

Technische Realisierung multimodaler Sensorik für humanoide Roboter

Diplomarbeit

im Fachgebiet Informatik



vorgelegt von: Christian Benckendorff

Matrikelnummer: 196765

Betreuer: Dr. Manfred Hild

Erstgutachter: Prof. Dr. Hans-Dieter Burkhard

Zweitgutachterin: Prof. Dr.-Ing. Beate Meffert

Berlin, den 14. März 2010

Zusammenfassung

In der vorliegenden Arbeit wird die technische Realisierung eines sensorischen Systems für einen humanoiden Roboter beschrieben. Im Vordergrund stehen dabei die Reichhaltigkeit der Sensormodalitäten, die kostengünstige Umsetzung des Systems und der Nutzen für die Bewegungssteuerung des Roboters.

Die humanoide Roboterplattform und deren elektronische Architektur werden vorgestellt, um ein besseres Verständnis der Arbeit zu ermöglichen. Sowohl die Arbeiten zur Erstellung der Hardware für die Sensorik, als auch die Programmierung der zugehörigen Software werden ausführlich diskutiert. Es wird im Detail auf die Wahl der Bauteile, den Entwurf der elektronischen Schaltungen und auf die zeitkritischen Aspekte der Firmware eingegangen. Der elektronische und mechanische Aufbau der Sensoren ermöglicht eine hohe Genauigkeit der Messung, bei gleichzeitig geringem Gewicht. Die Firmware garantiert eine Aktualisierungsrate von 100 Hz für alle Messwerte und eine störungsfreie Übertragung von einheitlich normierten Daten über einen globalen Datenbus. Ferner wird eine einfache Kalibrierung der gesamten Sensorik gewährleistet. Begleitend wird die Entwicklung eines kapazitiven Winkelsensors besprochen, der den Vorteil eines Hohlachsenaufbaus besitzt und eine kostengünstige Alternative zu vergleichbaren Sensoren darstellt.

Isolierte Experimente haben eine hohe Genauigkeit der Sensoren gezeigt und lassen auf einen großen Nutzen zur Bewegungssteuerung des humanoiden Roboters schließen.

Inhaltsverzeichnis

Glossar	III
1. Einleitung	1
1.1. Ziel der Arbeit	1
1.2. Aufbau der Arbeit	2
2. M-Serie	4
2.1. Elektronische Architektur der Datenverarbeitung	4
2.2. <i>AccelBoard3D</i>	7
2.3. Globales Bussystem – <i>SpinalCord</i>	9
3. Sensoren	11
3.1. Kapazitiver Winkelsensor	11
3.1.1. Grundidee – Messprinzip	12
3.1.2. Elektronisches Design	16
3.1.3. Mechanisches Design – Gehäuse	22
3.1.4. Kostenaufstellung	24
3.2. Winkelsensor mittels Potentiometer	24
3.2.1. Grundidee	24
3.2.2. Elektronisches Design	26
3.2.3. Kostenaufstellung	28
3.3. Accelerometer	28
3.4. Kraftsensoren im Fuß	34
3.4.1. Elektronisches Design	34
3.4.2. Mechanisches Design	35
3.5. Interne Winkelsensoren der Dynamixel-RX-28-Servomotoren	36
3.6. Spannung und Strommessung	37
3.7. Weitere Sensorik	38
3.7.1. Temperatur im Akkumodul	38
3.7.2. Kapazitive Berührungssensoren	38
3.7.3. Kamera	39
4. Firmware	40
4.1. Allgemeine Beschreibung und Timing	40

Inhaltsverzeichnis

4.2. Einbindung der Sensoren	42
4.2.1. Dynamixel-RX-28-Servomotor via RS485	43
4.2.2. <i>AD7150</i> , <i>ADC121C027</i> via I ² C	46
4.2.3. Accelerometer <i>MMA7455L</i> via SPI	49
4.2.4. Interner ADC des <i>STM32-Mikrocontrollers</i>	52
4.3. Kalibrierung mittels <i>AB3D-Testkonsole</i>	54
5. Experimente und Resultate	56
5.1. Test des kapazitiven Winkelsensors	56
5.2. Test des Winkelsensors mittels Potentiometer	60
5.3. Test des Accelerometers	63
5.4. Test der Kraftsensoren	65
6. Fazit und Ausblick	67
Literaturverzeichnis	69
A. Anhang	i
Selbstständigkeitserklärung	iii
Einverständniserklärung	iv

Glossar

ABS	Acrylnitril-Butadien-Styrol – Kunststoff
ADC	Analog-to-Digital Converter – auch mit A/D-Wandler bezeichnet
Bypassing	Methode zur Reduzierung von Störungen in der Spannungsversorgung – oft mittels Beipass-Kondensator
CDC	Capacitance-to-Digital Converter
Decoupling	Methode zur Reduzierung von Störungen in der Spannungsversorgung – Schaltkreise werden voneinander entkoppelt
Dielektrikum	Stoff, der elektrisch nicht oder nur schwach leitend ist
ESR	Equivalent Series Resistance – der äquivalente Serienwiderstand
GPIO	General Purpose Input/Output - individuell konfigurierbare Anschlüsse eines Mikrocontrollers
HF	High Frequency
I ² C	Inter-Integrated Circuit – serieller Datenbus
ICs	Integrated Circuits
MOSFET	Metal Oxide Semiconductor Field-Effect Transistor
Permittivitätszahl .	Durchlässigkeit eines Material von elektrischen Feldern
Propriozeption	Eigenempfindung von Körperbewegungen und -lage
Residuum	Fehler zwischen einer vorhandenen Funktion und einer Näherungsfunktion
SPS	Samples per Second – Abtastwerte pro Sekunde
Stopplack	Dient zum Schutz und zur Isolierung – wird am Ende einer Platinenfertigung auf die Platine aufgetragen
USART	Universal Synchronous Asynchronous Receiver Transmitter – ermöglicht eine digitale, serielle Datenübertragung

1. Einleitung

1 Einleitung

Menschen bewegen sich tagtäglich ohne sich im Klaren zu sein, welche schwierige Aufgaben sie dabei mit Leichtigkeit bewältigen. Die Sinne des Menschen ermöglichen ihm die Ausführung von komplexen Bewegungen und das Wahrnehmen seiner Umwelt. Anhand des Tastsinns kann er zum Beispiel beurteilen, wie fest er einen Gegenstand in den Händen hält. Der Gleichgewichtssinn ermöglicht ihm stattdessen das Balancieren auf schmalen Objekten. Ferner ist der Mensch in der Lage propriozeptive Informationen seines Körpers zu empfinden, sodass ihm stets bewusst ist, in welcher Pose sich seine Gliedmaßen befinden, zumindest in einem gewissen Maße.

Für die Bewegungssteuerung eines humanoiden Roboters ist es unabdingbar, dass der Roboter ähnliches propriozeptives Wissen besitzt. Dazu gehört sowohl die Eigenempfindung von Bewegungen, als auch das Gefühl der Körperlage im Raum. Darüber hinaus ist es für viele Bewegungen wünschenswert, Informationen über die Einflüsse der Umwelt zu besitzen, die auf den Roboter einwirken. Roboter benötigen zum Erlangen jenes Wissens passende Sensoren, die die entsprechenden physikalischen Größen in für den Roboter verarbeitbare Daten umwandeln. Das Wort Sensor leitet sich aus dem lateinischen Wort „sentire“ ab, das zu deutsch „fühlen“ oder „empfinden“ heißt.

Die heutige Technik erlaubt den Bau einer Fülle von Sensoren, die diverse physikalische Größen messen können. Viele davon sind für die Bewegungssteuerung eines Roboters uninteressant, doch andere sind dafür umso wichtiger. Die Gelenkstellungen eines Roboters werden durch Winkelsensoren, auch Drehgeber genannt, gemessen. Sie wurden in fast allen humanoiden Robotern vergangener Projekte verbaut, vergleiche [ARA⁺06, KMR⁺08]. Beschleunigungssensoren oder Gyroskope verleihen dem Roboter einen Gleichgewichtssinn, auch diese Sensoren werden in aktuellen Projekten, wie dem Projekt LOLA [BLK⁺07], in geringer Stückzahl verbaut. Viele weitere, sinnvolle Sensoren sind denkbar.

1.1 Ziel der Arbeit

Ein autonomer, humanoider Roboter soll mit einer möglichst vielfältigen Sensorik ausgestattet werden, die die erwähnten Sensoren umfasst und zusätzlich weitere sinnvolle sensorische Daten zur Verfügung stellt. Der humanoide

1. Einleitung

Roboter wird im Kapitel 2 vorgestellt und mit *M-Serie* bezeichnet. Die reichhaltige Sensorik ermöglicht ihm genaue Informationen über seinen Körper und seine Umwelt zu erlangen, die ihm eine adäquate Bewegungssteuerung ermöglichen. Im Hinblick auf die Schwierigkeit der zu erfüllenden Aufgabe der Bewegungssteuerung eines humanoiden Roboters ist es von Vorteil, wenn diese Informationen redundant und multimodal sind. Dabei sollen die verwendeten Sensoren Wissen bereitstellen, das dem des biologischen Vorbildes, dem Menschen, möglichst ähnlich ist. Weiterhin soll das Gewicht des sensorischen Systems möglichst gering sein, um eine hohe Mobilität des Roboters gewährleisten zu können. In Anbetracht der angestrebten Vielzahl der Sensoren sind niedrige Anschaffungskosten der einzelnen Sensoren wünschenswert.

Es werden in der vorliegenden Arbeit mehrere Aspekte thematisiert, die zur Realisierung des sensorischen Systems notwendig waren. Darunter fällt insbesondere die Wahl der Sensoren und der Aufbau der dazugehörigen Elektronik zur Messung der wahrgenommenen Informationen. Die darüber hinausgehende für die elektronische Datenverarbeitung notwendige Rechenarchitektur der *M-Serie* wird beschrieben, um ein besseres Verständnis zu gewährleisten. Des Weiteren wird die Einbettung der von den Sensoren erhobenen Messwerte ins datenverarbeitende *Framework* des Roboters, unter zeitlichen Vorgaben, garantiert. Durch abschließende, isolierte Tests wird die Funktionstüchtigkeit des sensorischen Systems aufgezeigt und eine Abschätzung für den praktischen Nutzen zur Bewegungssteuerung gegeben.

1.2 Aufbau der Arbeit

Zu Beginn der Arbeit werden der humanoide Roboter und die elektronische Architektur der Datenverarbeitung vorgestellt, für die die multimodale Sensorik entwickelt wurde. Dabei werden die physischen Merkmale des humanoiden Roboters angegeben. Ferner wird das wichtigste elektronische Bauteil der *M-Serie*, das *AccelBoard3D*, beschrieben. Dieses ermöglicht eine verteilte Rechenarchitektur und ist für die weiterführende Verarbeitung der Sensorikdaten zuständig. Außerdem wird die Struktur des globalen Datenbusses erläutert.

Kapitel 3 handelt von den verwendeten Sensoren mit zugehöriger Elektronik. Es wird detailliert auf die erstellten elektronischen Schaltungen eingegangen und begleitend dazu werden isolierte Tests gezeigt, durch deren Ergebnisse die Leistung der Sensoren gesteigert werden konnte. Zusätzlich wird die Entwick-

1. Einleitung

lung eines kapazitiven Winkelsensors beschrieben. Hierbei wird das zugrunde liegende Messprinzip sowie der elektronische und mechanische Aufbau präsentiert.

Das anschließende Kapitel beschreibt die softwareseitige Integration der sensorischen Messwerte in das datenverarbeitende *Framework* des humanoiden Roboters. Dafür werden die verwendeten Schnittstellen vorgestellt und kurze Auszüge des Quelltextes diskutiert. Der zeitkritische Aspekt der Ausführung der Software auf einem Mikrocontroller wird über das ganze Kapitel hinweg betrachtet.

Am Ende der Arbeit werden Ergebnisse von Experimenten mit der fertigen Plattform gezeigt und eine Abschätzung der Genauigkeiten der Sensoren getätigt. Außerdem wird die Durchführbarkeit von Bewegungen, anhand der vorliegenden Sensorqualitäten, thematisiert.

2 *M-Serie*

Dieses Kapitel beschreibt den humanoiden Roboter *M-Serie* und dient hauptsächlich dem Verständnis der anderen Kapitel. Der Roboter wurde im Rahmen des Projektes *ALEAR* von der Arbeitsgruppe *Neurorobotics Research Laboratory (NRL)* entwickelt und gebaut, vergleiche [ALE, NRL10]. Er hat eine Körpergröße von 125 cm und wiegt ca. 12,5 kg. Die *M-Serie* kann 30 Freiheitsgrade aktiv bewegen und hat zwei passive Zehengelenke. Ein gerendertes Bild ohne Außenverschalung des Roboters ist in der Abbildung 2.1 zu sehen. Die Motorik des Roboters wird durch RX-28-Dynamixel-Servomotoren ermöglicht. Einige Freiheitsgrade erfordern ein sehr hohes Drehmoment, deshalb werden bis zu vier Motoren pro Freiheitsgrad eingesetzt. Des Weiteren werden einige Aktuatoren mit Torsionsfedern an ihren Abtrieben versehen. Dadurch werden die Freiheitsgrade in gewissem Maße nachgiebig und dem biologischen Muskel ähnlicher. Damit der Roboter trotzdem ein ausreichendes Wissen über seine Körperstellung für eine adäquate Bewegungssteuerung erlangen kann, muss ein Winkelaufnehmer am Gelenk des Freiheitsgrades platziert werden. In dem Projekt BioRob werden die Winkelsensoren ähnlich platziert, vergleiche [KLvS⁺08, S. 5f].

Folgend wird im ersten Abschnitt die allgemeine Architektur der Datenverarbeitung erläutert. Daran schließt sich die Erläuterung des *AccelBoard3D* – die für die Bewegungssteuerung wichtigste Rechen- und Sensorikplatine – an. Abschließend wird das globale Bussystem im Detail beschrieben, dieses wird beim M-Serie-Roboter *SpinalCord* genannt.

2.1 Elektronische Architektur der Datenverarbeitung

Dieser Abschnitt widmet sich der Struktur der elektronischen Datenverarbeitung des M-Serie-Roboters. Der Roboter besitzt einen globalen Datenbus, über den verschiedene elektronische Komponenten miteinander kommunizieren. Zu diesen Komponenten gehören die *AccelBoard3D*, die *EnergyModules* und das *BrainModule*. Jedes dieser Bauteile erfüllt eine bestimmte für den Roboter wichtige Aufgabe. Einen vereinfachten Überblick der wichtigsten elektronischen Bestandteile der *M-Serie* ist in Abbildung 2.2 dargestellt. Die Grafik zeigt weiterhin die Verbindungen der einzelnen Teile, die sich an die physische Verkabelung anlehnen, welche aber weitaus komplexer ist. Der *SpinalCord* er-

2. M-Serie



Abbildung 2.1.: Rendering des M-Serie-Roboters.

2. *M-Serie*

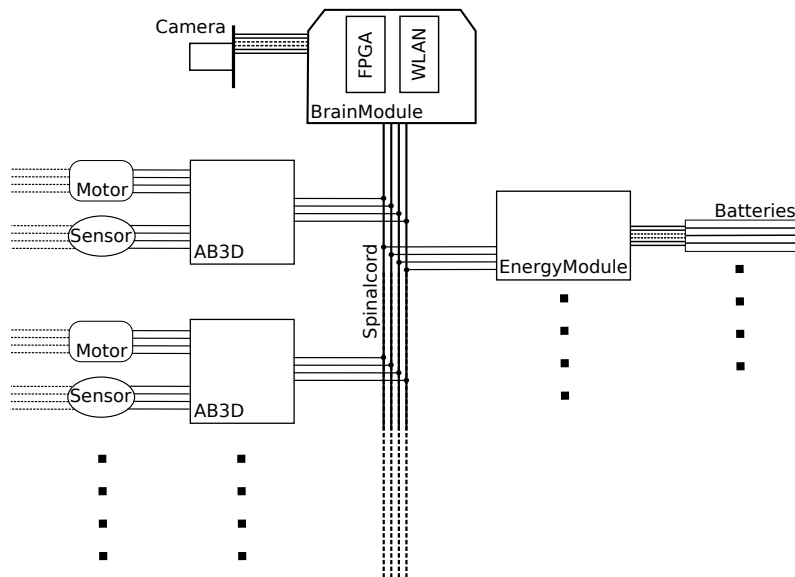


Abbildung 2.2.: Elektronische Architektur der Datenverarbeitung des M-Serie-Roboters.

streckt sich über alle Körperteile des Roboters, und zwar über den Kopf, den Torso, die Arme und die Beine. Die *M-Serie* ist modular aufgebaut, das heißt, dass alle Körperteile leicht von einander getrennt werden können. Dies kann sogar während des Betriebs geschehen. Alle Körperteile können nach einer Trennung eigenständig funktionieren, da jedes eine eigene Stromversorgung und Steuerungselektronik besitzt.

Das *BrainModule* besitzt ein WLAN- und ein FPGA-Modul, an das eine Kamera angeschlossen ist. Die Aufgaben sind somit im Besonderen die Bildverarbeitung und die kabellose Kommunikation mit einem Personal Computer. Jedes Körperteil hat genau ein *EnergyModule*, das für die Verwaltung von vier Lithium-Polymer-Zellen zuständig ist. Die wichtigste Komponente des Roboters, besonders im Hinblick auf die Sensorik und Bewegungssteuerung, ist das *AccelBoard3D*, auch *AB3D* genannt. An diesem Board sind alle Aktuatoren und die meisten Sensoren des Roboters angeschlossen. Das *AccelBoard3D* wird im Abschnitt 2.2 detaillierter beschrieben. Insgesamt sind 21 *AccelBoard3D* im Roboter verbaut, jedem ist eine eindeutige Identifikationsnummer, kurz ID, zugeordnet. Zusätzlich befindet sich ein *QuadCore*-Modul im Torso, das aus vier Recheneinheiten des *AccelBoard3D* besteht und auch als vier Teilneh-

2. M-Serie

mer – mit vier unterschiedlichen IDs – am *SpinalCord* teilnimmt. Allerdings muss es andere, für diese Arbeit unrelevante Aufgaben erledigen.

Die Grundfrequenz des Roboters beträgt 100 Hz, damit ist gemeint, dass jeder Teilnehmer am globalen Datenbus seine Daten genau in 10 ms-Abständen aktualisiert versendet. Dadurch stehen jedem Teilnehmer die Sensordaten aller Anderen in diesem Zeitabstand zur Verfügung.

2.2 *AccelBoard3D*

Das *AccelBoard3D* ist ein elektronisches Bauteil, das für die Erfassung multimodaler Sensorwerte, die Ansteuerung von Servomotoren und die Berechnung von neuronalen Netzen zuständig ist. Darüber hinaus kann es Befehle von einem Personal Computer entgegennehmen. Die Datenverarbeitung wird von einem Mikrocontroller übernommen. Der Mikrocontroller hat die Typenbezeichnung STM32F103CBT6. Er besitzt einen ARM CortexTM-M3 Kern und taktet mit maximal 72 MHz. Weitere Einzelheiten zum Mikrocontroller und seinen Schnittstellen können dem Datenblatt [STM09] entnommen werden.

Ein Foto des *AccelBoard3D* ist in der Abbildung 2.3 (links) ersichtlich. Neben dem *STM32-Mikrocontroller* sind noch zahlreiche andere ICs auf der Platine des *AccelBoard3D* vorhanden, die verschiedenste Aufgaben besitzen. Ein Großteil der Bauteile sorgt für die korrekte, stabilisierte und entstörte Spannungsversorgung der ICs und der externen Schnittstellen.

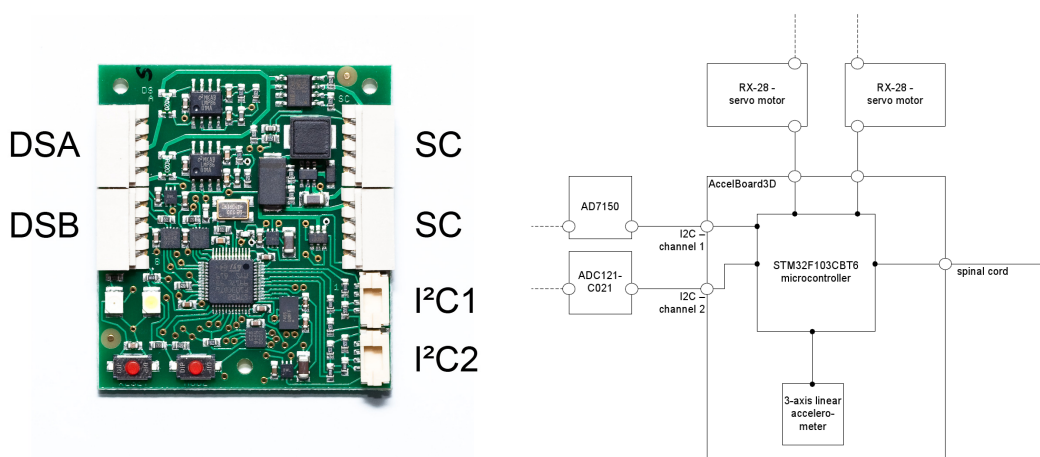


Abbildung 2.3.: Foto der Oberseite eines *AccelBoard3D* mit Beschriftung der Schnittstellen (links). Schematische Darstellung des *AccelBoard3D* mit den vorhandenen Schnittstellen (rechts).

2. *M-Serie*

Der Name des *AccelBoard3D* leitet sich von einem 3-Achsen-Accelerometer ab, welches auch auf der Platine des *AccelBoard3D* untergebracht ist. Weitere wichtige integrierte Schaltkreise sind die *Transceiver*, diese ermöglichen dem *STM32-Mikrocontroller* die Kommunikation über eine RS485-Schnittstelle. RS485 ist ein Schnittstellenstandard, bei dem die Daten differentiell und seriell übertragen werden, dadurch ist die Datenübertragung störungssicherer, in Bezug auf elektromagnetische Einstrahlungen. Auf dem *AccelBoard3D* sind insgesamt drei *Transceiver* verbaut, einer ist für die Kommunikation über den globalen Bus zuständig, im Bild 2.3 (links) sind die dazugehörigen Anschlüsse mit „SC“ gekennzeichnet. Die beiden anderen *Transceiver* ermöglichen die Anbindung von mehreren RX-28-Dynamixel-Servomotoren. Es stehen zwei separate Anschlüsse zur Verfügung, die *DoubleStrand A und B* genannt werden. Im Bild sind sie mit „DSA“ und „DSB“ beschriftet. Ein einzelner *DoubleStrand* wäre aus zeitlichen Gründen nicht ausreichend. Die Dynamixel-Servomotoren haben eine nicht deterministische Antwortzeit, sodass die Kommunikation im vorgegebenen 10 ms-Intervall, bei serieller Abfrage aller Motoren, zu lange dauern würde. Durch die Aufteilung der Anschlüsse ist eine parallele Abfrage der Motoren möglich. In der Firmware des *AccelBoard3D* ist die Ansteuerung von bis zu vier RX-28-Motoren vorgesehen, jeweils zwei pro *DoubleStrand*. Ferner wurde auf dem *AccelBoard3D* eine Spannungs- und Strommessung für jeden der *DoubleStrands* realisiert.

Darüber hinaus sind zwei Steckbuchsen für externe Sensorik vorhanden. Die externen Sensoren werden über einen I²C-Datenbus an den Mikrocontroller angebunden. Im Foto sind diese mit I²C1 und I²C2 bezeichnet.

Für die Benutzerinteraktion sind zwei Taster und zwei LEDs vorhanden. Mit dem einen Taster kann die Spannungsversorgung des Mikrocontroller unterbrochen werden und erlaubt daher einen Neustart. Der andere Taster wird zum einen für die Programmierung der Firmware in den Speicher des Mikrocontrollers benötigt und kann zum anderen während der Laufzeit mit beliebigen Funktionen belegt werden. Die beiden LEDs können durch die Firmware beliebig mit digitalen Signalen belegt werden.

Die Bewegungssteuerung des Roboters kann vollkommen autonom von den *AccelBoard3D* realisiert werden. Bei der *M-Serie* werden zur Ansteuerung der Aktuatoren neuronale Strukturen eingesetzt. Eine ausführliche Beschreibung neuronaler Module zur Bewegungssteuerung wird in [Hil07] getätigt. Die Da-

2. *M-Serie*

ten der neuronalen Netze werden von einem Personal Computer auf die *AccelBoard3D* übertragen. Zur Laufzeit des Roboters wird von allen *AccelBoard3D* ihr entsprechendes neuronales Netz berechnet. Jedem *AccelBoard3D* stehen dazu alle für die Bewegungssteuerung relevanten Sensorwerte des gesamten Roboters zur Verfügung, welche über den globalen Datenbus übertragen werden.

2.3 Globales Bussystem – *SpinalCord*

Der globale Datenbus des humanoiden Roboters ist, verglichen mit einem Wirbeltier, dem Rückenmark sehr ähnlich. Aus diesem Aspekt leitet sich sein Name im M-Serie-Roboter ab, dem englischen Wort für Rückenmark – *SpinalCord*.

Wie schon mehrfach erwähnt, werden alle Messwerte der Sensoren über den *SpinalCord* übertragen. Zusätzlich können Aktivitäten von Neuronen der neuronalen Strukturen, die für die Bewegungssteuerung verantwortlich sind oder andere Daten versendet werden. Alle versendeten Nutzdaten sind zwischen -1 und $+1$ normiert. Die im M-Serie-Roboter benutzten neuronalen Strukturen arbeiten im selben Wertebereich, sodass Berechnungen für die Bewegungssteuerung ohne weitere Normierungen vorgenommen werden können.

Die Werte sind in der Regel 16 Bit groß und in einem Festkommaformat dargestellt. Jeder Teilnehmer am *SpinalCord* hat ein bestimmtes Zeitfenster, in dem er seine Daten über den globalen Datenbus versenden darf. Das *BrainModule* und die *AccelBoard3D* versenden jeweils exakt 27 *Words*. Die *EnergyModules* versenden dagegen nur 3 *Words*. Folglich werden von den insgesamt 32 Teilnehmern Daten mit einer Größe von 1440 Bytes alle 10 ms über den *SpinalCord* versendet. Eine Übersicht der Bedeutung der versendeten Daten ist in Abbildung 2.4 zu finden. Die ersten beiden *Words* sind Metadaten, die unter anderem für die Synchronisation und die Identifikation zuständig sind. Im *Peephole* können Daten mit einer geringeren Aktualisierungsrate versendet werden. Alle lilafarben hinterlegten Datenfelder sind nicht fest zugeordnet und können daher zur Übertragung von individuellen Daten benutzt werden. Der grüne Abschnitt am Ende des Datenfeldes der *AccelBoard3D* gibt Auskunft über die aktuell erzeugten Drehmomente, aller an einem *AccelBoard3D* angeschlossenen Servomotoren. Die Messwerte der Sensorik des Roboters werden an die mit gelb gekennzeichneten Positionen im *SpinalCord* geschrieben. Sowohl

TECHNISCHE REALISIERUNG MULTIMODALER SENSORIK FÜR HUMANOIDE ROBOTER

2. M-Serie

	0 (BrainModule)	1 ... 6 (EnergyModule)	7 ... 31 (AccelBoard3D)
#00	SYNC (0x5555)	SYNC (0x5555)	SYNC (0x5555)
#01	ID(5) + Health(3) + Mode (8)	ID(5) + Health(3) + Mode (8)	ID(5) + Health(3) + Mode (8)
#02	Peephole	Peephole	Peephole
#03			Accel.X
#04			Accel.Y
#05			Accel.Z
#06			Current DS A
#07			Current DS B
#08			DS A, 1 Position
#09			DS A, 2 Position
#10			DS B, 1 Position
#11			DS B, 2 Position
#12			I ² C 1
#13			I ² C 2
#14			I ² C 3
#15			I ² C 4
#16			I ² C 5
#17			I ² C 6
#18			
#19			
#20			
#21			
#22			
#23			DS A, 1 Torque / Servo 1 Position
#24			DS A, 2 Torque / Servo 2 Position
#25			DS B, 1 Torque / Servo 3 Position
#26			DS B, 2 Torque / Servo 4 Position

Abbildung 2.4.: Struktur der über den *SpinalCord* versendeten Daten.
Autor: Christian Thiele.

die Angaben über die Drehmomente, als auch die sensorischen Informationen sind für jedes *AccelBoard3D* individuell. Das heißt, dass für ein bestimmtes *AccelBoard3D* der I²C-Datenwert einen Winkel eines Gelenkes darstellen könnte, aber bei einem anderen *AccelBoard3D* die gemessene Kraft eines Kraftsensors angibt. Deshalb existieren für alle *AccelBoard3D* separate Angaben über die Bedeutung der jeweiligen Datenwerte im *SpinalCord*.

Die sensorischen Werte im *SpinalCord* sind abhängig von physikalischen Größen. Die vorliegende Arbeit soll erläutern, wie diese physikalischen Größen gemessen und in einem geeigneten Datenformat in den *SpinalCord* gelangen.

3. Sensoren

3 Sensoren

Dieses Kapitel erläutert die verschiedenen Sensoren des M-Serie-Roboters. Dabei wird auf die Wahl der passenden Bauteile eingegangen und besonderen Wert auf die Umsetzung der elektronischen Schaltungen des jeweiligen Sensors gelegt. Alle softwareseitigen Arbeiten zur Erhebung der Messwerte sind dagegen im Kapitel 4 beschrieben.

3.1 Kapazitiver Winkelsensor

In diesem Abschnitt wird die Entwicklung eines kapazitiven Winkelsensors detailliert erklärt. Die Entwicklung wurde aus mehreren Gründen motiviert. Es ist von Vorteil, den Winkelsensor konzentrisch zur Achse zu platzieren. Dadurch kann auf eine zusätzliche Mechanik zur Umlenkung verzichtet werden. Jedoch wird durch diese Art der Anordnung das Gelenk wesentlich breiter. Die Benutzung von Hohlachsenschendrehgebern kann diesen Effekt minimieren. Auf Grund ihrer geringen Tiefe würde ihr Einbau nur ein geringfügig breiter konstruiertes Gelenk erfordern. Die genannten Drehgeber haben jedoch den Nachteil, dass sie sehr teuer oder nur bei einer Abnahme einer sehr großen Stückzahl zu beziehen sind, siehe Tabelle 3.1.

Tabelle 3.1.: Preise von ausgewählten Hohlachsenschendrehgebern. Stand: 11.11.2009

Hersteller	Bezeichnung	Preis	Anbieter
Hengstler	RI58-D/1000ED.32IB	353,86 €	Farnell GmbH
Baumer	BHK 16.05A.0500-B6-5	223,00 €	RS Components GmbH
HB5M	HB5M-32-197-I-S-D	173,84 \$	US Digital
BEI	HS35-XHS35F	605,00 \$	BEI Industrial Encoders
Koyo	TRDA-VA1000V6-29-3PR	130,00 \$	Koyo Encoder

Bezogen auf einen humanoiden Roboter würden die Kosten der Herstellung erheblich steigen, da er mit vielen Freiheitsgraden ausgestattet ist und ferner der Anspruch besteht, dass jeder davon mit einem Winkelsensor ausgestattet wird. Die beschriebenen Nachteile führten somit zu der Idee einer Eigenentwicklung eines Hohlachsen-Winkelsensors, der konzentrisch zur Achse angebracht werden kann und dabei auch kostengünstig herzustellen ist.

3. Sensoren

3.1.1 Grundidee – Messprinzip

Die Grundidee hinter dem kapazitiven Winkelsensor ist, dass zwei elektrisch leitende Flächen einen Plattenkondensator bilden, der die Kapazität ändert, sobald sich die Höhe der Überlagerung beider Flächen ändert. Durch die richtige Anordnung und Form der Flächen kann auf den Winkel geschlossen werden, der bei der Verdrehung beider Flächen zueinander entsteht. Die Flächen wurden in entsprechender Form direkt auf zwei Platinen platziert, damit die Bauhöhe des gesamten Sensors möglichst gering bleibt. In [CB01, FGMT04] werden kapazitive Winkelsensoren beschrieben, jedoch wird dort ein anderer Aufbau des Sensors bevorzugt. Die elektrisch leitenden Flächen des fertigen Winkelsensors

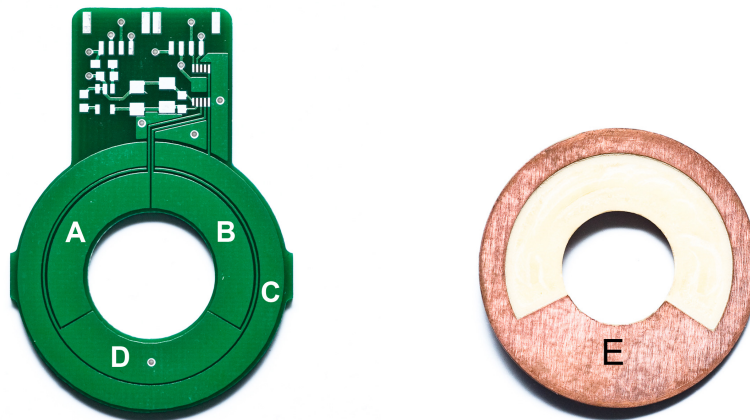


Abbildung 3.1.: Haupt- und Gegenplatine des kapazitiven Winkelsensors mit Beschriftung der wichtigsten Leiterbahnflächen. Mit den beiden Messflächen A und B, der Fläche zum Gegenstück C, der Abschirmungsfläche D und der Gegenstückfläche E.

sind in der Abbildung 3.1 dargestellt. Links ist die Hauptplatine zu sehen, im oberen Teil befindet sich gleichzeitig die Schaltung der Auswertungselektronik. Rechts daneben ist die Platine des Gegenstücks. Um den mechanischen Aufbau des Sensors möglichst einfach zu halten, wird auf eine direkte Verbindung beider Flächen, per Kabel oder Ähnlichem, verzichtet. Stattdessen wird neben dem Plattenkondensator, der seine Kapazität mit der Verdrehung der Flächen ändert, ein weiterer konstanter Plattenkondensator geschaffen. Dieser ist dann in Reihe mit dem anderen geschaltet. In der Abbildung 3.1 wird er zwischen der gesamten Fläche C und dem äußeren Ring der Fläche E gebildet. Um einen größeren Teil des Drehbereichs messen zu können, werden zwei un-

3. Sensoren

abhängige Plattenkondensatoren zur Messung des Drehwinkels genutzt. Diese werden jeweils zwischen den Flächen A und B, im Bild links, und dem inneren Teil der Fläche E, die nicht zum äußeren Ring gehört, gebildet. Sie sind folglich um 120° zueinander verdreht. Je nach Überschneidung der Flächen wird eine andere Kapazität der Kondensatoren erwartet. Die Fläche D dient ausschließlich der Abschirmung des statischen Kondensators von den zwei variierenden. Abbildung 3.2 zeigt die zu erwartenden Kapazitäten der variierenden Kondensatoren

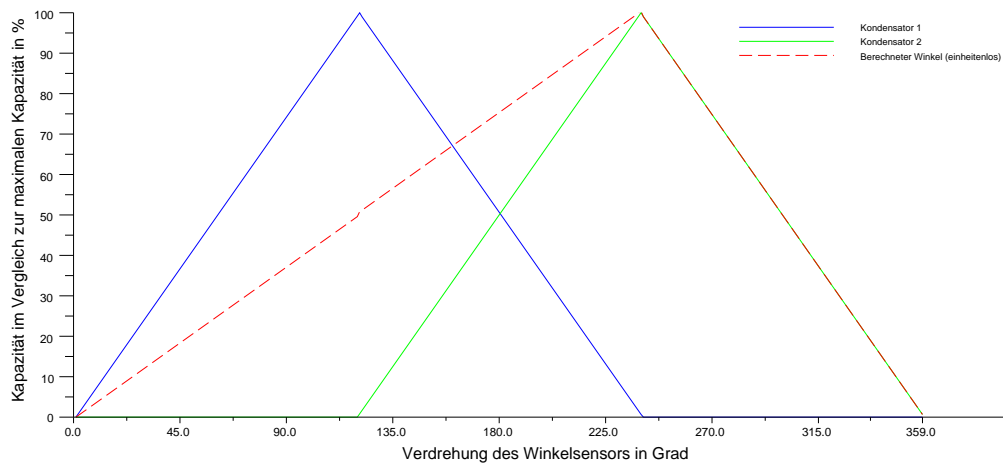


Abbildung 3.2.: Theoretische Kapazitäten der beiden variierenden Plattenkondensatoren in Abhängigkeit des Drehwinkels. Berechneter Winkel – einheitenlos und gestrichelt.

satoren in Abhängigkeit des Drehwinkels. Die erste Kapazität ist in blau und die zweite in grün eingezeichnet. Die Kurvenverläufe stellen dabei das Idealbild der Kapazitäten dar. Durch eine Vielzahl verschiedener Faktoren kann das spätere Ergebnis dementsprechend davon abweichen. Die rot gestrichelt eingezeichnete Kurve wurde aus den beiden anderen berechnet. Die dafür benutzte Formel lautet:

$$\alpha = \frac{C_1 + (2C_2)}{2},$$

in der C_1 die Kapazität des ersten Kondensator ist (blaue Kurve), C_2 die des zweiten (grüne Kurve) und α der errechnete Winkel ist. Werden die beiden einzelnen Kapazitäten ins Verhältnis gesetzt, erhalten wir für einen Winkelbereich von 0 bis 239° eine gültige Abbildung für den tatsächlichen Verdrehungswinkel des Sensors. Des Weiteren wird in der Abbildung 3.2 nur das

3. Sensoren

Verhältnis zwischen der maximalen Kapazität und den entsprechenden Kapazitäten des aktuellen Drehwinkels dargeboten. Die Abbildung gibt folglich noch keine Auskunft über die tatsächliche Größe der Kapazitäten der beiden Plattenkondensatoren. Daher soll nun die zu erwartende maximale Kapazität der Kondensatoren berechnet werden. Dabei bilden die Abmaße und der Aufbau des finalen kapazitiven Winkelsensor die Grundlage für die Berechnung. Die verwendeten Abmaße sind:

$$\begin{aligned}D_1 &= 17,00 \text{ mm} \\D_2 &= 26,20 \text{ mm} \\D_3 &= 27,80 \text{ mm} \\D_4 &= 34,00 \text{ mm} \\d_1 &= 0,18 \text{ mm} \\d_2 &= 0,01 \text{ mm}.\end{aligned}$$

Der innere Durchmesser der Fläche A und B aus Abbildung 3.1 ist dabei D_1 und der äußere D_2 . Dementsprechend ist D_3 und D_4 der innere und äußere Durchmesser der Fläche C. In Abschnitt 3.1.3 wird beschrieben, dass zwischen der Haupt- und Gegenplatine eine Teflonfolie platziert wird. Die Dicke dieser Folie ist d_1 . Der Wert d_2 steht für die Höhe des Stopplacks. Dieser ist auf der Platine aufgetragen und muss daher bei der Bestimmung des Abstands der Flächen beachtet werden. Zuerst müssen die Flächen der Kreissegmente berechnet werden, dies geschieht mit folgender Formel für einen Kreisring:

$$A = \pi(r_2^2 - r_1^2).$$

A ist die resultierende Fläche, r_2 der größere und r_1 der kleinere Radius. Der Flächeninhalt A_C der Fläche C aus Abbildung 3.1 berechnet sich daher wie folgt:

$$A_C = \pi\left(\left(\frac{1}{2}D_4\right)^2 - \left(\frac{1}{2}D_3\right)^2\right).$$

3. Sensoren

Mit den gegebenen Abmaßen des finalen Sensors ergibt sich für A_C eine Fläche von $300,93 \text{ mm}^2$. Der Flächeninhalt von A und B berechnet sich ähnlich, er beträgt nur ein Drittel der Fläche des zugehörigen Kreissegments:

$$A_{A/B} = \frac{\pi((\frac{1}{2}D_2)^2 - (\frac{1}{2}D_1)^2)}{3}.$$

Daher ist $A_{A/B}$ $104,05 \text{ mm}^2$ groß. Die Kapazität eines Plattenkondensators kann mit folgender Formel bestimmt werden:

$$C = \epsilon_0 \epsilon_r \frac{A}{d},$$

wobei C die Kapazität des Plattenkondensators, ϵ_0 die elektrische Feldkonstante, ϵ_r die Permittivitätszahl des Dielektrikums, A die Fläche und d der Abstand ist. Für ϵ_0 gilt:

$$\epsilon_0 = 8,854188 \cdot 10^{-12} \text{ A s/Vm}.$$

Die Permittivitätszahl ϵ_r kann nur angenähert werden, da die genaue Zusammensetzung des Stopplacks der Platinen nicht vorliegt. Daher wird nur die Permittivität von Teflon verwendet, sie beträgt:

$$\epsilon_r = 2,0.$$

Nun können die beiden unterschiedlichen maximalen Kapazitäten berechnet werden. Sie bestehen aus der Kapazität des statischen und den beiden identischen Kapazitäten der variierenden Plattenkondensatoren. Für C_C ergibt sich folgende Formel:

$$C_C = \epsilon_0 \epsilon_r \frac{A_C}{d_1 + d_2}.$$

Nach dem Einsetzen aller bereits bekannten Werte und Ausrechnen der Formel erhalten wir für C_C eine Kapazität von $28,048 \text{ pF}$. Analog lässt sich $C_{A/B}$ berechnen, wir erhalten $C_{A/B} = 9,698 \text{ pF}$. Da die beiden Kondensatoren mit den Kapazitäten C_C und $C_{A/B}$ in Reihe geschaltet sind, muss folgende Formel zur Berechnung der endgültigen Kapazität benutzt werden:

$$\frac{1}{C} = \frac{1}{C_1} + \frac{1}{C_2} + \dots + \frac{1}{C_n}.$$

3. Sensoren

In unserem Fall bedeutet dies, dass sich die maximale Gesamtkapazität durch Umstellung und Einsetzen der vorher ausgerechneten Einzelkapazitäten in die nachstehende Formel berechnet:

$$C = \frac{1}{\frac{1}{C_C} + \frac{1}{C_{A/B}}}.$$

Für die Gesamtkapazität C erhalten wir folgendes Ergebnis $C = 7,206$ pF. Wir haben die Berechnung mit den finalen Abmaßen des kapazitiven Winkelsensors zum besseren Verständnis durchgeführt. Bei der Entwicklung des Sensors wurde sie jedoch schon frühzeitig mit unterschiedlichen Maßen getätigt. Dies war nötig, damit die zu erwartende Kapazität den Ansprüchen der Auswertungs-elektronik genügt. Die Entwicklung der passenden Elektronik, zur Messung der Kapazitäten der beiden variierenden Kondensatoren, wird im nächsten Abschnitt 3.1.2 behandelt.

3.1.2 Elektronisches Design

Beim elektronischen Design mussten viele Bedingungen bedacht werden. Um Platz im mechanischen Aufbau des Roboters zu sparen und den Aufbau einfach zu halten, ist es sinnvoll die Schaltung direkt auf eine der schon vorhandenen Platinen der Sensorflächen zu bringen. Außerdem werden dadurch Verbindungskabel vermieden, die die Kapazitätsmessung verfälschen könnten. Eine Benutzung von einem integrierten Schaltkreis zur Kapazitätsmessung kann den Aufbau weiter vereinfachen. Durch die in den vergangenen Jahren immer häufigere Verwendung von kapazitiven Schaltflächen, wie sie in [AD08] beschrieben werden, werden vermehrt CDC-ICs entwickelt. Ein Vergleich von vier unterschiedlichen CDC-ICs ist in der Tabelle 3.2 abgebildet.

3. Sensoren

Tabelle 3.2.: Vergleich von ausgewählten ICs zur Messung von Kapazitäten.

IC	<i>UTI</i>	<i>Picostrain</i>	<i>AD7150</i>	<i>AD7147</i>
max. Auflösung	14 Bit	20 Bit	12 Bit	16 Bit
max. Frequenz	10 - 100 Hz	50 kHz	100 Hz	100 Hz
max. Anzahl Kap.	3	4	2	13
Kap. Bereich [pF]	0-2/0-12/300	0-100000	0-13	0-8
Interface	PWM	SPI	I ² C	I ² C
Bauform	DIP/SOIC	TQFP/QFN	MSOP	LFCSP
Anzahl Pins	16/18	48	10	24
Baugröße [mm²]	120	81/49	15	16
Spannungsbereich	2,9 - 5,5 V	1,8 - 3,6 V	2,7 - 3,6 V	2,6 - 3,6 V
Stromaufnahme	~14 mW	~0,54 mW	~0,36 mW	~5,4 mW
Preis in Euro	~11,43/10,26	~12	~3,31	~4,14

Das erste Kriterium, nach denen die Chips verglichen werden sollten, ist die maximale Auflösung. Der schlechteste Wert dieser vier ICs ist 12 Bit. Bei einem Winkelsensoraufbau, der 360° abdecken würde, hätten wir damit eine Auflösung von unter 0,1°. Für unseren Zweck, der Verwendung der Daten zur Bewegungssteuerung, also ausreichend. Da folglich alle in der Tabelle 3.2 aufgelisteten Chips eine ausreichende Auflösung liefern, sollte als nächstes die maximale *Updatefrequenz* betrachtet werden. Sie gibt an mit welcher Frequenz ein neu gemessener Wert zur Verfügung steht. Drei der CDCs haben eine maximale Frequenz von 100 Hz, nur der *Picostrain* kommt auf einen weit höheren Wert von 50 kHz. Die Grundfrequenz des M-Serie-Roboters, mit der alle Sensorwerte abgefragt und über den globalen Datenbus übertragen werden, beträgt auch 100 Hz, vergleiche Kapitel 2. Daher bieten die 50 kHz des *Picostrain*-Chips nur den Vorteil, dass über viele Werte gemittelt werden und dadurch das Sensorrauschen vermindert werden könnte. Somit dient auch die *Updatefrequenz* nicht als Ausschlusskriterium. Jedoch unterscheidet sich die Fläche, die der IC auf der Platine einnehmen würde und der Preis von den ersten beiden aufgeführten Chips gegenüber den anderen sehr deutlich, siehe Tabelle 3.2. Um die Kompaktheit und niedrige Kosten des Winkelsensors zu garantieren, sollte also eine Entscheidung zwischen den beiden letzten ICs (*AD7150* und *AD7147*) getroffen werden. Die Stromaufnahme ist wegen des Einsatzes in einem mobilen System sehr wichtig, besonders wenn man in Betracht zieht, dass der

3. Sensoren

Sensor gleich vielfach verbaut werden soll. Die Stromaufnahme stellt daher das entscheidende Kriterium dar. Der *AD7150* hat gegenüber dem *AD7147* eine 15fach niedrige Stromaufnahme, somit wurde der *AD7150* als CDC des kapazitiven Winkelsensors ausgewählt.

Anschließend sollte sicher gestellt werden, dass der *AD7150* den Ansprüchen genügt und auch in dem zu verbauendem Gesamtsystem einwandfrei funktioniert. Daher wurde der Einfluss der Güte der Versorgungsspannung auf die Genauigkeit der Messergebnisse getestet. Laut Datenblatt [AD07, S. 24] könnte der Chip sensibel auf ungewollte Einstreuungen in der Versorgungsspannung reagieren. Es wird daher eine Entkopplung und Filterung empfohlen. In einem humanoiden Roboter gibt es zahlreiche, unterschiedliche Komponenten und Vorgänge, die sich auf die Güte der Stromversorgung auswirken können. Einerseits sind es die vielen Motoren, die je nach Belastung viel Strom verbrauchen oder sogar als Generator fungieren können. Andererseits sind es die hochfrequenten Einstrahlungen der gesamten Datenkommunikation, die störende Schwankungen der Spannung verursachen.

Der Einfluss der Versorgungsspannung wurde folgendermaßen getestet: Es wurde dieselbe konstante Kapazität zweimal gemessen, jeweils ca. drei Sekunden lang bei einer Abtastrate von 100 Hz. Dabei wurden jeweils unterschiedliche Stromquellen für den *AD7150*-IC benutzt. Als kapazitive Messspanne wurden 2 pF gewählt, vergleiche dazu [AD07, S. 18]. Die Ergebnisse dieser Messungen sind als Histogramme, zur besseren Vergleichbarkeit, in der Abbildung 3.3 dargestellt. Rot abgetragen sind die Messwerte, bei denen der *AD7150*-IC über ein 12 V-Netzteil betrieben wurde. Das Netzteil war an einer separaten Platine, dem *AccelBoard3D*, angeschlossen. Auf der Platine wurde die Spannung von 12 V mittels Schalt- und Linearregler auf 3,3 V runter geregelt. Auf dem Board lief ein Programm auf einem *STM32-Mikrocontroller* ab, das die Daten vom *AD7150* per I²C abgefragt und per serieller Schnittstelle an einen Personal Computer weitergeleitet hat, vergleiche [STM09]. Die blau dargestellten Daten wurden mit dem selben Aufbau erfasst, allerdings wurde die Stromversorgung des *AD7150* durch eine 9 V Blockbatterie ersetzt. Die 3,3 V wurden durch einen dazwischen geschalteten Linearregler, einem *LP2950* [Nat09b], erzeugt.

Schon beim optischen Vergleich der beiden Histogramme wird deutlich, dass eine störungsfreiere Stromversorgung des *AD7150* die Qualität der gemessenen

3. Sensoren

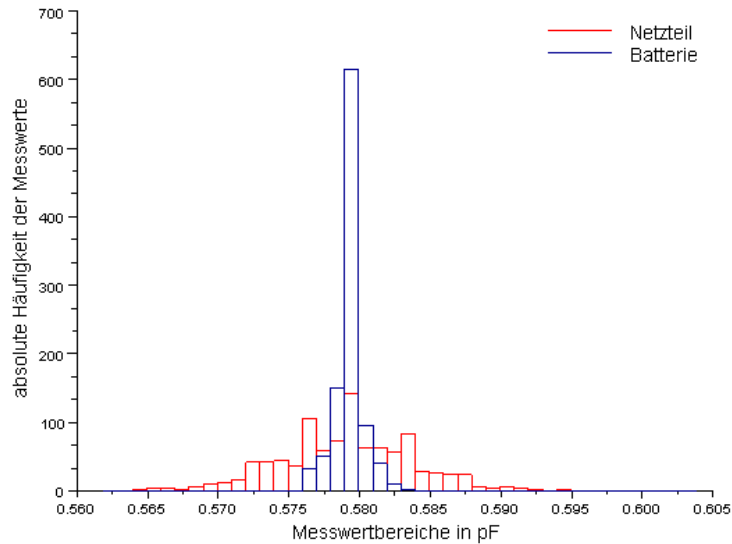


Abbildung 3.3.: Vergleich der ermittelten Messdaten des *AD7150* mit unterschiedlichen Spannungsquellen. Als Darstellung wurden zwei übereinander gelagerte Histogramme verwendet.

Daten verbessert. Die Standardabweichung der ca. 3000 Messwerte beträgt mit Netzteil 4,7671 fF und mit Batterieversorgung 1,0907 fF. Würde die Stromversorgung über das Netzteil erfolgen und die maximale Auflösung 12 Bit, bei einem eingestellten Messbereich von 2 pF, betragen, ergibt sich eine theoretische Genauigkeit von ca. 0,488 fF. Das bedeutet, dass mit einer ungünstigen Stromversorgung, ähnlich der hier im Test benutzten, nur noch eine Genauigkeit des Winkelsensors von maximal 8 Bit erwartet werden könnte. Voraussetzung dafür wäre allerdings, dass durch den Rest des Aufbaus des kapazitiven Winkelsensors keine weiteren Ungenauigkeiten entstehen. Wird jedoch eine bessere Stromversorgung gewährleistet, so kann die Grundgenauigkeit bei ca. 10 Bit gehalten werden. Daher werden mehrere Maßnahmen ergriffen, um eine stabile und saubere Spannung auf der Platine des kapazitiven Winkelsensors bereitzustellen, vergleiche hierzu den Schaltplan in Abbildung 3.4.

Die erste Maßnahme ist eine Ferritperle in der Stromversorgungsleitung. Diese filtert hochfrequente Störsignale aus der Versorgungsspannung. Die Ferritperle ist direkt hinter der Steckbuchse platziert, somit tritt der positive Effekt möglichst früh im Schaltkreis ein. Folglich werden HF-Störungen früher gefiltert und können deutlich schlechter auf andere Leitungen übersprechen.

3. Sensoren

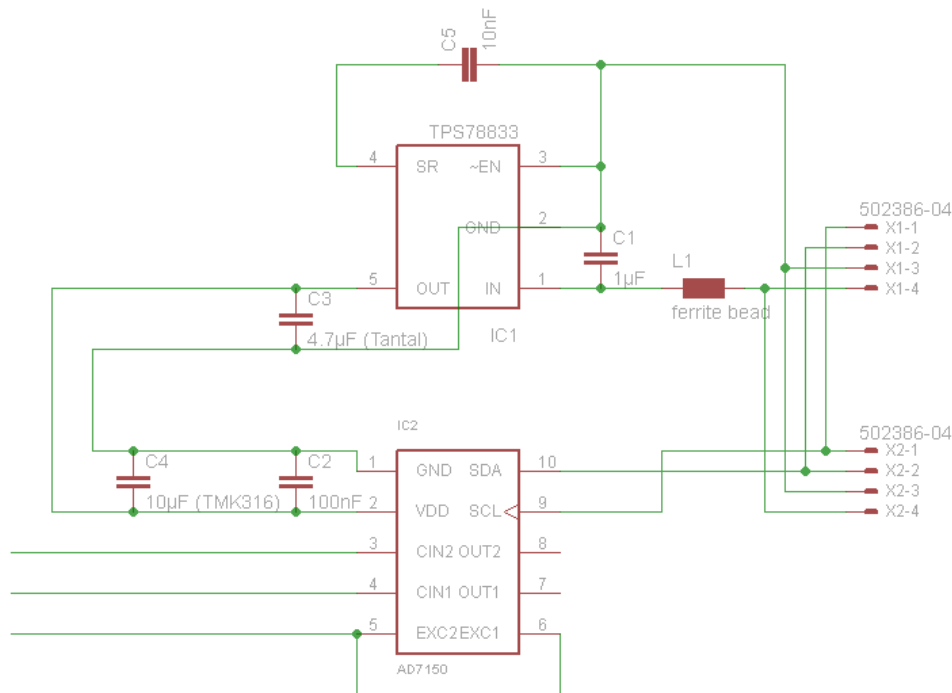


Abbildung 3.4.: Schaltplan der Hauptplatine des kapazitiven Winkelsensors.

Bei der verwendeten Ferritperle werden bereits Frequenzen ab einem Bereich von 10 MHz bis hin zu über 300 MHz effektiv abgeschwächt.

Außerdem wird nicht eine Spannung von 3,3 V zum Winkelsensor geleitet, sondern eine von 3,9 V. Diese wird dann mittels eines weiteren Linearreglers, dem *TPS78833*, auf 3,3 V geregelt, vergleiche [Nat01]. Dadurch werden die Schaltkreise vor und hinter dem Regler in gewissem Maße voneinander entkoppelt, ein sogenanntes *decoupling* findet statt. Die Abbildung 3.5 zeigt die Rauschunterdrückungsfähigkeiten des *TPS78833* in Abhängigkeit von der Frequenz unter bestimmten Bedingungen. Ersichtlich werden die Bedingungen links oben im Bild. Es sind zwei Kurven für eine unterschiedliche Leistungsaufnahme am Ausgang des Reglers abgetragen. Die maximale Leistungsaufnahme des *AD7150* liegt bei 120 μA und damit deutlich unter denen der Beispielkurven. Die Rauschunterdrückung dürfte daher etwas schlechter für die Anwendung im kapazitiven Sensor sein. Dennoch ermöglicht sie bis zu einer Störfrequenz von 1 kHz deutliche Verbesserungen der Versorgungsspannung. Die höherfrequenten Störungen sollten schon vorher von der Ferritperle eliminiert werden.

3. Sensoren

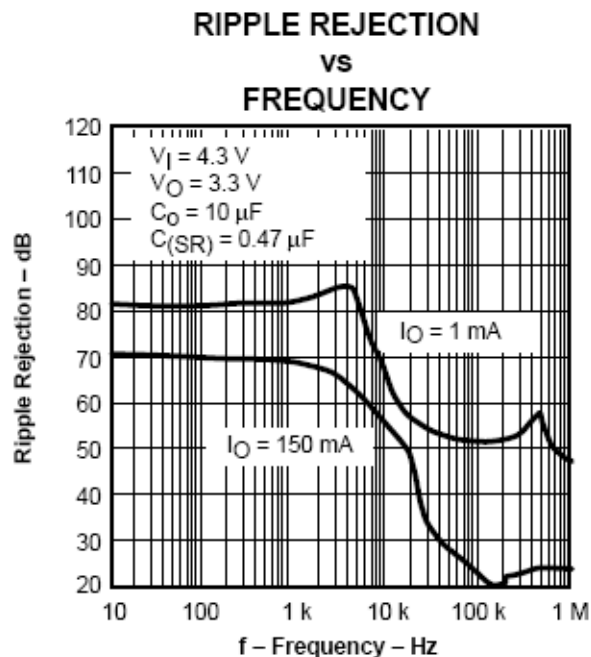


Abbildung 3.5.: Rauschunterdrückung in dB in Abhängigkeit der Frequenz des Linearreglers *TPS78833*. Quelle: [Nat01, S. 4]

Des Weiteren werden unterschiedlich große Stützkondensatoren vor bzw. hinter die ICs geschaltet. Das beschriebene Verfahren wird *bypassing* genannt. Das Datenblatt des Linearreglers empfiehlt, einen $1,0 \mu\text{F}$ Kondensator möglichst nahe an den Pins der Eingangsspannung zu platzieren, vergleiche [Nat01, S. 6]. Zusätzlich muss am Ausgang des Spannungsreglers ein Kondensator mit einer Kapazität von $4,7 \mu\text{F}$ vorhanden sein. Dieser sollte laut Datenblatt zwingend einen *Equivalent Series Resistance*-Wert (ESR) zwischen $0,2 \Omega$ und 10Ω haben. Ein Tantal-Elektrolytkondensator wird dafür empfohlen. Aus diesem Grund wird ein $4,7 \mu\text{F}$ Kondensator mit passenden Eigenschaften verwendet. Im Datenblatt des *AD7150* [AD07, S. 24] wird weiterhin empfohlen einen $10 \mu\text{F}$ - und 100 nF -Kondensator zwischen Versorgungsspannung und Masse zu schalten. Diese Empfehlung wurde in die Schaltung übernommen. Der 100 nF -Kondensator sollte dabei so nah wie möglich am IC liegen, um die Störungen am effektivsten filtern zu können. Hilfreiche Erläuterungen zum Thema der Rauschverminderung bei Spannungsversorgungen liefern [Kun04, AD, AD89]. Die genannte Literatur geht auf die bereits erwähnten Begriffe *bypassing* und *decoupling* ein.

3. Sensoren

Der 10 nF Kondensator zwischen dem SR-Pin des *TPS78833* und der Masse verzögert das Bereitstellen der Ausgangsspannung und damit das Starten des *AD7150*, um eine gewisse Zeitspanne. Der genaue Spannungsverlauf kann im verwiesenen Datenblatt des Spannungsreglers nachgeschlagen werden, vergleiche [Nat01, S. 1]. Der verzögerte Spannungsaufbau ist von Vorteil, da dadurch der Einschaltstrom des Gesamtsystems zeitlich besser verteilt wird.

3.1.3 Mechanisches Design – Gehäuse

Um verschiedenste physische Einwirkung von außen trotzen zu können, wurden mehrere Maßnahmen bezüglich des mechanischen Designs erstellt und getestet. Weitere schon in der Einleitung erwähnte Ansprüche sind ein geringes Gewicht und niedrige Kosten. Diese Ansprüche haben starke Auswirkungen darauf, welche Komponenten im Sensor verbaut werden können. Auf Grund des Messprinzips, das im Kapitel 3.1.1 erläutert wird, ist ein konstanter Abstand der beiden Messflächen unabdingbar. Schon kleinste Änderungen des Abstandes hätten große Auswirkungen auf die Kapazität und dadurch natürlich auch auf den gemessenen Winkel. Des Weiteren sollten die Messflächen immer exakt konzentrisch zueinander liegen, denn seitliche Verschiebungen der Flächen würden auch zu Messfehlern führen. Diese Verschiebungen treten bei einem Hohlachsenaufbau dann auf, wenn sich die Achse in Bezug zur Aufhängung ungewollt seitlich bewegt. Der Einbau eines Lagers zwischen den beiden Sensorteilen könnte das Problem lösen. Allerdings bedeutet das, dass zusätzliche Kosten und höheres Gewicht entstehen, daher wurde eine andere Herangehensweise gewählt. Statt ein schweres Lager einzubauen, werden die beiden Flächen durch ein passgenaues Gehäuse zueinander geführt und darüber hinaus wird die Achse von der Sensorgegenfläche, mittels eines elastischen Stoffes, mechanisch entkoppelt. Als elastischer Stoff wurde eine 2 mm dicke Schicht aus dichtem Schaumstoff genutzt. Diese wurde an der einen Seite an die Sensorgegenfläche und mit der anderen an ein Kunststoffteil geklebt, das auf die Achse geschoben und befestigt werden kann. Ein genauer Aufbau des Winkelsensors ist in Abbildung 3.6 in einer Explosionsansicht dargestellt. Durch den Schaumstoff werden Scherkräfte nicht mehr oder nur noch in geringem Maße auf den Sensoraufbau übertragen. Außerdem lässt sich durch den Schaumstoff eine kleine Andruckkraft zwischen den Sensorflächen erzeugen. Diese Kraft garantiert daher einen nahezu konstanten Abstand zwischen den beiden Flächen. Mit zunehmender

3. Sensoren

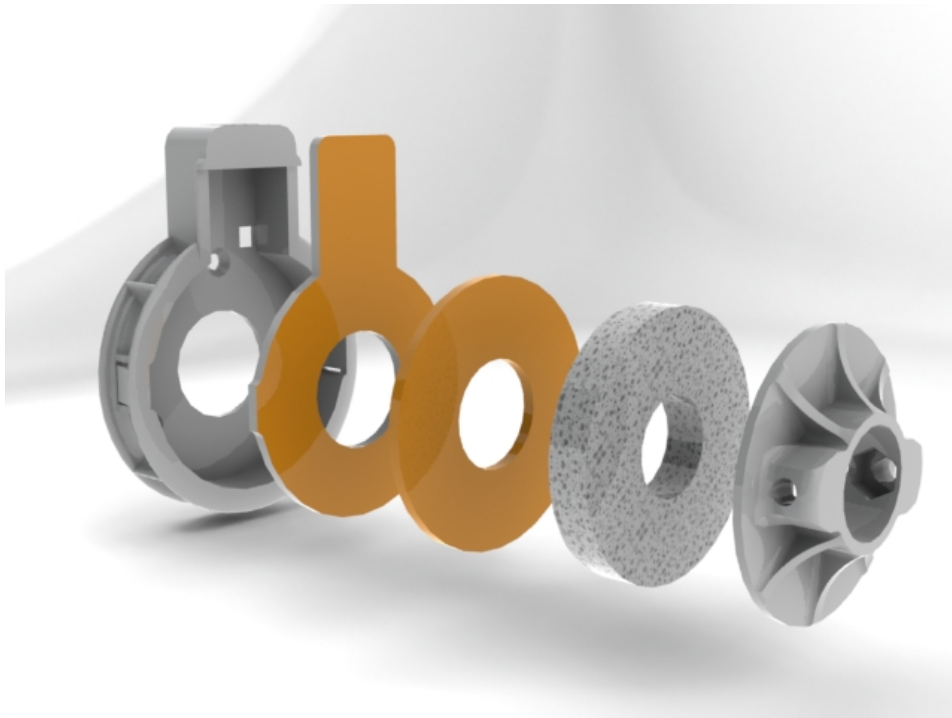


Abbildung 3.6.: Explosionsansicht des Aufbaus vom kapazitiven Winkelsensor.

Andruckkraft steigt aber auch die Reibung der Flächen zueinander, sodass der Sensor immer schwergängiger werden würde. Eine Abhilfe schafft hier eine Teflonfolie, die zwischen den Flächen platziert wird und somit die Reibung fast aufhebt. Gleichzeitig wirkt sie zusammen mit dem Lötstopplack der Grundplatte als Dielektrikum des erzeugten Plattenkondensators. Der Schaumstoff darf keinesfalls zu dick gewählt werden, da sonst ein merklicher Hystereseeffekt bei einer Richtungsumkehr der Achsbewegung auftreten würde. Bei der verbauten 2 mm-Schicht und dem verwendeten dichten Schaumstoff ist dieser Effekt allerdings für die Messgenauigkeit nicht mehr relevant. Das Gehäuse besteht aus ABS-Kunststoff und ist im Rapid-Prototyping-Verfahren mit einem 3D-Drucker entstanden. Die Konstruktion ist auf ein sehr geringes Gewicht von nur 3 g ausgelegt. Um trotzdem eine hohe Festigkeit zu erreichen, wurden am oberen und unteren Ende des Gehäuses dickere Verstärkungsringe platziert. Die Verstärkungsringe werden zusätzlich von kreisförmig angeordneten Verstärkungsrippen verbunden. Das Gehäuse des kapazitiven Winkelsensors ist in der Abbildung 3.6 (links außen) dargestellt. Das Gesamtgewicht des Sensors liegt bei ca. 11 g.

3. Sensoren

3.1.4 Kostenaufstellung

Die Kosten der verschiedenen Bauteile des kapazitiven Winkelsensors sind detailliert in der Tabelle 3.3 aufgelistet. Es wurde mit einer Anzahl von 25 Sensoren kalkuliert. Bei einer Abnahme von größeren Stückzahlen sinken einige Posten erheblich. Mit einem Anteil von über 50 Prozent ist die Platine das teuerste Bauteil des Sensors. Dafür bietet sie der Auswertungs elektronik Platz und stellt gleichzeitig die Messflächen.

Tabelle 3.3.: Kostenaufstellung des kapazitiven Winkelsensors.

Komponente	Preis in €
Platine	6,62
Gegenplatine	0,07
AD7150	2,54
TPS78833	0,83
Tantal-Kondensator	0,49
Steckbuchsen	~1,00
restliche elek. Komp.	~0,20
Gesamtkosten	11,75

Die Gesamtkosten sind mit 11,75 €, im Vergleich zu anderen Winkelsensoren, sehr gering. Das Ziel eine kostengünstige Herstellung zu ermöglichen wurde daher erreicht.

3.2 Winkelsensor mittels Potentiometer

Als Rückfalllösung und zur Nutzung an jenen Robotergelenken, bei denen eine konzentrische Platzierung des Winkelsensors nicht benötigt wird, soll der Einsatz eines Potentiometers zur Bestimmung des Winkels möglich sein. Dabei sollen die Spannungsversorgung und die Schnittstelle, im Vergleich zum kapazitiven Winkelsensor, unverändert bleiben.

3.2.1 Grundidee

Ein Potentiometer ist ein elektromechanisches Bauteil, das einen festgelegten Gesamtwiderstandswert besitzt. Dieser kann in zwei Teilwiderstände zerlegt werden. Das Verhältnis dieser beiden Widerstände kann mechanisch durch einen Schleifer verändert werden. Das Potentiometer verhält sich folglich wie

3. Sensoren

ein einstellbarer Spannungsteiler. Die Ausgangsspannung V_o kann mit folgender Formel berechnet werden:

$$V_o = \frac{r}{R} V_{ref},$$

wobei V_{ref} die Referenzspannung, r der Widerstand zwischen Schleifer und Masse und R der Gesamtwiderstand des Potentiometers ist, vergleiche [Eve95, S. 36ff].

Es werden verschiedene Potentiometer der Firma Vishay [Vis06] verbaut, die sich in der Bauhöhe und Lagerung der Welle, aber nicht bzw. nur gering in ihren elektrischen Eigenschaften und Anschlüssen, unterscheiden. Die große Bauhöhe des Potentiometers (ca. 18,8 mm zzgl. der Länge der Welle) verhindert eine sinnvolle Anbringung konzentrisch zur Achse des zu messenden Gelenks. Daher wird das Potentiometer versetzt zur Achse verbaut und mittels eines Seilzuges mit dieser verbunden. In Abbildung 3.7 sind zwei verbaute Potentiometer in einem Roboterbein dargestellt. Durch die vorliegende Art der Verbauung kann eine sehr genaue mechanische Kalibrierung des Winkelsensors garantiert werden. Dafür müssen die entsprechenden Gelenke in eine Referenzstellung gebracht werden. Am besten geeignet sind Gelenkstellungen am Anschlag, diese ermöglichen über alle Roboter hinweg eine fast identische Abwinklung. Daraufhin kann mit Hilfe einer Nabenschraube die Seilrolle der Achse gelöst und justiert werden. Ist weiterhin der aktuelle Messwert des Potentiometers bekannt, kann die Referenzstellung sehr genau mit einem absoluten Messwert synchronisiert werden.

Das Potentiometer hat einen Gesamtwiderstand von 10 k Ω und verfügt über drei Anschlüsse. Es liefert einen messtechnisch gültigen Widerstand im Bereich von 0 bis 340°. Über den restlichen Bereich werden keine Aussagen getroffen. Die Spannung am Schleifkontakt ändert sich nahezu linear mit der Winkeländerung der Welle. Die sich verändernde Spannung muss mit einer geeigneten Elektronik gemessen und per I²C-Schnittstelle versendet werden. Dazu muss folglich ein geeigneter *Analog-to-Digital-Converter* (ADC) benutzt werden. Auf einen ausführlichen Vergleich verschiedener ADCs wurde verzichtet. Stattdessen wurde ein einzelner, passender IC (*ADC121C027* – [Nat08]) mit ausreichender Genauigkeit von 12 Bit und I²C-Schnittstelle getestet. Ein Vorteil des ICs ist eine einstellbare I²C-Adresse, sodass mehrere dieser Chips

3. Sensoren

an einem I²C-Bus angeschlossen sein können. Mit einer Durchsatzrate von 188,9 kSPS bietet der *ADC121C027* eine weit höhere Abtastrate als die vom Gesamtsystem geforderten 100 Hz. Des Weiteren hat der IC eine sehr geringe typische Leistungsaufnahme von nur 0,26 mW und seine Baufläche beträgt inklusive Pins nur knapp 8,5 mm². Diese geringe Baugröße ermöglicht ein sehr kompaktes Design der gesamten Messvorrichtung (Potentiometer + Platine zur Messung). Die Platine der Auswertungs elektronik kann somit auf der Rückseite des Potentiometers an dessen Kontakten angebracht werden. Sollte trotz der geringen Größe der Platine kein Platz am Potentiometer selbst für sie vorhanden sein, kann die Platine auch in räumlicher Nähe platziert werden. Es sind zwei Verschraubungsbohrungen auf der Platine zur alternativen Befestigung vorhanden. Die Kontakte des Potentiometers werden dann mit kurzen Kabeln an den passenden Stellen der Platine verbunden. Eine Gegenüberstellung der beiden Verbauearten ist in Abbildung 3.7 ersichtlich.

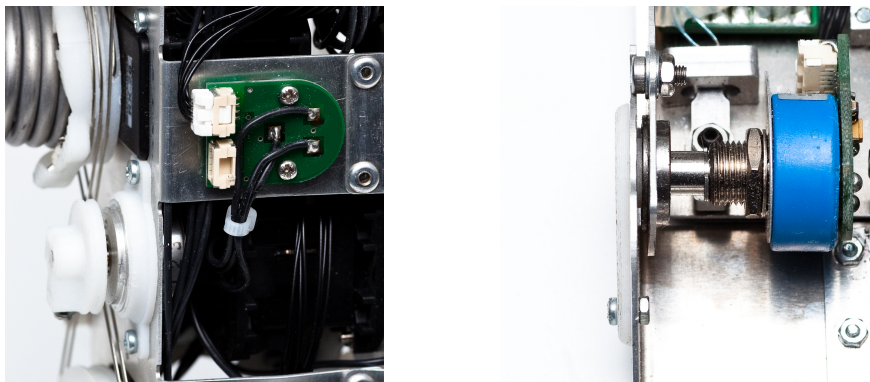


Abbildung 3.7.: Gegenüberstellung der Verbaueisen des Potentiometer mit dazugehöriger Auswertungs elektronik. Graues Potentiometer mit externer Platine (links). Blaues Potentiometer direkt mit der Platine verbunden (rechts).

3.2.2 Elektronisches Design

Es wurde bereits verdeutlicht, dass die Schnittstelle des Winkelsensors mittels Potentiometer zur in Abschnitt 3.1.2 vorgestellten Schnittstelle des kapazitiven Winkelsensors kompatibel sein soll. Die gleiche Datenübertragung ist durch die I²C-Schnittstelle des *ADC121C027* gewährleistet. Die Versorgungsspannung darf zwischen 2,7 V und 5,5 V betragen. Die vorliegenden 3,9 V könnten also schon direkt genutzt werden. Allerdings wird im Datenblatt [Nat08, S. 27] des ADCs eingehend darauf hingewiesen, dass die Versorgungsspan-

3. Sensoren

nung als Referenzspannung für die Messung genutzt wird und folglich ohne jede Störung anliegen sollte. Daher wurde der Schaltungsteil der Versorgungsspannung des kapazitiven Winkelsensors, der eine nahezu störungsfreie Versorgungsspannung des ICs ermöglicht, fast identisch übernommen. Es wurde also wieder der Linearregler *TPS78833* [Nat01] benutzt, der die Spannung auf konstante 3,3 V regelt. Die Stützkondensatoren, die sich direkt an dem ADC befinden, wurden ihren Anforderungen entsprechend angepasst. Diese betragen beim *ADC121C027* 100 nF und 4,7 μ F. Die Abbildung 3.8 zeigt den gesamten Schaltplan der Auswertungslektronik. Wie im Abschnitt 3.2.1 erwähnt, können am *ADC121C027* unterschiedliche I²C-Adressen eingestellt werden. Dies geschieht über einen Pin des ICs. Im Schaltplan wird dieser Pin mit ADDR bezeichnet. Es sind drei unterschiedliche Adressen möglich, dazu muss der Pin entweder zum niedrigen oder hohen Potential gezogen werden oder potentialfrei bleiben. Um eine möglichst hohe Flexibilität zu gewährleisten, ist dafür ein *SMD-Jumper* auf der Platine vorgesehen, der im Schaltplan mit SJ1 bezeichnet wird, vergleiche Abbildung 3.8. Durch die Verwendung eines *Jumpers* kann jederzeit die I²C-Adresse für jede Platine separat geändert werden.

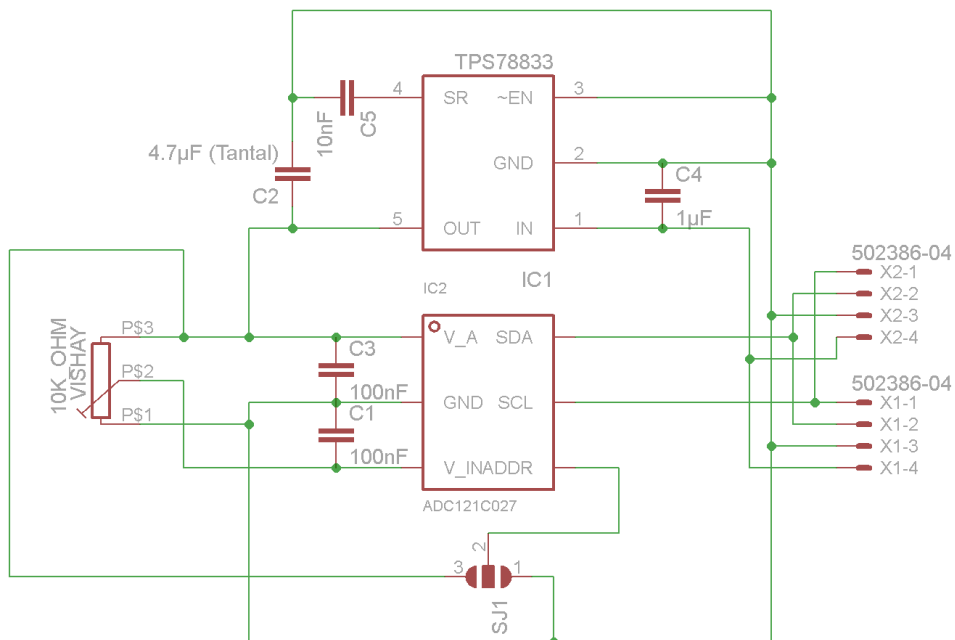


Abbildung 3.8.: Schaltplan der Platine zur Messung des ohmschen Widerstandes eines Potentiometers.

3. Sensoren

3.2.3 Kostenaufstellung

Die Gesamtkosten des Winkelsensors mittels Potentiometer setzen sich, wie in der Tabelle 3.4 angegeben, zusammen. Es sind zwei Varianten berechnet, die sich im verwendeten Potentiometer unterscheiden. Das teurere Potentiometer besitzt eine gelagerte Welle. Die günstigere Alternative muss außerhalb gelagert werden. Wie beim kapazitiven Sensor wurde wieder mit einer Stückzahl von 25 kalkuliert.

Tabelle 3.4.: Kostenaufstellung des Winkelsensors mittels Potentiometer.

Komponente	Preis in €
Potentiometer	~12,00 / ~25,00
Platine	3,70
ADC121C027	2,92
TPS78833	0,83
Tantal-Kondensator	0,49
Steckbuchsen	~1,00
restliche elek. Komp.	~0,10
Gesamtkosten	~21,04 / ~34,04

Im M-Serie-Roboter wird fast ausschließlich das Potentiometer mit Lager verbaut. Diese sind mit einem Preis von ca. 35 € dreimal so teuer wie der kapazitive Winkelsensor.

3.3 Accelerometer

Accelerometer können Beschleunigungen mindestens einer Richtung erfassen. Außerdem kann auf Grund der Erdbeschleunigung die Lage des Accelerometers im Raum bestimmt werden, sofern die Erdbeschleunigung nicht durch eine andere Beschleunigung überlagert wird. Wenn jedes Körperteil eines Roboters mit diesen Sensoren in allen drei räumlichen Achsen ausgestattet wäre und er sich in Ruhe befinden würde, könnte die aktuelle Pose des Roboters allein anhand der Messwerte der Accelerometer bestimmt werden. Es scheint in Folge dieser Überlegung sinnvoll zu sein möglichst viele verteilte Sensoren dieser Art auf dem Roboter zu platzieren. Wie in Abschnitt 2.2 schon erläutert wurde, sind die Accelerometer direkt auf dem *AccelBoard3D* verbaut worden,

3. Sensoren

sodass eine Vielzahl dieser Sensoren verteilt auf allen Körperteilen des Roboters vorhanden sind.

Eine standardisierte Schnittstelle, die bei den externen Sensoren Voraussetzung war, ist bei dem Accelerometer nicht nötig, da es wie beschrieben auf der Platine des *AccelBoard3D* platziert wurde. Jedoch sollte es die Beschleunigung aller drei räumlichen Achsen unabhängig voneinander messen und darüber hinaus große Beschleunigungen erfassen können. Auf Grund dessen wird ein integrierter Schaltkreis mit der Typenbezeichnung *MMA7455L* eingesetzt, vergleiche [Fre09]. Jener Schaltkreis kann mit einer Auflösung von 10 Bit Beschleunigungen von bis zu 8 g messen, d.h. das Achtfache der Erdbeschleunigung. Die kapazitive Sensoreinheit zur Messung der Beschleunigung und die Auswertungelektronik befinden sich beide im gleichen Gehäuse des *MMA7455L*. Die Messeinheit ist dabei hermetisch von den anderen Elementen getrennt und funktioniert nach folgendem Prinzip: Wie in Abbildung 3.9 (links) dargestellt, kann man sich den Messapparat als zwei fixierte Balken mit einem bestimmtem Abstand, der sich nicht verändert, vorstellen. Dazwischen befindet sich ein weiterer Balken, der nicht fixiert ist und sich daher zwischen den anderen Balken hin und her bewegen kann. Dabei werden zwei entstehende Kapazitäten zur Messung benutzt. Zum einen die Kapazität zwischen dem linken und mittleren Balken und zum anderen die Kapazität zwischen dem rechten und mittleren Balken. Die entsprechende Schaltung ist in der Abbildung 3.9 (rechts) angedeutet. Bei einer Beschleunigung nach rechts wird sich der mittlere Balken durch seine Trägheit, relativ zu den anderen Balken gesehen, nach links bewegen. Durch den veränderten Abstand der Balken zueinander, ändern sich auch die Kapazitäten der beiden Kondensatoren. Der IC *MMA7455L* schließt dann durch die entsprechende Differenz der beiden Messwerte auf die gegebene Beschleunigung.

Die Versorgungsspannung des Chips ist in eine digitale und analoge Spannung aufgeteilt, von denen die analoge möglichst störungsfrei sein sollte. Beide Spannungen sollten zwischen 2,4 V und 3,6 V liegen. Wie in der Einleitung beschrieben, soll das Accelerometer direkt auf der Platine des *AccelBoard3D* liegen. Dies hat zur Folge, dass die Funktionstüchtigkeit der Schaltung, die das Accelerometer betrifft, schon vor der Fertigung des *AccelBoard3D* getestet werden muss. Bei den externen Sensoren konnte dies bis jetzt immer unabhängig geschehen. Bei der ersten *Development*-Version des *AccelBoard3D* wurde die

3. Sensoren

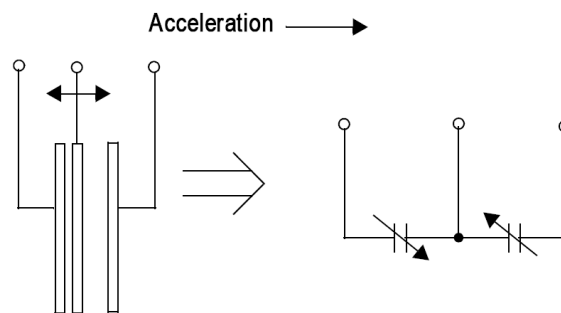


Abbildung 3.9.: Schematische Darstellung der Funktionsweise des Accelerometers. Quelle: [Fre09, S. 8]

SPI-Schnittstelle per Buchse nach außen gelegt, sodass damit das Accelerometer extern getestet werden konnte. Abbildung 3.10 zeigt ein Foto dieser ersten *AccelBoard3D*-Version, inklusive der Testschaltung des Accelerometers. Wie das Foto zeigt, wurde das Accelerometer an die Buchse der SPI-Schnittstelle angeschlossen. Die Spannungsversorgung, als rotes und blaues Kabel in der Abbildung zu erkennen, wurde über einen weiteren Stecker verbunden, um die Spannungsquelle schneller austauschen zu können. Vom *Development*-Board wird eine Spannung von 3,3V bereitgestellt, die direkt durch den Schaltregler *MAX1836* [MIP06] erzeugt wird. Auf der Seite des Accelerometers wurden jeweils für den analogen und digitalen Spannungseingang Stützkondensatoren mit den Werten von $10\ \mu\text{F}$ und $100\ \text{nF}$ – empfohlen im Datenblatt [Fre09, S. 20] – platziert. Lag das Accelerometer in fast vollkommener Ruhe, wiesen die Messdaten, in vorher beschriebener Konfiguration, eine Standardabweichung von ca. $0,045\ \text{g}$ auf. Die größten Abweichungen erreichten eine Höhe von $\pm 0,15\ \text{g}$, dies entspricht fast 2% des gesamten Messumfangs.

Die Dimension des Fehlers kann durch eine praktische Problemstellung greifbarer gemacht werden. Folgende Aufgabe könnte auftreten: Anhand der gemessenen Beschleunigung soll auf die Lage im Raum geschlossen werden, sofern keine andere überlagernde Beschleunigung vorliegt. Angenommen das Accelerometer ist in waagerechter Position, dann sollte der gemessene Wert der Beschleunigung in der X-Achse konstant $0\ \text{g}$ betragen. Wir wissen, dass in der oben beschriebenen Konfiguration, die Werte um bis zu $0,15\ \text{g}$ in positiver und negativer Richtung abweichen. Welche Winkelabweichung kann also erwartet

3. Sensoren

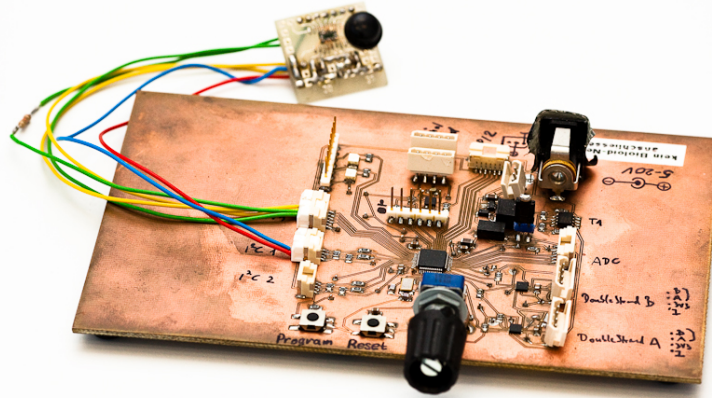


Abbildung 3.10.: Foto der ersten *Development*-Version des *Accelboard3D* mit angeschlossenem Testaufbau des MMA7455L-Accelerometers.

werden? Die zu erwartende Kraft in Abhängigkeit des Auslenkungswinkels berechnet sich wie folgt,

$$a = \sin(\alpha)g.$$

Wobei a die gemessene Beschleunigung des Sensors ist, g die Erdbeschleunigung und α der gesuchte Auslenkungswinkel. Eine Umstellung nach α ergibt folgende Formel,

$$\alpha = \arcsin\left(\frac{a}{g}\right).$$

Durch Einsetzen des größten zu erwartenden Fehlers von 0,15 g in a ergibt sich eine mögliche Winkelabweichung von ca. 8,65°. Wollte man einen humanoiden Roboter mit Hilfe der Beschleunigungswerte gerade Stehen lassen oder sogar Balancieren lassen, wird es mit einer Winkelabweichung von bis zu 8,65° unmöglich sein. Da dies aber möglich sein sollte, müssen die Messwerte des Beschleunigungssensors deutlich verbessert werden.

Der erste Ansatz ist wieder die besten Grundvoraussetzungen für den integrierten Schaltkreis zu schaffen. Beim *MMA7455L* bedeutet es die Spannungsversorgung des analogen Spannungseingangs möglichst konstant – frei von Störungen – zu halten. Daher wurde statt der 3,3 V starken Spannung am SPI-

3. Sensoren

Anschluss des *Development-AccelBoard3D* eine andere Versorgungsspannung gewählt. Dazu wurde an einer anderen Stelle des *AccelBoard3D* eine Spannung von 12 V abgegriffen, die mittels Linearregler auf die passende 3,3 V-Spannung für das Accelerometer gesenkt wurde. Der Linearregler war ein LP2950-IC, dessen Fähigkeiten der Rauschunterdrückung im Datenblatt [Nat09b, S. 11] nachgeschlagen werden können. Eine Gegenüberstellung der Qualitäten, im Hinblick auf die Ausgangsspannung der beiden unterschiedlichen Regler, ist in Abbildung 3.11 dargestellt. Dabei wurden für eine kurze Zeitspanne die Spannungsabweichungen mittels digitalem Oszilloskop aufgezeichnet.

Auf der linken Seite wurde die Messung in der Ausgangskonfiguration – mit Schaltregler – durchgeführt. Deutlich erkennbar sind die hohen Ausschläge von $\pm 0,25$ V, die ca. alle 40 μ s, das entspricht 25 kHz, auftreten. Mit großer Wahrscheinlichkeit treten sie durch die Schaltungsvorgänge des MAX1836-Reglers auf. Unter gegebenen Voraussetzungen, der benutzten Eingangsspannung von 12 V und einer geringen Stromlast am Ausgang, hat dieser eine Schaltfrequenz, die mit der Störfrequenz nahezu identisch ist. Im Datenblatt [MIP06, S. 5] sind einige Diagramme dargestellt, anhand dieser die Schaltfrequenz bestimmt werden kann.

Im Vergleich dazu sind auf der rechten Seite der Abbildung 3.11 die Ergebnisse der Messung unter Verwendung des vorgestellten Linearreglers abgetragen. Es sind zwar noch größere Störungen in den gleichen Abständen vorhanden, aber diese sind um den Faktor fünf schwächer. Um festzustellen, ob das Accelerometer unter den neuen Bedingungen bessere Messwerte liefert, wurden jeweils 100 Messwerte in 10 ms-Abständen vor und nach der Änderung der Spannungsversorgung erhoben. Das Ergebnis ist als Histogramm in Abbildung 3.12 dargestellt. In roter Farbe sind die Werte des Accelerometers mit der Spannungsversorgung direkt vom *AccelBoard3D* abgetragen, in blau die Anderen. Wie bereits erwähnt, war die Standardabweichung der gemessenen Daten 0,045 g vor der veränderten Spannungsquelle, danach liegt sie bei nur noch 0,008 g. Es ergibt sich eine Verbesserung um den Faktor fünf. Die maximale Abweichung liegt jetzt bei deutlich besseren $\pm 0,04$ g. Für unser Rechenbeispiel bedeutet dies folgende Veränderung: Statt der Winkelabweichungen von $\pm 8,65^\circ$ ist nur noch eine Abweichung von $\pm 2,30^\circ$ zu erwarten.

3. Sensoren

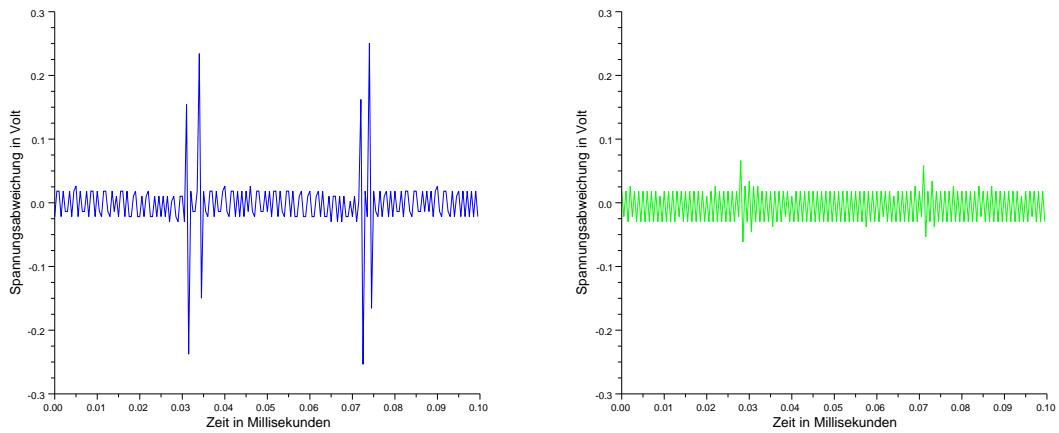


Abbildung 3.11.: Spannungsabweichungen der unterschiedlichen Spannungsquellen: *MAX1836* (links), *LP2950* (rechts).

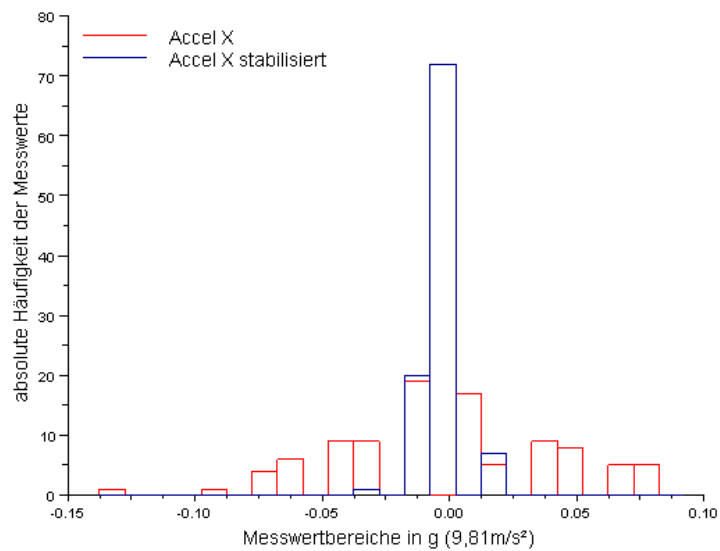


Abbildung 3.12.: Vergleich der ermittelten Messdaten des Accelerometers mit unterschiedlichen Spannungsquellen. Als Darstellung wurden zwei übereinander gelagerte Histogramme verwendet.

3. Sensoren

3.4 Kraftsensoren im Fuß

Die Kraftsensoren im Fuß dienen zur Messung der Kraft, die auf die Fußsohle des Roboters wirkt, welche durch dessen Eigengewicht und durch dynamische auftretende Kräfte bedingt ist. Interessant ist dabei vor allem die Kraftverteilung. Es werden insgesamt vier einzelne Kraftsensoren in der Fußsohle verbaut, wobei sich in jeder der vier Ecken ein Sensor befindet. Werden diese hypothetisch miteinander verbunden, so würden sie ein Rechteck aufspannen. Wenn der Massenschwerpunkt des Roboters lotrecht über dem Mittelpunkt dieses Rechteckes liegt, wirkt auf alle Sensoren die gleiche Kraft. Voraussetzung dafür ist, dass keine dynamischen Kräfte auftreten. Beim Auftreten von dynamischen Kräften, müsste stattdessen der *Zero Moment Point* über dem Mittelpunkt des Rechtecks liegen, damit die Sensoren wieder alle den gleichen Wert messen. Zumindest im statischen Fall könnte somit ein Verlieren des Gleichgewichts sehr leicht detektiert werden.

3.4.1 Elektronisches Design

Als Grundlage der Messvorrichtung zur Bestimmung der Kräfte dient ein Faraday-Kraftsensor, vergleiche [IB 06]. Er kann Kräfte bis zu einer Höhe von 100 N messen und hat dabei einen Hub von unter 5 μm . Die Genauigkeit liegt laut Datenblatt bei 0,002% des maximalen Messwertes, also bei 2 mN. Der Linearitätsfehler wird mit unter einem Prozent, bei einer maximalen Belastung von bis zu 60 N, angegeben. Des Weiteren ist er durch seine geringe Größe und dem damit einhergehendem geringen Gewicht, sehr gut für unsere Zwecke geeignet. Genaue Angaben können aus der Abbildung 3.13 entnommen werden. Der Faraday-Kraftsensor basiert, wie der in Abschnitt 3.1 beschriebene Winkelsensor, auf einer sich verändernden Kapazität. Um ihn nicht anfällig für äußere Störungen zu machen, wird er von einem schützenden Faraday-Käfig abgeschirmt. Im unbelasteten Zustand beträgt seine Grundkapazität 5 pF. Die maximale Kapazität unter Vollast ist 11 pF groß. Diese Tatsache ermöglicht den Einsatz der gleichen Auswertungs elektronik, die auch schon beim kapazitiven Winkelsensor benutzt wurde. Der in der Auswertungs elektronik benutzte CDC *AD7150* [AD07] kann nämlich Kapazitäten in einem Bereich von 0 pF bis 13 pF messen. Weitere Anpassungen sind daher nicht notwendig. Die dafür zu Grunde liegende Schaltung wurde ausführlich im Abschnitt 3.1.2 diskutiert.

3. Sensoren

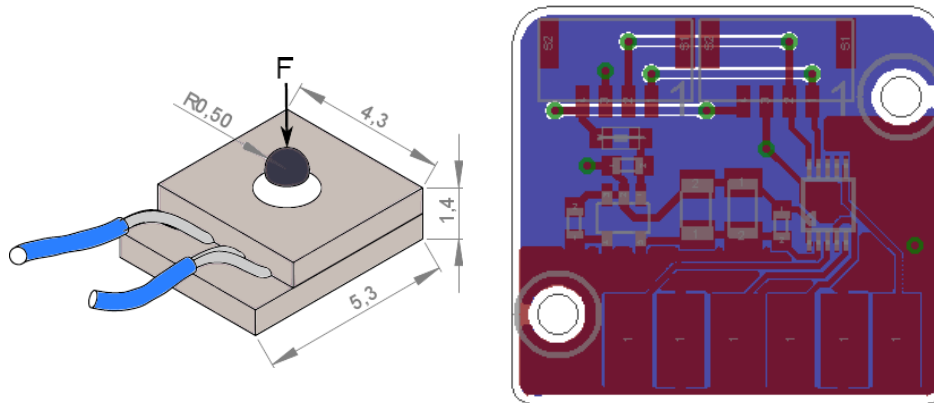


Abbildung 3.13.: Schematische Darstellung des Faraday-Kraftsensors (links).
Quelle: [LSB09]. Autor: Torsten Siedel. Layout der Auswertungs elektronik des
Faraday-Kraftsensors (rechts).

Der *AD7150* ist in der Lage zwei unabhängige Kapazitäten zu messen, daher können zwei Faraday-Kraftsensoren an einer Platine angeschlossen werden. Die Platine aus Abbildung 3.1 (links) wurde folgendermaßen geändert: Statt der unteren mit Buchstaben gekennzeichneten Flächen wurden insgesamt sechs Kontakte auf der Platine platziert. Diese befinden sich im unteren Bereich des Layouts, vergleiche Abbildung 3.13 (rechts). Die Kontaktflächen können mit Drähten oder Kabeln durch ein geeignetes thermisches Verfahren verbunden werden. Jeweils drei Kontakte gehören zusammen, von denen zwei für die Kapazität und einer für die Abschirmung ist. In unserem Fall werden dort die Koaxialkabel des Faraday-Sensors befestigt. Das heißt, es müssen insgesamt zwei Platinen zur Messung der vier Kapazitäten im Fuß untergebracht werden. Diese werden dann wieder per I²C-Schnittstelle mit einem *AccelBoard3D* verbunden.

3.4.2 Mechanisches Design

Die Kraftsensoren sind, wie in Abbildung 3.14 (links), in dem Fuß der *M-Serie* verbaut. Zur besseren Kraftübertragung – nur lotrechte Kraft soll auf den Sensor ausgeübt werden – wird der Sensor über eine Hebelmechanik mit der Fußsohle verbunden. Dies ist schematisch in der Grafik 3.14 (rechts) dargestellt. Über die Hebelmechanik kann außerdem ein Übersetzungsverhältnis

3. Sensoren

zwischen der Kraft, die auf die Fußsohle einwirkt, und der Kraft, die auf den Sensor ausgeübt wird, geschaffen werden. Dadurch kann zum Beispiel der gültige Messbereich vergrößert oder die Genauigkeit eines kleineren Bereiches verbessert werden. Die nach der Übersetzung lotrecht auf den Sensor einwirkende

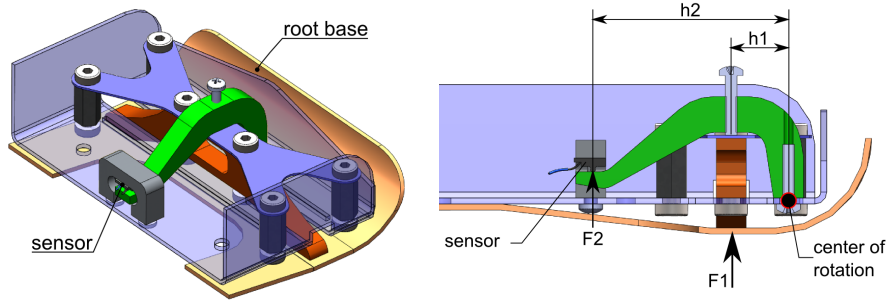


Abbildung 3.14.: Schematische Darstellung eines einzelnen Kraftsensors, verbaut im Fuß (links). Aufbau der Kraft reduzierenden Hebelmechanik (rechts). Quelle: [LSB09]. Autor: Torsten Siedel.

Kraft F_2 kann nach folgender Formel berechnet werden:

$$F_2 = \frac{h_1}{h_2} F_1 + F_0.$$

F_0 ist die Kraft, die aus einer mechanischen Vorspannung des Hebels resultiert. Die Bedeutung der restlichen Größen wird aus der Darstellung 3.14 (rechts) ersichtlich.

Eine grobe mechanische Kalibrierung der Kraftsensoren ist über eine Einstellschraube möglich. Sie sorgt auch für eine Spielfreiheit zwischen dem Hebel und dem Sensor. Eine exakte Kalibrierung kann in der Firmware vorgenommen werden, sodass alle Kraftsensoren den gleichen Messwert im unbelasteten Zustand liefern. Weitere technische Einzelheiten zum Fuß des M-Serie-Roboters werden im Artikel [LSB09] erläutert.

3.5 Interne Winkelsensoren der Dynamixel-RX-28-Servomotoren

Zum Antreiben aller für die Bewegungssteuerung wichtigen Freiheitsgrade des M-Serie-Roboters werden RX-28-Servomotoren eingesetzt. Die *RX-28* werden per RS485-Schnittstelle an die *AccelBoard3D* angeschlossen, vergleiche

3. Sensoren

hierzu Kapitel 2 und [RC08]. Damit der *STM32-Mikrocontroller* des *AccelBoard3D* die Servos ansprechen kann, muss ein *Transceiver*, der die Signale des Mikrocontrollers in RS485-konforme Signale umwandelt, eingesetzt werden. Deshalb wurde bei der Erstellung der Schaltung des *AccelBoard3D* ein LTC2854-IC integriert, vergleiche Abbildung A.2 und [Lin07]. Über die RS485-Schnittstelle kann die aktuelle Position des Ruderhornes abgefragt werden. Der *RX-28* besitzt dafür ein intern verbautes Potentiometer. Dieses liefert gültige Messwerte über einen Bereich von ca. 300° . Die Auflösung wird im Datenblatt mit $0,29^\circ$ angegeben. Der vom *RX-28* versendete Messwert hat folglich eine Genauigkeit von 10 Bit. Weitere Details zur Kommunikation mit den Servomotoren werden im Abschnitt 4.2 erläutert.

3.6 Spannung und Strommessung

Am *AccelBoard3D* sind zwei Anschlüsse vorhanden, an denen RX-28-Servomotoren angeschlossen werden können, vergleiche Abschnitt 2.2. Die Anschlüsse werden mit *DoubleStrand A* und *DoubleStrand B* bezeichnet. Es können jeweils bis zu zwei Servomotoren angeschlossen werden. Je nach eingestelltem Drehmoment und am Servo anliegender Last wird von den Motoren eine gewisse elektrische Leistung aufgenommen. Die aufgenommene elektrische Leistung P kann aus der Spannung U und der Stromstärke I berechnet werden. Dazu dient folgende Formel:

$$P = U \cdot I.$$

Daher werden genau diese beiden physikalischen Größen an den *DoubleStrands* gemessen. Die Spannung wird nur einmal erhoben, da sie an den relevanten Stellen identisch sein sollte. Spannungseinbrüche durch eine hohe aufgenommene elektrische Leistung, verursacht durch die Servomotoren, treten an beiden *DoubleStrands* gleichermaßen auf. Die Stromstärke wird allerdings für beide Anschlüsse separat gemessen. Sind mehrere Motoren an einem *DoubleStrand* angeschlossen, kann nur die Gesamtsumme des Stromes gemessen werden. Für die Strommessung sind zwei bidirektionale Stromstärke-Sensoren in die Schaltung des *AccelBoard3D* integriert, vergleiche Schaltplan A.2. Es werden ICs mit der Typenbezeichnung LMP8601 verwendet. Für genauere Angaben wird auf das Datenblatt [Nat09a] verwiesen. Der IC liefert je nach gemessener Stromstärke eine gewisse analoge Spannung. Es müssen also insgesamt drei analoge

3. Sensoren

Spannungen bestimmt werden. Der interne A/D-Wandler des auf dem *AccelBoard3D* verbauten *STM32-Mikrocontrollers* übernimmt diese Aufgabe. Er hat eine Auflösung von 12 Bit. Für die am *DoubleStrand* anliegende Spannung muss ein geeigneter Spannungsteiler verwendet werden. Dieser sollte die Spannung auf einen für den *STM32-Mikrocontroller* messbaren Bereich teilen. Die softwareseitige Realisierung wird im Abschnitt 4.2 diskutiert.

3.7 Weitere Sensorik

Abschließend werden die restlichen Sensoren vorgestellt. Sie werden nur der Vollständigkeit halber erwähnt, da sie entweder noch nicht im Roboter implementiert sind oder im Falle der Kamera, nur mit geeigneter Aufarbeitung der Daten für die Bewegungssteuerung relevant sind.

3.7.1 Temperatur im Akkumodul

Die Messung der Temperatur im Akkumodul soll über Thermistoren erfolgen. Diese sind in den austauschbaren Akkupacks des Roboters verbaut. Sollte die Temperatur einer Akkuzelle ungewollt steigen, würden die Thermistoren den ungewollten Vorgang registrieren können. Die Thermistoren haben einen elektrischen Widerstand, der von der Temperatur abhängig ist. Dadurch kann die Temperatur von einem einfachen A/D-Wandler digitalisiert werden. Der Thermistor muss dazu nur, mit einem festen Widerstand zusammen, als Spannungsteiler einer Referenzspannung fungieren. Der interne A/D-Wandler des Mikrocontrollers könnte die Spannung messen.

3.7.2 Kapazitive Berührungssensoren

Kapazitive Berührungssensoren, auch häufig *Touch*-Sensoren genannt, registrieren eine Annäherung eines leitfähigen Objekts an ihre Sensorflächen. Der Roboter könnte also feststellen, ob er berührt wird, vorausgesetzt in der Nähe der Berührung würde sich eine Sensorfläche befinden. Eine Beschreibung von speziellen Sensorflächen, wie *Buttons* und *Sliders*, ist in [AD08] zu finden. Beim M-Serie-Roboter könnten die Flächen großräumig an der Innenseite der Roboterverschalung angebracht und per Koaxialkabel mit einer passenden Auswertungselektronik verbunden werden. Die im Abschnitt 3.4.1 vorgestellte Schaltung, zur Bestimmung der Kapazität eines Faraday-Kraftsensors, kann als passende Auswertungselektronik eingesetzt werden. Für die Datenerhebung

3. Sensoren

sollte die Elektronik wieder per I²C-Schnittstelle mit einem *AccelBoard3D* verbunden werden.

3.7.3 Kamera

Im Kopf des humanoiden Roboters befindet sich eine Kamera mit der Typenbezeichnung 21K15XDIG. Die Kamera besitzt einen 1/4"-IL-CCD-Sensor mit einer Auflösung von 752 x 582 Bildpunkten. Die typische Konfiguration ist der *Interlaced*-Modus, bei dem 50 Halbbilder pro Sekunde übertragen werden. Eine alternative Konfiguration ist die Übertragung von 50 Vollbildern mit halbiertes vertikaler Auflösung. Die Bilddaten werden über zwei unterschiedliche Schnittstellen zur Verfügung gestellt. Zum einen über einen analogen Ausgang (Composite 1Vp-p CVBS) und zum anderen über eine digitale, parallele 8-Bit-Schnittstelle im YUV:422-Format. Die Konfiguration der Kamera erfolgt über weitere digitale Leitungen, die nach dem I²C-Standard angesprochen werden müssen. Alle digitalen Signalleitungen der Kamera sind mit dem *Field Programmable Gate Array* des *BrainModules* verbunden, vergleiche Abschnitt 2.1.

4 Firmware

In diesem Kapitel wird die auf dem *STM32-Mikrocontroller* ausgeführte Software vorgestellt, die in Teilen im Zuge dieser Arbeit programmiert wurde. Es wurde mit dem Programm *IAR Embedded Workbench* in der Sprache C programmiert. Zuerst wird kurz eine allgemeine Beschreibung gegeben und Näheres zum Timing erläutert. Der Vollständigkeit halber wird dabei auch auf Teile der Firmware eingegangen, die nicht von mir programmiert wurden. An jenen Stellen wird im Text explizit darauf hingewiesen. Anschließend wird die softwareseitige Einbindung der gesamten Sensorik des M-Serie-Roboters im Detail beschrieben und begleitend dazu die Eigenschaften der Schnittstellen erläutert. Im letzten Abschnitt wird die Erhebung von Kalibrierungsdaten der Sensoren erklärt und auf die Nutzung dieser Daten in der Firmware eingegangen.

4.1 Allgemeine Beschreibung und Timing

Das *AccelBoard3D* und die darauf ausgeführte Software, im weiteren Verlauf mit Firmware bezeichnet, bilden den Kern der elektronischen Architektur des M-Serie-Roboters. Dieser Aspekt begründet sich durch die Mannigfaltigkeit der Aufgaben, die vom *AccelBoard3D* und der dazu gehörenden Firmware erfüllt werden. Kommunikation untereinander, Befehle von außen entgegennehmen, externe ICs initialisieren, Sensorwerte abfragen und Servomotoren ansteuern sind nur eine kleine Auswahl der Aufgaben die, unter Einhaltung eines bestimmten Timings, erledigt werden müssen. Die Firmware kann daher in verschiedene Phasen bzw. Zustände unterteilt werden. Die wichtigsten sind dabei die Initialisierungsphase, direkt nach dem Einschalten, und die Standardbetriebsphase. Die Standardbetriebsphase ist zyklisch und wird daher andauernd ausgeführt. Sie ist dabei immer exakt 10 ms lang. Dieser Zeitabschnitt wird im Folgenden mit Zeitslot bezeichnet. Die Zeit von 10 ms wird durch den Einsatz eines exakten *Timers* auf dem Mikrocontroller gewährleistet und hat eine Frequenz von 100 Hz zur Folge.

Die erste Grafik in Abbildung 4.1 zeigt einen vollständigen Zeitslot eines *AccelBoard3D*. Der Zeitslot ist in mehrere Unterabschnitte geteilt, in denen bestimmte Aufgaben erledigt werden. Im ersten Abschnitt, der mit *State 0* bezeichnet ist, wird die Kommunikation aller im globalen Bussystem teilneh-

4. Firmware

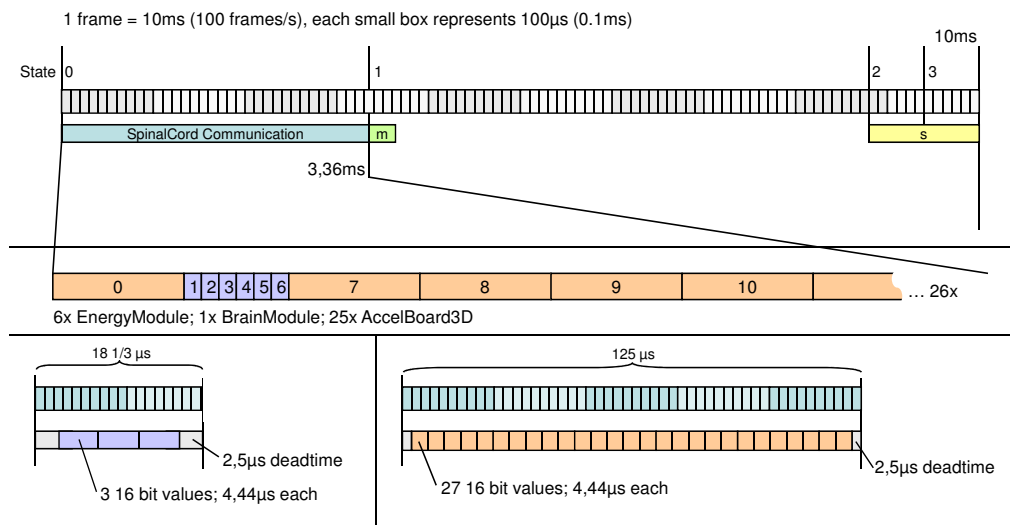


Abbildung 4.1.: Aufteilung eines Zeitslots (oben). Reihenfolge aller Kommunikationsteilnehmer am globalen Datenbussystem (mittig). Beschreibung und Dauer der Kommunikationsslots (unten). Autoren: Christian Thiele, Manfred Hild.

menden Partner vollzogen. Das globale Bussystem wird beim M-Serie-Roboter mit *SpinalCord* bezeichnet, vergleiche Abschnitt 2.3. Die Partner sind dabei vorrangig *AccelBoard3D*, eine Ausnahme bildet der erste Teilnehmer, dieser ist ein FPGA, der sich im Kopf auf dem *BrainModule* befindet. Außerdem sind die Teilnehmer zwei bis sieben speziell angepasste Versionen des *AccelBoard3D*, die *EnergyModules*. Diese werden für die Verwaltung der einzelnen Akkupacks des Roboters zuständig sein und haben nur den Mikrocontroller und Teile der Firmware mit den *AccelBoard3D* gemeinsam. Die Kommunikation ist folgendermaßen geregelt: Jedes *AccelBoard3D* bzw. *EnergyModule* hat eine eindeutige ID. Die ID bestimmt zu welchem Zeitpunkt das Board, mit der jeweiligen ID, seine Daten über den globalen Datenbus versenden darf. Der Zeitpunkt wird dabei relativ zum ersten Teilnehmer bestimmt. Für den Betrieb ist es nicht erforderlich, dass alle aufgelisteten Boards am *SpinalCord* angeschlossen sind. Darüber hinaus ist es sogar möglich die Anzahl der Teilnehmer dynamisch zur Laufzeit zu verändern. Durch die beschriebenen Eigenschaften ist es möglich, einzelne Roboterteile oder alleinstehende Testaufbauten in Betrieb zu nehmen. Die bis hier in dem Kapitel beschriebenen Umsetzungen sind in Zusammenarbeit mit Christian Thiele vom NRL entstanden.

4. Firmware

In *State 1*, der zweite Zeitabschnitt in Abbildung 4.1 (oben), werden von der Firmware zuerst die eventuell an das *AccelBoard3D* angeschlossenen Motoren angesprochen, in der Grafik wird dies mit „m“ gekennzeichnet. Es werden dabei aktuelle Werte für die Drehmomente übermittelt. Im Anschluss können Berechnungen von neuronalen Netzen durchgeführt oder Daten von einem Personal Computer empfangen werden. Diese Teile der Firmware werden nicht detaillierter betrachtet, da sie für diese Arbeit irrelevant sind.

State 2 und 3 sind die, im Hinblick auf die Sensorik, interessanten Abschnitte, sie sind mit „s“ gekennzeichnet. Die Aufspaltung in zwei separate Abschnitte ist durch das Abfragen der Motorenwinkel nötig. Die Kommunikation mit den Motoren ist die einzige bei der *Interrupts* auftreten. Die Antwortzeit der Dynamixel-Motoren ist leider nicht deterministisch, daher müssen sie mit einem größeren zeitlichen Abstand angesprochen werden. Abbildung 4.2 zeigt ei-

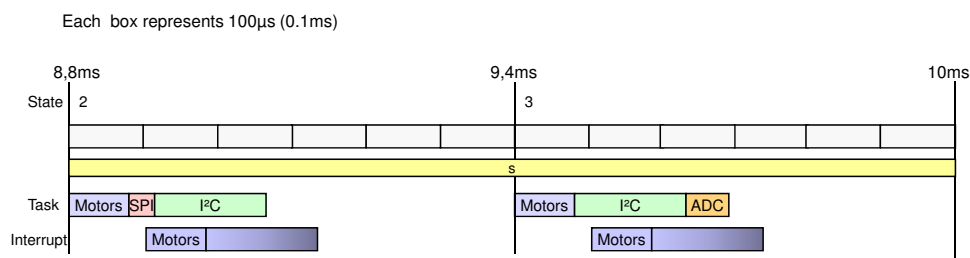


Abbildung 4.2.: Vergrößerung der *States 2 und 3* eines *AccelBoard3D*-Zeitslots.

ne Vergrößerung der beiden *States* und darüber hinaus genauere Angaben über die Reihenfolge und Dauer der einzelnen Schnittstellen-Kommunikationen, bei denen die Messwerte abgefragt werden. Detaillierte Angaben zum Timing und zur Initialisierungsphase werden im folgenden Abschnitt 4.2 für jede Schnittstelle bzw. für jeden Sensor eigenständig gemacht. Für die Gesamtdauer, im Bezug auf das Abfragen aller Sensoren, kann eine zweifache Sicherheit garantiert werden. Das heißt, dass das Abfragen der Sensoren fast doppelt so viel Zeit in Anspruch nehmen könnte, ohne den reibungslosen Betrieb zu gefährden.

4.2 Einbindung der Sensoren

In diesem Abschnitt werden die einzelnen Schnittstellen bzw. die Methoden zur Erfassung aller Sensorwerte ausführlich beschrieben. Dabei werden die Pro-

4. Firmware

tokolle der Schnittstellen kurz vorgestellt und Angaben zum Timing präzisiert. Des Weiteren wird der dazugehörige Programmcode in Auszügen vorgestellt.

Wie bereits in Abschnitt 2.3 erwähnt, werden die Messwerte der Sensoren in einem globalen Datenfeld gespeichert, welches über den *SpinalCord* übertragen wird. Intern besteht das Datenfeld aus Variablen des Typs *Short*. Folglich werden die Variablen durch 16 Bits im Zweierkomplement repräsentiert und haben einen Wertebereich von -32768 bis 32767 . Berechnungen zur Bewegungssteuerung, die die sensorischen Informationen benutzen, gehen von einer Festkommazahl aus, deren Wertebereich zwischen -1 und 1 liegt. Dementsprechend steht die ganze Zahl -32768 für die Festkommazahl -1 und die Zahl 32767 für annähernd 1 .

Die verwendeten Sensoren, mit Ausnahme des Accelerometers, liefern ganzzahlige, positive Messwerte in einer Auflösung unterhalb von 16 Bit. Damit ihre Werte den gesamten Bereich von -1 bis 1 abdecken, werden sie zuerst in eine Variable des Typs *Short* umgewandelt. Im Anschluss wird die Hälfte des Maximalwerts plus eins subtrahiert. Abschließend muss der Messwert nach links verschoben werden, bis er linksbündig innerhalb des 16 Bit-Wertes liegt. Die Messwerte des Accelerometers werden bereits in einer Zweierkomplement-Darstellung zwischen -512 und 511 abgebildet, sodass nur nach links verschoben werden muss.

4.2.1 Dynamixel-RX-28-Servomotor via RS485

Es soll nun erläutert werden, wie die Winkelmesswerte der bis zu vier Dynamixel-RX-28-Servomotoren abgefragt werden. Die vier Motoren sind auf zwei separate Anschlüsse aufgeteilt, den *DoubleStrand A und B*, vergleiche die Abschnitte 2.2 und 3.5. Der *STM32-Mikrocontroller* verfügt über keine eigene RS485-Schnittstelle. Die Motoren müssen daher wie beschrieben über einen *Transceiver* angesprochen werden. Die Daten werden dabei über eine USART-Schnittstelle des Mikrocontrollers an den *Transceiver* gesendet. Der *Transceiver* muss dafür vorher in den *Transmit*-Modus konfiguriert werden. Die Konfiguration geschieht dabei über zwei GPIO-Pins des Controllers. Sollen Daten versendet werden, müssen beide auf ein hohes Potenzial gezogen werden. Im Schaltplan A.2 werden die zwei Pins mit *DE* und \overline{RE} am jeweiligen *LTC2854-Transceiver* bezeichnet. Die seriellen Daten der USART-Schnittstelle 2 werden

4. Firmware

an den *DoubleStrand A*, die der USART-Schnittstelle 3 an den *DoubleStrand B* angelegt.

Um die Dynamixel-RX-28-Servomotoren korrekt ansprechen zu können, muss ein bestimmtes Protokoll eingehalten werden. Das Protokoll ist ausführlich im Datenblatt [RC08, S. 16] beschrieben. Die Struktur eines korrekten Instruktionpaketes ist in Abbildung 4.3 beispielhaft angegeben. Aus dem Instruktion-



Abbildung 4.3.: Struktur eines Instruktionpaketes zum Ansprechen eines Dynamixel-RX-28-Servomotors.

paket ist ersichtlich, dass jeder Motor eine eindeutige Nummer haben muss. Die Nummer ist in der Abbildung mit ID bezeichnet.

Wie in Abschnitt 4.1 angedeutet, ist das Abfragen der Winkelwerte eine der zeitkritischen Vorgänge. Deshalb werden einige Maßnahmen ergriffen, die den Vorgang beschleunigen. Die Existenz von zwei getrennten *DoubleStrands* ermöglicht es immer zwei Motoren zur gleichen Zeit abzufragen. Außerdem werden die zu sendenden Daten schon vorher in einem Buffer vorbereitet. Das Senden der Daten erfolgt dann jeweils im *State 2*, für einen der beiden Motoren pro *DoubleStrand*, und im *State 3* für die anderen Motoren, siehe Abbildung 4.2. Das Senden wird folgendermaßen begonnen: Ein Zeiger wird auf die richtige Stelle des vorgefüllten Buffers gerichtet und anschließend das erste Byte gesendet. Außerdem muss der entsprechende *Interrupt* aktiviert werden, der bei einem leeren *Transmit*-Register geworfen wird. Der Programmauszug 4.1 zeigt die dazugehörige *Interruptroutine*. Der vorgefüllte Buffer ermöglicht es die Zeit, die der *STM32-Mikrocontroller* zur Abarbeitung des *Interrupts* braucht, stark zu verkürzen. Außerdem ist zu sehen, dass die Daten für den *DoubleStrand B* absolut synchron in der *Interruptroutine* des *DoubleStrand A* versendet werden. Dies halbiert die benötigte Anzahl von *Interrupts*. Wurde das letzte Byte in das *Transmit*-Register kopiert, wird ein anderer *Interrupt* aktiviert. Dieser wird geworfen, wenn das letzte Byte vollständig gesendet wurde. Im Anschluss kann der *Transceiver* wieder in den *Receive*-Modus versetzt werden.

Das Senden aller Daten zur Abfrage des Winkels ist nach ca. 80 μ s abgeschlossen. Der Zeitstrahl in Abbildung 4.2 stellt dabei den ungünstigsten Fall

4. Firmware

Listing 4.1: Interruptroutine des USART 2.

```
1 /* *****  
2 TRANSMIT  
3 ***** */  
4 else if (USART2_SR_bit.TC) //transmission complete  
5 {  
6     USART2_CR1_bit.TCIE = 0; // disable Transmission complete interrupt  
7     USART2_SR_bit.TC = 0;  
8     //both strands receive mode  
9     DSA_DE = 0;  
10    DSA_REn = 0;  
11    DSB_DE = 0;  
12    DSB_REn = 0;  
13 }  
14 else if (USART2_SR_bit.TXE) //transmit register empty  
15 {  
16     USART2_SR_bit.TXE = 0; // clear interrupt pending flag  
17     if (DoubleStrandPointerA <= lastByteToSendA) //send next byte DSA  
18     {  
19         USART2_DR_bit.DR = DSATxBuffer[DoubleStrandPointerA++];  
20     }  
21     else //stop transmission, all bytes send  
22     {  
23         USART2_CR1_bit.TXEIE = 0; // disable transmit data register empty  
24         USART2_SR_bit.TXE = 0; // clear interrupt pending flag  
25         USART2_CR1_bit.TCIE = 1; // enable transmission complete interrupt  
26     }  
27     if (DoubleStrandPointerB <= lastByteToSendB) //send next byte DSB  
28     {  
29         USART3_DR_bit.DR = DSBTxBuffer[DoubleStrandPointerB++];  
30     }  
31 }
```

4. Firmware

dar, bei dem es zwischen den *Interrupts* des Sendens keine weitere Programmabarbeitung in der Hauptroutine geben würde. Tatsächlich wird schon früher mit der weiteren Abarbeitung fortgefahren, als es die Abbildung anzeigt.

Die Antwort wird von den Dynamixel-Servomotoren nach einer unbestimmten Zeit gesendet. Das Zeitfenster zwischen dem Abschluss der Anfrage und dem Beginn der Antwort ist zwischen $20\ \mu\text{s}$ und $170\ \mu\text{s}$ lang. Die Struktur der Antwort ist ähnlich zum Aufbau der Anfrage, enthält folglich auch eine Checksumme, die berechnet werden muss. Wird ein Byte auf einem der *DoubleStrands* empfangen, wird wieder ein *Interrupt* geworfen, mit dem Unterschied, dass es für jeden *DoubleStrand* separat geschieht. Die *Interrupt*-Routine kopiert alle empfangenen Bytes blind in einen Empfangsbuffer. Ist das letzte Byte empfangen, welches an der Anzahl der bisher empfangenen Bytes festgestellt werden kann, wird die Checksumme berechnet und mit der übertragenen verglichen. Wenn sie übereinstimmen, wird noch innerhalb der *Interrupt*-Routine der Winkelwert in das richtige Format umgewandelt und an die richtige Stelle im globalen Datenfeld geschrieben, das über den *SpinalCord* versendet wird.

4.2.2 AD7150, ADC121C027 via I²C

Der *Inter-Integrated Circuit*, kurz I²C, ist ein serieller Datenbus. Beim M-Serie-Roboter werden über die I²C-Schnittstelle sowohl der IC *AD7150*, als auch der IC *ADC121C027* angesprochen. Die ICs messen verschiedene physikalische Größen, genaue Angaben, wo diese ICs im Roboter eingesetzt werden, werden im Kapitel 3 gemacht. Am *AccelBoard3D* sind insgesamt zwei I²C-Anschlüsse vorhanden, vergleiche Abschnitt 2.2. Der *STM32-Mikrocontroller* bietet zwar eine hardwareseitige Unterstützung zur Serialisierung der Daten von bis zu zwei unabhängigen I²C-Schnittstellen, jedoch traten bei der Benutzung dieser verschiedene Probleme auf. Daher wurde eine softwareseitige Umsetzung über GPIO-Pins realisiert, dies wird auch *bit banging* genannt. Die umgesetzten Methoden wurden in einer eigenen C-Datei gekapselt und bieten eine ausreichende Funktionalität für unsere Zwecke. Des Weiteren konnte so die exakt gleichzeitige Übertragung von Daten über die beiden I²C-Schnittstellen ermöglicht werden.

Ein Beispiel einer I²C-Datenübertragung ist in der Darstellung 4.4 abgebildet. Es gibt zwei separate Datenleitungen, die Daten- und die Taktleitung. Die Taktrate beträgt 400 kHz. Die Datenübertragung über einen I²C-Bus erfolgt

4. Firmware

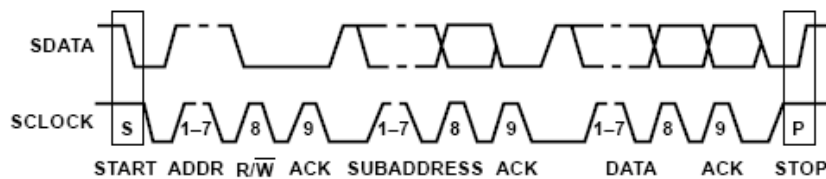


Abbildung 4.4.: Datentransfer über einen I²C-Bus. Quelle: [AD07]

immer nach dem selben Schema. Datenübertragungen werden vom *Master* initiiert, in unserem Fall ist dies der *STM32-Mikrocontroller*. Die angeschlossenen ICs sind immer die *Slaves*, zu denen Daten gesendet oder abgerufen werden. Der Takt wird nur vom *Master* erzeugt, allerdings könnten die *Slaves* ein *Clock Stretching* durchführen. Dabei halten sie die Taktleitung auf einem hohen Potenzial und der *Master* muss die Übertragung pausieren. Ein *Clock Stretching* kann notwendig sein, falls der *Master* Daten abfragt, die zum Zeitpunkt der Datenübertragung noch nicht zur Verfügung stehen oder der *Slave* anderweitig beschäftigt ist. Die an der I²C-Schnittstelle angeschlossenen ICs erfordern keine Unterstützung des *Clock Stretchings*, sodass dessen Implementierung entfallen konnte. Der Beginn eines Transfers wird vom *Master* durch die Erzeugung einer Startkondition eingeleitet, siehe Abbildung 4.4. Danach wird ein Byte vom *Master* versendet, bei dem die ersten sieben Bit die Adresse des *Slaves* sind und das letzte Bit angibt, ob im Folgenden gelesen oder geschrieben werden soll. Bei allen Übertragungen wird durch ein, sich am Byte anschließendes, Bestätigungsbit signalisiert, ob die Übertragung erfolgreich war oder nicht. Jeder am I²C-Bus angeschlossene IC muss eine einzigartige Adresse besitzen, anderenfalls kann es zu Datenkollisionen kommen. Beim Schreiben sendet der *Master* ein zweites Byte, die Subadresse. Diese gibt an auf welchen Wert der interne Registerzeiger des *Slaves* gesetzt werden soll. Weitere Datenbytes können folgen, die dann, ab der vom Registerzeiger angegebenen Stelle, in die Register des *Slaves* geschrieben werden. Die Beendigung des Datentransfers wird vom *Master* mit einer negativen Bestätigung und einer Stopkondition vollzogen. Beim Lesen werden vom *Slave* die Daten, ausgehend von der aktuellen Position des internen Registerzeigers, übertragen. Das heißt, dass der Zeiger vorher zur Spezifizierung mit einer Schreibsequenz gesetzt werden sollte.

Der *AD7150* muss in der Initialisierungsphase des *AccelBoard3D*, seinen Aufgaben entsprechend, initialisiert werden. Die zugehörige Routine ist im Listing 4.2 aufgeführt. Damit der interne Registerzeiger des *AD7150-ICs*

4. Firmware

Listing 4.2: Initialisierung des AD7150-ICs.

```
1 char AD7150_Registers[10];
2 // prefill sensor registers
3 AD7150_Registers[0] = 0x08; //Ch1 Sensitivity
4 AD7150_Registers[1] = 0x86; //Ch1 Timeout
5 AD7150_Registers[2] = 0x0B; //Ch1 Setup
6 AD7150_Registers[3] = 0x08; //Ch2 Sensitivity
7 AD7150_Registers[4] = 0x86; //Ch2 Timeout
8 AD7150_Registers[5] = 0x0B; //Ch2 Setup
9 AD7150_Registers[6] = 0x99; //Configuration
10 AD7150_Registers[7] = 0x00; //Power-Down Timer
11 AD7150_Registers[8] = 0x00; //Ch1 CAPDAC
12 AD7150_Registers[9] = 0x00; //Ch2 CAPDAC
13
14 if (i2cDevices [2]. device == 3) //angle sensor
15 {
16     AD7150_Registers[2] = 0xCB; //Ch1 Setup
17     AD7150_Registers[5] = 0xCB; //Ch2 Setup
18     AD7150_Registers[8] = 0x84; //Ch1 CAPDAC
19     AD7150_Registers[9] = 0x84; //Ch2 CAPDAC
20     I2C1_SendBytes(&AD7150_Registers[0], i2cDevices[2].address, 10, 0x09);
21 }
```

nicht mehrmals gesetzt werden muss, wird ein zusammenhängender Block, im Bezug auf die Registeradresse, zur Konfiguration gesendet. Dementsprechend weichen nicht alle übertragenen Registerwerte von ihren Standardwerten ab. Im *Configuration*-Register muss der *adaptive threshold mode* deaktiviert und der *continues conversion mode* aktiviert werden. Dadurch kann ein absoluter Wert gemessen werden, ohne dass eine automatische Anpassung des Offsets vom *AD7150* vorgenommen wird. Je nachdem, ob der *AD7150* im kapazitiven Winkelsensor oder zur Messung der Kapazität eines Faraday-Kraftsensors eingesetzt wird, müssen die Register *Channel Setup* und *Channel Capdac*, für den jeweiligen Kanal, modifiziert werden. Das *Channel Setup*-Register dient zur Anpassung der Spanne des Messwertbereichs. Der Wert des *Channel Capdac*-Registers gibt den festen Offset an. Für detailliertere Informationen zu den Registern des *AD7150* wird auf das Datenblatt [AD07, S. 14ff] verwiesen.

Die Standardkonfiguration des ADC121C027-IC muss nicht verändert werden. Er befindet sich bereits in einem für unsere Zwecke optimal konfigurierten Zustand.

In der Standardbetriebsphase des *AccelBoard3D* werden alle, an beiden I²C-Bus angeschlossenen, ICs nach ihren aktuellen Messwerten abgefragt. Dieser

4. Firmware

Vorgang geschieht in den *States 2 und 3*, nachdem die Winkel der Servomotoren angefragt und die Beschleunigungsdaten des Accelerometers übertragen wurden. Beim *AD7150* müssen insgesamt vier Byte an Messdaten übertragen werden, jeweils zwei pro gemessener Kapazität. Die Daten werden in den Registern eins bis vier zur Verfügung gestellt. Um ein zeitintensives Setzen des internen Registerzeigers zu vermeiden, wird das erste Byte im Register null mitübertragen. Dies funktioniert, da der Zeiger nach jeder Datenübertragung automatisch auf den Wert null zurückgesetzt wird. Es müssen folglich insgesamt sechs Bytes übertragen werden, ein *Adressbyte* und fünf Datenbytes, bei denen jeweils noch ein Bestätigungsbit hinzukommt. Bei einer vorliegenden Taktfrequenz von 400 kHz ergibt sich eine theoretische Übertragungsdauer von 135 μ s. Die gemessene maximale Übertragungsdauer war ca. 150 μ s lang. Die zeitliche Differenz ist durch *Interrupts* zu erklären, die von den Antworten der Dynamixel-RX-28-Servomotoren ausgelöst werden, vergleiche Abbildung 4.2.

Der *ADC121C027* misst nur eine analoge Spannung mit einer Auflösung von 12 Bit, sodass nur zwei Bytes an Nutzdaten übertragen werden müssen. Das dazugehörige 16-Bit-Register ist idealerweise das Register null, daher kann wieder auf das Setzen des internen Registerzeigers verzichtet werden. Der Übertragungsumfang ist infolgedessen drei Bytes groß und die Übertragungsdauer nur halb so lang wie beim *AD7150*.

In allen Fällen, wird direkt nach der Übertragung der Messwertdaten, der passend umgerechnete Wert in die jeweilige Stelle des globalen Datenfeldes geschrieben.

4.2.3 Accelerometer *MMA7455L* via SPI

Das *Serial Peripheral Interface*, kurz SPI, ist wie der I²C-Bus ein serieller Datenbus. Im Unterschied zum I²C werden aber oft getrennte Datenleitungen für die Hin- und Rückrichtung der Kommunikation verwendet. Diese werden MISO für *Master In / Slave Out* und MOSI für *Master Out / Slave In* genannt. Des Weiteren wird eine Taktleitung und eine *Chipsselect*-Leitung benötigt. Die *Chipsselect*-Leitung bestimmt, welcher *Slave* am Bus aktiv sein soll. Das heißt, dass für jeden am SPI-Bus teilnehmenden *Slave* eine *Chipsselect*-Leitung vorhanden sein muss. Auf dem *AccelBoard3D* wird nur ein Chip angesprochen, sodass nur eine *Chipsselect*-Leitung vorhanden ist. Die Taktrate wird wie beim I²C vom *Master* erzeugt.

4. Firmware

Für die Datenübertragung über die SPI-Schnittstelle wird die in Hardware realisierte Schnittstelle des *STM32-Mikrocontroller* benutzt. Diese muss zu Beginn konfiguriert werden und ist anschließend unkompliziert ansprechbar. Die Konfiguration umfasst unter anderem das Einstellen der Taktrate und des SPI-Modus, sowie die Festlegung, ob im *Master*- oder *Slave*-Modus gearbeitet werden soll. Das Accelerometer *MMA7455L* kann mit einer Taktrate von bis zu 8 MHz angesprochen werden, deshalb wurde am Mikrocontroller eine Rate von 4,5 MHz eingestellt. Die nächsthöhere Taktrate wäre mit 9 MHz zu schnell für das Accelerometer gewesen. Das *MMA7455L* fungiert immer als *Slave*, sodass beim *STM32* der *Master*-Modus aktiviert wurde.

Beim SPI entfallen Start- oder Stopkonditionen, da durch die *Chipselect*-Leitung eine Synchronisation zu jeder Zeit hergestellt werden kann. Des Weiteren wird auch kein Bestätigungsbit abgefragt. Das heißt, dass ein Datenbyte bei der Übertragung über die SPI-Schnittstelle auch nur acht Bit groß ist. Abbildung 4.5 zeigt den Lesevorgang eines Bytes vom Accelerometer. Der Da-

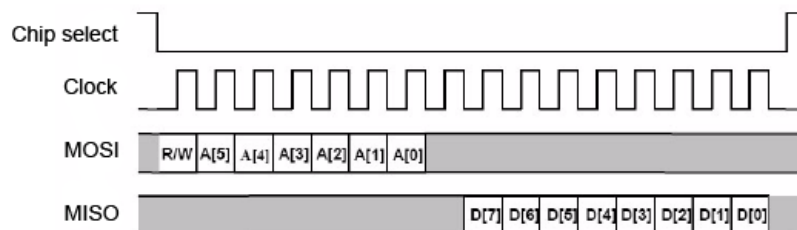


Abbildung 4.5.: SPI-Bus-Datentransfer des Accelerometers *MMA7455L*. Quelle: [Fre09]

tenntansfer wird dabei durch das Herunterziehen der *Chipselect*-Leitung auf das Massepotenzial initiiert. Im Anschluss sendet der *Master* ein Byte, indem die Daten seriell angelegt werden und ein Takt auf der Taktleitung erzeugt wird. Dieses Byte muss laut den Vorgaben des *MMA7455L* beim Lesevorgang folgendermaßen zusammengesetzt sein: Mit dem ersten Bit wird der Lesevorgang signalisiert; die sechs folgenden Bits geben die interne Registeradresse an, aus dessen Register die Daten gelesen werden sollen; der Wert des achten Bits wird nicht beachtet. Damit der *Slave* antworten kann, muss der *Master* weiterhin Takte auf der entsprechenden Leitung generieren. Im Falle des *MMA7455L* erfolgt die Antwort direkt mit dem nächsten Byte. Bei anderen ICs müssen unter Umständen mehrere *Dummybytes* gesendet werden, bis eine Antwort erfolgt.

4. Firmware

Listing 4.3: Abfrage der Beschleunigungswerte via SPI.

```
1 SPI1_CS = 0; //activate accelerometer via chip select
2 // read AccelData
3 SPI1_ReadByte(0x00);
4 SpinalCord[OwnSpinalCordPosition + 3] = SPI1_ReadByte(0x00) << 6; //AccelX low
5 SPI1_ReadByte(0x02);
6 SpinalCord[OwnSpinalCordPosition + 3] += SPI1_ReadByte(0x00) << 14; //AccelX
  high
7 SPI1_ReadByte(0x04);
8 SpinalCord[OwnSpinalCordPosition + 4] = SPI1_ReadByte(0x00) << 6; //AccelY low
9 SPI1_ReadByte(0x06);
10 SpinalCord[OwnSpinalCordPosition + 4] += SPI1_ReadByte(0x00) << 14; //AccelY
  high
11 SPI1_ReadByte(0x08);
12 SpinalCord[OwnSpinalCordPosition + 5] = SPI1_ReadByte(0x00) << 6; //AccelZ low
13 SPI1_ReadByte(0x0A);
14 SpinalCord[OwnSpinalCordPosition + 5] += SPI1_ReadByte(0x00) << 14; //AccelZ
  high
15 wait(100); //wait to deactivate SPI_device signal (SPI1_CS)
16 SPI1_CS = 1; //deactivate accelerometer via chip select
```

Das Schreiben in ein Register des Accelerometers erfolgt wie das Lesen. Es muss nur durch das erste Bit des ersten Bytes festgelegt werden. Die sechsstellige Adresse gibt dann das Register an, in das geschrieben werden soll. Das zweite Byte wird dann ebenfalls vom *Master* gesendet und vom *Slave* ins entsprechende Register übernommen. Für alle weiteren Bytes muss wieder zuerst die interne Adresse des Registers gesendet werden. Dies gilt sowohl beim Schreiben, als auch beim Lesen.

Während der Initialisierungsphase des *AccelBoard3D* muss das *MMA7455L* konfiguriert werden. Zum einen müssen die Kalibrierungsdaten, deren Ermittlung im Abschnitt 4.3 beschrieben wird, übermittelt werden und zum anderen muss der *Measurement*-Mode aktiviert werden.

Es müssen die Beschleunigungswerte aller drei räumlichen Achsen während der Standardbetriebsphase des *AccelBoard3D* über den SPI-Bus übertragen werden. Die Abfrage findet während des *State 2* statt, vergleiche Abbildung 4.2. Listing 4.3 zeigt den dazugehörigen Programmcode. Wie beschrieben, wird die SPI-Schnittstelle des Accelerometers durch die *Chipsselect*-Leitung für den Datenverkehr aktiviert. Im Anschluss werden nacheinander alle Beschleunigungsdaten abgefragt und im richtigen Format in das globale Datenfeld geschrieben. Es müssen insgesamt sechs Daten- und sechs Adressbytes übertra-

4. Firmware

gen werden. Mit einer Taktfrequenz von 4,5 MHz ergibt sich eine theoretische Übertragungsdauer von 21,33 μ s. Die maximal gemessene Dauer betrug 32 μ s.

4.2.4 Interner ADC des *STM32-Mikrocontrollers*

Zum besseren Verständnis sind im Referenzhandbuch [STM09, S. 198ff] detaillierte Beschreibungen zum ADC zu finden, an dieser Stelle soll nur kurz auf die in der Firmware genutzten Funktionalitäten des ADC eingegangen werden. Der interne 12 Bit-ADC des Mikrocontrollers muss drei verschiedene analoge Signale messen. Die Bedeutungen dieser sind ausführlich im Abschnitt 3.6 erläutert. Jedem dieser Signale ist ein bestimmter Kanal des A/D-Wandlers zugeordnet. Die Kanäle sind in einer Gruppe zusammengefasst und werden dadurch nacheinander vom ADC umgewandelt. Dazu muss im *STM32-Mikrocontroller* der *SCAN*-Modus aktiviert werden. Eine fortwährende zyklische Umwandlung der Kanäle wird durch die Benutzung des *Continuous-Conversion*-Modus erreicht. Die Modi werden in der Initialisierungsphase des *AccelBoard3D* festgelegt. Zusätzlich wird der ADC in der Initialisierungsphase kalibriert. Dies geschieht über einen internen Selbstkalibrierungsmechanismus, der durch das Einschalten des *Calibration*-Modus aktiviert wird. Nach einer gewissen Zeit wird durch Setzen eines *Flags* das Ende der Kalibrierung signalisiert und der reguläre Betrieb des ADC kann aufgenommen werden. Dafür muss allerdings noch die Abtastzeit für alle Kanäle festgelegt werden. Um einen möglichst genauen Messwert zu erlangen, wird die Abtastzeit auf das Maximum von 239,5 Taktzyklen eingestellt. Bei einer Taktfrequenz von 12 MHz des ADCs, ergibt sich daraus eine Dauer von ca. 20 μ s pro Kanal.

Die Messwerte lassen sich weiter entstören, indem zusätzlich über mehrere Abtastwerte gemittelt wird. Da nur alle 10 ms der entsprechende Wert über den *SpinalCord* übertragen werden muss, kann über eine sehr große Anzahl von Abtastwerten gemittelt werden. In der Firmware des M-Serie-Roboters wird zurzeit über 128 Messwerte pro Kanal gemittelt. Bei drei Kanälen ergibt sich daraus eine Dauer von 7,7 ms, sodass immer mit aktuellen Werten pro M-Serie-Zeitslot, der 10 ms lang ist, gerechnet werden kann.

Nach jeder Messung des ADCs wird der gemessene Wert in ein Register geschrieben. Der Messwert muss daraus gelesen und an anderer Stelle zwischengespeichert werden, damit die nächste Messung das vorherige Ergebnis nicht überschreibt. Die einfachste Methode dazu ist, dass nach jeder Messung

4. Firmware

Listing 4.4: Mittelung der A/D-Wandler-Werte.

```
1 while(counter < ADC_totalConversions) { // analyse data
2   DSA_Cur += ADCConvertedValues[counter++];
3   DSB_Cur += ADCConvertedValues[counter++];
4   UDS += ADCConvertedValues[counter++];
5 }
6 DSA_Cur /= ADC_Conversions;
7 DSB_Cur /= ADC_Conversions;
8 UDS /= ADC_Conversions;
```

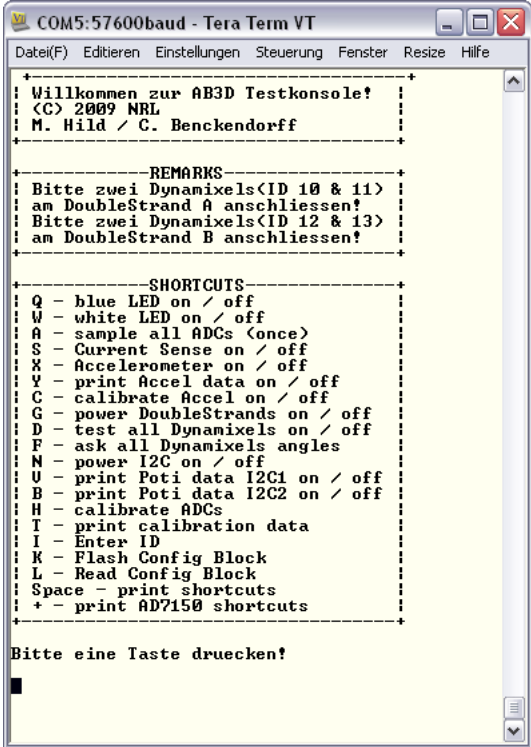
ein *Interrupt* geworfen wird, in dem der Messwert verschoben wird. Jedoch würde diese Herangehensweise das gesamte Timing der M-Serie-Firmware gefährden, da der *Interrupt* alle 20 μ s auftreten würde und zusätzliche Verzögerungen durch das Hinein- und Hinausspringen aus der *Interrupt*-Routine entstehen. Die Lösung dieses Problems ist die Benutzung eines *Direct-Memory-Accesses*, kurz DMA, der das Verschieben der Daten erledigt und ohne CPU-Nutzung durchgeführt wird. Die Initialisierung eines DMAs wird auf dem *STM32-Mikrocontroller* folgendermaßen durchgeführt: Es müssen die Adressen und das Datenformat des Quellregisters und des Zielspeichers angegeben werden. Weiterhin muss die Anzahl der Datenverschiebungen angegeben und in unserem Fall der *Circular*-Modus aktiviert werden. Bei diesem Modus wird der Zeiger, der die aktuelle Speicheradresse des Zielwerts angibt, nach der vollständigen Durchführung aller festgelegten Datenverschiebungen, auf seinen Ausgangswert zurückgesetzt. Daraus folgt, dass immer der älteste Messwert überschrieben wird und alle restlichen Werte zeitlich zusammenhängend sind.

Der Hauptprozessor kann jederzeit auf das vom DMA gefüllte Datenfeld zugreifen und weitere Berechnungen durchführen. Bei der M-Serie-Firmware geschieht das im *State 3* nach der Kommunikation über die I²C-Schnittstellen, vergleiche Abbildung 4.2. Es wird für jeden der ADC-Kanäle ein gemittelter Messwert berechnet und mit den entsprechenden Kalibrierungsdaten, deren Erhebung im Abschnitt 4.3 beschrieben ist, verrechnet. Anschließend werden die Messwerte im richtigen Format in das globale Datenfeld geschrieben, das über den *SpinalCord* versendet wird. Der Programmcode zur Mittelung der ADC-Werte ist im Listing 4.4 aufgeführt. Die Abarbeitung des Programmcodes pro M-Serie-Zeitslot, der die ADC-Auswertung betrifft, dauert maximal 65 μ s.

4. Firmware

4.3 Kalibrierung mittels *AB3D-Testkonsole*

Die *AB3D-Testkonsole* ist ein Testprogramm, das alternativ zur Firmware auf ein *AccelBoard3D* programmiert werden kann. Es dient dabei sowohl zum Test fast aller Bauteile, als auch zur Kalibrierung des Accelerometers und der Spannungs- und Strommessvorrichtung. Der Name *AB3D-Testkonsole* leitet sich von seiner Bedienung ab, es kann mit jedem beliebigen *Terminal*-Programm über die serielle Schnittstelle gesteuert werden. Eine Liste der möglichen Befehle ist in der Abbildung 4.6 unter dem Menüpunkt *Shortcuts* ersichtlich. Die für die Kalibrierung wichtigen Befehle haben die Abkürzung „C“ und



```
COM5:57600baud - Tera Term VT
Datei(F) Editieren Einstellungen Steuerung Fenster Resize Hilfe
+-----+
| Willkommen zur AB3D Testkonsole! |
| <C> 2009 NRL |
| M. Hild / C. Benckendorff |
+-----+
+-----REMARKS-----+
| Bitte zwei Dynamixels<ID 10 & 11> |
| an DoubleStrand A anschliessen! |
| Bitte zwei Dynamixels<ID 12 & 13> |
| an DoubleStrand B anschliessen! |
+-----+
+-----SHORTCUTS-----+
| Q - blue LED on / off |
| W - white LED on / off |
| A - sample all ADCs (once) |
| S - Current Sense on / off |
| X - Accelerometer on / off |
| V - print Accel data on / off |
| C - calibrate accel on / off |
| G - power DoubleStrands on / off |
| D - test all Dynamixels on / off |
| F - ask all Dynamixels angles |
| N - power I2C on / off |
| U - print Poti data I2C1 on / off |
| B - print Poti data I2C2 on / off |
| H - calibrate ADCs |
| T - print calibration data |
| I - Enter ID |
| K - Flash Config Block |
| L - Read Config Block |
| Space - print shortcuts |
| * - print AD7150 shortcuts |
+-----+
Bitte eine Taste druecken!
█
```

Abbildung 4.6.: Startmenü der *AB3D-Testkonsole* mit Auflistung der Befehle.

„H“. Mit ihnen können Kalibrierungsdaten für das Accelerometer, bzw. für die Spannungs- und Strommessung, erhoben werden. Die Daten werden direkt im *Flash*-Speicher des *STM32-Mikrocontrollers* abgelegt und bleiben dort permanent erhalten. Auch das Aufspielen der Firmware verändert die Daten nicht. Die Speicheradressen der Kalibrierungsdaten haben einen festen Wert, sodass die Firmware in der Lage ist, die Daten auszulesen und weiter zu verarbeiten.

4. Firmware

Das Accelerometer ist vom Werk aus kalibriert, allerdings wurde dafür eine Versorgungsspannung in einer Höhe von 2,8 V verwendet und damit nicht die auf dem *AccelBoard3D* anliegende Spannung von 3,3 V. Es muss folglich mit der veränderten Versorgungsspannung erneut kalibriert werden. Dazu muss das *AccelBoard3D* mit der Unterseite nach unten so fixiert werden, dass die Gravitation lotrecht zur Platine des *AccelBoard3D* wirkt. In dieser Stellung sollte das Accelerometer eine Beschleunigung von 0 g in der X- und Y-Achse und eine Beschleunigung von ein 1 g in der Z-Achse messen. Sind die aktuellen Messwerte von diesem Ideal abweichend, können in den Offset-Registern des Accelerometers entsprechende Werte zum Ausgleich der Abweichungen geschrieben werden. Das Testkonsolen-Programm verändert die Offsetwerte in kleinen Schritten, bis vom Benutzer entschieden wird, dass die aktuellen Offsetwerte den Fehler der Beschleunigungswerte minimieren. Die gewonnenen Kalibrierungsdaten werden nicht dauerhaft in den Registern des Accelerometers gespeichert und müssen deshalb, wie beschrieben, im *Flash*-Speicher des Mikrocontrollers abgelegt werden.

Für die Kalibrierung der Spannungs- und Strommessung der beiden *DoubleStrands* muss das *AccelBoard3D* mit einer Referenzspannung versorgt werden, die bei der Kalibrierung unterschiedlicher *AccelBoard3D* identisch ist. Des Weiteren wird ein Referenzlastwiderstand benötigt. Während der Kalibrierung werden die Ströme zuerst im unbelasteten Zustand gemessen. Anschließend wird der Lastwiderstand abwechselnd am *DoubleStrand A* und *B* angeschlossen und die jeweils gemessenen Ströme gespeichert. Ferner werden zwei Messwerte der an den *DoubleStrands* anliegenden Spannung erhoben, einer bei eingeschalteten *DoubleStrands* und einer im abgeschalteten Zustand. Die Spannung der *DoubleStrands* kann über einen GPIO-Pin (PC13) des Mikrocontrollers durch einen MOSFET-Transistor deaktiviert werden, vergleiche Abbildung A.1. Es wurden folglich sechs Messwerte erhoben, die wieder im *Flash*-Speicher hinterlegt werden. Die Messwerte werden dann in der Firmware bei der ADC-Messung genutzt, sodass alle *AccelBoard3D* vergleichbare Spannungen und Ströme messen.

5 Experimente und Resultate

Zahlreiche Testergebnisse der vorgestellten Sensoren werden im nachfolgenden Kapitel gezeigt und bewertet. Die Sensoren befanden sich dabei innerhalb des M-Serie-Roboters. Im Einzelnen wurde der kapazitive Winkelsensor, der Winkelsensor mittels Potentiometer, das Accelerometer und die Kraftsensoren getestet. Dazu wurden möglichst praxisnahe und für die Bewegungssteuerung relevante Testszenarien gewählt.

5.1 Test des kapazitiven Winkelsensors

Der folgende Abschnitt soll Testdaten vorstellen, die mit dem kapazitiven Winkelsensor erhoben wurden. Die dafür nötigen Experimente wurden mit einem Sensor, der bereits in der ersten Version des M-Serie-Roboters an mehreren Gelenken verbaut wurde, durchgeführt. Der getestete Sensor befindet sich im Gelenk der rechten Schulter. Zur Aufnahme der Daten wurde das Programm BrainDesigner benutzt, welches im Rahmen des Projekts ALEAR entwickelt wurde, siehe [Thi10]. Es erlaubt unter anderem die neuronale Ansteuerung des gesamten M-Serie-Roboters, aber auch das einfache Mitschreiben der Sensorwerte. Wie im Abschnitt 2.3 beschrieben, sind alle Werte, das heißt auch alle Sensorwerte, des Roboters zwischen $[-1, 1]$ normiert. Die im folgendem Abschnitt beschriebenen Daten sind daher einheitenlos.

Zunächst wurde ein einfacher Test durchgeführt. Dazu wurde die Schulter des humanoiden Roboters manuell von außen, konstant in eine Richtung bewegt. Die gemessenen Daten sind in der Abbildung 5.1 illustriert. Zusätzlich zu den Daten des kapazitiven Winkelsensors wurden gleichzeitig die Winkeldaten eines Dynamixel-RX-28-Servomotors aufgenommen. Der Servomotor treibt dasselbe Schultergelenk an und nimmt daher die gleichen Winkelwerte mit einem internen Potentiometer auf. Genauere Angaben zu den Sensoren im Servomotor werden im Unterkapitel 3.5 gemacht. In Abbildung 5.1 wird der bereits erwähnte Zusammenhang zwischen den einzelnen Kapazitäten des Winkelsensors und des berechneten Winkels nochmals verdeutlicht, vergleiche Abschnitt 3.1.1. Darüber hinaus ist auch ein grober Zusammenhang zwischen den beiden Winkeln der unterschiedlichen Sensoren erkennbar. Jedoch können anhand dessen keine genauen Aussagen über die Qualitäten des kapazitiven Winkelsensors getroffen werden. Folglich wurde ein weiterer Test durchgeführt.

5. Experimente und Resultate

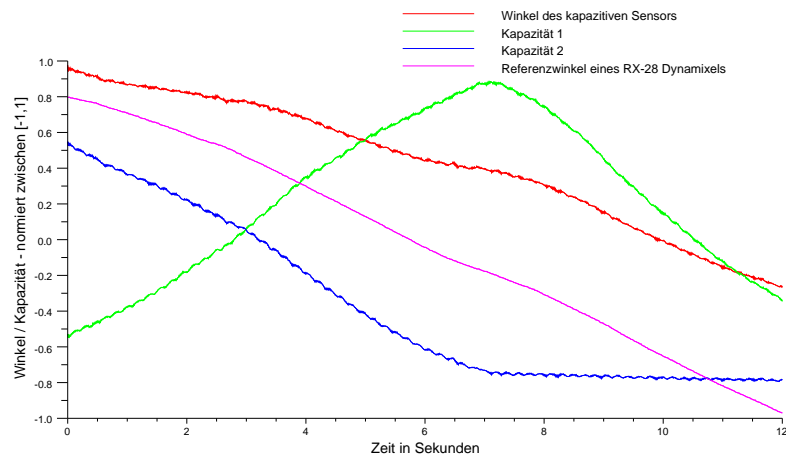


Abbildung 5.1.: Messwerte eines Tests des kapazitiven Winkelsensors und eines internen Winkelsensors eines Dynamixel-RX-28-Servomotors. Die zwei variierenden Kapazitäten sind in grün und blau abgetragen. Der daraus errechnete Winkel ist rot und der des Servos pink dargestellt.

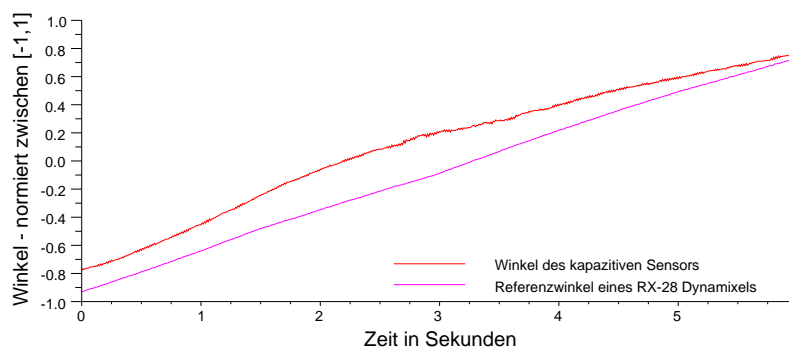


Abbildung 5.2.: Messwerte eines Tests des kapazitiven Winkelsensors und eines internen Winkelsensors eines Dynamixel-RX-28-Servomotors.

5. Experimente und Resultate

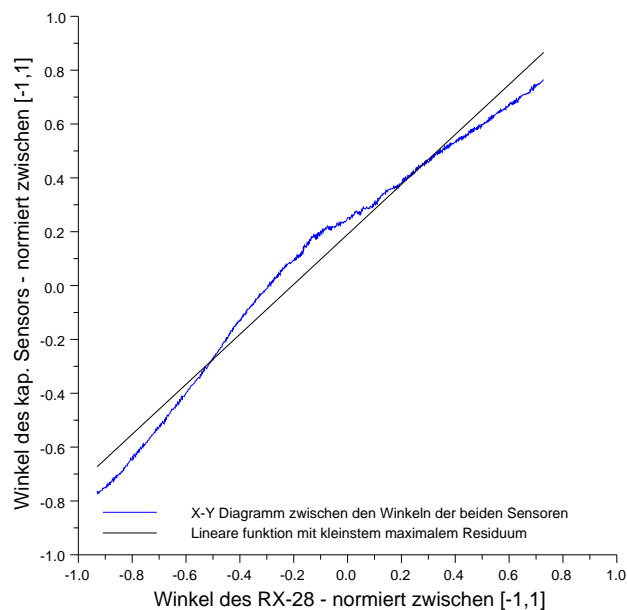


Abbildung 5.3.: X-Y-Diagramm zwischen dem Winkel des kapazitiven Sensors und dem Winkel des internen Potentiometers des Servomotors (blau). Linear approximierte Funktion mit kleinstem maximalem Residuum (schwarz).

Beim zweiten Test wurde der RX-28-Servomotor zur Variation des Drehwinkels eingesetzt, dadurch konnte eine gleichmäßigere Veränderung des Winkels erreicht werden. Die Messwerte dieses Tests sind in der Abbildung 5.2 über die Zeit abgetragen. Abbildung 5.3 zeigt das zugehörige XY-Diagramm, dafür wurden die Daten der beiden Messwertreihen gegeneinander eingezeichnet. Der entstehende Graph sollte möglichst linear sein und keine Unstetigkeiten aufweisen. Als Optimum ist eine approximierte lineare Funktion eingezeichnet. Sie ist so gewählt, dass das maximale Residuum möglichst minimal ist. Von dem maximalen Residuum lässt sich auf den zu erwartenden absoluten Fehler des kapazitiven Winkelsensors schließen, sofern keine weiteren Maßnahmen zur Verbesserung der Linearität durchgeführt werden. Mit den bisher beschriebenen Verfahren zur Berechnung des Winkels würde sich ein absoluter Fehler von bis zu 10° ergeben. Abweichungen in dieser Größenordnung wären für die Verwendung im Roboter nicht zweckmäßig.

Zur Verbesserung des absoluten Winkels wurde deshalb versucht den Zusammenhang zwischen dem Winkel des kapazitiven Sensors und dem des Referenzwinkel zu linearisieren. Dafür wurde eine Korrekturfunktion über Stützstellen errechnet. Diese Korrekturfunktion wird zum Winkel des kapazitiven

5. Experimente und Resultate

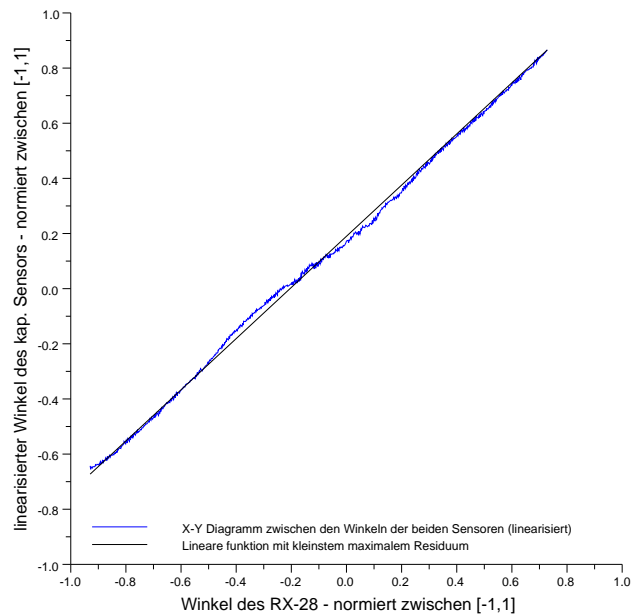


Abbildung 5.4.: X-Y-Diagramm zwischen dem linearisierten Winkel des kapazitiven Sensors und dem Winkel des internen Potentiometers des Servomotors (blau). Linear approximierte Funktion mit kleinstem maximalem Residuum (schwarz).

Sensors addiert und hat insgesamt drei äquidistante Stützstellen zwischen denen linear interpoliert wurde. Das Ergebnis dieser Korrekturmaßnahme ist in der Abbildung 5.4 ersichtlich. Der absolute Fehler konnte dadurch auf unter 5° gesenkt werden. Für die Benutzung in der M-Serie ist der Fehler weiterhin zu groß.

Während der praktischen Tests konnte ein maßgeblicher Grund für die Ungenauigkeit des kapazitiven Winkelsensors identifiziert werden. Der Abstand der beiden Messflächen zueinander ist stark von dem Anpressdruck des Gegenstücks abhängig. Abbildung 5.5 zeigt deutlich, dass schon minimale Abweichungen des Abstandes beider Messflächen von $20\ \mu\text{m}$ eine Winkelabweichung von über 10° zur Folge hätten. Grundlage für die Abbildung waren die Abmaße des kapazitiven Winkelsensors aus Abschnitt 3.1.1, mit denen eine theoretische maximale Kapazität von $7,206\ \text{pF}$ errechnet wurde. Ein gleichmäßiger, konstanter Anpressdruck, der keine Abweichungen des Abstandes im Mikrometerbereich zulässt, kann vom gegenwärtigen mechanischen Aufbau nicht gewährleistet werden. Eine Verbesserung des Aufbaus kann die Genauigkeit des kapazitiven Sensors deutlich erhöhen. Weitere Arbeiten müssen durchge-

5. Experimente und Resultate

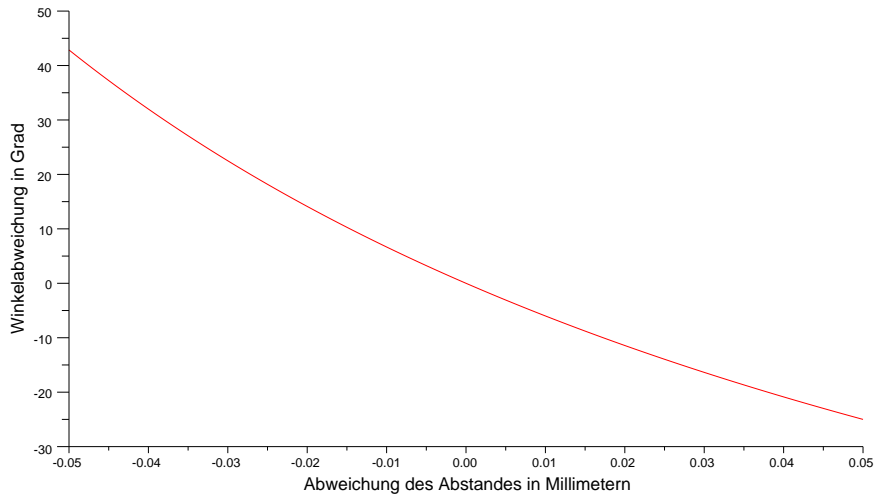


Abbildung 5.5.: Messabweichung des Winkels in Abhängigkeit von der Abstandsabweichung.

führt werden, um diese Annahme zu bestätigen. Im Rahmen der Diplomarbeit wurde stattdessen eine Alternative zum kapazitiven Winkelsensor gesucht. Die im Abschnitt 3.2 in Form des Winkelsensors mittels Potentiometer vorgestellt wurde.

5.2 Test des Winkelsensors mittels Potentiometer

Dieser Abschnitt widmet sich Experimenten, die zur Bestimmung der Funktionstüchtigkeit und Genauigkeit des Winkelsensors mittels Potentiometer gemacht wurden. Im ersten Test des Potentiometers, mit dazugehöriger Auswertungselektronik, war dieses bereits im Knöchelgelenk des humanoiden Roboters verbaut. Das genannte Gelenk ist für laterale Bewegungen zuständig und wird von einem RX-28-Servomotor angetrieben. Für den Test wurden mit Hilfe des BrainDesigners die Ausgangswerte eines $SO(2)$ -Oszillators zur Ansteuerung des *RX-28* benutzt. Eine sinusähnliche Schwingung des Roboterfußes war die Folge. Die von den Sensoren erfassten Daten sind im Diagramm der Abbildung 5.6 abgetragen. Der externe Winkelsensor mittels Potentiometer ist direkt mit der Achse des Gelenkes verbunden. Die von ihm gemessenen Werte sind rot dargestellt. Die pinke Kurve spiegelt die Messwerte vom internen Potentiometer des Dynamixel-Servomotors wider. Die Funktionstüchtigkeit des externen

5. Experimente und Resultate

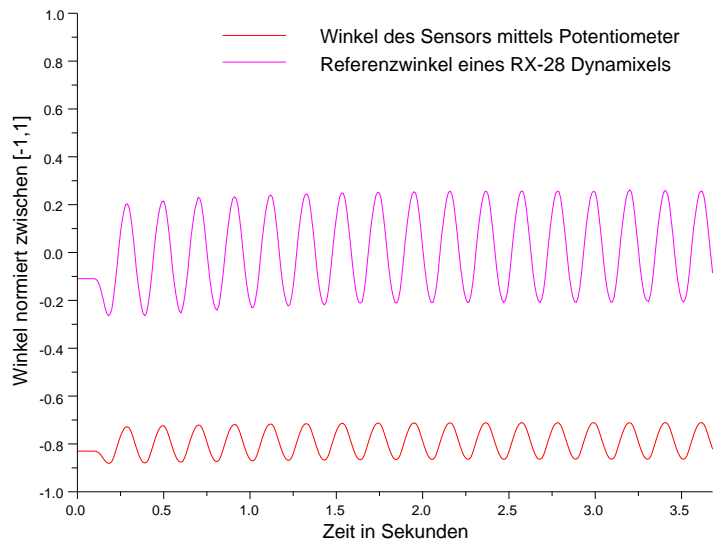


Abbildung 5.6.: Messwerte eines Tests des Winkelsensors mittels Potentiometer und eines internen Winkelsensors eines Dynamixel-RX-28-Servomotors.

Sensors ist auf Grund des fast identischen Kurvenverlaufs ersichtlich, der sich nur in der Amplitude und einem Versatz auf der Y-Achse unterscheidet. Eine genaue Abschätzung der Genauigkeit kann durch diesen Test aber nicht getätigt werden.

Dazu wurde ein weiterer Test vollzogen, bei dem ein hoch auflösender, optischer Winkelsensor als Referenz verwendet wurde, vergleiche [Hen08]. Durch ihn wird der Winkel über den gesamten Drehbereich von 360° mit einer Genauigkeit von 14 Bit abgebildet. Der optische Drehgeber befindet sich in einem Teststand. Für den zweiten Test wurde das Potentiometer in diesen Teststand integriert und mit der gleichen Abtriebswelle verbunden, an der auch der Referenzdrehgeber angebracht ist. Die Messwerte beider Sensoren sind im Bild 5.7 (oben) als Graph abgebildet. Beim Test wurde die Welle des Motorteststandes mehrmals hin und her bewegt, insgesamt wurden 1500 Messwerte über einen Zeitraum von 15 Sekunden aufgenommen. Das passende X-Y-Diagramm zwischen den beiden Winkeln befindet sich in der Abbildung 5.8. Es ist keine deutliche Nichtlinearität des Winkelsensors mittels Potentiometer sichtbar. Abbildung 5.7(unten) zeigt die Winkeldifferenz in Grad. Die Skala des letzten Diagramms wurde dabei deutlich zu der des ersten verändert, sodass die Dif-

5. Experimente und Resultate

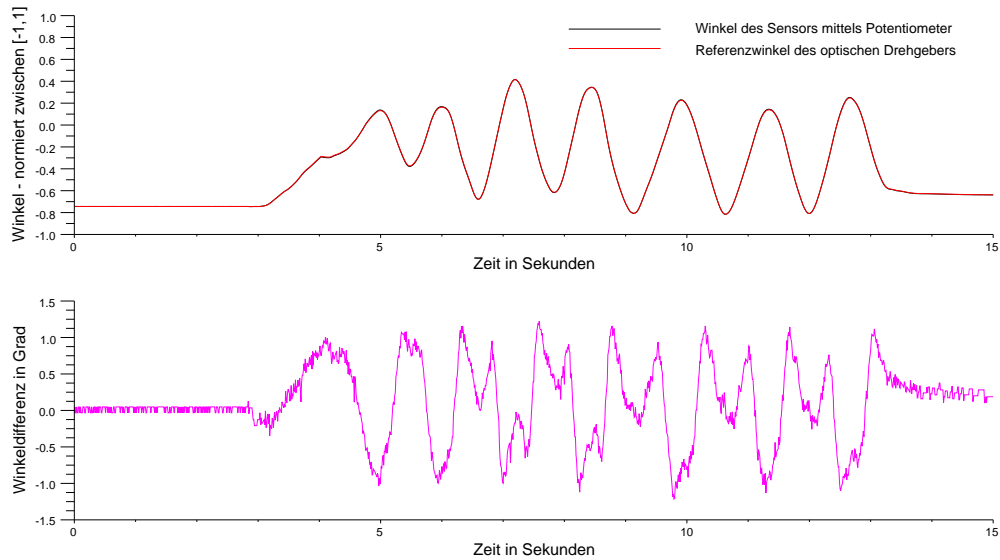


Abbildung 5.7.: Messwerte des Winkelsensors mittels Potentiometer und die des optischen Referenzdrehgebers (oben). Winkeldifferenz beider Sensoren über die Zeit in Grad mit veränderter Skala (unten).

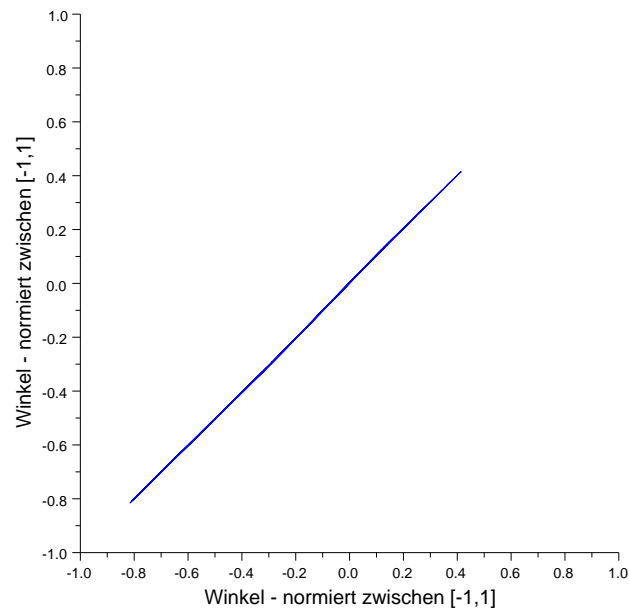


Abbildung 5.8.: X-Y-Diagramm zwischen dem Winkel des Potentiometers und dem des Drehgebers.

5. Experimente und Resultate

ferenz besser sichtbar ist. Während der ersten drei Sekunden, in denen keine Bewegung stattfindet, ist die Winkeldifferenz sehr gering. Die Standardabweichung beträgt dort nur ca. $0,039^\circ$. Sobald die Abtriebswelle bewegt wurde, erhöhte sich die Winkeldifferenz merklich. Mechanische Defizite im Aufbau des Motorteststandes könnten diesen Effekt deutlich verstärkt haben. Die Standardabweichung über den gesamten Messzeitraum erhöht sich damit auf ca. $0,516^\circ$. Die größte gemessene Differenz beträgt $1,225^\circ$. Dennoch sind die Ergebnisse beachtlich und deutlich besser als die des kapazitiven Winkelsensors, der im Abschnitt 3.1 vorgestellt wurde. Bezogen auf die Bewegungssteuerung eines humanoiden Roboters ist der Einsatz des Winkelsensors mittels Potentiometer von großem praktischen Nutzen, da eine sehr hohe Genauigkeit vorliegt.

5.3 Test des Accelerometers

Im Folgenden sollen Messergebnisse vorgestellt werden, die mit dem bisher verwendeten Accelerometer im finalen *AccelBoard3D* erhoben wurden. Dazu wurde das *AccelBoard3D* unterschiedlichen Bedingungen ausgesetzt, die nachfolgend als Phasen I bis VI bezeichnet werden. Jene Phasen wurden zeitlich nacheinander in einer einzigen Messung erzeugt. Die daraus resultierenden Beschleunigungsdaten sind in der Grafik 5.9 veranschaulicht. Es sind die Werte aller drei räumlichen Achsen in 10 ms Abständen über eine Zeit von zwölf Sekunden abgetragen. Die aus den äußeren Bedingungen resultierenden, differenzierenden Phasen sind am unteren Rand der Abbildung mit römischen Ziffern durchnummeriert und in der Bildunterschrift benannt worden.

Als Erstes wurde das AB3D in der X-Achse geneigt. Die Werte der Y-Achse ändern sich dabei wie erwartet nicht. In der zweiten Phase wurde das Board extremen Beschleunigungen ausgesetzt, indem es stark geschüttelt wurde. Dank des sehr großen Messwertbereiches von $\pm 8g$ werden die Daten nicht beschnitten und liefern immer gültige Werte. Die Phase III ist durch leichte Bewegungen der Hand und den damit einhergehenden kleinen Beschleunigungen von der absoluten Ruheposition der letzten Phase unterscheidbar. Der freie Fall in Phase IV ist durch die nicht vorhandene Beschleunigung in allen drei räumlichen Achsen, über mehrere Messschritte hinweg, festzustellen. Das *AccelBoard3D* fiel aus einer Höhe von ca. 1,5 m zu Boden. Der Aufprall hat Beschleunigungen von ca. 4 g zur Folge und ist im ersten Blick nicht von etwas leichterem Schütteln unterscheidbar. Vergleicht man allerdings den Ruck der in diesen

5. Experimente und Resultate

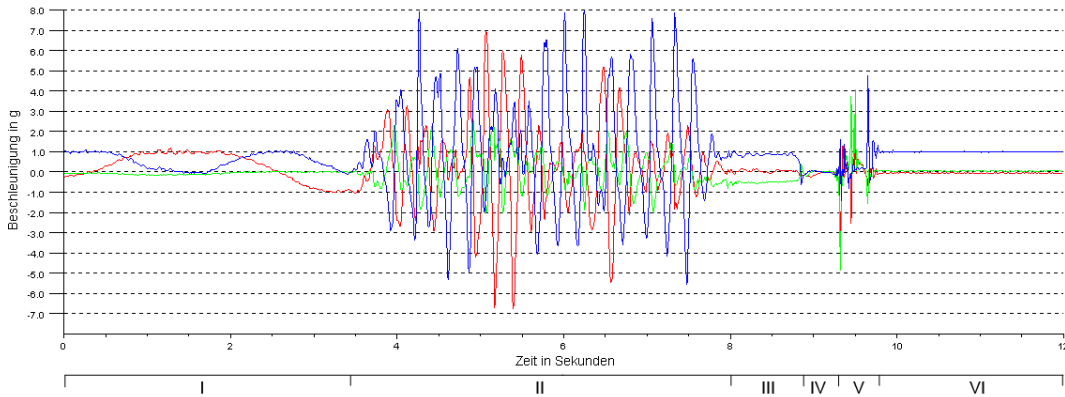


Abbildung 5.9.: Messwerte des Beschleunigungssensors *MMA7455L* im finalen *AccelBoard3D*. Es sind die drei Achsen separat abgetragen: X-Achse (rot), Y-Achse (grün) und Z-Achse (blau). Phasen I – VI: Neigen in der X-Achse (freihand), starkes Schütteln, ruhig halten, freier Fall, Aufprall, liegen am Boden.

beiden Phasen aufgetreten ist, kann festgestellt werden, dass dieser im Verhältnis zur Beschleunigung, beim Aufprall deutlich stärker ist. Der Ruck, im folgendem bezeichnet mit j , ist die zeitliche Ableitung der Beschleunigung:

$$\vec{j}(t) = \dot{\vec{a}}(t) = \ddot{\vec{v}}(t).$$

wobei a die Beschleunigung, v die Geschwindigkeit und t die Zeit ist.

In der letzten Phase liegt das *AccelBoard3D* in Ruhe auf dem Boden. Laut den Messwerten muss es fast plan – mit der Z-Achse nach unten gerichtet – orientiert sein, da die X- und Y-Achsen nahezu keine Beschleunigungen gemessen haben und die Beschleunigung der Z-Achse bei ca. 1 g lag, dies entspricht exakt der Gravitation der Erde.

Die anhand der äußeren Bedingungen von uns festgelegten Phasen können wahrscheinlich automatisch aus den aufgenommenen Daten bestimmt werden. Es können unterschiedliche Verfahren zur Hilfe genommen werden, beispielsweise die *Slow Feature Analysis*, welche sich hervorragend für solche Aufgabenstellungen eignet. Sie wird ausführlich in [WS02] beschrieben.

5. Experimente und Resultate

5.4 Test der Kraftsensoren

Die Kraftsensoren wurden im Hinblick auf ihre Funktionstüchtigkeit in einem M-Serie-Fuß getestet. Das Übersetzungsverhältnis zwischen der Kraft, die auf die Fußsohle einwirkt, und der Kraft, die auf den Sensor ausgeübt wird, war dabei 2 : 1. Das heißt, dass die gemessene Kraft halbiert ist. Der BrainDesigner wurde wieder als Programm zur Aufnahme der Messwerte genutzt. Der Fuß war beim Test vollständig mit einem M-Serie-Bein verbunden. Das Bein wurde mit dem Fuß auf den Boden gestellt und danach am oberen Ende leichten seitlichen Kräften ausgesetzt. Dadurch verschob sich der Massenschwerpunkt des Beines lotrecht über der Fläche des Fußes. Die Kräfte der vier Faraday-Sensoren sind in Abbildung 5.10 (oben) über die Zeit abgetragen. Während des Tests fiel das Bein nicht um, daher muss sich der Massenschwerpunkt immer lotrecht über der Fußsohle befinden haben. Aus den vier Kräften kann bei diesem Test auf einfache Weise auf die Höhe der Verschiebung des Massenschwerpunktes geschlossen werden. Folgende Formeln liefern die Verschiebungen in der sagittalen (V_s) sowie in der lateralen (V_l) Ebene:

$$V_s = \frac{F_{lh} + F_{rh}}{2} - \frac{F_{lv} + F_{rv}}{2}$$
$$V_l = \frac{F_{lh} + F_{lv}}{2} - \frac{F_{rh} + F_{rv}}{2}.$$

Die Indizes der gemessenen Kraft F setzen sich dabei aus der Position des Kraftsensors innerhalb des Fußes zusammen, wobei l für links, r für rechts, h für hinten und v für vorne steht. Die Verschiebungen V_s und V_l , die beim Test gemessen wurden, sind in der Abbildung 5.10 (unten) gegeneinander abgetragen. Der Nullpunkt des Diagramms spiegelt dabei folgende Situation wider: Der Massenschwerpunkt des Beines liegt exakt lotrecht über der Mitte des Rechtecks, welches von den vier Sensoren aufgespannt wird. Laut der Abbildung ist dieser Fall beim Test nicht eingetreten. Stattdessen befand sich der Schwerpunkt immer über dem hinteren Teil des Roboterfußes und schwankte zwischen der linken und rechten Seite hin und her.

Der Test zeigt die Funktionstüchtigkeit aller vier Kraftsensoren und lässt darauf schließen, dass der Roboter anhand der Daten dieser Sensoren balanciert werden kann, da schon kleinste Änderungen der Verschiebung des Massenschwerpunktes registriert werden.

5. Experimente und Resultate

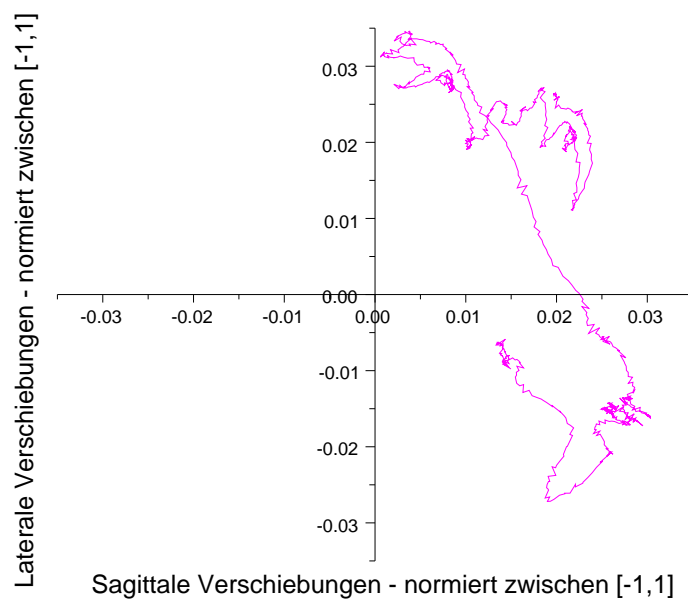
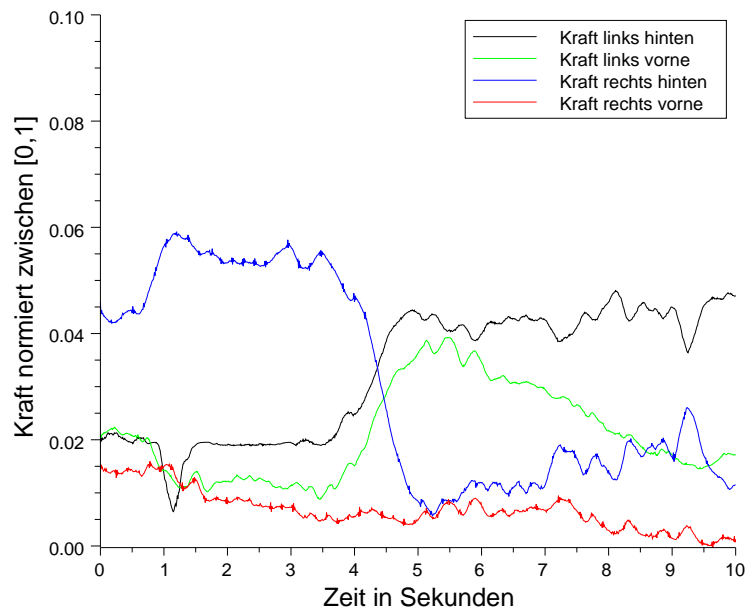


Abbildung 5.10.: Messwerte aller vier Kraftsensoren eines Fußes abgetragen über die Zeit (oben). Kraftverteilung in der sagittalen und lateralen Ebene (unten).

6 Fazit und Ausblick

Das Ziel, eine funktionstüchtige und multimodale Sensorik für den humanoiden Roboter *M-Serie* zu entwickeln, wurde erreicht. Insgesamt stehen dem Roboter 192 multimodale Messwerte zur Verfügung. Davon sind 63 Beschleunigungswerte, 44 Motorpositionen, 22 Winkelwerte, 35 Stromwerte, 20 Spannungswerte und 8 Kraftwerte. Die Firmware des *AccelBoard3D* garantiert eine Aktualisierungsrate von 100 Hz für alle Messwerte. Ferner ist jeder Wert auf einen einheitlichen Bereich normiert, sodass eine unkomplizierte Verwendung der Daten zur Bewegungssteuerung möglich ist. Die Kalibrierung der einzelnen Sensoren ist ohne großen zeitlichen Aufwand durchführbar. Dazu kommt entweder das vorgestellte Programm *AB3D-Testkonsole* oder eine mechanische Kalibrierung zum Einsatz.

Isolierte Experimente haben eine hohe Genauigkeit der Sensoren gezeigt und lassen auf einen großen Nutzen zur Bewegungssteuerung des humanoiden Roboters schließen. Erste praktische Bewegungstests zeigen, dass der Roboter einfache motorische Aufgaben, anhand seiner sensorischen Informationen, erledigen kann. Das Greifen und Heben eines Objektes und ein neuronales Laufen innerhalb eines Teststandes konnten vom Roboter durchgeführt werden.

Weiterhin konnte gezeigt werden, dass das Messprinzip des kapazitiven Winkelsensors funktioniert. Zur Zeit werden an allen Freiheitsgraden des M-Serie-Roboters kostenintensive Potentiometer zur Erfassung der Winkelstellung eingesetzt. Daher ist die Verwendung des kapazitiven Winkelsensors mit einem verbesserten mechanischen Aufbau innerhalb der *M-Serie* wünschenswert. Bei Freiheitsgraden, deren Winkelstellung nur mit geringer Auflösung gemessen werden muss, ist der sofortige Einsatz des gegenwärtigen Sensors problemlos möglich. Die niedrigen Herstellungskosten und das einfache Anbringen an den Gelenkachsen machen ihn dafür sehr attraktiv.

Diese Reichhaltigkeit der sensorischen Informationen hebt den M-Serie-Roboter deutlich von bisherigen Umsetzungen humanoider Roboter ab. Vergleichbare Projekte haben in der Regel keine Redundanzen im Hinblick auf die Propriozeption, siehe [ARA⁺06, KMR⁺08, BLK⁺07].

Die M-Serie-Roboter können auf verschiedenste Weise propriozeptives Wissen aus dem Vektor der sensorischen Werte extrahieren. Für die Feststellung der Lage im Raum einzelner Gliedmaßen sind folgende Ansätze denkbar:

6. *Fazit und Ausblick*

- Jedes Körperteil besitzt mindestens einen 3-Achsen-Beschleunigungssensor; im statischen Fall kann die Lage des Körperteils im Raum anhand der Beschleunigungswerte bestimmt werden.
- Ist der Servomotor über eine Feder von der Welle des Freiheitsgrades entkoppelt, gibt der Unterschied zwischen der Winkelstellung des Motors und der des Freiheitsgrades Auskunft über das am Freiheitsgrad von außen erzeugte Drehmoment. Dieses Drehmoment ist unter anderem von dem Gewicht und der Lage des Massenschwerpunktes der Gliedmaße abhängig. Wirken neben der Gravitation keine weiteren Kräfte, kann auf die Lage des Körperteils im Raum geschlossen werden.
- Das von einem Servomotor erzeugte Drehmoment und die dabei aufgenommene elektrische Leistung um eine Gliedmaße in Position zu halten, lassen unter den vorherig angenommen Voraussetzungen wieder auf die Lage des Körperteils im Raum schließen.

Die aufgelisteten Möglichkeiten veranschaulichen beispielhaft das Potenzial, das durch die vielen Sensorwerte gegeben ist. Diese Zusammenhänge sind für den äußeren Betrachter zwar nicht immer sofort ersichtlich, aber eine neuronale Bewegungssteuerung, die womöglich sogar über eine künstliche Evolution entstanden ist, vermag noch viel ausgefeiltere Zusammenhänge der Sensorwerte zu nutzen.

Der erste M-Serien-Roboter war mit der Fertigstellung dieser Arbeit vollständig einsatzbereit, sodass ab jetzt Experimente mit dem gesamten Roboter durchgeführt werden können. Die Ergebnisse der Experimente werden zeigen, welche komplexen Bewegungsabläufe durch die Qualität und die Mannigfaltigkeit der vorgestellten Sensoren möglich sind.

Literaturverzeichnis

- [AD] ANALOG DEVICES, INC.: *An IC Amplifier User's Guide to Decoupling, Grounding, and Making Things Go Right for a Change. AN-202. Application Note*), Rev. B Auflage.
- [AD89] ANALOG DEVICES, INC.: *Grounding for low- and high-frequency circuits. AN-345. (Application Note)*, Rev. 1 Auflage, 1989.
- [AD07] ANALOG DEVICES, INC.: *AD7150 (Datenblatt)*, Rev. 0 Auflage, 2007.
- [AD08] ANALOG DEVICES, INC.: *AN-925 - Sensors for the AD7147 and AD7148 CapTouch Controllers*, Rev. B Auflage, 2008.
- [ALE] *ALEAR project website*. Webseite. Online verfügbar unter <http://www.alear.eu/>; besucht am 01.03.2010.
- [ARA⁺06] ASFOUR, T., K. REGENSTEIN, P. AZAD, J. SCHRÖDER, A. BIERBAUM, N. VAHRENKAMP und R. DILLMANN: *ARMAR-III: An Integrated Humanoid Platform for Sensory-Motor Control*. Seite 7, 2006.
- [BLK⁺07] BUSCHMANN, THOMAS, SEBASTIAN LOHMEIER, KOLJA KÜHNLENZ, MARTIN BUSS, HEINZ ULBRICH und FRIEDRICH PFEIFFER: *LOLA - a Performance Enhanced Humanoid Robot (LOLA - ein leistungsgesteigerter humanoider Roboter)*. *it - Information Technology*, 49(4):218–, 2007.
- [CB01] CERMAK, S.P. und G. BRASSEUR: *A Planar Capacitive Sensor for Angular Measurement*. *IEEE Instrumentation and Measurement Technology Conference*, Seiten 1393–1396, 2001.
- [Eve95] EVERETT, H. R.: *Sensors for mobile robots : theory and application*. A K Peters, Ltd., Natick, 1995.
- [FGMT04] FERRARI, VITTORIO, ALESSIO GHISLA, DANIELE MARIOLI und ANDREA TARONI: *Capacitive Angular-Position Sensor with*

Literaturverzeichnis

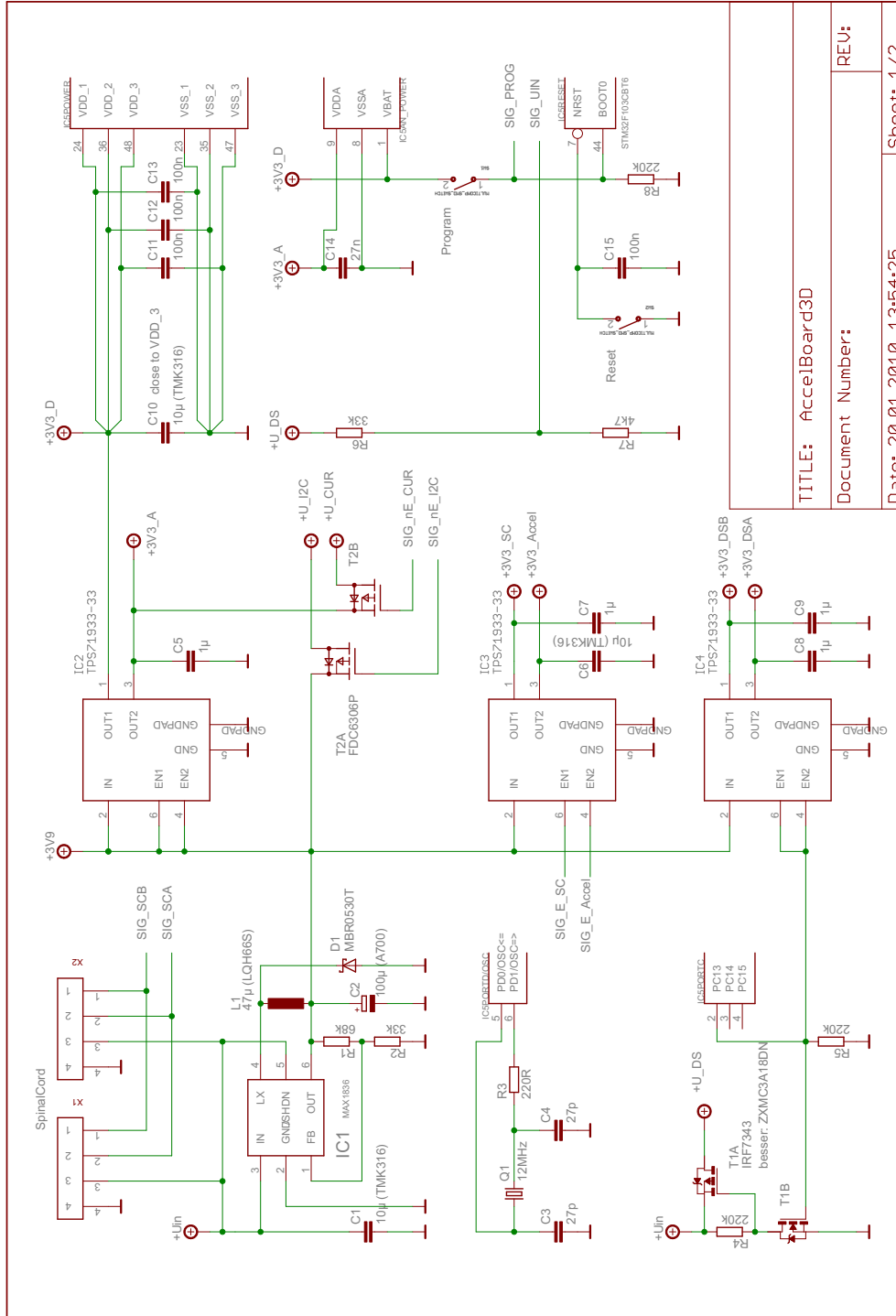
- Electrically-Floating Conductive Rotor and Measurement Redundancy*. IMTC 2004 - Instrumentation and Measurement Technology Conference, Seiten 195–200, 2004.
- [Fre09] FREESCALE SEMICONDUCTOR: *MMA7455L (Datenblatt)*, Rev. 7 Auflage, 2009.
- [Hen08] HENGSTLER GMBH: *Absolutgeber AC 36 (Datenblatt)*, Rev. 1 Auflage, 2008.
- [Hil07] HILD, MANFRED: *Neurodynamische Module zur Bewegungssteuerung autonomer mobiler Roboter*. Doktorarbeit, Humboldt-Universität zu Berlin, 2007.
- [IB 06] IB HOCH: *FARADAY-Kraftsensoren (Datenblatt)*, Rev. 1 Auflage, 2006.
- [KLvS⁺08] KLUG, SEBASTIAN, THOMAS LENS, OSKAR VON STRYK, BERNHARD MÖHL und ANDREAS KARGUTH: *Biologically Inspired Robot Manipulator for New Applications in Automation Engineering*. In: *Proceedings of Robotik 2008*, Nummer 2012 in *VDI-Berichte*, Munich, Germany, June 11-12 2008. VDI Wissensforum GmbH.
- [KMR⁺08] KONYEV, M., A. MELNIKOV, A. RUDSKYY, V. RUSIN, A. TELESK und Y. ZAVGORODNIY: *Echtzeitfähiges verteiltes Steuerungssystem für Schreitroboter*. Seite 5, 2008.
- [Kun04] KUNDERT, KEN: *Power Supply Noise Reduction*. Technischer Bericht, Designer's Guide Consulting, Inc., 2004.
- [Lin07] LINEAR TECHNOLOGY: *LTC2854/LTC2855 (Datenblatt)*, Rev. 1 Auflage, 2007.
- [LSB09] LUKAC, DUSKO, TORSTEN SIEDEL und CHRISTIAN BENCKENDORFF: *Designing the test Feet of the Humanoid Robot M-Series*. 2009.
- [MIP06] MAXIM INTEGRATED PRODUCTS, INC.: *MAX1836-MAX1837 (Datenblatt)*, Rev. 3 Auflage, 2006.

Literaturverzeichnis

- [Nat01] NATIONAL SEMICONDUCTOR: *TPS78833 (Datenblatt)*, JULY 2001 Auflage, 2001.
- [Nat08] NATIONAL SEMICONDUCTOR: *NS-ADC121C027 (Datenblatt)*, Rev. 1 Auflage, 2008.
- [Nat09a] NATIONAL SEMICONDUCTOR: *LMP8601/LMP8601Q*, Rev. 1 Auflage, 2009.
- [Nat09b] NATIONAL SEMICONDUCTOR: *LP2950 (Datenblatt)*, Rev. 1 Auflage, 2009.
- [NRL10] *Neurorobotics Research Laboratory (NRL) website*. Webseite, 2010. Online verfügbar unter <http://www.neurorobotics.eu/>; besucht am 01.03.2010.
- [RC08] ROBOTIS CO., LTD.: *Dynamixel RX-28 (Datenblatt)*, Rev. 1.10 Auflage, 2008.
- [STM09] STMICROELECTRONICS FOUNDATION: *STM32F101x8 - STM32F101xB (Datenblatt)*, Rev. 12 Auflage, 2009.
- [Thi10] THIELE, CHRISTIAN: *BrainDesigner website*. Webseite, 2010. Zukünftig online verfügbar unter <http://neurorobotics.eu/braindesigner/>.
- [Vis06] VISHAY: *Vishay Spectrol - Model 358 (Datenblatt)*, 19-Jul-06 Auflage, 2006.
- [WS02] WISKOTT, LAURENZ und TERRENCE J. SEJNOWSKI: *Slow Feature Analysis: Unsupervised Learning of Invariances*. *Neural Computation*, 14(4):715–770, 2002.

A. Anhang

A Anhang



TITLE: AccelBoard3D	REV:
Document Number:	
Date: 20.01.2010 13:54:25	Sheet: 1/2

Abbildung A.1.: Vollständiger Schaltplan des AccelBoard3D (Blatt 1/2).

TECHNISCHE REALISIERUNG MULTIMODALER SENSORIK FÜR HUMANOIDE ROBOTER

A. Anhang

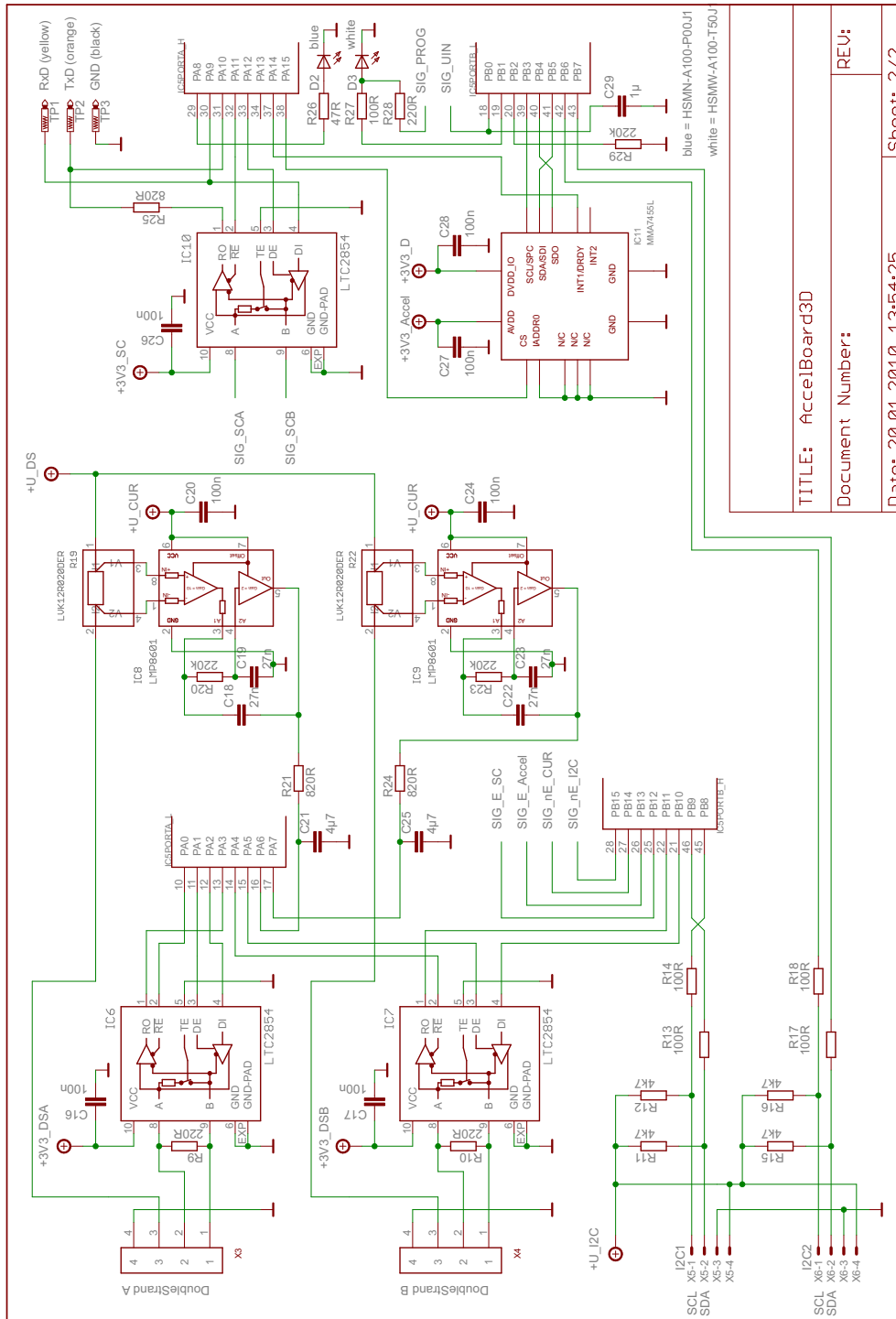


Abbildung A.2.: Vollständiger Schaltplan des *AccelBoard3D* (Blatt 2/2).

Selbstständigkeitserklärung

Ich, Christian Benckendorff, Matrikel-Nr. 196765, versichere hiermit, dass ich meine Diplomarbeit mit dem Thema

Technische Realisierung multimodaler Sensorik für humanoide Roboter

selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, wobei ich alle wörtlichen und sinngemäßen Zitate als solche gekennzeichnet habe. Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Mir ist bekannt, dass ich meine Diplomarbeit zusammen mit dieser Erklärung fristgemäß nach Vergabe des Themas in dreifacher Ausfertigung und gebunden im Prüfungsamt der Humboldt-Universität zu Berlin abzugeben oder spätestens mit dem Poststempel des Tages, an dem die Frist abläuft, zu senden habe.

Berlin, den 14. März 2010

CHRISTIAN BENCKENDORFF

Einverständniserklärung

Ich, Christian Benckendorff, Matrikel-Nr. 196765, erkläre hiermit mein Einverständnis, dass die vorliegende Diplomarbeit mit dem Thema

Technische Realisierung multimodaler Sensorik für humanoide Roboter

in der Bibliothek des Instituts für Informatik der Humboldt-Universität zu Berlin ausgestellt werden darf.

Berlin, den 14. März 2010

CHRISTIAN BENCKENDORFF