Diplomarbeit

Entwicklung einer benutzerfreundlichen Tennisballwurfmaschine

ausgeführt zum Zwecke der Erlangung des akademischen Grades eines

Diplom-Ingenieurs

unter der Leitung von

Ao.Univ.-Prof. Dipl.-Ing. Dr.techn. Manfred Grafinger
(E307-04 - Forschungsbereich Maschinenbauinformatik und Virtuelle Produktentwicklung)

eingereicht an der Technischen Universität Wien

Fakultät für Maschinenwesen und Betriebswissenschaften

Von

Andreas Eichberger
0909186 (066 482)
Lange Äcker 24
2120 Wolkersdorf im Weinviertel

Daniel Glatz
1126894 (066 482)
Hofstattgasse 22/7
1180 Wien

Wien, im November 2019

Andreas Eichberger _____ Daniel Glatz

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Ich habe zur Kenntnis genommen, dass ich zur Drucklegung meiner Arbeit unter der Bezeichnung

Diplomarbeit

nur mit Bewilligung der Prüfungskommission berechtigt bin.

Ich erkläre weiters Eides statt, dass ich meine Diplomarbeit nach den anerkannten Grundsätzen für wissenschaftliche Abhandlungen selbstständig ausgeführt habe und alle verwendeten Hilfsmittel, insbesondere die zugrunde gelegte Literatur, genannt habe.

Weiters erkläre ich, dass ich dieses Diplomarbeitsthema bisher weder im In- noch Ausland (einer Beurteilerin/einem Beurteiler zur Begutachtung) in irgendeiner Form als Prüfungsarbeit vorgelegt habe und dass diese Arbeit mit der vom Begutachter beurteilten Arbeit übereinstimmt.

Wien, im November 2019

_____

Andreas Eichberger

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Ich habe zur Kenntnis genommen, dass ich zur Drucklegung meiner Arbeit unter der Bezeichnung

Diplomarbeit

nur mit Bewilligung der Prüfungskommission berechtigt bin.

Ich erkläre weiters Eides statt, dass ich meine Diplomarbeit nach den anerkannten Grundsätzen für wissenschaftliche Abhandlungen selbstständig ausgeführt habe und alle verwendeten Hilfsmittel, insbesondere die zugrunde gelegte Literatur, genannt habe.

Weiters erkläre ich, dass ich dieses Diplomarbeitsthema bisher weder im In- noch Ausland (einer Beurteilerin/einem Beurteiler zur Begutachtung) in irgendeiner Form als Prüfungsarbeit vorgelegt habe und dass diese Arbeit mit der vom Begutachter beurteilten Arbeit übereinstimmt.

Wien, im November 2019

_____

Daniel Glatz

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Danksagung Eichberger

Zuerst möchte ich mich an dieser Stelle bei all jenen bedanken, die mich bei der Anfertigung dieser Diplomarbeit unterstützt haben.

Besonderer Dank gilt Dr. Manfred Grafinger für seine Betreuung während der Erstellung der Arbeit, meinem Freund und Studienkollegen Daniel Glatz für die reibungslose Zusammenarbeit, Johann Erhardt von der Firma EUROPTEN für seine unersetzliche Unterstützung in der Werkstatt sowie meiner Freundin Alexandra und meinen Eltern für ihre emotionale Unterstützung.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Danksagung Glatz

Zunächst möchte ich mich an dieser Stelle bei all denjenigen bedanken, die mich während der Anfertigung dieser Diplomarbeit unterstützt und motiviert haben.

Ganz besonders gilt dieser Dank Herrn Dr. Manfred Grafinger, der unsere Arbeit und somit auch mich betreut hat.

Daneben gilt mein Dank meiner Freundin Ruth Gremmel, welche in zahlreichen Stunden meinen Teil der Arbeit Korrektur gelesen hat. Zusätzlich konnte sie mich auch auf Schwächen hinweisen und als Fachfremde immer wieder zeigen, wo noch Erklärungsbedarf bestand.

Vor allem möchte ich mich bei dieser Gelegenheit bei meinem Freund und Studienkollegen Andreas Eichberger bedanken. Nur durch unsere erfolgreiche Zusammenarbeit war es überhaupt möglich, eine solche Arbeit zu realisieren.

Bedanken möchte mich auch bei meiner ganzen Familie für die motivierenden Worte und ihre große Unterstützung in den letzten Monaten und speziell bei dem Vater meiner Freundin, Franz Gremmel, für die praktische Unterstützung.

Weiters gilt der Dank auch der Fa. EUROPTEN für die finanzielle Unterstützung und Herrn Johann Erhardt für die technischen Hilfeleistungen bei der Fertigung des Funktionsmusters.

Nicht zuletzt gebührt meinen beiden Kindern Emma und Noah ein großer Dank, die es an zahlreichen Wochenenden akzeptierten, dass Papa jetzt keine Zeit zum Spielen hat, sondern an der Diplomarbeit arbeiten muss.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Kurzfassung

Tennisballwurfmaschinen sind schon seit den 1920er-Jahren am Markt erhältlich. Obwohl die Anzahl der Tennisspieler in Österreich kontinuierlich steigt, werden sie aber trotz zahlreicher Weiterentwicklungen kaum auf Tennisplätzen genutzt.

Ziel der vorliegenden Arbeit war es daher, mögliche Verbesserungen für Tennisballwurfmaschinen zu analysieren und technisch umzusetzen.

Dazu wurden in einem ersten Schritt bestehende Produkte analysiert und mit Kundenanforderungen verglichen, welche unter anderem durch Online-Umfragen ermittelt wurden. Als größte Probleme am Markt befindlicher Maschinen wurden mangelnde Bedienungsfreundlichkeit und schlechte Transportierbarkeit identifiziert.

Nach Festlegung einer verbesserten Lösungsvariante wurde diese durch ein Funktionsmuster erprobt. Dazu wurde eine funktionsfähige Maschine hergestellt und programmiert. Der Fokus lag dabei auf dem Erproben verbesserter und exakterer Bedienungsmöglichkeiten. So wurde unter anderem eine Simulation von Flugkurven unter Berücksichtigung strömungsmechanischer Effekte durchgeführt und vollständig programmiertechnisch implementiert und getestet. Dadurch soll ein intuitives und direktes Einstellen der Maschine sowie das einfache Anpassen auf individuelle Bedürfnisse durch den Benutzer ermöglicht werden.

Die mathematischen Berechnungen wurden mit MATLAB durchgeführt, während zur CAD-Konstruktion Inventor eingesetzt wurde. Die programmiertechnische Umsetzung erfolgte über Arduino- und Raspberry Pi- Boards.

Zuletzt werden die daraus gewonnenen Erkenntnisse aus mechanischer und elektronischer Sichtweise aufgezeigt und mögliche Verbesserungen vorgeschlagen.

Da die Arbeit von zwei Personen verfasst wurde, wurde diese nach Möglichkeit inhaltlich nach den Themen Produktentwicklung, Konstruktion und Fertigung sowie Programmierung, Elektronik und Versuche aufgeteilt. Der jeweilige Autor ist durch hochgestellte Zahlen im Inhaltsverzeichnis ersichtlich.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Abstract

Even though types of tennis ball machines are available since the 1920s and despite continuous development, they are seldom used on tennis courts today. At the same time though, the number of active tennis players in Austria is rising every year.

Therefore, the goal of this thesis was to analyze and implement possible improvements of tennis ball machines.

To do this, existing products were analyzed and compared to customer requirements, which were determined by online-surveys and interviews. Poor usability and insufficient mobility were identified as key problems of today's machines.

After defining an improved concept, it was tested with the help of a functional model, which was fully built and programmed. The goal of this was the testing of a user friendly and more exact method of operation of the machine. To do this, ball trajectories were simulated with regards to fluid mechanical phenomena and then implemented and tested. Thereby, intuitive and direct operation of the machine as well as easy individualization to the user's needs were made possible.

Mathematical calculations were first implemented in MATLAB, whereas CAD-modeling was done in Inventor. For programming, Arduino- and Raspberry Pi-boards were used.

At last, knowledge gained as well as possible improvements of the machine are documented.

Since the thesis was written by two authors, it was separated in terms of content in the parts of product development, design engineering and manufacturing as well as programming, electronics and testing. Authorship of the respective parts is highlighted by superscript numbers in the table of contents.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Inhaltsverzeichnis

[1] … Diplomarbeitsteil Andreas Eichberger
[2] … Diplomarbeitsteil Daniel Glatz

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

[1] … Diplomarbeitsteil Andreas Eichberger
[2] … Diplomarbeitsteil Daniel Glatz

1 Einleitung

Obwohl Tennisballwurfmaschinen bereits in 1920er-Jahren durch René Lacoste erfunden und seither nach zahlreichen Weiterentwicklungen durch unzählige Firmen weltweit vertrieben wurden, sind sie auf den wenigsten Tennisplätzen anzutreffen. Auch wenn sie von Privatpersonen teuer selbst angeschafft werden, fristen sie nach der anfänglichen Euphorie meist bald ein Dasein als Staubfänger.

Dies mag verwundern angesichts der Stundenpreise, die viele Anfänger und Hobbyspieler, neben nicht zu vernachlässigenden Platzmieten und Mitgliedsgebühren, jährlich bereitwillig an professionelle Trainer zahlen. Dabei ist zum motorischen Lernen beim Tennis, wie bei den meisten „technischen“ Sportarten, nach dem Beherrschen der Grundtechnik vor allem Repetition essenziell.

Ballwurfmaschinen mögen wenig mit dem Spielen gegen menschliche Gegner gemein haben, ermöglichen es aber, eine ausgewählte Bewegung hundert- oder tausendfach exakt gleich zu üben. Das gleichmäßige Zuspielen von Bällen auf diese Weise ist auch für Trainer eine Herausforderung und für Anfänger normalerweise gänzlich unmöglich. Beim Einsatz von Ballwurfmaschinen entfällt weiters die Notwendigkeit, Termine mit Mitspielern vereinbaren zu müssen.

Dabei ist zu bemerken, dass sie Einheiten mit menschlichen Trainern keineswegs vollständig ersetzen können. Vielmehr sollten nach dem Erlernen oder Umlernen einer Bewegung eine große Anzahl an Wiederholungen mit Ballwurfmaschinen geübt werden, bevor wieder eine Kontrolle und gegebenenfalls Änderung durch den Trainer erfolgt.

Aufbauend auf dem technischen und tennisspezifischen Fachwissen der Autoren soll in der vorliegenden Arbeit nun versucht werden, die Gründe für die geringe Nutzung von Ballwurfmaschinen zu analysieren und darauf aufbauend technische Verbesserungen vorzuschlagen und umzusetzen.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

2 Methodik

Die hier verfolgte Vorgehensweise orientiert sich im Wesentlichen an der Konstruktionsmethodik nach Pahl/Beitz (2007) und den darauf aufbauenden Werken von Pohn/Lindemann (2008) und Naefe (2018).

Die Basis dafür bildet die branchenunabhängige VDI-Richtlinie 2221, welche ein sequenzielles Vorgehensmodell zum Entwickeln und Konstruieren technischer Produkte vorschlägt (Abbildung 1). Dabei gehen aus jedem Schritt Arbeitsdokumente, beispielsweise Anforderungsliste oder prinzipielle Lösungsfestlegung, hervor. Gleichzeitig wird auch die Notwendigkeit von Iterationen betont. So soll der Ablauf nicht als starr angesehen werden, sondern ein häufiges Über- oder Zurückspringen zulässig sein (vgl. Pahl/Beitz 2007: 21; Pohn/Lindemann 2008: 15).

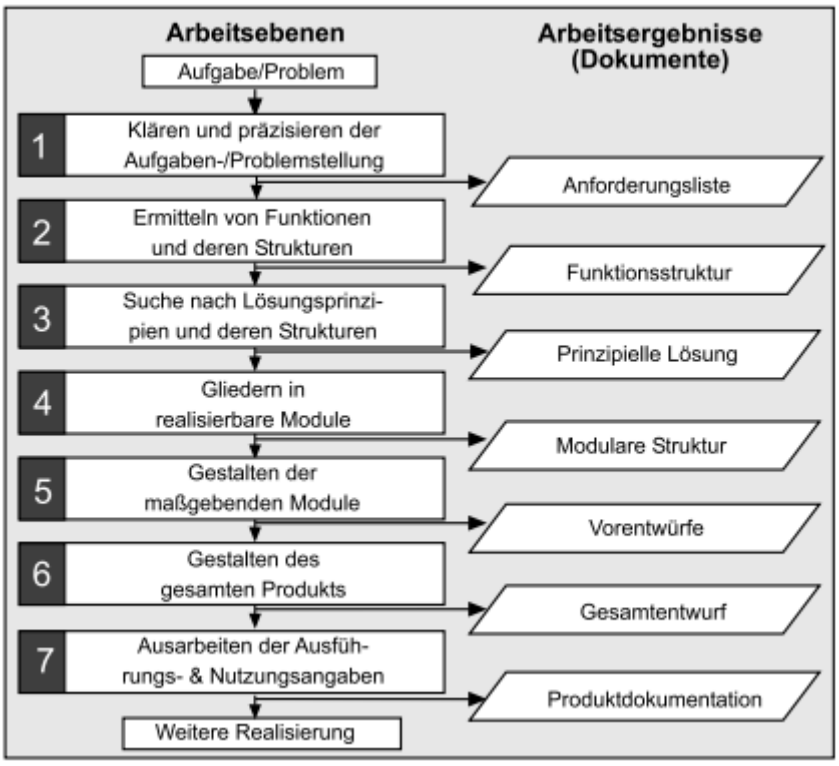


Abbildung 1: Vorgehensmodell nach VDI-Richtlinie 2221 (Pohn/Lindemann 2008: 15)

Aufbauend auf dieser Richtlinie wird von Pahl/Beitz (2007: 193-194) ein auf den Einsatz im Maschinenbau angepasster, vierstufiger Arbeitsfluss vorgeschlagen:

Die erste Phase stellt das *Planen und Klären der Aufgabenstellung* dar. Ziel ist dabei die Informationsbeschaffung über Anforderungen und Rahmenbedingungen. Das Ergebnis ist die Anforderungsliste. Darauf folgt mit dem *Konzipieren* jene Phase des Konstruierens, welche bereits die prinzipielle Festlegung einer Lösung anstrebt. Dazu müssen Wirk- und Lösungsprinzipien für das vorliegende Problem analysiert und bewertet werden. Anschließend kann mit der Phase des *Entwerfens* begonnen

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

werden, an dessen Ende eine gestalterische Festlegung der Lösung steht. Auch hier folgt eine technisch-wirtschaftliche Bewertung, da in vielen Fällen mehrere Entwürfe neben- oder hintereinander angefertigt werden müssen. Ist schließlich der endgültige Gesamtentwurf gefunden, so kann mit der *Ausarbeitung* begonnen werden. Das Ergebnis dieser Phase ist eine herstellungstechnische Festlegung, welche alle benötigten Details wie Form, Bemaßung, Werkstoffe, Oberflächenbeschaffenheit etc. umfasst (vgl. Pahl/Beitz 2007: 195-197). Der Anteil der vier Phasen am Gesamtaufwand der Konstruktion ist in Abbildung 2 qualitativ dargestellt.

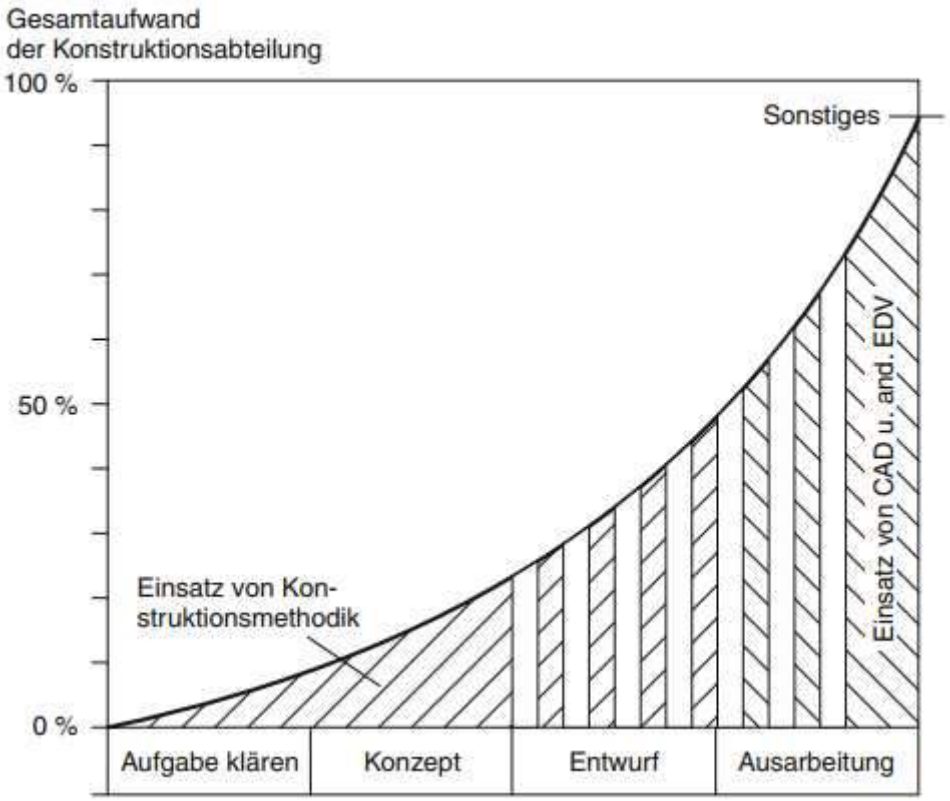


Abbildung 2: Arbeitsaufwand in der Konstruktion (Naefe 2018: 42)

Einen wesentlichen zeitlichen Anteil an der vorliegenden Arbeit stellte die Herstellung eines Funktionsmusters dar. Modelle und Prototypen werden von Pahl/Beitz (2007: 197-199) nicht explizit einer Phase zugeordnet, sondern sollen als Mittel der Informationsgewinnung je nach Erfordernis eingesetzt werden.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

3 Klären und Präzisieren der Aufgabenstellung

Der erste Teil dieses Kapitels umfasst die Klärung der Rahmenbedingungen sowie die Vorkalkulationen, die zur Entwicklung des Produkts benötigt werden. Durch das Finden von Kundenanforderungen soll schließlich eine vollständige Anforderungsliste erstellt werden.

3.1 Grundlagen und Definitionen

Spielregeln

Ziel des Spiels Tennis ist das Zurückschlagen eines Balles über ein fest gespanntes Netz mithilfe eines Schlägers. Wird der Ball nicht in das durch Linien begrenzte Spielfeld zurückgespielt, bevor er nach dem ersten Bodenkontakt im Feld erneut den Boden berührt, ist der Punkt verloren. Dabei ist jedoch auch das Schlagen in der Luft erlaubt, was als Volley bezeichnet wird. Wird der Ball erst nach dem Bodenkontakt gespielt, spricht man von Vor- und Rückhand. Die Spieler dürfen sich frei inner- und außerhalb des Spielfeldes auf ihrer Seite des Tennisplatzes bewegen (vgl. ITF 2019: 7-10).

Um strömungsmechanische Effekte auf den Ball auszunutzen, werden die meisten Bälle mit Vorwärtsrotation gespielt, was als Topspin bezeichnet wird. Alternativ kann auch Rückwärtsrotation (Backspin) oder seltener Seitrotation (Sidespin) angewendet werden. Ballwurfmaschinen dienen dem repetitiven Training von Schlagbewegungen. In dieser Funktion können sie einen menschlichen Trainer entlasten bzw. ersetzen.

Spezifikationen

Die durch die ITF (International Tennis Federation) festgelegten Hauptmaße eines Tennisplatzes sind in Abbildung 3 dargestellt. Das Netz hat eine Mindesthöhe von 0,91m in der Mitte und eine Maximalhöhe von 1,07m an den Netzpfosten (vgl. ITF 2019: 2).

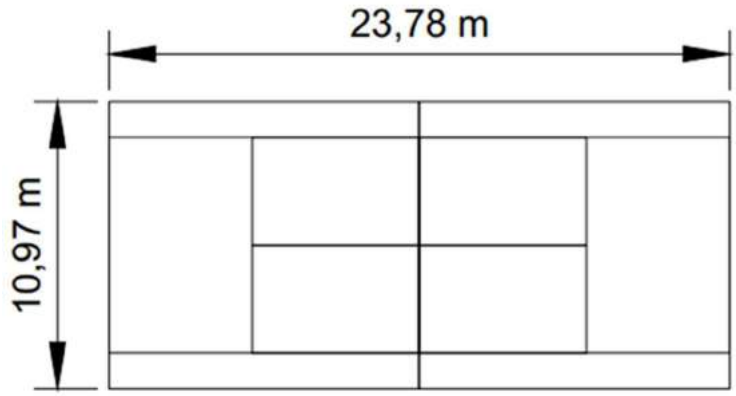


Abbildung 3: Dimensionen eines Tennisplatzes (ITF 2017: 40-41)

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die wichtigsten von der ITF für den Spielbetrieb zugelassene Oberflächen sind Sand, Beton, Gras, Teppich und Granulat. Sie werden anhand des Court Pace Ratings (CPR) nach slow bis fast kategorisiert (vgl. ITF 2017: 45). Sand ist im europäischen Raum die bevorzugte Oberflächenart auf Freiluftplätzen.

Die Spezifikationen von Standard-Tennisbällen nach Typ 1 (von der ITF für den Einsatz bis zu einer Meereshöhe von 1219m empfohlen) sowie von speziellen Kinder-Tennisbällen (Stage 1-3, geeignet für unterschiedliche Altersklassen) sind in Tabelle 1 aufgelistet:

	Typ 1	Stage 3 (Red)	Stage 2 (Orange)	Stage 1 (Green)
Masse (g)	56,0 – 59,4	36,0 – 49,0	36,0 – 46,9	47,0 – 51,5
Durchmesser (cm)	6,54 – 6,86	7,00 – 8,00	6,00 – 6,86	6,30 – 6,86
Rücksprunghöhe bei Fallhöhe von 254 ± 0,3 cm (cm)	138 – 151	90 – 105	105 – 120	120 – 135

Tabelle 1: Spezifikationen von Bällen nach ITF-Regeln (ITF 2017: 4-6)

Tennisbälle bestehen üblicherweise im Kern aus einer ca. 40g schweren Gummikugel, welche mit einer Filzbeschichtung überzogen wird. Die gebräuchlichste Ausführung sind Innendruckbälle, welche beim Herstellungsprozess mit erhöhtem Druck befüllt und luftdicht in einer Dose verpackt werden. Bedingt durch Druckverlust unterliegt ihr Spielverhalten damit aber einer zeitlichen Änderung. Aus diesem Grund werden auch sogenannte drucklose Bälle hergestellt, welche die geforderte Sprunghöhe durch eine größere Wandstärke ohne erhöhten Innendruck erreichen und so auch nach längerem Gebrauch annähernd konstante Spieleigenschaften aufweisen (vgl. Cross / Lindsey 2005: 87-88).

Damit eignen sich drucklose Bälle besser zum Einsatz in Trainingseinheiten, insbesondere mit Ballmaschinen. Ein entscheidender Nachteil ist jedoch ein oft als unangenehm empfundenes, „härteres“ Spielverhalten. Daher werden von Hobby- und Profispielern gleichermaßen mehrheitlich Innendruckbälle genützt. Eine übliche Praxis im Amateurbereich besteht darin, gebrauchte Innendruckbälle zu sammeln und für einige Zeit in Trainingseinheiten weiterzuverwenden, bevor sie ausgetauscht werden. Für den Einsatz in Ballmaschinen müssen also beide Arten von Bällen gleichermaßen berücksichtigt werden. Tennisbälle für Kinder weisen stark abweichende Spezifikationen auf (siehe Tabelle 1).

Je nach Marke und Modell werden bei Tennisbällen geringfügig unterschiedliche Filzbeschichtungen verwendet, welche das Flugverhalten beeinflussen. Weiters kann oftmaliges Spielen einerseits dazu führen, dass die Oberfläche unvollständig abgelöst und dadurch flaumiger wird, was den Ball im Flug bremst. Andererseits können sich

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

nach längerer Benutzung aber auch Teile der Filzbeschichtung von der Oberfläche abtrennen und so den Luftwiderstand senken (vgl. Cross / Lindsey 2005: 88).

Abbildung 4 vergleicht die Oberfläche eines benützten mit der eines neuen Tennisballs:

Abbildung 4: Vergleich der Oberflächen eines benützten (links) und eines neuen Tennisballes (rechts)

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

3.2 Flugverhalten von Tennisbällen

Zur Berechnung der Flugbahn eines Tennisballs muss bei rotierenden Körpern neben dem Luftwiderstand auch die Drehzahl und -richtung (im Weiteren zusammenfassend als „Spin“ bezeichnet) und deren strömungsmechanische Auswirkungen auf die Flugbahn berücksichtigt werden.

Zuerst soll eine einfache Abschätzung unter Vernachlässigung der oben genannten Effekte durchgeführt werden. Anschließend werden diese über experimentell ermittelte Werte in die Berechnung miteinbezogen und die Resultate verglichen.

Berechnung bei Vernachlässigung von Luftwiderstand und Spin

Für ein schief geworfenes Objekt unter Einfluss der Fallbeschleunigung g gelten im zweidimensionalen Raum bei Vernachlässigung von Luftwiderstand und Spin folgende Gleichungen in horizontaler x- bzw. vertikaler y-Richtung (vgl. Dreyer et al. 2012: 29-31):

$$x(t) = v_0 t \cos(\beta_0) \tag{3.1}$$

$$y(t) = v_0 t \sin(\beta_0) - \frac{g t^2}{2} \tag{3.2}$$

Damit kann nach Elimination von t die Flugbahn als Funktion von x berechnet werden:

$$y(x) = x \tan(\beta_0) - \frac{g}{2 v_o^2 \cos^2 \beta_0} x^2 \tag{3.3}$$

g … Fallbeschleunigung (in negativer y-Richtung)
v_0 … Anfangsgeschwindigkeit
β_0 … Abschusswinkel

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Berechnung bei Berücksichtigung von Luftwiderstand und Spin

Berücksichtigt man Luftwiderstand und Spin, erhält man nach Cross / Lindsey (2013: 6) folgendes System von gewöhnlichen Differentialgleichungen:

$$m\frac{dv_x}{dt} = -F_D\cos(\beta) - F_L\sin(\beta) \tag{3.4}$$

$$m\frac{dv_y}{dt} = -F_D\sin(\beta) + F_L\cos(\beta) - mg \tag{3.5}$$

Die Kräfte F_D und F_L lassen sich mit den dimensionslosen Konstanten C_D und C_L ausdrücken durch:

$$F_D = \frac{1}{2}C_D\rho A v^2 \tag{3.6}$$

$$F_L = \frac{1}{2}C_L\rho A v^2 \tag{3.7}$$

A … Querschnittsfläche des Balles
ρ … Dichte der Umgebungsluft
v … Momentangeschwindigkeit des Balles
C_D … dimensionslose Luftwiderstands-Konstante
C_L … dimensionslose Spin-Konstante

F_D bezeichnet jene Kraft, die dem Ball durch den Luftwiderstand entgegenwirkt. Sie wirkt immer entgegen der momentanen Flugrichtung. F_L entspricht der Kraft, die den Ball bei Rotation abhängig von der Drehrichtung zusätzlich hebt oder senkt. Sie wirkt immer normal zur momentanen Flugrichtung und zur Rotationsachse. Der dafür verantwortliche physikalische Effekt wird als Magnus-Effekt bezeichnet, benannt nach dem deutschen Physiker Heinrich Gustav Magnus. Beim Flug eines rotierenden Körpers bilden sich Verwirbelungen an dessen Oberfläche, welche sich auf einer Seite entgegen, auf der anderen Seite in Richtung der entgegenkommenden Luft bewegen. Dadurch entsteht an diesen Stellen ein höherer bzw. niedrigerer Druck. Aus dieser Druckdifferenz resultiert eine Kraft, welche die erwähnte Richtungsänderung hervorruft (vgl. Alam et al. 2008: 1).

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Die auf den rotierenden Ball in seiner Flugbahn wirkenden Kräfte werden in Abbildung 5 dargestellt:

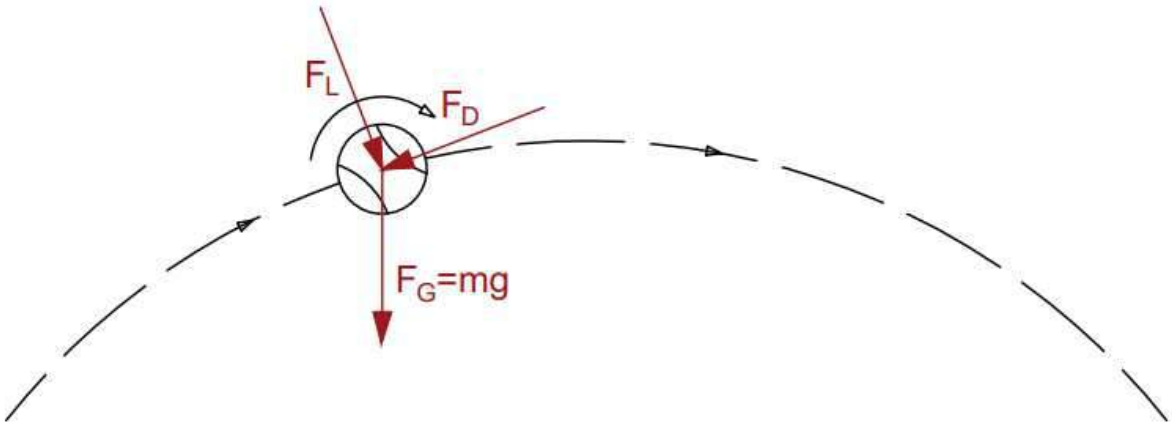


Abbildung 5: Auf einen rotierenden Ball im Flug wirkende Kräfte

Nach Einführung einer Hilfsgröße

$$k = \frac{A \cdot \rho}{2 \cdot m} \tag{3.8}$$

kann das Gleichungssystem vereinfacht werden zu (Cross / Lindsey 2013: 6):

$$\frac{d^2x}{dt^2} = kv\left(-C_D\frac{dx}{dt} - C_L\frac{dy}{dt}\right) \tag{3.9}$$

$$\frac{d^2y}{dt^2} = kv\left(-C_D\frac{dy}{dt} + C_L\frac{dx}{dt}\right) - g \tag{3.10}$$

Die von Cross / Lindsey (2013: 10) experimentell ermittelten Werte für C_D und C_L sind in Abbildung 6 über dem sogenannten Spin-Verhältnis S aufgetragen. Dieses errechnet sich durch:

$$S = \frac{\frac{d}{2} \cdot \omega}{v_0} \tag{3.11}$$

Zur Vereinfachung wird die Winkelgeschwindigkeit ω dabei im Folgenden über die gesamte Flugkurve als konstant angenommen.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

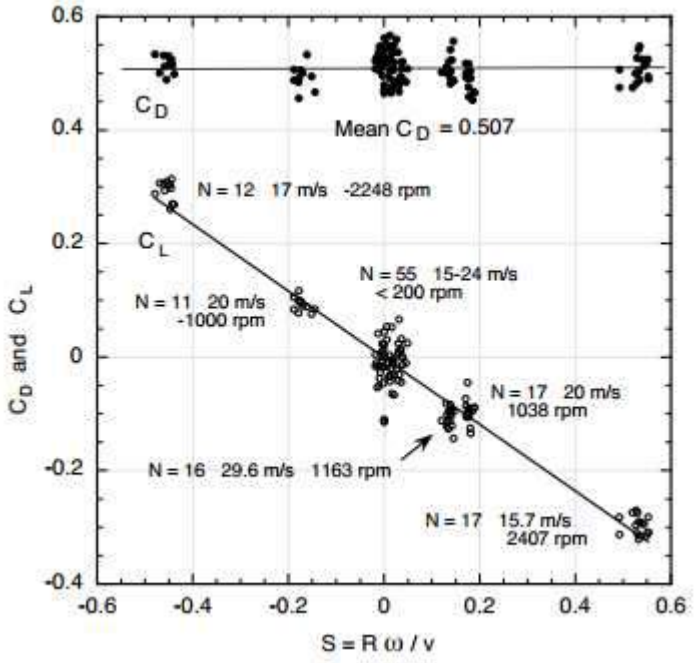


Abbildung 6: Für C_D und C_L experimentell ermittelte Werte (Cross / Lindsey 2013: 10)

Zu erkennen ist, dass der Beiwert für den Luftwiderstand C_D im betrachteten Rahmen praktisch unabhängig vom Spin ist. Dadurch wird er in der weiteren Rechnung als konstant angenommen. Der Mittelwert wird von Cross / Lindsey (2013: 11) als 0,507 angegeben. Die Standardabweichung beträgt 0,024.

Der Beiwert für den Spin C_L sinkt annähernd linear mit steigender Drehzahl. Aus dem Diagramm lässt sich grafisch folgende Beziehung herleiten:

$$C_L \approx -0{,}6 \cdot S \tag{3.12}$$

Damit lässt sich zeigen, dass positiver Spin ($\omega > 0$, „Topspin“) zu einer Kraft in negativer y-Richtung führt. Umgekehrt bewirkt negativer Spin ($\omega < 0$, „Backspin“) eine Kraft in positiver y-Richtung.

Diese experimentelle Ermittlung erfolgte mit Videoaufnahmen und Messungen tatsächlicher Flugkurven unter bekannten Anfangsbedingungen. Die Werte decken sich mit Studien mit vergleichbarem Versuchsaufbau. Windtunnelexperimente liefern teils deutlich höhere Werte für C_D und C_L (vgl. Alam et al. 2008: 6-7). Der Unterschied lässt sich möglicherweise darauf zurückführen, dass sich die strömende Luft in Windtunneln vor dem Experiment stabilisieren kann, während der Ball bei der Messung von Flugkurven gegen annähernd ruhende Luft beschleunigt wird (vgl. Cross / Lindsey 2013: 2-3). Die rechnerische Ermittlung von C_D und C_L mittels CFD (Computational Fluid Dynamics) erweist sich als schwierig, da ein vereinfachtes CAD-Modell eines Tennisballs hier unzureichende Ergebnisse liefert. Um die Genauigkeit zu verbessern, müsste die Filzbeschichtung exakt modelliert werden (vgl. Alam et al. 2008: 7).

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Das Differentialgleichungssystem (Gleichungen 3.9 und 3.10) lässt sich für $v_x \neq 0$ und $v_y \neq 0$ nur numerisch lösen (vgl. Cross / Lindsey 2013: 6). Da es sich um ein Anfangswertproblem handelt, wird zuerst das explizite Euler-Verfahren in MATLAB implementiert. Die Ergebnisse werden in Tabelle 2 mit dem in MATLAB integrierten ODE45-Solver verglichen, welcher ein Runge-Kutta-Verfahren verwendet (vgl. MathWorks 2019).

Lösung mit explizitem Euler-Verfahren

Das explizite Euler-Verfahren ist ein simples Ein-Schritt-Verfahren zur numerischen Lösung gewöhnlicher Differentialgleichungen mit zugehöriger Anfangsbedingung:

$$y'(x) = f(x, y) \quad , \quad y(x_0) = y_0 \tag{3.13}$$

Gesucht wird die Lösung $y(x)$, welche sowohl die Differentialgleichung als auch die Anfangsbedingung erfüllt.

Dazu wird die Funktion stückweise linear durch Tangenten angenähert. Die Formel lautet:

$$y_{k+1} = y_k + h \cdot f(x_k, y_k) \tag{3.14}$$

Der Fehler ist abhängig von der gewählten Schrittweite h (vgl. Huckle/Schneider 2006: 279-285).

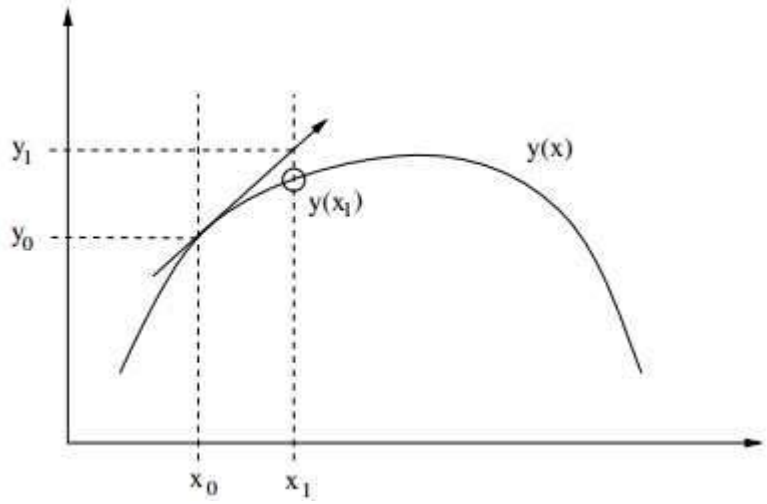


Abbildung 7: Explizites Euler-Verfahren (Huckle/Schneider 2006: 285)

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Im Folgenden wird das Verfahren in MATLAB für das System gewöhnlicher Differentialgleichungen implementiert. Dabei soll mit x die horizontale, mit y die vertikale Komponente der Flugbahn des Balls bezeichnet werden. Die Variable t beschreibt die Zeit.

Beim Aufrufen der Lösungsfunktion

```
function [t,z] = Flugbahn_Euler(m,d,v_0,beta,h_0,x_0,C_D,n)
```

müssen die folgenden Eingangsgrößen definiert werden:

```
m                                % (kg) Masse des Balls
d                                % (m) Durchmesser des Balls
v_0                              % (km/h) Anfangsgeschwindigkeit des Balls
beta                             % (°) Abschusswinkel
h_0                              % (m) Abschusshöhe
x_0                              % (m) Abschusspunkt
C_D                              % (1) Luftwiderstands-Wert
n                                % (1/min) Drehzahl des Balls
```

Folgende Konstanten werden für die weitere Rechnung definiert:

```
delta_t=0.001;                  % (s) Schrittweite

rho=1.2;                        % (kg/m^3) Dichte der Luft
g=9.81;                          % (m/s^2) Fallbeschleunigung

A=(d/2)^2*pi;                    % (m^2) Querschnittsfläche des Balls
k=A*rho/(2*m);                   % (1/m) Hilfswert

omega=2*pi*n/60;                % (1/s) Drehzahl des Balls
C_L=-0.6*d/2*omega/(v_0/3.6)    % (1) Lift-Wert
```

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Zur Lösung werden die Matrix $z = [x, \dot{x}, y, \dot{y}]$ (Lage- und Geschwindigkeitskoordinaten) sowie die Spaltenvektoren t (Zeit) und v (Momentangeschwindigkeit) eingesetzt. Deren Anfangswerte werden im nächsten Schritt definiert:

```
t(1,1)=0;

z(1,1)=-x_0;
z(1,2)=(v_0/3.6)*cos(beta*pi/180);
z(1,3)=h_0;
z(1,4)=(v_0/3.6)*sin(beta*pi/180);

v(1,1)=sqrt(z(1,2)^2+z(1,4)^2);
```

Mithilfe der Notation

```
z(i,j)
```

wird dabei auf das Element in der Zeile i und der Spalte j der Matrix z zugegriffen.

Nun kann das explizite Euler-Verfahren angewendet werden:

```
i=1;                              % Zähler für Anzahl an Iterationen
while 1

    t(i+1,1)=t(i,1)+delta_t;

    z(i+1,1)=z(i,1)+delta_t*z(i,2);
    z(i+1,2)=z(i,2)+delta_t*(k*v(i,1)*(-C_D*z(i,2)-C_L*z(i,4)));
    z(i+1,3)=z(i,3)+delta_t*z(i,4);
    z(i+1,4)=z(i,4)+delta_t*(k*v(i,1)*(-C_D*z(i,4)+C_L*z(i,2))-g);

    v(i+1,1)=sqrt(z(i+1,2)^2+z(i+1,4)^2);

    if z(end,3)<0                 % Ausstieg aus der Schleife bei Aufprall
        break
    end
    i=i+1;
end
```

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Lösung mit *ode45*-Solver

```
function [t,z] = Flugbahn_ode45(m,d,v_0,beta,h_0,x_0,C_D,n)
```

Eingangsgrößen und Konstanten entsprechen jenen der bereits gezeigten Lösung. Die Anfangswerte der Größen $x, \dot{x}, y, \dot{y}$ müssen jedoch als Vektor $\boldsymbol{z_0}$ gespeichert werden:

```
z0=[
   -x_0
   (v_0/3.6)*cos(beta*pi/180)
   h_0
   (v_0/3.6)*sin(beta*pi/180)
   ];
```

Die Lösung erfolgt über die in MATLAB integrierte Funktion *ode45*:

```
options = odeset('Events',@efun);            % für Terminierung

[t,z]=ode45(@odefun,sim_time,z0,options);    % Vektor z = (x,x°,y,y°)
    function dzdt=odefun(t,z)                % Vektor dzdt = (x°,x°°,y°,y°°)
        v=sqrt(z(2)^2+z(4)^2);               % Momentangeschwindigkeit
        dzdt=[
            z(2) ;
            k*v*(-C_D*z(2)-C_L*z(4)) ;
            z(4) ;
            k*v*(-C_D*z(4)+C_L*z(2))-g
            ];
    end
    function [value,isterminal,direction] = efun(t,z)
                                        % terminiert, wenn die Nullstelle
                                        % (=Aufprall) erreicht ist
        value=z(3);
        isterminal=1;
        direction=0;
    end
```

Um die Rechnung beim Nulldurchgang, welcher dem Aufprall des Balles entspricht, zu terminieren, wird die Funktion *odeset* eingesetzt.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Vergleich der Lösungen

Die Anfangswerte für die Rechnung sollen nun betragen:

$d = 0{,}065m$
$m = 0{,}055kg$
$v_0 = 80\frac{km}{h}$
$\beta = 10°$
$x_0 = 0m$
$h_0 = 0{,}3m$
$n = 1000\frac{U}{min}$
$\rho = 1{,}2\frac{kg}{m^3}$
$g = 9{,}81\frac{m}{s^2}$
$C_D = 0.507$

Die Ergebnisse werden in Tabelle 2 dargestellt. Die relative Abweichung wird in Relation zur Lösung des ode45-Solvers errechnet:

$$a_{rel} = \frac{|L - L_{ode45}|}{|L_{ode45}|} \tag{3.15}$$

Verfahren	Schrittweite (s)	Anzahl benötigter Schritte	errechneter Aufprall (m)	relative Abweichung (%)
Euler (explizit)	0,01	80	15,922	1,111
	0,005	159	15,869	0,775
	0,001	788	15,771	0,152
ode45	variabel	36	15,747	-

Tabelle 2: Vergleich von iterativen Lösungen

Für die folgenden Darstellungen wird das Euler-Verfahren mit Schrittweite 0,001s zur Lösung des Gleichungssystems eingesetzt.

In Abbildung 8 wird die simulierte Flugbahn eines Balles mit jener bei Vernachlässigung des Luftwiderstandes ($C_D = 0$) verglichen:

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

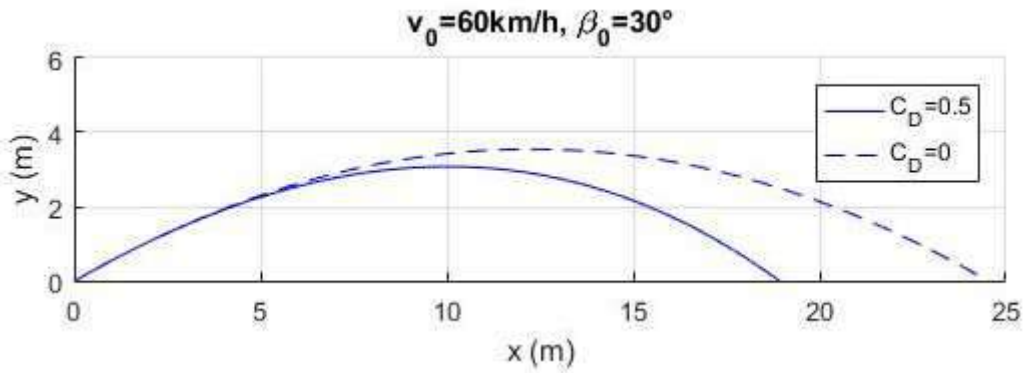


Abbildung 8: Vergleich von simulierten Flugbahnen mit Luftwiderstand (durchgängige Linie) und ohne Luftwiderstand

Abbildung 9 vergleicht Flugbahnen verschiedener Drehzahlen bei ansonsten gleichen Anfangsbedingungen:

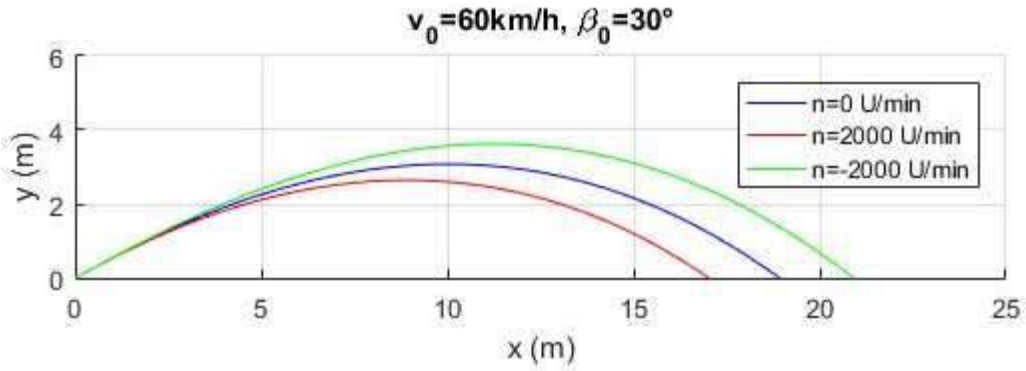


Abbildung 9: rotationsfreier Ball (blau), Topspin (rot), Backspin (grün)

Cross / Lindsey (2013: 11-14) geben allerdings zu bedenken, dass die ermittelten Werte für C_D und C_L selbst zwischen augenscheinlich identischen Bällen variieren können. Insbesondere die Variation von C_L bei geringen Drehzahlen ist hoch, was darauf schließen lässt, dass noch weitere Kräfte auf einen rotierenden Ball wirken.

Bei Zeitlupenaufnahmen lässt sich erkennen, dass eine asymmetrische Abnützung der Balloberfläche oftmals unerwartete Auswirkungen auf C_D und C_L haben kann. Es entstehen dadurch zusätzliche Seitenkräfte, die bei ausreichend schneller Rotation allerdings so oszillieren, dass die Effekte kaum merkbar sind. Hohe Drehzahlen haben also eine stabilisierende Wirkung. Im Gegensatz dazu können rotationsfreie Bälle in ihrer Flugbahn mehrmals merkbar die Richtung ändern. Dieses Phänomen ist zum Beispiel bei Fußball und Baseball als „Flatterball“ bzw. „Knuckleball“ bekannt. Bei diesen Sportarten verfügen die Bälle zwar über eine annähernd glatte Oberfläche, die Nähte bedingen aber bereits ausreichend große Asymmetrien für das Auftreten der genannten Effekte (Cross / Lindsey 2013: 14).

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Die von Cross / Lindsey (2013: 14-15) ermittelten Werte von C_D schwankten zwischen 0,45 und 0,57, jene von C_L in Abhängigkeit des Spin-Verhältnisses um etwa ±0,05. In Abbildung 10 sollen die extremsten Auswirkungen dieser Variationen verdeutlicht werden:

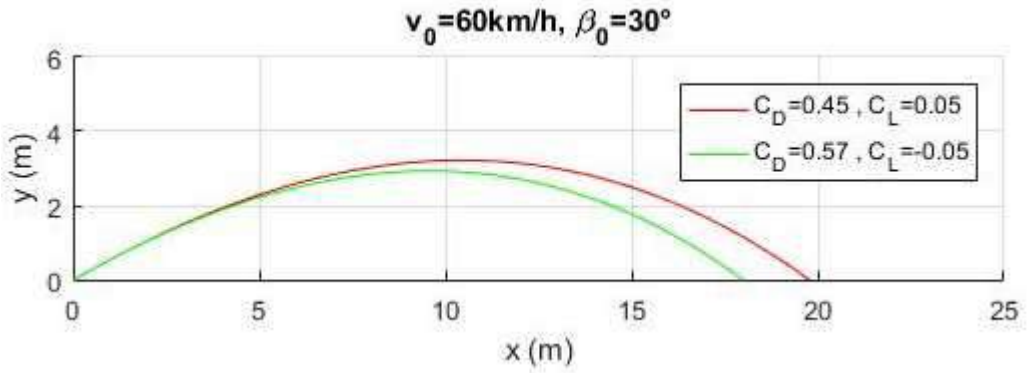


Abbildung 10: Vergleich zweier Flugbahnen mit extremster Variation von C_D und C_L

Die Simulation wird nun einem Vergleich mit einer realen Flugbahn unterzogen. Dazu wird ein Ball mit einem Tennisschläger annähernd rotationsfrei in einer Ebene exakt rechtwinkelig zur Kamerarichtung geschossen und gefilmt. Das Video wird mit der Videoanalyse-Software *Tracker* analysiert (siehe Abbildung 11).

Die verwendete Aufnahme weist 30 Bilder pro Sekunde auf, damit beträgt der Abstand zwischen zwei Messpunkten 0,033s. Zur korrekten Skalierung des Koordinatensystems wird ein exakt in der Ball-Ebene befindlicher, in seiner Länge bekannter Abstand zwischen zwei Bodenmarkierungen herangezogen.

Abbildung 11: Analyse einer Flugkurve in der Software „Tracker“: Messpunkte (hellblau), Koordinatenachsen (pink), Kalibrierungslänge (grün)

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Um Ungenauigkeiten beim Auswählen der Messpunkte auszugleichen, kann die Flugbahn als kubische Polynomfunktion aus den ermittelten Punkten angenähert werden (siehe Abbildung 12). Dafür wird die Methode der kleinsten Quadrate eingesetzt.

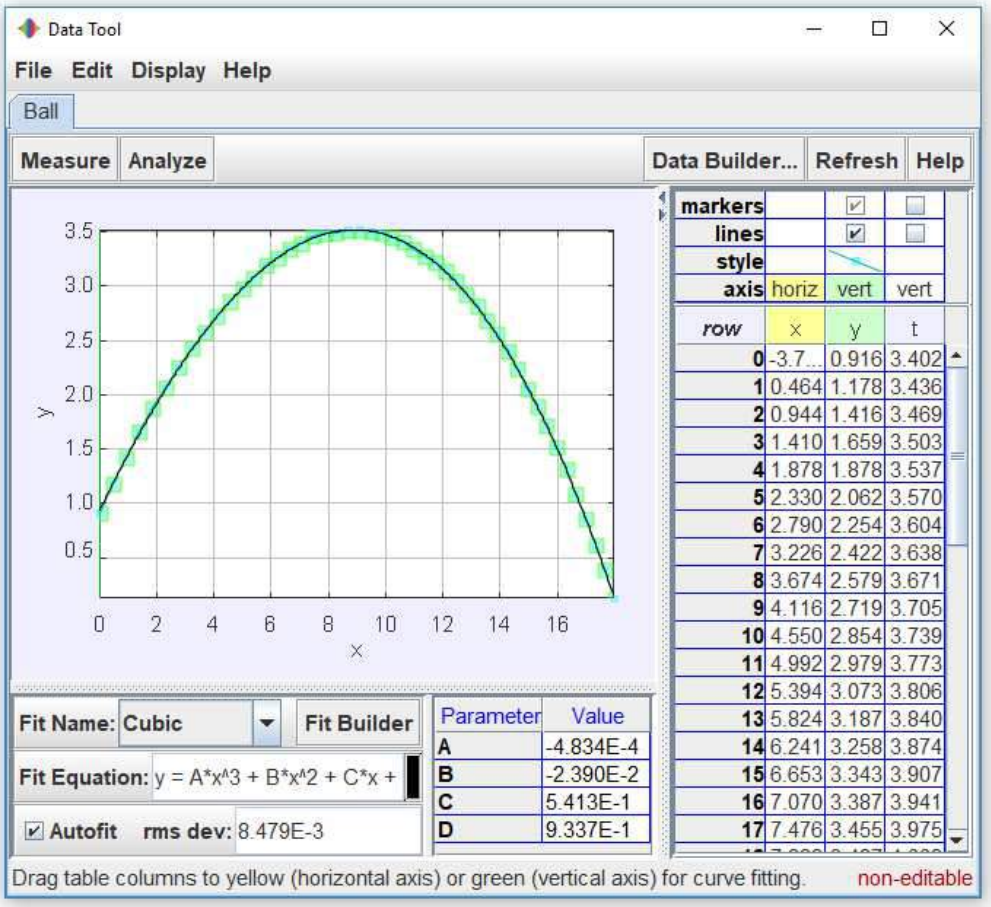


Abbildung 12: Messpunkte (grün), angepasste Polynomfunktion (schwarz)

Die Flugbahn wird nun mit den aus dem Video ermittelten Anfangswerten v_0 und β_0 simuliert und mit der Polynomfunktion der gemessenen Flugbahn verglichen (siehe Abbildung 13). Dazu wird für den Luftwiderstand der von Cross / Lindsey (2013: 11) angegebene Mittelwert $C_D = 0{,}507$ verwendet.

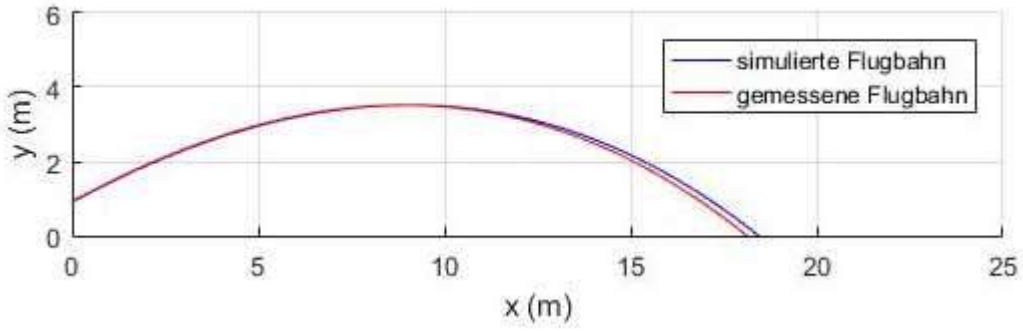


Abbildung 13: Vergleich von simulierter und gemessener Flugbahn

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

3.3 Kundenanforderungen

Unabhängig davon, ob ein Produkt für anonyme Kunden oder als Bestellung entwickelt wird, ist es sinnvoll, von Beginn an alle bekannten Anforderungen in einem zentralen Dokument zu sammeln. Die Anforderungsliste ist dann zugleich Ausgangsposition und aktuelle Arbeitsunterlage, sie soll also immer auf dem neuesten Stand gehalten und angepasst werden. Es wird empfohlen, nach Forderungen, die zwingend erfüllt werden müssen, und Wünschen, die gegebenenfalls einen Mehraufwand rechtfertigen, zu unterscheiden. Ihre Ausprägungen sollten idealerweise mit quantitativen Werten oder Wertbereichen festgelegt werden. Wo dies nicht möglich ist, sollte eine verbale, lösungsneutrale Beschreibung erfolgen (vgl. Pahl/Beitz 2007: 213-215).

Um keine Anforderungen zu vergessen, muss eine Vielzahl möglicher Quellen berücksichtigt werden (Abbildung 14). Eine Möglichkeit dazu besteht in der Verwendung einer Checkliste aus Hauptmerkmalen (Abbildung 15). Weitere nützliche Werkzeuge werden in der QFD (Quality Function Deployment) - Methode vorgeschlagen (vgl. Pahl/Beitz 2007: 213-220). Wenngleich auf die sehr umfangreiche Durchführung eines vollständigen QFD hier verzichtet wird, sollen dessen Konzepte zum Auffinden von Kundenanforderungen eingesetzt und diese erst später mit Hilfe der Hauptmerkmalliste ergänzt werden.

Im QFD wird nach externen Sekundäranforderungen (z.B. Analyse von Konkurrenzprodukten oder Fachzeitschriften), internen Sekundäranforderungen (z.B. Informationen durch Kundendienst oder Reklamationen) und Primäranforderungen (z.B. Kundenbefragungen oder -beobachtungen) unterschieden (vgl. Schloske 2017: 13).

Abbildung 14: Mögliche Quellen für Anforderungen (Ponn/Lindemann 2008: 37)

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Hauptmerkmal	Beispiele
Geometrie	Größe, Höhe, Breite, Länge, Durchmesser, Raumbedarf, Anzahl, Anordnung, Anschluss, Ausbau und Erweiterung
Kinematik	Bewegungsart, Bewegungsrichtung, Geschwindigkeit, Beschleunigung
Kräfte	Kraftgröße, Kraftrichtung, Krafthäufigkeit, Gewicht, Last, Verformung, Steifigkeit, Federeigenschaften, Stabilität, Resonanzen
Energie	Leistung, Wirkungsgrad, Verlust, Reibung, Ventilation, Zustandsgrößen wie Druck, Temperatur, Feuchtigkeit, Erwärmung, Kühlung, Anschlussenergie, Speicherung, Arbeitsaufnahme, Energieumformung
Stoff	Physikalische und chemische Eigenschaften des Eingangs- und Ausgangsprodukts, Hilfsstoffe, vorgeschriebene Werkstoffe (Nahrungsmittelgesetz u. ä.), Materialfluss und -transport
Signal	Eingangs- und Ausgangssignale, Anzeigeart, Betriebs- und Überwachungsgeräte, Signalform
Sicherheit	Unmittelbare Sicherheitstechnik, Schutzsysteme, Betriebs-, Arbeits- und Umweltsicherheit
Ergonomie	Mensch-Maschine-Beziehung: Bedienung, Bedienungsart, Übersichtlichkeit, Beleuchtung, Formgestaltung
Fertigung	Einschränkung durch Produktionsstätte, größte herstellbare Abmessung, bevorzugtes Fertigungsverfahren, Fertigungsmittel, mögliche Qualität und Toleranzen
Kontrolle	Mess- und Prüfmöglichkeit, besondere Vorschriften (TÜV, ASME, DIN, ISO, AD-Merkblätter)
Montage	Besondere Montagevorschriften, Zusammenbau, Einbau, Baustellenmontage, Fundamentierung
Transport	Begrenzung durch Hebezeuge, Bahnprofil, Transportwege nach Größe und Gewicht, Versandart und -bedingungen
Gebrauch	Geräuscharmut, Verschleißrate, Anwendung und Absatzgebiet, Einsatzort (z. B. schwefelige Atmosphäre, Tropen,...)
Instandhaltung	Wartungsfreiheit bzw. Anzahl und Zeitbedarf der Wartung, Inspektion, Austausch und Instandsetzung, Anstrich, Säuberung
Recycling	Wiederverwendung, Wiederverwertung, Entsorgung, Endlagerung, Beseitigung
Kosten	Max. zulässige Herstellkosten, Werkzeugkosten, Investition und Amortisation
Termin	Ende der Entwicklung, Netzplan für Zwischenschritte, Lieferzeit

Abbildung 15: Hauptmerkmalliste (Pahl/Beitz 2007: 220)

Das Kano-Modell, benannt nach dem japanischen Professor und Unternehmensberater Nariaki Kano, unterscheidet Kundenanforderungen in drei Kategorien (Abbildung 16). Die Basisfaktoren sind von großer Wichtigkeit, aber oft so selbstverständlich, dass sie nicht mehr explizit ausgesprochen werden und sich so einer Erfassung entziehen können. Werden sie ignoriert, kann das jedoch sehr negative Konsequenzen auf die Kundenzufriedenheit nach sich ziehen. Leistungsfaktoren bezeichnen jene Anforderungen, die vom Kunden klar formuliert und erwartet werden. Eine Erfüllung wird mit höherer Kundenzufriedenheit belohnt.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Begeisterungsfaktoren sind schwierig zu erfassen, aber bei der Produktentwicklung von großer Wichtigkeit, um sich von der Masse abzuheben. Im Folgenden sollen Basis- und Leistungsfaktoren durch die Analyse von Konkurrenzprodukten sowie durch Kundenbefragungen so vollständig wie möglich ermittelt werden. Um mögliche Begeisterungsfaktoren auffinden zu können, könnten zu einem späteren Zeitpunkt Anwender beobachtet und zu ihren Erfahrungen befragt werden (vgl. Saatweber 2011: 85-94).

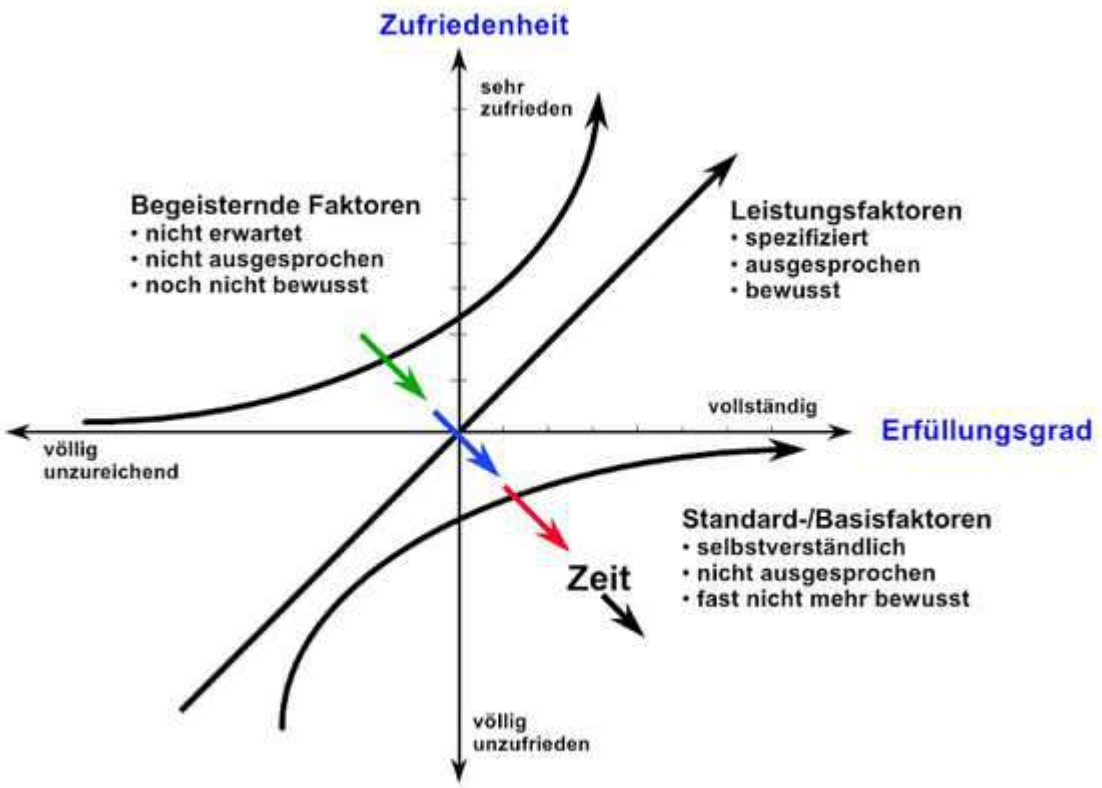


Abbildung 16: Kano-Modell (Saatweber 2011: 86)

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

3.3.1 Analyse von Konkurrenzprodukten

In Tabelle 3 werden ausgewählte am Markt erhältliche Ballmaschinen verglichen:

	Baseliner Slam 24	Spinshot Lite	Lobster Elite Freedom	Sports Tutor ProLite	Lobster Elite Liberty	Spinshot Pro	Lobster Elite One	Spinshot Plus	Spinshot Player	Lobster Elite Grand Five LE
ungefährer Grundpreis (€)	400	600	1200	1300	1350	1400	1550	1700	1900	3700
max. Geschw. (km/h)	65	50	105	96	130	120	130	120	120	130
min. Intervall (s)	8	2	2	2	2	2	2	2	2	2
Ball-Kapazität	24	50	150	125	150	120	150	120	120	150
Topspin	ja (fix)	ja (fix)	nein	nein	ja	ja	ja	ja	ja	ja
Backspin	nein	nein	nein	nein	ja	ja	ja	ja	ja	ja
elektr. vertikale Verstellung	nein	nein	nein	nein	nein	ja	ja	ja	nein	ja
horizontale Oszillation	nein	nein	ja	ja	ja	ja	ja	ja	ja	ja
vertikale Oszillation	nein	nein	nein	nein	nein	nein	nein	ja	ja	ja
programmierbare Drills	nein	nein	nein	nein	nein	nein	nein	nein	ja	ja
Stromversorgung	Kabel	Akku	Akku	Akku	Akku	Akku	Akku	Akku	Akku	Akku
Spieldauer lt. Hersteller (h)	-	3-4	2-4	2-3	2-4	2-3	4-8	2-3	2-3	4-8
Gewicht (kg)	8	10	16	14	16	17	19	21	19	20
Fernbedienung	nein	optional	optional	nein	optional	ja	optional	ja	ja	ja
Smartphone-Bedienung	nein	nein	nein	nein	nein	nein	nein	ja	ja	ja

Tabelle 3: Vergleich von Konkurrenzprodukten

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

3.3.2 Kundenbefragungen

Schriftliche Kundenbefragungen sind die am weitesten verbreiteten Methoden zur Informationsgewinnung im Marketing. Das Ziel ist, Kundenerwartungen sowie deren Bedeutung zu ermitteln. Dabei können sowohl quantitative als auch qualitative Sachverhalte gewonnen werden. Zum Auffinden von Begeisterungsfaktoren soll auch die Möglichkeit angeboten werden, freie Kommentare abzugeben (vgl. Saatweber 2011: 117-118). Neben persönlichen Gesprächen mit potenziellen Kunden wurde eine schriftliche Befragung mit 26 Teilnehmern durchgeführt. Die Resultate sollen im Folgenden grafisch dargestellt werden:

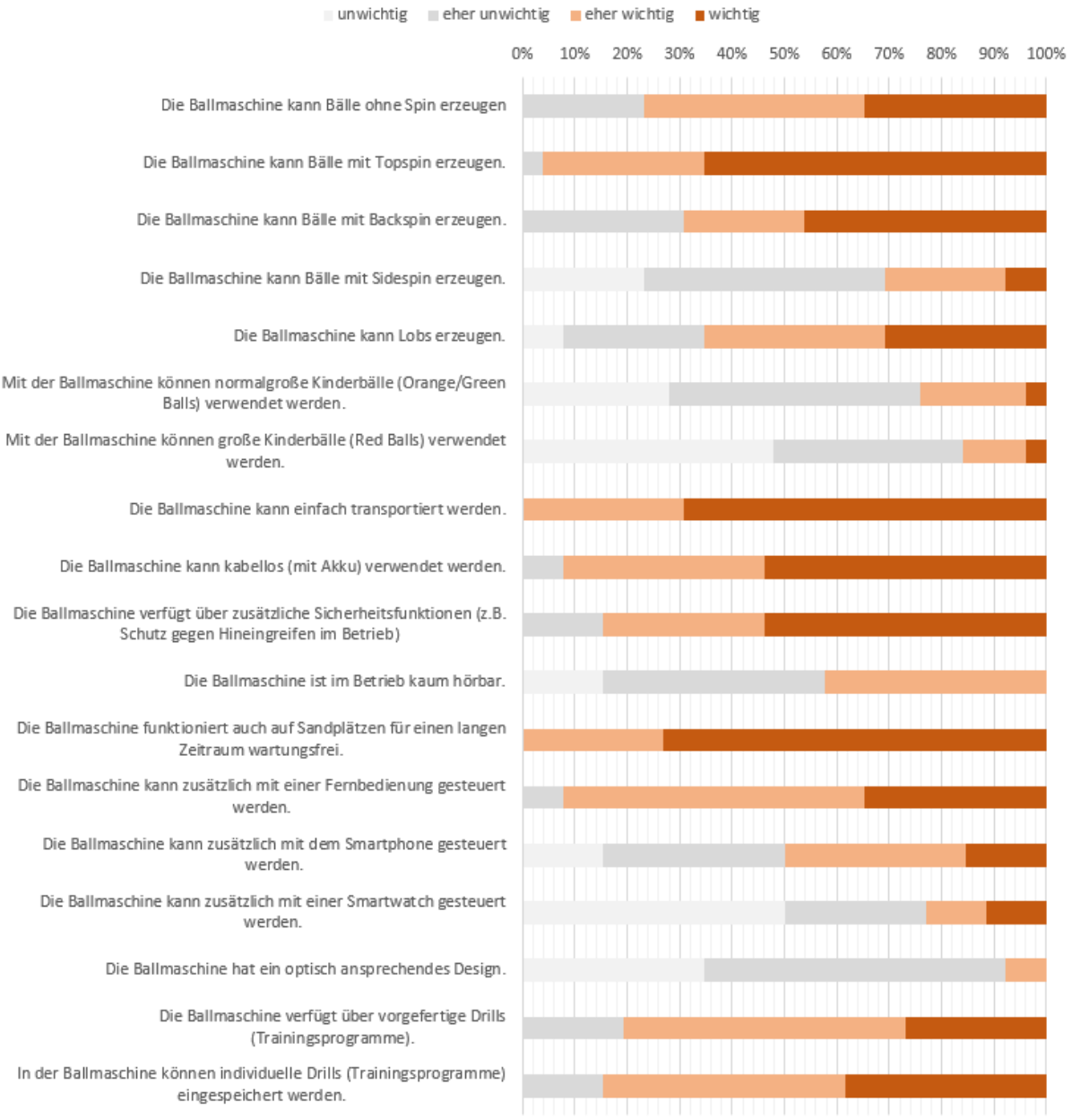


Abbildung 17: Auswertung der Kundenbefragung, Teil 1

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

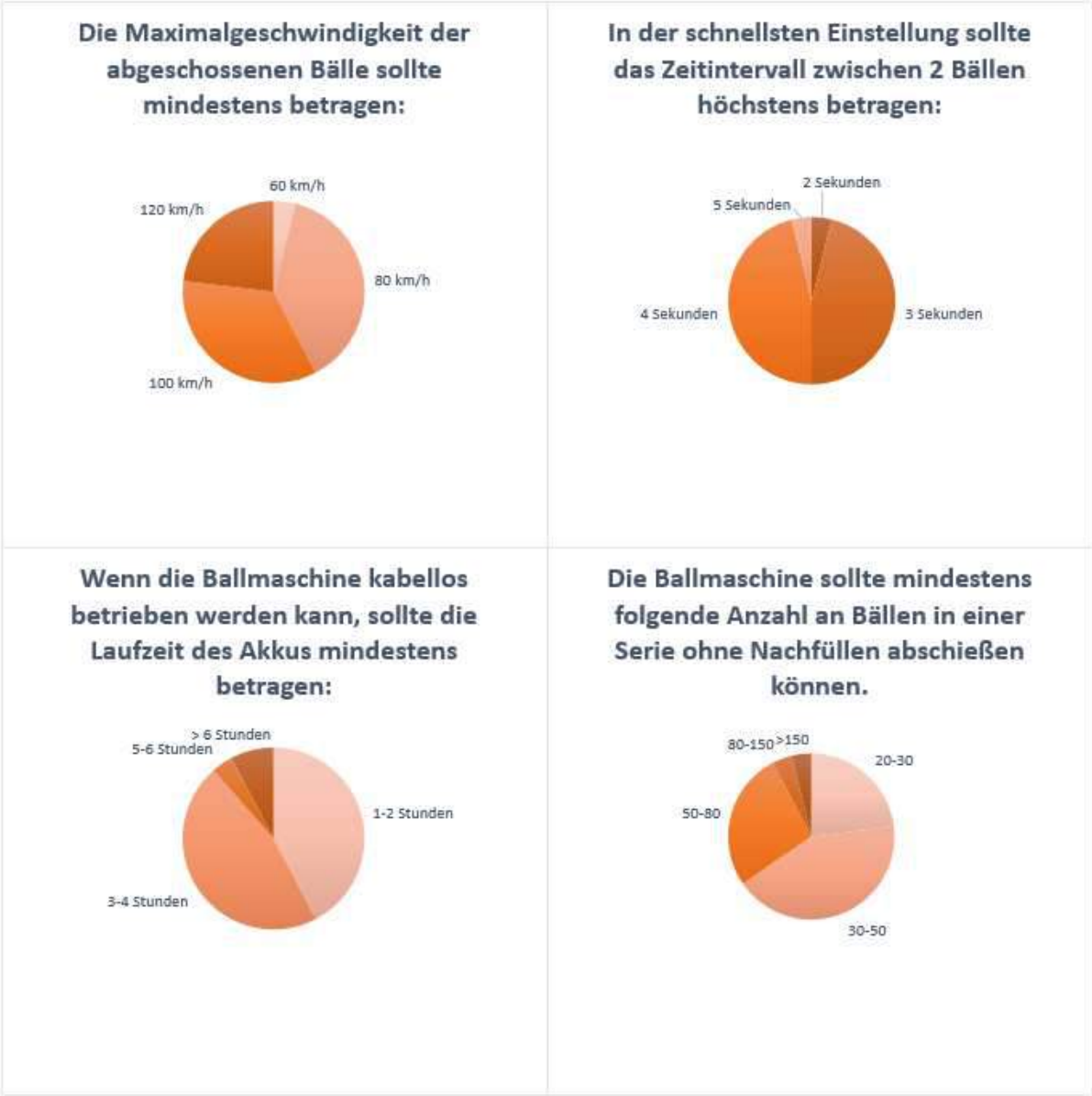


Abbildung 18: Auswertung der Kundenbefragung, Teil 2

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Zusammen mit persönlichen Interviews werden schließlich folgende Hauptprobleme aktuell erhältlicher Modelle identifiziert:

Bedienungsmöglichkeiten: Mit Ausnahme von High-End-Modellen erfolgt die Bedienung fast ausschließlich durch das manuelle Einstellen einzelner Parameter wie Winkel, Geschwindigkeit und Spin (siehe Abbildung 19). Damit muss erst durch mehrere „Test-Schüsse“ das gewünschte Resultat gefunden werden. Individualisierte Drills werden ebenso nur von wenigen Maschinen angeboten.

Abbildung 19: Kontroll-Panel einer Lobster Elite Three (Lobster Sports 2019)

Mobilität Fast alle erhältlichen Modelle sind auf Rädern transportierbar, lassen sich aber kaum ergonomisch über Treppen, aus Kofferräumen etc. befördern. Zum einfacheren Transport kann bei vielen Maschinen der Ballbehälter umgeklappt werden (siehe Abbildung 20), allerdings müssen die Bälle dann separat zum Spielort gebracht werden. Das Gewicht der meisten verglichenen Modelle bewegt sich zwar nur zwischen 14-21 kg, jedoch können sie normalerweise nicht einhändig getragen werden.

Abbildung 20: Lobster Elite Three, zum Transport vorbereitet (Lobster Sports 2019)

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Wiederbefüllen der Maschine: Da auf den meisten Anlagen keine Hilfen zum Einsammeln von Bällen vorhanden sind, müssen diese in der Regel händisch vom Boden aufgehoben und zur Ballwurfmaschine getragen werden. Dies kann bei intensiven Trainingseinheiten eine zusätzliche Belastung darstellen.

Entleerung des Ballbehälters: Die meisten Modelle verwenden eine vertikal gelagerte, rotierende Scheibe mit mehreren Durchlässen zum definierten Abgeben der Bälle. Dabei kann es passieren, dass Bälle hängenbleiben und unerwartet verzögert oder gar nicht abgeschossen werden und im Ballbehälter zurückbleiben.

Zusammenfassung

Folgende Anforderungen an das Produkt sollen sich in der Anforderungsliste wiederspiegeln:

Die freie Variation des Spins ist bei den meisten erhältlichen Ballwurfmaschinen möglich und kann damit neben Zuverlässigkeit als Basisanforderung angesehen werden.

Von großer Wichtigkeit für die Benutzer sind Mobilität sowie Bedienungsfreundlichkeit und -umfang. Diese sind ebenso wie Anschaffungskosten und Ballkapazität zu den Leistungsanforderungen zu zählen. Ebenso könnte die Genauigkeit der Maschine im Verhältnis zu bereits erhältlichen Modellen erhöht werden.

Als Begeisterungsanforderungen könnten zusätzliche Sicherheitsfunktionen angesehen werden, welche bei Konkurrenzprodukten kaum erhältlich sind. Außerdem könnten hier verbesserte Lösungen für das Einsammeln und das vollständige Entleeren der Bälle im Behälter gefunden werden.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

3.3.3 Anforderungsliste

ANFORDERUNGSLISTE			
F (Forderung) W (Wunsch)	**Nr.**	**Bezeichnung, Merkmal**	**Werte, Daten, Erläuterungen**
	1	**Termine & Kosten**	
W	1.1	Fertigstellung Funktionsmuster	1.9.2019
F	1.2	Herstellkosten Funktionsmuster	max. 1500 €
F	1.3	angestrebte Stückzahl	Kleinserie (max. 50 Stk.)
W	1.4	angestrebter Verkaufspreis	max. 800-1000 €
F	1.5	Absatzmarkt	Hobbyspieler, Trainer
	2	**Geometrie & Gewicht**	
F	2.1	Raumbedarf (Lagerung/Transport)	Kleinwagen-Kofferraum (ca.100x50x50cm)
F	2.2	Gewicht	max. 15 kg
	3	**Gebrauch**	
F	3.1	Einsatzort	Sandplatz, Indoor-Court
W	3.2	Stromversorgung	Batterie, min. 1-2h Laufzeit
F	3.3	max. Ballgeschwindigkeit	min. 80 km/h
F	3.4	Spin	Topspin und Backspin
W	3.5	max. Spin	min. 1000 U/min
F	3.6	Mindestintervall zw. zwei Bällen	max. 3-4 s
F	3.7	Ball-Kapazität	min. 20 Bälle
W	3.8	Abschusshöhe	realem Treffpunkt nachempfunden
F	3.9	Genauigkeit	max. Abweichung eines identischen Balles: ± 0,5m
F	3.10	Bedienung	einfache Eingabe der Zielparameter

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

	4	**Ergonomie**	
F	4.1	Bedienelemente	gute Zugänglichkeit zu Bedienelementen
W			drahtlose Bedienung möglich
F	4.2	Transportierbarkeit	mit PKW einfach zum und vom Tennisplatz zu transportieren
W	4.3	Wiederbefüllung mit Bällen	ergonomisches Wiederbefüllen ohne Bücken
	5	**Sicherheit**	
F	5.1	Schutzmaßnahmen	Absicherung rotierender Teile
F	5.2	Sicherheitshinweise	in Bedienungsanleitung
	6	**Instandhaltung**	
F	6.3	Säuberung	einfache Säuberung durch Benutzer möglich
F	6.4	Wartungsintervall	min. 10.000 Betriebsstunden
F	6.5	Wartungsort	Hersteller
F	6.6	Wartungspersonal	Hersteller
	7	**Recycling & Entsorgung**	
F	7.1	Nutzungsdauer	min. 20.000 Betriebsstunden
F	7.2	Entsorgung/Recycling	Entsorgungsanleitung in Bedienungsanleitung
F	7.3	Schadstoff- und Lärmemission	keine Schadstoffemission
W			Geräuschentwicklung entsprechend normalem Lärm auf Tennisplätzen
W	7.4	Ressourcenverbrauch (Hilfsstoffe)	kein Nachschmieren durch Benutzer erforderlich

Tabelle 4: Anforderungsliste

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

4 Konzipieren

Bevor mit der Formulierung von Lösungen begonnen wird, ist es zweckmäßig, sich gedanklich von bereits bekannten Prinzipien zu trennen. Durch eine solche Abstraktion können selbst bei Anpassungskonstruktionen oder solchen, wo bereits ähnliche Produkte existieren, oft neue und bessere Lösungsfelder erschlossen werden. Ein Hilfsmittel dazu bietet die Erstellung von Funktionsmodellen bzw. -strukturen. Das gebräuchliche umsatzorientierte Funktionsmodell fokussiert sich dabei auf die Analyse von Stoff-, Energie- oder Signalumsätzen im betrachteten System. Die Darstellung kann dann in Listen-, hierarchischer oder netzwerkartiger Struktur erfolgen. Es ist im Allgemeinen sinnvoll, ausgehend von Gesamtfunktionen hin zu Teilfunktionen nach Bedarf zu konkretisieren. Teil-Lösungen für die gefundenen Funktionen lassen sich später im sogenannten morphologischen Kasten mit diesen in Matrixform zusammenführen und daraus mögliche Gesamt-Lösungsvarianten ableiten (vgl. Ponn/Lindemann 2008: 53-61; Naefe 2018: 89-119). Im morphologischen Kasten (Tabelle 5) sind die gewählten Lösungsvarianten bereits durch farbige Linien gekennzeichnet.

4.1 Funktionsstruktur

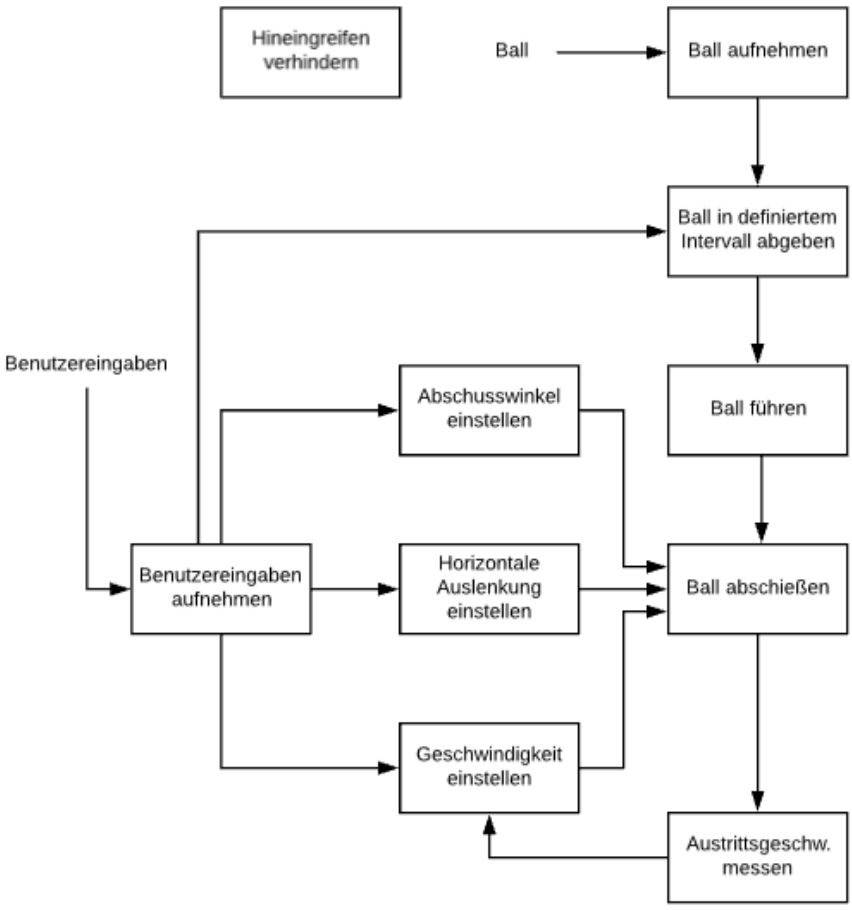
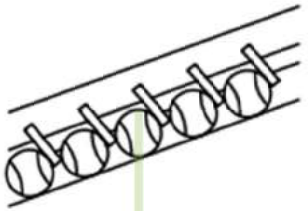
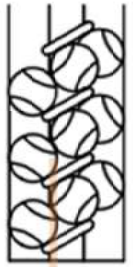
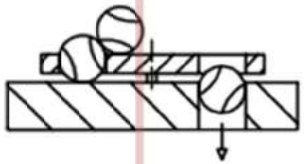
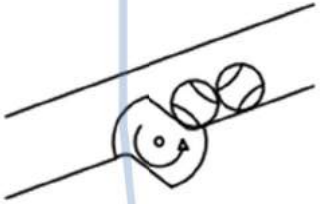
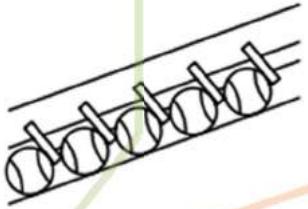
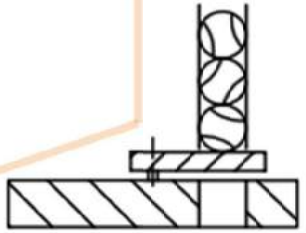
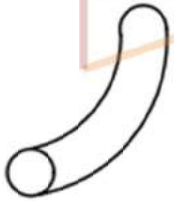


Abbildung 21: Funktionsstruktur einer Ballmaschine

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

4.2 Morphologischer Kasten

Teilfunktion	Lösung 1	Lösung 2	Lösung 3	Lösung 4
Ball aufnehmen	Korb 	Röhre 	Förder-Mechanismus 	Spirale 
Ball in definiertem Intervall abgeben	rotierende Scheibe mit Durchlass 	Ball-Förderrad 	Förder-Mechanismus 	Schranke 
Ball führen	flexibler Schlauch 	offene Führung 		

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

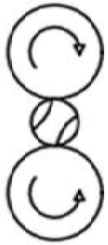
Teilfunktion	Lösung 1	Lösung 2	Lösung 3	Lösung 4	Lösung 5
Ball abschießen	2 Schwungräder 	1 Schwungrad 	Druckluft	Feder	
Benutzereingaben aufnehmen	Drehregler	Touchscreen	Fernsteuerung		
Austrittsgeschwindigkeit messen	Ball-Geschwindigkeit direkt messen	über Zustand der Maschine berechnen (z.B. Drehzahl)	nicht messen		
Geschwindigkeit einstellen	Steuerung	Regelung			
Abschusswinkel einstellen	manuell	Linearmotor	Spindel	Zahnräder	Zahnriemen
Horizontale Auslenkung einstellen	manuell	Linearmotor	Spindel	Zahnräder	Zahnriemen
Hineingreifen verhindern	Lichtschranke	Warnhinweis	Verengung	Klappe	

Tabelle 5: Morphologischer Kasten

Im Folgenden sollen die Teilfunktionen und deren mögliche Lösungen erläutert werden:

Ball aufnehmen

Zur Ballzufuhr muss eine Lagerung der Bälle vorgesehen werden. Gebräuchlich sind dabei einfache Körbe (Lösung 1). Alternativ könnten Röhren (Lösung 2) eingesetzt werden, die durch eine Verengung an einem Ende gleichzeitig dem ergonomischen Sammeln der Bälle dienen können. Ein Fördermechanismus (Lösung 3) würde eine Zufuhr vom Boden ermöglich. Mit einer spiralförmigen Lagerung (Lösung 4) könnte ein Verkeilen der Bälle ausgeschlossen werden.

Ball in definiertem Intervall abgeben

Die gebräuchlichste Variante zum Abgeben der Bälle ist eine rotierende Scheibe mit mehreren Löchern (Lösung 1). Kommt ein Ball mit einem darunter liegenden Loch zur Deckung, wird er abgegeben. Alternativ könnte ein Förderrad (Lösung 2) eingesetzt werden. Ein Fördermechanismus (Lösung 3) würde diese Funktion ebenso erfüllen. Eine Abwandlung der rotierenden Scheibe stellt eine Schranke (Lösung 4) dar.

Ball führen

Ziel dieser Funktion ist das Führen des Balls von der Aufnahme zur Abschussvorrichtung. Je nach gewählter Kombination könnte dies auch in offener Führung erfolgen (Lösung 2).

Ball abschießen

Um den Ball abzuschießen, werden in modernen Maschinen meist zwei gegengleich drehende Schwungräder eingesetzt (Lösung 1). Dies hat den Vorteil, dass beliebig hoher Top- oder Backspin erzeugt werden kann. Eine kompaktere und günstigere Variante stellt Lösung 2 dar. Hierbei könnten aber nur Bälle mit Topspin erzeugt werden. Dies kann bei höheren Geschwindigkeiten zum Problem werden, da die Drehzahl des Balles entsprechend steigt. Alternative Ausprägungen sind Druckluft (Lösung 3) und mechanische Federn (Lösung 4).

Benutzereingaben aufnehmen

Die gebräuchlichste Variante ist die Einstellung mittels Drehreglern (Lösung 1). Zur Erhöhung der Benutzerfreundlichkeit könnten Touchscreen (Lösung 2) oder Fernsteuerung mittels Fernbedienung oder Smartphone (Lösung 3) eingesetzt werden.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Austrittsgeschwindigkeit messen

Zur Erhöhung der Genauigkeit ist es sinnvoll, die Geschwindigkeit des Balls zu erfassen. Dies könnte entweder direkt (Lösung 1) oder durch Messungen an der Maschine selbst (z.B. Drehzahl der Schwungräder) erfolgen (Lösung 2). Alternativ könnte auf eine Messung verzichtet werden (Lösung 3).

Geschwindigkeit einstellen

Wird nur eine Steuerung (Lösung 1) eingesetzt, kann dies je nach gewählter Lösung eine höhere Beschleunigungszeit der Antriebe im Gegensatz zur Regelung (Lösung 2) zur Folge haben.

Abschusswinkel und horizontale Auslenkung einstellen

Das Einstellen kann entweder durch den Benutzer manuell erfolgen (Lösung 1) oder durch verschiedene elektrische Antriebe.

Hineingreifen verhindern

Als Sicherheitsmaßnahme sollte ein Hineingreifen in die Maschine zumindest durch Warnhinweise (Lösung 2) oder Verengungen (Lösung 3) verhindert werden, idealerweise aber zusätzlich durch eine geeignetere Absicherung, z.B. eine Lichtschranke (Lösung 1) oder eine Klappe (Lösung 4).

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

4.3 Auswahl geeigneter Lösungsvarianten

Durch Kombination von Teillösungen lassen sich eine sehr große Anzahl an Gesamtlösungen erstellen. Nach einer ersten Reduktion sollen folgende Varianten näher betrachtet werden:

Variante A (blau)

Aufnahme des Balles durch Röhre (2), Abgabe des Balles durch Förderrad (2), Ball offen führen (3), Abschießen durch 2 Schwungräder (1), Benutzereingaben mittels Touchscreen (2), Messung der Austrittsgeschwindigkeit des Balles (1), Einstellung der Geschwindigkeit durch Steuerung (1), Abschusswinkel manuell einzustellen (1), horizontale Auslenkung mittels Zahnriemen einzustellen (5), Hineingreifen durch Warnhinweis verhindern (2)

Variante B (rot)

Aufnahme des Balles durch Korb (1), Abgabe des Balles durch rotierende Scheibe mit Durchlass (1), Ballführung durch flexiblen Schlauch (1), Abschießen mit Feder (4), Benutzereingaben mittels Drehregler (1), keine Messung der Austrittsgeschwindigkeit (3), Einstellung der Geschwindigkeit durch Steuerung (1), Abschusswinkel manuell einzustellen (1), horizontale Auslenkung manuell einzustellen (1), Hineingreifen durch Warnhinweis verhindern (1)

Variante C (grün)

Aufnahme des Balles durch Fördermechanismus (3), Abgabe des Balles durch Ball-Fördermechanismus (3), Ball offen führen (3), Abschießen durch 2 Schwungräder (1), Benutzereingaben mittels Fernsteuerung (3), Messung der Austrittsgeschwindigkeit des Balles (1), Einstellung der Geschwindigkeit durch Regelung (2), Abschusswinkel mittels Linearmotor einzustellen (2), horizontale Auslenkung mittels Linearmotor einzustellen (2), Hineingreifen durch Lichtschranke verhindern (1)

Variante D (orange)

Aufnahme des Balles durch Spirale (4), Abgabe des Balles durch Schranke (4), Ballführung durch flexiblen Schlauch (1), Abschießen durch 1 Schwungrad (2), Benutzereingaben mittels Drehregler (1), Messung der Austrittsgeschwindigkeit über Zustand der Maschine (2), Einstellung der Geschwindigkeit mittels Steuerung (1), Abschusswinkel manuell einzustellen (1), horizontale Auslenkung manuell einzustellen (1), Hineingreifen durch Verengung verhindern (3)

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Folgende Lösungen wurden dabei nach der Vorauswahl nicht mehr berücksichtigt:

Abschussmechanismus mit Druckluft

Diese Lösung weist in ihrer Anwendung im Wesentlichen dieselben Vor- und Nachteile wie die mechanische Feder auf, ist jedoch technisch aufwendiger.

Winkeleinstellung mittels Spindel

Wird bei der Lösung ein automatisches Einstellen von Abschusswinkel und horizontaler Winkeleinstellung gewünscht, ist ein Linearantrieb einfacher umzusetzen.

Winkeleinstellung mittels Zahnräder

Der Einsatz von Zahnrädern erfordert geeignete Materialien, um auf eine Schmierung zu verzichten und Wartungsfreiheit garantieren zu können. Ebenso kann ein kleiner Achsabstand bei der Konstruktion unerwünscht sein. Aus diesen Gründen wird im direkten Vergleich der Einsatz von Zahnriemen bevorzugt.

Sicherheitsfunktion mittels Klappe

Diese Lösung ist technisch wesentlich schwieriger umzusetzen, bringt jedoch kaum Vorteile gegenüber den anderen Varianten.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

4.4 Bewertung von Lösungsvarianten

Um die vorgeschlagenen Gesamtlösungen hinsichtlich ihrer Erfüllung der Anforderungen vergleichen zu können, müssen in einem ersten Schritt die dafür relevanten Bewertungskriterien ermittelt werden. Als Basis dafür können die Anforderungsliste und die von Pahl/Beitz (2007: 269-270) vorgeschlagene Hauptmerkmalliste zum Bewerten in der Konzeptphase (Abbildung 22) dienen.

Hauptmerkmal	Beispiele
Funktion	Eigenschaften erforderlicher Nebenfunktionsträger, die sich aus dem gewählten Lösungsprinzip oder aus der Konzeptvariante zwangsläufig ergeben
Wirkprinzip	Eigenschaften des oder der gewählten Prinzipien hinsichtlich einfacher und eindeutiger Funktionserfüllung, ausreichende Wirkung, geringe Störgrößen
Gestaltung	Geringe Zahl der Komponenten, wenig Komplexität, geringer Raumbedarf, keine besonderen Werkstoff- und Auslegungsprobleme
Sicherheit	Bevorzugung der unmittelbaren Sicherheitstechnik (von Natur aus sicher), keine zusätzlichen Schutzmaßnahmen nötig, Arbeits- und Umweltsicherheit gewährleistet
Ergonomie	Mensch-Maschine-Beziehung befriedigend, keine unzulässige Belastung oder Beeinträchtigung, gute Formgestaltung
Fertigung	Wenige und gebräuchliche Fertigungsverfahren, keine aufwendigen Vorrichtungen, geringe Zahl einfacher Teile
Kontrolle	Wenige Kontrollen oder Prüfungen notwendig, einfach und aussagesicher durchführbar
Montage	Leicht, bequem und schnell, keine besonderen Hilfsmittel
Transport	Normale Transportmöglichkeiten, keine Risiken
Gebrauch	Einfacher Betrieb, lange Lebensdauer, geringer Verschleiß, leichte und sinnfällige Bedienung
Instandhaltung	Geringe und einfache Wartung und Säuberung, leichte Inspektion, problemlose Instandsetzung
Recycling	Gute Verwertbarkeit, problemlose Beseitigung
Aufwand	Keine besonderen Betriebs- oder sonstige Nebenkosten, keine Terminrisiken

Abbildung 22: Hauptmerkmalliste zum Bewerten in der Konzeptphase (Pahl/Beitz 2007: 270)

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

In der Konkretisierungsstufe der Konzeptphase ist es meist schwierig, exakte Voraussagen zu allen Bewertungskriterien zu machen. Dennoch sollen insbesondere wirtschaftliche Eigenschaften so früh wie möglich erfasst werden. Wo keine quantitativen Vergleiche angestellt werden können, müssen diese zumindest qualitativ einfließen (vgl. Pahl/Beitz 2007: 269). Für die Gewichtung der Kriterien wird im Folgenden die Methode des Paarvergleichs eingesetzt (Tabelle 6). Dies ist nicht zwingend nötig, wird jedoch von Ponn/Lindemann (2008: 115) empfohlen, wenn deutliche Unterschiede vorliegen.

Folgende Bewertungskriterien werden eingesetzt:

Mobilität: Transport im Verkehrsmittel, Beförderung vom Verkehrsmittel zum Einsatzort und zurück, einfache & ergonomische Manipulation

Genauigkeit: Exaktes Erreichen von eingestellter Geschwindigkeit, Spin und Aufprall-Punkt

Fertigung und Montage: Kosten der Produktion, Verfügbarkeit der Produktionsmaschinen, Einfachheit und Fehlerfreiheit der Montage

Bedienung: Intuitive Festlegung aller Parameter durch Benutzer, Umfang der möglichen Einstellungen, Klarheit visueller Darstellung

Sicherheit: Verhinderung von Verletzungen im Betrieb, Notabschaltung bei Fehlverhalten

Lautstärke: Lärmbelastung im Betrieb

Zuverlässigkeit: Wartungsintervalle, Einfachheit von Wartung und Reinigung

Flexibilität: maximal erreichbare Spin-Raten und Geschwindigkeiten, Einsatz verschiedener Ball-Typen (z.B. Kinderbälle)

Kapazität: Anzahl aufnehmbarer Bälle, Laufzeit

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

2 = Zeile wichtiger als Spalte 1 = gleich wichtig 0 = Zeile weniger wichtig als Spalte	Mobilität	Genauigkeit	Fertigung und Montage	Bedienung	Sicherheit	Lautstärke	Zuverlässigkeit	Flexibilität	Kapazität	Summe	Gewichtung g_i
Mobilität	1	2	1	1	1	2	1	2	2	13	0,87
Genauigkeit	0	1	0	0	0	1	0	1	1	4	0,27
Fertigung und Montage	1	2	1	0	2	2	1	2	2	13	0,87
Bedienung	1	2	2	1	2	2	1	2	2	15	1,00
Sicherheit	1	2	0	0	1	2	0	1	1	8	0,53
Lautstärke	0	1	0	0	0	1	0	1	0	3	0,20
Zuverlässigkeit	1	2	1	1	2	2	1	2	1	13	0,87
Flexibilität	0	1	0	0	1	1	0	1	1	5	0,33
Kapazität	0	1	0	0	1	2	1	1	1	7	0,47

Tabelle 6: Paarvergleichs-Matrix der Bewertungskriterien

Die Gewichtungsfaktoren g_i werden in Bezug zum wichtigsten Faktor (größte Summe) als relative Werte berechnet. Zum quantitativen Vergleich der Gesamtlösungen wird die Methode der gewichteten Punktbewertung angewendet. Dazu werden die Lösungsvarianten hinsichtlich der Bewertungskriterien mit Punkten w_i bewertet und diese mit ihrer Gewichtung multipliziert und zur einer Gesamtsumme addiert (Tabelle 7).

Alternativ steht eine große Anzahl weiterer Bewertungsmethoden zur Verfügung, zum Beispiel das Ausschließen nach KO-Kriterien, Vorteil-Nachteil-Vergleich, Nutzwertanalyse oder die technisch-wirtschaftliche Bewertung nach VDI-Richtlinie 2225. Welche Methode angewendet wird, hängt dabei im Allgemeinen von dem vertretbaren Aufwand sowie dem Konkretisierungsgrad des Produkts ab (vgl. Ponn/Lindemann 2008: 113-114; Ehrlenspiel et al. 2014: 75-76).

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

3 = sehr gute Lösung 2 = akzeptable Lösung 1 = suboptimale Lösung		Variante A		Variante B		Variante C		Variante D	
	g_i	w_i	$g_i \cdot w_i$	w_i	$g_i \cdot w_i$	w_i	$g_i \cdot w_i$	w_i	$g_i \cdot w_i$
Mobilität	0,87	3	2,61	3	2,61	1	0,87	3	2,61
Genauigkeit	0,27	2	0,54	1	0,27	3	0,81	2	0,54
Fertigung und Montage	0,87	2	1,74	3	2,61	1	0,87	2	1,74
Bedienung	1,00	3	3,00	1	1,00	3	3,00	1	1
Sicherheit	0,53	1	0,53	1	0,53	3	1,59	2	1,06
Lautstärke	0,20	2	0,40	3	0,60	1	0,20	2	0,4
Zuverlässigkeit	0,87	2	1,74	2	1,74	2	1,74	2	1,74
Flexibilität	0,33	3	0,99	1	0,33	3	0,99	1	0,33
Kapazität	0,47	1	0,47	2	0,94	2	0,94	2	0,94
Summe			12,04		10,63		11,01		10,36

Tabelle 7: Gewichtete Punktbewertung

Aufgrund der Bewertung wird Variante A im Weiteren als Basis des Funktionsmusters ausgewählt. Daraus gewonnene Erkenntnisse können später zur Zusammenstellung einer verbesserten Variante genützt werden und diese dann einer erneuten Evaluierung unterzogen werden.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

5 Entwerfen

5.1 Elektronik-Versuchsaufbauten

In einer Tennisballmaschine kommen zahlreiche unterschiedliche Elektronikteile zum Einsatz. Um die optimalen Bauteile zu finden, werden in den folgenden Kapiteln verschiedene Versuchsaufbauten beschrieben. Mit Hilfe dieser Aufbauten wurde versucht, die ideale Kombination der Elektronikteile zu finden. Die Datenblätter der verwendeten elektronischen Komponenten befinden sich im Anhang B.

5.1.1 Serielle Schnittstelle zwischen Arduino und Raspberry PI

Aufgabenstellung

In vielen komplexen Aufgabenstellungen ist es notwendig, dass ein Raspberry PI mit einem Arduino kommunizieren muss. Dies kann viele Gründe haben, wie z.B. nicht ausreichend genug PINs am Port oder wenn eine graphische Oberfläche erwünscht ist. In diesem Experiment soll die Kommunikation mittels einer USB-Verbindung erfolgen.

Benötigte Bauteile

(1) x Raspberry Pi 3 Model B+
(1) x Arduino UNO R3
(1) x USB-Kabel

Einführung in die Komponenten

Raspberry PI
Beim Raspberry PI handelt es sich um einen Einplatinencomputer. Das bedeutet, alle zum Betrieb nötigen elektronischen Komponenten befinden sich auf einer einzigen Leiterplatte. Ein anderes Beispiel für einen Einplatinencomputer wäre ein Router. Der Computer besitzt in etwa die Größe einer Kreditkarte, 512 MB Arbeitsspeicher, USB 2.0 Anschlüsse, HMDI- und FBAS-Ausgänge zum Anschluss eines Monitors sowie eine 3,5 mm Klinkenbuchse zur analogen Tonausgabe und eine Ethernet-Buchse. Zusätzlich besitzt der Raspberry PI einen SD-Karteneinschub, da im Raspberry Pi eine SD-Karte als externes Speichermedium eingesetzt wird. Als Eingabegeräte können – wie auch bei einem herkömmlichen Computer – eine Maus und eine Tastatur mittels USB angeschlossen werden.

Als Betriebssystem können *Raspbian*, *Libre-ELEC*, *Raspbian Lite*, *Lakka*, *OSMC*, *Windows 10 IoT Core* und viele weitere ausgewählt werden.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Das aktuellste Modell des Computers trägt den Namen *Raspberry PI 3 Modell B+*. Dieses ist mit seinen wichtigsten Komponenten in Abbildung 23 dargestellt.

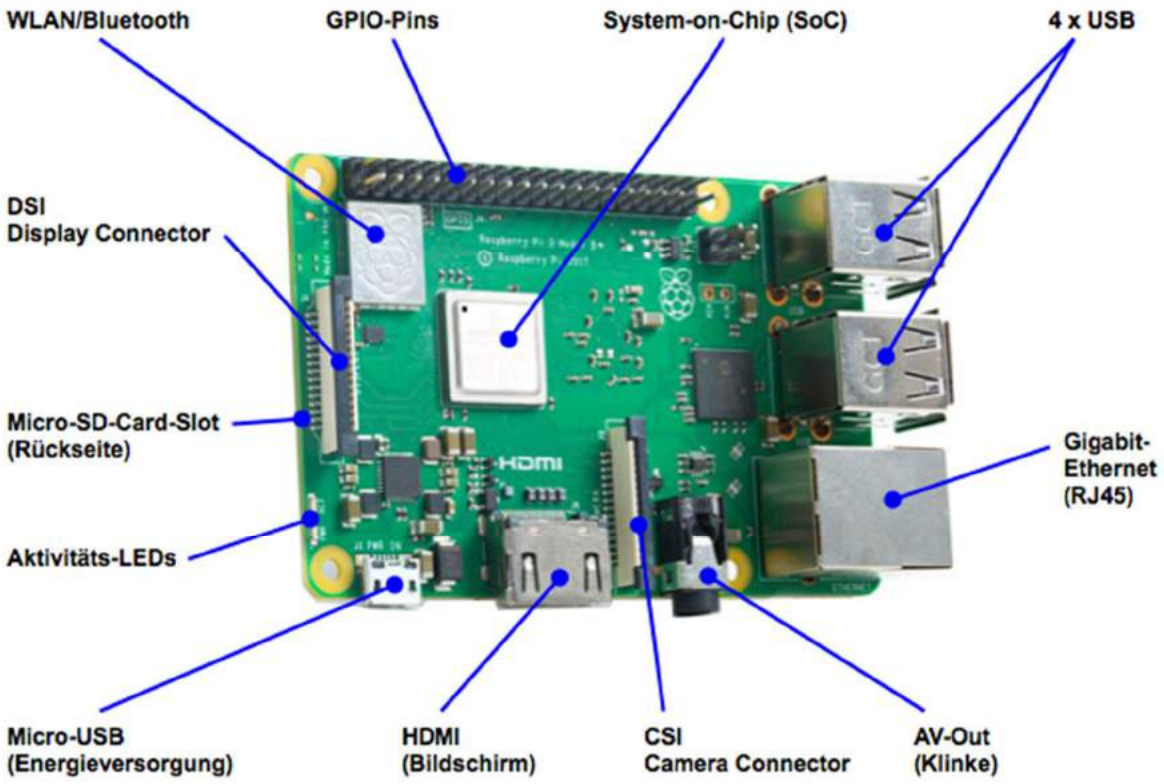


Abbildung 23: Wichtige Komponenten des Raspberry PI 3 Model B+

Arduino

Das erste bekannte Open-Source Projekt, welches die beiden Komponenten Software und Hardware miteinander verbindet, ist der Arduino. Mit diesem Gerät wurde versucht einen einfachen, kostengünstigen Zugang zur Programmierung und Hardware-Entwicklung zu ermöglichen. (Dembowski, 2015: 5)

Demnach besteht die Arduino-Plattform aus zwei Teilen: aus der Hardware und einem Software-System.

Das Mikrocontroller Board des Hardware-Systems stellt mittels Ein- und Ausgangsports die Verbindung zur physischen Welt her. Über die digitalen und analogen Eingänge können Daten von Sensoren wie Schaltern und Temperatursensoren oder auch von GPS-Modulen eingespielt werden. Die Ausgänge dienen zur Ansteuerung von Leuchtdioden, Relais oder Motoren.

Meist werden die Leiterplatten als *Board* bezeichnet. Am Markt gibt es verschiedene Boards, die je nach Anforderungen – wie Schnelligkeit und Anzahl der Ports – ausgewählt werden. Das meist verkaufte Board besitzt die Bezeichnung *Arduino UNO*.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Dieses basiert auf dem *Atmels Controller ATmega328* und hat insgesamt 14 digitale I/O-Pins (davon 6 als PWM-Ausgänge) und 6 analoge Eingänge.

In Abbildung 24 sind die wichtigsten Komponenten des Boards *Arduino UNO* dargestellt.

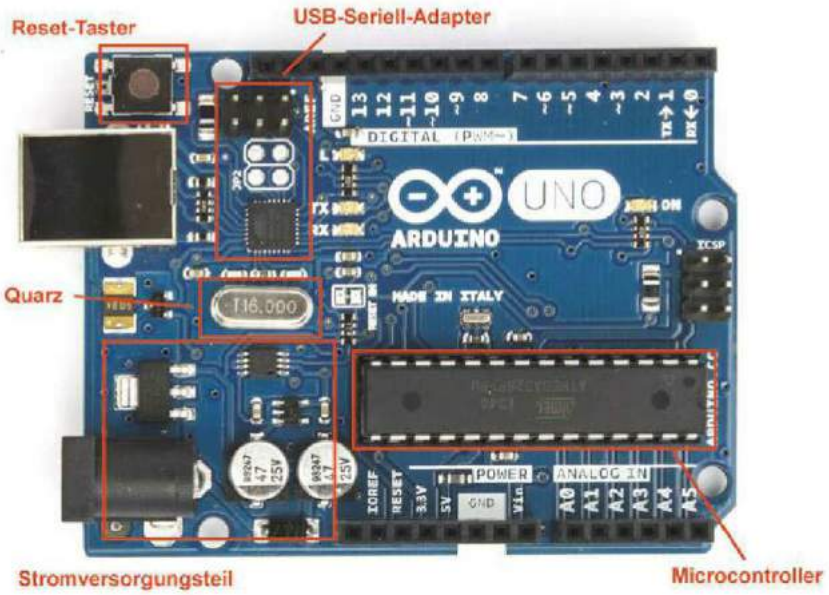


Abbildung 24: Wichtige Komponenten des Arduino UNO (BRÜH: 27)

- **Reset-Taster:** führt zum Zurücksetzten und Neustart des Mikrocontrollers.
- **USB-Seriell-Adapter**: wandelt das USB-Signal in ein serielles Signal um.
- **Quarz:** stellt den Takt für den Mikrocontroller bereit.
- **Stromversorgung:** regelt die Spannung auf 5 Volt.
- **Mikrocontroller:** führt die Programme aus und verarbeitet die Ein- und Ausgangssignale

Das Software-System ist die Entwicklungsumgebung, welche auch als *IDE* (Integrated Development Environment) bezeichnet wird. Diese basiert auf der Programmiersprache *Processing* (ähnelt der Programmiersprache *C++*), mit der die Programme – auch *Sketch* genannt – erstellt werden, welche später von der Hardware ausgeführt werden.

In Abbildung 25 werden die Anschlussmöglichkeiten vom *Arduino UNO* aufgezeigt.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

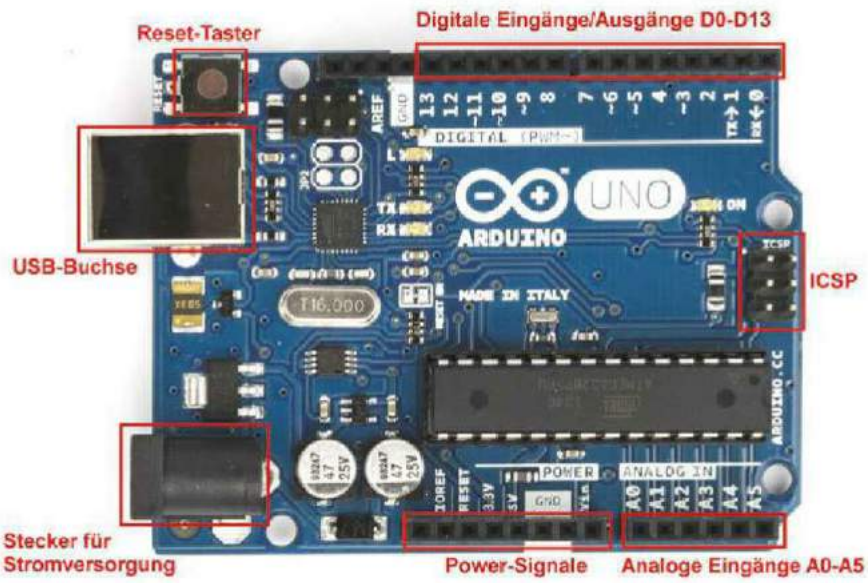


Abbildung 25: Anschlussmöglichkeiten beim Arduino UNO (BRÜH: 26)

Unterschied zwischen Raspberry PI und Arduino

Der Arduino ist im Gegensatz zum Raspberry PI nicht als Computer mit eigenem Betriebssystem ausgelegt und kann damit keine Anwendungen wie *Office*, *Media-center* oder *Web-Server* ausführen. Der Vorteil des Raspberry PIs beruht darauf, dass er auch ohne Programmierkenntnisse seitens des Anwenders als Mini-PC eingesetzt werden kann. Bei dem Arduino muss hingegen stets ein Mikrocontroller für eigene Projekte programmiert werden.

Der Arduino spult ein vorher aufgespieltes Programm stets in einer Schleife ab. Der Raspberry PI vermag hingegen mehrere Programme nacheinander und/oder parallel auszuführen.

Vorbereitungen am Raspberry PI

Um eine erfolgreiche Verbindung zwischen dem Arduino und dem Raspberry PI herstellen zu können, müssen am Raspberry PI einige Vorbereitungen getroffen werden. Zunächst einmal müssen mit folgendem Befehl zusätzliche Bibliotheken installiert werden:

```
sudo apt-get install minicom python-serial
```

Im nächsten Schritt muss festgestellt werden, an welchem Anschluss der Arduino angeschlossen ist. Dazu wird der nachfolgende Befehl zweimal ausgeführt: Einmal ohne Verbindung zum Arduino und einmal mit USB-Verbindung zum Arduino:

```
ls /dev/tty*
```

Der nun dazugekommene Eintrag ist der Name, mit dem der Arduino über die serielle Schnittstelle angesprochen werden kann.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Schaltplan

Wie in Abbildung 26 ersichtlich ist, sind der Raspberry PI und der Arduino lediglich mit einem USB-Kabel verbunden.

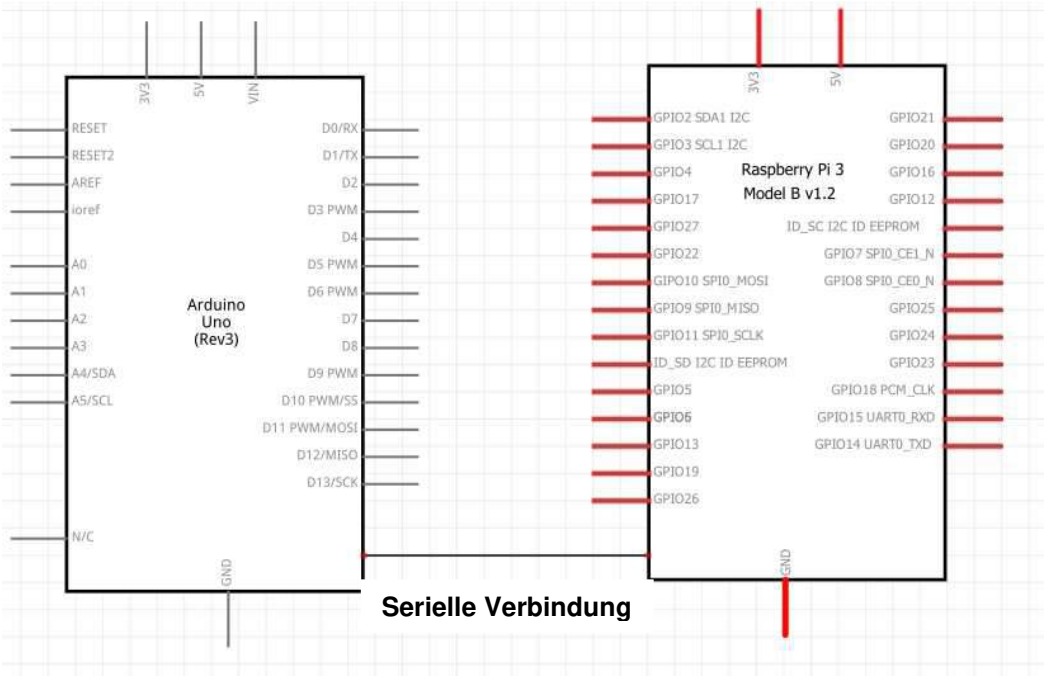


Abbildung 26: Serielle Schnittstelle zwischen Arduino und Raspberry PI – Schaltplan

Verbindungsschema

In Abbildung 27 ist das Verbindungsschema ersichtlich.

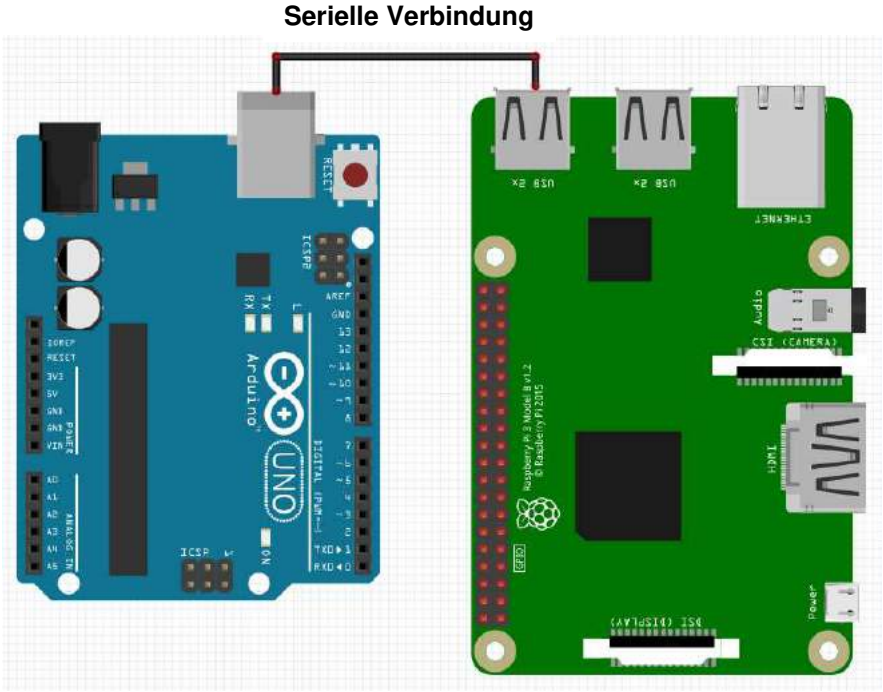


Abbildung 27: Serielle Schnittstelle zwischen Arduino und Raspberry PI – Verbindungsschema

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Versuchsaufbau

In Abbildung 28 wird der Versuchsaufbau dargestellt.

Abbildung 28: Serielle Schnittstelle zwischen Arduino und Raspberry PI – Versuchsaufbau

Code (Arduino)

```
void setup(){
    Serial.begin(9600);
    pinMode(13, OUTPUT); //PIN 13 als Ausgabe konfigurieren
}

void loop(){
    if (Serial.available() ){ //Abfrage, ob Zeichen gesendet wurde
        byte empfangen = Serial.read(); //Einlesen des empfangenen Zeichens
        if(empfangen == 49){ //Prüfen, ob das Zeichen "1" gesendet wurde
          digitalWrite(13, HIGH); //LED einschalten
          Serial.println("LED ist an"); //String zurück an den Raspberry schicken
        }else if(empfangen == 48){ //Prüfen, ob das Zeichen "0" gesendet wurde
          digitalWrite(13, LOW); //LED ausschalten
          Serial.println("LED ist aus"); //String zurück an den Raspberry schicken
        }else{ //anderes Zeichen geschickt
          Serial.println("Falsche Eingabe!!!"); //Zurück an den Raspberry schicken
        }
    }
}
```

Der Befehl `if (Serial.available())` prüft, ob ein Zeichen an den Arduino, vom Raspberry PI kommend, gesendet wurde. Wenn dies der Fall ist, werden die Zeichen über die Funktion `Serial.read()` eingelesen und auf der Variablen „empfangen“ gespeichert. Die Funktion `Serial.println(„…“)` ist quasi der Retourwert an den Raspberry PI. Das bedeutet in diesem Fall eine Ausgabe auf der Konsole des Raspberry PIs.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Code (Raspberry PI)

```
1.  import serial #Package für serielle Verbindungen
2.  import time #Package für die Funktion sleep
3.
4.  verbindung = serial.Serial('/dev/ttyACM0', 9600) #Port zuweisen, an dem der Arduino
    hängt
5.  verbindung.isOpen() #Verbindung aufbauen
6.  print('Verbindung wird aufgebaut...')
7.  time.sleep(3) #3 Sekunden warten
8.
9.  try:
10.     while True: #Endlosschleife
11.         eingabe = input("Bitte 0 oder 1 eingeben!")
12.         verbindung.write(eingabe.encode()) #eingabe wird an den Arduino gesendet
13.         rueckgabe = verbindung.readline() #Rückgabe-
    String vom Arduino wird gespeichert
14.         print(rueckgabe)
15. except KeyboardInterrupt: #Mit STRG+C kann das Programm beendent werden
16.     verbindung.close() # Verbindung schließen
17.     print("Verbindung abgebrochen")
```

Mit dem Befehl `serial.Serial('/dev/ttyACM0', 9600)` wird der Verbindung der Port zugewiesen, an dem der Arduino angeschlossen ist. Nun kann mit der Variablen „verbindung“ mit dem Arduino kommuniziert werden. Es können nun mit der Funktion `verbindung.write()` Zeichen an den Arduino übermittelt und mit der Funktion `verbindung.readline()` empfangen werden.

Über die Ausnahmebehandlung kann das Programm mit der Tastenkombination „STRG+C“ beendet werden. Damit wird dann auch über den Befehl `verbindung.close()` die Verbindung zum Arduino ordnungsgemäß abgebrochen.

Die Konsolenausgabe des Python-Programms ist in Abbildung 29 ersichtlich.

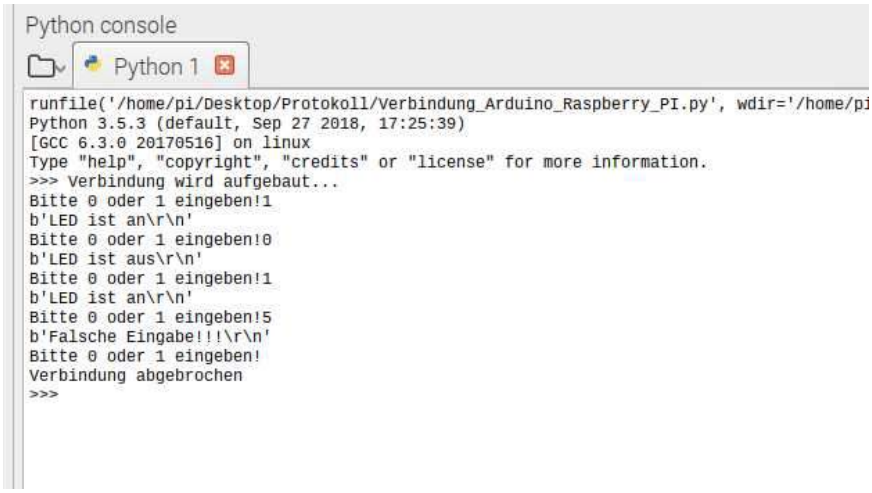


Abbildung 29: Serielle Schnittstelle zwischen Arduino und Raspberry PI – Konsolenausgabe

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

5.1.2 Lichtschranken-Aufbau mit Arduino

Aufgabenstellung

Um eine Reaktion auf bestimmte Zustände zu ermöglichen, kommen im Funktionsmuster Lichtschranken zum Einsatz. Bei diesem Versuchsaufbau soll der Arduino erkennen, dass die Lichtschranke unterbrochen wurde und diesen Zustand im Serial Monitor wiedergeben. Des Weiteren soll eine LED (rot) am Steckbrett aufleuchten, um die Unterbrechung des Lichtschrankens zu signalisieren.

Benötigte Bauteile

(1) x Arduino UNO R3
(1) x 5mm rote LED
(1) x Laser Rot 650 nm 5V
(1) x Fotodiode BPW 34
(1) x 220 Ω Widerstand
(1) x 100 Ω Widerstand
(1) x 10 kΩ Widerstand

Schaltplan

Wie in Abbildung 30 zu erkennen ist, dient der 220 Ω Widerstand als Vorwiderstand für die LED-Diode (rot) und der 100 Ω Widerstand als Vorwiderstand für den Laser. Des Weiteren ist ersichtlich, dass die Fotodiode mit einem 10 kΩ Widerstand in Serie geschaltet ist. Der feste Widerstand und die Fotozelle sollen ähnlich wie ein Potentiometer zusammenarbeiten. Bei starker Beleuchtung ist der Widerstand der Fotozelle im Vergleich zum festen Widerstand sehr gering. Das entspricht einem Potentiometer in der Maximalstellung. Wenn die Fotozelle sich in dunkler Umgebung befindet bzw. der Lichtschranken unterbrochen ist, wird ihr Widerstand größer als der des festen 10 kΩ Widerstands und es verhält sich wie ein Potentiometer in Minimalstellung (nach GND).

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

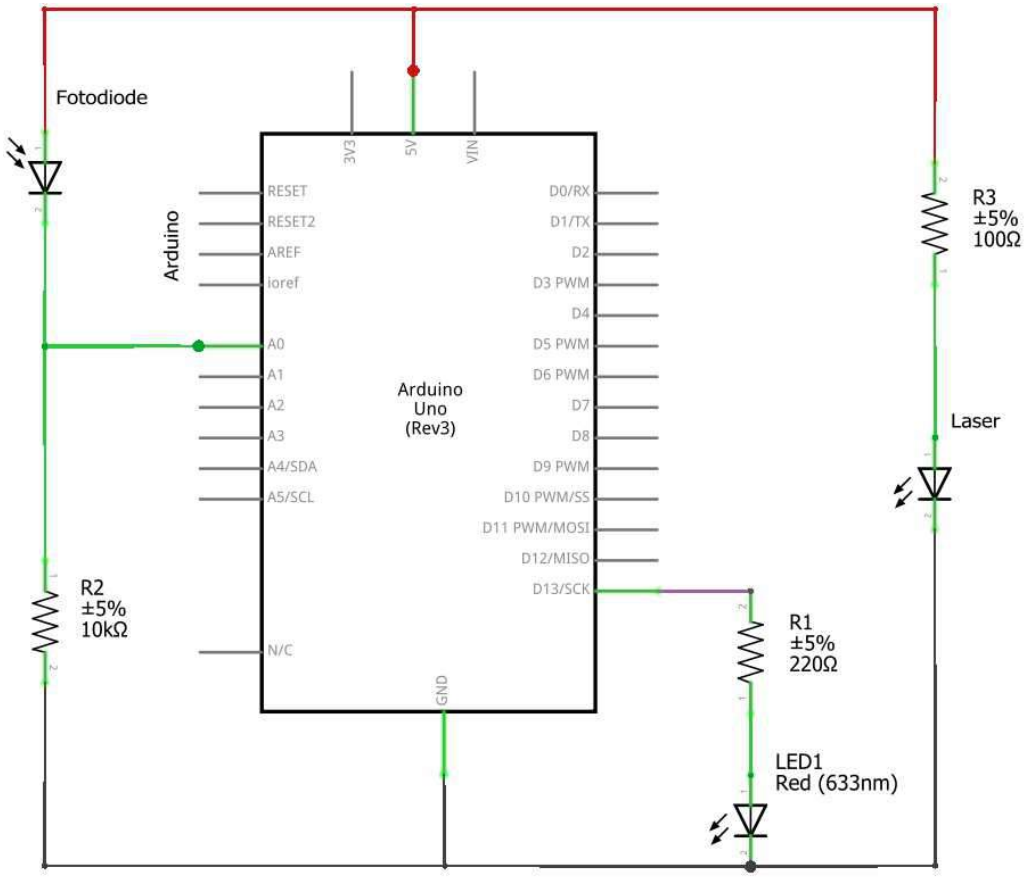


Abbildung 30: Lichtschranken-Aufbau mit Arduino – Schaltplan

Verbindungsschema

In Abbildung 31 ist das Verbindungsschema des Versuchsaufbaus *Lichtschranken-Aufbau mit Arduino* ersichtlich.

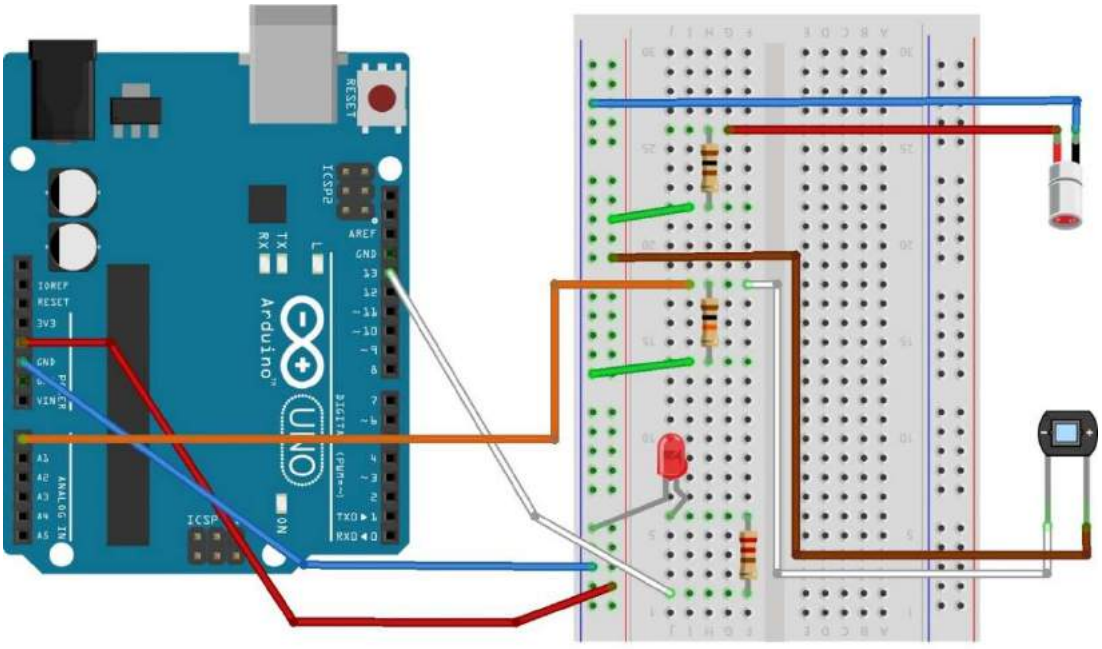


Abbildung 31: Lichtschranken-Aufbau mit Arduino – Verbindungsschema

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Versuchsaufbau

In Abbildung 32 wird der Versuchsaufbau dargestellt.

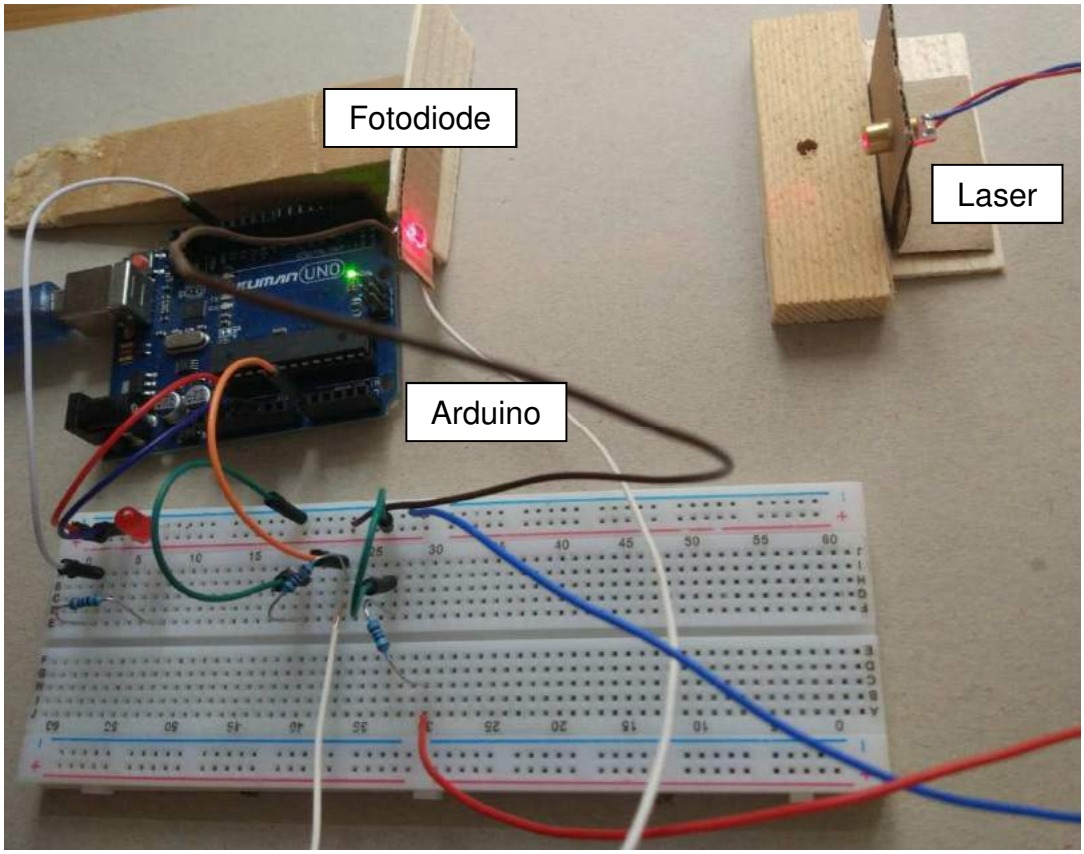


Abbildung 32: Lichtschranken-Aufbau mit Arduino – Versuchsaufbau

Code

```
#define LED_PIN 13 //LED PIN-Belegung

int licht;

void setup() {
  Serial.begin(9600);
  pinMode(LED_PIN, OUTPUT);
}

void loop() {
 licht = analogRead(0);
 //Serial.println(licht);

 if(licht>300){ //Lichtschranken offen
  digitalWrite(LED_PIN, LOW);
  Serial.println("Lichtschranken offen");
 }
 else{ //Lichtschranken unterbrochen
  digitalWrite(LED_PIN, HIGH); //
  Serial.println("Lichtschranken unterbrochen");
 }
 delay(200);

}
```

```
COM6 (Arduino/Genuino Uno)
Lichtschranken offen
Lichtschranken unterbrochen
Lichtschranken unterbrochen
Lichtschranken unterbrochen
Lichtschranken unterbrochen
Lichtschranken unterbrochen
Lichtschranken unterbrochen
Lichtschranken offen
Lichtschranken offen
Lichtschranken offen
```

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

5.1.3 Lichtschranken-Aufbau mit Raspberry PI

Aufgabenstellung

Die Aufgabenstellung ist ähnlich der vom Kapitel 5.1.2. Auch bei diesem Versuchsaufbau ist das Ziel zu erkennen, ob eine Lichtschranke unterbrochen wurde. Im Gegensatz zum vorherigen Versuchsaufbau soll dies nun mit einem Raspberry PI 3 Model B+ umgesetzt werden.

Benötigte Bauteile

(1) x Raspberry PI 3 Model B+
(1) x Laser Rot 650 nm 5V
(1) x Fotodiode BPW 34
(1) x 100 Ω Widerstand
(1) x 10 kΩ Widerstand

Schaltplan

Wie in Abbildung 33 zu erkennen ist, dient der 100 Ω Widerstand als Vorwiderstand für den Laser. Des Weiteren ist aus der Abbildung ersichtlich, dass die Fotodiode mit einem 10 kΩ Widerstand in Serie geschaltet ist.

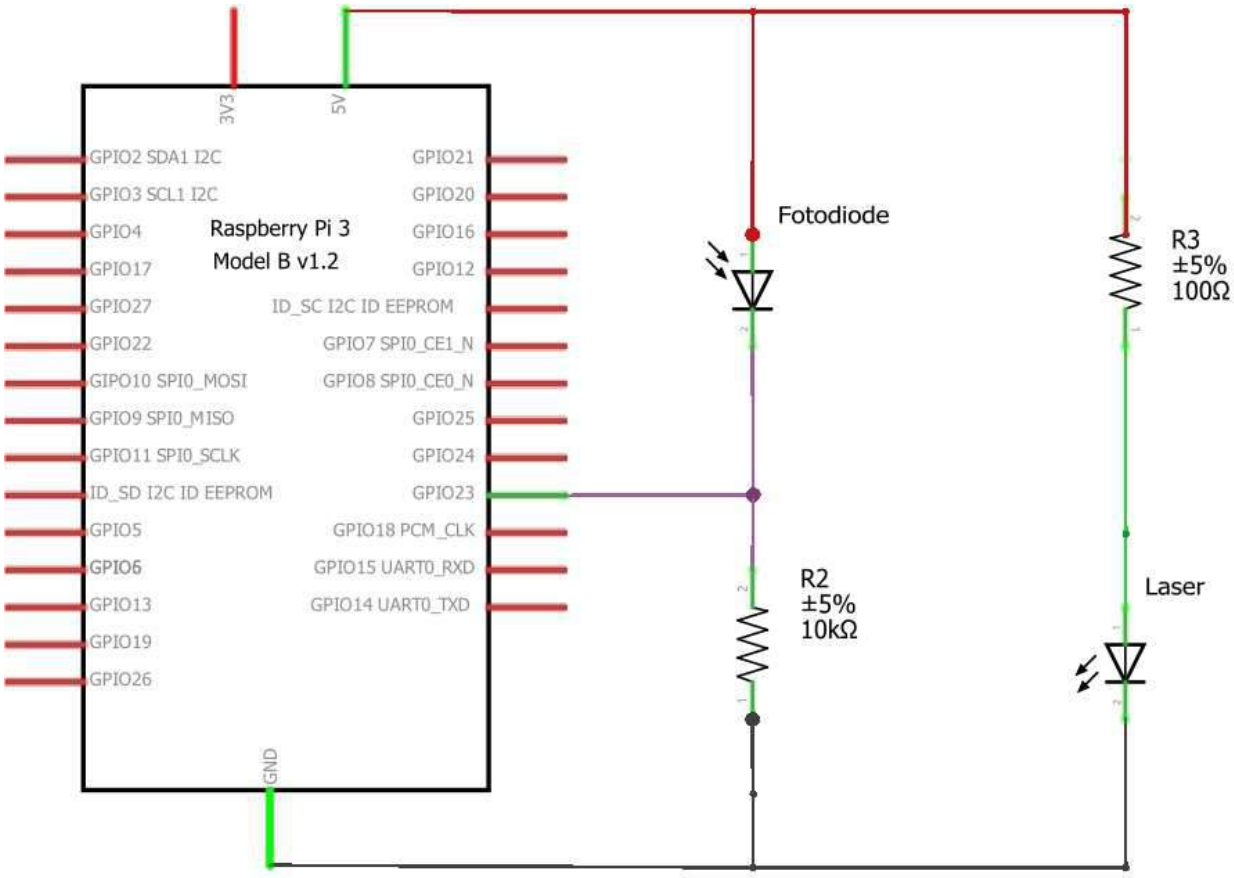


Abbildung 33: Lichtschranken-Aufbau mit Raspberry PI 3 – Schaltplan

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Verbindungsschema

In Abbildung 34 ist das Verbindungsschema des Versuchsaufbaus ersichtlich.

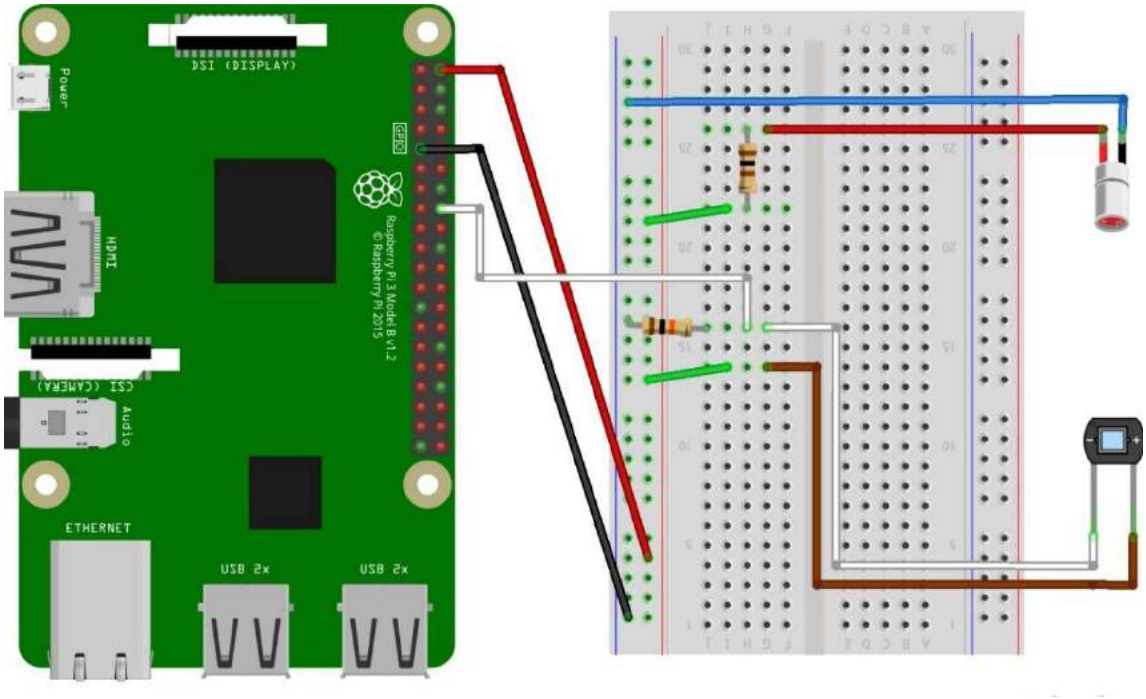


Abbildung 34: Lichtschranken-Aufbau mit Raspberry PI 3 – Verbindungsschema

Versuchsaufbau

In Abbildung 35 wird der Versuchsaufbau dargestellt.

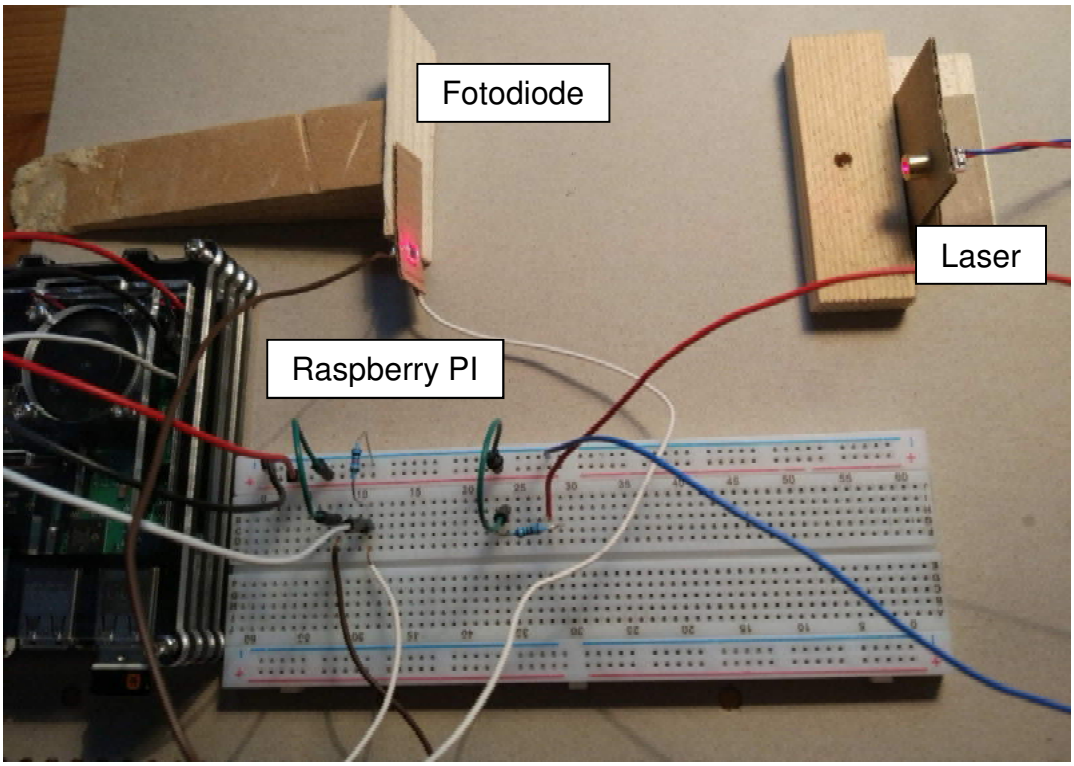


Abbildung 35: Lichtschranken-Aufbau mit Raspberry PI 3 – Verbindungsaufbau

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Code

```
1.  import RPi.GPIO as GPIO
2.  import time
3.
4.  RECEIVER_PIN = 23
5.
6.  def callback_func(channel):
7.      if GPIO.input(channel):
8.          print("Lichtschranken unterbrochen")
9.
10.
11. if __name__ == '__main__':
12.
13.     GPIO.setmode(GPIO.BCM)
14.     GPIO.setwarnings(False)
15.
16.     GPIO.setup(RECEIVER_PIN, GPIO.IN) # Pin 23 als Eingang definieren
17.     GPIO.add_event_detect(RECEIVER_PIN, GPIO.RISING, callback=callback_func, bouncet
    ime=50)
18.
19.     try:
20.         while True:
21.             time.sleep(0.5)
22.     except:
23.         # Event wieder entfernen mittels:
24.         GPIO.remove_event_detect(RECEIVER_PIN)
```

Die Konsolenausgabe des Python-Programms ist in Abbildung 36 ersichtlich.

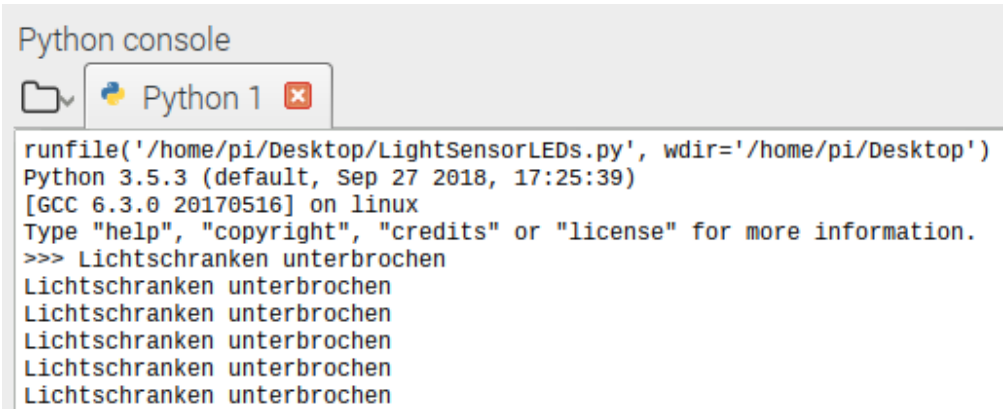


Abbildung 36: Lichtschranken-Aufbau mit Raspberry PI 3 – Konsolenausgabe

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

5.1.4 Geschwindigkeitsmessung-Aufbau

Aufgabenstellung

Ziel dieses Versuchsaufbaus ist es die Geschwindigkeit des Balles ermitteln zu können. Die Umsetzung erfolgt mithilfe zweier Lichtschranken, die jeweils aus einem Laser und einer Fotodiode bestehen. Die zwei Schranken werden in einem bestimmten Abstand voneinander positioniert. Die aktuell gemessene Momentangeschwindigkeit und die maximal gemessene Geschwindigkeit werden auf einem LCD-Display dargestellt.

Benötigte Bauteile

(1) x Arduino UNO R3
(1) x LCD-Display 4x20
(1) x LCD-I²C Modul
(2) x Laser Rot 650 nm 5V
(2) x Fotodiode BPW 34
(2) x 100 Ω Widerstand
(2) x 10 kΩ Widerstand

Schaltplan

Wie in Abbildung 37 zu erkennen ist, ist die Vorgehensweise und die Handhabung mit Laser und Fotodiode ident wie in Kapitel 5.1.2.

Um die Anzahl der Anschlüsse am Arduino zu reduzieren, wurde ein *LCD-I²C Modul* verwendet. Dieses Modul wird, wie in der Abbildung dargestellt, mit dem LCD-Display über 16 PINs verbunden. Das *LCD-I²C Modul* benötigt dann auch nur mehr 4 PINs – 2 PINs für die Stromversorgung und 2 PINs für die Ansteuerung der Ausgabe.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

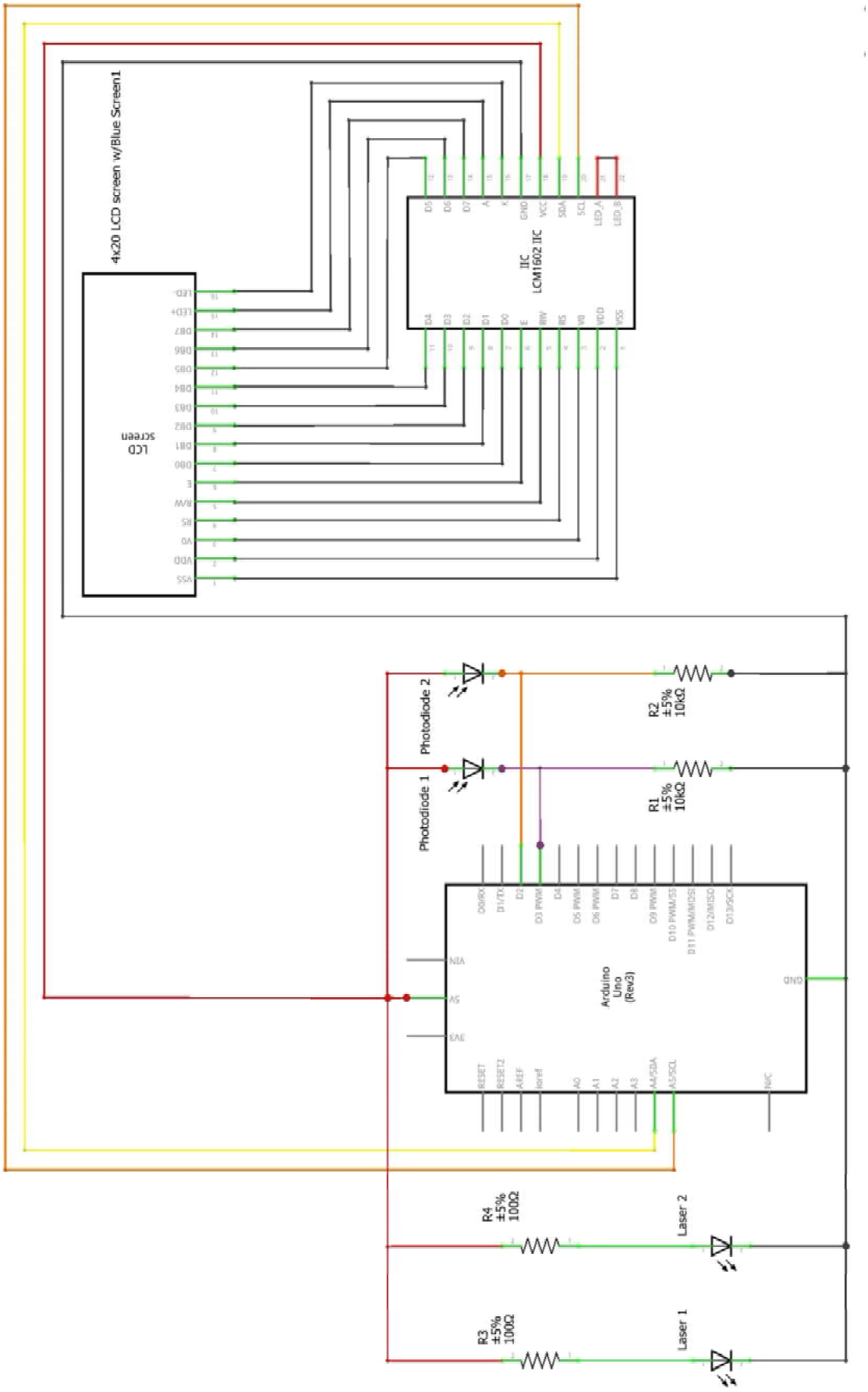


Abbildung 37: Geschwindigkeitsmessung-Aufbau – Schaltplan

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Verbindungsschema

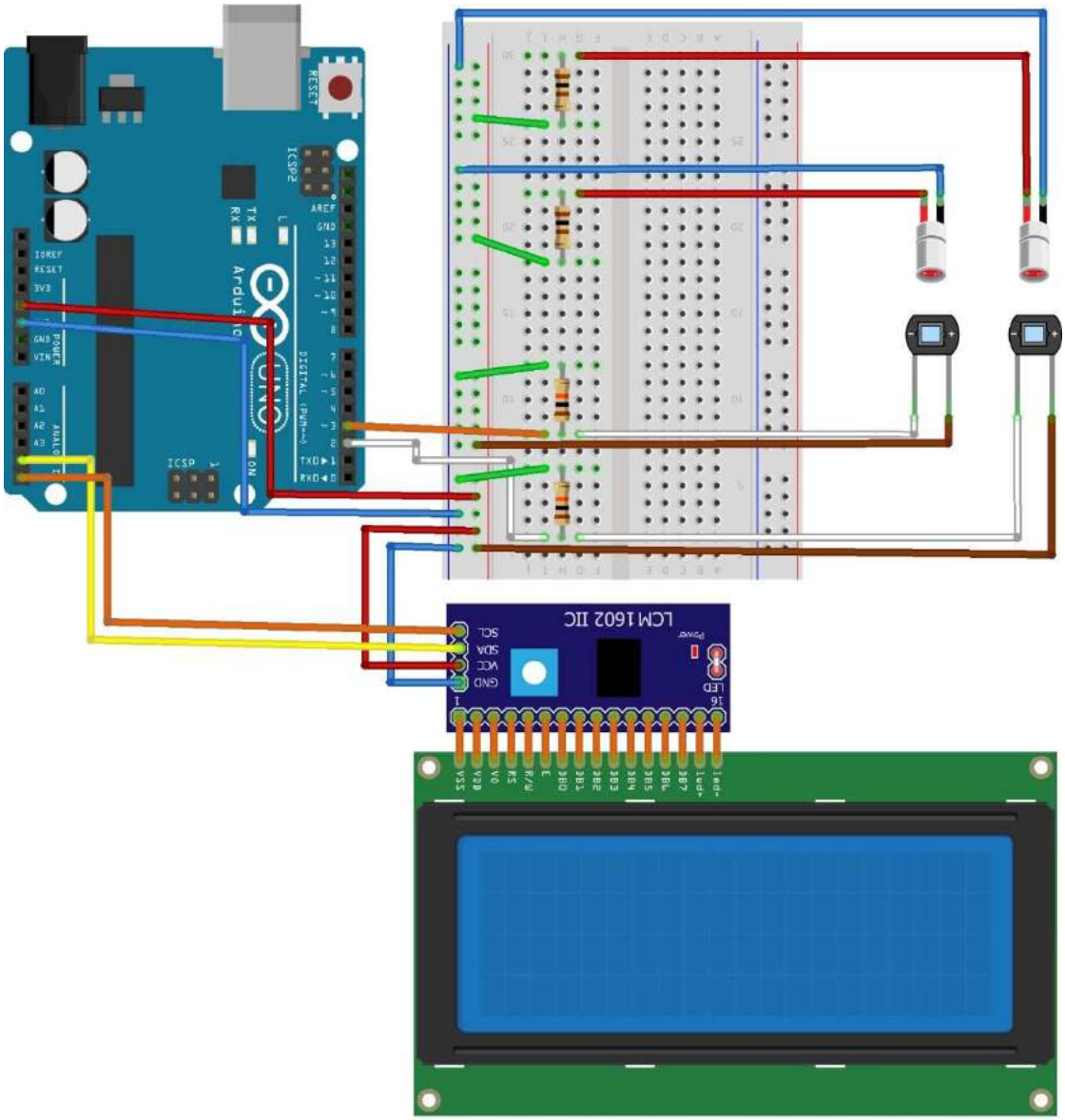


Abbildung 38: Geschwindigkeitsmessung-Aufbau – Verbindungsschema

In Abbildung 38 ist ersichtlich, dass das *LCD-I²C Modul* mit den analogen Eingang-PINs A4 und A5 am Arduino verbunden ist. Demnach wird der Vorteil des Moduls erkennbar. Die Ausgang-PINs bleiben somit frei für andere Geräte. Demnach ergibt sich folgendes Schema zum Anschließen des LCD-I²C Moduls:

- GND → GND
- VCC → 5V Kontakt am Arduino
- SDA → mit dem Analogen Eingang A4
- SCL → mit dem Analogen Eingang A5

Zusätzlich kann über ein Potentiometer der Kontrast des Displays eingestellt werden. Der LCD-Display verfügt über 4 Zeilen mit Platz für je 20 Zeichen.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Versuchsaufbau

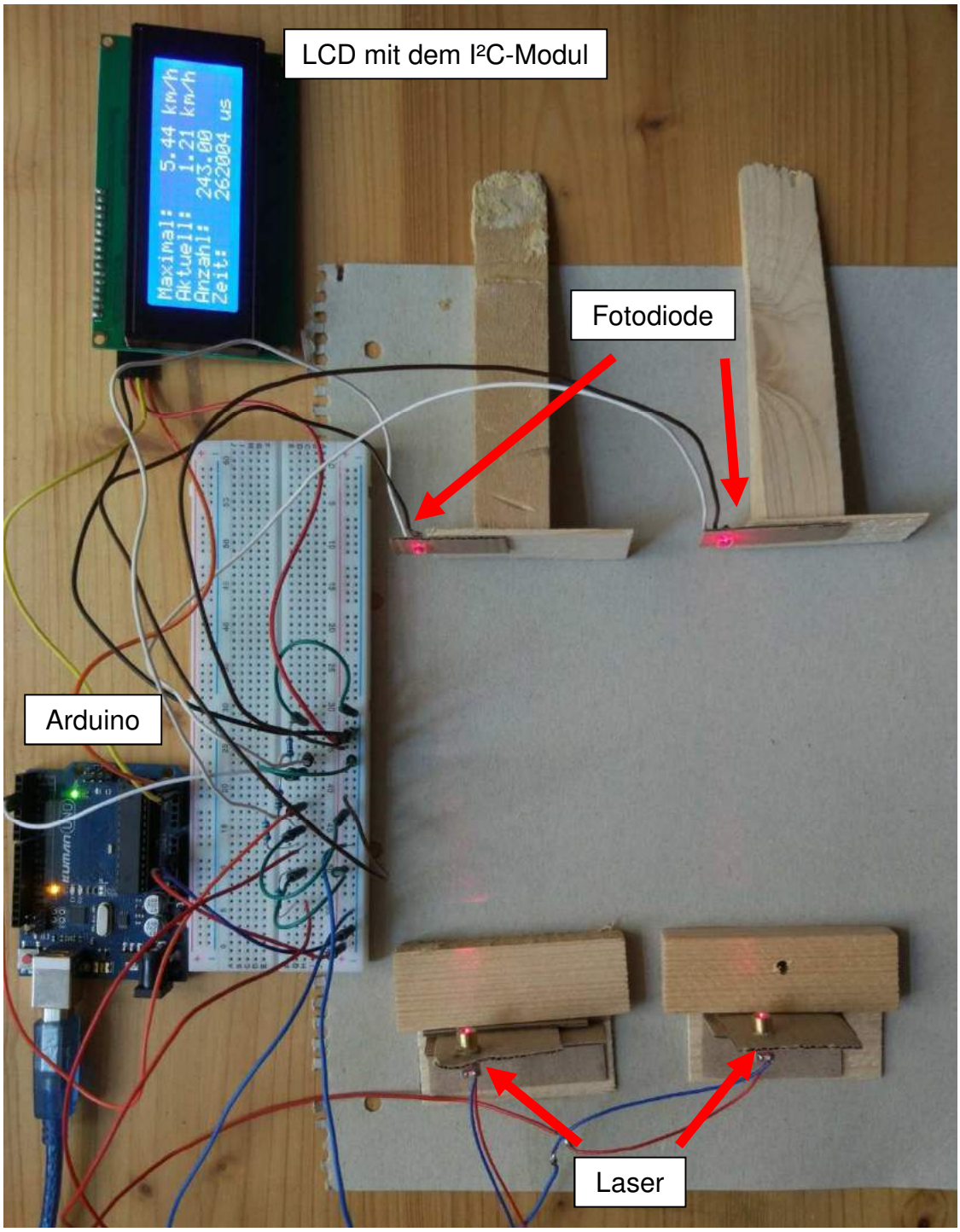


Abbildung 39: Geschwindigkeitsmessung-Aufbau – Versuchsaufbau

Wie in Abbildung 39 ersichtlich, werden auf dem LCD-Display folgende Werte dargestellt.

1.Zeile: maximale Geschwindigkeit
2.Zeile: aktuelle Geschwindigkeit
3.Zeile: Anzahl der Messungen
4.Zeile: Dauer der Messung

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Code

Damit das *LCD-I²C Modul* verwendet werden kann, wird zusätzlich das Library *LiquidCrystal I2C* benötigt, welches nicht am Arduino vorinstalliert ist. In diesem Versuchsaufbau wurde das verwendet. Zusätzlich muss auch ein Library für das LCD-Display eingebunden werden. In diesem Fall wurde dafür eine Verknüpfung zum Library *LCD* erstellt.

```
#include <Wire.h>
#include <LCD.h> //Package für den LCD-Bildschirm
#include <LiquidCrystal_I2C.h> //Package für das I²C-Modul
```

In der nachfolgenden Programmcode-Zeile wird ein Objekt des Typs *LiquidCrystal_I2C* mit dem Namen *lcd* angelegt. Mithilfe dieser Variable wird auf den LCD-Display zugegriffen.

```
LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7); //Konfig des I²C
```

Der vollständige Programmcode ist in Anhang A ersichtlich.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

5.1.5 Gleichstrommotor-Aufbau

Aufgabenstellung

Damit die Ballmaschine funktionsfähig wird, werden verschiedene Arten von Gleichstrommotoren zum Einsatz kommen – einerseits zur Beförderung der Bälle und andererseits zur Bewegung der Maschine. In diesem Aufbau soll per Eingabe auf einer graphischen Oberfläche am Raspberry PI, welcher über eine serielle Schnittstelle mit einem Arduino verbunden ist, die Drehzahl zweier Gleichstrommotoren gesteuert werden.

Benötigte Bauteile

(1) x Raspberry Pi 3 Model B+
(1) x Arduino UNO R3
(1) x USB-Kabel
(1) x 600W Netzteil 12V/50A
(2) x Motortreiber Cytron 13A
(2) x Gleichstrommotor RS PRO 12V

Einführung in die Komponenten

Gleichstrommotor

Mithilfe eines Gleichstrommotors – auch Gleichstrommaschine genannt – wird elektrische in mechanische Energie umgewandelt.

Die Funktionsweise eines Gleichstrommotors besteht darin, dass durch einen Feldmagneten ein elektrisches Feld – in dem ein Elektromagnet (Anker) drehbar gelagert ist – aufgebaut wird. Der Anker wird über Kohlebürsten, welche als Schleifkontakte dienen, an eine Stromquelle angeschlossen. Dadurch kommt es zu abstoßenden und anziehenden Kräften zwischen den Magnetpolen, welche in einer Drehbewegung des Ankers resultieren.

Der Aufbau eines Gleichstrommotors ist schematisch in Abbildung 40 dargestellt.

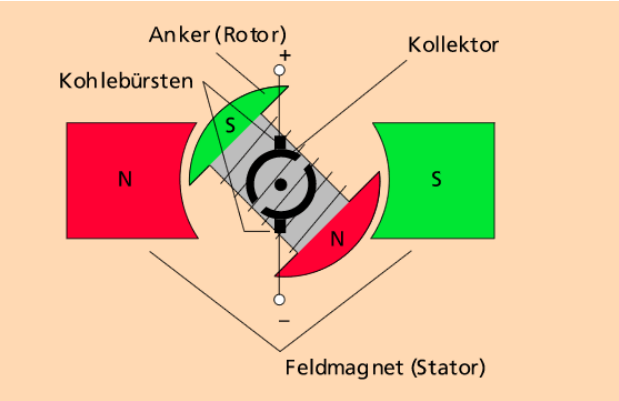


Abbildung 40: Bauteile eines Gleichstrommotors (Lernhelfer, 2019)

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Unterschieden wird grundsätzlich zwischen zwei Typen von Gleichstrommotoren, den bürstenlosen und den bürstenbehafteten Motor.

Bei einem Bürstenmotor sind die Permanentmagnete am Stator und die stromführenden Leiter am Drehteil montiert. Das Magnetfeld, welches durch den Dauermagneten und den Strom durch den Leiter verursacht wurde, übt nur eine Kraft in Drehrichtung des Rotors auf den Leiter aus, wenn sie in der richtigen Ausrichtung zueinander sind. Daraus folgt, dass der Strom während der Drehung umgekehrt wird. Dies wird auch Kommutierung genannt. Die Bürsten werden dazu benötigt, um einen Kontakt zwischen dem Stator und den Drähten des rotierenden Rotors herzustellen. Gebürstete Motoren haben den Vorteil, dass sie mit einer Gleichspannung – z.B. Batterie oder Akkumulator – betrieben werden können. Die Kommutierung mit den Bürsten sorgt dann für die richtige Durchflussrichtung. (Büchi, 2012: 9f)

Bei bürstenlosen Motoren sind die Permanentmagnete am rotierenden Teil, dem Rotor, und die stromführenden Leiter am Stator angebracht. Diese Situation ist daher genau umgekehrt wie bei bürstenbehafteten Motoren. Demnach muss der Strom nicht auf den Rotor übertragen werden und der Motor benötigt keine Bürsten. Der bürstenlose Motor hat jedoch den Nachteil, dass er nicht direkt mit einer Gleichspannung betrieben werden kann. Die Kommutierung muss daher auf eine andere Weise aufgelöst werden.

Betriebsverhalten eines Gleichstrommotors

Die Drehzahl-Drehmoment-Kennlinie eines Gleichstrom-Nebenschlussmotors weist einen linearen Verlauf auf, wie er qualitativ in Abbildung 41 dargestellt ist:

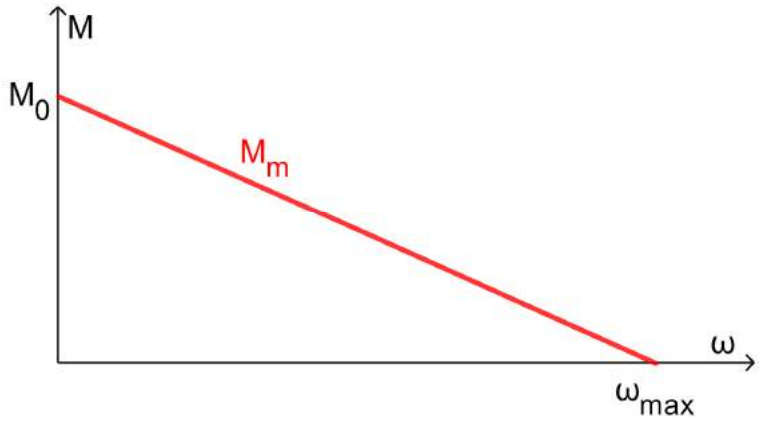


Abbildung 41: Abhängigkeit des Motormomentes von der Motordrehzahl

Wird der Motor bei Nennspannung ohne Lasten betrieben, bezeichnet man die dann maximal zu erreichende Drehzahl als Leerlaufdrehzahl ω_{max} . Sie wird nur durch die Reibung im Motor selbst beschränkt. Das maximale Motormoment bei $\omega = 0$ wird dann als Stillstandsmoment M_0 bezeichnet.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Dabei ist das Motor-Drehmoment M_m proportional zum Ankerstrom I. Leerlaufdrehzahl und Stillstandsmoment können durch Variation der angelegten Spannung verändert werden. Dies entspricht einer Parallelverschiebung der Kennlinie, wie in Abbildung 42 dargestellt wird:

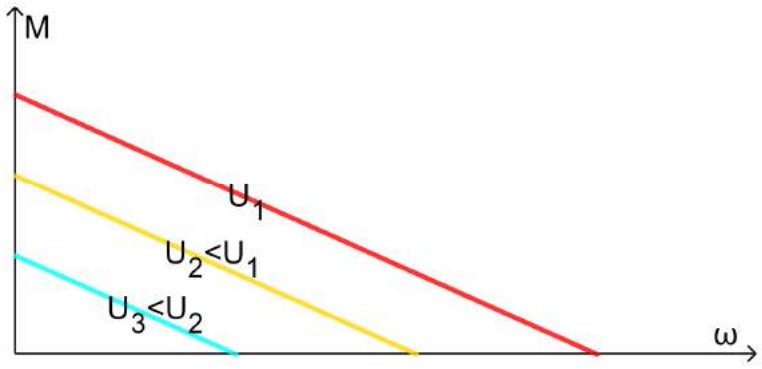


Abbildung 42: Parallelverschiebung der Kennlinie

Zum Beschleunigen wird eine positive Differenz aus Motormoment M_m (in Abhängigkeit der Drehzahl) und Belastungsmoment M_b, welches hier als konstant angenommen werden soll, benötigt (Formel 5.1).

$$\Delta M(\omega) = M_m(\omega) - M_b \tag{5.1}$$

Je größer diese Differenz ist, desto schneller beschleunigt der Motor. Der aktuelle Betriebspunkt kann sich dabei bei konstanter Spannung U nur entlang der in Abbildung 43 gezeigten Geraden bewegen. Wird er ungeregelt seiner eigenen Dynamik überlassen, nähert sich der Motor von z.B. einem Punkt P_1 in Abbildung 43 ausgehend dem Schnittpunkt von seiner Kennlinie mit der des konstanten Belastungsmoments (Punkt P_2) an. Da dabei die Momentendifferenz (dargestellt durch die blaue Fläche) und damit die Beschleunigung immer kleiner wird, kann der Schnittpunkt theoretisch nur im Unendlichen erreicht werden. Würde sich der Punkt P_1 rechts des Punktes P_2 befinden, wäre die Momentendifferenz negativ, was ein Abbremsen des Motors zur Folge hätte.

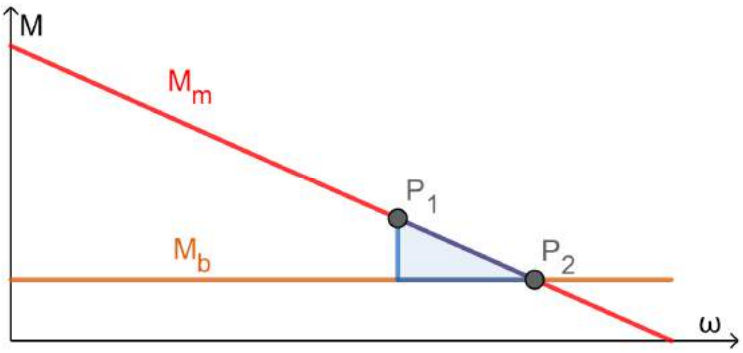


Abbildung 43: Momentendifferenz zwischen P_1 und P_2

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Berechnung der Beschleunigungszeit

Eine Masse mit dem Trägheitsmoment J soll nun innerhalb der Zeit dt um die Drehzahl $d\omega$ beschleunigt werden. Die Bewegungsgleichung lautet (Formel 5.2)

$$M_m(\omega) - M_b = J\frac{d\omega}{dt} \tag{5.2}$$

Die Separation der Variablen und Integration liefert (Formel 5.3).

$$\Delta t = J\int_{\omega_1}^{\omega_2}\frac{d\omega}{M_m(\omega) - M_b} \tag{5.3}$$

Das Motormoment lässt sich nach Abbildung 43 durch folgende Funktion ausdrücken:

$$M_m(\omega) = M_0 - \frac{M_0}{\omega_{max}}\omega \tag{5.4}$$

Nach Einsetzen des Motormoments erhält man für die Beschleunigungszeit (Formel 5.5):

$$\Delta t = J\int_{\omega_1}^{\omega_2}\frac{d\omega}{M_0 - \frac{M_0}{\omega_{max}}\omega - M_b} = -J\frac{\omega_{max}}{M_0}\ln(M_0 - \frac{M_0}{\omega_{max}}\omega - M_b)\Big|_{\omega_1}^{\omega_2} \tag{5.5}$$

Der Verlauf der Drehzahl über der Zeit wird qualitativ in Abbildung 44 dargestellt:

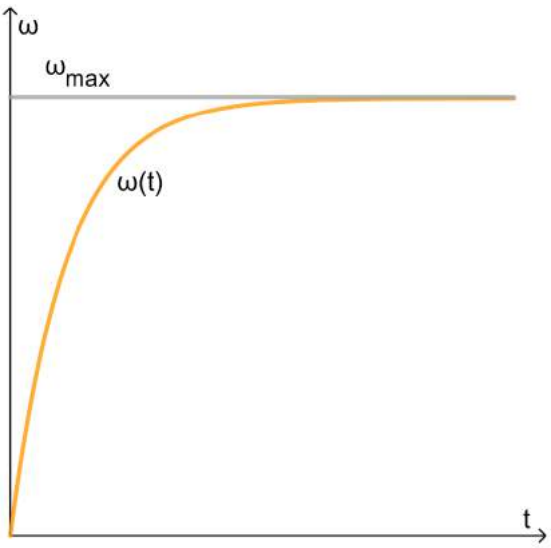


Abbildung 44: Verlauf der Drehzahl über der Zeit

Die Beschleunigungszeit kann damit analytisch oder numerisch errechnet werden.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Schaltplan

In Abbildung 45 ist der Schaltplan dargestellt.

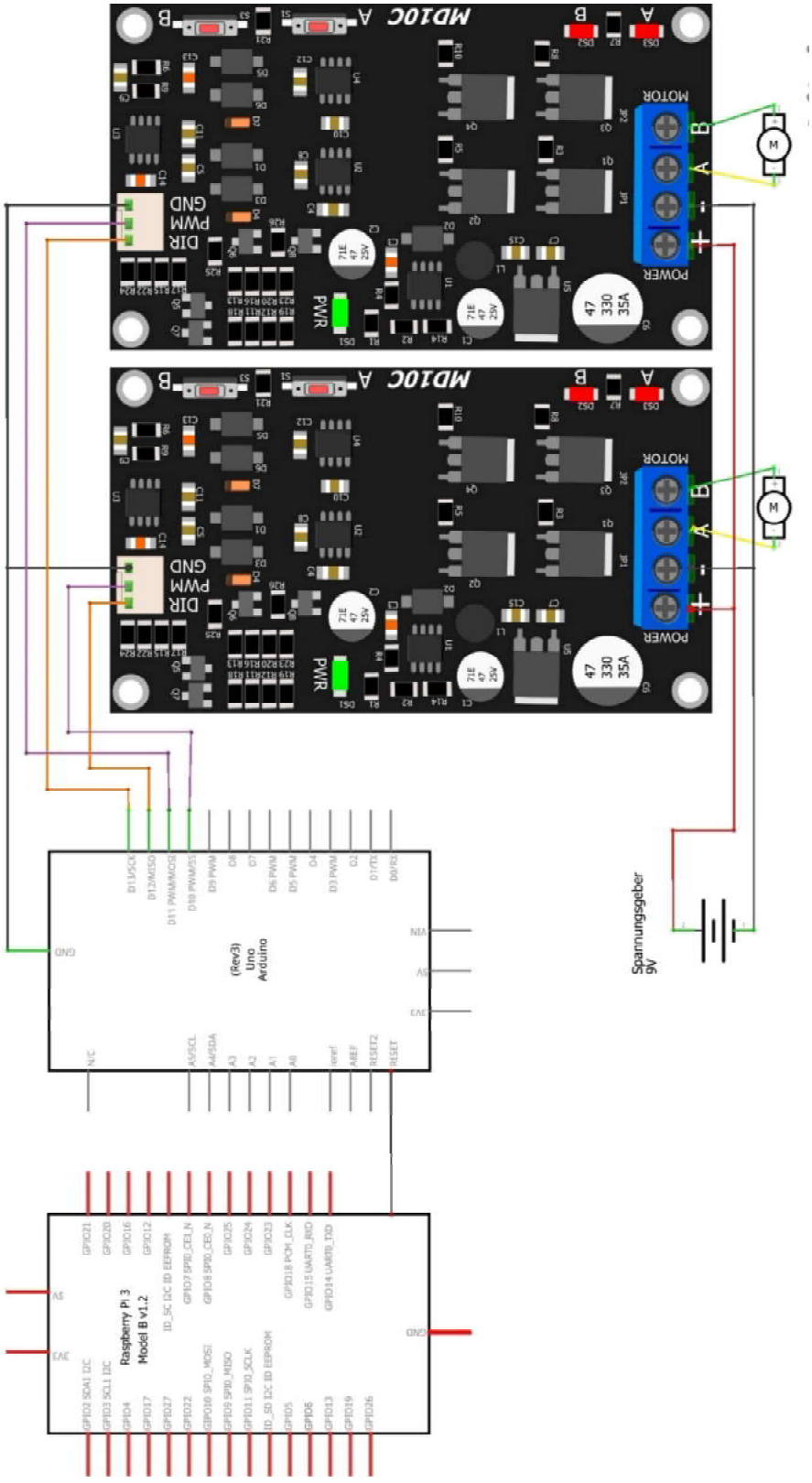


Abbildung 45: Gleichstrommotor-Aufbau – Schaltplan

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Verbindungsschema

In Abbildung 46 ist das Verbindungsschema des Versuchsaufbaus ersichtlich. Dabei bemerkt man, dass in der Abbildung eine 9 Volt Batterie abgebildet ist. Leider konnte ein 12 Volt Transformator nicht dargestellt werden. Deshalb soll dieser durch die eben genannte 9 Volt Batterie symbolisiert werden.

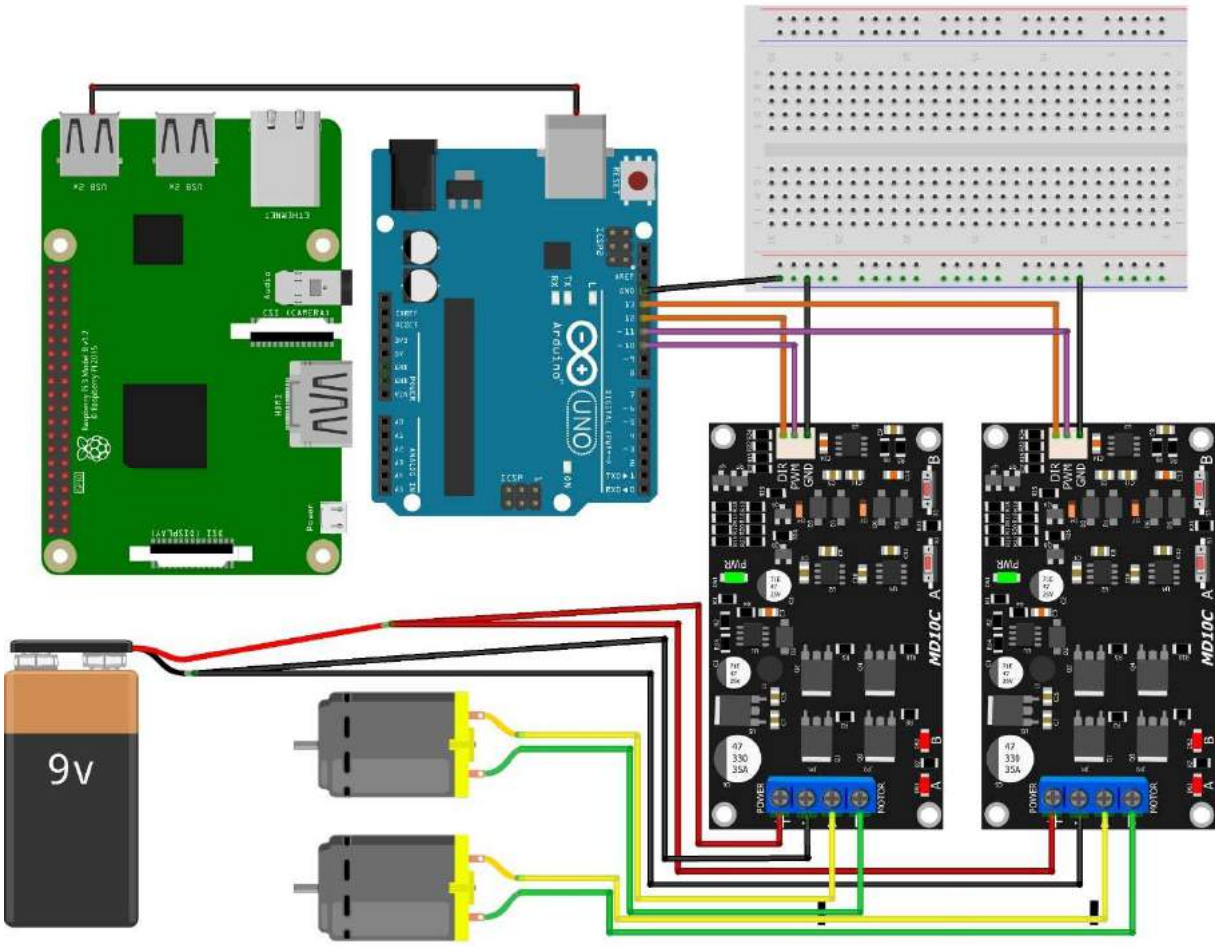


Abbildung 46: Gleichstrommotor-Aufbau – Verbindungsschema

In der graphischen Oberfläche am Raspberry PI kann durch einen *Slider* die Geschwindigkeit der Motoren eingestellt werden. Die Geschwindigkeit wird über einen sogenannten PWM-Wert („Pulse Width Modulation“), auch als Pulsweitenmodulation bezeichnet, gesteuert.

Bei diesem Verfahren wird durch Ein- und Ausschalten der Spannung am Motor die Drehzahl geregelt. Dabei spielen zwei Parameter eine wichtige Rolle, die Periode und Frequenz. Dabei ist diese Periode die Dauer eines Ein-Aus-Durchlaufs. Die Frequenz ist die Anzahl dieser Perioden in einer Sekunde.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Der Raspberry PI ist über eine serielle Schnittstelle mit einem Arduino verbunden. Dieser ist wiederum über Motortreibern mit den Motoren verbunden, welche auf die Einstellung der Slider reagieren.

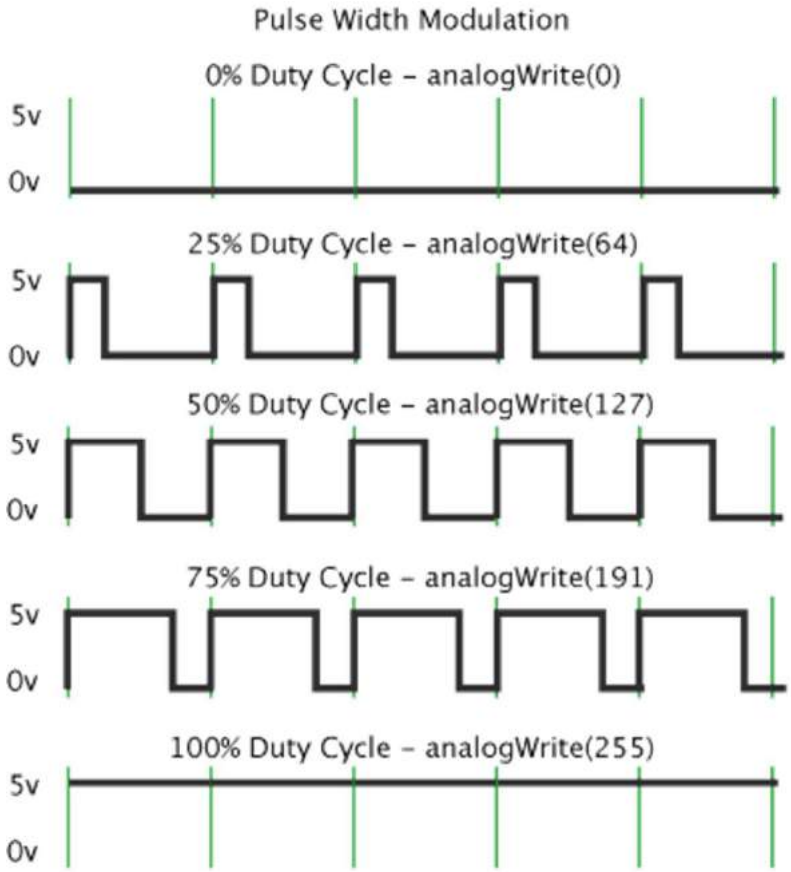


Abbildung 47: PWM-Werte für den DC-Motor

In Abbildung 47 ist das Handling der PWM-Werte mit dem Arduino abgebildet. Der Wert 255 stellt das Maximum dar, was in diesem Aufbau eine Spannung von 12 Volt am Motor bedeutet. Bei einem Wert von 191 (75%) würde sich der Motor wie bei einer konstanten Spannung von 9 Volt verhalten.

Die in der Abbildung 47 dargestellten PWM-Werte sind gerundete Zahlen, da diese nur ganzzahlig an die Motoren übermittelt werden können.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Versuchsaufbau

Abbildung 48 gibt den Versuchsaufbau wieder.

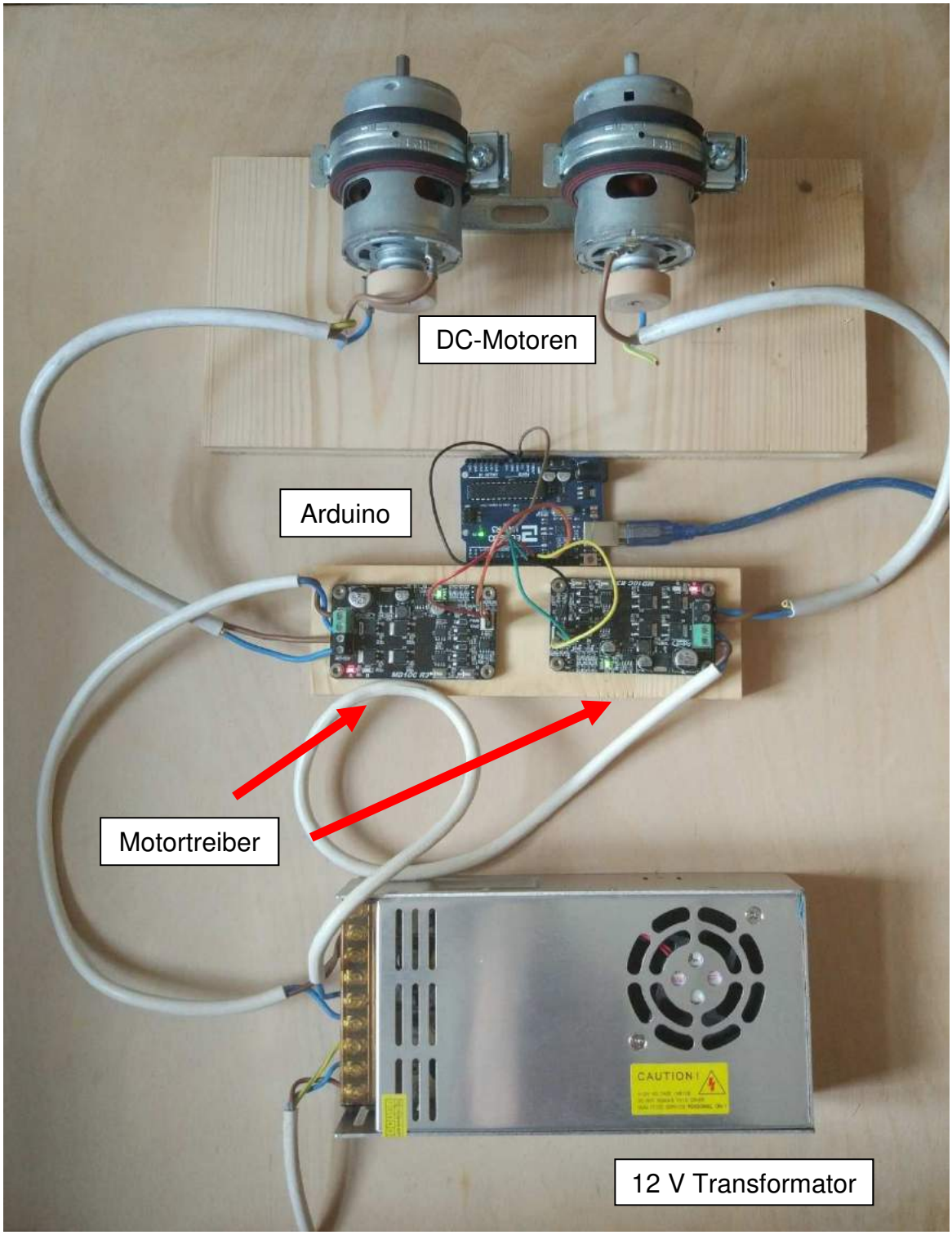


Abbildung 48: Gleichstrommotor-Aufbau – Versuchsaufbau

Leider war es aus Platzmangel nicht möglich, den Raspberry PI in der Abbildung darzustellen. Dieser ist per serieller Schnittstelle (blaues USB-Kabel) mit dem Arduino verbunden.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Code (Arduino)

Der Arduino Programmcode befindet sich im Anhang A.

Code (Raspberry PI)

Zusätzlich zum *Python*-Programm am Raspberry PI existiert die Datei „gleichstrommotor.ui“, in der die *Items* der graphischen Oberfläche gespeichert sind. Diese Programm-Oberfläche ist in Abbildung 49 dargestellt.

```
1.  app = widgets.QApplication(sys.argv)
```

Durch den Befehl `app = widgets.QApplication(sys.argv)` wird das zentrale Objekt von der Klasse *QApplication*, welches sich im Package *PyQt5.QtWidgets* befindet, abgeleitet. Bei der Erstellung bekommt das Objekt eine Liste der Argumente *sys.argv* mitgeliefert. Damit das Fenster permanent angezeigt wird, wird die Funktion *app.exec_()* aufgerufen, welche erst das Fenster schließt, wenn der Benutzer die Applikation beendet oder ein Fehler zum Abbruch des Programms führt.

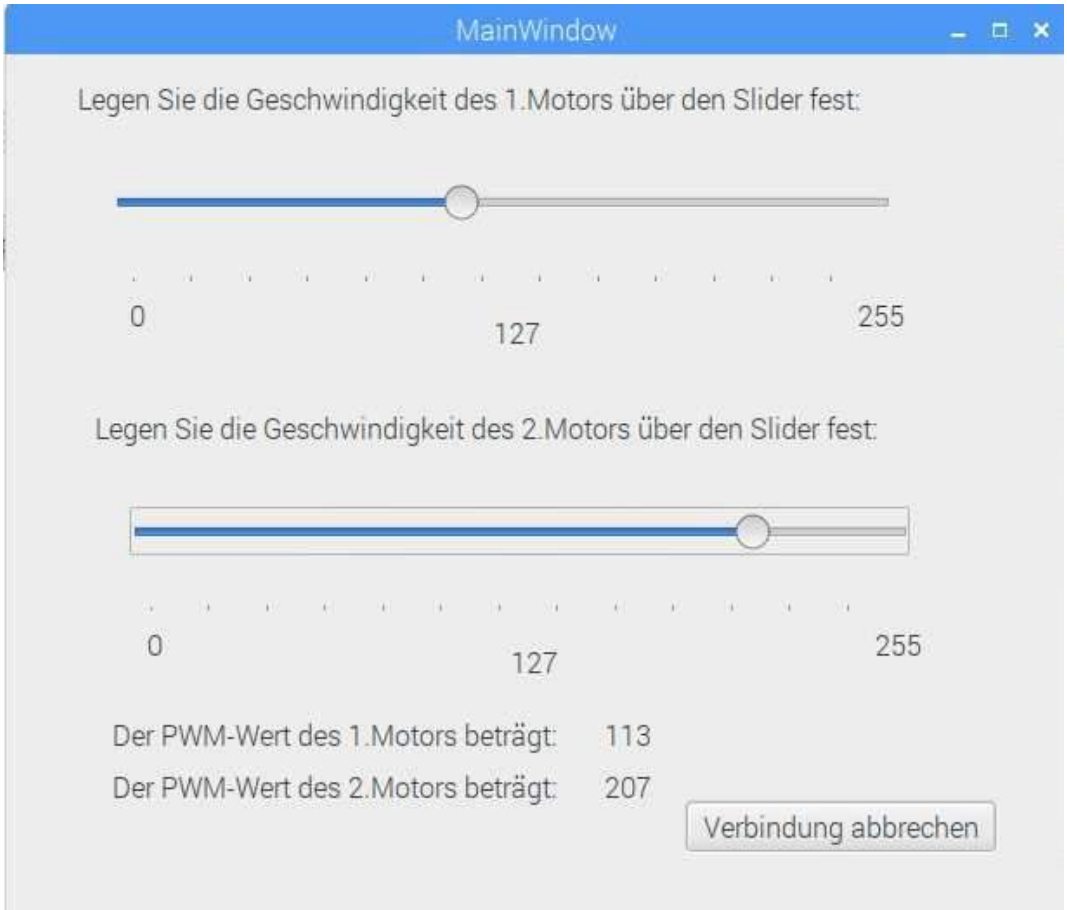


Abbildung 49: Gleichstrommotor-Aufbau – graphische Oberfläche

Beim Programmstart wird eine Verbindung vom Raspberry PI zu dem Arduino aufgebaut. Dieser Verbindung besteht solange, bis der Button „Verbindung abbrechen“ bestätigt wird.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

5.1.6 Antriebsdimensionierung

Berechnung der benötigten Winkelgeschwindigkeiten

Die Auslegung der DC-Motoren soll bei einer maximalen Ballgeschwindigkeit $v_B = 85\,\frac{km}{h}$ durchgeführt werden.

Unter der Annahme, dass es keine Verlustenergie durch Reibung und Wärme gibt, wird zunächst davon ausgegangen, dass auch die Schwungräder eine äußere Tangentialgeschwindigkeit von $v_R = 85\,\frac{km}{h}$ aufweisen.

Die Berechnung der äußeren Tangentialgeschwindigkeit erfolgt mit der Formel 5.6.

$$v_R = \omega_R \cdot r_R \tag{5.6}$$

Durch Umformung der Formel 5.6 nach der gesuchten Winkelgeschwindigkeit der Schwungräder ω_R und Einsetzen des Radius des Schwungrades $r_R = 0{,}07\,m$ ergibt sich

$$\boldsymbol{\omega_R} = \frac{v_R}{r_R} = \frac{85\,\frac{km}{h}}{0{,}07m} = \boldsymbol{3218{,}11\,\frac{U}{min}}$$

Die vorhin genannten Verluste werden mit 25% der Winkelgeschwindigkeit ω_R abgeschätzt. Demnach ergibt sich für die benötigte Winkelgeschwindigkeit der Schwungräder ω_{R_V} folgendes:

$$\boldsymbol{\omega_{R\_V}} = \omega_R + \omega_V = \omega_R + 0{,}25 \cdot \omega_R = \boldsymbol{4022{,}64\,\frac{U}{min}}$$

Annäherung der benötigten Winkelgeschwindigkeiten mithilfe der Stoßgleichungen

Eine weitere Annäherung der benötigten Winkelgeschwindigkeiten der Schwungräder soll mithilfe der Gleichungen des exzentrischen, teilelastischen Stoßes erreicht werden. Der Spin wird dabei vorerst vernachlässigt.

Es werden folgende Größen als konstant angenommen:

$I_R\ (kgm^2)$	… Massenträgheitsmoment eines Schwungrades
$m_B\ (kg)$	… Masse des Balles
$d_R\ (m)$	… Durchmesser des Schwungrades
$k\ (1)$	… Stoßzahl
$h\ (m)$	… Abstand der Außenflächen der Schwungräder

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Die weiteren Variablen sind:

$v_B / v_B{}' \ (m/s)$ … Abschussgeschwindigkeit des Balles vor bzw. nach dem Kontakt mit den Schwungrädern
$\omega_{Ro} / \omega_{Ru} \ (rad/s)$ … Winkelgeschwindigkeit des oberen bzw. unteren Schwungrades vor dem Kontakt mit dem Ball
$\omega_{Ro}{}' / \omega_{Ru}{}' \ (rad/s)$ ... Winkelgeschwindigkeit des oberen bzw. unteren Schwungrades nach dem Kontakt mit dem Ball
$S_o / S_u \ (Ns)$ … Impuls am oberen bzw. unteren Kontaktpunkt

Es wird angenommen, dass für die Geschwindigkeit des Balles vor dem Kontakt gilt:

$$v_B = 0$$

Die Drehimpulsbilanzen am oberen bzw. unteren Rad liefern dann

$$I_R \omega'_{Ro} - I_R \omega_{Ro} = -\frac{d_R}{2} S_o \tag{5.7}$$

$$I_R \omega'_{Ru} - I_R \omega_{Ru} = -\frac{d_R}{2} S_u \tag{5.8}$$

Impulsbilanz bzw. Drehimpulsbilanz am Ball liefern

$$m_B v'_B = S_o + S_u \tag{5.9}$$

$$0 = \frac{h}{2} S_o - \frac{h}{2} S_u \tag{5.10}$$

An den Kontaktpunkten folgen die Gleichungen

$$v'_B - \omega'_{Ro} \frac{d_R}{2} = k \omega_{Ro} \frac{d_R}{2} \tag{5.11}$$

$$v'_B - \omega'_{Ru} \frac{d_R}{2} = k \omega_{Ru} \frac{d_R}{2} \tag{5.12}$$

Durch Lösen des Gleichungssystems ergeben sich folgende Gleichungen:

$$\omega_{Ro} = \omega_{Ru} = \frac{(8 I_R + m_B d_R^2)}{4 I_R d_R (1 + k)} v_B{}' \tag{5.13}$$

$$\omega'_{Ro} = \omega'_{Ru} = \frac{(8 I_R - k m_B d_R^2)}{4 I_R d_R (1 + k)} v_B{}' \tag{5.14}$$

Dieser Ansatz wird aufgrund der Komplexität und der unbekannten Variablen (z.B. Stoßzahl), welche nur erahnt werden können, verworfen.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

5.1.7 Drehzahlmessung-Aufbau

Aufgabenstellung

Die Soll-Drehzahl der Motoren wird, wie in einem früheren Test-Aufbau beschrieben, über die PWM-Werte vorgegeben. Auch in diesem Aufbau soll über Eingabe der PWM-Werte am Raspberry PI die Drehzahl für die Motoren vorgegeben werden. Der Raspberry PI ist in diesem Fall wieder über eine serielle Schnittstelle mit dem Arduino verbunden. Zusätzlich soll am Arduino eine Drehzahlmessung des Motors, welche mit einem Hall-Sensor realisiert werden soll, eingebaut werden. Die Drehzahl des Motors soll auf einem LCD-Display wiedergegeben werden und viermal in der Sekunde aktualisiert werden.

Benötigte Bauteile

(1) x Raspberry Pi 3 Model B+
(1) x Arduino UNO R3
(1) x USB-Kabel
(1) x 600W Netzteil 12V/50A
(1) x Motortreiber Cytron 13A
(1) x Gleichstrommotor RS PRO 12V
(1) x LCD-Display 4x20
(1) x LCD-I²C Modul
(1) x Hall Sensor Modul
(1) x Magnet

Einführung in die Komponenten

Hall-Sensor
Der Hall-Effekt-Sensor arbeitet nach dem Prinzip des Hall-Effekts. Die Erkenntnis des Physikers Edwin Hall war, dass eine elektrische Spannung induziert wird, wenn sich ein stromdurchflossener Leiter in einem Magnetfeld befindet. Dabei fällt die Spannung senkrecht zur Richtung des Stromflusses ab. (Dullinger, 2019)

Diese Spannungsänderung dient nun als Indikator dafür, ob sich der Sensor in der Nähe eines Magneten befindet oder nicht. Mit Hilfe des Arduino oder eines anderen Mikrocontrollers kann diese Spannungsänderung ausgewertet werden. Beim Arduino bietet sich dafür einer der Interrupt-Pins an. Die grundlegende Funktionsweise eines an den Arduino angeschlossenen Hall-Effekt-Sensors ist in dem Blockbild in Abbildung 50 dargestellt.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

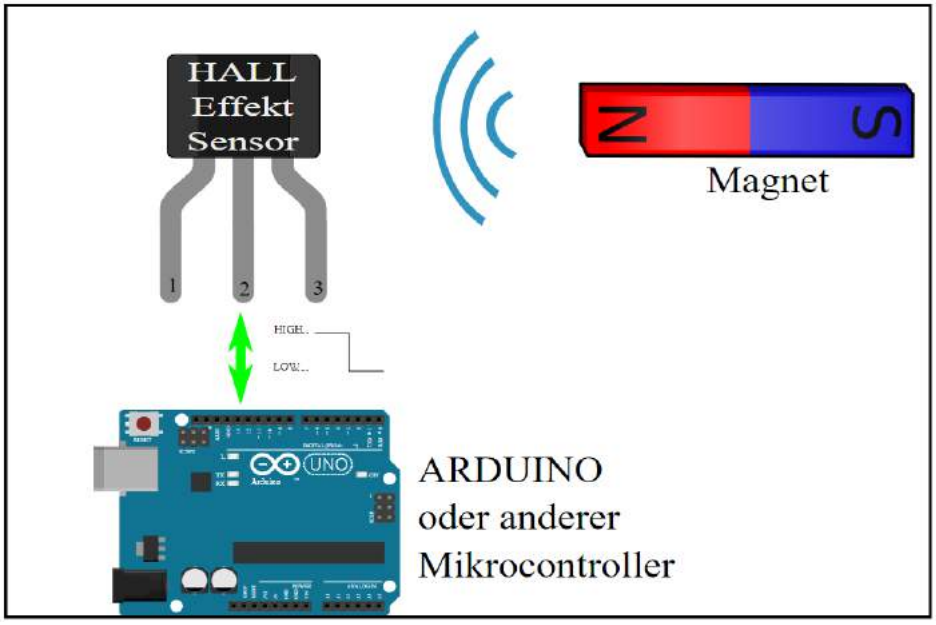


Abbildung 50: Arbeitsweise des Hall-Effekt-Sensor am Arduino (Dullinger, 2019)

In diesem Aufbau wird ein Hall-Sensor Modul verwendet, welches speziell für Projekte mit einem Arduino oder Raspberry PI entwickelt wurde. Das Hall-Sensor Modul besitzt 4 PINs. Die PIN-Belegung ist in Abbildung 51 dargestellt.

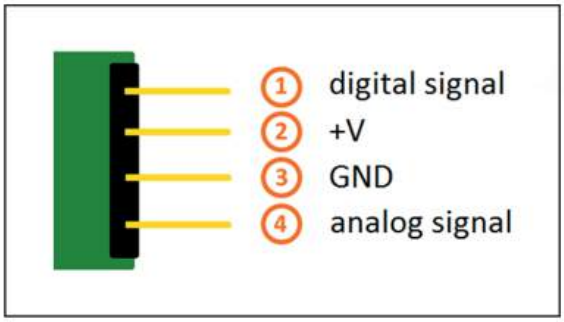


Abbildung 51: PIN-Belegung des Hall-Sensor Moduls

Des Weiteren besitzt das Modul ein Potentiometer, bei dem die Sensibilität des Sensors eingestellt werden kann. Dieser ist in Abbildung 52 dargestellt.

Abbildung 52: Potentiometer für Sensibilität des Hall-Sensors

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Alternativ kann die Drehzahl auch mithilfe einer Lichtschranke gemessen werden. Diese Methode basiert auf einer rotierenden Loch- oder Kontrastscheibe, wobei mit einer IR-Lichtschranke oder Reflexionslichtschranke Rechteckimpulse erzeugt werden. Mithilfe der Anzahl gezählter Impulse, die innerhalb einer bestimmten Zeit festgehalten wurde, kann die Drehzahl ermittelt werden. (Restian, 2019)

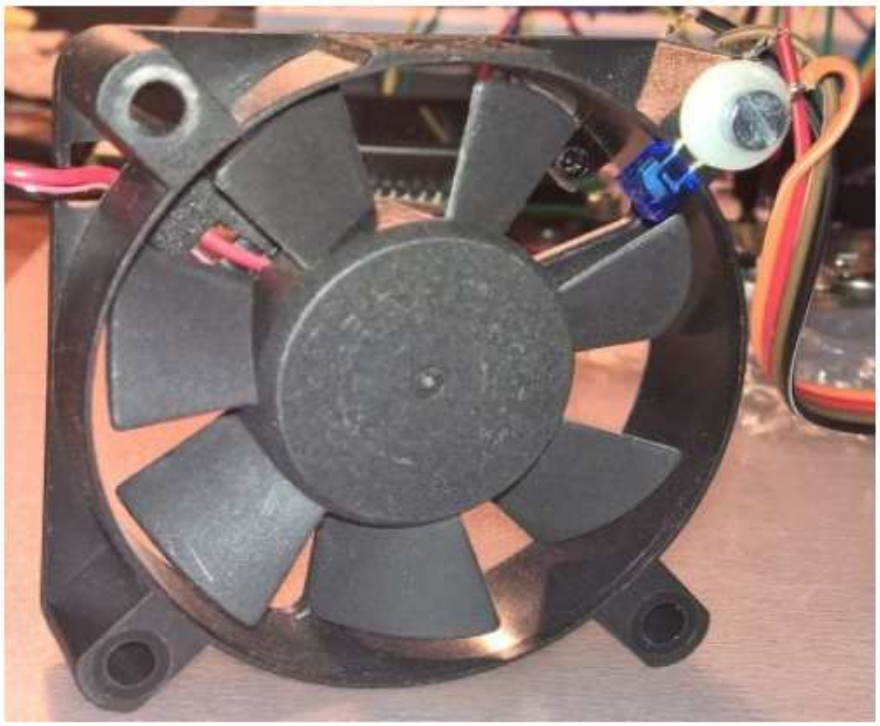


Abbildung 53: Drehzahlmessung mittels einer Lichtschranke (Restian, 2019)

In Abbildung 53 ist ein Lüfter eines Computers mit einer angebrachten Lichtschranke abgebildet. In diesem Beispiel ist zu beachten, dass bei einer Umdrehung die Lichtschranke sieben Zwischenräume wahrnimmt. Dies muss natürlich bei der Berechnung der Drehzahl am Arduino berücksichtigt werden.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Schaltplan

In Abbildung 54 ist der Schaltplan dargestellt.

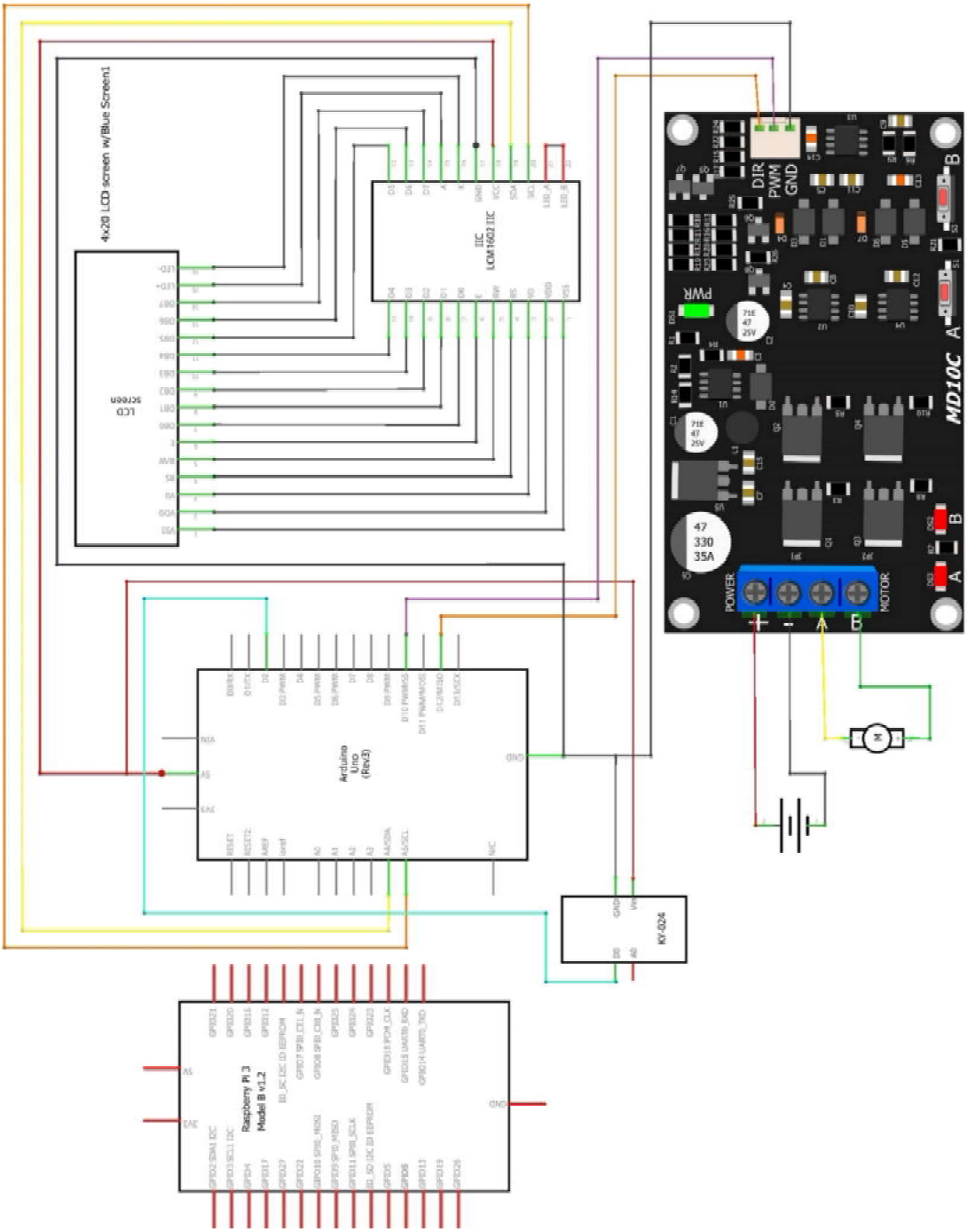


Abbildung 54: Drehzahlmessung-Aufbau– Schaltplan

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Verbindungsschema

Das Verbindungsschema ist in Abbildung 55 abgebildet.

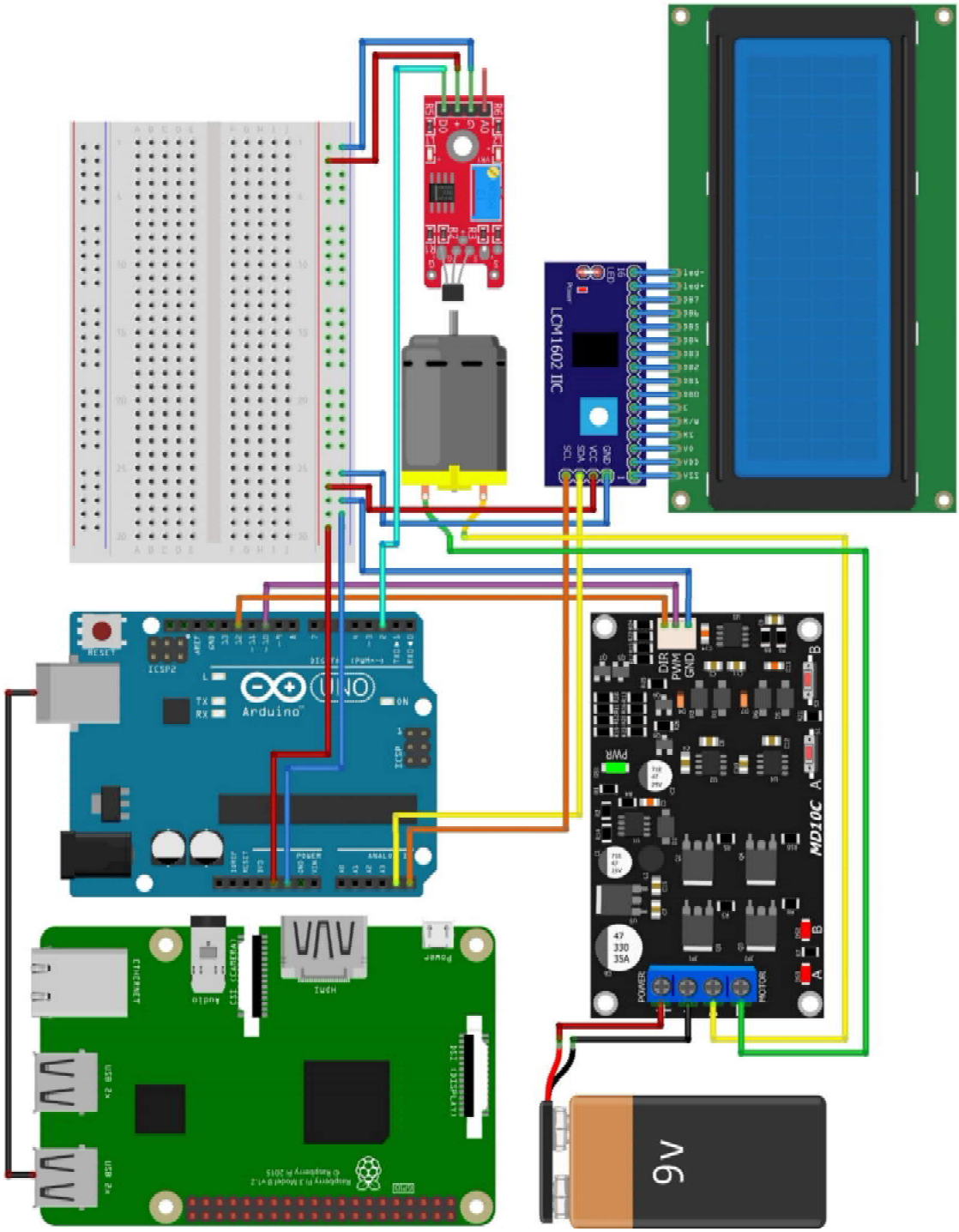


Abbildung 55: Drehzahlmessung-Aufbau – Verbindungsschema

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Versuchsaufbau

Auf dem Motor ist ein kleines Rad mit einem Sackloch (d = 4 mm) montiert. In diesem Sackloch wurde ein kleiner Magnet angebracht. Der Hall-Sensor wurde auf Höhe dieses Magnets eingestellt um eine korrekte Messung durchführen zu können. Bei jeder Umdrehung bzw. jedes Mal, wenn der Magnet den Hall-Sensor passiert, wird dieser Zustand an den Arduino weitergeleitet. Dies ist in Abbildung 56 ersichtlich.

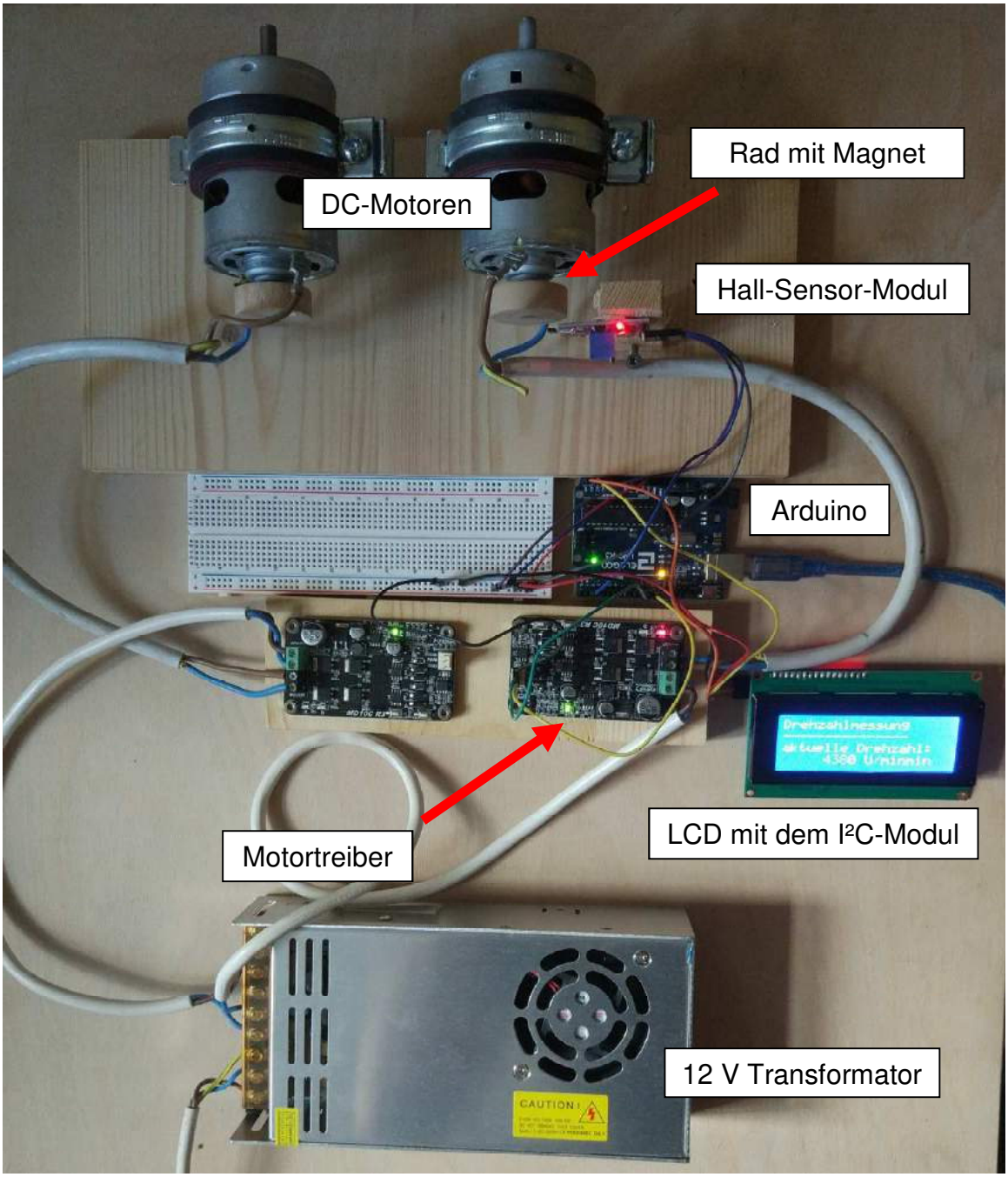


Abbildung 56: Drehzahlmessung-Aufbau – Versuchsaufbau

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Auf dem LCD-Display wird die aktuelle Drehzahl ausgegeben (siehe Abbildung 57).

Abbildung 57: Drehzahlmessung-Aufbau – LCD-Display

Code (Arduino)

Der Arduino Programmcode befindet sich im Anhang A.

Code (Raspberry PI)

Der Raspberry PI Programmcode befindet sich ebenfalls im Anhang A.

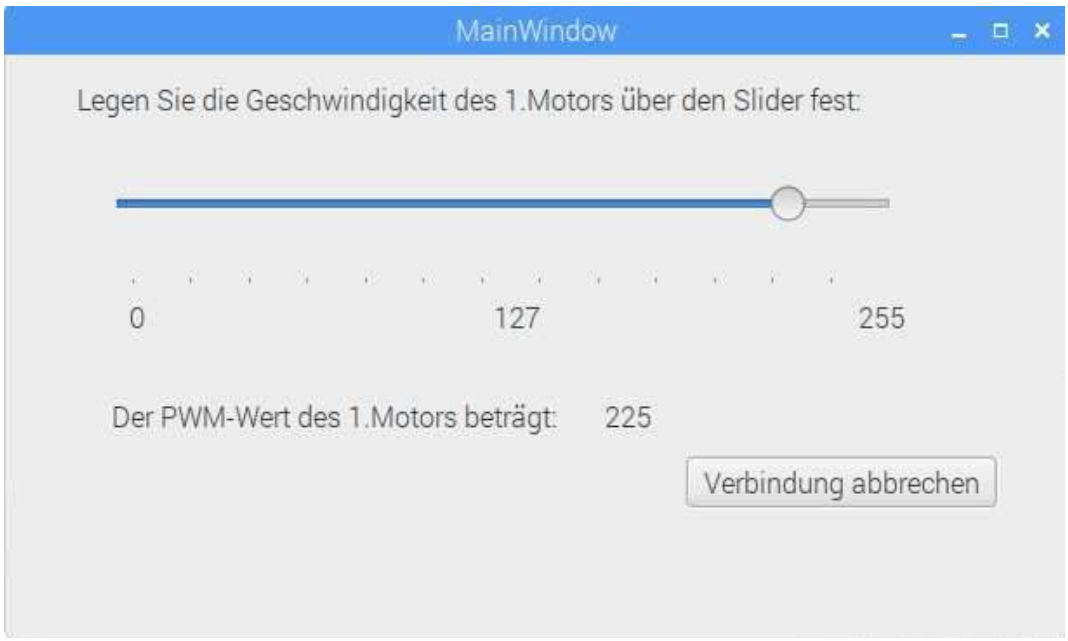


Abbildung 58: Drehzahlmessung-Aufbau – graphische Oberfläche

Die in der Abbildung 58 dargestellte graphische Oberfläche besitzt die gleiche Funktionsweise zur Bedienung der Elemente, wie die des vorherigen Versuchsaufbaus.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

5.1.8 Servomotor-Aufbau

Aufgabenstellung

Dieser Aufbau stellt eine Erweiterung des Versuchsaufbaus 5.1.5 dar. Dieser besitzt zusätzlich zu den beiden Gleichstrommotoren, welche über eine graphische Oberfläche am Raspberry PI gesteuert werden, noch einen Schrittmotor. Dieser Schrittmotor wird benötigt um die spätere Ballmaschine auf einen bestimmten Winkel einzustellen, damit die Bälle auf dem gesamten Platz verteilt werden können.

Benötigte Bauteile

(1) x Raspberry Pi 3 Model B+
(1) x Arduino UNO R3
(1) x USB-Kabel
(1) x 600W Netzteil 12V/50A
(2) x Motortreiber Cytron 13A
(2) x Gleichstrommotor RS PRO 12V
(1) x Servomotor SPRINGRC SM-S8166M

Einführung in die Komponenten

Servomotor
Servomotoren sind bürstenlose Gleichstrom– oder Drehstrommaschinen. Sie benötigen leistungselektronische Stellglieder in Form von Pulsstellern für Gleich– bzw. für Drehstrom. Diese werden umgangssprachlich auch als Servoverstärker, Servoregler oder Antriebsregelgerät bezeichnet. (Probst, 2011: 1)

Servomotoren basieren auf einem drehzahlvariablen elektromotorischen Antrieb, der meist geregelt betrieben wird. Der Sensor zur Positionsbestimmung übermittelt die aktuelle Drehposition der Motorwelle kontinuierlich an eine Regelelektronik (Servoregler), der die Bewegung des Motors anhand der eingestellten Soll-Winkelposition regelt. Servomotoren können – wie auch Schrittmotoren – schrittweise geregelt werden. Der Unterschied besteht allerdings darin, dass der Servomotor mithilfe des Sensors stets über seine Position Bescheid weiß. (Probst, 2011: 1f)

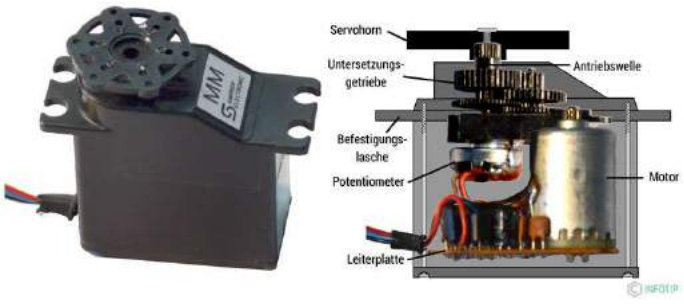


Abbildung 59: Aufbau eines RC-Servos (Kompendium, 2019)

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Schaltplan

Der Schaltplan ist in Abbildung 60 ersichtlich.

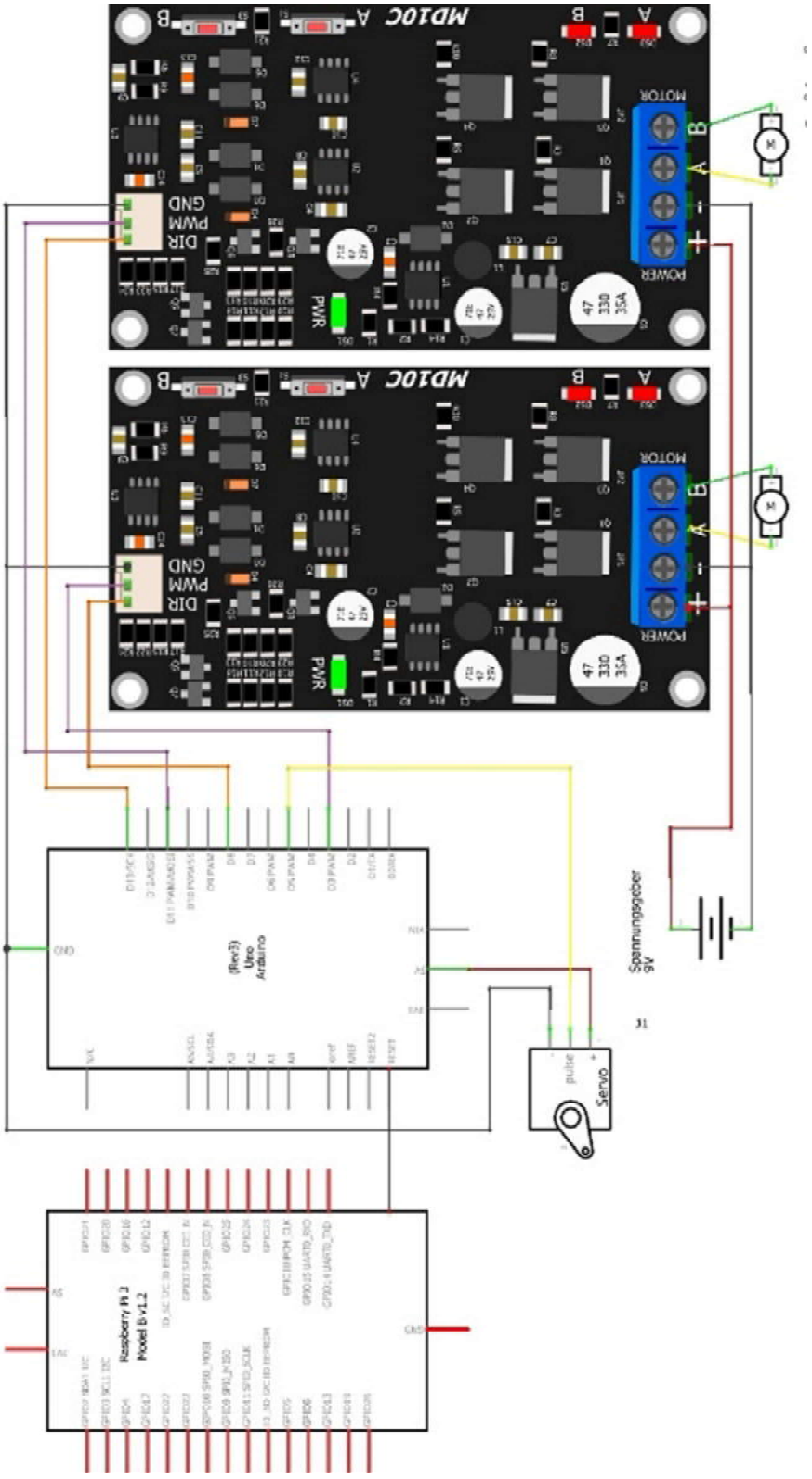


Abbildung 60: Servomotor-Aufbau – Schaltplan

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Verbindungsschema

Auch in diesem Fall ist eine 9 Volt Batterie anstelle des 12 V Transformators dargestellt (siehe Abbildung 61).

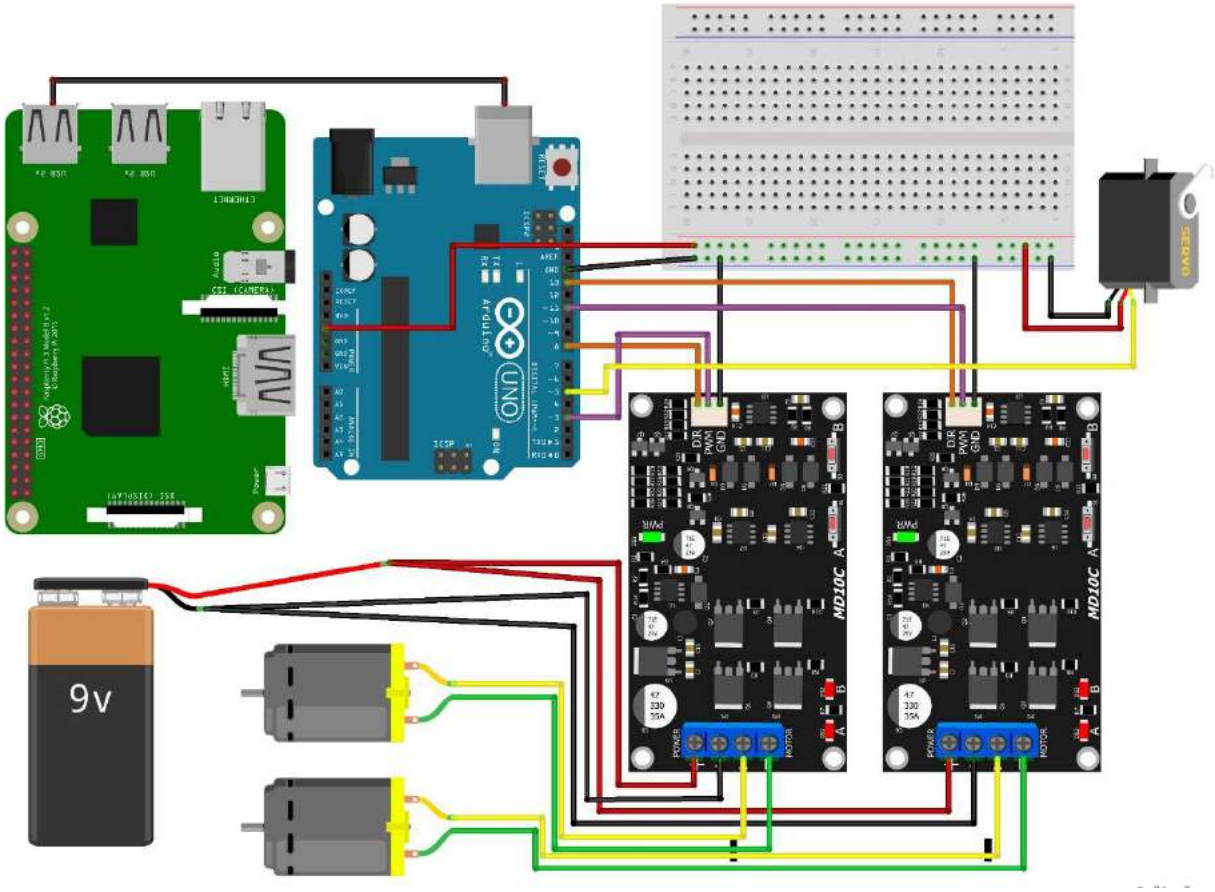


Abbildung 61: Servomotor-Aufbau – Verbindungsschema

Der Servomotor besitzt drei PINs mit denen der Motor versorgt wird. Die PIN-Belegung dazu lautet:

- Schwarzes Kabel → GND
- Rotes Kabel → 5V Kontakt am Arduino
- Gelbes/Weißes Kabel → digitaler Ausgang D5

In der graphischen Oberfläche am Raspberry PI kann mittels eines *Sliders* die Geschwindigkeit der Motoren eingestellt werden. Zusätzlich kann über einen dritten *Slider* die Winkelposition des Servomotors modifiziert werden.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Versuchsaufbau

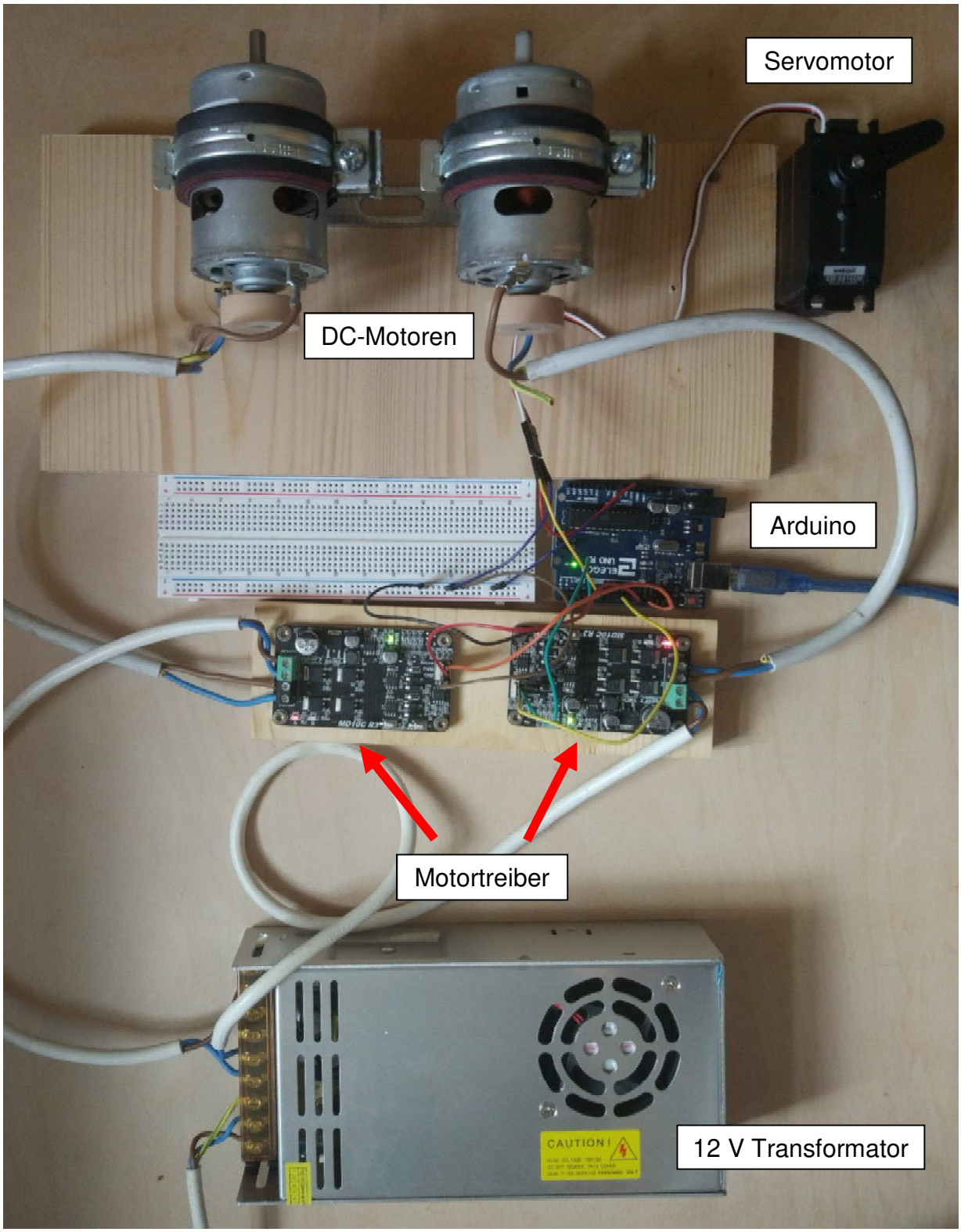


Abbildung 62: Servomotor-Aufbau – Versuchsaufbau

Auch hier ist der Raspberry PI nicht in der Aufnahme (Abbildung 62) abgebildet. Dieser ist per serieller Schnittstelle (blaues USB-Kabel) mit dem Arduino verbunden.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Code (Arduino)

Im Programmcode des Arduinos ist zusätzlich zum Code des Versuchsaufbaus *Gleichstrommotor-Aufbau* die Funktion `void servo_einstell()` ergänzt worden. In dieser Funktion wird dem Servomotor der Winkel zugewiesen, um eine bestimme Position einzunehmen.

```
void servo_einstell() //empfangene Daten verarbeiten
{
    Data[data_count] = '\0'; //String ende
    uint16_t zahl = atoi(Data); //String in Int umwandeln

    myservo.write(zahl, 20, true); //Servo an "zahl" fahren

    Serial.print("Neuer Wert fuer Servo: ");  //Return an den Raspberry PI
    Serial.println(zahl);

    clearData();
}
```

Der gesamte Programmcode des Arduinos ist im Anhang A ersichtlich.

Code (Raspberry PI)

Der Raspberry PI Programmcode befindet sich im Anhang A.

Zusätzlich zum *Python*-Programm am Raspberry PI wurde die Datei „servomotor.ui“, in der die Items der graphischen Oberfläche gespeichert sind, geladen.

Die Oberfläche des Programms ist in Abbildung 63 abgebildet.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

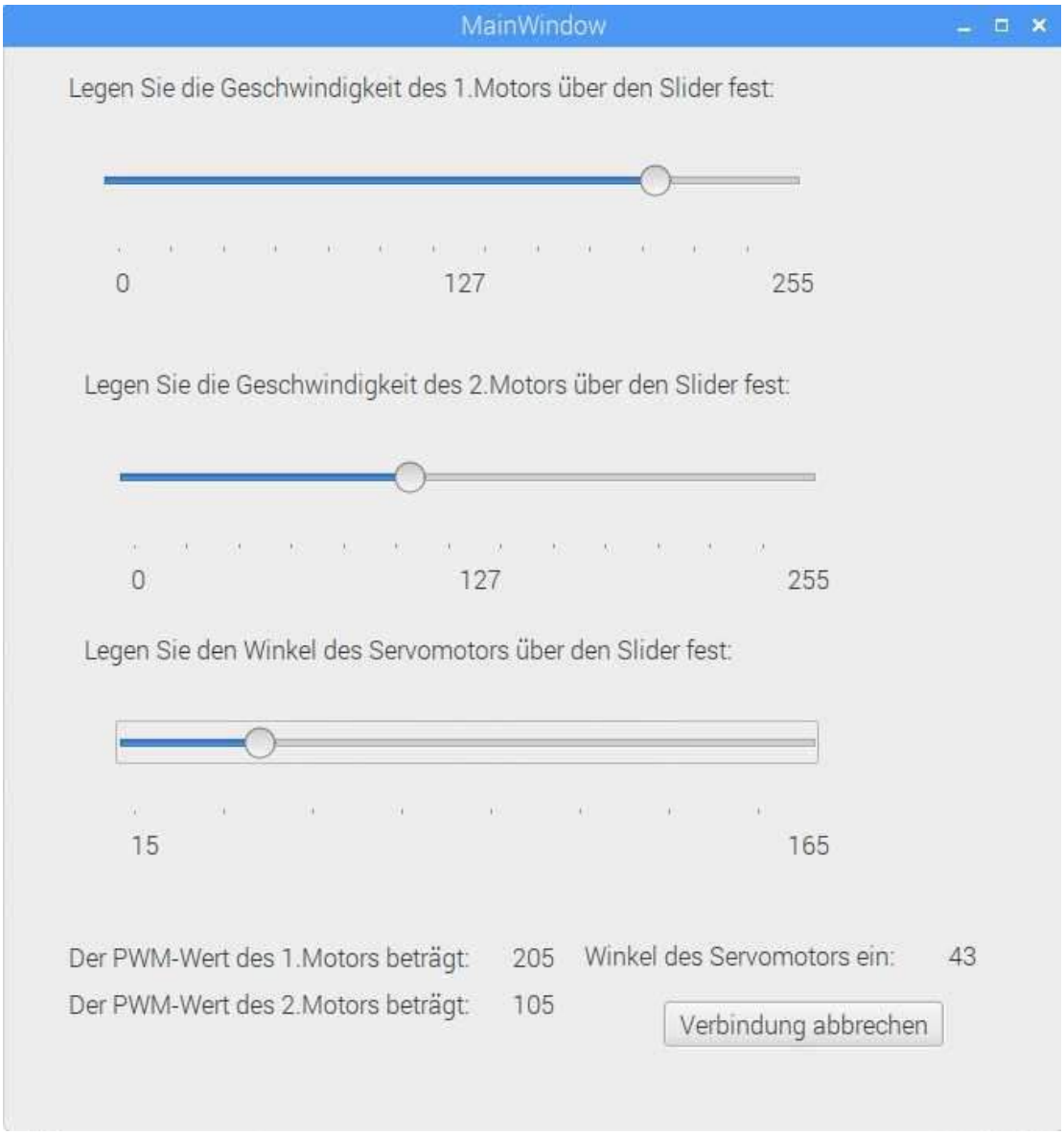


Abbildung 63: Servomotor-Aufbau – graphische Oberfläche

Bei Programmstart wird eine Verbindung vom Raspberry PI zum Arduino aufgebaut.

Durch eine Veränderung der ersten beiden *Slider* ändern sich die PWM-Werte der beiden Motoren und sie reagieren auf diese Änderungen. Beim Bewegen des dritten *Sliders* verändert sich der Winkel des Servomotors.

Die aktuellen Werte, welche über die drei *Slider* eingestellt wurden, werden bei den unteren *Labels* angezeigt.

Bei Betätigung des *Buttons* „Verbindung abbrechen“ werden die PWM-Werte der beiden Gleichstrommotoren auf 0 und der Winkel des Servomotors auf 15° gestellt. Zusätzlich wird die Verbindung zu Arduino beendet.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

5.1.9 Schrittmotor-Aufbau

Aufgabenstellung

Bei diesem Aufbau soll der Umgang mit einem Schrittmotor sowie die Kommunikation zwischen einem Schrittmotor und einem Hall-Sensor demonstriert werden. Dabei wird der Schrittmotor, an dessen verlängertem Arm sich ein Magnet befindet, sich eine definierte Anzahl von Schritten in eine bestimmte Richtung bewegen. Wenn der Hall-Sensor meldet, dass sich der Magnet auf einen bestimmten Abstand genähert hat, stoppt der Motor. Danach fährt dieser auf eine durch Zufall bestimmte Position. Umgesetzt wird diese Aufgabenstellung mit einem Arduino und dem dazugehörigen Schrittmotor-Treiber.

Benötigte Bauteile

(1) x Arduino UNO R3
(1) x USB-Kabel
(1) x 600W Netzteil 12V/50A
(1) x Schritt-Motortreiber 4A 9-42V (2/4 Phasen)
(1) x Hybrid-Schrittmotor Nema 23 (QSH6018-65-28)
(1) x Hall Sensor Modul
(1) x Magnet

Einführung in die Komponenten

Schrittmotor
Ein Schrittmotor ist ein elektromechanisches Gerät, welches elektrische Pulse in diskrete mechanische Bewegungen umwandelt. Die Axe des Motors dreht in kleinen mechanischen Schritten, wenn ein passender elektrischer Puls an ihn gesendet wird. Demnach dreht sich der Motor bei jeder Spulenumpolung einen Schritt weiter. Die Rotation des Motors ist abhängig von dem Eingangssignal bzw. den Pulsen. Die Sequenz der Pulse bestimmt direkt die Richtung der Rotation, während die Länge der Rotation von der Anzahl und Frequenz der Pulse abhängt.

Eine der größten Vorteile von Schrittmotoren ist, dass sie präzise in einem Schleifensystem gesteuert werden können.

Abbildung 64: Schrittmotor (Wikipedia, 2019b)

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Prinzipiell gibt es drei Typen von Schrittmotoren:

- Reluktanz Schrittmotor
  - Bei einem Reluktanz Schrittmotor besteht der Rotor aus einem gezahnten Weicheisenkern, welcher keine magnetischen Pole ausbildet. Demnach hat dieser Typ kein Rastmoment.
  - Vorteil: genauer Schrittwinkel
  - Nachteil: kleines Drehmoment

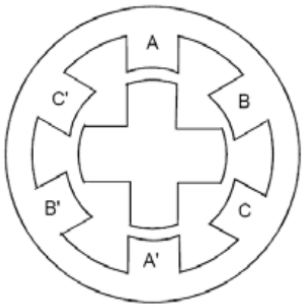


Abbildung 65: Prinzipskizze eines Reluktanz-Schrittmotors (Schaad, 2019)

- Permanentmagnet-Schrittmotor
  - Bei einem Permanentmagnet-Schrittmotor besteht der Rotor aus einem zylindrischen Permanentmagneten mit radialer Magnetisierung.
  - Vorteil: großes Rotor-Trägheitsmoment
  - Nachteil: magnetische Verluste

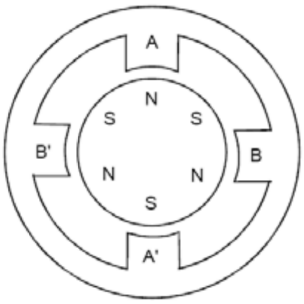


Abbildung 66: Prinzipskizze eines Permanentmagnet-Schrittmotors (Schaad, 2019)

- Hybrid-Schrittmotor
  - Der Hybrid-Schrittmotor vereint die Vorzüge beider Bauformen. Der Rotor besteht aus einem axialen Permanentmagneten, an dessen Enden gezahnte Kappen befestigt sind. Beide sind um eine halbe Zahnbreite gegeneinander versetzt, so dass sich Nord- und Südpole abwechseln.

Des Weiteren wird oft nach der Ansteuertechnik unterschieden. In diesem Fall kann zwischen unipolaren und bipolaren Motoren unterschieden werden:

- Unipolare Motoren
  - Diese Arten von Motoren besitzen zwei Spulen mit Mittelabgriff, welche meist über fünf oder sechs Anschlüsse verfügen.
  - Die Ansteuerung dieses Motors erfolgt durch wechselweises Einschalten von jeweils einem Spulenende, wodurch immer nur eine halbe Spule durch Strom durchflossen wird.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

- Bipolare Motoren
  - Ein bipolarer Motor hat meist zwei Spulen (vier Anschlüsse), bei welchem immer die gesamte Wickelung bestromt wird.
  - Manchmal besitzt dieser aber auch zwei Spulenpaare, welche durch Umpolen angesteuert werden. In diesem Fall besitzt der Motor acht Anschlüsse, welche auch unipolar angesteuert werden können.

In Abbildung 67 ist die Prinzipskizze eines unipolaren (a) und eines polaren Schrittmotors dargestellt.

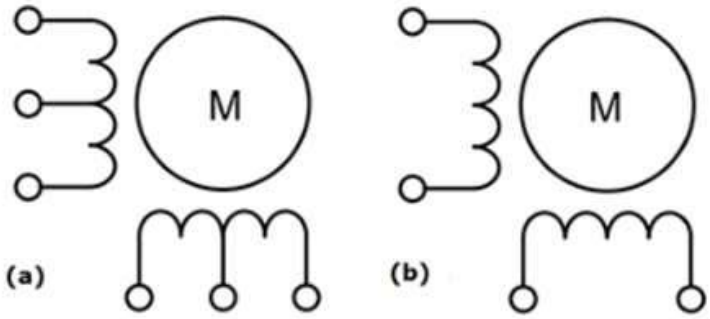


Abbildung 67: Prinzipskizze eines unipolaren (a) und polaren Schrittmotors (b) (Mark, 2019)

Eine Besonderheit, welche manche Schrittmotor-Treiber besitzen, ist der Mikro-Schritt-Betrieb. Bei diesem Betrieb kann der „normale“ Schrittwinkel des Motors in viel kleinere Winkel unterteilt werden. Diese besagten Mikroschritte werden erzeugt, indem der Strom in den zwei Spulen entsprechend einer Sinus- und Cosinus-Funktion fließt. Ein wesentlicher Vorteil dieses Betriebs besteht darin, dass der Motor sehr viel genauer arbeiten kann. In diesem Betrieb nehmen auch die Vibrationen und der Lärm stark ab.

In Abbildung 68 ist die Funktionsweise eines Schrittmotors im Mikro-Schritt-Betrieb dargestellt. Dabei ist bei der Spule A (blaue Linie) eine cosinusähnliche Funktion und bei der Spule B (rote Linie) eine sinusähnliche Funktion erkennbar.

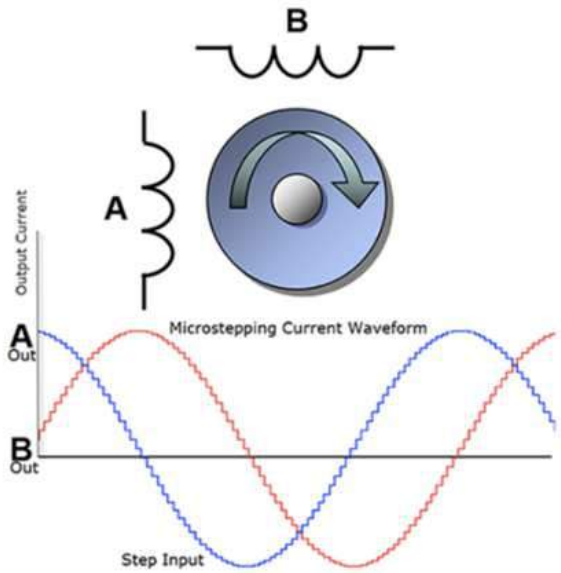


Abbildung 68: Funktionsweise im Mikro-Schritt-Betrieb (Mark, 2019)

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Schaltplan

Der Schaltplan ist in Abbildung 69 ersichtlich.

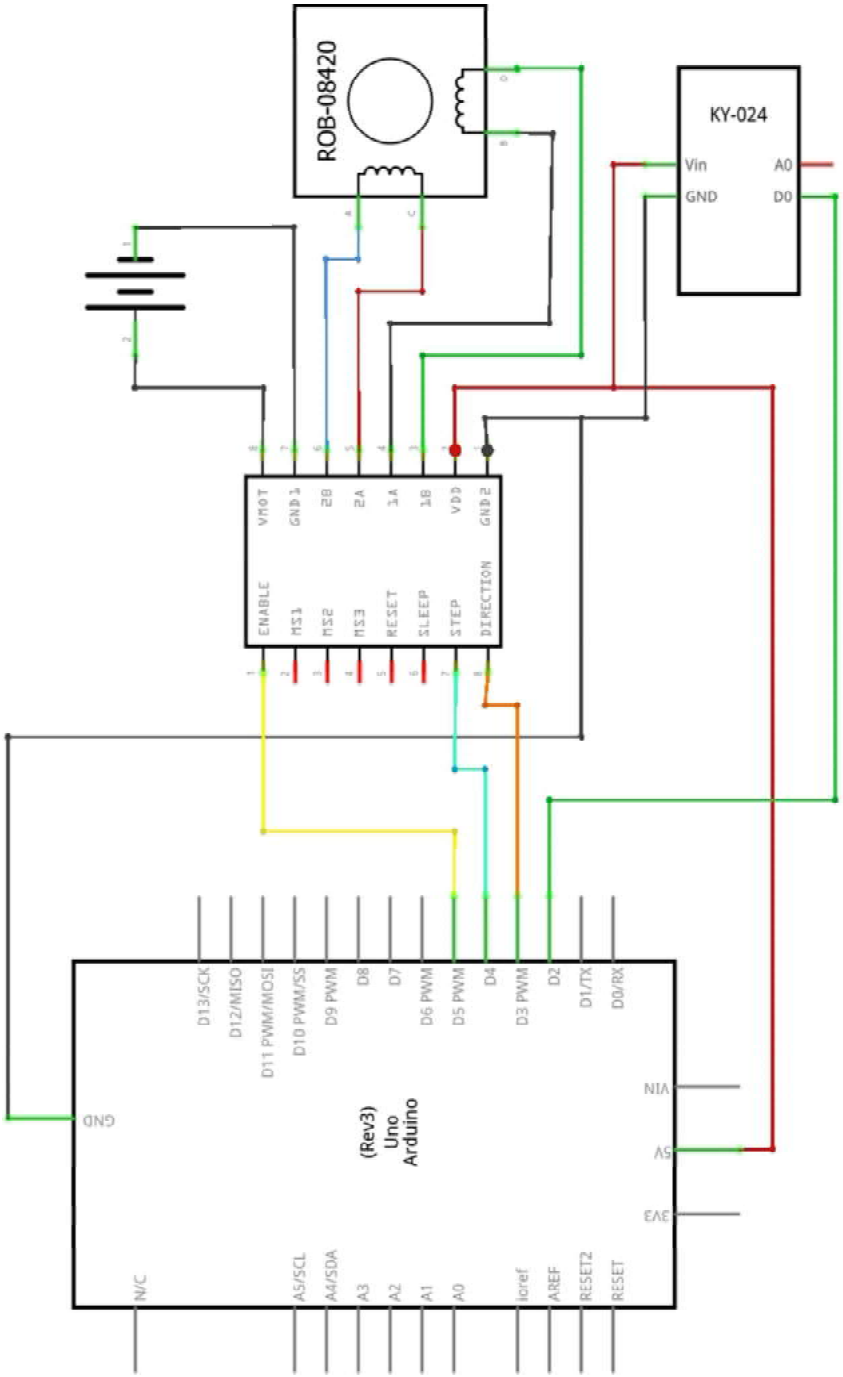


Abbildung 69: Schrittmotor-Aufbau – Schaltplan

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Verbindungsschema

Auch in diesem Fall ist eine 9 Volt Batterie anstelle des 12 V Transformators dargestellt (siehe Abbildung 70).

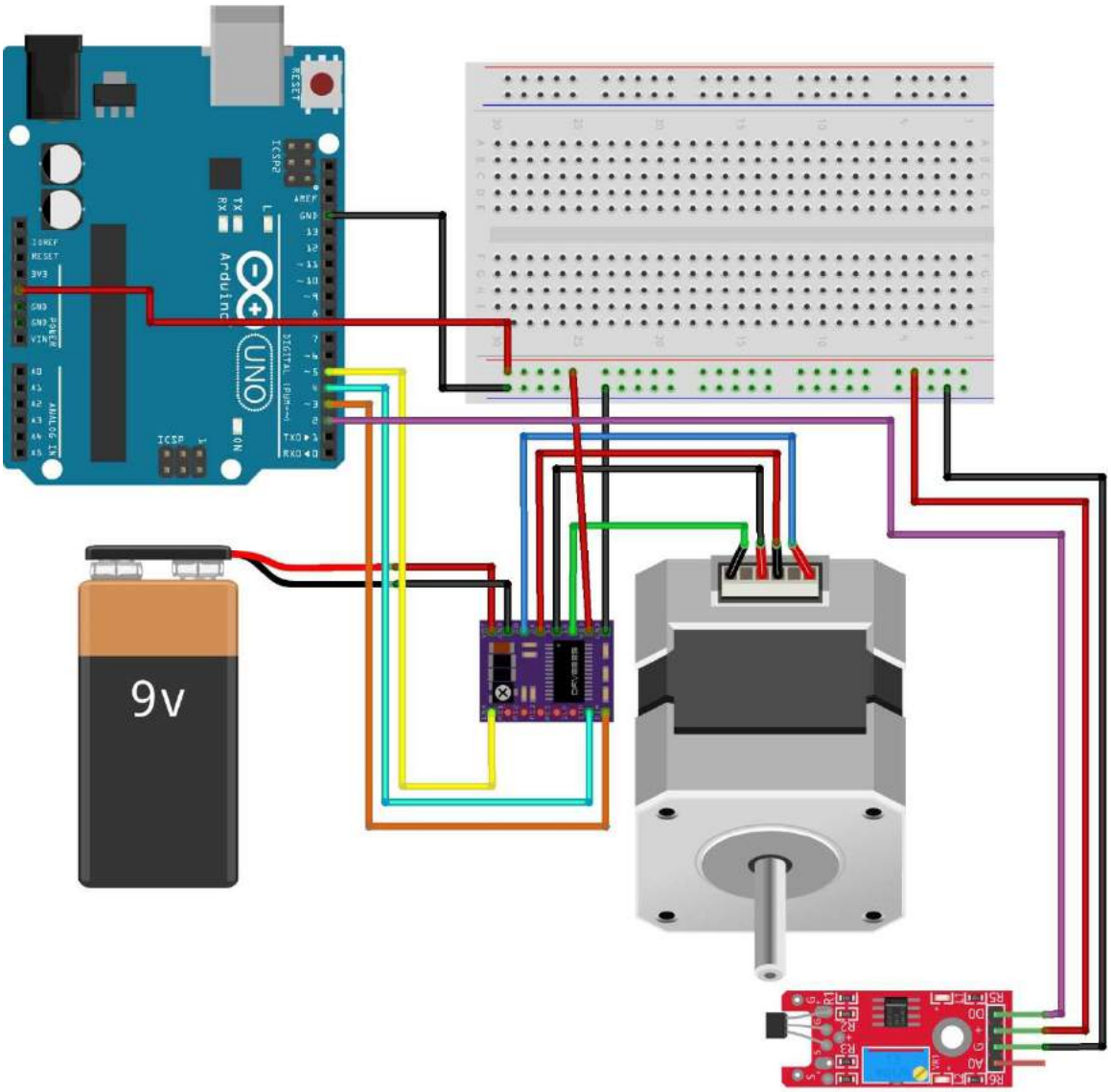


Abbildung 70: Schrittmotor-Aufbau – Verbindungsschema

Leider konnte der verwendete Treiber (TB6600) für den Schrittmotor nicht graphisch in dem Verbindungsschema dargestellt werden, daher ist im Verbindungsschema ein ähnlicher Treiber abgebildet.

Der verwendete Treiber TB6600 ist in Abbildung 71 dargestellt. Dabei handelt es sich um einen Microstep Driver. Demnach kann der Schrittmotor im Mikro-Schritt-Betrieb betrieben werden.

Bei diesem Versuchsaufbau wurde eine Auflösung der Schritte von 1/32 gewählt, um die Vibrationen des Motors zu minimieren.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

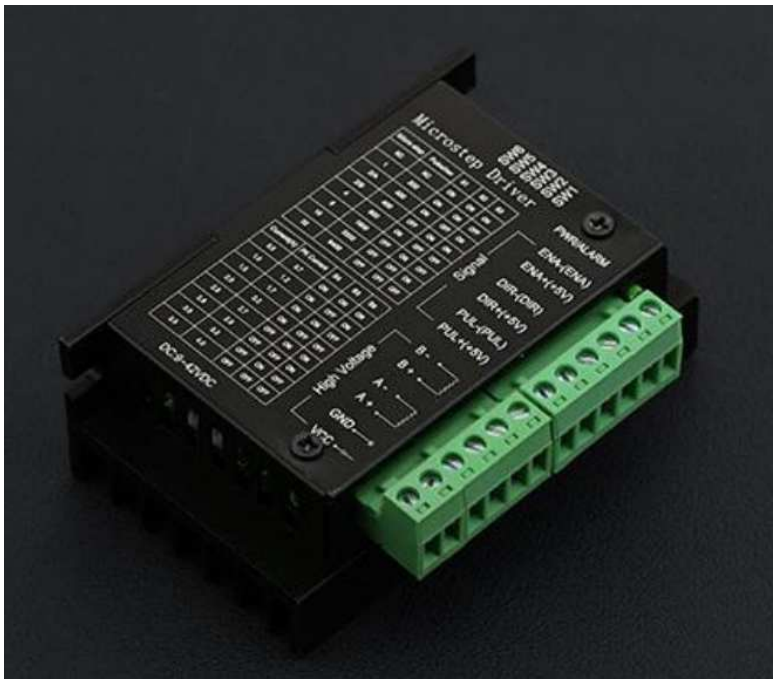


Abbildung 71: Microstep Driver für den Schrittmotor

Für den Versuchsaufbau wurde ein Schrittmotor Nema 23 ausgewählt. Bei diesem Motor handelt es sich um einen bipolaren Schrittmotor mit vier Anschlüssen. (siehe Abbildung 72)

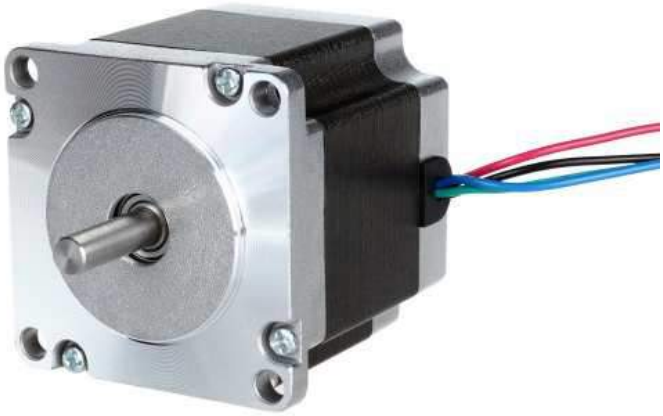


Abbildung 72: Bipolarer Schrittmotor Nema 23

Anhand der Tabelle 8 wurde der Schrittmotor am Treiber angeschlossen:

Cable type	Gauge	Coil	Function
Black	UL1007 AWG22	A	Motor coil A pin 1
Green	UL1007 AWG22	A-	Motor coil A pin 2
Red	UL1007 AWG22	B	Motor coil B pin 1
Blue	UL1007 AWG22	B-	Motor coil B pin 2

Tabelle 8: Verdrahtung des Schrittmotors

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Versuchsaufbau

Wie in Abbildung 73 dargestellt ist, wurde auf dem verlängerten Arm, welcher sich auf dem Schrittmotor befindet, ein Magnet angebracht. Dieser Magnet wird, wenn er sich dem Hall-Sernsor nähert, von diesem erkannt und der Motor dreht sich von nun an in die andere Richtung.

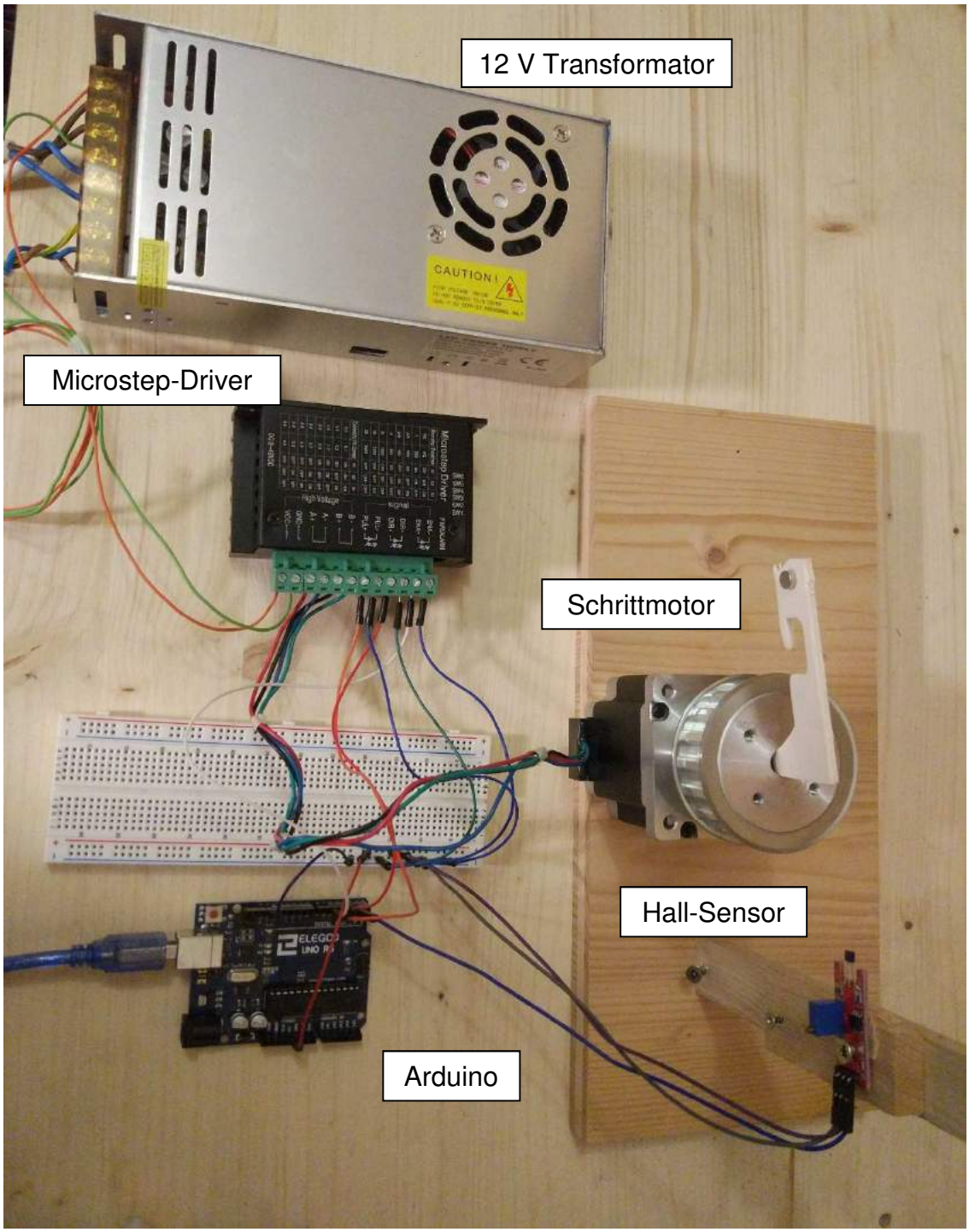


Abbildung 73: Schrittmotor-Aufbau – Versuchsaufbau

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Code

```
// Definieren der PINs
const int hallPin = 2; //PIN für den Hall-Sensor
const int dirPin  = 3; //PIN für die Richtung
const int stepPin = 4; //PIN für die Schritte
const int enPin   = 5; //PIN um Motor anzusprechen

int zufallswerte[] = {1500, 2500, 3200, 3700, 4200};

void setup() {
  // PINs auf Eingang bzw. Ausgang stellen
  pinMode(hallPin, INPUT);
  pinMode(stepPin,OUTPUT);
  pinMode(dirPin,OUTPUT);
  pinMode(enPin,OUTPUT);
  digitalWrite(enPin,LOW);

  randomSeed(analogRead(0)); //PIN für Zufallsgenerator
}

void loop() {
  int zuf_wert = random(0, 5); //Zufallswert festlegen

  while(digitalRead(hallPin) == 0) //bis der Hall-Sensor den Magneten
erkennt
  {
    digitalWrite(dirPin,HIGH); //Richtung einstellen
    digitalWrite(stepPin,HIGH); //einen Schritt bewegen
    delayMicroseconds(500); //Pause zwischen den Schritten
    digitalWrite(stepPin,LOW); //Schritt vollenden
    delayMicroseconds(500); //Pause zwischen den Schritten
  }
 delay(200); //0,2 Sekunden warten, bevor der Motor zurück fährt

  for(int x = 0; x < zufallswerte[zuf_wert]; x++)
  {
    digitalWrite(dirPin,LOW); //Richtung einstellen
    digitalWrite(stepPin,HIGH); //einen Schritt bewegen
    delayMicroseconds(500); //Pause zwischen den Schritten
    digitalWrite(stepPin,LOW); //Schritt vollenden
    delayMicroseconds(500); //Pause zwischen den Schritten
  }
 delay(5000); //5 Sekunden warten
}
```

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

5.2 Funktionsmuster

5.2.1 Ziel

In den Kapiteln 3,4 und 5.1 wurden einerseits die theoretischen Ansätze und Überlegungen zum Ballverhalten in der Luft wiedergegeben und anderseits auch schon erste Versuchsaufbauten erstellt. Diese enthalten die Elektronik-Komponenten, die sich auch in einer Ballmaschine wiederfinden.

Um die theoretischen Ansätze praktisch überprüfen zu können, muss ein Funktionsmuster entworfen und gebaut werden. Dieses Funktionsmuster wird schon als eine vollfunktionsfähige Ballmaschine ausgeführt sein, mit der das Flugverhalten der Maschine untersucht und die Antriebe sowie die Elektronik-Komponenten auf deren Eignung getestet werden können.

Dieses Funktionsmuster ist nicht auf Faktoren wie Größe oder Gewicht ausgelegt, sondern verfolgt das Ziel die vorhin genannten Punkte ausreichend überprüfen zu können. Dieses Modell kann eventuell in einer späteren Phase durch weitere Funktionen bzw. Motoren erweitert werden, um schlussendlich die gewünschte Ballmaschine zu erhalten.

Damit mit dem Funktionsmuster möglichst viele Messungen und Versuche durchgeführt werden können, wurden viele Langlöcher verwendet. Durch diese können die optimalen Einstellungen gefunden und später in der Endlösung durch einfache Löcher ersetzt werden.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

5.2.2 Auslegung belastungsrelevanter Bauteile

Als kritische Bauteile wurden die Wellen, auf denen die Schwungräder befestigt sind, identifiziert. Daher soll im Folgenden eine überschlagsmäßige Dimensionierung durchgeführt werden.

Zur Berechnung müssen alle wirkenden Kräfte und Momente erfasst werden. Die Welle wird mittig durch eine Kraft F belastet, die aus dem Zusammenpressen des Balles zwischen den Schwungrädern resultiert. Weiters wird sie durch ein Moment T tordiert, welche aus dem Abbremsen der Welle beim Abschuss eines Balles folgt. Eine schematische Darstellung der wirkenden Größen ist in Abbildung 74 ersichtlich:

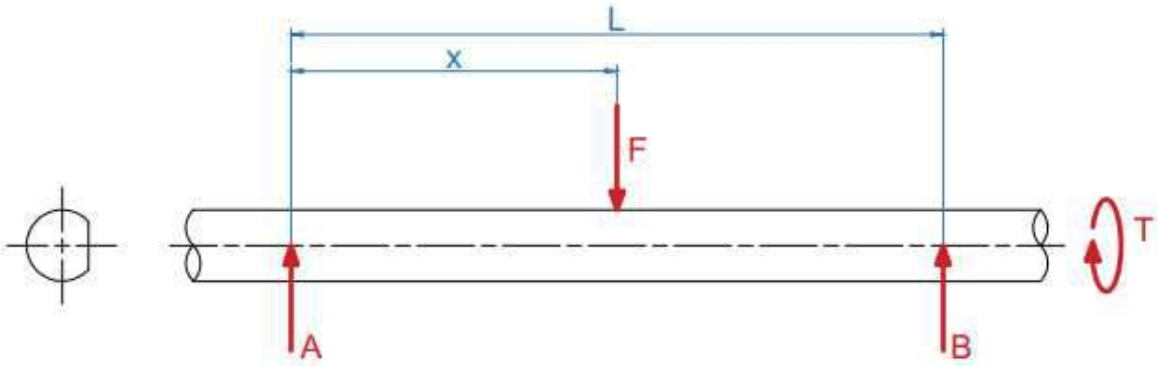


Abbildung 74: Schematische Darstellung der Schwungrad-Welle mit Belastungen

Die exakte Erfassung der Kräfte und Momente ist schwierig und soll daher im Folgenden durch überschlagsmäßige Überlegungen erfolgen. Durch einen ausreichend hoch gewählten Sicherheitsfaktor werden diese Unsicherheiten später berücksichtigt.

Berechnung der Kraft des Schwungrades auf die Welle

Um die Kraft, welche durch das Zusammenpressen des Balles entsteht, zu ermitteln, müssen zunächst mehrere Überlegungen angestellt werden.

Von Dignall (2005: 108-129) wurden Versuche durchgeführt, bei denen Tennisbälle mit verschiedenen Geschwindigkeiten auf eine Druckplatte geschleudert wurden. Daraus konnten in weiterer Folge die Kräfte bestimmt werden. Bei solchen Stößen kommt es ab einer gewissen Geschwindigkeit des Aufpralls zu einem sogenannten „Einknicken“ des Balls. Dieses Phänomen ist in Abbildung 75 rechts dargestellt. In solchen Fällen ist die Kraft höher als bei einem flachgedrückten Ball (links in Abbildung 75). Deshalb wird bei der weiteren Kraftermittlung der Fall (b) angenommen.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

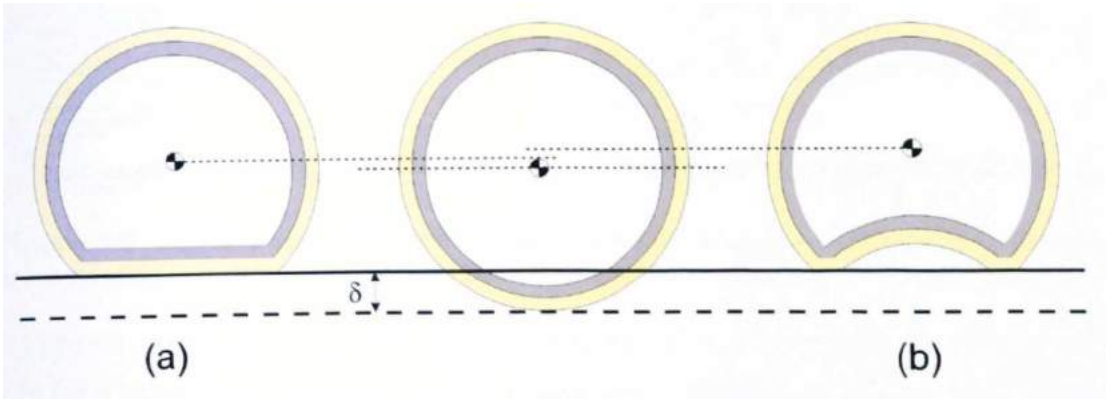


Abbildung 75: Abgeflachter Ball (a) und geknickter Ball (b) während eines Stoßes (Dignall, 2005: 124)

In weiterer Folge konnte ein Zusammenhang zwischen Verschiebung des Schwerpunktes und Deformation des Balles beschrieben werden (Dignall 2005: 126):

$$\delta = 74{,}830 \cdot x^2 + 0{,}841 \cdot x \tag{5.15}$$

Dabei entspricht δ der Deformation der Ball-Außenhülle und x der Verschiebung des Massenmittelpunkts. In Tabelle 9 sind die von Dignall (2005: 120) für Innendruckbälle berechneten Werte zusammen mit den daraus errechneten Deformationen dargestellt:

Aufprallgeschw. (m/s)	maximale gemessene Kraft (N)	Verschiebung des Massenmittelpunkts (mm)	Deformation (mm)
v	F_{max}	x	δ
2,90	100,00	5,00	6,08
5,80	225,00	8,50	12,55
13,50	590,00	15,00	29,45
16,50	725,00	16,75	35,08
20,00	900,00	18,75	42,08

Tabelle 9: Versuchsergebnisse für Innendruckbälle

Mithilfe dieser Daten werden die gesuchten Kräfte der Ballkompression für den maximal möglichen Balldurchmesser $d_{Ball} = 68{,}6\,mm$ für verschiedene Abstände Δh zwischen den Schwungrädern ermittelt.

Die Gesamt-Deformation δ errechnet sich durch

$$\delta = d_{Ball} - \Delta h \tag{5.16}$$

Da die Größe der Ballkompression auf beide Wellen gleichmäßig aufgeteilt wird, wird nun in weiterer Folge die halbierte Deformation verwendet. Durch Interpolation der Werte aus Tabelle 9 kann die zugehörige Kraft F gefunden werden. Die so errechneten Werte sind in Tabelle 10 aufgelistet:

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Schwungrad-Abstand (mm)	Gesamt-Deformation (mm)	Deformation einseitig (mm)	Kraft pro Welle [N]
Δh	δ	$\delta/2$	F
50	18,6	9,3	155
45	23,6	11,8	209
40	28,6	14,3	263
35	33,6	16,8	317
30	38,6	19,3	371

Tabelle 10: Kompressionskräfte bei unterschiedlichen Schwungrad-Abständen

Als maximale Kraft wird für die weitere Rechnung daher $F_{max} = 371\,N$ als ungünstigster Fall angenommen.

Berechnung des Bremsmoments

Das auf das System wirkende Bremsmoment kann berechnet werden durch:

$$M_t = I \cdot \alpha \tag{5.17}$$

mit dem Massenträgheitsmoment I des Systems sowie der Winkelbeschleunigung α, die es erfährt. Dabei sollen die Trägheitsmomente der Kupplungen und der Welle selbst vernachlässigt werden. Für ein ausreichend großes Schwungrad wird mit Inventor der Wert $I \approx 4 \cdot 10^{-3}\ kgm^2$ errechnet. Die Winkelbeschleunigung α muss der Einfachheit halber gemittelt werden und errechnet sich dann durch:

$$\alpha = \frac{\Delta\omega}{\Delta t} \tag{5.18}$$

Der Verlust an Winkelgeschwindigkeit $\Delta\omega$ der Schwungräder beim Abschussvorgang und die Kontaktzeit Δt zwischen Ball und Schwungrädern können nur grob berechnet oder gemessen werden. Basierend auf Zeitlupenaufnahmen bei anderen Modellen von Ballwurfmaschinen lässt sich $\Delta t \approx 0{,}005 s$ annehmen. Für den Verlust der Winkelgeschwindigkeit kann mithilfe einer händischen Lasermessung ein Drehzahlverlust $\Delta\omega \approx -100\frac{U}{min} = -10{,}5\frac{rad}{s}$ bei demselben Abschussvorgang angegeben werden.

Daraus folgt für die Winkelbeschleunigung:

$$\alpha = -2100\ rad/s^2 \tag{5.19}$$

und für das Bremsmoment:

$$M_t = I \cdot \alpha = 4 \cdot 10^{-3}\ kgm^2 \cdot -2100\frac{rad}{s^2} = -8{,}4\ Nm \tag{5.20}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Berechnung des maximalen Biegemoments

Nun soll das maximale Biegemoment der Welle berechnet werden. Dazu werden zunächst die Auflagerkräfte bestimmt. Eine schematische Darstellung der Welle mit den Kräften ist in Abbildung 74 ersichtlich. Für die Auflagerkräfte ergibt sich aus den Gleichgewichtsbedingungen durch die Symmetrie:

$$A = B = \frac{F}{2} = \frac{371\ N}{2} = 185{,}5\ N \tag{5.21}$$

Das maximale Biegemoment der Welle befindet sich in der Mitte der beiden Lager und beträgt bei einem angenommenen Lagerabstand von $L = 100mm$:

$$M_{b\_max} = A \cdot x = 185{,}5\ N \cdot 0{,}05\ m = 9{,}275\ Nm \tag{5.22}$$

Dabei entspricht x dem Abstand der Auflagerkräfte zur Wellenmitte.

Dimensionierung der Schwungrad-Welle

Aufgrund der gleichzeitigen Biege- und Torsionsbelastung muss im nächsten Schritt ein Vergleichsmoment gefunden werden. Dieses errechnet sich durch (vgl. Wittel et al. 2015: 373-374):

$$M_v = \sqrt{M_{b\_max}^2 + 0{,}75 \cdot \left(\frac{\sigma_{b,zul}}{\phi * \tau_{t,zul}} \cdot M_t\right)^2} = 11{,}75\ Nm \tag{5.23}$$

mit

$\phi = 1.73$ … Faktor zur Berechnung des Anstrengungsverhältnisses

$\sigma_{b,zul} = 180\,\frac{N}{mm^2}$ … zulässige Biegespannung für Werkstoff S235JR

$\tau_{t,zul} = 105\,\frac{N}{mm^2}$ … zulässige Torsionsspannung für Werkstoff S235JR

Für Vollwellen errechnet sich damit der Mindestdurchmesser durch (vgl. Wittel et al. 2015: 374):

$$d = \sqrt[3]{\frac{32 \cdot M_v}{\pi \cdot \sigma_{b,zul}}} = 8{,}73\ mm \tag{5.24}$$

Da das durch die Abflachung der Welle reduzierte Flächenträgheitsmoment der Welle noch nicht berücksichtigt wurde und um eine hohe Sicherheit aufgrund der Ungenauigkeiten in der Berechnung der auftretenden Belastungen zu gewährleisten, wird für das Funktionsmuster der Durchmesser $d = 15\ mm$ gewählt.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

5.2.3 Konstruktion

Abbildung 76 zeigt die gesamte Stückliste des Funktionsmusters ohne Schrauben und Normteile.

Stück	Bezeichnung	Teil	Werkstoff	Norm	Bemerkung
1	Zahnriemen	T46	Polyurethan		
1	Bolzen	T45	1.4301		
1	Holzplatte	T44	Birkenholz		
6	Distanzhülse 4	T43	1.4301		
1	Spannteil	T42	1.4301		
1	Halteplatte	T41	EN AW-5754 (AlMg3)		
1	Zahnriemenrad 1	T40	AlCuMg1		
1	Schrittmotor	T39			
2	Halterung Schrittmotor	T38	1.4301		
1	Zwischenscheibe 1	T37	EN AW-5754 (AlMg3)		
1	Zahnriemenrad 2	T36	AlCuMg1		
1	Gleitscheibe	T35	Bronze		
1	Gleitlager-Scheibe	T34	EN AW-5754 (AlMg3)		
1	Einschubblech	T33	1.4301		
1	Rohr 2	T32	Polypropylen		
1	Rohr 1	T31	Polypropylen		
2	Halterung Rohr 2	T30	EN AW-5754 (AlMg3)		
1	Kupplung 2	T29			
1	Ballrad-Welle	T28	S235JR		
1	Ballrad	T27	EN AW-5754 (AlMg3)		
2	Kupplung 1	T26			
2	Schwungrad-Welle	T25	S235JR		
2	Schwungrad	T24	EN AW-5754 (AlMg3)		
3	Fotodiode	T23			
6	Fotodioden-Aufnahme	T22	1.4301		
1	Diodenhalterung	T21	EN AW-5754 (AlMg3)		
3	Laser	T20			
1	Laserhalterung	T19	EN AW-5754 (AlMg3)		
1	Getriebemotor	T18			
2	Halterung Getriebemotor Teil 2	T17	1.4301		
1	Halterung Getriebemotor Teil 1	T16	1.4301		
2	Motor-Halterung	T15	EN AW-5754 (AlMg3)		
2	Gleichstrommotor	T14			
1	U-Profil	T13	Aluminium		
2	Stange	T12	1.4301		
1	Bodenplatte	T11	1.4301		
4	Flanschlager KFL002	T10	Zamak		
4	Lager-Halterung	T09	EN AW-5754 (AlMg3)		
4	Distanzhülse 3	T08	1.4301		
2	Distanzhülse 2	T07	1.4301		
7	Distanzhülse 1	T06	1.4301		
4	Gewindestange 3	T05	Stahl 4.6, verzinkt		
2	Gewindestange 2	T04	Stahl 4.6, verzinkt		
4	Gewindestange 1	T03	Stahl 4.6, verzinkt		
1	Seitenwand 2	T02	EN AW-5754 (AlMg3)		
1	Seitenwand 1	T01	EN AW-5754 (AlMg3)		

Abbildung 76: Stückliste des Funktionsmusters

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Abbildung 77 zeigt die Erzeugnisgliederung, aufgeteilt in Baugruppen (B) und Einzelteile (T).

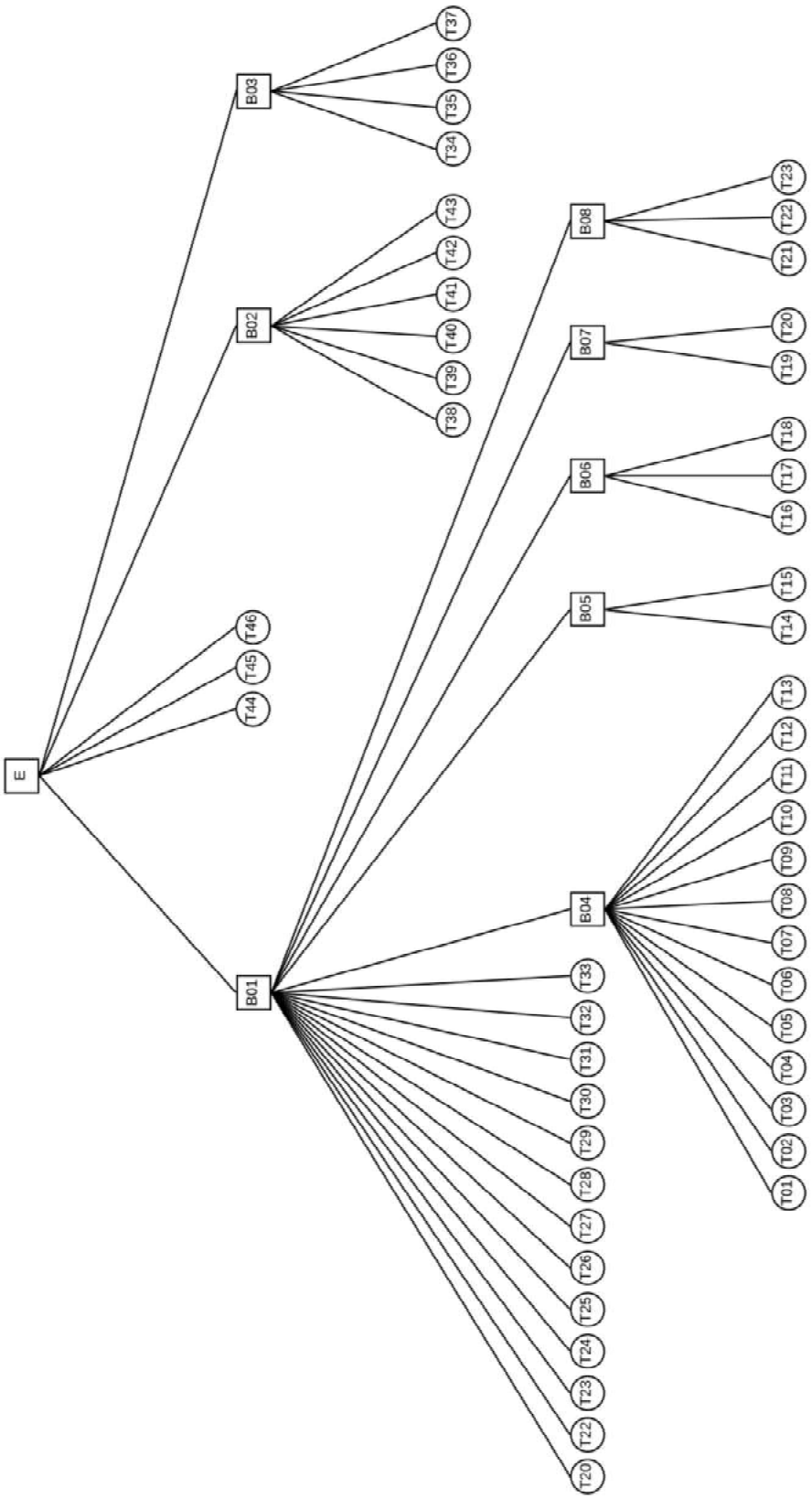


Abbildung 77: Erzeugnisgliederung des Funktionsmusters

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Die CAD-Konstruktion wurde mit Autodesk Inventor Professional 2019 durchgeführt. Abbildung 78 zeigt das fertige CAD-Modell des Funktionsmusters, Abbildung 79 die Außenmaße. Die vollständigen Zusammenstellungs- und Einzelteilzeichnungen befinden sich im Anhang.

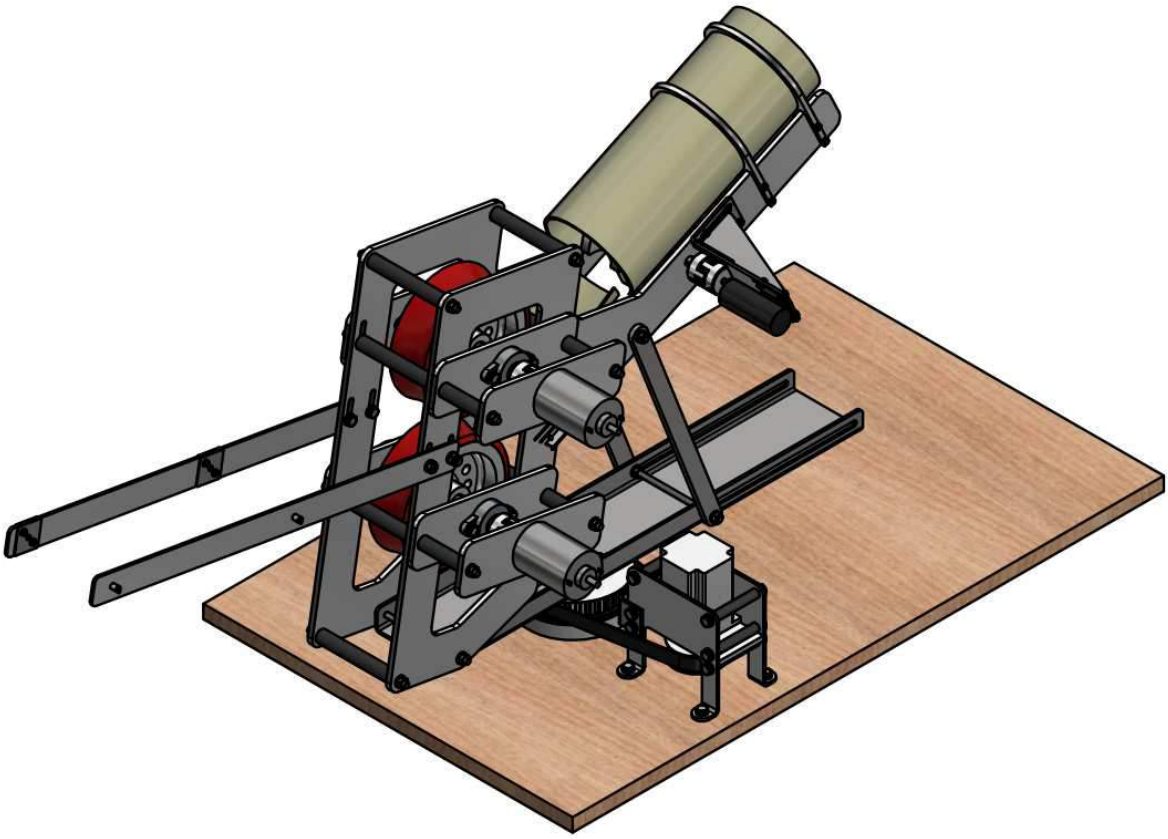


Abbildung 78: CAD-Modell des Funktionsmusters

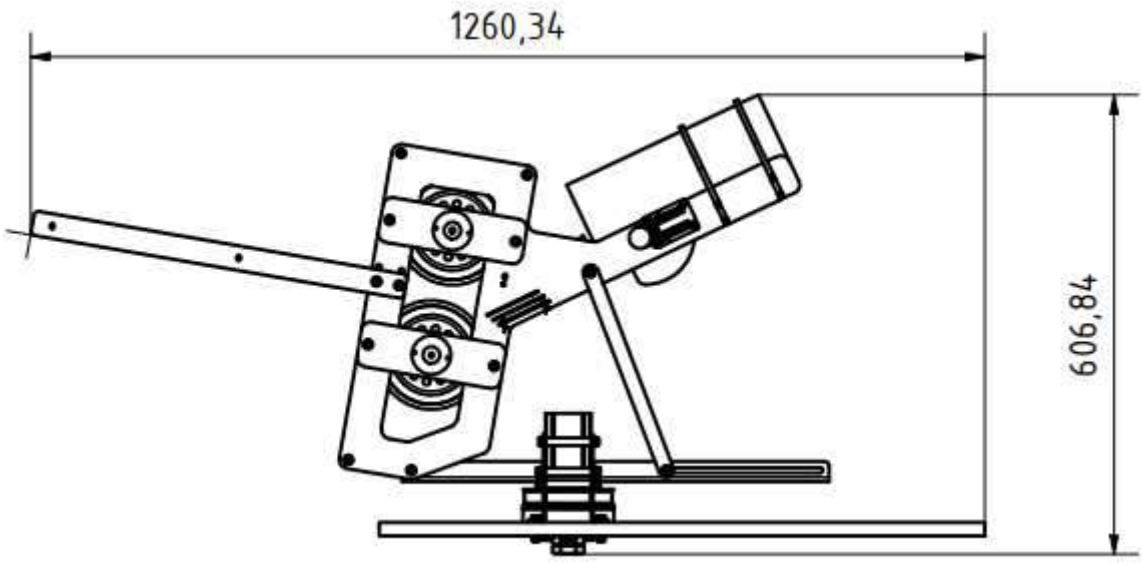
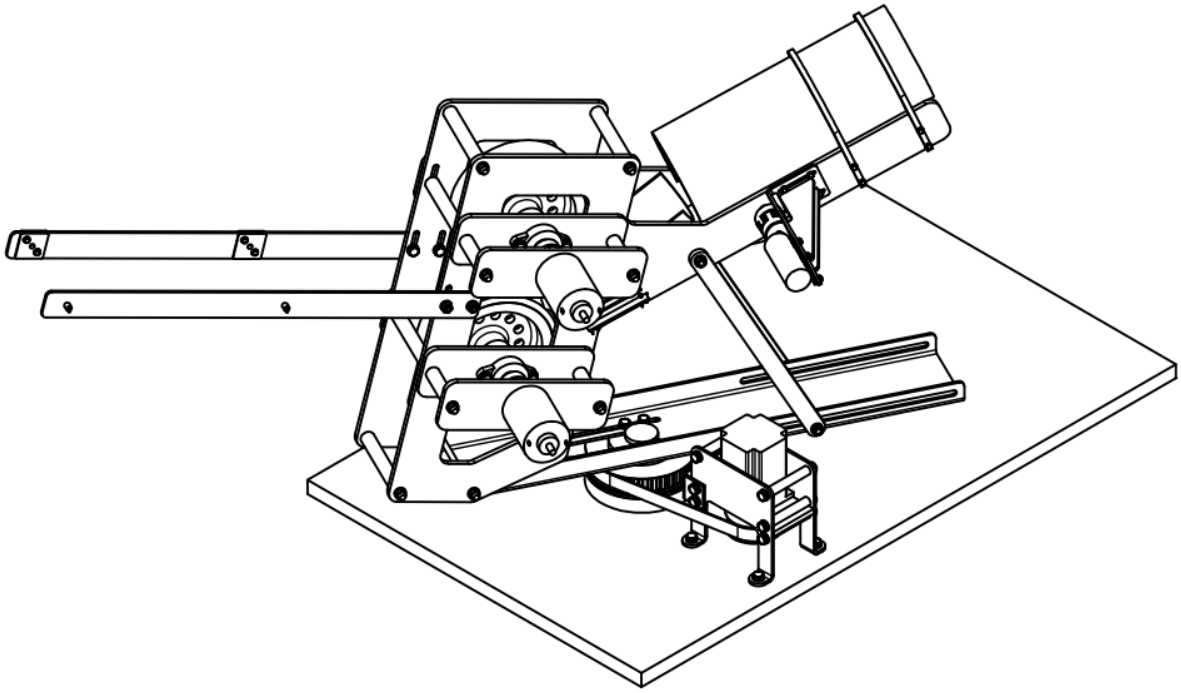


Abbildung 79: Außenmaße

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Im Folgenden sollen die Baugruppen sowie die zugehörigen Stücklisten überblicksmäßig dargestellt werden.

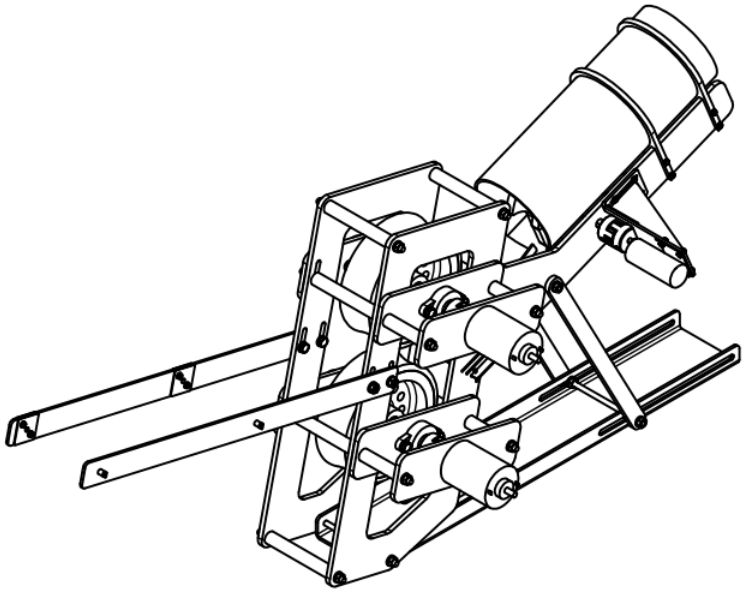
Funktionsmuster-Zusammenbau



Stück	Bezeichnung	Teil	Werkstoff	Norm	Bemerkung
2	Sechskantmutter M25	T62	Stahl	DIN 439	
4	Sechskantmutter M6	T61	Stahl	DIN 934	
4	Sechskantschraube M6x30	T57	Stahl	DIN 933	
4	Sechskantschraube M6x28	T56	Stahl	DIN 933	
4	Sechskantschraube M6x20	T54	Stahl	DIN 933	
16	Unterlegscheibe 6,4	T48	Stahl	DIN 125 A	
1	Zahnriemen	T46	Polyurethan		
1	Bolzen	T45	1.4301		
1	Holzplatte	T44	Birkenholz		
1	Gleitlager	B03			
1	Schrittmotor-Einheit	B02			
1	Abschuss-Einheit	B01			

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

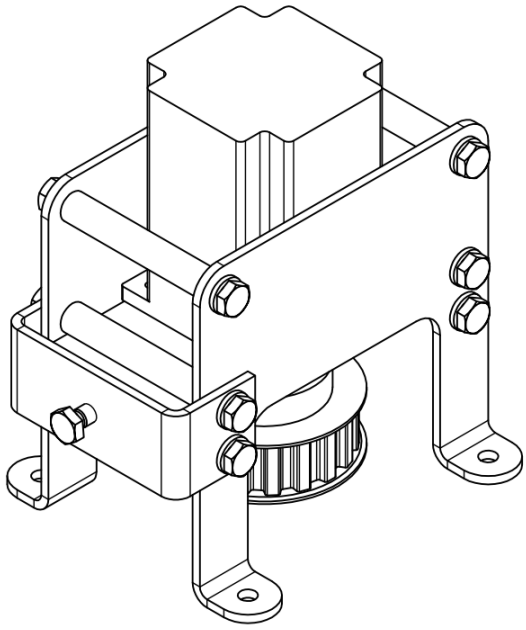
B01 Abschuss-Einheit



Stück	Bezeichnung	Teil	Werkstoff	Norm	Bemerkung
4	Splint 2x10	T68	Stahl	DIN 94	
8	Gewindestift M6x12	T66	Stahl	DIN 913	
2	Gewindestift M6x10	T65	Stahl	DIN 913	
2	Senkkopfschraube M3x8	T63	Stahl	DIN 7991	
12	Sechskantmutter M6	T61	Stahl	DIN 934	
16	Sechskantmutter M3	T60	Stahl	DIN 934	
4	Sechskantschraube M6x18	T53	Stahl	DIN 933	
8	Sechskantschraube M3x20	T52	Stahl	DIN 933	
6	Sechskantschraube M3x14	T51	Stahl	DIN 933	
16	Unterlegscheibe 6,4	T48	Stahl	DIN 125 A	
30	Unterlegscheibe 3,2	T47	Stahl	DIN 125 A	
1	Einschubblech	T33	1.4301		
1	Rohr 2	T32	Polypropylen		
1	Rohr 1	T31	Polypropylen		
2	Halterung Rohr 2	T30	EN AW-5754 (AlMg3)		
1	Kupplung 2	T29			
1	Ballrad-Welle	T28	S235JR		
1	Ballrad	T27	EN AW-5754 (AlMg3)		
2	Kupplung 1	T26			
2	Schwungrad-Welle	T25	S235JR		
2	Schwungrad	T24	EN AW-5754 (AlMg3)		
1	Fotodiode	T23			
2	Fotodioden-Aufnahme	T22	1.4301		
1	Laser	T20			
1	Diodenhalterungs-Einheit	B08			
1	Laserhalterungs-Einheit	B07			
1	Getriebemotor-Einheit	B06			
2	Gleichstrommotor-Einheit	B05			
1	Grundgerüst	B04			

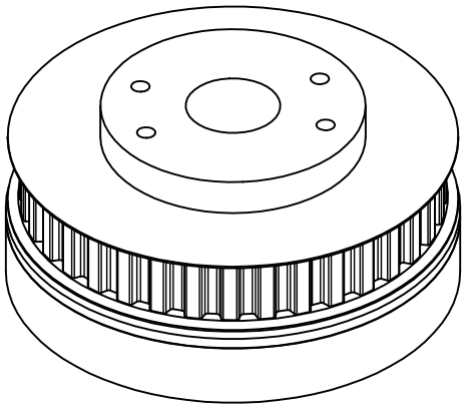
Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

B02 Schrittmotor-Einheit



Stück	Bezeichnung	Teil	Werkstoff	Norm	Bemerkung
1	Gewindestift M6x16	T67	Stahl	DIN 913	
6	Sechskantmutter M6	T61	Stahl	DIN 934	
2	Sechskantschraube M6x80	T59	Stahl	DIN 933	
4	Sechskantschraube M6x75	T58	Stahl	DIN 933	
1	Sechskantschraube M6x20	T54	Stahl	DIN 933	
12	Unterlegscheibe 6,4	T48	Stahl	DIN 125 A	
6	Distanzhülse 4	T43	1.4301		
1	Spannteil	T42	1.4301		
1	Halteplatte	T41	EN AW-5754 (AlMg3)		
1	Zahnriemenrad 1	T40	AlCuMg1		
1	Schrittmotor	T39	Aluminium		
2	Halterung Schrittmotor	T38	1.4301		

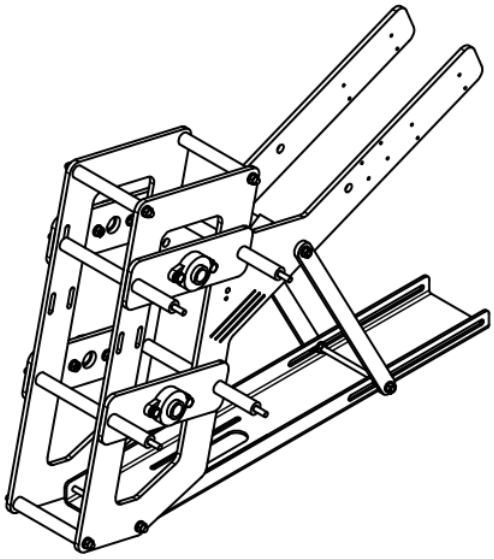
B03 Gleitlager



Stück	Bezeichnung	Teil	Werkstoff	Norm	Bemerkung
6	Senkkopfschraube M5x12	T64	Stahl	DIN 7991	
1	Zwischenscheibe 1	T37	EN AW-5754 (AlMg3)		
1	Zahnriemenrad 2	T36	AlCuMg1		
1	Gleitscheibe	T35	Bronze		
1	Gleitlager-Scheibe	T34	EN AW-5754 (AlMg3)		

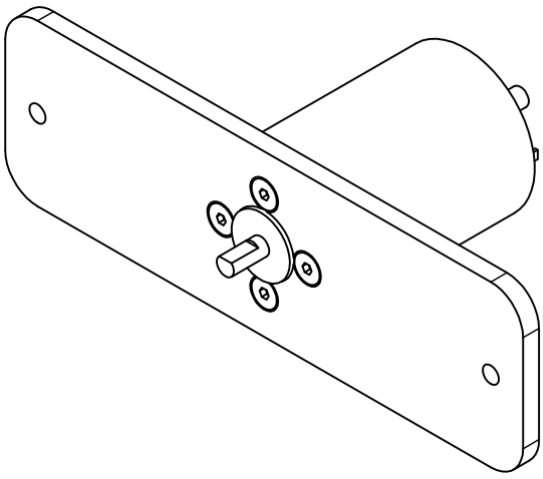
Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

B04 Grundgerüst



24	Sechskantmutter M6	T61	Stahl	DIN 934	
8	Sechskantschraube M6x22	T55	Stahl	DIN 933	
32	Unterlegscheibe 6,4	T48	Stahl	DIN 125 A	
1	U-Profil	T13	Aluminium		
2	Stange	T12	1.4301		
1	Bodenplatte	T11	1.4301		
4	Flanschlager KFL002	T10	Zamak		
4	Lager-Halterung	T09	EN AW-5754 (AlMg3)		
4	Distanzhülse 3	T08	1.4301		
2	Distanzhülse 2	T07	1.4301		
7	Distanzhülse 1	T06	1.4301		
4	Gewindestange 3	T05	Stahl 4.6, verzinkt		
2	Gewindestange 2	T04	Stahl 4.6, verzinkt		
4	Gewindestange 1	T03	Stahl 4.6, verzinkt		
1	Seitenwand 2	T02	EN AW-5754 (AlMg3)		
1	Seitenwand 1	T01	EN AW-5754 (AlMg3)		
Stück	Bezeichnung	Teil	Werkstoff	Norm	Bemerkung

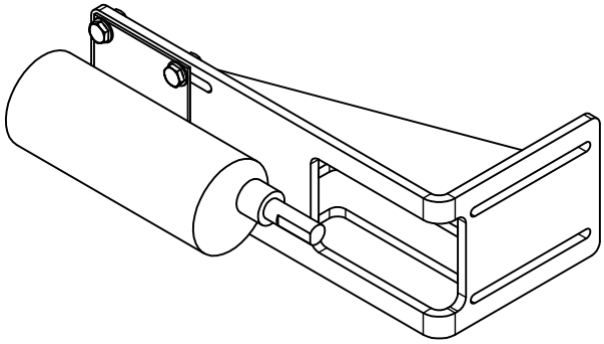
B05 Gleichstrommotor-Einheit



4	Senkkopfschraube M5x12	T64	Stahl	DIN 7991	
1	Motor-Halterung	T15	EN AW-5754 (AlMg3)		
1	Gleichstrommotor	T14			
Stück	Bezeichnung	Teil	Werkstoff	Norm	Bemerkung

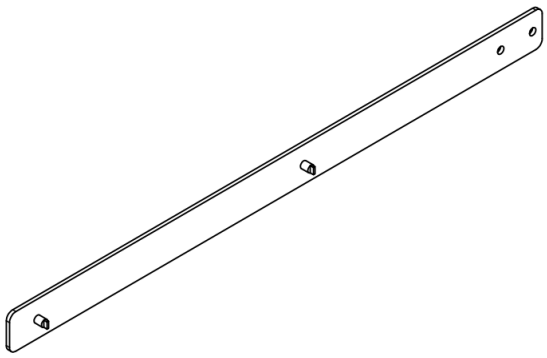
Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

B06 Getriebemotor-Einheit



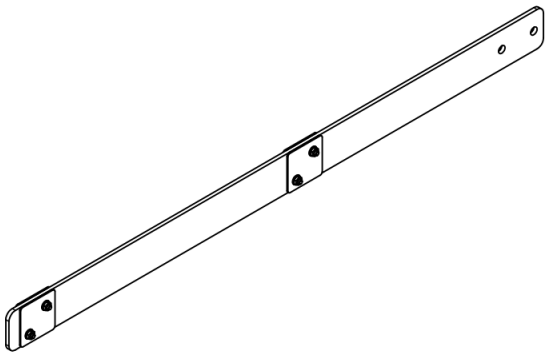
Stück	Bezeichnung	Teil	Werkstoff	Norm	Bemerkung
4	Sechskantmutter M3	T60	Stahl	DIN 934	
4	Sechskantschraube M3x8	T49	Stahl	DIN 933	
8	Unterlegscheibe 3,2	T47	Stahl	DIN 125 A	
1	Getriebemotor	T18			
2	Halterung Getriebemotor Teil 2	T17	1.4301		
1	Halterung Getriebemotor Teil 1	T16	1.4301		

B07 Laserhalterungs-Einheit



Stück	Bezeichnung	Teil	Werkstoff	Norm	Bemerkung
2	Laser	T20			
1	Laserhalterung	T19	EN AW-5754 (AlMg3)		

B08 Diodenhalterungs-Einheit

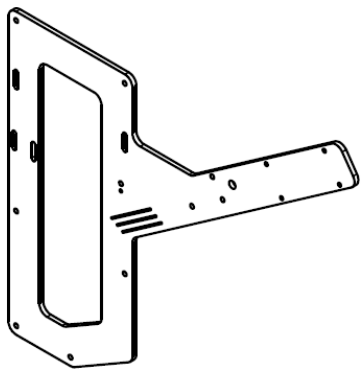
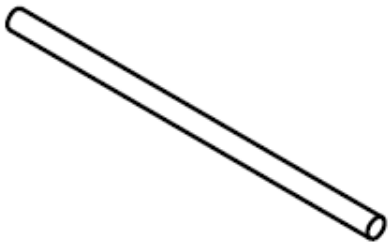
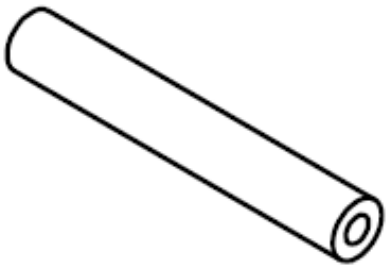


Stück	Bezeichnung	Teil	Werkstoff	Norm	Bemerkung
4	Sechskantmutter M3	T60	Stahl	DIN 934	
4	Sechskantschraube M3x10	T50	Stahl	DIN 933	
8	Unterlegscheibe 3,2	T47	Stahl	DIN 125 A	
2	Fotodiode	T23			
4	Fotodioden-Aufnahme	T22	1.4301		
1	Diodenhalterung	T21	EN AW-5754 (AlMg3)		

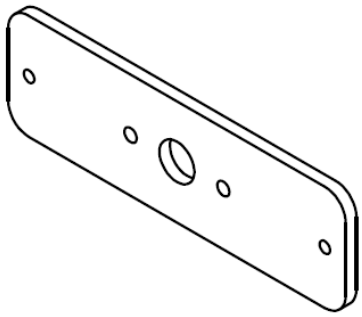
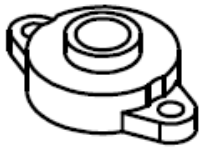
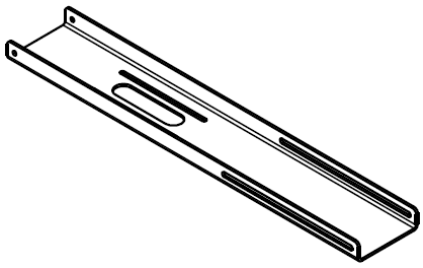
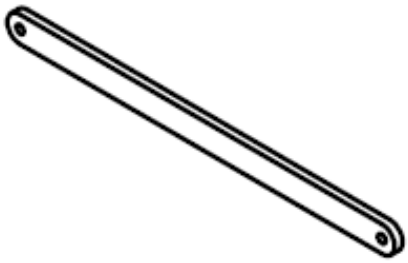
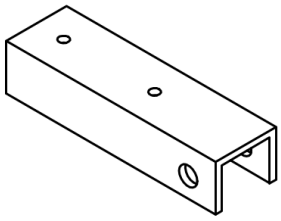
Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

5.2.4 Fertigung

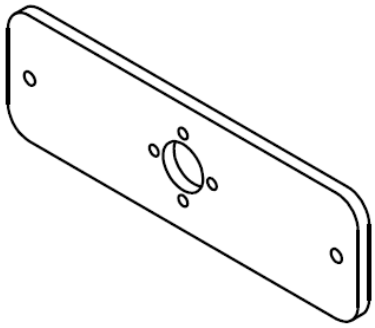
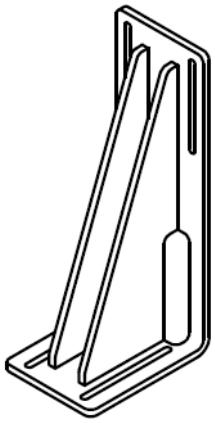
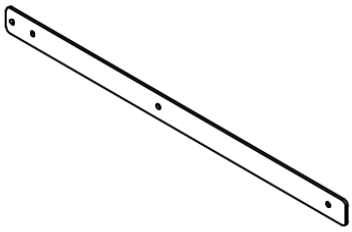
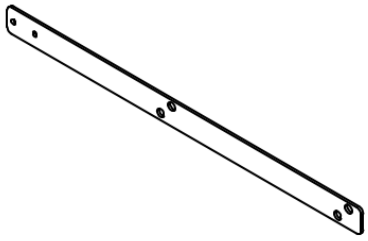
Im Folgenden soll kurz auf die Fertigung und Nachbearbeitung der Einzelteile sowie auf die Montage des Funktionsmusters eingegangen werden. Sie erfolgte großteils in der Werkstatt der Firma EUROPTEN unter der Aufsicht und Mitarbeit von Johann Erhardt. Einzelne Bauteile wurden durch die Firmen Dobetsberger und Bosche fremdgefertigt.

T01	**Seitenwand 1**	
T02	**Seitenwand 2**	
Material EN AW-5754 (AlMg3) Fertigung/Bezug laserschneiden (Fa. Dobetsberger) Nachbearbeitung nachbohren, entgraten, polieren		
T03	**Gewindestange 1**	
T04	**Gewindestange 2**	
T05	**Gewindestange 3**	
Material Stahl Fertigung/Bezug Zukaufteil Nachbearbeitung schneiden, entgraten		
T06	**Distanzhülse 1**	
T07	**Distanzhülse 2**	
T08	**Distanzhülse 3**	
Material EN AW-5754 (AlMg3) Fertigung/Bezug selbstständige Herstellung in der Werkstatt (Fa. EUROPTEN) Nachbearbeitung lackieren		

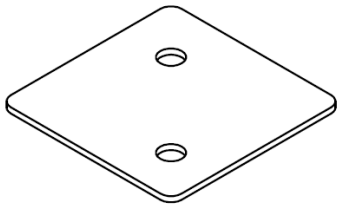
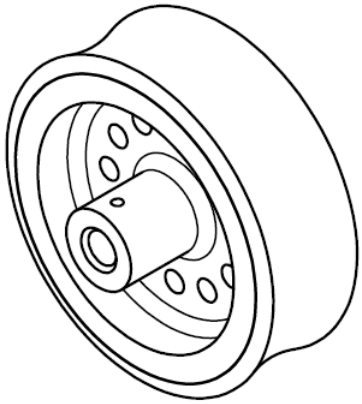
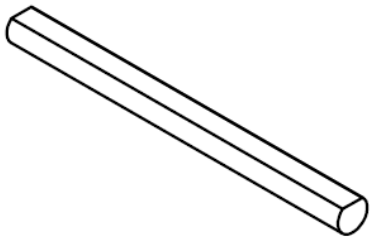
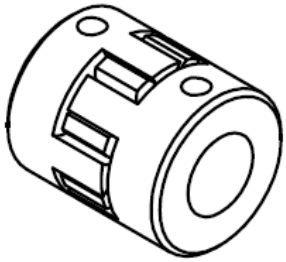
Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

T09	**Lager-Halterung**	
Material EN AW-5754 (AlMg3) Fertigung/Bezug laserschneiden (Fa. Dobetsberger) Nachbearbeitung nachbohren, entgraten, polieren		
T10	**Flanschlager KFL002**	
Material Zink-Druckguss (Zamak) Fertigung/Bezug Normteil Nachbearbeitung -		
T11	**Bodenplatte**	
Material 1.4301 (Austenitischer Chrom-Nickel-Stahl) Fertigung/Bezug laserschneiden, biegen (Fa. Dobetsberger) Nachbearbeitung nachbohren, entgraten, polieren		
T12	**Stange**	
Material 1.4301 (Austenitischer Chrom-Nickel-Stahl) Fertigung/Bezug laserschneiden (Fa. Dobetsberger) Nachbearbeitung nachbohren, entgraten, polieren		
T13	**U-Profil**	
Material Aluminium Fertigung/Bezug Zukaufteil Nachbearbeitung bohren, entgraten		

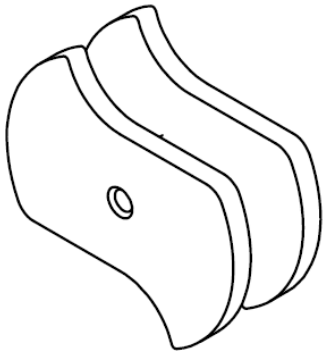
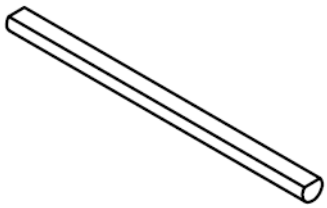
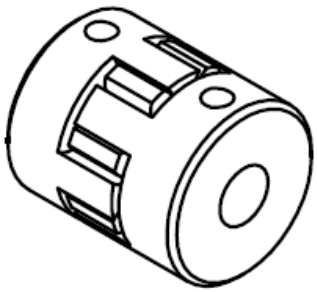
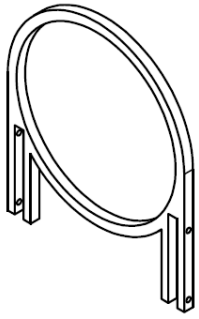
Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

T15	**Motor-Halterung**	
Material EN AW-5754 (AlMg3) Fertigung/Bezug laserschneiden (Fa. Dobetsberger) Nachbearbeitung nachbohren, entgraten, polieren		
T16	**Halterung Getriebemotor Teil 1**	
T17	**Halterung Getriebemotor Teil 2**	
Material 1.4301 (Austenitischer Chrom-Nickel-Stahl) Fertigung/Bezug laserschneiden, biegen (Fa. Dobetsberger) Nachbearbeitung entgraten, polieren, schweißen		
T19	**Laserhalterung**	
Material 1.4301 (Austenitischer Chrom-Nickel-Stahl) Fertigung/Bezug laserschneiden (Fa. Dobetsberger) Nachbearbeitung nachbohren, entgraten, polieren		
T21	**Diodenhalterung**	
Material 1.4301 (Austenitischer Chrom-Nickel-Stahl) Fertigung/Bezug laserschneiden (Fa. Dobetsberger) Nachbearbeitung nachbohren, entgraten, polieren		

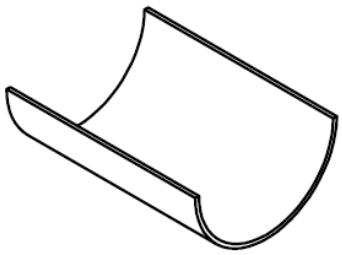
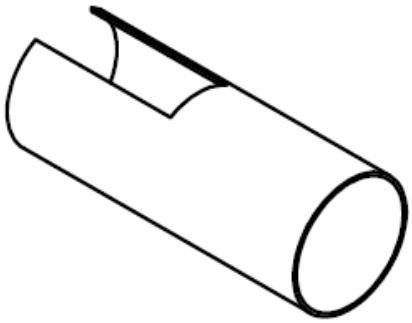
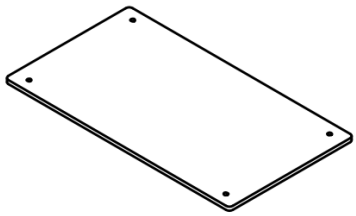
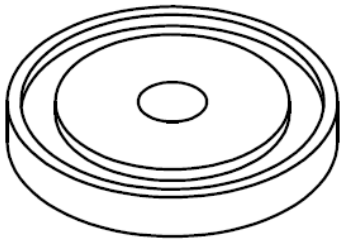
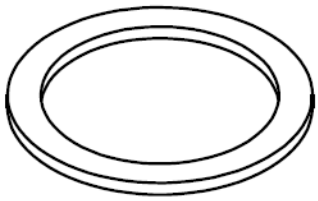
Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

T22	**Fotodioden-Aufnahme**	
Material 1.4301 (Austenitischer Chrom-Nickel-Stahl) Fertigung/Bezug laserschneiden (Fa. Dobetsberger) Nachbearbeitung nachbohren, entgraten, polieren		
T24	**Schwungrad**	
Material 1.4301 (Austenitischer Chrom-Nickel-Stahl) Fertigung/Bezug selbstständige Herstellung des Grundkörpers in der Werkstatt (Fa. EUROPTEN) Aufvulkanisierung der Außenhülle (ca. 70° Shore-Härte) Nachbearbeitung Stirnseiten drehen		
T25	**Schwungrad-Welle**	
Material S235JR Fertigung/Bezug selbstständige Herstellung in der Werkstatt (Fa. EUROPTEN) Nachbearbeitung verchromen (Fa. Bosche)		
T26	**Kupplung 1**	
Material Aluminium, Polyurethan (Shore-Härte 92°A) Fertigung/Bezug Zukaufteil Nachbearbeitung aufbohren, gewindeschneiden		

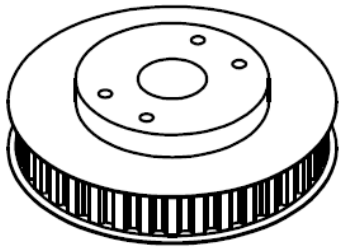
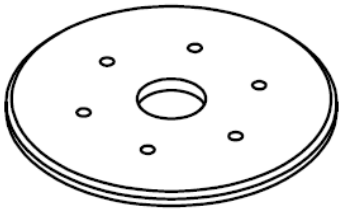
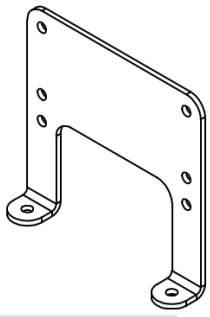
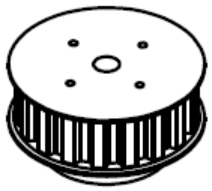
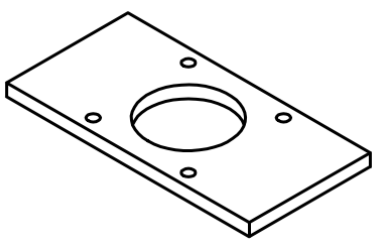
Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

T27	**Ballrad**	
Material EN AW-5754 (AlMg3) Fertigung/Bezug selbstständige Herstellung in der Werkstatt (Fa. EUROPTEN) Nachbearbeitung gewindeschneiden		
T28	**Ballrad-Welle**	
Material Stahl (S235JR) Fertigung/Bezug selbstständige Herstellung in der Werkstatt (Fa. EUROPTEN) Nachbearbeitung verchromen (Fa. Bosche)		
T29	**Kupplung 2**	
Material Aluminium, Polyurethan (Shore-Härte 92°A) Fertigung/Bezug Zukaufteil Nachbearbeitung aufbohren, gewindeschneiden		
T30	**Halterung Rohr 2**	
Material EN AW-5754 (AlMg3) Fertigung/Bezug laserschneiden (Fa. Dobetsberger) Nachbearbeitung nachbohren, entgraten, polieren		

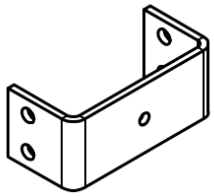
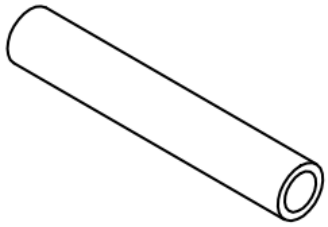
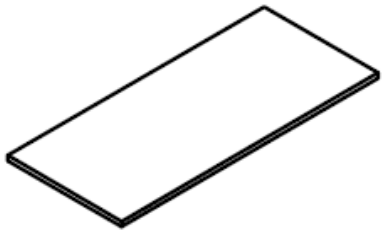
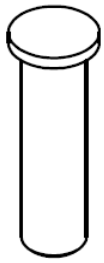
Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

T31	**Rohr 1**	
	Material Polypropylen (PP) Fertigung/Bezug Zukaufteil Nachbearbeitung zuschneiden, entgraten, schleifen	
T32	**Rohr 2**	
	Material Polypropylen (PP) Fertigung/Bezug Zukaufteil Nachbearbeitung zuschneiden, entgraten, schleifen	
T33	**Einschubblech**	
	Material 1.4301 (Austenitischer Chrom-Nickel-Stahl) Fertigung/Bezug laserschneiden (Fa. Dobetsberger) Nachbearbeitung entgraten, polieren	
T34	**Gleitlager-Scheibe**	
	Material EN AW-5754 (AlMg3) Fertigung/Bezug selbstständige Herstellung in der Werkstatt (Fa. EUROPTEN) Nachbearbeitung senken, entgraten, polieren	
T35	**Gleitscheibe**	
	Material Bronze Fertigung/Bezug selbstständige Herstellung in der Werkstatt (Fa. EUROPTEN) Nachbearbeitung -	

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

T36	**Zahnriemenrad 2**	
Material EN AW-2017A (AlCuMg1) Fertigung/Bezug Zukaufteil Nachbearbeitung bohren, entgraten, gewindeschneiden		
T37	**Zwischenscheibe**	
Material EN AW-5754 (AlMg3) Fertigung/Bezug selbstständige Herstellung in der Werkstatt (Fa. EUROPTEN) Nachbearbeitung -		
T38	**Halterung Schrittmotor**	
Material 1.4301 (Austenitischer Chrom-Nickel-Stahl) Fertigung/Bezug laserschneiden, biegen (Fa. Dobetsberger) Nachbearbeitung nachbohren, entgraten, polieren		
T40	**Zahnriemenrad 1**	
Material EN AW-2017A (AlCuMg1) Fertigung/Bezug Zukaufteil Nachbearbeitung bohren, entgraten, gewindeschneiden		
T41	**Halteplatte**	
Material EN AW-5754 (AlMg3) Fertigung/Bezug laserschneiden (Fa. Dobetsberger) Nachbearbeitung nachbohren, entgraten, polieren		

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

T42	**Spannteil**	
Material 1.4301 (Austenitischer Chrom-Nickel-Stahl) Fertigung/Bezug laserschneiden, biegen (Fa. Dobetsberger) Nachbearbeitung nachbohren, entgraten, polieren.		
T43	**Distanzhülse 4**	
Material EN AW-5754 (AlMg3) Fertigung/Bezug selbstständige Herstellung in der Werkstatt (Fa. EUROPTEN) Nachbearbeitung lackieren		
T44	**Holzplatte**	
Material Leimholz Fichte Fertigung/Bezug Zukaufteil Nachbearbeitung schleifen, versiegeln		
T45	**Bolzen**	
Material 1.4301 (Austenitischer Chrom-Nickel-Stahl) Fertigung/Bezug selbstständige Herstellung in der Werkstatt (Fa. EUROPTEN) Nachbearbeitung -		
T46	**Zahnriemen**	
Material Gießpolyurethan (PU) mit Stahlzugstrang Fertigung/Bezug Zukaufteil Nachbearbeitung -		

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Außerdem wurden folgende Schrauben und Normteile verwendet:

- Sechskantschrauben und Senkkopfschrauben M3, M5, M6
- Unterlegscheiben 3,2 und 6,4
- Sechskantmuttern M3, M6, M25
- Gewindestifte M6
- Splints 2x10

Nach der Fertigung aller Teile und Bezug der Zukaufteile wurden diese in der Werkstatt der Fa. EUROPTEN nachbearbeitet. Fotos dazu befinden sich im Anhang D.

Nach der Montage der Unterbaugruppen konnte die gesamte Abschuss-Einheit auf dem Gleitlager befestigt werden (siehe Abbildung 80) und alle weiteren Baugruppen eingebaut werden.

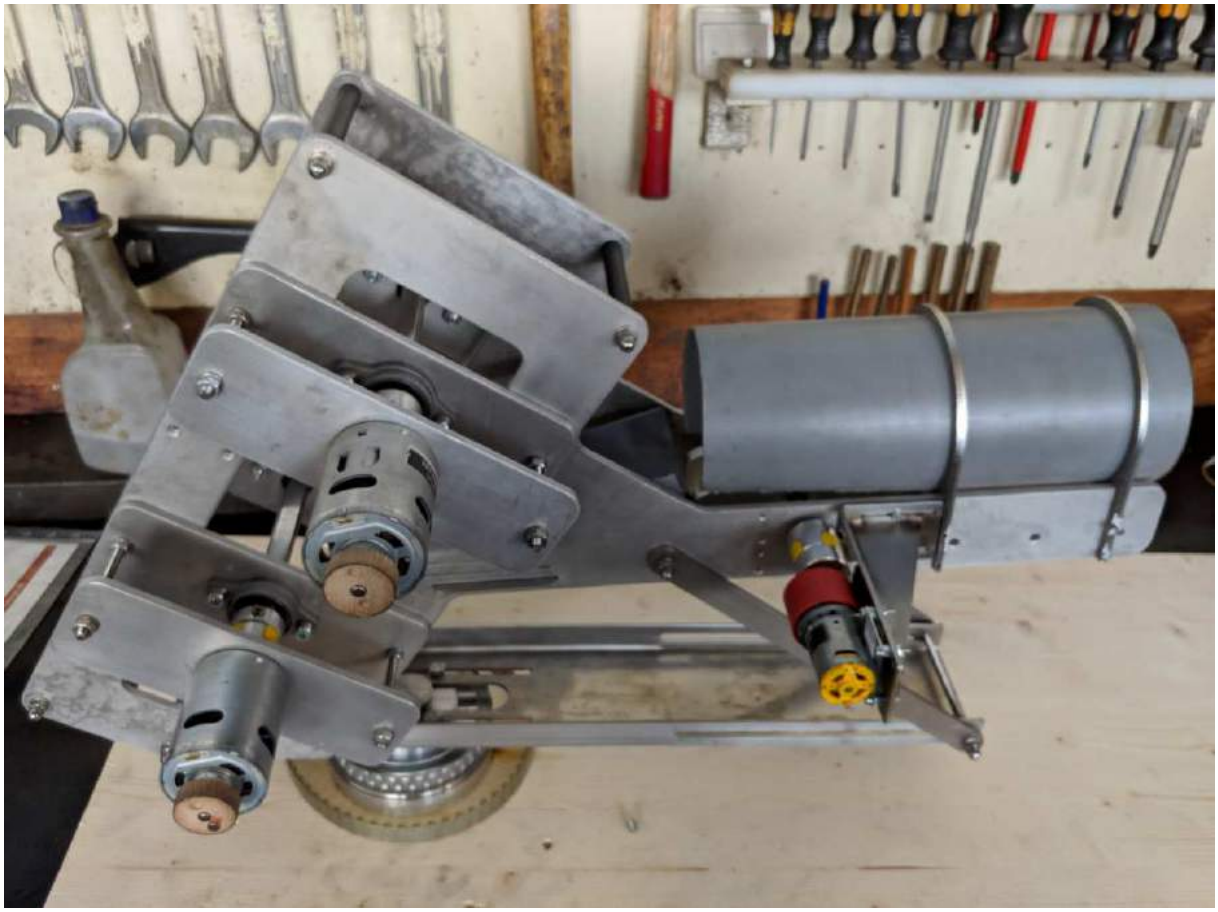


Abbildung 80: Abschuss-Einheit (ohne Schwungräder)

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

5.2.5 Messungen

Die Tennisballmaschine soll in ihrer fertigen Version als vollautomatische Maschine ausgeführt sein. Um dies gewährleisten zu können, müssen vorab einige Messungen durchgeführt werden. Einerseits sollen anhand der Messungen die theoretischen Ansätze der Berechnungen überprüft und andererseits die – für eine Vollautomatisierung noch fehlenden Variablen – ermittelt werden.

Dazu wurden die folgenden Messaufbauten realisiert.

Auswahl des optimalen Abstandes der Schwungräder

In der Abbildung 81 ist ersichtlich, dass für die Positionierung des oberen Schwungrades Langlöcher gewählt wurden. Der Abstand der Schwungräder kann zwischen 30 mm und 50 mm variieren.

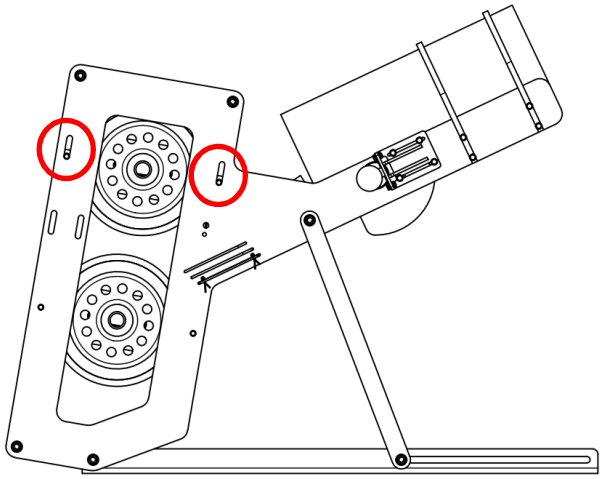


Abbildung 81: Positionierung der Schwungräder über die Langlöcher

Bei unterschiedlichen Abständen der Schwungräder wurden verschiedene Verhaltensweisen – wie Flugkurve, Flugverhalten etc. – der Tennisbälle beobachtet. Dabei erwies sich ein Abstand von 50 mm für die Maschine am geeignetsten, da der Ball in diesem Fall am besten steuerbar ist.

Messung der Drehzahl der Schwungräder

Bei dieser Messung wird die Drehzahl des oberen Schwungrades ermittelt.

Messaufbau

Die Messung der Drehzahl erfolgt mittels einem Hall-Sensor und einem Magneten, welcher auf dem oberen Schwungrad montiert ist. Die Funktionsweise des Hall-Sensors ist im Kapitel Drehzahlmessung-Aufbau beschrieben.

Die aktuelle Drehzahl wird auf einem LCD-Display angezeigt und zusätzlich alle 200 ms in einer TXT-Datei auf der SD gespeichert. Mithilfe dieser Datei können die gemessenen Daten ausgelesen und ausgewertet werden.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Der Messaufbau ist in Abbildung 82 schematisch dargestellt und beschrieben.

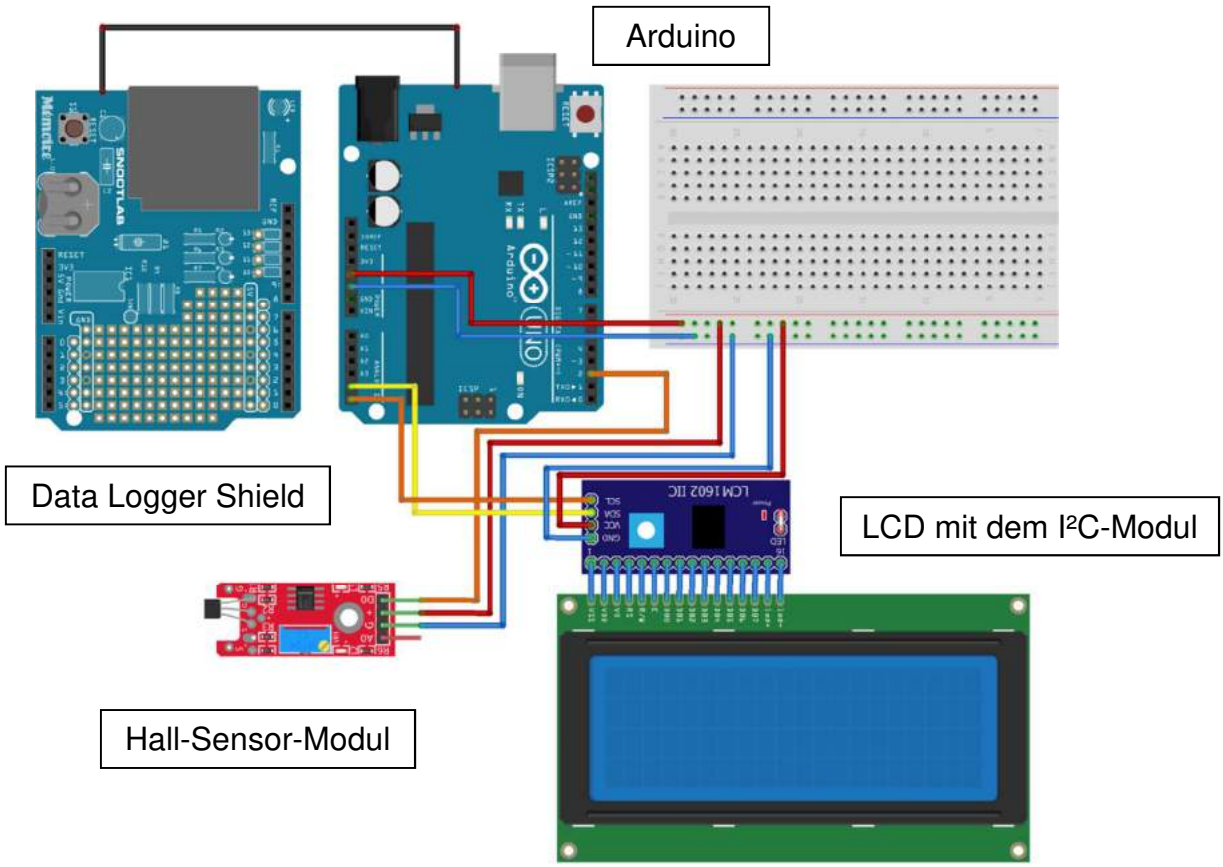


Abbildung 82: Messaufbau zur Messung der Drehzahl der Schwungräder

Messdurchführung

Zusätzlich zu diesem Messaufbau werden die beiden Schwungräder mithilfe eines Raspberry PIs und einem Arduino angesteuert. Über eine Eingabe-Konsole am Raspberry PI wird den beiden DC-Motoren, welche die Schwundräder antreiben, bestimmte PWM-Werte zugewiesen.

Die Messung beginnt, sobald die Schwungräder eine Drehzahl größer Null aufwiesen.

Bei den beiden DC-Motoren wird nun ein PWM-Wert von 240 eingestellt und die Motoren gestartet. Danach werden in bestimmten Abständen vier Tennisbälle abgeschossen.

Daten

Während dieser Messung werden die gemessenen Drehzahlen zeilenweise alle 200 ms in eine TXT-Datei geschrieben.

Auswertung

Die gemessenen Drehzahlen sind graphisch in Abbildung 83 dargestellt. Diese Darstellungsform ermöglicht folgende Schlussfolgerungen:

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Zunächst lässt sich daraus ablesen, dass der Motor ungefähr 55 Sekunden die Drehzahl erhöhte, bis der erste Schuss abgegeben wurde. Dieser Verlauf der Kurve bestätigt den in der Literatur gefundenen Verlauf, welcher in Abbildung 44 dargestellt ist.
Danach werden in Intervallen von je sieben Sekunden die weiteren Schüsse abgegeben.
Der DC-Motor hat nach ca. 12 Sekunden schon annähernd die eingestellte Drehzahl erreicht und ist von dort an nur noch marginal gestiegen. Demnach kann bei Betreiben der Maschine der erste Schuss nach ungefähr 12 Sekunden abgeschossen werden.

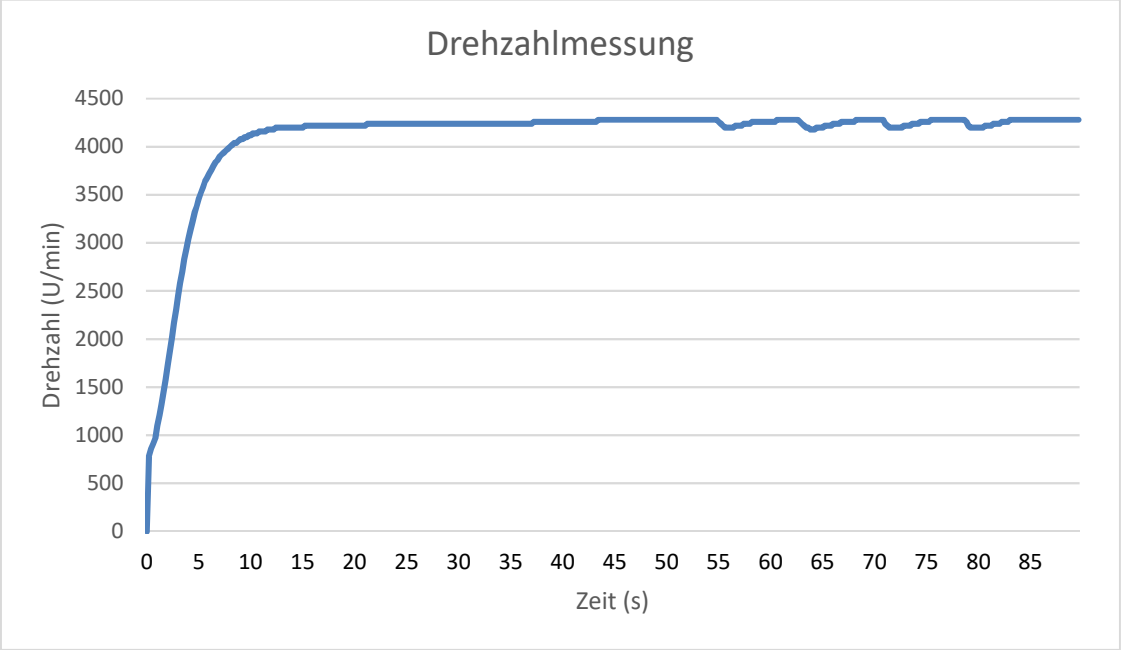


Abbildung 83: Drehzahlmessung mit vier Bällen

Dies kann mit der Formel 5.5, die wie folgt lautet,

$$\Delta t = -J \frac{\omega_{max}}{M_0} \ln(M_0 - \frac{M_0}{\omega_{max}} \omega - M_b) \Big|_{\omega_1}^{\omega_2}$$

berechnet werden.

Dazu werden die technischen Daten des ausgewählten Motors benötigt.

- Leerlaufdrehzahl $\omega_{max} = 5167 \frac{U}{min} = 541{,}087 \frac{1}{s}$
- Stillstandsmoment $M_0 = 0{,}483\ Nm$
- Start-Drehzahl $\omega_1 = 0 \frac{1}{s}$

Des Weiteren sind das Trägheitsmoment des Schwungrades $J = 3573{,}2 \cdot 10^{-6}\ kgm^2$ und das Belastungsmoment $M_b = 0{,}06\ Nm$ von Bedeutung.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Zuletzt muss noch die zu erreichende Drehzahl festgelegt werden. Bei einem PWM-Wert von 240 ergibt sich eine Drehzahl von $\omega_2 = 4280 \frac{U}{min} = 448{,}201 \frac{1}{s}$

Nach Einsetzen dieser Werte in die Formel 5.5 ergibt sich für die Beschleunigungszeit des Motors:

$$\boldsymbol{\Delta t = 11{,}758\ s}$$

Demnach bestätigen die Messungen und Beobachtungen der Beschleunigungszeit die theoretischen Überlegungen.

In Abbildung 84 ist eine Detailaufnahme der Drehzahlmessung von der 54. Sekunde bis zur 82. Sekunde dargestellt. Hierbei lässt sich der Drehzahlabfall bei einem Ballabschuss deutlicher erkennen. Des Weiteren ist ersichtlich, dass die DC-Motoren nach ungefähr fünf Sekunden wieder auf eingestellter Drehzahl sind.

Außerdem ist zu erkennen, dass sich bei jedem einzelnen Schuss die DC-Motoren fast identisch verhalten, was dem gewünschten Verhalten entspricht.

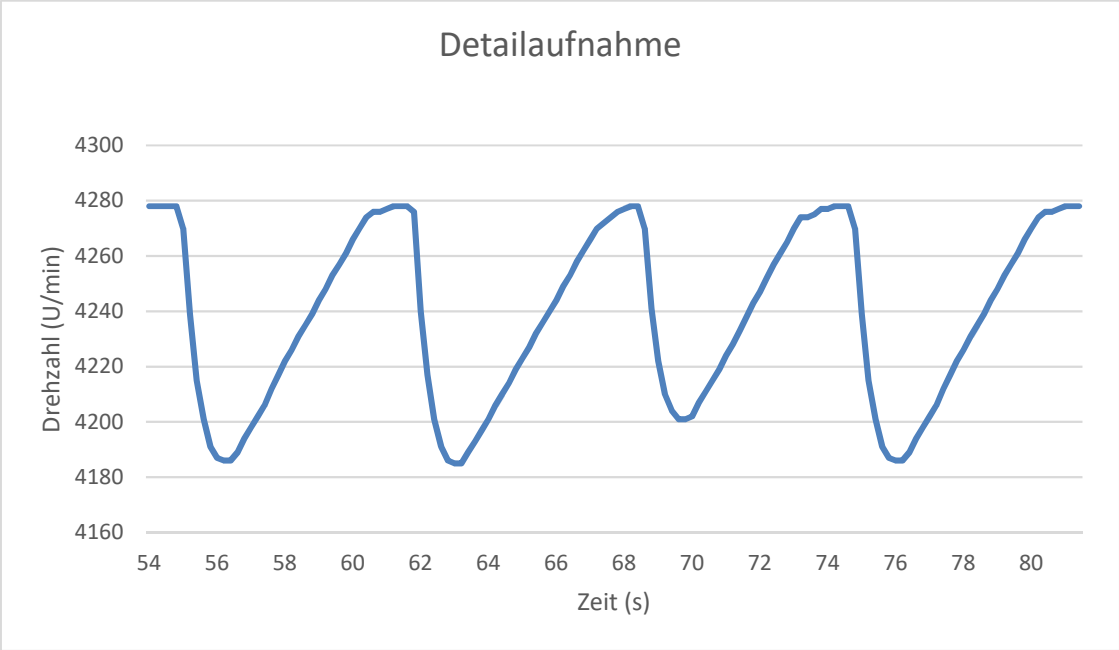


Abbildung 84: Detailaufnahme der Drehzahlmessung

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Zusammenhang zwischen Drehzahl eines Schwungrades und eingestellten PWM-Wert des DC-Motors

Ziel dieser Messung ist es, einen Zusammenhang in Form einer linearen Funktion zwischen der Drehzahl des oberen Schwungrades und dem eingestellten PWM-Wert des DC-Motors zu finden.

Messaufbau
Der Messaufbau ist identisch mit dem Messaufbau der vorherigen Messung.

Messdurchführung
Über eine Eingabe-Konsole am Raspberry PI wurden dem DC-Motor, welcher das obere Schwungrad antreibt, bestimmte PWM-Werte zugewiesen. Nach ungefähr 20 Sekunden wurde die dazugehörige Drehzahl vom LCD-Display abgelesen. Diese Messung wurde mit unterschiedlichen PWM-Werten wiederholt.

Daten
Die PWM-Werte mit den dazugehörigen gemessenen Drehzahlen sind der Tabelle 11 zu entnehmen.

PWM oben	100	105	125	150	170	175	180	185	190	198	205	225	240	255
Drehzahl (U/min)	1520	1620	2040	2530	2910	3000	3100	3200	3300	3420	3640	3980	4280	4580

Tabelle 11: PWM-Werte des oberen DC-Motors mit dazugehörigen Drehzahlen

Auswertung
Diese Messpunkte sind in Abbildung 85 graphisch dargestellt.

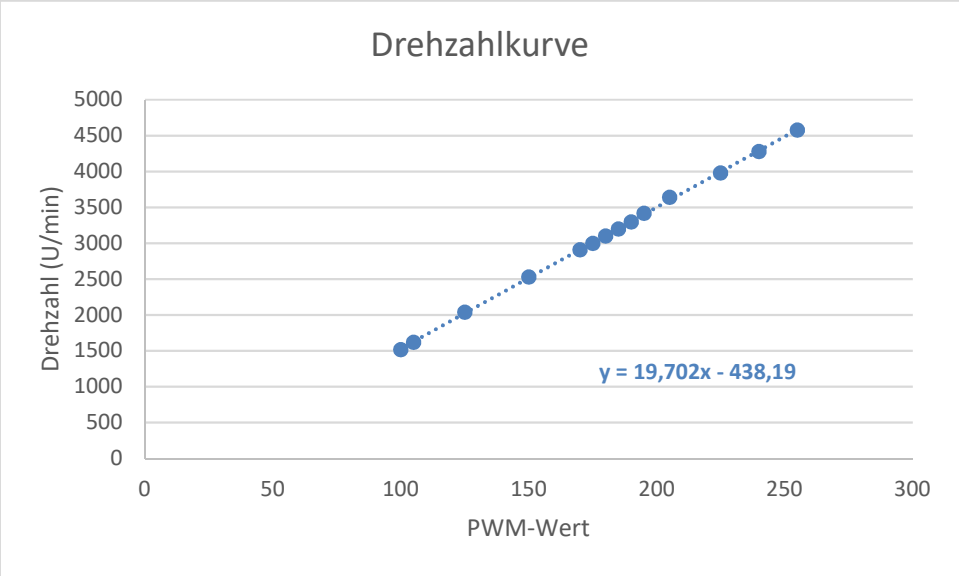


Abbildung 85: Drehzahlkurve

Mithilfe dieser Messpunkte konnte eine lineare Funktion (strichlierte Linie im Graphen) angenähert werden, welche den Zusammenhang zwischen der Drehzahl des oberen Schwungrades und dem eingestellten PWM-Wert des DC-Motors wiedergibt.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Messung der Anfangsgeschwindigkeit des Tennisballes

Bei dieser Messung wird die Anfangsgeschwindigkeit der Tennisbälle bei unterschiedlichen Drehzahlen der DC-Motoren und unterschiedlich abgenützten Tennisbällen ermittelt.

Messaufbau

Die Messung der Anfangsgeschwindigkeit erfolgt identisch wie in Kapitel 5.1.4 beschrieben.

Der schematische Messaufbau ist in Abbildung 86 abgebildet.

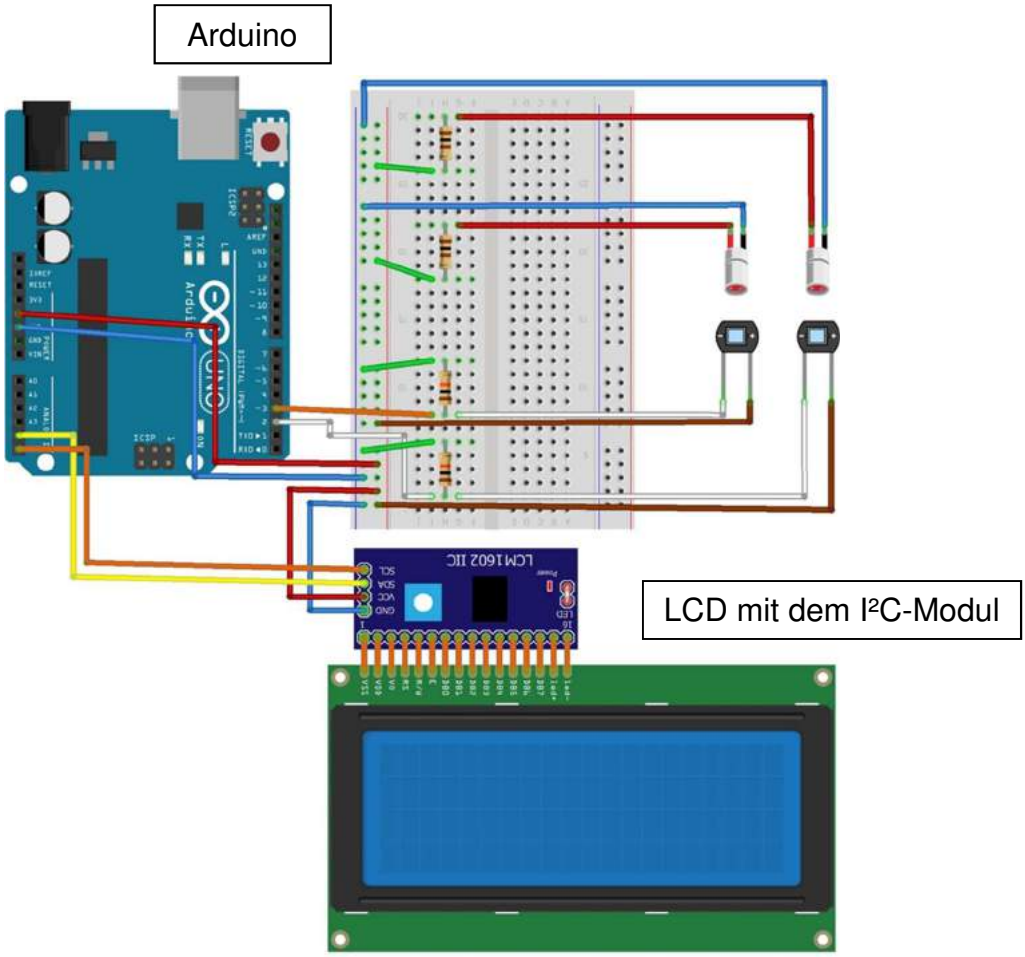


Abbildung 86: Messaufbau zur Messung der Anfangsgeschwindigkeit des Tennisballes

Die Laser und die Fotodioden werden paarweise auf zwei Stahlstangen in einem bestimmten Abstand zueinander am vorderen Teil der Maschine montiert.

Die Vorrichtung ist in Abbildung 87 ersichtlich.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

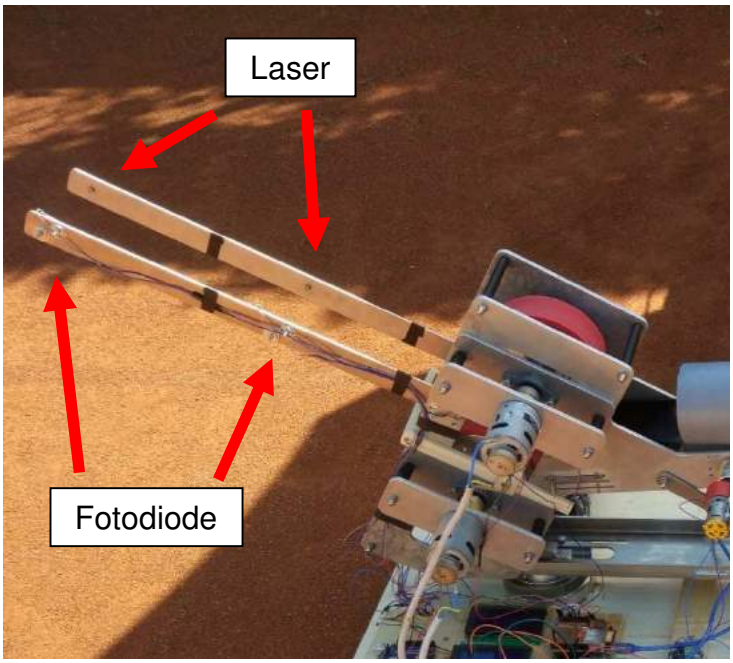


Abbildung 87: Vorrichtung für die Messung der Anfangsgeschwindigkeit auf der Maschine

Messdurchführung

Ziel dieser Messung war es, einen Zusammenhang der Anfangsgeschwindigkeit des Balles mit der Drehzahl der Motoren und in weiterer Folge mit der Flugweite des Balles auf dem Tennisplatz zu finden.

Da die Eigenschaften des Balles einen Einfluss auf die Flugbahn haben, wurden unterschiedliche Balldosen mit je drei oder vier Bällen verwendet. Diese Bälle unterschieden sich von der Marke und dem Alter (Intervall zwischen Öffnen der Dose bis zur Messung). Um die Bälle voneinander unterscheiden zu können, wurden sie – wie in Abbildung 88 abgebildet – markiert.

Abbildung 88: Markierte Tennisbälle für die Messung

Dabei beinhaltet die Ballserie 1 (links in Abbildung 88) die ältesten Bälle; bei dieser Serie wurde die Dose vor ungefähr fünf Monaten geöffnet. Die zweit- bzw. dritt- älteste Serie sind die Serien 3 bzw. 4. Die entsprechenden Dosen wurden vor ungefähr drei bzw. zwei Monaten geöffnet. Die Serie 7 ist mit einem Alter von etwa drei Wochen die neueste Serie.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Nach dem entsprechenden Markieren der Bälle wurden bei den DC-Motoren unterschiedliche PWM-Werte eingestellt und die Anfangsgeschwindigkeit und die Flugweite des Balles notiert. Diese Messungen wurden anfangs ausschließlich mit der Ballserie 7 durchgeführt.

Daten

Bei der Messung wurde zwischen den beiden Spielsituationen *Top-Spin* und *kein Spin* unterschieden. Bei der Spielsituation *Top-Spin* unterscheiden sich die PWM-Werte der DC-Motoren um 25%. Bei dem Fall, in dem kein Spin erwünscht ist, unterscheiden sie sich um 10% voneinander. Diese Differenz muss gewählt werden, um einen sogenannten „Flatterball", wie er auch aus dem Fußball bekannt ist, zu vermeiden.

Die gemessenen Anfangsgeschwindigkeiten und Flugweiten der Spielsituation *Top-Spin* sind in der Tabelle 12 dargestellt.

Ball	**PWM oben**	**240**	**220**	**200**	**180**	**170**	**165**	**160**	**155**	**150**	**145**	**140**
	PWM unten	**192**	**176**	**160**	**144**	**136**	**132**	**128**	**124**	**120**	**116**	**112**
7.1	Geschw. (km/h)	74,90	75,03	74,43	72,86	69,10	67,37	64,62	61,90	58,99	55,42	51,89
	Flugweite (m)	20,5	21	20,4	19,5	17	16,8	16,8	15,4	14,5		
7.2	Geschw. (km/h)	75,55	75,18	74,98	72,84	68,89	67,10	64,71	62,19	58,99	55,05	51,77
	Flugweite (m)	20,25	20,5	20,4	19,5	17	17,1	16	15,1	14,5		
7.3	Geschw. (km/h)	75,25	75,20	74,85	72,12	68,35	67,18	64,79	61,90	58,96	55,27	51,98
	Flugweite (m)	20,5	20,5	20,4	19	16,8	17	16,4	15	14,4		
	gemittelte Geschw. (km/h)	**75,23**	**75,14**	**74,75**	**72,61**	**68,78**	**67,22**	**64,71**	**62,00**	**58,98**	**55,25**	**51,88**
	gemittelte Flugweite (m)	**20,4**	**20,7**	**20,4**	**19,3**	**16,9**	**17,0**	**16,4**	**15,2**	**14,5**		

Tabelle 12: Anfangsgeschwindigkeit und Flugweiten bei unterschiedlichen PWM-Werten (*Top-Spin*)

Die gemessenen Daten der Situation *kein Spin* sind in der Tabelle 13 dargestellt.

Ball	**PWM oben**	**240**	**220**	**200**	**180**	**170**	**165**	**160**	**155**	**150**	**145**	**140**
	PWM unten	**218**	**200**	**182**	**164**	**155**	**150**	**145**	**141**	**136**	**132**	**127**
7.1	Geschw. (km/h)	76,45	77,00	75,53	74,06	73,67	71,82	69,53	67,41	65,27	62,28	58,84
	Flugweite (m)	21,5	22	21	20	20,25	19,5	18,5	18	17,25	15,75	14,75
7.2	Geschw. (km/h)	74,70	76,25	75,53	74,55	73,55	71,18	68,91	67,96	64,97	62,10	57,89
	Flugweite (m)	21	21,5	21	20,75	20,25	19	18,75	18	17	15,75	14,75
7.3	Geschw. (km/h)	77,08	75,25	74,93	73,82	73,92	71,82	69,53	67,93	64,51	61,95	58,17
	Flugweite (m)	21,5	21,5	20,75	20	20	19,25	18,5	18	16,75	15,6	14,5
	gemittelte Geschw. (km/h)	**76,08**	**76,17**	**75,33**	**74,14**	**73,71**	**71,61**	**69,32**	**67,77**	**64,92**	**62,11**	**58,30**
	gemittelte Flugweite (m)	**21,3**	**21,7**	**20,9**	**20,3**	**20,2**	**19,3**	**18,6**	**18,0**	**17,0**	**15,7**	**14,7**

Tabelle 13: Anfangsgeschwindigkeit und Flugweiten bei unterschiedlichen PWM-Werten (*kein Spin*)

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Auswertung

Aus diesen gemessenen Werten wird einerseits nun ein Zusammenhang zwischen den eingestellten PWM-Werten und der Anfangsgeschwindigkeit des Balles gesucht und anderseits ein Zusammenhang zwischen den eingestellten PWM-Werten bei den DC-Motoren und der Flugweite der Tennisbälle hergestellt. Bei diesen Zusammenhängen muss erneut zwischen den beiden Fällen *Top-Spin* und *kein Spin* unterschieden werden.

In Abbildung 89 sind die Flugweiten in Abhängigkeit von den eingestellten PWM-Werten dargestellt.

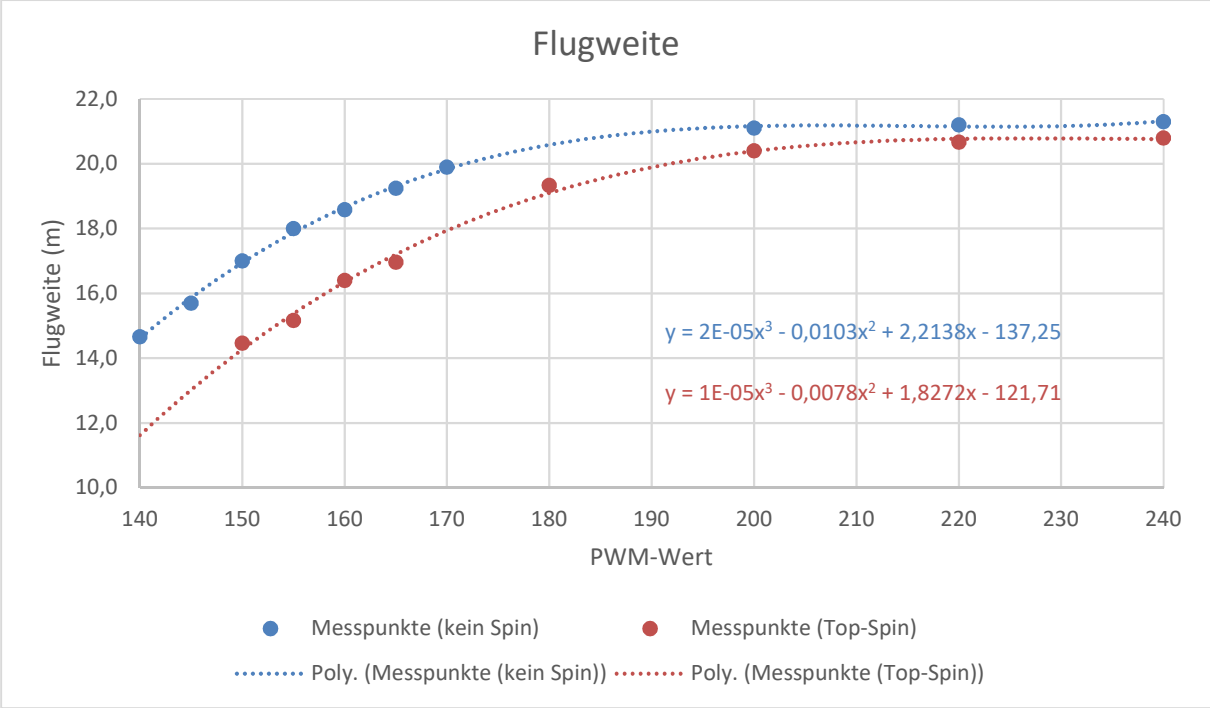


Abbildung 89: Zusammenhang zwischen den eingestellten PWM-Werten und der Flugweite

Die roten Messpunkte beschreiben die Spielsituation *Top-Spin* und die blauen die Situation *kein Spin*. Durch polynomiale Regression kann zu jeder Spielsituation eine Funktion (strichlierte Linie) ermittelt werden, die den Zusammenhang zwischen den eingestellten PWM-Werten und der Flugweite beschreibt.

Demnach beschreibt die folgende Formel den Zusammenhang bei einem Top-Spin

$$y = 2 \cdot 10^{-6} \cdot x^3 - 0{,}0025 \cdot x^2 + 0{,}8084 \cdot x - 57{,}52 \qquad (5.25)$$

und die Formel 5.26 den Zusammenhang bei keinem Spin.

$$y = 10^{-5} \cdot x^3 - 0{,}0095 \cdot x^2 + 2{,}0616 \cdot x - 127{,}71 \qquad (5.26)$$

Dabei ist y definiert als die Flugweite in Meter und x als der PWM-Wert des oberen DC-Motors. „Ausreißer“ der Messreihe wurden bei der Regression nicht berücksichtigt.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub

Auf die gleiche Weise wird dieser Zusammenhang zwischen den eingestellten PWM-Werten und der Anfangsgeschwindigkeit des Balles ermittelt. Die Messpunkte dazu sind in Abbildung 90 ersichtlich.

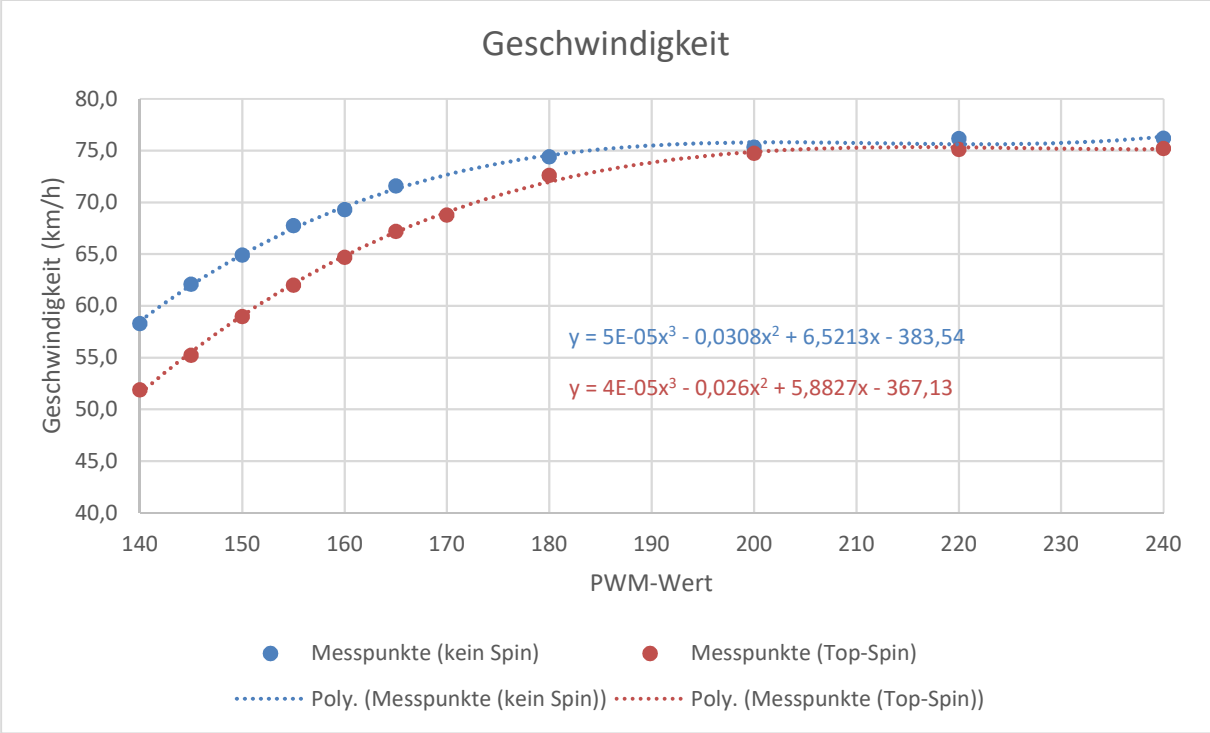


Abbildung 90: Zusammenhang zwischen den eingestellten PWM-Werten und der Anfangsgeschwindigkeiten

In diesem Fall beschreibt die Formel 5.27 den Zusammenhang bei einem Top-Spin.

$$y = 2 \cdot 10^{-6} \cdot x^3 - 0{,}0025 \cdot x^2 + 0{,}8084 \cdot x - 57{,}52 \qquad (5.27)$$

Die Formel 5.28 beschreibt den Zusammenhang bei der Spielsituation *kein Spin*.

$$y = 10^{-5} \cdot x^3 - 0{,}0095 \cdot x^2 + 2{,}0616 \cdot x - 127{,}71 \qquad (5.28)$$

Hier ist y die Anfangsgeschwindigkeit (in km/h) und x der PWM-Wert des oberen DC-Motors. Auch in diesem Fall wurden die „Ausreißer" der Messreihe bei der Regression nicht berücksichtigt.

Des Weiteren wird bei einer PWM-Einstellung von 200 des oberen DC-Motors und von 160 des unteren DC-Motors das Verhalten der unterschiedlichen Ballserien getestet. Die Ergebnisse dieser Messung sind in Tabelle 14 ersichtlich.

Ball	1.1	1.2	1.3	1.4	3.1	3.2	3.3	4.1	4.2	4.3	4.4
Geschw. (km/h)	73,46	74,62	74,19	72,92	73,63	75,13	74,95	73,28	74,92	74,96	74,58
Flugweite (m)	23,5	23,5	23	23	22,5	23	22,5	22	21,5	21	23

Tabelle 14: Anfangsgeschwindigkeit und Flugweiten bei unterschiedlichen Ballserien

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Bei Betrachtung der Messdaten aus Tabelle 14 fällt auf, dass alle Tennisbälle eine sehr ähnliche Anfangsgeschwindigkeit aufweisen, aber die einzelnen Flugweiten stark differieren. Dabei ist das Schema zu erkennen, dass die ältesten Tennisbälle die weitesten und die neuesten Bälle die kürzeste Flugweite aufweisen.

Um diesen Umstand zu veranschaulichen, wurde ein zusätzliches Experiment durchgeführt. Dabei wurden zuerst je sechs Tennisbälle der Ballserie 7 mit den oben genannten PWM-Einstellungen abgeschossen und anschließend sechs Bälle der Serie 1.

Die Positionen des Auftreffens werden jeweils mit anderen Tennisbällen markiert.

Der Weitenunterschied ist in Abbildung 91 deutlich erkennbar.

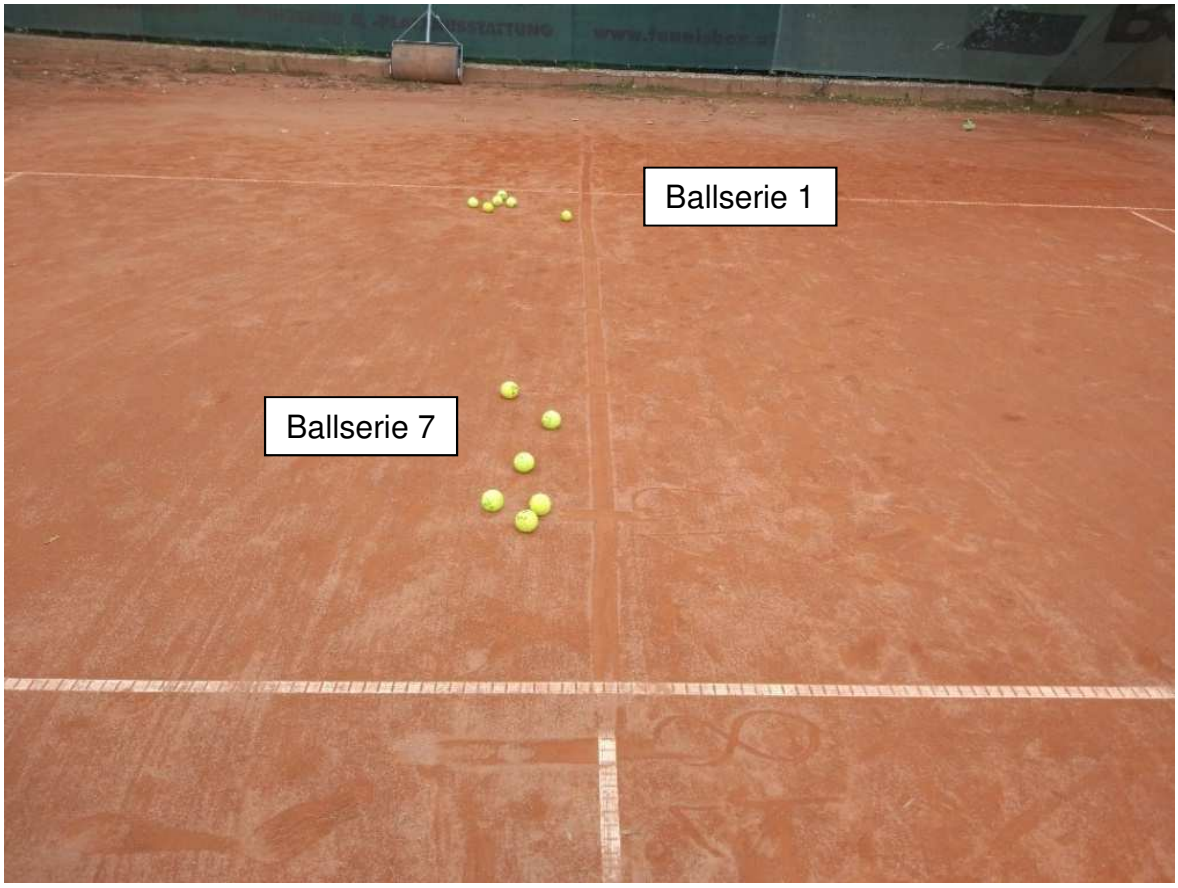


Abbildung 91: Flugweiten der Ballserie 1 und 7

Des Weiteren ist auf Abbildung 91 erkennbar, dass die Auftreffpunkte bei der Ballserie 7 auf einen Radius von 50 cm und bei der Ballserie 1 nur auf einen Radius von 30 cm verteilt waren.

Um die Ursache für die Weitenunterschieden der unterschiedlichen Ballserien zu eruieren, wurde ein weiterer Versuchsaufbau vorgenommen. Bei diesem Aufbau, welcher in Abbildung 92 ersichtlich ist, wurden die Tennisbälle der unterschiedlichen

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Ballserien aus einer Höhe von 80 cm fallen gelassen. Mithilfe einer Kamera wurden die Rücksprunghöhen der einzelnen Bälle aufgezeichnet und danach ermittelt.

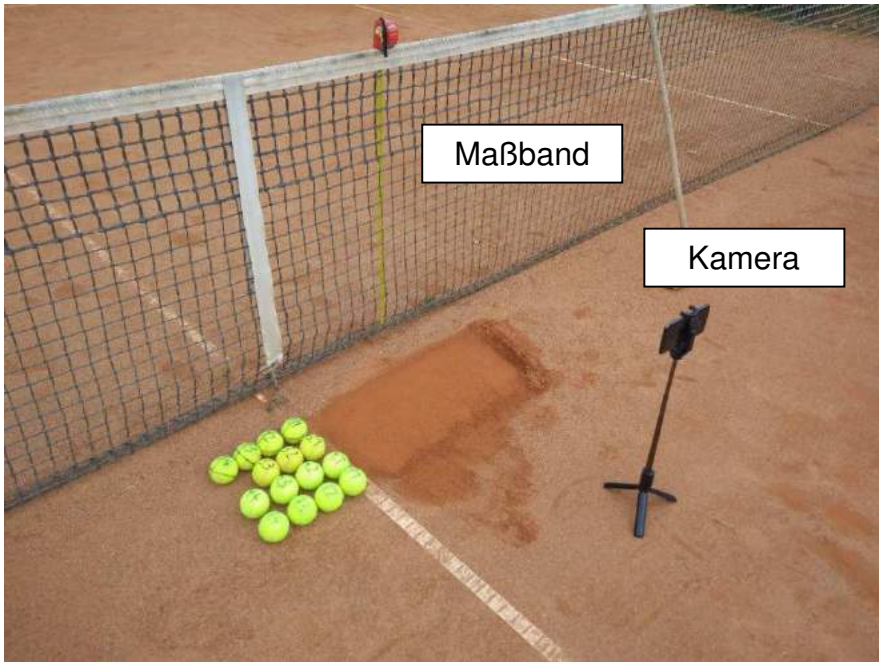


Abbildung 92: Versuchsaufbau: Messung der Rücksprunghöhe

Die Rücksprunghöhen von den einzelnen Bällen der unterschiedlichen Ballserien sind in Tabelle 15 notiert.

Ball	1.1	1.2	1.3	1.4	3.1	3.2	3.3	4.1	4.2	4.3	4.4	7.1	7.2	7.3
Rücksprunghöhe (m)	69	67	70	71	67	67	67	65	63	61	63	59	60	60

Tabelle 15: Rücksprunghöhe der unterschiedlichen Tennisbälle

Aus dieser Tabelle ist ersichtlich, dass die Rücksprunghöhe bei den ältesten Bällen (Ballserie 1) am höchsten und bei den neuesten Bällen (Ballserie 7) am niedrigsten ist.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Messung des Spins bei unterschiedlichen Drehzahlen

Bei dieser Messung wurde der Spin der Tennisbälle bei unterschiedlichen Drehzahlen der DC-Motoren ermittelt und ein Zusammenhang hergestellt.

Messvorbereitung

Um die Messung durchführen zu können, mussten vorerst einige Vorbereitungen getroffen werden:

- Im ersten Schritt wurde ein Tennisball mit einer Markierung (hier: einem schwarzen Punkt, siehe
- Abbildung 93) versehen.
- Im nächsten Schritt wurde die Kamera GoPro Hero 4 in einem Abstand von ungefähr zehn Meter Entfernung von der Maschine positioniert.

Abbildung 93: Markierter Ball

Messdurchführung

Nach diesen Vorbereitungen wurden bei den DC-Motoren unterschiedliche Drehzahlen eingestellt und der markierte Ball wurde mit der Markierung nach vorne in die Maschine gelegt und abgeschossen.

Mithilfe der Kamera wurden die Flugbahn und die Umdrehungen des Tennisballes aufgenommen.

Auswertung

Um die Drehzahl des Balles zu finden, wurden die aufgenommenen Videos im Anschluss ausgewertet. Dazu wurden die einzelnen Frames der Aufnahmen während einer Umdrehung des Tennisballes um seine Achse abgezählt. Diese Daten sind in Tabelle 16 mit der Spielsituation *Top-Spin* und in Tabelle 17 der Fall, in dem kein Spin erwünscht ist, dargestellt.

PWM oben	**240**	**200**	**170**	**160**	**155**	**150**
PWM unten	192	160	136	128	124	120
U/frame	2,5	3,5	4,5	5	5,5	6
Drehzahl (U/min)	1440	1029	800	720	655	600

Tabelle 16: Drehzahl des Balles bei unterschiedlichen PWM-Werten (*Top-Spin*)

PWM oben	**220**	**180**	**165**	**150**	**140**
PWM unten	200	164	150	136	127
U/frame	7	8	11	12	14
Drehzahl (U/min)	514	450	327	300	257

Tabelle 17: Drehzahl des Balles bei unterschiedlichen PWM-Werten (*kein Spin*)

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Mit diesen ermittelten Drehzahlen des Balles konnte für jede Spielsituation ein Zusammenhang zwischen diesen Drehzahlen und der Drehzahlen der DC-Motoren hergestellt werden.

In Abbildung 94 ist dieser Zusammenhang graphisch dargestellt.

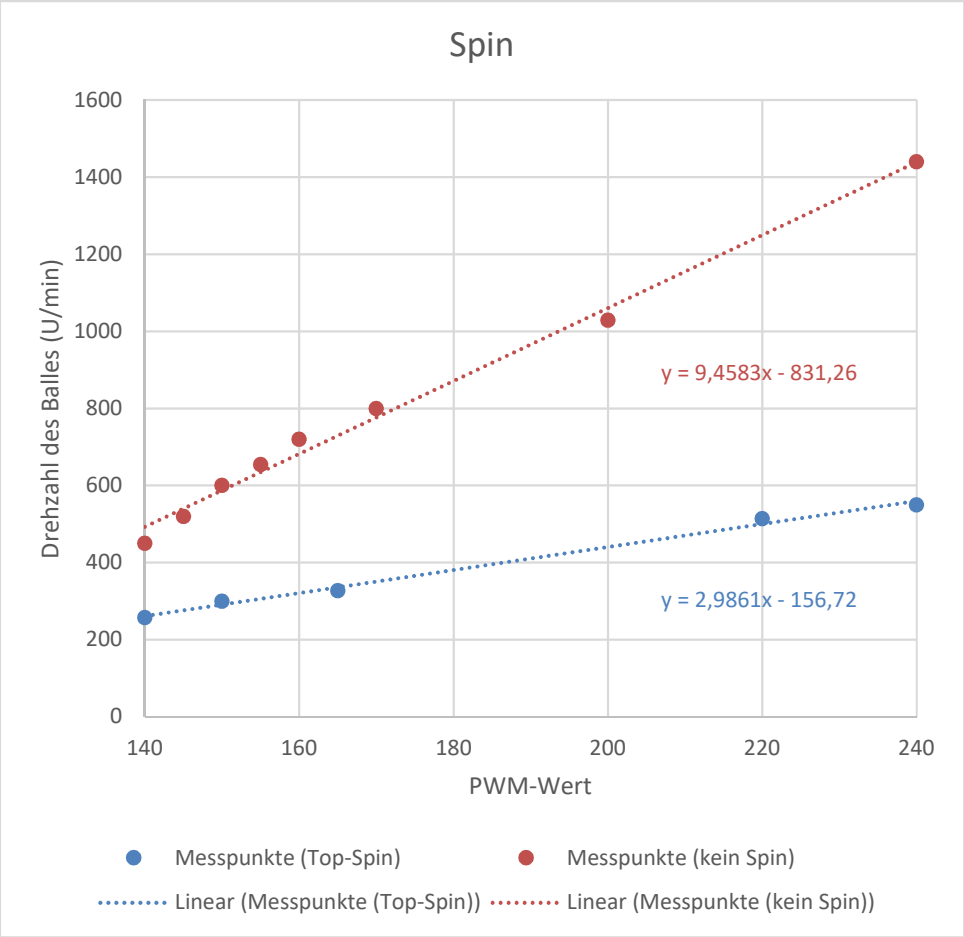


Abbildung 94: Zusammenhang zwischen den eingestellten PWM-Werten und der Drehzahl des Balles

Auch hier gibt die rote strichlierte Linie eine lineare Annäherung der Messpunkte bei einem Top-Spin und die blaue bei der Spielsituation *kein Spin* wieder.

Die Formel 5.29 beschreibt den Zusammenhang bei einem Top-Spin und die Formel 5.30 bei keinem Spin.

$$y = 9{,}0647 \cdot x - 750{,}24 \tag{5.29}$$

$$y = 3{,}3201 \cdot x - 198 \tag{5.30}$$

Dabei ist y als die Drehzahl des Balles (in U/min) und x der PWM-Wert des oberen DC-Motors definiert. „Ausreißer“ der Messreihe wurden bei der Regression nicht berücksichtigt.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Berechnung der tatsächlichen Verlustenergie bei einem Schussvorgang

In dem Kapitel 5.1.6 wurde die Verlustenergie mit 25% abgeschätzt. Diese sollen nun in der nachfolgenden Berechnung exakt ermittelt werden.

Die Berechnungen werden bei einer PWM-Einstellung des oberen Motors von 240 und des unteren Motors von 240 durchgeführt. Der mögliche Spin des Balles wurde dabei nicht berücksichtigt.

Zuerst muss die kinetische Energie vor und unmittelbar nach dem Abschuss berechnet werden. Dazu werden jeweils die Drehzahlen des oberen Schwungrades benötigt. Diese können aus dem Diagramm in Abbildung 84 abgelesen werden und betragen $\omega_{vor} = 4280\ U/min$ und $\omega_{nach} = 4185\ U/min$.

Die kinetischen Energien lauten dann

$$E_{R\_vor\_schuss} = \left(\frac{1}{2} \cdot I_R \cdot \omega_{vor}^2\right) \cdot 2 = I_R \cdot \omega_{vor}^2 = 3573{,}2 \cdot 10^{-6}\ kgm^2 \cdot \left(4280\ \frac{U}{min}\right)^2$$

$$\boldsymbol{E_{R\_vor\_schuss} = 717{,}796\ \mathrm{J}}$$

$$E_{R\_nach\_schuss} = \left(\frac{1}{2} \cdot I_R \cdot \omega_{nach}^2\right) \cdot 2 = I_R \cdot \omega_{nach}^2 = 3573{,}2 \cdot 10^{-6}\ kgm^2 \cdot \left(4185\ \frac{U}{min}\right)^2$$

$$\boldsymbol{E_{R\_nach\_schuss} = 686{,}286\ \mathrm{J}}$$

Nun wird die Differenz der beiden Energien ermittelt.

$$E_{diff} = E_{R\_vor\_schuss} - E_{R\_nach\_schuss} = 717{,}796\ \mathrm{J} - 686{,}286\ \mathrm{J}$$

$$\boldsymbol{E_{diff} = 31{,}51\ \mathrm{J}}$$

Bei E_{diff} handelt es sich um die Energie, welche während des Abschusses von den Schwungrädern abgegeben wird. Diese Energie setzt sich wiederum von der Energie, welche dem Ball mitgegeben wird, und der Verlustenergie zusammen.

Um die Energie, welche der Ball unmittelbar nach dem Verlassen der Maschine hat, ermitteln zu können, muss zuerst die Anfangsgeschwindigkeit des Balles festgestellt werden. Die Anfangsgeschwindigkeit kann aus der Tabelle 13 entnommen werden und beträgt bei einem PWM-Wert des oberen Schwungrades von 240 und der Spielsituation *kein Spin* $v_B = 76{,}08\ \frac{km}{h}$.

Die kinetische Energie der translatorischen Bewegung beträgt dann

$$\boldsymbol{E_{B_t} = \frac{1}{2} \cdot m_B \cdot v_B^2 = \frac{1}{2} \cdot 0{,}0575\ kg \cdot \left(76{,}08\ \frac{km}{h}\right)^2 = 12{,}836\ \mathrm{J}}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Für die Verlustenergie folgt

$$\boldsymbol{E_{Verlust}} = E_{diff} - E_{B_t} = 31{,}51\,\text{J} - 12{,}836\,\text{J} = \mathbf{18{,}674\,J}$$

Die Formel der kinetischen Energie kann analog auch für die Verlustenergie $\boldsymbol{E_{Verlust}}$ eingesetzt werden.

$$E_{Verlust} = \left(\frac{1}{2} \cdot I_R \cdot \omega_{Verlust}^2\right) \cdot 2 = I_R \cdot \omega_{Verlust}^2$$

Nun kann nach der fehlenden Variablen $\omega_{Verlust}$ umgeformt und diese berechnet werden.

$$\boldsymbol{\omega_{Verlust}} = \sqrt{\frac{E_{Verlust}}{I_R}} = \sqrt{\frac{18{,}674\,\text{J}}{3573{,}2 \cdot 10^{-6}\ kgm^2}} = \mathbf{690{,}34\ \frac{U}{min}}$$

Daraus lässt sich schließen, dass bei einem PWM-Wert von 240 und bei einem Abschuss ungefähr $700\frac{U}{min}$ durch Reibung und Wärme verloren gehen. Um diesen Wert prozentuell angeben zu können, muss zunächst der gleiche Ansatz wie in Kapitel 5.1.6 gewählt werden und die Winkelgeschwindigkeit der Schwungräder berechnet werden.

$$\boldsymbol{\omega_R} = \frac{v_R}{r_R} = \frac{76{,}08\ \frac{km}{h}}{0{,}07\ m} = \mathbf{2882{,}52\ \frac{U}{min}}$$

Nun kann die gesamte benötigte Winkelgeschwindigkeit des Schwungrades bei einem PWM-Wert von 240 ermittelt werden.

$$\boldsymbol{\omega_{R\_Gesamt}} = \omega_R + \omega_{Verlust} = 2882{,}52\ \frac{U}{min} + 690{,}34\ \frac{U}{min} = \mathbf{3572{,}86\ \frac{U}{min}}$$

Damit dieses Ergebnis mit dem Ansatz bzw. dem Ergebnis in Kapitel 5.1.6 verglichen werden kann, muss nun der prozentuelle Anteil von der Verlustenergie angegeben werden.

Dieser beträgt $\boldsymbol{p} = \frac{690{,}34\ \frac{U}{min}}{2882{,}52\ \frac{U}{min}} = \mathbf{23{,}95\ \%}$.

Demnach war die Annahme in Kapitel 5.1.6 mit 25 % eine gute Annäherung.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

5.2.6 Elektronik

Einführung

Um die Ziele von dem Funktionsmuster (Kapitel 5.2.1) bestmöglich zu erfüllen, wurde versucht, die in den Versuchsaufbauten beschriebenen Elektronik-Teile in einem Verbindungsschema unterzubringen und zusammenzuführen. Demnach weist dieses eine gewisse Komplexität auf.

Benötigte Bauteile

(1) x Raspberry Pi 3 Model B+
(1) x Arduino UNO R3
(1) x USB-Kabel
(1) x HDMI-Kabel
(1) x Monitor Touchscreen Display LCD (7 Zoll)
(1) x 600W Netzteil 12V/50A
(3) x Motortreiber Cytron 13A
(2) x Gleichstrommotor RS PRO 12V
(1) x Hybrid-Schrittmotor Nema 23 (QSH6018-65-28)
(1) x Schritt-Motortreiber 4A 9-42V (2/4 Phasen)
(1) x RS PRO DC-Getriebemotor, 4,5 V / 19,8 W
(1) x Hall Sensor Modul
(1) x Magnet
(1) x Laser Rot 650 nm 5V
(1) x Fotodiode BPW 34
(1) x 100 Ω Widerstand
(1) x 10 kΩ Widerstand

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Verbindungsschema

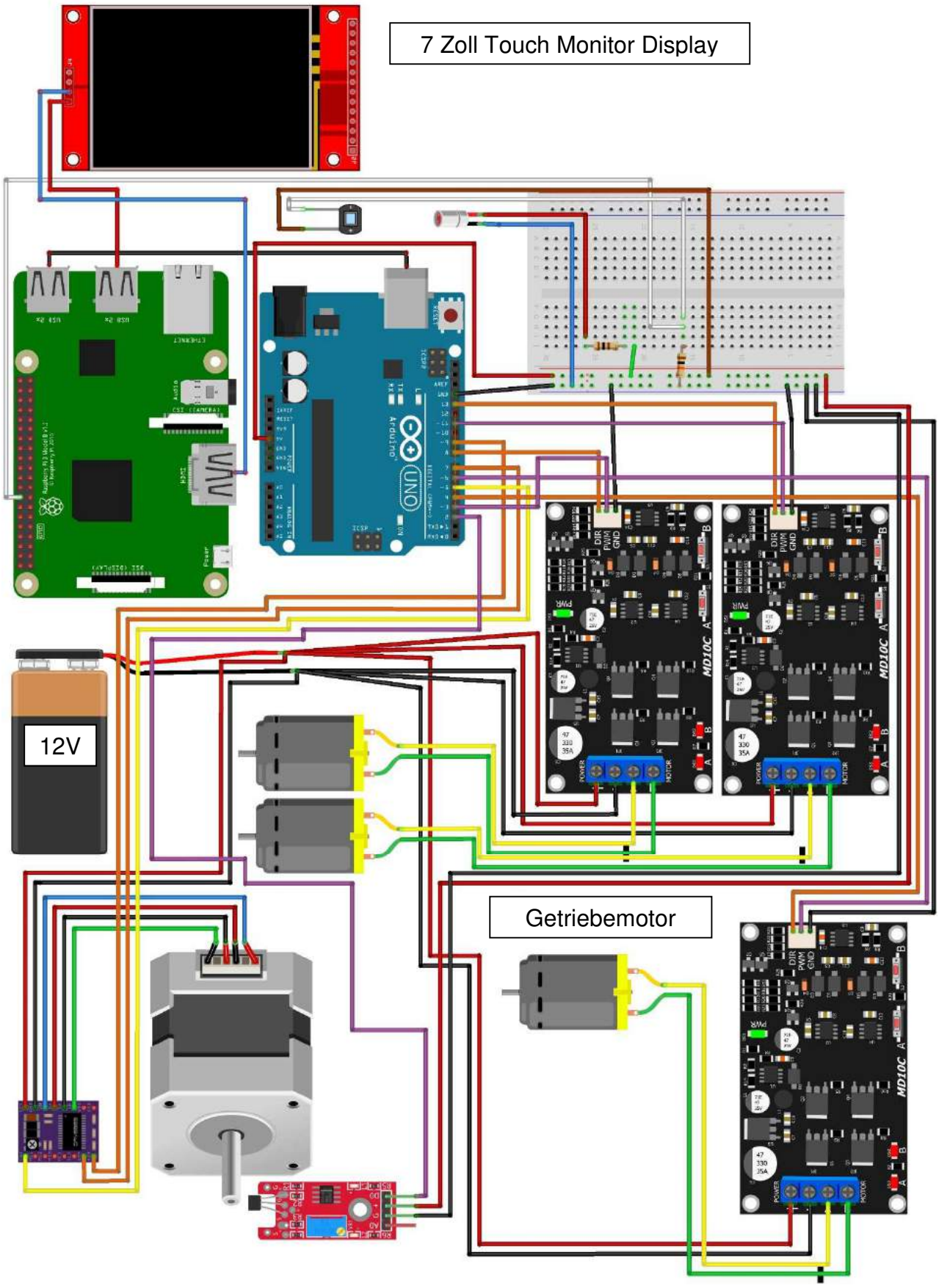


Abbildung 95: Funktionsmuster – Verbindungsschema

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

In Abbildung 95 ist das Verbindungsschema des Funktionsmusters dargestellt.

Der Raspberry PI und der Arduino sind über ein USB-Kabel miteinander verbunden. Der 7 Zoll Touch Monitor Display ist mit dem Raspberry PI einerseits über ein HDMI-Kabel verbunden, welches zur Datenübertragung benötigt wird und anderseits über eine USB-Schnittstelle, welche die Stromversorgung sicherstellt.

Um die Komplexität des Aufbaus des Funktionsmusters zu vereinfachen, wurde für die zwei DC-Motoren, welche die Schwungräder antreiben, und dem Getriebemotor, welcher für die Bewegung des Ballrades verantwortlich ist, der gleiche Motortreiber für die Steuerung verwendet.

Wie auf der Abbildung weiters ersichtlich ist, sind bei dem Arduino nahezu alle digitalen PINs belegt. Das Schema der Anschlüsse sieht folgendermaßen aus:

- PIN 2 → Hall-Sensor für Schrittmotor (INPUT)
- PIN 3 → PWM-Anschluss Motortreiber 1 (OUTPUT)
- PIN 4 → DIR-Anschluss Motortreiber 3 (OUTPUT)
- PIN 5 → ENABLE-Anschluss Schrittmotor (OUTPUT)
- PIN 6 → PWM-Anschluss Motortreiber 3 (OUTPUT)
- PIN 7 → STEP-Anschluss Schrittmotor (OUTPUT)
- PIN 8 → DIR-Anschluss Motortreiber 1 (OUTPUT)
- PIN 9 → DIR-Anschluss Schrittmotor (OUTPUT)
- PIN 11 → PWM-Anschluss Motortreiber 2 (OUTPUT)
- PIN 13 → DIR-Anschluss Motortreiber 2 (OUTPUT)

Bei dem Raspberry PI wird nur der Anschluss GPIO 23, welcher mit der Fotodiode verbunden ist, verwendet.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Aufbau am Funktionsmusters

In Abbildung 96 ist das Funktionsmuster mit der kompletten Elektronik dargestellt.

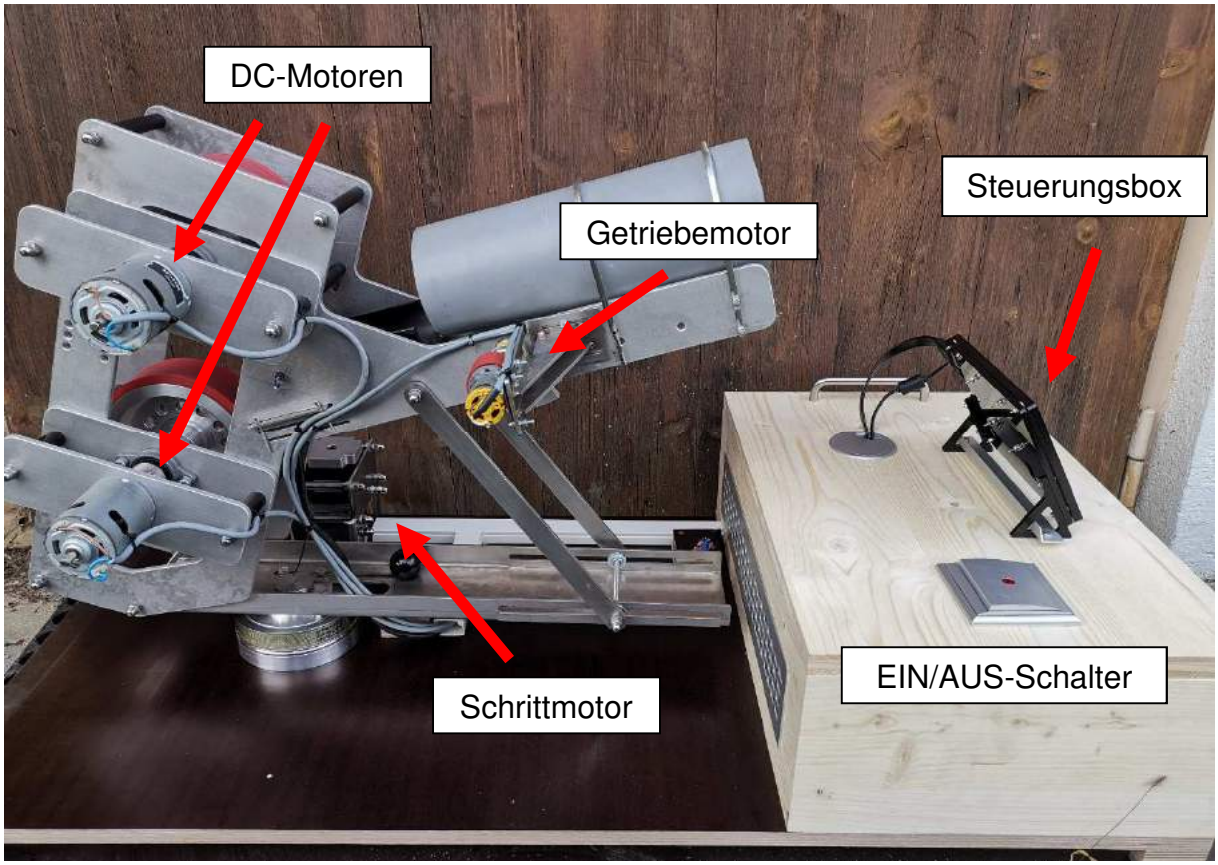


Abbildung 96: Funktionsmuster mit der kompletten Elektronik

Des Weiteren ist in der Abbildung 96 ersichtlich, dass am Funktionsmuster eine Steuerungsbox angebracht wurde. In dieser Box befinden sich alle Elemente, die notwendig sind, um die Motoren ansteuern zu können.

Auf der Box befindet sich ein EIN/AUS – Schalter, mit dem die Stromzufuhr zur Maschine ein- bzw. ausgeschaltet werden kann. Zusätzlich befindet sich am Deckel der Box der Touchscreen Display, mit dem die Maschine vom Benutzer gesteuert wird.

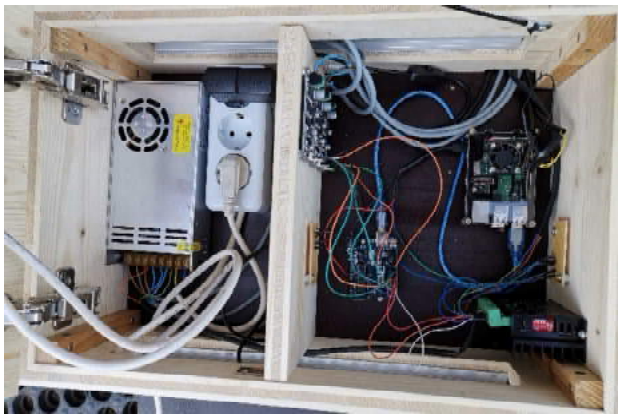


Abbildung 97: Steuerungsbox am Funktionsmuster

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Wie in der Abbildung 97 erkennbar ist, ist die Box im Inneren in zwei Bereich unterteilt.

Auf der linken Seite der Box (siehe Abbildung 98) befinden sich alle Elemente, die für die Energieversorgung notwendig sind. In dieser Box darf bei eingeschaltetem Zustand nicht hantiert werden. (Spannungen von 230 Volt). Generell sollte die Box, wenn die Maschine unter Spannung steht, nicht geöffnet werden.

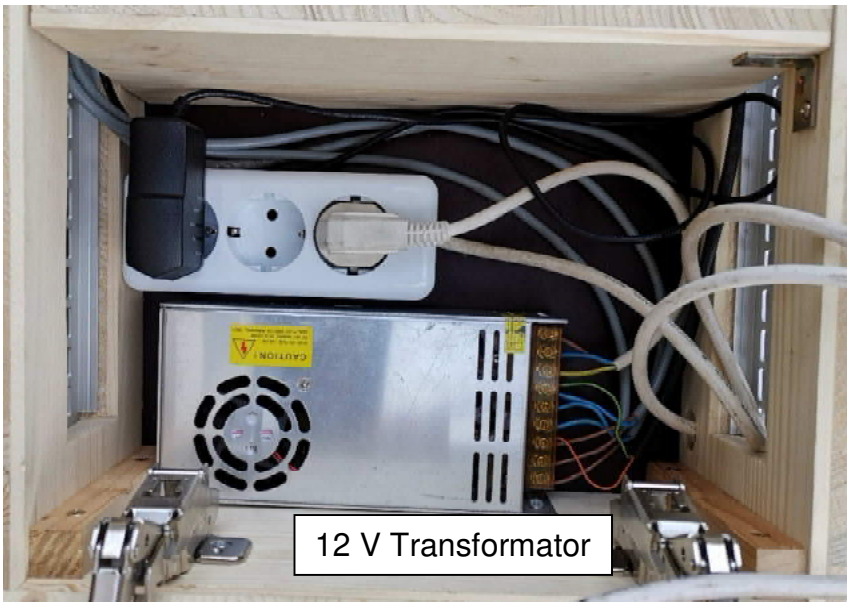


Abbildung 98: Linke Abteilung der Steuerungsbox

Im Gegensatz dazu treten in der rechten Box nur mehr Spannungen bis zu 12 Volt auf. In diesem Teil der Box befinden sich die Steuerungselemente, wie Motortreiber, Arduino und Raspberry PI (siehe Abbildung 99).

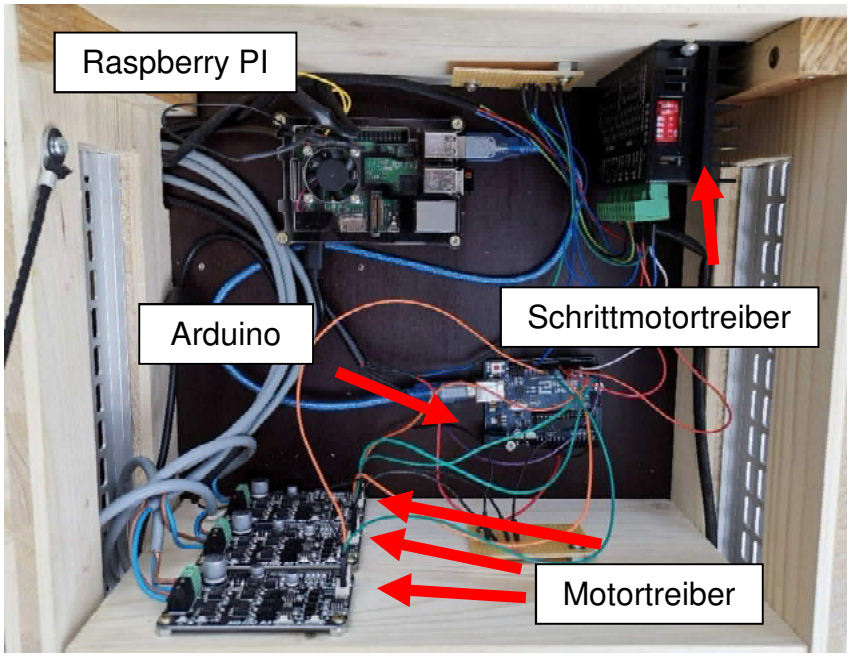


Abbildung 99: Rechte Abteilung der Steuerungsbox

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

5.2.7 Programmierung & Software

In diesem Kapitel wird zunächst auf die Programmierung eingegangen, welche einerseits auf dem Arduino und andererseits auf dem Raspberry PI stattgefunden hat. Anschließend wird die Software-Lösung, welche das Funktionsmuster ansteuert, im Detail erklärt.

Programmierung Raspberry PI

Im ersten Schritt musste eine Programmiersprache festgelegt werden, mit der das gewünschte Vorhaben umgesetzt werden kann. Zur näheren Auswahl standen die Sprachen C++ und Python, wobei die Programmiersprache Python gewählt wurde. Der ausschlaggebende Grund für diese Auswahl war vor allem die leichtere Umsetzbarkeit.

Des Weiteren wurde für die Realisierung der graphischen Oberfläche das *Framework Qt5* gewählt. Um die Programmierung dieser graphischen Oberfläche zu erleichtern, wurde diese mit dem Programm *QtCreator* umgesetzt. *QtCreator* diente dabei lediglich zur Auswahl und Positionierung der Elemente auf der Oberfläche. Ein Bildschirmabzug dieses Programms ist in Abbildung 100 zu finden. Die Funktion der Elemente wurde wiederum im Programm Python ausprogrammiert.

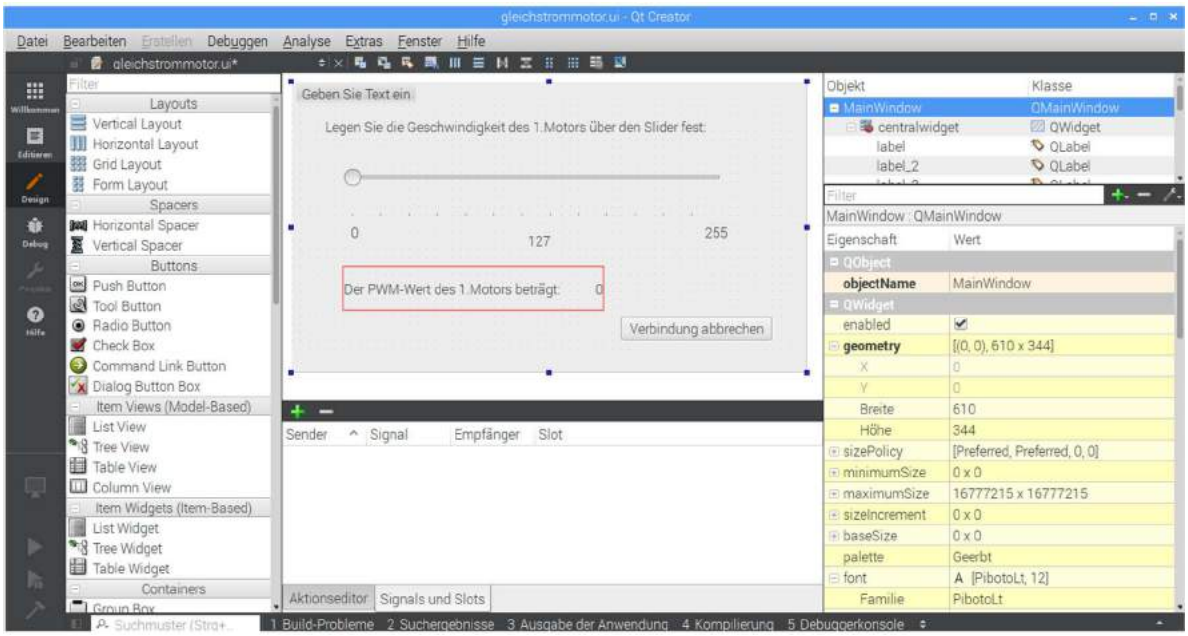


Abbildung 100: Programm *QtCreator*

Das Programm in Python wurde mithilfe der integrierten Entwicklungsumgebung *Spyder* erstellt (siehe Abbildung 101).

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

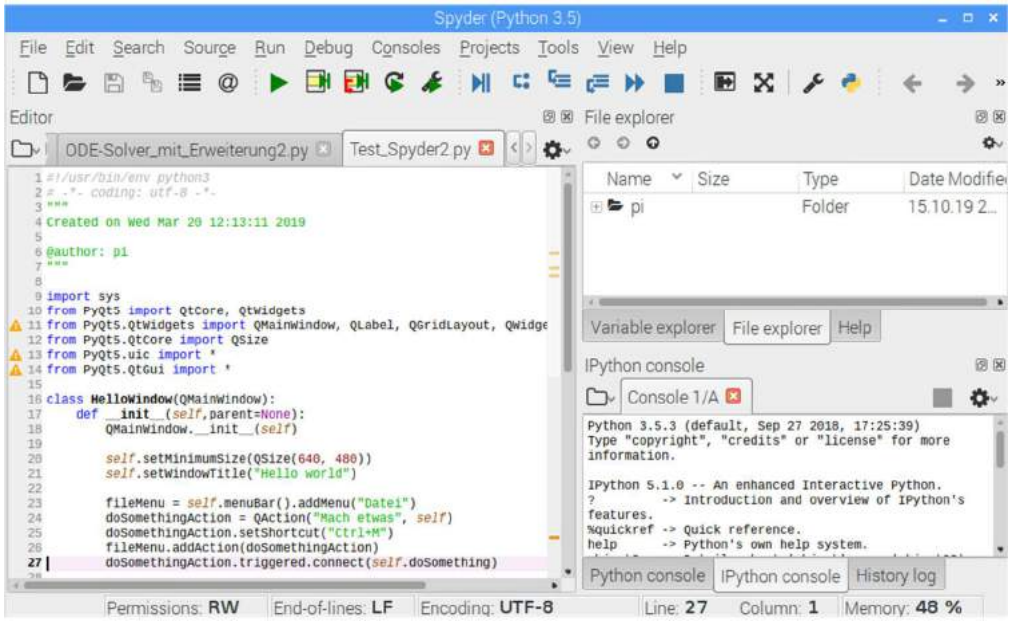


Abbildung 101: Integriete Entwicklungsumgebung *Spyder*

Programmierung Arduino

Das Programm am Arduino wurde mithilfe der Programmiersprache *Processing* umgesetzt. Die Abbildung 102 zeigt einen Screenshot der Entwicklungsumgebung.

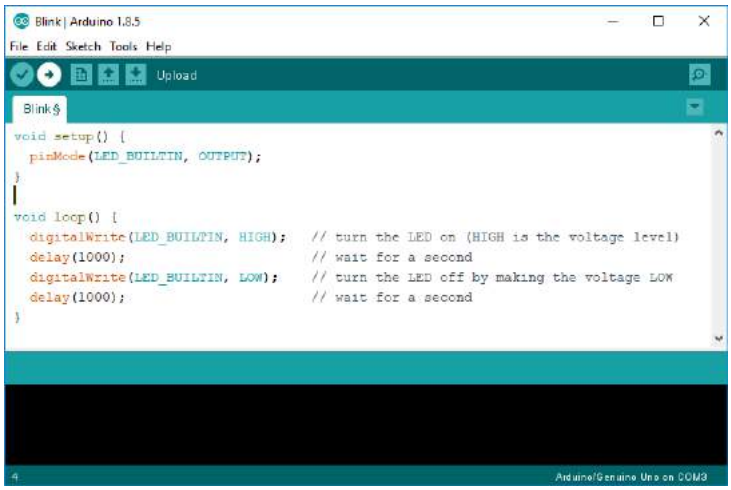


Abbildung 102: Programm am Arduino

Um ein funktionstüchtiges Programm erzeugen zu können, müssen folgende zwei Funktionen zu definiert sein:

- `setup()` – wird beim Start des Programms einmalig aufgerufen. In dieser Funktion wird beispielweise den PINs zugeteilt, ob es dabei um einen OUTPUT oder INPUT handelt.
- `loop()` – wird immer wieder durchlaufen, solange das Arduino-Board mit Strom versorgt ist.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Software

In Abbildung 95 ist ersichtlich, dass der Raspberry PI das zentrale Glied ist, von dem aus alle anderen Elemente angesteuert werden. Demnach wurde das Hauptprogramm mit der Benutzeroberfläche auf dem Raspberry PI umgesetzt. Dieses Programm besteht aus folgenden Teilen (siehe Abbildung 103):

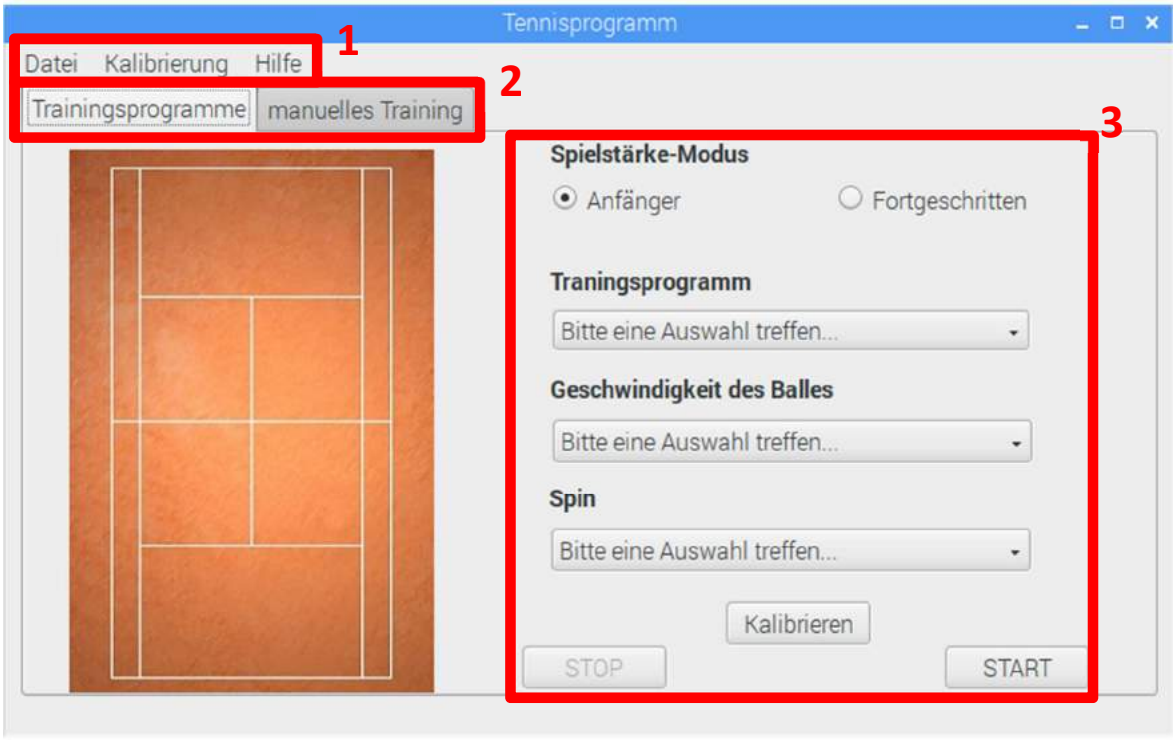


Abbildung 103: Hauptprogramm am Raspberry PI

1. Menüleiste
2. Auswahl des Unterprogramms
3. Eingabebereich der gewünschten Benutzereinstellungen

Unterprogramm *Trainingsprogramme*

In der Abbildung 103 ist weiters erkennbar, dass das Programm im Wesentlichen aus zwei verschiedenen Unterprogrammen besteht, wobei bei diesem Abbild das Unterprogramm *Trainingsprogramme* ausgewählt ist.

Bei diesem Unterprogramm können folgende Punkte vom Benutzer eingestellt werden:

- Auswahl der Spielerstärke des Benutzers (*Anfänger* oder *Fortgeschritten*)
  - Je nach Auswahl des Spielerstärke verändert sich das Intervall zwischen den abzuschießenden Tennisbällen.
  - Bei der Spielerstärke Anfänger beträgt die Zeit zwischen zwei Bällen ca. 5,5 Sekunden. Bei einem fortgeschrittenen Spieler sind es ca. 4,5 Sekunden.

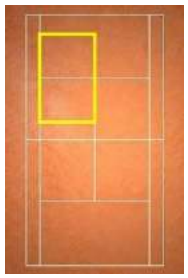
Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

- Auswahl des Trainingsprogramms
  - Bei diesem Unterprogramm kann aus sechs Trainingsarten gewählt werden. Je nach Auswahl ändert sich die Darstellung links neben den Benutzereinstellungen und zeigt in einem gelben Rahmen den Bereich an, in welchem die Bälle auf dem Platz aufspringen.
  - Grundlinie Vorhand Training
    - Bei dieser Einstellung werden die Bälle in Richtung der Vorhand des Benutzers geschossen.

  - Grundlinie Rückhand Training
    - Bei dieser Einstellung werden die Bälle in Richtung der Rückhand des Benutzers geschossen.

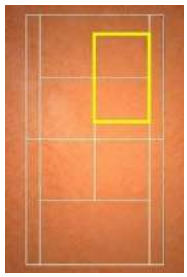
  - Grundlinien Training
    - Bei dieser Einstellung werden die Bälle in eine durch Zufall bestimmte Richtung geschossen.
    - Zusätzlich variiert auch die Flugweite bei jedem Schuss.
    - Die Maschine besitzt kurz vor dem Eintritt des Balles in die Schwungräder eine Lichtschranke. Bei Abschuss eines Balles wird diese Lichtschranke kurzzeitig unterbrochen und die Maschine beginnt sich neu auszurichten.

  - Volley Vorhand Training
    - Bei dieser Einstellung kann ein Vorhand Volley Training durchgeführt werden.

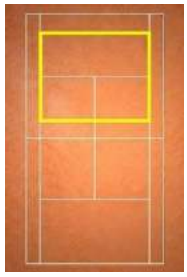


Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

- Volley Rückhand Training
  - Bei dieser Einstellung kann ein Rückhand Volley Training durchgeführt werden.



- Volley Training
  - Diese Einstellung ähnelt der Einstellung Grundlinien Training. Der einzige Unterschied besteht darin, dass die Flugweite der Bälle etwas geringer ist.



- Einstellung der Geschwindigkeit des Balles
  - Bei diesem Punkt muss der Benutzer die Geschwindigkeit des Balles wählen. Hierbei kann zwischen folgenden Einstellungen eine Auswahl getroffen werden:
    - schnell
    - mittel
    - langsam
    - Lob

- Spin
  - Zuletzt hat der Spieler die Möglichkeit, das Trainingsprogramm mit einem gewünschten Top-Spin der Bälle zu meistern. Dazu muss die Auswahlmöglichkeit Top-Spin gewählt werden.
  - Wenn kein Spin gewünscht ist, kann dies ebenfalls in dieser Auswahl eingestellt werden.

Falls eine Auswahl nicht getroffen wurde (z.B. kein Spin ausgewählt) und der Benutzer versucht die Maschine zu starten, wird dieser über nachfolgende Fehlermeldung (siehe Abbildung 104) darüber informiert und die Maschine startet nicht.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

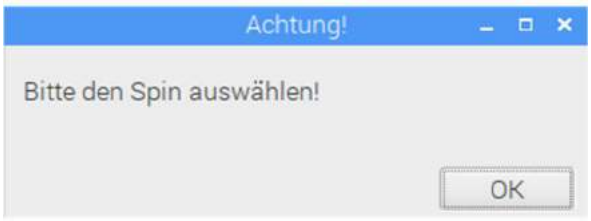


Abbildung 104: Fehlermeldung bei Nicht-Auswahl des Spins

Nachdem alle Benutzereinstellungen ausgewählt wurden, kann der START-Button betätigt werden.

Nach Betätigung des Buttons baut der Raspberry PI zuerst eine Verbindung zu dem Arduino auf. Über die nachfolgende Meldung wird der Benutzer darüber informiert (siehe Abbildung 105).

Abbildung 105: Meldung - Verbindungsaufbau zu den Motoren

Sobald eine Verbindung zum Arduino vorhanden ist, berechnet das Programm mithilfe des Schieß- und Eulerverfahrens die benötigte Anfangsgeschwindigkeit des Tennisballes. Auf Grundlage dieser Geschwindigkeit können anhand der gefundenen Funktionen in Abbildung 90 die benötigten PWM-Werte der beiden Schwungräder gefunden werden.

Des Weiteren wird die Höhe des Balles über dem Netz berechnet und überprüft, ob der Ball mit diesen gefundenen Einstellungen nicht im Netz landet.

Würde der Ball laut den Berechnungen im Netz landen, erhält der Benutzer eine Meldung wie in Abbildung 106 dargestellt.

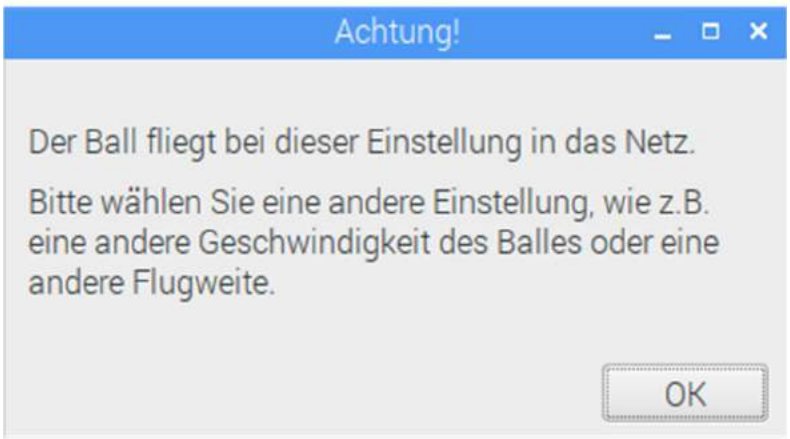


Abbildung 106: Meldung - Ball fliegt ins Netz

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Durch Betätigung des OK-Buttons, schließt sich das Fenster und der Benutzer kehrt zum vorherigen Bildschirm zurück um seine Auswahl anpassen zu können.

Falls der Ball – den Berechnungen nach – über das Netz fliegt, wird der Benutzer durch folgende Meldung (Abbildung 107) darauf aufgefordert, den vertikalen Winkel auf die vorgegebene Position zu stellen.

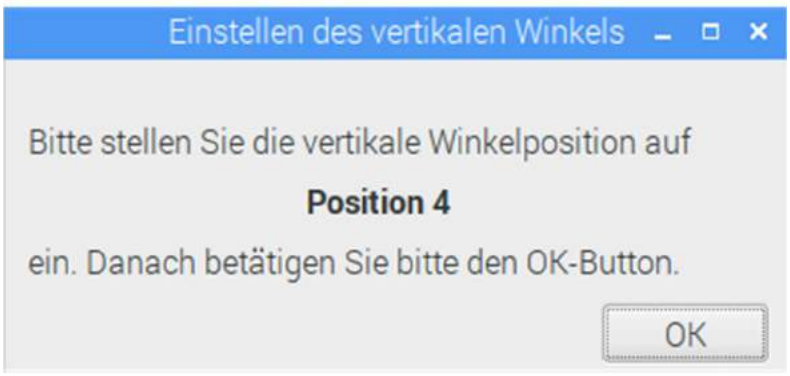


Abbildung 107: Meldung - Einstellen des vertikalen Winkels

Die Positionen sind dabei wie folgt bestimmt:

- Position 1: 15° vertikale Verstellbarkeit
- Position 2: 20° vertikale Verstellbarkeit
- Position 3: 25° vertikale Verstellbarkeit
- Position 4: 30° vertikale Verstellbarkeit

Nach erfolgreichem Einstellen des Winkels wird durch einen Klick auf den OK-Button das Fenster geschlossen.

Anschließend wird an den Arduino eine bestimmte Kette an Zahlen und Zeichen übermittelt, welche wie folgt aussehen könnte:

251#228%15$1*

Hierbei bildet der Arduino aus den ersten drei Ziffern eine Zahl und übergibt diese als PWM-Wert den oberen DC-Motor. Danach führt er dies mit den nächsten drei Zahlen durch und weist diese dem unteren DC-Motor zu. Die Ziffern zwischen % und $ bestimmen den horizontalen Winkel der Maschine, wobei es die Einstellungen 0°, 15° und 30° gibt. Zusätzlich gibt es noch die Einstellung 100, welche bedeutet, dass die Maschine die horizontale Einstellung nicht ändern soll. Die letzte Ziffer, welche sich zwischen den Zeichen $ und * befindet, gibt die Spielerstärke wieder. Dabei entspricht der Wert 1 der Spielerstärke *Anfänger* und 2 der Spielerstärke *Fortgeschritten*.

Da die Maschine eine gewisse Zeit (ca. 12 Sekunden) benötigt, bis sich die Schwungräder auf der gewünschten Drehzahl befinden und der Benutzer auch eine gewisse Zeit benötigt um auf die andere Seite des Tennisplatzes zu kommen, wird der

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

erste Ball nach ca. 20 Sekunden abgeschossen. Diese Wartezeit wird dem Benutzer über folgendes Warnfenster mitgeteilt (siehe Abbildung 108).

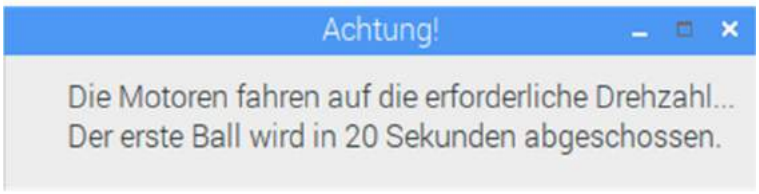


Abbildung 108: Meldung - Motoren fahren auf die gewünschte Drehzahl

Damit während des Betriebs die Maschine nicht verstellt werden kann, wird die gesamte Benutzeroberfläche – bis auf den STOP-Button – für den Benutzer gesperrt. Dies ist in der Abbildung 109 ersichtlich.

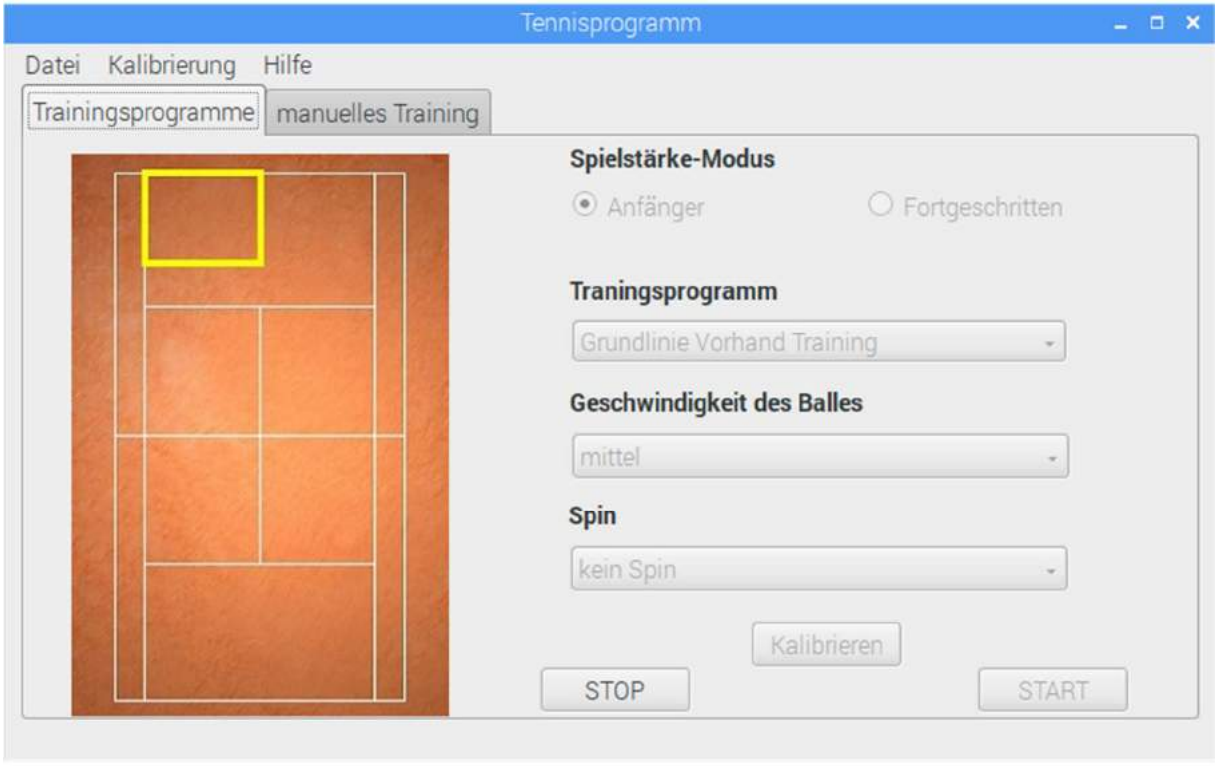


Abbildung 109: Unterprogramm *Trainingsprogramme* während des Betriebs

Bei Betätigen des STOP-Buttons wartet die Maschine noch drei Sekunden mit dem herunterfahren der beiden DC-Motoren und dem Getriebemotor, damit kein Ball in der Maschine bleibt. Der Schrittmotor fährt auf die Position 15°.

Danach steht dem Benutzer wieder die gesamte Benutzeroberfläche zur Verfügung.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Unterprogramm *manuelles Training*

Das zweite Unterprogramm *manuelles Training* ist in Abbildung 110 abgebildet.

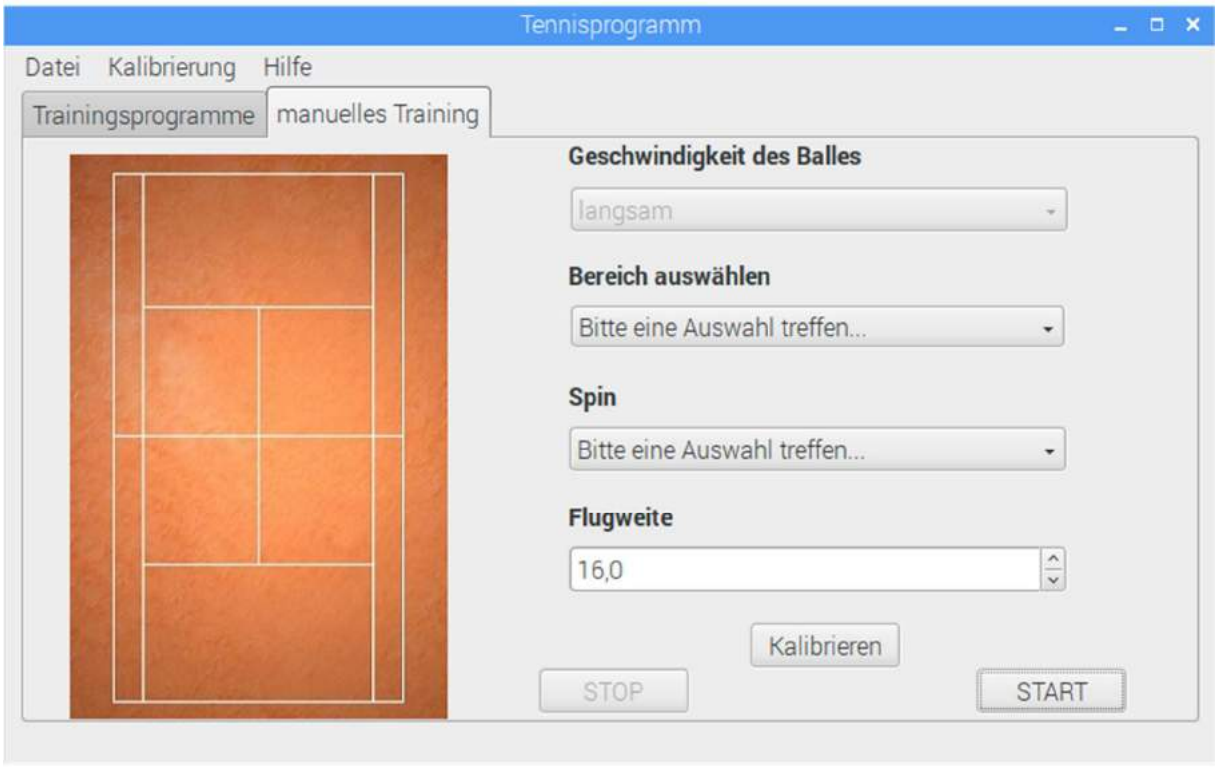
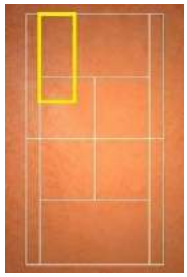


Abbildung 110: Unterprogramm *manuelles Training*

Wie in Abbildung 110 erkennbar ist, kann die Geschwindigkeit des Balles bei diesem Programm nicht verändert werden.

Danach muss der Benutzer folgende Einstellungen vornehmen:

- Auswahl des Bereichs
  - Bei diesem Unterprogramm kann aus drei verschiedenen Bereichen gewählt werden. Je nach Auswahl ändert sich die Darstellung links neben den Benutzereinstellungen und zeigt in einem gelben Rahmen den Bereich an, in welchem die Bälle auf dem Platz aufspringen.
  - Vorhand - Seite
    - Bei dieser Einstellung werden die Bälle in Richtung der Vorhand des Benutzers geschossen.



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

- Mitte
  - Bei dieser Einstellung werden die Bälle in die Mitte des Tennisplatzes geschossen.

- Rückhand - Seite
  - Bei dieser Einstellung werden die Bälle in Richtung der Rückhand des Benutzers geschossen.

- Spin
  - Auch in diesem Unterprogramm hat der Benutzer die Wahl zwischen Top-Spin und keinem Spin. Dies wird in dieser Auswahl eingestellt.

- Einstellung der Flugweite
  - Zuletzt muss der Benutzer nur noch die Flugweite des Balles (im Bereich zwischen 13 m bis 23 m) einstellen.

Auch in diesem Unterprogramm müssen alle Felder durch den Benutzer ausgewählt sein, um die Maschine starten zu können.

Nachdem alle Benutzereinstellungen ausgewählt wurden, kann der START-Button betätigt werden.

Die Maschine baut zunächst wieder die Verbindung zu dem Arduino auf. Wenn die Verbindung aufgebaut ist, errechnet das Programm erneut die benötigte Anfangsgeschwindigkeit des Balles und in weiterer Folge die PWM-Werte der einzelnen DC-Motoren.

Des Weiteren wird auch die Höhe des Balles über dem Netz berechnet und nach der gleichen Vorgehensweise – wie bei dem Unterprogramm *Trainingsprogramme* – überprüft. Nach dieser Überprüfung wird der Benutzer aufgefordert, die Maschine auf die Position 3 zu stellen.

Die berechneten PWM-Werte werden nun wieder in der gleichen Form – wie beim anderem Unterprogramm– an den Arduino übergeben und die Motoren starten.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Nach ca. 20 Sekunden nach dem Start der Maschine wird der erste Ball aus der Maschine geschossen.

Damit während des Betriebs die Maschine nicht verstellt werden kann, wird die gesamte Benutzeroberfläche auch bei dem Unterprogramm – bis auf den STOP-Button – für den Benutzer gesperrt (siehe Abbildung 111).

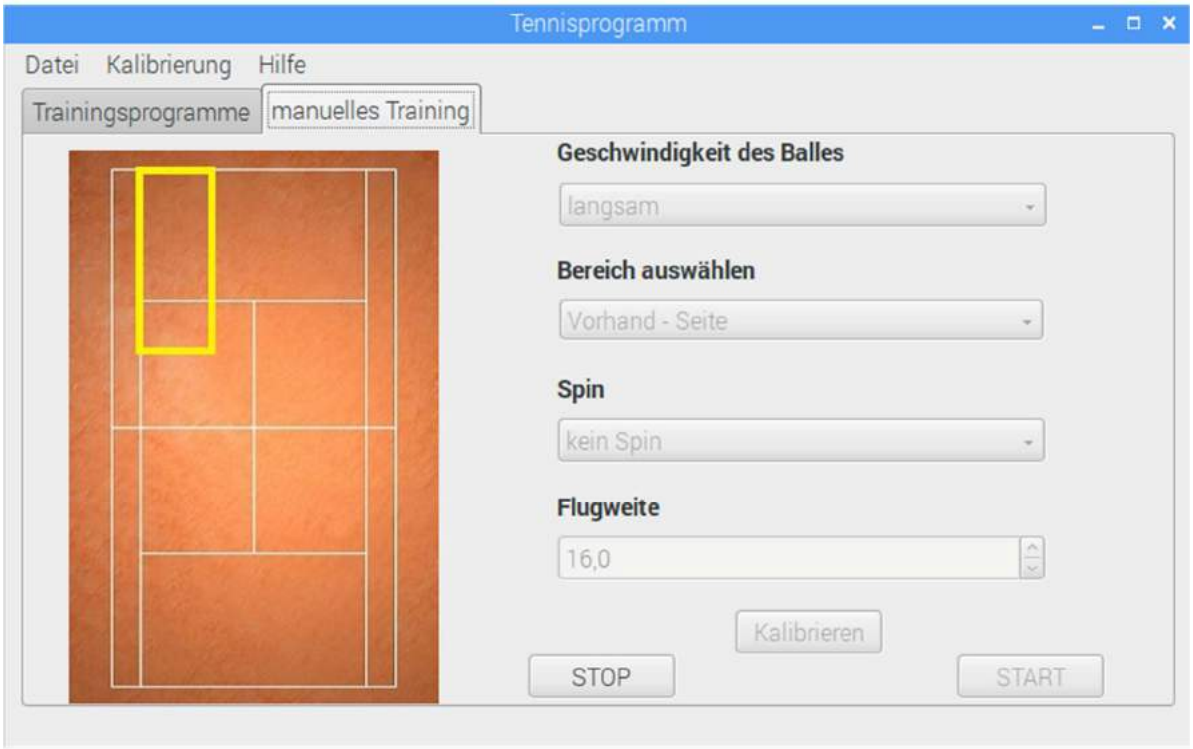


Abbildung 111: Unterprogramm *manuelles Training* während des Betriebs

Bei Betätigen des STOP-Buttons läuft die Maschine noch drei Sekunden weiter und erst danach fahren die beiden DC-Motoren und der Getriebemotor herunter. Der Schrittmotor fährt auf die Position 15°.

Danach steht dem Benutzer wieder die gesamte Benutzeroberfläche zur Verfügung.

Dialog-Fenster Kalibrieren

Da in dem Kapitel Messungen festgestellt wurde, dass die Flugweite der Tennisbälle auch sehr stark von der Marke und der Abnützung des Balles abhängt, ist es notwendig die Maschine auf die verwendeten Tennisbälle einzustellen.

Beim Betätigten des Kalibrieren-Buttons bei den jeweiligen Unterprogrammen oder durch Auswahl des Menüleisteneintrags *Kalibrierung* → *Kalibrieren* (siehe Abbildung 112) öffnet sich ein Dialog-Fenster, welches in Abbildung 113 dargestellt ist.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Abbildung 112: Menüleisteneintrag Kalibrieren

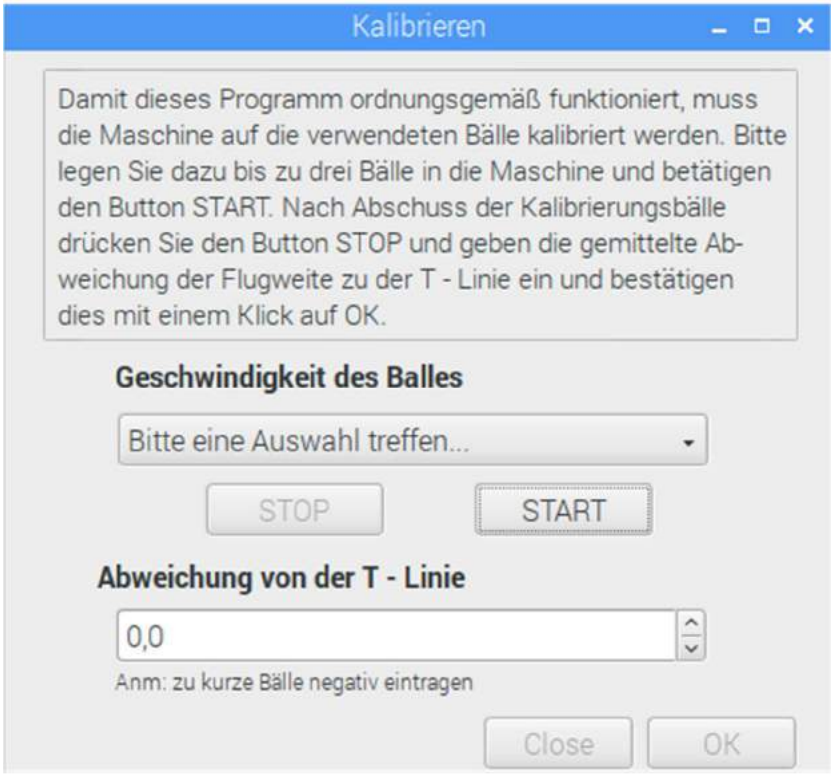


Abbildung 113: Dialog-Fenster Kalibrieren

In diesem Dialog-Fenster muss der Benutzer zuerst die Geschwindigkeit des Balles auswählen. Dabei stehen ihm wieder die schon vorhin erwähnten vier Auswahlmöglichkeiten (schnell, mittel, langsam, Lob) zur Verfügung. Danach kann der START-Button betätigt werden.

Die Raspberry PI baut nun eine Verbindung zum Arduino auf. Danach berechnet das Programm die benötigte Anfangsgeschwindigkeit des Balls, damit die Flugweite genau 17,3 m beträgt. Dies entspricht genau der Weite der T-Linie. Des Weiteren wird wieder die Höhe über dem Netz berechnet und überprüft. Anschließend an die Überprüfung wird der Benutzer wiederum aufgefordert, die Maschine auf die errechnete Position zu stellen. Nach dem die Kalibrierungsbälle abgeschossen wurden, kann die Maschine durch Betätigten des STOP-Buttons wieder abgeschaltet werden.

Nun kann der Benutzer die Abweichung der tatsächlichen Flugweite zu der Weite der T-Linie in das dafür vorgesehene Feld eintragen. Dabei ist zu beachten, dass zu kurze Bälle negative einzutragen sind.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Nach Betätigen des OK-Buttons berechnet nun das Programm die benötigte Anfangsgeschwindigkeit, die die benutzte Ball gebraucht hätten, um im Mittel eine Flugweite von 17,3 m zu erreichen.

Durch Vergleich der beiden Anfangsgeschwindigkeiten ergibt sich ein Korrekturfaktor. Dieser Faktor wird bei beiden Unterprogrammen zur Berechnung der benötigten Anfangsgeschwindigkeiten zugezogen.

Dialog-Fenster Händigkeit

Die Händigkeit spielt im Tennis eine wesentliche Rolle. Bei beiden Unterprogrammen ist es notwendig zu wissen, ob der Benutzer ein Rechts- oder Linkshänder ist. Die Grundeinstellung ist dabei für Rechtshänder voreingestellt.

Damit die Händigkeit umgestellt werden kann, muss der Menüleisteneintrag *Datei → Händigkeit* (siehe Abbildung 114) aufgerufen werden.

Abbildung 114: Menüleisteneintrag Händigkeit

Nach Betätigten des Menüleisteneintrags Händigkeit öffnet sich folgendes Dialog-Fenster (siehe Abbildung 115).

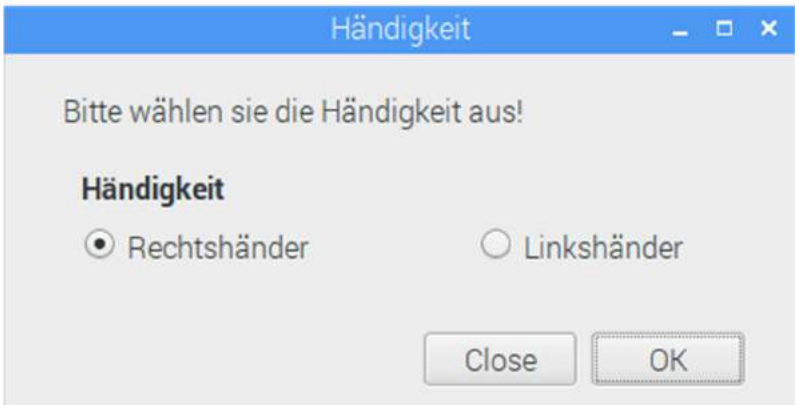


Abbildung 115: Dialog-Fenster Händigkeit

Nun kann die Händigkeit umgestellt und mit dem OK-Button bestätigt werden.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Dialog-Fenster Laser einstellen

Bei den Trainingsprogrammen *Grundlinien Training* und *Volley Training* im Unterprogramm *Trainingsprogramme* wird eine Lichtschranke benötigt.

Damit die Programme einwandfrei funktionieren, muss der Laser die Fotodiode exakt treffen. Da es bei Transporten passieren kann, dass der Laser bzw. die Fotodiode sich verschoben haben, kann die Fotodiode über zwei Schrauben neu ausgerichtet werden.

Um ein Feedback von dem Programm zu bekommen, ob der Laser exakt ausgerichtet ist, kann über den Menüleisteneintrag *Datei → Laser einstellen* (siehe Abbildung 116) das Dialog-Fenster *Laser einstellen* geöffnet werden (siehe Abbildung 117)

Abbildung 116: Menüleisteneintrag *Laser einstellen*

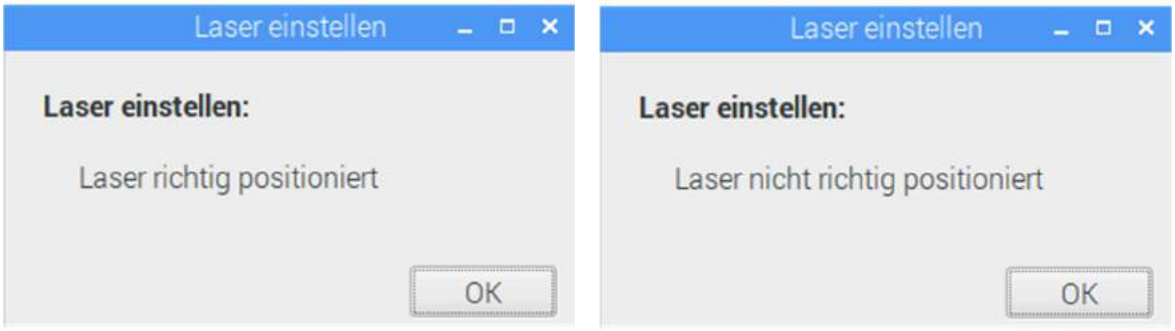


Abbildung 117: Dialog-Fenster *Laser einstellen*

In diesem Dialog-Fenster sieht der Benutzer nun, ob der Laser richtig positioniert ist. In der linken Darstellung auf der Abbildung 117 ist der Laser richtig und auf der rechten Darstellung nicht richtig positioniert.

Durch Betätigung des OK-Buttons schließt das Dialog-Fenster wieder.

Programmcode

Der vollständige Programmcode des Raspberry PIs und des Arduinos befindet sich im Anhang F.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

5.3 Kostenaufstellung

Die Kosten für die Herstellung des Funktionsmusters sind in Tabelle 18 aufgelistet. Bei diesen Kosten sind die investierten Arbeitsstunden nicht enthalten.

Bezeichung	Einzelpreis	Anzahl	Gesamtpreis
mechanische Bauteile			**€ 450,47**
Fertigung Laserteile	€ 80,00	1	€ 80,00
Holzplatte, Holz für Box	€ 33,00	1	€ 33,00
Wellen verchromen	€ 90,00	1	€ 90,00
Lager	€ 4,66	4	€ 18,64
kleines Riemenrad	€ 6,46	1	€ 6,46
großes Riemenrad	€ 19,76	1	€ 19,76
Riemen	€ 12,35	1	€ 12,35
PVC-Rohr 70	€ 4,45	1	€ 4,45
PVC-Rohr 100	€ 4,59	1	€ 4,59
Kupplung	€ 11,24	3	€ 33,72
Gewindestangen	€ 2,50	3	€ 7,50
Räder gummieren	€ 50,00	2	€ 100,00
sonstige Teile (Schrauben, Scheiben, Mutter, usw.)	€ 40,00	1	€ 40,00
elektronische Bauteile			**€ 462,83**
DC-Motoren	€ 38,20	2	€ 76,40
Getriebemotor	€ 39,67	1	€ 39,67
Schrittmotor	€ 79,55	1	€ 79,55
Raspberry PI + Gehäuse + Netzteil + SD-Karte	€ 63,37	1	€ 63,37
Arduino	€ 17,90	1	€ 17,90
Motortreiber	€ 20,10	3	€ 60,30
Schrittmotortreiber	€ 12,99	1	€ 12,99
Netzteil	€ 33,78	1	€ 33,78
Hall-Sensor	€ 2,15	1	€ 2,15
Laser	€ 0,64	3	€ 1,92
Fotodioden	€ 1,09	3	€ 3,27
Touchscreen mit Halterung	€ 71,53	1	€ 71,53
Sonstige Aufwändungen			**€ 127,74**
Verteiler	€ 6,99	1	€ 6,99
Schraniere	€ 9,99	2	€ 19,98
Stromschalter	€ 15,48	1	€ 15,48
Griff	€ 7,29	1	€ 7,29
Lüftungsgitter	€ 10,99	2	€ 21,98
Stromkabel	€ 7,99	1	€ 7,99
Füße	€ 1,59	4	€ 6,36
Leitungsdose	€ 4,59	1	€ 4,59
Kabelkanal	€ 7,08	1	€ 7,08
sonstige Teile (Kabeln, Schläuche, usw.)	€ 30,00	1	€ 30,00
GESAMT Funktionsmuster			**€ 1.041,04**

Tabelle 18: Kostenaufstellung des Funktionsmusters

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

6 Erkenntnisse

Im Folgenden sollen die bisher gewonnenen Erkenntnisse zusammengefasst und ein Ausblick auf Verbesserungen für ein mögliches Serienprodukt geboten werden.

6.1 Erkenntnisse aus mechanischer Sicht

Aus den Kundenbefragungen geht hervor, dass Bedienungsfreundlichkeit und Mobilität wichtige Leistungsanforderungen an das Produkt sind. Verbesserungen in Ersterem konnten mit dem Funktionsmuster ausgiebig erprobt und gezeigt werden. Darauf aufbauend soll daher ein Entwurf gestaltet werden, der den Fokus auf Mobilität und Ergonomie ebenso berücksichtigt.

Folgende Verbesserungen werden dabei angestrebt:

Mobilität

Der erweiterte Entwurf sieht vor, dass die Maschine einhändig getragen werden kann. Dafür wird der Ballbehälter abnehmbar gestaltet und mit einem Griff versehen, sodass er zusammen mit den Bällen in der freien Hand transportiert werden kann. So ist das Gewicht auf ergonomische Weise über den Körper aufgeteilt und der Bedarf nach einer getrennten Aufbewahrung zum Transport der Bälle entfällt.

Dabei soll der Ballbehälter so gestaltet werden, dass ein Verstopfen des Ballzulaufs, wie es bei vielen erhältlichen Modellen auftreten kann, nicht möglich ist. Dazu wird eine spiralförmige Lagerung vorgesehen:

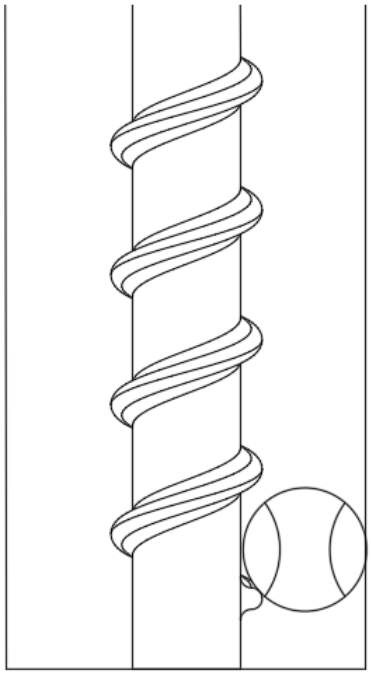


Abbildung 118: Konzept eines Ballbehälters mit spiralförmiger Lagerung

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Der Behälter kann bei einer Länge von 70cm je nach Ausführung ca. 40-50 Bälle aufnehmen, was die Vorgaben aus der Anforderungsliste erfüllt. Am Ende ist einseitig eine Verjüngung angebracht, welche vom Benutzer durch das auf den Boden pressen des Behälters zum stehenden, ergonomischen Aufsammeln von Bällen genutzt werden kann:

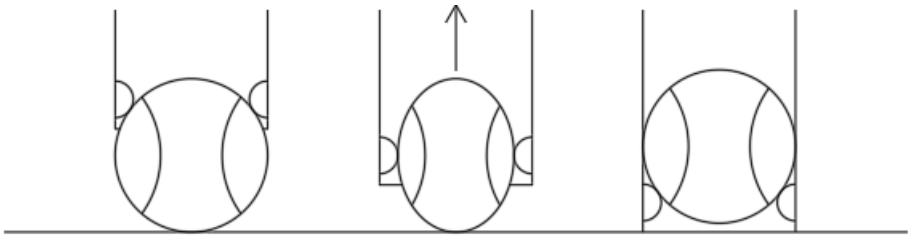


Abbildung 119: Aufsammeln von Bällen mittels einer Verjüngung

Indem der Behälter um 180° gedreht wieder auf die Maschine aufgesetzt wird, können die Bälle ungehindert aus dem Zylinder rollen. Dort können sie wahlweise mittels einer durch einen Motor angetriebenen Schranke oder eines Ballförderrads, wie es bereits im Funktionsmuster eingesetzt wurde, in definierten Intervallen an den Abschussmechanismus abgegeben werden. Abbildung 120 zeigt den fertigen Entwurf des Ballbehälters. Die Verjüngung zum Aufsammeln der Bälle ist unterhalb angebracht.

Abbildung 120: Entwurf des Ballbehälters

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Fertigung und Montage

Durch den Einsatz von Blechbiegeteilen können Fertigungskosten und Teileanzahl reduziert werden. Mithilfe geeigneter Motoren kann außerdem auf Schwungrad-Wellen, Kupplungen und Lager verzichtet werden, da die Schwungräder direkt auf den Motorwellen befestigt werden können. Dies reduziert die Anzahl der benötigten Teile und Montageschritte weiter. Ebenso muss die Welle nicht mehr in den Lagern justiert werden. Die Verbesserungen sind in Abbildung 121 zusammen mit einem Linearantrieb dargestellt:

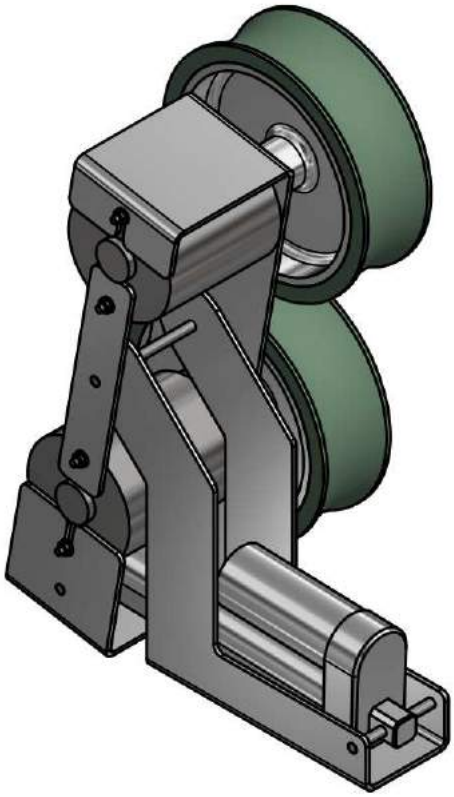


Abbildung 121: Blechbiegeteile als Motorhalterung

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

6.2 Erkenntnisse aus elektronischer Sicht

Aus elektronischer Sicht ergaben sich bei den Testschüssen und Messungen im Wesentlichen drei Erkenntnisse bzw. Verbesserungen, welche bei dem Gesamtentwurf zu berücksichtigen sind. Diese werden in diesem Kapitel kurz erläutert.

PID-Regler

In Abbildung 84 ist ersichtlich, dass die beiden DC-Motoren – wie bereits erwähnt – ungefähr fünf Sekunden benötigen, um wieder die gewünschte Drehzahl zu erreichen. Diese Zeit kann durch Anwendung eines Reglers deutlich verkürzt werden.

Bei dem Programm des Funktionsmusters gibt es die beiden Spielstärken *Anfänger*, bei der die Tennisbälle in einem Abstand von 5,5 Sekunden und *Fortgeschritten*, bei dem der Abstand 4,5 Sekunden beträgt. Mit diesen beiden Modi wird die Mehrheit der Tennisspieler abgedeckt. Damit die letzte Spielstärke – der Profi-Modus – auch noch realisiert werden kann, muss ein Regler eingesetzt werden.

Zur Auswahl stehen ein P-Regler, I-Regler, D-Regler, PD-Regler, PI-Regler und ein PID-Regler. In Abbildung 122 sind diese Regler mit deren Genauigkeit und Schnelligkeit abgebildet. Dabei ist ersichtlich, dass der PID-Regler der schnellste und am genaueste ist.

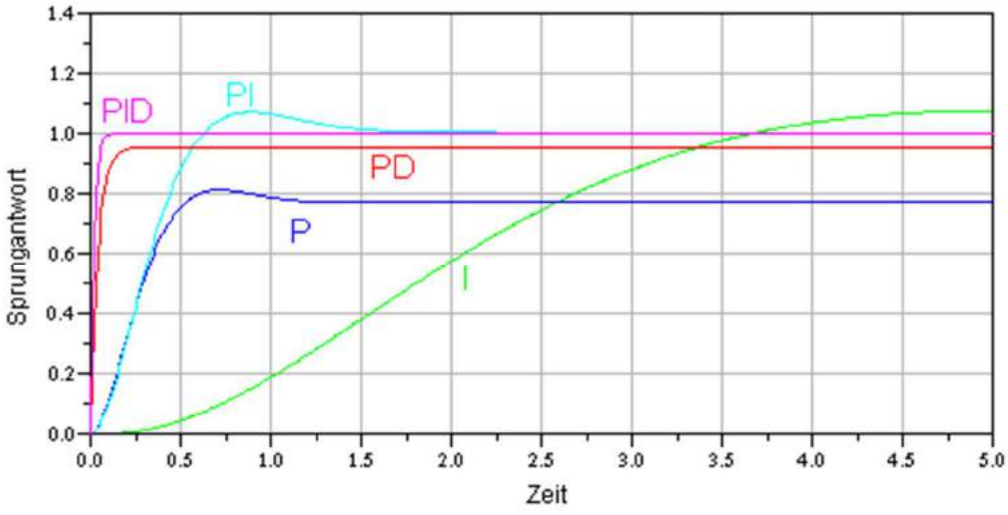


Abbildung 122: Vergleich der Reglertypen in einem Regelkreis (RN-Wissen.de, 2019)

Ein PID-Regler (proportional-integral-controller) besteht aus den Anteilen des P-Glieds, des I-Gliedes und des D-Glieds und entweder als Parallelstruktur oder Reihenstruktur ausgeführt werden (siehe Abbildung 123).

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

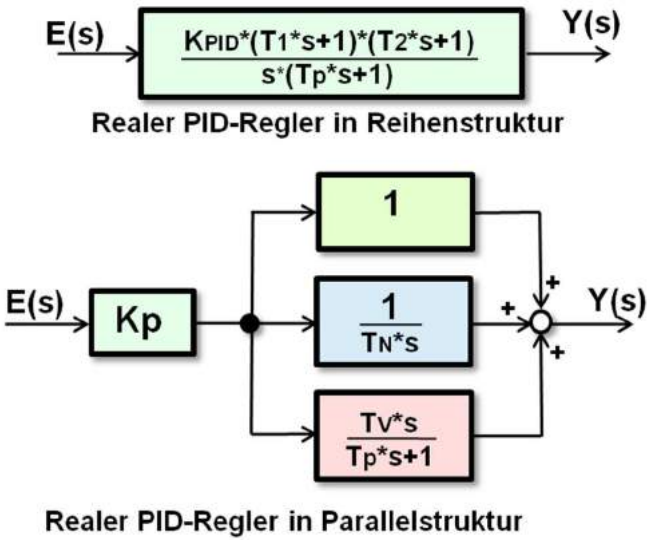


Abbildung 123: PID-Regler in Reihenstruktur (oben) und Parallelstruktur (unten) (Wikipedia, 2019a)

Der P-Regler, welcher proportionalwirkend ist, multipliziert die Regelabweichung mit seinem Verstärkungsfaktor K_p. Die Reglergleichung lautet

$$y(t) = K_p \cdot e(t) \tag{6.1}$$

Der I-Regler, welcher integralwirkend ist, summiert die Regelabweichung über der Zeit auf und multipliziert dieses Integral mit dem Faktor K_i. Die Reglergleichung dazu ist

$$y(t) = K_i \cdot \int_0^t e(\tau)\, d\tau \tag{6.2}$$

Der D-Regler, welcher nur in Verbindung mit einem P-Regler (PD-Regler) oder einem I-Regler (PI-Regler) eingesetzt werden kann, ist eine Differenzierer. Die Regelgleichung ist

$$y(t) = K_d \cdot \frac{de(t)}{dt} \tag{6.3}$$

Daraus ergibt sich für die Reglergleichung für den PID-Regler folgende Gleichung

$$y(t) = K_p \cdot e(t) + K_i \cdot \int_0^t e(\tau)\, d\tau + K_d \cdot \frac{de(t)}{dt} \tag{6.4}$$

Damit ein solcher Regler in dem Programm integriert werden kann, muss die Drehzahl der Schwungräder zu jedem Zeitpunkt bekannt sein. Diese Drehzahl kann zum Beispiel mit einem Hall-Sensor, wie er im Kapitel 5.1.7 eingesetzt wurde, ermittelt werden.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Linear Antrieb

Bei den Messungen bzw. Testschüssen des Funktionsmusters wurde schnell erkannt, dass die mechanische vertikale Winkelverstellbarkeit nicht die ideale Lösung ist. Einerseits beansprucht diese Zeit, welche der Benutzer nicht aufwenden mag und andererseits wird ein Werkzeug benötigt, um die Maschine verstellen zu können.

Eine alternative Lösung wäre ein Linearmotor, welcher die Maschine automatisch in die gewünschte Position stellt.

Ein Linearmotor wandelt eine Drehbewegung in eine lineare Bewegung um. Durch die Drehung des elektrischen Gleichstrommotors wird ein Getriebe angetrieben, welches schlussendlich über die Abtriebswelle mit der Führungsspindel und somit mit der Führungsspindelmutter, welche auf der Führungsspindel sitzt, verbunden ist. Durch die Drehung der Führungsspindel, verschiebt sich die Führungsspindelmutter und es wird dadurch eine lineare Bewegung hervorgerufen.

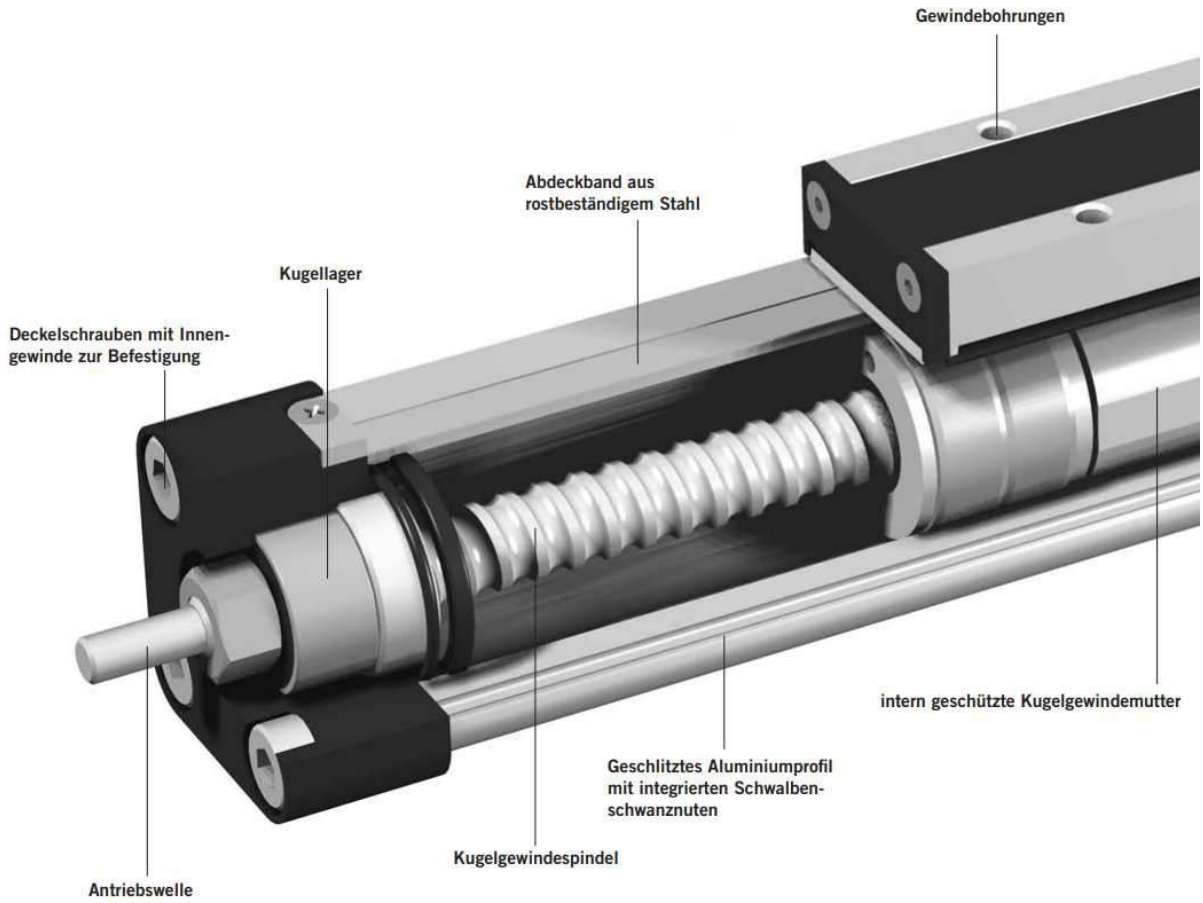


Abbildung 124: Linearantrieb mit Kugelgewindespindel (Bobry, 2019)

Ein kleiner Nachteil dieses Motors ist es, dass sich diese nur sehr langsam bewegen und demnach eine gewisse Zeit brauchen, bis sie sich in der gewünschten Position befinden. Die Winkelumstellung könnte allerdings passieren, währenddem die DC-

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Motoren auf die eingestellte Drehzahl beschleunigen. Somit würde die Winkelumstellung keine Mehrzeit in Anspruch nehmen.

Bei dieser Art des Linearantriebes befindet sich der Antrieb auf der Bodenplatte der Maschine. Der bewegliche Schlitten des Linearantriebs, auf dem die Stange für die vertikale Winkelstellung befestigt ist, kann somit verschiedene Positionen einnehmen und damit den vertikalen Winkel ändern.

Der Linearantrieb kann allerdings auch noch in einer anderen Ausführung verwendet werden. In diesem Fall wird anstelle des Schlittens eine Stange bewegt.

Abbildung 125: Linearantrieb mit ausfahrbarer Stange (Bobry, 2019)

Bei dieser Ausführung kann der Linearantrieb wiederum auf der Bodenplatte montiert werden. Die ausfahrbare Stange ist zwischen den Seitenwänden der Maschine befestigt. Durch Antreiben des Motors bewegt sich die Stange und womit die Winkelposition der Maschine verändert werden kann (siehe Abbildung 126).

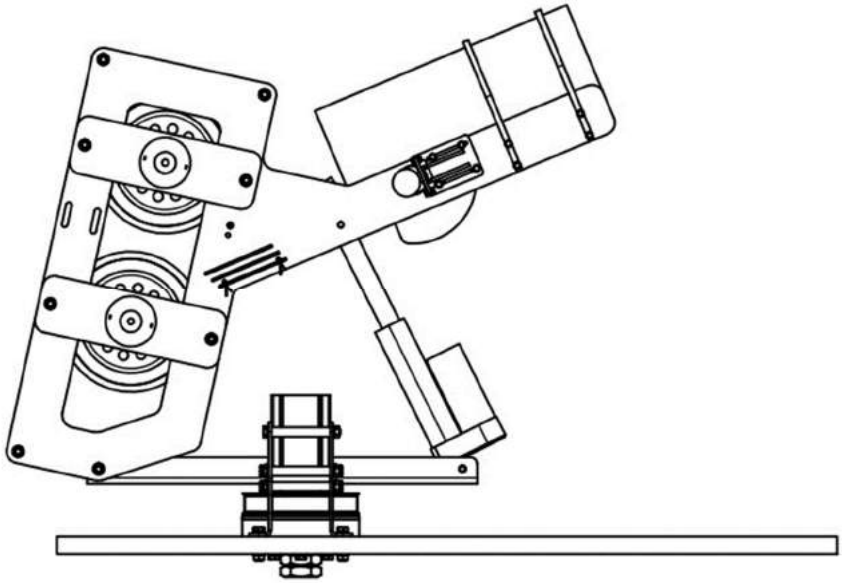


Abbildung 126: Linearantrieb im Modell

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Änderung der Schwungräder

Um den Profi-Modus ideal umsetzen zu können müssen schnellere bzw. schärfere Bälle gespielt werden können. Um dies zu verwirklichen, müssten die Schwungräder angepasst werden.

Wie unter dem Punkt PID-Regler beschrieben wird, sollte die Zeit zwischen den Abschüssen verkürzt werden. Bei einer Erhöhung des Massenträgheitsmomentes, etwa durch einer zusätzlichen Schwungmasse an den Schwungrädern, kann der Drehzahlverlust bei einem Ballabschuss verringert werden.

Dem gegenüber steht eine längere Beschleunigungszeit der Räder auf die gewünschte Drehzahl, welche jedoch mit einem PID-Regler verkürzt werden könnte.

Bei der Vergrößerung des Radius der Schwungräder wird die Winkelgeschwindigkeit der Schwungräder, welche benötigt wird, um dem Ball eine bestimmte Anfangsgeschwindigkeit zu erteilen, verringert. Dadurch können die Bälle schneller bzw. schärfer abgeschossen werden.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

7 Literaturverzeichnis

Alam, Firoz / Subic, Aleksandar / Naser, Jamal / Khan, M. Masud Kamal (2008): A Study of Spin Effects on Tennis Ball Aerodynamics, in: *WSEAS Transactions on Fluid Mechanics 3*

Bobry (2019) *Linearantrieb mit Kugelgewindespindel* [online] www.hoerbiger.com [20.10.2019]

Cross, Rod / Lindsey, Crawford (2005): *Technical Tennis. Racquets, Strings, Balls, Courts, Spin and Bounce,* Racquet Tech Publishing

Cross, Rod / Lindsey, Crawford (2013): Measurements of drag and lift on tennis balls in flight, in: *Sports Engineering*, Ausgabe 17, S. 89-96.

Dembowski, K. (2015): *Raspberry Pi - Das technische Handbuch*, Springer Fachmedien Wiesbaden

Dignall, Richard John (2004): *Modelling the impact of tennis balls on court sufaces*, Dissertation, University of Sheffield

Dreyer, Hans-Joachim / Eller, Conrad / Holzmann, Günther / Meyer, Heinz / Schumpich, Georg (2012): *Technische Mechanik. Kinematik und Kinetik*, Springer Vieweg

Dullinger, F. (2019): *Hall-Effekt-Sensoren am Arduino* [online] http://bollwerk-essen.bplaced.net/?hall_sensor [08.06.2019]

Ehrlenspiel, Klaus / Kiewert, Alfons / Lindemann, Udo / Mörtl, Markus (2014): *Kostengünstig Entwickeln und Konstruieren. Kostenmanagement bei der integrierten Produktentwicklung*, Springer-Verlag Berlin Heidelberg

Huckle, Thomas / Schneider, Stefan (2006): *Numerische Methoden*, Springer-Verlag Berlin Heidelberg

ITF (2017): *ITF Approved Tennis Balls, Classified Surfaces & Recognised Courts 2017. A Guide to Products and Test Methods*

ITF (2019): *ITF Rules of Tennis*

Kompendium (2019): *Servos - InfoTip Kompendium* [online] https://kompendium.infotip.de/servos.html [08.06.2019]

Lernhelfer (2019): *Gleichstrommotor in Physik | Schülerlexikon | Lernhelfer* [online] https://www.lernhelfer.de/schuelerlexikon/physik/artikel/gleichstrommotor [23.07.2019]

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Lobster Sports (2019): *Products* [online] https://www.lobstersports.com/products/el03.htm [25.10.2019]

Mark, P. (2019): *Eine Einführung in Schrittmotoren* [online] https://www.channel-e.de/designcorner/artikel/article/eine-einfuehrung-in-schrittmotoren.html [01.09.2019]

MathWorks (2019): ode45 [online] https://www.mathworks.com/help/matlab/ref/ode45.html [21.08.2019]

Naefe, Paul (2018): *Methodisches Konstruieren. Auf den Punkt gebracht*, Springer Vieweg

Pahl, Gerhard / Beitz, Wolfgang / Feldhusen, Jörg / Grote, Karl-Heinrich (2007): *Konstruktionslehre*, Springer-Verlag Berlin Heidelberg

Ponn, Josef / Lindemann, Udo (2008): *Konzeptentwicklung und Gestaltung technischer Produkte. Optimierte Produkte – systematisch von Anforderungen zu Konzepten*, Springer-Verlag Berlin Heidelberg

Probst, U. (2011): *Servoantriebe in der Automatisierungstechnik,* Wiesbaden: Vieweg+Teubner

Restian (2019): *Drehzahlmessung* [online] https://arduino-projekte.webnode.at/meine-projekte/drehzahlregelung-mit-arduino-uno-und-attiny45/drehzahlmessung/ [08.06.2019]

RN-Wissen.de (2019): *Regelungstechnik* [online] https://rn-wissen.de/wiki/index.php/Regelungstechnik#P-Regler [20.10.2019]

Saatweber, Jutta (2011): *Kundenorientierung durch Quality Function Deployment. Produkte und Dienstleistungen mit QFD systematisch entwickeln*, Symposion Publishing GmbH

Schaad, S. (2019): *Schrittmotor kurz erklärt* [online] https://wiki.ntb.ch/infoportal/_media/hardware/sysp/bauteile/schrittmotor_kurz_erklaert_d.pdf [31 .08.2019]

Schloske, Alexander (2017): Foliensammlung zur Vorlesung „Qualitätsmanagement in der Produktentwicklung“ an der TU Wien: *Kunden- und Wettbewerbsorientierte Produktentwicklung mit QFD*, digitale Fassung

Wikipedia (2019a): *Regler* [online] https://de.wikipedia.org/wiki/Regler#PID-Regler [20.10.2019]

Wikipedia (2019b): *Schrittmotor* [online] https://de.wikipedia.org/wiki/Schrittmotor [31.08.2019]

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Wittel, Herbert / Muhs, Dieter / Jannasch, Dieter / Voßiek, Joachim (2015): *Roloff/Matek Maschinenelemente,* Springer Vieweg

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

8 Abbildungsverzeichnis

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

9 Tabellenverzeichnis

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Anhang A (Programmcode Versuchsaufbauten)

Geschwindigkeitsmessung-Aufbau

Code (Arduino)

```
#include <Wire.h>
#include <LCD.h> //Package für den LCD-Bildschirm
#include <LiquidCrystal_I2C.h> //Package für das I²C-Modul

LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7); //Konfig des I²C
int abstand = 88; //Abstand der Lichtschranken in mm
float count1 = 0; //Anzahl der Messungen
float count2 = 0; //Anzahl der Durchgänge bei Lichtschranken 2
unsigned long time1 = 0; //Zeit bei Unterbrechung des Lichtschr. 1
unsigned long time2 = 0; //Zeit bei Unterbrechung des Lichtschr. 2
unsigned long T = 0; //Zeitdifferenz
float V = 0; //aktuelle Geschwindigkeit
float V1 = 0; //maximale Geschwindigkeit

void setup() {
  lcd.begin (20,4); //Zuweisung der Größe des LCD-Displays
  lcd.setBacklightPin(3,POSITIVE);
  lcd.setBacklight(HIGH);
  lcd.home();
  lcd.setCursor(0,0);
  lcd.print("Maximal:");
  lcd.setCursor(16,0);
  lcd.print("km/h");
  lcd.setCursor(0,2);
  lcd.print("Anzahl:");
  lcd.setCursor(0,3);
  lcd.print("Zeit:");
  lcd.setCursor(16,3);
  lcd.print("us");
  lcd.setCursor(16,1);
  lcd.print("km/h");
  lcd.setCursor(0,1);
  lcd.print("Aktuell:");

  attachInterrupt(0, Start, FALLING); //Messung Starten
  attachInterrupt(1, Ende, FALLING); //Messung Beenden
}

void loop() {
  if ((count2==count1))T=time2-time1; //prüft, ob Messung real ist
  V=3600.0*abstand/T; //Geschwindigkeit = Weg / Zeit
  if (V>10000) V=0; //unrealistische Messung?
  V1=max(V,V1); //maximale Geschwindigkeit finden
  lcd.setCursor(8,0);
  Leerstellen(V1); //Leerstellen einfügen
  lcd.print(V1,2); //Schreibt die maximale Geschwindigkeit am LCD
  lcd.setCursor(8,1);
  Leerstellen(V); //Leerstellen einfügen
  lcd.print(V,2); //Schreibt die aktuelle Geschwindigkeit am LCD
  lcd.setCursor(8,2);
  Leerstellen(count1); //Leerstellen einfügen
  lcd.print(count1); //Anzahl der Messungen
  lcd.setCursor(8,3);
```

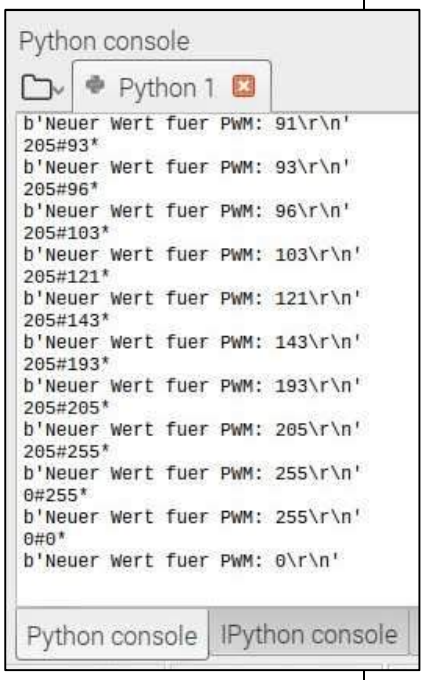
Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

```
  if (T > 9999999) T=0;
  if (T < 1000000) lcd.print(" ");
  if (T < 100000) lcd.print(" ");
  if (T < 10000) lcd.print(" ");
  if (T < 1000) lcd.print(" ");
  if (T < 100) lcd.print(" ");
  if (T < 10)  lcd.print(" ");
  lcd.print(T); //Schreibt die Zeitdifferenz am LCD
  delay(2000); //Abbruch der Messung nach 2 Sekunden
  count2=count1;
     }

void Start() { //Starten der Messung
  time1=micros();
  count1++;
 }

void Ende() { //Ende der Messung
  time2=micros();
  count2++;
}

void Leerstellen(float wert) { //Gewünschte Leerzeichen einfügen
  if (wert < 1000) lcd.print(" ");
  if (wert < 100) lcd.print(" ");
  if (wert < 10)  lcd.print(" ");
}
```

Gleichstrommotor-Aufbau

Code (Arduino)

```
#include <Wire.h>
#include <Keypad.h>

#define dir_1 12
#define pwm_1 10
#define dir_2 13
#define pwm_2 11

#define Buffer_Length 8

char Data[Buffer_Length]; //empfangene Daten
byte data_count = 0;  //Zeichenzähler

void setup(){
Serial.begin(9600);
pinMode(pwm_1,OUTPUT); //Zuweisung der PINS
pinMode(dir_1,OUTPUT);
pinMode(pwm_2,OUTPUT);
pinMode(dir_2,OUTPUT);
}

void loop(){

if (Serial.available())
  serialEvent();  //Aufruf der Funktion

}
```



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

```
void serialEvent() { //serielle Daten werden empfangen
  byte ch = Serial.read();  //Zeichen einlesen
  Data[data_count] = ch;  //auf String abspeichern
  data_count++;
  if (ch == 35) {  //Prüfen ob das Zeichen ein # ist
    motor1_einstell(); //Funktion aufrufen
    data_count = 0;  //Zähler auf 0 stellen
  }
  if (ch == 42) { //Prüfen ob das Zeichen ein * ist
    motor2_einstell(); //Funtkion aufrufen
  }
}

void motor1_einstell() //Drehzahl für Motor 1 einstellen
{
    Data[data_count] = '\0'; //String ende
    uint16_t zahl = atoi(Data); //String in Int umwandeln

    digitalWrite(dir_1,LOW); //Richtung für den Motor 1 geben
    analogWrite(pwm_1,zahl); //Drehzahl als PWM dem Motor 1 geben

    clearData();
}

void motor2_einstell() //Drehzahl für Motor 2 einstellen
{
    Data[data_count] = '\0'; //String ende
    uint16_t zahl = atoi(Data); //String in Int umwandeln

    digitalWrite(dir_2,LOW); //Richtung für den Motor 2 geben
    analogWrite(pwm_2,zahl); //Drehzahl als PWM dem Motor 2 geben

    Serial.print("Neuer Wert fuer PWM: "); //Return an den Raspberry PI
    Serial.println(zahl);

    clearData();
}

void clearData(){ //String leeren
while(data_count !=0){
Data[data_count--] = 0;
}
return;
}
```

Code (Raspberry PI)

```
1. import sys
2. import serial #Package fuer serielle Verbindungen
3. import time #Package fuer die Funktion sleep
4. import PyQt5.QtWidgets as widgets #Package fuer graphische Oberfläche
5. import PyQt5.uic as uic #Package fuer graphische Oberfläche
6.
7. wert1 = 0 #Hilfsvariable für PWM-Wert des Motors 1
8. wert2 = 0 #Hilfsvariable für PWM Wert des Motors 2
9.
10. verbindung = serial.Serial('/dev/ttyACM0', 9600) #Port zuweisen, an dem der Arduino
    haengt
11. verbindung.isOpen() #Verbindung aufbauen
12. print('Verbindung wird aufgebaut...')
13. time.sleep(3) #3 Sekunden warten
```

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

```
14.
15. app = widgets.QApplication(sys.argv)
16. mainWin = uic.loadUi("gleichstrommotor.ui") #eine QT-Datei einbinden
17.
18.
19. def changePMW_1(pwm): #Funktion zum Einstellen des 1.Motors
20.     mainWin.label_11.setNum(pwm) #akutellen PWM-Wert in Label schreiben
21.     global wert1 #Globale Variable
22.     wert1 = pwm
23.     wert_stern = str(pwm)+ "#" + str(wert2) + "*" #Übermittlungsstring für Arduino b
    asteln
24.     print(wert_stern)
25.     verbindung.write(wert_stern.encode()) #Übermittlungsstring wird an den Arduino g
    esendet
26.     antwort = verbindung.readline()
27.     print(antwort)
28.
29. def changePMW_2(pwm): #Funktion zum Einstellen des 2.Motors
30.     mainWin.label_12.setNum(pwm) #akutellen PWM-Wert in Label schreiben
31.     global wert2 #Globale Variable
32.     wert2 = pwm
33.     wert_stern = str(wert1)+ "#" + str(pwm) + "*" #Übermittlungsstring für Arduino b
    asteln
34.     print(wert_stern)
35.     verbindung.write(wert_stern.encode()) #Übermittlungsstring wird an den Arduino g
    esendet
36.     antwort = verbindung.readline()
37.     print(antwort)
38.
39. def verbindungabbr(): #Abbrechen der Verbindung
40.     changePMW_1(0) #Motor 1 zum Stillstand bringen
41.     changePMW_2(0) #Motor 2 zum Stillstand bringen
42.     verbindung.close() #serielle Verbindung zum Arduino abbrechen
43.
44. mainWin.slider_pwm_1.valueChanged['int'].connect(changePMW_1) #Motor 1 auf Wert des
    Sliders stellen
45. mainWin.slider_pwm_2.valueChanged['int'].connect(changePMW_2) #Motor 2 auf Wert des
    Sliders stellen
46. mainWin.pushButton.clicked.connect(verbindungabbr) #bei Klick auf Button Verbindung
    unterbrechen
47.
48. mainWin.show() #Fenster darstellen
49. sys.exit( app.exec_() ) #Fenster offen bleiben, bis Schließen gedrückt wird
```

Drehzahlmessung-Aufbau

Code (Arduino)

```
#include <Wire.h>
#include <LCD.h> //Package für den LCD-Bildschirm
#include <LiquidCrystal_I2C.h> //Package für das I²C-Modul

//define pin name
#define dir_1 12
#define pwm_1 10

#define Buffer_Length 4

int i=0;
LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);

char Data[Buffer_Length]; //empfangene Daten
byte data_count = 0;  //Zeichenzähler
```

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

```
const int Eingang = 2; //Pin 2 ist Eingang für Sensor
const int AVG = 4; //Glättung über mindestens 4 Messwerte

volatile unsigned long dauer=0; //microsekunden seit dem letzten Interrupt
volatile unsigned long last=0; //Zählerwert beim letzten Interrupt
volatile unsigned long average=1; //Integrierte Dauer // =1: divide by 0
vermeiden
volatile int avgcnt=0; //Anzahl Messwerte im Puffer

void setup() {
  pinMode(pwm_1,OUTPUT); //Zuweisung der PINs
  pinMode(dir_1,OUTPUT);
  pinMode(Eingang, INPUT); //Eingangspin auf Eingang stellen
  digitalWrite(Eingang, HIGH); //und Pullup-Widerstand einschalten

  lcd.begin(20, 4); //Zuweisung der Größe des LCD-Displays
  lcd.setBacklightPin(3,POSITIVE);
  lcd.setBacklight(HIGH);
  lcd.print("Drehzahlmessung"); //Überschrift ausgeben
  lcd.setCursor(0, 1); //cursor an den Anfang der 2. Zeile
  lcd.print("----------------");
  lcd.setCursor(0, 2); //cursor an den Anfang der 3. Zeile
  lcd.print("aktuelle Drehzahl: ");
  lcd.setCursor(10, 3);
  lcd.print("U/min");

  Serial.begin(9600);
  attachInterrupt(0, readmicros, RISING ); //Interrupt 0 auf Routine
readmillis setzen
}

void loop(){

if (Serial.available())
  serialEvent(); //Aufruf der Funktion

else {
  char buf[17]; //Pufferstring für sprintf
  unsigned long drehzahl=0;
  if (dauer != 0) {
    drehzahl = myround(60000000 / average); //Drehzahl ausrechnen und
runden
  } else {
    drehzahl = 0; //keine Messung? -> Stillstand
    avgcnt = 0; //entwerten des Average-Puffers
  }
  sprintf(buf, "    %5lu U/min", drehzahl); //als 5stellig formatierte Zahl
in den Puffer schreiben

  lcd.setCursor(0, 3); //cursor an den Anfang der 4. Zeile
  lcd.print(buf); //Puffer ausgeben
  dauer >>= 10; //Flag für Stillstand ( : 1024 )
}
}

void readmicros() { //Interrupt-Routine
  detachInterrupt(0); //Interrupt ausschalten damit er uns nicht beißt
  int avgmax;
  unsigned long us = micros(); //Microsekundenzähler auslesen
  if (last == 0) { //erster Messwert?
    last = us; //merken und nicht weiter bearbeiten
  } else {
    if ( us < last ) { //Zählerüberlauf
```

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

```
      dauer = 4294967295 - last + us; //erzeugt einen Fehler von 1µS -
vernachlässigbar
    } else {
      dauer = us - last; //Differenz zum letzten Durchlauf berechnen
    }
    if (dauer > 5000) { //ignorieren wenn <= 5ms (Kontaktpreller)
      average = dauer + average * avgcnt++; //Wert in buffer und mit Faktor
avgcnt glätten
      average /= avgcnt; //und zurückrechnen
      avgmax = 1000000 / dauer; //dynamische Größe des Integrationspuffers
      if (avgmax < AVG) avgmax = AVG; //Trägheit mindestens 1 Sekunde
      if (avgcnt >= avgmax) avgcnt--;
      last = us; //den letzten Wert merken
    }
  }
  attachInterrupt(0, readmicros, RISING );  //Interrupt wieder einschalten.
}

unsigned long myround(unsigned long value) {  //Gewichtete Rundung
  int rto;
  if (value > 3000) {  //Rundungswert bestimmen
    rto = 20;
  } else if (value > 1500) {
    rto = 50;
  } else if (value > 500) {
    rto = 10;
  } else if (value > 100) {
    rto = 5;
  } else {
    return (value);
  }
  return (_myround(value, rto));
}

unsigned long _myround(unsigned long value, int roundto) {
  value += (roundto >> 1); //halben roundto Wert addieren
  value /= roundto; //integer division
  value *= roundto; //integer multiplikation
  return (value);
}

void serialEvent() { //serielle Daten werden empfangen
  byte ch = Serial.read(); //Zeichen einlesen
  Data[data_count] = ch;  //auf String abspeichern
  data_count++;
  if (ch == 42) { //Prüfen ob das Zeichen ein * ist
    motor1_einstell(); //Funtkion aufrufen
  }
}

void motor1_einstell() //Drehzahl für Motor 1 einstellen
{
    Data[data_count] = '\0'; //String ende
    uint16_t zahl = atoi(Data); //String in Int umwandeln

    digitalWrite(dir_1,LOW); //Richtung für den Motor 1 geben
    analogWrite(pwm_1,zahl); //Drehzahl als PWM dem Motor 1 geben

    Serial.print("Neuer Wert fuer PWM: "); //Return an den Raspberry PI
    Serial.println(zahl);

    clearData();
}
```

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

```
void clearData(){ //String leeren
while(data_count !=0){
Data[data_count--] = 0;
}
return;
}
```

Code (Raspberry PI)

```
1.  import sys
2.  import serial #Package fuer serielle Verbindungen
3.  import time #Package fuer die Funktion sleep
4.  import PyQt5.QtWidgets as widgets #Package fuer graphische Oberfläche
5.  import PyQt5.uic as uic #Package fuer graphische Oberfläche
6.
7.  wert1 = 0 #Hilfsvariable für PWM-Wert des Motors 1
8.
9.  verbindung = serial.Serial('/dev/ttyACM1', 9600) #Port zuweisen, an dem der Arduino
    haengt
10. verbindung.isOpen() #Verbindung aufbauen
11. print('Verbindung wird aufgebaut...')
12. time.sleep(3) #3 Sekunden warten
13.
14. app = widgets.QApplication(sys.argv)
15. mainWin = uic.loadUi("drehzahlmessung.ui") #eine QT-Datei einbinden
16.
17. def changePMW_1(pwm): #Funktion zum Einstellen des 1.Motors
18.     mainWin.label_11.setNum(pwm) #akutellen PWM-Wert in Label schreiben
19.     wert_stern = str(pwm)+ "*" #Übermittlungsstring für Arduino basteln
20.     print(wert_stern)
21.     verbindung.write(wert_stern.encode()) #Übermittlungsstring wird an den Arduino g
    esendet
22.     antwort = verbindung.readline()
23.     print(antwort)
24.
25. def verbindungabbr(): #Abbrechen der Verbindung
26.     changePMW_1(0) #Motor 1 zum Stillstand bringen
27.     verbindung.close() #serielle Verbindung zum Arduino abbrechen
28.
29. mainWin.slider_pwm_1.valueChanged['int'].connect(changePMW_1) #Motor 1 auf Wert des
    Sliders stellen
30. mainWin.pushButton.clicked.connect(verbindungabbr) #bei Klick auf Button Verbindung
    unterbrechen
31.
32. mainWin.show() #Fenster darstellen
33. sys.exit( app.exec_() ) #Fenster offen bleiben, bis Schließen gedrückt wird
```

Servomotor-Aufbau

Code (Arduino)

```
#include <Wire.h>
#include <VarSpeedServo.h> //Package für Servomotor

#define dir_1 8
#define pwm_1 3
#define dir_2 13
#define pwm_2 11
#define servo_PIN 5
```

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

```
#define Buffer_Length 8

int i=0;
int pos= 15; //Speichert die aktuelle Servoposition

char Data[Buffer_Length]; //empfangene Daten
byte data_count = 0;  //Zeichenzähler

VarSpeedServo myservo; //Servomotor Objekt

void setup(){
Serial.begin(9600);
pinMode(pwm_1,OUTPUT); //Zuweisung der PINS
pinMode(dir_1,OUTPUT);
pinMode(pwm_2,OUTPUT);
pinMode(dir_2,OUTPUT);
myservo.attach(servo_PIN);

myservo.write(pos, 20, true); //auf Position "pos" fahren

}

void loop(){
if (Serial.available())
  serialEvent();  //Aufruf der Funktion

}

void serialEvent() { //serielle Daten werden empfangen
  byte ch = Serial.read();  //Zeichen einlesen
  Data[data_count] = ch;  //auf String abspeichern
  data_count++;
  if (ch == 35) {  //Prüfen ob das Zeichen ein # ist
    motor1_einstell(); //Funktion aufrufen
  }
    if (ch == 37) { //Prüfen ob das Zeichen ein % ist
    motor2_einstell(); //Funktion aufrufen
  }
  if (ch == 42) { //Eingabe mit * beenden
    servo_einstell(); //Funktion aufrufen
  }
}

void motor1_einstell() //Drehzahl für Motor 1 einstellen
{
    Data[data_count] = '\0'; //String ende
    uint16_t zahl = atoi(Data); //String in Int umwandeln

    digitalWrite(dir_1,LOW); //Richtung für den Motor 1 geben
    analogWrite(pwm_1,zahl); //Drehzahl als PWM dem Motor 1 geben

    clearData();
}

void motor2_einstell() //Drehzahl für Motor 2 einstellen
{
    Data[data_count] = '\0'; //String ende
    uint16_t zahl = atoi(Data); //String in Int umwandeln

    digitalWrite(dir_2,LOW); //Richtung für den Motor 2 geben
    analogWrite(pwm_2,zahl); //Drehzahl als PWM dem Motor 2 geben

    clearData();
```

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

```
}

void servo_einstell() //empfangene Daten verarbeiten
{
    Data[data_count] = '\0'; //String ende
    uint16_t zahl = atoi(Data); //String in Int umwandeln

    myservo.write(zahl, 20, true); //Servo an "zahl" fahren

    Serial.print("Neuer Wert fuer Servo: ");  //Return an den Raspberry PI
    Serial.println(zahl);

    clearData();
}

void clearData(){ //String leeren
while(data_count !=0){
Data[data_count--] = 0;
}
return;
}
```

Code (Raspberry PI)

```
1.  import sys
2.  import serial #Package fuer serielle Verbindungen
3.  import time #Package fuer die Funktion sleep
4.  import PyQt5.QtWidgets as widgets #Package fuer graphische Oberfläche
5.  import PyQt5.uic as uic #Package fuer graphische Oberfläche
6.
7.  wert1 = 0 #Hilfsvariable für PWM-Wert des Motors 1
8.  wert2 = 0 #Hilfsvariable für PWM Wert des Motors 2
9.  wert3 = 15 #Hilfsvariable für Winkel des Servomotors
10.
11. verbindung = serial.Serial('/dev/ttyACM0', 9600) #Port zuweisen, an dem der Arduino
    haengt
12. verbindung.isOpen() #Verbindung aufbauen
13. print('Verbindung wird aufgebaut...')
14. time.sleep(3) #3 Sekunden warten
15.
16. app = widgets.QApplication(sys.argv)
17. mainWin = uic.loadUi("servomotor.ui") #eine QT-Datei einbinden
18.
19. def changePMW_1(pwm): #Funktion zum Einstellen des 1.Motors
20.     mainWin.label_11.setNum(pwm) #akutellen PWM-Wert in Label schreiben
21.     global wert1 #Globale Variable
22.     wert1 = pwm
23.     wert_stern = str(pwm)+ "#" + str(wert2) + "%" + str(wert3) + "*" #Übermittlungss
    tring für Arduino basteln
24.     print(wert_stern)
25.     verbindung.write(wert_stern.encode()) #Übermittlungsstring wird an den Arduino g
    esendet
26.     antwort = verbindung.readline()
27.     print(antwort)
28.
29. def changePMW_2(pwm): #Funktion zum Einstellen des 2.Motors
30.     mainWin.label_12.setNum(pwm) #akutellen PWM-Wert in Label schreiben
31.     global wert2 #Globale Variable
32.     wert2 = pwm
33.     wert_stern = str(wert1)+ "#" + str(pwm) + "%" + str(wert3) + "*" #Übermittlungss
    tring für Arduino basteln
34.     print(wert_stern)
```

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

```
35.     verbindung.write(wert_stern.encode()) #Übermittlungsstring wird an den Arduino g
    esendet
36.     antwort = verbindung.readline()
37.     print(antwort)
38.
39. def changeServo(winkel): #Funktion zum Einstellen des Servomotors
40.     mainWin.label_15.setNum(winkel) #akutellen Winkel in Label schreiben
41.     global wert3 #Globale Variable
42.     wert3 = winkel
43.     wert_stern = str(wert1)+ "#" + str(wert2) + "%" + str(winkel) + "*" #Übermittlun
    gsstring für Arduino basteln
44.     print(wert_stern)
45.     verbindung.write(wert_stern.encode()) #Übermittlungsstring wird an den Arduino g
    esendet
46.     antwort = verbindung.readline()
47.     print(antwort)
48.
49. def verbindungabbr(): #Abbrechen der Verbindung
50.     changePMW_1(0) #Motor 1 zum Stillstand bringen
51.     changePMW_2(0) #Motor 2 zum Stillstand bringen
52.     changeServo(15) #Servo auf Startposition bringen
53.     verbindung.close() #serielle Verbindung zum Arduino abbrechen
54.
55. mainWin.slider_pwm_1.valueChanged['int'].connect(changePMW_1) #Motor 1 auf Wert des
    Sliders stellen
56. mainWin.slider_pwm_2.valueChanged['int'].connect(changePMW_2) #Motor 2 auf Wert des
    Sliders stellen
57. mainWin.slider_servo.valueChanged['int'].connect(changeServo) #Motor 2 auf Wert des
    Sliders stellen
58. mainWin.pushButton.clicked.connect(verbindungabbr) #bei Klick auf Button Verbindung
    unterbrechen
59.
60. mainWin.show() #Fenster darstellen
61. sys.exit( app.exec_() ) #Fenster offen bleiben, bis Schließen gedrückt wird
```

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Anhang B
(Datenblätter)

DATENBLATT

Artikel: Gleichstrommotor RS PRO 12V

Spezifikation

Versorgungsspannung: 12 V DC
Leistung: 36,88 W
Abtriebsdrehzahl: 4289 U/min
Schaftdurchmesser: 6.35mm
Abtriebsdrehmoment max.: 82,08 gcm
Länge: 69mm
Abmessungen: Ø 51,8 x 69 mm
Nennstrom: 5,28 A
Gehäusemontage: Eisenkern

Der RS PRO DC Motor ist eine Reihe von Motoren, die eine Vielzahl von Hochleistungsmotoren aufweisen, die für eine Änderung des Drehzahlmoments, der Motordrehzahl und der Spannung unerlässlich sind. Eine Durable Konstruktion gewährleistet, dass die Anwendung einen hohen Standard erreicht.

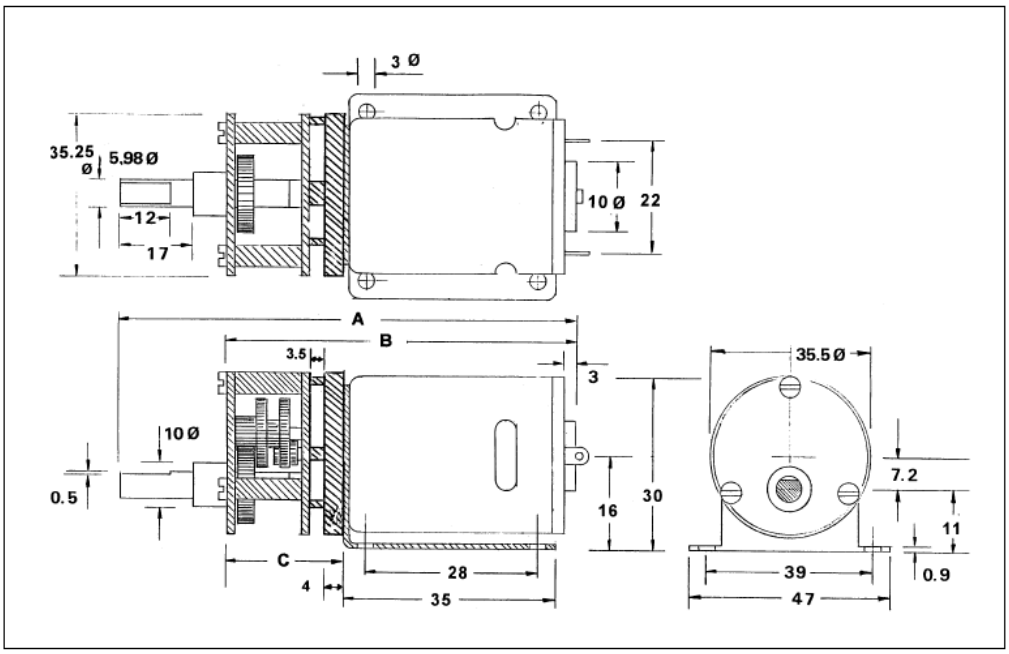
Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

DATENBLATT

Artikel: Getriebemotor RS PRO 12V

MOTOR DATA.

MODEL	VOLTAGE		NO LOAD		AT MAXIMUM EFFICIENCY						STALL TORQUE	
	OPERATING RANGE	NOMINAL	SPEED	CURRENT	SPEED	CURRENT	TORQUE		OUTPUT	EFF		
			R.P.M.	A	R.P.M.	A	oz - in	g - cm	W	%	oz - in	g - cm
	6.0 - 15.0	12v CONSTANT	11000	0.155	9281	0.837		65.3	6.21	61.85		417.6



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

QSH 6018

60 mm / NEMA24
1.8° step angle
high torque
hybrid stepper motor

INFO These two phase hybrid stepper motors are optimized for microstepping and give a good fit to the TRINAMIC family of motor controllers and drivers.

MAIN CHARACTERISTICS

- NEMA 23 mounting configuration
- flange max. 60.5 mm * 60.5 mm
- 8 mm axis diameter, 25 mm length, D-Cut
- step angle: 1.8°
- optimized for microstep operation
- optimum fit for TMC239 / TMC249 base driver circuits
- up to 75V operating voltage
- 4 wire connection
- neodymium magnets for maximum torque
- CE approved
- RoHS compliant

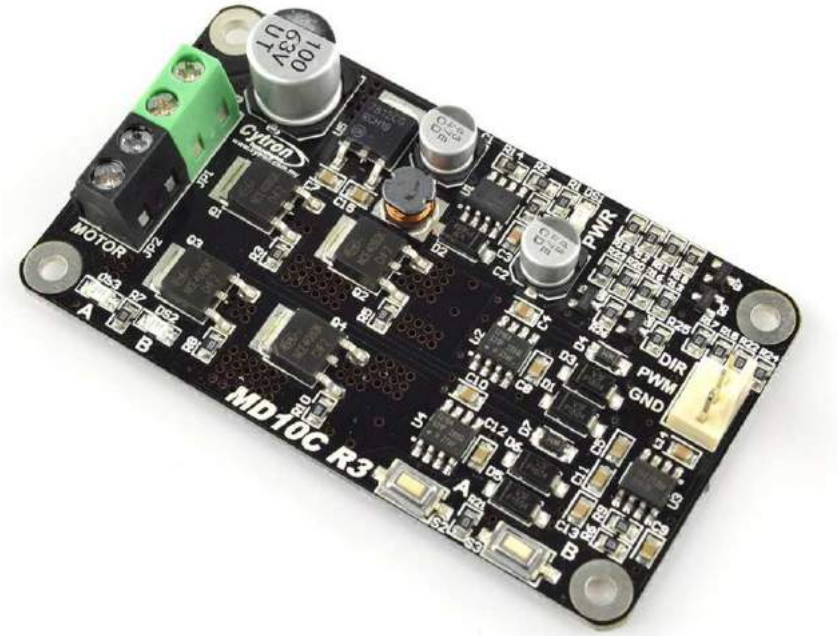
SPECIFICATIONS	UNIT	-45-28-110	-56-28-165	-65-28-210	-86-28-310
Rated phase current	A	2.8	2.8	2.8	2.8
Ph. resistance at 20°C		0.75	0.9	1.2	1.5
Ph. inductance (typ.)	mH	2	3.6	4.6	6.8
Holding torque (typ.)	Ncm	110	165	210	310
Rotor inertia	gcm²	275	400	570	840
Weight (mass)	kg	0.6	0.77	1.20	1.40
Motor length	mm	45.0	56.0	65.0	86.0

ORDER CODE	DESCRIPTION
QSH6018-45-28-110	QMot Steppermotor 60 mm, 2.8A, 1.10 Nm
QSH6018-56-28-165	QMot Steppermotor 60 mm, 2.8A, 1.65 Nm
QSH6018-65-28-210	QMot Steppermotor 60 mm, 2.8A, 2.10 Nm
QSH6018-86-28-310	QMot Steppermotor 60 mm, 2.8A, 3.10 Nm

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

DATENBLATT

Artikel: Motortreiber Cytron 13A



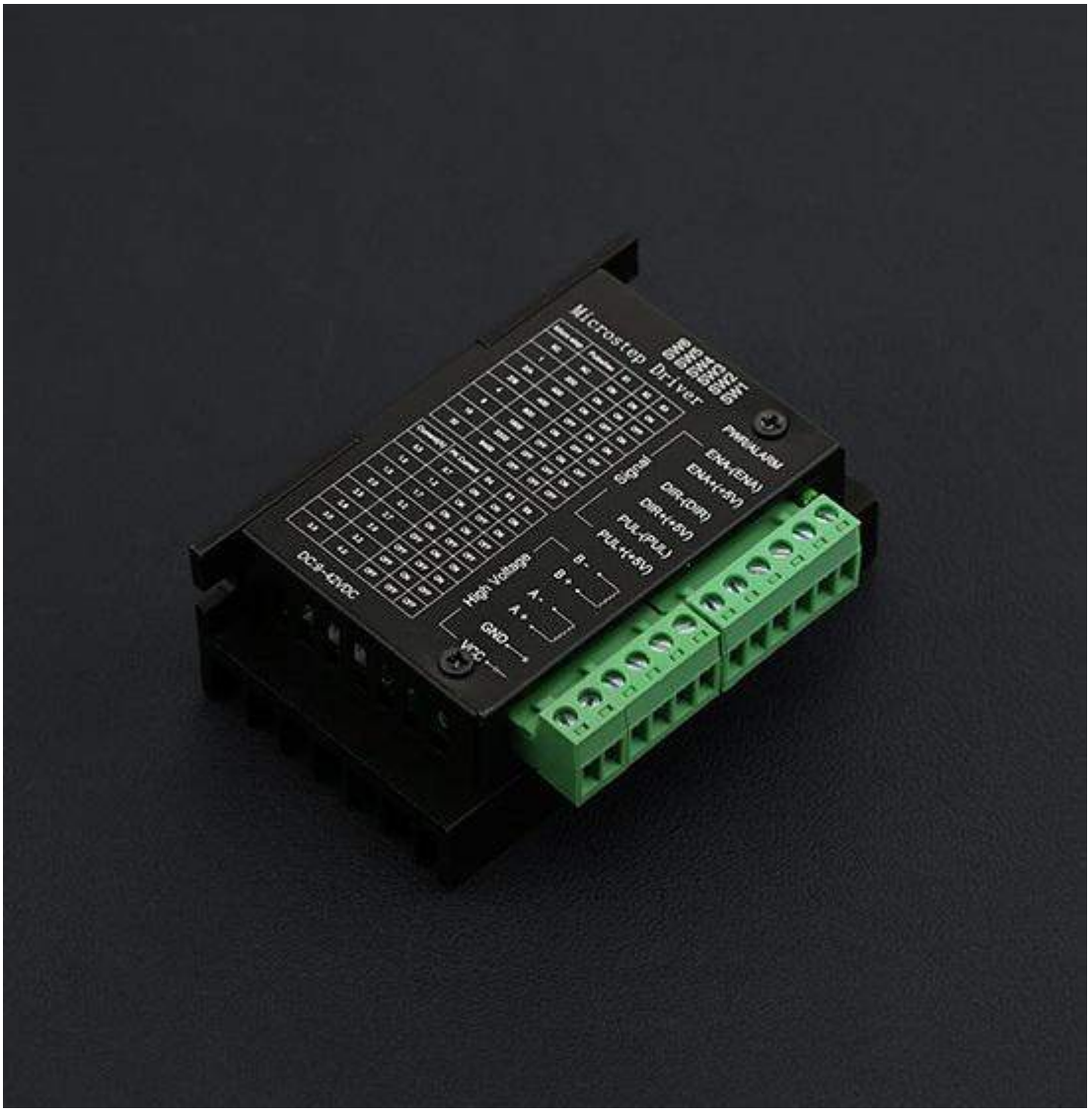
Spezifikation

- Bidirektionale Steuerung für 1 Gleichstrom-Bürstenmotor
- Unterstützt Motorspannungsbereiche von 5V bis 30V
- Maximaler Strom bis zu 13A Dauer und 30A Spitze (10 Sekunden)
- Vollständige NMOS H-Brücke für bessere Effizienz und keine Kühlkörper erforderlich

Der **Cytron 13A, 5-30V Single DC-Motorkontroller** ist eine verbesserte Version des MD10B, die entwickelt wurde, um **Gleichstrom-Bürstenmotoren** mit hohen Strömen bis zu 13A kontinuierlich zu betreiben. Er bietet mehrere Verbesserungen gegenüber dem MD10B, wie z. B. die Unterstützung sowohl für ein gesperrt-gegenphasiges als auch ein vorzeichenmagnituden PWM-Signal sowie für die Verwendung von Vollfestkörperkomponenten, die zu einer schnelleren Reaktionszeit führen und den Verschleiß des mechanischen Relais beseitigen.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

TB6600 Stepper Motor Driver User Guide



Version: V1.2

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Contents

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Safety Precautions:

- Before using this product, please read this instruction manual carefully
- Keep this manual in a safe place for future reference
- The appearance of the picture is just for reference, please prevail in kind

❖ This device is driven by DC power supply, make sure the power positive and negative before you power it.

❖ Please do not electrified plug

❖ Please do not mix conductive foreign matter such as screws or metal

❖ Please keep it dry, and pay attention to moisture-proof

❖ The equipment should be clean and well ventilated.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

1. Introduction

This is a professional two-phase stepper motor driver. It supports speed and direction control. You can set its micro step and output current with 6 DIP switch. There are 7 kinds of micro steps (1, 2 / A, 2 / B, 4, 8, 16, 32) and 8 kinds of current control (0.5A, 1A, 1.5A, 2A, 2.5A, 2.8A, 3.0A, 3.5A) in all. And all signal terminals adopt high-speed optocoupler isolation, enhancing its anti-high-frequency interference ability.

Features:

※ Support 8 kinds of current control
※ Support 7 kinds of micro steps adjustable
※ The interfaces adopt high-speed optocoupler isolation
※ Automatic semi-flow to reduce heat
※ Large area heat sink
※ Anti-high-frequency interference ability
※ Input anti-reverse protection
※ Overheat, over current and short circuit protection

Electrical Specification:

Input Current	0~5.0A
Output Current	0.5-4.0A
Power (MAX)	160W
Micro Step	1, 2/A, 2/B, 4, 8, 16, 32
Temperature	-10～45℃
Humidity	No Condensation
Weight	0.2 kg
Dimension	96*56*33 mm

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

INPUT & OUTPUT:

- **Signal Input:**

PUL+	Pulse +
PUL-	Pulse -
DIR+	Direction +
DIR-	Direction -
EN+	Off-line Control Enable +
EN-	Off-line Control Enable -

- **Motor Machine Winding:**

A+	Stepper motor A+
A-	Stepper motor A-
B+	Stepper motor B+
B-	Stepper motor B-

- **Power Supply:**

VCC	VCC (DC9-42V)
GND	GND

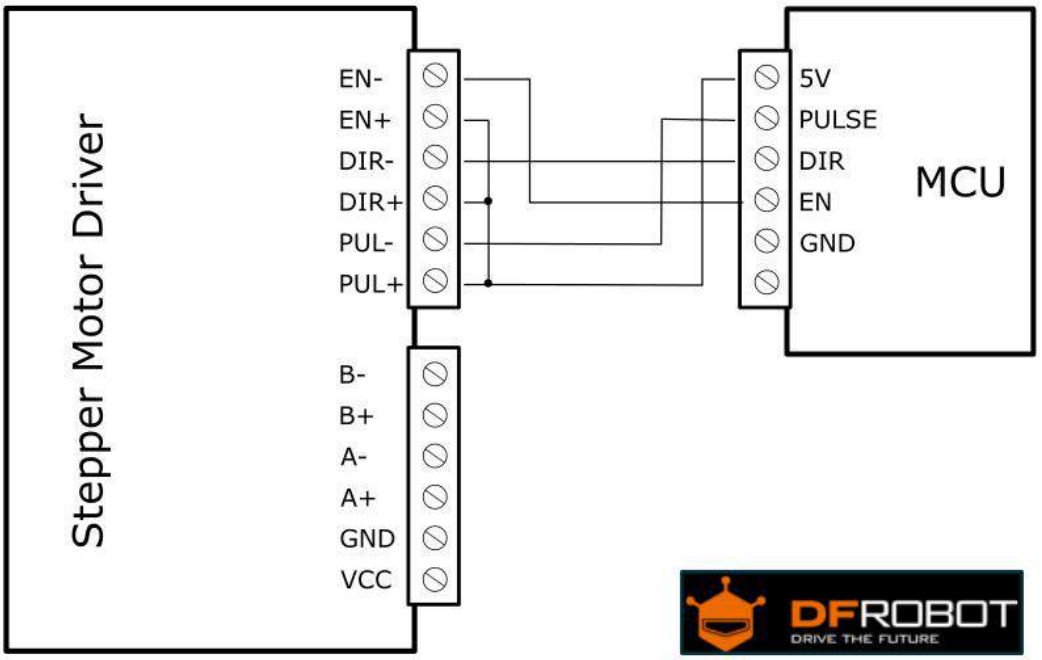
- **Wiring instructions**

There are three input signals in all: ① Step pulse signal PUL +, PUL-; ② Direction signal DIR +, DIR-; ③ off-line signal EN +, EN-. The driver supports common-cathode and common-anode circuit, you can select one according to your demand.

Common-Anode Connection:

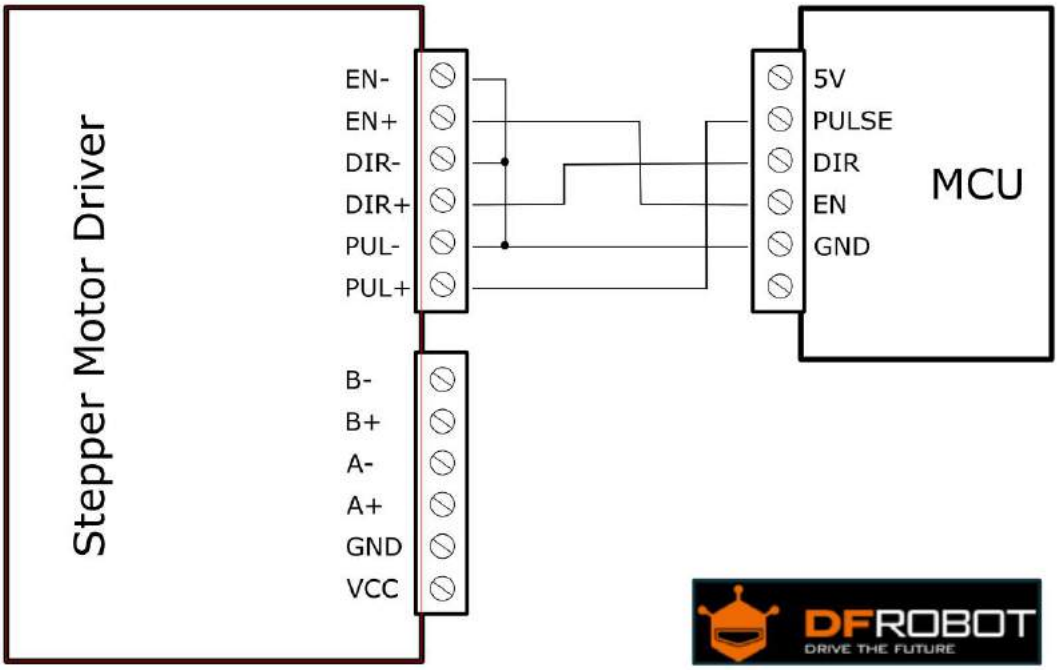
Connect PUL +, DIR + and EN + to the power supply of the control system. If the power supply is + 5V, it can be directly connected. If the power supply is more than + 5V, the current limiting resistor R must be added externally. To ensure that the controller pin can output 8 ~ 15mA current to drive the internal optocoupler chip. Pulse signal connects to PUL-; direction signal connects to Dir- ; Enable signal connects to EN-. As shown below:

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.



Common-Cathode Connection:

Connect PUL -, DIR - and EN - to the ground terminal of the control system. Pulse signal connects to PUL-; direction signal connects to Dir- ; Enable signal connects to EN-. As shown below:

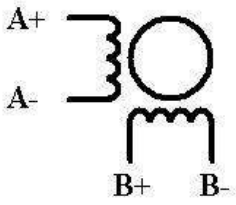


Note: When "EN" is in the valid state, the motor is in a free states (Off-line mode). In this mode, you can adjust the motor shaft position manually. When "EN" is in the invalid state, the motor will be in an automatic control mode.

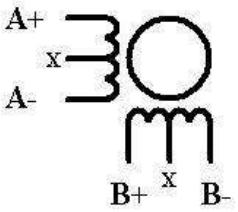
Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

2. Stepper Motor Wiring:

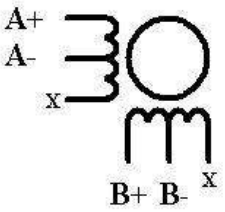
Two-phase 4-wire, 6-wire, 8-wire motor wiring, as shown below:



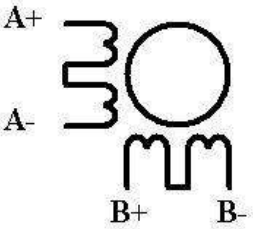
四线电机接线方法



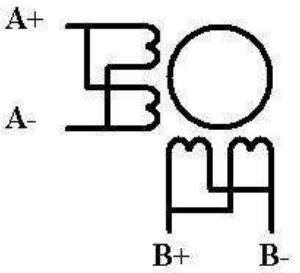
六线电机接线方法
高力矩输出



六线电机接线方法
高速度输出



八线电机接线方法
高力矩输出

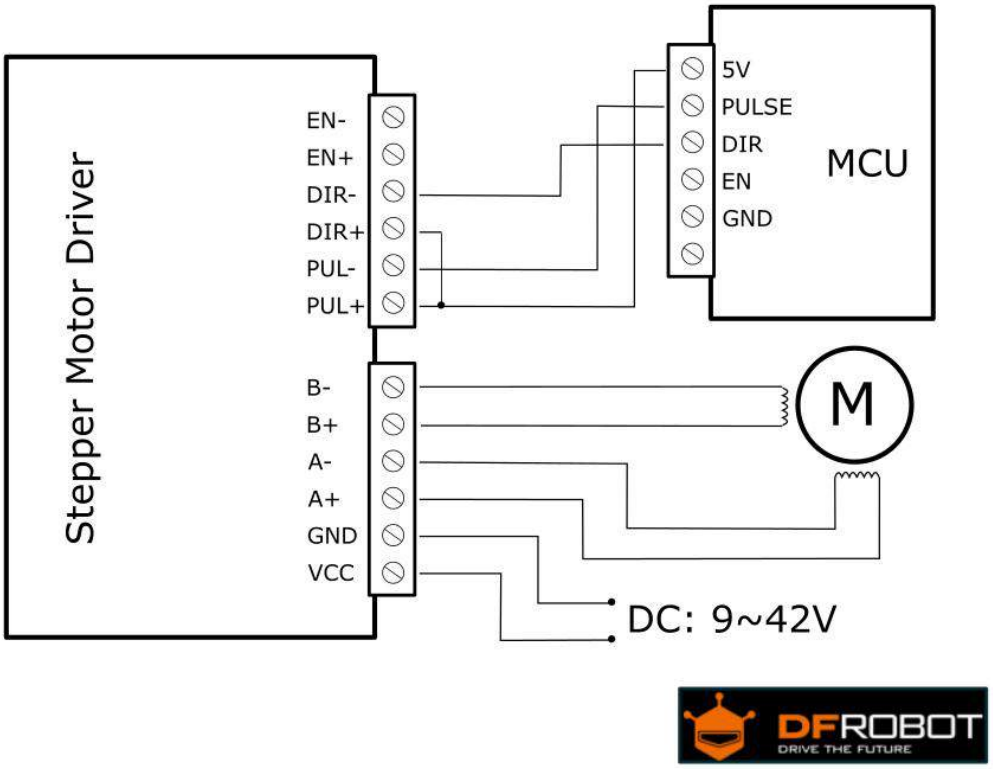


八线电机接线方法
高速度输出

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

3. Microcontroller Connection Diagram:

This is an example for the common-anode connection. (“EN” not connected)



Note: Please cut off the power when you connect the system, and ensure the power polar is correct. Or it will damage the controller.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

4. DIP Switch

Micro Step Setting

The follow tablet shows the driver Micro step. You can set the motor micro step via the first three DIP switch.

Step Angle = Motor Step Angle / Micro Step

E.g. An stepper motor with 1.8° step angle，the finial step angle under "Micro step 4" will be 1.8°/4=0.45°

Micro Step	Pulse/Rev	S1	S2	S3
NC	NC	ON	ON	ON
1	200	ON	ON	OFF
2/A	400	ON	OFF	ON
2/B	400	OFF	ON	ON
4	800	ON	OFF	OFF
8	1600	OFF	ON	OFF
16	3200	OFF	OFF	ON
32	6400	OFF	OFF	OFF

Current Control Setting

Current (A)	S4	S5	S6
0.5	ON	ON	ON
1.0	ON	OFF	ON
1.5	ON	ON	OFF
2.0	ON	OFF	OFF
2.5	OFF	ON	ON
2.8	OFF	OFF	ON
3.0	OFF	ON	OFF
3.5	OFF	OFF	OFF

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

5. Off-line Function (EN Terminal):

If you turn on the Off-line function, the motor will enter a free state. You can adjust the motor shaft freely, and the pulse signal will be no response. If you turn it off, it will be back into automatic control mode

Note: Generally, EN terminal is not connected.

6. FAQ

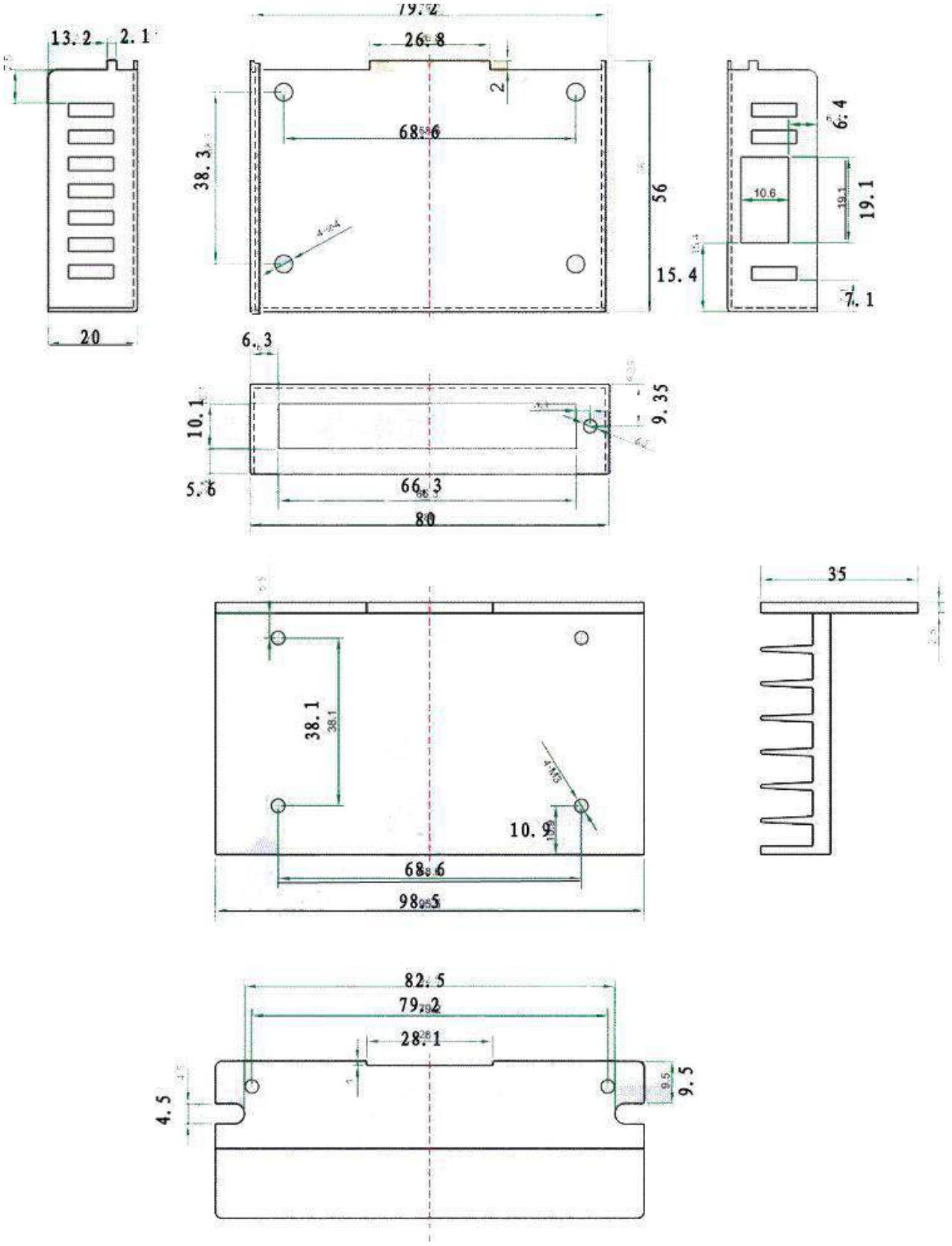
1. Q: If the control signal is higher than 5V, how do I connect?
A: You need add a resistor in series

2. Q: After connected the power, why the motor doesn’ t work? The PWR Led has been ON.
A: Please check the power supply, it must higher than 9V. And make sure the I/O limited current is higher than 5mA

3. Q: How do we know the right order of the stepper motor?
A: Please check the motor specification, it show you the right order. Or you can measure it with a multimeter.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

7. Dimension (96*56*33)



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

DATENBLATT

Artikel: 600 Watt LED Netzteil

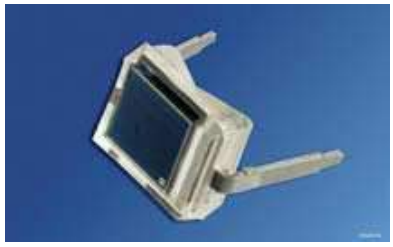
Spezifikation

Eingangsspannung: 175 ~ 240 Volt AC
Eingangsfrequenz: 50/60 Hz
Ausgangstyp: Konstante Spannung Einzelausgang
Ausgangsspannung: 12 Volt DC
Ausgangsstrom: 0 ~ 50 Ampere
Ausgangsleistung: 0 ~ 600 Watt
Ausgangsfrequenz: 100k Hz
Wirkungsgrad: 85%
Shell Material: Aluminium
Farbe: Silber
IP-Niveau: 20
Wasserdicht: Nein
Verwendung: Innenanwendung
Arbeitstemperatur: -40 ~ + 60 Celsius, 20% ~ 90% RH
Fan: JA
Kühlmethode: Freie Luftkonvektion
Variable Widerstand: Ja
Kundenspezifisch: Ja
Herkunft: Shenzhen, China

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Silicon PIN Photodiode
Silizium-PIN-Fotodiode
Version 1.1

BPW 34



BPW 34

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Features:
- Especially suitable for applications from 400 nm to 1100 nm
- Short switching time (typ. 20 ns)
- DIL plastic package with high packing density

Besondere Merkmale:
- Speziell geeignet für Anwendungen im Bereich von 400 nm bis 1100 nm
- Kurze Schaltzeit (typ. 20 ns)
- DIL-Plastikbauform mit hoher Packungsdichte

Applications
- Photointerrupters
- Industrial electronics
- For control and drive circuits
- IR remote control of hi-fi and TV sets, video tape recorders, dimmers, remote controls of various equipment

Anwendungen
- Lichtschranken
- Industrieelektronik
- Messen / Steuern / Regeln
- IR-Fernsteuerung von Fernseh- und Rundfunkgeräten, Videorecordern, Lichtdimmern, Gerätefernsteuerungen

Ordering Information
Bestellinformation

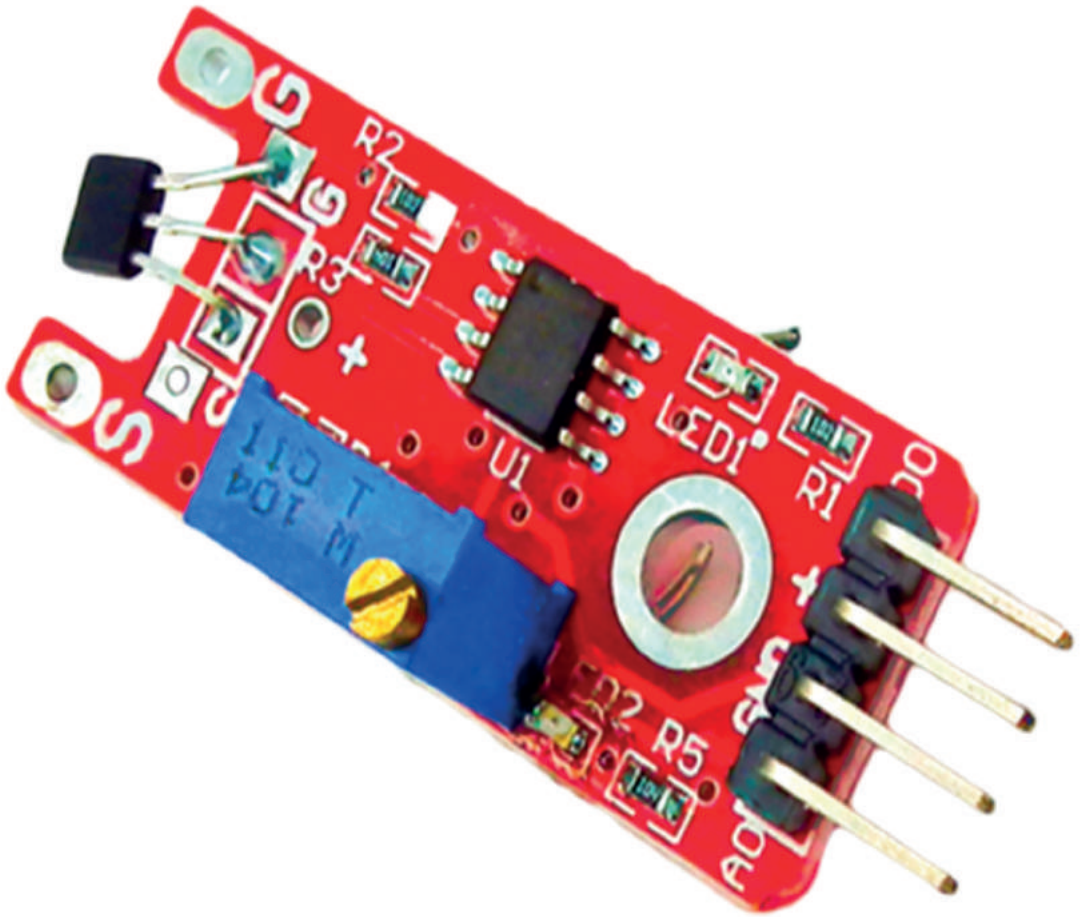
Type: Typ:	Photocurrent Fotostrom E_v = 1000 lx, Std. Light A, V_R = 5 V I_P [µA]	Ordering Code Bestellnummer
BPW 34	80 (≥ 50)	Q62702P0073

OSRAM Opto Semiconductors

AZ-Delivery

Ihr Experte für Mikroelektronik!

Hall Sensor Modul
Datenblatt



Contents:

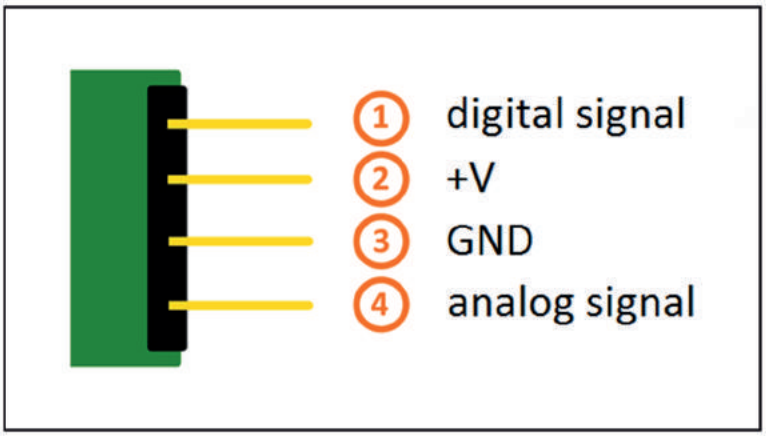
1. Short description

Chipset: A3141 | OP-amplifier: LM393
A magnetic field is detected by the sensor and will be printed as an analog voltage value.
You can control the sensitivity of the sensor with the potentiometer.

Digital out: If a magnetic field is detected by the sensor, a signal will be printed here
Analog out: Direct measurement of the sensor unit

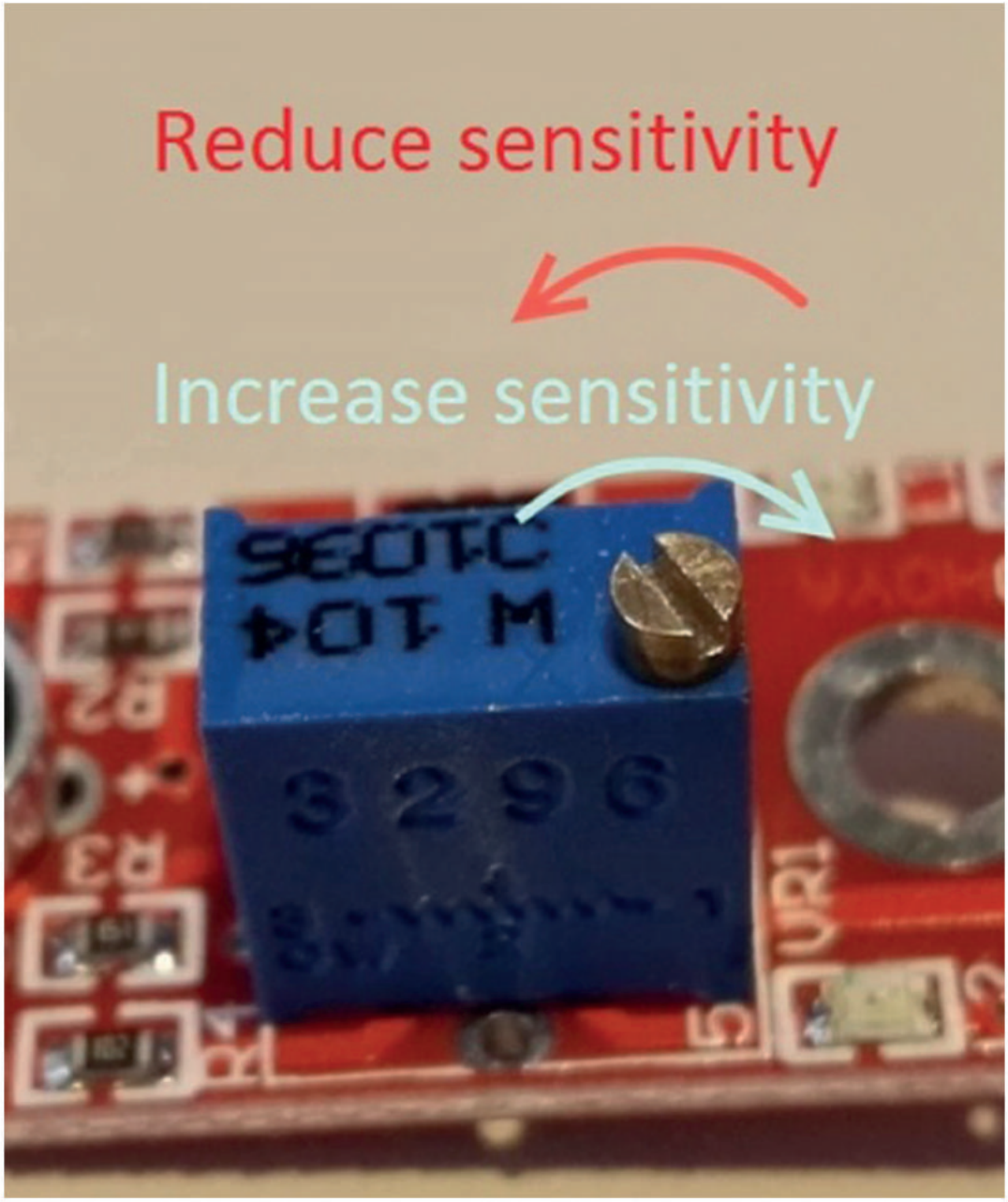
LED1: Shows that the sensor is supplied with voltage
LED2: Shows that the sensor detects a magnetic field

2. Pinout



3. Functionality of the sensor

The sensor has 3 main components on its circuit board. First, the sensor unit at the front of the module which measures the area physically and sends an analog signal to the second unit, the amplifier. The amplifier amplifies the signal, according to the resistant value of the potentiometer, and sends the signal to the analog output of the module.
The third component is a comparator which switches the digital out and the LED if the signal falls under a specific value.
You can control the sensitivity by adjusting the potentiometer.
Please notice: The signal will be inverted; that means that if you measure a high value, it is shown as a low voltage value at the analog output.



This sensor doesn't show absolute values (like exact temperature in °C or magneticfield strenght in mT).
It is a relative measurement: you define an extreme value to a given normal environment situation and a signal will be send if the measurement exceeds the extreme value.

It is perfect for temperature control (KY-028), proximity switch (KY-024, KY-025, KY-036), detecting alarms (KY-037, KY-038) or rotary encoder (KY-026).

4. Connections Arduino

digital signal = [Pin 3]
+V= [Pin 5V]
GND = [Pin GND]
analog signal = [Pin 0]

5. Connections Raspberry Pi

Sensor

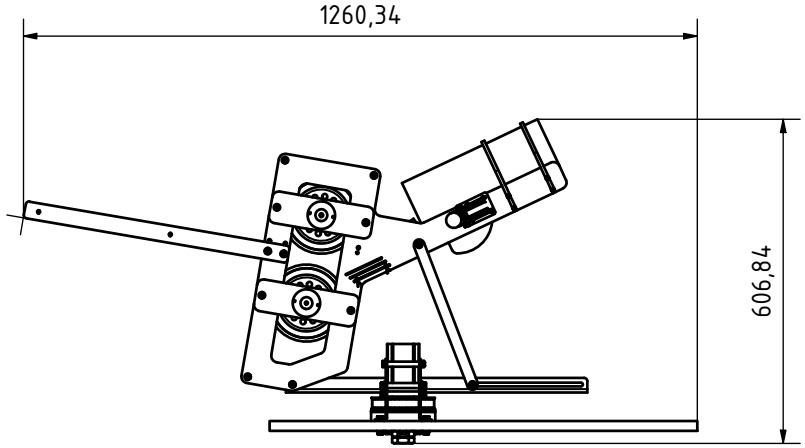
digital signal = GPIO 24 [Pin 18 (RPi)]
+V = 3,3V [Pin 1 (RPi)]
GND= GND[Pin 06 (RPi)]
analog signal = Analog 0 [Pin A0 (ADS1115 - KY-053)]

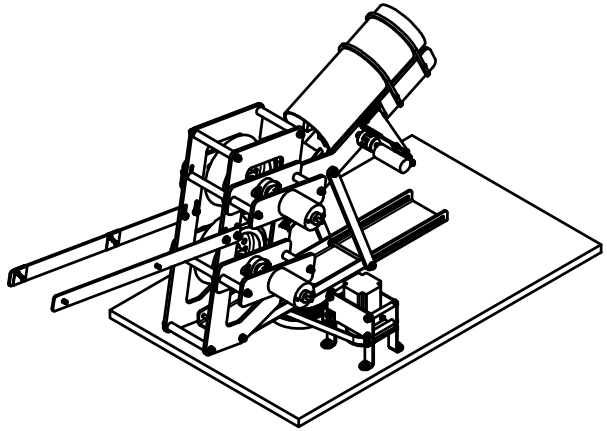
ADS1115 - KY-053:

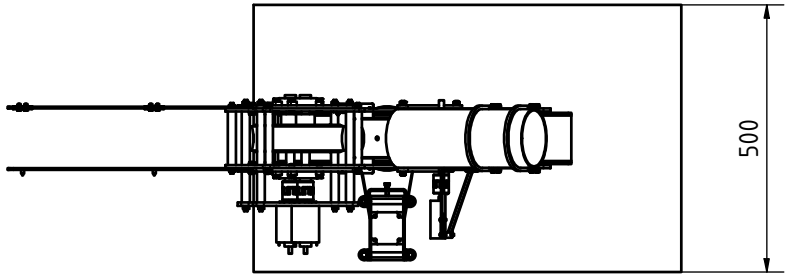
VDD= 3,3V [Pin 01]
GND= GND[Pin 09]
SCL = GPIO03 / SCL [Pin 05]
SDA= GPIO02 / SDA [Pin 03]
A0 = look above [Sensor: analog signal]

Anhang C (Zeichnungen Funktionsmuster)

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN







Stück	Bezeichnung	Teil	Werkstoff	Norm	Bemerkung
2	Sechskantmutter M25	T62	Stahl	DIN 439	
4	Sechskantmutter M6	T61	Stahl	DIN 934	
4	Sechskantschraube M6x30	T57	Stahl	DIN 933	
4	Sechskantschraube M6x28	T56	Stahl	DIN 933	
4	Sechskantschraube M6x20	T54	Stahl	DIN 933	
16	Unterlegscheibe 6,4	T48	Stahl	DIN 125 A	
1	Zahnriemen T10	T46	Polyurethan		Breite: 16mm, Wirklänge: 610mm
1	Bolzen	T45	1.4301		
1	Holzplatte	T44	Birkenholz		800 x 500 cm
1	Gleitlager	B03			
1	Schrittmotor-Einheit	B02			
1	Abschuss-Einheit	B01			

Maßstab: 1:10

Allg.-Toleranz: ISO 2768-1 m

Institut für Konstruktionswissenschaften

TU Wien

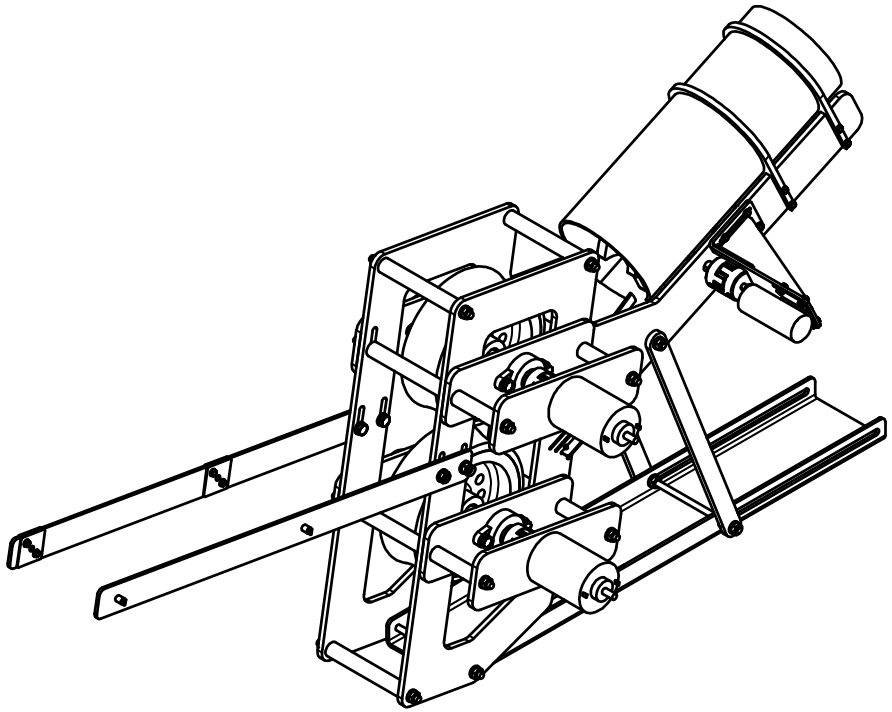
Benennung:

Zusammenbau

Name: Eichberger

Matr.-Nr.: 00909186

Datum: 09.11.2019



Stück	Bezeichnung	Teil	Werkstoff	Norm	Bemerkung
4	Splint 2x10	T68	Stahl	DIN 94	
7	Gewindestift M6x12	T66	Stahl	DIN 913	
2	Gewindestift M6x10	T65	Stahl	DIN 913	
2	Senkkopfschraube M3x8	T63	Stahl	DIN 7991	
8	Sechskantmutter M6	T61	Stahl	DIN 934	
16	Sechskantmutter M3	T60	Stahl	DIN 934	
4	Sechskantschraube M6x18	T53	Stahl	DIN 933	
8	Sechskantschraube M3x20	T52	Stahl	DIN 933	
6	Sechskantschraube M3x14	T51	Stahl	DIN 933	
12	Unterlegscheibe 6,4	T48	Stahl	DIN 125 A	
30	Unterlegscheibe 3,2	T47	Stahl	DIN 125 A	
1	Einschubblech	T33	1.4301		
1	Rohr 2	T32	Polypropylen		
1	Rohr 1	T31	Polypropylen		
2	Halterung Rohr 2	T30	EN AW-5754 (AlMg3)		
1	Kupplung 2	T29			
1	Ballrad-Welle	T28	S235JR		
1	Ballrad	T27	EN AW-5754 (AlMg3)		
2	Kupplung 1	T26			
2	Schwungrad-Welle	T25	S235JR		
2	Schwungrad	T24			
1	Fotodiode	T23			
2	Fotodioden-Aufnahme	T22	1.4301		
1	Laser	T20			
1	Diodenhalterungs-Einheit	B08			
1	Laserhalterungs-Einheit	B07			
1	Getriebemotor-Einheit	B06			
2	Gleichstrommotor-Einheit	B05			
1	Grundgerüst	B04			

Maßstab: 1:5

Allg.-Toleranz: ISO 2768-1 m

Institut für Konstruktionswissenschaften

TU Wien

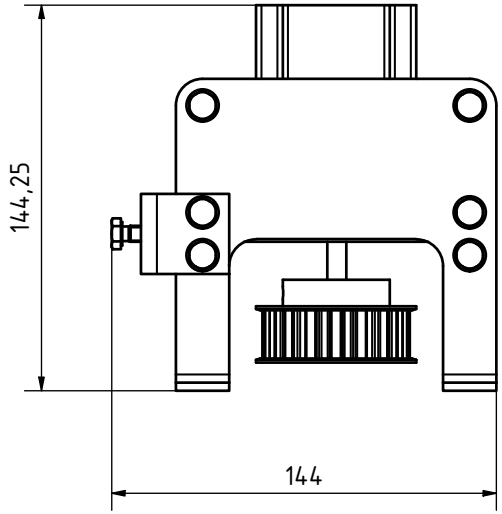
Benennung:

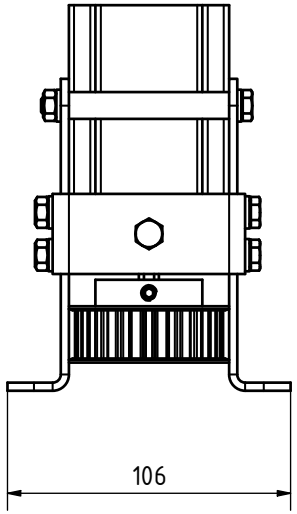
Abschuss-Einheit

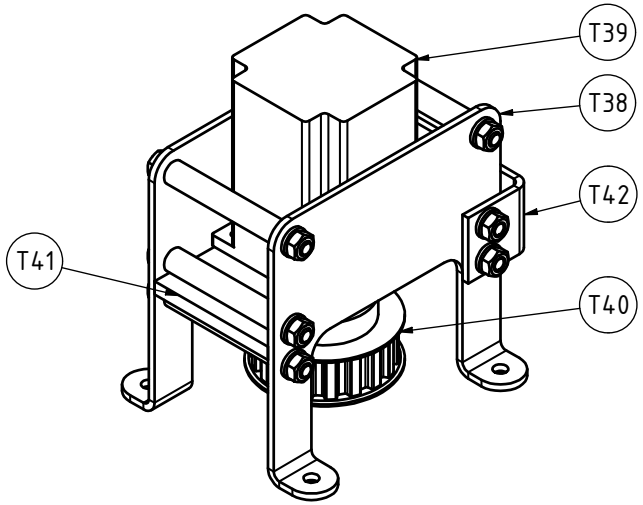
Name: Eichberger

Matr.-Nr.: 00909186

Datum: 09.10.2019







Stück	Bezeichnung	Teil	Werkstoff	Norm	Bemerkung
1	Gewindestift M6x16	T67	Stahl	DIN 913	
6	Sechskantmutter M6	T61	Stahl	DIN 934	
2	Sechskantschraube M6x80	T59	Stahl	DIN 933	
4	Sechskantschraube M6x75	T58	Stahl	DIN 933	
1	Sechskantschraube M6x20	T54	Stahl	DIN 933	
12	Unterlegscheibe 6,4	T48	Stahl	DIN 125 A	
6	Distanzhülse 4	T43	1.4301		
1	Spannteil	T42	1.4301		
1	Halteplatte	T41	EN AW-5754 (AlMg3)		
1	Zahnriemenrad 1	T40	AlCuMg1		
1	Schrittmotor	T39	Aluminium		
2	Halterung Schrittmotor	T38	1.4301		

Maßstab: 1:2

Allg.-Toleranz: ISO 2768-1 m

Institut für Konstruktionswissenschaften

TU Wien

Benennung:

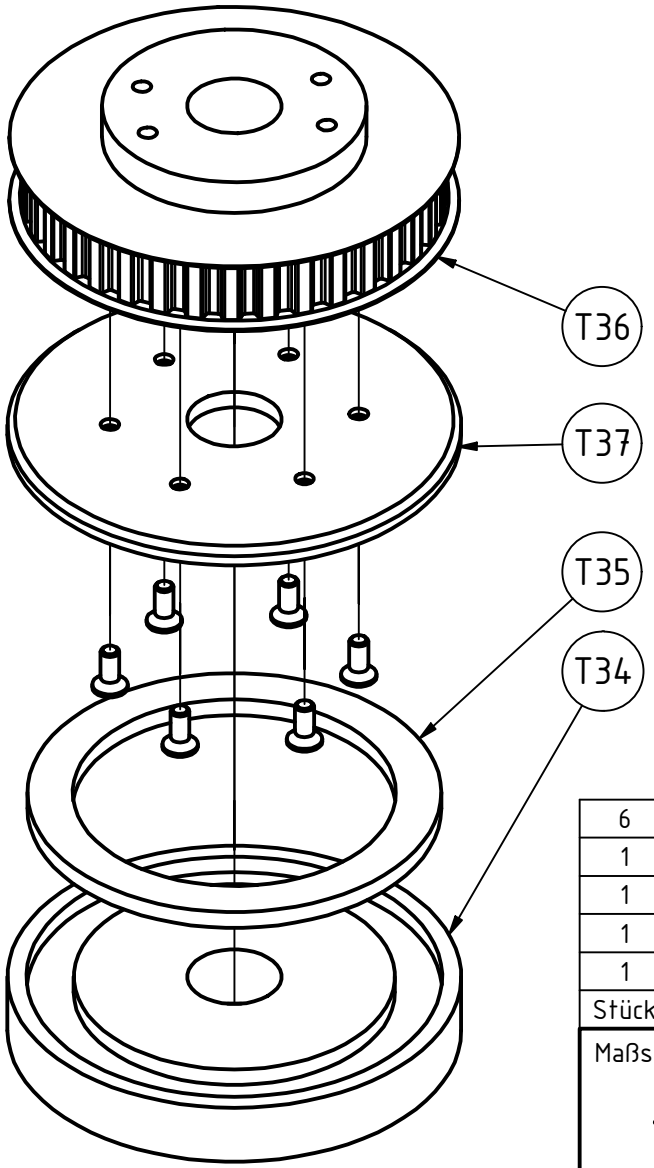
Schrittmotor-Einheit

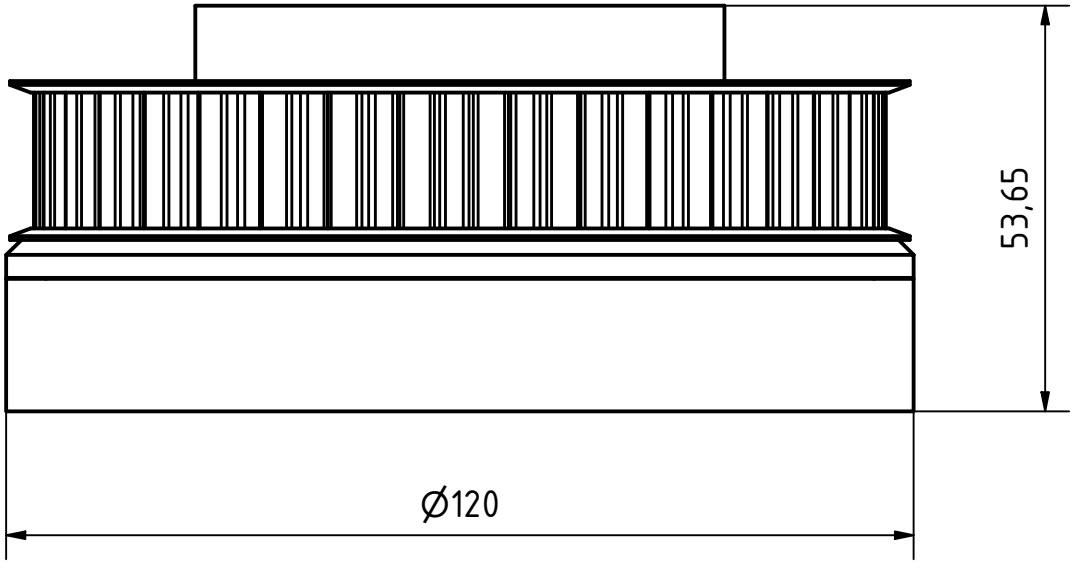
Name: Eichberger

Matr.-Nr.: 00909186

Datum: 07.10.2019

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub





Stück	Bezeichnung	Teil	Werkstoff	Norm	Bemerkung
6	Senkkopfschraube M5x12	T64	Stahl	DIN 7991	
1	Zwischenscheibe	T37	EN AW-5754 (AlMg3)		
1	Zahnriemenrad 2	T36	AlCuMg1		
1	Gleitscheibe	T35	Bronze		
1	Gleitlager-Scheibe	T34	EN AW-5754 (AlMg3)		

Maßstab: 1:1

Allg.-Toleranz: ISO 2768-1 m

Institut für Konstruktionswissenschaften

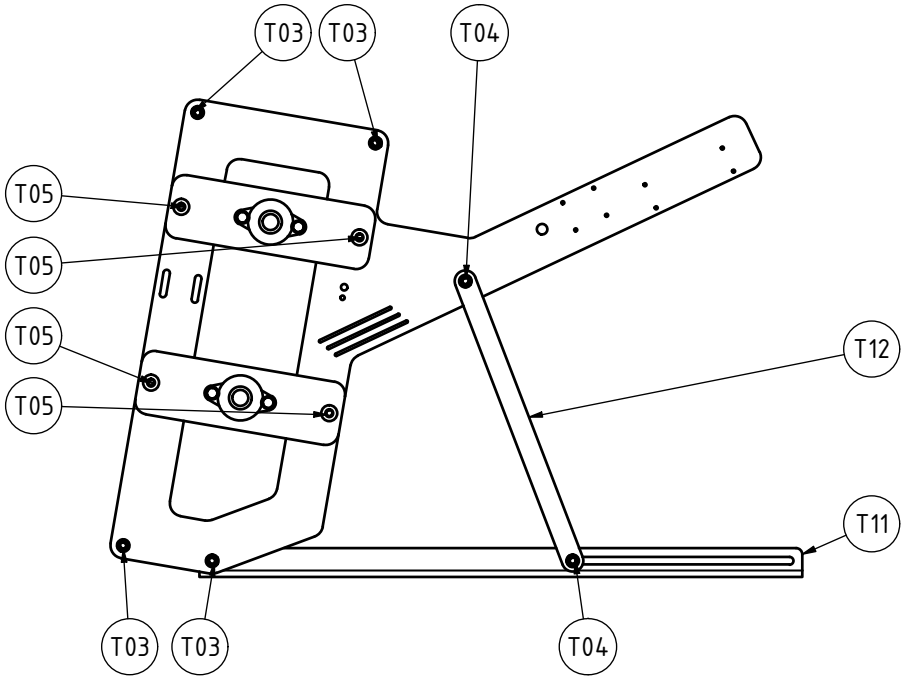
TU Wien

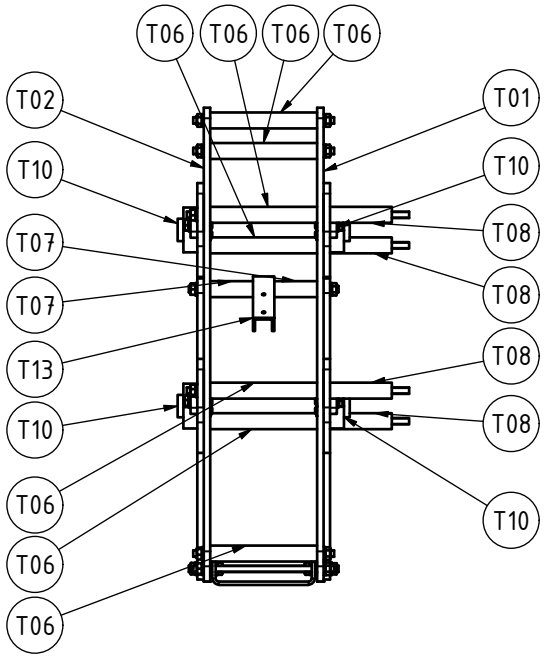
Benennung:

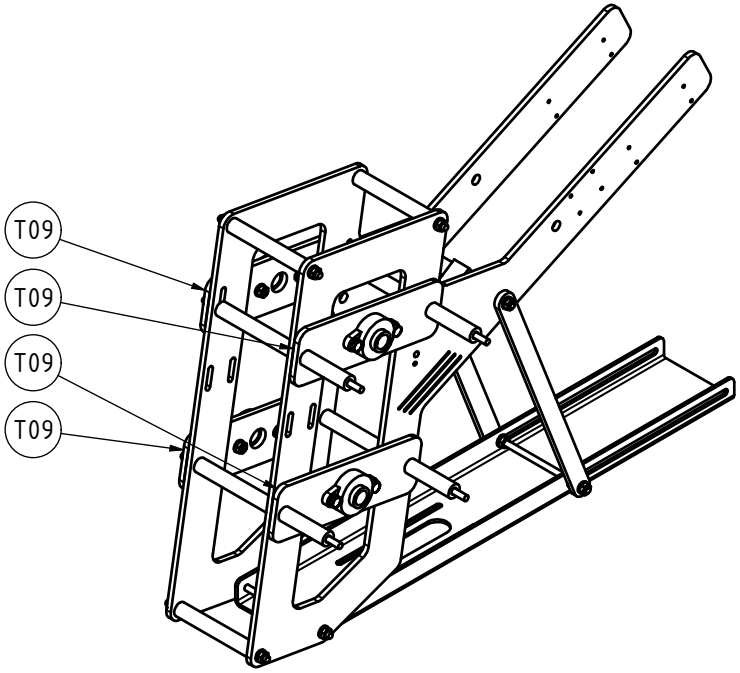
Gleitlager

Name: Eichberger
Matr.-Nr.: 00909186

Datum: 07.10.2019







Stück	Bezeichnung	Teil	Werkstoff	Norm	Bemerkung
28	Sechskantmutter M6	T61	Stahl	DIN 934	
8	Sechskantschraube M6x22	T55	Stahl	DIN 933	
36	Unterlegscheibe 6,4	T48	Stahl	DIN 125 A	
1	U-Profil	T13	Aluminium		
2	Stange	T12	1.4301		
1	Bodenplatte	T11	1.4301		
4	Flanschlager KFL002	T10	Zamak		
4	Lager-Halterung	T09	EN AW-5754 (AlMg3)		
4	Distanzhülse 3	T08	1.4301		
2	Distanzhülse 2	T07	1.4301		
7	Distanzhülse 1	T06	1.4301		
4	Gewindestange M6	T05	Stahl 4.6, verzinkt		L=208mm
2	Gewindestange M6	T04	Stahl 4.6, verzinkt		L=140mm
4	Gewindestange M6	T03	Stahl 4.6, verzinkt		L=132mm
1	Seitenwand 2	T02	EN AW-5754 (AlMg3)		
1	Seitenwand 1	T01	EN AW-5754 (AlMg3)		

Maßstab: 1:5

Allg.-Toleranz: ISO 2768-1 m

Institut für Konstruktionswissenschaften

TU Wien

Benennung:

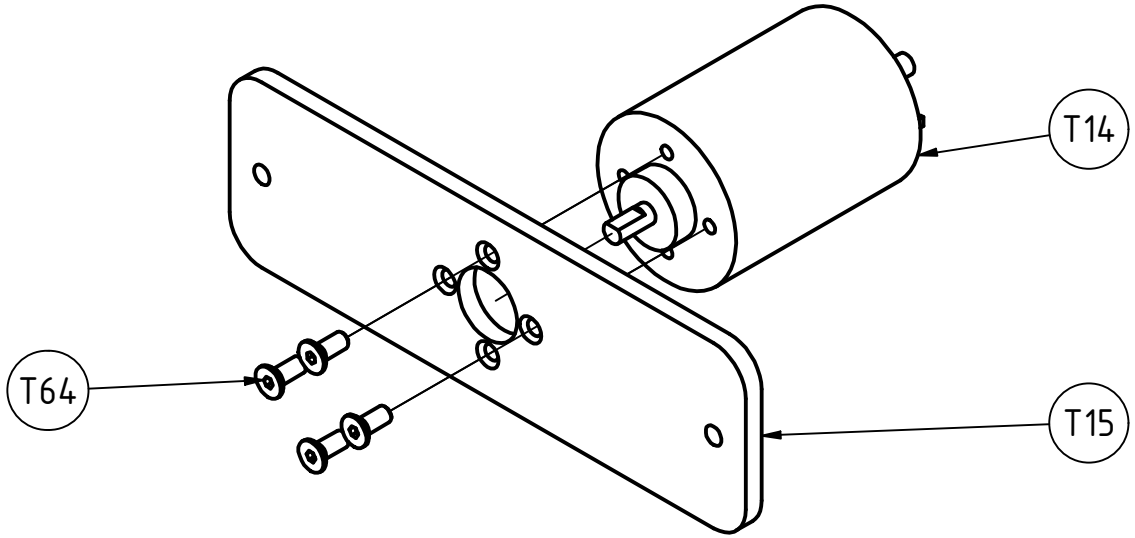
Grundgerüst

Name: Eichberger

Matr.-Nr.: 00909186

Datum: 09.10.2019

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub



Stück	Bezeichnung	Teil	Werkstoff	Norm	Bemerkung
4	Senkkopfschraube M5x12	T64	Stahl	DIN 7991	
1	Motor-Halterung	T15	EN AW-5754 (AlMg3)		
1	Gleichstrommotor	T14	Stahl		

Maßstab: 1:2

Allg.-Toleranz: ISO 2768-1 m

Institut für Konstruktionswissenschaften

TU Wien

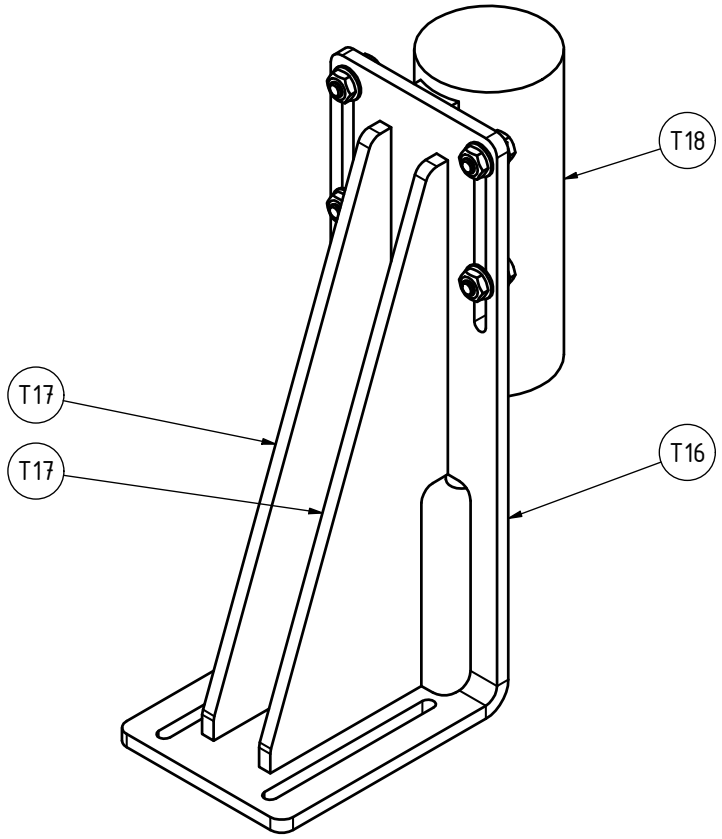
Benennung:

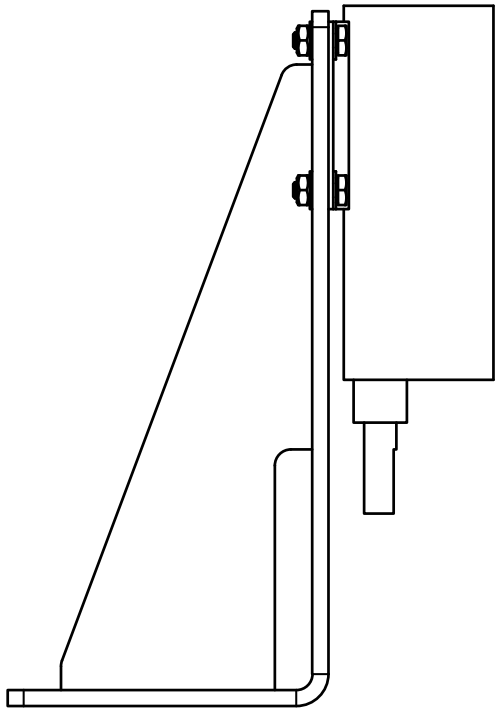
Gleichstrommotor-Einheit

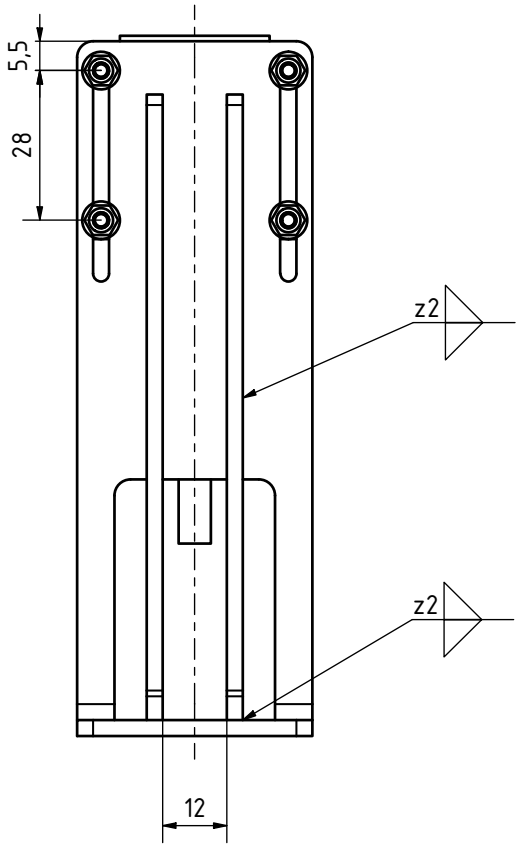
Name: Andreas

Matr.-Nr.: 00909186

Datum: 09.10.2019







Stück	Bezeichnung	Teil	Werkstoff	Norm	Bemerkung
4	Sechskantmutter M3	T60	Stahl	DIN 934	
4	Sechskantschraube M3x8	T49	Stahl	DIN 933	
8	Unterlegscheibe 3,2	T47	Stahl	DIN 125 A	
1	Getriebemotor	T18			
2	Halterung Getriebemotor Teil 2	T17	1.4301		
1	Halterung Getriebemotor Teil 1	T16	1.4301		

Maßstab: 1:1

Allg.-Toleranz: ISO 2768-1 m

Institut für Konstruktionswissenschaften

TU Wien

Benennung:

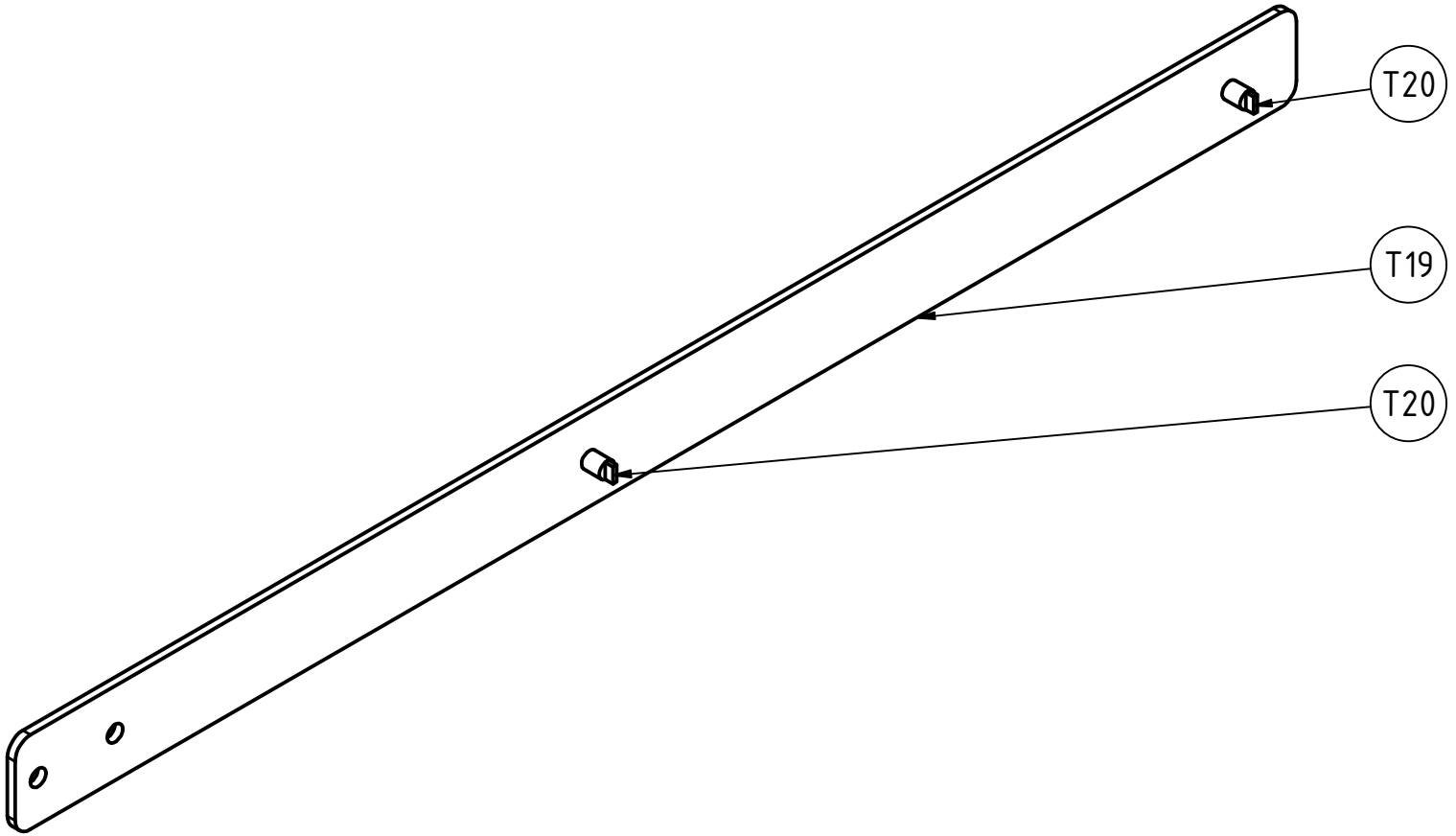
Getriebemotor-Einheit

Name: Eichberger

Matr.-Nr.: 00909186

Datum: 09.10.2019

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub



Stück	Bezeichnung	Teil	Werkstoff	Norm	Bemerkung
2	Laser	T20	Generic		
1	Laserhalterung	T19	EN AW-5754 (AlMg3)		

Maßstab: 1:2

Allg.-Toleranz: ISO 2768-1 m

Institut für Konstruktionswissenschaften

TU Wien

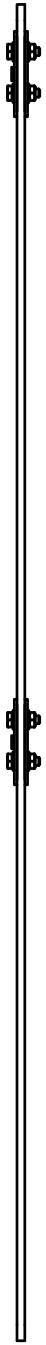
Benennung:

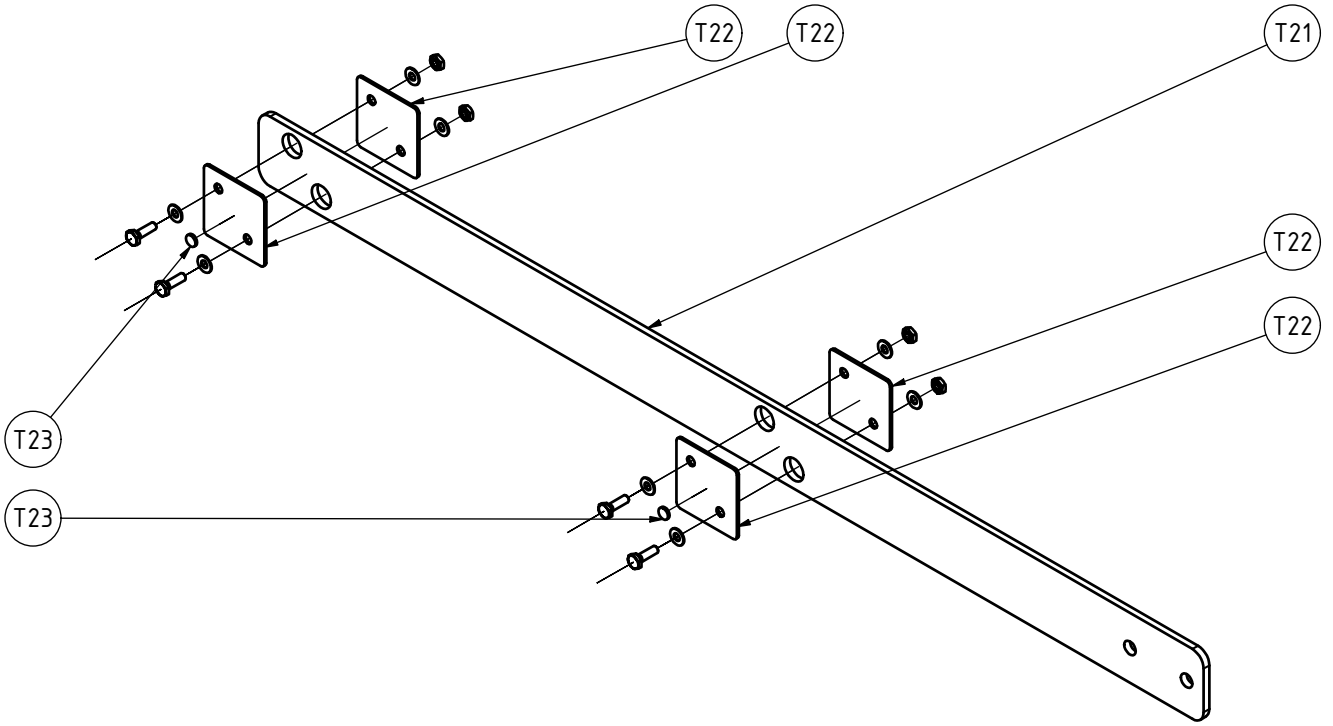
Laserhalterungs-Einheit

Name: Eichberger

Matr.-Nr.: 00909186

Datum: 09.10.2019





Stück	Bezeichnung	Teil	Werkstoff	Norm	Bemerkung
4	Sechskantmutter M3	T60	Stahl	DIN 934	
4	Sechskantschraube M3x10	T50	Stahl	DIN 933	
8	Unterlegscheibe 3,2	T47	Stahl	DIN 125 A	
2	Fotodiode	T23			
4	Fotodioden-Aufnahme	T22	1.4301		
1	Diodenhalterung	T21	EN AW-5754 (AlMg3)		

Maßstab: 1:2

Allg.-Toleranz: ISO 2768-1 m

Institut für Konstruktionswissenschaften

TU Wien

Benennung:

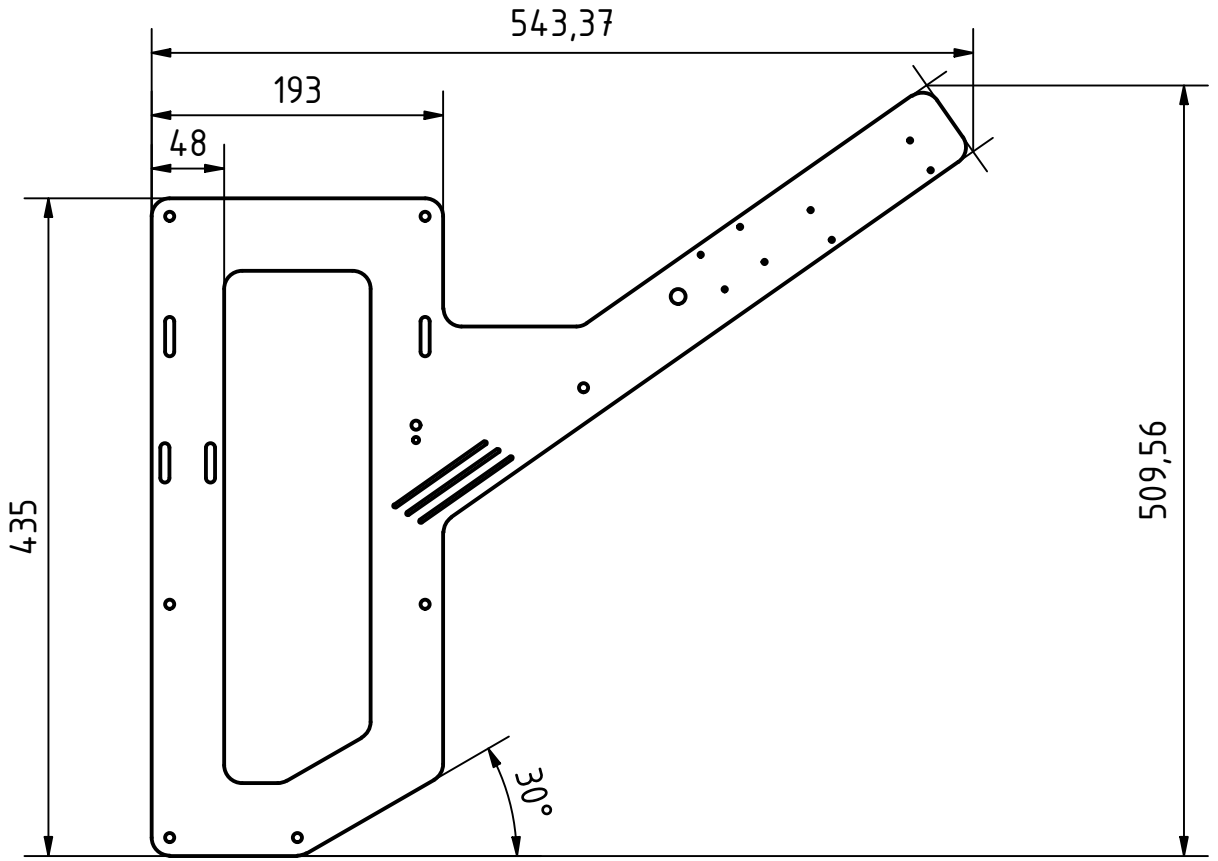
Diodenhalterungs-Einheit

Name: Eichberger

Matr.-Nr.: 00909186

Datum: 09.10.2019

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub



1	Seitenwand 1	T01	EN AW-5754 (AlMg3)		
Stück	Bezeichnung	Teil	Werkstoff	Norm	Bemerkung

Maßstab: 1:5

Allg.-Toleranz: ISO 2768-1 m

Institut für Konstruktionswissenschaften

TU Wien

Benennung:

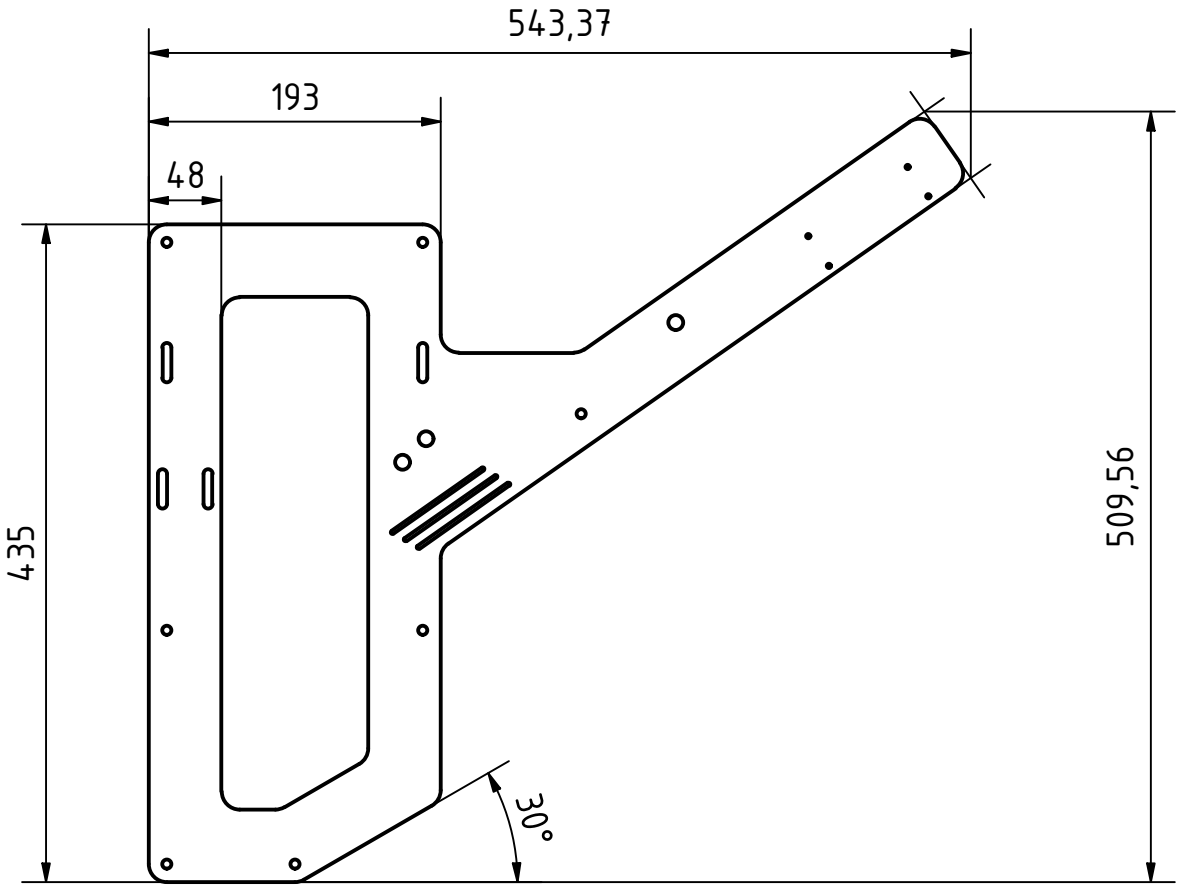
Seitenwand 1

Name: Eichberger

Matr.-Nr.: 00909186

Datum: 09.10.2019

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub



1	Seitenwand 2	T02	EN AW-5754 (AlMg3)		
Stück	Bezeichnung	Teil	Werkstoff	Norm	Bemerkung

Maßstab: 1:5

Allg.-Toleranz: ISO 2768-1 m

Institut für Konstruktionswissenschaften

TU Wien

Benennung:

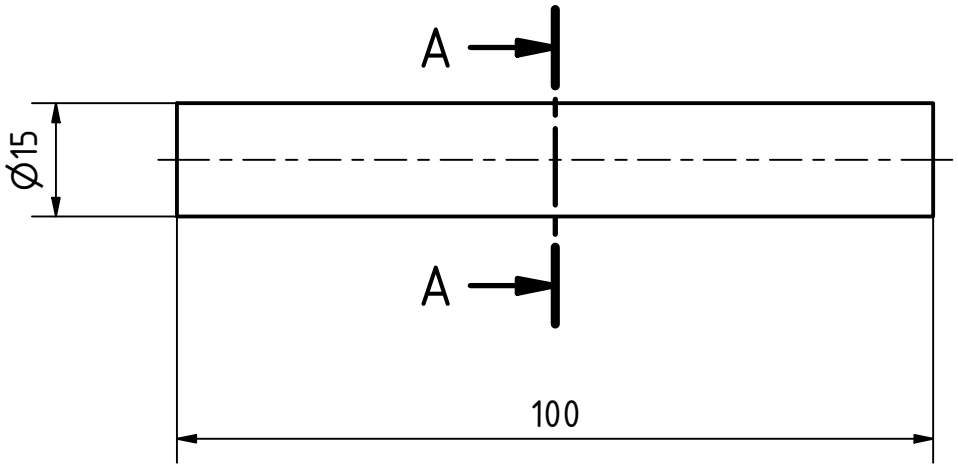
Seitenwand 2

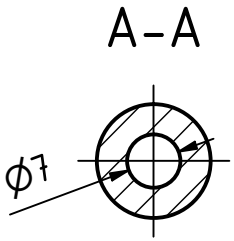
Name: Eichberger

Matr.-Nr.: 00909186

Datum: 09.10.2019

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub





7	Distanzhülse 1	T06	1.4301		
Stück	Bezeichnung	Teil	Werkstoff	Norm	Bemerkung

Maßstab: 1:1

Allg.-Toleranz: ISO 2768-1 m

Institut für Konstruktionswissenschaften

TU Wien

Benennung:

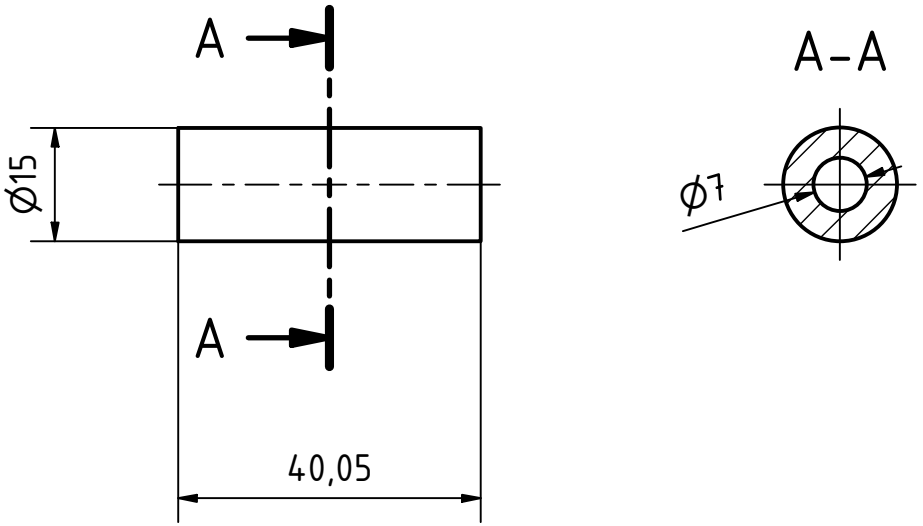
Distanzhülse 1

Name: Eichberger

Matr.-Nr.: 00909186

Datum: 08.10.2019

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub



2	Distanzhülse 2	T07	1.4301		
Stück	Bezeichnung	Teil	Werkstoff	Norm	Bemerkung

Maßstab: 1:1

Allg.-Toleranz: ISO 2768-1 m

Institut für Konstruktionswissenschaften

TU Wien

Benennung:

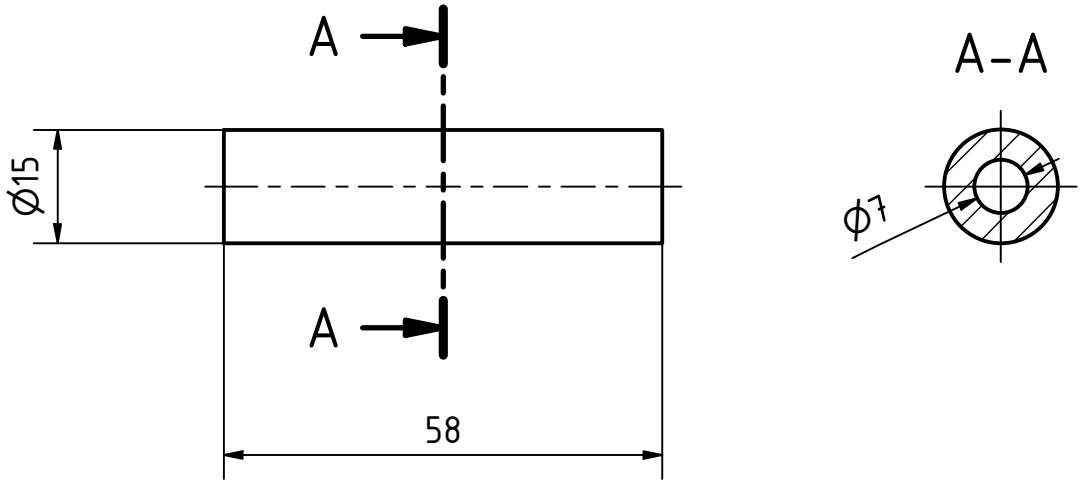
Distanzhülse 2

Name: Eichberger

Matr.-Nr.: 00909186

Datum: 08.10.2019

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub



4	Distanzhülse 3	T08	1.4301		
Stück	Bezeichnung	Teil	Werkstoff	Norm	Bemerkung

Maßstab: 1:1

Allg.-Toleranz: ISO 2768-1 m

Institut für Konstruktionswissenschaften

TU Wien

Benennung:

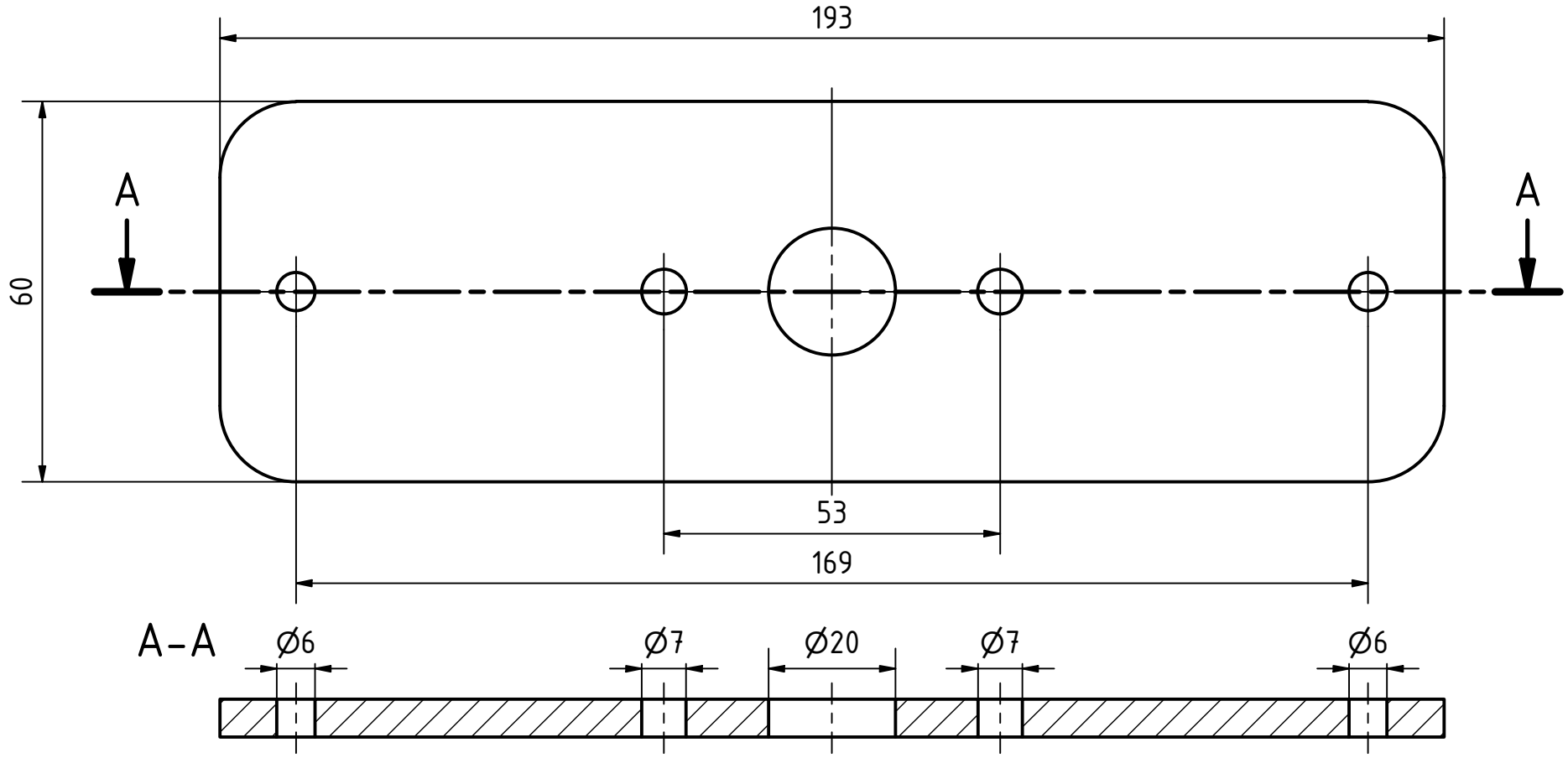
Distanzhülse 3

Name: Eichberger

Matr.-Nr.: 00909186

Datum: 08.10.2019

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub



4	Lager-Halterung	T09	EN AW-5754 (AlMg3)		
Stück	Bezeichnung	Teil	Werkstoff	Norm	Bemerkung

Maßstab: 1:1

Allg.-Toleranz: ISO 2768-1 m

Institut für Konstruktionswissenschaften

TU Wien

Benennung:

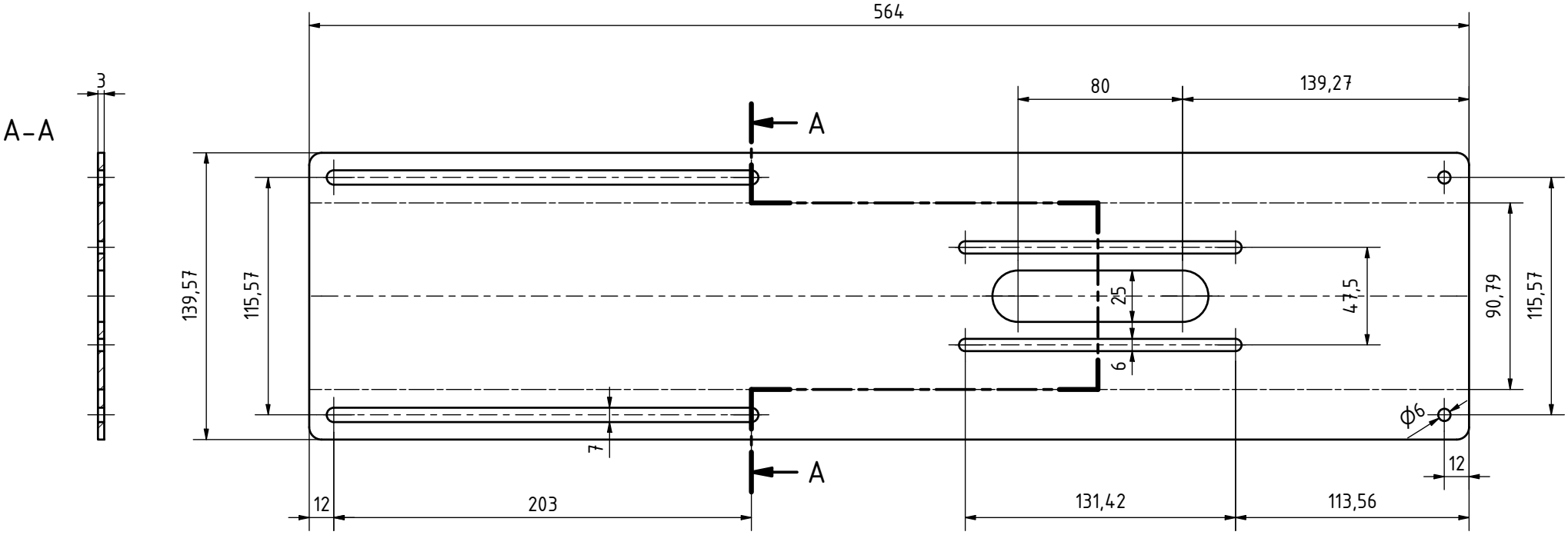
Lager-Halterung

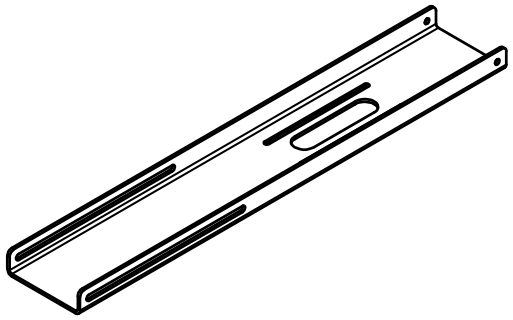
Name: Eichberger

Matr.-Nr.: 00909186

Datum: 09.10.2019

Alle nicht bemaßten Radien: R12



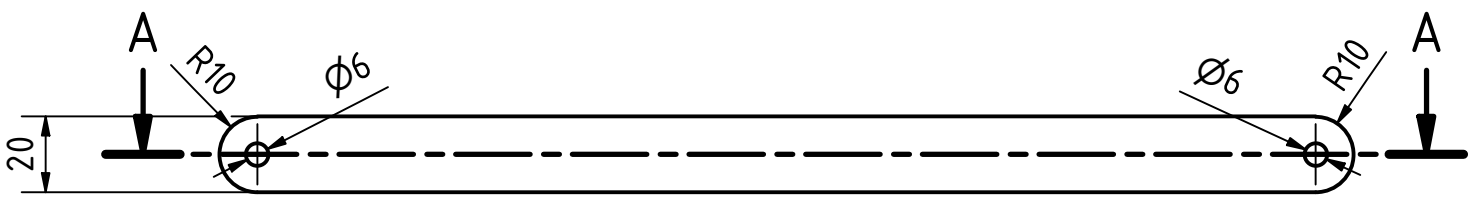


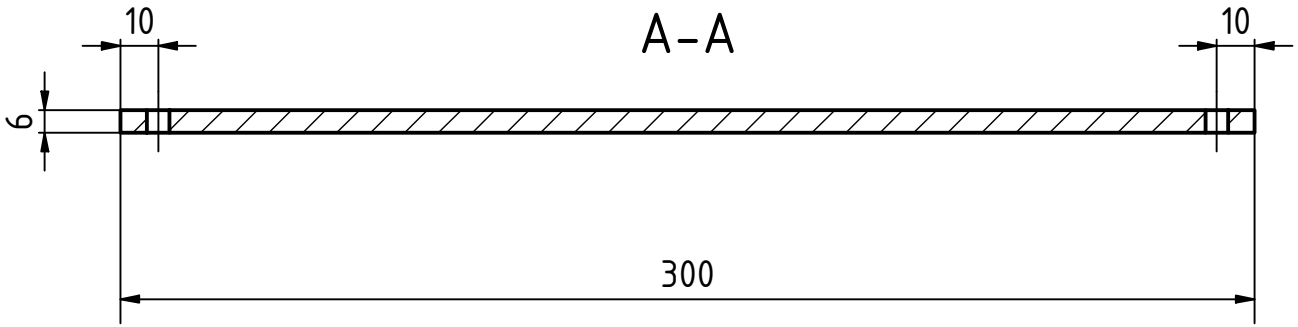
Alle nicht bemaßten Radien: R6

1	Bodenplatte	T11	1.4301		
Stück	Bezeichnung	Teil	Werkstoff	Norm	Bemerkung

Maßstab: 1:2	Institut für Konstruktionswissenschaften	TU Wien
Allg.-Toleranz: ISO 2768-1 m	Benennung: **Bodenplatte**	Name: Eichberger Matr.-Nr.: 00909186 Datum: 09.10.2019

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub

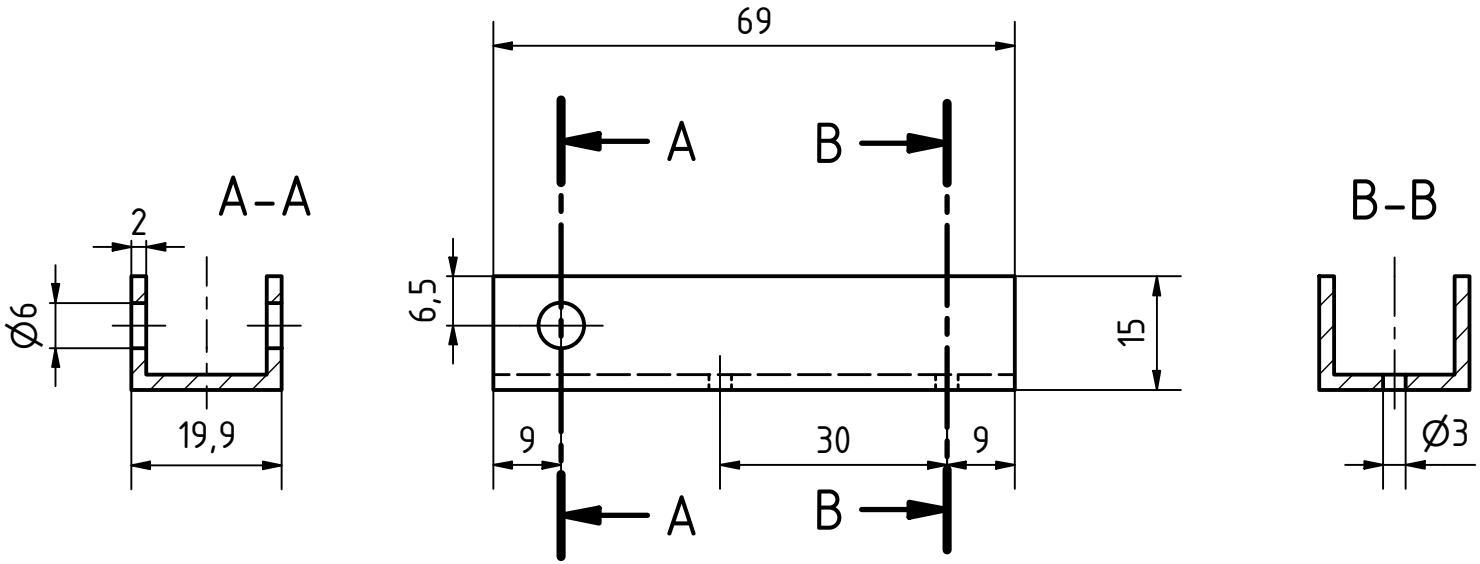




2	Stange	T12	1.4301		
Stück	Bezeichnung	Teil	Werkstoff	Norm	Bemerkung

Maßstab: 1:2	Institut für Konstruktionswissenschaften	TU Wien
Allg.-Toleranz: ISO 2768-1 m	Benennung: **Stange**	Name: Eichberger Matr.-Nr.: 00909186 Datum: 09.10.2019

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub



1	U-Profil	T13	Aluminium		
Stück	Bezeichnung	Teil	Werkstoff	Norm	Bemerkung

Maßstab: 1:1

Allg.-Toleranz: ISO 2768-1 m

Institut für Konstruktionswissenschaften

TU Wien

Benennung:

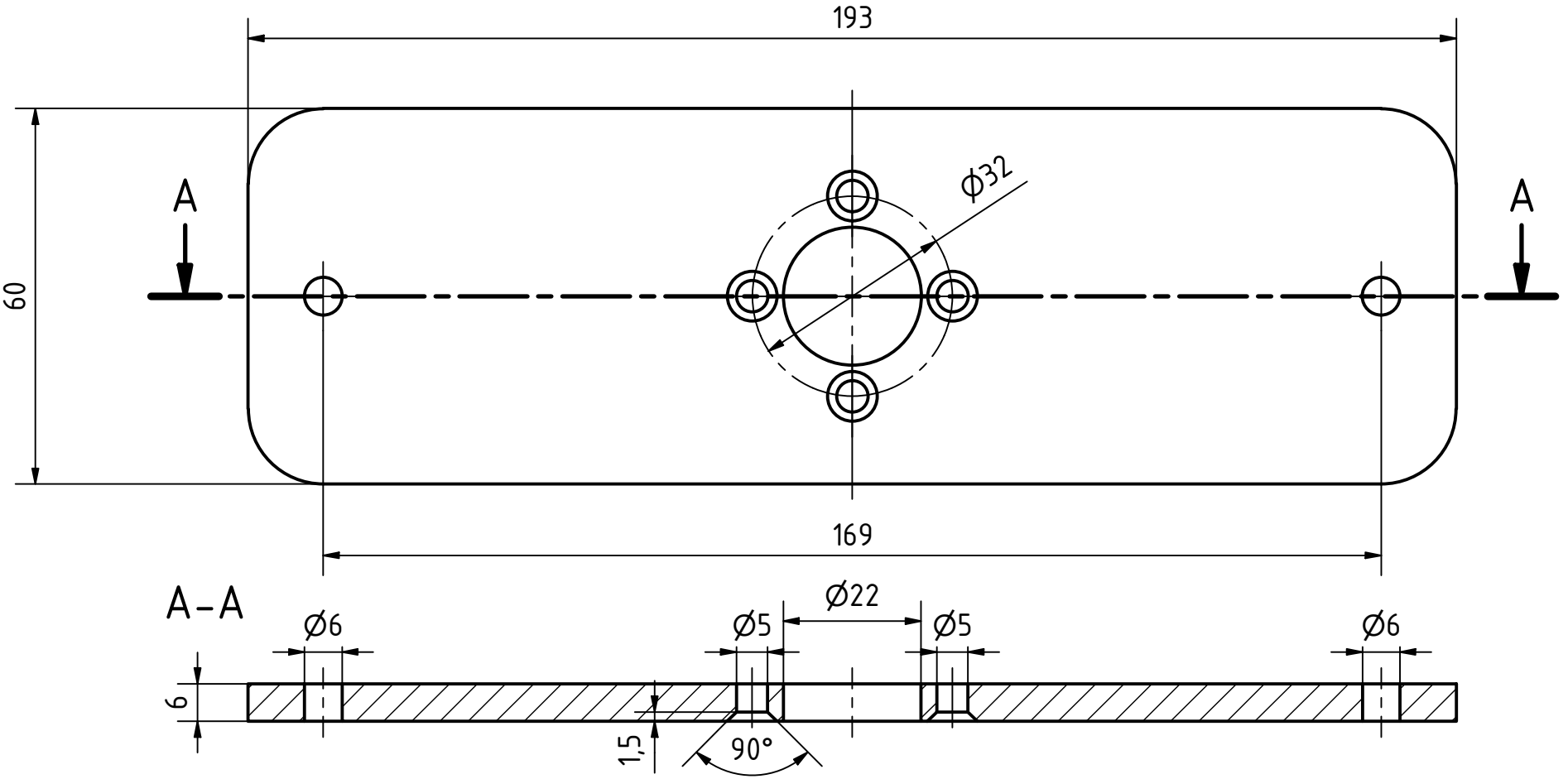
U-Profil

Name: Eichberger

Matr.-Nr.: 00909186

Datum: 09.10.2019

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub



2	Motor-Halterung	T15	EN AW-5754 (AlMg3)		
Stück	Bezeichnung	Teil	Werkstoff	Norm	Bemerkung

Maßstab: 1:1

Allg.-Toleranz: ISO 2768-1 m

Institut für Konstruktionswissenschaften

TU Wien

Benennung:

Motor-Halterung

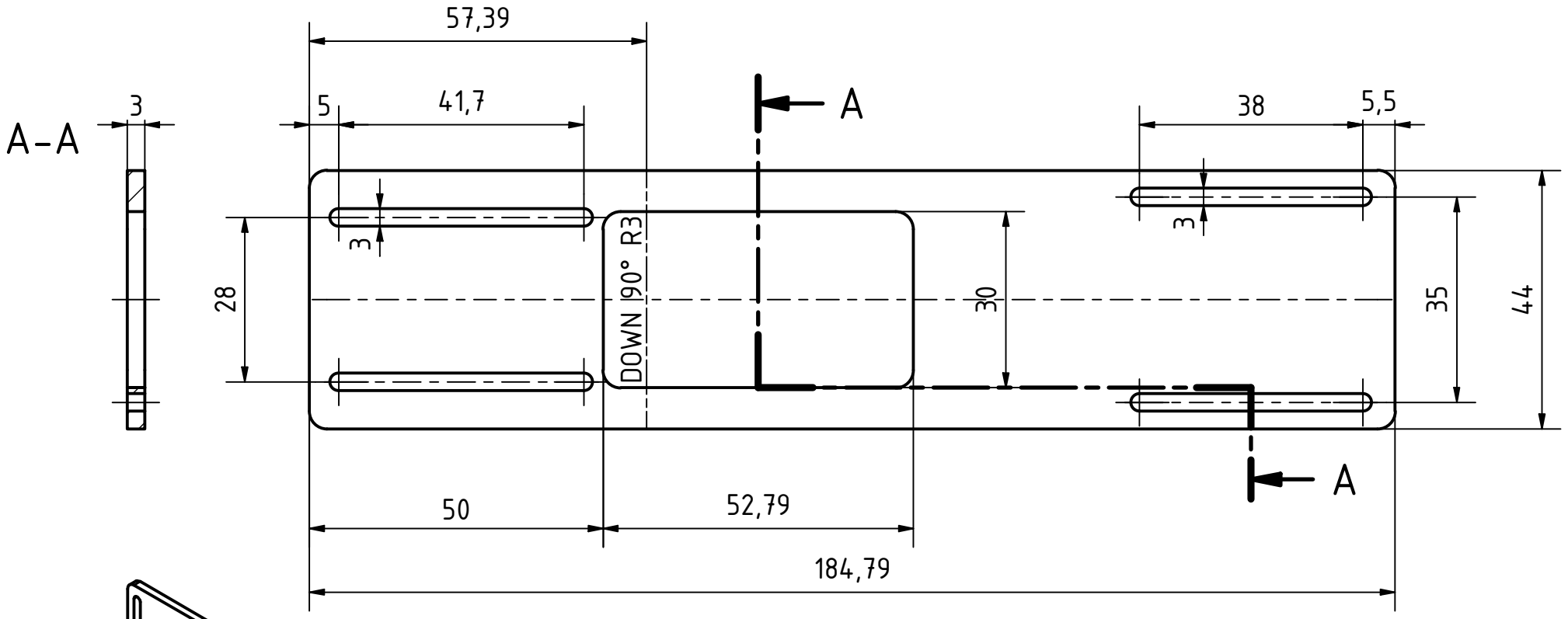
Name: Eichberger

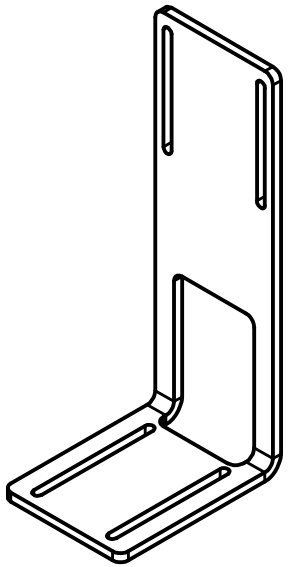
Matr.-Nr.: 00909186

Datum: 09.10.2019

Alle nicht bemaßten Radien: R12

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub





Alle nicht bemaßten Radien: R3

1	Halterung Getriebemotor Teil 1	T16	1.4301		
Stück	Bezeichnung	Teil	Werkstoff	Norm	Bemerkung

Maßstab: 1:1

Allg.-Toleranz: ISO 2768-1 m

Institut für Konstruktionswissenschaften

TU Wien

Benennung:

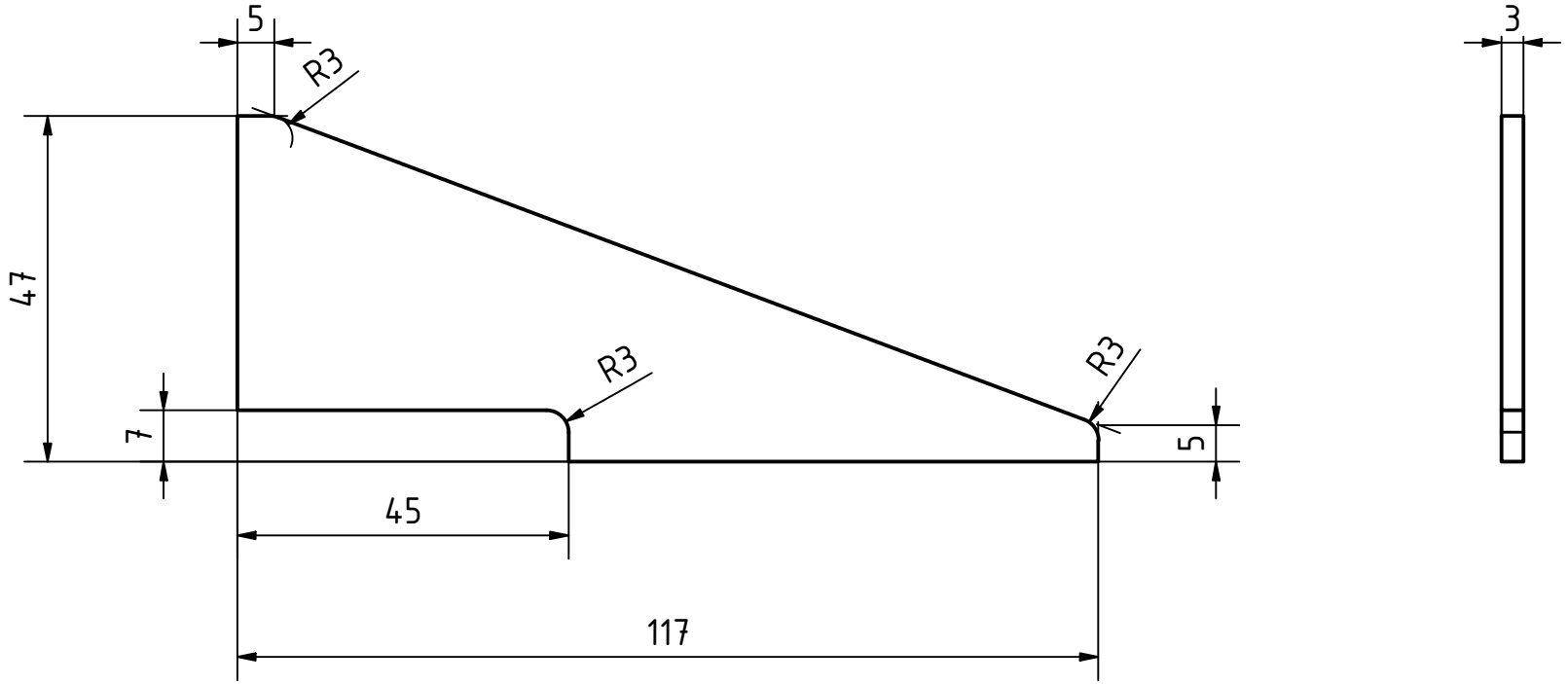
Halterung Getriebemotor Teil 1

Name: Eichberger

Matr.-Nr.: 00909186

Datum: 09.10.2019

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub



2	Halterung Getriebemotor Teil 2	T17	1.4301		
Stück	Bezeichnung	Teil	Werkstoff	Norm	Bemerkung

Maßstab: 1:1

Allg.-Toleranz: ISO 2768-1 m

Institut für Konstruktionswissenschaften

TU Wien

Benennung:

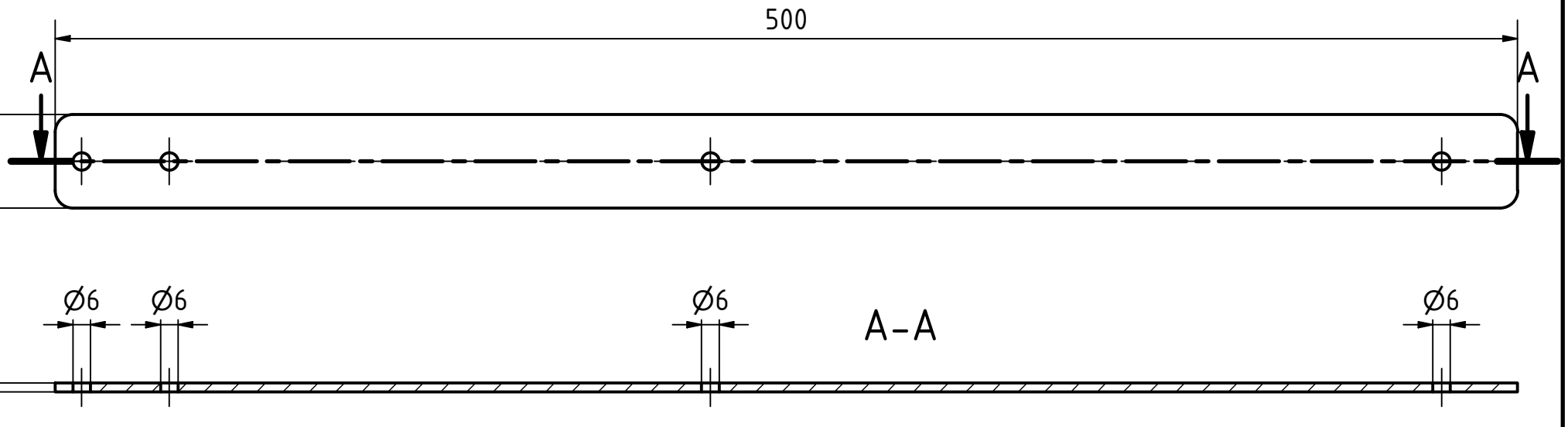
Halterung Getriebemotor Teil 2

Name: Eichberger

Matr.-Nr.: 00909186

Datum: 09.10.2019

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub



Alle nicht bemaßten Radien: R6

Stück	Bezeichnung	Teil	Werkstoff	Norm	Bemerkung
1	Laserhalterung	T19	EN AW-5754 (AlMg3)		

Maßstab: 1:2

Allg.-Toleranz: ISO 2768-1 m

Institut für Konstruktionswissenschaften

TU Wien

Benennung:

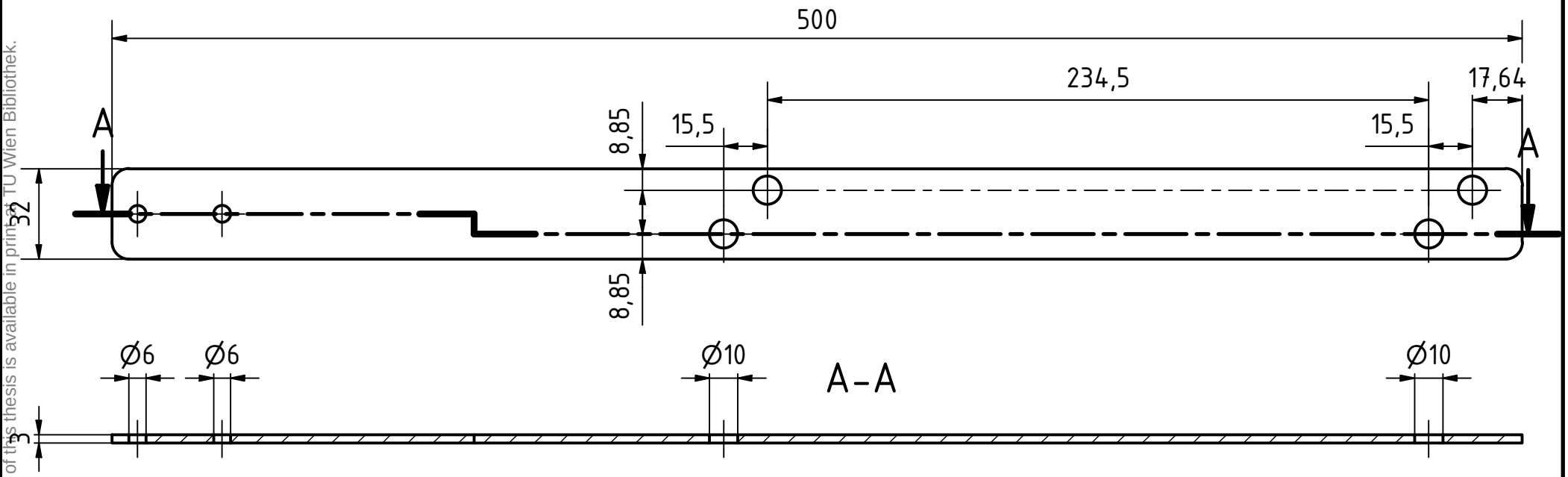
Laserhalterung

Name: Eichberger

Matr.-Nr.: 00909186

Datum: 09.10.2019

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.



Alle nicht bemaßten Radien: R6

1	Diodenhalterung	T21	EN AW-5754 (AlMg3)		
Stück	Bezeichnung	Teil	Werkstoff	Norm	Bemerkung

Maßstab: 1:2

Allg.-Toleranz: ISO 2768-1 m

Institut für Konstruktionswissenschaften

TU Wien

Benennung:

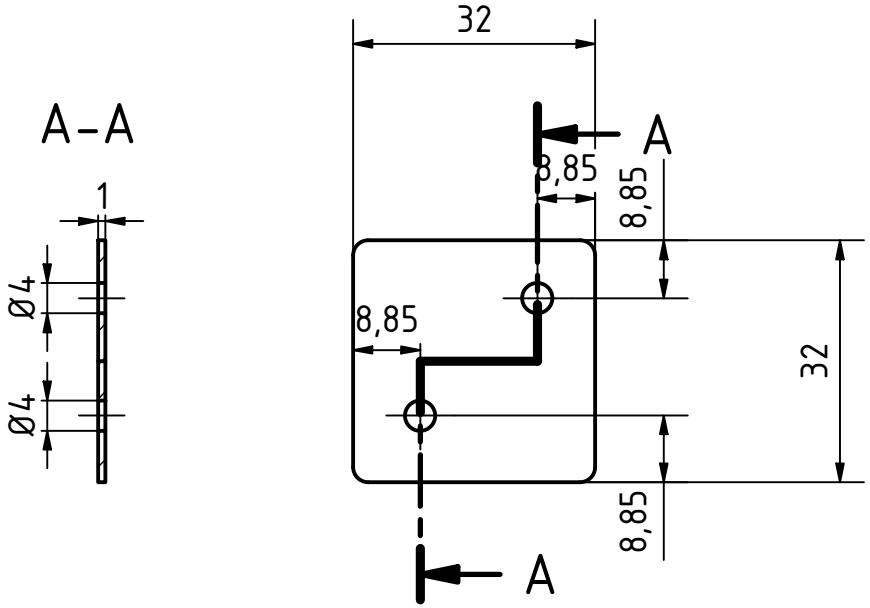
Diodenhalterung

Name: Eichberger

Matr.-Nr.: 00909186

Datum: 09.10.2019

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub



Alle nicht bemaßten Radien: R2

Stück	Bezeichnung	Teil	Werkstoff	Norm	Bemerkung
3	Fotodioden-Aufnahme	T22	1.4301		

Maßstab: 1:1

Allg.-Toleranz: ISO 2768-1 m

Institut für Konstruktionswissenschaften

TU Wien

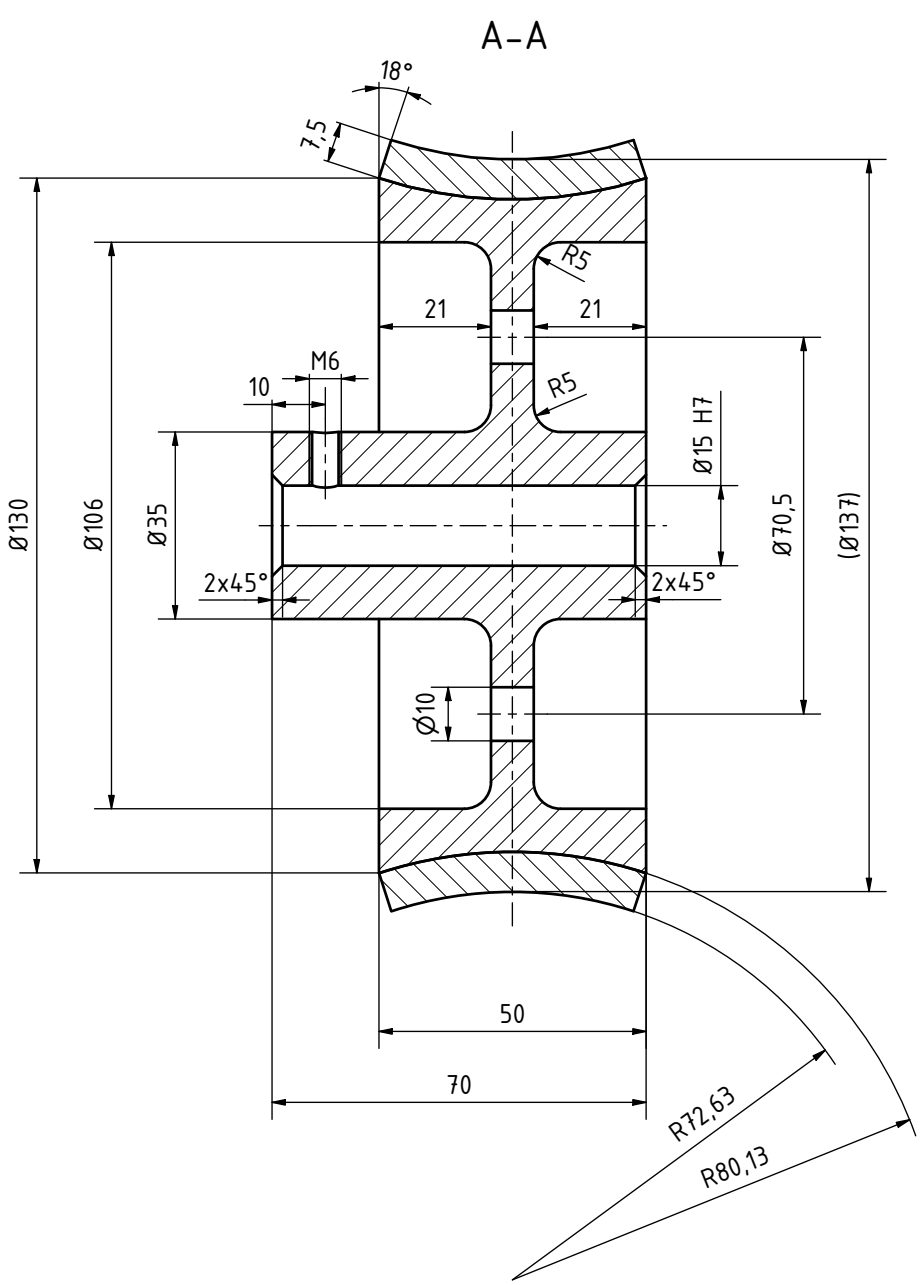
Benennung:

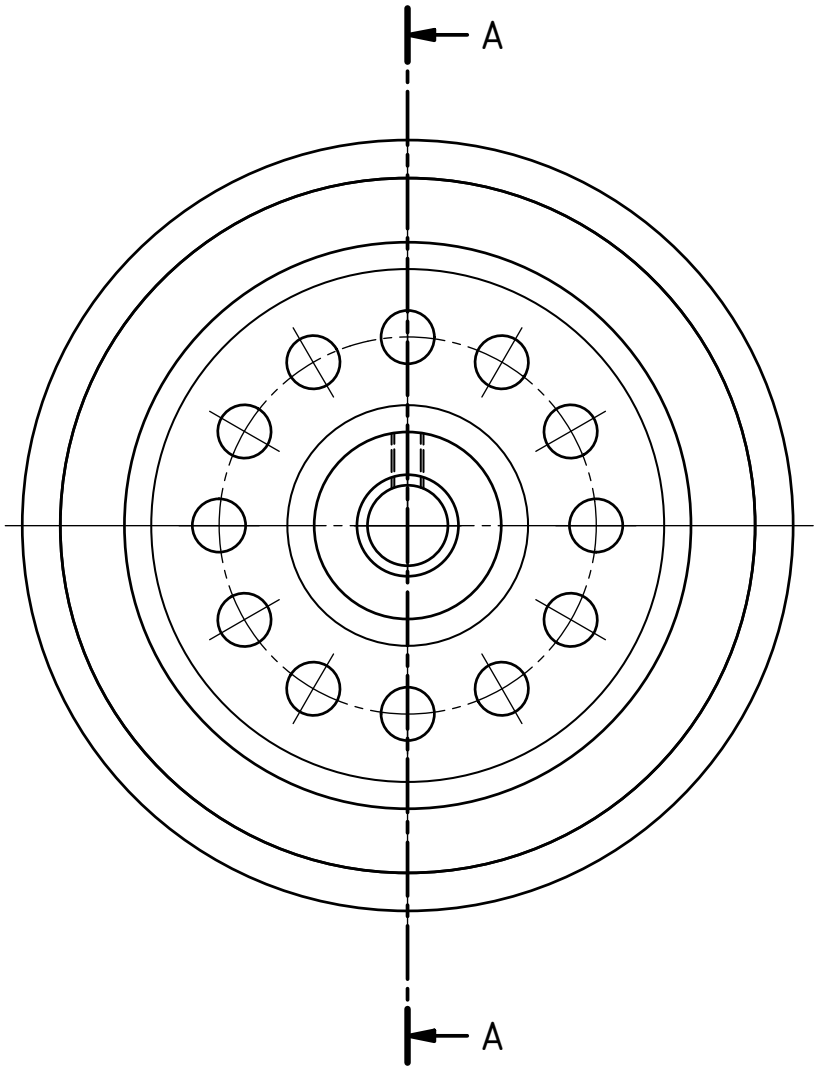
Fotodioden-Aufnahme

Name: Eichberger

Matr.-Nr.: 00909186

Datum: 08.10.2019

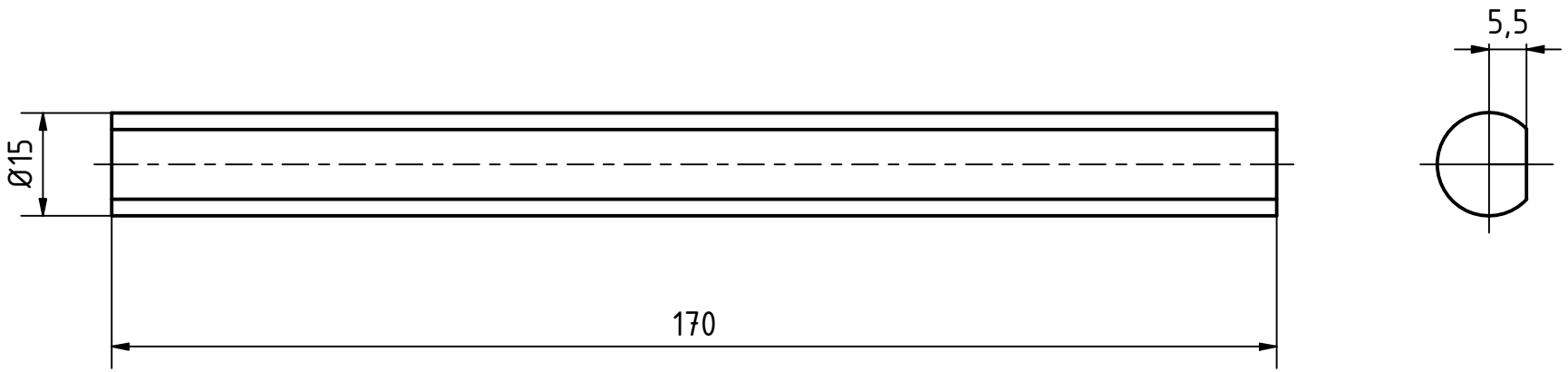




2	Schwungrad	T24			
Stück	Bezeichnung	Teil	Werkstoff	Norm	Bemerkung

Maßstab: 1:1	Institut für Konstruktionswissenschaften	TU Wien
Allg.-Toleranz: ISO 2768-1 m	Benennung: **Schwungrad**	Name: Eichberger Matr.-Nr.: 00909186 Datum: 09.10.2019

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub



2	Schwungrad-Welle	T25	S235JR		
Stück	Bezeichnung	Teil	Werkstoff	Norm	Bemerkung

Maßstab: 1:1

Allg.-Toleranz: ISO 2768-1 m

Institut für Konstruktionswissenschaften

TU Wien

Benennung:

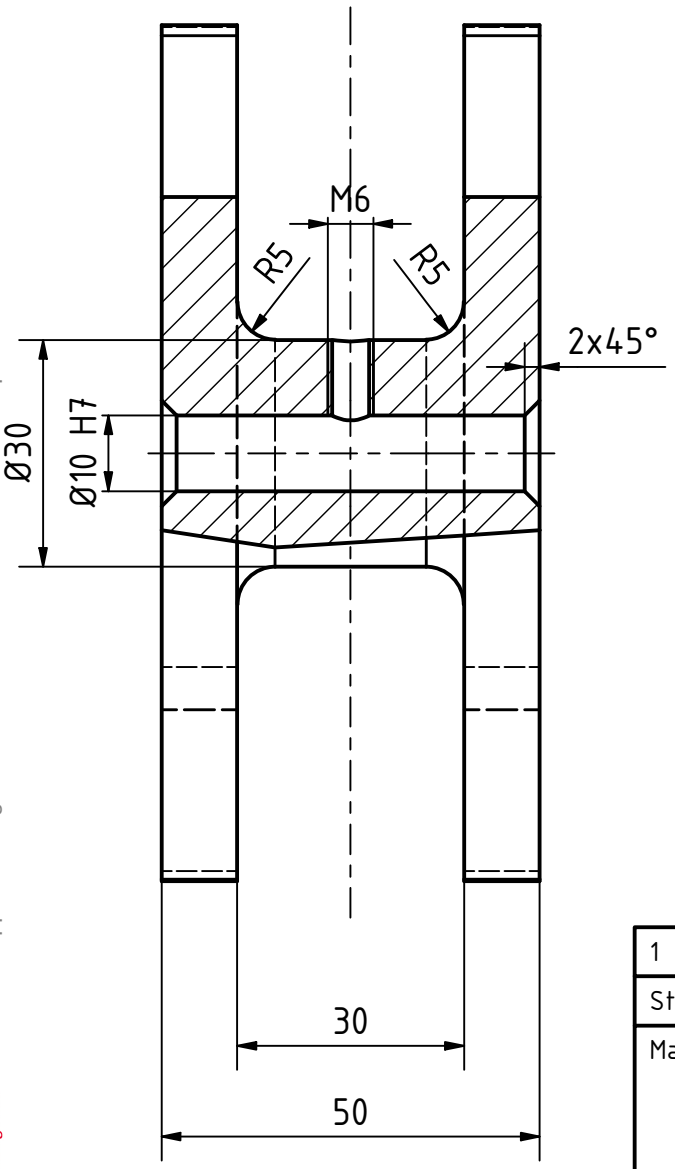
Schwungrad-Welle

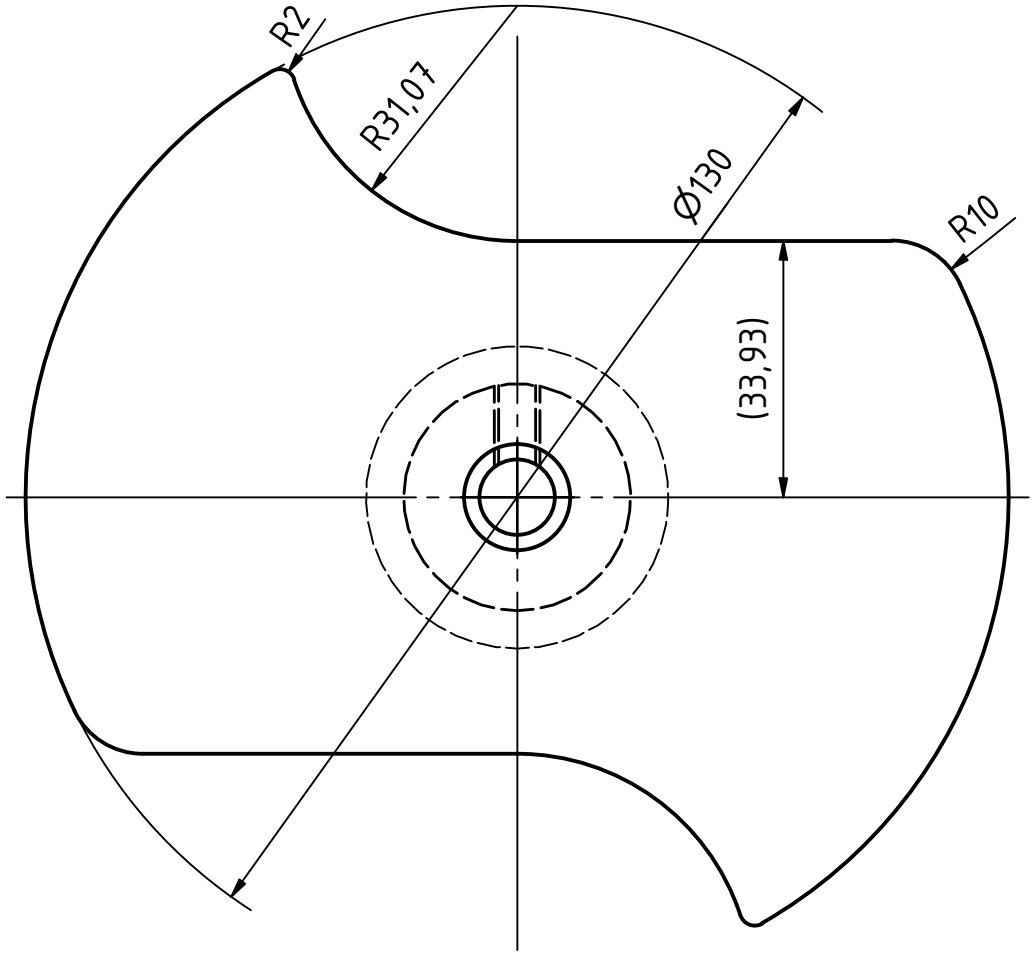
Name: Eichberger

Matr.-Nr.: 00909186

Datum: 08.10.2019

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub





1	Ballrad	T27	EN AW-5754 (AlMg3)		
Stück	Bezeichnung	Teil	Werkstoff	Norm	Bemerkung

Maßstab: 1:1

Allg.-Toleranz: ISO 2768-1 m

Institut für Konstruktionswissenschaften

TU Wien

Benennung:

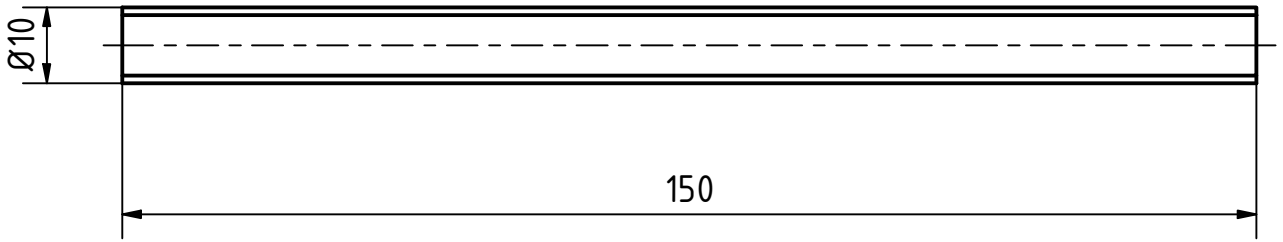
Ballrad

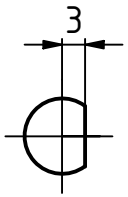
Name: Eichberger

Matr.-Nr.: 00909186

Datum: 08.10.2019

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub





1	Ballrad-Welle	T28	S235JR		
Stück	Bezeichnung	Teil	Werkstoff	Norm	Bemerkung

Maßstab: 1:1

Allg.-Toleranz: ISO 2768-1 m

Institut für Konstruktionswissenschaften

TU Wien

Benennung:

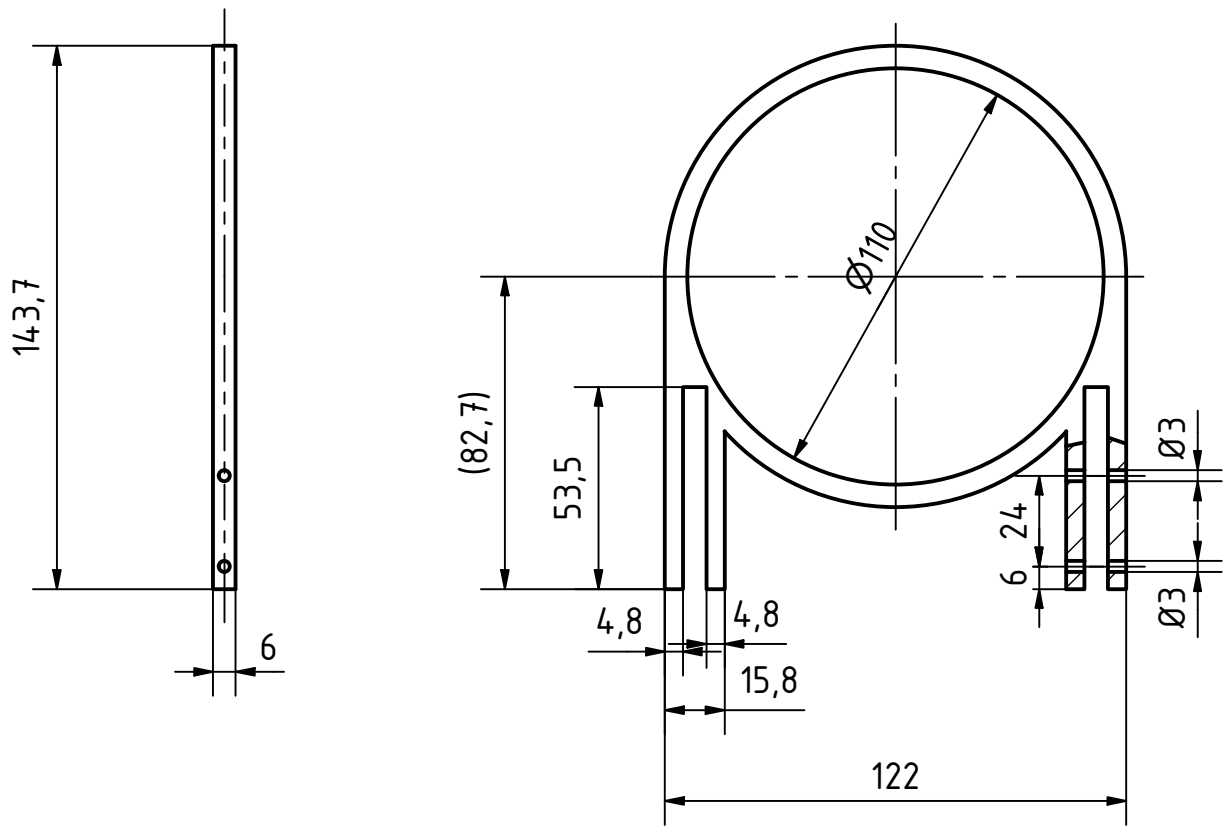
Ballrad-Welle

Name: Eichberger

Matr.-Nr.: 00909186

Datum: 08.10.2019

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub



2	Halterung Rohr 2	T30	EN AW-5754 (AlMg3)		
Stück	Bezeichnung	Teil	Werkstoff	Norm	Bemerkung

Maßstab: 1:2

Allg.-Toleranz: ISO 2768-1 m

Institut für Konstruktionswissenschaften

TU Wien

Benennung:

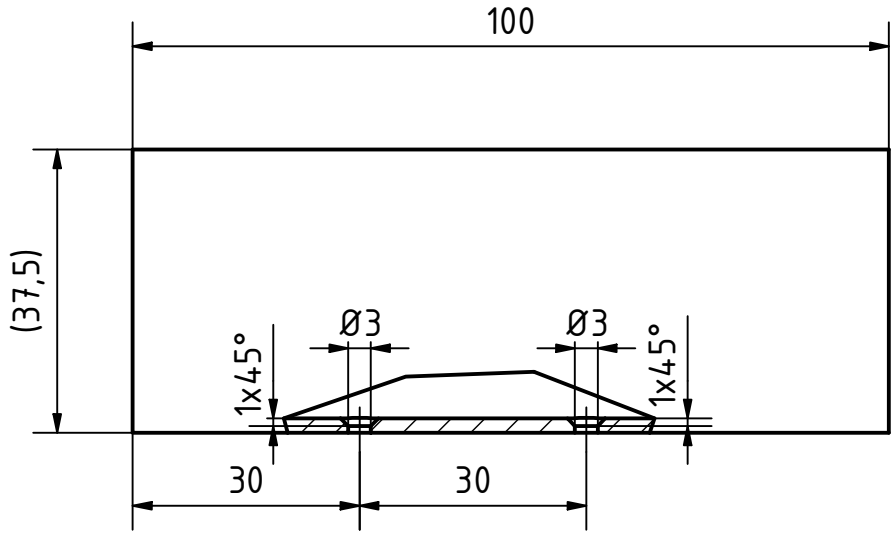
Halterung Rohr 2

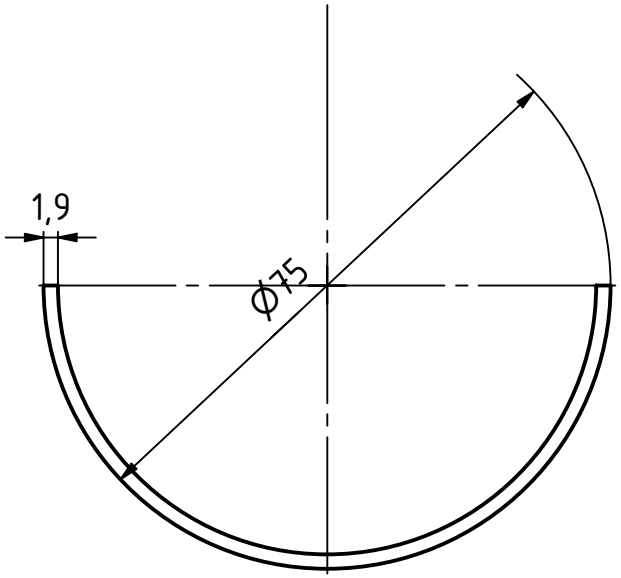
Name: Eichberger

Matr.-Nr.: 00909186

Datum: 08.10.2019

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub





1	Rohr 1	T31	Polypropylen		
Stück	Bezeichnung	Teil	Werkstoff	Norm	Bemerkung

Maßstab: 1:1

Allg.-Toleranz: ISO 2768-1 m

Institut für Konstruktionswissenschaften

TU Wien

Benennung:

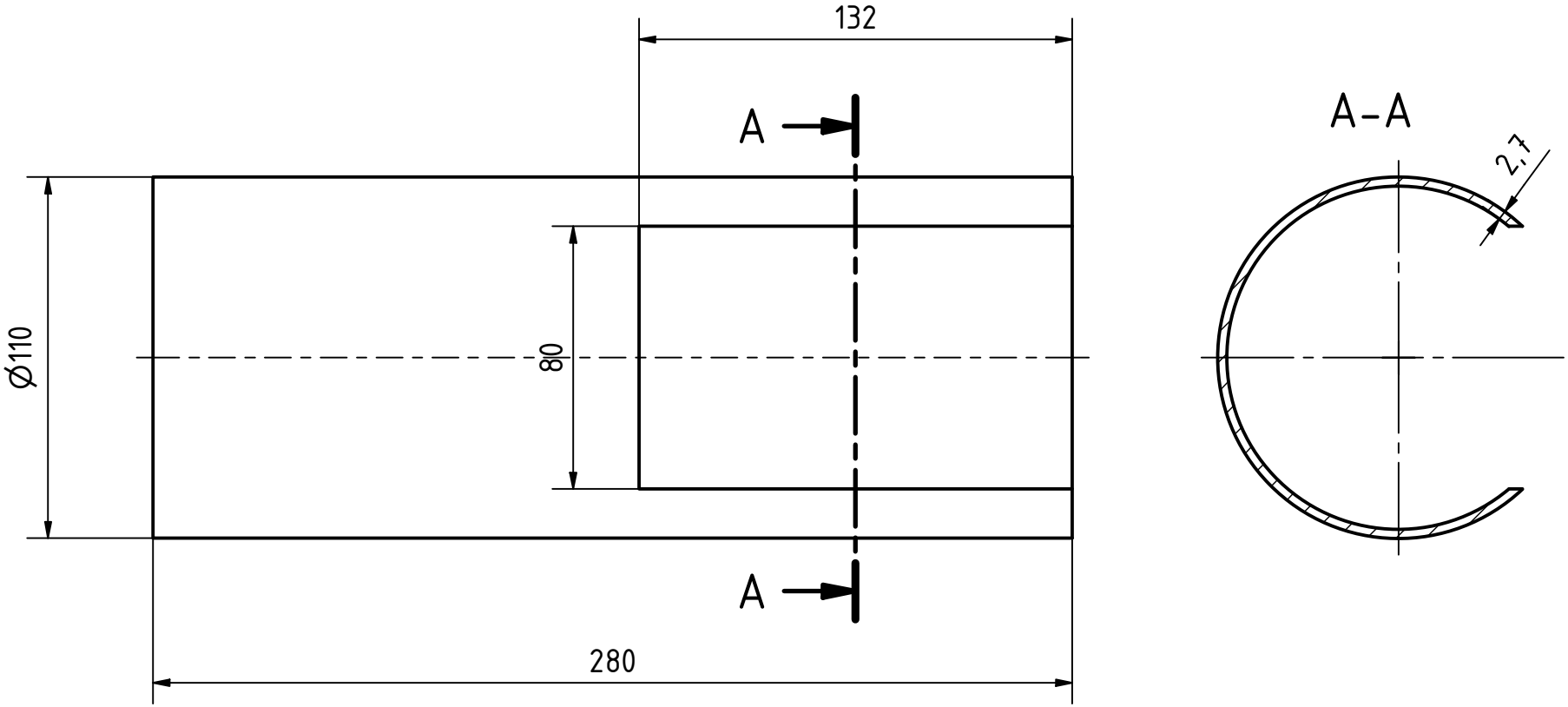
Rohr 1

Name: Eichberger

Matr.-Nr.: 00909186

Datum: 08.10.2019

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub



1	Rohr 2	T32	Polypropylen		
Stück	Bezeichnung	Teil	Werkstoff	Norm	Bemerkung

Maßstab: 1:2

Allg.-Toleranz: ISO 2768-1 m

Institut für Konstruktionswissenschaften

TU Wien

Benennung:

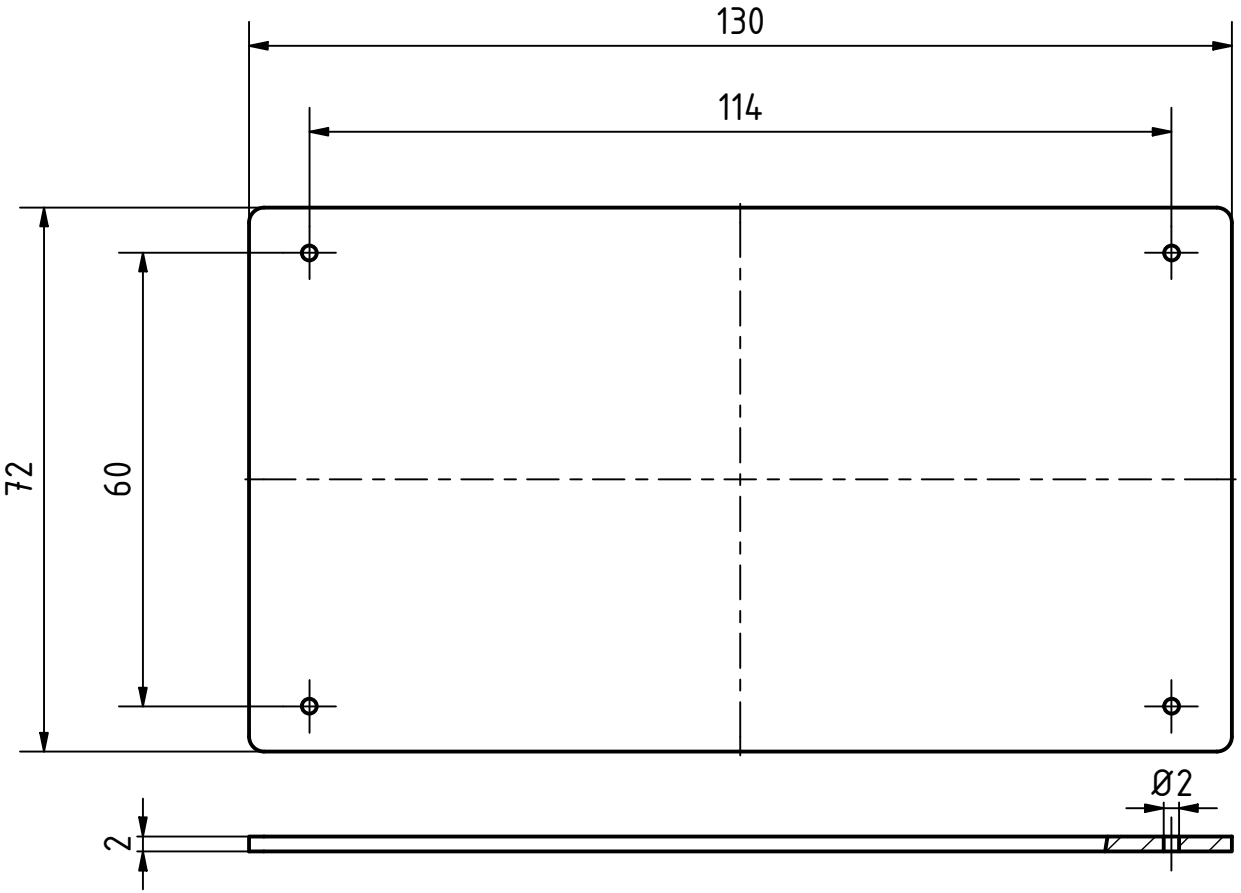
Rohr 2

Name: Eichberger

Matr.-Nr.: 00909186

Datum: 08.10.2019

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub



Alle nicht bemaßten Radien: R2

Stück	Bezeichnung	Teil	Werkstoff	Norm	Bemerkung
1	Einschubblech	T33	1.4301		

Maßstab: 1:1

Allg.-Toleranz: ISO 2768-1 m

Institut für Konstruktionswissenschaften

TU Wien

Benennung:

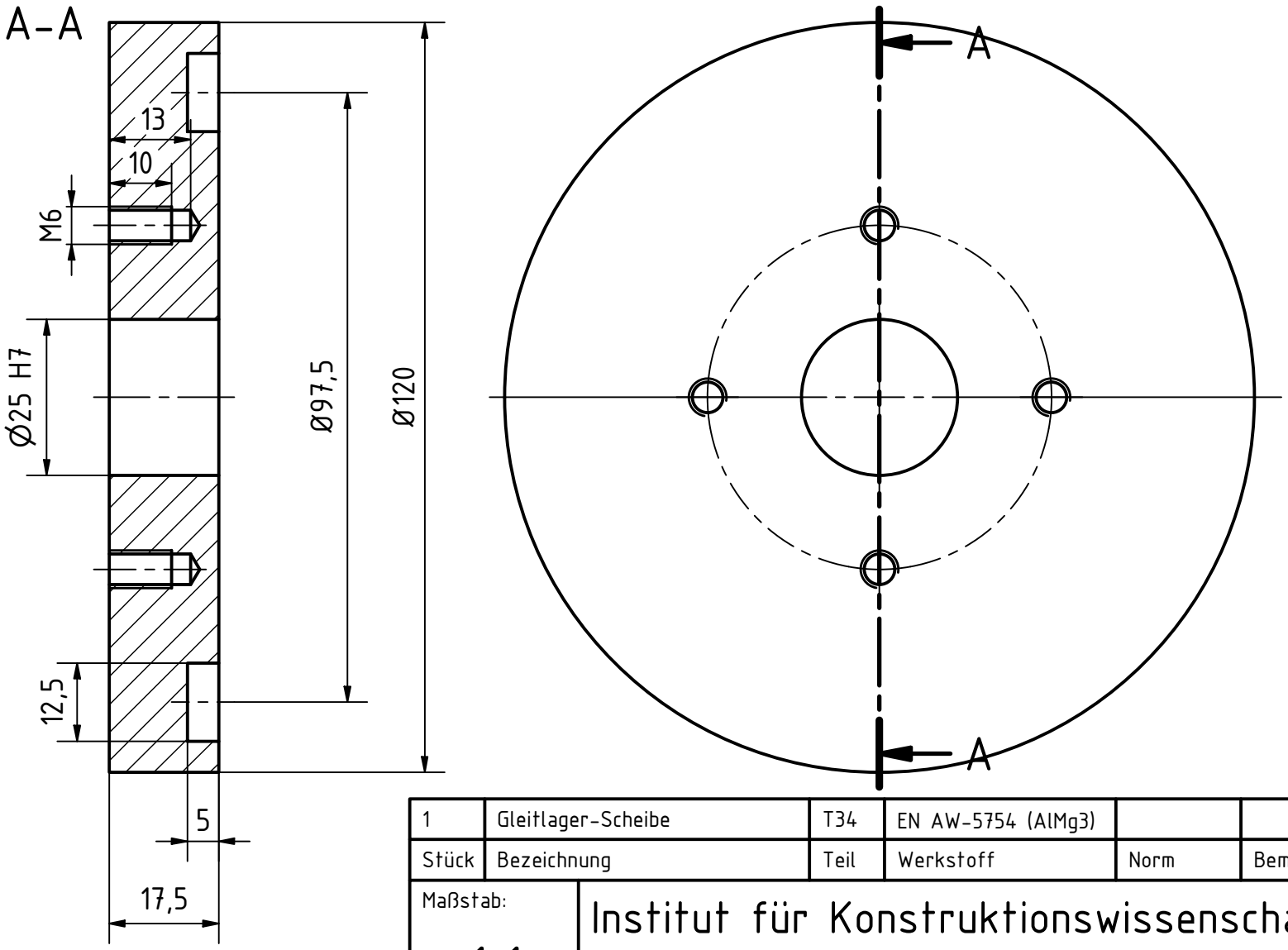
Einschubblech

Name: Eichberger

Matr.-Nr.: 00909186

Datum: 08.10.2019

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub



1	Gleitlager-Scheibe	T34	EN AW-5754 (AlMg3)		
Stück	Bezeichnung	Teil	Werkstoff	Norm	Bemerkung

Maßstab: 1:1

Allg.-Toleranz: ISO 2768-1 m

Institut für Konstruktionswissenschaften

TU Wien

Benennung:

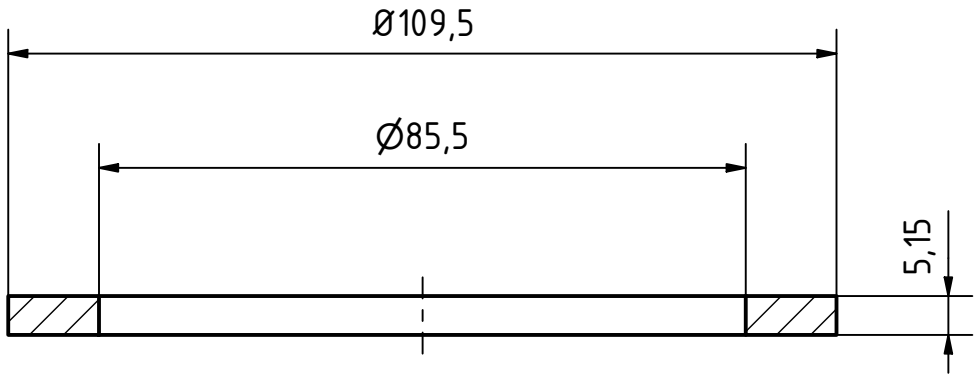
Gleitlager-Scheibe

Name: Eichberger

Matr.-Nr.: 00909186

Datum: 07.10.2019

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub



1	Gleitscheibe	T35	Bronze		
Stück	Bezeichnung	Teil	Werkstoff	Norm	Bemerkung

Maßstab: 1:1

Allg.-Toleranz: ISO 2768-1 m

Institut für Konstruktionswissenschaften

TU Wien

Benennung:

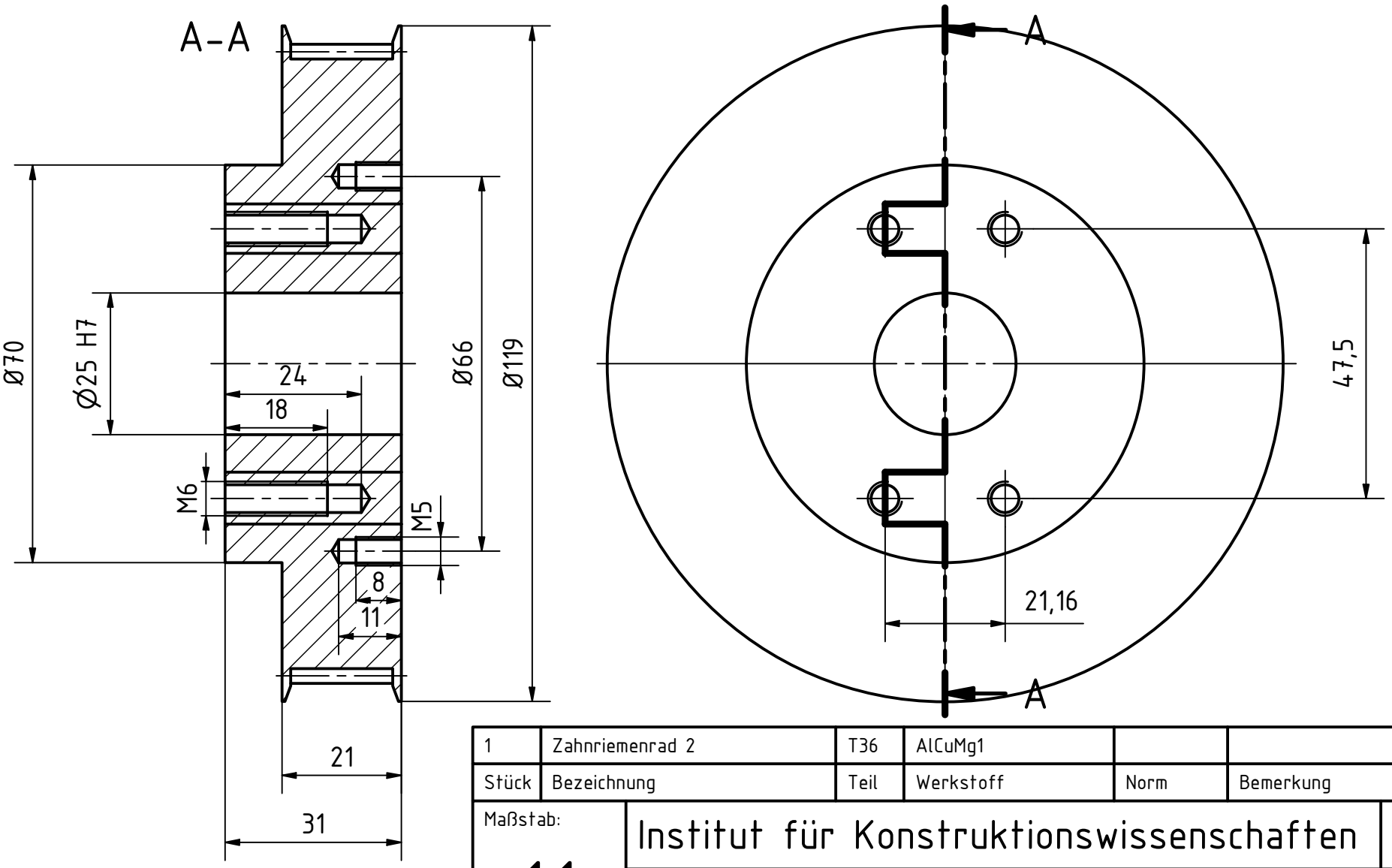
Gleitscheibe

Name: Eichberger

Matr.-Nr.: 00909186

Datum: 07.10.2019

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub



1	Zahnriemenrad 2	T36	AlCuMg1		
Stück	Bezeichnung	Teil	Werkstoff	Norm	Bemerkung

Maßstab: 1:1

Allg.-Toleranz: ISO 2768-1 m

Institut für Konstruktionswissenschaften

TU Wien

Benennung:

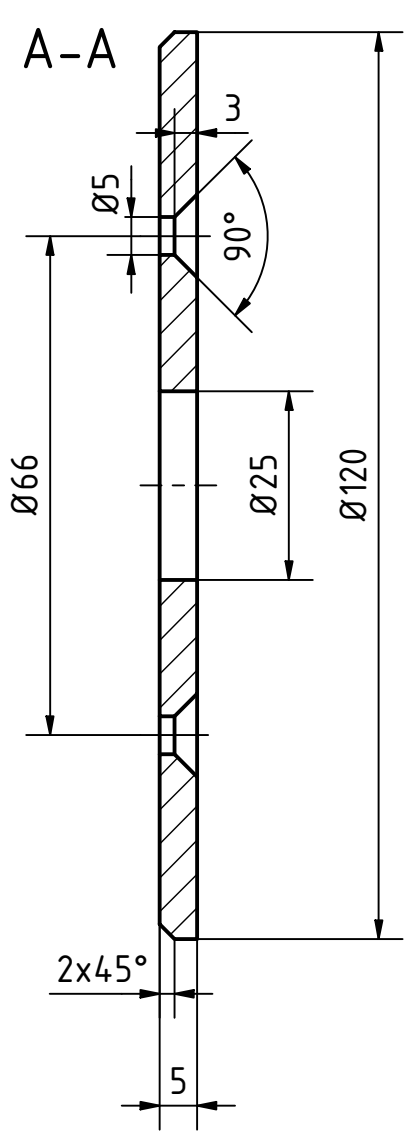
Zahnriemenrad 2

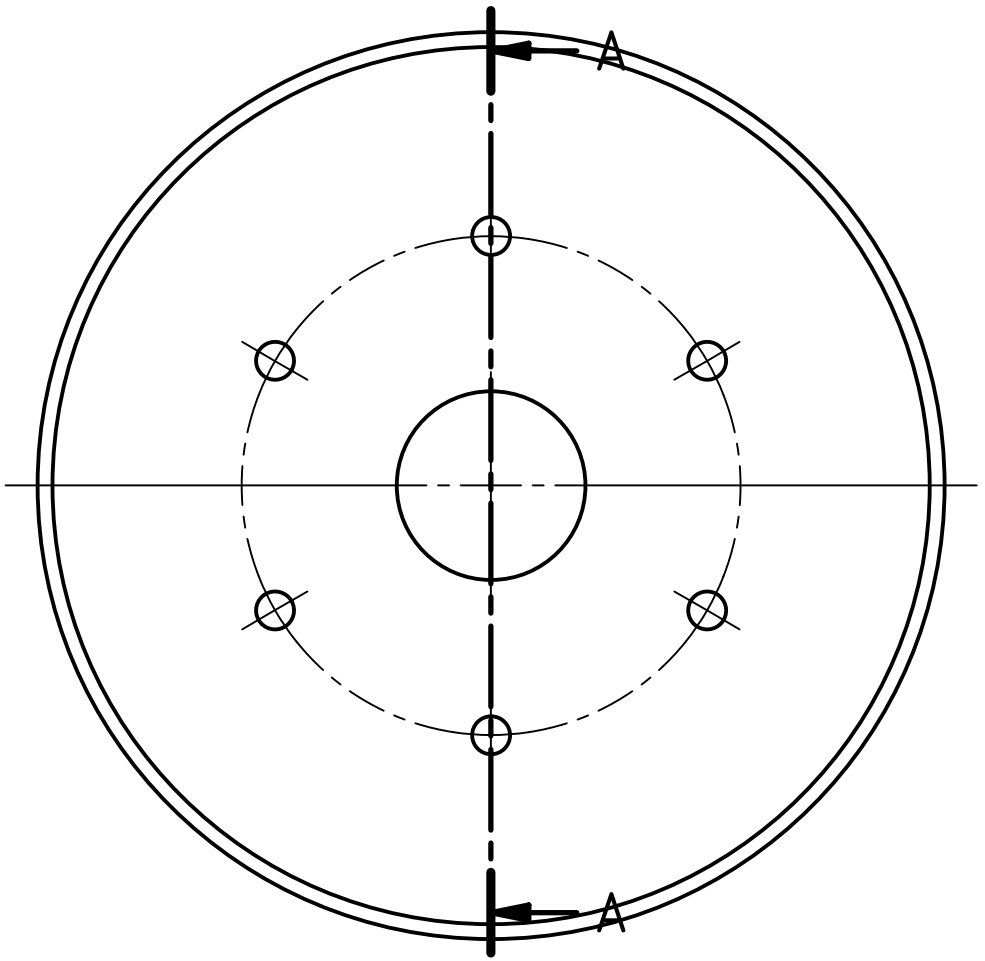
Name: Eichberger

Matr.-Nr.: 00909186

Datum: 07.10.2019

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub





Stück	Bezeichnung	Teil	Werkstoff	Norm	Bemerkung
1	Zwischenscheibe	T37	EN AW-5754 (AlMg3)		

Maßstab: 1:1

Allg.-Toleranz: ISO 2768-1 m

Institut für Konstruktionswissenschaften

TU Wien

Benennung:

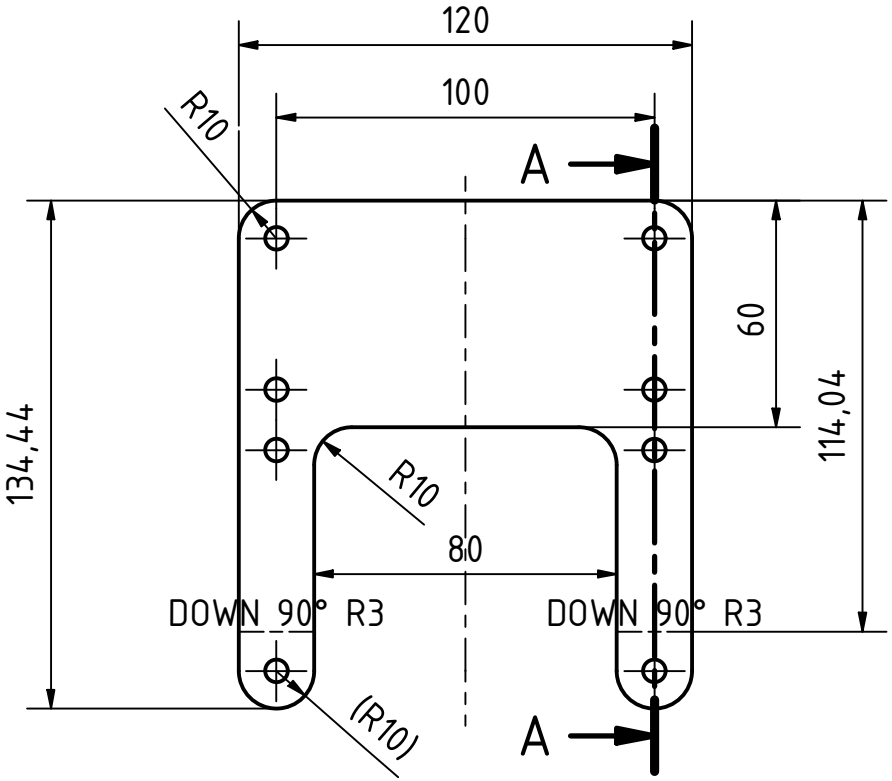
Zwischenscheibe

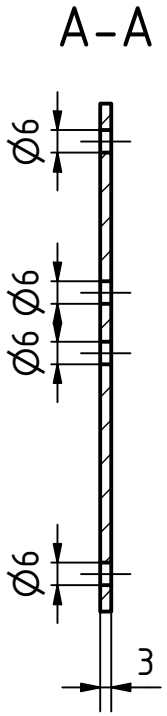
Name: Eichberger

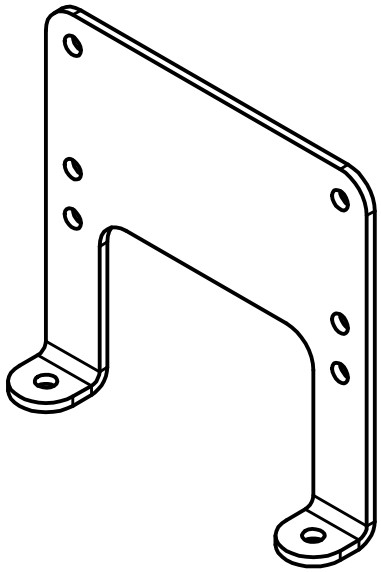
Matr.-Nr.: 00909186

Datum: 07.10.2019

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub







2	Halterung Schrittmotor	T38	1.4301		
Stück	Bezeichnung	Teil	Werkstoff	Norm	Bemerkung

Maßstab: 1:2

Allg.-Toleranz: ISO 2768-1 m

Institut für Konstruktionswissenschaften

TU Wien

Benennung:

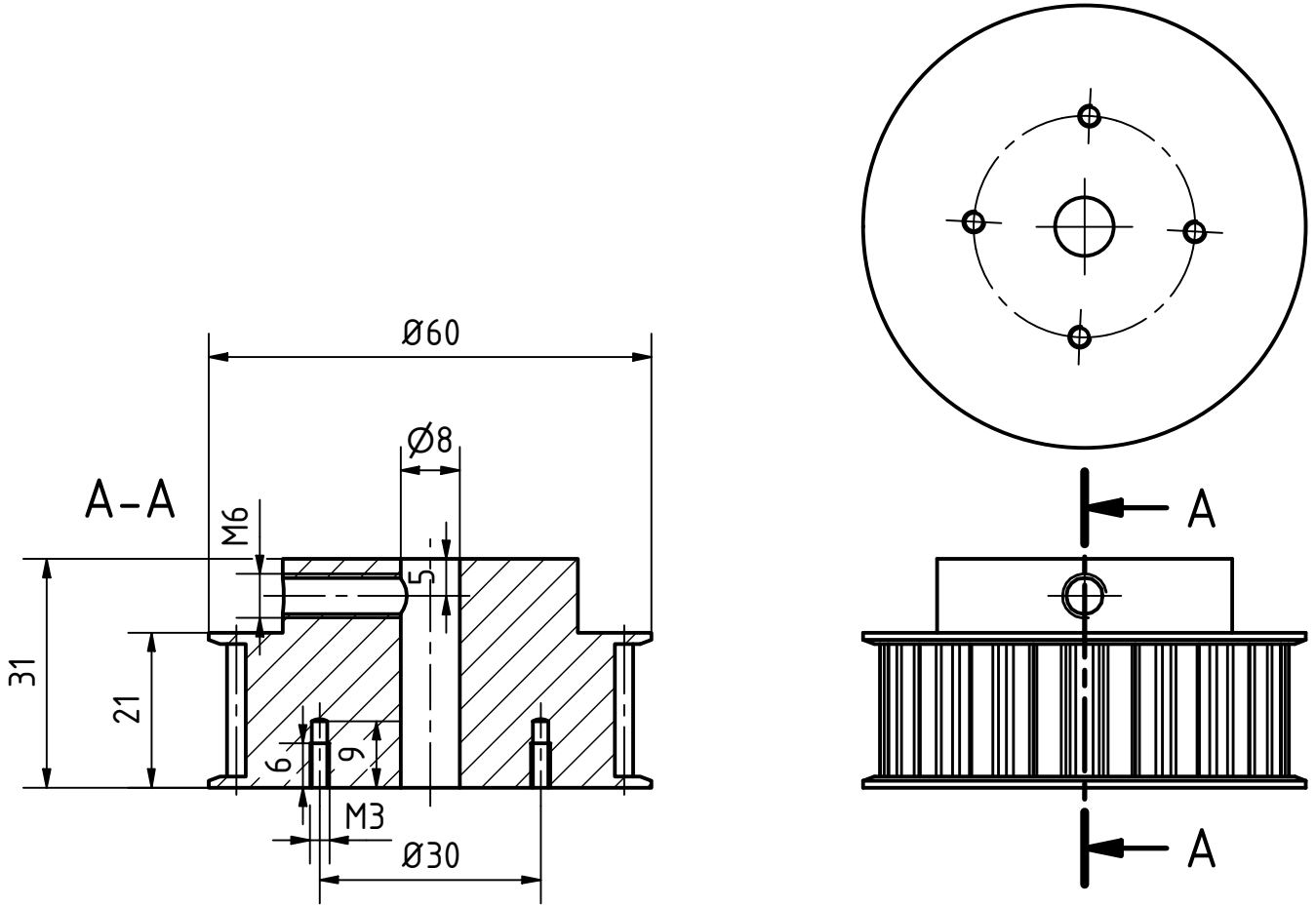
Halterung Schrittmotor

Name: Eichberger

Matr.-Nr.: 00909186

Datum: 07.10.2019

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub



1	Zahnriemenrad 1	T40	AlCuMg1		
Stück	Bezeichnung	Teil	Werkstoff	Norm	Bemerkung

Maßstab: 1:1

Allg.-Toleranz: ISO 2768-1 m

Institut für Konstruktionswissenschaften

TU Wien

Benennung:

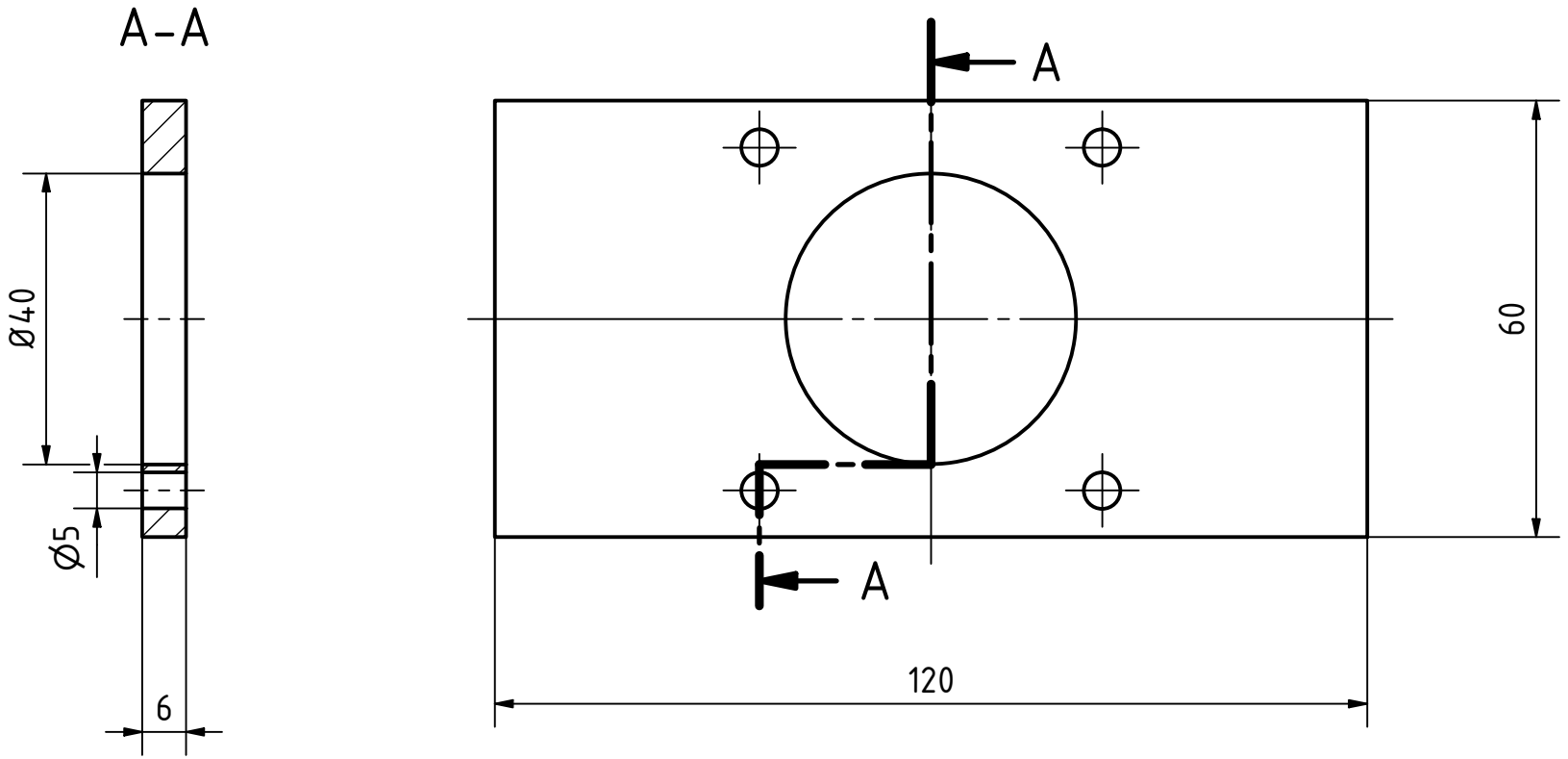
Zahnriemenrad 1

Name: Eichberger

Matr.-Nr.: 00909186

Datum: 07.10.2019

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub



1	Halteplatte	T41	EN AW-5754 (AlMg3)		
Stück	Bezeichnung	Teil	Werkstoff	Norm	Bemerkung

Maßstab: 1:1

Allg.-Toleranz: ISO 2768-1 m

Institut für Konstruktionswissenschaften

TU Wien

Benennung:

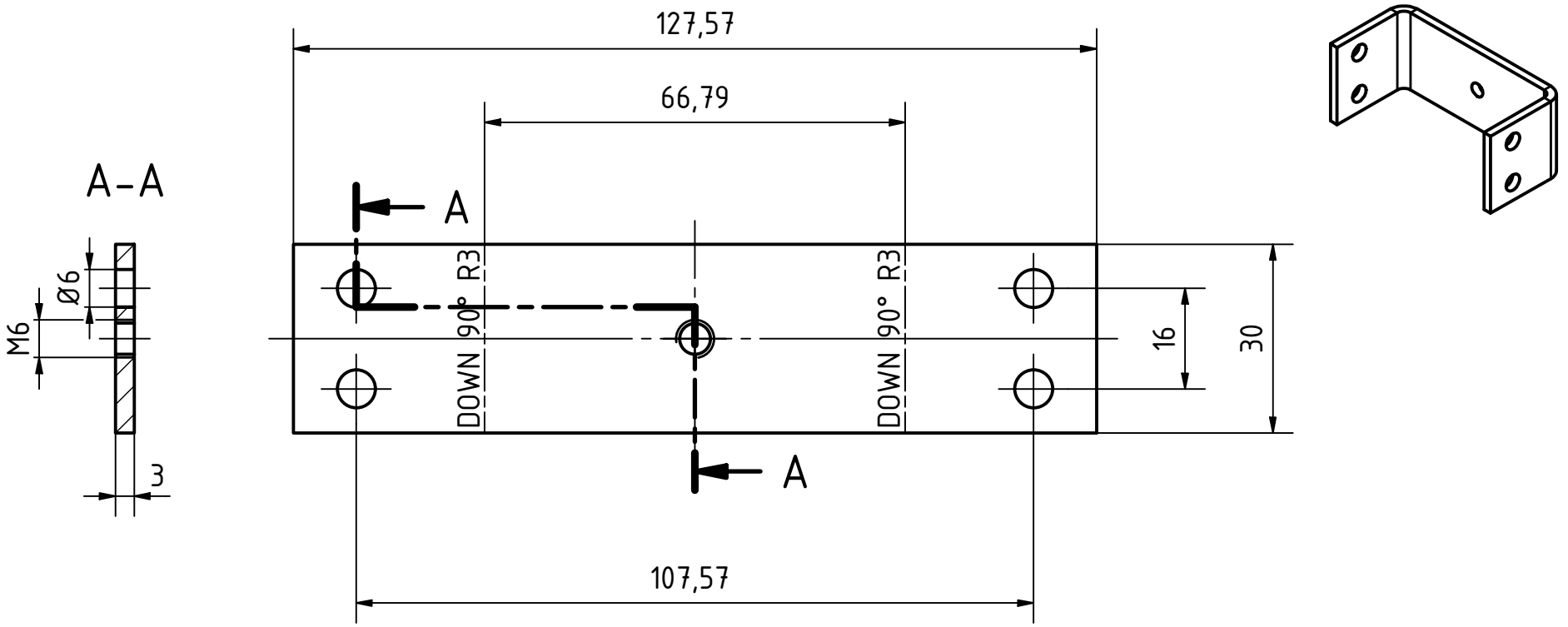
Halteplatte

Name: Eichberger

Matr.-Nr.: 00909186

Datum: 07.10.2019

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub



Stück	Bezeichnung	Teil	Werkstoff	Norm	Bemerkung
1	Spannteil	T42	1.4301		

Maßstab: 1:1

Allg.-Toleranz: ISO 2768-1 m

Institut für Konstruktionswissenschaften

TU Wien

Benennung:

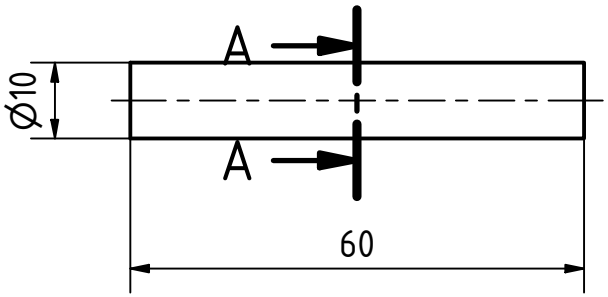
Spannteil

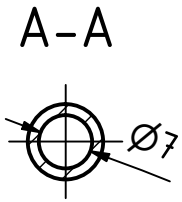
Name: Eichberger

Matr.-Nr.: 00909186

Datum: 07.10.2019

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub





6	Distanzhülse 4	T43	1.4301		
Stück	Bezeichnung	Teil	Werkstoff	Norm	Bemerkung

Maßstab: 1:1

Allg.-Toleranz: ISO 2768-1 m

Institut für Konstruktionswissenschaften

TU Wien

Benennung:

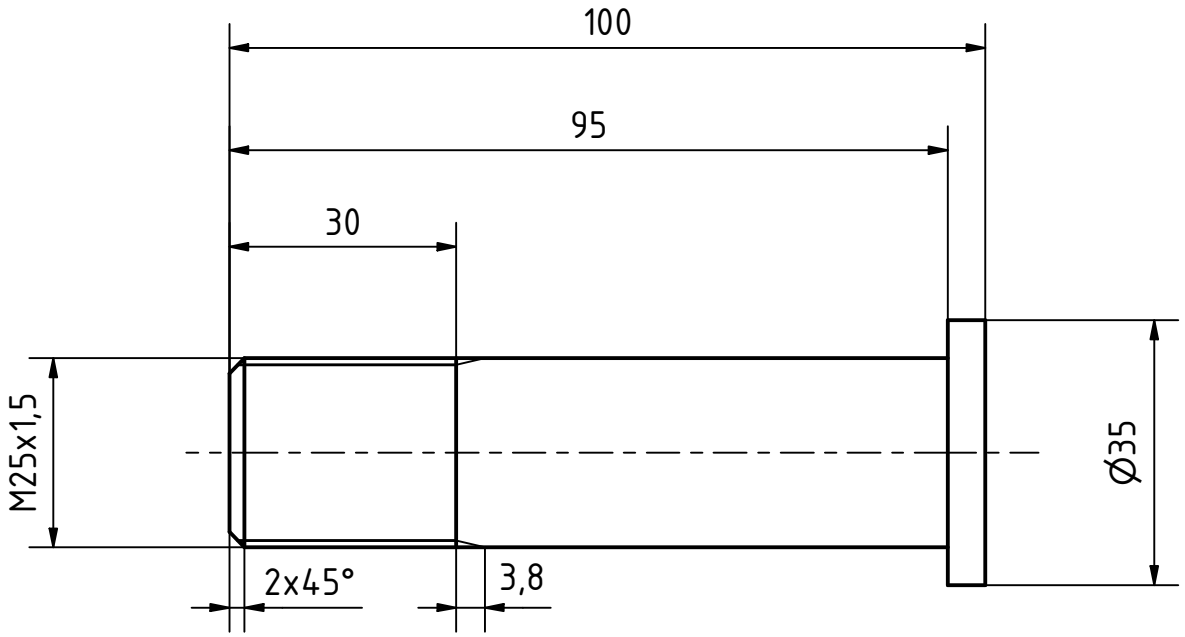
Distanzhülse 4

Name: Eichberger

Matr.-Nr.: 00909186

Datum: 08.10.2019

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub



1	Bolzen	T45	1.4301		
Stück	Bezeichnung	Teil	Werkstoff	Norm	Bemerkung

Maßstab: 1:1

Allg.-Toleranz: ISO 2768-1 m

Institut für Konstruktionswissenschaften

TU Wien

Benennung:

Bolzen

Name: Eichberger

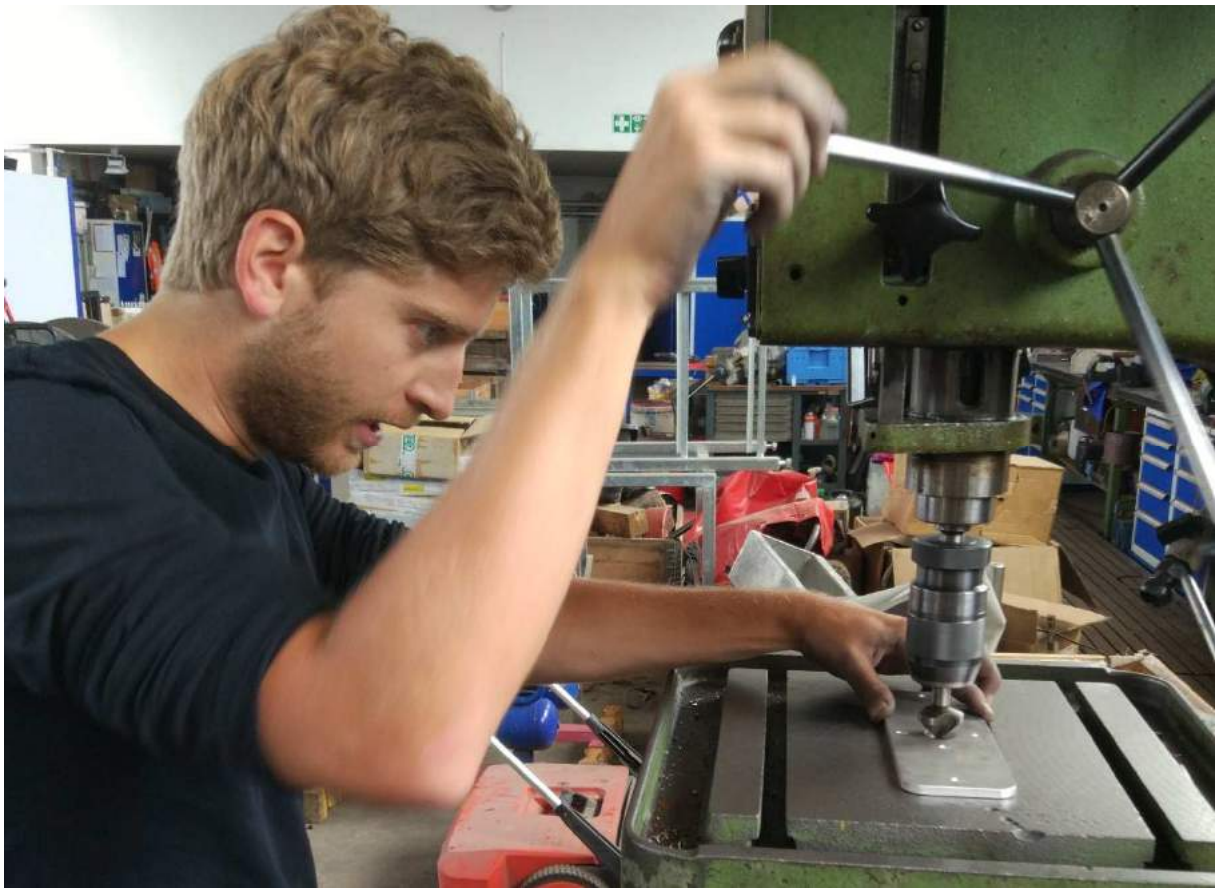
Matr.-Nr.: 00909186

Datum: 07.10.2019

Anhang D (Fertigung / Montage)

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

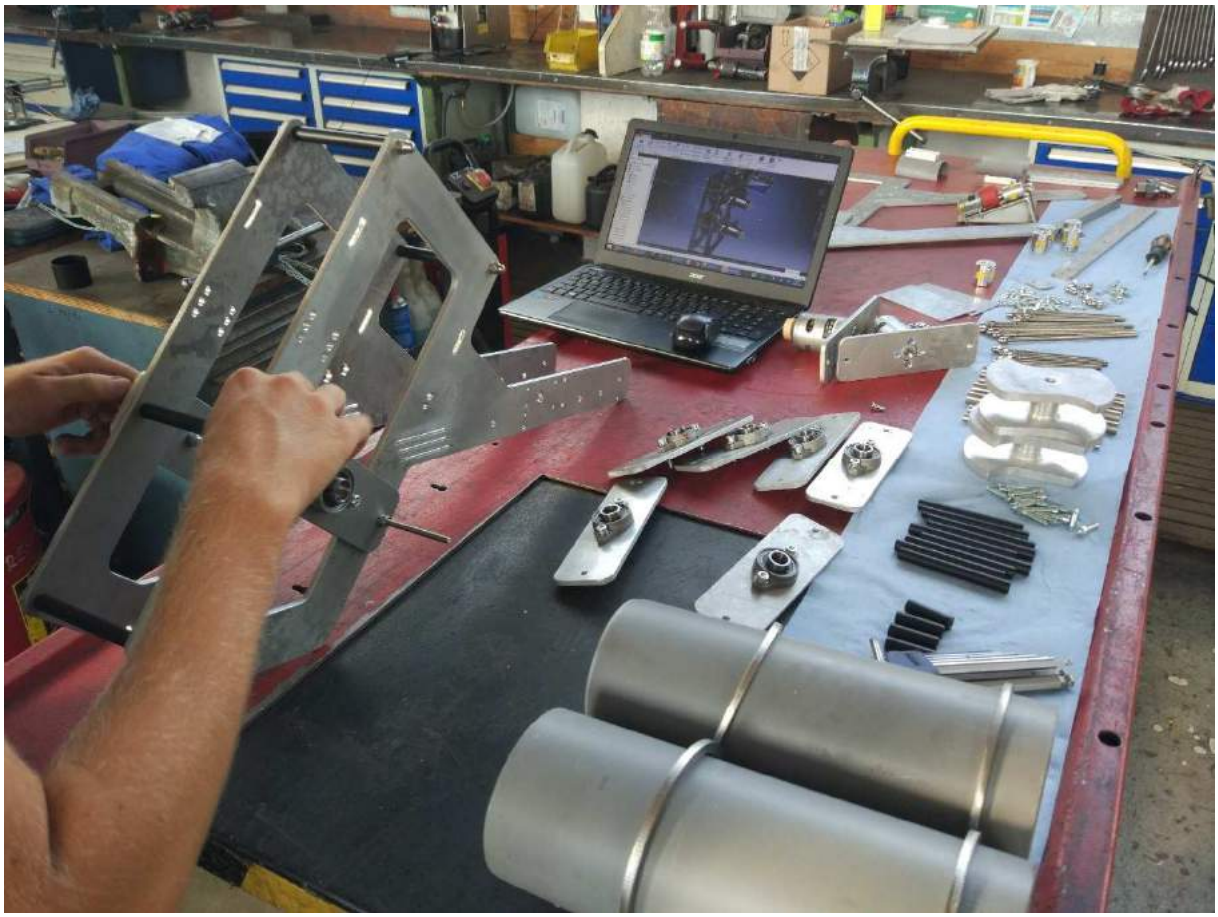
Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

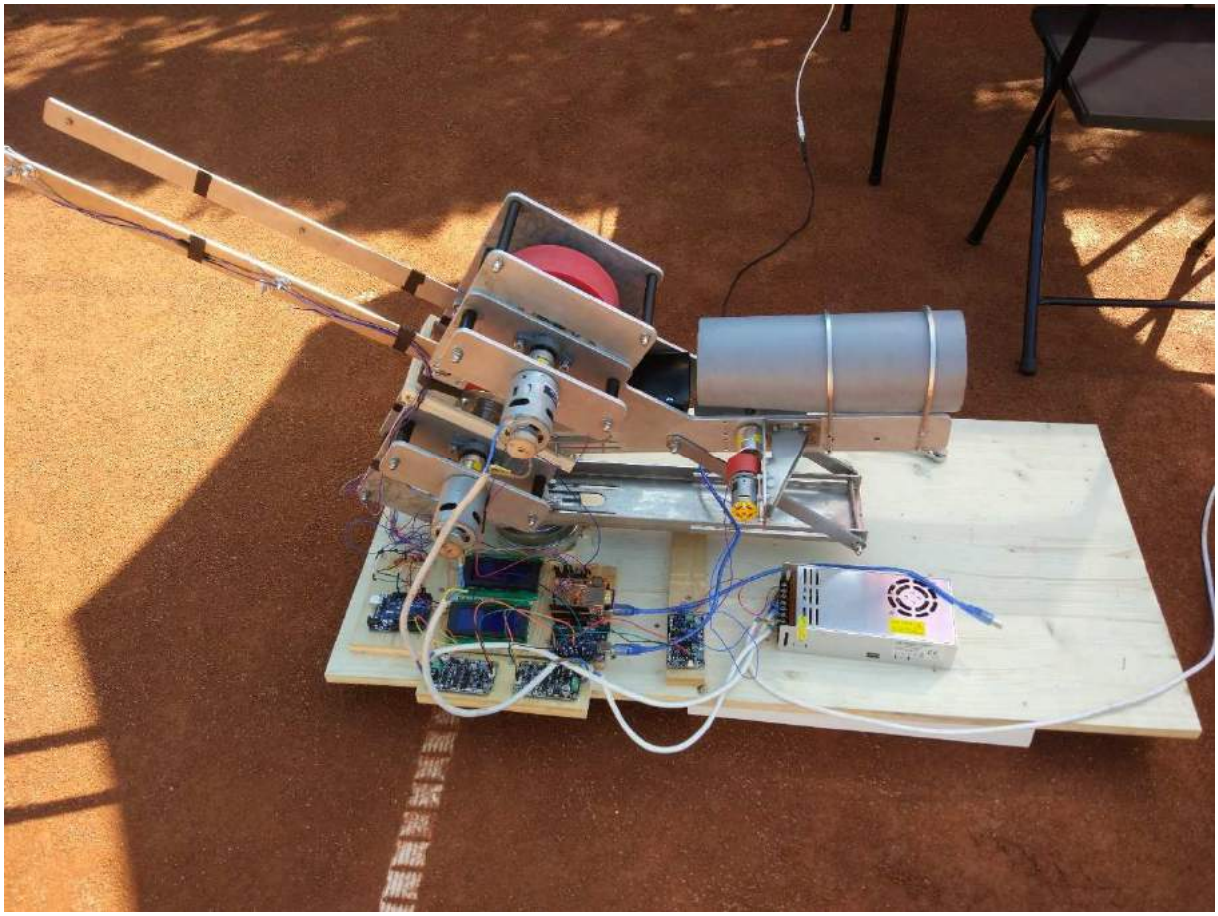
Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Anhang E (Messungen)



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Anhang F (Programmcode Funktionsmuster)

Code (Raspberry PI)

```
"""
Created August 2018 - November 2019
Titel: Diplomarbeit Tennisballmaschine
@author: Daniel Glatz
Version 1.1

"""

import sys
import serial
from RPi import GPIO as gpio
import math
import random
import numpy as np
from scipy.interpolate import interp1d
import PyQt5.QtWidgets as widgets
import PyQt5.QtCore as core
import PyQt5.uic as uic
import PyQt5.QtGui as gui
import functools as tools

# Variablen Definition
auswahl_var = "Bitte eine Auswahl treffen..."
training_var1 = "Grundlinie Vorhand Training"
training_var2 = "Grundlinie Rückhand Training"
training_var3 = "Grundlinien Training"
training_var4 = "Volley Vorhand Training"
training_var5 = "Volley Rückhand Training"
training_var6 = "Volley Training"
hohe_var1 = "schnell"
hohe_var2 = "mittel"
hohe_var3 = "langsam"
hohe_var4 = "Lob"
spin_var1 = "kein Spin"
spin_var2 = "Top-Spin"
bereich_var1 = "Vorhand - Seite"
bereich_var2 = "Mitte"
bereich_var3 = "Rückhand - Seite"
messageBox_verbindung = "Verbindung zu den Motoren wird aufgebaut...\n"
messageBox_verbindung_abb = "Motoren werden abgeschaltet...\n"
messageBox_verbindung_mot = "Die Motoren fahren auf die erforderliche Drehzahl...\n"

messageBox_Train = "Bitte ein Training auswählen!\n"
messageBox_Hohe = "Bitte die Geschwindigkeit des Balles auswählen!\n"
messageBox_Spin = "Bitte den Spin auswählen!\n"
messageBox_Bereich = "Bitte einen Bereich auswählen!\n"
messageBox_schliessen1 = "Fenster schließt automatisch in {0} Sekunden."
messageBox_schliessen2 = "Fenster schließt automatisch in einer {0} Sekunde."
messageBox_schlMot1 = "Der erste Ball wird in {0} Sekunden abgeschossen."
messageBox_schlMot2 = "Der erste Ball wird in einer {0} Sekunde abgeschossen."
hand = 1 # Hilfsvariable für die Händigkeit; 1=rechts, 2=links
global v_return  # Anfangsgeschwindigkeit
global hohe_eingabe # vertikale Winkeleinstellung
global modus # Spielstärke; 1=anfänger, 2=fortgeschritten
global lambda_Ab # Korrekturfaktor
global spin_eingabe # Spin; 0=kein Spin, 1=Top-Spin
lambda_Ab= 1
def_hohe_netz = 0.93 # maximale Netzhöhe bei einem Winkel von 15°
LED_PIN = 23   # PIN für den Lichtschranken
PORT = '/dev/ttyACM0' # Port für den Arduino
hilfsvar = 0 # Hilfsvariable für den Lichtschranken
```

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

```
61. global mylist_weite # Liste für die Flugweiten
62. global mylist_winkel # Liste für den horizontalen Winkel
63. mylist_winkel = [0, 15, 30]
64.
65. verbindung_motoren = serial.Serial(PORT, 9600) # Port Zuweisung für den Arduino
66.
67. app = widgets.QApplication(sys.argv)
68. mainWin = uic.loadUi("tennismaschine_neu_v1.ui") # Laden der graphischen Oberfläche
    des Programms
69. dialogKali = uic.loadUi("kalibrieren1.ui") # Laden der graphischen Oberfläche des Di
    alog-Fensters Kalibrieren
70. dialogHand = uic.loadUi("handigkeit.ui") # Laden der graphischen Oberfläche des Dial
    og-Fensters Händigkeit
71. dialogLaser = uic.loadUi("laser.ui") # Laden der graphischen Oberfläche des Dialog-
    Fensters Laser einstellen
72. dialogInfo = uic.loadUi("info.ui") # Laden der graphischen Oberfläche des Dialog-
    Fensters Info
73. dialogFehl = uic.loadUi("fehlermeldung.ui") # Laden der graphischen Oberfläche des D
    ialog-Fensters Fehlermeldung
74. dialogNetz = uic.loadUi("hohe_netz.ui") # Laden der graphischen Oberfläche des Dialo
    g-Fensters Hohe_Netz
75. dialogPos = uic.loadUi("pos_einstell.ui") # Laden der graphischen Oberfläche des Dia
    log-Fensters Position einstellen
76.
77. # MessageBox für den Verbindungsaufbau der Motoren
78. class TimerMessageBox(widgets.QMessageBox):
79.     def __init__(self, timeout, messageBox_text, messageBox_schl1, messageBox_schl2,
     parent=None):
80.         super(TimerMessageBox, self).__init__(parent)
81.         self.setWindowTitle("Achtung!")
82.         self.time_to_wait = timeout
83.         self.setText(messageBox_text + messageBox_schl1.format(timeout))
84.         self.setStandardButtons(widgets.QMessageBox.NoButton)
85.         self.timer = core.QTimer(self)
86.         self.timer.setInterval(1000)
87.         self.timer.timeout.connect(tools.partial(self.changeContent, messageBox_text
    , messageBox_schl1, messageBox_schl2))
88.         self.timer.start()
89.
90.     def changeContent(self, messageBox_text, messageBox_schl1, messageBox_schl2):
91.         self.time_to_wait -= 1
92.         if self.time_to_wait == 1:
93.             self.setText(messageBox_text + messageBox_schl2.format(self.time_to_wait
    ))
94.         elif self.time_to_wait <= 0:
95.             self.close()
96.         else:
97.             self.setText(messageBox_text + messageBox_schl1.format(self.time_to_wait
    ))
98.
99.     # Funktion, die beim Schließen der MessageBox aufgerufen wird. Beendet den Timer
    .
100.         def closeEvent(self, event):
101.             self.timer.stop()
102.             event.accept()
103.
104.     # PWM ermittlen mittels dem Schießverfahren
105.     def geschw_berechnen(L_0,beta,spin,lambda_Ab_ueber):
106.         m = 0.0575      # [kg] Masse des Balls
107.         d = 0.067       # [m] Durchmesser des Balls
108.         h_0 = 0.35      # [m] Abschusshöhe
109.         x_0 = 0         # Abschusspunkt
110.         C_D = 0.507     # [1] Luftwiderstands-Wert
111.         korr = 1        # Korrekturfaktor
112.         delta_t = 0.01 # [s] Schrittweite
113.         rho = 1.2       # [kg/m^3] Dichte der Luft
114.         g = 9.81        # (m/s^2) Fallbeschleunigung
```

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

```
115.
116.           A = (d/2)*(d/2)*math.pi  # [m^2] Querschnittsfläche des Balls
117.           k = A*rho/(2*m)          # [1/m] Hilfswert
118.           maxAbw_spin = 0.1        # maximale Abweichung für die Berechnung des Spi
    ns
119.
120.           # Berechnung der Anfangsgeschwindigkeit mittels Bisektionsverfahren
121.           def v_0_berechnen(L,n):
122.               v_1 = 40          # Startwert 1 für Bisektion
123.               v_2 = 150         # Startwert 2 für Bisektion
124.               maxAbw = 0.001    # max. erlaubte Abweichung vom Zielwert
125.               i = 0    # Zähler für Anzahl an Iterationen
126.               omega = 2*math.pi*n/60;  # [1/s] Drehzahl des Balls
127.
128.               while True:
129.                   i = 0
130.                   v_i = (v_1+v_2)/2
131.                   C_L = -0.6*d/2*omega/(v_i/3.6) # [1] Lift-Wert
132.
133.                   # --------Anfangsbedingungen----------
134.                   t = np.zeros( (1, 1) )
135.                   t[0][0] = 0
136.                   z = np.zeros( (1, 4) )
137.                   z[0][0] = -x_0
138.                   z[0][1] = (v_i/3.6)*math.cos(beta*math.pi/180)
139.                   z[0][2] = h_0
140.                   z[0][3] = (v_i/3.6)*math.sin(beta*math.pi/180)
141.                   v = np.zeros( (1, 1) )
142.                   v[0][0] = math.sqrt(z[0][1]*z[0][1]+z[0][3]*z[0][3])
143.
144.                   #------Lösung-----------
145.                   while True:
146.                       t1 = np.zeros((1,1))
147.                       t = np.concatenate((t, t1))
148.                       t[i+1][0] = t[i][0]+delta_t
149.                       z1 = np.zeros((1,4))
150.                       z = np.concatenate((z, z1))
151.                       z[i+1][0] = z[i][0]+delta_t*z[i][1]
152.                       z[i+1][1] = z[i][1]+delta_t*(k*v[i][0]*(-C_D*z[i][1]-
    C_L*z[i][3]))
153.                       z[i+1][2] = z[i][2]+delta_t*z[i][3]
154.                       z[i+1][3] = z[i][3]+delta_t*(k*v[i][0]*(-
    C_D*z[i][3]+C_L*z[i][1])-g)
155.                       v1 = np.zeros((1,1))
156.                       v = np.concatenate((v, v1))
157.                       v[i+1][0] = math.sqrt(z[i+1][1]*z[i+1][1]+z[i+1][3]*z[i+1][3]
    )
158.                       if z[-1][2]<0:
159.                           break
160.                       i = i+1
161.
162.                   xvalues = z[:,0]    # x-
    Koordinaten der Flugbahn als neuer Vektor
163.                   #xdotvalues = z[:,1]   # x°-
    Koordinaten der Flugbahn als neuer Vektor
164.                   yvalues = z[:,2]    # x-
    Koordinaten der Flugbahn als neuer Vektor
165.                   #ydotvalues = z[:,3]   # x°-
    Koordinaten der Flugbahn als neuer Vektor
166.
167.                   f = interp1d(yvalues,xvalues)
168.                   x_i = f(0)-L
169.
170.                   if abs(x_i)<maxAbw:   # Ausstieg aus der Schleife, wenn erlaubte
    Abweichung unterschritten
171.                       break
172.                   if x_i > 0:
```

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

```
                    v_2 = v_i     # Anpassung der Startwerte
                if x_i < 0:
                    v_1 = v_i    # Anpassung der Sartwerte

             #------Lösung-----------
            v_0 = v_i*korr
            return(v_0)

        # Berechnung der Flugweite
        def L_0_berechnen(v_0,n):
            i = 0   # Zähler für Anzahl an Iterationen
            omega = 2*math.pi*n/60;           # [1/s] Drehzahl des Balls
            C_L = -0.6*d/2*omega/(v_0/3.6)   # [1] Lift-Wert

            # --------Anfangsbedingungen----------
            t = np.zeros( (1, 1) )
            t[0][0] = 0
            z = np.zeros( (1, 4) )
            z[0][0] = -x_0
            z[0][1] = (v_0/3.6)*math.cos(beta*math.pi/180)
            z[0][2] = h_0
            z[0][3] = (v_0/3.6)*math.sin(beta*math.pi/180)
            v = np.zeros( (1, 1) )
            v[0][0] = math.sqrt(z[0][1]*z[0][1]+z[0][3]*z[0][3])

                #------Lösung-----------
            while True:
                t1 = np.zeros((1,1))
                t = np.concatenate((t, t1))
                t[i+1][0] = t[i][0]+delta_t
                z1 = np.zeros((1,4))
                z = np.concatenate((z, z1))
                z[i+1][0] = z[i][0]+delta_t*z[i][1]
                z[i+1][1] = z[i][1]+delta_t*(k*v[i][0]*(-C_D*z[i][1]-
    C_L*z[i][3]))
                z[i+1][2] = z[i][2]+delta_t*z[i][3]
                z[i+1][3] = z[i][3]+delta_t*(k*v[i][0]*(-
    C_D*z[i][3]+C_L*z[i][1])-g)
                v1 = np.zeros((1,1))
                v = np.concatenate((v, v1))
                v[i+1][0] = math.sqrt(z[i+1][1]*z[i+1][1]+z[i+1][3]*z[i+1][3])
                if z[-1][2]<0:
                    break
                i = i+1

            xvalues = z[:,0]    # x-Koordinaten der Flugbahn als neuer Vektor
            #xdotvalues = z[:,1]   # x°-
    Koordinaten der Flugbahn als neuer Vektor
            yvalues = z[:,2]    # x-Koordinaten der Flugbahn als neuer Vektor
            #ydotvalues = z[:,3]   # x°-
    Koordinaten der Flugbahn als neuer Vektor

            f = interp1d(yvalues,xvalues)
            L = f(0)
            f_hohe = interp1d(xvalues,yvalues)
            hohe_netz = round(float(f_hohe(23.77/2)),2)
            return L, hohe_netz

        # Funktion zur Berechnung des Spins bei der jeweiligen Anfangsgeschwindig
    keit
        def spin_berechnen(v_i):
            if spin == 0:
                spin_fkt_n = 0.17196515051721*v**3 - 33.4021288102979*v**2 + 2161
    .48601320986*v - 46302.4285340803
            elif spin == 1:
                spin_fkt_n = 0.151862657030911*v**3 - 27.6100414551534*v**2 + 168
    4.30133108852*v - 33840.6347656877
```

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

```
233.                return spin_fkt_n
234.
235.            v = v_0_berechnen(L_0,0)
236.            n_i = spin_berechnen(v)
237.            L_i, hohe_netz = L_0_berechnen(v,n_i)
238.
239.            while True:
240.                delta_weite = L_0 - L_i
241.                if abs(delta_weite) < maxAbw_spin:   # Ausstieg aus der Schleife, wen
    n erlaubte Abweichung unterschritten
242.                    break
243.                if delta_weite > 0:
244.                    L_i = L_0 + delta_weite/2
245.                elif delta_weite < 0:
246.                    L_i = L_0 - delta_weite/2
247.                n_i = spin_berechnen(v)
248.                v = v_0_berechnen(L_i,n_i)
249.                n_i = spin_berechnen(v)
250.                L_i, hohe_netz = L_0_berechnen(v,n_i)
251.
252.            v = v * lambda_Ab_ueber # Einbeziehung des Korrekturfaktors
253.            if(hohe_netz < def_hohe_netz):
254.                return 0,0,round(v, 2)
255.            else:
256.                # Berechnung der PWM-
    Werte bei der jeweiligen Anfangsgeschwindigkeit
257.                if spin == 0:
258.                    PWM_o = 0.0000682724238458832*v**6 - 0.0265352402084091*v**5 + 4.
    29069132651669*v**4 - 369.462020094676*v**3 + 17868.0963636373*v**2 - 460185.8227861
    82*v + 4931066.26085955
259.                    PWM_u = PWM_o * (1/1.1)
260.                elif spin == 1:
261.                    PWM_o = 0.0000648080724809 93*v**6 - 0.0244043993283596*v**5 + 3.8
    1772111056201*v**4 - 317.564062193107*v**3 + 14813.5892188489*v**2 - 367413.53468839
    4*v + 3785390.52796699
262.                    PWM_u = PWM_o * (1/1.25)
263.                return int(round(PWM_o,0)),int(round(PWM_u,0)),round(v, 2)
264.
265.        # Klick auf den Kalibrieren Button
266.        def clickKalibrieren():
267.            # Bei Klick auf den OK-Button
268.            if dialogKali.exec_() == widgets.QDialog.Accepted:
269.                global lambda_Ab
270.                abweichung = dialogKali.doubleSpinBox_abweichung.value()
271.                PWM_oben, PWM_unten, v_mit_Ab = geschw_berechnen(weite+abweichung,hoh
    e_eingabe,spin_eingabe,lambda_Ab) # Funktion aufrufen
272.                lambda_Ab = v_return / v_mit_Ab # Berechnung des Korrekturfaktors
273.
274.        # Klick auf den Menüeintrag Händigkeit
275.        def clickHandigkeit():
276.            # Bei Klick auf den OK-Button
277.            if dialogHand.exec_() == widgets.QDialog.Accepted:
278.                global hand
279.                # Benutzer ist Rechtshänder
280.                if dialogHand.radioButton_hand_r.isChecked():
281.                    hand = 1
282.                # Benutzer ist Linkshänder
283.                elif dialogHand.radioButton_hand_l.isChecked():
284.                    hand = 2
285.                bildAendern(mainWin.comboBox_training.currentText()) # Ändern des Tra
    inings bei Umstellung der Händigkeit
286.                bildAendern_2(mainWin.comboBox_bereich.currentText()) # Ändern des Tr
    ainings bei Umstellung der Händigkeit
287.
288.        # Klick auf den Menüeintrag Laser einstellen
289.        def clickLaser():
290.            gpio.setmode(gpio.BCM)
```

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

```
291.           gpio.setwarnings(False)
292.
293.           def callback_func(channel):
294.               if gpio.input(channel) == gpio.LOW: # Fotodiode wird nicht beleuchtet
    mit dem Laser
295.                   dialogLaser.label.setText("Laser nicht richtig positioniert")
296.
297.               elif gpio.input(channel) == gpio.HIGH: # Fotodiode wird beleuchtet vo
    n dem Laser
298.                   dialogLaser.label.setText("Laser richtig positioniert")
299.
300.           gpio.setup(LED_PIN, gpio.IN) # LED PIN als Eingang definieren
301.           gpio.add_event_detect(LED_PIN, gpio.BOTH, callback=callback_func, bouncet
    ime=50)
302.           # Bei Klick auf den OK-Button
303.           if dialogLaser.exec_() == widgets.QDialog.Accepted:
304.               gpio.cleanup() # LED-PIN deaktivieren
305.
306.       # Klick auf den Menüeintrag Info
307.       def clickInfo():
308.           dialogInfo.exec_() # Dialog-Fenster Info öffnen
309.
310.       # Wenn ein Auswahlmöglichkeit nicht ausgewählt wurde
311.       def clickFehl(meldung):
312.           dialogFehl.label.setText(meldung) # Label zeigt die Fehlermeldung an
313.           dialogFehl.exec_() # Dialog-Fenster Fehlermeldung öffnen
314.
315.       # Wenn der Ball die Netzhöhe nicht erreicht
316.       def clickNetz():
317.           dialogNetz.exec_() # Dialog-Fenster Hohe_Netz öffnen
318.
319.       def clickPos():
320.           dialogPos.exec_() # Dialog-Fenster Hohe_Netz öffnen
321.
322.       # Bei Klick auf den Stop-
    Button beim Unterprogramm Trainingsprogramme werden bestimmte Felder auswählbar bzw.
     nicht auswählbar
323.       def clickStopEnabled():
324.           mainWin.pushButton_start.setEnabled(True)
325.           mainWin.comboBox_training.setEnabled(True)
326.           mainWin.comboBox_hohe.setEnabled(True)
327.           mainWin.comboBox_spin.setEnabled(True)
328.           mainWin.pushButton_kali.setEnabled(True)
329.           mainWin.actionKalibrieren.setEnabled(True)
330.           mainWin.actionHandigkeit.setEnabled(True)
331.           mainWin.actionLaser_einstellen.setEnabled(True)
332.           mainWin.radioButton_modus1.setEnabled(True)
333.           mainWin.radioButton_modus2.setEnabled(True)
334.           mainWin.tab_2.setEnabled(True)
335.           mainWin.pushButton_stop.setEnabled(False)
336.
337.       # Bei Klick auf den Stop-
    Button beim Unterprogramm manuelles Training werden bestimmte Felder auswählbar bzw.
     nicht auswählbar
338.       def clickStopEnabled_2():
339.           mainWin.pushButton_start_2.setEnabled(True)
340.           mainWin.comboBox_bereich.setEnabled(True)
341.           mainWin.comboBox_spin_2.setEnabled(True)
342.           mainWin.pushButton_kali_2.setEnabled(True)
343.           mainWin.actionKalibrieren.setEnabled(True)
344.           mainWin.actionHandigkeit.setEnabled(True)
345.           mainWin.actionLaser_einstellen.setEnabled(True)
346.           mainWin.tab.setEnabled(True)
347.           mainWin.doubleSpinBox_weite.setEnabled(True)
348.           mainWin.pushButton_stop_2.setEnabled(False)
349.
```

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

```
350.        # Bei Klick auf den Stop-
    Button beim Dialog Kalibrieren werden bestimmte Felder auswählbar bzw. nicht auswähl
    bar
351.        def clickStopEnabled_3():
352.            dialogKali.pushButton_start_3.setEnabled(True)
353.            dialogKali.doubleSpinBox_abweichung.setEnabled(True)
354.            dialogKali.buttonBox.setEnabled(True)
355.            dialogKali.comboBox_hohe.setEnabled(True)
356.            dialogKali.pushButton_stop_3.setEnabled(False)
357.
358.        # Bei Klick auf den Start-
    Button beim Unterprogramm Trainingsprogramme werden bestimmte Felder auswählbar bzw.
     nicht auswählbar
359.        def clickStartEnabled():
360.            mainWin.pushButton_start.setEnabled(False)
361.            mainWin.comboBox_training.setEnabled(False)
362.            mainWin.comboBox_hohe.setEnabled(False)
363.            mainWin.comboBox_spin.setEnabled(False)
364.            mainWin.pushButton_kali.setEnabled(False)
365.            mainWin.actionKalibrieren.setEnabled(False)
366.            mainWin.actionHandigkeit.setEnabled(False)
367.            mainWin.actionLaser_einstellen.setEnabled(False)
368.            mainWin.radioButton_modus1.setEnabled(False)
369.            mainWin.radioButton_modus2.setEnabled(False)
370.            mainWin.tab_2.setEnabled(False)
371.            mainWin.pushButton_stop.setEnabled(True)
372.
373.        # Bei Klick auf den Start-
    Button beim Unterprogramm manuelles Training werden bestimmte Felder auswählbar bzw.
     nicht auswählbar
374.        def clickStartEnabled_2():
375.            mainWin.pushButton_start_2.setEnabled(False)
376.            mainWin.comboBox_bereich.setEnabled(False)
377.            mainWin.comboBox_spin_2.setEnabled(False)
378.            mainWin.pushButton_kali_2.setEnabled(False)
379.            mainWin.actionKalibrieren.setEnabled(False)
380.            mainWin.actionHandigkeit.setEnabled(False)
381.            mainWin.actionLaser_einstellen.setEnabled(False)
382.            mainWin.tab.setEnabled(False)
383.            mainWin.doubleSpinBox_weite.setEnabled(False)
384.            mainWin.pushButton_stop_2.setEnabled(True)
385.
386.        # Bei Klick auf den Start-
    Button beim Dialog Kalibrieren werden bestimmte Felder auswählbar bzw. nicht auswähl
    bar
387.        def clickStartEnabled_3():
388.            dialogKali.pushButton_start_3.setEnabled(False)
389.            dialogKali.doubleSpinBox_abweichung.setEnabled(False)
390.            dialogKali.buttonBox.setEnabled(False)
391.            dialogKali.comboBox_hohe.setEnabled(False)
392.            dialogKali.pushButton_stop_3.setEnabled(True)
393.
394.        # Zeichenkette an Arduino schicken
395.        def changePMW(verbindung, pwm):
396.            wert_stern = pwm + "*"
397.            verbindung.write(wert_stern.encode())
398.            verbindung.readline()
399.
400.        # Klick auf den Start-Button beim Unterprogramm Trainingsprogramme
401.        def clickStart():
402.            # Trainingsprogramm ausgewählt?
403.            if mainWin.comboBox_training.currentText() == auswahl_var:
404.                clickFehl(messageBox_Train)
405.            # vertikale Winkeleinstellung ausgewählt?
406.            elif mainWin.comboBox_hohe.currentText() == auswahl_var:
407.                clickFehl(messageBox_Hohe)
408.            # Spin ausgewählt?
```

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

```
409.           elif mainWin.comboBox_spin.currentText() == auswahl_var:
410.               clickFehl(messageBox_Spin)
411.
412.           # Zuweisung des Falles
413.           else:
414.               global hilfsvar
415.               hilfsvar = 0
416.               clickStartEnabled() # Aufruf der Funktion
417.               # Steht die Verbindung zum Arduino schon?
418.               if(verbindung_motoren.isOpen()==False):
419.                   verbindung_motoren.open() # Verbindung zum Arduino aufbauen
420.               messageboxVerbindung = TimerMessageBox(3, messageBox_verbindung, mess
    ageBox_schliessen1, messageBox_schliessen2, mainWin)
421.               messageboxVerbindung.exec_() # Messagebox aufrufen
422.               # Spin erwünscht?
423.               if mainWin.comboBox_spin.currentText() == spin_var1:
424.                   spin_eingabe = 0 # kein Spin
425.               elif mainWin.comboBox_spin.currentText() == spin_var2:
426.                   spin_eingabe = 1 # Top Spin
427.               # vertikale Winkeleinstellung
428.               if mainWin.comboBox_hohe.currentText() == hohe_var1:
429.                   hohe_eingabe = 15
430.                   dialogPos.label.setText("Position 1")
431.               elif mainWin.comboBox_hohe.currentText() == hohe_var2:
432.                   hohe_eingabe = 20
433.                   dialogPos.label.setText("Position 2")
434.               elif mainWin.comboBox_hohe.currentText() == hohe_var3:
435.                   hohe_eingabe = 25
436.                   dialogPos.label.setText("Position 3")
437.               elif mainWin.comboBox_hohe.currentText() == hohe_var4:
438.                   hohe_eingabe = 30
439.                   dialogPos.label.setText("Position 4")
440.               # Spielstärke-Modus festlegen
441.               if mainWin.radioButton_modus1.isChecked():
442.                   modus = 1 # Anfänger
443.               elif mainWin.radioButton_modus2.isChecked():
444.                   modus = 2 # Fortgeschritten
445.               # Rechts- oder Linkshänder?
446.               if hand == 1:
447.                   # horizontale Winkeleinstellung
448.                   if (mainWin.comboBox_training.currentText() == training_var1) or
    (mainWin.comboBox_training.currentText() == training_var4) :
449.                       winkel = 0
450.                   elif (mainWin.comboBox_training.currentText() == training_var2) o
    r (mainWin.comboBox_training.currentText() == training_var5) :
451.                       winkel = 30
452.                   elif (mainWin.comboBox_training.currentText() == training_var3) o
    r (mainWin.comboBox_training.currentText() == training_var6) :
453.                       winkel = 15
454.                       hilfsvar = 1
455.               elif hand == 2:
456.                   # horizontale Winkeleinstellung
457.                   if (mainWin.comboBox_training.currentText() == training_var1) or
    (mainWin.comboBox_training.currentText() == training_var4) :
458.                       winkel = 30
459.                   elif (mainWin.comboBox_training.currentText() == training_var2) o
    r (mainWin.comboBox_training.currentText() == training_var5) :
460.                       winkel = 0
461.                   elif (mainWin.comboBox_training.currentText() == training_var3) o
    r (mainWin.comboBox_training.currentText() == training_var6) :
462.                       winkel = 15
463.                       hilfsvar = 1
464.               # Training Grundlinie
465.               if (mainWin.comboBox_training.currentText() == training_var1) or (mai
    nWin.comboBox_training.currentText() == training_var2) or (mainWin.comboBox_training
    .currentText() == training_var3):
466.                   # Spin erwünscht?
```

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

```
467.                     if mainWin.comboBox_spin.currentText() == spin_var1:
468.                         if mainWin.comboBox_hohe.currentText() == hohe_var1:
469.                             mylist_weite = [18.2, 17.5, 18.0, 17.2]
470.                         elif mainWin.comboBox_hohe.currentText() == hohe_var2:
471.                             mylist_weite = [21.0, 21.3, 20.5, 21.6]

472.                         elif mainWin.comboBox_hohe.currentText() == hohe_var3:
473.                             mylist_weite = [21.7, 22.5, 22.0, 21.2]
474.                         elif mainWin.comboBox_hohe.currentText() == hohe_var4:
475.                             mylist_weite = [22.3, 21.0, 21.7, 22.7]
476.                     elif mainWin.comboBox_spin.currentText() == spin_var2:
477.                         if mainWin.comboBox_hohe.currentText() == hohe_var1:
478.                             mylist_weite = [16.9, 16.4, 16.7, 16.2]
479.                         elif mainWin.comboBox_hohe.currentText() == hohe_var2:
480.                             mylist_weite = [20.1, 19.7, 19.5, 19.0]

481.                         elif mainWin.comboBox_hohe.currentText() == hohe_var3:
482.                             mylist_weite = [21.3, 22.3, 22.0, 21.7]
483.                         elif mainWin.comboBox_hohe.currentText() == hohe_var4:
484.                             mylist_weite = [22.3, 22.5, 23.0, 22.0]
485.                 # Volley Training
486.                 elif (mainWin.comboBox_training.currentText() == training_var4) or (m
    ainWin.comboBox_training.currentText() == training_var5) or (mainWin.comboBox_traini
    ng.currentText() == training_var6) :
487.                     # Spin erwünscht?
488.                     if mainWin.comboBox_spin.currentText() == spin_var1:
489.                         if mainWin.comboBox_hohe.currentText() == hohe_var1:
490.                             mylist_weite = [17.3, 17.5, 17.0, 16.8]
491.                         elif mainWin.comboBox_hohe.currentText() == hohe_var2:
492.                             mylist_weite = [18.0, 17.5, 17.7, 18.2]

493.                         elif mainWin.comboBox_hohe.currentText() == hohe_var3:
494.                             mylist_weite = [17.7, 17.0, 17.4, 18.0]
495.                         elif mainWin.comboBox_hohe.currentText() == hohe_var4:
496.                             mylist_weite = [16.3, 16.5, 16.0, 15.8]
497.                     elif mainWin.comboBox_spin.currentText() == spin_var2:
498.                         if mainWin.comboBox_hohe.currentText() == hohe_var1:
499.                             mylist_weite = [16.5, 16.2, 16.0, 16.7]
500.                         elif mainWin.comboBox_hohe.currentText() == hohe_var2:
501.                             mylist_weite = [18.0, 17.2, 17.0, 16.7]

502.                         elif mainWin.comboBox_hohe.currentText() == hohe_var3:
503.                             mylist_weite = [17.7, 17.0, 17.3, 16.8]
504.                         elif mainWin.comboBox_hohe.currentText() == hohe_var4:
505.                             mylist_weite = [16.3, 17.0, 17.5, 16.8]
506.                 PWM_oben, PWM_unten, v_return = geschw_berechnen(mylist_weite[0],hohe
    _eingabe,spin_eingabe,lambda_Ab) # Funktion aufrufen
507.                 # Schafft der Ball die Netzhöhe?
508.                 if (PWM_oben == 0 and PWM_unten == 0):
509.                     clickNetz() # Aufruf der Funktion
510.                     verbindung_motoren.close() # Verbindung zum Arduino unterbrechen

511.                     clickStopEnabled() # Aufruf der Funktion
512.                 else:
513.                     clickPos() # Aufruf der Funktion
514.                     changePMW(verbindung_motoren, str(PWM_oben) + "#" + str(PWM_unten
    ) + "%"+ str(winkel) + "$" + "0") #Funktion aufrufen
515.                     messageboxVerbindungMotor = TimerMessageBox(20, messageBox_verbin
    dung_mot, messageBox_schlMot1, messageBox_schlMot2, mainWin)
516.                     messageboxVerbindungMotor.exec_() # Messagebox aufrufen
517.
518.                     # Training Grundlinie bzw. Volley?
519.                     if (mainWin.comboBox_training.currentText() == training_var3) or
    (mainWin.comboBox_training.currentText() == training_var6):
520.                         gpio.setmode(gpio.BCM)
521.                         gpio.setwarnings(False)
522.
```

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

```
523.                         def callback_func(channel):
524.                             if gpio.input(channel) == gpio.LOW:
525.                                 if hilfsvar == 1:
526.                                     PWM_oben, PWM_unten, v_return = geschw_berechnen(
    random.choice(mylist_weite),hohe_eingabe,spin_eingabe,lambda_Ab) # Funktion aufrufen
527.                                     changePMW(verbindung_motoren, str(PWM_oben) + "#"
     + str(PWM_unten) + "%"+ str(random.choice(mylist_winkel)) + "$" + str(modus)) # Fun
    ktion aufrufen
528.
529.                         gpio.setup(LED_PIN, gpio.IN, pull_up_down = gpio.PUD_DOWN)  #
     LED PIN als Eingang definieren
530.                         gpio.add_event_detect(LED_PIN, gpio.FALLING, callback=callbac
    k_func, bouncetime=200)
531.
532.                     changePMW(verbindung_motoren, str(PWM_oben) + "#" + str(PWM_unten
    ) + "%"+ "100" + "$" + str(modus)) # Funktion aufrufen
533.
534.         # Klick auf den Start-
    Button beim Unterprogramm manuelles Training
535.         def clickStart_2():
536.             # Bereich ausgewählt?
537.             if mainWin.comboBox_bereich.currentText() == auswahl_var:
538.                 clickFehl(messageBox_Bereich)
539.             # Spin ausgewählt?
540.             elif mainWin.comboBox_spin_2.currentText() == auswahl_var:
541.                 clickFehl(messageBox_Spin)
542.
543.             # Zuweisung des Falles
544.             else:
545.                 clickStartEnabled_2() # Aufruf der Funktion
546.                 # Steht die Verbindung zum Arduino schon?
547.                 if(verbindung_motoren.isOpen()==False):
548.                     verbindung_motoren.open() # Verbindung zum Arduino aufbauen
549.                 messageboxVerbindung = TimerMessageBox(3, messageBox_verbindung, mess
    ageBox_schliessen1, messageBox_schliessen2, mainWin)
550.                 messageboxVerbindung.exec_() # Messagebox aufrufen
551.                 # Spin erwünscht?
552.                 if mainWin.comboBox_spin_2.currentText() == spin_var1:
553.                     spin_eingabe = 0 # kein Spin
554.                 elif mainWin.comboBox_spin_2.currentText() == spin_var2:
555.                     spin_eingabe = 1 # Top Spin
556.                 #Rechts- oder Linkshänder?
557.                 if hand == 1:
558.                     # horizontale Winkeleinstellung
559.                     if mainWin.comboBox_bereich.currentText() == bereich_var1:
560.                         winkel = 0
561.                     elif mainWin.comboBox_bereich.currentText() == bereich_var2:
562.                         winkel = 15
563.                     elif mainWin.comboBox_bereich.currentText() == bereich_var3:
564.                         winkel = 30
565.                 elif hand == 2:
566.                     if mainWin.comboBox_bereich.currentText() == bereich_var1:
567.                         winkel = 30
568.                     elif mainWin.comboBox_bereich.currentText() == bereich_var2:
569.                         winkel = 15
570.                     elif mainWin.comboBox_bereich.currentText() == bereich_var3:
571.                         winkel = 0
572.                 modus = 2 # Fortgeschritten
573.                 hohe_eingabe = 25 # vertikale Winkeleinstellung
574.                 weite = mainWin.doubleSpinBox_weite.value() # gewünschte Weite
575.                 PWM_oben, PWM_unten, v_return = geschw_berechnen(weite,hohe_eingabe,s
    pin_eingabe,lambda_Ab) # Aufruf der Funktion
576.                 # Schafft der Ball die Netzhöhe?
577.                 if (PWM_oben == 0 and PWM_unten == 0):
578.                     clickNetz() # Aufruf der Funktion
```

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

```
579.                    verbindung_motoren.close() # Verbindung zum Arduino unterbrechen

580.                    clickStopEnabled_2() # Aufruf der Funktion
581.                else:
582.                    dialogPos.label.setText("Position 3")
583.                    clickPos() # Aufruf der Funktion
584.                    changePMW(verbindung_motoren, str(PWM_oben) + "#" + str(PWM_unten
    ) + "%"+ str(winkel) + "$" + "0") # Aufruf der Funktion
585.                    messageboxVerbindungMotor = TimerMessageBox(20, messageBox_verbin
    dung_mot, messageBox_schlMot1, messageBox_schlMot2, mainWin)
586.                    messageboxVerbindungMotor.exec_() # Messagebox aufrufen
587.                    changePMW(verbindung_motoren, str(PWM_oben) + "#" + str(PWM_unten
    ) + "%"+ "100" + "$" + str(modus)) # Aufruf der Funktion
588.
589.        # Klick auf den Start-Button beim Dialog-Fenster Kalibrieren
590.        def clickStart_3():
591.            global weite,hohe_eingabe,spin_eingabe, v_return
592.            # vertikale Winkeleinstellung ausgewählt?
593.            if dialogKali.comboBox_hohe.currentText() == auswahl_var:
594.                clickFehl(messageBox_Hohe)
595.            # Zuweisung des Falles
596.            else:
597.                clickStartEnabled_3() # Aufruf der Funktion
598.                # Steht die Verbindung zum Arduino schon?
599.                if(verbindung_motoren.isOpen()==False):
600.                    verbindung_motoren.open() # Verbindung zum Arduino aufbauen
601.                messageboxVerbindung = TimerMessageBox(3, messageBox_verbindung, mess
    ageBox_schliessen1, messageBox_schliessen2, dialogKali)
602.                messageboxVerbindung.exec_() # Messagebox aufrufen
603.                # vertikale Winkeleinstellung
604.                if dialogKali.comboBox_hohe.currentText() == hohe_var1:
605.                    hohe_eingabe = 15
606.                    dialogPos.label.setText("Position 1")
607.                elif dialogKali.comboBox_hohe.currentText() == hohe_var2:
608.                    hohe_eingabe = 20
609.                    dialogPos.label.setText("Position 1")
610.                elif dialogKali.comboBox_hohe.currentText() == hohe_var3:
611.                    hohe_eingabe = 25
612.                    dialogPos.label.setText("Position 1")
613.                elif dialogKali.comboBox_hohe.currentText() == hohe_var4:
614.                    hohe_eingabe = 30
615.                    dialogPos.label.setText("Position 1")
616.
617.                spin_eingabe = 0 # kein Spin
618.                modus = 2 # Fortgeschritten
619.                weite = 17.3  # Flugweite zur T - Linie
620.                winkel = 15 # vertikale Winkeleinstellung
621.                lambda_Ab = 1 # Korrekturfaktor
622.                PWM_oben, PWM_unten, v_return = geschw_berechnen(weite,hohe_eingabe,s
    pin_eingabe,lambda_Ab) # Aufruf der Funktion
623.                # Schafft der Ball die Netzhöhe?
624.                if (PWM_oben == 0 and PWM_unten == 0):
625.                    clickNetz() # Aufruf der Funktion
626.                    verbindung_motoren.close() # Verbindung zum Arduino unterbrechen

627.                    clickStopEnabled_3() # Aufruf der Funktion
628.                else:
629.                    changePMW(verbindung_motoren, str(PWM_oben) + "#" + str(PWM_unten
    ) + "%"+ str(winkel) + "$" + "0") # Aufruf der Funktion
630.                    messageboxVerbindungMotor = TimerMessageBox(20, messageBox_verbin
    dung_mot, messageBox_schlMot1, messageBox_schlMot2, dialogKali)
631.                    messageboxVerbindungMotor.exec_() # Messagebox aufrufen
632.                    changePMW(verbindung_motoren, str(PWM_oben) + "#" + str(PWM_unten
    ) + "%"+ "100" + "$" + str(modus)) # Aufruf der Funktion
633.
634.        # Klick auf den Stop-Button beim Unterprogramm Trainingsprogramme
635.        def clickStop():
```

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

```
636.            global hilfsvar
637.            # Training Grundlinie bzw. Volley?
638.            if hilfsvar == 1:
639.                gpio.cleanup() # LED-PIN deaktivieren
640.            hilfsvar = 0
641.            clickStopEnabled() # Aufruf der Funktion
642.            messageboxVerbindungAbb = TimerMessageBox(3, messageBox_verbindung_abb, m
    essageBox_schliessen1, messageBox_schliessen2, mainWin)
643.            messageboxVerbindungAbb.exec_() # Messagebox aufrufen
644.            changePMW(verbindung_motoren, "0#0%15$0") # Aufruf der Funktion
645.            verbindung_motoren.close() # Verbindung zum Arduino unterbrechen
646.
647.        # Klick auf den Stop-Button beim Unterprogramm manuelles Training
648.        def clickStop_2():
649.            clickStopEnabled_2() # Aufruf der Funktion
650.            messageboxVerbindungAbb = TimerMessageBox(3, messageBox_verbindung_abb, m
    essageBox_schliessen1, messageBox_schliessen2, mainWin)
651.            messageboxVerbindungAbb.exec_() # Messagebox aufrufen
652.            changePMW(verbindung_motoren, "0#0%15$0") # Aufruf der Funktion
653.            verbindung_motoren.close() # Verbindung zum Arduino unterbrechen
654.
655.        # Klick auf den Stop-Button beim Dialog-Fenster Kalibrieren
656.        def clickStop_3():
657.            clickStopEnabled_3() # Aufruf der Funktion
658.            messageboxVerbindungAbb = TimerMessageBox(3, messageBox_verbindung_abb, m
    essageBox_schliessen1, messageBox_schliessen2, dialogKali)
659.            messageboxVerbindungAbb.exec_() # Messagebox aufrufen
660.            changePMW(verbindung_motoren, "0#0%15$0") # Aufruf der Funktion
661.            verbindung_motoren.close() # Verbindung zum Arduino unterbrechen
662.
663.        #Beim Trainingsprogramm ist Auswahl ausgewählt
664.        def training_Auswahl():
665.            mainWin.label_tennisplatz.setPixmap(gui.QPixmap("Bilder-250-
    375/tennisplatz.jpg")) # Bild anpassen
666.
667.        #Beim Trainingsprogramm ist Grundlinie Vorhand ausgewählt
668.        def training_GrundVor():
669.            mainWin.label_tennisplatz.setPixmap(gui.QPixmap("Bilder-250-
    375/tennisplatz_grundVor.jpg")) # Bild anpassen
670.
671.        #Beim Trainingsprogramm ist Grundlinie Rückhand ausgewählt
672.        def training_GrundRueck():
673.            mainWin.label_tennisplatz.setPixmap(gui.QPixmap("Bilder-250-
    375/tennisplatz_grundRueck.jpg")) # Bild anpassen
674.
675.        #Beim Trainingsprogramm ist Grundlinie ausgewählt
676.        def training_Grund():
677.            mainWin.label_tennisplatz.setPixmap(gui.QPixmap("Bilder-250-
    375/tennisplatz_grund.jpg")) # Bild anpassen
678.
679.        #Beim Trainingsprogramm ist Volley Vorhand ausgewählt
680.        def training_VollVor():
681.            mainWin.label_tennisplatz.setPixmap(gui.QPixmap("Bilder-250-
    375/tennisplatz_vollVor.jpg")) # Bild anpassen
682.
683.        #Beim Trainingsprogramm ist Volley Rückhand ausgewählt
684.        def training_VollRueck():
685.            mainWin.label_tennisplatz.setPixmap(gui.QPixmap("Bilder-250-
    375/tennisplatz_vollRueck.jpg")) # Bild anpassen
686.
687.        #Beim Trainingsprogramm ist Volley ausgewählt
688.        def training_Voll():
689.            mainWin.label_tennisplatz.setPixmap(gui.QPixmap("Bilder-250-
    375/tennisplatz_voll.jpg")) # Bild anpassen
690.
691.        #Beim Bereich ist Auswahl ausgewählt
692.        def bereich_Auswahl():
```

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

```
693.            mainWin.label_tennisplatz_2.setPixmap(gui.QPixmap("Bilder-250-
    375/tennisplatz.jpg")) # Bild anpassen
694.
695.        #Beim Bereich ist Volley Vorhand ausgewählt
696.        def bereich_vor():
697.            mainWin.label_tennisplatz_2.setPixmap(gui.QPixmap("Bilder-250-
    375/tab_2/tennisplatz_vor.jpg")) # Bild anpassen
698.
699.        #Beim Bereich ist Volley Rückhand ausgewählt
700.        def bereich_mitte():
701.            mainWin.label_tennisplatz_2.setPixmap(gui.QPixmap("Bilder-250-
    375/tab_2/tennisplatz_mitte.jpg")) # Bild anpassen
702.
703.        #Beim Bereich ist Volley ausgewählt
704.        def bereich_rueck():
705.            mainWin.label_tennisplatz_2.setPixmap(gui.QPixmap("Bilder-250-
    375/tab_2/tennisplatz_rueck.jpg")) # Bild anpassen
706.
707.        # Bildänderung bei der Änderung der Trainingsart
708.        def bildAendern(argument):
709.            # Rechts- oder Linkshänder?
710.            if hand == 1:
711.                bild_options_r[argument]()
712.            elif hand == 2:
713.                bild_options_l[argument]()
714.
715.        # Bildänderung bei der Änderung des Bereichs
716.        def bildAendern_2(argument):
717.            # Rechts- oder Linkshänder?
718.            if hand == 1:
719.                bild_options_2_r[argument]()
720.            elif hand == 2:
721.                bild_options_2_l[argument]()
722.
723.        # Options-
    Variable für die Auswahl des Bildes für Rechtshänder (Trainingsprogramm)
724.        bild_options_r = {auswahl_var : training_Auswahl,
725.                    training_var1 : training_GrundVor,
726.                    training_var2 : training_GrundRueck,
727.                    training_var3 : training_Grund,
728.                    training_var4 : training_VollVor,
729.                    training_var5 : training_VollRueck,
730.                    training_var6 : training_Voll
731.                   }
732.
733.        # Options-
    Variable für die Auswahl des Bildes für Linkshänder (Trainingsprogramm)
734.        bild_options_l = {auswahl_var : training_Auswahl,
735.                    training_var1 : training_GrundRueck,
736.                    training_var2 : training_GrundVor,
737.                    training_var3 : training_Grund,
738.                    training_var4 : training_VollRueck,
739.                    training_var5 : training_VollVor,
740.                    training_var6 : training_Voll
741.                   }
742.
743.        # Options-
    Variable für die Auswahl des Bildes für Rechtshänder (manuelles Training)
744.        bild_options_2_r = {auswahl_var : bereich_Auswahl,
745.                    bereich_var1 : bereich_vor,
746.                    bereich_var2 : bereich_mitte,
747.                    bereich_var3 : bereich_rueck,
748.                   }
749.
750.        #Options-
    Variable für die Auswahl des Bildes für Linkshänder (manuelles Training)
751.        bild_options_2_l = {auswahl_var : bereich_Auswahl,
```

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

```
752.                 bereich_var1 : bereich_rueck,
753.                 bereich_var2 : bereich_mitte,
754.                 bereich_var3 : bereich_vor,
755.                }
756.
757.     # Abhängigkeiten programmieren
758.     mainWin.pushButton_kali.clicked.connect(clickKalibrieren)
759.     mainWin.pushButton_kali_2.clicked.connect(clickKalibrieren)
760.     mainWin.actionKalibrieren.triggered.connect(clickKalibrieren)
761.     mainWin.actionHandigkeit.triggered.connect(clickHandigkeit)
762.     mainWin.actionLaser_einstellen.triggered.connect(clickLaser)
763.     mainWin.actionInfo.triggered.connect(clickInfo)
764.     mainWin.comboBox_training.currentTextChanged['QString'].connect(bildAendern)
765.     mainWin.comboBox_bereich.currentTextChanged['QString'].connect(bildAendern_2)
766.     mainWin.pushButton_start.clicked.connect(clickStart)
767.     mainWin.pushButton_start_2.clicked.connect(clickStart_2)
768.     dialogKali.pushButton_start_3.clicked.connect(clickStart_3)
769.     mainWin.pushButton_stop.clicked.connect(clickStop)
770.     mainWin.pushButton_stop_2.clicked.connect(clickStop_2)
771.     dialogKali.pushButton_stop_3.clicked.connect(clickStop_3)
772.
773.     # STOP-Button deaktivieren
774.     mainWin.pushButton_stop.setEnabled(False)
775.     mainWin.pushButton_stop_2.setEnabled(False)
776.     dialogKali.pushButton_stop_3.setEnabled(False)
777.
778.     # ButtonBox beim Dialog Kalibrieren deaktivieren
779.     dialogKali.buttonBox.setEnabled(False)
780.
781.     # ComboBox Training initialisieren
782.     mainWin.comboBox_training.addItem(auswahl_var)
783.     mainWin.comboBox_training.addItem(training_var1)
784.     mainWin.comboBox_training.addItem(training_var2)
785.     mainWin.comboBox_training.addItem(training_var3)
786.     mainWin.comboBox_training.addItem(training_var4)
787.     mainWin.comboBox_training.addItem(training_var5)
788.     mainWin.comboBox_training.addItem(training_var6)
789.
790.     # ComboBox Höhe des Balles initialisieren
791.     mainWin.comboBox_hohe.addItem(auswahl_var)
792.     mainWin.comboBox_hohe.addItem(hohe_var1)
793.     mainWin.comboBox_hohe.addItem(hohe_var2)
794.     mainWin.comboBox_hohe.addItem(hohe_var3)
795.     mainWin.comboBox_hohe.addItem(hohe_var4)
796.     mainWin.comboBox_hohe_2.addItem(hohe_var3)
797.     dialogKali.comboBox_hohe.addItem(auswahl_var)
798.     dialogKali.comboBox_hohe.addItem(hohe_var1)
799.     dialogKali.comboBox_hohe.addItem(hohe_var2)
800.     dialogKali.comboBox_hohe.addItem(hohe_var3)
801.     dialogKali.comboBox_hohe.addItem(hohe_var4)
802.
803.     # ComboBox Spin initialisieren
804.     mainWin.comboBox_spin.addItem(auswahl_var)
805.     mainWin.comboBox_spin.addItem(spin_var1)
806.     mainWin.comboBox_spin.addItem(spin_var2)
807.     mainWin.comboBox_spin_2.addItem(auswahl_var)
808.     mainWin.comboBox_spin_2.addItem(spin_var1)
809.     mainWin.comboBox_spin_2.addItem(spin_var2)
810.
811.     # ComboBox Bereich initialisieren
812.     mainWin.comboBox_bereich.addItem(auswahl_var)
813.     mainWin.comboBox_bereich.addItem(bereich_var1)
814.     mainWin.comboBox_bereich.addItem(bereich_var2)
815.     mainWin.comboBox_bereich.addItem(bereich_var3)
816.
```

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

```
817.        #Text im Label im Dialog-Fenster Kalibrieren festlegen
818.        dialogKali.label.setText(" Damit dieses Programm ordnungsgemäß funktioniert,
    muss \n die Maschine auf die verwendeten Bälle kalibriert werden. Bitte \n legen Sie
     dazu bis zu drei Bälle in die Maschine und betätigen \n den Button START. Nach Absc
    huss der Kalibrierungsbälle \n drücken Sie den Button STOP und geben die gemittelte
    Ab-
    \n weichung der Flugweite zu der T - Linie ein und bestätigen \n dies mit einem Klic
    k auf OK.")
819.
820.        mainWin.show() # Hauptprogramm starten
821.        sys.exit( app.exec_() )
```

Code (Arduino)

```
//Pakete inkludieren
#include <Wire.h>
#include <Keypad.h>

//PINs zuweisen
#define hallPin 2
#define pwm_1 3
#define dir_3 4
#define enPin 5
#define pwm_3 6
#define stepPin 7
#define dir_1 8
#define dirPin 9
#define pwm_2 11
#define dir_2 13

#define Buffer_Length 8

int schritt = 0; //Hilfsvariable für Schrittmotor
int pwm_motor3 = 0; //Hilfsvariable für Getriebemotor

char Data[Buffer_Length]; //empfangene Daten
byte data_count = 0; //Zeichenzähler

void setup(){
  Serial.begin(9600);
  //Zuweisung der PINS
  pinMode(pwm_1,OUTPUT);
  pinMode(dir_1,OUTPUT);
  pinMode(pwm_2,OUTPUT);
  pinMode(dir_2,OUTPUT);
  pinMode(pwm_3,OUTPUT);
  pinMode(dir_3,OUTPUT);
  pinMode(hallPin, INPUT);
  pinMode(stepPin,OUTPUT);
  pinMode(dirPin,OUTPUT);
  pinMode(enPin,OUTPUT);
  digitalWrite(enPin,LOW);
}

void loop(){

if (Serial.available())
  serialEvent(); //Aufruf der Funktion

}
```

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

```
void serialEvent() { //serielle Daten werden empfangen
  byte ch = Serial.read(); //Zeichen einlesen
  Data[data_count] = ch; //auf String abspeichern
  data_count++;
  if (ch == 35) { //Prüfen ob das Zeichen ein # ist
    motor1_einstell(); //Funktion aufrufen
  }
  if (ch == 37) { //Prüfen ob das Zeichen ein % ist
    motor2_einstell(); //Funktion aufrufen
  }
  if (ch == 36) { //Prüfen ob das Zeichen ein $ ist
    schritt_einstell(); //Funktion aufrufen
  }
  if (ch == 42) { //Eingabe mit * beenden
    motor3_einstell(); //Funktion aufrufen
  }
}

void motor1_einstell() //Drehzahl für Motor 1 einstellen
{
    Data[data_count] = '\0'; //String ende
    uint16_t zahl = atoi(Data); //String in Int umwandeln

    digitalWrite(dir_1,LOW); //Richtung für den Motor 1 geben
    analogWrite(pwm_1,zahl); //Drehzahl als PWM dem Motor 1 geben

    Serial.print(" ");  //Return an den Raspberry PI
    Serial.println(zahl);

    clearData();
}

void motor2_einstell() //Drehzahl für Motor 2 einstellen
{
    Data[data_count] = '\0'; //String ende
    uint16_t zahl = atoi(Data); //String in Int umwandeln

    digitalWrite(dir_2,LOW); //Richtung für den Motor 2 geben
    analogWrite(pwm_2,zahl); //Drehzahl als PWM dem Motor 2 geben

    clearData();
}

void schritt_einstell() //empfangene Daten verarbeiten
{
    Data[data_count] = '\0'; //String ende
    uint16_t zahl = atoi(Data); //String in Int umwandeln
    if (zahl != 100) //Schrittmotor bewegen?
    {
      if (zahl == 0) { //Vorhand-Ecke
        schritt = 80;
      }
      else if (zahl == 15) { //Mitte
        schritt = 360;
      }
      else if (zahl == 30) { //Rückhand-Ecke
        schritt = 640;
      }
      while(digitalRead(hallPin) == 0) //bis der Hall-Sensor Magnet erkennt
      {
        digitalWrite(dirPin,HIGH); //Richtung einstellen
        digitalWrite(stepPin,HIGH); //einen Schritt bewegen
        delayMicroseconds(1500); //Pause zwischen den Schritten
```

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub

```
        digitalWrite(stepPin,LOW); //Schritt vollenden
        delayMicroseconds(1500); //Pause zwischen den Schritten
      }
      delay(200); //0,2 Sekunden warten, bevor der Motor zurück fährt

      for(int x = 0; x < schritt; x++) //Motor auf Position fahren
      {
        digitalWrite(dirPin,LOW); //Richtung einstellen
        digitalWrite(stepPin,HIGH); //einen Schritt bewegen
        delayMicroseconds(1500); //Pause zwischen den Schritten
        digitalWrite(stepPin,LOW); //Schritt vollenden
        delayMicroseconds(1500); //Pause zwischen den Schritten
      }
      clearData();
    }
    else {
      clearData();
    }
}

void motor3_einstell() //Drehzahl für Getriebemotor einstellen
{
    Data[data_count] = '\0'; //String ende
    uint16_t zahl = atoi(Data);  //String in Int umwandeln

    if (zahl == 0) { //Motor stoppen
      pwm_motor3 = 0;
    }
    else if (zahl == 1) { //Modus Anfänger
      pwm_motor3 = 120;
    }
    else if (zahl == 2) { //Modus Fortgeschritten
      pwm_motor3 = 180;
    }

    digitalWrite(dir_3,HIGH); //Richtung für den Getriebemotor geben
    analogWrite(pwm_3,pwm_motor3); //Drehzahl als PWM dem Getriebemotor
geben

    clearData();
}

void clearData(){ //String leeren
while(data_count !=0){
Data[data_count--] = 0;
}
return;
}
```

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN