

Diplomarbeit

Arne Wischmann

Sonarsensorik für mobile Roboter

Studiengang Technische Informatik
Betreuender Prüfer: Prof. Dr. Kai von Luck
Zweitgutachter: Prof. Dr. Gunter Klemke
Abgegeben am 21. August 2002

Sonarsensorik für mobile Roboter

Stichworte

Mobile Roboter, Microcontoller, Ultraschall, Navigation, Triangulation

Zusammenfassung

Die vorliegende Diplomarbeit befasst sich mit der Entwicklung eines Microcontroller gestützten Sonarsystems für mobile Roboter. Sie beschreibt die Auswahl und Entwicklung der erforderlichen Hardware sowie die Möglichkeit durch Sensoren mit weitem Öffnungswinkel, die für die Bestückung eines Roboters erforderliche Anzahl zu senken. Es wird gezeigt, dass es möglich ist, bei einer reduzierten Zahl von Sensoren durch Einsatz eines Microcontrollers alle erforderlichen Richtungsinformationen zu gewinnen.

An ultrasonic sensor system for mobile robots

Keywords

mobile robot, microcontroller, ultrasonic, navigation, triangulation

Abstract

This final thesis (diploma) deals with the development of a microcontroller based ultrasonic sensor system for mobile robots. It describes the process of selecting and developing the required hardware and the possibility of reducing the number of sensors needed by using special sensors with a wide beampattern. In addition to this it will be demonstrated that it is possible to get the same supply of direction data from a reduced amount of sensors by processing the incoming ultrasonic data with a microcontroller.

Inhaltsverzeichnis

1	Einleitung	5
1.1	Motivation	9
1.2	Überblick über die Arbeit	12
2	Auswahl eines Microcontrollers	13
2.1	Mögliche Architekturen	14
2.1.1	CISC	14
2.1.2	RISC	15
2.2	Mögliche Controller-Familien	16
2.2.1	Infineon C-166-Familie	16
2.2.2	Advanced Risc Machines – ARM-Familie	17
2.3	Die Entscheidung	20
3	Software und Library	21
3.1	ARM SDT - Software Development Toolkit	21
3.2	Die Atmel Library	24
3.3	Fazit	25
4	Ultraschall	26
4.1	Murata MA40R S	28
4.2	Polaroid	30
4.3	Airmar AT Serie	31
4.4	Marco	33
4.5	Zusammenfassung	34
5	Die Hardware	36
5.1	Evaluation Board AT91EB55	37
5.2	Marco Ultraschall Sende- Empfangssystem	38
5.3	Das Interface	40
5.3.1	Spannungsanpassung und Multiplexing	41
5.3.2	Gleichrichtung	42

5.3.3	Netzteil	42
5.4	Die Firmware	43
5.5	Hardware Tests	45
5.5.1	Burst Test	46
5.5.2	Analog Tests	49
5.6	Zusammenfassung	52
6	Erste Sonarmessungen	54
6.1	Sinnvolle Sensorenanordnungen	54
6.2	Leerlaufstest	55
6.3	Das Referenzobjekt	58
6.4	Paralleler Aufbau	59
6.5	Konvexer Aufbau	60
6.6	Konkaver Aufbau	62
6.7	Nachweis der Tauglichkeit für Triangulation	63
6.8	Fazit	64
7	Resümee	65
7.1	Zusammenfassung	65
7.2	Ausblick	66
A	Inhalt der CD-ROM	72

Kapitel 1

Einleitung

Roboter¹ gewinnen in unserer Umwelt mehr und mehr an Bedeutung. Unter den stationären Industrie-Robotern gibt es immer mehr hochspezialisierte, perfekt arbeitende Automaten, die in fast alle Bereiche der automatisierten Fertigung vorgedrungen sind und deren Möglichkeiten erheblich erweitert haben.

Beflügelt durch die immer weiter steigende Integrationsdichte und Leistungsfähigkeit moderner Hardware, gewinnt ein neuer Sektor der Robotik immer mehr an Bedeutung: Service-Roboter heißt das Schlagwort. Service-Roboter sind meist wesentlich kleiner als ihre „Kollegen“ aus der Fertigungsindustrie und normalerweise mobil. Sie wurden erdacht, dem Menschen einen Teil der täglichen Arbeit, wie zum Beispiel Staubsaugen (s. Abbildung 1.1) und Rasenmähen abzunehmen. Insbesondere in für Menschen unwirtschaftlichen oder gefährlichen Umgebungen gewinnen diese autonomen Service-Roboter immer mehr an Bedeutung.



Abbildung 1.1: Staubsauger Robot Trilobite von Electrolux

¹Der Begriff Roboter geht auf den tschechischen Schriftsteller Karel Capek zurück, der ihn 1921 von dem tschechischen Wort „robota“ ableitete, was soviel wie Zwangsarbeit bedeutet.

Was genau ist ein Roboter? Laut der Robotics Industry Association (RIA) ist ein Roboter „a re-programmable, multi-functional, manipulator designed to move material, parts, tools, or specialized devices through variable programmed motions for the performance of a variety of tasks“ (Jablosky und Posey 1985)[Arkin 1998, S. 1]. Diese Definition beschreibt gewiss einen großen Teil der Industrie-Roboter, lässt aber die genannten Service-Roboter weitgehend unbeachtet. Meines Erachtens definierte Ronald C. Arkin 1998 den Bereich der Service-Roboter umfassender und treffender: „...an intelligent robot is a machine able to extract information from its environment and use knowledge about its world to move safely in a meaningful and purposive manner.“ [Arkin 1998, S. 2]

Einige der von Arkin definierten Vertreter seien hier vorgestellt:

Am 6 Juli 1997 landete der auf Abbildung 1.2 gezeigte **Sojourner Rover** [King 2002] der NASA auf dem Mars. Hierbei handelte es sich um ein partiell autonomes System, da es seine Aufträge von der Erde bekam. Allerdings musste der Rover in der Lage sein, diese Aufträge selbstständig zu erfüllen, ohne sich dabei zu beschädigen. Da die Kommunikation zwischen dem Mars und der Erde aufgrund der großen Entfernung zwischen 10 und 15 Minuten verzögert ist, hätte es keine Möglichkeit gegeben, spontan von der Erde aus einzugreifen. Der Rover war in der Lage, Farb- und Schwarzweiß-Bilder des Mars' zur Erde zu schicken sowie mit Hilfe eines Alpha-Proton-Röntgen Spectrometers Analysen von Staub, Boden- und Gesteinsproben zu Erde zu funken. Seine Mission endete nach 84 Tagen am 27. September 1997, als die Funkverbindung aus ungeklärten Gründen plötzlich abbrach.

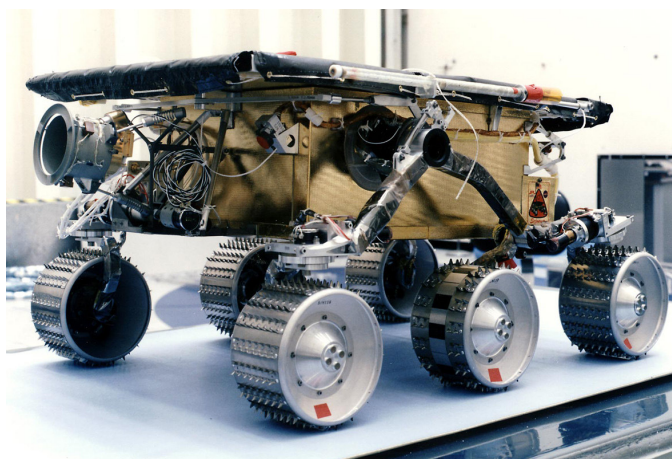


Abbildung 1.2: Sojourner Rover im Labor der NASA



Abbildung 1.3: Kanaluntersuchungsroboter MAKRO

Bei dem in Abbildung 1.3 gezeigten Roboter handelt es sich um den Kanaluntersuchungsroboter **MAKRO** [GMD]. Dieser mehrgliedrige Roboter wird entwickelt, um unwegsame Abwasserkanäle autonom zu durchfahren und dabei nach Beschädigungen zu suchen. Die Besonderheit dieser Konstruktion ist im rechten Teil der Abbildung 1.3 zu sehen. Alle Segmente sind durch elektrisch verfahrbare Kugelgelenke verbunden, was dem Roboter ermöglicht, sich wie eine Schlange fortzubewegen. Er ist somit in der Lage Hindernisse höher als er selber zu überwinden sowie größere Öffnungen zu überfahren. Er ist extrem an seine Einsatzumgebung, den Abwasserkanal, angepasst.

Seine Navigationssysteme sind ebenfalls auf die Kanalwelt abgestimmt. Jedes Robotersegment misst seine Entfernung mit Hilfe eines Infrarotstrahls nach links und rechts zur Kanalwand, um seitliche Kollisionen zu vermeiden. Nach vorne und hinten wird mittels eines Lasers ein Kreuz in den Kanal projiziert und mit einer Videokamera aufgezeichnet. Aus den Verzerrungen des Kreuzes auf den Kanalwänden kann man die Lage des Roboters sowie die Form des vor dem Roboter liegenden Kanals berechnen, so dass der Roboter in der Lage ist, sicher durch den dunklen Kanal zu navigieren.

Der Roboter **MOSRO 1** [Robowatch], zu sehen auf Abbildung 1.4, ist ein Überwachungs-Roboter. MOSRO 1 wurde von der Firma Robowatch Technologies GmbH in Berlin entwickelt und wird als „Nachtwächter“ eingesetzt. Er ist in der Lage, bis zu 18 Stunden ein Objekt selbstständig zu überwachen, indem er einen ihm vorher eingegebenen Weg entlang fährt und dabei mit einer Vielzahl von Sensoren die Umgebung abtastet. Da seiner Sensorausstattung praktisch keine Grenzen gesetzt sind, kann er sowohl für die Bewachung eines Objektes eingesetzt werden, als auch zur Überwachung von Produktionsanlagen, deren unmittelbares Umfeld für Menschen zu gefährlich ist.



Abbildung 1.4: Überwachungsroboter MOSRO 1

Der für die Bewachung von Objekten wie Hallen, Messeständen oder Betriebsgebäuden eingesetzte Roboter, ist unter anderem mit passiven Infrarot- und Radarsensoren ausgestattet. Diese ermöglichen ihm, in einem großen Umkreis alle Bewegungen sogar durch Wände hindurch wahrzunehmen. Kommt es zum Alarm, fordert der Roboter die vermeintliche Person auf, sich mittels Fingerprintsensor zu authentifizieren. Schlägt dies fehl, schaltet der Roboter optischen und akustischen Alarm. Über WLAN² ist er ständig mit einer Einsatzzentrale verbunden, zu der permanent seine Position, ein Videobild und der Zustand aller Sensoren übermittelt wird.

Die Navigation des Roboters erfolgt mit Ultraschall. Mit Hilfe von 16 Sensoren tastet er die Umgebung auf Hindernisse ab. Diese Sensoren gehören zur Kategorie der Laufzeitsensoren. Ein Ton wird ausgesandt und dann auf ein Echo gewartet. Die Zeit, die zwischen Aussenden des Tons und dem Empfang des Echos vergeht, lässt sich in eine Entfernung umrechnen.

Alle genannten Roboter gehören zur Familie der Service-Roboter. Für diese Familie soll die vorliegende Diplomarbeit Alternativen im Bereich der Sensorbe-

²WLAN - Wireless Local Area Network

stückung mit Ultraschall sowie deren Auswertung aufzeigen. Insbesondere MOSRO der Firma Robowatch, welche mich bei diesem Projekt unterstützte, weckte mein Interesse dahingehend, mit weniger Ultraschallsensoren eine gleichwertige, wenn nicht sogar bessere Erfassung der Umgebung zu ermöglichen.

1.1 Motivation

Sonarsysteme gehören heutzutage bei fast allen Service-Robotern zur Grundausstattung, da sie ein sehr gutes Kosten-Nutzen-Verhältnis aufweisen. Es gibt zwar wesentlich präziser arbeitende Systeme wie Laserscanner oder Stereokamera, die aber aufgrund der Tatsache, dass sie auf optischen Wirkungsprinzipien basieren, nicht ohne weiteres Sonarsysteme ersetzen können. Abgesehen von der nicht unerheblichen Rechenleistung, die für ihre Auswertung benötigt wird, führen Anwendungsgebiete in dunklen, nebeligen oder staubigen Umgebungen optische Systeme schnell an ihre Grenzen.

Der am weitesten verbreitete Ultraschallsensor ist das Polaroid Ranging System, bestehend aus einem Transducer der 600er Serie sowie dem 6500er Rangingsystem [Polaroid]. Die Transducer haben eine kegelförmige Richtcharakteristik bei ca. 13° bis 15° Abstrahlwinkel. Um einem Fahrzeug mit diesen Sensoren eine Rundumsicht (360°) zu ermöglichen, benötigt man 24 Sensoren, die im 15° Winkel angeordnet sind [Moravec 1988]. Aus dieser Technik ergeben sich derzeit vor allem 3 Probleme:

Erstens sind 24 Sensoren unter den Gesichtspunkten der Baukosten, der Datenauswertung und der Konstruktionstechnik eine erhebliche Anzahl. Selbst wenn wie bei MOSRO keine Rundumsicht realisiert wird, werden immer noch verhältnismäßig viele Sensoren benötigt, um eine ausreichende Abdeckung der Umgebung zu ermöglichen.

Zweitens warten die Polaroid Ranging Systeme „nur“ auf das erste Echo, welches sie erreicht. Das bedeutet, dass immer nur das am nächsten liegende Hindernis erkannt wird. Oszillographiert man das ankommende Echo, stellt man fest, dass der Großteil der Echoinformationen von dem Polaroidsystem nicht genutzt wird. Lediglich das erste Echo, welches einen Schwellwert übersteigt, wird ausgewertet.

Das dritte Problem entsteht beim gleichzeitigen Einsatz mehrerer Sensoren. Der einzelne Sensor ist dann nicht in der Lage zu entscheiden, ob der ausgesandte Ultraschallimpuls von ihm selber oder von einem seiner Nachbarn stammt. Aus diesem Grund ist es unmöglich, alle Sensoren gleichzeitig zu betreiben. Um bei einer Mehrfachbestückung ordentliche Ergebnisse zu erhalten, bleibt nur die Möglichkeit, einen Sensor nach dem anderen anzusteuern und empfangen zu lassen. Dies

bedeutet, dass, bei einer Bestückung mit 24 Sensoren mit jeweils 10 Meter Reichweite, eine vollständige Messung mit allen Sensoren ca. 1,43 Sekunden dauert.

Demnach existieren nur 2 Möglichkeiten, die Messungen zu beschleunigen:

- Gleichzeitiges Arbeiten der Sensoren ermöglichen.
- Reduzierung der Anzahl der Sensoren.

Das gleichzeitige Senden mit mehreren Sensoren lässt sich entweder mit Pseudo-Random-Sequenzen oder durch Generierung einer jeweils senderspezifischen Anzahl von Schwingungen realisieren. Bei dem ersten Verfahren werden keine Ultraschallimpulse fester Frequenz, sondern zufällig frequenzveränderte Impulse generiert. Die Sensoren erzeugen also unterschiedliche Frequenzmuster, wodurch sie identifizierbar werden. Korreliert man das gesendete Signal mit dem empfangenen unter Zuhilfenahme der Autokorrelationsfunktion, ergeben sich nur die Echos, die von dem Sendesignal verursacht wurden, alle anderen werden ausgelöscht. Somit können alle Sensoren unabhängig voneinander arbeiten, ohne dass sie sich gegenseitig stören. Zusätzlich gewinnt das System an Entfernungsauflösung, da diese nicht mehr von der Länge des ausgesendeten Impulses abhängt (weiterführend siehe [Markus Berg 1998; Berg 1996]).

Um dies zu realisieren, ist allerdings ein großer Aufwand von Nöten, da für die Korrelation der Signale ein erhebliches Maß an Rechenleistung benötigt wird. Das Generieren und Empfangen der Pseudo-Random-Frequenzen setzt voraus, dass alle Sensoren über jeweils einen eigenen A/D- sowie D/A Umsetzer an einen zentralen DSP³ angeschlossen sind, der die Signale der einzelnen Sensoren in weniger als 1,43 Sekunden korrelieren kann, da sonst gegenüber der konventionellen Laufzeitmessung wenig gewonnen wäre. Ein weiterer Nachteil dieser Methode ist, dass sie stark von der Beschaffenheit der Oberflächen der zu detektierenden Objekte abhängt. Wird der ausgesandte Pseudo-Random-Impuls zu sehr von der ihn reflektierenden Oberfläche verfälscht, so ergibt sich nach der Korrelation überhaupt kein Echo mehr.

Das zweite Verfahren zum gleichzeitigen Senden mit mehreren Sensoren wird von Shoval und Borenstein in [Shraga Shoval 2001] beschrieben. Sie generierten Ultraschallpings fester Frequenz, aber mit einer spezifischen Anzahl Schwingungen. Die empfangenen Echos können so aufgrund ihrer Schwingungsanzahl unterschieden werden. Dieses Verfahren hat den Nachteil, dass bei 24 Sensoren entsprechend lange Ultraschallimpulse benötigt werden, damit diese auch sicher

³Digitaler Signal Prozessor – Ein auf Signalverarbeitung optimierter Prozessor

zu unterscheiden sind. Die Länge der Impulse wirkt sich aber negativ auf die Entfernungsauflösung hintereinander liegender Objekte aus.

Diese Diplomarbeit verfolgt das Ziel die Messungen durch Reduktion der Anzahl der Sensoren zu beschleunigen. Um mit weniger Sensoren die gleiche Abdeckung der Umgebung zu erzielen, müssen die Sensoren einen weiteren Öffnungswinkel haben, denn nur bei ähnlicher Abdeckung sind die Verfahren vergleichbar. Vergrößert man den Abstrahlungswinkel der Sensoren, verliert man in gleichem Maße an Richtungsauflösung. Die Richtungsauflösung ist sehr wichtig, weil man nicht nur wissen möchte, in welcher Entfernung ein Hindernis vorhanden ist, sondern auch, ob es sich direkt vor dem Roboter befindet oder links beziehungsweise rechts davon. Es muss also ein Weg gefunden werden, um die fehlenden Richtungsinformationen zu kompensieren.

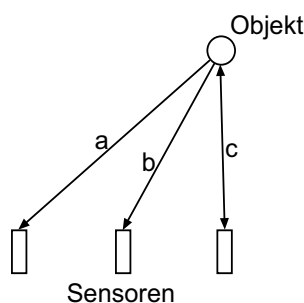


Abbildung 1.5: Laufzeitunterschiede beim dreier Sensorarray

Bei einem Array aus drei Sensoren mit großem Abstrahlungswinkel ist davon auszugehen, dass alle Sensoren in der Lage sind nicht nur die selbst ausgesandten Ultraschallpings zu empfangen, sondern auch die der anderen beiden Sensoren. Es besteht also nicht die Möglichkeit des gleichzeitigen Sendens.

Abbildung 1.5 zeigt, welcher Effekt ausnutzbar wird, wenn alle Sensoren alle Echos empfangen. Deutlich sind auf dem Bild die Laufzeitunterschiede (Länge der Strecke a, b oder c) zu erkennen, die durch ein nach rechts verschobenes Objekt entstehen. Da der Abstand der Sensoren zueinander bekannt ist sowie die Entfernung jedes Einzelnen zu dem Hindernis, kann man mittels Triangulation die Position des Objektes berechnen. Auf diesem Weg ließe sich die fehlende Richtungsauflösung kompensieren. Wenn die drei Sensoren alle einmal gesendet haben, haben sie jeweils drei Echoverläufe aufgezeichnet. Eine Messung ergibt demnach 9 Echokurven, welche die Umgebung vor dem Roboter abbilden. Mit Hilfe dieser hoch redundanten Daten, deren einziger Unterschied in der Position des Senders beziehungsweise Empfängers liegt, muss sich der Nachteil bei

der Richtungsauflösung vollständig kompensieren lassen. Gleichzeitig ist evident, dass die oben erwähnten Probleme, die beim Einsatz einer größeren Anzahl von Sensoren entstehen, durch Reduzierung der Sensoren eingedämmt werden kann.

1.2 Überblick über die Arbeit

In dieser Arbeit werde ich in Kapitel 2 auf die Auswahl eines geeigneten Microcontrollers eingehen und dabei die Vor- und Nachteile sowie die Technologie ausgewählter Systeme beschreiben. In Kapitel 3 wird die Software Entwicklungsumgebung vorgestellt, mit der die für dieses Projekt entwickelte Firmware programmiert wurde. Kapitel 4 schafft einen Überblick über eine Vielzahl der am Markt erhältlichen Ultraschallsysteme und diskutiert ihre Funktionsprinzipien. Das Kapitel 5 beschreibt die Entwicklung eines Interfaces, welches benötigt wurde, um das ausgewählte Prozessorsystem mit den Ultraschallgebern zu kombinieren. In Kapitel 6 werden mit dem entwickelten Sonar erste Messung gemacht, ausgewertet und diskutiert. Das Kapitel 7 fasst die Arbeit und Ergebnisse zusammen und eröffnet einen Ausblick für die Weiterentwicklung und Optimierung.

Kapitel 2

Auswahl eines Microcontrollers

Dieses Kapitel gibt einen Überblick über die Entscheidungskriterien und Möglichkeiten bei der Auswahl eines für dieses Projekt geeigneten Microcontrollers. Nach den grundlegenden Anforderungen an das System werden Architekturen, deren Vor- und Nachteile, sowie zwei Microcontroller vorgestellt.

Wichtigster Punkt bei der Auswahl eines Microcontrollers ist seine Leistungsfähigkeit. Dieses Projekt beschäftigt sich mit der Auswertung von Sonardaten zu Navigationszwecken, wodurch harte zeitliche Anforderungen an das Prozessorsystem gestellt werden. Vorgesehen sind bis zu acht Sonartransducer, mit denen durch Messung der Schalllaufzeiten die Umgebung erfasst werden soll. Die so gewonnenen Daten sollen mit Hilfe von Triangulation weiterverarbeitet werden. Triangulation erfordert den Umgang mit Fließkommazahlen, deren Einsatz sehr zeitaufwendig und rechenintensiv ist. Trotzdem sollte die Anzahl der vollständig ausgewerteten Messungen pro Sekunde möglichst hoch sein, um eine sichere Navigation zu gewährleisten. Aus diesem Grund scheidet die Gruppe der 8-Bit-Microcontroller von vornherein aus, nur ein 16- oder 32-Bit-System ist in der Lage diese Anforderungen zu erfüllen.

Ein weiterer Gesichtspunkt ist die Stromaufnahme. Da das Sonarmodul auf einem autonomen Roboter montiert werden soll, dessen einzige Stromversorgung ein mitgeführter Akku ist, sollte der Prozessor eine möglichst geringe Stromaufnahme aufweisen, um die Akkus nicht unnötig zu belasten.

Grundlegend für die Auswahl eines Controllers ist die Frage, welche Werkzeuge für die Entwicklung einer Applikation zur Verfügung stehen. Um eine Applikation in angemessener Zeit entwickeln zu können, werden verschiedene Werkzeuge (*Tools*) benötigt. Als erstes wäre hier eine geeignete möglichst komfortable Entwicklungsumgebung (*SDK, IDE*) zu nennen sowie ein Evaluation Board für den gewünschten Controllertyp. Außerdem wäre es zu begrüßen, auf ein möglichst breites Angebot vorgefertigter Bibliotheken und Funktionen zugreifen zu können.

Aus dem wirtschaftlichen Blickwinkel sind natürlich die Kosten pro Controller sowie dessen Verfügbarkeit am jetzigen als auch am zukünftigen Markt zu berücksichtigen.

2.1 Mögliche Architekturen

Im folgenden Abschnitt werden die beiden grundlegenden Prozessorarchitekturen RISC und CISC vorgestellt, um einen Überblick über Vor- und Nachteile beider Architekturen zu geben (vergleiche [Leitenberger]).

2.1.1 CISC

Bei dem **Complex Instruction Set Computer** kurz *CISC* handelt es sich, wie der Name schon sagt, um eine Prozessorarchitektur mit einem komplexen Befehlssatz. Bedingt durch die Evolution von Prozessoren, bei der es mit der Zeit möglich wurde, immer mehr Transistoren auf einem Chip zu integrieren, wurden auch immer mehr Befehle komplexerer Art implementiert, um die Programmierung zu erleichtern. Hierdurch wird die ganze Architektur sehr umfangreich, was ein Merkmal aller CISC-Prozessoren ist.

Die Befehle eines CISC-Prozessors sind je nach Häufigkeit ihrer Verwendung unterschiedlich lang und werden zur Ausführung in einen sogenannten Microcode übersetzt, der, je nach Komplexität des Befehls, aus unterschiedlich vielen Einzelschritten besteht (s. Abbildung 2.1). Dies erfordert einen hohen Zeitaufwand beim Dekodieren einer Instruktion und ist verantwortlich für unterschiedlich lange Ausführungszeiten der Befehle. Der Grundgedanke, der zu einer Integration hochkomplexer Befehle in die CPU führte, war dass der Code so eine höhere Dichte erreichen und die Ausführung schneller sein würde. Außerdem ließ sich so auch noch Platz im RAM sparen, was in Zeiten, in denen Computer nur über wenige Kilobytes RAM verfügten, durchaus sinnvoll war.

Ein konzeptioneller Nachteil der CISC-Architektur ist seine relativ kleine Anzahl an Registern, von denen einige üblicher Weise auch noch Spezialaufgaben haben. Somit muss ein Großteil der Operanden direkt aus dem, im Vergleich zu den Registern sehr langsamen, Hauptspeicher geladen werden, was für jeden Befehl erhebliche Verzögerungen nach sich zieht. Der Grund für die geringe Registeranzahl liegt in der Tatsache, dass bei den meisten Befehlen das Register, auf dem er wirkt, mit einkodiert ist, mit der Folge, dass sich mit steigender Registerzahl immer weniger Befehle implementieren lassen. Deshalb wurden auch immer wieder „Zwischenlösungen“ entwickelt, in denen einige Befehle nur auf bestimmten Registern arbeiten konnten, um so die Registeranzahl zu erhöhen.

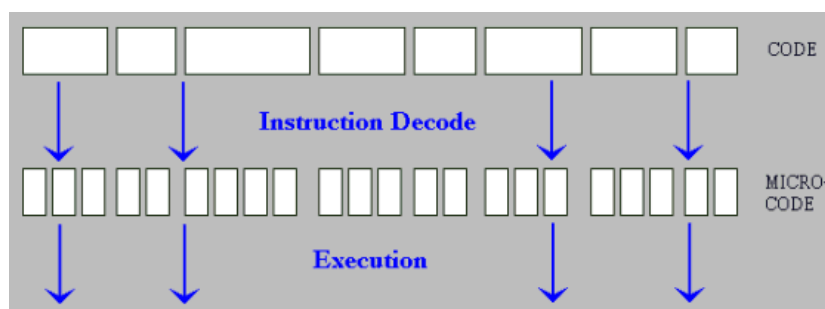


Abbildung 2.1: Instruction Decode eines CISC

2.1.2 RISC

Der **Reduced Instruction Set Computer** kurz *RISC* ist ein Prozessor mit reduziertem Befehlssatz, also das genaue Gegenteil von CISC. Grundlegend für die Entwicklung dieses Prozessors war die Idee, durch wenige einfache Befehle den Durchsatz an Instruktionen zu erhöhen. Die Erkenntnis, dass die meisten Hochsprachencompiler circa 80 Prozent des Hochsprachencodes mit 20 Prozent der Assemblerbefehle realisieren, bestätigte diesen Ansatz. Nach [Eberle] ermöglichen folgende Techniken die extrem hohe Durchsatzrate an Instruktionen bei heutigen RISC-Prozessoren.

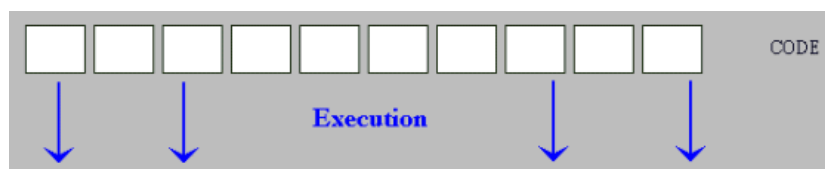


Abbildung 2.2: Instruction Decode eines RISC

- **Fixe Instruktionslänge**

Ein Instruktionsformat mit einer fixen Länge von typischer Weise 32 Bit erleichtert das Dekodieren der Instruktionen, da alle Befehle nach demselben Muster aufgebaut sind. Hierdurch ist es einfacher, einen Befehl pro Taktzyklus zur Ausführung zu bringen.

- **Wenige einfache Instruktionen**

RISC-Prozessoren kennen im Gegensatz zu CISC-Prozessoren nur einfache und wenige Instruktionen, wodurch man das Übersetzen in Microcode einsparen kann, was sowohl das Dekodieren des Befehls als auch seine Ausführung selber beschleunigt. Dies erleichtert den Aufbau einer Pipeline und ermöglicht höhere Taktraten. Dadurch, dass bei RISC-Prozessoren

der Arbeitsaufwand pro Instruktion etwa gleich groß ist, können innerhalb der Pipeline Befehle einfacher und effizienter überlappt werden. Weiterhin vereinfacht sich der Aufbau des Prozessors maßgeblich, was zu einer geringeren Stromaufnahme sowie geringeren Produktionskosten beiträgt (s. Abbildung 2.2).

- **Load Store Architektur**

Bei RISC-Prozessoren zerfällt der Befehlssatz in drei Kategorien. Arithmetische/logische Operationen, Speicherzugriffe und Sprungbefehle. Alle Operanden einer Operation müssen sich in Registern befinden. Speicherzugriffe sind den Instruktionen Load und Store vorbehalten, wodurch die Verarbeitung der Befehle von Speicherzugriffen entkoppelt und somit eine Optimierung der Pipeline ermöglicht wird.

- **Großer Registersatz**

Um die Anzahl der Load- und Storebefehle zu reduzieren, die benötigt werden um auf Operanden im Speicher zuzugreifen, wird eine möglichst große Anzahl von Registern zur Verfügung gestellt. Bei RISC-Prozessoren üblicherweise 32 für Integer- und, falls vorhanden, 32 für Floating-Point-Operationen.

2.2 Mögliche Controller-Familien

Dieser Abschnitt stellt jeweils eine Prozessorfamilie beider Architekturen vor. Bei dem Vertreter der CISC-Architektur handelt es sich um die C-166 Familie von Infineon und speziell um den C-167 als Vertreter dieser Familie. Der RISC Vertreter ist ein Atmel AT91M55800 basierend auf einem ARM 7TDMI-Kern, ein Derivat der ARM Prozessoren.

2.2.1 Infineon C-166-Familie

Die C-166-Familie wurde 1990 von Siemens (*heute: Infineon*) entwickelt. Es handelt sich bei dem Prozessor-Core um einen 16-Bit-Kern mit vielen RISC-Merkmalen, wie zum Beispiel einem reduzierten Befehlssatz und einer daraus resultierenden einfacheren Hardware-Struktur. Seit 1990 wurden viele Derivate entwickelt (C161, C167), sowohl von Infineon, als auch von ST Microelectronics. Sie unterscheiden sich hauptsächlich in der Größe des adressierbaren Speichers und in der auf den Chip integrierten Peripherie, wodurch Softwareportierungen sowie die Wiederverwendung von Code stark vereinfacht werden. Dieser Chip wird nun seit ca. 11 Jahren von 2 Herstellern produziert, so dass es eine Vielfalt an verschiedenen Compilern und Entwicklungsumgebungen gibt.

Infineon C167 Hierbei handelt es sich um einen 16-Bit-Prozessor, der weder ein reiner CISC noch ein reiner RISC ist. Vielmehr liegt eine Mischform aus beidem vor. Der Kern ist nach der Von-Neumann-Architektur aufgebaut, welche durch einen getrennten Daten- und Adressbus gekennzeichnet ist. Beide Busse werden sowohl für Daten, als auch für Instruktionen verwendet. Er unterstützt ein vierstufiges Pipelining, wodurch ein Grossteil der Befehle in einem Zyklus ausgeführt werden kann. Der Prozessor ist in der Lage, durch Segmentierung 16 MByte Speicher anzusprechen, der in 256 Segmenten zu 64 KByte adressiert wird. Der Kern selber besitzt 17 Spezialregister, die fast alle für Statusinformationen und die Verwaltung der Segmentierung benötigt werden. Die allgemeinen Register sind außerhalb des Kerns im RAM untergebracht und lassen sich über ein Basisregister plus Offset ansprechen. Er besitzt 55 Befehle, die 16 oder 32 Bit lang sind (vergleiche [Infineon Technologies]).

Das Beste am C-167 ist seine Peripherie. Ausgestattet mit einem seriellen Port und einem CAN-Bus-Interface lässt er sich problemlos in die meisten industriellen Umgebungen integrieren. Er besitzt 24 A/D-Umsetzer, hat neun Timer, 32 Capture-Compare-Register und vier Ports für Pulsweiten-Modulation (PWM). Diese Eigenschaften, wie auch die weiteren Peripheriemerkmale, machen diesen Prozessor fast zum idealen System für dieses Projekt.

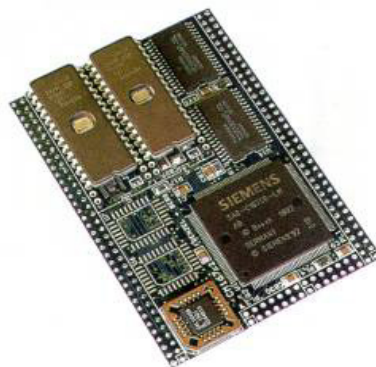


Abbildung 2.3: Infineon C167 mit Evaluation Board

2.2.2 Advanced Risc Machines – ARM-Familie

Bei den ARM-Prozessoren handelt es sich um einen 32-Bit-RISC-Prozessor mit sehr geringer Leistungsaufnahme. Er wurde 1985 von Acorn entwickelt und erstmals 1987 kommerziell realisiert mit dem ARM2 im Acorn Archimedes 305/310, welcher sich als Desktopcomputer aber nicht etablieren konnte. 1989 brachte Acorn etwa gleichzeitig zu Intels 486 den ARM3 heraus. Im selben Jahr trennte sich ARM als eigenständige Firma von seinem Gründungsunternehmen Acorn

und wurde zu einer „fabless“ Designschmiede, die nur noch IP (*Intellectual Property*) vermarktete [Stiller 2002]. In der zweiten Hälfte der neunziger Jahre schaffte es ARM, ihre Cores an fast alle Halbleiterhersteller zu lizenzieren, so dass dieser Prozessor heute in vielen Handys und PDAs eingesetzt wird.

Die Spezifikation von ARM Ltd. beschränkt sich allerdings auf den RISC-Core, was dazu geführt hat, dass es eine erhebliche Anzahl verschiedener Prozessor-Typen gibt. Alle unterscheiden sich nur in punkto integrierter Peripherie und Speicherausstattung voneinander, sofern sie auf der selben Core-Spezifikation basieren. Dies erleichtert Softwareportierungen und verbessert die Wiederverwendbarkeit von Code bei typ- und herstellerübergreifenden Entwicklungen. Selbst die verschiedenen Core-Typen, 7er Core und 9er Core, haben einen identischen Befehlssatz. Sie unterscheiden sich nur bezüglich ihrer Performanz, was eine Portierung auf leistungsfähigere Coretypen wesentlich vereinfacht.

Die stark zunehmende Verbreitung dieses RISC-Cores hat dazu geführt, dass für ihn eine äußerst diversifizierte Auswahl an Entwicklungsumgebungen und Evaluation Boards geschaffen wurde. Ein typischer Vertreter der ARM-Core-Familie sei hier vorgestellt.

ARM 7TDMI Core Der ARM 7TDMI Core ist ein reiner 32-Bit-RISC-Prozessorkern mit einer dreistufigen Pipeline. Auch hier werden fast alle Befehle innerhalb eines Zyklus' ausgeführt. Der Aufbau des Prozessors entspricht ebenfalls der Von-Neumann-Architektur. Er besitzt 31 Register, die für Daten oder Adressen genutzt werden können, sowie 6 Statusregister und ist in der Lage 16 MByte linear ohne Segmentierung zu adressieren. Er besitzt 34 Befehle a 32 Bit (ARM Mode) oder 36 Befehle a 16 Bit (THUMB Mode).

Die Besonderheit aller TDMI-Typen ist die doppelte Ausführung des Befehlssatzes. Es existieren der Standard-ARM-Befehlssatz mit einer Länge von 32 Bit und der THUMB-Befehlssatz mit einer Länge von 16 Bit. Der 32-Bit-Befehlssatz ermöglicht maximale Performanz zum Beispiel für Echtzeitanwendungen, der 16-Bit-Befehlssatz ermöglicht maximale Code-Dichte, wie sie in embedded Systems benötigt wird. Beide benutzen dieselben 32-Bit Register und sind in der Lage, 8-, 16- sowie 32-Bit-Datentypen zu verarbeiten. Der Prozessor ist in der Lage, während des Betriebes zwischen den beiden Befehlssätzen zu wechseln. So ist es möglich alle zeitkritischen Programmabläufe wie Interrupts oder DSP-Algorithmen vollständig in 32-Bit-Code zu implementieren, um so maximale Geschwindigkeit zu erzielen, während alle nichtkritischen Abläufe mit 16-Bit-Befehlen arbeiten, um die Code-Dichte zu erhöhen. Der für die Umschaltung benötigte Overhead wird vollständig während der Eintrittszeit in eine Subroutine abgearbeitet. Weiterhin ist der Prozessor in der Lage, sowohl das Little- als auch das Big-Endian-Format (siehe auch [Verts 1996]) zu verarbeiten.

Der ARM 7TDMI Core verfügt über ein integriertes Debug Interface basierend auf der IEEE ¹ Std. 1149.1-1990, *Standard Test Access Port and Boundary-Scan Architecture*. So ist es möglich, den Prozessor im normalen Betrieb anzuhalten und den Inhalt aller Register via JTAG-Interface auszulesen. Weitere Debugmöglichkeiten resultieren aus dem integrierten ICE-Braker-Interface. Bestehend aus zwei Echtzeit-Watchpoint-Einheiten ist es in der Lage den Prozessor zu stoppen, sobald einer der programmierten Werte gleich den Signalen auf dem Datenbus sowie dem Adressbus inklusive einiger Steuerleitungen ist. Alle Bits der Watchpointregister können individuell maskiert werden (siehe auch [Atmel Corporation 1999] und [Lindner 2002]).

Die Bezeichnung 7TDMI ist nicht nur ein Name, sondern beschreibt genau die Fähigkeiten des Cores. Die **7** bezeichnet die Generation des Kerns. Das **T** steht für THUMB-Mode, also Unterstützung des THUMB-Befehlssatzes. Das **D** und das **I** weisen auf die Debugfähigkeiten des Systems hin, wobei **D** für On-Chip Debug – und **I** für Integrated In Circuit Emulator (*ICE*) steht. Das **M** meint einen einfachen Multiplizierer, von dem es auch noch eine wesentlich verbesserte Version gibt.

Verfügbare ARM 7-Vertreter Microcontroller der ARM 7-Familie gibt es von fast jedem größeren Halbleiterhersteller, woraus eine sehr aufwendige Recherche nach geeigneten Typen resultierte. Die Forderung nach einem CAN Bus Interface erwies sich als außerordentlich restriktiv, so dass diese Eigenschaft gestrichen wurde. Die einzigen auffindbaren Typen mit CAN waren „noch nicht“ lieferbar. Sollte die CAN-Schnittstelle jedoch unabdingbar werden, ließe sie sich immer noch als externer Controller an den Prozessor anschließen.

Ein aufgrund seiner Verfügbarkeit sinnvoller Vertreter dieser Familie ist der Atmel AT9155800A. Er basiert auf einem ARM 7TDMI-Kern, der mit 33 MHz getaktet wird. Acht A/D- sowie zwei D/A Umsetzer mit einer Auflösung von 8 oder 10 Bit ermöglichen die Anbindung analoger Aktoren und Sensoren. Er hat sechs unabhängige Timer und lässt sich über einen seriellen Port mit der Außenwelt verbinden. Evaluation Boards sowie passende Software werden von Atmel bereitgestellt, so dass schnelle und unkomplizierte Vorbereitungen zum Einsatz dieses Microconrollers garantiert sind.

Obwohl der AT9155800A von der Peripherie nicht den Idealvorstellungen entspricht, ist er dennoch eine zukunftsweisende Plattform. Die Kompatibilität der ARM-Prozessoren würde es ermöglichen, später eventuell auf eine andere Hardware auszuweichen, die besser geeignet ist. Die Corestruktur ist einfacher und transparenter, vor allem aber gibt es keine Segmentierung bei der Adressierung des RAMs. Weiterhin ist der ARM-Prozessor auf Grund seiner 32-Bit-Struktur erheblich leistungsfähiger als der C-167. Diese Gründe, vor allem aber die Verfüg-

¹IEEE (Eye-triple-E), Institute of Electrical and Electronics Engineers, Inc. [IEEE 2002]

barkeit, sprachen generell für die ARM-Familie und speziell für den AT9155800A der Firma Atmel.



Abbildung 2.4: Atmel ARM AT91M55800A

2.3 Die Entscheidung

Die Entscheidung war ein Prozess, der sich über einen längeren Zeitraum hingog. Der Infineon C-167 galt, aufgrund seiner fast idealen Peripherie, lange als Favorit, obwohl es von Anfang an Bedenken wegen seiner Struktur sowie seiner Stromaufnahme gab. Besonders die Segmentierung des Speichers sei hier noch mal als deutlicher struktureller Nachteil genannt. Weiterhin lehnte die kooperierende Firma diesen Typ aufgrund seines Stromverbrauches und einiger anderer Erfahrungen mit dieser Familie ab, so dass die Vertreter der ARM-Familien ins Zentrum des Blickfeldes rückten.

Hier zeigte sich, dass die ARM Technologie noch nicht in alle Bereiche vorgezogen ist, da es unmöglich war einen ARM Controller mit integriertem CAN Bus und A/D-Umsetzern zu bekommen. Allerdings sind entsprechende Kombinationen bei mehreren Herstellern in Planung, so dass mit einer Verfügbarkeit dieser Typen ab Sommer 2002 zu rechnen ist. Auf Grund der Tatsache, dass eine Portierung etwaiger Software auf einen anderen ARM-Typ problemlos möglich ist, war also das Fehlen des integrierten CAN Busses kein ausreichender Grund eine weitere Prozessorfamilie in Betracht zu ziehen. Somit entschieden wir uns für den bereits vorgestellten AT9155800A von Atmel, da dieser am ehesten den Peripherieanforderungen dieses Projektes genügt, und sowohl eine gute Verfügbarkeit aufweist als auch im Rahmen einer denkbaren Erweiterung dieses Projekts eine wesentlich zukunftsorientiertere Plattform darstellt.

Kapitel 3

Software und Library

Dieses Kapitel stellt die Entwicklungsumgebung sowie die Software Library für den AT9155800 vor, mit dem dieses Projekt durchgeführt wurde. Es folgt eine Beschreibung der Umgebung und der Library, sowie eine Zusammenfassung.

3.1 ARM SDT - Software Development Toolkit

Das ARM SDT ist eine ältere Entwicklungsumgebung der Firma ARM. Sie besteht aus einem Projektmanager, diversen Compilern und einem Debugger.

Der Projektmanager dient hauptsächlich als Makefile-Ersatz. Mit ihm lassen sich sowohl die projektspezifischen Compileroptionen einstellen und verwalten als auch die Sourcefiles zuordnen, so dass man auf Knopfdruck ein auf den Prozessor herunterladbares Programm-Image erstellen kann. Möglichkeiten, um innerhalb eines Projektes ein Versionsmanagement zu betreiben, werden nicht unterstützt.

Die Funktion des internen Sourcefile-Editors ist sehr einfach, so dass es sich empfiehlt, einen externen Editor zu benutzen, um effektiv arbeiten zu können.

Ich habe eine so rudimentäre Entwicklungsumgebung gewählt, weil die Atmel-Library nur mit dieser problemlos zu kompilieren war (siehe Abschnitt 3.3).

Der Debugger des ARM SDT unterstützt 3 Debugmodi [ARM Limited a]:

- **Armulate Debug**

Der Debugger emuliert einen ARM-Prozessorkern ohne Peripherie. In diesem Mode lassen sich Algorithmen besonders effektiv debuggen, da der Emulator sehr schnell arbeitet und, wie für Emulatoren üblich, Unterbrechungen an jeder Stelle im Code möglich sind. Allerdings können alle Funktionen, die spezielle Hardware betreffen und Interruptserviceroutinen nicht ordentlich debugged werden, da hierfür die Ziel-Hardware benötigt wird.

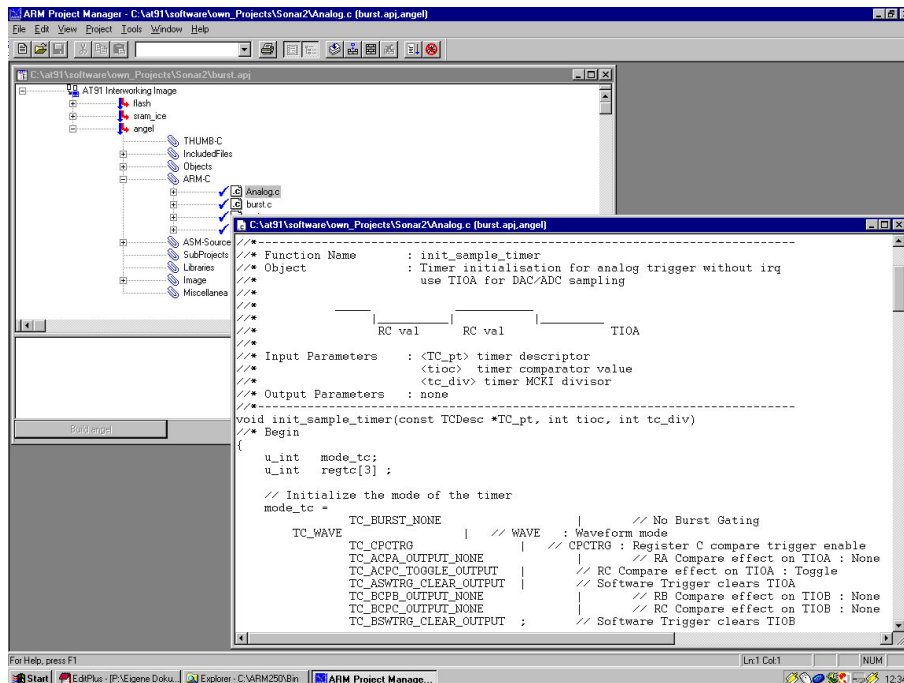


Abbildung 3.1: Projektmanager des ARM SDT mit geöffneten Editor

• Angel-Debug-Monitor

Der Angel-Debug-Monitor ist ein kleines Programm, welches auf dem Zielsystem läuft und so ein Debugging über die serielle Schnittstelle ermöglicht. Voraussetzung ist natürlich das Vorhandensein des Programms auf dem Zielsystem zum Bootzeitpunkt, da die Ziel-Hardware den Monitor booten muss, um auf Debug-Anfragen an der seriellen Schnittstelle reagieren zu können. Weiterhin teilt sich die Leistungsfähigkeit des Prozessors auf den Monitor und das zu debuggende Programm auf, so dass man in diesem Mode die Performance des Systems nicht abschätzen kann. Somit ist der Einsatz des Angel-Monitors nur auf einem Evaluation Board sinnvoll.

Mit Hilfe des Monitors lässt sich die Ziel-Hardware vollständig ansprechen und einsehen. Aber auch hier ist das Debugging von Interrupts schwierig, da das versehentliche Zerstören des Stacks auch den Angel-Debugger „abstürzen“ lässt, so dass es fast unmöglich ist, diese Stelle im Code mit dem Debugger zu finden.

Weiterhin ist das Ausführen von Programmen im Singlestep eine sehr zeitaufwendige Angelegenheit, da die Reaktionszeiten der Hardware bei dieser Art des Debugging sehr hoch sind.

- **ICE Debugging**

Besitzt man ein ICE (*In Circuit Emulator*, siehe Kapitel 2.2.2), so kann der Debugger die Ziel-Hardware ansprechen, ohne dass dort ein Programm vorhanden sein muss. Diese Art des Debugging ist am komfortabelsten und effektivsten. Auch hier lässt sich die Ziel-Hardware vollständig ansprechen und auslesen, Interrupts lassen sich einfach wie jedes andere Programm debuggen, da der Prozessor in jedem State eingefroren werden kann. So kann man die Software zu jeder Zeit unterbrechen und die Inhalte der Register sowie des Speichers auslesen, überprüfen und verändern.

Auch die Singlestep-Execution ist wesentlich effektiver, da keine zusätzliche Software auf dem Zielsystem läuft, die die Daten an den Debugger seriell überträgt, sondern ein Interface, das den Prozessor ohne seine eigene Beteiligung vollständig kontrolliert.

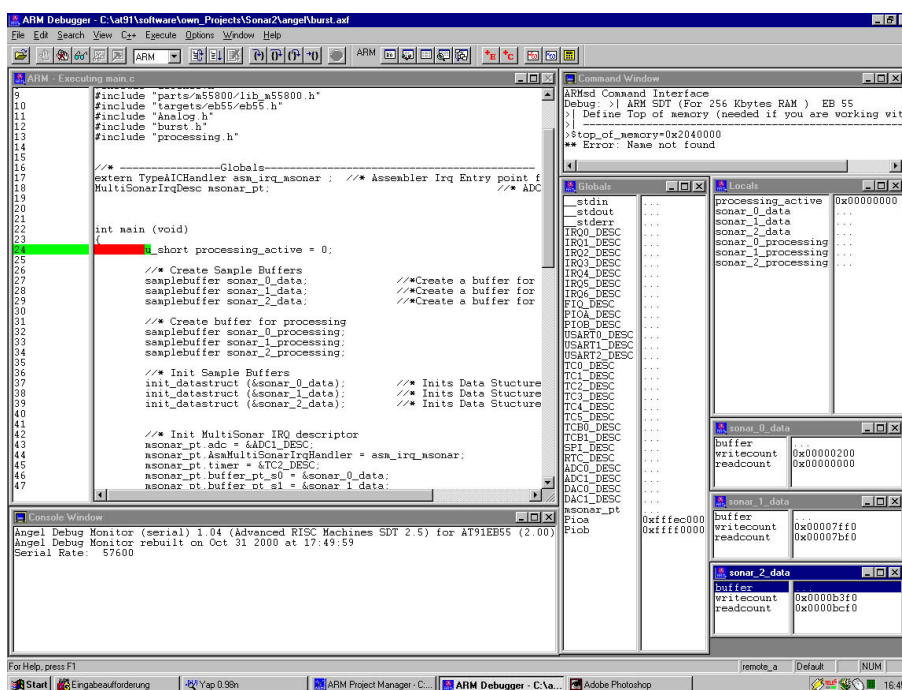


Abbildung 3.2: Debugger des ARM SDT

Eine einfache aber doch sehr effektive Methode des Debugging ist die Ausgabe von Variablen und Programmstatus via *printf*-Anweisung. Diese Möglichkeit wird von allen Debugmodi unterstützt (weiterführend siehe auch [ARM Limited b]). Hierbei ist jedoch zu berücksichtigen, dass ein *printf*-Befehl ein erhebliches Maß an Rechenleistung benötigt, weshalb diese Methode des Debuggings bei zeitkritischen Anwendungen nicht von Nutzen ist.

Ich habe während dieses Projektes die Software mit Hilfe des Angel-Debuggers kontrolliert, da ICE-Interfaces sehr teuer sind und der Angel-Debugger standardmäßig auf den Atmel Evaluation Boards vorhanden ist. Abbildung 3.2 zeigt den Debugger des SDT im Betrieb mit dem Angel-Monitor. Das große Fenster zeigt den Quellcode, darunter ist die Konsole mit den Ausgaben des Angel-Monitors zu sehen. Die Fenster auf der rechten Seite zeigen die globalen sowie die lokalen Variablen und einen Teil ihres Inhaltes.

3.2 Die Atmel Library

Atmel stellt für ihre Evaluation Boards eine Software Library zur Verfügung [Atmel Corporation]. Sie ist so aufgebaut, dass sie alle Evaluation Boards für ARM Prozessoren von Atmel unterstützt, sie muss also passend für das jeweilige Board kompiliert werden.

Folgende Softwarekomponenten sind in ihr enthalten:

- C-Headerfiles, die die komplette Hardware des AT91 Evaluation Boards beschreiben.
- Assembler-Includefiles, zum Ansprechen der Hardware in Assembler.
- Beispiele zum Ansprechen der Hardware mit und ohne Interrupt.
- C-Startupfiles zum Booten des Prozessors und Verzweigen in die „main“-Funktion.
- Projektbeispiele, die die verschiedenen Möglichkeiten des Debugging zeigen.
- Tools zum managen und programmieren der Hardware.

Die Library erleichtert den Einstieg in die Programmierung des Prozessors erheblich. Alle Register des Prozessors lassen sich mit Namen ansprechen, die Beispiele sind einfach und leicht für eigene Probleme adaptierbar. Es existiert eine Bootroutine, die direkt übernommen werden kann. Alle Quellcodefiles der Library sind dokumentiert, so dass eine Analyse problemlos möglich ist.

Weiterhin existieren für jeden Peripheriebaustein folgende Funktionen:

- `at91_<module>_open`
- `at91_<module>_close`
- `at91_<module>_get_status`
- `at91_<module>_trig_cmd`
- `at91_<module>_read`
- `at91_<module>_write`
- `at91_<module>_set_mode`

`<module>` steht für Peripheriemodule wie **adc** (*Analog Digital Converter*) oder **rtc** (*Real Time Clock*). Somit sind die Funktionen für alle Standardoperationen bereits vorhanden und brauchen „nur“ noch eingesetzt zu werden.

3.3 Fazit

Bei dem ARM SDT handelt es sich um eine einfache Entwicklungsumgebung, die zuverlässig und stabil läuft. Leider ist die Editorfunktion so gering, dass es ohne einen externen Editor kaum möglich ist, sinnvoll zu arbeiten. Die Nachfolgerin der ARM SDT, die ADS (*ARM Developer Suite*), ist wesentlich komfortabler, was das Editing von Quellcode betrifft¹. Jedoch ließ sich das C-Startupfile der Atmel-Library nur mit dem ARM SDT kompilieren, während alle anderen Compiler einen Fehler lieferten. Dies gab schließlich den Ausschlag für den Einsatz des ARM SDT als Entwicklungsumgebung.

Unter dem Aspekt des Debugging ist der Angel Monitor zwar eine sehr kostengünstige Möglichkeit, kann aber nur sinnvoll in Verbindung mit einem Evaluation Board genutzt werden. Um im Bereich der industriellen Anwendung effektiv arbeiten zu können, ist ein ICE- beziehungsweise JTAG-Interface unumgänglich. Hauptgrund dafür ist hier die Tatsache, dass sich ein „nacktes“ AT91M55800 Prozessorsystem, also ein System, auf dem noch keinerlei Software vorhanden ist, nur mittels eines ICE- bzw. JTAG-Interfaces oder eines vorher gefertigten ROMs/EPROMs programmieren lässt, da der Prozessor keinen Bootloader besitzt. Wird also aus irgendeinem Grund die Firmware gelöscht, so ist ein erneutes Installieren nur über ein externes Interface möglich.

¹ Auch Greenhills liefert eine Entwicklungsumgebung für ARM

Kapitel 4

Ultraschall

Wie schon in der Einleitung erwähnt, werden zur Roboternavigation meist mehrere Sensortypen eingesetzt, deren Daten zur Auswertung gegenseitig abgeglichen – fusioniert werden. Ultraschall ist eine (im Vergleich zu optischen Systemen) relativ kostengünstige Möglichkeit, Entfernungen im Bereich von 50 cm bis 15 m zu detektieren.

Um die Funktionsweise dieser Technologie zu verdeutlichen, werde ich erst einige physikalische Eigenschaften des Ultraschalls erörtern, um dann speziell auf einige Schallwandler einzugehen. Für umfassendere Informationen zum Thema Sensoren und Methoden zu deren Auswertung sei auf [Borenstein 1996] verwiesen.

Ultraschall ist eine hochfrequente (mind. 20 kHz) longitudinale Materiewelle. Das heißt, die Moleküle oder Atome der Materie schwingen in Richtung der Schallausbreitung. Die longitudinale Polarisierung der Schallwellen in Luft lässt sich darauf zurückführen, dass Gase zwar eine Volumenelastizität haben, aber keine Formelastizität. Festkörper hingegen sind auch formelastisch, so dass sich der Schall hier auch transversal fortpflanzen kann.

Prinzipiell ist der Ultraschall mit dem hörbaren Ton vergleichbar und breitet sich in Luft bis zu einer Frequenz von ca. 20 kHz auch genauso aus. Bei höheren Frequenzen reduziert sich seine Reichweite stark, und ab 1 MHz ist es so gut wie unmöglich den Schall durch die Luft zu übertragen. In Flüssigkeiten und festen Stoffen sind jedoch auch höhere Frequenzen problemlos möglich.

Die schnell schwingenden Moleküle oder Atome eines Mediums enthalten bei höheren Frequenzen schon große Mengen mechanischer Energie. Je höher die Frequenz ist, desto höher ist also die Energiedichte in einem vergleichbaren Volumen eines Stoffes. Solche Energiemengen können beachtliche Veränderungen in einem Medium hervorrufen. Wasser zum Beispiel zerstäubt an der Oberfläche, an der der Schall aus dem Wasser austritt, zu feinem Nebel (vergleiche [Chmela]).

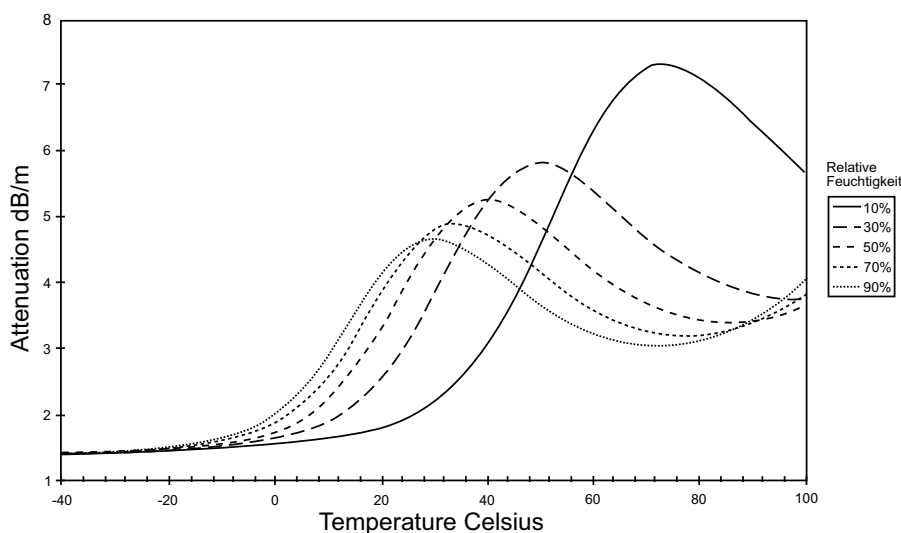


Abbildung 4.1: Signaldämpfung bei 100kHz

In der Medizinischen Sonographie werden derzeit Frequenzen im Bereich zwischen 2 und 20 MHz eingesetzt. Hierbei ist zu beobachten, dass sich mit steigender Frequenz zwar die Eindringtiefe in das Gewebe verringert, das räumliche Auflösungsvermögen aber aufgrund der steigenden Anzahl der Schwingungen pro Zeiteinheit verbessert wird (siehe [Medicine-Worldwide] und [Schröder 2000]).

Außer der Frequenz gibt es aber noch andere Faktoren, die die Laufzeit und Reichweite von Ultraschall in der Luft beeinflussen. Als wichtigste wären hier Temperatur und Luftfeuchtigkeit zu nennen. Temperaturunterschiede verändern die Schallgeschwindigkeit in der Luft sowie in festen Materialien, die Luftfeuchtigkeit verändert die Dämpfung des Schallsignals in der Luft. Wie der Abbildung 4.1 zu entnehmen ist, erfährt der Schall bei einer relativen Luftfeuchtigkeit von 70% und einer Raumtemperatur von 25°C schon auf 20 Meter Entfernung eine Dämpfung von 92 dB. Die Veränderung der Schallgeschwindigkeit durch Temperaturschwankungen in Luft lässt sich durch folgende Formel berechnen.

$$v_{T_{celsius}} = 331.5 * \sqrt[2]{1 + (0,003660992 * T_{celsius})} \quad (4.1)$$

Bei der ultraschallgestützten Navigation macht man sich die Tatsache zunutze, dass sich der Schall in der Luft mit $v_{T_{celsius}}$ fortpflanzt. Man sendet einen kurzen Ultraschallimpuls aus und misst die Zeit bis man ein Echo des ausgesandten Impulses empfängt. Aus der so gemessenen Laufzeit kann man die Entfernung des Objektes berechnen, welches das Echo verursacht hat. Da der Weg, den der Schall bei einer Messung zurücklegt, dem zweifachen der gemessenen Entfernung entspricht, wird man möglichst geringe Ultraschallfrequenzen einsetzen, um

die Reichweite in der Luft zu maximieren. Das durch die Frequenz vorgegebene räumliche Auflösungsvermögen liegt bei 40 kHz immer noch bei 8,45mm (Länge einer 40 kHz Schwingung in Luft), was für Navigationszwecke immer noch mehr als ausreichend ist.

Als Luftultraschallwandler (*Ultrasonic Transducer*) bezeichnet man alle „Lautsprecher“, die in der Lage sind Frequenzen im Ultraschallbereich, also ab 20 kHz aufwärts, in die Luft abzugeben. Sie lassen sich in zwei Hauptkategorien aufteilen:

- **Zweiteilige Systeme** bestehen aus einem Transmitter-Modul und einem Receiver-Modul. Beide Module können unabhängig von einander platziert werden, was vor dem Hintergrund, dass auch bei der Reflektion von Schall der Einfallswinkel gleich dem Ausfallswinkel ist, interessante Perspektiven eröffnet. Sie eignen sich besonders zur Realisierung von Sonarsystemen mit einem Messbereich von unter einem Meter, da das Nachschwingen (siehe Abschnitt 4.1) des Transmitters den Receiver nicht beeinflusst.
- **Einteilige Systeme** vereinigen die Eigenschaften eines Transmitters und eines Receivers in einem Transceiver. Der Transceiver ist also in der Lage ein Ultraschallsignal auszusenden und es auch wieder zu empfangen. Dies bietet den Vorteil, dass Sender und Empfänger exakt an derselben Stelle platziert sind, so dass Messfehler, bedingt durch den Abstand von Sender und Empfänger, nicht entstehen können.

Die Folgenden Abschnitte werden verschiedene Schallwandler vorstellen und deren Vor- und Nachteile im Detail aufzeigen.

4.1 Murata MA40R S



Abbildung 4.2: Murata Ultraschall Transmitter und Receiver

Hierbei handelt es sich um ein zweiteiliges System der Firma Murata (siehe Abbildung 4.2). Der Transmitter und der Receiver arbeiten beide nach dem Piezoelektrischen Prinzip [Dierkes]. Bei dem Transmitter wird monomorphe Piezokeramik durch elektrischen Strom zum Schwingen angeregt. Diese Schwingungen werden über eine konusförmige Membran an die Luft abgegeben. Der Receiver arbeitet entsprechend, mit dem Unterschied, dass hier die Schallschwingungen in der Luft ein ebenfalls mit einer Membran besetztes Stück Piezokeramik verformen. Diese Verformung veranlasst den Kristall, eine Spannung proportional zu dem Druck, der auf den Kristall wirkt, auszugeben.

Der entscheidende Vorteil dieses Systems besteht in seiner Aufteilung in Sender und Empfänger. Bei der Navigation mit Ultraschall werden kurze Ultraschallimpulse (*Bursts*) erzeugt und deren Laufzeit gemessen. Der Piezokristall hört mit Ende des Bursts allerdings nicht sofort auf zu schwingen, sondern schwingt noch eine Zeit lang nach. Dieses Nachschwingen, auch „*Ringing*“ genannt, erzeugt zwar kaum noch Schalldruck, ist aber an den elektrischen Anschlüssen deutlich zu messen. Will man den Transmitter auch als Receiver betreiben, wie bei einem Transceiver, dann muss man diese Nachschwingzeit (*Ringingtime*) erst abwarten bevor man ein Echo empfangen kann. Bei Messungen von Entfernungen unter 50 cm treten nun aber sehr kurze Schalllaufzeiten auf, die zum Teil vollständig durch das Nachschwingen überdeckt werden. Ein zweiteiliges System ist hiervon nicht betroffen, da ein anderer Piezo für den Empfang des Signals verantwortlich ist. Dieser muss nur im Moment des Bursts wegen des Übersprechens abgeschaltet werden.

Bei 40 kHz dauert ein Burst von 10 Schwingungen $250 \mu\text{s}$ und der Schall benötigt für eine Entfernung von 10 cm $300 \mu\text{s}$. Somit ist es möglich Objekte in Entfernungen ab 5 cm mit diesem System zu detektieren.

Beim Einsatz dieser Systeme muss der Abstand, den Sender und Empfänger zu einander haben, genau berücksichtigt werden, da der Schall sich diagonal durch das von Sender und Empfänger beschriebene Rechteck bewegt. (s. Abbildung 4.4) Wie aus Abbildung 4.3 zu entnehmen ist, ist die Richtcharakteristik des Transmitters kugelförmig und rotationssymmetrisch [Murata].

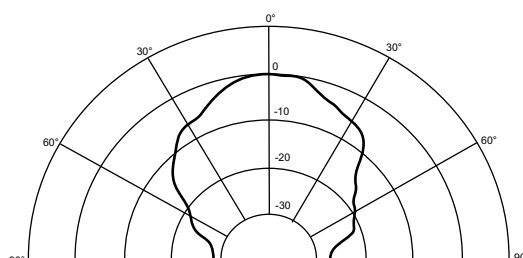


Abbildung 4.3: Richtcharakteristik Murata MA40R S

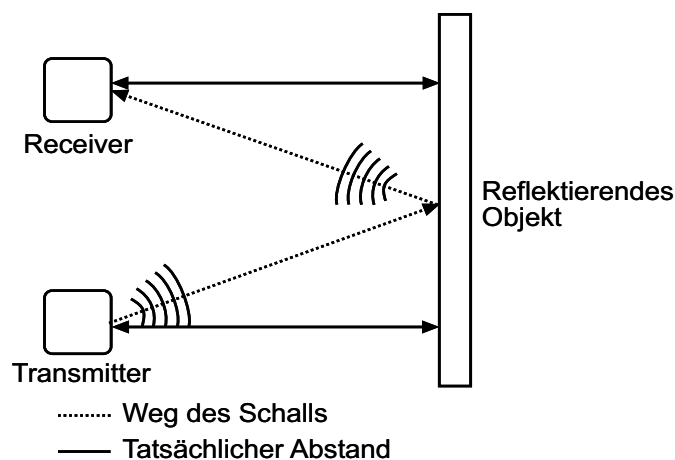


Abbildung 4.4: Laufzeitverlängerung durch Transmitter - Receiverabstand

4.2 Polaroid



Abbildung 4.5: Polaroid Transducer Serie 600

Ultraschallsysteme der Firma Polaroid (s. Abbildung 4.5) gehören zu den bekanntesten und am häufigsten verwendeten in der Robotikwelt. Sie wurden ursprünglich entwickelt für die Realisierung des Autofokus in Polaroid-Sofortbildkameras. Bei dem Transducer handelt es sich um ein einteiliges System, einen sogenannten Transceiver. Er unterscheidet sich stark von den anderen Wandlern, da er als einziger nach dem elektrostatischen Prinzip arbeitet. Die Funktionsweise beruht auf der Tatsache, dass sich ungleichnamige Ladungen anziehen und gleichnamige abstoßen. Somit ist es möglich zwei (wie auch immer) geladene Platten durch Veränderung ihres Ladungszustandes relativ zueinander zu bewegen. Der Schallwandler arbeitet genauso. Vor einer massiven Metallplatte ist mit geringem Abstand eine

sehr dünne Goldfolie gespannt. Die Metallplatte und die Goldfolie sind elektrisch voneinander isoliert. Legt man nun eine relativ hohe Spannung (400 V) an die Membran und die Metallplatte an, bewegt sich die Folie aufgrund der elektrostatischen Anziehungskraft auf die Metallplatte zu, indem sie sich ihr entgegenwölbt. Durch Anlegen einer Frequenz lässt sich die Goldfolie zum Schwingen anregen. Problematisch ist die Erzeugung der Spannung von 400 V, die benötigt werden um die Folie zum Schwingen anzuregen. Polaroid hat aus diesem Grund eine einfache Vorschaltplatine entwickelt, die in der Lage ist diese Spannung zu erzeugen und das Echo zu empfangen. Die Schaltung liefert am Ausgang sogar schon digitale Pegel. Man muss also „nur noch“ die Zeit messen, die zwischen dem Aussenden und dem Empfang des Signals vergangen ist, da ein eindeutiger Stopimpuls generiert wird.

Die Richtcharakteristik dieses Systems ist keulenförmig und, wie auch schon bei Murata, rotationssymmetrisch (s. Abbildung 4.6).

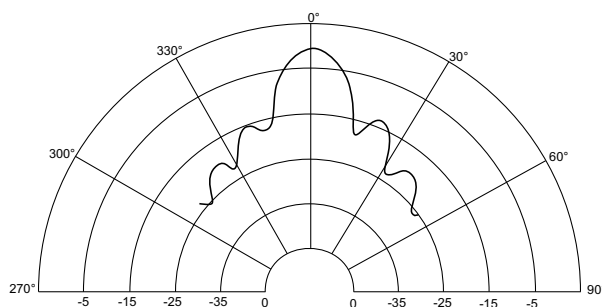


Abbildung 4.6: Richtcharakteristik Polaroid Transducer

4.3 Airmar AT Serie

Die Luftultraschallwandler der Firma Airmar, genannt *AirducerTM* (s. Abbildung 4.7) sind ebenfalls Transceiver, also einteilige Systeme. Auch hier macht man sich, wie schon bei Murata, den piezoelektrischen Effekt zunutze. Es handelt sich hierbei allerdings nicht um einen mit einem Konus bestückten kleinen Kristall, sondern um eine Schicht aus *PiezoflexTM*, die direkt auf einen runden Epoxyträger (im Bild weiss) geklebt ist. Dieses System wird in ein zylindrisches Gehäuse eingesetzt, so dass der Epoxyträger den Zylinder einseitig bündig verschließt. Wird diese Schicht zum Schwingen angeregt, so werden diese Schwingungen direkt auf den Epoxyträger übertragen und in die Luft abgestrahlt.

PiezoflexTM ist ein piezoelektrisches Polymer, basierend auf einem Homopolymer PVDF (Polyvinylidene Fluoride). Das Interessante an diesem Polymer ist, dass sich daraus sehr dünne, flexible Platten von 0,5 mm Stärke herstellen lassen,



Abbildung 4.7: AirMar Airducer™

denen man fast jede denkbare Form geben kann. Im Gegensatz zu keramischen Transducern werden *Piezoflex™* Transceiver nicht in Resonanz betrieben, wodurch eine relativ hohe Bandbreite in der Arbeitsfrequenz möglich ist. Weiterhin lässt sich *Piezoflex™* sehr günstig produzieren, was sich deutlich im Preis der *Airducer™* niederschlägt.

Die Größe des piezoelektrischen Elements ermöglicht sehr hohe Schalldrücke sowie eine hohe Empfindlichkeit für den Empfang des Echos. So sind mit einem *Airducer™* von 57 mm Durchmesser Reichweiten von bis zu 15 Meter möglich. Die Richtcharakteristik (*Beam pattern*) ist keulenförmig und rotationssymmetrisch mit einem typischen Öffnungswinkel von 12° bis 15° je nach Typ (s. Abbildung 4.8).

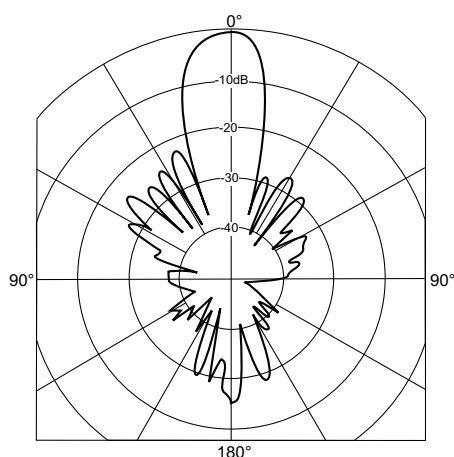


Abbildung 4.8: Richtcharakteristik Airmar AT50

Leider hat dieser Aufbau nicht nur Vorteile, da die Spannung, die benötigt wird um das Polymer zu verformen, mit dessen Größe ebenfalls zunimmt. Die Spannungen im Burstbetrieb liegen je nach Größe des Transducers zwischen 800 Vpp und 1500 Vpp. Es muss also ein erheblicher Aufwand getrieben werden, um eine solche Spannung zu erzeugen (siehe [Airmar Technology Corporation b] und [Airmar Technology Corporation a]).

4.4 Marco



Abbildung 4.9: Transducer der Firma Marco

Bei den Transducern der Firma Marco (s. Abbildung 4.9) handelt es sich wieder um Transceiver. Auch diese arbeiten nach dem piezoelektrischen Prinzip. Der hier zum Einsatz kommende „weiche“ piezoelektrische Werkstoff FPM 220 zeichnet sich durch seine niedrige relative Dielektrizitätskonstante und eine hohe piezoelektrische Druckkonstante aus.

Bei dem abgebildeten Ultraschallwandler handelt es sich um einen sogenannten Modulartransducer. Ein besonderes Merkmal der Modulartransducer ist ihre nicht rotationssymmetrische Richtcharakteristik.

Der Aufbau ist recht einfach. Drei keramische, piezoelektrisch aktive Streifen sind rechtwinklig auf der Rückseite eines balkenförmigen Keramikträgers befestigt, auf dessen Vorderseite zum Schutz ein rechteckiges Stück *Kapton*[®] Folie geklebt ist. Alle drei piezokeramischen Elemente werden gleichphasig und synchron zum Schwingen angeregt, die Schwingungen werden in den Keramikträger übertragen und über die *Kapton*[®] Folie abgestrahlt. Die Arbeitsfrequenz dieser Systeme liegt genau auf ihrer Resonanzfrequenz, wodurch eine sehr hohe Wandlereffizienz beim Abstrahlen sowie eine hohe Empfindlichkeit beim Empfang von Ultraschall erreicht wird. Dies geht allerdings zu Lasten der Bandbreite der Transducer.

Auch hier werden, wie auch schon bei Airmar, sehr hohe Spannungen von ca. 500 Vpp benötigt, um einen Ultraschallimpuls von angemessener Lautstärke zu erzeugen. Die Firma Marco bietet, ähnlich wie Polaroid, für ihre Transducer eine Vorschalt elektronik an, die die Erzeugung der Hochspannung sowie eine Vorverstärkung des Echosignals übernimmt. So ist es möglich ein relativ kleines Modul aus Transducer und Elektronik zu fertigen, welches sich mit 11 bis 24 Volt betreiben lässt.

Hervorzuheben bei diesem Transducer ist seine asymmetrische Schallkeule. Wie aus der Abbildung 4.10 hervorgeht ist das Schallfeld fächerförmig, also vertikal stark gebündelt, während es horizontal einen sehr weiten Winkel aufweist. Dies ist besonders für Anwendungsszenarien geeignet, die es erfordern mit wenigen Transducern eine große horizontale Fläche lückenlos abzudecken. Die durch den weiten horizontalen Öffnungswinkel verlorenen Richtungsinformationen lassen sich bei günstiger Anordnung der Modulartransducer mathematisch sehr genau durch Triangulation ermitteln. Somit sind diese Transducer, sofern mehrere eingesetzt werden, hervorragend für Navigationszwecke geeignet.

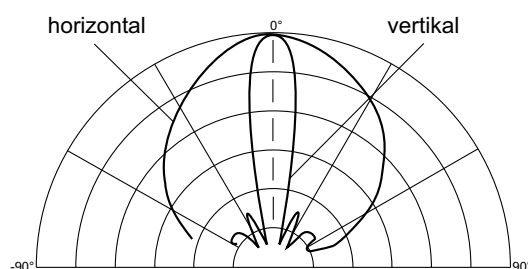


Abbildung 4.10: Richtcharakteristik der Modulartransducer

4.5 Zusammenfassung

Die oben aufgeführten Ultraschallwandler sind nur ein kleiner Auszug aus dem derzeitigen Marktangebot. Allerdings repräsentieren sie einen Großteil der möglichen Bauformen und Wirkungsmechanismen. Jeder der Sensoren hat seine Vor- und Nachteile sowie ein Anwendungsfeld, in dem seine Vorteile zur Geltung kommen.

Für Navigationszwecke eignen sich besonders die Transducer der Firma Marco. Aufgrund ihrer fächerförmigen Richtcharakteristik bieten sie ein sehr weites, lückenloses Beobachtungsfeld. Durch den Einsatz mehrerer Sensoren lassen sich große Bereiche abdecken, in denen durch Triangulation der Sensordaten einzelne Objekte genau geortet werden können.

Aus diesen Fakten und Überlegungen resultiert meine Entscheidung für den Einsatz von drei Modulartransducern von Marco zusammen mit der Ansteuerungselektronik.

Kapitel 5

Die Hardware

Das nun folgende Kapitel befasst sich mit der Kopplung der in den vorangegangenen Kapiteln selektierten Komponenten: Ultraschall Transducer und Prozessor. Ziel des Hardwaredesigns ist es, mindestens drei Transducer an den Prozessor anzuschließen (s. Abbildung 5.1), so dass sie sowohl einzeln als auch in Kombination genutzt werden können. Als Anforderung ergibt sich hieraus, dass die Echosignale der Transducer separat wie auch gleichzeitig einlesbar sein müssen. Das Senden eines Ultraschallbursts hingegen muss nur bei einem Transducer zur Zeit möglich sein, da ein gleichzeitiges Senden ohnehin unerwünscht ist, denn es lässt sich beim Empfangen der Echos nicht entscheiden, welcher Transducer den Burst gesendet hat. Es gibt zwar Verfahren, die ein paralleles Senden mehrerer Ultraschallgeber erlauben (Impulskompression, Pseudo-Random Frequenzen). Diese erfordern allerdings einen erheblichen Aufwand beim Samplen der Daten sowie bei deren Verarbeitung, so daß in dieser Arbeit vom Einsatz dieser Techniken abgesehen wurde. Weitere Informationen zu diesen Techniken finden sich in [Borenstein 1996],[Markus Berg 1998] und [Berg 1996].

Im Folgenden werde ich das Evaluation Board für den Atmel AT91M55800A sowie die Sonarmodule von Marco genau vorstellen, um dann über die Spezifikation zur Realisierung eines Interfaces zu kommen.

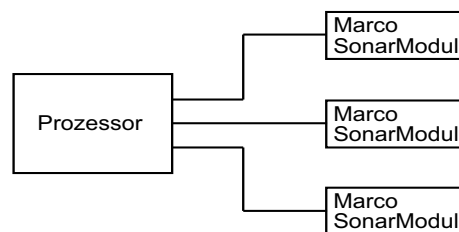


Abbildung 5.1: Prinzipieller Aufbau des Prozessors mit drei Transducern

5.1 Evaluation Board AT91EB55

Das Evaluation Board hat ungefähr die Größe einer Europlatine und besteht neben dem Prozessor aus einer Vielzahl von Peripherie und Speichertypen. Zu der bereits vorhandenen Peripherie zählen:

- Zwei serielle 9-polige Sub-D Anschlüsse,
- ein Reset- sowie 4 frei programmierbare Taster,
- acht LEDs, ebenfalls frei programmierbar,
- einen 2*32-Pin-Steckplatz für Speichererweiterung,
- einen 3*32-Pin-Steckplatz, auf dem alle I/O Pins des Prozessors zu finden sind,
- ein 20 poliger JTAG Anschluss.

Die Speicherbestückung des Boards besteht aus:

- 256 KByte SRAM 16 Bit breit (erweiterbar auf 1 MByte),
- 2 MByte Flash-RAM 16 Bit breit (1 MByte frei programmierbar),
- 4 Mbyte Serial Data Flash (erweiterbar auf 16 MByte),
- 64 KByte E^2PROM mit I^2C -Zugriff,
- 32 KByte SPI E^2PROM .

Das Board besitzt ein eigenes Netzteil und kann mit einer Spannung zwischen sieben und zwölf Volt betrieben werden. Der Prozessor und die meisten Peripheriebausteine werden von diesem Netzteil mit einer Spannung von 3,3 Volt versorgt. Somit betragen auch die Logikpegel, die an dem I/O Anschluss des Boards zur Verfügung gestellt werden, null und drei Volt.

In dem unteren Megabyte des Flash-RAM befindet sich die Firmware, bestehend aus einem Bootloader, einer Selbsttestroutine sowie einem SRAM Downloader und einem Angel-Debug-Monitor. Der Prozessor selber besitzt keinen Bootloader, so dass ein Update der Firmware nur über JTAG oder ICE Interface möglich ist. Der Bootvorgang lässt sich durch einen Jumper beeinflussen, so dass man entweder die Firmware aus dem unteren Megabyte oder ein selbstgeschriebenes Programm aus dem oberen Megabyte des Flashes booten kann. Die Firmware ist nicht geschützt, so dass beim flashen des Speichers darauf zu achten ist, dass nicht die Firmware überschrieben wird.

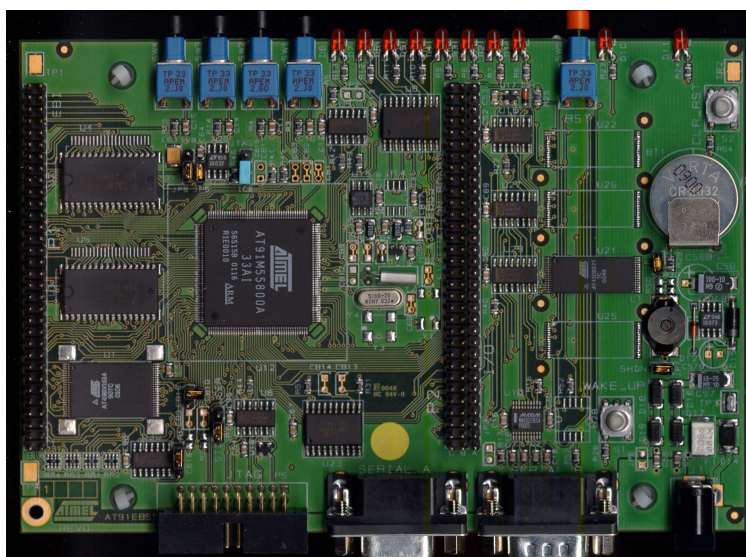


Abbildung 5.2: Evaluation Board AT91EB55

Der Angel-Debug-Monitor ermöglicht ein vollständiges Debugging, wie es sonst nur mit Zusatzhardware (JTAG Interface, ICE Interface) realisierbar ist. So ist man nicht nur in der Lage, das Board über die serielle Schnittstelle zu programmieren sondern auch vollständig zu debuggen. Es lassen sich Programme im Singlestep ausführen und alle Prozessorregister und Variablen auslesen.

Die Analog/Digital- sowie die Digital/Analog-Umsetzer des Prozessors sind an eine Referenzspannung von 2,5 Volt angeschlossen, so dass die Maximalpegel beider Umsetzer durch diese 2,5 Volt begrenzt werden. Da der Prozessor eine Real Time Clock besitzt, ist das Evaluation Board mit einer Knopfzelle zu deren Versorgung ausgestattet.

5.2 Marco Ultraschall Sende- Empfangssystem

Dieses Sende-Empfangssystem (*Typ ps/ms/w/a*) besteht aus einem Modulartransducer (*Typ ps/mt/5x30/d*) und einer Ansterelektronik (*Typ ase/se/w/b*). Der Modulartransducer hat eine Arbeitsfrequenz von 45,5 kHz bei einer Bandbreite von ± 2 kHz (max. 6 dB Dämpfung). Die Richtcharakteristik ist fächerförmig mit einem horizontalen Abstrahlwinkel von $\pm 40^\circ$ und einem vertikalen Abstrahlwinkel von $\pm 8^\circ$.

Die Ansterelektronik ist komplett in SMD Technik aufgebaut und kann aufgrund der kleinen Bauform direkt an die Transducer angelötet werden (s. Abbildung 5.3). Die Schaltung besteht aus einer Hochspannungsendstufe zur Erzeu-

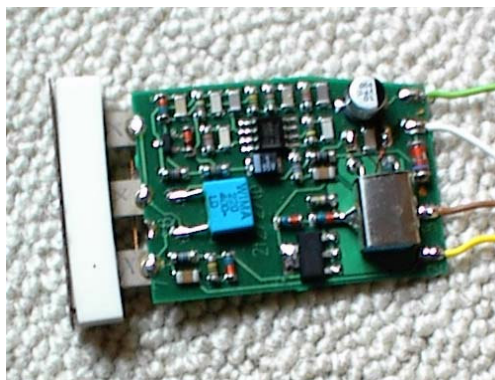


Abbildung 5.3: Marco Ultraschall Sende- Empfangssystem

gung der Hochspannung beim Senden eines Bursts sowie aus einer einfachen Operationsverstärkerschaltung zur Verstärkung des Echos. Der Sendeburst muss extern generiert werden, da die Schaltung nur eine Endstufe besitzt nicht aber einen Signalerzeuger.

Die Betriebsspannung lässt sich im Bereich von 11 bis 25 Volt frei wählen, während der Burstimpuls selber zwischen 0 V und Betriebsspannung schwingen muss, um den fehlerfreien Betrieb zu gewährleisten. Bei der Wahl der Betriebsspannung ist zu berücksichtigen, dass die Reichweite der Transducer sich mit Erhöhung der Betriebsspannung verbessert, da der abgestrahlte Impuls schlicht lauter wird.

Bei der Erzeugung des Bursts sind 2 Parameter genau einzuhalten:

- **Burstanzahl**

Die Anzahl der Schwingungen pro Burst darf 20 nicht überschreiten.

- **Burstfolgefrequenz**

Es dürfen nicht mehr als 50 Bursts pro Sekunde generiert werden.

Werden diese Parameter nicht eingehalten, wird der Schalttransistor der Hochspannungsendstufe zerstört, da dieser nicht in der Lage ist, die dann in ihm umgesetzte, erheblich höhere Energie abzuführen.

Während der Hardware Tests (siehe Abschnitt 5.5) ergaben sich zudem Schwierigkeiten, die eine besonders vorsichtige Montage des Transducers geboten scheinen ließen.

- Der Transducer ist sehr zerbrechlich, da der Keramikträger ein sehr spröder Werkstoff ist.
- Beim Anschließen der drei piezoelektrisch aktiven Streifen ist sehr darauf zu achten, dass die elektrischen Leitungen zwischen den Piezostreifen beim

Burst nicht zum Schwingen angeregt werden, da sich dies wieder direkt auf die Piezostreifen überträgt und so die Ringingtime um ein Vielfaches erhöht (von 2 auf 10 ms). Auch die von Marco selber vorgeschlagene Montage direkt an dem Print verdoppelt die Ringingtime.

- Die Kontaktflächen der Piezostreifen sind extrem empfindlich. Auch hier muss bei Arbeiten mit Vorsicht vorgegangen werden.

Die Abbildung 5.4 zeigt die Anschlussart der Transducer, bei der experimentell die geringste Ringingtime erzielt wurde.



Abbildung 5.4: Anschluss des Transducers

5.3 Das Interface

Aus den vorangegangenen Abschnitten lässt sich erkennen, dass das Evaluation Board und das Sende-Empfangssystem der Firma Marco nicht ohne weiteres kompatibel sind. Um beide aufeinander abzustimmen muss eine Anpassung der Spannungen von Prozessor und Sonar-Ansteuerungselektronik erfolgen. Aufgrund der Abhängigkeit zwischen Betriebsspannung und Reichweite des Transducers, sollte dieser mit seiner maximalen Spannung von 24 Volt betrieben werden. Das bedeutet, dass der Burst entsprechend zwischen 0 und 24 Volt schwingen muss. Da die komplette Signalverarbeitung mit dem Prozessor realisiert wird, soll auch der Burst generiert werden. Aufgrund der Tatsache, dass der Prozessor nur 3,3 Volt zur Verfügung stellt, ist hier eine Anpassung des Signalpegels nötig.

Bei dem Burst selber handelt es sich um 20 Rechteckschwingungen bei 45 kHz, gefolgt von einer Signlpause. Ein Signal dieser Art lässt sich am einfachsten

mit Hilfe zweier Timer realisieren: Einer dient zur Erzeugung der 45 kHz und ein zweiter zum Ein- und Ausschalten des Bursts. Wenn man auf diese Art drei Transducer unabhängig von einander betreiben will, braucht man sechs Timer. Der Prozessor selbst hat genau sechs Timer, allerdings werden diese auch noch für andere Zwecke (Sampling) benötigt, so dass ein Multiplexen des Burstsignals unumgänglich ist.

Ein weiteres Problem entsteht beim synchronen Samplen der Echos. Nach dem Nyquist-Abtasttheorem [Tanenbaum 1998] muss die Samplingfrequenz immer doppelt so hoch sein wie die höchste abzubildende Frequenz. Somit müsste zur Abbildung einer 45 kHz-Schwingung mit 90 kHz abgetastet werden, was bei drei Transducern zu einer erheblichen Datenmenge führt. Ein interessanter Ansatz ist in diesem Zusammenhang, dass für das angestrebte Messverfahren nicht die 45 kHz des Echos relevant sind, sondern nur dessen Hüllkurve. Mit Hilfe einer einfachen Gleichrichterschaltung lässt sich das Echo in eine gleichgerichtete geglättete Hüllkurve umwandeln, so dass die Samplingfrequenz erheblich gesenkt werden kann.

Aufgrund der Tatsache, dass für den Betrieb der Transducer 11 bis 24 Volt und für den Betrieb des Prozessors 6 bis 12 Volt benötigt werden, schien es sinnvoll ein einfaches Netzteil mit auf der Interface-Platine zu integrieren.

5.3.1 Spannungsanpassung und Multiplexing

Als Multiplexer kommt ein 4051 8-fach MUX/DMUX zum Einsatz. Wie die Bauteilbezeichnung 40XX vermuten lässt, handelt es sich um CMOS Technologie, welche aufgrund ihrer hohen Flexibilität in Bezug auf die Betriebsspannung gewählt wurde. CMOS Technologie ist in Abhängigkeit von der Betriebsspannung in der Lage, Spannungen von 4,5 bis 16 Volt zu verarbeiten. Da die Transducer mit ihrer maximalen Spannung von 24 Volt betrieben werden sollen, um ihre Impulsleistung zu maximieren, erfolgte die Pegelanpassung zweistufig.

Wie der Abbildung 5.5 zu entnehmen ist, werden in der ersten Stufe alle Signale, die zum Multiplexen benötigt werden, mit Hilfe eines Quadruple Voltage Translators vom Typ 4104 von 3,3 auf 10 Volt angehoben, um gute Spannungspegel für die Weiterverarbeitung zu gewährleisten. Es handelt sich hierbei um drei Signale zur Steuerung des Multiplexers sowie um das Burstsignal selber, welches aber zusätzlich noch invertiert wird. Nach dem Multiplexen steuert das Burstsignal einen CMOS-Transistor an, der aufgrund seiner Beschaltung (Sourceschaltung) wieder invertierend wirkt und das Burstsignal auf 24 Volt anhebt. Somit steht das Burstsignal wieder phasenrichtig mit einer Spannung von 24 Vpp am Sourcepin des Transistors an.

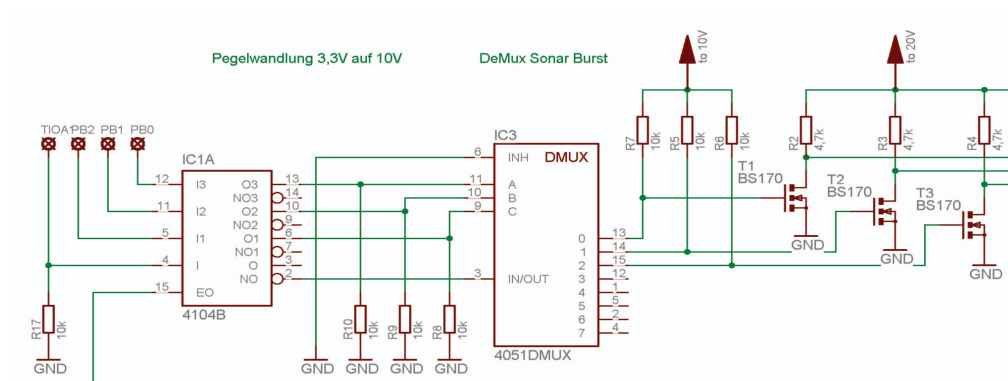


Abbildung 5.5: Ausschnitt aus dem Schaltplan: Pegelanpassung

5.3.2 Gleichrichtung

Das Echo, das von jedem Transducer empfangen wird, wird mit Hilfe eines einfachen Einweggleichrichters gerichtet, und mit einem Kondensator geglättet. Die Stärke der Glättung muss während der Tests genau überprüft werden, da eine zu starke wie auch eine zu geringe Glättung die Signale vollständig verfälschen kann. Für den Gleichrichter sollten Germanium- oder Lower-Shottky-Dioden zum Einsatz kommen, weil deren Schwellspannung wesentlich geringer als bei Silizium ist, wodurch sich der Spannungsabfall am Gleichrichter reduzieren lässt.

Die Ausgangsspannung der Transducer liegt nach der Gleichrichtung und Glättung zwischen 0 und 10 Volt. Der AD-Umsetzer des Prozessors ist aber nur in der Lage Spannungen im Bereich von 0 bis 2,5 Volt umzusetzen, so dass die Ausgangsspannung heruntergeteilt werden muß. Hierzu wurde ein einfacher Widerstandsspannungsteiler mit einem Verhältnis von eins zu drei eingesetzt. Zum Schutz der AD-Umsetzer sichert eine 2,4 Volt Z-Diode den Ausgang der Spannungsteiler. Das so aufbereitete Ausgangssignal der Transducer wird dann an den AD-Umsetzer im Prozessor weitergeleitet, um dort verarbeitet zu werden (s. Abbildung 5.6).

5.3.3 Netzteil

Das Netzteil wurde nur integriert, weil es die Handhabung der Apparatur erleichtert, da man so zum Betrieb von Transducer und Evaluation Board kein Labornetzteil mehr benötigt. Es besteht hauptsächlich aus einem Transformator und zwei Spannungsreglern vom Typ LM317, deren Spannung unabhängig voneinander einstellbar ist. Es werden sowohl 10 Volt zur Versorgung der CMOS-Bauteile und des Evaluation Boards als auch 24 Volt für den Betrieb der Transducer erzeugt.

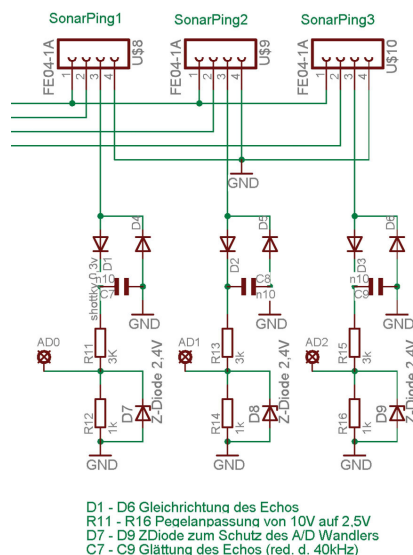


Abbildung 5.6: Ausschnitt aus dem Schaltplan: Echogleichrichter

5.4 Die Firmware

Die grundlegende Funktionalität des Sonars lässt sich als abstrakter Datentyp (*ADT*) darstellen, auf den über eine definierte Schnittstelle zugegriffen wird.

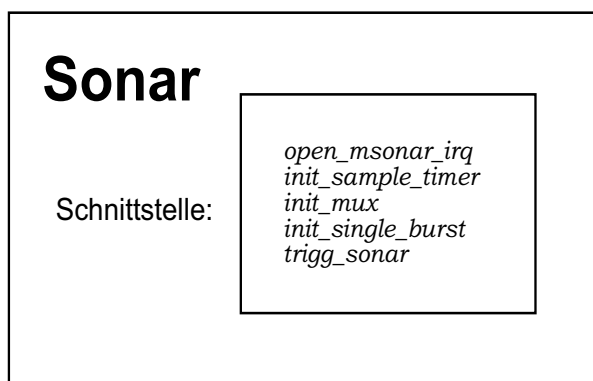


Abbildung 5.7: ADT Sonar

Die Schnittstelle besteht aus fünf Funktionen von denen vier für die Initialisierung der Hardware benötigt werden, um eine Individuelle Parametrisierung zu ermöglichen. Die fünfte Funktion ist diejenige, die die eigentliche Messung ausführt.

1. *open_msonar_irq(MultiSonarIrqDesc, Mode)*
Diese Funktion initialisiert den im MultiSonarIrqDesc definierten AD Umsetzer mit den in Mode übergebenen Parametern. Sie setzt die Interruptpriorität des ADC und initialisiert den AIC (*Advanced Interupt Controller*), so dass der ADC Interrupts auslösen kann. Als auslösender IRQ wird die beendete Umsetzung des dritten Kanals selektiert, da zu diesem Zeitpunkt die Kanäle eins und zwei bereits umgesetzt sind.

2. *init_sample_timer(TimerDescriptor, Countval, Masterclock)*
Hiermit wird der Sampletimer initialisiert, der durch den TimerDescriptor beschrieben wird. Dieser Timer ist derjenige, der den AD-Umsetzungsprozess anstößt. Countval legt den Wert fest, bei dem der Ausgang des Timers toggelt. Masterclock legt fest, mit welcher Geschwindigkeit gezählt wird. So hat man die Möglichkeit, die Samplingfrequenz an seine Bedürfnisse anzupassen.

3. *init_mux()*
Hier werden die unteren vier Bit des Port B als Ausgänge definiert und auf logisch 0 gesetzt, so dass der Mux den Sonartransducer 0 selektiert.

4. *init_single_burst(Bursttime, Periode, Frequenz)*
Mit dieser Funktion werden die Timer TC0 und TC1 initialisiert. Bursttime ist die Dauer des Bursts in Millisekunden, Periode ist die Zeit in millisekunden, nach der frühestens der nächste Burst generiert wird, und Frequenz gibt die Burstfrequenz in Hz an. Die Funktion initialisiert die Timer, startet den Burst aber noch nicht.

5. *trig_burst(Sonarnr)*
Der Aufruf dieser Funktion führt die eigentliche Sonarmessung durch. Mit dem Parameter Sonarnr muss die Nummer des Transducers angegeben werden, der den Sonarburst aussenden soll. Nachdem der Burst ausgelöst wurde, wird der Sampletimer gestartet und der AD Umsetzter beginnt die Daten zu digitalisieren. Die digitalisierten Werte werden in einem Datenpuffer in der MultiSonarIrqDesc Struktur gespeichert. Mit Erreichen des Pufferendes wird die Umsetzung der Daten beendet, der ADC deaktiviert und der Sampletimer angehalten.

Der zum Ansprechen und Auswerten der Sonardaten benötigte Interruptdescriptor MultiSonarIrqDesc ist eine C-Struktur, welche folgende Daten beinhaltet.

```

typedef struct
{
    const AdcDesc    *adc ;                /* ADC Descriptor
    TypeAICHandler  *AsmMultiSonarIrqHandler; /* Assembler IRQ entry point
    const TCDesc    *timer;               /* Timer Descriptor for sampling
    samplebuffer    *buffer_pt_s0;        /* Pointer to Sonar0 Dataarray
    samplebuffer    *buffer_pt_s1;        /* Pointer to Sonar1 Dataarray
    samplebuffer    *buffer_pt_s2;        /* Pointer to Sonar2 Dataarray
} MultiSonarIrqDesc ;

```

Diese Struktur definiert die benötigte Hardware, AD-Umsetzer und Sampletimer, den Einstiegspunkt der Assembler-IRQ-Routine und drei Datenstrukturen, welche die umgesetzten Daten aufnehmen können.

5.5 Hardware Tests

Nach Fertigstellung des Interfaces sind erste Tests mit allen Hardwarekomponenten erforderlich. Das Interface wird an den I/O Connector des Evaluation Boards angeschlossen, da dort alle I/O Pins des Prozessors zu finden sind, die für dessen Ansteuerung von Nutzen sind. Auf dem Interface selber sind drei Buchsen für den Anschluss der Transducer vorgesehen, so dass sich nach dem Verbinden ein Testaufbau wie in Abbildung 5.8 ergibt.

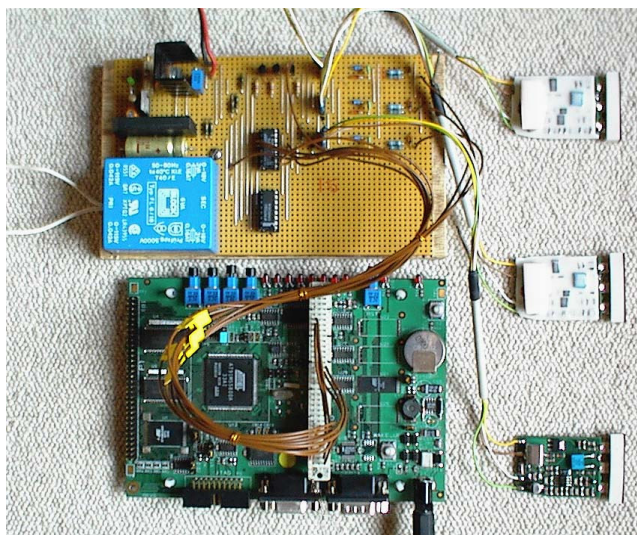


Abbildung 5.8: Testaufbau mit Evaluation Board, Interface und Transducern

Es werden insgesamt 9 Leitungen für die Ansteuerung des Interfaces, und somit der Transducer, benötigt.

- TIOA1
Ausgang des Timers TIOA1, der für die Generierung des Bursts vorgesehen ist.
- PB0, PB1, PB2
Standard I/O Pins an Port B des Prozessors zur Ansteuerung des Multiplexers auf dem Interface.
- AD0, AD1, AD2

Eingänge des 4-Kanal-Analog/Digital-Umsetzers ADC0 zum parallelen Samplen der Ultraschallechos aller Transducer.
- 3,3 V, GND
Spannungsversorgung für den Eingang des Voltage Translators auf dem Interface, sowie Potentialausgleich zwischen Interface und Evaluation Board.

Die Transducer selber sind alle drei mit jeweils vier Leitungen angeschlossen.

- 24 V, GND
Spannungsversorgung der Transducerelektronik.
- Burst in
Eingang für das Burstsinal.
- Signal out
Echoausgang der Transducerelektronik.

Zum Testen der Hardware musste ein Programm geschrieben werden, welches alle im späteren Betrieb vorkommenden Möglichkeiten der Ansteuerungen des Interfaces durchspielt. Das Programm ist in der Lage alle drei Transducer über die oben genannten Steuersignale anzusteuern, sowie die Echos der Transducer einzeln als auch gleichzeitig einzulesen. Die Testroutinen sind von vornherein so aufgebaut, dass sie den Anforderungen des ADTs (siehe Kapitel 5.4) genügen, um später in der weiteren Programmierung übernommen werden zu können.

5.5.1 Burst Test

Als erstes sollte der digitale Teil des Interfaces getestet werden. Diese Tests erfolgten ohne Transducer und dienten hauptsächlich der Dimensionierung des Burstimpulses, der Überprüfung der Multiplexerfunktion sowie der beiden Pegelanpassungsstufen.

Hierzu wurde die Funktion *init_single_burst()* geschrieben, welche unter Einsatz zweier Timer des Prozessors (TC0, TC1) einen 45 kHz Burst erzeugt. Der Burst am Sonar-0-Ausgang wird gestartet durch Drücken des SW1 Tasters auf dem Evaluation Board, erneutes Drücken dieses Tasters erzeugt einen Burst am Sonar-1-Ausgang und ein dritter Tastendruck erzeugt einen Burst am Sonar-2-Ausgang des Interfaces.

Die Folgende C-Funktion zeigt die Initialisierung der beiden kaskadierten Timer zur Erzeugung des Bursts:

```

/*-----
/* Function Name      : single_burst
/* Object            : Inits TC0 and TC1 for generating burst.
/*
/* Input Parameters  : <bursttime> in steps of 1 microsec. min. 8 microsec.
/*                   : <period> in steps of 1 microsec. min. 8 microsec.
/*                   : <burstfrequency> in Hz
/* Output Parameters : Returncode. if 0 everything is OKAY
/*-----

int init_single_burst (int bursttime, int period, int burstfrequency)
{
    if (bursttime > period)                // check if bursttime ist longer than period
        return 1;
    if ((bursttime / 8 > 65535) || (bursttime / 8 < 1)) // check for too long bursttime
        return 1;
    if ((period / 8 > 65535) || (period / 8 < 1)) // check for too long period
        return 1;

    // everything is okay start generating Burst
    TC0_BASE->TC_BMR = TC_TIOA0XC1; // connect TIOA0 to XC1 for generating burst

    // Init TC0 for burststimming
    at91_tc_open (&TC0_DESC,TC_CLKS_MCK128 | // count on 1/128 MCK (250KHz, MCK=32MHz)
                 TC_BURST_NONE | // No burst gating
                 TC_BEVTEDG_EDGE_NONE | // External Event Edge --> None
                 TC_EVT_XC0 | // Exter. Event set to XC0 (TIOB = Output)
                 TC_WAVE | // Set Wave Mode
                 TC_ACPC_CLEAR_OUTPUT | // Register A compare clear TIOB (burstend)
                 TC_ACPC_CLEAR_OUTPUT | // Register C Compare clear TIOB (periodend)
                 TC_CPCSTOP | // Stops counter on register C compare
                 TC_ASWTRG_SET_OUTPUT, // Software trig SET TIOA
                 FALSE,FALSE ); // Disable TIOA & TIOB output

    (&TC0_DESC)->tc_base->TC_RA = (2*bursttime)/8; //set bursttime in steps of 1msec.
    (&TC0_DESC)->tc_base->TC_RC = (2*period)/8; //set period in steps of 1msec.

    // Init TC1 for burstfrequency
    at91_tc_open (&TC1_DESC,TC_CLKS_MCK2 | // count on 1/2 MCK (16MHz, MCK=32MHz)
                 TC_BURST_XC1 | // Burst gated on XC1
                 TC_BEVTEDG_FALLING_EDGE | // External Event Edge --> Falling
                 TC_EVT_XC1 | // Exter. Event set to XC2 (TIOB = Output)
                 TC_WAVE | // Set wave mode
                 TC_ACPC_TOGGLE_OUTPUT | // Register C compare toggle TIOA
                 TC_CPCTRG | // Reset counter on register C compare
                 TC_AEEVT_CLEAR_OUTPUT | // To clear output if burst ends
                 TC_ASWTRG_SET_OUTPUT, // Software trig SET TIOA
                 TRUE,FALSE ); // Enable TIOA and disable TIOB output

    (&TC1_DESC)->tc_base->TC_RC = 8000000/burstfrequency; // Set frequency

    (&TC1_DESC)->tc_base->TC_CCR = TC_SWTRG; // Trigg timer1 (frequency)

    return 0;
}

```

Mit Abarbeitung dieser Funktion werden beide Timer initialisiert sowie Timer 1 gestartet. Mit den Befehlen

```

init_single_burst (400, 50000, 45000);
(&TC0_DESC)->tc_base->TC_CCR = TC_SWTRG;

```

kann nun ein einzelner Burstimpuls von 400 μ s Länge bei einer Frequenz von 45 kHz ausgelöst werden, das entspricht einer Anzahl von 18 Schwingungen.

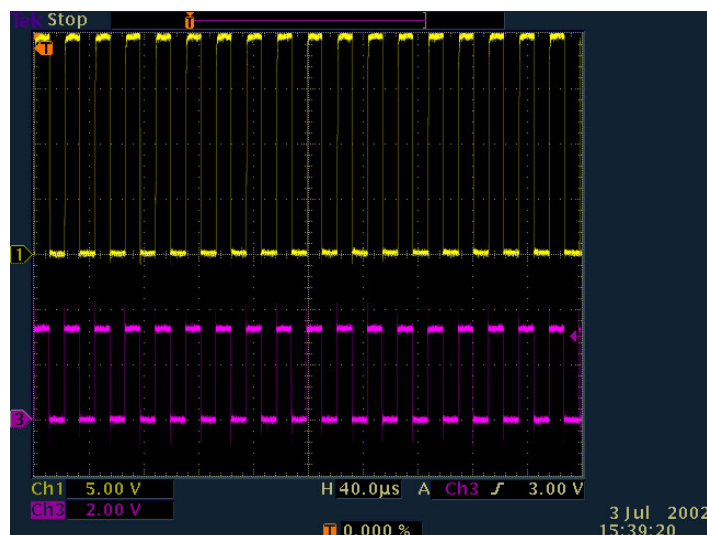


Abbildung 5.9: Messung des Bursts

Die Abbildung 5.9 zeigt im unteren Teil das Burstsignal am Ausgang des Timers TC1 bei 2 Volt/Div und im oberen Teil dasselbe Signal nach der zwei-stufigen Pegelanpassung und dem Multiplexing am Sonar-0-Ausgang bei 5 Volt/Div. Beide wurden mit einer Ablenkungsgeschwindigkeit von 40 μ s/Div aufgezeichnet. Deutlich lässt sich die Erhöhung der Spannung von 3,3 Volt auf 20 Volt bei gleichbleibender Phasenlage sowie eine Periodendauer von 22 μ s (\approx 45 kHz) bei einer Burstlänge von 400 μ s erkennen. Dies entspricht den von Marco geforderten Eckdaten für den Betrieb der Transducer (vgl. Abschnitt 5.2).

Die folgende Funktion steuert vor dem Triggern des Burstimpulses die unteren 4 Bit des Port B des Microcontrollers (PB0, PB1, PB2, PB3) an, wodurch mit Hilfe des Multiplexers auf dem Interface einer der drei Transducer selektiert wird.

```

/**-----
/** Function Name      : trigg_sonar
/** Object             : Triggs sonarburst on special sonar
/**                   : Timer has to be initialized with init_sigle_burst()
/**                   : It sets multiplexer and triggs timer 0
/** Input Parameters   : <number> Number of the sonar to be trigged (0 to 2)
/** Output Parameters  : None
/**-----
void trigg_sonar (u_short number)
{
    switch (number)        //switch on Sonar number
    {
        //Set the lowest 4 bit on PIOB to 0 for selecting sonar 0 on MUX
        case 0: at91_pio_write (&PIOB_DESC, 0x0000000F,PIO_CLEAR_OUT) ;
                break;

        //clear the lowest 4 bit on PIOB to 0001 for selecting sonar 1 on MUX
        case 1: at91_pio_write (&PIOB_DESC, 0x0000000E,PIO_CLEAR_OUT) ;
    }
}

```



```

        //Set the lowest 4 bit on PIOB to 0001 for selecting sonar 1 on MUX
        at91_pio_write (&PIOB_DESC, 0x00000001,PIO_SET_OUT) ;
        break;

        //clear the lowest 4 bit on PIOB to 0010 for selecting sonar 2 on MUX
    case 2: at91_pio_write (&PIOB_DESC, 0x0000000D,PIO_CLEAR_OUT) ;
        //Set the lowest 4 bit on PIOB to 0010 for selecting sonar 2 on MUX
        at91_pio_write (&PIOB_DESC, 0x00000002,PIO_SET_OUT) ;
        break;
    }
    //trigg the timer TC0 generating burst on selected sonar
    (&TC0_DESC)->tc_base->TC_CCR = TC_SWTRG;
}

```

5.5.2 Analog Tests

Der analoge Teil des Interfaces zur Vorverarbeitung des Echosignals der Transducer lässt sich nur in Verbindung mit den Transducern testen. Somit wurden Messungen des originalen Echosignals eines Transducers durchgeführt, um den Zustand der Rohdaten zu ermitteln.

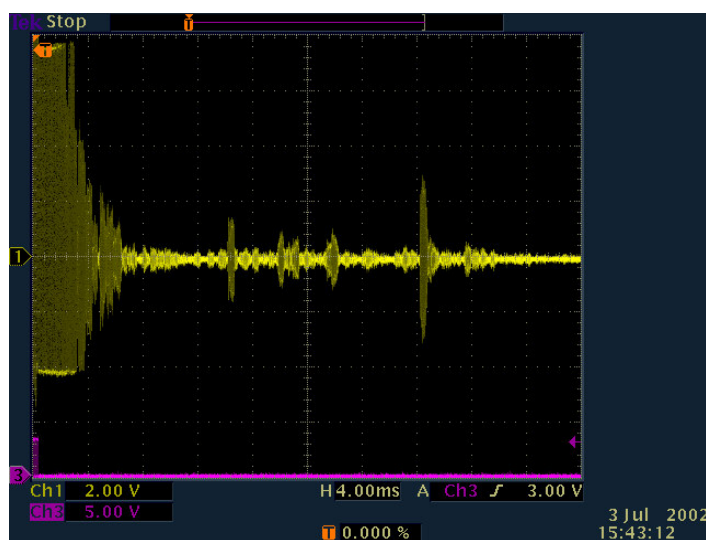


Abbildung 5.10: Echoausgang des Transducers ohne Vorverarbeitung

Die Abbildung 5.10 zeigt das Oszillographierte Echo eines Transducers. Die obere Kurve zeigt den Echoausgang des Transducers bei 2 Volt/Div, die untere das Burstsinal, mit dem der Transducer bei 5 Volt/Div angesteuert wurde. Beide wurden mit einer Ablenkungsgeschwindigkeit von 4 ms/Div aufgenommen.

Man kann deutlich erkennen, dass der Transducer auch nach Ende des Bursts noch lange nachschwing (zum Problem des Ringing siehe auch Abschnitt 4.1). Erst bei 4 ms hat er sich soweit beruhigt, dass man auf der Abbildung mit bloßem Auge die Echos erkennen kann. Im Bereich zwischen 6 und 8 ms ist das erste Echo zu sehen, im Abschnitt 10 bis 16 ms befinden sich drei weitere. Alle Echos waren,

wie auch der Burstimpuls, fast symmetrisch, sie schwangen also um den Nullpunkt. Aufgrund der Tatsache, dass der AD-Umsetzer nicht in der Lage war negative Spannungen umzusetzen und die negativen Halbwellen des Echos auch keine zusätzlichen Informationen lieferten, erschien die Gleichrichtung des Echos als sinnvolle Lösung.

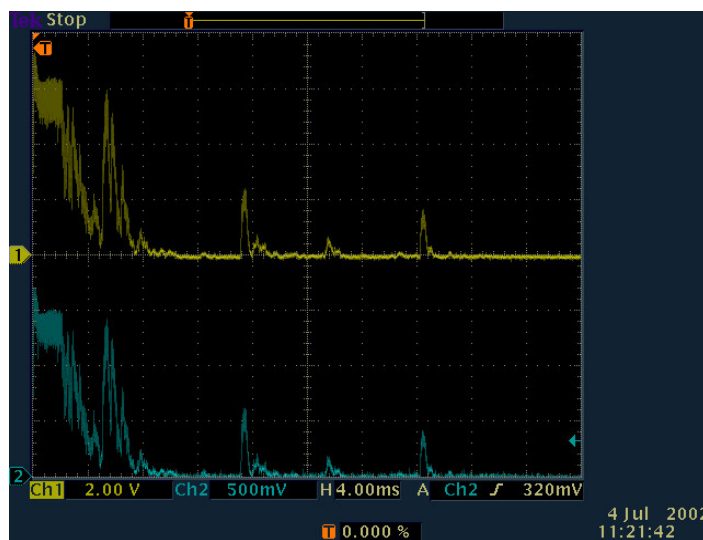


Abbildung 5.11: Echo des Transducers nach der Vorverarbeitung

Abbildung 5.11 zeigt in der oberen Hälfte das Ultraschallecho nach der Gleichrichtung und Glättung. Alle negativen Halbwellen sind durch den Gleichrichter ausgelöscht und die daraus resultierende pulsierende Gleichspannung ist durch den Kondensator fast vollständig geglättet. So ergibt sich die Hüllkurve des Echos. Im unteren Teil des Bildes sieht man das Signal, das schließlich den AD-Umsetzer erreicht. Es unterscheidet sich von dem oben abgebildeten nur in Bezug auf die Amplitude. Da beim Samplen die Eingangsspannung des AD-Umsetzers nicht höher als seine Referenzspannung sein darf, wird das originale Echo, bevor es dem AD-Umsetzer zugeführt wird, noch einmal heruntergeteilt.

Die Abbildung 5.12 visualisiert die Echos der drei Transducer so, wie sie von dem Prozessor eingelesen werden. An der horizontalen Achse sind die Zeitwerte bereits durch die korrespondierende Entfernung in Millimeter ersetzt. Die rosa Linie gehört zu dem Transducer, der den Ultraschallimpuls ausgesandt hat, und ist zu erkennen an dem starken Nachschwingen. Die gelbe sowie die blaue Linie zeigen die Echos der beiden anderen Transducer, die nicht gesendet haben. Signifikante Hindernisse sind deutlich bei 1520 mm und bei 3200 mm zu erkennen, in der Realität handelt es sich um meinen Papierkorb und die Wand meines Zimmers (siehe Abbildung 5.13).

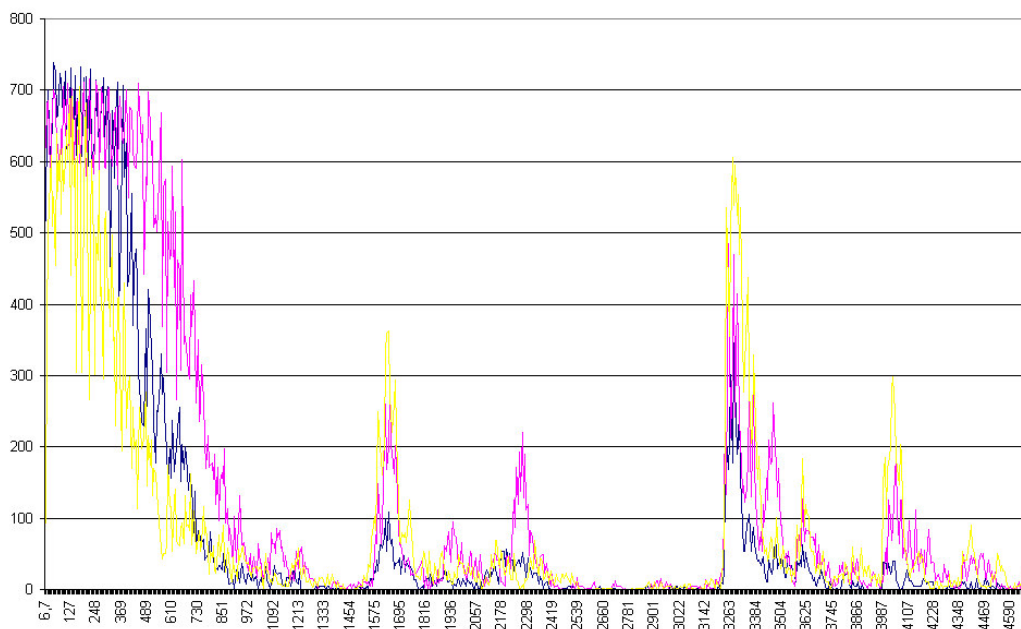


Abbildung 5.12: Gesampelte Echos aller drei Transducer

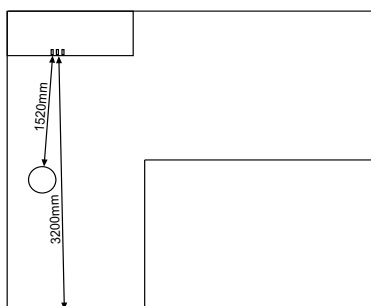


Abbildung 5.13: Grundriss meines Zimmers

Aufgezeichnet wurden die Kurven mit Hilfe der folgenden Interrupt Service Routine, die immer aufgerufen wird, sobald alle drei Kanäle vollständig umgesetzt sind. Weiterhin sorgt dieser IRQ Handler dafür, dass in dem Moment, in dem der Datenpuffer voll ist, der IRQ disabled und der AD-Umsetzter gestoppt wird. So braucht der Anwender nicht darauf zu achten den Interrupt wieder rechtzeitig auszuschaalten.

```

/*-----
/* Function Name      : c_handler_msonar
/* Object            : C handler interrupt function called by the interrupt
/*                   : assembling routine
/*                   : samples 3 channels into 3 samplebuffers
/*                   : if buffer full IRQ & timer are disabled
/* Input Parameters   : <Irq Descriptor> MultiSonarIrqDesc
/* Output Parameters  : none
/*-----
void c_handler_msonar(MultiSonarIrqDesc *msonar_pt)
/* Begin
{
    /*Check if all channels are already converted
    if ((msonar_pt->adc->adc_base->ADC_SR & (ADC_EOC0 | ADC_EOC1 | ADC_EOC2)) != 0x7)
    {
        // Set LED2 to ON indicating conversion not ready
        at91_pio_write (&PIOB_DESC, LED2, LED_ON );
    }

    /* Read Samples into bufferstructure
    msonar_pt->buffer_pt_s0->buffer[msonar_pt->buffer_pt_s0->writecount] = msonar_pt->adc->adc_base->ADC_CDR[0];
    msonar_pt->buffer_pt_s1->buffer[msonar_pt->buffer_pt_s1->writecount] = msonar_pt->adc->adc_base->ADC_CDR[1];
    msonar_pt->buffer_pt_s2->buffer[msonar_pt->buffer_pt_s2->writecount] = msonar_pt->adc->adc_base->ADC_CDR[2];

    /* Check if buffer full
    if (msonar_pt->buffer_pt_s0->writecount >= BUFFER_SIZE - 1)
    {
        //stops samplecounter & ADC & disables ADC IRQ
        close_adc_irq(msonar_pt);
        end_burst ();
    }
    else
    {
        /* Increase buffer writecounter
        msonar_pt->buffer_pt_s0->writecount++;
        msonar_pt->buffer_pt_s1->writecount++;
        msonar_pt->buffer_pt_s2->writecount++;
    }
}
/* End
}

```

Die Sampling-Rate betrug in diesem Test 25 kHz, circa 5 kHz unter der mit drei Kanälen maximal möglichen Frequenz. Die maximal mögliche Frequenz ergibt sich aus der Tatsache, dass der interne AD-Umsetzer nach dem Prinzip der Sukzessiven Approximation [Lattmann] arbeitet, und für die Umsetzung eines 10-Bit-Wertes 11 Clockzyklen benötigt. Die maximale Taktfrequenz des Umsetzers liegt bei 1 MHz, so dass die Umsetzung eines einzelnen Wertes bei 11 μ s liegt. Folglich werden für die Umsetzung von drei Kanälen 33 μ s benötigt. Dies entspricht einer maximalen Samplingfrequenz von 30,3 kHz.

5.6 Zusammenfassung

Mit Ende dieser Tests fand auch die Hardwareentwicklung ein Ende. Alle drei Sonare sind über das Interface an das Evaluation Board angeschlossen und einzeln sowie gemeinsam ansteuerbar und auslesbar. Es erfolgt eine einfache analoge Vorverarbeitung der Echos, um die Datenmenge zu reduzieren und so die benötigte Sampling-Frequenz zu senken. So ist es möglich mit einer Samplingfrequenz von 25 kHz umzusetzen, ohne signifikante Informationen zu verlieren.

Weiterhin ermöglicht die Hardware eine Erweiterung auf bis zu acht Sonartransducer, da der Prozessor über acht AD-Kanäle verfügt und auch der Multiplexer zur Verteilung des Burstimpulses acht Kanäle aufweist.

In Bezug auf die Transducer der Firma Marco bleibt zu prüfen, ob sie sich aufgrund ihrer Bauform und ihrer Empfindlichkeit gegenüber mechanischen Belastungen problemlos in die Zielumgebung integrieren lassen und dort die notwendige Ausfallsicherheit sowie Lebensdauer aufweisen.

Kapitel 6

Erste Sonarmessungen

In diesem Kapitel werden die ersten Messungen und deren Ergebnisse diskutiert. Die Tests erproben verschiedene Sensorpositionen in typischen Umgebungen.

6.1 Sinnvolle Sensorenanordnungen

Die Funktionalität einer Sensorenanordnung ist stark abhängig von deren Aufgabe und Einsatzgebiet. Bezogen auf die Positionsbestimmung von Robotern erschien es sinnvoll, die Winkel der Sensoren zueinander zu variieren. Im Rahmen dieser Arbeit war es dabei erforderlich, sich auf drei Anordnungsmodelle zu beschränken, obwohl weitere Anordnungen sich durchaus als sinnvoll erweisen könnten. Die gewählten Modelle erproben jeweils eine parallele, eine konkave und eine konvexe Sensorenanordnung.

1. Paralleler Aufbau

Alle Sensoren sind parallel im Abstand von 10 cm zueinander angeordnet. So entsteht eine gleichmässige Verteilung der Mehrfachabdeckungen.

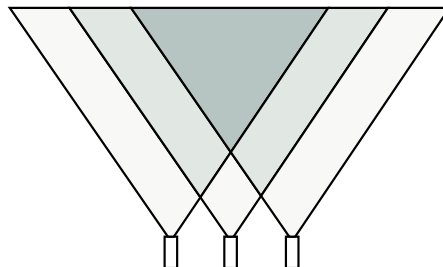


Abbildung 6.1: Parallele Anordnung

2. Konkaver Aufbau

Die Sensoren befinden sich zueinander wieder in einem Abstand von 10 cm. Die äußeren Sensoren strahlen in einem Winkel von 20° abweichend von der Parallelen nach innen. In Abbildung 6.2 ist deutlich zu sehen, dass diese Anordnung die Fläche, die durch drei Sensoren abgedeckt wird, maximiert. Sie wächst mit zunehmender Entfernung aber nicht ganz so schnell wie bei der parallelen Anordnung.

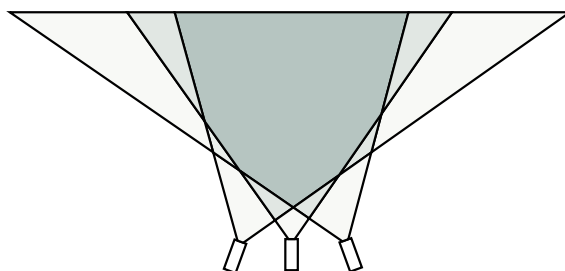


Abbildung 6.2: Konkave Anordnung

3. Konvexer Aufbau

Bei diesem Aufbau strahlen die beiden äußeren Sensoren um 20° abweichend von der Parallelen nach außen. Der Abstand beträgt wieder 10 cm. Hierdurch wird die Fläche mit dreifacher Abdeckung zwar erheblich reduziert, dafür vergrößert sich aber der Erfassungsbereich signifikant.

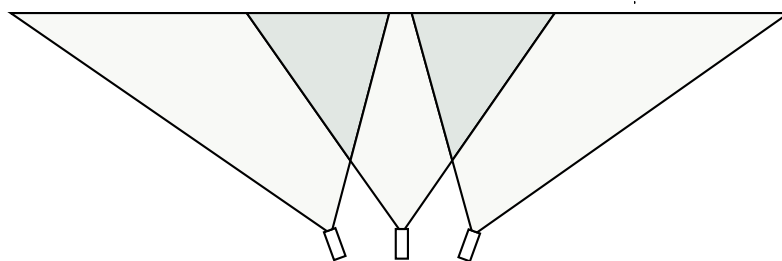


Abbildung 6.3: Konvexe Anordnung

6.2 Leerlauftest

Der Leerlauftest dient in erster Linie dazu das Verhalten der Sonare aufzuzeigen, wenn keine Hindernisse, also auch keine Echos vorhanden sind. Dies ist notwendig, um bei den späteren Messungen entscheiden zu können, ob die empfangenen

Signale das Echo eines Hindernisses sind, oder ob es sich um Rauschen und/oder systematisches Übersprechen unter den Sensoren handelt. Diese Messungen dienen sozusagen als Referenzmessungen und bilden somit den Bezugspunkt für die folgenden Tests.

Die Leerlauftests gestalteten sich unerwartet schwierig, da kein so großer Raum zur Verfügung stand, in dem man eine Messung hätte durchführen können, ohne überhaupt ein Echo zu empfangen. Der Versuch, dies Problem durch einen Testaufbau an einem offenen Fenster (s. Abbildung 6.4) im ersten Stock zu beseitigen, scheiterte daran, dass die Sensoren im Bereich von bis zu 1,5 m fast im rechten Winkel zu ihrer Längsachse abstrahlen und empfangen. Die Echos sind auf Abbildung 6.5 zu sehen. Die vertikale Achse bezeichnet die Amplitude des Echos, die horizontale Achse die Entfernung in Millimeter. Die Kurven zeigen die jeweiligen Echos der Sensoren: Blau die des ersten, rosa die des zweiten und gelb die des dritten Sensors. Nummeriert wurde von links nach rechts. Im Bereich ab 560 mm kann man deutlich die Echos erkennen, die durch die Begrenzung der Fensteröffnung links und rechts entstanden.



Abbildung 6.4: Testaufbau am Fenster.

Nach einigen Veränderungen bezüglich des Aufstellungswinkels der Sensoren innerhalb der Fensteröffnung gelang es dann aber doch eine Messung zu provozieren, in der fast kein Echo verzeichnet wurde. Abbildung 6.6 zeigt genau diese Messung. Der Bereich von null bis 550 mm Entfernung lässt keinerlei Messung zu, da er komplett vom Ringing (s. Abschnitt 4.1) des aktiven Sensors und dem direkten Übersprechen des Sonarpings auf die beiden passiven Sensoren überdeckt wird. Zwischen 580 mm und 1000 mm Entfernung sind wieder die Begrenzungen der Fensteröffnung zu sehen, dies mal aber mit wesentlich geringerer Amplitude.

Der Rest des Diagramms zeigt absolute Stille, so dass das Leerlaufrauschen der analogen Signalvorverarbeitung zu sehen ist. Dies kann aber aufgrund der geringen Lautstärke bei den folgenden Messungen vernachlässigt werden, da es keine relevante Größe darstellt.

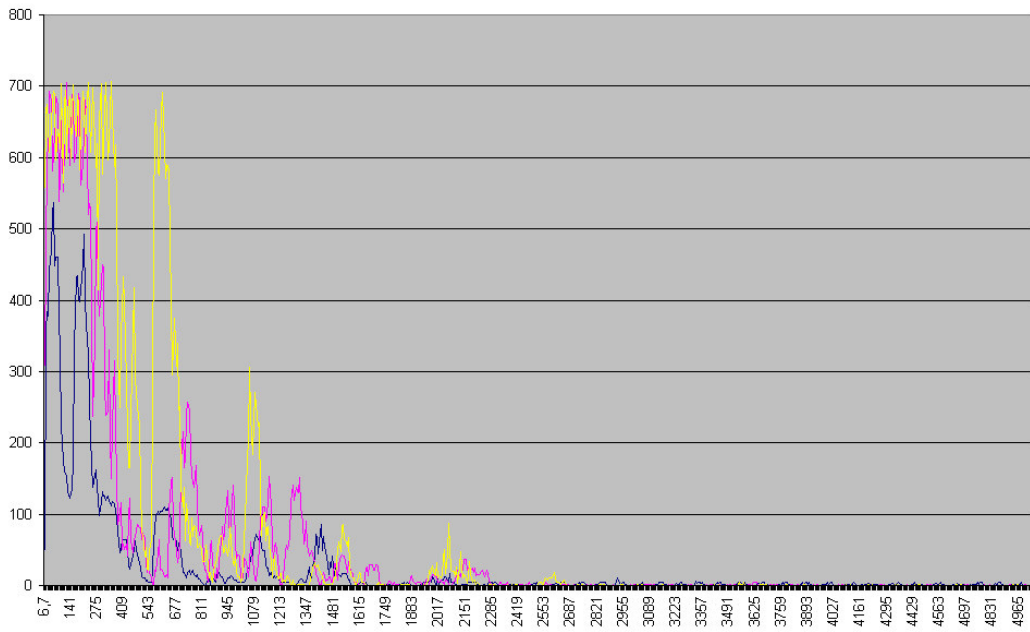


Abbildung 6.5: Echos am offenen Fenster.

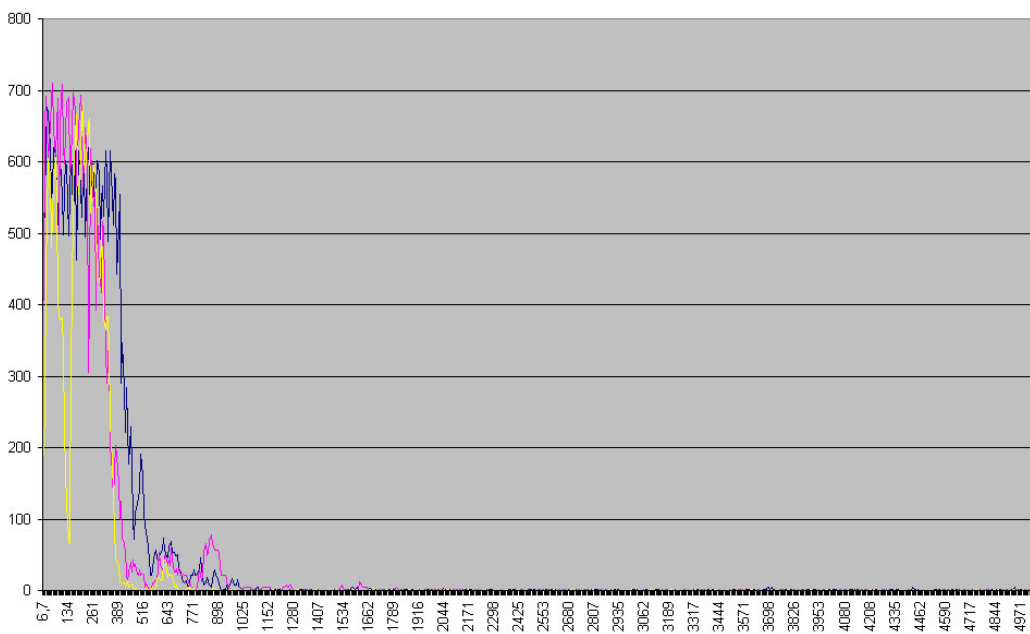


Abbildung 6.6: Keine Echos am offenen Fenster.

6.3 Das Referenzobjekt

Als Grundlage für die Messungen wurde ein Referenzobjekt definiert, von dem mit den unterschiedlichen Anordnungen Sonarmessungen gemacht werden sollten. Dies ist nötig, um die Echos der verschiedenen Sensoranordnungen direkt vergleichen zu können, da ja die günstigste der drei Anordnungen ermittelt werden soll.

Als Referenzobjekt diente der Eingangsbereich einer Halle, der eine markante Anordnung aus Innen- und Außenecken bot, sowie genügend Platz, um einen Testaufbau zu realisieren. Die markante Zusammenstellung der Ecken ist deshalb so interessant, da bei der Roboternavigation häufig mit Landmarken gearbeitet wird. Die wichtigste Eigenschaft einer Landmarke ist ihre eindeutige Identifizierbarkeit. Da bei der Navigation mit Hilfe von Landmarken natürlich mehrere Marken benutzt werden, ist es eine Herausforderung eine sichere Identifikation der verschiedenen Landmarken zu gewährleisten. Eine Verwechslung der Marken führt immer zu einer fehlerhaften Selbstlokalisierung, ein Fehler der kaum korrigierbar ist. Die nächste wichtige Eigenschaft ist, dass die Landmarke es ermöglichen muss die Verschiebungen zu ihr sowie Rotation sicher zu erkennen und zu korrigieren. Diese Gesichtspunkte sprachen alle für dieses Referenzobjekt.



Abbildung 6.7: Referenzobjekt für alle Messungen.

Abbildung 6.7 zeigt die Ecke der Halle, die für die Referenzmessungen gewählt wurde. Abbildung 6.8 zeigt den Grundriss dieser Ecke mit einigen signifikanten Entfernungen, welche sich bei fast allen Messungen wiederfinden lassen.

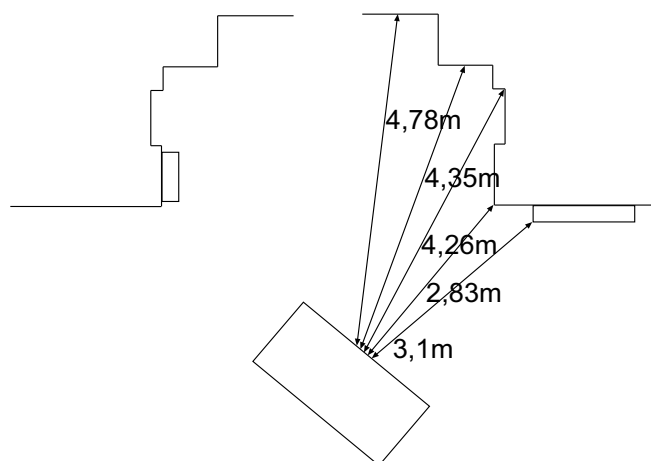


Abbildung 6.8: Grundriss der Referenzecke.

6.4 Paralleler Aufbau

In Abbildung 6.9 sieht man die Echos der drei Sensoren in paralleler Anordnung. Der erste Sensor (blau) ist aktiv, das heißt er ist derjenige, der den Sonarping ausgesandt hat. Deutlich kann man erkennen, dass die Dauer des Ringings länger geworden ist. Erst bei 1025 mm ist eine vollständige Beruhigung eingetreten. Schaut man allerdings genauer hin fällt auf, dass die drei Peaks im Bereich von 420 mm bis 770 mm der Reihenfolge der Sensoren entsprechen und erhebliche Laufzeitunterschiede aufweisen. Sollte es sich bei diesen Signalen um das Echo eines Objektes handeln, so müsste es sich sehr weit links und im Abstand von circa 50 cm bis 60 cm vom Testaufbau befinden. Mit Blick auf die Abbildung 6.7 stellt man fest, dass sich dort der Monitor auf dem Tisch befindet.

Es handelt sich also in diesem Fall nicht um das Ringing, sondern tatsächlich um das Echo eines Objektes. Fraglich bleibt allerdings, ob man in der Lage ist, Echos dieser Art softwaretechnisch zu analysieren.

Der nächste interessante Punkt liegt bei 2820 mm. Der Impuls tritt zeitgleich bei allen Sensoren auf, was dafür spricht, dass das korrespondierende Objekt sich mittig vor dem Aufbau in einer Entfernung von 2,82 m befinden muss. Nach dem Grundriss in Abbildung 6.8 muss es sich um die vordere Ecke handeln, da sie das einzige zentrische Hindernis ist.

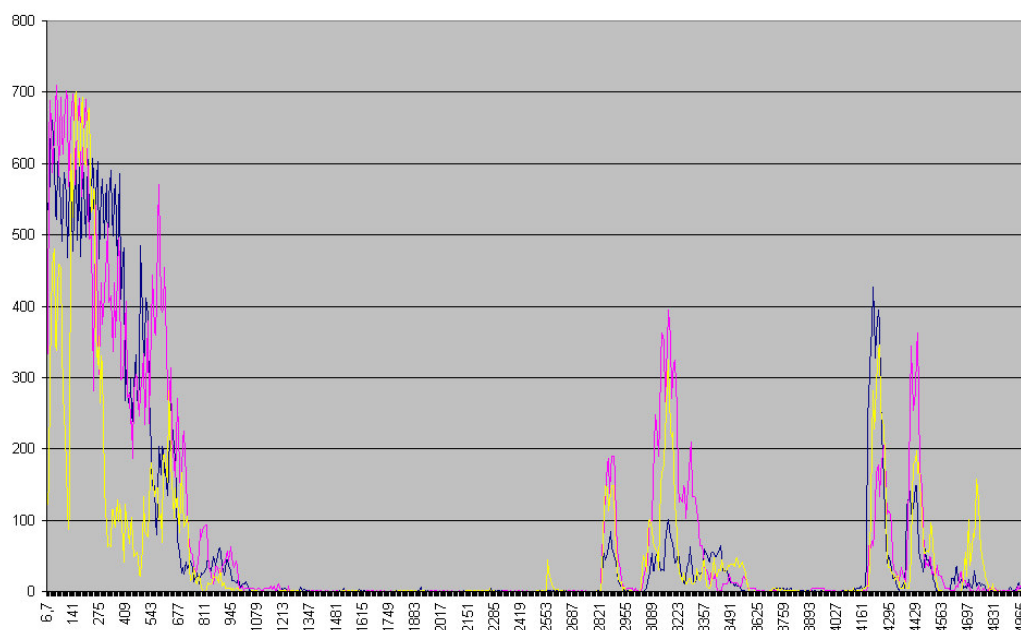


Abbildung 6.9: Echos beim parallelen Aufbau. Sensor eins (blau) ist aktiv.

Das folgende Echo bei 3035 mm zeigt eine deutliche Verschiebung der Anstiegsflanken an den drei Sensoren. Das Echo erreicht als erstes den dritten (gelb, rechts), dann den zweiten (rosa, mitte) und schließlich den ersten Sensor (blau, links). Die Laufzeitunterschiede bedeuten, dass der Schall verschiedene Entfernungen zu den Sensoren zurückgelegt hat. Das Hindernis muss demnach näher am rechten Sensor als am linken sein. Dies lässt darauf schließen, dass sich das Hindernis rechts vom Aufbau befindet. Anhand des Grundrisses wird deutlich, dass es sich um das Echo der Heizung handeln muss (der dunkelblaue Körper an der gelben Wand auf Abbildung 6.7).

Die letzten drei Echopeaks befinden sich bei 4174 mm, 4368 mm und 4640 mm. Bei allen Peaks ist wieder eine Verschiebung der Anstiegsflanken zu sehen, am deutlichsten bei dem dritten Peak. Sie entsprechen weitestgehend den im Grundriss eingetragenen Entfernungen und weisen, je weiter links die echoverursachenden Objekte sich befinden, eine zunehmende Verschiebung der Flanken auf.

6.5 Konvexer Aufbau

Die Abbildung 6.10 zeigt die gleiche Sonarmessung wie im vorangegangenen Abschnitt, nur mit konvexer Anordnung der Sensoren. Die Echos der einzelnen Sen-

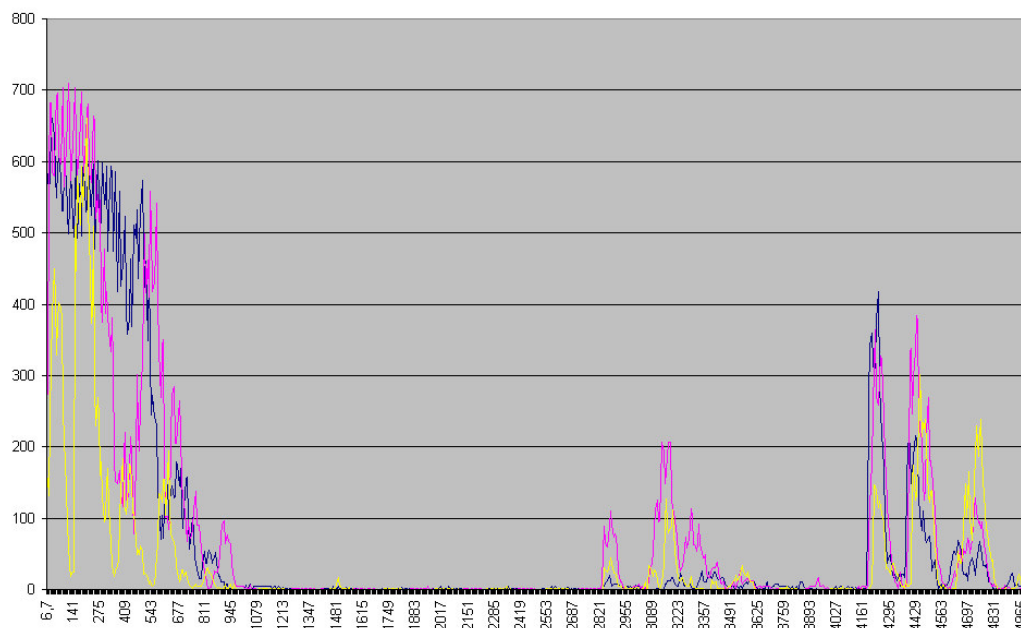


Abbildung 6.10: Echos beim konvexen Aufbau. Sensor eins (blau) ist aktiv.

soren entsprechen grob gesehen denen der Messung mit parallelem Aufbau, da es sich ja um dasselbe Objekt handelt. Im Detail gibt es allerdings diverse Unterschiede.

Im vorderen Bereich bis 1000 mm sind wieder die Echos des Monitors zu sehen. Sie sind bei dieser Messung aber wesentlich lauter, was sich auf die Vergrößerung des Öffnungswinkels durch den konvexen Aufbau zurückführen lässt. Der Monitor wird direkter angestrahlt.

Aber auch hier ist eine softwaregestützte Auswertung des Bereiches unter 1000 mm genauso fraglich wie schon beim parallelen Aufbau.

Die folgenden Echos bei 2820 mm und bei 3035 mm haben dieselben Ursachen wie bei dem parallelen Aufbau, jedoch sind die Amplituden, besonders die des ersten Sensors (blau), wesentlich geringer. Die korrespondierenden Objekte, vordere Ecke und Heizung, sind mittig beziehungsweise rechts vor dem Testaufbau angesiedelt. Durch den konvexen Aufbau (s. Abschnitt 3) sind der erste Sensor nach links und der dritte Sensor nach rechts gedreht. Somit wird klar, warum weiter rechts liegende Objekte schlechter vom ersten Sensor erkannt werden.

Die letzten drei Peaks zeigen bei dem Echo des dritten Sensors (gelb) allerdings etwas, das der eben aufgestellten Theorie über den Anordnungswinkel zu widersprechen scheint. Das erste der drei Echos ist zwar am weitesten rechts erzeugt, hat aber eine geringere Amplitude als weiter links liegende Objekte. Betrachtet

man nun den Grundriss, fällt auf, dass bei konvexer Anordnung der dritte Sensor wesentlich mehr durch die vordere Ecke verdeckt wird, als bei der parallelen Anordnung.

6.6 Konkaver Aufbau

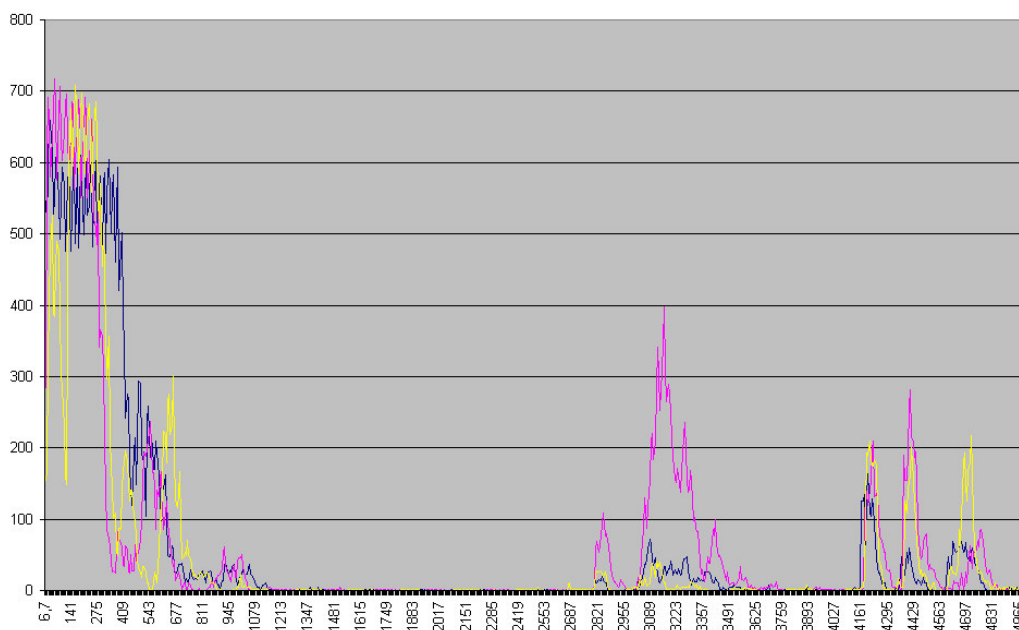


Abbildung 6.11: Echos beim konkaven Aufbau. Sensor eins (blau) ist aktiv.

Auf Abbildung 6.11 sind die Echos des konkaven Aufbaus zu sehen. Wieder sind alle Peaks der vorangegangenen Messungen vorhanden, weisen jedoch wieder andere Ausprägungen auf. Die Echos des Monitors im Anfangsbereich fallen deutlich leiser aus als bei den vorherigen Messungen. Dies lässt sich auch hier auf die Veränderung des Winkels zurückführen, da der erste Sensor bei diesem Aufbau dem Monitor abgewandt betrieben wurde. Erstaunlicherweise ermöglicht die geringere Amplitude hier erstmals eine softwaregestützte Auswertung, da sich die Echos deutlich vom Ringing abheben und besser in Beziehung zu setzen sind als in den vorangegangenen Messungen.

Während dies im vorderen Bereich also von Vorteil ist, wird es im hinteren Abschnitt zum Nachteil. Auch hier haben die Echos nämlich geringere Peakwerte als bei den vorangegangenen Messungen. Interessanterweise sind, anders als beim

parallelen Aufbau, im mittleren Bereich nur die Peaks der äußeren, nach innen gedrehten Sensoren reduziert. Im hinteren Bereich sind hingegen alle Sensoren von der Abschwächung der Amplituden betroffen. Für dieses Verhalten ist leider keine augenscheinliche Erklärung vorhanden. Um sinnvolle Aussagen zu diesem Verhalten machen zu können, wären weitere Tests vonnöten.

6.7 Nachweis der Tauglichkeit für Triangulation

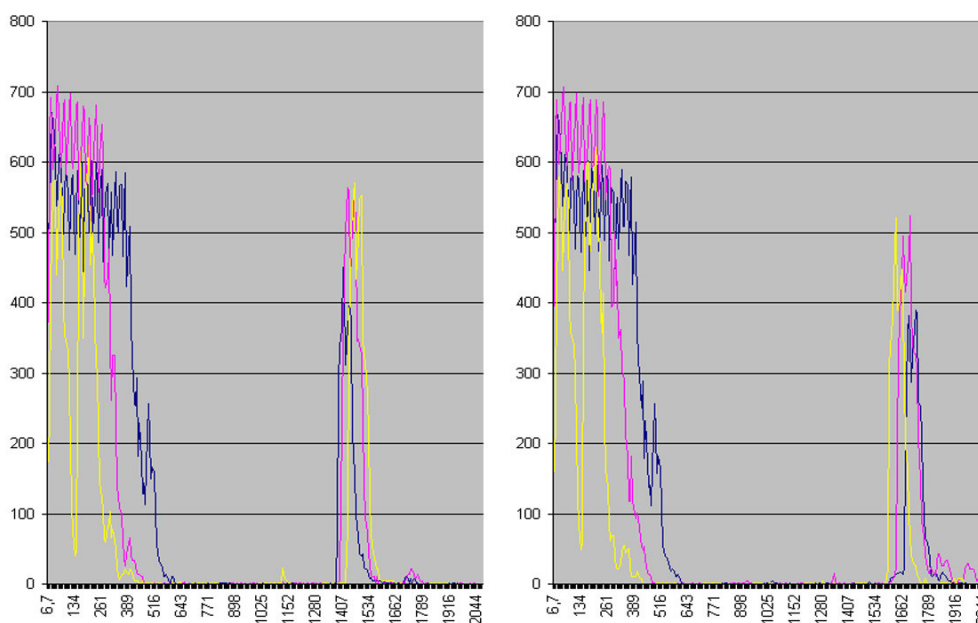


Abbildung 6.12: Freistehendes Hindernis rechts und links.

Die Echos auf Abbildung 6.12 zeigen ein freistehendes Hindernis. Es handelt sich bei beiden Messungen um dasselbe Hindernis, einmal auf der rechten Seite (Messung links) und einmal auf der linken Seite positioniert (Messung rechts).

Auf diesen Bildern ist die Verschiebung der Anstiegsflanken der drei Sensoren besonders gut zu sehen. Steht das Objekt auf der rechten Seite, so erreicht das Echo zuerst den rechten, dann den mittleren und schließlich den linken Sensor (linkes Bild). Steht das Hindernis auf der linken Seite, so erreicht das Echo die Sensoren in umgekehrter Reihenfolge (rechtes Bild).

Die Bestimmung der genauen Position des Objektes mit Hilfe von Triangulation sollte bei solchen Messungen ohne weiteres möglich sein. Zwar ergibt sich hierbei

das Problem der Korrespondenz der Echos, also der Klärung, ob die Echos von ein und demselben Hindernis stammen (siehe auch [Markus Berg 1998]). Dies lässt sich aber dadurch reduzieren, dass die Echos, wenn sie von einem Hindernis stammen, in der Reihenfolge Sensor 1, Sensor 2, Sensor 3 für ein links liegendes Objekt und Sensor 3, Sensor 2, Sensor 1 für ein rechts liegendes Objekt auftreten. Weiterhin besteht eine Messung, nachdem jeder Sensor ein Signal ausgesandt hat, aus jeweils drei Einzelmessungen, durch jeden der drei Sensoren. Sie führt somit zu drei mal drei Kurven. Alle drei Einzelmessungen unterscheiden sich in der Position ihres Initiators, wodurch die Echos der Einzelmessungen ebenfalls eine Verschiebung zueinander aufweisen. Die erhebliche Redundanz der Messergebnisse sollte es ermöglichen, fehlerhafte Korrespondenzen auszuschließen sowie relativ sichere Aussagen über die Umgebung vor den Sensoren machen zu können.

6.8 Fazit

Die Messungen haben gezeigt, welches Potential in der Sonographie mit wenigen Sensoren mit großem Öffnungswinkel steckt. Das Erfassen der Umgebung, die mit drei Sensoren ermöglicht wurde, lässt sich sonst nur mit sechs bis acht Sensoren erreichen. Die fehlenden Richtungsinformationen, resultierend aus dem weiten Öffnungswinkel der Sensoren, können über die Laufzeitdifferenzen der einzelnen Echos mittels Triangulation kompensiert werden.

Bezüglich der Sensoranordnungen haben die Tests keinen eindeutigen Favoriten hervorgebracht. Alle Veränderungen der Anordnungen haben zwar immer signifikante Unterschiede der empfangenen Echos zufolge gehabt, waren jedoch nicht ausreichend, um eine Anordnung als besonders geeignet erscheinen zu lassen. Alle Anordnungen haben bezüglich des Referenzobjektes sowohl Vor- als auch Nachteile gezeigt. Um eindeutige Aussagen über die besondere Eignung spezifischer Anordnungen zu treffen, sind weitere Testreihen an verschiedenen Objekten sowie intensive Analysen der Echokurven nötig.

Nach den vorhergegangenen Tests würde ich die parallele Anordnung bevorzugen, da sie die höchsten Amplituden auf allen Sensoren zur Folge hatte. Dies kann allerdings auch Zufall sein, da keine richtigen Vergleichsmessungen an anderen Referenzobjekten vorgenommen wurden.

Die Tests haben auch gezeigt, dass die recht komplizierten Echokurven noch einen erheblichen Aufwand an Analyse erfordern, um die relevanten Daten sicher zu isolieren und auszuwerten.

Kapitel 7

Resümee

In diesem Kapitel werde ich die entstandene Hard- und Software noch einmal als Ganzes beleuchten und die Ergebnisse zusammenfassen sowie im Ausblick Erweiterungsmöglichkeiten und Ergänzungen diskutieren.

7.1 Zusammenfassung

Diese Arbeit erfüllt im Großen und Ganzen die gestellte Aufgabe, Alternativen im Bereich der Sonarnavigation für mobile Roboter aufzuzeigen. Wie die Messungen bewiesen haben, lässt sich mit drei Ultraschallsensoren großen Abstrahlwinkels eine vergleichbare Abdeckung der Umgebung erreichen, wie sie sonst nur mit sechs bis acht Sensoren mit geringem Abstrahlwinkel erreicht werden kann. Anhand der Messergebnisse wurde gezeigt, dass die fehlenden Richtungsinformationen der Sensoren mit großem Abstrahlwinkel mittels Triangulation kompensierbar sind.

Zu diesem Zweck wurde eine vollständige Experimentierplattform entwickelt. Sie besteht aus einem Mikrocontroller, drei Ultraschallsensoren und einem Interface zur Kopplung von Controller und Sensoren. Sie ermöglicht eine vollständig unabhängige Lösung, da der Mikrocontroller so gewählt wurde, dass er neben dem Erzeugen und Empfangen des Ultraschallsignals auch in der Lage ist, erhebliche Rechenleistung für eine vollständige Auswertung der Echos zu bieten.

Der eingesetzte ARM-Controller hat sich in zweierlei Hinsicht als besonders geeignet erwiesen: Erstens ist man in der Lage die Software für leistungsfähigere Prozessortypen der ARM-Serie portabel zu machen, zweitens hat er eine sehr geringe Stromaufnahme, die sich durch geschickte Übergänge in den Sleepmode noch weiter senken lässt.

Die Sensoren der Firma Marco bewährten sich insofern außerordentlich, als ihr Abstrahl- und Empfangsverhalten die gestellten Erwartungen übertraf. Aufgrund

der Tatsache, dass sie fast vollständig aus Keramik bestehen, sind sie allerdings sehr sensibel in Bezug auf mechanische Belastung. Es existiert in der Form, wie sie für dieses Projekt vorlagen, fast keine Möglichkeit, sie sauber und langzeitstabil in einen Roboter zu integrieren, da keinerlei Gehäuse von Marco geliefert wurde.

Die Messungen haben gezeigt, dass es aufwendiger Programme bedarf, die Echo kurven sicher auszuwerten. Hier sind also noch einige komplexe Filter und Analysefunktionen erforderlich, um die korrespondierenden Echopeaks sauber zu isolieren und zu interpolieren, so dass eine sichere Triangulation durchgeführt werden kann.

7.2 Ausblick

Dieses Projekt bildet erst den Anfang unzähliger Möglichkeiten, ein vollständiges sonarbasierendes Navigationssystem für autonome Roboter zu realisieren. Die softwaretechnische Auswertung der Echos bietet Raum für alle erdenklichen Formen der digitalen Signalanalyse.

Der einfachste Fall wäre die Realisierung eines Schwellwertes wie bei Polaroid. Es wird solange gewartet, bis das erste Echo diese Schwelle übersteigt. Dieses Vorgehen würde allerdings den bis jetzt getriebenen Aufwand nicht rechtfertigen, da eine solche Lösung keines Microcontrollers bedarf.

Ein interessanterer Ansatz ist die vollständige digitale Auswertung des gesamten Echos mittels geeigneter Analysefunktionen. Hier wäre als erstes die Möglichkeit zu nennen, mit Hilfe eines Filters alle etwaigen Störgeräusche und Fremdfrequenzen auszufiltern. Das so gewonnene, störungsfreie Echo könnte dann durch eine kontinuierliche Mittelwertbildung weiter vereinfacht werden.

Die Detektion der Flanken des Echos lässt sich mit Hilfe der Kantenfindung, einer Technik aus der Bildverarbeitung, realisieren. Differenziert man die Echofunktion $E_{(t)}$ zweimal, so erhält man als Resultante d^2E/dt^2 , eine Funktion, deren Nullstellen exakt den steilsten Punkt der Flanke von $E_{(t)}$ bezeichnen. Die Kombination aus Mittelwertbildung und Kantenfindung nennt sich Laplace-Filter, auch bekannt unter dem Namen „Mexikanischer Hut“.

Eventuell lassen sich auch Informationen über Beschaffenheit, Form und Größe des Objektes gewinnen, was für die Wiedererkennung von Landmarken von Nutzen wäre.

Diese Ansätze sind nur ein sehr kleiner Ausschnitt der Möglichkeiten im Bereich der digitalen Filter- und Analysefunktionen. Sie lassen jedoch bereits erkennen, dass es im Rahmen dieses Projektes eine Fülle denkbarer Ansätze gibt, die signifikante Informationen aus den Echos zu extrahieren, um eine sichere Navigation zu gewährleisten.

Abbildungsverzeichnis

1.1	Staubsauger Robot Trilobite von Electrolux	5
1.2	Sojourner Rover im Labor der NASA	6
1.3	Kanaluntersuchungsroboter MAKRO	7
1.4	Überwachungsroboter MOSRO 1	8
1.5	Laufzeitunterschiede beim dreier Sensorarray	11
2.1	Instruction Decode eines CISC	15
2.2	Instruction Decode eines RISC	15
2.3	Infineon C167 mit Evaluation Board	17
2.4	Atmel ARM AT91M55800A	20
3.1	Projektmanager des ARM SDT mit geöffneten Editor	22
3.2	Debugger des ARM SDT	23
4.1	Signaldämpfung bei 100kHz	27
4.2	Murata Ultraschall Transmitter und Receiver	28
4.3	Richtcharakteristik Murata MA40R S	29
4.4	Laufzeitverlängerung durch Transmitter - Receiverabstand	30
4.5	Polaroid Transducer Serie 600	30
4.6	Richtcharakteristik Polaroid Transducer	31
4.7	AirMar <i>AirducerTM</i>	32
4.8	Richtcharakteristik Airmar AT50	32
4.9	Transducer der Firma Marco	33
4.10	Richtcharakteristik der Modulartransducer	34
5.1	Prinzipieller Aufbau des Prozessors mit drei Transducern	36
5.2	Evaluation Board AT91EB55	38
5.3	Marco Ultraschall Sende- Empfangssystem	39
5.4	Anschluss des Transducers	40
5.5	Ausschnitt aus dem Schaltplan: Pegelanpassung	42
5.6	Ausschnitt aus dem Schaltplan: Echogleichrichter	43
5.7	ADT Sonar	43

5.8	Testaufbau mit Evaluation Board, Interface und Transducern . . .	45
5.9	Messung des Bursts	48
5.10	Echoausgang des Transducers ohne Vorverarbeitung	49
5.11	Echo des Transducers nach der Vorverarbeitung	50
5.12	Gesampelte Echos aller drei Transducer	51
5.13	Grundriss meines Zimmers	51
6.1	Parallele Anordnung	54
6.2	Konkave Anordnung	55
6.3	Konvexe Anordnung	55
6.4	Testaufbau am Fenster.	56
6.5	Echos am offenen Fenster.	57
6.6	Keine Echos am offenen Fenster.	57
6.7	Referenzobjekt für alle Messungen.	58
6.8	Grundriss der Referenzecke.	59
6.9	Echos beim parallelen Aufbau. Sensor eins (blau) ist aktiv.	60
6.10	Echos beim konvexen Aufbau. Sensor eins (blau) ist aktiv.	61
6.11	Echos beim konkaven Aufbau. Sensor eins (blau) ist aktiv.	62
6.12	Freistehendes Hindernis rechts und links.	63

Literaturverzeichnis

- 1 Adler 2002** ADLER, Manfred: *Entwurf und Realisierung eines seriellen Anschlusses für Compact Flash Speicherkarten*, university of applied sciences, Fachbereich Informatik, Diplomarbeit, 2002
- 2 Airmar Technology Corporation a** Airmar Technology Corporation (Veranst.): *PiezofLEX Transducers*. – Datenblatt der PiezofLEX Technologie
- 3 Airmar Technology Corporation b** Airmar Technology Corporation (Veranst.): *Ultrasonic Air-Ranging Transducers*. – Datenblatt der Airmar Transducer
- 4 Arkin 1998** ARKIN, Ronald C.: *Behavior Based Robotics*. Cambridge, Massachusetts, London, England : The MIT Press, 1998. – ISBN 0–2620–1165–4
- 5 ARM Limited a** ARM Limited (Veranst.): *Application Note 31, Using Embedded ICE*
- 6 ARM Limited b** ARM Limited (Veranst.): *Debugging Embedded Systems*. – Examines some of the traditional ways of debugging processor systems
- 7 Atmel Corporation** Atmel Corporation (Veranst.): *AT91 Library V2.0 Application Note*. – This document provides user information for the AT91 Library V2.0 Software Package
- 8 Atmel Corporation 1999** Atmel Corporation (Veranst.): *ARM7TDMI Embedded RISC Microcontroller Core*. B. Januar 1999. – Datenblatt des 7TMDI Kerns
- 9 Balzerowski 2002** BALZEROWSKI, Rainer: *Realisierung eines Webcam basierten Kamera Systems für mobile Roboter*, university of applied sciences, Fachbereich Informatik, Diplomarbeit, 2002
- 10 Berg 1996** BERG, Markus: *Untersuchung über die Verwendbarkeit der Impulskompressionstechnik bei Ultraschallsensoren für Autonome Mobile Roboter*, Universität Kaiserslautern, Fachbereich Informatik, Diplomarbeit, 1996

- 11 Borenstein 1996** BORENSTEIN, Feng: *Where am I, Sensors and Methods for Mobile Robot Positioning*. April 1996
- 12 Chmela** CHMELA, Ing. H.: *Ultraschall*. online. – URL <http://www.hcrs.at/ULTRAS.HTM>. – Zugriffsdatum: 5.8.2002
- 13 Dierkes** DIERKES, Dipl.-Ing. M.: *Der Piezoelektrische Effekt*. online. – URL <http://sensorik.uni-paderborn.de/forschung/saw/piezo.html>. – Zugriffsdatum: 5.8.2002
- 14 Eberle** EBERLE, Dr. H.: *Architektur moderner RISC-Microprozessoren*
- 15 GMD** GMD: *Homepage des Fraunhofer AIS*. online. – URL <http://ais.gmd.de/projects/Makro/makro-d.html>. – Zugriffsdatum: 11.8.2002
- 16 IEEE 2002** IEEE: *About the IEEE*. online. 2002. – URL <http://www.ieee.org/about/>. – Zugriffsdatum: 5.8.2002
- 17 Infineon Technologies** Infineon Technologies (Veranst.): *C167CR Derivatives 16-Bit Single-Chip Microcontroller*. – Datenblatt der C-167 Derivate
- 18 King 2002** KING, J. H.: *Mars Pathfinder Rover*. online. März 2002. – URL <http://nssdc.gsfc.nasa.gov/database/MasterCatalog?sc=MESURPR>. – Zugriffsdatum: 5.8.2002
- 19 Kopka 2000** KOPKA, Helmut: *TEX Band 1 - Einführung*. Dritte überarbeitete Auflage. Martin-Kollar-Strasse 10-12, D-81829 München : Addison - Wesley, 2000. – ISBN 3-8273-1557-3
- 20 Lattmann** LATTMANN, M.: *Analog Digital Wandler*. online. – URL <http://www.educeth.ch/informatik/material/adwandler/docs/script.pdf>. – Zugriffsdatum: 5.8.2002
- 21 Leitenberger** LEITENBERGER, Bernd: *CISC und RISC - Die Gegensätze der Rechnerarchitekturen*. online. – URL <http://www.bernd-leitenberger.de/cisc-risc.html>. – Zugriffsdatum: 5.8.2002
- 22 Lindner 2002** LINDNER, Mathias: RISC, CISC, MISC ... In: *c't magazin für computer technik* 2 (2002), Januar, S. 198–200
- 23 Markus Berg 1998** MARKUS BERG, Klaus-Werner Jörg & Jan-Peter P.: *3D Ultraschall-Entfernungsmessung mit Pseudo-Random Frequenzen*. URL <http://ag-vp-www.informatik.uni-kl.de>. – Zugriffsdatum: 1.8.2002, 1998. – Forschungsbericht

- 24 Medicine-Worldwide** MEDICINE-WORLDWIDE: *Ultraschall*. online. – URL <http://www.m-ww.de/enzyklopaedie/medizingeraete/ultraschall.html>. – Zugriffsdatum: 5.8.2002
- 25 Moravec 1988** MORAVEC, H. P.: Sensor Fusion in Certancy Grids for Mobile Robots. In: *AI Magazine* (1988), Summer, S. 61–74
- 26 Murata** Murata (Veranst.): *Air Ultrasonic Ceramic Transducers*. – Datenblatt der Murata Transducer
- 27 Polaroid** POLAROID: *Technical Specifications for 600 Series Instrument Grade Electrostatic Transducers*. : Polaroid (Veranst.). – Spezifikation der Polaroid Transducer
- 28 Robowatch** ROBOWATCH: *Homepage der Firma Robowatch*. online. – URL <http://www.robowatch.de>. – Zugriffsdatum: 1.8.2002
- 29 Schröder 2000** SCHRÖDER, Dr.med. Ralf-Jürgen: *Hochauflösende farbkodierte Duplexsonographie von Hauttumoren*, Medizinische Fakultät Charité der Humboldt-Universität zu Berlin, Habilitationsschrift, 2000. – URL <http://dohost.rz.hu-berlin.de/habilitationen/schroeder-ralf-juergen-20%00-03-07/HTML/schroeder.html>. – Zugriffsdatum: 5.8.2002
- 30 Shraga Shoval 2001** SHRAGA SHOVAL, Johann B.: Using Coded Signals to Benefit from Ultrasonic Sensor Crosstalk in Mobile Robot Obstacle Avoidance / IEEE International Conference on Robotics and Automation. Seoul, Korea, May 21-26 2001. – Forschungsbericht
- 31 Stiller 2002** STILLER, Andreas: Die ARM-Story. In: *c't magazin für computer technik* 2 (2002), Januar, S. 70–73
- 32 Tanenbaum 1998** TANENBAUM, Andrew S.: *Computernetzwerke*. 3. revidierte Auflage. Hans-Pinsel-Straße 9b, D-85540 Haar bei München : Prentice Hall, 1998. – ISBN 3–8272–9568–8
- 33 Verts 1996** VERTS, Dr. William T.: *An Essay on Endian Order*. online. April 1996. – URL <http://www.cs.umass.edu/~verts/cs32/indian.html>. – Zugriffsdatum: 5.8.2002

Anhang A

Inhalt der CD-ROM

Die unten stehenden Angaben gehen davon aus, dass der Laufwerksbuchstabe H: das CD-ROM-Laufwerk kennzeichnet.

H:\Pdf_&_PS\	Diese Diplomarbeit als PDF- und als PS-Datei.
H:\Quellcode\	Der Quellcode, der während dieser Diplomarbeit entstand.
H:\Schaltplan\	Der Schaltplan des Interfaces im BMP- und im GIF-Format.
H:\Chipdoku\	Datenblätter für den AT91M55800A und die Transducer von Marco.

Versicherung über Selbständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §24(5) ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Ort, Datum

Unterschrift des Studenten