

Greifautomatenspiel nach IPERKA konzipieren

BMTL18a

Lionel Moser, Christoph Mayer, Nicola Scheiwiller, Elia De Lazzer

Coach: Tamara Wiedermann

16/12/21



Inhaltsverzeichnis

Vorwort	1
Kurzzusammenfassung	2
1 Einleitung	4
1.1 Unsere These.....	4
1.2 Ziele.....	4
2 Arbeitsmethoden	5
2.1 IPERKA	5
2.2 rapid Prototyping	5
2.2.1 Warum rapid Prototyping	5
2.2.2 Werkzeuge für rapid Prototyping	5
2.2.3 3D-Druck.....	5
2.2.4 Holz Laser.....	6
2.2.5 CAD	6
3 Informieren.....	7
3.1 Informationen sammeln.....	7
3.2 Projektjournal erstellen.....	7
4 Planen.....	8
4.1 Anforderungsliste	8
4.2 Terminplan	10
4.3 Aufgabenverteilung	11
4.4 Informationen für mögliche Einkaufskomponenten sammeln.....	11
5 Entscheiden	12
5.1 Funktionsstruktur.....	12
5.2 Morphologischer Kasten.....	12
5.3 Prinzip Skizzen.....	13
5.4 Vor - / Nachteilliste	13
5.5 Nutzwertanalyse mit S-Diagramm.....	15
5.6 Entscheidung.....	16
6 Realisieren	17
6.1 Massstäblicher Entwurf	17
6.2 Fertigungsunterlagen mittels CAD erstellen.....	17
6.2.1 Probleme beim Holzlaser.....	18

6.2.2	3D-Druck.....	18
6.3	Einkaufskomponente beschaffen	19
6.4	Kostenliste.....	19
6.5	Probleme bei Bestellungen	19
6.6	Montage und Inbetriebnahme	20
6.7	Probleme bei der Montage	21
6.8	Elektronik Experimentierphase.....	23
6.8.1	Arduino	23
6.8.2	Motor Tests.....	24
6.8.3	LCD Tests.....	26
6.8.4	CNC Shield, Greifer und On-Off	27
6.8.5	I2C Kommunikation	28
6.8.6	Analoge Joystickabfrage.....	28
6.8.7	Umstieg auf Arduino Mega	31
6.8.8	Probleme der Bauteilbestimmung.....	31
6.9	Elektronik Umsetzungsphase.....	31
6.9.1	Spannungsregler und Motortreiber.....	32
6.9.2	Motortreiber mit Arduino	33
6.9.3	Step Down Converter	34
6.9.4	Schema.....	35
6.9.5	Bestückung	38
6.10	Programmierung.....	41
6.11	Verdrahtung.....	46
6.12	Messungen.....	46
6.13	Fehlerbehebung	50
6.14	Namensfindung «Clawminator»	51
6.15	Sticker	52
6.16	Gewinn	52
7	Kontrollieren	53
7.1	Anforderungsliste überprüfen	53
7.2	Sicherheit überprüfen	54
7.3	Spielbarkeit überprüfen	54
8	Auswerten	56
8.1	mögliche Punkte in einer Auswertung	56
8.2	Auswertung	56

8.3	Verbesserungspunkte für Version 2.0	56
8.4	Ausblick	56
9	Probleme / Schwierigkeiten.....	57
9.1	Zeitlicher Aspekt.....	57
9.2	Lieferengpässe / Bestellungen	57
9.3	Ortsabhängige Zusammenarbeit.....	57
10	Schlussbemerkungen	59
11	Abbildungsverzeichnis	60
12	Literaturverzeichnis	63
13	Stückliste von CAD	66
14	Anhänge	69
15	Selbstständigkeitserklärung	70

Vorwort

Um ein gutes Produkt herstellen zu können, braucht man einen sinnvollen Ablauf. Dieser Ablauf nennt sich IPERKA und wir haben diesen anhand eines Beispiels einmal komplett durchsimuliert. Für das haben wir versucht gemeinsam ein Greifautomatenspiel zu planen und auch tatsächlich herzustellen.

Bereits im Juni beim IDAF Projekt waren wir vier eine gemeinsame Gruppe und konnten dort bereits durch eine harmonische Zusammenarbeit großartige Erfolge erzielen. Unser eingespieltes Team mit technischem Knowhow musste nun wieder vereint werden.

Wir haben uns für eine Arbeit entschieden, bei welcher wir unser vielfältiges Knowhow einsetzen können und am Ende ein fertiges Produkt in den Händen haben. Unsere Gruppe hat verschiedene Skills, welche wir in unserer Arbeit einsetzen konnten. Nach einiger Zeit kamen wir auf die Idee ein Greifautomatenspiel anhand der IPERKA Arbeitsmethode zu konzipieren.

Wir besitzen alle eine Faszination für Technik und haben Freude am Tüfteln. Zudem konnten wir in unserer Ausbildung bereits mit IPERKA und dem Entwickeln diverser Produkte Erfahrung sammeln. Dieses Wissen wollen wir nun selbständig anwenden.

Unsere gewählte Arbeit war sehr zeitintensiv, was wir jedoch sehr gerne in Kauf genommen haben, da wir Freude an der Arbeit hatten und am Ende auch ein Produkt in den Händen auf welches wir stolz sein können. Während der Arbeit hatten wir einiges an Problemen, welche uns sehr viel Zeit gekostet haben.

Unsere Arbeit hätten wir vier allein nicht mit gleicher Qualität vollenden können, wenn wir nicht die proaktive Unterstützung von folgenden Personen / Firmen erhalten hätten:

Wir bedanken uns herzlichst ans...

... Bühler Makerspace Team für die Hilfe und Unterstützung bei der Umsetzung....
(speziell Roman Tribelhorn, Damian Kobler, Timo Weiss)

... Frau Wiedermann für die Begleitung während des ganzen Prozesses

... Christian für die Mithilfe an der Elektronik und Verdrahtung

Wir wünschen Ihnen viel Freude beim Lesen dieser interdisziplinäre Projektarbeit!

Christoph, Lionel, Elia, Nicola

Kurzzusammenfassung

Das Ziel unserer Arbeit ist es, anhand eines Greifautomaten den Entwicklungsprozess eines Produktes aufzuzeigen. Wir zeigen unsere Vorgehensweise Schritt für Schritt detailliert nach IPERKA auf und welche Werkzeuge, Materialien und Maschinen wir für unsere Arbeit benötigten. Die IDPA befasst sich zudem mit unseren Beweggründen und Zielen, sowie die Probleme, denen wir in der Entwicklung begegnet sind.

Das IDPA Projekt begann, wie jedes andere Schulprojekt auch, mit der Gruppenfindung. Diese fiel uns leicht, da wir in einem vorigen Projekt bereits in derselben Zusammenstellung gearbeitet haben. Dabei haben wir gemerkt, dass wir gut miteinander harmonieren und sich unsere Fähigkeiten gut ergänzen. In einem nächsten Schritt musste ein Thema gefunden werden, dafür haben wir viel Zeit gebraucht, da wir uns nie alle einig waren. Mittels diverser Ideenfindungsmethoden kamen wir dann schliesslich auf das Greifautomatenspiel.

Als erstes haben wir einen Terminplan erstellt, dort haben wir aufgeschrieben was zu tun ist und wie viel Zeit wir dafür einplanen, die Zeit, die wir tatsächlich benötigt haben, haben wir parallel zum Projekt immer ergänzt. Als der Zeitplan gemacht war, konnten wir uns ans eigentliche Projekt wagen, dafür haben wir mittels eines Morphologischen Kasten ein paar Ideen zu den Funktionen, welche wir in der Funktionsstruktur festgelegt hatten, skizziert. Mit einem S-Diagramm und der Vor- und Nachteilliste haben wir uns für eine Variante entschieden, welche wir dann weiterverfolgen würden.

Nun begann jeder etwas selbstständiger zu Arbeiten. Christoph bestellte sich einen Arduino und begann sich über die Elektronik und Programmierung zu informieren; der Rest der Gruppe hat am Kasten selbst gearbeitet und noch etwas weiter am Endprodukt geplant. An einem Samstag haben wir uns dann zum Teile bestellen bei Christoph getroffen und das nötige Material bestellt.

Die nächsten paar IDPA Lektionen haben wir vom Makerspace gearbeitet, um dort unsere selbsthergestellten Teile zu fertigen. Wir hatten kleine Konstruktionsfehler, welche wir ganz klassisch mit Bohrmaschine und Feile ausbessern mussten. Nachdem alles zusammenpasste, konnten wir mit der Montage beginnen, diese war simpel da das Gehäuse so konstruiert ist, dass wir nur noch zusammenstecken, kleben und ein paar Teile mit Schrauben befestigen mussten.

Im elektronischen Teil befassten wir uns näher mit dem Arduino und verschiedenen Motoren, wir haben dabei mit verschiedenen Schaltungen und Programmen experimentiert, bevor es

zur Umsetzung kam. In der Umsetzung haben wir uns tiefer mit diversen Bauteilen befasst, haben eigene Schaltungen konzipiert, ein Schema gezeichnet, bestückt, verdrahtet und schlussendlich eine Fehlerbehebung mit anschliessenden Messungen durchgeführt. Insgesamt war die Planung wie auch die Umsetzung der Elektronik ein langwieriges und zähes Unterfangen mit vielen Schwierigkeiten, die wir aber allesamt nach unserem besten Wissenstand lösen konnten.

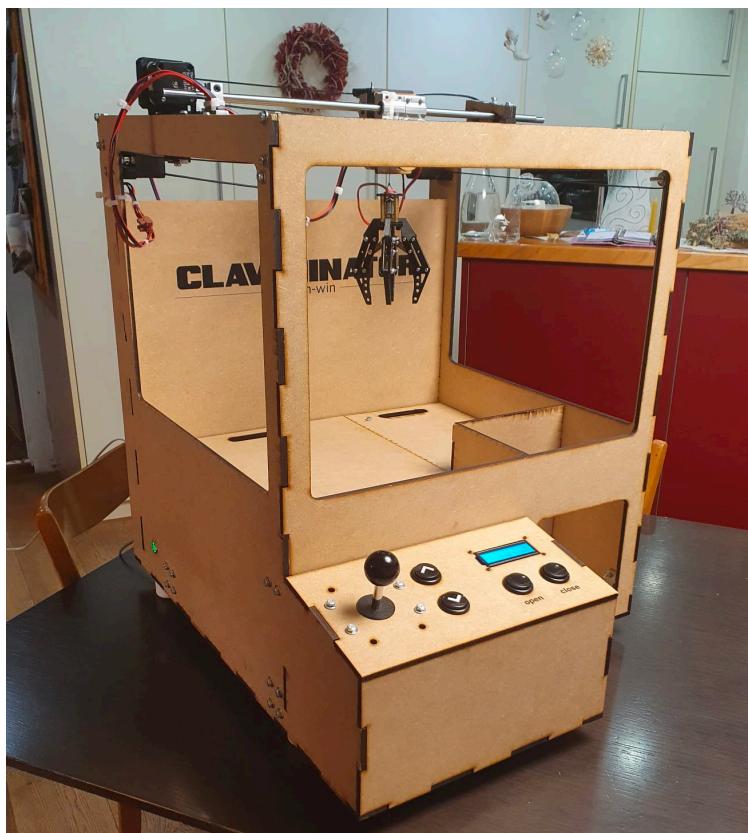


Abbildung 1: Fertiger Greifautomat

1 Einleitung

Sicher kennen Sie den Frust, welchen man verspürt, wenn man beim Greifautomatenspiel sein Taschengeld verspielt hat und vielleicht die Chance verpasst, einem besonderen Mitmenschen einen Teddy zu schenken. Eine Situation, die wir alle kennen und uns auch frustriert.

Mit dieser Motivation beim Greifautomatenspiel endlich ein Erfolgserlebnis zu schaffen, haben wir unser Projekt gestartet.

1.1 Unsere These

Ein Automat mit welchem man tatsächlich eine gute Chance zum Gewinnen hat, damit das Spielen auch Freude macht.

1.2 Ziele

Ein Automat, der technische so gut funktioniert, dass die Klaue die Preise auch tatsächlich rausziehen kann, musste her. Wir wollen nach IPERKA-Prinzip ein Greifautomatenspiel erarbeiten und realisieren welches technisch so gut funktioniert, dass man einfach kein «Pech» mehr haben kann.

2 Arbeitsmethoden

Damit ein Produktionsprozess gelingt, gibt es Arbeitsmethoden, die einen dabei unterstützen können. Sie geben Struktur. Besonders wenn im Team gearbeitet wird, sind solche Methoden von Nutzen. Es gibt verschiedene Vorgehensweisen. Wir werden uns in dieser Arbeit besonders auf IPERKA konzentrieren. Rapid Prototyping werden wir auch noch kurz anschneiden.

2.1 IPERKA

IPERKA ist eine bekannte Arbeitsmethode, die in der Produktentwicklung angewendet wird. Sie besteht aus den sechs Schritten; Informieren, Planen, Entscheiden, Realisieren, Kontrollieren und Auswerten. Die einzelnen Teilschritte zu den sechs Phasen werden wir anhand unseres Projektes, dem Greifautomaten, aufzeigen und erklären.

Wir haben uns für IPERKA entschieden, da wir diese Arbeitsmethode in unserer Ausbildung studiert und schon mehrfach angewendet haben.

2.2 rapid Prototyping

Rapid Prototyping bedeutet schneller Prototypenbau. Der Zweck davon ist die Abschätzung der Möglichkeiten in der wirklichen Umsetzung.

2.2.1 Warum rapid Prototyping

«Über das Rapid Prototyping-Verfahren können Prototypen wesentlich schneller hergestellt werden als mit herkömmlichen Methoden. ... Dank der schnellen Herstellung können Modelle und Prototypen wesentlich früher und auch häufiger verwendet werden.» (Hintsteiner, 2021)

2.2.2 Werkzeuge für rapid Prototyping

Damit Rapid Prototyping funktioniert werden modernste Techniken angewandt, wie der 3D-Druck, Holzlaser oder sonstige, an dieser Stelle werden wir diese etwas vertiefen

2.2.3 3D-Druck

Grundlage für den 3D-Druck ist immer eine am Computer zuvor erstellte dreidimensionale CAD-Zeichnung, die ein Volumenmodell sein muss. Bei einem Volumenmodell sind die ganzen Oberflächen beschrieben im Vergleich zu Netzmodellen, wo nur die Kante beschrieben ist (Fastermann, 2012, S. 7). Ist ein solches Modell erst einmal erarbeitet, kann es relativ einfach im 3D-Drucker eingelesen und ausgedruckt werden.

Mit Hilfe von 3D-Druckern kann man komplexe Bauteile über Nacht ausdrucken. Man ist in der Grösse der Bauteile jedoch sehr eingeschränkt. Die Bauteile sind im Vergleich zu gleichgeformten Stahlteilen bei weitem nicht so stabil. Dafür sind sie sehr leicht. Die fehlende Festigkeit ist für die meisten Prototypen aber ausreichend. Besonders im Vordergrund stehen aber schnelle und erschwingliche Fertigung. Die Genauigkeit variiert je nach Gerät.

2.2.4 Holz Laser

Der Holz Laser ist ein sehr praktisches Werkzeug, um sehr schnell etwas herzustellen. Er kann Holz- und Kunststoffplatten bearbeiten. Man kann damit viel grössere Sachen herstellen als mit dem 3D-Drucker. Man ist in der Form der Bauteile eingeschränkt, da man nur platten Lasern kann.

Wir haben uns dazu entschieden den Grossteil der Bauteile mithilfe des Holz Lasers herzustellen. Dieses Verfahren hat sich sehr bewährt. Besonders die Geschwindigkeit, in der die Bauteile gefertigt werden, ist überzeugend. Sobald man die komplizierte Anlage einmal eingerichtet hat.

2.2.5 CAD

CAD bedeutet Computer - aided design. Beim CAD handelt es sich um ein Programm, womit man Bauteile aus einer zweidimensionalen Skizze mit ein paar einfachen Tools in einen dreidimensionalen Körper zeichnet. Aus dem dreidimensionalen Model erstellt man dann Fertigungszeichnungen und sonstige Fertigungsunterlagen. Die dreidimensionalen Modelle sind auch sehr anschaulich, was sehr nützlich sein kann, wenn man jemandem mit einem weniger guten Vorstellungsvermögen etwas erklären oder zeigen möchte.

In der Industrie sind sie weit verbreitet. Sie sind das Hauptwerkzeug eines Konstrukteurs und haben deren Arbeitsweise grundlegend verändert. Auf diesen Wandel vom Handzeichnen zum 3D-Modellieren werden wir später noch eingehen.

In unserer Arbeit benutzen wir Autodesk Inventor sowie Siemens NX10. Mit beiden Programmen haben wir ca. 2 Jahre Erfahrung.

3 Informieren

Informieren ist der erste Schritt im IPERKA-Prozess. Ihm wird oft nicht genügend Zeit geschenkt. Hier ist es wichtig die Aufgabenstellungen genau zu studieren und möglichst viele Informationen zu sammeln. Damit legt man sich den Grundstein für die restliche Arbeit.

3.1 Informationen sammeln

Wir haben uns verschiedene Greifautomaten in Kaufhäusern, Kinos und Jahrmärkten angesehen und ihre Funktionsweise studiert. Zusätzlich haben wir im Internet nach selbstgebauten Greifautomaten gesucht. Dabei haben wir einige Projekte gefunden, von denen wir inspiriert wurden. Zudem haben wir auch von der Optik der Greifautomaten inspirieren lassen und kamen zum Entschluss, dass das Optische eher nebensächlich sein wird und bei Bedarf in Anspruch genommen werden kann. Das Wichtigste ist, dass das Ganze funktioniert.

3.2 Projektjournal erstellen

In den ersten Lektionen haben wir in Excel ein Projektjournal erstellt, in welchen wir alle Dokumente für unsere Arbeit abgespeichert haben. Dieses Projektjournal haben wir anschließend auf Teams abgelegt, so haben wir alle, Zugriff darauf. Dadurch können wir gleichzeitig und ortsunabhängig daran arbeiten. Darin haben wir alle wichtigen Entscheidungen, Projektdaten, Kontakte, Termine und viele weitere Dokumente festgehalten. Somit hat man schnell einen Überblick über das ganze Projekt und findet alles wichtigen Daten auf einen Blick.

4 Planen

Das Planen ist ein sehr zentraler Punkt unserer Arbeit. Dafür haben wir auch ordentlich Zeit in Anspruch genommen. Denn ein festgehaltener Plan ist essenziell für die Erreichung von Erfolg.

4.1 Anforderungsliste

Bevor man mit dem Planen beginnt, muss man sich zuerst überlegen, was das Endprodukt alles können muss. Wir haben das in Form einer Anforderungsliste gemacht. Im Normalfall bekommt man den Auftrag mit den Anforderungen vom Kunden. In unserem Beispiel durften wir die Anforderungen selbst bestimmen.

In der Anforderungsliste schreibt man zuerst alle Anforderungen auf. Dann unterteilt man sie nach Funktions- und Leistungsanforderungen, Dimensionen, Qualitätsmerkmale, Marktanforderungen, Fertigungskriterien und Kosten. Manche Anforderungen müssen unbedingt erfüllt werden. Sie werden als Forderung gekennzeichnet. Andere sind nur Wünsche. Unsere Wünsche haben wir von 1-5 gewichtet, wobei 1 ein schwacher und 5 ein sehr starker Wunsch ist.

Die Anforderungsliste für unser Projekt befindet sich auf der folgenden Seite.

Kostenschätzung

Was	Wer / Wo	Kosten
Rohmaterial		CHF 50
Einkaufskomponenten		CHF 300
Gesamtkostenschätzung	ca. (+/- 10%)	CHF 350

Abbildung 2: Kostenschätzung aufgrund der Anforderungsliste

Wir haben den Fokus auf die Funktion gelegt. Andere Punkte, wie Design oder Display haben wir als Wunschkriterien bestimmt.

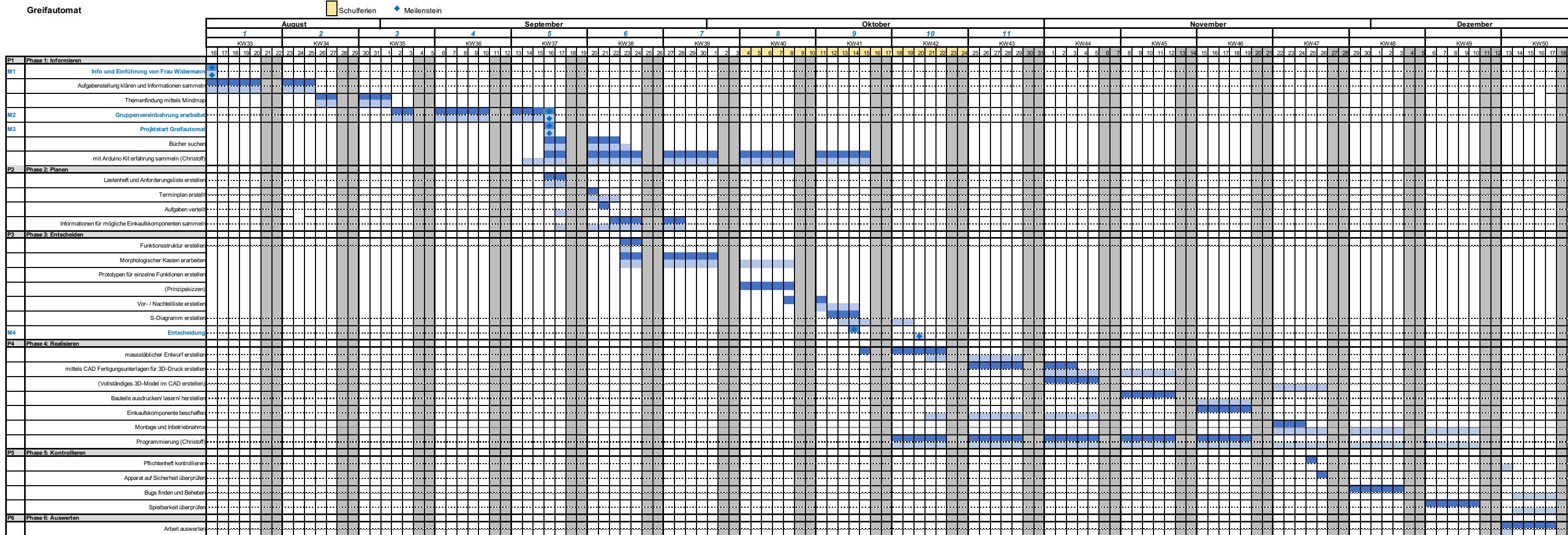
Änderungsdatum:	Anforderung	F = Forderung W = Wunsch 5 = starker 1 = schwache	Bemerkungen	Verantwortlich
	Funktions- und Leistungsanforderungen			
	<i>Greifer ist Horizontal steuerbar</i>			
	<i>Greifer kann vertikal heruntergelassen werden</i>			
	<i>Greifer kann schliessen</i>			
	<i>Fallengelassenes Produkt ist von aussen greifbar</i>			
	<i>Musik beim spielen</i>	2		
	<i>Display</i>	4		
	<i>Steuerung voll elektronisch</i>	5		
	<i>Münzeinwurf</i>	1		
	Dimensionen			
	<i>max. 600 mm x 600 mm x 600 mm</i>			
	<i>min. 300 mm x 300 mm x 300 mm</i>			
	Qualitätsmerkmale			
	<i>System läuft zuverlässig (keine Bugs)</i>			
	<i>Endprodukt fühlt sich hochwertig an</i>			
	Marktanforderungen			
	<i>wenig/ keine Wartung nötig</i>	5		
	<i>ansprechendes Design</i>	4		
	Fertigungskriterien			
	<i>Rahmen grösstenteils mit 3D-Drucker oder Holzlaser herstellbar</i>			
	<i>Stückzahl 1</i>			
	<i>Fertigung und Motage Zuhause, Schule oder im Makerspace möglich</i>			
	Kosten			
	<i>ca. CHF 200.- (maximalkosten CHF 400.-)</i>			

Abbildung 3: Anforderungsliste

4.2 Terminplan

Als nächsten haben wir einen Terminplan erstellt (siehe Anhang). Darin werden alle geplanten Teilschritte eingetragen. Die geschätzte Zeit, die man dafür benötigt, zeichnet man im Soll-Plan ein. Während der Arbeit trägt man laufend den Ist-Plan nach. So kann man Abweichungen direkt erkennen und schnell Änderungen vornehmen. Wichtige Meilensteine haben wir mit kleinen Vierecken markiert. Der Terminplan wird normalerweise vom Auftraggeber unterzeichnet. In unserem Beispiel sind wir selbst die Arbeitgeber. Darum ist in diesem Fall keine Unterzeichnung nötig.

PROJEKTTERMINPLAN



4.3 Aufgabenverteilung

Für den Arbeitsprozess haben wir die Verantwortlichkeiten klar definiert.

Lionel Moser übernimmt die Aufgaben eines Projektleiters: Den Terminplan überprüfen, Überblick behalten, und wenn nötig Anpassungen treffen. Zusätzlich erstellte er mit seinen konstruktiven Fähigkeiten die CAD-Zeichnungen.

Elia De Lazzer übernimmt die Handwerklichen Arbeiten, wie das Montieren und das Lasern. Zudem übernimmt er den Kontakt zum Makerspace-Team. Der Makerspace ist eine kreative Umgebung im Bühler zu der jeder Mitarbeiter Zugang hat. Dort steht uns eine Werkstatt zur Verfügung, in der es verschiedene Geräte, wie 3D-Drucker, Holz Laser, Akkuschrauber und vieles mehr hat. Der Ort ist ausgelegt, um schnelle Prototypen herzustellen und darf auch privat genutzt werden. Nachdem wir uns dem Team dort vorgestellt und ihnen unser Projekt erläutert haben, wurden wir herzlich empfangen.

Da wir schon zu Beginn wussten, dass wir den Greifautomaten elektronisch ansteuern werden, haben wir Christoph die Aufgabe gegeben, sich mit dem Arduino auseinanderzusetzen, da er bereits Kompetenzen in der Elektronik und der Programmierung besitzt. Ihm haben wir die ganze Elektronik und Programmierung anvertraut.

Nicola übernimmt die Verantwortung über die Dokumentation, kümmert sich um die grafische Visualisierung und unterstützt die anderen bei ihren Aufgaben.

Die Aufgabenverteilung haben wir im Projektjournal festgehalten.

4.4 Informationen für mögliche Einkaufskomponenten sammeln.

Es ist wichtig, sich schon früh im Internet und in Fachgeschäften nach möglichen Bauteilen umzusehen. So kann man einschätzen, was uns zur Verfügung steht und in welchem Rahmen die Preise ungefähr liegen werden.

Wenn wir ein interessantes und passendes Bauteil im Internet oder bei einem lokalen Händler gefunden haben, haben wir das im Projektjournal festgehalten.

5 Entscheiden

In der Entscheidungsphase werden verschiedene Varianten erarbeitet, ausgewertet und schlussendlich entschieden, welche der Varianten umgesetzt wird.

5.1 Funktionsstruktur

Mit der Funktionsstruktur (auch im Anhang) werden die wichtigsten Funktionen aus der Anforderungsliste nochmals aufgeteilt und visualisiert. Ganz oben in der Mitte ist die Hauptfunktion des Produktes. Dieses wird dann in kleinere Teifunktionen unterteilt. Die Teifunktionen werden anschliessend immer weiter unterteilt. Um der Funktionsstruktur noch ein wenig Charakter zu verleihen, haben wir bei unserem Beispiel noch den Greifer ins Design miteingebunden.

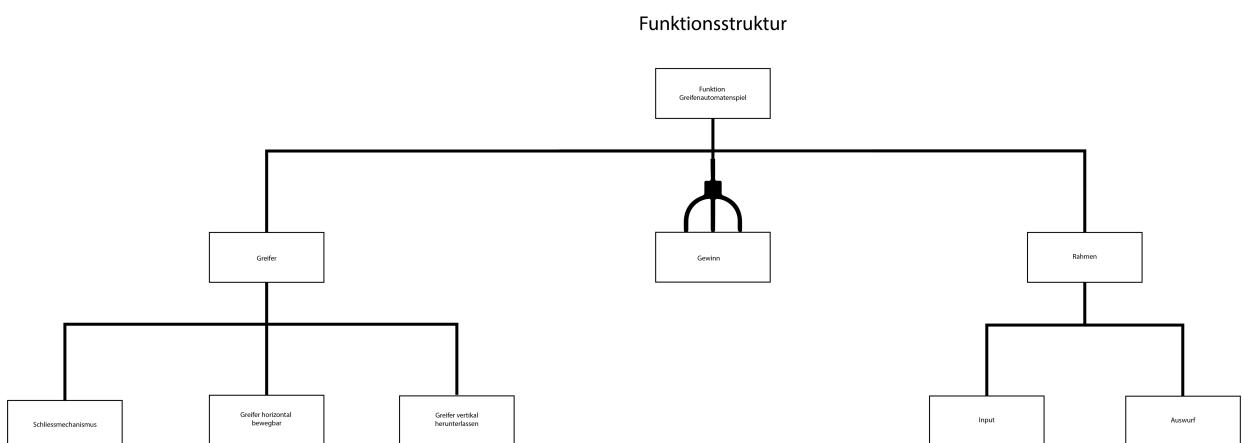


Abbildung 4: Funktionsstruktur

5.2 Morphologischer Kasten

Im morphologischen Kasten werden zu jeder Funktion aus der Funktionsstruktur mehrere Lösungsvarianten gesucht. Das ist einer der zeitaufwendigsten Punkte. Wir haben zuerst von Hand schnelle Skizzen von unseren Ideen gemacht, die wir dann im Morphologischen Kasten sauber dargestellt haben.

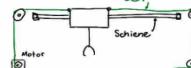
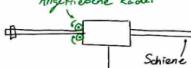
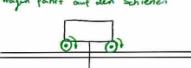
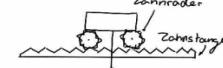
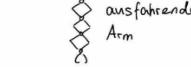
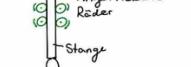
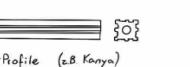
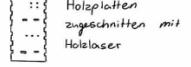
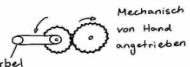
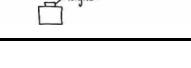
Morphologischer Kasten				
Teilfunktionen	Lösungsprinzipien			
Greifer horizontal Bewegen	A 	B 	C 	D 
Greifer herunterlassen				
Greifer				
Rahmenmaterial				
Steuerung / Interface				
Auswurf				
Gewinn				

Abbildung 5: Morphologischer Kasten

5.3 Prinzip Skizzen

Die vielen einzelnen Ideen aus dem Morphologischen Kasten werden jetzt zusammengefügt zu verschiedenen Varianten. Wir haben aus unseren Ideen vier Varianten erzeugt.

In der Ausbildung haben wir gelernt zu jeder Variante eine Skizze von Hand zu machen, die alle Funktionen darin darstellt. So hat man nicht nur eine Auswahl an Funktionen, sondern eine übersichtliche Zeichnung, die man später bewerten kann.

An der Stelle mussten wir darauf achten, dass wir im Zeitplan bleiben und nicht zurückfallen. Deshalb haben wir uns dazu entschieden keine Prinzip Skizzen zu erstellen. Damit wir trotzdem vier Varianten erstellen und später auswerten können, haben wir die ausgewählten Funktionen untereinander aufgeregelt und auf unser Vorstellungsvermögen vertraut.

Jede der vier Varianten wäre grundsätzlich realisierbar. Als Nächstes geht es daran die Varianten auszuwerten und die beste auszusuchen.

5.4 Vor - / Nachteilliste

Die Vor - / Nachteilliste ist weit verbreitet. Zu beachten ist, dass man hier die Punkte aus der Anforderungsliste als Bewertungskriterien nimmt. Bei jedem Punkt wird entschieden, ob er

als Vorteil oder Nachteil gilt. Am Ende fasst man alle Variationen zusammen, notiert und begründet, ob sie weiterverfolgt wird oder nicht.

Lösungsvariante 1		Lösungsvariante 2	
Vorteile	Nachteile	Vorteile	Nachteile
horizontale Bewegung vertikales herunterlassen Greifmechanismus Steuerung einfache Fertigung günstige Herstellung schnelle Herstellung	Produktausgabe kein toller Gewinn	horizontale Bewegung vertikales herunterlassen Greifmechanismus Produktausgabe Steuerung toller Gewinn einfache Fertigung schnelle Herstellung	keine günstige Herstellung
Fazit: Insgesamt werden technische Kriterien gut erfüllt (Platz 2 in S-Diagramm). Wirtschaftliche Kriterien sind sehr gut erfüllt (Platz 1 in S-Diagramm). Die Kombination von schneller, einfacher und relativ günstiger Herstellung mit der guten Funktionalität, macht diese Variante ideal für unser Projekt.		Fazit: Technisch ist diese Variante mit Abstand die beste und interessanteste. (Platz 1 in S-Diagramm). Jedoch ist sie deswegen auch schwierig, zeitintensiv und sehr teuer in der Umsetzung. (Platz 3 in S-Diagramm) Die hohen Kosten und die Zeitintensive Herstellung macht diese Variante für unser Projekt eher ungeeignet.	
Entscheid: Lösungsvariante 1 wird weiterverfolgt, weiterentwickelt und realisiert.		Entscheid: Lösungsvariante 2 wird nicht weiterverfolgt.	
Lösungsvariante 3		Lösungsvariante 4	
Vorteile	Nachteile	Vorteile	Nachteile
vertikales Herunterlassen toller Gewinn einfache Fertigung günstige Herstellung	horizontale Bewegung Greifmechanismus Produktausgabe Steuerung schnelle Herstellung	vertikales Herunterlassen Greifmechanismus toller Gewinn günstige Herstellung	horizontale Bewegung Produktausgabe Steuerung schwierige Fertigung lange Herstellungszeit
Fazit: Technisch hat diese Variante im S-Diagramm am schlechtesten abgeschlossen. Wäre aber unserer Meinung trotzdem durchführbar. Wirtschaftlich ist sie sehr ähnlich wie Variante 1. Durch die langen Wartezeiten der grossen 3D-Druck-Teile, ist sie jedoch nur auf Platz 2 im S-Diagramm. Da diese Variante technisch nicht sehr gut abgeschlossen hat ist sie weniger interessant als die Varianten 1 und 2		Fazit: Die Mechanische Steuerung ist zwar auch interessant, aber im Vergleich mit der Elektronik kann sie nicht mithalten. Es ist sehr günstig in der Herstellung, da Motoren, Sensorik usw. nicht gebraucht werden. Zudem werden keine teuren Maschinen wie Holzlasern gebraucht, und es könnte in einer einfachen Werkstatt hergestellt werden. Da wir schon sehr viel Zeit investiert haben, um mit dem Arduino Erfahrung zu sammeln, möchten wir den Greifautomaten unbedingt elektronisch machen.	
Entscheid: Lösungsvariante 3 wird nicht weiterverfolgt.		Entscheid: Lösungsvariante 4 wird nicht weiterverfolgt. (S-Diagramm wurde nicht ausgefüllt, da es für uns nicht in Frage kommt.)	

Abbildung 6: Vor- und Nachteilliste

5.5 Nutzwertanalyse mit S-Diagramm

Die Nutzwertanalyse ist ein weiteres wichtiges Werkzeug, um die verschiedenen Varianten zu bewerten. Sie ist jedoch nicht so bekannt, da sie etwas komplizierter ist als die ausgewählte Vor - / Nachteillisten.

Hier bedient man sich von wichtigsten Punkten aus der Anforderungsliste. Die Funktionen werden gewichtet. Wichtige Funktionen, wie das Bewegen des Greifers bekommen eine höhere Gewichtung als z.B. der Gewinn. Schliesslich folgt die Beurteilung der verschiedenen Funktionen und verteilt Punkte. Die einzelnen Punkte werden mit der Gewichtung multipliziert und am Ende kommt ein Wert raus, der aussagt wie gut die Varianten im Technischen und im Wirtschaftlichen Bereich abgeschlossen haben. Das S-Diagramm visualisiert das Ergebnis so dass man einen klaren Gewinner herauslesen kann.

Wir möchten hier erwähnen, dass wir Variante 4 nicht im S-Diagramm bewertet haben. Der Grund dafür ist, dass wir von Anfang an den Greifautomaten elektronisch ansteuern wollten und eine Manuelle Bedienung für uns nicht in Frage kam. Auch weil wir schon viel Ressourcen in die Arduino gesteckt haben.

KONZEPTBEWERTUNG

S-Diagramm

Projekt	Greifautomat	LEGENDE					
		Gewichtungsfaktor		Punkte		Ergebnis	
		10 = niedrig	100 = hoch				
		Punkte 0 = sehr schlecht	10 = hervorragend				
TECHNISCHE WERTIGKEIT							
Kriterien	Gewichtungsfaktor	<i>Lösungsvariante 1</i>		<i>Lösungsvariante 2</i>		<i>Lösungsvariante 3</i>	
horizontal verschiebbar	80	Punkte	7	Punkte	8	Punkte	5
vertikal herunterlassen	80	Ergebnis	560	Ergebnis	640	Ergebnis	400
Greifmechanismus	100		7		7		200
Produktausgabe	30		2		8		60
Steuerung	50		6		9		150
Toller Gewinn	10		5		9		60
Gesamt	350			2230		2760	
Rang			2		1		3
WIRTSCHAFTLICHE WERTIGKEIT							
Kriterien	Gewichtungsfaktor	<i>Lösungsvariante 1</i>		<i>Lösungsvariante 2</i>		<i>Lösungsvariante 3</i>	
einfache Fertigung	40	Punkte	7	Punkte	8	Punkte	7
günstig	100	Ergebnis	280	Ergebnis	320	Ergebnis	280
schnelle Herstellung	40		6		1		600
Gesamt	180			1160		700	
Rang			1		3		2

Abbildung 7: Nutzwertanalyse

S-DIAGRAMM

	Wertigkeit	
	Technisch	Wirtschaftlich
Lösungsvariante 1	2.23	1.16
Lösungsvariante 2	2.76	0.7
Lösungsvariante 3	1.43	1.08

Erklärung zum s-Diagramm:

Die Werte der Tabelle links sind aus der wirtschaftlichen und technischen Wertigkeits-Tabelle oben.
Die perfekte Lösung hätte einen Wert 1 in der technischen und der wirtschaftlichen Wertigkeit.

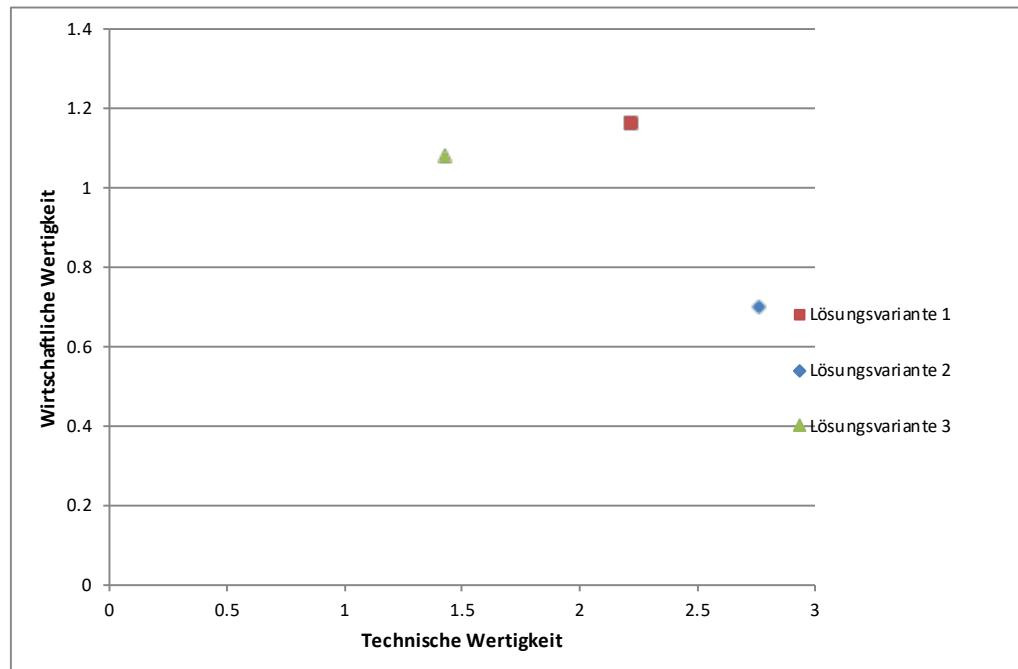


Abbildung 8: S-Diagramm

5.6 Entscheidung

Aus der Vor - / Nachteilliste sowie dem S-Diagramm kann bei unserem Projekt ein klarer Gewinner herausgelesen werden. Wie auf der Vor - / Nachteilliste bereits beschrieben werden wir die erste Variante weiterverfolgen. Im Moment werden daran keine Änderungen vorgenommen.

6 Realisieren

Endlich kann man mit dem Realisieren beginnen und sehen, woran man schon so lange gearbeitet hat. Voller Elan und Vorfreude waren wir bereit für die Herausforderung! Hier wird die ausgewählte Variante weiter ausgearbeitet und realisiert. Dank unserer super Planung haben wir den Überblick in der Umsetzung nie verloren und konnten stets darauf zurückgreifen.

6.1 Massstäblicher Entwurf

Als Konstrukteur versteht man unter einem massstäblichen Entwurf eine von Hand zusammengestellte Zeichnung mit Lineal. Die Normteile sind ausgewählt und alle offenen Punkte ausgearbeitet. In der Praxis trifft man das aber selten an. Denn von Hand zu zeichnen, beansprucht einfach zu viel Zeit. Meist geht man von den Skizzen direkt zum CAD. Dort wird der Greifautomat zuerst grob im 3D modelliert. Zudem überprüft man laufend, ob es so funktioniert, wie man es geplant hat und arbeitet sich immer mehr ins Detail. Der Vorteil davon ist, dass Fehler früh erkannt werden, bevor man viel Zeit verbracht hat. So kann man direkt reagieren.

6.2 Fertigungsunterlagen mittels CAD erstellen

Der grobe Entwurf vom Anfang wird jetzt immer mehr ausgearbeitet. Bei unserem Fall heisst das, Steckverbindungen an allen Holzteilen, Löcher Für Schrauben vorsehen, usw. Wenn das 3D-Modell fertig ist, geht es normalerweise an die Zeichnungen. Da wir die Teile selbst herstellen und dazu keine Normgerechte Fertigungszeichnungen brauchen, haben wir uns entschlossen keine Zeichnungen zu erarbeiten. Ausserdem besteht unser Greifautomat.

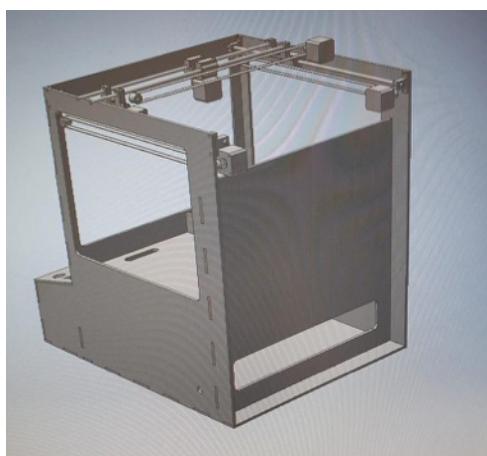


Abbildung 9: finales CAD-Modell

6.2.1 Probleme beim Holzlaser

Die exportierten Files aus dem CAD kann man über eine Software auf den Laser laden. Ähnlich wie bei einem Drucker. Wenn man am Laser die nötigen Vorbereitungen getroffen hat, kann man loslegen. Wir verwenden 6 mm dicke MDF Platten.

Die Anlage ist inzwischen schon in die Jahre gekommen und schneidet nicht überall gleich gut. Obwohl wir die Kanten zweimal bearbeitet haben, ist der Laser nicht an allen Stellen durchgekommen. Beim Abtrennen reißt das Material dann leicht aus. Das sieht scheußlich aus, ist aber in unserem Fall nicht funktionsrelevant. Darum haben wir uns entschieden es so zu belassen und die Bauteile zu verwenden, auch wenn sie nicht einwandfrei aussehen. Dazu muss man noch anmerken, dass wir das Material umsonst zur Verfügung gestellt bekommen haben. Das wollten wir nicht unnötig ausnutzen.

6.2.2 3D-Druck

Die CAD Files können vom Arbeitsplatz oder bequem von Zuhause aus über eine Webseite auf die 3D-Drucker geladen und ausgedruckt werden. Damit man den Prozess beobachten kann ist sogar ein Livestreamer der einzelnen Drucker verfügbar.

Das hört sich einfacher an als es im Endeffekt ist. Es gibt hier jegliche Punkte, die zu beachten sind. Vom Aufbau und der Gestaltung der Bauteile bis zum Positionieren im 3D-Drucker.

Als Konstrukteure haben wir schon öfters mit verschiedenen 3D-Druckern gearbeitet und einige Erfahrungen damit gesammelt. Beim Hochladen auf der Webseite kamen dann aber plötzlich kuriose Fehlermeldungen und wir waren mit unserm Latein am Ende. Damit wir hier nicht unendlich viel Zeit verbrauchen gingen wir auf die Mitarbeiter im Makerspace zu. Wir haben schnell sehr kompetente Hilfe bekommen und konnten unsere Bauteile übers Wochenende ausdrucken.

Die Ergebnisse waren nicht perfekt, aber sie erfüllten die Funktion. Wir wollten den Makerspace auch nicht unnötig belasten da sie gerade in dem Moment einen Corona-Fall hatten und dadurch der Arbeit kaum nachkamen.

An einigen Stellen musste dann noch das Stützmaterial abgetragen werden und die Löcher nochmal nachgebohrt werden.

6.3 Einkaufskomponente beschaffen

In den Herbstferien haben wir uns bei Christoph Zuhause getroffen, um die Einkaufskomponenten zu bestellen. Es war schwieriger als gedacht, die Komponenten in der richtigen Grösse zu finden. Besonders auf die Lieferzeiten mussten wir achten, denn bei diesem zeitlichen Projekt konnten durften wir nicht zögern.

Die meisten Bauteile konnten wir im Internet bestellen. Einiges mussten wir danach noch in lokalen Läden besorgen. Wir führen eine Liste mit allen Einkaufsteilen und den Kosten. Es ist erstaunlich, wie schnell man auch für so ein kleines Projekt einige Hundert Franken in die Hand nehmen muss.

6.4 Kostenliste

KOSTENRECHNUNG

Projekt Greifautomat

Datum 12.12.21

Greifautomat						
Baugruppen	Teil / Arbeitsschritt	Bemerkungen	Kosten pro Stk	Anzahl	Kosten gesamt	
Elektronik						
Zubehör für Arduino	Zubehör im Arduino Starter Kit (z. B. Handbuch, Steckboard, kleiner Motor, Bauteile)	CHF 53.70	1	CHF 53.70		
9V Batterie	Dient als externe Spannungsquelle für den Motor in der Testphase	CHF 5.00	1	CHF 5.00		
Stepper Motor Kit + Driver	TWO TREES® CNC Shield + UNO R3 Board +4x A4988 Stepper Motor Driver	CHF 57.20	1	CHF 57.20		
Schaltkreisplatte + Kleinteile	Distrelec Bestellung	CHF 57.00	1	CHF 57.00		
Step-Down Converter	Whadda Modul DC-DC Step Down LM2596S bei Ditec	CHF 14.70	1	CHF 14.70		
Total Elektronik			CHF 188.00		CHF 187.60	
Bauteile						
Schiene + Lagerblock	Amazon	CHF 16.25	4	CHF 65.00		
Greifarm	Acryl Mechanische Klaue 3D Druck	CHF 23.00	1	CHF 23.00		
Joystick + Druckknöpfe	Amazon	CHF 25.20	1	CHF 25.20		
Power Button	Amazon	CHF 10.80	1	CHF 10.80		
Zahnriemen Kit	Afuna 3D-Drucker-Zubehör	CHF 19.75	1	CHF 19.75		
Total Bauteile			CHF 95.00		CHF 143.75	
Kleinteile						
Kleinteile und Leiterplattenreiniger	erste Bestellung bei Reichelt	CHF 40.00	1	CHF 40.00		
Kleinteile	zweite Bestellung bei Reichelt	CHF 40.00	1	CHF 40.00		
Gummifüsse	Coop B+H	CHF 3.50	4	CHF 14.00		
Kugellager	ERNSTTOBLER&CO AG Nideruzwil	CHF 3.05	6	CHF 18.30		
Nylonkordel	Coop B+H	CHF 6.20	1	CHF 6.20		
Aufkleber	Bühler Printshop	CHF 7.00	1	CHF 7.00		
Total Kleinteile			CHF 99.75		CHF 125.50	
Kleinteile Landi						
Sechskantschraube M5x20 50 Stk.		CHF 5.95	2	CHF 11.90		
Unterlagscheibe M5 100 Stk.		CHF 2.50	1	CHF 2.50		
Sicherungsmuttern M5 100 Stk.		CHF 4.95	1	CHF 4.95		
Befestigungswinkel		CHF 0.95	16	CHF 15.20		
Schrauben-Set		CHF 7.60	1	CHF 7.60		
Total Kleinteile Landi			CHF 21.95		CHF 42.15	
TOTAL				CHF -	CHF 499	
PROJEKTKOSTEN TOTAL	Greifautomat			CHF 499		

Abbildung 10: Kostenliste

6.5 Probleme bei Bestellungen

Unsere Teile haben wir auf vielen verschiedenen Webseiten bestellt. Da kann man schnell den Überblick über verlieren. Dabei hat uns die Kontaktliste und Kostenliste geholfen.

Als die ersten Sachen ankamen, haben wir sie mit Freude ausgepackt und überprüft, ob alle Teile vorhanden sind.

Das bestellte Zahnriemen-Set hatte sehr lange Verzögerung. Als wir uns langsam Sorgen machten, haben wir dann die Mail bekommen, dass der Einkauf storniert wurde. Ein Grund wurde nicht angegeben. Im ersten Moment hat das bei uns für Panik gesorgt, denn der Zahnriemen ist ein wichtiger Bestandteil unseres Greifautomaten. Dann haben wir uns eifrig daran gemacht, Alternativen zu finden. Im Internet hatten wir zuerst keinen Erfolg da die Lieferzeiten dort viel zu lange waren und der Artikel an vielen Orten ausverkauft war. Dann kam uns die Idee bei den standard-Lieferanten von Bühler nachzufragen. Dort bekamen wir schnell kompetente Hilfe, da wir direkt auf die zuständigen Personen zugehen konnten. Aber auch hier würde es mit den Lieferzeiten knapp werden. Wir haben auch in vielen lokalen Baumärkten und Fachgeschäften für Normteile nachgefragt aber leider ohne Erfolg. Dann sind wir wie durch ein Wunder auf eine Webseite gestossen, die ein Zahnriemen-Set nach unseren Bedürfnissen anbietet und dessen Lieferzeit noch ausreichen würde. Wir haben direkt zugeschlagen. Als es dann angekommen ist, war die Erleichterung gross.

Bei der Schiene haben zwei Befestigungsteile gefehlt. Da wir keine Zeit haben, um auf eine neue Bestellung zu warten mussten wir eine Lösung dafür suchen. Nach kurzer Überlegung sind wir zum Schluss gekommen, dass wir die fehlenden Teile einfach mit gelaserten Platten nachbauen werden. Unser Nachgebautes Teil sieht leider nicht so elegant aus, wie die bestellten Teile aber es erfüllt die Funktion ganz okay. Somit reicht es für unseren Prototypen aus. Gerne hätten wir noch mehr Zeit investiert und die fehlenden Teile nachbestellt oder aus Metall selbst gefertigt. Dafür war uns aber die Zeit schon zu knapp.

Trotz unserer Überprüfung der Einkaufskomponenten sind uns nicht alle Mängel direkt aufgefallen. Einige Bauteile hatten nicht die Abmasse, die auf der Webseite beschrieben wurde. Auf einige Fehler konnten wir noch rechtzeitig reagieren und das 3D-Modell am CAD entsprechend anpassen. Ein paar Fehler haben wir aber erst bei der Montage bemerkt. Wie wir auf diese Probleme reagiert haben, werden wir im Abschnitt «Probleme bei der Montage» noch genauer erläutern.

6.6 Montage und Inbetriebnahme

Nachdem wir alle Teile hergestellt und geliefert bekommen hatten haben konnten wir endlich mit der langersehnten Montage starten. Voller neuer Motivation sind wir während den IDPA Lektionen jeweils zum Bühler gefahren, um dort im Makerspace alles zu montieren.

Zum Montieren brauchten wir nicht viel Werkzeuge, was für unseren Prototypen spricht. Die ganze Rahmenkonstruktion aus den gelaserten Holzplatten konnten wir mit Hilfen von kleinen Metallwinkel zuerst grob befestigen, um zu schauen, ob alles passt. Dann haben wir

Stück für Stück zusammengeklebt. Dafür benutzten wir einen schnelltrocknenden Holzkleber, der uns von einer Mitarbeiterin von Bühler empfohlen wurde. Ihre Erfahrung hat sich bestätigt. Nach ca. 10 Minuten war der Kleber schon ziemlich trocken und nach 30 Minuten war es Fest verbunden. Die Wirkung des Klebers hat uns persönlich überrascht. Es hat uns so überzeugt, dass wir der Meinung sind, dass es auch ohne die angeschraubten Befestigungswinkel funktioniert hätte. Aber mehr Stabilität schadet wohl nicht. Darum haben wir uns entschieden die Befestigungswinkel zusätzlich doch zu verbauen.

Als dann die ganze Rahmenkonstruktion getrocknet war, konnten wir die Teile aus dem 3D-Drucker und die Einkaufsteile einbauen.

6.7 Probleme bei der Montage

Als wir die Letzte Platte ankleben wollten, haben wir bemerkt, dass die unterste Steckverbindung nicht ganz passt. Das lag daran, dass wir nicht die optimale Reihenfolge gewählt haben, um die Platten zu verkleben. So haben sich die Holzplatten ein wenig verzogen und es passte nicht mehr. Um Den Fehler zu beheben, haben wir die klemmende Steckverbindung abgetrennt. Es hält trotzdem richtig gut und man sieht den Fehler kaum.

Als wir den Joystick an der Bedienplatte anschrauben wollten, haben wir bemerkt, dass der Lochabstand nicht übereinstimmt. Den Fehler konnten wir uns im Moment nicht erklären. Später haben wir bemerkt, dass im Internet ein fehlerhaftes Messbild angezeigt wurde. Da der Holzlaser zu diesem Zeitpunkt besetzt war, haben wir uns entschieden die Platte nicht nochmals neu zu machen, sondern die Löcher um 90 ° versetzt zu bohren. Somit hat es jetzt vier ungebrauchte Löcher in der Bedienplatte, die etwas unschön aussehen.

Das gelaserte Loch für den Einschaltknopf war ein wenig zu klein. Gerade so, dass wir ihn nicht einsetzen konnten. Das lag daran, dass wir den Durchmesser des Einschaltknopfes Zuhause ohne passendes Werkzeug gemessen haben. Die Messungenauigkeit mussten wir jetzt ausbaden. Nach einigen Minuten Feilen war das Loch dann genug gross und der Einschaltknopf konnte, wie vorgesehen, montiert werden.

Den Ausschnitt für das Display haben wir Rechteckig gemacht. Bei Montieren haben wir bemerkt, dass auf einer Seite am Display noch ein Schwarzes Teil ein wenig vorsteht. Schnell haben wir uns entschlossen, diesen Teil noch auszusägen und schön zu feilen. Jetzt passte das Display perfekt in die Vorgesehene Öffnung. Als Christoph den Apparat später verkaufte, musste er feststellen, dass er falsch herum montiert wurde. Die ganze Arbeit musste jetzt also auf der anderen Seite nochmals gemacht werden. Dafür ist es jetzt symmetrisch.

Der Lochabstand an den Lagerblöcken hat nicht mit den Massen in der Bestellung übereingestimmt. Die Teile für den 3D-Drucker konnten wir noch anpassen aber die Teile aus dem Holzlaser hatten wir bereits gefertigt. Deshalb musste eine Platte nochmals angepasst und gelasert werden.

Nachdem wir den grössten Teil montiert hatten, haben wir festgestellt, dass es sich fast nicht auf den Schienen bewegen liess und alles klemmte. Wir haben dann versucht. Die einzelnen Komponenten sorgfältiger und genauer anzuschrauben. Dabei haben wir bemerkt, dass es bei etwas Lockeren Schrauben besser läuft. Daraufhin haben wir einige Schrauben ein wenig gelockert. Jetzt kann man es auf den Schienen einwandfrei hin und her bewegen.

Um die Elektromotoren zu befestigen, benötigen wir lange M3 Schrauben. Solche hatten wir gerade nicht zur Hand. Da bemerkten wir, dass in den Elektromotoren selbst schon solche Schrauben eingeschraubt waren. Ideal in der Länge, wie wir sie brauchen. Also dachten wir, die Schrauben seien dafür vorgesehen, dass man die Motoren damit befestigen kann. Wir haben also die Schrauben rausgedreht und die Motoren damit montiert. Dann plötzlich fiel uns ein Spalt in einem der Motoren auf. Als wir ihn genauer untersuchten, hatten wir plötzlich die Hälfte des Motors in der Hand. Er ist auseinandergefallen. Es Stellte sich heraus, dass die verbauten Schrauben in den Elektromotoren nicht für die Montage vorgesehen waren, sondern ein wichtiger Bestandteil des Elektromotors. Wir konnten es dann ohne weitere Probleme wieder zusammensetzen und haben die Schrauben an ihren Platz zurückgetan. Es war ein kurzer, aber grosser, Schrecken. Auf jeden Fall war es auch interessant, so einen Elektromotor man von innen zu betrachten.



Abbildung 11: Frontplattenfertigung



Abbildung 12: Elia beim Feilen

6.8 Elektronik Experimentierphase

In der ersten Phase der Elektronik musste wir uns überlegen, wie wir das ganze umsetzen wollen. Eine Reihe von Fragen mussten beantwortet werden, bevor es überhaupt losgehen konnte. Umso mehr Schwierigkeiten tauchten mit der Zeit auf; den Umfang in Sachen Elektronik und Programmierung haben wir schlichtweg unterschätzt.

6.8.1 Arduino

Bereits zu Beginn war klar, dass wir ein Arduino-Board zur Umsetzung nutzen werden. Arduino ist eine aus Soft- und Hardware bestehende Physical-Computing-Plattform, die mit einem ATmega328 Mikrocontroller bestückt ist, (Arduino Uno Specification, 2021). Sie liefert eine Open-Source IDE mit sich, sodass der Controller sehr einfach programmiert werden kann. Der Arduino eignet sich besonders für Anfänger, die möglichst unkompliziert in die Welt der Elektronik und Programmierung eintauchen wollen. Man muss sich dabei keine Gedanken zu Spezifikationen, Bibliotheken und der ganzen Hardware machen, Arduino bietet quasi ein komplettes plug n'play-System mit gut verständlicher Syntax und einer umfassenden Dokumentation, sodass die Umsetzung von Prototypen viel leichter fällt (Arduino Documentation, 2021).

Wir haben uns ein Arduino Starter-Kit bestellt, dieses beinhaltet eine Reihe von Bauteilen und ein Beginner-Guide, sodass wir mit ersten Tests sofort loslegen konnten.

6.8.2 Motor Tests

In der ersten Phase haben wir mit dem Arduino und dem Steckboard experimentiert.

Obschon eine gewisse Vorerfahrung in Elektronik und Programmierung bestand, haben wir nie ein Projekt in diesem Umfang umgesetzt. Ausserdem hatten wir beispielsweise nie einen Motor angesteuert.

Das Starter-Kit von Arduino beinhaltet einen DC-Motor, H-Brücke und einen MOS-FET. Mit diesen Bauteilen, und der im Buch enthaltenen Anleitung konnten wir einen Motor zu Testzwecken relativ schnell ansteuern. Die Programmierung für die Testschaltungen werden wir hier nicht näher erläutern.

Der erste Versuchsaufbau hat lediglich den Motor angesteuert, solange man den Schalter drückte. Der Arduino läuft bei einer Betriebsspannung von 5V und kann maximal 20 mA pro I/O-Pin liefern, weshalb der MOS-FET hier als Schalter für den Motor fungiert (Arduino Uno Specification, 2021). Der Motor selbst wird mit einer 9V-Batterie betrieben.

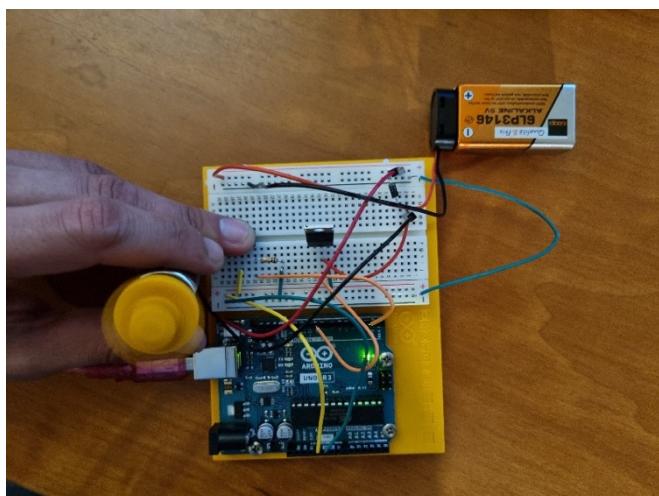


Abbildung 13: Erster Versuchsaufbau mit rotierendem Motor, MOS-FET Schaltung

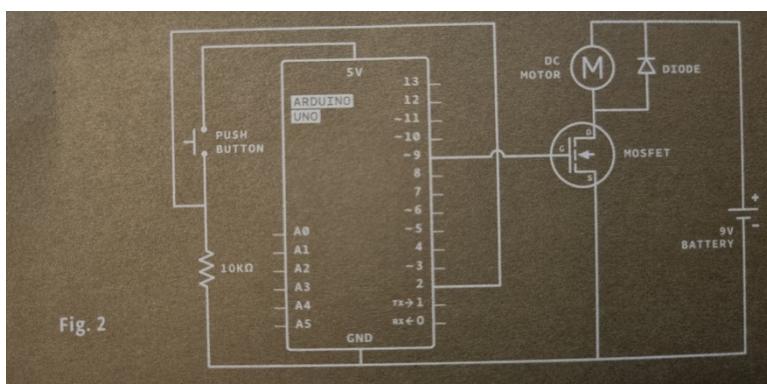


Abbildung 14: Erster Schematischer Aufbau (Arduino SA, 2015, p. 96)

Im zweiten Versuchsaufbau haben wir den MOS-FET durch eine H-Brücke ersetzt (Texas Instruments H-Bridge Datasheet, 2021). Diese eröffnet uns noch mehr Möglichkeiten mit dem Motor. Mit ein paar Anpassungen in Schaltung und Programmierung konnten wir den Motor nun bereits An- und Ausschalten, die Richtung ändern und mit einem Potentiometer die Geschwindigkeit beeinflussen.

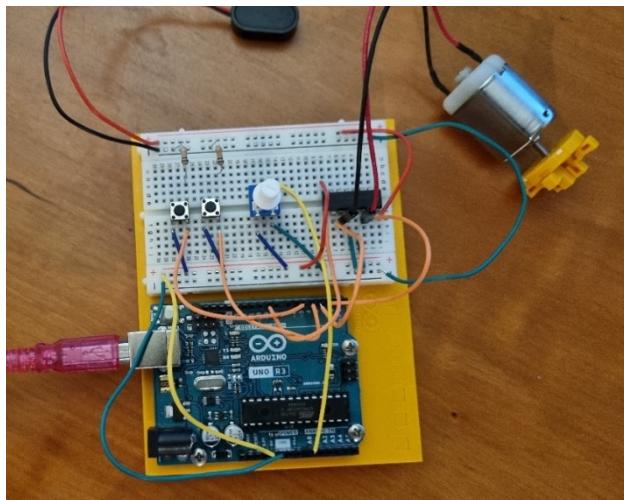


Abbildung 15: Zweiter Versuchsaufbau mit H-Brücke

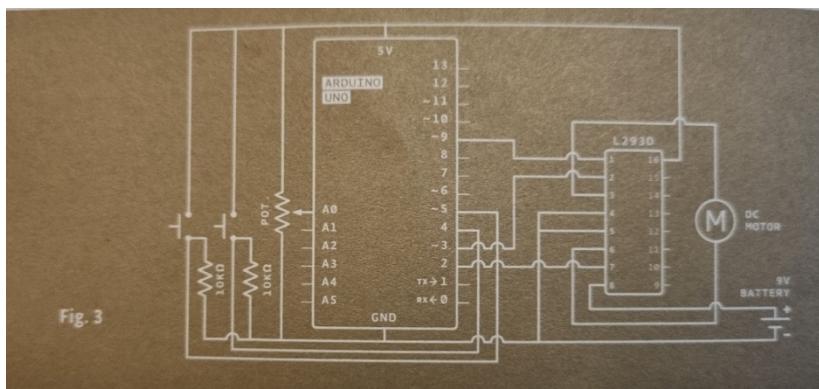


Abbildung 16: Schema mit H-Brücke (Arduino Projects Book, 2015, p. 104)

Im dritten Versuchsaufbau haben wir den Taster durch einen Kippschalter ersetzt, den wir noch herumliegen hatten. Mit dem Kippschalter konnten wir in Theorie einen Joystick simulieren. Ziel war es nämlich, am Schluss einen Joystick für die Steuerung der Motoren zu nutzen. Da ein Joystick ein Schalter mit 4 bis 8 Richtungen darstellt, musste man sich dazu vorsätzlich bereits Gedanken machen.

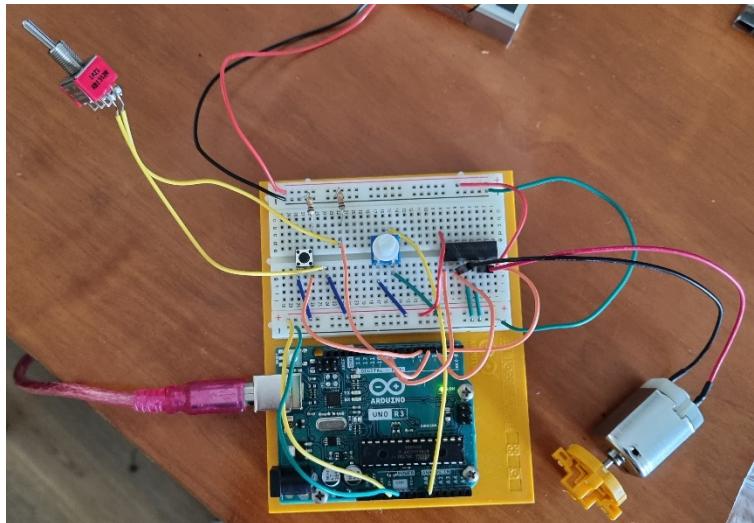


Abbildung 17: Aufbau der Schaltung mit Kippschalter

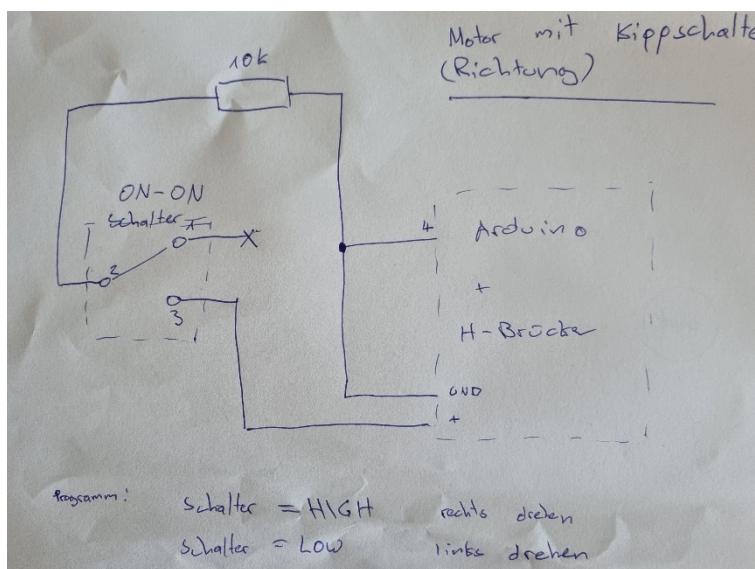


Abbildung 18: Schema und Code-Logik Kippschalter

6.8.3 LCD Tests

Neben den Motoren war im Kit auch ein LCD enthalten (16x2 LCD Datasheet). Es ist ein 16x2 Display, was bedeutet, dass er sich zur Darstellung von bis zu 32 Zeichen eignet. Auch hier lieferte der Guide ein Schema zur Beschaltung mit dem LCD (Arduino SA, 2015, p. 114). Sobald wir aber unser erstes Hello World auf dem Display hatten (SM, 2019), haben wir das Ganze mit dem Motor von vorhin verschmolzt. Am Schluss konnten wir den Motor steuern, sowie die vergangene Zeit und ein paar Zeilen Text auf dem Display anzeigen lassen.

Für diesen Versuchsaufbau hatten wir kein Schema. Bei den meisten, folgenden Schaltungen wurde kein Schema gezeichnet, erst für die definitive Schaltung. Dies, um den Aufwand zu reduzieren.

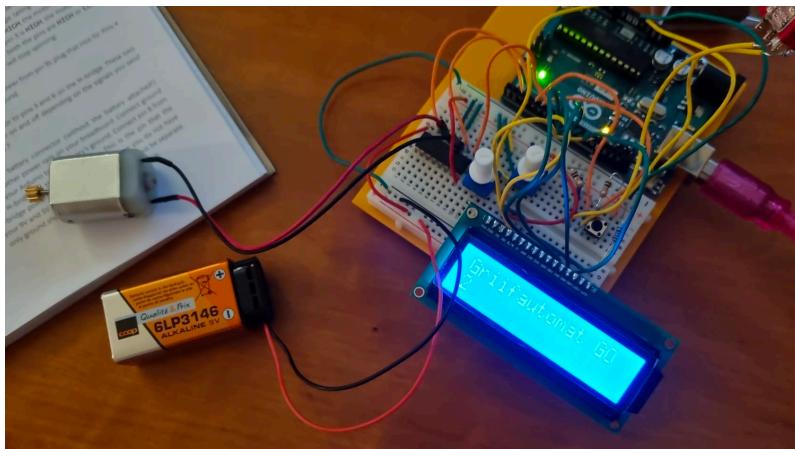


Abbildung 19: LCD Test mit H-Brücke

6.8.4 CNC Shield, Greifer und On-Off

Nach dieser ersten Steckboard-Phase hatten wir ein ungefähres Bild von ersten Bauteilen, welche wir benötigten. Wir haben daraufhin fleissig Bauteile bestellt, darunter beispielsweise den Greifer, ein CNC-Shield mit Treibern und Stepper-Motoren (da wir mehr Motoren mit grösserer Leistung brauchten), weitere Arduinos und einen sehr schönen On-Off-Schalter.

Ein CNC-Shield ist ein Modul, dass sich eignet, um Stepper-Motoren mit hoher Leistung einfach anzusteuern. Wir haben den Greifer sowie das CNC-Shield in China bestellt, da es sehr viel Geld spart. Problematisch dabei war, das chinesische Hersteller kaum Angaben geschweige denn ein Datenblatt mitliefern. Wir mussten uns mit dem Internet behelfen, um herauszufinden, wie das CNC Shield angeschlossen werden muss, wie man es betreibt, wie man es ansteuert und welche Pins im Arduino belegt sind (Electronic Clinic Blog, 2021) (Electronic Clinic, 2020) (crix, 2020).

Den On-Off Schalter haben wir bei Amazon bestellt, und auch dieser kam leider ohne Datenblatt. Im Internet hat man kaum Angaben zu diesem spezifischen Schalter gefunden, weshalb nichts als Testen mit dem Multimeter blieb.

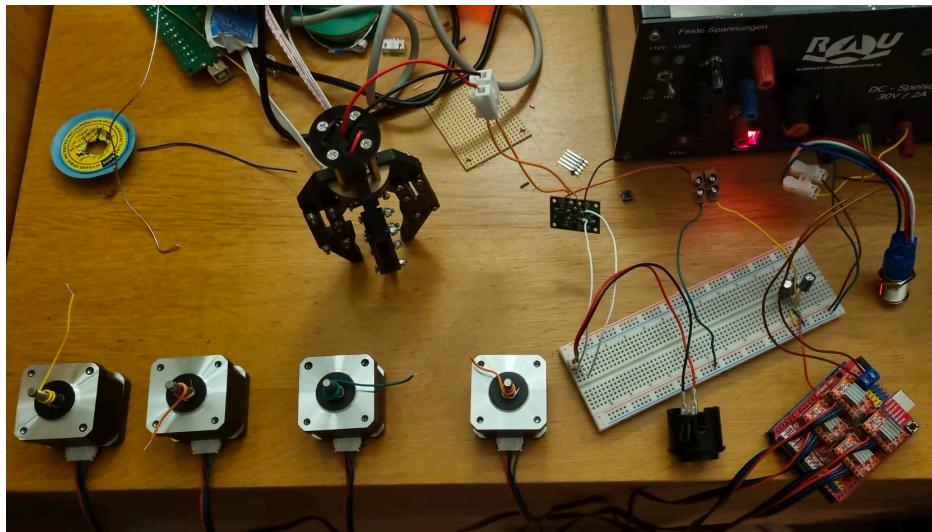


Abbildung 20: CNC Shield mit Stepper-Motoren und Greifer

6.8.5 I2C Kommunikation

Hier zeigte sich auch ein erstes Problem, dass uns noch sehr viel Zeit kosten würde. Es schien, als würde das CNC Shield praktisch alle Pins des Arduinos aufbrauchen. Wir hätten somit nicht genug, um LCD, Buttons und Joystick anzusteuern. Also müsste ein zweiter Arduino her, der aber mit dem ersten Arduino kommunizieren müsste. Um zu testen, ob dies den auch funktionieren würde, haben wir einen Arduino Nano hinzugenommen und diese Arduinos mittels einem I2C-Bus kommunizieren lassen (Mallari, 2020).

6.8.6 Analoge Joystickabfrage

Wir hatten die I2C Kommunikation getestet, es schien uns aber dennoch unnötig, zwei Arduinos verbauen zu müssen. Um drei Pins zu sparen haben wir daher mit dem analogen Lesen eines Inputs experimentiert (Arduino Documentation, 2021). Nach langem Überlegen und Testen kamen wir schlussendlich auf folgende Schaltung:

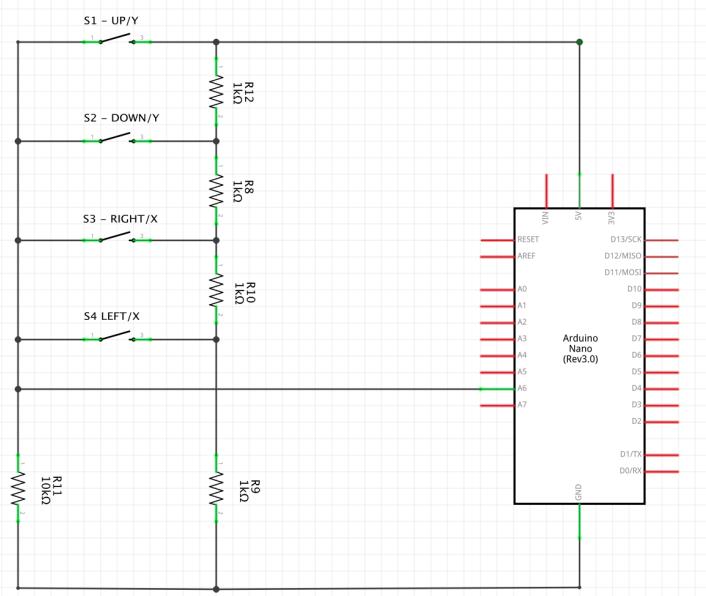


Abbildung 21: Schematische Darstellung der analogen Joystickabfrage

Die vier Schalter repräsentieren den Joystick, den wir am Schluss verbaut haben. Ziel dieser Schaltung ist es, alle Richtungen des Joysticks mit nur einem Pin zu lesen, statt insgesamt vier Pins dafür zu brauchen. Wir lösen diese Aufgabe ganz einfach mit einem Spannungsteiler. Je nachdem wie der Joystick steht, verändert sich der Widerstand, was wiederum zu verschiedenen Spannungsabfällen führt. Diese Differenz machen wir uns zunutze, um klare Signale für jede Richtung auszulesen.

Wir haben zwischen den Schaltern jeweils einen 1k Ohm Widerstand verbaut. Wir haben alle gleich dimensioniert, sodass die Spannung in regelmässigen Abständen fällt. Die 5V Eingangsspannung fällt also in vier Schritten nach unten.

Nach oben fahren wird bei 5V ausgelöst, nach unten fahren bei 3.75V, nach rechts bei 2.5V und zu guter Letzt nach links bei 1.25V. GND bzw. 0V bedeutet nichts machen.

```
// previousUpState = !previousUpState;
Serial.begin(9600);

}

void loop() {
    sensorValue = analogRead(sensorPin);
    int test = map(sensorValue, 0, 1023, 0, 3);

    if (sensorValue != 0) {
        Serial.println(sensorValue);
        digitalWrite(ledPin, HIGH);
        delay(sensorValue);
        digitalWrite(ledPin, LOW);
        delay(sensorValue);
    }

    /*
    switch (sensorValue) {
        case 0:
            stateDelay = 1000;
    }
}

```

The screenshot shows the serial monitor window of an Arduino IDE. The title bar says "/dev/cu.usbmodem11401". The main area displays a continuous stream of data from the serial port. The data consists of timestamped values ranging from 323 to 916. Below the serial monitor are several control buttons: 'Send', 'Autoscroll', 'Show timestamp', 'Newline', '9600 baud', and 'Clear output'.

Abbildung 22: Serieller Output beim Betätigen des Joysticks

Der Arduino beinhaltet einen Multikanal 10-bit Analog-Digital-Converter, dies bedeutet, dass er einen analogen Input in Abstufungen von 10 Bit (1024 Abstufungen) aufteilen kann (Arduino Documentation Analog Read, 2021). Im Programm ist zu erkennen, dass wir für verschiedene Richtungen auch wie erwartet verschiedene Werte bekommen.

Wir können dies sogar nachrechnen und überprüfen, welche Richtung wir zu diesem Zeitpunkt betätigten hatten. Nehmen wir die Zahl 511, die der Sensor gelesen hat.

Bei $U_{max} = 5V$, 1024 Abstufungen, und 511 als Output ergibt sich für U_e :

$$U_e = 5V / 1024 * 511 = 2.49 V$$

Wie man sieht, stimmt die Spannung mit unserer Planung überein, wir erhalten beim Output 511 eine Spannung von ungefähr 2.5 V, der Joystick wurde während dieser Eingabe also nach rechts bewegt.

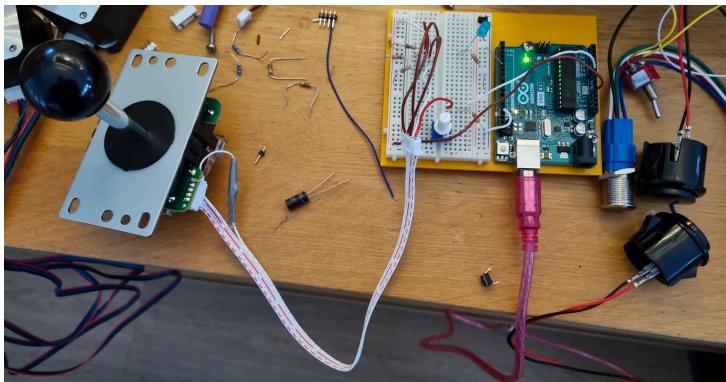


Abbildung 23: Aufbau der Joystickabfrage

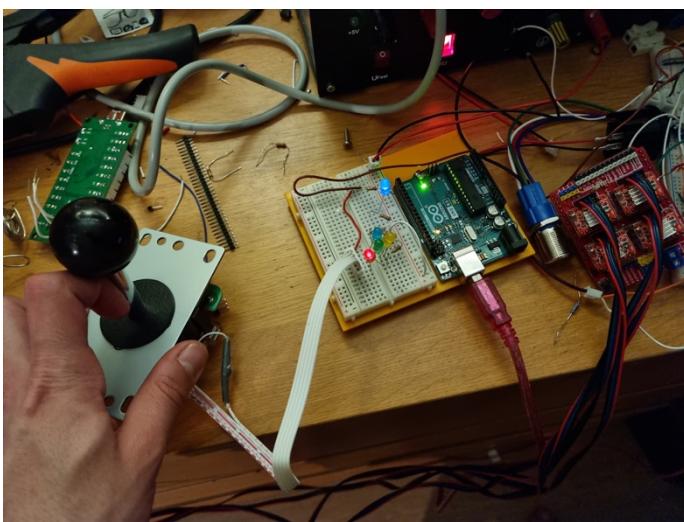


Abbildung 24: Joystickabfrage visualisiert mit LEDs

6.8.7 Umstieg auf Arduino Mega

Obwohl die Lösung der I2C Kommunikation funktioniert hätte, wäre es unschön und vom Aufwand, sowohl in der Programmierung wie auch in der Hardware, grösser gewesen. Wir wussten bereits, dass es auch grössere Arduinos gibt, der sogenannte Arduino Mega. Theoretisch könnten wir den Joystick nun auch digital auslesen, da wir mehr Pins zur Verfügung haben. Dies würde aber in der Hardware mehr Aufwand mit sich bringen, vor allem da unsere Testschaltung bereits funktioniert.

Der Arduino Mega eignet sich auch für höchst komplexe Projekte, die sehr viele Pins benötigen (Arduino Mega Specification, 2021). Wahrscheinlich ist er für unser Vorhaben sogar zu gross, wir brauchen bei weitem nicht so viel. Es erleichtert aber die Hardware immens, da wir ja kein richtiges Produkt entwickeln. In Realität wurde man hier seinen Mikrocontroller spezifisch anpassen und nach seinen Bedürfnissen dimensionieren, dies würde den Rahmen dieser IDPA aber sprengen, weshalb der Arduino Mega der perfekte Kompromiss ist.

6.8.8 Probleme der Bauteilbestimmung

Generell gestaltete sich die Bestimmung der Bauteile gerade in der Elektronik als langwierig und kompliziert. Da wir lediglich unser Arduino Kit als Grundausstattung hatten, musste wir uns viele Gedanken dazu machen, welche Bauteile wir den wirklich bräuchten.

Das Problem dabei ist, dass wir Ideen nicht einfach ausprobieren können. Der Umstand, dass wir schlecht ausgerüstet sind, machte die ganze Entwicklung um einiges schwieriger. In Realität steht einem Team bei der Produktentwicklung ein breites Sortiment an Bauteilen und Werkzeugen zur Verfügung, sodass ein Prototyp rasch und unkompliziert realisiert werden kann. Das sogenannte Try-and-Error-Verfahren kommt bei jeder guten Produktentwicklung zur Anwendung, den man kann noch so gut planen, es treten immer unvorhergesehene Effekte auf.

6.9 Elektronik Umsetzungsphase

Die definitive Umsetzung und Bestückung der elektronischen Komponenten brauchte viel mehr Zeit als erwartet. Dies war es auch, was wir unterschätzten hatten.

In vorherigen Versuchen hatten wir kaum Probleme, das Zusammenstecken und Programmieren mit dem Arduino erwies sich als ein Kinderspiel. Also schien es, als könnte man das Ganze ohne weitere Probleme umsetzen. Und in Theorie stimmt das auch, die Funktion könnten wir ohne Probleme gewährleisten.

Die Umsetzung inklusive Fehlerbehebung des Prototypen hat allerdings mehr Zeit in Anspruch genommen als es in der Planung vorgesehen war. Dies führte zu etwas Zeitdruck, da wir der Planung Anfangs mehr Gewichtung schenkten als der Umsetzung. Glücklicherweise würde uns dieser Umstand aber noch rechtzeitig bewusst werden.

6.9.1 Spannungsregler und Motortreiber

Man könnte sagen, dieses Erlebnis fungierte als kleiner Schlüsselmoment, der dazu führte, dass wir der Umsetzung in der Elektronik mehr Beachtung widmeten.

In einem mühsamen Akt haben wir ein Board mit Kondensatoren und einem Spannungsregler (Texas Instruments LM7805 Datasheet, 2021) konzipiert. Dieser sollte später als Spannungsquelle für den Greifer und den Arduino dienen.

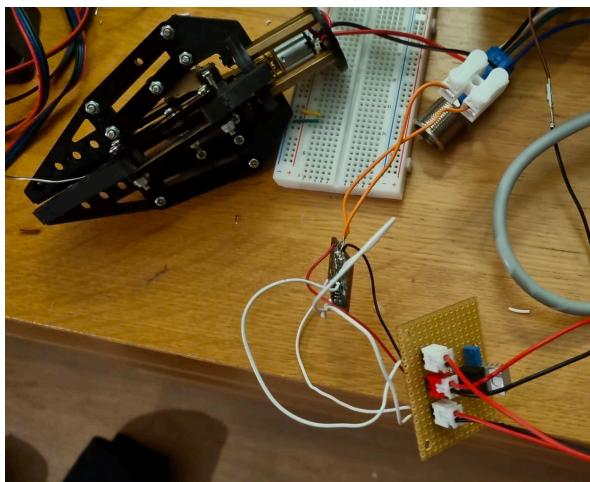


Abbildung 25: Spannungsregler mit Motortreiber

Nur für dieses kleinen Ding hat es Stunden an Arbeit gekostet, einfach weil es nicht funktionieren wollte. In Sachen Elektronik war dieser Tag regelrecht demotivierend.

Irgendwann hatte es aber funktioniert, wir konnten den Greifer betreiben. Nun aber stellten wir die ersten Probleme fest, an die wir zuvor nicht gedacht hatten:

Der Greifer verträgt bis zu 6 V (N20 DC Motor Datasheet), wir hatten aber nur einen 5 V Spannungsregler zur Verfügung (Texas Instruments LM7805 Datasheet, 2021). Diese 1 V Differenz führten bereits dazu, dass der Greifer merklich langsamer fährt. Dies wäre aber dennoch akzeptabel gewesen, wäre da nicht die grosse Überraschung gewesen:

Der Spannungsregler wird sehr heiß. Bereits bei kleinen Spannungen bemerkte man eine starke Hitzeentwicklung. Da unsere Stepper-Motoren aber einiges an Power brauchen, war

zu Beginn klar, dass wir ein 24 V-Netzteil brauchen werden. Wir mussten nachforschen, wie genau ein Spannungsregler funktioniert.

Ein Spannungsregler kann eine Eingangsspannung auf eine stabile Ausgangsspannung regeln. Dies war uns klar, jedoch wussten wir nicht, dass er dies erreicht, indem er die Energie schlichtweg in Wärme übersetzt (Forumsdiskussion zu LM7805 , 2006).

Dies mag für kleine Spannungen funktionieren, wird aber bei zunehmender Differenz höchst ineffizient und vor allem heiss. Würden wir unsere 24 V auf 5 V regeln, würde dies bei einem Strom von 0.5 A (Greifer Motor + Arduino Spitzenstrom) zu 9.5 Watt an Energie führen, die wir als Wärme verbraten würde.

Bei Messungen zeigte sich, dass die vier Stepper-Motoren mit einem Arduino (die grössten Stromfresser) eine Spitzenleistung von bis zu 30 Watt erreichen. Man müsste das Netzteil also um 10 Watt grösser dimensionieren, nur um diese Energie als Hitze wieder verbrauchen zu können. Nicht zu vergessen ist auch der Kühlkörper, die Notwendigkeit eines Bauteils das mehr Hitze aushält (Leistungsspannungsregler) und im schlimmsten Fall noch ein Lüfter der nötig wäre. Insgesamt wären diese Vorhaben ineffizient, kostspielig und schwierig zu verbauen gewesen. Also mussten wir das Board verwerfen.

Zur Krönung ging dann noch der Motortreiber des Greifers kaputt, dies war eine Folge des Löten mit schlechter Ausrüstung und vor allem der schlechten Verarbeitung des Treibers (China-Ware).

6.9.2 Motortreiber mit Arduino

Nachdem der Motortreiber des Greifers kaputt ging, musste dafür natürlich eine Lösung her, noch bevor wir uns dem Problem des Spannungsreglers resp. der Hitze widmen konnten.

Bei dieser Gelegenheit wollten wir es gleich richtig machen und den Motor auch bei 6 V betreiben, sodass der Greifer schneller geöffnet resp. geschlossen werden kann. Außerdem mussten wir den Treiber für den Greifer ersetzen, wobei es aber das Problem gab, dass wir Ersatz für diesen nur schwer gefunden hätten. Der Treiber kam zusammen mit dem Greifer aus China, es war nur schon schwierig herauszufinden, was für ein Treiber überhaupt verwendet wurde. Wir entschieden uns kurzerhand für die Steuerung über den Arduino mit unserer zuvor getesteten H-Brücke. Wir mussten sowieso schon auf den Arduino Mega aufrüsten und hätten somit genug Pins, um auch diese Funktion abzudecken. Vorteil dieser Lösung ist es auch, dass der Greifer nun im Programm berücksichtigt werden kann. Sonst wäre er aus

logischer Sicht vom Rest der Maschine isoliert gewesen, was bedeutet, man könnte den Greifer nicht automatisch abschalten, wenn das Spiel zu ende wäre.

Da wir den Joystick bereits über einen analogen Input lesen, stellte es kein Problem dar, zwei Schalter über je einen digitalen Input zu lesen, da der Anspruch hier tiefer ist.

6.9.3 Step Down Converter

Es musste eine Lösung her, um eine Gleichspannung effizient, also ohne Wärmeerzeugung, auf ca. 6 V zu regeln. Uns kam dabei gleich ein Schaltnetzteil in den Sinn, denn Schaltnetzteile lösen eine ähnliche Aufgabe und können Wechselstrom effizient in Gleichstrom umwandeln. Wir brauchten aber ja eine DC/DC Konvertierung, also fiel dies Weg.

Es gibt noch Bauteile namens DC/DC-Wandler, zu Beginn dachten wir, solche könnte unser Problem lösen. Es ist aber schwierig DC/DC-Wandler mit unseren gewünschten Spezifikationen zu finden. Außerdem sind sie nicht sehr flexibel, falls es Anpassungen geben sollte.

Beim Durchstöbern eines Forums sind wir dann auf die perfekte Lösung gestossen, einen Step-Down Converter oder auch Schaltregler genannt (Forumsdiskussion zu LM7805 , 2006). Dieser ist dazu fähig, eine Eingangsspannung zu einer definierten Ausgangsspannung zu regeln, ohne dabei heiss zu werden bzw. die Energie in Hitze umzuwandeln (Texas Instruments LM2596 Datasheet, 2021). Er funktioniert dabei ähnlich wie ein Schaltnetzteil, es war also genau das, was wir brauchten.

Um den Aufwand zu reduzieren, haben wir ein fertiges Modul mit einem Potentiometer bestellt (Whadda Modul Step-Down Converter, 2021). Wir konnten nun unsere 24 V Eingangsspannung problemlos regeln, das Potentiometer auf dem Modul dient zum Einstellen unserer gewünschten Spannung.

Zeitgleich haben wir festgestellt, dass obwohl der Arduino Mega bei 6 V betrieben werden könnte, sind trotzdem mindestens 7 V empfohlen, um eine flüssige Funktion zu gewährleisten (Arduino Mega Specification, 2021).

Da wir sowieso noch den Arduino Mega bestellen mussten, haben wir auch gleich einen 6 V Spannungsregler bestellt, was uns erlaubte, den Step-Down Converter für den Arduino Mega zu nutzen. Wir können mit dem Step-Down Converter den Arduino nun bei 7 V betreiben, und den Motor für den Greifer betreiben wir mit dem Spannungsregler. Dieser wird etwas heiss, aber eine Differenz von 1 V erzeugt bei weitem nicht so viel Hitze wie eine von 18 V.

6.9.4 Schema

Wir hatten nun einiges experimentiert und uns viele Gedanken zur Umsetzung gemacht. Es kam die Zeit, an dem wir ein definitives Schema zeichnen mussten. Da die ganze Elektronik nun doch einige verschiedene Funktionen umfasste, würde ein Schema gerade bei der Be-stückung und Programmierung immens helfen.

Das Zeichnen des Schemas war ein Prozess und auch dieses wurde laufend verändert. Anfangs hatten wir mit zwei Arduinos geplant, der Entscheid für den Arduino Mega fiel spät. Das Schema musste daraufhin wieder angepasst werden.

Im Folgenden beleuchten wir kurz die Schlussversion des Schemas (siehe Anhang für komplettes Schema). Das Schema für das CNC Shield wurde vereinfacht dargestellt, mit den Pins, die wir effektiv brauchten. Dies nur um einen generellen Überblick auch zum CNC Shield zu schaffen, was wiederum der Programmierung zugutekommt.

Das ganze Schema wurde mit Fritzing gezeichnet.

Die Spannungsversorgung erfolgt über die Steckdose, welche 230 VAC liefert. Wir haben einen Power Button dazwischen, mit welchem man das Gerät einschalten kann. Das 24 V Netzteil (V2) wird von der Steckdose versorgt. Das Netzteil selbst liefert eine stabile Ausgangsspannung von 24 VDC. Hier ist das Netzteil als eine Batterie dargestellt, das richtige Schema würde anders aussehen. Keinesfalls sollte man eine Batterie parallel zur Steckdose schalten, es ist hier lediglich zur Veranschaulichung, da wir keine Netzteile als fertiges Modul im Programm hatten. Zwischen Netzteil und dem Ausgang haben wir noch eine 2 Ampere Sicherung eingebaut, welche zum Schutz dient.

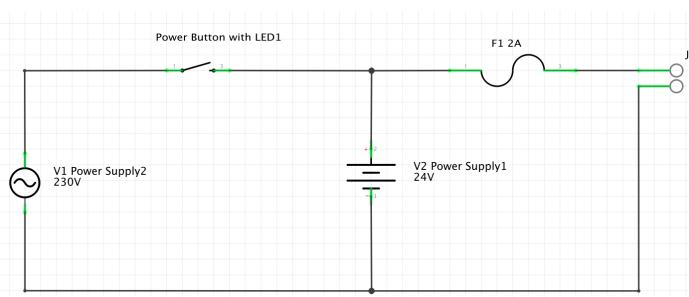


Abbildung 26: Schematische Darstellung der Spannungsversorgung (AC)

Die nächsten drei Ausschnitte zeigen die Spannungsversorgung verschiedener Elemente. Zuerst wäre die Versorgung des CNC Shields resp. der Stepper-Motoren, welche direkt mit 24 V aus unserem Netzteil erfolgt. Im zweiten Bild sehen wir den Step-Down Converter (V2),

dieser sorgt für eine stabile Spannung von 7 V, ohne dabei Hitze zu Erzeugung. Mit diesem wiederum betreiben wir einerseits den Arduino Mega, andererseits aber auch den Spannungsregler im letzten Bild. Der Spannungsregler liefert uns 6 V für unseren Greifer, die Differenz von 1 V übersetzt er dabei in Wärme. Bei einem Spitzstrom von 0.3 A (Messung des Greifmotors bei hoher mechanischer Last) ergibt dies eine Leistung von gerade mal 300 mW, was einen Kühlkörper hinfällig macht (Electronics Forum Heatsink).

$$U_{\Delta} = 1 \text{ V}, I = 0.3 \text{ A}$$

$$P = 1 \text{ V} * 0.3 \text{ A} = 300 \text{ mW}$$

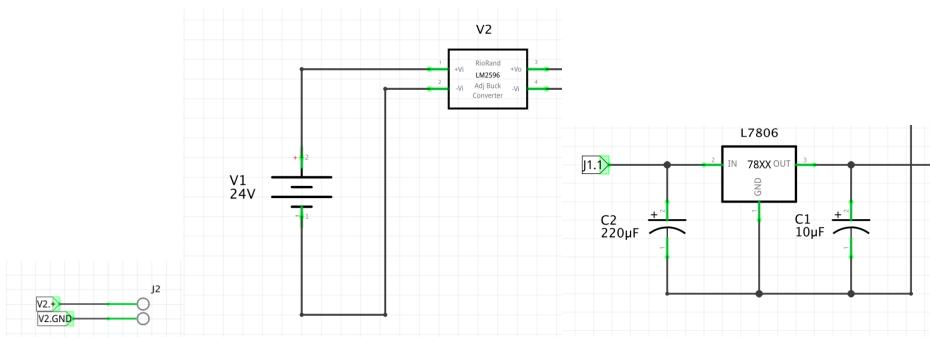


Abbildung 27: Schematische Darstellung der Spannungsversorgung (DC)

Im folgenden Bild sehen wir die H-Brücke und den Motor des Greifers. Die H-Brücke stellt unseren neuen Treiber für den Greifer dar. Wir brauchen die H-Brücke für die Stepper-Motoren nicht, hatten aber eine Schaltung mit dieser ausprobiert. Nun konnten wir diese als neuen Treiber einsetzen, sodass wir uns nicht um einen Ersatz kümmern mussten.

Unser Festspannungsregler von vorhin liefert die 6 V an die H-Brücke, diese wiederum schaltet den Greifer in die eine oder andere Richtung. Die Steuerung erfolgt über den Arduino Mega, weshalb noch Datenleitungen zwischen dem Mega und der H-Brücke bestehen.

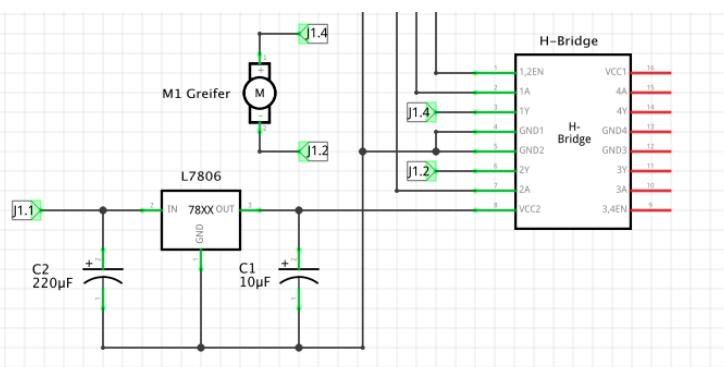


Abbildung 28: Schematische Darstellung der H-Brücke

Dieser Teil der Schaltung (im Schema um 90° gedreht) repräsentiert die Eingabe für den Motor des Greifers. Wird ein Button gedrückt, so erkennt der Arduino (Datenleitung GCLOSE/GOPEN), dass dort eine Spannung anliegt. Wir können dieses Signal nutzen, um die H-Brücke anzusteuern, sodass der Greifer in Bewegung kommt.

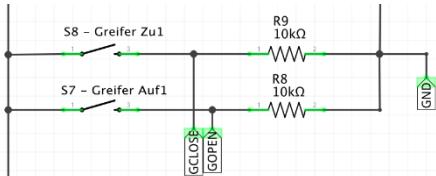


Abbildung 29: Schematische Darstellung der Eingabe des Greifers

Dieser Teilausschnitt zeigt die Beschaltung des LCDs. Es sieht sehr wild aus, beinhaltet aber vor allem Datenleitungen, die zum Arduino müssen und dort einfach initialisiert werden. Dies geschieht über ein Flachbandkabel, FBK.X bezeichnet den jeweiligen Pin des Kabels. Mit dem Potentiometer R13 kann man außerdem den Kontrast einstellen.

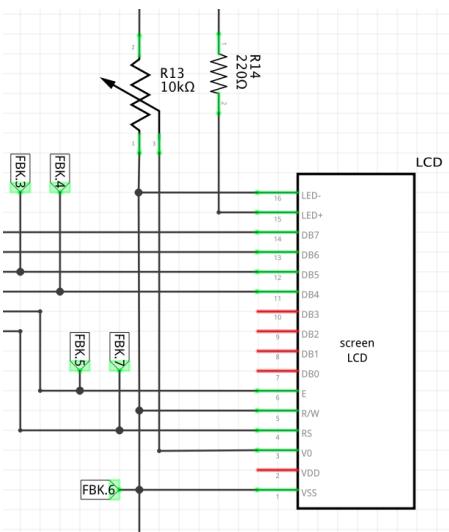


Abbildung 30: Schematische Darstellung der LCD-Beschaltung

Die Schaltung des Joysticks über einen analogen Input wurde im Kapitel bereits näher erläutert, die Funktion bleibt gleich, nur der Arduino hat sich geändert, weshalb wir hier nicht nochmals darauf eingehen.

Zu guter Letzt sieht man hier noch die Beschaltung des Pin-Headers und der Leiterplattenklemme. Unser selbst konzipiertes Board wird über eine Stifteleiste vertikal auf den Arduino gesteckt. Dies sieht cool aus und ist praktisch für Prototypen, da wir das Kabelgewirr

reduzieren können. Das Gerät wird wegen den Motoren schon genug an Kabel brauchen, weshalb diese Lösung praktisch schien.

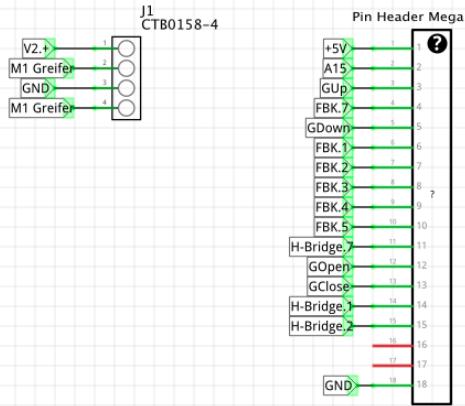


Abbildung 31: Schematische Darstellung des Pin-Headers / Leiterplattenklemme

6.9.5 Bestückung

Nun kam die Zeit des Bestückens. Da wir eine schlechte Ausrüstung hatten, erwies sich dieses Vorhaben als kleiner Krampf. Wir haben die Bestückung ausserdem auf einem Veroboard vorgenommen (Veroboard Farnell Datasheet). Dies eignet sich perfekt, um Prototypen herzustellen, ist aber durch seine Konzeption schwieriger zu löten.

Bei einer richtigen Produktentwicklung würde man ein Layout zeichnen. Das Schema würde dabei übersetzt werden in ein PCB, welches man dann ätzen lassen kann. Die Bauteile hätten dann ihren festen Platz, und das fertige Board könnte ganz einfach bestückt werden. Auf der unteren Abbildung ist dies gut ersichtlich. Den Zeitrahmen der IDPA würde dies sprengen, weshalb ein Veroboard die ideale Lösung war.

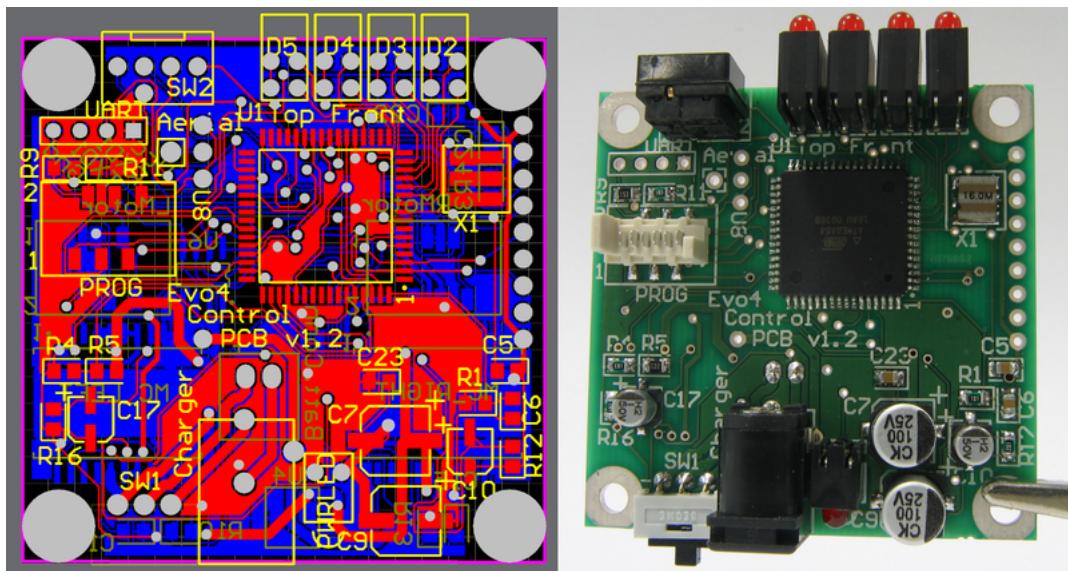


Abbildung 32: Vom Layout zur Printplatte (PCB Design, 2007)

Die Planung für die Bestückung eines Veroboard ist schwierig umzusetzen mit einem Programm, weshalb wir diesen Schritt einfach übersprungen haben. Es ist nicht ideal, denn die fehlende Planung macht sich spätestens bei der Fehlersuche bemerkbar. Es ist sehr unwahrscheinlich, dass man alle Verbindungen ohne Planung richtig zieht, weshalb man dann wirklich lange mit der Fehlersuche beschäftigt ist.

Wir haben das Schema also Stück um Stück auf das Veroboard übersetzt und jeweils direkt Messungen durchgeführt, um Fehlerquellen von Beginn an wenigstens einzuschränken.

Angefangen haben wir mit einem LCD-Board. Da der LCD am Gehäuse befestigt ist, mussten wir uns überlegen, wie wir die vielen LCD-Datenleitungen und auch die Stromversorgung übermitteln. Wir haben dafür ein Flachbandkabel benutzt, das vom LCD-Board zum Arduino geführt wird. Auf dem LCD-Board selbst findet sich ein Potentiometer, um den Kontrast des Bildschirms einzustellen. Außerdem werden die entsprechenden Leitungen zu den Pins des Flachbandkabels gezogen. Der LCD selbst kann einfach auf die schwarzen Buchsen gesteckt werden.

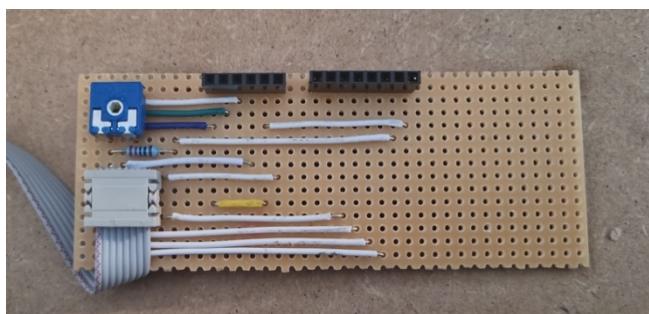


Abbildung 33: Fertig bestücktes LCD-Board

Beim ersten Betreiben dieses Boards ist auch direkt das Potentiometer in Rauch aufgegangen, wir hatten einen Kurzschluss. Nachdem wir aber den Fehler ausfindig machen konnten, und natürlich das Potentiometer ersetzt hatten, lief der LCD einwandfrei.

Schliesslich kam es zum Mainboard, also jenes, dass effektiv auf den Arduino gesteckt wird. Hier haben wir den Rest der Schaltung verbaut. Es hat einen Sockel für das Flachbandkabel vom LCD-Board, vier Stiftleistenpaare für unsere Buttons, eine grosse Buchse für unseren Joystick und schliesslich noch eine Leiterplattenklemme. In der grünen Leiterplattenklemme hat es vier Anschlüsse, zwei für die Stromversorgung vom Step-Down Converter und zwei für den Motor des Greifers. Die H-Brücke sowie der 6 V Regler sind ebenfalls auf dem Board verbaut. Alle Daten werden auf die Stiftleiste unten am Board gezogen, welche wiederum direkt auf den Arduino gesteckt ist. Dort kann dann die Datenverarbeitung über das Programm erfolgen.

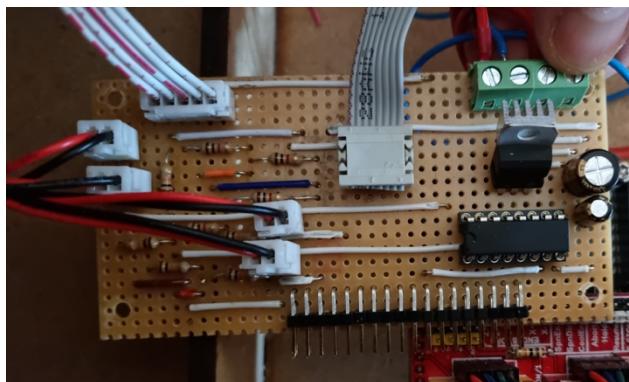


Abbildung 34: Fertig bestücktes Mainboard

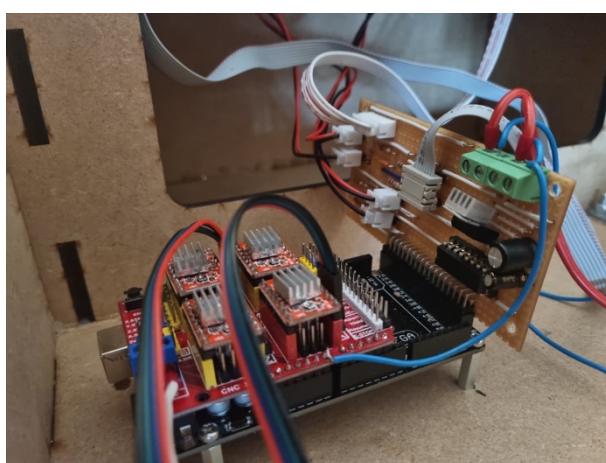


Abbildung 35: CNC-Shield und Mainboard, Steckverbindung zum Arduino Mega

6.10 Programmierung

Die Programmierung war ein laufender Prozess. Wir hatten bereits in der Experimentierphase einige Programme geschrieben und diese stetig verbessert. Wir hatten somit bereits funktionierende Codeschnipsel, bspw. für den LCD, der Ansteuerung der H-Brücke bzw. des Greifmotors, der Ansteuerung der Stepper-Motoren für die X-, Y- und Z-Achse und auch für die analoge Datenverarbeitung des Joysticks. Im letzten Schritt ging es also darum, all diese kleinen Programme in ein grosses zu verschmelzen. Durch die Vorarbeit mussten wir nicht bei null anfangen. Im Folgenden gehen wir auf das fertige Programm etwas näher ein. Das vollständige Programm ist wie auch das Schema im Anhang zu finden.

In den ersten 60 Zeilen wird einiges initialisiert und definiert, es ist aber sehr repetitiv, weswegen wir diesen Teil im Anhang lassen. Wir weisen dort lediglich unseren LCD, die Motoren, die H-Brücke, unsere Buttons und unseren Joystick spezifischen Arduino Pins zu. Außerdem definieren wir eine Reihe von Hilfsvariablen, die wir später brauchen werden.

Weiter unten im Programm haben wir das Setup, welchem wir uns hier zuerst widmen. Das Setup wird ausgeführt, sobald der Arduino gestartet wird. Zuvor definierte Pins setzt man hier als Output oder Input. Beispielsweise sind die Buttons Inputs, da wir Signale lesen müssen. Für die Motoren nutzen wir die Pins als Outputs, weil wir dort ein Signal ausgeben müssen.

Den LCD setzen wir hier auf einen Anfangswert, sodass der LCD bei Start der Maschine den richtigen Text anzeigt.

```
void setup() {
    pinMode(closeButton, INPUT);
    pinMode(openButton, INPUT);
    pinMode(upButton, INPUT);
    pinMode(downButton, INPUT);

    pinMode(hBridgePinOne, OUTPUT);
    pinMode(hBridgePinTwo, OUTPUT);
    pinMode(hBridgeEnable, OUTPUT);

    pinMode(StepX, OUTPUT);
    pinMode(DirX, OUTPUT);
    pinMode(StepY, OUTPUT);
    pinMode(DirY, OUTPUT);
    pinMode(DirZ, OUTPUT);
```

```

pinMode(StepZ, OUTPUT);
pinMode(enDRV, OUTPUT);

digitalWrite(hBridgeEnable, LOW);
digitalWrite(enDRV, HIGH);

lcd.begin(16, 2);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Druecke alle Ta-");
lcd.setCursor(0, 1);
lcd.print("sten fuer start");
Serial.begin(9600);
}

```

Abbildung 36: Setup-Schleife im Code

Direkt nach dem Setup befindet sich unser Main-Loop, dies ist im Grunde das Hauptprogramm. Der Arduino arbeitete diese Zeilen von oben nach unten durch, sobald er unten angekommen ist, beginnt er wieder von oben. Wir prüfen in der ersten If-Schlaufe ob alle vier Buttons gleichzeitig gedrückt wurden. Ist dies der Fall, beginnt das Spiel.

Die zweite If-Schlaufe prüft, ob das Spiel begonnen hat. Falls ja, werden die entsprechenden Unterprogramme wie lcdControl(), joystickControl, clawControl und zAxis gestartet.

```

// main loop

void loop() {

    // check for game start
    if (debounce(openButton) == 1 && debounce(closeButton) == 1 && debounce(upButton)
    == 1 && debounce(downButton) == 1 && start == false) {
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("restl. Zeit:");

        delay(500);
        start = true;
        startTime = millis() / 1000;
    }

    // start control-programs when game started
    if (start == true) {
        lcdControl();
        joystickControl();
        clawControl();
        zAxis();
    }
    else {
        hBridgeMotor(LOW, LOW, LOW);
    }
}

```

Abbildung 37: Main-Loop, Hauptprogramm des Arduinos

Wir kommen nun zu den verschiedenen Unterprogrammen, welche vor dem Setup definiert wurden. Das erste Unterprogramm Debounce ist dazu da, unsere Buttons zu entprellen. Mechanische Schalter neigen beim Ein- und Ausschalten zum sogenannten Prellen, das heisst, sie schalten sich mehrfach aus und ein (mechanische Vibration). Dieser Effekt tritt nur ein Bruchteil von einer Sekunde auf, nämlich genau beim Drücken und Loslassen der Taste. Es

kann zu Störsignalen führen, weshalb wir diese im Programm filtern. Jedes Mal, wenn wir einen Schalter brauchen, nutzen wir die debounce()-Funktion, um die Entprellung zu aktivieren. Prellt der Schalter zur Zeit der Betätigung, wartet das Programm ab, bis das Signal eindeutig ist. Wie bereits erwähnt geschieht das Ganze in unter einer Sekunde, das Abwarten auf ein eindeutiges Signal ist für den Spieler also nicht bemerkbar.

```
/*
*****debounce*****
/** debounce buttons ***
/*****debounce*****  

int debounce(int button) {
    buttonState = digitalRead(button);

    if (buttonState != previousState) {
        previousState = buttonState;
        if (buttonState == 1) {  

            return true;  

        }  

        else {  

            return false;  

        }
    }
}
```



Abbildung 38: Schalterentprellung mit Veranschaulichung (Artikel zur Entprellung, 2017)

Beim Aufruf des Unterprogramm hBridgeMotor() wird lediglich die H-Brücke angesteuert, so dass sich der Greifer entweder öffnet oder schliesst, je nachdem, welche Daten der Funktion übergeben wurden.

```
/*
*****hBridgeMotor*****
/** Start H-Bridge and set direction for claw ***
/*****hBridgeMotor*****  

int hBridgeMotor(int hBridgeEnabled, int varhBridgePinOne, int varhBridgePinTwo) {  

    digitalWrite(hBridgeEnable, hBridgeEnabled);  

    digitalWrite(hBridgePinOne, varhBridgePinOne);  

    digitalWrite(hBridgePinTwo, varhBridgePinTwo);  

}
```

Abbildung 39: Start der H-Brücke im Code

Das vorherige Unterprogramm hBridgeMotor() nutzen wir beispielsweise in unserer Funktion clawControl(). Diese Funktion wird im Main-Loop, den wir bereits beschrieben haben, ausgeführt. ClawControl() liest die Buttons für die Greifer ein. Wird ein Button betätigt, so werden die entsprechenden Variablen gesetzt. Diese wiederum werden am Schluss der Schlaufe an hBridgeMotor() geschickt, der den Greifer in die entsprechende Richtung bewegt, ihn also öffnet oder schliesst.

```
/*
***** H-Bridge Control, read digital Pins for up / down / open / close ****/
***** H-Bridge Control, read digital Pins for up / down / open / close ****/

int clawControl() {

    // set variables to open claw with H-Bridge
    if (debounce(openButton)) {
        hBridgeEnabled = HIGH;
        varhBridgePinOne = HIGH;
        varhBridgePinTwo = LOW;
    }

    // set variables to close claw with H-Bridge
    if (debounce(closeButton)) {
        hBridgeEnabled = HIGH;
        varhBridgePinOne = LOW;
        varhBridgePinTwo = HIGH;
    }

    // start motor only when different states of buttons true
    if (debounce(openButton) != debounce(closeButton)) {
        hBridgeMotor(hBridgeEnabled, varhBridgePinOne, varhBridgePinTwo);
    }
    else {
        hBridgeMotor(LOW, LOW, LOW);
    }
}
```

Abbildung 40: Einlesen der Buttons

Mit `zAxis()` können wir den Greifer bzw. die Z-Achse bewegen. Gleich wie beim Öffnen und Schliessen des Greifers prüfen wir, ob ein Button betätigt wurde. Falls ja, wird das Programm `stepperControl()` gestartet, welches den entsprechenden Stepper-Motor (hier die Z-Achse) rotieren lässt.

```
int zAxis() {
    boolean motorDirection;
    int button;

    // start motor for claw down (z-axis)
    if (debounce(downButton)) {
        motorDirection = false;
        button = downButton;
        // stepperControl(false, DirZ, StepZ, downButton); //Z, Clockwise
    }

    // start motor for claw up (z-axis)
    if (debounce(upButton)) {
        motorDirection = true;
        button = upButton;
        // stepperControl(true, DirZ, StepZ, upButton); //X, Counterclockwise
    }

    if (debounce(upButton) != debounce(downButton)) {

        stepperControl(motorDirection, DirZ, StepZ, button); //X, Counterclockwise
    }
}
```

Abbildung 41: Steuerung der Z-Achse

Wie bereits erwähnt startet die Funktion `stepperControl()` die Stepper-Motoren. Stepper-Motoren sind etwas speziell in der Ansteuerung. Statt einfach den Strom einzuschalten und der Motor startet, muss man ihm sagen, wie viele Steps, also Umdrehungen, er machen soll. In unserem Fall soll er sich drehen, solange der Joystick oder aber ein Button der Z-Achse betätigt wird.

```
/*
 **** start Stepper Motor with arguments ***
 ****
 int stepperControl(boolean motorDirection, byte directionPin, byte stepperPin, int button) {
    int statement = 1;
    int n = 0;
    // set direction
    digitalWrite(directionPin, motorDirection);

    // set stepper and start as long as statement and start is true
    for (int i = 0; statement && start; i++) {

        digitalWrite(stepperPin, HIGH);
        delayMicroseconds(delayTime);
        digitalWrite(stepperPin, LOW);
        delayMicroseconds(delayTime);

        lcdControl();

        // check statement for button or joystick
        if (button == A15) {
            statement = analogRead(button);
        }
        else {
            statement = digitalRead(button);
        }
        delay(1);

    }
}
```

Abbildung 42: Steuerung der Stepper-Motoren

Die Stepper-Motoren werden auch im Programm joystickControl() gebraucht. Dieses Programm liest den Input des Joysticks und je nachdem wie der steht, wird ein anderer Motor über stepperControl() angesteuert.

```
/*
 **** Control Joystick, map analog input for stepperControl ***
 ****
 int joystickControl() {

    int sensorValue = analogRead(joystickPin);
    delay(1);

    // map analog input into switch case statement for every direction (Y/X), start stepper with arguments
    if (sensorValue > 100) {
        int joystickDirection = map(sensorValue, 0, 1023, 0, 3);
        switch (joystickDirection) {
            case 0: stepperControl(false, DirY, StepY, joystickPin); delay(100); break; // Clockwise
            case 1: stepperControl(true, DirY, StepY, joystickPin); delay(100); break; //

            -----
            COUNTERCLOCKWISE
            case 2: stepperControl(false, DirX, StepX, joystickPin); delay(100); break;
            case 3: stepperControl(true, DirX, StepX, joystickPin); delay(100); break;
        }
    }
}
```

Abbildung 43: Einlesen des Joysticks

Im letzten Programm lcdControl() (siehe Anhang) wird der LCD angesteuert. Wir definieren den anzugegenden Text und lassen einen Timer laufen. Ist der Timer bei null, ist das Spiel vorbei und der Text auf dem LCD wird entsprechend angepasst. Außerdem werden alle Motoren ausgeschaltet, man kann also nicht mehr weiterspielen, wenn die Zeit vorbei ist.

6.11 Verdrahtung

Im letzten Schritt mussten wir die ganze Elektronik im Gehäuse verdrahten sowie verschiene Kabel zu den Motoren ziehen. Wir hatten bei der Verdrahtung ausserdem tatkräftige Unterstützung von Christian (Christophs Bruder), der mit seinem breiten Wissen im Bereich der Automations- und Verdrahtungstechnik einen grossen Beitrag zum Endprodukt leisten konnten. An dieser Stelle möchten wir uns für seine Zeit wie auch für Werkzeuge, Bauteile und Litzen, die er uns zur Verfügung gestellt hat, bedanken.

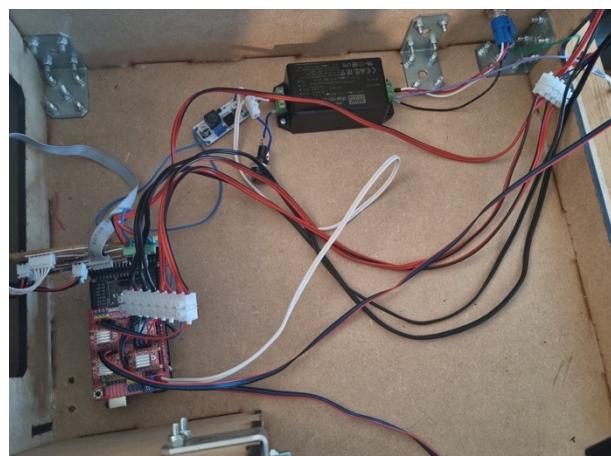
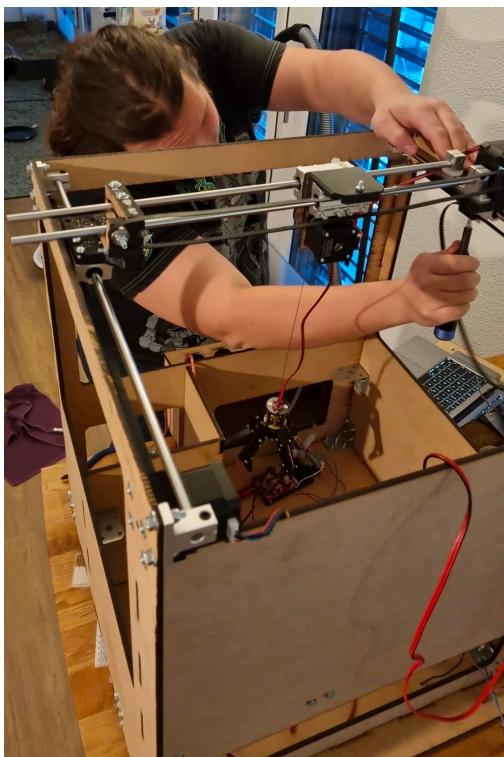


Abbildung 44: Christian bei der Verdrahtung (links) und Verdrahtung im Gerät (rechts)

6.12 Messungen

Eine der ersten Messungen, die wir machten, hat sich auf die Leistungsaufnahme der Stepper-Motoren und des Arduino bezogen. In einem einfachen Aufbau haben wir die vier Stepper-Motoren, den Greifer und den Arduino zusammen gestartet und diese parallel betrieben. Aus diesen Angaben konnten wir danach ein passendes Netzteil mit genug Leistung dimensionieren, in diesem Fall also mindestens ein 30 Watt Netzteil. In unserer Maschine werden nie alle Motoren gleichzeitig angesteuert. Man kann mit dem Joystick nicht beide Achsen gleichzeitig bewegen, weshalb nie so viel Leistung verbraucht wird, wie hier gemessen. Nichtsdestotrotz wollten wir uns für die Spitzenleistung der Motoren absichern, auch wenn dieser Fall in der Praxis nicht eintritt.



Abbildung 45: Leistungsaufnahme, Spitzenwert

Als wir die ganze Schaltung fertig verbaut hatten, haben wir noch die effektive Stromaufnahme der Schaltung gemessen. Während dem Messen haben wir eine Achse sowie den Greifer bewegt, also die maximale Anzahl an Motoren, die man gleichzeitig bewegen kann. Dabei hat das 24 V Netzteil einen Strom von 0.51 Ampere abgegeben.

Aus diesen Angaben ergibt sich folgende Leistung:

$$U_{\text{out}} = 24 \text{ V}, I_{\text{total}} = 0.51 \text{ A}$$

$$P_{\text{total}} = U_{\text{out}} * I_{\text{total}} = 24 \text{ V} * 0.51 \text{ A} = 12.24 \text{ Watt}$$

Unser Netzteil wie auch unsere Sicherung sind somit grosszügig dimensioniert. Dies ist nicht ideal in Sachen Effizienz, stellt aber kein Problem dar, ausserdem ist man so auch auf der sicheren Seite.

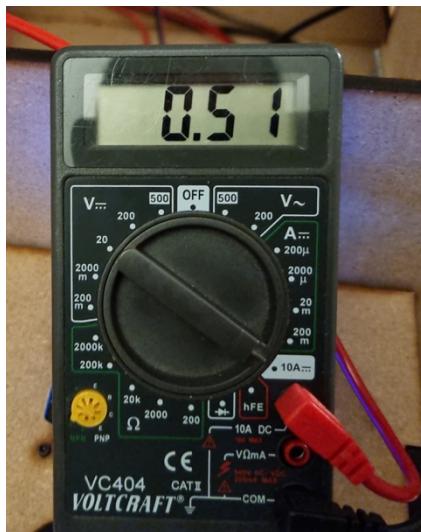


Abbildung 46: Effektive Stromaufnahme

Den Step-Down Converter haben wir als fertiges Modul mit einem Potentiometer bestellt. Über das Potentiometer konnten wir unsere gewünschte Ausgangsspannung justieren, in unserem Fall 7 V für den Arduino.



Abbildung 47: Einstellen der Ausgangsspannung, Step-Down Converter

Um die Stromaufnahme des Greifers abzuschätzen, haben wir diese ebenfalls gemessen. Da wir den Greifer zusammen mit dem Motor aus China bestellt hatten, gab es keine Angaben zu diesem spezifischen Motor. Wir haben zwar ähnliche Bauteile gefunden, ein paar Messungen würden aber dennoch nicht schaden.

Der Greifer hat bei Normalbetrieb ca. 40 mA an Strom aufgenommen, wobei er aber an den Endpositionen bis zum 8-fachen verbrauchte. Das heisst, ist der Greifer bereits am Anschlag, man drückt aber weiterhin den Button, so zieht er massiv viel Strom. Dies liegt daran, dass der Motor den Greifer mit voller Kraft versucht zu öffnen oder zu schliessen, der mechanische Widerstand ist aber zu gross dafür.

In einem richtigen Produkt müsste dieses Problem wahrscheinlich hardwaremässig angegangen werden. Sobald der Motor am Anschlag ist, darf er entweder nicht zu viel Strom ziehen, oder aber wird nach einer gewissen Zeit über einen Widerstand geschalten. Für unsere IDPA hätte es sich nicht gelohnt, dieses Problem anzugehen, da der Nutzen gering, der Aufwand aber gross ist. Es bedeutet lediglich, dass man den Greifer nicht dauerbelasten sollte. Würde man den Greifer beispielsweise schliessen, und den Button weitere 10 Minuten gedrückt halten, obwohl der Greifer bereits geschlossen ist, könnte es sein, dass der Motor ab einem gewissen Punkt durchbrennt. Da eine Runde Programmatisch nur 100 Sekunden dauert, wäre dies nicht mal möglich. Man müsste das Spiel also immer wieder neustarten,

um den Motor gezielt zu beschädigen, was in der Praxis höchstwahrscheinlich nicht passieren wird.

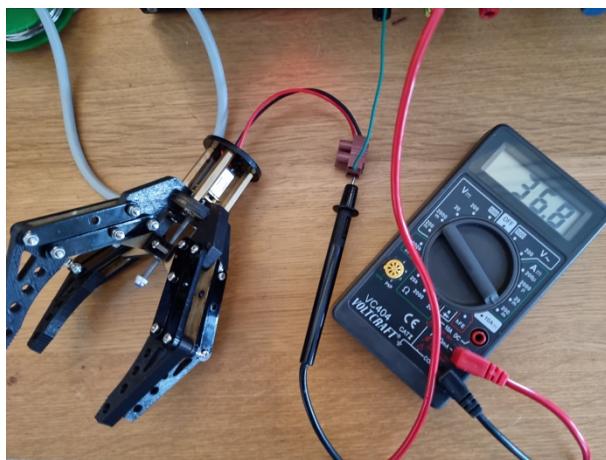


Abbildung 48: Stromaufnahme Greifer im Normalbetrieb

Diese Messung bezieht sich auf einen Fehler, der viel Zeit gekostet hat und welchen wir bis jetzt nicht erklären können. Wir nennen diesen Fehler deshalb auch «die unerklärliche 1». Dieser wird in der Fehlerbehebung näher beschrieben. Die 1 bezieht sich auf das Signal, dass wir messen, also 5 V wie im Bild ersichtlich.

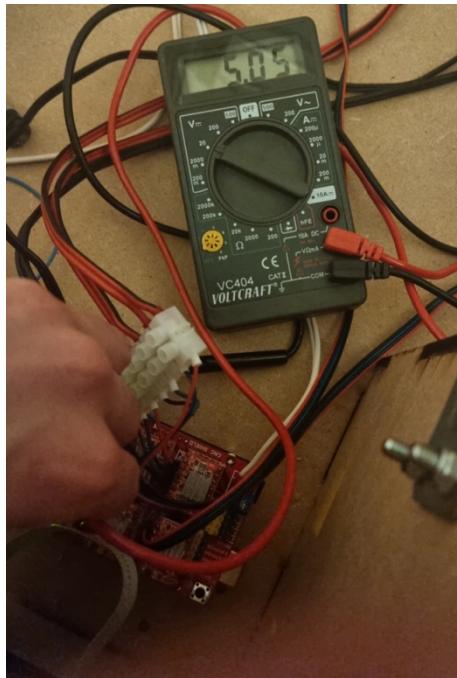


Abbildung 49: Messung einer unerklärlichen 1 (5 V)

6.13 Fehlerbehebung

Während der elektronischen Umsetzung traten eine Reihe von Fehler auf, denen wir alle-samt mit messen und genauem prüfen des Schemas, des Codes sowie der Bestückung auf den Grund gehen konnten. Die wesentlichen Probleme waren unschöne Lötstellen, fehlende Leiterplattenunterbrechungen, Kurzschlüsse und die unerklärliche 1.

Unschöne Lötstellen waren insofern ein Problem, da sie Kurzschlüsse versucht hatten. Diese auf dem Veroboard ausfindig zu machen und zu beseitigen hat einiges an Zeit gekostet. Oft-mals sind Kurzschlüsse nicht direkt ersichtlich, weshalb wir die Stellen auch mehrmals mit dem Multimeter prüfen mussten.

Da wir kein Layout geplant hatten, mussten wir im Nachhinein ein paar Verbindungen su-chen, die wir auf dem Veroboard vergessen hatten, zu durchtrennen. Wir haben Verbindun-gen auf dem Veroboard entweder mit dem Cutter, oder aber mit der Bohrmaschine durch-trennt. Ohne Planung war es auch schwierig, diese zu überprüfen, weshalb wir Stelle um Stelle anschauen und messen mussten.

Der letzte Fehler, dem wir einiges an Zeit widmen mussten, war die unerklärliche 1. Wir ha-ben auf dem einen Pin des Arduinos eine 1 gemessen, also 5 V, die da nicht sein sollten. Dies war das Eingangssignal für den Greifer-Motor auf der Z-Achse. Der Fehler führte dazu, dass der Greifer sofort endlos nach oben und unten fuhr, sobald man das Spiel startete. Ko-mischerweise trat dieser Fehler nur auf, wenn das Programm des LCDs aktiv war, nicht aber, wenn die Funktion des LCDs im Code auskommentiert wurde. Dies ist auch der Grund, wa-rum wir zuerst Annahmen, es handle sich um einen Fehler in unserem Code. Nachdem wir aber jede Funktion nochmals gründlich überprüft hatten, haben wir den Schluss gezogen, dass es ein Hardware-Fehler sein muss.

Also prüften wir unser Board wieder gründlich, und wieder wurden wir nicht fündig. Es schien, als würden wir den Fehler nicht finden.

Bei weiteren Messungen fiel uns schlussendlich auf, dass der Arduino selbst am Pin 25 die 1, also 5 V liefert. Es hat also gar nicht direkt mit unserer Schaltung zu tun, es kommt vom Arduino. Nichtsdestotrotz passiert es aber auch nur, wenn wir unser Board einstecken. Wir sind dem ganzen ratlos gegenüber, konnten es aber lösen, indem wir den Pin 25 einfach ab-geschnitten haben. Unsere Stiftleiste am Board hat zwei Reihen, also für jede Verbindung haben wir zwei Pins am Arduino zur Verfügung. Wir brauchen lediglich einen, um die Z-Achse des Greifers zu bewegen (Pin 24), weshalb wir den Pin 25 problemlos abschneiden

konnten. So konnte das Problem behoben werden, obwohl wir immer noch nicht genau wissen, warum.

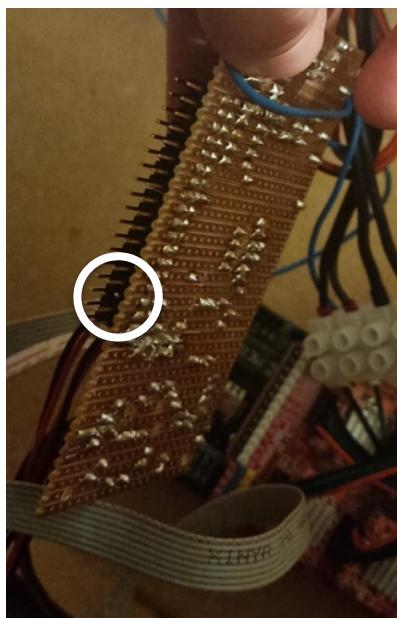


Abbildung 50: Abgeschnittener Pin 25

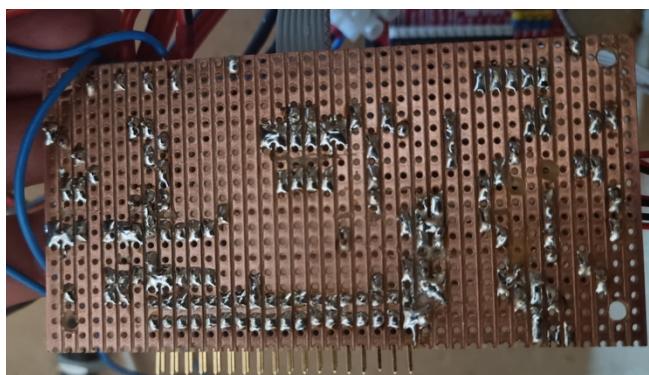


Abbildung 51: Mainboard von hinten

6.14 Namensfindung «Clawminator»

Wie jedes gute Produkt auch, brauchte unser Greifautomaten einen treffenden Namen. Dafür haben wir uns gefragt, wie können wir unsere Zielpersonen damit erreichen. Nach langen Überlegungen kam uns plötzlich den Namen «Clawminator» in den Sinn. Ein Wortspiel mit den Wörtern «Claw», das englische Wort für Klaue und «Terminator», weil unser Produkt eine Maschine ist, genau wie «Terminator».

6.15 Sticker

Damit wir unseren Greifautomaten noch den letzten Feinschliff geben können und ihn Optisch ansprechend gestalten können, dekorieren wir den Automaten mit Stickern. Die Sticker haben wir für den Namen «Clawminator» und die Beschriftung der Buttons grafisch realisiert.

Die Umsetzung haben wir in Adobe Illustrator 2021 realisiert. Das Programm dient zur Herstellung von Grafiken und Illustrationen. Die Beschriftung der Buttons für das Rauf- und Runterlassen des Greifers haben wir mit weissen, dreieckigen Pfeilen erstellt. Für das Öffnen und Schliessen des Greifers entschieden wir uns für Sticker mit dem Schriftzug «open» und «close».

Als die Sticker fertig designt waren, haben wir die Designs druckfertig als PDF exportiert. Für die Herstellung durften wir im Bühler einen speziell gebauten Drucker für Sticker benutzen. Zu Schluss mussten wir die PDF-Datei an den Drucker senden und schon hatten wir die finalen Sticker in unseren Händen, welche wir nur noch bekleben mussten.

6.16 Gewinn

Damit die Spieler einen großartigen Preis mit nach Hause nehmen können, fehlte nur noch die Auswahl des Gewinnes, welchen wir in unseren Greifautomaten platzierten. Von der Grösse her mussten wir uns für Objekte entschieden, die vom Durchmesser maximal 5 Zentimeter gross sind. Dafür eignet sich perfekt kleine Süßigkeiten wie beispielsweise Gummibärchen-Päckchen. Mit verschiedenen Süßigkeiten wollen wir unseren Spielern ein kleines Lächeln ins Gesicht zaubern.

Da es schon bald Weinachten war, spielten wir noch mit dem Gedanken, Mandarinen und Nüsse als Gewinn zu nehmen. Den Gedanken haben wir jedoch rasch wieder verworfen, da die Präsentation erst nach den Weihnachtsferien stattfinden wird.

7 Kontrollieren

Bei der Kontrolle geht es in erster Linie darum die Kriterien, welche man in der Anforderungsliste und im Pflichtenheft fixiert hat, zu überprüfen. Hierbei muss darauf geachtet werden, ob alle Forderungen, sowie die Wünsche mit hoher Priorität erfüllt sind. Wenn das nicht zutrifft, muss man das Produkt nochmals überarbeiten und die nötigen Änderungen vornehmen. Die Kontrolle ist einer der wichtigsten Schritte bei einer solchen Projektarbeit, den jeder weiß, oberste Priorität hat die Funktion und die Zufriedenheit des Kunden.

7.1 Anforderungsliste überprüfen

Wir haben die Anforderungsliste überprüft. Fast alle Anforderungen und Wünsche wurden erfüllt. Der Wunsch Musik, während dem Spiel laufen zu lassen, sowie der Münzeinwurf wurde nicht erfüllt. Diese Punkte sind jedoch nur schwache Wünsche und darum nicht schwerwiegend. Die Kosten haben wir als Forderung auf 400.- CHF beschränkt. Diese konnten wir leider nicht einhalten. Wir sind selbst überrascht, wie schnell sich kleine Beträge summieren. Schlussendlich haben wir für den ganzen Greifautomaten 500.- CHF ausgegeben. Um unter den geforderten Höchstbetrag von 400.- CHF zu kommen, müsste man günstigere Komponenten benutzen. Da wir jedoch von der Zeit her kaum Spielraum hatten, mussten wir teils auf teurere Produkte zugreifen mit diesem Resultat zufriedengeben. Auch die Dimensionen konnten wir nicht einhalten. Wir haben uns im Laufe des Projekts entschieden, den Greifautomaten ein wenig grösser zu gestalten, da der Spielraum im Verhältnis zum Greifer zu klein gewesen wäre.

ANFORDERUNGSLISTE

Änderungsdatum:	Anforderung	F = Forderung W = Wunsch S = Sparer 1 = schwierig	Bemerkungen	Verantwortlich
	Funktions- und Leistungsanforderungen			
	<i>Greifer ist Horizontal steuerbar</i>		erfüllt	
	<i>Greifer kann vertikal heruntergelassen werden</i>		erfüllt	
	<i>Greifer kann schliessen</i>		erfüllt	
	<i>Fallengelassenes Produkt ist von aussen greifbar</i>		erfüllt	
	<i>Musik beim spielen</i>	2	nicht erfüllt	
	<i>Display</i>	4	erfüllt	
	<i>Steuerung voll elektronisch</i>	5	erfüllt	
	<i>Münzeinwurf</i>	1	nicht erfüllt	
	Dimensionen			
	<i>max. 600 mm x 600 mm x 600 mm</i>		nicht erfüllt	
	<i>min. 300 mm x 300 mm x 300 mm</i>		erfüllt	
	Qualitätsmerkmale			
	<i>System läuft zuverlässig (keine Bugs)</i>		erfüllt	
	<i>Endprodukt fühlt sich hochwertig an</i>		erfüllt (kleine Mängel)	
	Marktanforderungen			
	<i>wenig/ keine Wartung nötig</i>	5	erfüllt	
	<i>ansprechendes Design</i>	4	erfüllt	
	Fertigungskriterien			
	<i>Rahmen grösstenteils mit 3D-Drucker oder Holzlaser herstellbar</i>		erfüllt	
	<i>Stückzahl 1</i>		erfüllt	
	<i>Fertigung und Montage Zuhause, Schule oder im Makerspace möglich</i>		erfüllt	
	Kosten			
	<i>ca. CHF 200.- (maximalkosten CHF 400.-)</i>		nicht erfüllt (Gesamtkosten betragen ca. 500.- CHF)	

Kostenschätzung

Was	Wer / Wo	Kosten
<i>Rohmaterial</i>		CHF 50
<i>Einkaufskomponenten</i>		CHF 300
Gesamtkostenschätzung	ca. (+/- 10%)	CHF 350

Abbildung 52: Ausgewertete Anforderungsliste

7.2 Sicherheit überprüfen

Es hat keine zugängliche gefährliche Rotierende Teile. Die Elektronik ist auch nicht zugänglich. Somit ist eine allgemeine Sicherheit gewährleistet.

7.3 Spielbarkeit überprüfen

Wir konnten den Greifautomaten noch vor den Weihnachtsferien in die Schule mitbringen und verschiedene Lernende und Lehrer damit spielen lassen. Das Spiel hat ohne Bugs funktioniert und die Spieler hatten stets Freude am Spiel. Leichte Dinge, wie Plüschtiere kann der Greifer ohne Probleme aufnehmen. Bei schwereren Gegenständen kommt er aber an seine

Grenzen. Aber nach all unseren Versuchen können wir den Automaten ohne Bedenken, wie geplant, mit Süßigkeiten auffüllen. Die Gewinnchancen sind sehr hoch. So dass ein Gewinn in der vorgegebenen Zeit fast garantiert ist. Auch weniger geschickte Spieler konnten mit ein wenig Hilfe einen Gewinn erzielen. Wir sind selbst mit dem Ergebnis sehr zufrieden, die ganze Arbeit war ein totaler Erfolg. Die vielen guten Rückmeldungen unterstreichen unsere Meinung.

8 Auswerten

Der Punkt, der am meisten vergessen wird. Die Erkenntnisse, die hier gemacht worden sind, aber eines der Wichtigsten Dinge, die aus solch einen Projekt entstehen.

8.1 mögliche Punkte in einer Auswertung

Eine Auswertung der Arbeit kann man verschieden gestalten. Je nachdem, welche Punkte einem wichtig sind. Es könnte eine Reflektion, Auswertung der Lösung, Zusammenfassung des Entwicklungsprozesses oder ein Ausblick sein.

Wir haben uns für eine Auswertung der Lösungsvariante und einen Ausblick entschieden.

8.2 Auswertung

Unsere Lösung ist gut gelungen. Es ist bei weitem nicht perfekt. Aber zu jeder mangelhaften Stelle kann eine Geschichte erzählt werden und eine Lehre daraus gezogen werden. Es zeigt auf, welche Schwierigkeiten einem in der Entwicklung begegnen und wie man darauf reagieren kann. Der Greifautomat ist jetzt im Stand eines gut ausgearbeiteten ersten Prototyps. Aus den gesammelten Erfahrungen könnte man jetzt ein zweiter Prototyp herstellen.

8.3 Verbesserungspunkte für Version 2.0

In einer verbesserten Version würden wir sicherlich noch Plexiglasscheiben einbauen. Bis jetzt wollten, wir noch keine einbauen, damit der Zugriff zur Elektronik gewährleistet ist. Außerdem würden wir die Wände nochmals sauber lasern und die Halterung für die Schiene nochmal überarbeiten. Die Elektronik müsste verbessert und effizienter umgesetzt werden, die Auswahl der Bauteile müsste angepasst werden. Außerdem könnte man ein PCB erstellen.

8.4 Ausblick

Wir werden keine Version 2.0 herstellen. Dafür werden wir unsere erste Version noch etwas besser ausbauen. Die kleinen Macken, die im Moment noch vorhanden sind, werden wir ausbessern. Z.B. Unterlagscheiben als Führung für den Zahniemen anbringen.

9 Probleme / Schwierigkeiten

Wie auch in anderen Projekten gab es auch bei uns in der Umsetzung Schwierigkeiten, diese haben ziemlich viel Zeit in Anspruch genommen. Unsere Planung war so weit in Ordnung, jedoch gibt es einfach immer wieder Probleme, die man nicht einkalkulieren kann. Einiges konnten wir auch nicht umsetzen wie geplant aufgrund von kleineren Fehlern, welche erst bei der Realisierung zur Geltung kamen.

9.1 Zeitlicher Aspekt

Dieser Punkt hat uns am meisten Schwierigkeiten bereitet, denn unser gewähltes Thema ist ziemlich zeitaufwändig. Wir wussten von Beginn an, dass unsere Eigenarbeit sehr umfangreich und herausfordernd ist. Doch wir haben die Herausforderung angenommen, weil uns diese Arbeit fasziniert hat und wir auch etwas wollten, auf dass wir dann später stolz sein können.

Meist haben wir zu wenig Zeit eingeplant und so gerieten wir in Zeitdruck. Dies hatte auch zur Folge, dass wir für die schriftliche Arbeit weniger Zeit hatten.

Was uns am meisten Zeit gekostet hat war:

- Die fehlerhafte Elektronik
- Die langandauernde Montage

9.2 Lieferengpässe / Bestellungen

Im Voraus wussten wir bereits, dass wir uns früh genug um Bestellungen kümmern müssten, denn zurzeit gibt es öfters Lieferengpässe aufgrund der Pandemie. Deshalb haben wir frühzeitig eine Einkaufsliste erstellt und uns auf die Suche nach den Materialien, Komponenten und Teilchen im Internet gemacht. Da wir spezifische Waren benötigten, mussten wir bei mehreren Händlern gleichzeitig bestellen. Trotz frühem Bestellen konnten wir Verzögerungen leider nicht verhindern. Gewisse Teile kamen erst Ende November und sogar im Dezember an. Aus diesen Gründen mussten wir manchmal die Bestellung abwarten.

9.3 Ortsabhängige Zusammenarbeit

Dieses Problem betrifft nur die Arbeit an unserem Greifautomaten, denn für die physische Arbeit am Greifautomaten mussten wir uns gemeinsam vor Ort treffen damit wir Fortschritte machen. Wir nutzten die Zeit in der Schule und gingen öfters in den Bühler, damit wir dort zusammenarbeiten konnten. Zudem trafen wir uns auch manchmal an einem Abend, unter

dem Wochenende oder am Samstagnachmittag, um zu arbeiten. Nichtsdestotrotz muss man das Ganze koordinieren und sich absprechen, zudem wohnen wir auch nicht alle in derselben Umgebung, was es noch schwieriger macht. Wir haben aber das Beste daraus gemacht.

Die Zusammenarbeit bezüglich der Planung sowie bei der Dokumentation verlief dafür reibungslos und ortsunabhängig.

10 Schlussbemerkungen

Die Arbeit war ein voller Erfolg. Der ganze Arbeitsprozess hat uns stets Freude bereitet, da wir uns alle für unser gewähltes Thema brennend interessierten. Allerdings haben wir nie damit gerechnet, dass der Greifautomat so viel Zeit in Anspruch nimmt. Für Kleinigkeiten wie beispielsweise das Montieren der einzelnen Teilchen brauchten wir länger, wie wir es in der Planung definiert haben. Doch diese Hürde konnten wir durch frühzeitiges und proaktives arbeiten überwältigen. Wir haben uns mehrmals in unserer Freizeit getroffen und konnten somit trotz allen Schwierigkeiten unsere Zeitmanagement ziemlich gut einhalten.

Nicola: Unsere grösste Stärke war unsere Zusammenarbeit, durch unseren ehrlichen und aktiven Austausch konnten wir eine harmonische Arbeitsstimmung kreieren. Durch diese Harmonie hat uns die Arbeit noch mehr Freude bereitet und wir kamen dem Ziel immer schneller näher. Auch wenn wir manchmal frustriert waren, haben wir uns nie aus der Ruhe bringen lassen und uns gegenseitig motiviert. Wir haben gemerkt, wie wertvoll es ist, ein gutes Arbeitsklima miteinander zu haben. Die Arbeit war schlussendlich ein totaler Erfolg und wir sind überglucklich damit!

Christoph: Ich denke eine Arbeit gerade in diesem Umfang ist immer zäh, gerade den zunehmenden Zeitdruck haben wir gegen Ende des Projekts zu spüren bekommen. Umso grösser ist aber dann die Freude, wenn man das Endprodukt bestaunen darf, und ich denke wir können voller Stolz sagen, dass wir diese Aufgabe ausserordentlich gut gelöst haben. Die Maschine sieht zumindest super aus und funktioniert. Ich würde daher sagen, wir haben unser Ziel erreicht, wenn nicht sogar übertroffen.

Elia: An erster Stelle möchte ich ein grosses Kompliment an meine Teamkameraden aussprechen, ich würde mit jedem von ihnen nochmals ein solches Projekt absolvieren. Unsere Zusammenarbeit verlief äusserst harmonisch und qualitativ hochstehend. Zudem bin ich extrem stolz auf das Endprodukt, ein voller Erfolg! Ich bin überglucklich und überzeugt von unserem Clawminator.

Lionel: Ich hatte stets Freude an der Zusammenarbeit und am gewählten Thema. Das funktionierende Endprodukt zu bestaunen, an dem wir so viele Stunden gearbeitet haben war für mich der Höhepunkt.

11 Abbildungsverzeichnis

Abbildung 1: Fertiger Greifautomat	3
Abbildung 2: Kostenschätzung aufgrund der Anforderungsliste	8
Abbildung 3: Anforderungsliste	9
Abbildung 4: Funktionsstruktur.....	12
Abbildung 5: Morphologischer Kasten.....	13
Abbildung 6: Vor- und Nachteilliste	14
Abbildung 7: Nutzwertanalyse.....	15
Abbildung 8: S-Diagramm	16
Abbildung 9: finales CAD-Modell.....	17
Abbildung 10: Kostenliste.....	19
Abbildung 11: Frontplattenfertigung	22
Abbildung 12: Elia beim Feilen.....	23
Abbildung 13: Erster Versuchsaufbau mit rotierendem Motor, MOS-FET Schaltung	24
Abbildung 14: Erster Schematischer Aufbau (Arduino SA, 2015, p. 96)	24
Abbildung 15: Zweiter Versuchsaufbau mit H-Brücke.....	25
Abbildung 16: Schema mit H-Brücke (Arduino Projects Book, 2015, p. 104)	25
Abbildung 17: Aufbau der Schaltung mit Kippschalter	26
Abbildung 18: Schema und Code-Logik Kippschalter	26
Abbildung 19: LCD Test mit H-Brücke	27
Abbildung 20: CNC Shield mit Stepper-Motoren und Greifer.....	28
Abbildung 21: Schematische Darstellung der analogen Joystickabfrage.....	29

Abbildung 22: Serieller Output beim Betätigen des Joysticks	29
Abbildung 23: Aufbau der Joystickabfrage	30
Abbildung 24: Joystickabfrage visualisiert mit LEDs	30
Abbildung 25: Spannungsregler mit Motortreiber	32
Abbildung 26: Schematische Darstellung der Spannungsversorgung (AC)	35
Abbildung 27: Schematische Darstellung der Spannungsversorgung (DC)	36
Abbildung 28: Schematische Darstellung der H-Brücke	36
Abbildung 29: Schematische Darstellung der Eingabe des Greifers	37
Abbildung 30: Schematische Darstellung der LCD-Beschaltung	37
Abbildung 31: Schematische Darstellung des Pin-Headers / Leiterplattenklemme	38
Abbildung 32: Vom Layout zur Printplatte (PCB Design, 2007)	39
Abbildung 33: Fertig bestücktes LCD-Board	40
Abbildung 34: Fertig bestücktes Mainboard	40
Abbildung 35: CNC-Shield und Mainboard, Steckverbindung zum Arduino Mega	40
Abbildung 36: Setup-Schleife im Code	42
Abbildung 37: Main-Loop, Hauptprogramm des Arduinos	42
Abbildung 38: Schalterentprellung mit Veranschaulichung (Artikel zur Entprellung, 2017) ...	43
Abbildung 39: Start der H-Brücke im Code	43
Abbildung 40: Einlesen der Buttons	44
Abbildung 41: Steuerung der Z-Achse	44
Abbildung 42: Steuerung der Stepper-Motoren	45
Abbildung 43: Einlesen des Joysticks	45

Abbildung 44: Christian bei der Verdrahtung (links) und Verdrahtung im Gerät (rechts)	46
Abbildung 45: Leistungsaufnahme, Spitzenwert	47
Abbildung 46: Effektive Stromaufnahme	47
Abbildung 47: Einstellen der Ausgangsspannung, Step-Down Converter	48
Abbildung 48: Stromaufnahme Greifer im Normalbetrieb	49
Abbildung 49: Messung einer unerklärlichen 1 (5 V).....	49
Abbildung 50: Abgeschnittener Pin 25	51
Abbildung 51: Mainboard von hinten.....	51
Abbildung 52: Ausgewertete Anforderungsliste	54

12 Literaturverzeichnis

Arduino SA. (2015). *Arduino Projects Book* (2 ed.). Ivrea, Italy: arduino.cc.

Arduino Documentation. (2021). Von arduino.cc: <https://www.arduino.cc/en/main/docs> abgerufen

Arduino Uno Specification. (2021). Von arduino.cc: <https://store-usa.arduino.cc/products/arduino-uno-rev3/> abgerufen

Arduino Mega Specification. (2021). Von arduino.cc:
<https://store.arduino.cc/collections/boards/products/arduino-mega-2560-rev3> abgerufen

SM. (5. 4 2019). *Arduino Tutorial Library*. Von arduino.cc:
<https://www.arduino.cc/en/Tutorial/LibraryExamples>HelloWorld> abgerufen

Official Arduino Forum. (2021). Von arduino.cc: <https://forum.arduino.cc/> abgerufen

17HS4401S Stepper-Motor Datasheet. (kein Datum). Von datasheetspdf.com:
<https://datasheetspdf.com/pdf-file/1310364/Handson/17HS4401S/1> abgerufen

Electronic Clinic Blog. (14. 8 2021). Von <https://www.electronicclinic.com/arduino-cnc-shield-v3-0-and-a4988-hybrid-stepper-motor-driver-joystick/> abgerufen

Electronic Clinic. (7. 4 2020). Von youtube.com:
https://www.youtube.com/watch?v=TMK_fLgpESQ abgerufen

crix. (15. 1 2020). *GRBL Arduino Pinout*. Von myhobby-cnc.de: <https://myhobby-cnc.de/wiki/doku.php?id=myhobby-cnc:grbl-pinout> abgerufen

Mallari, J. (2020). *Arduino I2C Communication*. Von circuitbasics.com:
<https://www.circuitbasics.com/how-to-set-up-i2c-communication-for-arduino/> abgerufen

niiraz. (19. 11 2020). *IC 7805 Voltage Regulator*. Von electronicsforu.com:
<https://www.electronicsforu.com/technology-trends/learn-electronics/7805-ic-voltage-regulator> abgerufen

Texas Instruments LM7805 Datasheet. (2021). Von ti.com:

<https://www.ti.com/lit/ds/symlink/lm340.pdf> abgerufen

N20 DC Motor Datasheet. (kein Datum). Von handsontec.com:

<https://www.handsontec.com/dataspecs/GA12-N20.pdf> abgerufen

Whadda Modul Step-Down Converter. (2021). Von digitec.ch:

<https://www.digitec.ch/de/s1/product/whadda-modul-dc-dc-step-down-lm2596s-fuer-arduino-entwicklungsboard-kit-14560458> abgerufen

Texas Instruments LM2596 Datasheet. (2021). Von ti.com:

<https://www.ti.com/lit/ds/symlink/lm2596.pdf> abgerufen

Forumsdiskussion zu LM7805 . (12. 3 2006). Von mikrocontroller.net:

<https://www.mikrocontroller.net/topic/41575> abgerufen

Veroboard Farnell Datasheet. (kein Datum). Von farnell.com:

<http://www.farnell.com/datasheets/10822.pdf> abgerufen

Artikel zur Entprellung. (7. 4 2017). Von mikrocontroller.net:

<https://www.mikrocontroller.net/articles/Entprellung> abgerufen

Texas Instruments H-Bridge Datasheet. (2021). Von ti.com:

<https://www.ti.com/lit/ds/symlink/l293.pdf> abgerufen

16x2 LCD Datasheet. (kein Datum). Von arduino.cc:

<https://www.arduino.cc/documents/datasheets/LCDscreen.PDF> abgerufen

Electronics Forum Heatsink . (kein Datum). Von electronics.stackexchange.com:

<https://electronics.stackexchange.com/questions/210198/when-to-use-a-heatsink-for-a-voltage-regulator> abgerufen

Arduino Documentation Analog Read. (2021). Von

<https://www.arduino.cc/reference/en/language/functions/analog-io/analogread/> abgerufen

PCB Design. (2007). Von upload.wikimedia.org:

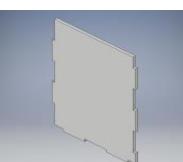
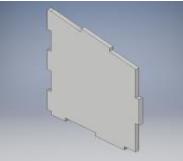
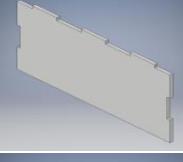
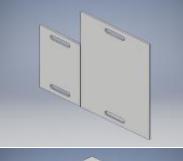
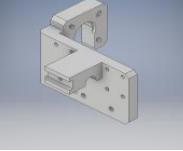
https://upload.wikimedia.org/wikipedia/commons/6/67/PCB_design_and_realisation_sm_mt_and_through_hole.png abgerufen

Fastermann, P. (2012). *3D-Druck/Rapid Prototyping: Eine Zukunftstechnologie*. Springer Vieweg.

Hintsteiner, M. (17. 8 2021). *Hinsteiner Group*. Von hinsteiner-group.com:
<https://www.hinsteiner-group.com/blog/rapid-prototyping> abgerufen

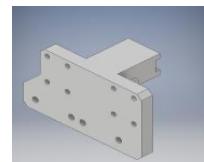
13 Stückliste von CAD

Item	Part Number	Thumbnail
1	Rahmen_Boden	
2	Rahmen_vorne	
3	Rahmen_Links	
4	Halter_Schiene	
5	Schiene	
6	Wagen_Schiene	
7	Rahmen_Rückwand	
8	Zahnriemen	
9	Motor	
10	Rahmen_Rechts	

11	Auswurf_hinten	
12	Auswurf_Rechts	
13	Bedienung_Seitenwand	
14	Bedienplatte	
15	Wagenplatte	
16	Bedienung_Vorderwand	
17	Auflage	
18	3D_Wagen_Mitte	
19	3D_Wagen_Links	
20	3D_Motorhalter	

21

3D_Wagen_Rechts



|

22

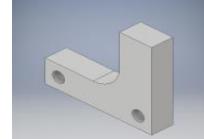
Halter_Schiene_Gelasert



|

23

Halter_Schiene_Gelasert_2



|

14 Anhänge

- Anforderungsliste
- Terminplan
- Funktionsstruktur
- Morphologischer Kasten
- Vor- / Nachteilliste
- Nutzwertanalyse mit S-Diagramm
- Kostenrechnung
- CAD-Daten (nur elektronisch)
- DXF-Files für Holz Laser (nur elektronisch)
- STL-Files für 3D-Drucker (nur elektronisch)
- 3D-PDF vom Greifautomat
- Schema
- Code
- Verdrahtungsplan
- Kontrollierte Anforderungsliste

15 Selbstständigkeitserklärung

Hiermit erklären wir, dass wir die vorliegende Arbeit selbstständig verfasst, alle beanspruchten Hilfsmittel und Quellen angegeben und keine unerlaubten Hilfen in Anspruch genommen haben. Wir haben zur Kenntnis genommen, dass bei Verstößen gegen diese Erklärung ein Abzug vorgenommen wird.

Uzwil, 16.12.2021

A handwritten signature consisting of four distinct parts: 'Elia', 'Nicola', 'Lionel', and 'Christoph' written vertically and joined together.