

Bachelorarbeit
Fakultät Mechatronik und Medizintechnik
Studiengang Bachelor Mechatronik

**Entwicklung eines digitalen standalone
Ultraschall-Multigate-Doppler-Systems**

vom 01.10.2013 bis 31.01.2014

Verfasser: Andreas Rehn
König-Wilhelm-Str. 23
89073 Ulm
rehn.andreas86@gmail.com

1. Gutachter: Prof. Dr.-Ing. Rainer Brucher
2. Gutachter: Prof. Dr. Malte Groß

Zusammenfassung

In der diagnostischen Medizin werden vorwiegend invasive Verfahren zur Bestimmung des Hämatokritwertes eingesetzt. Diese haben den Nachteil, zeitaufwändig, kostenintensiv und schmerhaft zu sein. Nichtinvasive Verfahren hingegen können den genannten Defiziten entgegenwirken und werden bereits in der diagnostischen Medizin eingesetzt.

Darum befasst sich die vorliegende Arbeit mit der Entwicklung eines nicht-invasiven Verfahrens auf Grundlage der gepulsten Doppler-Technologie.

Auf Basis dieser Technologie wurde ein digitales standalone Ultraschall-Multigate-Doppler-System entwickelt. Dieses besteht aus einer Messplatine, einer Steuerplatine sowie aus einem resistiven Touchdisplay. Dabei kann das System über die integrierte Touchfunktion oder über ein Personal Computer (PC) Programm gesteuert und die gewünschten Messtiefen eingestellt werden. Anschließend werden die getroffenen Einstellungen an das Complex Programmable Logic Device (CPLD) der Messplatine übertragen, welcher einen Transmitter mit differentiellen Ausgang ansteuert. Über Diesen wird die erforderliche Leistung für die Ultraschallsonde bereitgestellt, woraufhin durch die Sonde eine Ultraschallwelle in das Gewebe des Patienten eingestrahlt wird.

Während die Ultraschallwelle das Gewebe durchdringt, reflektieren Zellwände und Blutteilchen Teile der Welle. Diese Reflexionen können je nach Bedarf durch die Ultraschallsonde oder durch zum Beispiel (z.B.) ein Hydrophon in elektrische Signale umgewandelt werden. Folgend werden diese verstärkt und mithilfe eines Analog-Digital Converters (Analog-Digital Converter (ADC)) digitalisiert. Dabei steuert das CPLD die Digitalisierung und bereitet die aufgenommenen Daten durch Demodulation und Filterung auf.

Die demodulierten Daten werden anschließend durch ein Serial Peripheral Interface (SPI) an den ARM Cortex-M3 Mikrokontroller (uC) übertragen, welcher diese auf den Touchdisplay als graphische Kurven darstellt oder an den PC weiterreicht.

In den zur Verfügung stehenden Zeitraum konnte die Ansteuerung des Transmitters und des Receivers, eine Kommunikation zwischen PC und dem System aufgebaut, sowie ein Graphical User Interface (GUI) mit Touch-Auswertung realisiert werden.

Jedoch konnte die Digitalisierung der Eingangssignale nicht realisiert werden, wodurch eine Inbetriebnahme des gesamten Systems nur teilweise ausführbar war. Als Folge dessen kann nicht garantiert werden, dass die angestrebte Datenmenge ohne Verluste verarbeitet wird, wodurch auch die resultierende Darstellung diese verarbeitungsbedingten Fehler enthalten kann. Den-

Entwicklung eines digitalen standalone Ultraschall-Multigate-Doppler-Systems

noch konnte hier die grundlegende Funktionweise der Analysemethode nachgewiesen werden. Somit wurde die Grundlage für weitere Nachforschungen geschaffen um die hier dargestellte Analysemethode weiter zu optimieren.

Abstract

In diagnostic medicine are primarily used invasive procedures for determining the hematocrit value. These procedures have the disadvantage of being time-consuming, expensive and painful. Non-invasive procedures, however, can avoid the aforementioned deficits and are already used in diagnostic medicine.

A study of the development of a non-invasive method based on the pulsed Doppler technology, is reported here.

Based on this technology, a digital stand-alone ultrasonic multi-gate Doppler system was developed. The system of this study consists of a measuring board, a control board and a resistive touch screen. In this case the system is controlled via the integrated touch function or a Personal Computer (PC) program whereby desired measurement depths can be adjusted. Subsequently, the settings made are transferred to the Complex Programmable Logic Device (CPLD) on the measuring board, which controls a transmitter with differential output. By this component part the power required for the ultrasound probe is provided, after which an ultrasonic wave is irradiated into the tissue of the patient through the probe.

While the ultrasonic wave passes through the tissue, cell walls and blood corpuscle reflect parts of the shaft. These reflections are converted into electrical signals by the ultrasonic probe, or by for example, a hydrophone. Subsequently these electrical signals are amplified and digitized by a high speed Analog-Digital Converter (ADC). Here, the CPLD controls the digitization and prepares the recorded data by demodulation and filtering.

The demodulated data is then transmitted through a Serial Peripheral Interface (SPI) to the ARM Cortex-M3 microcontroller (uC), which displays the data in form of graphical curves or forwards the data to the PC.

During the available period of time the control of the transmitter and the receiver, a communication between the PC and the system was built, and a GUI with touch evaluation was realized.

However, the digitization of the input signals could not be realized, therefore putting the entire system was only partially executable. As a result it can not be guaranteed, that the desired data rate or amount of data can be processed lossless and therefore the resulting graph can show these processing dependent problems, too. Nevertheless the basic mechanisms and the functionality of the shown analysis method have been demonstrated. Therefore the basis for further research in order to optimize the shown analysis method was made.

Eidesstattliche Erklärung

Hiermit versichere ich, Andreas Rehn, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen oder Hilfsmittel verwendet habe. Alle Ausführungen, die fremden Quellen wörtlich oder sinngemäß entnommen wurden, sind als solche kenntlich gemacht.

Diese Arbeit war in gleicher oder ähnlicher Form noch nicht Bestandteil einer Studien- oder Prüfungsleistung.

Ulm, den 6. Februar 2014

Andreas Rehn

Ort, Datum

Unterschrift

Inhaltsverzeichnis

Zusammenfassung	II
Abstract	IV
Eidesstattliche Erklärung	V
Abbildungsverzeichnis	X
Tabellenverzeichnis	XI
1. Abkürzungen und Begriffsdefinition	1
1.1. Abkürzungen	1
1.2. Begriffsdefinitionen	3
2. Einleitung und Aufgabenstellung	4
2.1. Einleitung	4
2.2. Aufgabenstellung	5
2.2.1. Aufgabenstellung der Bachelorarbeit	5
2.2.2. Vorstellung des vorhandenen Systems	6
2.2.3. Anforderungen an das Messsystem	7
2.3. Beschreibung der Methodik	10
3. Grundlagen	11
3.1. Ultraschall	11
3.1.1. Definition	11
3.1.2. Erzeugung und Empfang	12
3.1.3. Ausbreitung	12
3.1.4. Reflexion und Brechung	13
3.1.5. Absorption und Streuung	14
3.1.6. Dopplereffekt	14
3.2. Pulsed Wave Dopplerverfahren	15
4. Stand der Technik	17
4.1. Auswertung der Ultraschallmessung und Bildentstehung	17
4.2. Touchpanel Technologien	18
4.2.1. Definition	18

4.2.2. Resistiv	18
4.2.3. Induktiv	19
4.2.4. Kapazitiv	19
4.2.5. Akustisch	20
4.2.6. Optisch	20
5. Material und Methode	21
5.1. Material	21
5.1.1. Verwendete Programme	21
5.1.2. Verwendete Geräte und Evalkits	22
5.2. Methode	23
5.2.1. Quadraturdemodulation	23
5.2.2. Serial Peripheral Interface	24
5.2.3. Fast Fourier Transformation	25
6. Systemdesign und Implementierung	27
6.1. Messplatine	28
6.1.1. Schnittstellen	28
6.1.2. Complex Programmable Logic Device	29
6.1.3. Transmitter	30
6.1.4. Receiver	31
6.2. Display Platine	31
6.3. Steuer- und Auswertungsplatine	33
6.3.1. Energieversorgung	33
6.3.2. Audio Verstärker	33
6.3.3. ARM Cortex M3	33
6.4. Software	35
6.4.1. CPLD	36
6.4.2. ARM Cortex M3	40
6.4.3. Kalkulation der State-Zählerwerte N	43
7. Test und Ergebnisse	45
7.1. Komponententest (Unit-Test)	45
7.1.1. Debuggen des Cortex M3	45
7.1.2. Cortex M3 - USB	46
7.1.3. Display	47
7.1.4. Touchfunktion	48
7.1.5. AudioOut	49
7.1.6. Datenkommunikation via DMA und SPI	49

7.1.7. CPLD - Transmitter	51
7.1.8. CPLD - Statehandler	51
7.1.9. CPLD - Output Interface	52
7.2. Systemtest (Integration-Test)	53
8. Diskussion und Ausblick	55
8.1. Zusammenfassung der Ergebnisse	55
8.1.1. Messplatine	55
8.1.2. Steuerplatine	56
8.1.3. Display & Touch	56
8.1.4. System	56
8.2. Verbesserungspotentiale und Ausblick	57
8.2.1. Messplatine - Hardware	57
8.2.2. CPLD - Software	57
8.2.3. Display	57
8.2.4. Steuerplatine - Hardware	58
8.2.5. ARM Cortex-M3 - Software	58
8.2.6. Kostenreduzierung	59
8.2.7. Mobilität	59
Schlusswort	61
Literaturverzeichnis	62
A. Quellcode	65
A.1. Audio Test	65
A.2. Draw Images	67
B. Datenträger Inhalt	68
B.1. Dokumentation	68
B.2. Hardware	68
B.3. Software	68
C. Schaltpläne und Layouts	69
C.1. Cortex Debugger	69
C.2. Ultraschall Doppler System	72

Abbildungsverzeichnis

2.1. Methodik: Multi-gate Ultraschall Doppler - Vereinfachte Darstellung	6
2.2. Systemaufbau Thesis Stemplewitz	7
3.1. Frequenzbereiche Schall	11
3.2. Piezoelektrischer Effekt	12
3.3. Ereignis-Zeitdiagramm	15
4.1. Modes der Bildentstehung	18
4.2. Aufbau eines resistiven Touchpanels	19
5.1. Blockdiagramm Quadraturdemodulation und Tiefpass	23
5.2. SPI Schieberegister Master-Slave	25
5.3. SPI Datenübertragung mit unterschiedlichen Einstellungen	25
6.1. Systemübersicht	27
6.2. Messplatine - Revision 002	28
6.3. Transmitter Schaltung	30
6.4. Receiver Schaltung	31
6.5. LCD TFT breakout board	32
6.6. Audioverstärker und Tiefpass	33
6.7. Blockdiagramm NXP LPC176X	34
6.8. Module CPLD und Architektur	37
6.9. Modul State-Handler - Statemachine	38
6.10. kalibriertes Menu	42
7.1. Grafiktest	46
7.2. USB Audio Test	47
7.3. Grafiktest	47
7.4. Tests - Touchfunktion	48
7.5. Ergebnis direct memory access (DMA) & SPI Mode 3	50
7.6. Ergebnis Transmitter Steuerung	51
7.7. Ergebnisse StateHandler	52
7.8. Ergebnis CPLD SPI slave Mode 3	52
7.9. Test Trägerfrequenz	53
7.10. Test Burst Tiefe und Pulse Repetition Frequency (PRF)	53

Entwicklung eines digitalen standalone Ultraschall-Multigate-Doppler-Systems

8.1. Schaltung zur energieeffiziente Spannungsmessung	59
---	----

Tabellenverzeichnis

1.1. Begriffsdefinitionen	3
2.1. Bestandteile der Platine	7
3.1. Schallgeschwindigkeiten in unterschiedlichen Materialien und Geweben	16
3.2. Eindringtiefen als Funktion der Frequenz in menschlichen Geweben	16
5.1. SPI Modi Einstellungen	25
6.1. Aufbau eines States	35
6.2. Actions	36
7.1. Messwerte X-, Y-Positionen	49

1. Abkürzungen und Begriffsdefinition

1.1. Abkürzungen

ADC	Analog-Digital Converter
ALU	Arithmetic logic unit
BOM	Bill of Materials
bzw.	beziehungsweise
ca.	circa
CMSIS	Cortex Microcontroller Software Interface Standard
CPLD	Complex Programmable Logic Device
DAC	Digital-Analog Converter
DC	Direct Current
DFT	Diskrete Fourier Transformation
d.h.	das heißt
DMA	direct memory access
FFT	Fast Fourier Transformation
FIFO	FirstInFirstOut
FIR	finite impulse response
FPGA	Field Programmable Gate Array
FPU	floating-point unit
GHz	Giga Herz
GUI	Graphical User Interface
HAL	Hardware Abstraction Layer
HDK	Hardware Development Kit
HDL	Hardware Description Language
IC	Integrated Circuit
IDE	integrated development environment
ITO	Indium-Tin-Oxide

Entwicklung eines digitalen standalone Ultraschall-Multigate-Doppler-Systems

kHz	Kilo Herz
LUT	Look-up Table
mA	milli Ampere
MHz	Mega Herz
MCV	mittleres corpusculäres Volumen der Erythrozyten
MOSFET	metal oxide semiconductor field-effect transistor
MSps	Mega Samples per second
NF	Niederfrequenz
OOP	Objektorientierte Programmierung
PC	Personal Computer
PCB	Printed Circuit Board
pp	peak-peak
PRF	Pulse Repetition Frequency
PW	Pulsed Wave
RAM	Random-Access Memory
R&D	research and development
resp.	respektive
RTC	Real Time Clock
SAW	Surface Acoustic Wave
SSP	Streaming Serial Port
SPI	Serial Peripheral Interface
SPP	Streaming Parallel Port
uC	Mikrokontroller
USB	Universal Serial Bus
VHDL	Very High Speed Integrated Circuit Hardware Description Language
z.B.	zum Beispiel

1.2. Begriffsdefinitionen

Tabelle 1.1.: Begriffsdefinitionen

Begriff	Definition
Arithmetic logic unit (ALU)	Arithmetisch-logische Einheit ist ein elektronisches Rechenwerk welche in Prozessoren zum Einsatz kommt.
Bill of Materials (BOM)	Stückliste
finite impulse response (FIR)	Filter mit endlicher Impulsantwort oder Transversalfilter welcher ein diskreter, meist digitaler Filter darstellt
Hardware Abstraction Layer (HAL)	Hardwareabstraktionsschicht
integrated development environment (IDE)	integrierte Entwicklungsumgebung
Interface	Schnittstelle zwischen Funktionen oder Geräten
Indium-Tin-Oxide (ITO)	Molekülverbindung aus Indium-Zinn-Oxid
Printed Circuit Board (PCB)	Leiterplatte
Pulsed Wave (PW) Dopplerverfahren	Gepulstes Dopplerverfahren
Schallimpedanz	Verhältnis von Schalldruck zu Schallschnelle
Surface Acoustic Wave (SAW)	Akustische Oberflächenwelle

2. Einleitung und Aufgabenstellung

2.1. Einleitung

Blut ist eine der wichtigsten Flüssigkeiten des menschlichen Organismus. Es setzt sich aus festen - Erythrozyten, Leukozyten und Thrombozyten - und flüssigen Bestandteilen - Plasma - zusammen. Das Gesamtvolumen besteht zu 45% aus den festen Bestandteilen wobei das Plasma zu 90% aus Wasser besteht.

Erythrozyten werden auch Blutkörperchen genannt und enthalten das wichtige Hämoglobin, welches auch als eisenhaltiges, Sauerstoff transportierendes Protein bezeichnet wird. Die gebundenen Eisenionen des Proteins geben dabei dem Blut seinen "roten Farbstoff". Erythrozyten bilden die Mehrheit der gesamten Blutteilchen mit circa (ca.) 96%.

Leukozyten dagegen, sind weiße Blutkörperchen und haben die wichtige Aufgabe der Immunabwehr.

Thrombozyten werden auch als Blutplättchen bezeichnet und besitzen keinen Kern. Sie initialisieren die Freisetzung von Plättchenfaktoren, welche die Blutgerinnung einleiten.

Bluteigenschaften sind Teil der Grundlagen für die primären Analyseverfahren der diagnostischen Medizin. Mit diesen Eigenschaften können kurz- und langfristig Rückschlüsse auf das Wohlbefinden des Patienten geschlossen werden. Bei Bedarf können mit diesen Informationen weitere Analysen veranlasst oder notwendige Maßnahmen gegen ein vorzeitiges Ableben des Patienten durchgeführt werden.

Dabei sind folgende Eigenschaften des Blutes von besonderer Bedeutung für die analytische Medizin.

Blutdruck, welcher den Druck der Systole und die Diastole angibt

Hämoglobingehalt, welcher auch unter Sauerstoffsättigung bekannt ist

Flussgeschwindigkeit ist das Produkt aus Schlagvolumen und Herzfrequenz

Hämatokritwert bildet das Verhältnis von Erythrozyten zu dem gesamten Teilchen im Blut ab.

Dabei wird das mittleres corpusculäres Volumen der Erythrozyten (MCV) angegeben und man kann die Anzahl der Erythrozyten bestimmen.

Diese Eigenschaften können durch invasive¹ und nicht invasive Methoden ermittelt werden, wobei die nicht invasiven Methoden zu bevorzugen sind. Diese haben den Vorteil bei der Untersuchung

¹latein invadere *einfallen, eindringen*

Entwicklung eines digitalen standalone Ultraschall-Multigate-Doppler-Systems

keine Gewebsschäden zu verursachen.

Im Gegensatz dazu haben invasive Methoden mehrere Nachteile, da es sich meist um einen invasiven Eingriff mit den dadurch verursachten Gewebsschäden handelt, worunter auch die Blutentnahme zählt.

Ein detailliertes Blutbild kann zwar durch die Blutentnahme mit anschließender Laboranalyse ermittelt werden, jedoch bestehen Risiken, welche sich durch Infektionen oder Vertauschen der Messproben ergeben. Nicht zu vernachlässigen sind auch die erhöhten Zeit- und Kostenfaktoren welche in Notfallsituationen entscheidend sein können.

Abhilfe schaffen die nicht invasiven Methoden. Durch eine erhöhte Investition können Geräte erworben werden, welche den Blutdruck, den Hämoglobingehalt und die Flussgeschwindigkeit des Blutes nichtinvasiv messen können.

Bei Langzeitüberwachungen von Patienten durch einen langfristigen ambulanten Aufenthalt werden bis zu drei mal pro Woche Blutabnahmen durchgeführt. Die invasive Methode der Blutabnahme kann zu Anämien führen, da die Erythrozyten nur alle fünf bis neun Tage neu gebildet werden, was zur Reduzierung der Sauerstoffversorgung führen kann. Durch die Bestimmung des Hämatokritwertes kann entschieden werden, ob bei zu hohen Wert eine Kochsalzlösung und bei zu geringen Wert Sauerstoff verabreicht werden muss.

Die vorliegende Arbeit befasst sich aus diesem Grund mit der Umsetzung einer nicht invasiven Methode zur schnellen und eindeutigen Hämatokritwertbestimmung. [1] [2]

2.2. Aufgabenstellung

2.2.1. Aufgabenstellung der Bachelorarbeit

Das Ziel dieser Bachelorarbeit ist ein System für die Hämatokritwertbestimmung auf Basis der nichtinvasiven gepulsten Ultraschall-Dopplertechnologie (Abschnitt 3.2) zu realisieren. Eine Visualisierung in Spektrogramm und M-Mode sowie eine akustische Wiedergabe der Signale sind dabei unverzichtbar, um eine sichere Messung zu gewährleisten. Abb. 2.1 visualisiert dabei das grundlegende Vorhaben.

Diese Thesis soll eine Weiterentwicklung einer vorangegangen Arbeit (Abschnitt 2.2.2) vorantreiben. Dabei soll das Doppler System kompakt und somit flexibel einsetzbar sein. Beim Designen der Elektronik ist darauf zu achten, dass eine 2 MHz, 4 MHz sowie eine 8 MHz Ultraschallsonde verwendbar sein soll. Des Weiteren soll eine Optimierung und Separierung der bereits entwickelten Platine erfolgen, indem die Signalgenerierung und Messung durch ein zu entwickelndes Interface unterschiedlichen Mikrocontrollern zu Verfügung gestellt werden kann. Die Visualisierung der Messung soll dabei auf einen Display sowie auf dem PC dargestellt werden können. Ein benutzerfreundliches Graphical User Interface (GUI) darf dabei nicht vernachlässigt werden.

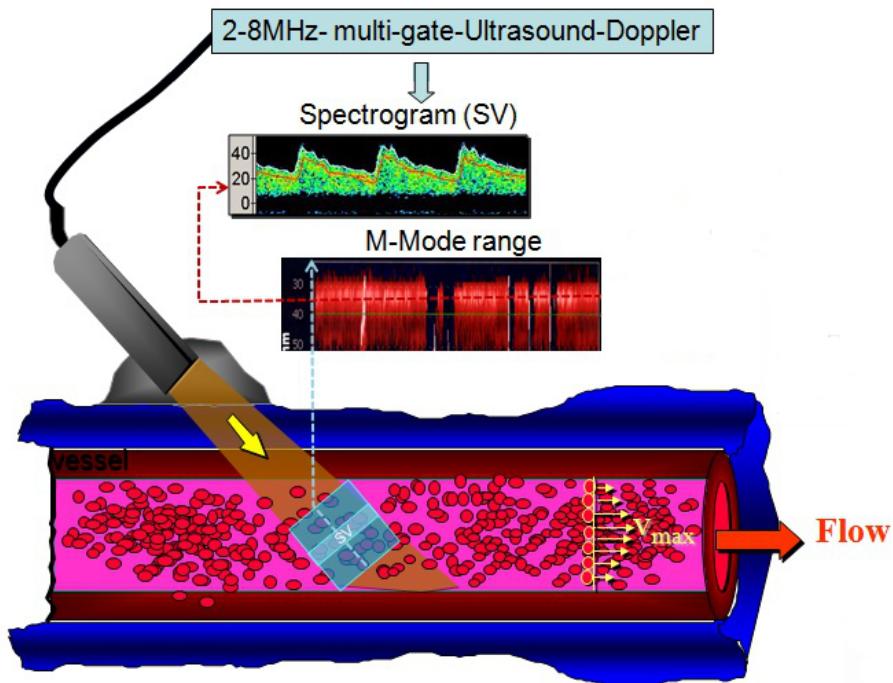


Abbildung 2.1.: Methodik: Multi-gate Ultraschall Doppler - Vereinfachte Darstellung

2.2.2. Vorstellung des vorhandenen Systems

Das digitale Ultraschall-Multigate-Doppler System

Herr Sebastian Stemplewitz entwickelte in seiner Bachelorarbeit an der Hochschule Ulm einen Prototypen für die PC gestützte Hämatokritwertmessung. Dies soll eine kostengünstige und intravasale Alternative zu derzeitigen Bestimmungen des Hämatokritwertes werden.

Das System ist soweit minimalisiert sowie miniaturisiert (Abb. 2.2), dass eine Weiterentwicklung des Systems die logische Schlussfolge war. Derzeit kann das System mit 8 MHz Sonden erfolgreich Messdaten aufnehmen und darstellen. Dabei dient ein PIC18F4550 uC der Firma Microchip als Schnittstelle zwischen der Messsteuerung und dem PC durch die Implementierung eines Universal Serial Bus (USB) Audio-Profil, welches mit der selbst entwickelten Software² kommuniziert. Jedoch wurde in der Testphase festgestellt, dass eine Optimierung der Transmitter- und Receiververschaltung notwendig ist um Artefakte zu reduzieren und die Messgenauigkeit zu erhöhen. Aus dem Wunsch, die Kompatibilität mit 2 und 4 MHz Ultraschallsonden zu gewährleisten, entstand die Idee dieses System mit einem ARM-Cortex uC zu betreiben und die Datenverarbeitung und Darstellung über diesen bereitzustellen. [3]

²auf Basis von C++ und QT

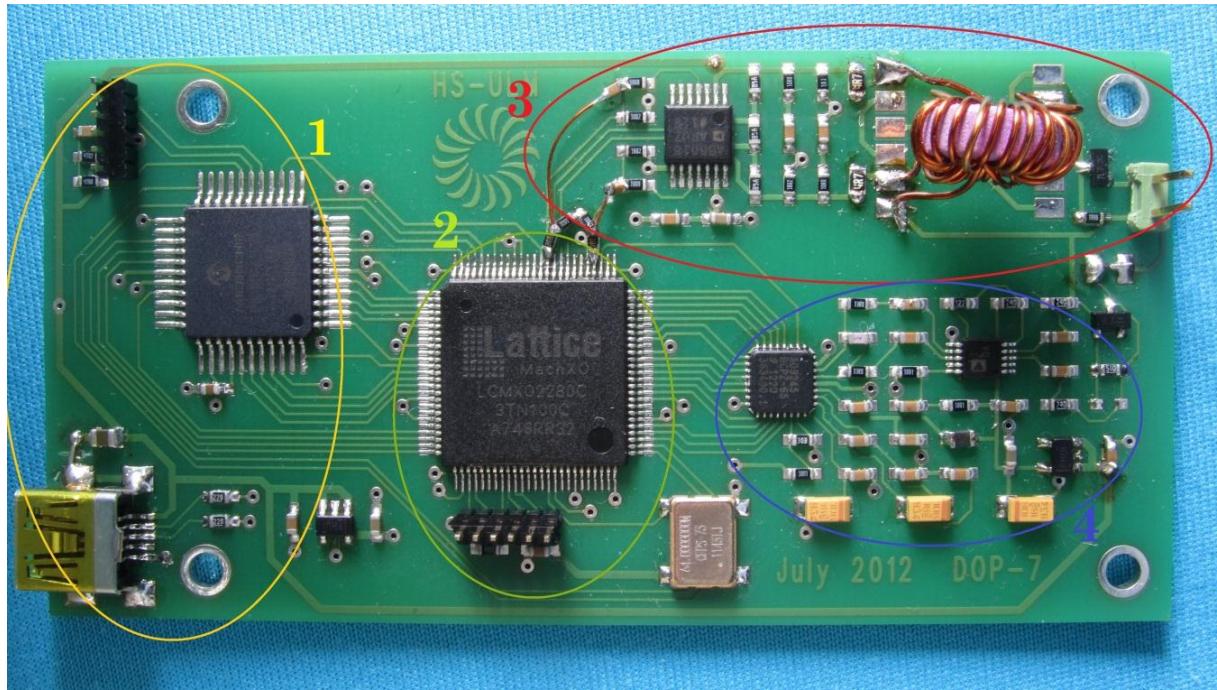


Abbildung 2.2.: Systemaufbau Thesis Stemplewitz

Tabelle 2.1.: Bestandteile der Platine

Nr.	Beschreibung
1	uC und USB-Kommunikation
2	Signalerzeugung und Demodulation CPLD
3	Transmitter mit Differenzverstärker und Übertrager
4	Receiver mit Vorverstärker und ADC

2.2.3. Anforderungen an das Messsystem

Es werden drei Ebenen von Anforderungen unterschieden: die Benutzeranforderungen, die Systemanforderungen und die Software- und Hardwareanforderungen.

Benutzer und deren Anforderungen werden dabei als höchste Ebene angesehen, da sie die Bedürfnisse des Kunden an das System darstellen. Darunter werden sowohl grundlegende Anforderungen, Basisanforderungen sowie Begeisterungsfaktoren beschrieben, welche entschlüsselt, gegliedert und anschließend strukturiert werden müssen.

Die grundlegenden Anforderungen, welche Richtlinien und Normen beinhalteten, müssen erfüllt sein. Basisanforderungen beschreiben die Funktionalität des Systems. Nur wenn grundlegende, sowie Basisanforderungen erfüllt sind, kann das System zertifiziert, auf den Markt gebracht und dem Kunden die Möglichkeit gewährt werden, das Produkt zu kaufen. Begeisterungsfaktoren sind Anforderungen, die aus Sicht des Kunden nicht unbedingt erfüllt sein müssen um das Produkt zu kaufen. Jedoch bewegen genau diese Faktoren den Kunden, das Produkt zu kaufen, da

Entwicklung eines digitalen standalone Ultraschall-Multigate-Doppler-Systems

es sich durch diese von Konkurrenzprodukten abheben kann. Benutzeranforderungen enthalten derweil keinerlei technische Vorgaben.

Systemanforderungen bilden die mittlere Anforderungsebene, welche die allgemeinen Leistungen des Produktes spezifiziert. Diese Leistungen dienen als Grundlage des Systemdesigns.

Die Software- und Hardwareanforderungen werden aus dem Systemdesign abgeleitet und detaillierter beschrieben.

Bindende und optionale Anforderungen können anhand der Kundenformulierung unterschieden werden. Dabei werden bindende Anforderungen durch "soll" oder "darf nicht" hervorgehoben. Normativ bindende Anforderungen werden durch "muss" Formulierungen dargestellt. Eine optionale Anforderung wird dabei durch "sollte" und eine Zusatzfunktion mit "kann" formuliert. Die in dieser Arbeit durchgeführte Optimierung wurde durch die folgenden Anforderungen detailliert definiert.

Benutzeranforderungen

- Das bestehende USD-System soll in eine Steuer- und eine Messplatine modularisiert werden.
- Ein 2,8 Zoll TFT-Display mit Touchfunktionalität soll integriert werden.
- Die Auswertung sowie die Steuerung der Messungen sollen auf dem TFT-Display lesbar dargestellt werden.
- Das System soll mit Ultraschallsonden im Frequenzbereich von 2, 4 und 8 MHz arbeiten.
- Für eine Pulse Repetition Frequency sollen bis zu 40 Werte (20 Real-, 20 Imaginärteil) zu Verfügung gestellt werden, um weitere Analysen zu ermöglichen.
- Das System soll weiterhin über die vorhandene PC Software betrieben werden können.

Systemanforderungen

- Das System soll über eine externe Spannungsversorgung betrieben werden.
- Das System soll als Steuereinheit einen ARM Cortex M3 haben.
- Das System soll einen funktionsfähigen Audioausgang besitzen.
- Das System sollte einen Composite-Out Ausgang besitzen.
- Das System soll einen USB-micro Anschluss haben.
- Die Messplatine soll einen Triggerausgang für die Zertifizierung von Ultraschallsonden bereitstellen.
- Die Messplatine soll einen separaten Ausgang sowie Eingang für Messungen bereitstellen, welche leicht miteinander verbunden werden können.

- Das System soll einen zusätzlichen Modus, zur Datenübertragung an einen PC, erhalten.

Software- und Hardwareanforderungen

• Energieversorgung

- Das System soll die Hardware des Energieversorgers vor Fehlfunktionen des Systems schützen.
- Das System sollte eine maximale Leistungsaufnahme von 2.5W bei 5V DC besitzen.
- Das System soll eine maximale Eingangsspannung von 5.5V aushalten.
- Das System soll aus der angelegten Spannung die für das System benötigten Spannungen erzeugen.

• Messelektronik

- Die Messelektronik soll eine Spannungsversorgung von 5V besitzen.
- Die Messelektronik soll separat aus einer Platine bestehen.
- Ein Trigger Signal sollte vorhanden sein, um das Zertifizierungsverfahren der Sonden und des Systems zu ermöglichen.
- Die Messelektronik sollte separierbare Masseflächen besitzen.
- Die Messelektronik sollte einen Schutz vor zu großen Signalamplituden besitzen.
- Die Messelektronik soll eine Pulse Repetition Frequency von 2 kHz bis zu 12 kHz und eine Abtastrate von 64 MHz besitzen.
- Die Platine soll eine steckbare Verbindung aufweisen.

• Auswerteelektronik

- Die Platine der Auswerteelektronik soll auf die Platine der Messelektronik aufsteckbar sein.
- Das Audiosignal sollte durch einen Verstärker unterstützt werden.

• Display

- Das Display soll eine gut lesbare Auflösung bieten.
- Das Display sollte eine Touchfunktion vorweisen.
- Das Display soll eine Mindestgröße von 2 Zoll besitzen.
- Das Display sollte auf die Platine der Auswerteelektronik aufsteckbar sein.

• GUI

- Das GUI soll ein Menu bereitstellen, um die Parameter der Messung einzustellen.
- Das GUI soll ein linearen, für die Messwerte und ein Fast Fourier Transformation (FFT)-Graphen, für das Zeit-Schallsignal bereitstellen.

2.3. Beschreibung der Methodik

Die Ausgangssituation wurde analysiert und mit dem aktuellen Stand der Forschung verglichen, indem die zur Verfügung stehenden Unterlagen studiert und mit aktueller Fachliteratur abgeglichen wurden. Durch die Recherchen des aktuellen Stands der Technik zeigte sich entsprechendes Optimierungspotenzial im Vergleich zur vorangegangenen Arbeit. Insbesondere zeigte sich, dass eine deutliche Verbesserung der Nutzbarkeit erreicht werden kann, wenn das System nicht nur stationär, sondern auch mobil eingesetzt werden kann. Des Weiteren ergab sich Optimierungspotential im Nutzungskomfort. Durch die Anpassung der Hard- und Software und der Verlegung von Steuerungs- und Kontrollfunktionen vom PC, der an das Messsystem angeschlossen ist, direkt in den mobilen Teil des Messsystems, wurden im Rahmen der hier beschriebenen Studienarbeit Verbesserungen im Hinblick auf den mobilen Einsatz und des Nutzungskomforts des Messsystems erreicht.

3. Grundlagen

3.1. Ultraschall

3.1.1. Definition

Schallwellen sind mechanische Wellen. Dabei unterscheidet man Infraschall¹, Ultraschall² und Schallwellen, welche das menschliche Gehör wahrnehmen kann³. Die höchsten, technisch reali-

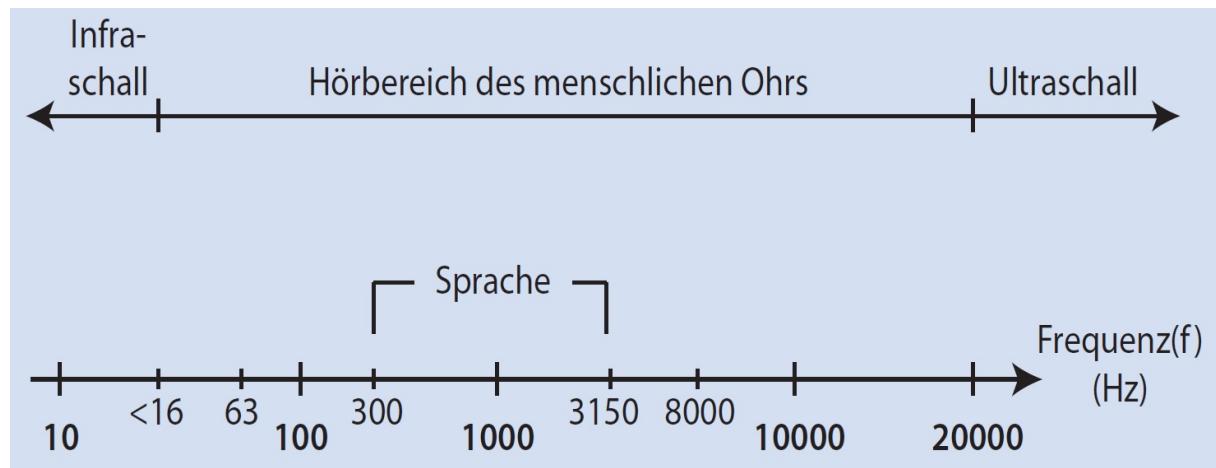


Abbildung 3.1.: Frequenzbereiche Schall

sierbaren Schallfrequenzen liegen bei ca. 1 Giga Herz (GHz). Für die medizinische Diagnostik sind dabei Frequenzen im Bereich von 2 bis 10 Mega Herz (MHz) interessant. Unterhalb von 2 MHz ist die Auflösung zu gering und oberhalb von ca. 10 MHz ist die Absorption im Gewebe zu stark.

In menschlichem Gewebe beträgt die Schallgeschwindigkeit c etwa 1500 m/s. Bei Frequenzen im Bereich von 2 bis 10 MHz ist deshalb die Wellenlänge λ im Bereich von < 10 mm. Somit kann man erreichen, dass sich Ultraschall im Gewebe wie ein optischer Strahl ausbreitet. Er kann fokussiert, reflektiert, gestreut und absorbiert werden. Durch diese Effekte kann somit eine Abbildung von Organen erzielt werden, welche die Basis der Sonographie oder Ultraschalldiagnostik bildet. [4] [5] [6]

¹Frequenzen < 20 Hz

²Frequenzen > 20.000 Hz beziehungsweise (bzw.) 20 kHz

³Frequenzen von 20 bis 20.000 Hz

3.1.2. Erzeugung und Empfang

Im Jahr 1880 wurde der *Piezoelektrische Effekt* von Pierre Curie entdeckt. Der Effekt bezieht sich auf Materialien, welche einen permanenten elektrischen Dipolmoment⁴ aufweisen. Diese erzeugen eine Spannung, wenn eine Kraft (respektive (resp.) ein Druck) angelegt wird. Mit diesem Effekt ist es möglich Kräfte, jedoch aber auch Torsion oder wie in dieser Arbeit Ultraschall zu messen. Dieser Effekt kann aber auch zur Erzeugung von Kraft in Form von

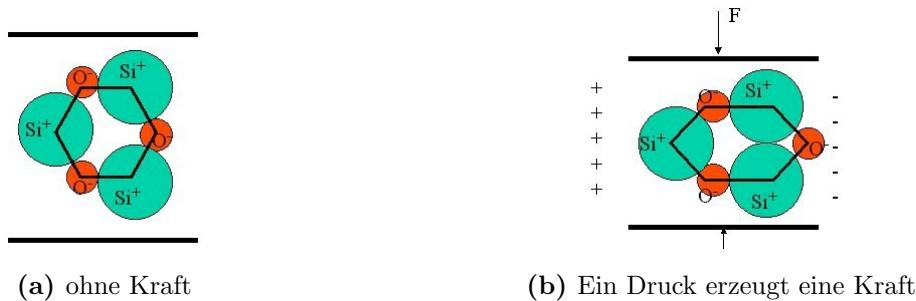


Abbildung 3.2.: Piezoelektrischer Effekt

mechanischen Wellen genutzt werden. Dies bezeichnet man als den *Indirekten piezoelektrischen Effekt*. Durch Anlegen einer Wechselspannung an einen elastischen Körper wird dieser mit der Frequenz der Wechselspannung verformt und erzeugt in Abhängigkeit der Körpereigenschaften, der Amplitude der angelegten Wechselspannung und deren Frequenz Schallwellen. [4] [5] [6]

3.1.3. Ausbreitung

Eine Schallwelle entspricht einer zeitlichen und räumlichen periodischen Auslenkung von Druck und Dichte des Mediums. Dabei interessieren vor allem die Änderungen des Drucks (nicht der Mittelwert). Somit sind Schallwellen an Materie gebunden und können sich im Vakuum nicht ausbreiten. In Luft, Flüssigkeiten sowie biologischen Gewebe breiten sich Schallwellen dabei in Form von Longitudinalwellen⁵ aus.

Dabei hängt die Schallgeschwindigkeit c in Festkörpern von der Dichte ρ , der Poissonzahl μ und dem Elastizitätsmodul E ab. Es ist dabei die Schallgeschwindigkeit c im Festkörper:

$$c_{\text{longitudinal}} = \sqrt{\frac{E(1-\mu)}{\rho(1-\mu-2\mu^2)}} \quad (3.1.1)$$

$$c_{\text{transversal}} = \sqrt{\frac{E}{2\rho(1+\mu)}} \quad (3.1.2)$$

Mit der Gleichung 3.1.1 werden für die Medizin wichtigen Schallgeschwindigkeiten für die aufgeführten Materialien bestimmt (siehe Tabelle 3.1). Der Schalldruck und die Schallimpedanz

⁴Materialien, bei denen die Schwerpunkte der positiven und negativen Ladungen nicht zusammenfallen

⁵von Zonen mit Über- und Unterdruck (Verdichtungs- und Verdünnungszonen)

Entwicklung eines digitalen standalone Ultraschall-Multigate-Doppler-Systems

darf dabei nicht vernachlässigt werden, da die Ausbreitung von Druck und Dichte des Mediums abhängig ist.

Für die Schallimpedanz Z findet man:

$$Z = \frac{\Delta\rho_0}{v_0} = \sqrt{K * \rho_0} \quad (3.1.3)$$

$$Z = c * \rho \quad (3.1.4)$$

Aus den Formeln 3.1.3 und 3.1.4 ist erkennbar, dass die Schallimpedanz Z eine Materialkonstante ist.

Die Intensität einer Schallschwelle⁶ ist

$$j = \frac{Kraft * Weg}{Fläche * Zeit} = \rho * c \quad (3.1.5)$$

Mit der kinetischen Energiedichte ρv^2 kann die Schallintensität auch ausgedrückt werden durch

$$j = \rho c v^2 \quad (3.1.6)$$

In Bezug auf die Zeit ist sie damit

$$j = \rho_0 c A_0^2 \omega^2 \sin^2(\omega t) \quad (3.1.7)$$

wobei die Amplitude A_0 die Auslenkung $A_0 \sin(\omega t)$ darstellt. [4] [5] [6]

3.1.4. Reflexion und Brechung

Wie alle Wellen werden auch Schallwellen an Grenzflächen teilweise reflektiert. Diese Grenzflächen befinden sich zwischen Gebieten mit unterschiedlicher Schallimpedanz. Bei senkrechtem Einfall im linearen Bereich gilt für die transmittierte Intensität I_t mit der emittierten Intensität I_e

$$\frac{I_t}{I_e} = 4 \frac{Z_1 Z_2}{(Z_1 + Z_2)^2} \quad (3.1.8)$$

und für die Intensität der reflektierten Welle I_r

$$\frac{I_r}{I_e} = \frac{(Z_1 - Z_2)^2}{(Z_1 + Z_2)^2} \quad (3.1.9)$$

Da in der Sonographie hauptsächlich mit den reflektierten Wellen gearbeitet wird, beziehen sich die nächsten Aussagen auf die Formel 3.1.9. Unter Betrachtung der Tabelle 3.1 und der Formel 3.1.4 erschließt sich, dass die Schallimpedanzen der biologischen Materialien sehr gering sind. Somit kann Formel 3.1.9 weiter vereinfacht und der Reflektionsfaktor R bestimmt werden

$$R \approx \frac{(\Delta Z)^2}{4Z^2} \quad (3.1.10)$$

Hingegen ist der Reflektionsfaktor zwischen Luft und dem biologischen Gewebe extrem groß und reduziert die Intensität dementsprechend stark. Diesem Effekt muss durch ein spezielles Kontaktgel entgegengewirkt werden. [4] [5] [6]

⁶Energietransport pro Fläche und Zeiteinheit

3.1.5. Absorption und Streuung

Wie in Kapitel 3.1.4 dargestellt nimmt die Gesamtintensität der Ultraschallwelle bei jeder Grenzfläche ab. Zudem wird das Medium durch die einstrahlende Welle in Schwingung versetzt und strahlt somit selbst eine Welle aus.

Findet diese Schwingung in Phase mit der einfallenden Welle statt⁷, beeinflusst die Interferenz zwischen den Wellen lediglich die Phasengeschwindigkeit der Ultraschallwelle.

Eine Überlagerung der Wellen hingegen⁸ führt zu einer Schallabstrahlung in alle Richtungen und somit zu einer Streuung, was eine Dämpfung zu Folge hat.

Da das biologische Gewebe jedoch eher einem homogenen Medium entspricht, ist die Abnahme für eine ebene Welle proportional zur Intensität und nimmt somit exponentiell ab

$$I(x) = I_0 e^{-\mu x} \quad (3.1.11)$$

Dabei besteht der Dämpfungs- oder Schwächungskoeffizient μ aus einem Absorptions- und einem Streuanteil $\mu = \mu_{Abs} + \mu_{Streu}$. Im Gewebe beträgt die Dämpfung ca. $1 \frac{dB}{cm \text{ MHz}}$. Die Effizienz der Streuung hängt von der Frequenz / Wellenlänge λ , der Größe der streuenden Inhomogenitäten und dem Unterschied der Schallimpedanz ab.

Streuung und Absorption bestimmen zusammen die Eindringtiefe der Schallwellen. Somit ist die Dämpfung abhängig von dem **Koeffizienten** μ , dem **Weg** x und der **Sendefrequenz** f . [4] [5] [6]

3.1.6. Dopplereffekt

Der Effekt tritt bei allen mechanischen Wellen auf, die sich durch den Raum bewegen. Dabei erregt die Welle stationäre und sich bewegende Teilchen gleichermaßen, wodurch eine weitere Welle durch das erregte Teilchen ausgesendet wird. Bei stationären Teilchen wird die Trägerfrequenz f_0 reflektiert. Die bewegenden Teilchen jedoch führen je nach Bewegungsrichtung zur Welle kinetische Energie zu der Reflektion der Welle hinzu oder ab⁹. Bewegt sich ein Teilchen entgegen der Longitudinalwellenrichtung so wird die reflektierte Wellenlänge λ größer. Umgekehrt wird λ kleiner, wenn sich ein Teilchen mit der Longitudinalwellenrichtung bewegt.

Die Differenz Δf zwischen emittierter und empfangener Trägerfrequenz nennt man Dopplerschiebefrequenz. Die Differenz ist abhängig von der Trägerfrequenz f und dem Geschwindigkeitsvektor \vec{v} des bewegten Teilchens. Somit ist der Winkel θ zwischen Ausbreitungsrichtung der Welle und des Richtungsvektors des Teilchens nicht vernachlässigbar.

Die Dopplerschiebefrequenz berechnet sich nach folgender Formel

$$\Delta f = \frac{2fv \cos(\theta)}{c} \quad (3.1.12)$$

⁷homogenes Medium

⁸in inhomogenen Medium

⁹Teilchen können beschleunigt oder abgebremst werden

Anwendung findet dieser Effekt nicht nur in der diagnostischen Medizin zur Bestimmung von Blutströmungsgeschwindigkeiten. Der Effekt dient seit Jahren in der Industrie und in Haushalten zur Überwachung des Durchflussvolumens und der Erkennung von Fremdkörpern in Flüssigkeiten. Die Dopplerschiebefrequenzen liegen dabei in der diagnostischen Medizin im hörbaren Bereich von einigen kHz.

3.2. Pulsed Wave Dopplerverfahren

Beim gepulsten Dopplerverfahren werden Bursts¹⁰ durch den Transduktorkombination mit einem Piezoelement erzeugt. Diese werden in periodischen Abständen (Pulse Repetition Frequency (PRF)) in das zu messende Material transmittiert. Dabei werden durch Teilchen oder Dichteänderungen Reflexionen verursacht (3.1.4), welche mit demselben Piezoelement oder einen in Ausbreitungsrichtung liegenden Piezoelement erfasst und anhand der Laufzeit (3.1.3) bestimmten Materialtiefen zugeordnet werden kann.

Statische Reflexionen verursachen dabei stärkere Signale als die dynamischen Dopplersignale und müssen aus der Messung gefiltert werden. Dies geschieht durch die sogenannte Demodulation.

Abb. 3.3 visualisiert den Ereignisablauf für zwei überlagerte Messtiefen in Abhängigkeit der Peripheriesteuerung. Dabei wird die PRF durch den Impuls *Retransmit* realisiert, welches die Erzeugung des Burstsignals zur Folge hat. Anschließend werden durch die Zeitdifferenzen, die zu messenden Tiefen digitalisiert sowie demoduliert.

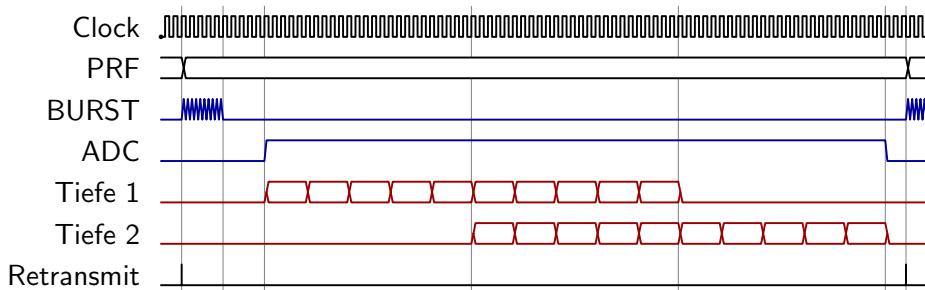


Abbildung 3.3.: Ereignis-Zeitdiagramm

Durch Multiplikation mit der Sendefrequenz f_0 in Kombination mit einem Tiefpassfilter werden statische Reflexionen in Gleichanteile umgewandelt und die dynamischen Dopplerschiebefrequenzen in den Niederfrequenz (NF) Bereich transformiert. Durch den Tiefpassfilter werden nachfolgend positive mit negativen Gleichanteilen addiert, was eine Eliminierung der statischen Reflexionen zur Folge hat.

¹⁰kurze Signale

Tabelle 3.1.: Schallgeschwindigkeiten in unterschiedlichen Materialien und Geweben

Material	c [$\frac{m}{s}$]
Luft	340
Fett	1400
Wasser ($37^{\circ}C$)	1540
Leber	1549
Niere	1561
Muskel	1568
Blut	1570
Knochen	3600

Tabelle 3.2.: Eindringtiefen als Funktion der Frequenz in menschlichen Geweben

Frequenz f in MHz	Eindringtiefe in cm	Anwendung
1	50	
3.5	15	Fötus, Leber, Herz, Niere
5	10	Gehirn
7.5	7	Prostata
10	5	Pankreas
20	1.2	Auge, Haut
40	0.6	Intravaskulär

4. Stand der Technik

4.1. Auswertung der Ultraschallmessung und Bildentstehung

A-Mode auch "Amplitudenmodus" genannt, ist die erste Darstellungsform in der Sonographie und die einfachste Umsetzung des Impuls-Echo-Prinzips. Es ist eine eindimensionale Abbildung der reflektierten Schallwellen in einem Diagramm und stellt die empfangen Echos in Abhängigkeit von der Tiefe dar, wie in Abb. 4.1a zu sehen ist.

B-Mode auch "Brightness-Mode" genannt, stellt die Echos nicht als Ausschläge, sondern als Bildpunkte mit unterschiedlicher Helligkeit auf dem Bildschirm dar. Dabei entspricht jede Amplitude einem Helligkeits- bzw. Grauwertbild und ist abhängig von der Intensität der elektrischen Signale¹. Bei modernen Ultraschallgeräten sind 256 Grauwerte zwischen schwarz und weiß möglich. Ein schwarzes Bild wird dabei durch zu geringe Schallintensität erzeugt, welches die Folge von Totalreflexion oder fehlenden Impedanzunterschied² ist.

M-Mode auch "Motion-Mode" genannt, stellt Gewebestrukturen an einem bestimmten Ort als Funktion der Zeit dar. Dabei werden die Amplituden der Ultraschallechos wie im B-Mode [4.1] aber zu einem bestimmten Zeitpunkt dargestellt. Über ein Ort-Zeit-Diagramm werden örtliche Veränderungen echogener Strukturen über die Zeit dargestellt³, wie in Abb. 4.1b zu sehen ist. Dabei wird die Amplitude auf der vertikalen Achse und die von den wiederholten Impulsen erzeugten Echos auf der horizontalen Achse (Zeitachse) abgetragen.

Doppler Spektrogramm ist eine Darstellung, wobei auf der X-Achse die Zeit und auf der Y-Achse die Frequenzverteilung dargestellt sind. Da die dargestellten Frequenzen abhängig der Fließgeschwindigkeit der Blutteilchen sind, können Aussagen über die Durchschnittsgeschwindigkeit des Blutes, sowie über Schlagvolumen und Herzfrequenz gemacht werden (visualisiert in Abb. 4.1b). Dies ist mithilfe der Kurzzeit-Fourier-Transformation (eng: short-time Fourier transform, STFT) realisierbar, in der kurze Zeitabschnitte in den Spektralbereich überführt werden.

¹je stärker das Echo, desto heller der Bildpunkt

²keine Reflexion möglich

³Time-Motion Verfahren

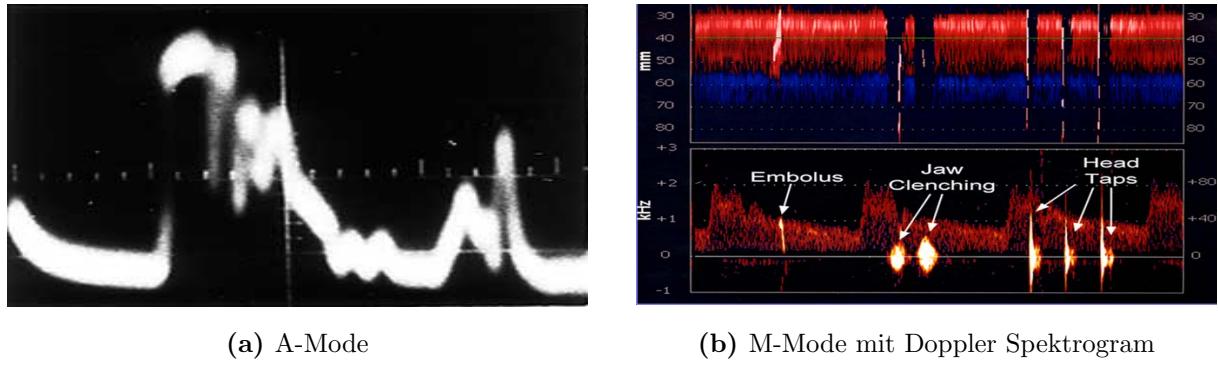


Abbildung 4.1.: Modes der Bildentstehung

4.2. Touchpanel Technologien

4.2.1. Definition

Welche Technik auch benutzt wird, das Verfahren der Berührungserkennung funktioniert immer auf die selbe Weise. Zunächst erfasst die Elektronik die Position der Berührung auf dem Bildschirm und übermittelt diese Informationen an die Auswertungseinheit. Diese vergleicht die Information mit den Positionen der angezeigten Bedienelemente. Wenn die Werte übereinstimmen wird eine entsprechende Funktion im Programm aufgerufen und abgearbeitet. Wenn ein Touchscreen mehrere Berührungen der Oberfläche gleichzeitig aufnehmen kann spricht man von der Multitouch-Technik. Diese Fähigkeit kann bisher nur von kapazitiven und optischen Systemen verarbeitet werden.

Vorteile der Touch-Technologien liegen dabei in der Benutzerfreundlichkeit und der Reduzierung von Hardwarekomponenten. Eine einfachere Bedienung ist vor allem bei kleineren Geräten zu erkennen. Diese können keine ausreichende Fläche für eine Tastatur bereitstellen. Ältere Mobiltelefone z.B. kompensierten dies durch mechanische Slide-Funktionen⁴ mit erhöhten Materialkosten. Auch die Menüführung hat sich mit den Touchscreens intuitiver entwickelt. Im Vergleich zu herkömmlichen Displays und Monitoren sind die Kosten für einen Touchscreen durchaus höher. Jedoch kosteten Tastaturen zum Aufschieben oder Ausklappen durch ihre komplexere Mechanik mehr, sodass sich Touchscreens durchsetzen konnten. Des Weiteren ermöglicht es den Herstellern, die Fläche des Displays zu vergrößern, da Informationen und Eingabefunktionen kombiniert auf dem Display dargestellt werden können. [7] [8]

4.2.2. Resistiv

Ein resistives Touchpanel, auch Widerstands-Touchpanel genannt, ist eine ältere Technik, welche auf Druck reagiert und am preisgünstigsten zu beziehen ist. Es besteht aus zwei gegenüberliegenden leitfähigen ITO-Schichten welche auf einer Polyesterschicht und einer Kunststoff-

⁴Aufschieben oder Ausklappen der Tastatur

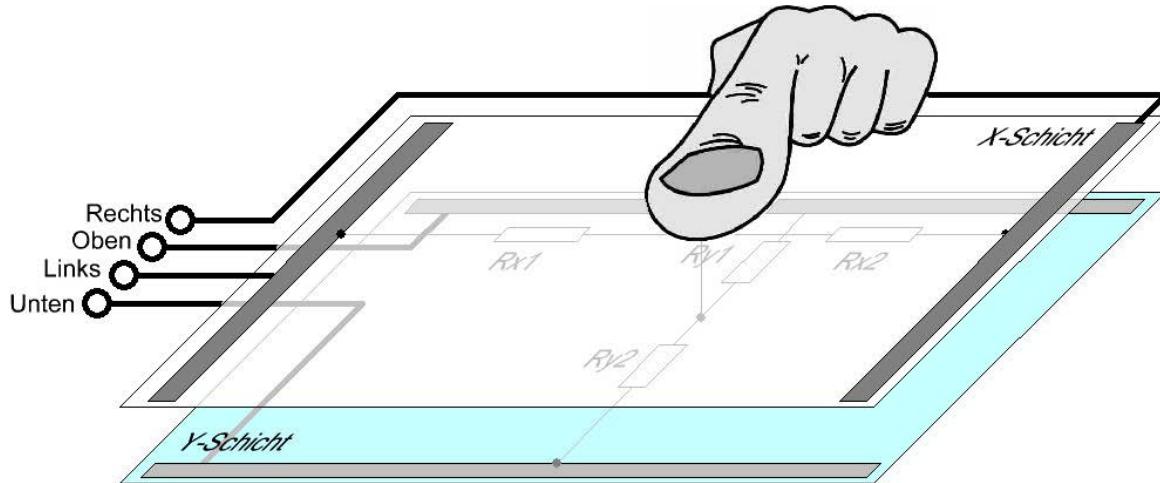


Abbildung 4.2.: Aufbau eines resistiven Touchpanels

oder Glasschicht aufgebracht sind. Dabei werden diese durch kleine Abstandshalter voneinander getrennt. Der Touchscreen verfügt über Messleitungen⁵, die bei einer Berührung mit dem Finger oder Stift über den Berührungsplatz eine Spannung leiten.

Gemessen in horizontaler und vertikaler Richtung, wird durch Interpolation die Position der Betätigung bestimmt. Dabei werden lediglich zwei ADC (mit Tristate mode) und zwei digitale Tristate IO's benötigt. [7] [9] [8]

4.2.3. Induktiv

Die Technik der induktiven Touchscreens wird primär in Grafiktablets eingesetzt. Sie basiert auf den physikalischen Grundlagen der elektromagnetischen Wellen in Kombination mit Spule und Kondensatoren. Dabei dient ein spezieller Eingabestift mit einer integrierten Spule und Kondensatoren in der Spitze als Eingabegerät. Die Technik zeichnet sich durch eine sehr hohe Genauigkeit aus. Nachteilig ist jedoch, dass nur mit den speziellen Eingabestift gearbeitet werden kann. [7] [8]

4.2.4. Kapazitiv

Ein kapazitiver Touchscreen besteht aus einer Glasscheibe, welche wie ein resistives Touchpanel mit ITO beschichtet wird. Somit kann auf der Oberfläche der Glasscheibe ein elektrisches Feld generiert werden. Dabei wird an jeder Ecke der Glasscheibe eine Spannung angelegt, welche sich gleichmäßig konstant über das Glas verteilt. Eine Änderung des Feldes wird dabei durch Berührung eines Fingers erzeugt. Somit wird das elektrische Feld verändert, da sich das Spannungsverhältnis zwischen den Seiten teilt und ein Ladungsfluss durch den Finger entsteht. Die

⁵4, 5 oder auch 8 Leiter Technik

Entwicklung eines digitalen standalone Ultraschall-Multigate-Doppler-Systems

auftretende Differenz wird gemessen und von einen uC ausgewertet. [7] [9] [8]

4.2.5. Akustisch

Auch SAW-Touch genannt, hat an den Seitenflächen des Glassubstrates Signalgeber montiert. Diese Geber senden Ultraschallwellen in horizontaler und vertikaler Richtung aus. Dabei werden die Signale von der Oberfläche reflektiert und auf der gegenüberliegenden Seite detektiert. Durch Berührung mittels eines Fingers oder Gegenstandes wird ein Teil der Wellen absorbiert und die Auswerteeinheit kann durch Messung der Amplituden die Positionsbestimmung durchführen. [8]

4.2.6. Optisch

Optische Touchscreens sind kostenaufwendig und arbeiten auf Basis von Infrarot-LEDs. Dafür kann diese Technik auf sehr großen Flächen bereitgestellt werden. An zwei der Ränder des Bildschirms werden dabei eine Reihe von Infrarotleuchtdioden angebracht, welche ein Gitter aus Lichtstrahlen hervorbringen. Auf den gegenüberliegenden Rändern des Bildschirms werden Fotodetektoren angebracht, welche das Infrarotlicht empfangen. Wenn ein Gegenstand in dieses Gitternetz gebracht wird, werden an dieser Position die X- und Y-Richtung der Strahlengänge unterbrochen. Dies kann von einen uC verarbeitet werden, wodurch die Position des Gegenstandes berechnet werden kann. [9] [10] [8]

5. Material und Methode

5.1. Material

5.1.1. Verwendete Programme

Eagle

Cadsoft Eagle ist eine PCB Design Software, welche für Microsoft[©] Windows[®], Linux Distributionen sowie für Apple OS X verfügbar ist und aktuell die Versionsnummer 6.5.0 trägt. Diese Software wurde für die Erstellung eines Schaltplans, sowie für das Erstellen eines Leiterplattenlayout genutzt, was sich durch die Ableitung des zuvor erstellten Schaltplans schnell realisieren lässt. Dazu stehen der Schaltplan-Editor und der Layout-Editor zur Verfügung, welche sich gegenseitig synchronisieren und Änderungen in dem jeweils anderen Editor sichtbar machen. Die Überprüfung des Leiterplattenlayout durch "Design Rules", welche man bei den gewünschten Hersteller der Platinen beziehen kann, wird bereitgestellt, wodurch die Entwicklungszeit und Fehler reduziert werden. [11]

Diamond

Diamond[®] ist eine Design Software der Firma Lattice[©], welche für Microsoft[©] Windows[®] und Linux Distributionen verfügbar ist. Diese wurde für die Erstellung von Logikmodulen (in der Programmiersprache Verilog 2.0) und die Verknüpfung der erstellten Module in einem System Design verwendet. Dieses Design konnte durch den integrierten Compiler für die Programmierung des Lattice MachXOTM bereitgestellt sowie übertragen werden.

Active-HDL

Active-HDLTM ist eine Design Erstellungs- und Simulations-IDE für Field Programmable Gate Array (FPGA)'s der Firma Aldec, welche für Microsoft[©] Windows[®] verfügbar ist. Diese Software ist in Lattice Diamond[®] integriert und wurde für die Simulation bzw. Synthese der Logikmodule und des System Designs des MachXOTM genutzt.

Visual Studio Ultimate

Visual Studio[®] ist eine Software Entwicklungs IDE der Firma Microsoft[©], welche unter Microsoft[©] Windows[®] verfügbar ist. Diese IDE wird für die Code Dokumentation verwendet und bietet zusätzlich sehr gute Code Analyse Tools für die Programmiersprachen C, C++ und alle .NET Sprachen.

NXP Semiconductors LPCXpresso

LPCXpresso ist wie Visual Studio® eine IDE, jedoch auf der Basis von Eclipse. Sie wurde verwendet, um die geschriebene Firmware des ARM-Cortex zu debuggen. [12]

Texmaker

Bei Texmaker handelt es sich um ein Textsatzprogramm, welches von Pascal Brachet© unter Verwendung von Qt 5.1.0 sowie Poppler 0.22.5 compiliert wurde und für Microsoft© Windows®, Linux Distributionen sowie für Apple OS X verfügbar ist. Die aktuelle Version, welche verwendet wurde trägt die Versionsnummer 4.1.

GIMP

Bei GIMP (GNU Image Manipulation Program) handelt es sich um ein kostenloses und freies Bildbearbeitungsprogramm. Es ist für Microsoft© Windows®, Linux und Unix Distributionen sowie für Apple OS X und AmigaOS 4 verfügbar. Die aktuelle Version, welche unter Microsoft© Windows® verwendet wurde trägt die Nummer 2.8.10. [13]

5.1.2. Verwendete Geräte und Evalkits

mbed LPC1768 / OM11043 von NXP

Das OM11043 von NXP ist ein Prototyping Board, das eine komplette Abstrahierung der low-level uC Befehle durch einen Online-Compiler mit sich bringt. Das Board besteht aus 2 Komponenten. Den eigentlichen Cortex M3 Chip LPC1768 der Firma NXP und einen Debugger Chip. Dazu gehören Funktionen wie SPI, I2C, UART, CAN, GPIO, PWM, ADC, Digital-Analog Converter (DAC), sowie Ethernet, USB OTG.

Durch den HAL in Kombination mit dem ARM Cortex Microcontroller Software Interface Standard (CMSIS) kann der entwickelte Programmcode nicht nur schneller, sondern auch auf alle Cortex Modelle von NXP übertragen werden¹. [14]

Funktionsgenerator PM5139 von Philips (Flunke Corporation)

Für die Evaluierung der Messplatine wurde dieser Funktionsgenerator vorgesehen, welcher Testsignale erzeugt. Dieser unterstützt die Signalformen Gleichstrom, Sinus, Dreieck, Quadrat, Puls und Sägezahn und kann diese Signalformen mit einer Wiederholfrequenz von 0.1 mHz bis 20 MHz mit einer maximalen Spitzenspannung von 20 V generieren. Die Signale werden mit einer Präzision von 10-Bit generiert. [15]

¹sofern diese die programmierten Funktionen unterstützen

Digital Oscilloscope HMO3524 von HAMEG Instruments

Mit dem Oszilloskop von HAMEG Instruments werden Signale graphisch dargestellt. Dieses Gerät bietet eine Abtastrate von 4×2 GigaSamples/Sekunde und visualisiert Signale bis 350 MHz. Die Vertikale Auflösung beträgt 8 Bit und im HighResolution Mode bis zu 10 Bit. Es besitzt 2 MB internen Speicher und zählt zu den Speicheroszilloskopen. Es bietet auch Funktionen im Bereich der Mathematik an, wie z.B. eine FFT Analyse für die Darstellung der gemessenen Signalfrequenzen. Für die Dokumentation wird das Speichern von Bildern auf einen USB-Stick oder über die mitgelieferte Software auf dem PC bereitgestellt. [16]

5.2. Methode

Für das generieren des M-Mode (4.1) und des Doppler Spektrogramm (4.1) Graphen aus den digitalisierten Signalen der Ultraschallsonde muss eine Verarbeitung des Signales erfolgen. Zunächst muss das Signal demoduliert und in den NF Bereich transformiert werden, anschließend findet ein Datentransfer statt, welcher durch einen Streaming Serial Port (SSP) stattfindet. Nachdem die demodulierten Daten an den ARM-Cortex M3 transferiert wurden, muss eine FFT zur Darstellung des Doppler Spektrogramm als Audiosignal sowie als Graph erfolgen. In den folgenden Abschnitten werden die dafür benötigten Methoden detaillierter beschrieben, wobei sich für den SSP das SPI anbot.

5.2.1. Quadraturdemodulation

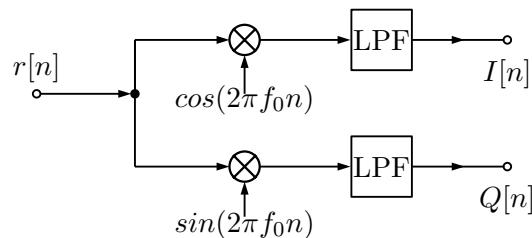


Abbildung 5.1.: Blockdiagramm Quadraturdemodulation und Tiefpass

Die Quadraturdemodulation ist eine Möglichkeit, Hochfrequente Eingangssignale in den Niedrfrequenzbereich umzuwandeln. Dabei wird das Eingangssignal $r[n]$ mit der Trägerfrequenz f_0 multipliziert.

Für die Digitale Umsetzung der Demodulierung benötigt man eine Look-up Table (LUT) mit Sinus- und oder Cousinuswerten sowie 2 Multiplizierer und 2 Addierer. Letztere müssen in doppelter Ausführung vorliegen, da je eine Multiplikation und Addition für den reellen Anteil und imaginären Anteil durchgeführt werden muss (siehe Abb. 5.1). Die LUT rotiert dabei mit der

Trägerfrequenz f_0 . Die Bestimmung der LUT Stützstellen N_S errechnet sich wie folgt:

$$N_S = \frac{f_{Sampling}}{f_0} \text{ mit } N_S \in \mathbb{N} \quad (5.2.1)$$

Es ist zu Beachten, dass die Anzahl der Stützstellen eine Ganze Zahl \mathbb{N} ist. Somit sollte die Samplingfrequenz $f_{Sampling}$ ein Vielfaches der Trägerfrequenz f_0 darstellen um Fehler der Berechnung zu vermeiden.

Durch die nachgestellte Addition des Realteils und des Imaginärteils werden die Summenfrequenzen der Multiplikation eliminiert. Diese Technik wird auch als FIR oder moving average Filter bezeichnet.

Realisierung Digitaler Tiefpass

Ein Digitaler Tiefpass wurde in den vorhandenen System von Herrn Stemplewitz durch sogenannte Microgates mit einer Periode von $1\mu s$ realisiert. [3] Basis dieser Methode ist die FIR Filter Technik. Dabei werden Messwerte eines definierten Zeitbereichs addiert, wodurch sich die positiven mit den negativen Werten eliminieren. Man spricht bei dieser Technik auch von moving average (Gleitender Mittelwert), da die Grundfrequenz f_0 eliminiert wird, wobei der NF Anteil vorhanden bleibt.

5.2.2. Serial Peripheral Interface

Das SPI ist ein Bus-System, welches von Motorola entwickelt wurde. Es besteht aus drei oder vier Signalleitungen und dient der seriellen, synchronen Datenübertragung zwischen einem Master und einem Slave (3-Wire Mode) oder einem Master und mehreren Slaves (3-/4-Wire Mode).

Die Signalleitungen des SPI setzen sich dabei wie folgt zusammen:

SCK (Serial Clock) ist das Taktsignal, welches vom Master bereitgestellt wird.

MOSI (Master Out, Slave In) dient dem Master für die Datenweitergabe an den Slave.

MISO (Master In, Slave Out) dient dem Master für den Empfang der Daten des selektierten Slaves.

CS (Chip Select) dient dem Selektieren des Slaves und ist low Aktiv.

Der Vorteil des SPI ist sein Vollduplex Modus, welcher durch Schieberegister in Slave und Master realisiert wird (siehe Abb. 5.2). Für die Datenübertragung ergeben sich 4 verschiedene Modi, welche sich durch einstellen der *Clock Polarity (CPOL)* und der *Clock Phase (CPHA)* ergeben (siehe Tab. 5.1).

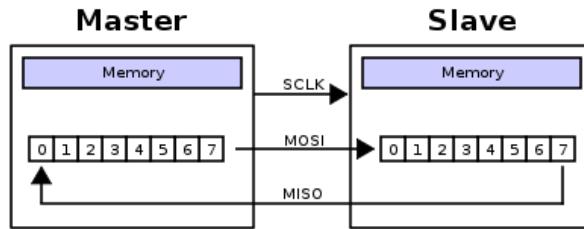


Abbildung 5.2.: SPI Schieberegister Master-Slave

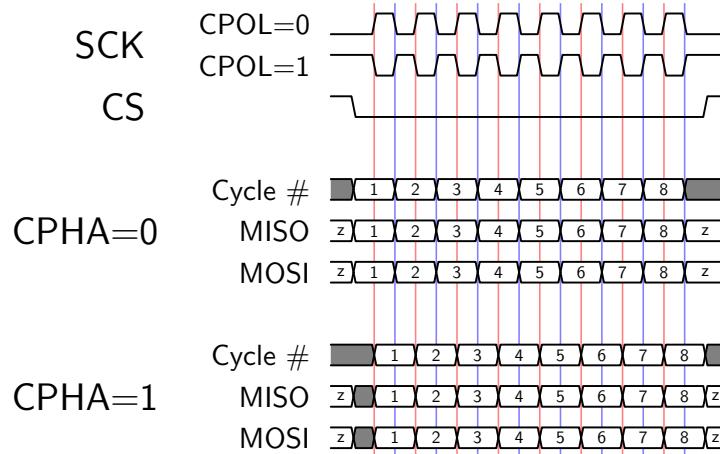


Abbildung 5.3.: SPI Datenübertragung mit unterschiedlichen Einstellungen

Tabelle 5.1.: SPI Modi Einstellungen

Mode	Clock Polarity	Clock Phase
0	0	0
1	0	1
2	1	0
3	1	1

5.2.3. Fast Fourier Transformation

Die FFT ist eine Methode, welche Daten aus dem Zeitbereich in den Frequenzbereich überführt. Diese Methode beruht auf einen Algorithmus, welcher die Eingangswerte vertauscht. Dabei wird in jedem Schritt das Abtastintervall zwischen zwei Werten verdoppelt. Dies wird solange wiederholt, bis alle Eingangswerte in 2-er Gruppen gespalten sind. Dies hat zur Folge, dass für die FFT immer ein ganzzahliges Vielfaches der Auflösungsfrequenz korrekt ermittelbar ist. Die Diskrete Fourier Transformation (DFT) benötigt N^2 und die FFT $N * \log_2(N)$ Berechnungen. Somit werden für einen Zeitbereich von z.B. 10^9 ns für die FFT ≈ 30 s und die DFT ≈ 31.2 Jahre benötigt.

Aus der Anzahl N der Punkte und der Abtastfrequenz f_A ist der Linienabstand im Frequenzbereich mit $\frac{f_A}{N}$ berechenbar. Weiterhin ist zu beachten, dass die FFT spiegelsymmetrisch zur

Entwicklung eines digitalen standalone Ultraschall-Multigate-Doppler-Systems

Mitte ist und somit nur die Frequenzen von $k = 0$ bis $k = \frac{N}{2}$ betrachtet werden und die k-te Frequenz f_k durch $f_k = f * \frac{f_A}{N}$ definiert ist.

6. Systemdesign und Implementierung

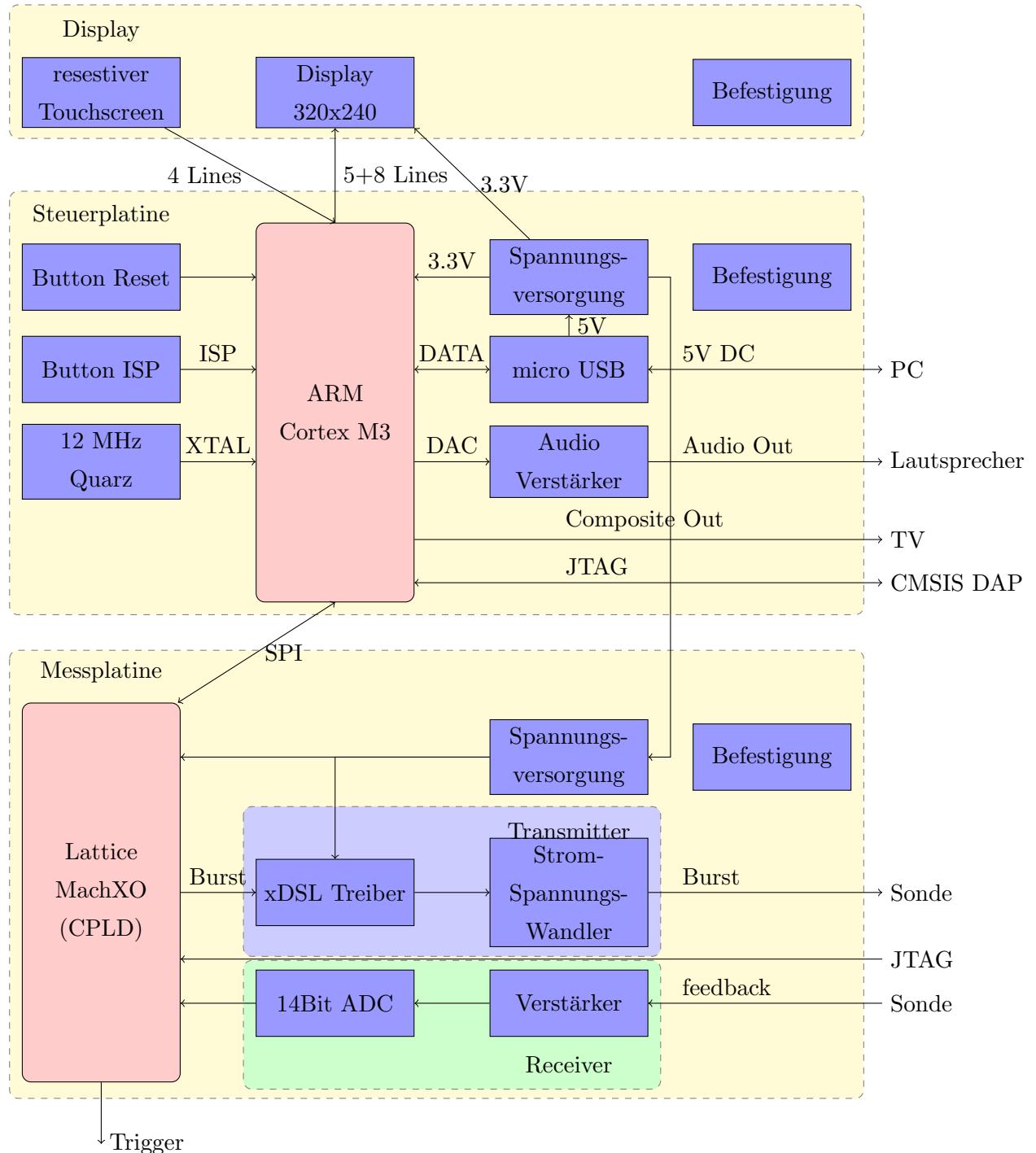
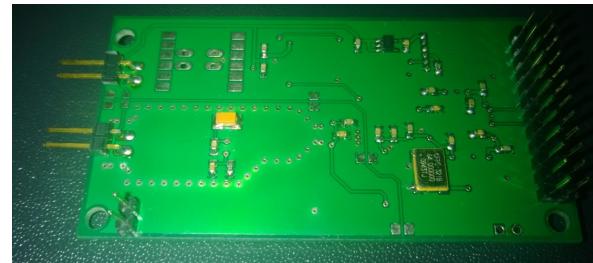


Abbildung 6.1.: Systemübersicht

6.1. Messplatine



(a) Oberseite



(b) Unterseite

Abbildung 6.2.: Messplatine - Revision 002

Die Messplatine wurde erstellt, um die Ansteuerung von dem Messsystem zu separieren. Aus diesem Grund wurde die Messplatine mit einer separaten Stromversorgung versehen. Das PCB besitzt zwei von einander getrennte Masseflächen, eine für den digitalen und eine für den analogen Teil. Diese können durch Ferrite mit einander verbunden werden und entkoppeln gleichzeitig die hohen digitalen Frequenzen des CPLD's wodurch die Messschaltung vor diesen Signalen geschützt werden kann.

Aufgeteilt werden kann die Messplatine in Transmitter-, Receiverschaltung, CPLD und Schnittstellen. Diese werden in den nächsten Unterpunkten genauer beschrieben.

6.1.1. Schnittstellen

Eine Benutzeranforderung ist die Modularisierung des Systems (siehe 2.2.3). Dafür wurden die erforderlichen Schnittstellen definiert und in diesen Abschnitt abgebildet. Dies soll eine leichtere Integration in das vorhandene und neue Systeme gewährleisten.

Die Schnittstellen wurden dabei in zwei Bereiche aufgeteilt: Daten und Funktionen. In dem vorhandenen System [3] wurde ein PIC18F4550 der Firma Microchip verbaut, welcher die Daten über einen 8 Bit breiten Streaming Parallel Port (SPP) empfängt und über eine USB Schnittstelle mit dem PC kommuniziert. Zudem wird eine Schnittstelle benötigt, welche den zyklischen Programmablauf (sogenannte States) an das CPLD überträgt sowie diverse Signale für die Steuerung dessen.

In folgenden wird das Ergebnis der definierten Schnittstellen für die Messplatine festgehalten.

JTAG Schnittstelle dient der Programmierung des CPLD's und wurde durch eine 6x1 Pinleiste mit dem Pinabstand von 50 mil bereitgestellt.

Trigger Schnittstelle besteht aus einem N-Kanal MOSFET welcher als Leistungsschalter dient und einer Pinleiste 2x1 mit dem Pinabstand 100 mil. Diese Schnittstelle wurde für die Zertifizierung des Systems vorgesehen.

Sonde Für die Verbindung der Sonde an den Prototypen wurden 2 Pinleisten mit 2 Pins mit Pinabstand 100 mil vorgesehen, welche durch zwei Lötpunkte miteinander verbunden werden können. Dies hat den Vorteil, dass sowohl eine Sonde als Sender und Empfänger oder auch eine Sonde als Sender und z.B. ein Hydrophon als Empfänger benutzt werden kann.

DATA IN dient der Steuerung sowie der Übertragung des zyklischen Programmablaufs. Diese Schnittstelle wurde durch ein SPI realisiert indem eine 2x2 Pinleiste mit Pinabstand 100 mil zu Verfügung gestellt wird.

DATA OUT ist für die Weitergabe der gesammelten demodulierten Werte verantwortlich. Damit die vorhandene SPP Schnittstelle weiterhin verwendet werden kann, wurden die zwei Steuersignale¹ für die Schnittstelle beibehalten. Zudem wurde die SSP Schnittstelle auf 16 Bit verbreitert, um höhere Datentransferraten zu ermöglichen. Eine 9x2 Pinleiste mit dem Pinabstand 100 mil wurde bereitgestellt, um ein SSP auf Basis von SPI, ein 8 Bit SSP sowie ein 16 Bit SSP zu ermöglichen.

Spannungsversorgung dient der Separierung sowie Stabilisierung der Spannung. Diese Schnittstelle ist notwendig, da eine Versorgungsspannung von 5V DC sichergestellt werden muss, damit die Messplatine in den definierten Bereich arbeiten kann. Diese Schnittstelle wurde durch das Bereitstellen von zwei 2x1 Pinleisten mit Pinabstand 100 mil sichergestellt.

6.1.2. Complex Programmable Logic Device

Für diese Einheit wurde ein MachXO 2280 der Firma Lattice gewählt, um zeitkritische Funktionen auszuführen. Dieser Integrated Circuit (IC) besitzt interne Matrizen aus logischen Und- und Oder-Gattern welche auch als Rückkopplungen dienen können. Weiterhin besitzt er Ein- und Ausgabeblöcke, welche als schnelle Speicher² programmierbar sind.

Im Gegensatz zu FPGAs kann durch die Matrixstrukturierung des ICs die Durchlaufzeit von Signalen bestimmt werden, was Ideal für zeitkritische Funktionen ist. Des weiteren besitzt dieser CPLD 2280 LUTs, welche als Speicher dienen können.

Ein CPLD arbeitet im Gegensatz zu einem uC nicht zyklisch, sondern parallel. Somit können die programmierten Funktionen gleichzeitig ausgeführt werden, wodurch sich eine Statemaschine und ALUs realisieren lassen.

Für die Messplatine dient das CPLD als Schnittstelle zwischen der Steuerplatine und der Peripherie aus Transmitter und Receiver. Nach Übertragung des zyklischen Programmablaufs steuert das CPLD selbstständig alle Aktionen für die kontinuierliche Signalerzeugung des Transmitters und der Aufbereitung der digitalisierten ADC Werte des Receivers.

¹DATA-ready und DATA-read-through

²Latches, D-Flipflops oder Register

Entwicklung eines digitalen standalone Ultraschall-Multigate-Doppler-Systems

Programmiert wurde das CPLD mit Hardware Description Language (HDL), wobei die Sprachen Very High Speed Integrated Circuit Hardware Description Language (VHDL) und Verilog bereitstellt werden. In dieser Arbeit wurde ausschließlich mit der Sprache Verilog gearbeitet um die Einarbeitungszeit zu reduzieren, eine Strukturierung des Programm abzubilden, sowie Testbenches³ zu erstellen. Das Programm des CPLDs wird dabei in Kapitel 6.4.1 näher beschrieben.

6.1.3. Transmitter

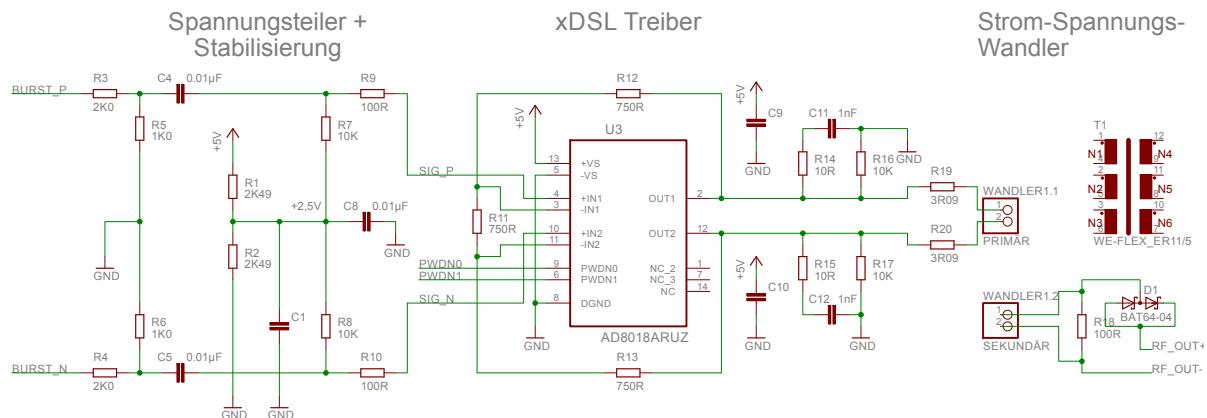


Abbildung 6.3.: Transmitter Schaltung

Das CPLD kann die erforderliche Leistung für den Betrieb der Ultraschallsonde nicht erbringen. Ein Verstärker wurde dabei integriert, welcher die erforderliche Leistung bereitstellen kann. Primär besteht die Transmitterschaltung aus einem differenziellen xDSL Verstärker der Firma Analog Devices Inc. [Abb. 6.3 U3] [17], welcher eine Bandbreite von bis zu 130 MHz schalten kann. Der Ausgang ist bis maximal 8 Vpeak-peak (pp) (differential) aussteuerbar (rail-to-rail). Er liefert bei 10Ω Last maximal 400 mA, wobei die Verzerrung dabei vergleichsweise gering mit -3 dB ausfällt. Für den erfolgreichen Betrieb des Verstärkers werden die Eingangssignale, welche das CPLD generiert durch einen Spannungsteiler von 3,3 V auf ca. 1,6 V reduziert. Anschließend werden die Signale durch den differenziellen Verstärker auf ca. 4,6 Vpp verstärkt. Dabei ist zu beachten, dass der Verstärker durch die Eigenschaltung und hohen Schaltfrequenzen eine Verlustleistung aufweist und das Potenzial reduziert.

Der letzte Teil des Transmitters besteht aus einem Strom-Spannungs-Wandler, welcher die erforderliche Spannung für das Piezoelement der Ultraschallsonde generiert. Dieser Wandler wurde selbst gewickelt indem ein Amidon-Ringkern⁴ auf der Primärseite mit einem $0,45 \text{ mm}^2$ starken Kupferdraht 8 mal und auf der Sekundärseite mit $0,5 \text{ mm}^2$ starken Kupferdraht 18 mal umwickelt wurde. Dies ergibt ein theoretisches Wandlungsverhältnis der Spannung von 1:2.25. Die Sekundärseite des Wandlers wurde mit einem 100Ω Widerstand und mit zwei anti-parallel

³Testfunktionen für die Simulation

⁴Farbe Rot, Material ... 2, Frequenzbereich: 1 MHz bis 30 MHz

Entwicklung eines digitalen standalone Ultraschall-Multigate-Doppler-Systems

Schutzdioden [Abb. 6.3 D1] versehen um den Verstärker vor Induktionsströmen zu schützen, welche durch das Ausschwingen des Piezokristalls über den Wandler erzeugt werden.

6.1.4. Receiver

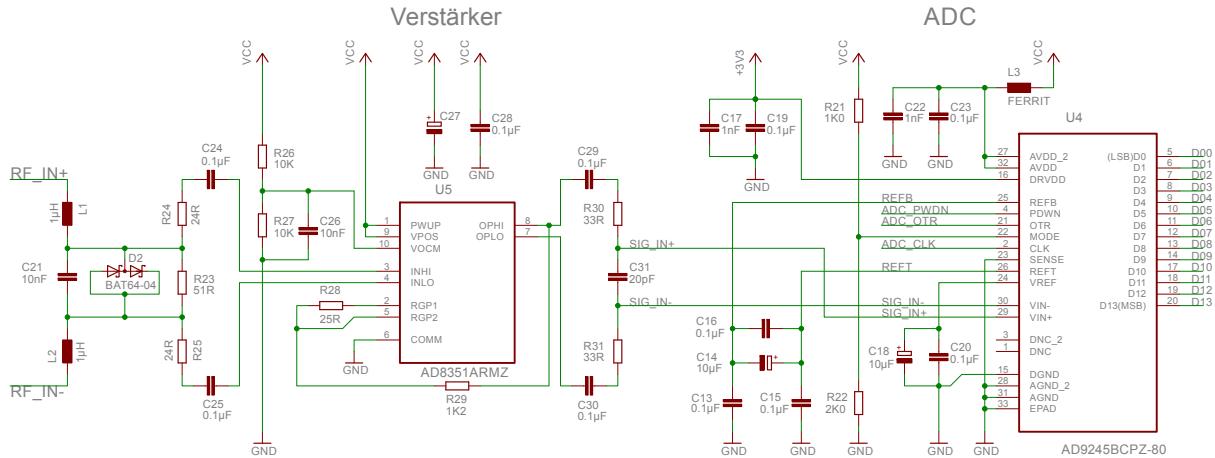


Abbildung 6.4.: Receiver Schaltung

Die Empfängerschaltung muss eine Bandbreite von 64 MHz abdecken um möglichst viele Stützstellen bei der Digitalisierung zu ermöglichen. Das Niquistkriterium wird dabei beachtet, da die maximale Trägerfrequenz von 8 MHz acht Mal in der Bandbreite vorhanden ist. Des Weiteren darf nicht vernachlässigt werden, dass mit steigender Messtiefe, die Signalstärke und somit die Auslenkung der Messamplituden reduziert wird.

Aus diesem Sachverhalt wurde eine Digitalisierung entwickelt, welche sich durch einen Vorverstärker [Abb. 6.4 U5] [18] für die geringen Amplitudenauslenkungen und einen 14 Bit ADC [Abb. 6.4 U4] [19], welcher 80 Mega Samples per second (MSps) messen kann, auszeichnet. Dabei wird der ADC mit dem Systemtakt von 64 MHz betrieben wodurch er 64 MSps messen kann.

Für den Schutz der IC's wurden wie bei der Transmitterschaltung zwei anti-parallele Schutzdioden [Abb. 6.4 D2]. vorgesehen.

6.2. Display Platine

Diese Platine wurde gewählt, da die Ansteuerung eines LCD TFT Displays ohne Bibliotheken einen enormen Ressourcenaufwand mit sich führt. Das 2.8 Zoll LCD TFT Display breakout board [20] besitzt eine Auflösung von 320x240 Pixel bei einer Farbtiefe von 18 Bit und einen resistiven Touchscreen. Für dieses Produkt stellt die Firma Adafruit ein Tutorial sowie Bibliotheken für die Ansteuerung, Grafikfunktion, Textfunktion und des Touchscreens für die Arduino

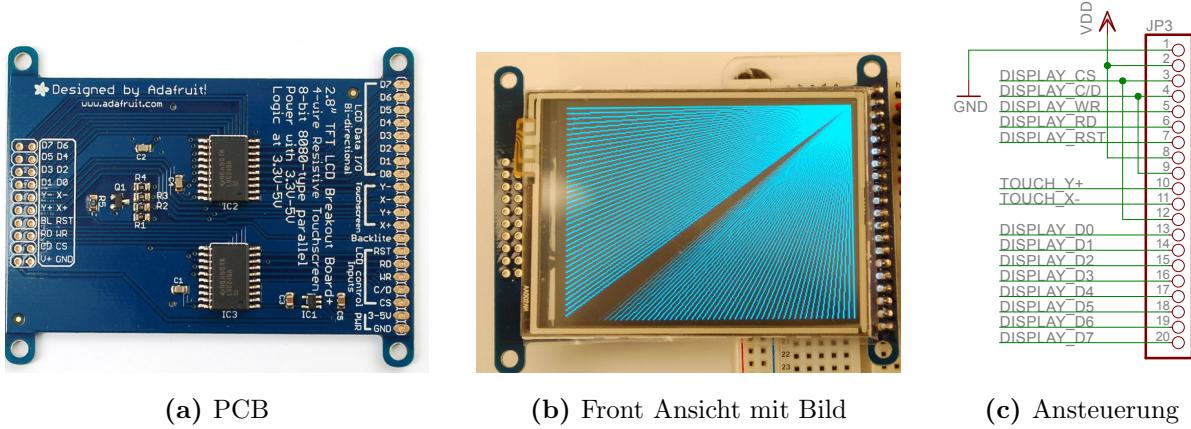


Abbildung 6.5.: LCD TFT breakout board

Plattform⁵ bereit.

Somit konnte die Zeit im Bereich research and development (R&D) für die Inbetriebnahme der Displayfunktion reduziert werden. Dafür wurden die vorhandenen Bibliotheken erfolgreich an den ARM Cortex M3 angepasst. Dabei wurde festgestellt, dass die Objektorientierte Programmierung (OOP) einen erhöhten Speicherbedarf im Random-Access Memory (RAM) des Cortex M3 benötigt. Dies liegt an der Initialisierung durch den Compiler, da dieser alle Eigenschaften des Objektes anlegen muss, auch wenn diese nicht benutzt werden. Eine Umstrukturierung in sogenannte Namespaces und Funktionen (C-Style) wurde durchgeführt und der Speicherbedarf drastisch gesenkt⁶.

Das Display hat mit den Maßen 70x50x4 mm eine Anzeigefläche von 2,8 Zoll und ist auf einer PCB geklebt, welche eine Fläche von 82x64 mm besitzt und 4 Montagelöcher mit einem Durchmesser von 3,2 mm in einen Abstand von 57,5x75 mm (2.25 x 2.95 inch).

Diese Maße wurden für die Steuerplatine übernommen und bilden die maximalen Maße des PCB. Somit kann gewährleistet werden, dass die Komponente Display board und Steuerplatine auf einander montierbar sind.

Das Display stellt 4 weiße LED's für die Hintergrundbeleuchtung bereit. Des weiteren besitzt das board einen ILI9325 oder einen ILI9328 IC in denen ein integrierter Video Speicher bereitgestellt wird. Das Interface besteht aus einen 8 Bit breiten Port und 4 bis 5 Steuerungsleitungen. Eine Ansteuerung durch einen 8 Bit uC ist aus diesen Grund möglich. Die Implementierung des Displays ist in Abb. 6.5c dargestellt. Eine Pin Kombination zwischen Pin 9 (Signal Y-) mit Signal C/D und Pin 12 (Signal X+) mit Signal CS wurde durchgeführt, um das routen des PCB zu erleichtern, sowie die benötigte Anzahl der uC Pins zu reduzieren.

⁵Rapid Prototyping Plattform auf Basis des ATTEL Atmega uC

⁶OOP belegt 32% des RAM, C-Style 3% des RAM

6.3. Steuer- und Auswertungsplatine

6.3.1. Energieversorgung

Die Energieversorgung der Steuerplatine beherbergt 2 Schnittstellen. Eine USB-Micro Buchse für 5 V DC und eine 2,1 mm DC Power Jack Buchse für Eingangsspannungen von bis zu 30 V DC. Diese Kombination wurde gewählt, da eine Stromaufnahme von 450 bis 550 mA kalkuliert wurde und eine USB Schnittstelle eines PCs maximal 500 mA zu Verfügung stellen kann. Zu Sicherung der USB Schnittstelle wurde eine BAT64 Schottky Diode verwendet, die jedoch den Nachteil hat, die Systemspannung auf 4,2 V zu reduzieren. Für den Schutz des DC Netzteils wurde hingegen eine B0540W Schottky Diode verwendet, da diese einen höheren Durchlassstrom verarbeiten kann.

Für das Umschalten der Energiequellen wurde ein Analog Switch [21] benutzt, welcher bei gleichzeitigen Anlegen der Netzteilspannung und USB-Spannung das externe Netzteil durchschaltet.

6.3.2. Audio Verstärker

Um die Audioqualität zu erhöhen, wurden ein Tiefpassfilter und ein Audioverstärker (siehe Abb. 6.6) [22] verwendet. Mit dieser Schaltung kann ein 8Ω 325 mW Lautsprecher verwendet werden.

Durch Anpassung von Widerstand R50 des Spannungsteilers kann die Lautstärke weiterhin vergrößert oder verkleinert werden.

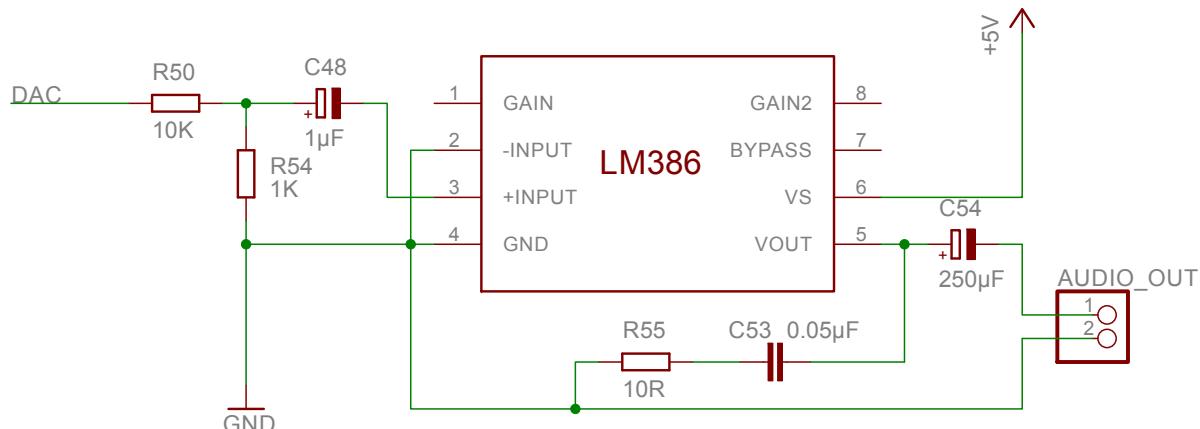


Abbildung 6.6.: Audioverstärker und Tiefpass

6.3.3. ARM Cortex M3

Für die weitere Aufbereitung der Signale sowie der Darstellung dieser wurde ein ARM Cortex M3 Kern der Firma NXP ausgewählt. Der LPC1768 zählt zu der LPC ConnectPlus Serie. Diese bietet standardmäßig Ethernet, USB host / device / OTG, CAN2.0B, PWM, RTC, I2S, ADC,

Entwicklung eines digitalen standalone Ultraschall-Multigate-Doppler-Systems

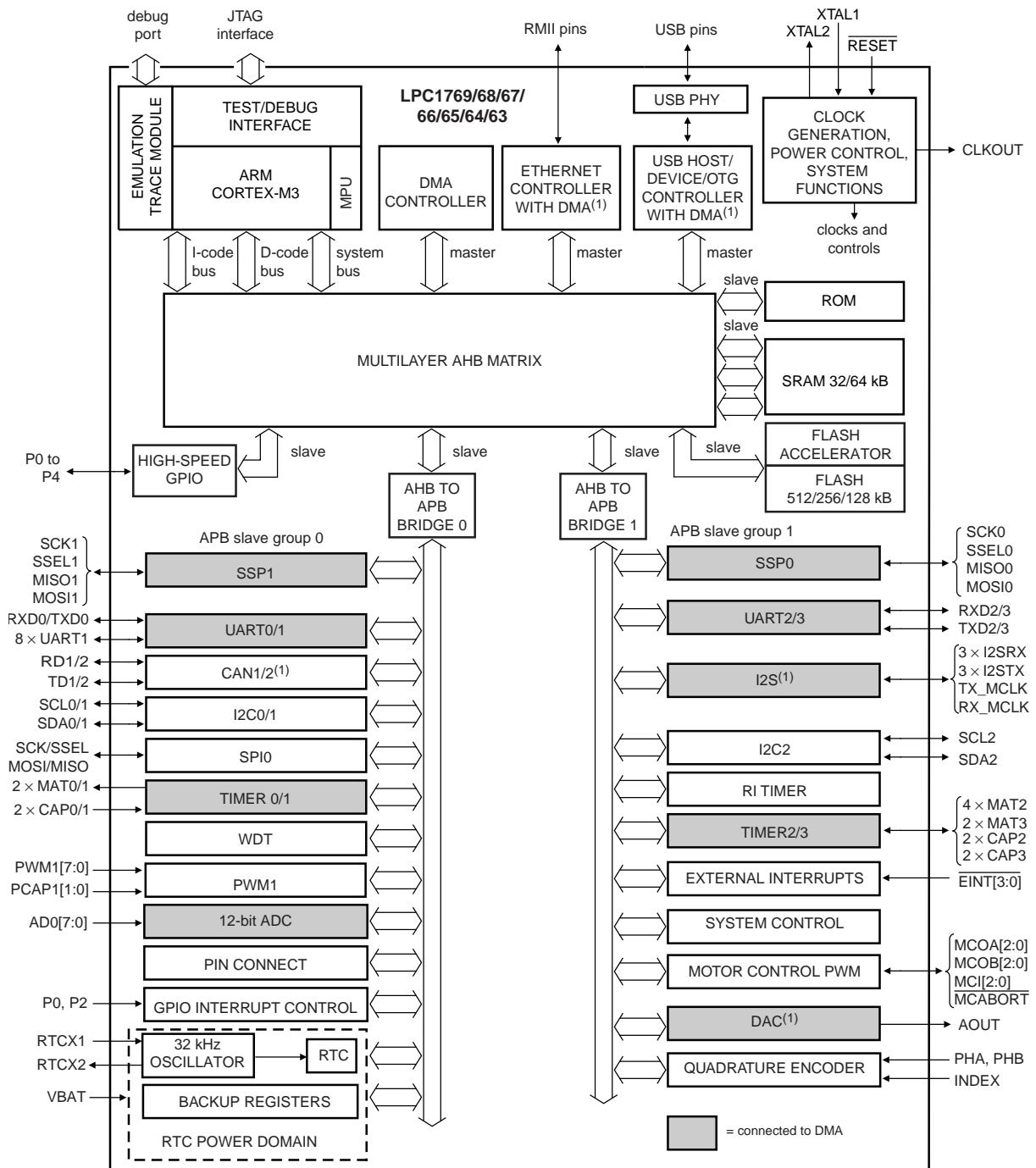


Abbildung 6.7.: Blockdiagramm NXP LPC176X

DAC, UART, I2C und SPI (siehe Abb. 6.7) an. Einen weiteren Vorteil der Serie bietet die Pin Kompatibilität zwischen den Kernen. So kann nachträglich der M3 Kern durch einen M4 Kern mit floating-point unit (FPU) ersetzt werden, wenn diese Funktionen benötigt werden.

Der LPC1768 wird mit 96 MHz betrieben und besitzt einen für das Projekt wichtigen DMA Controller. Dieser Controller hat den Vorteil, Daten intern sowie auch extern zu transportieren,

während die Hauptschleife weitere Aktionen parallel zu dem DMA Controller ausführt. Zudem besitzt der LPC1768 einen 512 KB großen Flash sowie einen 64 KB großen RAM Speicher. Dies hat den Vorteil, auch kleinere Grafiken und genügend Pufferspeicher für die Signalverarbeitung ohne externen Speicher bereitzustellen. [23] [24] [25] [26]

Der uC kommuniziert über die in 6.1.1 beschriebenen Schnittstellen **DATA IN** und **DATA OUT**. Für die Übertragung des zyklischen Programmablaufs über die **DATA IN** Schnittstelle wurde ein Software SPI mit Enable / Disable Funktion des CPLDs implementiert. Der Datentransfer der demodulierten Signale erfolgt über die **DATA OUT** Schnittstelle, welche durch das Hardware SPI angesteuert wird. Dabei wird das SPI durch den internen DMA Controller des M3 Kerns angesteuert, und erzeugt einen Interrupt bei Beendigung der Übertragung. 4 Pins des Kerns können für die Flankenerkennung genutzt werden. Somit kann der „Almost-Full“ Flag des CPLD Ausgangspuffers genutzt werden, um den DMA Controller anzusteuern und die Datenübertrag zu starten.

6.4. Software

Nach Erstellung des Hardwaredesigns wurde der Aufbau des zyklischen Programmablaufs für das CPLD definiert. Dabei wurde festgelegt, dass der Programmablauf aus mehreren Zyklen (sogenannte States) besteht. Diese States besitzen einen 16 Bit Zählerwert sowie Aktionen, welche in diesem Zyklus durchgeführt werden. Der Zählerwert ist für die aktive Dauer des Zyklus verantwortlich, wodurch eine Statemachine (6.4.1) in Kombination mit einem Zähler und Vergleicher realisierbar ist. Tabelle 6.1 beschreibt den groben Aufbau eines States, wobei darauf geachtet wurde, dass der 16 Bit lange Counterwert ohne größere Bearbeitung im uC an das CPLD übertragen werden kann. Eine Länge von 4 Byte wurde gewählt, da 26 Bit für einen State bereitgestellt werden müssen. Dabei ergibt sich eine Reserve von 6 Bit für zukünftige Aktionen des CPLD's ohne die Datenübertragungsfunktionen ändern zu müssen.

Tabelle 6.1.: Aufbau eines States

	16 Bit Countervalue	6 Empty Bits	10 Action Bits
State Bit	[31...16]	[15...10]	[9...0]
State Length	4 Byte		

In Tabelle 6.2 werden die einzelnen Aktionen samt Deklaration der Bit Kombinationen aufgeführt. Bit Kombinationen reduzieren die Anzahl an Bit, die benötigt werden um Aktionen zu deklarieren. Jedoch erhöht diese Technik den Aufwand an Logikgattern da keine direkte Zuweisung von Bit zu Aktion möglich ist.

Es wurde sich für diese Technik entschieden, da diese flexibler ist und spätere Änderungen durch das Verändern der Bit Kombination vereinfacht wird.

Tabelle 6.2.: Actions

Action	State Bit	Bit combination description		
Frequency	0	0	0	DefaultValue: 8 MHz
		0	1	2 MHz
	1	1	0	4 MHz
		1	1	8 MHz
Transmitter	2	0		no action
		1		generate burst
Sampling	3	0	1	write SampleVolume 1 to buffer
		1	0	write SampleVolume 2 to buffer
	4	A = B		On value changed, write calculated data to buffer
Retransmit	5	0		no action
		1		reset counter and go to state zero
Output Interface	6	0	0	SPI
		0	1	8 Bit SSP
	7	1	0	16 Bit SSP
		1	1	No Function
Measure Type	8	0		Measure ultrasonic reflections without echoes
		1		Measure ultrasonic reflections with echoes ⁷
Trigger	9	0		Off
		1		On

6.4.1. CPLD

Architektur

Die Logik des CPLDs wurde wie in 6.1.2 beschrieben mit der Programmiersprache Verilog umgesetzt. Die Syntax der Sprache ist dabei ähnlich der von C wodurch es möglich war, die Architektur der Software Agil abzubilden. Dies hat den Vorteil, dass einzelne Module separat getestet werden können und die Weiterentwicklung erleichtert wird. Eine Abstrahierung der Architektur wurde entwickelt und in Abb. 6.8 dargestellt. Diese Architektur diente dabei der Systemmaischen Implementierung der Module.

⁷attractive for material measurements

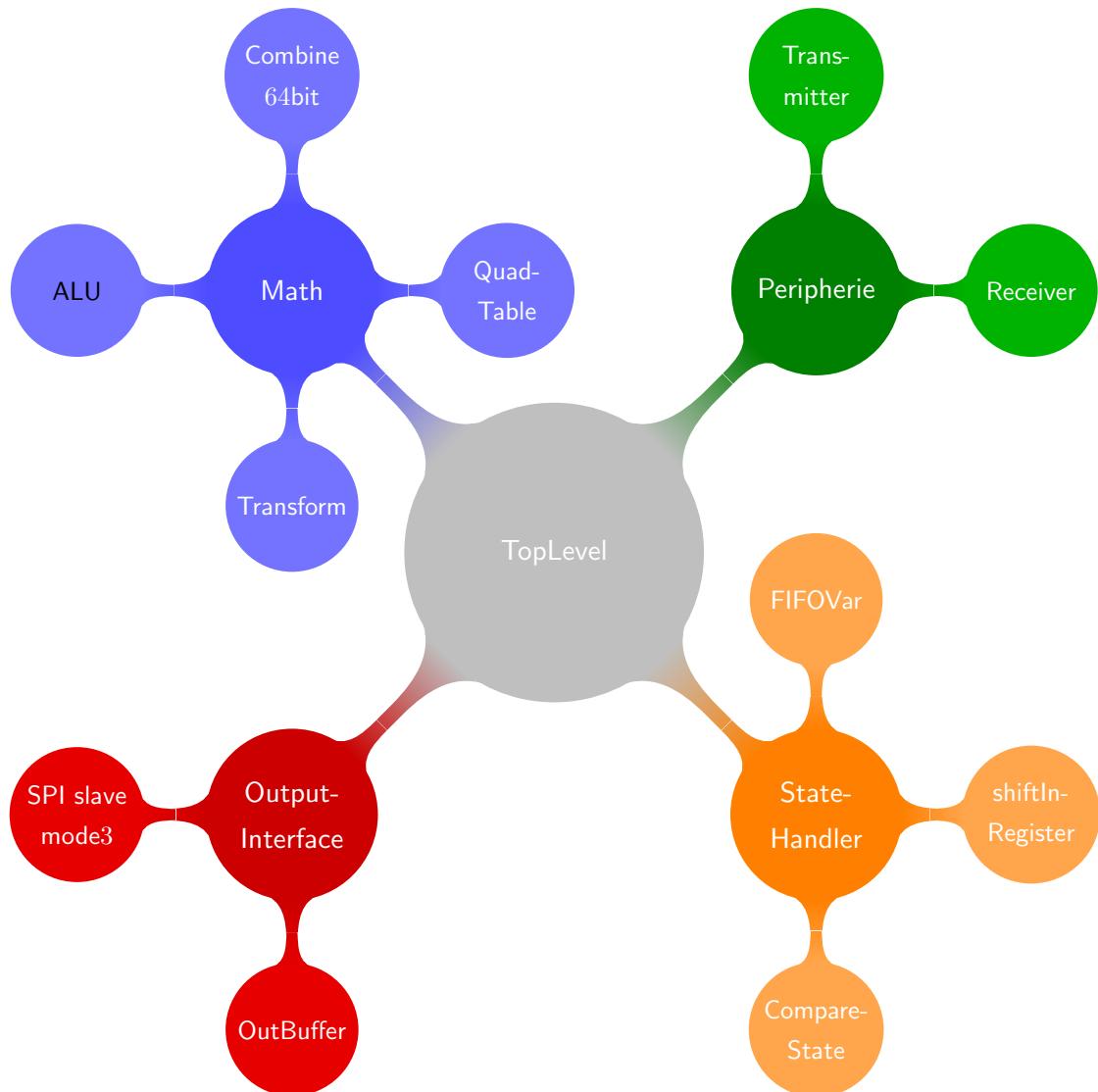


Abbildung 6.8.: Module CPLD und Architektur

Modul TopLevel

Das *TopLevel* Modul, welches erstellt wurde verwaltet die Einzelnen Hauptmodule und dient als Schnittstelle zwischen diesen. Dabei werden die Signale des *State-Handler* Moduls weitergereicht und Datenbusse zwischen dem *Peripherie* Modul, dem *Math* Modul und den *Output-Interface* Modul bereitgestellt. Somit können leicht neue modifizierte Module in das vorhandene System integriert werden.

Modul State-Handler

Das Untermodul *State-Handler* besteht aus 3 Modulen. Dabei dient das *shiftIn-Register* als Programmierung des *FIFOVar* Moduls. Es empfängt die States, welche von der Steuerung ge-

Entwicklung eines digitalen standalone Ultraschall-Multigate-Doppler-Systems

neriert wurden und schreibt diese in das *FIFOVar* Modul, welches einen Ringspeicher darstellt. Zudem wird mit dem Modul *Compare-State* ein Zähler mit Vergleichsfunktion implementiert, welche den 16 Bit Countervalue des aktuellen States (siehe Tabelle 6.1) mit dessen Zählwert vergleicht. Solange der Wert des Zählers kleiner des Wert des States ist, bleibt dieser State aktiv. Wenn der Zähler den Zählwert solange inkrementiert hat, sodass die Werte identisch sind, wird das Leseregister des *VIFOVar* Moduls um Eins erhöht, wodurch der nächste State aktiviert wird.

Die Aktion **Retransmit** (siehe Tabelle 6.2) hingegen setzt dabei das Leseregister des *FIFO-Var* und den Zähler des *Compare-State* Moduls auf Null wodurch eine Reinitialisierung erfolgt und ein zyklischer Ablauf realisiert werden kann. Um dies Ablauf zu unterbrechen wurde das Signal **Enable** implementiert. Dieses dient in negierter Form wie die Aktion **Retransmit** dem Zurücksetzen der Werte und muss anliegen, damit die Statemachine und dessen Komponenten arbeiten dürfen.

In Abbildung 6.9 wurde versucht, diesen Ablauf grafisch darzustellen.

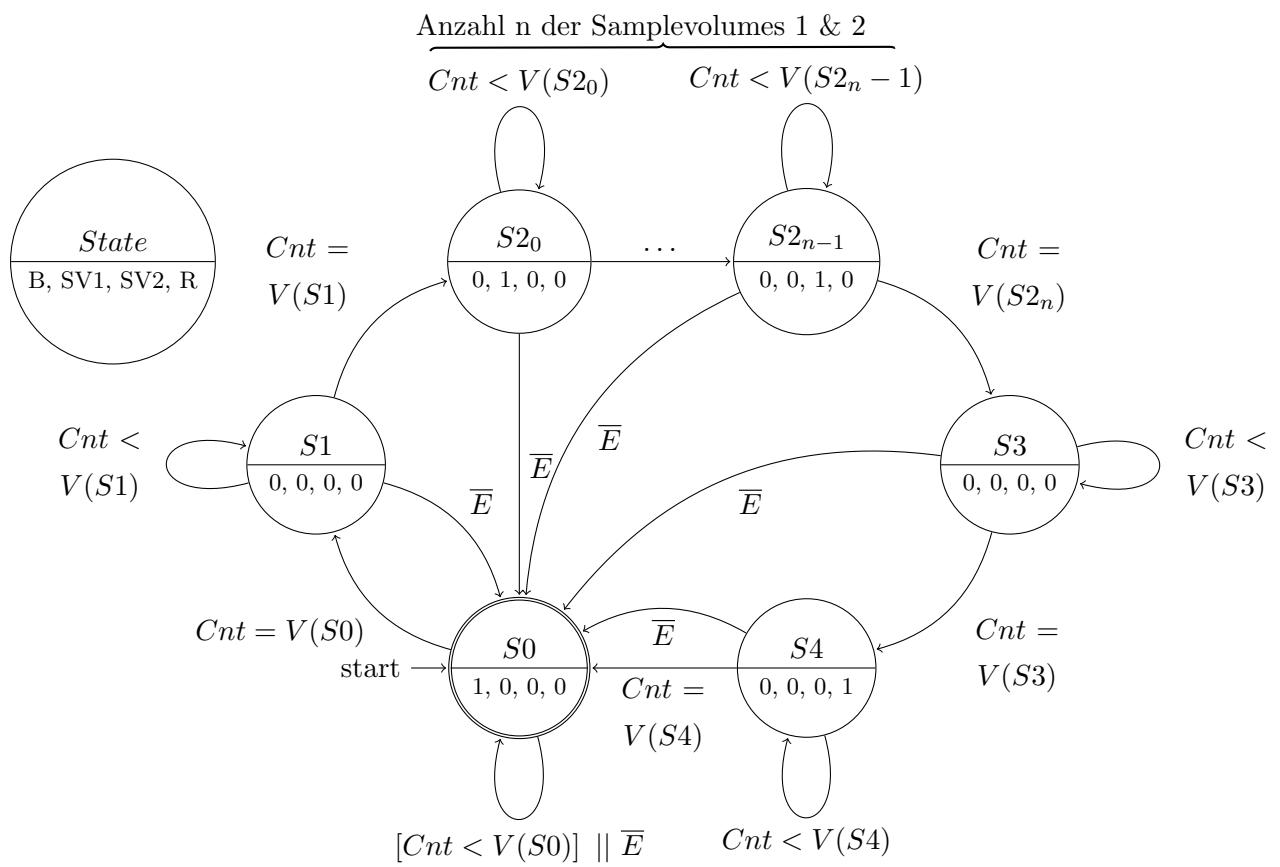


Abbildung 6.9.: Modul State-Handler - Statemachine

Die in Abb. 6.9 dargestellten States S0 bis S4 stellen den normalen Ablauf der Statemachine dar. Dabei ist **B** die Aktion Bustsignal, **SV1** die Aktion Samplevolume1, **SV2** die Aktion Samplevolume2, **R** die Aktion Retransmit, **Cnt** der Zählwert des *Compare-State* Moduls und **V(X)**

der Vergleichswert (16 Bit Countervalue) des jeweiligen States.

Zuerst wird das Burstsignal erzeugt (State S0), welches eine bestimmte Zeit andauert und durch den State S1 abgeschaltet wird. State S1 wird benötigt, da nach dem Burst nicht sofort die Dopplersignale gelesen werden. Anschließend werden die States $S2_0$ bis $S2_{n-1}$ abgearbeitet, welche das Samplen und die Weiterverarbeitung der digitalisierten ADC Werte aktivieren. Dabei kann die Anzahl der States variieren, was eine optimierte Ausnutzung der Daten darstellt. Nachdem die benötigten Messungen durchgeführt wurden, wird durch den State S3 ein Delay erzeugt. In Kombination mit State S4 wird somit die PRF realisiert. Grund dafür ist, dass die Messungen (State $S2_0$ bis $S2_{n-1}$) nicht bis Ende der PRF laufen und eine Falsche PRF erzeugen würden. State S4 setzt dabei den Zähler **Cnt** sowie das Leseregister des *FIFOVar* zurück und schaltet somit zu State S0.

Modul Peripherie

Das Modul *Peripherie* dient als Schnittstelle zwischen den *TopLevel* Modul und der Peripherie, wobei eine Initialisierung der Module *Transmitter* und *Receiver* mit den erforderlichen Parametern durchgeführt wird.

Das *Transmitter* Modul besteht dabei aus einen Frequenzteiler mit Ausgangsansteuerung sowie einer enable Funktion. Ähnliche Funktionen werden dabei auch in den Modul *Receiver* verarbeitet. Ein Frequenzteiler für die Samplingfrequenz des ADCs sowie einen Signal, welches nach 7 Zyklen der reduzierten Frequenz aktiviert wird, werden bereitgestellt. Dieses Signal wurde integriert, um das Delay des ADC weiteren Modulen zu Verfügung zu stellen und somit Fehler zu reduzieren.

Modul Math

In dem Modul *Math* wird die Demodulierung des Eingangssignales sowie ein FIR Filter realisiert. Dabei werden die Eingangssignale mit den aktuellen Wert der rotierenden Sin/Cos-Tabelle multipliziert, transformiert und addiert(3.2). Dafür wurde eine kleine ALU (Modul *ALU*) implementiert, welches die Multiplikation, Division, Addition sowie Division beherrscht.

Bei diesen Funktionen ist zu beachten, dass die Bitanzahl N mehrmals vergrößert wird. Bei der Multiplikation beträgt das Resultat

$$N_{res} = N_a + N_b \quad (6.4.1)$$

wodurch aus den 14 Bit ADC Werten 22 Bit Werte entstehen. Diese 22 Bit müssen anschließend addiert werden, wobei ein Überlauf des Ergebnisses möglich ist. Aus diesem Grund wird eine Transformation von 22 Bit in 32 Bit durchgeführt, um diesen Fehler zu vermeiden. Dabei wird anhand des *Most Significant Bit* (msb) entschieden, ob den Bits 23 bis 32 der Wert Null oder Eins zugewiesen werden. Anschließend wird eine weitere Addition durchgeführt. Die 32 Bit des

Entwicklung eines digitalen standalone Ultraschall-Multigate-Doppler-Systems

Sinuswertes werden anschließend mit den 32 Bit des Cosinuswertes kombiniert, wodurch ein definiertes Speichern der Werte ermöglicht werden kann.

Modul OutputInterface

Im Modul *OutputInterface* wird der Datenaustausch mit der Steuerplatine bereitgestellt. Dabei werden die vom Modul *Math* bereitgestellten Daten in ein FirstInFirstOut (FIFO) Puffer geschrieben und anschließend über das Modul *SPIslave*, welches ein SPI mit Konfiguration Mode 3 bereitstellt, übertragen.

6.4.2. ARM Cortex M3

DMA Transfer

Für den DMA Transfer muss die Quelle, das Ziel sowie Datenmenge definiert werden. Dabei wurde ein chararray mit Größe 1 (Wert 0) als Quelle und ein der SSP mit SPI als Ziel für den ersten Transfer gewählt. Des weiteren muss beachtet werden, dass der SPI initialisiert ist, da dies nicht über die DMA Register geschieht. Dieser Transfer betreibt jedoch nur die MOSI und SCK Leitungen, wodurch ein weiterer Transfer initialisiert werden muss. Der selbe SPI dient dabei als Quelle und ein chararray mit Größe 1280 (160x64 Bit Werte = 160 Messungen) als Ziel. Zudem werden bei Initialisierung des Transfers Interrupts registriert, welche einen möglichen Fehler und den Abschluss des Transfers der Hauptschleife übergeben. Es stellte sich heraus, dass bei Benutzung der Bibliothek eine zusätzliche Konfiguration benötigt wird, welche nach Beendigung des Transfers als Dummy dient. Dabei wird die aktuelle Konfiguration des DMA Controllers in diese Dummy Konfiguration geladen und anschließend gelöscht sowie freigegeben. Die Grundkonfigurationen somit können weiter von dem System genutzt werden, wodurch keine Reinitialisierung der Werte notwendig ist und eine Verzögerung des Datentransfers vermiedet. [24]

Touchfunktion

Für die Touchfunktion wurde die template Klasse „FastIO“ um ADC Funktionalitäten erweitert. Diese Funktionen wurden anschließend in die Touchfunktionen integriert.

Dabei werden die Grundfunktionen zum Lesen der X-, Y-Werte, eine Druckvalidierung und eine Skalierungsfunktion bereitgestellt. Diese wurden in der Funktion *Read()* kombiniert. Dabei werden zuerst die X- und Y-Werte gelesen und nachfolgend der Mindestdruck validiert. Wenn ein zu geringer Druck auf den Touchscreen ausgeübt wird, erfolgt keine weitere Verarbeitung der Werte um Rechenzeit zu gewinnen. Wenn der Druck jedoch hoch genug ist, wird eine Skalierung der Werte durchgeführt. Dabei werden die Werte ein weiteres Mal aufgenommen und validiert, um eine Fehlinterpretation der Werte zu verhindern.

Der Aufruf der Funktion ist einfach zu implementieren, da nur die Funktion *Read()* aufgerufen werden muss, und anschließend der Wahrheitswert der Variable *IsTouched* abgefragt werden kann. Anschließend kann auf die X- und Y-Werte über die Variablen *pointx*, *pointy* zugegriffen werden.

Messwerte wurden aufgenommen, mit denen die Konstanten der Skalierung durch folgende Formeln errechnet werden konnten.

$$Wert_{Offset} = Wert_{min} \quad (6.4.2)$$

$$X_{scale} = (X_{max} - X_{min}) * 1000 / 320 \quad (6.4.3)$$

$$Y_{scale} = (Y_{max} - Y_{min}) * 1000 / 240 \quad (6.4.4)$$

Dabei dient der Wert 1000 für die Fehlerreduzierung, da eine Datenverarbeitung mit Ganzen Zahlen weniger Zyklen als eine Gleitkommaverarbeitung benötigt. Die Werte 320 und 240 stellen dabei die Werte der Displayauflösung 320x240 dar. [24]

Menüführung

In diesen Abschnitt wird der grundlegenden Ablauf des Programms beschrieben.

Zuerst werden die Strukturen und Klassen initialisiert. Dabei wird das Display resetet und initialisiert, der Timer für die Touchfunktion auf 0,2 Sekunden gesetzt und Interrupts implementiert. Anschließend wird die Hauptschleife aufgerufen, welche aus 2 Abfragen besteht: Wurde ein Interrupt für das Abholen der CPLD Daten gefeuert und soll die Touchfunktion ausgelesen werden. Dabei hat die Touchfunktion eine niedrigere Priorität wie der Datentransfer.

Wenn der Interrupt für das Abholen der Daten gefeuert wurde, wird überprüft, ob sich das Programm in den richtigen Menupunkt (Graph) befindet. Wenn dies nicht der Fall ist, wird keine Aktion durchgeführt. Hingegen wird der Datentransfer eingeleitet, wenn sich der Benutzer in dem Menupunkt Graph befindet. Zudem wird überprüft, ob der Datentransfer das erste mal angestoßen wurde. Diese Abfrage wurde implementiert, da sich bei den ersten Aufruf der Funktion noch keine definierten Daten in dem Zielspeicherbereich des DMA Transfers befinden können. Nachdem der erste Transfer durchgeführt wurde, werden die Daten graphisch angezeigt. Es ist zu beachten, dass immer zurück zur Hauptschleife gesprungen wird, wenn diese Funktionen durchgeführt wurden.

Da alle 0,2 Sekunden die Touchfunktion verarbeitet werden soll, wird zuerst geprüft, ob der erforderliche Druck anliegt. Wenn dies nicht der Fall ist, springt das Programm zurück zur Hauptschleife. Hingegen wird bei einen Mindestdruck die aktuelle Position des Druckpunktes skaliert. Nachdem wird überprüft, in welchen Menupunkt sich der Benutzer befindet.

Im Menupunkt Graph können lediglich zwei Funktionen ausgeführt werden. Eine zum Umschalten des Graphen (M-Mode und Doppler-Spekrogramm) und eine Funktion für das Wechseln in das Einstellungsmenu.

Entwicklung eines digitalen standalone Ultraschall-Multigate-Doppler-Systems

Das Einstellungsmenu beherbergt 4 Pfeile (hoch, links, rechts, runter) um eine Navigation zu ermöglichen. Des weiteren einen Button um die USB Funktion zu aktivieren und einen Button um zu dem Menupunkt Graph zu gelangen. Für die Parametrierung der Messung können die Trägerfrequenzen auf 2, 4 und 8 MHz, die PRF von 2 bis 12 Kilo Herz (kHz), die Bursttiefe von 2 bis 20 mm, sowie Starttiefen, Endtiefen und Verteilung der Messwerte der zwei Messtiefen eingestellt werden.

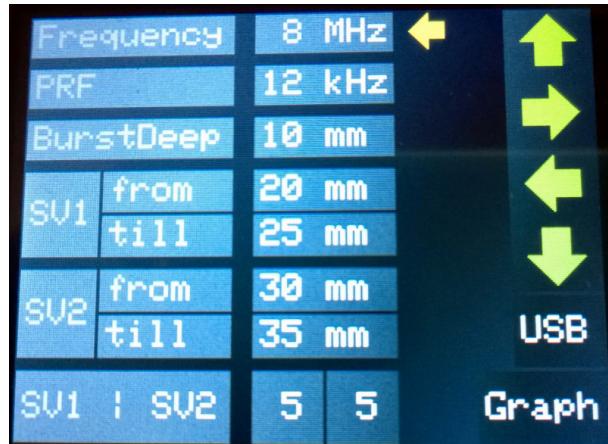


Abbildung 6.10.: kalibriertes Menu

Um eine Fehleingabe entgegenzuwirken wurde eine Errorhandling implementiert, welches nach Änderung einer Einstellung die Parameter prüft. Durch diese Funktionalität kann dem Benutzer visualisiert werden, ob eine Messung erfolgreich durchgeführt werden können. Dabei wurde auf die Alarmfarbe rot zurückgegriffen, welche den aktuellen Parameter und die Schaltfläche Graph färbt, wenn die Parametrierung fehlerhaft ist. Bei der Überprüfung werden die Parameter PRF, Bursttiefe, sowie die Start- und Endtiefen verarbeitet. Zudem wird das Verhältnis der Messungen überprüft. Grund dafür ist, dass der erste Messbereich mindestens einen Messwert haben sollte, bevor der zweite Messbereich verarbeitet werden kann.

Darstellung von Bildern

Um Grafiken effektiv mit den ARM Cortex nutzen zu können, sollte die Farbtiefe auf 1 Bit (Schwarz-Weiß) reduziert werden. Dies verringert den benötigten Speicherplatz drastisch.

Dafür wurde das Programm GIMP (siehe 5.1.1) genutzt. Es bietet alle Funktionen um die Informationen einfach in die C/C++ Umgebung zu überführen. Dabei wird das Bild mit dem Programm geöffnet und folgende Funktion aufgerufen: Bild → Modus → Indiziert → Option Schwarz/Weiß-Palette (1-Bit) aufrufen und bestätigen mit Umwandeln. Danach wurde eine Nachbearbeitung des Bildes durchgeführt um ein bestmögliches Ergebnis zu erzielen. Anschließend wird das Bild mit Datei → Exportieren ... → mit der Option X-BitMap-Bild (*.xbm, *.icon, *.bitmap) in eine Datei gespeichert, welche direkt in der Programmiersprache C verarbeitet werden kann.

Für die Darstellung auf dem Display wurde die Funktion A.2 entwickelt.

6.4.3. Kalkulation der State-Zählerwerte N

Allgemein muss bei allen Berechnungen beachtet werden, dass der Maximale Zählerwert von 65.535 nicht überschritten wird. Dies liegt der zu Verfügung stehenden Bitanzahl des Counter-values (siehe 6.1) zu Grunde.

$$N_{max} = 2^{16} = 65.536 \quad (6.4.5)$$

Für den Zählerwert N der Burst Tiefe benötigt man die Ausbreitungsgeschwindigkeit $Velocity$ und die Taktfrequenz des CPLDs f_{CPLD} womit folgende Formel entwickelt wurde

$$N(Burstdeep) = \frac{f_{CPLD}[Hz]}{Velocity[mm/s]} * Burstdeep[mm] \quad (6.4.6)$$

$$N(Burstdeep) = \frac{64000000}{1500000} * Burstdeep \quad (6.4.7)$$

$$N(10mm) = \frac{640}{15} * 10mm = 426, \bar{6} \quad (6.4.8)$$

Wie in Formel 6.4.8 erkennbar ist, entsteht als Ergebnis eine Gleitkommazahl. Somit muss eine Datentypkonvertierung von Typ Integer zu Double durchgeführt werden, um Rundungsfehler zu vermeiden. Eine Abrundung des Wertes wird immer bei Schreiben des States automatisch durchgeführt.

Die Kalkulation der Samples hingegen ist komplizierter. Ein Samplevolume SV hat dabei eine Starttiefe $SVS[mm]$, eine Endtiefe $SVE[mm]$ sowie eine Variable Anzahl von Samples $SVN \in \mathbb{N}$. Hinzu kommt, ob die Messung wie im Anwendungsfall durch Reflexion den doppelten Weg oder wie im Zertifizierungsfall den einfachen Weg zurücklegen muss. Dabei wurde folgende Formel entwickelt, welche auf Basis der Formel 6.4.8 beruht und für den einfachen Weg gilt:

$$N(SampleN, SVN, SVS, SVE) = \frac{640}{15} \left(\frac{SVE - SVS}{SVN} * SampleN + SVS \right) \quad (6.4.9)$$

$$N(1, 10, 20, 25) = \frac{640}{15} \left(\frac{25 - 20}{10} * 1 + 20 \right) = 874, \bar{6} \quad (6.4.10)$$

Wie in Formel 6.4.8 muss auch in Formel 6.4.9 eine Datentypkonvertierung durchgeführt werden. Für den doppelten Weg wird das Ergebnis der Formel 6.4.9 mit 2 multipliziert.

Die Berechnung der PRF[Hz] ist zeitabhängig und beruht nicht auf der Ausbreitungsgeschwindigkeit. Somit wird für den richtigen Zählerwert N die Taktfrequenz des CPLDs $f_{CPLD}[Hz]$ und die PRF benötigt.

$$N(PRF) = \frac{f_{CPLD}}{PRF} = \frac{64000000}{PRF * 1000} \quad (6.4.11)$$

$$N(PRF) = \frac{64000}{PRF} \quad (6.4.12)$$

$$N(12) = \frac{64000}{12} = 5333, \bar{3} \quad (6.4.13)$$

Entwicklung eines digitalen standalone Ultraschall-Multigate-Doppler-Systems

Durch die Formel 6.4.12 wird bestätigt, dass mit einer PRF von 12 kHz ein Zählwert von 5333 erreicht wird. Für eine PRF von 2 kHz ergibt sich aus der Formel 6.4.12 ein Zählwert von 32000. Somit kann mit einen 16 bit Zählwert und einer CPLD Taktfrequenz von 64 MHz eine minimale PRF von 976,5625 Hz realisiert werden.

7. Test und Ergebnisse

Das Testen ist Zentrales Element im Bereich R&D. Es ermöglicht die Sicherstellung der Funktionalität der Komponenten und des Systems. In Projekten mit mehreren Gruppenmitgliedern werden die Funktionen durch Teilprojekte gegliedert und Schnittstellen deklariert, wodurch das System modularisiert wird. Somit können für die einzelnen Funktionen und Module sogenannte Unit-Tests¹ durchgeführt werden. Nachdem alle Unit-Tests erfolgreich absolviert wurden kann ein Integrationstest² durchgeführt werden. Meist werden an den Systemschnittstellen unter Laborbedingungen definierte Signale angelegt, um die erwartete Reaktion des Systems zu bestätigen. Nach den Labortests erfolgt der Systemtest unter natürlichen Bedingungen in der Umgebung, in dem das System später eingesetzt werden soll, das heißt (d.h.) an den Schnittstellen werden keine definierten Signale mehr angelegt.

Auf Grundlage der funktionalen Benutzer- und Systemanforderungen kann mit den Unit- und Integrationstests das System verifiziert werden. Der Unit-Test erlaubt die Überprüfung der funktionalen Software- und Hardwareanforderungen.

7.1. Komponententest (Unit-Test)

7.1.1. Debuggen des Cortex M3

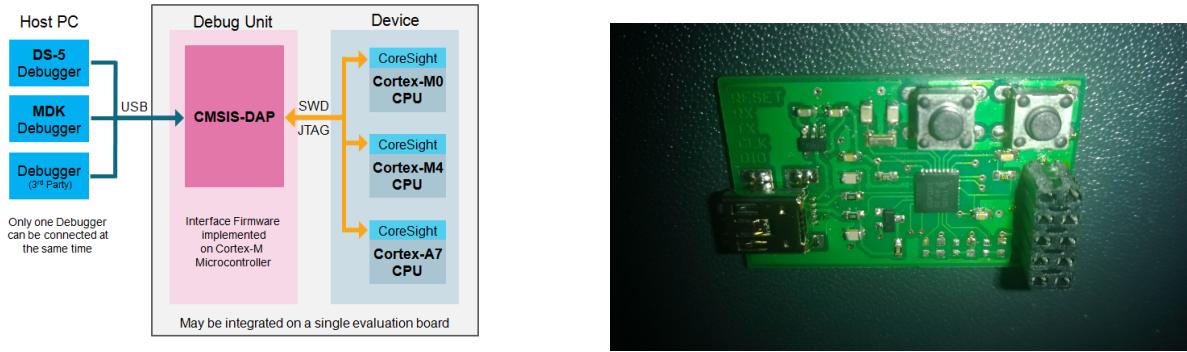
Ein besonderes Augenmerk gilt den Debuggen. mbed.org stellt hierfür ein Hardware Development Kit (HDK) zu Verfügung, welches über die CMSIS-DAP (Firmware for the CoreSight Debug Access Port) Schnittstelle arbeitet. Dabei dient ein ARM Cortex M0 Kern als Debugger und Flasher, welcher via USB mit dem PC kommuniziert. Das HDK von mbed.org bietet zudem die Drag&Drop Programmierung an.

Testaufbau und -durchführung

Für diesen Test wurde ein Microsoft[©] Windows[®] 7 und 8.1 System verwendet. Dabei wurde die Steuerplatine mit den Debugger PCB verbunden. Anschließend wurde der Debugger über die USB Schnittstelle mit dem Testsystem verbunden und über die LPCXpresso[5.1.1] Software versucht, ein Programm zu übertragen.

¹Komponententest

²Systemtest, welcher die einzelnen Module und Funktionen miteinander verbindet



(a) CMSIS DAP

(b) Debugger

Abbildung 7.1.: Grafiktest

Ergebnisse

Das Programmieren und Debuggen unter Microsoft[©] Windows[®] 7 funktioniert wie erwartet mit LPCXpresso[5.1.1], unter Microsoft[©] Windows[®] 8.1 hingegen jedoch nicht. Eine Implementierung des Debuggers auf die Steuerungsplatine erscheint als logisch, da der Hardwareaufwand und die Kosten sehr gering erscheinen. Zudem könnte der Debugger bei der Produktion nicht bestückt werden. Die ständige Weiterentwicklung durch das mbed.org Team und der Online Community lässt auf zukunftssichere Benutzung des Debuggers schließen. Zudem ist das HDK als open-source deklariert, wodurch man den Debugger selbst modifizieren kann.

7.1.2. Cortex M3 - USB

Durch die in 2.2.3 geforderte USB Schnittstelle wurde dieser Test durchgeführt. Dabei diente dieser auch der Inbetriebnahme der Steuerplatine und der Überprüfung des externen Frequenzgebers. Anzumerken ist, dass die USB Schnittstelle nur mit Hilfe eines externen Frequenzgebers betrieben werden kann.

Testaufbau und -durchführung

Für diesen Test wurde der in A.1 aufgeführte Code erstellt. Dieser diente auch dem AudioOut Test[7.1.5]. Dabei wurde die Steuerplatine[6.3] mit dem erstellten Code programmiert und über ein USB-micro Kabel mit den PC verbunden.

Ergebnisse

Es stellte sich heraus, dass der gewählte 12 MHz Oszilator nicht kompatibel mit dem LPC1768 Cortex M3 ist. Das System arbeitete, jedoch war die Amplitude des Oszillators zu gering, um von dem M3 Kern erkannt zu werden. Dies veranlasste, den Real Time Clock (RTC) zu entfernen, welches auch nicht zu den gewünschten Ergebnis führte.

Entwicklung eines digitalen standalone Ultraschall-Multigate-Doppler-Systems

Ein Austauschen des 12 MHz Oszillators durch einen 12 MHz Resonator erbrachte schließlich das gewünschte Ergebnis und die Steuerplatine[6.3] wurde erfolgreich von dem TestPC erkannt.



Abbildung 7.2.: USB Audio Test

7.1.3. Display

Dieser Test prüft die integrierten Funktionen der Displaybibliothek. Dies war notwendig, um den Erfolg des erstellten Systemmodells sicherzustellen.

Testaufbau und -durchführung

Für den Test diente das Evalkit OM110443 von NXP (siehe 5.1.2). Hierfür wurde das Kit mit dem Display verbunden, Testfunktionen von Adafruit in ein Programm implementiert und anschließend auf das Evalkit geflashed.

Ergebnisse

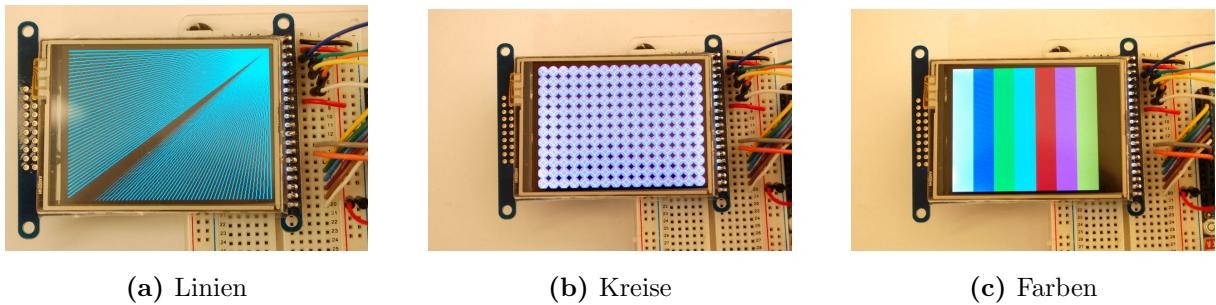


Abbildung 7.3.: Grafiktest

Wie in Abb. 7.3 dargestellt, konnte das Display mit den implementierten Funktionen erfolgreich in Betrieb genommen werden. Dabei stellte sich jedoch heraus, dass das togglen der Pins beschleunigt werden kann. Dabei wurde die Abstrahierung der Pins von C++ Klassen durch ein C++ template ersetzt, welches die Register des M3 Kerns direkt schreibt und somit die Anzahl der Programmzyklen reduziert. Diese template Klasse mit dem Namen „FastIO“ wurde von Igor

Entwicklung eines digitalen standalone Ultraschall-Multigate-Doppler-Systems

Skochinsky auf der mbed Community Seite zu Verfügung gestellt, womit eine Geschwindigkeitssteigerung von 390 Prozent (von 281 ns auf 72 ns) erzielt wird. [27] Diese template Klasse hat überzeugt und wurde fortlaufend in das gesamte System integriert.

7.1.4. Touchfunktion

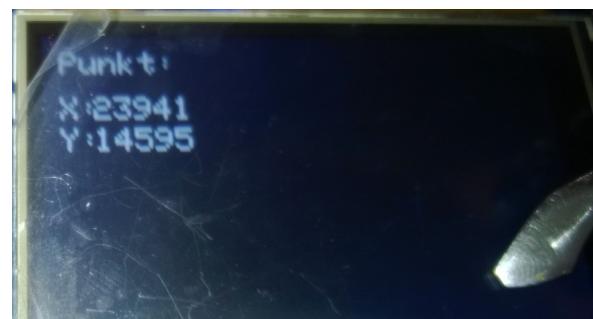
Dieser Test prüft die integrierten Funktionen der Touchbibliothek. Dies war notwendig, um die Benutzerführung durch eine GUI sicherzustellen.

Testaufbau und -durchführung

Für den Test diente das Evalkit OM110443 von NXP (siehe 5.1.2). Hierfür wurde das Kit mit dem Display verbunden und das erstellte Testprogramm implementiert und anschließend auf das Evalkit geflashed. Anschließend konnten die Positionsdaten über das Display abgelesen werden (Abb. 7.4). Dabei wurden Positionsdaten in den Ecken sowie an den Mittelpunkten der Kanten gemessen. Mit diesen konnte eine Mittelwertbildung der Positionsdaten durchgeführt und wie in 7.10 beschrieben skaliert werden.



(a) X-, Y-Position ohne Krafteinwirkung



(b) X-, Y-Position bei Krafteinwirkung

Abbildung 7.4.: Tests - Touchfunktion

Ergebnisse

Durch die gemittelten Messwerte (Tabelle 7.1) konnte eine Skalierung der Werte für die Auflösung von 320x240 durchgeführt werden wodurch eine bedienbare Menuführung (Abb. 6.10) erstellt werden konnte.

Jedoch musste auf die Register des integrierten ADCs direkt zugegriffen werden, da die Analogeingänge von Analog- auf Digitalfunktionen und umgekehrt geschallten werden müssen. Zudem wurde festgestellt, dass die digitalen Pullup-Widerstände aktiviert werden müssen, um eine genaue Messung zu ermöglichen. Dafür wurden der template Klasse „FastIO“ ADC Funktionalitäten hinzugefügt. Es musste beachtet werden, dass bei der Initialisierung des ADC Channels ein delay von $10 \mu s$ vorzusehen ist, damit der Channel eine korrekte Messung durchführen kann.

Tabelle 7.1.: Messwerte X-, Y-Positionen

	Wert
X_{min}	530
X_{max}	1900
Y_{min}	500
Y_{max}	1500

7.1.5. AudioOut

Testaufbau und -durchführung

Für den Funktionstest des Audiosignals wurde ein 8Ω Lautsprecher an den AudioOut Pinheader angeschlossen. Dabei ist auf die Polung zu achten, da sonst die Membran des Lautsprechers nach innen und nicht wie gewollt nach außen schwingt.

Für diesen Test wurde ein Programm (siehe A.1) benutzt, was ein Monosignal mit 48 kHz durch das Audio-Profil des PC's über USB empfangen kann und dies über den DAC des Cortex M3 an den Audio-Verstärker weiterleitet.

Ergebnisse

Das System wurde erfolgreich von Microsoft[©] Windows[®] 7 und 8 als Audio Gerät erkannt und die abgespielte Musik wurde wiedergegeben.

Dabei wurde festgestellt, dass selbst bei inaktiven DAC Ausgang ein nicht definiertes Signal verstärkt wird, welches zu Schädigung des Gehörs führen kann. Dieser Unit-Test wurde zudem als Video auf YouTube.com dokumentiert.

Es empfiehlt sich daher, einen metal oxide semiconductor field-effect transistor (MOSFET) zum einschalten des Audio-Verstärkers zu integrieren. Mit diesem kann auch Energie gespart werden, wenn das Audiosignal nicht benötigt wird. Dies verlängert im mobilen Betrieb die Laufzeit des Systems. Zudem sollte der USB Descriptor für die endgültige Anwendung definiert und angepasst werden.

7.1.6. Datenkommunikation via DMA und SPI

Dieser Test wurde durchgeführt, um die Grundfunktionalität des SPI und des DMA zu testen. Dies ist nötig gewesen, da die Kommunikation mit den CPLD nicht validiert war. Des weiteren kann durch erfolgreiches Bestehen dieses Tests die Fehlersuche bei Inbetriebnahme des gesamten Systems reduziert werden.

Testaufbau und -durchführung

Für den Test diente das Evalkit OM110443 von NXP (siehe 5.1.2). Hierfür wurden die MOSI und MISO Pinouts des Kits miteinander verbunden. Anschließend wurde ein Testprogramm erstellt, was ein definierten Zeichensatz über die MOSI Leitung versendet, in der MISO Leitung empfängt und anschließend über das CDC Profil der USB Schnittstelle an den PC sendet. Über ein Terminal Programm konnten der Test ausgewertet werden.

Dieser Test wurde so durchgeführt, da die Einstellungen für die maximale Geschwindigkeit des SSP mit SPI ermittelt werden mussten. Dies wäre mit der Methodik den Debugger zu benutzen erschwert möglich gewesen, da der DMA Transfer sowie die Geschwindigkeit getestet werden musste.

Ergebnisse

Über diesen zwar schlichten aber effektiven Test konnte schnell die maximale Frequenz des SSP in SPI Konfiguration bestimmt werden. Dabei stellte sich heraus, dass die Herstellerangaben von 1/8 (12 MHz) des Systemtaktes (96 MHz) überschritten werden können. Eine Taktfrequenz von 32 MHz kann fehlerfrei eingesetzt werden, wodurch eine Realisierung des Systemmodells möglich ist.

Abb. 7.5 visualisiert dabei den Test, welche auch unter YouTube.com dokumentiert wurde. Durch diesen können schnell Übertragungsfehler gesichtet werden, da bei einen Übertragungsfehler der dargestellte Text variiert.

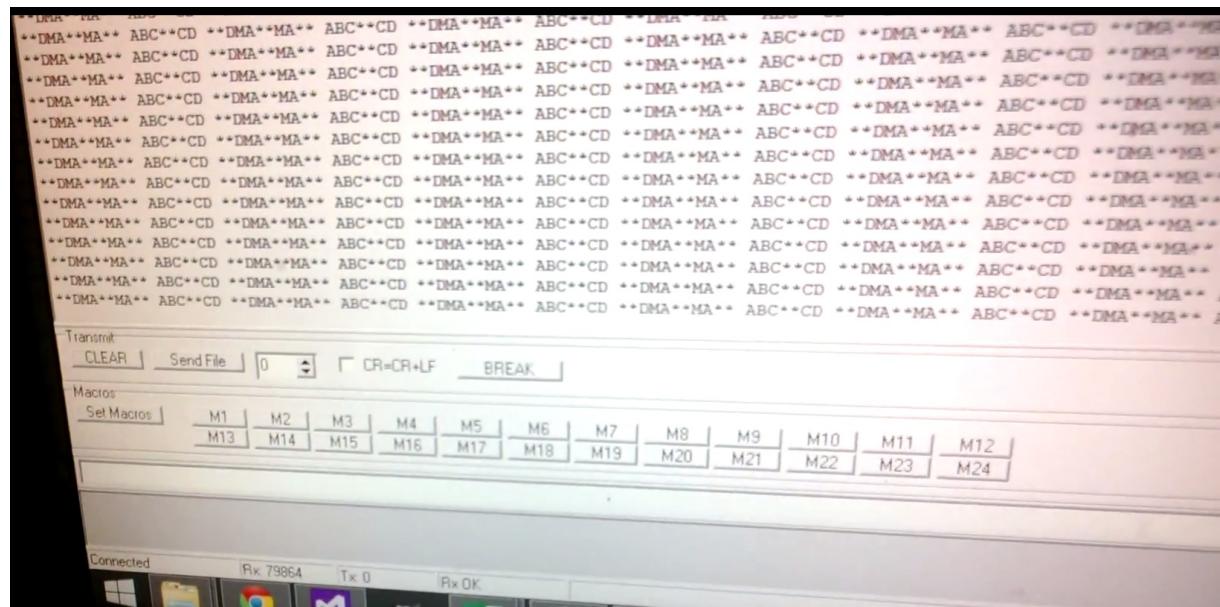


Abbildung 7.5.: Ergebnis DMA & SPI Mode 3

Das Ergebnis dieses Tests verdeutlicht, dass das gewählte Vorgehen durch anstoßen des DMA Transfers via Interrupt Vorteile bietet. Zudem können definiert Datenpakete in beliebiger Menge

transferiert werden und Schleifen Funktionen über Interrupts oder durch sogenannte *linked list items (LLI)* realisiert werden, wodurch die Rechenleistung des M3 Kerns gegen 0 geht.

7.1.7. CPLD - Transmitter

Testaufbau und -durchführung

Der Testaufbau kann durch die Zeitkritische Funktion als Unit-Test nur in Form einer Simulation durchgeführt werden. Dabei wurde ein Testbench (Prüfstand) in der Sprache Verilog erstellt³, wodurch dieser Test mit dessen Ergebnissen reproduzierbar ist.

Ergebnisse

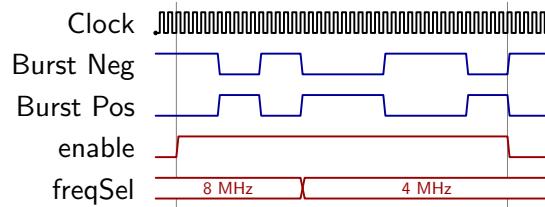


Abbildung 7.6.: Ergebnis Transmitter Steuerung

In Abb. 7.6 erkennt man, dass bei Aktivierung des Transmitters der erste Flankenwechsel der Ausgänge erst nach dem erreichen des Zählerwertes des Frequenzteilers geschieht. Dabei ist zu beachten, dass die Aktivierung bei fallender Flanke des Systemtaktes übernommen wird. Es ist des weiteren zu erkennen, dass die Sollfrequenzen 2, 4 und 8 MHz generiert werden können.

7.1.8. CPLD - Statehandler

Testaufbau und -durchführung

Dieser Test kann wie in Abschnitt 7.1.7 nur durch eine Simulation bestimmt werden. Dabei wurden vier Testbenches⁴ erstellt, welche das Shift-Register für den Empfang der States, den Zähler mit der Vergleichsfunktion, den FIFO sowie das Hauptmodul testet.

Ergebnisse

Die Testbenches wurden erfolgreich durchgeführt und die Funktionalität der Module sichergestellt. Exemplarisch werden nachfolgend 2 dieser Module dargestellt.

In der Abb. 7.7a ist zu erkennen, dass der *Compare* Modul bei erreichen des zu vergleichenden Wertes das Signal equal und bei Überlauf das Signal error erzeugt, wobei letzteres für den Reset des Modules genutzt wird.

³Datei: tb_Peripherie.v - Modul *tb_transmitter*

⁴Datei: tb_StateHandler.v - Module *tb_shifter*, *tb_compare*, *tb_FIFOVar* und *tb_statehandler*

Entwicklung eines digitalen standalone Ultraschall-Multigate-Doppler-Systems

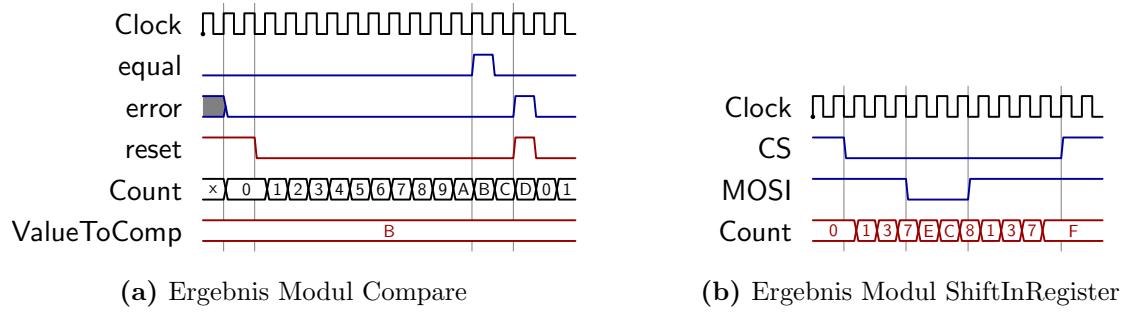


Abbildung 7.7.: Ergebnisse StateHandler

Das Modul *ShiftInRegister* wurde in Abb. 7.7b dargestellt. In diesem ist zu erkennen, dass die Werte bei steigender Flanke übernommen und ein Shift nach Links durchgeführt wird. Der Chip Select muss dabei aktiv (low) sein.

7.1.9. CPLD - Output Interface

Testaufbau und -durchführung

Dieser Test kann wie die Tests in den Abschnitten zuvor nur durch eine Simulation bestimmt werden. Dafür wurde der SPI Verilog Code und der dazu passende Testbench von ElecDude ED [28] zur Verfügung gestellt. In diesem Testbench werden die Empfangs-, Sendefunktion des Verilog Codes validiert.

Ergebnisse

Wie zu erwarten, konnte dieser Test ohne weitere Änderungen des Codes durchgeführt werden.

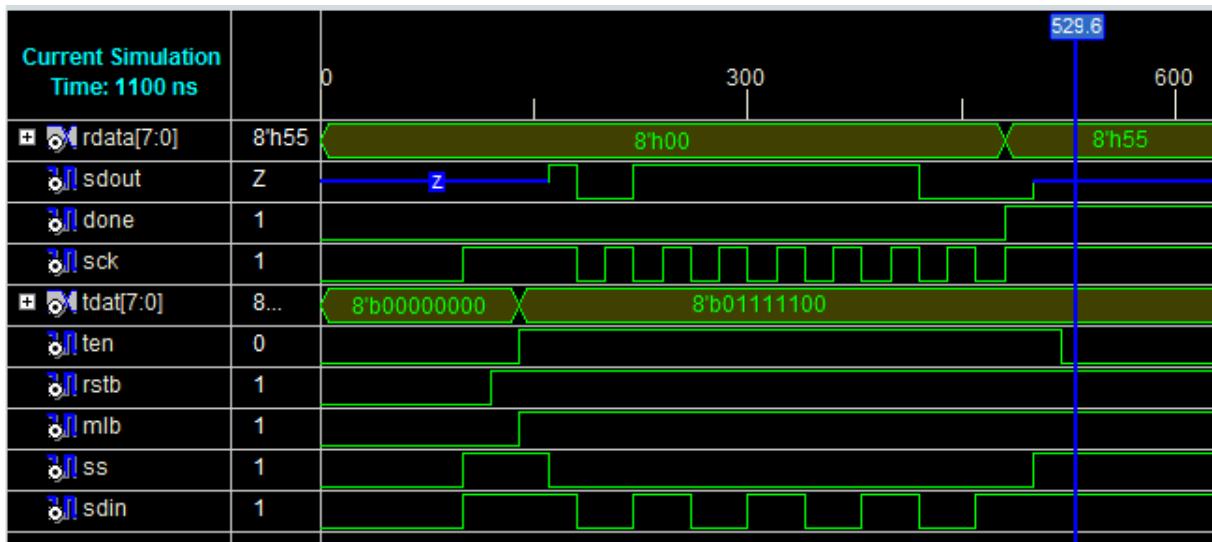


Abbildung 7.8.: Ergebnis CPLD SPI slave Mode 3

In Abb. 7.8 ist ersichtlich, das bei anliegenden SCK und nach Aktivierung des Chip Select (ss) Signales die das Schieberegister arbeitet und einen done Flag (Datenübertragung beendet) nach 8 steigenden SCK Flanken setzt.

Der FIFO des Ausgangsbuffers ist eine von Lattice zur Verfügung gestellter Puffer, wodurch dieser erst im Integrationstest (7.2) getestet wurde.

7.2. Systemtest (Integration-Test)

Nachdem die grundlegenden Funktionalitäten verifiziert wurden, konnte das System in Betrieb genommen werden.

Dabei wurden die für die Messung notwendigen Parameter über den Touchscreen eingestellt und an das CPLD der Messplatine übertragen.

Zunächst wurde die Transmittersteuerung durch Hinzunahme des in Kapitel 5.1.2 aufgeführten Oszilloskop validiert. Dabei konnten die geforderten Trägerfrequenzen (Abb. 7.9), die PRF (Abb. 7.10a) sowie die Bursttiefe (Abb. 7.10b) erfolgreich gemessen werden.

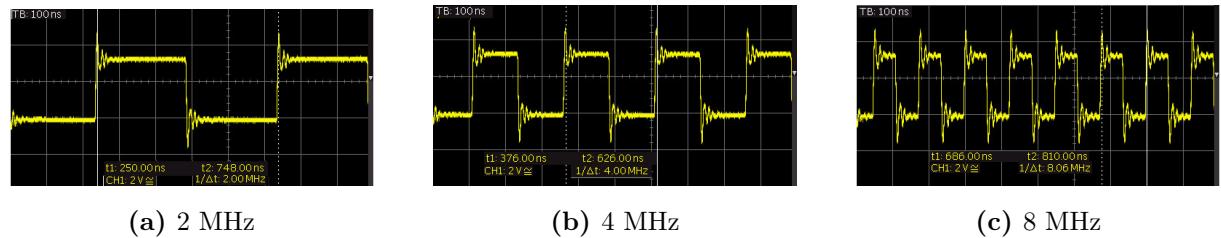


Abbildung 7.9.: Test Trägerfrequenz

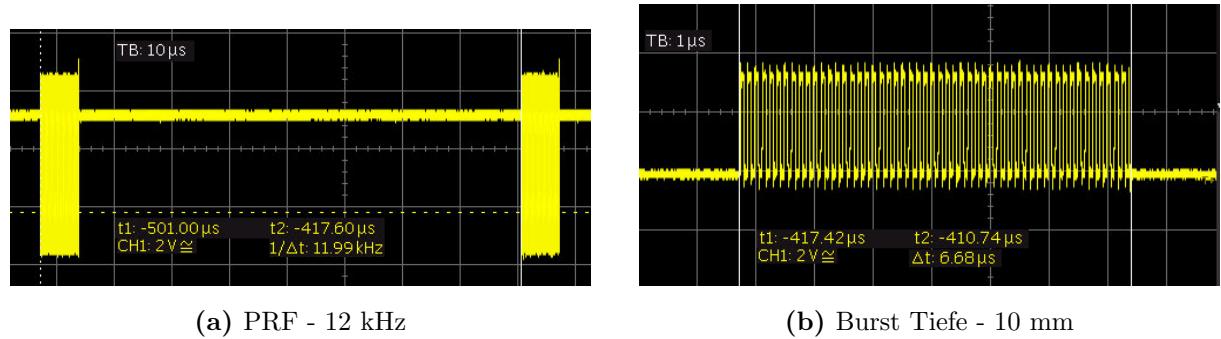


Abbildung 7.10.: Test Burst Tiefe und PRF

Nachdem diese Tests erfolgreich waren, wurden in das CPLD Testzähler implementiert, welche die Daten generieren um den Datentransfer zu validieren. Dabei inkrementiert Ein Zähler von 3 bis 180 wobei der Andere von 180 auf 3 dekrementiert. Dies wurde getan, damit eine schnelle und leichte Möglichkeit zur Darstellung der Generierten Daten zur Verfügung stand.

Eine Visualisierung konnte somit schnell ermöglicht werden, welche die Testdaten auf dem Touchscreen darstellte. Somit kann sichergestellt werden, dass der DMA Transfer über SSP möglich ist.

Entwicklung eines digitalen standalone Ultraschall-Multigate-Doppler-Systems

Jedoch wurde festgestellt, dass der DMA Transfer nicht reaktionsschnell auf die steigende Flanke des *Almost Full* Flags des CPLD reagiert, was zu einen Speicher Überlauf führt. Der Grund für diese Zeitverzögerung liegt dabei an der Routine zur Visualisierung der Daten. Während die Visualisierung durchgeführt wird, kann eine Flanke durch das CPLD erzeugt werden. Durch die Visualisierungsroutine wird jedoch erst nach Beendigung dieser der aufgenommene Interrupt verarbeitet, was ein verspätetes Anstoßen des DMA Transfer zur Folge hat. Diese Fehlfunktion wurde als zweitrangig erachtet, da die Inbetriebnahme der Messplatine eine höhere Priorität erhielt.

Nachdem die eine Darstellung der Testdaten trotz Datenverlust möglich war, wurde die Digitalisierungsfunktion des CPLDs aktiviert. Unter Verwendung des in Kapitel 5.1.2 beschriebenen Funktionsgenerator wurde ein 4 MHz erzeugt, und an den Eingang des Receivers eingekoppelt. Jedoch konnten keine Daten visualisiert werden. Ein Grund dafür war, dass der differenzielle Vorverstärker des Receivers aus nicht nachvollziehbaren Gründen zerstört wurde. Jedoch kann nicht ausgeschlossen werden, dass die programmierte Funktionalität Störungen aufweist. Eine Fehleranalyse mit Überprüfung der verbauten Elemente konnte kein verwendbares Resultat erzielen, wodurch eine vollständige Inbetriebnahme des Systems nicht möglich war.

Bei Durchführung des Integrationstestes wurde gehäuft festgestellt, dass die Touchfunktion optimiert werden musste, um ein besseren Bedienkomfort zu ermöglichen. Dies konnte erfolgreich umgesetzt werden.

Des weiteren konnten Grundlegende Funktionalitäten für die Ansteuerung des Systems über die USB Schnittstelle realisiert werden.

8. Diskussion und Ausblick

8.1. Zusammenfassung der Ergebnisse

Im Rahmen der vorliegenden Arbeit wurde ein digitales standalone Ultraschall-Multigate-Doppler-System entwickelt, welches das Messen von Ultraschall Dopplerschiebefrequenzen erlaubt. Den aus den Messverfahren resultierenden Hämatokritwert in M-Mode und Doppler Spektrogramm darzustellen gelang in den zur Verfügung stehenden Zeitraum nicht. Dafür konnten die grundlegenden Funktionen für die Realisierung des Messverfahrens, sowie der Visualisierung umgesetzt werden. Eine Echtzeitvisualisierung der generierten Daten konnte erfolgreich auf dem Display dargestellt werden, was die Grundlage für die Visualisierung der Graphen bildet.

8.1.1. Messplatine

Eine Ansteuerung des Transmitters konnte erfolgreich mit den geforderten Frequenzen umgesetzt und die Ausgangsamplitude auf 4,2 Vpp erhöht werden. Hingegen konnte eine Integration eines zukauf Strom-Spannungs-Wandlers nicht realisiert werden.

Der Receiver kann mit den geforderten Frequenzen betrieben werden. Es wurde festgestellt, dass bei der zyklischen Aktivierung des ADCs hörbare Frequenzen durch die PRF generiert werden, welche zu einer akustischen Belästigung führen können. Diese Störung konnte durch eine permanente Aktivierung des ADC eliminiert werden. Hingegen konnte eine Validierung des Eingangsverstärkers nicht durchgeführt werden, da dieser durch einen Schaltplanfehler zerstört wurde.

Eine Störeinkopplung der Hochfrequenzsignale konnte erfolgreich durch Entkopplung der Massenflächen mit einem Ferrit verhindert werden.

Die zyklische Abfolge des CPLD durch die Statemachine konnte erfolgreich umgesetzt werden. Es sollten jedoch die Ergebnisse von Kapitel 7.1.7 validiert werden, um ein präzises Ansteuern des Transmitters zu gewährleisten.

Die Demodulierung des Signales wurde implementiert, der Tiefpassfilter hingegen nicht. Die implementierten mathematischen Funktionen wurden dabei erfolgreich simuliert.

Der Datentransfer wurde über SPI Mode 3 realisiert, dabei ist der FIFO Puffer des Ausgangs mit 256x64 Bit für den kontinuierlichen Datentransfer grundsätzlich geeignet. Jedoch ist hierbei darauf zu achten, dass es zu keinen Timing Problemen kommt, da diese nicht automatisch behoben werden können. Daher muss ein Stream realisiert werden, welcher den Puffer weder komplett leert, noch ein Stoppen des Schreibens wegen Überlaufs erzwingt.

8.1.2. Steuerplatine

Der RTC des ARM-Cortex M3 konnte nicht realisiert werden. Dieser beeinträchtigt jedoch die Funktionalität des Systems nicht.

Alle Schnittstellen konnten realisiert werden, wodurch eine Grundfunktionalität gewährleistet wird. Der DMA Transfer über SPI konnte nicht validiert werden, da die endgültig zu transportierenden Daten nicht erstellt werden konnten. Somit war eine Realisierung eines flexiblen Datenstreams nicht möglich. Des weiteren konnten mathematische Funktionen für die Wiedergabe der Signale nicht implementiert werden, wodurch eine Darstellung der Signale nicht möglich ist.

Die Stromversorgung bricht weder bei Versorgung über USB, noch bei Versorgung über die DC Power Jack Buchse ein. Eine Validierung wurde nicht vorgenommen, da die endgültige Leistungsaufnahme nicht realisiert werden kann.

Der Datentransfer über USB wurde durch das Bulk Protokoll implementiert, wodurch auch das Einstellen der Messung über den PC realisiert werden konnte.

8.1.3. Display & Touch

Das Display kann erfolgreich mit den implementierten Funktionen betrieben werden. Die Aktualisierungsgeschwindigkeit des Displays ist für die gewählten Darstellungen ausreichend. Ein Defizit ist die geringe Größe des Displays, wodurch es zu Verwechslungen der Graphischen Darstellung kommen könnte.

Die resistive Touchfunktion konnte des weiteren erfolgreich implementiert werden. Hierbei ist auf die schlechte Justierung des Druckpunktes zu verweisen, wodurch Fehleingaben grundsätzlich möglich sind und durch geeignete Anordnung vermieden werden müssen. Bei der Kalibrierung wurde festgestellt, dass die Widerstandswerte an der Folienecken zu den Mittelpunkten der Folienkanten variieren. Dies wurde durch Mittelwertbildung kompensiert. Die Touchfunktion wird alle 0,2 Sekunden aufgerufen, womit eine gefühlte Reaktionsschnelle des Systems bereitgestellt wird.

Das GUI stellt die benötigten Informationen und Funktionen für das Einstellen der Messung mit Error-Handling bereit, die geforderten Graphen-Darstellungen hingegen nicht, da Auswertungsfunktionen nicht in dem uC implementiert sind.

8.1.4. System

Es wurde grundsätzlich gezeigt, dass das entwickelte System die Anforderungen erfüllen kann. Dies wurde anhand der Validierung grundlegender Funktionen exemplarisch gezeigt. Weitere Optimierungen des Systems sind vor allem bei dem Datentransfer zwischen CPLD und dem ARM-Cortex M3 nötig, da dieser der kritischste Punkt zum erfüllen der Anforderungen dar-

stellt. Softwareanpassungen können durch die verwendeten Techniken leicht realisiert werden, wodurch sich die Validierung des Transmitters, Receivers sowie des Strom-Spannungs-Wandlers als kritische Punkte darstellen.

8.2. Verbesserungspotentiale und Ausblick

8.2.1. Messplatine - Hardware

Die Transmitterschaltung ist für die Erzeugung der Ausgangsamplituden geeignet, wobei ein zukauf Strom-Spannungs-Wandler zu integrieren ist. Ein Mögliches Bauteil wäre ein RF Transformer von Mini-Circuits mit der Bezeichnung ADT8-1T(+) [29], welcher zu Verfügung gestellt werden kann.

Die Eigenschaften des Receivers müssen noch in Betrieb genommen und validiert werden, um Aussagen für die Verwendbarkeit des Receivers zu treffen.

Die Schnittstellen für den Trigger sowie der Sonden müssen definiert werden. BNC Steckverbindungen haben sich in der Messtechnik durchgesetzt, wodurch diese Steckverbindungen für die Sonden empfohlen werden kann. Eine Erkennung, ob eine Ultraschallsonde oder zwei an das System angeschlossen sind wäre von Vorteil, um die Messtechnik zu schützen und den Laufweg (einfach oder zweifach) der Messungen automatisch zu bestimmen oder den Benutzer auf nicht angeschlossene Sonden hinzuweisen.

Die Positionierung des Ferrits sollte mit den vorgesehenen Verbindungen validiert werden, um eine ideale Unterdrückung der Hochfrequenten Störsignale zu erzielen.

Das CPLD mit den Oszillatoren könnte auf die Steuerplatine positioniert werden, wodurch eine Idealisierung der Platinengrößen ermöglicht werden könnte. Zudem können durch den erhöhten Leitungsweg der digitalen Signale eine Abschwächung der Störsignale erzielt werden.

8.2.2. CPLD - Software

In die Software muss ein FIR Filter implementiert werden, damit eine Darstellung der Dopplerschiebefrequenzen ermöglicht werden kann. Zudem muss die Digitalisierung validiert werden, da diese nicht exakt arbeitet. Eine Validierung der Transmitteransteuerung wird empfohlen, um die erforderliche Leistung für das Sendesignal zu generieren und mögliche Fehler zu reduzieren. Eine Implementierung für die Messung statischer Echos ist zwar nicht relevant für die Erfüllung der Hämatokritwertmessung, jedoch kann das System auf anderen Märkten angeboten werden, was zu einen erhöhten Absatz führen kann.

8.2.3. Display

Eine Vergrößerung des Displays von 2,8 auf 3,2 oder 4,3 Zoll würde die Positionierung der Einstellungsmöglichkeiten optimieren. Zudem können bei der Darstellung des Graphen Zusatz-

Entwicklung eines digitalen standalone Ultraschall-Multigate-Doppler-Systems

informationen dargestellt werden. Ein Widescreen mit Verhältnis 16:9 bietet sich an, da eine größere Zeitachse dargestellt werden kann. Hierbei sollte beachtet werden, dass bei Erhöhung der Auflösung der Rechenaufwand des uC steigt.

Die Touchfunktion ist für den Einkaufspreis zufriedenstellend. Wenn möglich sollte jedoch eine kapazitive Touchfunktion realisiert werden, da sich auf den Folien des resistiven Touchscreens Bakterien anreichern können. Durch einen kapazitiven Touchscreen mit Glasscheibe wäre dies nur bedingt möglich. Zudem können gängige Reinigungsmittel zur Desinfektion genutzt werden, was bei den Folien des resistiven Touchscreens eine drastische Erhöhung des Verschleißes mit sich führen würde.

8.2.4. Steuerplatine - Hardware

Für die Optimierung der Stromversorgung muss der gesamte Leistungsbedarf ermittelt werden, sowie die endgültigen Einsatzmöglichkeiten definiert werden.

Ein MOSFET für den Verpolungsschutz mit Sicherung bietet sich als Optimierung der eingesetzten Schutzfunktion der Hardware an, da die verwendeten Schottky Dioden einen zu hohen Leistungsabfall verursachen und das Spannungspotenzial reduzieren. Zudem könnte ein Akkumulator das System mobilisieren, wodurch sich neue Einsatzmöglichkeiten erschließen.

Durch eine statische Nutzung des Systems für Langzeitmessungen hingegen würde sich die Ethernet Schnittstelle des ARM Cortex-M3 anbieten, wodurch die Messdaten an einen Server weitergeleitet werden können.

Die Anordnung der Schnittstellen sollte validiert werden, um eine benutzerfreundlichen Umgang zu gewährleisten.

Für die Benutzung eines größeren Displays empfiehlt sich ein uC mit integrierten LCD Treiber um die Rechenleistung zu reduzieren. Die Rechenleistung des ARM Cortex-M3 zum bewältigen der FFT muss validiert werden. Durch das Ergebnis der Validierung kann entschieden werden, ob eine Leistungssteigerung durch einen M4 Kern mit FPU notwendig ist.

8.2.5. ARM Cortex-M3 - Software

Die Strukturierung der Methoden sollte optimiert werden, da eine Einarbeitung selbst durch die Teilung in mehrere Dateien nicht leicht ist. Zudem muss der DMA Transfer als variablen Stream realisiert werden, da die Datenrate von der PRF abhängig ist und somit einen **Buffer Underrun oder overflow** des CPLDs verursacht werden kann. Zudem müssen Funktionen für die Darstellung der Messungen (Audio & Graph) implementiert werden.

Eine Validierung des USB Descriptors mit Funktionsumfang muss definiert werden, um die geforderten Funktionalitäten über eine definierte Schnittstelle bereitstellen zu können.

8.2.6. Kostenreduzierung

Ein Lattice MachXO CPLD mit 2280 LUTs kostet 17,04€ und der ARM Cortex-M3 von NXP mit Bezeichnung LPC1768 7,18€ pro Stück (Einzelpreis) was eine Summe von 24,22€ entspricht.

Die Funktionen des CPLD können jedoch durch 2 patentierte Techniken¹ von NXP in einen uC (System on Chip) abgebildet werden. Dabei stehen ein ARM Cortex-M3 für 10,15€ und ein DualCore mit M4, FPU und M0 Kern für 16,83€ zu Verfügung, welche zudem auch einen LCD Treiber bereitstellen.

Da das System sich derzeit in einen Prototypen Status befindet und eine Optimierung des Konzeptes zielführend ist, sollte eine Kosteneffizienz des Systems baldmöglichst anzustreben sein.

8.2.7. Mobilität

Derzeit verfügt das System über keine Möglichkeit mobil bzw. ohne Kabel betrieben zu werden. Dies kann durch einen LiPo-Akkumulator, einen StepDown oder LowDrop Regler und der in Abb. 8.1 dargestellten Schaltung erzielt werden.

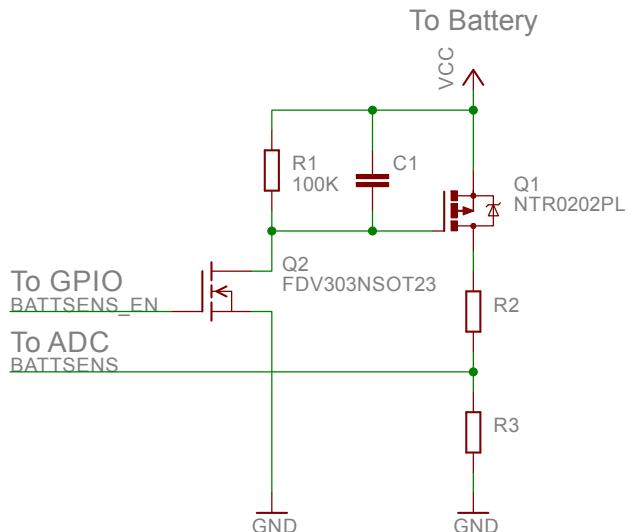


Abbildung 8.1.: Schaltung zur energieeffiziente Spannungsmessung

Dabei sollte beachtet werden, dass das System mindestens 300 mA im Betrieb aufnimmt. Hierbei wird ein externes Ladegerät für die LiPo-Akkumulatoren mit einem Austauschakkumulator empfohlen. Somit kann das System betrieben werden während ein Akkumulator aufgeladen wird. Weiterhin sollte die Peripherie abschaltbar sein. Somit kann der DeepSleep Modus des ARM

¹Serial GPIO für das Buffern des ADC Daten und ein State Configurable Timer (SCT) für die Realisierung der Statemachine

Entwicklung eines digitalen standalone Ultraschall-Multigate-Doppler-Systems

Cortex-M3 in Kombination mit einen Taster zum Einschalten des Systems für einen Energiesparmodus genutzt werden.

Die Widerstände R2 und R3 der Schaltung in Abb.8.1 können mit folgender Formel bestimmt werden.

$$BATTSENS = \frac{R3}{(R2 + R3)} * VCC \quad (8.2.1)$$

Schlusswort

Ich möchte mich bei allen bedanken, die zu dieser Arbeit beigetragen haben. Besonders möchte ich mich bei meinen betreuenden Professor Herr Prof. Dr.-Ing. Rainer Brucher für die Betreuung und die Überlassung des Themas bedanken. Des Weiteren Herrn Schilling-Kästle für die schnelle Umsetzung des PCB Layouts, der Bestellung der BOM und seiner beispiellosen Unterstützung. Hiermit möchte ich mich auch bei meinen Eltern, meiner Schwester und meinen Freunden für Ihre Unterstützung sowie das Korrekturlesen der Arbeit bedanken.

Außerdem möchte ich diese Gelegenheit nutzen, mich bei den Mitarbeitern meiner ehemaligen Abteilung FCM-Ec bei Bosch Rexroth zu bedanken, da Sie mir die Möglichkeit boten, neue Perspektiven und Sichtweisen zu entwickeln, wie sie durch das Studium nicht erschließbar sind.

Literaturverzeichnis

- [1] W. Pschyrembel, *Pschyrembel Klinisches Wörterbuch* -, 261st ed. Berlin: De Gruyter, 2004.
- [2] P. Elter, “Methoden und system zur nichtinvasiven, kontinuierlichen und belastungsfreien blutdruckmessung,” Dissertation, Universität Karlsruhe, März 2001.
- [3] S. Stemplewitz, “Entwicklung eines digitalen ultraschall multigate-dopplers zur intravasalen bestimmung des hämatokritwertes,” Bachlor Thesis, Hochschule Ulm, September 2012.
- [4] P. D. D. Suter, “Ultraschall,” Universität Dortmund, 2006. [Online]. Available: http://e3.physik.uni-dortmund.de/~suter/Vorlesung/Medizinphysik_06/8_Ultraschall.pdf
- [5] ——, “Ultraschall,” Universität Dortmund, 2009. [Online]. Available: http://e3.physik.uni-dortmund.de/~suter/Vorlesung/Medizinphysik_09/8_Ultraschall.pdf
- [6] ——, “Ultraschall,” Universität Dortmund, 2010. [Online]. Available: http://e3.physik.uni-dortmund.de/~suter/Vorlesung/Elektronik_SS10/Skript/12_Sensoren.pdf
- [7] M. Bogdon, M. Thielsch, M. Wagner, and B. Woinowski, “Tablet computing,” Fallstudienarbeit, Hochschule für Ökonomie & Management Neuss, 2013. [Online]. Available: <http://winfwiki.wi-fom.de/images/tmp/f1325679742-1428476813.html>
- [8] mikroElektronika, “Webeartikel über touchscreens,” mikroElektronika, Juli 2009. [Online]. Available: <http://www.mikroe.com/products/view/272/touchscreen-article/>
- [9] A.-L. Geyssel, “Wie funktionieren touchpad und touchscreen?” Welt der Physik, Mai 2011. [Online]. Available: <http://www.weltderphysik.de/thema/hinter-den-dingen/elektronische-geraete/touchpad-und-screen/>
- [10] D. Datenbank, “Touch-displays, die wichtigsten technischen grundlagen zum thema touch,” Luxion AG. [Online]. Available: <http://www.displaydatenbank.de/fachartikel/touch-displays-7.html>
- [11] C. C. GmbH, *EAGLE, EINFACH ANZUWENDENDER GRAPHISCHER LAYOUT-EDITOR, Handbuch Version 6*, 4th ed., CadSoft, CadSoft, Pleiskirchen, 2013.
- [12] lpcpresso support, “Lpcpresso 6,” lpcware, 2013. [Online]. Available: <http://www.lpcware.com/lpcpresso/home>

- [13] T. G. D. Team, “Gnu image manipulation program,” The GIMP Documentation Team, 2013. [Online]. Available: <http://docs.gimp.org/2.8/de/>
- [14] D. Ros, “mbed.org,” mbed.org, 2010. [Online]. Available: <http://mbed.org>
- [15] FLUKE, *PM5139, Function Generator 0.1 mHz - 20 MHz*, 2nd ed., FLUKE, FLUKE, assets.fluke.com, Februar 1999. [Online]. Available: https://courses.engr.illinois.edu/ece445/documents/manuals/PM5139_manual.pdf
- [16] dataTec, *350MHz 2[4] Kanal Digital-Oszilloskop HMO3522 [HMO3524]*, 2nd ed., dataTec, dataTec, www.datatec.de, November 2013. [Online]. Available: http://www.datatec.de/shop/artikelpdf/hmo3524_d.pdf
- [17] A. Devices, *AD8018, 5 V, Rail-to Rail, High-Output Current, xDSL Line Drive Amplifier*, a ed., Analog Devices, Analog Devices, www.analog.com, 2000. [Online]. Available: http://www.analog.com/static/imported-files/data_sheets/AD8018.pdf
- [18] ——, *AD8351, Low Distortion Differential RF/IF Amplifier*, b ed., Analog Devices, Analog Devices, www.analog.com, 2004. [Online]. Available: http://www.analog.com/static/imported-files/data_sheets/AD8351.pdf
- [19] ——, *AD9245, 14-Bit, 20 MSPS, 40 MSPS, 65 MSPS, 80 MSPS, 3 V A/D Converter*, e ed., Analog Devices, Analog Devices, www.analog.com, 2013. [Online]. Available: http://www.analog.com/static/imported-files/data_sheets/AD9245.pdf
- [20] Adafruit, “2.8”18-bit color tft lcd with touchscreen breakout board - ili9325,” adafruit.com. [Online]. Available: <http://www.adafruit.com/products/335>
- [21] T. Instruments, *TPS2114A/TPS2115A, AUTOSWITCHING POWER MUX*, sbvs044f ed., Texas Instruments, Texas Instruments, www.ti.com, Mai 2012. [Online]. Available: <http://www.ti.com/lit/ds/sbvs044f/sbvs044f.pdf>
- [22] ——, “Lm386,” Texas Instruments. [Online]. Available: <http://www.ti.com/lit/ds/symlink/lm386.pdf>
- [23] NXP, *LPC1769/68/67/66/65/64/63, 32-bit ARM Cortex-M3 microcontroller; up to 512 kB flash and 64 kB SRAM with Ethernet, USB 2.0 Host/Device/OTG, CAN*, 9th ed., NXP, NXP, www.nxp.com, Januar 2014. [Online]. Available: http://www.nxp.com/documents/data_sheet/LPC1769_68_67_66_65_64_63.pdf
- [24] ——, *UM10360, LPC176X/5X User manual*, 3rd ed., NXP, NXP, www.nxp.com, Dezember 2013. [Online]. Available: http://www.nxp.com/documents/user_manual/UM10360.pdf

Entwicklung eines digitalen standalone Ultraschall-Multigate-Doppler-Systems

- [25] ——, *AN10913, DSP library for LPC1700 and LPC1300*, 3rd ed., NXP, NXP, www.nxp.com, Juni 2010. [Online]. Available: http://www.nxp.com/documents/application_note/AN10913.pdf
- [26] ——, *AN10934, Using M3 DSP library filter functions*, 1st ed., NXP, NXP, www.nxp.com, Mai 2010. [Online]. Available: http://www.nxp.com/documents/application_note/AN10934.pdf
- [27] I. Skochinsky, “Fastio,” mbed.com. [Online]. Available: <http://mbed.org/users/igorsk/notebook/fast-gpio-with-c-templates/>
- [28] E. ED, *SPI MASTER SLAVE Verilog Code - SPI Working Modes of Operation - Applications - Advantages Disadvantages*, 1st ed., ElecDude ED, ElecDude ED, <http://www.elecdude.com/>, September 2013. [Online]. Available: <http://www.elecdude.com/2013/09/spi-master-slave-verilog-code-spi.html>
- [29] Mini-Circuits, *RF Transformer ADT8-1T+/ADT8-1T*, c ed., Mini-Circuits, Mini-Circuits, www.minicircuits.com, 2013. [Online]. Available: <http://217.34.103.131/pdfs/ADT8-1T.pdf>

A. Quellcode

A.1. Audio Test

Quellcode A.1: Audio Test Code

```
// USBAudio speaker example
#include "mbed.h"
#include "USBAudio.h"

// frequency: 48 kHz
#define FREQ 48000

// 1 channel: mono
#define NB_CHA 1

// length of an audio packet: each ms, we receive 48 * 16bits ->48 * 2 bytes.
// as there is one channel, the length will be 48 * 2 * 1
#define AUDIO_LENGTH_PACKET 48 * 2 * 1

// USBAudio (we just use audio packets received, we don't send audio packets
// to the computer in this example)
USBAudio audio(FREQ, NB_CHA, 8000, 1, 0x7180, 0x7500);

// speaker connected to the AnalogOut output. The audio stream received over
// USB will be sent to the speaker
AnalogOut speaker(p18);

// ticker to send data to the speaker at the good frequency
Ticker tic;

// buffer where will be store one audio packet (LENGTH_AUDIO_PACKET/2 because
// we are storing int16 and not uint8)
int16_t buf[AUDIO_LENGTH_PACKET/2];

// show if an audio packet is available
volatile bool available = false;

// index of the value which will be send to the speaker
int index_buf = 0;

// previous value sent to the speaker
```

Entwicklung eines digitalen standalone Ultraschall-Multigate-Doppler-Systems

```

uint16_t p_val = 0;

// function executed each 1/FREQ s
void tic_handler() {
    float speaker_value;

    if (available) {
        //convert 2 bytes in float
        speaker_value = (float)(buf[index_buf]);

        // speaker_value between 0 and 65535
        speaker_value += 32768.0;

        // adjust according to current volume
        speaker_value *= audio.getVolume();

        // as two bytes has been read, we move the index of two bytes
        index_buf++;

        // if we have read all the buffer, no more data available
        if (index_buf == AUDIO_LENGTH_PACKET/2) {
            index_buf = 0;
            available = false;
        }
    } else speaker_value = p_val;

    p_val = speaker_value;

    // send value to the speaker
    speaker.write_u16((uint16_t)speaker_value);
}

int main() {
    // attach a function executed each 1/FREQ s
    tic.attach_us(tic_handler, 1000000.0/(float)FREQ);

    while (1) {
        // read an audio packet
        audio.read((uint8_t *)buf);
        available = true;
    }
}

```

A.2. Draw Images

Quellcode A.2: DrawImage Code

```

/*Display an 1Bit image on Position X, Y, with width w and height h using the
    frontcolor color and backgroundcolor bgcolor
*
* @param *xbm           Pointer to imageArray (e.g. const static unsigned char
    xyz[] = {...}; ref to GIMP)
* @param x               Start X Position
* @param y               Start Y Position
* @param w               width of image
* @param h               height of image
* @param color           Color of image
* @param drawbg          performance boost with drawbg = false
* @param bgcolor         (optional) backgroundcolor of image, default = Black
*/
void Draw1BitImage(const unsigned char *xbm, unsigned short x, unsigned short
y, unsigned short w, unsigned short h, unsigned short color, bool drawbg =
false, unsigned short bgcolor = Black){
char D;
for(uint16_t zeile=0 ; zeile < h ; zeile++){
    for(uint16_t spalte=0 ; spalte<(w/8) ; spalte++){
        for(int i=0;i<=7;i++){
            D = *(xbm+(zeile*w/8)+spalte);
            if(D & 1<<i) Adafruit_TFT_0x932x::drawPixel(x+i+(spalte*8),
                y+zeile, color);
            else if(drawbg) Adafruit_TFT_0x932x::drawPixel(x+i+(spalte*8),
                y+zeile, bgcolor);
        }
    }
}
delete &D;
}

```

B. Datenträger Inhalt

B.1. Dokumentation

- Thesis in PDF
- Thesis in L^AT_EX

B.2. Hardware

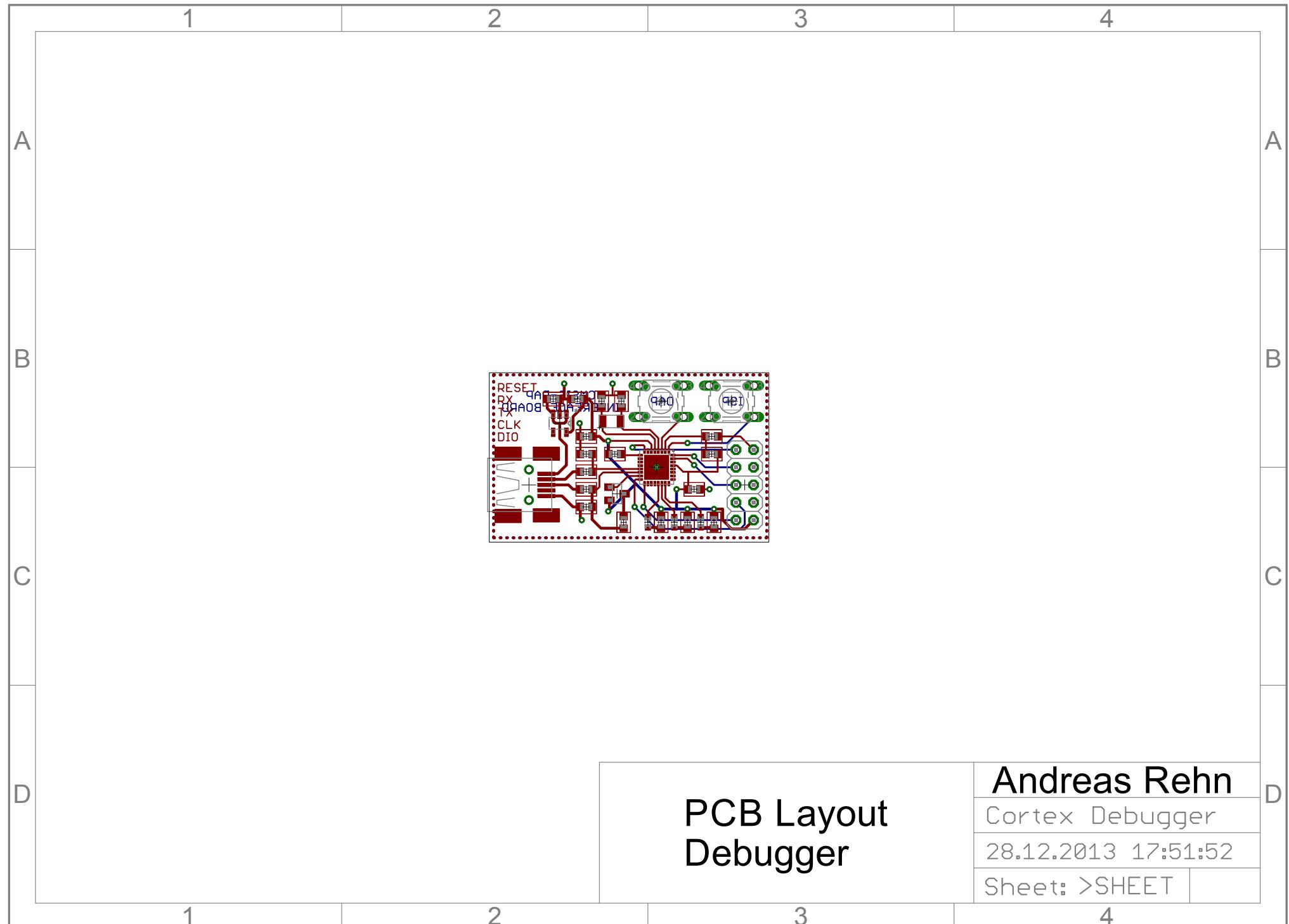
- Eagle Files
 - Bibliothek
 - Debugger
 - Ultraschall Doppler System
- Bill of Materials

B.3. Software

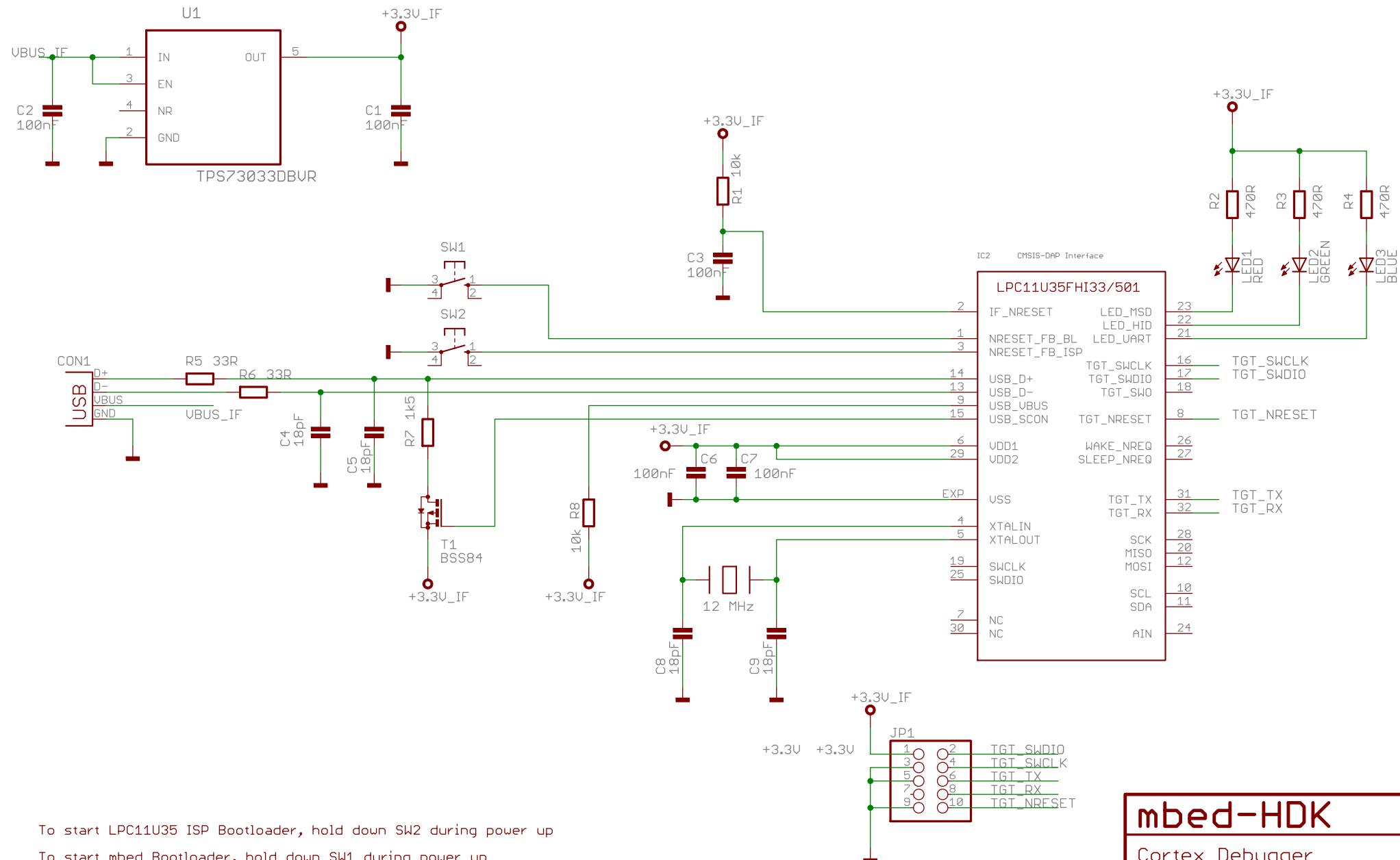
- CPLD - Verilog Files
 - Transmitter
 - EventHandler (Statemaschine)
 - SPI (Interface)
 - Receiver
 - Testbench Transmitter
 - Testbench EventHandler
 - Testbench SPI
 - Testbench Receiver
- ARM Cortex M3 - NXP LPC1768 C++ Projekt
 - Bibliothek TFT & Touch
 - Bibliothek UltrasonicDoppler
 - function Hauptroutine

C. Schaltpläne und Layouts

C.1. Cortex Debugger



CMSIS-DAP Interface



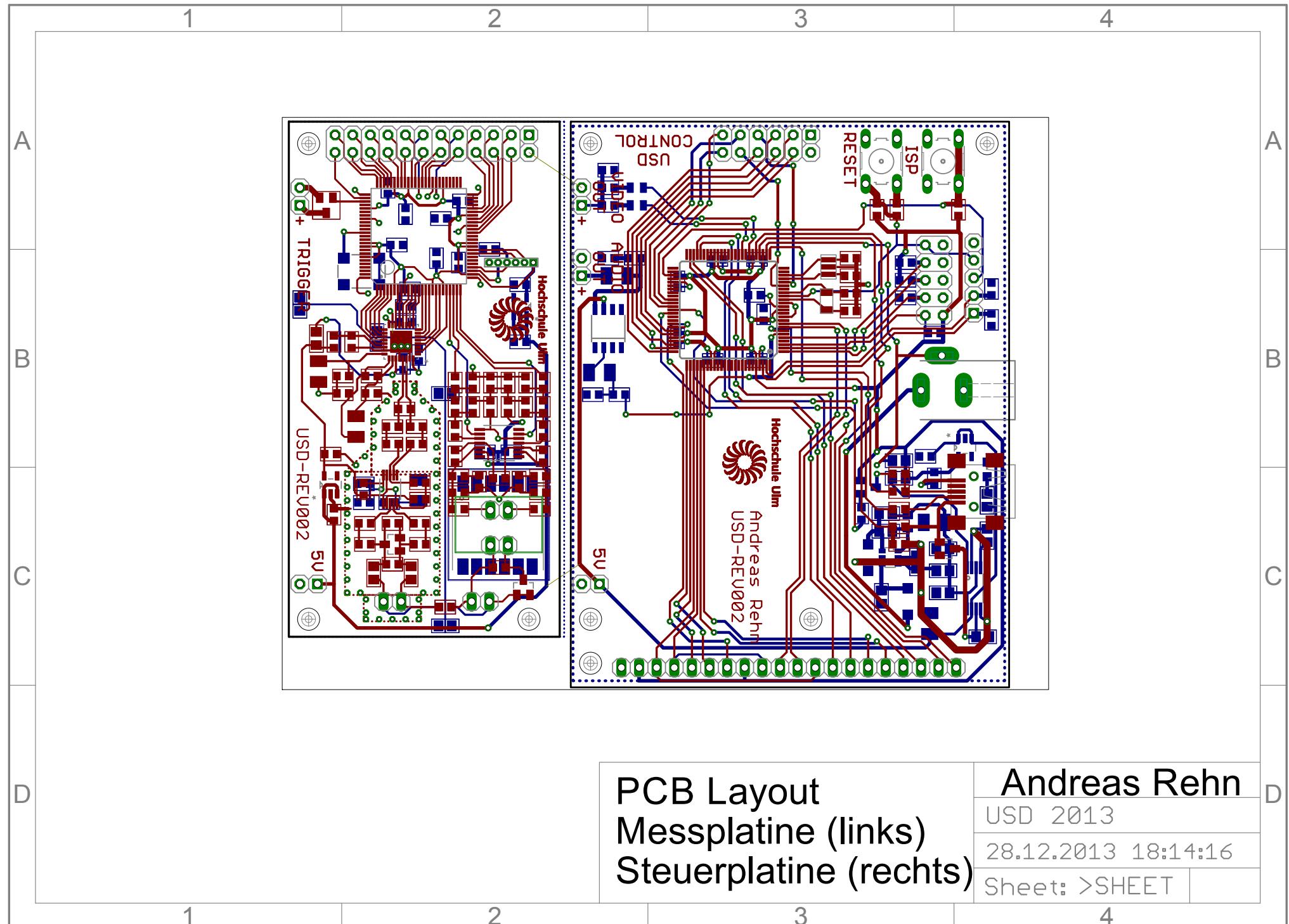
mbed-HDK

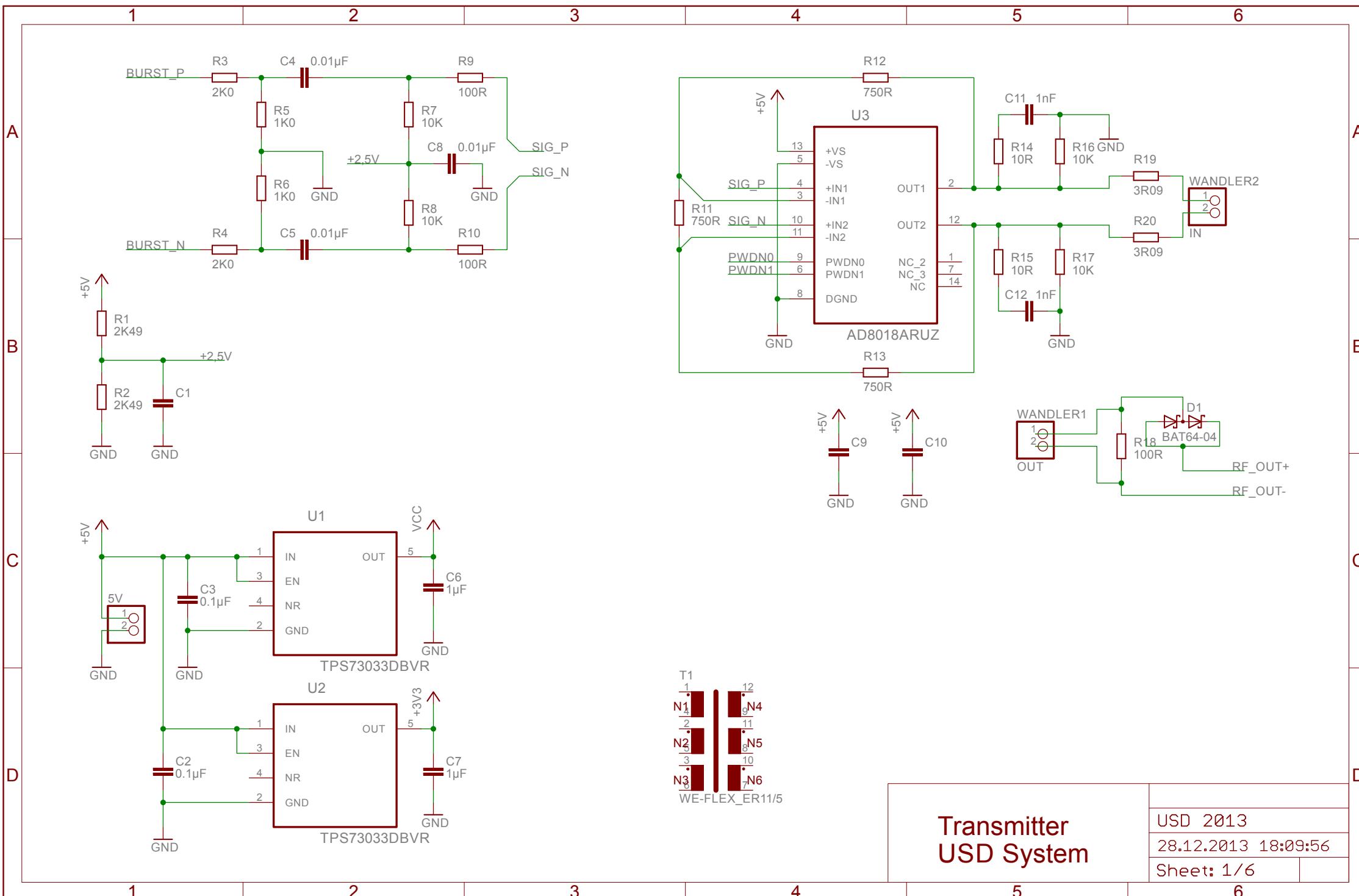
Cortex Debugger

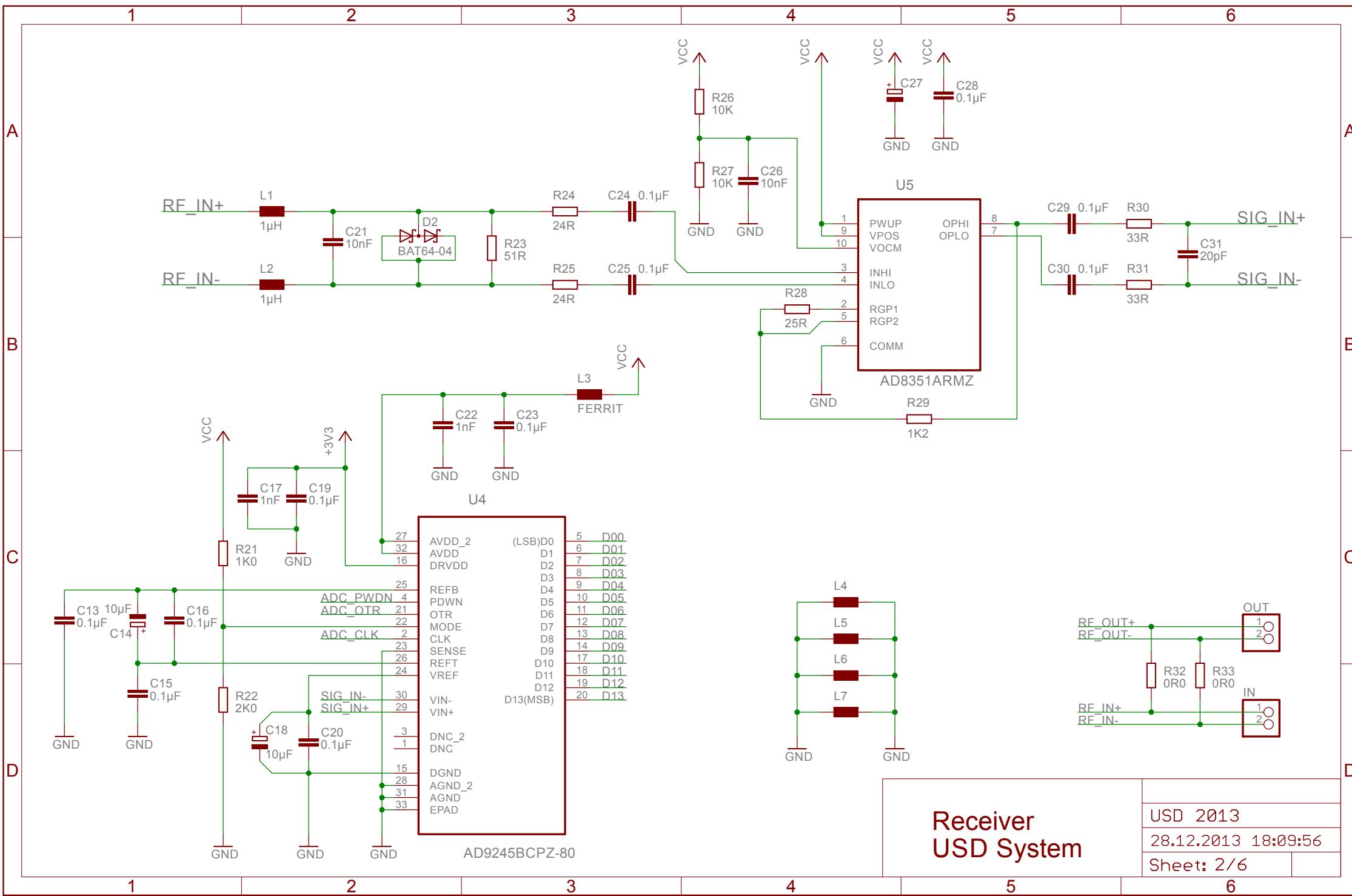
28.12.2013 17:59:23

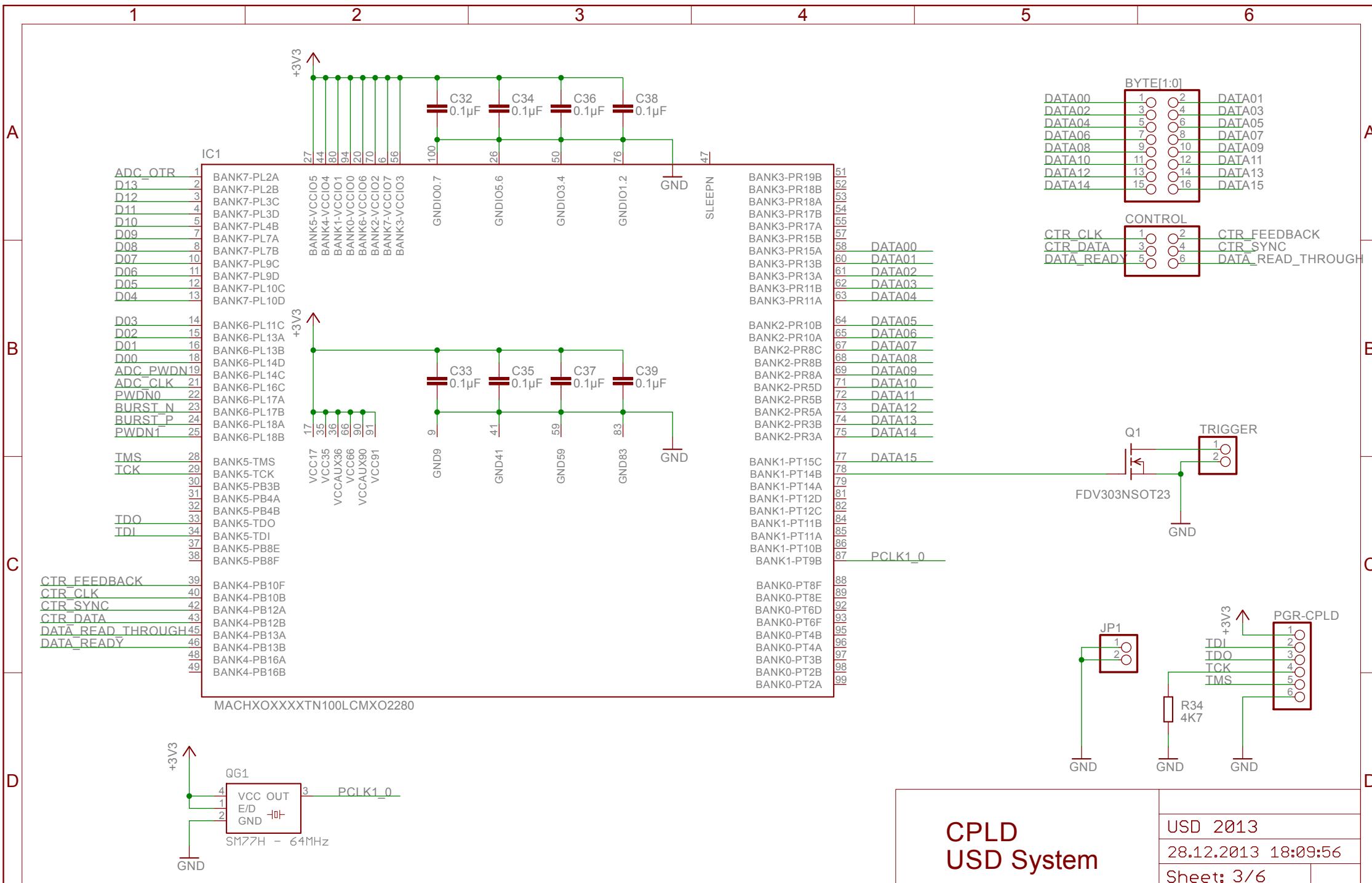
1/1

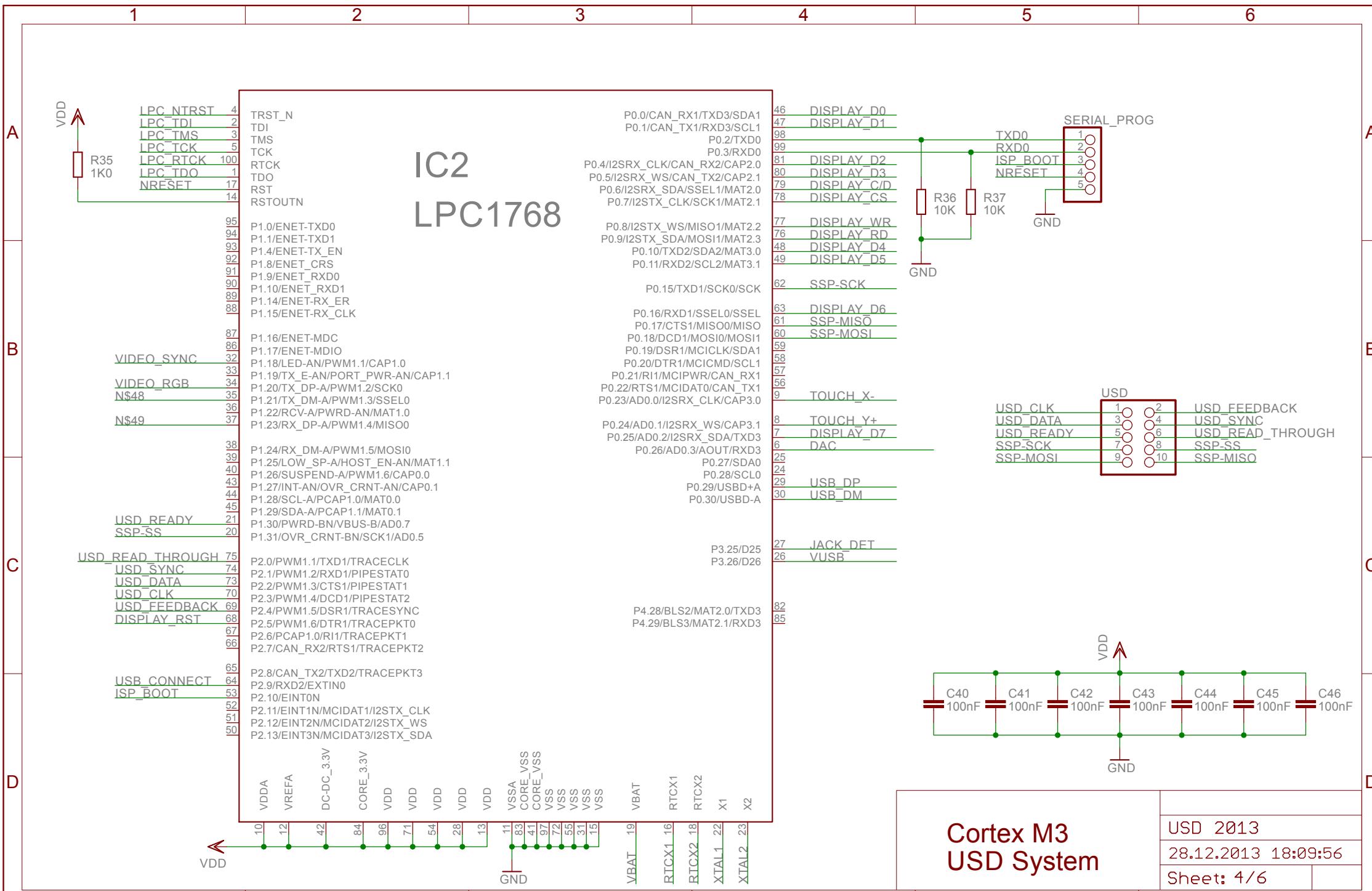
C.2. Ultraschall Doppler System





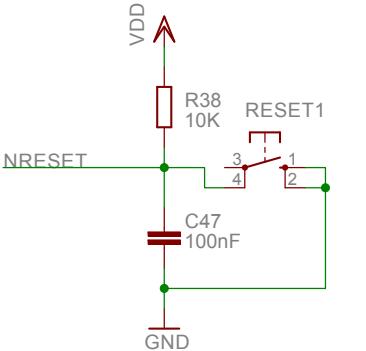




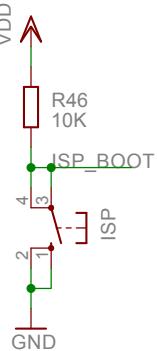


1 2 3 4 5 6

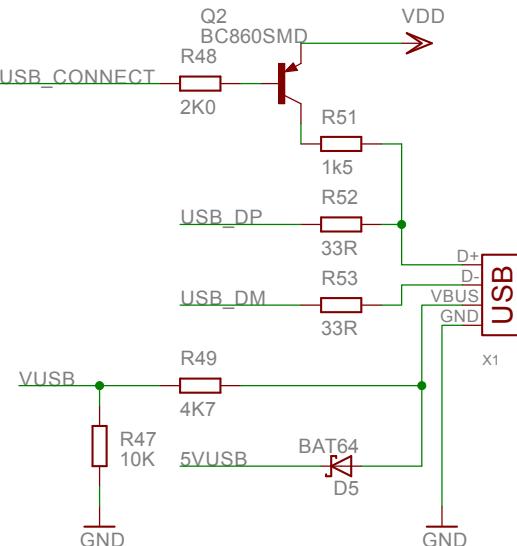
RESET SWITCH



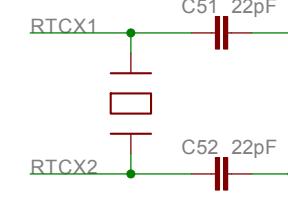
ISP SWITCH



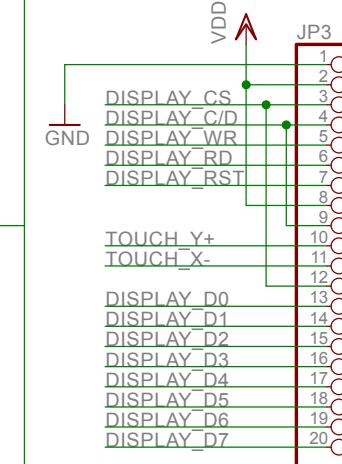
USB



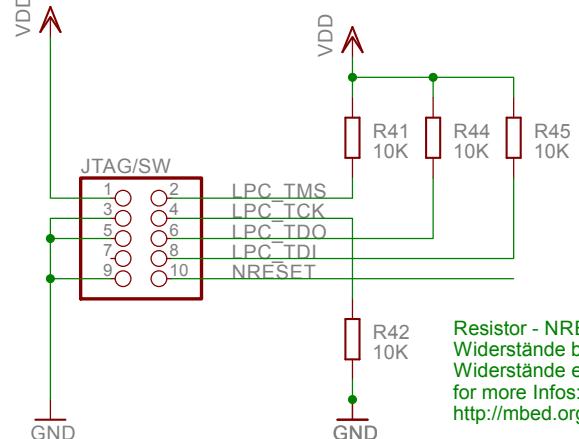
**RTC @
32.768 kHz**



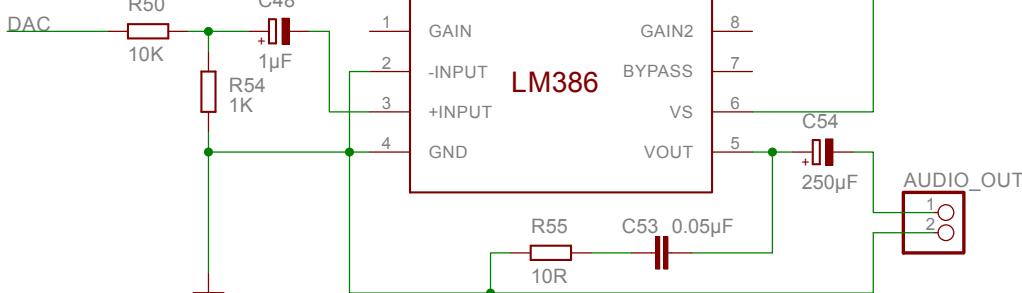
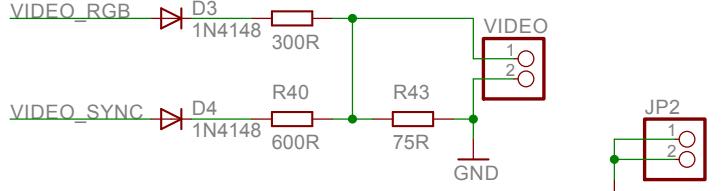
**2.8" TFT
inc. TOUCH**



JTAG/SWD



Resistor - NRESET bei MBED -> On RESET SWITCH
Widerstände bei MBED Config...
Widerstände entfallen bei LPC Link
for more Infos:
<http://mbed.org/users/chris/notebook/prototype-to-hardware/>



**Pherepherie Cortex
USD System**

USD 2013
28.12.2013 18:09:56
Sheet: 5/6

1 2 3 4 5 6

A

B

C

D

