

## **DIPLOMARBEIT**

# **Distanzmesssysteme und Objekterkennung für autonome Rollstühle**

**ausgeführt am Institut für  
Industrielle Elektronik und Materialwissenschaften  
der Technischen Universität Wien**

unter der Anleitung von o. Univ. Prof. Dr. Michiel J. Vellekoop  
und Dipl.- Ing. Dr. Wolfgang Zagler  
als verantwortlich mitwirkendem Universitätsassistenten

durch

**JORDAN Harald  
Breite Gasse 12  
2136 Laa an der Thaya  
9825829**

**Wien, 29.09.2003**

-----  
Unterschrift:

## **Zusammenfassung**

In diesem Schriftstück wird dem Leser der Lernprozess vermittelt der notwendig war, um passend zu vorgegebenen Randbedingungen ein zuverlässiges und präzises Erfassungssystem (Objekterfassung und Distanzmessung) zu realisieren.

Schnell wird man merken, dass dazu mehrere Anläufe notwendig waren, die jeweils ihre Vorteile, jedoch auch Nachteile mit sich brachten. Dies hatte zur Folge, dass aufbauend von den gewonnenen Kenntnissen neue, verbesserte Systeme aufgebaut werden konnten.

An dieser Stelle soll noch angemerkt werden, dass für dieses Werk ein gewisses Maß an Vorwissen ( $\mu$ Computertechnik, Elektronik und Softwareentwicklung) erforderlich ist, um das eine oder andere Detail zu verstehen. Für die Erklärung elementarer Methoden und deren Anwendung ist in dieser Diplomarbeit kein Platz. Dieser wohl überlegte Schritt wurde nicht aus einer Arroganz oder plötzlichen Laune heraus beschränkt, sondern soll es dem interessierten Fachpublikum ermöglichen schnell an das gesammelte Wissen heranzukommen.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>4</b>
<b>2</b>	<b>Optische Distanzmessung</b>	<b>7</b>
2.1	Laufzeitmessung . . . . .	7
2.2	Interferenzmessung . . . . .	9
2.3	Triangulation . . . . .	10
2.3.1	Strahlungsphysikalische Größen . . . . .	10
2.3.2	Funktionsweisen / Produktvarianten . . . . .	13
2.3.3	Prinzipbedingte Möglichkeiten und Probleme der Triangulation . . . . .	18
<b>3</b>	<b>SHARP Distanzsensoren</b>	<b>20</b>
3.1	Sensoren mit digitalem Signalausgang . . . . .	21
3.1.1	Funktionsprinzip . . . . .	21
3.1.2	Auslesen der Abstandsinformation . . . . .	22
3.1.3	Kegel der Sendediode . . . . .	25
3.1.4	Beschaltung des Sensors . . . . .	27
3.2	Sensoren mit analogem Signalausgang . . . . .	28
3.2.1	Funktionsprinzip . . . . .	28
3.2.2	Auslesen der Abstandsinformation . . . . .	28
3.2.3	Dynamisches Verhalten . . . . .	30
<b>4</b>	<b>Akustische Distanzmessung</b>	<b>32</b>
4.1	Grundlagen . . . . .	32
4.1.1	Was ist Ultraschall . . . . .	32
4.1.2	Eigenschaften von Ultraschall . . . . .	32
4.1.3	Einsatzgebiete von ultraschallbasierten Einrichtungen . . . . .	34
4.2	Das Puls-Echo-Verfahren . . . . .	37
4.3	Schlussfolgerungen, Erfahrungswerte . . . . .	38

<b>5</b>	<b>Messsystem mit GP2D120</b>	<b>39</b>
5.1	Einführung und Aufgabenstellung . . . . .	39
5.1.1	Aufgabenstellungen . . . . .	39
5.1.2	Funktionsanalyse . . . . .	40
5.2	Recherche und Wahl des Sensors . . . . .	42
5.2.1	Auffistung und Erläuterung von aktuell erhältlichen Distanzsensoren . . . . .	42
5.2.2	Wahl des Sensors . . . . .	43
5.2.3	Sensoraufnahme für Profilerstellung . . . . .	46
5.2.4	Aktoren für den Schwenkantrieb . . . . .	48
5.3	Wahl der Steuereinheit . . . . .	56
5.4	Sensor-Controllerkopplung mittels Modellbauservo . . . . .	58
5.4.1	Messanordnung . . . . .	58
5.4.2	Softwarebeschreibung . . . . .	59
5.5	Sensor-Controllerkopplung mittels Schrittmotor . . . . .	63
5.5.1	Messanordnung . . . . .	63
5.5.2	Softwarebeschreibung . . . . .	64
5.6	Das Hostsystem . . . . .	67
5.6.1	Anforderungsliste an die Software . . . . .	68
5.6.2	Beschreibung der Software . . . . .	68
5.7	Eindrücke, Vergleich der Antriebssysteme . . . . .	74
<b>6</b>	<b>Messsystem nach Triangulationsprinzip</b>	<b>76</b>
6.1	Einführung und Aufgabenstellung . . . . .	76
6.1.1	Aufgabenstellungen . . . . .	76
6.1.2	Funktionsanalyse . . . . .	76
6.2	Beschreibung der hardwaregebundenen Komponenten . . . . .	78
6.2.1	Optische Sendeeinheit . . . . .	78
6.2.2	optische Empfangseinheit . . . . .	80
6.2.3	Schwenkeinrichtung . . . . .	82
6.3	Beschreibung der Software . . . . .	86
6.3.1	Beschreibung der Firmwareroutinen . . . . .	86
6.3.2	Beschreibung der Routinen des Hostsystems . . . . .	88
6.4	Bedienung der Software . . . . .	92
6.5	Bewertung der Messvorrichtung . . . . .	98

<b>7</b>	<b>Messsystem zur Bodenabtastung</b>	<b>101</b>
7.1	Einführung und Aufgabenstellung . . . . .	101
7.1.1	Aufgabenstellungen . . . . .	101
7.1.2	Funktionsanalyse . . . . .	101
7.2	Beschreibung der Komponenten . . . . .	105
7.2.1	Linienlaser . . . . .	105
7.2.2	Detektionseinheit . . . . .	106
7.2.3	Steuereinheit . . . . .	107
7.3	Beschreibung der Software . . . . .	108
7.3.1	Beschreibung der Firmwareroutinen . . . . .	108
7.3.2	Beschreibung der Routinen des Hostsystems . . . . .	108
7.4	Bedienung der Software . . . . .	110
7.5	Bewertung der Messvorrichtung . . . . .	115

# Kapitel 1

## Einleitung

Zu Beginn dieser Arbeit möchte ich einige Gedanken und einleitende Worte anführen, die das restliche Werk rechtfertigen sollen und dem Leser einige neue Methoden der Messtechnik näher bringen.

Ziel soll dabei nicht eine einzelne, für eine bestimmte Aufgabenstellung maßgeschneiderte Lösung sein. Vielmehr ist es dem Verfasser dieses Werks ein Anliegen, die grundlegenden Prinzipien und deren gezielte Anwendung zu erklären. Dies macht es möglich, offen für künftige Aufgabenstellungen zu sein und dennoch genügend Spezialwissen zu vermitteln.

Wie aus dem Titel der Diplomarbeit zu schließen ist, werden in diesem Schriftstück Methoden zur Objekterkennung und Distanzmessung für einen automatisch gesteuerten Rollstuhl behandelt.

In einer Gesellschaft, in der Werte wie Rehabilitation, Altenversorgung und Mobilität eine immer wichtigere Rolle spielen, ist es verwunderlich, dass Personen mit eingeschränkter oder fehlender Gehfähigkeit immer noch auf Hilfsmittel zurückgreifen müssen, deren Ursprung weit in die Vergangenheit reicht. Zwar wurden diese inzwischen mit diversen Hilfsantrieben wie zum Beispiel Elektromotoren ausgestattet, in anderen Punkten hat man diese Apparaturen jedoch nicht verändert. Problematisch ist dies besonders dann, wenn die zu befördernde Person zusätzlich Probleme mit der Handmotorik hat oder nicht sichergestellt werden kann, dass das geistige Vermögen ausreicht, um diese Hilfsmittel sicher<sup>1</sup> zu bedienen.

Aktuell lassen sich Rollstühle in vier große Gruppen einteilen:

**Konventionell mechanische Rollstühle:** Werden ausschließlich durch Muskelkraft betätigt.

---

<sup>1</sup>die Gesundheit der Person und das Wohl der Umwelt betreffend

**Konventionell elektrische Rollstühle:** Werden durch Elektromotoren angetrieben, wobei die Steuerung durch den/die Benutzer/in erfolgt und sich dabei Joysticks bedient. In Tabelle 1.1 [17] ist eine kurze Übersicht bezüglich der Häufigkeit gängiger Steuerungsmethoden.

Steuerungsmethode	Häufigkeit
Joystick	81%
Kopf- oder Kinnsteuerung	9%
Saug/Blas- Steuerung	6%
Andere (Fuß, Augen, ...)	4%

Tabelle 1.1: Häufigkeit der Steuerungsmethode [17]

**Semi-autonome und autonome Rollstühle:** Diese mit Sensoren bestückten Hilfsgeräte nehmen dem/der Benutzer/in die Steuerung und Navigation teilweise oder sogar ganz ab.

**Rollstühle, die aufgrund vorgegebener Daten/Symbole navigieren:**

Der große Unterschied zwischen dieser Variante und der vorher genannten ist, dass in diesem Fall der umgebende Raum bekannt sein muss und eventuell mit Erkennungsmarken versehen werden.

Wunderlich ist in diesem Zusammenhang, dass selbstfahrende Transportwagen in Produktionshallen schon seit gut zehn Jahren Einzug gehalten haben, diese Technik jedoch keinen Anklang im Bereich der Medizintechnik gefunden hat, obwohl der Bedarf nach solchen autonomen beziehungsweise semiautonomen Systemen gegeben wäre. Die kleine Statistik in Tabelle 1.2 gibt uns Auskunft über den Bedarf an Rollstühlen.

benötigtes Hilfsmittel	Häufigkeit
herkömmlicher Elektrorollstuhl	50-60%
autonome Rollstühle	20-25%
können keinen Rollstuhl verwenden	Rest

Tabelle 1.2: Bedarf an unterschiedlichen Systemen [17]

Dies wurde als Anlass genommen, am Institut für IEMW (Forschungsgruppe für Rehabilitationstechnik) der TU-Wien ein System aufzubauen, welches die Aufgabe erfüllen soll, Personen programmgesteuert zu bewegen.

Wie man schon jetzt ahnen kann, ist diese Aufgabenstellung äußerst komplex und umfangreich. Die gestellte Aufgabe kann somit sicherlich nicht durch eine einzige Diplomarbeit in einem fertigen Prototypen enden.

Ziel dieser Arbeit ist, wie schon angedeutet, die sichere Erkennung von Objekten, die sich in Fahrtrichtung befinden und eine entsprechende Distanzmessung als Navigationshilfe.

Weitere noch zu behandelnde Themenschwerpunkte sind unter anderem der Entwurf einer Benutzerschnittstelle (soweit erforderlich<sup>2</sup>) und die Realisierung einer Steuereinheit mit Navigationsfunktion.

In den nun folgenden Kapiteln werden zunächst einige der wichtigsten Messprinzipien vorgestellt und auf deren Verwendbarkeit<sup>3</sup> hin untersucht.

Der zweite Teil dieses Dokuments beschäftigt sich schließlich mit der Realisierung gebrauchsfertiger Messsysteme. Die Erklärung der einzelnen Messaufbauten erfolgt ausgehend von einem abstrakten Modell, das die Grundanforderungen abdeckt und endet im Detailbereich der Hard- und Software. Es sollen jedoch gerade hinsichtlich der Software keine Code-Listings vorgestellt werden, was Aufgabe entsprechender Programmwerkzeuge ist, sondern ein Überblick über die Struktur des jeweiligen Programms gegeben werden, um das Verständnis bezüglich der Funktionsweise der jeweiligen Applikation zu erhöhen.

---

<sup>2</sup>was kann man dem/der Benutzer/in zumuten?

<sup>3</sup>im Zusammenhang mit diesem Themenschwerpunkt

# Kapitel 2

## Optische Distanzmessung

### 2.1 Laufzeitmessung

Dieses Verfahren, welches in der englischsprachigen Literatur [1] unter der Bezeichnung *Pulsed Time-of-flight Method* zu finden ist, ist der akustischen Distanzmessung mittels Ultraschall (siehe Kapitel 4) sehr ähnlich.

Auch hier wird die Distanz aus dem Produkt Geschwindigkeit und benötigter Zeit  $t$  ermittelt, was in Gleichung 2.1 zu sehen ist.

$$s = \frac{c \cdot t}{2} \quad (2.1)$$

Der Faktor 0.5 wurde als Korrekturfaktor in die Gleichung 2.1 eingebracht, da die Laufzeit den Hin- und Rückweg abdeckt, die Distanz jedoch nur der halben Laufzeit proportional ist.

$c$  in Gleichung 2.1 entspricht dabei der Lichtgeschwindigkeit in jeweiligen Medium. Die bei diesem Messverfahren verwendeten Lichtstrahlen können dabei im sichtbaren oder im infraroten Bereich liegen.

Diese Lichtstrahlen, in der Regel wird ein Laser verwendet, müssen möglichst kurz gepulst und steilflankig sein und eine hohe Intensität besitzen. Diese Voraussetzungen machen es möglich den reflektierten Strahl auch bei großen Distanzen sicher zu detektieren und somit ein befriedigendes Ergebnis zu erhalten.

Eine schematische Übersicht über das Gesagte ist in Abbildung 2.1 zu sehen. Mit diesem Rüstzeug könnte man nun versuchen ein geeignetes Messgerät zu bauen. In den meisten Fällen wird sich dieser Versuch als schwieriges Unterfangen bei der Lösung der Probleme erweisen.

Grund dafür ist der Umstand, dass man zwar Zeiten recht gut messen kann<sup>1</sup>,

---

<sup>1</sup>mit modernen und oftmals kostengünstigen  $\mu$ Controllern oder FPGA

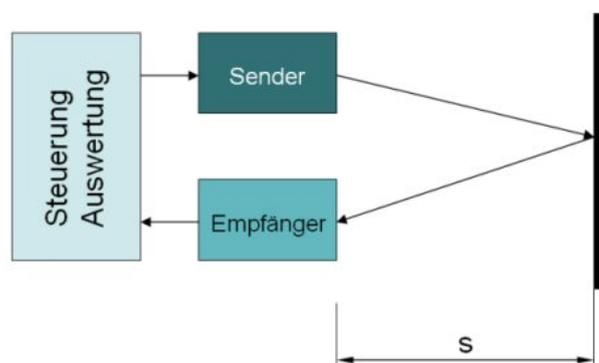


Abbildung 2.1: Schematischer Aufbau

dies gilt jedoch nur für Zeitbereiche oberhalb einer  $\mu\text{s}^2$ .

Um aber eine annehmbare Auflösung (cm) im Messbereich von einigen wenigen Metern zu erhalten, benötigt man jedoch ein Zeitraster von Bruchteilen einer  $\text{ns}^3$ , was sich mit einfachen Laborgeräten nicht mehr ohne weiteres erreichen lässt.

Eine Methode um jedoch trotzdem solch ein Messsystem mit vertretbaren Kosten und Aufwand aufzubauen nennt sich in der angloamerikanischen Literatur [1] *Modulated Beam System*.

Auch hier wird wiederum die Laufzeit des Lasers als wegproportionale Größe herangezogen. In diesem Fall wird jedoch das modulierte Signal zur Laufzeitmessung herangezogen. Wird das Signal periodisch mit einem Sinus moduliert, so ist es möglich, die Phasenverschiebung, welche sich aufgrund der endlichen Ausbreitungsgeschwindigkeit ergibt, zwischen austretendem und einfallendem Signal zu messen.

Aber auch hier ergibt sich das Problem der zu geringen Auflösung in Messbereichen im Meterbereich.

Ihre großen Vorteile können diese hier vorgestellten Verfahren oftmals nur im militärischen Bereich als Zielortungssysteme ausspielen.

Aus den oben genannten Gründen erscheint es im Kontext mit dieser Arbeit als nicht sehr sinnvoll näher auf diese Technologien einzugehen. Um der Vollständigkeit Rechnung zu tragen, ist die Entscheidung gefallen, diese Systeme zumindest peripher in dieser Arbeit anzuführen und deren Grenzen aufzuzeigen. Mehr soll und kann in diesem Werk nicht zu diesem Thema nicht angeführt werden.

---

<sup>2</sup>1  $\mu\text{s} = \frac{1}{10^6}$  s  
<sup>3</sup>1 ns =  $\frac{1}{10^9}$  s

## 2.2 Interferenzmessung

Dieses hochgenaue Messverfahren soll an dieser Stelle nur kurz erwähnt werden, da es für die nachfolgenden Arbeiten keinerlei Relevanz besitzt. Grund dafür ist, dass dieses System keine absolute Entfernung bestimmen kann, sondern nur Distanzänderungen detektiert werden können und somit immer nur Relativangaben möglich sind.

Der Effekt, der diesem Verfahren zugrundeliegt, baut auf dem Wellencharakter des Lichts auf. Dies bedeutet, dass je nach Phasenlage von zwei, entlang einer Achse laufenden Lichtstrahlen, positive oder negative (Auslöschung) Interferenz auftreten kann. Die Anzahl der Interferenzwechsel ist ein Maß für die Distanzänderung.



stimmten Emissionswinkel  $\epsilon_S$  in einem gegebenen Raumwinkel  $\Omega$  emittiert.

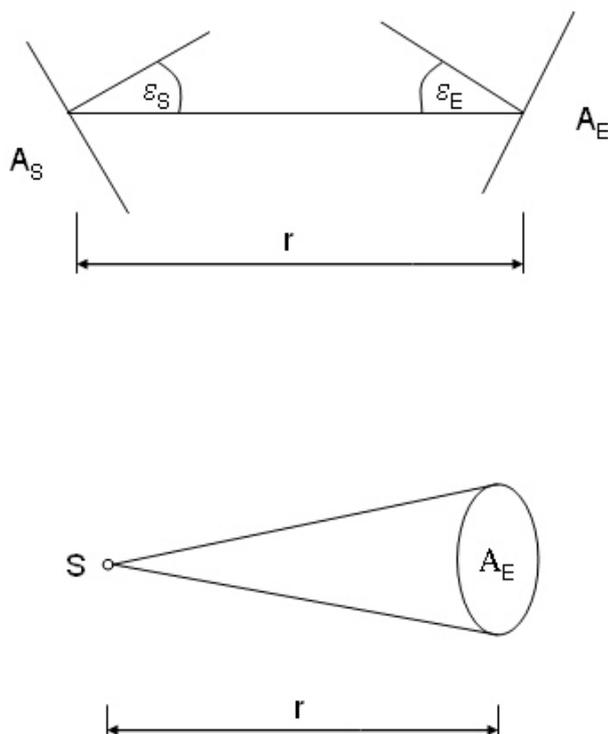


Abbildung 2.3: Skizze der wichtigsten Größen

Gleichung 2.5 (siehe [3]) stellt den Zusammenhang zwischen Strahlungsleistung  $\Phi$  und Raumwinkel  $\Omega$  dar.

Der Raumwinkel  $\Omega$  (siehe Abbildung 2.3) ist das Verhältnis aus Fläche des Empfängers und dem Quadrat des Abstandes, was in Gleichung 2.4 detailliert zu sehen ist.

$$L = \frac{\Phi}{A_S \cdot \cos(\epsilon_S) \cdot \Omega} \quad (2.3)$$

$$\Omega = \frac{A_E}{r^2} \cdot \Omega_0 \quad (2.4)$$

In den angegebenen Gleichungen 2.5 und 2.4 haben die unbekanntenen Symbole folgende Bedeutung:

$A_S$ ,  $A_E$ : Sende-, bzw. Empfängerfläche

$\varepsilon_S, \varepsilon_E$ : jeweilige Neigungswinkel zur Verbindungsgerade

Für einen Lambert-Strahler (siehe Abbildung 2.4) stellt das photometrische Grundgesetz den Zusammenhang zwischen der Strahldichte  $L$  eines Senders und der Strahlungsleistung  $\Phi$  am Ort des Empfängers her.

Zu beachten ist, dass die vom Empfänger empfangene Strahlungsleistung mit dem Quadrat des Abstandes von Sender und Empfänger abnimmt.

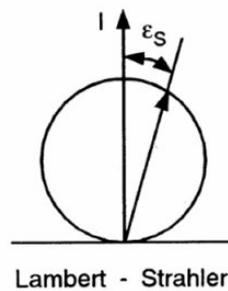


Abbildung 2.4: Lambertstrahler

$$\Phi = L \cdot \frac{A_S \cdot \cos(\varepsilon_S) \cdot A_E \cdot \cos(\varepsilon_E)}{r^2} \cdot \Omega_0 \quad (2.5)$$

### 2.3.2 Funktionsweisen / Produktvarianten

Lasertriangulation (LTS) ist ein berührungslos arbeitendes Messverfahren, mit dem der Abstand eines Punktes auf der Oberfläche eines Objektes gemessen werden kann.

Der aus dem Sender austretende Laserstrahl wird mittels einer Optik fokussiert und auf dem Messobjekt diffus reflektiert.

Als Sender werden Halbleiterlaserdioden verwendet<sup>4</sup>. Das Messobjekt muss sich im Messbereich befinden welcher durch eine Bezugsebene (meist Sensorvorderfläche) und einem Grundabstand definiert (siehe Abbildung 2.5) wird.

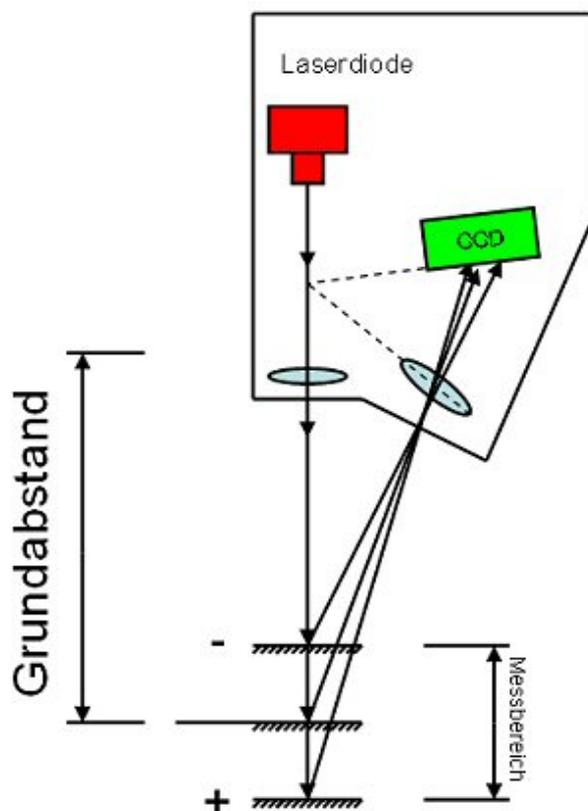


Abbildung 2.5: Mögliche Sensoranordnung

Aus dem Bündel diffus reflektierter Strahlen muss wieder genau ein Messstrahl gewonnen werden, der zur Triangulationsmessung herangezogen wird. Dies geschieht mit einer Linse. Damit der Lichtpunkt auf dem Empfänger scharf abgebildet wird, müssen sich nach der *Scheimpflugbedingung* [3] die

<sup>4</sup>typisch ist 670nm mit 1mW Leistung, Laserklasse 2

Ebenen der Abbildungsoptik und der des Empfängers in der Achse des Beleuchtungsstrahlengangs schneiden. Je nach Abstand zwischen Triangulationssensor und Messobjekt, wird der Lichtpunkt auf einer anderen Stelle am Empfänger abgebildet.

Grundsätzlich stehen drei Empfängertypen zur Auswahl:

**PSD:** Photo Sensitive Device

**CCD:** Charge Coupled Device<sup>5</sup>

**CMOS Image Sensor:** Bildsensoren<sup>6</sup> basierend auf Complementary MOS

**PSD:**

PSD (wird oft auch für Position Sensing Detector verwendet) Sensoren stellen prinzipiell eine Fotodiode mit streifenförmig gezogener, beleuchtungsempfindlicher Fläche dar.

Je nach Lage des Lichtpunktes wird ein analoges Signal ausgegeben, das der Lage des Lichtpunktes proportional ist.

Die P-Schicht ist dabei lichtempfindlich. Während die N-Schicht durchgehend metallisiert ist, befinden sich nur an den beiden Enden, der P-Schicht, die Metallelektroden  $k_1$  und  $k_2$ , was in Abbildung 2.6 zu sehen ist.

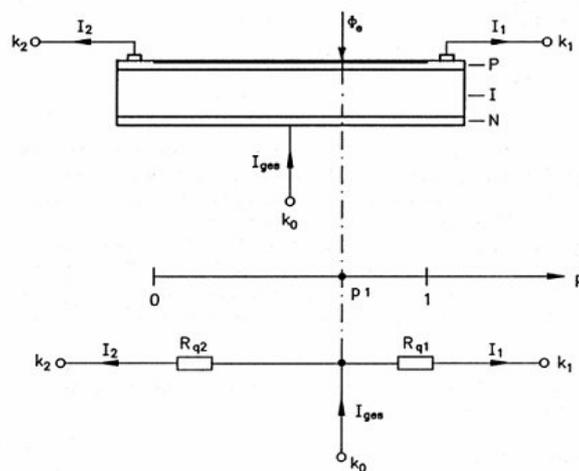


Abbildung 2.6: Aufbau eines PSD

<sup>5</sup>sowohl Zeilen als auch Matrix

<sup>6</sup>Zeilen und Matrix

Wird ein beliebiger Punkt des PSD von einem Lichtstrahl getroffen, werden dort Ladungsträger freigesetzt, die einen Stromfluss zu den beiden Elektroden bewirken. Da die unbeleuchteten Teile der Schicht wie Widerstände wirken, ist das Verhältnis der Ströme  $I_1$  und  $I_2$  von der Lage des Lichtpunktes abhängig. Das Verhältnis der Ströme ist dabei von der einfallenden Lichtmenge unabhängig, wodurch ein stabiles Entfernungsmessen, ohne Beeinflussung durch das Reflexionsvermögen eines Objekts, ermöglicht wird.

$$R_{q1} = (1 - p_1) \cdot R_q \quad (2.6)$$

$$R_{q2} = p_1 \cdot R_q \quad (2.7)$$

$$p_1 = \frac{I_1}{I_1 + I_2} \quad (2.8)$$

Wird die Position des Lichtstrahls allerdings von der Mitte des Sensors aus gemessen, so ist dieser Abstand proportional zu  $\frac{I_1 - I_2}{I_1 + I_2}$ .

PSD-Sensoren zeichnen sich durch eine einfache Beschaltung und Dimensionierung aus. Grundsätzlich ist keine Rasterung (wie bei CCD-Sensoren) möglich und der am Empfänger auftreffende Lichtpunkt ist nicht beliebig stark zu fokussieren. Daraus resultiert, dass der auftreffende Lichtpunkt immer einen gewissen Durchmesser besitzen muss. Dies kann zu einer verringerten Auflösung und zu Messfehlern führen.

Die Bestimmung der Lichtpunktmitte und damit der Messobjektposition erfolgt beim PSD-Detektor anhand der Verteilung der Strahlungsenergie über die gesamte Fläche des in das Empfangselement eintretenden Lichtpunkts. Die Oberflächenbeschaffenheit des Messobjekts wirkt sich auf diese Strahlungsenergieverteilung so aus, dass Messwertabweichungen auftreten.

Der Vorteil dieses Empfängertyps ist, wie schon erwähnt die Möglichkeit einer einfach zu realisierende Messelektronik und seine hohe Geschwindigkeit (bis zu 10MHz). Nachteile sind das geringe Auflösungsvermögen welches sich im  $\mu\text{m}$ -Bereich befindet. Zudem können Störeinflüsse nicht festgestellt werden (bei CCD Sensoren möglich).

Eine Messanordnung mit PSD ist in Abbildung 2.7 zu sehen.

### CCD:

Ein CCD-Photodetektor transformiert hingegen eine örtliche Strahlungintensität in eine zeitlich veränderliche elektrische Spannung.

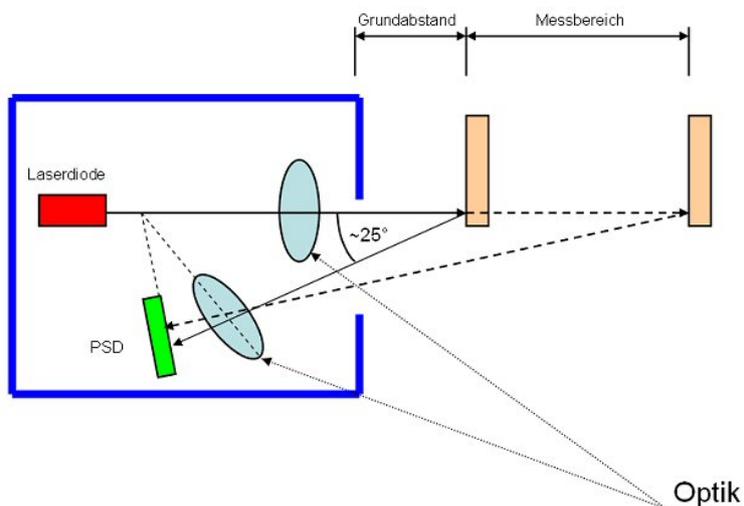


Abbildung 2.7: Mögliche Sensoranordnung mit PSD

Während der Belichtungszeit werden Photoelektronen gesammelt und anschließend über ein Transfergatter und ein Schieberegister an den Ausgangsverstärker geleitet. Dort werden die Ladungspakete der einzelnen Photosensoren in analoge Spannungen umgewandelt, die am Ausgang des Sensors zur Verfügung stehen. Eine solche Anordnung ist in Abbildung 2.8 zu sehen.

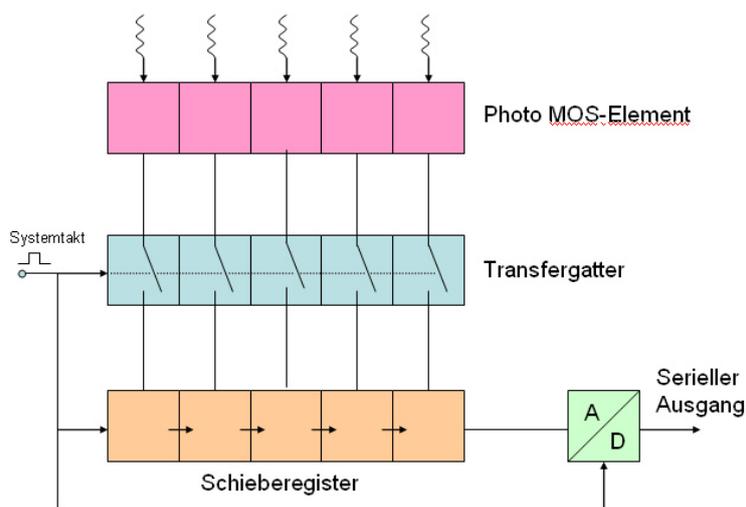


Abbildung 2.8: Aufbau eines CCD Zeilensensors

Es entsteht somit ein zeitlich diskretisiertes Ausgangssignal, aus dem die Position des Lichtpunkts ermittelt werden kann. Im Gegensatz zum PSD-Sensor ermöglicht der CCD-Sensor ein genaues Aus-

werten der Strahlungsenergieverteilung. Damit lässt der CCD-Sensor eine zuverlässige und hochpräzise Wegmessung, unabhängig von der Strahlungsenergieverteilung über die Lichtpunktfläche, zu. Dies setzt aber eine aufwändige Signal- und Informationsverarbeitung voraus, die ihrerseits wieder beherrscht werden will.

In LTS werden sogenannte CCD-Zeilen eingesetzt, die seriell ausgelesen werden<sup>7</sup> und in vielen Fällen kaskadierbar sind, um teure und aufwändige Optiken hinfällig machen. Als Beispiele von Anwendungsgebieten solch einer Technologie sind Scanner oder moderne, digitale Kopiergeräte.

Das serielle Auslesen ist auch die Ursache für die im Vergleich zum PSD kleineren Verarbeitungsgeschwindigkeit die, fair betrachtet, immer noch im MHz Bereich liegt und für sehr viele Anwendungen ausreicht.

LTS mit CCD-Sensoren zeichnen sich im Allgemeinen durch eine höhere Auflösung (Zehntel  $\mu\text{m}$ -Bereich) und über einen weiten Messbereich aus.

### CMOS Bildsensor:

Die CMOS Bildsensoren unterscheiden sich nur technologisch, haben aber grundsätzlich die gleichen Anwendungsgebiete und in einem gewissen Bereich ähnliche Eigenschaften wie CCD-Sensoren.

Besonders positiv hervorzuheben ist die Möglichkeit, den Sensor mittels Si-Planartechnik<sup>8</sup> mit vorverarbeitenden digitalen Schaltkreisen zu kombinieren.

Dies ermöglicht es dem Hersteller, hochintegrierte, leistungsfähige Bauteile kostengünstig anzubieten.

Es ist daher leicht einzusehen, dass diese Bauteile sowohl alt hergebrachte Methoden zum Auslesen der Bilddaten unterstützen (siehe CCD), als auch hoch moderne Übertragungsprotokolle und oftmals auch Bildbearbeitungsalgorithmen implementiert haben, um ein zeit- und kostensparendes Hardware-Design mit diesen Bausteinen zu ermöglichen.

---

<sup>7</sup>man spricht auch von einem analogen Schieberegister

<sup>8</sup>was üblicherweise auch gemacht wird

### 2.3.3 Prinzipbedingte Möglichkeiten und Probleme der Triangulation

In diesem Abschnitt werden einige Punkte angesprochen, die das Einsatzgebiet mehr oder weniger stark vorgeben, denn es hat sich gezeigt, dass die LTS sich nicht in allen (jedoch sehr vielen) Bereichen bewährt haben.

#### Nichtlinearität der Kennlinie

Ein wesentlicher Punkt ist das stark nichtlineare Verhalten solcher Sensoren. Ein beispielhafter Verlauf solch einer Kennlinie ist in Abbildung 2.9 zu sehen.

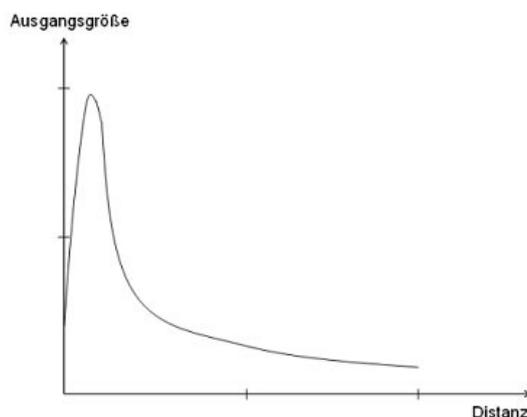


Abbildung 2.9: Möglicher Kennlinienverlauf bei Triangulation

Wie man in Abbildung 2.9 erkennen kann besitzt diese Kennlinie bei kleinen Messwerten<sup>9</sup> eine große Steigung, was eine hohe Messwertauflösung zur Folge hat.

Messwerte am anderen Ende der Messwertskala hingegen weisen eine verminderte Messwertauflösung aufgrund der geringen Steigung der Kennlinie auf. Vereinfacht ausgedrückt kann man sagen, dass Objekte im Fernbereich mit geringerer Genauigkeit erfasst werden können als Objekte im Nahbereich. Dies mag zwar zunächst etwas vernichtend klingen, es gibt jedoch sehr viele Anwendungen (zum Beispiel Näherungssensoren, ...) bei denen gerade dieses Verhalten gewünscht wird, beziehungsweise keinen Einfluss auf den Rest des Systems hat.

<sup>9</sup>klein in Bezug auf den Messbereich

### Abschattung

Bei allen Triangulationssensoren muss immer darauf geachtet werden, dass der Laserpunkt von der Empfängeroptik direkt gesehen werden kann, und dass sich keine Hindernisse vor der Empfängeroptik befinden.

Dies mag zwar zunächst als selbstverständlich erscheinen, es gibt jedoch einige Situationen (siehe Abbildung 2.10) in denen solch eine Abschattung des Laserstrahls erfolgt, ohne dies vorher vermutet zu haben.

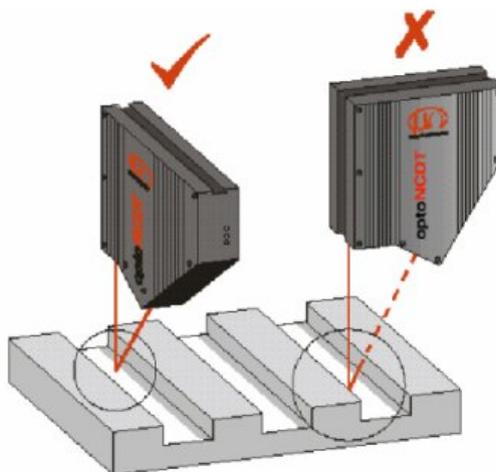


Abbildung 2.10: Abschattung bei Triangulation

Es ist also immer genau darauf zu achten, dass der direkte Sichtkontakt zwischen Laserpunkt und dem Sensorelement gegeben ist.

Ist dies trotz einer Neuordnung der Komponenten nicht möglich, so muss auf dieses Verfahren verzichtet werden und es ist ein alternatives Messsystem zu installieren.

### Winkelabhängigkeit

Optimale Messergebnisse setzen ein rechtwinkliges Auftreffen des Laserstrahls auf die Messoberfläche voraus.

Bei Verkippung unterliegt die Messung einer Winkelabhängigkeit die selbstverständlich nicht zu tolerieren ist.

Dies lässt sich wie folgt erklären. Da der Laserpunkt je nach Schräglage der Reflektorfläche mehr oder weniger zu einer kurzen Linie verschleift und somit eine Unsicherheit in der Auswerteeinheit hervorruft, wirkt sich dies negativ auf das Ergebnis aus.

# Kapitel 3

## SHARP Distanzsensoren

Dieser Abschnitt beschäftigt sich mit der Erkenntnissammlung bezüglich der Sensorreihe zur absoluten Distanzmessung der Firma *SHARP* [7].

Diese, bereits seit einigen Jahren erhältlichen Sensoren, wurden ursprünglich für den Einsatz im Sanitärbereich als Näherungsschalter (zum Beispiel zum Einschalten des Händetrockners) konzipiert.

Firmenseitig ist man schließlich dazu übergegangen ein Redesign anzustreben und bietet diese Sensoren heute als kompakte Baueinheiten zur Distanzmessung für den allgemeinen Gebrauch an.

Besonderer Beliebtheit erfreut sich dieser Sensortyp in den Bereichen der Robotik<sup>1</sup> und im Kfz-Bau als Einparkhilfen sowie für Anwendungen im Umfeld der industriellen Automation.

Diese Sensoren gibt es in verschiedenen Ausführungen, die sich durch die Signalweiterleitung, den Messbereich und ihre Funktion unterscheiden.

Betrachten wir zunächst die Signalweiterleitung, die auf folgende zwei Arten erfolgen kann.

**digital:** Distanz wird als 8bit-Zahl übertragen.

**analog:** Distanz wird durch eine analoge Spannung repräsentiert.

Wenn im obigen Text steht, dass sich einige Sensortypen durch die Funktion unterscheiden, so ist nicht das Messprinzip betroffen, sondern lediglich die Art der Signalausgabe welche je nach verwendetem Sensortyp variieren kann.

---

<sup>1</sup>sowohl im professionellen als auch im semiprofessionellen Bereich

**kontinuierlicher Messwert:** Es wird immer der aktuelle Messwert ausgegeben.

**Schwellwertmessung:** Der Sensor besitzt eine adjustierbare Schaltschwelle, die als Kriterium für die Ausgangsgröße herangezogen wird. Unterhalb dieser Schaltschwelle hat der Ausgang den logischen Pegel *Low*, während oberhalb derselben, der Ausgang den logischen Pegel *High* annimmt.

In den nächsten Abschnitten wird nun auf den Sensoraufbau näher eingegangen, bevor die Signalverarbeitung in Angriff genommen werden kann.

## 3.1 Sensoren mit digitalem Signalausgang

### 3.1.1 Funktionsprinzip

Das Herz des Sensors, dessen Interior in Abbildung 3.1 zu sehen ist, bildet ein PSD-Element, das hinter einer optischen Linse angeordnet ist (siehe [6]).

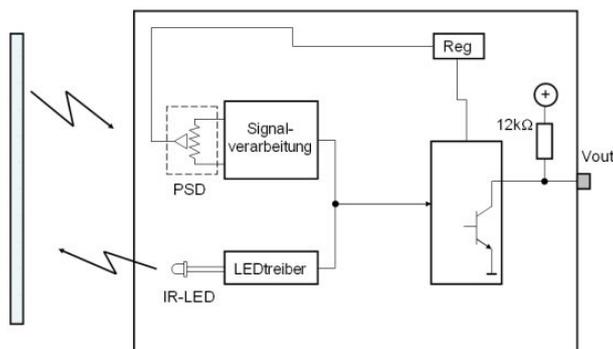


Abbildung 3.1: Blockschaltbild des Sensors

Das Messprinzip beruht auf den Gesetzmäßigkeiten der optischen Triangulation, welches schon im Abschnitt 2.3 erläutert wurde.

Auch wenn die in den folgenden Passagen angeführten Methoden sich mit denen aus dem Abschnitt 2.3 decken, so geschieht dies nur, um ein besseres Verständnis für den Sensor aufbauen zu können und bitte daher um Verständnis für etwaige Wiederholungen.

Der Sender emittiert einen durch eine Präzisionsoptik scharf gebündelten Lichtstrahl, der vom Messobjekt reflektiert wird.

Der Winkel des reflektierten Signals variiert mit dem Abstand zwischen Sensor und Messobjekt, was in Abbildung 3.2 zu sehen ist.

Das Licht passiert im Empfänger wieder eine Optik, die den Strahl punktförmig auf ein Fotoelement (PSD) fokussiert und eine elektrische Reaktion hervorruft. Eine Änderung des Abstandes zwischen Messobjekt und Sensor führt nun zu einer Verschiebung des Lichtflecks auf dem Fotodetektor und daher zu einer Änderung der elektrischen Größe.

Dieses positionsabhängige, schwache Ausgangssignal wird verstärkt, digitalisiert und über eine Schnittstelle zur Auswerteeinheit übertragen.

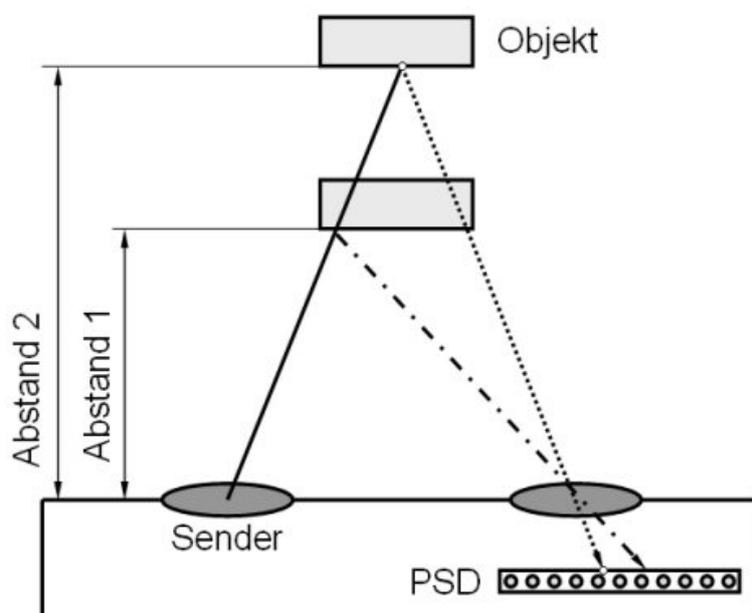


Abbildung 3.2: Abstandsmessung durch Triangulation

### 3.1.2 Auslesen der Abstandsinformation

Um Daten auslesen zu können, muss zunächst noch auf eine geeignete Stromversorgung und dem korrekten Anschluss selbiger geachtet werden, worauf in Abschnitt 3.1.4 noch näher eingegangen wird.

Der Sensor wird ausgelesen, indem der Eingangsspannung  $V_{in}$  für mindestens 70 ms auf dem Low-Pegel gehalten wird.

In dieser Zeit werden von der Infrarotsendediode 16 Messdurchläufe durchgeführt, die anschließend gemittelt werden, um so einen genauen Wert zu erhalten.

$V_{in}$  wird anschließend acht Mal getaktet und mit jeder fallenden Flanke kann ein Bit des digitalen Messwertes vom Objektabstand am Pin  $V_{out}$  übernommen werden.

Eine Erkenntnis die jetzt schon gewonnen werden kann ist, dass der Sensor immer von außen angeregt werden muss, und daher immer externe Komponenten wie Schaltwerke oder noch besser  $\mu$ Controller benötigt.

Abbildung 3.3 zeigt das entsprechende Timingdiagramm eines vollständigen Arbeitszyklus (Initiierung, und Datenauslesen).

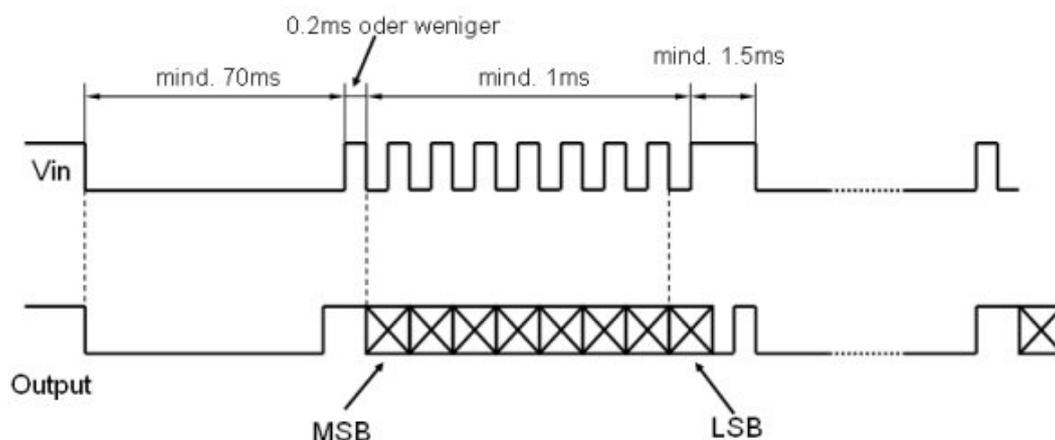


Abbildung 3.3: Timingdiagramm für das Auslesen der Daten

Prinzipbedingt (siehe Abschnitt 2.3.3) ist der Zusammenhang zwischen digitalem Wert und tatsächlichem Abstand nicht linear.

Die Auflösung liegt im Nahbereich bei etwa 1cm und fällt mit zunehmender Entfernung auf etwa 10cm ab.

Dieses Fehlverhalten wird letztlich auch durch den relativ geringen Abstand von Sender und Empfänger innerhalb des Gehäuses provoziert.

Abbildung 3.4 zeigt exemplarisch den Zusammenhang zwischen dem Messwert (dezimale Zahlendarstellung) und dem tatsächlichen Abstand.

Von besonderem Interesse sind die Bereiche unterhalb etwa 10cm und Distanzen ab ungefähr 80cm.

Im ersten Fall steigt der Messwert mit abnehmender Entfernung wieder an. Diesem Umstand muss in der Auswerteelektronik auf jedem Fall Rechnung getragen werden, denn andernfalls können große Probleme durch falsch interpretierte Messwerte auftreten.

Der zweite Fall beschreibt das Verhalten der Kennlinie bei großen Distanzen. Hier ist der Anstieg der Kurve so gering, dass eine Messung mit einer vernünftigen Auflösung praktisch nicht mehr möglich ist.

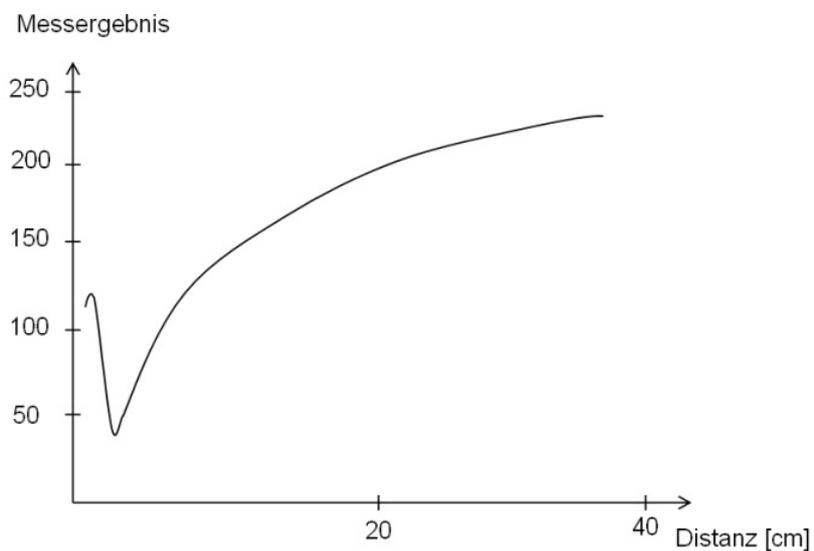


Abbildung 3.4: Kennlinie des Sensors

In diesem Zusammenhang sollte noch erwähnt werden, dass die in Abbildung 3.4 gezeigte Kennlinie nur für ein bestimmtes Reflexionsmedium gilt. Werden andere (meist keine speziellen Reflexionsmedien wie zum Beispiel Wände oder sonstige Gegenstände in Räumen) Medien verwendet, so ist damit zu rechnen, dass die Kennlinie einen anderen Verlauf (eventuell noch schlechter im Fernbereich) aufweisen wird als erwartet.

### 3.1.3 Kegel der Sendediode

Ein weiteres wichtiges Kriterium für die Bewertung von Distanzsensoren<sup>2</sup> ist die Geometrie der Sendewelle.

Mit Sendewelle wurde leider kein präziser Ausdruck gewählt, soll aber zum Ausdruck bringen, dass sich Schallwellen genauso von einer punktförmigen Quelle kegelig ausbreiten können wie dies bei Lichtstrahlen der Fall ist.

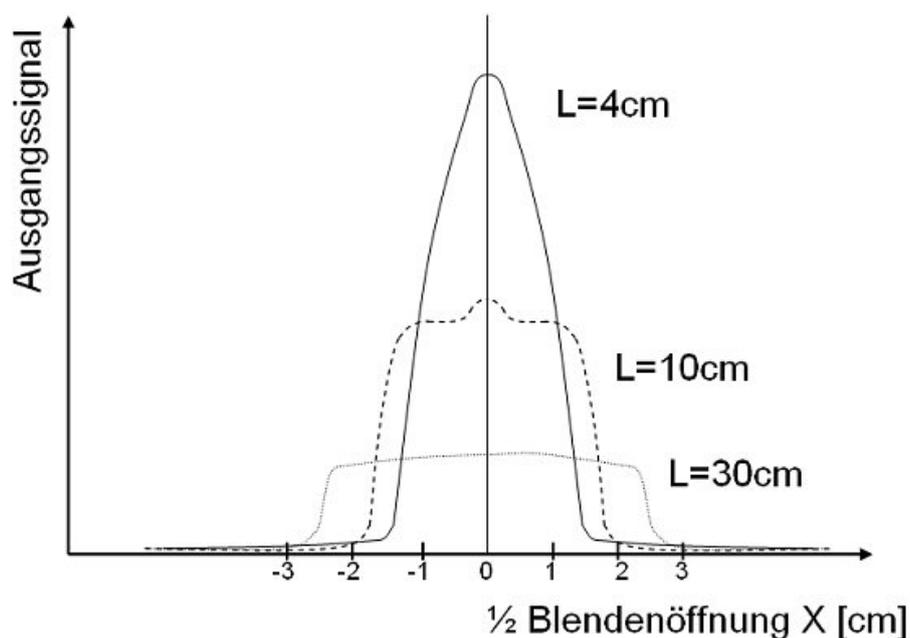


Abbildung 3.5: Ausgangsgröße in Abhängigkeit des Detektionsabstandes

Das in Abbildung 3.5 gezeigte Kennlinienfeld kann in dieser Sache für Klarheit sorgen.

Auf der Abszisse ist dabei der Detektionsabstand (Größe der Blende) aufgetragen, während die Ordinate den möglichen Ausgangswert darstellt. Es wurde in diesem Schaubild bewusst auf konkrete Zahlenwerte verzichtet, da dieses Kennlinienfeld in ähnlicher Form für alle Distanzsensoren der Firma SHARP Gültigkeit besitzt und somit allgemeiner gehalten werden kann.

Als weiterer Parameter ist der Objektabstand  $L$  gegeben.

Zur Verdeutlichung der gegebenen Abstände ist Abbildung 3.6 zu Rate zu ziehen.

<sup>2</sup>gilt sowohl für optische als auch akustische Distanzsensoren

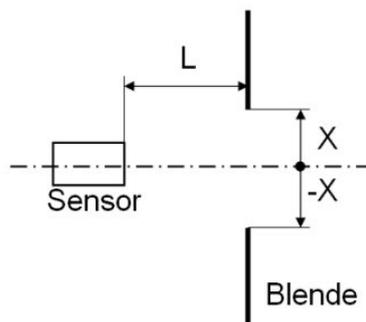


Abbildung 3.6: Blendensystem zur Kennlinienfeldermittlung

Doch kehren wir zurück zur Abbildung 3.5. Dieses Schaubild gibt recht deutlich Aufschluss über den ausgesendeten Lichtkegel. Bei völlig geschlossener Blende ( $\pm X = 0$ ) ist der Maximalwert der jeweiligen Kennlinie erreicht. Links und rechts davon fällt die Kurve mehr oder weniger stark ab.

So wirkt sich eine Blendenverschiebung bei großen Objektabständen  $L$  im Bereich um den Nullpunkt weit weniger stark auf das Ergebnis aus als selbiges im Nahbereich, wo es passieren kann, dass der Lichtstrahl ungehindert die Blende passiert und nicht mehr (beziehungsweise am falschen Objekt) reflektiert wird.

Recherchen haben ergeben, dass dieser Lichtkegel für Distanzsensoren der Firma *SHARP* [7] mit einer maximalen Messdistanz von 30cm einen Öffnungswinkel von ca.  $80^\circ$  hat. Bei Sensoren mit einer Reichweite von 2m liegt dieser Wert bei etwa  $30^\circ$ .

### 3.1.4 Beschaltung des Sensors

Die Sensorreihe mit digitalem Ausgang ist zwar für eine Betriebsspannung im Bereich zwischen 4.4V und 7V ausgelegt, der Anschluss  $V_{in}$  darf jedoch nur mit maximal 3V versehen werden.

Es hat sich in der Praxis als günstig<sup>3</sup> erwiesen, eine Schutzdiode zu verwenden, was in Abbildung 3.7 zu sehen ist. Die Anschlussbelegung kann der Abbildung 3.8 entnommen werden.

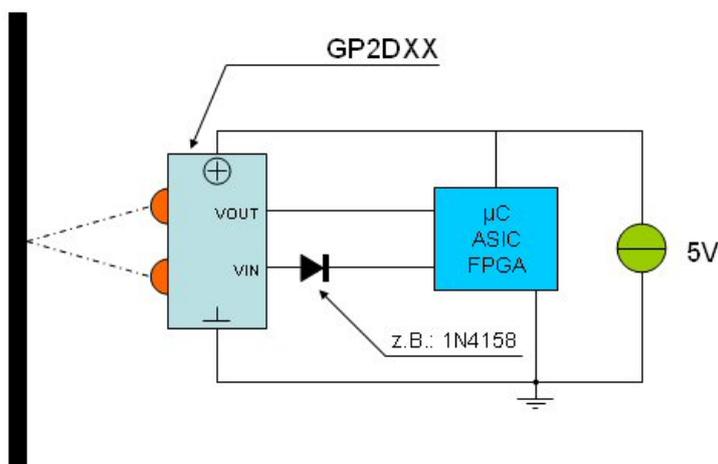


Abbildung 3.7: Beschaltung des GP2DXX

Aufgrund der geringen Ansprüche an die Versorgungseinheit können hierfür kostengünstige Festspannungsregler oder Zenerdioden verwendet werden beziehungsweise ist es auch möglich, diese Sensoren mit Ausgangspins von  $\mu$ Controllern zu betreiben<sup>4</sup>.

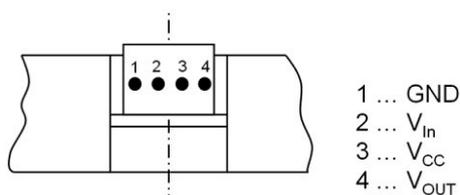


Abbildung 3.8: Pinbelegung

<sup>3</sup>einerseits betreffend den Aufwand, und andererseits die Betriebssicherheit

<sup>4</sup>es ist aber vorher genau zu prüfen ob der jeweilige  $\mu$ Controller dafür ausgelegt ist

## 3.2 Sensoren mit analogem Signalausgang

### 3.2.1 Funktionsprinzip

Auch für diese Sensorfamilie gilt das in Abschnitt 3.1.1 ausführlich skizzierte Prinzip.

Der große Unterschied liegt darin, dass das interne, verstärkte Signal direkt zum Ausgang geleitet wird und nicht noch eine ADC-Stufe durchläuft.

Die Vorteile, die sich daraus ergeben liegen auf der Hand:

- Für einfache und billige Anwendungen wird keine zusätzliche Digitaltechnik benötigt.
- Diese Sensoren weisen eine höhere Dynamik auf, was es ermöglicht ca. doppelt so hohe Abtastraten zu verwenden.

### 3.2.2 Auslesen der Abstandsinformation

Wie im vorangehenden Abschnitt 3.2.1 schon angesprochen, liegt hier der Messwert in Form eines analogen Spannungssignales vor.

Diese Ausgangsspannung liegt je nach gemessener Distanz in einem Bereich zwischen 0.5V und 2.9V, wobei der in Abbildung 3.9 gezeigte Zusammenhang zwischen gemessener Distanz und Ausgangsspannung besteht.

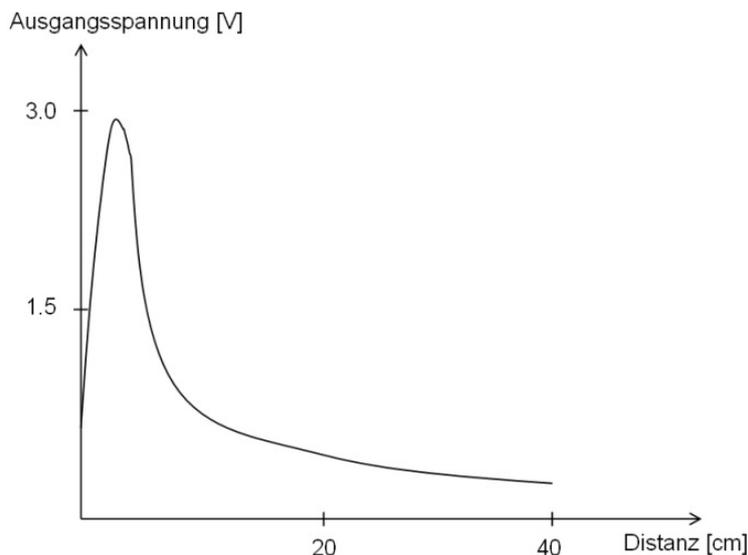


Abbildung 3.9: Kennlinie des analogen Distanzsensors

Die Kennlinie weist ein ausgeprägtes Maximum bei etwa  $4\text{cm}^5$  auf und fällt links und rechts davon mehr oder weniger stark ab.

Weitere charakteristische Punkte liegen bei einer Distanz von  $0\text{cm}$  (Verdunkelung des Sensors) oder einer zu messenden Distanz größer  $40\text{cm}$ . Hier beträgt die Spannung am Ausgang ca.  $0.5\text{V}$ .

Dazwischenliegend verläuft die Kennlinie (besonders im rechten Bereich) nichtlinear.

Für analoge Schaltungen erscheint diese Kennlinie doch etwas zu nichtlinear um brauchbare Ergebnisse damit erzielen zu können<sup>6</sup>. Eine Ausnahme stellen Sensoren mit Schaltschwelle dar, denn bei diesen Sensoren ist die Kennlinie für den Benutzer transparent und ohne Bedeutung.

Digitale Schaltungen haben, wie schon angedeutet, den großen Vorteil, mittels geeigneter Softwaremethoden (Tabellen, Teach-In Verfahren, mathematische Formulierung der Kennlinie) immer exakte Ergebnisse liefern zu können und sollten daher verstärkt eingesetzt werden.

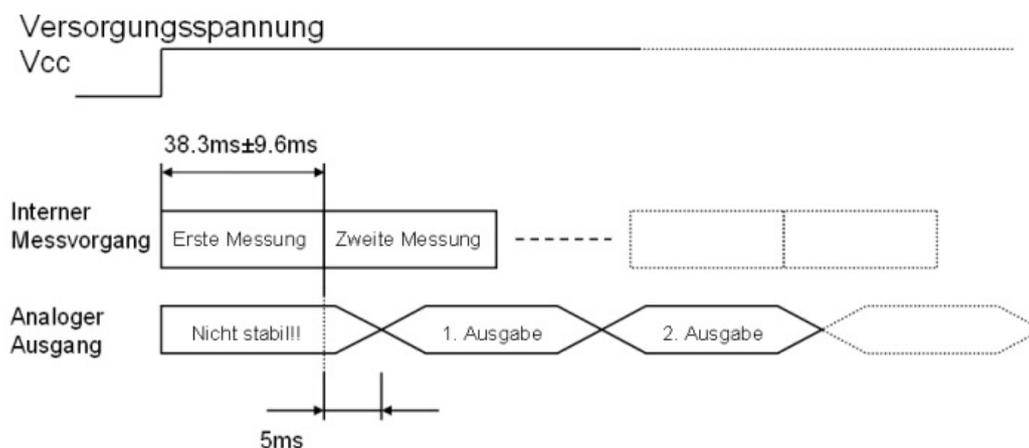


Abbildung 3.10: Zeitverhalten des analogen Distanzsensors

Aufgrund der internen Logik und Schaltverzögerungen benötigt der Sensor eine bestimmte Zeit bis sich der Ausgangspegel auf einen stationären Wert einpendelt.

Es ist also immer darauf zu achten, dass das vorgeschriebene Sensortiming eingehalten wird. Ein exemplarischer Betrieb bezüglich der zeitlichen Abfolge einer solchen Einheit ist in Abbildung 3.10 zu sehen.

<sup>5</sup>bei Sensortypen mit höherem Messbereich ist dieser Wert natürlich anders

<sup>6</sup>Knickzugnäherung mit Dioden ist möglich, in diesem Fall jedoch sehr aufwändig

### 3.2.3 Dynamisches Verhalten

Es wurde nun schon öfters erwähnt, dass diese Sensoren einen höheren Dynamikbereich aufweisen, doch diese Aussagen beruhen immer auf den aktuellen Datenblattangaben.

Im Rahmen dieser Arbeit wurde dieses Verhalten überprüft und soll zunächst kommentarlos präsentiert werden.

Die Messung erfolgte mittels einer provisorischen Vorrichtung, die aus einem Elektromotor und einer Scheibe mit angebrachten Distanzkörpern besteht. Diese Vorrichtung wurde nun so vor dem Distanzsensor platziert, dass sich je nach Winkellage der Scheibe eine unterschiedliche, zu messende Distanz einstellt.

Von besonderem Interesse sind in diesem Fall die Abhängigkeit des Ausgangssignales von der Drehzahl bzw. das Rauschverhalten des Distanzensors.

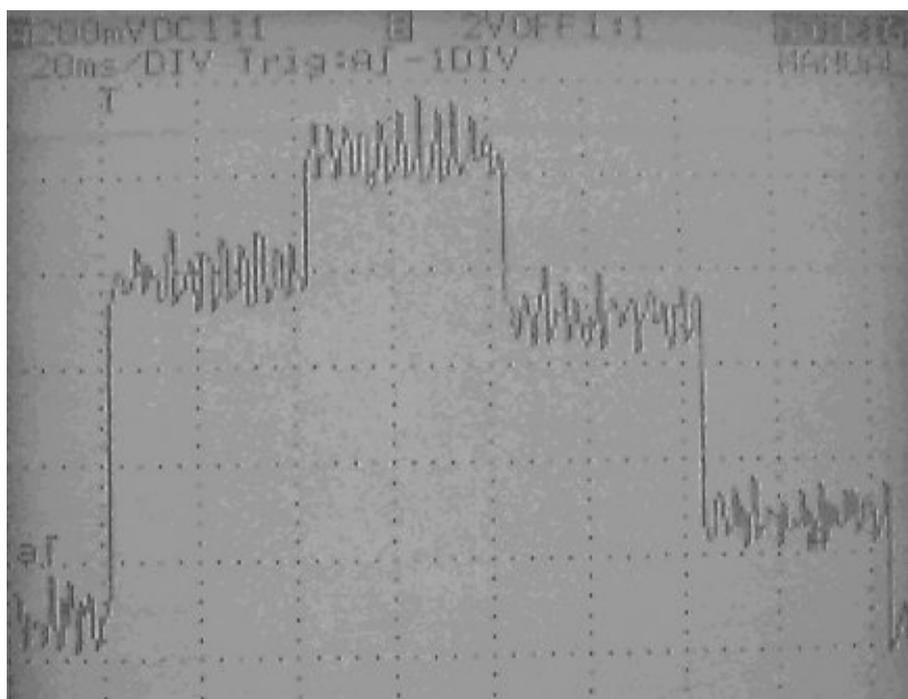


Abbildung 3.11: Verhalten bei zeitlich schnell veränderlichen Messbereichen

Abbildung 3.11 ist ein exemplarisches Abbild eines Oszilloskopschirmes dieser Messanordnung. Deutlich zu erkennen ist schon jetzt das doch relativ

große Rauschen dieses Sensors, welches in einem Bereich von  $\pm 100\text{mV}$  liegt. Gleich vorweg soll hier angemerkt werden, dass diesem Umstand Rechnung getragen werden muss, sei es durch Mittelwertbildung oder anderen Maßnahmen. Es würde sich ohne eine Korrektur eine hohe Messwertunsicherheit ergeben, was gerade im Mittel- und Fernbereich<sup>7</sup> fatale Folgen haben könnte.

Der zweite Punkt, der zu erkennen ist, betrifft das Umschaltverhalten des Sensors bei Distanzänderungen. Hier muss zunächst gesagt werden, dass dieses Ergebnis mit den Werten aus dem Datenblatt übereinstimmen. Es hat sich auch gezeigt, dass diese minimale Einstellzeit auch unterschritten werden kann. Hierzu ist jedoch anzumerken, dass dann aufgrund von Exemplarstreuungen nicht garantiert werden kann, dass unterschiedliche Sensoren immer noch ein vergleichbares Verhalten aufweisen und somit empfohlen wird, die geforderte Einstellzeit auch tatsächlich einzuhalten.

---

<sup>7</sup>Führt man sich die Kennlinie vor Augen, so erkennt man, dass gerade im Mittelbereich der Kennlinie eine kleine Änderung der Spannung einen merklich anderen Messwert ergibt

# Kapitel 4

## Akustische Distanzmessung

### 4.1 Grundlagen

#### 4.1.1 Was ist Ultraschall

Als Ultraschall bezeichnet man Schallwellen, deren Frequenz oberhalb des menschlichen Hörvermögens also oberhalb von 20 kHz liegt.

Das Prinzip der Ultraschallortung beziehungsweise Ultraschallnavigation ist der Natur entnommen. Fledermäuse beispielsweise benutzen es zur Orientierung, was es ihnen ermöglicht auch bei völliger Dunkelheit auf Nahrungssuche gehen zu können.

Fledermäuse senden Schall mit einer Frequenz von 40 bis 90 kHz (siehe [5]) und können durch das zurückgeworfene Echo feststellen, ob Hindernisse wie Bäume oder Felsen in der Nähe sind. Das Auflösungsvermögen dieser einzigartigen Messeinrichtung ist aber noch deutlich höher [18]. Auf diese Weise können sie sogar Insekten oder feine Drähte (zum Beispiel Maschendrahtzäune) orten.

#### 4.1.2 Eigenschaften von Ultraschall

Die Absorption steigt mit dem Quadrat der Frequenz. In größerem Abstand zur Quelle können nur noch tiefe Frequenzen wahrgenommen werden. Die Luft absorbiert den Rest.

Bei Frequenzen um 3kHz (für Ultraschallbetrachtung also ohne Bedeutung) kann die Luft die gleiche Absorbierung verursachen wie Wände. Die Beugung ist ebenfalls Frequenzabhängig. Die tiefen Frequenzen werden an Hindernisse kaum gebeugt. Die Reflexion von Schall kann wie mit Licht oder das Stoßen beim Billard verglichen werden (Einfallswinkel = Ausfallswinkel).

Materietype	Medium	Ausbreitungsgeschwindigkeit
Gase (Luftschall)	Luft bei 20°C	343,8 m/s
	Luft bei 15°C	340,0 m/s
	Luft bei 0°C	331,8 m/s
	Luft bei -20°C	319,3 m/s
Flüssigkeiten (Flüssigkeitsschall)	Wasserstoff (flüssig)	1305 m/s
	Wasser	1480 m/s
Festkörper (Körperschall)	Gummi	50 m/s
	Stahl	5000 m/s

Tabelle 4.1: Tabelle der wichtigsten Ausbreitungsgeschwindigkeiten

Hierzu muss die Reflexionsfläche *akustisch glatt* sein, d.h. eine unverputzte raue Wand ist für ein 8kHz Ton (Wellenlänge = 4,25cm) sehr rau und für ein 50Hz Ton (Wellenlänge = 6,8m) Spiegel glatt. Ultraschall kann sich in Gasen, Flüssigkeiten und Festkörpern ausbreiten. Die Geschwindigkeit mit der sich Schallwellen im Raum ausbreiten ist abhängig von der Materie und ist somit nicht immer gleich schnell.

Die markanten Schallgeschwindigkeiten für einige Materialien sind der Tabelle 4.1 [5] zu entnehmen.

### 4.1.3 Einsatzgebiete von ultraschallbasierten Einrichtungen

Abhängig von der Frequenz und/oder der Leistung wird Ultraschall in vielen Gebieten eingesetzt. Grob kann man Ultraschalltechnik nach Kleinsignalanwendungen (Diagnostik, Materialprüfung) und Leistungsschallanwendungen, welche z.B. für die Ultraschall-Reinigung relevant ist, aufteilen.

**Kleinsignalanwendungen:** Leistung  $< 1 \text{ W/cm}^2$ , Frequenz  $> 100 \text{ kHz}$

**Leistungsschallanwendungen:** Leistung  $> 1 \text{ W/cm}^2$ , Frequenz  $< 100 \text{ kHz}$

Ein Beispiel für die Kleinsignalanwendungen ist die Materialprüfung.

Hier wird der Ultraschall in einem Bereich von 0.25 bis 10 MHz verwendet. Damit kann die Oberfläche beziehungsweise das Bauteilinnere auf Materialfehler wie Risse, oder Lunker (beim Metallgießen entstandene Fehlstellen) in Millimeterbereichen untersucht werden. Außerdem kann an unzugänglichen Stellen die Materialdicke gemessen werden. Hohen Anklang hat dieses Verfahren in der Pipelineindustrie gefunden, denn damit ist es möglich, Rohranlagen kostengünstig zu überprüfen<sup>1</sup>.

Mit höheren Frequenzen (bis maximal 120 MHz) lassen sich auch spezielle Gefügeuntersuchungen durchführen. Für Abstandsmessung werden hingegen Frequenzen ab zirka 35kHz eingesetzt (Abstandswarner - Einparkhilfen, Abstandsmessgeräte für Innenräume). Zur medizinischen Bildgebung (Schnittbildtechnik) werden Frequenzen zwischen 2 MHz und 50 MHz verwendet.

Der Großteil der diagnostischen Fragestellungen (innere Organe- oder Schwangerschaftsuntersuchung) wird mit Frequenzen aus den Bereichen 3,5 MHz, 5,0 MHz und 7,5 MHz abgedeckt und stellt in diesem Bereich einen Quasistandard dar.

Spezielle Anwendungen wie zum Beispiel dermatologische Ultraschallbildgebung verwenden Frequenzen bis zu 50 MHz.

Zur Reinigung wird Ultraschall im Allgemeinen im Bereich von 20 kHz - 4 MHz und einer Leistungsdichte von 5 bis 20 Watt pro Liter (der Reinigungsflüssigkeit) verwendet. Dabei haben sich zwei Frequenzbereiche besonders bewährt: 20...40 kHz und 400...800 kHz (oft auch als Megaschall bezeichnet).

Zur Reinigung mit Ultraschall wird der Effekt der Kavitation ausgenutzt, welcher durch intensive Ultraschallwellen in Flüssigkeiten hervorgerufen wird. Die durch die großen Druckwellen entstehenden Vakuumbubbles, welche anschließend implodieren, führen lokal zu sehr hohen Drücken und Temperaturen, sowie zu Flüssigkeitsstrahlen mit einer Geschwindigkeit von bis zu

---

<sup>1</sup>alternativ ist sonst nur eine Durchstrahlung mittels Röntgenstrahlen möglich

400km/h. Treffen diese mechanischen Kräfte auf die zu reinigende Oberfläche, werden anhaftende Verschmutzungen gelöst und mit der Reinigungsflüssigkeit weggespült. Für eine intensive Kavitation und damit für eine intensive Reinigung, sind hohe Amplituden und eine niedrige Ultraschallfrequenz (ca. 20kHz) notwendig. In diesem Prozess werden Verschmutzungen, wie zum Beispiel Öle, Fette und Stäube entfernt und zwar so gründlich wie durch keine andere Reinigungsmethode. Die Intensivreinigung mit Ultraschall ist heute das gründlichste und umweltfreundlichste Feinreinigungsverfahren. Mit diesem Verfahren reinigt man zum Beispiel stomatologische Werkzeuge wie auch Endlosprofile, wie Drähte und Bänder. Auf diesem Prinzip basiert auch die, seit August 2001 von der Firma *Sanyo*, vertriebene Waschmaschine, die ohne Waschpulver auskommt. Sie wird aber nur auf dem japanischen Markt angeboten (Preis zirka 1350 Euro). Ultraschall wird außerdem auch zur Haar- und Pickelentfernung eingesetzt.

Ultraschall wird mit Hilfe elektroakustischer Wandler unter Ausnutzung des piezoelektrischen oder magnetostriktiven Effektes (elastische Körperverformung beim Magnetisieren  $\leftrightarrow$  Veränderung der magnetischen Eigenschaften bei Körperverformung) erzeugt.

Diese Wandler (zum Beispiel Quarz, Bariumtitanat, Piezokeramik beziehungsweise Nickel-Blechkpakete) werden durch elektrische, hochfrequente Wechselspannungen zu mechanischen Schwingungen unter Ausnutzung des Resonanzeffektes angeregt. Andererseits wird durch mechanische Verformung in einem Piezoelement eine elektrische Spannung erzeugt. Dieser physikalische Zusammenhang heißt Piezoeffekt. Die Frequenz der Schwingung ist vom Aufbau und der Technologie des Elementes abhängig. Die Resonanzfrequenz hängt von der Dicke  $d$ , zum Beispiel des Quarzscheibchens ab (siehe Gleichung 4.1).

$$f[\text{MHz}] \cong 2.8 \cdot d^{-1}[\text{mm}] \quad (4.1)$$

In Tabelle 4.2 ist eine kurze Auflistung von markanten Wertepaaren gegeben.

Plättchendicke [mm]	Resonanzfrequenz [MHz]
2.8	1
1.4	2
0.7	4
0.47	6

Tabelle 4.2: Auflistung einiger Wertepaare Plättchendicke/Resonanzfrequenz

Ein elektrischer Impuls stößt das Schallelement an. Die resultierende mechanische Schwingung pflanzt sich fort - wenn das Schallelement von der Luft umgeben wird - als Schallwelle. Sonst spricht man vom Körperschall (z.B. medizinische Ultrasonografie, Materialprüfung ...). An einem Zielobjekt wird der Schall reflektiert. Die zurücklaufende Schallwelle erzeugt am Schallelement<sup>2</sup> wiederum ein elektrisches Signal. Die Zeit zwischen dem Senden und Empfangen ist das Maß für die Distanz von Schallelement und reflektierendem Objekt (Pulsecho-Verfahren) in Abhängigkeit der Schallgeschwindigkeit/en des beziehungsweise der zu durchlaufenden Medien.

---

<sup>2</sup>Es kann sich dabei um das gleiche akustische Element handeln, was vor allem bei kostengünstigen Varianten oft gemacht wird. Auch sind Lösungen mit einem zusätzlichen Ultraschallwandler praktikabel. Dies vor allem im Bereich der Echolotung bei Schiffen.

## 4.2 Das Puls-Echo-Verfahren

Beim Pulsecho-Verfahren [4] werden (möglichst) kurze Ultraschall-Impulse in von der Impulsfolgefrequenz (10 Hz bis 100 kHz) bestimmten Abständen gesandt. In der Zeit dazwischen kann das Gerät Ultraschall-Echos empfangen.

Strahlt man Ultraschall in ein Werkstück ein, so wird im allgemeinen der Ultraschall an der gegenüberliegenden Wand wegen der dort angrenzenden Luft vollständig reflektiert und vom Prüfkopf als Echo empfangen. Andererseits werden nach dem Senden einer Schallwelle in Luft die gesendeten Wellen von allen Umgebungsgegenständen je nach Materialeigenschaften (wie zum Beispiel Dichte, Porosität, Dicke etc. ) ebenfalls reflektiert und nach der, der Entfernung entsprechenden Zeit, empfangen.

Wenn die Zeit zwischen dem Sendeimpuls und dem empfangenem Echo gemessen wurde, kann man aus der Gleichung 4.2

$$s = v \cdot \frac{t}{2} \quad (4.2)$$

die zu bestimmende Distanz  $s$  ermitteln.

Die in Gleichung 4.2 verwendeten Variablen haben folgende Bedeutungen:

$v$  Schallgeschwindigkeit (siehe Tabelle 4.1)

$t$  gemessene Zeit

In Abbildung 4.1 ist ein komplettes Messsystem mit Kommunikationsschnittstelle und Ausgabereinheit schematisch dargestellt.

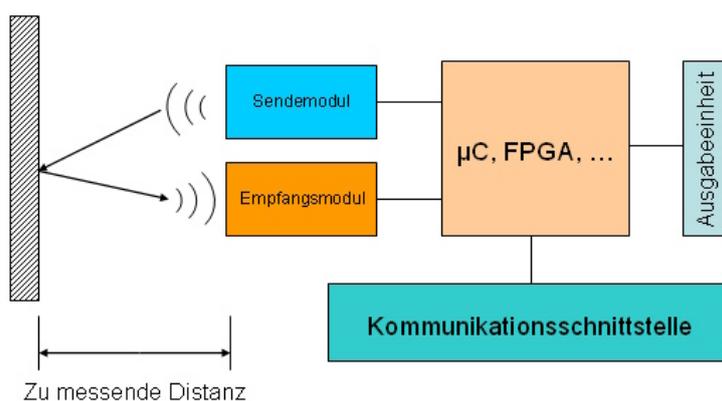


Abbildung 4.1: Schematischer Aufbau einer Puls-Echo-Messanlage mit zwei getrennten Schallwandlern

### 4.3 Schlussfolgerungen, Erfahrungswerte

Die Ultraschallabstandsmessung lässt sich wie gesagt in der Theorie auf eine Zeitmessung zurückführen.

Die Ausbreitungsgeschwindigkeit der Schallwellen in Luft ist konstant, solange Temperatur und Luftdruck sich nicht gravierend ändern. Sendet man also einen Ultraschallimpuls aus, so kann man aus der Zeit, die er bis zum Empfänger braucht, auf seinen zurückgelegten Weg schließen.

Mit heutigen Technologien<sup>3</sup> ist die notwendige Auflösung der Zeitmessung auch mit kostengünstigen Bauteilen ohne größere Probleme zu erreichen. Dies wird letztlich dadurch erreicht, dass neben der Kernarchitektur (betrachtet man  $\mu$ Controller) gerade in den letzten Jahren immer mehr Peripherie (z.B.: ADC, Timer, ...) Platz auf dem Chip gefunden hat, die unabhängig vom restlichen Programm autark arbeiten kann. Softwarefehler (Mängel der Software auf Grund von mangelnder Programmiererfahrung beziehungsweise Unberechenbarkeit der Programmlaufzeit bei Programmfragmenten, die mittels Compilertools erstellt wurden) haben somit fast keinen Einfluss auf das Messergebnis.

Auch ist es mit integrierten ADC sehr einfach und vor allem kostengünstig möglich, eine geeignete, digitale Temperaturkompensation durchzuführen.

Als großes Manko in vielen Bereichen hat sich der recht breite Schallkegel erwiesen, beziehungsweise dass ab einer bestimmten Distanz<sup>4</sup> die Schallwellen an kleineren Objekten nicht mehr ausreichend reflektiert werden, und somit das System versagt.

Trotzdem hat sich das System in vielen Bereichen, vor allem der industriellen Automation erfolgreich durchsetzen können. Dies nicht zuletzt wegen der geringen Verschmutzungsgefahr, der absoluten Lichtunempfindlichkeit und den hohen Erfahrungswerten.

---

<sup>3</sup> $\mu$ Controller, FPGA, ASIC, ...

<sup>4</sup>hängt vom Messbereich, der Schaltung, usw. ab

# Kapitel 5

## Messsystem mit GP2D120

### 5.1 Einführung und Aufgabenstellung

#### 5.1.1 Aufgabenstellungen

Die primäre Aufgabenstellung betrifft die Realisierung einer kostengünstigen Messanordnung (mit in der Industrie üblichen Bauteilen) zur sicheren Objekterkennung und Raumabtastung<sup>1</sup>, um Objekte (Wände, Türen, ...) von einem fahrenden Rollstuhl aus zu erkennen.

In diesem Kapitel soll zunächst eine kurze Recherche Aufschluss über den möglichen Einsatz von kostengünstigen, industriell eingesetzten Distanzsensoren durchgeführt werden.

Aus der später angeführten Liste ist eine Sensortype zu wählen, die folgende Kriterien erfüllen sollte:

- Reichweite von 1,5 - 5m
- Konversionszeit von  $< 40\text{ms}$
- kostengünstig (aktueller Preis: je nach Type circa 30 Euro)

Der nächste Schritt deckt die Wahl eines geeigneten Prozessortyps beziehungsweise eines gesamten Controllersystems ab.

Dies dient in dieser Diplomarbeit dazu, um die gemessenen Daten mittels standardisierter Schnittstelle auf ein Hostsystem zu übertragen und dort zu

---

<sup>1</sup>Erstellung eines Profils, des umgebenden Raumes

visualisieren.

Des Weiteren fällt in diesen Bereich die Realisierung einer geeigneten Firmware<sup>2</sup>.

Die genaueren Anforderungen und Realisierungsmöglichkeiten eines solchen  $\mu$ Controllersystems werden im Abschnitt 6.2.3 näher erläutert.

Als letzter Punkt muss noch die Erstellung der nötigen Software zur graphischen Darstellung der Daten behandelt werden.

### 5.1.2 Funktionsanalyse

Um eine korrekte Funktionsanalyse durchführen zu können, sind zunächst die dafür benötigten Komponenten<sup>3</sup> zu bestimmen.

Großteils ergeben sich diese Komponenten aus den im obigen Abschnitt 5.1.1 angeführten Punkten.

Folgende Einheiten werden benötigt, um die primäre Aufgabe erfüllen zu können:

- Sensor
- Auswerteeinheit auf unterer Ebene
- Auswerteeinheit auf höherer Ebene - Benutzerinteraktion

Gleich vorweg kann gesagt werden, dass die *Auswerteeinheit auf unterer Ebene* mittels eines  $\mu$ Controllersystems auszuführen sein wird.

Bei der *Auswerteeinheit auf höherer Ebene* wird es sich um einen handelsüblichen PC<sup>4</sup> handeln, wobei die Fragen nach dem Betriebssystem und der zu verwendeten Programmiersprache beziehungsweise Programmierumgebung in den späteren Abschnitten noch zu klären sind.

Grundsätzlich hat diese Messanordnung ein wie in Abbildung 5.1 gezeigtes Aussehen.

Erklärungsbedarf der Abbildung 5.1 ist bezüglich der nicht eindeutigen Schnittstelle zwischen Sensor und Kontrollerschaltung gegeben. Dies deshalb,

---

<sup>2</sup>Software, die auf unterster Ebene eines embedded Systems läuft

<sup>3</sup>zunächst abstrakt, später Auswahl Hardware/Software

<sup>4</sup>Desktop, Notebook, ...

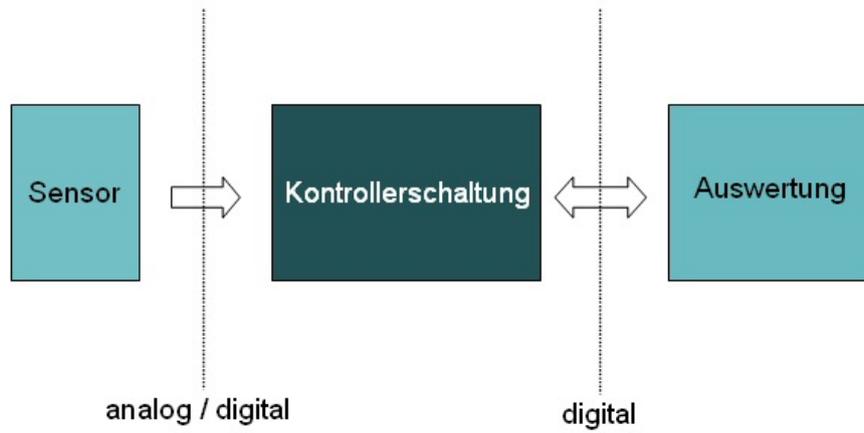


Abbildung 5.1: Schematisch- abstrakter Aufbau dieser Messanlage

weil noch nicht gewiss ist, welcher Sensortyp zu wählen ist und wie sich dieser physikalisch nach außen hin verhält<sup>5</sup>

---

<sup>5</sup>hat er beispielsweise eine analoge oder digitale Schnittstelle, wie groß ist der Spannungsbereich, ...

## 5.2 Recherche und Wahl des Sensors

### 5.2.1 Auflistung und Erläuterung von aktuell erhältlichen Distanzsensoren

Da mir noch keine weitreichenden Erfahrungen auf dem Gebiet der Sensorik zur Distanzmessung vorliegen, war es zunächst erforderlich in Erfahrung zu bringen, welche Messprinzipien und Sensoraufbauten kommerziell angeboten werden.

Nach einer intensiven Internetrecherche und einer anschließenden Bewertung bezüglich der Verwendbarkeit konnte eine Auflistung gängiger Produkte durchgeführt werden.

Erzeuger	Bezeichnung	Messprinzip	Besonderheiten
SHARP	GP2D02	Triangulation	Der Messwert liegt nach der Konversion in digitaler Form vor.
SHARP	GP2D120	Triangulation	Es sind mit dieser Type höhere Abtastraten möglich als bei GP2D02.
IPF Elektronik	PT 65 20	Triangulation	Linearisiertes Ausgangssignal ab 250 Euro
Welotec	W2300	Triangulation	Hohe Genauigkeit bei Distanzen im Bereich von 300mm. (ab 240 Euro)
Leuze electronic	ODS 96	Triangulation	Flexibel in der Konfiguration. Sensor kann digitales oder analoges Ausgangssignal liefern.
acam Messelektronik	TDC	Time Of Flight	Hat einen sehr hohen Messbereich, jedoch eine sehr geringe Auflösung.
SICK		Time Of Flight	180°, sehr teuer (einige tausend Euro)

Tabelle 5.1: Auswahl kommerzieller Sensoren (Preise beziehen sich auf das damals aktuelle Datum)

Diese Auflistung ist in Tabelle 5.1 zu sehen. Darin enthalten ist die jeweilige

Firmenbezeichnung nebst Sensorbezeichnung und verwendetem Messprinzip. Zusätzlich wurde noch eine Spalte hinzugefügt (Besonderheiten), in welcher eventuell vorhandene Sonderfunktionen oder atypische Merkmale aufgelistet sind.

### 5.2.2 Wahl des Sensors

Nach Abwägung der jeweiligen Vor- und Nachteile erschien es mir grundsätzlich als sinnvoll nur noch optische Sensoren der Firma *SHARP* [7] in Betracht zu ziehen.

Dies letztlich auch deswegen, da diese Sensoren äußerst kleine Abmessungen, bei gleichzeitig relativ guten Messergebnissen<sup>6</sup> und günstigen Anschaffungspreisen, vorweisen.

Eine weitere Annehmlichkeit ist die Verwendung von Licht im infraroten Bereich, sodass diese Sensoren nahezu völlige Störungssicherheit aufweisen.

Nachdem diese erste Frage geklärt ist, bleibt nur noch zu klären, ob Sensoren mit digitaler oder analoger Schnittstelle zu verwenden sind.

Führt man sich die Vor- und Nachteile der jeweiligen Sensorarten, die im Abschnitt 3 besprochen wurden vor Augen, so war auch recht schnell klar, dass nur Sensoren mit einem analogen Ausgangssignal zu verwenden sind. Die Vorteile die sich dadurch ergeben sind:

- Einfache Anbindung an einen Rechner
- Höhere Abtastraten sind möglich
- Sensor ist leicht für Analogtechnik zugänglich

Die grundlegenden Daten bezüglich Funktionsweise und Timing sind dem Abschnitt 3 zu entnehmen.

Aus Gründen der Verfügbarkeit wurde für die ersten Funktionstests der Distanzsensor mit der Bezeichnung *GP2D120* der Firma *SHARP* gewählt.

Dieser weist einen geringeren Messbereich auf, ist aber punkto Timing sowie Funktionsweise und mechanischen Anschlussdimensionen dem *großen Bruder GP2Y0A02YK* äquivalent und kann daher jederzeit ausgetauscht werden.

In Abbildung 5.2 ist der oben genannte Sensor bildlich dargestellt.

Zu sehen sind dabei die Teile betreffend die Optik sowie die Anschlussflansche und das dreipolige Anschlusskabel.

---

<sup>6</sup>Anmerkung: ich beziehe mich auf das Datenblatt und Publikationen der Firma *SHARP*, diese Angaben werden in den Funktionstests geprüft

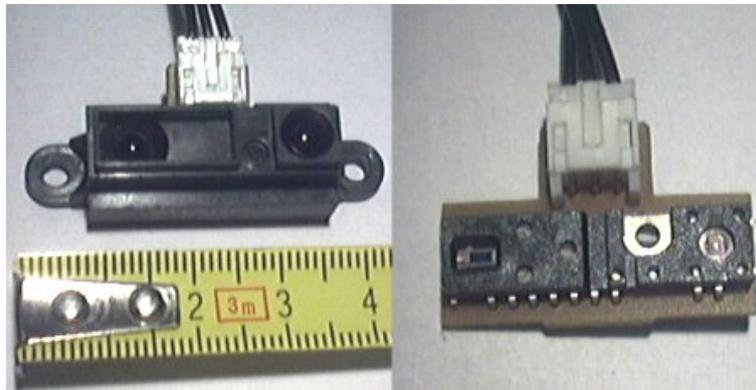


Abbildung 5.2: Photo des Distanzsensors GP2D120. Rechtes Bild zeigt den geöffneten Sensor.

Um sich ein genaueres Bild über die mechanischen Daten des Sensors machen zu können, wird das Studium des zugehörigen Datenblatts [7] empfohlen. Es seien jedoch die wichtigsten Abmessungen in der Abbildung 5.3 gegeben, um eine geeignete Sensoraufnahme konstruieren und fertigen zu können.

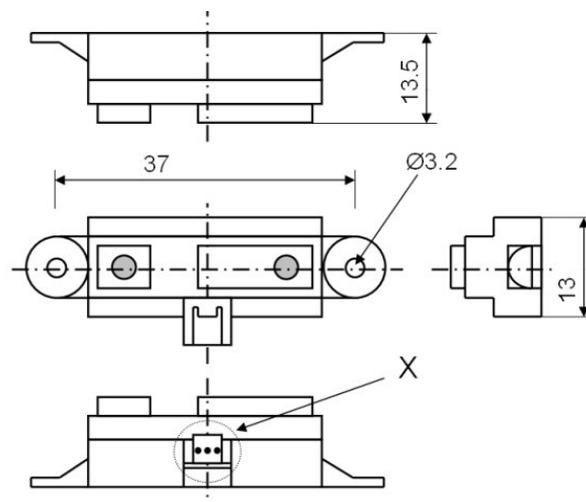


Abbildung 5.3: Skizze mit den wichtigsten Abmessungen des Distanzsensors GP2D120

Die in der Abbildung 5.3 mit X gekennzeichnete Detailansicht soll nunmehr die elektrischen Anschlüsse hervorheben und ist in der Abbildung 5.4 vergrößert dargestellt.

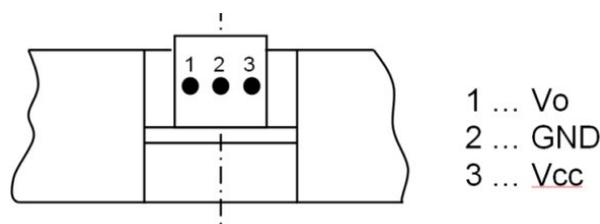


Abbildung 5.4: Skizze mit den elektrischen Anschlüsse des GP2D120

Wie der Legende der Skizze zu entnehmen ist, haben die drei Anschlüsse des Distanzsensors<sup>7</sup> folgende Funktion:

**Vcc** Versorgungsspannung:  $4.5V < V_{cc} < 5.5V$

**GND** Systemmasse

**Vo** Ausgangssignal:  $0.5V < V_o < 2.9V$

Erwähnt werden sollten an dieser Stelle noch wichtige Informationen bezüglich der Stromversorgung.

Seitens der Firma *SHARP* werden keine speziellen Anforderungen an die Stromversorgung gestellt. Es ist lediglich darauf zu achten, dass ein Mindeststrom von 50mA zur Verfügung steht und die Spannung sich im geforderten Bereich befindet.

Diese Forderungen sind mit einfachen Festspannungsreglern beziehungsweise wenn aus Kostengründen nicht anders möglich mit Zenerdioden<sup>8</sup> und Pufferkondensatoren einfach zu erfüllen.

Alle weiteren Betriebsdaten, insbesondere das Zeitverhalten bei sprunghaften Messwertänderungen sind dem Abschnitt 3 zu entnehmen.

<sup>7</sup>alle Typen der analogen SHARP-Distanzsensordfamilie haben die gleiche Anschlussbelegung

<sup>8</sup>4.7V

### 5.2.3 Sensoraufnahme für Profilerstellung

Bisher wurde stillschweigend zu Kenntnis genommen, dass die Distanzsensoren der Firma *SHARP* nur Messwerte erfassen können, die sich in einer bestimmten Richtung befinden. Um jedoch ein vollständiges Profil des umgebenden Raumes aufnehmen zu können, bedarf es zusätzlich zur Sensoraufnahme noch eines Aktors, der eine Schwenkbewegung vollzieht.

Dieser Abschnitt spricht nun diese beiden Teilaufgaben an, nämlich einerseits die Schaffung einer Universalaufnahme für die Distanzsensoren und andererseits die Wahl und Konzeption eines Schwenkantriebs.

#### Universalaufnahme für Distanzsensoren der Firma SHARP

An die Gestaltung solch einer Aufnahmevorrichtung werden nun folgende Anforderungen gestellt:

**Material:** Um ein geringes Massenträgheitsmoment zu erhalten ist es erforderlich, Kunststoffe oder Leichtmetalle zu verwenden.

**Fixierung:** Die Befestigung der Vorrichtung an unterschiedliche Aktoren soll mit einfachen Mitteln erfolgen.

**Herstellung:** Die Herstellung mit einfachen Werkzeugmaschinen muss möglich sein.

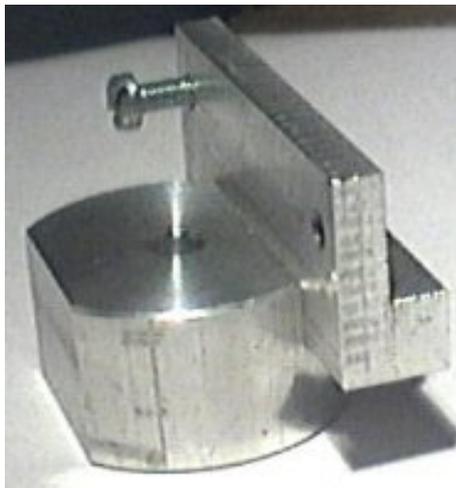


Abbildung 5.5: Photo der Universalaufnahme

Die Konstruktion der Universalaufnahme ergab schlussendlich zwei Teile, die aus Aluminium gefertigt wurden und alle Forderungen zur Zufriedenheit

erfüllten.

Abbildung 5.5 zeigt das gefertigte und zusammengebaute Bauteil.

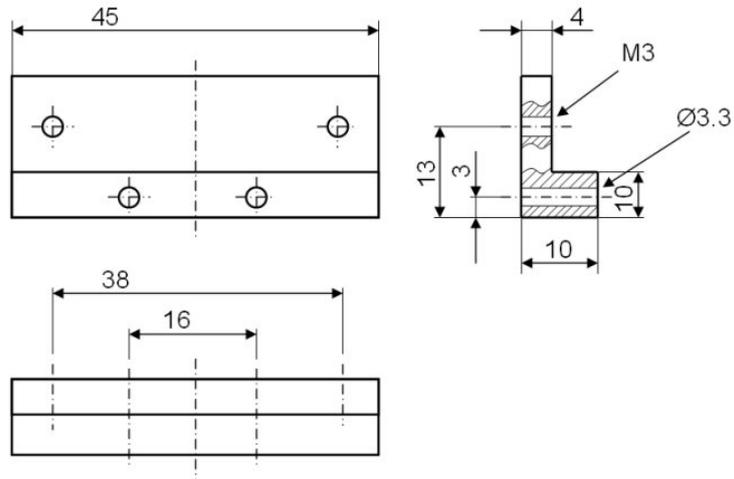


Abbildung 5.6: Skizze des ersten Teiles

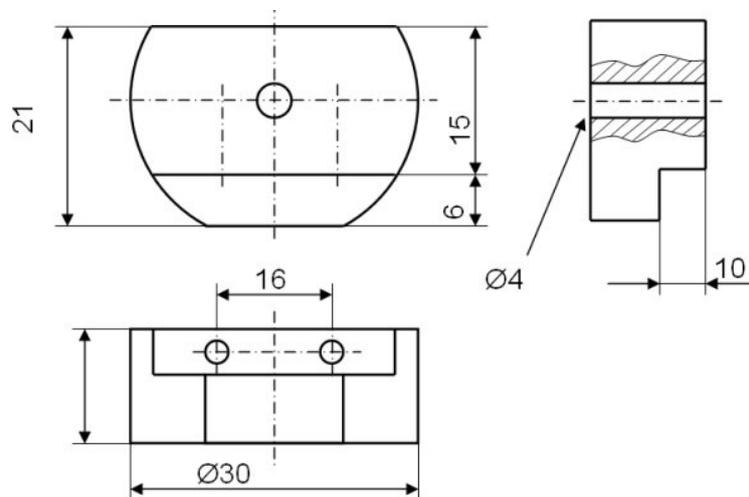


Abbildung 5.7: Skizze des zweite Teiles

Die Abbildungen 5.6 und 5.7 zeigen die genaueren Abmessungen der beiden Bauteilgruppen.

### 5.2.4 Aktoren für den Schwenkantrieb

Dieser Punkt behandelt nun die Wahl geeigneter Antriebe für den Schwenkantrieb zu Aufnahme von Messwertprofilen.

Um ein befriedigendes Ergebnis zu erhalten soll der Antrieb folgende Voraussetzungen erfüllen:

- möglichst geringe Stromaufnahme
- möglichst schnelle rotatorische Bewegungen sollen erreicht werden
- Winkelmesseinrichtung sollte inkludiert sein
- hohe Wiederholgenauigkeit

All diese Punkte werden nur von zwei möglichen Antriebsarten mehr oder weniger gut erfüllt.

- Schrittmotoren
- Modellbauservos

Sinn und Zweck dieser Arbeit ist es nicht, die Funktionsprinzipien dieser Antriebe näher zu erläutern. Wenn erforderlich, wird dem Leser empfohlen die doch recht umfangreichen Literaturwerke in Form von Printmedien oder elektronischem Material zu studieren.

Einzig die grundlegenden Methoden für den korrekten Betrieb dieser Einheiten werden an geeigneter Stelle beschrieben, was es auch dem Laien ermöglichen sollte, die später vorgestellten Softwareroutinen besser zu verstehen.

Welcher der beiden Antriebsarten für den praktischen Gebrauch besser geeignet ist, kann zu diesem Zeitpunkt noch nicht gesagt werden. Es erschien daher zweckmäßig beide zu testen um so einen Vergleich durchführen zu können, der offenlegt wo die jeweiligen Grenzen für einen etwaigen Einsatz liegen. Die so aufgebauten Systeme wurden schlussendlich einer praktischen Erprobung unterzogen und ausgewertet.

#### **Modellbauservos**

Im Modellbau schon seit Jahrzehnten im Einsatz, erobern diese recht kostengünstigen Baugruppen auch das Marktsegment der industriellen Automation.

Gründe dafür sind:

- Robustheit
- Geringe Stromaufnahme bei hoher Kraftentwicklung (präzise ausgedrückt: Hohe Momentenentwicklung)
- Gasdicht zu kapseln, können daher an explosionsgefährdeten Stellen platziert werden
- Inkludiertes Winkelmesssystem
- keine externe Leistungselektronik nötig

Das in Abbildung 5.8 gezeigte Schema verdeutlicht gerade die letzten zwei Punkte, denn Mechanik und Elektronik sind in einem Gehäuse fest miteinander verbunden und mittels Winkelgeber informationstechnisch miteinander gekoppelt.

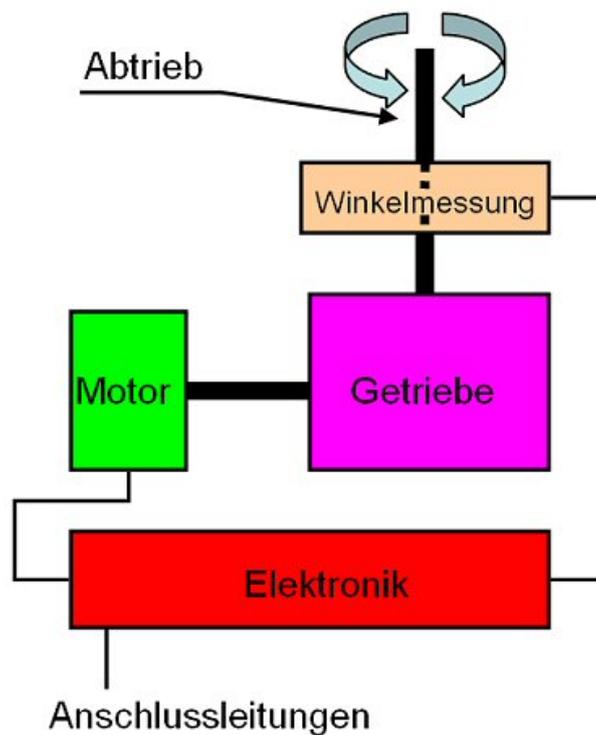


Abbildung 5.8: Schematischer Aufbau eines Modellbauservos

Der Winkelgeber (im Normalfall ein Drehpotentiometer) ist jedoch auch die größte Schwäche eines solchen Servoantriebs und dies gleich in mehrerer Hinsicht:

- Drehpotentiometer unterliegt Verschleiß
- Maximaler Drehwinkel von  $120^\circ$ - $180^\circ$
- Wiederholgenauigkeit leidet unter dem veränderlichen Übergangswiderstand des Schleiferkontaktes

Wie und ob sich dieser Umstand auch tatsächlich negativ bemerkbar macht wird im Abschnitt 5.7 behandelt.

Angesteuert werden diese Baugruppen mittels einer TTL-kompatiblen digitalen Leitung. Man gibt durch entsprechendes Anlegen eines Pulsmusters einen Winkel vor, der von der internen Regelung angefahren wird. Ein solches Pulsmuster ist in Abbildung 5.9 zu sehen.

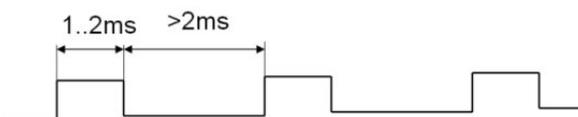


Abbildung 5.9: Pulsmuster zur Ansteuerung eines Modellbauservos

Für die interne Logik einzig ausschlaggebend ist die Zeitspanne des *High-Pegels*. Dieser sollte<sup>9</sup> im Normalfall zwischen 1ms und 2ms liegen, wobei 1ms dem kleinsten Drehwinkel entspricht und analog dazu ergibt eine Länge von 2ms den Maximalwert.

Grundsätzlich ist zu sagen, dass das Tastverhältnis nicht konstant sein muss, es sollte jedoch 50% nicht unterschreiten, da es sonst zu größeren Problemen führen könnte (die interne Logik kann die High-Pegel aufeinander folgender Pulse nicht mehr unterscheiden).

Die Generierung dieses Pulsmusters erfolgt in den meisten Fällen mit einem  $\mu$ Prozessor beziehungsweise  $\mu$ Controller.

Früher war es üblich diese Pulse mit Zählschleifen zu erzeugen, was äußerst prozessorzeitraubend war. Außerdem war man mit der höchst möglichen Anzahl von anzusprechenden Servos schnell am Ende, daher sind heute andere Methoden gefragt.

Moderne Derivate von  $\mu$ Prozessoren beziehungsweise  $\mu$ Controllern besitzen oftmals geeignete Peripherieeinheiten, mit denen auf elegante Weise dieses Pulsmuster erzeugt werden kann, ohne den Prozessorkern zu belasten.

<sup>9</sup>kann je nach Hersteller leicht abweichen

Bei diesen Peripherieeinheiten handelt es sich um PWM<sup>10</sup> und CCP<sup>11</sup> Einheiten, mit denen sehr einfach jedes beliebige, periodische Rechtecksignal erzeugt werden kann.

Um also eine automatische Profilerstellung zu ermöglichen muss der Schrittmotor mit dem Sensor mittels der, in Abschnitt 5.2.3 beschriebenen Sensoraufnahme verbunden und mit einem geeigneten Pulsmuster angeregt werden. Zyklisch dazu ist der aktuelle Messwert aufzunehmen, zwischenspeichern und schlussendlich als Teil einer Messwertreihe zur Auswerteeinheit zu senden.

Um einen möglichst großen Schutz des  $\mu$ Controllers zu gewährleisten und das Modellbauservo mit der erforderlichen elektrischen Energie versorgen zu können, war zusätzlich noch die Entwicklung einer einfachen Leiterplatte mit Pufferschaltung und Festspannungsregler notwendig.

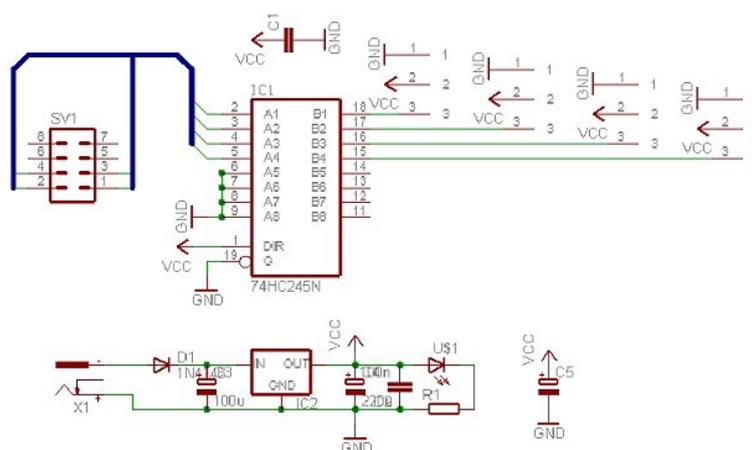


Abbildung 5.10: Schaltschema der Pufferschaltung mit inkludiertem Festspannungsregler

Sieht man sich das, in Abbildung 5.10 gezeigte Schaltschema an, so sind zwei Teilschaltungen zu erkennen.

**Festspannungsregelung:** Diesem Schaltungskomplex kommt die Aufgabe der Spannungsaufbereitung sowie der Betriebsanzeige zu. Kernelement dieser Teilschaltung stellt dabei IC2 dar, nämlich ein Standard Festspannungsregler mit der Typenbezeichnung *7805*.

<sup>10</sup>Pulse Width Modulation

<sup>11</sup>Capture Compare Einheiten



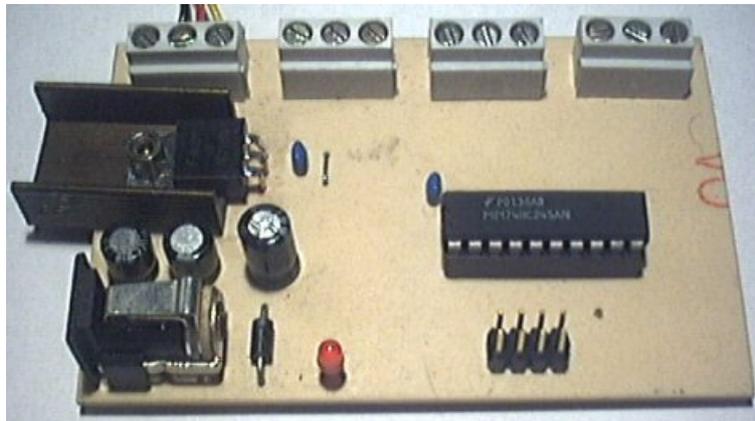


Abbildung 5.12: hergestellte und bestückte PCB der Pufferschaltung

der Vor- und natürlich auch Nachteile wird man deren positiven Eigenschaften recht deutlich erkennen.

Vorteilhafte Eigenschaften von Schrittmotoren sind unter anderem:

- konstante Schrittweite
- hohes Anfahrmoment
- bei geringen Drehzahlen geräuscharm
- sehr robust
- verschleißfrei

Nachteilig machen sich folgende Punkte bemerkbar:

- niedriges Moment bei höheren Drehzahlen
- keine konstante Winkelgeschwindigkeit
- keine sichergestellte Drehbewegung<sup>14</sup>
- kein absoluter Nullpunkt wie beim Modellbauservo

Man erkennt nach einer kurzen Analyse der oben erwähnten Auflistungen, dass die Nullpunktsuche sowie die Sicherstellung der tatsächlichen Schrittfolge mit Bedacht zu behandeln sind.

<sup>14</sup>es können Schritte verloren gehen, gilt besonders bei hohen Drehzahlen

Dies lässt sich zum Beispiel mit einfachen Endschaltern, Lichtschranken oder Ähnlichem bewerkstelligen.

Zur Ansteuerung sollen an dieser Stelle auch noch einige Worte verloren werden.

War es beim Modellbauservo noch möglich, mit einer Pulsleitung die gewünschte Winkelposition anzufahren, erfordert gleiches Verhalten beim Schrittmotor einen etwas höheren Aufwand, der wie noch gezeigt wird, mit modernen Mitteln leicht und elegant zu minimieren ist. Grundsätzlich ist bei der Ansteuerung zwischen *unipolaren Schrittmotoren* und *bipolaren Schrittmotoren* zu unterscheiden. Es wird in diesem Schriftwerk jedoch nur das grundlegende Prinzip der bipolaren Schrittmotoren erklärt. Dieser Schrittmortyp (Schrittmotor mit Dauermagnet als Rotor) hat ein Ständergehäuse aus Weicheisen (um den magnetischen Fluss zu leiten). Außerdem findet man am Ständer 4 Polschuhe mit den zugehörigen Ständerwicklungen, von denen jeweils die zwei gegenüberliegenden Spulen in Reihe geschaltet werden. Die Anschlüsse der Wicklungspaare müssen so beschaffen sein, dass sich ihre Magnetfelder zu einem homogenen Feld im Motorinneren ergänzen (siehe Abbildung 5.13).

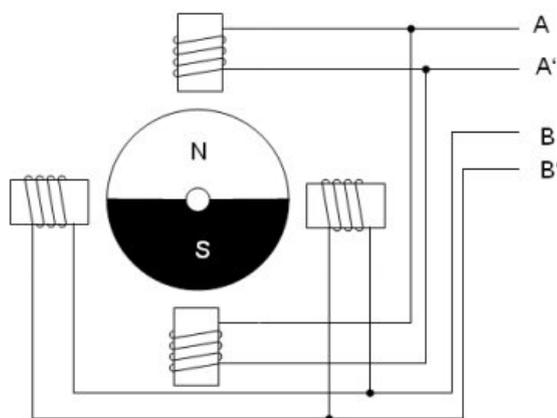


Abbildung 5.13: Schematischer Aufbau eines bipolaren Schrittmotors

Kommen wir nun zur elektrischen Beschaltung dieser Motoren. Der Name bipolarer Schrittmotor (weitere gibt es noch unipolare Schrittmotoren und Schrittmotoren mit variabler Reluktanz) leitet sich daraus ab, da an den Spulenanschlüssen je nach Zustand genau zwei mögliche Stromrichtungen auftreten können.

In der Abbildung 5.13 sind diese beiden Spulenstränge zu sehen. Aufgabe einer  $\mu$ Prozessoranschaltung ist es nun, die Spulen mit einem geeigneten, zykli-

schen Pulsmuster zu versorgen um eine rotatorische Bewegung zu erhalten. Wie leicht zu verstehen ist, können die dafür benötigten Spulenströme nicht von einfachen Ausgängen der Steuereinheit kommen. Es werden daher zahlreiche Treiberbausteine (Vollbrücken) am Markt angeboten, die nur für den Betrieb von Schrittmotoren bestimmt sind. Moderne Derivate genannter Bauteile besitzen zusätzlich noch die Möglichkeit des Chopperbetriebs<sup>15</sup> oder anderer Feinessen.

Bei der für diese Arbeit bestimmten Platine wurde ein alt bekanntes Bauteil der Firma *SGS Thompson* verwendet. Der Baustein besitzt die Typenbezeichnung *L293E* und benötigt für einen einwandfreien Betrieb eine Stromversorgung für die interne Logik und eine separate Versorgung für die Treiberausgänge. Die interne Logik ist TTL-kompatibel und ist daher mit 5V Gleichstrom zu versorgen.

Die fertig aufgebaute Platine<sup>16</sup> ist in der Abbildung 5.14 von oben und in der Abbildung 5.15 von unten zu sehen.

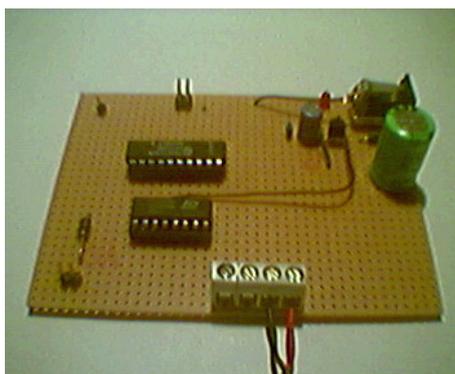


Abbildung 5.14: Treiberplatine zur Schrittmotoransteuerung von oben

Um einen reibungsfreien Betrieb gewährleisten zu können, liegt es am Programmierer der Firmware auf die Echtzeitfähigkeit des Systems zu achten. Mehr dazu wird im Kapitel 5.5.2 angeführt.

---

<sup>15</sup>Pulsweitenmodulierter Strom- beziehungsweise Spannungsverlauf um höhere Spannung zulässig zu machen

<sup>16</sup>2.54mm Lochrasterplatte

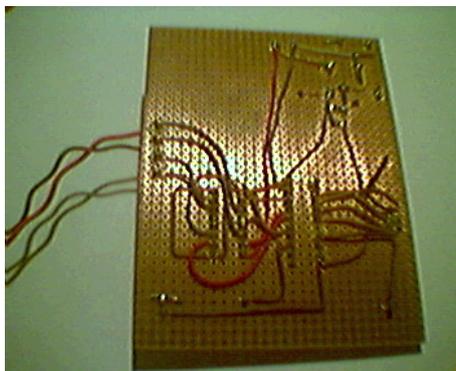


Abbildung 5.15: Treiberplatine zur Schrittmotoransteuerung von unten

### 5.3 Wahl der Steuereinheit

Als Bindeglied zwischen der Sensor-Aktorkombination und dem Hostsystem soll ein  $\mu$ Controllersystem eingesetzt werden. Es wurden dabei keine speziellen Anforderungen an das System gestellt. Es sollten jedoch nur Systeme verwendet werden, die einerseits eine kurze Einarbeitungszeit benötigen beziehungsweise bereits bekannt sind und andererseits kostengünstige Entwicklungswerkzeuge vorweisen können.

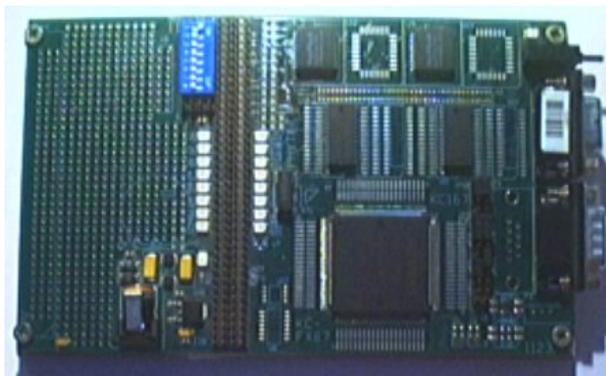


Abbildung 5.16: Bild des Phytec Entwicklungsboards

Auch sollte darauf Bedacht genommen werden, dass dieser  $\mu$ Controller eventuell auch noch andere Funktionalitäten zugewiesen bekommen könnte. Mögliche Aufgaben wären unter anderem eine Richtungsmessung mittels Kompassmodul<sup>17</sup>, Steuerungsaufgaben und Antriebsregelung.

<sup>17</sup>Hall-Sonden mit denen das Erdmagnetfeld gemessen werden kann und somit die Na-

Um diese Aufgaben durchzuführen zu können, bedarf es einer genügend hohen Rechenleistung und eventuell inkludierter On-Chip-Peripherieeinheiten. Es wurde daher ein 16Bit  $\mu$ Controller der Firma *Infineon* gewählt, der die Typenbezeichnung *C167CR-LM* trägt. Für diesen Controller gibt es eine Fülle an Entwicklungswerkzeugen in Form von Softwaretools und Entwicklungshardware. Auch war mir dieser Industrie-Standardcontroller bestens bekannt und ich konnte daher auf eine umfangreiche Erfahrung zurückgreifen, was eine rasche Softwareentwicklung ermöglichte. Reine Softwareentwicklung deswegen, da als Hardwareplattform ein Entwicklungsboard der Firma *Phytec* verwendet wurde, welches bereits voll funktionsfähig geliefert wurde und mit den restlichen Hardwarekomponenten<sup>18</sup> verbunden werden konnte.

Neben einem 256kByte großen Flash-Speicher für die Firmware besitzt dieses Board auch noch einen 64kByte großen RAM-Speicher für Datenfelder und Variable.

Erwähnenswert ist in diesem Zusammenhang auch noch, dass zur eleganten und rechenzeitparenden Ansteuerung von Modellbauservos vier PWM-Einheiten zur Verfügung stehen. Für die Kommunikation mit dem Hostsystem hingegen steht ein RS232-Modul bereit, das einen eigenen Baudraten-generator besitzt und somit äußerst flexibel und effizient eingesetzt werden kann.

Alle weiteren Details zum  $\mu$ Controller und dem Entwicklungsboard werden, sofern erforderlich, an gegebener Stelle näher erläutert oder müssen je nach Bedarf in Eigenregie in den entsprechenden Unterlagen welche unter [8] und [9] zu finden sind, gesucht werden.

---

vigation erleichtert wird.

<sup>18</sup>Sensor und Treiberkarten

## 5.4 Sensor-Controllerkopplung mittels Modellbauservo

Nach dem nun die einzelnen Komponenten der Messanordnung ausreichend erklärt wurden, ist es nun an der Zeit, den nächsten Schritt Richtung Endanordnung zu gehen. Dies ist nun die Kopplung der notwendigen Baugruppen und die Erstellung einer geeigneten Firmware.

Nun stellt die richtige Verbindung der einzelnen Komponenten kein großes Problem dar und soll auch nur kurz angesprochen werden. Einen etwas größeren Aufwand bereitete hingegen die Ausarbeitung der Softwarestruktur und deren anschließenden Erprobung.

### 5.4.1 Messanordnung

Das in Abbildung 5.17 gezeigte Schema gilt es nun mit geeigneten Kabeln, Klemmen und Ähnlichem in die Realität umzusetzen.

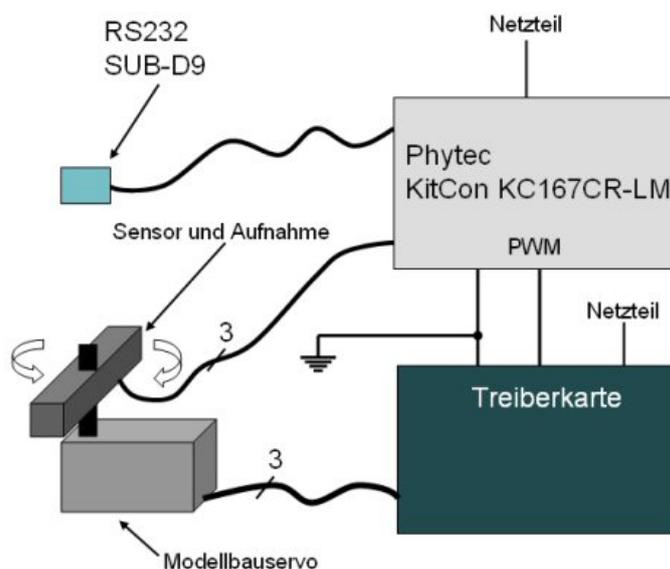


Abbildung 5.17: Verbindungsschema der Messanordnung mit Modellbauservo

Die direkte Verbindung vom Sensor *GP2D120* zum Controller erfolgt mit drei Leitungen.

Zwei Leitungen sind für die Stromversorgung, während die dritte Leitung das Ausgangssignal des Sensors darstellt und mit einem Analogeingang<sup>19</sup> des

<sup>19</sup>Portpin P5.0

$\mu$ Controllers verbunden wird.

Der Modellbauservo wird hingegen zunächst mit ebenfalls drei Leitungen an der Treiberkarte (siehe Abschnitt 5.2.4) angeschlossen. Zwei der Leitungen sind auch hier wieder Zuleitungen für die Stromversorgung, müssen aber aufgrund der höheren Werte der Stromaufnahme<sup>20</sup> (wie schon in Abschnitt 5.2.4 besprochen) von einer separaten Spannungsquelle angespeist werden.

Die verbleibende Leitung wird schließlich am Puffertreiber angeschlossen. Dieser Puffertreiber wird seinerseits so mit dem PWM-Modul des C167 verbunden, dass der angeschlossene Modellbauservo von der ersten verfügbaren PWM-Leitung angesteuert wird<sup>21</sup>.

Besonders zu achten ist auf eine gesicherte Masseverbindung zwischen Treiberkarte und Entwicklungsboard. Ob dies durch eine eigene Masseleitung oder durch ein gemeinsam genutztes Netzteil erfolgt, ist in diesem Fall jedoch irrelevant.

Die letzte Verbindungsleitung stellt ein ebenfalls dreipoliges Kabel dar, welches die Sende- und Empfangsleitungen der seriellen Schnittstelle sowie eine geeignete Masseleitung beinhaltet.

Kurz ansprechen möchte ich noch das (die) benötigte(n) Netzteil(e). Jeder der angesprochenen Elektronikschaltungen benötigt eingangsseitig eine Mindestspannung von 9V und einen minimalen Spitzenstrom von 500mA. Bei der Verwendung von nur einer Spannungsquelle ist also darauf zu achten, dass in Summe mindestens 1A zur Verfügung steht.

## 5.4.2 Softwarebeschreibung

Zunächst stellte sich die Frage, ob die Software mit einer höheren Programmiersprache wie zum Beispiel C oder mittels Assemblercode erstellt werden soll. Aus verschiedenen Gründen entschied ich mich für ANSI C und wählte als Entwicklungsumgebung die Werkzeuge der Firma *Keil* [10], die den Namen  *$\mu$ Vision* trägt.

Dieses Softwarepaket enthält neben dem Editor und dem Compiler auch noch ein eigenes Simulations- beziehungsweise Onlinedebugsystem, mit dem sehr effizient gearbeitet werden kann und dem Entwickler entscheidend geholfen wird.

Weitere Details entnehmen Sie bitte der umfangreichen Literatur der Firma *Keil* [10].

Gleich vorweg bitte ich um Verständnis, dass gewisse Fachausdrücke aus der

---

<sup>20</sup>bis zu 1A sind durchaus möglich

<sup>21</sup>Portpin: P7.0

Welt der  $\mu$ Computertechnik, die sich nicht vermeiden lassen, verwendet werden und dazu keinerlei Beschreibung zu finden ist. Es würde den Rahmen dieser Arbeit bei weitem sprengen, würde man auf alle Details der hier verwendeten Technologien eingehen.

Um einen Echtzeitbetrieb zu gewährleisten, erschien es am sinnvollsten, die gesamte Programmstruktur so aufzubauen, dass alle zeitkritischen Abläufe mittels Timer(interrupts) und sonstigen Interruptservices abgearbeitet werden. So ist es auch möglich, verschiedene Prioritäten zu vergeben, um ein konstantes Zeitraster für die Profilabtastung zu erhalten.

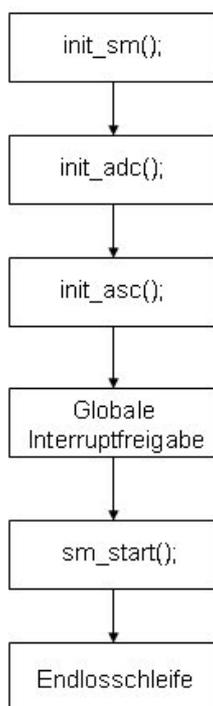


Abbildung 5.18: Flussdiagramm der *main()* Routine

Zunächst müssen jedoch die Peripherieeinheiten richtig konfiguriert werden. Dazu ruft die Routine *main()* die beiden Initialisierungsfunktionen *init\_sm()* und *init\_adc()* auf (siehe Abbildung 5.18).

***init\_sm()*** Initialisiert das PWM-Modul und den Timer 6. Timer 6 wird benötigt um eine Wartezeit einzuhalten, bis der Modellbauservo eine gewünschte Position angefahren hat. Durch das Einrichten des PWM-Moduls fährt der Modellbauservo bereits die erste Position an. In diesem Fall die minimal mögliche Winkelposition.

*init\_adc()* Hier wird der Analogeingang zur Messwerterfassung eingerichtet. Des Weiteren wird ein PEC<sup>22</sup>-Kanal eingerichtet, um die Daten interruptgesteuert in ein Datenfeld zu transferieren. Danach wird T5 so konfiguriert, dass er nach jeweils 40ms einen Messwert erfasst.

Die nächsten Programmabläufe sind das Freigeben des globalen Interrupt-Registers und das Anstoßen des Messvorganges durch die Funktion *sm\_start()*.

*sm\_start()* stellt nun zunächst den Timerwert für Timer 6 ein und startet diesen anschließend. Nach der abgelaufenen Zeit generiert der  $\mu$ Controller einen Interrupt und es wird die zugehörige Interrupt-Service-Routine *t6\_isr(void)* durchlaufen. Dieses Programmfragment deaktiviert den Timer 6 wieder, steuert die Winkelvorgabe und initiiert die Messwerterfassung durch Aktivierung des Timer 5.

Wie oben besprochen, löst dieser Timer nach 40ms ebenfalls einen Interrupt aus, der schlussendlich den internen ADC anregt und so der aktuelle Messwert in das Datenfeld *distances[cnt]* eingetragen wird, wobei cnt der aktuellen Winkelposition entspricht. Auch dies geschieht mittels Interrupt, welcher nach vollzogener *analog nach digital* Konversion erfolgt. Danach wird der nächste Konversionsdurchgang mit der Funktion *sm\_start()* eingeleitet.

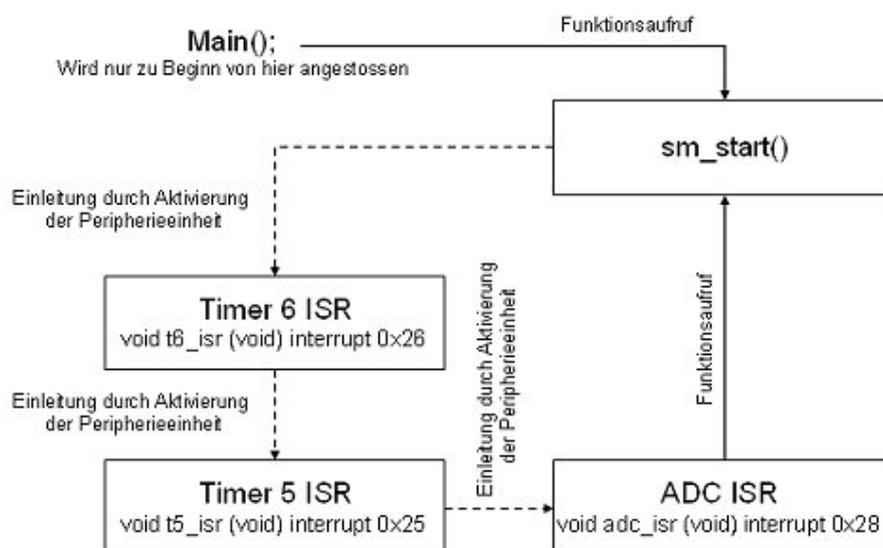


Abbildung 5.19: Ablaufschema der Programmabarbeitung

<sup>22</sup>Peripheral Event Controller: entspricht einem einfachen DMA-Kanal

Das nun besprochene Szenario ist bildlich in Abbildung 5.19 dargestellt. Dabei bedeuten durchgezogene Pfeile einen direkten Funktionsaufruf, während strichlierte Pfeile einen indirekten Funktionsaufruf ergeben. Unter indirektem Funktionsaufruf ist zu verstehen, dass eine Interrupt-Service-Routine nicht wie eine gewöhnliche Funktion aufzurufen ist, sondern nur eine Aktion gesetzt werden kann, welche durch das Erzeugen eines entsprechenden Interrupts einen Aufruf der gewünschten Funktion zur Folge hat.

Eine Datenanforderung des Hostsystems wird dadurch signalisiert, dass das Entwicklungsboard genau ein Datenbyte mit dem Inhalt *0xAA* empfängt. Der  $\mu$ Controller quittiert dies durch Senden der ausstandigen Messwertdaten. Dieses Senden der Messwertdaten erfolgt mit Hilfe eines PEC-Transfers. Dabei werden in die Befehlspipeline einzelne Kommandos (in diesem Fall Byte-Transferkommandos) injiziert und belasten somit den Prozessorkern nicht. Auch mussen hier nicht, wie bei allen anderen Interrupts, die Register gesichert werden, was eine zusatzliche Entlastung mit sich bringt.

## 5.5 Sensor-Controllerkopplung mittels Schrittmotor

Grundsätzlich ist gleich vorweg zu sagen, dass obwohl sich diese Antriebsarten grundlegend unterscheiden die Softwarestruktur sehr ähnlich ist. Doch beginnen wir auch hier zunächst wieder mit dem genauen Aufbau der Anordnung.

### 5.5.1 Messanordnung

Das in Abbildung 5.20 gezeigte Verbindungsschema weist auch schon den grundlegenden Aufbau des Messsystems vor.

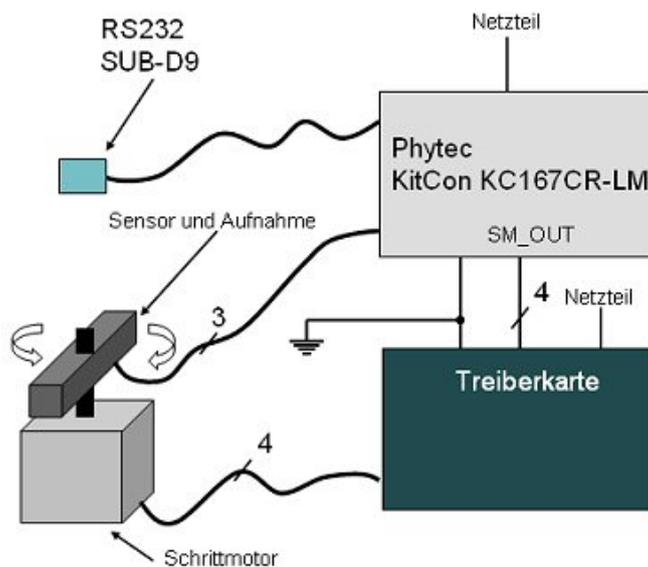


Abbildung 5.20: Verbindungsschema der Messanordnung mit einem Schrittmotor

Man muss schon etwas genauer hinsehen um die wirklich gravierenden Unterschiede zur Abbildung 5.17 zu finden. Diese betreffen die Verbindungen zwischen Antrieb (in diesem Fall der Schrittmotor) und Treiberkarte sowie der Verbindung zwischen Treiberkarte und Entwicklungsboard.

Benötigte das Modellbauservo zwei Versorgungsleitungen und eine Steuerleitung, so sind zum Betrieb eines bipolaren Schrittmotors vier Leitungen notwendig. Über diese vier Leitungen wird, wie schon erwähnt, die Information und die elektrische Energie übertragen. Es ist also sehr leicht verständlich,

dass natürlich auch vom Entwicklungsboard zur Treiberkarte nun vier Datenleitungen verlaufen müssen um die gewünschte Rotation zu erreichen.

Alle restlichen Verbindungen (insbesondere die Masseverbindung) bleiben wie in Abschnitt 5.4.1 beschrieben selbstverständlich erhalten.

## 5.5.2 Softwarebeschreibung

Auch die bei der Software entstandenen Unterschiede halten sich in Grenzen.

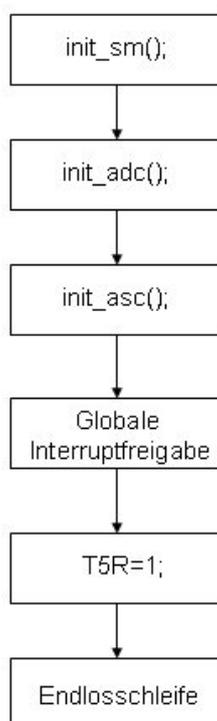


Abbildung 5.21: Flussdiagramm der *main()* Routine

Betrachtet man zunächst wieder den Ablauf der Hauptroutine *main()*, was in Abbildung 5.21 zu sehen ist, so unterscheidet sich dieser Programmablauf nur durch den fehlenden Funktionsaufruf von *sm\_start()*. Statt dessen wird die Programmabarbeitung durch den Start des Timers 5 *T5R=1*; eingeleitet.

Der restliche Programmablauf ist auch beim schrittmotorgesteuerten System vollkommen durch Interrupt-Service-Routinen realisiert.

Das in Abbildung 5.22 gezeigte Ablaufschema soll auch hier wieder Aufschluss über die einzelnen Programmschritte geben.

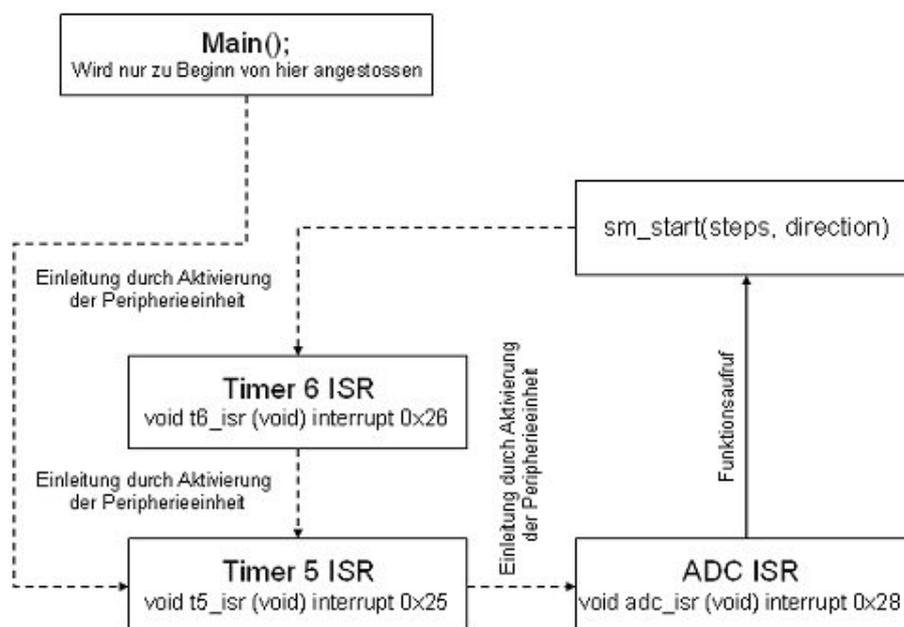


Abbildung 5.22: Ablaufschema der Programmabarbeitung

Zunächst muss der Programmablauf in der Routine *main()* angestoßen werden. Dies geschieht durch starten des Timer 5  $T5R=1$ ; und der dadurch in der ISR eingeleiteten Messwerterfassung durch den eingebauten ADC.

Nach erfolgter Konversion wird ebenfalls ein Interrupt generiert, auf den mit der entsprechenden ISR reagiert wird. In dieser Funktion wird der aktuelle Messwert in das richtige Datenfeld kopiert, die Drehrichtung festgelegt und die Rotation des Schrittmotors durch Aufruf der Prozedur *sm\_start(steps,direction)* gestartet. *steps* gibt in diesem Fall an wieviele Schritte in *direction* Richtung abzuarbeiten sind.

Hat der Schrittmotor die neue Winkelposition erreicht, so wird durch Aktivierung des Timer 5 die ganze Kette wieder von vorne angeregt.

Erwähnen möchte ich an dieser Stelle noch, dass zu Erhöhung der Betriebsfunktion der Schrittmotor beim Anfahren beziehungsweise beim Abbremsen keine abrupte Drehzahländerung erfährt, sondern durch einen Rampengenerator sanft beschleunigt beziehungsweise verzögert wird (siehe Abbildung 5.23).

Es hat sich später gezeigt, dass durch diese Maßnahme weit höhere Schrittfrequenzen möglich sind und die Massenträgheit der rotierenden Teile (Sensor,

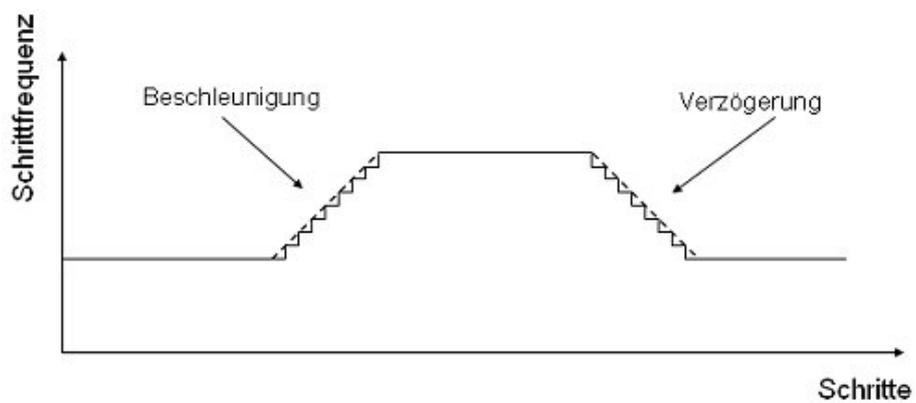


Abbildung 5.23: Beschleunigung und Verzögerung eines Arbeitszyklus

Aufnahme und Rotor) keinen sonderlich großen Einfluss auf die Winkeltreue mehr hat.

Das Auslesen der Nutzdaten durch das Hostsystem erfolgt in gleicher Weise, wie es beim Betrieb mit einem Modellbauservo bereits beschrieben wurde.

## 5.6 Das Hostsystem

Dieser Abschnitt beschäftigt sich nun mit der Wahl des Hostsystems und der Erstellung einer Visualisierungssoftware, um die aufgenommenen Profildaten für den/die Benutzer/in graphisch sichtbar zu machen.

Das Hostsystem gliedert sich dabei in zwei Bereiche:

- Hardwareplattform
- Betriebssystem

Die Wahl der Hardwareplattform fiel schnell auf einen gewöhnlichen PC. Alternativ hätte auch ein PDA oder etwas Ähnliches verwendet werden können, was bei eventuellen weiterführenden Studien auch der Fall sein könnte.

Bei der Wahl des Betriebssystems konnte zwischen LINUX und der WindowsNT -Schiene<sup>23</sup> gewählt werden.

Letztlich durchgesetzt hat sich das Produkt der Firma *Microsoft*, da folgende Punkte stärker wiegen als offenes Betriebssystem und kostenlose Entwicklungsumgebungen.

- Es ist sinnvoller Produkte außerhalb des wissenschaftlichen Bereiches unter dieser gewohnten Umgebung zu betreiben.
- Bessere Unterstützung der Hardware durch Treiber
- komfortablere Entwicklungsumgebungen (VC++, Borland C++, ...)

Zur Programmierung der Software wurde das Produkt *Visual Studio VC++* der Firma *Microsoft* verwendet. Zusammen mit der *MFC*<sup>24</sup>-Erweiterung bildet diese Werkzeugsammlung ein mächtiges Instrument zur effizienten Entwicklung von benutzerdefinierten Softwareprodukten.

Es wird sich noch zeigen, dass sich auch hier der Entwicklungsaufwand in Grenzen halten wird und schlussendlich eine kleine Demonstrationssoftware mit gefordertem Verhalten zur Verfügung steht.

---

<sup>23</sup>Win2000, WinXP

<sup>24</sup>Microsoft Foundation Classes

### 5.6.1 Anforderungsliste an die Software

Grundsätzlich gibt es nur zwei Hauptanforderungen, die direkt an die Funktionalität gerichtet sind.

Punkte wie selbsterklärend, ergonomisch, benutzerfreundlich usw. werden vom *Anwendungsassistenten*<sup>25</sup> übernommen, der ein rohes Programmgerüst erstellt, in welches anschließend das eigentliche Problem eingearbeitet wird. Die wirklichen Anforderungen sind:

- Kommunikation mit  $\mu$ Controller über RS232 Schnittstelle
- Graphische Darstellung der Messwerte am Bildschirm

In den folgenden Abschnitten wird nun der Aufbau der Software und die Bedienung selbiger beschrieben.

### 5.6.2 Beschreibung der Software

Es soll an dieser Stelle nicht der gesamte Quellcode Stück für Stück vorexerziert werden. Vielmehr soll dieser Abschnitt ein Wegweiser durch den Quellcode und dessen Ablauf sein, um die Struktur besser verstehen zu können.

Die erste grobe Programmübersicht möchte ich durch Präsentieren der Klassenhierarchie geben. Die genaue Funktion und Aufgabe von Klassen und Methoden, die durch den Anwendungsassistenten erstellt wurden, können auch hier wieder nicht vollständig beschrieben werden. Dafür reichen oftmals nur Literaturwerke mit einigen hundert Seiten. Es wäre also völlig absurd näher darüber zu berichten.

**CAboutDlg:** Darstellung und Abarbeitung des Informationsdialogfensters.

**CMainFrame:** Diese Klasse ist verantwortlich für den Aufbau der Anwendung. Unter anderem werden hier die Statuszeile, Toolbar und das Menü generiert.

**CSerialPort:** Diese in Eigenregie erstellte Klasse hat die Aufgabe, Kommunikationsschnittstellen für andere Programmteile zur Verfügung zu stellen. Diese Klasse stellt Funktionen bereit, um eine serielle Schnittstelle zu öffnen, zu schließen und Daten zu senden oder zu empfangen.

**CWinrollerApp:** Diese Klasse hat die Aufgabe erhalten, den Aufbau der Anwendung zu steuern und bedient sich anderer Klassen wie zum Beispiel *CMainFrame*.

---

<sup>25</sup>Teilstück des Visual Studio

**CWinrollerDoc:** Regelt das Erstellen eines neuen Dokuments der Anwendung. Unter Dokument versteht man den Arbeitsbereich der für den Benutzer zugänglich ist. Es muss nicht zwangsweise ein Text sein, vielmehr ist darunter nur ein Bereich zu verstehen mit dem das Programm mit dem Benutzer kommuniziert, und dies auf abstrakter Ebene.

**CWinrollerView:** Das in CWinrollerDoc abstrakt aufgebaute System wird mit der Klasse CWinrollerView konkretisiert. Diese Klasse wurde um einige Methoden erweitert, die das gewünschte Verhalten der Anforderungsliste ergeben.

Nach dieser kurzen Einführung ist es nun an der Zeit, etwas ins Detail zu gehen (zumindest was den Codeteil betrifft, welcher selbst erstellt wurde).

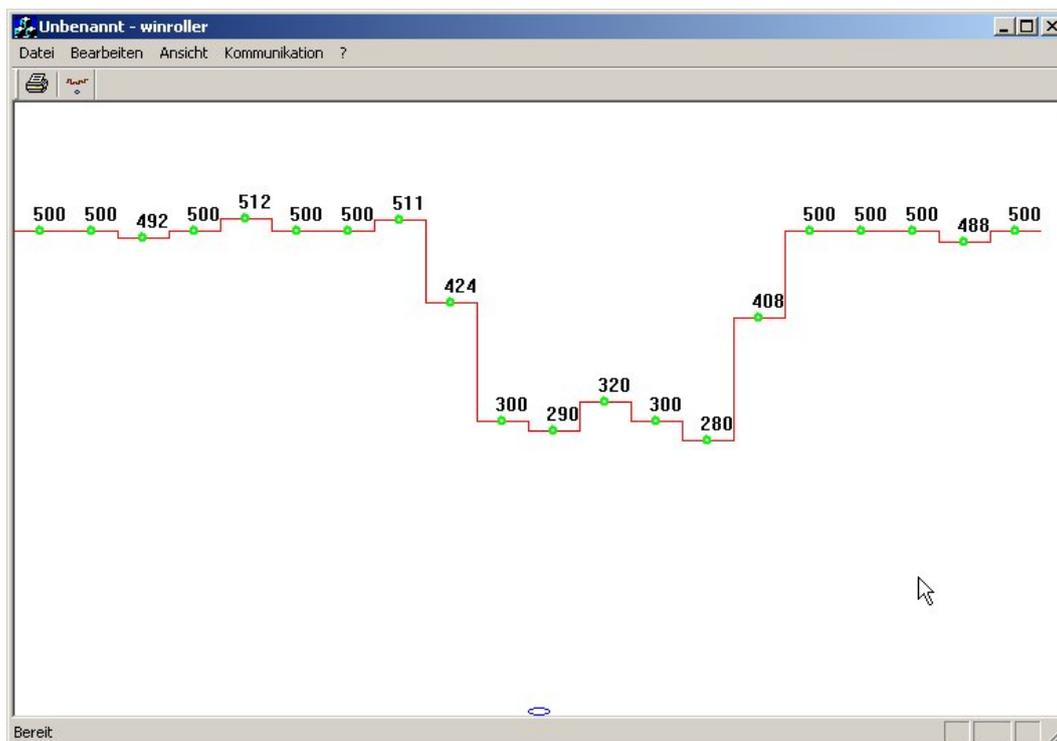


Abbildung 5.24: Hauptansicht der Anwendung *winroller*

Wird die Anwendung, welche den Namen *winroller* trägt, gestartet, so werden zunächst Objekte<sup>26</sup> der Klassen *CMainFrame*, *CWinrollerApp*, *CWinrollerDoc* und *CWinrollerView* erzeugt. Somit steht also das Grundgerüst

<sup>26</sup>von Objekt spricht man, wenn eine Klasse tatsächlich einen Speicherbereich zugewiesen bekommen hat

der Applikation.

Es geschehen aber zeitgleich<sup>27</sup> noch weitere wichtige Ereignisse.

So wird zum Beispiel das Objekt *port*, das eine Membervariable der Klasse *CWinrollerView* darstellt (*CSerialPort port;*), erzeugt und es wird der Konstruktor desselben durchlaufen. Hier wird die serielle Schnittstelle mit der Bezeichnung *COM1* geöffnet und konfiguriert.

Wird diese Schnittstelle bereits von einem anderen Programm benutzt oder ist diese defekt beziehungsweise nicht vorhanden, so meldet dies die Software durch eine entsprechende Nachricht.

Nun sollte am Bildschirm das Anwendungsprogramm *winroller* wie in Abbildung 5.24 gezeigt, erscheinen.

Betrachtet man diese Anwendung, so sind zunächst drei Bereiche erwähnenswert.

**Graphikbereich:** Ausgabe der Messwertdaten

**Menüleiste:** setzen von Aktionen durch den/die Benutzer/in

**Symbolleiste:** dient ebenfalls und gleichberechtigt der Interaktion mit dem/der Benutzer/in

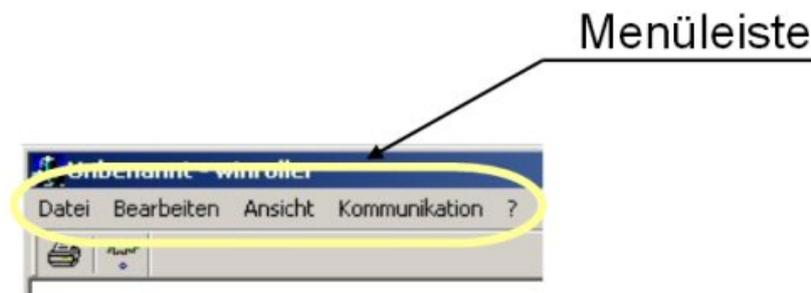
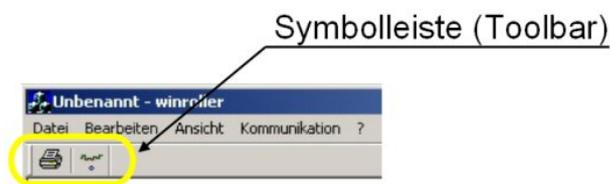


Abbildung 5.25: Detailansicht der Menüleiste

Die Menüleisten und Symbolleisten sind in den Abbildungen 5.25 und 5.26 vergrößert und hervorgehoben nochmals zu sehen.

---

<sup>27</sup>Dies ist nur bildlich zu verstehen, denn zeitgleich kann dies aufgrund des Vorhandenseins von nur einem Prozessor nicht geschehen; den genauen zeitlichen Ablauf kennt man jedoch auch nicht; es geschieht in einer derart kurzen Zeit, dass der/die Benutzer/in den Eindruck vermittelt bekommt, Parallelität des Systemes zu haben.

Abbildung 5.26: Hauptansicht der Anwendung *winroller*

Doch nun zurück zur Programmbeschreibung. Da es sich um eine verhältnismäßig kleine Anwendung handelt, ist es sehr einfach zu verstehen, dass es nicht sehr viele Interaktionsmöglichkeiten gibt. In diesem Fall lediglich eine einzige, nämlich das Aktivieren der zyklischen Messwertabfrage. Diese Messwertabfrage kann nun entweder über den entsprechenden Menüeintrag, wie es in Abbildung 5.27 zu sehen ist gestartet werden oder es lässt sich alternativ durch Betätigung des Startsymbols der Symbolleiste, wie es in Abbildung 5.28 zu sehen ist, das gleiche Ergebnis erreichen.



Abbildung 5.27: Start der Messwertabfrage über das Menü



Abbildung 5.28: Start der Messwertabfrage über die Symbolleiste

Ist die, zuvor erläuterte Initialisierung der seriellen Schnittstelle *COM1* erfolgreich gewesen und es wurde ein funktionsfähiges Entwicklungsboard mit dem dazugehörigen Sensor-Aktorsystem angeschlossen, so ergibt sich je nach vorgelegtem Raumprofil am Bildschirm ein Abbild desselben. Das in Abbildung 5.29 gezeigte Profil entspricht einer exemplarischen Objektanordnung. Diese Objektanordnung ist in Abbildung 5.30 zu sehen.

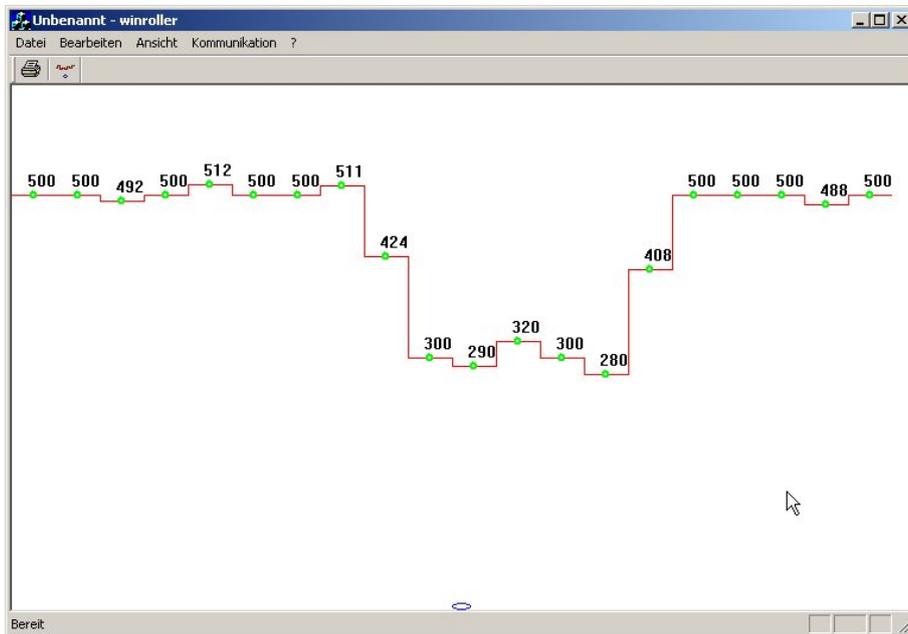


Abbildung 5.29: Exemplarische Profilaufnahme

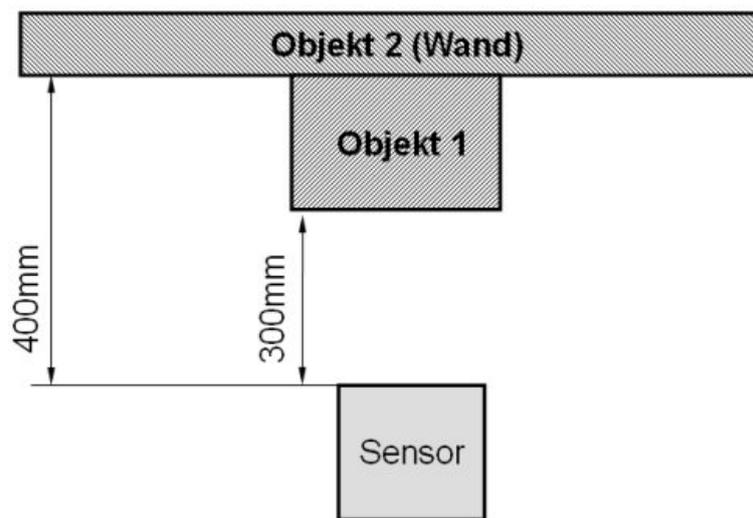


Abbildung 5.30: Skizze der Objektausrichtung

Programmtechnisch lassen sich diese Vorgänge wie folgt beschreiben.

Das Betätigen des Menüeintrages *Start* beziehungsweise das Anwählen des entsprechenden Symbols (siehe Abbildung 5.28) hat zur Folge, dass über das *Windows Messaging System* die Memberfunktion *void CWinrollerView::OnStart()* der Klasse *CWinrollerView* aufgerufen wird. In dieser Routine werden die ersten Messwertdaten angefordert und natürlich auch gespeichert. Der nächste Schritt betrifft das Anzeigen der Daten am Bildschirm. Dazu bedient man sich der Funktion *void CWinrollerView::OnDraw(CDC\* pDC)*, die leider nicht direkt aufgerufen werden kann. Sie wird normalerweise immer dann benötigt, wenn sich das Ansichtsfenster der Anwendung geändert hat, was in unserem Fall nicht zutrifft, da ja nur Daten in den Speicher transferiert wurden. Durch einen gezielten Einsatz einer von *Microsoft* bereitgestellten Routine kann jedoch ein solches Verhalten erzwungen werden und man erhält Zugriff auf die Funktion, um die Messwertdaten anzuzeigen.

Um ein zyklisches Anzeigen der Profildaten zu ermöglichen, ist es nun noch notwendig, einen Timer zu starten, welcher zu genau definierten Zeitpunkten Messwerte anfordert und diese anschließend wie gerade angeführt anzeigt. Dies geschieht durch das Programmfragment *SetTimer(1,250,NULL)*; wobei die Zahl 250 die Zeitangabe in Millisekunden ist. Die restlichen Parameter haben bei dieser Demonstrationssoftware keinerlei Relevanz.

## 5.7 Eindrücke, Vergleich der Antriebssysteme

Kommen wir zunächst zum Vergleich der Antriebssysteme des Schwenkantriebs.

Ohne nochmals näher auf die jeweiligen Vor- und Nachteile eingehen zu wollen ist gleich vorweg zu sagen, dass für eine präzise Winkelvorgabe das Drehpotentiometer des Modellbauservos schon nach kurzer Zeit nicht mehr ausreichend ist. Es hat sich gezeigt, dass sich auf Grund von Widerstandsänderungen des Schleifkontaktes<sup>28</sup> eine mehr oder weniger große Zitterbewegung bemerkbar gemacht hat. Diese Zitterbewegung, die in einer geschätzten Größenordnung von zirka zwei Grad liegt wirkt sich störend aus und verfälscht das Ergebnis. Je nach aktueller Lage auf der Kennlinie können sich dadurch recht beachtliche Änderungen des Ausgangswertes ergeben, deren Auswirkungen unter Umständen doch beträchtlich sein können. Dieses Problem entfällt beim Schrittmotor hingegen komplett.

Ein weiteres Manko des Modellbauservos war die Ungewissheit über den Verlauf des Rotors während der Neupositionierung. Insbesondere ist damit gemeint, dass nicht exakt festgestellt werden kann, wann der Rotor die Endposition erreicht hat. Dies mag zunächst recht banal klingen, hat aber in der Praxis dahingehend Auswirkungen, dass bei jeder Positionierung eine Zeitübergabe berücksichtigt werden muss und erst danach der Messwert aufgenommen werden kann. Diese zusätzliche Zeitspanne setzt nun die so wichtige Rasterzeit der Profilabtastung auf einen höheren Wert. Dies wiederum bedeutet, dass entweder weniger Abtastwerte aufgenommen werden können, um die Gesamtzykluszeit konstant zu halten oder es wird die Reaktionszeit des Rollstuhls verkürzt. Falls die Verkürzung der Reaktionszeit des Rollstuhls problematisch erscheint, besteht noch die Möglichkeit der Reduktion der maximalen Höchstgeschwindigkeit, um so einen geringeren Bremsweg zu erhalten.

Grundsätzlich ist in diesem Zusammenhang der Schrittmotor ebenfalls unproblematischer. Einzig verlorene Schritte könnten ein ähnliches Verhalten hervorrufen, dann hätte man aber das Problem, dass auch alle nachfolgenden Messwerte nicht dem tatsächlichen Verlauf repräsentieren und somit falsch sind. Unter Berücksichtigung der Massenträgheiten und entsprechendes Verhalten beim Betrieb des Schrittmotors (siehe Abschnitt 5.5.2) lassen sich solche Szenarien jedoch vermeiden.

---

<sup>28</sup>Anmerkung: ist nur eine Vermutung, deckt sich jedoch mit den Meinungen anderer Kollegen

Das Resümee zu dem Vergleich der Antriebsarten fällt ziemlich eindeutig aus. Das Modellbauservo kann in diesem Anwendungsbereich seine Vorzüge nicht geltend machen und kommt im Vergleich zum Schrittmotor immer ins Hintertreffen. Als Alternative zum Schrittmotor wäre es denkbar eine Eigenentwicklung mit Winkelgeber und gewöhnlichem Gleichstrommotor anzustreben, die maßgeschneidert auf die Problemstellung alle Vorteile in sich vereint.

Kommen wir schlussendlich zur Bewertung des Distanzsensors.

Als Universalsensor für Industrie- und Büroprodukte angepriesen waren dessen Grenzen recht bald ausgelotet. Ansatzweise konnte man dies schon im Abschnitt 3 erkennen, es wurde dort jedoch bewusst keine Wertung vorgenommen, da dies chronologisch zum Verlauf der Diplomarbeit nicht passen würde.

Bevor einige kritische Punkte aufgelistet werden, sollte unbedingt positiv hervorgehoben werden, dass dieser Sensortyp praktisch unempfindlich gegen Fremdlichteinstrahlung ist. Tests an unterschiedlichen Orten mit wechselnden Lichtverhältnissen hatten kaum bis keine Auswirkungen auf das Messergebnis.

Deutlich schlechter waren die Ergebnisse, die sich ergaben wenn Objekte sehr nahe beieinander liegen. Es konnten dann aufgrund des Konstruktionsprinzips des Sensors die Objekte nicht mehr getrennt detektiert werden. Mögliche Auswirkungen kann dies besonders im Fernbereich haben, wenn zum Beispiel aus einem Durchgang (Türstock) eine festes Hindernis wird. Negativ aufgefallen ist auch das Fehlverhalten bei Messungen, die in das Innere einer Wandecke weisen. Hier kam es je nach Ausrichtung des Sensors zu recht merkwürdigen Ergebnissen.

Der Abschlusskommentar kann also nur lauten, dass diese Sensortype sicherlich in vielen Bereichen vorzügliche Dienste leisten kann. Im Bereich der Objekterkennung automatisierter Rollstühle ist sie nach meiner Einschätzung jedoch fehl am Platz. Ganz so harsch sollte die Kritik wiederum auch nicht ausfallen, denn es wäre denkbar diesem Sensor Randaufgaben zukommen zu lassen. Mögliche Subanwendungen sind Seitenabstandsmessung zur Portal-durchfahrt oder ein kostengünstiges, autarkes Abstandswarnsystem. Kurzum Bereiche, die mit größter Wahrscheinlichkeit im Nahbereich arbeiten oder die eine geringe Auflösung in Bezug auf Objekterkennung tolerieren (es ist zum Beispiel nur wichtig zu erkennen, ob etwas im Weg ist oder nicht, die Größe spielt dabei keine Rolle).

# Kapitel 6

## Messsystem nach Triangulationsprinzip

### 6.1 Einführung und Aufgabenstellung

#### 6.1.1 Aufgabenstellungen

Nun hat sich in Bezug auf die Aufgabenstellung und das geforderte Leistungsspektrum, welches im Abschnitt 5.1.1 angegeben wurde, nichts geändert. Man ist lediglich zur Einsicht gekommen, dass eine kostengünstige Singlechip-Lösung in dieser Form derzeit nicht existent ist.

Dies bedeutet nicht zwangsweise, dass die Kosten in den Himmel wachsen dürfen und werden, es soll lediglich angedeutet werden, dass um gefordertes Ergebnis zu erhalten etwas mehr Aufwand auf der Sensorebene betrieben werden muss.

#### 6.1.2 Funktionsanalyse

War im Kapitel 5 noch eine klare Abgrenzung zwischen den Komponenten und deren Aufgaben möglich, so ist dies hier zwar noch auf den ersten Blick möglich, bei näherer Betrachtung der Dinge kommt man sehr schnell zu dem Schluss, dass hier noch tiefer auf der Abstraktionsebene vorzudringen ist, um eine Funktionsanalyse anzustellen.

Ausgehend von einer Fülle von Anwendungen muss es ein System von zunächst unabhängigen Komponenten<sup>1</sup> geben, welches selbst als eigenständige Anwen-

---

<sup>1</sup>nicht unbedingt als eigenständige Hardware

dung existiert und Schnittstellen oder ähnliche Möglichkeiten zur Kommunikation für andere Anwendungen zur Verfügung stellt. Diese Anwendung bezeichnen wir als Distanzsensor.

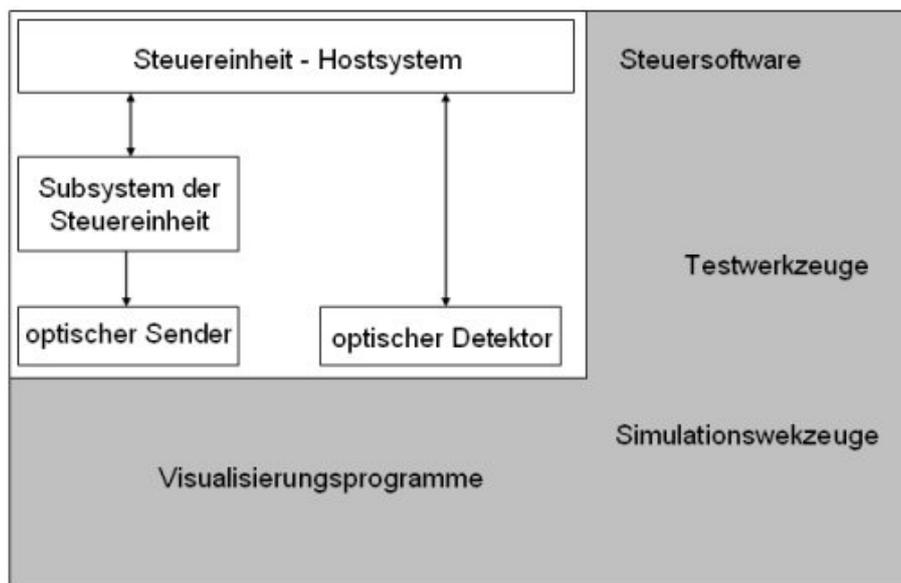


Abbildung 6.1: Schematisch-Abstrakter Aufbau dieser Messanlage

Der in Abbildung 6.1 weiß umrandete Bereich stellt genau diesen Systemkomplex dar.

Notwendige, eigenständige Komponenten sind dabei sicherlich das *optische Sendeelement* und das *optische Detektionselement*. Die weiteren Systemelemente (Steuereinheit - Hostsystem und Subsystem der Steuereinheit) wurden in dieser Skizze als eigenständige Systeme angesehen, können jedoch auch beliebig zusammengefasst oder auch erweitert werden. Dieses Schaubild soll nur einen ungefähren Einblick in den künftigen Leistungsumfang geben und ist nicht als Postulat anzusehen.

In den nun folgenden Erklärungen und Ausführungen werden einige Methoden erläutert, um solch ein System zu realisieren. Dabei wird insbesondere darauf geachtet, dass das Software-Hardware-Agglomerat möglichst offen zu halten, um Anwendungsgebiete außerhalb der Rehabilitationstechnik zu erschließen.

## 6.2 Beschreibung der hardwaregebundenen Komponenten

Aus dem schematischen Aufbau, welcher in Abbildung 6.1 zu sehen ist, lassen sich folgende Komponenten finden, die unbedingt einen hardwaremäßigen Aufbau verlangen.

- optische Sendeeinheit
- optische Empfangseinheit
- Schwenkeinrichtung

Die Schwenkeinrichtung unterteilt sich ihrerseits wieder in die Komponenten:

- Schwenkantrieb
- Aufnahmeteil für die optischen Einheiten
- Steuereinrichtung

In den nun folgenden Abschnitten wird detailliert auf die Wahl und Beschreibung aller Komponenten eingegangen. Besonderes Augenmerk wird dabei auf die Miniaturisierbarkeit geworfen, denn es erscheint schon von dieser Warte aus als sehr wichtig alle Bauteile möglichst klein zu halten, um keine zu große Messeinrichtung zu erhalten.

### 6.2.1 Optische Sendeeinheit

Dieser Begriff deckt alle in Frage kommenden Lichtquellen ab, die in der Lage sind, Lichtstrahlen auch über weite (einige Meter) Distanzen gebündelt auszustrahlen.

Eine kurze Recherche hat ergeben, dass die einfachste und kostengünstigste Variante ein fertig aufgebautes Lasermodul ist (alternativ könnte jedoch auch eine Laserdiode, versehen mit der entsprechende Leistungselektronik eingesetzt werden).

Diese Module werden in verschiedenen Leistungsklassen, Größen und mit unterschiedlichen Formen<sup>2</sup> des Ausgangsstrahles feilgeboten.

Preislich liegen diese Module im Bereich von ungefähr 60 Euro. Eine Möglichkeit, die letztlich auch genutzt wurde, besteht darin, einen preiswerten Laserpointer beziehungsweise dessen Lasermodul im Inneren zu verwenden.

---

<sup>2</sup>punktförmig, kurzer Strich, kreuzförmig

Die so erhaltene Lichtquelle weist folgende äußere Dimensionen<sup>3</sup> auf:

$$\phi 6,5\text{mm} \times 9\text{mm}$$

Zum Schutz der filigranen Leichtmetallhülle und um das Modul besser befestigen zu können, war es notwendig, eine Distanzhülse zu fertigen, die den Moduldurchmesser von 6,5mm auf 12mm erhöht.

In Abbildung 6.2 ist die fertig aufgebaute Modul-Distanzhülsen Kombination zu sehen.

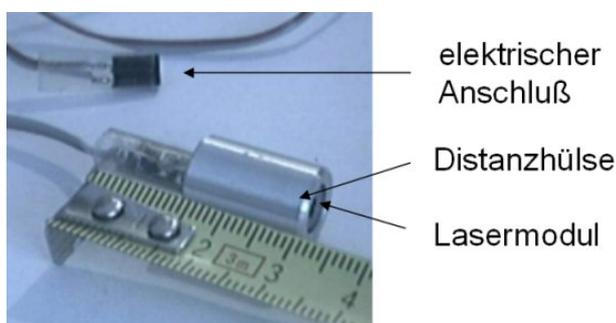


Abbildung 6.2: Lasermodule mit Distanzhülse und Anschlusskabel

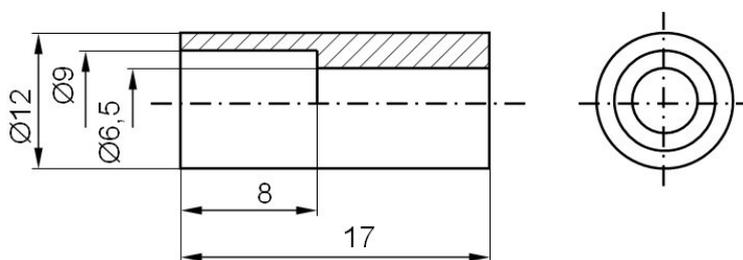


Abbildung 6.3: Bemaßungsplan der Distanzhülse

Die genauen Abmaße der Distanzhülse, welche aus Aluminium gefertigt wurde, sind in Abbildung 6.3 zu sehen.

Als Innenpassung zur Aufnahme des Lasermoduls wurde eine Spielpassung verwendet, was es notwendig machte, das Lasermodul einzukleben.

Die elektrischen Daten des verwendeten Lasermoduls sind in Tabelle 6.1 aufgelistet und wurden mit einem handelsüblichen Digitalmultimeter unter normalen Umweltbedingungen bestimmt.

<sup>3</sup>Platine mit Leistungstreiber wird nicht berücksichtigt

Bezeichnung	Wert	Dimension
Versorgungsspannung	4.5	V
Strom	35	mA

Tabelle 6.1: Elektrischen Anschlussdaten

### 6.2.2 optische Empfangseinheit

Die nächste notwendige Komponente zur Erstellung eines Triangulationssensors ist, wie schon im Kapitel 2.3 besprochen, die Detektionseinheit.

Um eine digitale Weiterverarbeitung zu ermöglichen, stellte sich erst gar nicht lange die Frage ob eine PSD oder ein CCD-Chip verwendet werden sollte und der CCD-Chip<sup>4</sup> kam zum Zug.

Nun musste noch geklärt werden ob, ein kommerzielles Produkt<sup>5</sup> oder eine Eigenentwicklung einer Kameraplatine verwendet werden sollte.

Dabei hat sich herausgestellt, dass beide Lösungswege eklatante Nachteile aufweisen. Sind es bei der Eigenentwicklung die moderaten Preise der Einzelkomponenten<sup>6</sup> die verlocken, wäre immer noch ein erheblicher Arbeitsaufwand zur Herstellung der elektronischen Schaltung notwendig. Dies auch deswegen, da auf diesem Gebiet keinerlei Erfahrungswerte vorhanden sind und eine zeitaufwendige Entwicklungsprozedur über etliche Prototypen in Kauf genommen werden müsste.

Bei kommerziell angebotenen Produkten ist die Sachlage komplett anders. So sind die meisten Angebote bestens dokumentiert (Anmerkung: diese Aussage beruht auf Internetrecherchen und Korrespondenzen mit den einzelnen Firmen wie zum Beispiel [13] und [14]) und mit Beispielapplikationen versehen. Ernüchternd hingegen sind die recht hohen Preise bei der Anschaffung, die im Bereich von 500 Euro und darüber liegen.

Es wurde daher zunächst eine kostengünstige und auch softwaretechnisch brauchbare Lösung anvisiert und schlussendlich auch gefunden. Es handelt sich dabei um *WebCams*<sup>7</sup>.

Abstriche müssen allerdings im Bereich der Reaktionszeit und in der maximalen Bildrate gemacht werden, doch dazu komme ich später nochmals

<sup>4</sup>oder natürlich CMOS

<sup>5</sup>Framegrabber-Karte und Kameramodul, ...

<sup>6</sup>einzig ein schneller ADC wäre im Bereich von ungefähr 100 Euro angesiedelt

<sup>7</sup>kleine, kostengünstige Kameramodule, die über Standardschnittstellen wie zum Beispiel USB oder FireWire mit einem PC verbunden werden können und dienen dem/der Benutzer/in als Eingabe von Bild- und Tondaten zur Kommunikation mit abgesetzten Empfangspartnern

zurück.

Gewählt wurde eine WebCam der Firma *Philips* [11] die folgende Typenbezeichnung aufweist: *WIA-Philips ToUcam Pro Camera*

Diese Kamera hat eine maximale Bildauflösung von 640x480 Bildpunkten bei einer Farbtiefe von 32Bit. Die höchstmögliche Bildübertragungsrate von 60fps<sup>8</sup> wird jedoch nur bei Bildgrößen von maximal 160x120 unterstützt. Darüberliegende Bildformate werden nur noch mit maximal 30fps übertragen.

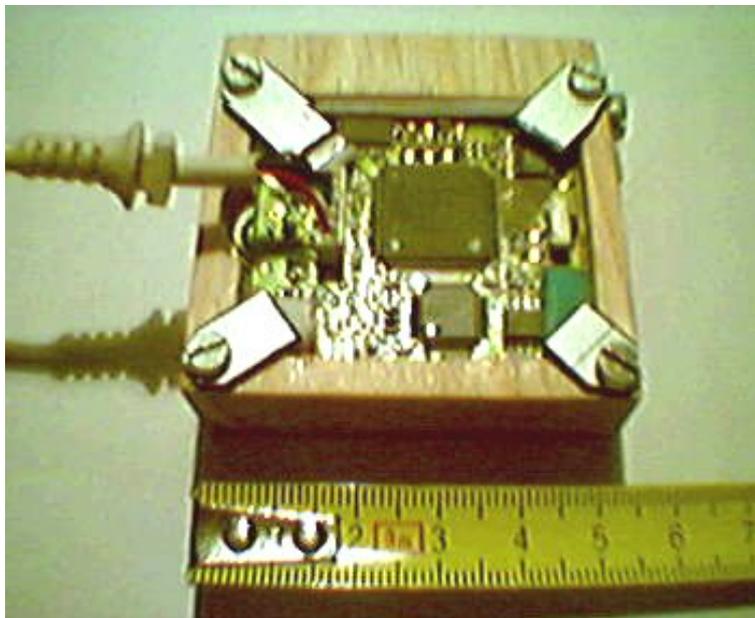


Abbildung 6.4: PCB der Philips WebCam

In Abbildung 6.4 ist die demontierte Platine der WebCam zu sehen. Auch enthalten ist die erforderliche Optik zur genauen Fokussierung. Das Kameramodul wird über einen freien USB-Anschluss mit einem handelsüblichen PC verbunden und erhält damit zugleich die notwendige Stromversorgung was ein zusätzliches Netzteil überflüssig macht. Weiterführende Informationen sind unter [11] zu finden.

<sup>8</sup>frames per second = Bilder pro Sekunde

### 6.2.3 Schwenkeinrichtung

#### Befestigungskörper

Kommen wir zunächst zum Befestigungskörper. Dieses Bauteil dient dazu, die optischen Sende- und Empfangskomponenten gemäß Kapitel 2.3 mechanisch miteinander zu koppeln.

Dazu dienen die beiden Teile, die in den Werkstattzeichnungen Abbildung 6.5 und Abbildung 6.6 zu sehen sind.

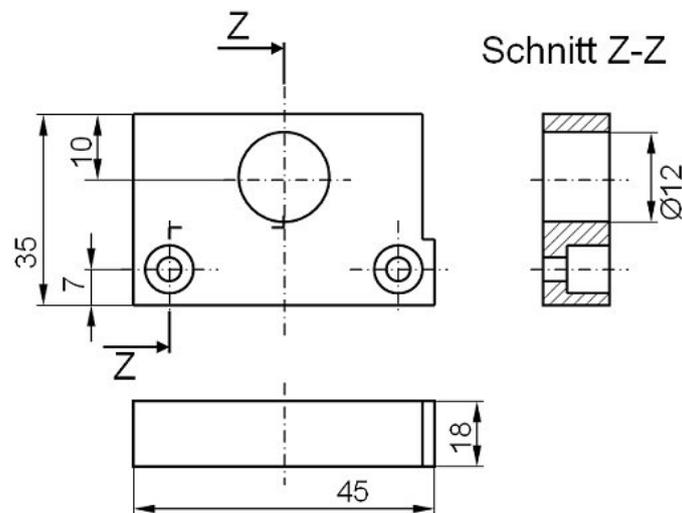


Abbildung 6.5: Unterer Befestigungskörper

In Abbildung 6.5 ist der untere Befestigungskörper zu sehen. Dieses Gebilde wird später mit der Universalaufnahme, welche im Abschnitt 5.2.3 beschrieben wurde, starr verbunden.

Des Weiteren kommen diesem Bauteil noch zwei weitere Aufgaben zu.

- Fixierungsmöglichkeit für Lasermodul (12mm Bohrung)
- Fixierungsmöglichkeit für Sensoroberteil (Seitliche Ausnehmung)

Der obere Befestigungskörper beherbergt die Platine der Philips WebCam und wird mit dem unteren Teil verbunden.

Die Verbindung wird über eine Aluminiumplatte derart hergestellt, dass je nach Wunsch ein Winkel zwischen den beiden Sensorteilen eingestellt werden kann, der charakteristisch für die Sensorkennlinie ist.

Gefertigt wurden diese beiden Sensorteile aus Holz. Dies mag zunächst etwas antiquiert wirken, im Prototypen- beziehungsweise Modellbau hat der

Werkstoff Holz jedoch immer noch seinen angestammten Platz, auch wenn dieser in letzter Zeit immer mehr an Bedeutung verliert. Bei einem Redesign wäre es daher durchaus möglich und sinnvoll diese Teile aus Leichtmetall<sup>9</sup> oder Kunststoff zu fertigen.

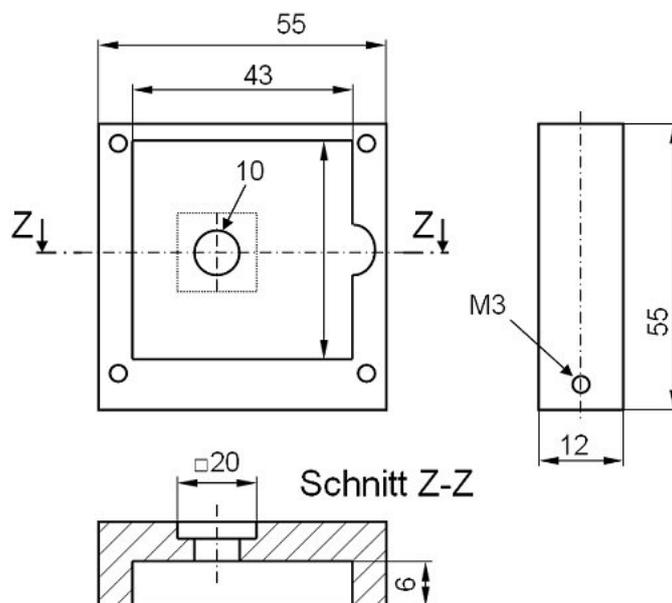


Abbildung 6.6: Oberer Befestigungskörper

Nicht angeführt wurden Detailzeichnungen des Verbindungselements und sonstiger Kleinteile wie Schrauben oder Befestigungsklemmen, was aber für das Verständnis dieser Arbeit keinerlei Hindernis darstellt.

### Steuereinrichtung

Wie schon mehrfach erwähnt, ist es auch bei dieser Anwendung unabdingbar, eine Drehbewegung zur Profilaufnahme durchzuführen. Längst war klar, dass diese Aufgabe mit einem Schrittmotor zu lösen ist und dementsprechend wieder etwas an elektronischer Entwicklung erforderlich sein wird. Weitere Aufgaben sind das Ein- und Ausschalten des Lasermoduls und die Kommunikation mit einem Hostsystem durch eine RS232-Schnittstelle.

Der Neuentwurf einer Treiberplatine für den Schrittmotor, wie in Abschnitt 5.2.4 ausführlich erklärt, ist daher obsolet.

<sup>9</sup>man muss bei drehbeweglichen Teilen immer die Massenträgheitsmomente berücksichtigen

Vielmehr hat sich die Frage aufgedrängt, ob für eine einfache Schwenkbewegung tatsächlich ein teurer und leistungsfähiger  $\mu$ Controller verwendet werden muss, denn anders als im Kapitel 5 angedeutet, besteht in diesem Fall kein Bedarf an zusätzlicher externer Rechenleistung zur Bewältigung von Steuerungsaufgaben. Grund dafür ist, dass diesmal der Sensor<sup>10</sup> viel enger mit dem Hostsystem verbunden ist (ja sogar verschmolzen) und somit auch alle restlichen Aufgaben durch entsprechende Softwareroutinen abgewickelt werden können.

Da ich auf dem Gebiet der 8Bit  $\mu$ Controller Erfahrungen mit den Produkten der Firma *Microchip* [15] habe, wurde ein Controller der Serie *18FXXX* gewählt.

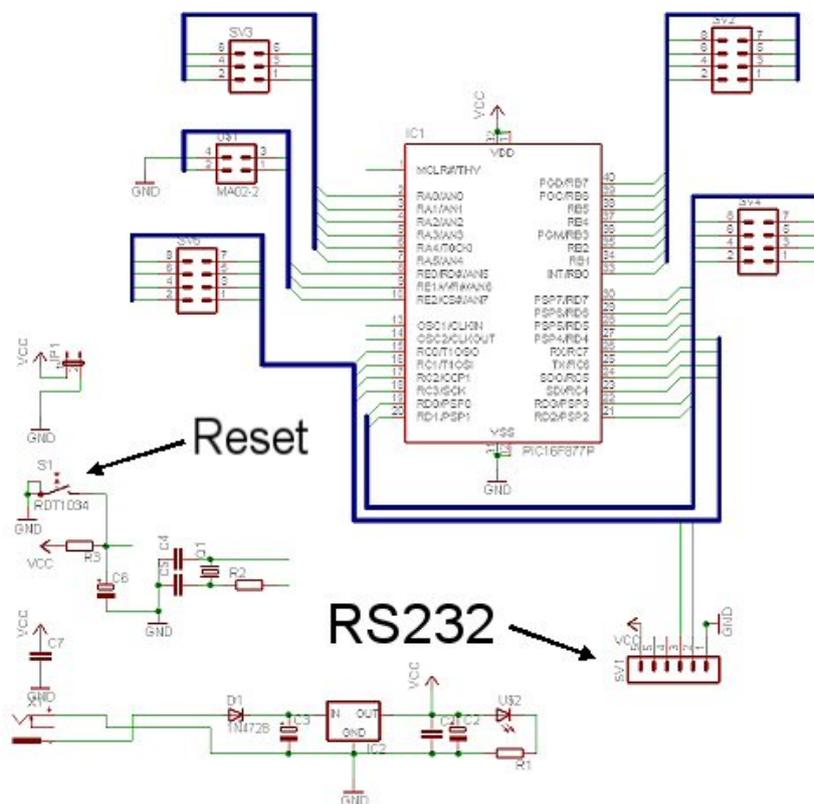


Abbildung 6.7: Schaltschema des Entwicklungsboards

Diese neue, überarbeitete Prozessorarchitektur vereint höchste 8Bit-Rechenleistung mit etlichen Peripherieeinheiten (zum Beispiel ADC, DAC und PWM

<sup>10</sup>eigentlich nur Teile davon

um nur einige zu nennen) gepaart mit einem flexiblen Registersatz. Diese  $\mu$ Controller, auch PIC genannt haben nun als großen Vorteil gegenüber dem zuvor vorgestellten  $\mu$ Controller alle Speicherarten (dazu gehören Festwertspeicher für die Programme, und flüchtige Speicher zur Aufnahme der Variablen und Daten) in einem vierzigpoligen DIL<sup>11</sup> Gehäuse vereint, was Platz und Kosten spart.

Von der Firma *Microchip* wurde ein entsprechendes Derivat kostenlos zur Verfügung gestellt welches die Typenbezeichnung *PIC18F452* trägt. Nähere Information zur Architektur und den inkludierten Peripherieeinheiten finden Sie unter [15].

Es wurde nun ein Entwicklungsboard mit dem schon bekannten Softwaretool Eagle der Firma CADSoft [16] entworfen und mit einer Isolationsfräsmaschine gefertigt. Dieses Entwicklungsboard versorgt den Controller mit dem erforderlichen Strom, Resetsignal und dem Systemtakt, was in Abbildung 6.7 zu sehen ist. Weiters werden alle Portpins durch Stiftleisten zugänglich gemacht, was besonders für Prototypenfertigungen vorteilhaft ist.

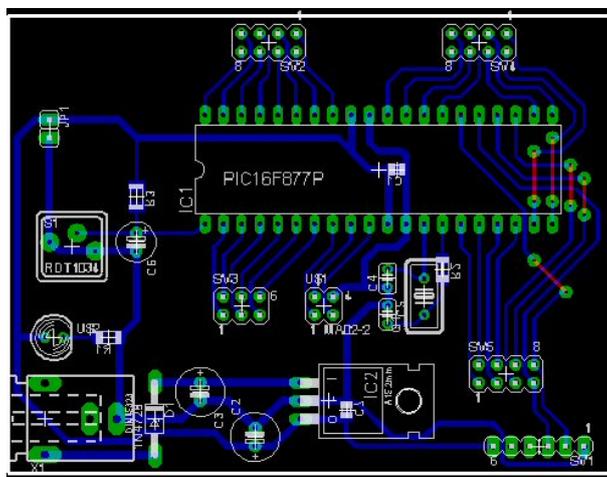


Abbildung 6.8: Entflechtete PCB des Entwicklungsboards

Besonders hervorgehoben wurde dabei der Anschluss für die RS232-Schnittstelle. Es ist äußerst wichtig für etwaige Adapterschaltungen diese Signalbelegung einzuhalten um keine Überraschungen bezüglich Bauteilerstörung bei der Inbetriebnahme zu erhalten.

<sup>11</sup>Dual In Line

Abbildung 6.8 zeigt schließlich den fertig entflochtenen Print mit allen Bauteilen und Durchkontaktierungen der einseitigen Schaltung.

In Abbildung 6.9 ist die gefertigte Printplatte mit allen bestückten Bauteilen zu sehen.

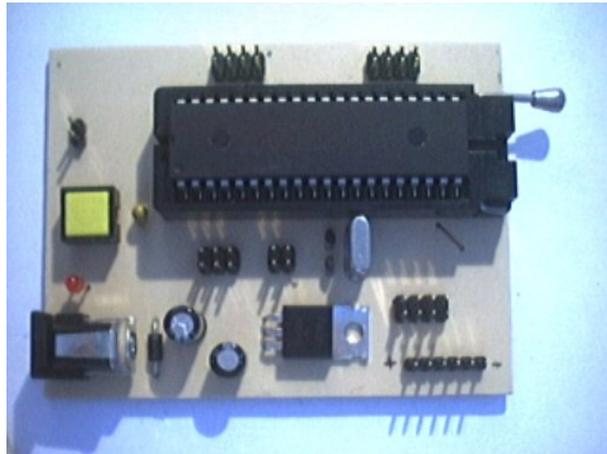


Abbildung 6.9: Bauteilansicht des gefertigten Entwicklungsboards

## 6.3 Beschreibung der Software

### 6.3.1 Beschreibung der Firmwareroutinen

Die Programmierung der Firmware erfolgte mit dem *MPLAB Studio* der Firma *Microchip* [15] unter Verwendung der Sprache ANSI-C.

Doch nun zum eigentlichen Teil der Aufgabe. Der Programmstart wird immer mit dem ersten Befehl der Routine *void main(void)* eingeleitet. Hier werden zunächst einige Variablen mit sinnvollen Werten versehen und anschließend wird die Initialisierungsfunktion *void init(void)* durch den Aufruf *init();* angestoßen.

Wie schon aus dem Namen zu schließen ist, werden hier die Peripherieeinheiten, das sind

- Timer zur Schrittmotorsteuerung
- USART: RS232-Schnittstelle

initialisiert und auf einen definierten Anfangszustand gebracht. Danach wird eine Programmstruktur durchlaufen, wie sie in Abbildung 6.10 schematisch dargestellt wird.

Das Programm wartet, wie leicht zu sehen ist am Beginn jedes Schleifendurchlaufes auf Daten vom Hostsystem, die über die serielle RS232-Schnittstelle übertragen werden.

Im Falle des Eintreffens von Daten muss entschieden werden, ob sie verworfen werden sollen (im Falle dass Daten übermittelt wurden, die keinen Sinn ergeben) oder welche Aktion gesetzt werden muss.

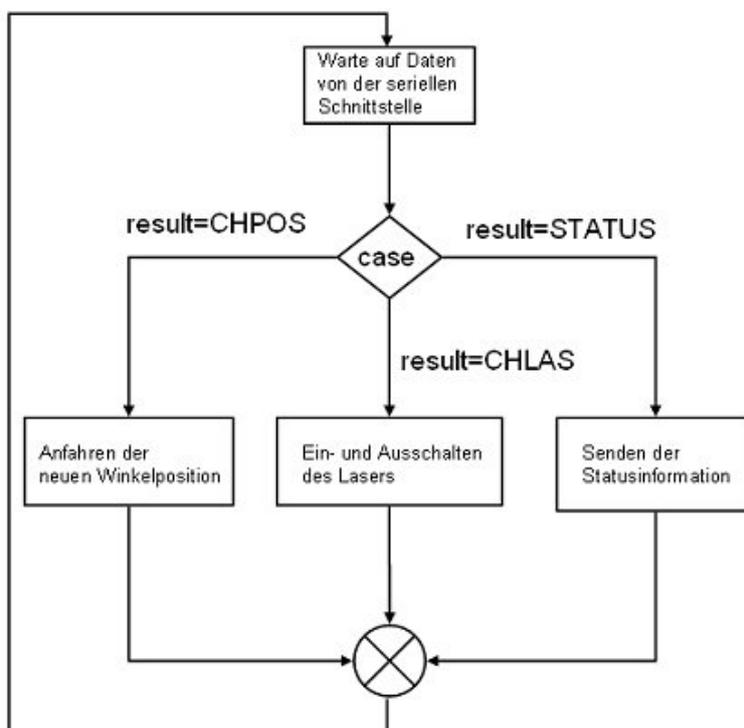


Abbildung 6.10: Flussdiagramm der Main-Routine

Mögliche Aktionen sind dabei:

**CHPOS:** Wurde dieser Zahlenwert übermittelt (CHPOS ist nur ein Synonym für einen Zahlenwert, gilt auch für die beiden noch ausstehenden Werte), so hat dies eine Positionsänderung zur Folge. Die anzufahrende neue Winkelposition wird absolut angegeben und benötigt bei der Übertragung vom Hostsystem zwei Byte.

**CHLAS:** Mit diesem Kommando ist es möglich das Lasermodul einzuschalten oder auszuschalten. Um die genaue Aktion durchführen zu können, wird ein zusätzliches Byte mit dem gewünschten Verhalten gesendet.

**STATUS:** Hier wird lediglich der Systemstatus zum Hostsystem übertragen. Systemstatus bedeutet hier lediglich die aktuelle Winkelposition.

Wurde das entsprechende Softwarefragment abgearbeitet, so wird in jedem Fall eine Befehlsquittierung *EOC*<sup>12</sup> an das Hostsystem übertragen, um damit die folgerichtige Abarbeitung zu signalisieren.

Danach wird wieder zum Schleifenanfang zurückgesprungen und auf das nächste Kommando gewartet.

Alle weiteren Details dieses recht einfachen Übertragungsprotokolls sind dem Quellcode zu entnehmen.

### 6.3.2 Beschreibung der Routinen des Hostsystems

Kommen wir zu den Softwareroutinen die am Hostsystem ablaufen. Grundsätzlich gilt allgemein betrachtet das Gleiche wie in Abschnitt 5.6 angeführt. Auch die Anforderungsliste kann als gültig betrachtet werden.

Das Hauptaugenmerk soll in diesem Abschnitt nur auf die Softwarestruktur gerichtet werden, denn es macht keinen Sinn alle Detailaufgaben in diesem Werk zu beschreiben.

An dieser Stelle möchte ich nun eine Übersicht über die Klassenhierarchie geben, um so die globale Funktionalität besser zu verstehen.

**CAboutDlg:** Darstellung und Abarbeitung des Informationsdialogfensters.

**CSerialPort:** Diese in Eigenregie erstellte Klasse hat die Aufgabe, Kommunikationsschnittstellen für andere Programmteile zur Verfügung zu stellen. Diese Klasse stellt Funktionen bereit, um eine serielle Schnittstelle zu öffnen, zu schließen und Daten zu senden oder zu empfangen.

**CCamHead:** Implementiert ein Objekt der Klasse CSerialPort und stellt Funktionen zur Kommunikation mit dem  $\mu$ Controllerboard bereit.

**CImageProc:** Auch diese Klasse wurde von Grund auf selbst erstellt und dient der Bereitstellung von Funktionen zur Bildbearbeitung.

Einige der wichtigsten Funktionen sind unter anderem Routinen die

---

<sup>12</sup>End Of Command

den Labelingalgorithmus enthalten oder Funktionen zum Speichern der Bilddaten.

**CVFWImageProcessor:** Diese selbst verfasste Klasse, welche nach zahlreichen Internetrecherchen zusammengestellt wurde, stellt Funktionalitäten zur Verfügung, um Bilder von WebCams<sup>13</sup> in den Speicher zu transferieren.

**CWintriApp:** Diese Klasse hat die Aufgabe erhalten, den Aufbau der Anwendung zu steuern und bedient sich anderer Klassen wie zum Beispiel CWintriDlg.

**CWintriDlg:** Diese Klasse beinhaltet Funktionen welche die Grundfunktionalitäten der Anwendung repräsentieren. Sie regeln unter anderem den zeitgerechten Datenfluss der Daten von der Kamera und deren anschließenden Bearbeitung. Es wurden an dieser Stelle Methoden zum automatisch generierten Teil hinzugefügt.

Eine, für den Entwickler beziehungsweise geschulten Benutzer wichtige Quellcodedatei sollte hier jedoch besonders erwähnt werden. Es handelt sich dabei um das Headerfile *Define.h*, welches wichtige Definitionen, vor allem bezüglich der Bildgröße beinhaltet.

- #define HOEHE 160
- #define MHOEHE 240
- #define BREITE 320

Mehr zu diesen Werten etwas später.

Kommen wir nun zum Ablauf einer Messwertaufnahme, der in Abbildung 6.11 zu sehen ist.

Durch entsprechendes Aktivieren (dazu später) wird ein Bild von der Web-Cam angefordert. Das Lasermodul ist zu diesem Zweck vorerst aktiv.

Eine Eigenheit der Kamera macht es nun notwendig, die Daten von dem bekannten, wenn auch für diesen Zweck unpraktischen Format *YUV420* in ein anderes zu konvertieren. Für diese Konversion bietet sich das 24Bit RGB<sup>14</sup> Format an.

---

<sup>13</sup>oder allgemein: Geräte die Video For Windows unterstützen

<sup>14</sup>RGB: Ein Bildpunkt wird durch die drei Farben Rot, Grün und Blau repräsentiert → es werden jeweils 8Bit benötigt was in Summe 24Bit Informationsvolumen bedeutet.

Es werden in dieser Anwendung zwar nur Graustufenbilder verarbeitet, was es ermöglichen würde auf 8Bit Graustufenbilder umzusteigen. Dies ist jedoch nicht der Fall, weil auf diese Option bewusst verzichtet wurde, um für weitere Aufgabenstellungen auch die Farbwerte zur Verfügung zu haben. Der nächste Schritt wird dadurch gekennzeichnet, dass zunächst das Lasermodul abgeschaltet wird und ein weiteres Bild angefordert wird, welches wieder zu konvertieren ist.



Abbildung 6.11: Flussdiagramm des Hostsystems

Von den nun erhaltenen Bildern wird die Differenz gebildet, welche anschließend einem Labelingalgorithmus unterzogen wird, um alle zusammenhängenden Bereiche festzulegen. Mit diesen zusammenhängenden Bereichen ist es

leicht möglich, eine Analyse durchzuführen und eine endgültige Auswertung anzustreben.

In Abbildung 6.12 ist dieser Sachverhalt nochmals bildlich dargestellt. Dabei ist der Werdegang der Nutzinformation (Laser) und leider auch von Bildstörungen gut dokumentiert.

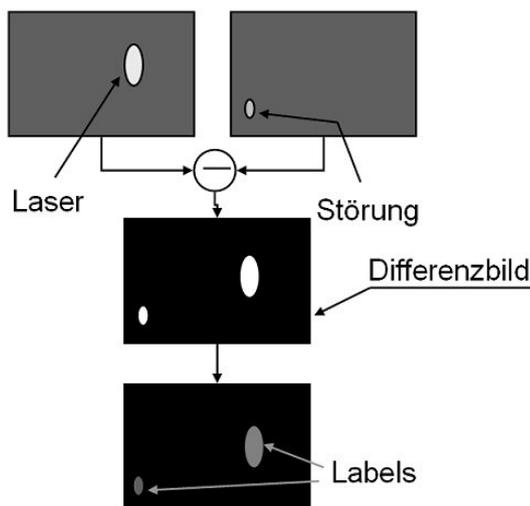


Abbildung 6.12: Ablauf der Bildbearbeitung

Eine Anmerkung möchte ich in diesem Zusammenhang noch bezüglich der Bilddaten anführen.

Betrachtet man das in Abbildung 6.12 Gezeigte, so fällt auf, dass zur Messwerterfassung lediglich Bilddaten im Bereich des Lasers relevant sind. Dieser Bereich muss natürlich ein Streifen sein, der vom linken bis zum rechten Bildrand reicht und eine gewisse Höhe hat. Man kann durch eine solche Einschränkung, wie sie in Abbildung 6.13 gezeigt ist, zwei positive Effekte erreichen:

- Die Geschwindigkeit der Bildverarbeitung wird erhöht.
- Störungen außerhalb des Streifens haben keinen Einfluss mehr auf das Messergebnis.

Nun ergeben auch die kryptischen Werte HOEHE, MHOEHE und BREITE einen Sinn, denn diese legen die Geometrien des Bildausschnitts fest, der zu verarbeiten ist.

Ein paar Worte sollten noch über die Methoden bezüglich *Video For Windows* verloren werden.

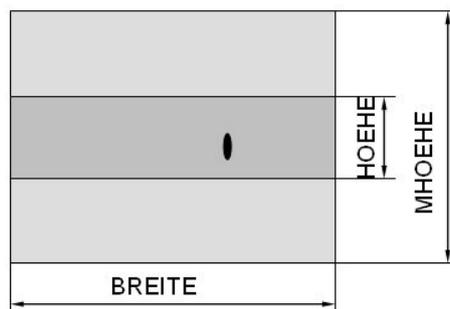


Abbildung 6.13: Darstellung des relevanten Bildausschnitts

Dieses Schnittstellensystem wird von der Firma *Microsoft* zur Verfügung gestellt um auf einfache Weise Zugriff auf Videodaten zu bekommen. Verglichen mit dem bekannten, standardisierten und bestens dokumentierten System *TWAIN* bietet dieses System die Möglichkeit, Bilder ohne Interaktion des/der Benutzers/in anzufordern<sup>15</sup>. Leider sind die API's<sup>16</sup> bei *Video For Windows* weniger gut dokumentiert und auch erhältliche Demonstrationssoftware ist in diesem Bereich eine Mangelware.

Es bedurfte also einer sorgfältigen Internetrecherche und eines gewissen Maßes an Programmieraufwand, um aus den vorhandenen Demoapplikationen ein zufriedenstellendes Ergebnis zu erhalten.

Alle relevanten Methoden und Variablen, um ein Bild von einer WebCam anzufordern, sind in der Klasse *CVFWImageProcessor* zu finden.

## 6.4 Bedienung der Software

Ich möchte nun etwas von der Struktur des Dokumentes abweichen und der Bedienung der Software, die zweifelsohne von der Softwarebeschreibung im Sinne von Quellcodekommentar zu unterscheiden ist, einen eigenen Abschnitt widmen.

Die Software läuft unter dem Namen *WinTri* und ist als dialogfensterbasierende Windowsapplikation erhältlich.

Für einen erfolgreichen Programmstart ist es jedoch erforderlich zwei Dinge vorbereitet zu haben:

- Die externe Steuereinheit muss mit dem PC verbunden und mit Strom

<sup>15</sup>TWAIN eröffnet bei einer Bildanforderung ein Dialogfenster, welches der/die Benutzer/in bedienen muss

<sup>16</sup>Application Programmers Interface

versorgt sein.

- Eine funktionstüchtige WebCam oder Ähnliches muss ebenfalls mit dem PC verbunden und mit allen nötigen Treibern versehen sein.

Sollte alles korrekt erledigt worden sein, so meldet sich nach einiger Zeit die Anwendung mit folgendem Bildschirmbild (siehe Abbildung 6.14).

Um einen einwandfreien Gebrauch zu garantieren muss zunächst die WebCam konfiguriert werden, was über die Menüleiste möglich ist.

Wie in Abbildung 6.15 zu sehen ist, stehen dazu zwei Dialogfenster bereit.

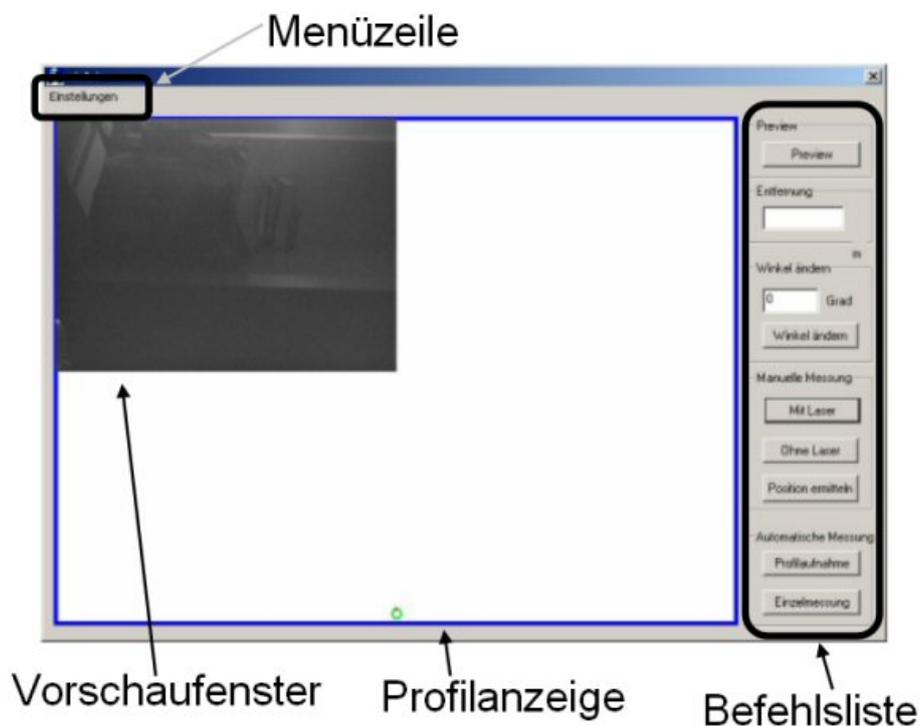


Abbildung 6.14: Bildschirm der gestarteten Anwendung



Abbildung 6.15: Menüleiste der Applikation

**Source:** Hier können Bildeigenschaften wie Helligkeit, Kontrast und vieles mehr eingestellt werden. Des Weiteren lassen sich über die unterschiedlichen Registerfenster des Dialoges auch unterschiedliche Kameraeinstellungen wie zum Beispiel die Belichtungsdauer vornehmen. Bei dieser Applikation ist es immer erforderlich, dass bei den Betriebsarten immer der *Schwarz Weiß* Modus aktiviert ist. Alle restlichen Einstellungsmöglichkeiten sind nach Bedarf durchzuführen und zu prüfen.

**Format:** Legt die aktuelle Bildgröße und das zu verwendende Format fest. In diesem Beispiel wurde eine Bildgröße von 320x240 Pixel gewählt<sup>17</sup>.

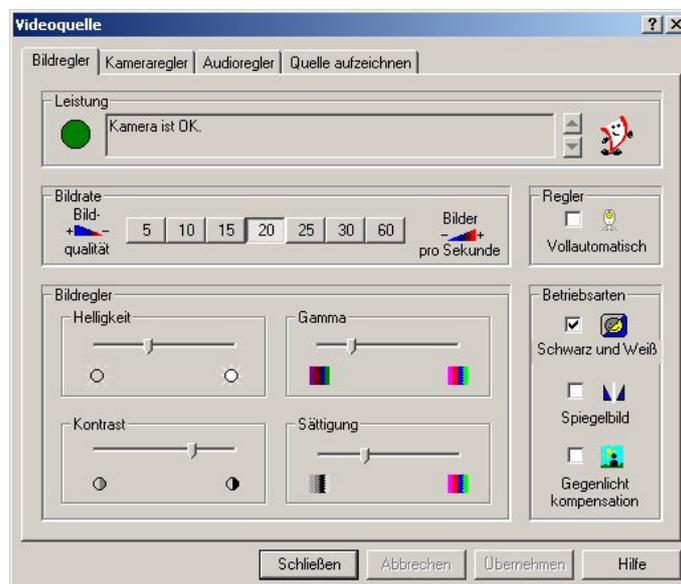


Abbildung 6.16: Dialogfenster zur Bildeinstellung

Die in Abbildung 6.16 und Abbildung 6.17 gezeigten Dialogfenster sind mögliche<sup>18</sup> Dialogfenster zur Konfiguration der Hardware.

Nachdem nun die entsprechenden Einstellungen vorgenommen wurden (in der Regel muss dies dann geschehen, wenn die Kamera das erste Mal verwendet wird oder die Einstellungen von einer anderen Applikation aus geändert wurden, eine Kontrolle schadet jedoch nie) kann das System überprüft und verwendet werden.

<sup>17</sup>Wichtig: dieser Wert muss exakt mit den Werten BREITE und MHOEHE übereinstimmen.

<sup>18</sup>je nach Hersteller der WebCam können diese Dialogfenster variieren

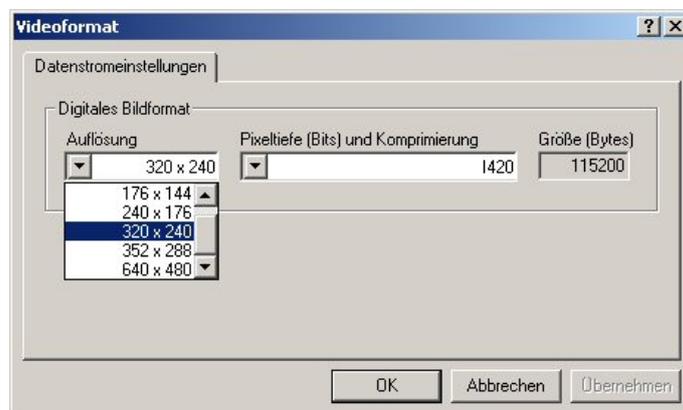


Abbildung 6.17: Dialogfenster zur Formateinstellung

Dazu bemächtigt man sich der Befehlsliste, welche folgende Funktionen parat hält:

**Preview:** Mit dieser Schaltfläche ist es möglich, die Voransicht (Preview) der WebCam ein- beziehungsweise auszuschalten, um die gesamte Fläche der Profilansicht sehen zu können. Die Voransicht ist besonders zur Kalibration der Helligkeit und des Kontrastes nützlich.

**Entfernung:** Dieses Textfeld dient lediglich zur Informationsgabe über die aktuell gemessene Distanz.

**Winkel ändern:** Mit diesem Befehlsfeld ist man in der Lage, mit dem Schrittmotor eine neue Winkelposition anzufahren. Dazu muss zunächst im Textfeld die entsprechende Position eingegeben werden und anschließend kann die Aktion durch Anwählen der Schaltfläche Winkel ändern abgeschlossen werden.

**Manuelle Messung:** Über die hier aufzufindenden Schaltflächen kann der Bildbearbeitungsablauf gemäß Abbildung 6.12 nachgestellt werden.

**Automatische Messung:** Über diese Schaltflächen kann sowohl eine Einzelmessung als auch eine Profilaufnahme initiiert werden. Die Einzelmessung muss nicht weiter erläutert werden. Die Profilaufnahme sollte jedoch schon kurz erklärt werden.

Ausgehend von  $0^\circ$  werden alle  $9^\circ$  bis zu einem Winkel von  $180^\circ$  genau zwanzig Messwerte aufgenommen und in der Profilansicht skaliert dargestellt.

Mit diesem Rüstzeug ausgestattet kann man mit der Erprobung der Software beginnen.

Anhand einiger Beispiele soll das Leistungsvermögen der Messvorrichtung veranschaulicht werden.

Als erstes Beispiel ist ein Modell einer Wandecke gegeben. Wie in Abbildung 6.18 unschwer zu erkennen ist, wurde für dieses Modell keine schöne oder neue Kartonage verwendet. Bewusst wurde nach diesem hochglänzenden Material gesucht, um das reale Verhalten dieser Messanlage zu testen.



Abbildung 6.18: Beispiel einer Profilaufnahme: Bild der Anordnung

Das Messergebnis dieser Versuchsreihe ist in Abbildung 6.19 zu sehen (Dabei enthält der, in Abbildung 6.18 mit dem Pfeil gekennzeichnete Kreis, im Inneren den Auftreffpunkt des Laserstrahls.).

Auf den ersten Blick ist natürlich die Dreiecksform der Messanordnung zu erkennen. Daneben fällt zunächst jedoch auch noch eine gewisse Krümmung der gemessenen Wandflächen auf - hat das System auf so kurze Distanzen etwa eine eklatante Messungenauigkeit?

Ich darf den Leser beruhigen, denn es gibt zusätzlich zur zugegebenermaßen nicht sehr geraden Wandfläche noch mehrerer Erklärungen dafür, die dieses Verhalten erklären. Ein wesentlicher Grund ist, dass anstatt mit einem Millimeterraster lediglich eine Auflösung von Zentimetern verwendet wurde und daher eventuell Rundungen entstehen können.

Eine weitere Ungenauigkeit ist dadurch zustande gekommen, dass bei der Umrechnung von Polarkoordinaten in kartesische Koordinaten dieser Rundungsfehler nochmals verstärkt wird. Auch weist die automatische Konversion von Gleitkommazahlen in Ganzzahlen einen Fehler in dieser Kommandostruktur auf. Letztlich wird auch der Umstand entscheidend sein, dass die glänzende und etwas geknickte Wandfläche sein Scherflein zur Messwertverzerrung bei-

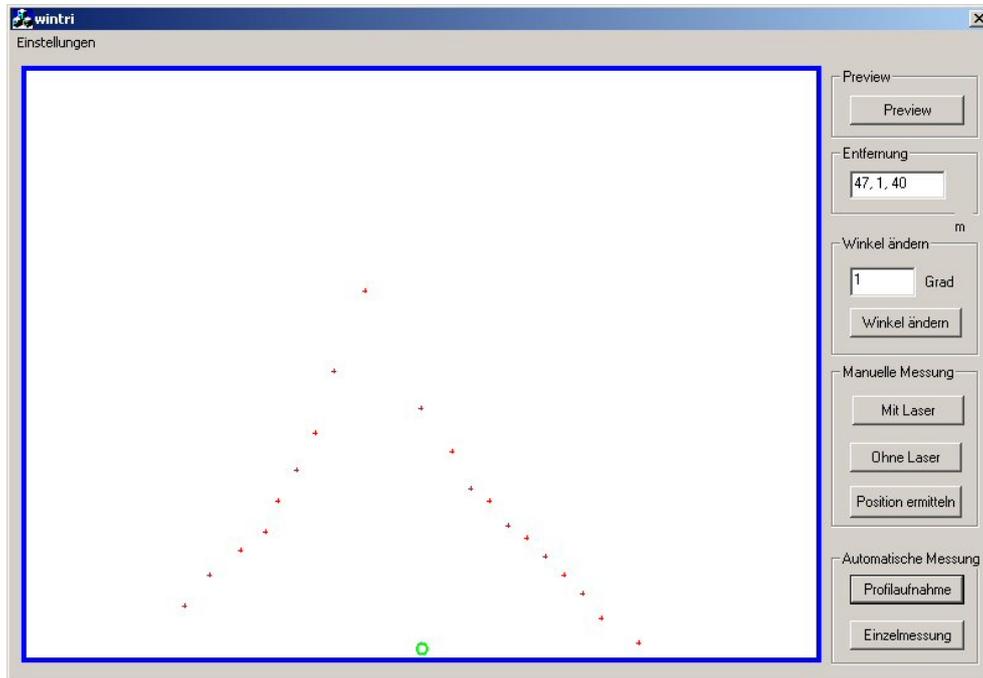


Abbildung 6.19: Beispiel einer Profilaufnahme: Modell einer Raumecke

tragen wird.

Als Gesamtergebnis betrachtet ist dieses System jedoch trotzdem in der Lage, hervorragende Ergebnisse zu liefern, die durch entsprechende Maßnahmen noch verbessert werden könnten (insbesondere Rundungsfehler usw.)

## 6.5 Bewertung der Messvorrichtung

Abschließend möchte ich noch einige Worte zum vorgestellten Messaufbau und dessen Charakteristik anführen.

Der im Kapitel 2.3 angeführte Kennlinienverlauf kann für dieses Messinstrument konkretisiert werden, was in Abbildung 6.20 zu sehen ist.

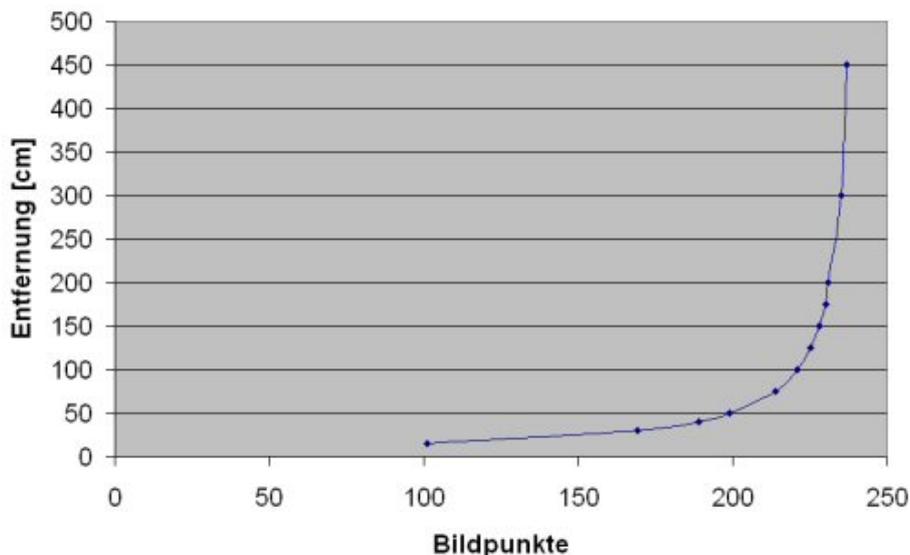


Abbildung 6.20: Mögliche Kennlinie der Anwendung

Diese Kennlinie<sup>19</sup> entspricht genau dem Erwarteten. Sie weist bei kleinen Distanzen eine geringe Steigung auf, um auf Distanzänderungen präzise reagieren zu könne. Zusätzlich fällt noch der annähernd lineare Verlauf der Kennlinie bis etwa 50cm auf.

Im Bereich der großen Distanzen (ab etwa 100cm) verhält sich dieser Umstand jedoch anders. Hier existiert ein Zusammenhang zwischen Messwert und tatsächlicher Distanz der ebenfalls einen linearen Charakter aufweist, jedoch eine sehr hohe Steigung hat. Dies wirkt sich in der Tat auch so aus, dass das Auflösungsvermögen sehr drastisch herabgesetzt wird.

Sind es im Nahbereich durchaus Auflösungen von 5mm, so kann man im Fernbereich (ab 2m) davon ausgehen, dass nur noch Distanzunterschiede von 25cm festgestellt werden können.

Man nun behaupten, dass ein Erhöhung der Bildauflösung (möglich wären 640x480 Bildpunkte) eine Verbesserung der Systemcharakteristik mit sich

<sup>19</sup>gilt für eine Bildauflösung von 320x240 Bildpunkten und dem festgelegten Neigungswinkel zwischen Lasermodul und WebCam

bringen würde, was natürlich auch stimmt.

Das wäre aber nur die halbe Wahrheit. Im Nahbereich hätte man tatsächlich ein besseres Verhalten, das jedoch für unsere Anwendungen nicht verlangt werden muss und man auch keinen Nutzen davon hat.

Für den Fernbereich würde sich jedoch nichts mehr ändern, mehr noch hat sich das System als störanfällig erwiesen.

Grund dafür ist die Tatsache, dass das Bild mehr Details der Umgebung aufnehmen kann. Bewegt sich nun die Messanordnung (was beim Rollstuhl sicher der Fall ist), so ergeben sich aufgrund der Differenzbilder sehr viele Störstellen die sich unter Umständen nicht mehr korrigieren lassen.

Dies nahm ich zum Anlass die maximale Bildauflösung unterhalb des Maximalwertes anzusiedeln. Auch konnte damit eine Erhöhung der Bildrate erzielt werden, was sich in einer höheren Rate der Messwertabtastung niederschlug.

Passend zu dem gerade Erwähnten, möchte ich gleich den großen Schwachpunkt des Systems ansprechen. Es handelt sich dabei um die Notwendigkeit der Bildung eines Differenzbildes, um den Laserpunkt zu finden. Funktioniert dies bei stehender Umgebung beziehungsweise stillstehendem Sensor sehr gut, stellt dies im Falle einer Bewegung ein großes Hindernis dar. Grund ist die zu geringe Bildrate der WebCam. Die Wegstrecke die zwischen den beiden Bildaufnahmen zustandekommt ist dann so groß, dass sich bei der Differenzbildung keine Bereiche mehr decken und somit das Verfahren versagt.

Möglichkeiten um dem gegenzusteuern wären, wie anfangs erwähnt spezielle Framegrabberkarten die viel höhere Bildübertragungsraten aufweisen oder die Angleichung der beiden für die Differenzbildung herangezogenen Bilder durch Korrelation. Auch wäre die Verwendung von Filteroptiken und der vollständige Verzicht auf eine Differenzbildung (der Rest der Anwendung wäre immer noch gleich) geeignet, um ein sauberes Ergebnis zu erhalten.

Die nächste Erkenntnis, die aus oben Gesagtem gewonnen werden kann, ist die zu hohe Messzeit. Mathematisch lässt sich zunächst nur folgender Zusammenhang darstellen, wie er in Gleichung 6.1 zu sehen ist ( $t_{fps}$  ist dabei die Bildübertragungsrate).

$$T = \frac{1}{t_{fps} \cdot 2} = \frac{1}{30} \cdot 2 = 0.06666 = 0.07s = 70ms \quad (6.1)$$

Es ergibt sich zunächst eine Messzeit von 70ms, was ja nur doppelt so groß wäre wie in Kapitel 5 erreicht.

Berücksichtigt wurde jedoch noch nicht, dass erstens eine gewisse Rechenzeit zur Auswertung erforderlich ist und zweitens die Anforderung der Bilddaten mittels *Video For Windows* etwas Zeit in Anspruch nehmen wird.

Es wurde nun durch Messungen festgestellt, dass im günstigsten Fall 200ms-250ms benötigt werden. Nach eingehender Untersuchung und Recherche hat sich der Verdacht erhärtet, dass der Hauptverantwortliche für diese Misere das System rund um *Video For Windows* ist. Hier kommen leider alle Nachteile von Windows und dessen aufgeblähter Treiberhierarchie zum Tragen, die derart hohe Verzögerungen hervorrufen.

Als Resümee ist nun anzuführen, dass das Messverfahren und der Gesamtaufbau grundsätzliche einwandfrei funktioniert. Lediglich die angesprochenen Probleme trüben das Bild etwas, es sei jedoch nochmals vermerkt, dass der Rest der Anlage bestens funktioniert und ausgezeichnete Ergebnisse liefert.

# Kapitel 7

## Messsystem zur Bodenabtastung

### 7.1 Einführung und Aufgabenstellung

#### 7.1.1 Aufgabenstellungen

In diesem abschließenden Kapitel wird der Aufbau einer Messanlage zur Abtastung des Bodens vor dem Rollstuhl diskutiert.

Es soll mit dieser Messeinrichtung möglich sein, den Fußboden kontinuierlich zu beobachten, um potenzielle Gefahrenherde des Alltags (Stufen, Objekte, Haustiere, ...) ausfindig zu machen, um je nach Größe und Form des Gegenstandes Entscheidungen bezüglich einer Ausweichstrategie oder eines Bremsvorganges treffen zu können.

Aussagen bezüglich der Größe sind, wie noch erläutert wird, mit Vorsicht zu genießen, denn es können sehr irreführende Situationen auftreten.

#### 7.1.2 Funktionsanalyse

##### Klärung der Rahmenbedingungen

Es stellt sich nun die Frage wie weit die Umgebung, gemessen von der Rollstuhlvorderkante von Interesse ist. Nun gilt grundsätzlich, von Interesse ist alles, was sich in einer Entfernung befindet, welche dem Bremsweg entspricht. Messungen haben ergeben, dass der für meine Versuche verwendete Rollstuhl bei einer Vollbremsung ausgehend von der maximalen Geschwindigkeit (5km/h) bis zum Stillstand zirka einen Meter benötigt. Wichtig dabei ist natürlich noch die Angabe der Masse der Testperson, welche zu diesem Zeitpunkt 85kg betrug.

In dieser Arbeit wird also davon ausgegangen, dass der Bremsweg einen Meter beträgt. Es hätte jedoch fatale Folgen, würde man schon diesen Wert als Detektionsgrenze festlegen, denn es muss selbstverständlich noch eine Reaktionszeit des Gesamtsystems berücksichtigt werden.

Zunächst wollen wir die maximale Geschwindigkeit wie in Gleichung 7.1 zu sehen ist in anderen Einheiten darstellen:

$$v_{max} = 5 \frac{km}{h} = \frac{5}{3.6} \frac{m}{s} = 1.37 \frac{m}{s} \quad (7.1)$$

Mit diesem Wert ist es nun möglich, die Dauer einer Notbremsung anzugeben, was in Gleichung 7.4 zu sehen ist. Um diesen Wert zu erhalten, muss zunächst durch Auflösen des Gleichungssystems, welches in den Gleichungen 7.2 und 7.3 zu sehen ist, eine geeignete Berechnungsgrundlage gefunden werden (Die Bedeutungen der verwendeten Variablen sind in Tabelle 7.1 zu finden).

$$v = \sqrt{2 \cdot a \cdot s} \Rightarrow a = \frac{v^2}{2 \cdot s} \quad (7.2)$$

$$a = \frac{v}{t} \quad (7.3)$$

$$t_{Notbremsung} = \frac{2s}{v_{max}} = \frac{2 \cdot 1m}{1.4 \frac{m}{s}} = 1.43s \quad (7.4)$$

Symbol	Bezeichnung	SI-Einheit
v	Geschwindigkeit	$\frac{m}{s}$
s	Wegstrecke	m
t	Zeit	s

Tabelle 7.1: Verwendete Variablen

Wie man sieht, benötigt man für eine Notbremsung unwesentlich mehr als 1,4 Sekunden. Nehmen wir nun an, es ist ein Reaktionsweg von 20cm ausreichend. Wie lange die Reaktionszeit bei einem Reaktionsweg von 20cm wäre, ist in Gleichung 7.5 zu sehen ist. Man erkennt, dass diese Zeitspanne etwa 10% der Bremszeit ausmacht und 20% des Bremswegs beträgt und somit einen vernünftigen Wert darstellt. Auch ist es mit den bereits bekannten Messmethoden (siehe Abschnitt 6.5) möglich innerhalb dieser Zeiten zu reagieren.

$$t_{Reaktionszeit} = \frac{20cm}{v_{max}} = 0.143s \quad (7.5)$$

### Struktur der Messeinrichtung

Wie aus den obigen Anmerkungen unschwer herauszulesen ist, wird der Boden vor dem Rollstuhl in einem Abstand von 1,2 Metern kontinuierlich abgetastet. Nun könnte man annehmen, dass ein System, welches in Kapitel 6 beschrieben wurde hier seinen Dienst verrichten könnte, was jedoch nicht der Fall ist.

Grund dafür ist, dass dieses Messverfahren nur punktuelle Messwerte aufnehmen kann. Der Schwenkantrieb könnte hier weiterhelfen, wäre jedoch viel zu langsam, um in der geforderten Reaktionszeit auch nur annähernd ein gültiges Profil erstellen zu können. Es ist also eine neue Messeinrichtung aufzubauen. Ich möchte den Leser nicht auf die Folter spannen und gleich das Endergebnis präsentieren.

Grundsätzlich ist zu sagen, dass auch für diese Anwendung die Lasertriangulation zielführend sein wird.

Es haben sich jedoch zwei wesentliche Dinge geändert:

- Es wird ein Linienlaser anstatt eines Punktlasers verwendet.
- Der ganze Rollstuhl ist nun Teil des Sensors.

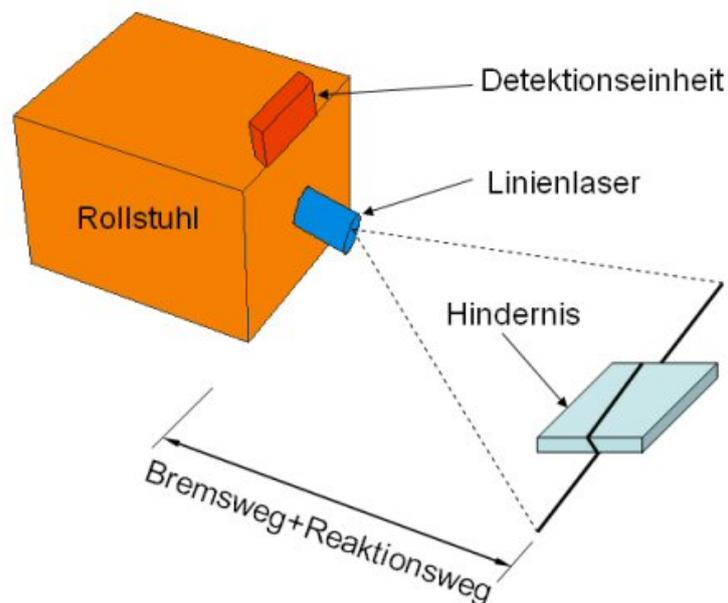


Abbildung 7.1: Schemenhafter Aufbau der Messvorrichtung

Die Verwendung eines Linienlasers ermöglicht es, das ganze Profil in einem Arbeitsdurchgang aufzunehmen. Dies bedeutet nun, dass eine zusätzliche

Schwenkeinrichtung entfallen kann und somit einfacher wird. Nachteilig ist, dass die Messrichtung niemals geändert werden kann und die Messanlage nur für diesen einen Zweck verwendet werden kann.

Der zweite Punkt betrifft den mechanischen Aufbau der Messeinrichtung. Man bedient sich dabei des ganzen Rollstuhls um den geforderten Geometrien (siehe Kapitel 2.3) der Lasertriangulation gerecht zu werden. Eine Skizze des angestrebten Aufbaus ist in Abbildung 7.1 zu sehen. Dieses Bild enthält alle benötigten Komponenten und zeigt deren ungefähre Platzierung. Auch zu sehen sind die Auswirkungen eines eventuellen Hindernisses auf die Laserlinie. Dabei wird die Laserlinie unterbrochen und verläuft entlang der Hindernisoberfläche, was mit der Detektionseinheit zu erfassen ist.

## 7.2 Beschreibung der Komponenten

Führt man sich nochmals den schematischen Aufbau der Messvorrichtung vor Augen, so erkennt man lediglich drei Komponenten die in Hardware ausgeführt werden müssen.

Diese Komponenten sind:

- Linienlaser
- Detektionseinheit
- Steuereinheit zur Lasersteuerung

Die folgenden Textpassagen beschäftigen sich kurz mit der Beschreibung der Komponenten und deren Verwendung. Es ist jetzt schon anzumerken, dass sich verglichen zum Kapitel 6 weit weniger Hardwareaufwand verzeichnen lässt und daher dieser Punkt rasch abgehandelt werden kann.

### 7.2.1 Linienlaser

Wie schon mehrfach erwähnt, wurde zur Generierung der Linie ein Linienlaser verwendet. Diese Bauteile gibt es in verschiedenen Ausführungsformen und Leistungsklassen. Mit Ausführungsformen ist an dieser Stelle jedoch primär nur die Art der Linienprojektion gemeint. Dies bedeutet konkret den Öffnungswinkel  $\alpha$  der ausgesendeten Linie, der je nach Hersteller und Type variieren kann (siehe Abbildung 7.2). Gängige Winkel sind dabei:  $60^\circ$ ,  $70^\circ$ ,  $90^\circ$ ,  $100^\circ$  und  $120^\circ$ .

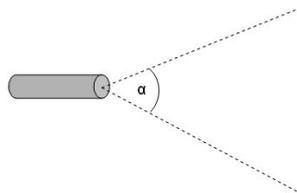


Abbildung 7.2: Öffnungswinkel  $\alpha$  des Lasermoduls

Auch gibt es Unterschiede in der Leistungsaufnahme beziehungsweise der optischen Leistungsabgabe. Wie sehr dieses Kriterium die projizierte Linie beeinflusst, konnte vor dem Kauf der Einheit jedoch nicht gesagt werden und war zunächst von nur geringer Bedeutung.

Viel wichtiger war es, ein Exemplar mit passendem Öffnungswinkel zu bekommen. Um hier einen einigermaßen richtigen Wert der benötigten Linienlänge<sup>1</sup>

---

<sup>1</sup>hängt vom verwendeten Öffnungswinkel  $\alpha$  ab

bei einer Projektionsdistanz von 1.2 Metern (ist nicht ganz richtig, denn 1.2 Meter ist der Abstand von der Rollstuhlvorderkante zur Linie; der Fehler ist jedoch nur minimal und für eine erste Abschätzung belanglos) zu erhalten, wurde mit der WebCam aus Kapitel 6 eine gezeichnete Linie im Abstand von 1.2 Metern vermessen. Die Länge dieser Linie betrug dabei 90cm. Zur Berechnung des notwendigen Winkels kann Abbildung 7.3 zur näheren Erklärung herangezogen werden.

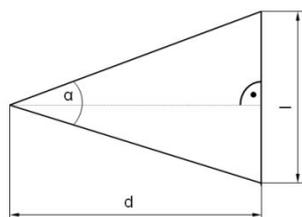


Abbildung 7.3: Geometrieübersicht der Laserprojektion

Wie man auf den ersten Blick sieht, sind die Größen Projektionsdistanz  $d$  und die Linienlänge  $l$  gegeben. Mit der Tangens-Funktion kann nun die fehlende Größe, das ist der gesuchte Winkel  $\alpha$ , berechnet werden (siehe Gleichung 7.6).

$$\alpha = 2 \cdot \arctan \frac{\frac{l}{2}}{d} = 2 \cdot \arctan \frac{\frac{900\text{mm}}{2}}{1200\text{mm}} = 41,11^\circ \quad (7.6)$$

Die Berechnung hat ergeben, dass eine mindeste Winkelöffnung von  $41.11^\circ$  benötigt wird um die geforderte Länge der Linie zu erhalten.

Mit diesem Wert im Hinterkopf konnte ein entsprechendes Lasermodul gesucht werden. Schlussendlich wurde ein Modul der Firma *Conrad Elektronik* gewählt, welches einen Öffnungswinkel von  $70^\circ$  hat und eine optische Leistung von unter 1mW besitzt. Dieses Modul hat zwar zirka  $30^\circ$  zuviel an Winkelöffnung, was nicht weiter hinderlich ist, war aber eines der preiswertesten und daher für einen Erstversuch bestens geeignet.

## 7.2.2 Detektionseinheit

An dieser Einheit hat sich verglichen zum Abschnitt 6.2.2 nichts geändert. Alle zuvor gegebenen Erklärungen und Kommentare zu diesem Themenpunkt können vollständig übernommen werden.

### 7.2.3 Steuereinheit

Es soll hier nochmals erwähnt werden, dass die Notwendigkeit einer Schwenkeinrichtung für dieses Verfahren nicht mehr gegeben ist.

Dies alleine macht jedoch nicht die ganze Steuereinheit überflüssig, denn die Aufgabe der Lasersteuerung ist immer noch aktuell. Man könnte nun die Software der Steuereinheit dahingehend vereinfachen und alle Teile der Schrittmotorsteuerung entfernen, was jedoch nicht getan wurde. Grund dafür ist, dass genügend Speicher zur Verfügung steht und die Routinen nicht weiter störend sind.

Mehr muss über diesen Punkt auch nicht mehr gesagt werden und ich verweise für nähere Informationen auf den Abschnitt 6.2.3.

## 7.3 Beschreibung der Software

### 7.3.1 Beschreibung der Firmwareroutinen

Siehe Abschnitt 7.2.3.

### 7.3.2 Beschreibung der Routinen des Hostsystems

Wie schon in den anderen Kapiteln üblich, wird zu Beginn eine Übersicht über die Klassenstruktur gegeben, was auch hier der Fall ist.

**CAboutDlg:** Siehe Abschnitt 6.3.2.

**CSerialPort:** Siehe Abschnitt 6.3.2.

**CCamHead:** Siehe Abschnitt 6.3.2.

**CImageProc:** Siehe Abschnitt 6.3.2. Zusätzlich wurden noch einige Methoden hinzugefügt, um die neuen Aufgaben bewältigen zu können.

**CVFWImageProcessor:** Siehe Abschnitt 6.3.2.

**CWintriDlg:** Siehe Abschnitt 6.3.2.

Wie man merkt, unterscheidet sich die Klassenstruktur nur geringfügig von der vorgestellten Struktur im Kapitel 6.

Ebenfalls gleich geblieben ist die Verarbeitungskette wie sie in Abbildung 6.12 zu sehen ist. Unterscheiden wird sich hier lediglich die Form des Lasers, der vorher punktförmig gewesen ist und nun einer Geraden beziehungsweise einem unterbrochenen Linienzug entspricht.

Als Zwischenergebnis liegt jedenfalls ein Bild mit den enthaltenen *labels* vor, welches essentiell für die weiteren Schritte sein wird.

### Lineare Regression

Wie erhält man nun aus den digitalisierten Bilddaten den begehrten Linienzug? Nun dafür gibt es die Methode der linearen Regression, doch sehen wir uns zunächst Abbildung 7.4 an.

Wie zu erkennen ist, wird aus den diskreten Messwerten durch mathematische Umformungen eine Gerade durch das Messwertfeld gelegt. Man nennt

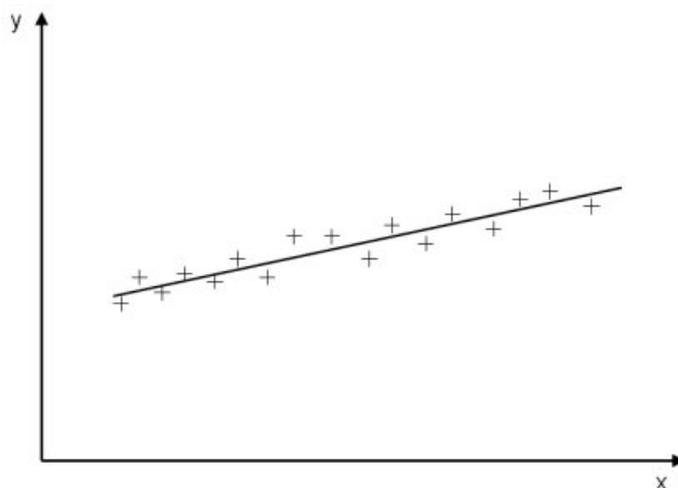


Abbildung 7.4: Beispiel einer linearen Regression

diese Gerade die Regressionsgerade und ist durch Gleichung 7.7 mathematisch definiert.

$$y = a + b \cdot x; b = \frac{S_{xy}}{S_{xx}}; a = \bar{y} - b \cdot \bar{x} \quad (7.7)$$

mit

$$S_{xy} = \sum (x_i - \bar{x})(y_i - \bar{y}) \quad (7.8)$$

$$S_{xx} = \sum (x_i - \bar{x})^2 \quad (7.9)$$

Nun wissen wir jedoch, dass bei der Aufnahme und Weiterverarbeitung der Laserlinie keine wirkliche Linie sondern eine rechtecksförmige<sup>2</sup> Ansammlung von Punkten mit Liniencharakter vorliegt.

Man muss also durch Mittelwertbildung diese Werte diskretisieren wie es in Abbildung 7.5 zu sehen ist und kann diese anschließend dem Regressionsverfahren unterziehen.

### Rekonstruktion des Geradenverlaufes

Soweit ist dieses Verfahren ganz einfach. Leider versagt das Verfahren bei zusammenhängenden, geknickten Linienzügen, wie es in Abbildung 7.6 zu sehen ist.

<sup>2</sup>wenn man in erster Näherung davon ausgeht, dass die Laser-Linie auf annähernd ebene Oberflächen fällt

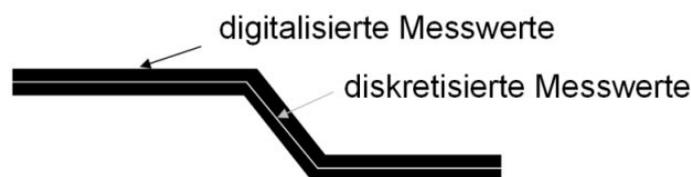


Abbildung 7.5: Diskretisierung der Messwerte

Grund dafür ist, dass im linken, fehlerhaften Teil die Messwerte zweier unterschiedlicher Linienzüge verwendet wurden, um eine *einzig*e Gerade zu bekommen. Richtig muss es natürlich sein, diese beiden Geradenstücke getrennt von einander zu behandeln, was in der rechten Bildhälfte zu sehen ist.



Abbildung 7.6: Mögliche Szenarien bei der Bildung der Regressionsgeraden

Dieses Problem stellte sich als echte Herausforderung dar und bedurfte einiger Anstrengungen, um ein zufriedenstellendes Ergebnis zu erhalten. Dazu wird der gesamte Linienzug in Teilstücke zu je 10 Pixel Länge unterteilt und jedes Teilstück für sich alleine in eine Linie umgewandelt. Der nächste Schritt besteht nun darin, benachbarte Linienelemente mit gleicher Steigung und gleicher Verschiebung zusammenzufassen und in einem Linienfeld abzuliegen.

Die weitere Vorgehensweise der Visualisierungssoftware besteht nun darin, diese ausgefilterten Liniendaten graphisch darzustellen und zu analysieren. Die Analyse hat das Ziel, Nullpunktverschiebungen zu bewerten um damit mögliche Objekte, die den Laserstrahl kreuzen, erkennen zu können.

## 7.4 Bedienung der Software

Wie auch schon bei den anderen Beispielapplikationen erwähnt, läuft diese Anwendung unter Windows und benötigt für den richtigen Gebrauch neben einer aktiven WebCam noch die betriebsbereite Steuereinheit, welche an der seriellen Schnittstelle mit der Bezeichnung *COM1* angeschlossen ist. Weitere Anmerkungen dazu sind im Abschnitt 6.4 zu finden.

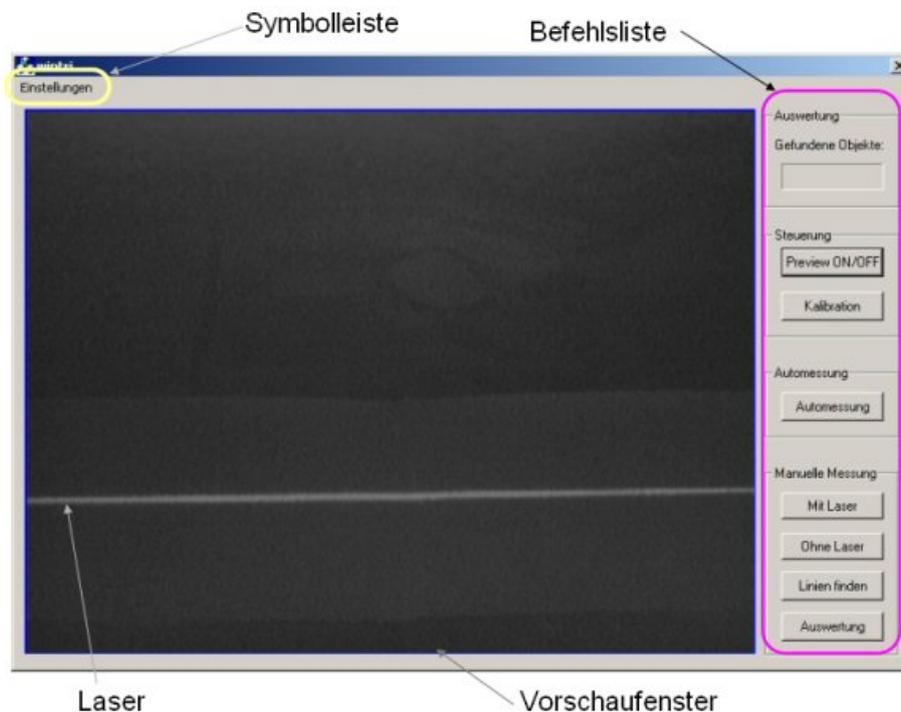


Abbildung 7.7: Startschirm der Anwendung winfloor

Auch in der Struktur der Anwendung hat sich nichts geändert. Es gibt auch hier wieder eine Menüleiste und die Befehlsliste am rechten Rand. Ist die Funktionalität der Menüleiste gleich geblieben (siehe Abschnitt 6.4), hat sich bei der Befehlsliste doch einiges geändert (zum Beispiel sind die Eingabefelder zur Winkeländerung weggefallen). Die Bedeutungen der restlichen Bedienelemente lauten wie folgt:

**Preview:** Mit dieser Schaltfläche ist es möglich die Voransicht (Preview) der WebCam ein- beziehungsweise auszuschalten um die gesamte Fläche der Profilansicht sehen zu können. Die Voransicht ist besonders zur Kalibration der Helligkeit und des Kontrastes nützlich.

**Manuelle Messung:** Mit dieser Schaltfläche ist es möglich, eine manuelle Messung vorzunehmen.

**Automessung:** Damit wird die gesamte Messkette, wie oben beschrieben, angestoßen.

Analog zu den vorangehenden Kapiteln, können nun Beispielmessungen präsentiert werden, was auch der Fall ist.

Die erste Beispielmessung wurde bei verdunkeltem Raum durchgeführt. Weiters wurde die Projektionsdistanz in diesem Beispiel auf 80cm verkürzt. Eine weitere wichtig Angabe betrifft den Abstand zwischen Lasermodul und Detektionseinheit, welcher 40mm beträgt. Dieser Wert erscheint schon jetzt als sehr gering und es ist zu bezweifeln, ob mit diesem Parameter eine vernünftige Messung möglich ist.



Abbildung 7.8: Voransicht der ersten Messung (Abstand Lasermodul und Kamera: 40mm)

Wir können zunächst in Abbildung 7.8 die aufgenommene Laserlinie sehen. Wie man richtig erkennt, verursacht das Hindernis, welches eine Höhe von 4cm aufweist tatsächlich nur eine geringe Verschiebung des unterbrochenen Liniezuges.

Besser ist diese Verschiebung in der Abbildung 7.9 zu sehen, welche die fertig analysierte Laserlinie darstellt. Eingezeichnet wurde weiters die Sprungstelle, an der eine Kante des Hindernisses verläuft. Auch hier ist deutlich zu sehen, dass auf Grund des geringen Konstruktionsabstandes zwischen Laser und WebCam ein sehr schleifender Übergang zustande kommt, was eine Objekterkennung sehr schwierig macht.

Für den zweiten Messversuch wurde der Abstand zwischen WebCam und Lasermodul auf 11cm erhöht. Auch wurde der Projektionsabstand des Lasers

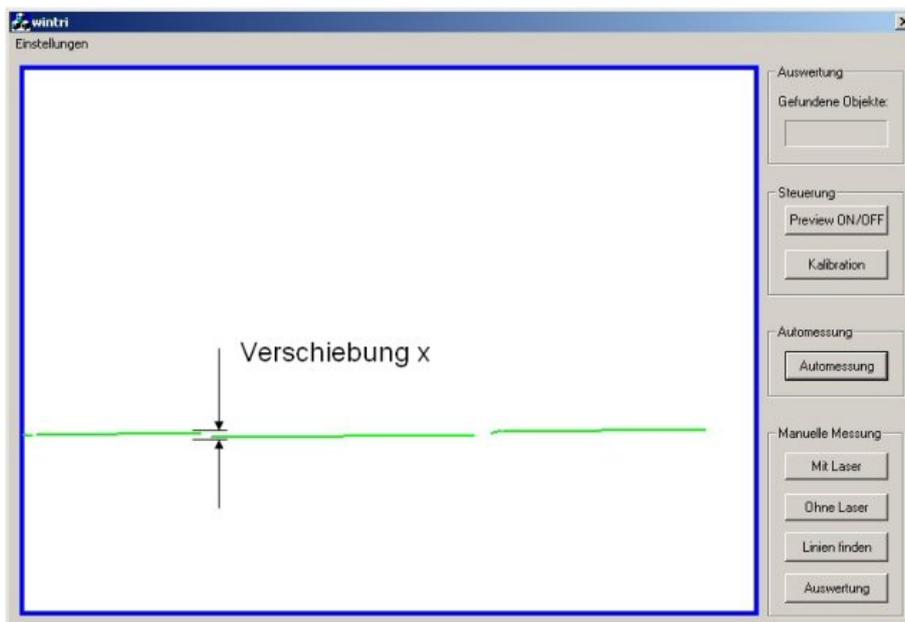


Abbildung 7.9: Auswertungsbild der ersten Messung Abstand Lasermodul und Kamera: 40mm)

auf 120cm vergrößert. Das zu detektierende Objekt wurde hingegen nicht verändert.

Grundsätzlich kann man annehmen, dass durch die Vergrößerung des Abstandes zwischen WebCam und Laser die Objekterkennung verbessert wird, während die Verschiebung der Projektionslinie eine größere Unsicherheit bedeutet sollte.

Doch betrachten wir zunächst das Vorschaubild der Anwendung, welches in Abbildung 7.10 zu sehen ist (es ist nur die untere Hälfte des Bildes zu sehen).

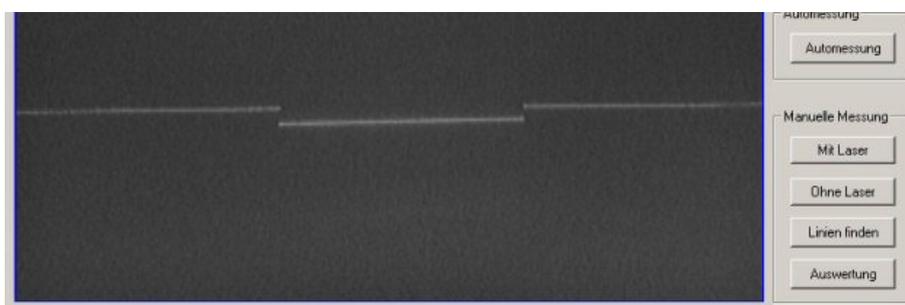


Abbildung 7.10: Voransicht der zweiten Messung Abstand Lasermodul und Kamera: 110mm)

Wie in diesem Bild deutlich zu erkennen ist, verursacht das Objekt eine merkliche Verschiebung der Laserlinie, was hoffen lässt, dass die anschließende Auswertung ebenfalls ein eindeutigeres Ergebnis bringen wird.

Der Leser kann sich von der Richtigkeit dieser Annahme durch betrachten der Abbildung 7.11 überzeugen.

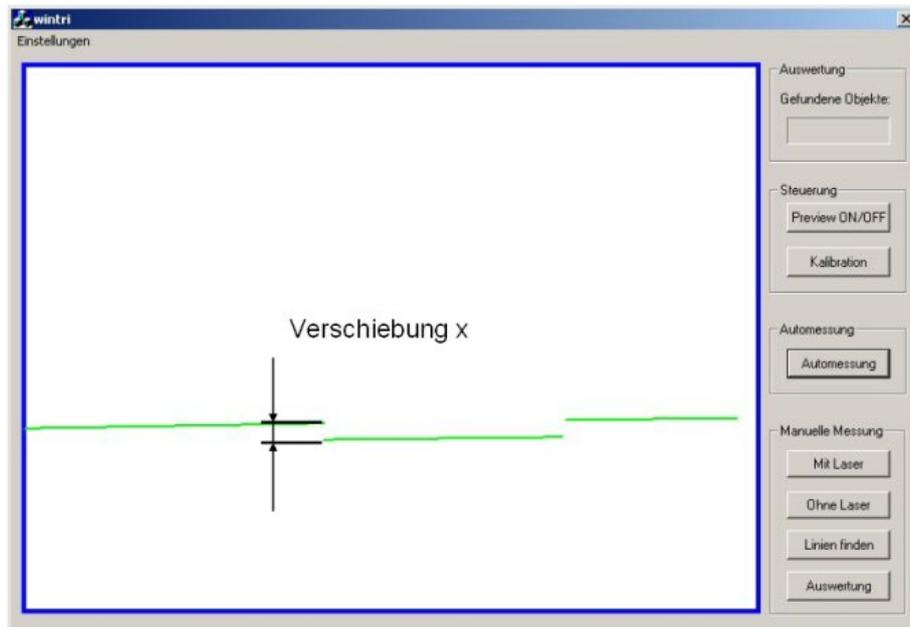


Abbildung 7.11: Auswertungsbild der zweiten Messung Abstand Lasermodul und Kamera: 110mm)

Deutlich ist der Abstand zwischen den Linien zu sehen, was eine genaue Detektion ermöglicht. Der zweite Punkt der erwähnt werden sollte ist, dass die Versetzung der Projektionslinie keinen störenden Einfluss auf das Ergebnis hat (Man muss sich jedoch immer vor Augen halten, dass diese Messungen in einem verdunkelten Raum, also bei günstigen Lichtverhältnissen stattgefunden haben.).

## 7.5 Bewertung der Messvorrichtung

Bezüglich des Schwachpunkts WebCam sei auf den Abschnitt 6.5 hingewiesen, denn dort wurde dieses Thema ausführlich behandelt.

Wichtiger erscheint es mir hingegen, Kommentare bezüglich des Verfahrensweges zu geben.

Im überwiegendem Maße ist festzustellen, dass dieses Verfahren recht zuverlässig arbeitet und als Grundlage für weitere Forschungen dienen kann. Kritisch betrachtet kann es jedoch in manchen Fällen zu irreführenden Ergebnissen kommen, was jedoch nichts mit dieser Implementierung gemein hat, sondern prinzipbedingt zu erklären ist.

Im Zusammenhang mit dem verwendeten Lasermodul ist nochmals anzumerken, dass dieses zu leistungsschwach ist und nur bis Distanzen von etwa 70cm zuverlässig einzusetzen ist. Bei darüberliegenden Projektionsdistanzen kann es insbesondere bei schlechten Lichtverhältnissen zu keiner eindeutigen Rekonstruktion zufolge fehlendem Kontrast mehr kommen<sup>3</sup>.

Es wäre also angebracht ein leistungsstärkeres Derivat zu verwenden, um so immer brauchbare Ergebnisse zu erhalten.

Aus dem direkten Vergleich der beiden Messreihen ist weiters zu schließen, dass eine sinnvolle Messung nur mit einem großen Abstand zwischen WebCam und Lasermodul möglich ist. 110 Millimeter sind in diesem Fall ein Richtwert der gute Messergebnisse erwarten lässt.

Als letztes möchte ich noch erwähnen, dass zur Verbesserung der Messung ein Rotfilter<sup>4</sup> getestet wurde. Dieses Filterelement hatte jedoch keine Auswirkungen (weder positive noch negative) auf das Ergebnis und kann daher mit ruhigem Gewissen eingespart werden.

---

<sup>3</sup>Wie gezeigt wurde, ist es jedoch bei günstigen Verhältnissen durchaus möglich Distanzen bis zu 120cm zuzulassen.

<sup>4</sup>*Kodak Wratten* gelatin filter Nr.: 25

# Abbildungsverzeichnis

2.1	Schematischer Aufbau . . . . .	8
2.2	Schematischer Aufbau eines Triangulationssystemes . . . . .	10
2.3	Skizze der wichtigsten Größen . . . . .	11
2.4	Lambertstrahler . . . . .	12
2.5	Mögliche Sensoranordnung . . . . .	13
2.6	Aufbau eines PSD . . . . .	14
2.7	Mögliche Sensoranordnung mit PSD . . . . .	16
2.8	Aufbau eines CCD Zeilensensors . . . . .	16
2.9	Möglicher Kennlinienverlauf bei Triangulation . . . . .	18
2.10	Abschattung bei Triangulation . . . . .	19
3.1	Blockschaltbild des Sensors . . . . .	21
3.2	Abstandsmessung durch Triangulation . . . . .	22
3.3	Timingdiagramm für das Auslesen der Daten . . . . .	23
3.4	Kennlinie des Sensors . . . . .	24
3.5	Ausgangsgröße in Abhängigkeit des Detektionsabstandes . . . . .	25
3.6	Blendensystem zur Kennlinienfeldermittlung . . . . .	26
3.7	Beschaltung des GP2DXX . . . . .	27
3.8	Pinbelegung . . . . .	27
3.9	Kennlinie des analogen Distanzsensors . . . . .	28
3.10	Zeitverhalten des analogen Distanzsensors . . . . .	29
3.11	Verhalten bei zeitlich schnell veränderlichen Messbereichen . . . . .	30
4.1	Schematischer Aufbau einer Puls-Echo-Messanlage mit zwei getrennten Schallwandlern . . . . .	37
5.1	Schematisch- abstrakter Aufbau dieser Messanlage . . . . .	41
5.2	Photo des Distanzsensors GP2D120. Rechtes Bild zeigt den geöffneten Sensor. . . . .	44
5.3	Skizze mit den wichtigsten Abmessungen des Distanzsensors GP2D120 . . . . .	44

5.4	Skizze mit den elektrischen Anschlüsse des GP2D120 . . . . .	45
5.5	Photo der Universalaufnahme . . . . .	46
5.6	Skizze des ersten Teiles . . . . .	47
5.7	Skizze des zweite Teiles . . . . .	47
5.8	Schematischer Aufbau eines Modellbauservos . . . . .	49
5.9	Pulsmuster zur Ansteuerung eines Modellbauservos . . . . .	50
5.10	Schaltschema der Pufferschaltung mit inkludiertem Festspannungsregler . . . . .	51
5.11	PCB der Pufferschaltung . . . . .	52
5.12	hergestellte und bestückte PCB der Pufferschaltung . . . . .	53
5.13	Schematischer Aufbau eines bipolaren Schrittmotors . . . . .	54
5.14	Treiberplatine zur Schrittmotoransteuerung von oben . . . . .	55
5.15	Treiberplatine zur Schrittmotoransteuerung von unten . . . . .	56
5.16	Bild des Phytex Entwicklungsboards . . . . .	56
5.17	Verbindungsschema der Messanordnung mit Modellbauservo . . . . .	58
5.18	Flussdiagramm der <i>main()</i> Routine . . . . .	60
5.19	Ablaufschema der Programmabarbeitung . . . . .	61
5.20	Verbindungsschema der Messanordnung mit einem Schrittmotor . . . . .	63
5.21	Flussdiagramm der <i>main()</i> Routine . . . . .	64
5.22	Ablaufschema der Programmabarbeitung . . . . .	65
5.23	Beschleunigung und Verzögerung eines Arbeitszyklus . . . . .	66
5.24	Hauptansicht der Anwendung <i>winroller</i> . . . . .	69
5.25	Detailansicht der Menüleiste . . . . .	70
5.26	Hauptansicht der Anwendung <i>winroller</i> . . . . .	71
5.27	Start der Messwertabfrage über das Menü . . . . .	71
5.28	Start der Messwertabfrage über die Symbolleiste . . . . .	71
5.29	Exemplarische Profilaufnahme . . . . .	72
5.30	Skizze der Objektausrichtung . . . . .	72
6.1	Schematisch-Abstrakter Aufbau dieser Messanlage . . . . .	77
6.2	Lasermodule mit Distanzhülse und Anschlusskabel . . . . .	79
6.3	Bemaßungsplan der Distanzhülse . . . . .	79
6.4	PCB der Philips WebCam . . . . .	81
6.5	Unterer Befestigungskörper . . . . .	82
6.6	Oberer Befestigungskörper . . . . .	83
6.7	Schaltschema des Entwicklungsboards . . . . .	84
6.8	Entflechtete PCB des Entwicklungsboards . . . . .	85
6.9	Bauteilansicht des gefertigten Entwicklungsboards . . . . .	86
6.10	Flussdiagramm der Main-Routine . . . . .	87
6.11	Flussdiagramm des Hostsystems . . . . .	90
6.12	Ablauf der Bildbearbeitung . . . . .	91

6.13	Darstellung des relevanten Bildausschnitts . . . . .	92
6.14	Bildschirm der gestarteten Anwendung . . . . .	93
6.15	Menüzeile der Applikation . . . . .	93
6.16	Dialogfenster zur Bildeinstellung . . . . .	94
6.17	Dialogfenster zur Formateinstellung . . . . .	95
6.18	Beispiel einer Profilaufnahme: Bild der Anordnung . . . . .	96
6.19	Beispiel einer Profilaufnahme: Modell einer Raumecke . . . . .	97
6.20	Mögliche Kennlinie der Anwendung . . . . .	98
7.1	Schemenhafter Aufbau der Messvorrichtung . . . . .	103
7.2	Öffnungswinkel $\alpha$ des Lasermoduls . . . . .	105
7.3	Geometrieübersicht der Laserprojektion . . . . .	106
7.4	Beispiel einer linearen Regression . . . . .	109
7.5	Diskretisierung der Messwerte . . . . .	110
7.6	Mögliche Szenarien bei der Bildung der Regressionsgeraden . .	110
7.7	Startschirm der Anwendung winfloor . . . . .	111
7.8	Voransicht der ersten Messung (Abstand Lasermodul und Kamera: 40mm) . . . . .	112
7.9	Auswertungsbild der ersten Messung Abstand Lasermodul und Kamera: 40mm) . . . . .	113
7.10	Voransicht der zweiten Messung Abstand Lasermodul und Kamera: 110mm) . . . . .	113
7.11	Auswertungsbild der zweiten Messung Abstand Lasermodul und Kamera: 110mm) . . . . .	114

# Tabellenverzeichnis

1.1	Häufigkeit der Steuerungsmethode [17] . . . . .	5
1.2	Bedarf an unterschiedlichen Systemen [17] . . . . .	5
4.1	Tabelle der wichtigsten Ausbreitungsgeschwindigkeiten . . . . .	33
4.2	Auflistung einiger Wertepaare Plättchendicke/Resonanzfrequenz	35
5.1	Auswahl kommerzieller Sensoren (Preise beziehen sich auf das damals aktuelle Datum) . . . . .	42
6.1	Elektrischen Anschlussdaten . . . . .	80
7.1	Verwendete Variablen . . . . .	102

# Literaturverzeichnis

- [1] A. Schmitt: Acuity Research Inc., Introduction to Optical Distance Measurement, <http://www.acuityresearch.com/index.html>, 24.02.2003
- [2] LASER DISTANCE MEASUREMENT, <http://cord.org/cm/leot/Module6/module6.htm>, 23.04.2001
- [3] LaserTriangulationsSensoren, [http://www.ntb.ch/Pubs/sensordemo/NTB\\_24.LTS.pdf](http://www.ntb.ch/Pubs/sensordemo/NTB_24.LTS.pdf), 21.03.2002
- [4] W. D. Lüke, Taschenbuch der Elektrotechnik und Elektronik, Fachbuchverlag Leipzig 1998
- [5] Einführung in die Sensorik, [prados@caae.com](mailto:prados@caae.com), 05.06.2003
- [6] Paul Goossens, Elektor - Elektronik & Computertechnik: Artikel Abstandsmessung mit Infrarot, Ausgabe April 2002
- [7] SHARP Ltd., Companies Homepage: Produkt Datasheets, <http://www.sharp.com>, 12.06.2002
- [8] Infineon Technologies, Companies Homepage, <http://www.infinition.com>, 20.11.2002
- [9] Phyttec GmbH., Companies Homepage, <http://www.phyttec.com>, 14.05.2003
- [10] Keil Software GmbH., Companies Homepage, <http://www.keil.com>, 16.07.2003
- [11] Royal Dutch Philips, Companies Homepage, <http://www.philips.com>, 10.02.2003
- [12] Organisation: Sensors Online, Homepage, <http://www.sensorsonline.org>, 11.12.2002

- [13] HaSoTec GmbH, Companies Homepage, <http://www.hasotec.de>, 14.03.2003
- [14] Framos Electronic Vertriebs GmbH, Companies Homepage, <http://www.framos.de>, 14.03.2003
- [15] Microchip Ltd., Companies Homepage, <http://www.microchip.com>, 10.01.2003
- [16] CadSoft Computer GmbH, Companies Homepage, <http://www.cadsoft.de>, 16.07.2003
- [17] W. Zagler, Wien August 2003
- [18] F. Seifert, M. Brandl, K. Dabrowski, C. Hausleitner; Will Radar Ever Reach the Signal Processing Quality of Bat Biosonar?; IEMW University of Technology Vienna