



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorthesis

Niklas Kölln

Entwicklung eines Ultraschallsensors mit
Objektklassifizierung

Niklas Kölln
Entwicklung eines Ultraschallsensors mit
Objektklassifizierung

Bachelorthesis eingereicht im Rahmen der Bachelorprüfung
im Studiengang Mechatronik
an der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. rer. nat. Rasmus Rettig
Zweitgutachter : Prof. Dr.Ing. Lutz Leutelt

Abgegeben am 25. Mai 2015

Niklas Kölln

Thema der Bachelorthesis

Entwicklung eines Ultraschallsensors mit Objektklassifizierung

Stichworte

Ultraschall, Objektklassifizierung, Mikrocontroller, Arduino DUE

Kurzzusammenfassung

Diese Bachelorthesis beschreibt die Entwicklung eines Ultraschallsensors mit Objektklassifizierung für den Einsatz als Fahrerassistenzsystem im Automobilbereich. Die Arbeit umfasst neben einer analogen Schaltungsentwicklung den Entwurf einer Software für einen Mikrocontroller.

Niklas Kölln

Title of the paper

Development of an Ultrasound-sensor with objectclassification

Keywords

ultrasound, objectclassification, microcontroller, Arduino DUE

Abstract

This bachelor thesis describes the development of an ultrasonic sensor with object classification applied to a driver assistance system in the automotive sector. In addition to an analog circuit design the thesis includes the development of a microcontroller software.

Inhaltsverzeichnis

Tabellenverzeichnis	6
Abbildungsverzeichnis	7
1. Einführung	9
1.1. Fahrerassistenzsysteme	9
1.2. Aufgabenstellung	11
1.3. Stand der Technik	11
1.3.1. Analog Devices CN0343	12
1.3.2. MAXIM MAXQ7667	13
1.3.3. HC-SR04	14
2. Grundlagen	15
2.1. Systeme zur Objektklassifizierung	15
2.2. Ultraschall	16
3. Konzept und Implementierung	19
3.1. Arduino DUE Mikrocontrollerboard	19
3.2. Charakterisierung von Ultraschallwandlern	20
3.2.1. Prowave 400Sx160	23
3.2.2. Murata MA40MF14-0B	24
3.3. Analoge Schaltungsentwicklung	25
3.3.1. Eingangverstärkerschaltung	25
3.3.2. Sendeschaltung	33
3.4. Gesamtsystem	35
3.5. Parameter	38
3.5.1. ADC	38
3.5.2. Aufnahmedauer	39
3.5.3. PWM-Frequenz	39
3.5.4. Sendesignallänge	40
4. Objektklassifizierung mittels Ultraschall	41
4.1. Methode	41
4.2. Umsetzung/Aufnahme der Referenzsignale	44

4.3. Korrelationsanalyse mit Scilab	49
4.3.1. Objekt 1	50
4.3.2. Objekt 2	54
4.3.3. Objekt 3	58
4.3.4. Abstandsmessung	62
5. Umsetzung in Software	63
6. Abschließender Systemtest	66
7. Zusammenfassung und Ausblick	67
A. Quellcode	69
A.1. Objectclassification	69
A.2. getDistance	75
A.3. receive	76
A.4. getTemperature	77
A.5. sendBurst	78
A.6. Initialisierung	78
Literaturverzeichnis	80

Tabellenverzeichnis

1.1. Umfeldsensorik und dessen Einsatzgebiete in FAS	10
2.1. Ultraschalleinteilung	16
3.1. Resonanzfrequenzmessung Prowave 400Sx160	23
3.2. Resonanzfrequenzmessung Typ Murata MA40MF14-0B	24
6.1. Ergebnisse des Systemtests	66

Abbildungsverzeichnis

1.1. Erfassungsbereich Umfeldsensorik [Rei10]	10
1.2. Prinzip "Time Of Flight"	11
1.3. Analog Devices Eval-Board CN0343	12
1.4. MAXQ7667	13
1.5. MAXQ7667 GUI	13
1.6. HC-SR04	14
2.1. c_{Luft} in Abhängigkeit der Temperatur	18
3.1. Arduino DUE	19
3.2. Ersatzschaltbild piezoel. US-Wandler	20
3.3. Amplitudengang US-Wandler Prowave 400ST160	22
3.4. 400Sx160	23
3.5. MA40MF14-0B	24
3.6. HC-SR04 Verstärkerschaltung	25
3.7. aktiver Hochpass	26
3.8. Simulation Frequenzgang 1. Hochpassstufe	27
3.9. Messung Frequenzgang 1. Hochpassstufe	27
3.10. Mehrfach rückgekoppelter Bandpass	28
3.11. Simulation Frequenzgang Bandpass	29
3.12. Messung Frequenzgang Bandpass	29
3.13. Simulation Frequenzgang angepasster Bandpass	30
3.14. Messung Frequenzgang angepasster Bandpass	30
3.15. Simulation Frequenzgang zweite Hochpassstufe	31
3.16. Messung Frequenzgang zweite Hochpassstufe	31
3.17. Empfangsverstärkerschaltung	32
3.18. Simulation komb. Verstärkerstufen angepasst	33
3.19. Messung komb. Verstärkerstufen angepasst	33
3.20. HC-SR04 Verstärkerschaltung	34
3.21. Sendeschaltung mit Trafo	35
3.22. Blockschaltbild des Sensorsystems	36
3.23. Schaltbild Gesamtsystem	37
3.24. Ultraschallmessung	39

4.1. Foto Objekt 1	46
4.2. Signalverlauf und extr. Echo Objekt 1	46
4.3. Foto Objekt 2	47
4.4. Signalverlauf und extr. Echo Objekt 2	47
4.5. Foto Objekt 3	48
4.6. Signalverlauf und extr. Echo Objekt 3	48
4.7. Messsignale	49
4.8. Korrelationen Objekt 1 mit Referenzen	50
4.9. Korrelationen Objekt 2 mit Referenzen	54
4.10. Korrelationen Objekt 3 mit Referenzen	58
5.1. Ausgabe serieller Monitor	64
5.2. Aufrufhierarchie	64
5.3. Ablaufplan der Funktion "objectclassification"	65

1. Einführung

Ultraschallsensoren finden Anwendung in sämtlichen Bereichen der Industrie. Sie werden eingesetzt, um auch unter widrigen Bedingungen zuverlässig Objekte zu detektieren oder Abstände zu messen. Sie sind anderen Sensoren häufig überlegen, da ihr Messprinzip in nahezu jeder Umgebung funktioniert. Die Ausarbeitung dieser Arbeit beschäftigt sich mit der Rolle von Ultraschallsensoren im Automobilbereich. Dort kommen sie der besonderen Rolle der Nutzung in Fahrerassistenzsystemen zu.

1.1. Fahrerassistenzsysteme

Fahrerassistenzsysteme (FAS) gewinnen durch die fortwährende Zunahme des Straßenverkehrs zunehmend an Bedeutung für die Sicherheit und den Komfort der Verkehrsteilnehmer. Sie wirken unterstützend im Erkennen kritischer oder gefährlicher Situationen und helfen dem Fahrer ggf. durch Warnungen oder direkten Eingriff in die Fahrzeugführung, Unfälle zu vermeiden. Für die Erkennung solcher potentiell gefährlichen Situationen nutzen FAS intelligente Sensorik, mit der das Umfeld des Fahrzeugs überwacht wird.

Um das Umfeld in seiner Gesamtheit erfassen zu können, werden unterschiedliche Sensoren eingesetzt, die sich jeweils für die Überwachung eines bestimmten Bereiches eignen. [Abbildung 1.1](#) zeigt den Erfassungsbereich unterschiedlicher Sensorsysteme.

[[Sie15](#)] [[Rei10](#)]

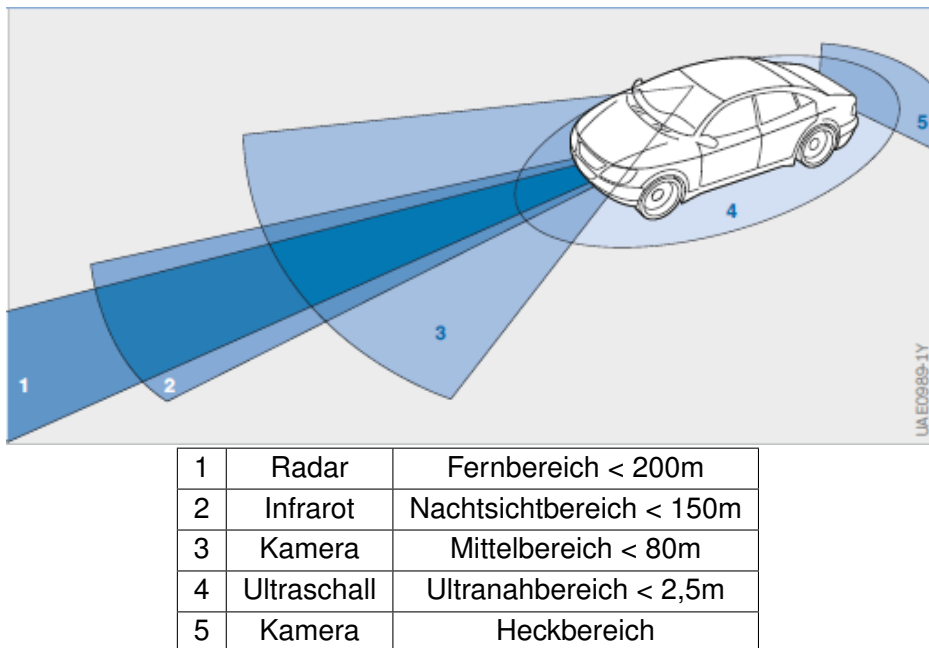


Abbildung 1.1.: Erfassungsbereich Umfoldsensoren [Rei10]

Sensorik	Fahrerassistenzsystem
Ultraschall	- Parkassistent (autom. Einparken) - Einparkhilfe (Abstandswarnung)
Radar	- Stauassistent (autom. Bremsen und Anfahren) - Abstandswarnung - etc.
Kamera	- Fußgängererkennung - Rückfahrkamera - Autom. - Verkehrsschilderkennung - Spurassistent (auto. Spurhalten) - etc.
Infrarotkamera	- Nachtsichtassistent

Tabelle 1.1.: Umfoldsensoren und dessen Einsatzgebiete in FAS

Table 1.1 gives an overview of the FAS and the respective sensor. Ultrasonic sensors are used for the monitoring of the so-called ultrasonic range. All functions of this FAS are based on the measurement of distances. If the vehicle crosses a

Mindestabstand zu einem Hindernis z.B. beim Einparken oder Rangieren, wird der Fahrer in der Regel durch visuelle und/oder akustische Hinweise vor einer Kollision gewarnt. Des Weiteren ist es möglich, die Größe einer Parklücke im Vorbeifahren auszumessen und daraufhin das Fahrzeug automatisch einparken zu lassen. Das Prinzip der Abstandsmessung basiert auf einer sog. "Time Of Flight"-Messung (TOF). Der Sensor sendet ein Ultraschallsignal aus. Trifft dieses auf ein Hindernis, so wird es von diesem reflektiert. Nach einer gewissen Zeit trifft das Ultraschallecho wieder beim Sensor ein und wird von diesem detektiert. Die Zeit zwischen dem Aussenden und Wiedereintreffen des Ultraschallsignals wird von dem Sensor gemessen und in einen Abstand umgerechnet. Die [Abbildung 1.2](#) zeigt das Prinzip einer "Time Of Flight" Messung.

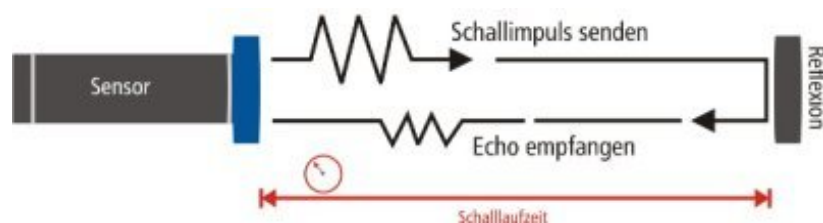


Abbildung 1.2.: Prinzip "Time Of Flight"
[Die14]

1.2. Aufgabenstellung

Die Aufgabe dieser Bachelorarbeit besteht darin, einen Ultraschallsensor im Hinblick auf die Nutzung als Fahrerassistenzsystem zu entwickeln, der neben der Abstandsermittlung mittels TOF-Messung in der Lage ist, unterschiedliche Objekte anhand des reflektierten Ultraschallechos zu klassifizieren. Die Aufgabe umfasst neben der Hardwareentwicklung zur Ansteuerung des Ultraschallsensors und Aufbereitung des empfangenen Signals, die Erstellung einer Software für die digitale Signalverarbeitung auf einem Mikrocontroller. Als Basis der Entwicklung dienen kommerziell erhältliche Ultraschall-Sende- und Empfängerkomponenten sowie das Mikrocontrollerboard Arduino DUE. [Ard15]

1.3. Stand der Technik

Im Rahmen dieser Arbeit wurden mehrere Evaluations-Boards von Ultraschall-Distanzmesssystemen, die den heutigen Stand der Technik darstellen auf ihren Funktionsumfang und ihre Besonderheiten untersucht. In diesem Abschnitt werden sie anhand eines kurzen Steckbriefes vorgestellt.

1.3.1. Analog Devices CN0343

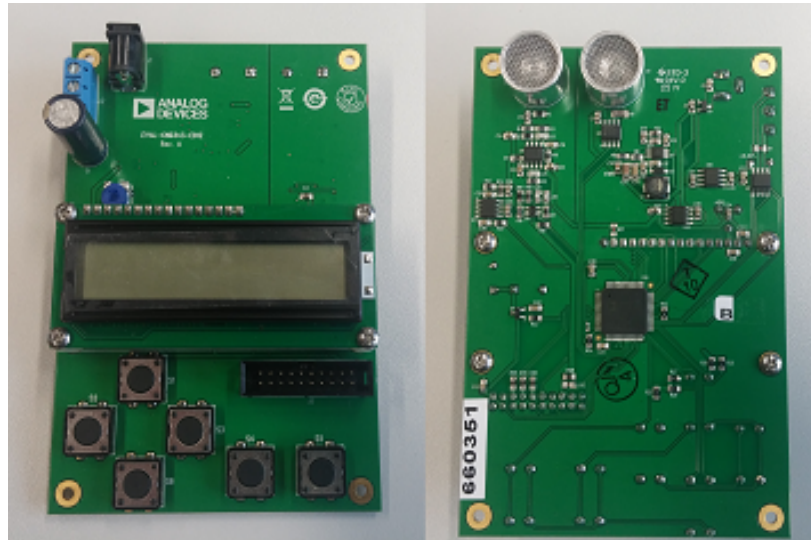


Abbildung 1.3.: Analog Devices Eval-Board CN0343

Das CN0343 Evaluation-Board von Analog Devices [Ana15] ist ein vollständig autarker Ultraschall Distanzsensor auf Mikrocontrollerbasis. Sende- und Empfangspfad sind getrennt ausgeführt. Der Sensor erzeugt ein Ultraschallsignal mit einer Frequenz von 40 kHz. Der Zeitpunkt des Eintreffens des Echos über wird über einen empfindlichen Fensterkomparator ermittelt. Die Signalverarbeitung erfolgt rein analog. Über- oder unterschreitet die Spannung am Empfänger einen bestimmten Schwellenwert, wird die TOF-Messung gestoppt und der Abstand auf einem Microcontroller berechnet.

Um die Abstandsberechnung möglichst genau ausführen zu können, misst der Sensor parallel zur Abstandsmessung die Temperatur der Umgebungsluft und passt die Berechnung entsprechend an. Das Evaluationsboard verfügt über ein LCD-Display auf dem der gemessene Abstand und die Temperatur angezeigt werden. Der Messbereich beschränkt sich auf 50cm bis 10m mit einer Auflösung von 2cm.

1.3.2. MAXIM MAXQ7667



Abbildung 1.4.: MAXQ7667

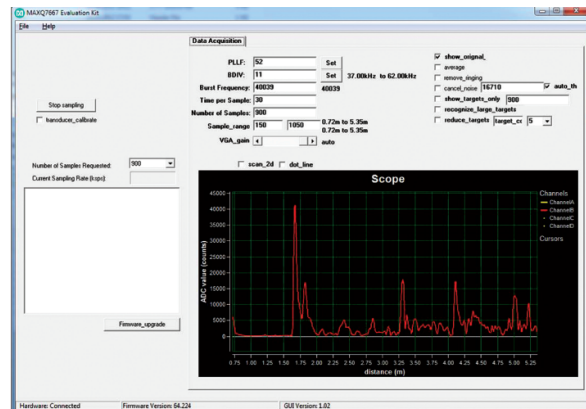


Abbildung 1.5.: MAXQ7667 GUI

Das MAXQ7667 von Maxim Devices [Max] ist ein mikrocontrollerbasiertes Ultraschall-Messsystem, dessen Einsatzgebiet den Automobilbereich ebenso abdeckt wie die Prozessindustrie und die Automatisierungstechnik.

Die TOF-Messung kann wahlweise mit einem kombinierten oder getrennt ausgeführten Sende- und Empfangspfad durchgeführt werden. Zum Anregen des Sendewandlers wird eine Transformator-Verstärkerschaltung verwendet. Das empfangene analoge Echosignal wird mit einem 16 Bit ADC in ein digitales Signal gewandelt. Bei dem MAXQ7667 handelt es sich um einen sog. Smart-Sensor [HS11] d.h. er übernimmt die komplette Signalaufbereitung und Signalverarbeitung auf einem integrierten digitalen Signalprozessor. Der Sensor ist in der Lage, Sender- und Empfängerparameter selbstständig optimal auf die Gegebenheiten anzupassen. So kann die Sendefrequenz in einem Bereich von 25 kHz bis 100 kHz variiert werden, sowie die Verstärkung des Empfängers in Abhängigkeit des Abstandes des zu messenden Objektes erhöht oder verringert werden. Dadurch ist es möglich, mit einer großen Verstärkung des Eingangssignals, sowohl Abstände von Objekten in großer Entfernung zu messen, als auch das Übersteuern des Eingangs durch Echos von nahen Objekten durch Verringern der Verstärkung zu verhindern. Des Weiteren ist der Sensor in der Lage, Abstände mehrerer Objekte mit einer Messung zu erfassen. Über das mitgelieferte sog. General User Interface (GUI) werden die aufbereiteten Ultraschallechos grafisch als Signalamplitude über dem Abstand dargestellt. [Abbildung 1.5](#)

1.3.3. HC-SR04



Abbildung 1.6.: HC-SR04

Der HC-SR04 ist ein speziell für die Arduino-Plattform entwickeltes Ultraschallsensormodul. Sende- und Empfangspfad sind getrennt ausgeführt. Die Messung wird mit dem verbundenen Arduino Mikrocontroller durch eine fallende Flanke am Triggereingang des HC-SR04 gestartet. Ein auf dem Modul vorhandener Mikrocontroller sendet daraufhin ein $200\ \mu\text{s}$ langes $40\ \text{kHz}$ Pulweitenmoduliertes Signal (PWM Signal) und setzt einen digitalen Ausgang auf HIGH. Das empfangene Ultraschallecho wird durch eine analoge Verstärkerschaltung verstärkt und anschließend mit einer Referenzspannung verglichen. Ist die Spannung des Echos größer als die Referenzspannung, wird der digitale Ausgang zurück auf LOW gesetzt. Auf dem Arduino Mikrocontroller wird die Dauer des HIGH, d.h. die Zeit zwischen Senden und Empfangen des Signals ausgewertet und in einen Abstand umgerechnet.

2. Grundlagen

In diesem Kapitel werden die theoretischen Grundlagen, auf denen die Entwicklung des Sensorsystems basiert, erläutert.

2.1. Systeme zur Objektklassifizierung

Sensorsysteme zum Erkennen und Klassifizieren von Objekten können entsprechend ihrer Komplexität in drei Kategorien unterteilt werden: *Bildgebende Systeme*, *objektselektive Systeme* und *Inspektionssysteme*.

Bildgebende Systeme extrahieren aus den empfangenen Signalen die Informationen, die nötig sind um ein möglichst genaues Abbild der Objektgeometrie zu rekonstruieren. Die Klassifizierung des Objektes erfolgt anhand dieses generierten Abbildes durch einen Beobachter oder durch bildauswertende Systeme.

Objektselektive Systeme prüfen die Empfangssignale auf charakteristische Merkmale, die eindeutig einem bestimmten Objekt, einer Situation oder einer bestimmten Eigenschaft zugeordnet werden können.

Inspektionssysteme erkennen Objekte durch einen Vergleich der Empfangssignale mit hinterlegten Mustersignalen. Weist ein Empfangssignal gleiche oder ähnliche Merkmale wie die eines Mustersignals auf, kann auf das dem Mustersignal zugeordnete Objekt geschlossen werden. [TR14]

2.2. Ultraschall

Als Schall werden mechanische Schwingungen bezeichnet, die sich in elastischen Medien wie Luft, Wasser oder Festkörpern ausbreiten. Im Allgemeinen wird Schall im Bezug auf seine Frequenz in verschiedene Bereiche unterteilt.

Frequenzbereich	Bezeichnung
0 - 20 Hz	Infraschall
20 Hz - 20 kHz	Hörschall
20 kHz - 1 GHz	Ultraschall
1 GHz - 10 THz	Hyperschall

Tabelle 2.1.: Ultraschalleinteilung
[SS14]

Die Schallgeschwindigkeit c gibt an, wie schnell sich die Schwingungen im jeweiligen Medium ausbreiten. Diese hängt neben anderen Faktoren im Wesentlichen von der Dichte und der Temperatur des Mediums in dem sich der Schall ausbreitet ab. Da das Ultraschallsystem für Messungen im Medium Luft entwickelt wird, beschränken sich die weiteren Betrachtungen des Schalls auf dessen Eigenschaften in Gasen.

Im Allgemeinen gilt für die Schallgeschwindigkeit in Gasen:

$$c_{Gas} = f(K, \rho) = \sqrt{\frac{K}{\rho}} \quad (2.1)$$

c_{Gas} : Schallgeschwindigkeit in Gasen

K : Kompressionsmodul

ρ : Dichte

Der Einfachheit halber wird das Medium Gas in seiner idealisierten Form angenommen. Da der Kompressionsmodul eines idealen Gases nur von dem Adiabatenexponenten κ und dem Druck p abhängt, folgt:

$$c_{Luft} = \sqrt{\kappa \cdot \frac{p_0}{\rho_0}} \quad (2.2)$$

Durch die Betrachtung der thermischen Zustandsgleichung für ideale Gase,

$$\frac{p}{\rho} = R \cdot T \quad (2.3)$$

folgt für die Schallgeschwindigkeit in idealen Gasen:

$$c_{Gas_{ideal}} = \sqrt{\kappa \cdot R \cdot T} \quad (2.4)$$

$$R_{Luft} = \frac{R_m}{M_{Luft}} \quad (2.5)$$

$$c_{Luft} = \sqrt{\kappa \cdot \frac{R_m \cdot T}{M_{Luft}}} \quad (2.6)$$

Für eine Temperatur von 20°C (entspricht einer absoluten Temperatur von 273,15 K + 20 K = 293,15 K) ergibt sich für die Schallgeschwindigkeit in Luft:

$$c_{Luft_{20^\circ C}} = \sqrt{1,40 \frac{8,314 \frac{Nm}{kgK} \cdot 293,15K}{28,949E-3 \frac{kg}{mol}}} = 343,32 \frac{m}{s} \quad (2.7)$$

κ : Adiabatenexponent, $\kappa_{Luft} = 1,40$

p_0 : absoluter Luftdruck

R : Gaskonstante

R_m : universelle Gaskonstante, $R_m = 8,314 \left[\frac{Nm}{kgK} \right]$

M : Molmasse $\left[\frac{kg}{kmol} \right]$, $M_{Luft} = 28,949 \frac{g}{mol}$

T : absolute Temperatur [K]

[SS14, LGW09]

Die Temperaturabhängigkeit der Schallgeschwindigkeit hängt direkt mit der Temperaturabhängigkeit der Dichte von Gasen zusammen. Die Dichte eines Gases nimmt mit steigender Temperatur ab. Mit sinkender Dichte nimmt die Schallgeschwindigkeit zu.

Solange [Gleichung 2.3](#) gilt, wird aus [Gleichung 2.4](#) ersichtlich, dass die Schallgeschwindigkeit mit der Wurzel der absoluten Temperatur zunimmt.

Die [Abbildung 2.1](#) stellt die Temperaturabhängigkeit der Schallgeschwindigkeit grafisch dar.

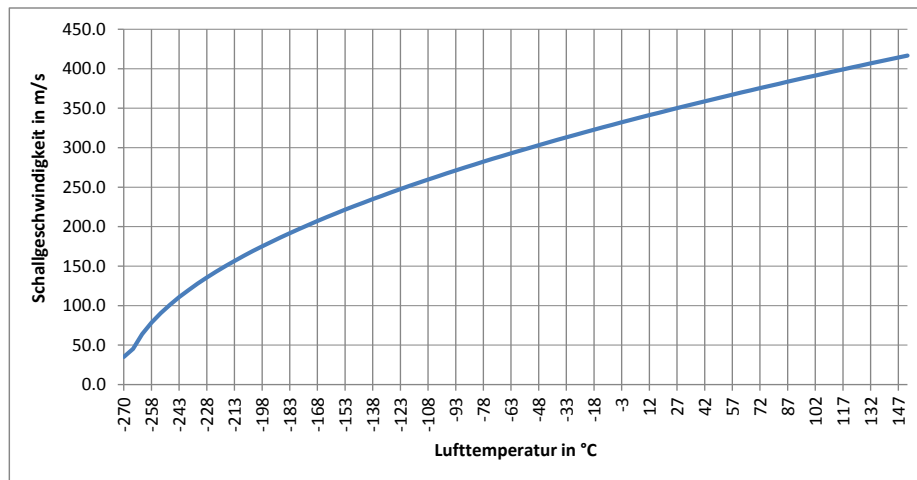


Abbildung 2.1.: c_{Luft} in Abhängigkeit der Temperatur

Die Schallgeschwindigkeit in Luft hängt laut [Gleichung 2.6](#) ebenfalls von der Molmasse M der Luft ab. Da sich die Molmasse mit der Luftfeuchtigkeit ändert, zeigt die Schallgeschwindigkeit auch eine indirekte Abhängigkeit von der Luftfeuchtigkeit, die jedoch gegenüber der Temperaturabhängigkeit sehr klein ist. Im weiteren Verlauf dieser Arbeit wird deshalb dieser Einfluss nicht weiter berücksichtigt.

3. Konzept und Implementierung

In diesem Kapitel wird die Auswahl bzw. die Entwicklung der Einzelkomponenten des Sensorsystems bis zur Hardware-Implementierung des Gesamtkonzepts beschrieben.

3.1. Arduino DUE Mikrocontrollerboard

Im Folgenden wird ein Überblick über das verwendete Mikrocontrollerboard Arduino DUE [Ard15] gegeben. Es werden dabei vornehmlich die Eigenschaften bzw. die Peripherie beschrieben, die für die Entwicklung des Ultraschallsensors von Bedeutung sind, bzw. benötigt werden.

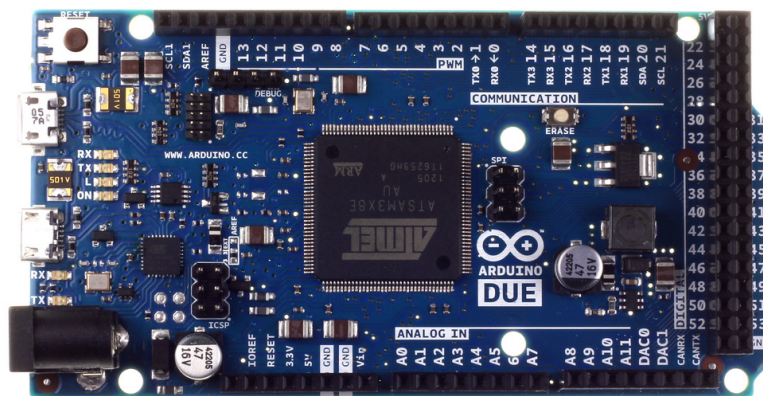


Abbildung 3.1.: Arduino DUE
[Ard15]

Das Arduino DUE Mikrocontrollerboard basiert auf einem 32-Bit Atmel SAM3X8E ARM Cortex-M3 Prozessor [Atm15] mit einem 84 MHz-Takt. Es verfügt u.a. neben den 54 digitalen Ein- und Ausgängen, von denen 12 als PWM- Ausgänge genutzt werden können, über 12 analoge Eingänge. Die Betriebsspannung beträgt 3,3V. Die Spannungsversorgung kann wahlweise über den USB-Port mit 5V oder über ein Netzteil mit einer Spannung von 7V bis maximal 12V erfolgen. Um externe Peripherie zu versorgen, sind Spannungsausgänge mit

5V und 3,3V vorhanden.

Der Analog-Digital-Wandler (ADC) kann mit einer Auflösung von 10- oder 12 Bit bei einer maximalen Abtastfrequenz von ca. 1 MHz betrieben werden. Ein ADC-Kanal ist standardmäßig mit einem Temperatursensor, der sich auf dem Board befindet, verbunden. Als Speicher stehen 96 kByte SRAM Arbeitsspeicher und 512 kByte Flash-Speicher für die Speicherung von Code zur Verfügung.

Die Programmierung erfolgt direkt über die serielle Schnittstelle des Boards ohne externes Programmiergerät. Da das Board über einen USB-Seriell-Konverter verfügt, kann die Programmierung von der USB-Schnittstelle eines PC erfolgen. Für die Erstellung der Software stellt Arduino eine eigene Entwicklungsumgebung, die sog. "Arduino IDE" zur Verfügung.

Um den Arduino DUE als digitalen Signalprozessor (DSP) nutzen zu können, kann die CMSIS-DSP-Bibliothek [ARM15] für Cortex-M Prozessoren von Atmel verwendet werden. Sie stellt spezielle Funktionen für die Verarbeitung von digitalen Signalen bereit.

3.2. Charakterisierung von Ultraschallwandlern

Um geeignete Ultraschallwandler für das Ultraschallsystem und dessen Funktion der Objektklassifizierung zu ermitteln, wurden zwei kommerziell erhältliche Wandler analysiert, die im Folgenden auf Ihre Eigenschaften hin untersucht und untereinander verglichen werden.

Beide Typen nutzen den piezoelektrischen Effekt um das Ultraschallsignal zu erzeugen. Durch das Anlegen einer Wechselspannung an eine Piezokeramik, verformt sich diese plastisch in der Frequenz der Spannung (indirekter piezoelektrischer Effekt). Die dadurch entstehenden Schwingungen werden über eine sog. Anpassungsschicht auf die Luft übertragen. Die Anpassungsschicht sorgt mit ihrer geringen akustischen Impedanz für eine verbesserte Impedanzanpassung an Luft. [TR14]

Das elektrische Ersatzschaltbild eines piezoelektrischen Ultraschallwandlers kann vereinfacht als LCR- Serienschwingkreis dargestellt werden. [Abbildung 3.2](#) [Bon97]

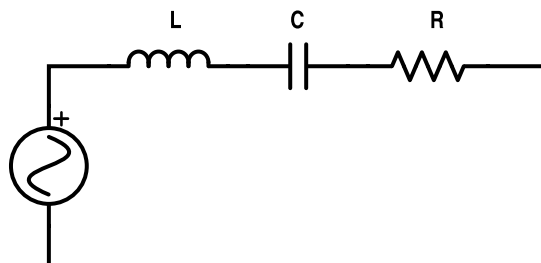


Abbildung 3.2.: Ersatzschaltbild piezoelektr. US-Wandler

Für die komplexe Impedanz \underline{Z} dieses Schwingkreises gilt:

$$\underline{Z}(\omega) = j\omega L + \frac{1}{j\omega C} + R \quad (3.1)$$

ω = Kreisfrequenz

$\frac{1}{j\omega C}$ = komplexe Impedanz des Kondensators

$j\omega L$ = komplexe Impedanz der Spule

R = reeller Widerstand

Bei Resonanzfrequenz kompensieren sich in Schwingkreisen die Induktivitäten und Kapazitäten. Die Impedanz wird minimal. Es ist nur der reelle Widerstand R wirksam. [TSG12]

$$\underline{Z}(\omega_R) = R \quad (3.2)$$

Dieses Verhalten lässt sich nutzen um die exakte Resonanzfrequenz der Ultraschallwandler zu ermitteln. Ist die Impedanz minimal, so ist der Strom maximal, bzw. die Spannung minimal. Durch eine Messung des Amplitudengangs lässt sich die Resonanzfrequenz direkt an der Stelle des Spannungseinbruchs ablesen. Eine solche Amplitudengangmessung ist in [Abbildung 3.3](#) exemplarisch dargestellt. Sie wurde im Frequenzbereich von 20 kHz bis 60 kHz mit einem USB-Oszilloskop Picoscope 3406B [Pic12] durchgeführt. Bei 40,34 kHz zeigt der Amplitudengang einen deutlichen Einbruch und signalisiert damit die Resonanzfrequenz. Die aus dem Amplitudengang abgelesene -6dB Bandbreite beträgt 2,1 kHz, die -3dB Bandbreite beträgt 0,9 kHz. Der reelle Widerstand R beträgt ca. 500 Ω .

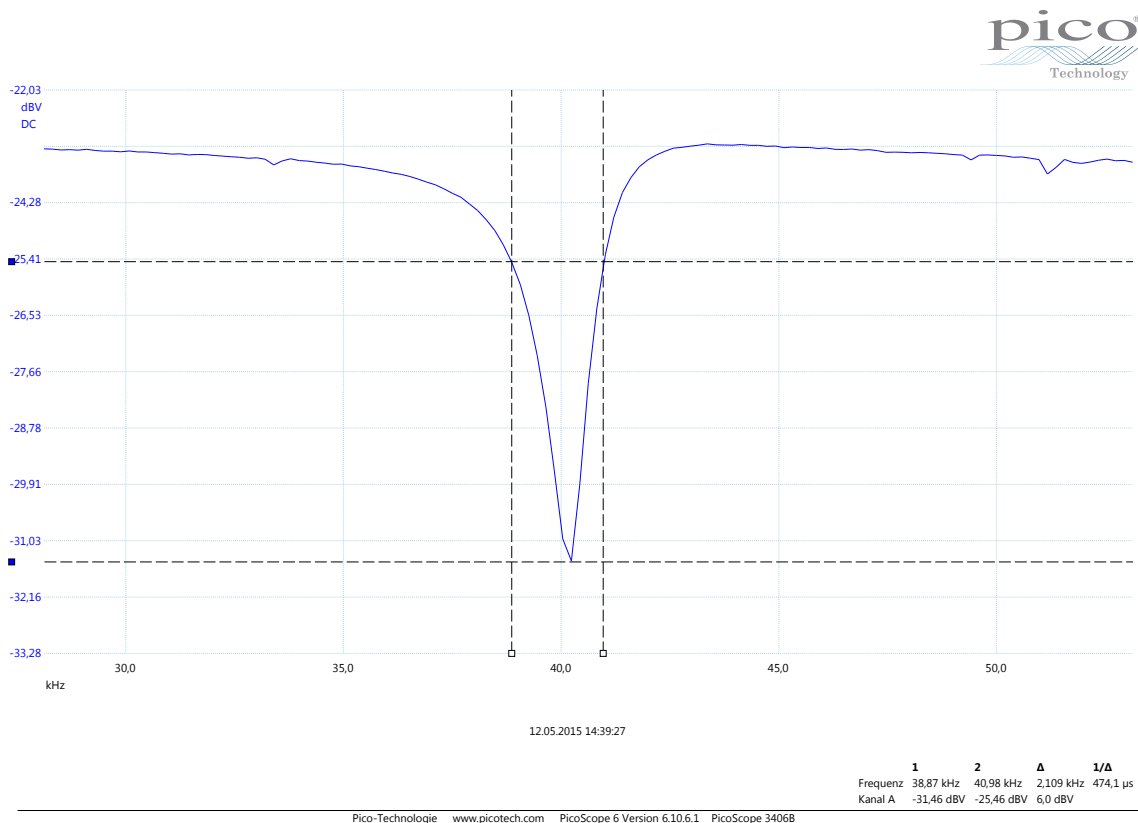


Abbildung 3.3.: Amplitudengang US-Wandler Prowave 400ST160

Um die Sendefrequenz exakt auf die Resonanzfrequenz des verwendeten Ultraschallwandlers anpassen zu können, werden im Folgenden alle zum Vergleich vorliegenden Wandler mit der Amplitudengangmessung charakterisiert. Wenn Sender- und Empfängerwandler unterschiedliche Resonanzfrequenzen aufweisen, muss deren Bandbreite ausreichend groß sein, damit das gesendete Signal vom Empfänger hinreichend gut empfangen werden kann.

Um einfache Systemabschätzungen machen zu können, werden für Ultraschallwandler die Empfangs- und Sendeempfindlichkeiten angegeben. Die Empfangsempfindlichkeit wird in $[\frac{V}{\mu\text{Bar}}]$ gemessen, wobei 0dB $1 \frac{V}{\mu\text{Bar}}$ entsprechen. Die Sendeempfindlichkeit wird in $[\frac{\mu\text{Bar}}{V}]$ gemessen, wobei 0dB einem Schalldruck von 0,0002 μBar in 30cm Entfernung bei einer Anregung mit einem $10 V_{RMS}$ -Signal entsprechen. Über diese Angaben kann abgeschätzt werden, wie hoch der vom Sender abgegebene Schalldruck in Abhängigkeit der angelegten Spannung, bzw. wie hoch die Spannung an den Anschlüssen in Abhängigkeit des auf den Empfänger einwirkenden Schalldrucks ist. [TR14]

3.2.1. Prowave 400Sx160

Ultraschallwandler vom Typ Prowave 400Sx160 [Pro05a] werden u.a. in dem Ultraschallmodul HC-SR04 1.3.3 und dem Sensorsystem CN0343 1.3.1 eingesetzt. Sie sind in spezifische Sende- (400ST160) und Empfängerversionen (400SR160) unterteilt. Das Gehäuse besteht aus Aluminium. Die Schallaustrittsöffnung ist offen und durch ein Metallgitter geschützt. **Abbildung 3.4** zeigt neben dem Ultraschallwandler dessen Aufbauweise in einer Schnittdarstellung.

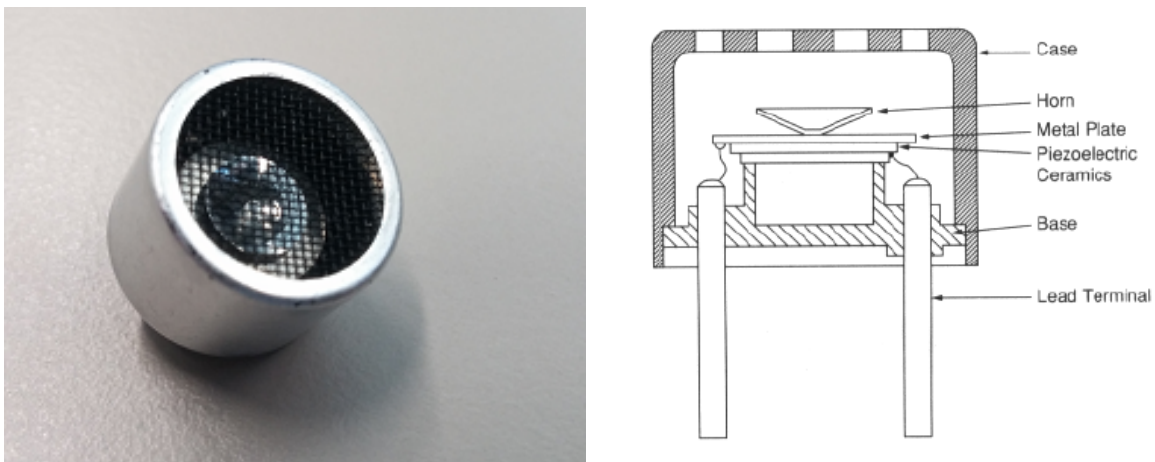


Abbildung 3.4.: 400Sx160 [Pro05a, Mur11]

Die **Tabelle 3.1** zeigt die mit der Amplitudengangmessung ermittelten Resonanzfrequenzen für den jeweiligen Wandler des Typs 400ST160 und 400SR160.

US-Wandler Nr.	Resonanzfreq.(kHz)	-6dB Bandbr.(kHz)	-3dB Bandbr.(kHz)	Trans/Rec
1	40,2	2,1	0,9	T
2	38,1	2,3	0,9	R
3	39,9	1,7	0,8	T
4	38,1	1,3	0,6	R
5	40,3	nicht messbar	1,4	T
6	38,2	nicht messbar	1,2	R

Tabelle 3.1.: Resonanzfrequenzmessung Prowave 400Sx160

Laut Datenblatt liegt die Resonanzfrequenz bei $40,0 \pm 1$ kHz. Die Messungen zeigen, dass die Empfänger-Versionen eine niedrigere Resonanzfrequenz als die Senderversionen aufweisen und leicht außerhalb der Toleranz liegen. Die -6dB Bandbreite ist für die Senderversion mit 2kHz, für die Empfängerversion mit 2,5kHz angegeben. Die Messungen zeigen teilweise deutliche Abweichungen von diesen spezifizierten Werten.

Die Empfindlichkeit des Senders 400ST160 beträgt laut Datenblatt min. 120dB. Die Empfangsempfindlichkeit des 400SR160 wird mit min. -61dB angegeben.

3.2.2. Murata MA40MF14-0B

Ultraschallwandler vom Typ Murata MA40MF14-0B [Mur11] sind sowohl für den Sende- als auch für den Empfangsbetrieb ausgelegt. Das Aluminiumgehäuse ist vollständig geschlossen und auf der Rückseite mit Kunststoff vergossen. Durch seine geschlossene Bauweise ist dieser Ultraschallwandler wasserdicht und dadurch besonders für den Betrieb in rauen Umgebungen (z.B. Automobil) geeignet.

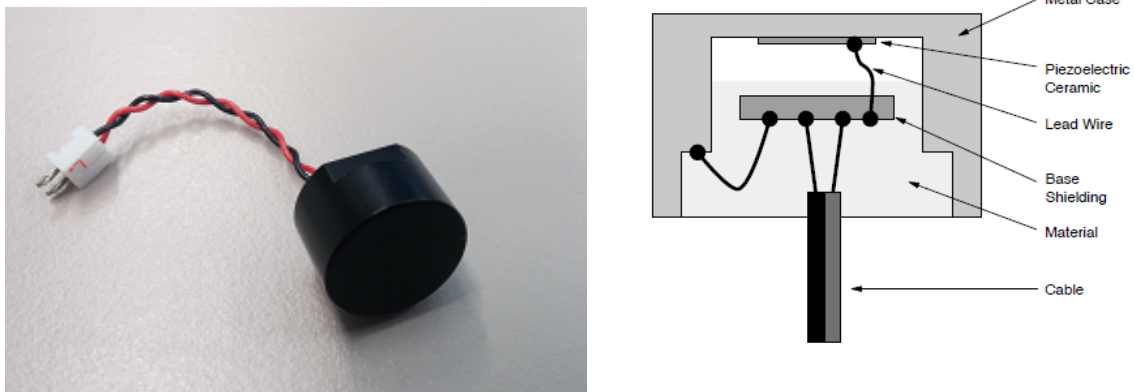


Abbildung 3.5.: MA40MF14-0B [Mur11]

Die [Tabelle 3.2](#) zeigt die Ergebnisse der Resonanzfrequenzmessungen für den jeweiligen Wandler des Typs Murata MA40MF14-0B

US-Wandler Nr.	Resonanzfreq./kHz	-3dB Bandbr. (kHz)
1	40,3	1,4
2	40,4	2,1
3	40,3	1,3
4	40,4	1,2
5	40,6	1,3
6	40,3	1,2

Tabelle 3.2.: Resonanzfrequenzmessung Typ Murata MA40MF14-0B

Im Datenblatt wird die Resonanzfrequenz mit 40,0kHz angegeben. Die höchste Abweichung von diesem Wert von +1,5% weist der US-Wandler 5 auf. Die Sendempfindlichkeit der Murata MA40MF14-0B beträgt laut Datenblatt min. 101dB Die Empfangsempfindlichkeit beträgt min. -87dB. Da die -3dB-Bandbreiten einen Wert von mindestens 1,2 kHz aufweisen, sind theoretisch alle Wandler untereinander als Sender und Empfänger kombinierbar.

3.3. Analoge Schaltungsentwicklung

Dieses Unterkapitel beschreibt die Entwicklung der analogen Schaltungen, die für das Senden und Empfangen des Ultraschallsignals zuständig sind.

3.3.1. Eingangsverstärkerschaltung

Um das von dem Ultraschallwandler in ein Wechsellspannungssignal gewandelte Echosignal optimal digitalisieren zu können, ist es notwendig, es bestmöglich in dem Eingangsspannungsbereich des Analog-Digital Wandlers von 0V bis 3,3V abzubilden. Dies geschieht einerseits über die Addition eines Gleichspannungsanteils von der Hälfte der maximalen Eingangsspannung, da der ADC nur positive Spannungen wandeln kann, andererseits durch die Spannungsverstärkung des Eingangssignals. Um nur das Nutzsignal zu verstärken und eventuelle Störungen aus dem Signal zu filtern, ist die Verstärkerschaltung als dreistufiger aktiver Filter aus Operationsverstärkern vom Typ TI TLC274C [Tex01] realisiert. Beim Entwurf der Schaltung wurde sich an der Verstärkerschaltung des HC-SR04 orientiert. Im Folgenden wird diese Schaltung analysiert und auf die Anforderungen des Sensorsystems angepasst. Zur Verifizierung der errechneten Werte wurden die einzelnen Verstärkerstufen mit dem Programm LTSpice 4.22x simuliert und anschließend mit dem Picoscope [Pic12] einer Messung unterzogen.

Abbildung 3.6 zeigt die Verstärkerschaltung des HC-SR04:

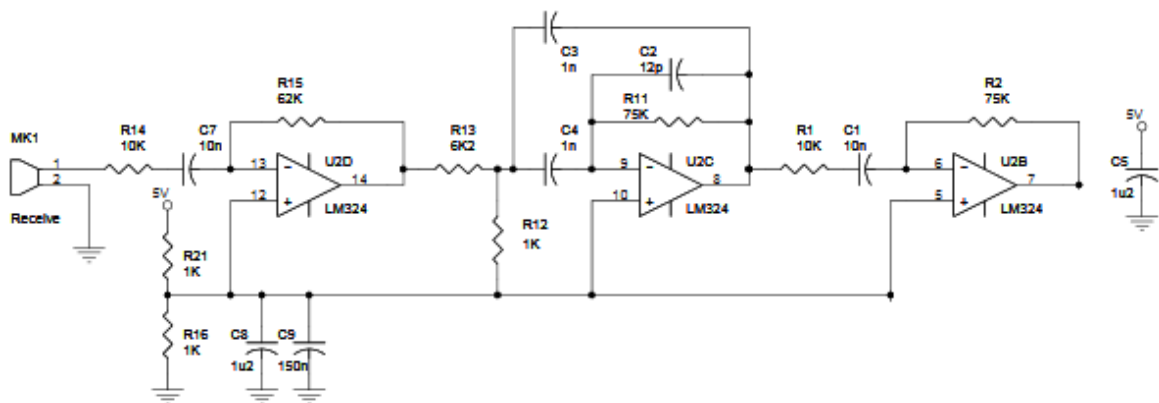


Abbildung 3.6.: HC-SR04 Verstärkerschaltung [Emi14]

Um auf die Signalspannung ein Gleichstrom-Offset zu addieren, wird über einen symmetrischen Spannungsteiler aus den Widerständen R_{21} und R_{16} die Hälfte der Versorgungsspannung auf die nichtinvertierenden Eingänge der Operationsverstärker gelegt. In [Abbildung 3.6](#) wird als Versorgungsspannung 5V angenommen. Da die maximale Eingangsspannung des ADC vom Arduino DUE jedoch 3,3V beträgt, muss auch der Spannungsteiler mit 3,3V versorgt werden. Die Blockkondensatoren C_5 , C_8 und C_9 dienen der Spannungsstabilisierung.

Erste Verstärkerstufe

Die erste invertierende Verstärkerstufe zeigt durch den Kondensator C1 am Eingang Hochpassverhalten. 3.7

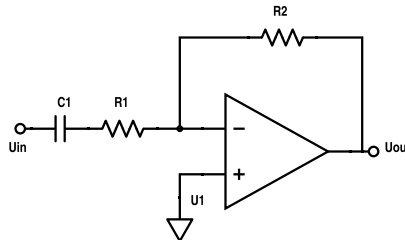


Abbildung 3.7.: aktiver Hochpass

Die 3dB Grenzfrequenz des Hochpasses errechnet sich durch:

$$f_g = \frac{1}{2\pi \cdot RC} \quad (3.3)$$

$$f_g = \frac{1}{2\pi \cdot 10 \text{ k}\Omega \cdot 10 \text{ nF}} = 1591 \text{ Hz} \quad (3.4)$$

Für die Verstärkung v gilt:

$$v = \frac{R2}{R1} \quad (3.5)$$

$$v = \frac{6,2 \text{ k}\Omega}{1 \text{ k}\Omega} = 6,2 \quad (3.6)$$

Für die Verstärkung $v[\text{dB}]$ gilt:

$$20 \cdot \log_{10}(v) = v_{dB}[\text{dB}] \quad (3.7)$$

Daraus folgt für die Verstärkung der ersten Verstärkerstufe:

$$v_{dB} = 20 \cdot \log_{10}(6,2) = 15,85 \text{ dB} \quad (3.8)$$

Die [Abbildung 3.8](#) zeigt den mit LTSpice simulierten Amplitudengang. Die Simulation bestätigt die errechnete Werte von $v_{dB} = 15,85 \text{ dB}$ und $f_g = 1,59 \text{ kHz}$

Die [Abbildung 3.9](#) zeigt den mit dem Picoscope gemessenen Amplitudengang der ersten Verstärkerstufe. Die Messung ergibt eine Verstärkung von $v_{dB} = 15,9 \text{ dB}$ und eine 3dB Grenzfrequenz von $f_g = 1,59 \text{ kHz}$.

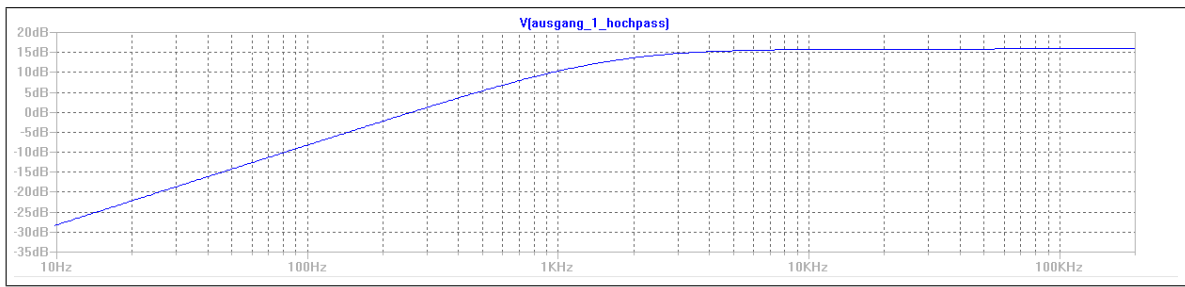


Abbildung 3.8.: Simulation Frequenzgang 1. Hochpasstufe

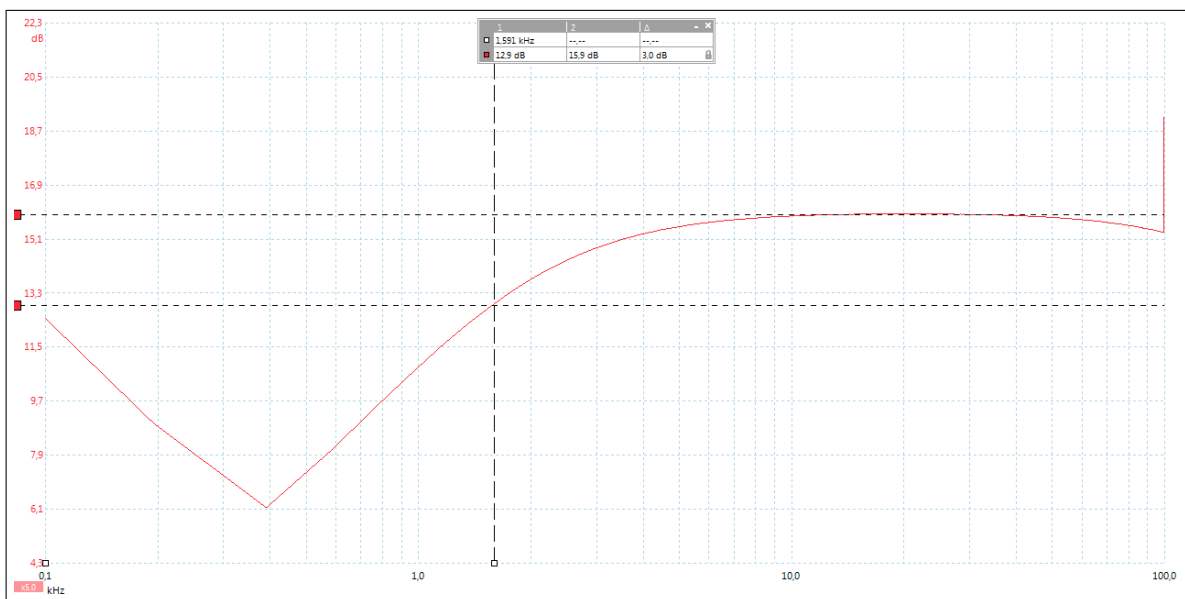


Abbildung 3.9.: Messung Frequenzgang 1. Hochpasstufe

Zweite Verstärkerstufe

Die zweite Stufe stellt einen invertierenden, mehrfach rückgekoppelten aktiven Bandpassfilter dar. [Abbildung 3.10](#) [TSG12]

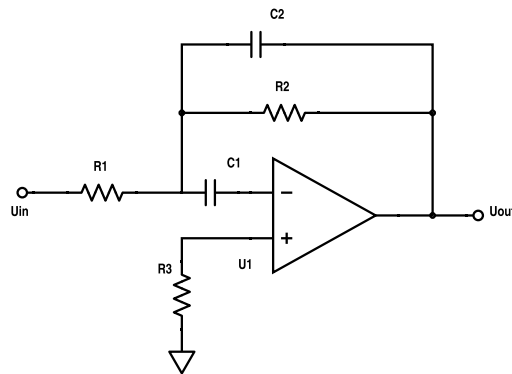


Abbildung 3.10.: Mehrfach rückgekoppelter Bandpass

Die Mittenfrequenz dieses Filters errechnet sich im Fall $C_1 = C_2$ durch:

$$f_0 = \frac{1}{2\pi C} \sqrt{\frac{R_1 + R_3}{R_1 \cdot R_2 \cdot R_3}} \quad (3.9)$$

$$f_0 = \frac{1}{2\pi \cdot 1 \text{ nF}} \cdot \sqrt{\frac{6,2 \text{ k}\Omega + 1 \text{ k}\Omega}{6,2 \text{ k}\Omega \cdot 75 \text{ k}\Omega \cdot 1 \text{ k}\Omega}} = 19,8 \text{ kHz} \quad (3.10)$$

Die Verstärkung bei der Mittenfrequenz errechnet sich im Fall $C_1 = C_2$ durch:

$$v = \frac{R_2}{2R_1} \quad (3.11)$$

$$v = \frac{75 \text{ k}\Omega}{2 \cdot 6,2 \text{ k}\Omega} = 6,04 \quad (3.12)$$

Mit der [Gleichung 3.7](#) folgt:

$$v_{dB} = 20 \cdot \log_{10}(6,04) = 15,62 \text{ dB} \quad (3.13)$$

Die [Abbildung 3.11](#) zeigt den mit LTSpice simulierten Frequenzgang des Bandpasses. Die Simulation ergibt Werte von $v_{dB} = 15,66 \text{ dB}$ und $f_g = 19,731 \text{ kHz}$.

Die [Abbildung 3.12](#) zeigt den mit dem Picoscope gemessenen Frequenzgang des Bandpasses. Die Messung ergibt Werte von $v_{dB} \approx 13 \text{ dB}$ und $f_g \approx 20 \text{ kHz}$

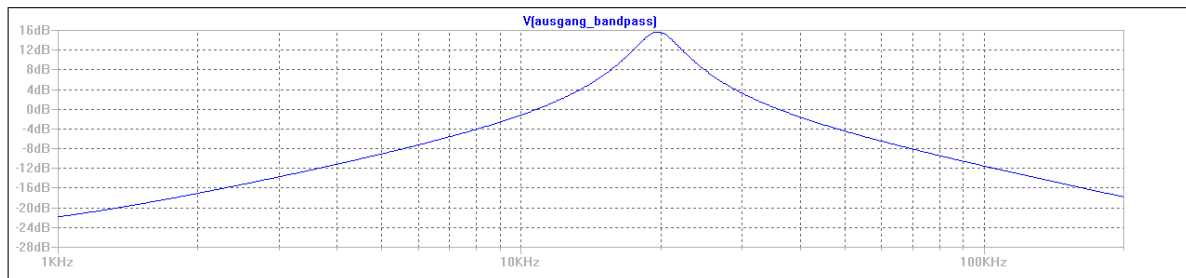


Abbildung 3.11.: Simulation Frequenzgang Bandpass

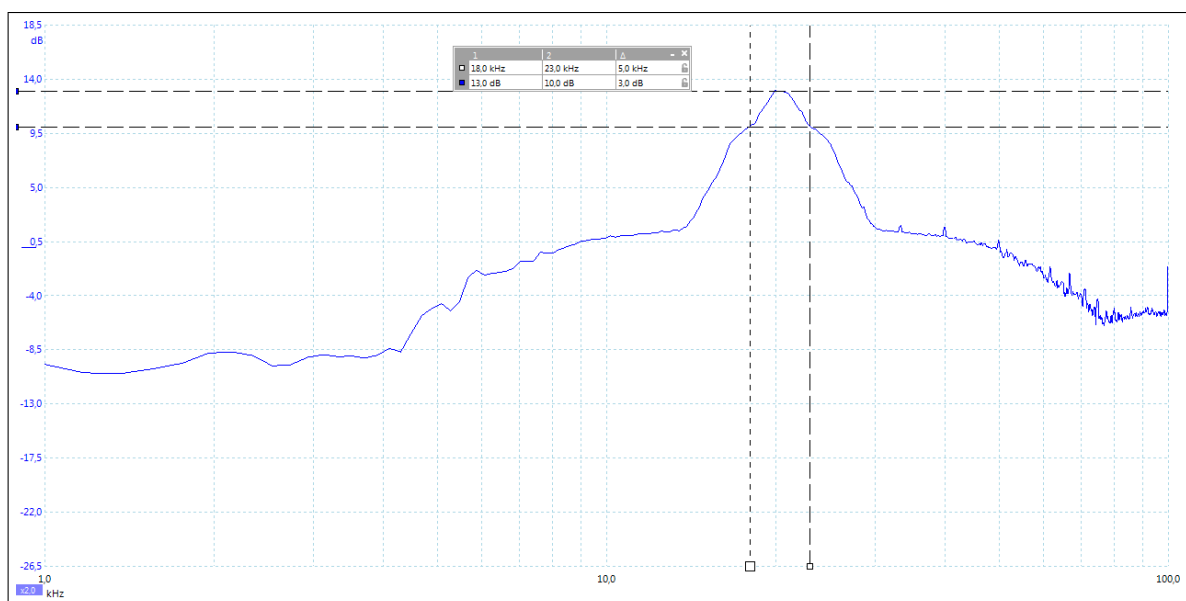


Abbildung 3.12.: Messung Frequenzgang Bandpass

Bei der Analyse fällt auf, dass sowohl Berechnung, Simulation und Messung eine Mittenfrequenz von $f_0 \approx 20 \text{ kHz}$ zeigen. Da der Sensor jedoch mit einer Frequenz von $\approx 40 \text{ kHz}$ arbeiten soll, muss die Beschaltung des Bandpasses auf diese Frequenz angepasst werden. Dies geschieht gemäß [Gleichung 3.9](#) durch die Änderung der Widerstände $R1 = 3 \text{ k}\Omega$, $R2 = 510 \Omega$ und $R3 = 36 \text{ k}\Omega$.

Damit gilt für die Mittenfrequenz des angepassten Bandpasses:

$$f_0 = \frac{1}{2\pi \cdot 1 \text{ nF}} \cdot \sqrt{\frac{3 \text{ k}\Omega + 510 \Omega}{3 \text{ k}\Omega \cdot 36 \text{ k}\Omega \cdot 510 \Omega}} = 40,18 \text{ kHz} \quad (3.14)$$

Die Anpassung hat keine Auswirkung auf die Verstärkung bei Mittenfrequenz:

$$v = \frac{36 \text{ k}\Omega}{2 \cdot 3 \text{ k}\Omega} = 6 \quad (3.15)$$

$$v_{dB} = 20 \cdot \log_{10}(6) = 15,56 \text{ dB} \quad (3.16)$$

Die [Abbildung 3.13](#) zeigt den mit LTSpice simulierten Frequenzgang nach der Anpassung mit einer Verstärkung von $v_{dB} = 15,59 \text{ dB}$ bei einer Mittenfrequenz von $f_g = 39,685 \text{ kHz}$. Die [Abbildung 3.13](#) zeigt den mit dem Picoscope gemessenen Frequenzgang des Bandpasses nach der Anpassung. Die Messung ergibt eine Verstärkung von $v_{dB} \approx 11,9 \text{ dB}$ bei einer Mittenfrequenz von $f_0 \approx 40 \text{ kHz}$. Die Abweichung der Werte für die gemessene und für die errechnete Verstärkung bei der Mittenfrequenz sind vermutlich durch Bauteiltoleranzen hervorgerufen.

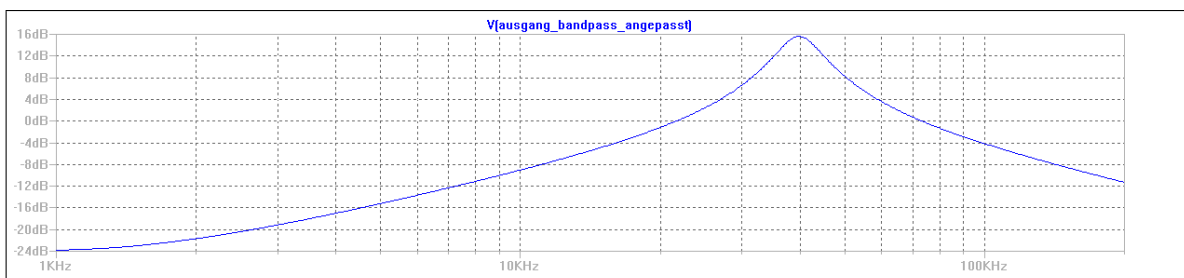


Abbildung 3.13.: Simulation Frequenzgang angepasster Bandpass

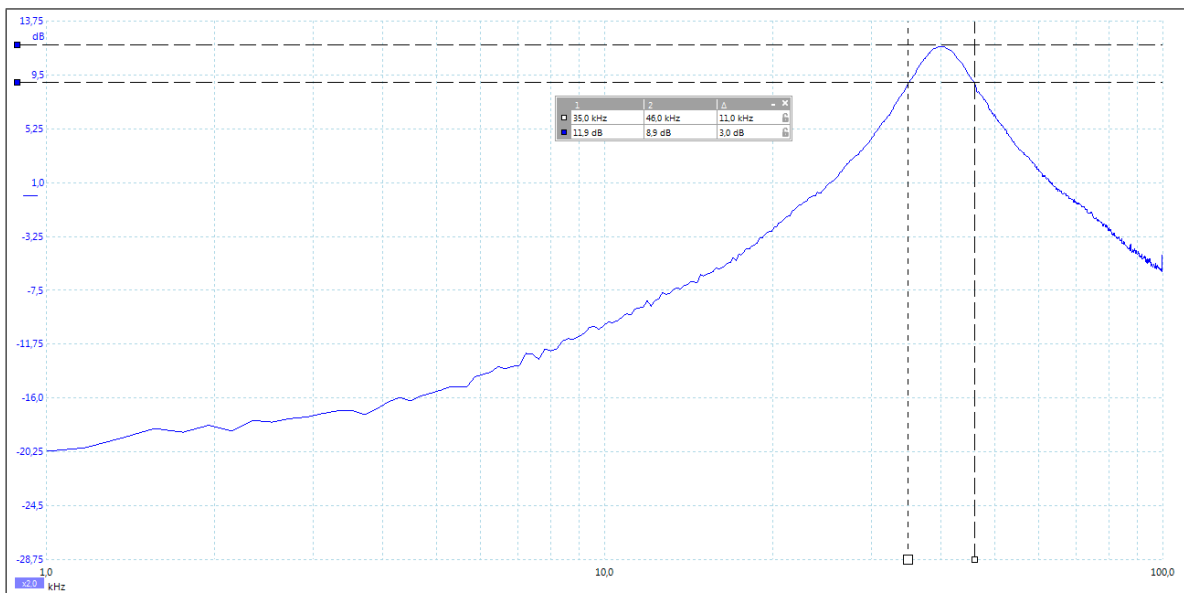


Abbildung 3.14.: Messung Frequenzgang angepasster Bandpass

Dritte Verstärkerstufe

Die dritte Stufe ist wie die erste Stufe ebenfalls als invertierender aktiver Hochpassfilter beschaltet. Mit der Gleichung 3.3 gilt für die Grenzfrequenz:

$$f_g = \frac{1}{2\pi \cdot 1 \text{ k}\Omega \cdot 10 \text{ nF}} = 1,59 \text{ kHz} \quad (3.17)$$

Mit der Gleichung 3.5 gilt für die Verstärkung:

$$v = \frac{75 \text{ k}\Omega}{10 \text{ k}\Omega} = 7,5 \quad (3.18)$$

Mit der Gleichung 3.7 gilt für die Verstärkung in dB:

$$v_{dB} = 20 \cdot \log_{10}(7,5) = 17,50 \text{ dB} \quad (3.19)$$

Die Abbildung 3.15 zeigt den simulierten Frequenzgang der dritten Verstärkerstufe. Die Simulation bestätigt die errechneten Werte von $v_{dB} = 17,5 \text{ dB}$ und $f_g = 1,59 \text{ kHz}$

Die Abbildung 3.16 zeigt den gemessenen Frequenzgang der dritten Verstärkerstufe. Die Messung ergibt $f_g \approx 1,59 \text{ kHz}$ und $v_{dB} \approx 17,65 \text{ dB}$

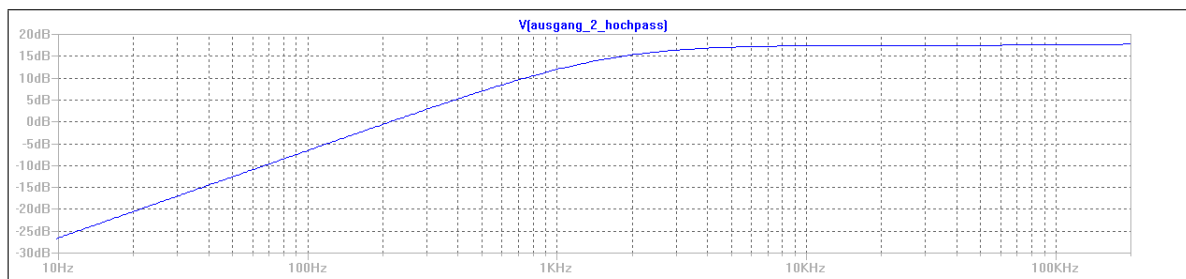


Abbildung 3.15.: Simulation Frequenzgang zweite Hochpassstufe

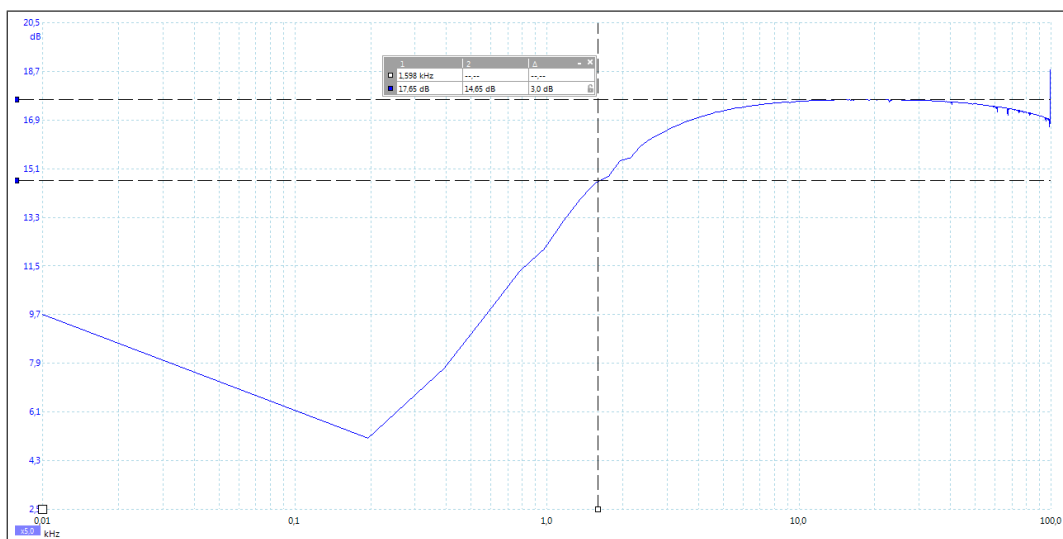


Abbildung 3.16.: Messung Frequenzgang zweite Hochpassstufe

Kombinierte Verstärkerstufen

Abbildung 3.17 zeigt den Schaltplan der gesamten Verstärkerschaltung.

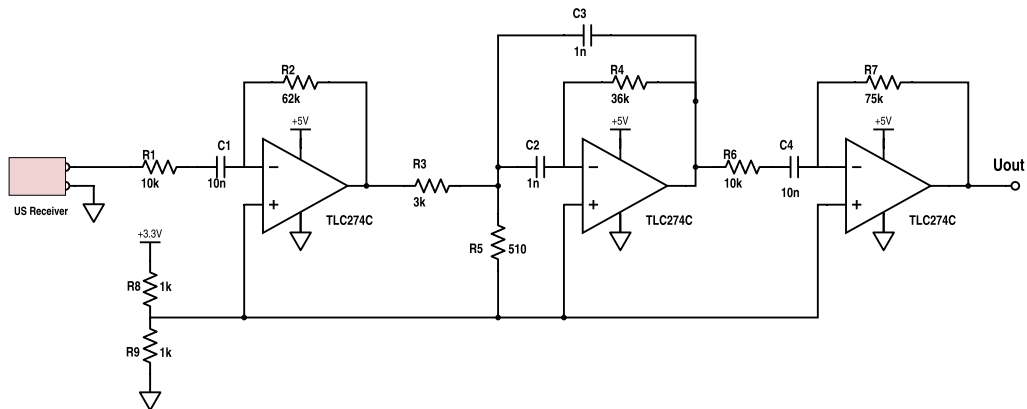


Abbildung 3.17.: Empfangsverstärkerschaltung

Die Kombination der drei Verstärkerstufen ergibt mit den Verstärkungen der einzelnen Stufen eine theoretische Gesamtverstärkung von:

$$V_{dB_{g,theo.}} = V_{dB_1} + V_{dB_2} + V_{dB_3} \quad (3.20)$$

Mit 3.16, 3.16, 3.19 folgt:

$$V_{dB_{g,theo.}} = 15,85 \text{ dB} + 15,56 \text{ dB} + 17,5 \text{ dB} = 48,91 \text{ dB} \quad (3.21)$$

Die 3.18 zeigt den simulierten Frequenzgang der kombinierten Verstärkerstufen mit einer Verstärkung von $V_{dB} = 48,78 \text{ dB}$ bei einer Mittenfrequenz von $f_0 = 39,876 \text{ kHz}$

Die Abbildung 3.19 zeigt den mit dem Picoscope gemessenen Frequenzgang der kombinierten Verstärkerstufen. Die reale Gesamtverstärkung bei Mittenfrequenz berechnet sich aus den jeweiligen gemessenen Einzelverstärkungen. Daraus folgt mit den Einzelverstärkungen aus den Abbildungen 3.9, 3.12, 3.16, für die Gesamtverstärkung:

$$V_{dB_{g,real}} = 15,9 \text{ dB} + 11,9 \text{ dB} + 17,65 \text{ dB} = 45,45 \text{ dB} \quad (3.22)$$

Dies entspricht nach Gleichung 3.7 einer 187,3-fachen Verstärkung.

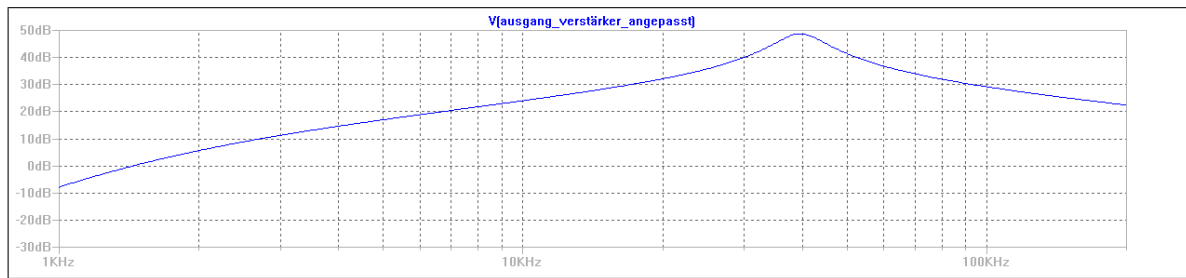


Abbildung 3.18.: Simulation komb. Verstärkerstufen angepasst

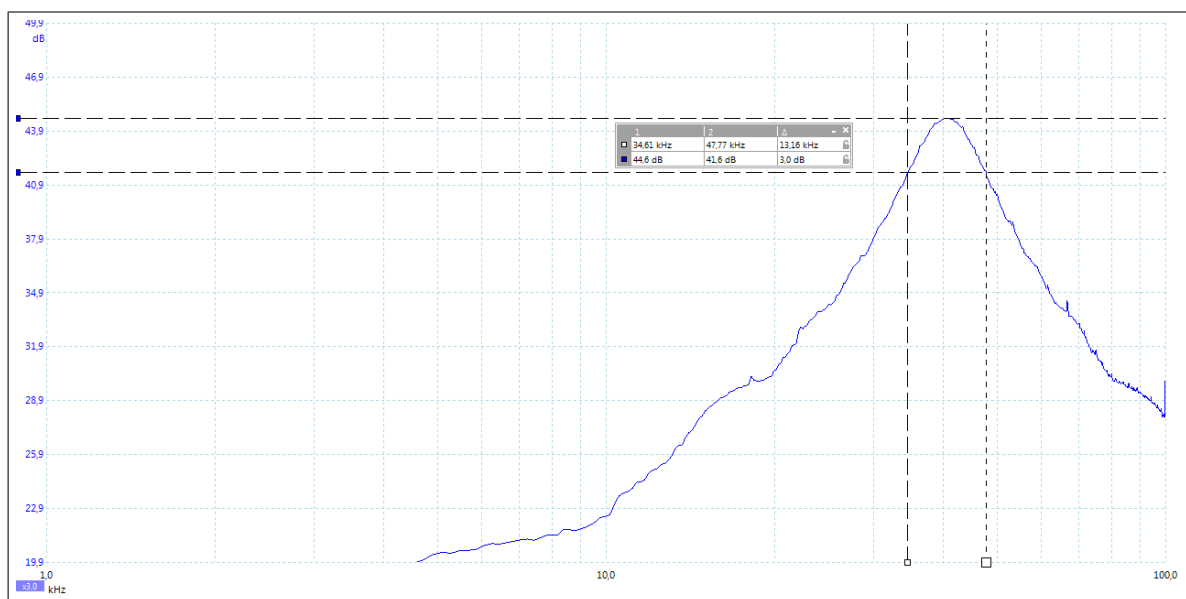


Abbildung 3.19.: Messung komb. Verstärkerstufen angepasst

3.3.2. Sendeschaltung

Da der Arduino DUE über eine maximale Ausgangsspannung von lediglich 3,3 V verfügt, die Ultraschallwandler jedoch mit Spannungen von bis zu $30 V_{p-p}$ (Prowave 400ST160) bzw. bis zu $160 V_{p-p}$ (Murata MA40) betrieben werden können, muss die Ansteuerspannung verstärkt werden um das Potential der Wandler besser ausnutzen zu können. Im Rahmen dieser Arbeit wurden zwei unterschiedliche Konzepte zur Spannungsverstärkung getestet.

Spannungsverstärkung mit MAX232 Pegelwandler

Das Konzept der Spannungsverstärkung mit einem MAX232 Pegelwandler [Tex14] basiert auf der Sendeschaltung des HC-SR04 (Abbildung 3.20). Der MAX232 ist ein integrierter Schaltkreis, der TTL-Pegel (0V/5V) auf RS232-Pegel (-12/+12V) wandelt. Als Versorgung dient eine 5V Gleichspannung. Er verfügt über zwei Sendekanäle, die mit einem DC-DC Wandler jeweils eine Spannung von -7V und +7V erzeugen können. Zur Ansteuerung der Sendeschaltung dienen zwei komplementäre PWM-Signale an den Eingängen T1IN und T2IN mit einer Frequenz von 40kHz. Zwischen den Ausgängen T1OUT und T2OUT entsteht dadurch ein rechteckiges $14 V_{p-p}$ Wechsellspannungssignal, welches für die Ansteuerung des Ultraschallwandlers genutzt wird. s

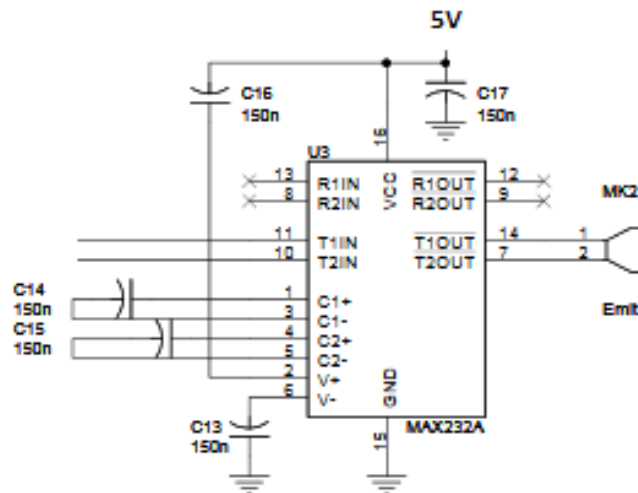


Abbildung 3.20.: HC-SR04 Verstärkerschaltung [Emi14]

Spannungsverstärkung mit Transformator

Das Konzept der Spannungsverstärkung mit einem Transformator basiert auf der Sendeschaltung des MAXQ7667. Der verwendete Transformator vom Typ Prowave K4000004 [Pro05b] ist speziell für Ultraschallwandler mit 40kHz Resonanzfrequenz ausgelegt. Das Übertragungsverhältnis beträgt 1:10. Er wird über einen n-Kanal MOS-FET IRLML6346PBF [Int01] vom Mikrocontroller mit einem PWM-Signal in der Resonanzfrequenz des Ultraschallwandlers angesteuert. Die verstärkte Spannung auf der Sekundärseite treibt den Ultraschallwandler an. Um die Sendespannung stufenlos einstellen zu können, wurde auf der Sekundärseite ein Potentiometer R_1 vorgesehen. Durch das Schalten der Induktivität des Transformators entstehen hohe Induktionsspannungsspitzen über der Drain-Source-Strecke des

MOS-FETs ($U = L \cdot \frac{di}{dt}$). Um eine Zerstörung des MOS-FETs durch diese Spannungsspitzen zu vermeiden, wurde eine RCD-Schutzbeschaltung (Snubber-Schaltung) vorgesehen. Der Ladungsträgerstau, der im Moment des Ausschaltens entsteht, fließt über die Freilaufdiode D1 und lädt den Kondensator C3. Beim Wiedereinschalten des MOS-FETs entlädt sich C3 im Widerstand R3. Die Spannungsspitzen werden dadurch gedämpft. [Bab13]

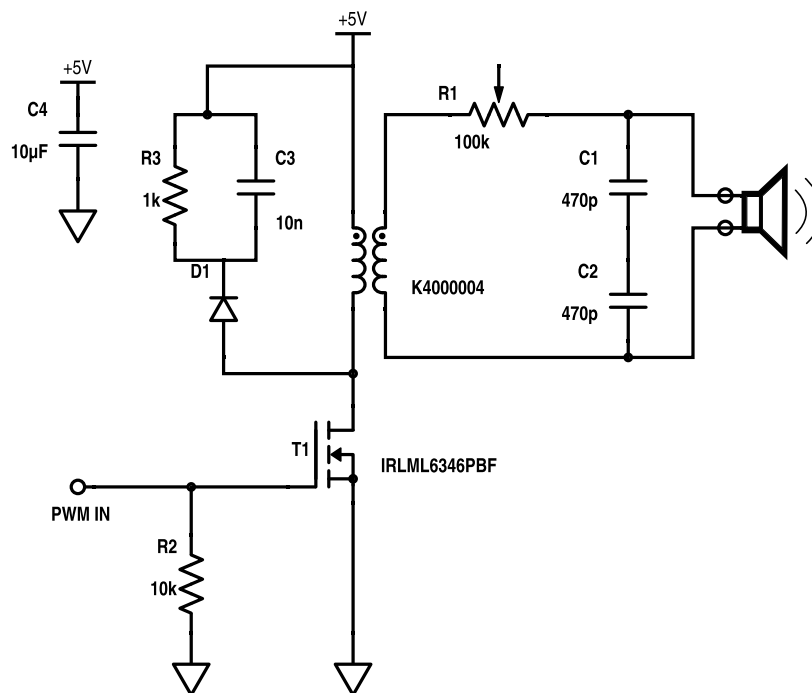


Abbildung 3.21.: Sendeschaltung mit Trafo

3.4. Gesamtsystem

Abbildung 3.22 zeigt das Blockschaltbild des entwickelten Sensorsystems, bestehend aus der Empfängerverstärkerschaltung, einer Sendeschaltung und dem Mikrocontrollerboard Arduino DUE. Für die Kommunikation und die Spannungsversorgung ist das Arduino DUE Mikrocontrollerboard per USB-Schnittstelle mit einem PC verbunden. Die Spannungsversorgung der Sende- und Empfängerschaltung erfolgt über die 5V und 3,3V Ausgänge des Arduino DUE. Ein digitales 40kHz PWM-Signal mit einem LOW-Pegel von 0V und einem HIGH-Pegel von 3,3V wird auf die Sendeschaltung gegeben, die einen Ultraschallwandler antreibt. Das ausgesendete Ultraschallsignal trifft auf ein Objekt und wird von diesem reflektiert. Das Echo wird von dem Empfängerwandler in eine elektrische Wechselspannung gewandelt, die

von der Empfangsschaltung um einen Gleichspannungsanteil angehoben und verstärkt wird. Dieses aufbereitete Signal wird auf den Analog-Digital-Wandler des Arduino DUE gegeben. Auf einem digitalen Signalprozessor findet die Verarbeitung des Signals statt. Die Ergebnisse des Prozesses werden über die serielle Schnittstelle an den PC übermittelt.

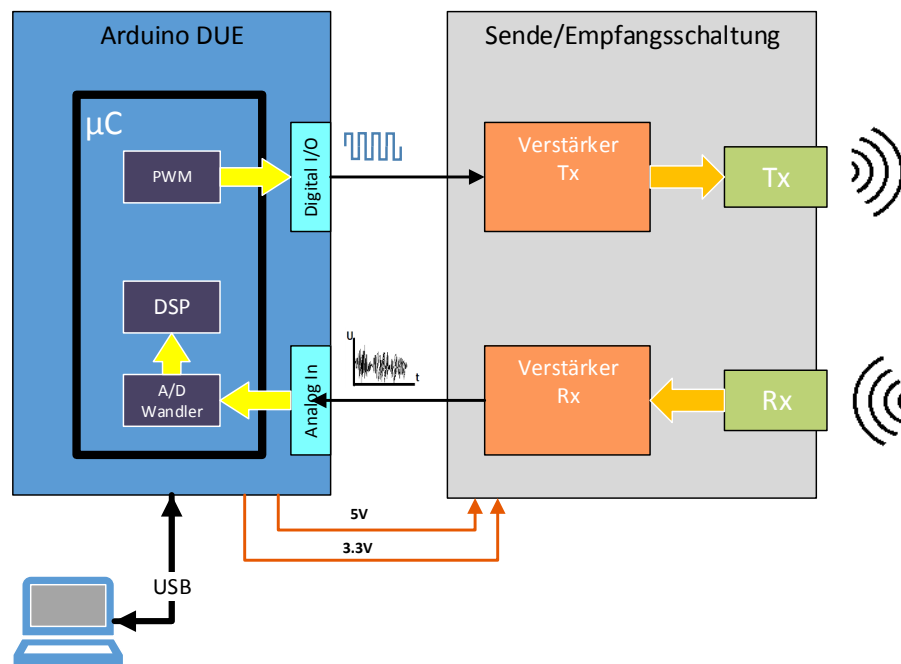


Abbildung 3.22.: Blockschaltbild des Sensorsystems

Da im Vorfeld zwei unterschiedliche Typen von Ultraschallwandlern und zwei Konzepte für die Sendeschaltung untersucht wurden, musste sich für den weiteren Verlauf für jeweils eine Variante entschieden werden.

Es wurden Testmessungen von allen Kombinationen aus den zur Verfügung stehenden Ultraschallwandlern und Sendeschaltungen in Verbindung mit der im [Unterabschnitt 3.3.1](#) entwickelten Empfangsschaltung durchgeführt.

Die Ultraschallwandler wurden gegen eine ebene Wand in einem Meter Abstand ausgerichtet und mit unterschiedlich langen Signalen angeregt. Das durch die Empfangsschaltung verstärkte Echosignal wurde mit dem Picoscope analysiert.

Die Messungen zeigen, dass die Ultraschallwandler vom Typ Murata MA40MF14-0B in Kombination mit beiden Sendeschaltungen keinen verwertbaren Spannungspegel im Erfassungsbereich des ADC liefern. Auch die Variation der Anregedauer liefert keine zufrieden-

stellenden Ergebnisse.

Die Messungen mit den Ultraschallwandlern vom Typ Prowave 400Sx160 hingegen zeigen, dass sie in der Kombination mit beiden Sendeschaltungen verwertbare Spannungspegel liefern.

Im folgenden Verlauf dieser Arbeit wird das Sensorsystem mit den Prowave 400Sx160 Ultraschallwandlern in der Kombination mit der Trafo-Sendeschaltung ([Abschnitt 3.3.2](#)) betrieben, um den Vorteil der stufenlos regelbaren Sendespannung nutzen zu können.

Für das Gesamtsystem ergibt sich damit folgendes Schaltbild:

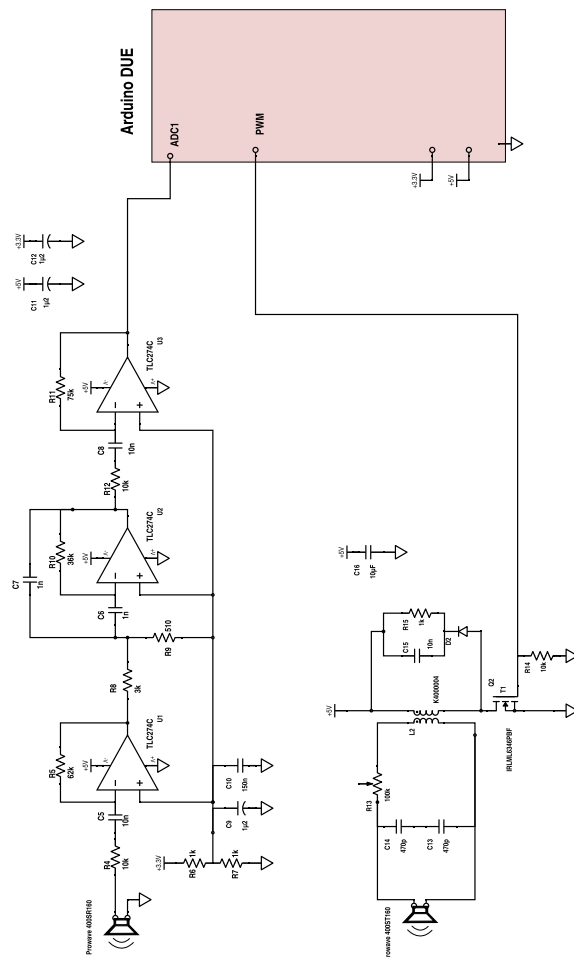


Abbildung 3.23.: Schaltbild Gesamtsystem

3.5. Parameter

Das Sensorsystem verfügt über einige einstellbare Parameter die für die Erfüllung der angestrebten Funktion sinnvoll gewählt werden müssen. In diesem Abschnitt wird erklärt welchen Einfluss die Parameter auf das Sensorsystem haben und wie sie für den weiteren Verlauf der Systementwicklung gewählt werden.

3.5.1. ADC

Der Analog-Digital-Wandler des Arduino DUE lässt sich hinsichtlich der Abtastfrequenz und Auflösung konfigurieren.

Abtastfrequenz

Um eine geeignete Abtastfrequenz f_a zu wählen, muss das Shannon'sche Abtasttheorem beachtet werden. Es besagt, dass ein zeitkontinuierliches Signal $x(t)$ eindeutig aus seinen Abtastwerten $x(nT)$ rekonstruiert werden kann, wenn $x(t)$ auf die halbe Abtastfrequenz bandbegrenzt ist. [OL14] Daraus folgt für das Sensorsystem, dass die Abtastfrequenz mindestens doppelt so hoch wie Ultraschallfrequenz f_{US} gewählt werden muss, um das Signal vollständig rekonstruieren zu können.

$$f_{amin} = 2 \cdot f_{US} = 80 \text{ kHz}$$

Um eine möglichst gute diskrete Abbildung des Signals zu erhalten, wird eine deutlich höhere Abtastfrequenz als f_{amin} von $f_a = 181 \text{ kHz}$ gewählt.

Auflösung

Die Auflösung des ADC ist wahlweise auf 10- oder 12 Bit einstellbar. Für die Speicherung von Ultraschallechoprofilen sind 8 Bit meistens ausreichend, da infolge von Signalschwankungen durch Luftfluktuationen höhere Umsetzungsweiten keinen Zuwachs an Amplitudenauflösung bringen. [TR14] Aus diesem Grund wird sich für eine Auflösung von 10 Bit entschieden. Dies entspricht einer Quantisierung des zeitkontinuierlichen Signals mit $2^{10} = 1024$ Spannungsstufen. Für die Spannungsweite einer Quantisierungsstufe U_{LSB} gilt:

$$U_{LSB} = \frac{U_{REF}}{2^N} \quad (3.23)$$

Daraus folgt:

$$U_{LSB} = \frac{3,3 \text{ V}}{2^{10}} = 3,22 \text{ mV} \quad (3.24)$$

3.5.2. Aufnahmedauer

Über die Anzahl der aufgenommenen ADC-Werte n lässt sich die Dauer der Signalaufnahme einstellen. Für die Aufnahmedauer t_a gilt:

$$t_a = \frac{n}{f_a} \quad (3.25)$$

Für den Verlauf dieser Arbeit wird eine Signalaufnahmedauer von 8,3 ms gewählt. Bei einer Abtastfrequenz von $f_a = 181 \text{ kHz}$ entspricht dies 1500 Abtastwerten. Der Schall legt in dieser Zeit einen Weg von 2,8m zurück. Damit können Echosignale von Hindernissen in 1,4m Entfernung erfasst werden. Aus den Abtastwerten wird das Echosignal mit einer Länge von 235 Abtastwerten extrahiert. Dies entspricht einer Signaldauer von 1,3 ms. Da das Echosignals mit einer Schwellenwertoperation in der Gesamtaufnahme gesucht wird, muss darauf geachtet werden, dass das Abklingen des Übersprechen vom Sendesignals auf den Empfänger abgewartet wird. Der Effekt des Übersprechens ist in [Abbildung 3.24](#) zu sehen.

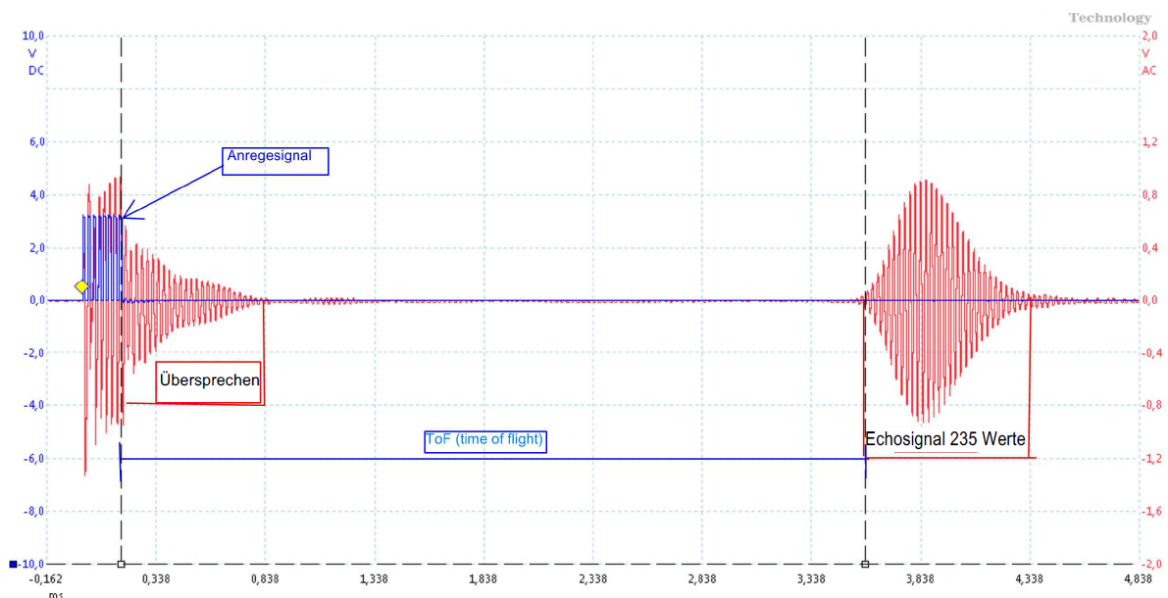


Abbildung 3.24.: Ultraschallmessung

3.5.3. PWM-Frequenz

Die Frequenz der PWM bestimmt die Frequenz des ausgesendeten Signals. Im Idealfall ist sie auf die Resonanzfrequenz des verwendeten Ultraschallsenders einzustellen.

3.5.4. Sendesignallänge

Die PWM-Signallänge bestimmt direkt die Anregedauer des Ultraschallsenders. Für die folgenden Untersuchungen des Ultraschallsystems wurde die Sendesignaldauer auf $185\mu s$ eingestellt, was bei einer PWM-Frequenz von 40kHz 8 Impulsen entspricht. Versuche haben gezeigt, dass der Ultraschallwandler bei dieser Anregedauer einen eingeschwungenen Zustand erreicht hat.

4. Objektklassifizierung mittels Ultraschall

Der Entwicklung des Sensorsystems mit Objektklassifizierung liegt der Gedanke zugrunde, dass Objekte verschiedener Geometrie Ultraschallechos erzeugen, die sich in ihrem Profil, d.h. in ihrer Signalform voneinander unterscheiden. Die Objektklassifizierung kann demnach anhand der Signalform ihrer Ultraschallechos erfolgen.

In diesem Kapitel wird zunächst eine Methode für den Vergleich diskreter Signale theoretisch beschrieben. Im Anschluss folgt ein Test der Methode mit aufgenommenen Echosignalen von Testobjekten mit der Software Scilab 5.5.1., bevor die Methode im nächsten Kapitel in Software auf dem Mikrocontroller umgesetzt wird.

4.1. Methode

Um ein Ähnlichkeitsmaß zwischen zwei Signalverläufen zu erhalten, bietet sich das Verfahren der Korrelation an. [\[OL14\]](#)

Als diskrete Kreuzkorrelationsfunktion (KKF) $r[m]$ wird die Summe der Produkte der Amplitudenwerte zweier diskreter Signalverläufe, in Abhängigkeit einer Verschiebung einer dieser Zeitverläufe um m , bezeichnet.

$$r_{sg}[m] = \sum_{n=-\infty}^{\infty} s_n \cdot g_{n+m}. \quad (4.1)$$

Für jede diskrete Zeitverschiebung m ergibt die Kreuzkorrelationsfunktion $r[m]$ einen Wert, der ein Ähnlichkeitsmaß der Signalverläufe darstellt. Für den Zeitpunkt, an dem sich die Verläufe am ähnlichsten sind, wird dieser Wert maximal.

Stellen $s[n]$ und $g[n]$ identische Zeitverläufe dar, so wird die Korrelation Autokorrelation (AKF) genannt. Das Maximum tritt bei der Verschiebung von $m = 0$ auf.

Für die Objektklassifizierung lässt sich dieses Verfahren nutzen, indem man das digitalisierte Ultraschallecho, welches je nach Objekt eine bestimmte Signalform aufweist, mit hinterlegten Referenzsignalformen, die einem Objekt zugeordnet sind, korrelieren lässt. Korreliert

das Ultraschallecho besonders gut mit einem der Referenzsignale, so kann auf das, diesem Muster zugeordneten Objekt, geschlossen werden. Nach der Definition in [Abschnitt 2.1](#) ist ein System welches diese Methode zur Objektklassifizierung nutzt ein *Inspektionssystem*. Die Signalenergie des Ultraschallechos kann durch die Größe und die Formgebung der unterschiedlichen Objekte unterschiedlich hoch ausfallen. Auch der Abstand des Objektes zum Sensor beeinflusst die Energie des reflektierten Signals. Korreliert man nun ein Ultraschallecho mit einem Referenzsignal welches dem Echo in seiner Signalform unähnlich ist, jedoch über einen höheren Energiebetrag verfügt, so kann die Korrelation dieser beiden Signale ein höheres absolutes Ergebnis liefern als die Korrelation des gleichen Echos mit seinem zugehörigen Referenzsignal. Die Signalenergie hat somit einen maßgeblichen Einfluss auf das absolute Ergebnis der Korrelation.

Um ein Ähnlichkeitsmaß zu erhalten, welches Signalamplituden- und Signalenergieunabhängig ist, muss die Korrelation auf den quadratischen Mittelwert der Leistungen beider Signale normiert werden. [\[TR14\]](#) [\[OL14\]](#)

Allgemein gilt für die Signalenergie E_s eines reellwertigen, zeitdiskreten Signals:

$$E_s = \sum_{n=-\infty}^{\infty} s_n^2. \quad (4.2)$$

Dieses Signal s_n heißt Energiesignal wenn gilt:

$$E_s = \sum_{n=-\infty}^{\infty} s_n^2 < \infty. \quad (4.3)$$

Den Ausdruck in [Gleichung 4.4](#) erhält man analog durch die AKF des Signals bei der Verschiebung von $m=0$:

$$E_s = \sum_{n=-\infty}^{\infty} s_n^2 \iff r_{ss}[m=0] = \sum_{n=-\infty}^{\infty} s_n \cdot s_{n+m} \quad (4.4)$$

$\sqrt{E_s}$ ist der Normungsfaktor des Signals.

Damit gilt für das energienormierte Signal $s_{norm}[n]$:

$$s_{norm}[n] = \frac{s[n]}{\sqrt{E_s}} \quad (4.5)$$

Sind $s[n]$ und $g[n]$ als Reihen aus N Abtastwerten definiert, so gilt für die KKF:

$$r_{sg}[m] = \sum_{n=0}^{2N-2} s_n \cdot g_{n+m} \quad (4.6)$$

Es folgt für die normierte Kreuzkorrelationsfunktion (NKKF) $R_{sg}[n]$:

$$R_{sg}[m] = \sum_{n=0}^{2N-2} s_{n_{norm}} \cdot g_{n+m_{norm}} = \sum_{n=0}^{2N-2} \frac{s_n \cdot g_{n+m}}{\sqrt{E_s E_g}} \quad (4.7)$$

oder analog durch den Ausdruck in [Gleichung 4.4](#):

$$R_{sg}[m] = \sum_{n=0}^{2N-2} \frac{s_n \cdot g_{n+m}}{\sqrt{\sum_{n=0}^{N-1} s_n^2 \cdot \sum_{n=0}^{N-1} g_n^2}} \quad (4.8)$$

Die NKKF kann Werte im Bereich von -1 bis 1 annehmen. Bei Autokorrelation, d.h. bei völliger Übereinstimmung der Signale, wird der Wert 1 bei der Verschiebung von $m = 0$ erreicht.

$$s[n] = g[n] \longrightarrow r_{sg}[m=0] = \sum_{n=0}^{2N-2} \frac{s_n \cdot g_{n+m}}{\sqrt{E_s E_g}} = 1 \quad (4.9)$$

Da die NKKF Amplituden- und Energieunabhängig ist, wird der Wert 1 auch bei Signalen, die sich lediglich durch einen konstanten Amplitudenfaktor unterscheiden, erreicht. Der Wert -1 ist ein Indikator für größte Unähnlichkeit und wird bei Korrelation zweier identischer, exakt gegenphasiger Signale erzielt. Unterscheiden sich die beiden Signalverläufe jedoch in ihrer Form, so gilt für jede Verschiebung m :

$$-1 < R_{sg}[m] < 1$$

Ein weiteres Kriterium für die Ähnlichkeit zweier Signalverläufe stellt die Symmetrie der Korrelationsfunktion der beiden Signale dar. [TR14] Handelt es sich um eine Autokorrelation, so ist diese exakt symmetrisch um den Punkt $m = m_{\max} = 0$. Sind zwei identische Signalverläufe um m_v verschoben, so ist die Korrelationsfunktion symmetrisch um m_v . Ein Maß für die Symmetrie stellt das Verhältnis a der Summe der Beträge Q rechts und links vom Maximum der Korrelationsfunktion dar.

Für die Summe der Beträge links und rechts von m_{max} gilt:

$$Q_{links} = \sum_{n=0}^{m_{max}-1} \sqrt{(r_{sg_n})^2} \quad (4.10)$$

$$Q_{rechts} = \sum_{n=m_{max}+1}^{2N-2} \sqrt{(r_{sg_n})^2} \quad (4.11)$$

Wenn $Q_{links} < Q_{rechts}$ gilt für die Symmetrie der Korrelationsfunktion:

$$a = \frac{Q_{links}}{Q_{rechts}} \quad (4.12)$$

analog dazu gilt wenn $Q_{links} > Q_{rechts}$:

$$a = \frac{Q_{rechts}}{Q_{links}} \quad (4.13)$$

4.2. Umsetzung/Aufnahme der Referenzsignale

Um die Korrelationsanalyse mit dem Ultraschallsensor für die Objektklassifizierung zu nutzen, wurden im Vorfeld Echosignale verschiedener Objekte mit dem Picoscope analysiert. Da der Ultraschallsensor im Hinblick auf dessen Nutzung als Fahrerassistenzsystem entwickelt wird, wurden insbesondere Objekte, die im Straßenverkehr vorkommen können, untersucht. Die Analyse zeigt, dass die Signalformen der Echos stark vom Winkel der Ultraschalleinstrahlung auf das Objekt abhängt. Sieht man sich z.B. Ultraschallechos von einem Hosenbein an, so wird deutlich, dass man kaum reproduzierbare Messergebnisse erhält. Kleinste Veränderungen der Objektform können die Echosignale stark verändern. Um eine zuverlässige Objektklassifizierung durchführen zu können, braucht es in der aktuellen Entwicklungsstufe des Sensors Objekte, die reproduzierbare Echos erzeugen. Um diese Reproduzierbarkeit zu erhalten, wird sich im Folgenden auf die Untersuchung drei verschiedener Objekte beschränkt. Bevor die Methode in [Kapitel 5](#) auf dem Mikrocontroller umgesetzt wird, erfolgt eine Simulation mit dem Programm Scilab 5.5.1.

Im folgenden Abschnitt werden die drei Testobjekte und deren Eigenschaften bezüglich der Reflexion von Ultraschallsignalen vorgestellt. Anhand der in diesem Abschnitt aufgenommenen Referenzechos wird im Folgenden die Simulation der Korrelationsanalyse stattfinden. Die Aufnahme der Referenzechos findet mit dem Sensorsystem mit den in [Abschnitt 3.5](#) gewählten Parametern in einem Abstand zum Testobjekt von ca. 50cm statt. Die Oszilloskopbilder in den Abbildungen [4.2](#), [4.4](#) und [4.6](#) veranschaulichen den kompletten Signalverlauf vom Sendesignal bis zum Objektecho.

Die aus dem Signalverlauf extrahierten, digitalisierten Echos werden vom Mikrocontroller per seriellem Monitor auf dem PC ausgegeben und für die Verarbeitung in das Programm Scilab extrahiert.

Um Vergleichssignale für die Korrelationsanalyse zu erhalten werden die Objekte abermals in ähnlicher Position vor dem Sensor gemessen. Die aus dieser Messung entstandenen Signale ([Abbildung 4.7](#)) werden ebenfalls zur Verarbeitung in Scilab exportiert.

Objekt 1



Abbildung 4.1.: Foto Objekt 1

Das Objekt 1 ist eine ebene, gerade Fläche und stellt im Wesentlichen eine Wand in den Maßen 40cm X 55cm (BxH) dar. Die Charakteristik des Objektes ist durch sein Echosignal in Form eines einzelnen "Bauches" gegeben.

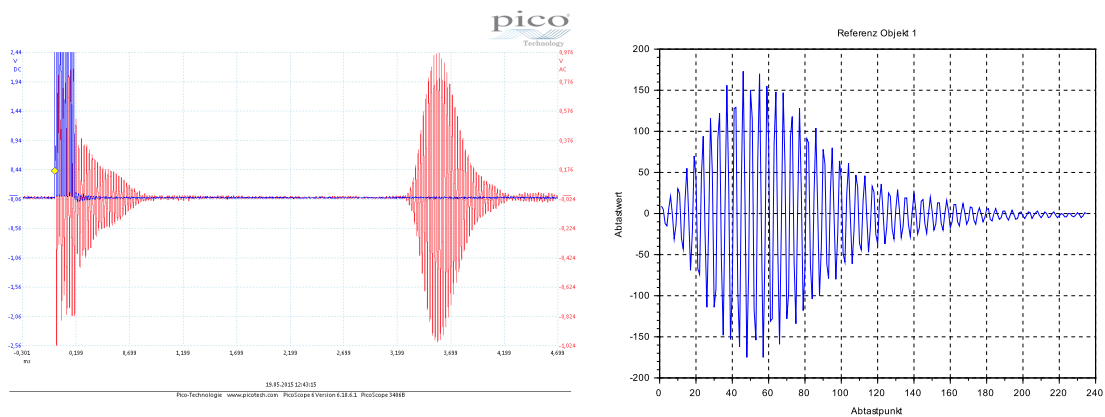


Abbildung 4.2.: Signalverlauf und extr. Echo Objekt 1

Objekt 2

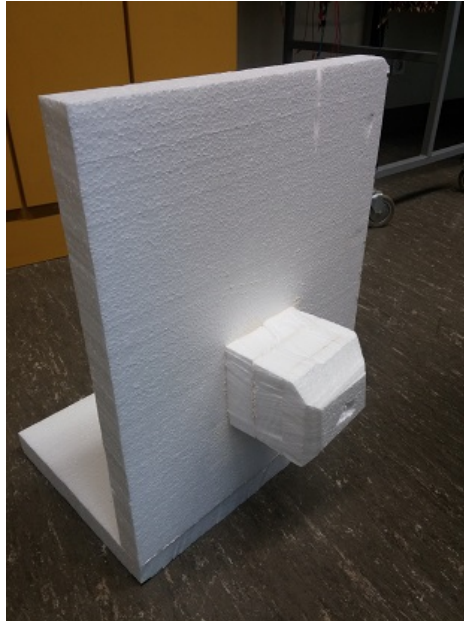


Abbildung 4.3.: Foto Objekt 2

Objekt 2 besteht aus einer ebenen geraden Fläche, auf dessen Höhe von ca. 25cm ein Quader in den Maßen von ca. 12x12x12cm angebracht ist. Die charakteristische Signalform zeigt zwei "Signalbäuche" die durch Reflexion an der Fläche der "Wand" und des Quaders entstehen.

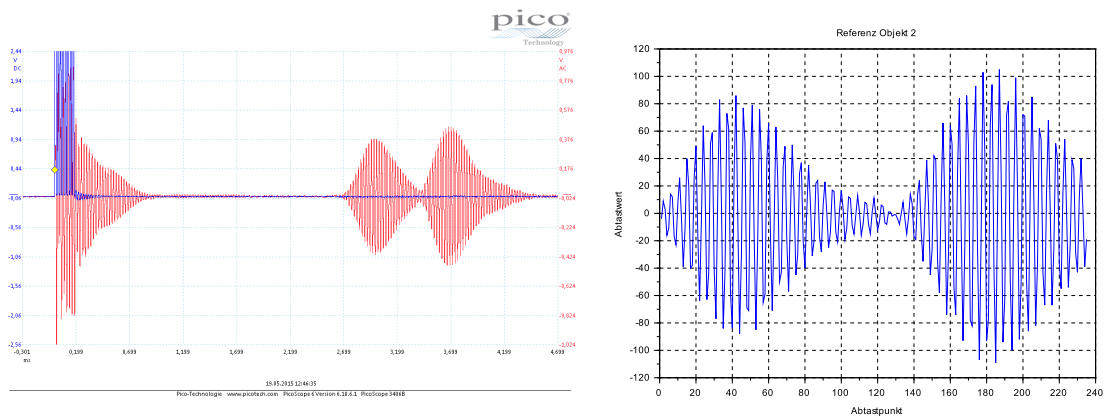


Abbildung 4.4.: Signalverlauf und extr. Echo Objekt 2

Objekt 3

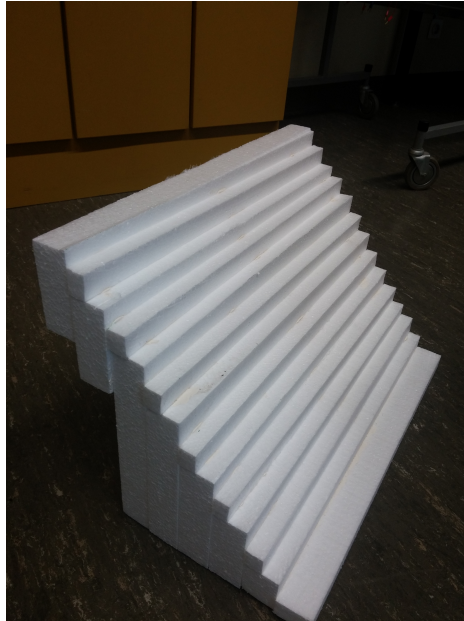


Abbildung 4.5.: Foto Objekt 3

Objekt 3 stellt eine Treppe aus 14 Stufen in dem Maßen 50x35x35cm dar. Die vielen Flächen in leicht unterschiedlichen Abständen, die das Ultraschallsignal reflektieren, erzeugen die Charakteristik in Form eines breiten Echos.

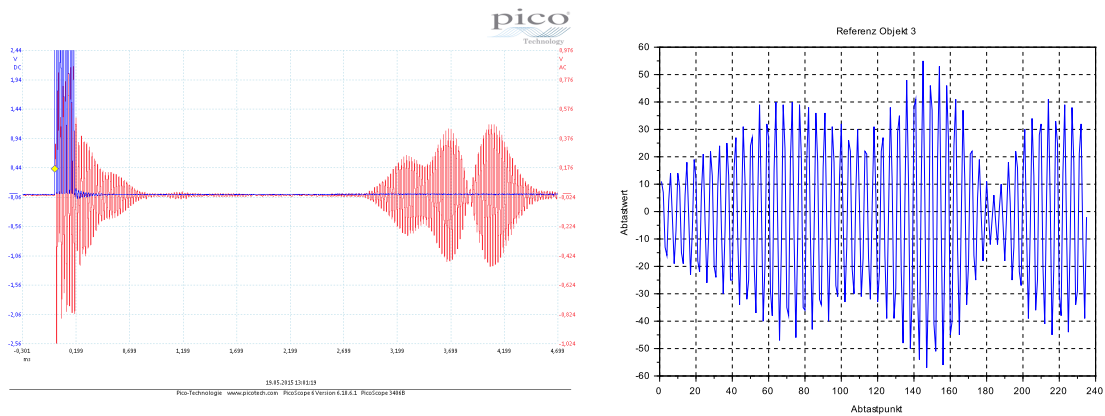


Abbildung 4.6.: Signalverlauf und extr. Echo Objekt 3

Vergleichssignale

Abbildung 4.7 zeigt die aufgenommenen Vergleichssignale, mit denen die Korrelationsanalyse erfolgt. Die Objektposition bei der Vergleichssignal-Aufnahme war der Objektposition bei der Referenzsignal-Aufnahme ähnlich.

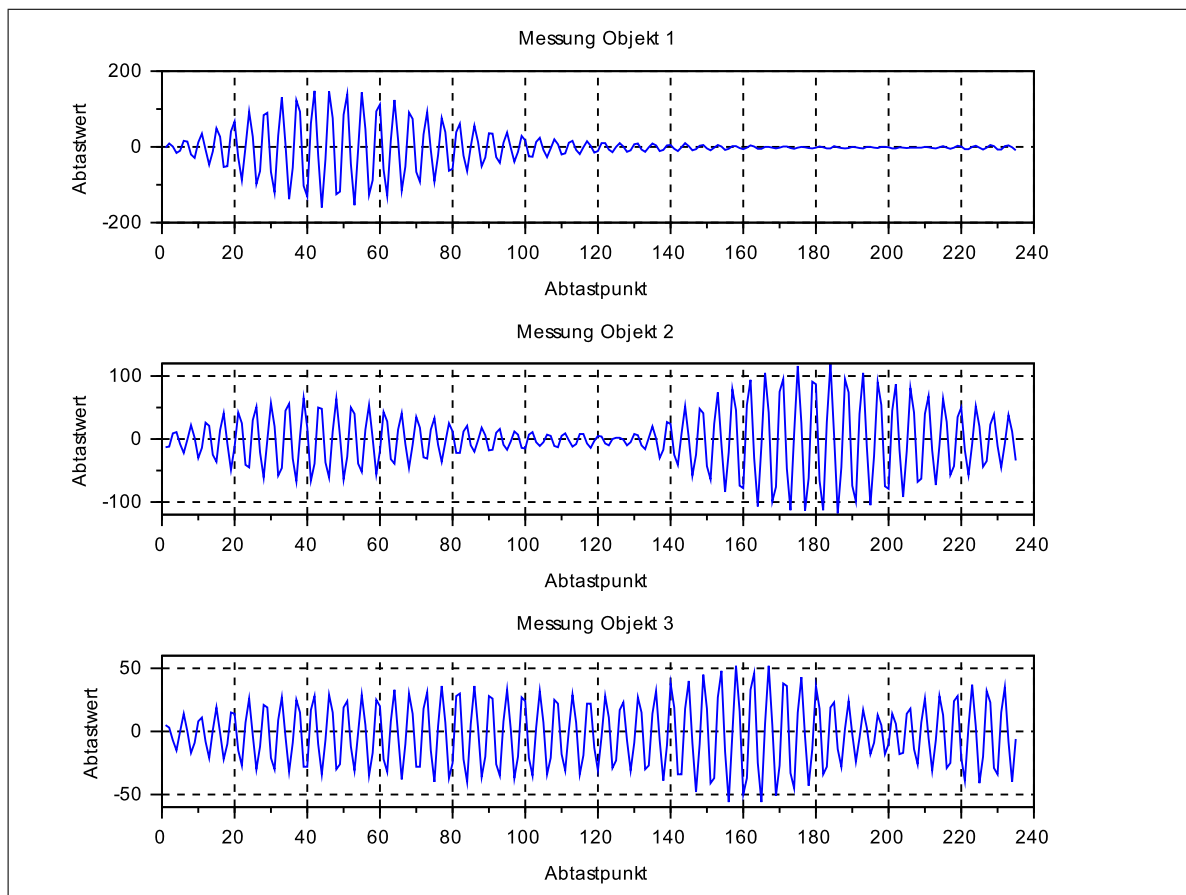


Abbildung 4.7.: Messsignale

4.3. Korrelationsanalyse mit Scilab

Die Korrelationsanalyse der Referenz- und Vergleichssignale erfolgt zunächst auf dem PC mit der Software Scilab 5.5.1. Die nachfolgenden Abbildungen zeigen jeweils links das Vergleichssignal eines Objektes. In der Mitte sind die Korrelationsfunktionen von diesem Vergleichssignal mit allen Referenzsignalen abgebildet. Jeweils rechts neben der Korrelationsfunktion befindet sich die normierte Korrelationsfunktion.

Der Übersichtlichkeit halber werden die Signale wie folgt abgekürzt:

Messsignale: M_i mit fortlaufendem Index $i \in \{1, 2, 3\}$

Referenzsignale: R_k mit fortlaufendem Index $k \in \{1, 2, 3\}$

4.3.1. Objekt 1

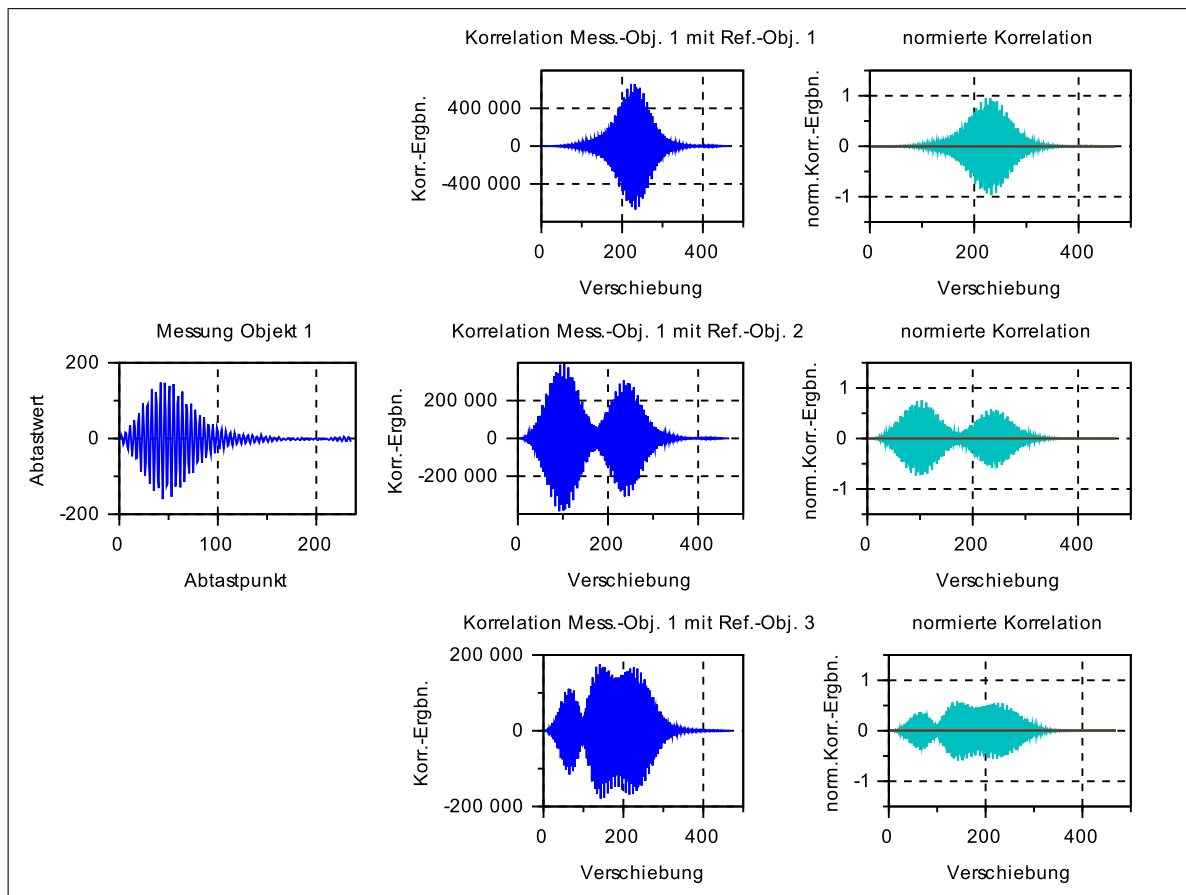


Abbildung 4.8.: Korrelationen Objekt 1 mit Referenzen

Abbildung 4.8 zeigt die Korrelationsanalyse für das Objekt 1.

Der Signalverlauf *Messung Objekt 1* stellt das Vergleichssignal für Testobjekt 1 dar. Es wurde jeweils mit den Referenzsignalen aller Testobjekte korreliert.

Messung Objekt 1 / Referenz Objekt 1 Für die Korrelation des Vergleichssignals *Messung Objekt 1* (Abbildung 4.7) mit dem Referenzsignal *Referenz Objekt 1* ergeben sich folgende Ergebnisse:

Für die Energiebeträge der Signale folgt mit Gleichung 4.4:

$$E_{M_1} = 554290$$

$$E_{R_1} = 864198$$

Da die Korrelationsfunktion an der Stelle (P_{max}) der größten Ähnlichkeit ihr Maximum erreicht, wird dieses Maximum zur Analyse mit der NKKF genutzt.

Ermittelter Punkt P_{max} : 222

Dies entspricht einer Verschiebung m_{max} beider Signale um:

$$m_{max} = N - P_{max} = 235 - 222 = 13$$

Für das Ergebnis der Korrelation bei der Verschiebung m_{max} folgt:

$$r_{M_1R_1}[m_{max}] = \sum_{n=0}^{468} M_{1n} \cdot R_{1n+m_{max}} = \underline{658077} \quad (4.14)$$

Für den normierten Korrelationsfaktor nach Gleichung 4.7 folgt:

$$R_{M_1R_1}[m_{max}] = \frac{658077}{\sqrt{554290 \cdot 864198}} = \underline{\underline{0.951}} \quad (4.15)$$

Für die Symmetrieberechnung:

Betragssumme links von P_{max} : $Q_{links} = 20118275$

Betragssumme rechts von P_{max} : $Q_{rechts} = 24732470$

Daraus folgt für die Symmetrie:

$$a_{M_1R_1} = \frac{20118275}{24732470} = \underline{\underline{0.81}} \quad (4.16)$$

Messung Objekt 1 / Referenz Objekt 2 Die Korrelation des Signals *Messung Objekt 1* mit dem Referenzsignal *Referenz Objekt 2* ergibt folgende Ergebnisse:

Ermittelter Punkt P_{max} : 103

Dies entspricht einer Verschiebung m_{max} beider Signale um:

$$m_{max} = N - P_{max} = 235 - 103 = 132$$

$$E_{M_1} = 554290$$

$$E_{R_2} = 500788$$

Für das Ergebnis der Korrelation bei der Verschiebung m_{max} folgt:

$$r_{M_1R_2}[m_{max}] = \sum_{n=0}^{468} M_{1n} \cdot R_{2n+m_{max}} = \underline{399086} \quad (4.17)$$

Für den normierten Korrelationsfaktor nach [Gleichung 4.7](#) folgt:

$$R_{M_1R_2}[m_{max}] = \frac{399086}{\sqrt{554290 \cdot 500788}} = \underline{\underline{0.757}} \quad (4.18)$$

Für die Symmetrieberechnung:

$$Q_{rechts} = 27560595$$

$$Q_{links} = 11595251$$

Damit folgt für die Symmetrie:

$$a_{M_1R_2} = \frac{11595251}{27560595} = \underline{\underline{0,42}} \quad (4.19)$$

Messung Objekt 1 / Referenz Objekt 3 Die Korrelation des Signals *Messung Objekt 1* mit dem Referenzsignal *Referenz Objekt 3* ergibt folgende Werte:

Ermittelter Punkt P_{max} : 141

Dies entspricht einer Verschiebung m_{max} beider Signale um:

$$m_{max} = N - P_{max} = 235 - 141 = 94$$

Für die Energiebeträge der Signale folgt mit [Gleichung 4.4](#):

$$E_{M_1} = 554290$$

$$E_{R_3} = 165173$$

Für das Ergebnis der Korrelation bei der Verschiebung m_{max} folgt:

$$r_{M_1 R_2}[m_{max}] = \sum_{n=0}^{468} M_{1n} \cdot R_{2n+m_{max}} = \underline{176253} \quad (4.20)$$

Für den normierten Korrelationsfaktor nach [Gleichung 4.7](#) folgt:

$$R_{M_1 R_3}[m_{max}] = \frac{176253}{\sqrt{554290 \cdot 165173}} = \underline{\underline{0.582}} \quad (4.21)$$

Für die Symmetrie folgt:

$$Q_{rechts} = 14733671$$

$$Q_{links} = 7151260$$

Damit gilt für die Symmetrie:

$$a_{M_1 R_3} = \frac{7151260}{14733671} = \underline{\underline{0,49}} \quad (4.22)$$

Auswertung

Sowohl der normierte Korrelationsfaktor als auch die Symmetrie sind bei der Korrelation vom Messsignal *Messung Objekt1* mit dem Referenzsignal *Referenz Objekt 1* am größten. Damit kann das gemessene Objekt eindeutig als Objekt 1 klassifiziert werden.

4.3.2. Objekt 2

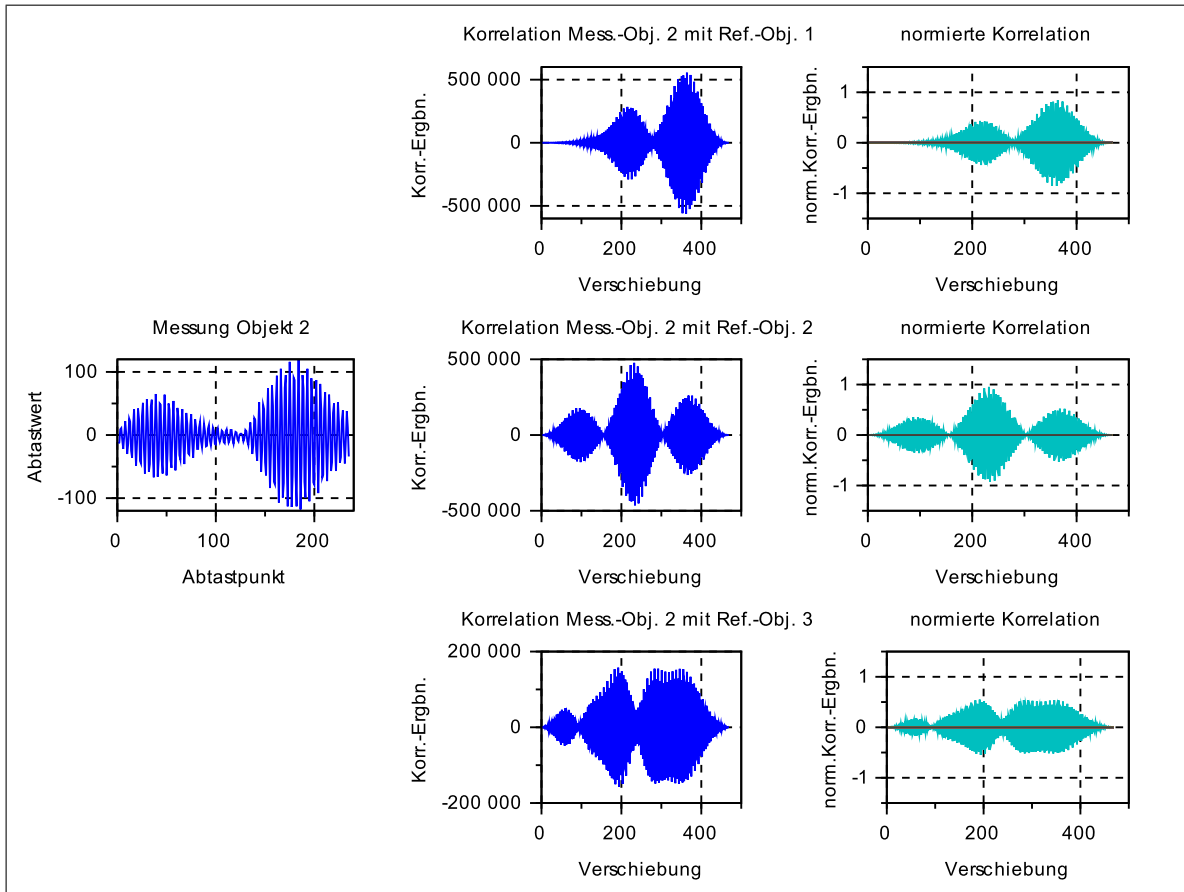


Abbildung 4.9.: Korrelationen Objekt 2 mit Referenzen

Abbildung 4.9 zeigt die Korrelationsanalyse für das Objekt 2.

Der Signalverlauf *Messung Objekt 2* stellt das Vergleichssignal für Testobjekt 2 dar. Es wurde jeweils mit den Referenzsignalen aller Testobjekte korreliert.

Messung Objekt 2 / Referenz Objekt 1 Die Korrelation des Signals *Messung Objekt 2* mit dem Referenzsignal *Referenz Objekt 1* ergibt folgende Werte:

Ermittelter Punkt P_{max} : 364

Dies entspricht einer Verschiebung m_{max} beider Signale um:

$$m_{max} = N - P_{max} = 235 - 364 = -129$$

Für die Energiebeträge der Signale folgt mit [Gleichung 4.4](#):

$$E_{M_2} = 498870$$

$$E_{R_1} = 864198$$

Für das Ergebnis der Korrelation bei der Verschiebung m_{max} folgt:

$$r_{M_2R_1}[m_{max}] = \sum_{n=0}^{468} M_{2n} \cdot R_{1n+m_{max}} = \underline{555531} \quad (4.23)$$

Damit folgt für den normierten Korrelationsfaktor nach [Gleichung 4.7](#):

$$R_{M_2R_1}[m_{max}] = \frac{555531}{\sqrt{498870 \cdot 864198}} = \underline{\underline{0.846}} \quad (4.24)$$

Für die Symmetrie folgt:

$$Q_{rechts} = 15414966$$

$$Q_{links} = 35149093$$

Damit gilt für die Symmetrie:

$$a_{M_2R_1} = \frac{15414966}{35149093} = \underline{\underline{0,44}} \quad (4.25)$$

Messung Objekt 2 / Referenz Objekt 2 Die Korrelation des Signals *Messung Objekt 2* mit dem Referenzsignal *Referenz Objekt 2* ergibt folgendes Ergebnis:

Ermittelter Punkt P_{max} : 232

Dies entspricht einer Verschiebung m_{max} beider Signale um:

$$m_{max} = N - P_{max} = 235 - 232 = 3$$

Für die Energiebeträge der Signale folgt mit [Gleichung 4.4](#):

$$E_{M_2} = 498870$$

$$E_{R_2} = 500788$$

Für das Ergebnis der Korrelation bei der Verschiebung m_{max} folgt:

$$r_{M_2R_2}[m_{max}] = \sum_{n=0}^{468} M_{2n} \cdot R_{2n+m_{max}} = \underline{476784} \quad (4.26)$$

Damit folgt für den normierten Korrelationsfaktor nach [Gleichung 4.7](#):

$$R_{M_2R_2}[m_{max}] = \frac{476784}{\sqrt{498870 \cdot 500788}} = \underline{\underline{0.954}} \quad (4.27)$$

Für die Symmetrie folgt:

$$Q_{rechts} = 26547801$$

$$Q_{links} = 22153575$$

Damit gilt für die Symmetrie:

$$a_{M_2R_2} = \frac{22153575}{26547801} = \underline{\underline{0,83}} \quad (4.28)$$

Messung Objekt 2 / Referenz Objekt 3 Die Korrelation des Signals *Messung Objekt 2* mit dem Referenzsignal *Referenz Objekt 3* ergibt folgendes Ergebnis:

Ermittelter Punkt P_{max} : 192

Dies entspricht einer Verschiebung m_{max} beider Signale um:

$$m_{max} = N - P_{max} = 235 - 192 = 43$$

Für die Energiebeträge der Signale folgt mit [Gleichung 4.4](#):

$$E_{M_2} = 498870$$

$$E_{R_3} = 165173$$

Für das Ergebnis der Korrelation bei der Verschiebung m_{max} folgt:

$$r_{M_2R_3}[m_{max}] = \sum_{n=0}^{468} M_{2n} \cdot R_{3n+m_{max}} = \underline{157539} \quad (4.29)$$

Damit folgt für den normierten Korrelationsfaktor nach [Gleichung 4.7](#):

$$R_{M_2R_3}[m_{max}] = \frac{157539}{\sqrt{498870 \cdot 165173}} = \underline{\underline{0.549}} \quad (4.30)$$

Für die Symmetrie folgt:

$$Q_{rechts} = 18104494$$

$$Q_{links} = 7269821$$

Damit gilt für die Symmetrie:

$$a_{M_2R_3} = \frac{7269821}{18104494} = \underline{\underline{0,40}} \quad (4.31)$$

Auswertung

Sowohl der normierte Korrelationsfaktor als auch die Symmetrie sind bei der Korrelation vom Messsignal *Messung Objekt2* mit dem Referenzsignal *Referenz Objekt 2* am größten. Damit kann das gemessene Objekt eindeutig als Objekt 2 klassifiziert werden.

4.3.3. Objekt 3

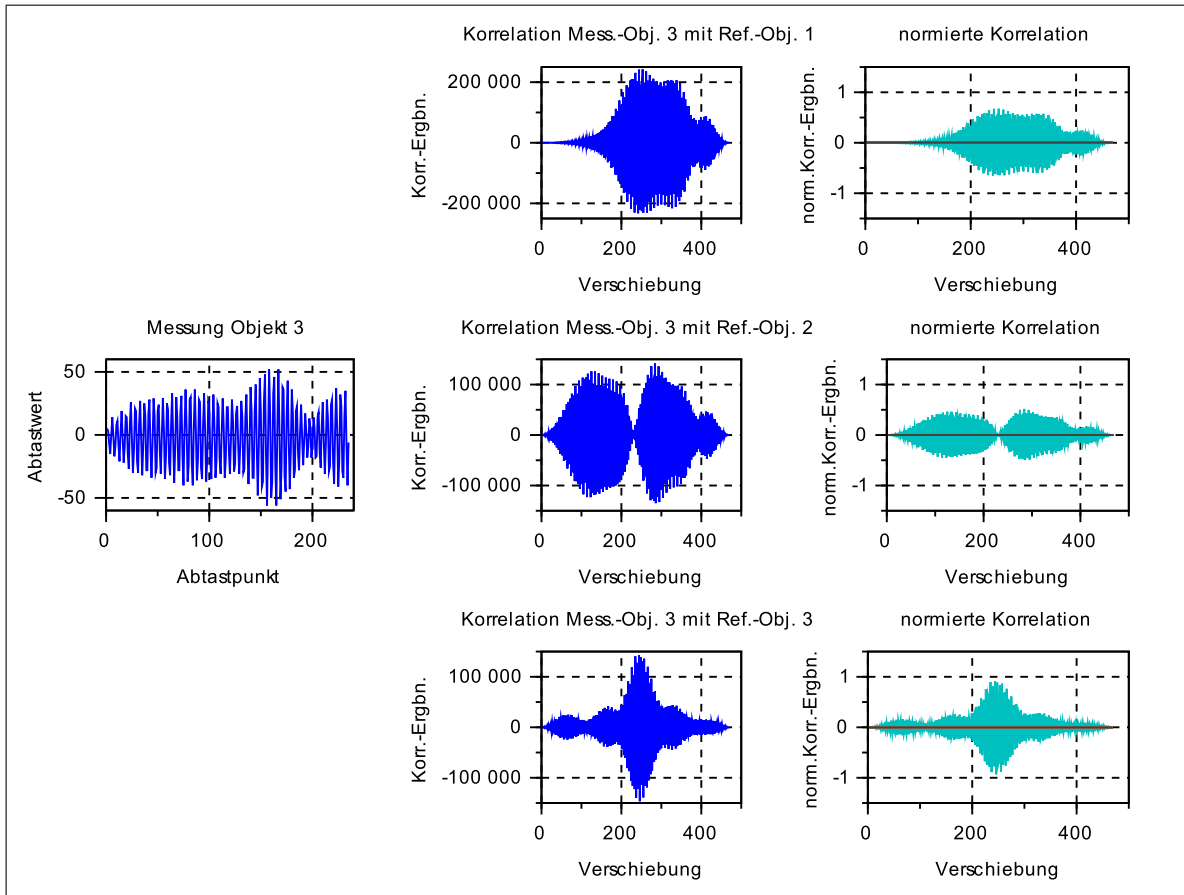


Abbildung 4.10.: Korrelationen Objekt 3 mit Referenzen

Abbildung 4.10 zeigt die Korrelationsanalyse für das Objekt 3.

Der Signalverlauf *Messung Objekt 3* stellt das Vergleichssignal für Testobjekt 3 dar. Es wurde jeweils mit den Referenzsignalen aller Testobjekte korreliert.

Messung Objekt 3 / Referenz Objekt 1 Die Korrelation des Signals *Messung Objekt 3* mit dem Referenzsignal *Referenz Objekt 1* ergibt folgendes Ergebnis:

Ermittelter Punkt P_{max} : 253

Dies entspricht einer Verschiebung m_{max} beider Signale um:

$$m_{max} = N - P_{max} = 235 - 192 = -18$$

Für die Energiebeträge der Signale folgt mit [Gleichung 4.4](#):

$$E_{M_3} = 149392$$

$$E_{R_1} = 864198$$

Für das Ergebnis der Korrelation bei der Verschiebung m_{max} folgt:

$$r_{M_3R_1}[m_{max}] = \sum_{n=0}^{468} M_{3n} \cdot R_{1n+m_{max}} = \underline{242662} \quad (4.32)$$

Damit folgt für den normierten Korrelationsfaktor nach [Gleichung 4.7](#):

$$R_{M_3R_1}[m_{max}] = \frac{242662}{\sqrt{149392 \cdot 864198}} = \underline{\underline{0.675}} \quad (4.33)$$

Für die Symmetrie folgt:

$$Q_{rechts} = 18900495$$

$$Q_{links} = 10892479$$

Damit gilt für die Symmetrie:

$$a_{M_3R_1} = \frac{10892479}{18900495} = \underline{\underline{0,57}} \quad (4.34)$$

Messung Objekt 3 / Referenz Objekt 2 Die Korrelation des Signals *Messung Objekt 3* mit dem Referenzsignal *Referenz Objekt 2* ergibt folgendes Ergebnis:

Ermittelter Punkt P_{max} : 284

Dies entspricht einer Verschiebung m_{max} beider Signale um:

$$m_{max} = N - P_{max} = 235 - 284 = -18$$

Für die Energiebeträge der Signale folgt mit [Gleichung 4.4](#):

$$E_{M_3} = 149392$$

$$E_{R_2} = 500788$$

Für das Ergebnis der Korrelation bei der Verschiebung m_{max} folgt:

$$r_{M_3R_2}[m_{max}] = \sum_{n=0}^{468} M_{3n} \cdot R_{2n+m_{max}} = \underline{141663} \quad (4.35)$$

Damit folgt für den normierten Korrelationsfaktor nach [Gleichung 4.7](#):

$$R_{M_3R_2}[m_{max}] = \frac{141663}{\sqrt{142984 \cdot 500788}} = \underline{\underline{0,518}} \quad (4.36)$$

Für die Symmetrie folgt:

$$Q_{rechts} = 10377805$$

$$Q_{links} = 14509100$$

Damit gilt für die Symmetrie:

$$a_{M_3R_1} = \frac{10377805}{14509100} = \underline{\underline{0,71}} \quad (4.37)$$

Messung Objekt 3 / Referenz Objekt 3 Die Korrelation des Signals *Messung Objekt 3* mit dem Referenzsignal *Referenz Objekt 2* ergibt folgendes Ergebnis:

Ermittelter Punkt P_{max} : 244

Dies entspricht einer Verschiebung m_{max} beider Signale um:

$$m_{max} = N - P_{max} = 235 - 244 = -9$$

Für die Energiebeträge der Signale folgt mit [Gleichung 4.4](#):

$$E_{M_3} = 149392$$

$$E_{R_3} = 165173$$

Für das Ergebnis der Korrelation bei der Verschiebung m_{max} folgt:

$$r_{M_3R_3}[m_{max}] = \sum_{n=0}^{468} M_{3n} \cdot R_{3n+m_{max}} = \underline{142984} \quad (4.38)$$

Damit folgt für den normierten Korrelationsfaktor nach [Gleichung 4.7](#):

$$R_{M_3R_3}[m_{max}] = \frac{142984}{\sqrt{149392 \cdot 165173}} = \underline{\underline{0.910}} \quad (4.39)$$

Für die Symmetrie folgt:

$$Q_{rechts} = 5116817$$

$$Q_{links} = 5702668$$

Damit gilt für die Symmetrie:

$$a_{M_3R_3} = \frac{5116817}{5702668} = \underline{\underline{0,90}} \quad (4.40)$$

Auswertung

Sowohl der normierte Korrelationsfaktor als auch die Symmetrie sind bei der Korrelation vom Messsignal *Messung Objekt3* mit dem Referenzsignal *Referenz Objekt 3* am größten. Damit kann das gemessene Objekt eindeutig als Objekt 3 klassifiziert werden.

4.3.4. Abstandsmessung

Neben der Klassifizierung von Objekten soll der Sensor in der Lage sein, den Abstand eines Objektes relativ zum Sensor zu ermitteln. Dies kann durch eine "Time of flight" Messung per Korrelationsanalyse erfolgen. Da der Punkt, an dem die Korrelation zweier ähnlicher Signalverläufe maximal wird, direkt die zeitliche Verschiebung dieser Signale zueinander angibt, kann durch Ermittlung dieses Punktes der räumliche Abstand beider Signale bestimmt werden. Das nach der Aussendung eines Ultraschallsignals aufgenommene Echo aus 1500 Abtastwerten wird mit einem hinterlegten Referenzsignal korreliert. In dem Korrelationsergebnis wird nach der Position des Maximalwerts P_{max} gesucht. Mit der Abtastfrequenz f_a des ADC und der Schallgeschwindigkeit c lässt sich über diese Position der Abstand berechnen. Um ein möglichst genaues Ergebnis zu erhalten wird die Temperaturabhängigkeit der Schallgeschwindigkeit bei der Berechnung mit berücksichtigt.

Für den Abstand d gilt:

$$d = \frac{P_{max}}{f_a} \cdot \frac{c}{2} \quad (4.41)$$

Die theoretische Auflösung A der Abstandsmessung ist durch die Abtastrate des ADC gegeben:

$$A = \frac{c}{f_a} \quad (4.42)$$

für $c = 343 \frac{m}{s}$ und $f_a = 181 \text{ kHz}$:

$$A = \frac{343 \frac{m}{s}}{181000 \frac{1}{s}} = 1,89 \text{ mm} \quad (4.43)$$

5. Umsetzung in Software

Neben der Entwicklung der Hardware für den Ultraschallsensor umfasst diese Arbeit die Erstellung eines Softwareprogramms, um die Objektklassifizierung mit der im [Kapitel 4](#) beschriebenen Methode auf dem Mikrocontroller umzusetzen.

Die Erstellung der Software erfolgt in der Sprache "C". Um die Entwicklungsumgebung "Visual Studio 2012" von Microsoft nutzen zu können, wird das Freeware-Plugin "Visual Micro 1.1503.18" [\[Vis\]](#) verwendet. Die Erstellung der Software gestaltet sich dadurch deutlich komfortabler und übersichtlicher als mit der Arduino IDE-Entwicklungsumgebung. In diesem Kapitel wird ein Überblick über den Ablauf des Programms gegeben.

Die Aufgabe des Mikrocontrollers besteht zum Einen darin, die Sendeschaltung des Sensorsystems mit einem geeigneten Signal anzusteuern. Zum Anderen hat er die Aufgabe, das analoge Echosignal zu digitalisieren um dieses anschließend auf einem digitalen Signalprozessor zu verarbeiten. Für die Bearbeitung der verschiedenen Teilaufgaben des Mikrocontrollers wurden einzelne Funktionen geschrieben.

Insgesamt besteht das Programm aus fünf Funktionen, die folgende Aufgaben übernehmen:

Senden des Ultraschallsignals (*sendBurst*, Quellcode: [Abschnitt A.5](#)):

Schaltet die zuvor konfigurierte PWM-Peripherie für eine bestimmte Zeit ein.

Empfangen des Echosignals (*receive*, Quellcode: [Abschnitt A.3](#)):

Konfiguriert den Analog-Digital-Wandler und nimmt nach dem Aufruf eine bestimmte Menge an Werten auf. Als Parameter werden der Funktion die zu nutzende ADC-Kanal und ein Array übergeben, das per call-by-reference mit Werten gefüllt wird.

Ermitteln der Temperatur (*getTemperature*, Quellcode: [Abschnitt A.4](#)):

Konfiguriert den ADC und liest den Temperatursensor aus. Der Rückgabewert enthält die Temperatur in Grad Celsius.

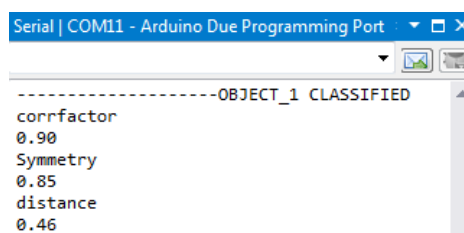
Ermitteln des Abstandes (*getDistance*, Quellcode: [Abschnitt A.2](#)):

Erhält als Übergabeparameter ein Array. Dieses Array wird mit einem hinterlegtem Array korreliert. Das Korrelationsergebnis wird nach dem Maximalwert durchsucht. Mit der Position des Maximalwerts und der durch den Aufruf der Funktion *getTemperature* ermittelten Temperatur wird ein Abstand berechnet. Die Funktion gibt den Abstand in Metern zurück.

Klassifizierung der Objekte (*objectclassification*, Quellcode: [Abschnitt A.1](#)):

Führt die Klassifizierung der Objekte durch. Nach dem Aufruf von *sendBurst* wird der Funktion *receive* ein leeres Array übergeben. Nach der Berechnung der Korrelationsergebnisse wird der Funktion *getDistance* das Array übergeben um den Abstand des Objekts zu erhalten. Der genaue Ablauf der Funktion ist [Abbildung 5.3](#) zu sehen.

Die Ergebnisse werden über die serielle Schnittstelle auf einem Monitor ausgegeben (siehe [Abbildung 5.1](#)).



```
Serial | COM11 - Arduino Due Programming Port
-----OBJECT_1 CLASSIFIED
corrfactor
0.90
Symmetry
0.85
distance
0.46
```

Abbildung 5.1.: Ausgabe serieller Monitor

[Abbildung 5.2](#) zeigt die Aufrufhierarchie des entwickelten Programms.

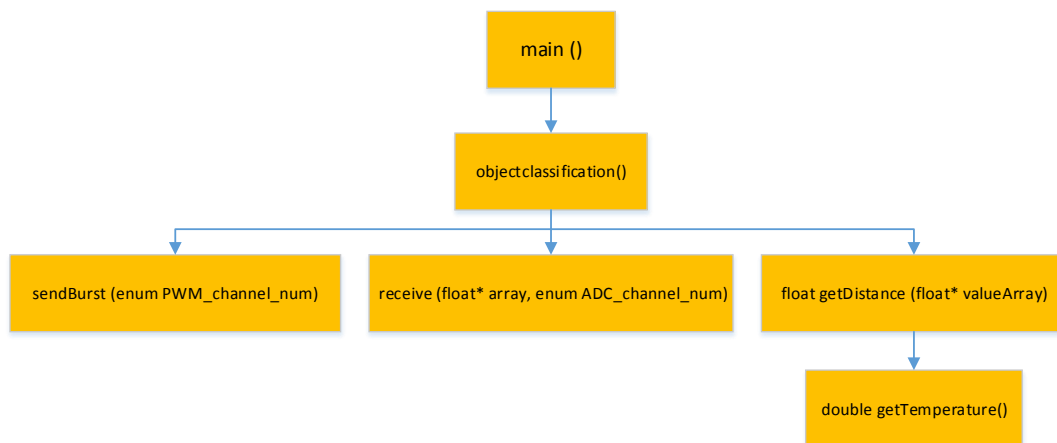


Abbildung 5.2.: Aufrufhierarchie

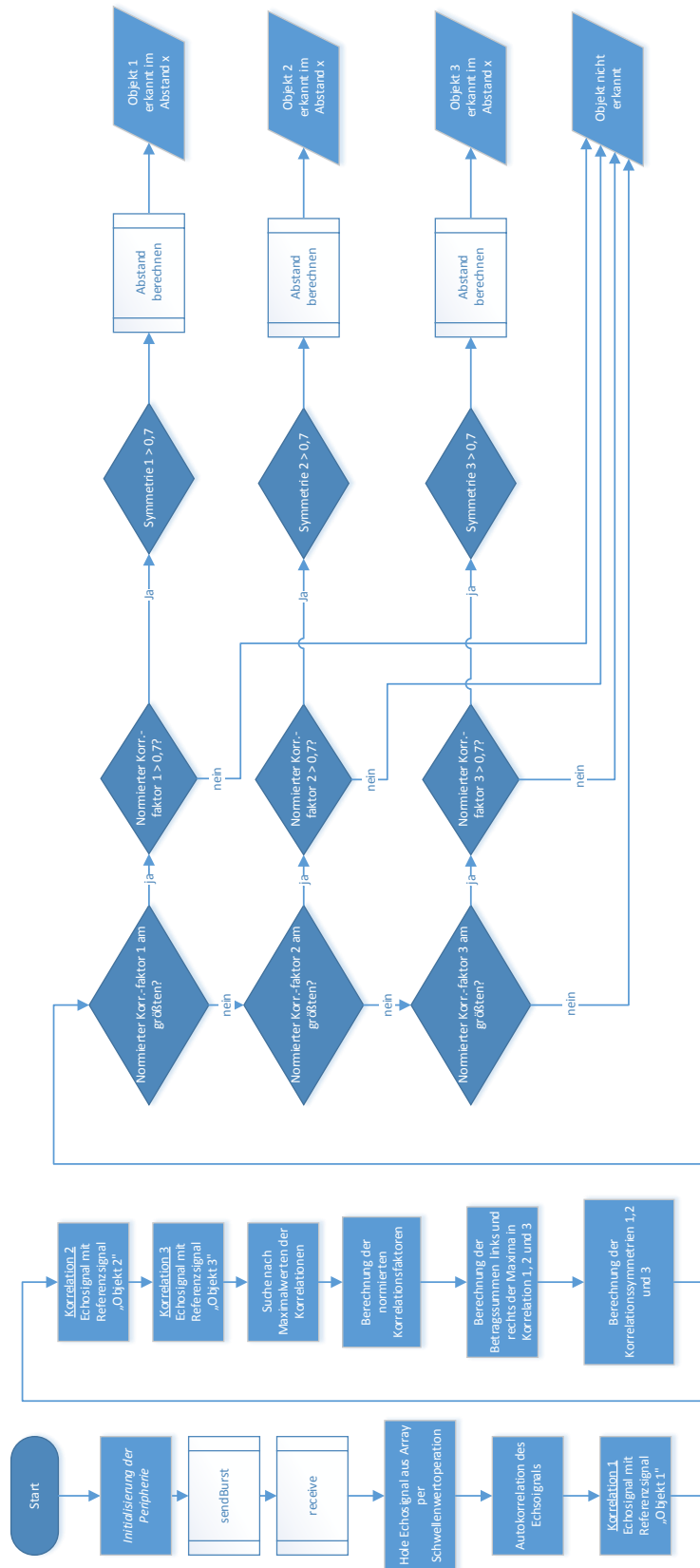


Abbildung 5.3.: Ablaufplan der Funktion "objectclassification"

6. Abschließender Systemtest

Um das System abschließend auf seine Fähigkeiten zu untersuchen, wird es einer Messreihe unterzogen. Dafür werden die Referenzechos der drei Objekte in einem Abstand von ca. 50cm aufgenommen. Der Sensor befindet sich auf einer Höhe von ca. 25cm über dem Boden. Jedes Objekt wird jeweils 10 mal in den Abständen 30cm, 60cm und 90cm einer Messung unterzogen. Zwischen jeder Messung wird das jeweilige Objekt aus dem Erfassungsbereich des Sensors entfernt und anschließend in der möglichst exakt gleichen Position erneut aufgestellt.

Die Klassifikation erfolgt anhand der Auswertung der normierten Korrelationsfaktoren und der Korrelationssymmetrien. Für den Systemtest werden folgende Kriterien gewählt:

- Eine Korrelation ergibt einen größeren normierten Korrelationsfaktor als die zwei anderen Korrelationen und weist einen Wert von mindestens 0,7 auf.
- Die Symmetrie dieser Korrelation ergibt einen Wert von mindestens 0,7.

Table 6.1 zeigt die Ergebnisse des Systemtests. Ein ✓ bedeutet: "Objekt erkannt". Ein X bedeutet: "Objekt nicht erkannt". Eine rote Zahl zeigt an, dass das System ein falsches Objekt erkannt hat.

Objekt/Abstand	30cm	60cm	90cm
1	✓✓✓✓✓✓✓✓✓✓	✓✓✓✓✓✓✓✓✓✓	✓✓✓✓✓✓✓✓✓✓
2	X ✓ 1 1 1 X X 1 X 1	✓✓✓✓✓✓✓✓✓✓	✓✓✓ X ✓✓✓✓✓✓✓✓
3	✓ X 2 2 ✓✓✓ 2 2 X	✓✓✓✓✓✓✓✓✓✓	✓✓✓✓✓✓✓✓✓ X ✓✓

Table 6.1.: Ergebnisse des Systemtests

Die Ergebnisse zeigen, dass alle Objekte in den Abständen 60cm und 90cm sehr zuverlässig erkannt werden. Einzelne Fehler können durch nicht exakt gleiches Positionieren zwischen den Messungen auftreten. Im Abstand von 30 cm wird nur Objekt 1 zuverlässig erkannt. Objekt 2 und 3 werden oft gar nicht, oder als falsches Objekt erkannt.

Diese Fehlmessungen treten auf, da sich die Echoprofile der Objekte 2- und 3 mit dem Abstand zum Sensor verändern. Das charakteristische "Doppelecho" von Objekt 2 (s. [Abbildung 4.4](#)) nimmt bei geringem Abstand zum Sensor die Gestalt des Echoprofils von Objekt 1 (s. [Abbildung 4.2](#)) an, da das Ultraschallsignal beim Verringern des Abstandes zunehmend nur von der Stirnfläche des abgesetzten Quaders reflektiert wird.

Der gleiche Effekt tritt bei der Messung von Objekt 3 auf. Das Echoprofil kann bei geringem Abstand zum Sensor die Gestalt des Echoprofils von Objekt 2 annehmen.

7. Zusammenfassung und Ausblick

Die Ausarbeitung dieser Arbeit umfasst die Entwicklung eines Ultraschallsensors mit Objektklassifizierung im Hinblick auf dessen Nutzung als Fahrerassistenzsystem.

Neben der analogen Schaltungsentwicklung für den Sende- und Empfangsbetrieb, umfasst die Ausarbeitung eine Analyse und anschließende Implementierung einer Methode für die digitale Verarbeitung von Ultraschallsignalen auf einem Mikrocontroller.

Die Schaltungsentwicklung berücksichtigt mehrere Konzepte vorhandener Ultraschallsysteme, die auf ihre Eigenschaften hin näher untersucht werden. Wegen der anpassbaren Sendespannung ist die Entscheidung auf die Trafosendesaltung gefallen(3.3.2). Zur Auswahl stehen Wandler in offener bzw. geschlossener Bauweise. Aufgrund der Sende- und Empfangsempfindlichkeit ist der offenen Wandler ausgewählt worden, obwohl der geschlossene Wandler für den Einsatz im Automobilbereich im Hinblick auf die Robustheit die bessere Lösung darstellen würde.

Zur Erkennung von Objekten wird die normierte Korrelationsmethode verwendet. Dieses ist eine Methode um die Ähnlichkeit zweier Signalverläufe signalenergie- und amplitudenunabhängig numerisch auszudrücken.

Um das Sensorsystem auf seine Fähigkeit der Objektklassifikation zu testen werden exemplarisch drei verschiedene Objekte ausgewählt, eine plane Wand (Objekt 1), eine Wand mit einem abgesetzten Quader (Objekt 2) und eine Treppe (Objekt 3). Hinterlegte Referenzsignale der 3 Objekte in 50 cm Abstand dienen der Analyse.

Bevor die Methode in Software auf dem Microcontroller implementiert wird, erfolgt eine Simulation mit der Software Scilab. Die Ergebnisse der Simulation lassen eine eindeutige Klassifikation aller drei Objekte zu.

Nach der Implementierung der Methode auf dem Mikrocontroller wird abschließend ein Systemtest durchgeführt. Dabei werden alle Objekte jeweils in den Abständen 30, 60 und 90 cm einer Korrelationsanalyse unterzogen. Die Ergebnisse zeigen, dass die drei Objekte in 60 und 90 cm Abstand eindeutig klassifiziert werden können, während bei einem Abstand von 30 cm die Signalform des Echos beeinflusst wird, was dazu führt, dass keine eindeutige Klassifikation der Objekte 2 und 3 bei diesem Abstand möglich ist. Das Echosignal von Objekt 1 ist gegenüber einer Veränderung des Abstandes weniger empfindlich und somit auch in dieser Entfernung eindeutig klassifizierbar. Ergebnis dieser Ausarbeitung ist nach der Definition von [Abschnitt 2.1](#) ein *Ultraschall-Inspektionssystem*.

Die Ergebnisse zeigen, dass das entwickelte Sensorsystem grundsätzlich in der Lage ist,

verschiedene Objekte anhand ihrer Ultraschallechos zu klassifizieren. Voraussetzung dafür ist die Reproduzierbarkeit der Echosignalforn. Diese Erkenntnis stellt für die Klassifizierung ein großes Problem dar, da sich durch die Änderung der Objektposition erfahrungsgemäß auch dessen Echosignalforn ändern kann.

Um das Sensorsystem als Fahrerassistenzsystem, z.B. für die Erkennung von Fußgängern einsetzen zu können bedarf es daher insbesondere einer Erweiterung der Anzahl an klassifizierbaren Objekten. Dies könnte z.B. durch die Verbesserung der Signalauswertung durch die Nutzung anderer Methoden geschehen. Eine weitere denkbare Funktion ist die Analyse von Bodenechos, um Echosignale von Objekten eindeutig von Echosignalen des Bodens zu unterscheiden. Neben der Weiterentwicklung der Signalauswertung, ist es erforderlich das System an die raue Umgebung im Straßenverkehr anzupassen und außerdem als Ein-Wandler System mit kombiniertem Sende- und Empfangspfadler auszuführen.

Grundsätzlich ist das entstandene Sensorsystem als Grundlage für der Entwicklung weiterer Funktionen auch neben der Objektklassifizierung anzusehen. So ist es z.B. denkbar, durch die Auswertung der Echosignale unter der Berücksichtigung des Doppler-Effekts Geschwindigkeiten von Objekten, die sich auf den Sensor zubewegen, zu messen und sicherheitskritisch zu bewerten.

A. Quellcode

A.1. Objectclassification

```
1
2 void Objectclassification(void)
3 {
4
5 // signals of reference objects , sampled with 181kHz, 10bit resolution
6
7
8 static float32_t reference_object1_f32[235]=
9 {...};
10
11 static float32_t reference_object2 [235]=
12 {...};
13
14 static float32_t reference_object3 [235]=
15 {...};
16
17 float32_t valueArray_f32[1500]={0};
18 float32_t signalArray_f32 [235]={0};
19
20 //correlation-Arrays
21 float32_t corrArray_object1_f32[469]={0};
22 float32_t corrArray_object2_f32[469]={0};
23 float32_t corrArray_object3_f32[469]={0};
24 float32_t autocorrSignal_f32[469]={0};
25
26 //max-values of autocorrelation of Objects , used for scaling
27 const float maxVal_autocorrObj1= 2080153.00;
28 const float maxVal_autocorrObj2= 318841.00;
29 const float maxVal_autocorrObj3= 184531.00;
30
31 // max-values of correlation signal with objects
32 float32_t maxVal_object1=0;
33 float32_t maxVal_object2=0;
34 float32_t maxVal_object3=0;
35
36 // max-value of autocorrelation of signal
```

```
37 float32_t maxVal_autocorrSignal=0;
38
39 // distance
40 float32_t distance=0;
41
42 // normed correlationfactors
43 float32_t corrObj1=0;
44 float32_t corrObj2=0;
45 float32_t corrObj3=0;
46
47 //length of correlated signals
48 uint32_t srcALen= 235;
49 uint32_t srcBLen= 235;
50
51 // control variable
52 int l=0;
53 int j=0;
54 int m=0;
55
56 // deadtime offset
57 int i=110;
58
59 // positions of max-values in correlation-arrays
60 int maxPosObj1=0;
61 int maxPosObj2=0;
62 int maxPosObj3=0;
63
64 // left and right averages of correlation
65 double rightAverageObject1=0;
66 double leftAverageObject1=0;
67
68 double rightAverageObject2=0;
69 double leftAverageObject2=0;
70
71 double rightAverageObject3 = 0;
72 double leftAverageObject3 = 0;
73
74 //symmetry of correlation
75 double symmetryObject1=0;
76 double symmetryObject2=0;
77 double symmetryObject3 =0;
78
79 const int n=235;
80
81 sendBurst(PWM_CHANNEL_0);
82
83 receive(valueArray_f32 , ADC_CHANNEL_0);
84
```

```
85 while ((valueArray_f32[i] <= 540) && (valueArray_f32[i] >= 480))
86     {
87         i++;
88     }
89
90 i = i - 5;
91
92 // filling of the signalArray with the DC compensated signal
93
94 for (j = 0; j <= 234; j++)
95     {
96         signalArray_f32[j] = ((valueArray_f32[i] - 512));
97         i++;
98     }
99
100
101 // Correlation of signal with reference signals
102
103 arm_correlate_f32(signalArray_f32, srcALen, reference_object1_f32, srcBLen,
104                 corrArray_object1_f32);
105
106 arm_correlate_f32(signalArray_f32, srcALen, reference_object2, srcBLen,
107                 corrArray_object2_f32);
108
109 arm_correlate_f32(signalArray_f32, srcALen, reference_object3, srcBLen,
110                 corrArray_object3_f32);
111
112 // Autocorrelation of signal
113
114 arm_correlate_f32(signalArray_f32, srcALen, signalArray_f32, srcBLen,
115                 autocorrSignal_f32);
116
117 // search for maximum values in correlation-arrays
118
119 for (l = 0; l <= n*2 - 2; l++)
120     {
121         if (maxVal_object2 <= corrArray_object2_f32[l])
122             {
123                 maxVal_object2 = corrArray_object2_f32[l];
124                 maxPosObj2 = l;
125             }
126
127         if (maxVal_object3 <= corrArray_object3_f32[l])
128             {
129                 maxVal_object3 = corrArray_object3_f32[l];
130                 maxPosObj3 = l;
131             }
132     }
```

```
129         }
130
131     if (maxVal_object1 <= corrArray_object1_f32[l])
132     {
133         maxVal_object1 = corrArray_object1_f32[l];
134         maxPosObj1 = l;
135     }
136
137
138     if (maxVal_autocorrSignal <= autocorrSignal_f32[l])
139     {
140         maxVal_autocorrSignal = autocorrSignal_f32[l];
141     }
142
143 }
144
145
146 // calculation of normed correlationfactors
147
148 corrObj2 = maxVal_object2 / (sqrt(maxVal_autocorrObj2 * maxVal_autocorrSignal));
149
150 corrObj1 = maxVal_object1 / (sqrt(maxVal_autocorrObj1 * maxVal_autocorrSignal));
151
152 corrObj3 = maxVal_object3 / (sqrt(maxVal_autocorrObj3 * maxVal_autocorrSignal));
153
154 // calculation of the quadratic mean besides left and right from the position of
155 // the maximum
156 for (l = maxPosObj2 - 1; l >= 0; l--)
157 {
158     leftAverageObject2 = leftAverageObject2 + corrArray_object2_f32[l]
159     * corrArray_object2_f32[l] / 10000;
160 }
161 for (l = maxPosObj2 + 1; l <= 468; l++)
162 {
163     rightAverageObject2 = rightAverageObject2 + corrArray_object2_f32[l]
164     * corrArray_object2_f32[l] / 10000;
165 }
166 for (l = maxPosObj1 - 1; l >= 0; l--)
167 {
168     leftAverageObject1 = leftAverageObject1 + corrArray_object1_f32[l] *
169     corrArray_object1_f32[l] / 10000;
170 }
171 for (l = maxPosObj1 + 1; l <= 468; l++)
172 {
```



```
173         rightAverageObject1=rightAverageObject1 + corrArray_object1_f32[l
           ]*corrArray_object1_f32[l]/10000;
174     }
175
176     for (l=maxPosObj3-1;l >=0;l--)
177     {
178         leftAverageObject3=leftAverageObject3+ corrArray_object3_f32[l]*
           corrArray_object3_f32[l]/10000;
179     }
180
181     for (l=maxPosObj3+1;l <=468;l++)
182     {
183         rightAverageObject3=rightAverageObject3+ corrArray_object3_f32[l
           ]*corrArray_object3_f32[l]/10000;
184     }
185
186
187     // calculation of the quadratic-mean ratio
188
189     if (leftAverageObject2 <=rightAverageObject2)
190     {
191         symmetryObject2= leftAverageObject2 /rightAverageObject2 ;
192     }
193
194     else
195     {
196         symmetryObject2 =rightAverageObject2 /leftAverageObject2 ;
197     }
198
199     if (leftAverageObject3 <=rightAverageObject3)
200     {
201         symmetryObject3= leftAverageObject3 /rightAverageObject3 ;
202     }
203
204     else
205     {
206         symmetryObject3 =rightAverageObject3 /leftAverageObject3 ;
207     }
208
209     if (leftAverageObject1 <=rightAverageObject1)
210     {
211         symmetryObject1= leftAverageObject1 /rightAverageObject1 ;
212     }
213
214     else
215     {
216         symmetryObject1= rightAverageObject1 /leftAverageObject1 ;
217     }
```

```
218
219 Serial.print("—————");
220
221 if ((corrObj3>corrObj1)&&(corrObj3>corrObj2)&&(corrObj3 >=0.7)&&(symmetryObject3
    >=0.7))
222     {
223         distance= getDistance(valueArray_f32);
224         Serial.println("OBJECT_3_CLASSIFIED");
225         Serial.println("corrfactor");
226         Serial.println(corrObj3,2);
227         Serial.println("symmetry");
228         Serial.println(symmetryObject3);
229         Serial.println("distance");
230         Serial.println(distance);
231     }
232
233 else if ((corrObj2>corrObj1)&&(corrObj2>corrObj3)&&(corrObj2 >=0.7)&&(
    symmetryObject2 >=0.7))
234     {
235         distance= getDistance(valueArray_f32);
236         Serial.println("OBJECT_2_CLASSIFIED");
237         Serial.println("corrfactor");
238         Serial.println(corrObj2,2);
239         Serial.println("symmetry");
240         Serial.println(symmetryObject2);
241         Serial.println("distance");
242         Serial.println(distance);
243     }
244
245 else if ((corrObj1>corrObj2)&&(corrObj1>corrObj3)&&(corrObj1 >=0.7)&&(
    symmetryObject1 >=0.7))
246     {
247         distance= getDistance(valueArray_f32);
248         Serial.println("OBJECT_1_CLASSIFIED");
249         Serial.println("corrfactor");
250         Serial.println(corrObj1,2);
251         Serial.println("symmetry");
252         Serial.println(symmetryObject1);
253         Serial.println("distance");
254         Serial.println(distance);
255     }
256
257 else
258     {
259         Serial.println("OBJECT_NOT_DETECTED");
260     }
261 }
```

A.2. getDistance

```
1 float32_t getDistance (float32_t* valueArray)
2
3 {
4     const int blockSizeA=1500;
5     const int blockSizeB=235;
6
7     //length of correlated signals
8     const uint32_t srcALen=1500;
9     const uint32_t srcBLen=235;
10
11     int i=0;
12     //value and position of maximum in correlation-array
13
14     int maxPos=0;
15     int maxValue=0;
16
17     float32_t valueArray_f32_normed[1500]={0};
18     float32_t distance=0;
19
20     q15_t valueArray_q15[1500]={0};
21     q15_t corrArray[2999]={0};
22
23     //reference signal for correlation
24     static q15_t referenceSignal_q15[235]=
25     {...};
26
27     //Offsetcompensation of valueArray: -512 to compensate DC offset
28     //Division by 5000 to save as q15_t
29
30     for (i=0;i <=1500;i++)
31     {
32         valueArray_f32_normed[i]= (valueArray[i]-512)/5000;
33     }
34
35     // save Arrays as q_15
36
37     arm_float_to_q15(valueArray_f32_normed ,valueArray_q15 ,blockSizeA);
38
39     // correlate valueArray with reference signal and save result into corrArray
40
41     arm_correlate_fast_q15(valueArray_q15 ,srcALen ,referenceSignal_q15 ,srcBLen ,
42         corrArray);
43
44     //search corrArray for biggest value to get the distance
45
46     for (i=1500;i <=2998;i++)
```

```
46     {
47         if (corrArray[i] >= maxValue)
48             {
49                 maxPos = i;
50                 maxValue = corrArray[i];
51             }
52     }
53
54 // calculate distance
55
56 distance = ((maxPos - 1499) / 181E3) * sqrt(1.4 * (8.314 * (getTemperature() + 273.15))
57         / (28.949E-3)) / 2;
58 return (distance);
59 }
```

A.3. receive

```
1 void receive (float32_t* pArray, enum adc_channel_num_t adc_ch)
2
3 {
4
5     adc_disable_all_channel(ADC);
6
7     // set ADC to 181kHz samplerate, 10bit resolution
8
9     REG_ADC_MR = ADC_MR_STARTUP_SUT8 | ADC_MR_TRGEN_DIS | ADC_MR_LOWRES_BITS_10 |
10         ADC_MR_FREERUN_ON | ADC_MR_PRESCAL(10) | ADC_MR_SETTLING_AST3;
11
12     adc_enable_channel(ADC, adc_ch); // enable ADC channel
13
14     float32_t analogValue;
15     uint32_t EOC_bit = 0;
16     int i = 0;
17
18     if (adc_ch == 0)
19         { EOC_bit = ADC_ISR_EOC0; }
20
21     else if (adc_ch == 1)
22         { EOC_bit = ADC_ISR_EOC1; }
23
24     else
25         { Serial.println("wrong_ADC_channel_number, choose_channel_0_or_1"); }
26
27     for (i = 0; i <= 1499; i++)
28     {
29         adc_start(ADC); // start ADC conversion
```

```
29
30         while (!(ADC->ADC_ISR & EOC_bit));
31
32         analogValue = ADC->ADC_CDR[adc_ch];
33         pArray[i] = analogValue;
34     }
35
36     adc_disable_channel(ADC, adc_ch); // disable ADC channel
37     return;
38 }
```

A.4. getTemperature

```
1
2 double getTemperature (void)
3 {
4
5     REG_ADC_MR = ADC_MR_STARTUP_SUT8 | ADC_MR_TRGEN_DIS | ADC_MR_LOWRES_BITS_12 |
6         ADC_MR_FREERUN_ON | ADC_MR_PRESCAL(20) | ADC_MR_SETTLING_AST3;
7
8     uint32_t ADC_temperature_value=0;
9
10    adc_enable_ts(ADC);
11    adc_enable_channel(ADC, ADC_TEMPERATURE_SENSOR);
12
13    float trans = 3.3/4096;
14    float offset = 0.8;
15    float factor = 0.0256;
16    int fixtemp = 27;
17
18    adc_start(ADC);
19
20    while (!(ADC->ADC_ISR & ADC_ISR_EOC15));
21
22    ADC_temperature_value= ADC->ADC_CDR[ADC_TEMPERATURE_SENSOR];
23
24    double treal = 27 + (( (3.3/4096) * ADC_temperature_value ) - 0.8 ) / 0.0265;
25
26    adc_disable_channel(ADC, ADC_TEMPERATURE_SENSOR);
27
28    return treal;
29 }
```

A.5. sendBurst

```

1 void sendBurst(enum _pwm_ch_t pwm_ch) //sends burst of 40kHz
2 {
3
4 pwm_channel_enable(PWM, pwm_ch);
5 delayMicroseconds(185);
6 pwm_channel_disable(PWM, pwm_ch);
7
8 }

```

A.6. Initialisierung

```

1
2 #define ARM_MATH_CM3 // define for arm_math.h
3
4 #include <math.h>
5 #include <stdio.h>
6 #include <assert.h>
7 #include <stdint.h>
8 #include "pwm.h"
9 #include <arm_math.h> //CMSIS DSP library
10 #include "Arduino.h"
11 #include <core_cm3.h>
12
13
14 void setup() {
15
16     pinMode(34, OUTPUT); // configure PIN 34 as Output
17     pinMode(36, OUTPUT); // configure PIN 36 as Output
18
19     Serial.begin(115200);
20
21     //ADC SETTINGS—————
22
23     NVIC_EnableIRQ(ADC_IRQn); //enable general Interrupt for
24     ADC
25
26     //PWM SETTINGS—————
27
28     pmc_enable_periph_clk (PWM_INTERFACE_ID) ; // turn on clocking to PWM
29     unit
30     pwm_channel_t pwm_channel_instance;
31     pwm_channel_disable(PWM, PWM_CHANNEL_0);
32     pwm_channel_disable(PWM, PWM_CHANNEL_1);
33
34     pwm_clock_t clock_setting = {

```

```
33         .ul_clka = 2000 * 3950,                                // set
           PWM frequency
34         .ul_clkb = 0,
35         .ul_mck = SystemCoreClock,
36     };
37
38     pwm_init(PWM, &clock_setting);
39
40     // initialize PWM channel 0
41     pwm_channel_instance.ul_prescaler = PWM_CMR_CPRE_CLKA;
42     pwm_channel_instance.ul_period = 100;
43     pwm_channel_instance.ul_duty = 50;
44     pwm_channel_instance.channel= PWM_CHANNEL_0;
45     pwm_channel_init(PWM, &pwm_channel_instance);
46
47     // initialize PWM channel 1
48     pwm_channel_instance.ul_prescaler = PWM_CMR_CPRE_CLKA;
49     pwm_channel_instance.ul_period = 100;
50     pwm_channel_instance.ul_duty = 50;
51     pwm_channel_instance.channel= PWM_CHANNEL_1;
52     pwm_channel_init(PWM, &pwm_channel_instance);
53
54     /// Configure pin 34 (PC2) and pin 36 (PC4) to be driven by peripheral B
55     (PWM channel 0 L) and (PWM channel 1 L)
56
57     PIOC->PIO_PDR = 0x14 ;
58     // disable PIO control
59     PIOC->PIO_IDR = 0x14 ;
60     // disable PIO interrupts
61     PIOC->PIO_ABSR |= PIO_ABSR_P2| PIO_ABSR_P4 ;           // switch to B
62     peripheral
63 }
```

Literaturverzeichnis

- [Ana15] ANALOG DEVICES, INC.: *Ultrasonic Distance Measurement CN0343*, 2015. URL: <http://www.analog.com/en/design-center/reference-designs/hardware-reference-design/circuits-from-the-lab/cn0343.html#rd-overview> , [zuletzt abgerufen am 10.05.2015].
- [Ard15] ARDUINO: *Arduino Due*, 2015. URL: <http://www.arduino.cc/en/pmwiki.php?n=Main/ArduinoBoardDue>, [zuletzt abgerufen am 21.05.2015].
- [ARM15] ARM LTD.: *CMSIS DSP Software Library*, 2015. URL: <http://www.keil.com/pack/doc/CMSIS/DSP/html/index.html>, [zuletzt abgerufen am 10.05.2015].
- [Atm15] ATMEL CORPORATION: *SAM3X / SAM3A Series Datasheet*, 2015. URL: http://www.atmel.com/Images/Atmel-11057-32-bit-Cortex-M3-Microcontroller-SAM3X-SAM3A_Datasheet.pdf, [zuletzt abgerufen am 21.05.2015].
- [Bab13] BABIEL, GERHARD: *Bordnetze und Powermanagement*. Springer, 2013.
- [Bon97] BONFIG, KARL WALTER: *Sensorik, Sensoren und Sensorsignalverarbeitung Bd. 8*. expert Verlag, 1997.
- [Die14] DIETZ SENSORTECHNIK: *Ultraschallsensoren*, 2014. URL: http://www.dietz-sensortechnik.de/img_info/ultra_schallpuls.jpg, [zuletzt abgerufen am 01.05.2015].
- [Emi14] EMIL'S PROJECTS: *Making a better HC-SR04 Echo Locator*, 2014. URL: http://uglyduck.ath.cx/ep/archive/2014/01/Making_a_better_HC_SR04_Echo_Locator.html, [zuletzt abgerufen am 19.03.2015].
- [HS11] HESSE, STEFAN und GERHARD SCHELL: *Sensoren für die Prozess- und Fabrikautomation*. Vieweg+Teubner, 2011.

- [Int01] INTERNATIONAL RECTIFIER: *IRLML6346 30V Single N-Channel HEXFET Power MOSFET Datasheet*, 2001. URL: <http://www.irf.com/product-info/datasheets/data/irlml6346pbf.pdf>, [zuletzt abgerufen am 21.05.2015].
- [LGW09] LERCH, REINHARD, MARTIN GERHARD und DIETRICH WOLF: *Technische Akustik: Grundlagen und Anwendungen*. Springer, 2009.
- [Max] MAXIM INTEGRATED: *MAXQ7667 16-Bit, RISC, Microcontroller-Based, Ultrasonic Distance-Measuring System*. URL: <http://www.maximintegrated.com/en/products/digital/microcontrollers/MAXQ7667.html> [zuletzt abgerufen am 22.04.2015].
- [Mur11] MURATA MANUFACTURING CO.: *Specification of Ultrasonic Transducer Type: MA40MF14-0B*, 2011. URL: <http://www.murata.com/~media/webrenewal/products/sensor/ultrasonic/proof/ma40mf14-0b-e.ashx?la=en-gb>, [zuletzt abgerufen am 10.05.2015].
- [OL14] OHM, JENS-RAINER und HANS DIETER LÜKE: *Signalübertragung : Grundlagen der digitalen und analogen Nachrichtenübertragungssysteme*. Springer Berlin Heidelberg, 2014.
- [Pic12] PICO TECHNOLOGY LTD.: *PicoScope® 3000-Serie*, 2012. URL: <https://www.picotech.com/download/datasheets/PicoScope3400-de.pdf>, [zuletzt abgerufen am 21.05.2015].
- [Pro05a] PRO-WAVE ELECTRONICS CORPORATION: *Air Ultrasonic Ceramic Transducers 400ST/R160*, 2005. URL: <http://www.prowave.com.tw/english/products/ut/open-type/400s160.htm>, [zuletzt abgerufen am 10.05.2015].
- [Pro05b] PRO-WAVE ELECTRONICS CORPORATION: *K4000004 Transformer Datasheet*, 2005. URL: <http://www.prowave.com.tw/pdf/Transformer.pdf>, [zuletzt abgerufen am 21.05.2015].
- [Rei10] REIF, KONRAD: *Fahrstabilisierungssysteme und Fahrerassistenzsysteme*. Vieweg + Teubner ISBN: 978-3-8348-1314-5, 2010.
- [Sie15] SIEBENPFEIFFER, WOLFGANG: *Fahrerassistenzsysteme und effiziente Antriebe*. Springer Vieweg, 2015.
- [SS14] SINAMBARI, GH. REZA und STEFAN SENTPALI: *Ingenieurakustik*. Springer Vieweg, 2014.

- [Tex01] TEXAS INSTRUMENTS INC.: *TLC274C precision Quad Operational Amplifier Datasheet*, 2001. URL: <http://www.ti.com/general/docs/lit/getliterature.tsp?literatureNumber=slos092d>, [zuletzt abgerufen am 21.05.2015].
- [Tex14] TEXAS INSTRUMENTS INC.: *MAX232x Dual EIA-232 Drivers/Receivers (Rev. M)*, 2014. URL: <http://www.ti.com/lit/gpn/MAX232>, [zuletzt abgerufen am 02.05.2015].
- [TR14] TRÄNKLER, HANS-ROLF und LEONHARD(HRSG.) REINDL: *Sensortechnik: Handbuch für Praxis und Wissenschaft*. Springer Vieweg, 2014.
- [TSG12] TIETZE, ULRICH, CHRISTOPH SCHENK und EBERHARD GAMM: *Halbleiter-Schaltungstechnik*. Springer Vieweg, 2012.
- [Vis] VISUAL MICRO LIMITED: *Arduino IDE for Microsoft Visual Studio and Atmel Studio*. URL: www.visualmicro.com, [zuletzt abgerufen am 02.05.2015].

Versicherung über die Selbstständigkeit

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

Hamburg, 25. Mai 2015

Ort, Datum

Unterschrift