

Forschungsbereich
Maschinenbau Informatik
und **Virtuelle**
Produktentwicklung

Univ.-Prof. Dr.-Ing. Detlef Gerhard

Handhabungsgerät für Werkstückträger Konzeption, Aufbau und Inbetriebnahme

Diplomarbeit

Daniel Wondrak

Matrikel-Nr.: 9871404

Inhaltsverzeichnis

Abkürzungsverzeichnis	III
1 Einleitung	1
1.1 Aufgabenstellung	1
1.2 Aufbau der Diplomarbeit	2
2 Technische Grundlagen	3
2.1 Einführung & Theorie zu Handhabungssystemen	3
2.1.1 Definition Werkstückhandhabung	3
2.1.2 Auswahl eines geeigneten Handhabungssystems	3
2.1.3 Werkstückträger	5
2.1.4 Flexible Ordnungssysteme	6
2.2 Fördertechnik	9
2.2.1 Stetigförderer	9
2.2.2 Unstetigförderer	13
2.3 Industrieroboter	16
2.3.1 Theorie	16
2.3.2 Hersteller und Geräte	16
2.3.3 Einsatzmöglichkeiten	17
2.3.4 Greifer	18
2.4 Individuelles Beschickungssystem	19
2.4.1 Lineare Bewegung (Arm des Handhabungsgeräts)	19
2.4.2 Spannsysteme	22
2.4.3 Elektromotoren	24
2.5 Steuerung und Regelung	25
2.5.1 Systeme der Werkstückhandhabung	25
2.5.2 CAM	25
2.5.3 SPS	25
2.5.4 Raspberry Pi	27
2.5.5 Arduino	28
2.5.6 Python	28
2.5.7 Telegram Messenger	29

3	Verworfenne Konzepte	30
3.1	Konzept Scherengitter	30
3.1.1	Geometrieberechnung	30
3.1.2	Kräfte in der Gabel	32
3.1.3	CAD-Modell	34
3.1.4	Geschwindigkeit	34
3.1.5	Ideenbewertung	34
3.2	Konzept Förderband	35
3.3	Konzept Regalbediengerät (Version 1 und 2)	36
3.3.1	CAD-Modell	36
3.3.2	Variante 1	36
3.3.3	Variante 2	37
4	Realisiertes Konzept	38
4.1	Konzept der Fertigungszelle	38
4.2	Aufbau und Funktionsweise	39
4.2.1	Zugekaufte Teile	39
4.3	Berechnungen zur Konstruktion und Auslegung	42
4.3.1	Schrankeneinheit	42
4.3.2	Getriebemotor	43
4.3.3	Durchbiegung der Außenschiene	44
4.4	Steuerung	46
4.4.1	Steuerung des Handhabungsgeräts durch den Chatbot	46
4.4.2	Programmierung	49
4.4.3	Zeitmessung	51
5	Schlussbetrachtung	53
5.1	Sicherheitskonzept	53
5.2	Steuerung	54
5.2.1	Telegram Chatbot	54
6	Verzeichnisse	56
	Tabellenverzeichnis	56
	Abbildungsverzeichnis	57
	Literaturverzeichnis	59
	Stichwortverzeichnis	62
7	Anhang	63
7.1	Schaltplan	63
7.2	Programmcode	65

Abkürzungsverzeichnis

AGV Automated Guided Vehicle

Abb. Abbildung

BDE Betriebsdatenerfassung

bzw. beziehungsweise

CAD Computer-Aided Design/rechnerunterstütztes Konstruieren

CPU Central Processing Unit/Zentrale Verarbeitungseinheit

exkl. exklusiv

FTF Fahrerloses Transportfahrzeug

FTS Fahrerloses Transportsystem

ggf. gegebenenfalls

GPIO General Purpose Input/Output

HAT Hardware attached on top

Mio. Millionen

RAM Random-Access Memory

RBG Regalbediengerät

ROM Read-Only Memory

SPS Speicherprogrammierbare Steuerung

TCP Tool Center Point

USB Universal Serial Bus

usw. und so weiter

WLAN Wireless Local Area Network

z.B. zum Beispiel

Kurzfassung/Abstract

Kurzfassung (in deutscher Sprache)

Ziele und Grundkonzept

Das Hauptziel dieser Diplomarbeit ist, ein Handhabungsgerät zur Be- und Entladung einer Portalfräse zu konzipieren und anschließend aufzubauen. Dazu musste im Vorfeld ein Konzept für den Aufbau einer flexiblen Fertigungszelle gefunden werden. Da eine große Werkstückdiversität prognostiziert wird, werden Werkstückträger eingesetzt, auf denen die Werkstücke vorher eingespannt werden. Für den Weg des Werkstückträgers bis zum Handhabungsgerät gibt es mehrere Möglichkeiten, welche im Abschnitt 4.1 näher erläutert werden. Eine Möglichkeit besteht darin, das Handhabungsgerät auf ein Fahrerloses Transportfahrzeug (FTF) zu montieren. Um dies zu ermöglichen, wurde auf eine Kompatibilität mit einem FTF geachtet: Mit dem Handhabungsgerät wird ausschließlich per Funk über WLAN kommuniziert, wodurch eine weitere Verkabelung außer der Stromversorgung von 12V DC nicht notwendig ist. Dieser Strom kann aus dem Akku des FTF bezogen werden.

Konstruktion

In Abbildung 4.1 auf Seite 38 ist die Konstruktion abgebildet. Sie besteht aus zwei Innenschienen, die in beide Richtungen ausfahren können. Schranken an beiden Enden der Schiene fixieren das Werkstück. Das Werkstück kann so verschoben werden. Zur Verminderung des Reibwiderstandes wurde der Boden des Handhabungsgeräts mit vier Röllchenleisten bestückt. Die Schiene wird von je zwei Getriebemotoren bewegt, welche sich auf beiden Seiten des Handhabungsgeräts befinden. Zur Positionsbestimmung der Innenschiene dienen insgesamt vier Schalter, die an den Ecken des Handhabungsgeräts befestigt sind. Die vier Schranken werden durch Servomotoren bewegt.

Steuerung

Zur Steuerung wurde ein Raspberry Pi am Handhabungsgerät angebracht. Dieser ist über WLAN verbunden. Der Benutzer kommuniziert mit dem Handhabungsgerät über den Instant Messenger Telegram mit einem am Raspberry implementierten Chatbot.

Abstract (in English)

Objective and basic concept

The objective of this thesis is to create a handling device to load or unload a CNC portal milling. In advance a research was done to make a concept for the flexible production cell with the portal milling. The portal milling machine has to handle with very different work pieces. So it was decided to use a work piece holder. The work pieces have to be fixed on the piece holder before the production process starts. There is more than one possibility how the work piece holder can be transported to the handling device. One possibility is to put the handling device on an Automated Guided Vehicle (AGV). Therefore the compatibility between the handling device and the AGV is important: The communication between user and handling device is only done via WIFI. Therefore the power supply of 12V DC is the only needed physical connection.

Construction

The construction can be viewed in figure 4.1 on page 38. There are two telescopic slides which can go in both directions. Barriers at the ends can be closed to fix the work piece. When the work piece is fixed, it can be transported. To reduce friction there are roller strips at the bottom of the handling device. The slides are moved by gear motors. To determine the position of the slides there are four switches, one on each corner of the handling device. The barriers are moved by servomotors.

Controlling

A Raspberry Pi controls the device. It is connected to a WIFI network. The user communicates with the device with the Instant Messenger Telegram with a chat bot, which runs on this device.

Kapitel 1

Einleitung

1.1 Aufgabenstellung

Für den Aufbau einer flexiblen Fertigungszelle mit einer Portalfräse der Firma CNC-Step (Abbildung 1.1) wird ein Handhabungssystem benötigt, um dieses mit Rohmaterialien, Werkstücken und ggf. Werkzeughaltern be- und entladen zu können. Im Rahmen der Diplomarbeit soll das Handhabungssystem entwickelt und in Betrieb genommen werden. Dazu ist zunächst der Materialfluss für das Be- und Entladen zu definieren und eine geeignete Prinziplösung auf Basis einer Recherche am Markt verfügbarer Lösungskomponenten durchzuführen. Anschließend ist das Konzept auszuarbeiten. Nach Beschaffung der benötigten Komponenten sollen diese in Betrieb genommen werden. Für die Steuerung der Einheit ist dabei eine entsprechende Software- bzw. Steuerungsumgebung erforderlich.



Abbildung 1.1: CNC-Portalfräse der Firma CNC-Step [4]

Nach einer eingehenden Analyse wurde beschlossen, Werkstücke auf einem Werkstückträger zu bewegen. Die Werkstückträger sind 300x300mm groß und maximal 10kg schwer. Die Kosten dieses Projekts sollten 5.000€ nicht überschreiten. Der Werkstückträger trägt Werkstücke aus Kunststoff, Holz, Aluminium oder vergleichbare Werkstoffe.

1.2 Aufbau der Diplomarbeit

Im **Kapitel 2** werden die technischen Grundlagen erläutert. Dabei wird ein breites Themenspektrum von Handhabungs- und Beschickungssystemen über Industrieroboter bis zu Steuerungs- und Regelungsmöglichkeiten abgedeckt.

Im Rahmen der Diplomarbeit wurden mehrere Konzepte erarbeitet, wie ein dementsprechendes Handhabungssystem funktionieren könnte. Die nicht realisierten Konzepte werden im **Kapitel 3** vorgestellt.

Im **Kapitel 4** wird das realisierte Konzept vorgestellt und näher auf die Konstruktion sowie die Programmierung eingegangen.

Kapitel 2

Technische Grundlagen

In diesem Kapitel wird der aktuelle Stand der Wissenschaft und Technik zu Handhabungssystemen erläutert. Im ersten Abschnitt gibt es eine allgemeine Einführung zu den Handhabungssystemen sowie deren Anforderungen. In den darauffolgenden Abschnitten werden die verschiedenen Konzepte und Techniken näher erläutert. Im letzten Abschnitt dieses Kapitels wird auf die für das gewählte Konzept passenden Steuerungsmöglichkeiten eingegangen.

2.1 Einführung & Theorie zu Handhabungssystemen

2.1.1 Definition Werkstückhandhabung

Man unterscheidet im Materialfluss Fördern, Lagern (Speichern) und Handhaben. Das Handhaben ist in der VDI-Richtlinie 2860 näher definiert¹. Nach dieser VDI-Richtlinie ist Handhaben *das Schaffen, das definierte Verändern oder das vorübergehende Aufrechterhalten einer vorgegebenen räumlichen Anordnung von geometrisch bestimmten Körpern in einem Bezugssystem*. [23, S.8]

2.1.2 Auswahl eines geeigneten Handhabungssystems

Es gibt unzählige Handhabungssysteme. Um das passende System auszuwählen und zu entwickeln, müssen zunächst die Randbedingungen untersucht werden, wie z.B.: [16, S. 73]

- Bearbeitungszeit pro Fertigungsschritt
- Stückzahl pro Jahr bzw. Losgrößen
- Variantenanzahl des Produktes
- Platzbedarf

¹Die VDI-Richtlinie 2860 wurde im Juni 2016 zurückgezogen (www.vdi.de)

- Investitionsaufwand
- Flexibilität (rasche Umrüstbarkeit auf andere Werkstücke und/oder Betriebsmittel)

Ein übliches Handhabungsgerät für Fräsmaschinen hat folgende Eigenschaften: [16, Abb. 4/1/1]

Kinematik: Standgeräte und Portalgeräte sind üblich, Anbaugeräte unüblich

Mobilität: ortsfest üblich, ortsveränderlich unüblich

Anzahl der Achsen: 4-5 üblich, mehr als 5 unüblich

Gerätekoordination: kartesisch, zylindrisch, kugelförmig oder gelenkig

Antriebe: elektrisch üblich, hydraulisch/pneumatisch unüblich

Tragfähigkeit: bis 500 N üblich, über 500N unüblich

Steuerung: PTP üblich, MP unüblich, CP nicht bekannt

DNC-Betrieb: unüblich

Programmiersysteme: Teach-In üblich, Offline unüblich

Verfahrensgeschwindigkeit: 0,1 bis 1 $\frac{m}{s}$ üblich

Positioniergenauigkeit: weniger als 1 mm üblich

Sensoren: keine üblich, taktil unüblich

Ein Handhabungsgerät im industriellen Einsatz sollte dabei folgenden Anforderungen genügen: [16, S.9]

- geringe Umstellungszeiten und -kosten
- kurze Durchlaufzeiten bei maßgeschneiderten Produkten
- zeitlich und inhaltlich definierte Reaktion auf Marktveränderungen
- Reduktion von Informations- und Organisationskosten
- hoher zeitlicher Nutzungsgrad der Fertigungsanlagen
- bedienarme Fertigung in der 2. und 3. Schicht
- Abbau von Kontrolltätigkeiten

- Vermeidung manueller Wiederhol Tätigkeiten
- Zuverlässigkeit trotz häufiger Umstellungen
- wettbewerbsfähige Fertigungskosten auch bei kleineren Losgrößen

2.1.3 Werkstückträger

Eine Möglichkeit ist, die Be- und Entladung über Werkstückträger zu realisieren. Ein Werkstückträger ist eine Palette mit einer Aufspannvorrichtung. Diese dient als einheitliche Schnittstelle zwischen der Portalfräse und dem Beschickungssystem.

Das Werkstück wird zu Beginn in den Werkstückträger eingespannt und durchläuft dann auf dem Werkstückträger die Produktion. Eine Skizze eines Systems mit Werkstückträgern ist in Abbildung 2.1 zu sehen. [17]

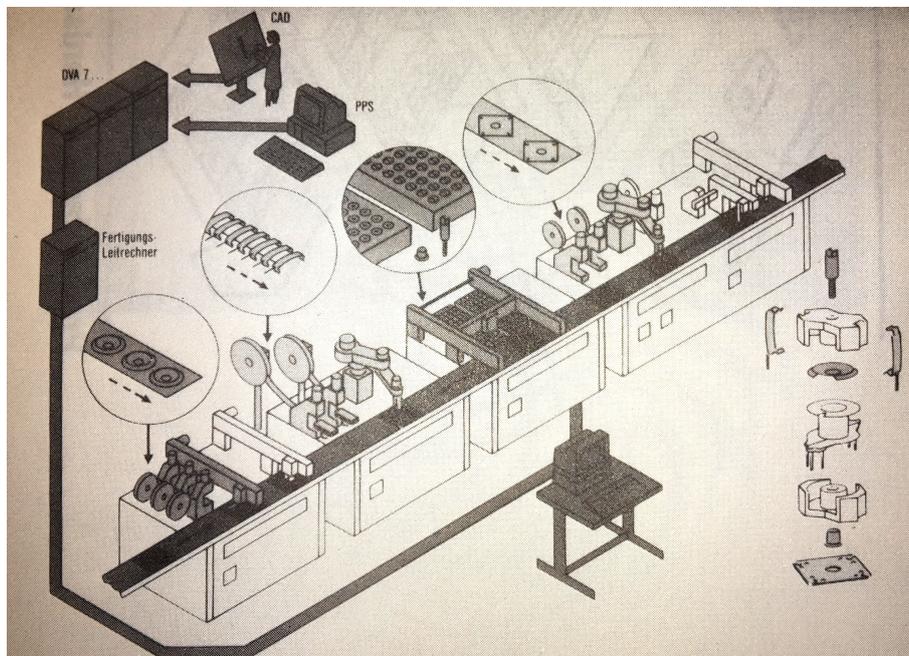


Abbildung 2.1: Montagelinie mit Werkstückträger [17, S.123]

Ob ein System mit Werkstückträger sinnvoll ist, hängt im wesentlichen von der Losgröße, der Teilevielfalt und dem Ähnlichkeitsgrad der Werkstücke ab. Bei **kleiner Losgröße, hoher Teilevielfalt und geringem Ähnlichkeitsgrad** werden am besten Werkstückträger eingesetzt. Bei **hohen Stückzahlen und geringer Teilevielfalt** wird am besten ein frei programmierbares Beschickungsgerät eingesetzt. Dieses entnimmt das Werkstück einem Speicher und legt es in die Spannvorrichtung. Zusätzlich kann auch ein Greiferwechselsystem installiert werden. Dies kann zum Beispiel mit einem Portalroboter realisiert werden (Abbildung 2.2). [16, S.18]

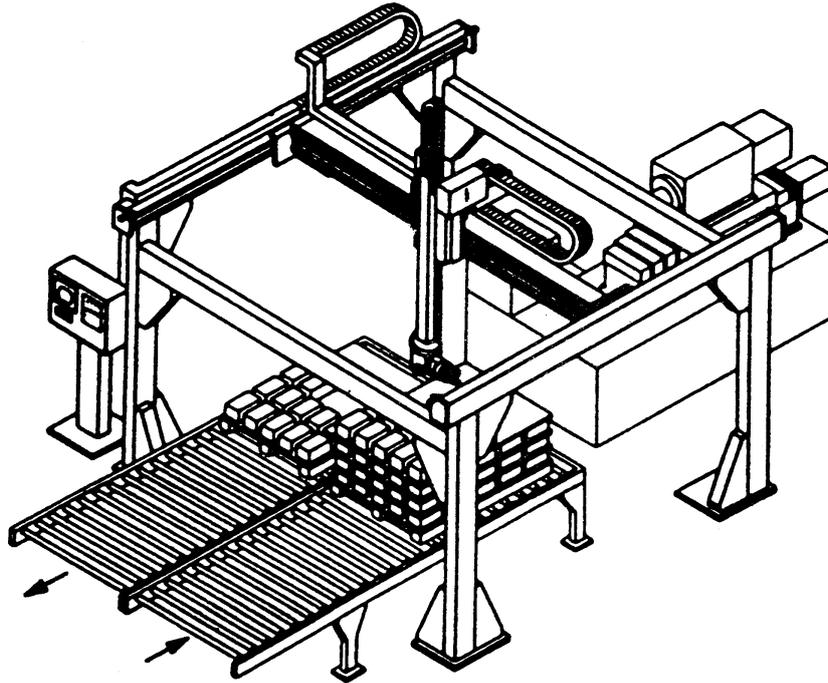


Abbildung 2.2: Portalroboter [16, S.19]

2.1.4 Flexible Ordnungssysteme

Der Vorteil flexibler Ordnungssysteme ist, dass sie sich an unterschiedliche Werkstücktypen und Handhabungsaufgaben anpassen lassen. Im Buch *Rechnerunterstützte Bereitstellung von ungeordneten Kleinteilen für die automatisierte Montage* [23, S.11-S.14] werden vier flexible Ordnungssysteme vorgestellt.

Linearförderer

Funktionsweise: In der Orientierungsstrecke gibt es mehrere werkstückabhängige Orientierungselemente

Eigenschaften: einfacher Aufbau, geringer Platzbedarf, kurze Umrüstzeiten, niedrige Kosten

Entwickler: Universität Salford in England

Abbildung: 2.3-a

Vibrationswendelförderer mit Solid-State-Kamera

Funktionsweise: Das Werkstück wird fotografiert. Ist es richtig positioniert, wird es weiterbefördert. Ist die Orientierung falsch, so wird es in den Topf zurückbefördert.

Eigenschaften: hoher Aufwand, niedrige Austragsleistung

Entwickler: J.W.Hill

Abbildung: 2.3-b

Vibrationsförderer mit mehreren Ebenen

Funktionsweise: Das Werkstück wird an eine Ecke der Werkstückfassungseinheit geschoben, um die möglichen Orientierungsrichtungen zu reduzieren. Anschließend wird es von einer Kamera erfasst. Ist die Orientierung richtig, wird es weiter in einen der bis zu fünf Ebenen befördert. Ist die Orientierung falsch, so kommt es zurück in den Quelltopf.

Eigenschaften: geringer Platzbedarf, gute Wiederverwendbarkeit der Systemkomponenten, hohe Komplexität, Werkstückabhängigkeit, unflexible Umrüstung

Entwickler: Hitachi

Abbildung: 2.3-c

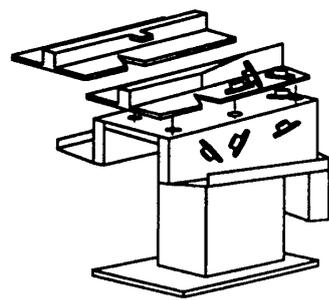
Doppelrichtungsförderer

Funktionsweise: Das System besteht aus zwei endlosen Förderbahnen, welche sich in entgegengesetzte Richtungen bewegen. Ein Sensor (Lichtschranke) misst bei der oberen Förderbahn, ob die Orientierung stimmt. Stimmt sie nicht, so wird das Werkstück auf die zweite Förderbahn zurückgeworfen. Nur wenn die Orientierung stimmt darf das Werkstück passieren.

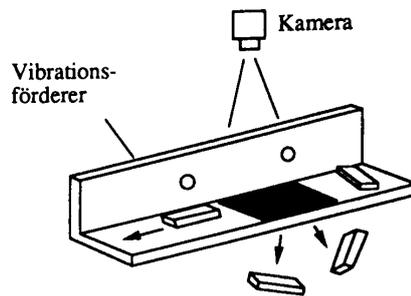
Eigenschaften: hohe Austragsleistung, gute Wiederverwendbarkeit der Systemkomponenten, Verklemmungsgefahr der Werkstücke, großer Platzbedarf

Entwickler: Massachusetts-Universität in Amherst

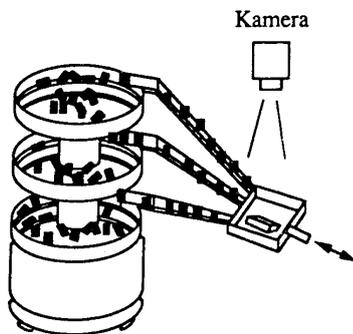
Abbildung: 2.3-d



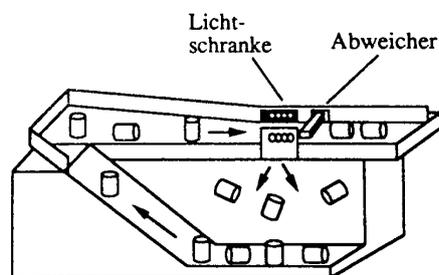
a: Linearförderer



b: Vibrationsförderer mit Kamera



c: Hitachi-System



d: Doppelrichtungsförderer

Abbildung 2.3: Beispiele für flexible Ordnungssysteme [23, S.14]

2.2 Fördertechnik

In der Fördertechnik unterscheidet man Stetigförderer und Unstetigförderer. In diesem Abschnitt werden insbesondere jene Konzepte näher vorgestellt, die als Ideengrundlage für ein Handhabungssystem dienen können. Als Literaturgrundlage dient das Buch „Transport- und Lagerlogistik“ [24].

2.2.1 Stetigförderer

Ein Stetigförderer arbeitet kontinuierlich. Daraus ergibt sich ein gleichbleibender Transportgutstrom. Charakteristisch für Stetigförderer sind der geringe Energiebedarf, die große Betriebssicherheit und die einfache Bauweise. Ihre Automatisierung ist einfach zu erreichen. Man unterscheidet Stetigförderer für Stückgut und für Schüttgut. Beispiele für Stetigförderer für Stückgut sind:

- Bandförderer
- Gliederförderer
- Rutschen
- Pneumatische Förderer
- Schleppkettenförderer
- Kreisförderer
- Umlaufförderer
- Rollenförderer
- Tragkettenförderer
- Sortierförderer

Im Allgemeinen wird die Gruppe der Stetigförderer in Bandförderer, Gliederbandförderer und Rutschen unterteilt.

Dimensionierungsgrundlage

Der Gesamtwiderstand F_w setzt sich aus dem Hubwiderstand F_{wh} und dem Reibungswiderstand F_{wR} zusammen. Nach Heinrich Martin [24] werden die Widerstände folgendermaßen berechnet:

$$F_{wh} = \frac{\dot{m}gH}{3.6v} [N] \quad (2.1)$$

$$F_{wR} = f_{ges}L \left(m_1g + \frac{\dot{m}g}{3.6v} \right) [N] \quad (2.2)$$

$$f_{ges} = \frac{F_R}{F_N} \quad (2.3)$$

Die Gesamtreibungskraft F_w ist die Summe der beiden Kräfte: $F_w = F_{wR} \pm F_{wh}$

F_{wh}	in N	Hubwiderstand
\dot{m}	in t/h	Massenstrom
g	in m/s ²	Erdbeschleunigung
H	in m	Hubhöhe
v	in m/s	Transportgeschwindigkeit
f_{ges}		Gesamtreibungszahl
F_R	in N	Gesamtreibungskraft
F_n	in N	Gesamtreibungsnormalkraft
F_{wR}	in N	Gesamtreibungswiderstand
L	in m	Transportlänge
m_1	in kg/m	längenbezogenes Eigengewicht des Transportmittels
\dot{m}	in t/h	Massenstrom
v	in m/s	Transportgeschwindigkeit

Tabelle 2.1: Formelzeichenerklärung

Bandförderer

Am häufigsten wird der Bandförderer als Gummigurtförderer ausgeführt. Der Gummigurtförderer hat die unterschiedlichsten Anwendungsgebiete, unter anderem auch die Fließfertigung. Die Vorteile sind: [24]

- einfache Bauart
- hohe Transportgeschwindigkeit
- geringer Energiebedarf
- universell einsetzbar
- geeignet für kurze als auch für lange Transportwege
- schonender Transport
- Gefälle- und Steigungsmöglichkeit

Die Antriebe von Bandförderern sind meistens auf der Kopfseite (Gutabwurf) angeordnet. Bei kleinen Gurtförderern können sie sich auch am Schluss (Gutaufgabe) befinden. In Abbildung 2.4 sind Möglichkeiten der Anbringung der Antriebstrommel skizziert.

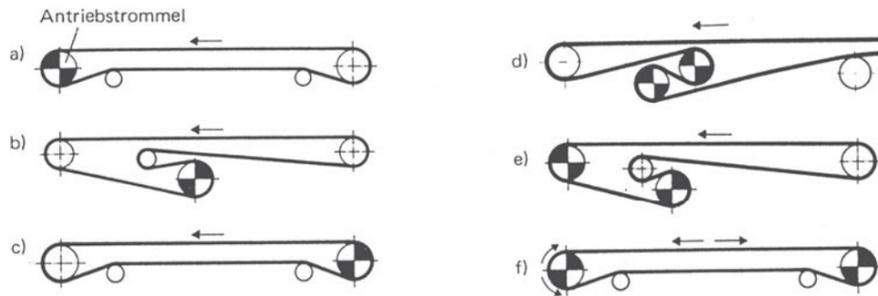


Abbildung 2.4: Anordnungsmöglichkeiten von Antrieben bei Gummigurtförderern [24, S.139]

Um den Riemen zu spannen werden zwei grundsätzliche Arten unterschieden: Starre Spannanlagen und selbstständige Spannanlagen. Starre Spannanlagen eignen sich vor allem für kleine Fördermittel und werden mittels Spannwinde oder Schraubenspindel realisiert (siehe Abbildung 2.5 a). Selbstständige Spannanlagen können mittels Schwerkraft, pneumatisch, hydraulisch oder elektrisch betrieben werden (Abbildung 2.5 b-e).

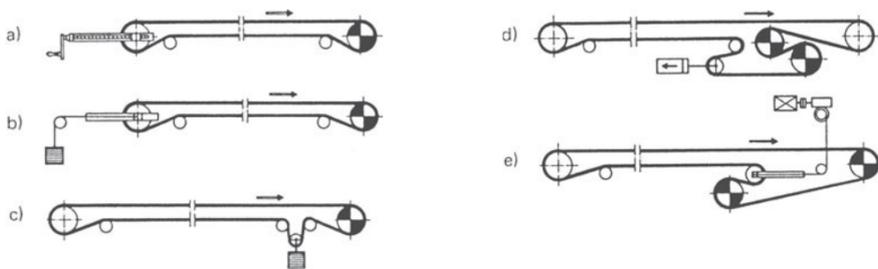


Abbildung 2.5: Spannvorrichtungen bei Gummigurtförderern [24, S.140]

Eine besondere Bauform der Bandförderer ist der **Riemenförderer**. Dieser kann für verschiedenste Stückgüter verwendet werden, so auch für große und flache Güter wie Glasscheiben. Als Riemen können Zahn-, Rund- und Keilriemen verwendet werden. Zum Ausschleusen von Stückgütern können auch Hubriemenförderer verwendet werden. Der Hubantrieb erfolgt pneumatisch, mechanisch oder hydraulisch. (Abbildung 2.6)

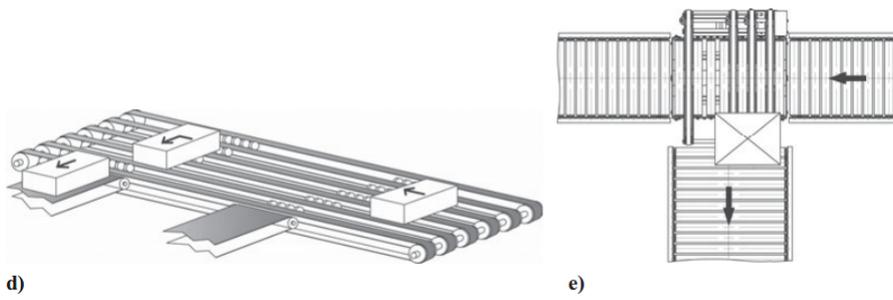


Abbildung 2.6: Riemenförderer [24, S.151]

2.2.2 Unstetigförderer

In Abbildung 2.7 wurde die Unterteilung der Unstetigkeitsförderer abgebildet.

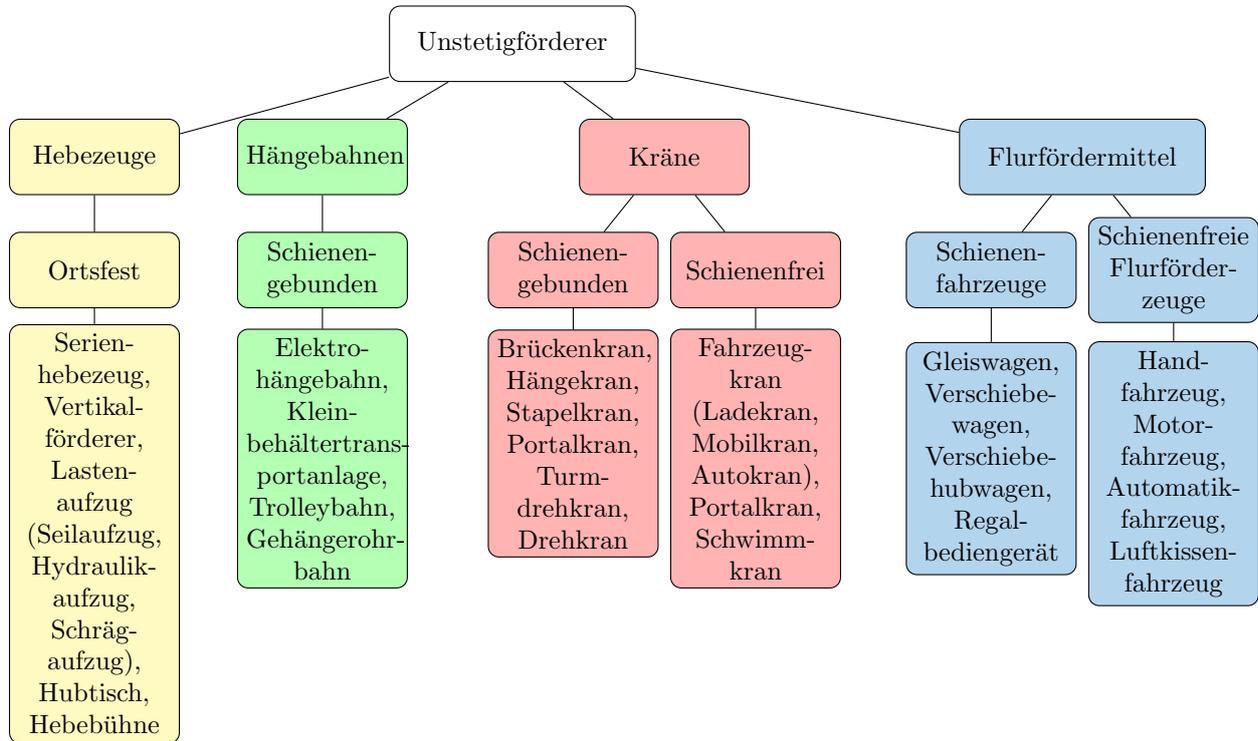


Abbildung 2.7: Einteilung der Unstetigförderer [24, S.214]

Gabelstapler

Gabelstapler gehören in die Gruppe der motorisch angetriebenen, schienenfreien Flurförderzeuge. Der Gabelstapler besitzt ein Hubgerüst, welches vertikale Lastbewegungen ausführen kann. Dieses Hubgerüst kann einfach, zweifach oder auch dreifach ausgeführt sein. Das Zweifachteleskop-Hubgerüst besteht aus einem festen Außenmasten und einem teleskopierbaren Innenmasten. Freisichthubgerüste haben entweder eine spezielle Kettenübersetzung (Abbildung 2.8-c) oder zwei parallel am Mastprofil angebrachte Hubzylinder.

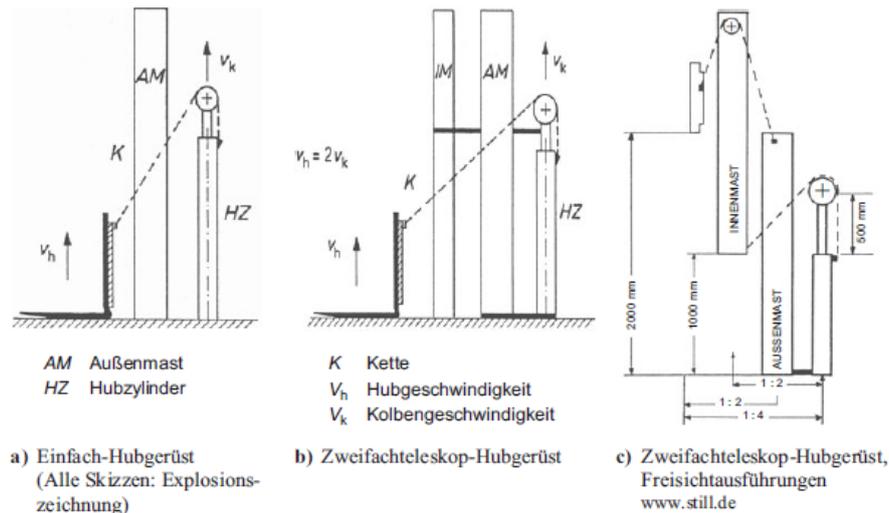


Abbildung 2.8: Hubgerüstkonstruktionen [24, S.243]

Fahrerlose Transportfahrzeuge

Ein Fahrerloses Transportfahrzeug (FTF) dient dazu, ohne Personal Güter von einem Ort zum anderen zu bewegen. Es gibt unzählige Arten von fahrerlosen Transportfahrzeugen. Ein FTF ist die Hauptkomponente in einem fahrerlosen Transportsystem (FTS). Jedoch bezeichnet ein Fahrerloses Transportsystem (FTS) das ganze System mit samt seinen Fahrzeugen, der Leitsteuerung und der Bodenanlage. [28, S.19]

Angefangen hat die Entwicklung von FTFs 1950. Damals wurden sie hauptsächlich in der Automobilindustrie eingesetzt. Um Personal einzusparen werden fahrerlose Transportfahrzeuge mittlerweile in fast allen Branchen eingesetzt, was auch die große Variantenvielfalt erklärt. In der Literatur werden 11 Kategorien von FTFs genannt: [28, S.140ff]

1. **Gabelhub-FTF, eigens konstruiert:** Die Lastaufnahme erfolgt in der Bodenebene mittels Gabeln. Die Last sind üblicherweise Paletten bis 1 t.
2. **Gabelhub-FTF als automatisiertes Seriengerät:** Wie 1., nur ein in Serie hergestelltes Gerät.
3. **Huckepack-FTF:** Diese sind ähnlich zum Gabelhub-FTF, sie können aber das Lademittel nicht vom Boden aufnehmen, sondern es muss eine bestimmte Standard-Übergabehöhe eingehalten werden.
4. **Schlepper:** Dieses FTF zieht mehrere Anhänger mit 5 t Gewicht.
5. **Unterfahr-FTF:** Es unterfährt Rollcontainer und hebt sie für den Transport an.
6. **Montage-FTF:** Diese FTF werden in der Serienmontage eingesetzt.

7. **Schwerlast-FTF:** Transportiert werden Stahlcoils oder schwere Papierrollen bis 35 t.
8. **Mini-FTF:** Mini-FTFs sind aktuell wenig verbreitet. Viele kleine, schnelle, intelligente und flexible Fahrzeuge sollen z.B. bei Kommissionieraufgaben unterstützend wirken. Es gibt keine zentrale FTS-Leitsteuerung, die FTFs kommunizieren intelligent miteinander. Hier sind vor allem preiswerte Lösungen gefragt, die allerdings aufgrund der erforderlichen Sicherheitstechnik für den Personenschutz wie Laserscanner schwierig zu erreichen sind.
9. **PeopleMover:** Ein FTF zur Beförderung von Fahrgästen
10. **Diesel-FTF:** Outdoorfahrzeuge, die meist dieselektrisch angetrieben sind.
11. **Sonder-FTF:** Sonderlösungen für Sonderaufgaben

2.3 Industrieroboter

Industrieroboter können zur Handhabung, Montage oder Bearbeitung von Werkstücken eingesetzt werden. Einmal programmiert können sie Arbeitsabläufe selbstständig durchführen.

2.3.1 Theorie

Ein Industrieroboter führt einen Effektor durch den Raum. Der **Effektor** ist jener Teil des Roboterarms, der mit der Umgebung in Kontakt tritt, um das Werkstück aufzunehmen. Dieser Teil wird auch Greifer genannt. Man definiert einen charakteristischen Punkt des Effektors und bezeichnet diesen als **Tool Center Point (TCP)**. Bewegungsbeschreibungen in Koordinaten definieren diesen Punkt als Bezugspunkt. [29, S.15]

Regelungstypen

Man unterscheidet die Kraftregelung und Lagerregelung. Die **Kraftregelung** regelt die Kräfte und Drehmomente, welche auf die Arbeitsumgebung ausgeübt werden. Die **Lagerregelung** regelt die Bewegung des Roboters. Eine Mischform zwischen diesen beiden Regelungsarten ist die **hybride Regelung**. Jeder Industrieroboter muss über eine Lagerregelung verfügen. Zusätzliche Kraftregelungen sind seltener vorhanden. [29, S.21]

Lagebeschreibung

Man unterscheidet Weltkoordinaten und Roboterkoordinaten. Zur Lagebeschreibung in **Weltkoordinaten** von Robotereffektoren, Armteilen, etc. werden raumfeste, kartesische Koordinaten (Rechtssystem) verwendet. Es gelten die üblichen Rechenregeln für Koordinaten. Unter **Roboterkoordinaten** versteht man die relativen Koordinaten der Gelenke des Roboters. Möchte man durch die Roboterkoordinaten die Weltkoordinaten berechnen, so muss eine Vorwärtstransformation der Koordinaten durchgeführt werden. Der umgekehrte Vorgang von Weltkoordinaten zu Roboterkoordinaten heißt **Rückwärts-Transformation**. [29, Kapitel 2-3]

2.3.2 Hersteller und Geräte

Hier werden zwei Industrieroboter der unteren Preiskategorie vorgestellt, die auch als Handhabungssystem dienen könnten.

Industrieroboter Franka Emika der Firma KBee AG

Der Industrieroboter Franka Emika ist ein kollaborativer Industrieroboter. Sein Roboterarm hat eine sehr gute Beweglichkeit und Feinfühligkeit, somit ist es möglich ohne

Schutzzäune direkt mit Menschen zusammenarbeiten. [12] Der Preis eines Komplettpakets liegt bei 21.900 €. Es können bis zu 3kg gehoben werden. Die Reichweite beträgt 80cm. [3]



Abbildung 2.9: Industrieroboter Franka Emika [3]

Industrieroboter Einsteigerpaket RH-3FH von Mitsubishi Electric

Ebenfalls ein günstiges Einsteigerpaket wird von der Firma Geva angeboten. Hier gibt es ein Einsteigerpaket für einen Industrieroboter von Mitsubishi Electric für 14.990 €. Die Traglast beträgt auch bei diesem Industrieroboter 3kg. Die Reichweite ist mit 550/150mm angegeben. [6]

2.3.3 Einsatzmöglichkeiten

Ein Industrieroboter kann ein Werkstück aufnehmen und der Fräse übergeben. Ein andere Möglichkeit ist auch, dass der Industrieroboter während des Fräsvorgangs das Werkstück festhält. Diese Variante wird meist dann gewählt, wenn das zu bearbeitende Werkstück leichter als das Werkzeug ist. [14]

2.3.4 Greifer

Es gibt über 300 unterschiedliche Greiferlösungen. In Abbildung 2.10 ist die Vorkommenshäufigkeit der wichtigsten Greiferarten dargestellt. Die wichtigsten Greiferarten sind der Scherengreifer und der Sauggreifer.

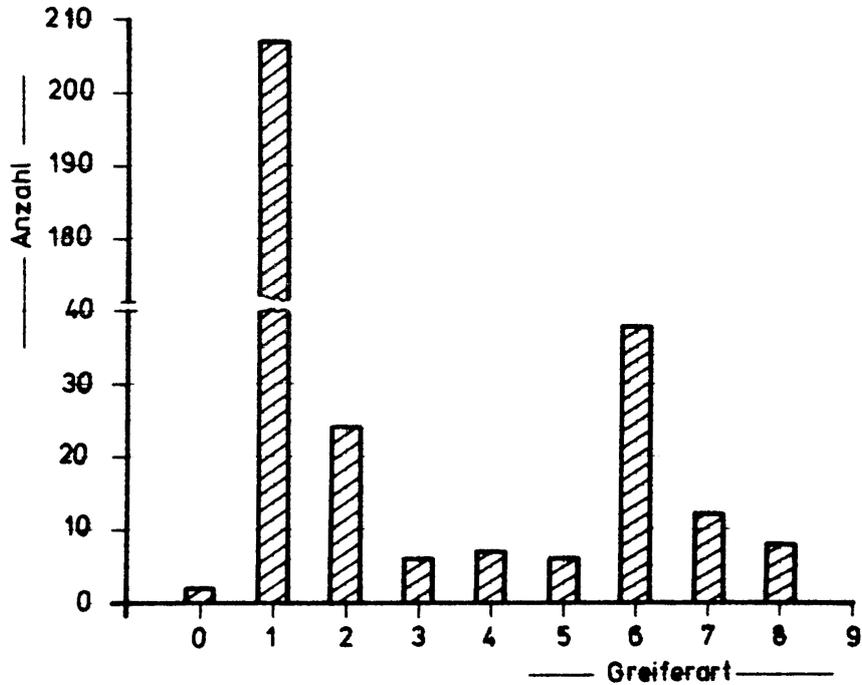


Bild 8-1. Vorkommenshäufigkeit einzelner Greiferarten.

Gesamtzahl: 309; Legende: 0: Werkzeugträger,

- | | |
|-----------------------|---------------------|
| 1 Scherengreifer | 6 Sauggreifer |
| 2 Schraubstoffgreifer | 7 Mehrfingergreifer |
| 3 Einfingergreifer | 8 Adhäsionsgreifer |
| 4 Mehrnadelgreifer | 9 Sondergreifer |
| 5 Balchengreifer | |

Abbildung 2.10: Greiferarten und deren Vorkommenshäufigkeit [14, S.210]

2.4 Individuelles Beschickungssystem

Falls die Bewegungsvielfalt und rasche Umrüstbarkeit von Industrierobotern nicht voll ausgenutzt wird, ist es eine gute Alternative, ein individuelles Beschickungssystem zu konstruieren. Dies hat vor allem preisliche Vorteile. Mittels modularer Linear- und Drehantriebe kann ein an die jeweilige Maschinsituation angepaßtes Be- und Entladesystem aufgebaut werden. Die Hauptelemente, auf die auch in diesem Abschnitt näher eingegangen werden, sind dabei die lineare Bewegung des Arms des Handhabungsgeräts sowie ein Spannsystem, mit dem das Werkstück auf der Fräse eingespannt wird. [16, S.91]

2.4.1 Lineare Bewegung (Arm des Handhabungsgeräts)

Um den Werkstückträger von der Handhabungseinheit auf die Fräse zu bewegen, sind mehrere Komponenten notwendig. Zunächst benötigt man einen Arm, der den Greifer von der Einheit auf die Portalfräse bewegt. Im Buch *Grundlagen der Robotertechnik* [22] werden dazu Konzepte vorgestellt. In diesem Abschnitt werden die für dieses Projekt relevanten Konzepte zusammengefasst.

Da Pneumatik- oder Hydrauliklösungen technisch aufwändiger zu realisieren sind, werden bei dem in der Diplomarbeit entwickelten Handhabungsgerät alle Antriebe elektrisch betrieben.

Elektromotoren erzeugen eine Rotationsbewegung. Um diese Rotationsbewegung in eine lineare Bewegung umzuwandeln, gibt es mehrere Lösungen, welche im Folgenden näher erläutert werden.

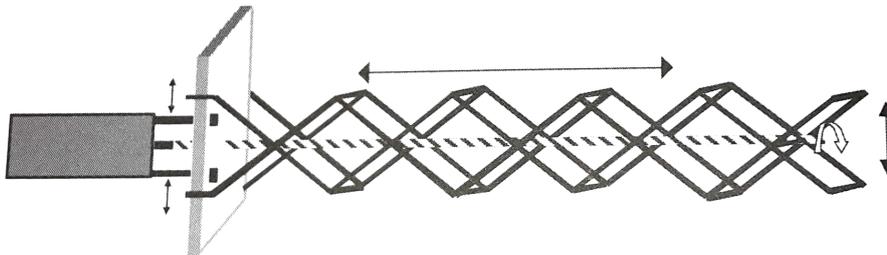


Abbildung 2.11: Linearbewegung mittels Scherengitter [22, S.98]

Schraubenspindel: Eine Gewindestange rotiert und bewegt dadurch eine Platte mit Innengewinde. Zusätzlich wird die Platte mit Führungsstäben gestützt. Diese Variante hat zwei entscheidende Nachteile: Erstens können die Führungsstangen nicht zurückgezogen werden und ragen daher immer in die Fräsmaschine. Zweitens ist dieser Antrieb relativ langsam.

Scherengitter: Scherengitter finden oft bei Hubtischen Verwendung. Dessen Hauptvorteil liegt darin, mit kleiner Verstellgeschwindigkeit eine hohe Auslenkung zu erzielen.

Bei einer großen Anzahl an Gelenken ist die Gesamtreibung des Systems allerdings sehr groß. Im Idealfall sollten die Gelenke kugelgelagert sein. Haben die Lager zu viel Toleranz, so könnte die Konstruktion äußerst wackelig werden. Eine Prinzipskizze des Scherengitters befindet sich in Abbildung 2.11. [22, S.98]

Zahnstangengetriebe: Hier treibt ein drehendes Zahnrad eine Zahnstange an.

Seilrollengetriebe: Eine Seilrolle treibt ein Seil an, welches eine lineare Masse bewegt. Alternativ könnte man statt einem Seil einen Riemen verwenden.

Kurvengetriebe: Mittels Excenter- oder Kurvenscheiben wird eine Drehbewegung in eine lineare Bewegung umgewandelt. Eine weitere Möglichkeit wäre die Verwendung eines Pleuelstangenantriebs.

Positioniergetriebe: Zwei Servo- oder Schrittmotoren werden hintereinander angeordnet. Ein Selbstbau kommt hierbei kaum in Frage, da hohe Anforderungen an die Lager, die Motoren und die Ansteuerung gestellt werden.

Hersteller und Geräte

Elektrische Kompaktschlitten der Firma SMC: SMC ist ein japanisches Unternehmen und hat ein breites Produktangebot im Bereich Automatisierung. Ein Produktionsstandort des internationalen Unternehmens ist auch in Korneuburg.

In Abbildung 2.12 befindet sich ein Anwendungsbeispiel für einen Kompaktschlitten. Es wäre möglich, auf diese Art das Gut auf die Fräse zu heben.

Am 23.12.2016 wurde folgendes Angebot übermittelt: LES16RJ-50-AN1 (Elektr. Schlittenantrieb, Hub 50mm, ohne Feststelleinheit, mit Schrittmotor, inkl. Pulstreiber, NPN, 1,5m I/O-Kabel) für 464,89 € exkl. USt. (geförderter Preis, nur für Lehrzwecke einzusetzen).

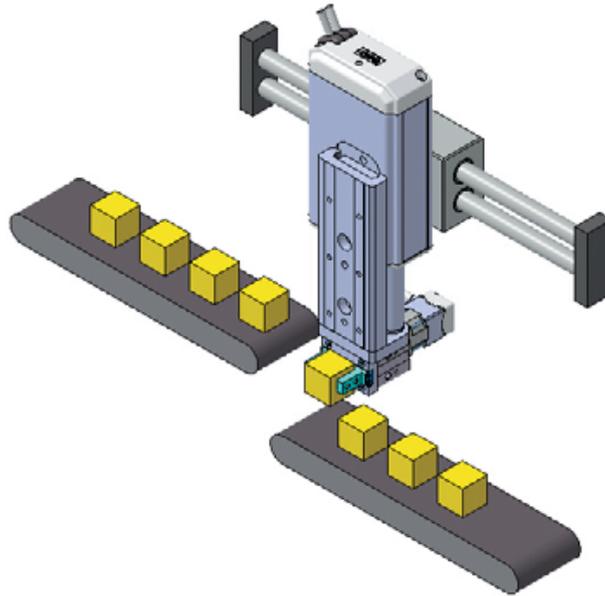


Abbildung 2.12: Anwendungsbeispiel elektrischer Kompaktschlitten der Firma SMC [27]

Der Elektrohändler **Conrad** besitzt ebenfalls ein großes Sortiment an Linearantrieben.

- Elektrozyylinder 230 V/AC Hublänge 610 mm 3500 N Drive-System Europe, 719,99 € inkl. USt. (Abbildung 2.13)
- Elektrozyylinder 24 V/DC Hublänge 500 mm 200 N Drive-System Europe; 191,99 € inkl. USt. (Abbildung 2.14)



Abbildung 2.13: Elektrozyylinder 3,5kN; 719,99 € (Quelle: Conrad)



Abbildung 2.14: Elektrozyylinder 0,2kN; 119,99 € (Quelle: Conrad)

2.4.2 Spannsysteme

Theorie

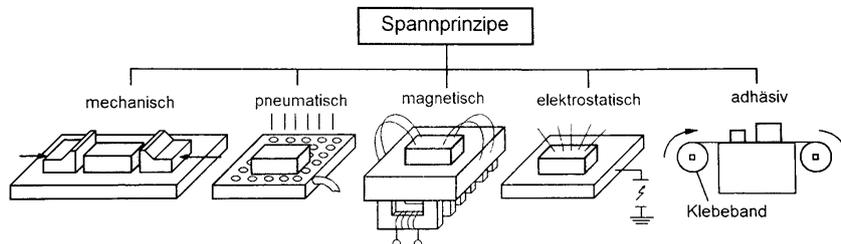


Abbildung 2.15: Spannprinzipie [20, S.327]

Im Buch *Handhabungssysteme* [14, S.327] werden Spanneinrichtungen nach ihrem Prinzip unterteilt. Man unterscheidet mechanische, pneumatische, magnetische, elektrostatische und adhäsive Prinzipien. Die Funktionsweisen sind grob in Abbildung 2.15 dargestellt.

Nach VDI-Richtlinie 2860² wird Spannen folgendermaßen definiert:

Vorübergehendes Sichern eines Körpers in einer definierten Orientierung und Position unter Beteiligung von Kraftschluss. Setzt man allein auf Formschluss, dann spricht man vom „Halten“ (VDI-Richtlinie 2860).

Hersteller und Geräte

Hier werden einige Hersteller und Geräte von Spannsystemen vorgestellt.

²Die VDI-Richtlinie 2860 wurde im Juni 2016 zurückgezogen (www.vdi.de)

RKE 125 der Firma Röhm: Abbildung 2.16, Betätigung: Händisch, Spannweite: 0-208 mm, Spannkraft: 25 kN, Backenbreite: 92 mm, Zubehör: Krallenbackensatz, Preis: 2.320 € exkl. USt

Firma Gressel: Hersteller von hydraulisch betätigten Einfachspannern

Firma Zentra: Elektrischer Schraubstock, Spannkraft: 4000kg; Gewicht: 70kg; 3.571,80 € exkl. USt; elektrisch betrieben; Abbildung 2.17³

Roemheld : Hersteller hydraulischer Schraubstöcke⁴



Abbildung 2.16: RKE 125 der Firma Röhm [17, S.123]



Abbildung 2.17: elektrischer Schraubstock der Firma Zentra (Quelle: <http://www.shop.santool.de/>)

³Quelle: <http://www.shop.santool.de/>

⁴Quelle: <https://www.youtube.com/watch?v=kH0uzp4kvdM>

2.4.3 Elektromotoren

Elektromotoren sind Energiewandler, die elektrische in mechanische Energie umwandeln. [25, S.793]

Gleichstrommaschinen

Gleichstrommaschinen werden vor allem für regelbare Antriebe eingesetzt. Sie haben dabei besonders günstige Betriebseigenschaften. Ein Nachteil gegenüber Drehstromantrieben ist, dass sie für die Rotorwicklung einen Stromwandler benötigen. Dieser besteht meist aus Kollektor und Bürsten, die sich mit der Zeit abnutzen. [25, S.820f]

Schrittmotor

Mit einem Schrittmotor kann ein elektrischer Impuls in eine Drehbewegung mit definiertem Schrittwinkel umgeformt werden. Durch gezielte Erregung der Spulen wird der Motor bewegt. Der Schrittwinkel hängt dabei von der Polpaarzahl und der Phasenzahl des Motors ab. [25, S.819]

Servomotor

Servomotoren sind Motoren, mit denen sich Winkel, Wege und Lagen hochdynamisch und beschleunigungsoptimal einstellen lassen. Um dies zu ermöglichen, ist im Servomotor eine Regelungstechnik eingebaut. Ein Lagegeber übergibt der Steuerelektronik die Lage des Rotors. Je nach Taktung des Gleichstroms bewegt sich der Servomotor in die gewünschte Winkelposition. [25, S.819f]

2.5 Steuerung und Regelung

In diesem Abschnitt werden Konzepte, Systeme und Geräte vorgestellt, die der Steuerung und Regelung einer Anlage oder Maschine dienen können.

2.5.1 Systeme der Werkstückhandhabung

Systeme mit variabler Hauptfunktion lassen sich folgendermaßen unterteilen: [23, S.9]

- Manuell gesteuerte Systeme (z.B. Teleoperator)
- Programmgesteuerte Systeme
- Festprogrammierbare Systeme (z.B. Einlegegerät)
- Freiprogrammierbare Systeme (z.B. Industrieroboter)

2.5.2 CAM

Computer Aided Manufacturing (CAM) übernimmt die gesamte technische Steuerung und Überwachung der Betriebsmittel, Transport- und Lagersysteme. [16, S.13]

Operative Funktionen: Maschinensteuerung, Auftragsverteilung, Fortschrittsüberwachung, Zellenüberwachung, Diagnose

Logistische Funktionen: Planung und Reservierung für Maschinen, Werkzeuge und Material

Informationsmittel: Betriebsdatenerfassung (BDE), die alle erforderlichen organisatorischen Ist-Daten aus dem Betrieb sammelt, aufbereitet und für allfällige Änderungen und besondere Stör Diagnosen bereitstellt

2.5.3 SPS

Eine SPS wird zur Steuerung oder Regelung einer Maschine eingesetzt und digital programmiert. Die SPS-Programmierung (Speicherprogrammierbare Steuerung) ist in der DIN EN 61131 genormt. Das Buch *Automatisieren mit SPS* [30] baut auf dieser Norm auf. Aus diesem Buch wurden auch die Informationen dieses Abschnitts entnommen. Die Verwendung einer SPS ist im Vergleich zu den weiteren vorgestellten Geräten sehr kostenintensiv und eignet sich vor allem im industriellen Bereich.

Wichtige Begriffe

Hier werden die wichtigsten Begriffe im Zusammenhang mit SPS erläutert. [30]

Automatisierung: Automatisierungsgeräte befolgen selbsttätig Programme. Basis sind vorgegebene Führungsgrößen, rückgeführte Prozessgrößen und Daten aus internen Speichern.

Steuerung: Eingangsgrößen beeinflussen nach vorgegebenen Gesetzmäßigkeiten die Ausgangsgrößen.

Regelung: Eine Regelgröße wird laufend erfasst und derart beeinflusst, dass sie mit der Führungsgröße möglichst genau übereinstimmt. Dieser Wirkungsablauf findet in einem geschlossenen Regelkreis statt.

Automatisierungsgeräte: Als Automatisierungsgeräte kommen SPS-basierte Automatisierungssysteme, PCs oder Mischformen in Frage. SPS hat die Eigenschaft der Prozessnähe, während ein PC übergeordnet und datenverarbeitend genutzt wird. Man kann in Industrie-PCs SPS-Karten und zusätzliche Feldbus-Schnittstellen einbauen. Für die Programmerstellung macht es keinen Unterschied, ob die Hardware-Plattform ein PC oder eine SPS ist.

Programmiersprachen

Es gibt fünf Programmiersprachen für SPS: Zwei für die grafische Programmierung (Kontaktplansprache KOP und Funktionsbausteinsprache FBS), zwei für die textuelle Programmierung (Anweisungsliste AWL und Strukturierter Text ST) und eine Programmiersprache mit grafischen und textuellen Elementen (Ablaufsprache AS).

Funktionsweise Hardware-SPS

Die derzeit am verbreitetste Hardware-Plattform der Steuerungstechnik ist SPS. Eine SPS ist modular aufgebaut und besteht zumindest aus folgenden Komponenten:

- Stromversorgung (PS)
- Zentraleinheit (CPU)
- Signalbaugruppen (SM)
- internes Bussystem

Stand der Technik ist der Einsatz von Feldbussystemen, mit dieser die Prozessperipherie an das SPS-Steuerungssystem einfach angekoppelt werden kann.

Das Programm wird auf einem PC programmiert und auf einer SIMATIC Micro Memory Card gespeichert. Diese wird in die CPU der SPS eingeführt (siehe Abbildung 2.18).

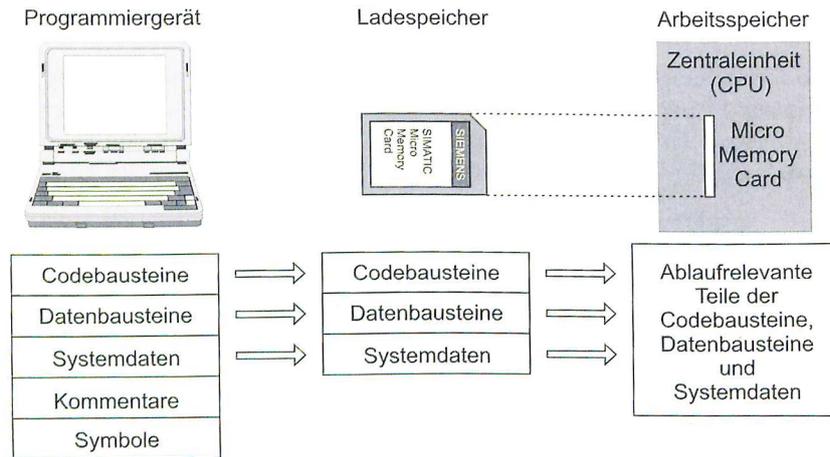


Abbildung 2.18: Daten im Ladespeicher der SPS [30, S.12]

Bussysteme in der Automatisierungstechnik

Beispiele für Bussysteme in der Automatisierungstechnik sind AS-i, Profibus, Profinet und Industrial Ethernet. Dies sind standardisierte, offene Bussysteme nach der Norm IEC 61158. Eine derzeitige Entwicklung ist, das Ethernet-TCP/IP-Bussystem der Bürowelt auch im Produktionsbereich zu verwenden. Dies macht das Bussystem Profinet, welches in das Feldebussystem Profibus integriert ist. [30]

2.5.4 Raspberry Pi

Der Raspberry Pi ist einer der bekanntesten und weitverbreitetsten Einplatinen-Kleinstrechner. Alle wichtigen Komponenten, die ein Computer erfordert, sind dabei auf einer etwa kreditkartengroßen Platine zusammengefasst [21, S.9], [26, S.29]. Der Preis für das aktuelle Modell (Raspberry PI 3 Modell B) liegt bei circa 40 €⁵. Dieser Minicomputer verfügt über USB, WLAN und Bluetooth, wodurch sich zum Beispiel leicht ein Home-Server realisieren lässt [2]. Weiters gibt es die GPIO-Schnittstelle, mit der sich Hardwareprojekte realisieren lassen. Dies sind PINs, die man sowohl als Input als auch als Output verwenden kann und sich durch eine entsprechende Software steuern lassen. Als Programmiersprache wird dabei meistens Python verwendet. [15, S.108ff]

HATs

Ein Hardware attached on top (HAT) ist ein Board, dass man einfach auf bestehende Hardware aufstecken kann. Für den Raspberry PI gibt es unzählige dieser Add-On-Boards. Ein Beispiel für ein HAT ist der Adafruit Schrittmotor HAT. An diesem können bis zu vier Gleichstrommotoren oder bis zu 2 Schrittmotoren angehängt werden. In Abbildung

⁵<https://www.conrad.at/de/raspberry-pi-3-model-b-1-gb-ohne-betriebssystem-1419716.html> (aufgerufen am 5.3.2018)

2.19 sieht man einen Raspberry Pi mit aufgesetztem Motor-HAT und zwei angeschlossene Schrittmotoren. [5]

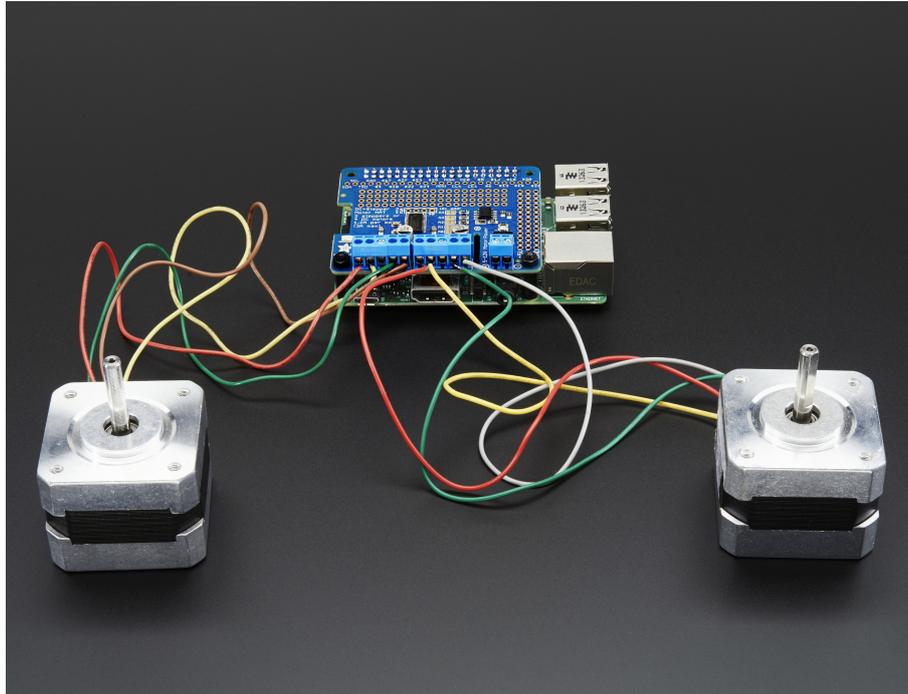


Abbildung 2.19: Raspberry mit aufgesetzten HAT [1]

2.5.5 Arduino

Arduino ist der bekannteste Vertreter unter den Mikrocontrollern. Ein Mikrocontroller besteht üblicherweise aus den folgenden Komponenten:

- zentrale Recheneinheit (CPU)
- Speicher (ROM und RAM)
- Ein- und Ausgabeports
- Taktgeber

Anwendungsgebiete von Mikrocontrollern sind unter anderem Robotik, Haushaltsgeräte mit Programmsteuerung, Herzschrittmacher usw. [13, S.67f] Der Preis für ein Arduino-Board liegt bei 28 €⁶.

2.5.6 Python

Python ist eine relativ moderne, objektorientierte Programmiersprache. Entwickelt wurde sie 1990 in C. Python wird sehr vielseitig eingesetzt, wie z.B. als Skriptsprache zur Sys-

⁶<https://www.conrad.at/de/arduino-board-leonardo-65163-192458.html> (aufgerufen am 21.2.2018)

temverwaltung oder zur Entwicklung von Web-Applikationen. In Python können leicht weitere Module importiert werden, welche den Funktionsumfang erweitern. [19, S. 375]

2.5.7 Telegram Messenger

Telegram ist ein Instant Messaging Dienst, der es erlaubt, Chatnachrichten inklusive Fotos, Videos und sonstiger Dateien zu verschicken und zu empfangen. Telegram kann auf allen gängigen internetfähigen Endgeräten wie Smartphones, Tablets oder Computer installiert werden. Auch eine installationsfreie Webversion wird von den Entwicklern zur Verfügung gestellt. Die Anwendersoftware ist dabei größtenteils quelloffen und kann mit entsprechenden Fachkenntnissen auf weitere Plattformen oder Geräte adaptiert oder individuell angepasst werden. [7].

Telegram ist dabei eine der beliebtesten Alternativen zu den am weitest verbreiteten Instant-Messengern WhatsApp und Facebook. Weltweit gibt es 100 Mio. aktive Telegramnutzer, damit liegt es auf Platz 9. [8]

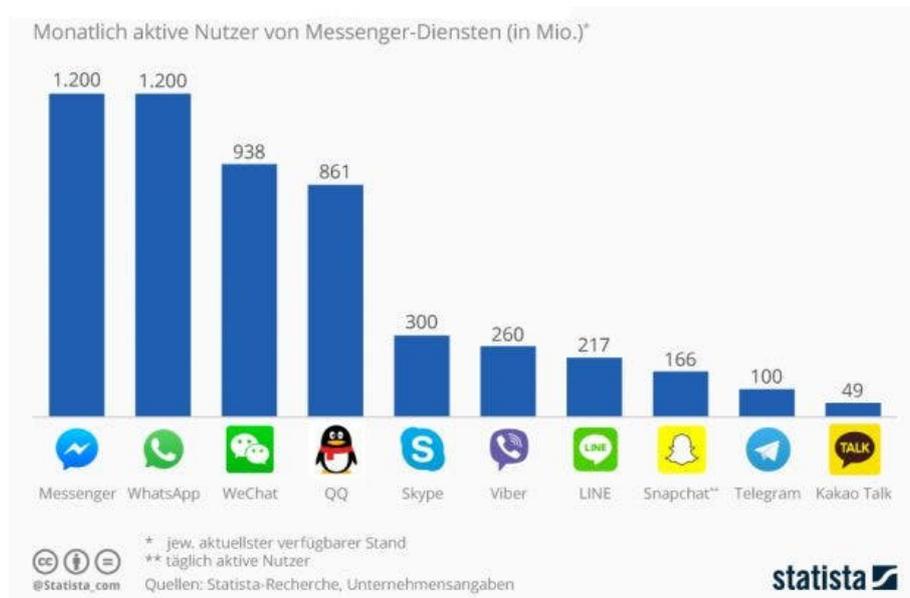


Abbildung 2.20: Messenger-Nutzung weltweit [8]

Telegram Bot Plattform

Seit Juni 2015 bietet Telegram an, eigene Chatbots zu entwickeln. Ein Bot ist ein Telegramaccount, der anstatt durch einen Menschen durch eine Software verwaltet wird. Ein User kann dem Bot Nachrichten oder Befehle schicken. Befehle beginnen mit /, z.B. /start. Der Bot kann dem User Nachrichten, Fotos oder andere Dateien zurücksenden oder weitere softwaregesteuerte Befehle ausführen. [7]

Kapitel 3

Verworfenne Konzepte

Dieses Kapitel zeigt entworfene Konzepte, die nicht realisiert wurden.

3.1 Konzept Scherengitter

Die technischen Grundlagen für Scherengitter sind im Abschnitt 2.4.1 zu finden. Das Ziel dieses Konzepts war es, ein individuelles Beschickungssystem zu konzipieren. Das Handhabungsgerät sollte dabei eine eigene Einheit darstellen. Man könnte diese Einheit in einem Folgeprojekt zum Beispiel auch zu einer selbstfahrenden Einheit umgestalten. Das Werkstück wird in einen Werkstückträger eingespannt. Dieser Werkstückträger liegt mit dem eingespannten Werkzeug auf der Handhabungseinheit. Die Beladung des Werkstückträgers sowie die Beladung der Handhabungseinheit sind nicht Teil dieser Diplomarbeit. Das Handhabungssystem nimmt den Werkstückträger auf und legt diesen in ein Einspannsystem. Das Einspannsystem ist dabei fest mit der Portalfräse verbunden. Nach dem Fräsvorgang platziert das Handhabungsgerät den Werkstückträger mit dem bearbeiteten Werkstück wieder auf der Handhabungseinheit.

Das Handhabungsgerät muss 600 mm ausfahren können, um den Werkstückträger auf der Fräsmaschine abzulegen. Hierzu wurde eine Vorauslegung für ein Scherengitter mit drei Kreuzen vorgenommen.

3.1.1 Geometrieberechnung

Da der zurückgelegte Weg insgesamt 600 mm betragen soll und es drei Scheren gibt, muss die Differenz zwischen offener und geschlossener Schere 200 mm betragen. Es wird ein Schließwinkel α und ein Öffnungswinkel β definiert, die angeben, wie weit die Schere maximal schließen bzw. öffnen kann. Die Variable $2a$ bezeichnet die Länge eines Stabes.

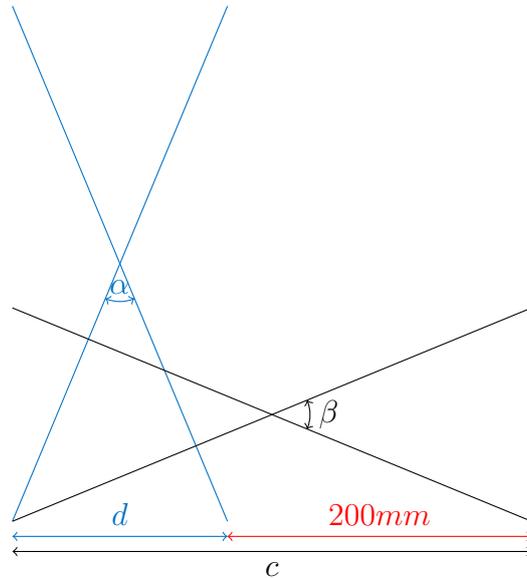


Abbildung 3.1: Berechnungsskizze für die Varianten Schere offen und Schere zu

Durch den Sinussatz können folgende geometrische Beziehungen hergestellt werden:

$$a = c \cdot \underbrace{\frac{\sin\left(\frac{\beta}{2}\right)}{\sin(180^\circ - \beta)}}_{=e} = c \cdot e \quad (3.1)$$

$$a = d \cdot \underbrace{\frac{\sin\left(\frac{180^\circ - \alpha}{2}\right)}{\sin(\alpha)}}_{=f} = d \cdot f \quad (3.2)$$

Formt man diese beiden Gleichungen nach a um, so erhält man:

$$a = 200 \cdot \frac{f \cdot e}{f - e} \quad (3.3)$$

In der Tabelle 3.1 wurde die Armlänge $2a$ sowie die geschlossene Scherenbreite d für bestimmte Öffnungs- und Schließwinkel berechnet.

Öffnungswinkel β	Schließwinkel α	geschlossene Scherenbreite d	Armlänge $2a$
35°	45°	095 mm	316 mm
45°	45°	141 mm	370 mm
45°	35°	137 mm	358 mm
35°	35°	092 mm	306 mm

Tabelle 3.1: Armlängen und geschlossene Scherenbreite bei verschiedenen Öffnungs- und Schließwinkel (auf ganze mm gerundet)

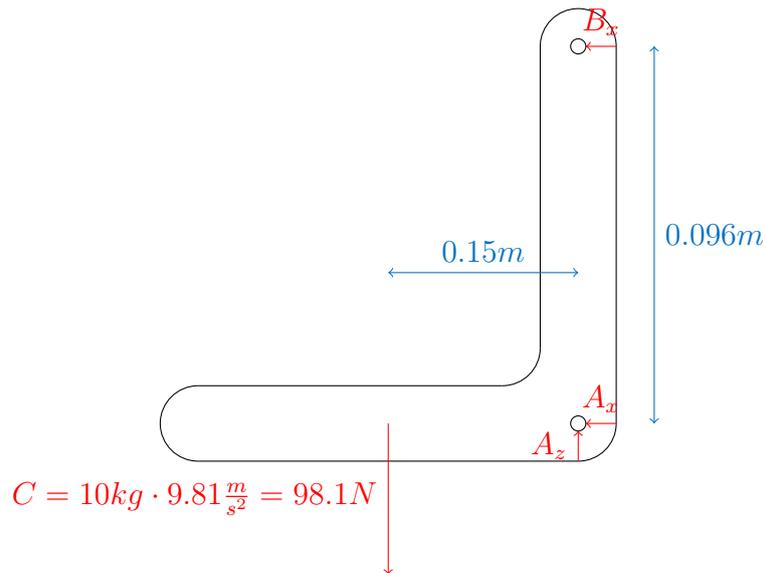


Abbildung 3.2: Kräfteskitze für die offene Schere (Lagerbelastung)

3.1.2 Kräfte in der Gabel

Der Beladekran wird für eine Belastung von 10kg ausgelegt. Die Streckgrenze für handelsübliches Aluminium beträgt $34\text{N}/\text{mm}^2$ [11]. Hierfür wurde davon ausgegangen, dass die halbe Armlänge $a = 160\text{mm}$ beträgt und der Schließwinkel $\alpha = 35^\circ$ groß ist.

Gabelkräfte

Aus der Figur 3.2 ergeben sich folgende Gleichungen:

$$A_z = C = 98.1\text{N} \quad (3.4)$$

$$A_x + B_x = 0 \quad (3.5)$$

$$C \cdot 0.15 + B_x \cdot 0.096 = 0 \quad (3.6)$$

Daraus folgt:

$$B_x = \frac{C \cdot 0.15}{0.096} = 153.28\text{N} \quad (3.7)$$

$$A_x = -B_x = -153.28\text{N} \quad (3.8)$$

Diese Kraft teilt sich auf 2 parallele Gabeln auf. Die Lagerbelastung beträgt daher:

$$\text{Lagerbelastung}_{\max} = \frac{\sqrt{153.28^2 + 98.1^2}}{2} = 91\text{N} \quad (3.9)$$

Die Streckgrenze für handelsübliches Aluminium beträgt 34N/mm^2 [11, S.363]. Der erforderliche Querschnitt in der Gabelaufhängung beträgt daher mindestens:

$$A_{\min} = \frac{91\text{N}}{34\frac{\text{N}}{\text{mm}^2}} = 2.68\text{mm}^2 \quad (3.10)$$

Kräfte im Kranportal

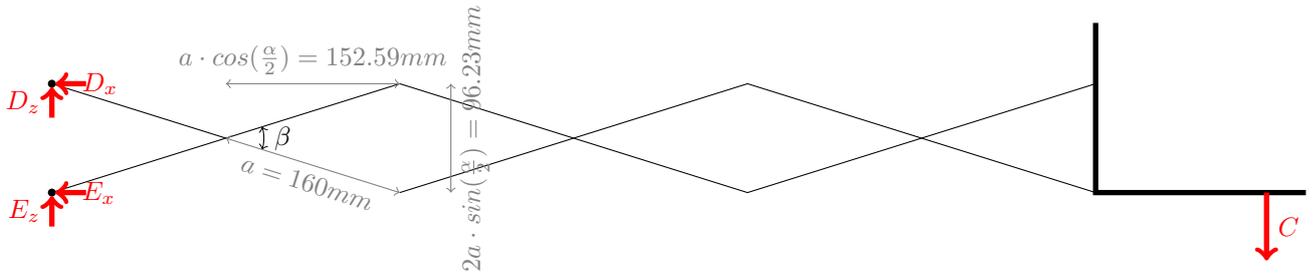


Abbildung 3.3: Kräfteskizze für die Lagerkräfte im Kranportal



Abbildung 3.4: Vereinfachte Kräfteskizze für die Lagerkräfte im Kranportal

$$\sum M_{iE,y} : -C \cdot 1065.57\text{mm} + D_x \cdot 96.23\text{mm} = 0 \quad (3.11)$$

$$\Rightarrow D_x = \frac{C \cdot 1065.57\text{mm}}{96.23\text{mm}} = 1086.26\text{N} \quad (3.12)$$

$$E_x = -D_x = -1086.28\text{N} \quad (3.13)$$

$$D_z = E_z = \frac{C}{2} = 4905\text{N} \quad (3.14)$$

$$\text{Lagerbelastung}_{Kran} = \frac{\sqrt{49.05^2 + 1086.28^2}}{2} = 543.69\text{N} \quad (3.15)$$

Die Streckgrenze für handelsübliches Aluminium beträgt (wie bereits im vorhergehenden Abschnitt erwähnt) 34N/mm^2 [11, S.363]. Der erforderliche Querschnitt in den Lagern des Portalkrans beträgt daher mindestens:

$$A_{\min} = \frac{543.69\text{N}}{34\frac{\text{N}}{\text{mm}^2}} = 15.99\text{mm}^2 \quad (3.16)$$

Bei einer Leistendicke von $s = 5\text{mm}$ muss daher die Wandstärke mindestens $\frac{15.99\text{mm}^2}{5\text{mm}} = 3.2\text{mm}$ betragen.

3.1.3 CAD-Modell

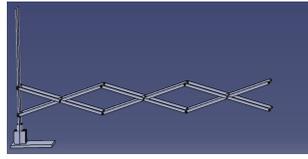
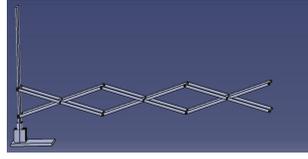
Zustand	untere HP	obere HP	Grafik
1. Startposition	90mm	405mm	
2. Anheben	100mm	415mm	
3. Endposition	100mm	200mm	
4. Absenken	90mm	190mm	

Tabelle 3.2: Zustände des Scherengitterroboters

3.1.4 Geschwindigkeit

Im Ausfahrbetrieb muss die obere Hebeplatte 200 mm an Höhe zurücklegen. Experimente an einem 3D-Drucker¹, an dem die Z-Achsenbewegung ebenfalls mittels Gewindestange realisiert wurde, ergaben, dass für das Zurücklegen diese Strecke eine Minute benötigt wird. Dies ergibt eine Geschwindigkeit von $v_z = \frac{0.2\text{m}}{60\text{s}} = 0.00333\frac{\text{m}}{\text{s}} = 0.012\frac{\text{km}}{\text{h}}$.

Für den Belade- oder Entladeprozess der Fräsmaschine muss daher mit ungefähr 2 Minuten gerechnet werden.

3.1.5 Ideenbewertung

Ein großer Nachteil von Scherengitterkonstruktionen ist die große Anzahl an Gelenken. Um die Beweglichkeit zu gewährleisten und trotzdem die Konstruktion nicht zu wacklig

¹3D-Drucker: BQ Prusa i3 Hephestos

werden zu lassen, müssen die Gelenke sehr genau ausgearbeitet sein. Die Konstruktion sieht sehr einfach aus, jedoch wäre eine praktische und brauchbare Ausführung äußerst aufwendig und nur schwer zu realisieren. [22, S.98]

3.2 Konzept Förderband

Die Theorie über Förderbänder wurde bereits im Abschnitt 2.2.1 beschrieben. Dieses Konzept kann man auch für die Beladung und Entladung der Fräsmaschine benutzen.

Mittels Förderband wird hier der Werkstückträger in das Innere der Fräsmaschine befördert. Anschließend wird die Fördereinrichtung um 140mm abgesenkt und der Werkstückträger auf der Einspannvorrichtung abgesetzt. Im kostengünstigsten Fall werden folgende Zukaufteile verwendet (Preise exkl. USt.):

Motoren	Schrittmotor; reprop-world.com	8 Stück	15.00 €/Stk	100.00 €
Zahnriemen	Zahnriemen SIT HTD (Profil 5M, b=25mm, l=835mm, 167 Zähne); Conrad	4 Stück	14.16 €/Stk	56.64 €
Zahnscheiben	Zahnscheibe HTD0155M025 (Stahl, 15 Zähne, Passend für Riemenbreite: 25 mm); Conrad	8 Stk	4.16 €/Stk	33.28 €

Tabelle 3.3: Zukaufteile, Preise exkl. USt.

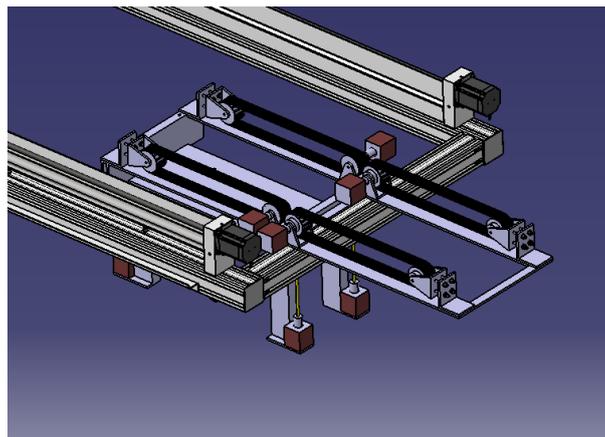


Abbildung 3.5: Förderbandkonzept: Mittels Bandförderer wird der Werkstückträger in das Innere der Fräsmaschine befördert

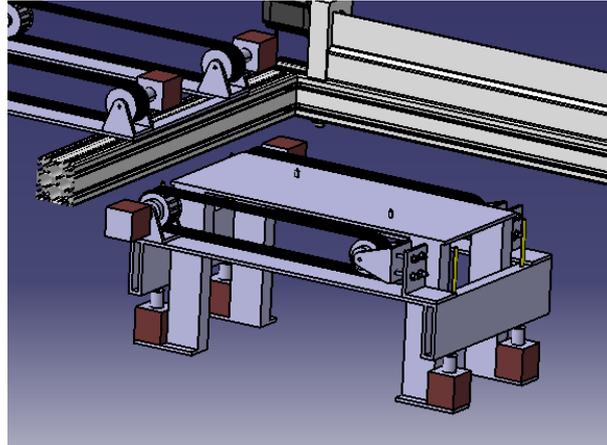


Abbildung 3.6: Förderbandkonzept: Das Förderband wird abgesenkt und der Werkstückträger auf die Spannvorrichtung aufgesetzt.

3.3 Konzept Regalbediengerät (Version 1 und 2)

Vorbild für dieses Handhabungsgerät war ein Regalbediengerät der Firma Schäfer. Dieses wurde in folgendem Youtube-Video präsentiert: https://youtu.be/2f_Q6T0fkRE?t=16s

3.3.1 CAD-Modell

Das Konzept 1 wurde mit 3 ineinander gelegenen und gegenseitig bewegbaren U-Profilen realisiert. Die äußere Schiene ist dabei fix auf dem Fahrschlitten montiert. Der Fahrschlitten, welcher für die horizontale Bewegung zuständig ist, ist in Abbildung 3.7 noch nicht abgebildet.

Die mittlere Schiene wird mittels Zahnstange-Zahnrad-Verbindung bewegt. Sie kann in beide Richtungen ausfahren. Die Innenschiene wird mittels Drahtseil bewegt. Das Drahtseil ist symmetrisch zwischen der Außenschiene und der Innenschiene eingespannt und wird von Rollen, die an der Mittelschiene montiert sind, bewegt. Dadurch wird die Innenschiene genau zweimal so schnell bewegt und doppelt so weit ausgefahren wie die mittlere Schiene. Diese Funktionsweise entspricht etwa dem Zweifachteleskophubgerüst mit Freisicht bei Gabelstapler aus Abschnitt 2.2.2. Die Prinzipskizze (Abbildung 2.8) des Hubgerüsts bei Gabelstapler befindet sich auf Seite 14.

3.3.2 Variante 1

In der 1. Variante konnte das Werkstück hochgehoben werden. Wenn die Schienen ausgefahren sind, wird der Werkstückträger mittels der Backen gehoben. Die Backen sind 360° drehbar. Nach dem Anheben wird der Werkstückträger in das Handhabungsgerät eingefahren. Der Förderschlitten bewegt sich anschließend zum Einlegegerät der Fräsmaschine. Dort wird der Werkstückträger auf das Einlegegerät abgelegt. Die Backen werden da-

bei um etwa 290° bewegt, so dass sie den Werkstückträger am Einlegegerät ablegen und anschließend in den Schienen verschwinden. Eine Skizze der 1. Variante befindet sich in Abbildung 3.7.

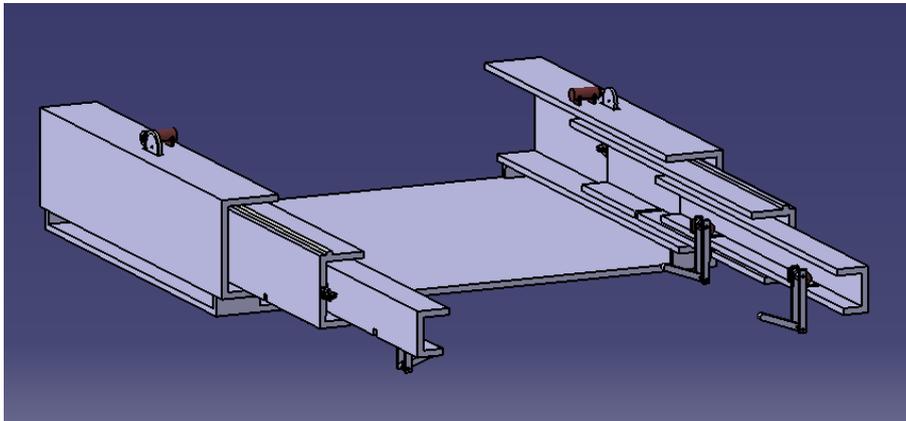


Abbildung 3.7: Regalbediengerät, Variante 1: Der Werkstückträger wird hochgehoben und in die gewünschte Position gebracht.

3.3.3 Variante 2

In der 2. Variante wird der Werkstückträger nicht mehr hochgefahren, sondern am Boden entlang geschoben. Der Vorteil dabei ist, dass geringere vertikale Kräfte auf das Handhabungsgerät wirken. Dafür sind die horizontalen Kräfte größer, da beim Ein- und Ausfahren des Handhabungsgeräts auch Reibkräfte zwischen Werkstückträger und Oberfläche, auf dem der Werkstückträger bewegt wird, entstehen. Da der Mehrfachauszug technisch aufwendig zu konstruieren ist, wurde in Variante 3 anstatt der mehrfach verschachtelten Schienen eine in beide Richtungen ausfahrende Teleskopschiene mit Überauszug zwischen Innenschiene und Außenschiene montiert. Diese Variante 3 wurde realisiert und wird im nächsten Kapitel ausführlich beschrieben.

Kapitel 4

Realisiertes Konzept

In diesem Kapitel werden die Details zu der tatsächlich realisierten Version des Handhabungsgeräts behandelt. Dieses wird in Abbildung 4.1 dargestellt.

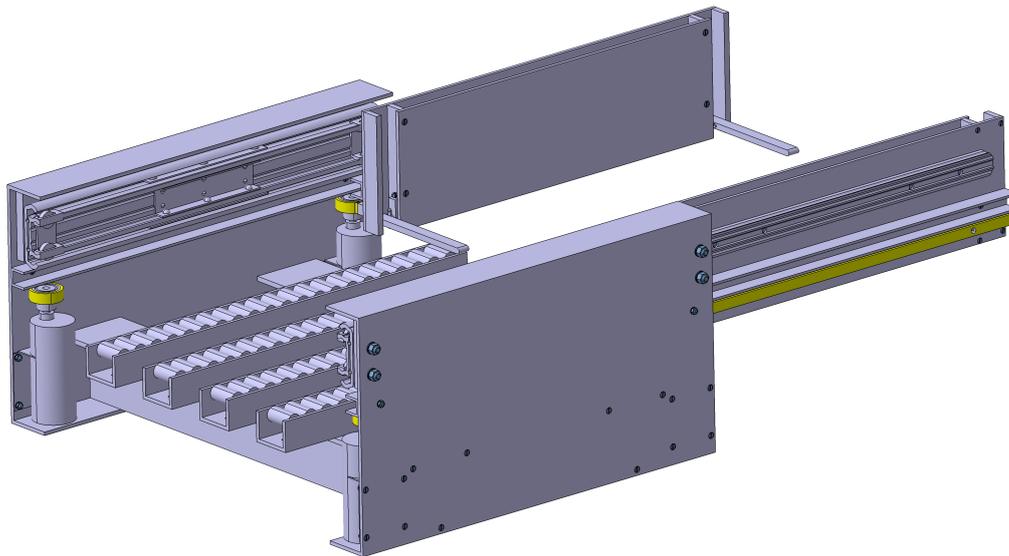


Abbildung 4.1: Endgültige Version des Handhabungsgeräts

4.1 Konzept der Fertigungszelle

Das Konzept der Fertigungszelle sieht vor, dass durch das Handhabungsgerät Werkstückträger in die Fräsmaschine geschoben werden. Dort muss eine separate Spannvorrichtung den Werkzeugträger fixieren. Da eine große Werkstückdiversität prognostiziert wird, werden Werkstückträger eingesetzt, auf denen die Werkstücke vorher eingespannt werden.

Für den Weg des Werkstückträgers bis zum Handhabungsgerät wurden mehrere Szenarien entwickelt:

1. Ein Regalbediengerät kann den Werkstückträger bis zum Handhabungsgerät liefern. Hierfür ist notwendig, dass sich ein Regal mit den zu bearbeiteten Stücken in der Nähe der Fräse befindet.
2. Ein Fahrerloses Transportfahrzeug (FTF) bringt den Werkstückträger mit dem Werkstück bis zum Handhabungsgerät, das Handhabungsgerät schiebt den Werkstückträger direkt vom FTF in die Fräse.
3. Das Handhabungsgerät befindet sich auf einem FTF. Da das Handhabungsgerät mit einer Spannung von 12V DC versorgt und über WLAN gesteuert wird, ist dies ebenfalls eine realistische und leicht umsetzbare Möglichkeit. Fahrerlose Transportsysteme werden meist von einem Akku gespeist. Dieser Akku könnte auch das Handhabungsgerät mit Strom versorgen. Die meisten FTFs haben eine Betriebsspannung von 24 - 96V, wobei für kleinere FTFs der niedrigere Spannungsbereich üblicher ist. [28, S.158] Um den Strom des Akkus zu nutzen, müsste die Spannung mittels eines DC-DC-Wandlers heruntertransformiert werden.

4.2 Aufbau und Funktionsweise

Der Innenschlitten wird ausgefahren. Anschließend werden die Schranken abgesenkt. Diese Schranken schieben den Werkstückträger auf das Handhabungsgerät. Um die Reibung zwischen Untergrund und Werkstückträger zu reduzieren, wurden im Handhabungsgerät Röllchenleisten angebracht. Diese können als Fertigteil zugekauft werden. Die Innenschiene mit den Schranken wird mithilfe einer Zahnstange ein- oder ausgefahren. Diese Zahnstangen werden mittels Zahnräder angetrieben, die sich auf Getriebemotoren befinden. Eine Teleskopschiene, die sich in beide Richtungen ausfahren lässt, sorgt für die statische Verbindung zwischen Innen- und Außenschlitten.

An der beweglichen Innenschiene sind auf beiden Seiten vorne und hinten Schranken angebracht, die zu- und aufgemacht werden können. Diese werden mittels Servomotor betrieben.

4.2.1 Zugekaufte Teile

In Tabelle 4.1 sind alle zugekauften Teile aufgelistet. Die Gesamtmaterialekosten für dieses Handhabungsgerät belaufen sich auf 732,23 €.

Shop	Bauteil	Stückpreis	Menge	Gesamt
Prokilo	Aluprofile (genaue Aufschlüsselung in Abb. 4.2)	120,48 €	1	120,48 €
Conrad	Reely Micro-Servo S-0403 Analog-Servo	6,99 €	4	27,96 €
Conrad	Getriebemotor 12 V/DC MFA 919D1481 148:1	24,99 €	4	99,96 €
Conrad	Polyacetal Stirnzahnrad Reely Modul-Typ: 1.0	3,79 €	4	15,16 €
Bauhaus	8x Mutter ISO 4032 M4			
Bauhaus	16x Beilagscheibe ISO 7089 M4			
Bauhaus	8x Mutter ISO 4032 M5			
Bauhaus	8x Beilagscheibe ISO 7089 M5x10			
Bauhaus	8x Schraube ISO 4017 M6x16			
Bauhaus	28x Beilagscheibe ISO 7089 M6x12			2,81 €
Conrad	Außensechskantschrauben M4x12	4,19 €	1 Set	4,19 €
Conrad	Unterlegscheiben 2.5 mm DIN125 Stahl	1,99 €	1 Set	1,99 €
Conrad	Sechskantmuttern M2.5 DIN934 100 St.	1,49 €	1 Set	1,49 €
Conrad	Senkschrauben DIN 963 M2.5x8 mm; 100 Stk.	3,99 €	1 Set	3,99 €
Conrad	Zahnstange	27,98 €	2	55,96 €
Profilscope	Teleskopschiene	87,00 €	2	174,00 €
DS Rollen	Röllchenleiste		4	56,42 €
Conrad	Experimentierplatine	0,88 €	1	0,88 €
Conrad	Kohleschicht-Widerstand 10 kOhm	0,13 €	1	0,13 €
Conrad	Flachbandkabel Bunt	1,79 €	1	1,79 €
Conrad	Niedervolt-Steckverbinder Buchse, Einbau	0,64 €	1	0,64 €
Conrad	Raspberry Pi Verbindungskabel RB-CB3-25	8,99 €	1	8,99 €
Conrad	Mikroschalter	2,59 €	4	10,36 €
Amazon	Raspberry Pi 3 Model B	34,38 €	1	34,38 €
Amazon	LAOMAO DC-DC-Wandler	9,99 €	1	9,99 €
Amazon	LEICKE Netzteil 60W 12 V	15,12 €	1	15,12 €
Amazon	Adafruit Schrittmotor HAT	29,99 €	1	29,99 €
Ebay	USB-Kabel	1,00 €	1	1,00 €
Bauhaus	Innensechskantschraube M5x30	31 €/kg	0,038 kg	1,18 €
Bauhaus	Alu-Welle	4,68 €	1	4,68 €
Amazon	Gewindebohrer M2.5	9,05 €	1	9,05 €
Ebay	Wire Jumper Pin Header	3,67 €	1 Set	3,67 €
Amazon	Jumperkabel	7,99 €	1 Set	7,99 €
Libro	SD-Karte für Raspberry	14,99 €	1	14,99 €
Libro	Etiketten	1,39 €	1	1,39 €
Atzler	Beilagscheiben 5,3x10	1,99 €	2 Sets	3,98 €
Ebay	Gewindebohrer M2.5	8,90 €	1 Set	8,90 €
Gesamtkosten Material				732,33 €

Tabelle 4.1: zugekaufte Bauteile

Anzahl	Bezeichnung	Menge ME	Preis PE	Rabatt	Wert in EUR
1	Aluminium Aluminium-U-Profil 40 x 200 x 40 x3 AlMgSi 0.5** 1 St 730 mm	1,62 kg	17,25 /1		27,95
1	Aluminium-Flach 60X 8 AlMg Si 0.5 1 St 730 mm	0,95 kg	17,25 /1		16,39
1	Aluminium-Flach 15x5 AlMg Si 0.5 1 St 1.000 mm	0,20 kg	17,25 /1		3,45
1	Aluminium-Flach 100X 3 AlMg Si 0.5 1 St 1.400 mm	1,13 kg	17,25 /1		19,49
1	Aluminium-Flach 10X 5 AlMg Si 0.5 1 St 500 mm	0,07 kg	37,50 /1		2,63
1	A T-Profil 60X 4 1 St 850 mm	1,06 kg	17,25 /1		18,29
1	Aluminium-U-Profil 15 x 2 AlMgSi 0.5 1 St 1.500 mm	0,33 kg	17,25 /1		5,69
	Standard - Schneidkosten bis 7,50 kg / Meter	7 St	2,00 /1		14,00
	Paketversand	1 St	12,59 /1		12,59
Liefergewicht:		5,36 kg			

Abbildung 4.2: Liste zugekaufter Aluprofile

4.3 Berechnungen zur Konstruktion und Auslegung

4.3.1 Schrankeneinheit

Hier wurde das erforderliche Mindestmoment im Drehpunkt berechnet, um die notwendige Größe des Schrankenmotors zu berechnen. Hierbei wurden nur statische Kräfte (Gewichtskraft) beachtet, da die Beschleunigung gering ist und daher vernachlässigt werden darf.

Dichte Aluminium $\rho_{Al} = 0.0027 \frac{kg}{cm^3}$ [9, S.160/Tab 6.1]

Länge Schranken $l_{Schr} = 100mm$

Breite Schranken $b_{Schr} = 5mm$

Tiefe Schranken $t_{Schr} = 10mm$

Volumen Schranken $V_{Schr} = l_{Schr} \cdot b_{Schr} \cdot t_{Schr} = 0.000005m^3$

Gewicht Schranken $m_{Schr} = V_{Schr} \cdot \rho_{Al} = 0.01355kg$

Fallbeschleunigung Erde $g = 9.81 \frac{m}{s^2}$ [18, S.63]

Moment im Drehpunkt $M_{Dr} = m_{Schr} \cdot g \cdot \frac{l_{Schr}}{2} = 0.0066Nm = 0.66Ncm$

Der ausgewählte Servomotor¹ hat ein Stellmoment von 8 Ncm und kann den Schranken daher problemlos bewegen.

¹Reely Micro-Servo AS0403 (<https://www.conrad.at/de/reely-micro-servo-as0403-analog-servo-getriebe-material-kunststoff-stecksystem-jr-1365554.html>)

4.3.2 Getriebemotor

In diesem Abschnitt werden die Anforderungen an den Getriebemotor berechnet.

Moment

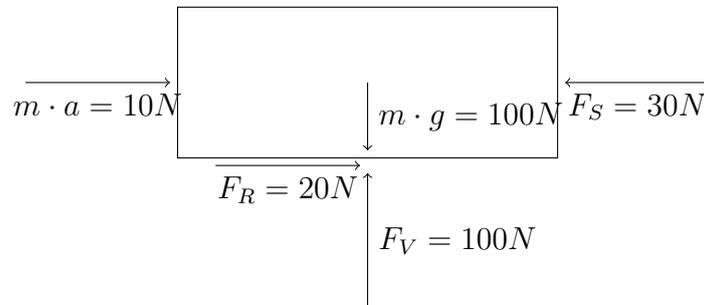


Abbildung 4.3: Skizze der Kräfte am zu bewegenden Werkstückträger

In Abbildung 4.3 befindet sich eine mechanische Skizze der auftretenden Kräfte. Die Masse m wurde mit $m = 10\text{kg}$ angenommen, die Gewichtskraft des Körpers beträgt daher etwa $F_W = 100\text{N}$. Die größte Reibung tritt bei Haften auf. Der Haftreibungskoeffizient für Stahl auf Aluminium ist rund $\mu = 0,2$ [10, S.262]. Die Reibkraft beträgt daher $F_R = F_V \cdot \mu = 20\text{N}$. Der Arm beschleunigt mit etwa $a = 1\frac{\text{m}}{\text{s}^2}$. Dies ergibt eine Beschleunigungskraft von $m\dot{a} = 10\text{N}$. Um Reibkraft und Beschleunigungskraft zu kompensieren, muss die Schubkraft daher mindestens $F_S = 30\text{N}$ groß sein.

Der Radius des Zahnrads beträgt $r = 7,25\text{mm}$. Das erforderliche Drehmoment beträgt daher mindestens $M = F_S \cdot x = 30\text{N} \cdot 0,00725\text{m} = 0,2175\text{Nm} = 21,75\text{Ncm}$. Der ausgewählte Getriebemotor² hat ein Spitzendrehmoment von 60Ncm und hält somit den Belastungen stand.

Drehzahl

Das Handhabungsgerät fährt $x = 300\text{mm}$ aus. Für diese Strecke sollte es in der Voraussetzung $t = 3\text{s}$ benötigen. Daraus folgt für die Geschwindigkeit $v = \frac{x}{t} = \frac{0,3\text{m}}{3\text{s}} = 0,1\frac{\text{m}}{\text{s}}$. Die Winkelgeschwindigkeit des Zahnrades mit dem Wälzkreisradius $r = 7,25\text{mm}$ beträgt somit $\omega = \frac{0,1\frac{\text{m}}{\text{s}}}{0,00725\text{m}} = 1,38\frac{1}{\text{s}}$. Die Drehzahl des Motors sollte somit $n = \frac{30\omega}{\pi} = 131,8\frac{\text{U}}{\text{min}}$ betragen. Der ausgewählte Getriebemotor hat eine Leerlauf-Drehzahl an 12V von $106\frac{\text{U}}{\text{min}}$ und liegt damit im Toleranzbereich. Damit liegt die theoretische Ausfahrzeit bei $t = \frac{30\frac{\text{U}}{\text{min}} \cdot x}{n\pi r} = \frac{30\frac{\text{U}}{\text{min}} \cdot 0,3\text{m}}{160\frac{\text{U}}{\text{min}} \cdot \pi \cdot 0,00725\text{m}} = 3,73\text{s}$. Diese Zeitangabe enthält keine Reaktionszeit des Handhabungsgeräts inklusive Chatbot und auch keine Beschleunigungs- und Bremsphase. Messungen am aufgebautem Modell haben ergeben, dass die Gesamtzeit zum Ausfahren bis zu 10 Sekunden beträgt.

²Getriebemotor 12 V/DC MFA 919D1481 148:1 (<https://www.conrad.at/de/getriebemotor-12-vdc-mfa-919d1481-1481-222366.html>)

4.3.3 Durchbiegung der Außenschiene

Es wurde untersucht, wie stark sich die Außenschiene unter der Last der Innenschiene verformt. Im unteren Bereich der Außenschiene sorgt die Verschraubung mit den T-Profilen für Stabilität. Diese Verschraubung nimmt auch das Moment auf, welches durch die Last der Innenschiene verursacht wird (siehe Skizze 4.4).

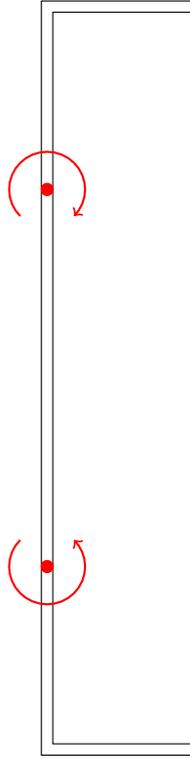


Abbildung 4.4: Skizze der Außenschiene

$$\text{Volumen Innenschiene } V_{IS} = 0.0004905m^3$$

$$\text{Dichte Aluminium } \rho_{Al} = 2700 \frac{kg}{m^3} \text{ [9, S.160/Tab 6.1]}$$

$$\text{Masse Innenschiene } m_{IS} = V_{IS} \cdot \rho_{Al} = 1.32435kg$$

$$\text{Fallbeschleunigung } g = 9.81 \frac{m}{s^2} \text{ [18, S.63]}$$

$$\text{Anliegendes Moment } M_{min} = m_{IS} \cdot 0.03m \cdot g = 0.39Nm \Rightarrow M = 0.06Nm \text{ (aufgerundet mit Sicherheitsaufschlag)}$$

$$\text{Elastizitätsmodul Aluminium } E_{Al} = 70000 \frac{N}{mm^2} \text{ [9, S.160/Tab 6.1]}$$

$$\text{Trägheitsmoment } I = \frac{b^3 \cdot h}{12} = \frac{3^3 \cdot 360}{12} = 810mm^4$$

$$\text{Biegelinie } w''(x) = \frac{M(x)}{E \cdot I_x} \Leftrightarrow E \cdot w'(x) = M \cdot x + C_1 \Leftrightarrow E \cdot w(x) = \frac{M \cdot x^2}{2} + C_1 \cdot x + C_2$$

$$\text{Einspannbedingungen } w(0) = C_2 = 0, w'(0) = C_1 = 0$$

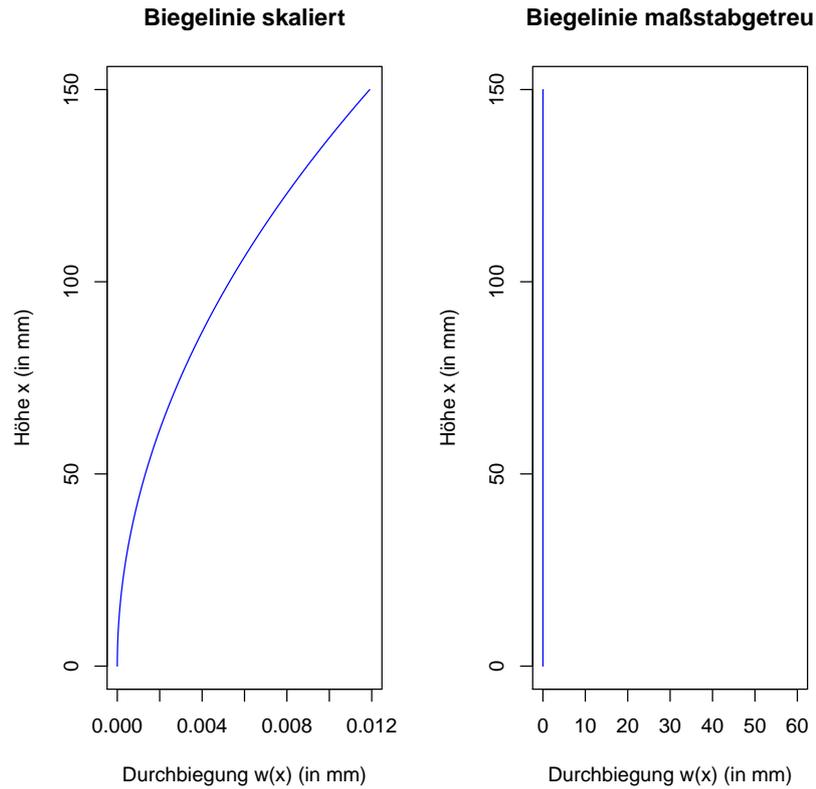


Abbildung 4.5: Biegelinie, verzerrte Ansicht (links) und maßstabgetreue Ansicht (rechts)

gelöste Biegelinie $w(x) = \frac{M \cdot x^2}{2 \cdot E \cdot I_x}$

Ablenkung durch die Biegung bei der Teleskopschiene $w(100mm) = 0.005mm$

Ablenkung durch die Biegung an der Spitze des U-Profiles $w(150mm) = 0.012mm$

Die Berechnungen ergeben somit eine äußerst kleine Abweichung aufgrund der Last der Innenschiene und müssen daher nicht weiter beachtet werden. In Abbildung 4.5 wird die Biegelinie dargestellt.

4.4 Steuerung

Zur Steuerung wurde der Mikrocomputer Raspberry Pi 3 in das Handhabungsgerät eingebaut. Die Programmierung der Steuerung erfolgte in Python 2. Es wurde hier nicht die aktuelle Version Python 3 verwendet, da das Pythonmodul für das Motorhat nur in Python 2 vorliegt. Die Programmcodes sind im Anhang unter Abschnitt 7.2 zu finden. Zur Interaktion mit dem Handhabungsgerät wird ein Telegram Chatbot eingesetzt (siehe auch Abschnitt 2.5.7). Um diesen Chatbot zu nutzen, muss sich der Anwender auf Telegram registrieren.

4.4.1 Steuerung des Handhabungsgeräts durch den Chatbot

Chatbot-Selbstregistrierung

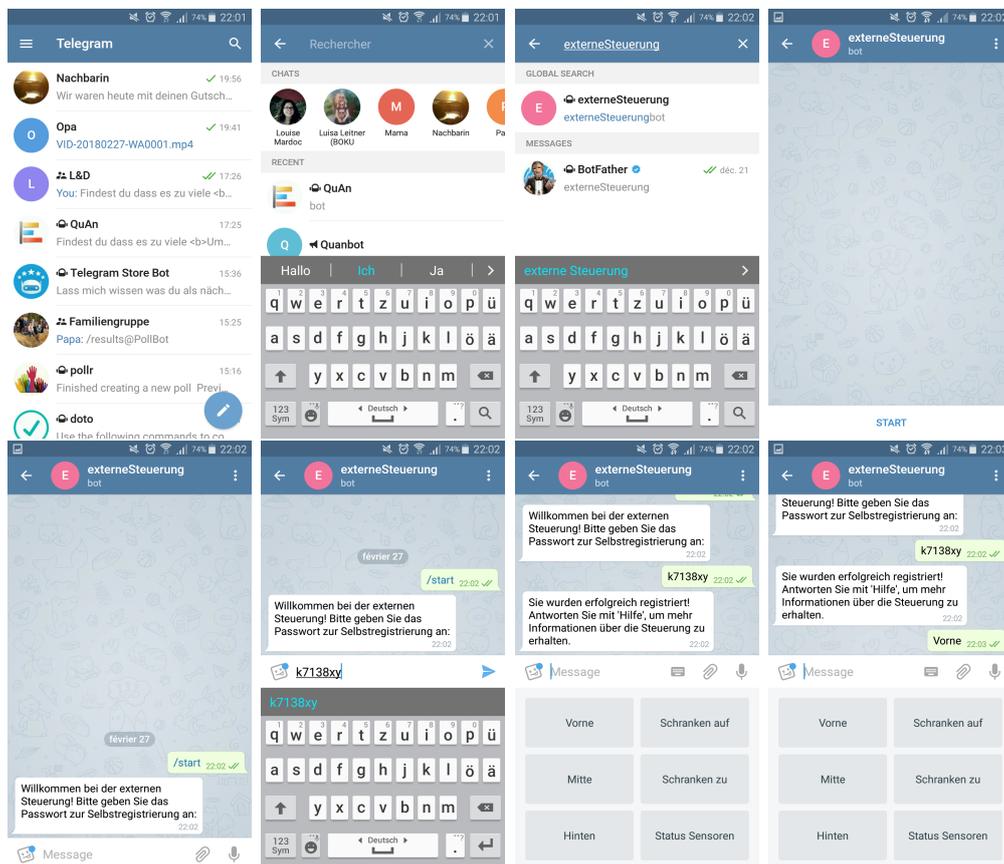


Abbildung 4.6: Telegram Chatbot

In Abbildung 4.6 werden 8 Screenshots von den ersten Schritten gezeigt. Als erstes muss über die Suchfunktion der Chatbot „externeSteuerung“ gesucht und ausgewählt werden. Hat man den Chatbot ausgewählt, erscheint beim erstmaligen Aufrufen unten ein Button mit der Aufschrift „Start“. Klickt man auf ihn, so erscheint eine Aufforderung, ein Passwort einzugeben. Dieses Passwort ist im Python-Code in der Datei *Telegramkommunikation.py* im oberen Abschnitt hinterlegt (siehe auch Abschnitt 7.2).

21 `password="k7138xy" #Passwort für die Selbstregistrierung`

Es wird empfohlen, das Passwort zu ändern, da sonst möglicherweise auch nicht berechnete Personen auf das Gerät Zugriff erlangen. Nach erfolgreicher Selbstregistrierung steht einem der Chatbot mit all seinen Funktionen zur Verfügung. Es wurde eine individuell angepasste Tastatur erstellt, um die wichtigsten Befehle schnell aufrufen zu können.

Steuerung der Schiene

Zur Orientierung wird das Handhabungsgerät in eine linke Seite (L) und eine rechte Seite (R) unterteilt. Weiters unterscheidet man zwischen Vorne (V), Mitte (M) und Hinten (H). Es gibt eine rechte und eine linke Innenschiene, die getrennt oder gemeinsam verfahren werden können. Um beide Schienen gemeinsam nach vorne zu verfahren, kann man auf der individuellen Tastatur auf *vorne* klicken, dem Chatbot eine Nachricht mit dem Text *vorne* oder eine Nachricht mit dem Text *v* schicken. Weiters existiert noch der Befehl `/ausfahrenVLR`, wobei das V für vorne und LR für Links und rechts steht. Für die Position Mitte (M) und Hinten (H) funktioniert die Steuerung genauso.

Möchte man nur eine Schiene (links oder rechts) verfahren, so funktioniert dies ausschließlich über die Kurzbefehle. Es stehen folgende Befehle zur Verfügung:

- `/ausfahrenVR` Rechte Schiene in vordere Position verfahren
- `/ausfahrenMR` Rechte Schiene in mittlere Position verfahren
- `/ausfahrenHR` Rechte Schiene in hintere Position verfahren
- `/ausfahrenVL` Linke Schiene in vordere Position verfahren
- `/ausfahrenML` Linke Schiene in mittlere Position verfahren
- `/ausfahrenHL` Linke Schiene in hintere Position verfahren
- `/ausfahrenVLR` Linke und rechte Schiene in vordere Position verfahren
- `/ausfahrenMLR` Linke und rechte Schiene in mittlere Position verfahren
- `/ausfahrenHLR` Linke und rechte Schiene in hintere Position verfahren

Position der Schiene

Die aktuelle Position der Schiene wird mittels 4 Sensoren (elektrische Schalter) ermittelt. Um den Status der Sensoren abzurufen, kann auf den Button *Status Sensoren* geklickt werden oder als Nachricht *Sensoren*, *Status Sensoren* oder *S* gesendet werden. HL=0 bedeutet, dass der hintere Schalter nicht gedrückt ist, HL=1 dagegen bedeutet, dass er aktiv ist. Sind nur die hinteren Schalter aktiv, so befindet sich die Schiene in der hinteren

Position. Sind sowohl hintere als auch vordere Schalter aktiv, so befindet sich die Schiene in der mittleren Position.

Steuerung der Schranken

Es gibt vier Schranken, diese werden ähnlich wie bei der Schiene VL, HL, VR und HR genannt. Man kann diese individuell oder gemeinsam steuern. Zum gemeinsamen Öffnen oder Schließen der Schranken gibt es die Befehle `/alleauf` bzw. `/allezu`, die Buttons *Schranken auf* und *Schranken zu*, sowie die Möglichkeit, als Nachricht folgenden Text zu senden: *Schranken auf*, *auf*, *Schranken zu* oder *zu*.

Weitere Chatbotfunktionen

Weitere Chatbotfunktionen bestehen, wenn man folgende Wörter an den Chatbot schickt:

Wer bin ich? Informationen über den User, wie z.B. die Userid

Befehle Übersicht über alle Steuerungsbefehle

Info Informationen über den User sowie die IP-Adresse des Raspberry

Infos ein Benachrichtigungen, welche Befehle gerade ausgeführt werden, werden eingeschaltet

Infos aus Benachrichtigungen, welche Befehle gerade ausgeführt werden, werden ausgeschaltet

Userliste Liste mit allen zur Steuerung berechtigten Personen

Fernwartung

Um den Programmcode zu ändern, kann man auch mittels Remotedesktopverbindung auf den Raspberry zugreifen. Dazu muss man sich im selben WLAN-Netz wie der Raspberry befinden. Um die IP-Adresse des Raspberrys herauszufinden, kann man im Chatbot das Codewort *Info* benutzen. Unter Windows steht für eine Remotedesktopverbindung das Programm *mstsc.exe* zur Verfügung. Man findet es am besten über die Suche, in die man *mstsc* eingibt. In das Feld *Computer* kann man entweder *Raspberry* oder die IP-Adresse des Raspberrys schreiben. Die IP-Adresse ist dabei der bessere Weg, da sich unter Umständen auch mehrere Raspberrys im Netz befinden könnten. Anschließend muss man sich noch anmelden. Die Session sollte auf *Xorg* gestellt werden, der Benutzername lautet *pi* und das Passwort *prog171921*. Die Python-Dateien befinden sich im Pfad */home/pi/Diplomarbeit*. Auf dem Desktop befindet sich die Datei *Infos* mit weiteren Informationen zur Programmierung.

4.4.2 Programmierung

In diesem Abschnitt wird der Programmcode näher erläutert. Der vollständige Programmcode ist im Anhang im Abschnitt 7.2 zu finden.

Das Steuerungsprogramm wurde am Raspberry Pi 3, Modell B in Python 2 geschrieben. Es besteht aus zwei Dateien: *Telegramkommunikation.py* und *getriebemotor_multiprocessing.py*. Zusätzlich mussten noch die Python-Module für das Schrittmotoren-HAT und für den Telegram Chatbot installiert werden.

Telegramkommunikation.py

Telegramkommunikation.py ist die Hauptdatei. Beim Starten des Raspberry wird diese Datei ausgeführt.

Zunächst werden alle erforderlichen Module importiert, der Telegram Chatbot eingerichtet, die PIN-Belegung beim Raspberry konfiguriert und die Datenbank für die Benutzerverwaltung geladen. Außerdem wird auch die zweite Python-Datei, *getriebemotor_multiprocessing.py*, importiert, um später auf dort implementierte Funktionen zuzugreifen.

Anschließend werden alle notwendigen Funktionen definiert, um das Handhabungsgerät zu steuern und Informationen über den Chatbot empfangen zu können. Die Steuerung der Schranken und Schienen wird im nächsten Absatz beschrieben.

Telegram unterscheidet zwischen normalen Textnachrichten und Befehlen. Ein Befehl ist ein Wort, das mit einem Slash beginnt, z.B. `/ausfahrenVLR`. Befehle werden in Telegram blau hinterlegt und können angeklickt werden, um den Befehl nochmals auszuführen. Befehle verknüpft man direkt mit einer Funktion im Pythoncode. Diese Verknüpfung sieht so aus:

```
350 dispatcher.add_handler(CommandHandler('ausfahrenVLR',
    ausfahrenVLR))
```

Hier wurde der Befehl `/ausfahrenVLR` mit der Pythonfunktion *ausfahrenVLR()* verknüpft.

Am Ende des Python-Codes wird noch die individuelle Tastatur in Telegram konfiguriert, mit der es möglich ist, die wichtigsten Befehle schnell auszuführen.

```
374 from telegram import ReplyKeyboardMarkup #Konfiguration der
    individuellen Tastatur
375 custom_keyboard=[[ 'Vorne', 'Schranken auf'], ['Mitte', 'Schranken
    zu'], ['Hinten', 'Status Sensoren']]
376 reply_markup=ReplyKeyboardMarkup(custom_keyboard)
```

Schrankensteuerung

Die Schranken werden durch Servomotoren betrieben. Die Servomotoren bekommen ein Steuerungssignal. Da die Motoren unterschiedlich montiert sind, benötigt jeder Motor ein anderes Steuerungssignal zum Öffnen und Schließen. Um alle Schranken zu öffnen, wird als erstes die Funktion *alleauf(bot,update)* in der Datei *Telegramkommunikation.py* aufgerufen. Diese ruft wiederum die Funktionen *HLauf(bot,uptdate)*, *HRauf(bot,uptdate)*, *VLauf(bot,uptdate)* und *VRauf(bot,uptdate)* auf. Die Funktion *HLauf(bot,update)* wiederum ruft die Funktion *pos(pin,winkel)* auf und übergibt dieser die PIN-ID und den Positions-Sollwert. Damit sich alle Schranken gleichzeitig öffnen und schließen, wird in der Funktion *pos(pin,winkel)* ein eigener Prozess gestartet, der für eine bestimmte Zeit ausgeführt wird.

Getriebemotorsteuerung

Die Getriebemotorsteuerung ist der komplexeste Teil der Programmierung. Gestartet wird der Ausfahrprozess mit der Funktion *ausfahren(bot,update,position,seite)*. Dieser Funktion wird die gewünschte Position (V, M, H) und die Seite (L, R, LR) als Parameter übergeben. In die Mitte kann nur Verfahren werden, wenn der alte Standort bekannt ist. Ist der Standort nicht bekannt, wird nichts unternommen und eine Mitteilung dem User gesendet. Ansonst wird als neuer Prozess die Funktion *ausfahren(position,seite)* in der Datei *getriebemotor_multiprocessing.py* gestartet. Diese Funktion analysiert als erstes, wo sich die Innenschiene derzeit befindet. Wird nur die linke oder rechte Seite verfahren, wird die alte Position zusammen mit der gewünschten Position und Seite an die Funktion *ausfahrsteuerung(position,seite,altepos)* übergeben. Sollen beide Seiten gleichzeitig verfahren werden, so wird die Ausfahrsteuerung in getrennten Prozessen für die linke und rechte Seite gestartet. Dies ist erforderlich, damit beide Seiten zeitgleich verfahren.

Die **Ausfahrsteuerung** startet den Prozess der Motorsteuerung. Der Prozess wird abgebrochen, wenn die gewünschte Position erreicht ist.

Die **Motorsteuerung** stellt zunächst die Richtung der Motoren ein, je nachdem welche Endposition gewünscht ist. In Abhängigkeit von Start- und Endposition sowie der Seite sind unterschiedliche Wegzeiten und Maximalgeschwindigkeiten definiert. Da die Kabel zu den linken Motoren länger sind als zu den rechten Motoren, ist auch der Widerstand dorthin größer. Daher bekommen die rechten Motoren etwas weniger Strom, damit sich beide Seiten gleich schnell bewegen. Weiters wird die Maximalgeschwindigkeit sehr stark reduziert, wenn die Startposition nicht bekannt ist. Dies ist jedoch nur in Ausnahmesituationen der Fall, z.B. ein plötzlicher Stromausfall während des Verfahrens oder nicht richtig angeschlossene Kabel.

Die Motoren werden kontinuierlich bis zur Maximalgeschwindigkeit beschleunigt. Nach einer bestimmten Zeit wird die Motorengeschwindigkeit gedrosselt. Im Regelbetrieb wird der Prozess von der Ausfahrsteuerung nun abgebrochen. Wird der Prozess nicht abge-

brochen, bewegt sich der Motor für eine bestimmte Zeit weiter und wird anschließend gestoppt.

4.4.3 Zeitmessung

Zur Untersuchung der Verfahrens- und Reaktionszeiten wurde eine Versuchsreihe gestartet. In diesem Abschnitt werden die Ergebnisse der Untersuchung diskutiert. Das Messprotokoll befindet sich in Tabelle 4.2. Die Zeiten wurden händisch gemessen, die Genauigkeit der Messungen liegt bei $\pm 50ms$.

Nr.	Datum & Uhrzeit	Beschreibung	Zeit (mm:ss:msms)	Raumtemperatur (°C)	Ort
1.	2.3.2018 12:42	Strom ein → erste Bewegung	01:43:59	13.2 °C	Daniel
2.	3.3.2018 13:58	Strom ein → erste Bewegung	01:44:86	13.3 °C	Daniel
3.	4.3.2018 19:32	Strom ein → erste Bewegung	00:57:95	12.8 °C	Daniel
4.	4.3.2018 19:37	M → V	00:06:90	13.0 °C	Daniel
5.	4.3.2018 19:38	V → M	00:06:48	13.0 °C	Daniel
6.	4.3.2018 19:38	M → V	00:07:06	13.0 °C	Daniel
7.	4.3.2018 19:38	V → M	00:07:12	13.0 °C	Daniel
8.	4.3.2018 19:37	M → H	00:07:31	13.0 °C	Daniel
9.	4.3.2018 19:37	H → M	00:08:62	13.0 °C	Daniel
10.	4.3.2018 19:37	M → H	00:06:67	13.0 °C	Daniel
11.	4.3.2018 19:37	H → M	00:07:09	13.0 °C	Daniel
12.	4.3.2018 19:40	M → V	00:06:60	13.0 °C	Daniel
13.	4.3.2018 19:40	V → M	00:07:06	13.0 °C	Daniel
14.	4.3.2018 19:40	M → H	00:07:07	13.0 °C	Daniel
15.	4.3.2018 19:40	H → V	00:12:10	13.0 °C	Daniel
16.	4.3.2018 20:00	V → H	00:12:09	14.1 °C	Daniel
17.	4.3.2018 20:02	H → V	00:11:83	14.1 °C	Daniel
18.	4.3.2018 20:02	V → H	00:13:72	14.1 °C	Daniel
19.	4.3.2018 20:02	H → V	00:11:17	14.1 °C	Daniel
20.	4.3.2018 20:03	V → H	00:11:59	14.1 °C	Daniel
21.	4.3.2018 20:04	Strom ein → erste Bewegung	00:57:30	14.1 °C	Daniel
22.	4.3.2018 20:05	Strom ein → erste Bewegung	00:57:01	14.1 °C	Daniel
23.	5.3.2018 07:37	Strom ein → erste Bewegung	00:56:34	14.1 °C	Daniel

Tabelle 4.2: Zeitmessungen

Einschaltzeit

Um die Einschaltzeit zu messen wurde zu Beginn das Handhabungsgerät vom Strom getrennt. Dann wurde in Telegram ein Befehl zum Ausfahren gegeben. Anschließend wurde der Strom eingeschaltet und gleichzeitig die Stoppuhr gestartet. Sobald sich das Gerät in Bewegung setzte, wurde die Zeit angehalten.

Die Einschaltzeit variiert sehr stark und beträgt zwischen 0:56 und 1:45 Minuten. Keine Auswirkung auf die Einschaltzeit hatte, ob der Raspberry vorher ordnungsgemäß heruntergefahren oder einfach der Strom abgeschaltet wurde.

Verfahrenszeit

Die Verfahrenszeit für die kurzen Strecken von der Mitte nach Vorne, von der Mitte nach Hinten und umgekehrt, liegt zwischen 6 und 9 Sekunden. Für die Verfahrenszeit sind zwei Faktoren ausschlaggebend: Die Reaktionszeit des Chatbots und die Justierung der Verfahrensgeschwindigkeit. Liegt die Verfahrenszeit über 10 Sekunden, so sollte im Pythoncode in der Datei *getriebemotor_multiprocessing.py* ab Zeile 270 die Ausfahrzeiten manuell nachjustiert werden.

Für die lange Verfahrensstrecke (Vorne nach Hinten und retour) liegen die Zeiten zwischen 11 und 12 Sekunden.

Kapitel 5

Schlussbetrachtung

Das in dieser Arbeit entwickelte Handhabungsgerät wurde speziell für die CNC-Portalfräsmaschine High-z-S-1000t der Firma CNC-Step entwickelt.¹ Allerdings lässt es sich auch leicht für andere Maschinen adaptieren. Man kann die Breite der Maschine verändern, indem man die Länge der beiden T-Träger verringert. Es ist auch möglich, nur eine Seite zu bewegen. Diese Möglichkeit ist steuerungstechnisch bereits implementiert und über den Telegram Chatbot ausführbar. Somit könnte man das Handhabungsgerät an einer großen Vielfalt von Maschinen einsetzen, zum Beispiel auch bei 3D-Druckern.

Bei der Konstruktion wurde entschieden, kein Portal für das Handhabungsgerät über der Fräse zu errichten. Dies erhöht maßgeblich die Flexibilität, da somit das Handhabungsgerät für jede annähernd bauähnliche Maschine verwendet werden kann.

5.1 Sicherheitskonzept

Da es sich bei dem Handhabungsgerät um einen ersten Prototypen handelt, muss dieser erst in einem Usecase ausführlich getestet werden. Insbesondere muss noch ein geeignetes Sicherheitskonzept für den Arbeitsraum entwickelt werden. Ein Sicherheitskonzept könnte zum Beispiel die Stromversorgung zu den Motoren unterbrechen, welche bei Betätigung eines Not-Aus-Schalters oder bei Durchbrechen einer Lichtschranke ausgelöst werden würde. Eine andere Möglichkeit wäre die komplette Trennung des Handhabungsgeräts vom Stromnetz. Dies ist ebenfalls ohne Schaden für Software und Hardware möglich. Allerdings muss man hier die Einschaltzeit von bis zu zwei Minuten beachten, bis das Gerät wieder betriebsbereit ist.

Grundsätzlich wird das Handhabungsgerät mit geringer Spannung und geringem Strom betrieben. Daher besteht kein unmittelbares oder hohes Risiko für sich in der Nähe befindlichen Personen. Dennoch könnten durch eine Verkettung unglücklicher Umstände

¹<https://www.cnc-step.de/cnc-fraese-high-z-s-1000t-kugelgewindetrieb-1000x600mm/>

Personen gefährdet werden, dies macht ein Sicherheitskonzept im Regelbetrieb notwendig.

5.2 Steuerung

Ein Raspberry Pi 3 übernimmt die Steuerung des Handhabungsgeräts. Für die optimale Verfahrensgeschwindigkeit mit einer sanften Bremsung wurden die dafür notwendigen Zeiten und Geschwindigkeiten im Pythoncode implementiert. Pflege wie regelmäßiges Ölen der Schiene, Sauberkeit und andere Umgebungsbedingungen haben Einfluss auf die tatsächlichen Verfahrenszeiten. Bei einer Änderung der Umgebungsbedingungen könnte es daher erforderlich sein, diese Zeiten zu ändern. Dies ist allerdings nur manuell im Pythoncode möglich. Falls diese Zeiten zu oft nachjustiert werden müssen, könnte auch ein Mechanismus programmiert werden, der die Verfahrenszeiten automatisch anpasst und nachjustiert.

5.2.1 Telegram Chatbot

Die Programmierung erfolgte derart, dass eine Adaptierung auf andere Benutzerinteraktionsmöglichkeiten leicht umsetzbar ist. Allerdings hat sich der Telegram Chatbot in den ersten Tests als sehr zuverlässig herausgestellt. Die Reaktionszeit kann ebenfalls als sehr gut bewertet werden, da sie in der Regel weniger als eine Sekunde beträgt. Daher lautet die Empfehlung, bei einer Implementierung in ein Gesamtsystem den Telegram Chatbot weiter zu entwickeln und weiterhin zu verwenden. Es wäre auch möglich, ein Kameramodul an den Raspberry anzuhängen. Dem User könnte so Fotos geschickt werden. Der User könnte von der Ferne überprüfen, ob alles in Ordnung ist und auch von der Ferne über den Weg des Werkstückes entscheiden.

Der Telegram Chatbot besticht nicht nur durch seine Zuverlässigkeit und Schnelligkeit, er bietet zusätzlich eine große Funktionsbreite und ist obendrein einfach zu programmieren.

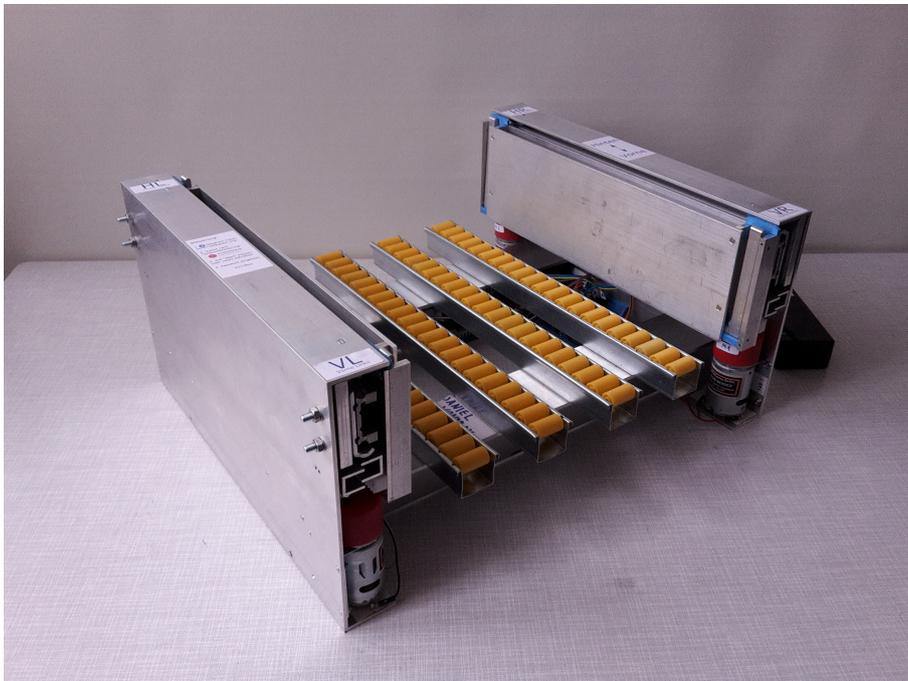


Abbildung 5.1: Realisiertes Handhabungsgerät

Kapitel 6

Verzeichnisse

Tabellenverzeichnis

2.1	Formelzeichenerklärung	10
3.1	Armlängen und geschlossene Scherenbreite bei verschiedenen Öffnungs- und Schließwinkel (auf ganze mm gerundet)	31
3.2	Zustände des Scherengitterroboters	34
3.3	Zukaufteile, Preise exkl. USt.	35
4.1	zugekaufte Bauteile	40
4.2	Zeitmessungen	51

Abbildungsverzeichnis

1.1	CNC-Portalfräse der Firma CNC-Step [4]	1
2.1	Montagelinie mit Werkstückträger [17, S.123]	5
2.2	Portalroboter [16, S.19]	6
2.3	Beispiele für flexible Ordnungssysteme [23, S.14]	8
2.4	Anordnungsmöglichkeiten von Antrieben bei Gummigurtförderern [24, S.139]	11
2.5	Spannvorrichtungen bei Gummigurtförderern [24, S.140]	11
2.6	Riemenförderer [24, S.151]	12
2.7	Einteilung der Unstetigförderer [24, S.214]	13
2.8	Hubgerüstkonstruktionen [24, S.243]	14
2.9	Industrieroboter Franka Emika [3]	17
2.10	Greiferarten und deren Vorkommenshäufigkeit [14, S.210]	18
2.11	Linearbewegung mittels Scherengitter [22, S.98]	19
2.12	Anwendungsbeispiel elektrischer Kompaktschlitten der Firma SMC [27] . .	21
2.13	Elektrozylinder 3,5kN; 719,99 € (Quelle: Conrad)	21
2.14	Elektrozylinder 0,2kN; 119,99 € (Quelle: Conrad)	22
2.15	Spannprinzip [20, S.327]	22
2.16	RKE 125 der Firma Röhm [17, S.123]	23
2.17	elektrischer Schraubstock der Firma Zentra (Quelle: http://www.shop.santool.de/)	23
2.18	Daten im Ladespeicher der SPS [30, S.12]	27
2.19	Rasperry mit aufgesetzten HAT [1]	28
2.20	Messenger-Nutzung weltweit [8]	29
3.1	Berechnungsskizze für die Varianten Schere offen und Schere zu	31
3.2	Kräfte-skizze für die offene Schere (Lagerbelastung)	32
3.3	Kräfte-skizze für die Lagerkräfte im Kranportal	33
3.4	Vereinfachte Kräfte-skizze für die Lagerkräfte im Kranportal	33
3.5	Förderbandkonzept: Mittels Bandförderer wird der Werkstückträger in das Innere der Fräsmaschine befördert	35
3.6	Förderbandkonzept: Das Förderband wird abgesenkt und der Werkstückträger auf die Spannvorrichtung aufgesetzt.	36
3.7	Regalbediengerät, Variante 1: Der Werkstückträger wird hochgehoben und in die gewünschte Position gebracht.	37
4.1	Endgültige Version des Handhabungsgeräts	38
4.2	Liste zugekaufter Aluprofile	41
4.3	Skizze der Kräfte am zu bewegenden Werkstückträger	43

4.4	Skizze der Außenschiene	44
4.5	Biegelinie, verzerrte Ansicht (links) und maßstabgetreue Ansicht (rechts) .	45
4.6	Telegram Chatbot	46
5.1	Realisiertes Handhabungsgerät	55
7.1	Elektrischer Schaltplan des Handhabungsgeräts	64

Literaturverzeichnis

- [1] Adafruit DC and Stepper Motor HAT for Raspberry Pi. <https://learn.adafruit.com/adafruit-dc-and-stepper-motor-hat-for-raspberry-pi/overview>. Accessed: 2018-02-28.
- [2] *c't Raspberry Pi*. c't special. Heise Medien GmbH, Hannover.
- [3] Franka Emika (Website des Herstellers). <https://www.franka.de>. Accessed: 2016-12-13.
- [4] HIGH-Z S-1000/T CNC FRÄSE. <https://www.cnc-step.de/cnc-fraese-high-z-s-1000t-kugelgewindetrieb-1000x600mm/>. Accessed: 2018-03-01.
- [5] Introducing Raspberry PI HATs. <https://www.raspberrypi.org/blog/introducing-raspberry-pi-hats>. Accessed: 2018-02-28.
- [6] Mitsubishi Electric. <http://www.geva.at/pdf/angebote/RH-3FH.pdf>. Accessed: 2016-12-14.
- [7] Telegram Website. <https://telegram.org/>. Accessed: 2018-02-27.
- [8] Andreas Floemer. Top 10: Das sind die meistgenutzten Messenger-Apps weltweit. <https://t3n.de/news/top-10-meistgenutzte-messenger-weltweit-2017-839993/>. Accessed: 2018-02-27.
- [9] Bozena Arnold. *Werkstofftechnik für Wirtschaftsingenieure*. Springer Berlin Heidelberg, Berlin, Heidelberg.
- [10] Michael F. Ashby, David R. H. Jones, and Michael [Hrsg.] Heinzlmann. *Eigenschaften, Mechanismen und Anwendungen; Werkstoffe; An Introduction to Properties, Applications and Design; Engineering materials*. Elsevier, München, 2006.
- [11] Donald R. Askeland. *Materialwissenschaften; Grundlagen, Übungen, Lösungen; The science and engineering of materials*. Spektrum-Lehrbuch. Spektrum Akad. Verl., Heidelberg [u.a.], 1996.
- [12] Jo Bager. Hannover Messe: Industrieroboter zum Kampfpreis. <https://www.heise.de/make/meldung/Hannover-Messe-Industrieroboter-zum-Kampfpreis-3186599.html>. Published: 26-04-2016, Accessed: 2016-12-13.
- [13] Erik Bartmann. *Faszinierende Elektronik-Projekte mit Scratch, Arduino und Raspberry Pi*. O'Reilly basics. O'Reilly, Köln [u.a.].

- [14] Bernd [Hrsg.] Brodbeck and Arbeitsgemeinschaft Handhabungssysteme. *Handhabungssysteme; Entscheidungshilfen und Einsatzerfahrungen aus technischer Sicht*. Schriftenreihe Humanisierung des Arbeitslebens ; 11. VDI-Verl., Düsseldorf, 1981.
- [15] Erich F. Engelhardt. *Coole Projekte mit Raspberry Pi*. Franzis, Haar bei München.
- [16] Heinz Fuchs. *Flexible Automation*. Schriftenreihe des Wirtschaftsförderungsinstituts ; 203 : Technologie. Wien, 1990.
- [17] Materialfluss und Logistik Gesellschaft Fördertechnik. *Werkstückträger für die automatisierte Fertigung und Montage; Tagung Stuttgart, 13. Oktober 1988*. VDI-Berichte ; 708. VDI-Verl., Düsseldorf, 1988.
- [18] Ulrich Harten. *Physik; Einführung für Ingenieure und Naturwissenschaftler*. Springer-Lehrbuch. Springer Berlin Heidelberg, Berlin, Heidelberg, Berlin, Heidelberg, 2005.
- [19] Peter A. [Hrsg.] Henning. *Taschenbuch Programmiersprachen*. Fachbuchverl. Leipzig im Hanser-Verl., München Wien.
- [20] Stefan Hesse. *Grundlagen der Handhabungstechnik; mit 25 Tabellen, 5 Tafeln sowie 26 Übungsaufgaben und 115 Kontrollfragen ; mit einer CD-ROM mit Zusatzmaterialien, Videos und Wörterbüchern Deutsch - Englisch und Englisch - Deutsch*. Hanser, München, 2013.
- [21] Christian Immler. *Schnelleinstieg Raspberry Pi*. Franzis, Haar bei München.
- [22] Heinz W. Katzenmeier. *Grundlagen der Robotertechnik; Tipps und Tricks für den Selbstbau*. Elektor-Verl., Aachen, 2004.
- [23] Gamal Lashin. *Rechnerunterstützte Bereitstellung von ungeordneten Kleinteilen für die automatisierte Montage*. Produktionstechnik - Berlin ; 110. Hanser, München ; Wien, 1993. Zugl.: Berlin, Techn. Univ., Diss., 1992.
- [24] Heinrich Martin. *Transport- und Lagerlogistik; Planung, Struktur, Steuerung und Kosten von Systemen der Intralogistik ; mit 48 Tabellen*. Praxis. Vieweg + Teubner, Wiesbaden, 2011.
- [25] Wilfried [Hrsg.] Pläßmann and Detlef [Hrsg.] Schulz. *Handbuch Elektrotechnik*. Springer Fachmedien Wiesbaden Imprint: Springer Vieweg, Wiesbaden.
- [26] Dennis Patrick Rühmer and Galileo Press GmbH [Verlag]. *Heimserver mit Raspberry und Banana Pi*. Rheinwerk Computing. Rheinwerk Computing, Bonn.
- [27] SMC. Elektrische Kompaktschlitten (Katalog). https://content2.smcetech.com/pdf/LES-E_DE.pdf. Accessed: 2016-12-22.

- [28] Günter Ullrich. *Fahrerlose Transportsysteme: Eine Fibel - mit Praxisanwendungen - zur Technik - für die Planung*, volume 22 of *Fortschritte der Robotik*. Springer Fachmedien Wiesbaden, Wiesbaden, 2014.
- [29] Wolfgang Weber. *Industrieroboter; Methoden der Steuerung und Regelung ; mit 33 Übungsaufgaben sowie einer begleitenden Internetseite*. Fachbuchverl. Leipzig im Carl-Hanser-Verl., München, 2009.
- [30] Günter Wellenreuther and Dieter Zastrow. *Automatisieren mit SPS - Theorie und Praxis; Programmierung: DIN EN 61131-3, STEP 7, CoDeSys, Entwurfsverfahren, Bausteinbibliotheken ; Applikationen: Steuerungen, Regelungen, Antriebe, Safety ; Kommunikation: AS-i-Bus, PROFIBUS, Ethernet-TCP/IP, PROFINET, Web-Technologien, OPC, WLAN. Mit 108 Steuerungsbeispielen und 8 Projektierungen; Automatisieren mit Speicherprogrammierbarer Steuerung*. Studium. Vieweg + Teubner, Wiesbaden, 2011. Online-Ausg. *Automatisieren mit SPS - Theorie und Praxis*.

Stichwortverzeichnis

- Ablaufsprache (AS), 26
- Anweisungsliste (AWL), 26
- Arduino, 28
- Automatisierung, 26
- Automatisierungsgerät, 26

- Bandförderer, 10
- Betriebsdatenerfassung (BDE), 25
- Bot, 29
- Bussystem, 27

- Chatbot, 29
- Computer Aided Manufacturing (CAM), 25

- Effektor, 16
- Elektromotor, 24

- Führungsgröße, 26
- Fahrerloses Transportfahrzeug (FTF), 14
- Freisichthubgerüst, 13
- Funktionsbausteinsprache (FBS, FUP), 26

- Gabelstapler, 13
- Gummigurtförderer, 10

- Handhaben, 3
- HAT, 27
- Hubgerüst, 13

- Kontaktplansprache (KOP), 26

- Mikrocontroller, 28

- Portalroboter, 5
- Python, 28

- Rückwärtstransformation, 16
- Raspberry Pi, 27
- Regelgröße, 26
- Regelkreis, 26
- Regelung, 26
- Riemenförderer, 11
- Roboterkoordinaten, 16

- Schrittmotor, 24
- Servomotor, 24
- SIMATIC Micro Memory Card, 26
- SMC, 20
- Spannanlagen, 11
- Speicherprogrammierbare Steuerung (SPS), 25
- Steuerung, 26
- strukturierter Text (ST, SCL), 26

- Telegram, 29
- Tool Center Point (TCP), 16

- Vorwärtstransformation, 16

- Weltkoordinaten, 16
- Werkstückträger, 5

- Zweifachteleskop-Hubgerüst, 13

Kapitel 7

Anhang

7.1 Schaltplan

In der Abbildung auf der nächsten Seite findet man den Schaltplan des Handhabungsgeräts. Die Farbe der Verbindungen in der Abbildung entspricht der Kabelfarbe im realisierten Handhabungsgerät.

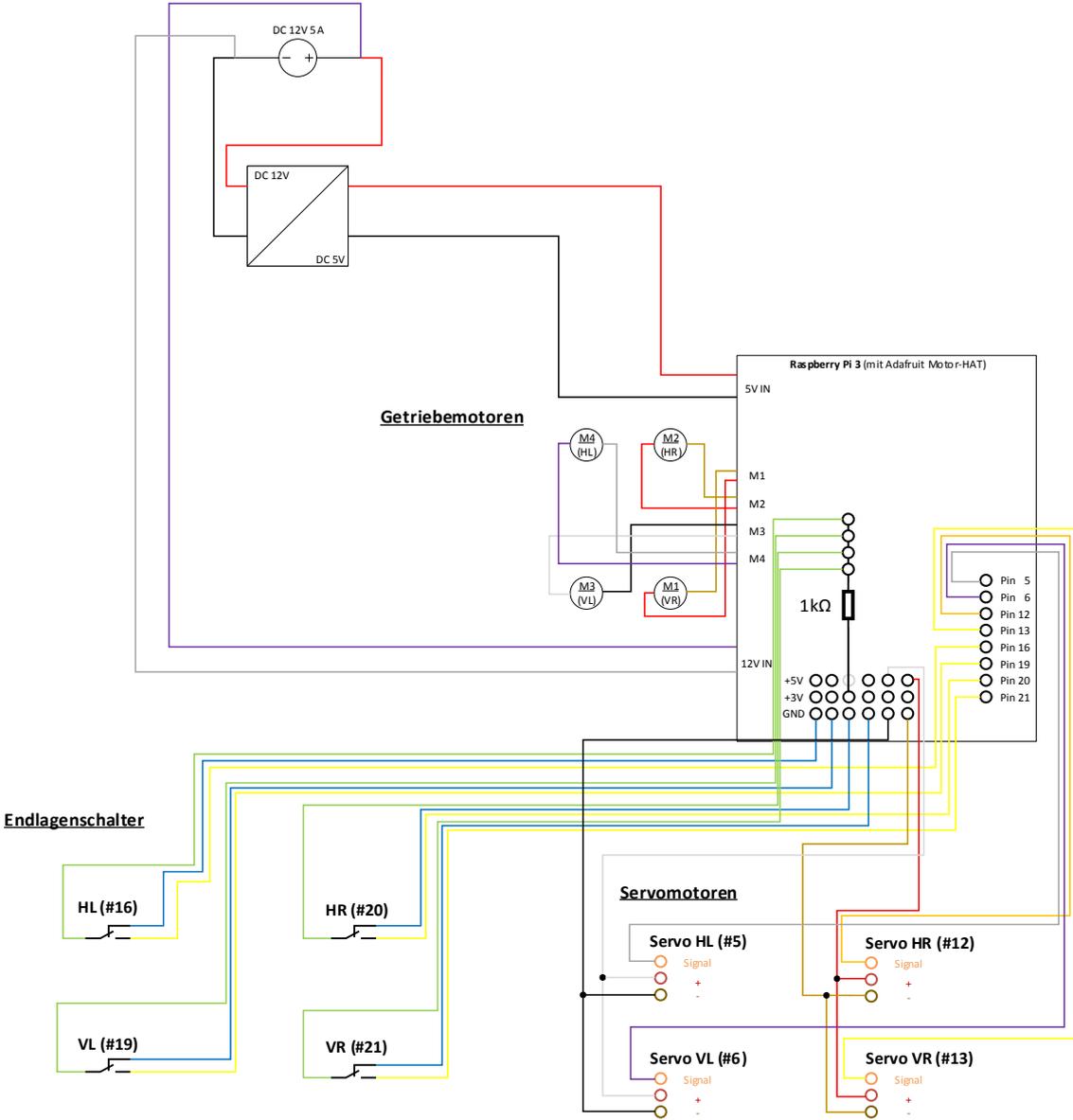


Abbildung 7.1: Elektrischer Schaltplan des Handhabungsgeräts

7.2 Programmcode

Listing 7.1: Telegramkommunikation.py

```

1  # -*- coding: utf-8 -*-
2  #https://github.com/python-telegram-bot/python-telegram-bot/wiki
   /Extensions-%E2%80%93-Your-first-Bot
3  import os
4  os.chdir('/home/pi/Diplomarbeit') #Arbeitsverzeichnis umstellen
5  from multiprocessing import Process
6  from telegram.ext import Updater
7  #from getriebemotor_multiprocessing import ausfahren
8  updater = Updater(token='490646029:AAGLtli91Mp-
   DvqfaYMQ9GTsoaURHuJAR58') #Telegram Chatbot
9  dispatcher = updater.dispatcher
10 import logging
11 logging.basicConfig(format='%(asctime)s - %(name)s - %(levelname)
   )s - %(message)s', level=logging.INFO)
12
13 import RPi.GPIO as GPIO #PIN-Belegung am Raspberry
14 GPIO.setmode(GPIO.BCM)
15 GPIO.setup(13,GPIO.OUT) #VR
16 GPIO.setup(12,GPIO.OUT) #HR
17 GPIO.setup(6,GPIO.OUT) #VL
18 GPIO.setup(5,GPIO.OUT) #HL
19
20 import time
21
22 password="k7138xy" #Passwort für die Selbstregistrierung
23 selbstregistrierung="nein"
24
25 import sqlite3
26 connection=sqlite3.connect("externeSteuerung.db") #Laden und
   Aufbau der Datenbank für die User
27 c=connection.cursor()
28 c.execute("""create table if not exists users(userid int primary
   key, fname varchar(20), lname varchar(20),
   benachrichtigungen varchar(20));""")
29 #c.execute("""delete from users""") #hier werden alle User gelö
   scht

```

```

30 connection.commit()
31 connection.close()
32
33
34 import getriebemotor_multiprocessing
35
36 def userpruefung(userid):
37     connection=sqlite3.connect("externeSteuerung.db")
38     cursor=connection.cursor()
39     cursor.execute("select * from users where userid=?", (userid
40         ,))
41     res=cursor.fetchone()
42     connection.close()
43     if res==None:
44         return False
45     else:
46         return True
47
48 def bf(userid): #benachrichtigungen funktion
49     connection=sqlite3.connect("externeSteuerung.db")
50     cursor=connection.cursor()
51     cursor.execute("select benachrichtigungen from users where
52         userid=?", (userid,))
53     res=cursor.fetchone()
54     connection.close()
55     if res==None:
56         return "ja"
57     else:
58         return res[0]
59
60 def start(bot, update):
61     global selbstregistrierung
62     if userpruefung(update.message.chat.id)==False:
63         bot.send_message(chat_id=update.message.chat_id, text="
64             Willkommen bei der externen Steuerung! Bitte geben
65             Sie das Passwort zur Selbstregistrierung an:")
66         selbstregistrierung="start"
67     else:

```

```

65     bot.send_message(chat_id=update.message.chat_id ,
66         reply_markup=reply_markup , text="Willkommen bei der
        externen Steuerng! Sie besitzen bereits alle
        erforderlichen Berechtigungen. Für mehr Informationen
        mit 'Hilfe' antworten.")
67 def echo(bot , update): #wird ausgeführt, wenn der User einen
        Text in Telegram an den Chatbot schickt
68     benachrichtigungen=bf(update.message.chat.id)
69     global selbstregistrierung
70     if selbstregistrierung=="start":
71         selbstregistrierung="nein"
72         if update.message.text.lower()=="password":
73             connection2=sqlite3.connect("externeSteuerung.db")
74             c2=connection2.cursor()
75             c2.execute("insert or ignore into users values
                (?, ?, ?, 'aus ')" ,(update.message.chat.id , update.
                message.chat.first_name , update.message.chat.
                last_name))
76             connection2.commit()
77             connection2.close()
78             bot.send_message(chat_id=update.message.chat_id ,
                reply_markup=reply_markup , text="Sie wurden
                erfolgreich registriert! Antworten Sie mit 'Hilfe
                ', um mehr Informationen über die Steuerung zu
                erhalten.")
79         else:
80             bot.send_message(chat_id=update.message.chat_id ,
                text="Das eingegebene Passwort ist falsch. Für
                einen erneuten Versuch mit /start antworten.")
81
82     elif update.message.text.lower()=="wer bin ich?" or update.
        message.text.lower()=="wer bin ich":
83         print update.message
84         t="Hallo "
85         try:
86             t+=update.message.chat.first_name
87             t+=" "
88         except:
89             print "No first name"

```

```

90     try:
91         t+=update.message.chat.last_name
92     except:
93         print "No last name"
94         t+=" , ihre ID lautet: "
95         t+=str(update.message.chat.id)
96         bot.send_message(chat_id=update.message.chat_id , text=t)
97 elif userpruefung(update.message.chat.id)==False:
98     bot.send_message(chat_id=update.message.chat_id , text="
99         Hierfuer fehlt Ihnen leider die Berechtigung! Fuer
100        Userinformationen 'Wer bin ich?' fragen , um sich
101        selbst zu registrieren mit /start antworten.")
102 elif update.message.text==" Befehle":
103     bot.send_message(chat_id=update.message.chat_id , text="
104        Kurzbefehle: 'auf', 'zu' (alle Schranken auf oder zu);
105        'V', 'M', 'H' (beide Innenschienen nach vorne, mitte,
106        hinten); Schranken: /VLlauf /VLzu ... Schiene: /
107        ausfahrenVL /ausfahrenML /ausfahren HL /ausfahrenVLR
108        /ausfahrenMLR /ausfahrenHLR /ausfahrenVR ...")
109 elif update.message.text==" Hilfe":
110     t="Eine Uebersicht ueber alle Steuerungsbefehle erhalten
111     Sie , in dem Sie 'Befehle' schreiben."
112     t+="Ihre ID erhalten Sie mit 'Wer bin ich?'.
113     Informationen gibt es mit dem Codeword 'Info'. "
114     t+="Benachrichtigungen können mit 'Infos ein' oder '
115     Infos aus' ein- und ausgeschaltet werden."
116     t+="Eine Liste mit allen berechtigten Usern kann mittels
117     'Userliste' abgefragt werden."
118     bot.send_message(chat_id=update.message.chat_id , text=t)
119
120 elif update.message.text==" Hallo":
121     bot.send_message(chat_id=update.message.chat_id , text="
122     Hallo zurueck")
123 elif update.message.text==" Info":
124     print update.message
125     t=" Hallo "
126     try:
127         t+=update.message.chat.first_name
128         t+=" "
129     except:

```

```

117         print "No first name"
118     try:
119         t+=update.message.chat.last_name
120
121     except:
122         print "No last name"
123         t+=" , ihre ID lautet: "
124         t+=str(update.message.chat.id)
125         ip=os.popen('ip addr show wlan0').read().split("inet ")
126             [1].split("/")[0]
127         t+=" . Die WLAN-IP des Handhabungsgeraets lautet: "
128         t+=ip
129         bot.send_message(chat_id=update.message.chat_id, text=t)
130     elif update.message.text.lower()=="auf" or update.message.
131         text.lower()=="schracken auf":
132         alleauf(bot, update)
133     elif update.message.text.lower()=="zu" or update.message.
134         text.lower()=="schracken zu":
135         allezu(bot, update)
136     elif update.message.text=="H" or update.message.text.lower
137         ()=="hinten":
138         ausfahrenHLR(bot, update)
139     elif update.message.text=="M" or update.message.text.lower()
140         == "mitte":
141         ausfahrenMLR(bot, update)
142     elif update.message.text=="V" or update.message.text.lower()
143         == "vorne":
144         ausfahrenVLR(bot, update)
145     elif update.message.text.lower()=="initialisieren" or update
146         .message.text.lower()=="initialisierung" or update.
147         message.text.lower()=="init":
148         initialisieren(bot, update)
149     elif update.message.text.lower()=="infos ein":
150         conn=sqlite3.connect("externeSteuerung.db")
151         c=conn.cursor()
152         c.execute("update users set Benachrichtigungen='ein'
153             where userid=?", (update.message.chat.id,))
154         conn.commit()
155         conn.close()

```

```

147     bot.send_message(chat_id=update.message.chat_id, text="
        Benachrichtigungen wurden eingeschaltet.")
148     elif update.message.text.lower()=="infos aus":
149         conn=sqlite3.connect("externeSteuerung.db")
150         c=conn.cursor()
151         c.execute("update users set Benachrichtigungen='aus'
            where userid=?", (update.message.chat.id,))
152         conn.commit()
153         conn.close()
154         bot.send_message(chat_id=update.message.chat_id, text="
            Benachrichtigungen wurden ausgeschaltet.")
155     elif update.message.text.lower()=="sensoren" or update.
        message.text.lower()=="s" or update.message.text.lower()
        =="status sensoren" :
156         #print "#16:", GPIO.input(16), "#19:", GPIO.input(19),
            "#20:", GPIO.input(20), "#21:", GPIO.input(21), "->
            altepos=H (#54)"
157         t=" Sensoren: HL="
158         t+=str(GPIO.input(16))
159         t+=" , VL="
160         t+=str(GPIO.input(19))
161         t+=" , HR="
162         t+=str(GPIO.input(20))
163         t+=" , VR="
164         t+=str(GPIO.input(21))
165         bot.send_message(chat_id=update.message.chat_id, text=t)
166     elif update.message.text.lower()=="userliste":
167         connection=sqlite3.connect("externeSteuerung.db")
168         c=connection.cursor()
169         c.execute("select * from users")
170         res=c.fetchall()
171         bot.send_message(chat_id=update.message.chat_id, text=
            res)
172         connection.close()
173     else :
174         bot.send_message(chat_id=update.message.chat_id, text="
            Ich habe Sie leider nicht verstanden. Falls Sie Hilfe
            benoetigen schreiben Sie 'Hilfe'. Eine Liste aller
            Befehle erhalten Sie mit dem Stichwort 'Befehle'.")
175

```

```

176 def ausfahrenVR (bot , update) :
177     ausfahren (bot , update , 'V' , 'R' )
178 def ausfahrenMR (bot , update) :
179     ausfahren (bot , update , 'M' , 'R' )
180 def ausfahrenHR (bot , update) :
181     ausfahren (bot , update , 'H' , 'R' )
182 def ausfahrenVL (bot , update) :
183     ausfahren (bot , update , 'V' , 'L' )
184 def ausfahrenML (bot , update) :
185     ausfahren (bot , update , 'M' , 'L' )
186 def ausfahrenHL (bot , update) :
187     ausfahren (bot , update , 'H' , 'L' )
188 def ausfahrenVLR (bot , update) :
189     ausfahren (bot , update , 'V' , 'LR' )
190     if GPIO.input (16)==0 and GPIO.input (19)==1 and GPIO.input
191         (20)==0 and GPIO.input (21)==1:
192         if bf (update . message . chat . id)==" ein" :
193             bot . send_message ( chat_id=update . message . chat_id ,
194                 text="Endposition wurde erreicht." )
195     else :
196         bot . send_message ( chat_id=update . message . chat_id , text="
197             FEHLER! Endposition wurde nicht erreicht." )
198 def ausfahrenMLR (bot , update) :
199     ausfahren (bot , update , 'M' , 'LR' )
200     if GPIO.input (16)==1 and GPIO.input (19)==1 and GPIO.input
201         (20)==1 and GPIO.input (21)==1:
202         if bf (update . message . chat . id)==" ein" :
203             bot . send_message ( chat_id=update . message . chat_id ,
204                 text="Endposition wurde erreicht." )
205     else :
206         bot . send_message ( chat_id=update . message . chat_id , text="
207             FEHLER! Endposition wurde nicht erreicht." )
208 def ausfahrenHLR (bot , update) :
209     ausfahren (bot , update , 'H' , 'LR' )
210     if GPIO.input (16)==1 and GPIO.input (19)==0 and GPIO.input
211         (20)==1 and GPIO.input (21)==0:
212         if bf (update . message . chat . id)==" ein" :
213             bot . send_message ( chat_id=update . message . chat_id ,
214                 text="Endposition wurde erreicht." )
215     else :

```

```

208         bot.send_message(chat_id=update.message.chat_id, text="
           FEHLER! Endposition wurde nicht erreicht.")
209 def initialisieren(bot, update):
210     alleauf(bot, update)
211     ausfahren(bot, update, 'init', 'LR')
212     if GPIO.input(16)==1 and GPIO.input(19)==0 and GPIO.input
       (20)==1 and GPIO.input(21)==0:
213         if bf(update.message.chat.id)=="ein":
214             bot.send_message(chat_id=update.message.chat_id,
                text="Endposition wurde erreicht.")
215     else:
216         bot.send_message(chat_id=update.message.chat_id, text="
           FEHLER! Endposition wurde nicht erreicht.")
217
218
219 def ausfahren(bot, update, position, seite):
220     benachrichtigungen=bf(update.message.chat.id)
221     if userpruefung(update.message.chat.id)==False:
222         bot.send_message(chat_id=update.message.chat_id, text="
           Hierfuer fehlt Ihnen leider die Berechtigung!")
223     #elif position=='M' and (((seite=='R' or seite=='LR') and
       GPIO.input(21)==0 and GPIO.input(20)==0) or ((seite=='L'
       or seite=='LR') and GPIO.input(16)==0 and GPIO.input(19)
       ==0)):
224     elif position!="init" and (((seite=='R' or seite=='LR') and
       GPIO.input(21)==0 and GPIO.input(20)==0) or ((seite=='L'
       or seite=='LR') and GPIO.input(16)==0 and GPIO.input(19)
       ==0)):
225         bot.send_message(chat_id=update.message.chat_id, text="
           Standort nicht bekannt. Bitte das Gerat /
           initialisieren. Dazu werden die Schranken geoffnet
           und die Schiene nach vorne und nach hinten verfahren.
           ")
226         print "#19", GPIO.input(19)
227         print "#16", GPIO.input(16)
228         print "seite", seite
229     else:
230         if benachrichtigungen=="ein":
231             bot.send_message(chat_id=update.message.chat.id,
                text="RBG wird nun verfahren.")

```

```

232     print("Geraet wird nun verfahren (#93)")
233     p2=Process(target=getriebemotor_multiprocessing.
                ausfahren, args=(position, seite))
234     p2.start()
235     p2.join()
236     print "Ausfahrprozess beendet!"
237
238 def VRauf(bot, update):
239     benachrichtigungen=bf(update.message.chat.id)
240     if userpruefung(update.message.chat.id)==False:
241         bot.send_message(chat_id=update.message.chat_id, text="
                Hierfuer fehlt Ihnen leider die Berechtigung!")
242     else:
243         print("VRauf")
244         pos(13,6.5)
245         if benachrichtigungen=="ein":
246             bot.send_message(chat_id=update.message.chat_id,
                text="Schranken VR wurde geoeffnet.")
247
248 def VRzu(bot, update):
249     benachrichtigungen=bf(update.message.chat.id)
250     if userpruefung(update.message.chat.id)==False:
251         bot.send_message(chat_id=update.message.chat_id, text="
                Hierfuer fehlt Ihnen leider die Berechtigung!")
252     else:
253         print("VRzu")
254         pos(13,13)
255         if benachrichtigungen=="ein":
256             bot.send_message(chat_id=update.message.chat_id,
                text="Schranken VR wurde geschlossen.")
257
258 def HRauf(bot, update):
259     benachrichtigungen=bf(update.message.chat.id)
260     if userpruefung(update.message.chat.id)==False:
261         bot.send_message(chat_id=update.message.chat_id, text="
                Hierfuer fehlt Ihnen leider die Berechtigung!")
262     else:
263         print("HRauf")
264         pos(12,12.5)
265         if benachrichtigungen=="ein":

```

```

266         bot.send_message(chat_id=update.message.chat_id ,
267                          text="Schranken HR wurde geoeffnet.")
268 def HRzu(bot , update):
269     benachrichtigungen=bf(update.message.chat.id)
270     if userpruefung(update.message.chat.id)==False:
271         bot.send_message(chat_id=update.message.chat_id , text="
272             Hierfuer fehlt Ihnen leider die Berechtigung!")
273     else:
274         print("HRzu")
275         pos(12,7.5)
276         if benachrichtigungen=="ein":
277             bot.send_message(chat_id=update.message.chat_id ,
278                             text="Schranken HR wurde geschlossen.")
279
280 def VLauf(bot , update):
281     benachrichtigungen=bf(update.message.chat.id)
282     if userpruefung(update.message.chat.id)==False:
283         bot.send_message(chat_id=update.message.chat_id , text="
284             Hierfuer fehlt Ihnen leider die Berechtigung!")
285     else:
286         print("VLauf")
287         pos(6,10.8)
288         if benachrichtigungen=="ein":
289             bot.send_message(chat_id=update.message.chat_id ,
290                             text="Schranken VR wurde geoeffnet.")
291
292 def VLzu(bot , update):
293     benachrichtigungen=bf(update.message.chat.id)
294     if userpruefung(update.message.chat.id)==False:
295         bot.send_message(chat_id=update.message.chat_id , text="
296             Hierfuer fehlt Ihnen leider die Berechtigung!")
297     else:
298         print("VLzu")
299         pos(6,6.0)
300
301 def HLauf(bot , update):
302     benachrichtigungen=bf(update.message.chat.id)
303     if userpruefung(update.message.chat.id)==False:

```

```

299         bot.send_message(chat_id=update.message.chat_id, text="
           Hierfuer fehlt Ihnen leider die Berechtigung!")
300     else:
301         print("HLauf")
302         pos(5,6.5)
303         if benachrichtigungen=="ein":
304             bot.send_message(chat_id=update.message.chat_id,
                               text="Schranken HL wurde geoeffnet.")
305
306 def HLzu(bot, update):
307     benachrichtigungen=bf(update.message.chat.id)
308     if userpruefung(update.message.chat.id)==False:
309         bot.send_message(chat_id=update.message.chat_id, text="
           Hierfuer fehlt Ihnen leider die Berechtigung!")
310     else:
311         print("HLzu")
312         pos(5,11.2)
313         if benachrichtigungen=="ein":
314             bot.send_message(chat_id=update.message.chat_id,
                               text="Schranken HL wurde geschlossen.")
315
316 def alleauf(bot, update):
317     benachrichtigungen=bf(update.message.chat.id)
318     print "Alle Schranken werden nun geoeffnet"
319     HLauf(bot, update)
320     VLauf(bot, update)
321     HRauf(bot, update)
322     VRauf(bot, update)
323     if benachrichtigungen=="ein":
324         bot.send_message(chat_id=update.message.chat_id,
                               text="Alle Schranken sind nun geoeffnet. (Um
                               Benachrichtigungen auszuschalten bitte mit 'infos
                               aus' antworten.)")
325
326 def allezu(bot, update):
327     benachrichtigungen=bf(update.message.chat.id)
328     print "Alle Schranken werden nun geschlossen"
329     HLzu(bot, update)
330     VLzu(bot, update)
331     HRzu(bot, update)

```

```

332 VRzu(bot , update)
333     if benachrichtigungen=="ein" :
334         bot.send_message(chat_id=update.message.chat_id ,
335                             text="Alle Schranken sind nun geschlossen. (Um
336                                 Benachrichtigungen auszuschalten bitte mit 'infos
337                                 aus' antworten.)")
338
339
340
341
342 def pos(pin , winkel) :
343     p3=Process(target=posprocess , args=(pin , winkel))
344     p3.start()
345
346
347
348 from telegram.ext import CommandHandler , MessageHandler , Filters
349 start_handler = CommandHandler('start' , start)
350 dispatcher.add_handler(start_handler)
351
352 echo_handler = MessageHandler(Filters.text , echo)
353 dispatcher.add_handler(echo_handler)
354
355
356
357 #Ab Zeile345: Verknüpfung der Telegrammbefehle mit Funktionen
358 dispatcher.add_handler(CommandHandler('ausfahrenVR' , ausfahrenVR)
359                             )
360 dispatcher.add_handler(CommandHandler('ausfahrenMR' , ausfahrenMR)
361                             )
362 dispatcher.add_handler(CommandHandler('ausfahrenHR' , ausfahrenHR)
363                             )
364 dispatcher.add_handler(CommandHandler('ausfahrenVL' , ausfahrenVL)
365                             )
366 dispatcher.add_handler(CommandHandler('ausfahrenML' , ausfahrenML)
367                             )

```

```

363 dispatcher.add_handler(CommandHandler('ausfahrenHL', ausfahrenHL)
    )
364 dispatcher.add_handler(CommandHandler('ausfahrenVLR',
    ausfahrenVLR))
365 dispatcher.add_handler(CommandHandler('ausfahrenMLR',
    ausfahrenMLR))
366 dispatcher.add_handler(CommandHandler('ausfahrenHLR',
    ausfahrenHLR))
367 dispatcher.add_handler(CommandHandler('V', ausfahrenVLR))
368 dispatcher.add_handler(CommandHandler('M', ausfahrenMLR))
369 dispatcher.add_handler(CommandHandler('H', ausfahrenHLR))
370 dispatcher.add_handler(CommandHandler('initialisieren',
    initialisieren))
371
372 dispatcher.add_handler(CommandHandler('VRauf', VRauf))
373 dispatcher.add_handler(CommandHandler('VRzu', VRzu))
374 dispatcher.add_handler(CommandHandler('HRAuf', HRAuf))
375 dispatcher.add_handler(CommandHandler('HRzu', HRzu))
376 dispatcher.add_handler(CommandHandler('VLauf', VLauf))
377 dispatcher.add_handler(CommandHandler('VLzu', VLzu))
378 dispatcher.add_handler(CommandHandler('HLauf', HLauf))
379 dispatcher.add_handler(CommandHandler('HLzu', HLzu))
380 dispatcher.add_handler(CommandHandler('alleauf', alleauf))
381 dispatcher.add_handler(CommandHandler('allezu', allezu))
382 dispatcher.add_handler(CommandHandler('auf', alleauf))
383 dispatcher.add_handler(CommandHandler('zu', allezu))
384
385
386 #Ab Zeile 373: Konfiguration des individuellen Keyboards
387 from telegram import ReplyKeyboardMarkup
388 custom_keyboard=[[ 'Vorne', 'Schranken auf'],[ 'Mitte', 'Schranken
    zu'],[ 'Hinten', 'Status Sensoren']]
389 reply_markup=ReplyKeyboardMarkup(custom_keyboard)
390
391 updater.start_polling()

```

Listing 7.2: getriebemotor_multiprocessing.py

```

1  # -*- coding: utf-8 -*-
2  #python -m idlelib.idle <- Pythonshell muss so gestartet werden
   wenn man den Output lesen moechte.
3  #!/usr/bin/python
4  from Adafruit_MotorHAT import Adafruit_MotorHAT ,
   Adafruit_DCMotor
5  import os
6  from multiprocessing import Process
7  import time
8  import atexit
9  import RPi.GPIO as GPIO
10 GPIO.setmode(GPIO.BCM)
11 GPIO.setup(16,GPIO.IN) #HL
12 GPIO.setup(19,GPIO.IN) #VL
13 GPIO.setup(20,GPIO.IN) #HR
14 GPIO.setup(21,GPIO.IN) #VR
15
16 # create a default object, no changes to I2C address or
   frequency
17 mh = Adafruit_MotorHAT(addr=0x60)
18
19
20 # recommended for auto-disabling motors on shutdown!
21 def turnOffMotors():
22     mh.getMotor(1).run(Adafruit_MotorHAT.RELEASE)
23     mh.getMotor(2).run(Adafruit_MotorHAT.RELEASE)
24     mh.getMotor(3).run(Adafruit_MotorHAT.RELEASE)
25     mh.getMotor(4).run(Adafruit_MotorHAT.RELEASE)
26
27 atexit.register(turnOffMotors)
28
29 #myMotor=mh.getMotor(1)
30 #myMotor.run(Adafruit_MotorHAT.FORWARD)
31
32 def ausfahren(position,seite): #Positionen: H,V,M Seite: L, R,
   LR
33     print "Programm ausfahren() wurde gestartet"
34     #Positionserkennung
35     altepos='unbekannt' #unbekannt/V/H/M

```

```

36  if position=="init":
37      altepos="unbekannt"
38  elif seite=='L':
39      if GPIO.input(19)==1 and GPIO.input(16)==0:
40          altepos='V'
41      elif GPIO.input(19)==0 and GPIO.input(16)==1:
42          altepos='H'
43      elif GPIO.input(19)==1 and GPIO.input(16)==1:
44          altepos='M'
45  elif seite=='R':
46      if GPIO.input(21)==1 and GPIO.input(20)==0:
47          altepos='V'
48      elif GPIO.input(21)==0 and GPIO.input(20)==1:
49          altepos='H'
50      elif GPIO.input(21)==1 and GPIO.input(20)==1:
51          altepos='M'
52  elif seite=='LR':
53      if GPIO.input(21)==1 and GPIO.input(19)==1 and GPIO.
54         input(20)==0 and GPIO.input(16)==0:
55          altepos='V'
56      elif GPIO.input(21)==0 and GPIO.input(19)==0 and GPIO.
57         input(20)==1 and GPIO.input(16)==1:
58          altepos='H'
59      print "#16:", GPIO.input(16), "#19:", GPIO.input(19)
60         , "#20:", GPIO.input(20), "#21:", GPIO.input(21),
61         "->altepos=H (#54)"
62      elif GPIO.input(21)==1 and GPIO.input(19)==1 and GPIO.
63         input(20)==1 and GPIO.input(16)==1:
64          altepos='M'
65
66  if seite=='R' or seite=="L":
67      ausfahrsteuerung(position, seite, altepos, "eineSeite")
68  elif seite=='LR':
69      if position=="init":
70          print 'Initialisierung wird gestartet.'
71          pL=Process(target=ausfahrsteuerung, args=("V", "L",
72             altepos, "init"))
73          pR=Process(target=ausfahrsteuerung, args=("V", "R",
74             altepos, "init"))
75          pL.start()

```

```

69         pR.start()
70         pL.join()
71         pR.join()
72         pL=Process(target=ausfahrsteuerung, args=("H", "L",
73             altepos, "init"))
74         pR=Process(target=ausfahrsteuerung, args=("H", "R",
75             altepos, "init"))
76         pL.start()
77         pR.start()
78         pL.join()
79         pR.join()
80     else:
81         print 'Seite L und Seite R werden gestartet.'
82         pL=Process(target=ausfahrsteuerung, args=(position, "
83             L", altepos, "keine"))
84         pR=Process(target=ausfahrsteuerung, args=(position, "
85             R", altepos, "keine"))
86         pL.start()
87         pR.start()
88         pL.join()
89         pR.join()
90
91     else:
92         print 'FEHLER: Die angegebene Seite wurde nicht gefunden
93             !'
94
95 def ausfahrsteuerung(position, seite, altepos, anmerkung): #Als
96     Seite nur R und L zugelassen! #anmerkung: eineSeite, keine,
97     init
98     print "Ausfahrsteuerung gestartet (#39)"
99     test=Process(target=motorsteuerung, args=(position, seite,
100         altepos, anmerkung))
101     test.start()
102     #print test.is_alive()
103     if seite=='R':
104         Vpin=21
105         Hpin=20
106     elif seite=='L':
107         Vpin=19

```

```

101     Hpin=16
102 else:
103     print "FEHLER: Eine nichtausfuehrbare Seite wurde
          uebergeben (#44)"
104 if position=='V':
105     if altepos=="V":
106         print "Alte Position ist V, daher wird sofort
          abgebrochen (#86)"
107     else:
108         if altepos=="H":
109             time.sleep(4)
110         else:
111             time.sleep(0.2)
112         while True:
113             if GPIO.input(Hpin)==1:
114                 time.sleep(1)
115             if (GPIO.input(Vpin)==1 and GPIO.input(Hpin)==0)
          or test.is_alive()==False:
116                 print "Ausstieg beantragt (#41)"
117                 time.sleep(0.1)
118                 if (GPIO.input(Vpin)==1 and GPIO.input(Hpin)
          ==0) or test.is_alive()==False:
119                     print "Ausstieg gestattet (#43)!"
120                     break
121     print "test.is_alive()", test.is_alive()
122     print "Vpin", Vpin, GPIO.input(Vpin)
123     print "Hpin", Hpin, GPIO.input(Hpin)
124     test.terminate()
125     if seite=="R":
126         mh.getMotor(1).setSpeed(0)
127         mh.getMotor(2).setSpeed(0)
128     elif seite=="L":
129         mh.getMotor(3).setSpeed(0)
130         mh.getMotor(4).setSpeed(0)
131
132     print "Ausstieg erfolgreich!"
133 elif position=='H':
134     if altepos=="H":
135         print "Alte Position ist H, daher wird sofort
          abgebrochen (#110)"

```

```

136     else:
137         time.sleep(0.1)
138         while True:
139             if GPIO.input(Vpin)==1:
140                 time.sleep(1)
141             if (GPIO.input(Vpin)==0 and GPIO.input(Hpin)==1)
142                 or test.is_alive()== False:
143                 time.sleep(0.1)
144             if (GPIO.input(Vpin)==0 and GPIO.input(Hpin)
145                 ==1) or test.is_alive()== False:
146                 print "Ausstieg beantragt!"
147                 print "test.is_alive()", test.is_alive()
148                 print "Vpin", Vpin, GPIO.input(Vpin)
149                 print "Hpin", Hpin, GPIO.input(Hpin)
150                 break
151
152     test.terminate()
153     if seite=="R":
154         mh.getMotor(1).setSpeed(0)
155         mh.getMotor(2).setSpeed(0)
156     elif seite=="L":
157         mh.getMotor(3).setSpeed(0)
158         mh.getMotor(4).setSpeed(0)
159     print "Ausstieg erfolgreich!"
160
161 elif position=='M':
162     if altepos=="M":
163         print "Alte Position ist M, daher wird sofort
164             abgebrochen (#131)"
165     elif altepos=="H" or altepos=="V":
166         time.sleep(1)
167         while True:
168             if (GPIO.input(Vpin)==1 and GPIO.input(Hpin)==1)
169                 or test.is_alive()== False:
170                 print "Ausstieg beantragt! (#142)"
171                 time.sleep(0.2)
172                 break
173     else:
174         time.sleep(0.4)
175         while True:

```

```

172         if (GPIO.input(Vpin)==1 and GPIO.input(Hpin)==1)
173             or test.is_alive()== False:
174             time.sleep(0.1)
175             if (GPIO.input(Vpin)==1 and GPIO.input(Hpin)
176                 ==1) or test.is_alive()== False:
177                 print "Ausstieg beantragt! (#138)"
178                 break
179             print "test.is_alive()", test.is_alive()
180             print "Vpin", Vpin, GPIO.input(Vpin)
181             print "Hpin", Hpin, GPIO.input(Hpin)
182             test.terminate()
183             if seite=="R":
184                 mh.getMotor(1).setSpeed(0)
185                 mh.getMotor(2).setSpeed(0)
186             elif seite=="L":
187                 mh.getMotor(3).setSpeed(0)
188                 mh.getMotor(4).setSpeed(0)
189             print "Ausstieg erfolgreich!"
190         else:
191             print 'Position unbekannt'
192
193 def motorsteuerung(position, seite, altepos, anmerkung): #seite: L/
194     R (LR nicht erlaubt!) #anmerkung: eineSeite
195     print("Motorsteuerung() gestartet")
196     motorstartbereit=1
197     strecke='normal' #normal/lang
198
199     if seite=='R':
200         myMotor1=mh.getMotor(1)
201         myMotor2=mh.getMotor(2)
202
203     elif seite=='L':
204         myMotor1=mh.getMotor(3)
205         myMotor2=mh.getMotor(4)
206
207     else:
208         print 'FEHLER: Diese Seite ist nicht bekannt!'

```

```

209
210
211
212     if position=='V':
213         mh.getMotor(1).run(Adafruit_MotorHAT.FORWARD)
214         mh.getMotor(2).run(Adafruit_MotorHAT.BACKWARD)
215         mh.getMotor(3).run(Adafruit_MotorHAT.BACKWARD)
216         mh.getMotor(4).run(Adafruit_MotorHAT.FORWARD)
217         if altepos=='H':
218             strecke='lang'
219
220
221     elif position=='H':
222         mh.getMotor(1).run(Adafruit_MotorHAT.BACKWARD)
223         mh.getMotor(2).run(Adafruit_MotorHAT.FORWARD)
224         mh.getMotor(3).run(Adafruit_MotorHAT.FORWARD)
225         mh.getMotor(4).run(Adafruit_MotorHAT.BACKWARD)
226         if altepos=='V':
227             strecke='lang'
228
229     elif position=='M':
230         if seite=='R' or seite=='LR':
231             if GPIO.input(21)==1 and GPIO.input(20)==0:
232                 mh.getMotor(1).run(Adafruit_MotorHAT.BACKWARD)
233                 mh.getMotor(2).run(Adafruit_MotorHAT.FORWARD)
234
235             elif GPIO.input(21)==0 and GPIO.input(20)==1:
236                 mh.getMotor(1).run(Adafruit_MotorHAT.FORWARD)
237                 mh.getMotor(2).run(Adafruit_MotorHAT.BACKWARD)
238
239             elif GPIO.input(21)==1 and GPIO.input(20)==1:
240                 print "Bereits in Mittelposition"
241                 motorstartbereit=0
242
243             else:
244                 motorstartbereit=0
245                 print "Standort nicht bekannt, Geraet muss
246                     initialisiert werden!"
247         if seite=='L' or seite=='LR':

```

```

248     if GPIO.input(19)==1 and GPIO.input(16)==0:
249         mh.getMotor(4).run(Adafruit_MotorHAT.BACKWARD)
250         mh.getMotor(3).run(Adafruit_MotorHAT.FORWARD)
251
252     elif GPIO.input(19)==0 and GPIO.input(16)==1:
253         mh.getMotor(4).run(Adafruit_MotorHAT.FORWARD)
254         mh.getMotor(3).run(Adafruit_MotorHAT.BACKWARD)
255
256     elif GPIO.input(19)==1 and GPIO.input(16)==1:
257         print "Bereits in Mittelposition"
258         motorstartbereit=0
259
260     else:
261         motorstartbereit=0
262         print "Standort nicht bekannt, Geraet muss
                initialisiert werden!"
263
264
265 else:
266     print "FEHLER: Angegebene Richtung wurde nicht gefunden!"
267     "
268
269
270
271 #Ab Zeile 270: Manuelle Einstellung der Ausfahrzeiten
272 if motorstartbereit==1:
273     vmax=255
274     wartezeit=2.4 #Zeiten muessen fuer alle Moeglichkeiten
                individuell angepasst werden
275     if altepos=="unbekannt":
276         wartezeit=10 #Zeitdauer, in der mit voller
                Geschwindigkeit gefahren wird.
277     if seite=="R":
278         vmax=90
279     else:
280         vmax=140 #muss groesser als vmin sein
281
282     elif altepos=="H":
283         if position=="M":

```

```
284         if seite=="R":
285             vmax=235
286             wartezeit=2.9
287         elif seite=="L":
288             wartezeit=2.8
289     elif position=="V":
290         if seite=="R":
291             vmax=235
292             wartezeit=6.7
293         elif seite=="L":
294             wartezeit=6.6
295     elif altepos=="M":
296         if position=="H":
297             if seite=="R":
298                 vmax=235
299                 wartezeit=3.0
300             elif seite=="L":
301                 wartezeit=2.8
302         elif position=="V":
303             if seite=="R":
304                 vmax=235
305                 wartezeit=3.1
306             elif seite=="L":
307                 wartezeit=3.1
308     elif altepos=="V":
309         if position=="H":
310             if seite=="R":
311                 wartezeit=6.7
312                 vmax=240
313             elif seite=="L":
314                 wartezeit=6.6
315         elif position=="M":
316             if seite=="R":
317                 vmax=240
318                 wartezeit=3.0
319             elif seite=="L":
320                 wartezeit=3.1
321
322
323     if anmerkung==" eineSeite":
```

```

324     print("Wartezeit wird verkürzt da nur eine Seite
        verfahren wird.")
325     if seite=="R":
326         wartezeit=wartezeit-0.4
327         if (altepos=="V" and position=="H") or (altepos
            =="H" and position=="V"):
328             wartezeit=wartezeit-0.6
329     else:
330         wartezeit=wartezeit-0.3
331         if (altepos=="V" and position=="H") or (altepos
            =="H" and position=="V"):
332             wartezeit=wartezeit-0.5
333
334
335     print("\tSpeed up...")
336     for i in range(1,vmax,5):
337         #print i
338         myMotor1.setSpeed(i)
339         myMotor2.setSpeed(i)
340         time.sleep(0.01)
341     print("Beschleunigung fertig")
342     #Zeit verlaengern wenn von ganz hinten nach ganz vorne
343     #if strecke=='lang' or altepos=="unbekannt":
344         #time.sleep(5.8) #4.8 zu kurz
345
346     time.sleep(wartezeit)
347
348     print("\tSlow down...")
349     for i in range(vmax,50,-5):
350         #print i
351         myMotor1.setSpeed(i)
352         myMotor2.setSpeed(i)
353         time.sleep(0.01)
354
355     print("Bremsung fertig")
356     if altepos=="unbekannt":
357         time.sleep(50)
358     else:
359         time.sleep(5)
360     myMotor1.setSpeed(0)

```

```
361     myMotor2.setSpeed(0)
362     print "FEHLER: Endposition konnte nicht erreicht werden!"
        "
```