

Diplomarbeit

Thorsten Eger

Entwicklung von Hard- und Software eines
Readers für drahtlose Sensorik mit
Resonanzabgleich

Thorsten Eger

Entwicklung von Hard- und Software eines Readers für
drahtlose Sensorik mit Resonanzabgleich

Diplomarbeit eingereicht im Rahmen der Diplomprüfung
im Studiengang Informations- und Elektrotechnik
Studienrichtung Informationstechnik
am Department Informations- und Elektrotechnik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. Ing Karl Ragmar Riemschneider
Zweitgutachter : Prof. Dr. Ing. Franz Schubert

Abgegeben am 27. Oktober 2008

Thorsten Eger

Thema der Diplomarbeit

Entwicklung von Hard- und Software eines Readers für drahtlose Sensorik mit Resonanzabgleich

Stichworte

Drahtlos, Sensorik, Resonanzabgleich, RFID, Mikrocontroller, CPLD, MSP430, C, VHDL

Kurzzusammenfassung

Für drahtlose Sensoren entstehen gegenwärtig eine Reihe von Anwendungsfeldern. Im Gegensatz zu den typischen Anwendungen der Radio Frequency Identification RFID ist bei drahtlosen Sensoren die Versorgung der Sensor-Transponder, mit der notwendigen Betriebsspannung, anspruchsvoller. Dies ist durch einen erhöhten Energiebedarf von Sensor-Transpondern begründet. Diese Arbeit beschreibt die Entwicklung eines Lesegeräts, welches für einen Einsatz mit Sensor-Transpondern konzipiert ist. Bei der Realisierung wird auf ein besonderes Abstimmverfahren zur Resonanzabstimmung eingegangen.

Thorsten Eger

Title of the paper

Development of the hard- and software for a wireless sensors reader with resonance-tuning

Keywords

Wireless, sensors, resonancetuning, RFID, microcontroller, CPLD, MSP430, C, VHDL

Abstract

Presently there is a row of application fields for wireless sensors. In contrast to the typical applications of Radio Frequency Identification RFID the supply of the sensor transponders with the necessary operating voltage is more demanding by wireless sensors. This is founded by a raised power demand of sensor transponders. This work describes the development of a reader which is conceived for an application with sensor transponders. By the realisation it is entered on a special tuning procedure for the resonance tuning.

Danksagung

Ich bin denen im Fachbereich Informationstechnik angestellten Laborassistenten zu Dank verpflichtet. Besonders möchte ich den Assistenten Herr Wolff und Herr Pflüger für die geduldige Beantwortung meiner Fragen und der notwendigen Versorgung mit der verwendeten Hardware danken. Ebenfalls danke ich Herrn Otte für die hilfreichen Hinweise zum Gebiet der Analogschaltungen.

Weiterhin danke ich Herrn Prof. Dr. Ing. Riemschneider für seine Betreuung und für das Aufstellen des Themas. Ebenso sei an dieser Stelle Herrn Prof. Dr. Ing. Schubert für die Übernahme der Aufgabe des Zweitgutachters gedankt.

Auch meinen beiden Studienkollegen Tobias Krannich und Stephan Plaschke, welche im gleichen Zeitraum ihre Diplomarbeit bei Herrn Riemschneider absolviert haben, möchte ich für ihre Unterstützung danken. Es war mir stets ein großes Vergnügen fachliche Diskussionen mit ihnen führen zu können.

Abschließend gebührt mein besonderer Dank meiner Freundin und meiner Familie für ihre Geduld und Unterstützung.

Inhaltsverzeichnis

1. Einleitung und Motivation	1
1.1. Problemstellung	1
1.2. Zielstellung	1
2. Grundlagen	3
2.1. RFID	3
2.1.1. RFID-Grundsystem	4
2.1.2. Energieversorgung von Transpondern	5
2.1.3. Frequenzbereiche von RFID-Systemen	5
2.1.4. Codierung der zu übertragenen Daten	6
2.1.5. Modulation der zu übertragenen Daten	7
2.1.6. Normierung von RFID-Systemen	8
2.2. Grundlagen zum Resonanzabgleich	10
2.2.1. Abgleichmethoden aus Vorarbeiten	10
2.2.2. Vorstellung der Messmethode zur Einstellung der Resonanzfrequenz nach Giesler, Böh und Tobergte	12
2.2.3. Zeitliches Auflösungsproblem bei Verwendung des MSP430 Mikrocontroller	13
2.3. Grundlagen zum Reader	15
2.3.1. Reihenschwingkreis	15
2.3.2. Schwingkreisanregung	16
2.3.3. Elektromagnetisches Feld	18
2.3.4. Transformatorprinzip	20
3. Implementierung des Readers	24
3.1. Darstellung der Systemkomponenten	24
3.2. Beschreibung der Readerbox	24
3.2.1. Räumliche Abmaße der Readerbox	24
3.2.2. Dimensionierung der Antennenspulen	25
3.2.3. Modifikation der Readerbox	27
3.3. Beschreibung der Readerplatine	30
3.3.1. Aufbau der Readerplatine	30
3.4. Darstellung der angewandten Messmethode zum Resonanzabgleich	31
3.4.1. Spannungsgleichrichtung zur Amplitudenmessung	32

3.4.2. Schwingkreisanregung und Kapazitätsänderung	33
3.4.3. Spannungskomparator	36
3.4.4. CPLD	38
3.5. Taktgenerierung zur Erzeugung der Sendefrequenz	42
3.6. Modulation	43
3.7. Demodulation	45
3.8. Antennenselektion	47
3.9. Software des Readers	48
3.10. Darstellung der erstellten Readerplatine	51
4. Funktionsnachweis des Readers	53
4.1. Ermittlung der Spannungsamplitude und des Stromflusses an der Antennenspule	53
4.1.1. Auswertung der ermittelten Spannungswerte	55
4.1.2. Auswertung der Stromflussmessung an der Antennenspule	57
4.2. Transponderversorgung	58
4.2.1. Bestimmung des Kopplungsfaktors k	58
4.2.2. Abschätzung des Feldstärkenverlaufs $H_T(x)$ am Transponder	59
4.2.3. Praxistest unter Einsatz des vorhandenen Transponders	61
4.3. Filterkette	64
5. Fazit und Ausblick	66
Literaturverzeichnis	69
Abkürzungsverzeichnis	77
A. Verwendete Hardware	79
A.1. MSP430 Mikrocontroller	79
A.2. Transponder	82
A.3. XC95108 CPLD	83
A.4. 74ACT125 und 74ACT245 Bustreiber	86
A.5. LM311 Spannungskomparator	86
A.6. 4060 Zählerbaustein zur Takterzeugung	87
A.7. HE721 Reed Relais	87
A.8. PI74LVC Pegelumsetzer	88
A.9. LM1086 3,3V Spannungsregler	88
A.10.L7805 5V Spannungsregler	89
A.11.MAX232 Treiber-IC für die serielle Schnittstelle	90
B. Verwendete Software	92
B.1. Mspgcc Toolchain	92

B.2. Eclipse Entwicklungsumgebung	93
B.3. Eagle Platinenlayout-Software	93
B.4. Matlab	94
B.5. PSPICE Simulationssoftware	95
B.6. Modelsim Simulationssoftware	95
B.7. ISE Project Navigator	96
C. Quellcode	99
C.1. Matlab Code	99
C.2. VHDL Code	104
C.3. C Code	112
D. Schaltplan	125
D.1. Schaltplan zum Reader	125

1. Einleitung und Motivation

1.1. Problemstellung

Für drahtlose Sensoren entstehen gegenwärtig eine Reihe von Anwendungsfeldern. Dazu gehören beispielsweise die Medizintechnik oder die Batterieüberwachung. Im Gegensatz zu den typischen Anwendungen der Radio Frequency Identification RFID ist bei drahtlosen Sensoren die Versorgung der Sensor-Transponder, mit der notwendigen Betriebsspannung, anspruchsvoller.

Der Energiebedarf der Sensor-Transponder ist mit einem typischen Wert von $0,1 - 1\text{mW}$ zwar gering, aber durch die zusätzlichen Sensorfunktionen ist er meist höher als bei einfachen RFID-Transpondern. Deren Energiebedarf liegt typischerweise in einem Bereich von $30 - 100\mu\text{W}$.

Die räumliche Ausdehnung des Arbeitsbereiches wird in den angestrebten Anwendungsfällen vergleichsweise groß ausfallen ($10 - 100$ Kubikdezimeter). Dadurch ergeben sich benötigte Arbeitsabstände zum Reader von einigen Dezimetern. Zudem sind die Nullstellen (lokal begrenzte Gebiete unzureichender Versorgung) herkömmlicher Readersysteme weniger akzeptabel, da voraussichtlich feste räumliche Lagebeziehungen der Reader- und Sensor-Transponder-Antennen bestehen werden. In den Anwendungsfeldern werden, durch metallische Gegenstände, im Arbeitsbereich Abschattungseffekte und Feldstärkeverluste nachteilig wirken, insbesondere ist mit einer Verstimmung der Resonanzfrequenz der Antennenschwingkreise zu rechnen.

1.2. Zielstellung

Die Zielstellung dieser Diplomarbeit ist es, einen Reader zu entwickeln, der den genannten Anforderungen gerecht werden kann. Es ist daher zu untersuchen, durch welche geeignete Maßnahmen die Feldstärke im gesamten Arbeitsbereich des Readers ausreichend und stabil gestaltet werden kann. Dazu gehört die schaltungstechnische Realisierung einer ausreichenden Treiberleistung für die Anregung der Antennenschwingkreise.

Im Rahmen der hier vorgestellten Diplomarbeit wird insbesondere auf die Steuerung des Resonanzverhaltens am Antennenschwingkreis eingegangen. Hierbei wird auf eine geeignete Art der Beeinflussung der hierfür notwendigen Parameter eingegangen. Als

Eingangsgröße für die Abstimmung bietet sich die Phasenbeziehung der Treiberspannung zusammen mit der Spannungsamplitude des am Schwingkreis entnommenen Spannungsverlaufs an. Durch elektronisch zuschaltbare Kapazitäten wird auf den Schwingkreis während des Readerbetriebs Einfluss genommen.

Der Sendebereich des Readers wird für den dargestellten Anwendungsfall zu einem Wert von 125kHz definiert. Die Antennen des Readers werden durch einen geeigneten Schaltungsaufbau ansteuerbar. Ein Prozessor, in Kombination mit der notwendigen Hardware, übernimmt die Aufgabe der Ansteuerung der Antennenschwingkreise. Die eingesetzte Hardware ergänzt die zu entwerfende Software durch eine digitale Steuerung und eine Phasenvergleichsschaltung. Wesentliche Teile der Hardware werden als programmierbare Logik realisiert. Die Hardwarefunktion wird in der Beschreibungssprache VHDL und die Ansteuersoftware für den Readerbetrieb wird in der Programmiersprache C realisiert.

Für die Treiber mit den zuschaltbaren Kondensatoren werden geeignete Schaltungen auf der Basis von verfügbaren Standardschaltkreisen entworfen. Die jeweiligen Schaltungen werden in Baugruppen durch die Simulationssoftware PSPICE simuliert und validiert.

Zur Erprobung des Lesegeräts wird ein Transponder aus nicht eigenen Vorarbeiten [DIPL01] verwendet. Dieser Transponder verfügt über Eigenschaften, welche zur drahtlosen Sensorik verwendet werden können. Der zum Einsatz kommende Sensor-Transponder sendet bei einer entsprechenden Anregung Daten, welche die intern ermittelte Betriebsspannung und die ebenfalls intern ermittelte Betriebstemperatur enthalten. Der beschriebene Sensor-Transponder wird im weiteren Verlauf der Arbeit vereinfachend als Transponder bezeichnet.

Weiterhin bildet eine eigene Vorarbeit, in Form einer Studienarbeit [STUD01], den Ausgangspunkt für die Gestaltung der Antennenspulen.

Die Funktion der entworfenen Hardware wird abschließend exemplarisch an einigen Messreihen nachgewiesen.

2. Grundlagen

2.1. RFID

Die Abkürzung RFID steht für den Begriff Radio Frequency Identification und lässt sich ins Deutsche zu Identifikation mittels Funkwellen übersetzen. Die RFID-Technik wird in der Regel zur automatischen Erkennung von Gegenständen und Lebewesen verwendet. Die Besonderheit hierbei ist, dass die Identifikation über einen begrenzten Raum berührungslos und ohne direkten Sichtkontakt durchgeführt werden kann.

Gemäß [INT01] lässt sich die Entwicklungsgeschichte zur RFID wie folgt beschreiben: Erste Vorläufer zur heutigen Technik stammen aus dem Zweiten Weltkrieg und wurden von der britischen Armee als Sekundär-Radar zur Freund-Feind-Erkennung verwendet. Im zivilen Bereich kommt es gegen Ende der 60er Jahre des letzten Jahrhunderts zur Entwicklung eines Identifikationssystems zur Erkennung von Fahrzeugen und Fahrzeugteilen durch die Firma Siemens. Das System wurde als Siemens Car Identification SICARID bezeichnet. In den 70er Jahren kommt es zur Einführung des elektronischen Warensicherungssystems Electronic Article Surveillance EAS im Einzelhandel. Hierbei wurden die ersten Transponder verwendet. Diese hatten eine Speicherkapazität von einem Bit. Weiter geht die Entwicklung in den 80er Jahren im Bereich der Nutztierererkennung und mit der Einführung von elektronischen Mautsystemen. Im Jahre 1999 kommt es zur Einführung eines globalen Standards zur eindeutigen Warenidentifikation, des Electronic Product Code EPC. Mittels dieses Standards ist es möglich, jedes durch einen bestimmten Hersteller erzeugte Produkt, eindeutig zu identifizieren. Dieser Produktcode gilt als Nachfolger zum bisher verwendeten Strichcode auf Waren.

Aktuell findet die RFID-Technik in vielen unterschiedlichen Bereichen ihre Anwendung. Eine genaue Übersicht würde den Rahmen dieser Diplomarbeit sprengen, daher wird hier lediglich eine kleine Auswahl der allgemein gebräuchlichsten Einsatzbereiche gegeben. RFID wird verwendet zur:

- Fahrzeugidentifikation bei Mautsystemen,
- zur Banknotenidentifikation,
- zur Personenidentifikation, z.B deutscher Reisepass,

- zur Patientenidentifikation in Krankenhäusern,
- als Echtheitsmerkmal für Medikamente,
- zur Warenverfolgung in der Logistik,
- als Automobilwegfahrsperre,
- zu Abrechnungszwecken bei der Müllentsorgung an Hausmülltonnen
- und als Zutrittskontrolle bei Großveranstaltungen, z.B EM-Ticket.

In jüngster Zeit wird diese Technik auch zu Sensorikzwecken verwendet. Diese Diplomarbeit behandelt ebenfalls den Versuch, dies zu bewerkstelligen. Hierbei werden, anstelle von reinen Identifikationsdaten, Sensordaten zwischen dem Transponder und dem Reader übertragen. Voraussetzung ist, dass der Transponder über einen oder mehrere Sensoren verfügt, von dem Daten ermittelt werden können. Es ist zu beachten, dass bei der Verwendung von passiven Transpondern die mittels des Sendefelds zur Verfügung stehenden Energie ausreichend ist, um sowohl den Transponder als auch die Sensoren versorgen zu können.

2.1.1. RFID-Grundsystem

Im wesentlichen besteht ein Grundsystem, welches für RFID-Anwendungen konzipiert ist, aus zwei Elementen:

1. Einem kontaktlosen Datenträger, dem Transponder oder Tag.
2. Einem Lesegerät, dem Transceiver

Dabei ist es nicht zwingend vorgegeben, dass das Lesegerät nur lesend auf den Transponder zugreifen kann. Bei geeigneter Hardware ist es ebenfalls möglich, einen Lese- und Schreibzugriff auf den Transponder zu erhalten.

Der Begriff Transponder setzt sich aus den beiden englischen Wörtern Transmitter und Responder zusammen. Wobei Transmitter hier für Sender und Responder für Empfänger steht. Es handelt sich hiermit um eine kombinierte Sende- und Empfangseinheit. Der Transponder ist in den meisten Fällen direkt am zu identifizierenden Objekt befestigt und stellt den eigentlichen Datenträger im System dar.

Das Bild 2.1. zeigt eine schematische Übersicht der enthaltenen Komponenten.

Die Verbindung zwischen den beiden Hauptkomponenten erfolgt über die Luft-Schnittstelle. Die Darstellung zeigt ein System mit einem passiven Transponder, da das Lesegerät den

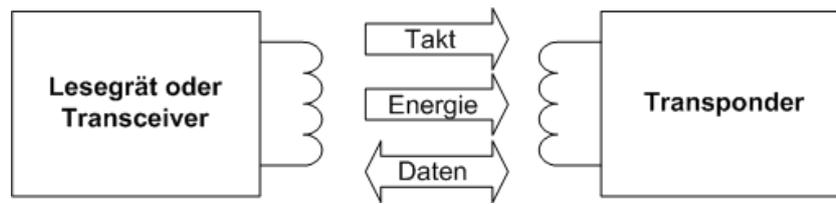


Bild 2.1.: Bestandteile eines RFID-Grundsystems

Transponder mit der notwendigen Energie versorgt, um ein Rücksenden der gespeicherten Daten zu bewerkstelligen. Auf den Unterschied zwischen einem aktiven und passiven Transponder wird in 2.1.2 eingegangen.

Welches die üblichen Frequenzbereiche sind, kann im Abschnitt 2.1.3 eingesehen werden und eine Beschreibung der zu übertragenden Daten ist unter 2.1.4 aufgeführt.

2.1.2. Energieversorgung von Transpondern

Eines der wesentlichen Unterscheidungsmerkmale bei der Beschreibung von Transpondern ist die Art der Energieversorgung.

Dabei wird zwischen zwei verschiedenen Versorgungsarten unterschieden. Verfügt der Transponder über eine eigene Energieversorgung, z.B. über eine Batterie, so spricht man von einem aktiven Transponder. Ist dies nicht der Fall, so handelt es sich um einen passiven Transponder. Bei dem, im Rahmen dieser Diplomarbeit, verwendeten Transponder handelt es sich um einen passiven Transponder.

Hierbei wird die Versorgungsenergie direkt aus dem Sendefeld des Lesegeräts entnommen. Das an der Empfangsspule ankommende Feld induziert eine Spannung, die mittels eines Kondensators für eine bestimmte Dauer gespeichert wird. Mittels dieser Kondensatorspannung wird dann der Transponder versorgt.

Dies bedeutet, dass der Transponder sich nur aktivieren lässt, wenn er sich im Sendebereich der Basisstation befindet. Daher die Bezeichnung als passiver Transponder. Bei aktiven Transpondern ist der Tag nicht auf das Sendefeld zur Energieversorgung angewiesen. Im Allgemeinen befindet sich diese Transponderart so lange im Ruhezustand, bis sie über ein bestimmtes Aktivierungssignal zum Senden aufgefordert wird.

2.1.3. Frequenzbereiche von RFID-Systemen

Die zum Einsatz kommenden Arbeitsfrequenzen bei RFID-Systemen lassen sich in vier Hauptgruppen einteilen. Dabei bewegt sich das Spektrum über den Low Frequency LF,

High Frequency HF, Ultra High Frequency UHF bis hin zum Mikrowellen Bereich. Die Einteilung erfolgt nach den Frequenzbereichen gemäß des Industrial, Scientific and Medical Band ISM Standard.

Die Tabelle 2.1. zeigt eine Übersicht der Frequenzspektren.

Bezeichnung	Frequenzbereich	ISM Frequenzen
LF	30 – 300kHz	< 135kHz
HF	3 – 30MHz	6, 78MHz; 13, 56MHz; 27, 125MHz; 40.680MHz
UHF	300MHz – 3GHz	433, 920MHz; 869MHz; 915MHz
Mikrowelle	> 3GHz	2, 45MHz; 5, 8MHz; 24, 125MHz

Tabelle 2.1.: Frequenzbereiche von RFID-Systemen

Das für diese Diplomarbeit entwickelte RFID-System arbeitet im LF-Bereich bei einer Arbeitsfrequenz von 125kHz.

Aus den unterschiedlichen Frequenzspektren ergeben sich typische maximale Reichweiten bei der Verwendung vom passiven Transpondern. Die Tabelle 2.2. stellt diese dar.

Frequenzbereich	Typische maximale Reichweite
LF	50cm
HF	3m
UHF	9m
Mikrowelle	> 10m

Tabelle 2.2.: Typische Reichweiten bei den Frequenzbereichen unter Verwendung eines passiven Transponders

Die in diesem Abschnitt verwendeten Angaben wurden der Quelle [RFID01] entnommen.

2.1.4. Codierung der zu übertragenen Daten

Die Datenübertragung zwischen dem Lesegerät und dem Transponder erfolgt über ein digital codiertes Signal. Es gibt verschiedene Arten der Signalcodierung, für diese Diplomarbeit wird die Methode der Manchester-Codierung angewendet.

Bei dieser Codierung wird mittels eines Flankenwechsels erkannt, ob sich bei dem zu übertragenen Bit um eine Eins oder Null handelt. Per Definition erfolgt jeweils mindestens ein Flankenwechsel pro übertragenem Bit. Hierbei entspricht der Wechsel von einem HIGH-Pegel zu einem LOW-Pegel einer logischen Eins. Soll eine Null übertragen werden, so ist dies über eine Flankenwechsel von LOW nach HIGH möglich.

Bei dieser Codierungsmethodik resultiert eine Verdopplung der Datenrate, da die Information über einen Flankenwechsel codiert wird. Als wesentlicher Vorteil bleibt festzuhalten, dass bei dieser Art der Datenübertragung eine Taktrückgewinnung möglich ist. Das Bild 2.2. stellt eine Übersicht über die am häufigsten verwendeten Codierungsarten dar.

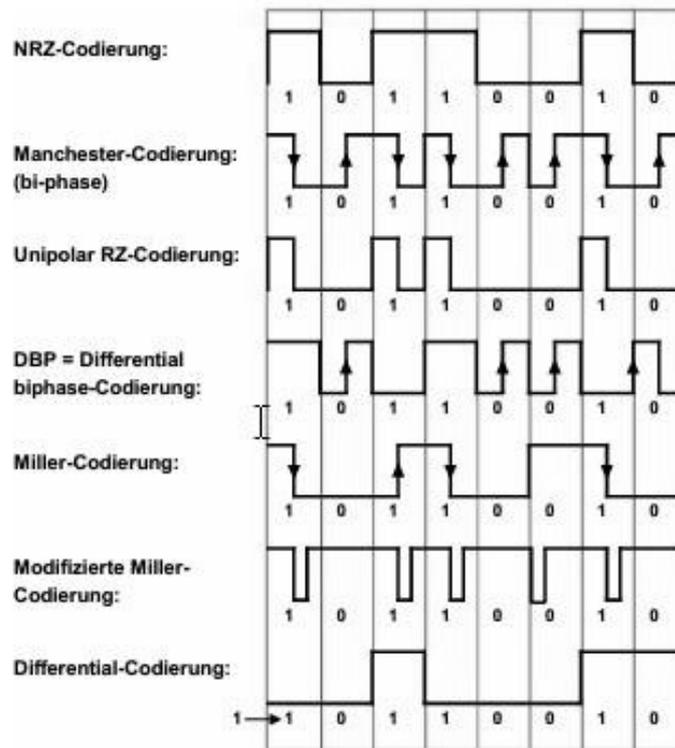


Bild 2.2.: Übersicht der häufig verwendeten Signalcodierungsarten [RFID02]

2.1.5. Modulation der zu übertragenen Daten

Das codierte Signal wird zur endgültigen Übertragung auf ein höherfrequentes Signal aufmoduliert. Im Rahmen dieser Diplomarbeit wird das Signal mittels einer Amplitudentastung, im Englischen Amplitude Shift Keying ASK, übertragen.

Bei dieser digitalen Übertragungsart wird über eine Änderung der Amplitude des Trägersignals die Information übertragen. Das Bild 2.3. zeigt ein per ASK übertragenes Signal.

Das Signal wird hierbei zwischen den beiden Zuständen \hat{u}_0 und \hat{u}_1 umgeschaltet. Diesen Vorgang bezeichnet man auch als Tastung, daher der Begriff Amplitudentastung.

Das Verhältnis zwischen diesen beiden Amplitudenzuständen wird als Tastgrad m be-

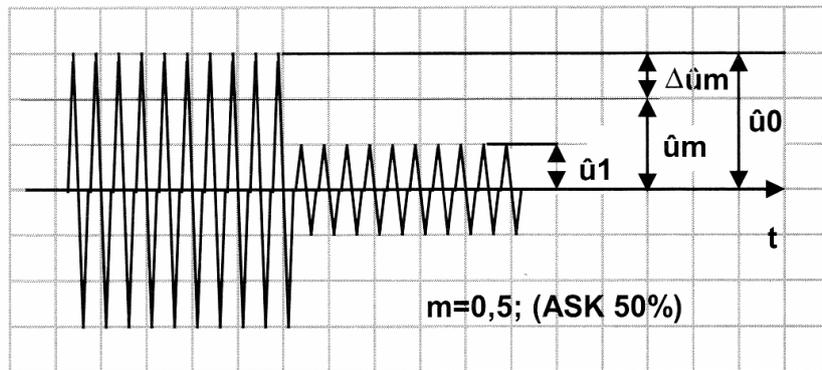


Bild 2.3.: Ein mittels ASK moduliertes Signal [RFID02]

zeichnet und lässt sich mittels des Quotienten:

$$m = \frac{\hat{u}_0 - \hat{u}_1}{\hat{u}_0 + \hat{u}_1} \quad (2.1)$$

bestimmen.

In Bild 2.3. ist somit ein Signal bei einem Tastgrad von $m = 50\%$ dargestellt.

Um eine Signaländerung möglichst einfach bestimmbar zu machen, wird bei RFID-Systemen häufig das Signal bei einem Tastgrad vom 100% übertragen. Hierbei wird das Signal zwischen den beiden Zuständen 0 und \hat{u}_0 umgeschaltet. Eine Datenübertragung mit diesem Tastgrad wird auch als On Off Keying OOK bezeichnet. Bei der Übertragung der Daten zwischen dem Transceiver und dem Transponder wird in dieser Arbeit ebenfalls diese Methode verwendet.

Neben dieser Modulationsart existieren noch weitere Verfahren, da diese aber nicht angewendet werden, wird auf eine weitere Beschreibung verzichtet. Eine weiterführende Beschreibung der einsetzbaren Modulationsverfahren ist unter [RFID02] aufgeführt.

2.1.6. Normierung von RFID-Systemen

Die Rahmenbedingungen für den industriellen Einsatz von RFID-Systemen werden in diversen Normen beschrieben. Diese Normen werden von den Gremien der International Organization for Standardization ISO spezifiziert.

Das Bild 2.4. zeigt eine Übersicht der Normierungszuordnungen.

Die Tabelle 2.3. fasst die Vorschriften zum Betrieb von RFID-Systemen zusammen, die von besonderer Relevanz für die Themenbereiche der hier vorgestellten Diplomarbeit sind.

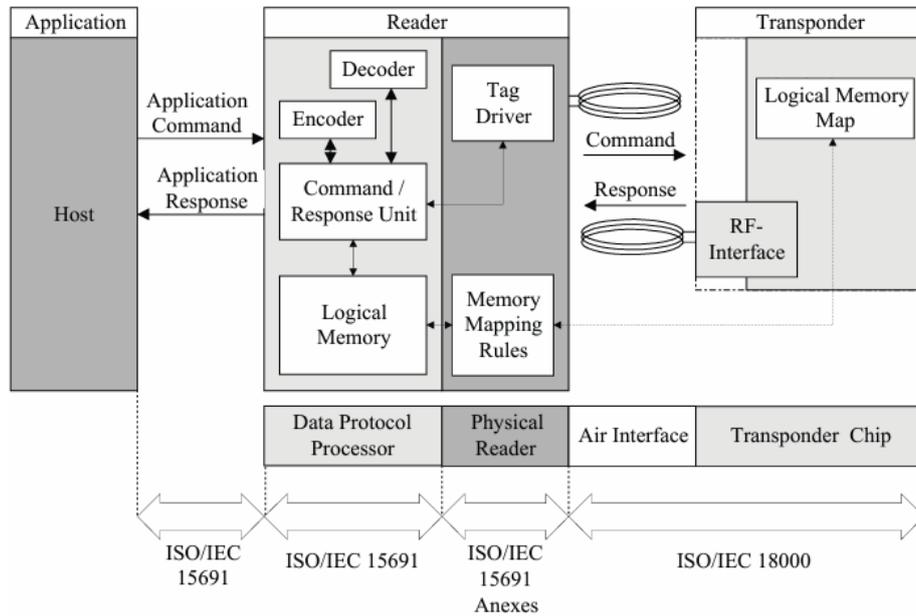


Bild 2.4.: Übersicht der Normen für RFID-Systeme [RFID02]

Standard	Bezeichnung des Standards
ISO 15961	RFID for Item Management: Tag functional commands
ISO 15962	RFID for Item Management: Data Syntax
ISO 18000	RFID for Item Management: Air Interface
ISO 18000-1	Generic Parameter for Air Interface Communication
ISO 18001-2	Parameters for Air Interface Communication below 135kHz

Tabelle 2.3.: Tabelle der relevanten Normen für RFID-Systeme im Frequenzbereich von 125kHz

2.2. Grundlagen zum Resonanzabgleich

2.2.1. Abgleichmethoden aus Vorarbeiten

In der Diplomarbeit [DIPL01] von Tobias Krannich werden bereits einige Abgleichverfahren vorgeschlagen, da die Arbeit als Vorarbeit zu betrachten ist, werden die dort vorgestellten Verfahren an dieser Stelle kurz umrissen.

Abgleich durch Amplitudenmessung

Bei diesem Verfahren wird die optimale Kapazität für den Antennenschwingkreis mittels einer Spannungsmessung am Mikrocontroller ermittelt. Es wird eine iterative Bestimmung der Spannung angewendet, mit dem Ziel den maximalen Spannungswert zu finden. Der eingesetzte Kapazitätswert ist hierbei abhängig vom gemessenen Spannungswert. Der Kondensatorwert bei dem sich das Spannungsmaximum ergibt, wird als Optimum betrachtet und wird für den weiteren Betrieb verwendet. Die Abstufung der Kondensatorwerte erfolgt in binären Schritten, dies bedeutet, dass der nächst höhere Kondensatorwert immer doppelt so groß ist wie der seines Vorgängers.

Im Bild 2.5. ist eine Darstellung des Blockschaltbilds zum Messverfahren aufgeführt.

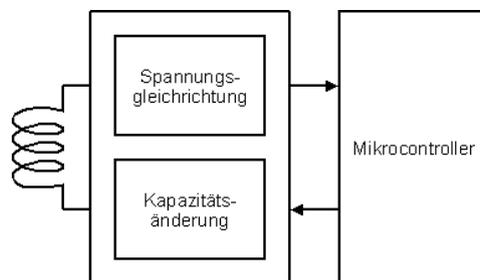


Bild 2.5.: Blockschaltbild zur Methode der Amplitudenmessung

Diese Messmethode wird bei der Realisierung des Readers berücksichtigt.

Abgleich durch Phasenmessung

Auch bei diesem Verfahren wird die Kapazität im Schwingkreis variiert. Die Kapazitätsänderung erfolgt mittels eines Phasenvergleichs zwischen der anregenden Spannung und der gemessenen Spannung im Antennenschwingkreis. Der Vergleich erfolgt mit Hilfe des Komparatoreingangs am eingesetzten MSP-Mikrocontroller. Das Verfahren ist grundsätzlich dem unter Abschnitt 2.2.2. erläuterten Verfahren ähnlich.

Das Bild 2.6. zeigt eine Übersicht zur Messmethode.

Auf die in [DIPL01] vorgestellte Schaltung zu dieser Abgleichsmethode wird, aufgrund

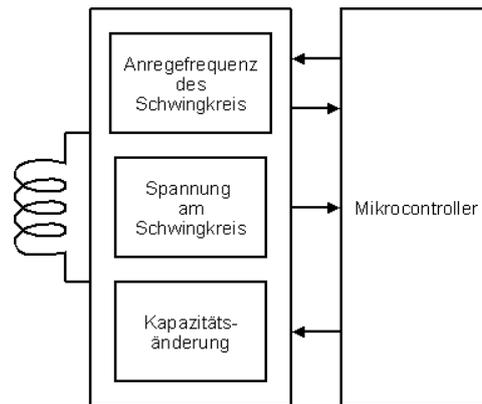


Bild 2.6.: Blockschaltbild zur Methode der Phasenmessung

der zu geringen Taktfrequenz des verwendeten Mikrocontrollers, nicht weiter zurückgegriffen (vgl. 2.2.3.).

Ableich durch Frequenzanpassung

Bei dieser Methode wird die Frequenz der anregenden Spannung verändert. Die dem Antennenschwingkreis zugeführte Spannung wird durch einen Mikrocontroller erzeugt und kann daher einfach geändert werden. Eine Resonanzfrequenzänderung ist somit durch eine Variation, der vom Controller ausgegebenen Rechteckspannung erreichbar.

Da mit diesem Verfahren direkt auf die Anregfrequenz und hierbei auf die angestrebte Sendefrequenz eingewirkt wird, wird diese Methode bei der hier aufgeführten Implementierung nicht weiter berücksichtigt.

Wobei anzumerken ist, dass es sich hierbei nicht um ein völlig unübliches Verfahren handelt. Laut [RFID02] gibt es unterschiedliche Verfahren die zur Antikollision verwendet werden können. Der Begriff Antikollision beschreibt den Umstand, dass falls sich mehrere Transponder im Einflussbereich des Readerfelds befinden, diese zu einem gleichen Zeitpunkt ihre Daten an das Lesegerät senden könnten. Somit wäre eine eindeutige Zuordnung des Sendedaten zu einem bestimmten Transponder nicht ohne weiteres möglich. Eines dieser Verfahren arbeitet nach dem Verfahren des Frequenzmultiplexing. Bei dieser Methode sendet das Lesegerät auf leicht unterschiedlichen Frequenzen. Die jeweiligen Transponder wären je auf eine dieser Sendefrequenzen abgestimmt. Die Kommunikation zwischen dem Reader und dem Transponder würde nur dann zustande kommen, wenn die jeweils erforderliche Frequenz eingehalten wird.

Diese Methode ist mit einem hohen Aufwand bei der Realisierung der Hardware verbunden. Der Reader muss so beschaffen sein, dass die unterschiedlichen Frequenzen erzeugt werden können. Auch jeder Transponder müsste in seinem Aufbau etwas unterschiedlich sein, um lediglich bei einer bestimmten Frequenz angeregt zu werden.

Die Methode des Frequenzmultiplexing findet ihre Anwendung nur in wenigen Spezialbereichen. Dies ist durch den erhöhten Hardwareaufwand begründet. Die hier vorgestellte Implementierung hat zum Ziel, überhaupt ein funktionsfähiges Element eines RFID Systems zu entwickeln. Die Realisierung dieses speziellen Anwendungsfalls liegt daher nicht im Fokus der Arbeit.

2.2.2. Vorstellung der Messmethode zur Einstellung der Resonanzfrequenz nach Giesler, Böh und Tobergte

Die vorgestellte Messmethode wurde aus der Patentanmeldung [PAT01] entnommen und wird hier in Kürze erläutert.

Dargestellt wird eine Schaltungsanordnung über die es möglich ist zu erkennen, ob ein Schwingkreis sich im abgestimmten Zustand befindet. Als Hauptkriterium für einen Abgleich wird die Phasenlage der Generatorspannung zur Phasenlage der Spannung am Referenzpunkt innerhalb des Antennenschwingkreises betrachtet. Als Generatorspannung wird die Spannung bezeichnet, die dem Schwingkreis zur Anregung zugeführt wird. Der Referenzpunkt liegt zwischen dem Kondensator und der Spule im Schwingkreis. Der Schwingkreis wird als abgestimmt erachtet, wenn die Phasenlage zwischen den beiden Spannungen genau 90° beträgt.

Bei der Offenlegungsschrift wird die Annahme getroffen, dass es sich bei der Generatorspannung um eine Rechteckspannung handelt. Die ermittelte Spannung am Referenzpunkt hat einen sinusförmigen Verlauf. Um die beiden Spannungen bezüglich ihrer Phasenlage bewertbar zu machen, ergibt sich die Notwendigkeit, die Sinusspannung in eine Rechteckspannung zu überführen. Aus diesem Grund wird die Referenzspannung einem Spannungskomparator zugeführt. Dieser übernimmt die Aufgabe der Umsetzung. Der eigentliche Spannungsvergleich erfolgt im Phasenkomparator. Die Generatorspannung wird zum Vergleich um 90° verschoben. Es wird vorgeschlagen, dass am Ausgang des Komparators beispielsweise ein High-Pegel ausgegeben wird, wenn die Referenzspannung der Generatorspannung vorausseilt. Sollte die Generatorspannung der Referenzspannung vorausseilen, so wird vorgeschlagen, dies mittels eines Low-Pegels am Komparatorausgang kenntlich zu machen.

Es wird weiterhin angeregt, das Signal des Komparators einem Mikrocontroller zuzuführen. Zusätzlich wird der Referenzpunkt mit einer Abstimmereinheit verbunden. Welche aus Abstimmkapazitäten und einer Abstimminduktivität besteht. Letztere wird dauerhaft mit dem Antennenschwingkreis verbunden. Mittels dieser Hardwareanordnung soll es ermöglicht werden, eine Verschiebung der Resonanzfrequenz in die positive, sowie die negative Richtung zu beeinflussen.

Die Abstimmkondensatoren sind über elektrisch beeinflussbare Schalter zu- oder ab-

schaltbar. Abhängig vom Ausgangssignal des Phasenkomparators, werden vom Mikrocontroller die Schalterstellungen für die Abstimmkondensatoren vorgegeben. Durch die Variation der Kondensatoren wird die Resonanzfrequenz des Schwingkreises unmittelbar beeinflusst. Die Änderung der Schalterstellung wird solange wiederholt, bis der Antenschwingkreis abgestimmt ist.

Das Blockschaltbild zur beschriebenen Messmethode ist in Bild 2.7. dargestellt.

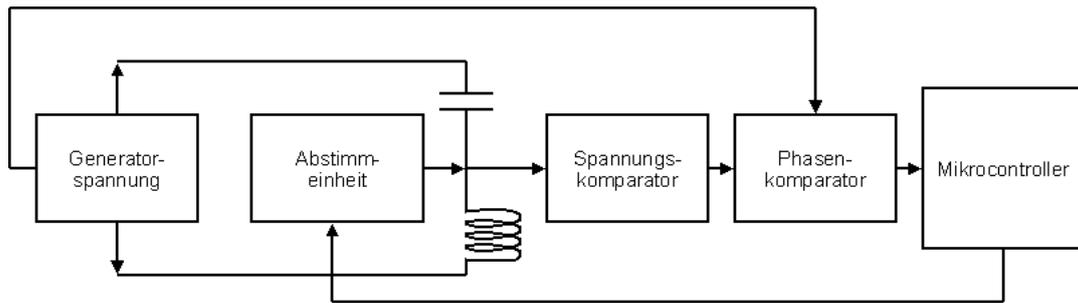


Bild 2.7.: Blockschaltbild zur Messmethode nach Giesler, Böh und Tobergte

Die hier beschriebene Messmethode stellt das Fundament für die weitere Implementierung der Resonanzabstimmung dar. Sie wird in leicht modifizierter Art angewendet.

2.2.3. Zeitliches Auflösungsproblem bei Verwendung des MSP430 Mikrocontroller

Aus dem Handbuch [MSP01] des verwendeten Controllers A.1 lässt sich entnehmen, dass es möglich ist, einen externen Quarzbaustein zur Taktgenerierung zu verwenden. Die Arbeitsfrequenz kann hierbei im Bereich von $450kHz$ bis $8MHz$ liegen. Für die weiteren Betrachtungen wird davon ausgegangen, dass der Controller mit einem Quarz von $8MHz$ betrieben wird.

Aus dieser Wahl ergibt sich eine zeitliche Auflösung pro Takt von:

$$t_t = \frac{1}{f_t} = \frac{1}{8MHz} = 0,125\mu s \quad (2.2)$$

Die Sendefrequenz f_s der zu messenden Signale beträgt $125kHz$, somit ergibt sich eine Zeitdauer pro Periode von:

$$t_s = \frac{1}{f_s} = \frac{1}{125kHz} = 8\mu s \quad (2.3)$$

Die Anzahl n_{takt} der Taktzyklen des MSP430 pro Periodendauer des Sendesignals beträgt:

$$n_{takt} = \frac{t_s}{t_t} = \frac{8\mu s}{0,125\mu s} = 64 \quad (2.4)$$

Dieses Ergebnis bedeutet, dass dem Controller während der vollen Signalperiode Zeit für 64 Operationsschritte zur Verfügung steht. Diese Schrittzahl ist entschieden zu gering, um einen gesicherten und funktionalen Programmablauf gewährleisten zu können. Es kann gezeigt werden, dass sich die Anzahl der zur Verfügung stehenden Taktschritte noch weiter verringert.

Die eigentliche Messaufgabe besteht darin, den Phasenunterschied zwischen der anregenden Spannung und der Phasenlage des Spannungsverlaufs im Schwingkreis zu detektieren. Der Schwingkreis gilt als abgestimmt, wenn die Phasenlage bei 90° Unterschied liegt.

Die vollen $8\mu s$ Zeitdauer pro Signalperiode entsprechen einer Phasenänderung von 360° . Da es gefordert ist, eine Phasenlage von 90° zu erkennen, ergibt sich ein Teilverhältnis von $\frac{360^\circ}{90^\circ} = 4$. Eine Umrechnung dieses Verhältnisses in eine Zeiteinheit ergibt eine Dauer von $\frac{8\mu s}{4} = 2\mu s$.

Diese Zeitdifferenz bedeutet in Taktzyklen für den MSP, dass sich die Anzahl der möglichen Takte zu:

$$n_{takt2\mu s} = \frac{t_{diff}}{t_t} = \frac{2\mu s}{0,125\mu s} = 16 \quad (2.5)$$

verringert.

Innerhalb dieser Taktanzahl müssen alle Arbeitsschritte vom Controller abgearbeitet sein. Geht man von der Annahme einer Messung über eine Interruptroutine aus, so werden laut [MSP01] 6 Takte benötigt, um einen Interrupt zu akzeptieren. Um aus dieser Routine ins Hauptprogramm zurückzuspringen, werden 5 Taktschritte benötigt. Es ergeben sich daher bereits 11 Zyklen für den Ein- bzw. Austritt aus dem Interrupt. Die verbleibenden 5 Takte lassen keinen Handlungsspielraum für eine gesonderte Abfrage im Interrupt.

Aus diesem Umstand kann der Controller nicht für die Messaufgabe verwendet werden. Daher wird die Aufgabe der Messung der Phasendifferenz in ein Complex Programmable Logic Device CPLD A.3 ausgelagert. Dieses Bauteil lässt sich mittels eines externen Oszillators deutlich höher takten. Auf die eigentliche Umsetzung der Messung wird im weiteren Verlauf eingegangen.

2.3. Grundlagen zum Reader

2.3.1. Reihenschwingkreis

Das Sendefeld des Readers wird mittels eines Reihenschwingkreises erzeugt. Dieser besteht aus einer Induktivität und einer Kapazität. Beide Bauelemente sind bei dieser Art des Schwingkreises in Reihe geschaltet.

Ausgangspunkt für die Dimensionierung der Bauteile ist die Formel zur Berechnung der Resonanzfrequenz:

$$f_0 = \frac{1}{2\pi\sqrt{L_{ANT}C_{RES}}} \text{ [Hz]} \quad (2.6)$$

Als Induktivitätswert wird der bereits aus [STUD01] verwendete Wert von $725\mu H$ verwendet. Mittels dieser Vorgabe und dem geforderten Wert von $125kHz$ für die Resonanzfrequenz lässt sich C_{RES} mittels der Formel:

$$C_{RES} = \frac{1}{(2\pi f_0)^2 L_{ANT}} \text{ [F]} \quad (2.7)$$

bestimmen.

Es ergibt sich ein Kapazitätswert von $2,2nF$ für C_{RES} .

Das Bild 2.8. zeigt die Hardwareanordnung für den Reihenschwingkreis.

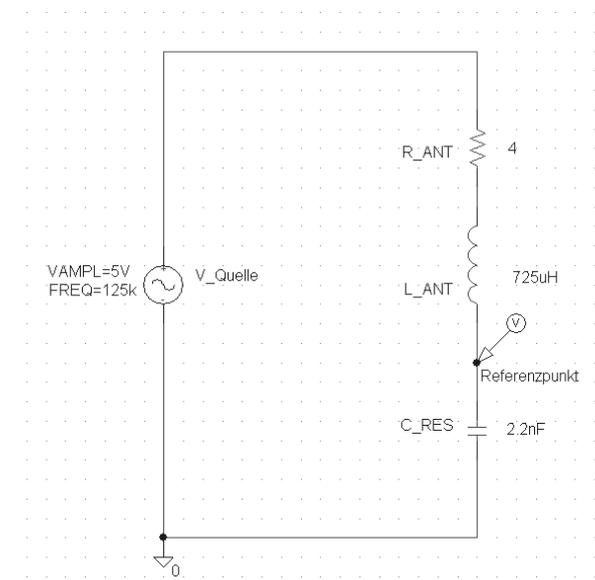


Bild 2.8.: Schaltungsanordnung zum Reihenschwingkreis

Zur Verdeutlichung wird die Schaltung mittels der Software PSPICE B.5 simuliert. Das Bild 2.9. zeigt den Frequenzgang des Reihenschwingkreises.

Der Parameter R_{ANT} steht hier für den ohmschen Widerstand der Antennenspule. Das Einfügen eines Widerstands in den Antennenschwingkreis wirkt sich direkt auf die Band-

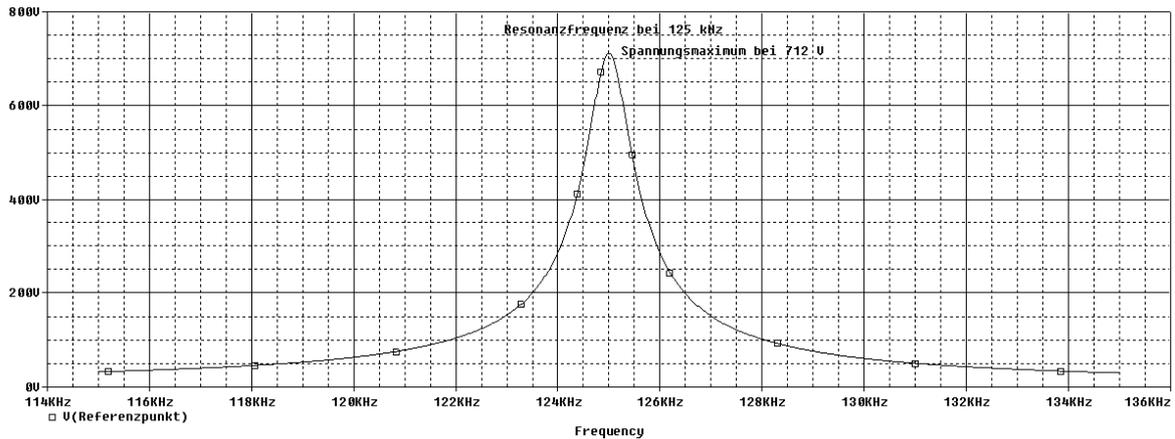


Bild 2.9.: Frequenzgang des simulierten Reihenschwingkreises

breite und die Güte des Schwingkreises aus.

Die Bandbreite B kann über den Quotienten:

$$B = \frac{R}{2\pi L_{ANT}} [Hz] \quad (2.8)$$

errechnet werden.

Die Güte Q ergibt sich aus dem Verhältnis der Resonanzfrequenz zur Bandbreite:

$$Q = \frac{f_0}{B} \quad (2.9)$$

In diesem Grundlagenbeispiel ergibt sich eine Bandbreite von:

$$B = \frac{4\Omega}{2\pi 725\mu H} = 878,1 Hz \quad (2.10)$$

Die Güte der Beispielschaltung beträgt:

$$Q = \frac{125 kHz}{0,878 kHz} = 142,4 \quad (2.11)$$

Mit Hilfe der Güte kann auf die Ausgangsspannung im Resonanzfall geschlossen werden. Hier ergibt sich die Ausgangsspannung bei einer Eingangsspannung von $U_e = 5V$ und der ermittelten Güte $Q = 142,4$ zu:

$$U_a = Q \cdot U_e = 5V \cdot 142,4 = 712V \quad (2.12)$$

Dieses Ergebnis lässt sich ebenfalls in Bild 2.9. ablesen.

2.3.2. Schwingkreisanregung

Bei der Realisierung des Readers wird der Antennenschwingkreis mittels zweier gegenphasig angesteuerten Bustreiber angeregt.

Das Bild 2.10. zeigt einen beispielhaften Aufbau zur Schwingkreisanregung.

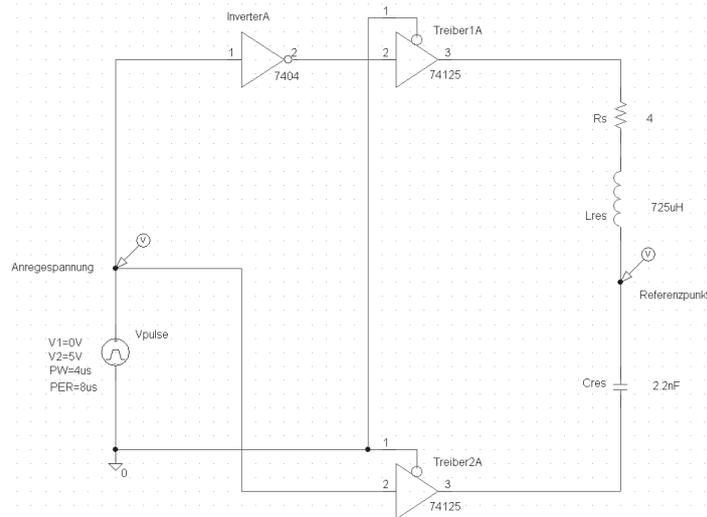


Bild 2.10.: Beispielhafte Schaltungsanordnung zur Schwingkreis Anregung

Die Anregung der Bustreiber erfolgt mittels einer Rechteckspannung. Wobei ein Treiberbaustein direkt mit der Rechteckspannung beaufschlagt wird und der Zweite mit der invertierten Spannung. Der hiermit gewonnene Spannungsverlauf dient zur Anregung der im Antennenschwingkreis vorhandenen Bauelemente.

Es resultiert hieraus eine zyklische Umladung der Spannung am Kondensator und der Induktivität.

Das Bild 2.11 zeigt die beiden Spannungsverläufe an der simulierten Schaltung.

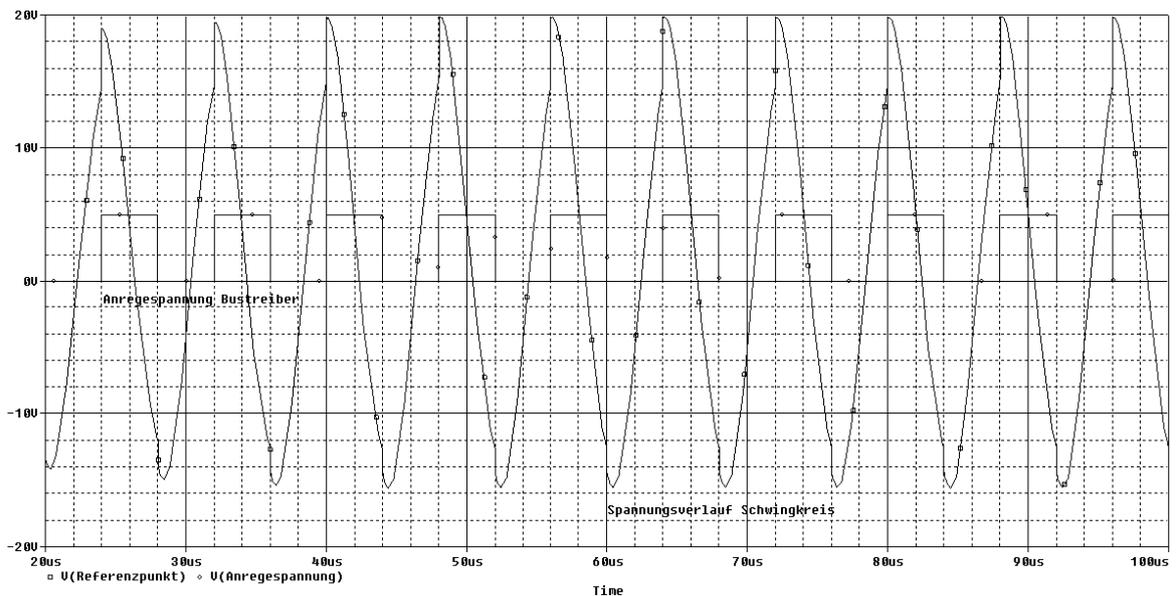


Bild 2.11.: Darstellung der Spannungsverläufe der Beispielschaltung zur Schwingkreis-anregung

In dem hier vorgestellten Simulationsergebnis wird nur die nichtinvertierte Anregespannung dargestellt. Der Verlauf der resultierenden Spannung wird im wesentlichen durch die Dimensionierung des Kondensators und der Spule beeinflusst. Auf die genaue Dimensionierung wird in den später folgenden Punkten, der Implementierung des Readers, vertiefend eingegangen.

2.3.3. Elektromagnetisches Feld

Die Spule des Readers fungiert als Antenne über die ein elektromagnetisches Feld H emittiert wird. Mittels dieses Feldes werden die Daten und bei passiven Transpondern ebenfalls die Energie übertragen.

Die Stärke des ausgesendeten Feldes hängt von einigen Parametern ab und lässt sich nach der folgenden Formel berechnen [RFID04]:

$$H_R(x) = \frac{I_R N_R r_R^2}{2(r_R^2 + x^2)^{3/2}} \left[\frac{\text{A}}{\text{m}} \right] \quad (2.13)$$

Die Variable x beschreibt das Maß des Abstands zur Antenne. I_R steht für den Strom der über die Antenne fließt. Die Anzahl der Spulenwicklungen wird über N_R angegeben und der Radius r_R der Antennenspule spielt ebenfalls eine Rolle bei der Berechnung. Bild 2.12. zeigt den Feldverlauf für 3 unterschiedliche Spulenradien. Der Strom I wurde mit 50mA definiert und die Wicklungszahl N zu 30 Windungen angenommen.

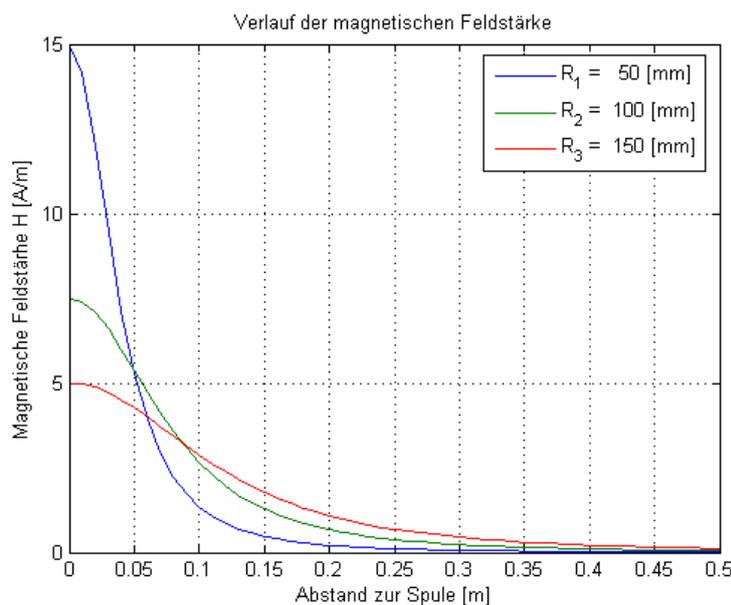


Bild 2.12.: Feldverlauf von $H(x)$ bei 3 unterschiedlichen Spulenradien

Es ist ersichtlich, dass die Feldstärke in Abhängigkeit zur Entfernung schnell abfällt.

Bei Verwendung eines kleineren Spulenradius lässt sich eine Steigerung der anfänglichen Feldstärke bewirken. Es ist aber zu beachten, dass die Feldstärke hierbei, bei steigender Entfernung, auch schneller wieder abfällt. Dieses Verhalten stellt sich bei R_1 ein.

Am Verlauf des Graphen von R_3 kann abgeleitet werden, dass bei einem vergleichsweise größeren Spulenradius der maximale Feldstärkewert zwar nicht so hoch ist, aber dafür ist der Feldstärkeabfall nicht ganz so rapide. Generell lässt sich erkennen, dass einstellige Feldstärkewerte zu erwarten sind.

Betrachtet man die Entfernung zur Sendeantenne als konstant, so lässt sich der jeweilige optimale Spulenradius zur Entfernung bestimmen. An diesem Punkt stellt sich das Maximum der magnetischen Feldstärke ein. Dieser Punkt kann durch eine Differenzierung von (4.6) nach r_R ermittelt werden.

Als Ergebnis erhält man:

$$r_R = x\sqrt{2} \quad (2.14)$$

Der optimale Spulenradius stellt sich somit bei einer Entfernung vom $\sqrt{2}$ -fachen zum gewünschten maximalen Abstand ein. Oder anders ausgedrückt, sollte der Antennenradius ca. 40% größer sein als der maximal angestrebte Leseabstand.

Das Bild 2.13. stellt den Verlauf der magnetischen Feldstärke bei einem festen Abstand von $x = 10\text{cm}$ dar. Die Werte für den Strom und die Wicklungszahl bleiben unverändert.

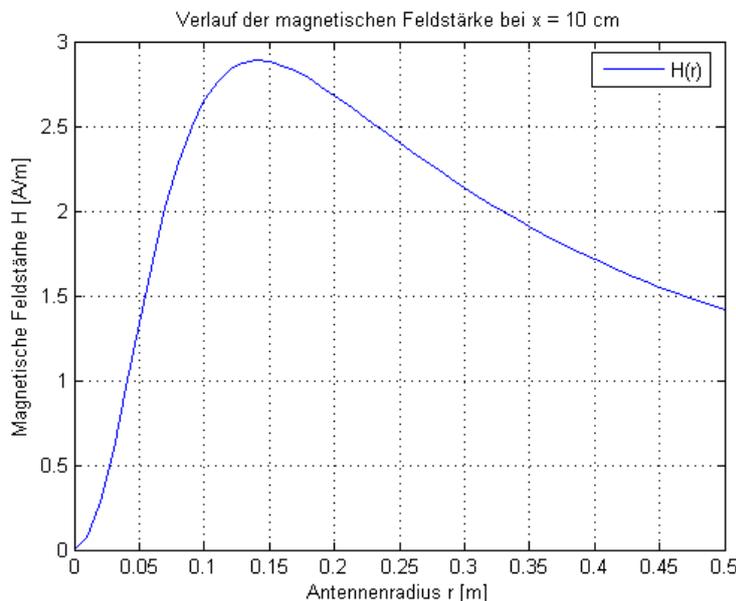


Bild 2.13.: Feldverlauf von $H(r)$ bei einer Entfernung von 10cm zur Antenne

Die Feldstärkemaximum stellt sich hier bei einem Radius von $r_R = 10\text{cm}\sqrt{2} = 14,1\text{cm}$ ein.

Bei der weiteren Implementierung kann auf eine Optimierung des Spulenradius nicht in

besonderer Weise eingegangen werden. Da vielmehr die äußeren Abmaße der verwendeten Readerbox den Hauptausschlag für die Dimensionierung der Antennenspulen geben.

2.3.4. Transformatorprinzip

Wie bereits beschrieben, beziehen passive Transponder die notwendige Versorgungsenergie aus dem elektromagnetischen Feld, welches vom Reader ausgeht. Mittels des Transformatorprinzips kann der Vorgang der Energieübertragung näher beschrieben werden.

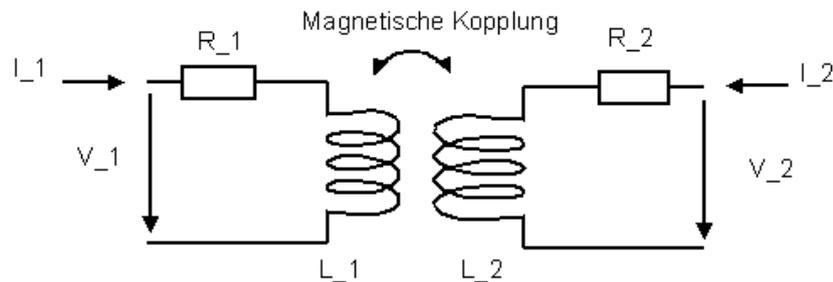


Bild 2.14.: Schematische Darstellung des Transformatorprinzips

Das Bild 2.14. zeigt eine schematische Darstellung des Transformatorprinzips.

Die Antennenspule des Readers ist hierbei als L_1 dargestellt und die Spule L_2 repräsentiert die Antennenspule des Transponders. Die beiden Widerstände R_1 und R_2 stehen für den jeweiligen ohmschen Anteil der Spule. Beide Spulen sind magnetisch gekoppelt. Die Readerseite wird als Primärseite und die Transponderseite als Sekundärseite des Transformators betrachtet.

Aus der, an der Readerspule anliegenden Spannung, resultiert ein Stromfluss, mittels diesem das elektromagnetische Feld induziert wird. Dieses Feld erzeugt an der Transponderspule ebenfalls eine Spannung. Diese Spannung ist gleichbedeutend mit der, dem Transponder zur Verfügung stehenden Versorgungsspannung.

Eine Schaltungsanordnung mit real verwendeten Bauteilgrößen zeigt das Bild 2.15.

Die Readerseite ist hier als Reihenschwingkreis realisiert. Die Transponderseite besteht aus einem Parallelschwingkreis. Der Grund für diese Art der Realisierung liegt daran, dass sich beim Reihenschwingkreis der Impedanzwert bei Erreichen der Resonanzfrequenz auf sein Minimum reduziert. Hieraus ergibt sich ein maximaler Stromfluss. Beim Parallelschwingkreis hingegen ergibt sich der Maximalwert der Impedanz um den Resonanzpunkt des Schwingkreises. Es folgt hierdurch ein minimaler Stromflusswert.

Die Werte für die Transponderspule und dem Antennenwiderstand stimmen mit denen des verwendeten Transponders überein. Eine generelle Beschreibung des verwendeten

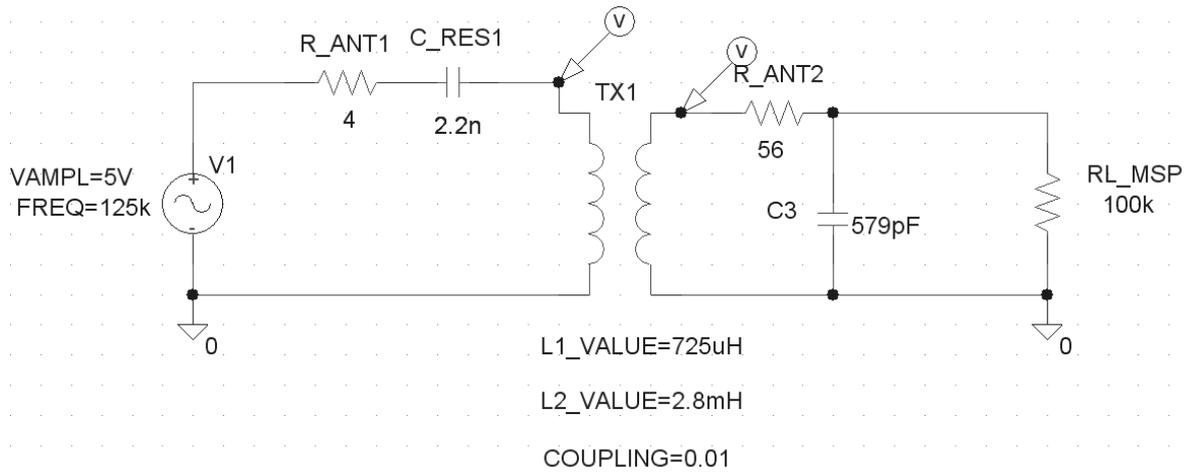


Bild 2.15.: Schaltungsanordnung zum Transformatorprinzip

Transponders erfolgt in A.2.

Das Simulationsergebnis der dargestellten Schaltung zeigt das Bild 2.16.

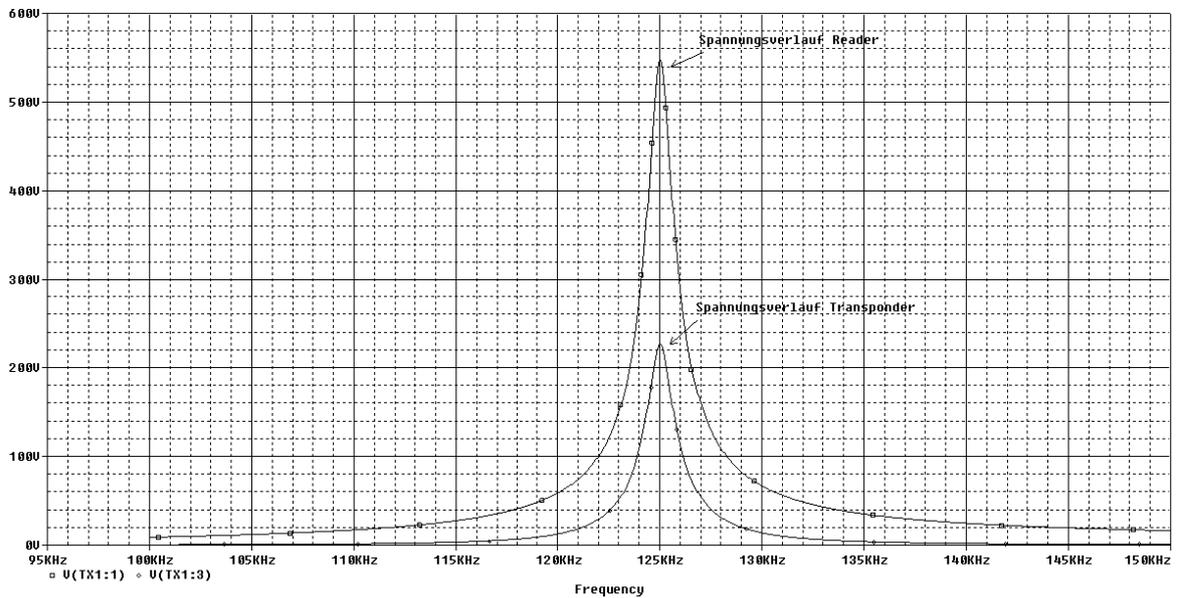


Bild 2.16.: Spannungsverlauf der simulierten Transformatorschaltung

Die Spannung am Transponder ist deutlich kleiner als die Spannung am Reader. Die Spannungsübersetzung wird von der Gegeninduktivität M beeinflusst. Diese kann mittels:

$$M = k\sqrt{L_1 L_2} \text{ [H]} \quad (2.15)$$

bestimmt werden [RFID04].

Der Faktor k steht für den Kopplungsfaktor. Per Definition liegt k im Bereich von $0 \geq k \leq 1$. In der vorausgegangenen Simulation wurde k zu 0.01 definiert.

Berechenbar wird der Kopplungsfaktor mittels der Formel [RFID02]:

$$k(x) = \frac{r_R^2 r_T^2}{\sqrt{r_R r_T} \sqrt{(r_R^2 + x^2)^3}} \quad (2.16)$$

Betrachtet man die Formel (2.15) so wird ersichtlich, dass der Kopplungsfaktor k nur von den geometrischen Eigenschaften der beiden Antennenspulen und der Entfernung x abhängt.

Zur Verdeutlichung wird in Bild 2.17. der Verlauf von $k(x)$ für drei unterschiedliche Readerspulenradien dargestellt. Als Spulenradius wird der für den Transponder verwendete Wert von 12mm angewendet.

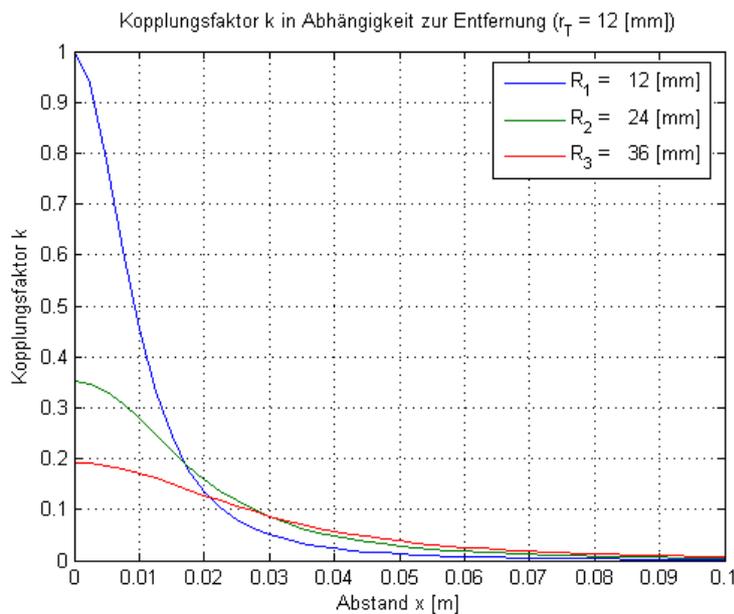


Bild 2.17.: Verlauf des Kopplungsfaktors k für 3 Spulenradien

Der Wert für den Kopplungsfaktor fällt ebenfalls stark mit zunehmender Entfernung zur Antennenspule. Als Entfernungsstrecke wird hier der Bereich von $0 - 10\text{cm}$ betrachtet. Die Readerspulenradien werden so gewählt, dass die beiden Antennen zuerst gleich groß sind und anschließend wird der Readerantennenradius zunächst verdoppelt und daraufhin verdreifacht. Mit ansteigender Vergrößerung des Antenneradius der Readerspule verringert sich der anfängliche Wert für den Kopplungsfaktor. Dies lässt erkennen, dass je mehr sich die Radien gleichen, desto höher wird der Kopplungsfaktorwert.

Allerdings liegt der Wert für alle drei Radien bereits ab der Entfernung von 3mm unterhalb von 0,1. Bei den gleichgroßen Radien fällt der Abfall steiler aus, als bei den vergrößerten Radien.

Es wird hieraus abgeleitet, dass Kopplungsfaktoren von $\leq 0,1$ beim Transponderbetrieb am Reader zu erwarten sind.

Die magnetische Kopplung der beiden Spulen fällt daher vergleichsweise gering aus. Dies liegt darin begründet, dass das Spulenpaar über das Medium der Luft miteinander gekoppelt ist. Hier zeigt sich der Hauptunterschied zu den Transformatoren, die beispielsweise zu Spannungsumsetzung verwendet werden. Bei dieser Art von Transformatoren sind die beiden vorhandenen Spulen über einen gemeinsamen Eisenkern miteinander verbunden. Die magnetische Leitfähigkeit von Eisen liegt deutlich höher als die von Luft oder Sauerstoff. Somit lässt sich ein höherer Grad der Kopplung und daraus resultierend, eine bessere Spannungsübertragung erreichen.

Um diesen Umstand genauer zu beschreiben, wird in der Literatur der Begriff der magnetischen Flussdichte B genannt.

Eine Beziehung der magnetischen Flussdichte B zum resultierenden elektromagnetischen Feld H lässt sich über:

$$B = \mu H \text{ [T]} \quad (2.17)$$

beschreiben.

Der Faktor μ wird als magnetische Permeabilität oder magnetische Leitfähigkeit bezeichnet. Dieser Faktor setzt sich aus magnetischen Leitfähigkeit im Vakuum μ_0 und der Leitfähigkeit des übertragenden Mediums μ_r zusammen.

Es folgt daher:

$$\mu = \mu_0 \mu_r \quad (2.18)$$

Bei μ_0 handelt es sich um eine physikalische Konstante, deren Wertigkeit mit:

$$\mu_0 = 4\pi 10^{-7} \left[\frac{\text{Vs}}{\text{Am}} \right] \quad (2.19)$$

bezeichnet wird.

Bei μ_r handelt es sich um eine dimensionslose Größe. Die Permeabilität von Sauerstoff wird mit ≈ 1 angegeben. Wohingegen es sich bei Eisen um einen ferromagnetischen Werkstoff handelt, dieser hat eine Permeabilitätszahl im Bereich von $\mu_r = 300 - 10.000$. Es ergibt sich daher bei dem Einsatz von Eisen als verbindendes Medium ein verstärkender Effekt. Dieser resultiert durch die höhere magnetische Leitfähigkeit des Werkstoffs. Bei der Übertragung über die Umgebungsluft entfällt dieser verstärkende Faktor. Der wirksame Anteil der magnetischen Flussdichte B ergibt sich zu:

$$B = \mu_0 H \quad (2.20)$$

Über die Betrachtung dieser physikalischen Eigenschaften, lässt sich der geringe Wert der magnetischen Kopplung erklären.

3. Implementierung des Readers

3.1. Darstellung der Systemkomponenten

Das Lesegerät besteht aus mehreren Systemkomponenten. Das Bild 3.1 stellt diese in einer Übersicht dar.

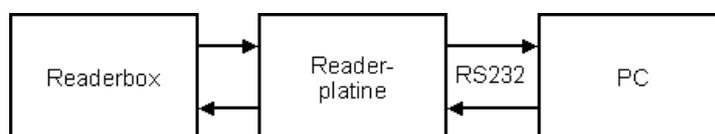


Bild 3.1.: Darstellung der Systemkomponenten

Die beiden Hauptelemente sind die Readerbox und die Readerplatine. Es ist vorgesehen, dass die Steuerung des Lesegerätes mittels eines externen Rechners erfolgt. Die Verbindung zur Readerplatine kann über die serielle Schnittstelle des verwendeten PCs hergestellt werden.

3.2. Beschreibung der Readerbox

Bei der Realisierung des Readers wird auf die bereits in [STUD01] verwendete Readerbox zurückgegriffen.

3.2.1. Räumliche Abmaße der Readerbox

Die zur Verfügung stehende Holzbox bildet den äußeren Rahmen für die räumliche Ausdehnung des Arbeitsbereichs des zu entwerfenden Readers. Beim Entwurf des Lesegeräts wird bewusst auf das Material Holz zurückgegriffen. Dies geschieht um störende Einflüsse, welche von metallischen Werkstoffen ausgehen würden, von vornherein zu minimieren.

Die Bemaßung der Readerbox wird in Bild 3.2. angegeben.

Für die weitere Betrachtung wird jeder der 6 Seitenflächen der Readerbox eine Zuordnungszahl zugewiesen. Das Bild 3.3. und die Tabelle 3.1. geben eine weitere Auskunft

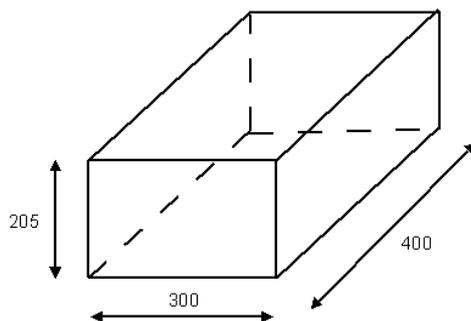


Bild 3.2.: Räumliche Abmaße der Readerbox

über die Zuordnung.

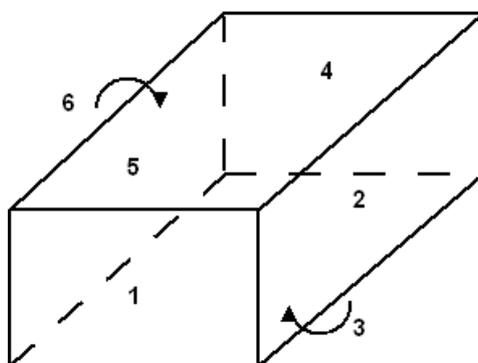


Bild 3.3.: Verknüpfung der Zuordnungszahl mit den Seitenflächen der Readerbox

Benennung	Seitenfläche
Vorderseite	1
Rückseite	4
Rechte Seite	2
Linke Seite	5
Unterseite	3
Oberseite	6

Tabelle 3.1.: Tabellarische Übersicht der Zuordnung der Seitenflächen der Readerbox

3.2.2. Dimensionierung der Antennenspulen

Als Dimensionierungsgrundlage wird die Annahme getroffen, dass die zu wickelnden Antennenspulen kreisförmig sind. Diese Annahme lässt sich, durch den rechteckförmigen Aufbau der Readerseitenflächen, nicht vollständig erfüllen. Es kann aber gezeigt werden, dass hiermit eine hinreichend genaue Berechnungsgrundlage geschaffen wird.

In [RFID03] wird die folgende Formel zur Berechnung einer kreisförmigen Antennenspule aufgeführt:

$$L_{\mu H} = \frac{0,31(aN)^2}{6a + 9h + 10b} \quad (3.1)$$

Wobei die Parameter:

a für den mittleren Spulenradius,

N für die Anzahl der Wicklungen,

b für die Wicklungsbreite und

h für die Wicklungshöhe stehen.

Durch eine Umstellung der Formel nach N folgt:

$$N = \sqrt{\frac{L_{\mu H}(6a + 9h + 10b)}{0,31a^2}} \quad (3.2)$$

Dieser Ausdruck stellt die Grundlage für die weitere Berechnung der notwendigen Wicklungszahlen der einzelnen Antennenspulen dar.

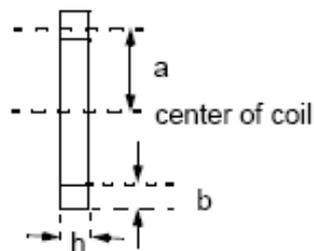


Bild 3.4.: Parameterzuordnung zur Berechnung der Antennenspulen [RFID03]

Als Induktivitätswert für die Antennenspulen wird ein Wert von $L = 725\mu H$ angestrebt. Zum Wickeln der Spule wird ein $0,4mm$ starker Kupferdraht verwendet. Da die Vorgabe besteht die Spule einlagig zu wickeln, wird die Stärke des Kupferdrahts als Wickelhöhe h definiert. Die Wicklebreite b wird für die weiteren Betrachtungen zu $1,5cm$ definiert. Der mittlere Radius $R_m = a$ der einzelnen Antennenspulen lässt sich über eine Mittelung der Abstandswerte der vorgesehenen Aufnahmepunkte zum jeweiligen Spulenmittelpunkt bestimmen.

In Bild 3.5. wird der Verlauf der Induktivitätssteigerung bei steigender Erhöhung der Wicklungszahl im Bereich von 20 – 40 Wicklungen dargestellt.

Durch den symmetrischen Aufbau der Readerbox gelten die berechneten Werte für die mittleren Radien und der Induktivität der Antennenspulen auch jeweils für die gegenüberliegende Seitenfläche.

Die Tabelle 3.2. zeigt die mittels der Berechnung bestimmten Wicklungszahlen. Die jeweils gewonnen Ergebnisse sind als gerundete Ganzzahlen dargestellt. Dies hat den

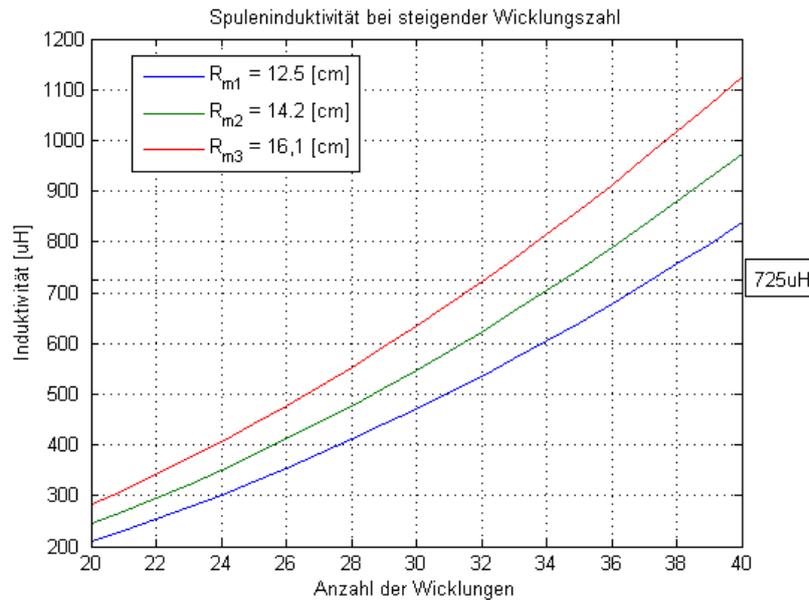


Bild 3.5.: Verlauf der berechneten Induktivitätszunahme der einzelnen Seitenflächen des Readers

Hintergrund, dass beim Wickeln der Spulen immer nur ganze Lagen zu- oder abgewickelt werden. Hiermit soll erreicht werden, dass jeweils ein identischer Start- und Endpunkt für die Spulen vorhanden ist.

Seitenfläche	$a_m = R_m mm$	N berechnet
1	125	37
2	142	35
3	161	32
4	125	37
5	142	35
6	161	32

Tabelle 3.2.: Berechnete Wicklungszahl der einzelnen Seitenflächen der Readerbox

3.2.3. Modifikation der Readerbox

Ausgehend von den gewonnenen Ergebnissen werden die Antennenspulen an die jeweils äußere Seitenfläche der Readerbox angebracht. Zur Aufnahme der Spulen werden Bauelemente aus Kunststoff verwendet. Das direkte Einbringen von metallischen Werkstoffen wird auch hier vermieden.

Die Drahtenden der jeweiligen Luftspulen werden, soweit möglich, auf einer Seitenfläche

(2) der Readerbox zusammengeführt. Von diesem Anschlusspunkt erfolgt die weitere Verbindung zur Readerplatine.

Um eventuell auftretende Störungen durch gegenläufige Signalwege durch die Spulen zu minimieren, wird der Draht gegeneinander verdreht auf den Anschlusspunkt geführt. Die direkte Verbindung zur Readerplatine wird mittels Steckverbindungen realisiert. Der hierfür notwendige Kabelweg wird ebenfalls verdreht realisiert.

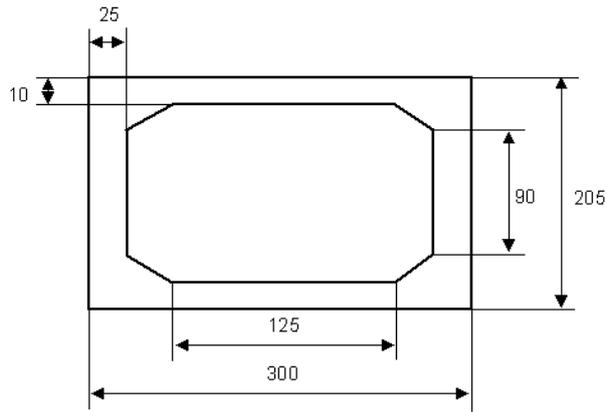
Aus der Entscheidung, nur vollständige Wicklungsumläufe bei der Spulenrealisierung zuzulassen, ergibt sich eine gewisse Quantisierung. Eine messtechnische Überprüfung hat ergeben, dass eine Wickellage einer Induktivität von $\approx 40\mu H$ entspricht. Weiterhin wird ersichtlich, dass die ermittelten Werte für die Wicklungszahl N unterschritten werden können. Durch die Quantisierungsschritte ist es nicht möglich, den angestrebten Wert von $L_{ANT} = 725\mu H$ genau einzuhalten. Bei der Realisierung wird die Wicklungszahl als Endwert angenommen, bei der sich die kleinste Differenz zum vorgegebenen Induktivitätswert ergibt.

Die Tabelle 3.3. zeigt eine Auflistung der Spulenwerte. Die Angabe der Spuleninduktivität wird hier inklusive des notwendigen Verbindungskabels zur Readerplatine angegeben.

Seitenfläche	N berechnet	N verwendet	Induktivität in μH
1	37	36	730
2	35	33	746
3	32	30	722
4	37	36	725
5	35	33	750
6	32	30	760

Tabelle 3.3.: Übersicht der Spulenwerte

Die folgenden Bilder 3.6. bis 3.9. zeigen die einzelnen Seitenflächen und deren Bemaßung. Die Oberseite (Seitenfläche 6) ist so konzipiert, dass sie abnehmbar ist. Um einen Zugriff auf die Innenfläche der Readerbox gewährleisten zu können, ist hier zusätzlich noch ein Durchbruch vorhanden. Die Anschlussleiste für die Kabelverbindung zur Readerplatine befindet sich für diese Seitenfläche gesondert auf der Oberseite der Readerbox.

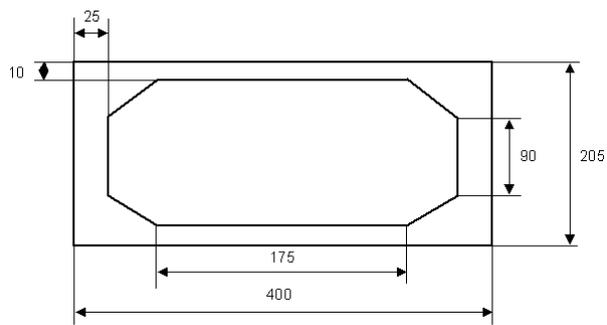


(a) Abmaße der Seitenflächen 1 und 4



(b) Antennenspule der Seitenflächen 1 und 4

Bild 3.6.: Seitenflächen 1 und 4 der Readerbox

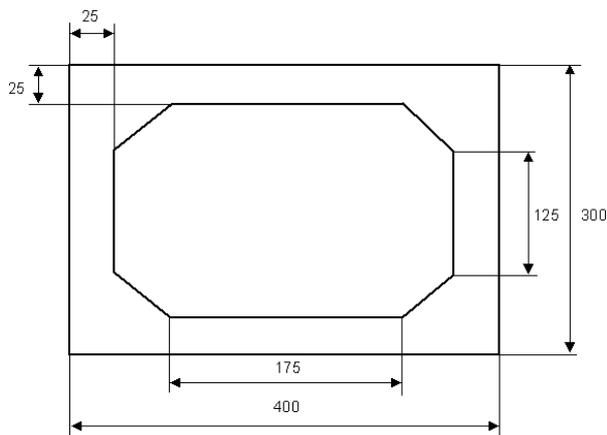


(a) Abmaße der Seitenflächen 2 und 5

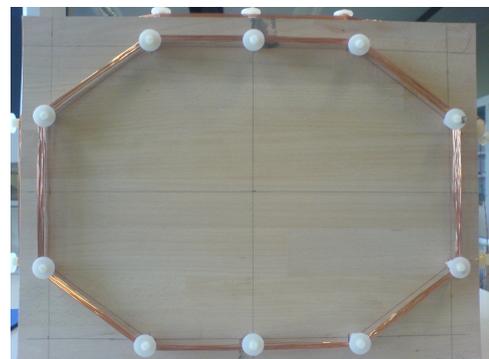


(b) Antennenspule der Seitenflächen 2 und 5

Bild 3.7.: Seitenflächen 2 und 5 der Readerbox

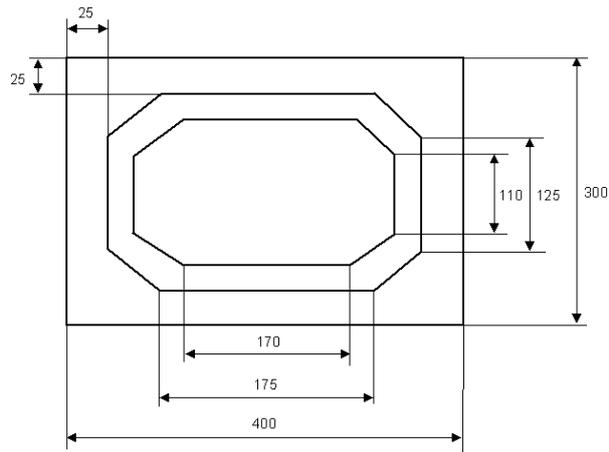


(a) Abmaße der Unterseite des Readers



(b) Antennenspule der Unterseite des Readers

Bild 3.8.: Unterseite der Readerbox



(a) Abmaße der Oberseite des Readers



(b) Antennenspule der Oberseite des Readers

Bild 3.9.: Oberseite der Readerbox

3.3. Beschreibung der Readerplatine

3.3.1. Aufbau der Readerplatine

Auf der Readerplatine befinden sich diverse Bauteile mit unterschiedlichen Aufgabengebieten. Das zur Platine gehörige Blockschaltbild zeigt das Bild 3.10.

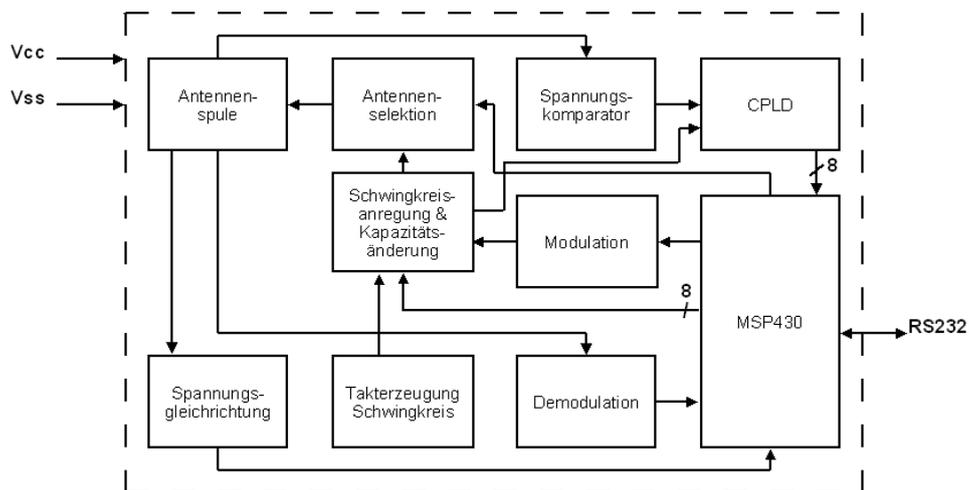


Bild 3.10.: Blockschaltbild zur Readerplatine

Der primär wichtigste Baustein auf der Platine ist der verwendete Mikrocontroller. Der zum Einsatz kommende MSP430 Mikrocontroller bildet die Schnittstelle, um mit den übrigen Funktionsgruppen zu interagieren. Die folgenden Abschnitte geben einen tiefergehenden Einblick in den Aufbau und die Funktion der dargestellten Funktionsblöcke.

3.4. Darstellung der angewandten Messmethode zum Resonanzabgleich

Die bei der Realisierung des Lesegeräts angewandte Messmethode zum Resonanzabgleich stellt eine Kombination der unter 2.2.1 und 2.2.2 vorgestellten Methoden dar.

Es handelt sich hierbei um ein zweistufiges Verfahren. Da der Reader über 6 unterschiedliche Seitenflächen verfügt, erfolgt im ersten Schritt das Ermitteln der jeweils optimalen Kapazität zur Antennenspule. Hierbei wird auf die Abgleichmethode mittels Amplitudenmessung von Tobias Krannich zurückgegriffen.

Im zweiten Schritt wird eine modifizierte Variante der Messmethode zur Einstellung der Resonanzfrequenz nach Giesler, Böh und Tobergte angewendet.

Als Hauptunterschied sei darauf hingewiesen, dass der Phasenvergleich in ein CPLD ausgelagert wird. Hierbei wird direkt der Phasenunterschied zwischen der Anregungsspannung des Antennenschwingkreises und der, dem Referenzpunkt zwischen Kapazität und Induktivität entnommenen, Spannung bewertet. Weiterhin wird auf die vorgeschlagene Abstimmereinheit verzichtet. Eine Resonanzabstimmung erfolgt, mittels des Mikrocontrollers, über eine Zu- oder Abschaltung von Kapazitäten.

Die Phasenbewertung kann, wegen dem unter 2.2.3. dargestellten Problem, nicht vom Mikrocontroller selbst übernommen werden.

Das Bild 3.11. zeigt eine schematische Übersicht beteiligten Funktionsblöcke.

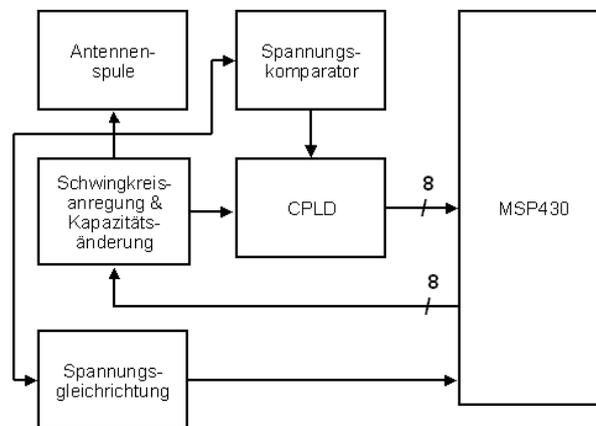


Bild 3.11.: Blockschaltbild zum angewendeten Resonanzabgleich

Der Block Spannungsgleichrichtung dient zur Ermittlung der Größenordnung der Amplitude. Die übrigen Blöcke gehören zur Methode der Resonanzabstimmung. Wobei der Block der Schwingkreisanregung und der Kapazitätsänderung in beiden Abgleichabschnitten zur Einflussnahme auf die Schwingkreisparameter verwendet wird.

Für die Resonanzabstimmung sind insgesamt 8 Kapazitäten vorgesehen. Hierdurch ergeben sich 256 unterschiedliche Kapazitätsstufen. Für die Detektion der Phasenlage werden

8 unterschiedliche Abstufungen vorgesehen. Auf deren genaue Wertigkeiten wird in 3.4.4. eingegangen.

3.4.1. Spannungsgleichrichtung zur Amplitudenmessung

Der hier beschriebene Funktionsblock ist der Methode der Amplitudenmessung zuzuordnen. Hauptaufgabe ist es hier, die dem Schwingkreis am Referenzpunkt zwischen Induktivität und Kapazität entnommene Spannung so aufzubereiten, dass sie dem Mikrocontroller zur Bewertung zugeführt werden kann. Die Bewertung erfolgt über einen Analog to Digital Converter ADC Eingang des eingesetzten MSP430 Controllers. Wobei hier die Größe der Spannungsamplitude ermittelt und ausgewertet wird.

Zu diesem Zweck wird die vom Referenzpunkt abgeleitete Spannung zuerst mittels eines Spannungsteilers begrenzt. Anschließend erfolgt eine Gleichrichtung und eine Glättung der Spannung. Die durch diese Maßnahmen erhaltene Spannung wird abschließend mittels des Mikrocontrollers bewertend betrachtet.

Der eingesetzte Schaltungsaufbau ist in Bild 3.12. ersichtlich.

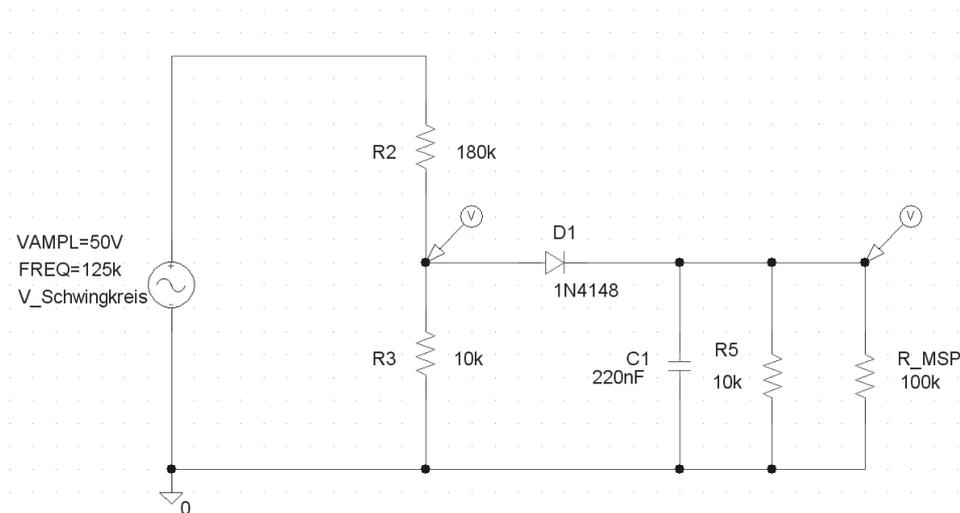


Bild 3.12.: Schaltplan der verwendeten Schaltung zur Spannungsgleichrichtung

Die hier dargestellte Schaltung ist mit der unter [DIPL01] vorgestellten Schaltung zur Amplitudenbewertung zu großen Teilen identisch. Lediglich der Spannungsteiler, bestehend aus den beiden Widerständen R2 und R3, wurde angepasst. Der Spannungsteiler wird für eine Maximalspannung von $50V_{pp}$ ausgelegt, da nicht mit einer höheren Spannung am Schwingkreis gerechnet wird. Ziel dieser Maßnahme ist es, den Spannungsbereich in eine Spannung von $\pm 3V$ zu begrenzen.

Die Gleichrichtung erfolgt mittels der Diode D1 und die Spannungsglättung wird über den Tiefpass, bestehend aus den beiden Bauelementen C1 und R5, bewerkstelligt.

Das Bild 3.13. zeigt das Ergebnis der Simulation der vorgestellten Schaltung.

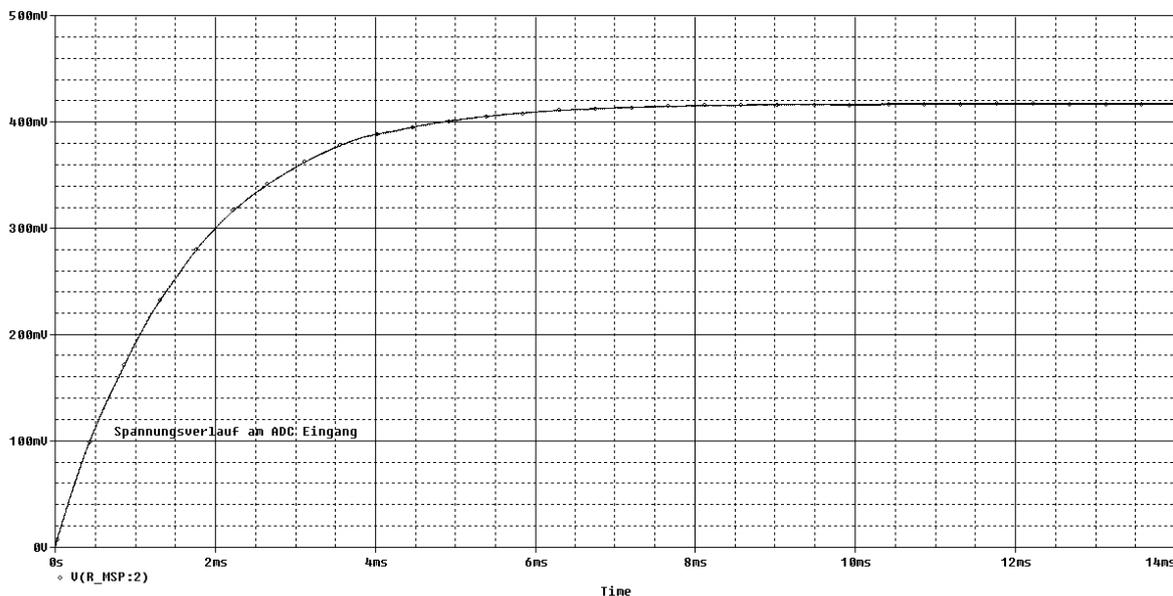


Bild 3.13.: Simulationsergebnis der Schaltung zur Spannungsgleichrichtung

Als Ergebnis der Simulation kann festgestellt werden, dass sich die Schaltung zur Spannungsbegrenzung und Spannungsstabilisierung einsetzen lässt. Daher wird sie ohne weitere Anpassungen auf die Readerplatine übernommen.

3.4.2. Schwingkreisanregung und Kapazitätsänderung

Wie bereits in 2.3.2. erläutert, wird der Antennenschwingkreis der jeweils aktiven Readerseite über Bustreiber (vgl. A.4.) angeregt.

Über das Zu- oder Abschalten von Kapazitäten kann eine Einflussnahme auf den Antennenschwingkreis erwirkt werden. Ziel ist es hier, die jeweils optimale Kapazität zur aktiven Spulenseite zum Einsatz zu bringen. Als optimale Kapazität wird die Kapazität betrachtet, bei deren Zuschaltung sich die höchste Spannungsamplitude im Schwingkreis einstellt. Bei den gerade Beschriebenen handelt es sich um die hauptsächliche Aufgabe des hier zu erläuternden Funktionsblocks.

Eine Kapazitätsänderung lässt sich unter Zuhilfenahme des eingesetzten Mikrocontrollers bewerkstelligen. Mittels diesem lassen sich einzelne Bustreiber aktivieren und/oder deaktivieren. Dies erfolgt über die jeweils vorhandenen Enable-Eingänge der auf der Readerplatine vorhandenen Bustreiber. Bei der Realisierung des Lesegeräts wurde die Entscheidung getroffen, 8 änderbare Kapazitäten zur Verfügung zu stellen. Wobei sich die Auswahl der Wertigkeiten über den Bereich von $22pF$ als Minimalwert und $1nF$ als

Maximalwert erstreckt. Die Tabelle 3.4. beinhaltet die zur Verfügung stehenden Kapazitätswerte.

Benennung	Kapazitätswert in pF
C_1	1000
C_2	1000
C_3	470
C_4	220
C_5	220
C_6	100
C_7	47
C_8	22

Tabelle 3.4.: Übersicht der gewählten Kapazitätswerte

Die Kondensatoren wurden so gewählt, dass sich eine Steigerung der Kapazität durch eine Verdoppelung des vorausgegangenen Kapazitätswerts ergibt. Wobei zu beachten ist, dass einige Kapazitätswerte in zweifacher Ausführung vorhanden sind. Dies geschieht zu Abgleichszwecken in der später erläuterten Phase des Resonanzabgleichs.

Aus den gewählten Kapazitäten ergibt sich eine größtmögliche Kapazität von $3079pF$. Aus der vorausgegangenen Dimensionierung ergibt sich ein Kapazitätswert von etwa $2,2nF$. Somit stellt sich durch die hier dargestellte Kapazitätsauswahl eine Abstimmreserve von $\approx 0,8nF$ ein.

Die eigentliche Schwingkreisanregung erfolgt mit der Taktung, welche vom Funktionsblock der Taktgenerierung zur Erzeugung der Sendefrequenz vorgegeben wird (vgl. 3.5.). Die Taktfrequenz liegt hier idealerweise bei der gewünschten Arbeitsfrequenz von $f_o = 125kHz$. Es werden alle auf der Readerplatine vorhandenen Bustreiber dauerhaft mit dem beschriebenen Takt versorgt. Über den bereits erwähnten Enable-Eingang der jeweiligen Treiber lässt sich eine separate Einflussnahme auf die Schwingkreisanregung bewirken.

Zur Verdeutlichung ist in Bild 3.14. eine Schaltungsanordnung dargestellt, in der die erforderliche Kapazität über 3 aktive Bustreiber in den Schwingkreis eingebracht wird.

Die hier vorgestellte Schaltung beschreibt den tatsächlich eingesetzten Hardwareaufbau lediglich in einer vereinfachten Weise. Daher ist sie nur als Beispielschaltung zu betrachten.

Die erforderliche Kapazität von $C_{RES} = 2,2nF$ wird hier über einer Parallelschaltung von 3 Kondensatoren erreicht. Die übrigen 5 auf der Platine vorhandenen Kapazitäten werden hier nicht aufgeführt. Sie können aber auf die gleiche Weise zugeschaltet werden, wie die dargestellten Kapazitäten.

Die verwendeten Bustreiber sind Low-Aktiv dies bedeutet, dass sie sich mittels des

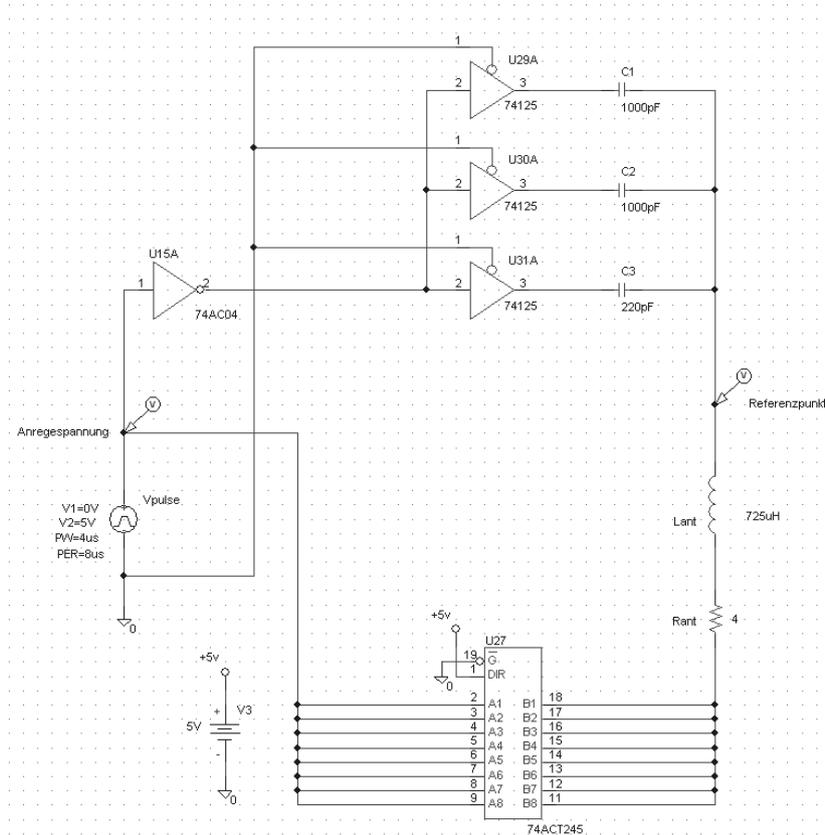


Bild 3.14.: Beispielschaltung zur Kapazitätsänderung

Enable-Eingangs aktivieren lassen, wenn dort eine logische '0' anliegt.

Der Takt zur Anregung des Schwingkreises wird hier von der Spannungsquelle V_{pulse} erzeugt. Bei der Implementierung des Readers werden 2 unterschiedliche Bustreiber eingesetzt. Die Kapazitätsänderung erfolgt mittels einzeln aktivierbaren Treiberstufen. Die Anregung der gegenüberliegenden Schwingkreisseite erfolgt über einen Treiberbaustein bei dem es möglich ist, alle vorhandenen Treiberausgänge gleichzeitig zu schalten. Diese Auswahl ist durch einen etwas geringeren Schaltungsaufwand begründet. Die nicht unmittelbar zur Kapazitätsanpassung verwendeten Treiber sind als hochohmig geschaltet anzusehen.

Das Bild 3.15. beschreibt das Simulationsergebnis zur dargestellten Schaltung.

Die mittels der Simulation erhaltene Spannungsamplitude am Referenzpunkt des Schwingkreises konnte durch eine messtechnische Überprüfung am Reader, nicht in der hier aufgeführten Größenordnung nachgewiesen werden. Auf die tatsächlich gemessenen Spannungsverläufe wird im weiteren Verlauf dieser Arbeit eingegangen.

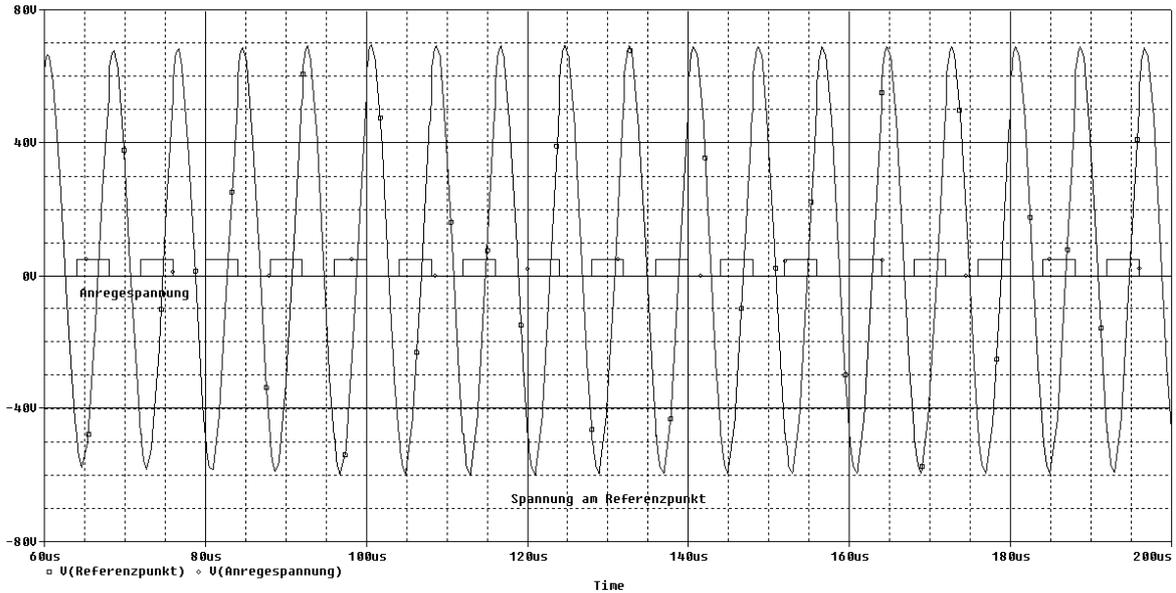


Bild 3.15.: Simulationsergebnis zur Beispielschaltung der Kapazitätsänderung

3.4.3. Spannungskomparator

Dieser Funktionsblock hat die Aufgabe, die tatsächlich am Schwingkreis vorhandene Spannung so aufzubereiten, dass sie dem CPLD zur Bewertung übergeben werden kann. Hierfür wird die dem Referenzpunkt entnommene Spannung zuerst mittels eines Spannungsteilers begrenzt und anschließend einem Spannungskomparator-IC zugeführt. Bei der hier beschriebenen Implementierung wird ein LM311 Komparator A.5 verwendet. Das Bild 3.16. zeigt den Schaltplan der verwendeten Schaltung.

Die beiden Widerstände R1 und R2 bilden den Spannungsteiler. Dieser wird so dimensioniert, dass Spannungen bis $50V_{pp}$ in dem Bereich von $\leq 5V_{pp}$ abgebildet werden. Der Spannungskomparator wird so beschaltet, dass sich eine Hysterese einstellt. Deren Verlauf wird durch das Verhältnis der beiden Widerstände R3 und R4 mitbestimmt. Durch die verwendeten Widerstände ergibt sich ein Verhältnis von $\frac{10}{100k}$. Der Ausgang des Komparators wird mit 5V beschaltet, um diese Spannung als obere Schaltschwelle zu definieren. Der Widerstand R5 ist hier nur als Pullup-Widerstand vorgesehen. Die untere Schaltschwelle ergibt sich zu: $\frac{5V}{10.000} = 500\mu V$. Die Hysterese wird verwendet, um den Komparator störunabhängiger gegenüber Spannungsschwankungen am Signaleingang zu machen.

Da es sich bei dem verwendeten Komparator um einen invertierenden Komparator handelt, besteht die Notwendigkeit das Ausgangssignal zu invertieren. Diese Aufgabe übernimmt der eingesetzte Logikbaustein U3A aus der 7404 Baureihe. Das Bild 3.17. zeigt das Simulationsergebnis der Schaltung.

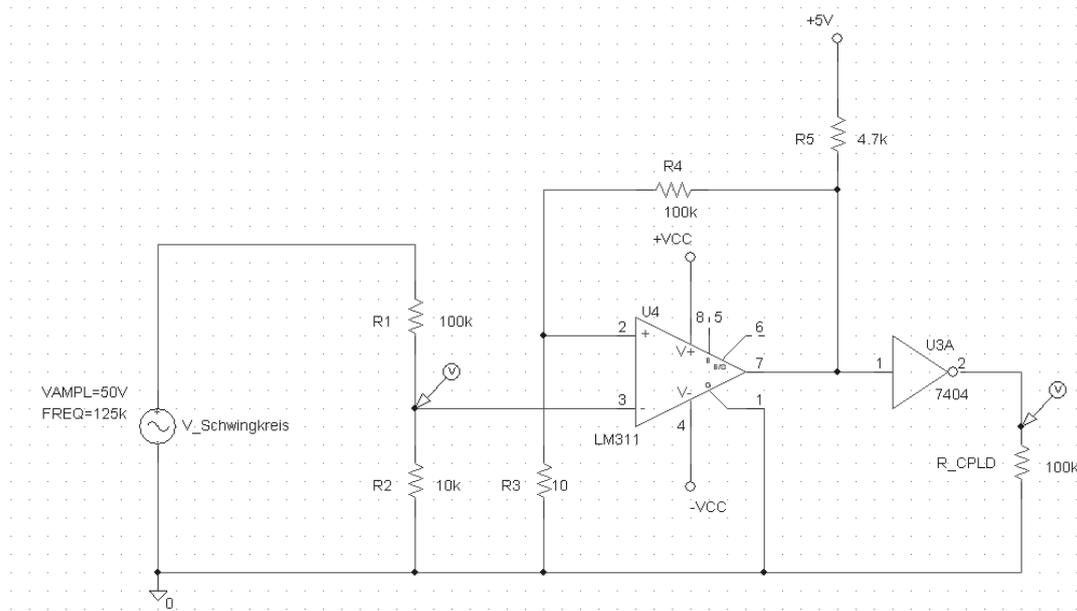


Bild 3.16.: Schaltplan der verwendeten Komparatorschaltung

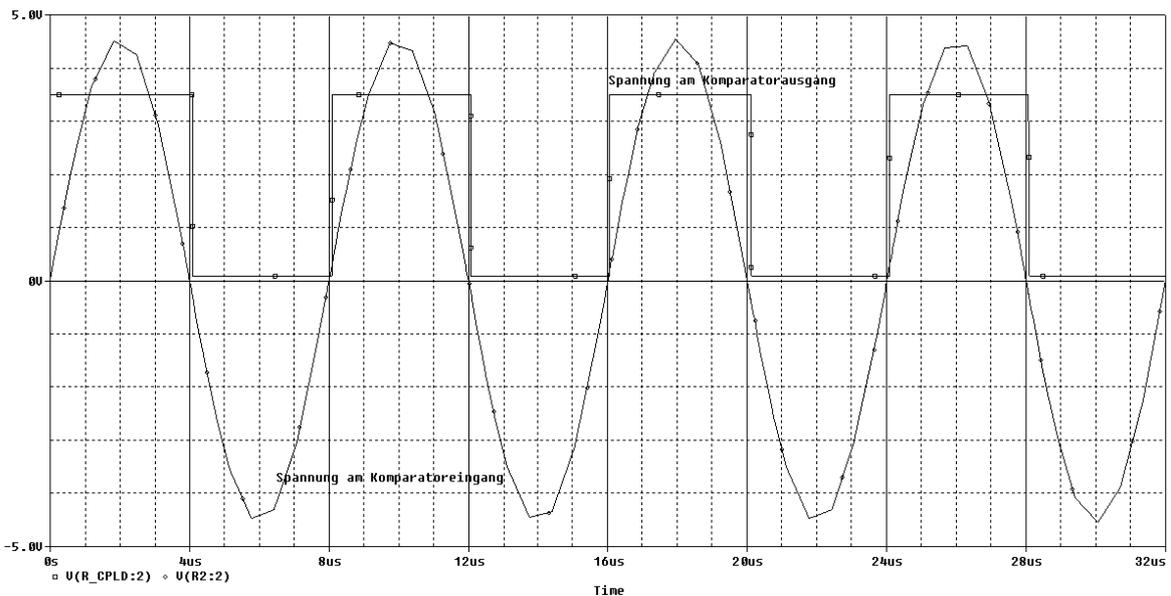


Bild 3.17.: Simulationsergebnis der Komparatorschaltung

Eine Messung an der Readerplatine bestätigt ebenfalls die gewünschte Funktion der Komparatorschaltung. Das Bild 3.18 zeigt die mittels eines Oszilloskops gemessenen Spannungsverläufe.

Die Messung wird hier exemplarisch an der Antennenspule 2 der Readerbox durchgeführt. Der Kanal 1 zeigt den Verlauf der Rechteckspannung vor der Inverterstufe. Der zweite Kanal bildet die Spannung am Komparatoreingang ab. Die im oberen Verlauf der

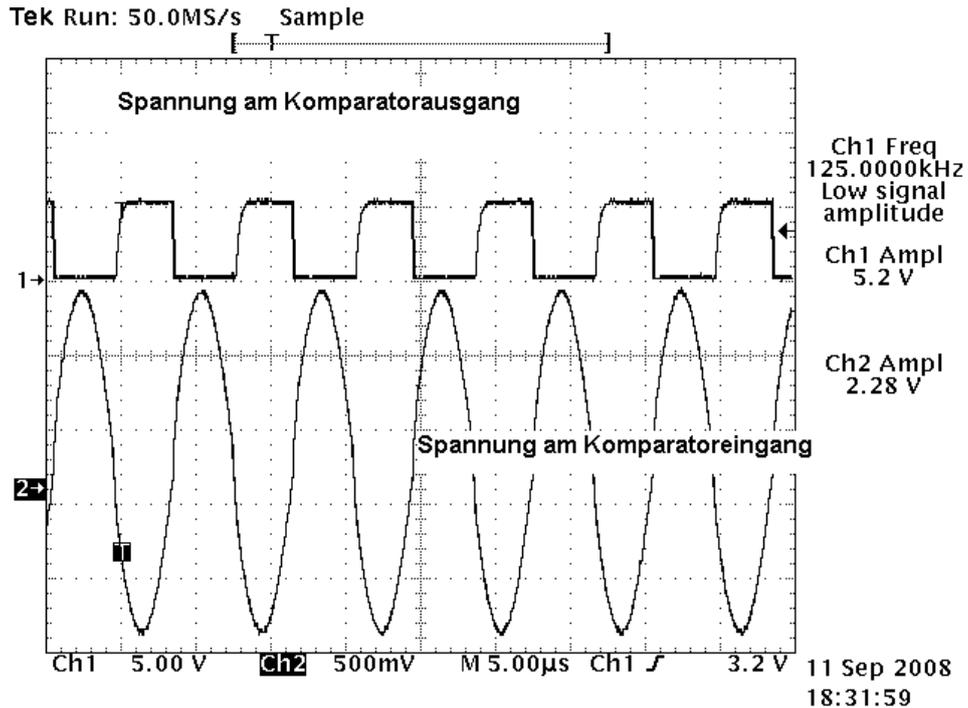


Bild 3.18.: Oszillierte Spannungsverläufe zur Komparatorschaltung

Darstellung aufgeführte Rechteckspannung stellt die Spannung am Komparatorausgang dar. Da es sich um einen invertierenden Komparator handelt, sind die Signalzustände hier gegensätzlich ersichtlich. Dieser Umstand wird durch den nachgeschalteten Inverter wieder umgekehrt.

3.4.4. CPLD

Diesem Funktionsblock obliegt es, die Hauptaufgabe zur Bewertung der Phasenlage der zum Resonanzabgleich herangezogenen Signale zu übernehmen. Wie bereits beschrieben, handelt es sich hierbei um die Rechteckspannung, welche zur Anregung des Schwingkreises mittels der vorhandenen Bustreiber und der am Referenzpunkt zwischen Kondensator und Spule entnommenen Spannung. Letzte wird, wie im vorherigen Abschnitt erläutert, so aufbereitet, dass sie ebenfalls als Rechteckspannung vorliegt.

Als Bewertungsgrundlage dient die Auswertung der Phasenlage der beiden Signale zueinander. Zu diesem Zweck ist die zeitliche Dauer zu ermitteln, welche benötigt wird, damit beide Signale einen Zustandswechsel von einer Low-Phase in eine High-Phase vollzogen haben. Bei diesem Verfahren wird daher die Zeit ermittelt, die verstreicht bis bei beiden Signalen eine steigende Flanke detektiert wurde.

Die Auswertung dieser zeitlichen Differenz kann wegen der in 2.2.3. erläuterten Sachlage nicht direkt vom eingesetzten Mikrocontroller ausgeführt werden. Daher erfolgt die

Bewertung unter Zuhilfenahme eines CPLDs (siehe A.3.). Dieser programmierbare Baustein lässt sich über einen externen Oszillator wesentlich höher takten, als es bei dem MSP430 Mikrocontroller möglich ist. Bei der hier vorliegenden Implementierung wird der CPLD mit einem Takt von $40MHz$ betrieben.

Aus dieser Auswahl ergibt sich eine mögliche Anzahl der Taktschritte des CPLD pro Signalperiode des zu prüfenden Signals von:

$$n_{takt} = \frac{40MHz}{125kHz} = 320 \quad (3.3)$$

Setzt man die Anzahl der zur Verfügung stehenden Taktschritte mit der Phasenänderung von 360° pro voller Signalperiode in Beziehung, so ergibt sich eine mögliche Auflösung von:

$$\frac{360^\circ}{320} = 1,125^\circ \quad (3.4)$$

pro Taktschritt.

Zur Umsetzung der Bewertung wird der CPLD mittels der Hardwarebeschreibungssprache VHDL programmiert. Der zu diesem Zweck erstellte Code ist unter C.2. aufgeführt. Dem CPLD werden 4 Eingangs- und 8 Ausgangssignale zugeordnet. Das Bild 3.19. stellt diese dar. Wobei es sich bei den Signalen auf der linken Seite um die Eingänge handelt. Auf der rechten Seite werden die Ausgänge dargestellt.

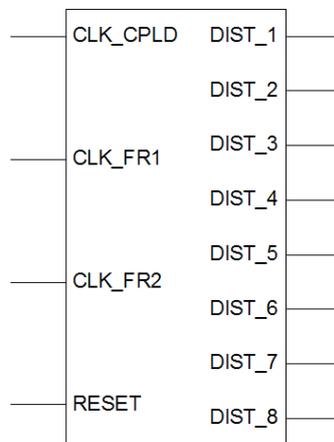


Bild 3.19.: Beschreibung der Ein- und Ausgangssignale des CPLD

Über den Eingang CLK_{CPLD} wird der Baustein mit seiner Arbeitsfrequenz von $40MHz$ versorgt. Mit Hilfe des Eingangssignals $RESET$ lässt sich der CPLD zurücksetzen. Die beiden zu bewertenden Signale sind den Eingängen CLK_{FR1} und CLK_{FR2} zugeordnet. Wobei hier die Taktfrequenz, die zur Bustreiberanregung dient, dem Eingangssignal CLK_{FR1} zuzuordnen ist. Die mittels des Spannungskomparators aufbereitete Spannung, wird dem CPLD über den Eingang CLK_{FR2} zugeführt. Das Ergebnis der Auswertung wird über die Ausgänge $DIST_1$ bis $DIST_8$ ausgegeben.

Diese Ausgänge werden auf der Readerplatine mit dem vorhandenen Mikrocontroller verbunden.

Die Bewertungsaufgabe wird mittels zweier Zähler bewerkstelligt. Über diese wird zuerst die jeweilige Periodendauer des anliegenden Signals erfasst.

Als zweiter Schritt wird eine Flankenzustandserkennung der Rechteckspannung durchgeführt, welche den Spannungsverlauf, der am Referenzpunkt des Schwingkreises entnommenen Spannung, repräsentiert. Wird hier ein Flankenwechsel von Low nach High erkannt, so wird dies als Beginn einer neuen Signalperiode gewertet. Als Folge hierauf werden die beiden Zählerstände subtrahiert. Das darüber erhaltene Ergebnis beschreibt die Zeitdifferenz, die als weitere Bewertungsgrundlage zur Resonanzabstimmung herangezogen wird.

Im weiteren Programmablauf wird die ermittelte Differenz bezüglich ihrer Größe bewertet. In Abhängigkeit hiervon wird jeweils ein Ausgangssignal generiert.

Der hier beschriebene Ablauf wird innerhalb des CPLDs als ein getakteter Prozess durchlaufen. Dieser Prozess reagiert sensitiv auf die beiden Eingänge CLK_{CPLD} und $RESET$. Im Falle eines anliegenden Resetsignals werden die internen Zählerstände auf den Initialwert von 0 zurückgesetzt.

Tritt dieser Fall nicht ein, so werden die beschriebenen Anweisungen mit Beginn der steigenden Taktflanke des CPLD Clocks ausgeführt.

Als Resultat dieser Art der Implementierung ist es möglich, ein dauerhaftes Ausgangssignal zu erhalten. Über dieses Signal können Rückschlüsse über den Grad der Resonanzabstimmung getroffen werden.

Über das Ausgangssignal lässt sich eine Differenzdauer im Bereich von $0\mu s$ bis $\geq 3\mu s$ beschreiben. Als Phasenlage ausgedrückt entspricht dies einem Winkel von 0° bis $\geq 135^\circ$. Als Idealfall wird eine zeitliche Differenz von $2\mu s$ oder 90° Phasenlage angestrebt. In diesem Fall wird der Schwingkreis als abgestimmt betrachtet. Da sich damit der bestmögliche Fall der Resonanzlage eingestellt hat.

Dieses Ergebnis kann bei der Implementierung aber nicht immer exakt erreicht werden. Daher wird die zeitliche Differenz als Optimum definiert, welche die geringste Differenz zu den $2\mu s$ aufweist.

Die Tabelle 3.5. stellt die Zuordnung der Ausgangssignale zur jeweiligen Differenzbeziehung dar.

Der erstellte VHDL Code wurde mittels der Software Modelsim (vgl. B.6.) simuliert. Das Ergebnis der Simulation wird in Bild 3.20. dargestellt.

Für diese Simulation wird der CPLD mit eine Taktfrequenz von $40MHz$ betrieben. Die oberen Signale beschreiben die Zustände der Eingangs- und der Ausgangssignale. Danach folgen die Zustandsbeschreibungen der notwendigen internen Signale. Die beiden

Ausgang	Differenz in μs	Differenz in $^\circ$
$DIST_1$	0 – 1	0 – 45
$DIST_2$	> 1 – 1,5	> 45 – 67,5
$DIST_3$	> 1,5 – 1,75	> 67,5 – 78,75
$DIST_4$	> 1,75 – 2	> 78,75 – 90
$DIST_5$	> 2 – 2,25	> 90 – 101,25
$DIST_6$	> 2,25 – 2,5	> 101,25 – 112,5
$DIST_7$	> 2,5 – 3,0	> 112,5 – 135
$DIST_8$	> 3,0	> 135

Tabelle 3.5.: Zuordnung der Ausgangssignale zur Differenzbeziehung

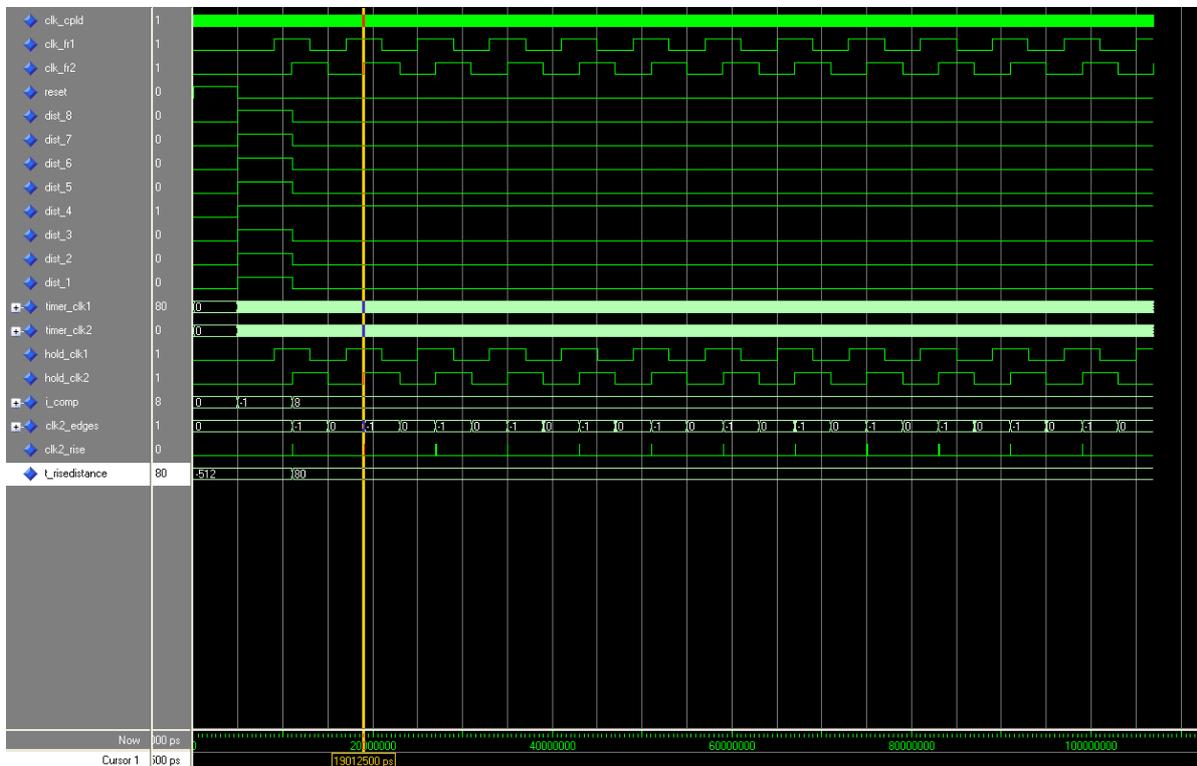


Bild 3.20.: Simulationsergebnis des VHDL Codes

verwendeten Zähler werden mit $timer_{clk1}$ und $timer_{clk2}$ bezeichnet. Die Signale $hold_{clk1}$ und $hold_{clk2}$ werden zur Ermittlung der Signalfrequenz benötigt. Mittels i_{comp} wird der Zustand der Ausgangssignale innerhalb des getakteten Prozesses beschrieben. Die Zuweisung der Ausgangszustände erfolgt mit Hilfe einer nebenläufigen Anweisung. Das Signal $clk2_{edges}$ dient zur Erkennung eines Flankenwechsels des zweiten Taktsignals. Als Auslöser für die Berechnung der Taktdifferenz zwischen den beiden Eingängen clk_{fr1} und clk_{fr2} dient ein Zustandswechsel des Signals $clk2_{rise}$ von Low nach High. Über den

internen Zählerstand von $t_{risedistance}$ wird das jeweils zu setzende Ausgangssignal ermittelt.

Zu Beginn der Simulation wird ein Reset ausgeführt. Dies hat zur Folge, dass alle Ausgänge eine logische '1' anzeigen. Hierüber lässt sich im weiteren Verlauf ein möglicher Fehler signalisieren.

Das hier aufgeführte Simulationsergebnis beschreibt den Fall, dass beide Signale sich in der Phasenlage zueinander befinden, welche den gewünschten Fall des abgestimmten Schwingkreis repräsentiert. Zwischen den beiden Eingangssignalen liegt eine zeitliche Differenz von $2\mu s$. Die Differenz wird durch eine logische '1' am Ausgang $DIST_4$ zur Anzeige gebracht.

Der zu diesem Design gehörige Synthesereport wird ebenfalls unter C.2. aufgeführt. Der zugehörige Fitterreport ist aus Gründen der besseren Übersichtlichkeit lediglich auf dem beigefügten Datenträger enthalten.

3.5. Taktgenerierung zur Erzeugung der Sendefrequenz

Über die Ansteuerung der Bustreiber mit einem definierten Takt lässt sich die Sendefrequenz des Antennenschwingkreises beeinflussen. Diese Sendefrequenz wird für diese Arbeit zu $125kHz$ definiert. Die Erzeugung des Taktes zur Bustreiberanregung wird in einen separaten IC ausgelagert. Für diese Aufgabe wird ein Zählerbaustein A.6 verwendet.

Der verwendete IC wird mit einem zusätzlichen Quarz beschaltet. Der Baustein hat die Eigenschaft, den zugeführten Takt über die vorhandenen Ausgänge ganzzahlig dividiert zur Verfügung zu stellen. Die Division erfolgt intern in binären Stufen.

Die Taktversorgung des Zähler-ICs erfolgt über einen $16MHz$ Quarz. Die gewünschte Taktfrequenz zur Treiberansteuerung erfolgt über den Ausgang, der dem Quotienten von $f_0 = \frac{16MHz}{2^7} = 125kHz$ entspricht.

Das Bild 3.21. stellt die Schaltung dar, die diesen Funktionsblock repräsentiert.

IC22 stellt den Stecksockel zur Aufnahme des ICs dar. Die erforderlichen $125kHz$ werden dem sechsten Pin entnommen und dienen zur Ansteuerung der Bustreiber. Das Taktsignal des Quarzes wird dem IC über die beiden Eingänge 10 und 11 zugeführt. Damit der Quarz die erwartete Funktionalität bereitstellen kann, ist die dargestellte Beschaltung notwendig.

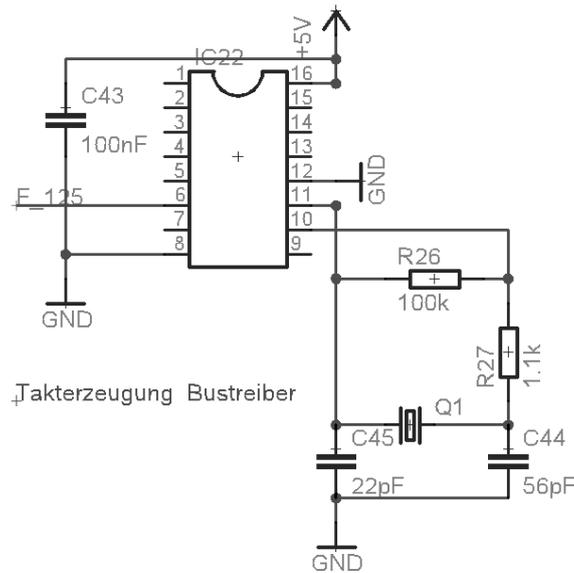


Bild 3.21.: Schaltung zur Erzeugung der Taktfrequenz zur Bustreiberanregung

3.6. Modulation

Es ist vorgesehen, dass eine mögliche Datenübertragung vom Reader zu einem oder mehreren Transpondern generell nach dem Prinzip der Amplitudenmodulation zu erfolgen hat. Die zu diesem Zweck notwendige Hardware ist auf der Readerplatine vorhanden. Lediglich eine softwareseitige Umsetzung konnte nicht mehr realisiert werden.

Auch für diesen Teil des Readers wird auf die Vorarbeit von Tobias Krannich [DIPL01] zurückgegriffen. Im Rahmen dieser Arbeit wird eine schaltungstechnische Anordnung vorgestellt, welche geeignet ist die geforderte Funktionalität bereitzustellen. Diese Schaltung wird bei der hier vorgestellten Implementierung des Lesegeräts in adaptiver Form angewendet.

Das Bild 3.22. zeigt das Blockschaltbild zur prinzipiellen Realisierung der Modulation.

In die Zweige der Bustreiberanregung werden zusätzlich zwei UND-Gatter eingefügt. Über diese Logikbausteine ist es möglich, Daten auf das Sendesignal zu modulieren. Die zu übertragenden Daten können als serieller Bitstrom vom Mikrocontroller erzeugt werden. Durch diese Methode lassen sich manchestercodierte Daten übermitteln.

Für diese Art der Amplitudenmodulation wird ein Modulationsgrad von 100% angewendet. Es handelt sich hierbei um das Verfahren des On Off Keying, da jeweils zwischen einer nicht vorhandenen und der vollen Signalamplitude umgeschaltet wird.

Eine Beispielschaltung zum Funktionsblock der Modulation wird in Bild 3.23. aufgeführt.

Als Ausgangspunkt für diese Beispielschaltung dient die unter 3.4.2. eingeführte Schal-

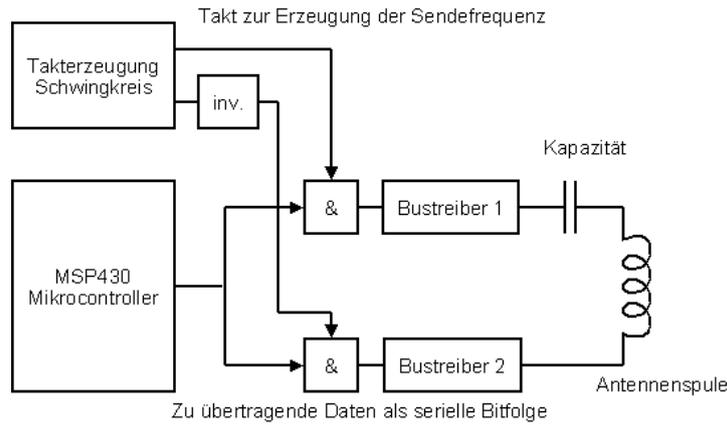


Bild 3.22.: Blockschaltbild zur Modulation

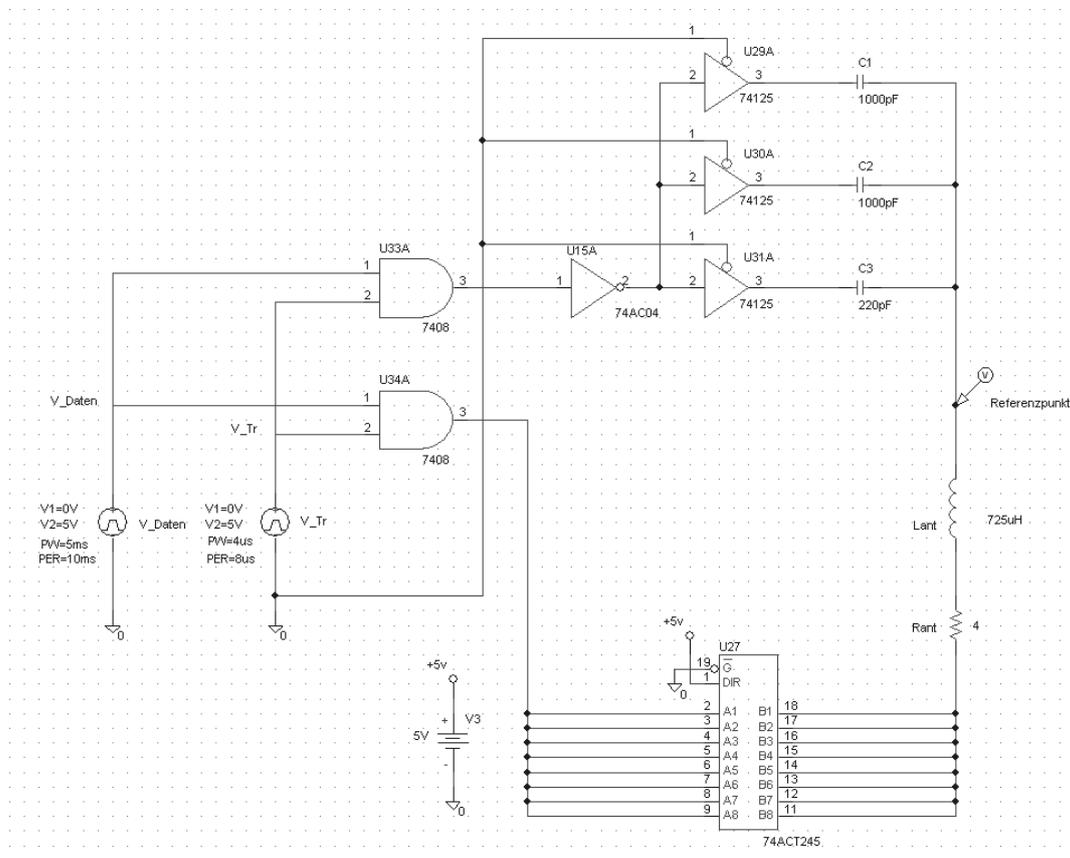


Bild 3.23.: Beispielschaltung zur Modulation

tion zur Schwingkreisanregung und Kapazitätsänderung.

Die vorhandenen Bustreiber werden mit einer Taktfrequenz von $f_{Tr} = 125kHz$ angeregt. Diese Frequenz beschreibt die Taktfrequenz der Trägerspannung.

Zusätzlich wird über die beiden vorhandenen UND-Gatter ein Rechtecksignal mit einer Frequenz von $f_{Daten} = 100Hz$ zum Träger hinzugefügt. Diese Rechteckspannung stellt die zu modulierenden Daten dar.

Das Bild 3.24. zeigt das Simulationsergebnis zur vorgestellten Modulationsschaltung.

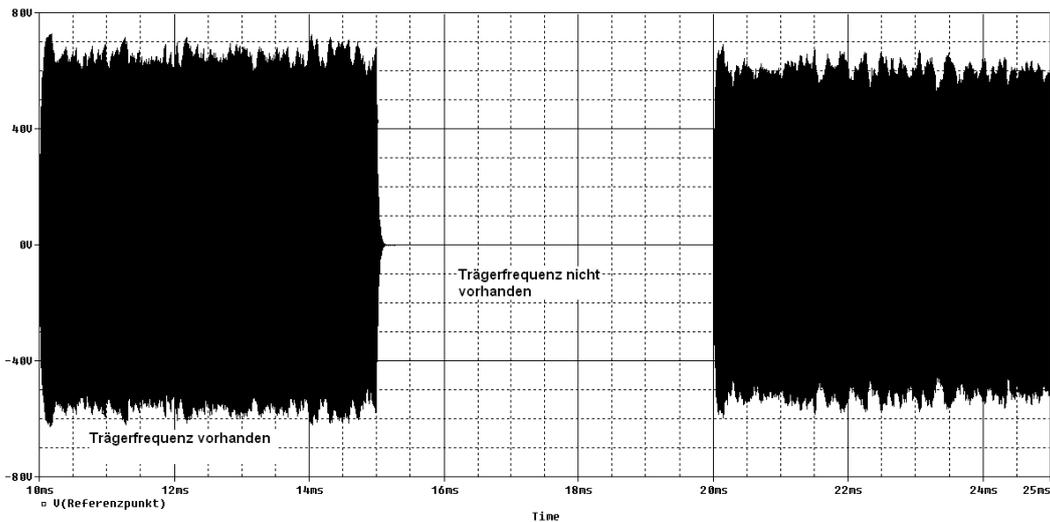


Bild 3.24.: Simulationsergebnis der Beispielschaltung zur Modulation

Aus diesem Simulationsergebnis wird ersichtlich, dass die vorhandene Trägerfrequenz im Takt der Signalfrequenz zu- oder abgeschaltet wird. Dieses Verhalten entspricht der Methode der Amplitudenmodulation mit einem Modulationsindex von $m = 1$. Es wird daher das gewünschte Verhalten beschrieben.

3.7. Demodulation

Diesem Funktionsblock obliegt es, die vom Transponder gesendeten Daten in geeigneter Art und Weise aufzubereiten, damit eine weitere Verarbeitung durch den verwendeten Mikrocontroller erfolgen kann. Da die Datenübertragung über das Verfahren der Amplitudenmodulation erfolgt, wobei hier die Nutzdaten auf die vom Reader emittierte Trägerfrequenz moduliert werden, ergibt sich die Notwendigkeit der Demodulation des detektierten Signalfusses.

Zu diesem Zweck wird das, an der jeweils aktiven Antennenspulenseite, empfangene Signal über eine analoge Filterkette geführt. Mittels dieser Filterkette werden die empfangenen Daten von der Trägerfrequenz getrennt. Diesen Vorgang bezeichnet man als Demodulation.

Bei der Realisierung dieses Funktionsblocks wird ein weiteres Mal auf die positiven Ergebnisse der Diplomarbeit von Tobias Krannich [DIPL01] zurückgegriffen. Die dort vorgeschlagene Realisierung der Demodulation findet auch in dieser Diplomarbeit ihre Anwendung. Wobei die hier verwendete Filterkette an einigen Punkten leicht modifiziert wurde.

Das Bild 3.25. beschreibt den Aufbau der analogen Filterkette.

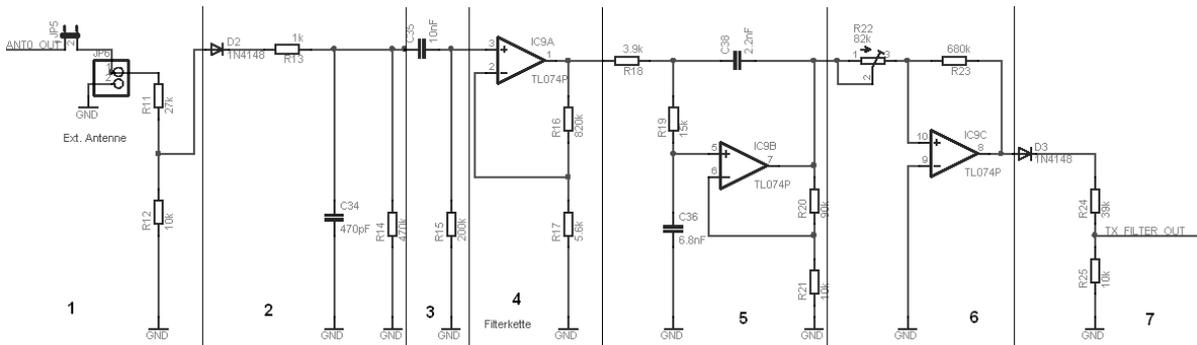


Bild 3.25.: Schaltplan der analogen Filterkette zur Demodulation

Die Filterung des Antennensignals erfolgt in unterschiedlichen Teilabschnitten.

1. Begrenzung der Eingangsspannung

Der Arbeitsbereich der verwendeten Operationsverstärker beschränkt sich auf einen Bereich von $\pm 15V$. Über den verwendeten Spannungsteiler wird sichergestellt, dass Eingangsspannungen bis zu einer Maximalamplitude von $50V_{pp}$ so begrenzt werden, dass sie den Arbeitsbereich der Operationsverstärker nicht überschreiten.

2. Hüllkurvendemodulation

Mit Hilfe der vorhandenen Diode erfolgt eine Gleichrichtung des Signals. Hierbei wird die negative Halbwelle der Antennenspannung abgetrennt. Für die weitere Betrachtung wird lediglich die obere Halbwelle betrachtet. Dieses Spannungssignal besteht aus der hochfrequenten Trägerfrequenz und den aufmodulierten Daten. Im nächsten Schritt erfolgt eine erste Tiefpassfilterung des Signals.

Die Gleichrichtung erfolgt nicht wie im Schaltplan aufgeführt über eine Diode vom Typ 1N4148, sondern über eine Diode vom Typ BAT83. Hierbei handelt es sich um eine Shottky Diode mit einer Durchlassspannung von $\approx 0,3V$.

3. Entfernung des Gleichanteils

Es erfolgt an dieser Stelle eine Hochpassfilterung des anliegenden Signals. Dies hat zum Ziel, eine Entfernung des enthaltenen Gleichanteils zu bewirken. Damit in nächsten Schritt eine Verstärkung des Spannungssignals erfolgen kann.

Die gewünschten Informationen lassen sich aus dem Wechselanteil der Spannung gewinnen, der Gleichanteil verfügt nicht über die gewünschte Aussagekraft.

4. Verstärkung des gefilterten Signals

Der hier aufgeführte invertierende Verstärker bewirkt eine Anhebung des anliegenden Signals um den Faktor von ≈ 140 .

5. Trennung der Nutzdaten vom Trägersignal

An dieser Stelle der Filterkette erfolgt die endgültige Trennung der aufmodulierten Daten vom Trägersignal. Dies geschieht über die dargestellte Schaltungsanordnung. Es handelt sich hierbei um einen aktiven Tiefpass der zweiten Ordnung. Die Grenzfrequenz des Filters liegt hier bei $f_g = 4kHz$. Am Ausgang des Filters findet auch hier eine Spannungsverstärkung statt. Dabei wird die Spannung um einen Faktor von 10 verstärkt.

6. Digitalisierung der Nutzdaten

Das an dieser Stelle anliegende Signal beschreibt die vom Transponder gesendeten Daten in Form einer Sinusschwingung. Zur Weiterverarbeitung der Nutzdaten, durch den nachgelagerten Mikrocontroller, sollten die Daten in eine Rechteckspannung überführt werden. Die geschieht unter Zuhilfenahme der aufgeführten Schmitt Trigger Schaltung.

7. Gleichrichtung und Spannungsbegrenzung des Ausgangssignals

Der hier anliegende Spannungsverlauf liegt maximal im Bereich von $\pm 15V$. Zur weiteren Betrachtung der Daten wird abermals nur der positive Anteil herangezogen. Aus diesem Grund erfolgt auch hier eine Spannungsgleichrichtung durch eine Schottky Diode. Zum Abschluss der Filterkette erfolgt eine Spannungsbegrenzung des Ausgangssignals bis zu einer Maximalamplitude von $3,3V$. Der verwendete Mikrocontroller ist in der Lage, Signale bis zu dieser Eingangsspannung verarbeiten zu können.

Ein Nachweis über die Funktionalität der hier beschriebenen Filterkette erfolgt unter Abschnitt 4.3.

3.8. Antennenselektion

Wie in Bild 3.10. ersichtlich, lässt sich mittels dieses Funktionsblocks eine Umschaltung der aktuell aktiven Antennenspule erwirken. Jede der 6 individuellen Antennenspulen ist einer der Seitenfläche der Readerbox zugeordnet. Über geeignete Steuersignale, welche vom eingesetzten Mikrocontroller ausgehen, lässt sich eine Umschaltung initiieren. Die Umschaltung erfolgt über die auf der Readerplatine vorhandenen Reed Relais (vgl. A.7.). Hierbei werden jeder Antennenspule 2 Relais zugeordnet. Es ergibt sich hierdurch eine Gesamtanzahl von 12 Relais für die vorhandenen 6 Readerseiten.

Das Bild 3.26. zeigt eine Darstellung, in der nur eine Antennenspule vorhanden ist.

Die einzelnen Relais lassen sich über das Anlegen einer logischen '1' schließen oder durch eine logische '0' in einen geöffneten Zustand versetzen. Dabei ist zu beachten, dass sich die jeweiligen Relais hinter den zugeschalteten Bustreibern und den Kondensatoren befinden. Die Relais sind somit direkt vor den Antennenspulen positioniert.

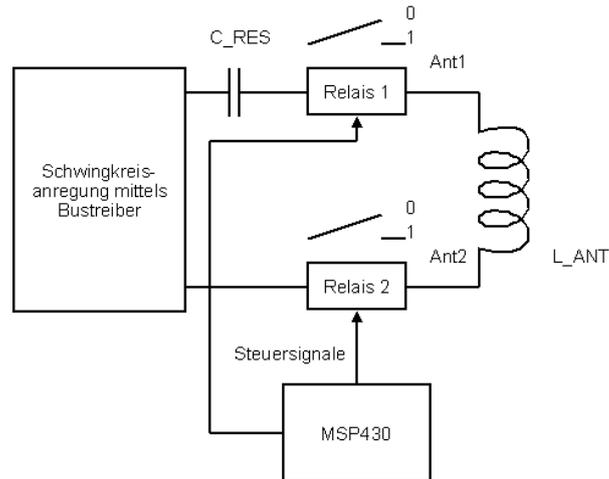


Bild 3.26.: Beispielschaltung zur Antennenselektion

Die Reed Relais benötigen eine Signalspannung von $+5V$, um in einen geschlossenen Zustand versetzt werden zu können. Der vorhandene MSP430 Mikrocontroller arbeitet aber lediglich mit einer Ausgangsspannung von $3,3V$. Daher werden die Steuersignale, bei der Implementierung der Readerplatine, über zusätzliche Pegelumsetzer (siehe A.8.) geführt. Diese Umsetzer sind in der Lage, die $3,3V$ Pegel in die geforderten $5V$ Pegel zu überführen.

3.9. Software des Readers

Die Software zum Betrieb des Readers wurde in der Hochsprache C entwickelt. Der Entwurf der Software geschieht unter Zuhilfenahme der Entwicklungsumgebung Eclipse (siehe B.2.). Das Übersetzen des C Codes in einen, vom eingesetzten Mikrocontroller, ausführbaren Maschinencode geschieht unter Einsatz der Mspgcc Toolchain (vgl. B.1.).

Der erstellte Quellcode wird in vier unterschiedliche Dateien unterteilt. Deren vollständiger Inhalt ist unter C.3. aufgeführt. An dieser Stelle erfolgt eine generelle Beschreibung der erstellten Software.

1. main.h

In dieser Headerdatei werden die wesentlichen Grundeinstellungen definiert. Zu diesen gehören beispielsweise die Portzuordnungen der relevanten Signale und die eigentliche Definition der Ein- und Ausgangsbelegung des Mikrocontrollers. Weiterhin werden an dieser Stelle die Funktionsprototypen der im weiteren Verlauf verwendeten Software-routinen deklariert. Ebenso werden hier die im Programmablauf eingesetzten globalen Variablen bezüglich ihres Datentyps definiert.

2. main.c

In dieser Datei wird der eigentliche Programmablauf festgelegt. Zuerst erfolgt die notwendige Grundinitialisierung der eingesetzten Peripheriebausteine. Anschließend werden die Startwerte für die zuvor definierten Variablen gesetzt. Nach der Ausgabe der Startmeldung wechselt das Programm in die Hauptschleife und es wird auf eine Eingabe über die serielle Schnittstelle gewartet. Über eine geeignete Eingabe lassen sich die jeweiligen Antennen an den Seitenflächen der Readerbox zu- oder abschalten.

3. MSP_Functions.c

In diesem Teil des Quellcodes werden alle zum Zwecke des Readerbetriebs erstellten Funktionen beschrieben. Die zur Aktivierung der Peripherie des Mikrocontrollers benötigten Funktionen sind hier zu finden. Ebenso wie die Funktionen die notwendig sind, um den Controller selbst zu betreiben.

Die Softwareroutinen, die zur Kapazitätsanpassung und zur Antennenselektion verwendet werden, sind ebenfalls an dieser Stelle zusammengefasst.

4. ISR.c

Diese Datei wird zur Beschreibung der eingesetzten Interruptroutinen verwendet. Bei dieser Implementierung des Lesegeräts ist lediglich eine Interruptroutine vorgesehen. Beim Empfang eines Zeichens über die serielle Schnittstelle wird ein Interrupt ausgeführt. Das empfangene Zeichen wird einer globalen Variable zugewiesen. Zusätzlich wird noch ein Flag, welches den Empfang des Eingabezeichens signalisiert, gesetzt. Dieses Flag wird in der Hauptschleife der Software detektiert und anschließend erfolgt eine Auswertung des empfangenen Zeichens.

Der Programmablaufplan zur erstellten Software ist in Bild 3.27. aufgeführt.

Aufgrund der aufgetretenen Probleme bei der Realisierung der Hardware des Readers, konnte die erstellte Software leider nicht in dem Umfang realisiert werden, wie ursprünglich vorgesehen. Die Teile der Software, die nötig wären um das Senden und Empfangen von Daten zu ermöglichen, fehlen bei der hier vorgestellten Implementierung. Auch die Software zur Auswertung des Resonanzabgleichs befindet sich bisher noch in einem rudimentären Status. Durch den modularen Aufbau der Readersoftware sollte dieser Umstand aber bei eventuellen Nachfolgearbeiten änderbar sein.

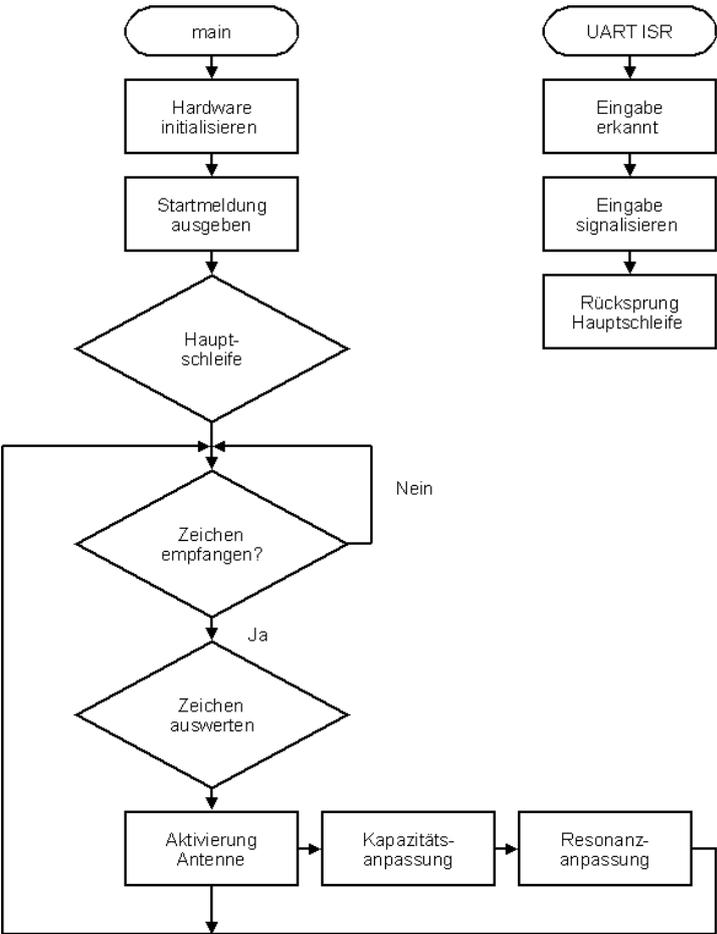


Bild 3.27.: Programmablaufplan zur Readersoftware

3.10. Darstellung der erstellten Readerplatine

Das Bild 3.28. zeigt eine Darstellung der für diese Diplomarbeit entworfenen Readerplatine.

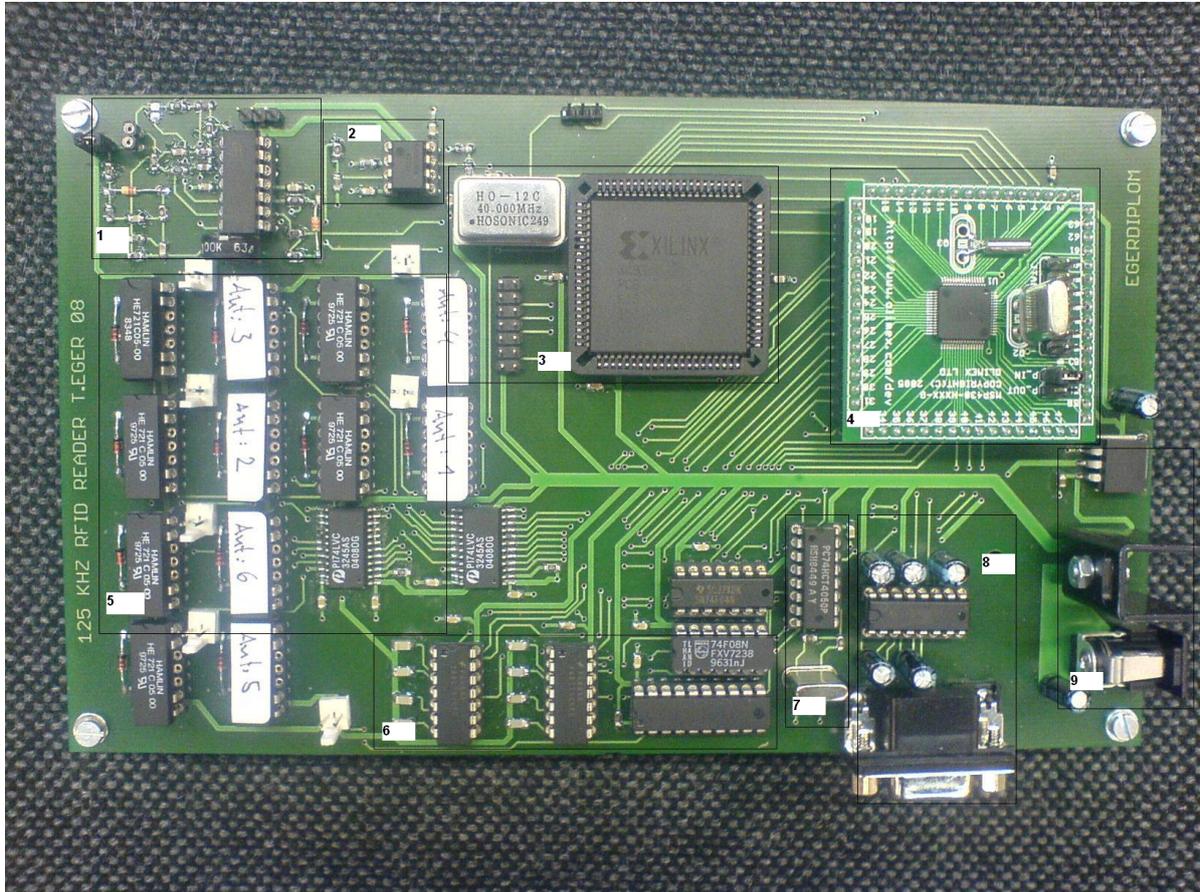


Bild 3.28.: Darstellung der entworfenen Readerplatine

Die folgenden Baugruppen werden gesondert hervorgehoben:

1. Analoge Filterkette
2. Spannungskomparatorschaltung
3. CPLD
4. MSP430 Mikrocontroller
5. Reed Relais zur Antennenselektion
6. Bustreiber zur Schwingkreis Anregung

7. Takterzeugung zur Bustreiberanregung
8. Serielle Schnittstelle
9. Spannungsversorgung der Platine

4. Funktionsnachweis des Readers

4.1. Ermittlung der Spannungsamplitude und des Stromflusses an der Antennenspule

Bei dieser ersten Messreihe wird der Verlauf der Spannung am Referenzpunkt des Schwingkreises und der Stromverlauf durch die Antennenspule mit Hilfe eines Oszilloskops ermittelt. Es wird hier jeweils eine Readerseite aktiviert und anschließend werden die zu ermittelnden Größen festgehalten.

Der Stromfluss wird mit Hilfe einer Strommesszange gemessen. Diese ist so eingestellt, dass jeweils $2mA$ Stromstärke einer Spannung von $1mV$ entsprechen.

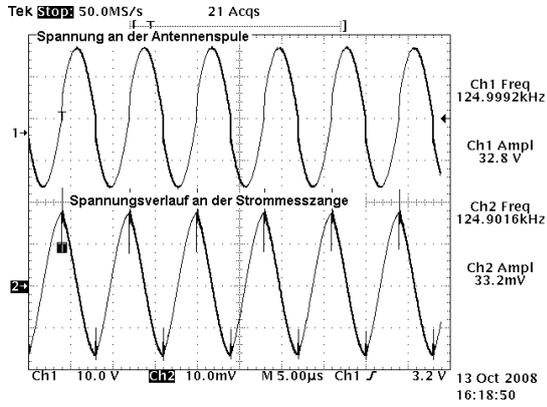
Die Messung erfolgt unmittelbar nach der jeweiligen Aktivierung der gewünschten Readerseite. Dies bedeutet, dass bereits die Kapazitätsanpassung erfolgt ist und somit die bestmögliche Stufe der Resonanzanpassung erreicht wurde. Es befindet sich kein Transponder im Einflussbereich des Readerfelds.

Das Bild 4.1. zeigt die mit Hilfe des Oszilloskops ermittelten Spannungsverläufe. Der erste Kanal zeigt den Verlauf der Spannungsamplitude an der Antennenspule. Im zweiten Kanal wird der ermittelte Spannungsverlauf abgebildet, der den Stromfluss durch die Antenne beschreibt.

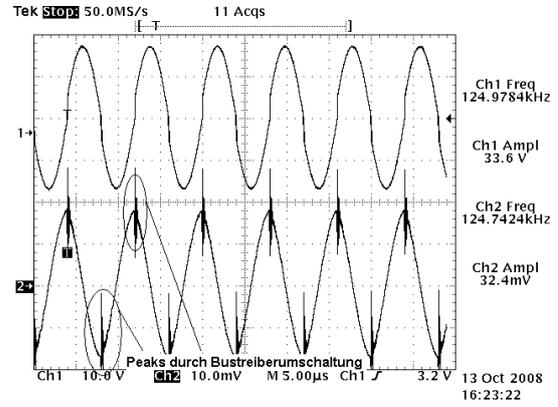
Zur besseren Übersicht werden die ermittelten Werte in einer gemeinsamen Tabelle 4.1. zusammengefasst. Die Reihenfolge der Auflistung orientiert sich an den geometrischen Ausmaßen der jeweiligen Seitenflächen. Es werden die Seitenflächen untereinander aufgeführt, die die gleiche Antennenfläche besitzen. Aus diesem Grund folgt auf die Seitenfläche 1 die Seitenfläche 4.

Das unter "Ausgang CPLD" aufgeführte Ergebnis beschreibt das Ausgangssignal welches vom CPLD gesetzt wurde. Dieses Signal beschreibt das Ergebnis der Resonanzanpassung. Bei allen Seitenflächen liegt das angezeigte Ergebnis im mittleren Bereich der berechenbaren Distanzwerte. Die ermittelte zeitliche Differenz liegt hier nahe an den gewünschten $2\mu s$. Dieses Ergebnis entspricht eine Phasendifferenz von 90° . Somit ist der Schwingkreis als abgestimmt zu betrachten.

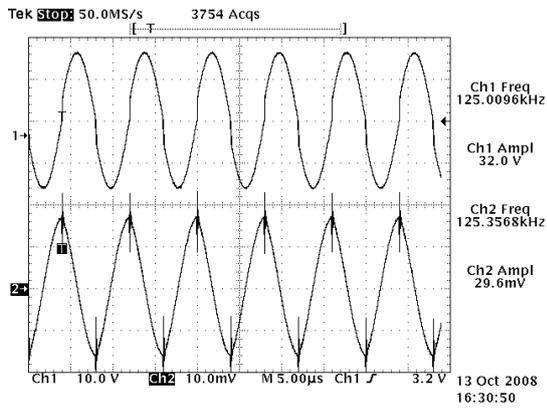
Wie in Bild 4.1. zu erkennen ist, kommt es zu schlagartig auftretenden Stromspitzen



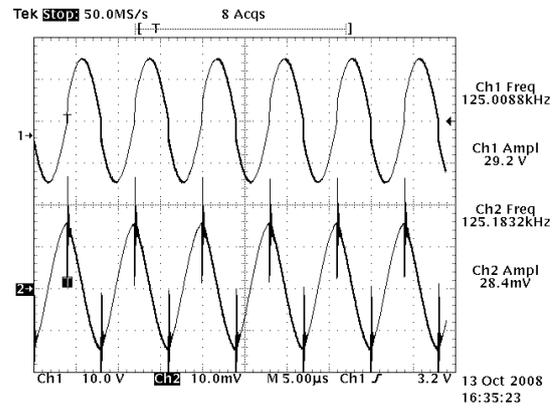
(a) Ergebnisse der Seitenfläche 1



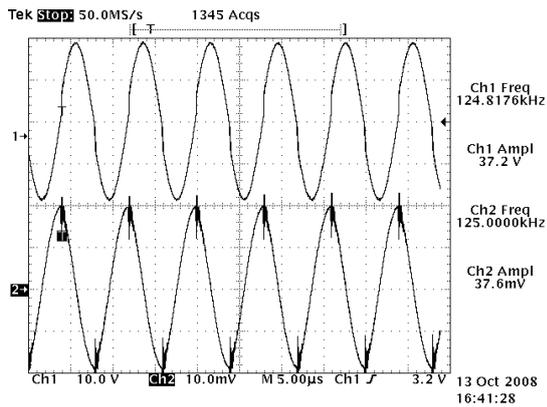
(b) Ergebnisse der Seitenfläche 4



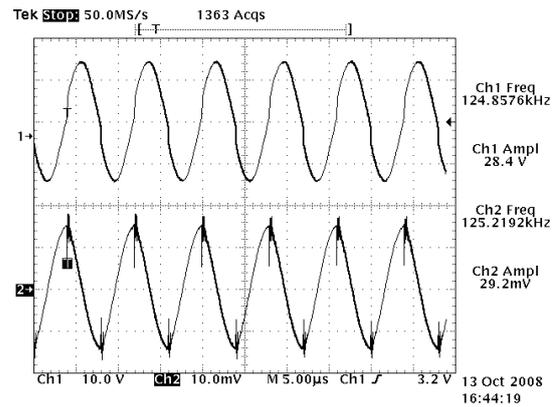
(c) Ergebnisse der Seitenfläche 2



(d) Ergebnisse der Seitenfläche 5



(e) Ergebnisse der Seitenfläche 3



(f) Ergebnisse der Seitenfläche 6

Bild 4.1.: Ergebnisse der Messreihe an der jeweils aktiven Antennenspule

Seitenfläche	Antennenspannung V_{pp}	Antennenstrom mA	Ausgang CPLD
1	32,8	66,4	$DIST_4$
4	33,6	64,8	$DIST_4$
2	32,0	59,2	$DIST_4$
5	29,2	56,8	$DIST_5$
3	37,2	75,2	$DIST_4$
6	28,4	58,4	$DIST_4$

Tabelle 4.1.: Ergebnisse der ersten Messreihe

bei der Messung des Antennenstroms. Diese lassen sich durch die auftretenden Peaks im Verlauf der im Kanal 2 dargestellten Spannung erkennen. Das hier dargestellte Ergebnis lässt sich durch eine Betrachtung der Art der Schwingkreisanregung erklären. Durch die gegenphasige Anregung des Schwingkreises, mittels der vorhandenen Bustreiber, ist jeweils ein Bustreiber aktiv und der gegenüberliegende inaktiv. Die Peaks treten an den Punkten auf, an denen der jeweilige Umschaltzeitpunkt bei der Aktivierung der Bustreiber liegt.

Die in Tabelle 4.1. aufgeführten Werte bezüglich der ermittelten Spannungsamplitude an der Antennenspule bleiben hinter den Erwartungen zurück. Ausgehend von den unter 3.4.2. gewonnenen Simulationsergebnissen, wurden Spannungen von etwa der doppelten Größenordnung erwartet. Aus diesem Grund erfolgt im nächsten Teilabschnitt eine Auswertung der gewonnenen Spannungswerte.

4.1.1. Auswertung der ermittelten Spannungswerte

Unter Berücksichtigung der ermittelten Werte, der am Referenzpunkt des Schwingkreises entnommenen Antennenspannung, lassen sich einige Parameter bestimmen die den jeweiligen Antennenschwingkreis auszeichnen. Es werden daher die Güte Q , die Bandbreite B und der im Antennenschwingkreis vorhandene ohmsche Anteil R rechnerisch ermittelt.

Berechnung der Schwingkreisgüte Q

Die Berechnung der Schwingkreisgüte erfolgt unter Betrachtung des Verhältnisses zwischen anregender Spannung und der gemessenen Ausgangsspannung am Antennenschwingkreis.

Es folgt daher die Beziehung:

$$Q = \frac{U_a}{U_e} \tag{4.1}$$

Berechnung der Bandbreite B

Aus der Kenntnis über die Schwingkreisgüte kann auf die Bandbreite B des Schwingkreises geschlossen werden.

Die Güte Q lässt sich ebenfalls über die folgende Beziehung berechnen:

$$Q = \frac{f_0}{B} \tag{4.2}$$

Bei der Frequenz f_0 handelt es sich um die Resonanzfrequenz des Schwingkreises. Diese ist mit der angetriebenen Sendefrequenz von $f = 125kHz$ gleichzusetzen.

Daher kann die Bandbreite über den Quotienten:

$$B = \frac{f_0}{Q} \text{ [Hz]} \tag{4.3}$$

bestimmt werden.

Berechnung des ohmschen Anteils R im Antennenschwingkreis

Die Bandbreite B lässt sich ebenfalls über eine andere Beziehung bestimmen:

$$B = \frac{R}{2\pi L_{ANT}} \text{ [Hz]} \tag{4.4}$$

Über diese Art der Berechnung kann auf den ohmschen Anteil R im Antennenschwingkreis geschlossen werden.

$$R = B2\pi L_{ANT} \text{ [\Omega]} \tag{4.5}$$

Bei L_{ANT} handelt es sich um den Induktivitätswert der Antennenspule.

Die Ergebnisse der Berechnungen werden zusammenfassend in der Tabelle 4.2. aufgeführt.

Seitenfläche	Güte Q	Bandbreite $B[kHz]$	Ohmscher Anteil $R[\Omega]$
1	6,6	18,9	85,6
4	6,7	18,7	85,0
2	6,4	19,5	91,5
5	5,8	21,6	101,6
3	7,4	16,9	76,6
6	5,7	21,9	104,7

Tabelle 4.2.: Aufistung der berechneten Schwingkreisparameter

Bei der Berechnung des ohmschen Anteils R werden, die in der Tabelle 3.3. aufgeführten, Induktivitätswerte für die Antennenspulen verwendet.

In [RFID03] werden typische Werte für die Schwingkreisgüte Q von Schwingkreisen, welche auf Luftspulen basieren, in einer Größenordnung von 15 – 20 angegeben. Die hier

berechneten Werte liegen unterhalb der beschriebenen Angaben.

Besonderes Augenmerk ist ebenfalls auf die berechneten Werte für den ohmschen Anteil R zu lenken. Deren Wertigkeit fällt höher als bislang erwartet aus. Als bisherige Betrachtungsgrundlage wurde ein ohmscher Anteil von $\approx 4\Omega$ angenommen. Dieser Wert wurde über eine Ermittlung des Widerstandswerts der jeweiligen Antennenspule gewonnen. Wobei diese Messung unter Zuhilfenahme eines LCR-Meters erfolgte.

Die nun erhaltenen Werte liegen deutlich oberhalb der zuvor bestimmten Ergebnisse.

Es wird die Annahme getroffen, dass dieses Ergebnis aus der bei der Implementierung der Readers angewendeten Methode der Schwingkreisanzregung abgeleitet werden kann. Durch das Verfahren der Kapazitätsänderung und der geforderten Möglichkeit der Resonanzfrequenzanpassung besteht die Notwendigkeit, dass eine Kondensatorumschaltung erfolgen kann. Diese Kapazitätsänderung erfolgt über eine Aktivierung oder Deaktivierung der jeweils zugeordneten Bustreiber. Die verwendeten Bustreiber werden im Falle einer nicht Aktivierung in einen hochohmigen Zustand versetzt. Hierbei sind die Ausgänge der Treiber als Tri-State-Ausgänge zu betrachten.

Somit ergibt sich eine Beeinflussung auf den Antennenschwingkreis, da sich die Bustreiber weiterhin direkt im Schwingkreis befinden. Dies hat die Folge, dass sich die Bandbreite durch den vorhandenen Einfluss der Bustreiber erhöht und sich in Abhängigkeit hiervon die Schwingkreisgüte Q vermindert.

4.1.2. Auswertung der Stromflussmessung an der Antennenspule

Ausgehend von den Ergebnissen der Stromflussmessung an der aktiven Antennenspule erfolgt eine auswertende Betrachtung der Messresultate.

Das von den Antennenspulen emittierte elektromagnetische Feld lässt sich, wie bereits beschrieben, über den folgenden Quotienten berechnen:

$$H_R(x) = \frac{I_R N_R r_R^2}{2(r_R^2 + x^2)^{3/2}} \left[\frac{\text{A}}{\text{m}} \right] \quad (4.6)$$

Da nun alle für die Berechnung notwendigen Werte bekannt sind, kann eine Abschätzung des Feldstärkeverlaufs in Abhängigkeit der Entfernung zur Readerantenne erfolgen.

Das Bild 4.2. zeigt das Ergebnis der Berechnung.

Der Stromwert zur Berechnung des Feldverlaufs wurde über eine geometrische Mittelung der unter Tabelle 4.1. aufgeführten Stromstärken ermittelt. Die Mittelwertbildung erfolgt jeweils bei den Seitenflächen, welche über identische räumlichen Abmaße verfügen.

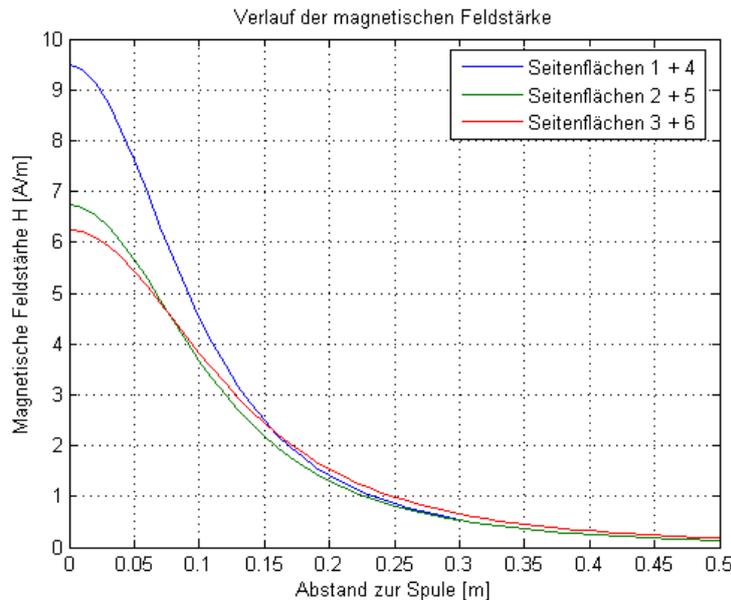


Bild 4.2.: Feldverlauf von $H(x)$ unter Berücksichtigung der ermittelten Antennenparameter

Mittels der Berechnung des Feldverlaufs können erste Rückschlüsse über die Stärke des über die Readerantennen emittierte elektromagnetischen Felds gewonnen werden. Weiterhin lassen sich hierüber erste Erkenntnisse über eine mögliche Energieversorgungsreichweite von Transpondern, welche sich im Einflussbereich des Sendefelds befinden, ableiten.

4.2. Transponderversorgung

4.2.1. Bestimmung des Kopplungsfaktors k

Die vom Readerfeld ausgehende Energie dient zur Versorgung des vorhandenen passiven Transponders (vgl. A.2.). Die Energieübersetzung erfolgt nach dem unter 2.3.4. vorgestellten Prinzip. Wobei dem Kopplungsfaktor k eine besondere Rolle bei der Transponderversorgung zuteil wird.

Dieser Faktor lässt sich über die bereits vorgestellte Formel berechnen:

$$k(x) = \frac{r_R^2 r_T^2}{\sqrt{r_R r_T} \sqrt{(r_R^2 + x^2)^3}} \quad (4.7)$$

Der Kopplungsfaktor k hängt hierbei von der Entfernung zwischen Readerantenne und Transponder und von den jeweils vorhandenen Spulenradien ab. Da die Antennenradien bekannt sind, lässt sich die Wertigkeit von k rechnerisch ermitteln.

Unter Bild 4.3. wird das Ergebnis der Berechnung aufgeführt.

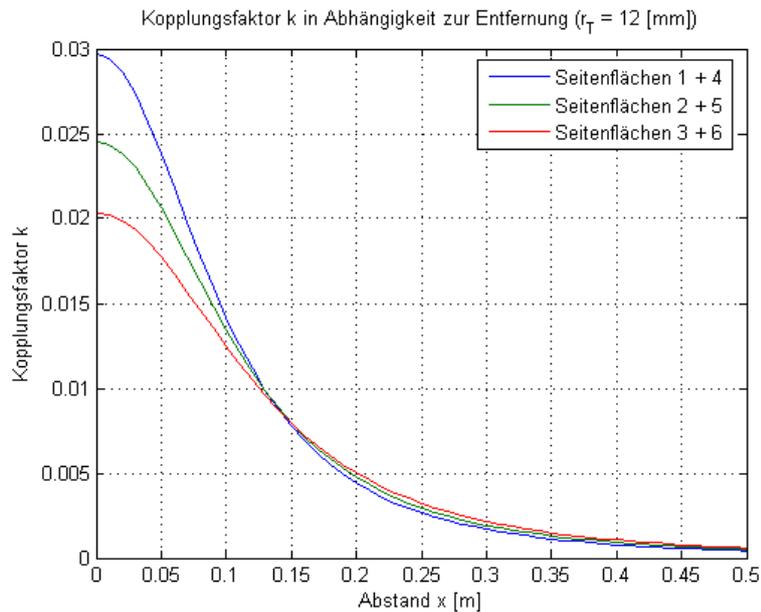


Bild 4.3.: Feldverlauf von $k(x)$ unter Berücksichtigung der realen Antennenradien

In der Tabelle 4.3. werden die als Berechnungsgrundlage verwendeten Spulenradien zusammengefasst.

Benennung	Spulenradius [mm]
$r_{Transponder}$	12
$r_{Seiten14}$	125
$r_{Seiten25}$	142
$r_{Seiten36}$	161

Tabelle 4.3.: Zusammenfassung der jeweiligen Spulenradien

Als Ergebnis dieser Berechnung lässt sich feststellen, dass sich im absoluten Nahbereich Kopplungsfaktoren ergeben, die im einstelligen Prozentbereich liegen. Der Wert von k vermindert sich mit zunehmender Entfernung zur Readerantenne. Ab einer Distanz von $\approx 12,5\text{cm}$ ergeben sich Kopplungsfaktoren im Bereich von einigen Promille.

4.2.2. Abschätzung des Feldstärkenverlaufs $H_T(x)$ am Transponder

Betrachtet man die Ergebnisse der letzten beiden Berechnungen in kombinatorischer Art und Weise, so lässt sich ein Abschätzung des Feldstärkenverlaufs $H_T(x)$ erreichen. Dieser Verlauf beschreibt die vom Transponder nutzbare elektrische Feldstärke.

Für diese Betrachtung werden die Ergebnisse von $H(x)$ multiplikativ mit den Kopplungsfaktor k in Verbindung gesetzt.

Als Berechnungsgrundlage ergibt sich somit:

$$H_T(x) = H(x) * k \left[\frac{\text{A}}{\text{m}} \right] \quad (4.8)$$

Bei der Berechnung des Kopplungsfaktors k wird hier die räumliche Beziehung x nicht weiter betrachtet. Da der Feldstärkenabfall mit der Entfernung bereits durch die Berechnung von $H(x)$ berücksichtigt wird.

Daher wird der Kopplungsfaktor k zu:

$$k = \frac{r_R^2 r_T^2}{\sqrt{r_R r_T} \sqrt{r_R^2}^3} \quad (4.9)$$

definiert.

Das Ergebnis dieser Betrachtung wird in Bild 4.4. aufgeführt.

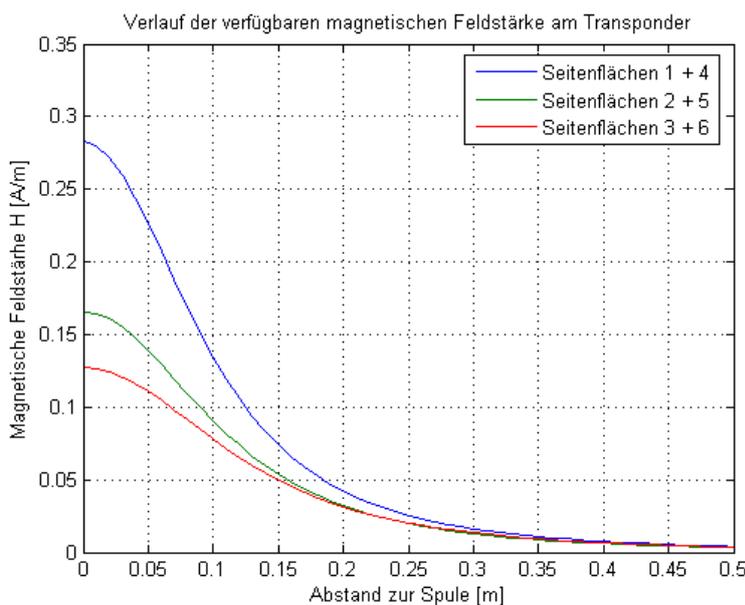


Bild 4.4.: Ermittelter Feldverlauf von $H_T(x)$

Der hier ermittelte Feldverlauf ist lediglich als Abschätzung zu betrachten. In der Diplomarbeit von Tobias Krannich [DIPL01] wurde eine minimale Ansprechfeldstärke H_{min} für den Transponder ermittelt. Diese Feldstärke beschreibt die untere Grenze der Transponderversorgung. Unterhalb dieses Werts wird der Transponder nicht mehr ausreichend mit Energie versorgt, um dessen Funktion sicher stellen zu können. Dieser Wert wird mit $H_{min} = 0,577 \text{ A/m}$ angegeben.

Die hier berechneten Werte sind von der Größenordnung nahezu identisch mit dem angegebenen Wert für die minimale Ansprechfeldstärke. Dennoch liegen sie im gesamten

Betrachtungsspielraum unterhalb des geforderten Werts.

Als Konsequenz hierauf ließe sich ableiten, dass die von Readerfeld ausgehende Energie in keinem Fall ausreichend wäre, um den Transponder versorgen zu können. Dieser Effekt lässt sich durch eine praktische Überprüfung nicht in dieser absoluten Schärfe nachweisen. Daher ist das Berechnungsergebnis nur als Näherung zu betrachten.

4.2.3. Praxistest unter Einsatz des vorhandenen Transponders

Nach den aufgeführten theoretischen Betrachtungen erfolgt nun ein praktischer Test der Readers.

Zu diesem Zweck wird der Transponder im Sendefeld der jeweils aktiven Readerseite platziert. Zur Programmierung des Transponders kommt eine modifizierte Version der im Rahmen der Diplomarbeit [DIPL01] erstellten Firmware zum Einsatz. Dabei beginnt der Transponder mit dem Senden eines festen Datenworts, sobald er sich im Einflussbereich des Readerfelds befindet. Der Betrachtung des genauen Inhalts des Datenworts wird bei dieser Erprobung keine besonders gewichtige Rolle zugeteilt. Vielmehr wird geprüft, ob sich überhaupt ein übertragenes Datenwort am Ausgang der auf der Readerplatine vorhanden Filterkette ermitteln lässt. Als Nachweis dessen, wird der Spannungsverlauf am Ausgang der Filterkette mit einem Oszilloskop festgehalten.

Das Bild 4.5. zeigt das übertragene Datenwort als beispielhafte Messung an der Seitenfläche 1 der Readerbox.

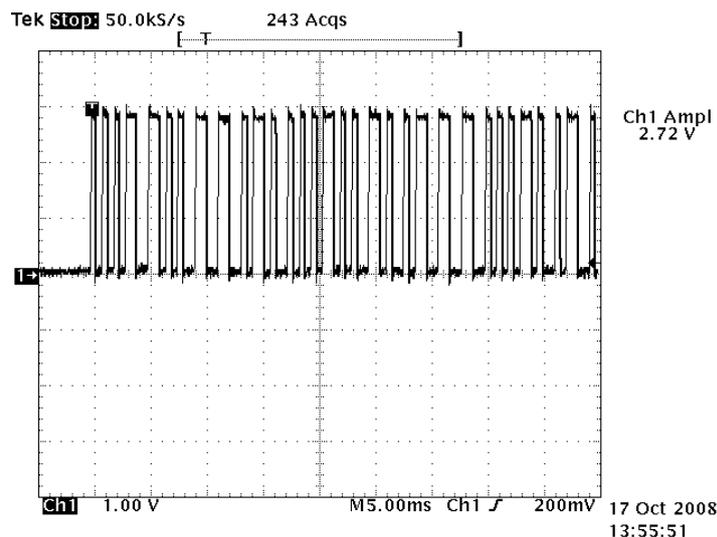


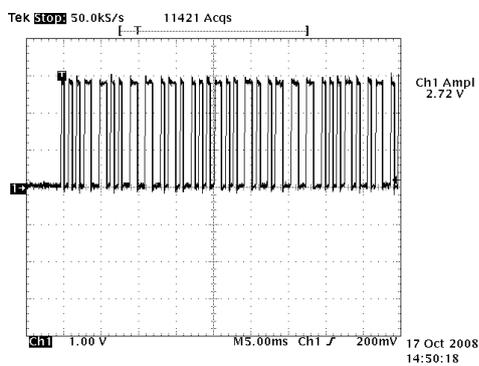
Bild 4.5.: Fehlerfrei übertragenes Datenwort am Ausgang der Filterkette

Bei den empfangenen Daten handelt es sich um Daten, die nach der Methode der Manchesterkodierung übertragen werden. Die Übertragung erfolgt über ein vorgegebenes Pro-

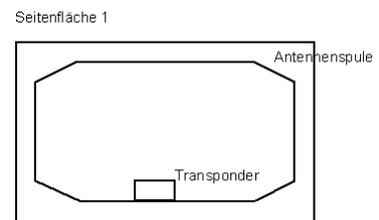
tokoll. Auf dessen Beschreibung wird in dieser Arbeit verzichtet. Für eine Erläuterung des eingesetzten Protokolls sei an dieser Stelle auf die Diplomarbeit [DIPL01] verwiesen.

Leider ist dieses Verhalten der fehlerfreien Übetragung an eng gesteckte Bedingungen gebunden. Innerhalb der Readerbox lässt sich kein Nachweis über die Übetragung des Datenworts erbringen. Das dargestellte Datenwort wird nur dann am Ausgang der Filterkette nachweisbar, wenn sich der Transponder in unmittelbarer Nähe zur Antennenspule befindet. Genauer gesagt gelingt dies nur, wenn sich der Transponder außerhalb der Readerbox auf der Spulenflächenebene befindet.

Zur Verdeutlichung sei auf die im folgenden Anschluss aufgeführte Bilderreihe hingewiesen.

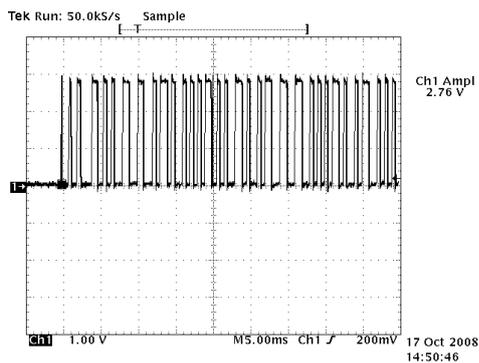


(a) Datenwort am Filterkettenausgang

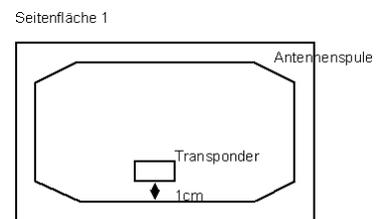


(b) Positionierung des Transponders

Bild 4.6.: Empfangenes Datenwort bei einem Transponderabstand von 0cm



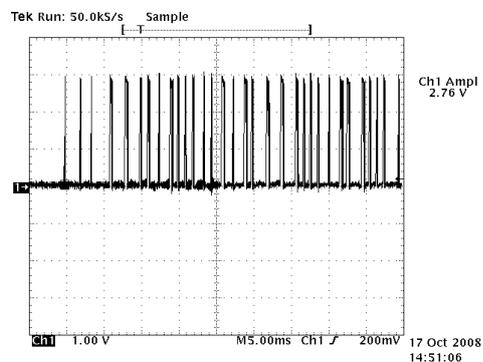
(a) Datenwort am Filterkettenausgang



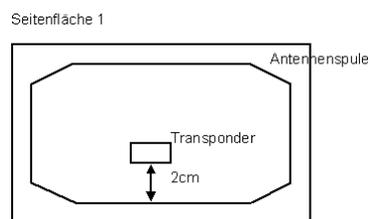
(b) Positionierung des Transponders

Bild 4.7.: Empfangenes Datenwort bei einem Transponderabstand von 1cm

Die hier aufgeführte Messreihe wurde exemplarisch an der Seitenfläche 1 der Readerbox durchgeführt. Die gezeigten Ergebnisse lassen sich auf die übrigen Seitenflächen übertragen.

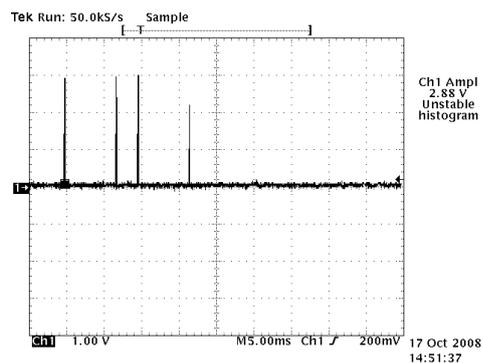


(a) Datenwort am Filterkettenausgang

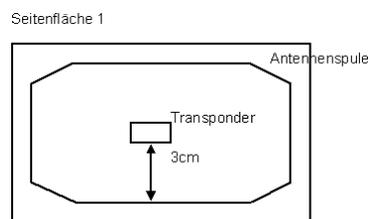


(b) Positionierung des Transponders

Bild 4.8.: Empfangenes Datenwort bei einem Transponderabstand von 2cm



(a) Datenwort am Filterkettenausgang



(b) Positionierung des Transponders

Bild 4.9.: Empfangenes Datenwort bei einem Transponderabstand von 3cm

Bereits in Bild 4.7 lassen sich erste Anzeichen einer fehlerhaften Übertragung erkennen. Wie in Bild 4.8. ersichtlich, wird das Datenwort ab einer Distanz von 2cm definitiv nicht mehr fehlerfrei am Ausgang der Filterkette dargestellt. Bei einer Distanz von 3cm zwischen Readerspule und Transponder lässt sich kein zusammenhängendes Datenwort mehr erkennen.

Die aufgeführten Ergebnisse bleiben weit hinter den getroffenen Erwartungen zurück. Der Umstand, dass innerhalb der Readerbox keine Versorgung des vorhandenen Transponders erreichbar ist, wurde so nicht vorausgesehen.

4.3. Filterkette

Um zu prüfen, ob die erstellte Filterkette die erforderliche Funktionalität bereitstellt, erfolgt an dieser Stelle eine messtechnische Erprobung der Filterkettenfunktion.

Zu diesem Zweck wird die Readerplatine ohne die zugehörige Readerbox betrieben. Dies bedeutet, dass keine der vorhandenen Antennenspulen verwendet wird. Vielmehr wird direkt am Eingang der Filterkette ein amplitudenmoduliertes Signal eingespeist. Dies geschieht unter Zuhilfenahme eines geeigneten Funktionsgenerators. Wie bereits in Bild 3.25. aufgeführt, befindet sich vor dem Eingang der Filterkette eine Anschlussmöglichkeit für eine externe Antenne. Über diesen Steckplatz wird die Generatorspannung zugeführt. Die zugeführte Spannung wird zusammen mit dem Spannungssignal am Ausgang der Filterkette oszilliert.

Das Bild 4.10. zeigt eine Darstellung des Ergebnisses.

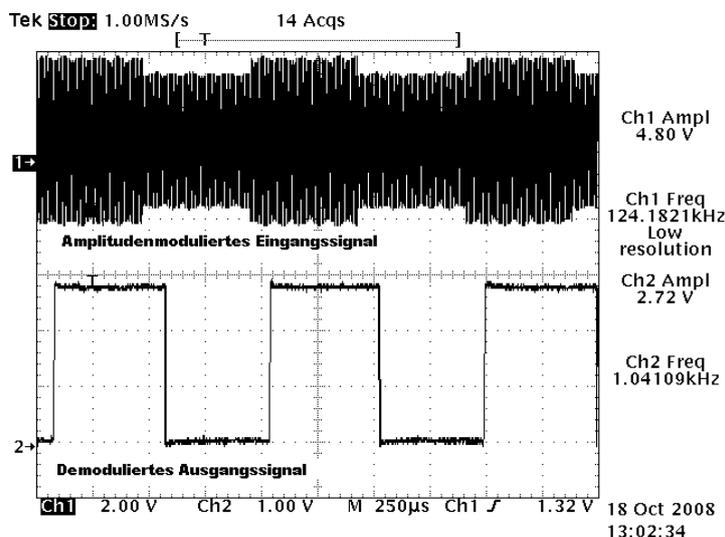


Bild 4.10.: Oszillierte Spannungsverläufe zur Funktionsprüfung der Filterkette

Der obere Signalverlauf beschreibt die Spannung am Eingang der Filterkette. Wobei die untere Spannung den Verlauf des Ausgangssignals der Demodulationskette darstellt. Die Eingangsspannung setzt sich aus einer Sinusschwingung mit einer Frequenz von $f_{Tr} = 125kHz$ und einem aufmodulierten Signal zusammen. Bei diesem Signal handelt es sich um eine Rechteckspannung mit einer Frequenz von $f_{Sig} = 1kHz$. Dieses Signal lässt sich am Ausgang der Filterkette nachweisen, wie der untere Spannungsverlauf in Bild 4.10. beweist.

Der Modulationsgrad beträgt bei dieser Messung: $m = 7\%$. Dieser Wert lässt sich wie folgt bestimmen:

$$m = \frac{U_{Tr} - U_{Sig}}{U_{Tr} + U_{Sig}} \quad (4.10)$$

Die zur Berechnung erforderlichen Spannungswerte wurden mit Hilfe des Oszilloskops ermittelt.

Es folgt daher:

$$m = \frac{3,7V - 3,2V}{3,7V + 3,2V} = 0,072 \approx 7\% \quad (4.11)$$

Dieser Modulationsgrad wurde ausgewählt, da bis zu diesem Wert für m der Verlauf der Eingangsspannung eindeutig am Oszilloskop darstellbar bleibt. Eine weitergehende Überprüfung hat ergeben, dass die Filterkette für einen Modulationsgrad im Bereich von $m = 3\% - 100\%$ zuverlässige Ergebnisse liefert.

Als Ergebnis dieser Funktionsprüfung wird abgeleitet, dass die Filterkette die Aufgabe der Demodulation des Antennensignals in zufriedenstellender Weise erfüllt. Somit scheidet sie als mögliche Fehlerquelle bei der verminderten Empfangsmöglichkeit des Transpondersignals aus.

Aufgrund des vorhandenen Zeitrahmens besteht nicht die Möglichkeit einer weiterführenden messtechnische Erprobung des Zusammenspiels zwischen dem vorhandenen Transponder und dem entworfenen Lesegerät. Diese Aufgabe bleibt daher weiterführenden Arbeiten vorbehalten.

5. Fazit und Ausblick

Die generelle Realisierung des Lesegeräts ist als erfolgreich zu bezeichnen. Durch diese Arbeit konnte gezeigt werden, dass sich die im Anfangsbereich dargestellten Methoden zur Beeinflussung der Parameter des einzusetzenden Schwingkreises erfolgreich miteinander verbinden lassen.

Durch die dargestellte Art der Schwingkreisanregung lässt sich eine Anpassung der vorhandenen Kapazitäten erreichen. Dies hat einen entscheidenden Effekt auf die im Antennenschwingkreis ermittelbare Größe der Spannungsamplitude und damit ebenso auf die emittierte Sendefrequenz. Es sei aber an dieser Stelle darauf hingewiesen, dass diese Art der Realisierung durchaus Effekte auf die jeweiligen Eigenschaften des Sendeschwingkreises hat. So konnte beispielsweise nachgewiesen werden, in welcher Gestalt sich dieses Verfahren auf die Bandbreite und die Güte des Antennenschwingkreises auswirkt. Diese Effekte sind bei weiterführenden Realisierungen zu bedenken.

Der experimentelle Nachweis der Ergebnisse, welche sich durch die Messung der Phasenlage der zur Bewertung der Resonanzanpassung herangezogenen Spannungsverläufe ergeben, konnte aus zeitlichen Gründen nicht vollständig erbracht werden. Im Rahmen der vorliegenden Arbeit konnte aber dennoch eine Realisierung vorgestellt werden, durch diese sich aussagekräftige Bewertungsaspekte gewinnen lassen.

Der hierfür vorgesehen Zeitrahmen wurde unter anderem auf eine Anpassung der zur Demodulation der Empfangsdaten verwendeten analogen Filterkette verwendet.

Dies wurde als sinnvoll erachtet, nachdem sich herausgestellt hat, in welchem Bereich sich eine ausreichende Versorgung des vorhandenen Transponders erreichen lässt.

Dieser Teil der anfänglichen Zielsetzung hat sich, bei der dargestellten Implementierung, als nicht umsetzbar erwiesen. Die anfängliche Forderung der vollständigen Versorgung des Transponders im vorgesehenen Arbeitsraum konnte nicht umgesetzt werden. Der zu Nachweiszwecken einsetzbare Transponder lässt sich lediglich in einem engen Bereich in unmittelbarer Nähe zur Sendeantenne betreiben. In der bereits mehrfach zitierten Vorararbeit [DIPL01] ließ sich eine Versorgung des Transponders über einen Bereich nachweisen, welcher etwa der dreifachen Abstandsstrecke des hier ermittelten Versorgungsbereichs entspricht.

Im Gegensatz zur Vorarbeit wurden bei der hier aufgeführten Implementierung zwar leistungstärkere Treiber zur Schwingkreisanregung verwendet, dennoch ergibt sich aus dieser Entscheidung nicht zwangsläufig ein erweiterter Versorgungsbereich. Bei der vorausgegangenen Arbeit wurde ein höherer Induktivitätswert für die Antennenspule verwendet.

Weiterhin fällt hier die Spulenfläche geringer aus. Es ergibt sich dabei eine höhere Anzahl der vorhandenen Spulenwicklungen, als bei der hier dargestellten Realisierung. Es wird vermutet, dass sich durch diese Konstellation ein stärkeres elektrisches Feld zur Transponderversorgung erzeugen lässt.

Diese Behauptung stützt sich auf die Ergebnisse der Vergleichsrechnung des Einflusses des Spulenradius auf die Stärke des von der Antennenspule emittierten elektromagnetischen Felds im Abschnitt der Grundlagenbetrachtungen zum Reader (siehe 2.3.3.). In dem dort aufgeführten Ergebnis (Bild 2.12.) wird ersichtlich, dass die Feldstärke im Nahbereich der Spule umso höher ist, je weniger sich die jeweiligen Antennenradien, zwischen Transponderspule und Readerspule, unterscheiden. Das jeweilige Verhältnis der beiden Spulenradien hat ebenso einen direkten Einfluss auf die Größe des Kopplungsfaktors und somit auch auf den möglichen Grad der Energieübertragung. Bei der Bestimmung dieses Beschreibungsfaktors lässt sich ebenfalls ein deutlich höheres Ergebnis in den Fällen erzielen, in denen sich, die als Berechnungsgrundlage herangezogenen Radien weniger unterscheiden. Dieser Effekt lässt sich unter dem Abschnitt 2.3.4. in Bild 2.17. erkennen. Anscheinend sind diese Erkenntnisse zu Beginn der Realisierung nicht im ausreichenden Maße berücksichtigt worden. So dass sich nun die dargestellten Ergebnisse einstellen. Es wird dazu angeregt, diese Umstände in weiterführenden Arbeiten in eine besonderen Maß bei Realisierung mit einzubeziehen. Als Vorschlag zur Effizienzsteigerung wird eine Verringerung der eingesetzten Antennenradien genannt. So könnte beispielsweise die Anzahl der pro Seite der Readerbox vorhandenen Antennenspulen verdoppelt werden. Auch unter dem Hintergrund des daraus resultierenden erhöhten Schaltungsaufwands. Eventuell könne dieser nachteilige Effekt durch einen deutlich erhöhten Versorgungsbereich des Transponders aufgewogen werden.

Es ist weiterhin denkbar, eine Steigerung der Versorgungseffizienz durch eine Größenänderung der zur Abstimmung vorgesehenen Kapazitäten zu erreichen. Es hat sich gezeigt, dass der ermittelbare Antennenstrom steigt, je mehr Einzelkapazitäten benötigt werden, um die erforderliche Gesamtkapazität zur Schwingkreisabstimmung bereitzustellen. Durch eine Erhöhung des Stromflusses lässt sich ebenfalls eine Steigerung der emittierten Feldstärke erreichen. Würden daher mehr Kondensatoren mit einer etwas kleineren Kapazität verwendet, so würden jeweils mehr Treiberstufen aktiviert. Aus dieser Maßnahme würde eine gewisse Steigerung des zur Verfügung gestellten Anregestroms resultieren. Die bisher aufgeführten Anregungen zur Erhöhung des möglichen Versorgungsbereichs, beziehen sich lediglich auf eventuelle Änderungen am Reader. Es sei zusätzlich noch auf einige Änderungsmöglichkeiten bei der Realisierung des Transponders hingewiesen.

Der Transponder verwendet eine einfache Luftspule aus Kupferdraht zur Energiegewinnung. Eventuell könnte die Energiemenge zur Transponderversorgung durch den Einsatz einer Spule gesteigert werden, die um einen Ferritkern gewickelt ist. Spulen die auf diese Art realisiert sind, verfügen zwar über eine geringere Grundfläche, aber dennoch ergibt sich eine höhere Spulengüte durch den vorhandenen Spulenkern. Durch diese Maßnahme

ließe sich eine Effizienzsteigerung bei der Ausnutzung der Entnahme der im Sendefeld enthaltenen Energiemenge erreichen.

Ebenso werden Verbesserungsmöglichkeiten im Bereich Sendedatenerzeugung am Transponder gesehen. Zur Übertragung der Daten zum Lesegerät wird die Methode der Lastmodulation verwendet. Bei diesem Verfahren werden die zu übertragenden Daten über eine zyklische Beeinflussung des Sendefelds über einen Lastwiderstand erzeugt. Das Schalten des Widerstands setzt eine gewisse Energiemenge voraus, welche der Versorgungsenergie des Transponders entzogen wird. Ist die hierfür notwendige Energie zu groß, so kann nicht ausgeschlossen werden, dass die resultierende Versorgungsenergie zu gering ist, um den Transponder weiterhin zu versorgen. In diesem Fall ergibt sich eine Unterversorgung des Transponders.

Möglicherweise tritt genau dieser Fall bei den verwendeten Transponder ein. Diese Möglichkeit lässt sich nach den bisherigen Erkenntnissen nicht vollends ausschließen. Es sei daher auf eine weitere Untersuchung der Sendedatenmodulation des Transponders hingewiesen.

Abschließend sei betont, dass die Konzeption und Realisierung beider zuvor erwähnten Bauteile als generell richtig und erfolgreich anzusehen sind. Es besteht aber weiterhin noch einiger Bedarf der Optimierung, um ein zuverlässiges Zusammenarbeiten der beiden Komponenten gewährleisten zu können.

Durch die hier aufgeführten Ansätze wurde der Versuch unternommen, einige Impulse für eventuell nachfolgende Arbeiten zu geben.

Literaturverzeichnis

- [CPLD01] Xilinx, *XC95108 In-System Programmable CPLD*,
http://www.xilinx.com/support/documentation/data_sheets/DS066.pdf.
- [DIPL01] Tobias Krannich, *Experimentalsystem für einen Sensor-Controller mit drahtloser Energie-und Datenübertragung*, HAW Hamburg, August 2008.
- [EAG01] Cadsoft.de, *eagle-win-ger-4.16r2.zip*,
<ftp://ftp.cadsoft.de/eagle/program/4.16r2>.
- [EAG02] Cadsoft.de, *Eagle Handbuch Version 4.1*,
<ftp://ftp.cadsoft.de/eagle/program/4.16r2/manual-ger.pdf>.
- [ECL01] Eclipse.org, *eclipse-SDK-3.3.2-win32.zip*,
<http://download.eclipse.org/eclipse/downloads/>.
- [FAI01] Fairchild Semiconductor, *74ACT125, Quad Buffer with 3-STATE Outputs*,
<http://www.fairchildsemi.com/pf/74/74ACT125.html>.
- [FAI02] Fairchild Semiconductor, *74ACT245, Octal Bidirectional Transceiver with 3-STATE Inputs/ Outputs*, <http://www.fairchildsemi.com/pf/74/74ACT245.html>.
- [GCC01] Sourceforge.net, *GCC toolchain for MSP430*,
http://sourceforge.net/project/showfiles.php?group_id=42303.
- [HAM01] Hamlin, *HE700 D.I.L Relais Features and Benefits*,
<http://www.hamlin.com/specsheets/HE700.pdf>.
- [INT01] Deutsche Wikipedia, *Radio Frequency Identification*,
<http://de.wikipedia.org/wiki/RFID#Entwicklungsgeschichte>,
Datum: 03.06.2008.
- [MAT01] FH Regensburg, *Matlab Eine Einführung*,
<http://homepages.fh-regensburg.de/~wah39067/Matlab/MTut2-1.pdf> .
- [MAX01] Maxim, *+5V-Powered, Multichannel RS-232 Drivers/Receivers*,
<http://www.datasheetcatalog.org/datasheet/maxim/MAX220-MAX249.pdf>.

- [MSP01] Texas Instruments, *MSP430x1xx Family User's Guide*, <http://focus.ti.com/lit/ug/slau049f/slau049f.pdf>.
- [MSP02] OLIMEX Ltd., *MSP430-Hxxx-E Header Board for MSP430F14x and MSP430F41x Microcontrollers*, <http://www.olimex.com/dev/pdf/msp430-hxxx-e.pdf>.
- [MSP03] OLIMEX Ltd., *MSP430-JTAG for programming and flash emulation with MSP430 Microcontrollers*, <http://www.olimex.com/dev/pdf/msp430-jtag-d.pdf>.
- [NS01] National Semiconductor, *LM311 Voltage Comparator*, <http://www.national.com/mpf/LM/LM311.html>.
- [NS02] National Semiconductor, *LM1086 , 1,5 A Low Dropout Positive Regulators*, <http://cache.national.com/ds/LM/LM1086.pdf>.
- [PAT01] Thomas Giesler, Frank Böh, Wolfgang Tobergte, *Schaltungsanordnung zum Einstellen der Resonanzfrequenz*, Offenlegungsschrift DE 197 55 250 A1 beim Deutschen Patent- und Markenamt, 01.07.1999.
- [PER01] Pericom, *PI7LVC3245A, 8-Bit Dual Supply Bus Transceiver with 3-State Outputs*, <http://www.pericom.com/pdf/datasheets/PI74LVC3245A.pdf>.
- [PHI01] Philips Semiconductors, *74HC/74HCT4060, 14-stage binary ripple counter with oscillator*, http://www.datasheetcatalog.com/datasheets_pdf/7/4/H/C/74HCT4060N.shtml.
- [PSPICE01] HAW Hamburg, *91pspstu.exe*, http://users.etech.haw-hamburg.de/users/mueller/LE/PSPICE%20student%20version%209_1/.
- [PSPICE02] Spicelab.de, *Robert Heinemanns PSPICE-Seiten*, <http://spicelab.de>.
- [RFID01] Bill Glover, Himanshu Bhat, *RFID Essentials*, O'Reilly Verlag, 1. Auflage, 2006.
- [RFID02] Klaus Finkenzeller, *RFID-Handbuch Grundlagen und praktische Anwendungen induktiver Funkanlagen, Transponder und kontaktloser Chipkarten*, Carl Hanser Verlag, 4. Auflage, 2006.
- [RFID03] Microchip, *AN678, RFID Coil Design*, <http://ww1.microchip.com/downloads/en/AppNotes/00678b.pdf>.
- [RFID04] EM Microelectronic - Marin SA, *AppNote 411 RFID Made Easy*, <http://www.emmicroelectronic.com/webfiles/product/rfid/an/an411.pdf>.

- [ST01] ST Microelectronics, *LS7800 Series, Positive Voltage Regulators*,
http://www.datasheetcatalog.com/datasheets_pdf/L/7/8/0/L7805.shtml.
- [STUD01] Thorsten Eger, Studienarbeit, *Entwicklung, Aufbau und Test eines RFID-Readers mit Feldvektorsteuerung*, HAW Hamburg, April 2008.
- [VHDL01] Xilinx.com, *ISEWebPack 9.1*,
http://www.xilinx.com/ise/logic_design_prod/webpack.htm.
- [VHDL02] HAW Hamburg, *Anleitung_ISE8_2_ModelSim_V3.pdf*,
<http://users.etch.haw-hamburg.de/users/ITLabor>.

Listingverzeichnis

C.1. Berechnung des Feldverlaufs von $H(x)$ für 3 unterschiedliche Spulenradien	99
C.2. Berechnung des Feldverlaufs von $H(r)$ bei einer Entfernung von $10cm$ zur Antenne	99
C.3. Verlauf des Kopplungsfaktors k für 3 unterschiedliche Spulenradien	100
C.4. Berechnung des Spuleninduktivitätssteigerungsverlaufs der einzelnen Seitenflächen des Readers	100
C.5. Berechnung des Feldverlaufs von $H(x)$ unter Berücksichtigung der realen Antennenparameter	102
C.6. Berechnung des Feldverlaufs von $k(x)$ unter Berücksichtigung der realen Antennenradien	102
C.7. Berechnung des Feldverlaufs von $H_T(x)$	103
C.8. VHDL Code zum Resonanzabgleich	104
C.9. Do Datei zur Simulation des VHDL Codes	107
C.10. UCF Datei zur Definition der Ein- und Ausgänge des CPLDS	107
C.11. Synthesereport des entworfenen VHDL Models	108
C.12. Header File	112
C.13. Main File	114
C.14. Function File	116
C.15. ISR File	123

Bildverzeichnis

2.1. Bestandteile eines RFID-Grundsystems	5
2.2. Übersicht der häufig verwendeten Signalcodierungsarten [RFID02]	7
2.3. Ein mittels ASK modulierte Signal [RFID02]	8
2.4. Übersicht der Normen für RFID-Systeme [RFID02]	9
2.5. Blockschaltbild zur Methode der Amplitudenmessung	10
2.6. Blockschaltbild zur Methode der Phasenmessung	11
2.7. Blockschaltbild zur Messmethode nach Giesler, Böh und Tobergte	13
2.8. Schaltungsanordnung zum Reihenschwingkreis	15
2.9. Frequenzgang des simulierten Reihenschwingkreises	16
2.10. Beispielhafte Schaltungsanordnung zur Schwingkreisanregung	17
2.11. Darstellung der Spannungsverläufe der Beispielschaltung zur Schwingkreisanregung	17
2.12. Feldverlauf von $H(x)$ bei 3 unterschiedlichen Spulenradien	18
2.13. Feldverlauf von $H(r)$ bei einer Entfernung von 10cm zur Antenne	19
2.14. Schematische Darstellung des Transformatorprinzips	20
2.15. Schaltungsanordnung zum Transformatorprinzip	21
2.16. Spannungsverlauf der simulierten Transformatorschaltung	21
2.17. Verlauf des Kopplungsfaktors k für 3 Spulenradien	22
3.1. Darstellung der Systemkomponenten	24
3.2. Räumliche Abmaße der Readerbox	25
3.3. Verknüpfung der Zuordnungszahl mit den Seitenflächen der Readerbox	25
3.4. Parameterzuordnung zur Berechnung der Antennenspulen [RFID03]	26
3.5. Verlauf der berechneten Induktivitätszunahme der einzelnen Seitenflächen des Readers	27
3.6. Seitenflächen 1 und 4 der Readerbox	29
3.7. Seitenflächen 2 und 5 der Readerbox	29
3.8. Unterseite der Readerbox	29
3.9. Oberseite der Readerbox	30
3.10. Blockschaltbild zur Readerplatine	30
3.11. Blockschaltbild zum angewendeten Resonanzabgleich	31
3.12. Schaltplan der verwendeten Schaltung zur Spannungsgleichrichtung	32
3.13. Simulationsergebnis der Schaltung zur Spannungsgleichrichtung	33
3.14. Beispielschaltung zur Kapazitätsänderung	35

3.15. Simulationsergebnis zur Beispielschaltung der Kapazitätsänderung	36
3.16. Schaltplan der verwendeten Komparatorschaltung	37
3.17. Simulationsergebnis der Komparatorschaltung	37
3.18. Oszillierte Spannungsverläufe zur Komparatorschaltung	38
3.19. Beschreibung der Ein- und Ausgangssignale des CPLD	39
3.20. Simulationsergebnis des VHDL Codes	41
3.21. Schaltung zur Erzeugung der Taktfrequenz zur Bustreiberanregung	43
3.22. Blockschaltbild zur Modulation	44
3.23. Beispielschaltung zur Modulation	44
3.24. Simulationsergebnis der Beispielschaltung zur Modulation	45
3.25. Schaltplan der analogen Filterkette zur Demodulation	46
3.26. Beispielschaltung zur Antennenselektion	48
3.27. Programmablaufplan zur Readersoftware	50
3.28. Darstellung der entworfenen Readerplatine	51
4.1. Ergebnisse der Messreihe an der jeweils aktiven Antennenspule	54
4.2. Feldverlauf von $H(x)$ unter Berücksichtigung der ermittelten Antennenparameter	58
4.3. Feldverlauf von $k(x)$ unter Berücksichtigung der realen Antennenradien	59
4.4. Ermittelter Feldverlauf von $H_T(x)$	60
4.5. Fehlerfrei übertragenes Datenwort am Ausgang der Filterkette	61
4.6. Empfangenes Datenwort bei einem Transponderabstand von $0cm$	62
4.7. Empfangenes Datenwort bei einem Transponderabstand von $1cm$	62
4.8. Empfangenes Datenwort bei einem Transponderabstand von $2cm$	63
4.9. Empfangenes Datenwort bei einem Transponderabstand von $3cm$	63
4.10. Oszillierte Spannungsverläufe zur Funktionsprüfung der Filterkette	64
A.1. Blockschaltbild des MSPF1232-Controllers [MSP02]	80
A.2. Blockschaltbild des MSPF169-Controllers [MSP02]	81
A.3. Darstellung des MSP-H169 Header-Boards [www.olimex.com]	81
A.4. JTAG-Programmier-Kabel zum Anschluss an eine Parallel-Schnittstelle [www.olimex.com]	82
A.5. Blockschaltbild des Transponders [DIPL01]	82
A.6. Vereinfachter Schaltplan des Transponders	83
A.7. Versorgungspannungsverlauf der simulierten Transponderschaltung	84
A.8. Fotografie des Transponders [DIPL01]	84
A.9. Blockschaltbild des verwendeten CPLDs [CPLD01]	85
A.10. Darstellung der Pinbelegung der eingesetzten Bustreiber	86
A.11. Pinbelegung des Spannungskomparators [NS01]	87
A.12. Pinbelegung des 4060 Zähler-ICs [PHI01]	87

A.13. Beispielschaltung zum 4060 Zähler-IC [PHI01]	88
A.14. Pinbelegung und interner Aufbau des HE721 Reed Relais [HAM01]	88
A.15. Pinbelegung des PI74LVC Pegelumsetzers [PER01]	89
A.16. Bauform des 3,3 V Spannungsreglers [NS02]	89
A.17. Beschaltung des LM1086 Spannungsreglers [NS02]	89
A.18. Bauform des 5V Spannungsreglers [ST01]	90
A.19. Beschaltung des L7805 Spannungsreglers [ST01]	90
A.20. Pinbelegung des MAX232 Treiber-ICs [MAX01]	90
A.21. Beschaltung des Treiber-ICs für die serielle Schnittstelle [MAX01]	91
B.1. Arbeitsoberfläche der Eclipse IDE	93
B.2. Arbeitsoberfläche der Layout-Software Eagle	94
B.3. Arbeitsoberfläche von Matlab	95
B.4. Arbeitsoberfläche der PSPICE Software	96
B.5. Arbeitsoberfläche von Modelsim	97
B.6. Arbeitsoberfläche des ISE Projekt Navigator	98
D.1. Schaltplan der Readerplatine Seite 1	126
D.2. Schaltplan der Readerplatine Seite 2	127
D.3. Schaltplan der Readerplatine Seite 3	128
D.4. Schaltplan der Readerplatine Seite 4	129

Tabellenverzeichnis

2.1. Frequenzbereiche von RFID-Systemen	6
2.2. Typische Reichweiten bei den Frequenzbereichen unter Verwendung eines passiven Transponders	6
2.3. Tabelle der relevanten Normen für RFID-Systeme im Frequenzbereich von $125kHz$	9
3.1. Tabellarische Übersicht der Zuordnung der Seitenflächen der Readerbox .	25
3.2. Berechnete Wicklungszahl der einzelnen Seitenflächen der Readerbox . .	27
3.3. Übersicht der Spulenwerte	28
3.4. Übersicht der gewählten Kapazitätswerte	34
3.5. Zuordnung der Ausgangssignale zur Differenzbeziehung	41
4.1. Ergebnisse der ersten Messreihe	55
4.2. Auflistung der berechneten Schwingkreisparameter	56
4.3. Zusammenfassung der jeweiligen Spulenradien	59
D.1. Portzuordnung der gewählten Kapazitäten	125
D.2. Portzuordnung der Eingangssignale zum Resonanzabgleich	130
D.3. Portzuordnung der Ausgangssignale zur Antennenselektion	130

Abkürzungsverzeichnis

ADC	Analog to Digital Converter
ASK	Amplitude Shift Keying
CG	Constant Generator
CPLD	Complex Programmable Logic Device
EAS	Electronic Article Surveillance
EPC	Electronic Product Code
GCC	GNU Compiler Collection
GNU	GNU is not UNIX
GPU	General Purpose Register
HF	High Frequency
IC	Intergrated Circuit
IDE	Intergrated Development Environment
ISM	Industrial, Scientific and Medical Band
ISO	International Organization for Standardization
LF	Low Frequency
OOK	On Off Keying
PC	Program Counter
PLCC	Plastic Leaded Carrier Chip
PSPICE	Personalcomputer Simulation Program with Intergrated Circuit Emphasis
RFID	Radio Frequency Identification
SICARID	Siemens Car Identification
SP	Stack Pointer
SR	Status Register

UHF Ultra High Frequency

VHDL Very High Speed Circuit Hardware Description Language

A. Verwendete Hardware

A.1. MSP430 Mikrocontroller

Als Mikrocontroller wird eine 16-Bit Central Processing Unit CPU der Firma Texas Instruments verwendet. Es handelt sich hierbei um eine Reduced Instruction Set Computing RISC CPU aus der MSP430 Serie. Controller aus dieser Reihe zeichnen sich besonders durch ihren niedrigen Strombedarf aus, daher sind sie bestens für den Einsatz in RFID-Systemen geeignet. Zur Programmierung des Controllers wird die Hochsprache C verwendet.

MSP430 Controller verfügen über 6 verschiedene Operations-Modi, einer davon ist der aktive Modus AM und die Anderen 5 sind verschiedene Stromsparmodi. Eine genaue Beschreibung kann unter [MSP01] entnommen werden.

Die CPU besitzt 16 Register, wobei dem Register R0 der Programm Counter PC zugeeignet ist. Der Stack Pointer SP befindet sich im Register R1, das Status Register SR im Register R2 und dem Register R3 ist der Constant Generator CG zugeordnet. Die übrigen Register R4 bis R15 sind General Purpose Register GPU und dienen zur Aufnahme von Daten und Adressen.

Im Rahmen dieser Diplomarbeit werden 2 verschiedene Modelle des MSP430 verwendet: Für den Transponder wird ein MSP430F1232 und für den Reader wird ein MSP430F169 Controller verwendet. Der MSP430H169 Mikrocontroller wird in der Header-Board-Variante der Firma Olimex verwendet. Die Programmierung der Controller erfolgt über die Joint Test Action Group JTAG Schnittstelle.

MSP430F1232

Zu den wesentlichen Merkmalen des MSP430F1232 gehören:

- 8K Byte Programm-Flash-Speicher
- 256 Byte Daten-Flash-Speicher
- 256 Byte Random-Access-Memory RAM

- 1x Universal Asynchronous Receiver Transmitter UART-Schnittstelle
- 1x 10 Bit Analog to Digital Converter ADC
- 1x 16 Bit Timer

Das Bild A.1. zeigt das Blockschaltbild des Controllers. In der Darstellung ist ersichtlich, dass der Controller zwei 8-Bit breite Ports und einen 6-Bit breiten Port anbietet. Die beiden Ports 1 und 2 sind bei diesem Controller interruptfähig.

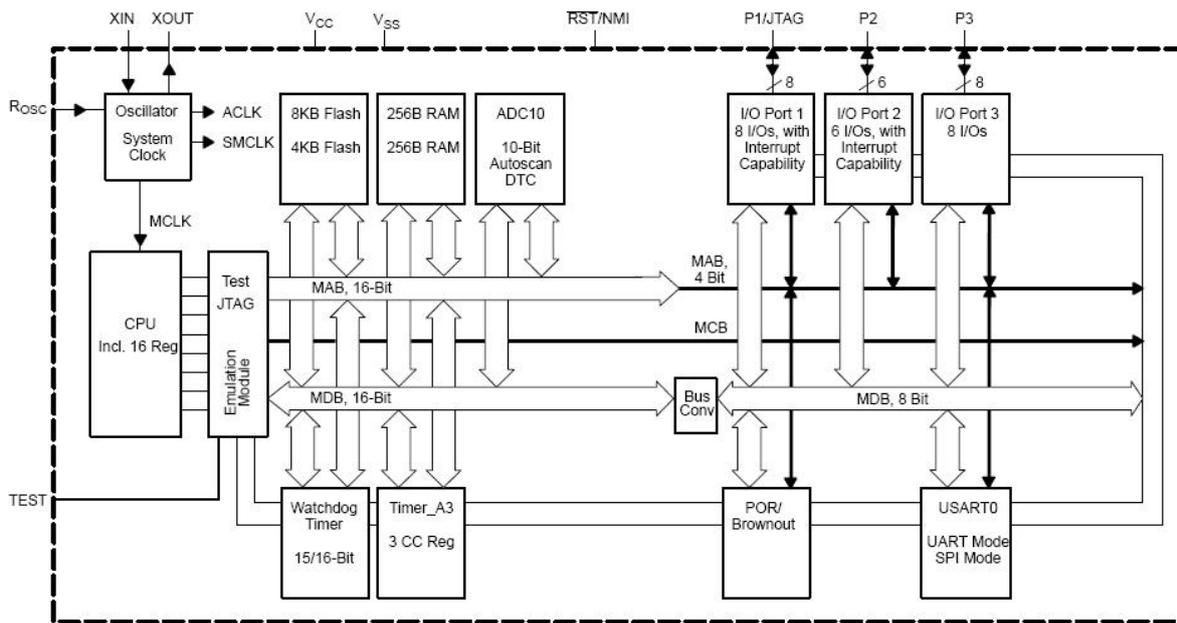


Bild A.1.: Blockschaltbild des MSPF1232-Controllers [MSP02]

MSP430F169

Einige der Merkmale des zweiten Controllers, dem MSP430F169 sind:

- 60K Byte Programm-Flash-Speicher
- 256 Byte Daten-Flash-Speicher
- 2K Byte Random-Access-Memory RAM
- 2x UART-Schnittstellen
- 1x 12 Bit ADC
- 2x 16 Bit Timer

Das Blockschaltbild ist unter Bild A.2. zu finden. Dieser Controller besitzt sechs 8-Bit breite Ports. Bei dieser Controller-Variante sind die beiden Ports 1 und 2 ebenfalls interruptfähig.

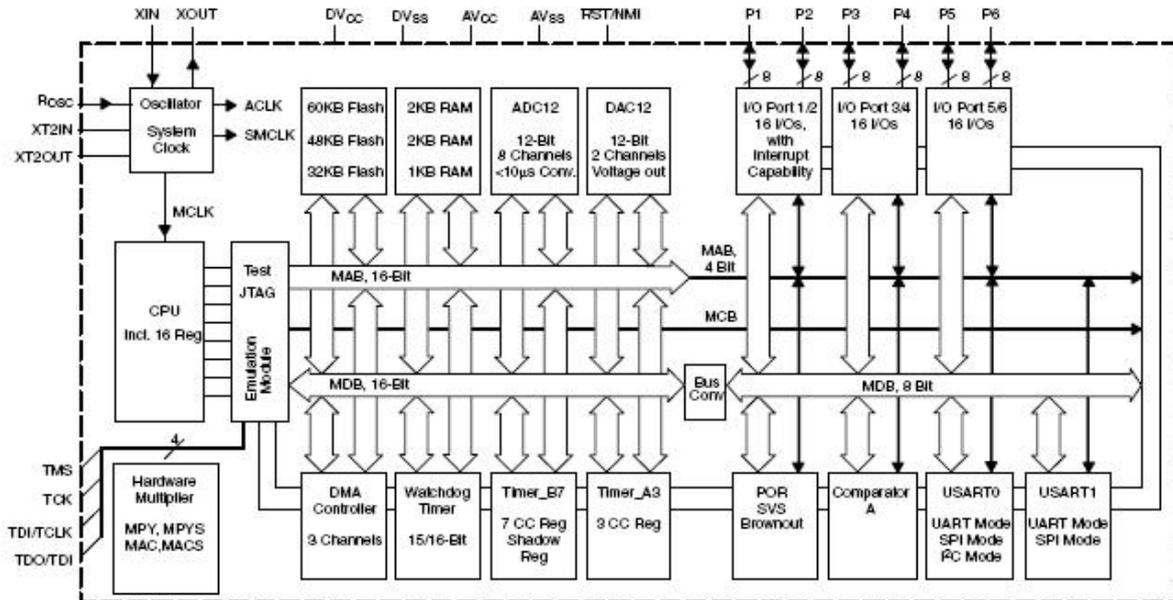


Bild A.2.: Blockschaltbild des MSPF169-Controllers [MSP02]

Eine Darstellung des MSP-H169-Header Boards zeigt das Bild A.3. Die Beschreibung des Boards und der Schaltplan sind unter [MSP02] aufrufbar.

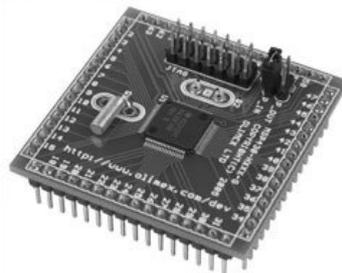


Bild A.3.: Darstellung des MSP-H169 Header-Boards [www.olimex.com]

JTAG Programmier Adapter

Zur Programmierung des Controllers wird ein JTAG Programmier Adapter benötigt. Dieser kann ebenfalls von der Firma Olimex bezogen werden und wird an die Parallelschnittstelle des Rechners angeschlossen.

[MSP03] zeigt die Beschreibung des Adapters.



Bild A.4.: JTAG-Programmier-Kabel zum Anschluss an eine Parallel-Schnittstelle
[www.olimex.com]

A.2. Transponder

Der verwendete Transponder wurde im Rahmen der Diplomarbeit von Tobias Krannich [DIPL01] erstellt. Für eine genaue Beschreibung des Transponders sei daher auf die genannte Diplomarbeit verwiesen. Es soll an dieser Stelle nur ein Kurzüberblick über die Funktion und den Aufbau der verwendeten Hardware gegeben werden.

Das Bild A.5. zeigt das Blockschaltbild des Transponders

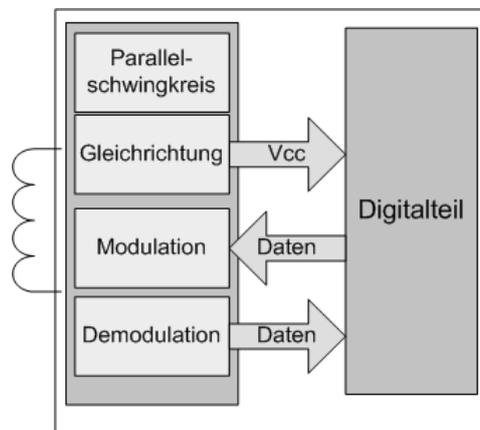


Bild A.5.: Blockschaltbild des Transponders [DIPL01]

Das Blockschaltbild lässt sich in zwei Hauptteile separieren. Auf der linken Seite befindet sich der Analogteil, während rechts der Digitalteil dargestellt ist.

Die Energieversorgung des Transponders erfolgt mittels der am Parallelschwingkreis anliegenden Spannung. Diese wird gleichgerichtet und dient unter anderem zur Versorgung des Digitalteils. Die aus der Demodulation gewonnenen Daten werden dem Digitalteil zugeführt und können dort weiterverarbeitet werden. Die zu übertragenden Daten werden vom Digitalteil generiert und können an den Modulationsblock übergeben werden.

Als Hardwarebasis für den Digitalteil des Transponders wird ein MSP430F1232 Mikrocontroller verwendet.

Das Bild A.6. stellt den Schaltplan des Transponders mit dem verwendeten Parallelschwingkreis und der Gleichrichterschaltung dar.

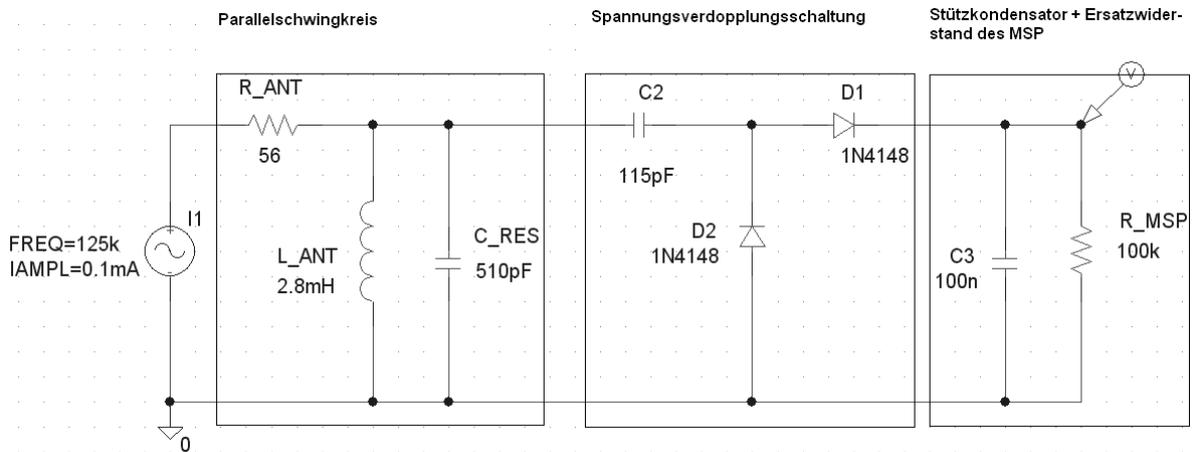


Bild A.6.: Vereinfachter Schaltplan des Transponders

In der Darstellung werden nur die Teile des Transponders aufgeführt, die zu unmittelbaren Energiegewinnung und Versorgung genutzt werden. Der verwendete Mikrocontroller ist hier als Ersatzwiderstand von $R_{MSP} = 100k$ aufgeführt.

Der schaltungstechnische Aufbau zur allgemeinen Spannungsbegrenzung und der Lastmodulation sind an dieser Stelle nicht weiter aufgeführt.

Der Wert für C_{RES} ist in dieser Schaltung um die parasitäre Kapazität der Gleichrichterschaltung reduziert dargestellt. Dieser Wert beträgt $68pF$, somit ergibt sich der notwendige Wert zur Abstimmung auf die $125kHz$ Resonanzfrequenz zu $C_{RES} = 579pF$.

Mittels dieser Hardwareanordnung ist es generell möglich, die notwendige Versorgungsspannung für den MSP zu erzeugen. Voraussetzung hierfür ist eine ausreichende Versorgung des Parallelschwingkreises. Das Bild A.7 zeigt das Ergebnis der Simulation mit PSPICE.

Das Bild A.8 zeigt eine Fotografie der bestückten Transponderplatine.

A.3. XC95108 CPLD

Die Bezeichnung CPLD steht für Complex Programmable Logic Device. Es handelt sich hierbei um einen programmierbaren Baustein. CPLDs können digitale Schaltungen enthalten. Diese lassen sich mittels einer Hardwarebeschreibungssprache, wie z.B. Very High Speed Hardware Description Language VHDL, beschreiben. Die Programmierung

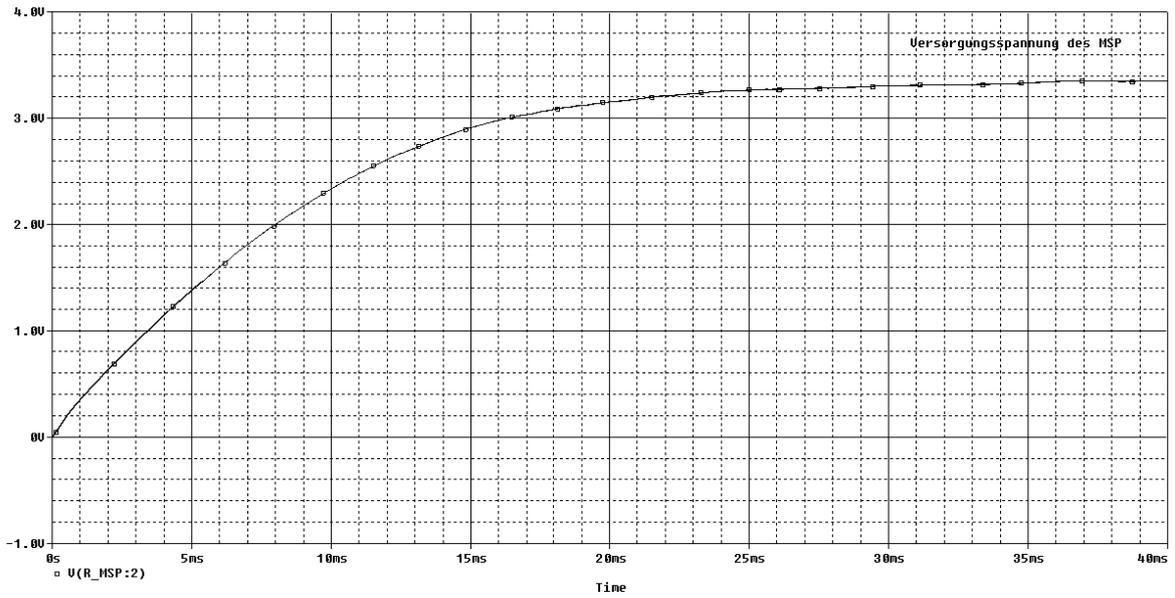
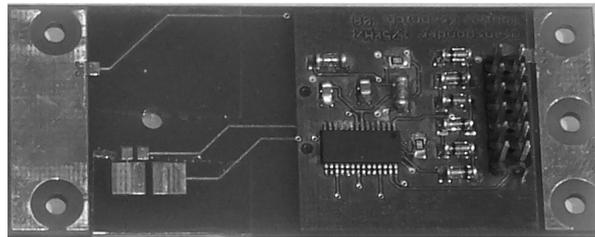
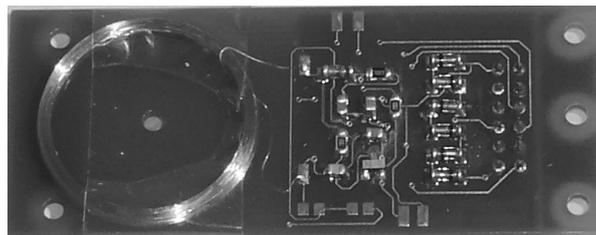


Bild A.7.: Versorgungsspannungsverlauf der simulierten Transponderschaltung



(a) Vorderseite der Transponderplatine



(b) Rückseite der Transponderplatine

Bild A.8.: Fotografie des Transponders [DIPL01]

des Bausteins bleibt auch bei nicht vorhandener Versorgungsspannung erhalten. Zum Einsatz kommt für diese Diplomarbeit ein Baustein der Firma Xilinx. Es wird ein CPLD aus der XC95108 Reihe verwendet.

Zu den Merkmalen des Bausteins gehören:

- f_{CNT} bis $125MHz$
- $5V$ tolerante Eingänge
- $3,3V$ Ausgangsspannung
- 84 definierbare Ein- und Ausgänge
- $24mA$ Ausgangsstrom
- 108 Makrozellen mit 2400 verwendbaren Gattern
- Plastic Leaded Chip Carrier PLCC Gehäuse

Die Programmierung des CPLDs erfolgt auch hier über die JTAG Schnittstelle.

Das Bild A.9 zeigt den internen Aufbau des Bausteins. Das Datenblatt kann unter [CPLD01] bezogen werden.

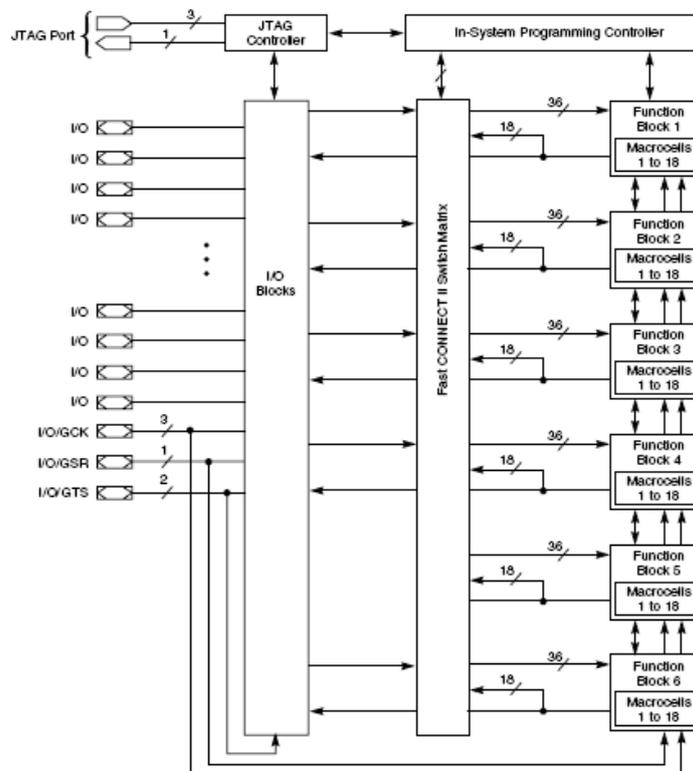


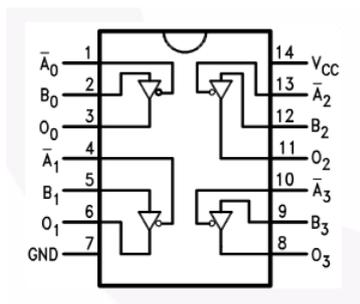
Bild A.9.: Blockschaltbild des verwendeten CPLDs [CPLD01]

A.4. 74ACT125 und 74ACT245 Bustreiber

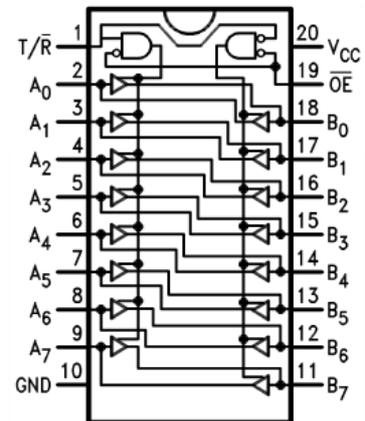
Zur Schwingkreisanregung werden 2 unterschiedliche Bustreiber eingesetzt. Beide Bustreiber werden von der Firma Fairchild Semiconductor vertrieben.

Die jeweils zuschaltbaren Kapazitäten werden über einen 74ACT125 Bustreiber in den Antennenschwingkreis eingebracht. Ein Treiberbaustein aus der 74ACT245 Familie übernimmt die Aufgabe der Schwingkreisanregung auf der gegenphasig angeregten Reihenschwingkreisseite.

Die jeweilige Pinbelegung und die interne Verbindungen der beiden verwendeten Bustreiber werden in Bild A.10. dargestellt.



(a) Pinbelegung des 74ACT125 Bustreibers [FAI01]



(b) Pinbelegung des 74ACT245 Bustreibers [FAI02]

Bild A.10.: Darstellung der Pinbelegung der eingesetzten Bustreiber

A.5. LM311 Spannungskomparator

Als Spannungskomparator kommt ein Baustein der Firma National Semiconductors zum Einsatz. Es wird ein LM311 Spannungskomparator in der DIL Bauform verwendet.

Das Bild A.11. gibt einen Einblick in die Pinbelegung des ICs.

Der Komparator wird mit einer Betriebsspannung von $\pm 15V$ versorgt. Diese Spannung wird der Readerplatine über ein externes Labornetzteil zugeführt.

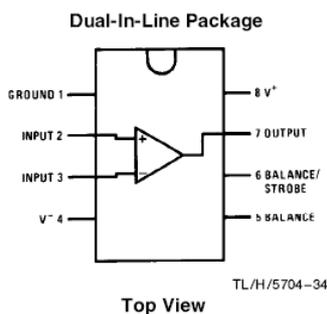


Bild A.11.: Pinbelegung des Spannungskomparators [NS01]

A.6. 4060 Zählerbaustein zur Takterzeugung

Zur Erzeugung der 125kHz Frequenz, welche zur Ansteuerung der Bustreiber genutzt wird, wird ein Zählerbaustein aus der 4060 Reihe verwendet.

Die Pinbelegung des ICs ist in Bild A.12. dargestellt.

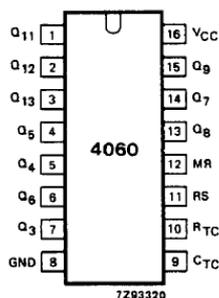


Bild A.12.: Pinbelegung des 4060 Zähler-ICs [PHI01]

Der IC wird zur Taktgewinnung von einem externen Quarz gespeist. Hierdurch ergibt sich die Notwendigkeit, den IC weiter zu beschalten. Das Bild A.13 zeigt eine beispielhafte Beschaltung, diese wurde aus [PHI01] entnommen.

A.7. HE721 Reed Relais

Die Antennenspulenselektion wird über Relais erwirkt. Für diese Implementierung des Readers wird dies mittels 5V Reed Relais der Firma Hamlin bewerkstelligt. Es handelt sich hierbei um Relais im DIL Gehäuse. Die Bezeichnung dieses Bauelements ist: HE721C0500.

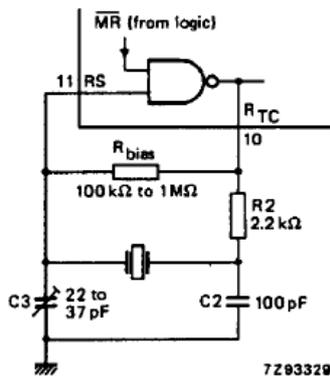


Bild A.13.: Beispielschaltung zum 4060 Zähler-IC [PHI01]

Die Pinbelegung des Reed Relais wird in Bild A.14 zusammen mit dem internen Aufbau dargestellt.

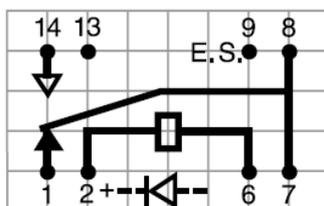


Bild A.14.: Pinbelegung und interner Aufbau des HE721 Reed Relais [HAM01]

A.8. PI74LVC Pegelumsetzer

Die jeweilige Antennenauswahl erfolgt mittels des eingesetzten Mikrocontrollers und den darüber geschalteten Relais. Der MSP430 Controller liefert 3,3V Ausgangspegel. Zum Schalten der Reed Relais sind aber laut Hersteller 5V Pegel vorgesehen. Aus diesem Grund wird zwischen diesen beiden Bauelementen ein Pegelumsetzer eingesetzt. Dieser Umsetzer übernimmt die Spannungsumsetzung von 3,3V auf 5V. Zum Einsatz kommt hier ein PI74LVC Pegelumsetzer der Firma Pericom.

Die Pinbelegung des Pegelumsetzers wird in Bild A.15 aufgeführt.

A.9. LM1086 3,3V Spannungsregler

Für die Versorgung des MSP430 Mikrocontrollers ist eine 3,3V Spannungsversorgung vorgesehen. Die Aufgabe der Versorgung übernimmt ein 3,3V Spannungsregler LM1086

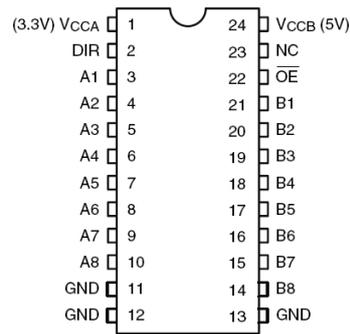


Bild A.15.: Pinbelegung des PI74LVC Pegelumsetzers [PER01]

der Firma National Semiconductors.

Die verwendete Bauform wird in Bild A.16. gezeigt und Bild A.17. zeigt die gewählte Beschaltung für den Spannungsregler.

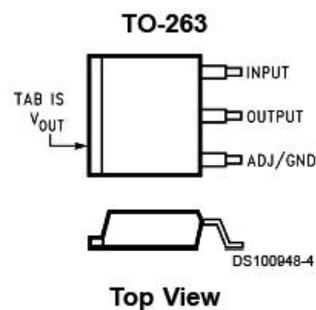


Bild A.16.: Bauform des 3,3 V Spannungsreglers [NS02]

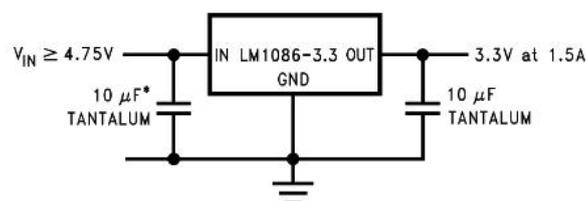


Bild A.17.: Beschaltung des LM1086 Spannungsreglers [NS02]

Die Herstellerbeschreibung kann unter [NS02] eingesehen werden.

A.10. L7805 5V Spannungsregler

Bei diesem Baustein der Firma ST Microelectronics handelt es sich um einen 5V Spannungsregler. Er wird, im Rahmen dieser Arbeit, in der TO-220F Bauform zur Anwendung

gebracht.

Das Bild A.18. zeigt die Pinbelegung des Spannungsreglers. Die notwendige Beschaltung ist in Bild A.19 dargestellt.

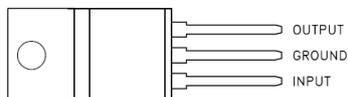


Bild A.18.: Bauform des 5V Spannungsreglers [ST01]

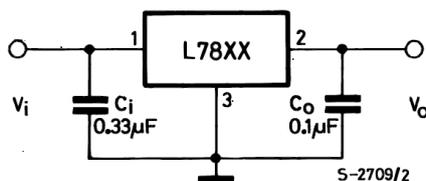


Bild A.19.: Beschaltung des L7805 Spannungsreglers [ST01]

A.11. MAX232 Treiber-IC für die serielle Schnittstelle

Die Kommunikation mit dem Reader erfolgt über die RS232 Schnittstelle. Zu diesem Zweck ist auf der Readerplatine ein Treiber-IC vorhanden. Hierbei handelt es sich um einen IC aus der MAX232 Reihe. Hersteller ist hier die Firma Maxim.

Die Pinbelegung des ICs ist in Bild A.20. zu finden. Um eine gesicherte Funktion der Kommunikation gewährleisten zu können, wird auch hier eine vorgegebene Schaltung vorausgesetzt. Diese Beschaltung ist in Bild A.21. dargestellt.

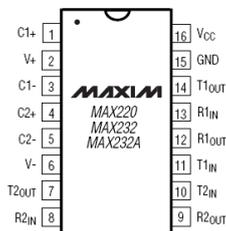


Bild A.20.: Pinbelegung des MAX232 Treiber-ICs [MAX01]

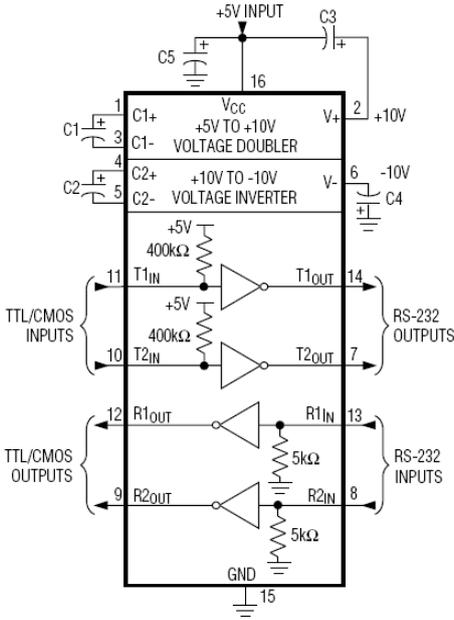


Bild A.21.: Beschaltung des Treiber-ICs für die serielle Schnittstelle [MAX01]

B. Verwendete Software

B.1. Mspgcc Toolchain

Zur Übersetzung des in C geschriebenen Programmcodes für den Controller, wird eine auf den MSP430 angepasste Version der GNU Compiler Collection GCC verwendet. Die Bezeichnung GNU steht hier für das Akronym "GNU is not Unix". Die für die Übersetzung benötigten Programme sind in der Mspgcc-Toolchain zusammengefasst. Diese Toolchain besteht aus den Binutils, dem Compiler und dem Debugger. Unter den Binutils sind der Assembler und der Linker für den Programmcode zusammengefasst. Der Übersetzungsprozess des Quellcodes in einen ausführbaren Maschinencode besteht aus den Schritten:

1. Durchlaufen des C-Codes mit dem Präprozessor
2. Kompilieren des C-Codes
3. Assemblieren des C-Codes
4. Linken des Assembler-Codes
5. Erstellen einer ausführbaren Programmdatei

Die so erstellte Programmdatei kann anschließend auf den Controller übertragen werden. Zu finden ist die verwendete Version der Toolchain 3.2.3 und die benötigte Proxy-Software mspg430-gdbproxy unter [GCC01].

Die Proxy-Software befindet sich als Vermittlungsschicht zwischen dem Debugger und der JTAG Schnittstelle des Controllers. Das Programm kommuniziert mittels einer TCP/IP Verbindung mit der Debugger-Software gdb und reicht die Befehle an den Controller weiter. Hierdurch kann das erstellte Programm während der Laufzeit auf dem Controller auf eine korrekte Funktion hin überprüft werden.

B.2. Eclipse Entwicklungsumgebung

Als Entwicklungsumgebung, zum Erstellen des Quellcodes, wird die Eclipse Integrated-Development-Environment IDE in der Version 3.3.2 verwendet. Ursprünglich wurde Eclipse für die Java Entwicklung entworfen, mittlerweile ist die IDE über Plugins auch auf andere Programmiersprachen erweiterbar. Im Rahmen dieser Diplomarbeit wird Eclipse mit der Mspgcc-Toolchain verwendet, um mittels der Open-Source Entwicklungsumgebung C-Code erstellen und übersetzen zu können. Eine detaillierte Installationsanleitung und eine Anleitung für das Erstellen eines Beispielprojekts sind auf der beigelegten CD vorhanden. Mein Dank an dieser Stelle an Stephan Plaschke für die Erstellung der Anleitung. Die Version 3.3.2 ist unter [ECL01] downloadbar.

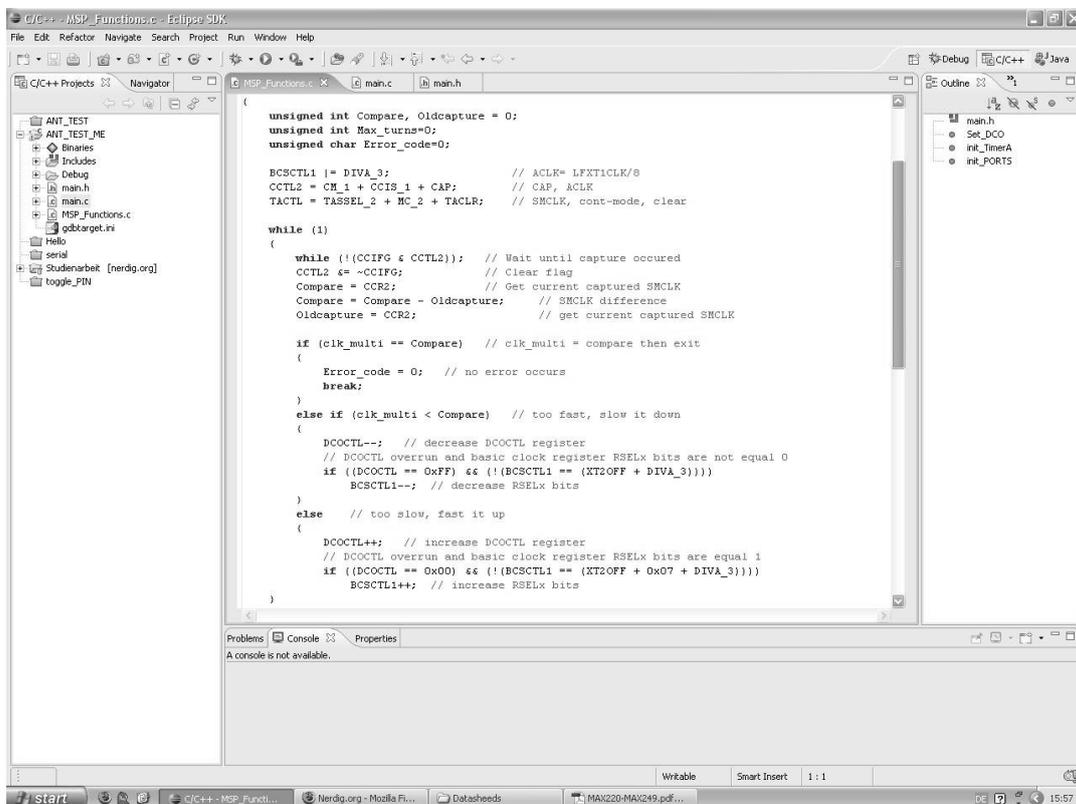


Bild B.1.: Arbeitsoberfläche der Eclipse IDE

B.3. Eagle Platinenlayout-Software

Für die Erstellung des Platinenlayouts, wird das Programm Eagle der Firma CadSoft verwendet. Über dieses Programm besteht die Möglichkeit, einen Schaltplan mit den gewünschten Bauteilen zu erstellen. Anschließend können über den enthaltenen Layout-Editor die gewählten Bauelemente auf der zu erstellenden Platine angeordnet und mit-

einander über Leiterbahnen verbunden werden.

Zur Fertigung der Platine ist es notwendig, das Layout an einen Leiterplatten-Hersteller zu geben, dieser erstellt daraufhin die Platine nach den Vorgaben.

Als Version kommt die Vollversion 4.11 zum Einsatz. Eine Freeware-Version mit einer Größenbeschränkung der zu erstellenden Platine kann unter [EAG01] heruntergeladen werden. Eine Einführung in die Software ist unter [EAG02] zu finden.

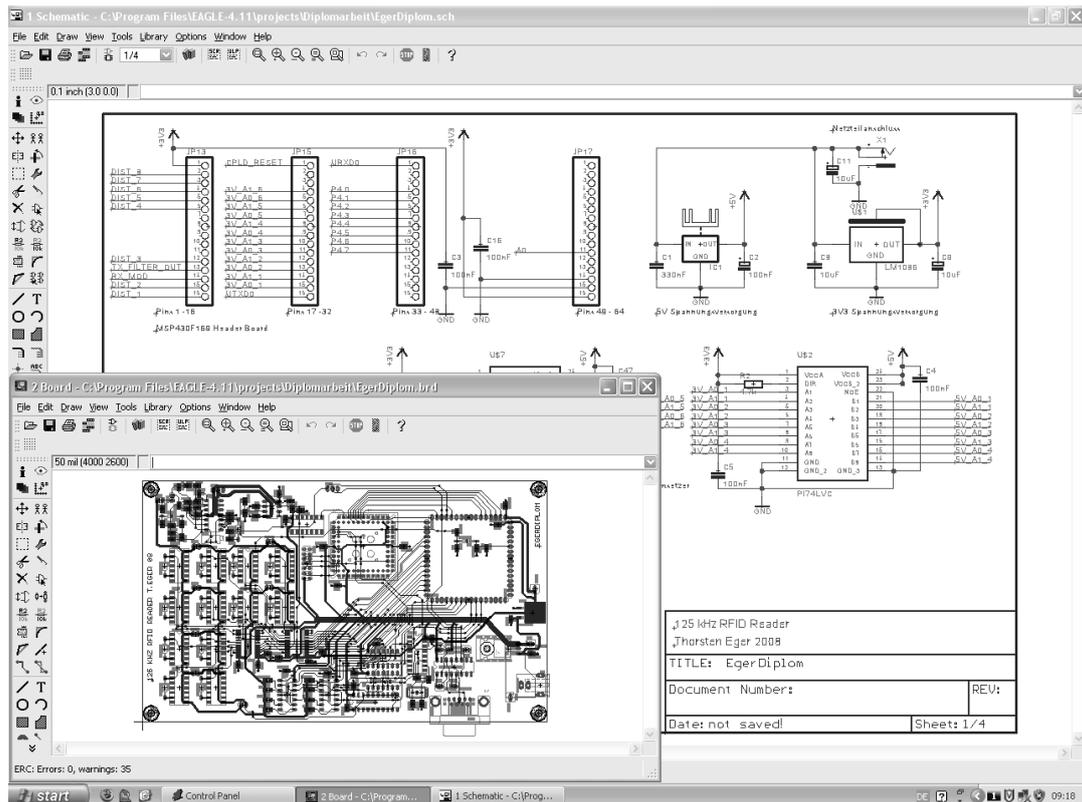


Bild B.2.: Arbeitsoberfläche der Layout-Software Eagle

B.4. Matlab

Das Programm Matlab wird von der Firma The MathWorks entwickelt und dient zur Darstellung und Berechnung von komplexen mathematischen Problemstellungen. Es verfügt über eine eigene Skriptsprache, mittels dieser Berechnungen definiert und gelöst werden können.

Eine Einführung in die Umgebung ist unter [MAT01] zu finden.

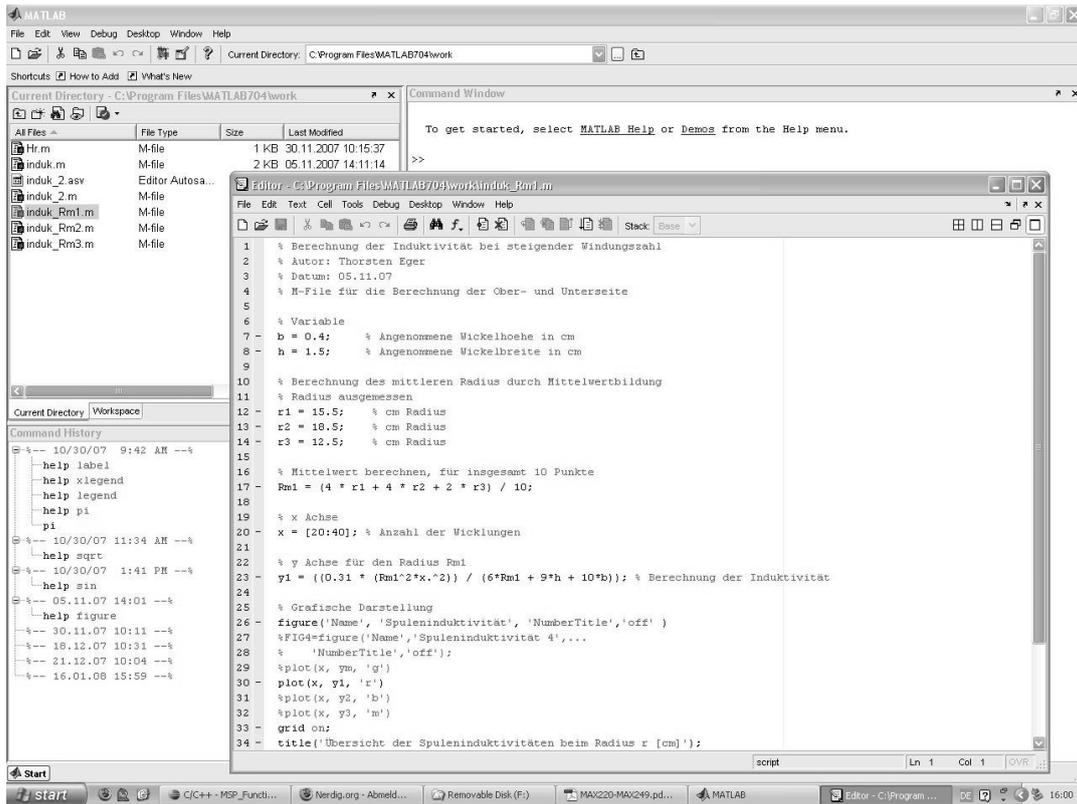


Bild B.3.: Arbeitsoberfläche von Matlab

B.5. PSPICE Simulationssoftware

Die Bezeichnung PSPICE steht für Personalcomputer Simulation Program with Integrated Circuit Emphasis. Es handelt sich bei dieser Software um ein Programm zur Simulation von elektronischen Schaltungen. Vertrieben wird es von der Firma Cadence Design Systems.

Es besteht die Möglichkeit zur Simulation von analogen oder digitalen Schaltungen. Die Studentenversion kann unter [PSPICE01] heruntergeladen werden. Eine Einführung in die Software kann beispielsweise unter [PSPICE02] eingesehen werden.

B.6. Modelsim Simulationssoftware

Die Software Modelsim dient als Simulationsumgebung für Hardwarebeschreibungssprachen, wie z.B. VHDL oder Verilog. Entwickelt und vertrieben wird sie von der Firma Xilinx. Die Software ist Bestandteil des ISEWebpacks und kann kostenfrei unter [VHDL01] heruntergeladen werden. Für eine Benutzung ist allerdings eine ebenfalls kostenlose Registrierung bei der Firma Xilinx notwendig. Ist dies erfolgt, so muss noch eine Lizenzdatei

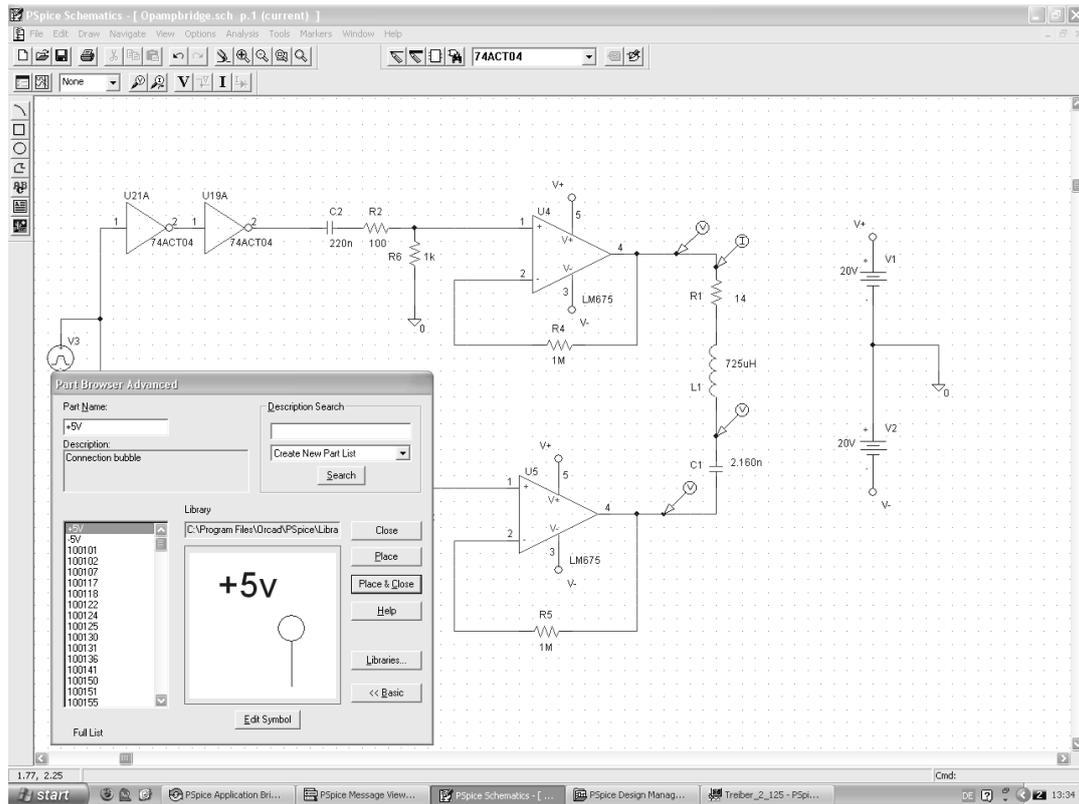


Bild B.4.: Arbeitsoberfläche der PSPICE Software

angefordert werden, diese wird an die angegebene Emailadresse versendet. Nach der Registrierung und der Lizenzierung lässt sich die Software benutzen.

Über den enthaltenen Editor können eigene Entwürfe beschrieben und anschließend kompiliert werden. Geschieht dies fehlerfrei, so ist es möglich den Entwurf mit dem enthaltenen Simulator zu prüfen. Über eine Textdatei können die jeweiligen Simulationsparameter gesetzt werden. Diese Angaben sind in einer Datei mit der Endung .do zu speichern. Die Simulationsergebnisse lassen sich anhand der sich ergebenden Signalverläufe überprüfen. Die verwendete Version ist Modelsim XE III/Starter 6.2c.

B.7. ISE Project Navigator

Dieses Programm ist ebenfalls im ISEWebpack enthalten und lässt sich unter [VHDL01] beziehen.

Der Project Navigator dient zur Übersetzung und zur Implementierung des erstellten VHDL Codes. Die somit erhaltenen Programmdateien lassen sich mittels dieser Software auf den CPLD übertragen. Weiterhin ist es möglich, sich einen Schaltplan der generierten Hardware anzeigen zu lassen. Hiermit lässt sich prüfen, ob die gewünschten Bauelemente erzeugt wurden.

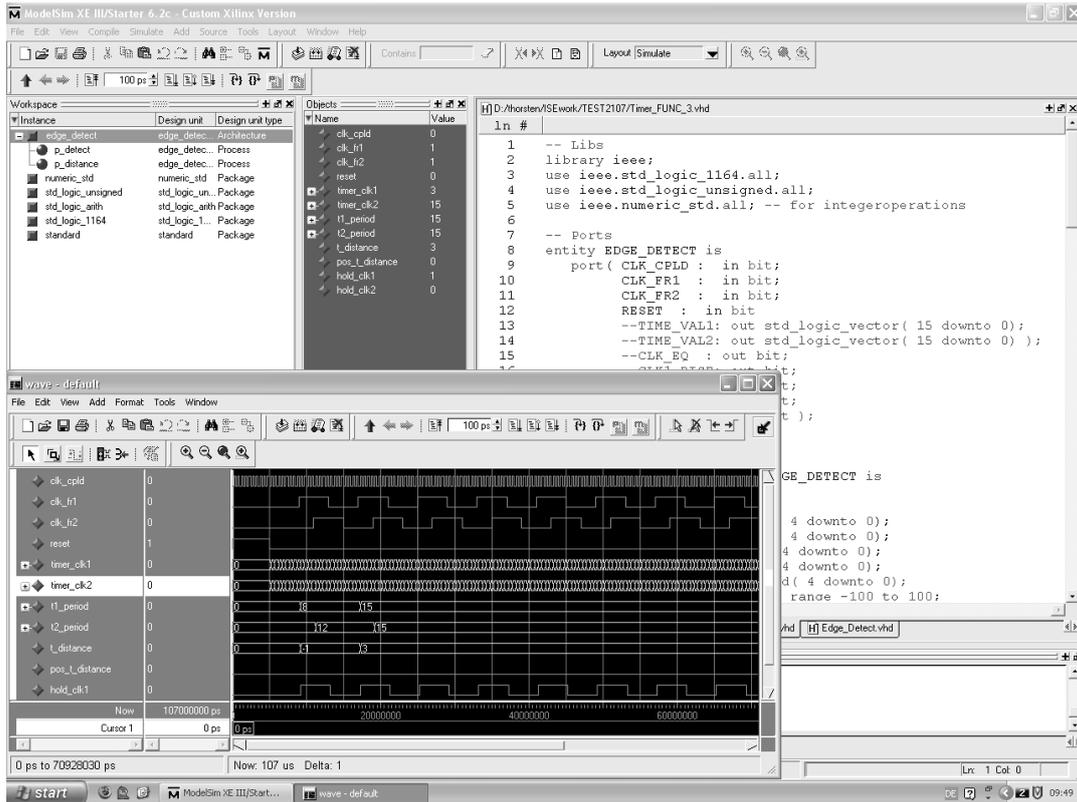


Bild B.5.: Arbeitsoberfläche von Modelsim

Eine Anleitung für den Umgang der beiden Programme Modelsim und des ISE Project Navigators ist unter [VHDL02] aufgeführt. Es wird die Version 9.1i im Rahmen dieser Arbeit verwendet.

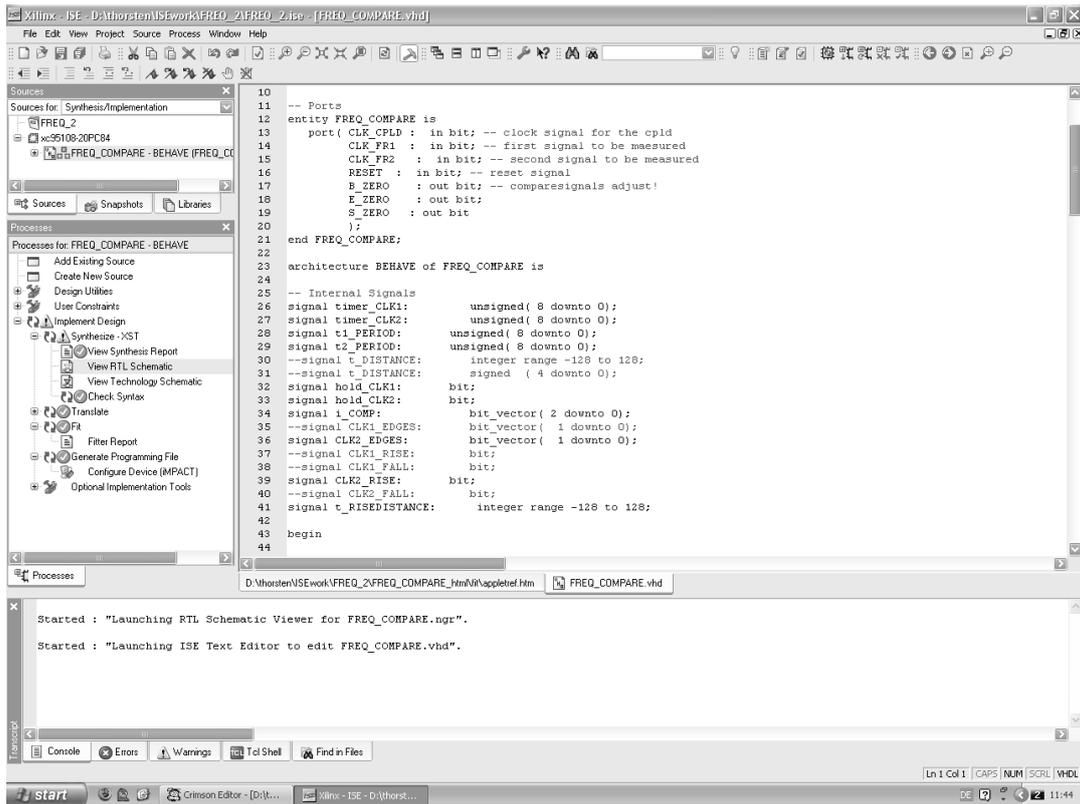


Bild B.6.: Arbeitsoberfläche des ISE Projekt Navigator

C. Quellcode

C.1. Matlab Code

Listing C.1: Berechnung des Feldverlaufs von $H(x)$ für 3 unterschiedliche Spulenradien

```
1 % Berechnung der magnetischen Feldstärke bei Erhöhung des Abstands zur
2 % Antenne
3 % Autor: Thorsten Eger
4 % Datum: 05.05.08
5
6 % Alte Werte löschen
7 clear all;
8 close all;
9 clc;
10
11 % Konstante
12 I = 50 * 10^-3;      % Strom in A
13 N_1 = 30;          % Anzahl der Antennenwicklungen
14
15 % Radius der Antennenspule in m
16 R_1 = 50 * 10^-3;
17 R_2 = 100 * 10^-3;
18 R_3 = 150 * 10^-3;
19
20 % Abstandsstrecke in m
21 x = (0:0.01:0.50);
22
23 % Berechnung der Feldstärke
24 H_1 = (I * N_1 * R_1^2) ./ ( 2 .* sqrt( (R_1^2 + x.^2).^3 ) );
25 H_2 = (I * N_1 * R_2^2) ./ ( 2 .* sqrt( (R_2^2 + x.^2).^3 ) );
26 H_3 = (I * N_1 * R_3^2) ./ ( 2 .* sqrt( (R_3^2 + x.^2).^3 ) );
27
28 % Grafische Darstellung
29 plot(x, H_1, x, H_2, x, H_3);
30 grid on;
31 title('Verlauf der magnetischen Feldstärke');
32 xlabel('Abstand zur Spule [m]');; ylabel('Magnetische Feldstärke H [A/m]');
33 legend('R_1 = 50 [mm]', 'R_2 = 100 [mm]', 'R_3 = 150 [mm]');
```

Listing C.2: Berechnung des Feldverlaufs von $H(r)$ bei einer Entfernung von 10cm zur Antenne

```
1 % Berechnung der magnetischen Feldstärke bei Erhöhung des Abstands zur
2 % Antenne
3 % Autor: Thorsten Eger
4 % Datum: 05.05.08
5
6 % Alte Werte löschen
7 clear all;
8 close all;
```

```

9  clc;
10
11 % Konstante
12 I = 50 * 10-3;      % Strom in A
13 N_1 = 30;          % Anzahl der Antennenwicklungen
14
15 % Entfernung x in m
16 X_1 = 0.1;
17
18 %Antennenradius in m
19 R_1 = (0:0.01:0.5);
20
21 % Berechnung der Feldstärke
22 H_1 = (I * N_1 * R_1.^2)./(2 .* sqrt( (R_1.^2 + X_1^2).^3 ));
23
24 % Grafische Darstellung
25 plot(R_1, H_1);
26 %plot(x, H_1,x,H_2,x, H_3);
27 grid on;
28 title('Verlauf der magnetischen Feldstärke bei x = 10 cm');
29 xlabel('Antennenradius r [m]'); ylabel('Magnetische Feldstärke H [A/m]');
30 legend('H(r)')

```

Listing C.3: Verlauf des Kopplungsfaktors k für 3 unterschiedliche Spulenradien

```

1  % Berechnung des Kopplungsfaktors bei Erhöhung des Abstands zur
2  % Antenne
3  % Autor: Thorsten Eger
4  % Datum: 28.08.08
5
6  % Alte Werte löschen
7  clear all;
8  close all;
9  clc;
10
11 % Radien der Spulen
12 r_trans = 12 * 10-3;      % Transponderradius in m
13 r1_read = 12 * 10-3;      % 1. Readerradius in m
14 r2_read = 24 * 10-3;      % 2. Readerradius in m
15 r3_read = 36 * 10-3;      % 3. Readerradius in m
16
17 % Abstandsstrecke in m
18 x = (0:0.0025:0.10);
19
20 % Berechnung des Kopplungsfaktors
21 k_1 = (r_trans^2 * r1_read^2) ./ ( sqrt( r_trans * r1_read) * (sqrt( (x.^2 + r1_read^2)
    .^3 )) );
22 k_2 = (r_trans^2 * r2_read^2) ./ ( sqrt( r_trans * r2_read) * (sqrt( (x.^2 + r2_read^2)
    .^3 )) );
23 k_3 = (r_trans^2 * r3_read^2) ./ ( sqrt( r_trans * r3_read) * (sqrt( (x.^2 + r3_read^2)
    .^3 )) );
24
25 % Grafische Darstellung
26 plot(x, k_1, x, k_2, x, k_3);
27 grid on;
28 title('Kopplungsfaktor k in Abhängigkeit zur Entfernung (r_T = 12 [mm])');
29 xlabel('Abstand x [m]'); ylabel('Kopplungsfaktor k');
30 legend('R_1 = 12 [mm]', 'R_2 = 24 [mm]', 'R_3 = 36 [mm]')

```

Listing C.4: Berechnung des Spuleninduktivitätssteigerungsverlaufs der einzelnen Seitenflächen des Readers

```

1 % Berechnung der Induktivität bei steigender Windungszahl
2 % Autor: Thorsten Eger
3 % Datum: 10.07.08
4 % Seitenflächen 1 und 4
5 % Abmaße der Seitenflächen: 300*205 mm
6 % Seitenflächen 2 und 5
7 % Abmaße der Seitenflächen: 400*205 mm
8 % Seitenflächen 3 und 6
9 % Abmaße der Seitenflächen: 400*300 mm
10
11 % Alte Werte löschen
12 clear all;
13 close all;
14 clc;
15
16 % Konstante
17 b = 0.4; % Angenommene Wickelhoehe in cm
18 h = 1.5; % Angenommene Wickelbreite in cm
19
20 % x Achse
21 x = [20:40]; % Anzahl der Wicklungen
22
23 % Seitenflächen 1 und 4
24 % Berechnung des mittleren Radius durch Mittelwertbildung
25 % Radius ausgemessen
26 S14_r1 = 13.5; % cm Radius
27 S14_r2 = 11.5; % cm Radius
28 % Mittelwert berechnen, für insgesamt 8 Punkte
29 S14_Rm = (4 * S14_r1 + 4 * S14_r2) / 8;
30 % y Achse für den Radius Rml
31 % Berechnung der Induktivität
32 y1 = ((0.31 * (S14_Rm^2*x.^2)) / (6*S14_Rm + 9*h + 10*b));
33 % Wicklungszahl für 725 uH
34 N1 = sqrt( (725*(6*S14_Rm + 9*h + 10*b)) / (0.31 * S14_Rm^2) )
35
36 % Seitenflächen 2 und 5
37 % Berechnung des mittleren Radius durch Mittelwertbildung
38 % Radius ausgemessen
39 S25_r1 = 18.5; % cm Radius
40 S25_r2 = 12.5; % cm Radius
41 S25_r3 = 9.0; % cm Radius
42 % Mittelwert berechnen, für insgesamt 10 Punkte
43 S25_Rm = (4 * S25_r1 + 4 * S25_r2 + 2 * S25_r3) / 10;
44 % y Achse für den Radius S25_Rm
45 % Berechnung der Induktivität
46 y2 = ((0.31 * (S25_Rm^2*x.^2)) / (6*S25_Rm + 9*h + 10*b));
47 % Wicklungszahl für 725 uH
48 N2 = sqrt( (725*(6*S25_Rm + 9*h + 10*b)) / (0.31 * S25_Rm^2) )
49
50 % Seitenflächen 3 und 6
51 % Berechnung des mittleren Radius durch Mittelwertbildung
52 % Radius ausgemessen
53 S36_r1 = 15.5; % cm Radius
54 S36_r2 = 18.5; % cm Radius
55 S36_r3 = 12.5; % cm Radius
56 % Mittelwert berechnen, für insgesamt 10 Punkte
57 S36_Rm = (4 * S36_r1 + 4 * S36_r2 + 2 * S36_r3) / 10;
58 % y Achse für den Radius Rml

```

```

59 % Berechnung der Induktivität
60 y3 = ((0.31 * (S36_Rm^2*x.^2)) / (6*S36_Rm + 9*h + 10*b));
61 % Wicklungszahl für 725 uH
62 N3 = sqrt( (725*(6*S36_Rm + 9*h + 10*b)) / (0.31 * S36_Rm^2) )
63
64 % Grafische Darstellung
65 figure('Name', 'Spuleninduktivität', 'NumberTitle','off' )
66 plot(x, y1, x, y2, x, y3)
67 grid on;
68 title('Spuleninduktivität bei steigender Wicklungszahl');
69 xlabel('Anzahl der Wicklungen'); ylabel('Induktivität [uH]');
70 legend('R_m_1 = 12.5 [cm]', 'R_m_2 = 14.2 [cm]', 'R_m_3 = 16,1 [cm]')

```

Listing C.5: Berechnung des Feldverlaufs von $H(x)$ unter Berücksichtigung der realen Antennenparameter

```

1 % Berechnung der magnetischen Feldstärke bei Erhöhung des Abstands zur
2 % Antenne unter Zuhilfenahme der realen Werte
3 % Autor: Thorsten Eger
4 % Datum: 14.10.08
5
6 % Alte Werte löschen
7 clear all;
8 close all;
9 clc;
10
11 % Stromwerte
12 I_1 = 66 * 10^-3; % Strom in A
13 I_2 = 58 * 10^-3; % Strom in A
14 I_3 = 67 * 10^-3; % Strom in A
15
16 % Anzahl der Antennenwicklungen
17 N_1 = 36; % Seitenflächen 1 + 4
18 N_2 = 33; % Seitenflächen 2 + 5
19 N_3 = 30; % Seitenflächen 3 + 6
20
21 % Radius der Antennenspule in m
22 R_1 = 125 * 10^-3; % Seitenflächen 1 + 4
23 R_2 = 142 * 10^-3; % Seitenflächen 2 + 5
24 R_3 = 161 * 10^-3; % Seitenflächen 3 + 6
25
26 % Abstandsstrecke in m
27 x = (0:0.01:0.50);
28
29 % Berechnung der Feldstärke
30 H_1 =(I_1 * N_1 * R_1^2)./( 2 .* sqrt( (R_1^2 + x.^2).^3) );
31 H_2 =(I_2 * N_2 * R_2^2)./( 2 .* sqrt( (R_2^2 + x.^2).^3) );
32 H_3 =(I_3 * N_3 * R_3^2)./( 2 .* sqrt( (R_3^2 + x.^2).^3) );
33
34 % Grafische Darstellung
35 plot(x, H_1,x,H_2,x, H_3);
36 grid on;
37 title('Verlauf der magnetischen Feldstärke');
38 xlabel('Abstand zur Spule [m]'); ylabel('Magnetische Feldstärke H [A/m]');
39 legend('Seitenflächen 1 + 4', 'Seitenflächen 2 + 5', 'Seitenflächen 3 + 6')

```

Listing C.6: Berechnung des Feldverlaufs von $k(x)$ unter Berücksichtigung der realen Antennenradien

```

1 % Berechnung des Kopplungsfaktors bei Erhöhung des Abstands zur

```

```

2 % Antenne unter Berücksichtigung der realen Werte
3 % Autor: Thorsten Eger
4 % Datum: 16.10.08
5
6 % Alte Werte löschen
7 clear all;
8 close all;
9 clc;
10
11 % Radien der Spulen
12 r_trans = 12 * 10^-3; % Transponderradius in m
13 r1_read = 125 * 10^-3; % 1. Readerradius in m
14 r2_read = 142 * 10^-3; % 2. Readerradius in m
15 r3_read = 161 * 10^-3; % 3. Readerradius in m
16
17 % Abstandsstrecke in m
18 x = (0:0.01:0.50);
19
20 % Berechnung der Kopplungsfaktoren
21 k_1 = (r_trans^2 * r1_read^2) ./ ( sqrt( r_trans * r1_read) * (sqrt( (x.^2 + r1_read^2).^3 )) );
22 k_2 = (r_trans^2 * r2_read^2) ./ ( sqrt( r_trans * r2_read) * (sqrt( (x.^2 + r2_read^2).^3 )) );
23 k_3 = (r_trans^2 * r3_read^2) ./ ( sqrt( r_trans * r3_read) * (sqrt( (x.^2 + r3_read^2).^3 )) );
24
25 % Grafische Darstellung
26 plot(x, k_1, x, k_2, x, k_3);
27 grid on;
28 title('Kopplungsfaktor k in Abhängigkeit zur Entfernung (r_T = 12 [mm])');
29 xlabel('Abstand x [m]'); ylabel('Kopplungsfaktor k');
30 legend('R_1 = 125 [mm]', 'R_2 = 142 [mm]', 'R_3 = 161 [mm]');
31 legend('Seitenflächen 1 + 4', 'Seitenflächen 2 + 5', 'Seitenflächen 3 + 6')

```

Listing C.7: Berechnung des Feldverlaufs von $H_T(x)$

```

1 % Berechnung des Verlaufs der elektrischen Feldstärke am Transponder
2 % Autor: Thorsten Eger
3 % Datum: 17.10.08
4
5 % Alte Werte löschen
6 clear all;
7 close all;
8 clc;
9
10 % Stromwerte
11 I_1 = 66 * 10^-3; % Strom in A
12 I_2 = 58 * 10^-3; % Strom in A
13 I_3 = 67 * 10^-3; % Strom in A
14
15 % Anzahl der Antennenwicklungen
16 N_1 = 36; % Seitenflächen 1 + 4
17 N_2 = 33; % Seitenflächen 2 + 5
18 N_3 = 30; % Seitenflächen 3 + 6
19
20 % Radien der Antennenspulen in m
21 R_1 = 125 * 10^-3; % Seitenflächen 1 + 4
22 R_2 = 142 * 10^-3; % Seitenflächen 2 + 5
23 R_3 = 161 * 10^-3; % Seitenflächen 3 + 6
24 r_trans = 12 * 10^-3; % Transponderradius
25
26 % Abstandsstrecke in m
27 x = (0:0.01:0.50);

```

```

28
29 % Berechnung der Feldstärke
30 H_1 =(I_1 * N_1 * R_1^2)./( 2 .* sqrt( (R_1^2 + x.^2).^3 ) );
31 H_2 =(I_2 * N_2 * R_2^2)./( 2 .* sqrt( (R_2^2 + x.^2).^3 ) );
32 H_3 =(I_3 * N_3 * R_3^2)./( 2 .* sqrt( (R_3^2 + x.^2).^3 ) );
33
34 % Radien der Spulen
35 r_trans = 12 * 10^-3; % Transponderradius in m
36 r1_read = 125 * 10^-3; % 1. Readerradius in m
37 r2_read = 142 * 10^-3; % 2. Readerradius in m
38 r3_read = 161 * 10^-3; % 3. Readerradius in m
39
40 % Berechnung der Kopplungsfaktoren
41 k_1 = (r_trans^2 * R_1^2) ./ ( sqrt( r_trans * R_1 ) * (sqrt( (R_1^2).^3 )) );
42 k_2 = (r_trans^2 * R_2^2) ./ ( sqrt( r_trans * R_2 ) * (sqrt( (R_2^2).^3 )) );
43 k_3 = (r_trans^2 * R_3^2) ./ ( sqrt( r_trans * R_3 ) * (sqrt( (R_3^2).^3 )) );
44
45 % Berechnung der effektiven Feldstärke am Transponder
46 H_Trans_1 = H_1 .* k_1;
47 H_Trans_2 = H_2 .* k_2;
48 H_Trans_3 = H_3 .* k_3;
49
50 % Grafische Darstellung
51 plot(x, H_Trans_1,x,H_Trans_2,x, H_Trans_3);
52 grid on;
53 title('Verlauf der verfügbaren magnetischen Feldstärke am Transponder');
54 xlabel('Abstand zur Spule [m]'); ylabel('Magnetische Feldstärke H [A/m]');
55 legend('Seitenflächen 1 + 4', 'Seitenflächen 2 + 5', 'Seitenflächen 3 + 6')

```

C.2. VHDL Code

Listing C.8: VHDL Code zum Resonanzabgleich

```

1 — VHDL Code for the resonancematching
2 — Author: Thorsten Eger
3 — Date: 29.09.08
4
5 — Libs
6 library ieee;
7 use ieee.std_logic_1164.all; — standarlibrary
8 use ieee.numeric_std.all; — for integeroperations
9
10 — Ports
11 — Define the inputs and the outputs
12 entity FREQ_COMPARE is
13     port(
14         CLK_CPLD : in bit; — clock signal for the cpld
15         CLK_FR1  : in bit; — first signal to be maesured
16         CLK_FR2  : in bit; — second signal to be measured
17         RESET    : in bit; — reset signal
18         DIST_8   : out bit; — outputsignals
19         DIST_7   : out bit; — to show the actual distance
20         DIST_6   : out bit;
21         DIST_5   : out bit;
22         DIST_4   : out bit;
23         DIST_3   : out bit;
24         DIST_2   : out bit;
25         DIST_1   : out bit
26     );

```



```
86         else
87             timer_CLK2 <= (others => '0'); -- clear timer
88         end if;
89     end if;
90
91     -- Edge Detection
92     -- This is only performed for the second signal
93     -- Determine the signal values
94     CLK2_EDGES(0) <= CLK_FR2;
95     CLK2_EDGES(1) <= CLK2_EDGES(0);
96
97     -- Set this bit to 1 if the previous state is 0 and the actual
98     -- state is 1
99     CLK2_RISE <= not CLK2_EDGES(1) and CLK2_EDGES(0); -- 01
100
101     -- calculate the distancevalue
102     if CLK2_RISE = '1' then
103         t_RISEDISTANCE <= to_integer(timer_clk1) - to_integer(timer_clk2);
104     end if;
105
106     -- estimate the distancevalue
107     -- distance: > 3.0 us
108     if ( t_RISEDISTANCE >= 120 ) then
109         i_COMP <= "10000000";
110     end if;
111     -- distance: 2.525 - 3.0 us
112     if (( t_RISEDISTANCE > 100) and (t_RISEDISTANCE < 120)) then
113         i_COMP <= "01000000";
114     end if;
115     -- distance: 2.275 - 2.5 us
116     if (( t_RISEDISTANCE > 90) and (t_RISEDISTANCE <= 100)) then
117         i_COMP <= "00100000";
118     end if;
119     -- distance: 2.025 - 2.25 us
120     if (( t_RISEDISTANCE > 80) and (t_RISEDISTANCE <= 90)) then
121         i_COMP <= "00010000";
122     end if;
123
124     -- distance: 1.775 - 2.0 us
125     if (( t_RISEDISTANCE > 70) and (t_RISEDISTANCE <= 80)) then
126         i_COMP <= "00001000";
127     end if;
128
129     -- distance: 1.525 - 1.75 us
130     if (( t_RISEDISTANCE > 60) and (t_RISEDISTANCE <= 70)) then
131         i_COMP <= "00000100";
132     end if;
133
134     -- distance: 1.025 - 1.5 us
135     if(( t_RISEDISTANCE > 40) and (t_RISEDISTANCE <= 60)) then
136         i_COMP <= "00000010";
137     end if;
138
139     -- distance: 0 - 1.0 us
140     if (( t_RISEDISTANCE > 0) and ( t_RISEDISTANCE <= 40)) then
141         i_COMP <= "00000001";
142     end if;
143
144     if (t_RISEDISTANCE <= 0) then
145         i_COMP <= (others => '1');
```

```

146         end if;
147
148         end if; -- of the clock detection
149
150     end process P_DETECT;
151
152     -- signal assingment
153     DIST_8 <= i_COMP(7);
154     DIST_7 <= i_COMP(6);
155     DIST_6 <= i_COMP(5);
156     DIST_5 <= i_COMP(4);
157     DIST_4 <= i_COMP(3);
158     DIST_3 <= i_COMP(2);
159     DIST_2 <= i_COMP(1);
160     DIST_1 <= i_COMP(0);
161
162 end BEHAVE;

```

Listing C.9: Do Datei zur Simulation des VHDL Codes

```

1 -- Datei: freq_COMP.do
2 -- Autor: Thorsten Eger
3 -- Datum: 29.09.08
4
5 -- initials
6 vsim work.FREQ_COMPARE
7 restart
8 view wave
9 radix decimal
10 add wave -height 25 -radix default sim:/FREQ_COMPARE/*
11
12 -- define the clocksignal
13 force CLK_CPLD      0, 1 12.5ns -r 25ns
14
15 -- perform a reset
16 force RESET 1
17 run 5us
18 force RESET 0
19
20 -- describe the clocksignals
21 force CLK_FR1      0, 1 4us -r 8us
22 -- define distance
23 run 2 us
24 force CLK_FR2      0, 1 4us -r 8us
25
26 -- define runtime
27 run 100us

```

Listing C.10: UCF Datei zur Definition der Ein- und Ausgänge des CPLDS

```

1 # Datei: freq_COMP.ucf
2 # Autor: Thorsten Eger
3 # Datum: 29.09.08
4
5 # Outputsignals
6 NET DIST_1 LOC = P84;
7 NET DIST_2 LOC = P1;
8 NET DIST_3 LOC = P2;
9 NET DIST_4 LOC = P3;
10 NET DIST_5 LOC = P4;

```

```

11 NET DIST_6 LOC = P5;
12 NET DIST_7 LOC = P6;
13 NET DIST_8 LOC = P7;
14
15 # Inputsignals
16 NET CLK_CPLD LOC = P9;
17
18 NET CLK_FR1 LOC = P17;
19 NET CLK_FR2 LOC = P18;
20
21 NET RESET LOC = P74;

```

Listing C.11: Synthesereport des entworfenen VHDL Modells

```

1 Release 9.1i - xst J.30
2 Copyright (c) 1995-2007 Xilinx, Inc. All rights reserved.
3 --> Parameter TMPDIR set to ./xst/projnav.tmp
4 CPU : 0.00 / 1.00 s | Elapsed : 0.00 / 1.00 s
5
6 --> Parameter xsthdpdir set to ./xst
7 CPU : 0.00 / 1.00 s | Elapsed : 0.00 / 1.00 s
8
9 --> Reading design: FREQ_COMPARE.prj
10
11 TABLE OF CONTENTS
12 1) Synthesis Options Summary
13 2) HDL Compilation
14 3) Design Hierarchy Analysis
15 4) HDL Analysis
16 5) HDL Synthesis
17 5.1) HDL Synthesis Report
18 6) Advanced HDL Synthesis
19 6.1) Advanced HDL Synthesis Report
20 7) Low Level Synthesis
21 8) Partition Report
22 9) Final Report
23
24 =====
25 *                      Synthesis Options Summary                      *
26 =====
27 --- Source Parameters
28 Input File Name          : "FREQ_COMPARE.prj"
29 Input Format              : mixed
30 Ignore Synthesis Constraint File : NO
31
32 --- Target Parameters
33 Output File Name         : "FREQ_COMPARE"
34 Output Format             : NGC
35 Target Device            : XC9500 CPLDs
36
37 --- Source Options
38 Top Module Name          : FREQ_COMPARE
39 Automatic FSM Extraction : YES
40 FSM Encoding Algorithm   : Auto
41 Safe Implementation      : No
42 Mux Extraction           : YES
43 Resource Sharing         : YES
44
45 --- Target Options
46 Add IO Buffers          : YES

```

```

47 MACRO Preserve           : YES
48 XOR Preserve            : YES
49 Equivalent register Removal : YES
50
51 — General Options
52 Optimization Goal       : Speed
53 Optimization Effort     : 1
54 Library Search Order    : FREQ_COMPARE.lso
55 Keep Hierarchy          : YES
56 RTL Output               : Yes
57 Hierarchy Separator     : /
58 Bus Delimiter           : <>
59 Case Specifier          : maintain
60 Verilog 2001            : YES
61
62 — Other Options
63 wysiwyg                  : NO
64
65 =====
66
67
68 =====
69 *                        HDL Compilation                        *
70 =====
71 Compiling vhdl file "D:/thorsten/ISEwork/freq_COMP_2/Freq_COMP.vhd" in Library work.
72 Entity <freq_compare> compiled.
73 Entity <freq_compare> (Architecture <behave>) compiled.
74
75 =====
76 *                        Design Hierarchy Analysis              *
77 =====
78 Analyzing hierarchy for entity <FREQ_COMPARE> in library <work> (architecture <behave>).
79
80
81 =====
82 *                        HDL Analysis                          *
83 =====
84 Analyzing Entity <FREQ_COMPARE> in library <work> (Architecture <behave>).
85 Entity <FREQ_COMPARE> analyzed. Unit <FREQ_COMPARE> generated.
86
87
88 =====
89 *                        HDL Synthesis                          *
90 =====
91
92 Performing bidirectional port resolution...
93
94 Synthesizing Unit <FREQ_COMPARE>.
95   Related source file is "D:/thorsten/ISEwork/freq_COMP_2/Freq_COMP.vhd".
96 WARNING:Xst - Property "use_dsp48" is not applicable for this technology.
97   Found 2-bit register for signal <CLK2_EDGES>.
98   Found 1-bit register for signal <CLK2_RISE>.
99   Found 1-bit register for signal <hold_CLK1>.
100  Found 1-bit register for signal <hold_CLK2>.
101  Found 8-bit register for signal <i_COMP>.
102  Found 11-bit comparator lessequal for signal <i_COMP$cmp_1e0000> created at line 144.
103  Found 11-bit comparator lessequal for signal <i_COMP$cmp_1e0001> created at line 140.
104  Found 11-bit comparator lessequal for signal <i_COMP$cmp_1e0002> created at line 135.
105  Found 11-bit comparator lessequal for signal <i_COMP$cmp_1e0003> created at line 130.
106  Found 11-bit comparator lessequal for signal <i_COMP$cmp_1e0004> created at line 125.

```

```

107     Found 11-bit comparator lessequal for signal <i_COMP$cmp_le0005> created at line 120.
108     Found 11-bit comparator lessequal for signal <i_COMP$cmp_le0006> created at line 116.
109     Found 11-bit comparator less for signal <i_COMP$cmp_lt0000> created at line 112.
110     Found 11-bit register for signal <t_RISEDISTANCE>.
111     Found 11-bit subtractor for signal <t_RISEDISTANCE$sub0000> created at line 103.
112     Found 10-bit up counter for signal <timer_CLK1>.
113     Found 10-bit up counter for signal <timer_CLK2>.
114     Summary:
115     inferred    2 Counter(s).
116     inferred    5 D-type flip-flop(s).
117     inferred    1 Adder/Subtractor(s).
118     inferred    8 Comparator(s).
119 Unit <FREQ_COMPARE> synthesized.
120
121 WARNING:Xst - Property "use_dsp48" is not applicable for this technology.
122
123 =====
124 HDL Synthesis Report
125
126 Macro Statistics
127 # Adders/Subtractors                : 1
128   11-bit subtractor                  : 1
129 # Counters                          : 2
130   10-bit up counter                  : 2
131 # Registers                         : 7
132   1-bit register                     : 5
133   11-bit register                    : 1
134   8-bit register                     : 1
135 # Comparators                       : 8
136   11-bit comparator less             : 1
137   11-bit comparator lessequal       : 7
138
139 =====
140
141 =====
142 *                               Advanced HDL Synthesis                               *
143 =====
144
145
146 =====
147 Advanced HDL Synthesis Report
148
149 Macro Statistics
150 # Adders/Subtractors                : 1
151   11-bit subtractor                  : 1
152 # Counters                          : 2
153   10-bit up counter                  : 2
154 # Registers                         : 24
155   Flip-Flops                        : 24
156
157 =====
158
159 =====
160 *                               Low Level Synthesis                               *
161 =====
162
163 Optimizing unit <FREQ_COMPARE> ...
164   implementation constraint: INIT=r   : hold_CLK2
165   implementation constraint: INIT=r   : CLK2_RISE
166   implementation constraint: INIT=r   : CLK2_EDGES_0

```

```

167 implementation constraint: INIT=r : CLK2_EDGES_1
168 implementation constraint: INIT=r : hold_CLK1
169 implementation constraint: INIT=r : i_COMP_0
170 implementation constraint: INIT=r : i_COMP_1
171 implementation constraint: INIT=r : i_COMP_2
172 implementation constraint: INIT=r : i_COMP_3
173 implementation constraint: INIT=r : i_COMP_4
174 implementation constraint: INIT=r : i_COMP_5
175 implementation constraint: INIT=r : i_COMP_6
176 implementation constraint: INIT=r : i_COMP_7
177 implementation constraint: INIT=r : t_RISEDISTANCE_0
178 implementation constraint: INIT=r : t_RISEDISTANCE_1
179 implementation constraint: INIT=r : t_RISEDISTANCE_2
180 implementation constraint: INIT=r : t_RISEDISTANCE_3
181 implementation constraint: INIT=r : t_RISEDISTANCE_4
182 implementation constraint: INIT=r : t_RISEDISTANCE_5
183 implementation constraint: INIT=r : t_RISEDISTANCE_6
184 implementation constraint: INIT=r : t_RISEDISTANCE_7
185 implementation constraint: INIT=r : t_RISEDISTANCE_8
186 implementation constraint: INIT=s : t_RISEDISTANCE_9
187 implementation constraint: INIT=s : t_RISEDISTANCE_10
188
189 =====
190 *                      Partition Report                      *
191 =====
192
193 Partition Implementation Status
194 -----
195
196     No Partitions were found in this design.
197
198 -----
199
200 =====
201 *                      Final Report                          *
202 =====
203 Final Results
204 RTL Top Level Output File Name      : FREQ_COMPARE.ngr
205 Top Level Output File Name         : FREQ_COMPARE
206 Output Format                       : NGC
207 Optimization Goal                   : Speed
208 Keep Hierarchy                      : YES
209 Target Technology                   : XC9500 CPLDs
210 Macro Preserve                      : YES
211 XOR Preserve                       : YES
212 wysiwyg                             : NO
213
214 Design Statistics
215 # IOs                               : 12
216
217 Cell Usage :
218 # BELS                               : 426
219 #   AND2                             : 120
220 #   AND3                             : 11
221 #   AND4                             : 3
222 #   AND5                             : 1
223 #   AND6                             : 1
224 #   AND7                             : 2
225 #   AND8                             : 1
226 #   INV                              : 177

```

```

227 #      OR2                : 67
228 #      OR3                : 5
229 #      XOR2               : 38
230 # FlipFlops/Latches      : 44
231 #      FD                 : 22
232 #      FDC                : 22
233 # IO Buffers             : 12
234 #      IBUF               : 4
235 #      OBUF               : 8
236 =====
237 CPU : 12.33 / 13.48 s | Elapsed : 12.00 / 14.00 s
238
239 —>
240
241 Total memory usage is 124148 kilobytes
242
243 Number of errors   :    0 (    0 filtered)
244 Number of warnings :    2 (    0 filtered)
245 Number of infos   :    0 (    0 filtered)

```

C.3. C Code

Der hier dargestellte C Code orientiert sich in einigen Teilen an den Vorarbeiten von Tobias Krannich und Stephan Plaschke.

Die allgemeine Struktur des C Codes und die grundlegende Hardwareinitialisierung wurde von Stephan Plaschke übernommen. Im Weiteren wurde das wesentliche Konzept zur Kapazitätsanpassung aus der Diplomarbeit von Tobias Krannich entnommen.

Die übrigen Softwarekomponenten wurden in Eigenregie entwickelt.

Listing C.12: Header File

```

1  /*-----
2      Project:          Reader Firmware
3      File:            main.h
4      Description:     Heder-File for the project
5      Used components: MSP430-169 Header Board
6      Date:            20.10.2008
7      Author:          Thorsten Eger
8  -----*/
9  #ifndef MAIN_H_
10 #define MAIN_H_
11
12 /*-----
13      Define the settings for the used msp
14  -----*/
15
16 /* Defines for selected frequencies, claculation: */
17 /* low power crystal frequency divided by 8 = 4096Khz */
18 /* timer uses DCO frequency and counts up in one ACLK wave */
19 /* e.g. 1/(2MHz)*488 = 0,000244s = 4098kHz */
20 #define MHZ_1    244
21 #define MHZ_2    488
22 #define MHZ_4    976
23 #define MHZ_8   1952
24

```

```

25 /* to enable USART Interrupts */
26 #define RX_INT      1 /* Receive Interrupt */
27 #define TX_INT      2 /* Transmit Interrupt */
28 #define RX_TX_INT   3 /* Receive and Transmit Interrupt */
29
30 /* to choose the USART directions */
31 #define RX          1 /* Receive */
32 #define TX          2 /* Transmit */
33 #define RX_TX      3 /* Receive and Transmit */
34
35 #define uC_FREQUENCY  MHZ_8 /* select the MSP430 frequency */
36 /* it depends on the used quartz */
37
38 #define USART_SEL     0x30 /* define USART pins */
39 #define USART_DIR     0x10 /* define USART pin direction */
40 #define USART_BAUD    19200 /* define USART BAUDRATE (4800,9600,19200 BAUD) */
41
42 #define ADC12_SEL     0x01 /* use P6.0 as A0 input */
43 /*-----*/
44     Define the relevant signals
45 /*-----*/
46 #define CPLD_RESET    (P1OUT |= BIT5)
47 #define CPLD_nRESET   (P1OUT &= ~BIT5)
48 #define CPLD_RESET_DIR BIT5
49
50 #define DIST_8        (P6IN & BIT3)
51 #define DIST_7        (P6IN & BIT4)
52 #define DIST_6        (P6IN & BIT5)
53 #define DIST_5        (P6IN & BIT6)
54 #define DIST_4        (P6IN & BIT7)
55 #define DIST_3        (P1IN & BIT0)
56 #define DIST_2        (P1IN & BIT3)
57 #define DIST_1        (P1IN & BIT4)
58
59 /*-----*/
60     Port settings, 1->output, 0->input
61     x_SEL is used for the special functions of the port
62 /*-----*/
63 #define PORT1_DIR     CPLD_RESET_DIR
64 #define PORT1_SEL     0x00
65
66 #define PORT2_DIR     0xFF
67 #define PORT2_SEL     0x00
68
69 #define PORT3_DIR     USART_DIR | 0x0F
70 #define PORT3_SEL     USART_SEL
71
72 #define PORT4_DIR     0xFF
73 #define PORT4_SEL     0x00
74
75 #define PORT5_DIR     0x00
76 #define PORT5_SEL     0x00
77
78 #define PORT6_DIR     0x00
79 #define PORT6_SEL     0x00
80
81 /*-----*/
82     Headerfiles
83 /*-----*/
84 #include <msp430x16x.h>

```

```

85 #include <stdio.h>
86 #include <signal.h>
87
88 /*-----
89     Prototypes
90 -----*/
91 unsigned char Set_XT2 (unsigned int);
92 void init_PORTS(void);
93 void init_USART0 (unsigned char, unsigned char);
94 void send_USART0 (unsigned char);
95 void init_ADC12(void);
96 int sampling_ADC120(void);
97 void set_CAPA(unsigned char);
98 void measure_AMPLITUDE(void);
99 void check_DIST(void);
100 void delay_FUNC(unsigned int);
101 void deactivate_all_ANT(void);
102 void activate_ANT1(void);
103 void activate_ANT2(void);
104 void activate_ANT3(void);
105 void activate_ANT4(void);
106 void activate_ANT5(void);
107 void activate_ANT6(void);
108 void reset_CPLD(void);
109
110 /*-----
111     Global variables
112 -----*/
113 unsigned short MAX_index;
114 unsigned short ADC_val;
115 unsigned int WAIT_val;
116 unsigned char CHANGE_antenna;
117 unsigned int SET_active;
118
119 #endif /*MAIN_H_*/

```

Listing C.13: Main File

```

1 /*-----
2     Project:           Reader Firmware
3     File:              main.c
4     Description:      Main-File for the project
5     Used components:  MSP430-169 Header Board
6     Date:             20.10.2008
7     Author:           Thorsten Eger
8 -----*/
9 #include <main.h>
10
11 int main (void)
12 {
13     unsigned char error_OSC;
14
15     /* Textmessages */
16     unsigned char welcome[] = "Wireless Sensors Reader \r\n"
17         "Activate a antenna by entering a number between 1 to 6 \r\n"
18         "Enter 0 to deactivate all antennas \r\n";
19
20     unsigned char error[] = "Wrong input \r\n"
21         "Enter an number between 1 to 6 to active a antenna\r\n";
22

```

```
23     WDTCTL = WDTPW | WDTHOLD; /* watchdog disable */
24
25     error_OSC = Set_XT2(uC_FREQUENCY); /* use external oscillator */
26
27     /* initialise the hardware */
28     init_USART0(RX_INT, RX_TX);
29     init_PORTS();
30     init_ADC12();
31
32     /* global interrupt enable */
33     _EINT();
34
35     /* initialise the used variables */
36     MAX_index = 0;
37     ADC_val = 0;
38     WAIT_val = 50000; /* value for nearly 50 ms delay */
39     CHANGE_antenna = 0; /* nothing to change */
40     SET_active = '0'; /* no active antenna */
41
42     /* reset the cpld */
43     reset_CPLD();
44
45     /* disable all antennas */
46     deactivate_all_ANT();
47
48     printf(welcome); /* show welcomemessage */
49
50     while(1)
51     {
52         if(CHANGE_antenna == 1)
53         {
54             switch(SET_active)
55             {
56                 case '0':
57                     deactivate_all_ANT();
58                     printf("All antennas deactivated \r\n");
59                     CHANGE_antenna = 0;
60                     break;
61
62                 case '1':
63                     activate_ANT1();
64                     measure_AMPLITUDE();
65                     check_DIST();
66                     CHANGE_antenna = 0;
67                     break;
68
69                 case '2':
70                     activate_ANT2();
71                     measure_AMPLITUDE();
72                     check_DIST();
73                     CHANGE_antenna = 0;
74                     break;
75
76                 case '3':
77                     activate_ANT3();
78                     measure_AMPLITUDE();
79                     check_DIST();
80                     CHANGE_antenna = 0;
81                     break;
82
```

```

83         case '4':
84             activate_ANT4();
85             measure_AMPLITUDE();
86             check_DIST();
87             CHANGE_antenna = 0;
88             break;
89
90         case '5':
91             activate_ANT5();
92             measure_AMPLITUDE();
93             check_DIST();
94             CHANGE_antenna = 0;
95             break;
96
97         case '6':
98             activate_ANT6();
99             measure_AMPLITUDE();
100            check_DIST();
101            CHANGE_antenna = 0;
102            break;
103
104         default:
105             printf(error);
106             deactivate_all_ANT();
107             printf("All antennas deactivated \r\n");
108             CHANGE_antenna = 0;
109             break;
110
111     } /* end of switch */
112 } /* end of if */
113 } /* end of while(1) */
114
115 return 0;
116
117 }

```

Listing C.14: Function File

```

1  /*-----
2  Project:           Reader Firmware
3  File:             MSP_Functions.c
4  Description:      Function-File for the project
5  Used components:  MSP430-169 Header Board
6  Date:            20.10.2008
7  Author:          Thorsten Eger
8  -----*/
9  #include <main.h>
10
11 /*-----
12 Set XT2 to selected frequency
13 clk_multi is the multiplier of the A_clk (32768KHz)
14 Error_code:
15 0 -> no error
16 1 -> external oscillator error
17 2 -> clock wrong, set to 4MHZ */
18 unsigned char Set_XT2 (unsigned int clk_multi)
19 /*-----*/
20 {
21     unsigned int Max_turns = 1000;
22     unsigned char i, Error_code = 0;

```

```

23
24     BCCTL1 &= ~XT2OFF;           /* XT2on */
25
26     do
27     {
28         IFG1 &= ~OFIFG;         /* Clear OSCFault flag */
29         for (i = 0xFF; i > 0; i--); /* Time for flag to set */
30
31         if (!(--Max_turns)) /* stop after 1k tries, error with ext osc! */
32         {
33             Error_code = 1;
34             break;
35         }
36     }
37     while ((IFG1 & OFIFG));      /* OSCFault flag still set? */
38
39     BCCTL2 |= SELM_2 | SELS;     /* MCLK = SMCLK = XT2 (safe) */
40
41     switch (clk_multi)
42     {
43     case MHZ_8: BCCTL2 &= ~(DIVS_3 | DIVM_3); /* SMCLK&MCLK = XT2 / 1 */
44                 break;
45     case MHZ_4: BCCTL2 &= ~(DIVS_0 | DIVM_0); /* SMCLK&MCLK = XT2 / 2 */
46                 BCCTL1 &= ~DIVA_0; /* ACLK = XT2 / 2 */
47                 break;
48     case MHZ_2: BCCTL2 |= DIVS_1 | DIVM_1; /* SMCLK&MCLK = XT2 / 4 */
49                 BCCTL1 |= DIVA_1; /* ACLK = XT2 / 4 */
50                 break;
51     case MHZ_1: BCCTL2 |= DIVS_2 | DIVM_2; /* SMCLK&MCLK = XT2 / 8 */
52                 BCCTL1 |= DIVA_2; /* ACLK = XT2 / 8 */
53                 break;
54     default:    Error_code = 2;
55                 BCCTL2 |= DIVS_1 | DIVM_1; /* SMCLK&MCLK = XT2 / 2 */
56                 BCCTL1 |= DIVA_1; /* ACLK = XT2 / 2 */
57                 break;
58     }
59
60     return (Error_code);
61 }
62
63 /*-----
64     putchar */
65 int putchar(int c)
66 /*-----*/
67 { /* putchar output USART */
68     send_USART0((unsigned char)c);
69     return c;
70 }
71
72 /*-----
73     send character */
74 void send_USART0 (unsigned char out_char)
75 /*-----*/
76 {
77     while(!(UTCTL0&TXEPT)); /* look if USART0 is still busy */
78     TXBUF0 = out_char;     /* write char to transmit buffer */
79 }
80
81 /*-----
82     PORTS init */

```

```

83 void init_PORTS(void)
84 /*-----*/
85 {
86     P1SEL = PORT1_SEL;      /* Port1 function */
87     P1DIR = PORT1_DIR;     /* Port1 select direction */
88
89     P2SEL = PORT2_SEL;     /* Port2 function */
90     P2DIR = PORT2_DIR;     /* Port2 select direction */
91
92     P3SEL = PORT3_SEL;     /* Port3 function */
93     P3DIR = PORT3_DIR;     /* Port3 select direction */
94
95     P4OUT = PORT4_SEL;     /* Port4 function */
96     P4DIR = PORT4_DIR;     /* Port4 select direction */
97
98     P5SEL = PORT5_SEL;     /* Port5 function */
99     P5DIR = PORT5_DIR;     /* Port5 select direction */
100
101     P6SEL = PORT6_SEL;     /* Port6 function */
102     P6DIR = PORT6_DIR;     /* Port6 select direction */
103 }
104
105 /*-----*/
106     Init USART0 */
107 void init_USART0 (unsigned char USART0_IRQ, unsigned char USART0_MODE)
108 /*-----*/
109 {
110     UCTL0 = SWRST + CHAR; /* set 8bit, none parity and 1 stoppbit */
111     UTCTL0 |= SSEL_2; /* set SMCLK as USART0_CLK */
112
113     switch (uC_FREQUENCY) /* set UART to selected mode with selected uC frequency */
114     {
115     case MHZ_1: switch (USART_BAUD)
116         {
117         case 19200: UBR00 = 0x34; /* see manual */
118             UBR10 = 0x00;
119             UMCTL0 = 0x20;
120             break;
121         case 9600:  UBR00 = 0x68; /* see manual */
122             UBR10 = 0x00;
123             UMCTL0 = 0x04;
124             break;
125         case 4800:  UBR00 = 0xD0; /* see manual */
126             UBR10 = 0x00;
127             UMCTL0 = 0x92;
128             break;
129         };
130         break;
131     case MHZ_2: switch (USART_BAUD)
132         {
133         case 19200: UBR00 = 0x68; /* see manual */
134             UBR10 = 0x00;
135             UMCTL0 = 0x04;
136             break;
137         case 9600:  UBR00 = 0xD0; /* see manual */
138             UBR10 = 0x00;
139             UMCTL0 = 0x92;
140             break;
141         case 4800:  UBR00 = 0xA0; /* see manual */
142             UBR10 = 0x01;

```

```
143         UMCTL0 = 0x6D;
144         break;
145     };
146     break;
147     case MHZ_4: switch (USART_BAUD)
148     {
149         case 19200: UBR00 = 0xD0; /* see manual */
150                 UBR10 = 0x00;
151                 UMCTL0 = 0x92;
152                 break;
153         case 9600:  UBR00 = 0xA0; /* see manual */
154                 UBR10 = 0x01;
155                 UMCTL0 = 0x6D;
156                 break;
157         case 4800:  UBR00 = 0x41; /* see manual */
158                 UBR10 = 0x03;
159                 UMCTL0 = 0x92;
160                 break;
161     };
162     break;
163     case MHZ_8: switch (USART_BAUD)
164     {
165         case 19200: UBR00 = 0xA0; /* see manual */
166                 UBR10 = 0x01;
167                 UMCTL0 = 0x6D;
168                 break;
169         case 9600:  UBR00 = 0x41; /* see manual */
170                 UBR10 = 0x03;
171                 UMCTL0 = 0x92;
172                 break;
173         case 4800:  UBR00 = 0x82; /* see manual */
174                 UBR10 = 0x6;
175                 UMCTL0 = 0x6D;
176                 break;
177     };
178     break;
179 }
180
181 switch (USART0_MODE)
182 {
183     case RX: ME1 |= URXE0; /* enable UART0 RX */
184             break;
185     case TX: ME1 |= UTXE0; /* enable UART0 TX */
186             break;
187     case RX_TX: ME1 |= (URXE0 | UTXE0); /* enable UART0 TX&RX */
188             break;
189     default: ME1 &= ~(UTXE0 | URXE0); /* disable UART0 TX&RX */
190             break;
191 }
192
193 UCTL0  &= ~SWRST; /* clear SWRST */
194
195 switch (USART0_IRQ)
196 {
197     case RX_INT: IE1 |= URXIE0; /* enable UART0 RX ISR */
198             break;
199     case TX_INT: IE1 |= UTXIE0; /* enable UART0 TX ISR */
200             break;
201     case RX_TX_INT: IE1 |= UTXIE0+URXIE0; /* enable UART0 TX&RX ISR */
202             break;
```

```

203         default: IE1 &= ~UTXIE0+URXIE0; /* disable UART0 TX&RX ISR */
204             break;
205     }
206 }
207
208 /*-----*/
209     init ADC12 */
210 void init_ADC12(void)
211 /*-----*/
212 {
213     /* activate ADC, no reference voltage */
214     ADC12CTL0 = ADC12ON;
215     P6SEL |= 0x01; /* use P6.0 as analog input A0 */
216 }
217
218 /*-----*/
219     get ADC12 result */
220 int sampling_ADC120(void)
221 /*-----*/
222 {
223     ADC12CTL0 |= ADC12SC + ENC; /* Sampling open */
224     ADC12CTL0 &= ~ADC12SC; /* Sampling closed, start conversion */
225     while ((ADC12CTL1 & ADC12BUSY) == 1); /* ADC12 busy? */
226     return(ADC12MEM0); /* return the value read from ADC */
227 }
228
229 /*-----*/
230     set the capacitance */
231 void set_CAPA(unsigned char cap)
232 /*-----*/
233 {
234     unsigned char flip = 0;
235     unsigned char i;
236
237     for(i=0; i<8; i++)
238     { /* reverse the bitorder, see board schematic */
239         flip |= ( (cap >> i) & 0x01 ) << (7 - i);
240     }
241     /* set the bustrivers to activate the wished capacitance
242     * the drivers are low-active, therefore they are activated
243     * by zero */
244     P4OUT = ~flip;
245 }
246
247 /*-----*/
248     determine the right capacitance*/
249 void measure_AMPLITUDE(void)
250 {
251     unsigned short i;
252     unsigned short ADC_voltage = 0;
253     unsigned short MAX_ADC_voltage = 0;
254
255     /* test all combinations */
256     for(i=0; i<0xFF; i++)
257     {
258         set_CAPA(i); /* activate capacitance */
259         /* determine voltage */
260         ADC_voltage = sampling_ADC120();
261         /* set the actual maximum */
262         if(ADC_voltage > MAX_ADC_voltage)

```

```
263     {
264         MAX_ADC_voltage = ADC_voltage;
265         MAX_index = i;
266     }
267 }
268
269 /* activate capacitance with the highest amplitude */
270 set_CAPA(MAX_index);
271 }
272
273 /* -----
274 determine the measured signal distance */
275 void check_DIST(void)
276 {
277     /* first reset the cpld */
278     reset_CPLD();
279
280     /* show the measured value */
281     if (DIST_1)
282     {
283         printf("DIST_1\r\n");
284     }
285     else if (DIST_2)
286     {
287         printf("DIST_2\r\n");
288     }
289     else if (DIST_3)
290     {
291         printf("DIST_3\r\n");
292     }
293     else if (DIST_4)
294     {
295         printf("DIST_4\r\n");
296     }
297     else if (DIST_5)
298     {
299         printf("DIST_5\r\n");
300     }
301     else if (DIST_6)
302     {
303         printf("DIST_6\r\n");
304     }
305     else if (DIST_7)
306     {
307         printf("DIST_7\r\n");
308     }
309     else if (DIST_8)
310     {
311         printf("DIST_8\r\n");
312     }
313     else
314     {
315         printf("Error: check_DIST()\r\n");
316         printf("Reconfigure the capacitances \r\n");
317         reset_CPLD();
318         measure_AMPLITUDE();
319         check_DIST();
320         CHANGE_antenna = 0;
321     }
322 }
```

```
323
324 /* -----
325     delay function, wait nearly n microseconds at 8MHz */
326 void delay_FUNC(unsigned int delay)
327 {
328     do
329     {
330         nop();
331     } while(--delay);
332 }
333
334 /* -----
335     deactivate all antennas */
336 void deactivate_all_ANT(void)
337 {
338     /* deactivate all antennas */
339     P2OUT = 0x00;
340     P3OUT &= ~0x0F;
341     delay_FUNC(WAIT_val);
342 }
343
344 /* -----
345     activate antenna 1 */
346 void activate_ANT1(void)
347 {
348     deactivate_all_ANT();
349
350     /* activate antenna 1*/
351     P3OUT |= 0x0C;
352     delay_FUNC(WAIT_val);
353     printf("Antenna 1 activated \r\n");
354 }
355
356 /* -----
357     activate antenna 2 */
358 void activate_ANT2(void)
359 {
360     deactivate_all_ANT();
361
362     /* activate antenna 3*/
363     P3OUT |= 0x03;
364     delay_FUNC(WAIT_val);
365     printf("Antenna 2 activated \r\n");
366 }
367
368 /* -----
369     activate antenna 3 */
370 void activate_ANT3(void)
371 {
372     deactivate_all_ANT();
373
374     /* activate antenna 3*/
375     P2OUT = 0xC0;
376     delay_FUNC(WAIT_val);
377     printf("Antenna 3 activated \r\n");
378 }
379
380 /* -----
381     activate antenna 4 */
382 void activate_ANT4(void)
```

```

383 {
384     deactivate_all_ANT();
385
386     /* activate antenna 4*/
387     P2OUT = 0x30;
388     delay_FUNC(WAIT_val);
389     printf("Antenna 4 activated \r\n");
390 }
391
392 /* -----
393     activate antenna 5 */
394 void activate_ANT5(void)
395 {
396     deactivate_all_ANT();
397
398     /* activate antenna 5*/
399     P2OUT = 0x0C;
400     delay_FUNC(WAIT_val);
401     printf("Antenna 5 activated \r\n");
402 }
403
404 /* -----
405     activate antenna 6 */
406 void activate_ANT6(void)
407 {
408     deactivate_all_ANT();
409
410     /* activate antenna 6 */
411     P2OUT = 0x03;
412     delay_FUNC(WAIT_val);
413     printf("Antenna 6 activated \r\n");
414 }
415
416 /* -----
417     reset the cpld */
418 void reset_CPLD(void)
419 {
420     /* reset the cpld */
421     CPLD_RESET;
422     delay_FUNC(WAIT_val);
423     CPLD_nRESET;
424 }

```

Listing C.15: ISR File

```

1  /*-----
2      Project:           Reader Firmware
3      File:             ISR.c
4      Description:      ISR-File for the project
5      Used components:  MSP430-169 Header Board
6      Date:            20.10.2008
7      Author:          Thorsten Eger
8  -----*/
9  #include <main.h>
10
11 /*-----
12     UART0 RX interrupt service routine */
13 interrupt (USARTORX_VECTOR) UART0_RX(void)
14 /*-----*/
15 {

```

```
16     SET_active = RXBUF0; /* use input to activate a antenna */
17     CHANGE_antenna = 1; /* symbolise to change a antenna */
18 }
```

D. Schaltplan

D.1. Schaltplan zum Reader

Der erstellte Schaltplan erstreckt sich über mehrere Seiten. Auf der ersten Seite bildet die Spannungsversorgung der Platine und die Aufnahme des Mikrocontrollers den Hauptteil. Auf der zweiten Seite sind der verwendete CPLD und der Spannungskomparator dargestellt. Die Schwingkreisanregung und die Antennenselektion befinden sich auf Seite 3 des Schaltplans. Die letzte Seite des Schaltplans beinhaltet die verwendeten Filterschaltungen.

Zur besseren Übersicht folgen an dieser Stelle noch einige tabellarischen Auflistungen der Portzuordnung des eingesetzten Mikrocontrollers.

Benennung	Kapazitätswert in pF	Port	Pin
C_1	1000	$P4.0$	36
C_2	1000	$P4.1$	37
C_3	470	$P4.2$	38
C_4	220	$P4.3$	39
C_5	220	$P4.4$	40
C_6	100	$P4.5$	41
C_7	47	$P4.6$	42
C_8	22	$P4.7$	43

Tabelle D.1.: Portzuordnung der gewählten Kapazitäten

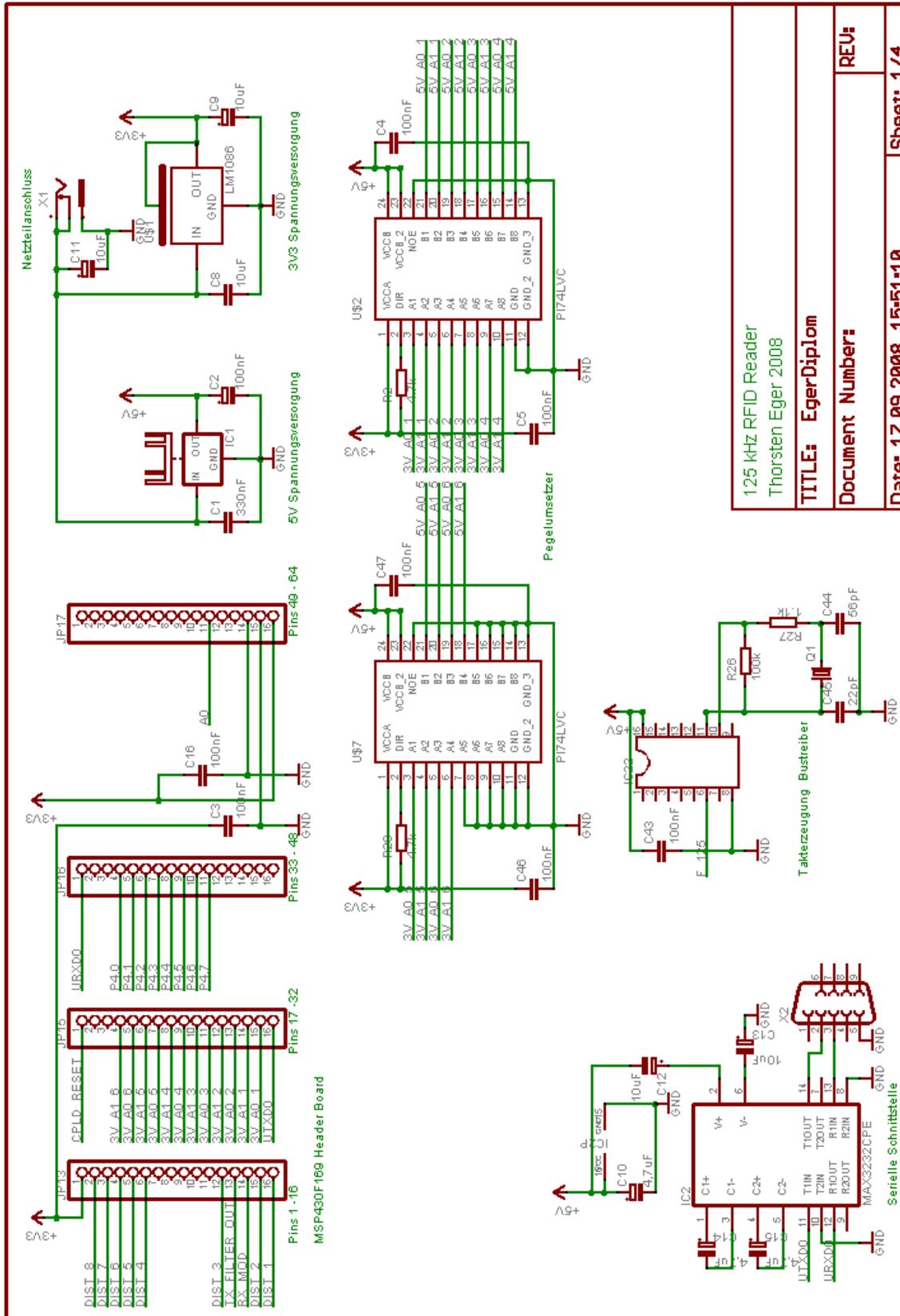


Bild D.1.: Schaltplan der Readerplatine Seite 1

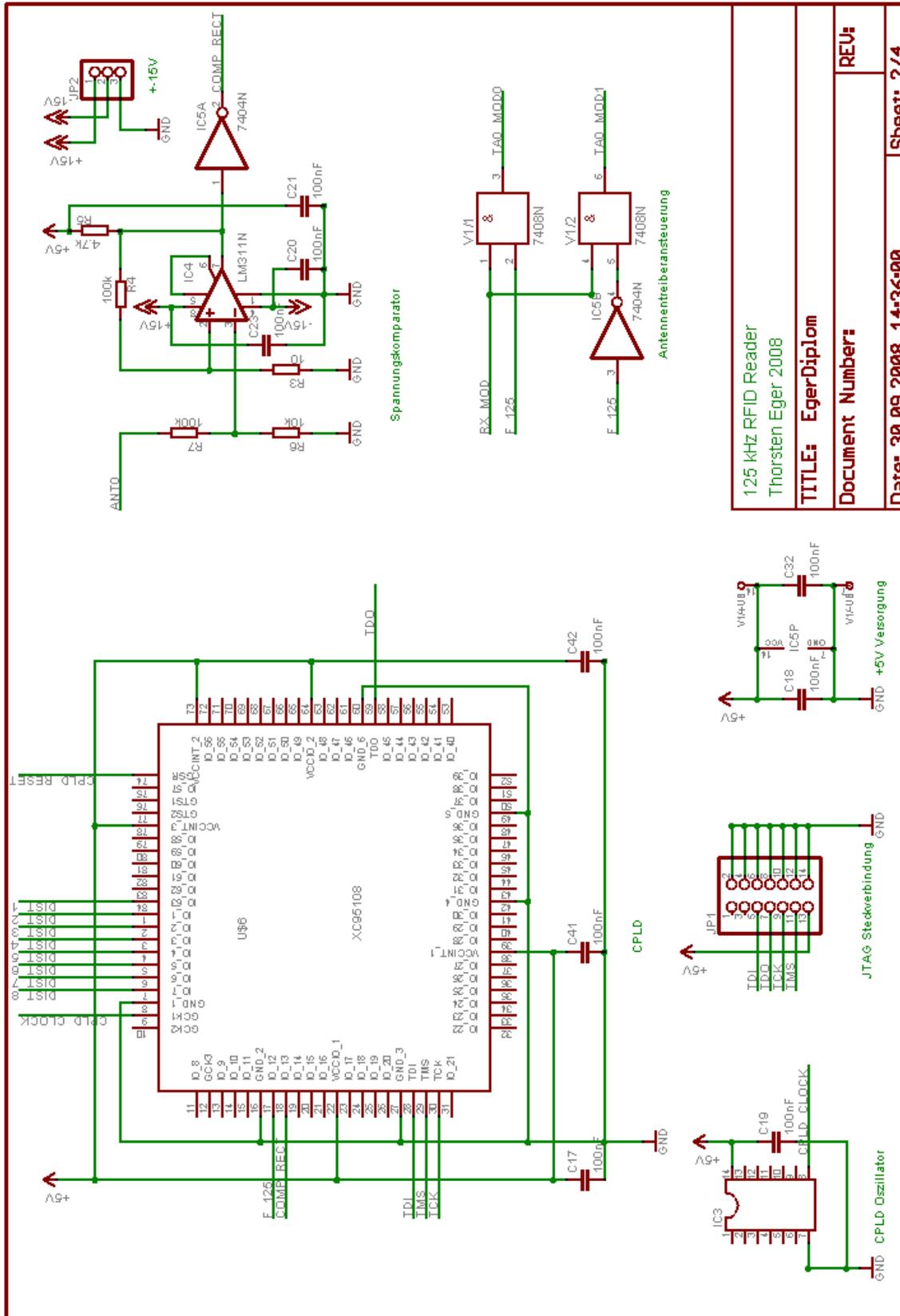


Bild D.2.: Schaltplan der Readerplatine Seite 2

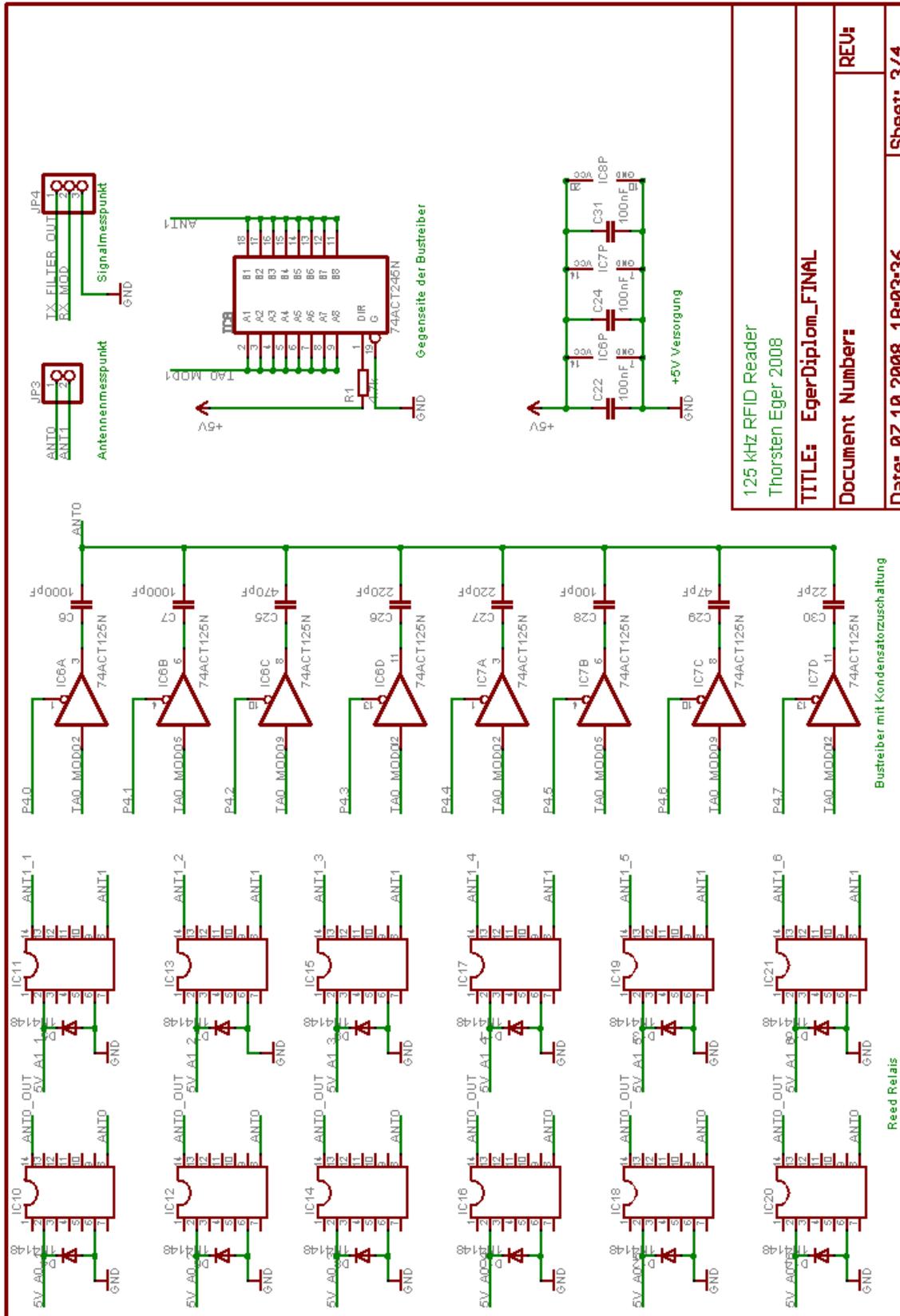
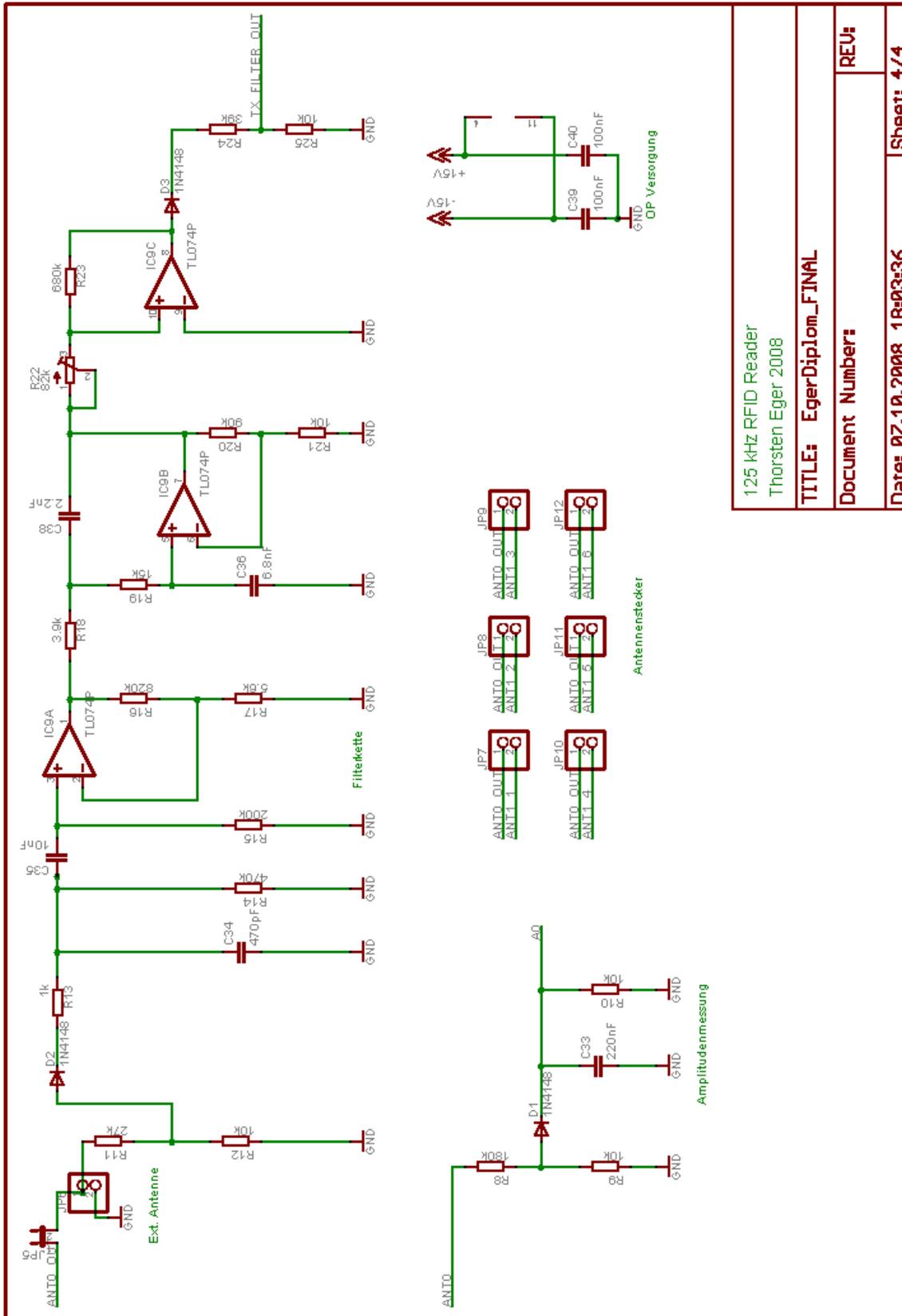


Bild D.3.: Schaltplan der Readerplatine Seite 3



125 kHz RFID Reader
Thorsten Eger 2008

TITLE: EgerDiplom_FINAL

Document Number:

REV:

Date: 07.10.2008 18:03:36

Sheet: 4/4

Bild D.4.: Schaltplan der Readerplatine Seite 4

Signal	Port	Pin
$DIST_1$	$P1.4$	16
$DIST_2$	$P1.3$	15
$DIST_3$	$P1.0$	12
$DIST_4$	$P6.7$	6
$DIST_5$	$P6.6$	5
$DIST_6$	$P6.5$	4
$DIST_7$	$P6.4$	3
$DIST_8$	$P6.3$	2

Tabelle D.2.: Portzuordnung der Eingangssignale zum Resonanzabgleich

Signal	Port	Pin
$A0_1$	$P3.3$	31
$A1_1$	$P3.2$	30
$A0_2$	$P3.1$	29
$A1_2$	$P3.0$	28
$A0_3$	$P2.7$	27
$A1_3$	$P2.6$	26
$A0_4$	$P2.5$	25
$A1_4$	$P2.4$	24
$A0_5$	$P2.3$	23
$A1_5$	$P2.2$	22
$A0_6$	$P2.1$	21
$A1_6$	$P2.0$	20

Tabelle D.3.: Portzuordnung der Ausgangssignale zur Antennenselektion

Versicherung der Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §25(4) ohne fremde Hilfe selbständig 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.

Ort, Datum:

Unterschrift: