

Entfernungsmessung

Mechatronische Aufgabe – Team 4

Im Modul „Mechatronische Systeme in der Anwendung“
T3M10207.1

von

Chris Papke, Matrikelnummer: 5710671

Ruslan Adilgereev, Matrikelnummer: 7719665

05.03.2024

Bearbeitungszeitraum: 01.02. – 05.03.2024

Dozenten: Prof. Dr.-Ing. Schlegel & Prof. Dr.-Ing. Wühl

Inhaltsverzeichnis

Abbildungsverzeichnis.....	III
Tabellenverzeichnis.....	III
1 Einleitung	1
1.1 Aufgabenstellung.....	2
2 Physikalische Grundlage	2
2.1 Piezoelektrischer Effekt	2
2.2 Berechnung der Entfernung.....	3
3 Ausstattung	4
3.1 XC886 Entwicklungsboard.....	5
3.2 Ultraschallmodul SRF04	6
3.3 Analog Discovery Oszilloskop.....	7
4 Umsetzung	9
4.1 Hardware-Aufbau.....	9
4.2 Abstandsmessung 10 cm bis 80 cm	10
Initialisierung.....	11
Sensor Triggerung	12
Timer Start für Echo-Zeitmessung	13
Echo Interrupt Service Routine	13
Berechnung des Abstands.....	14
4.3 Erweiterung des Messbereichs	16
Timer Überlauf Handhabung.....	17
Berechnung der Distanz	17
LCD-Anzeige	18
5 Ergebnisse & Fazit.....	18
Problematik der Messung von geringen Distanzen mittels Ultraschallsensoren	18
Messergebnisse.....	19
Literaturverzeichnis	23
Anhang.....	24
Programmcode 10cm bis 80cm	24
Programmcode 3cm bis 3m.....	27

Abbildungsverzeichnis

Abbildung 1: Ultraschallsensoren für den Gebrauch in einem PKW zur Abstandsbestimmung & Schematische Darstellung der Abstandsmessung mittels Ultraschallprinzip [1].....	1
Abbildung 2: Funktionsweise von Piezoelektrizität [2]	3
Abbildung 3: XC886 Trainingsboard	5
Abbildung 4: Ultraschallmodul SRF04 PCB [4].....	6
Abbildung 5: SRF04 Timing Diagramm[4].....	7
Abbildung 6: Analog Discovery 2 Oszilloskop [5]	8
Abbildung 7: Hardwareaufbau und Verdrahtung des gesamten Messaufbaus	9
Abbildung 8: Flussdiagramm - Teilaufgabe 1	10
Abbildung 9: Struktogramm - Teilaufgabe 1	11
Abbildung 10: Darstellung des 15 μ s langen Triggerimpulses am Oszilloskop	13
Abbildung 11: Bestimmung des Laufzeit-Offsets.....	15
Abbildung 12: Flussdiagramm - Teilaufgabe 2	16
Abbildung 13: Struktogramm - Teilaufgabe 2	17
Abbildung 14: Sensorsignal, wenn kein Objekt erkannt wird	19
Abbildung 15: Messaufbau zur Bestimmung der Genauigkeit	20
Abbildung 16: Oszilloskopbild zur Abstandsmessung von 10cm	21
Abbildung 17: Oszilloskopbild zur Abstandsmessung von 300cm	22

Tabellenverzeichnis

Tabelle 1: Dichte, Schallgeschwindigkeit und Schallkennimpedanz einiger Stoffe beim Normdruck $p = 1013 \text{ hPa}$ [3]	4
Tabelle 2: Messergebnisse	20

1 Einleitung

Das Ultraschall-Entfernungsmessverfahren basiert auf der Verwendung von Ultraschallwellen, die Frequenzen oberhalb des menschlichen Hörbereichs (über 20 kHz) aufweisen. Diese Wellen werden entweder pneumatisch oder elektrisch (beispielsweise piezoelektrisch) erzeugt. Da Schallwellen sich nur in Materie ausbreiten können, erfolgt die Nutzung in der Automatisierungs- oder Automobiltechnik hauptsächlich in Luft. Ein wesentlicher Aspekt der Ultraschallmessung ist die Abhängigkeit der Schallausbreitungsgeschwindigkeit von Umgebungsbedingungen, insbesondere von der Lufttemperatur, da für Messungen die Ausbreitungsgeschwindigkeit des Schalls herangezogen wird. In der Automobiltechnik findet man Ultraschallsensoren meist angewandt als Parksensor.



Abbildung 1: Ultraschallsensoren für den Gebrauch in einem PKW zur Abstandsbestimmung & Schematische Darstellung der Abstandsmessung mittels Ultraschallprinzip [1]

Das Kernprinzip der Ultraschall-Entfernungsmessung besteht darin, dass ein Ultraschall-Sender Impulse aussendet, die von einem Objekt reflektiert werden. Diese reflektierten Wellen werden dann von einem Empfänger aufgefangen. Die Zeit, die zwischen dem Aussenden des Impulses und dem Empfang des Echos verstreicht, wird gemessen. Da die Schallgeschwindigkeit bekannt ist, lässt sich aus der Laufzeit des Ultraschallimpulses die Entfernung zum reflektierenden Objekt berechnen.

In der Praxis werden häufig Ultraschall-Abstandssensoren eingesetzt, bei denen Sender und Empfänger im selben Gehäuse untergebracht sind. Solche Sensoren

sind in der Lage, Objektabstände über die Laufzeit von Ultraschallimpulsen und die bekannte Schallgeschwindigkeit zu ermitteln.

1.1 Aufgabenstellung

Das Ziel dieses Projekts besteht darin, die Entfernung zwischen einem Sensor und einem Objekt mittels Ultraschalltechnologie zu messen, wobei der Messbereich zwischen 0,1 Meter und 3,0 Meter liegen soll. Um dieses Ziel zu erreichen, wurde die Gesamtaufgabe in mehrere Teilbereiche aufgeteilt.

1. Der Programmablaufplan soll in einem Struktogramm dargestellt werden
2. Realisieren eines Programms zur Entfernungsmessung zwischen 0,1 m und 0,8 m und Anzeige auf dem LC-Display
3. Erweiterung der Messung auf 3 m
4. Schwierigkeiten bei Messung von sehr kleinen Distanzen aufzeigen. Programmerweiterung auf Distanzen bis 0,03 m.

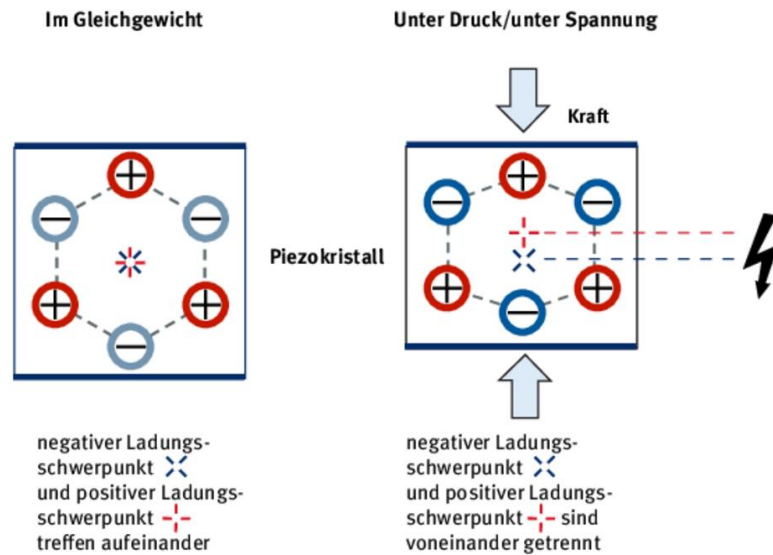
2 Physikalische Grundlage

In diesem Abschnitt sollen die physikalischen Grundlagen vorgestellt werden, die zur Bearbeitung der Aufgaben notwendig sind. Vor allem dreht es sich um die Entfernungsmessung mittels Ultraschallwellen und deren Erzeugung.

2.1 Piezoelektrischer Effekt

Der piezoelektrische Effekt beschreibt die Fähigkeit bestimmter Materialien, elektrische Ladung zu erzeugen, wenn sie mechanisch verformt (gedrückt oder gezogen) werden. Ebenso können diese Materialien sich verformen, wenn ein elektrisches Feld angelegt wird. Dieser Effekt tritt auf, weil die atomare Struktur der Materialien keine Symmetrie in Bezug auf ihre elektrische Ladungsverteilung besitzt, was bei Druck oder Zug zu einer Verschiebung der Ladungen führt. Diese einzigartige Eigenschaft macht piezoelektrische Materialien besonders nützlich für Sensoren, die

Druck, Schall oder Vibrationen in elektrische Signale umwandeln können, sowie für Aktuatoren, die elektrische Signale in mechanische Bewegung umsetzen.



CC by-nc-nd | www.weltderphysik.de

Abbildung 2: Funktionsweise von Piezoelektrizität [2]

Wenn das elektrische Feld hochfrequent mit 40 kHz Wechselspannung angelegt wird, gerät der Kristall in Schwingung. Wird er an eine geeignete Membran gekoppelt, kann diese Ultraschallwellen aussenden.

2.2 Berechnung der Entfernung

Das vom Sensor ausgesandte Signal wird mit einer konstanten Geschwindigkeit v durch die Luft übertragen, reflektiert nach einer gewissen Strecke s an einer Oberfläche und wird von einem Ultraschalldetektor nach einer Zeit t erfasst. Zur Berechnung des Abstandes zwischen Sensor und Gegenstand wird der Zusammenhang dieser drei physikalischen Größen genutzt.

$$v = \frac{s}{t}$$

Wenn man nun nach der Strecke umstellt, erhält man folgenden Zusammenhang.

$$s = v * t$$

Wie oben bereits beschrieben, ist die Streck s der zurückgelegte Weg des Schalls vom Sender zum Empfänger. Da sich in unserem Aufbau Sender und im Empfänger im selben Gehäuse befinden legt der Schall den doppelten Weg der Entfernung des Gegenstandes zurück. Das muss bei der Kalkulation der Strecke beachtet werden. Somit gilt für die Distanz d des Messobjektes zum Sensor folgender Zusammenhang.

$$d = \frac{s}{2} = \frac{1}{2} v * t$$

Als Geschwindigkeit wird die Schallgeschwindigkeit und Luft bei 20°C und einem Umgebungsdruck von 1013 hPa mit $v = 344 \text{ m/s}$ herangezogen.

	Dichte ρ in $\frac{\text{kg}}{\text{m}^3}$	Schallgeschwindigkeit c in $\frac{\text{m}}{\text{s}}$	Schallkennimpedanz Z_0 in $\frac{\text{kg}}{\text{m}^2 \text{s}}$
Luft –20 °C trocken	1,396	319	445
Luft 0 °C trocken	1,293	331	427
Luft 20 °C trocken	1,21	344	416
Luft 100 °C trocken	0,947	387	366
Wasserstoff 0 °C	0,090	1260	113
Wasserdampf 130 °C	0,54	450	243
Wasser 0 °C	1000	1400	$1,40 \cdot 10^6$
20 °C	998	1480	$1,48 \cdot 10^6$
Glycerin	1260	1950	$2,46 \cdot 10^6$
Eis	920	3200	$2,94 \cdot 10^6$
Holz	600	4500	$2,70 \cdot 10^6$
Glas	2500	5300	$13,0 \cdot 10^6$
Beton	2100	4000	$8,4 \cdot 10^6$
Stahl	7700	5050	$39 \cdot 10^6$

Tabelle 1: Dichte, Schallgeschwindigkeit und Schallkennimpedanz einiger Stoffe beim Normdruck $p = 1013 \text{ hPa}$ [3]

3 Ausstattung

Um die Aufgaben zu lösen, wurde jedem Teilnehmer des Teams ein Set zur Verfügung gestellt welches aus Hardware als auch Software besteht. Nachfolgend werden die Bestandteile kurz vorgestellt.

3.1 XC886 Entwicklungsboard

Im Kern des Boards befindet sich ein Mikrocontroller mit einem Mikroprozessor aus der 8051-Serie von Infineon. Die Programmierung des Boards erfolgt über einen USB-Anschluss, der ebenfalls zur Versorgung mit Betriebsspannung genutzt wird. Eine LED (LED1) dient als Indikator für die Betriebsspannung. Außerdem verfügt das Board über ein LC-Display, das vier Zeilen mit je 20 Zeichen anzeigen kann. Oben am Board sind die Ports 0, 1 und 3 mit je 8 Bits zu sehen und teilweise nach außen geführt. Diese können flexibel als Ein- oder Ausgänge konfiguriert werden. Einige dieser Bits sind mit LEDs verbunden, während die restlichen Bits von Port 0 und 1 für interne Programmierschnittstellen genutzt werden, z.B. für den Anschluss des Displays. Dies ermöglicht eine visuelle Darstellung der logischen Zustände der Portanschlüsse. Zudem sind 2 mm-Buchsen vorhanden, über die Trainingsmodule mittels 2 mm-Steckkabel angeschlossen werden können. Das Board enthält auch zwei analoge Potentiometer (AN4 und AN5), mit denen ein 0... 5 V-Signal erzeugt werden kann, dass zur Steuerung anderer Komponenten dient. Diese Potentiometer sind an einen 10-Bit-Analog-Digital-Wandler angeschlossen, was die Ausgabe eines Digitalwerts von 0 bis 1023 bei der Ablesung dieser Eingänge ermöglicht. Zusätzliche Anschlussbuchsen dienen der Versorgung weiterer Module mit 5 V sowie der Vereinheitlichung der Masseverbindung (GND). Für die Programmierung des Mikrocontrollers wird die Entwicklungssoftware Keil verwendet.

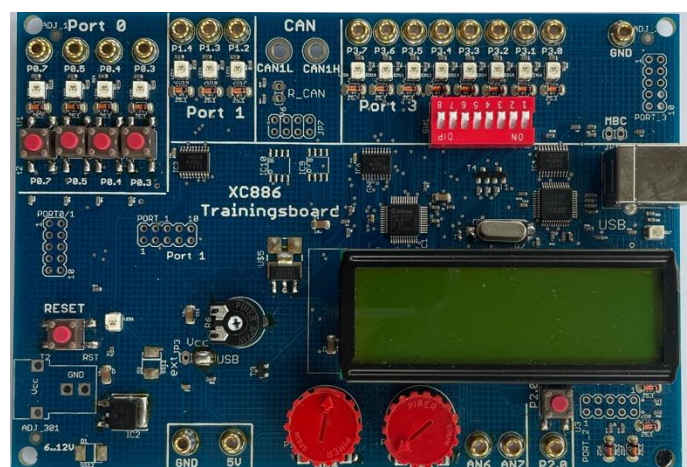


Abbildung 3: XC886 Trainingsboard

3.2 Ultraschallmodul SRF04

Wie bereits in Kapitel 2 erwähnt, basiert die Entfernungsmessung auf dem Piezoelektrischen Effekts mit einem Sender und einem Empfänger auf einem Board. Laut Datenblatt wird eine Spannungsversorgung mit 5 V und 30 mA benötigt. Damit diese bereitgestellt werden kann, muss das Board extern über ein beigelegtes Netzteil versorgt werden. Der Ultraschall befindet sich mit 40 kHz weit über dem hörbaren Bereich von 20 kHz. Es hat einen Messbereich, laut Datenblatt, von 3 cm bis zu 3 m und kann einen Besenstiel mit 3 cm Durchmesser in einem Abstand von 2 m sicher erkennen.

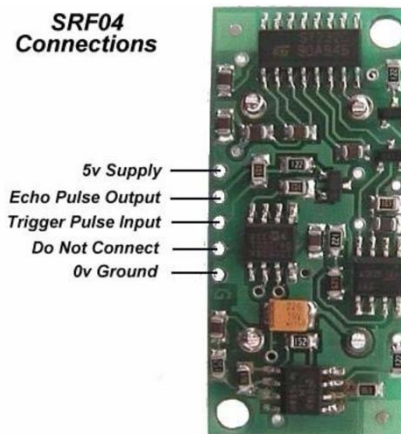


Abbildung 4: Ultraschallmodul SRF04 PCB [4]

Das Board verfügt über 4 Anschlüsse. Einen für die 5 V Versorgungsspannung, einen, um die Verbindung zum Ground des Netzteils herzustellen. Die anderen beiden dienen der Bedienung des Sensors. Der Trigger Anschluss ist der Input des Boards. Der Echo-Pulse Pin dient als Output.

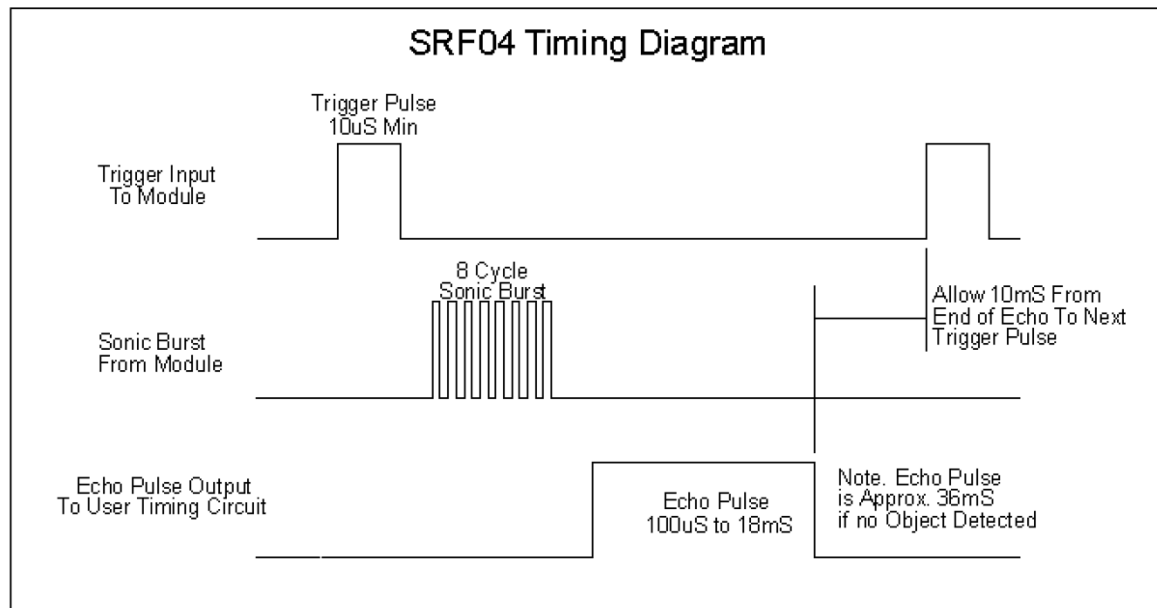


Abbildung 5: SRF04 Timing Diagramm[4]

In der Darstellung (Abbildung 5: SRF04 Timing Diagramm) wird der Ablauf der Messung dargestellt. Zuerst muss der Trigger-Pin mit einem mindestens 10 µs langen 5 V Puls vom Microcontroller versorgt werden, um eine Messung zu starten. Nachdem das Ultraschallmodul den Trigger erkannt hat, sendet es acht Zyklen eines Ultraschallsignals aus. Nach dem Aussenden wird der Echo-Pin so lange auf den High-Pegel gesetzt bis der Detektor den reflektierten Ultraschallburst detektiert. Die Länge des 5 V Signals ist proportional zum doppelten Abstand des Messobjektes. Wird kein Objekt erkannt, wird der Echo-Ausgang nach 36 ms auf 0 V gesetzt.

Eine erneute Messung kann 10 ms nach fallender Flanke des Echosignals gestartet werden.

3.3 Analog Discovery Oszilloskop

Das Digilent Analog Discovery 2™ ist ein vielseitiges Multifunktionsinstrument, das in Zusammenarbeit mit Analog Devices® entwickelt wurde. Es ermöglicht Benutzern, gemischte Signalströme jeder Art zu messen, zu visualisieren, zu generieren, aufzuzeichnen und zu steuern.



Abbildung 6: Analog Discovery 2 Oszilloskop [5]

Das Gerät umfasst Funktionen wie ein zweikanaliges, 14-Bit, 100 MS/s Oszilloskop zur Erfassung analoger Daten, einem Funktionsgenerator zur Ausgabe von analogen Spannungen, zwei programmierbare Stromversorgungen, ein einfaches Voltmeter und einen Datenlogger zur Erfassung großer analoger Eingabedaten. Es fungiert auch als Logikanalysator und bietet virtuelle Ein-/Ausgabegeräte durch seine digitalen I/O-Kanäle.

Zusätzlich dient das Analog Discovery 2 als Spektrumanalysator zur Analyse der Frequenzbereichskomponenten von Signalen, als Netzwerkanalysator zur Betrachtung der Amplituden- und Phasenantwort von Schaltkreisen und als Impedanzanalysator für verschiedene Frequenzantwortcharakteristika. Es beinhaltet auch einen Skripteditor zur Automatisierung der Steuerung seiner Instrumente durch JavaScript. Das WaveForms Software Development Kit (SDK) ermöglicht die Entwicklung benutzerdefinierter Anwendungen in mehreren Programmiersprachen und erweitert so seinen Nutzen.

Dieses umfassende Werkzeug integriert sich nahtlos in die WaveForms-Software und bietet eine breite Palette von Funktionalitäten zum Testen und Analysieren elektronischer Schaltkreise.

4 Umsetzung

In diesem Kapitel wird sowohl die Verkabelung des Trainingsboards als auch die Programmierung beschrieben. Der Hardware-Aufbau bleibt hierbei für beide Teilaufgaben identisch. Die Software muss für die unterschiedlichen Anforderungsbereiche angepasst werden.

4.1 Hardware-Aufbau

Die Abstandsmessung erfolgt mit dem Sensor SRF04 und dem XC886 Board.

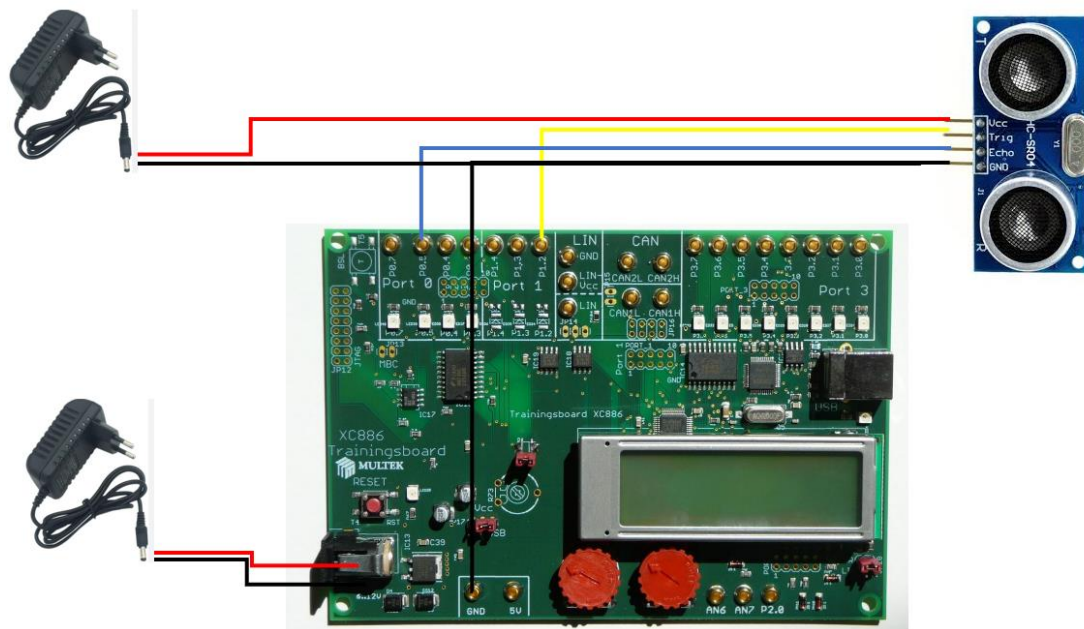


Abbildung 7: Hardwareaufbau und Verdrahtung des gesamten Messaufbaus

Der Sensor wird mit dem XC886-Board wie folgt verbunden:

Stromversorgung: Der SF04-Sensor benötigt eine Versorgungsspannung, die typischerweise bei 5 V DC liegt. Die roten und schwarzen Drähte in der Abbildung stellen die Spannungsversorgung des Sensors sowie des Boards dar.

Trigger- und Echo-Pins: Der Trigger-Pin des Sensors, der für das Auslösen der Ultraschallimpulse verantwortlich ist, wird an den dafür vorgesehenen Port-Pin (hier P1.2) des XC886-Boards angeschlossen, wie im Code als TRIGGER_PIN definiert. Der Echo-Pin, der das reflektierte Signal empfängt, wird ebenfalls an den

entsprechenden Port-Pin (hier P0.5) angeschlossen, der im Code als ECHO_PIN definiert ist. Die gelbe und die blaue Leitung repräsentieren die Signalleitungen für Trigger und Echo. Sie werden an die zuvor erwähnten Pins auf dem XC886-Board angeschlossen, um die Kommunikation zwischen dem Sensor und dem Mikrocontroller zu ermöglichen.

Die korrekte Verdrahtung ist entscheidend, um eine präzise Signalübertragung zu gewährleisten. Ebenso ist es wichtig den Ground vom Sensor und vom Board miteinander zu verbinden damit kein floatendes Signal und somit falsche Signalinterpretationen entstehen.

Nachdem die physischen Verbindungen hergestellt wurden, erfolgt die Initialisierung der Pins und Schnittstellen im Programmcode. Die Konfiguration der Timer und Interrupts auf dem Mikrocontroller ermöglicht eine genaue Zeitmessung der Ultraschall-Signale, die für die Abstandsberechnung erforderlich ist.

4.2 Abstandsmessung 10 cm bis 80 cm

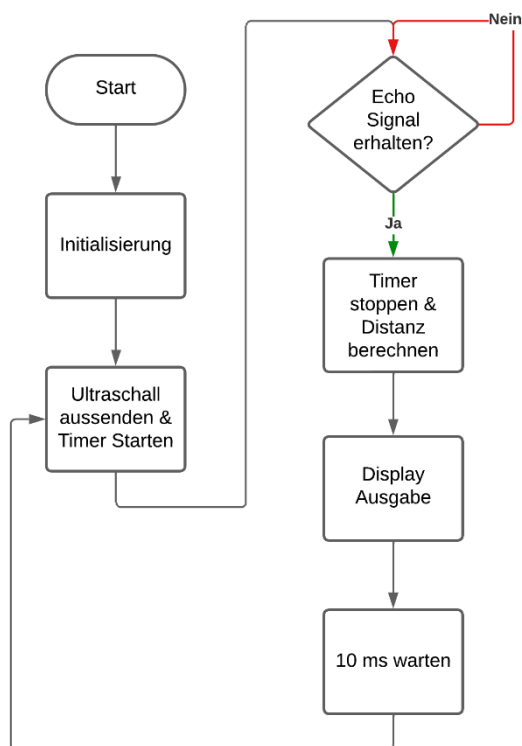


Abbildung 8: Flussdiagramm - Teilaufgabe 1

Im folgenden Abschnitt wird die Teilaufgabe 1 dargestellt. Es handelt sich hierbei um die Programmierung der Entfernungsmessung von 10 cm bis 80 cm. Im Ablaufdiagramm (Abbildung 8: Flussdiagramm - Teilaufgabe 1) kann die Abfolge der einzelnen Schritte nachvollzogen werden. Im Struktogramm (Abbildung 9: Struktogramm - Teilaufgabe 1) ist derselbe Ablauf Anwendungsspezifischer dargestellt. Die Unterkapitel führen durch die Schritte im Struktogramm und Erläutern deren Aufbau sowie Funktion.

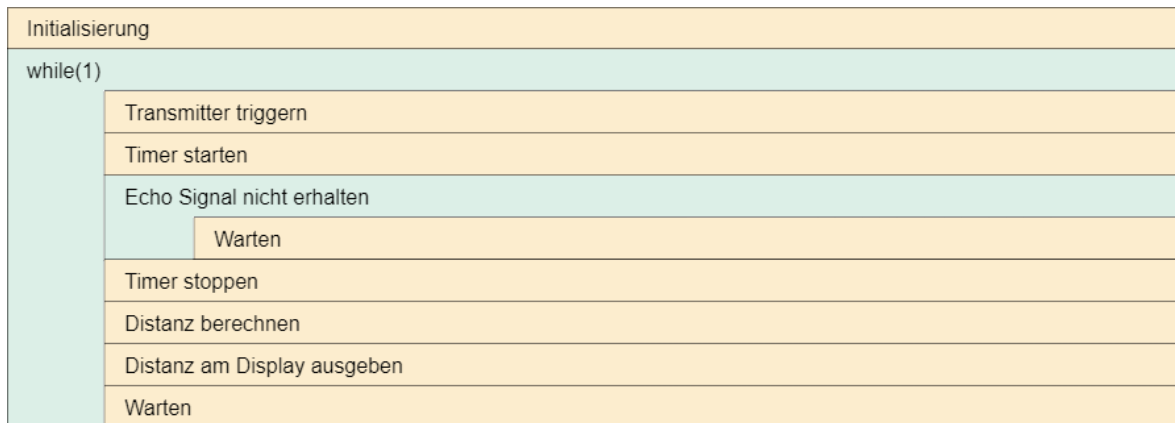


Abbildung 9: Struktogramm - Teilaufgabe 1

Initialisierung

Die Initialisierung des Systems wird durch die Funktion `init()` durchgeführt, die zu Beginn des `main()`-Programms aufgerufen wird. Diese Funktion ist entscheidend, um das System in einen definierten Ausgangszustand zu bringen und umfasst mehrere Schritte:

LCD-Initialisierung: `lcd_init()` konfiguriert das LCD-Display, welches für die Ausgabe der Messergebnisse verwendet wird. Mit `lcd_clr()` wird das Display anschließend gelöscht, um sicherzustellen, dass keine vorherigen Daten angezeigt werden.

Port-Konfiguration: `P0_DIR` und `P1_DIR` sind Konfigurationen für die digitale I/O (Input/Output). Der Wert `0x00` konfiguriert alle Pins von Port 0 als Eingänge, und der Wert `0xFF` konfiguriert alle Pins von Port 1 als Ausgänge. Diese Konfiguration ist wichtig, da der Trigger- und Echo-Pin entsprechend als Ausgang bzw. Eingang eingestellt werden müssen.

Timer-Modus: `TMOD = 0x11`; setzt die Timer 0 und 1 in den 16-Bit-Zählmodus. Dies ist wesentlich für die genaue Zeitmessung, die benötigt wird, um die Dauer des Ultraschallsignals zu erfassen.

Timer 1 Start: `TR1 = 1`; startet Timer 1, der für die `wait`-Funktion verwendet wird, um eine bestimmte Verzögerungszeit zu erzeugen.

Interrupt-Konfiguration: Der externe Interrupt wird mit $EA = 1$; global aktiviert und für den externen Interrupt 0 durch $EX0 = 1$; spezifisch freigeschaltet. $IT0 = 1$; konfiguriert den Interrupt 0 als Edge Triggered, um auf eine Signaländerung zu reagieren, und $EXICON0 = 0x00$; stellt ihn so ein, dass er auf eine fallende Flanke (Übergang von High zu Low) des Echsignals reagiert.

Sensor Triggerung

Im Hauptprogramm `main()` wird nach der Initialisierung in einer Endlosschleife `while(1)` die Funktion `sendTriggerPulse()` aufgerufen, um den Messvorgang zu starten.

Setzen des Trigger-Pins: `P1_DATA = (1 << TRIGGER_PIN)`; setzt den Trigger-Pin auf High. Durch die Bit-Verschiebung wird sichergestellt, dass nur das Bit, das dem Trigger-Pin zugeordnet ist, gesetzt wird, während alle anderen Bits auf ihrem vorherigen Zustand (0 = Low) bleiben.

Wartezeit für Trigger-Impuls: `wait(1)`; ruft eine Verzögerungsfunktion auf. Die Funktion `wait` ist so programmiert, dass sie eine Schleife verwendet, um eine genau definierte Verzögerung zu erzeugen, basierend auf der Timer1-Konfiguration. Jede Iteration der Schleife dauert rechnerisch 10 Mikrosekunden, was durch die Einstellung des Timer1 Registers erreicht wird. Allerdings entspricht es unseren Messungen nach 15 Mikrosekunden (siehe Abbildung 10: Darstellung des 15 μ s langen Triggerimpulses am Oszilloskop). So lange die Funktion läuft ist der Pegel des Trigger-Pins auf 5 V gesetzt.

Rücksetzen des Trigger-Pins: `P1_DATA = ~(1 << TRIGGER_PIN)`; setzt den Trigger-Pin zurück auf Low und beendet somit den Trigger-Impuls.

Diese Triggerung ist notwendig, um den Ultraschall-Sensor zu veranlassen, einen Schallimpuls auszusenden. Die Länge des Impulses muss kurz sein, da die Dauer des Triggersignals die minimale Messdistanz beeinflussen kann.

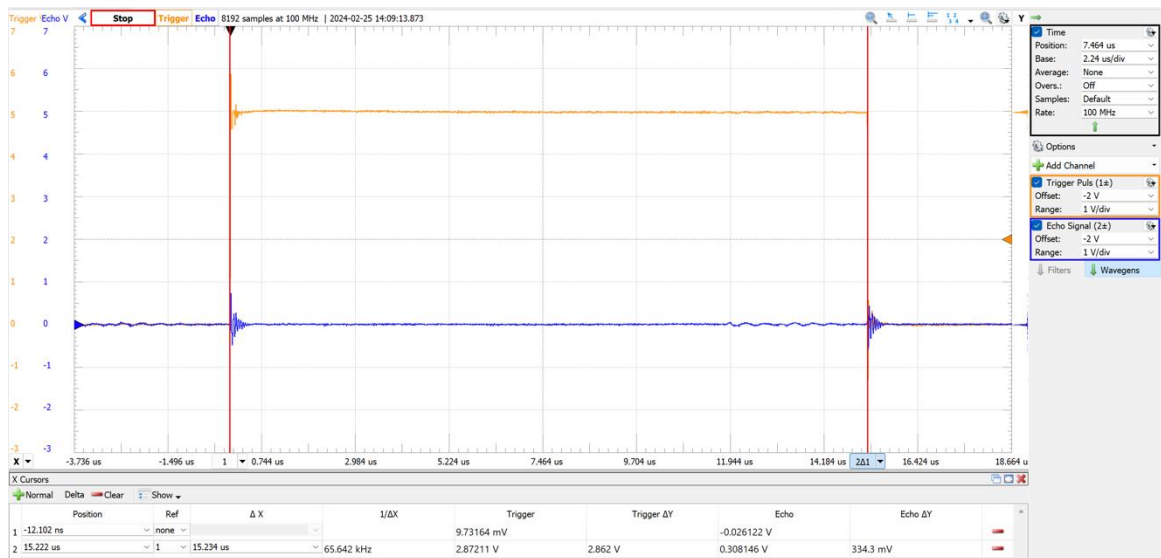


Abbildung 10: Darstellung des 15 µs langen Triggerimpulses am Oszilloskop

Timer Start für Echo-Zeitmessung

Zurücksetzen des Timer-Registers: Die Werte TH0 und TL0 werden auf 0 gesetzt, um den Timer0 zu initialisieren. Dies stellt sicher, dass die Messung der Echo-Laufzeit von einem bekannten Ausgangspunkt startet.

Timer 0 Start: TR0 = 1; schaltet Timer 0 ein, was bedeutet, dass der Timer jetzt läuft und die Zeit zählt, bis das Echo-Signal empfangen und somit eine fallende Flanke erkannt wird.

Die Schleife while(TR0 == 1); hält die Ausführung des Programms an dieser Stelle, bis der externe Interrupt ausgelöst wird, was bedeutet, dass das Echo-Signal empfangen wurde. Die Bedingung in der while Schleife wird vom Interrupt beeinflusst.

Echo Interrupt Service Routine

Timer Stopp bei Echo-Empfang: Wenn das Echo-Signal empfangen wird, wird die Interrupt Service Routine echo_interrupt() ausgelöst, was zur Folge hat, dass TR0 = 0; ausgeführt wird und somit Timer 0 gestoppt wird. Die Interruptfunktion wird mit dem Zusatz interrupt 0 deklariert.

Löschen der Interrupt-Anforderung: IRCON0 = 0; löscht die Interrupt-Anforderung, um sicherzustellen, dass eine erneute Messung möglich ist, andernfalls würde der

Interrupt nur einmal auslösen, da allerdings kontinuierlich gemessen wird, muss die Flag zurückgesetzt werden.

Das Stoppen des Timers bei Empfang des Echsignals ist entscheidend, da die gemessene Zeit direkt zur Berechnung der Entfernung verwendet wird.

Diese Schritte bilden zusammen den Kern des Messvorgangs und sind entscheidend für die Genauigkeit des Systems. Der Code zeigt, dass die Zeitmessung vom Senden bis zum Empfangen des Echsignals genutzt wird, um die Distanz zu einem Objekt zu bestimmen.

Berechnung des Abstands

Der Abstand wird, wie bereits erwähnt, mit den Timer Ticks berechnet. Dafür werden die Register TH0 und TL0 ausgewertet. Da der Timer auf 16 Bit konfiguriert wurde, müssen aus den zwei Registern zunächst die Gesamtzahl der Ticks bestimmt werden. TH0 entspricht den 8 höher Wertigen Bits und TL0 den 8 nieder Wertigen Bits. Mit der Formel $TH = (TH0 \cdot 256) + TL0$ werden die zwei Register zu einem Zusammengefasst. 16 Bit können maximal 65536 Ticks darstellen. Mit der Frequenz des Mikrocontrollers von 12MHz kann dementsprechend eine Zeit von maximal 5,46ms erfasst werden.

$$t[ms] = \frac{65536}{12MHz} = 5,46ms$$

In 5,46ms kann der Schall eine Distanz (bei Raumtemperatur) von 1,82m zurücklegen. Der Reale Abstand zum Objekt entspricht der Hälfte, da der Hin- und Rückweg vom Schall zurückgelegt wird.

$$d[m] = \frac{1}{2} \cdot 344 \text{ m/s} \cdot 5,46 \cdot 10^{-3} \text{ s} = \frac{1}{2} \cdot 1,872m = 0,939 \text{ m}$$

Das entspricht allerdings nicht der realen maximalen messbaren Entfernung, da unser Programm den Timer startet, sobald der Trigger Puls auf Low fällt, bis allerdings der Transmitter das Signal startet vergehen weitere 475 Mikrosekunden (siehe Abbildung 11: Bestimmung des Laufzeit-Offsets). Dieser Offset wird im Code

berücksichtigt, um die Entfernung korrekt zu berechnen. 475 Mikrosekunden entsprechen bei einer Taktfrequenz von 12MHz genau 5700 Ticks vom Timer.

$$offset = 475 \mu s * 12 MHz = 5700$$

Durch die Hardwareseitige Verzögerung des Signals und des Timer Start verringert sich die tatsächlich Messbare Distanz auf ca. 85cm.

$$t[ms] = \frac{65536 - 5700}{12MHz} = 4,98ms$$

$$d[m] = \frac{1}{2} \cdot 344 \frac{m}{s} \cdot 4,98 \cdot 10^{-3} s = \frac{1}{2} \cdot 1,708 m = 0,854 m$$

Die Messanforderung ist es einen Abstand von 10 cm bis 80 cm zu realisieren, so dass diese Anforderung mit dem Offset erfüllt werden kann. Die programmierte Formel entspricht:

$$d[cm] = \frac{344 \frac{m}{s}}{2 \cdot 12MHz} \cdot (ticks - offset) = \frac{172}{120.000} cm \cdot (ticks - offset)$$

$$d[cm] = 43 \cdot (ticks - offset)/30.000$$

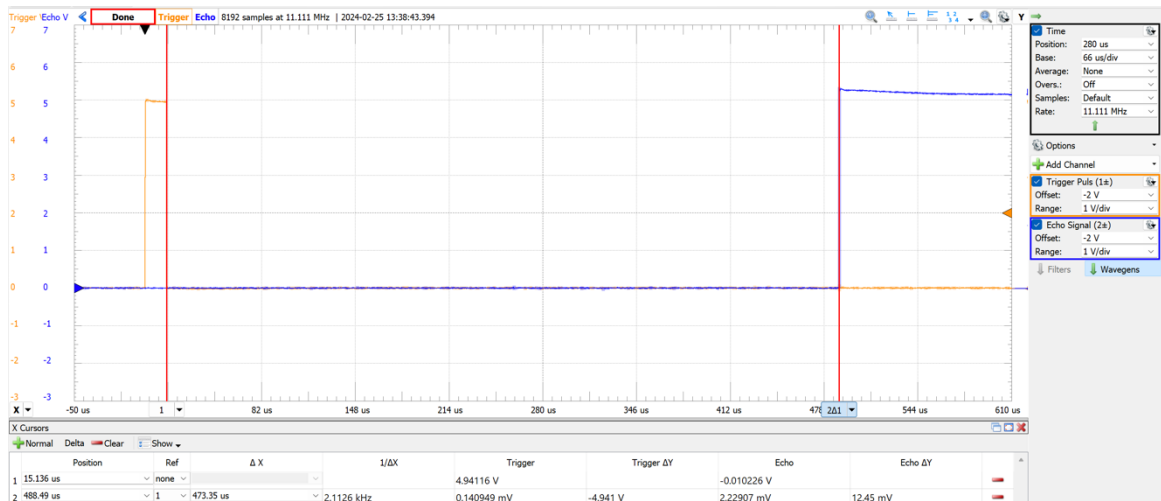


Abbildung 11: Bestimmung des Laufzeit-Offsets

4.3 Erweiterung des Messbereichs

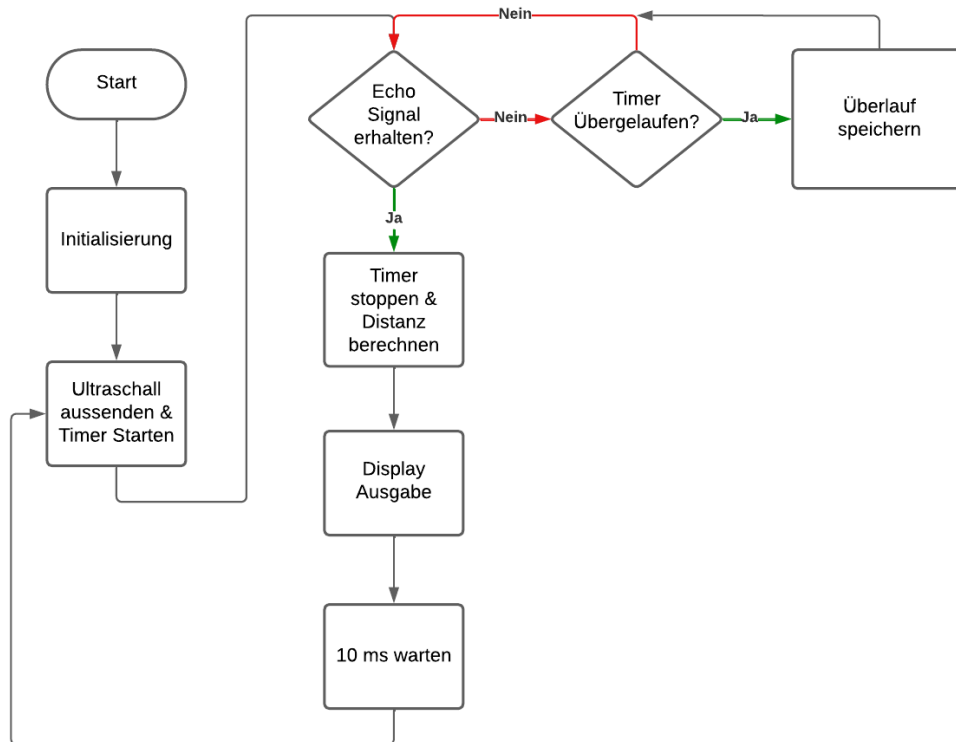


Abbildung 12: Flussdiagramm - Teilaufgabe 2

In der Weiterentwicklung des Abstandsmesssystems wurde die Funktionalität des Codes dahingehend erweitert, dass nun auch größere Entfernungen bis zu 3 Metern erfasst werden können. Wieder sind sowohl Flussdiagramm als auch das Struktogramm zu sehen.

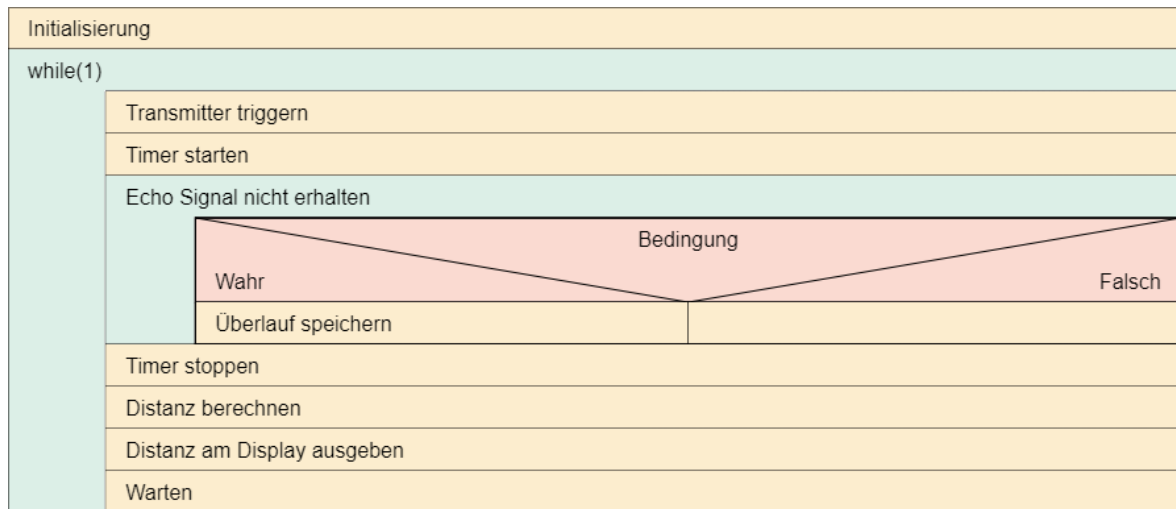


Abbildung 13: Struktogramm - Teilaufgabe 2

Timer Überlauf Handhabung

Um die Erfassung größerer Entfernungen zu ermöglichen, wurde der Code um eine Interrupt Service Routine (ISR) für den Timer-Überlauf ergänzt. Die `timer_ueberlauf()`-Routine wird aktiviert, sobald der Timer 0 sein Zählmaximum überschreitet. Innerhalb dieser Routine wird der Variable `t0_ueberlauf` jeweils der Wert 65536 hinzugeaddiert, welcher der maximalen Zählmenge des 16-Bit-Timers entspricht. Diese Maßnahme erlaubt es, die Zeitspanne für längere Distanzen zuverlässig zu messen. Bei Distanzen größer als 85,4 cm kommt es zu einem Timer-Überlauf. Dieser kann nun mehrmals erfolgen, je nach Entfernung des Objektes zum Sensor. Die oben beschriebene ISR wird nun bei jedem Überlauf aufgerufen.

Berechnung der Distanz

Die Funktion `calculate_distance()` wurde modifiziert, um die Distanz akkurat zu kalkulieren. Hierfür wird die Summe aus den Timer-Überläufen und den aktuellen Timer-Werten gebildet und ein zuvor bestimmter Offset subtrahiert. Dieser Schritt ist notwendig, um die reine Laufzeit des Ultraschall-Echos ohne systembedingte Verzögerungen zu erfassen. Die resultierende Tick-Anzahl wird dann in Mikrosekunden umgerechnet und zur Berechnung der Distanz in Zentimetern verwendet. Nach jedem Messvorgang erfolgt ein Zurücksetzen des `t0_ueberlauf`-Zählers.

LCD-Anzeige

Eine wesentliche Erweiterung betrifft die `lcd_distance()`-Funktion, welche nun die Behandlung von Messwerten außerhalb des spezifizierten Messbereichs vornimmt. Für Werte, die größer als 3 Meter oder kleiner als 3 Zentimeter sind, werden entsprechende Warnmeldungen auf dem LCD angezeigt. Innerhalb des zulässigen Bereichs werden die Distanzwerte in einer strukturierten Form präsentiert, wobei leere Stellen nur angezeigt werden, wenn sie für die Wertdarstellung relevant sind.

5 Ergebnisse & Fazit

Durch die Integration dieser Funktionen in den Code konnte das Messsystem erfolgreich auf einen erweiterten Bereich von 3 Zentimetern bis zu 3 Metern ausgebaut werden. Die sorgfältige Implementierung der Timer-Überlaufbehandlung und die präzise Berechnung der Entfernungen gewährleisten eine hohe Genauigkeit der Messergebnisse. Die benutzerorientierte Anzeige der Messwerte auf dem LCD erhöht zudem die Benutzerfreundlichkeit des Systems.

Problematik der Messung von geringen Distanzen mittels Ultraschallsensoren

Bei der Konzeption des Abstandsmesssystems, das auf der Ultraschalltechnologie basiert, ergibt sich eine inhärente Schwierigkeit bei der Erfassung von sehr geringen Distanzen. Der implementierte SRF04 Ultraschallsensor ist aufgrund seiner Bauweise und Funktionsprinzipien nicht in der Lage, Objekte zu detektieren, die näher als 3 Zentimeter am Sensor platziert sind. Dieses Limit ist durch die physischen Eigenschaften der Schallausbreitung und die konstruktive Gestaltung des Sensors bedingt.

Ultraschallsensoren messen Distanzen durch das Aussenden von Schallwellen und das Empfangen des Echos dieser Wellen nach der Reflexion an einem Objekt. Bei sehr kurzen Distanzen kann es vorkommen, dass das Echo des ausgesendeten Signals nicht zurück zum Empfängerteil des Sensors gelangt. (Abbildung 14: Sensorsignal, wenn kein Objekt erkannt wird) Ein Grund hierfür ist, dass bei einer

derartigen Nähe des Objekts zum Sensor die Schallwellen direkt absorbiert oder seitlich wegreflektiert werden können, ohne dass sie den Empfänger erreichen. Wenn beispielsweise ein Objekt unmittelbar vor dem Sensor platziert wird, kann dies dazu führen, dass der Sensor kein Echo empfängt und daher fälschlicherweise annimmt, dass kein Objekt vorhanden ist.

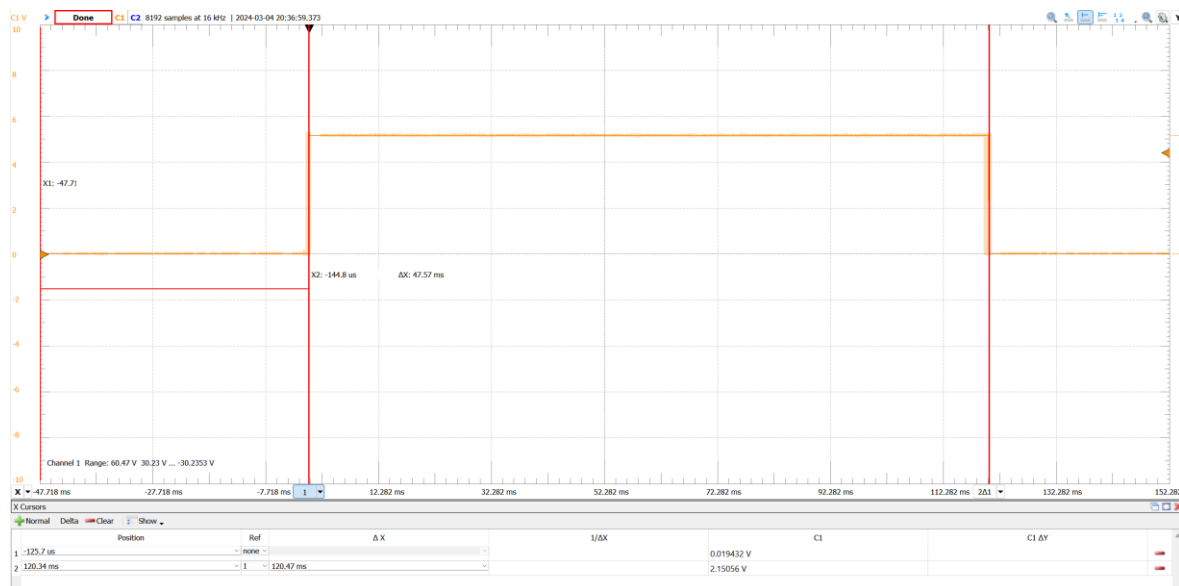


Abbildung 14: Sensorsignal, wenn kein Objekt erkannt wird

Diese Begrenzung ist nicht nur auf die physische Blockierung des Echosignals zurückzuführen, sondern auch auf die zeitliche Überlappung von Sende- und Empfangssignal bei sehr kurzen Distanzen. Die Zeit, die das Signal benötigt, um zum Objekt und zurück zum Sensor zu gelangen, kann so kurz sein, dass das Echo noch während der Dauer des ausgesendeten Impulses ankommt. Das ist problematisch, da der Empfänger erst kurze Zeit nach dem Aussenden des Signal scharf gestellt wird. Wird der Empfänger erst aktiv so bald schon einige Bursts angekommen sind, erkennt er die Sequenz entweder falsch oder gar nicht. Die zurückgegebene Distanz ist somit fehlerhaft.

Messergebnisse

Mit folgendem Aufbau wurde die Referenzmessung durchgeführt. Der Transceiver (rot) wurde am Rand einer Erhöhung platziert. Die Membran wurde als Nullpunkt festgelegt. Als zu messendes Objekt dient ein Karton, der nun immer weiter vom

Sensor wegbewegt wird. Parallel zum Ablesen des Abstandes am Display wurde zu jedem Messpunkt auch eine Bild mit dem Oszilloskop aufgenommen. Die Messergebnisse sowie die Abweichungen sind der Tabelle zu entnehmen.

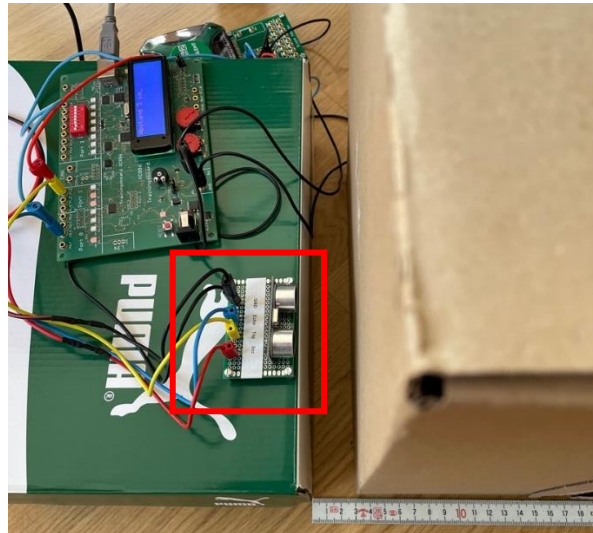


Abbildung 15: Messaufbau zur Bestimmung der Genauigkeit

Referenzabstand [cm]	Gemessener Abstand [cm]	Abweichung [abs]	Abweichung [%]
3	3	0	0
10	10	0	0
20	20	0	0
30	30	0	0
40	40	0	0
50	50	0	0
60	60	0	0
70	69	1	1,428571429
80	79	1	1,25
90	90	0	0
100	99	1	1
120	119	1	0,833333333
140	139	1	0,714285714
160	159	1	0,625
180	179	1	0,555555556
200	199	1	0,5
250	249	1	0,4
300	298	2	0,666666667

Tabelle 2: Messergebnisse

Bezüglich der Aufnahme der Messpunkte ist zu sagen, dass eine wirkliche Referenzmessung nur bedingt möglich ist. Insgesamt besteht im Aufbau eine große Toleranzkette bestehend aus: Platzierung des Sensors an der Kante, Platzierung des Meterstabs zur Kante, Toleranz des Meterstabs, sowie Toleranz bei der Platzierung des zu messenden Gegenstands.

Wie der Tabelle zu entnehmen ist, stimmen die gemessenen Werte sehr gut mit dem Referenzabstand überein. Die maximale absolute Abweichung beträgt 2 cm bei einem realen Abstand von 3 m. Prozentual beträgt diese Abweichung nur 0.66%.

In den beiden Abbildungen sind beispielhaft die Oszilloskopbilder der Laufzeit Messungen für 10 cm (Abbildung 16: Oszilloskopbild zur Abstandsmessung von 10cm) und 3 m (Abbildung 17: Oszilloskopbild zur Abstandsmessung von 300cm) zu sehen.

Die Laufzeit des ersten Screenshots beträgt dabei 1.0952 ms. Diese Zeit entsprechen mit 12 MHz Timertakt 13142 Ticks. Mit dem Abzug des Offsets von 5700 und der Formel in Kapitel 4.2 entspricht das einem Abstand von 10 cm.

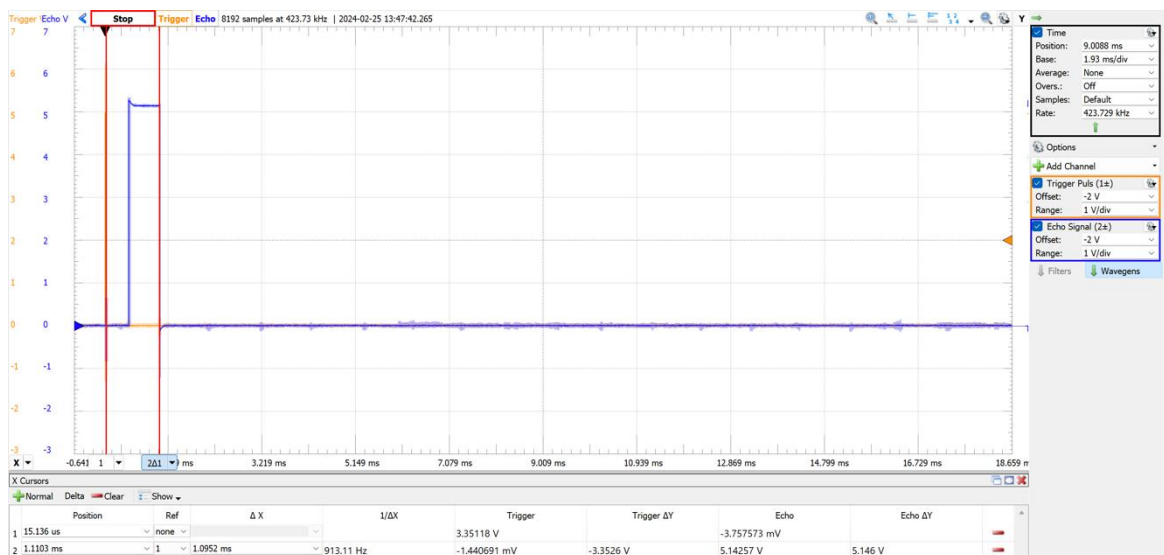


Abbildung 16: Oszilloskopbild zur Abstandsmessung von 10cm

Analog zum ersten Messpunkt berechnet sich für eine Zeit von 17,873 ms ein Abstand von 299 cm.

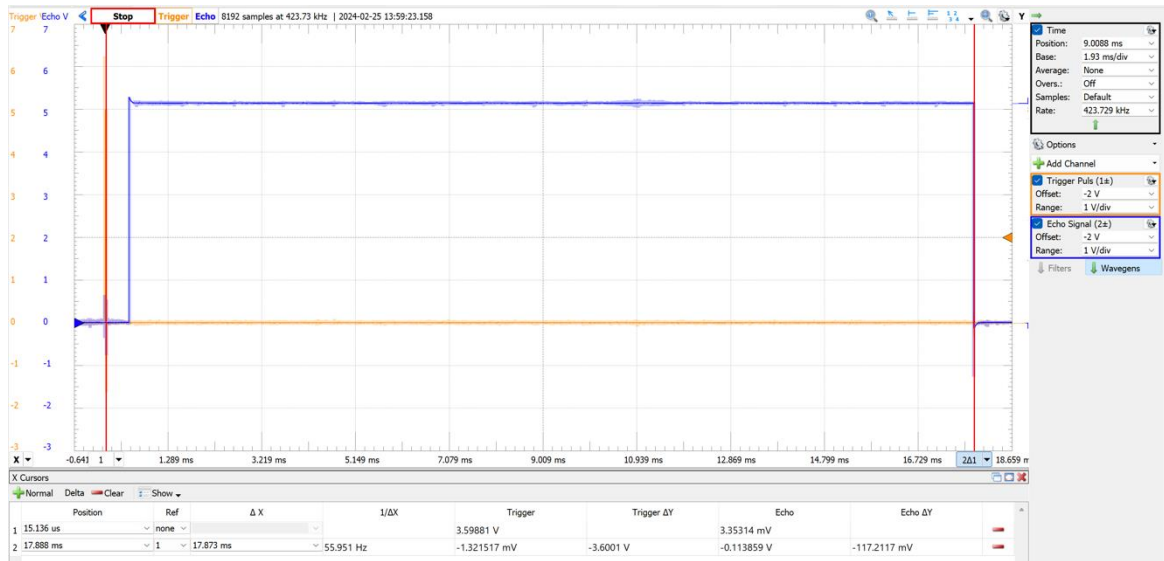


Abbildung 17: Oszilloskopbild zur Abstandsmessung von 300cm

Literaturverzeichnis

- [1] Bosch Mobility Solutions, „Ultraschallsensor,“ 2024. [Online]. Available: <https://www.bosch-mobility.com/de/loesungen/sensoren/ultraschallsensor/>. [Zugriff am 29 02 2024].
- [2] B. Mirwald, „Vom unhörbaren Ton zum sichtbaren Bild – wie funktioniert Ultraschall?,“ 20 12 2012. [Online]. Available: <https://www.weltderphysik.de/thema/hinter-den-dingen/ultraschall/>. [Zugriff am 29 02 2024].
- [3] E. Hering, Physik für Ingenieure, Springer Vieweg, 2021.
- [4] Parallax, „Datasheet4U,“ [Online]. Available: <https://datasheet4u.com/datasheet-pdf/Parallax/SRF04/pdf.php?id=523588>. [Zugriff am 29 02 2024].
- [5] Digilent, „Analog Discovery 2,“ 2024. [Online]. Available: https://digilent.com/reference/_media/reference/test-and-measurement/analog-discovery-2/analog_discovery_2_obl_600.png. [Zugriff am 29 02 2024].

Anhang

Programmcode 10cm bis 80cm

```
#include <xc886.h> // XC Board Bibliothek

#include <DIP204_XC886.c> // Display Funktionen


// Definition der Pins für den Ultraschall-Sensor

#define TRIGGER_PIN 2 // P1.2 als Trigger-Pin für den Ultraschall-Sender
#define ECHO_PIN 5 // P0.5 als Echo-Pin für den Ultraschall-Empfänger


// Variablen für die Entfernungsmessung

unsigned long distance; // Entfernung in Zentimetern
unsigned long TH; // Zeit in Timer-Ticks


void init(void);

void wait(int t);

void sendTriggerPulse(void);

void lcd_distance(unsigned long value);


// Hauptfunktion

void main(void)
{
    init(); // Initialisiert die Hardware


    while(1)
    {
        sendTriggerPulse(); // Sendet einen Trigger-Impuls


        // Startet Timer 0 für die Zeitmessung

        TH0 = 0;
        TL0 = 0;
        TR0 = 1;


        // Warten, bis Timer 0 stoppt -> erfolgt im Interrupt

        while(TR0 == 1);
    }
}
```

```
// Berechnet die Entfernung basierend auf Timer-Werten
TH = (TH0 * 256 + TL0) - 5703; // Korrigiert um Offset
distance = (TH * 80) / 55765; // Umrechnung in Zentimeter

// Zeigt die Entfernung auf dem LCD an
lcd_distance(distance);

wait(50000); // Wartezeit zwischen den Messungen
}
}

// Initialisierungsfunktion für Hardware und Timer
void init(void)
{
    lcd_init();      // Initialisiert das LCD
    lcd_clr();       // Löscht das LCD

    P0_DIR = 0x00;    // Konfiguriert Port 0 als Eingang
    P1_DIR = 0xFF;    // Konfiguriert Port 1 als Ausgang

    TMOD = 0x11;      // Timer0 und Timer1 als 16-Bit-Zähler initialisiert

    // Timer 1 für die Wartefunktion konfigurieren
    TR1 = 1;          // Startet Timer 1

    // Externer Interrupt für Echo-Pin konfigurieren
    EA = 1;           // Global Interrupt Enable
    EX0 = 1;          // External Interrupt 0 Enable
    IT0 = 1;          // Interrupt 0 auf Edge Triggered einstellen
    EXICON0 = 0x00;    // Interrupt auf fallende Flanke einstellen
}

// Wartefunktion, blockiert für 't' Schleifendurchläufe
void wait(int t)
```

```
{  
  
    // Eine Schleife entspricht 10uS -> 0xffff - ( 10us - 0.0833us) == 0xff87  
  
    // Entspricht allerdings laut Messung ca 15us  
  
    unsigned int i;  
    for(i = 0; i < t; i++)  
    {  
        TH1 = 0xFF;  
        TL1 = 0x87;  
        while(TF1 == 0);  
        TF1 = 0;  
    }  
}  
  
  
// Interrupt Service Routine für das Echo-Signal  
void echo_interrupt() interrupt 0 {  
    TR0 = 0; // Stoppt Timer 0 bei Echo-Empfang  
    IRCON0 = 0; // Löscht Interrupt-Anforderung  
}  
  
  
// Funktion zur Ausgabe eines langen Wertes auf dem LCD  
void lcd_distance(unsigned long value)  
{  
    unsigned char i;  
  
    lcd_curs(0); // Setzt Cursor zurück  
    lcd_clr();  
    lcd_str("Abstand ");  
  
    if(value > 80)  
    {  
        lcd_str("zu gross!");  
        lcd_curs(20);  
        lcd_str("Mehr als 80 cm");  
    }  
}
```

```

        else if(value < 10)
        {
            lcd_str("zu klein!");
            lcd_curs(20);
            lcd_str("Kleiner als 10 cm");
        }
        else
        {
            i = value / 10;
            value %= 10;
            if(i != 0) asc_out(i + 0x30); // Hundertstel
            asc_out(value + 0x30);
            lcd_str(" cm");
        }
        lcd_curs(0); // Setzt Cursor zurück
    }

// Funktion zum Senden eines Trigger-Impulses an den Ultraschall-Sender
void sendTriggerPulse(void) {
    P1_DATA = (1 << TRIGGER_PIN); // Setzt Trigger-Pin High
    wait(1);                      // Kurze Wartezeit
    P1_DATA = ~(1 << TRIGGER_PIN); // Setzt Trigger-Pin Low, beendet den Impuls
}

```

Programmcode 3cm bis 3m

```

#include <xc886.h> // XC Board Bibliothek
#include <DIP204_XC886.c> // Display Funktionen

// Definition der Pins für den Ultraschall-Sensor
#define TRIGGER_PIN 2 // P1.2 als Trigger-Pin für den Ultraschall-Sender
#define ECHO_PIN 5    // P0.5 als Echo-Pin für den Ultraschall-Empfänger

// Variablen für die Entfernungsmessung
unsigned long distance; // Entfernung in Zentimetern

```

```
unsigned long passed_time; // Zeit in Timer-Ticks
unsigned long passed_ticks; // Zeit in Timer-Ticks
unsigned long t0_ueberlauf; // Überlauf für Messbereichserweiterung

// Eigene Funktionen
void init(void);
void calculate_distance(void);
void wait(int t);
void sendTriggerPulse(void);
void lcd_distance(unsigned long value);

// Hauptfunktion
void main(void)
{
    init(); // Initialisiert die Hardware

    while(1)
    {
        sendTriggerPulse(); // Sendet einen Trigger-Impuls

        while(TR0 == 1); // Warten, bis Timer 0 stoppt -> erfolgt im Interrupt

        calculate_distance(); // Berechnung der Distanz

        lcd_distance(distance); // Zeigt die Entfernung auf dem LCD an

        wait(50000); // Wartezeit zwischen den Messungen
    }
}

// Interrupt Service Routine für das Echo-Signal
void echo_interrupt() interrupt 0
{
    TR0 = 0; // Stoppt Timer 0 bei Echo-Empfang
```

```
IRCON0 = 0; // Löscht Interrupt-Anforderung
}

// Interrupt Service Routine für den Überlauf des Timer0
void timer_ueberlauf(void) interrupt 1
{
    t0_ueberlauf += 65536; // Addiert die maximale Zählmenge bei einem Überlauf
}

// Initialisierung der Flags und Bildschirms
void init(void)
{
    lcd_init();      // Initialisiert das LCD
    lcd_clr();       // Löscht das LCD

    P0_DIR = 0x00;    // Konfiguriert Port 0 als Eingang
    P1_DIR = 0xFF;    // Konfiguriert Port 1 als Ausgang

    TMOD = 0x11;      // Timer0 und Timer1 als 16-Bit-Zähler initialisiert
    TR1 = 1;          // Startet Timer 1

    EA = 1;           // Global Interrupt Enable
    EX0 = 1;          // External Interrupt 0 Enable
    ET0 = 1;          // Interrupt Timer0 Enable
    IT0 = 0;          // Interrupt 0 auf Edge Triggered einstellen
    //EXICON0 = 0x00; // Interrupt auf fallende Flanke einstellen
}

// Berechnet die Entfernung basierend auf Timer-Werten
void calculate_distance(void)
{

```



```
passed_ticks = t0_ueberlauf + (TH0 * 256 + TL0) - 5703; // Korrigiert um Offset: Zeit zwischen Trigger  
Signal und Echo High
```

```
passed_time = passed_ticks * 43; //
```

```
distance = passed_time / 30000; // Umrechnung in Zentimeter
```

```
t0_ueberlauf = 0; // Zurücksetzen des Überlaufs
```

```
}
```

```
// Wartefunktion, blockiert für 't' Schleifendurchläufe. Ein Schleifendurchlauf entspricht 15uS.
```

```
void wait(int t)
```

```
{
```

```
    unsigned int i;
```

```
    for(i = 0; i < t; i++)
```

```
    {
```

```
        TH1 = 0xFF;
```

```
        TL1 = 0x87; // Eine Schleife entspricht 10uS -> 0xffff - ( 10us - 0.0833us) == 0xff87 ; Entspricht allerdings laut Messung  
ca 15us
```

```
        while(TF1 == 0); // Wartet auf Timer-Flag
```

```
        TF1 = 0; // Setzt Timer-Flag zurück
```

```
    }
```

```
}
```

```
// Funktion zum Senden eines Trigger-Impulses an den Ultraschall-Sender
```

```
void sendTriggerPulse(void)
```

```
{
```

```
    TH0 = 0; // Timer 0 zurücksetzen
```

```
    TL0 = 0; // Timer 0 zurücksetzen
```

```
    P1_DATA = (1 << TRIGGER_PIN); // Setzt Trigger-Pin High
```

```
    wait(1); // Kurze Wartezeit
```

```
    P1_DATA = ~(1 << TRIGGER_PIN); // Setzt Trigger-Pin Low, beendet den Impuls
```

```
    TR0 = 1; // Startet Timer 0 für die Zeitmessung
```

```
}
```

```
// Funktion zur Ausgabe eines langen Wertes auf dem LCD
void lcd_distance(unsigned long value)
{
    unsigned char i, j;

    lcd_curs(0); // Setzt Cursor zurück
    lcd_clr();
    lcd_str("Abstand ");
    //lcd_curs(8); // Setzt Cursor zurück

    if(value > 300)
    {
        lcd_str("zu gross!");
        lcd_curs(20);
        lcd_str("Mehr als 3m");
    }

    else if(value < 3)
    {
        lcd_str("zu klein!");
        lcd_curs(20);
        lcd_str("Kleiner als 3cm");
    }

    else
    {
        i = value / 100;
        value %= 100;
        if(i != 0) asc_out(i + 0x30); // Hundertstel

        j = value / 10;
        value %= 10;
```

```
        if(j != 0 || (j == 0 && i != 0)) asc_out(j + 0x30); // Zehner, korrigiert: ausgegeben nur wenn j
nicht 0

        value += 0x30;

        asc_out((char)value); // Einer

        lcd_str(" cm");

    }

}
```