
BACHELORARBEIT

Herr
Wenzhe Shen

**Entwicklung und Umsetzung
eines System zur Messung
von Winkeln mittels zweier
Ultraschallsensoren mit
Mikrokontroller zur
Auswertung**

Mittweida, 2022

BACHELORARBEIT

Entwicklung und Umsetzung eines System zur Messung von Winkeln mittels zweier Ultraschallsensoren mit Mikrokontroller zur Auswertung

Autor:

Herr Wenzhe Shen

Studiengang:

Elektro- und Informationstechnik

Seminargruppe:

EI18wA-BC

Erstprüfer:

Prof. Dr. -Ing. Michael Kuhl

Zweitprüfer:

M. Sc. Kevin Blümel

Einreichung:

Mittweida, 02.08.2022

Verteidigung/Bewertung:

Mittweida, 2022

BACHELOR THESIS

Development and implementation of a system for measuring angles by means of two ultrasonic sensors with microcontroller for evaluation

author:

Mr. Wenzhe Shen

course of studies:

Electric Information Technology

seminar group:

EI18wA-BC

first examiner:

Prof. Dr. -Ing. Michael Kuhl

second examiner:

M. Sc. Kevin Blümel

submission:

Mittweida, 02.08.2022

defence/ evaluation:

Mittweida, 2022

Bibliografische Beschreibung:

Shen,Wenzhe

Entwicklung und Umsetzung eines Systems zur Messung von Winkeln mittels zweier Ultraschallsensoren mit Mikrokontroller zur Auswertung - 2022

Development and implementation of a system for measuring angles by means of two ultrasonic sensors with microcontroller for evaluation - 2022

Seiten 46, Mittweida, Hochschule Mittweida, Fakultät Ingenieurwissenschaften, Bachelorarbeit, 2022

Referat:

Ziel dieser Arbeit ist es, den Winkel eines Objekts mit zwei Ultraschallsensoren zu messen. Die Entfernung des Objekts wird zunächst von den beiden Ultraschallsensoren gemessen und dann der Winkel mit Hilfe der trigonometrischen Formel berechnet.

Die Bachelorarbeit entstand in der Zeit der Corona-Pandemie hauptsächlich im Home-Office.

Inhalt

Inhalt.....	I
Abbildungverzeichnis.....	III
Tabellenverzeichnis.....	IV
Abkürzungsverzeichnis.....	V
1 Einleitung.....	1
2 Aufgabenstellung.....	2
3 These.....	3
4 Stand der Technik.....	4
4.1 Ultraschallsensor.....	4
4.2 Vorteile eines mikrocontrollerbasierten Ultraschallsensor.....	6
5 Konzept des Systementwurfs.....	8
5.1 Gesamtkonzept.....	8
5.2 Berücksichtigung von Temperaturfaktoren.....	9
5.3 Interpretation von Winkelmessungen.....	10
5.4 Hardware und Software.....	11
5.4.1 Arduino Mega 2560.....	11
5.4.2 STM32L053 Mikrocontroller.....	12
5.4.3 Ultraschallsensor CH201.....	13
5.4.4 Hardwarekonzept.....	18
5.4.5 Softwarekonzept.....	18
6 Erläuterungen zu einigen Codes.....	21
7 Inbetriebnahme.....	24

Inhalt	II
7.1	Für die Messung verwendete Werkzeuge..... 24
8.	Experimentelle Messungen..... 26
8.1	Durchführung von Messverfahren..... 26
8.2	Analyse der Versuchsdaten..... 28
8.3	Analyse der experimentellen Ergebnisse..... 33
9	Zusammenfassung und Ausblick..... 34
	Literaturverzeichnis..... 35
	Selbstständigkeitserklärung

Abbildungsverzeichnis

- Abbildung 1: Ultraschallsensor CH201
- Abbildung 2: Zwei Ultraschallsensoren mit Mikrocontroller-Platine
- Abbildung 3: Flussdiagramm des Gesamtkonzepts
- Abbildung 4: Schallgeschwindigkeit in Abhängigkeit von der Lufttemperatur
- Abbildung 5: Formel für Schallgeschwindigkeit und Temperatur
- Abbildung 6: Schematische Winkelmessung
- Abbildung 7: Arduino Mega 2560
- Abbildung 8: Vorderseite von STM32L053 Mikrocontroller
- Abbildung 9: Rückseite von STM32L053 Mikrocontroller
- Abbildung 10: Ultraschallsensor CH201
- Abbildung 11: Single-Sensor Pulse-Echo
- Abbildung 12: ADC-Ausgang V_{RX} und verarbeitetes A-Scan-Signal $A(t)$
- Abbildung 13: In Hterm angezeigte I/Q-Daten
- Abbildung 14: In Hterm angezeigte I/Q-Daten
- Abbildung 15: Blockdiagramm des Hardwarekonzepts
- Abbildung 16: Flussdiagramm des Softwarekonzept
- Abbildung 17: Auswahl der erforderlichen Bibliothek
- Abbildung 18: Code für Baudrate der seriellen Schnittstelle
- Abbildung 19: Code zur Berechnung der Entfernung
- Abbildung 20: Code für die Amplitudenberechnung
- Abbildung 21: Code für Amplitudenvergleich
- Abbildung 22: Code für die Winkelberechnung
- Abbildung 23: Maßband
- Abbildung 24: Winkelmesser
- Abbildung 25: Tatsächliche Messung von Entfernungen und Winkeln von Objekten
- Abbildung 26: Daten vom seriellen Monitor
- Abbildung 27: Ergebnisse der tatsächlichen Messungen
- Abbildung 28: Immer wiederkehrende Daten
- Abbildung 29: Dreieckstheorem
- Abbildung 30: Schema der Messung
- Abbildung 31: Theoretisches Diagramm

Tabellenverzeichnis

Tabelle 1: Erste Messreihe (Teil 1)

Tabelle 2: Erste Messreihe (Teil 2)

Tabelle 3: Zweite Messreihe (Teil 1)

Tabelle 4: Zweite Messreihe (Teil 2)

Abkürzungsverzeichnis

IDE.....	Integrated Development Environment
I ² C.....	Inter-Integrated Circuit
I/O - Port.....	Input/Output – Port
LED.....	Light Emitting Diode
PC.....	Personal Computer
GND.....	Grund
VCC.....	Voltage of circuit

1. Einleitung

Mit der ständigen Weiterentwicklung der Technologie hat die Ultraschalltechnologie wichtige Anwendungen in vielen Bereichen wie Medizin, Militär, Industrie usw. gefunden. Es kann für die Ultraschallchirurgie, Abstandsmessung, Geschwindigkeitsmessung, Reinigung, Lithotripsie usw. verwendet werden. Die häufigste Anwendung im Leben ist die Verwendung von Ultraschall-Entfernungsmessung. Ein Beispiel ist das Einparkhilfesystem in Autos, das mit Hilfe von Ultraschall umliegende Objekte erkennt und den Fahrer akustisch warnt. Die Vorteile sind, dass es einfach zu bedienen ist und anstelle von Menschen für Messaufgaben in komplexen Umgebungen eingesetzt werden kann. Der weit verbreitete Einsatz von Ultraschalltechnologie scheint also wichtig zu sein.

Die Bestimmung der Position eines Objekts erfordert nicht nur die Kenntnis seiner Entfernung, sondern auch die Bestimmung seines genauen Winkels. Daher ist es sinnvoll, den Einsatz von zwei Ultraschallsensoren zur Messung von Winkeln zu untersuchen.

2. Aufgabenstellung

Ziel der Arbeit ist die prototypische Aufbau eines Systems zur Messung von Winkeln bzw. Entfernungen zwischen dem Sensorsystem und einem zu messenden Objektes mittels zweier Ultraschallsensoren mit Mikrokontroller zur Auswertung über ein Triangulationsverfahren. Um dieses Ziel zu erreichen, müssen folgende Aufgaben erledigt werden:

- Einarbeitung in die Thematik
- Erarbeitung eines Systemkonzeptes (Hardware, Software, realer Aufbau) unter Nutzung eines vorgegebenen Mikrokontrollers, ggf. Übergabe der Parameter über einen PC
- Entwurf und Umsetzung der Schaltung inklusive eigener Leiterplatte
- Durchführung und Auswertung von Basisversuchen
- Bewertung, Darstellung und Dokumentation der Projektergebnisse

3. These

Annahme: Der Sensor sendet Ultraschallwellen aus, die reflektiert werden, wenn sie auf ein Objekt treffen. Die Schallgeschwindigkeit ist bekannt, und wenn die Zeit zwischen Aussendung und Empfang bekannt ist, kann die Schallgeschwindigkeit mit der Zeit multipliziert und durch zwei geteilt werden, um die Entfernung zu ermitteln, d. h. die Entfernung zwischen dem Objekt und dem Sensor. Wenn der Abstand mit beiden Sensoren gemessen wird, kann der Winkel des Objekts mit Hilfe der trigonometrischen Formel berechnet werden.

4. Stand der Technik

4.1 Ultraschallsensor

Der Ultraschallsensor ist ein Wegaufnehmer und spezieller Sensor, der den Ultraschall verwendet, um Abstände von Objekten oder Füllstände zu messen. Ultraschallsensoren kommen sehr häufig in der Industrie vor, da sie millimetergenaue Messungen erlauben. [1] Zu Beginn des 20. Jahrhunderts, nach der Entdeckung des piezoelektrischen und des antipiezoelektrischen Effekts in der Physik, fand man eine Methode, um Ultraschall mit Hilfe von Elektronik zu erzeugen, und von da an wurde ein neues Kapitel in der Geschichte der Entwicklung und Förderung der Ultraschalltechnologie aufgeschlagen. Der Ultraschallsensor TDK InvenSense CH201 ist ein Sensor, der Signale sendet und empfängt. Er wird häufig in Mikrocontrollern wie dem STM32L053 verwendet. Er hat den Vorteil, dass er kleiner und stromsparender ist als herkömmliche Ultraschallsensoren und dass er über größere Entfernungen messen kann. Wenn dieser Ultraschallsensor funktioniert, sendet er regelmäßig einen kurzen Hochfrequenzimpuls aus, der sich mit Schallgeschwindigkeit durch die Luft bewegt und sofort reflektiert wird und als Echo zum Ultraschallsensor zurückkehrt, wenn er mit einem Objekt in Berührung kommt. Der Ultraschallsensor berechnet dann die Entfernung zum Objekt, indem er die Zeit zwischen der Aussendung des Schallimpulses und dem Empfang des Echosignals durch den Mikrocontroller berechnet.



Abbildung 1: Ultraschallsensor CH201 [2]

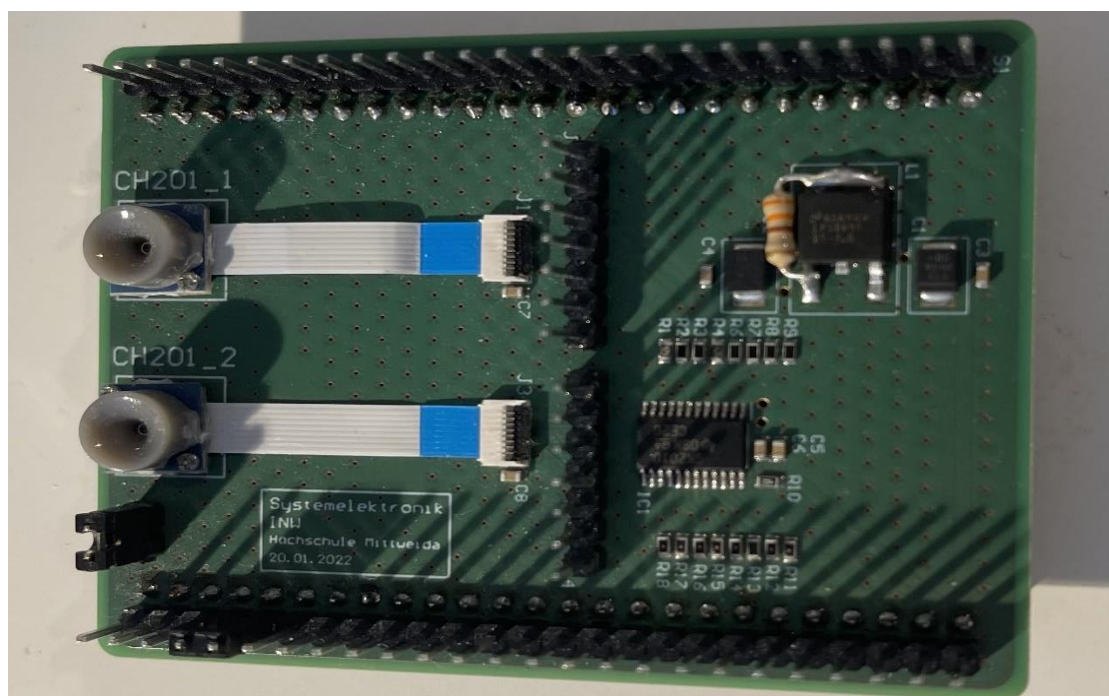


Abbildung 2: Zwei Ultraschallsensoren mit Mikrocontroller-Platine

Welcher direkte Zusammenhang besteht also zwischen der Entfernung und dem Ultraschallsensor? Die Übertragungsfunktion zwischen der Entfernung L und der Zeit t kann logisch als lineare Gleichung ausgedrückt werden:

$$L = \frac{V_{Schall} * t}{2}$$

Dabei ist L die Entfernung vom Sensor zum Objekt, V_{Schall} die Schallgeschwindigkeit und t die Zeit, die der Sensor vom Senden bis zum Empfangen des Signals benötigt.

Der obige Teil beschreibt, wie Ultraschallsensoren Entfernungen messen. Die Abstandsmessung ist der erste Schritt der Winkelmessung. In dieser Arbeit werden zwei Ultraschallsensoren verwendet, um den Abstand und den Winkel des Objekts zu messen. Hier werden die beiden Ultraschallsensoren zuerst verwendet, um die beiden Abstände zu messen und dann wird die trigonometrische Formel verwendet, um den Winkel des Objekts zu berechnen.

4.2 Vorteile eines mikrocontrollerbasierten Ultraschallsensor

Die Messung von Winkeln und Entfernungen von Objekten ist in Automobilen, Flugzeugen und Schiffen wichtig, zum Beispiel beim Rückfahrradar und beim Navigationsradar für Flugzeuge. Früher wurden Winkel und Abstände manuell gemessen, was zu großen Messfehlern führte. Mit der Entwicklung der Technologie rückt die Ultraschallortung allmählich in den Blickpunkt der Öffentlichkeit. Die Ultraschallmessung kann die manuelle Messung zur schnelleren und genaueren Positionierung von Objekten ersetzen. Bisherige Ultraschallgeräte waren zu groß und unpraktisch in der Anwendung. Um den Bedürfnissen einiger Wissenschaftsbegeisterter gerecht zu werden,

wurde ein mikrocontrollerbasiertes Ultraschallgerät entwickelt. Die Vorteile dieses Ultraschallgeräts mit Mikrocontroller sind seine geringe Größe, seine Tragbarkeit und seine einfache Handhabung. Es wandelt die gemessene Position des Objekts direkt in ein digitales Signal um, das in einen Mikrocontroller eingespeist wird, der die Messdaten berechnet, speichert und anzeigt. Der Mikrocontroller verfügt auch über eine PC-Schnittstelle für einen Computer, über die die Messdaten direkt an den Computer zur Verarbeitung und Berechnung übertragen werden können.

5. Konzept des Systementwurfs

5.1 Gesamtkonzept

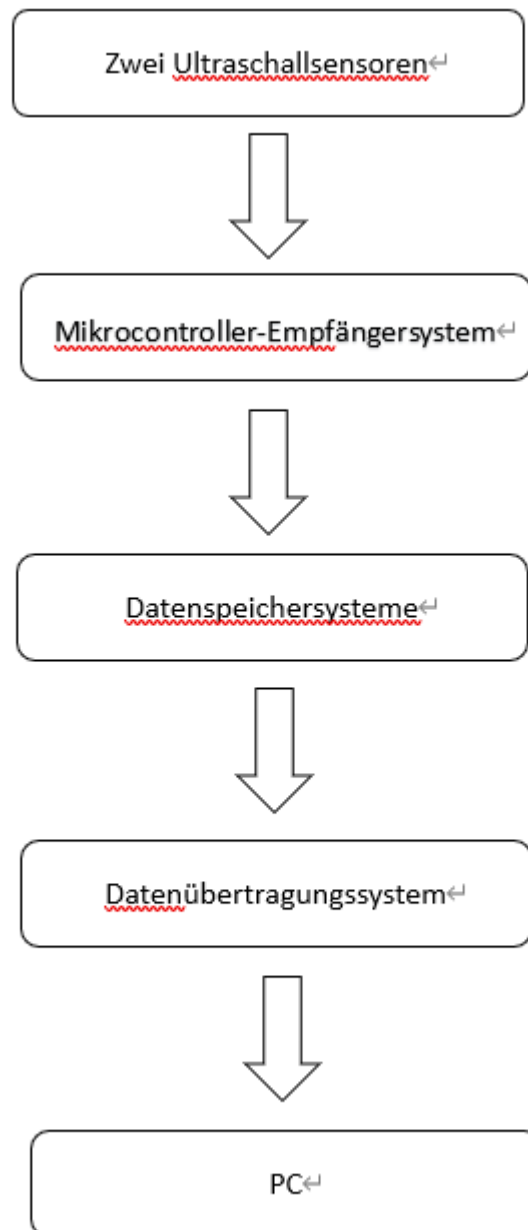


Abbildung 3: Flussdiagramm des Gesamtkonzepts

Die Abbildung 3 zeigt ein Flussdiagramm des Gesamtkonzepts. Der STM32-Mikrocontroller wandelt das von den beiden CH201-Ultraschallsensoren gemessene analoge

Signal in ein digitales Signal um und sendet die Daten über den seriellen I²C-Kommunikationsanschluss an den Arduino Mega 2560 und speichert sie. Die Daten werden dann in der Arduino IDE programmiert, um die Daten zu empfangen und zu verarbeiten, und schließlich werden die Entfernung und der Winkel auf dem seriellen Monitor vom PC angezeigt.

5.2 Berücksichtigung von Temperaturfaktoren

Genauere Untersuchungen und Messungen der Schallgeschwindigkeit haben gezeigt, dass die genaue Ausbreitungsgeschwindigkeit von mehreren Faktoren abhängt, wie z. B. der Temperatur der Luft und der Luftfeuchtigkeit, die im Vergleich zur Temperatur nur eine sehr geringe Auswirkung auf die Schallgeschwindigkeit hat, so dass hier nur die Wirkung der Temperatur betrachtet wird. Die Schallgeschwindigkeit hat einen deutlichen Einfluss auf die Messergebnisse und muss daher bei unterschiedlichen Temperaturen gemessen werden. Das folgende Diagramm zeigt den Zusammenhang zwischen Schallgeschwindigkeit und Temperatur.

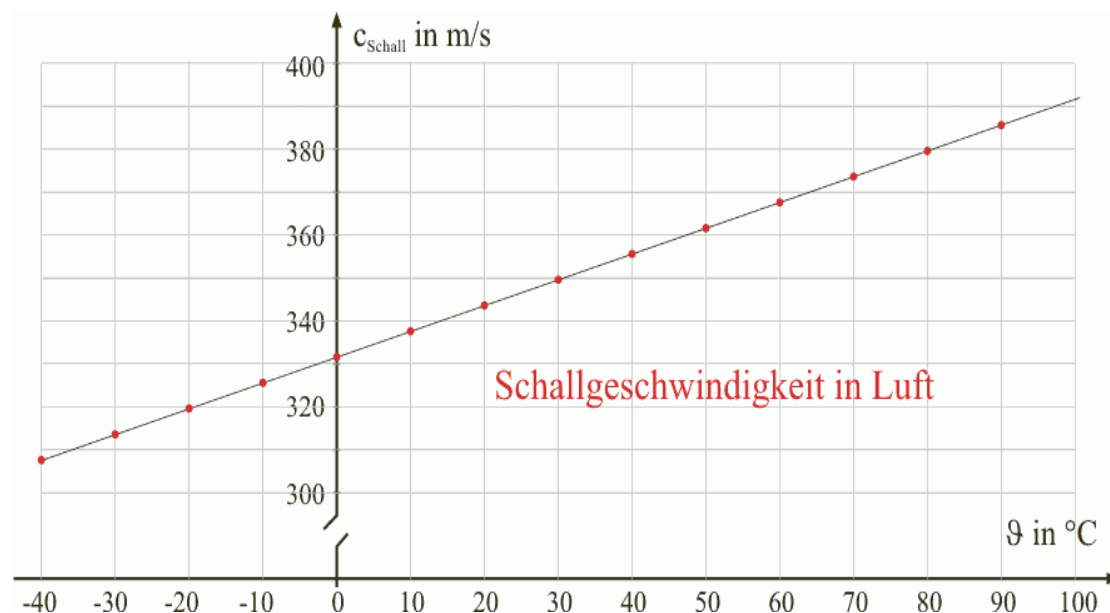


Abbildung 4: Schallgeschwindigkeit in Abhängigkeit von der Lufttemperatur [3]

Wie aus diesem Diagramm hervorgeht, ist die Schallgeschwindigkeit linear mit der

Temperatur verknüpft und kann daher durch eine einfache mathematische Formel ausgedrückt werden. Die Formel lautet wie folgt:

$$c_{\text{Schall}} = \left(331,6 + 0,6 \frac{\vartheta}{^{\circ}\text{C}} \right) \frac{\text{m}}{\text{s}}$$

Abbildung 5: Formel für Schallgeschwindigkeit und Temperatur [3]

Die Schallgeschwindigkeit ist in der Arduino-Programmierung definiert, so dass es notwendig ist, die Schallgeschwindigkeit in Echtzeit zu berechnen, indem die vom Thermometer gemessene Temperatur in die Formel eingesetzt und in der Programmierung geändert wird.

5.3 Interpretation von Winkelmessungen

Dieses in Abbildung 6 dargestellte Projekt erfordert den Einsatz von zwei Ultraschallsensoren zur Messung des Winkels eines Objekts.

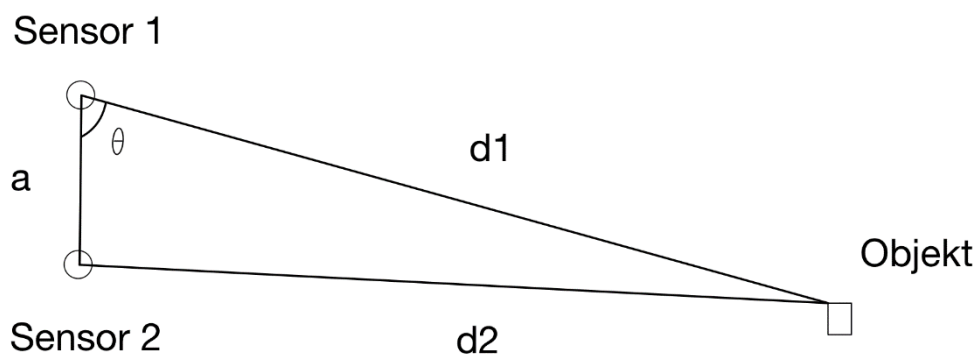


Abbildung 6: Schematische Winkelmessung

Der Abstand a zwischen den beiden Sensoren ist fest und bekannt. Mit zwei Sensoren wird die Entfernung des Objekts als d_1 bzw. d_2 gemessen, und anschließend wird der Winkel des Objekts mit Hilfe der trigonometrischen Formel berechnet. Die mathematische Gleichung lautet wie folgt:

$$\cos \theta = \frac{a^2 + d_1^2 - d_2^2}{2ad_1}$$
$$\theta = \cos^{-1}(\cos \theta)$$

5.4 Hardware

In diesem Abschnitt wird die für diese Arbeit verwendete Hardware beschrieben.

5.4.1 Arduino Mega 2560

Der Arduino Mega 2560 ist ein Mikrocontroller-Board auf Basis des ATmega2560. Es verfügt über 54 digitale Eingangs-/Ausgangs-Pins (von denen 15 als PWM-Ausgänge verwendet werden können), 16 analoge Eingänge, 4 UARTs (serielle Hardware-Ports), einen 16-MHz-Quarzoszillator, einen USB-Anschluss, eine Stromversorgungsbuchse, einen ICSP-Header und eine Reset-Taste. Es enthält alles, was zur Unterstützung des Mikrocontrollers benötigt wird; schließen Sie es einfach mit einem USB-Kabel an einen Computer an oder versorgen Sie es mit einem AC/DC-Adapter oder einer Batterie, um loszulegen.[4]

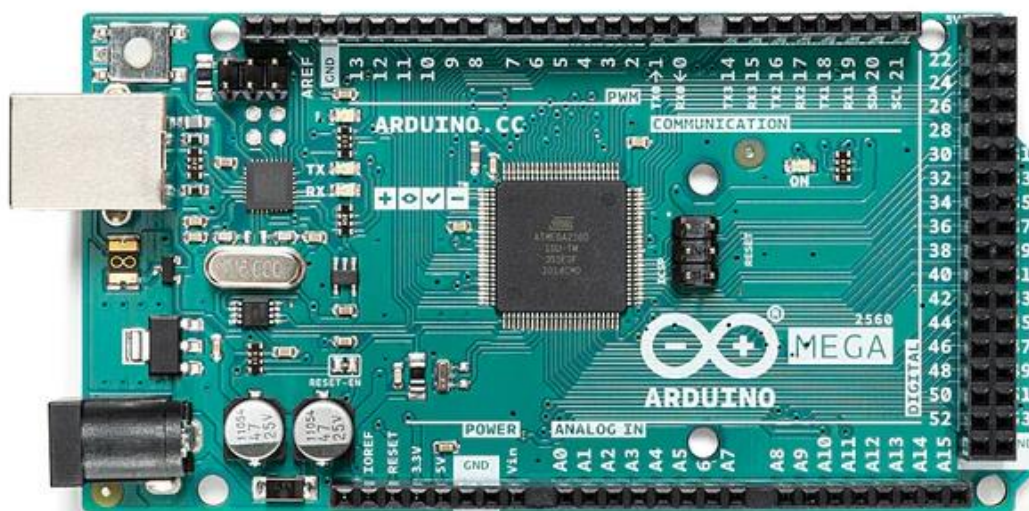


Abbildung 7: Arduino Mega 2560 [4]

5.4.2 STM32L053 Mikrocontroller

Die extrem stromsparenden STM32L053x6/8-Mikrocontroller verfügen über die Anschlussleistung des universellen seriellen Busses (USB 2.0 ohne Kristall) mit dem leistungsstarken Arm® Cortex-M0®+ 32-Bit-RISC-Kern, der mit einer Frequenz von 32 MHz arbeitet, einer Speicherschutzeinheit (MPU), eingebetteten Hochgeschwindigkeitsspeichern (bis zu 64 KByte Flash-Programmspeicher, 2 KByte Daten-EEPROM und 8 KByte RAM) sowie einer umfangreichen Auswahl an erweiterten I/Os und Peripheriegeräten. Die STM32L053x6/8-Bausteine bieten eine hohe Energieeffizienz für einen breiten Leistungsbereich. Dies wird mit einer großen Auswahl an internen und externen Taktquellen, einer internen Spannungsanpassung und mehr Low-Power-Modi erreicht.[5]

Die STM32L053x6/8-Bausteine bieten mehrere analoge Funktionen, einen 12-Bit-ADC mit Hardware-Oversampling, einen DAC, zwei Ultra-Low-Power-Komparatoren, mehrere Timer, einen Low-Power-Timer (LPTIM), drei allgemeine 16-Bit-Timer und einen Basistimer, einen RTC und einen SysTick, die als Zeitbasis verwendet werden können. Sie verfügen auch über zwei Watchdogs, einen Watchdog mit unabhängiger Uhr- und Fensterfunktion und einen Window Watchdog basierend auf der Busuhr. Darüber hinaus enthalten die STM32L053x6/8-Geräte Standard- und erweiterte Kommunikationsschnittstellen: bis zu zwei I²C, zwei SPIs, einen I2S, zwei USARTs, einen Low-Power-UART (LPUART) und einen kristalllosen USB-Anschluss. Die Geräte bieten bis zu 24 kapazitive Sensorkanäle, um jeder Anwendung einfach Touch-Sensing-Funktionalität hinzuzufügen.[5]

Der STM32L053x6/8 verfügt außerdem über eine Echtzeituhr und eine Reihe von Backup-Registern, die im Standby-Modus mit Strom versorgt bleiben.[5]

Schließlich verfügt ihr integrierter LCD-Controller über einen integrierten LCD-Spannungsgenerator, mit dem bis zu 8 gemultiplexte LCDs mit Kontrast unabhängig von der Versorgungsspannung angetrieben werden können.[5]

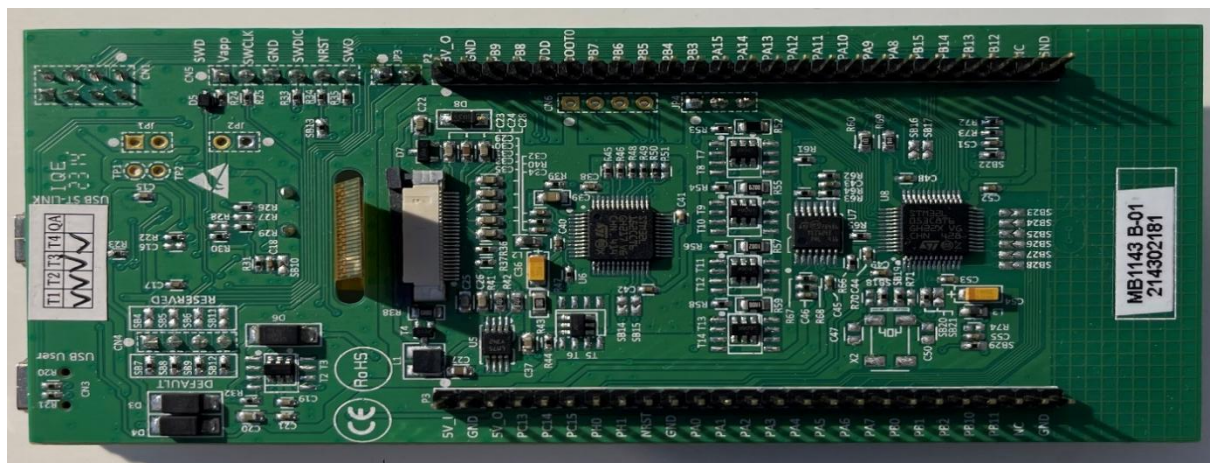


Abbildung 8: Vorderseite von STM32L053 Mikrocontroller

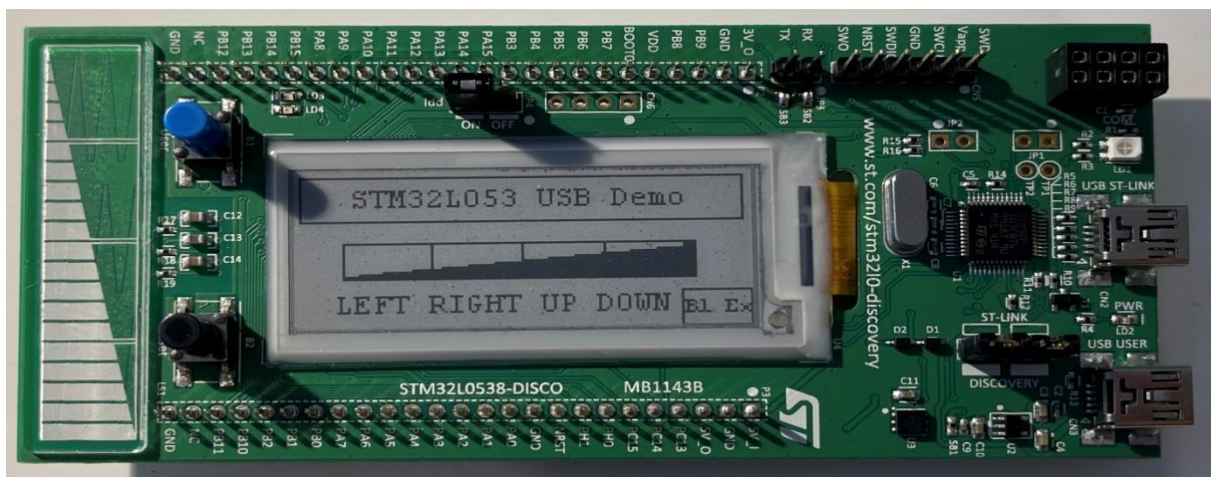


Abbildung 9: Rückseite von STM32L053 Mikrocontroller

5.4.3 Ultraschallsensor CH201

Der CH201 ist der in diesem Projekt verwendete Ultraschallsensor. Es handelt sich um einen Ultraschallsensor, der von dem Unternehmen TDK entwickelt wurde. Es hat einen Messbereich von 20 cm bis 5 m und eine Betriebsspannung von 1,8 Volt. Der maximale Winkel, den er messen kann, beträgt 40°, und er kann auch die Amplitude eines Objekts messen.

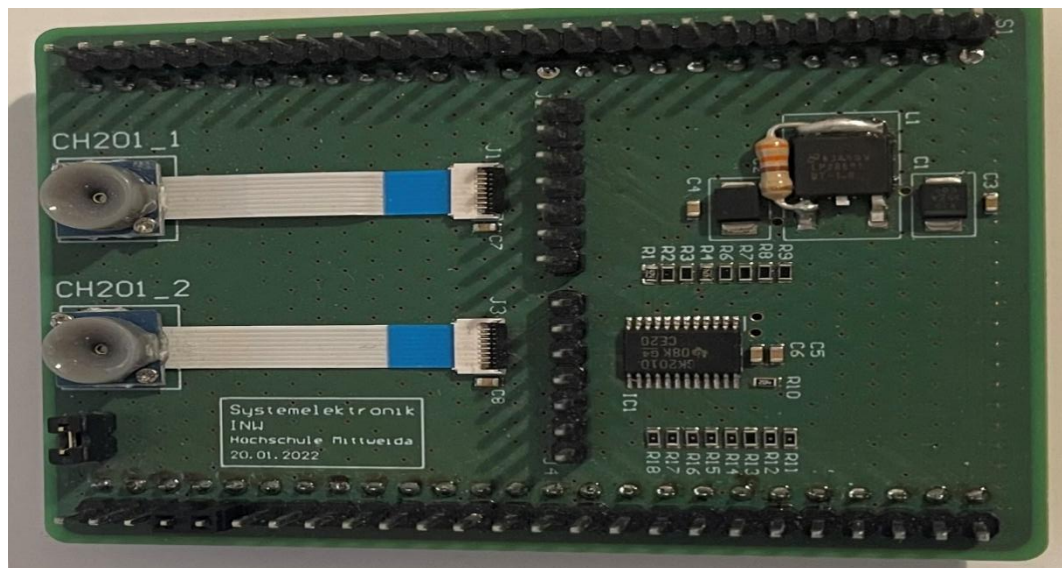


Abbildung 10: Ultraschallsensor CH201

Das CH201-Sensor kann allein oder in Kombination mit einem oder mehreren anderen Messwertaufnehmern verwendet werden. Ein einzelner Sensor arbeitet immer im "Puls-Echo"-Modus, bei dem er sowohl Ultraschallimpulse sendet als auch empfängt. Mehrere Sensoren können im "Sende- und Empfangsmodus" arbeiten, bei dem ein Sensor einen Ultraschallimpuls sendet und ein oder mehrere Sensoren diesen empfangen.[6] In diesem Projekt werden zwei CH201-Ultraschallsensoren verwendet, die im Impuls-Echo-Modus arbeiten. Wenn der Sensor in Betrieb genommen wird, senden und empfangen die Sensoren Ultraschallwellen, um die Messung durchzuführen. Sie hören das Echo ihres eigenen Ultraschallsignals ab, berechnen die Zeit von der Übertragung bis zum Empfang des Echos und informieren dann das Host-System, dass die Messung abgeschlossen ist. Dies wird als "Puls-Echo"-Verfahren bezeichnet.

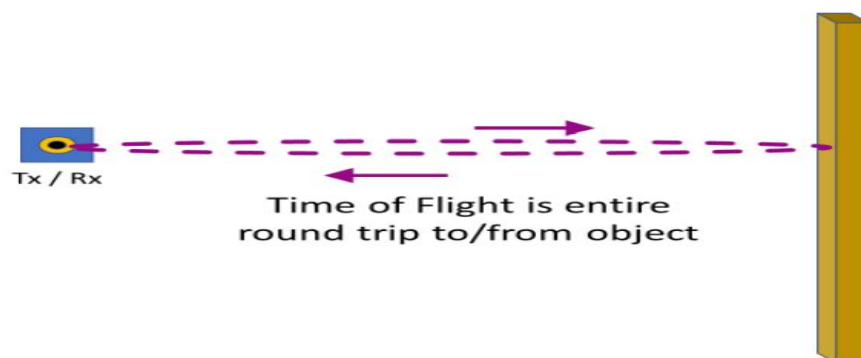


Abbildung 11: Single-Sensor Pulse-Echo [6]

Neben der Abstandsmessung ist der CH201-Sensor einzigartig in seiner Amplitudenabtastfunktion. Wie verarbeitet er die empfangenen Wellen, um einen Amplitudenscan zu erstellen? Der CH201-Wandler besteht aus einem piezoelektrischen mikromechanischen Ultraschallwandler und einem ASIC-Mikrocontroller, die beide im selben Gehäuse untergebracht sind. Der piezoelektrische mikromechanische Ultraschallwandler wandelt die empfangenen Schallwellen in ein analoges Spannungssignal um, das dann vom internen ADC des ASICs digitalisiert wird.[6] Ein Beispiel für dieses Signal ist in Abbildung 12 zu sehen, wo der direkte Ausgang des ADC als V_{RX} dargestellt ist.

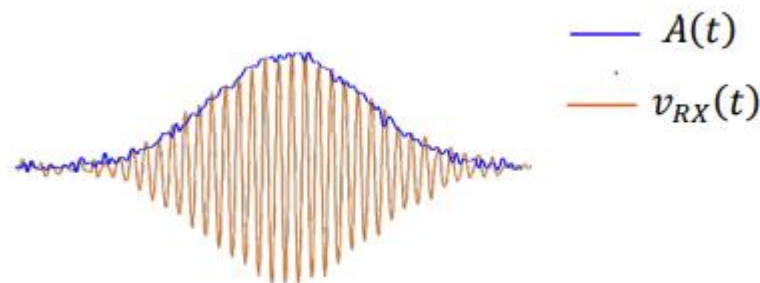


Abbildung 12: ADC-Ausgang V_{RX} und verarbeitetes A-Scan-Signal $A(t)$ [6]

Die Speicherung und Bearbeitung der Rohausgang des ADC ist beim CHx01 ineffizient und zeitaufwändig, daher wird die ADC-Ausgang direkt an den Demodulator und den Filter angeschlossen. Der Demodulator teilt das Ausgangssignal in eine In-Phase- (I) und eine Quadraturkomponente (Q) auf, die dann gefiltert und als ein Paar vorzeichenbehafteter 16-Bit-Ganzzahlen im internen Speicher des CHx01 gespeichert werden. Diese gefilterten I/Q-Basisbanddaten enthalten immer noch die Amplitude und Phase des ursprünglichen Signals und sind im Hinblick auf den Speicherplatz- und Verarbeitungszeitbedarf wesentlich effizienter als die Bearbeitung der Ultraschall-Rohdaten. Der Messwertaufnehmer verwendet die I/Q-Daten intern, um den Bereich zu messen, oder die I/Q-Daten können zur weiteren Verarbeitung an den Computer ausgelesen werden, um z. B. ein A-Scan zu erzeugen.[6]

Jedes Paar der erhaltenen I/Q-Daten ist eine Komponente der komplexen Zahl $(I+jQ)$,

die dann zur Berechnung der Amplitude des empfangenen Ultraschallsignals verwendet werden kann. Die Amplitude wird wie folgt berechnet:

$$\text{Amplitude: } A = \sqrt{I^2 + Q^2}$$

Über die I²C-Schnittstelle können die I/Q-Daten von einem externen Computer gelesen und die Amplitude jedes I/Q-Datensatzes verglichen werden, um den maximalen Amplitudenwert zu ermitteln, der der Amplitude des vom Sensor erfassten Objekts entspricht. Die Formel lautet wie folgt:

$$\text{Entfernung: } d = \frac{V_{\text{Schall}} * n}{2f_s}$$
$$\text{Abtastfrequenz : } f_s = \frac{1}{f_o}$$

Das V_{Schall} in der Formel steht für die Schallgeschwindigkeit, die sich mit der Umgebungstemperatur ändert. n steht für die n -te Probe. f_s ist die Abtastfrequenz, die einem Achtel der Betriebsfrequenz f_o entspricht, wobei die Betriebsfrequenz direkt aus den vom Computer angezeigten Messdaten berechnet werden kann.

Die empfangenen I/Q-Daten können mit der Hilfssoftware Hterm wie in Abbildung 13 dargestellt angezeigt werden. Wenn das System "0x33, 0x00, 0x56, 0x99, 0xFF, 0x00" erkennt, beginnt es, Daten zu empfangen, zuerst die Anzahl der Abtastungen, die zwei Bytes einnimmt, dann die Betriebsfrequenz f_o , die vier Bytes einnimmt, gefolgt von der Betriebsfrequenz, Q und I, die jeweils zwei Bytes einnehmen. Wenn das System eine Reihe von Daten erhalten hat, tritt es in eine Schleife ein.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
3	v	V	a	a	a	a	a	a	a	a	a	a	a	a	a	a	K	a	K	a	z	a	a
33	00	56	99	FF	00	01	20	00	01	33	D5	00	00	00	00	FF	4B	01	4B	FB	7A	08	Aa
051	000	086	153	255	000	001	032	000	001	051	213	000	000	000	000	255	075	001	075	251	122	008	164
a	a	a	a	a	a	a	a	K	a	a	a	l	a	a	a	6	a	a	a	a	a	a	a
00	1E	FF	CE	00	1A	FF	4B	00	B1	FD	6C	02	92	FA	36	04	0C	F8	93	FC	03	F9	89
000	030	255	206	000	026	255	075	000	177	253	108	002	146	250	054	004	012	248	147	252	003	249	133
a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a
F9	C7	04	3B	F7	AE	EF	93	F3	DC	EE	0A	F2	CE	F0	E4	F4	FF	03	02	F8	C7	05	30
249	199	004	059	247	174	239	147	243	220	238	010	242	206	240	228	244	255	003	002	248	199	005	048
a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a
00	2B	FF	B0	00	9F	01	DE	00	AE	FF	6F	00	BF	FF	AF	FF	66	FF	8C	FE	D6	FF	50
000	043	255	176	000	159	001	222	000	174	255	111	000	191	255	175	255	102	255	140	254	214	255	080
a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a
01	19	00	4C	01	70	00	81	01	60	00	4A	02	06	FE	50	01	64	FF	4C	00	D6	FF	19
001	025	000	076	001	112	000	129	001	096	000	074	002	006	254	080	001	100	255	076	000	214	255	021
a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a
FF	9C	00	C1	FF	86	00	9E	FF	DA	00	60	FF	A4	00	21	00	38	FF	E2	00	57	FF	B4
255	156	000	193	255	134	000	158	255	218	000	096	255	164	000	033	000	056	255	226	000	087	255	180
a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a
FF	CF	00	36	FF	D2	00	03	FF	FC	FF	F1	00	F1	FF	F1	00	4B	FF	87	FF	F2	00	03

Abbildung 13: In Hterm angezeigte I/Q-Daten

In diesem Projekt werden zwei Sensoren benötigt, und wenn zwei Sensoren zusammen verwendet werden, werden die empfangenen Daten um eins erhöht. Der Frame-Header lautet "0x33, 0x00, 0x56, 0x99, 0xFF, 0x01", wie in Abbildung 14 gezeigt.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	
3	v	V	a	a	a	a	*	v	a	>	\	v	v	v	v	a	-	a	a	a	f	a	X	a	a	a	a	a	,	a	a	v	l	a	a	a	a		
33	00	56	99	FF	01	01	2A	00	01	3E	5C	00	00	00	00	FF	2D	01	6F	BF	2E	09	66	F6	58	15	9C	F7	DF	17	2C	FD	C7	0C	6C	00	92	03	
a	a	a	a	a	a	a	a	a	a	*	a	a	a	x	a	a	I	a	n	a	a	a	a	a	a	a	/	a	a	a	5	a	a	v	a	a	D		
D6	F9	C9	04	A1	F9	E0	03	DB	FB	2A	03	1E	FD	78	02	8C	FF	49	02	6E	FF	D8	02	86	FF	94	02	2F	FF	1C	01	35	FE	AF	00	1C	FE	44	
a	a	a	4	a	a	a	a	v	8	a	a	v	a	}	a	a	a	-	a	a	a	E	a	a	a	a	a	/	a	a	a	a	R	v	:	a	i	v	
FF	9F	FF	34	FF	FA	FF	8C	00	38	FF	C2	00	20	FF	7D	FF	D3	FF	2D	FF	B5	FF	45	FF	B7	FF	A0	FF	8D	FF	FA	FF	52	00	3B	FF	69	00	
1	v	a	a	a	v	{	a	a	v	a	a	a	a	Q	a	a	a	a	a	a	a	\	a	a	a	a	a	v	a	a	a	a	a	a	a	a	a	a	
31	00	FF	FF	DF	00	7B	FF	EB	00	1D	FF	C5	FF	E2	FF	51	FF	BA	FF	1E	FF	A1	FF	5C	FF	BC	FF	B3	00	1D	FF	D7	00	84	FF	C4	00	A0	
v	]	v	a	v	y	v	a	v	8	v	a	v	8	v	a	v	a	c	v	a	v	a	a	a	a	a	a	v	a	v	a	a	a	a	a	a	a	a	a
00	5D	00	1C	00	79	00	18	00	5F	00	13	00	38	00	22	00	19	00	63	00	07	00	A4	00	02	00	AB	00	08	00	81	00	17	00	42	00	22	00	
a	a	a	a	a	A	v	v	U	v	L	v	T	v	s	v	3	v	U	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a
1F	00	1D	00	1E	00	41	00	20	00	55	00	4C	00	54	00	73	00	33	00	55	FF	FF	00	18	FF	EA	FF	FD	FF	F6	00	04	00	04	00	0B	00	02	
a	a	a	a	a	a	a	a	a	a	a	a	a	a	3	v	a	b	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	
FF	CA	00	13	FF	F4	FF	F5	00	17	FF	F9	00	33	00	01	00	62	FF	F1	00	77	FF	E4	00	53	FF	EB	00	2C	00	06	00	14	00	24	FF	FB	00	

Abbildung 14: In Hterm angezeigte I/Q-Daten

5.4.4 Hardwarekonzept

Die Hardware der Wahl ist der Arduino Mega 2560, der zwei Dupont-Kabel verwendet, um RX und TX des STM32-Mikrocontrollers mit den RX1- und TX1-Kommunikationsports des Arduino Mega 2560 zu verbinden. und schließen Sie den Arduino Mega 2560 und STM32 separat an den Computer an. Der Computer kann mit Betriebsstrom versorgt werden.

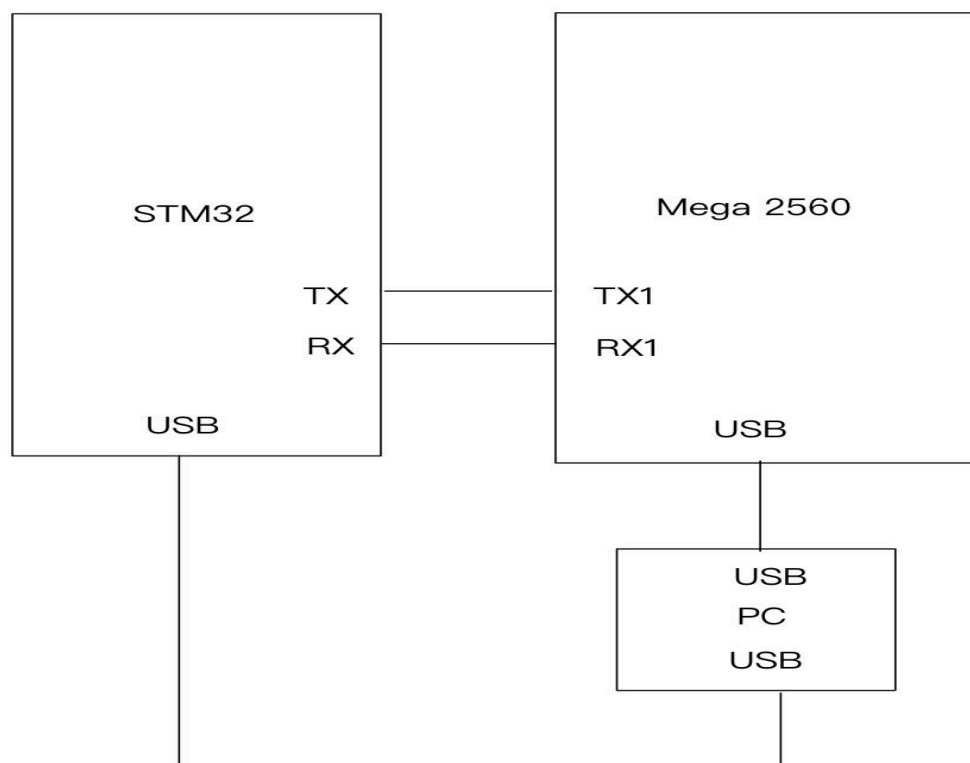


Abbildung 15: Blockdiagramm des Hardwarekonzepts

5.4.5 Softwarekonzept

In Abbildung 16 ist das Flussdiagramm des Systembetriebs dargestellt. Zu Beginn der Messung sendet der STM32 I/Q-Pakete an den Arduino. Der Arduino nimmt die Pakete an und führt ein Beurteilungsverfahren durch, um festzustellen, ob die Pakete vollständig angenommen werden; wenn nicht, werden sie weiterhin empfangen, wenn ja, wird zum nächsten Schritt übergegangen: Nummerierung der Abtastwerte und Berechnung

ihrer Amplitudenbetriebsfrequenz, dann künstliche Festlegung eines Amplitudenschwellenwertes. ermittelt und filtert die Samples heraus, deren Amplitude größer als der Amplitudenschwellenwert ist. Wenn die Amplitude der $(n-1)$ -Abtastung $<$ die Amplitude der n -ten Abtastung $>$ die Amplitude der $(n+1)$ -Abtastung erfüllt ist, filtert das System die n -te Abtastung heraus und berechnet den Abstand der beiden Sensoren zu ihr und anschließend ihren Winkel anhand der trigonometrischen Formel.

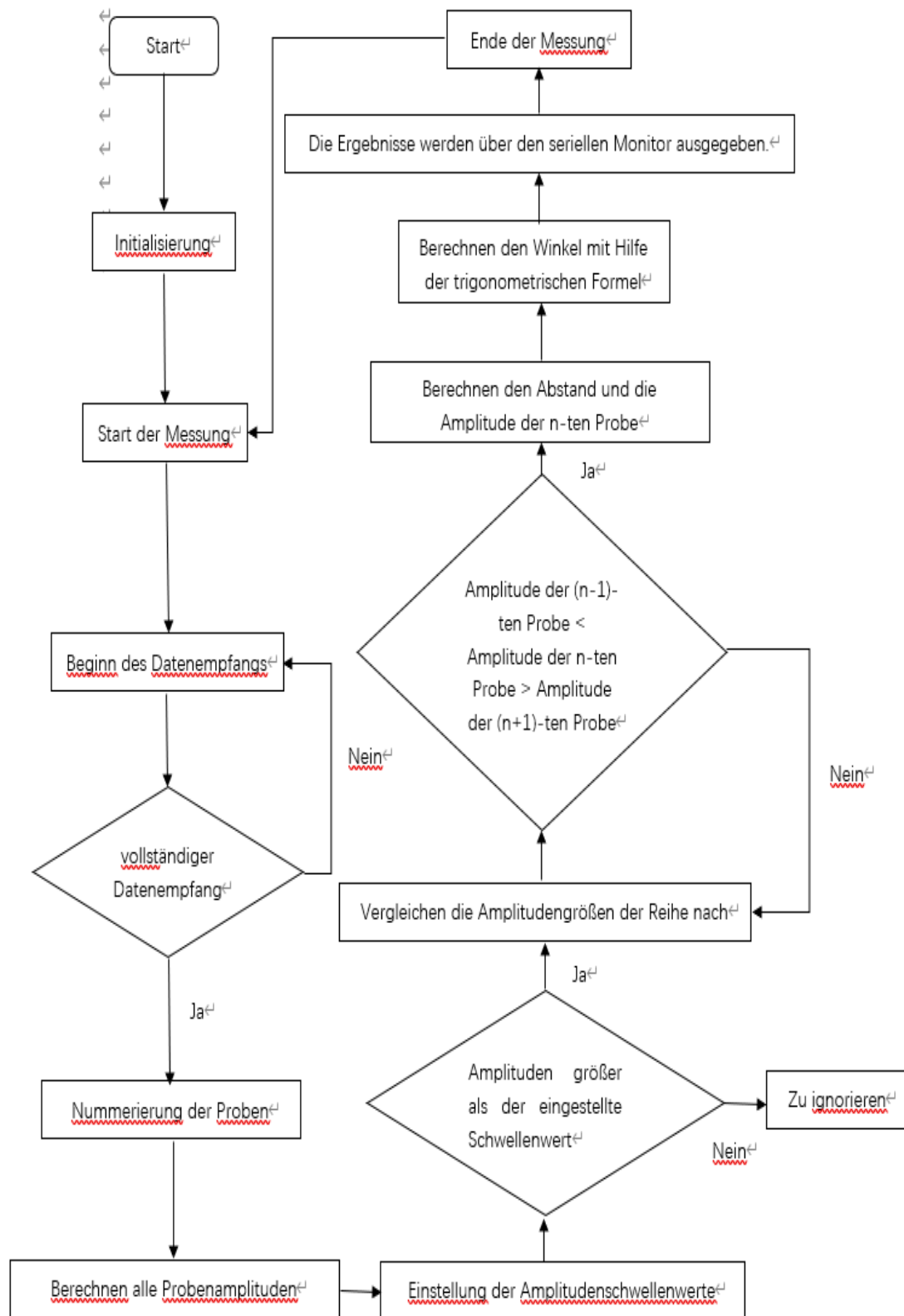


Abbildung 16: Flussdiagramm des Softwarekonzepts

6. Erläuterungen zu einigen Codes

Der wichtigste Schritt vor dem Schreiben von Code ist die Auswahl der richtigen Bibliothek. Es gibt drei Bibliotheken, die für diese Aufgabe zu verwenden sind: die Bibliothek, die die LEDs definiert, die zweite für die Ultraschallsensoren und die dritte für die serielle Schnittstelle.

```
#include "WorkLed.h"
#include "Ultrasonic.h"
#include "SerialCOM.h"
```

Abbildung 17: Auswahl der erforderlichen Bibliothek

Der erste und wichtigste Schritt beim Schreiben eines Programms ist das Einstellen der richtigen Baudrate, ein Schritt, der auch als Initialisierung der seriellen Schnittstelle bezeichnet wird. In diesem Auftrag ist die Baudrate auf eine Million eingestellt. Er definiert die Geschwindigkeit, mit der Daten bei der Kommunikation zwischen dem STM32-Mikrocontroller und dem Arduino gesendet und empfangen werden.

```
Serial.begin(1000000);
SerialCOM_Config();
Ultrasonic_Config();
```

Abbildung 18: Code für Baudrate der seriellen Schnittstelle

Der Code hier beschreibt die Berechnung des Abstands zwischen zwei Sensoren, die ein Objekt messen. Der hier gewählte Datentyp ist eine Gleitkommazahl mit doppelter Genauigkeit.

```
dblDistance = (double)1 / dblSamplingFrequency;
dblDistance = (double)SOUND_SPEED_IN_AIR * dblDistance;
dblDistance = dblDistance * stUltrasonicData.SampNo[ucPortNO][i];
dblDistance = dblDistance / (double)2;
stUltrasonicData.dblDistanceSensor[ucPortNO][i] = dblDistance * 1000;
```

Abbildung 19: Code zur Berechnung der Entfernung

Dieser Code wird verwendet, um die Amplitude des Objekts zu berechnen und benötigt die Funktion "sqrt". Er befindet sich in der Arduino-Bibliothek und steht für die Wurzel-Operation, die eine nicht-negative Quadratwurzel liefert.

```
// dblAmp = sqrt(dblQ * dblQ + dblI * dblI);

dblAmp = dblQ * dblQ;
dblTemp = dblI * dblI;
dblAmp = dblAmp + dblTemp;
dblAmp = sqrt(dblAmp);
```

Abbildung 20: Code für die Amplitudenberechnung

Dieser Code bestimmt den Wert von "n" in der Abstandsformel, indem er den maximalen Amplitudenwert ermittelt. Das Programm vergleicht die berechneten Amplitudenwerte der Reihe nach, und wenn die Bedingung „ $(n - 1)$ te Probenamplitude < nte Probenamplitude < $(n + 1)$ te Probenamplitude“

erfüllt ist, werden die Daten der n-ten Probe herausgefiltert und der von den beiden Sensoren gemessene Abstand und Winkel entsprechend berechnet.

```
for(i=1; i<iSampleCount - 1; i++)
{
    if ((stUltrasonicData.dblAmp[ucPortNO][i] > stUltrasonicData.dblAmp[ucPortNO][i - 1]) &&
        (stUltrasonicData.dblAmp[ucPortNO][i] > stUltrasonicData.dblAmp[ucPortNO][i + 1]))
    {
        if (stUltrasonicData.dblAmp[ucPortNO][i] > 3000)
        {
            stUltrasonicData.SampNo[ucPortNO][stUltrasonicData.SampCount] = i;
            stUltrasonicData.SampCount += 1;
        }
    }
}
```

Abbildung 21: Code für Amplitudenvergleich

In diesem Code wird erklärt, wie der Winkel zu berechnen ist, und die genaue Formel ist in den Kommentaren des Codes angegeben. Die trigonometrischen Funktionen werden hier benötigt und können direkt in der Arduino-Bibliothek verwendet werden. Es sollte auch erklärt werden, dass die hier berechneten Werte in Radiant angegeben sind und in Grad umgerechnet werden müssen.


```

// COSθ = (a*a + c*c - b*b)/(2*a*c)
// θ = arc(cosθ)
// a : Abstand zwischen zwei Sensoren in mm
// b : Abstand von Sensor 2 zum zu messenden Objekt, in mm
// c : Abstand von Sensor 1 zum zu messenden Objekt, in mm
// θ : Der zu messende Winkel
// Berechnung von a*a
stUltrasonicData.dblDegree[i] = DELTA_DISTANCE_OF_TWO_SENSORS * DELTA_DISTANCE_OF_TWO_SENSORS;
// Berechnung von c*c
dblTemp = stUltrasonicData.dblDistanceSensor[0][i] * stUltrasonicData.dblDistanceSensor[0][i];
// Berechnung von (a*a + c*c)
stUltrasonicData.dblDegree[i] += dblTemp;
// Berechnung von b*b
dblTemp = stUltrasonicData.dblDistanceSensor[1][i] * stUltrasonicData.dblDistanceSensor[1][i];
// Berechnung von (a*a + c*c - b*b)
stUltrasonicData.dblDegree[i] -= dblTemp;
// Berechnung von 2*a*c
dblTemp = 2 * DELTA_DISTANCE_OF_TWO_SENSORS;
dblTemp *= stUltrasonicData.dblDistanceSensor[0][i];
// COSθ = (a*a + c*c - b*b)/(2*a*c), wobei das Ergebnis "stUltrasonicData.dblDegree[i] = COSθ"
stUltrasonicData.dblDegree[i] /= dblTemp;
// Vorläufige Speicherung von COSθ
dblTemp = stUltrasonicData.dblDegree[i];

// Man erhält θ, hier den Winkel im Bogenmaß
stUltrasonicData.dblDegree[i] = acos(stUltrasonicData.dblDegree[i]);
// Umrechnung in Winkel in Grad
stUltrasonicData.dblDegree[i] *= 180;
stUltrasonicData.dblDegree[i] /= PI;
// Nach dem Berechnen des Winkels verwenden von " 90° - θ "
stUltrasonicData.dblDegree[i] = 90 - stUltrasonicData.dblDegree[i];

```

Abbildung 22: Code für die Winkelberechnung

7. Inbetriebsnahme

Es ist notwendig, vor der eigentlichen Messung zu überlegen, welche Variablen gemessen werden sollen. Die Aufgabe umfasst die Messung der Entfernung und des Winkels des Objekts sowie die Messung der maximalen Entfernung und des maximalen Winkels, in denen das Objekt gemessen werden kann. Bei den Messungen können experimentelle Fehler auftreten, die sich nicht vermeiden lassen: Die Temperatur beeinflusst beispielsweise die Schallgeschwindigkeit, was zu Fehlern bei der Messung der Entfernung zum Objekt führt, was wiederum Fehler bei der Messung des Winkels des Objekts zur Folge hat, da der Winkel des Objekts durch Einsetzen der Entfernung in die trigonometrische Formel berechnet wird.

7.1 Für die Messung verwendete Werkzeuge

Abbildung 23 zeigt ein Maßband mit einer maximalen Reichweite von 5 m, das hauptsächlich zur Messung der Entfernung von realen Objekten verwendet wird. Auf diese Weise wird der Abstand mit dem experimentell ermittelten Abstand verglichen, um seine Genauigkeit zu bestimmen.



Abbildung 23: Maßband [7]

Abbildung 24 zeigt ein Lineal, das Winkel messen kann, die direkt auf dem Display

abgelesen werden können. In der Praxis muss damit der tatsächliche Winkel des Objekts gemessen werden, um ihn mit dem experimentell gemessenen Winkel zu vergleichen.



Abbildung 24: Winkelmesser

8. Experimentelle Messungen

8.1 Durchführung von Messverfahren

Vor dem eigentlichen Experiment muss zunächst festgelegt werden, welche Daten gemessen werden sollen. Zum Beispiel die minimalen und maximalen Winkel, die der Sensor messen kann, die minimalen und maximalen Entfernungen und den Messfehler.

Zum Start der Messung wird eine Amplitudenschwelle vorgegeben, damit ungültige Daten herausgefiltert werden können und die Entfernung und der Winkel des zu messenden Objekts genauer auf dem seriellen Monitor ausgegeben werden können.

Wie in Abbildung 25 dargestellt, ist die tatsächliche Messung das aufgenommene Bild.

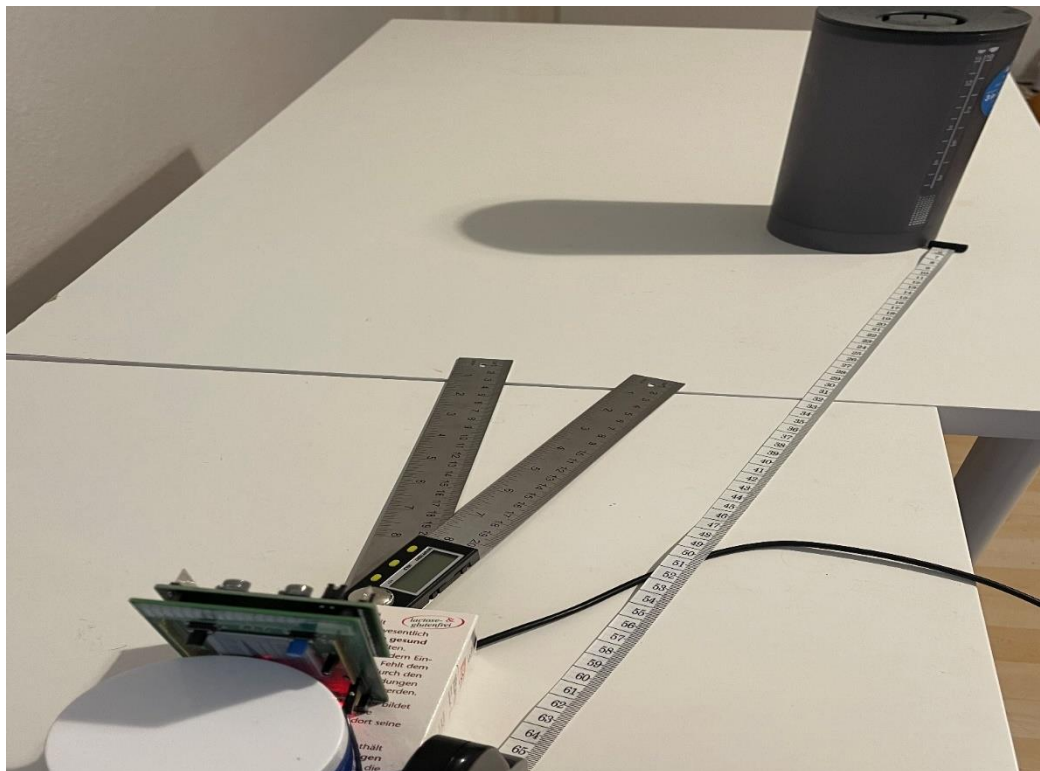


Abbildung 25: Tatsächliche Messung von Entfernungen und Winkeln von Objekten

Wie in Abbildung 26 dargestellt, können die Messdaten direkt vom seriellen Monitor abgelesen werden: Der Winkel des gemessenen Objekts beträgt $21,06^\circ$ und der Abstand 610.96 mm.

```

Port: 0, NO.:4, Range:66.05 mmPort: 1, NO.:4, Range:67.30 mmDegree=1.67
Port: 0, NO.:37, Range:610.96 mmPort: 1, NO.:36, Range:605.73 mmDegree=21.06
-----
Port: 0, NO.:4, Range:66.05 mmPort: 1, NO.:4, Range:67.30 mmDegree=1.67
Port: 0, NO.:38, Range:627.47 mmPort: 1, NO.:36, Range:605.73 mmDegree=nan
-----
Port: 0, NO.:4, Range:66.05 mmPort: 1, NO.:4, Range:67.30 mmDegree=1.67
Port: 0, NO.:37, Range:610.96 mmPort: 1, NO.:36, Range:605.73 mmDegree=21.06
-----
Port: 0, NO.:4, Range:66.05 mmPort: 1, NO.:4, Range:67.30 mmDegree=1.67
Port: 0, NO.:37, Range:610.96 mmPort: 1, NO.:36, Range:605.73 mmDegree=21.06
-----
Port: 0, NO.:4, Range:66.05 mmPort: 1, NO.:4, Range:67.30 mmDegree=1.67
Port: 0, NO.:38, Range:627.47 mmPort: 1, NO.:36, Range:605.73 mmDegree=nan
-----
Port: 0, NO.:4, Range:66.05 mmPort: 1, NO.:4, Range:67.30 mmDegree=1.67
Port: 0, NO.:37, Range:610.96 mmPort: 1, NO.:36, Range:605.73 mmDegree=21.06
-----
Port: 0, NO.:4, Range:66.05 mmPort: 1, NO.:4, Range:67.30 mmDegree=1.67
Port: 0, NO.:38, Range:627.47 mmPort: 1, NO.:36, Range:605.73 mmDegree=nan
-----
Port: 0, NO.:4, Range:66.05 mmPort: 1, NO.:4, Range:67.30 mmDegree=1.67
Port: 0, NO.:37, Range:610.96 mmPort: 1, NO.:36, Range:605.73 mmDegree=21.06
-----
Port: 0, NO.:4, Range:66.05 mmPort: 1, NO.:4, Range:67.30 mmDegree=1.67
Port: 0, NO.:38, Range:627.47 mmPort: 1, NO.:36, Range:605.73 mmDegree=nan
-----
Port: 0, NO.:4, Range:66.05 mmPort: 1, NO.:4, Range:67.30 mmDegree=1.67
Port: 0, NO.:37, Range:610.96 mmPort: 1, NO.:36, Range:605.73 mmDegree=21.06
-----
Port: 0, NO.:4, Range:66.05 mmPort: 1, NO.:4, Range:67.30 mmDegree=1.67
Port: 0, NO.:37, Range:610.96 mmPort: 1, NO.:36, Range:605.73 mmDegree=21.06
-----
Port: 0, NO.:4, Range:66.05 mmPort: 1, NO.:4, Range:67.30 mmDegree=1.67
Port: 0, NO.:38, Range:627.47 mmPort: 1, NO.:36, Range:605.73 mmDegree=nan
-----
Port: 0, NO.:4, Range:66.05 mmPort: 1, NO.:4, Range:67.30 mmDegree=1.67
Port: 0, NO.:37, Range:610.96 mmPort: 1, NO.:36, Range:605.73 mmDegree=21.06
-----

```

Abbildung 26: Daten vom seriellen Monitor

Der tatsächlich gemessene Winkel beträgt $19,6^\circ$ und die Entfernung 580,3 mm, wie auf der Anzeige des Maßbandes und des Maßbandes abgelesen werden kann. Daraus lässt sich schließen, dass der experimentell gemessene Winkel vom tatsächlichen Winkel um $1,46^\circ$ und der experimentell gemessene Abstand vom tatsächlichen Abstand um 30,66 mm abweicht.

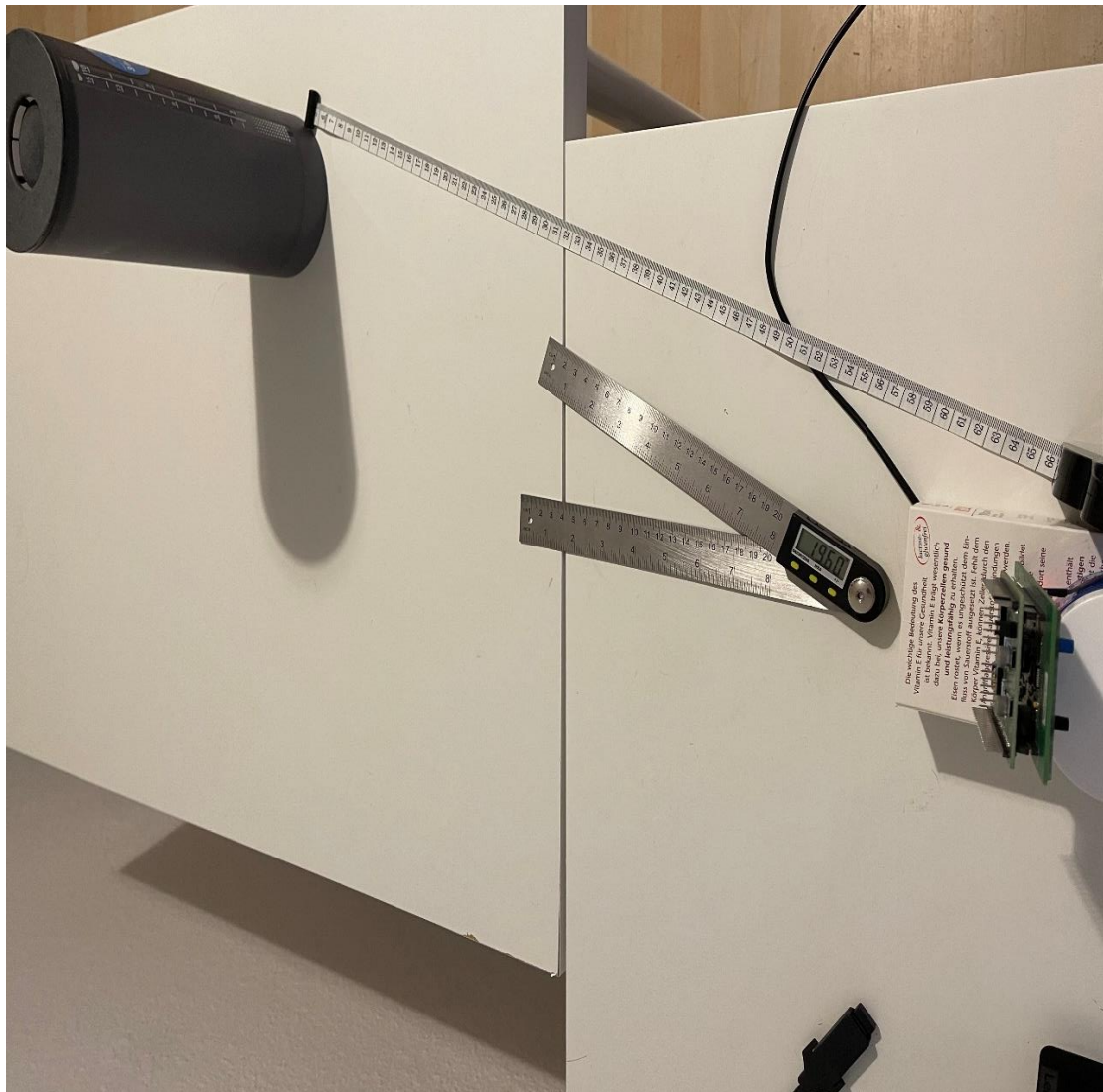


Abbildung 27: Ergebnisse der tatsächlichen Messungen

8.2 Analyse der Versuchsdaten

Die folgenden Daten wurden auf die gleiche Weise gemessen, wobei der Messbereich von 20 cm bis 100 cm mit einer Mindestreichweite von 20 cm kontrolliert wurde. Die Analyse der Daten zeigt, dass die durchschnittlichen absoluten Abweichungen für beide Sensoren 2,0 cm für den gemessenen Abstand und $4,4^\circ$ für den gemessenen Winkel betragen. Der Durchschnittswert wird hier nur für gültige Daten berechnet, und wenn der absolute Fehler zu groß ist, werden die Messdaten als ungültig betrachtet. Die in der Tabelle rot markierten Daten sind ungültig.

	Referenzabstand (cm)	Von Sensor 1 gemessene Entfernung (cm)	Absolute Abweichung (cm)	Von Sensor 2 gemessene Entfernung (cm)	Absolute Abweichung (cm)
1					
2	23.4	28.0	4.6	25.9	2.5
3	24.6	26.4	1.8	26.7	2.1
4	25.4	28.0	2.6	26.9	1.5
5	29.4	31.3	1.9	30.3	0.9
6	33.2	34.6	1.4	33.7	0.5
7	33.4	36.3	2.9	35.4	2
8	34.4	36.2	1.8	37.0	2.6
9	44.5	44.6	0.1	43.7	0.8
10	54.2	51.6	2.6	49.8	4.4
11	63.3	61.2	2.1	60.5	2.8
12	73.2	74.1	0.9	72.4	0.8
13	85.5	87.3	1.8	82.5	3
14		Mittelwert der absoluten Abweichung	2.0	Mittelwert der absoluten Abweichung	2.0

Tabelle 1: Erste Messreihe (Teil 1)

1	Tatsächlicher Winkel (°)	Winkeln der Messungen (°)	Absolute Abweichung (°)
2	43.4	34.1	9.3
3	-16.4	-14.7	1.7
4	34.1	33.4	0.7
5	27.4	31.5	4.1
6	35.2	29.1	6.1
7	16.6	17.9	1.3
8	-44.3	-21.9	22.4
9	22.5	22.2	0.3
10	9.7	64.6	54.9
11	32.8	21.2	11.6
12	20.4	61.1	40.7
13	17.2	nan	#VALUE!
		Mittelwert der absoluten Abweichung	4.4

Tabelle 2: Erste Messreihe (Teil 2)

Aus dem zweiten Teil von Tabelle 1 lässt sich schließen, dass Winkelmessungen besser sind, wenn das Objekt zwischen 20 cm und 50 cm entfernt ist und der Winkel zwischen 15° und 30° liegt. Befindet sich das Objekt in einem Winkel von mehr als 30° , ist die Messung nicht zufriedenstellend. Dies zeigt sich daran, dass manchmal beide Ultraschallwellen das Objekt nicht gleichzeitig erfassen können und daher Entfernung und Winkel nicht gemessen werden können. Wenn das Objekt zwischen 50 cm und 100 cm entfernt ist, kann der Winkel manchmal nicht mit beiden Ultraschallwellen gemessen werden. In diesem Fall wird das System entscheiden, dass die Daten ungültig sind, und der endgültige Winkel wird angezeigt "nan".

Wie aus den Daten in Tabelle 2 ersichtlich ist, ist es schwierig, Winkel in Entfernungen von mehr als 100 cm zu messen. Beide Ultraschallwellen können jedoch die Entfernung von Objekten innerhalb des Messbereichs messen. Allerdings erhöht sich der Messfehler im Vergleich zur Nahbereichsmessung, z. B. beträgt der absolute Messfehler für Sensor 1 3.0 cm und für Sensor 2 2.6 cm.

Referenzabstand (cm)	Von Sensor 1 gemessene Entfernung (cm)	Absolute Abweichung (cm)	Von Sensor 2 gemessene Entfernung (cm)	Absolute Abweichung (cm)
101.5	105.3	3.8	104.2	2.7
110.9	114.5	3.6	112.0	1.1
120.4	122.4	2.0	124.6	4.2
125.5	123.5	2.0	124.4	1.1
152.3	153.5	1.2	155.6	3.3
220.7	224.8	4.1	222.7	2
245.5	248.5	3.0	250.6	5.1
285.5	288.4	2.9	282.7	2.8
320.4	322.7	2.3	319.7	0.7
357.5	361.7	4.2	359.6	2.1
420.6	422.5	1.9	424.6	4
450.4	454.4	5.0	452.8	2.4
540.6	20.6	520.0	27.8	512.8
	Mittelwert der absoluten Abweichung	3.0	Mittelwert der absoluten Abweichung	2.6

Tabelle 3: Zweite Messreihe (Teil 1)

1	Tatsächlicher Winkel (°)	Winkeln der Messungen (°)	Absolute Abweichung (°)
2	26.6	34.8	8.2
3	9.2	nan	#VALUE!
4	-20.7	nan	#VALUE!
5	-21.5	-27.5	6.0
6	-20.2	nan	#VALUE!
7	35.5	nan	#VALUE!
8	-30.7	nan	#VALUE!
9	16.7	nan	#VALUE!
10	20.6	nan	#VALUE!
11	34.6	nan	#VALUE!
12	-30.0	nan	#VALUE!
13	15.6	nan	#VALUE!
14	25.8	nan	#VALUE!

Tabelle 4: Zweite Messreihe (Teil 2)

8.3 Analyse der experimentellen Ergebnisse

Bei der Durchführung von Messungen sind u. a. folgende Probleme aufgetreten, die hier zusammengefasst sind. Wie in Abbildung 28 dargestellt, werden immer die gleichen Messdaten angezeigt, unabhängig davon, ob sich ein Objekt vor dem Sensor befindet. Wenn jedoch der Schwellenwert der Amplitude ausreichend groß geändert wird, verschwindet dieser Datensatz. Dies hat jedoch keine Auswirkungen auf die normalen Entfernungs- und Winkelergebnisse. Es wird davon ausgegangen, dass das Problem im Inneren des Sensors liegt.

```

-----
Port: 0, NO.:4, Range:65.93 mmPort: 1, NO.:4, Range:67.36 mmDegree=4.55
-----
Port: 0, NO.:4, Range:65.93 mmPort: 1, NO.:4, Range:67.36 mmDegree=4.55
-----
Port: 0, NO.:4, Range:65.93 mmPort: 1, NO.:4, Range:67.36 mmDegree=4.55
-----
Port: 0, NO.:4, Range:65.93 mmPort: 1, NO.:4, Range:67.36 mmDegree=4.55
-----
Port: 0, NO.:4, Range:65.93 mmPort: 1, NO.:4, Range:67.36 mmDegree=4.55
-----
Port: 0, NO.:4, Range:65.93 mmPort: 1, NO.:4, Range:67.36 mmDegree=4.55
-----
Port: 0, NO.:4, Range:65.93 mmPort: 1, NO.:4, Range:67.36 mmDegree=4.55
-----
Port: 0, NO.:4, Range:65.93 mmPort: 1, NO.:4, Range:67.36 mmDegree=4.55
-----
Port: 0, NO.:4, Range:65.93 mmPort: 1, NO.:4, Range:67.36 mmDegree=4.55
-----
Port: 0, NO.:4, Range:65.93 mmPort: 1, NO.:4, Range:67.36 mmDegree=4.55
-----
Port: 0, NO.:4, Range:65.93 mmPort: 1, NO.:4, Range:67.36 mmDegree=4.55
-----
Port: 0, NO.:4, Range:65.93 mmPort: 1, NO.:4, Range:67.36 mmDegree=4.55
-----
Port: 0, NO.:4, Range:65.93 mmPort: 1, NO.:4, Range:67.36 mmDegree=4.55
-----

```

Abbildung 28: Immer wiederkehrende Daten

Es gibt auch das Problem, dass der Winkel des Objekts vom Sensor mit einem zu großen Fehler gemessen wird und manchmal wird der Winkel des Objekts gar nicht gemessen und das System zeigt "nan" an. Der Grund dafür ist, dass zwei Sensoren, die dasselbe Objekt messen, unterschiedliche Fehler produzieren, und wenn dieser Fehler zu groß ist, kann der Sensor den Winkel des Objekts nicht messen. Der Grund dafür ist, dass die Messung des Winkels eines Objekts die Verwendung der trigonometrischen Formel erfordert, die ohne ein Dreieck nicht angewendet werden

kann. Es gibt ein Theorem für Dreiecke: Die Länge einer Seite eines Dreiecks ist damit stets größer als die Differenz aus den Längen der beiden anderen Seiten. [8]

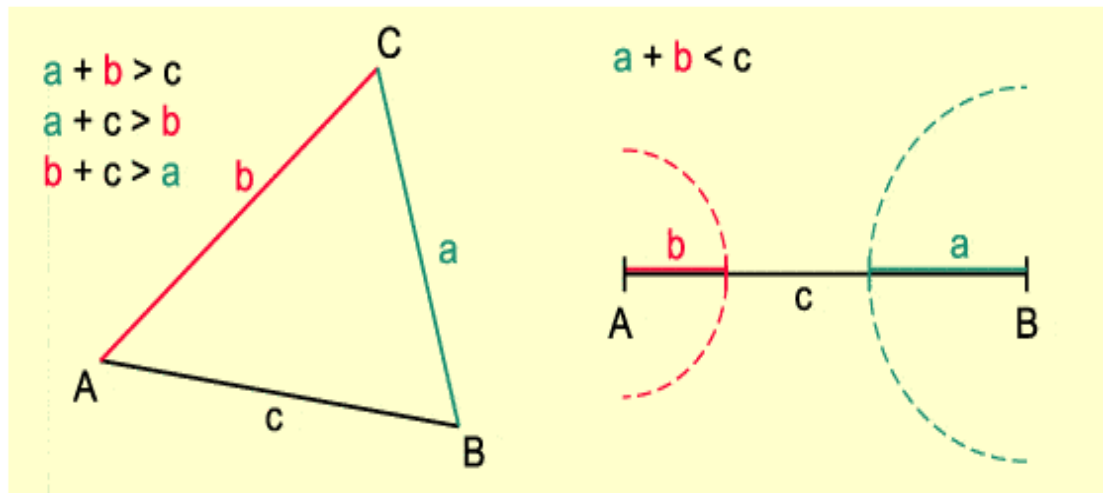


Abbildung 29: Dreieckstheorem [8]

Wie in Abbildung 30 dargestellt, ist der Abstand zwischen den beiden Sensoren auf 1,5 cm festgelegt. d_1 und d_2 werden von den beiden Sensoren gemessen. Da es einen Fehler zwischen den beiden Sensoren gibt, die das Objekt messen, wenn $|d_1 - d_2| > 1.5 \text{ cm}$ ist, erfüllt dies nicht den Satz des Dreiecks und daher kann der Winkel nicht mit der trigonometrischen Formel berechnet werden, das System bestimmt dann das Ergebnis als "nan".

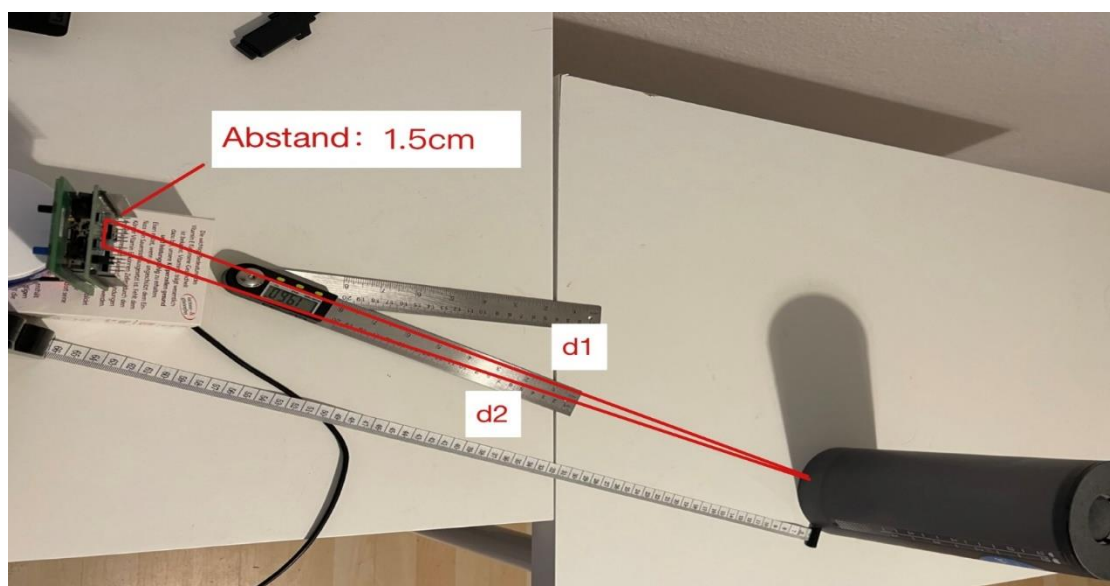


Abbildung 30: Schema der Messung

Die Lösung für dieses Problem besteht darin, den Abstand zwischen den beiden Ultraschallwellen zu vergrößern und die Position einer Ultraschallwelle als Referenzposition bei der Winkelmessung zu verwenden und dann den Winkel zu berechnen. Dies hat den Vorteil, dass der Winkelmessbereich vergrößert wird.

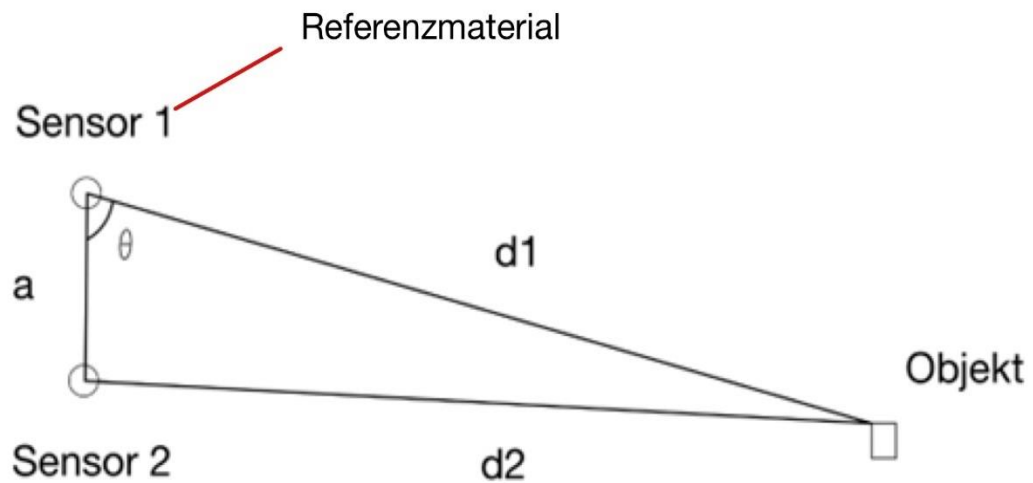


Abbildung 31: Theoretisches Diagramm

9. Zusammenfassung und Ausblick

Die gesamte Bachelorarbeit dauerte etwa drei Monate, und die allgemeine Planung wurde durch die Berücksichtigung einer großen Menge an relevantem Material zu Beginn des Entwurfs abgeschlossen. Die Hauptaufgabe in diesem Projekt bestand darin, die Arduino-Software zu programmieren und zu testen. Während des Projekts traten viele Schwierigkeiten auf, wie z. B. die Art und Weise, wie die Hardware-Schaltung die Messungen durchführt, wie man Daten sendet und wie man die serielle Kommunikation zwischen dem Arduino Mega 2560 und dem STM32 herstellt, und es wurden viele Simulationen durchgeführt, um diese Probleme zu lösen, sowie viel Material konsultiert.

Das Projekt wurde schließlich fertiggestellt, aber es gibt immer noch einige Mängel, die erklärt werden müssen, wie z.B. die Tatsache, dass der STM32 den dritten Modus verwendet, um Daten zu schnell zu senden, der Arduino zu lange braucht, um sie zu verarbeiten und die Ergebnisse nicht zufriedenstellend sind, was manchmal zu einem zu großen Fehler in der Entfernungsmessung führt und somit der Messwinkel nicht zufriedenstellend ist. Und manchmal ist das Arduino-IDE-System aufgrund der zu verarbeitenden Datenmenge instabil. Wenn jedoch Modus 1 angenommen wird, ist dieser Aspekt gelöst. Während des Messvorgangs empfängt der Sensor die vom Objekt reflektierten Ultraschallwellen aufgrund seines eigenen Aufstellwinkels nicht. Dies wird auch dazu führen, dass man nicht messen kann.

Aus dieser Arbeit kann man viel lernen und das Verständnis für die elektronische Informationstechnologie verbessern. Zugleich ist dies ein guter Test für die eigenen Fähigkeiten.

Literaturverzeichnis

Internet:

[1] Wikipedia: "Ultraschallsensor"

URL: <https://wiki.induux.de/Ultraschallsensoren> (abgerufen am 19.06.2022)

[2] invensense.tdk: "Ultraschallsensor CH201"

URL: <https://invensense.tdk.com/products/ch201/> (abgerufen am 19.06.2022)

[3] leifiphysik: "Beziehung zwischen Temperatur und Schallgeschwindigkeit"

URL: <https://www.leifiphysik.de/akustik/schallgeschwindigkeit/grundwissen/einfluss-faktoren-auf-die-schallgeschwindigkeit> (abgerufen am 20.06.2022)

[4] Arduino: "Arduino Mega 2560"

URL: <https://store.arduino.cc/products/arduino-mega-2560-rev3> (abgerufen am 21.06.2022)

[5] st.com: "STM32"

URL: <https://www.st.com/en/microcontrollers-microprocessors/stm32l053r8.html>
(abgerufen am 22.06.2022)

[6] invensense.tdk: "invensense.tdk"

URL: <https://invensense.tdk.com/products/ch201/> (abgerufen am 23.06.2022)

[7] baike.sogou.com: "Maßband"

URL: https://www.baike.com/wiki/%E5%8D%B7%E5%B0%BA?view_id=30lljqqiha000

(abgerufen am 08.07.2022)

[8] lernhelfer.de: "Dreieckstheorem"

URL: <https://www.lernhelfer.de/schuelerlexikon/mathematik/artikel/saetze-ueber-dreiecke>

(abgerufen am 08.07.2022)

Selbstständigkeitserklärung

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

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

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

Mittweida, den 25.Juli 2022

Wenzhe, Shen

Ort, Datum

Vorname Nachname