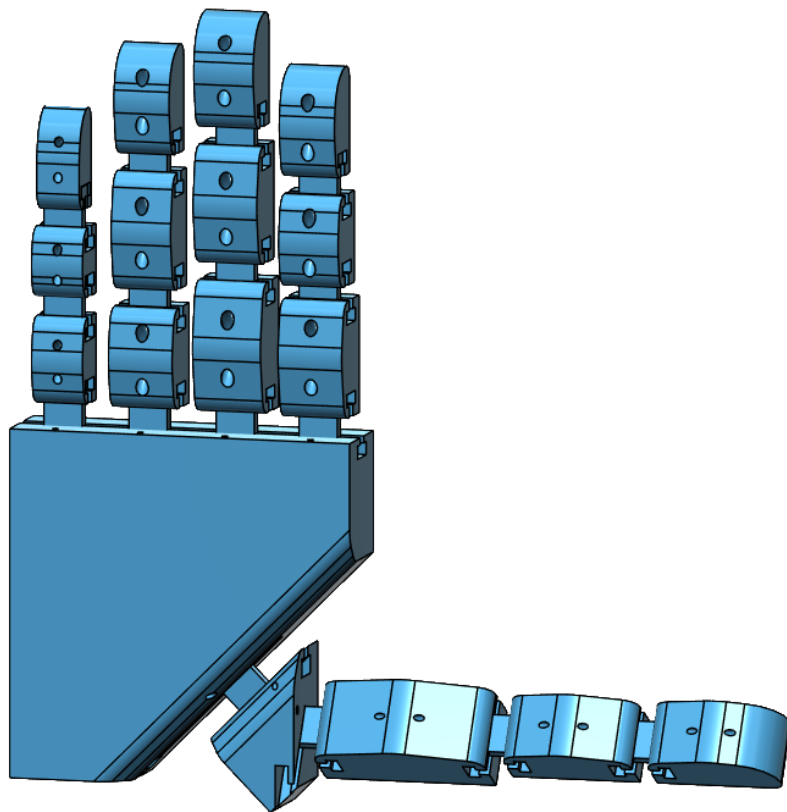

Entwicklung und Herstellung einer computergesteuerten Roboterhand mit Hilfe des 3D-Druckes

Maturitätsarbeit von Ivan Robuschi

Klasse 4G, Kantonsschule Im Lee Winterthur

Betreuung: Wolfgang Pils

04/01/2021



Inhaltsverzeichnis

1 - Vorwort	3
2 - Abstract	4
3 - Die Herstellung der Roboterhand	5
3.1 - Kurze Einführung zum 3D-Druck.....	5
3.2 - Messungen am Vorbild	8
3.3 - Die Fingerglieder.....	9
3.4 - Die Verbindungen zwischen den Fingergliedern	14
3.5 - Die Handfläche	19
3.6 - Die Motorisierung.....	25
4 - Die Steuerung	30
4.1 - Die Bedienung der Roboterhand	30
4.2 - Der Microcontroller „Arduino“	31
4.3 - Die Servomotoren.....	31
4.4 - Der Joystick.....	32
4.5 - Die Steckplatine	33
4.6 - Der Schaltkreis.....	35
4.7 - Der Code.....	37
5 - Zusammenbau, Materialkosten und Auswertung.....	43
5.1 - Zusammenbau	43
5.2 – Materialkosten.....	44
5.3 - Auswertung	45
6 - Quellen und Abbildungsverzeichnis	47
6.1 - Quellen.....	47
6.2 - Abbildungsverzeichnis	48

1 - Vorwort

In diesem ersten Kapitel möchte ich meine Inspiration erläutern, die mich zur Themenwahl der Herstellung einer Roboterhand durch den 3D-Druck, welche mit dem Microcontroller «Arduino» gesteuert wird, geführt hat.

Seit Jahren beschäftige ich mich in meiner Freizeit sowohl mit dem 3D-Drucker als auch mit dem Microcontroller «Arduino»: Ich habe mit dem ersteren zahlreiche Objekte (zum Beispiel mein Bleistifthalter) hergestellt und mit dem zweiten verschiedene Projekte realisiert (ein Beispiel davon ist die Steuerung meiner Modellbahnanlage). Das Interesse dafür habe ich von meinem Vater, einem professionellen Softwareentwickler mit einer Ausbildung im Bereich der Elektrotechnik.

Vor mehr als fünf Jahren, als ich diese beiden Objekte noch kaum kannte, hat ein Bekannter meines Vaters mit mir über seine Arbeit geredet: Er arbeitete nämlich in der Forschung und hatte eine Roboterhand hergestellt. Ich habe die Details dieser Unterhaltung längst vergessen, aber erinnere mich, dass ich wirklich daran interessiert war. Nach diesem Gespräch ist diese Idee hängen geblieben: Ich wollte auch eine solche Hand herstellen.

Damals hatte ich weder die Erfahrung noch die theoretischen Kenntnisse, welche für ein solches Projekt nötig sind. Mittlerweile habe ich aber durch meinen Vater und meine Mutter, welche in der Vergangenheit auch als Informatikerin gearbeitet hat, ein grundlegendes Amateurwissen in den Bereichen der Programmierung, des 3D-Druckes und der Elektrotechnik erlangt.

Die Maturaarbeit stellte für mich die perfekte Gelegenheit dar, diesen Traum zu realisieren und somit ein Element aus meiner Bucket List zu streichen.

Ein herzliches Dankeschön an meinem Vater, dem ich die Leidenschaft für Maschinenbau zu verdanken habe, meiner Mutter, die mich motiviert hat, mit dem Programmieren anzufangen, und Herrn Wolfgang Pils, der mich mit grosser Professionalität und Sorgfalt bei meiner Maturitätsarbeit betreut und begleitet hat.

2 - Abstract

Das Ziel dieses Projektes war es, mit Hilfe des 3D-Druckes eine durch einen Arduino-Microcontroller gesteuerte Roboterhand herzustellen, welche Objekte fassen kann. Da sie von einem menschlichen Vorbild inspiriert war, musste sie das entsprechende Aussehen besitzen.

Die Idee wurde wie folgt umgesetzt: Fingerglieder und Handfläche wurden aus starrem Kunststoff (PLA) gedruckt und durch aus elastischem Kunststoff hergestellte Verbinder miteinander verbunden. Durch jeden Finger wurde ein Stück Angelschnur gezogen, welches einerseits am Ende des Fingers und andererseits an einer Fadenspule in der Handfläche gebunden ist. Jede Fadenspule ist an einem an der Handfläche aufgeschraubten Servomotor fixiert. Indem man durch den Arduino die Servomotoren drehen lässt, wickelt sich die Angelschnur um die Fadenspulen. Die Finger werden dadurch gebogen, wie in der Abbildung 1. Sobald man die Servomotoren in die andere Richtung drehen lässt, kehren die Finger wegen der elastischen Eigenschaften der Verbinder zur originalen Position zurück. Die Roboterhand lässt sich mit Hilfe eines Joysticks steuern.

Es wurde am Ende des Projektes gezeigt, dass die Hand tatsächlich Objekte mit einer Masse von bis zu 310g halten kann. Obwohl sie nicht so stark wie ursprünglich erwartet ist, ist das Ziel trotzdem erreicht.

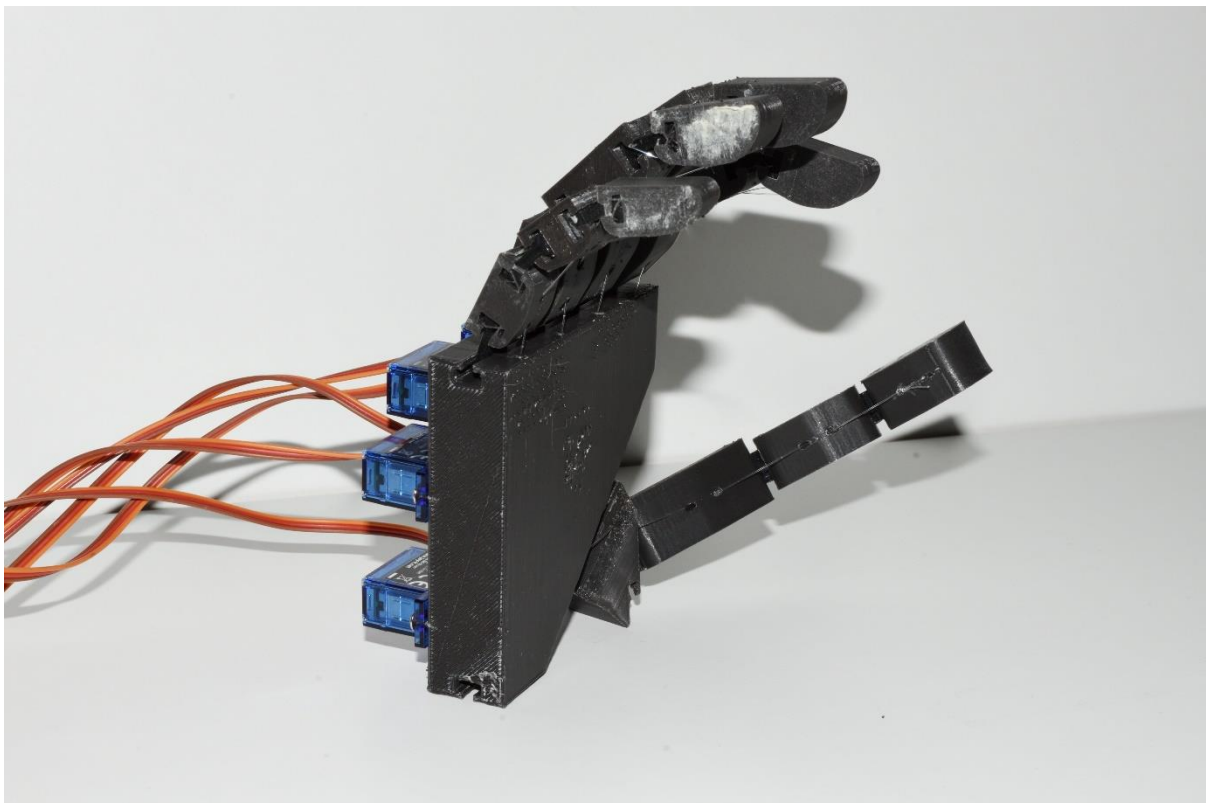


Abbildung 1 - Die Roboterhand mit gebogenen Fingern

3 - Die Herstellung der Roboterhand

3.1 - Kurze Einführung zum 3D-Druck

In diesem Kapitel werden die Grundlagen des 3D-Druckes und die dazugehörigen in dieser Arbeit verwendeten Fachbegriffe erklärt.

Der 3D-Druck ist ein Prozess, wobei durch eine geeignete computergesteuerte Maschine (den 3D-Drucker) Objekte aus verschiedenen Materialien hergestellt werden.

Der im Rahmen dieser Arbeit angewendete 3D-Drucker, ein Wanhao Duplicator 4S (Vgl. Abb. 2), funktioniert mit dem Verfahren des «Fused Deposition Modeling»¹: Das Material, anfangs ein festes Filament mit einem Durchmesser von 1.75mm (vgl. Abbildung 3), wird durch einen beheizten Teil des Druckers, den Extruder, geleitet und aufgeschmolzen. Der Extruder ist in X- und Y-Richtung (das heisst, auf einer zum Boden parallelen Ebene) beweglich und deponiert Schicht für Schicht das zähflüssige Material auf eine in Z-Richtung (senkrecht zum Boden) bewegliche Arbeitsplatte, sodass sich das hergestellte Objekt am Ende des Druckvorganges sich auf dieser Platte befindet.

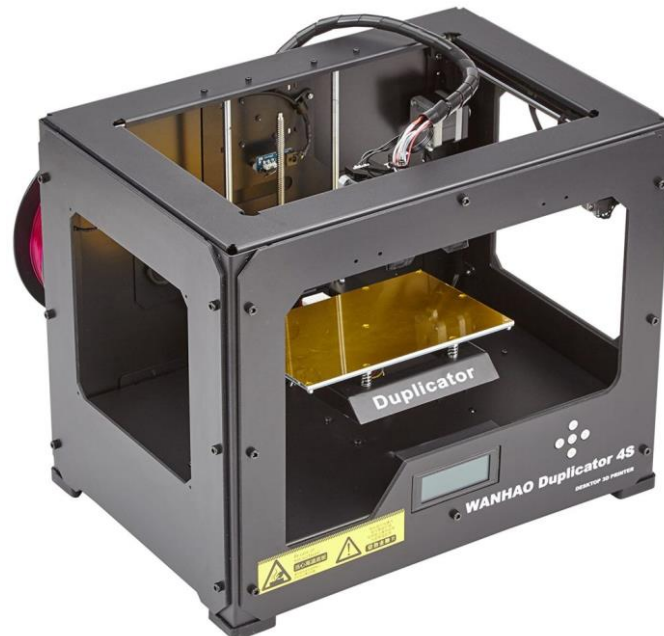


Abbildung 2 - Ein Wanhao Duplicator 4S²

¹ <https://www.printer-care.de/de/drucker-ratgeber/wie-funktioniert-ein-3d-drucker#:~:text=Definition%20des%203D%2DDruckers,ein%20dreidimensionaler%20Gegenstand%20erzeugt%20wird.&text=Heute%20werden%203D%2DDrucker%20vor,Industrie%20oder%20in%20Forschungseinrichtung en%20verwendet.> (Stand 13/12/2020)

² <https://www.my3d.guru/product/wanhao-duplicator-4s-3d-drucker-test-und-erfahrungsberichte/> (Stand 13/12/2020)



Abbildung 3 - Filament aus PLA für den 3D-Drucker³

Bei der Herstellung eines Objektes wird folgendermassen vorgegangen: Zuerst zeichnet man ein 3D-Modell des Objektes mit einer geeigneten Software (in diesem Fall wurde Autodesk 123D Design verwendet). Dieses Modell wird erstmals im für diese Software spezifischen Dateiformat gespeichert. In dieser Phase kann das Modell jederzeit bearbeitet werden. Wenn das Muster fertig ist, wird dieses zu einer anderen, nicht softwarespezifischen Datei kompiliert, die von den weiteren Programmen gelesen werden kann. Sehr häufig, wie auch in diesem Fall, ist dieses Format «.stl» («Standard Triangulation Language»). Bei STL-Dateien wird die Form des Modelles durch Dreiecke approximiert (die Genauigkeit ist einstellbar: Je mehr Dreiecke verwendet werden, desto genauer ist die Approximation). So kann näherungsweise jede beliebige Form auf sehr einfache Weise und in relativ kleinen Dateien gespeichert werden (vgl. Abbildung 4).

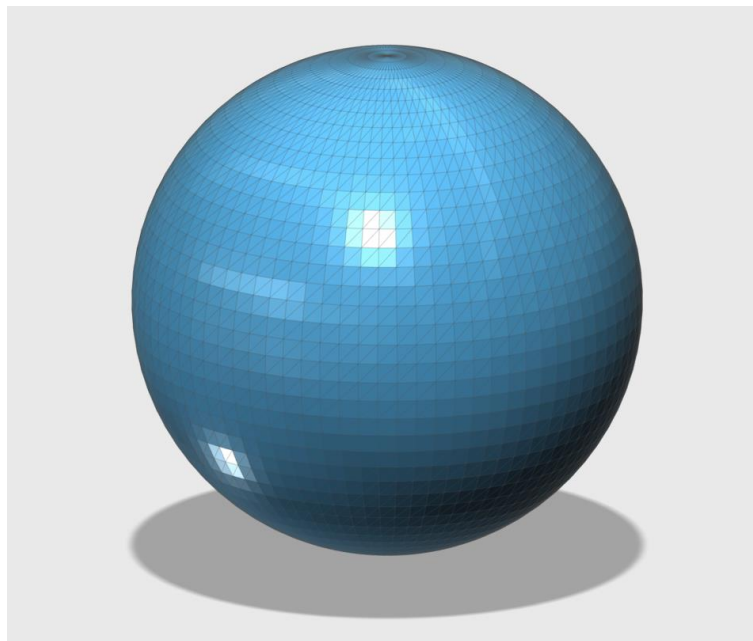


Abbildung 4 - Eine Kugel im STL-Format

³ <https://www.conrad.ch/de/p/filament-renkforce-pla-1-75-mm-schwarz-1-kg-1528349.html> (Stand 13/12/2020)

Die STL-Datei wird mit einer zweiten Software geöffnet: Dem Slicer (in diesem Fall ist dieser die Software «Simplify3D»). Diese Anwendung hat als Input ein 3D-Modell und als Output eine druckerspezifische Datei, mit deren Hilfe der 3D-Drucker das Objekt herstellen kann. Durch dieses Programm wird alles Entscheidende für den Druck eingestellt. Die wichtigsten Parameter, die man bei den allermeisten Slicern findet, sind die folgenden:

- Drucktemperatur: Temperatur des Extruders während des Druckvorgangs. Sie muss selbstverständlich grösser als die Schmelztemperatur des jeweils angewandten Kunststoffes sein.
- Temperatur der Plattform: Einstellbare Temperatur der Plattform, auf welcher gedruckt wird.
- Infill Percentage: In den allermeisten Fällen besteht das Innere 3D-gedruckter Objekte nur zu einem relativ kleinen Anteil aus Kunststoff, der Rest ist Luft. So spart man viel Material und Zeit. Dieser Anteil, der sogenannte Infill Percentage (auch «Infill»), ist einstellbar und wird in Prozent des Volumens angegeben.
- Druckgeschwindigkeit: Geschwindigkeit, mit der der Extruder sich während des Drucks bewegt. Sie wird in mm/min angegeben.
- X/Y Axis Movement Speed: Geschwindigkeit, mit der der Extruder sich bewegt, wenn er nicht druckt. Ebenfalls wird sie in mm/min angegeben.

Diese Parameter beeinflussen die Qualität des Endproduktes sehr stark und müssen sorgfältig eingestellt werden.

Die Drucktemperatur und die Temperatur der Plattform sind während des ganzen Projektes konstant geblieben. Die erste blieb bei 185° sowohl beim elastischen Material (vgl. Kapitel 3.4) als auch beim für die starren Teile verwendeten PLA, die zweite bei 40°C. Die anderen Einstellungen wurden während der ganzen Entwicklung der Roboterhand mindestens einmal geändert.

Wichtig ist, es zu wissen, dass es sogar möglich ist, Objekte mit vorstehenden Teilen zu drucken. Das erfolgt über das sogenannte «Support Material» (Stützmaterial): Der Slicer fügt wo nötig Stützstrukturen hinzu, auf welche das zähflüssige Material deponiert werden kann. Diese Strukturen müssen nachträglich entfernt werden. Wenn nötig können solche Strukturen bei Maschinen mit mehreren Extrudern aus speziellen Materialien hergestellt werden, die gewisse Eigenschaften besitzen. Beispielsweise ist das sehr verbreitete Stützmaterial «PVA» wasserlöslich, somit kann es durch einfaches Tauchen in Wasser entfernt werden. Bei dieser Arbeit bestanden diese Strukturen immer aus demselben Material wie das Produkt selbst: Das Stützmaterial befand sich in leicht von Hand oder mit einfachen Werkzeugen erreichbaren Orten, sodass die Anwendung von löslichem Material nicht nötig war. Materialien wie PVA sind sehr teuer im Vergleich zu PLA, somit wurden hier Kosten erspart.

3.2 - Messungen am Vorbild

Um die 3D-Modelle zu zeichnen wurde zuerst eine echte Hand gemessen. In der untenstehenden Tabelle sind die Masse des Vorbilds zu sehen. Alle Breiten der Fingerglieder wurden mit einer Lehre und alle anderen Längen mit einem Massstab gemessen.

Es wurde entschieden, bei dieser Arbeit eine rechte Hand herzustellen, jedoch kann auch das entsprechende linke Gegenstück durch die hier vorgestellten Muster hergestellt werden, indem man die asymmetrischen Modelle vor dem Druckvorgang spiegelt.

	Daumen	Zeigefinger	Mittelfinger	Ringfinger	Kleiner Finger
3. Fingerglied (Fingerspitze)	Länge: 33mm Breite: 15mm	Länge: 25mm Breite: 13mm	Länge: 26mm Breite: 14mm	Länge: 25mm Breite: 13mm	Länge: 24mm Breite: 10mm
2. Fingerglied	Länge: 30mm Breite: 15mm	Länge: 21mm Breite: 14mm	Länge: 27mm Breite: 15mm	Länge: 27mm Breite: 14mm	Länge: 16mm Breite: 12mm
1. Fingerglied	Länge: 40mm Breite: -	Länge: 27mm Breite: 15mm	Länge: 30mm Breite: 16mm	Länge: 23mm Breite: 15mm	Länge: 20mm Breite: 12mm

Die Handfläche wurde durch ein Quadrat mit einer Kantenlänge von 8cm approximiert.

Diese Masse entsprechen nicht unbedingt denjenigen der später hergestellten 3D-Modelle: Sie wurden jeweils angepasst, um eine bessere Funktion des Produktes zu ermöglichen. Sie wurden jedoch bei jedem Entwurf von 3D-Zeichnungen berücksichtigt, damit das Endprodukt von den Massen her nicht zu realitätsfern aussah.

Wichtig ist es zu merken, dass bei der Messung die Sektion der Finger als quadratisch approximiert worden ist. Es wird also angenommen, dass die Dicke jedes Fingergliedes der Breite desselben ungefähr gleich ist. Diese Entscheidung wurde getroffen, weil es zu diesem Zeitpunkt bereits bekannt war, dass jede Dicke später aufgrund der Funktionsweise der Hand angepasst werden müssen hätte. Die Dicke der Handfläche wurde ausnahmsweise nicht gemessen: Die Räume für die Servomotoren bedingten eine Mindestdicke von 2.0cm, sodass sie angenommen wurde.

Um genauere Längen bei den Fingergliedern zu erhalten, wurden diese von der Seite gemessen: Die Trennlinien sind dort am eindeutigsten erkennbar. Zu diesem Zweck wurde die linke Seite (vom Rücken der Hand betrachtet) gewählt.

Der Messfehler beträgt bei den Fingergliedern 1mm und bei der Handfläche 1cm (es ist viel schwieriger, die Grenzen der Handfläche eindeutig zu definieren).

3.3 - Die Fingerglieder

In diesem Kapitel geht es um die Entwicklung der Fingerglieder. Sie wurden, wie auch die Handfläche, aus dem Bio-Kunststoff „PLA“ (Polylactid) gedruckt. PLA ist einer der am meisten für den 3D-Druck verwendeten Kunststoffe und hat zwei grosse Vorteile:

- Es fällt sehr leicht, mit PLA zu arbeiten, da die allermeisten Drucker für dieses Material gedacht sind (zum Beispiel ist der Schmelzpunkt von allen Druckern leicht erreichbar);
- Es gibt keine toxischen Dämpfe während des Druckes, was es ermöglicht, den Drucker in einem Wohnzimmer ohne besondere Schutzmassnahmen zu behalten.

Die Idee hinter der Funktionsweise der Finger ist es, dass die starren Fingerglieder mit elastischen Verbindern miteinander verbunden sind (die T-förmigen Löcher in der Abbildung 5 sind die Sitze dafür) und durch jeden Finger ein Stück Angelschnur gezogen wird, durch dessen Anspannung der Finger gebogen werden kann. Zur geraden Position sollte er dank der elastischen Verbinder zurückkehren.

Versuchswise wurde anfangs ein Finger hergestellt, dessen Fingerglieder dieselbe Grösse haben wie die durchschnittliche Breite von allen Fingern einer menschlichen Hand (Daumen nicht eingeschlossen). Dies wurde gemacht, um eventuelle Probleme sofort zu erkennen. In den folgenden Bildern ist dieser Versuch dargestellt.

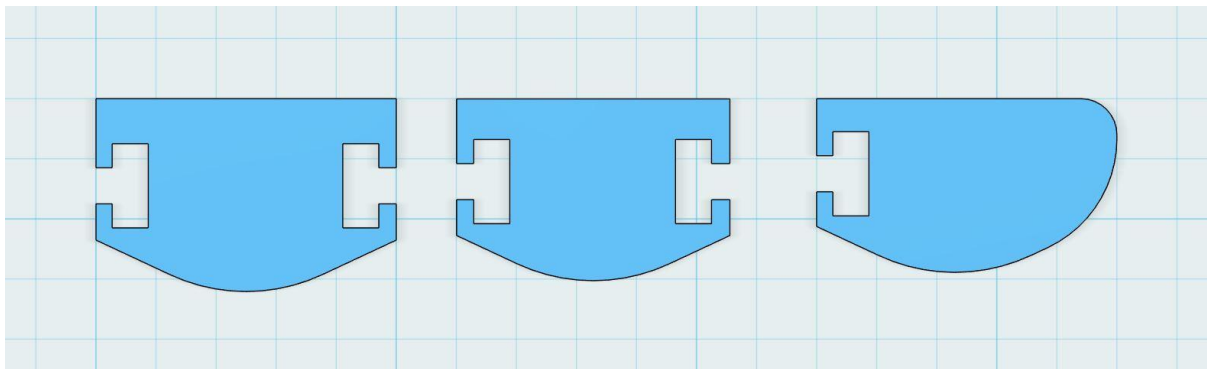


Abbildung 5 - Das Modell des ersten Fingers von oben (1 Häuschen = 5mm)

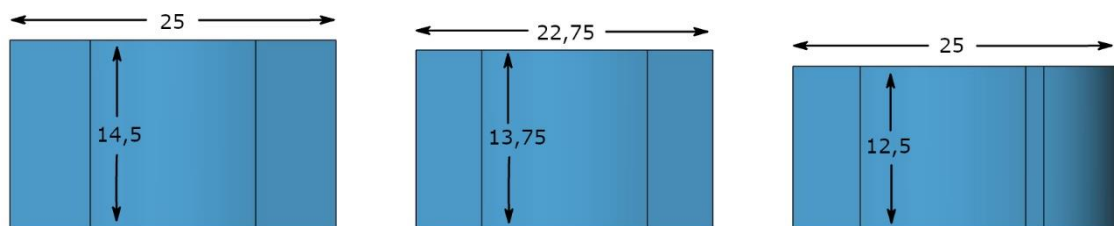


Abbildung 6 - Das Modell des ersten Fingers von der Seite (alle Angaben in mm)

Der erste Finger wurde mit einem Infill von 10% gedruckt, jedoch war der Abstand zwischen den inneren Stützstrukturen zu gross. Das hat dazu geführt, dass die obersten Schichten (diejenigen, die die Oberfläche einer Seite des Fingers bilden sollen hätten) nicht bester Qualität waren: Das zähflüssige Filament konnte während des Druckes aufgrund der mangelnden Stützung die vorgeschriebene Form nicht behalten und ist oft gestürzt, darum ist es zu ungewünschten Löchern im Finger gekommen (vgl. Abbildung 7).



Abbildung 7 - Löcher, welche auf der Oberfläche bei einem zu kleinen Infill entstehen

Das Problem wurde gelöst, indem der Infill auf 20% gesetzt wurde. Leider hat diese Wahl das Gewicht des Produktes, den Materialaufwand sowie die Dauer des Druckvorgangs beeinträchtigt, jedoch wurde nach diesem Versuch diese Einstellung für alle starren Teile aufgrund der besseren Qualität übernommen.

Die anderen Druckeinstellungen blieben konstant: Die Druckgeschwindigkeit war 3600mm/min und die X/Y-Geschwindigkeit (vgl. Kapitel 3.1) 4200mm/min. Die Temperaturen des Extruders und der Plattform sind im Kapitel 3.1 angegeben.

Nach diesem Experiment wurde in einem zweiten Modell die Form der Fingerglieder verbessert: Die Ecken, die in der Abbildung 5 auf der unteren Seite dargestellt sind, wurden gerundet, da sie sich ansonsten während des Biegens als Hindernisse erweisen würden. Um dies zu ermöglichen, wurden die Sitze der Verbinder nach hinten verschoben. Das zweite Modell sieht von oben wie in der Abb. 8 und von der Seite wie in der Abb. 9 aus. Das Modell wurde gedruckt, mit Verbindern ausgestattet (vgl. Kapitel 3.4) und erfolgreich durch Biegen getestet.

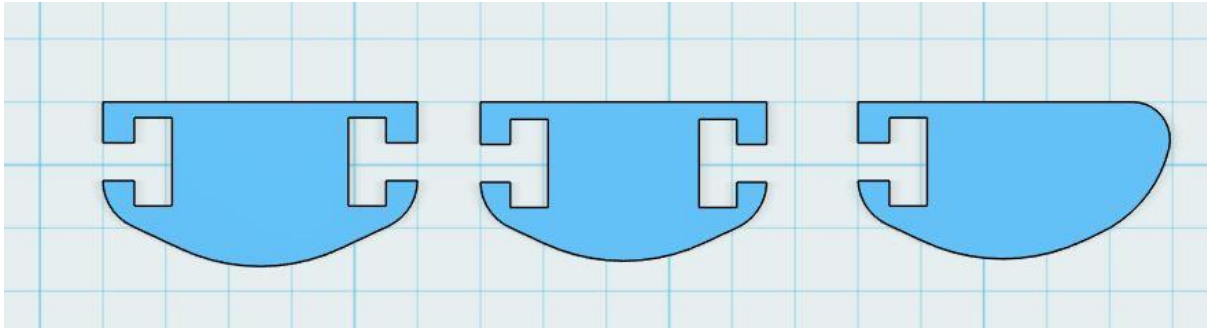


Abbildung 8 - Das zweite Modell von oben (1 Häuschen = 5mm)

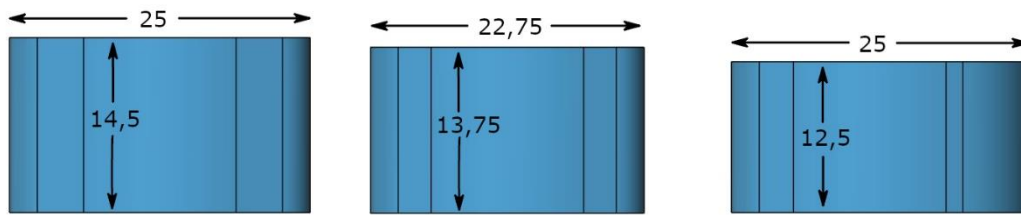


Abbildung 9 - Das zweite Modell von der Seite (alle Angaben in mm)

Nach diesem Versuch wurden alle Finger modelliert und erfolgreich gedruckt. In den Abbildungen 10 und 11 sind deren technische Zeichnungen zu finden.

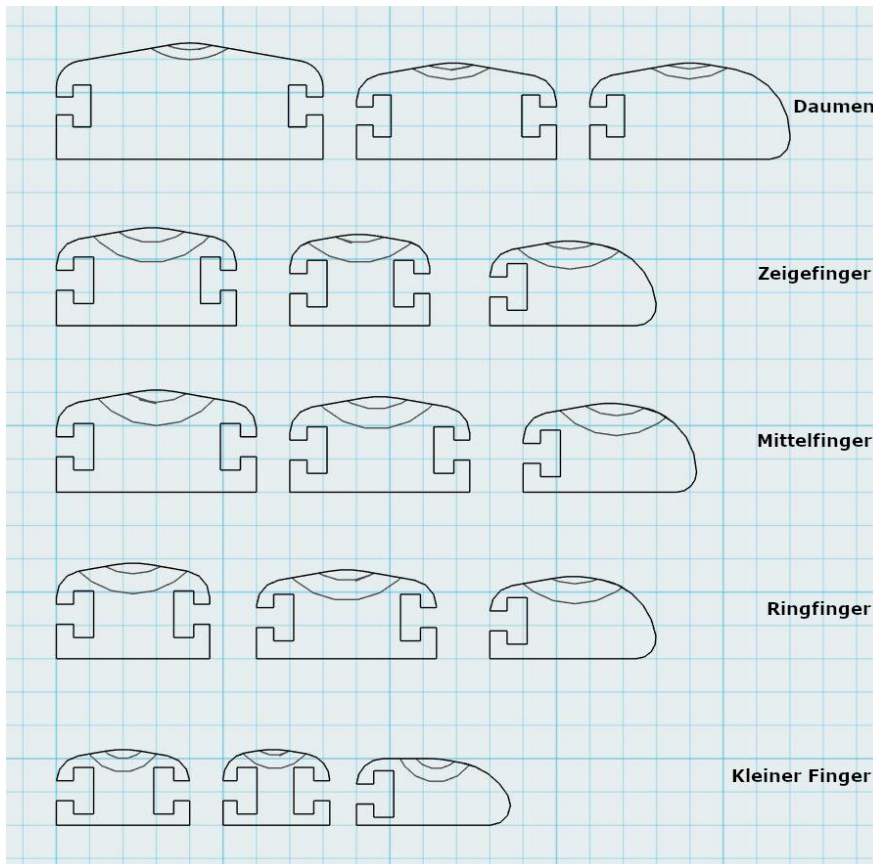


Abbildung 10 - Die definitiven Fingerglieder von der Seite (1 Häuschen = 5mm)

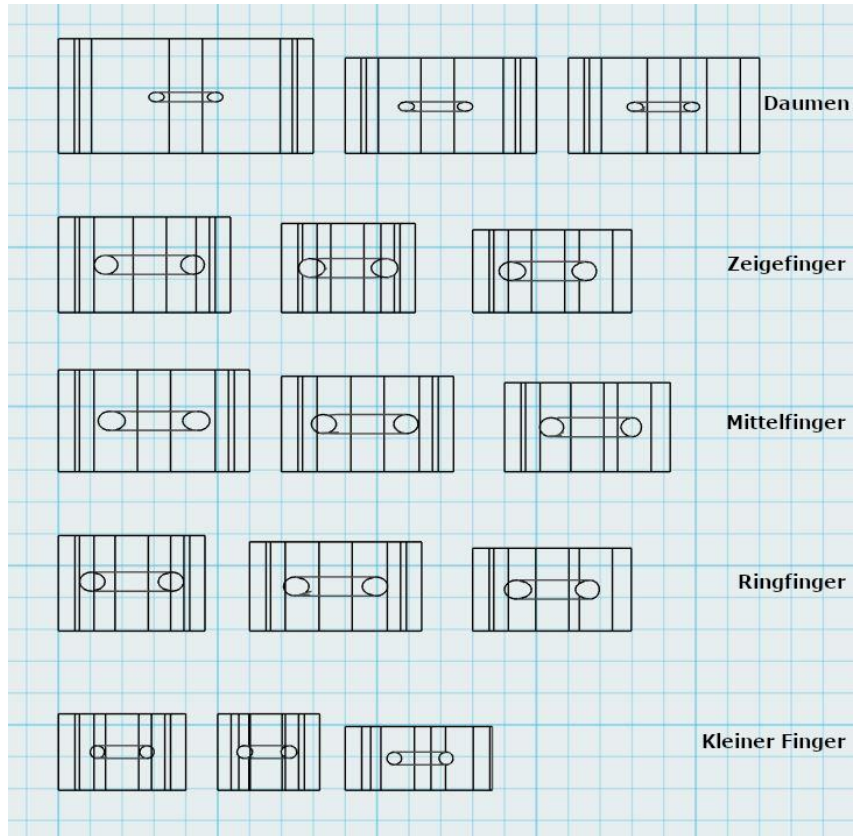


Abbildung 11 - Die definitiven Fingerglieder von oben (die gewölbte Seite ist in dieser Zeichnung nach unten gerichtet)

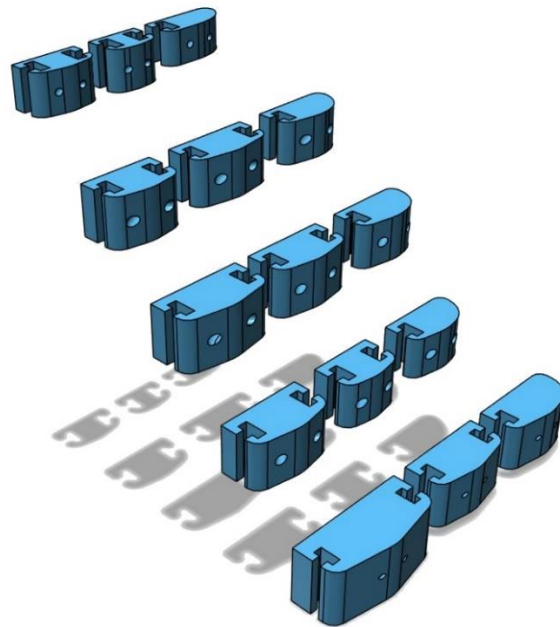


Abbildung 12 - 3D-Bild der definitiven Finger

Um den Durchgang der später angewandten Angelschnur zu ermöglichen, sind die entsprechenden Löcher schon in der Phase der 3D-Zeichnung modelliert worden (vgl. Abbildung 12). Es wurde erstmals versucht, die Fingerglieder ohne Löcher zu drucken, um sie nachträglich mit der Bohrmaschine zu bohren, jedoch gab es mit dieser Lösung zwei wesentliche Probleme: Die Oberfläche wurde durch die Bohrung beschädigt und es musste auch durch die inneren Stützstrukturen gebohrt werden, was einerseits schwierig umzusetzen war und andererseits es sehr kompliziert machte, die Schnur durch die Fingerglieder zu ziehen. Da der Finger im Inneren nicht komplett gefüllt ist, kann der Faden schnell in der inneren Struktur stecken bleiben.

Diese Röhren sind aus dem folgenden Grund gekrümmt: Wenn sie gerade gewesen wären, wäre die Angelschnur näher an den Verbindern gewesen. Wegen des Hebelgesetzes hätte es eine grössere Kraft gebraucht, um den Finger zu biegen. Dies hätte zwei Effekte verursacht: Einerseits wäre der Griff der Hand viel schwächer gewesen, andererseits hätten die Motoren unter einem grösseren Verschleiss aufgrund der höheren Belastung gelitten. Dies kann besser mit Hilfe der Abbildung 13 verstanden werden.

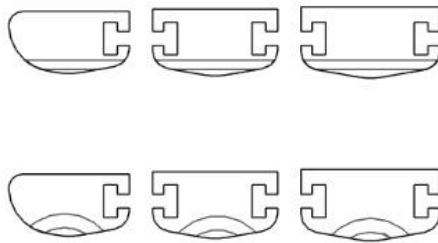


Abbildung 13 - Vergleich zwischen geraden und gekrümmten Röhren

Nach dieser letzten Änderung sind die Finger bis zum Ende des Projektes unverändert geblieben.

3.4 - Die Verbindungen zwischen den Fingergliedern

Das ganze Funktionsprinzip der Hand basiert sich auf elastische Verbinder zwischen den Fingergliedern, die es den Fingern erlauben, gebogen zu werden und in die gerade Position zurückzukehren sobald die jeweiligen Schnüre von den Motoren von der Spule abgerollt werden (vgl. Kapitel 2). In diesem Kapitel geht es um den Vorgang bei der Entwicklung und Herstellung der Verbinder.

Zuerst wurde nach einem passenden Filament gesucht. Die Bestellung des Filaments über Internet wäre aufgrund der laufenden Pandemie zu riskant gewesen, weil die allermeisten Online-Händler sich im Ausland befinden und Versände über die Grenzen von einem Tag zum anderen verzögert oder sogar blockiert werden können. Um Verspätungen zu vermeiden, musste darum in einem physischen Laden gesucht werden. Das einzige sofort verfügbare und für den vorhandenen Drucker passende elastische Filament (das heisst, mit einem Durchmesser von 1,75mm und einer Drucktemperatur tiefer als 220°C) stammt von Renkforce (auf dem Markt wird es derzeit als „Renkforce Elastic“ verkauft).

Nachher musste ein erster Verbinder skizziert werden. Vom Anfang an war es erfahrungsgemäss schon klar, dass kleine Teile wie dieser wegen des Problems vom sog. Overheating (mehr dazu unten) ziemlich schwierig zu drucken sein und somit mehrere Versuche und Anpassungen brauchen würden. Um Zeit und Material zu sparen, wurde der erste Verbinder kürzer als der definitive Teil gemacht. Kann das Überhitzen bei einem kleineren Prototyp überwunden werden, so ist die Qualität des grösseren Teils (das heisst, des definitiven, längeren Verbinders) noch höher als diejenige des Ersten.

Die allgemeine Form des Verbinders ist wie in der Abbildung 14, wobei jedes Ende in einem passenden Loch in jedem Fingerglied stecken bleibt.

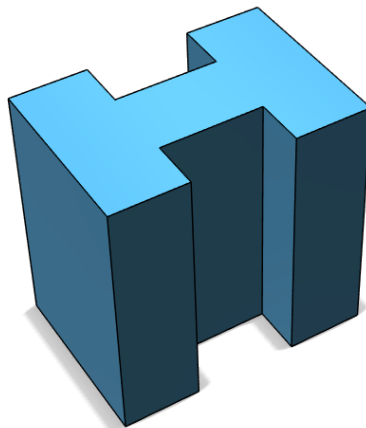


Abbildung 14 - Die Form des Verbinders

In den Abbildungen 15 und 16 sind die Messungen des ersten Modelles von oben und von der Seite dargestellt.

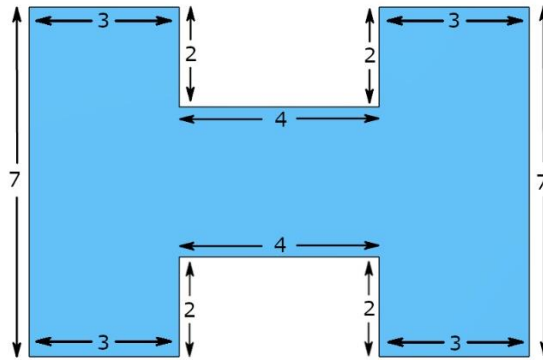


Abbildung 15 - Die erste Version des Verbinders von oben betrachtet

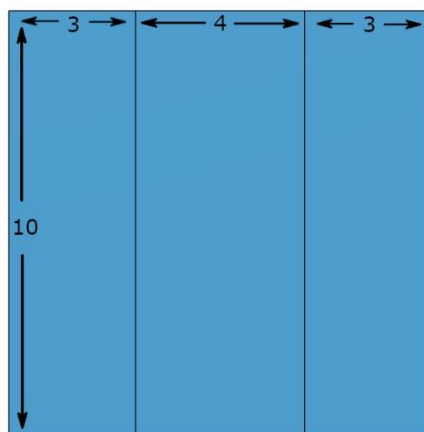


Abbildung 16 - Die erste Version des Verbinders von der Seite betrachtet

Beim ersten Versuch wurde nach der passenden Grösse gesucht, damit jeder Verbinder richtig in den entsprechenden Löchern zusammenpasst. Die obigen Messungen entsprechen genau denjenigen der Löcher, der Verbinder darf also nur idealerweise genauso gross sein. In der Realität muss er ein wenig kleiner sein. Es wurden die Verbinder und nicht die Löcher angepasst, weil der Druck eines Fingerglieders zeitaufwändiger ist als derjenige eines Verbinders. Um die richtige Grösse herausfinden zu können, wurde erstmals den folgenden Versuch gemacht: Es wurden mit den Faktoren 1, 0,9, 0,8, 0,7, 0,6 und 0,5 skalierte Verbinder gedruckt und jeweils zwischen zwei Fingergliedern gesteckt. Das 3D-Modell sah wie in der Abbildung 17 aus.

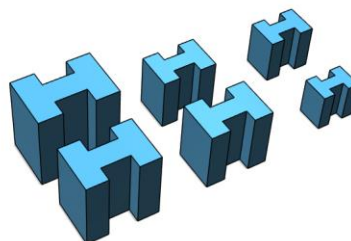


Abbildung 17 - 3D-Modell der mit verschiedenen Faktoren skalierten Verbinder

Beim Druck des Modells tauchte, wie vorhergesehen, das Problem des Überhitzens tatsächlich auf (vgl. Abbildung 18): Die Teile sind so klein, dass sich während des Druckvorgangs die jeweils zuletzt gedruckte Schicht nicht genug abkühlt, bevor die nächste darauf gedruckt wird. Demzufolge werden die noch nicht komplett verfestigten Schichten vom Extruder mitgezogen, was schlussendlich zu einer sehr schlechten Qualität des Produktes führt. Bei grösseren Teilen gibt es dieses Problem meistens nicht.

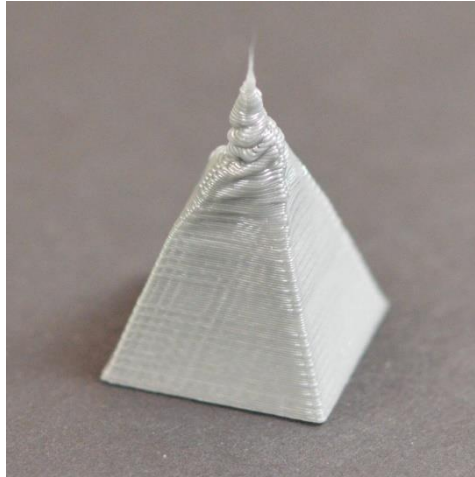


Abbildung 18 - Beispiel von Overheating⁴

Das so gedruckte erste Modell war somit unbrauchbar. Der oben beschriebene Test konnte nicht durchgeführt werden, bis das Problem des Überhitzens gelöst worden war. Um direkt den Druckvorgang beeinflussen zu können, kann man auf Ebene des Slicers (vgl. Kapitel 3.1) eingreifen und die Einstellungen des Drucks ändern.

Erfahrungsgemäss haben zwei Parameter die grössten Auswirkungen auf das Phänomen des Überhitzens: Die Drucktemperatur und die Extrudergeschwindigkeit. Eine Senkung der Drucktemperatur bedeutet, dass sich die Schichten weniger abkühlen müssen, um sich zu verfestigen, und darum weniger Zeit dazu brauchen. Eine Senkung der Extrudergeschwindigkeit bringt zwei Folgen mit sich: Einerseits haben die unteren Schichten mehr Zeit zur Abkühlung und andererseits wird der eventuell noch nicht komplett verfestigte Kunststoff weniger vom langsameren Extruder mitgezogen. Die beiden aufgelisteten Änderungen wirken sich also auf das Überhitzen reduzierend aus.

Die Drucktemperatur war schon beim ersten Versuch bei 185°C eingestellt und konnte nicht gesenkt werden: Die vom Hersteller angegebene Drucktemperatur war 180 bis 220°C und eine weitere Senkung dieses Parameters hätte zu einer ungenügenden Verschmelzung des Materials während des Druckvorgangs und dadurch zu stärkerem Verschleiss der Maschine oder sogar Schäden führen können. Der sinnvollste Eingriff war also, in diesem Fall, die Extrudergeschwindigkeit zu verändern. Beim ersten Versuch war diese 3600mm/min, beim zweiten wurde sie jedoch auf 1800mm/min verringert. Zudem wurde die X/Y-Geschwindigkeit (vgl. Kapitel 3.1) auf 2100mm/min halbiert. Dies hat sofort die Qualität der Teile erhöht, sodass sie brauchbar waren. Obwohl die Effekte des Überhitzens nach dieser Änderung noch sichtbar waren, war eine weitere Senkung der Extrudergeschwindigkeit unnötig, da diese die gesamte Herstellung noch langsamer machen und keinen funktionellen Vorteil mehr bringen würde.

⁴ <https://www.simplify3d.com/support/print-quality-troubleshooting/overheating/#:~:text=The%20most%20common%20cause%20for,not%20being%20cooled%20fast%20enough.&text=If%20your%20printer%20includes%20a,and%20selecting%20the%20Cooling%20tab.> (Stand 20/12/2020)

Mit den so erhaltenen Teilen konnte der oben erklärte Test endlich durchgeführt werden. Der nicht skalierte Verbinder war zu gross, um überhaupt in die Löcher passen zu können, und der mit dem Faktor 0,8 skalierte Teil war ein wenig zu klein und konnte sehr einfach von den Fingergliedern rausgeschoben werden, was zur Instabilität der ganzen Hand hätte führen können. Der mit dem Faktor 0,9 skalierte Verbinder war allerdings der perfekte Kompromiss.

Nun waren die Verbinder mit voller Länge zu zeichnen und mit Berücksichtigung der Ergebnisse der vorherigen Versuche zu drucken (die bisher gedruckten Teile waren viel zu kurz und hielten die Fingerglieder so nah aneinander, dass sich diese gegenseitig fast berührten und somit die Biegung des Fingers unmöglich machten). In den Abbildungen 19 und 20 sind die technischen Zeichnungen der definitiven Version zu sehen.

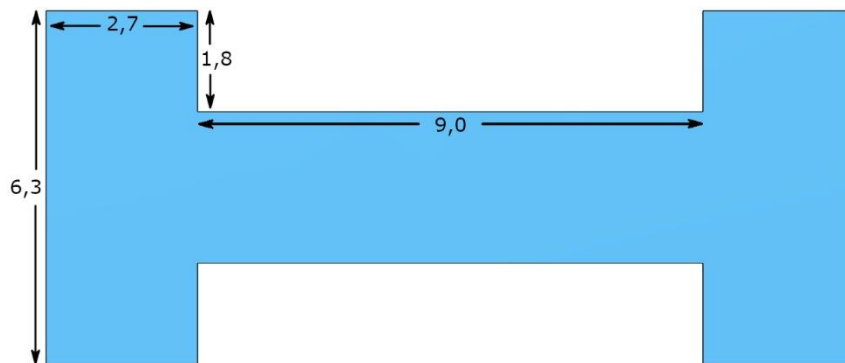


Abbildung 19 - Der definitive Verbinder von oben betrachtet (das Teil besitzt eine vertikale Symmetrieachse in der Mitte).

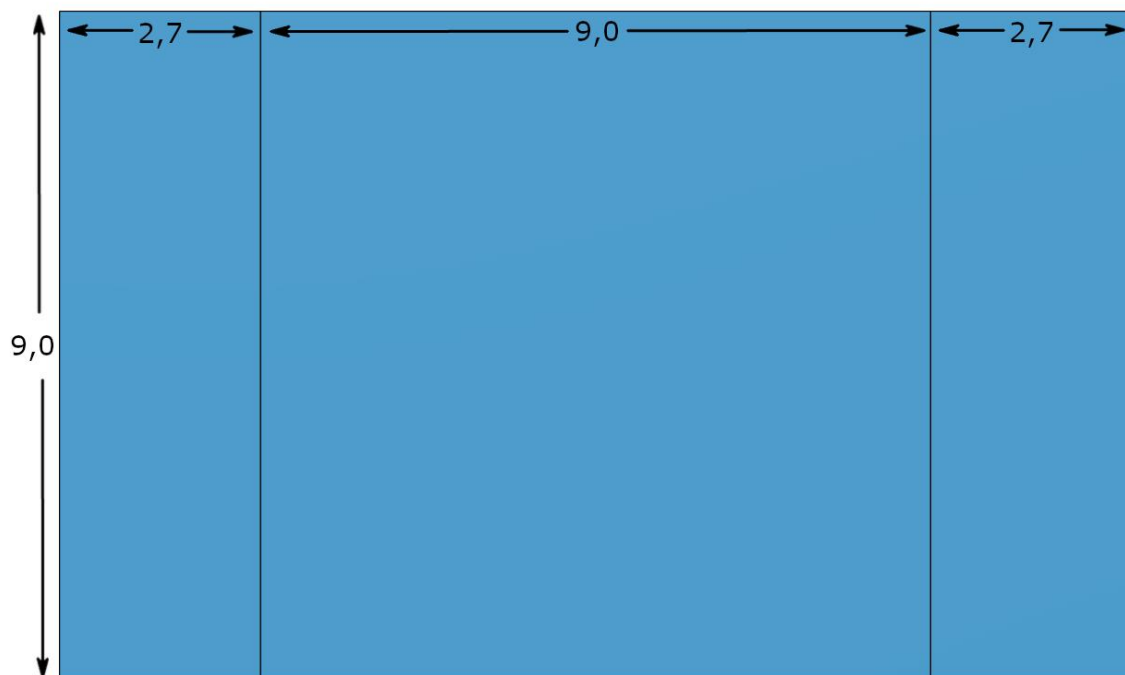


Abbildung 20 - Der definitive Verbinder von der Seite betrachtet

Vier dieser Teile wurden gedruckt (das soll sicherstellen, dass die Qualität des Drucks konsistent bleibt) und der Test-Fit mit den Fingern lief einwandfrei. Es bleibt nur noch ein Aspekt, der noch stark verbessert werden konnte: Diese Teile waren viel zu starr und unnötigerweise schwer zu biegen. Das lag daran, dass das Innere der Teile zu 30% aus einem Skelett aus Kunststoff bestand. Dieses Skelett bewirkte einen sehr starken Widerstand bei der Biegung des Verbinders. Um später Probleme wie ein zu starker Verschleiss oder eine ungenügende Kraft bei den Motoren zu vermeiden, musste dieses Problem unmittelbar gelöst werden. Der zu ändernde Parameter beim Slicer ist der sogenannte „Infill“ (der Bruchteil in Prozent des Volumens des Inneren des Teils, der aus Kunststoff besteht). Dieser wurde erstmals von 30% auf 20% gesenkt, danach weiter auf 10%. Der mit 20% Infill gedruckte Verbinder war viel elastischer als der Vorherige, jedoch immer noch ein wenig zu starr. Im Gegensatz dazu ist der mit 10% Infill gedruckte Teil ziemlich leicht zu biegen, allerdings besitzt dieser aufgrund der automatisch vom Slicer generierten Struktur des Skeletts (vgl. Abbildung 21) zwei wichtige Schwachstellen, wo sich der Verbinder viel zu leicht biegen und verformen lässt. Da der Teil in diesen besonderen Stellen in Kombination mit dem Fingerglied aufgrund der starren Plastik des letzteren nicht gebogen werden kann, stellen diese Schwachstellen kein Problem dar. Der Infill wurde jedoch bei den weiteren Druckvorgängen nicht weiter gesenkt, um die Entstehung noch grösserer Schwachstellen in den Verbindern zu vermeiden.

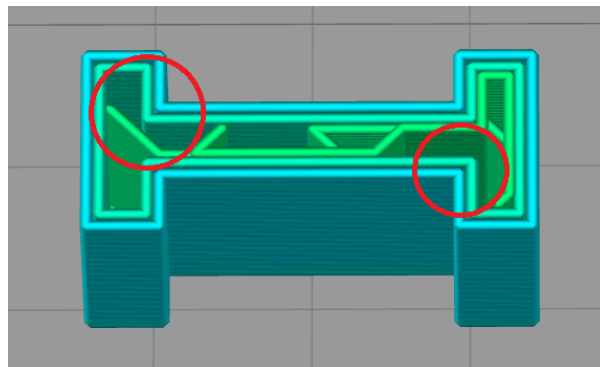


Abbildung 21 - Schwachstellen bei der Struktur des Verbinders

Der so entwickelte elastische Verbinder erfüllt seine Funktion sehr gut. Das einzige derzeit nicht lösbare Problem ist die unvollständige Elastizität des Materials: Keiner der bisher gedruckten Teile kann nach der Biegung von selbst zu 100% zur originalen Form zurückkehren. Wenn die Finger ein paar Mal stark gebogen werden, dann muss man sie von Hand zurückbiegen, damit sie wieder perfekt gerade sind. Dies geschieht wegen des Materials selbst und kann nicht durch eine einfache Änderung der Druckeinstellungen behoben werden.

3.5 - Die Handfläche

In diesem Kapitel geht es um die Herstellung und Entwicklung der Handfläche der Roboterhand. Sie hat eine doppelte Rolle: Sie muss sowohl die Finger als auch die Motoren stützen.

Da die Handfläche ein starres Teil ist, wurde sie wie die Fingerglieder aus PLA gedruckt (vgl. Kapitel 3.3).

Zuerst wurde ein Prototyp ohne Sitze für die Motoren hergestellt. Dieser wurde gedruckt, um mögliche Probleme frühzeitig zu erkennen.

In den Abbildungen 22 - 24 sind die technischen Zeichnungen sowie ein 3D-Bild dieses Prototyps zu sehen.

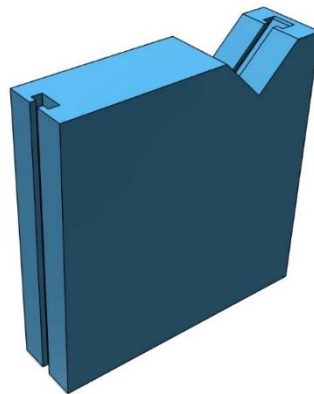


Abbildung 22 - 3D-Bild des ersten Prototyps der Handfläche

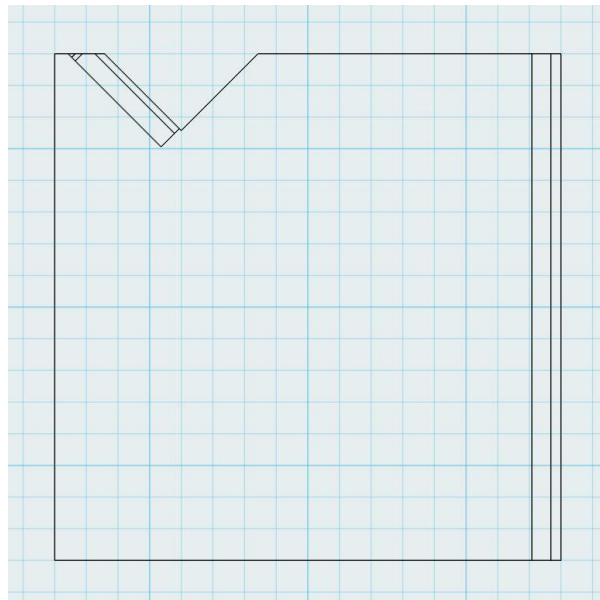


Abbildung 23 - Technische Zeichnung des ersten Prototyps von der Seite (1 Häuschen = 5mm)

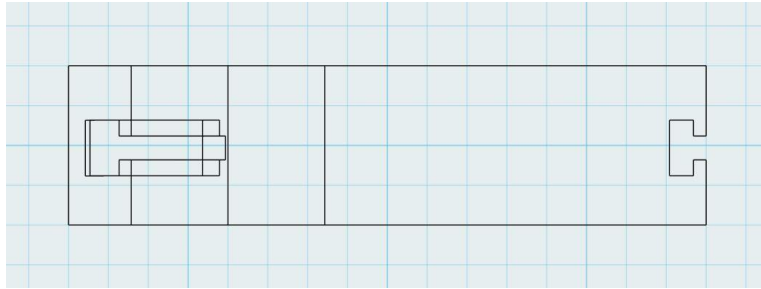


Abbildung 24 - Technische Zeichnung des ersten Prototyps von oben (1 Häuschen = 5mm)

Oben rechts in der Abbildung 22 sollte der Daumen und links die anderen Finger fixiert werden. Dieser Teil wurde mit einem Infill von 20%, einer Druckgeschwindigkeit von 3600mm/min und einer X/Y-Geschwindigkeit (vgl. Kapitel 3.1) von 4200mm/min gedruckt. Die Temperaturen der Plattform und des Extruders sind im Kapitel 3.1 zu finden. Der Druckvorgang ist problemlos abgelaufen, deswegen wurden diese bereits bei der Herstellung der Finger (vgl. Kapitel 3.3) verwendeten Druckeinstellungen bei sämtlichen Versionen der Handfläche angewandt. Ein Problem ist jedoch sofort beim Teil selbst aufgetaucht: Da der Daumen sich einfach senkrecht zur Ebene der Handfläche bog, hätte eine motorisierte Hand mit dieser Handfläche kaum Objekte fassen können.

Um dieses Problem zu lösen, wurde der Daumen bei der zweiten Version um 30° um sich selbst gedreht (bei einer echten Hand ist es auch so, obwohl nicht unbedingt um 30°). Sie ist in Form eines 3D-Bilds sowie technischer Zeichnungen in den Abbildungen 25 - 28 zu sehen. Ab dieser Version wurden die Sitze für die Motoren und die Röhren für die Angelschnur eingeführt (in den technischen Zeichnungen werden diese Elemente ersichtlich gemacht).

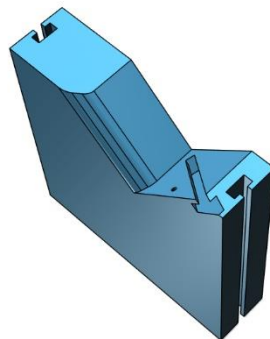


Abbildung 25 - Die Vorderseite der zweiten Version der Handfläche

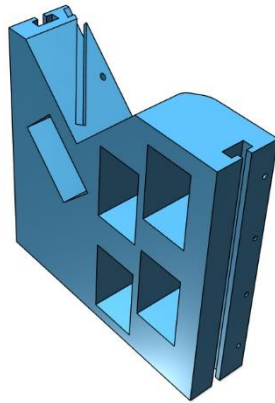


Abbildung 26 - Die Rückseite der zweiten Version der Handfläche

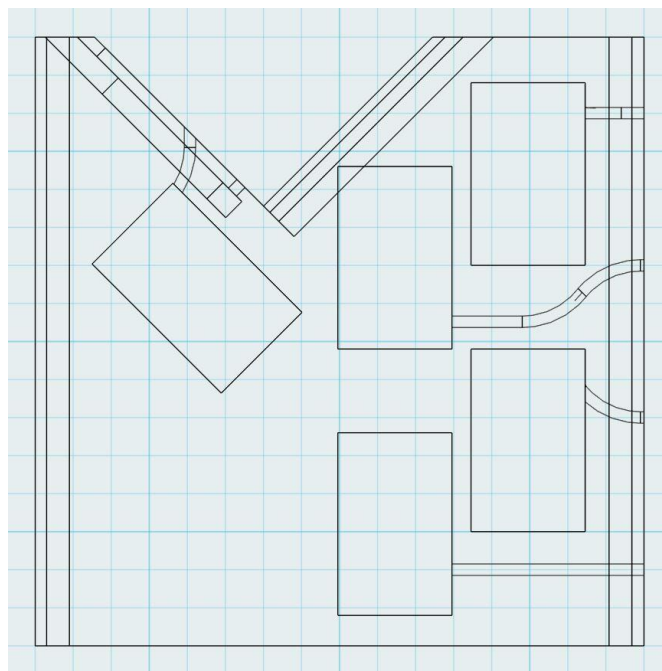


Abbildung 27 - Die zweite Version der Handfläche von der Seite betrachtet (1 Häuschen = 5mm)

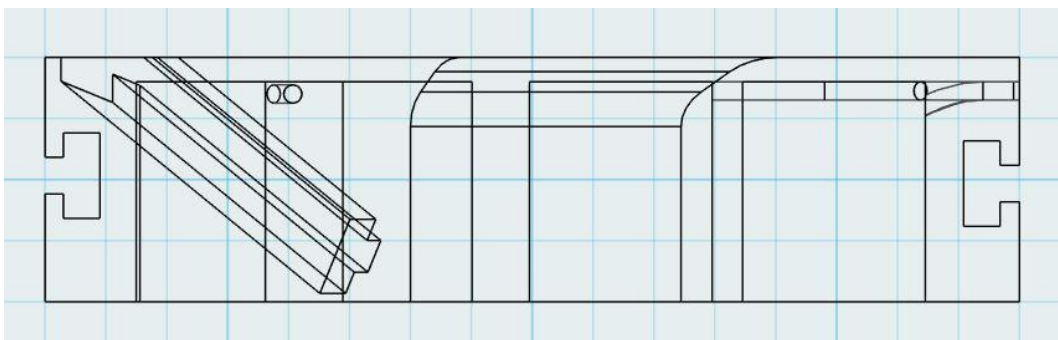


Abbildung 28 - Die zweite Version der Handfläche von oben betrachtet (1 Häuschen = 5mm)

Die Finger konnten zwar mit dieser Handfläche Objekte besser als vorher fassen, jedoch gab es immer noch das Problem, dass der Daumen nicht opponierbar war. Aus diesem Grund war der Griff immer noch nicht zufriedenstellend (vgl. Abbildung 29).

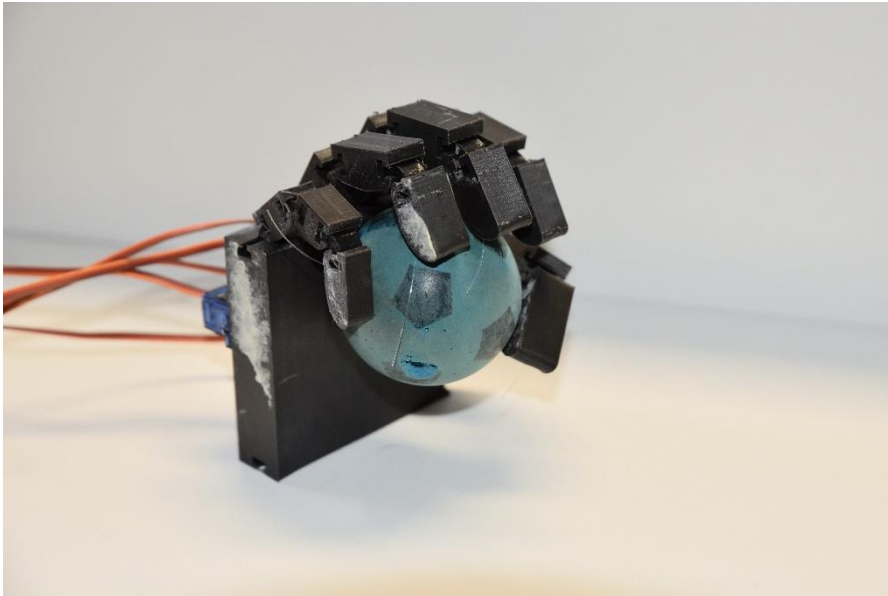


Abbildung 29 - Der Griff der ersten motorisierten Version der Hand

Ein weiteres Problem war, dass die gekrümmten Röhren zum Teil nicht sehr gut gedruckt waren (der Grund dafür ist unbekannt). Dies hat dazu geführt, dass es sehr schwierig war, die Angelschnur durch sie zu ziehen.

Das letzte Problem entstand durch die Schrauben, durch welche die Motoren auf der Handfläche fixiert waren: Da der Infill nur 20% war (vgl. Kap. 3.1), waren die Schrauben nicht fixiert genug aufgrund des fehlenden Materials im Inneren. Weil die Motoren ziemlich stark an die Angelschnur ziehen, war diese Verbindung zu schwach, um deren Kraft tragen zu können. Die Motoren haben sich sogar geneigt, während sie die Finger gebogen haben. Mit der Zeit hätte diese Version der Handfläche dadurch sehr leicht kaputtgehen können.

In der dritten Version wurde ein semi-opponierbarer Daumen eingeführt, indem die Handfläche in zwei relativ zueinander bewegliche Teile geteilt war, welche miteinander durch einen der bereits für die Fingerglieder entwickelten Verbinder (vgl. Kapitel 3.4) verbunden waren. Die Röhren für die Angelschnur waren gerade und es wurden Löcher für die Schrauben eingeführt. Diese Löcher besaßen echte Wände, um die Verbindung zwischen Schrauben und Handfläche zu verstärken. Zudem wurde dieser Prototyp auch auf der Seite ohne Finger mit einem Sitz für die Verbinder ausgestattet, sodass er an andere Objekte (bsp. das Handgelenk) fixiert werden kann. In den folgenden Abbildungen (30 – 32) ist diese Version abgebildet.

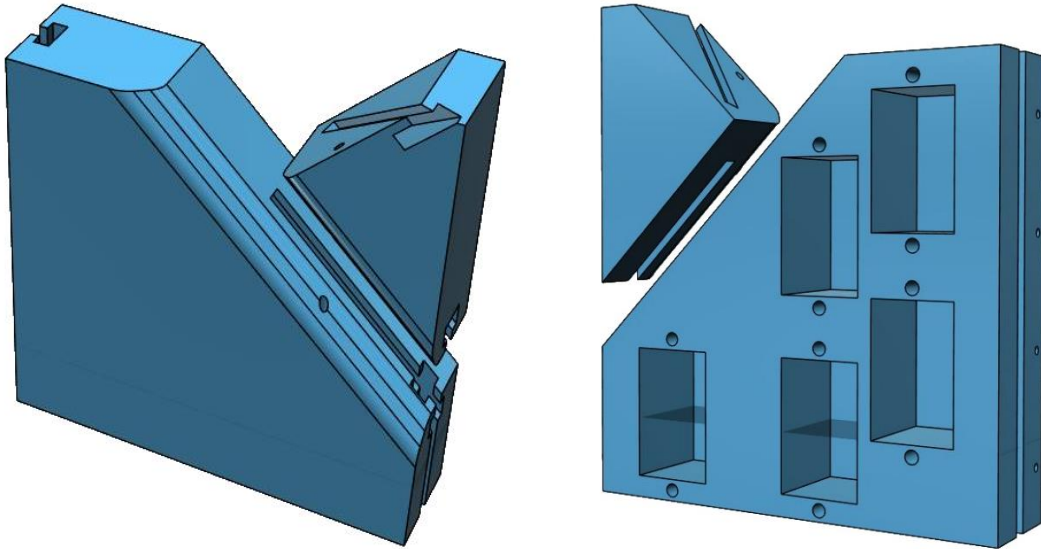


Abbildung 30 – 3D-Modell der dritten Version der Handfläche von vorne und hinten betrachtet

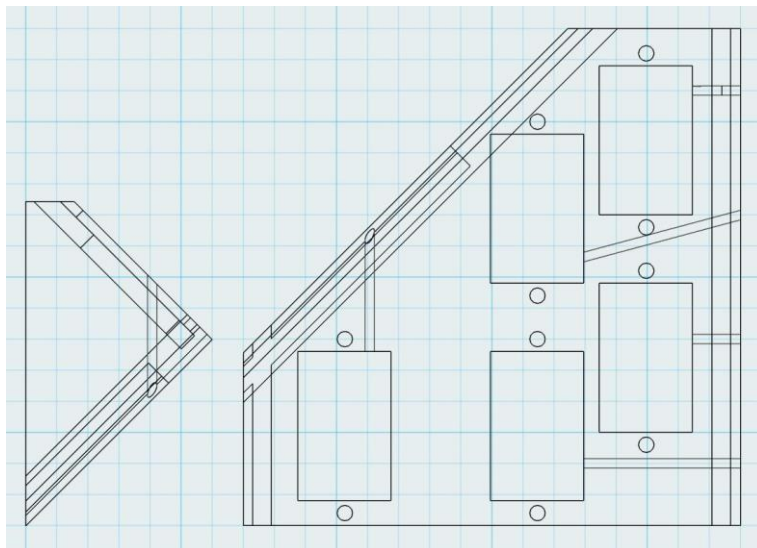


Abbildung 31 – Die dritte Version der Handfläche von der Seite betrachtet (1 Häuschen = 5mm)

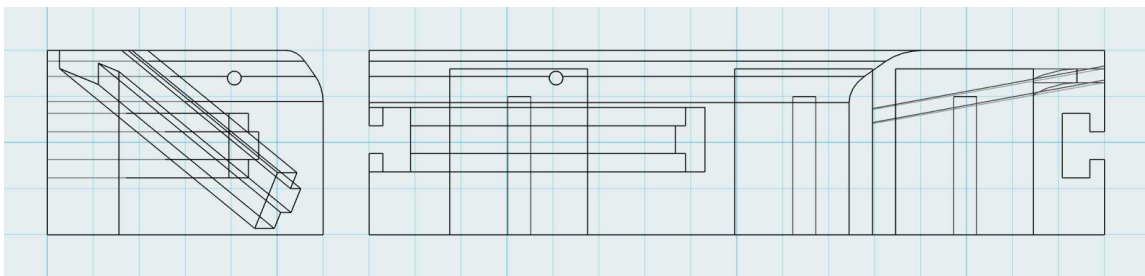


Abbildung 32 - Die dritte Version der Handfläche von oben betrachtet (1 Häuschen = 5mm)

Der Griff der Hand konnte noch überzeugender gemacht werden, indem das kleine, dreieckige Teil neu modelliert und gedruckt wurde (vgl. Abbildungen 33 und 34). Mit diesem neuen Stück ist der Daumen senkrecht zu den anderen Fingern (früher war er nur um 45 Grad geneigt). Die so erhaltene Handfläche wurde als definitiv angenommen.

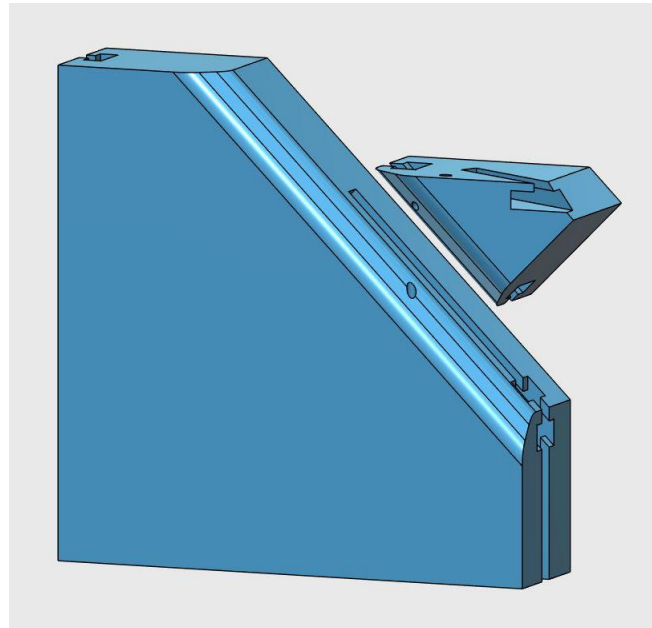


Abbildung 33 - 3D-Modell der letzten Version der Handfläche

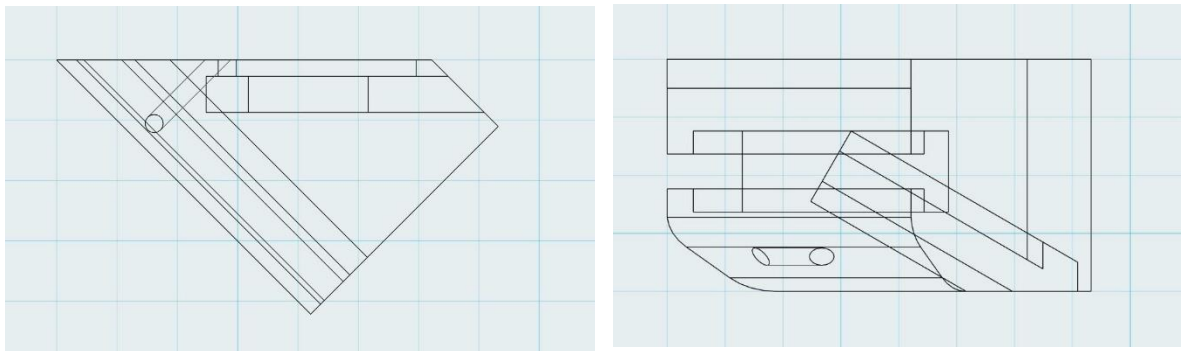


Abbildung 34 – Das dreieckige, bewegliche Teil der letzten Version der Handfläche von der Seite und von oben betrachtet
(1 Häuschen = 5mm)

3.6 - Die Motorisierung

In diesem Kapitel wird die Motorisierung der Hand besprochen.

Wie im Abstract (vgl. Kapitel 2) erklärt, wurde die Roboterhand so gedacht, dass die Fingerglieder durch elastische Verbinder aneinander fixiert sind und durch das Ziehen einer Schnur gebogen werden können. Die Motoren, deren Aufgabe diejenige ist, diese Schnüre zu ziehen, sollen an der Handfläche montiert werden.

Die Schnur, die gewählt wurde, ist eine Angelschnur, weil sie auch unter hoher Spannung nicht reisst und, da sie nicht faserig ist, kann man sie sehr einfach in die Löcher einfädeln.

Es konnte zwischen zwei Arten von üblicherweise in der Robotik verwendeten elektrischen Motoren gewählt werden: Servomotoren und Schrittmotoren.

Die Hauptunterschiede zwischen Servomotoren und Schrittmotoren sind die folgenden: Schrittmotoren besitzen sehr viel Pole und besitzen darum das grössere Drehmoment bei niedrigen Drehzahlen. Servomotoren haben hingegen weniger Pole und sind für höhere Drehzahlen geeignet. Schrittmotoren haben die Vorteile, dass sie einerseits in der Regel kostengünstiger sind und andererseits ein höheres Haltedrehmoment (Widerstand bei Drehung aufgrund externer Faktoren) besitzen⁵. Aufgrund ihrer Eigenschaften finden Sie in Maschinen wie 3D-Drucker eine Anwendung. Servomotoren sind aber sehr oft kleiner und diese Eigenschaft ist beim Einbau in die Handfläche sehr wichtig, da der Raum stark begrenzt ist.

Es wurde darum gewählt, erstmals einen Servomotor in Verbindung mit einem Finger zu testen. Mit diesem Versuch musste bewiesen werden, dass der Motor einerseits genug Drehmoment besitzt, um die Finger biegen zu können, und andererseits über ein genug grosses Haltedrehmoment verfügt, damit der Finger trotz der elastischen Eigenschaften der Verbinder in gebogener Position gehalten werden kann.

Für dieses Experiment wurde der Motor „Joy-it FS90r“ gewählt.

Um diesen Versuch ausführen zu können, musste zunächst die notwendige Struktur eingerichtet werden. Mit dem 3D-Drucker wurde einen Teil der Handfläche hergestellt, der einen Sitz für den Motor besass. In den Abbildungen 35 - 37 sind die technischen Zeichnungen dieses Stückes und ein 3D-Bild davon zu sehen.

⁵ <https://www.smart-production.de/etz/news-detailansicht/nsctrl/detail/News/schritt-versus-servomotoren-2016822/> (Stand 12/12/2020)

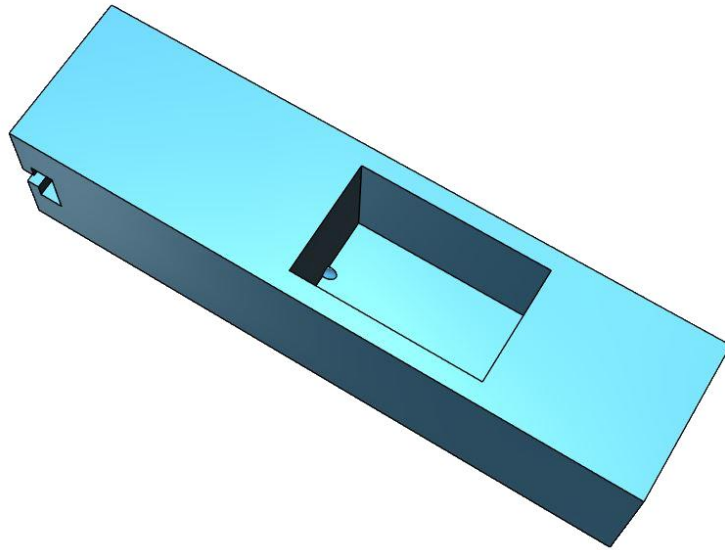


Abbildung 35 - Das Teil, das beim ersten Motorisierungsversuch als Handfläche diente

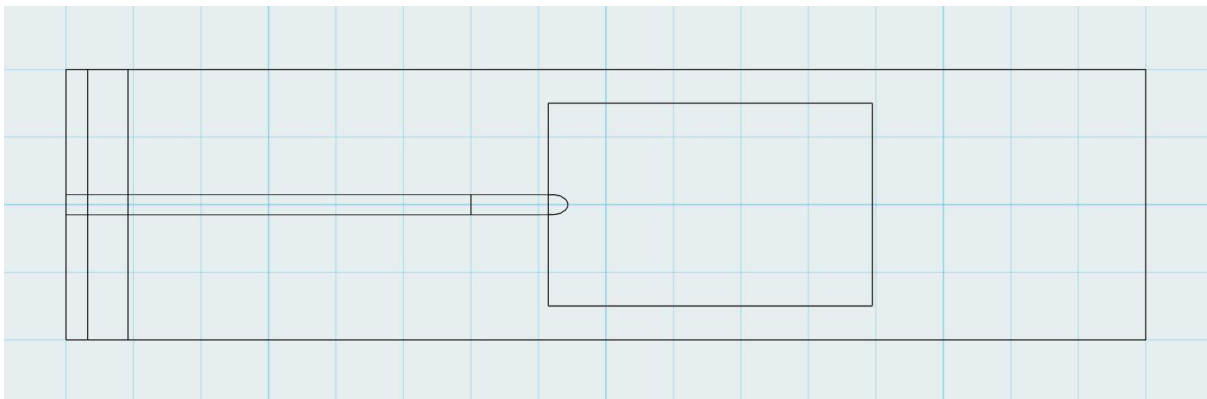


Abbildung 36 - Die Handfläche des ersten Motorisierungsversuchs von oben betrachtet (1 Häuschen = 5mm)

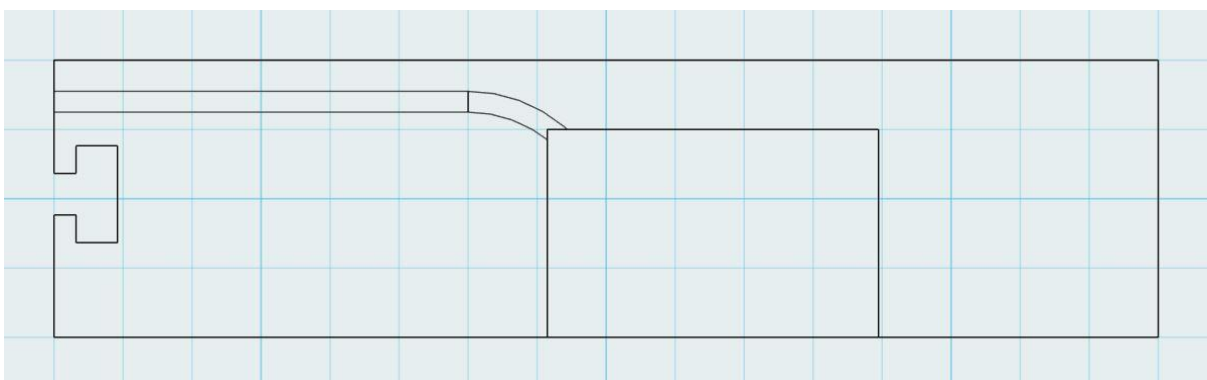


Abbildung 37 - Die Handfläche des ersten Motorisierungsversuchs von der Seite betrachtet (1 Häuschen = 5mm)

An diesem Teil wurde ein Finger montiert. Der Motor sollte nun mit der Achse nach innen auf die Handfläche durch die mit dem Motor gelieferten Schrauben aufgeschraubt werden. An der Achse des Motors befindet sich eine Fadenspule, um welche sich der Faden wickelt (vgl. Abbildung 38). Die Fadenspule wurde durch Umbau eines mit dem Motor gelieferten Teil, das als Hebel gedacht wäre (vgl. Abbildung 39), erhalten. Um dieses Teil umzubauen wurde es, nachdem die vier Hebel mit einer

Zange entfernt wurden, an der Spitze einer Bohrmaschine fixiert und mit Schleifpapier geschliffen. Diese selbstgebastelte Drehbank ermöglichte es, sehr schnell runde Teile zu erhalten, deren Form mit einer zufriedenstellenden Genauigkeit derjenigen eines Kreises entspricht. In den Bildern 39 und 40 wird dieser Umbauprozess gezeigt. Diese Stücke sind die einzigen, die von Hand hergestellt werden mussten. Dabei musste auf die Qualität gut aufgepasst werden, um die Funktion der Hand nicht zu kompromittieren.

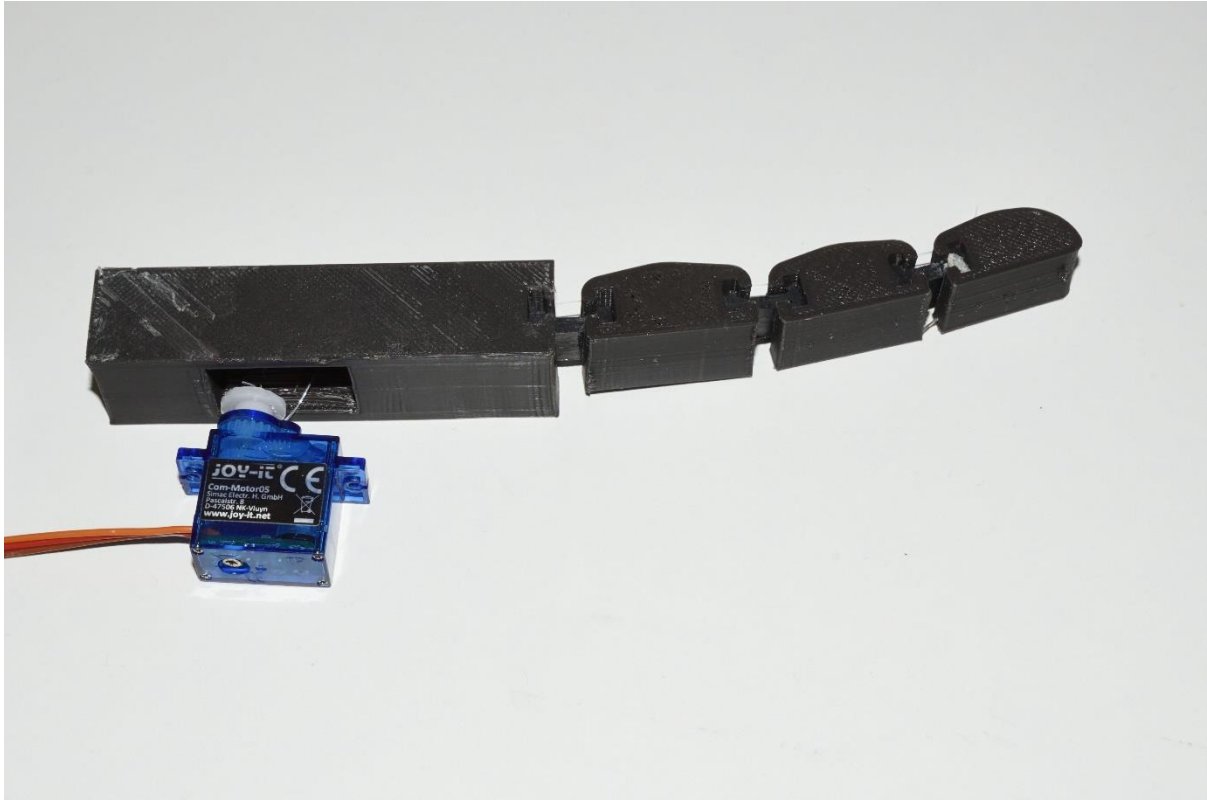


Abbildung 38 - Finger mit demontiertem Motor

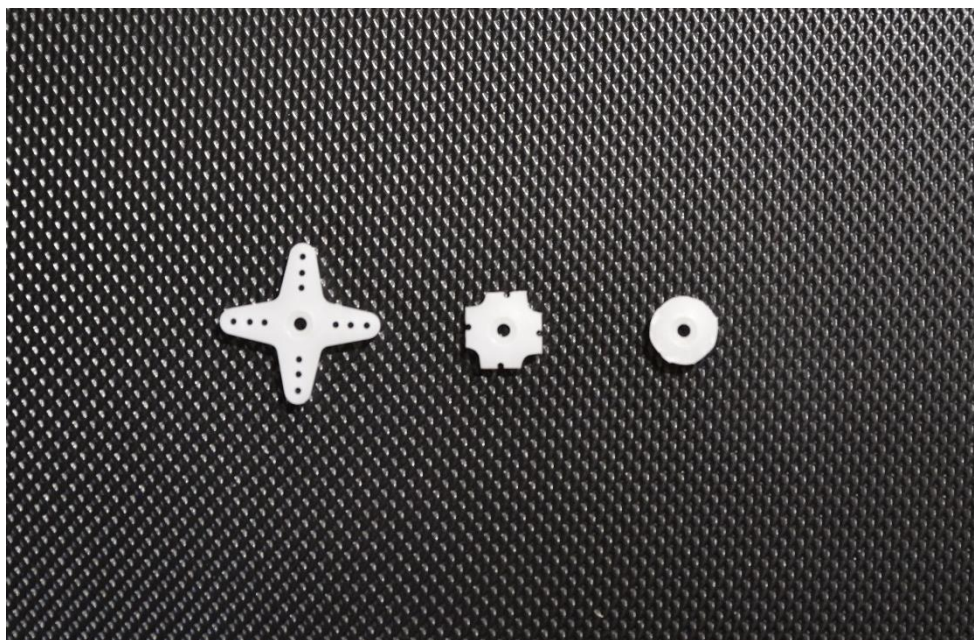


Abbildung 39 - Die verschiedenen Phasen der Herstellung der Fadenspulen aus den mit dem Motor gelieferten Hebeln



Abbildung 40 - Die Fadenspule ist an der Bohrmaschine fixiert, um geschliffen zu werden

In jede Fadenspule wurden drei kleine Löcher gebohrt, um die Schnur zu verknoten. Die Anzahl der Löcher ist genau 3, wie in der Abbildung 41 gezeigt, sodass sich der Knoten ausserhalb der Fadenspule befindet (und somit die Funktion des Mechanismus möglichst wenig stört) und der Faden im Inneren der Spule eingerollt wird.

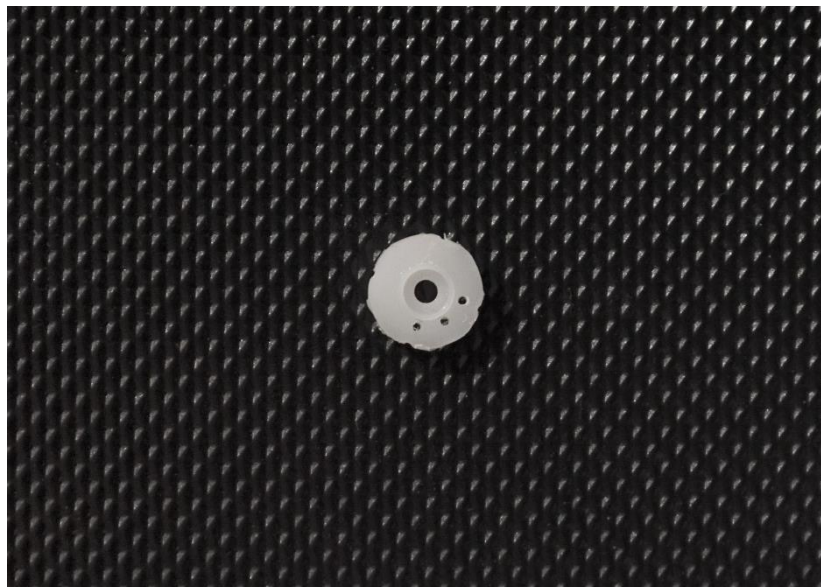


Abbildung 41 - Die fertige Fadenspule mit den Löchern für die Angelschnur

Die Fadenspulen wurden aus zwei Gründen nicht mit dem Drucker hergestellt: Erstens hätte es während des Druckvorgangs wesentliche Probleme aufgrund des Overheating gegeben (vgl. Kapitel 3.4), zweitens ist es günstig, die schon vorhandene Teile zu verwenden, denn sie besitzen schon die perfekte Form (mit kleinen Zähnen), um mit den Achsen der Motoren perfekt zusammenzupassen (der vorhandene Drucker wäre zu ungenau gewesen, um diese kleinen Zähne reproduzieren zu können, vgl. Abbildung 42).

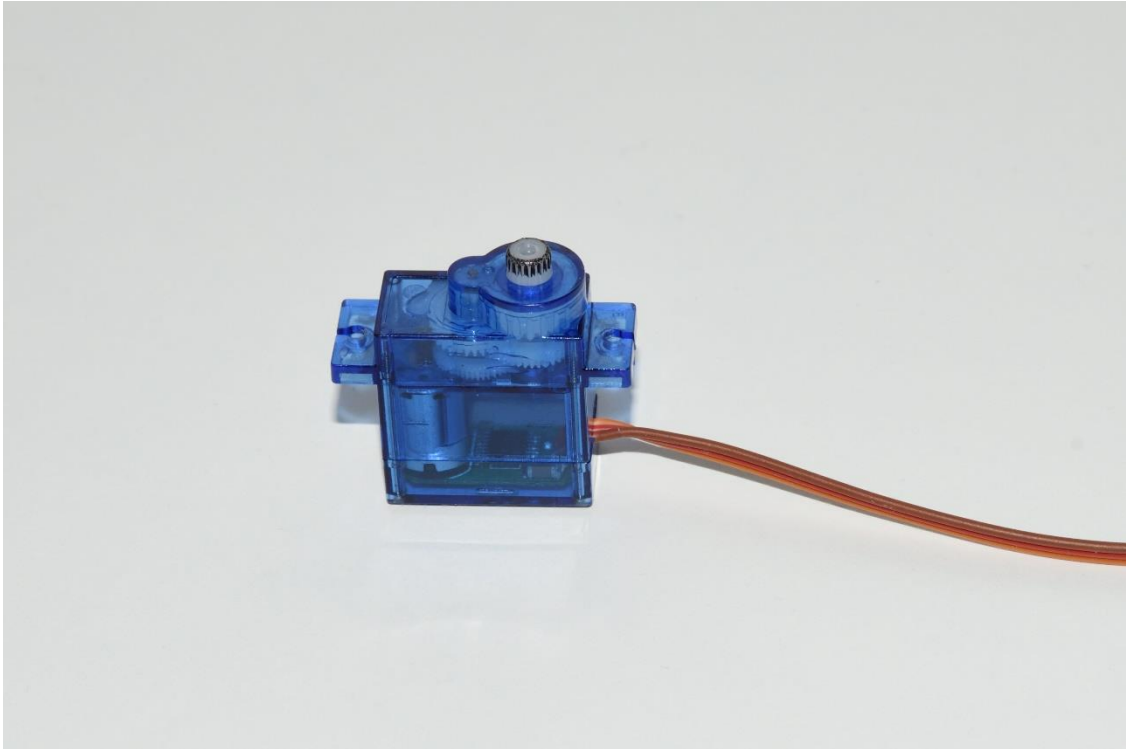


Abbildung 42 - Achse des Motors mit kleinen Zähnen (hier wurden sie geschwärzt, um sie ersichtlicher zu machen)

Weil der Motor selbst bereits mit einer Übersetzung ausgestattet ist, konnte mit diesem Test bewiesen werden, dass er beide Bedingungen (genügend Drehmoment und Haltedrehmoment) erfüllt: Der Finger konnte problemlos gebogen werden (vgl. Abbildung 43) und der Motor konnte ihn auch im Stillstand gebogen halten. Es musste also kein Schrittmotor verwendet werden, somit wurde viel Platz gespart.

Diese Struktur wurde für den Rest des Projektes angenommen.

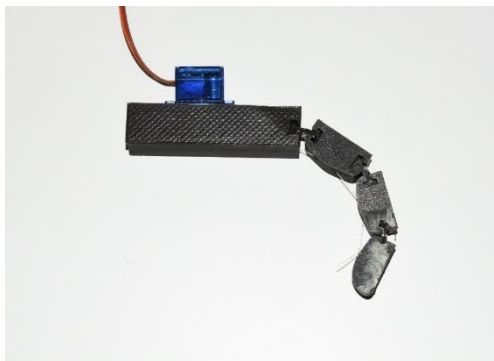


Abbildung 43 - Test des Servomotors an einem Finger

4 - Die Steuerung

4.1 - Die Bedienung der Roboterhand

Die Bedienung der Hand erfolgt durch einen Joystick:

- Die Bewegung entlang der X-Achse ermöglicht es, verschiedene Finger zu selektieren, indem man durch die zur Verfügung gestellten Kombinationen scrollt (die Finger können alle zusammen oder einzeln gesteuert werden, es gibt also insgesamt sechs Kombinationen). Der selektierte Finger wird auf dem Bildschirm des Computers angezeigt, jedoch könnte man auch ein Display direkt in die Steuerung einbauen und programmieren, falls ein Computer während der Bedienung der Roboterhand nicht vorhanden sein sollte.
- Die Bewegung entlang der Y-Achse steuert die Geschwindigkeit der selektierten Motoren: Je weiter von der Mitte der Joystick bewegt wird, umso höher die Drehzahl in entsprechender Richtung.

Dabei gilt zu beachten: Um es zu vermeiden, dass während der Bedienung kleine ungewollte Bewegungen entlang der falschen Achse Probleme verursachen (beispielsweise muss vermieden werden, dass man während der Bedienung der Motoren versehentlich andere Finger selektiert), ist ein Messfehler implementiert worden (vgl. Kapitel 4.7): Wenn sich der Joystick bei einer Koordinate nah an der Mitte befindet, dann wird dessen Position nicht als Befehl interpretiert.

Zudem wurde eine weitere Massnahme implementiert, um die Bedienung einfacher und bequemer zu machen: Wenn man durch die selektierbaren Finger scrollt, gibt es zwischen jeder Selektion eine Wartezeit von 0,5s. Dies soll es vermeiden, dass man zu schnell durch die Finger geht und Mühe hat, den gewünschten Finger zu selektieren.

4.2 - Der Microcontroller „Arduino“

Für die Steuerung dieser Hand wurde ein Arduino Mega 2560 verwendet. Beim Arduino handelt es sich um eine programmierbare Karte, die es ermöglicht, elektronische Komponenten und Schaltungen zu steuern. Er kann mit dem Computer während der Ausführung des Programmes durch ein USB-Kabel kommunizieren, es ist aber nicht notwendig. Das Programm wird im geeigneten IDE (Integrated Development Environment; eine Anwendung, welche Funktionen für Softwareentwickler anbietet) in einer spezifisch für Arduino entwickelten Programmiersprache geschrieben, kompiliert und via USB auf den Arduino geladen. Der Arduino führt dann das Programm aus, sobald er mit Strom versorgt wird (dabei ist es egal, ob der Computer verbunden ist).



Abbildung 44 - Ein Arduino Mega 2560⁶

4.3 - Die Servomotoren

Für dieses Projekt wurden Servomotoren des Modells „Joy-it FS90r“ gewählt (vgl. Kapitel 3.6).

Die Steuerung der angewandten Motoren funktioniert folgendermassen: Jeder Motor besitzt 3 Kabel. Das braune ist die Masse, das rote die Stromversorgung und das gelbe die Steuerung. Zwischen dem roten und dem braunen Kabel sollte es ständig eine Spannung von 5V geben, die zur Bewegung des Motors dienen soll. Durch das braune Kabel wird in digitalem Format eine Zahl zwischen 0 und 180 dem Motor gesendet, die dessen Geschwindigkeit bestimmen soll. Dabei heissen 90 der Stillstand, 180 die maximale Drehzahl in eine Richtung und 0 die maximale Drehzahl in die andere. Um die Übertragung dieser Informationen kümmert sich ein spezifisch für Servomotoren entwickeltes Modul namens „Servo.h“.



Abbildung 45 - Ein Servomotor des Typs "Joy-it FS90R"⁷

⁶ <https://store.arduino.cc/arduino-mega-2560-rev3> (Stand 13/12/2020)

⁷ <https://joy-it.net/de/products/COM-Motor05> (Stand 13/12/2020)

4.4 - Der Joystick

Der benutzte Joystick stammt vom Hersteller „Elegoo“. Er besitzt fünf Anschlüsse: Es gibt einen Pluspol für die Stromversorgung, eine Masse und drei Kabel, durch welche Informationen über den Zustand vom Joystick übertragen werden. Über das erste dieser Kabel wird die X-Koordinate des Joysticks kommuniziert, über das zweite die Y-Koordinate und über das dritte den Zustand der eingebauten Drucktaste (diese letzte Funktion wird im Rahmen dieses Projektes nicht benutzt). Die X- und Y-Koordinaten werden über veränderbare Spannungen (zwischen 0 und 5V) durch die jeweiligen Kabel dem Arduino gesendet. Durch die Funktion `analogRead()` kann dieser solche Spannungen als Werte zwischen 0 und 1024 ablesen. Dabei bedeutet 512, dass der Joystick sich bei der betrachteten Koordinate in der Mitte befindet.



Abbildung 46 - Der angewandte Joystick⁸

⁸ <https://www.amazon.com/Joystick-Module-Arduino-Atomic-Market/dp/B00WH89RTS> (Stand 13/12/2020)

4.5 - Die Steckplatine

Um die nötigen Verbindungen zu machen, wurde eine sogenannte Steckplatine benutzt. Sie ist ein Brett mit vielen Steckern, wodurch Kabel und Komponenten mechanisch befestigt und elektrisch verbunden werden können⁹. Der grosse Vorteil dieser Platine besteht darin, dass bei der Verkabelung das Löten nicht notwendig ist. Dies ermöglicht es, schnell und einfach Schaltkreise zu erstellen und bearbeiten. Sie wird sehr oft bei der Herstellung von Prototypen angewandt. In der Abbildung 47 ist eine solche Platine zu sehen.



Abbildung 47 - Eine Steckplatine¹⁰

Um den Schaltkreis verstehen zu wissen, ist es nötig zu kennen, wie die einzelnen Stecker der Platine miteinander verbunden sind. In der Abbildung 48 werden diese Verbindungen ersichtlich gemacht.

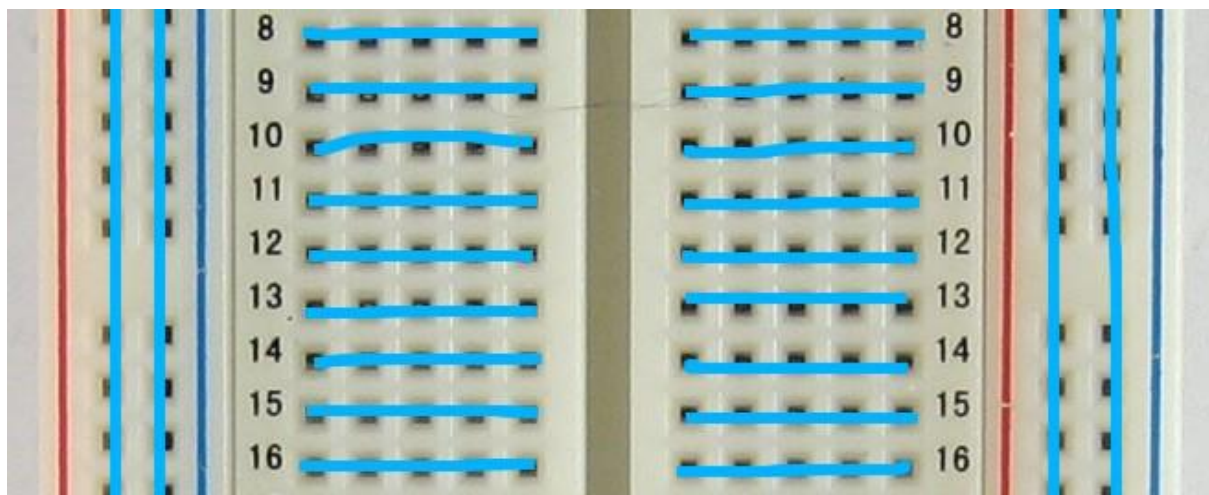


Abbildung 48 - Schema der Verbindungen einer Steckplatine¹¹

⁹[https://de.wikipedia.org/wiki/Steckplatine#:~:text=Eine%20Steckplatine%20\(englisch%20breadboard\)%2C,Bauteilen%20f%C3%BCr%20Versuchsschaltungen%20und%20Experimente](https://de.wikipedia.org/wiki/Steckplatine#:~:text=Eine%20Steckplatine%20(englisch%20breadboard)%2C,Bauteilen%20f%C3%BCr%20Versuchsschaltungen%20und%20Experimente) (Stand 13/12/2020)

¹⁰<https://www.sparkfun.com/products/12002> (Stand 13/12/2020)

¹¹<https://os.mbed.com/handbook/Breadboard> (Stand 13/12/2020)

4.6 - Der Schaltkreis

In diesem Abschnitt wird erklärt, wie der Schaltkreis funktioniert, der aus den verschiedenen Teilen besteht, die in den vorherigen Kapiteln beschrieben wurden.

Ein Schema der ganzen Verschaltung ist in der Abbildung 50 zu finden. Es wurde mit der Software „Fritzing“ gezeichnet. Bei Fritzing handelt es sich um eine open-source Software, die es ermöglicht, genau solche Schemas sehr schnell und einfach zu zeichnen.

In diesem Schema wurden die Position und Farbe einiger Teile verändert, um die Abbildung verständlicher zu gestalten. Obwohl die Position der Kabel beim physischen Modell anders aussieht, sind die Verbindungen genau dieselben. Somit bleibt die Funktionsweise des Schaltkreises des Prototyps unverändert.

Das grosse, grüne, rechteckige Teil (links im Bild) ist der Arduino Mega 2560. In der Mitte ist die Steckplatine mit dem entsprechenden Netzteil (dem roten, rechteckigen Teil oben) abgebildet. Unten auf der Platine befindet sich der Joystick mit seinen fünf Anschlüssen (den grünen Punkten). Die fünf blauen Rechtecke sind die Servomotoren, welche die einzelnen Finger biegen können.

Die Funktion der einzelnen Komponenten wird in den Kapiteln 4.2 – 4.5 erklärt.

Der Schaltkreis funktioniert wie folgt: Das Netzteil der Steckplatine versorgt sämtliche Komponenten (auch den Arduino) mit einer Gleichspannung von 5V, darum sind alle Teile mit den äusseren Spalten der Steckplatine verbunden (vgl. Kapitel 4.5). Alle roten Kabel sind mit der Kathode und alle schwarzen (bzw. braunen im Falle der Servomotoren) mit der Anode verbunden.

Die beiden lila Kabel, die den Joystick mit dem Arduino verbinden, dienen zur Übertragung der X- und Y-Koordinaten des Joysticks. Da die Spannung in diesen beiden Kabeln jeden Wert zwischen 0 und 5V annehmen kann (vgl. Kapitel 4.4), müssen diese an analogen Anschlüssen des Arduinos angeschlossen werden: Dieser kann für die Anschlüsse dieser Art die genaue Spannung messen und somit die Position des Joysticks bestimmen. Aus diesem Grund wurden die Schnittstellen 0 und 1 gewählt (sie besitzen nämlich diese Fähigkeit).

Die fünf blauen Kabel, welche den Arduino mit den gelben Kabeln der jeweiligen Servomotoren verbinden (vgl. Kapitel 4.3) haben die Funktion, die Steuerung der Motoren zu ermöglichen. Da die Spannung in diesen Kabeln nur die Werte 0 und 5V annehmen kann (diese entsprechen den Bits 0 bzw. 1, die Steuerung erfolgt also digital), sind für diese Verbindungen keine analogen Anschlüsse beim Arduino notwendig: Über die digitalen Schnittstellen wird eine entweder 0 oder 5V betragende Spannung gemessen und in Bits (0 oder 1) übersetzt. Aus diesem Grund wurden die Schnittstellen 2 bis 6 gewählt.

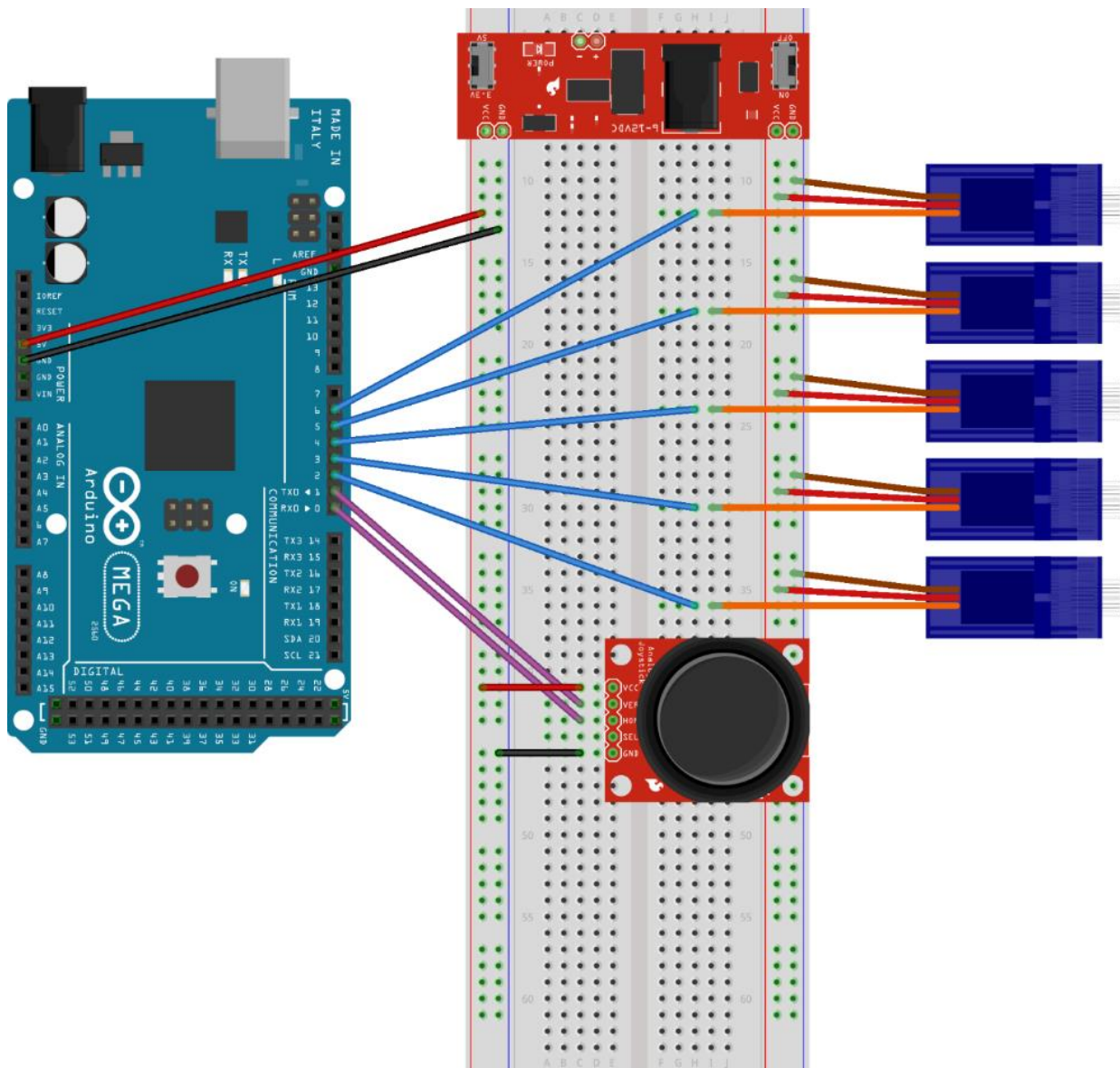


Abbildung 50 - Schema des Schaltkreises

Um das Ganze in Betrieb zu nehmen, muss man nur sicherstellen, dass im Arduino das passende Programm gespeichert und das Netzteil der Platine mit Strom versorgt ist.

Der Schaltkreis könnte auch mit dem kostengünstigeren Controller „Arduino Uno“ statt des Arduino Mega 2560 hergestellt werden (beim Arduino Mega handelt es sich nur um eine Version des Arduinos, die mehr Anschlüsse besitzt, somit ist er auch für grössere Projekte geeignet). Dabei würden die Nummern der Anschlüsse wie auch der Code genau gleichbleiben. Bei diesem Prototyp wurde der Arduino Mega 2560 verwendet, denn er stand bereits zur Verfügung. Darum wird bei den Herstellungskosten nicht der Preis eines Arduino Mega 2560 gelistet, sondern derjenige eines Arduino Uno.

4.7 - Der Code

In diesem Abschnitt wird das im Arduino gespeicherte Programm erklärt. Um diesen Teil vollständig zu verstehen, können Kenntnisse im Bereich der Programmierung sehr hilfreich sein.

Ein Programm eines Arduinos besteht im Allgemeinen aus zwei Teilen: Einem Teil, der am Anfang ausgeführt wird, und einem, der zyklisch unendlich oft läuft (vorausgesetzt, dass der Arduino mit Strom versorgt wird). Der Erste muss in der Methode `setup()` und der Zweite in der Methode `loop()` geschrieben werden.

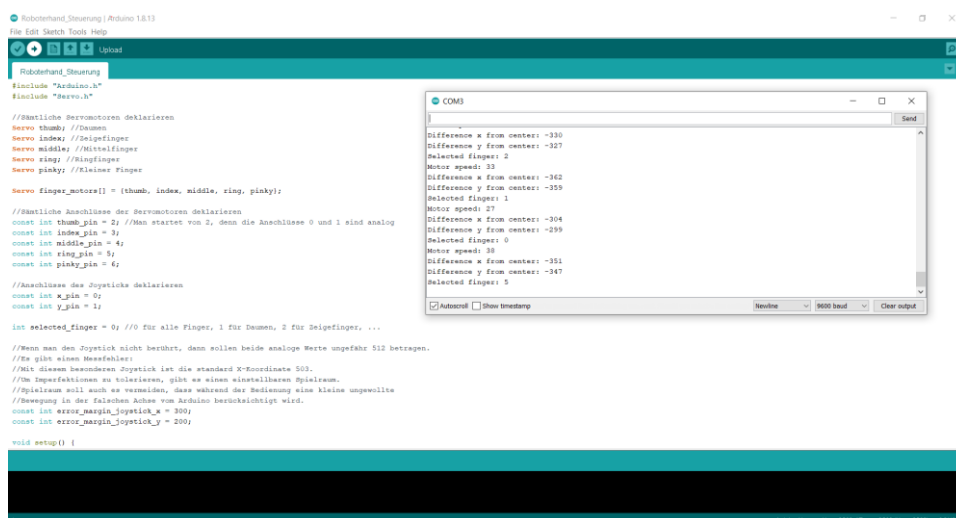
Der Code funktioniert auf folgender Weise:

Zuerst werden die nötigen Module importiert (Arduino.h ermöglicht den Zugang zu den Standardtypen und -Konstanten der Programmiersprache und Servo.h ist das im Kapitel 4.3 beschriebene Modul).

Im Anschluss werden die nötigen Variablen definiert, nämlich:

- Die Servomotoren, als vom Modul „Servo.h“ stammende Objekte des Typs „Servo“;
- Die Nummern der Anschlüsse der Servomotoren;
- Die Nummern der Anschlüsse des Joysticks (diese Anschlüsse müssen analog sein, vgl. Unterkap. „Der Joystick“ und „Der Schaltkreis“);
- Der im Moment selektierte Finger (der Standardwert dieser Variable ist 0): Wenn diese Variable 0 beträgt, dann sind alle Finger selektiert. Wenn sie als Wert eine ganze Zahl zwischen 1 und 5 besitzt, dann sind einzelne Finger selektiert (1 für den Daumen, 2 für den Zeigefinger usw...);
- Die Messfehler in X- und Y-Richtung des Joysticks (vgl. Kapitel 4.4): Wenn der Betrag der Abweichung der X- oder Y-Koordinate von der Mitte kleiner als der Messfehler ist, dann wird kein Befehl aus der betrachteten Koordinate abgeleitet.

In der Methode `setup()` wird jedem Servomotor dem entsprechenden Anschluss zugeordnet. Zudem wird der Serial Monitor initialisiert. Beim Serial Monitor handelt es sich um ein Programm, welches ermöglicht, Text auf dem Computerbildschirm durch den via USB verbundenen Arduino anzuzeigen (vgl. Abb. 51). Mit dessen Hilfe können Informationen wie die Geschwindigkeit der Motoren oder der selektierte Finger ersichtlich gemacht werden.



```
Roboterhand_Steuerung | Arduino 1.8.13
File Edit Sketch Tools Help
Upload
Roboterhand_Steuerung
#include "Arduino.h"
#include "Servo.h"

//Mittliche Servomotoren deklarieren
Servo thumb; //Daumen
Servo index; //Zeigefinger
Servo middle; //Mittelfinger
Servo ring; //Ringfinger
Servo pinky; //Kleiner Finger

Servo finger_motors[] = {thumb, index, middle, ring, pinky};

//Mittliche Anschlüsse der Servomotoren deklarieren
const int thumb_pin = 2; //Man startet von 2, denn die Anschlüsse 0 und 1 sind analog
const int index_pin = 3;
const int middle_pin = 4;
const int ring_pin = 5;
const int pinky_pin = 6;

//Anschlüsse des Joysticks deklarieren
const int x_pin = 0;
const int y_pin = 1;

int selected_finger = 0; //0 für alle Finger, 1 für Daumen, 2 für Zeigefinger, ...

//Wenn man den Joystick nicht berührt, dann sollen beide analoge Werte ungefähr 512 betragen.
//Es gibt einen Messfehler
//Mit diesem besonderen Joystick ist die standard X-Koordinate 503.
//Die Imperfektionen zu tolerieren, gibt es einen einstellbaren Spielraum.
//Spielraum soll auch so verweisen, dass während der Bedienung eine kleine ungewollte
//Bewegung in der falschen Achse vom Arduino berücksichtigt wird.
const int error_margin_joystick_x = 300;
const int error_margin_joystick_y = 200;

void setup() {
```

```
COM3
-----
Difference x from center: -530
Difference y from center: -327
Selected finger: 2
Motor speed: 33
Difference x from center: -362
Difference y from center: -355
Selected finger: 1
Motor speed: 27
Difference x from center: -304
Difference y from center: -399
Selected finger: 0
Motor speed: 38
Difference x from center: -351
Difference y from center: -347
Selected finger: 5
Autostart  show timestamp
Neuheit 9600 baud Clear output
```

Abbildung 51 - Arduino-IDE mit Serial Monitor

Innerhalb der Methode loop() sind dann die Befehle implementiert, welche sich tatsächlich um die Funktion der Roboterhand kümmern.

Am Anfang dieser Methode werden sämtliche nicht selektierte Motoren angehalten. Im Anschluss werden die X- und Y- Koordinaten des Joysticks gelesen und durch zwei Variablen gespeichert. Die Abweichung jeder dieser Koordinaten von der Mitte (vgl. Kapitel 4.4) wird berechnet und durch den Serial Monitor auf dem Bildschirm angezeigt.

Es folgt der Codeabschnitt, der sich bei einer Bewegung des Joysticks entlang der X-Achse um die Selektion der Finger kümmert (vgl. Kapitel 4.1). Dabei wird der Messfehler berücksichtigt (vgl. oben). Wenn der Betrag der Abweichung der X-Koordinate von der Mitte grösser als der Messfehler ist, dann unterscheidet man zwei Fälle: Ist die Abweichung positiv, so wird „nach oben geschaltet“ (den nächsten Finger wird selektiert), ansonsten wird „nach unten geschaltet“. Die neu selektierten Finger werden auf dem Serial Monitor angezeigt. Wenn man geschaltet hat, dann gibt es eine Wartezeit von 0,5s (vgl. Kapitel 4.1).

Der nachfolgende Codeabschnitt aktiviert die selektierten Motoren bei einer Betätigung des Joysticks entlang der Y-Achse. Auch hier wird der Messfehler berücksichtigt (vgl. Kapitel 4.1). Im Wesentlichen wird hier die Geschwindigkeit der Motoren, welche einen Wert zwischen 0 und 180 besitzt, aus der Abweichung der Y-Koordinate des Joysticks von der Mitte, welche einen Wert zwischen -512 und 512 annehmen kann, berechnet (dies erfolgt durch die Formel $v_{Motor} = Abw_Y \cdot (180/1024) + 90$). Dieser Wert wird daraufhin den selektierten Motoren zugeordnet. Diese Berechnung muss mit Variablen des Typs „long“ erfolgen, denn die im Prozess entstehenden Zahlen sind zu gross für eine Variable des Typs „int“.

Falls der Betrag der oben erwähnten Abweichung kleiner als der Messfehler ist, dann werden sämtliche Motoren angehalten (das heisst, jedem Motor wird die Geschwindigkeit 90 zugeordnet, vgl. Kapitel 4.3).

Die Geschwindigkeit der Motoren wird auf dem Serial Monitor angezeigt.

Der vollständige Code lautet wie folgt (die Kommentare, die nach „//“ zu lesen sind, werden vom Arduino ignoriert und sollen dazu dienen, den Code besser zu verstehen):

```
#include "Arduino.h"
#include "Servo.h"

//Sämtliche Servomotoren deklarieren
Servo thumb; //Daumen
Servo index; //Zeigefinger
Servo middle; //Mittelfinger
Servo ring; //Ringfinger
Servo pinky; //Kleiner Finger

Servo finger_motors[] = {thumb, index, middle, ring, pinky};

//Sämtliche Anschlüsse der Servomotoren deklarieren
const int thumb_pin = 2; //Man startet von 2, denn die Anschlüsse 0 und 1 sind analog
```

```

const int index_pin = 3;
const int middle_pin = 4;
const int ring_pin = 5;
const int pinky_pin = 6;

//Anschlüsse des Joysticks deklarieren
const int x_pin = 0;
const int y_pin = 1;

int selected_finger = 0; //0 für alle Finger, 1 für Daumen, 2 für Zeigefinger, ...

//Wenn man den Joystick nicht berührt, dann sollen beide analoge Werte ungefähr 512 betragen.
//Es gibt einen Messfehler:
//Mit diesem besonderen Joystick ist die standard X-Koordinate 503.
//Um Imperfektionen zu tolerieren, gibt es einen einstellbaren Spielraum.
//Spielraum soll auch es vermeiden, dass während der Bedienung eine kleine ungewollte
//Bewegung in der falschen Achse vom Arduino berücksichtigt wird.
const int error_margin_joystick_x = 300;
const int error_margin_joystick_y = 200;

void setup() {
//Diese Methode wird nur einmal am Anfang ausgeführt

//Jedem Servomotor einen Anschluss zuordnen
thumb.attach(thumb_pin);
index.attach(index_pin);
middle.attach(middle_pin);
ring.attach(ring_pin);
pinky.attach(pinky_pin);

//Serial Monitor initialisieren
Serial.begin(9600);
}

void loop() {
//Diese Methode wird zyklisch unendlich ausgeführt

//Zuerst werden alle die nicht selektierten Motoren angehalten
if(selected_finger != 0) { //Wenn die Steuerung nicht auf "alle Finger" eingestellt ist

```

```

for (int count = 0; count<6; count++) { //Für jeden Finger
    if(selected_finger != count + 1) { //Wenn der Finger nicht selektiert ist
        finger_motors[count].write(90); //Motor anhalten
    }
}

//X- und Y-Koordinaten vom Joystick lesen
int x_coordinate_joystick = analogRead(x_pin);
int y_coordinate_joystick = analogRead(y_pin);

//Abweichung der Koordinaten von der Mitte berechnen
int difference_x_from_center = x_coordinate_joystick-512;
int difference_y_from_center = y_coordinate_joystick-512;

//Abweichungen auf dem Serial Monitor aus Testzwecken anzeigen
Serial.print("Abweichung von der Mitte in X-Richtung: ");
Serial.print(difference_x_from_center);
Serial.print("\n");
Serial.print("Abweichung von der Mitte in Y-Richtung: ");
Serial.print(difference_y_from_center);
Serial.print("\n");

//Mit diesem Codeabschnitt werden die Motoren über die Bewegung vom Joystick in X-Richtung
↳selektiert

//Kontrollieren, ob die Abweichung in X-Richtung grösser als der oben definierte Messfehler
↳ist
if(difference_x_from_center > error_margin_joystick_x || difference_x_from_center < -
↳1*error_margin_joystick_x){

    if(difference_x_from_center > 0) { //Wenn die Abweichung positiv ist
        selected_finger += 1; //Nächsten Finger selektieren (nach oben schalten)

        //Wenn der kleine Finger selektiert ist und nach oben geschaltet wird,
        //dann zu 0 ("alle Finger") zurückkehren
        if(selected_finger > 5) {
            selected_finger = 0;
        }
    }

    else { //Wenn die Abweichung negativ ist

```

```

selected_finger -= 1; //Nach unten schalten

//Wenn alle Finger selektiert sind und nach unten geschaltet wird,
//dann den kleinen Finger selektieren
if(selected_finger < 0) {
    selected_finger = 5;
}
}

//Selektierten Finger auf dem Serial Monitor anzeigen
Serial.print("Selektierter Finger: ");
Serial.print(selected_finger);
Serial.print("\n");

delay(500); //0,5s warten, damit es nicht zu schnell durch die Finger gescrollt wird
}

//Mit diesem Codeabschnitt werden die Motoren über die Bewegung vom Joystick in Y-Richtung
↳gesteuert
//Kontrollieren, ob die Abweichung in Y-Richtung grösser als der oben definierte Messfehler
↳ist
if(difference_y_from_center > error_margin_joystick_y || difference_y_from_center < -
↳1*error_margin_joystick_y){

//Geschwindigkeit des Motors (0 - 180) berechnen.
//Es können hier keine int-Variablen benutzt werden,
//da während dieser Berechnung zu grosse Zahlen entstehen würden
long difference_y_long = long(difference_y_from_center);
long motor_speed = difference_y_long*180/1024 + 90;

//Geschwindigkeit auf dem Serial Monitor anzeigen
Serial.print("Geschwindigkeit: ");
Serial.print(motor_speed);
Serial.print("\n");

//Den selektierten Motoren die Geschwindigkeit zuordnen
if(selected_finger != 0) { //Wenn nur ein Finger selektiert ist
    finger_motors[selected_finger-1].write(motor_speed); //Geschwindigkeit zuordnen
}
else { //Wenn alle Finger selektiert sind

```

```

    for (int count = 0; count<6; count++) { //Für jeden Finger
        finger_motors[count].write(motor_speed); //Geschwindigkeit zuordnen
    }
}

//Wenn die Abweichung der Y-Koordinate des Joysticks kleiner als der Messfehler ist, dann
↳alle Motoren anhalten

else {
    for (int count = 0; count<6; count++) { //Für jeden Finger
        finger_motors[count].write(90); //Motor anhalten
    }
    //Meldung auf dem Serial Monitor anzeigen
    Serial.print("Sämtl. Motoren angehalten");
    Serial.print("\n");
}
}

```

Dabei gilt zu beachten, dass das Zeichen „↳“ darauf hinweist, dass es im Code zu keinem Zeilenumbruch kam. Die bestehende Zeile gehört somit noch zu der Vorherigen.

Die Farben spielen keine Rolle bei der Funktion des Programmes, sondern dienen dazu, den Code verständlicher zu machen. Die Färbung ist genau diejenige, mit der der Code in der Entwicklungsumgebung (IDE, vgl. Kapitel 4.2) angezeigt wird.

5 - Zusammenbau, Materialkosten und Auswertung

5.1 - Zusammenbau

Die Hand wurde am Ende der Entwicklung der verschiedenen Teile zusammengebaut: Zuerst wurden die starren Teile (Fingerglieder und Handfläche, vgl. Kapitel 3.3 bzw. 3.5) aus PLA-Filament durch die aus elastischem Material hergestellten Verbinder (vgl. Kapitel 4.4) miteinander verbunden. Im Anschluss darauf wurde am Ende der Finger je ein Stück Angelschnur gebunden. Dieses wurde danach durch die entsprechenden Röhren bis zum Sitz des Motors in der Handfläche gezogen und an die Fadenspule (vgl. Kapitel 3.6) gebunden, welche wiederum am Servomotor fixiert wurde. Die Motoren wurden an die Handfläche aufgeschraubt und deren Kabel mit der in den Kapiteln 4.6 und 4.7 beschriebenen Steuerung verbunden. Die so erhaltene Roboterhand ist in der Abbildung 52 zu sehen.

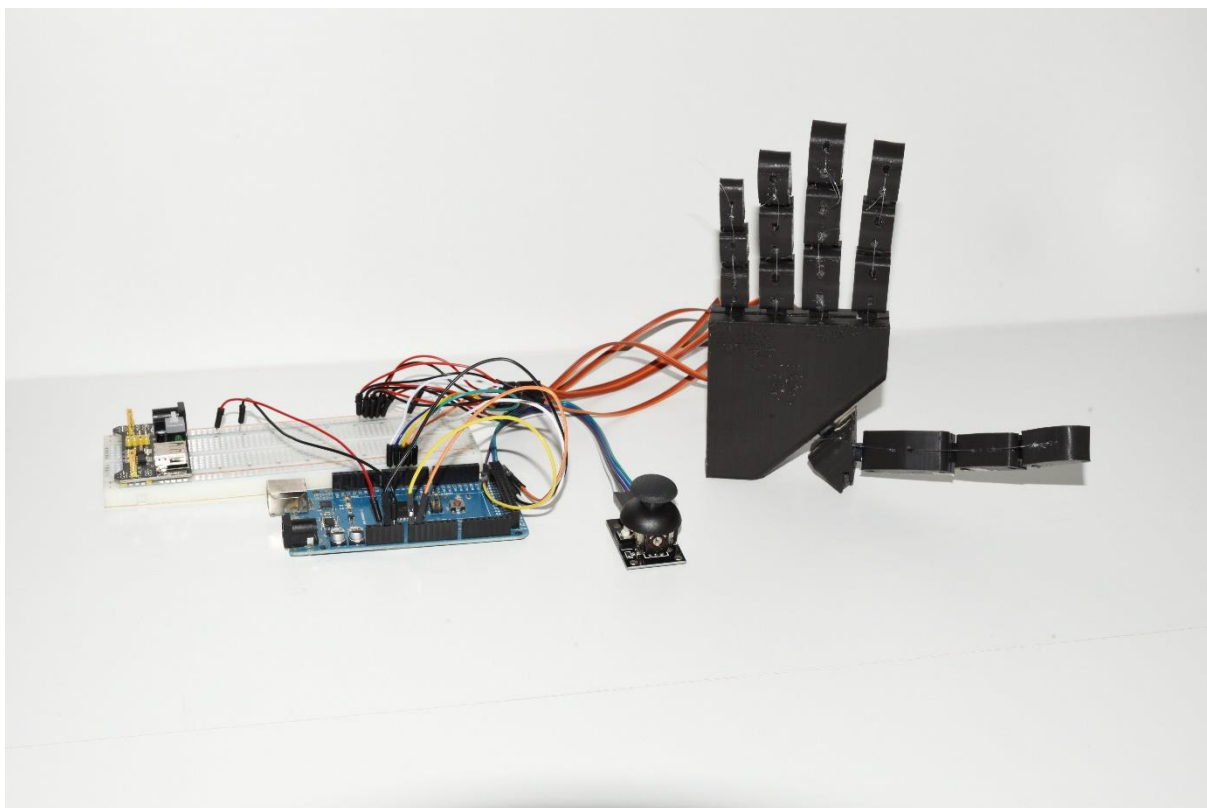


Abbildung 52 - Die fertige Hand und die dazugehörige Steuerung

5.2 – Materialkosten

In diesem Kapitel geht es um die Materialkosten der Herstellung der Roboterhand geredet. Die Idee hinter diesem Abschnitt ist es, eine Idee darüber zu geben, in welcher Grössenordnung sich der Preis der Herstellung eines solchen Produktes befinden könnte.

Es ist Folgendes zu beachten: Die Versandkosten wurden nicht berücksichtigt. Das liegt daran, dass sich diese je nach Situation stark ändern können ohne, das Endprodukt zu beeinflussen. Ferner wurden die Produkte im Detailhandel gekauft, bei einer serienmässigen Herstellung könnten also die Preise in signifikanter Weise von den unten aufgelisteten abweichen. Des Weiteren waren Arduino, Kabel, Joystick und Steckplatine mit entsprechendem Netzteil waren schon vorher vorhanden. Für diese Teile handelt es sich bei den hier angegebenen Preisen um grobe Schätzungen, welche auf Basis von Preisen gleichwertiger Teile gemacht wurden. Ausserdem wurde in der Preisliste mit einem Arduino Uno statt eines Arduino Mega 2560 gerechnet (vgl. Kapitel 4.6). Der 5V-Power Adapter wird im Rahmen dieser Kostenanalyse nicht berücksichtigt, denn es ist streng genommen nicht Teil der Roboterhand oder der Steuerung. Vernachlässigt wird zudem der Preis der Angelschnur, weil sie sehr billig ist und in der Hand nur wenige Zentimeter davon benutzt werden. Nicht aufgelistet sind auch die Kosten, welche mit Maschinen, Werkzeugen, Software und Messinstrumenten verbunden sind: Mit der Zeit werden diese Elemente aufgrund der Entwicklung der Technologie in der Regel besser und ökonomisch günstiger. Daneben ist der hier für das Kunststofffilament aufgelistete Preis bereits auf das Gewicht der 3D-gedruckten Teile umgerechnet worden. Schliesslich ist es wichtig zu erinnern, dass diese Kostenberechnung im Dezember 2020 und mit Preisen Schweizer Verkäufer stattgefunden hat. Zu anderen Zeitpunkten und mit anderen Verkäufern könnte diese Preisliste dementsprechend ungenau sein.

Diese Kostenanalyse ist also, wie schon oben erwähnt, eine Schätzung der Materialkosten unter gewissen Annahmen, der hier angegebene Gesamtpreis kann jedoch viel kleiner als die tatsächlichen Herstellungskosten sein (je nach dem, wie man diese definiert).

Im Folgenden ist die Preisliste zu finden.

Teil	Einzelpreis (CHF, approximiert)	Angewandte Menge (approximiert)	Gesamtpreis (CHF, approximiert)
Arduino Uno	22,00	1	22,00
Steckplatine	5,00	1	5,00
Netzteil der Steckplatine	6,00	1	6,00
Joystick	5,00	1	5,00
Joy-it FS90r Servomotor	8,10	5	40,50
20er Set Kabel	10,00	11	5,50
Renkforce PLA-Filament 1kg	28,00	85g	2,50
Renkforce elastisches Filament 0,5kg	32,00	10g	1,00

Die gesamten Materialkosten für die Herstellung dieser Roboterhand betragen also ungefähr 87 Fr.

5.3 - Auswertung

Um die Roboterhand numerisch auswerten zu können, wurde folgendes Experiment durchgeführt: Die Hand musste einen kleinen Ball aus Stoff greifen, an dem eine Plastiktüte mit einer Schnur gebunden war. Die Hand war so ausgerichtet, dass die Handfläche horizontal war (die Finger waren dabei nach unten gebogen). Im Anschluss darauf wurde die Tüte langsam mit kleinen Gewichten (je 50g) gefüllt, bis der Ball von der Hand gefallen ist. Schliesslich wurden sämtliche von der Hand gehaltene Teile mit Ausnahme des letzten hinzugefügten Gewichtes (dieses konnte von der Hand nicht gehalten werden) gewogen. Das Ziel des Experimentes war, die ungefähre maximale Belastbarkeit der Hand zu bestimmen. Die Einrichtung des Experimentes ist in der Abbildung 53 zu sehen.



Abbildung 53 - Einrichtung des zur Auswertung des Produktes durchgeführten Experimentes

Der oben beschriebene Versuch wurde dreimal durchgeführt. Jedes Mal ist man auf 310g gekommen, bevor die Einrichtung eingestürzt ist (die Anzahl Gewichte war bei jedem Einsturz gleich und die Massen des Balles und der Tüte blieben unverändert). Die ganze Hand ist also ziemlich schwach im Vergleich zur menschlichen.

Weil die einzelnen Finger sehr stark (man kann an einem beliebigen gebogenen Finger problemlos 0,5kg anhängen) und die angewandten Materialien sehr robust sind, muss die Schwäche der Roboterhand an deren Form liegen. Im Rahmen dieser Arbeit ist es leider nicht gelungen, den Daumen komplett opponierbar zu machen (man musste der Hand ein menschliches Aussehen verleihen, darum konnte man nicht den Daumen einfach bereits opponiert fixieren). Zudem wurde

die Form der Handfläche stark approximiert, um die Servomotoren daran leicht fixieren zu können (eine menschliche Handfläche ist weder viereckig noch komplett flach). Zuletzt spielt die Haftreibung zwischen Finger und Objekt eine wichtige Rolle. Die menschliche Hand im Gegensatz zur Roboterhand besteht aus sehr haftendem Material und die Fingerabdrücke erhöhen zusätzlich die Griffestigkeit. Leider haben solche Faktoren die Funktion des Produktes beeinträchtigt.

Das Ziel der Arbeit wurde trotzdem erfolgreich erreicht: Die Roboterhand kann tatsächlich Objekte fassen (vorausgesetzt, dass sie nicht zu schwer sind).

Die einzelnen Finger sind sehr stark und die Steuerung ist besonders einfach zu bedienen und zu verstehen, dieses Projekt hat also ein grosses Potential für mögliche Weiterentwicklungen und Anwendungen. Es folgen einige Ideen:

- Die Handfläche und die allgemeine Form der Hand könnten verbessert werden, damit das Produkt im obigen Test bessere Ergebnisse liefert;
- Die Steuerung könnte so umprogrammiert werden, dass die Hand vollständig von einem Computer via USB gesteuert wird, um sie eventuell mit anderen Systemen verbinden zu können;
- Man könnte einen ganzen Arm herstellen und ihn mit der Roboterhand verbinden;
- Die Handfläche könnte so modelliert werden, dass die Motoren vollständig in deren Inneren sind, um die Roboterhand realitätsnäher zu gestalten;
- Ebenfalls um Realitätsnähe könnte ein spezieller Handschuh entwickelt werden, der der Hand ein möglichst realistisches Aussehen verleiht und die Griffestigkeit und die Haftreibung verbessert;
- Eine stark weiterentwickelte und optimierte Version dieser Hand könnte schliesslich als Prothese angewandt werden.

6 - Quellen und Abbildungsverzeichnis

6.1 - Quellen

- <https://www.printer-care.de/de/drucker-ratgeber/wie-funktioniert-ein-3d-drucker#:~:text=Definition%20des%203D%2DDruckers,ein%20dreidimensionaler%20Gegens tand%20erzeugt%20wird.&text=Heute%20werden%203D%2DDrucker%20vor,Industrie%20o der%20in%20Forschungseinrichtungen%20verwendet.> (Stand 18/12/2020)
- <https://www.my3d.guru/product/wanhao-duplicator-4s-3d-drucker-test-und-erfahrungsberichte/> (Stand 13/12/2020)
- <https://www.conrad.ch/de/p/filament-renkforce-pla-1-75-mm-schwarz-1-kg-1528349.html> (Stand 13/12/2020)
- <https://www.simplify3d.com/support/print-quality-troubleshooting/overheating/#:~:text=The%20most%20common%20cause%20for,not%20b eing%20cooled%20fast%20enough.&text=If%20your%20printer%20includes%20a,and%20se lecting%20the%20Cooling%20tab.> (Stand 20/12/2020)
- <https://www.smart-production.de/etz/news-detailansicht/nsctrl/detail/News/schritt-versus-servomotoren-2016822/> (Stand 12/12/2020)
- <https://store.arduino.cc/arduino-mega-2560-rev3> (Stand 13/12/2020)
- <https://joy-it.net/de/products/COM-Motor05> (Stand 13/12/2020)
- <https://www.amazon.com/Joystick-Module-Arduino-Atomic-Market/dp/B00WH89RTS> (Stand 13/12/2020)
- [https://de.wikipedia.org/wiki/Steckplatine#:~:text=Eine%20Steckplatine%20\(englisch%20br eadboard\)%2C,Bauteilen%20f%C3%BCr%20Versuchsschaltungen%20und%20Experimente](https://de.wikipedia.org/wiki/Steckplatine#:~:text=Eine%20Steckplatine%20(englisch%20br eadboard)%2C,Bauteilen%20f%C3%BCr%20Versuchsschaltungen%20und%20Experimente) (Stand 13/12/2020)
- <https://www.sparkfun.com/products/12002> (Stand 13/12/2020)
- <https://os.mbed.com/handbook/Breadboard> (Stand 13/12/2020)
- <https://www.canadarobotix.com/products/1755> (Stand 13/12/2020)

6.2 - Abbildungsverzeichnis

Abbildung 1 - Die Roboterhand mit gebogenen Fingern.....	4
Abbildung 2 - Ein Wanhao Duplicator 4S.....	5
Abbildung 3 - Filament aus PLA für den 3D-Drucker	6
Abbildung 4 - Eine Kugel im STL-Format.....	6
Abbildung 5 - Das Modell des ersten Fingers von oben (1 Häuschen = 5mm)	9
Abbildung 6 - Das Modell des ersten Fingers von der Seite (alle Angaben in mm)	9
Abbildung 7 - Löcher, welche auf der Oberfläche bei einem zu kleinen Infill entstehen.....	10
Abbildung 8 - Das zweite Modell von oben (1 Häuschen = 5mm)	11
Abbildung 9 - Das zweite Modell von der Seite (alle Angaben in mm)	11
Abbildung 10 - Die definitiven Fingerglieder von der Seite (1 Häuschen = 5mm)	11
Abbildung 11 - Die definitiven Fingerglieder von oben (die gewölbte Seite ist in dieser Zeichnung nach unten gerichtet)	12
Abbildung 12 - 3D-Bild der definitiven Finger.....	12
Abbildung 13 - Vergleich zwischen geraden und gekrümmten Röhren	13
Abbildung 14 - Die Form des Verbinders.....	14
Abbildung 15 - Die erste Version des Verbinders von oben betrachtet.....	15
Abbildung 16 - Die erste Version des Verbinders von der Seite betrachtet	15
Abbildung 17 - 3D-Modell der mit verschiedenen Faktoren skalierten Verbinder.....	15
Abbildung 18 - Beispiel von Overheating	16
Abbildung 19 - Der definitive Verbinder von oben betrachtet (das Teil besitzt eine vertikale Symmetrieachse in der Mitte).	17
Abbildung 20 - Der definitive Verbinder von der Seite betrachtet	17
Abbildung 21 - Schwachstellen bei der Struktur des Verbinders.....	18
Abbildung 22 - 3D-Bild des ersten Prototyps der Handfläche	19
Abbildung 23 - Technische Zeichnung des ersten Prototyps von der Seite (1 Häuschen = 5mm)	19
Abbildung 24 - Technische Zeichnung des ersten Prototyps von oben (1 Häuschen = 5mm).....	20
Abbildung 25 - Die Vorderseite der zweiten Version der Handfläche.....	20
Abbildung 26 - Die Rückseite der zweiten Version der Handfläche	21
Abbildung 27 - Die zweite Version der Handfläche von der Seite betrachtet (1 Häuschen = 5mm) ...	21
Abbildung 28 - Die zweite Version der Handfläche von oben betrachtet (1 Häuschen = 5mm)	21
Abbildung 29 - Der Griff der ersten motorisierten Version der Hand	22
Abbildung 30 – 3D-Modell der dritten Version der Handfläche von vorne und hinten betrachtet.....	23
Abbildung 31 – Die dritte Version der Handfläche von der Seite betrachtet (1 Häuschen = 5mm)	23
Abbildung 32 - Die dritte Version der Handfläche von oben betrachtet (1 Häuschen = 5mm).....	23
Abbildung 33 - 3D-Modell der letzten Version der Handfläche	24
Abbildung 34 – Das dreieckige, bewegliche Teil der letzten Version der Handfläche von der Seite und von oben betrachtet (1 Häuschen = 5mm)	24
Abbildung 35 - Das Teil, das beim ersten Motorisierungsversuch als Handfläche diente	26
Abbildung 36 - Die Handfläche des ersten Motorisierungsversuchs von oben betrachtet (1 Häuschen = 5mm).....	26
Abbildung 37 - Die Handfläche des ersten Motorisierungsversuchs von der Seite betrachtet (1 Häuschen = 5mm)	26
Abbildung 38 - Finger mit demontiertem Motor	27
Abbildung 39 - Die verschiedenen Phasen der Herstellung der Fadenspulen aus den mit dem Motor gelieferten Hebeln.....	27
Abbildung 40 - Die Fadenspule ist an der Bohrmaschine fixiert, um geschliffen zu werden	28

Abbildung 41 - Die fertige Fadenspule mit den Löchern für die Angelschnur	28
Abbildung 42 - Achse des Motors mit kleinen Zähnen (hier wurden sie geschwärzt, um sie ersichtlicher zu machen).....	29
Abbildung 43 - Test des Servomotors an einem Finger.....	29
Abbildung 44 - Ein Arduino Mega 2560.....	31
Abbildung 45 - Ein Servomotor des Typs "Joy-it FS90R"	31
Abbildung 46 - Der angewandte Joystick.....	32
Abbildung 47 - Eine Steckplatine.....	33
Abbildung 48 - Schema der Verbindungen einer Steckplatine	33
Abbildung 49 - Eine Steckplatine mit Netzteil	34
Abbildung 50 - Schema des Schaltkreises.....	36
Abbildung 51 - Arduino-IDE mit Serial Monitor	37
Abbildung 52 - Die fertige Hand und die dazugehörige Steuerung.....	43
Abbildung 53 - Einrichtung des zur Auswertung des Produktes durchgeführten Experimentes.....	45