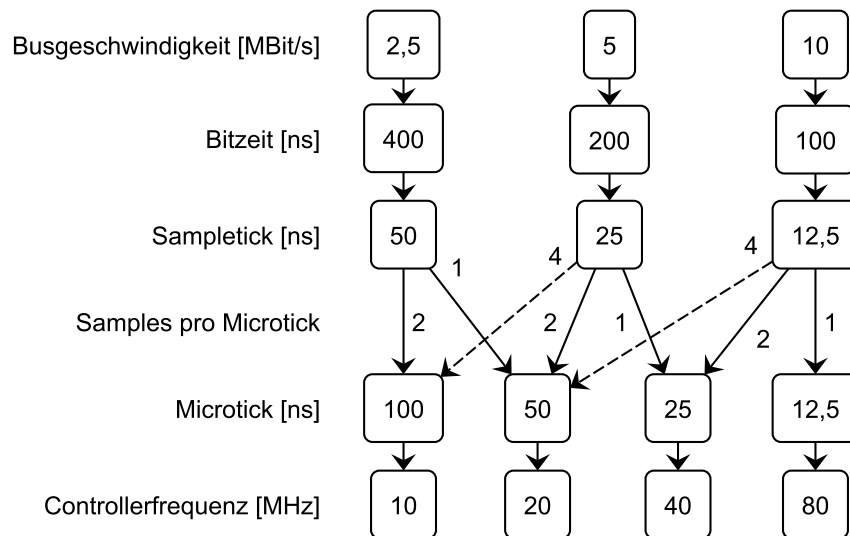


Abbildung 2.6.: „Abhängigkeiten zwischen Bitrate und Microtick-Länge“ [Rau07, Bild 6.4]

Die nächstgrößere Zeiteinheit wird als Macrotick bezeichnet und setzt sich aus mehreren Microticks zusammen. Im Gegensatz zu den Microticks, die sich durch die Oszillatorungenauigkeit des jeweiligen FlexRay-Controllers unterscheiden, ist die Länge eines Macroticks auf allen Kommunikationsteilnehmern gleich definiert. Die Anzahl der Microticks pro Macrotick kann dabei variiert werden, um für jeden Teilnehmer die gleiche Macrotick-Länge einzustellen. Bei der Planung des Systems kann frei zwischen einer Macrotick-Länge zwischen $1\ \mu\text{s}$ und $6\ \mu\text{s}$ gewählt werden. Die minimale Anzahl an Microticks, die einen Macrotick bilden, ist dabei auf 40 festgelegt, damit die Länge eines Macroticks nicht zu klein ausfällt. [Rau07]

Die Abbildung 2.7 zeigt die Zeithierarchie des FlexRay-Protokolls. Die Zeiteinheiten werden hier innerhalb von vier Ebenen dargestellt, die von oben nach unten in ihrer zeitlichen Auflösung steigen. Neben den feingranularen Ebenen für Microtick und Macrotick Level existiert eine mittelgranulare Ebene, auf der die Zeiteinheit für den Buszugriff der Teilnehmer definiert wird. Auf diesem sogenannten Arbitration Grid Level wird das Verständnis für die Zeiteinheit im statischen Segment (Static-Slot) und im dynamischen Segment (Minislot) festgelegt. Sowohl Static-Slot, als auch Minislot setzen sich aus mehreren, fest definierten Macroticks zusammen. Die höchste Ebene der FlexRay-Zeithierarchie beinhaltet als Einheit die Zeitdauer eines Kommunikationszyklus. Wie bereits die Länge der Slots wird die Länge eines Kommunikationszyklus durch eine, bei allen Kommunikationsteilnehmern gleich definierte Anzahl von Macroticks eingestellt.

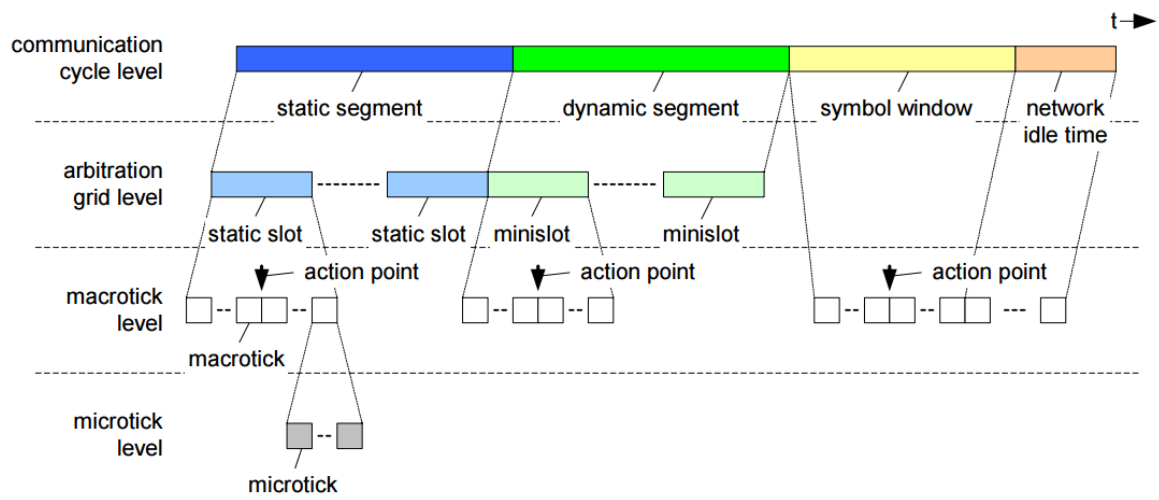


Abbildung 2.7.: Schematische Darstellung der FlexRay Zeithierarchie [Tem]

3. Anforderungen an die Evaluierungsplattform

Bei der Entwicklung eines Systems gilt grundsätzlich folgende Annahme: „Je früher ein Fehler gemacht und je später er dann entdeckt wird, desto höher werden die Kosten für die Korrektur“ [Kle13, S. 55]. Um dieses Risiko zu minimieren wird eine Anforderungsanalyse durchgeführt. Insbesondere aus der Softwareentwicklung bekannt, wird diese aber auch in der Entwicklung von Hardware-Systemen angewendet. Wie in der Softwareentwicklung, dient die Anforderungsanalyse dazu, die Anforderungen an das System zu ermitteln, zu strukturieren und zu bewerten. Durch diese Analyse werden alle identifizierbaren Anforderungen dokumentiert und die Abhängigkeiten der Anforderungen berücksichtigt. Außerdem erfolgt eine Priorisierung der Anforderungen. Dieses Kapitel beschreibt die der Arbeit zugrunde liegende Anforderungsanalyse und benennt anschließend die ausgewählten Hardware-Komponenten.

3.1. Anforderungsanalyse

Die Anforderung an das System ergeben sich aus den geplanten Aktivitäten des X-by-Wire(less)¹-Projekts im Rahmen des Urban Mobility Lab der HAW Hamburg. Das Ziel des Projektes ist es, die Machbarkeit von drahtloser Kommunikation im Automobil zu evaluieren und eine technische Lösung dafür zu schaffen. Als Startpunkt soll der Physical Layer eines FlexRay-Clusters auf ein Funksystem umgestellt werden. Für die zu erstellende Evaluierungsplattform liegt damit die Hauptanforderung im Bereitstellen einer FlexRay-Kommunikation mittels FlexRay-Controller. Dabei wird zwischen sogenannten stand-alone Controllern und Mikrocontrollern mit integriertem FlexRay Controller unterschieden (vgl. Abschnitt 2.1). Stand-alone Controller müssen mit einem Host-Controller verbunden werden und stellen selbst nur die FlexRay-Funktionen zur Verfügung. Sie sind im Umgang flexibel, da auf bereits bekannte und erprobte Mikrocontroller als Hosts zurückgegriffen werden kann.

¹X-by-Wire bezeichnet die Umstellung von mechanischen auf elektronische Steuersysteme [Wik15g]. X-by-Wireless kennzeichnet den Ersatz von drahtgebundener durch drahtlose Signalübertragung.

Bereits 2007 zeigt sich jedoch der Trend, die Produktion auf integrierte Bausteine zu fokussieren. Diese sind günstiger in der Produktion und kleiner als vergleichbare Stand-Alone Controller. Zudem wird die Funktionalität eines FlexRay-Teilnehmers an einem Ort gebündelt und dadurch besser planbar [Ste07]. Auszuwählen ist somit ein Mikrocontroller mit integriertem FlexRay-Controller, der dem aktuellsten Standard entspricht, vor allem also die maximale Bitrate von 10 Mbit/s bereitstellen kann.

Neben der eigentlichen FlexRay-Datenkommunikation sollen grundsätzlich auch weitere Feldbusse in die Evaluierung einbezogen werden können. Die Wahl fällt hier auf die weit verbreiteten Bussysteme CAN und LIN. Ein relativ neuer Ansatz in Sachen Bussysteme im Automobilbereich ist die Verwendung von Real-Time-Ethernet [Wil14]. Im Vergleich zu FlexRay wird hier eine zehnmal höhere Übertragungsrate geboten. Allerdings setzt FlexRay nach wie vor den Fokus auf Sicherheit und erlaubt verteilte Multimaster-Kommunikation, während Ethernet Punkt-zu-Punkt-Kommunikation und einen zentralen Switch erfordert, mit dessen Ausfall auch die gesamte Kommunikation zu den Teilnehmern ausfällt. Um diese Bussysteme vergleichen zu können, soll die Evaluierungsplattform auch eine frei verwendbare Ethernet-Schnittstelle bieten. Durch die Anforderung mehrere Bussysteme zur Verfügung zu stellen, leiten sich weitere Anforderungen ab, nämlich die Bereitstellung von entsprechenden Pegelwandlern, bzw. Transceivern für den Physical Layer. Zu beachten ist, dass die Bereitstellung des FlexRay Physical Layers höchste Priorität hat. Die weiteren Bussysteme dienen nur dem Vergleich einer möglichen Drahtlos-Kommunikation und sind daher niederwertiger zu priorisieren.

Die Konfiguration des Bussystems soll zur Laufzeit möglich und über einen Rechner fernwartbar sein. Dafür soll eine weitere Ethernet-Verbindung verwendet werden, um eine ausreichend hohe Übertragungsrate (100 Mbit/s) zu gewährleisten und das FlexRay-Cluster in ein bestehendes Rechnernetz einbinden zu können. Die in vielen Mikrocontrollern ohnehin integrierte Universal Asynchronous Receiver Transmitter (UART)-Schnittstelle bietet hingegen relativ langsame Übertragungsraten und wird im Rahmen dieser Arbeit nur als Debugging-Interface verwendet (siehe Kapitel 5). Eine drahtlose Kommunikation über WLAN ist nicht vorgesehen um mögliche Interferenzen zwischen Konfigurationsfunk und der zukünftigen drahtlosen Kommunikation des FlexRay Physical Layers zu vermeiden.

Für den Test des Gesamtsystems ist ein Aufbau mit mehreren Teilnehmern nötig. Um den Aufwand und die Kosten dieses Aufbaus möglichst gering zu halten, wird die Entwicklung einer einseitig bestückten Platine mit zwei Layern angestrebt. Dabei soll die Platine zudem möglichst kompakt bleiben um später gegebenenfalls eine Platzierung der Evaluierungsplattform im Automobil zu ermöglichen, bevor eine Neuentwicklung in den üblichen Abmessungen eines Kfz-Steuergeräts erfolgt.

Die Entwicklung des Drahtlos-Moduls selbst ist nicht Teil der Arbeit, jedoch müssen die Anforderungen an ein solches Modul bereits berücksichtigt werden. Hierzu gehört eine aus-

reichende Fläche auf der Evaluierungsplattform, eine geeignete Spannungsversorgung, sowie die Zuführung geeigneter und in der Zahl ausreichender Signalleitungen zu den anderen Komponenten der Platine. Da die Verfahren der digitalen Übertragung Gegenstand der späteren Entwicklung und Evaluation der drahtlosen Kommunikation sein sollen, wird ein programmierbarer Digitalbaustein benötigt, der den damit verbundenen hohen Geschwindigkeitsanforderungen gerecht wird. Für diese rechenintensive Aufgabe soll durch die Evaluierungsplattform ein FPGA bereit gestellt werden, welcher ausreichend leistungsstark sein muss, um die gewünschte Übertragungsgeschwindigkeit einhalten zu können.

An dieser Stelle sind die Hauptanforderungen noch einmal zusammengefasst:

- Bereitstellung der Bussysteme FlexRay (höchste Priorität), CAN und LIN
- Drahtgebundene FlexRay-Kommunikation mit zwei Kanälen bei einer Bitdatenrate von 10 Mbit/s pro Kanal
- Bereitstellung von zwei Ethernet-Schnittstellen für die Verbindung mit dem PC zur Konfiguration und für Automotive Ethernet
- Möglichst einfaches Platinendesign (einseitig bestückt, zwei Layer)
- Leistungsstarkes FPGA für den Modulationsalgorithmus
- Steckplatz für das zukünftige Drahtlos-Modul

3.2. Entscheidung für Mikrocontroller- und FPGA-Modul

Nachdem die Hauptanforderungen aufgelistet sind, erläutert dieser Abschnitt die Auswahl der Hardwarekomponenten. Im ersten Teil werden dazu die Entscheidungskriterien für ein geeignetes Mikrocontroller-Modul erläutert und die ausgewählte Hardware beschrieben. Im zweiten Teil werden die ausgewählten, zusätzlich benötigten FlexRay-Transceiver vorgestellt. Zuletzt wird die Auswahl des leistungsstarken FPGA-Moduls erörtert.

Entscheidung für das Mikrocontroller-Modul

Bei der Entscheidung für einen geeigneten FlexRay-Controller wurde im Vorfeld zunächst ein stand-alone Controller präferiert, da dieser flexibel mit einem bereits bekannten Mikrocontroller verwendet werden kann. Im speziellen wurden der MB88121B von Fujitsu [Fuj06] und der MFR4310 von Freescale Semiconductors [Fre15a] erwogen. Wie bereits in Abschnitt 3.1 erwähnt, sind die Hersteller von FlexRay-Controller jedoch bereits 2007 hauptsächlich auf die Produktion von integrierten FlexRay-Controllern umgestiegen. Viele Bereiche von Fujitsu, unter anderem auch die Produktparte der FlexRay-Controller, sind mittlerweile an die Firma Spansion [Wik15e] übergeben worden. Weder der Controller selbst noch ursprünglich angebotene Entwicklungssysteme sind auf gängigen Verkaufsplattformen erhältlich. Der MFR4310 von Freescale wird zwar unter [Fre15b] als aktiv gelistet, für den Standort Deutschland liefert die Suchfunktion jedoch keinen Distributor. Die Marktsituation zeigt also, dass ein integrierter FlexRay-Controller ausgewählt werden sollte.

Bei der Recherche nach einem integrierten FlexRay-Controller bzw. eines Entwicklungssystems für diesen kommen zwei Firmen in die engere Auswahl. Zum einen ist dies Freescale Semiconductors, Marktführer in der Sparte FlexRay-Controller, zum anderen Texas Instruments. Das preiswerteste Entwicklungssysteme von Freescale ist das S12XFSTARTERKITE Starter Kit [Fre15c]. Dieses bietet zwei Platinen für den Aufbau eines FlexRay-Clusters mit zwei Teilnehmer und kostet \$450. Gegen die Verwendung spricht jedoch die unbekannte Entwicklungsumgebung, welche die Entwicklungsarbeit mit einer Begrenzung von Codezeilen einschränkt, sofern keine teure Lizenz erworben wird. Außerdem wird die Anforderung eine Ethernet-Schnittstelle bereit zu stellen nicht erfüllt. Die Entwicklungsplatinen sind bereits mit FlexRay Transceivern ausgestattet, sodass die FlexRay-Signale nur umständlich an das FPGA-Modul geführt werden können. Dies gefährdet die Signalintegrität der Datenkommunikation. Weitere Entwicklungsboards von Freescale sind in erster Linie zu teuer und weisen die gleichen, oben genannten Problem auf. Die Wahl fällt also auf Texas Instruments. Im Gegensatz zu Freescale ist die Entwicklungsumgebung Code Composer Studio von Texas Instruments (TI) an der Hochschule bereits bekannt und erprobt. Des Weiteren existiert ein eigenes Wiki für die Automotive-Security Produktserie Hercules [Tex15b] und bietet mit Beispiel-Codes eine gute Unterstützung bei der Entwicklungsarbeit. In der Produktserie der Hercules-Mikrocontroller gibt es mehrere Entwicklungsboards, in Frage kommen jedoch nur

die mit FlexRay-Funktionalität. Dabei handelt es sich um die sogenannten Hercules Development Kit (HDK) Entwicklungsboards TMS570LS31x [Tex15g] und TMS570LS12x [Tex15e]. Tabelle 3.1 zeigt eine Auswahl der wichtigsten Kenndaten der zwei Mikrocontroller der Endauswahl. Ausgewählt wird das HDK Entwicklungsboard, welches mit dem TMS570LS3137-ZWT bestückt ist. Diese Variante bietet bei gleichen Boardkosten (199\$) den etwas umfangreicheren Mikrocontroller.

Kenndaten	TMS570LS3137-ZWT	TMS570LS1227-ZWT
Taktfrequenz	180 MHz	180 MHz
Flash-Speicher	3072 kB	1280 kB
RAM	256 kB	192 kB
GPIO	120	101
FlexRay	2 Channel	2 Channel
Ethernet (EMAC)	10/100 Mbit/s	10/100 Mbit/s
SCI/LIN	2/1	2/1
CAN	3 Channel	3 Channel
ADC	2 x 12-Bit (24 Channel)	2 x 12-Bit (24 Channel)

Tabelle 3.1.: Vergleich der wichtigsten Funktionen der beiden Mikrocontroller der Endauswahl [Tex15f], [Tex15d]

Das ausgewählte TI HDK TMS57031x bietet alle unter Abschnitt 3.1 geforderten Funktionen, bis auf die Transceiver für FlexRay und LIN. Die nötigen Transceiver für FlexRay werden durch die Evaluierungsplattform zur Verfügung gestellt, da die Plattform die Voraussetzungen für die Entwicklung eines drahtgebunden und drahtlosen FlexRay Physical Layers bieten soll. Gemäß den Anforderungen dienen die anderen Bussysteme nur dem Vergleich gegenüber FlexRay und sind in der Priorisierung niederwertiger. Daher werden die Signalleitungen des Bussystems LIN lediglich mit dem FPGA-Modul verbunden, um den zukünftigen Wireless Physical Layer auch mit diesem Bussystem zu realisieren. Sofern ein drahtgebundener Physical Layer benötigt wird, können die Signale zudem vom FPGA-Modul unverändert auf das Drahtlos-Modul durchgeschleift werden, um auf eben diesem Modul eine LIN-Transceiverschaltung zu entwickeln. In der Tabelle 3.2 sind die wichtigsten Features des HDK aufgelistet.

-
- Integrierter USB JTAG Emulator (XDS100v2) für erweiterte Debugging-Funktion
 - Externe JTAG-Schnittstelle
 - 10/100 Mbit/s Ethernet-Schnittstelle
 - Jeweils einen Licht- und Temperatursensor
 - Zwei CAN Transceiver
 - UART über einen virtuellen USB COM-Port
 - Ein 8 MB SDRAM
 - Acht frei verwendbare LEDs
 - Ein frei verwendbarer Taster (Ansteuerung über GPIO-Pin)
 - Ein SD-Card Slot
 - Eine 5 V bzw. 3,3 V Referenzspannungsquelle für den ADC
-

Tabelle 3.2.: Features des Texas Instruments TMS570LS31x Hercules Development Kit [Tex13c]

Entscheidung für die zusätzlich benötigten FlexRay-Transceiver

Für die Anbindung eines Teilnehmers werden je zwei Node-Transceiver benötigt, da zwei FlexRay-Kanäle verbunden werden müssen. Als FlexRay-Transceiver werden die TJA1082 FlexRay Node Transceiver [NXP15] von NXP Semiconductors ausgewählt. Die TJA1082 sind NXPs neueste Revision und bieten die Unterstützung der FlexRay-Spezifikation Rev 2.1. Nach dieser Spezifikation werden Bitraten von 2,5 Mbit/s, 5 Mbit/s und den geforderten maximalen 10 Mbit/s gewährleistet. Neben der eigentlichen Hauptfunktion, die Daten des Host-Controllers auf den geforderten Buspegel zu bringen, wird eine Kommunikation zum Host via Serial Peripheral Interface (SPI) ermöglicht. Dadurch können Statusbits ausgewertet werden, welche Informationen über Betriebsmodus und mögliche Fehler geben.

Entscheidung für das FPGA-Modul

Nach internen Gesprächen an der HAW Hamburg wird für das benötigte FPGA-Modul das von Avnet angebotene ZedBoard erwogen [Avn15e]. Dieses ist bereits an der HAW Hamburg vorhanden und wurde in mehreren Arbeiten und Projekten erfolgreich eingesetzt. Der Umgang mit der Entwicklungsumgebung (Vivado) ist ebenfalls bekannt und somit kann bei möglichen Fragen zu der Toolchain die in der Hochschule vorhandene Expertise zu Rate gezogen werden. Gegen die Verwendung des ZedBoards spricht jedoch die Größe und der hohe Preis (\$395 vgl. [Avn15a]), da für jeden Teilnehmer des Testsystems ein FPGA-Modul benötigt wird.

Avnet bietet neben dem großen und teuren ZedBoard mehrere kompaktere und günstigere Versionen an, von denen sich das sogenannte MicroZed [Avn15d] am besten für die Erfüllung der hier gestellten Anforderungen eignet. Es ist flexibel einsetzbar, da die I/O-Pins über Steckerleisten geroutet werden können. Zudem wird die benötigte zweite Ethernet-Schnittstelle für ein Ethernet-Bussystem bereit gestellt. Wie das ZedBoard ist auch das Mi-

croZed mit einem leistungsfähigen Zynq-7010 bzw. Zynq-7020 ausgestattet, welcher neben der konventionellen FPGA-Architektur zusätzlich einen Mehrkernprozessor für System On Chip (SOC)-Anwendungen bereitstellt.

Tabelle 3.3 zeigt die beiden MicroZed-Platinen der Endauswahl. Da die eigentliche Anforderung an das FPGA zum Zeitpunkt der Arbeit schwierig abzuschätzen ist, wird die Version mit dem Zynq-7020 gewählt. Diese ist bei geringem Aufpreis deutlich leistungsstärker und bietet mehr Spielraum, um das Risiko von zu schwacher Hardware zu minimieren.[Avn15d]

Kenndaten	MicroZed SOM 7Z010	MicroZed SOM 7Z020
Verwendeter Chip	Z-7010	Z-7020
Maximale Taktfrequenz	866 MHz	866 MHz
Programmierbare Logik	28000 Zellen	85000 Zellen
Look-Up Tables	17600	53200
Programmierbare DSP Slices	80	220
Preis	\$192	\$289

Tabelle 3.3.: Vergleich der beiden MicroZed-Platinen der Endauswahl [Xil14], [Avn15a]

Eine Eigenentwicklung eines Mikrocontrollerboards ist im zugrunde liegenden Fall aufwendig und risikoreich. Sowohl bei dem ausgewählten Mikrocontroller-Gehäuse als auch bei dem ausgewählten FPGA-Gehäuse handelt es sich um ein schwierig zu verarbeitendes Ball Grid Array (BGA)-Gehäuse [Wik15a] mit mehreren hundert Kontakten [Tex13d] [Xil15b]. Dadurch wird das Verlegen von Signalleitungen ohne Mehrlayer-Platine unmöglich. Die ausgewählte Hardware (TI HDK) kommt als beidseitig bestückte Platine mit acht Layern. Die Herstellung von Mehrlayer-Platinen in Kleinserien ist wiederum sehr teuer. Daher wird eine Evaluierungsplattform realisiert, welche die benötigten Hardware-Komponenten, also Mikrocontroller- und FPGA-Modul, untereinander verbindet und die anfallende Peripherie bereit stellt.

4. Hardware

Nachdem bereits die Auswahl von Mikrocontroller- und FPGA-Modul in Abschnitt 3.2 erläutert wird, geht das Kapitel Hardware näher auf diese Module und die weiteren Hardware-Komponenten ein. Dazu wird zunächst die Stromaufnahme der Komponenten abgeschätzt und die für die realisierte Spannungsversorgung zuständige Hardware beschrieben. Nachfolgend werden die Beschaltung der FlexRay-Transceiver sowie die Verwendung des Mikrocontroller-Moduls erläutert. Anschließend erfolgt noch eine Erklärung der Verbindungen bzw. Signalleitungen der Evaluierungsplattform. Zuletzt wird das Layout der Platine vorgestellt.

4.1. Abschätzung der Stromaufnahme

Da die Evaluierungsplattform möglichst wenig externe Kabelverbindung aufweisen soll, wird eine zentrale Spannungsversorgung vorgesehen. Diese versorgt die ausgewählten Module (Mikrocontroller-, FPGA- und Drahtlos-Modul) und wird entsprechend der abgeschätzten Stromaufnahme dieser Module dimensioniert. Eine exakte Vorhersage der Stromaufnahme ist nicht möglich, da je nach Konfiguration der Module eine höhere bzw. niedrigere Aufnahme zu erwarten ist. Insbesondere die Stromaufnahme des FPGA-Moduls kann stark variieren. Der Strombedarf wird daher anhand von worst-case Werten aus den Datenblättern der Komponenten abgeschätzt.

Zunächst wird der Verbrauch des Mikrocontroller-Moduls untersucht. Das Benutzerhandbuch des HDKs gibt als typische Stromaufnahme 130 mA bei 12 V an [Tex13c, S. 7]. Dabei bezieht sich diese Angabe auf den Normalbetrieb (fully active) bei einer Taktfrequenz von 180 MHz. Die realisierte Software läuft bei einer Taktfrequenz von 160 MHz (vgl. Kapitel 5) ebenfalls im Normalbetrieb, also ohne in einen Stromsparmodus zu wechseln. Die Angabe kann also als Referenz verwendet werden.

Das Datenblatt der FlexRay-Transceiver listet für die Stromaufnahme nur Maximalwerte auf. Die Hauptaufnahme mit 35 mA erfolgt über $V_{CC} = 5\text{ V}$ (Normal Mode bei $R_{BUS} \geq 45\ \Omega$). Zusätzlich wird über $V_{IO} = 3,3\text{ V}$ noch maximal 1 mA Betriebsstrom aufgenommen [NXP12]. Für die redundante FlexRay-Kommunikation werden pro Evaluierungsplattform zwei Transceiver

benötigt. Für die Stromaufnahme der Transceiver werden also insgesamt 70 mA bei 5 V und 2 mA bei 3,3 V angenommen.

Wie bereits erwähnt ist die Stromaufnahme des FPGAs schwer abzuschätzen. Ist die benötigte Konfiguration des FPGAs bekannt, kann ein Software-Tool von Xilinx [Xil15a] verwendet werden um die Stromaufnahme zu bestimmen. Die Implementierung eines Modulationsalgorithmus für den Wireless Physical Layer ist jedoch nicht Teil dieser Arbeit und daher liegt kein konkretes Programmbeispiel vor. Die Stromaufnahme wird deshalb aus Angaben des MicroZed Benutzerhandbuchs abgeschätzt.

Betrieibt man das MicroZed als stand-alone Modul wird zunächst eine 5 V Spannungsversorgung benötigt, welche über die Micro-USB-Buchse oder die Steckerleisten auf der Unterseite des MicroZed zugeführt werden kann. Ausgehend von dieser Spannungsversorgung generiert das MicroZed nun weitere interne Spannungen. Tabelle 18 des Handbuchs bestimmt die benötigte Leistung für das Zynq7010 FPGA zu 4,8 W [Avn15c, S. 30]. Im Rahmen dieser Arbeit wird jedoch die MicroZed Variante mit dem leistungsstärkeren Zynq7020 FPGA verwendet (vgl. Tabelle 3.3). Um die Stromaufnahme der ausgewählten Variante zu bestimmen, muss berücksichtigt werden, dass der Zynq7020 im internen 1 V Netz etwa 0,9 A und im internen 1,8 V Netz 100 mA mehr aufnimmt als der Zynq7010 [Avn15c, S. 28, Tab. 17]. Damit ergibt sich eine Leistungsaufnahme von 5,8 W. Bei einer Versorgungsspannung von 5 V ermittelt sich die Stromaufnahme für das MicroZed dann zu 1,18 A.

Im beschriebenen stand-alone Betrieb stehen dem Anwender jedoch nur acht frei konfigurierbare I/O-Pins auf dem MicroZed zur Verfügung. Damit zukünftig alle weiteren I/O-Pins verwendet werden können, muss eine zusätzliche Spannungsversorgung bereitgestellt werden. Für diese benötigte Spannungsversorgung, wird auf die Schaltung der Breakout Carrier Card [Avn15b] von Avnet zurückgegriffen. Diese bietet einen in der Ausgangsspannung einstellbaren Schaltregler mit einem maximalen Ausgangsstrom von 1 A. Die nähere Erläuterung dieser Spannungsversorgung erfolgt in Unterabschnitt 4.2.3. Da es sich um eine Abwärtswandlung handelt, wird angenommen, dass die Stromaufnahme des einstellbaren Reglers diesen Wert von 1 A nicht überschreitet. Somit kann in erster Näherung die gesamte Stromaufnahme des MicroZeds im 5 V-Netz 2,18 A abgeschätzt werden. Dieser Wert ergibt sich aus der Summe der zuvor für den stand-alone Betrieb angenommenen Stromaufnahme von 1,18 A und der hier abgeschätzten Stromaufnahme von 1 A.

Um eine Stromaufnahme des Drahtlos-Moduls abzuschätzen, wird zunächst angenommen, dass die Übertragung im 13-Zentimeter-Band (2,3 GHz - 2,45 GHz) stattfindet. Laut Verordnung der Bundesnetzagentur darf die maximale äquivalente isotrope Strahlungsleistung (EIRP) ohne weitere Berechtigungsnachweise 10 mW nicht überschreiten [Bun14, S. 4]. Als Referenz für das noch unbekannte Drahtlos-Modul wird das weit verbreitete WLAN-Modul Xbee von Digi [Dig13] verwendet, welches ebenfalls im 13-Zentimeter-Band arbeitet. Dieses besitzt eine maximale Stromaufnahme von 309 mA bei einer Versorgungsspannung von

3,3 V. Die maximale Sendeleistung beträgt in diesem Fall allerdings +16 dBm, also etwa 39,8 mW und müsste daher bereits gedrosselt werden. Es ist also zu erwarten, dass die Stromaufnahme für 10 mW Sender unter dem angegebenen Wert liegt. Um der Entwicklung des Drahtlos-Moduls Freiraum zu geben, wird für die Stromaufnahme ein höherer Wert, nämlich 500 mA bei 3,3 V angenommen.

Die Tabelle 4.1 fasst die ermittelten Kenndaten in einer Übersicht zusammen.

Komponente	Stromaufnahme	Spannungsnetz
Mikrocontroller-Modul	130 mA	12 V
FPGA-Modul	2,18 A	5 V
FlexRay-Transceiver	70 mA / 2 mA	5 V / 3,3 V
Drahtlos-Modul	500 mA	3,3 V

Tabelle 4.1.: Stromaufnahme der Komponenten

4.2. Spannungsversorgung

Aus der abgeschätzten Stromaufnahme für die jeweiligen Komponenten leiten sich die Anforderungen für die Spannungsversorgung ab. Wie bereits aus Tabelle 4.1 ersichtlich ist, muss die Evaluierungsplattform im Wesentlichen drei verschiedene Spannungen bereitstellen. Hinzu kommt eine zusätzliche, einstellbare Spannungsversorgung für die I/O-Pins des FPGA-Moduls. Abbildung 4.1 zeigt den Aufbau der Evaluationsplattform schematisch als Blockdiagramm.

Als externe Spannungsquelle (gelber Block) wird ein Steckernetzteil mit maximal 1A Ausgangsstrom verwendet. Dieses liegt dem Lieferumfang des TI HDK bei und genügt für die Untersuchungen im Rahmen dieser Arbeit (vgl. Unterabschnitt 6.1.1). Für einen möglichen Einsatz im Automobil sind 12 V eine gute Wahl, da diese Spannung von der Bordbatterie bereitgestellt wird. Um die Hardware der Evaluierungsplattform vor Überspannung und Überlastung zu schützen, befindet sich hinter dem 12 V-Netzteil eine Schutzschaltung. Neben der direkten Versorgung des Mikrocontroller-Moduls speist die 12 V Versorgungsspannung den Schaltregler TPS54350 für das 5 V-Netz. Dieser bietet mit bis zu 3 A genügend Ausgangsstrom, um den Abwärtsregler TPS54350 des 3,3 V-Netzes und die einstellbare Spannungsversorgung zu versorgen.

Mit Ausnahme der einstellbaren Spannungsversorgung für das FPGA-Modul werden dem Drahtlos-Modul alle Boardspannungen zur Verfügung gestellt. Dadurch stehen die drei gängigen Spannungen (12 V, 5 V und 3,3 V) zur Verfügung, wodurch die Notwendigkeit zur Entwicklung einer zusätzlichen Spannungsversorgung auf dem Drahtlos-Modul unwahrscheinlich ist.

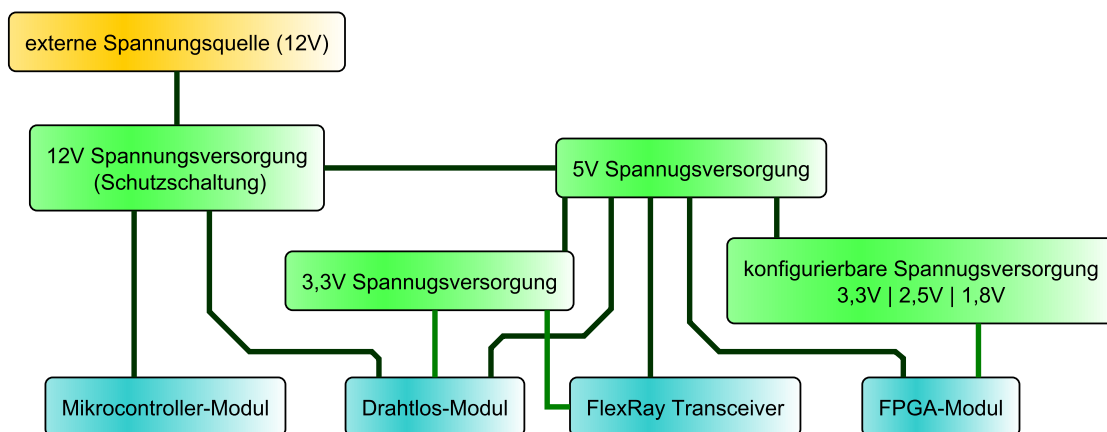


Abbildung 4.1.: Hierarchischer Aufbau der Spannungsversorgung auf der Evaluierungsplattform

4.2.1. Eingangsschutzschaltung

Hinter dem Eingang der externen Spannungsversorgung ist eine Schutzschaltung (vgl. Abbildung 4.2) realisiert. Diese orientiert sich am Schaltplan des TI HDK [Tex12, S. 27] und greift dadurch auf erprobtes Wissen zurück. Eine Surpressordiode (D_1) schützt die Hardware der Evaluierungsplattform zunächst vor Überspannung, insbesondere bei Electrostatic Discharge (ESD)-Ereignissen. Dabei kann die Diode Spitzenleistungen von 1500 W bei einer typischen Anspruchszeit von 1 ps absorbieren. Anders als bei Varistoren bricht die Spannung nicht gänzlich ein, sondern nur bis zu einer definierten Grenze (hier 18 V) [Fai14]. Der Schaltregler hinter der Schutzschaltung kann dieser Spannung standhalten und arbeitet somit auch im Falle einer Überspannung. Außerdem kommt eine SMD-Sicherung (F_1) zum Einsatz, um die Hardware vor Überlastung zu schützen. Für die Sicherung ist eine Haltevorrichtung vorgesehen, um diese schnell austauschen zu können.

Neben den genannten Schutzelementen kommt zudem ein SMD-Ferrit BLM41P750 für die Filterung hochfrequenter Störsignale zum Einsatz. Die Impedanz des SMD-Ferrits ist bei 100 MHz mit 75Ω angegeben (vgl. [Mur11, S. 10]). Da es sich hier nur um generelle Filterung auf der Leistungsseite handelt, sind andere Werte ebenfalls geeignet, solange das SMD-Ferrit den Nennstrom der Schaltung verträgt. Das hier eingesetzte Ferrit weist eine Stromfestigkeit von 3 A auf.

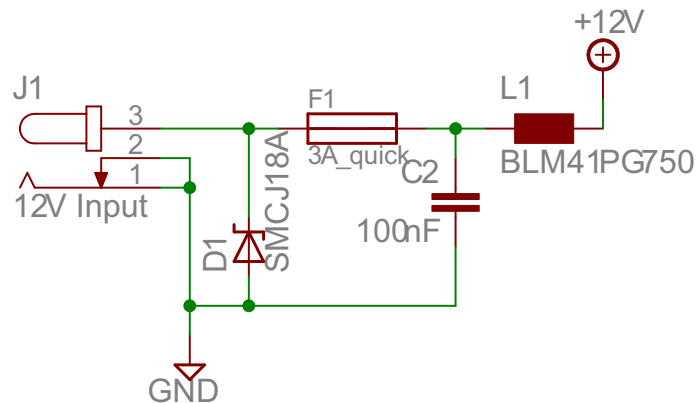


Abbildung 4.2.: Schaltplan der Schutzschaltung für die 12V Spannungsversorgung

4.2.2. Boardspannungsnetz für 5V und 3,3V

Die 5 V-Boardspannung dient der Versorgung der FlexRay-Transceiver und der einstellbaren Spannungsversorgung für das MicroZed FPGA-Modul. Weiterhin speist die 5 V-Boardspannung auch den Regler für das 3,3 V-Netz. Aus der Tabelle 4.1 kann abgelesen werden, dass zunächst ein Ausgangsstrom von $2,18 \text{ A} + 70 \text{ mA} = 2,25 \text{ A}$ benötigt wird. Hinzu kommt die Anforderung einen Strom von etwa 500 mA im 3,3 V-Netz zu gewährleisten, um das zukünftige Drahtlos-Modul zu versorgen. Der benötigte zusätzliche, maximale Ausgangsstrom des 5 V-Reglers kann dann durch Gleichsetzen der Leistung errechnet werden (vgl. Gleichung 4.2).

$$P_{5V} = P_{3,3V} \rightarrow U_{3,3V} \cdot I_{3,3V} = U_{5V} \cdot I_{5V} \cdot \eta \quad (4.1)$$

$$I_{5V} = \frac{U_{3,3V} \cdot I_{3,3V}}{U_{5V} \cdot \eta} \quad (4.2)$$

Geht man zunächst von einem relativ schlechten Wirkungsgrad von 80 % aus, errechnet sich der Ausgangsstrom des Reglers zu $412,5 \text{ mA}$. Insgesamt wird dann also eine Stromzufuhr von $2,66 \text{ A}$ benötigt, die von dem ausgewählten Schaltregler TPS54350 bereit gestellt werden kann [Tex04].

Bei dem TPS54350 handelt es sich um einen Abwärts-Schaltregler mit konfigurierbarer Ausgangsspannung und einem Wirkungsgrad von 90 % und mehr. Der Vorteil von Schaltreglern liegt im Allgemeinen in dem hohen Wirkungsgrad im Vergleich zu einfachen Linearreglern, die ihre Verlustleistung in Wärme umsetzen. Gespeist wird der Regler von dem 12 V-Boardspannungsnetz der Evaluierungsplattform. Das Datenblatt zeigt eine Konfiguration des Schaltreglers für die Abwärtswandlung auf 5 V (vgl. [Tex04, S.21]). In dieser Konfiguration liefert der Schaltregler einen maximalen Ausgangsstrom von 3 A und deckt die Anforderung an den Strombedarf somit ab. Die benötigten Bauteile der Reglerschaltung können über Gleichungen des Datenblatts dimensioniert werden. In diesem Fall wird auf die Bauteilvorgaben bzw. Empfehlungen des Beispiels zurückgegriffen, was die Entwicklungsarbeit vereinfacht. Neben den empfohlenen Stütz- und Entstörkondensatoren kommt ein SMD-Ferrit zum Einsatz (vgl. Abbildung 4.3). Dieses dämpft die am Ausgang des Reglers auftretenden Anteile der Schaltfrequenz und verhindert dadurch ein Übersprechen dieser auf die Signalleitungen der Evaluierungsplattform. Um generell zu verhindern, dass Frequenzanteile des Schaltreglers in den Nutzbereich des Systems fallen, kann die Schaltfrequenz über den Widerstand R_1 an Pin RT des Reglers eingestellt werden. Der gewünschte Bauteilwert kann über Gleichung 4.3 berechnet werden (vgl. [Tex04, S.15]).

$$\frac{R_1}{k\Omega} = \frac{46000}{\frac{f_s}{\text{kHz}} - 35,9} \quad (4.3)$$

Bei Nicht-Bestückung (Do Not Populate (DNP)) oder Verbindung mit Masse wird die interne Standardfrequenz von 500 kHz eingestellt. Im zugrunde liegenden Fall wird R_1 zu $80,6\text{ k}\Omega$ dimensioniert. Durch Umstellen der Gleichung 4.3 errechnet sich die Schaltfrequenz zu $606,6\text{ kHz}$. Die Messung der Schaltfrequenz (siehe Abbildung A.9) bei 588 kHz zeigt eine Abweichung von errechnetem Wert zu gemessenem Wert von etwa 3%. In Anwendungsfällen bei denen trotz der Dämpfung (hier wird eine Dämpfung von etwa $26,25\text{ dBm}$ gemessen) Anteile der Schaltfrequenz in den Nutzfrequenzbereich fallen, muss diese Abweichung bei der Konfiguration der Schaltfrequenz berücksichtigt werden.

Hängt eine externe Komponente von der Ausgangsspannung ab, so muss die Startzeit des Spannungsreglers beachtet werden. Dazu dient der PWRGD-Pin, welcher über den Test Point 22 für externe Schaltungen erreichbar ist. Wird die gewünschte Ausgangsspannung erreicht, liefert dieser einen HIGH-Pegel.

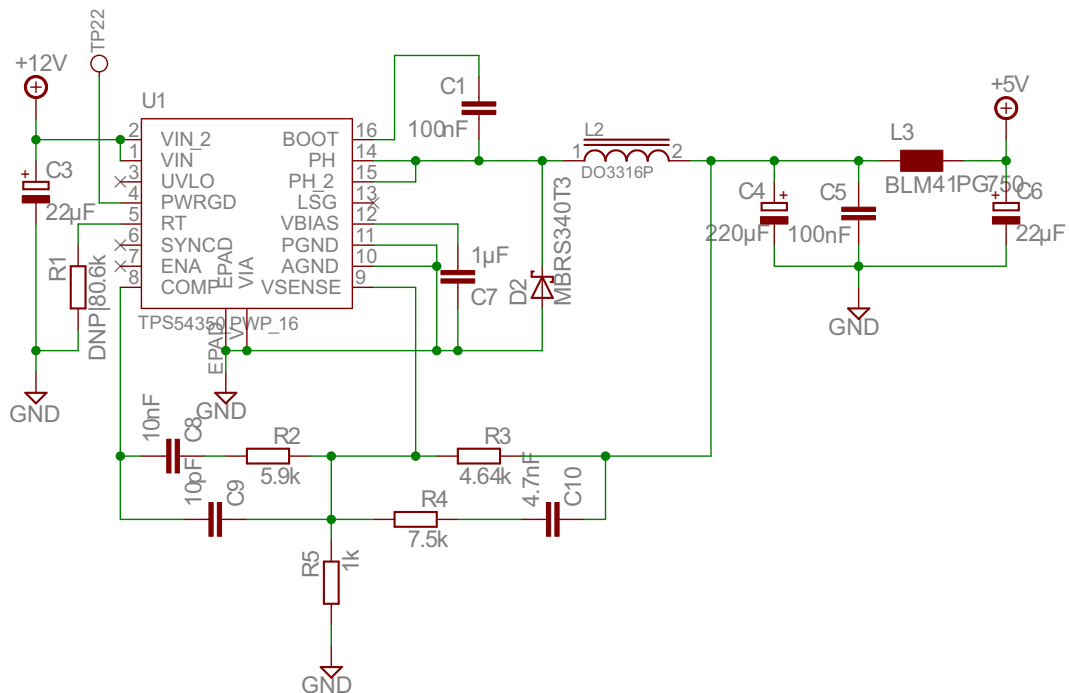


Abbildung 4.3.: Beschaltung des 5 V Abwärtswandlers

Der Spannungsregler TPS54350 speist zwei weitere Spannungsnetze, das $3,3\text{ V}$ Spannungsnetz und die einstellbare Spannungsversorgung für die FPGA-Aufsteckplatine (siehe Abbildung 4.1). Die $3,3\text{ V}$ -Boardspannung dient zum einen der Versorgung der FlexRay-Transceiver, zum anderen stellt sie die geplante Versorgungsspannung für das Drahtlos-Modul zur Verfügung. Die Wahl des Reglers fällt hier auf den Low-Dropout (LDO)-

Spannungsregler TPS73733 [Tex13a]. Bei einem Low-Dropout Spannungsregler handelt es sich um einen Linearregler mit minimal erforderlicher Differenz zwischen Ein- und Ausgangsspannung [Wik15d]. Der Betrieb über die 5 V-Boardspannung verringert somit die Verlustleistung des Reglers bei, im Vergleich zu ähnlichen Schaltreglern, geringem Kostenfaktor. Mit einem maximalen Ausgangsstrom von 1 A deckt der Spannungsregler zudem die Anforderung, mindestens 500 mA für den Betrieb des Drahtlos-Moduls zu liefern (vgl. Abschnitt 4.1).

Abbildung 4.4 zeigt die Beschaltung des LDO-Spannungsreglers für das 3,3 V Spannungsnetz. Die Parametrisierung folgt auch hier den Datenblattangaben. Die Kondensatoren C_{13} und C_{14} am Ausgang des Reglers gewährleisten die Stabilität der Schaltung. Laut Datenblatt sollte hier mindestens insgesamt $1 \mu\text{F}$ eingesetzt werden (vgl. [Tex13a, S.12]). Über den Widerstand R_9 wird der Enable (EN)-Pin dauerhaft auf HIGH-Level gezogen und der Spannungsregler bleibt aktiviert, sofern das 5 V Spannungsnetz stabil bleibt. Der Kondensator C_{12} leitet interne Störspannungen des Reglers über den Pin NR/FB nach Masse ab. Das Datenblatt empfiehlt hier einen 10 nF Kondensator, welcher einen Dämpfungsfaktor der Frequenzanteile um etwa 3,2 im Bereich 10 Hz bis 100 kHz zur Folge hat (vgl. [Tex13a, S.13]). Die LEDs (LED_1 , LED_2 und LED_3) sind direkt ans jeweilige Spannungsnetz der Evaluierungsplattform angeschlossen und bieten eine optische Betriebsanzeige für die Spannungsversorgung.

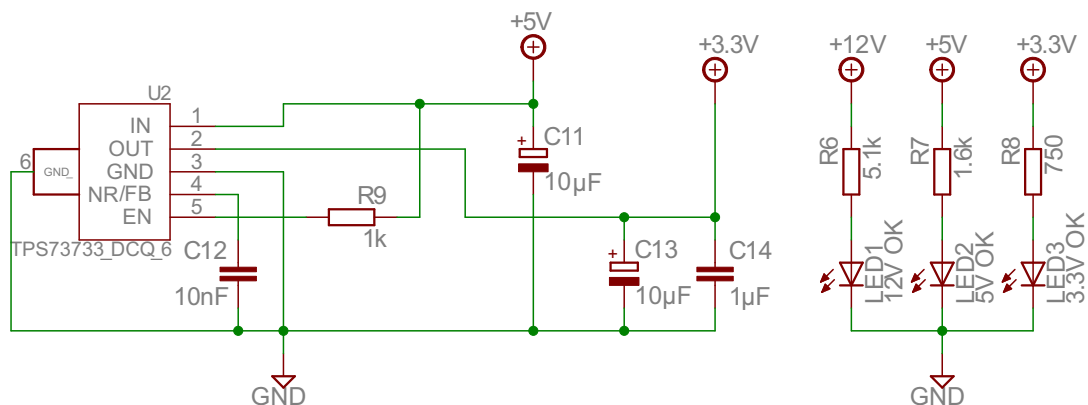


Abbildung 4.4.: Beschaltung des 3,3V Abwärtswandlers und der Status-LEDs der Spannungsversorgung

4.2.3. Spannungsnetz für das FPGA-Modul

Das MicroZed FPGA-Modul kann zur Entwicklung eines SOC zunächst als stand-alone Modul verwendet werden. In dieser Konfiguration können jedoch nur acht I/O-Pins onboard verwendet werden. Die Anbindung an das Mikrocontroller- und Drahtlos-Modul über lediglich acht I/O-Pins läuft Gefahr, einer zukünftigen Entwicklungsarbeit nicht genügend Spielraum für die Kommunikation zur Verfügung zu stellen. Abhilfe schaffen weitere I/O-Pins, die über Steckerleisten auf die Evaluationsplattform geführt werden. Um die Treiber der I/O-Pins zu versorgen, wird eine zusätzliche Spannungsversorgung benötigt, die von der Trägerplatine gestellt werden muss. Dabei muss beachtet werden, die Spannungsversorgung in die Startsequenz des MicroZeds zu integrieren. Abbildung 4.5 zeigt das Ablaufdiagramm für den Startvorgang der Spannungsversorgung (vgl. Benutzerhandbuch des MicroZeds [Avn15c]). Sobald die auf dem MicroZed generierte 1,8 V Spannung ihren Endzustand erreicht, wird der Startvorgang der Spannungsversorgung auf der Evaluierungsplattform eingeleitet. Erreichen sowohl die weiteren Spannungsversorgungen auf dem MicroZed, als auch die auf der Evaluierungsplattform ihren Endzustand, bootet das Zynq7020 FPGA.

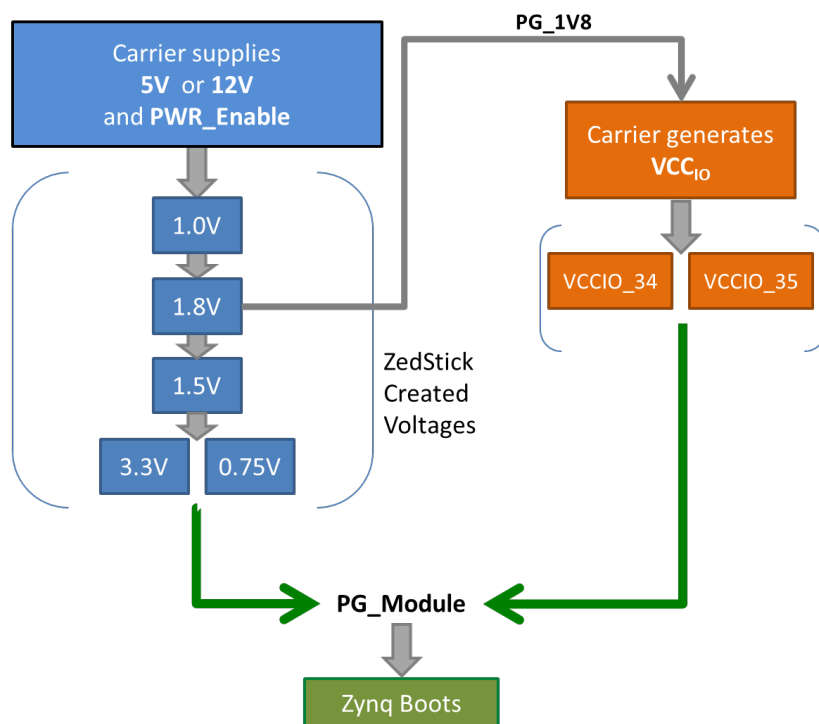


Abbildung 4.5.: Sequenzdiagramm der MicroZed-Spannungsversorgung während des Startvorgangs [Avn15c, S. 29]

Die Spannungsversorgung der Break-Out-Carrier Trägerkarte von Avnet [Avn15b] erfüllt die oben genannten Anforderungen. Die Spannungsversorgung auf der Evaluierungsplattform wird daher von dem Schaltplan der Trägerplatine von Avnet übernommen [Avn14a, S. 3]. Kern der Schaltung ist der Abwärts-Schaltregler LMR10510 [Tex13b]. Dieser eignet sich vor allem wegen seiner kleinen Gehäusegröße, da die Spannungsversorgung platzsparend unter dem FPGA-Modul untergebracht wird. Zudem bietet die Wahl eines Schaltreglers, wie bereits im Fall der 5 V Boardspannung, einen hohen Wirkungsgrad von über 90 %.

Eingeschaltet wird die Spannungsversorgung durch Setzen des Schalters S_1 (PWR_SW) auf die Position ON. Über JP_1 kann die Ausgangsspannung des Reglers mittels Jumper zu 3,3 V (Standard, oberer Jumper gesetzt), 2,5 V (mittlerer Jumper gesetzt) und 1,8 V (unterer Jumper gesetzt) eingestellt werden [Avn14b, S. 14]. Dabei ist zu beachten, dass das Mikrocontroller-Modul und FPGA-Modul über diverse Leitungen miteinander verbunden sind und daher zunächst nur die 3,3 V Einstellung verwendet werden sollte, da das Mikrocontroller-Modul nur mit diesem Spannungspegel arbeiten kann. Die Konfigurierbarkeit der Spannungsversorgung macht eine stand-alone Entwicklung ohne Mikrocontroller-Modul möglich und bietet daher Entwicklungsspielraum für zukünftige Arbeiten.

Über die PG_LED (LED₄) wird die Betriebsbereitschaft der Spannungsversorgung signalisiert. Die LED wird dazu vom MicroZed über den N-Fet Transistor Q_3 aktiviert, sobald die gesamte Spannungsversorgung ihren Endzustand erreicht und das Power Good Signal gesetzt wird (vgl. Signal PG_Module in Abbildung 4.5). Wird die Spannungsversorgung über den Schalter S_1 ausgeschaltet, zieht der N-Fet Transistor das PWR_EN Signal auf Masse und deaktiviert dadurch auch die interne Spannungsversorgung auf dem MicroZed. Wird der Jumper JP_2 gesetzt, kann die Spannungsversorgung zunächst ohne MicroZed in Betrieb genommen und geprüft werden. Die Bauteile U_5 (Op-Amp) und U_6 (Spannungsreferenzdiode) bilden einen Spannungskomparator zur Kontrolle des Startvorgangs der Spannungsversorgung. Durch die Beschaltung wird sicher gestellt, dass der Schaltregler erst dann über das Signal VADJ_EN aktiviert wird, sobald die Eingangsspannung der gesamten Schaltung etwa 4 V erreicht. [Avn14b, S. 12 ff.]

Die Abbildung 4.6 zeigt den Schaltplan der einstellbaren Spannungsversorgung für die Versorgung der externen I/O-Pins des MicroZed FPGA-Moduls.

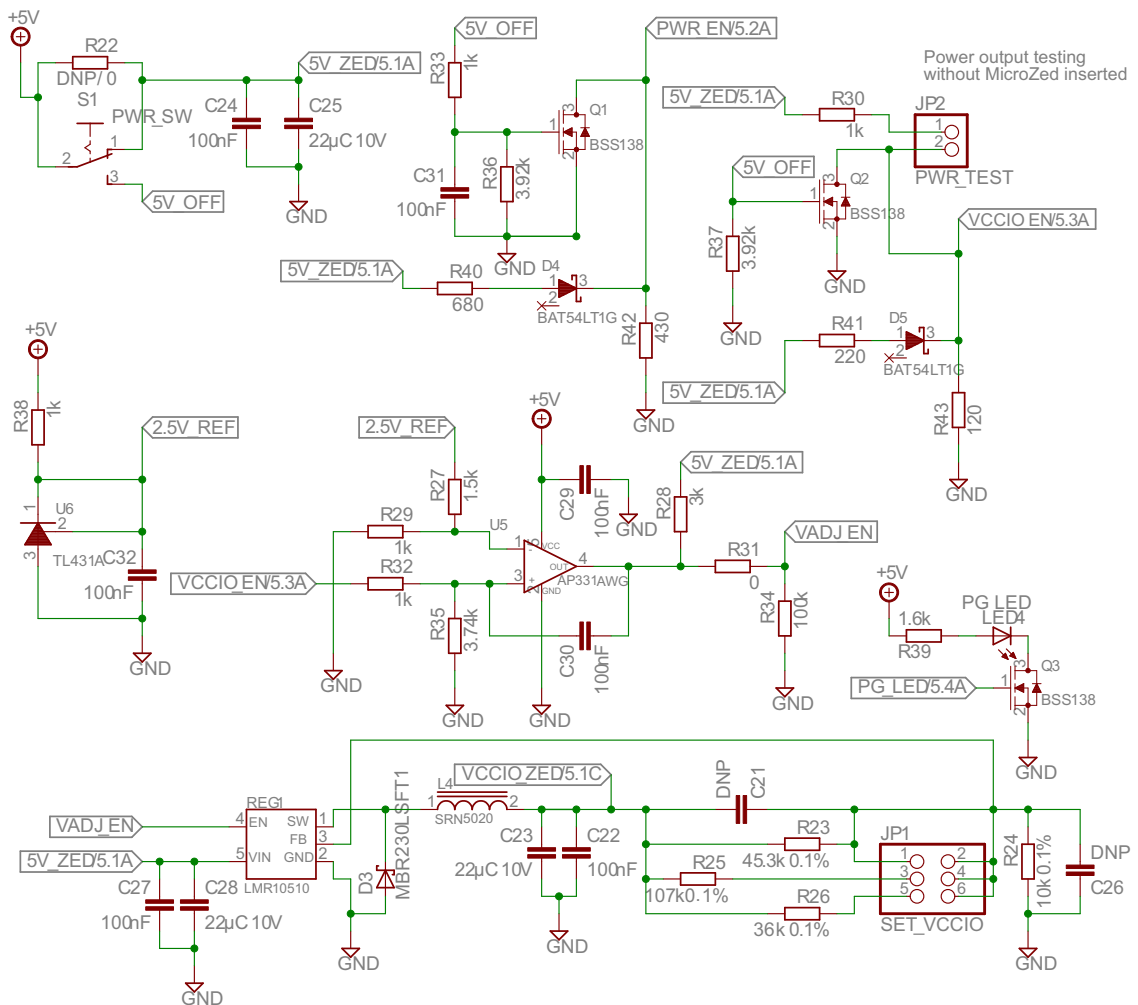


Abbildung 4.6.: Schaltplan für die zusätzliche Spannungsversorgung der FPGA-Aufsteckplatine

4.3. FlexRay-Transceiver

Für die kabelgebundene FlexRay-Datenkommunikation werden für jeden Teilnehmer zwei FlexRay-Transceiver benötigt (je einer pro Kanal). Diese passen die vom Host empfangenen Logikpegel auf den geforderten Buspegel an und codieren die Daten (vgl. Abschnitt 2.2). Wie aus Abschnitt 3.2 bereits hervorgeht, werden hier die FlexRay-Transceiver TJA1082 von NXP eingesetzt, da diese neben der aktuellen FlexRay Spezifikation auch eine Fehleranalyse über eine SPI-Schnittstelle bieten. Die Beschaltung der Transceiver folgt maßgeblich den Application Hints von NXP [NXP10, S.13 ff.]. Zusätzlich erfolgen einige Abwandlungen, die zur Störsicherheit beitragen sollen. Diese basieren auf dem Schaltplan des TI Experimentier Board [Tex11, S. 9]. Dieses Board dient als Referenz für Entwickler und ist nicht käuflich erhältlich. Abbildung 4.7 zeigt die Beschaltung der FlexRay-Transceiver TJA1082. Dargestellt wird hier der Kanal eins bzw. Channel A des FlexRay-Bussystems. Der zweite Kanal bzw. Channel B ist analog aufgebaut.

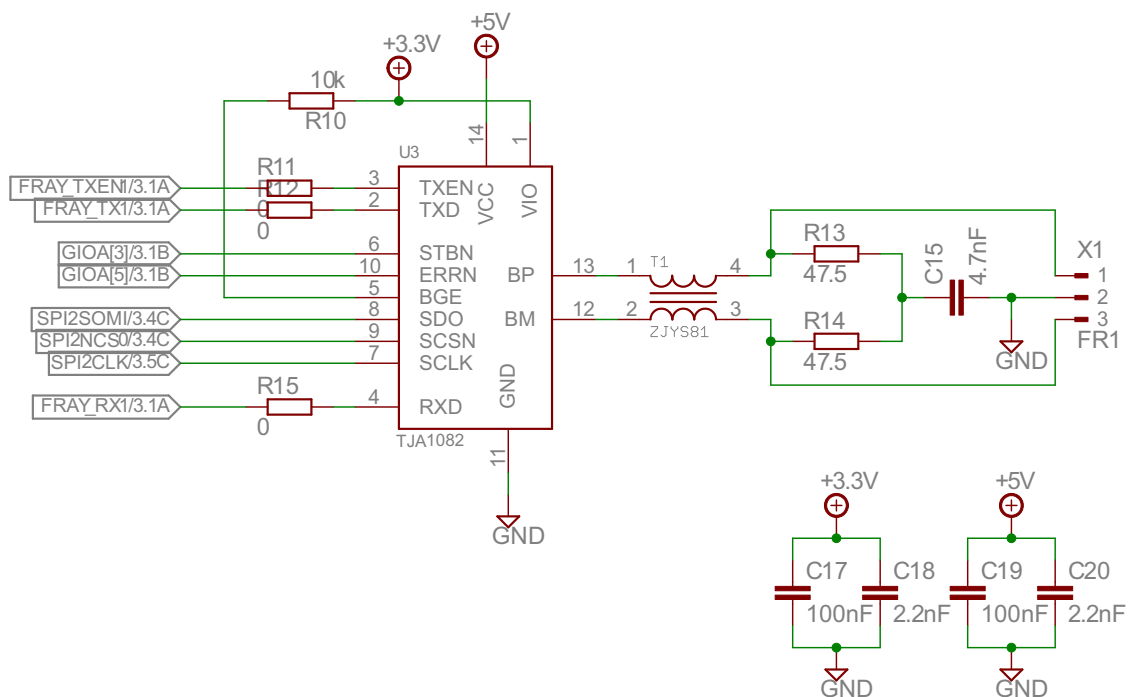


Abbildung 4.7.: Ausschnitt zur Beschaltung der FlexRay-Transceiver TJA1082 (hier gezeigt wird Kanal A)

Die Entstördrossel T_1 vom Typ ZJYS81R5-2P50T-G01 dient der Filterung von Gleichtaktstörungen und wird nach dem Schaltplan von TI ausgewählt. Für die Stütz- und Endstörkondens-

satoren C_{17} bis C_{20} der Versorgungsspannung werden die laut NXP empfohlenen 100 nF mit 2,2 nF Kondensatoren ergänzt. Durch den Einsatz der 2,2 nF Kondensatoren können auch Störungen höherer Frequenzen abgeleitet werden [Tex06, S. 13].

Der Split-Kondensator C_{15} dient, wie auch die Entstördrossel, als Filterung der Gleichtaktstörung. Die Störanteile im hohen Frequenzbereich werden nach Masse kurzgeschlossen [Ort05, S. 119 ff.]. NXP gibt hier 4,7 nF bis 47 nF als empfohlenen Wertebereich an. Die Widerstände der Buserminierung sollen im gesamten System eingehalten werden [Fle06a, S. 10]. In diesem Fall handelt es sich um ein 95Ω System. Die Widerstände R_{13} und R_{14} sind demnach auf $Z_{0/2} = 47,5 \Omega$ zu dimensionieren. Um die elektromagnetischen Emissionen möglichst gering zu halten, sollen die Widerstände der Terminierung R_{13} und R_{14} zudem möglichst kleine Bauteiltoleranzen aufweisen. Die dazu notwendige Bedingung für die Abweichung wird durch Gleichung 4.4 beschrieben.

$$\frac{2 \cdot |R_{13} - R_{14}|}{R_{13} + R_{14}} \leq 2\% \quad (4.4)$$

Laut den Application Notes der FlexRay Physical Layer Specification ist diese Voraussetzung mit Bauteiltoleranzen von 1 % erfüllt [Fle06a, S. 10].

Über die 0Ω Widerstände R_{11} , R_{12} und R_{15} kann die Verbindung von Mikrocontroller- oder FPGA-Modul zum jeweiligen Transceiver auf Hardwareebene getrennt werden. Dadurch kann ein eventuell auftretender Spannungsabfall an einem der Module verhindert werden. Um die Node-Transceiver zu aktivieren, muss der BGE Pin auf HIGH-Potential gebracht werden. Da dieses System keinen Bus-Supervisor zur Aktivierung vorsieht, wird der Pin über R_{10} auf V_{IO} -Pegel gebracht. Die Transceiver sind somit durchgehend aktiviert. Über den STBN Pin kann der Host, hier also das HDK, die Transceiver via LOW-Pegel auf Standby-Betrieb schalten. Ein HIGH-Pegel initiiert den Übergang in den Normal-Betrieb.

Der ERRN Pin dient der simplen Fehleridentifikation. Registriert der Transceiver einen Fehler, so teilt er dies mittels LOW-Pegel an ERRN dem Host mit. Für eine nähere Betrachtung der Fehler können nun über die SPI-Schnittstelle die 16 Statusbits des Transceivers ausgewertet werden. Unterabschnitt 6.2.2 geht näher auf die Statusbits und deren Auswertung ein. Die für die FlexRay-Transceiver interessanten Signalleitungen auf der Evaluierungsplattform sind in Tabelle 4.2 zusammengefasst.

Signalname	Pin TI HDK	Pin MicroZed	Pin TJA1082 (U3,U4)
FRAY_RX1	13, HDK_J11	29, JX1	RXD, U3
FRAY_TX1	15, HDK_J11	17, JX1	TXD, U3
FRAY_TXEN1	17, HDK_J11	23, JX1	TXEN, U3
FRAY_RX2	14, HDK_J11	47, JX1	RXD, U4
FRAY_TX2	16, HDK_J11	35, JX1	TXD, U4
FRAY_TXEN2	18, HDK_J11	41, JX1	TXEN, U4
GIOA[3]	21, HDK_J11		STBN, U3
GIOA[4]	24, HDK_J11		STBN, U4
GIOA[5]	23, HDK_J11		ERRN, U3
GIOA[6]	26, HDK_J11		ERRN, U4
SPI2SOMI	59, HDK_J10		SDO, U3 and U4
SPI2NCS1	62, HDK_J10		SCSN, U4
SPI2NCS0	53, HDK_J10		SCSN, U3
SPI2CLK	64, HDK_J10		SCLK, U3 and U4

Tabelle 4.2.: Signalleitungen der Node Transceiver (vgl. A.1)

4.4. Signalleitungen der Evaluierungsplattform

Für die Signalleitungen des Mikrocontroller-Moduls ist zunächst zu beachten, dass der verwendete TMS570LS3137 Mikrocontroller die Funktionen seiner Peripherie-Module über Multiplex-Pins zur Verfügung stellt. Dabei sind einige Funktionen auf ein und demselben Pin verfügbar. Im Umkehrschluss bedeutet dies, dass nicht gleichzeitig alle angebotenen Peripherie-Module des Mikrocontrollers genutzt werden können, da sich Konflikte der Pin-Konfiguration ergeben. Das von TI angebotene Software-Tool HALCOGEN [Tex15a] bietet eine graphische Benutzeroberfläche, mit der die gewünschten Funktionen der Peripherie-Module ausgewählt werden können. Konflikte, die sich aus der Pin-Konfiguration ergeben, werden in einer Tabelle angezeigt und können dort behoben werden, indem auf weniger hoch priorisierte Funktionen verzichtet wird. Dieses Tool bietet zwar eine gute Unterstützung im Umgang des Mikrocontrollers, jedoch berücksichtigt es nicht die Infrastruktur des HDK. Daher wird zusätzlich mit dem Schaltplan des HDK abgeglichen [Tex12], welche Signale unter welchen Voraussetzungen verwendet werden können. Um möglichst alle Funktionen des HDK verwenden zu können, werden nur diejenigen Signalleitungen nach außen geführt, die in jedem Fall frei sind. Die Ausnahme bilden die SPI2-Signalleitungen, die auf dem HDK zur Kommunikation mit einer SD-Card vorgesehen sind. Dieses Interface wird jedoch für die Kommunikation mit den FlexRay-Transceivern benötigt (vgl. Abschnitt 4.3). Alle anderen SPI-Signalleitungen werden über die Multiplex-Pins für andere Funktionen vorgesehen.

SPI1 und SPI5 sind bereits für die Kommunikation mit dem Ethernet-Controller vorgesehen. SPI3 fällt als Ausweichoption ebenfalls weg, da diese für die UART-Kommunikation mit dem PC benötigt wird. Die Pins des SPI4-Interface werden bereits in einer General Purpose Input/Output (GPIO)-Konfiguration für die Ansteuerung der LEDs benötigt. Die Tabelle 4.3 zeigt die wichtigsten Signalleitungen des Mikrocontroller-Moduls. Eine vollständige Tabelle aller Signalleitungen der Evaluierungsplattform befindet sich im Anhang (vgl. Tabelle A.1).

Für die Kommunikation mit dem Drahtlos- und dem FPGA-Modul sind jeweils zwei frei verwendbare GPIO-Pins vorgesehen. Zusätzlich stehen zwei Timer-Pins (High-End Timer (HET)) zur Verfügung, welche ebenfalls als GPIO-Pins konfiguriert werden können. Statussignale können dadurch flexibel zwischen Mikrocontroller- und Drahtlos-Modul ausgetauscht werden. Sollte zudem eine analoge Messgröße ausgewertet werden müssen, kann dies über zwei Pins des 12-Bit Analog-to-Digital Converter (ADC) erfolgen. Häufig bieten Integrated Circuit (IC)s eine erweiterte Kommunikationsfunktion über die Bereitstellung einer UART-Schnittstelle. Benötigt ein zukünftiger Controller des Drahtlos-Moduls diese Funktion, kann über die gerouteten SCIRX und SCITX Signalleitungen dann eine Kommunikation über UART aufgebaut werden.

Die Signalleitungen der Bussysteme LIN und CAN werden mit dem FPGA-Modul verbunden. Dadurch kann der zukünftige Wireless Physical Layer auch mit diesen Bussystemen realisiert werden. Wird ein drahtgebundener Physical Layer benötigt, kann im Fall von CAN auf die Transceiver des HDK zurückgegriffen werden. Für die Entwicklung einer LIN-Transceiverschaltung können die Signale zudem vom FPGA-Modul unverändert auf das Drahtlos-Modul durchgeschleift werden.

Signalname	Pin TI HDK	Pin MicroZed	Pin Drahtlos-Modul
LIN_RX	5, HDK_J11	53, JX1	
LIN_TX	6, HDK_J11	61, JX1	
CAN3_RX	11, HDK_J11	67, JX1	
CAN3_TX	12, HDK_J11	73, JX1	
GIOB[4]	32, HDK_J11		93, J3
GIOB[5]	31, HDK_J11	81, JX1	
GIOB[6]	34, HDK_J11		91, J3
GIOB[7]	33, HDK_J11	82, JX1	
NHET1[2]	40, HDK_J11		81, J3
NHET1[4]	42, HDK_J11		79, J3
SCIRX	44, HDK_J11		97, J3
SCITX	49, HDK_J11		99, J3
AD1IN[0]	32, HDK_J9		85, J3
AD1IN[2]	34, HDK_J9		87, J3

Tabelle 4.3.: Wichtigste Signalleitungen des Mikrocontroller-Moduls (vgl. A.1)

Die Verbindung zwischen FPGA- und Drahtlos-Modul erfolgt über insgesamt 14 Pins. Bei einer Single-Ended Übertragung stehen demnach 14 Signalleitungen zur Verfügung. Alternativ können die Leitungen auch paarweise für eine differentielle Übertragung verwendet werden. Die Tabelle 4.4 zeigt die Signalleitungen für die Kommunikation zwischen FPGA- und Drahtlos-Modul. Die Signale MOD_DQ_1 und MOD_DQ_2 werden auf der Platine mit möglichst kurzen Leiterbahnen verlegt und sind für die Datenübertragung mittels I&Q-Verfahren vorgesehen [Wik15c].

Signalname	Pin MicroZed	Pin Drahtlos-Modul
MOD_DQ_1	11, JX1	98, J3
MOD_DQ_2	13, JX1	100, J3
R_MIX_3_P	18, JX1	94, J3
R_MIX_3_N	20, JX1	92, J3
R_MIX_5_P	24, JX1	88, J3
R_MIX_5_N	26, JX1	86, J3
R_MIX_7_P	30, JX1	82, J3
R_MIX_7_N	32, JX1	80, J3
R_MIX_9_P	36, JX1	76, J3
R_MIX_9_N	38, JX1	74, J3
R_MIX_11_P	42, JX1	70, J3
R_MIX_11_N	44, JX1	68, J3
R_MIX_13_P	48, JX1	64, J3
R_MIX_13_N	50, JX1	62, J3

Tabelle 4.4.: Signalleitungen für die Kommunikation zwischen FPGA- und Drahtlos-Modul (vgl. A.1)

Neben den beschriebenen Signalleitungen sind noch zwei zusätzliche Features auf der Platine der Evaluierungsplattform implementiert. Über die Kupferflächen des Jumpers JP₃ kann eine Batterie mit einer Spannung von 1 V bis 1,89 V angeschlossen werden. Diese versorgt einen RAM auf dem MicroZed, in dem eine Verschlüsselungsfolge (Encryption Key) gespeichert werden kann. Der Encryption Key dient dem sicheren Booten des Zynq FPGAs [Avn15c, S. 30]. Werden mehr Signalleitungen des FPGA-Moduls benötigt, können acht weitere I/O-Pins über die Kupferflächen der Test Points am linken Platinenrand abgegriffen werden (vgl. Abbildung A.1). Für die Verbindung externer Signale an das Drahtlos-Modul stehen ebenfalls acht Test Points zur Verfügung, welche auf die linke Steckerleiste des Drahtlos-Moduls führen (vgl. Abbildung A.8). Für eine mögliche Anbindung externer Bauteile an das HDK wird zudem ein HET-Pin (N2HET1[20]) auf die Platine geroutet (Abgriff an Test Point 20). Dieser kann als GPIO-Pin konfiguriert werden und bietet beispielsweise auch eine Möglichkeit, ein einfaches Hardware-Debugging durchzuführen.

4.5. Layout der Platine

Dieser Abschnitt erläutert zunächst die Layout-Entwicklung der Evaluierungsplattform und beschreibt anschließend den Bestückungsprozess der realisierten Platine. Die Entwicklung erfolgt mit der PCB-Design Software EAGLE (Version 5.2.0). Viele Bauteile, unter anderem die ICs der Spannungsversorgung stehen nicht in den EAGLE-Standardbibliotheken zur Verfügung. Daher werden die benötigten Bauteile nach Datenblattangaben nachgezeichnet und in eine eigens angelegte Bibliothek gepflegt. Diese liegt der Arbeit auf der CD bei und kann für eine Weiterentwicklung der Evaluierungsplattform verwendet werden. Die Platine trägt in EAGLE die Bezeichnung FlexRay Junction Board (FRJB) (dt. FlexRay-Verbindungsboard).

Die für das Drahtlos-Modul vorgesehene Fläche auf der Evaluierungsplattform hat die Abmessungen von 80 mm in der Länge und 65 mm in der Breite. Eine Entwicklung mit der freien Software EAGLE (Light Version) ist dadurch möglich, da diese eine Platinengröße von maximal 80 mm mal 100 mm erlaubt. Die Entwicklung erfolgt somit losgelöst von teuren Software-Lizenzen. Außerdem kann bei der Verwendung von EAGLE auf die bereits erwähnte Bauteilbibliothek zurückgegriffen werden. Generell sind auch andere Maße für das Drahtlos-Modul möglich, sofern ein Überstehen der Modulplatine über die Platine der Evaluierungsplattform in Kauf genommen wird. Die Verbindung von Drahtlos-Modul und Evaluierungsplattform erfolgt über zwei Bergstak-Steckerleisten. Diese Stecker bieten eine Stromfestigkeit von mindestens 800 mA pro Pin und sind für höchste Übertragungsraten geeignet (weniger als 10 dB Rückflusdämpfung bei 5 GHz) [FCI15]. Auf kleinstem Raum werden zudem 100 Pins für diverse Signalleitungen oder Versorgungsleitungen bereitgestellt. Die Steckerleisten werden auch bei dem MicroZed FPGA-Modul verwendet, welches mit einer maximalen Taktfrequenz von 866 MHz im Rahmen der Übertragungsraten der Steckerleisten liegt [Xil14]. Die Tabelle A.1 bietet eine Übersicht über alle Signalleitungen der Evaluierungsplattform.

Hinweise zur Leiterbahnbreite

Der in den Leiterbahnen fließende Strom hat eine Wärmeentwicklung zur Folge, die berücksichtigt werden muss, um thermische Schäden der Leiterbahnen zu vermeiden. Im worst case würde eine unzureichend abgeführte Wärme zum Durchbrennen einer Kupferleitung führen. Insbesondere bei den Leiterbahnen der Spannungsnetze, in denen hohe Ströme fließen, muss daher eine ausreichend hohe Strombelastbarkeit der Kupferbahnen gewährleistet sein. Dazu wird die Leiterbahnbreite des jeweiligen Spannungsnetzes nach der Tabelle 4.5 dimensioniert.

Leiterbahnbreite	Strombelastbarkeit
0,010 in	0,8 A
0,015 in	1,2 A
0,020 in	1,5 A
0,050 in	3,2 A

Tabelle 4.5.: Strombelastbarkeit von 35µm dicken Kupferleiterbahnen in Abhängigkeit der Leiterbahnbreite

Die Werte der Tabelle sind aus [Tex05] übernommen und zeigen die Strombelastbarkeit von 35µm dicken Kupferleiterbahnen in Abhängigkeit der Leiterbahnbreite. Für die angegebene Strombelastung bei entsprechender Leiterbahnbreite wird dann eine Temperaturerhöhung um 10°C gegenüber der Raumtemperatur bei Kupfer-Layern an der Luft angenommen. Die nötige Leiterbahnbreite ergibt sich aus den maximalen Ausgangsströmen der verwendeten Spannungsregler (vgl. Abschnitt 4.2). Die Tabelle 4.6 zeigt die gewählten Leiterbahnbreiten der Signal- bzw. Spannungsnetze auf der realisierten Platine.

Signalnetz bzw. Spannungsnetz	Strombelastung (worst case)	Leiterbahnbreite
5V Spannungsnetz	3 A	0,050 in
3,3V Spannungsnetz	1 A	0,015 in
MicroZed Spannungsnetz	1 A	0,020 in
12V Spannungsnetz	3 A	0,050 in
Kommunikationspfade	≈ 500 mA (gepulst)	0,012 in

Tabelle 4.6.: Leiterbahnbreite der jeweiligen Signal- bzw. Spannungsnetze auf der realisierten Platine

Neben der Strombelastung auf den Kupferleiterbahnen muss zudem die Verlustwärme der Spannungsregler berücksichtigt werden. Insbesondere für den Schaltregler der 5 V Versorgungsspannung sollten die Designvorgaben des Datenblatts [Tex04, S. 13] eingehalten werden. Laut diesen Vorgaben sind mindestens vier Vias unter dem Schaltregler zu platzieren, um eine ausreichend hohe Wärmeabfuhr auf die darunter liegende Kupferfläche zu gewährleisten. Diese muss eine mindestens 3 in mal 3 in große Fläche aufweisen. Da die gesamte Unterseite der Platine als Groundplane ausgelegt wird, ist diese Anforderung erfüllt. Zusätzlich sollen die Kupferbahnen an den Pins der Ein- und Ausgangsspannung als Flächen ausgelegt werden. Abbildung 4.8 zeigt einen Ausschnitt des PCB-Designs. Die erwähnten Vias sind unter dem Schaltregler U_1 zu erkennen. Die Leiterbahnflächen zu den Pins der Ein- und Ausgangsspannung liegen südlich des SMD-Ferrits L_1 bzw. südlich der Diode D_2 .

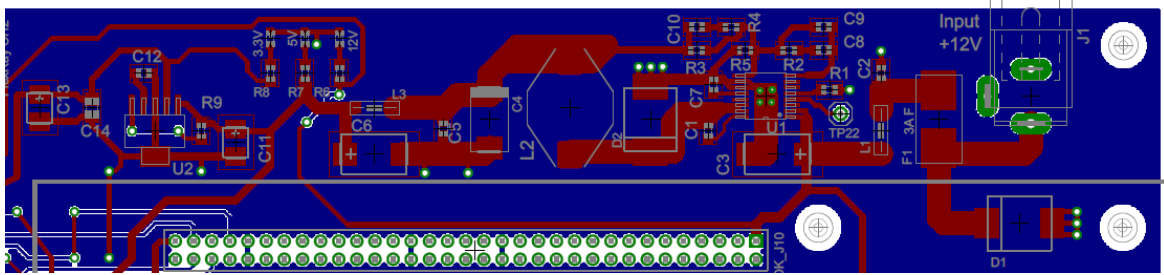


Abbildung 4.8.: PCB Layout für die Sektion der 5 V und 3,3 V Spannungsversorgung

Herstellungsprozess

Die Fertigung der Platine wird von der Firma Beta Layout GmbH übernommen [Bet15a]. Die Bestückung selbst findet intern statt. Dazu werden die Kupferflächen der Platine mithilfe einer Schablone (engl. stencil) mit Lötpaste benetzt. Nach dem Setzen der Bauteile erfolgt dann der Reflow-Lötvorgang in einem elektrischen Ofen mit temperaturgesteuertem Netzregler (der Netzregler ist ebenfalls bei Beta Layout erhältlich, siehe [Bet15b]). Bilder des Aufbaus finden sich in Abschnitt A.6.

Tabelle 4.7 zeigt den Ablauf des Lötvorgangs. Die erste Phase (Preheat) dient dem Vorheizen der Platine, um temperaturbedingte mechanische Spannungsrisse zu vermeiden. In der nächsten Phase, *Soak*, verflüssigt sich das Flussmittel. In der Phase (Reflow) wird die maximale Löttemperatur erreicht. Um sicherzustellen, dass sich solide Lötverbindungen ausbilden, verbleibt diese Temperatur noch während der Phase (Dwell). Insbesondere bei empfindlichen Bauteilen muss auf Einhaltung des korrekten Prozessablaufs geachtet werden. Der hier eingesetzte Prozessablauf ist an der HAW Hamburg bereits erprobt und erwies sich auch bei der Bestückung der zwei hergestellten Prototypen als korrekte Verfahrensweise.

Bezeichnung	Zeitintervall	Temperaturschwelle
Preheat	180 s	130 °C
Soak	90 s	180 °C
Reflow	140 s	230 °C
Dwell	40 s	230 °C

Tabelle 4.7.: Prozessablauf des Reflow-Lötvorgangs

Die Abbildung 4.9 zeigt die Oberseite der Platine des ersten Prototyps. Die Abmessungen der Platine sind 240 mm in der Breite und 130 mm in der Höhe. Bei dieser Größe wird ausreichend Platz für die drei Aufsteck-Module geboten, während für die Bauteile der Spannungsversorgung und der FlexRay-Transceiver möglichst wenig zusätzliche Fläche vorgesehen wird. Die mit weißen Rändern abgegrenzten Steckplätze kennzeichnen die Anbindung für das Drahtlos-Modul („wireless shield“, Abmessung 80 mm mal 65 mm), das FPGA-Modul („MicroZed“, Abmessung 101,6 mm mal 57,15 mm bzw. 4 in mal 2,25 in) und das Mikrocontroller-Modul („TI TMS570LS31x HDK“, Abmessung 125 mm mal 110 mm).

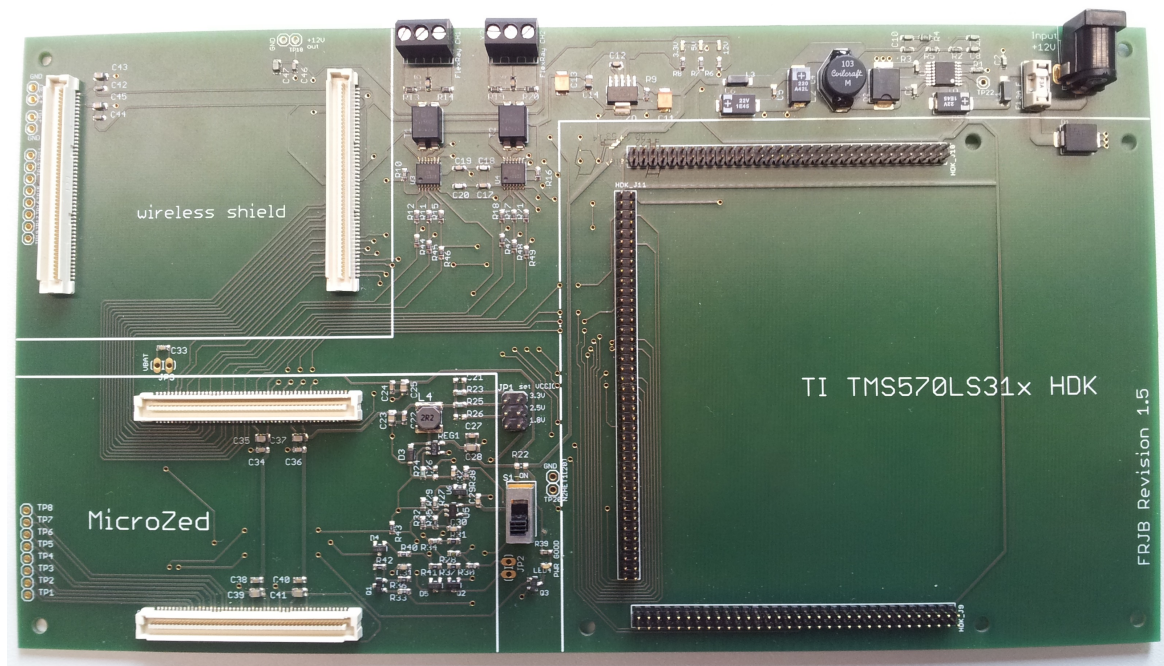


Abbildung 4.9.: Die Oberseite der realisierten Platine (Revision 1.5, Prototyp 1)

5. Mikrocontroller Software

Neben der Hardwareentwicklung umfasst diese Arbeit ebenfalls die Entwicklung einer Software für den Mikrocontroller TMS570LS3137. Diese dient der Verifikation der drahtgebundenen FlexRay-Datenkommunikation. Dieses Kapitel umfasst zunächst grundlegende Informationen der verwendeten Software-Tools und beschreibt anschließend in Abschnitt 5.1 den Aufbau und Ablauf der erstellten Programme für Kommunikationsteilnehmer A und B. Abschnitt 5.2 beschreibt die Protokollzustandsmaschine des integrierten FlexRay-Controllers E-Ray Communication Controller (CC) von Bosch Semiconductors [Rob12]. Abschnitt 5.3 erläutert zudem die Konfiguration des FlexRay-Controllers.

Die Grundlage der Software ist das Demo-Projekt von TI [Tex15c], dessen Funktion die Konfiguration der FlexRay-Controller und die Übertragung von Testdaten zwischen zwei Teilnehmern eines FlexRay-Clusters ist. Ausgehend von den Code-Dateien des Demo-Projekts wird die Software um weitere Funktionen ergänzt, um Nutzdaten zu übermitteln und diese auszuwerten. Für die Entwicklung der Software wird die Programmiersprache C und als Entwicklungsumgebung Code Composer Studio Version 6.0.1.00040 von TI verwendet. Die Funktionen der Peripherie-Module des Mikrocontrollers (z.B. ADC) werden über vordefinierte Code-Dateien zur Verfügung gestellt. Diese können mit Hilfe des Software-Tools HALCOGEN [Tex15a], nach Eingabe der Vorgaben über eine graphische Benutzeroberfläche, generiert werden. Dabei ist zu beachten, dass die Projektdatei der FlexRay-Demo mit HALCOGEN inkompatibel ist. Die generierten Code-Dateien müssen also manuell eingebunden werden, wobei auf mögliche Abhängigkeiten zwischen Header-Dateien zu achten ist. Die Konfiguration des FlexRay-Controllers selbst wird durch HALCOGEN nicht unterstützt.

5.1. Aufbau und Programmablauf der Software

Kernaufgabe der Software ist der Austausch von Daten zwischen zwei FlexRay-Teilnehmern (engl. Nodes). Das Demo-Projekt von TI bietet dazu zunächst eine Konfiguration des FlexRay-Controllers, für den Austausch von Testdaten zwischen den Teilnehmern. Bei den Testdaten handelt es sich um Zahlenfolgen, welche auf jeweils beiden Nodes vor der Übertragung bekannt sind. Die empfangenen Daten werden dann mit der erwarteten Zahlenfolge verglichen, um bei Abweichung eine Variable für die Fehlerzählung zu inkrementieren. Eine

Error-LED signalisiert zudem, dass mindestens eine fehlerhafte Übertragung stattgefunden hat. Um neben den Testdaten auch Nutzdaten zu übermitteln, wird die Software von Node A um weitere Funktionen der Peripherie-Module des Mikrocontrollers ergänzt.

Für die Übertragung von Nutzdaten wird der Temperatursensor des HDKs über den 12-Bit ADC ausgewertet (vgl. Unterabschnitt 6.2.3). Node A sendet diesen dann über die FlexRay-Schnittstelle an Node B. Beide Nodes vergleichen den gemessenen bzw. empfangenen Wert mit einem voreingestellten Maximalwert. Wird dieser überschritten, signalisiert dies jeweils eine LED auf der Platine des HDK.

In der Software für Node A werden außerdem die für die Ansteuerung der SPI-Schnittstelle benötigten Routinen implementiert, um die Statusbits der FlexRay-Transceiver empfangen zu können. Der Unterabschnitt 6.2.2 geht näher auf die Auswertung der Statusbits ein.

Die Kommunikation zwischen PC und Node A erfolgt über ein UART-Modul, welches über eine virtuelle Schnittstelle per USB angesteuert werden kann. TI bezeichnet das dafür zuständige Peripherie-Modul des Mikrocontrollers als Serial Communication Interface (SCI). Über ein Terminal-Programm können die von Node A gesendeten Daten empfangen werden. Listing 5.1) zeigt die Terminal-Einstellung und das Format der übertragenen Daten.

```
Baud:      128000
Data bits: 8
Stop bits: 2
Parity:    None

Read received data as follows:
Line 1 -> ADC value (temp sensor)
Line 2 -> SPI2 NCS[0] data
Line 3 -> SPI2 NCS[1] data
Line 4 -> ERRN pin on bus transceiver channel 1 (A)
Line 5 -> ERRN pin on bus transceiver channel 2 (B)
```

Listing 5.1: Terminal-Einstellung (Auszug aus `sys_main.c`)

Um eine möglichst schnelle Übertragung zu gewährleisten, muss eine möglichst große Baudrate bei einem möglichst kleinen relativen Fehler gewählt werden. Tabelle 5.1 zeigt für ausgewählte Werte des Baud Rate Selection (BRS) Registers, die resultierende Baudrate und den relativen Fehler zur idealen, gewünschten Baudrate. Für die Ermittlung des relativen Fehlers wird Gleichung 58 aus [Tex13d, S. 1470] verwendet. Bei einer Taktfrequenz des Mikrocontrollers von 80 MHz und einer möglichst hohen Datenrate ergibt sich bei einer Baudrate 128000 ein relativer Fehler von 0,16%. Bei einer Symbolbreite von 8 Bit beträgt die Bitrate dann $1,024 \text{ MBit/s}$.

BRS-Wert	Baudrate ideal	Baudrate real	relativer Fehler
38	128000	128205	0,16 %
42	115200	116279	0,94 %
86	57600	57471	0,22 %

Tabelle 5.1.: Baudraten und resultierende relative Fehler

Abbildung 5.1 zeigt schematisch den Aufbau der wichtigsten Software-Komponenten und Funktionen für Node A. Die Software für Node B gleicht diesem Aufbau, besitzt jedoch keine Software-Module für SPI, SCI und ADC. Abbildung 5.2 zeigt zudem den Programmablauf für die Hauptprogramme auf Node A und B.

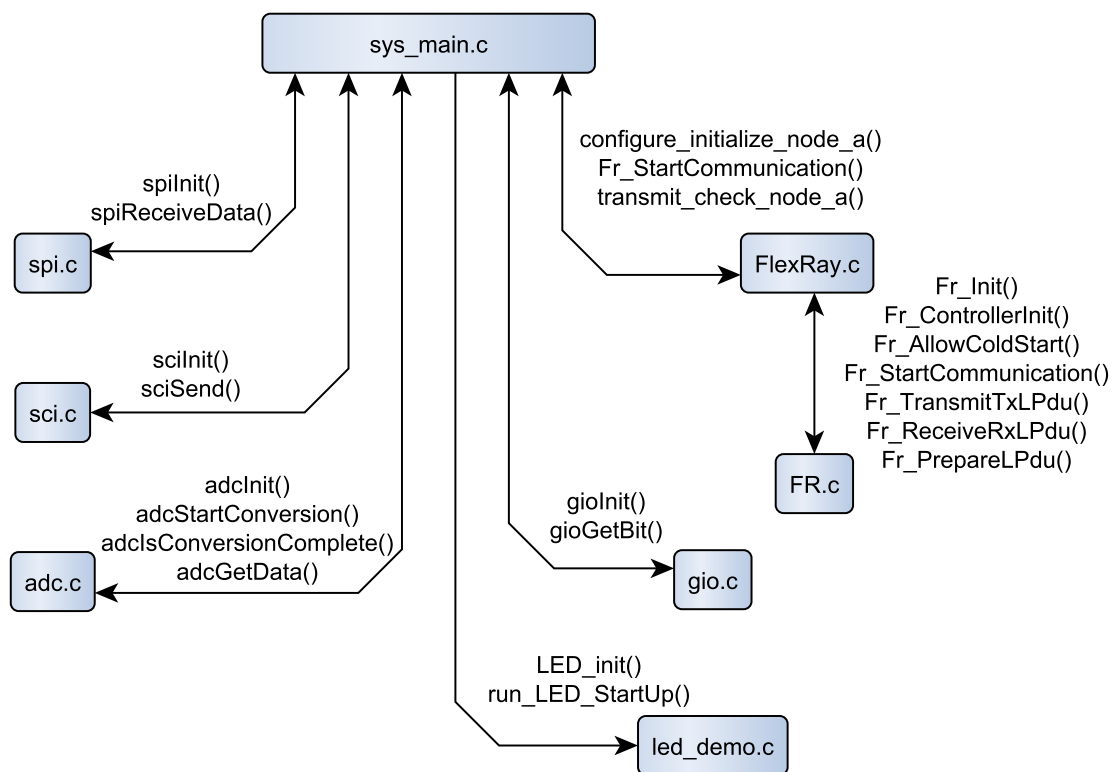


Abbildung 5.1.: Haupt-Programmkomponenten der Software für Node A

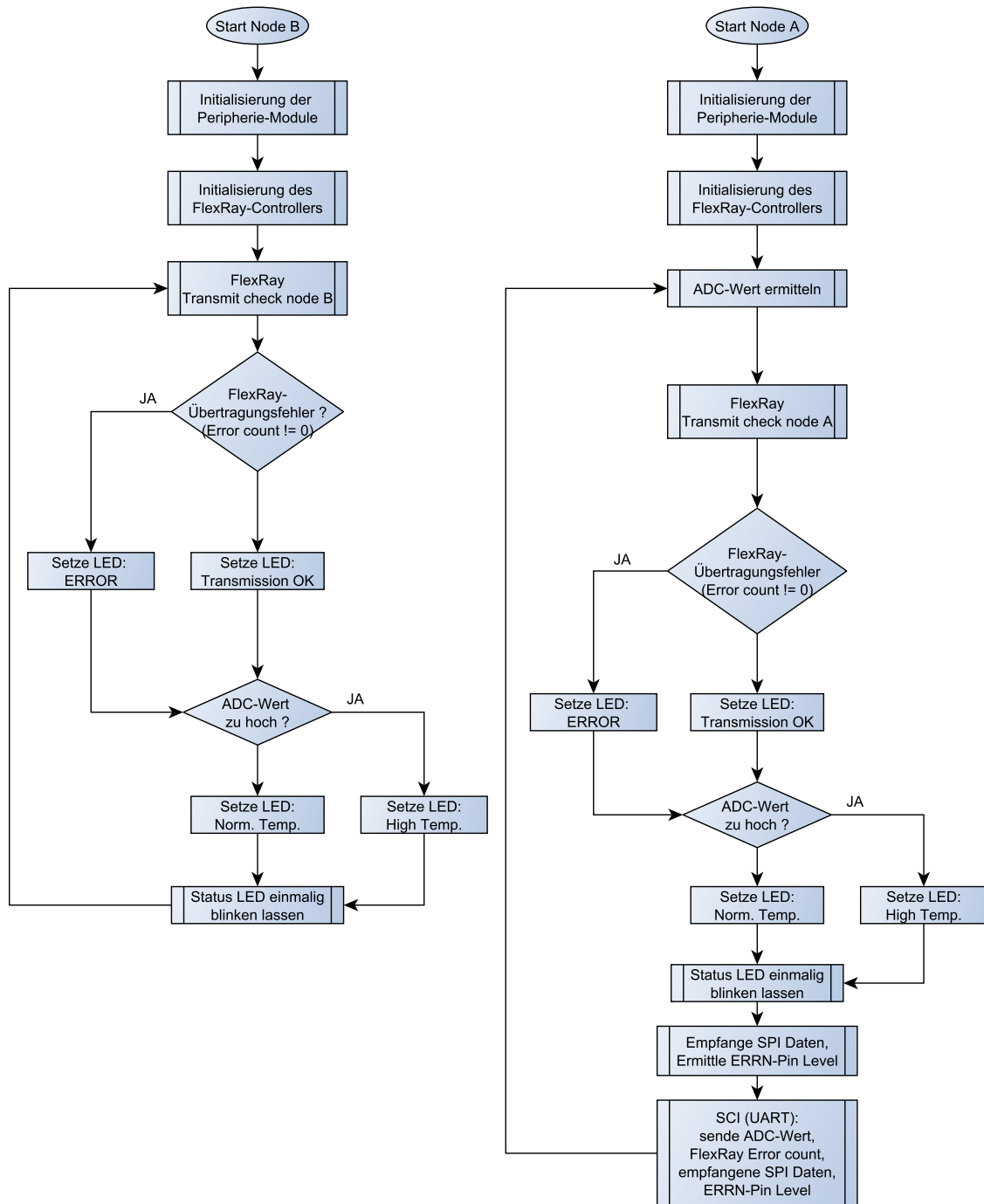


Abbildung 5.2.: Programmablauf der Hauptprogramme von Node A (rechts) und Node B (links)

5.2. Protokollzustandsmaschine des FlexRay-Controllers

Das Verhalten der internen Zustände eines FlexRay-Controllers wird über eine Protokollzustandsmaschine beschrieben. Die Protokollspezifikation bezeichnet diese Zustandsmaschine als Protocol Operational Control (POC). Abbildung 5.3 ist dem Benutzerhandbuch des E-Ray CC [Rob09] entnommen und zeigt das Zustandsdiagramm des Controllers, wobei nicht jede Implementierung zwangsweise jeden Zustand durchläuft. Daher wird im Folgenden nur auf die Zustände eingegangen, die in der realisierten Software durchlaufen werden.

Nach Einschalten der Spannungsversorgung geht der FlexRay-Controller zunächst in den Zustand `POC:DEFAULT_CONFIG`. Der Host-Controller ruft nun die Funktion `configure_initialize_node_a()` auf, in der die Konfiguration aller Parameter auf Host-Ebene erfolgt. Über die Funktion `Fr_ControllerInit()` bringt der Host den FlexRay-Controller nun in den Zustand `POC:CONFIG`, in dem die Konfigurations-Register des FlexRay-Controllers gesetzt werden. Ist dieser Vorgang abgeschlossen, wechselt der FlexRay-Controller in den Zustand `POC:READY`. Dieser Zustand ist ein Wartezustand, in dem der FlexRay-Controller weder Nachrichten senden noch empfangen kann, jedoch in der Lage ist, Wakeup-Signale zu erkennen. Für den Start eines Clusters ist mindestens ein sogenannter Kaltstartknoten (engl. cold start node) nötig. Die Software der Mikrocontroller gibt sowohl Teilnehmer A als auch Teilnehmer B die Berechtigung, einen Kaltstart einzuleiten. Dazu setzt der Host das Flag `CMD_ALLOW_COLDSTART` in `Fr_AllowColdStart()`. Die Funktion `Fr_StartCommunication()` bringt den Controller anschließend über das Flag `CMD_RUN` in den `POC:STARTUP`-Zustand. Der FlexRay-Controller beginnt nun mit dem Versuch, die Kommunikation im Cluster zu starten. Gelingt dies, wechselt der Controller in den Betriebszustand `POC:NORMAL_ACTIVE`, in dem die eigentliche Kommunikation, also der Austausch von Nutznachrichten stattfindet. Wird ein Teilnehmer vom Cluster abgekoppelt eingeschaltet, startet dieser eine definierte Anzahl von Versuchen (festgelegt in `gColdStartAttempts`), das Cluster zu starten. Sind die Versuche aufgebraucht, wird das interne Cycle Start Interrupt (CYCS)-Flag nicht gesetzt. Die Software bleibt an dieser Stelle in einer Abfrageschleife für dieses Flag hängen, der Controller muss dann neugestartet werden. [Rau07, S. 36 ff.], [Rob09]

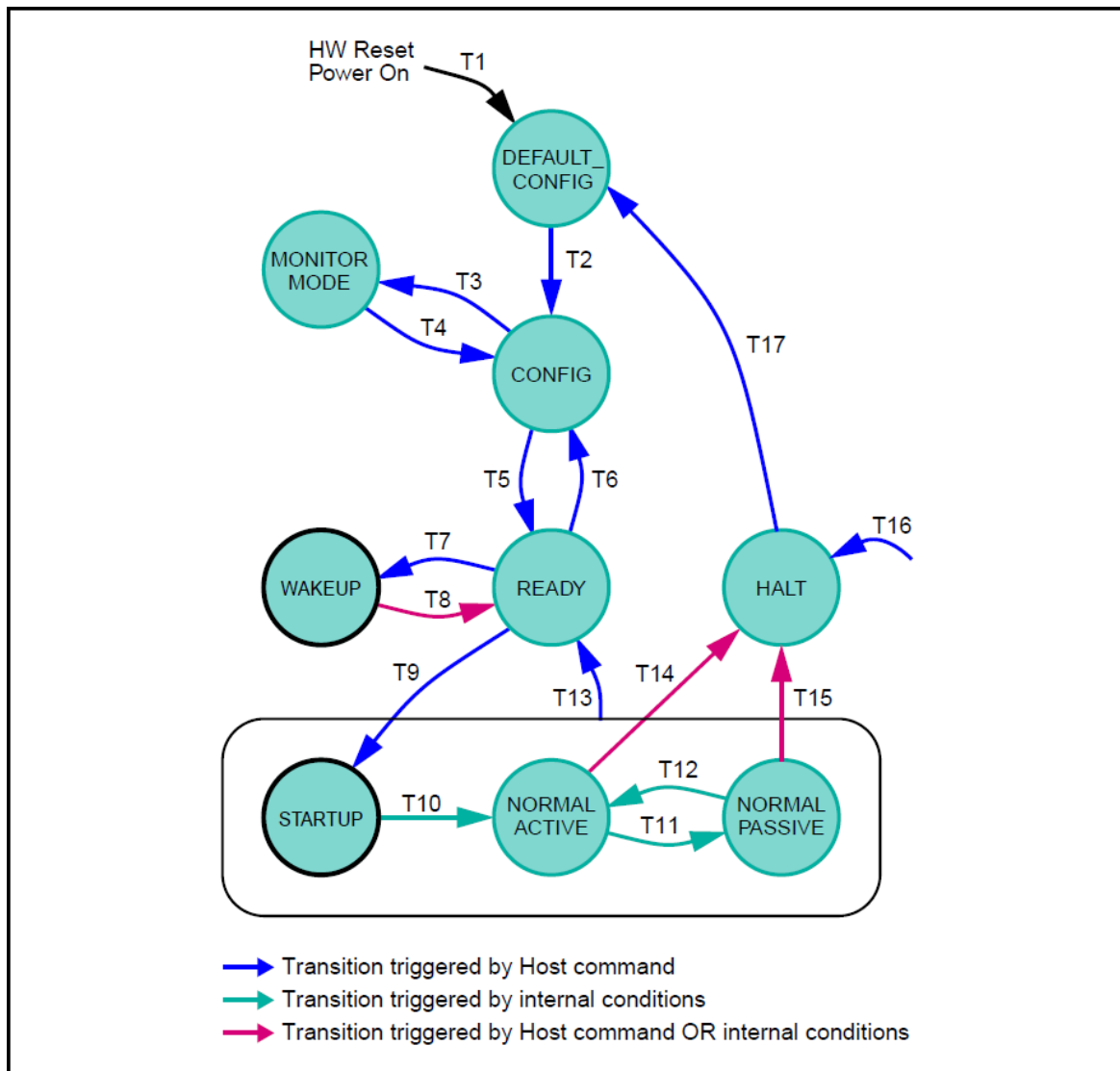


Abbildung 5.3.: Protokollzustandsdiagramm des E-Ray FlexRay-Controllers [Rob09, S. 107]

5.3. Konfiguration des FlexRay-Controllers

Die Konfiguration des FlexRay-Controllers richtet sich nach der Demo von Texas Instruments und bietet zunächst eine gute Unterstützung durch umfangreiche Kommentare. Allerdings fanden sich auch mehrere Fehler in den Kommentaren, beispielsweise errechnet ein Kommentar eine Zykluslänge von 5,6 μ s, anstelle der korrekten Zykluslänge von 5,6 ms. Obgleich die Parametrisierung mit dem E-Ray Benutzerhandbuch [Rob09] abgeglichen wurde, kann eine vollständige Korrektur der Kommentare nicht garantiert werden.

An dieser Stelle werden diejenigen Parameter vorgestellt, welche für die Verifikation des FlexRay-Timings verwendet werden. Eine vollständige Erläuterung aller Konfigurations-Register ist zu umfangreich und nicht zweckmäßig.

Um zunächst die kleinste Zeiteinheit (Microtick) einzustellen, wird der Baud Rate Prescaler auf 0x00 gesetzt [Rob09, S. 47]. Dies ist die Standardeinstellung für eine Bitrate von 10 MBit/s . Das Benutzerhandbuch des E-Ray-Controllers setzt für diese Einstellung eine Taktfrequenz von 80 MHz vorausgesetzt. Bei einer achtfachen Abtastung eines Bits, ist ein Sampletick `gdSampleClockPeriod` dann 12,5 ns lang. Die Anzahl der Samples pro Microtick `pSamplesPerMicrotick` in dieser Konfiguration ist zwei. Ein Microtick dauert dann $pSamplesPerMicrotick \cdot gdSampleClockPeriod = 25 \text{ ns}$.

Die Anzahl der Microticks pro Kommunikationszyklus wird im Global Time Unit 1 (GTU1)-Register festgelegt. In der hier verwendeten Konfiguration beinhaltet ein Kommunikationszyklus 224000 Microticks (`pMicroPerCycle`). Die gewählte Anzahl an Microticks muss mindestens 40 mal so groß sein wie die Anzahl an Macroticks pro Zyklus. In der hier verwendeten Konfiguration beträgt die Anzahl an Macroticks 5600 (`gMacroPerCycle`), also genau 40 mal die Anzahl an Microticks. Damit beträgt die Zeitdauer eines Macroticks das 40-zig fache eines Microticks, also $40 \cdot 25 \text{ ns} = 1 \mu\text{s}$. Bei 5600 Macroticks errechnet sich die Zeitdauer eines Kommunikationszyklus dann zu 5,6 ms

Für eine umfangreiche Dokumentation bezüglich der Konfiguration des FlexRay-Controllers sei auf das Benutzerhandbuch des E-Ray-Controllers verwiesen [Rob09]. Die Firma Bosch stellt zusätzlich weitere Application Notes für Entwickler zur Verfügung. Insbesondere die für die Konfiguration der Message Buffer sei hier erwähnt [Rob07].

6. Messergebnisse

Nachdem die vorangegangenen Kapitel die Hard- und Software der Evaluierungsplattform beschrieben haben, beinhaltet dieses Kapitel nun die Messergebnisse der hergestellten Prototypen. Zunächst werden dazu in Abschnitt 6.1 die Messergebnisse der Hardware-Komponenten erläutert. Darauf folgt die Verifizierung der Funktionalität der Software in Abschnitt 6.2. Am Ende des Kapitels wird noch eine zweite Revision des PCB-Designs vorgestellt, welche einige Optimierungen der realisierten Revision 1.5 bereitstellt.

6.1. Testmessung der Hardware-Komponenten

In diesem Abschnitt werden die hardwarenahen Messungen vorgestellt und ausgewertet. Für die Messungen werden die zwei Prototypen der Evaluierungsplattform über die FlexRay-Transceiver miteinander verbunden und in Betrieb genommen. Um die fehlerfreie Übertragung zu gewährleisten, muss die Impedanz der Kabel laut [Fle06b, S. 21] $80\ \Omega$ bis $110\ \Omega$ betragen. Daher werden für die Punkt-zu-Punkt Verbindung Unshielded Twisted Pair (UTP) Kabel des Typs AWG 24 (1/24) gewählt. Diese werden auch bei Cat5 Ethernet Kabeln eingesetzt und besitzen einen Wellenwiderstand von $100\ \Omega$ (Angabe bei einer Frequenz von 100 MHz und häufig mit Toleranzen von $\pm 10\ \Omega$ bis $\pm 15\ \Omega$ vgl. [Wik15h]). Für die hier verwendeten selbst verdrillten Kabel wird ebenfalls ein Wellenwiderstand von $100\ \Omega$ angenommen.

Für die Auswertung der SPI-Daten (vgl. Unterabschnitt 6.2.2) sowie den Nutzdaten der FlexRay-Datenkommunikation (vgl. Unterabschnitt 6.2.3), wird Teilnehmer eins (Node A) über USB zusätzlich mit einem PC verbunden. Die Bussignale der FlexRay-Übertragung werden über Tastköpfe mit dem PicoScope gemessen und aufgezeichnet. Das PicoScope ist ein mobiles USB-Oszilloskop und bietet eine serielle Dekodierung, unter anderem auch für FlexRay-Signale und ermöglicht dadurch eine komfortable Funktionsanalyse. Abbildung 6.1 zeigt den Messaufbau. Bei dem oberen (unbedruckten) Board handelt es sich um Node A, bei dem unteren (bedruckten) um Node B.



Abbildung 6.1.: Foto des Messaufbaus für die drahtgebundene FlexRay-Datenkommunikation

Die Tabelle 6.1 zeigt die bei der Vermessung der Hardware verwendeten Geräte und, sofern verwendet, den Verweis auf das Datenblatt des Geräts.

Bezeichnung	Typ	Datenblatt
Multimeter	Extech EX470	[Ext12]
Software-Oszilloskop	PicoScope 3406B	[Pic13]
Tastkopf	PicoScope TA131	-
Oszilloskop	Tektronix MSO 2024	-
Tastkopf	Tektronix TPP0201	-

Tabelle 6.1.: Übersicht der für die Messungen verwendeten Geräte

6.1.1. Verifikation der Spannungsversorgung

Für die Verifikation der Spannungsversorgung wird der Startvorgang der Abwärtsregler im Leerlauf aufgenommen. Die Messung erfolgt mit dem Oszilloskop Tektronix MSO2024. Der Abgriff über Tastköpfe erfolgt jeweils über das Via, welches dem Ausgang des jeweiligen Reglers am nächsten ist. Abbildung 6.2 zeigt das Messergebnis. Kanal 1 zeigt die 12 V Versorgungsspannung hinter der Schutzschaltung. Die Spannungsspitzen, auf allen drei Kanälen bei etwa $-80 \mu\text{s}$, entstehen durch das Einstecken des Netzsteckers. Neben diesen sind keine weiteren Spannungsspitzen zu beobachten, was auf die eingesetzten SMD-Ferrite und Stör- und Stützkondensatoren zurückzuführen ist. Kanal 3 zeigt den Spannungsverlauf des TPS54350 für das 5 V-Netz. Die Spannung steigt linear an und erreicht nach 2,57 ms ihren Endzustand. Falls eine Abhängigkeit von dieser Spannung besteht, muss diese Startzeit berücksichtigt werden. Dazu kann der Test Point 22 ausgewertet werden, welcher bei Erreichen der gewünschten Ausgangsspannung einen HIGH-Pegel liefert (vgl. Seite 36). Kanal 2 zeigt den Verlauf des 3,3 V Spannungsreglers. Die geforderte minimale Eingangsspannung errechnet sich aus der Summe der geforderten Ausgangsspannung V_{OUT} und der Drop-Out-Spannung V_{DO} . Die geforderte Ausgangsspannung liegt bei 3,3 V und die typische Drop-Out-Spannung bei 130 mV [Tex13a, S. 4]. Damit errechnet sich eine minimale Eingangsspannung von 3,43 V. In der Messung wird die gewünschte schon bei einer Eingangsspannung von 3,32 V erreicht, was auf den Betrieb im Leerlauf zurückzuführen ist.

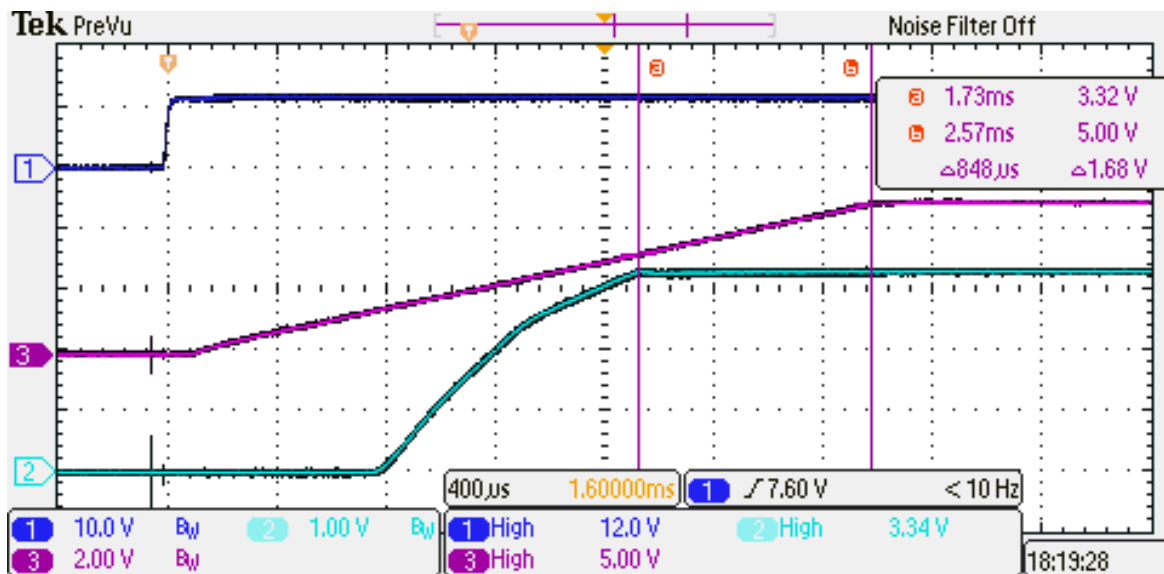


Abbildung 6.2.: Startvorgang der Spannungsversorgung für das 5 V und 3,3 V Spannungsnetz

Neben der Messung mit dem Oszilloskop wird eine weitere Spannungsmessung mit dem Multimeter EX470 durchgeführt. Tabelle 6.2 zeigt die gemessenen Spannungen bei Leerlauf und bei Betrieb der FlexRay-Datenkommunikation (Belastungsfall). In der rechten Spalte sind die Genauigkeiten der Regler laut Datenblatt festgehalten. Sowohl im Leerlauf als auch unter Belastung werden die Spannung ausreichend genau eingehalten. Die Spannungsversorgung der 5 V- und 3,3 V-Regler ist damit verifiziert.

Spannungsnetz	Messwert (Leerlauf / belastet)	Genauigkeit des Reglers
12 V	12,33 V / 12,16 V	-
5 V	5,01 V / 5,00 V	1 % [Tex04]
3,3 V	3,325 V / 3,324 V	1 % bis 3 % (belastet) [Tex13a]

Tabelle 6.2.: Messwerte Spannungen der Spannungsversorgung

Wie in Unterabschnitt 4.2.3 bereits beschrieben, benötigt das FPGA-Modul eine zusätzliche Spannungsversorgung für die Verwendung der externen I/O-Pins, dessen Aufbau von dem Schaltplan der Break-Out-Carrier Trägerkarte übernommen wurde. Das Benutzerhandbuch dieser Trägerkarte beschreibt den wesentlichen Startvorgang der Spannungsversorgung nach dem Einschalten (vgl. [Avn14b, S. 12 ff.]). Die Abbildung 6.3 zeigt die schematische Darstellung dieses Startvorgangs, mit dem die folgende Messung verglichen wird. Zuerst wird die Versorgungsspannung V_{in} (auf der Evaluierungsplattform als 5V_ZED bezeichnet) über den Schalter S_1 eingeschaltet. Über die Diode D_4 wird das Signal PWR_EN gesetzt. In der schematischen Darstellung wird eine relativ große Verzögerung zwischen V_{in} und PWR_EN eingezeichnet. In der Messung fällt diese Verzögerung nicht auf, was laut Benutzerhandbuch nicht unüblich ist. Nachdem die interne 1,8 V Spannungsversorgung auf dem MicroZed ihren Endzustand erreicht hat, wird das Signal VCCIO_EN gesetzt, welches den einstellbaren Spannungsregler LMR10510 der Evaluierungsplattform aktiviert. Sobald sowohl der LMR10510, als auch die weiteren internen Spannungsregler des MicroZeds eine stabile Ausgangsspannung liefern, wird über das Signal PG_Carrier (auf der Evaluierungsplattform als PG_LED bezeichnet) die Betriebsbereitschaft signalisiert. Das MicroZed beginnt dann mit dem eigentlichen Bootvorgang (vgl. Signal PG_Module in Abbildung 4.5).

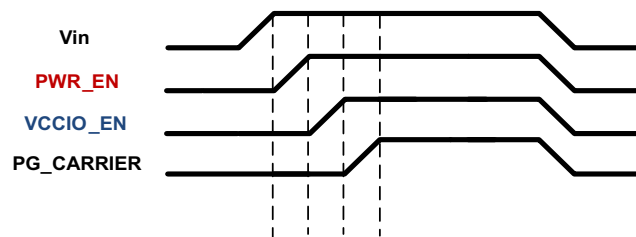


Abbildung 6.3.: Schematische Darstellung des zeitlichen Ablaufs der MicroZed-Spannungsversorgung während des Startvorgangs [Avn14b, S. 13]

Die realisierte Spannungsversorgung auf der Evaluierungsplattform wird mit dem Oszilloskop vermessen, indem die zu betrachtenden Signale 5V_ZED (Kanal 1), PWR_EN (Kanal 2), VCCIO_EN (Kanal 3) und PG_LED (Kanal 4) über Vias abgegriffen werden. Die Abbildung 6.4 zeigt den zeitlichen Verlauf der gemessenen Signale, welche im Wesentlichen dem zu erwartenden Verlauf entsprechen. Die Spannungseinbrüche sind dabei auf interne Schaltvorgänge der internen Spannungsversorgung des MicroZeds zurückzuführen.

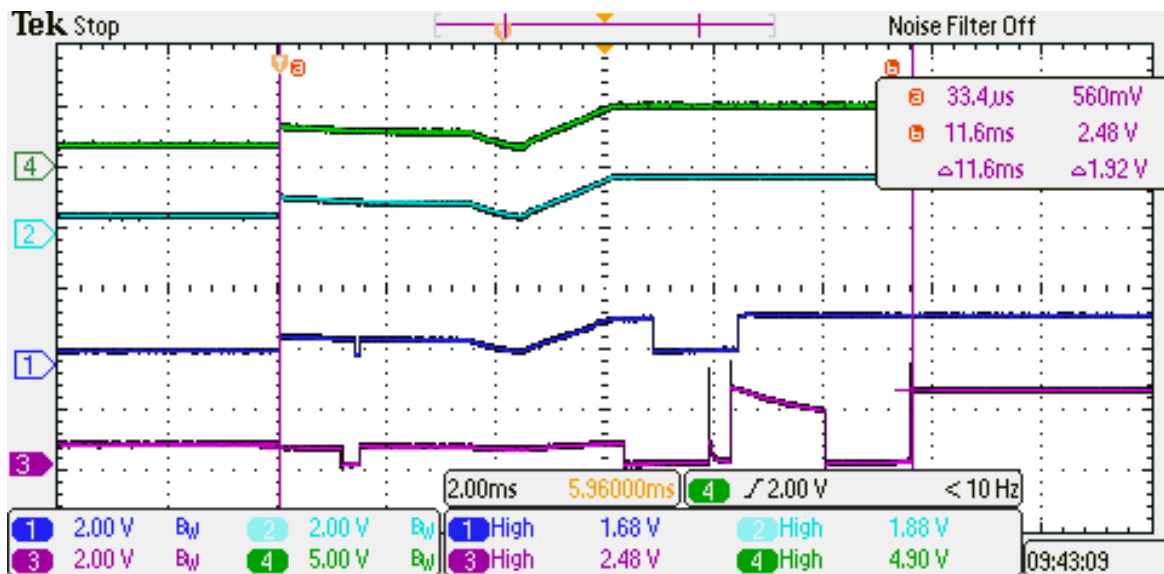


Abbildung 6.4.: Startvorgang der Spannungsversorgung für das MicroZed

6.1.2. Stromaufnahme der Evaluierungsplattform

Nachdem in Abschnitt 4.1 die Stromaufnahme der Hardware-Komponenten abgeschätzt wird, folgt in diesem Unterabschnitt der Vergleich der abgeschätzten Werte mit den gemessenen. Gemessen wird die Stromaufnahme am Eingang der jeweiligen Platine, bei einer Eingangsspannung von etwa 12 V. Wie unter Abschnitt 6.1 beschrieben, handelt es sich bei Node A um die unbedruckte, bei Node B um die bedruckte Platine.

Die Tabelle 6.3 zeigt die gemessene Stromaufnahme von Node A und Node B für verschiedene Betriebsarten. Die erste Zeile, „Nur FRJB (S_1 OFF)“ zeigt die Stromaufnahme der Platine ohne Mikrocontroller- und FPGA-Modul. Die einstellbare Spannungsversorgung ist ausgeschaltet, S_1 also auf Position OFF. Gemessen wird jeweils ein Strom von 18,4 mA auf der Platine für Node A und 17,8 mA auf der Platine für Node B. Die Stromaufnahme setzt sich aus dem Betriebsstrom der LEDs, sowie aus Verlustströmen zusammen. Wird die einstellbare Spannungsversorgung eingeschaltet (S_1 auf ON), kommt der Verluststrom dieser Schaltung hinzu.

Die Stromaufnahme in Zeile drei ist der jeweiligen HDK-Platine im stand-alone Betrieb zuzuordnen. Node A weist eine etwas höhere Stromaufnahme auf, da hier mehr Peripherie-Module aktiv sind als bei Node A (insbesondere der ADC und das SCI). Die Stromaufnahme von Node A beträgt maximal 166,3 mA und liegt damit höher als die laut Datenblatt typische Stromaufnahme von 130 mA bei einer Taktfrequenz des Mikrocontrollers von 180 MHz

[Tex13c, S. 7]. Dabei ist zu beachten, dass die typische Stromaufnahme nicht dem Maximalstrom entspricht und eine andere Taktfrequenz (160 MHz) eingestellt wird. Daher ist die gemessene Stromaufnahme plausibel.

Da die Messung über die Eingangsspannung keine genauen Werte für die Stromaufnahme in weiteren Boardspannungsnetzen liefert, kann die Stromaufnahme der FlexRay-Transceiver und des FPGA-Moduls nur durch Überschlagsrechnungen ermittelt werden. Werden die Verlustströme auf der Platine, sowie die Stromaufnahme des HDK abgezogen, errechnet sich die Stromaufnahme der FlexRay-Transceiver im 12 V-Netz zu $(200,1 \text{ mA} - 166,3 \text{ mA} - 18,4 \text{ mA}) = 15,4 \text{ mA}$. Bei einem Wirkungsgrad des 5 V-Reglers von 80% errechnet sich dann die geschätzte Stromaufnahme der FlexRay-Transceiver zu 46,2 mA. (vgl. Gleichung 4.2 bzw. Unterabschnitt 4.2.2). Dieser Wert liegt unter der angenommenen Stromaufnahme von 70 mA im worst-case (vgl. Tabelle 4.1).

Nach gleichem Vorgehen wird die Stromaufnahme des FPGA-Moduls berechnet. Die Stromaufnahme im 12 V-Netz beläuft sich hier auf $(380,0 \text{ mA} - 166,3 \text{ mA} - 24,5 \text{ mA} - 15,4 \text{ mA}) = 173,8 \text{ mA}$. Bei einem angenommenen Wirkungsgrad von abermals 80% errechnet sich die Stromaufnahme des FPGA-Moduls im 5 V-Netz dann zu 521,4 mA. Damit liegt der Wert bei etwa einem Viertel der abgeschätzten Stromaufnahme von 2,18 A im worst-case (vgl. Tabelle 4.1). Dabei ist zu beachten, dass sich die Funktion des FPGA hier nur auf ein einfaches Gatter beschränkt (vgl. Unterabschnitt 6.1.3). Der Wert ist also plausibel.

Insgesamt zeigt die Messung, dass die jeweilige Stromaufnahme den abgeschätzten Wert für den worst-case nicht überschreitet und die Spannungsversorgung damit den Anforderungen gerecht wird.

Betriebsart	Eingangsstrom Node A	Eingangsstrom Node B
Nur FRJB (S_1 OFF)	18,4 mA	17,8 mA
Nur FRJB (S_1 ON)	25,2 mA	24,5 mA
Nur das jeweilige TI HDK	166,3 mA	159,0 mA
FRJB + TI HDK	200,1 mA	185,5 mA
FRJB + TI HDK + MicroZed	380,0 mA	357,2 mA

Tabelle 6.3.: Maximale Stromaufnahme der Hardware in verschiedenen Betriebsarten

6.1.3. Laufzeitmessung der FlexRay-Signale über das FPGA-Modul

Um die Kommunikation zwischen Mikrocontroller und FPGA zu testen, wird die Laufzeitdifferenz zwischen den Signalen FRAY_TX1 und MOD_DQ_1 gemessen. Die Funktion des FPGA beschränkt sich hier auf ein einfaches Gatter, welches das Signal durchschleift. Abbildung 6.5 zeigt das Messergebnis, aufgenommen mit dem Tektronix MSO 2024 Oszilloskop. Dabei ist die Bandbreitenbegrenzung oberhalb von 20 MHz eingeschaltet, um die Harmonischen des Rechtecksignals zu unterdrücken.

Kanal eins (Signalverlauf in blau) zeigt den Verlauf des Signals FRAY_TX1, abgegriffen am Via zwischen den Widerständen R_{12} und R_{44} . Kanal zwei (Signalverlauf in türkis) zeigt den Verlauf des Signals MOD_DQ_1, abgegriffen an Pin 98 von Stecker J_3 . Gemessen wird eine Differenz von 4,8 ns. Da bei dieser Messung das FPGA-Modul prinzipiell den FlexRay Transceiver in der Signalkette ersetzt, kann die Laufzeitdifferenz mit der des Transceivers verglichen werden. Das Datenblatt der TJA1082 Transceiver [NXP12, S. 23] gibt eine Signalverzögerung zwischen dem Eingang TXD und Ausgang BP bzw. BM des Transceivers von maximal 50 ns an. Die Verzögerung durch das FPGA-Modul ist wesentlich geringer, jedoch sollte berücksichtigt werden, dass das Signal lediglich unverändert durchgereicht wird und eine weitere Signalverarbeitung wesentlich aufwendiger ist. Die Verzögerungszeit muss in jedem Fall eingeplant werden, um die strenge Zeithierarchie einzuhalten.

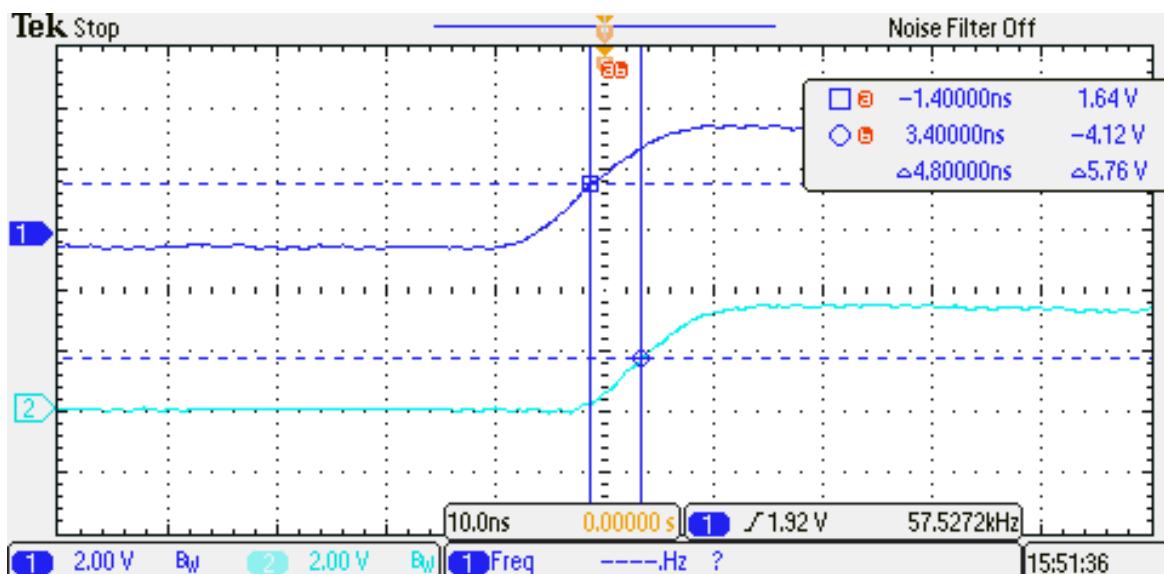


Abbildung 6.5.: Messung der Laufzeitdifferenz zwischen Signals FRAY_TX1 und MOD_DGQ_1

6.2. Verifizierung der Software-Funktionalität

In diesem Abschnitt werden die Untersuchungen zur Verifikation der Software vorgestellt. Um die Einhaltung der strengen Zeithierarchie des FlexRay-Protokolls zu gewährleisten, wird zunächst in Unterabschnitt 6.2.1 die Zeitdauer des Kommunikationszyklus gemessen und mit dem erwarteten Wert verglichen. Unterabschnitt 6.2.2 beschreibt die Auswertung der empfangenen Status-Daten der FlexRay-Transceiver. Im Anschluss wird in Unterabschnitt 6.2.3 die Verifikation der drahtgebundenen FlexRay-Datenkommunikation erläutert.

6.2.1. Verifikation der Zeitdauer eines Kommunikationszyklus

Da es sich bei FlexRay um ein deterministisches Bussystem handelt, ist die strikte Einhaltung von definierten Zeiteinheiten von großer Bedeutung. Die Messungen in diesem Unterabschnitt verifizieren daher die Dauer eines Kommunikationszyklus, welche durch die Parametrisierung laut Abschnitt 5.3 eingestellt ist. Zunächst wird das Zeitverhalten direkt hinter dem FlexRay-Controller mit einem Oszilloskop gemessen. Die Daten liegen hier unkodiert vor und können daher einfacher interpretiert werden. Die Tastköpfe werden dazu mit den Signalen FRAY_TX1 und FRAY_TX2 an J₁₁ der HDK-Platine verbunden. Das Mikrocontroller-Modul wird dazu von der Evaluierungsplattform getrennt. Abbildung 6.6 zeigt den Signalverlauf der FlexRay-Zyklen eins und zwei. Signalverlauf eins (blau) zeigt jeweils den Kanal eins (bzw. Kanal A), der Signalverlauf zwei analog dazu den Kanal zwei (bzw. Kanal B). Die Cursor des Oszilloskops begrenzen jeweils den 6-Bit Cycle Count des FlexRay-Frames.

Die obere Messung zeigt die Bitfolge $000001_2 = 1_{10}$ und kennzeichnet damit Zyklus eins. Die untere Messung zeigt die Bitfolge $000010_2 = 2_{10}$, kennzeichnet damit wie zu erwarten den Zyklus zwei. Aus der Zeitdifferenz der Triggerzeitpunkte in den beiden Messbildern kann eine Zykluslänge von $11,2936 \text{ ms} - 5,69406 \text{ ms} = 5,59954 \text{ ms}$ ermittelt werden. Dieser Wert entspricht in etwa der erwarteten Zyklusdauer von 5,6 ms. Die absolute Abweichung beträgt 460 ns und ist auf das verwendete Uhrenkorrekturverfahren der FlexRay-Controller zurückzuführen. Zudem muss eine gewisse Messungenauigkeit des Messaufbaus und die Ungenauigkeit des lokalen Quarzoszillators berücksichtigt werden. Der verwendete Quarzoszillator ASEM1-25.000MHZ-LCT weist eine Ungenauigkeit von 50 ppm auf [ABR13] und verursacht bei einer Zyklusdauer von 5,6 ms somit eine Abweichung von bis zu 280 ns.

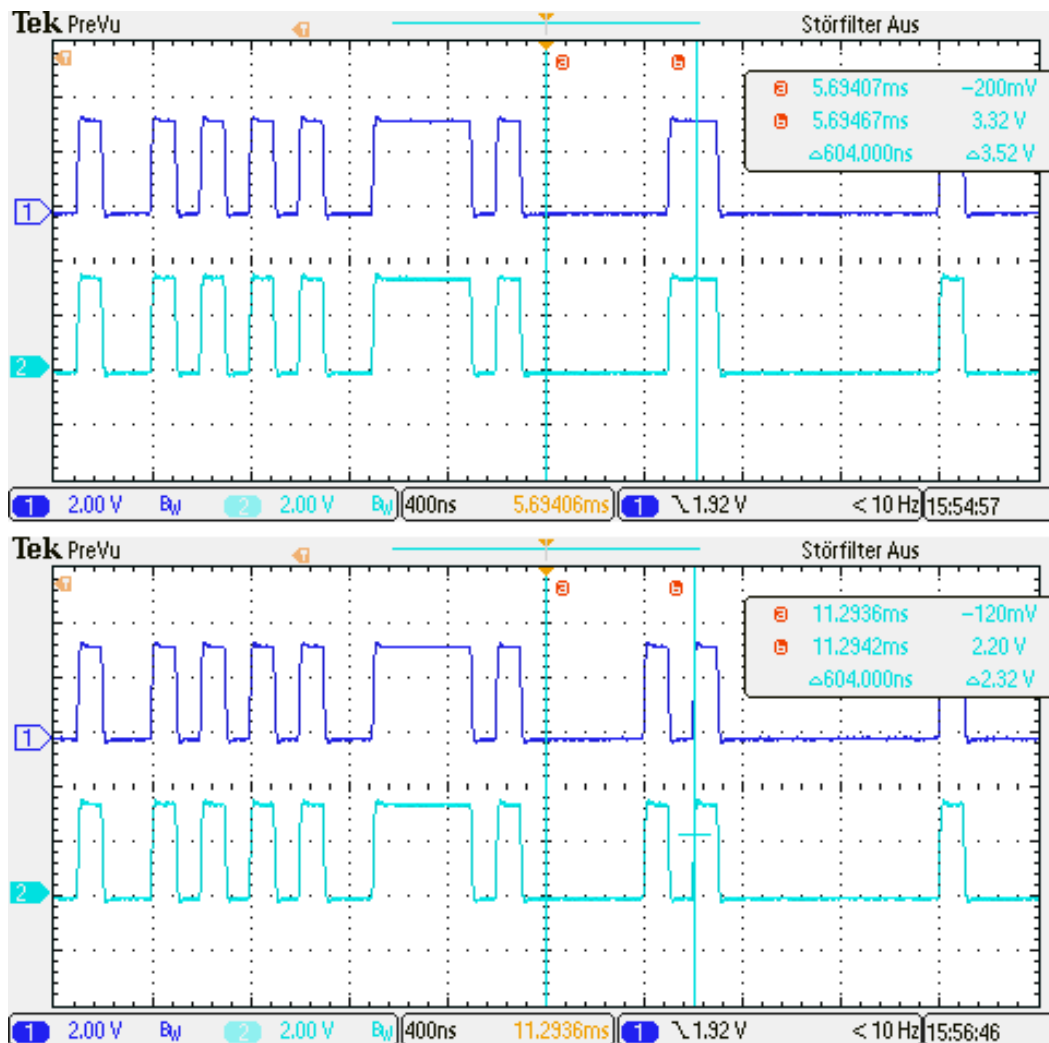


Abbildung 6.6.: Messung der ersten beiden Cycle Counts

Eine weitere Messung verifiziert das Timing der drahtgebundenen FlexRay-Datenkommunikation. Dazu wird der Buspegel BP während der Datenkommunikation gemessen (vgl. Messaufbau Abbildung 6.1). Die Abbildung A.11 zeigt einen vollständigen Zyklus der FlexRay-Datenkommunikation, aufgenommen mit dem Software Oszilloskop PicoScope. Betrachtet wird hier die Zykluslänge bezogen auf Kanal A. Der Zyklus startet vom Trigger (gelbe Raute) und endet bei dem ersten Frame auf der rechten Bildschirmseite. Auf der Zeitachse ist für den Zyklus dann eine Dauer von 5,594 ms abzulesen.

6.2.2. Verifikation der SPI-Schnittstelle

Für die Verifikation der SPI-Schnittstelle wird zunächst das Mikrocontroller-Modul über USB mit dem PC verbunden. Über die serielle Schnittstelle kann dann der Datenaustausch über ein Terminal-Programm überwacht werden. Die Abbildung 6.7 zeigt einen Screenshot des verwendeten Terminal-Programms HTerm [Tob08]. Zeile eins zeigt den Zähler für eine registrierte fehlerhafte FlexRay-Kommunikationen. Zeile zwei zeigt den gemessenen Wert des ADC. Unterabschnitt 6.2.3 geht näher auf die Auswertung dieser Daten ein. Die folgenden Zeilen zeigen die empfangenen SPI-Daten der FlexRay-Transceiver für Kanal A (SPI2 NCS[0] data) sowie für Kanal B (SPI2 NCS[1] data). Zusätzlich wird in den letzten beiden Zeilen der Pegel des ERRN-Pins der beiden Transceiver angezeigt.

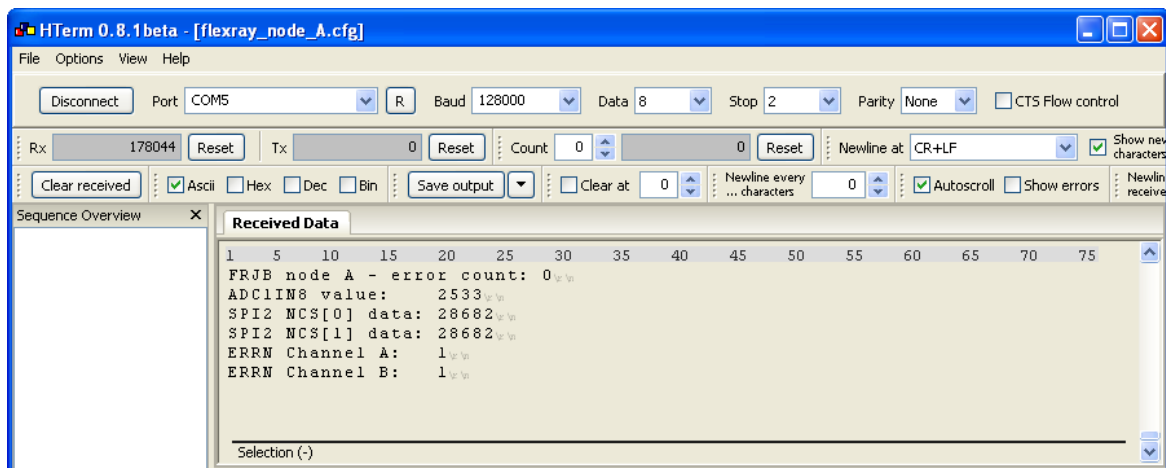


Abbildung 6.7.: Screenshot des Terminalprogramms für die seriellen Kommunikation zwischen Mikrocontroller und Rechner

Der ERRN-Pin ist low-aktiv, der Pin wird also auf Masse gezogen sobald ein Fehlerfall eintritt. In Abbildung 6.7 zeigt der ERRN-Pin eine eins und signalisiert somit den fehlerfreien Betrieb der Transceiver. Eine weitere Zustandsanalyse erfolgt durch die Statusbits, welche über das SPI ausgewertet werden. Beide FlexRay Transceiver senden im Normalbetrieb den Wert 28682_{10} , im Binärsystem also $0111\ 0000\ 0000\ 1010_2$. Tabelle 6.4 gibt Auskunft über die Bedeutung der Nachricht. Statusbit S11 bis S14 sind für eine Kennungsfolge reserviert und sind korrekt am Ende der Binärfolge abzulesen. Statusbit S1 ist gesetzt und zeigt Normalbetrieb an. Werden die Transceiver mit einem HIGH-Potential am BGE-Pin aktiviert, wird wie im hier gemessenen Fall Statusbit S2 gesetzt, um diese Aktivierung anzuzeigen. Das letzte Bit S15 wird bei gerader Parität gesetzt, ist in diesem Fall also nicht gesetzt, da eine ungerade Parität der Statusbits empfangen wird.

Als weiterer Test für die Funktion der SPI wird der Widerstand R_{10} entfernt. Dieser zieht den BGE-Pin auf HIGH-Potential und aktiviert so den Transceiver (vgl. Abschnitt 4.3). Der Transceiver ist in diesem also deaktiviert. Ohne den Widerstand meldet die SPI einen Wert von 16394_{10} , also die Binärfolge $0100\ 0000\ 0000\ 1010_2$. Die Statusbits unterscheiden sich von der vorherigen Nachricht nur um die Bits S2 und S3, welche nicht gesetzt sind. Das bedeutet, dass sich der Transceiver zwar im normal mode befindet, jedoch wie zu erwarten inaktiv ist.

Status bit	Flag name	Set condition
S0	bus wake	wake-up event detected
S1	normal mode	entering normal mode
S2	transmitter enabled	transmitter enabled
S3	BGE clamped	BGE clamped HIGH
S4	PWON	Power-up
S5	bus error	bus error detected
S6	TEMP HIGH	temperature limit exceeded
S7	TXEN clamped	TXEN clamped LOW
S8	UVV_{CC}	under-voltage V_{CC}
S9	UVV_{IO}	under-voltage V_{IO}
S10	SPI error	SPI error detected
S11	reserved	always LOW
S12	reserved	always HIGH
S13	reserved	always LOW
S14	reserved	always HIGH
S15	parity bit	odd parity of status bits

Tabelle 6.4.: Übersicht des Statusregisters der Node Transceiver [NXP10, S.22]

6.2.3. Nutzdatenauswertung der FlexRay-Kommunikation

Um die Datenkommunikation zwischen den FlexRay-Knoten zu verifizieren, wird auf Knoten A die Temperatur gemessen und an Knoten B gesendet. Die Messung der Temperatur ist ein häufiger Anwendungsfall im Automobil und daher praxisnah. Zur Messung der Temperatur kommt ein Heißeleiter zum Einsatz. Dieser befindet sich bereits auf dem TI HDK und wird über den 12-Bit ADC ausgewertet. Abbildung 6.8 zeigt Beschaltung des Heißeleiters (R_{148}). Zu beachten ist der Jumper J_8 . Dieser muss gebrückt werden um die Referenzspannung des ADC auf die benötigten 3,3V einzustellen, da die Versorgung des Spannungsteilers ebenfalls mit 3,3V erfolgt. Lässt man den Jumper offen, wird eine Referenzspannung von 5V eingestellt und müsste in der Berechnung berücksichtigt werden.

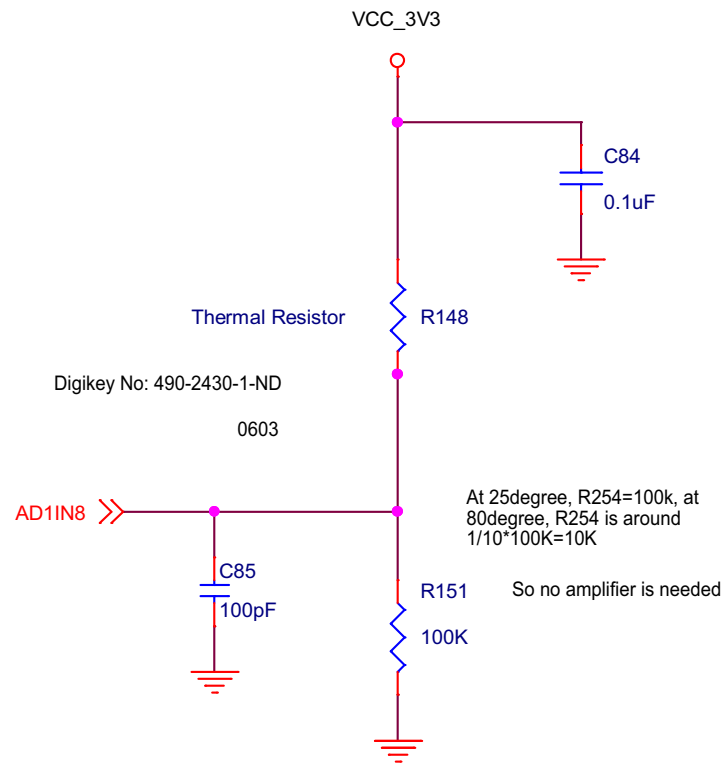


Abbildung 6.8.: Beschaltung des Heibleiters auf dem TI Hercules Development Kit [Tex12, S.14]

Der resultierende ADC-Wert lsst sich durch das Verhltnis der Spannung am ADC-Eingang und der Referenzspannung berechnen.

$$ADC = \frac{U_{ADC}}{U_{ref}} \cdot 2^N = \frac{R}{R + R_{NTC}} \cdot 2^N \quad (6.1)$$

Fr die weitere Berechnung wird nach R_{NTC} umgestellt.

$$R_{NTC} = R \cdot \left(\frac{2^N}{ADC} - 1 \right) \quad (6.2)$$

Mit $R_{NTC} = R_{148}$ und $R = R_{151}$ lsst sich dann der Widerstand des Heibleiters bestimmen.

Nherungsweise kann die resultierende Betriebstemperatur T_t durch die B-Parameter-Gleichung (engl. B parameter equation) berechnen werden [Wik15f].

$$\{T_t\}^{\circ C} = \frac{1}{\frac{1}{\{T_0\}_K} + \frac{1}{B} \cdot \ln\left(\frac{R_{NTC}}{R_0}\right)} - 273,15 \quad (6.3)$$

Der Parameter B wird dem Datenblatt des Herstellers entnommen [Mur14, S. 13]. Für den Messbereich 25°C bis 80°C gilt $B = 4303$. Bei einer Raumtemperatur von $T_0 = 298,15$ K gilt für den Widerstandswert des Heißleiters $R_0 = 100$ k Ω . Für die Messung wird der Knoten A etwa zwei Stunden im Standard-Betrieb belassen (FlexRay-Kommunikation zwischen Knoten A und Knoten B, FPGA-Aufsteckplatine eingeschaltet im Ruhemodus). Es wird ein ADC-Wert von 2533 ermittelt (vgl. Abbildung 6.7). Nach Gleichung 6.3 wird demnach eine Temperatur von $T_t(2533) \approx 35,3$ °C berechnet. Zur Überprüfung wird mit dem Multimeter Extech EX470 per Infrarot eine Temperatur von 34°C gemessen. Dabei ist zu beachten, dass der Heißleiter direkt auf der 8-Layer Platine platziert ist. Diese leitet die Wärme wesentlich besser als die Luftstrecke zwischen der Platine und dem Multimeter. Bei einer Genauigkeit von ± 2 °C im Messbereich über 25°C [Ext12, S. 13] ist dieser Wert also plausibel und zufriedenstellend.

Abbildung 6.9 zeigt den Ausschnitt einer FlexRay-Datenkommunikation. Aufgenommen ist diese mit dem PicoScope (vgl. Tabelle 6.1). Dieses unterstützt eine serielle Dekodierung für FlexRay und andere Busprotokolle und bietet dadurch ausgezeichnete Debugging-Eigenschaften für ein Bussystem. Eine zuverlässige Dekodierung konnte erreicht werden, indem die Tastköpfe an den Signalleitungen BP hinter den Filter-Induktivitäten platziert wurden (Schraubklemmenanschluss nahe dem Bezeichner X_1 bzw. X_2). Der obere Signalverlauf (rot) zeigt den FlexRay Kanal A, der untere Signalverlauf (blau) zeigt Kanal B. Frame eins und zwei sind, übereinstimmend mit der Parametrisierung des FlexRay-Controllers, auf beiden Kanälen zu sehen. Auf Kanal A werden zusätzlich noch die dynamischen Nachrichten in Frame neun (im Signalverlauf zu erkennen als 0x009) und Frame zehn (0x00A) übertragen.

Über den Parameter `gPayloadLengthStatic` in der Datei `FlexRay.c` wird die Länge einer Payload im Static-Segment definiert. In der verwendeten Konfiguration beinhaltet eine Payload eines Static-Frames 9 Wörter zu je 16-Bit. Das bedeutet das 18 Byte übertragen werden. In der Abbildung wird wie erwartet eine Payload von 18 Byte angezeigt (Darstellung im Hexadezimalsystem). Dabei erfolgt die Übertragung byteweise und im Big-Endian-Format, das Most Significant Bit (MSB) wird also zuerst übermittelt. Nachrichten mit unvollständig gefülltem Message Buffer werden mit Nullen aufgefüllt, welche hier nach den ersten drei Wörtern, also den ersten 6 Bytes bzw. 12 Nibbles zu erkennen sind. Im dritten 16-Bit-Wort wird der Wert des ADC übermittelt. Dieser beträgt hier $09F2_{16} = 2546_{10}$, ähnelt also dem oben genannten Messwert von 2533_{10} . Die ADC-Werte sind nicht absolut gleich, da die Messungen zu leicht versetzten Zeitpunkten aufgenommen wurden. Die drahtgebundene Daten-Kommunikation über FlexRay ist damit verifiziert.

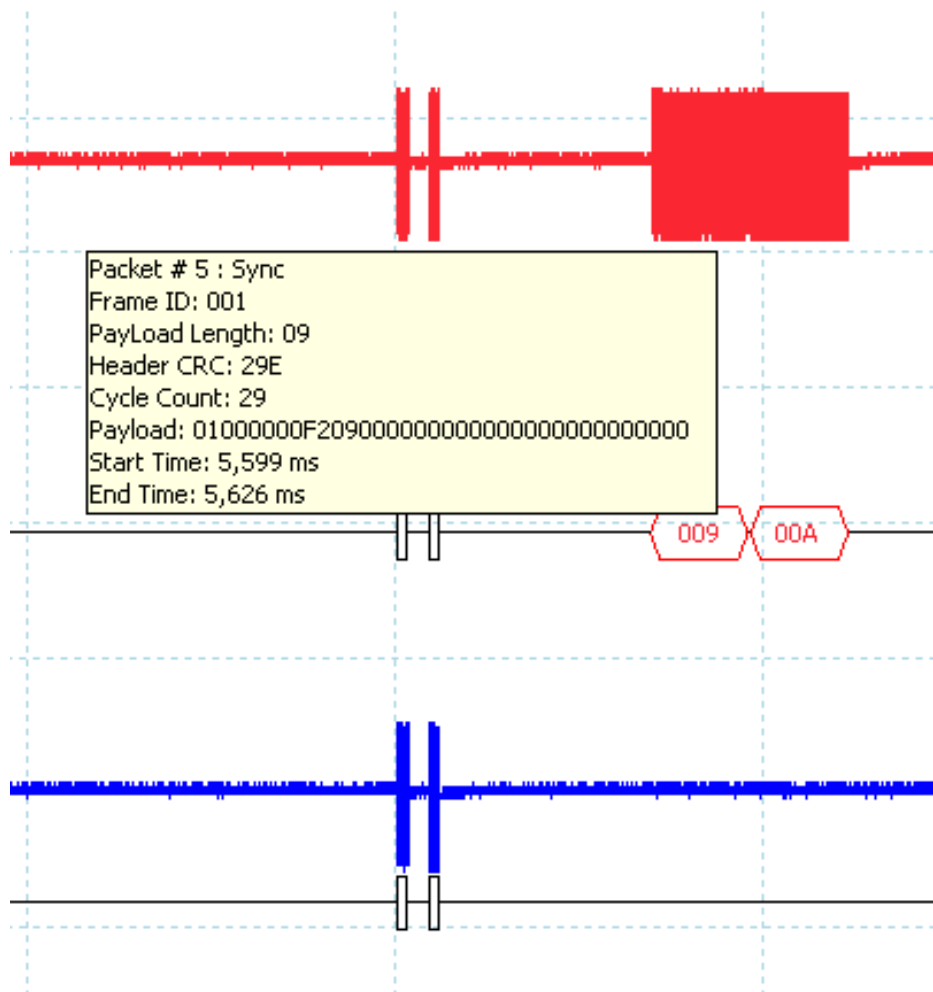


Abbildung 6.9.: Ausschnitt der PicoScope-Messung einer vollständigen FlexRay-Datenkommunikation (vgl. Abbildung A.11)

6.3. Optimierung der Evaluierungsplattform

Gegen Ende der Arbeit wurden einige Optimierungsmöglichkeiten der Evaluierungsplattform deutlich. Daher wird eine zweite Revision des PCB-Layouts erstellt, welche diese Optimierungsansätze aufgreift. Für eine generelle Signalisierung von Zuständen auf dem FPGA-Modul bietet Revision 2 des FRJB zwei frei verwendbare LEDs, die über den Stecker JX2 der FPGA-Aufsteckplatine an Pin 41 bzw. 43 angesteuert werden können. Dadurch lassen sich Betriebszustände oder ähnliche grundlegende Informationen schnell über die LEDs anzeigen. Weiterhin werden zwei weitere Pins (NHET1[15] und NHET1[16]) des TMS570LS3137 mit dem FPGA verbunden (vgl. Tabelle A.1) und bieten dadurch erweiterte Kommunikationsmöglichkeiten zwischen den Modulen. Neben diesen funktionellen Erweiterungen sind zudem einige Bauteile und Schriftzüge der Übersichtlichkeit bzw. Lesbarkeit halber versetzt.

Obgleich keine signifikante Hitzeentwicklung des Schaltreglers TPS54350 (Bauteilname U₁) festgestellt werden konnte, werden zwei weitere Vias unter dem Gehäuse des Bauteils platziert. Damit wird eine bessere Wärmeableitung der Verlustwärme des Reglers erreicht, um bei einer höheren Belastung der Spannungsversorgung einen sicheren Betrieb der Evaluierungsplattform zu gewährleisten. Die Layout-Dateien der EAGLE-Projekte liegen sowohl für die Revision 1.5, sowie Revision 2.0 auf der CD bei und bieten eine Grundlage für weitere Optimierungen.

7. Zusammenfassung und Ausblick

Im Rahmen dieser Arbeit wurde eine Evaluierungsplattform für drahtlose und drahtgebundene FlexRay-Datenkommunikation entwickelt. Ziel war es, eine Testumgebung für die Untersuchung möglicher Lösungen zur Umstellung des Physical Layers von Bussystemen zu schaffen. Dazu wurde zunächst die Anforderung an die benötigten Hardware-Komponenten erläutert und anschließend die realisierte Elektronik vorgestellt. Die Messergebnisse zeigen, dass sowohl die entwickelte Hardware als auch die Software die an die Plattform gestellten Anforderungen erfüllt (vgl. Seite 25). Die Platine stellt alle geforderten Peripherie-Module sowie eine leistungsstarke Spannungsversorgung bereit.

Die drahtgebundene FlexRay-Kommunikation mit zwei Kanälen wurde erfolgreich bei der maximalen Bitrate von 10 Mbit/s verifiziert. Dazu übermittelt ein Teilnehmer Nutzdaten, in Form von Temperaturmessdaten an einen zweiten Teilnehmer, der ggf. eine Warnung bei hoher Temperatur ausgibt. Die Kommunikation konnte bisher nur zwischen zwei Teilnehmern getestet werden. Ein sinnvoller nächster Schritt wäre ein Test mehrerer Teilnehmer. Für die Herstellung weiterer Platinen liegen der Arbeit die EAGLE-Projektdateien der Platine für die in ihrer Funktion verifizierte Revision 1.5 und Revision 2.0 bei. Dabei stellt Revision 2.0 einige Optimierungen des Platinendesigns bereit und bietet ggf. eine Grundlage für weitere Entwicklungsarbeiten an der Hardware. Die entwickelte Software kann ohne großen Aufwand angepasst werden, da die Konfiguration des FlexRay-Controllers auf allen Teilnehmern dieselbe ist. Auf den hinzukommenden Teilnehmern muss lediglich ein weiterer Message Buffer angelegt werden.

Die entwickelte Software bietet eine bereits verifizierte Konfiguration des FlexRay-Controllers. Für eine komfortable Neukonfiguration des Controllers kann als nächster Schritt ein Software-Tool mit graphischer Oberfläche am PC entwickelt werden. Die benötigte Anbindung an den PC erfolgt über eine Ethernet-Schnittstelle, welche von dem Mikrocontroller-Modul TI HDK bereit gestellt wird.

Die Umstellung des drahtgebundenen Physical Layers auf einen drahtlosen Layer benötigt zunächst einen Modulationsalgorithmus, welcher zukünftig entwickelt werden muss. Für diese Aufgabe steht ein leistungsstarkes FPGA-Modul bereit, dessen Grundfunktion verifiziert wurde. Außerdem wurde ein Steckplatz für ein zukünftiges Drahtlos-Modul vorgesehen. Die Entwicklung dieses Moduls stellt einen weiteren sinnvollen nächsten Schritt im Rahmen des X-by-Wire(less)-Projekts dar.

A. Unterlagen zur Evaluierungsplattform

Der Anhang A bietet weitere Dokumentation zur Evaluierungsplattform. In Abschnitt A.1 ist die Tabelle aller Signalleitungen auf der Platine zu finden. Abschnitt A.2 zeigt Abbildungen des PCB Layouts der realisierten und der geplanten Revision der Evaluierungsplattform. Die Schaltpläne der Evaluierungsplattform, die mit dem Schaltplan Editor EAGLE erstellt wurden, sind unter Abschnitt A.3 abgebildet. Abschnitt A.4 zeigt das Messbild der Schaltfrequenz des 5 V Schaltreglers. Weiter Messbilder der FlexRay-Datenkommunikation sind in Abschnitt A.5 abgebildet. Der letzte Abschnitt A.6 zeigt ein Foto des Aufbaus für den Reflow-Lötvorgang.

A.1. Übersichtstabelle zu den Signalleitungen der Evaluierungsplattform

Die Tabelle A.1 zeigt die Verbindung aller Signalleitungen des Evaluationsboards untereinander. Signalleitungen die an Lötunkten (Test Points) enden oder nicht für die Kommunikation zwischen den Aufsteckplatinen verantwortlich sind, sind nicht in der Tabelle enthalten.

Signalname	Pin TI HDK	Pin MicroZed	Pin Drahtlos-M.	Pin TJA1082 (U3,U4)
LIN_RX	5, HDK_J11	53, JX1		
LIN_TX	6, HDK_J11	61, JX1		
CAN3_RX	11, HDK_J11	67, JX1		
CAN3_TX	12, HDK_J11	73, JX1		
FRAY_RX1	13, HDK_J11	29, JX1		RXD, U3
FRAY_TX1	15, HDK_J11	17, JX1		TXD, U3
FRAY_TXEN1	17, HDK_J11	23, JX1		TXEN, U3
FRAY_RX2	14, HDK_J11	47, JX1		RXD, U4
FRAY_TX2	16, HDK_J11	35, JX1		TXD, U4
FRAY_TXEN2	18, HDK_J11	41, JX1		TXEN, U4
GIOA[3]	21, HDK_J11			STBN, U3
GIOA[4]	24, HDK_J11			STBN, U4
GIOA[5]	23, HDK_J11			ERRN, U3
GIOA[6]	26, HDK_J11			ERRN, U4
GIOB[4]	32, HDK_J11		93, J3	

GIOB[5]	31, HDK_J11	81, JX1	
GIOB[6]	34, HDK_J11		91, J3
GIOB[7]	33, HDK_J11	82, JX1	
NHET1[2]	40, HDK_J11		81, J3
NHET1[4]	42, HDK_J11		79, J3
SCIRX	44, HDK_J11		97, J3
SCITX	49, HDK_J11		99, J3
NHET1[15]	51, HDK_J11	Rev.2: 62, JX2	
NHET1[16]	54, HDK_J11	Rev.2: 64, JX2	
AD1IN[0]	32, HDK_J9		85, J3
AD1IN[2]	34, HDK_J9		87, J3
SPI2SOMI	59, HDK_J10		SDO, U3 and U4
SPI2NCS1	62, HDK_J10		SCSN, U4
SPI2NCS0	53, HDK_J10		SCSN, U3
SPI2CLK	64, HDK_J10		SCLK, U3 and U4
MOD_DQ_1		11, JX1	98, J3
MOD_DQ_2		13, JX1	100, J3
R_MIX_3_P		18, JX1	94, J3
R_MIX_3_N		20, JX1	92, J3
R_MIX_5_P		24, JX1	88, J3
R_MIX_5_N		26, JX1	86, J3
R_MIX_7_P		30, JX1	82, J3
R_MIX_7_N		32, JX1	80, J3
R_MIX_9_P		36, JX1	76, J3
R_MIX_9_N		38, JX1	74, J3
R_MIX_11_P		42, JX1	70, J3
R_MIX_11_N		44, JX1	68, J3
R_MIX_13_P		48, JX1	64, J3
R_MIX_13_N		50, JX1	62, J3

Tabelle A.1.: Übersicht aller Signalleitungen der Evaluationsplattform

A.2. PCB Layout der Evaluierungsplattform

Abbildung A.1 zeigt das PCB Layout der Evaluierungsplatine Revision 1.5. Diese Ausführung wurde zweimal hergestellt, bestückt und in Betrieb genommen. Abbildung A.2 zeigt eine weitere Ausführung, Revision 2 der Evaluierungsplattform. Diese wurde nicht hergestellt, sondern dient als Optimierungsansatz für zukünftige Platinen (vgl. Abschnitt 6.3).

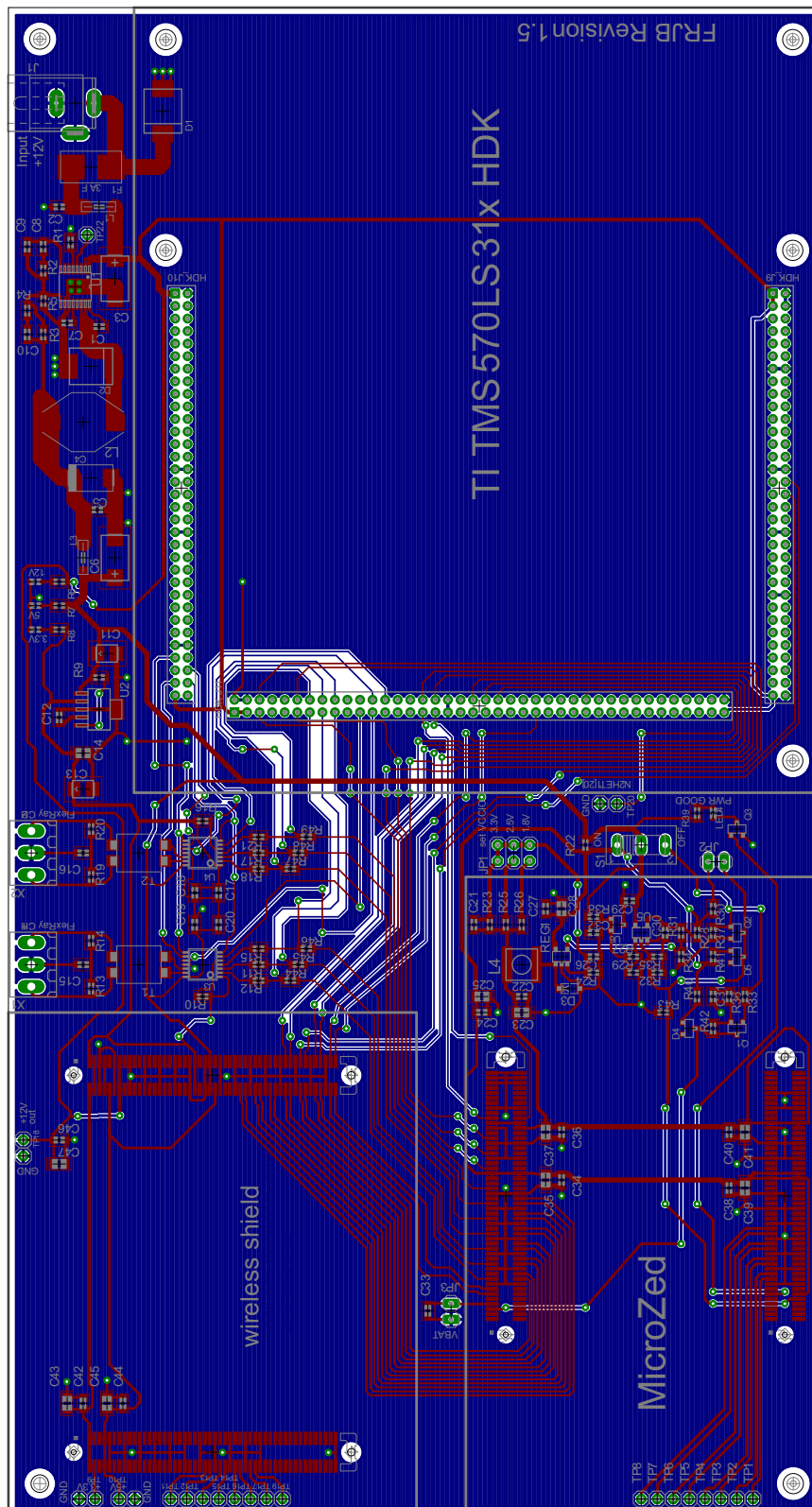


Abbildung A.1.: PCB Layout der Evaluierungsplattform Revision 1.5 (realisierter Prototyp)

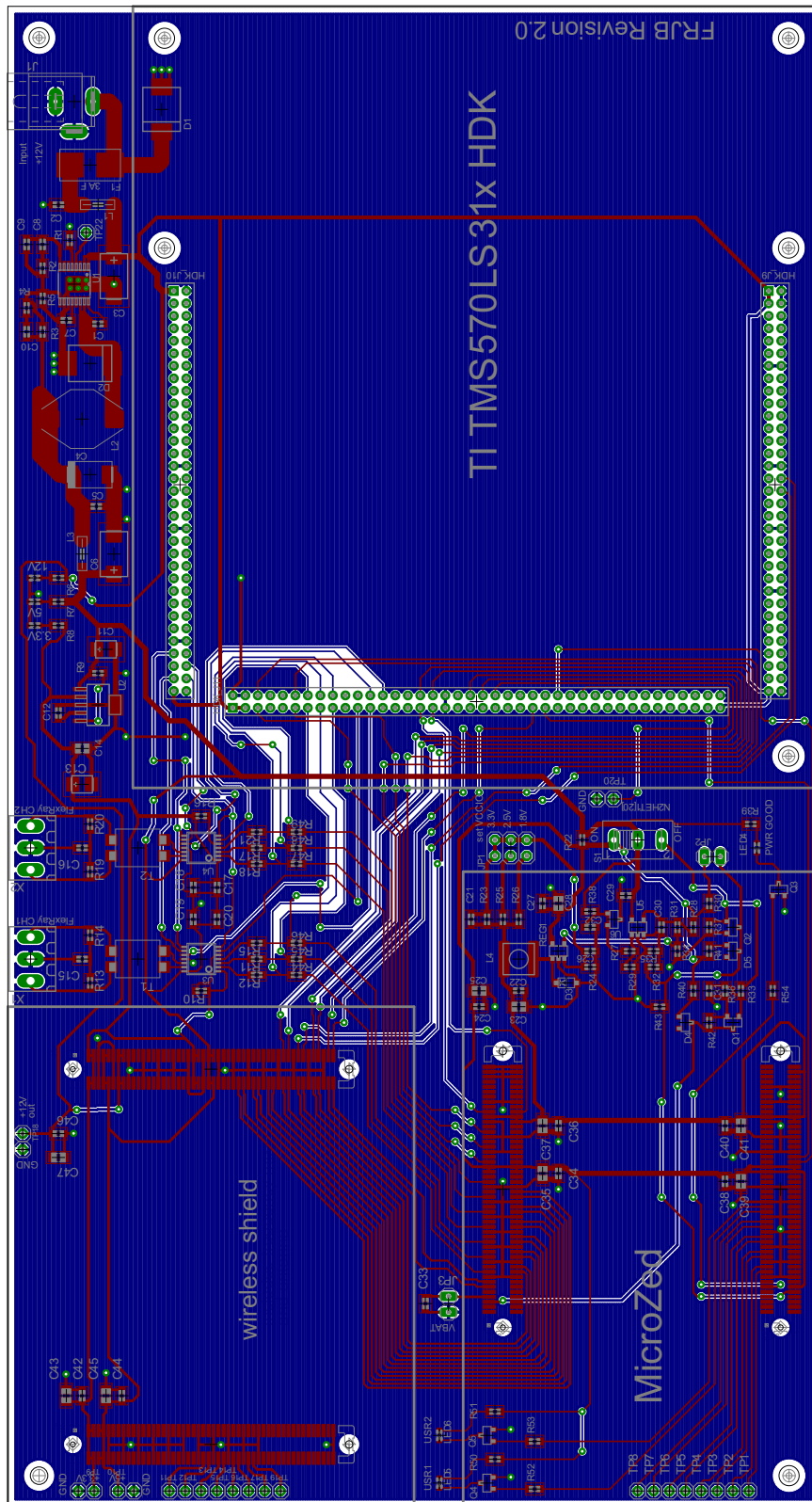


Abbildung A.2.: PCB Layout der Evaluierungsplattform Revision 2

A.3. Schaltplan der Evaluierungsplattform

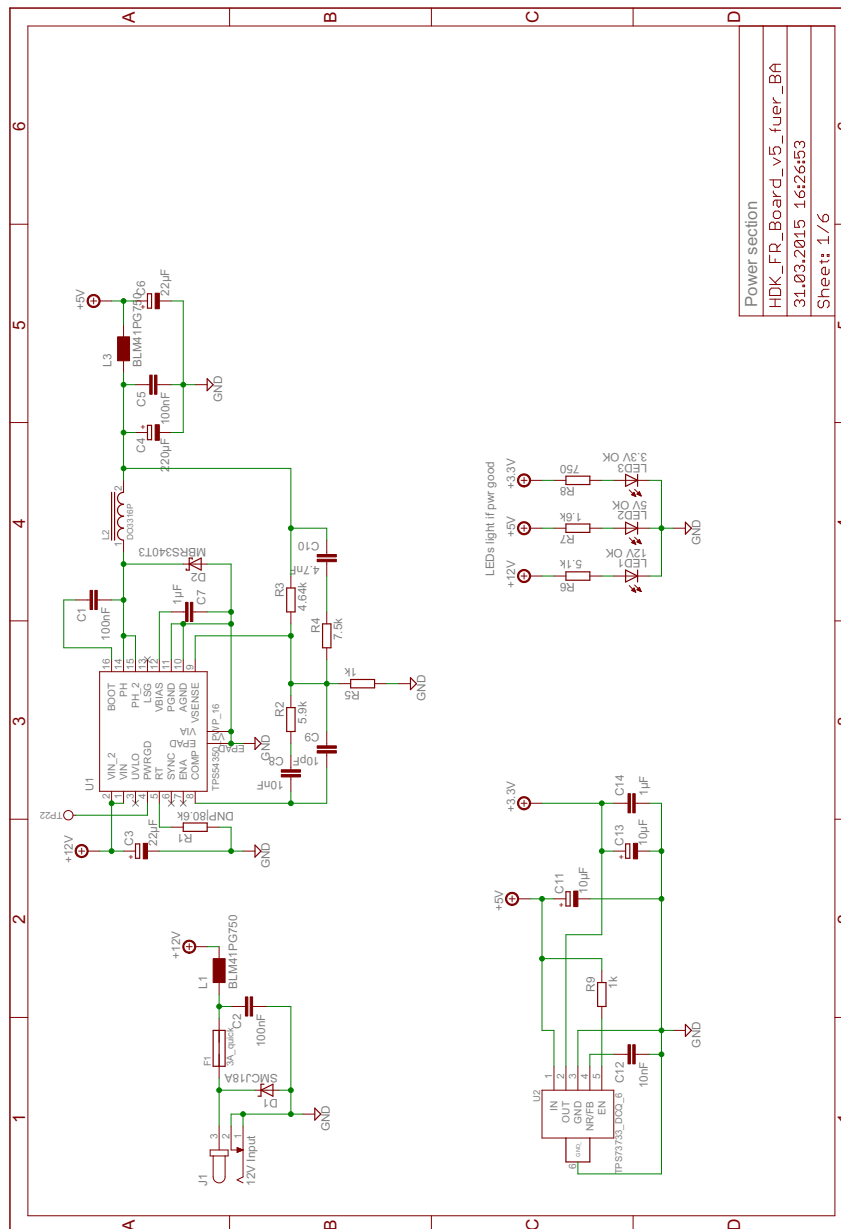


Abbildung A.3.: Schaltplan der Spannungsversorgung für das 12 V, 5 V und 3,3 V-Netz

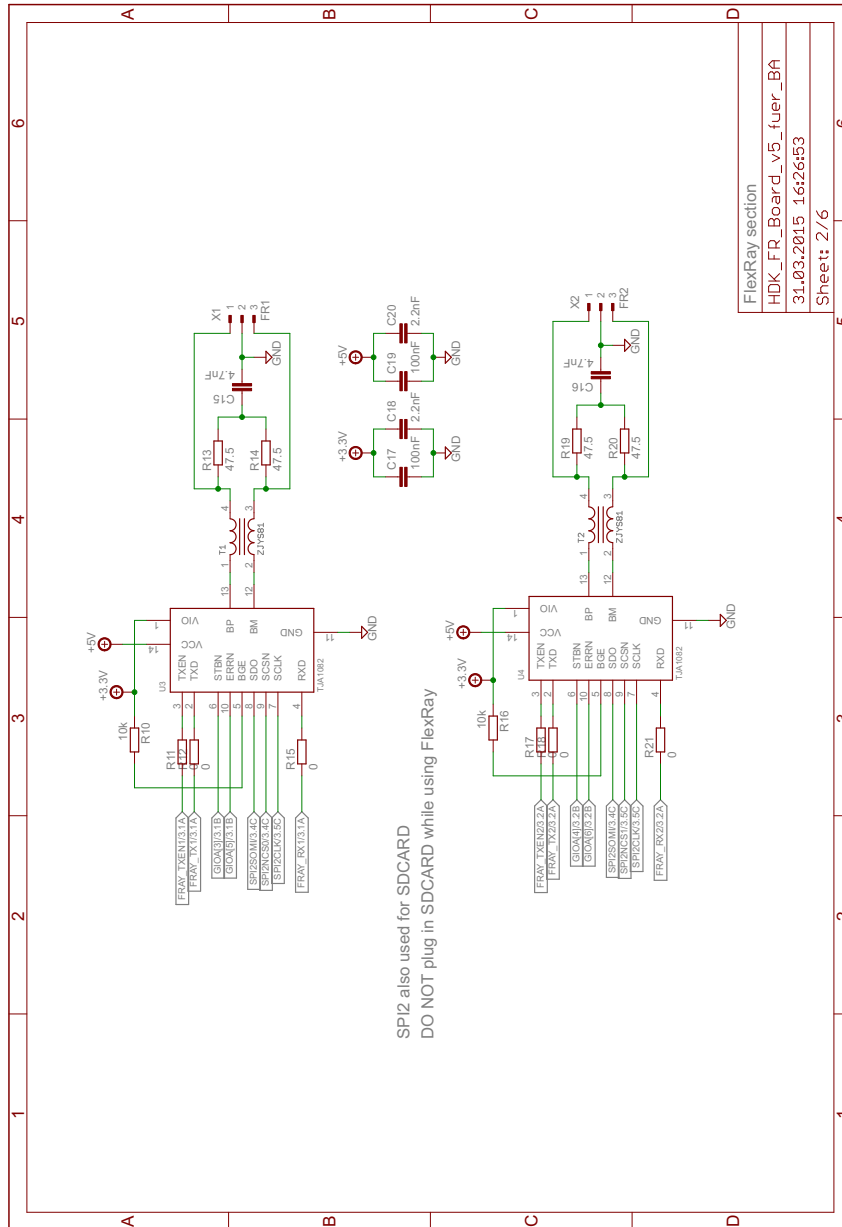
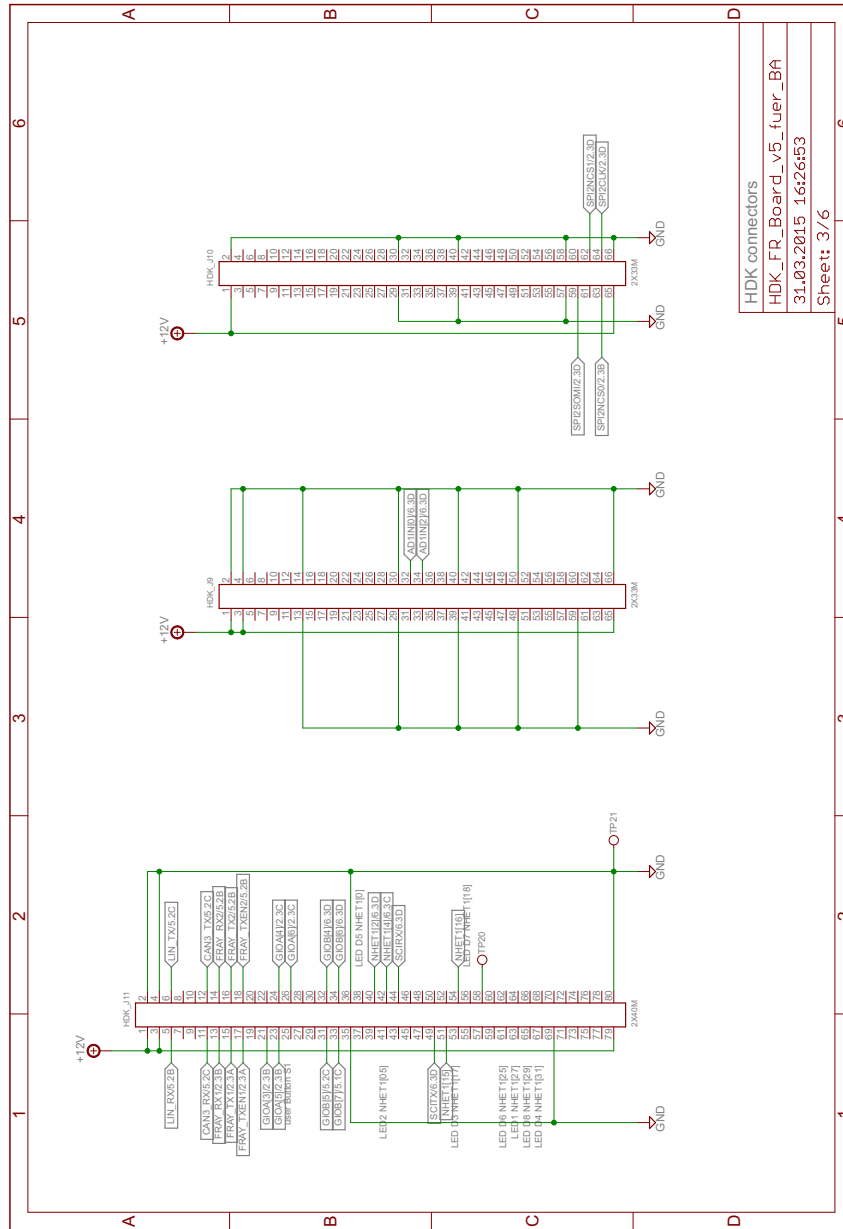


Abbildung A.4.: Schaltplan der beiden FlexRay Transceiver



HDK connectors
 HDK_FR_Board_v5_fuer_BA
 31.03.2015 16:26:53
 Sheet 3/6

Abbildung A.5.: Schaltplan der Steckverbinder zum Mikrocontroller-Modul TI HDK

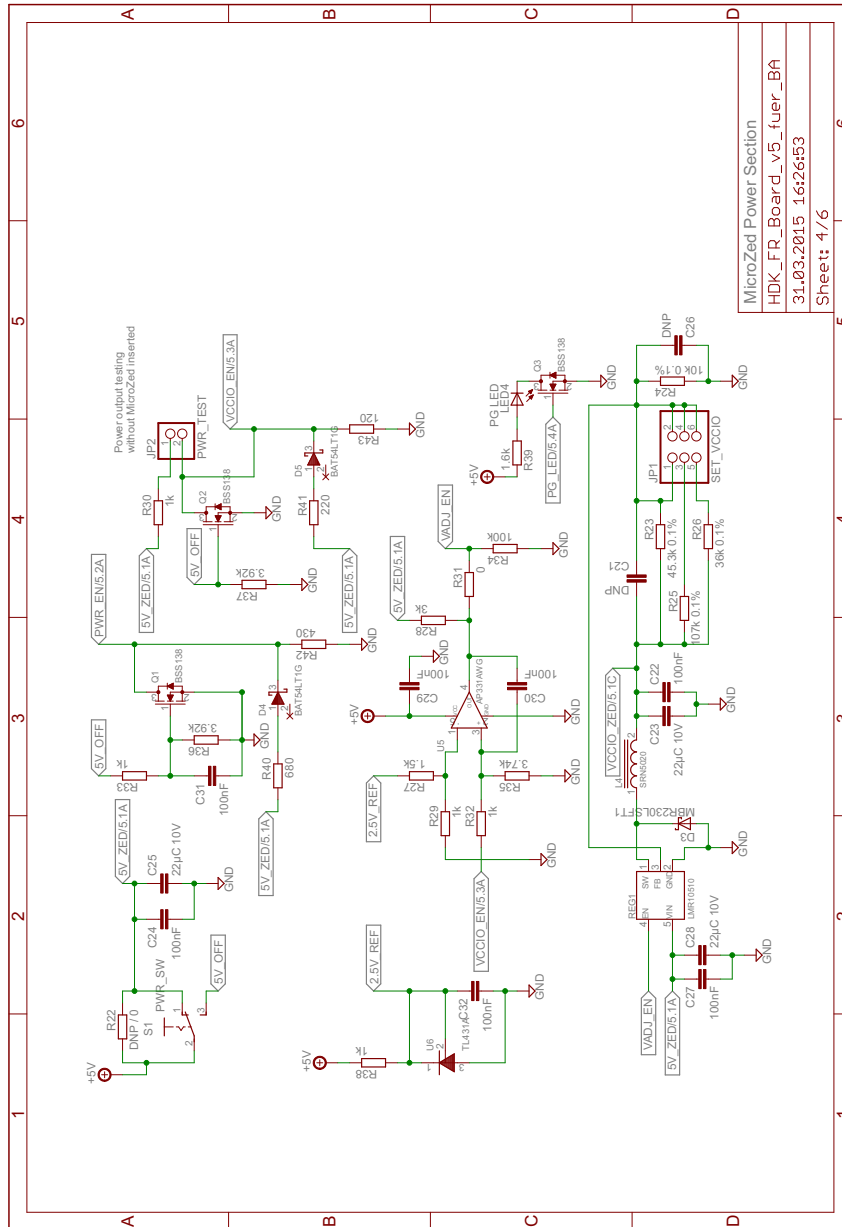


Abbildung A.6.: Schaltplan der einstellbaren Spannungsversorgung für das FPGA-Modul

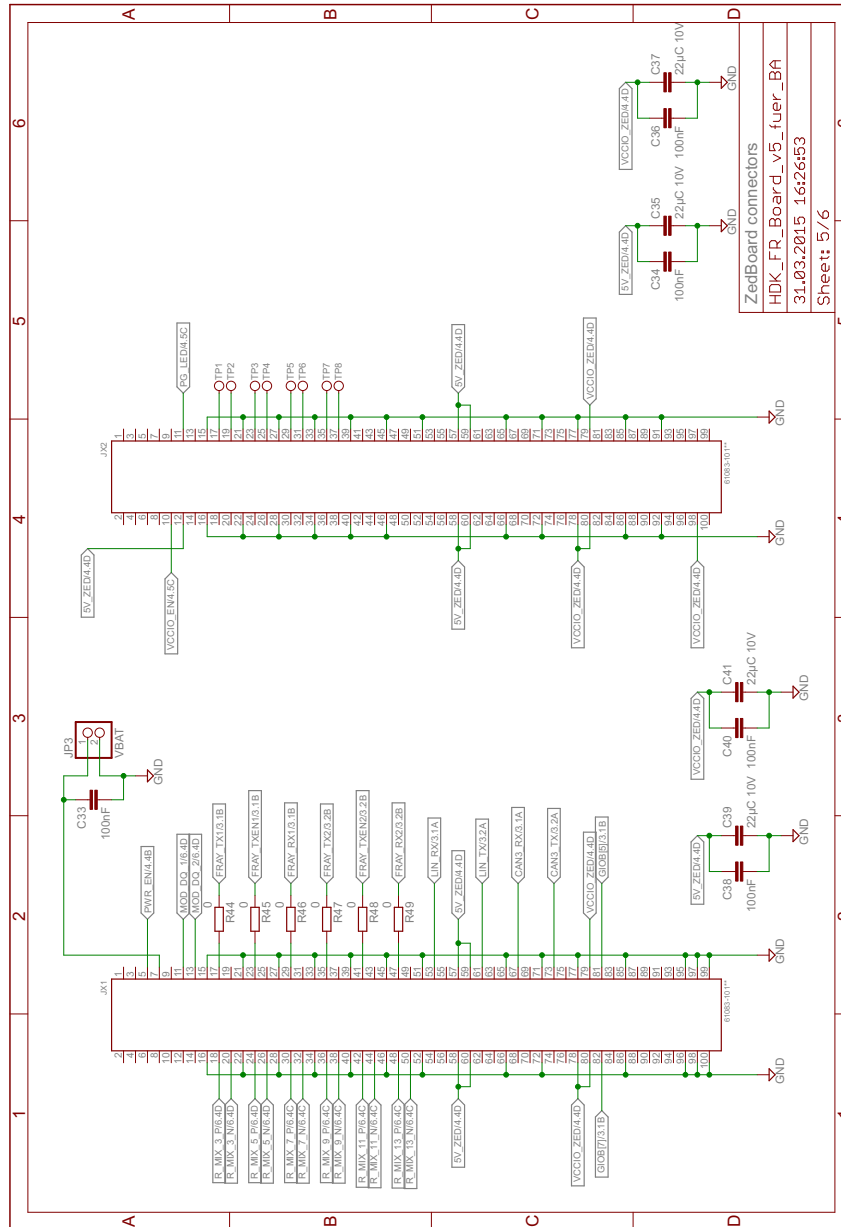


Abbildung A.7.: Schaltplan der Steckverbinder zum FPGA-Modul

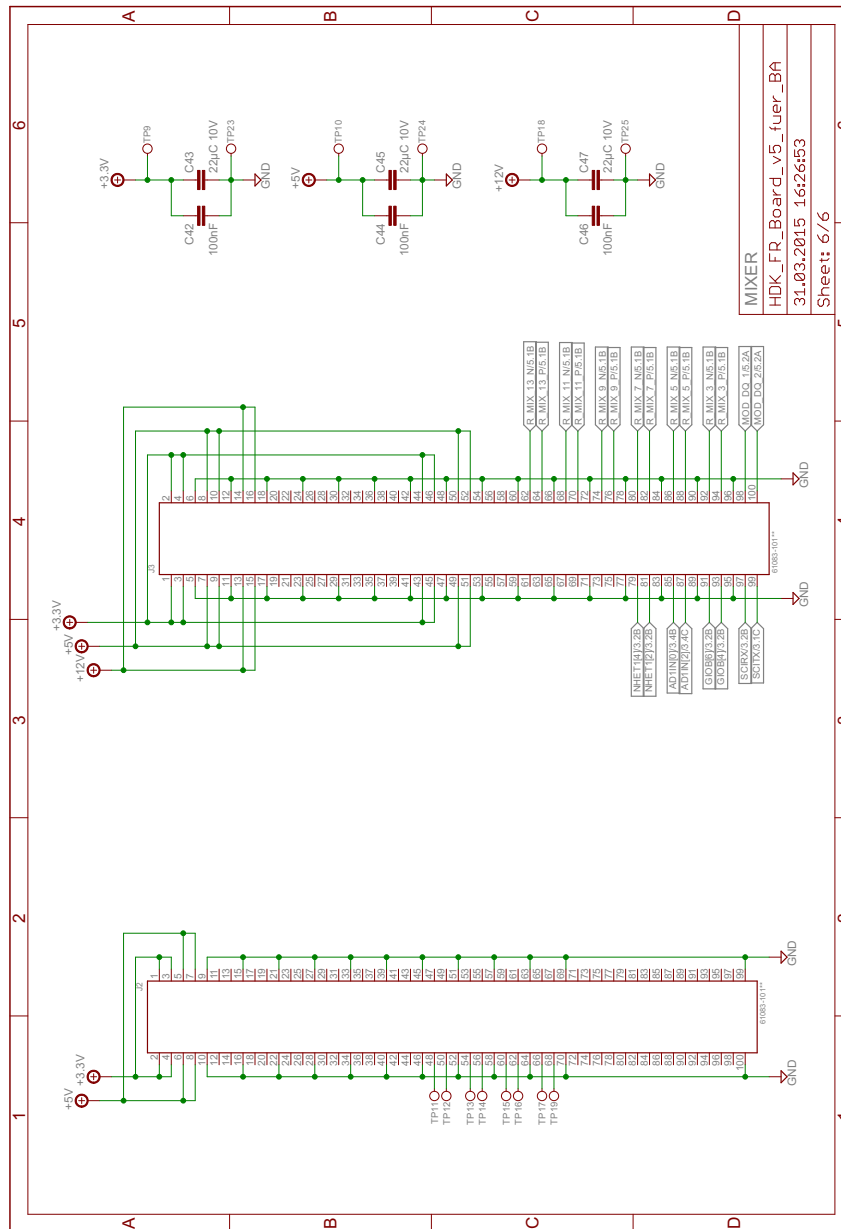


Abbildung A.8.: Schaltplan der Steckverbinder zum Drahtlos-Modul

A.4. Schaltfrequenz des 5V Schaltreglers

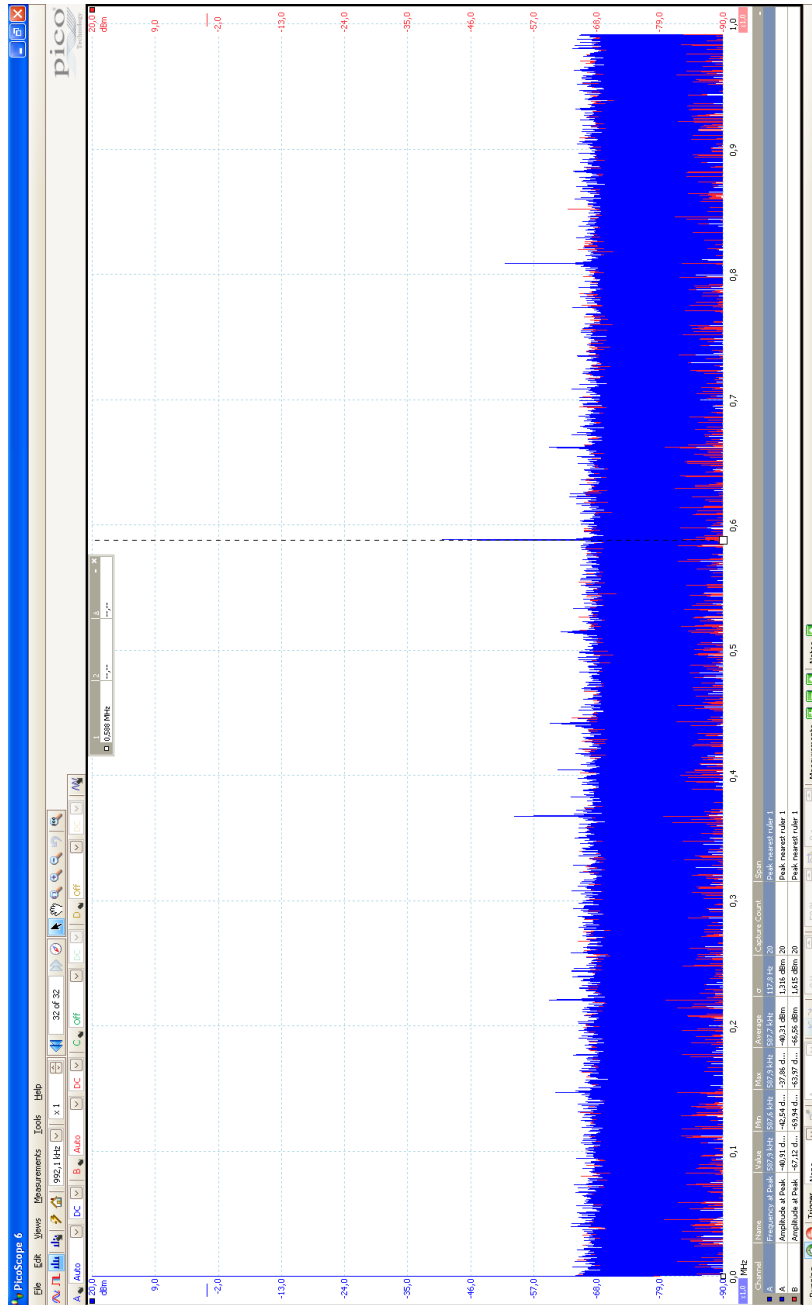


Abbildung A.9.: Messung der Schaltfrequenz des 5 V Schaltreglers vor dem SMD-Ferrit L3 (Kanal A) und hinter dem SMD-Ferrit (Kanal B)

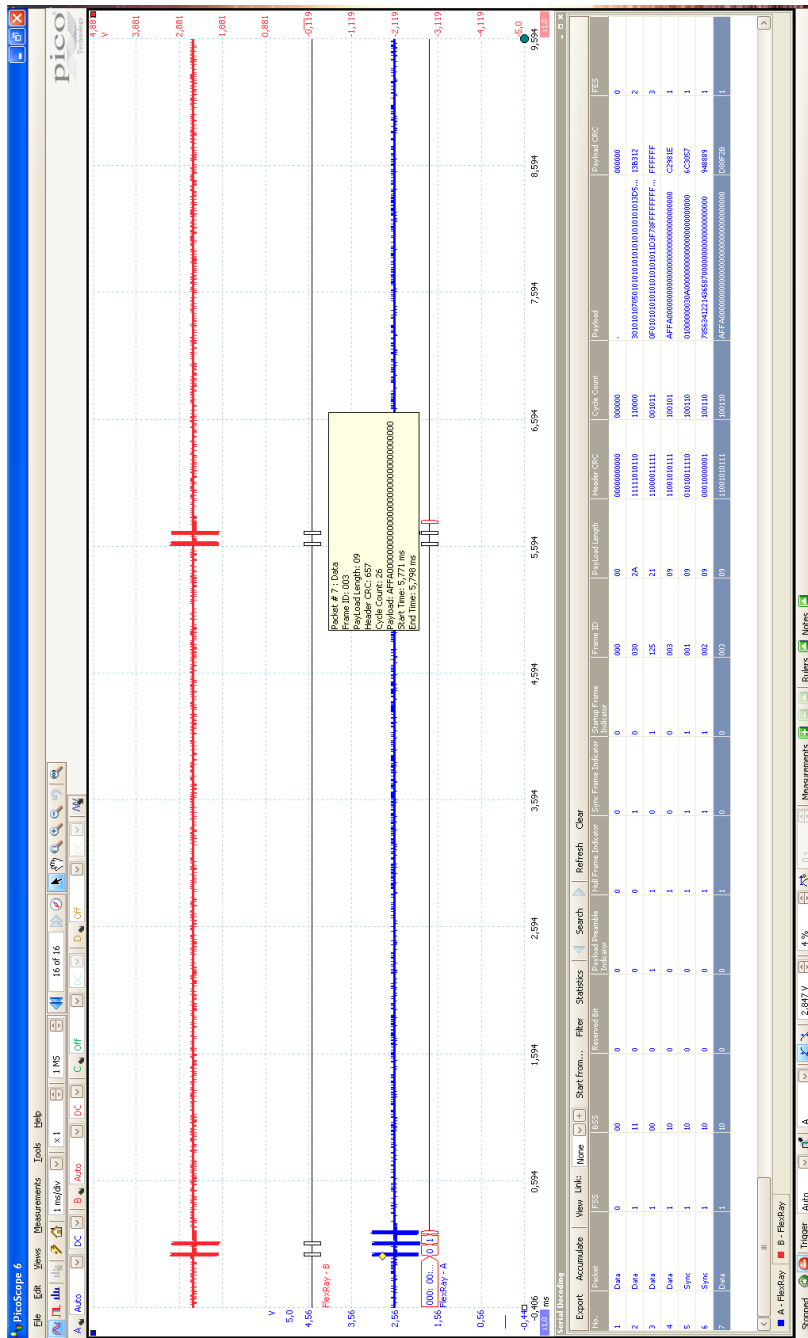


Abbildung A.12.: Messung von drei statischen FlexRay-Botschaften auf Kanal A (Scope Kanal A zeigt FlexRay Kanal A, Scope B zeigt FlexRay B)

A.6. Reflow-Ofen



Abbildung A.13.: Reflow-Ofen (links) mit temperaturgesteuertem Netzregler (rechts)

Literaturverzeichnis

- [ABR13] ABRACON Corporation. Pure Silicon Clock Oscillator - ASEM1-25.000MHZ-LC-T, 2013. URL: <http://www.mouser.com/ds/2/3/ASEM-44258.pdf>, [letzter Zugriff am 06.04.2015].
- [Ant07] Anton Schedl. Goals and Architecture of FlexRay at BMW, 2007. URL: http://vector.com/portal/medien/cmc/speeches/FlexRay_Symposium_2007/FRS07_02_Schedl.pdf, [letzter Zugriff am 21.02.2015].
- [Avn14a] Avnet, Inc. MicroZed Breakout Carrier Card Schematics Rev A , 2014. URL: http://zedboard.org/sites/default/files/documentations/MBCC-BKO_RevA_Schematic_131114.pdf, [letzter Zugriff am 01.03.2015].
- [Avn14b] Avnet, Inc. MicroZed Breakout Carrier Card User's Guide , 2014. URL: http://zedboard.org/sites/default/files/documentations/MBCC_BKO_UG_1_1.pdf, [letzter Zugriff am 01.03.2015].
- [Avn15a] Avnet, Inc. Auflistung der Zedboard-Varianten mit Preisen, 2015. URL: <http://zedboard.org/buy>, [letzter Zugriff am 27.03.2015].
- [Avn15b] Avnet, Inc. MicroZed Breakout Carrier Card, 2015. URL: <http://zedboard.org/product/microzed-breakout-carrier-card>, [letzter Zugriff am 27.03.2015].
- [Avn15c] Avnet, Inc. MicroZed Hardware User's Guide Revision 1.6, 2015. URL: http://zedboard.org/sites/default/files/documentations/MicroZed_HW_UG_v1_6.pdf, [letzter Zugriff am 07.03.2015].
- [Avn15d] Avnet, Inc. MicroZed Produktseite, 2015. URL: <http://zedboard.org/product/microzed>, [letzter Zugriff am 27.03.2015].
- [Avn15e] Avnet, Inc. ZedBoard - Produktseite, 2015. URL: <http://zedboard.org/product/zedboard>, [letzter Zugriff am 05.04.2015].
- [Bet15a] Beta LAYOUT GmbH. Homepage Beta Layout , 2015. URL: <https://www.beta-layout.com/>, [letzter Zugriff am 05.04.2015].

- [Bet15b] Beta LAYOUT GmbH. REFLOW-KIT V3 PRO, 2015. URL: http://www.beta-estore.com/download/rk/RK-10590_640.pdf, [letzter Zugriff am 05.04.2015].
- [Bun14] Bundesnetzagentur. Vfg 30/2014: Allgemeinzuteilung für Short Range Devices (SRD), 2014. URL: http://www.bundesnetzagentur.de/SharedDocs/Downloads/DE/Sachgebiete/Telekommunikation/Unternehmen_Institutionen/Frequenzen/Allgemeinzuteilungen/2014_69_SRD_pdf.pdf?__blob=publicationFile&v=1, [letzter Zugriff am 03.04.2015].
- [Dig13] Digi International Inc. XBee Wi-Fi - Embedded Wi-Fi Module for OEMs, 2013. URL: http://dlnmh9ip6v2uc.cloudfront.net/datasheets/Wireless/WiFi/ds_xbeewifis6b.pdf, [letzter Zugriff am 27.03.2015].
- [Ext12] Extech Instruments. User's guide Extech470, 2012. URL: http://www.extech.com/instruments/resources/manuals/EX470_UM.pdf, [letzter Zugriff am 13.02.2015].
- [Fai14] Fairchild Semiconductor. SMCJ5V0(C)A - SMCJ170(C)A 1500 Watt Transient Voltage Suppressors, 2014. URL: <https://www.fairchildsemi.com/datasheets/SM/SMCJ18A.pdf>, [letzter Zugriff am 17.02.2015].
- [FCI15] FCI. BERGSTAK® 0.8mm Mezzanine Connectors, 2015. URL: http://portal.fciconnect.com/Comergent/fci/documentation/datasheet/boardwiredtoboard/bwb_bergstak_08mm_mezzanine.pdf, [letzter Zugriff am 07.03.2015].
- [Fle06a] FlexRay Consortium. FlexRay Communications System Electrical Physical Layer Application Notes Version 2.1 Revision B, 2006. Auf Nachfrage erhältlich bei Herrn Prof. Dr. Lutz Leutelt.
- [Fle06b] FlexRay Consortium. FlexRay Communications System Electrical Physical Layer Specification Version 2.1 Revision B, 2006. Auf Nachfrage erhältlich bei Herrn Prof. Dr. Lutz Leutelt.
- [Fre15a] Freescale Semiconductors. MFR4310: FlexRay Communication Controller, 2015. URL: http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=MFR4310, [letzter Zugriff am 03.04.2015].
- [Fre15b] Freescale Semiconductors. MFR4310: FlexRay Communication Controller - Buy / Parametrics, 2015. URL: <http://www.freescale.com/webapp/sps/>

- site/prod_summary.jsp?code=MFR4310&cof=0&am=0&tab=Buy_Parametric_Tab&fromSearch=false, [letzter Zugriff am 03.04.2015].
- [Fre15c] Freescale Semiconductors. S12XFSTARTERKITE: S12XF 2-Node Starter Kit, 2015. URL: http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=S12XFSTARTERKITE, [letzter Zugriff am 03.04.2015].
- [Fuj06] Fujitsu. MB88121 FlexRay Communication Controller, 2006. URL: http://www.fujitsu.com/downloads/CN/fmc/MCU/flexray/MB88121_Flexray_Nov_2006.pdf, [letzter Zugriff am 03.04.2015].
- [HAW15] HAW Hamburg. Zukunftsprogramm der Fakultät TI, Kategorie II: Neue Lehrangebote, 2015. URL: <http://www.haw-hamburg.de/fakultaeten-und-departments/ti/unsere-fakultaet/zukunftsprogramm-der-fakultaet-ti/geofoerderte-projekte.html#c125639>, [letzter Zugriff am 09.03.2015].
- [Int13a] International Organization for Standardization. ISO 17458-1:2013, 2013. URL: http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=59804, [letzter Zugriff am 05.04.2015].
- [Int13b] International Organization for Standardization. ISO 17458-5:2013, 2013. URL: http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=59809, [letzter Zugriff am 05.04.2015].
- [Kle13] Stephan Kleuker. *Grundkurs Software-Engineering mit UML: Der pragmatische Weg zu erfolgreichen Softwareprojekten (German Edition)*. Springer Vieweg, 3., korr. und erw. Aufl. 2013 edition, 8 2013.
- [Mur11] Murata Seisakusho. SMD/Block Type EMI Supression Filters EMIFIL, 2011. URL: <http://www.mouser.com/ds/2/281/Murata%20BL%20Series-202933.pdf>, [letzter Zugriff am 19.02.2015].
- [Mur14] Murata Seisakusho. Murata NTC Thermistors, 2014. URL: <http://www.murata.com/~media/webrenewal/support/library/catalog/products/thermistor/ntc/r44e.ashx>, [letzter Zugriff am 11.02.2015].
- [NXP10] NXP Semiconductors. An10854_1 application hints rev1.1 tja1082, 2010. erhältlich bei NXP Semiconductors mittels Formular http://www.nxp.com/documents/application_note/AN10854_1_ApplicationHints_Rev1_1_TJA1082.doc, [letzter Zugriff am 06.02.2015].

- [NXP12] NXP Semiconductors. FlexRay Node Transceiver TJA1082 datasheet, 2012. URL: http://www.nxp.com/documents/data_sheet/TJA1082.pdf, [letzter Zugriff am 06.02.2015].
- [NXP15] NXP Semiconductors. TJA1082TT FlexRay Node Transceiver, 2015. URL: http://www.nxp.com/products/interface_and_connectivity/can_lin_flexray_transceivers/flexray_transceivers/TJA1082TT.html, [letzter Zugriff am 27.03.2015].
- [Ort05] Lorena Diaz Ortega. *Physical Layer Modellierung der Bussysteme CAN und Flex-Ray im Kraftfahrzeug*. Shaker, 1., aufl. edition, 12 2005.
- [Pic13] Pico Technology Ltd. PicoScope @ 3000 Series, 2013. URL: <https://www.picotech.com/download/datasheets/PicoScope3400.pdf>, [letzter Zugriff am 01.04.2015].
- [Rau07] Mathias Rausch. *FlexRay - Grundlage, Funktionsweise, Anwendung*. Hanser Fachbuchverlag, 11 2007.
- [Rob07] Robert Bosch GmbH. E-Ray - Application Note AN003 - Message RAM Configuration, 2007. URL: http://www.bosch-semiconductors.de/media/en/pdf_1/ipmodules_1/flexray/071203_an003_1r02.pdf, [letzter Zugriff am 19.03.2015].
- [Rob09] Robert Bosch GmbH. E-Ray - FlexRay IP-Module - Users Manual - Revision 1.2.7, 2009. URL: http://www.bosch-semiconductors.de/media/en/pdf_1/ipmodules_1/flexray/eray_users_manual_1_2_7.pdf, [letzter Zugriff am 19.03.2015].
- [Rob12] Robert Bosch GmbH. FlexRay Communication Controller IP , 2012. URL: http://www.bosch-semiconductors.de/en/ubk_semiconductors/safe/ip_modules/flexray/flex_ray_communication_controller_ip.html, [letzter Zugriff am 04.04.2015].
- [Ste07] Stephan Janouch. Wachstum auf dem FlexRay-Chipmarkt - Teil 3: Stand-alone Communication Controller als Auslaufmodell, 2007. URL: <http://www.elektroniknet.de/automotive/sonstiges/artikel/294/2/>, [letzter Zugriff am 21.02.2015].
- [Tem] Christopher Temple. Networking the FlexRay Way, -. URL: http://cache.freescale.com/files/peripherals_coprocessors/doc/reports_presentations/NETWRKFRWAY.pdf, [letzter Zugriff am 06.02.2015].

- [Tex04] Texas Instruments Inc. 4.5V to 20V Input, 3-A Output Synchronous PWM Switcher w/Integrated FET (SWIFT) (Rev. C), 2004. URL: <http://www.ti.com/lit/ds/symlink/tps54350.pdf>, [letzter Zugriff am 17.02.2015].
- [Tex05] Texas Instruments Inc. PCB Layout Guidelines for Power Controllers, 2005. URL: <http://www.ti.com/lit/ml/slua366/slua366.pdf>, [letzter Zugriff am 13.02.2015].
- [Tex06] Texas Instruments Inc. High-Speed Layout Guidelines, 2006. URL: <http://www.ti.com/lit/an/scaa082/scaa082.pdf>, [letzter Zugriff am 27.03.2015].
- [Tex11] Texas Instruments Inc. HDK Experimenter Schematic Revstar, 2011. URL: http://processors.wiki.ti.com/images/5/53/Hdk_experimenter_revstar.zip, [letzter Zugriff am 06.02.2015].
- [Tex12] Texas Instruments Inc. TMS570LS31x HDK Schematics Revision E, 2012. URL: http://processors.wiki.ti.com/images/e/ec/TMS570LS31x_HDK_Schematics_RevE.pdf, [letzter Zugriff am 11.02.2015].
- [Tex13a] Texas Instruments Inc. 1A Low-Dropout Regulator with Reverse Current Protection TPS737xx, 2013. URL: <http://www.ti.com/lit/ds/sbvs067p/sbvs067p.pdf>, [letzter Zugriff am 17.02.2015].
- [Tex13b] Texas Instruments Inc. LMR10510 SIMPLE SWITCHER 5.5Vin, 1A Step-Down Voltage Regulator in SOT-23, 2013. URL: <http://www.ti.com/lit/ds/symlink/lmr10510.pdf>, [letzter Zugriff am 27.03.2015].
- [Tex13c] Texas Instruments Inc. TMS570LS31x Hercules Development Kit (HDK) User's Guide, 2013. URL: <http://www.ti.com/lit/ug/spnu509b/spnu509b.pdf>, [letzter Zugriff am 27.03.2015].
- [Tex13d] Texas Instruments Inc. TMS570LS31x/21x 16/32-Bit RISC Flash Microcontroller - Technical Reference Manual, 2013. URL: <http://www.ti.com/cn/cn/lit/ug/spnu499b/spnu499b.pdf>, [letzter Zugriff am 19.03.2015].
- [Tex15a] Texas Instruments Inc. HAL Code Generator tool, 2015. URL: <http://www.ti.com/tool/halcogen#descriptionArea>, [letzter Zugriff am 23.02.2015].
- [Tex15b] Texas Instruments Inc. TI Wiki - Category:TMS570, 2015. URL: <http://processors.wiki.ti.com/index.php/Category:TMS570>, [letzter Zugriff am 03.04.2015].

- [Tex15c] Texas Instruments Inc. TI Wiki - TMS570LS31x HDK Kit - Example Programs, 2015. URL: http://processors.wiki.ti.com/index.php/TMS570LS31x_HDK_Kit#Example_Programs, [letzter Zugriff am 03.04.2015].
- [Tex15d] Texas Instruments Inc. TMS570LS1227 16- and 32-Bit RISC Flash Microcontroller, 2015. URL: <http://www.ti.com/product/tms570ls1227>, [letzter Zugriff am 27.03.2015].
- [Tex15e] Texas Instruments Inc. TMS570LS12x/11x Hercules Development Kit, 2015. URL: <http://www.ti.com/tool/tmdx570ls12hdk>, [letzter Zugriff am 27.03.2015].
- [Tex15f] Texas Instruments Inc. TMS570LS3137 16- and 32-Bit RISC Flash Microcontroller, 2015. URL: <http://www.ti.com/product/tms570ls3137>, [letzter Zugriff am 27.03.2015].
- [Tex15g] Texas Instruments Inc. TMS570LS31x Hercules Development Kit, 2015. URL: <http://www.ti.com/tool/TMDS570LS31HDK>, [letzter Zugriff am 27.03.2015].
- [Tob08] Tobias Hammer. HTerm (Homepage), 2008. URL: <http://www.der-hammer.info/terminal/>, [letzter Zugriff am 04.04.2015].
- [Vec15a] Vector Informatik GmbH. Vector E-Learning - Lernmodul FlexRay - Einheit 15/32, 2015. URL: https://elearning.vector.com/index.php?&wbt_ls_seite_id=457657&root=376493&seite=vl_flexray_introduction_de, [letzter Zugriff am 09.03.2015].
- [Vec15b] Vector Informatik GmbH. Vector E-Learning - Lernmodul FlexRay - Einheit 2/32, 2015. URL: https://elearning.vector.com/index.php?wbt_ls_kapitel_id=454126&root=376493&seite=vl_flexray_introduction_de, [letzter Zugriff am 09.03.2015].
- [Vec15c] Vector Informatik GmbH. Vector E-Learning - Lernmodul FlexRay - Einheit 25/32, 2015. URL: https://elearning.vector.com/index.php?&wbt_ls_seite_id=463125&root=376493&seite=vl_flexray_introduction_de, [letzter Zugriff am 09.03.2015].
- [Vec15d] Vector Informatik GmbH. Vector E-Learning - Lernmodul FlexRay - Einheit 9/32, 2015. URL: https://elearning.vector.com/index.php?&wbt_ls_seite_id=457650&root=376493&seite=vl_flexray_introduction_de, [letzter Zugriff am 09.03.2015].

- [Wik15a] Wikipedia - die freie Enzyklopädie. Ball Grid Array, 2015. URL: http://de.wikipedia.org/wiki/Ball_Grid_Array, [letzter Zugriff am 04.04.2015].
- [Wik15b] Wikipedia - die freie Enzyklopädie. FlexRay, 2015. URL: <http://de.wikipedia.org/wiki/FlexRay>, [letzter Zugriff am 04.04.2015].
- [Wik15c] Wikipedia - die freie Enzyklopädie. I und Q-Verfahren, 2015. URL: <http://de.wikipedia.org/wiki/I%26Q-Verfahren>, [letzter Zugriff am 05.04.2015].
- [Wik15d] Wikipedia - die freie Enzyklopädie. Spannungsregler - Low-Drop-Längsregler, 2015. URL: <http://de.wikipedia.org/wiki/Spannungsregler#Low-Drop-L.C3.A4ngsregler>, [letzter Zugriff am 04.04.2015].
- [Wik15e] Wikipedia - die freie Enzyklopädie. Spansion, 2015. URL: <http://de.wikipedia.org/wiki/Spansion>, [letzter Zugriff am 03.04.2015].
- [Wik15f] Wikipedia - die freie Enzyklopädie. Thermistor - B parameter equation, 2015. URL: http://en.wikipedia.org/wiki/Thermistor#B_or_.CE.B2_parameter_equation, [letzter Zugriff am 04.04.2015].
- [Wik15g] Wikipedia - die freie Enzyklopädie. X-by-Wire, 2015. URL: <http://de.wikipedia.org/wiki/X-by-Wire>, [letzter Zugriff am 05.04.2015].
- [Wik15h] Wikipedia - The free encyclopedia. Category 5 cable - Characteristics, 2015. URL: http://en.wikipedia.org/wiki/Category_5_cable#Characteristics, [letzter Zugriff am 07.04.2015].
- [Wil14] Wilfried Steiner. Echtzeitanwendungen mit automotive ethernet, 2014. URL: <http://www.elektroniknet.de/automotive/bussysteme/artikel/113002/>, [letzter Zugriff am 05.04.2015].
- [Xil14] Xilinx, Inc. Zynq®-7000 All Programmable SoCs - Combined Product Table, 2014. URL: http://www.xilinx.com/publications/prod_mktg/zynq7000/Zynq-7000-combined-product-table.pdf, [letzter Zugriff am 28.03.2015].
- [Xil15a] Xilinx Inc. Xilinx Power Estimator (XPE), 2015. URL: http://www.xilinx.com/products/design_tools/logic_design/xpe.htm, [letzter Zugriff am 04.04.2015].
- [Xil15b] Xilinx, Inc. Zynq-7000 All Programmable SoC, 2015. URL: http://www.xilinx.com/support/documentation/data_sheets/ds187-XC7Z010-XC7Z020-Data-Sheet.pdf, [letzter Zugriff am 07.03.2015].

Abkürzungsverzeichnis

Bezeichnung	Beschreibung	Seiten
ABS	Antiblockiersystem	9
ADC	Analog-to-Digital Converter	28, 44, 50–52, 62, 67–70
BGA	Ball Grid Array	29
CAN	Controller Area Network	9, 10, 24, 28, 44
CC	Communication Controller	50, 54
CRC	Cyclic Redundancy Check	19
CYCS	Cycle Start Interrupt	54
DNP	Do Not Populate	36
EMV	Elektromagnetische Verträglichkeit	10
ESD	Electrostatic Discharge	34
FPGA	Field Programmable Gate Array	10, 25–33, 35, 38, 39, 42, 44–46, 49, 61–64, 70, 72, 73

Bezeichnung	Beschreibung	Seiten
FRJB	FlexRay Junction Board	46, 72
GPIO	General Purpose Input/Output	28, 44, 45
GTU1	Global Time Unit 1	56
HAW	Hochschule für angewandte Wissenschaften	23, 28, 48
HDK	Hercules Development Kit	27, 29, 30, 33, 34, 43–45, 51, 62, 63, 65, 73
HET	High-End Timer	44, 45
IC	Integrated Circuit	44, 46
ISO	International Organization for Standardization	13
JTAG	Joint Test Action Group	28
Kfz	Kraftfahrzeug	9, 24
LDO	Low-Dropout	36, 37
LED	Light-Emitting Diode	28, 37, 39, 44, 51, 62, 72
LIN	Local Interconnect Network	9, 10, 24, 27, 44
MSB	Most Significant Bit	70

Bezeichnung	Beschreibung	Seiten
NIT	Network Idle Time	17, 18
PCB	Printed Circuit Board	12, 46, 47, 57, 72, 74, 75
POC	Protocol Operational Control	54
RAM	Random Access Memory	45
SCI	Serial Communication Interface	51, 52, 62
SDRAM	Synchronous Dynamic Random Access Memory	28
SMD	Surface-Mount Device	34, 35, 47, 59
SOC	System On Chip	29, 38
SPI	Serial Peripheral Interface	28, 41–44, 51, 52, 57, 67, 68
TDMA	Time Division Multiple Access	7, 13, 17, 18
TI	Texas Instruments	26, 27, 29, 33, 34, 41, 43, 50, 51, 73
UART	Universal Asynchronous Receiver Transmitter	24, 28, 44, 51
USB	Universal Serial Bus	28, 31, 51, 57, 67

Bezeichnung	Beschreibung	Seiten
UTP	Unshielded Twisted Pair	57
WLAN	Wireless Local Area Network	24

Versicherung über die Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §16(5) APSO-TI-BM ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen habe ich unter Angabe der Quellen kenntlich gemacht.

Neu Wulmstorf, 12. April 2015

Ort, Datum

Unterschrift