



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Marvin Kirchner

Entwicklung eines Manipulators mit Endeffektor für eine autonome, mobile Roboterplattform.

*Fakultät Technik und Informatik
Department Fahrzeugtechnik und Flugzeugbau*

*Faculty of Engineering and Computer Science
Department of Automotive and
Aeronautical Engineering*

Marvin Kirchner

**Entwicklung eines Manipulators mit
Endeffektor für eine autonome, mobile
Roboterplattform.**

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang Mechatronik
am Department Fahrzeugtechnik und Flugzeugbau
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Erstprüfer: Prof. Dr. rer. nat. Thomas Lehmann
Zweitprüferin : Prof. Dr. -Ing. Jutta Abulawi

Abgabedatum: 04. Januar 2018

Zusammenfassung

Marvin Kirchner

Thema der Bachelorthesis

Entwicklung eines Manipulators mit Endeffektor für eine autonome, mobile Roboterplattform.

Stichworte

Manipulator, Endeffektor, Mechatronik, Mechanik, Elektronik, Aktoren, Sensoren, Steuerung, Mikrocontroller, methodisches Konstruieren, Software Engineering, Linearmotor, Schrittmotor, Arduino.

Kurzzusammenfassung

Diese Arbeit befasst sich mit der Entwicklung eines Manipulators mit Endeffektor für eine autonome, mobile Roboterplattform. Es werden alle Entwicklungsphasen beginnend bei der Analyse der Randbedingungen über das Design und dessen Implementierung bis hin zum Testen durchlaufen. Die Aufgabe des Manipulators ist es als Lernplattform von den Studierenden der HAW Hamburg genutzt zu werden. Der Manipulator soll eine mobile Roboterplattform dazu befähigen eine Rolltafel im 7. Stock des Hauptgebäudes der HAW Hamburg bewegen. Die Studierenden lernen und üben bei dem Rangieren der Rolltafel Fahralgorithmen zu entwickeln.

Marvin Kirchner

Title of the paper

Development of a manipulator with endeffector for an autonomous, mobile robot platform.

Keywords

Manipulator, endeffector, mechatronic, mechanic, electronic, actuators, sensors, steering, microcontroller, methodical construction, software engineering, linear motor, stepper motor, Arduino.

Abstract

This thesis is about developing a manipulator with endeffector for an autonomous, mobile robot platform. All stages of the development, beginning with the initial analysis of given conditions over the design and its implementation as well as final testing of the product, are be part of this thesis. The objective of the manipulator is to be used as a learning device for students of the HAW Hamburg. The manipulator should enable a mobile robot platform to move a rollable board around the 7th floor of the HAW main building. The students learn and practice to develop driving algorithms by programming the mobile robot platform to move the board.

Inhaltsverzeichnis

Abbildungsverzeichnis	9
Tabellenverzeichnis	11
Quellcodeverzeichnis	12
Akronyme	13
1. Einleitung	14
1.1. Problemstellung	15
1.2. Systemidee	15
1.3. Aufbau der Arbeit	16
2. Analyse der Randbedingungen	17
2.1. Zur Verfügung stehende Mittel für die Fertigung	17
2.2. Analyse der Basisarchitektur	18
2.2.1. Rolltafel	18
2.2.2. Fahrzeug	19
2.3. Stakeholder	21
2.4. Use Case	22
2.5. Requirements	23
2.6. Systemanalyse	25
2.6.1. Systemakteure und Objektflüsse	25
2.6.2. Funktionsanalyse	27
3. Design	29
3.1. Konzeptbildung	30
3.1.1. Lösungsfindung und Konzeptdefinition	30
3.1.2. Konzeptbewertung	34
3.2. Mechanik	35
3.2.1. Konstruktion	35
3.2.2. Safety	43
3.2.3. Überschlägige Berechnungen von Kräften und Stabilität	45

3.3. Elektronik	53
3.3.1. Aktoren	54
3.3.2. Sensoren	56
3.3.3. Steuergeräte	60
3.3.4. Safety	61
3.4. Software	62
3.4.1. Klassen	63
3.4.2. State Machine	64
4. Implementierung	67
4.1. Mechanik und Elektronik	67
4.1.1. Montage	67
4.1.2. Vergleich mit Design	68
4.1.3. PCB Fertigung	72
4.2. Software	73
4.2.1. updateStatus	74
4.2.2. steeringSTM	75
4.2.3. Auswertung der Sensoren	77
5. Testen	80
6. Fazit und Ausblick	81
Literatur	82
A. Anhang	84
A.1. Inhalt der beigefügten CD	84
A.2. Verwendete Geräte und Software	84
Glossar	85

Abbildungsverzeichnis

2.1. Rolltafel	18
2.2. Messung der Losbrechkraft	19
2.3. Mobile Roboterplattform [ROB17]	20
2.4. Messung der Traktion	20
2.5. Systemkontext Manipulator	25
2.6. Systemkontext mit Objektfluss	26
3.1. Konzept 1	31
3.2. Konzept 2	32
3.3. Konzept 3	33
3.4. Zusammenbau Manipulator mit Fahrzeug	35
3.5. Fahrzeug unter Rolltafel	36
3.6. Untergestell isometrisch vorne links	37
3.7. Untergestell isometrisch hinten rechts	38
3.8. Untergestell isometrisch unten links	39
3.9. Antriebsstrang	39
3.10. Hubarm isometrisch hinten rechts	40
3.11. Hubarm isometrisch unten rechts	41
3.12. Endeffektor isometrisch vorne rechts	42
3.13. Positionierungstoleranzen	43
3.14. Klemmbereiche	44
3.15. Schutzbleche	44
3.16. Nutbelastung	47
3.17. Zug an Schraube im Dachblech	49
3.18. FEM Analyse der Führungsschale	50
3.19. Nickmoment	51
3.20. Elektronik Blockschaltbild	53
3.21. Linearmotor [Ama17]	55
3.22. Schrittmotor [Pol17a]	55
3.23. Linearmotor [Ele17]	56
3.24. Adapter Schiebepoti Schnittansicht	57
3.25. Diametralmagnet über Magnetsensor [AMS17]	58

3.26. AS5601 Magnetfeldsensor [AMS17]	58
3.27. PCB Halterung	59
3.28. PCB Design	60
3.29. Motorshield [Ada17]	61
3.30. Notausschalter	62
3.31. Klassendiagramm	63
3.32. STM Manipulator	64
3.33. Statusinfo String	65
3.34. STM Steering	65
4.1. Schwere Zugänglichkeit bei der Montage	68
4.2. Einsetzen der T-Matik Gewinde mit T25 Torx	68
4.3. Implementierung Manipulator mit Fahrzeug	69
4.4. Implementierung Untergestell	70
4.5. Implementierung Hubarm	71
4.6. Kabelführung	71
4.7. Implementierung Endeffektor	71
4.8. Fertigungsschritte der Winkelsensorsorik	72
4.9. Protokoll zur I2C Datenübertragung mit AS5601 [AMS17]	77

Tabellenverzeichnis

2.1. Stakeholder	21
2.2. Requirements	24
2.3. Funktionsanalyse	28
3.1. Morphologischer Kasten	30
3.2. Konzeptbewertung	34
5.1. Trace Matrix Testfälle	80

Quellcodeverzeichnis

4.1. Hauptprogramm	74
4.2. ManipulatorSTM	74
4.3. Processupdate	75
4.4. Regelung des Markenflusses	76
4.5. Aktionen in den Zuständen	77
4.6. Auswertung des Winkelsensors	78
4.7. Auswertung des Höhensensors	78

Akronyme

bdd block definition diagram.

FEM Finite Elemente Methode.

FZG Fahrzeug.

GS Gesamtsystem.

ibd internal block diagram.

IDE Integrated Development Environment.

KISS Keep it simple and stupid.

MAN Manipulator.

PCB Printed Circuit Board.

RT Rolltafel.

STM State Machine.

SysML Systems Modeling Language.

SYSMOD System Modeling Process.

1. Einleitung

Autonomes Fahren und *Industrie 4.0* sind zwei der großen Schlagworte in der aktuellen technischen Forschung. Forscher prognostizieren, dass die Mobilität der Zukunft durch rechnergestützte Systeme mit intelligenten Fahralgorithmen geprägt ist. Das Auto der Zukunft wird nicht mehr von Menschen, sondern von Computern gesteuert. Bereits heute unterstützen zahlreiche Steuergeräte den Fahrer, wovon er in der Regel nichts bemerkt. Treten Gefahrensituationen ein, greift die Traktionskontrolle, das Antiblockiersystem oder der Bremsassistent ein, um nur einige Beispiele zu nennen. Der Trend geht dahin, dass der Mensch sich gar nicht mehr um die Kontrolle des PKW kümmern muss. In einer Zukunftsvision lehnt sich der Mensch entspannt zurück, während das Auto ihn an seinen Zielort bringt.

Der Trend nach mehr autonomer Technik beschränkt sich jedoch nicht nur auf die Mobilität zur Beförderung von Personen, sondern hat auch längst Einzug in der Industrie gehalten. Der Blick richtet sich auch auf den Transport von Gütern und Waren oder Produktionsprozesse. In modernen Logistikzentren übernehmen autonome Roboterplattformen den Warentransport oder unterstützen in Werkshallen den Produktionsprozess. Paketzusteller, wie HERMES oder DHL, experimentieren mit autonomen Roboterplattformen, die bereits Pakete aufnehmen und zustellen können.

Der Bedarf an Wissen über autonome Robotersysteme ist vorhanden und wächst stetig an. Das dritte große Schlagwort ist die *Lehre*. Universitäten und Hochschulen selbst haben nicht das Ziel Waren in großen Stückzahlen herzustellen und über den Markt zu verkaufen. Sie sind vielmehr Produzenten von Wissen und haben den Anspruch dieses Wissen zu vermitteln. Ingenieure arbeiten in der Industrie, werden aber von Universitäten und Hochschulen ausgebildet. Die Hochschulen vermitteln den angehenden Ingenieuren das Wissen, welches sie später in der Industrie anwenden müssen. Der Hochschulalltag mit theoretischen Vorlesungen wird zunehmend durch Praktika und Projektaufgaben ergänzt. Die Theorie wird sehr viel besser verstanden, wenn sie durch die Praxis gefestigt wird. Die finanziellen Mittel, die der Lehre im Allgemeinen zur Verfügung stehen sind in der Regel geringer, als in der Industrie. Es ist daher sinnvoll, Studierenden günstige Hardware für ihre ersten Programmerversuche zur Verfügung zu stellen. Eine effiziente und gute Option sind Lernplattformen. Sie haben einzig und allein den Zweck, dass an ihnen das theoretische Wissen erprobt und gefestigt wird. Spielerische und experimentelle Aspekte sind hierbei ebenfalls vorhanden.

1.1. Problemstellung

Der Hintergrund des Themas für diese Bachelorarbeit ergibt sich aus dem Vorhaben einer Projektgruppe des Informatikstudiengangs der HAW Hamburg. Die Projektgruppe hat die Aufgabe erhalten mit einem Fahrzeug (FZG) eine Rolltafel (RT) im 7. Stock des Hauptgebäudes der HAW Hamburg rangieren. Die Aufgabe ist von einem wissenschaftlichen Mitarbeiter der HAW Hamburg als Lernprojekt initialisiert worden. Es geht dabei um das Erlernen und Üben von Fahralgorithmen. Grundsätzlich muss der zu rangierende Gegenstand keine RT sein. Diese aber sind im 7. Stock des Hauptgebäudes der HAW Hamburg ausreichend vorhanden und, nach Einschätzung des wissenschaftlichen Mitarbeiters, durch ihre rollenden Eigenschaften voraussichtlich verhältnismäßig leicht zu handhaben.

Das Rangieren der RT hat zunächst einen Selbstzweck. Studenten der Fakultät Technik und Informatik, wie z. B. die bereits erwähnte Projektgruppe, sollen hieran in erster Linie lernen wie man Gegenstände, in diesem speziellen Fall eine RT, im Raum rangiert.

Die Zielsetzung dieser Arbeit besteht darin, eine Vorrichtung zu entwerfen mit dessen Hilfe eine RT mit einem bereits vorhandenem FZG bewegt werden kann. Diese Vorrichtung soll konstruiert, gebaut und getestet werden. Die Entwicklung soll dazu nach der Methodik wie z. B. aus [Nae09] und die Modellierung mit Werkzeugen wie z. B. der SysML [Wei14] erfolgen. Die Vorrichtung soll auch in Zukunft für weitere Projekte genutzt werden.

1.2. Systemidee

Die zu entwickelnde Vorrichtung soll im weitesten Sinne ein Objekt bewegen oder verrücken. Der Fachterminus aus der Robotertechnik ist hierfür manipulieren. Im Folgenden wird die Vorrichtung Manipulator (MAN) genannt. Ein Manipulator ist ein Mechanismus, der gewöhnlich aus mehreren Segmenten besteht, die sich relativ zueinander bewegen können, wie z. B. drehen oder gleiten, um Objekte zu greifen oder zu bewegen [San99]. Jeder MAN verfügt über einen Endeffektor. Ein Endeffektor ist das letzte Glied in der Bewegungskette eines Manipulators, welches in direktem Kontakt zu dem zu manipulierenden Objekt steht. Dies kann z. B. ein Werkzeug oder Greifer sein [San99]. Das FZG ist zugleich auch eine Roboterplattform, da auf ihr der MAN, montiert werden soll. Im Verlauf wird weiterhin der Begriff FZG verwendet. Das Gesamtsystem ist das Fahrzeug, der Manipulator und die Rolltafel im Verbund.

Zusammengefasst ist die Systemidee, auf einem FZG einen MAN zu montieren, dessen Endeffektor Kräfte auf eine RT ausübt, sodass sich diese kontrolliert bewegt.

1.3. Aufbau der Arbeit

Grundsätzlich ist der Aufbau einer wissenschaftlichen Arbeit in seiner übergeordneten Struktur immer gleich. Vergleicht man den System Modeling Process (SYSMOD) aus [Wei14] mit dem methodischen Konstruieren nach [Nae09], so stellt man fest, dass jede Disziplin ihre eigenen Schwerpunkte und Akzente setzt. Die Gliederung bzw. der Aufbau dieser Arbeit ist eine hybride Form. Ein Beispiel hierfür ist der morphologische Kasten, welcher in [Wei14] nicht zum Tragen kommt, jedoch in der methodischen Konstruktion des Maschinenbaus ein grundlegender Pfeiler ist. Eine Gliederung, die sich aus unterschiedlichen Disziplinen bedient, bietet sich an, da das Thema dieser Arbeit einen gänzlich mechatronischen Charakter aufweist und somit die drei Bereiche Maschinenbau, Elektrotechnik und Softwareengineering abdeckt und miteinander vereint. Es wird der komplette Entwicklungszyklus mit den Phasen der Analyse, Planung, Beschaffung und dem Fertigen von Bauteilen, bis hin zur Implementierung und dem Testen durchlaufen.

Von außen könnte der Eindruck entstehen, dass diese Arbeit über alle seine Arbeitsphasen in einem Guss entstanden ist. In der Regel ist der Entwicklungsprozess jedoch häufig ein iterativer. Das bedeutet Widersprüche im Design fallen erst bei der Implementierung oder beim Testen auf. Dadurch werden Anforderungen angepasst und das Design wiederum umgeschrieben, so lange, bis ein akzeptables Resultat vorliegt.

Die Arbeit weist einen großen Workload an praktischer und handwerklicher Arbeit auf. Dies ist mit aussagekräftigen Abbildungen dokumentiert.

2. Analyse der Randbedingungen

Die Analyse der Randbedingungen ist ein wichtiger Schritt zur Vorbereitung für das Design. Das Design richtet sich hiernach aus und wird teilweise sogar limitiert.

2.1. Zur Verfügung stehende Mittel für die Fertigung

Die hier beschriebenen Faktoren sind in erster Linie Limitierungen. Es ist ein klares Ziel dieser Arbeit nicht nur theoretisch, sondern auch praktisch einen funktionsfähigen Prototypen zu implementieren. Es ist sehr wichtig im Vorfeld zu wissen, welche Möglichkeiten in der Fertigung und bei der Implementierung vorhanden sind. Es macht keinen Sinn im Design davon auszugehen, dass ein Bauteil mit einer CNC Fräse gefertigt wird, die jedoch gar nicht zur Verfügung steht, oder geschriebene Arbeitsaufträge für Werkstätten zu erarbeiten, die wochenlange Vorlaufzeiten haben, wodurch diese zu spät bearbeitet werden.

Räumlichkeiten

Vom Department Technik und Informatik wird im 7. Stock des HAW Hauptgebäudes Berliner Tor 7 eine Werkstatt mit allen gängigen Handwerkzeugen, wie Feilen, Sägen, Schraubendreher usw. zur Verfügung gestellt. Für Studenten, die eine Abschlussarbeit schreiben, oder in Projekte involviert sind, wird ein kleiner Gemeinschaftsraum als Arbeitsplatz zur Verfügung gestellt. Dieser Raum ist abschließbar und kann nur von den Mitarbeitern der HAW und den hierfür freigeschalteten Studenten benutzt werden.

Werkzeuge und Maschinen

In der Werkstatt befindet sich eine Standbohrmaschine. Weiterhin besteht die Möglichkeit 3D Druckteile in Auftrag zu geben. Hierfür müssen keine Arbeitsaufträge geschrieben werden, was die Fertigung beschleunigt. Es muss lediglich eine STL-File an einen entsprechenden Mitarbeiter geschickt werden, der den Druck in ein bis zwei Tagen fertigstellen kann. Es handelt sich dabei um das Stereolithografieverfahren. Es ist eine breite Palette an Druckmitteln vorhanden, deren Bauteile zwar mechanisch nicht sehr belastbar, aber äußerst maßhaltig sind.

Finanzielle Mittel

Alle Bestellungen können über die HAW mit einer VOL Bestellung abgeschlossen werden.

Voraussetzung ist, dass der Händler dieses Art der Bestellung akzeptiert. Vielen Händlern ist dieses Verfahren nicht bekannt, was die Bestellungen erschwert. Andere Händler sind grundsätzlich nicht bereit eine VOL Bestellung anzunehmen. Alle weiteren Bestellungen erfolgen vor dem Hintergrund eines Studentenbudgets.

2.2. Analyse der Basisarchitektur

Hier wird die bereits vorhandene Hardware sowie dessen physikalisches Verhalten vorgestellt.

2.2.1. Rolltafel

Die RT ist der Gegenstand, welcher von dem FZG bewegt werden soll. Die Maße der RT sind in Abbildung 2.1 eingetragen.

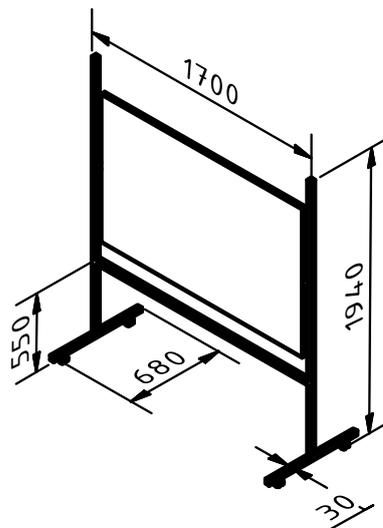


Abbildung 2.1.: Rolltafel

Ihr Gesamtgewicht beträgt 26 kg, welches sich gleichmäßig auf die Rollenpaare verteilt. Der Bodenkontakt wird über vier frei drehbare Rollen hergestellt. Der Rahmen besteht aus 30 mm × 30 mm Profilen. In Abbildung 2.2 ist dargestellt, wie die Losbrechkraft F_{los} gemessen wird.

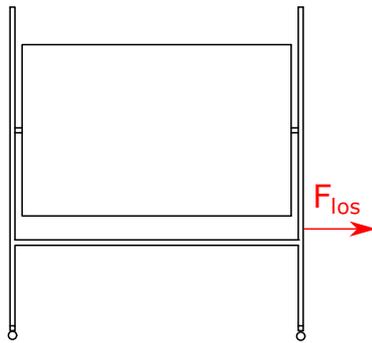


Abbildung 2.2.: Messung der Losbrechkraft

Um die senkrechte Strebe der RT wird eine Kofferwaage geschnallt. Die Zugkraft wird dann mit der Kofferwaage solange erhöht, bis die RT anfängt sich zu bewegen. Die Haftreibung ist stets größer als die Gleitreibung. Daher wird die Messung beendet, sobald die RT rollt. Da die Kofferwaage in kg misst, wird der Messwert in Newton umgerechnet $F_{los} = 2,1 \text{ kg} \cdot 10 \text{ m/s}^2 = 21 \text{ N}$.

Bei einem weitere Handversuch wird die RT ca. 10 m gerollt und dabei auf ihre Spurtreue beobachtet. Die Kraft greift dabei, wie in Abbildung 2.2, an. Während die RT rollt, ist keine Driftbewegung zu erkennen.

2.2.2. Fahrzeug

Die Ausgangsbasis ist das FZG, eine mobile Roboterplattform der Firma NEXUS (Abbildung 2.3).

Der MAN soll hierauf montiert werden. Hier die wichtigsten Kenndaten [ROB17]:

- Material: Aluminiumlegierung
- Speed: 0,3 m s
- Motor Power: 17 W
- Motortype: 12 V DC
- Gewicht: 4,3 kg
- Microcontroller: Atmega 328
- Batterie: 12 V Ni-Mh 1,8 A h



Abbildung 2.3.: Mobile Roboterplattform [ROB17]

Die Abmaße der Dachplatte, welche nicht in der Abbildung 2.3 zu sehen ist, sind $200\text{ mm} \times 245\text{ mm} \times 1\text{ mm}$. Um zu überprüfen, ob das FZG die RT ziehen kann, wird eine Traktionsmessung durchgeführt. Hierfür wird das FZG sukzessive mit Gewichten beschwert. Mit einer Kofferwaage wird das FZG entgegen seiner Bewegungsrichtung festgehalten. Das Ergebnis der Messung ist in Abbildung 2.4 festgehalten.

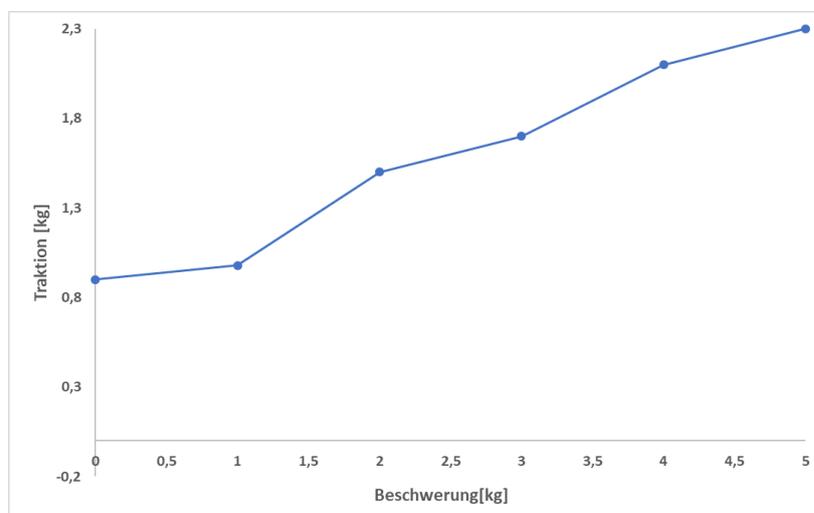


Abbildung 2.4.: Messung der Traktion

Umgerechnet hat das FZG ohne Beschwerung eine Traktion von $< 10\text{ N}$ und ist somit nicht in der Lage die RT zu bewegen. Das FZG benötigt somit einen Anpressdruck, um die Traktion zu steigern.

2.3. Stakeholder

Die Bedürfnisse der Stakeholder sollten im weitesten Sinne befriedigt werden, da sie die Quelle der Anforderungen sind. Es stellt sich die Frage wer Interesse an dem MAN hat. Wer ist von der Entwicklung des MAN betroffen? Wer wird ihn verwenden und schlussendlich entsorgen? Gibt es Personen oder Institutionen die, evtl. auch erst in der Zukunft, negativ beeinflusst wären, wenn man sie nicht berücksichtigt oder der MAN ausfällt [Wei14]? Die Tabelle 2.1 listet die Stakeholder in kompakter Weise, beginnend mit der höchsten Priorität, auf. Der Tabellenaufbau ist an [Leh14] angelehnt und um die Unterscheidung von sekundären Stakeholdern erweitert.

Tabelle 2.1.: Stakeholder

Stakeholder	Interessen
Primäre Stakeholder	
Wissenschaftliche Mitarbeiter der HAW Hamburg (Enrico Christophers)	Erstellung von Lern-angeboten/plattformen und Projektaufgaben für Studierende der HAW Hamburg. Wecken von Begeisterung und Motivation bei den Studierenden.
Studenten der HAW Hamburg	Nutzung des Manipulators als Übungs- und Lernobjekt.
Sekundäre Stakeholder	
Gutachter der Thesis	Bewerten von Qualität und Quantität des Entwicklungsprozesses
Professoren	Benötigen zu Beginn der Vorlesung eine Tafel. Wecken von Begeisterung bei Vorlesungsteilnehmern.
Reinigungspersonal	Rolltafel steht dem Reinigungspersonal nicht im Weg.

Die primären Stakeholder sind die offensichtlichen und auch wichtigsten. Sie haben das größte Interesse und werden am stärksten mit dem MAN konfrontiert. Sie sind es auch, die den MAN am Ende seiner Lebensdauer entsorgen oder recyceln. Fast alle Anforderungen kommen vom wissenschaftlichen Mitarbeiter Enrico Christophers. Dieser wird im Folgenden als Kunde bezeichnet.

Die sekundären Stakeholder sind nicht in erster Linie betroffen oder nicht offensichtlich als solche zu erkennen. Die Professoren und das Reinigungspersonal sind optionale Stakeholder, die selbst nicht zu ihren Wünschen bzw. Anforderungen befragt werden, man jedoch davon ausgeht, dass diese in Zukunft auch einen Mehrwert durch den MAN haben könnten. Bei der Entwicklung handelt es sich hier um einen Sonderfall, nämlich der Erstellung einer Abschlussarbeit. Es existieren daher nicht nur Stakeholder, die den MAN benutzen möchten, sondern auch Gutachter, die dessen Entwicklungsprozess und Funktionalität beurteilen. Ihre Meinung zu Annahmen, Berechnungen, Konzepten, Designentscheidungen usw. wird berücksichtigt.

2.4. Use Case

Um zu verstehen, was der MAN tun soll, ist es sinnvoll sich den täglichen Betrieb vor Augen zu führen. Dieser wird als Anwendungsfall modelliert. Ein Anwendungsfall ist ein Use Case mit bestimmter Form. Der folgende Aufbau ist aus [Leh14] entnommen.

Titel: Rangieren einer RT.

Akteuer: FZG, MAN und RT.

Ziel: Eine RT von Position A nach Position B rangieren.

Auslöser:

Interner Programmablauf des FZG.

Vorbedingungen:

- Der MAN ist auf dem FZG befestigt.
- Der MAN ist an die Stromversorgung des FZG angeschlossen.
- Die Datenverbindung zwischen MAN und FZG wurde hergestellt.
- Der Akku ist ausreichend geladen.
- Das FZG und der MAN sind eingeschaltet.

Nachbedingungen:

Akku wird aufgeladen.

Erfolgsszenario:

1. Das FZG bewegt sich im 7. Stock des HAW Hauptgebäudes selbstständig zur RT.
2. Das FZG positioniert den MAN in einen definierten Toleranzbereich zur RT (vgl. Abschnitt 3.2.1).
3. Das FZG richtet den Endeffektor durch Kommandos an den MAN aus. Endeffektor bewegt sich an die gewünschte Position. Der MAN sendet dauerhaft Statusinformationen an das FZG.
4. Das FZG erkennt anhand der Statusinformationen, dass das Kommando ausgeführt wurde und sendet dem MAN ein Kommando zum Bewegen des Endeffektors.
5. Der MAN führt seinen Endeffektor zur RT an die gewünschte Position. Der MAN sendet dauerhaft Statusinformationen an das FZG.

6. Das FZG erkennt anhand der Statusinformation, dass das Kommando ausgeführt wurde und rangiert die RT an die Zielposition.
7. An der Zielposition angekommen sendet das FZG ein Kommando zum Einziehen des Endeffektors an den MAN. Der MAN sendet dauerhaft Statusinformationen an das FZG.
8. Das FZG erkennt anhand der Statusinformation, dass das Kommando ausgeführt wurde und fährt in eine Parkposition.

2.5. Requirements

In Tabelle 2.2 sind die verbindlichen Anforderungen der Stakeholder aufgelistet. Der Aufbau der Tabelle ist an [Leh14] [Abu17] angelehnt. Die Requirements einstelliger ID sind die ursprünglichen Anforderungen des Kunden. Alle Requirements mit einer zwei- oder dreistelligen ID sind abgeleitete Requirements. Die abgeleiteten Requirements sind während des Design- und Implementierungsprozesses hinzugefügt worden.

Das Aufstellen und Verstehen der Anforderungen ist ein sehr wichtiger Schritt. Im schlimmsten Fall könnte es sein, dass der Kunde ein Produkt bekommt, dass er gar nicht wollte. Die Anforderungen sind eine Art Vertrag zwischen dem Kunden und dem Entwickler. Jede Partei kann sich auf den Inhalt der Anforderungsliste berufen. Die Anforderungen sollten nach Möglichkeit so formuliert werden, dass diese quantifiziert werden können, um zu überprüfen, ob ein Requirement erfolgreich umgesetzt wurde. Die Anforderungen sind in Kategorien eingeteilt, damit der Fokus der Entwicklung auf die wichtigen Requirements gerichtet ist. Es ist wichtig zu wissen, ob eine Anforderung Wunsch oder Pflicht ist [Wei14].

Tabelle 2.2.: Requirements

Req.	Beschreibung	Quelle	Kategorie
1.	Der MAN soll auf dem FZG montierbar sein.	Kunde	must
2.	Der MAN soll das FZG befähigen eine RT zu bewegen.	Kunde	must
2.1.	Das GS muss sich vor und zurück, nach rechts und links bewegen, sowie Drehungen ausführen können.	Kunde Entwickler	must
3.	Die Verbindung des MAN zur RT soll im Betrieb hergestellt bzw. gelöst werden können.	Kunde	must
3.1.	Der Endeffektor soll sich von -95° bis +95° selbständig drehen können (vgl. Abbildung 3.13).	Entwickler	must
3.2.	Der Endeffektor soll sich um mindestens 40mm selbständig heben und senken können.	Entwickler	must
3.2.1.	An dem MAN sollen Schutzmaßnahmen zur Vermeidung von Verletzungen an Extremitäten installiert sein.	Kunde	must
3.3.	Der MAN muss die Verbindung zur RT aus einer mit Toleranzen behafteten Position aus herstellen können (siehe Gleichung 3.1).	Kunde	must
4.	Zwischen dem MAN und dem FZG sollen Daten über eine serielle 5 V Schnittstelle mit TX und RX ausgetauscht werden.	Studenten	should
4.1.	Der MAN soll Statusinformationen als Datenstring in einem bestimmtem Protokoll senden (siehe Abbildung 3.33).	Kunde Entwickler	must
4.2.	Der MAN soll Kommandos in einem bestimmten Protokoll empfangen können (siehe Abschnitt 3.4.2).	Kunde Entwickler	must
4.3.	Der MAN soll bei einem ungültigen Kommando einen Fehlercode über die Statusinformation ausgeben (siehe Abschnitt 3.4.2.).	Kunde	shall
5.	Das GS soll sich nur im 7. Stock des HAW Gebäudes Berliner Tor 7 bewegen. Die Bewegung soll sich dabei auf barrierefreie Bereiche beschränken (siehe hierzu Barrierefreiheit im Glossar).	Kunde	shall
5.1.	Der MAN soll seine Funktionalität auch bei Kollisionen mit der Umwelt beibehalten.	Kunde	must
6.	Der MAN soll nur für eine bestimmte RT entwickelt werden. Diese ist an der Querstrebe mit einem Aufkleber „Projektgruppe“ gekennzeichnet.	Kunde	shall
7.	Das Drehmoment zwischen FZG und RT soll entkoppelt sein.	Kunde	must

2.6. Systemanalyse

Bei jeder Problemstellung erscheinen unweigerlich Lösungsvarianten vor dem geistigen Auge, welche sich nach kurzer Zeit eingefurcht haben. Der Blick für innovative und evtl. sogar bessere Lösungswege kann dadurch eingeschränkt werden. Es ist hilfreich den Sachverhalt zu abstrahieren und das zu entwickelnde System ganz allgemein zu betrachten.

2.6.1. Systemakteure und Objektflüsse

Die Systems Modeling Language (SysML) bietet zur grafischen Darstellung von Systemkontexten das sogenannte block definition diagram (bdd), wie in Abbildung 2.5 an.

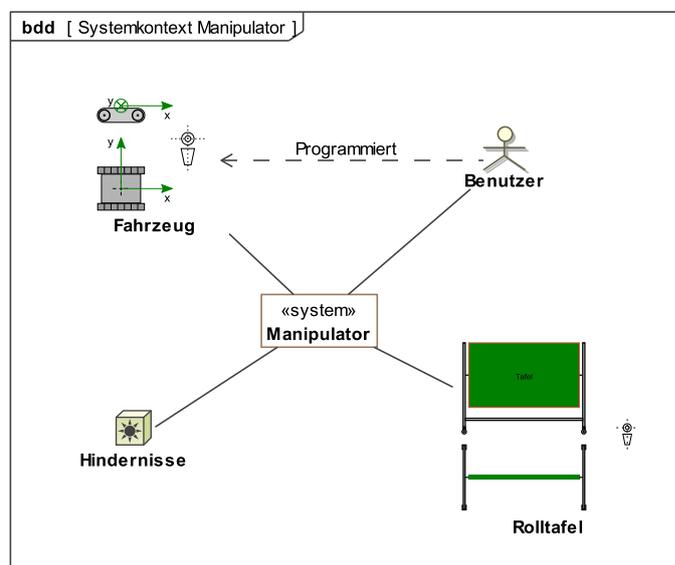


Abbildung 2.5.: Systemkontext Manipulator

Die Modellierung des Systemkontextes ist in jedem Projekt Pflicht, selbst wenn dies trivial erscheint. In dem bdd werden die Systemakteure bzw. Umwelt identifiziert und somit die Systemgrenzen abgesteckt [Wei14]. In diesem Fall besteht die Umwelt des MAN aus dem FZG, das von einem Benutzer programmiert wird, einer RT und Hindernissen.

Zur Darstellung der Objektflüsse bietet ein SysML das internal block diagram (ibd), wie in Abbildung 2.6, an.

Dieses Diagramm hat nicht den Anspruch vollständig zu sein, bietet aber einen guten Überblick über die Interaktionen zwischen den Akteuren. Es werden nur relevante Objektflüsse dargestellt. Es ist z. B. klar, dass Actio gleich Reactio bzw. jede Kraft eine entsprechende

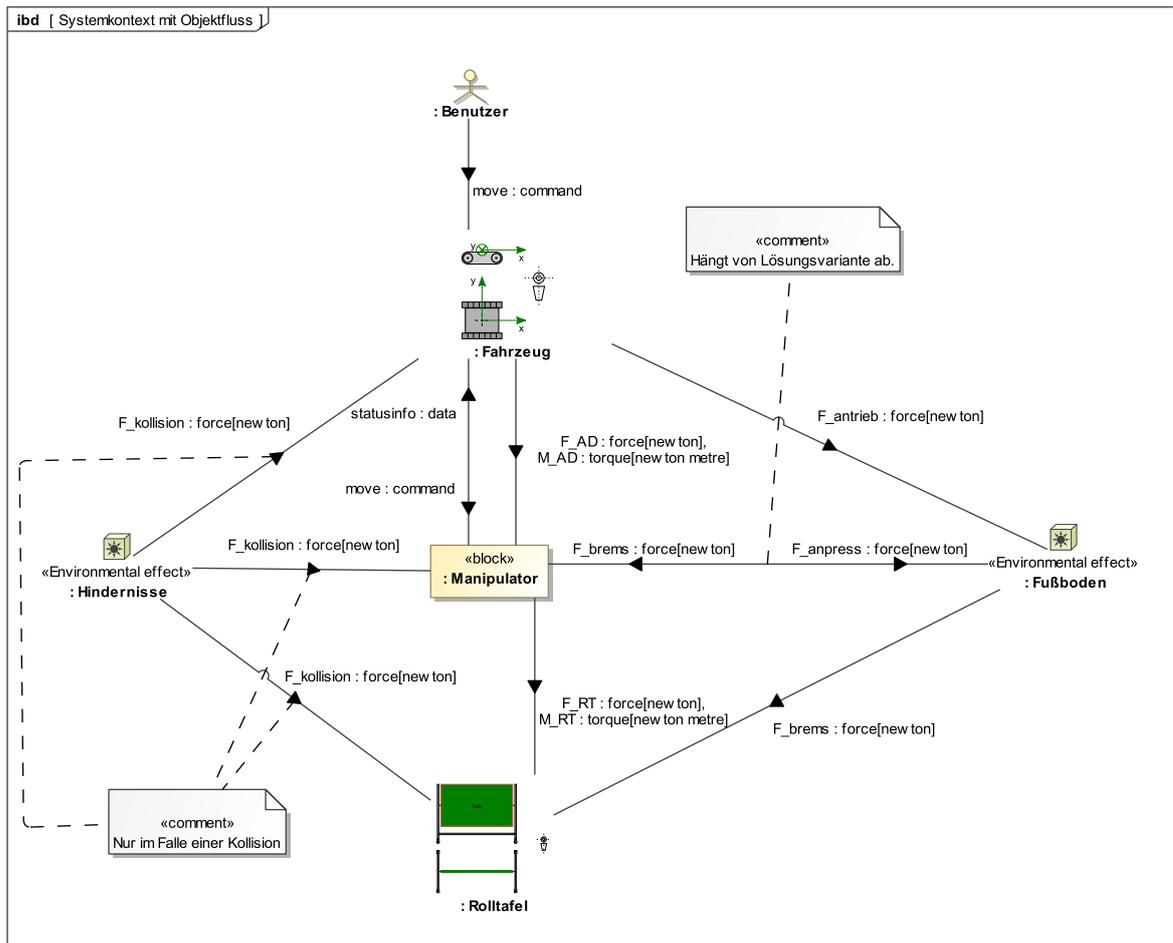


Abbildung 2.6.: Systemkontext mit Objektfluss

Gegenkraft hat. Das Eintragen dieser Kräfte hätte jedoch keinen Mehrwert. Überflüssige Informationen können und sollen außen vor bleiben.

Ein Benutzer programmiert das FZG, sodass dieses sich selbstständig fortbewegt. Die zu bewegendende RT und das FZG haben keine direkten Objektflüsse untereinander. Der MAN ist das Bindeglied zwischen dem FZG und der RT. Vom FZG zum MAN fließen Kräfte und Momente sowie Steuerkommandos. In die andere Richtung fließen Statusinformationen. Hindernisse können im Falle einer Kollision Kräfte auf das FZG, die RT und den MAN ausüben. Das FZG übt auf den Fußboden eine Kraft aus, um sich fortzubewegen. Ebenso übt der Fußboden Reibung auf die RT aus, welche diese beim Rollen bremst. Je nach Lösungsvariante kann es auch sein, dass der Fußboden eine Bremskraft auf den MAN ausübt und in umgekehrter Richtung der MAN auf den Fußboden einen Anpressdruck erzeugt. Vom MAN fließen lediglich Kräfte in Richtung RT.

2.6.2. Funktionsanalyse

Ein anderes Mittel der Abstraktion ist das Aufstellen einer Funktionstabelle wie in Tabelle 2.3. Der Aufbau der Tabelle hat sich an [Abu17] orientiert. Zum Teil fließen hier auch die Erkenntnisse aus Abschnitt 2.6.1 mit ein.

In der Tabelle 2.3 sind die Funktionsklassen der Haupt-, Unter-, Teil- und Teilfunktion der Unterfunktion dargestellt. Es werden zunächst die übergeordneten Hauptfunktionen aufgestellt. Jede Funktion wird in Unterfunktionen und diese wiederum in Teilfunktionen aufgespalten. Es sind hier nur die Funktionsarten der Gebrauchsfunktionen interessant, Geltungsfunktionen werden hier nicht berücksichtigt. Diese müssen möglichst allgemein bzw. lösungsneutral formuliert werden. Ein zu großer Abstraktionsgrad kann zum Abdriften der ursprünglichen Problemstellung führen, ein zu feiner führt wiederum zu Einschränkungen.

Für jede Funktion stellt sich die Frage nach dem „*Warum?*“. Die Hauptfunktionen ergeben sich aus den Anforderungen der Stakeholder. Die Frage nach dem „*Wie?*“ spaltet die Funktionen auf. Theoretisch ist eine Aufspaltung in noch feinere Ebenen möglich, in der Regel hat die vierte Ebene aber eine ausreichende Granularität.

Sofern ein morphologischer Kasten verwendet wird, ist es üblich die unterste Ebene der Funktionsaufspaltung heranzuziehen. Grundsätzlich ist die Aufstellung der Funktionsanalyse ein getrennter Schritt für sich, aber unter Umständen kann es sein, dass die unterste Ebene der Funktionsanalyse nicht für den morphologischen Kasten geeignet ist. Man stimmt die Aufspaltung und Formulierung der Unterfunktionen dahingehend ab, dass diese auch später im morphologischen Kasten verwendbar sind.

Tabelle 2.3.: Funktionsanalyse

Nr.	Hauptfunktion	Nr.	Unterfunktion	Nr.	Teilfunktion	Nr.	Teilfunktion der Unterfunktion	
F1	Zug-/Druckkraft an RT übertragen.	F1.1	Kraft-/Formschlüssige Verbindung erzeugen.	F1.1.1	Starre Verbindung zum FZG herstellen.		Verbindungsmechanismus heben/senken.	
				F1.1.2	Lösbare Verbindung zur RT herstellen.			Verbindungsmechanismus heben/senken.
				F1.1.3	Verbindungsmechanismus zur RT positionieren bzw. Fehlstellungen kompensieren.			Verbindungsmechanismus drehen.
		F1.2	Drehmoment zur RT entkoppeln.					
F2	Daten austauschen.	F2.1	Statusinformationen senden.					
		F2.2	Kommandos empfangen.					
F3	Traktion von FZG erhöhen.	F3.1	Anpressdruck auf FZG erzeugen.					

3. Design

Bei der Entwicklung eines Produktes zahlt sich die Investition von Zeit in eine gute Planung bei der Implementierung wieder aus und kann bis zu 60 - 70% des Arbeitsaufwandes ausmachen. Im Gegensatz hierzu wird bei der sogenannten „Basterei“ wenig Zeit bei der Planung investiert aber viel Zeit bei der Implementierung benötigt. In der Summe spart ein gutes Design Zeit und Geld. Es ist aber zu erwähnen, dass jedes noch so gute Design nicht alle Fallstricke vorhersehen kann und bei der Implementierung Entscheidungen getroffen werden müssen, die nicht geplant waren, es liegt ein iterativer Prozess vor. Ein gutes Design zeichnet sich durch das Prinzip Keep it simple and stupid (KISS) aus. Am Anfang steht eine Idee bzw. Konzept, das noch recht einfach ist. Bei der detaillierten Ausarbeitung wird das Design immer komplexer und komplizierter. Die „Basterei“ bleibt stehen, sobald sich nach außen das gewünschte Verhalten eingestellt hat. Die Konstruktion tut zwar, was sie soll, ist aber auf ihr Funktionsprinzip hin schwer zu verstehen und zu erweitern, schlecht zu warten usw. Die Konstruktion wirkt wie eine Art „Frankenstein“, dem überall dort, wo es nützlich erscheint, Module oder Bauteile angenäht wurden. Ein gutes Design bleibt hier nicht stehen, sondern versucht die Konstruktion auf ihren Kern zu reduzieren. Dadurch nimmt die Komplexität wieder ab und die Durchschaubarkeit zu. Schaut man als Außenstehender das erste Mal auf ein funktionierendes Produkt und befindet den Aufbau und die Funktionalität für geradezu primitiv, so ist dies positiv zu bewerten. Man kann davon ausgehen, dass dies einem guten Design zu verdanken ist.

Die nächsten Abschnitte entsprechen grob dem der Entwicklungsreihenfolge. Diese ist hier sinnvoll aber nicht zwingend für ein mechatronisches Design. Letztendlich wird sich in kaum einem Entwicklungsprozess eine strikte Reihenfolge einhalten lassen, bei welcher der zweite Schritt erst beginnt, wenn der Erste abgeschlossen ist [Wei14]. Den Grundstock des Designs legt in dieser Arbeit zwar die Mechanik, aber auch hier tauchen schnell Querverbindungen zur Elektronik und Software auf, die sich gegenseitig beeinflussen. Manche Schritte laufen also nicht nacheinander, sondern parallel und iterativ ab. Die inhaltliche Abgrenzung der Abschnitte hat zwangsläufig eine Unschärfe. Das Design versucht hierbei stets auf die aus Abschnitt 2 beschriebenen Randbedingungen einzugehen bzw. ist durch diese limitiert.

3.1. Konzeptbildung

Für die Konzeptbildung werden aus einem morphologischen Kasten drei Konzepte extrahiert und anschließend für eine finale Auswahl bewertet. Was in den morphologischen Kasten eingetragen wird, ist keinesfalls bindend und kann durchaus auch zu einem gewissen Grad „absurde“ Ideen enthalten, um den Blickwinkel zu öffnen.

3.1.1. Lösungsfindung und Konzeptdefinition

Tabelle 3.1 zeigt den morphologischen Kasten mit den Lösungsvarianten zur Entwicklung von Konzepten nach [Abu17]. Die erste Spalte beinhaltet die Funktionen, welche aus der Funktionsanalyse übernommen wurden. Für jede Funktion sind fünf Lösungsvarianten vorgestellt. Ein Konzept setzt sich aus einer Kombination der Funktionsvarianten zusammen. In dem morphologischen Kasten wird dies durch einen Pfad gekennzeichnet. Grundsätzlich hat dieser Pfad Zugang zu allen Varianten, jedoch ist nicht jede Kombination sinnvoll bzw. miteinander verträglich [Nae09, S.91]. Die Abbildungen der drei vorgestellten Konzepte sollen das Zusammenspiel der Lösungsvarianten illustrieren. Die Einfärbung der Bauteile soll lediglich eine optische Trennung erzeugen und hat keine weitere Bedeutung.

Tabelle 3.1.: Morphologischer Kasten

Funktionen	Variante 1	Variante 2	Variante 3	Variante 4	Variante 5
Starre Verbindung zum FZG herstellen	Verschraubung	Kleben	Schweißen	Klemmen	Nieten
Lösbare Verbindung zur RT herstellen	Normalkraft durch Linearaktuator	Scherengreifer	Saugnapf	Gefrierklebung	Lineargreifer
Verbindungsmechanismus heben/senken	Seilzug	Spindelantrieb / Linearmotor	Servomotor	Hydraulikzylinder	Pneumatikzylinder
Verbindungsmechanismus drehen	Servomotor	Riemenantrieb	Schrittmotor	Linearaktuator	Riemenantrieb
Drehmoment zur RT entkoppeln	Regelung gleicht Moment durch Drehung aus.	Spannen/Entspannen des Drehriemens	Mitdrehender Motor	Kupplungsscheiben auseinanderziehen/zusammendrücken	Trennung des Antriebs durch verschieben von Zahnrädern
Statusinformationen senden	SPI	I2C	Seriell	CAN Bus	Parallel
Kommandos empfangen	SPI	I2C	Seriell	CAN Bus	Parallel
Reibkraft zum Bodenbelag erhöhen	Normalkraft durch Hubservo	Normalkraft durch Eigengewicht des MAN	Normalkraft durch Schwerpunktverlagerung	Normalkraft durch Eigengewicht der RT	Beheizen der Antriebsriemen

Konzept 1 Konzept 2 Konzept 3

Konzept 1

Das erste Konzept ist in Abbildung 3.1 illustriert. Zwei um 90° verdrehte Servomotoren werden übereinander auf das FZG angeklebt. Der untere Servomotor dreht den oberen in der horizontalen Ebene. An dem oberen Servomotor ist ein Hubarm mit Dehnungsmessstreifen befestigt, welcher sich in der vertikalen Ebene bewegen kann. Auf dem Hubarm befindet sich ein Linearmotor, der über zwei Bautenzüge zwei Anpressbacken an dem Hubarm ziehen kann.

Der untere Servomotor positioniert den Hubarm zur RT. Anschließend senkt der obere Servomotor den Hubarm, so dass sich die Querstrebe der RT zwischen dem Hubarm und den Anpressbacken befindet. Nun zieht der Linearmotor die Anpressbacken über die Bautenzüge an. Zwischen dem Hubarm und der Querstrebe besteht nun eine kraft- und formschlüssige Verbindung. Damit das FZG über genügend Traktion zum Ziehen der RT verfügt, wird der obere Servomotor so angesteuert, dass der Hubarm leicht gehoben wird. Dies erzeugt einen Anpressdruck auf das FZG zum Fußboden. Zur Drehmomentenkopplung detektieren die Dehnungsmessstreifen geringe Verbiegungen des Hubarmes. Über eine Regelung wird der Drehservo so angesteuert, dass er der Verbiegung entgegenwirkt bzw. diese aufhebt. Von außen wirkt es, als ob der Drehmechanismus frei gelagert ist.

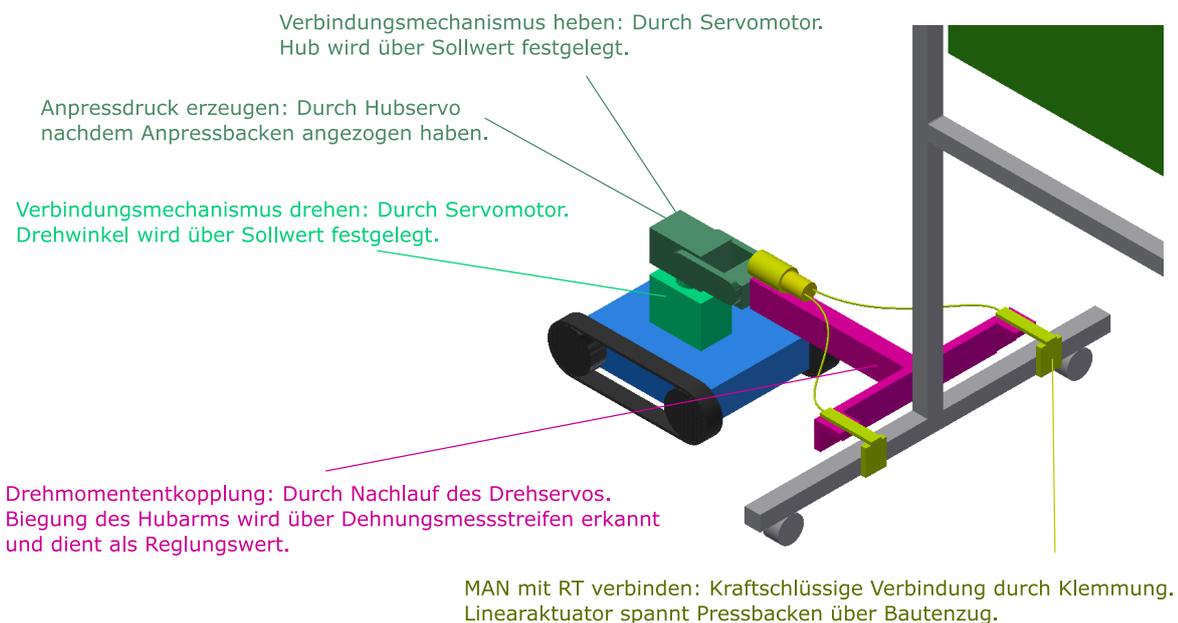


Abbildung 3.1.: Konzept 1

Konzept 2

Das zweite Konzept ist in Abbildung 3.2 illustriert. Ein Elektromotor treibt über einen Riemen eine größere Riemenscheibe an. Der Riemen läuft über einen schaltbaren Riemenspanner. An der großen Riemenscheibe befindet sich ein Drehencoder. Des Weiteren ist an der großen Riemenscheibe ein Ausleger mit eigenen Fußbodenrollen befestigt. Der Ausleger ist in der Mitte über ein Scharnier klappbar. Auf dem Ausleger sind zwei Linearantriebe montiert. Der hintere Linearantrieb kann den Ausleger ein- bzw. ausklappen, der vordere hebt bzw. senkt eine Hubgabel.

Durch den Elektromotor wird der Ausleger zur RT positioniert. Die Encoderscheibe liefert die hierfür benötigten Positionsdaten. Anschließend wird der Riemen entspannt, sodass sich der Ausleger frei drehen kann. Der vordere Linearantrieb hebt die RT mit der Hubgabel an der Querstrebe an. Um genügend Traktion für das FZG zu erzeugen, wird der Schwerpunkt zum FZG hin verlagert, indem der hintere Linearantrieb den Ausleger einklappt. Zwischen der Hubgabel und der Querstrebe besteht nun eine kraft- und formschlüssige Verbindung.

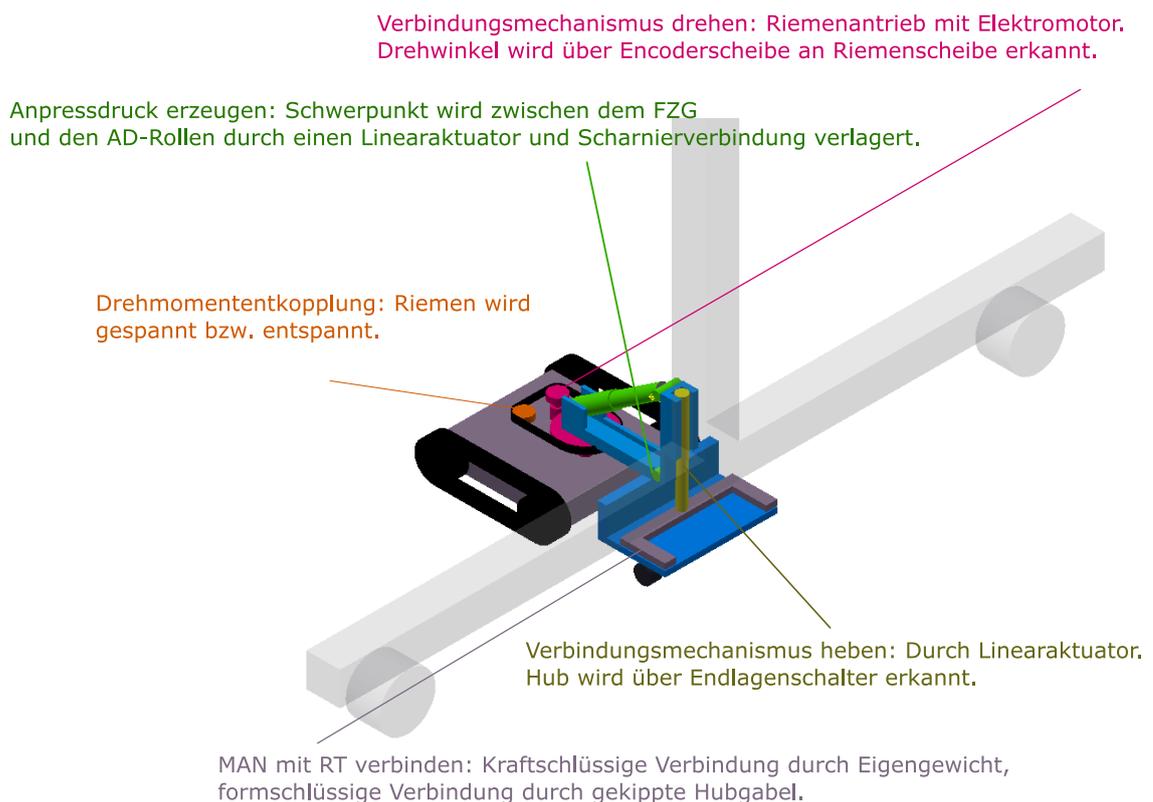


Abbildung 3.2.: Konzept 2

Konzept 3

Das dritte Konzept ist in Abbildung 3.3 illustriert. Auf dem FZG ist ein Schrittmotor mit Magnetsensor zur Winkelbestimmung festgeschraubt. Der Schrittmotor dreht einen Hubarm auf dem ein Linearantrieb befestigt ist. Der Linearantrieb hebt eine Anpressfläche mit Führungszangen an. Der Hub ist durch den Hubarm geführt.

Der Schrittmotor positioniert den Hubarm zur RT. Der Linearantrieb presst die Anpressfläche und die Führungszange gegen die Längsstrebe der RT. Durch den Anpressdruck wird die Traktion des FZG erhöht. Die Führungszangen können Positionierfehler durch ihre Formgebung ausgleichen und sorgen dafür, dass die Längsstrebe nicht zur Seite hin abrutschen kann. Zwischen den Führungszangen und der Längsstrebe besteht nun eine kraft- und formschlüssige Verbindung.

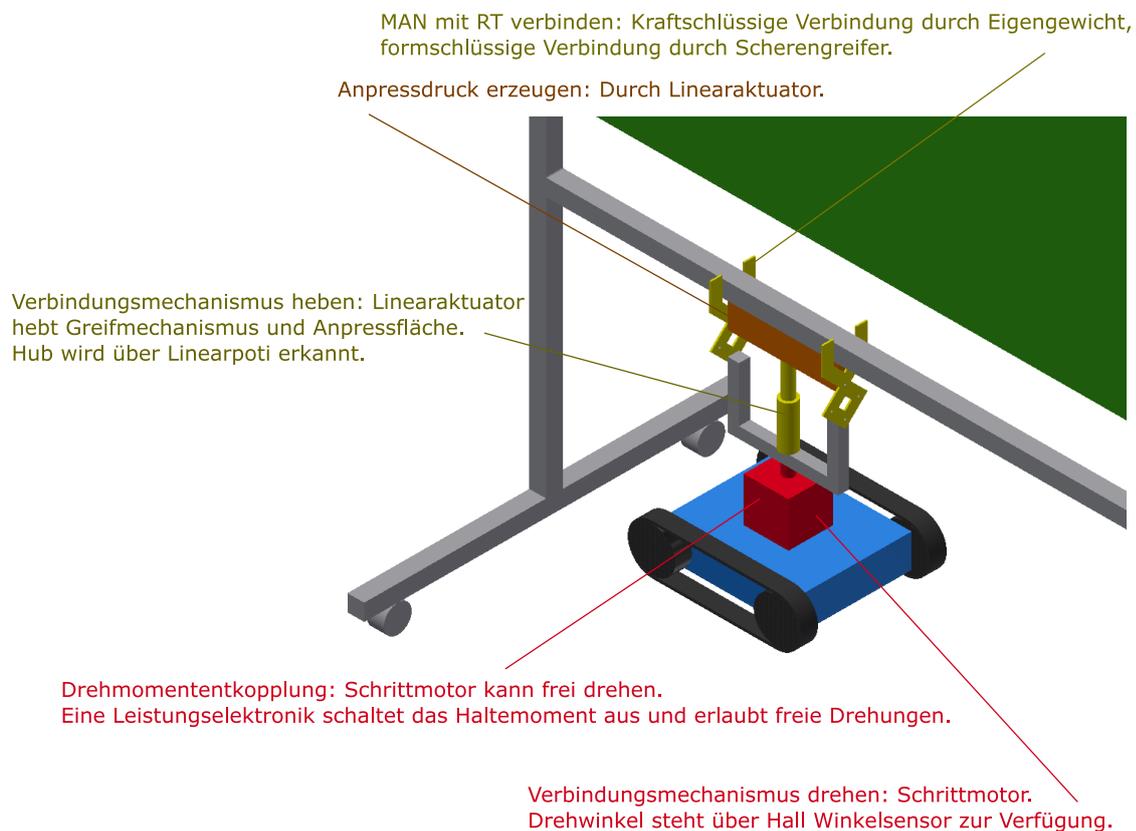


Abbildung 3.3.: Konzept 3

3.1.2. Konzeptbewertung

Die vorgestellten Konzepte werden nun systematisch miteinander verglichen, um zu einer finalen Auswahl zu gelangen. Die Bewertungskriterien sind in Tabelle 3.2 dargestellt. Der Aufbau von Tabelle 3.2 ist an [Abu17] und [Nae09] angelehnt.

Tabelle 3.2.: Konzeptbewertung

Kriterium	Gewichtungs- faktor	Konzept 1		Konzept 2		Konzept 3	
		Punkte	Einzelnutz- werte	Punkte	Einzelnutz- werte	Punkte	Einzelnutz- werte
Konstruktionsaufwand	10	5	50	4	40	5	50
Fertigungsaufwand	20	4	80	3	60	5	100
Installationsaufwand	20	4	80	4	80	5	100
Störanfälligkeit	10	4	40	4	40	5	50
Instandhaltung	10	4	40	5	50	5	50
Energieverbrauch	20	1	20	7	140	8	160
Kostenabschätzung	5	5	25	4	20	5	25
Gewicht	5	5	25	4	20	5	25
Gesamtnutzwert	100		360		450		560

Punkte geben den Erfüllungsgrad auf einer Skala von 0 bis 10 an (0 = schlecht, 10 = gut)

Die einzelnen Kriterien werden mit Gewichtungsfaktoren versehen. Als besonders wichtig werden also die Faktoren Fertigung- und Installationsaufwand sowie der Energieverbrauch erachtet, da es ein klares Ziel dieser Bachelorarbeit ist das Produkt nicht nur theoretisch zu entwickeln, sondern auch einen funktionsfähigen Prototyp zu implementieren. Die Summe aller Gewichtungsfaktoren ergibt immer 100.

Bei jedem Konzept werden entsprechend dem Kriterium Punkte auf einer Skala von 0 bis 10 vergeben, wobei 0 eine schlechte und 10 eine gute Erfüllung des Kriteriums bedeutet. Die vergebenen Punkte werden mit dem entsprechenden Gewichtungsfaktor multipliziert und ergeben dann den Einzelnutzwert. Je höher der Gesamtnutzwert ist, desto besser ist das Konzept in der Gesamtwertung. Das **Konzept 3** hat den größten Gesamtnutzwert und ist somit das Konzept, welches für die Detailkonstruktion verwendet wird.

Welche Kriterien verwendet werden und wie die Gewichtungsfaktoren und Punktevergabe ausfällt, hängt von der subjektiven Einschätzung des Entwicklungsingenieurs ab. Je mehr Entwicklungserfahrung vorhanden ist, desto besser wird die Konzeptbewertung ausfallen.

3.2. Mechanik

Der mechanische Aufbau wurde mit dem CAD Zeichenprogramm INVENTOR PROFESSIONAL 2018 designed. Die Arbeit mit INVENTOR hat es ermöglicht die Konstruktion im Vorfeld auf Maße und mechanische Funktionalität zu evaluieren, sowie überschlägige Aussagen zur Festigkeit zu treffen. Bei der Zusammensetzung der Bauteile zu Baugruppen werden Abhängigkeiten wie starre Verbindungen, Gelenke, Winkel, Abstände, Gleiter usw. vergeben. Dadurch kann die Funktionalität auf sein logisches, jedoch nicht auf sein tatsächliches physikalisches Verhalten überprüft werden. Auf der beigelegten CD im Anhang befindet sich eine Videoanimation im Verzeichnis CD:/Videos/Animation.mp4. Man muss sich darüber im Klaren sein, dass dies ideale Abhängigkeiten sind. Ein starre Verbindung oder ein quer belastetes Lager sind z. B. unendlich fest, genauso wie die Bauteile. Eine Verbiegung bei Belastung wird nicht dargestellt. Eine dynamische Simulation aller Bauteile im Zusammenspiel ist zwar denkbar, aber für diese Arbeit viel zu aufwendig und würde den zeitlichen Rahmen sprengen. Abbildung 3.4 zeigt den fertigen Entwurf des MAN auf dem FZG montiert.

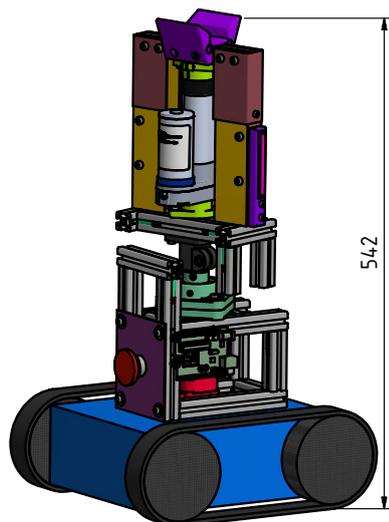


Abbildung 3.4.: Zusammenbau Manipulator mit Fahrzeug

Die folgenden Abschnitte beschreiben die einzelnen Entwicklungsschritte für diesen Zusammenbau.

3.2.1. Konstruktion

Dieser Abschnitt widmet sich der Beschreibung der CAD-Zeichnungen. Hier wird die Position und Dimensionierung der Bauteile erläutert. In erster Linie handelt es sich, bis auf wenige

Ausnahmen, um Kaufteile. Es werden lediglich die Längen der Zuschnitte von Bauteilen, die Meterware sind, angegeben. Weitere Händlerinformationen sind in der Stückliste verlinkt. Auf maximale Außenmaße der Baugruppen wird verzichtet, da sich diese durch die bereits vorhanden Angaben schnell erschließen lassen und die Abbildungen dadurch nicht überladen werden. Es handelt sich bei der mechanischen Konstruktion also weitestgehend um ein Assembly Design. Alle Bauteile sind als CAD-Daten im Anhang im Verzeichnis CD:/CAD/ enthalten.

Das übergeordnete Leitmaß ist der Abstand vom Fußboden zur Längsstrebe der RT (siehe Abbildung 3.5) und beträgt 55 cm. Die Dachplatte des FZG befindet sich 10 cm über dem Fußboden. Diese Platte bildet das Fundament auf dem der MAN aufgebaut wird.

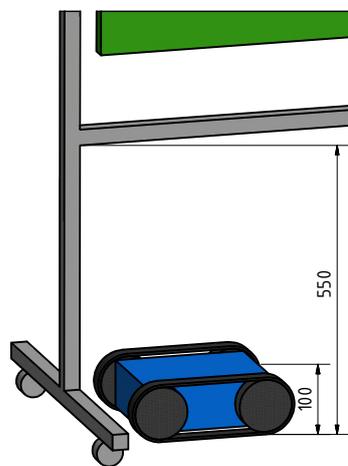


Abbildung 3.5.: Fahrzeug unter Rolltafel

Für den Rahmenaufbau werden hauptsächlich $20\text{ mm} \times 20\text{ mm}$ Strangpressprofile aus Aluminium mit Nut5 aus der Industrie verwendet [ite17]. Umgangssprachlich werden diese auch gerne als „Fischertechnik für Große“ bezeichnet, da sie äußerst flexibel einsetzbar sind. Als Konstruktionsrichtlinie gilt es, die formschlüssigen den kraftschlüssigen Verbindungen vorzuziehen. Bei den hier auftretenden Belastungen spielt dies jedoch keine Rolle. Jegliche Verbindung, die über die T-Matik Verbinder hergestellt wird, erzeugt eine derart große Flächenpressung, dass die so erzeugte Verbindung als ausreichend steif angesehen werden kann. Voraussetzung hierfür ist, dass die Zuschnitte senkrecht zum Profil bzw. plan sind.

Alle Schrauben, die in der Profilnut befestigt sind, benötigen einen entsprechenden Nutenstein. Eine M3 Schraube benötigt z. B. einen M3 Nutenstein. Die Nutensteine sind in den folgenden Abbildungen nicht immer zu erkennen bzw. werden nicht immer explizit erwähnt.

Der Zusammenbau aus Abbildung 3.4 ist eine Baugruppe, die aus den Unterbaugruppen

Untergestell mit Rotationsantrieb zur Ausrichtung, dem rotierenden Hubarm mit Hubführung und dem Endeffektor mit Führungsschale zusammengesetzt ist.

Untergestell

Die Abbildungen 3.6, 3.7 und 3.8 zeigen das komplette Untergestell, welches auf die Dachplatte des FZG montiert wird, aus verschiedenen isometrischen Ansichten. Das Untergestell nimmt Kräfte und Momente vom FZG über eine Verschraubung auf. Es dient als Träger für die Elektronik und den Antriebstrang (vgl. Abbildung 3.9).

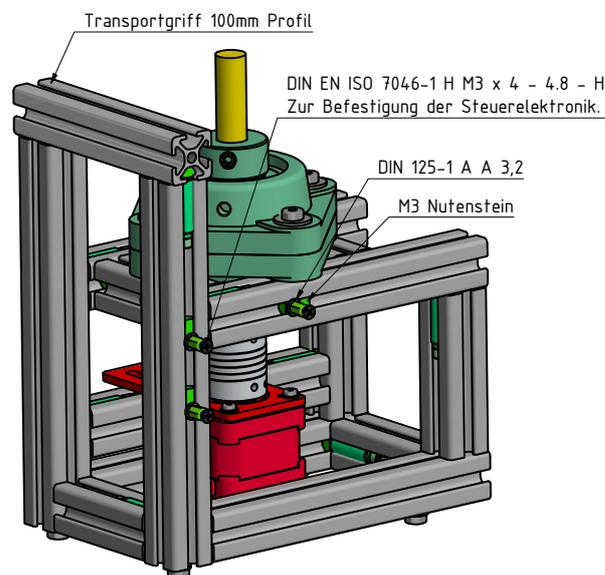


Abbildung 3.6.: Untergestell isometrisch vorne links

Die Länge der Profile ist an die Abmaße des Antriebsstrangs (siehe Abbildung 3.9) angepasst. Das 100 mm Profil in Abbildung 3.6 dient zum einen als mechanischer Anschlag für den Hubarm und zum anderen eignet es sich als Haltegriff, um den MAN mit inklusive FZG zu transportieren.

Die DIN EN ISO 7046-1 H M3 x 4 Schrauben mit entsprechenden Nutensteinen und Unterlegscheiben dienen zur Befestigung der Steuerungselektronik, die im Abschnitt 3.3.3 genauer erläutert wird.

Die 56 mm Profile in Abbildung 3.7 flankieren das Flanschlager. Die Länge entspricht dem Durchmesser des Flanschlagers ober- bzw. unterhalb der Haltebohrungen. Das Lager muss mit 45° Versatz eingesetzt werden, da der Abstand der Haltebohrungen ansonsten zu klein und somit nicht über der Profilvernut liegen würden.

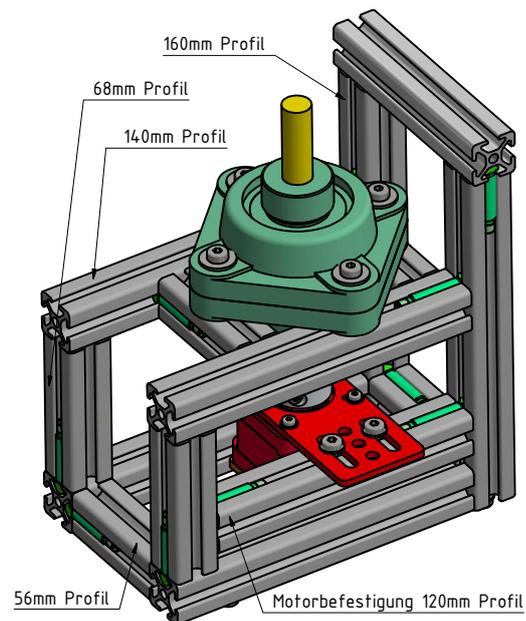


Abbildung 3.7.: Untergestell isometrisch hinten rechts

Alle T-Matik Verbinder benötigen einen M4 Nutenstein. Abbildung 3.8 zeigt wie eine solche Stoßverbindung aufgebaut ist. Das 160 mm Profil ist zum einen für eine ausreichende Höhe des Drehanschlags da, zum anderen ist in der Profilbohrung (siehe Abbildung 3.8) ein M5 Gewinde geschnitten, welches eine M5 Schraube mit Unterlegscheibe hält, die durch das Dachblech des FZG geht. Die anderen M5 Schrauben, die durch das Dachblech des FZG gehen sind über Nutensteine in den 140 mm Profilen verschraubt.

Die 56 mm und 140 mm Profile liegen zudem als eine Art Rahmen auf dem Dachblech auf, sodass sich aufkommende Lasten gleichmäßig auf dem Blech verteilen. Der Antriebsstrang (vgl. Abbildung 3.9) beginnt von unten mit der Printed Circuit Board (PCB) Halterung für den Drehwinkelsensor, die an den Schrittmotor geklebt ist. An den Motor ist ein Halterungsblech, zur Befestigung an das 120 mm Profil aus Abbildung 3.7, angeflanscht. Das Profil ist in seiner vertikalen Positionierung variabel. Die $\varnothing 5$ mm Welle des Schrittmotors wird über eine Flexkupplung mit einer $\varnothing 12$ mm Welle verbunden. Sollte der Motor nicht exakt konzentrisch zur $\varnothing 12$ mm Welle montiert werden, so wird dies von der Flexkupplung ausgeglichen. Die $\varnothing 12$ mm Welle wird mit vier Madenschrauben in den beiden Flanschlagern eingeklemmt.

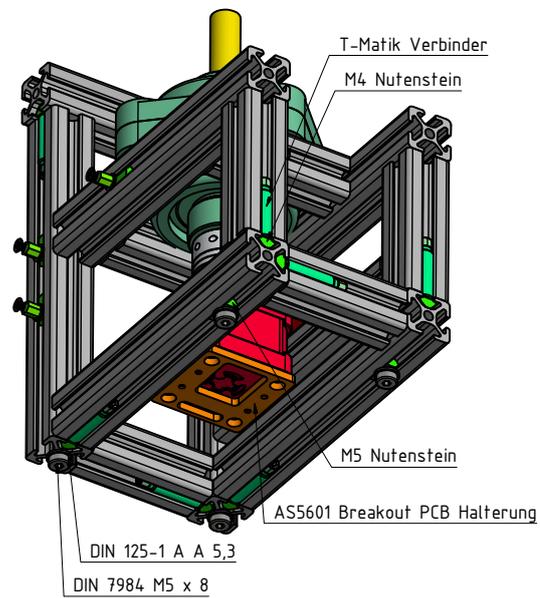


Abbildung 3.8.: Untergestell isometrisch unten links

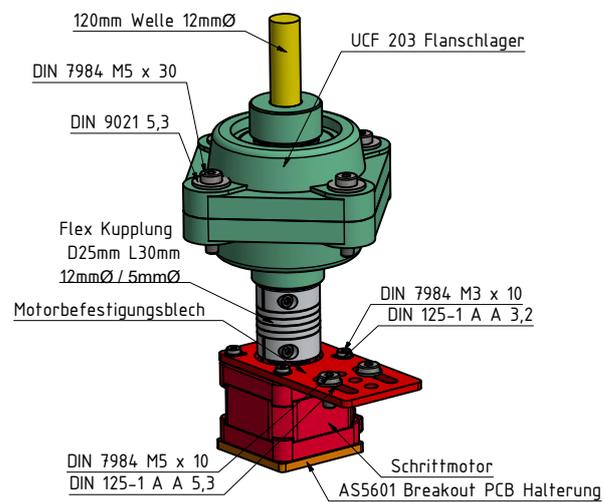


Abbildung 3.9.: Antriebsstrang

Hubarm

Der Hubarm hat die Aufgabe den Endeffektor durch Rotation auszurichten und dient als Führung für die Hubfunktion.

Der Hubarm (siehe Abbildung 3.10 und 3.11) wird über einen Ganter mit einer $\varnothing 12$ mm Aufnahme auf die $\varnothing 12$ mm Welle des Untergestells aufgesteckt und festgeklemmt. Die Welle muss nicht ganz im Ganter versenkt werden. Dadurch ist die Höhe des Hubarms variabel.

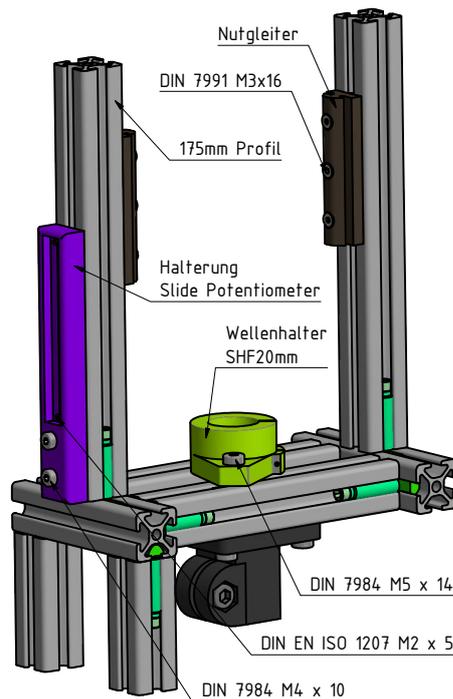


Abbildung 3.10.: Hubarm isometrisch hinten rechts

Die Bohrungen des Ganter sind maßgebend für den Abstand der 120 mm Profile Abbildung 3.11. Die 175 mm Profile sind der Höhe des später eingesetzten Linearmotors angepasst.

Die zwei Nutgleiter sind 3D Druckteile aus PETG und mit jeweils drei DIN 7991 M3 x 16 Schrauben in der Profilvernut verschraubt. Die Schrauben wurden jedoch nicht selbst gedruckt, sondern eingekauft. Zusammen mit den Nuten des Endeffektors bildet dies eine Linearführung, durch die der Endeffektor seine Hubbewegung gerichtet ausführen kann. Der Wellenhalter hat eine $\varnothing 20$ mm Aufnahme und wird zur Befestigung des Linearmotors verwendet. Die Halterung des Schiebepotentiometers ist ein 3D Druckteil und wird mit zwei DIN 7984 M4 x 10 Schrauben in der Profilvernut angeschraubt.

Die 56 mm Profile dienen als Rotationsanschläge, wobei die 80 mm Profile dafür sorgen, dass diese weit genug auseinander sind, damit die Drehbewegung des Hubarms ausreichend eingeschränkt wird (siehe Abbildung 3.11). Damit soll verhindert werden, dass sich die Anschlusskabel des Schiebepotentiometers und des Linearmotors um die Drehachse aufwickeln.

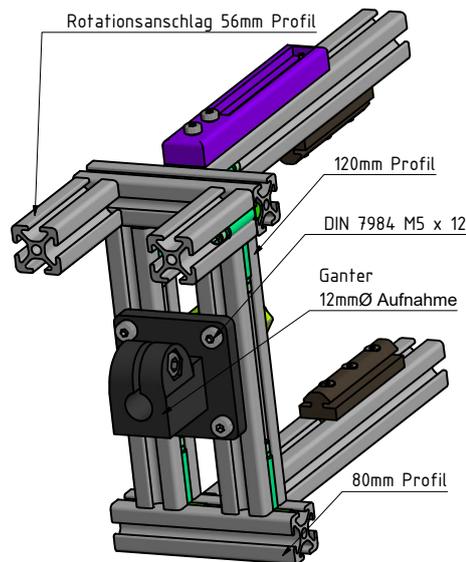


Abbildung 3.11.: Hubarm isometrisch unten rechts

Endeffektor

Abbildung 3.12 zeigt den Endeffektor, also das Bauteil, das in direktem Kontakt mit der RT steht. Er gleicht Positionierungsfehler aus und überträgt Kräfte auf die RT.

Der hier angeschraubte Wellenhalter ist identisch mit dem des Hubarms. Die 175 mm Profile, in welche die Nutgleiter des Hubarms geschoben werden, liegen hier fast stramm an. Der Wellenhalter ist somit maßgebend für das 160 mm Querprofil, das bündig mit dem Hubarm abschließt. Die Länge der 120 mm Profile des Hubarms ist hiervon auch beeinflusst.

Das wichtigste Bauteil des Endeffektors ist die Führungsschale, die im 3D Druckverfahren hergestellt wurde. Sie ist ein Hohldruck mit 1 mm Wandung, welche mit Acrylharz über zwei 10 mm Bohrungen ausgegossen wurde. Dadurch konnte sie stabiler und sehr viel kostengünstiger, als ein Volldruck, hergestellt werden. Das Druckmaterial CLEAR ist zwar eines der günstigeren Rohmaterialien des FORM2 3D Stereolithografiedruckers, aber trotzdem immer noch verhältnismäßig teuer. Die Schale wird über vier DIN 7984 M5 x 12 Schrauben an das 160 mm Querprofil geschraubt.

Die geöffnete Formgebung sorgt dafür, dass Fehlstellungen während des Hubvorganges korrigiert werden. Dies ist möglich, da das Drehmoment der Hauptdrehachse am Untergestell entkoppelt ist bzw. muss der Schrittmotor in dem Moment abgeschaltet sein.

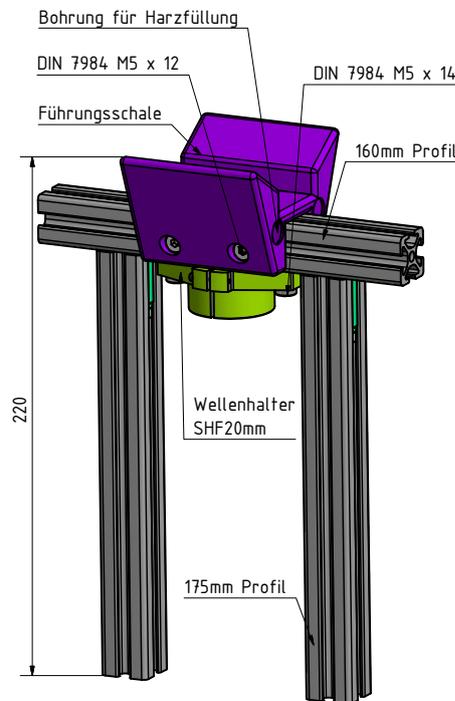


Abbildung 3.12.: Endeffektor isometrisch vorne rechts

Die Toleranz der Positionierung ist von zwei Parametern abhängig, nämlich der Verschiebung x und dem Verdrehwinkel ϕ . Gleichung 3.1 beschreibt, wie sich die Parameter gegenseitig beeinflussen. Wird die Gleichung verletzt, verliert die Führungsschale ihre Führungseigenschaften. Die Gleichung bezieht sich auf Abbildung 3.13 und ist nur damit verständlich.

$$\left| \frac{x}{22 \text{ mm}} \right| + \left| \frac{\phi}{10^\circ} \right| \stackrel{!}{=} 1 \quad (3.1)$$

Das Koordinatensystem ist auf der Führungsschale verankert, wobei die Z-Achse konzentrisch mit der Drehachse des Hubarms ist. Angenommen die Querstrebe der RT befindet sich genau mittig über der Z-Achse $x = 0$, dann darf die Querstrebe vom Betrag um nicht mehr als $|\phi| \leq 10^\circ$ verdreht sein. Sollte die Querstrebe exakt parallel $\phi = 0$ zur Y-Achse stehen, so darf der Betrag der Verschiebung nicht größer als $|x| \leq 22 \text{ mm}$ sein. Je größer die Verschiebung wird, desto kleiner wird der maximal erlaubte Verdrehwinkel. Genauso wird die maximale Verschiebung immer kleiner, wenn der Verdrehwinkel größer wird.

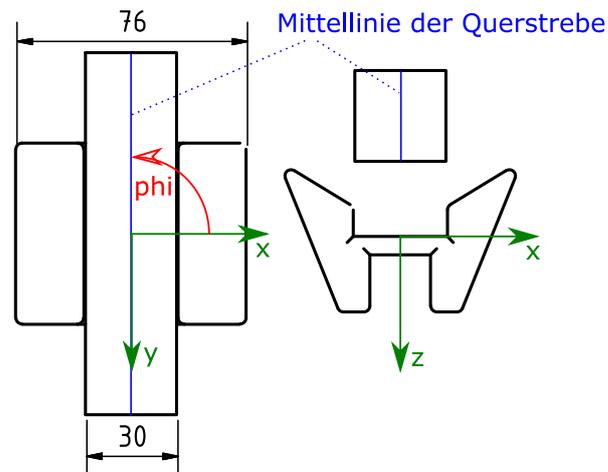


Abbildung 3.13.: Positionierungstoleranzen

3.2.2. Safety

Der Abschnitt 3.2.2 ist nicht als klassische Safetyanalyse zu verstehen, sondern behandelt Gefahrenquellen, die während der ersten Funktionstests aufgefallen sind und deren entsprechenden Schutzmaßnahmen.

In Abbildung 3.14 sind die Bereiche, welche großes Verletzungspotential in sich bergen, gekennzeichnet. Der Linearmotor, welcher noch in Abschnitt 3.3.1 vorgestellt wird, ist in der Lage 1500 N Zug- und Druckkraft auszuüben. Um zu verhindern, dass Extremitäten in die Klemmbereiche gelangen, werden Schutzbleche, wie in Abbildung 3.15 angebracht. Damit das Blech nicht an dem Endeffektor schleift, wird es mit DIN 9021 5,3 Unterlegscheiben abgesetzt. Ebenso das Schulterblech, welches nicht an dem darunterliegenden Blech schleifen soll. Eines der Schulterbleche wird zusätzlich als Mitnehmer für den Poti-Nutgleiter verwendet. Mehr hierzu in Abschnitt 3.3.2.

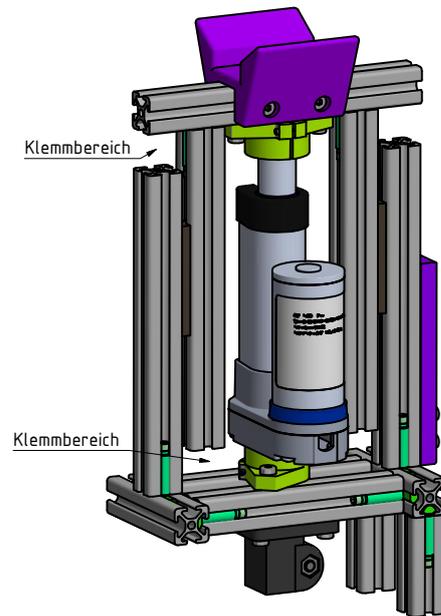


Abbildung 3.14.: Klemmbereiche

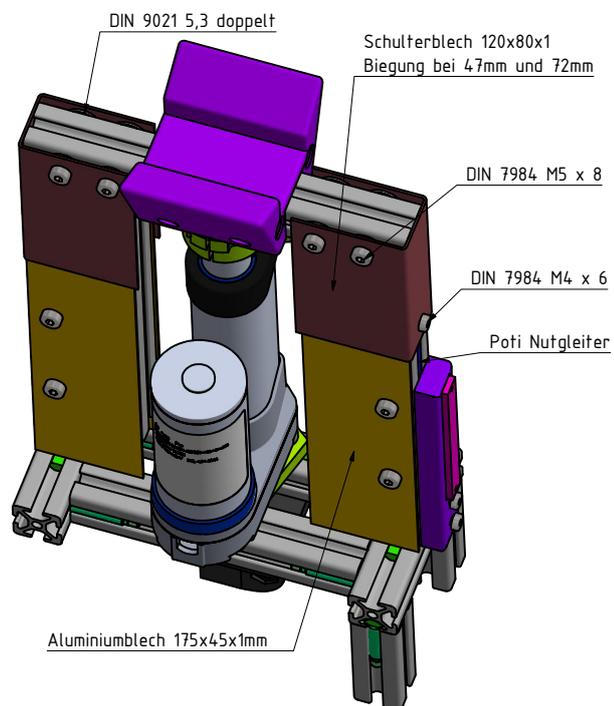


Abbildung 3.15.: Schutzbleche

3.2.3. Überschlägige Berechnungen von Kräften und Stabilität

Um im Vorfeld zu überprüfen, ob ein Bauteil versagt, macht es keinen Sinn jede einzelne Schraube zu berechnen. Stattdessen wird ein Versagenskriterium festgelegt. Dies ist mit dem schwächsten Glied in einer Kette zu vergleichen. Versagt das schwächste Glied, so reißt die ganze Kette und sie ist nicht mehr zu gebrauchen. Hält das schwächste Glied jedoch, so kann man davon ausgehen, dass die anderen Glieder auch halten werden und muss diese nicht zusätzlich überprüfen. Das Kriterium festzulegen ist nicht trivial. Hierfür müssen „*ingenieurmäßige Annahmen*“ getroffen werden, die das Kriterium oder auch mehrere Kriterien identifizieren. Die Herangehensweise ist konservativ. Das bedeutet, dass alle Annahmen so getroffen werden, dass das Ergebnis den stärksten Fall der Belastung oder auch „*worst case scenario*“ darstellt. Sofern eine Überdimensionierung nicht mit den Requirements in Konflikt steht, können diese Annahmen sehr großzügig ausfallen. Überlegungen und Berechnungen verkürzen und vereinfachen sich dadurch. Alle Annahmen werden also bewusst vereinfacht.

Für alle folgenden Kräftepfeile, die in den Abbildungen eingetragen sind, soll deren Länge nicht den vektoriellen Betrag, sondern nur die Wirkrichtung darstellen.

Schraubenberechnung

Es wird hier exemplarisch die DIN7991 M3 x 16 Schraube aus Abbildung 3.10 nach [Fis11, S.220] auf Versagen berechnet. Alle anderen Schrauben weisen eine höhere Festigkeit bzw. Dimensionierung auf. Die Festigkeitsklasse der Schraube ist 8.8, was einer Streckgrenze von $R_e = 640 \text{ N/mm}^2$ entspricht. Für die Montage ist keine Kontrolle des Anzugsmoment angedacht. Für das Anziehen von Hand liegen jedoch Erfahrungswerte aus [Fis11] für verschiedenen Gewinde vor: Vorspannung $\sigma_v = 350 \text{ N/mm}^2$, Vorspannkraft $F_v = 3000 \text{ N}$. Die Mindeststreckgrenze berechnet sich nach Gleichung 3.2.

$$R_{\text{erf}} \leq 1,5 \cdot \sigma_v \quad (3.2)$$

Demnach ist $R_e > R_{\text{erf}}$, die Mindeststreckgrenze wird also durch das Anziehen nicht überschritten. Der Spannungsquerschnitt für ein Regelgewinde berechnet sich nach Gleichung 3.3

$$S = \frac{\pi}{4} \left(\frac{d_2 + d_3}{2} \right)^2 = \frac{F_B}{\sigma_{zul}} \quad (3.3)$$

wobei F_B die Betriebskraft und σ_{zul} die maximal zulässige Spannung in der Schraube ist. Die Werte für die Durchmesser sind aus [Fis11, S.204] mit $d_2 = 2,68 \text{ mm}$ und $d_3 = 2,39 \text{ mm}$

zu entnehmen. Die maximal zulässige Spannung mit dem Sicherheitsfaktor $\nu = 2,5$ ergibt sich aus Gleichung 3.4

$$\sigma_{zul} = \frac{R_e}{\nu} \quad (3.4)$$

Nach Umstellung der Gleichungen kann die Betriebskraft nach Gleichung 3.5 berechnet werden.

$$F_B = \frac{\pi}{4} \left(\frac{d_2 + d_3}{2} \right)^2 \sigma_{zul} \quad (3.5)$$

$F_B = 1,29$ kN. Mit anderen Worten lässt sich sagen, dass man ca. 130 kg an die Schraube hängen kann, ohne dass diese abreißt.

Bei Schraubverbindungen ist es nicht vorgesehen die Schraube selbst einer Querkraft auszusetzen. Schrauben werden immer nur auf Zug belastet. Dies wird durch eine ausreichende Flächenpressung erreicht. Je fester man zwei Flächen aneinander presst, desto schwieriger ist es diese Flächen aneinander vorbei gleiten zu lassen. Die Reibkraft, die dem Aneinandergleiten entgegen wirkt ergibt sich nach Gleichung 3.6

$$F_{reib} = \mu \cdot N \quad (3.6)$$

wobei N die Normalkraft, die immer senkrecht zur Fläche steht und μ die Reibzahl für die Haftreibung ist. Aus [Fis11, S.220] kann bei einem Anziehen von Hand angenommen werden, dass $N = F_v$ ist. Nach [WMR15] wird angenommen, dass $\mu \approx 0,19$. Es wird hier davon ausgegangen, dass μ an keiner anderen Schraubverbindung geringer ist. Somit ergibt sich $F_{reib} = 570$ N was ≈ 57 kg entspricht. Mit anderen Worten hält jede Schraubverbindung einer seitlichen Belastung von 57 kg stand.

Profilnut

Laut Datenblatt [ite17, S.2] darf eine maximale Zugkraft von $F_{zul} = 500$ N an den Nutflanken angreifen. Die Nennlast beinhaltet bereits einen Sicherheitsfaktor von $S > 2$ gegen plastische Verformung. Der Anbau von Bauteilen an der Nut ist ausschließlich für die standardisierten Nutzensteine vorgesehen. Abbildung 3.16 zeigt den Bereich des MAN, an dem die größte Belastung für die Nut vermutet wird. Der Abstand vom Drehpunkt und Nutzenstein ist hier sehr klein, was einen kleinen Hebelarm L_H zur Folge hat, wobei L_V verhältnismäßig groß ist. Ganz allgemein lässt sich das Hebelgesetz mit Gleichung 3.7 beschreiben.

$$M = F_1 \cdot l_1 = F_2 \cdot l_2 \quad (3.7)$$

Stellt man die Gleichung 3.7 um, kann man die Gleichung 3.8 für die Zugkraft an den Nutflanken hieraus ableiten.

$$F_{zug} = \frac{F_{brems} \cdot L_V}{L_H} \quad (3.8)$$

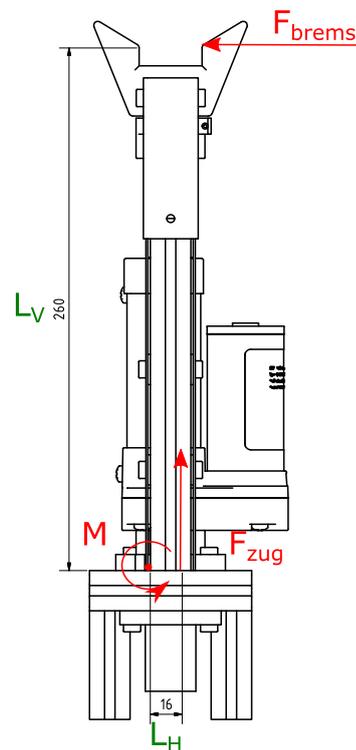


Abbildung 3.16.: Nutbelastung

Setzt man die Werte aus Abbildung 3.16 mit $L_V = 260 \text{ mm}$, $L_H = 16 \text{ mm}$ und $F_{brems} = 21 \text{ N}$ in die Gleichung 3.8 ein, so erhält man $F_{zug} = 341,25 \text{ N}$. Da $F_{zul} > F_{zug}$ und die zweite Strebe des Hubarmes, welche in Wirklichkeit das Moment teilen würde, außer Acht gelassen wurde, wird die Nut nicht aufbördeln bzw. den aufkommenden Belastungen standhalten.

Lagerberechnung

Für die Ermittlung der maximalen Traglasten wird nur eines der beiden Flanschlager beachtet und angenommen, dass dieses fest eingespannt ist. Um Montagefehler auszugleichen, können die Lager in ihrem Gehäuse geschwenkt werden, also kein Moment aufnehmen. Um zu gewährleisten, dass die Drehachse im Betrieb vertikal ausgerichtet bleibt, wurden zwei Flanschlager symmetrisch gegenüberliegend (siehe Abbildung 3.9) montiert. In Wirklichkeit teilen sich die Lager also die Belastung lediglich als eine axiale und radiale Last, nehmen jedoch kein Drehmoment auf. Die weiteren Berechnungen folgen demnach wieder einer konservativen Betrachtung.

Laut Datenblatt [Mäd17] beträgt die radiale statische Tragzahl $C_{0radial} = 6,2 \text{ kN}$. Weiterhin ist dort vermerkt, dass die axiale Tragzahl 20% der radialen Tragzahl, also $C_{0axial} = 6,2 \text{ kN} \cdot 0,2 = 1,24 \text{ kN}$, ausmacht. Man kann sagen, dass das Lager radial mit 620 kg und axial mit 124 kg belastet werden kann.

Das maximal zumutbare Drehmoment wird mit der Gleichung 3.9 des Lagerherstellers SKF [SKF17] durchgeführt

$$M_{zul} = 0,23 d_1 \left(\frac{C_{0axial}}{S_0} - F_{axial} \right) \quad (3.9)$$

wobei der Sicherheitsfaktor $S_0 = 3$, der Außendurchmesser des Innenringes $d_1 = 17 \text{ mm}$ und die zentrisch wirkende Axialbelastung $F_{axial} = 260 \text{ N}$ beträgt. Hierbei wird angenommen, dass das Gesamtgewicht der RT aufliegt. Dies ist jedoch ein eher unwahrscheinlicher Betriebsfall, da der Endeffektor genau an dem Schwerpunkt der RT angreifen müsste. Im Normalfall wird ein Rollenpaar der RT immer den Boden berühren. Für das maximal zumutbare Drehmoment am Lager ergibt sich nach der konservativen Betrachtung $M_{zul} = 7,064 \text{ N}$.

Betrachtet man nun Abbildung 3.16, kann man auf das wirkende Moment am Lager mit $M = F_{brems} \cdot L_v = 21 \text{ N} \cdot 260 \text{ mm} = 5,46 \text{ N m}$ schließen. Da $M_{zul} > M$, würde selbst ein Lager allein den aufkommenden Belastungen standhalten.

Durchstanzen

Um zu verhindern, dass die DIN 7984 M5 x 8 Schrauben aus Abbildung 3.8 durch die Dachplatte des FZG durchreißen bzw. durchstanzen, wurden zusätzliche DIN 125-1 A A 5,3 Unterlegscheiben verbaut. Trotzdem soll hier eine rechnerische Überprüfung des Durchstanzens nach DIN EN 1993 (EC 3) [Feh93, S.70] ohne Unterlegscheiben durchgeführt werden. Die Norm besagt, dass der Durchstanznachweis bei Stahlkonstruktionen nicht maßgebend ist, wenn

$$d \approx \sqrt{50 \cdot t_{min} - 2} \quad (3.10)$$

erfüllt ist, wobei $t_{min} = 1 \text{ mm}$ die minimale Blechdicke und $d = 5 \text{ mm}$ der Gewindedurchmesser der Schraube ist. Berechnet man die Wurzel aus Gleichung 3.10, so ergibt dies 6,93 mm. Dieses Ergebnis ist zwar nicht weit von den erforderlichen 5 mm entfernt, jedoch besteht die Dachplatte des FZG nicht aus Stahl, sondern aus einer Aluminiumlegierung. Aufgrund dessen wird die Durchstanztragfähigkeit mit der Gleichung 3.11 durchgeführt.

$$B_{p,Rd} = 0,6 \cdot \pi \cdot d_m \cdot t \cdot \frac{f_u}{\gamma_{m2}} \quad (3.11)$$

Der mittlere Durchmesser ist der Mittelwert aus Eckmaß und Schlüsselweite des Schraubenkopfes oder der Schraubenmutter (kleinerer Wert maßgebend). Da es sich hier um eine

Rundkopfschraube handelt ist der Durchmesser immer gleich mit $d_m = 8,5$ mm. Die Zugfestigkeit des Blechmaterial muss geschätzt werden, da der Hersteller des FZG nur angibt, dass es sich um eine Aluminiumlegierung handelt. Bei einem Vergleich mit gängigen Legierungen, wie Dural AlCu4Mg1, wird $f_u = 450$ N/mm² angenommen. Der Teilsicherheitsbeiwert ist $\gamma_{m2} = 1,25$. Daraus ergibt sich $B_{p,Rd} = 5,7$ kN. Das bedeutet, dass an nur einer Schraube ohne Unterlegscheibe 570 kg ziehen müssten, damit diese durch das Blech reißt. Die Betrachtung ist nun ähnlich wie in Abschnitt 3.2.3. Es wird angenommen, dass der Drehpunkt bei einem benachbarten Schraubenpaar liegt (vgl. Abbildung 3.17). Bei dieser Ausrichtung des FZG würde F_{brems} eigentlich um 90° verdreht in die Blattebene gehen, sodass das um 90° gedrehte Schraubenpaar als Drehpunkt dienen würde. Da aber auch hier wieder ein konservativer Ansatz gelten soll, wird L_H mit dem kleineren Abstand gewählt. Mit den neuen

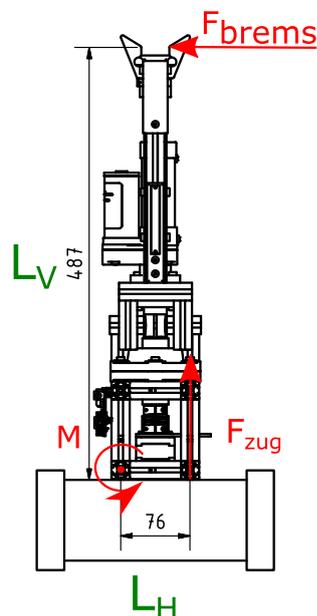


Abbildung 3.17.: Zug an Schraube im Dachblech

Abständen $L_V = 487$ mm und $L_H = 76$ mm ergibt sich nach der bekannten Gleichung L3.8 $F_{zug} = 134,57$ N. Da $F_{zug} \ll B_{p,Rd}$ und auch noch zusätzliche Unterlegscheiben verbaut sind, wird keine Schraube durch das Blech reißen.

Führungsschale FEM

Vorweg ist gleich klar zu stellen, dass es sich hier nicht um eine ausführliche und präzise Finite Elemente Methode (FEM) Analyse handelt. Grundsätzlich könnten auch andere Annahmen für eine überschlägige Berechnung der Festigkeit für die Führungsschale getroffen

werden. Da jedoch die Geometrie etwas komplexer ist und die CAD Daten bereits in INVENTOR vorhanden sind, ist es tatsächlich einfacher ein grobes Ergebnis über das interne FEM Tool von INVENTOR zu erhalten. Abbildung 3.18 zeigt das Ergebnis dieser Simulation. Es sollen hier nur einige wenige, allgemeine Anmerkungen zur FEM Analyse folgen.

Mit dem Mashing (oder auch Vernetzung) steht und fällt die Rechnung. Je feiner das Mashing ist, desto genauer sind die Ergebnisse. Auch die Anordnung der Knotenpunkte spielt eine Rolle. INVENTOR hat hier automatisch eine Triavernetzung gewählt, welche durch ihre spitz zulaufenden Knotenpunkte jedoch nicht die Beste ist. Ein guter FEM-Analytiker passt die Form der Vernetzung den Gegebenheiten selbst an. Das Mashing ist mit INVENTOR nur schwer steuerbar. Die im Hintergrund ablaufenden Rechenvorgänge sind zudem unklar.

Für eine überschlägige Berechnung ist dieses Tool jedoch eine einfache und gute Alternative, sofern die Geometrie bereits in INVENTOR vorhanden ist. Die Führungsschale ist ein

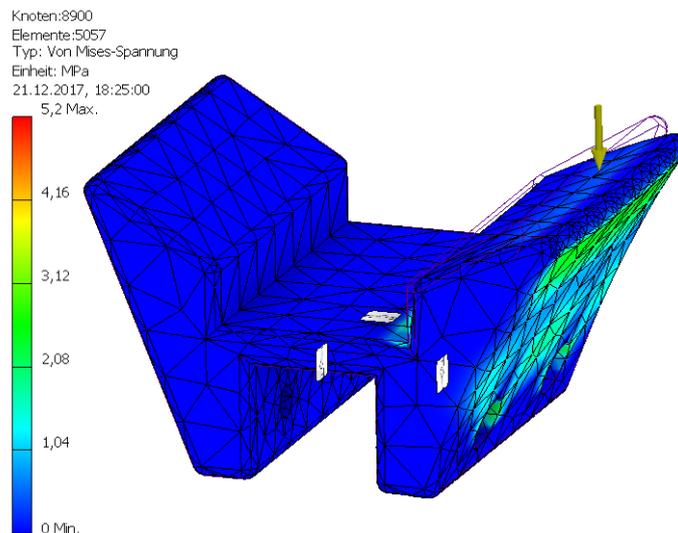


Abbildung 3.18.: FEM Analyse der Führungsschale

3D Druckteil, welches mit einer 1 mm Wandung hohl gedruckt und anschließend mit einem Acrylharz ausgegossen wurde. Für die Simulation wurde angenommen, dass es sich um einen Vollkörper aus ABS bzw. Acrylnitril-Butadien-Styrol mit einer Streckspannung 20 MPa und Zugfestigkeit 29 MPa handelt. Die Streckgrenze von Acrylgießharzen ist in der Regel doppelt so groß. Es wird simuliert, dass die Kraft an nur einer Seite von oben mit dem ganzen Gewicht der Tafel von 26 kg eingeleitet wird. Die Durchbiegung kann wahlweise übertrieben dargestellt werden, was nicht der Realität entspricht. Dadurch kann jedoch verifiziert werden, ob die Lastannahmen plausibel sind. Die nun maximal auftretende Spannung im Bauteil beträgt 5,2 MPa oder auch $5,2 \text{ N/mm}^2$. Das Bauteil wird somit den Belastungen stand halten.

Maximale Beschleunigung

Beim Anfahren ist es möglich, dass das FZG mit dem angeschraubten MAN kippen könnte. Um dies zu verhindern darf eine bestimmte Beschleunigung nicht überschritten werden. Eine leichtes Nicken, dass nicht zum völligen Umsturz führt, soll hier toleriert sein.

Hierzu müssen die Kräfte ermittelt werden, die im Zusammenhang mit der Nickbewegung stehen. Dies ist zum einen die Losbrechkraft F_{los} und zum anderen die d'Alembertsche Trägheitskraft $F_{d'Alembert}$. Die Losbrechkraft muss aufgebracht werden, um die RT in Bewegung zu versetzen. Die Losbrechkraft wurde in Abschnitt 2.2.1 experimentell ermittelt. Die Bremskraft F_{brems} ist vom Betrag her identisch mit F_{los} wirkt aber in entgegengesetzter Richtung. Um genau zu sein, muss F_{los} etwas größer als F_{brems} sein, da sonst ein Kräftegleichgewicht herrschen würde. Die RT würde sich dann nicht bewegen. Ist die Haftreibung erst einmal überwunden, rollt die RT. Nun ist nur noch Gleitreibung vorhanden, welche stets kleiner, als die Haftreibung ist.

Jeder Körper möchte seinen momentanen Beschleunigungszustand beibehalten. Wird der Körper z. B. mit a beschleunigt, so wirkt die d'Alembertsche Trägheitskraft entgegen der Beschleunigungsrichtung [Gro+15, S.202]. Vom Betrag her ist $F_{d'Alembert}$ genauso groß, wie die beschleunigende Kraft. All diese Kräfte sind in Abbildung 3.19 illustriert.

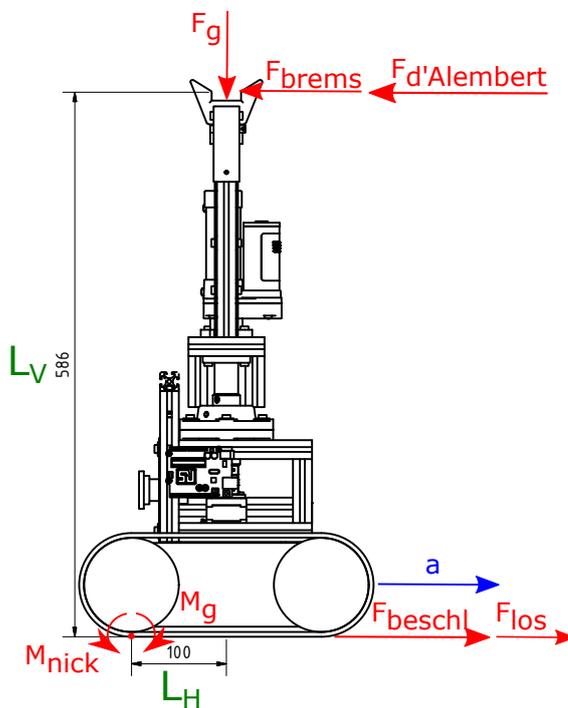


Abbildung 3.19.: Nickmoment

Die Nickbewegung wird also durch F_{brems} und $F_{d'Alembert}$ verursacht. F_{los} und die Kraft zur Beschleunigung des Gesamtsystem (GS) F_{beschl} werden vom FZG aufgebracht. Diese haben aber, in Bezug auf den Drehpunkt, um den das GS kippen würde, keinen Hebelarm und können der Nickbewegung somit nicht entgegenwirken. Man kann nicht sagen, dass F_{beschl} keine Rolle spielt, denn je größer F_{beschl} ist, desto größer ist auch $F_{d'Alembert}$. Die einzige Kraft, die einem Nicken entgegen wirkt, ist die Gewichtskraft der aufliegenden RT und dem Eigengewicht des GS. Zusammengesetzt ergeben diese Kräfte F_g . Das Moment, welches eine Nickbewegung forciert berechnet sich nach Gleichung 3.12 und das Gegenmoment nach Gleichung 3.13

$$M_{nick} = (F_{brems} + F_{d'Alembert}) \cdot L_V \quad (3.12)$$

$$M_g = F_g \cdot L_H \quad (3.13)$$

Da sich das FZG bewegen soll, muss es etwas mehr als F_{los} in Bewegungsrichtung aufbringen, um F_{brems} zu überwinden. Die zusätzliche Kraft F_{beschl} , die das FZG aufwenden muss, um das GS zu beschleunigen, darf jedoch nur so groß werden, dass $F_{d'Alembert}$ nicht die Überhand gewinnt und das FZG zum Kippen bringt. Der kritische Punkt liegt bei $M_g = M_{nick}$. Verbindet man Gleichung 3.13 und Gleichung 3.12 miteinander und stellt diese um, so erhält man eine neue Gleichung 3.14 für den maximalen Wert von $F_{d'Alembert}$.

$$F_{d'Alembert} \leq \frac{F_g L_H}{L_V} - F_{brems} \quad (3.14)$$

Über das allgemein bekannte zweite Newtonsche Gesetz $F = m \cdot a$ lässt sich $F_{d'Alembert}$ in Bezug zur Masse des GS setzen und erhält so die maximal erlaubte Beschleunigung a_{max} für das GS in Gleichung 3.15. $m_{GS} = m_{RT} + m_{MAN} + m_{FZG}$, wobei $m_{RT} = \frac{26 \text{ kg}}{2}$. Dies ist das minimale Gewicht mit dem die RT, je nach Aufnahmepunkt, auf den Endeffektor drückt.

$$a_{max} \leq \frac{F_{d'Alembert}}{m_{GS}} \quad (3.15)$$

Setzt man Gleichung 3.14 in Gleichung 3.15 mit $F_g = m_{GS} \cdot g$ ein so erhält man Gleichung 3.16

$$a_{max} \leq \frac{g \cdot L_H}{L_V} - \frac{F_{brems}}{m_{GS}} \quad (3.16)$$

Mit den konkreten Werten $L_H = 100 \text{ mm}$, $L_V = 586 \text{ mm}$, $m_{GS} = 23 \text{ kg}$, $F_{brems} = 21 \text{ N}$ ergibt sich $a_{max} = 0,761 \text{ m/sec}^2$.

Wird das GS mit nicht mehr als $0,761 \text{ m/sec}^2$ beschleunigt, so ist nicht zu erwarten, dass dieses nickt bzw. kippt.

3.3. Elektronik

Die Bestandteile der Elektronik und deren Zusammenhänge sind in Abbildung 3.20 illustriert. Die Elektronik hat die Aufgabe den Drehwinkel des Hubarmes und den Hub des Endeffektors mit Hilfe von Sensoren zu digitalisieren und so für die Software bereitzustellen. Weiterhin soll die Elektronik die Steuerkommandos der Software in mechanische Bewegungen des Linear- und Schrittmotors über eine Leistungselektronik umwandeln

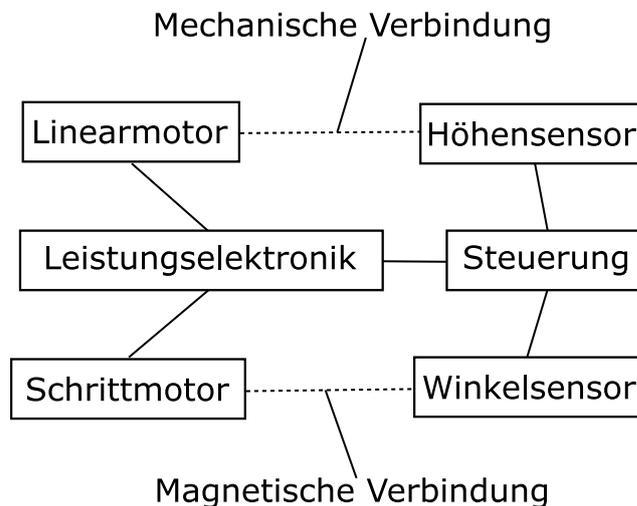


Abbildung 3.20.: Elektronik Blockschaltbild

Die Steuerung empfängt Steuerkommandos und sendet Statusinformationen über eine serielle Schnittstelle. Je nach Kommando wird ein Linear- oder Schrittmotor über eine Leistungselektronik angesteuert. Die Steuerung liest dabei jeweils die Daten eines Höhen- und Winkelsensors aus. Der Linearmotor ist mechanisch mit einem Schiebepotentiometer, welcher als Höhengensensor fungiert, verbunden. An dem Schrittmotor befindet ein Magnet, sowie ein Magnetsensor. Beide zusammen bildet einen Winkelsensor.

Für die Aktoren und vor Sensoren ist es wichtig, dass diese korrekt positioniert werden. Andernfalls leiten die Aktuatoren Kräfte in falsche Richtungen oder arbeiten durch unnötige Reibung ineffizient. Weiterhin würden die Sensoren nicht das messen, was gemessen werden soll, oder sie werden unter Umständen unerlaubt mechanisch belastet. Für die Fertigung

von Adaptern zur Montage der Sensorik ist das 3D Druck Sterolithografieverfahren sehr gut geeignet. Die Druckteile können zwar keine großen mechanischen Belastungen aushalten, sind aber äußerst maßhaltig.

In den kommenden Abschnitten werden diese Komponenten genauer erläutert.

3.3.1. Aktoren

Es kommen zwei Aktoren zum Einsatz. Zum einen ein Linearmotor und zu anderen ein Schrittmotor. Hier werden die Kenndaten, Schnittstellen und mögliche Kräfte bzw. Momente welche die Aktuatoren aufbringen können vorgestellt.

Linearmotor

Es wird Linearmotor, ähnlich wie in Abbildung 3.21, verwendet. Das entscheidende Auswahlkriterium ist die Hubkraft des Kolbens, welche >260 N sein muss, um die RT anzuheben. Der Hub muss ≥ 30 mm sein, damit die Führungsschale unter die Längsstrebe der RT positioniert werden kann und danach bis an die Anpressfläche und darüber hinaus angehoben wird. Durch den dritten Auswahlfilter Kosten wurde die Auswahl drastisch reduziert, da alle anderen Linearactuatoren um den Faktor ≥ 6 teurer wären.

Das Typenschild des Linearmotors gibt folgende Kenndaten an:

- Model: JS-TGZ-U2
- Trip: 50 mm
- Voltage: 12 V DC
- Speed: 6 mm/s
- Maximum thrust: 1500 N

Es fehlt die Angabe der Leistung bzw. des Stromes. Messungen ohne Last haben ergeben, dass der Motor bei 12 V DC Versorgungsspannung ca. 1 A Strom verbraucht.

Der Kolben wird über einen Spindelantrieb nach außen bzw. innen bewegt. Der Elektromotor des Antriebes ist in dem seitlichen Zylinder untergebracht und treibt die Spindel über Zahnräder und einen Riemen an. Intern verfügt der Motor über Endlagenschalter, welche den Antrieb abschalten, sobald der Kolben entweder ganz aus- oder eingefahren ist. Die 12 V DC Stromversorgung erfolgt über zwei Litzen mit positiver und negativer Polung. Wird eine ausreichende Spannung an diesen Litzen angelegt, fängt der Motor an sich zu bewegen. Sobald eine der Endlagen erreicht ist, schaltet der Motor ab. Durch eine Umpolung der



Abbildung 3.21.: Linearmotor [Ama17]

Anschlüsse, kann der Kolben sich in der entgegengesetzten Richtung wieder bewegen. Der Kolben hat $\varnothing 20$ mm, ebenso der Montagestumpf an der Unterseite (vgl. Abbildung 3.21). Mit einem Wellenhalter SHF 20 mm, wie in Abbildung 3.12, wird der Linearmotor befestigt.

Schrittmotor

Zum Einsatz kommt der Schrittmotor aus Abbildung 3.22. Das Auswahlkriterium des Schrittmotors basiert auf der Auswahl der Leistungselektronik. Diese kann pro Phase maximal 1,2 A liefern. Der benötigte Strom pro Phase sollte also ≤ 1 A sein.



Abbildung 3.22.: Schrittmotor [Pol17a]

Aus dem Datenblatt werden die wichtigsten Kenndaten entnommen [Pol17b]:

- Phase: 2 Phase
- Step Angle: $1,8^\circ/\text{STEP} \pm 5\%$
- Voltage: 12 V
- Current: 0,4 A/PHASE
- Resistance: $30 \Omega/\text{PHASE} \pm 10\%$
- Holding torque: 38 N cm

Das Flanschmaß des Motors entspricht der NEMA17 Norm. Mit einem Montageblech, wie aus Abbildung 3.7, kann der Motor befestigt werden. Über zueinander versetzte Spulen können präzise Positionen angefahren werden. Wird eine oder mehrere Spulen, die sich im Rotor befinden, bestromt, wird ein Magnetfeld aufgebaut. Im Stator befinden sich Permanentmagnete. Der Rotor fängt allerdings nicht an sich ständig im Kreis zu drehen, sondern rotiert nur um ein kleines Stück. Für eine inkrementelle, zyklische Rotation müssen die Spulen mit einem Motortreiber bzw. einer Leistungselektronik betrieben werden.

3.3.2. Sensoren

Höhensensor

Um zu messen, wie weit der Endeffektor ausgefahren ist, wird als Höhensensor ein Schiebepotentiometer (siehe Abbildung 3.23) verwendet. Das Funktionsprinzip ist, wie bei einem Drehpotentiometer der Abgriff eines variablen Spannungsteilers. Das Poti besitzt drei Anschlüsse: GND , V_{CC} und V_{mess} , welche über einen Schleifkontakt abgegriffen wird. Der gesamte Spannungsteiler wird über GND und V_{CC} an 5 V angeschlossen. Die abgegriffene Spannung V_{mess} kann also Werte zwischen 0 V und 5 V annehmen. 0 V liegen an, wenn der Schieber an seinem unteren und 5 V am oberen Anschlag ist.

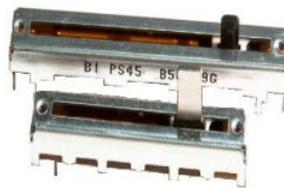


Abbildung 3.23.: Linearmotor [Ele17]

Hier die wichtigsten Kenndaten aus dem Datenblatt [Ele17]:

- Typ: PS6010MA1B
- Widerstand: 50 k Ω linear
- Schiebewege: 60 mm
- Belastbarkeit: 0,2 W
- Toleranz: $\pm 20\%$

Die Toleranz darf hier nicht falsch verstanden werden. Sie bezieht sich auf den Widerstandswert des ganzen Spannungsteilers. Ob der Spannungsteiler nun einen Widerstand von $50\text{ k}\Omega$ oder $40\text{ k}\Omega$ hat ist für die Messung unerheblich. Grundsätzlich sollte der Spannungsteiler hochohmig sein, damit der dauerhaft fließende Strom zwischen GND und V_{CC} möglichst klein bleibt. Von Seiten der Elektronikhersteller existiert daher kein Angebot an niederohmigen Potentiometern. Da der maximale Hub des Linearmotors 50 mm beträgt, hat das Poti sogar noch 10 mm Reservehub zur Verfügung. Dies ist auch zwingend notwendig, da immer Maßtoleranzen vorhanden sind und somit ein Poti mit lediglich 50 mm Verfahrweg zu stramm kalkuliert wäre. Eine Kalibrierung erfolgt über die Software.

Das Potigehäuse ist mit zwei M2 Gewindebohrungen für die Montage flankiert. Die Zubehörerauswahl der Strangpressprofile bietet für solch kleine Gewinde jedoch keine Auswahl an. Daher wird ein Adapter (siehe Abbildung 3.24) im 3D Druck Sterolithografieverfahren hergestellt.

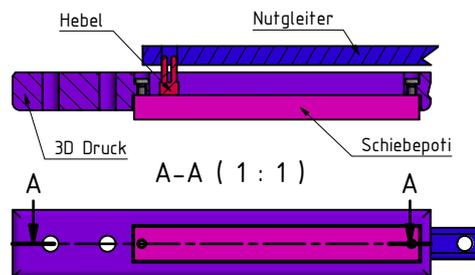


Abbildung 3.24.: Adapter Schiebepoti Schnittansicht

Der Adapter wird durch seine zwei $5,3\text{ mm}$ Bohrungen an dem Hubarm (vgl. Abbildung 3.10) festgeschraubt. Das Poti selbst wird über seine M2 Gewindebohrungen an den Adapter geschraubt. Der Hebel des Schiebepotis kann sich in einem Schlitz des Adapters frei bewegen und taucht auf der gegenüberliegenden Seite in einen Nutgleiter ein. Dieser läuft beweglich in der Nut des Hubarmes. Im Schulterblech des Endeffektors ist eine $M4 \times 6$ Schraube (vgl. Abbildung 3.15). Fährt der Endeffektor hoch, nimmt die Schraube den Nutgleiter mit.

Winkelsensor

Als Winkelsensor war ursprünglich ein Drehpotentiometer angedacht, da diese sehr günstig sind und eine Drehung über 200° durch die Rotationsanschlänge ohnehin verhindert wird. Die Welle des Schrittmotors ist von zwei Seiten her zugänglich, wobei diese nur auf einer Seite aus dem Motorgehäuse ragt. Erste Prototypen mit 3D Druckteilen haben entweder durch mechanische Belastung nach einiger Zeit versagt oder haben während des Betriebes axiale und radiale Kräfte auf das Drehpoti ausgeübt, welches sich dafür auf Dauer nicht eignet.

Das grundsätzliche Problem besteht darin die Bauteile exakt konzentrisch zueinander zu befestigen. Es soll nicht ausgeschlossen werden, dass dies möglich ist, aber es ist auf jeden Fall mit Aufwand verbunden. Eine sehr gute Lösung zur Erfassung des Drehwinkels bietet eine kontaktlose Messung mit einem Magnetsensor. Auf die Stirnseite der Motorwelle, die im Motorgehäuse versenkt ist, wird ein Diametralmagnet, wie in Abbildung 3.25, geklebt.

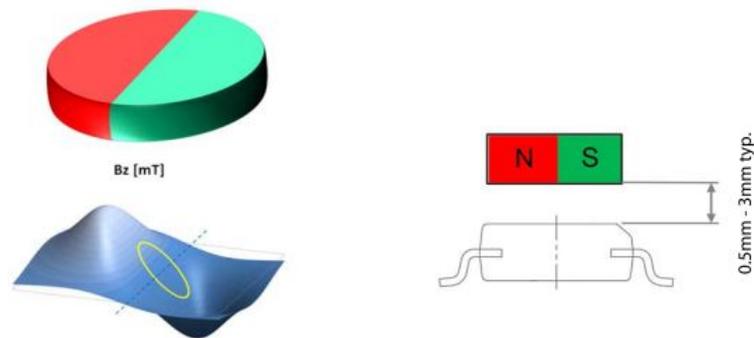


Abbildung 3.25.: Diametralmagnet über Magnetsensor [AMS17]

Dicht über dem Magnet verlaufen die Magnetfeldlinien von Nord nach Süd parallel. Die Drehung dieses homogenen Magnetfeldes können spezielle Magnetfeldsensoren, wie der AS5601 aus Abbildung 3.26, detektieren.

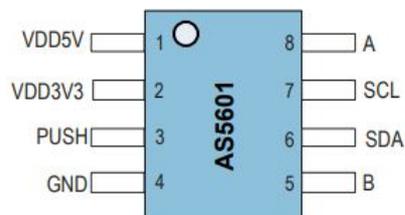


Abbildung 3.26.: AS5601 Magnetfeldsensor [AMS17]

Hier die wichtigsten Kenndaten [AMS17]:

- Typ: AS5601 12-bit Programmable Contactless Encoder
- Auflösung: Ausgabe des Winkels als digitaler Wert mit $2^{11} = 2048 \frac{STEPS}{180^\circ}$
- Stromversorgung: 5 V oder 3,3 V
- Package: SOIC-8
- Schnittstellen: I2C und PWM

Der AS5601 ist in einem SOIC-8 Package untergebracht, welches nicht auf dem Schrittmotor angebracht werden kann. Der Sensor muss auf einem Breakoutboard verlötet sein, um ihn über dem Diametralmagneten an dem Schrittmotor positionieren zu können. Abbildung 3.27 zeigt die Halterung für das Breakoutboard, welche im 3D Druck hergestellt wurde, für dieses Breakoutboard. Dort sind Einlassungen für M3 Muttern und Bohrungen vorhanden, damit die Schrauben des Schrittmotors zugänglich bleiben. Die in der Abbildung 3.27 sichtbare Seite wird bündig auf den Flansch des Schrittmotors geklebt. Die eingesetzten M3 Muttern sind durch die Halterung gekontert und dienen als Befestigungsgewinde für das Breakoutboard.

Der entscheidende Vorteil des AS5601 ist, dass Fehlstellungen des Magneten zum Sensor keine unerwünschte mechanische Belastung erzeugt. Laut Datenblatt darf der axiale Versatz des Magneten zum Sensor nur 0,25 mm betragen. Die gilt jedoch für eine Winkelauflösung von $> 0,1^\circ$. Ein Auflösung von $\pm 2^\circ$ ist völlig ausreichend. Leichte Schrägstellungen des Magneten (vgl. Abbildung 3.25) kann der Sensor durch eine differentielle interne Messung kompensieren.

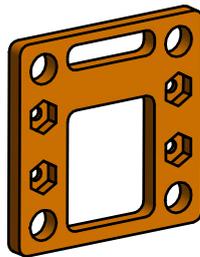


Abbildung 3.27.: PCB Halterung

Breakoutboard kaufen oder selbst herstellen?

Grundsätzlich gibt es den AS5601 auf einem Breakoutboard zu kaufen auf das PCB Halterung hätte problemlos angepasst werden können. Während der Testphase schien die Lösung mit einem Drehpoti befriedigend zu sein. Die Entscheidung einen Magnetfeldsensor zu verwenden fiel erst relativ spät. In Anbetracht der noch zur Verfügung stehenden Zeit war es nicht mehr möglich bzw. sinnvoll das Breakoutboard zu bestellen, da dieses mehrere Wochen Lieferzeit gehabt hätte. Da ein AS5601 sofort zur Verfügung stand, wurde das Breakoutboard selbst designed.

Mit der Freeware DESIGNSPARK PCB wurde das Platinenlayout erstellt. Abbildung 3.28 zeigt die 3D Darstellung der Platine. Dem Datenblatt wurde entnommen, dass zwei Keramik Kondensatoren jeweils mit $1\ \mu\text{F}$ zwischen *GND* und *VDD3V3* und mit $100\ \text{nF}$ zwischen *GND* und *VDD5V* gesetzt werden müssen[AMS17]. Zur Verbindung mit der Steuereinheit ist eine abgewinkelte, 4-polige Stiftleiste mit einer entsprechenden Buchse als Gegenstück vorgesehen.

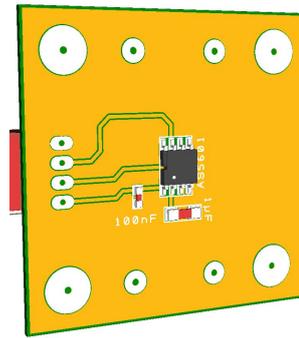


Abbildung 3.28.: PCB Design

Welche Richtlinien bei dem Platinenlayout einzuhalten sind, soll hier nicht weiter erörtert werden.

3.3.3. Steuergeräte

Die Steuerung hat die Aufgabe das Zusammenspiel der Aktoren und Sensoren zu koordinieren, sowie Statusinformationen nach Außen zu senden und Steuerkommandos entgegenzunehmen. Als Steuerung wird ein Arduino Uno Clone verwendet. Die Uno Plattform ist sehr bekannt und in Verbindung mit den Arduino Bibliotheken leicht zu verwenden. Zudem gibt es viele Hersteller, die einen modularen Aufbau durch passende Shields unterstützen.

Hier die wichtigsten Kenndaten [[Cen17](#)]:

- Typ: Duino Open UNO
- Spannungsversorgung: min. 6 V bis max. 20 V
- Level der Ausgänge: 5 V oder 3,3 V wählbar
- Schnittstellen: ICSP, SPI, I2C, Seriell Tx/RX (USB)
- Prozessor: Atmega 328P
- 6 Eingänge für 12 bit ADC Wandlung

Über den USB Anschluss kann das Board mit der Arduino Integrated Development Environment (IDE) programmiert werden. Zudem kann auch der serielle Monitor der IDE zu Evaluationszwecken verwendet werden. Hierüber können Statusinformation abgerufen und Kommandos an den MAN gesendet.

Als Leistungselektronik wird für das Unoboard ein passendes Motorshield von ADAFRUIT wie in [Abbildung 3.29](#) verwendet.

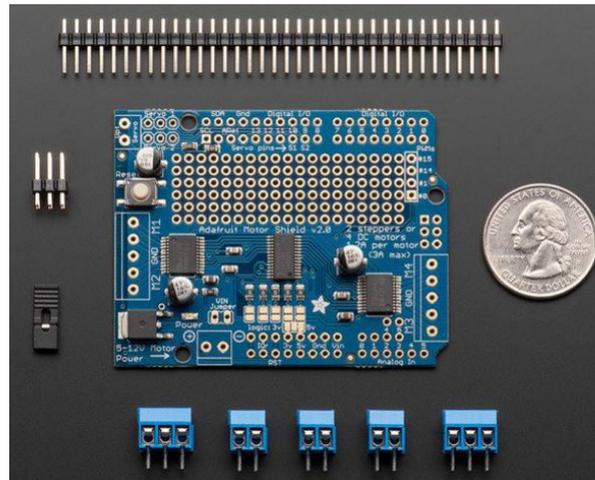


Abbildung 3.29.: Motorshield [Ada17]

Bei solchen Shields ist es üblich, dass die Stiftheisten selbst angelötet werden müssen. Über einen 2-poligen Schraubklemmblock wird das Shield mit 12 V versorgt. Wird der mitgelieferte Jumper gesetzt, brückt dieser die Spannungsversorgung zum Unoboard. Das Shield ist in der Lage Servo, DC- und Schrittmotoren zu steuern. Dieses Shield hat den Vorteil, dass es einen gemischten Betrieb der Motorarten bedienen kann. Linear- und Schrittmotoren können also mit diesem Shield gesteuert werden.

Hier die wichtigsten Kenndaten [Ada17]:

- Typ: Adafruit Motor Shield V2 for Arduino
- Spannungsversorgung: min. 5 V bis max. 12 V
- bis zu zwei 5 V Servos
- bis zu vier bidirektionale DC Motoren
- bis zu zwei Schrittmotoren
- bis zu 1,2 A pro Phase, kurzzeitig (20 ms) auch bis zu 3 A
- TB6612 chipset mit thermischer Überlastabschaltung

3.3.4. Safety

Der Abschnitt 3.3.4 ist nicht als klassische Safetyanalyse zu verstehen, sondern behandelt Gefahrenquellen, die während den ersten Funktionstests aufgefallen sind und die entsprechenden Schutzmaßnahmen.

Das mechanische Gefahrenpotential wurde bereits in Abschnitt 3.2.2 beschrieben. Als zusätzlich Sicherheitsmaßnahme wird ein Notausschalter für die Elektronik, wie in Abbildung 3.30, verwendet.

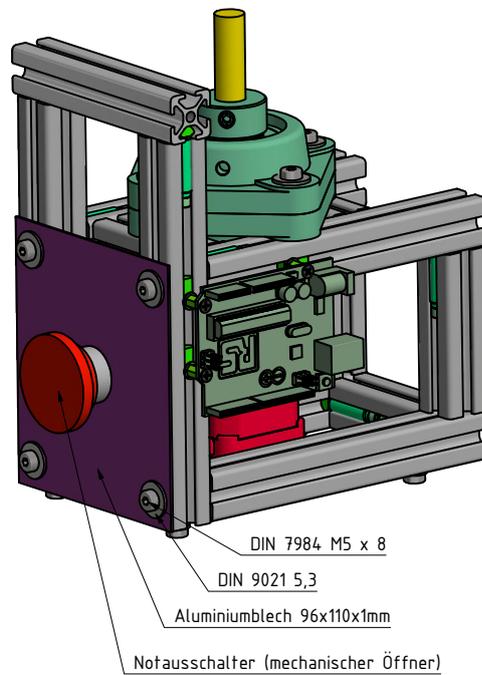


Abbildung 3.30.: Notausschalter

Der Notausschalter ist über ein Aluminiumblech an dem Untergestell befestigt. Wird der Notaus eingedrückt, so trennt ein mechanischer Schalter die Plusleitung der Stromversorgung bzw. des Akkus. Das bedeutet auch, dass nicht nur die Stromversorgung des MAN, sondern auch die des FZG unterbrochen ist. Wird der Notaus im Uhrzeigersinn gedreht, schließt der mechanische Kontakt und die Stromversorgung ist wieder hergestellt.

3.4. Software

Die Umsetzung des gewünschten Programmverhaltens wird mit einer State Machine (STM) aus der SysML modelliert. Das Kontextobjekt ist die Manipulatorklasse. Eine Manipulator STM beschreibt das Verhalten dieser Klasse (vgl. [Wei14, S.269]).

3.4.1. Klassen

Das Klassendiagramm aus Abbildung 3.31 stellt die Klassen nicht vollständig dar, sondern die Attribute und Methoden, die verwendet werden.

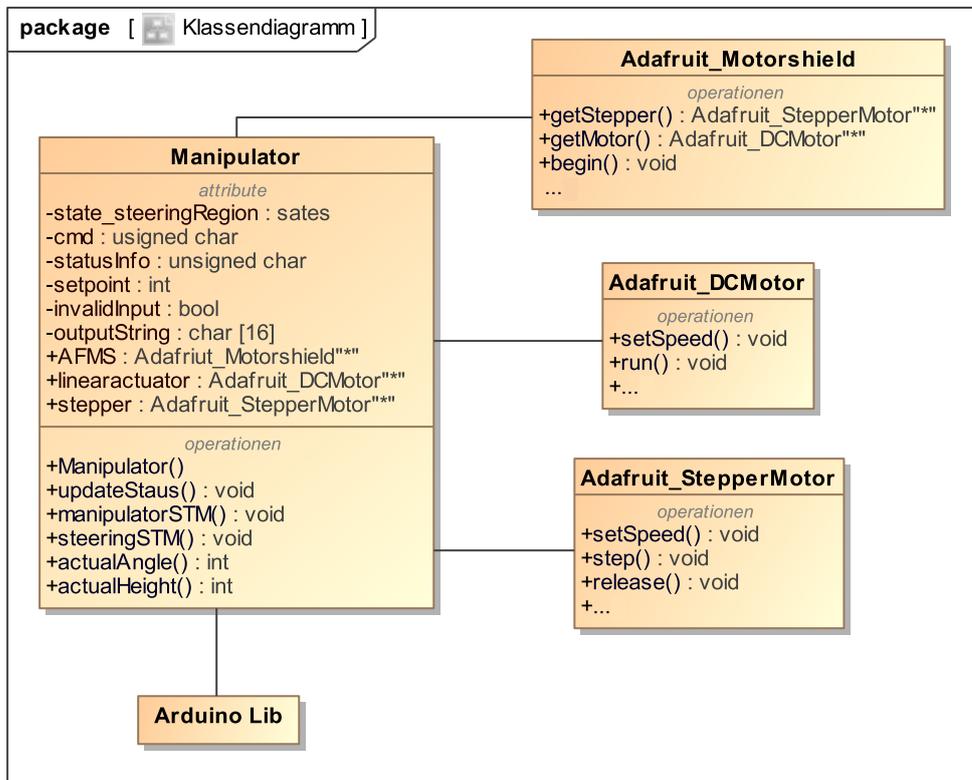


Abbildung 3.31.: Klassendiagramm

Die Manipulatorklasse verfügt über eine Reihe von Attributen und Methoden. Die Methoden `getMotor()` und `getStepper()` der Klasse `Adafruit_Motorshield` liefern als Rückgabewert die Adressen von vordefinierten Objekten. Diese Adressen werden den Attributen `AFMS`, `linearactuator` und `stepper` der Manipulatorklasse zugewiesen. Die Manipulatorklasse erhält so Zugriff auf die vordefinierten Objekte und kann damit die an das Shield angeschlossenen Aktuatoren steuern.

Die Methode `begin()` der Klasse `Adafruit_Motorshield` initialisiert das Shield, unter anderem die I2C Verbindung. Mit Objekten der Klasse `Adafruit_Motor` können Gleichstrommotoren vor und zurück mit variabler Geschwindigkeit gedreht werden oder stromlos bleiben. Mit Objekten der Klasse `Adafruit_StepperMotor` kann man Schrittmotoren in verschiedenen Modi mit unterschiedlichen Drehsinn, Schrittweite und Geschwindigkeit betreiben. Sehr wichtig ist auch die Methode `release()`, welche das Haltemoment des Schrittmotors abschaltet und ihn somit frei drehbar macht.

Alle Objekte bzw. Objektpointer werden über den Konstruktor der Manipulatorklasse erzeugt. Da das Programm in einer Endlosschleife läuft, ist kein Destruktor der Objekte erforderlich.

Die Methode `updateStatus()` sorgt dafür, dass die Statusinformationen ständig über die serielle Schnittstelle versendet werden und überprüft, ob neue Kommandos im seriellen Eingangspuffer liegen.

Die Methode `manipulatorSTM()` setzt die STM in Gang. Die Methode `steeringSTM()` ist eine untergeordnete STM.

3.4.2. State Machine

Der Markenfluss in der Manipulator STM (Abbildung 3.32) teilt sich gleich zu Beginn in zwei parallele Regionen auf, wobei in der oberen Region lediglich ein Zustand vorhanden ist, der immer aktiv ist. In der anderen Region wird die Steering STM aktiviert. Abbildung 3.32 soll darstellen, dass die Steuerung der Aktoren und das Auslesen der Sensoren parallel zum Datenaustausch statt findet.

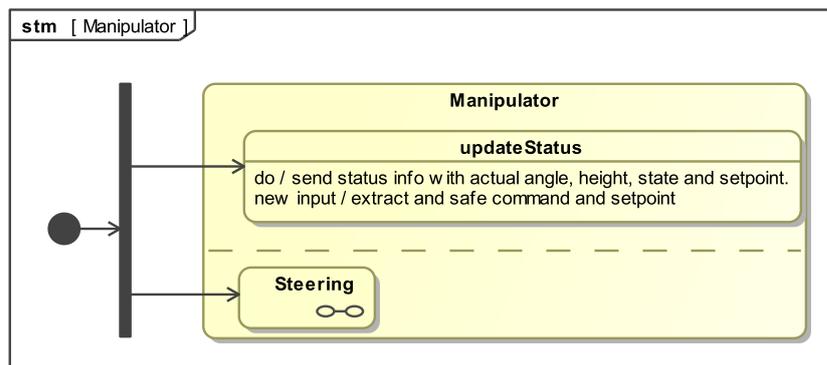


Abbildung 3.32.: STM Manipulator

Der Zustand `updateStatus` ist immer aktiv. Er überprüft, ob neue Steuerkommandos vorhanden und ob diese valide sind. Es gibt drei Kommandos: T für turn, L für lift und S für stop. Auf das Kommando T folgt direkt der Parameter des an zufahrenden Winkels in Grad mit entsprechendem Vorzeichen (vgl. Abbildung 3.33). Es können ganzzahlige Werte von -95 bis +95 als Parameter verwendet werden. Auf das Kommando L folgt ebenfalls direkt ein Parameter, welcher die an zufahrende Höhe in Millimeter angibt. Der Parameter darf ganzzahlige Werte von 0 bis 50 annehmen. Eingaben, die keinem Kommando entsprechen, oder Parameter die außerhalb ihres erlaubten Wertebereiches liegen, werden als invalid bzw. ungültig erklärt. Nach außen hin werden die Statusinformationen in folgender Form als Datenstring gesendet (siehe Abbildung 3.33).

1digit	3digit	1digit	1digit	2digit	1digit	1digit	1digit	3digit	1digit
sign+/-	angle[°]	space	sign+	height[mm]	space	stateInfo	sign+/-	addInfo	\0

Abbildung 3.33.: Statusinfo String

Die Stellen 1 bis 4 geben die aktuellen Winkel mit Vorzeichen in Grad an, 6 bis 8 die aktuelle Höhe in Millimeter, 10 die Statusinformation und 11 bis 14 die Zusatzinformation. Bei einer validen Eingabe wird in der Statusinformation das aktuelle Kommando und in der Zusatzinformation der Sollwert angezeigt. Eine invalide Eingabe wird in der Statusinformation mit einem E für Error angezeigt. Die Zusatzinformation gibt einen Fehlercode an, wobei 001 falsches Kommando und 002 unerlaubter Parameter bedeutet.

Insgesamt ist der Datenstring 15 Zeichen, inkl. der Nullterminierung, lang. Die Nullterminierung wird auf einem seriellen Monitor jedoch nicht angezeigt. Die Funktionen der Arduino Library fügen die Nullterminierung automatisch beim Versenden von Strings an.

Der Ablauf der Steuerung mit Aktoren und Sensoren findet in der Steering STM (Abbildung 3.34) statt.

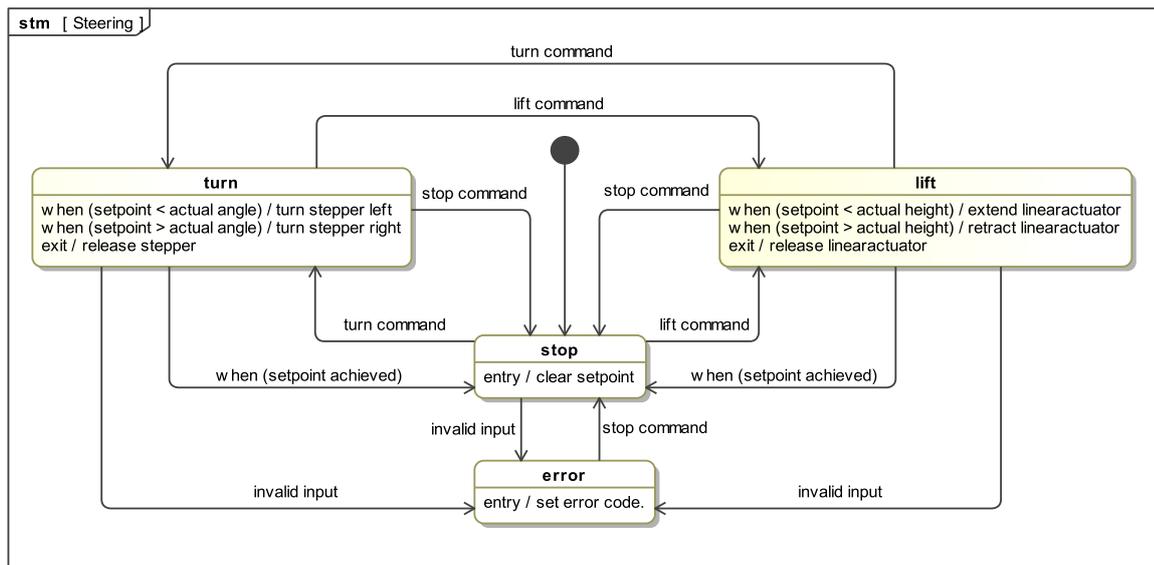


Abbildung 3.34.: STM Steering

Die STM läuft nach dem Prinzip des Markenflusses. Die Marke wandert über die Transitionen von einem Zustand zum nächsten. Es ist immer nur der Zustand aktiv, in dem sich die Marke befindet. Die Transitionen lassen die Marke nur dann passieren, wenn ein bestimmtes Ereignis eintrifft. Das Verhalten eines Zustandes kann intern auch durch Ereignisse beeinflusst werden, was aber nicht zum Verlassen des Zustands führt [OMG17, S.318].

Zu Beginn wird der Zustand stop aktiviert. Bei jeder Aktivierung des Zustands wird einmalig der Sollwert gelöscht. Von hier aus können je nach Kommando die Zustände turn, lift und error erreicht werden. Ist das aktuelle Kommando T, so wird der Zustand turn aktiviert. In dem Zustand wird über ein ChangeEvent der Sollwert mit dem aktuellen Winkel verglichen [Wei14, S.277]. Ist der Winkel kleiner, so soll sich der Schrittmotor nach links, ist er größer nach rechts drehen. Sobald der Sollwert erreicht ist, schaltet die anliegende Transition und die Marke wandert wieder in den Zustand stop. Beim jedem Verlassen von turn wird das Haltemoment des Schrittmotors abgeschaltet, sodass sich dieser frei drehen kann. Ist das aktuelle Kommando L, wandert die Marke in den Zustand lift. Hier wird über ein ChangeEvent der Sollwert mit der aktuellen Höhe verglichen. Ist die aktuelle Höhe geringer, so fährt der Kolben des Linearmotors aus, ist er größer, so fährt er ein. Sobald der Sollwert erreicht ist, wandert die Marke in den Zustand stop. Bei jedem Verlassen von lift, wird der Linearmotor nicht mehr mit Strom versorgt. Dessen Funktionsprinzip hält den Kolben aber an seiner Position. Es ist auch möglich direkt von turn oder lift in den jeweilig anderen Zustand zu wechseln. Die Zustände lift und turn können jederzeit über das Kommando S abgebrochen werden.

Sollte eine invalide Eingabe vorliegen, wird die Marke immer in den Zustand error wandern. Bei der Aktivierung von error wird der entsprechende Fehlercode gesetzt. Nur durch das Kommando S kann der Fehlerzustand quittiert und somit verlassen werden. Die Marke wandert dann wieder in den Zustand stop. Dadurch ist der Anwender gezwungen, die fehlerhafte Eingabe wahrzunehmen.

Zwischen Regionen in einem orthogonalem Zustand darf es formal keine Transitionen geben [Wei14, S.273]. In Abbildung 3.32 wurden zwar keine Transitionen explizit eingetragen, jedoch verwendet die Steering STM Kommandos, die updateStatus empfängt und gleichzeitig sendet updateStatus Statusinformationen über die Steering STM. Die Regionen in einem orthogonalem Zustand können nach außen hin senden, jedoch keinen Austausch untereinander haben. Zur Illustration wird hier jedoch die Annahme getroffen, dass die Kommandos und Statusinformationen in den entsprechenden Regionen zur Verfügung stehen. Die formal korrekte Lösung wäre den Inhalt von updateStatus in jedem einzelnen Zustand der Steering STM mitzuschleifen. Eine andere Modellierung über ein Aktivitätsdiagramm ist auch vorstellbar, da es sich hier aber faktisch um eine STM handelt, soll auch die SysML Notation hierfür verwendet werden.

Der grundsätzliche Sinn und Zweck der SysML Diagramme besteht darin ein Verhalten bzw. Ablauf einfach und kompakt darzustellen. Die Abbildungen 3.32 und 3.34 werden diesem Zweck voll und ganz gerecht, sodass die formale Abweichung ganz bewusst in Kauf genommen wird.

4. Implementierung

Die Implementierung setzt die im Design geplanten Schritte um. Ist das Design gut, verläuft die Implementierung ähnlich wie das Abarbeiten einer Bauanleitung. Manche Aspekte sind jedoch im Design nicht plan- oder vorhersehbar. Wie bereits erwähnt sind das Design und die Implementierung iterativ. Die Implementierung hat also auch rückwirkend einen Einfluss auf das Design. Es bietet sich an die Mechanik und Elektronik bei der Beschreibung der Implementierung gemeinsam, jedoch getrennt von der Software, zu behandeln.

4.1. Mechanik und Elektronik

In diesem Abschnitt erfolgt der Vergleich vom realen Aufbau mit den Zeichnungen aus Abschnitt 3. Darüber hinaus folgen Erläuterungen zur Fertigung und Montage.

4.1.1. Montage

Die Montage ist einer der Punkte, bei denen es in der Implementierung häufig zu Problemen kommt, obwohl dies in der Designphase berücksichtigt wird. Mit dem CAD Programm lassen sich alle Bauteile in sehr beengten Verhältnissen ohne Probleme zusammensetzen. Kann ein Bauteil nicht vernünftig montiert werden, muss das Design geändert werden. Der Entwickler muss nicht nur wissen, welche Werkzeuge es gibt, sondern auch, ob diese vor Ort sind. Ein Beispiel hierfür ist in Abbildung 4.1.

Ganz rechts ist es nicht möglich, den T-Matikverbinder mit einem um 90° abgewinkelten Inbusschlüssel festzuziehen. Ganz links wird ein Inbusschlüssel verwendet der flacher abgewinkelt ist. Hiermit kann der T-Matik Verbinder angezogen werden. In der Mitte ist ein abgerundeter Inbusschlüssel zu sehen, welcher grundsätzlich die Montage in den Profilmuten erleichtert. Mit den abgewinkelten Inbusschlüsseln ist lediglich eine Vierteldrehung möglich, bis der Schlüssel wieder neu angesetzt werden muss. Mit den abgerundeten Enden kann der T-Matik Verbinder bis zur letzten Umdrehung festgezogen werden, ohne den Schlüssel neu ansetzen zu müssen.



Abbildung 4.1.: Schwere Zugänglichkeit bei der Montage

Abbildung 4.2 zeigt wie die selbst schneidenden Gewinde der T-Matik Verbinder mit einem T25 Torx in das Profil gedreht werden.



Abbildung 4.2.: Einsetzen der T-Matik Gewinde mit T25 Torx

Die Stoßverbindungen der T-Matik Verbinder sind nur dann ausreichend fest, wenn die Zugschnitte der Profile plan sind. Mit den beschriebenen Werkzeugen aus Abschnitt 2.1 ist dies nicht möglich. Die Profile müssen vom Händler auf die passende Länge zugeschnitten sein.

4.1.2. Vergleich mit Design

Abbildung 4.3 zeigt den komplettierten MAN, wie er in der Realität implementiert und bereits auf das FZG geschraubt wurde. Vergleicht man Abbildung 4.3 und Abbildung 3.4, kann man erkennen, dass die Implementierung und das Design übereinstimmen.

Ebenso stimmen Abbildungen 3.6 und 4.4 überein. In Abbildung 4.4 ist zusätzlich die Steuerung und Leistungselektronik bzw. das Motorshield sowie der Notausschalter als Sicherheitsmaßnahme zu erkennen. Das Motorshield reicht alle Pinkontakte des Unoboards durch. Auf dem Shield sind ebenfalls Lötaugen vorhanden, die zur Verbindung einer 3-poligen Stiftleiste zum Anschluss an das Schiebepotentiometer und eine 4-polige Stiftleiste für die Winkelsensorik genutzt werden. Die Gegenstücke der Stiftleisten können nur in einer Orientierung aufgesteckt werden. Diese Steckkontakte sind also verpolungssicher. Auf die H-Brücken der Leistungselektronik wurden zusätzliche Kühlkörper mit Wärmeleitkleber befestigt.

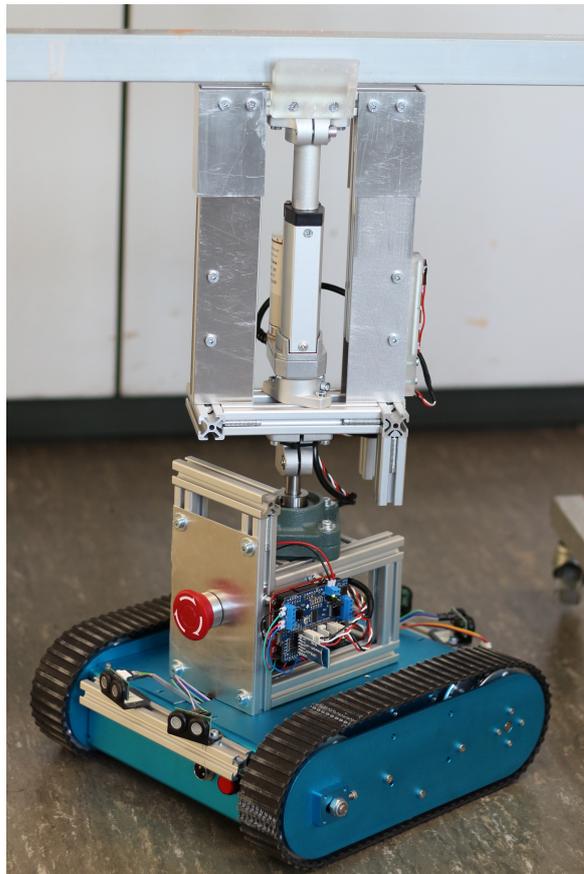


Abbildung 4.3.: Implementierung Manipulator mit Fahrzeug

Für Evaluationszwecke wurde zudem eine 4-polige Buchsenleiste mit GND , TX , RX und V_{CC} auf das Shield gelötet, um dort ein HC-05 Bluetoothmodul aufzustecken. Dieses Modul erzeugt eine virtuelle, serielle Verbindung über Bluetooth z. B. zu einem andere PC. Dort kann man über einen seriellen Monitor die Statusinformationen abrufen und Steuerkommandos an den MAN schicken. Diese Buchsenleiste kann darüber hinaus auch für eine kabelgebundene Übertragung genutzt werden. Der USB Port des Unoboards kann zur Programmierung und ebenfalls zum Datenaustausch genutzt werden.

Abbildungen 4.5 zeigt den implementierten Hubarm und stimmen mit Abbildungen 3.10 überein. In Abbildung 4.5 sind zusätzlich die Schutzbleche zum Verdecken der Klemmbereiche (vgl. Abbildung 3.14) angebracht. In Abbildung 4.5 ist auch ein Teil der Kabelführung des Linearmotors zu erkennen. Das Stromversorgungskabel des Linearmotors ist mit einem Kabelbinder am Ganter fixiert.

Abbildung 4.6 zeigt wie die restliche Kabelführung mit Kabelschellen in der Profilmutter umgesetzt wurde. Die Kabelführung ist ein Punkt, der lediglich durch die Rotationsanschlüsse im

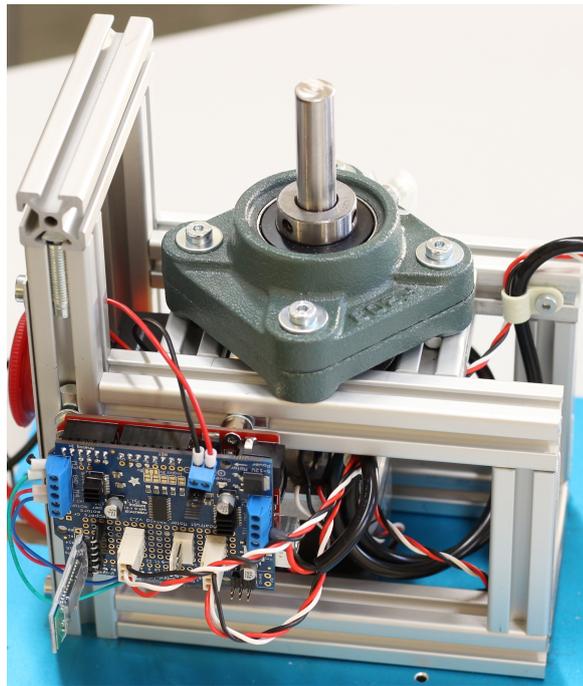


Abbildung 4.4.: Implementierung Untergestell

Design berücksichtigt wurde. Die Kabelführung lässt sich im Detail nur schwer im Vorfeld planen. Es ist viel effektiver diese erst bei der Implementierung am fertigen Prototypen zu verlegen. Da der Rahmen vollständig aus Strangpressprofilen besteht, kann die Kabelführung, durch die Montage in der Profilvernut, der Kontur des MAN angepasst werden. Hierbei wird darauf geachtet, dass die Kabel nicht auf Zug beansprucht werden und die Biegeradien während der Bewegung möglichst groß bleiben. Zudem darf kein Kabel an einer Kante hängen bleiben oder gequetscht werden.

Abbildung 4.7 zeigt den Endeffektor mit Schulterblechen zum Schutz vor Einklemmungen. Der Vergleich mit Abbildung 3.12 zeigt, dass auch dieser Implementierungsschritt dem Design entsprechen konnte.

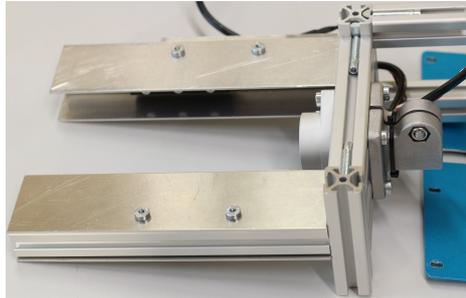


Abbildung 4.5.: Implementierung Hubarm



Abbildung 4.6.: Kabelführung

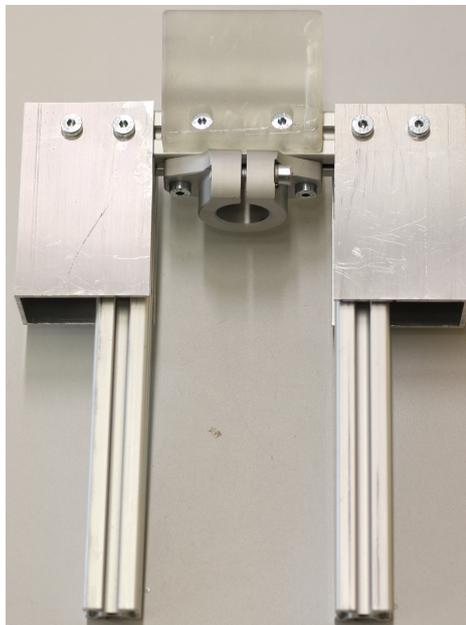


Abbildung 4.7.: Implementierung Endeffektor

4.1.3. PCB Fertigung

Abbildung 4.8 zeigt die einzelnen Schritte bei der Anfertigung der Winkelsensorik. Hierfür wird die Tonerstransfermethode angewendet. Das Platinenlayout aus Abschnitt 3.3.2 wird mit einem Laserdrucker spiegelverkehrt auf ein geeignetes Transfermedium gedruckt. Dies kann z. B. eine Katalogseite des Elektronikhändlers REICHELT sein. Der Druck wird ausgeschnitten und mit einem modifizierten Laminiergerät bis 200° erhitzt und auf einen Platinenrohling gewalzt. Der Rohling besteht aus Epoxyd und ist mit einer $35\ \mu\text{m}$ starken Kupferauflage beschichtet. Durch die Temperatur und den Druck schmilzt der Toner und bleibt auf der Kupferschicht haften. Nachdem das Papier mit Wasser von dem Rohling entfernt wurde, bleibt nur noch der Toner übrig. Die Kontur des Breakoutboards mit $42\ \text{mm} \times 42\ \text{mm}$ kann nun aus dem Rohling geschnitten werden. In Abbildung 4.8 links oben liegt der Ausschnitt in einem Natriumpersulfat-Säurebad, welches auf ca. 40°C beheizt ist. Überall dort, wo der rote Toner auf dem Kupfer haftet, kann die Säure das Kupfer nicht auflösen. Nach ca. einer halben Stunde ist der Ätzvorgang beendet. Der Toner wird nun mit Aceton von der Platine entfernt. Das Resultat ist in Abbildung 4.8 oben in der Mitte zu sehen. Grundsätzlich ist es nicht nö-

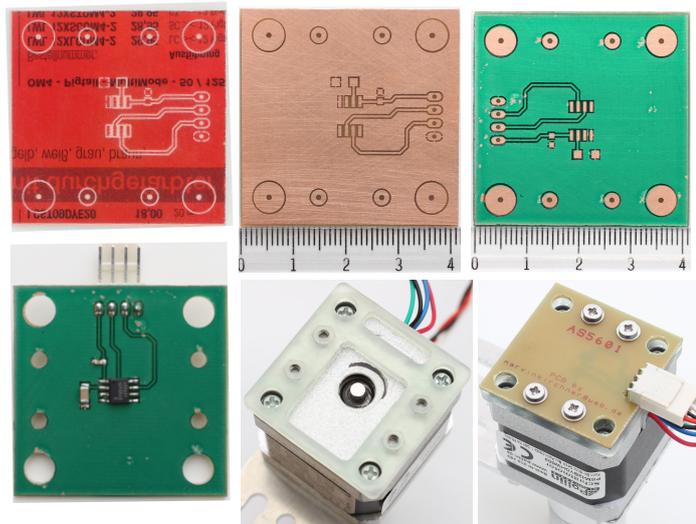


Abbildung 4.8.: Fertigungsschritte der Winkelsensorik

tigt die Platine mit einem Lötstopplaminat zu überziehen, da alle Komponenten von Hand verlötet werden. Die Lötstopmmaske ist aber von Vorteil, denn sie hindert das Lot daran Kontaktbrücken zu benachbarten Pads zu erzeugen und schützt die Leiterbahnen vor äußeren Einflüssen. Das Lötstopplaminat wird mit ca. 120°C auf die Platine laminiert. Das Negativ der Lötstopmmaske wird auf eine Transparentfolie gedruckt und auf die Platine gelegt. Nach ca. 1 min Bestrahlung mit UV-Licht muss das Laminat ca. 1 h polymerisieren. In einer Lösung aus Natrium Carbonate löst sich das Laminat an den Stellen, die nicht mit UV-Licht bestrahlt

wurden. Das Ergebnis ist in Abbildung 4.8 oben rechts zu sehen. Die Bestückung der Platine samt Bohrungen ist in Abbildung 4.8 unten links zu sehen. In Abbildung 4.8 unten Mitte ist die PCB Halterung aus Abbildung 3.27 an den Schrittmotorflansch und ein Diametralmagnet an die Motorwelle geklebt. Abbildung 4.8 unten rechts zeigt wie das Breakoutboard an den Schrittmotor befestigt ist. Dies ist eine sehr platzsparende und robuste Implementierung des Winkelsensors.

4.2. Software

In diesem Abschnitt werden ausgewählte Teile des Quellcodes erläutert. Der vollständige Quellcode befindet sich auf der beigelegten CD in dem Verzeichnis `CD:/Quellcode/`. Zur Programmierung wird die Arduino IDE, mit integriertem seriellen Monitor, verwendet. Die IDE verwendet bei der Erstellung eines neuen Projektes Dateien mit der Endung `.ino` und nicht, wie gewohnt `.cpp` oder `.c`. Diese IDE hat den Vorteil, dass sie genau auf die Plattformen von Arduino und auch dessen Clone zugeschnitten ist. Über den USB Port des Unoboards können Programme sehr bequem hochgeladen werden. Voraussetzung ist, dass auf dem Chip des Boards ein Bootloader vorhanden ist, welcher bei dem Kauf vom Fachhändler aber immer vorhanden ist. Eine weitere Eigenart der IDE ist die augenscheinlich fehlende `main()` Funktion. Die Philosophie, welche hinter Arduino steckt, ist das Programmieren von Mikrocontrollern für jedermann so einfach wie möglich zu gestalten und den Programmierer von komplizierten Codezeilen zu befreien. Offenbar gehört hierzu auch die `main()`, welche im Hintergrund durch die IDE mit eingebunden wird. Die IDE hat die weiteren Vorteile, dass sie kostenlos, einfach zu bedienen, sehr populär ist und sich dadurch ein umfangreicher Support durch Internetforen bietet.

Die Funktionen `setup()` und `loop()` werden standardmäßig beim Erstellen eines neuen Projektes von der IDE vorgegeben. Das Hauptprogramm befindet sich im Verzeichnis `CD:/Quellcode/manipulator_control.ino`. Die Zeilennummern der kommenden Quellcodeauszüge sind nicht identisch mit den Zeilennummern in den Sourcedateien. Die original Kommentare werden in den Auszügen weggelassen, um die Darstellung kompakter zu halten.

In Quellcode 4.1 ist zu sehen, dass das Hauptprogramm aus nur wenigen Zeilen besteht. In Zeile 3 wird ein Pointer für ein Manipulatorobjekt erstellt. Der Scope dieses Pointers ist global. Aus dem Block von `setup()` und `loop()` kann hierauf zugegriffen werden. Würde der Pointer innerhalb des Blockes von `setup()` initialisiert werden, hätte man innerhalb des Blockes von `loop()` keinen Zugriff mehr auf den Pointer. Der Konstruktor der Manipulatorklasse initialisiert alle verwendeten Objekte und Schnittstellen und die serielle Verbindung mit Hilfe der Arduino Library. Die Konstruktoren der erzeugten Objekte führen wiederum Initialisierungen durch, wie z. B. die I2C Verbindung. Die Initialisierungen der Arduino Library

Quellcode 4.1: Hauptprogramm

```
1 #include "Manipulator.h"
2
3 Manipulator *manipulator;
4
5 void setup() {
6     manipulator = new Manipulator();
7 }
8
9 void loop() {
10    manipulator->manipulatorSTM();
11 }
```

müssen in dem Block von `setup()` erfolgen, um z. B. die Klasse `Serial` benutzen zu können. In Zeile 22 wird ein `Manipulator` Objekt erstellt und somit auch dessen Konstruktor aufgerufen. Die Adresse des `Manipulator` Objektes wird dem Pointer `manipulator` zugewiesen. `loop()` ist eine Endlosschleife in welcher die STM über die Methode `manipulatorSTM()` gestartet wird.

Nach Abbildung 3.32 teilt sich die STM in einen orthogonalen Zustand mit zwei Regionen auf. Es folgt der Code aus dem Verzeichnis `CD:/Quellcode/Manipulator.cpp`. In Quellcode 4.2 startet die Methode `manipulatorSTM()` wiederum zwei Methoden, die den Regionen aus Abbildung 3.32 entsprechen. Dessen Implementierung wird in den folgenden Abschnitten erläutert.

Quellcode 4.2: ManipulatorSTM

```
1 void Manipulator::manipulatorSTM(void){
2
3     updateStatus();
4
5     steeringSTM();
6 }
```

4.2.1. updateStatus

Hier wird die obere Region aus Abbildung 3.32 beschrieben. Es handelt sich um nur einen Zustand, der immer aktiv ist. Er hat die Aufgabe dauerhaft Statusinformationen nach außen hin zu senden und Kommandos über die serielle Schnittstelle zu empfangen. Die Ausgabe erfolgt gemäß Abbildung 3.33 mit `Serial.println(outputString)`.

In Quellcode 4.3 ein Ausschnitt der Kommandoabfrage.

Quellcode 4.3: Processupdate

```
1 if(Serial.available()){
2     cmd = Serial.peek();
3
4     if('S' == cmd){
5         // Do nothing.
6     }
7     else if( ('T' == cmd) && (!invalidInput) ){
8         setpoint = Serial.parseInt();
9         additionalInfo = setpoint;
10
11         if( !( (LEFT_BORDER >= setpoint) && (setpoint >= RIGHT_BORDER) ) ){
12             invalidInput = true;
13             additionalInfo = 002;
14         }
15     }
```

`Serial.available()` hat einen Rückgabewert größer 0, wenn Daten im seriellen Eingangspuffer vorhanden sind. Der Rückgabewert von `Serial.peek()` ist das erste Byte aus dem Eingabepuffer. Das Protokoll der Kommandos sieht vor, dass dies immer das Kommando ist (vgl. Abschnitt 3.4.2). Das Byte wird auf Gleichheit mit den erlaubten Kommandos überprüft. In Zeile 168 wird überprüft, ob das Kommando T ist und keine unquitierte Falscheingabe vorliegt. Erst dann wird der Sollwert über `Serial.parseInt()` ausgelesen. Diese Methode extrahiert den ersten Integerwert aus einem String. Je nach Kommando sind unterschiedliche Eingaben für den Sollwert möglich. Eine solche Kontrolle ist hier z. B. in Zeile 172. Sollte der entsprechende Parameter ungültig sein, wird der Zusatzinformation der Fehlercode 002 zugewiesen. Sollte der Kommandovergleich kein bekanntes Kommando feststellen, ist das Kommando ungültig und der Zusatzinformation wird der Fehlercode 001 zugewiesen. Bei einer korrekten Eingabe bleibt der Sollwert nach Zeile 170 in der Zusatzinformation gespeichert. Schlussendlich wird der Eingabepuffer geleert, indem `Serial.read()` so lange aufgerufen wird, bis nichts mehr im Eingabepuffer vorhanden ist.

4.2.2. steeringSTM

Hier wird die untere Region aus Abbildung 3.32 beschrieben. Die `steeringSTM` regelt das Zusammenspiel der Sensoren und Aktoren in Hinblick auf die Kommandos. Die Zustände werden als `enum` Variablen in der Headerdatei `CD:/Quellcode/Manipulator.h` deklariert. Für die Implementierung von STM werden häufig zwei `switch-case` verwendet [Leh14].

In einem switch-case werden die Transitionen auf ihre Schaltfähigkeit überprüft, evtl. Aktionen durchgeführt, die nur einmal beim Übergang ausgeführt werden und der Markenfluss geregelt bzw. der entsprechende Zustand gesetzt. In dem anderen switch-case werden die Aktionen, die während der Zustand aktiv ist, ausgeführt. Quellcode 4.4 zeigt einen Ausschnitt der steeringSTM für den Markenfluss. Mit switch-case wird überprüft, welcher Zustand mo-

Quellcode 4.4: Regelung des Markenflusses

```
1 case turn:
2     if(invalidInput){
3         stepper->release();
4         state_steeringRegion = error;
5         statusInfo = 'E';
6     }
7     else if( ( (actualAngle() <= setpoint + 1)
8             && (actualAngle() >= setpoint - 1))
9             || (cmd == 'S')){
10        stepper->release();
11        state_steeringRegion = stop;
12        setpoint = 0;
13        additionalInfo = setpoint;
14        cmd = 'S';
15        statusInfo = 'S';
16    }
17    else if(cmd == 'L'){
18        stepper->release();
19        state_steeringRegion = lift;
20        statusInfo = 'L';
21    }
22    break;
23 }
```

mentan aktiv ist. In Quellcode 4.4 ist dies der Zustand turn. Innerhalb eines cases wird überprüft, ob eine Transition schaltfähig ist und dies zum Verlassen des Zustandes führt. Eine ungültige Eingabe macht z. B. in Zeile 2 die Transition zum Zustand error schaltfähig, sodass die Marke dorthin fließt bzw. der Zustand error in Zeile 4 gesetzt wird. Eine Aktion, die einmalig beim Verlassen von turn ausgeführt wird ist das Abschalten des Haltemoments des Schrittmotors, hier in den Zeilen 3, 10 und 18. Da eine Marke nur von Zustand zu Zustand fließen kann, ist dies auch immer automatisch eine Aktion, die beim setzen eines anderen Zustandes ausgeführt wird.

In Quellcode 4.5 ist ein Ausschnitt aus dem anderen switch-case. Hier wird überprüft welcher Zustand aktiv ist und dessen Aktionen durchgeführt. Aber auch Zustände können intern Kontrollstrukturen aufweisen [OMG17, S.318]. Ist der Zustand lift aktiv wird dies mit dem

Quellcode 4.5: Aktionen in den Zuständen

```

case lift:
  if(setpoint < actualHeight(A0)){
    linearactuator->run(BACKWARD);
  }
  else if(setpoint > actualHeight(A0)){
    linearactuator->run(FORWARD);
  }
  break;

```

1
2
3
4
5
6
7
8

switch-case in Quellcode 4.5 abgefangen. In dem Zustand wird überprüft, ob der aktuelle Winkel größer oder kleiner als der Sollwert ist. Entsprechend wird der Motor vor- oder zurück gedreht.

4.2.3. Auswertung der Sensoren

Hier wird die Auswertung der Sensordaten vorgestellt. Der Winkel wird über I2C vom AS5601 mit der Methode `actualAngle()` abgefragt. Die Arduino Library bietet dazu die Klasse `Wire` an. Abbildung 4.9 gibt die Abfolge der Datenübertragung an.

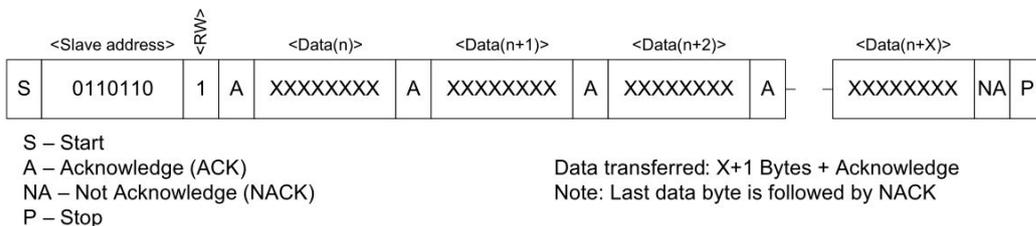


Abbildung 4.9.: Protokoll zur I2C Datenübertragung mit AS5601 [AMS17]

In Quellcode 4.6 sind die Schritte nach Abbildung 4.9 umgesetzt. In Zeile 4 wird die Verbindung zu AS5601 hergestellt, indem er über seine Adresse angesprochen wird. Als nächstes wird der Registerpointer des AS5601 auf das Register mit dem aktuellen Winkel geführt. Bis hierhin agiert der AS5601 als slave receiver. Die Verbindung wird gestoppt und in Zeile 8 erneut aufgenommen. Diesmal agiert der AS5601 als slave transmitter und übermittelt den Winkel aufgeteilt in zwei Bytes. In Zeile 12 wird der digitale Wert in ganzzahlige Winkel in Grad umgerechnet. Eine Multiplikation des Digitalwertes mit der Auflösung $2^{11} = 2048 \frac{STEPS}{180^\circ}$ liefert einen Winkel in Grad. Zur Anpassung an das Design (vgl. Abbildung 3.13) wird der Winkel um 180° verschoben. Der Nullpunkt sollte mechanisch so genau wie möglich eingestellt werden, eine Offsetverschiebung bietet hier eine nachträglich Kalibrierung. Um der

Quellcode 4.6: Auswertung des Winkelsensors

```

1 int Manipulator::actualAngle(){
2     int angle = 0;
3
4     Wire.beginTransmission(AS5601_I2C_ADDRESS);
5     Wire.write(byte(ANGLE_REGISTER));
6     Wire.endTransmission();
7
8     Wire.requestFrom(AS5601_I2C_ADDRESS, NUMB_REQUESTED_BYTES);
9     angle = Wire.read();
10    angle = angle << 8;
11    angle |= Wire.read();
12    angle = (int) (-(angle * DEG_PER_STEP - SHIFT - OFFSET_DEG) );
13    return angle;
14 }

```

Abbildung 3.13 genüge zu tun, wird noch das Vorzeichen gewechselt. Da eine ganzzahlige Winkelauflösung ausreichend ist, werden alle Nachkommastellen des Winkels durch ein casting auf integer verworfen. In Quellcode 4.4 Zeile 7 bis 9 wird die Funktion `actualAngle()` angewendet. Der Schrittmotor ist durch sein Funktionsprinzip nicht in der Lage jeden Winkel anzufahren. Die anzufahrende Position muss mit einer Unschärfe von $\pm 1^\circ$ angegeben werden. Der Endeffektor kann also mit einer Genauigkeit von $\pm 1^\circ$ positioniert werden.

Die Erfassung, wie weit der Endeffektor ausgefahren ist, wird mit der Methode `actualHeight()` abgefragt. Quellcode 4.7 zeigt die Implementierung von `actualHeight()`. Der analoge Pin des Unoboards, an dem das Schiebepotentiometer ab-

Quellcode 4.7: Auswertung des Höhensensors

```

1 int Manipulator::actualHeight(unsigned char analogPin){
2     static int smoothValue;
3     smoothValue = 0.4 * smoothValue + 0.6 * analogRead(analogPin);
4     return smoothValue * MM_PER_STEP - OFFSET_MM;
5 }

```

gegriffen wird, ist über den Parameter angegeben. Oft weist der digitalisierte Spannungswert ein Zittern bzw. jitter auf, dessen Ursachen hier nicht erläutert werden sollen. Um stabile Werte zu erhalten, muss eine Gewichtung zwischen dem alten und neuen Wert, wie in Zeile 3, hergestellt werden. Damit die Methode den Wert nicht „vergisst“, wird die Variable, die den digitalisierten Spannungswert abgespeichert als `static` deklariert. Bei der Gewichtung zählt der alte Wert immer etwas mehr als der neue. Je nachdem wie stark das jitter ist, muss die Gewichtung angepasst werden. In Summe muss die Gewichtung immer 1 ergeben.

Für die Umrechnung des Digitalwertes in einen Verfahrweg in Millimeter, wird eine Multiplikation mit $0,059 \text{ STEP/mm}$ durchgeführt. Dieser Wert wurde durch Versuche ermittelt. Zum Ausgleich von Montagetoleranzen, wird der Nullpunkt über einen Offset kalibriert. Der Linearmotor kann Positionen mit einer Auflösung von unter einem Millimeter anfahren. Der Höhenwert kann also hart auf Gleichheit überprüft werden.

5. Testen

In diesem Abschnitt werden die durchgeführten Tests kurz beschrieben. Genauere Informationen liefert das Testtemplate im Anhang im Verzeichnis CD:/Testtemplate.pdf. Im Anhang befindet sich ebenfalls ein Video in dem Verzeichnis CD:/Videos/Manipulator.mp4, in dem die Tests teilweise dokumentiert sind. In Tabelle 5.1 ist gekennzeichnet, wie Testfälle den Requirements zugeordnet werden. Für manche Requirements ist es nicht nötig, bzw. möglich Testfälle zu erstellen, da diese bereits durch das Design und oder die Implementierung bestätigt sind.

Tabelle 5.1.: Trace Matrix Testfälle

	T1	T2	T3	T4	T5	T6	T7	T8	T9
Req. 2.1.	x								
Req. 3.1.		x							
Req. 3.2.			x						
Req. 3.2.1.				x					
Req. 3.3.					x				
Req. 4.1.						x			
Req. 4.2.							x		
Req. 4.3.								x	
Req. 7.									x

Hier ist eine kurze Auflistung der durchgeführten Tests:

- T1: Testen des Bewegungsverhaltens des GS.
- T2: Testen der Rotationsfunktion.
- T3: Testen der Hubfunktion.
- T4: Schutz vor Verletzungen überprüfen.
- T5: Ausgleich von Positionierungsfehlern testen.
- T6: String für Statusinformationen überprüfen.
- T7: Verhalten des MAN bei Kommandos überprüfen.
- T8: Testen der Fehlercodes.
- T9: Freie Drehbarkeit des Hubarmes testen.

6. Fazit und Ausblick

Schlussbetrachtung

In der Schlussbetrachtung kann man sagen, dass die Problemstellung gelöst wurde. Der MAN erfüllt seinen Zweck. Herr Christophers, wissenschaftlich Mitarbeiter der HAW Hamburg, wurde als primärer Stakeholder zufriedengestellt. Der MAN befähigt das FZG eine gekennzeichnete RT im 7. Stock der HAW Hamburg zu bewegen. Ein gutes Design nach dem Prinzip KISS macht ein Produkt einfach und funktional. Eine gute Vorbereitung spart bei der Implementierung Zeit, Mühe und Geld. Allerdings birgt jedes Projekt seine eigenen, ganz individuellen Herausforderungen. Je intensiver man sich mit einer Problemstellung beschäftigt, desto mehr gibt es, das man beachten kann bzw. muss. Möchte man ein Projekt von Anfang bis Ende vernünftig planen und durchführen, sollte man realistische Zielvorstellungen haben. Es kommen fast immer unvorhergesehene Komplikationen hinzu oder die Bearbeitung erweist sich als umfangreicher, als zuvor erwartet.

Ausblick

Für die Zukunft könnte der MAN um folgende Funktionen erweitert werden:

- Drucksensoren: Um den nötigen Anpressdruck zu erzeugen, muss die RT nicht ganz angehoben werden. Es würde ausreichen, wenn der Anpressdruck auf das FZG etwas mehr als 40 N betragen würde. Ein Dehnungsmessstreifen könnte dies detektieren. Das FZG und der MAN würden dann entlastet werden, wobei die Stabilität des MAN absolut ausreichend ist.
- Kontaktsensoren: Der MAN erhält Kommandos und gibt Rückmeldung, ob er das Kommando ausgeführt hat. Man weiß jedoch nicht, ob das Kommando seinen Zweck erfüllt hat. Über Kontaktschalter könnte man herausfinden, ob der Endeffektor die RT auch wirklich getroffen hat oder die RT bei einer Kollision mit der Umwelt vom Endeffektor gesprungen ist. Grundsätzlich sollte das FZG über Sensoren verfügen, welche ohnehin die Geometrie der RT erkennen, da das FZG die RT im autonomen Betrieb selbst finden muss. Ein Kontaktsensor wäre also ein Zusatz.

Diese Erweiterungen könnten wiederum von Studenten durchgeführt werden, sodass der MAN in seiner Daseinsberechtigung als Lern- und Übungsplattform gestärkt werden würde.

Literatur

- [Abu17] Jutta Abulawi. *Konstruktion Maschinenelemente*. Vorlesungsmaterial. 2017.
- [Ada17] Lady Ada. *Adafruit Motor Shield V2 for Arduino Datasheet*. adafruit. Oktober 2017. URL: <https://learn.adafruit.com/adafruit-motor-shield-v2-for-arduino/overview>.
- [Ama17] Amazon. *Linearmotor Bild*. Dezember 2017. URL: <https://www.amazon.de/ECO-WORTHY-Stellantrieb-Linearantrieb-Multi-Funktion-Chromatisches/dp/B014SJWXHQ>.
- [AMS17] AMS. *AS5601 12-bit Programmable Contactless Encoder Datasheet*. AMS. Dezember 2017. URL: <http://ams.com/eng/Products/Magnetic-Position-Sensors/Angle-Position-0n-Axis/AS5601>.
- [Cen17] Conrad Technologie Centrum. *C-Control OPEN UNO*. Conrad Technologie Centrum. Dezember 2017. URL: www.conrad.de.
- [Ele17] TT Electronics. *Slide Potentiometer Datasheet*. TT Electronics. Oktober 2017. URL: http://www.produktinfo.conrad.com/datenblaetter/675000-699999/675430-da-01-en-60MM_SING_SCHIEBEPOTI_PS60_50K_20__LIN.pdf.
- [Feh93] Jürgen Fehlau. „Einführung in DIN EN 1993 (EC 3)“. In: *Beuth-Hochschule für Technik Berlin* (1993). URL: <http://docplayer.org/26618530-Inhaltsverzeichnis-einfuehrung-in-din-en-1993-ec-3-1-einfuehrung-in-din-en-1993-ec-3-1-1-allgemeines-ueber-eurocodes.html>.
- [Fis11] Ulrich Fischer. *Tabellenbuch metall (mit Formelsammlung)*. Europa-Lehrmittel, 2011.
- [Gro+15] Dietmar Gross u. a. *Technische Mechanik 3: Kinetik*. Springer-Verlag, 2015.
- [ite17] item. *Strangpressprofil Datasheet*. item. Oktober 2017. URL: http://catalog.item24.de/images/medienelemente/DOK/DATA/DOK_DATA_profil-tech-daten__SDE__AIN__V2.pdf.
- [Leh14] Thomas Lehmann. *Software und Systems Engineering*. Vorlesungsmaterial. 2014.

- [Mäd17] Mädler. *Kugel-Flanschlager UCF (Grauguss) Datasheet*. Mädler. 2017. URL: <http://info.maedler.de/product/1643/1629/2184/kugel-flanschlager-ucfl-graugussl>.
- [Nae09] Paul Naefe. „Einführung in das methodische konstruieren“. In: *Vieweg+ Teubner GWV Fachverlage GmbH, Wiesbaden* (2009).
- [OMG17] OMG. *OMG Unified Modeling Language Version 2.5*. Dezember 2017. URL: <http://www.omg.org/spec/UML/About-UML/>.
- [Pol17a] Pollin. *Schrittmotor PSM42BYGHW603 Bild*. Dezember 2017. URL: <https://www.pollin.de/p/schrittmotor-psm42byghw603-1-80-310453>.
- [Pol17b] Pollin. *Schrittmotor PSM42BYGHW603 Datasheet*. Pollin. Oktober 2017. URL: <https://www.pollin.de/productdownloads/D310453D.PDF>.
- [ROB17] NEXUS ROBOT. *New Tracked Mobile Robot Tank Kit Datasheet*. NEXUS ROBOT. Oktober 2017. URL: <http://robotchassisparts.com/wp-content/uploads/2016/09/New-Tracked-Mobile-Robot-Tank-Kit-10022.pdf>.
- [San99] Ben Zion Sandler. *Robotics: designing the mechanisms for automated machinery*. Academic Press, 1999.
- [SKF17] SKF. *Zulässige Momentbelastung*. Nov. 2017. URL: <http://www.skf.com/de/products/bearings-units-housings/super-precision-bearings/axial-radial-crb/permmissible-moment-load/index.html>.
- [Wei14] Tim Weilkiens. *Systems Engineering mit SysML/UML: Anforderungen, Analyse, Architektur. Mit einem Geleitwort von Richard Mark Soley*. dpunkt. verlag, 2014.
- [WMR15] Herbert Wittel, Wilhelm Matek und Hermann Roloff. *Maschinenelemente-Normung, Berechnung, Gestaltung*. Springer Vieweg, 2015.

A. Anhang

A.1. Inhalt der beigefügten CD

- Bachelorthesis
- CAD Daten
- Stückliste
- Testtemplate
- Schaltpläne
- Quellcode
- Videodateien
- Datenblätter

A.2. Verwendete Geräte und Software

- Inventor Professional 2018
- Inkscape
- MagicDraw
- Arduino IDE
- DesignSpark PCB
- Digital Multimeter MASTECH MS8229
- Kofferwaage Luxebell

Glossar

Endeffektor Ein Endeffektor ist das letzte Glied in der Bewegungskette eines Manipulators, welches in direktem Kontakt zu dem zu manipulierenden Objekt steht. Dies kann z. B. ein Werkzeug oder Greifer sein [[San99](#)].

Gesamtsystem Das Gesamtsystem ist das Fahrzeug, der Manipulator und die Rolltafel im Verbund.

Manipulator Ein Manipulator ist ein Mechanismus, der gewöhnlich aus mehreren Segmenten besteht, die sich relativ zueinander bewegen können, wie z. B. drehen oder gleiten, um Objekte zu greifen oder zu bewegen [[San99](#)].



Erklärung zur selbstständigen Bearbeitung einer Abschlussarbeit

Gemäß der Allgemeinen Prüfungs- und Studienordnung ist zusammen mit der Abschlussarbeit eine schriftliche Erklärung abzugeben, in der der Studierende bestätigt, dass die Abschlussarbeit „– bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit [(§ 18 Abs. 1 APSO-TI-BM bzw. § 21 Abs. 1 APSO-INGI)] – ohne fremde Hilfe selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt wurden. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich zu machen.“

Quelle: § 16 Abs. 5 APSO-TI-BM bzw. § 15 Abs. 6 APSO-INGI

Dieses Blatt, mit der folgenden Erklärung, ist nach Fertigstellung der Abschlussarbeit durch den Studierenden auszufüllen und jeweils mit Originalunterschrift als letztes Blatt in das Prüfungsexemplar der Abschlussarbeit einzubinden.

Eine unrichtig abgegebene Erklärung kann -auch nachträglich- zur Ungültigkeit des Studienabschlusses führen.

Erklärung zur selbstständigen Bearbeitung der Arbeit

Hiermit versichere ich,

Name: Kirchner

Vorname: Marvin

dass ich die vorliegende Bachelorarbeit bzw. bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit – mit dem Thema:

Entwicklung eines Manipulators mit Endeffektor für eine autonome, mobile Roboterplattform.

ohne fremde Hilfe selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

- die folgende Aussage ist bei Gruppenarbeiten auszufüllen und entfällt bei Einzelarbeiten -

Die Kennzeichnung der von mir erstellten und verantworteten Teile der -bitte auswählen- ist erfolgt durch:

Hamburg

Ort

04.01.2018

Datum

Unterschrift im Original