Kantonsschule Im Lee

Maturitätsarbeit HS 2022/23

Sparring-Roboter

-

Bau eines mechanischen Trainingspartners

Dustin Ngauv 4e

Betreuung: Wolfgang Pils
Winterthur, 9. Januar 2023

Inhalt

1 Vorwort

Jeder begeisterte Kampfsportler kennt die Box-Champions Mike Tyson und Muhammad Ali. Einer der vielen Gründe, weshalb diese beiden es im Boxring so weit gebracht haben, ist, dass sie nebst KO-Künstlern auch Meister in der Defensive waren. Ich selbst als Kampfsportler bewundere ihre Fähigkeit, gegnerischen Angriffen auszuweichen, sich zu verteidigen und es gleichzeitig noch kontrollierter und geschmeidiger aussehen zu lassen, als es in einem vorchoreographierten Kampf eines Actionfilms der Fall wäre. Um sich diese Fähigkeiten bis zu diesem Grad anzueignen, benötigt es nebst hartem Training auch jahrelange Kampferfahrung. Doch wenn es schon an fehlenden Trainingspartnern scheitern sollte, stellt sich die ganze Angelegenheit als schier unmöglich heraus. Trainingspartner zu finden, die immer verfügbar sind, ist nicht immer einfach, was während der Pandemie umso mehr der Fall war. Während dieser Zeit bin ich auf den Gedanken gekommen, dieses Problem mit einem Roboter zu lösen. Als ich dann erstmals auf ein vielversprechendes Design gestossen bin und sich mir durch die Maturarbeit eine gute Möglichkeit für die Umsetzung ergeben hat, ist aus dem Gedanken eine Vision und allmählich eine klare Idee geworden.

Vielen herzlichen Dank an alle Beteiligten, die mich während dieser Arbeit unterstützt haben.

Ich möchte mich bei meiner Betreuungsperson, Herrn Pils, bedanken, der mir durch die Besprechungen geholfen hat, eine Richtung für dieses Projekt zu finden.

Einen besonderen Dank möchte ich auch Michael Wick von der Werkstatt Wiederverwerkle in Winterthur Töss aussprechen, der mir während der stressigen Weihnachtszeit als einzige Werkstatt die Gelegenheit geboten hat, die Metall-Arbeiten fertigzustellen. Dazu hat er mir wertvolle Tipps gegeben, was ich sehr zu schätzen weiss.

Auch Unterstützung für die Näharbeit habe ich netterweise von meiner Grossmutter erhalten, weswegen ich ihr meinen grossen Dank äussern möchte.

Nicht zuletzt möchte ich mich auch bei meinen Eltern und bei meinem Bruder bedanken, die mich, wie immer, bei meinen Vorhaben und mit Korrekturlesen unterstützt haben.

2 Abstract

Ziel dieser Arbeit ist der Bau eines Roboters, welcher als Trainingspartner im Kampfsport dient. Im Gegensatz zu konventionelleren Trainingsmitteln wie Boxsäcke oder Pratzen soll der Roboter selbst Angriffe ausführen können, um die Reaktions- und Koordinationsfähigkeit des Trainierenden zu fördern. Er besitzt zusätzlich zwei Schlagpolster. Die Angriffe erfolgen durch Arme aus Schaumstoff, welche von Servomotoren gesteuert werden. Die Steuerung erfolgt über ein eingebautes Touch-Display und selbst geschriebene Software. Für diese Arbeit wird der Roboter nachmodelliert und zum grossen Teil 3D-gedruckt.

3 Einleitung

Im Kampfsport sind die Möglichkeiten des Einzeltrainings stark eingeschränkt. Viele Aspekte und Techniken können nur mithilfe eines Partners erlernt und trainiert werden. Eine der wichtigsten Fähigkeiten ist die Verteidigung gegnerischer Angriffe. Es reicht aber meist nicht, nur die Bewegung einer Technik auszuführen, solange nicht auf die Angriffe reagiert werden kann oder das richtige Timing für die Verteidigung nicht beherrscht wird. Um diese Fertigkeiten im Einzeltraining ohne Partner zu erlernen und zu verbessern, wird in dieser Arbeit ein Roboter gebaut, welcher Angriffe simuliert.

Ein solcher Roboter wurde vor wenigen Jahren vom amerikanischen Unternehmen «STRYK» entwickelt [1], doch war er bislang, bis zum Projektstartpunkt, noch nicht kommerziell erwerbbar.

4 Material und Methoden

4.1 Design

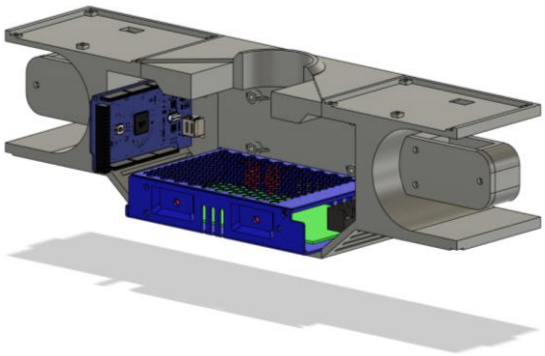


Abbildung 1: CAD-Modell des Roboters

Als Vorlage für das Design dient der RXT-1, ein Roboter von der Firma STRYK [1]. Zunächst wird mit Computer-Aided Design (CAD) das Gerät nachmodelliert. Verwendet wird dafür die Software Fusion 360 von Autodesk [2]. Als Referenzen, dienen dazu Bild- und Videomaterial des RXT-1, welches auf ihrer Website [1] und ihrem Instagram-Account [3] zu finden ist. Mithilfe eines 3D-Druckers lassen sich Designs schnell testen und allenfalls können aufgrund der Prototypen Anpassungen vorgenommen werden. Da manche Hersteller 3D-Modelle ihrer Produkte zum Download anbieten, werden diese ebenfalls beim Designprozess genutzt, um zum Beispiel ein Gefühl für Dimensionen zu bekommen oder um sicherzustellen, dass elektronische Bauteile in ihre Halterungen und ins Gehäuse passen. Natürlich werden auch Datenblätter bei diesem Prozess zur Hilfe herangezogen, jedoch werden die meisten Bauteile von Hand mit einem Messschieber nachvermessen. Entscheidend dabei ist, Toleranzen richtig einschätzen zu können und genügend Spielraum einzukalkulieren. Auch jeweilige Herstellungsmethoden und Verfahren schon beim Konstruieren zu bedenken, helfen sehr dabei, Konstruktionen später so umzusetzen, wie vorgestellt wird.

4.1.1 Struktur

Aufgebaut ist der Roboter aus einer Kontrolleinheit, einem Metallgerüst, einer Wandhalterung, aus je einem Kopf- und Torsoschlagpolster und aus vier Schaumstoffarmen. Die gesamte Elektronik wird im Gehäuse der Kontrolleinheit behaust und die Arme und das Kopfschlagpolster sind daran befestigt. Im Zentrum der Kontrolleinheit sitzt ein Touch-Display, womit der Roboter bedient wird. Die Kontrolleinheit lässt sich über einen simplen Mechanismus am Gerüst befestigen und kann jederzeit wieder abgenommen werden. Das Torsoschlagpolster sitzt ebenfalls direkt am Gerüst. Das Gerüst lässt sich in eine Halterung schieben, welche direkt an der Wand montiert wird. Das Gerüst besitzt eingebohrte Löcher, wodurch sich die Höhe verstellen lässt. Die Höhenverstellung erfolgt über einen Indexkolben, welcher an der Wandhalterung sitzt.

4.1.2 Gehäuse und Servohalterungen

Das Gehäuse, in welchem die ganze Elektronik lebt, kann grob in Deckelteil (Front Shell), Hauptgehäuse (Main Shell) und zwei zusätzliche Servohalterungen aufgeteilt werden. Front Shell und Main Shell werden jeweils in drei Teile aufgeteilt, da sie sonst zu gross für den 3D-Drucker wären. Aufgrund der groben durchschnittlichen Schulterbreite [4] wird beim Gehäuse für eine Breite von 40 cm entschieden.

Abbildung 2: Front Shell

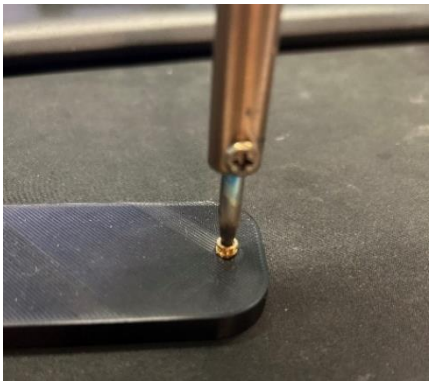


Abbildung 3: Heat Set Insert

Da in diesem Projekt sehr viele Schrauben verwendet werden, um Teile und Komponenten zu befestigen, wird das Gehäuse so konstruiert, um Gewindeeinsätze (Heat Set Inserts) installieren zu können. Die Vorteile hierbei, im Vergleich zu direktem Eindrehen der Schrauben oder Bohren der Gewinde, sind, dass Schrauben auch nach häufigem Gebrauch noch halten und die Gewindeeinsätze aus Messing nicht so schnell abgenutzt werden, was beim Kunststoff der Fall wäre. Weitere Vorteil sind auch die erhöhte Festigkeit und die Zuverlässigkeit der Verbindung. Insbesondere, da in diesem Projekt viele kleine Schrauben wie M2.5, M3 und M4 benötigt werden, ist der Gebrauch von Gewindeeinsätzen wichtig für funktionelle Teile. Die Löcher, in denen diese Einsätze installiert werden, sind dabei ein wenig kleiner als der Durchmesser des Inserts, damit das Gewinde das Material richtig greift, dazu etwas tiefer, damit das geschmolzene Material beim Einsetzen abfliessen kann, um das Hineinfliessen ins Gewinde zu verhindern. Ferner werden sogenannte Bosse verwendet, um einen Grossteil der Elektronik zu montieren. Diese Elemente ragen ein wenig aus dem Gehäuse heraus und haben Löcher, um Gewindeeinsätze zu installieren [5]. Für das Festschrauben der elektronischen Komponenten werden standardgemäss M3 Schrauben verwendet. M2.5 Schrauben, welche deutlich kleiner sind, werden dann eingesetzt, um Teile des Gehäuses aneinanderzuschrauben.

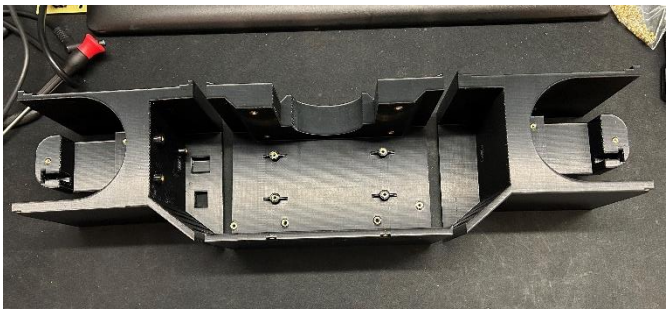


Abbildung 4: Main Shell in drei Teilen

An der hinteren Gehäusewand der Main Shell befinden sich 2 Löcher, um ein Netzteil festzuschrauben und dazu vier Bosse, um eine Leiterplatte zu befestigen. Diese vier Bosse werden von Rippen unterstützt und erleichtern auch den späteren 3D-Druck. Drei weitere Bosse befinden sich an der linken Innenwand, um einen Mikrocontroller zu montieren. Das Loch in der hinteren Gehäusewand ermöglicht den Zugang zum USB-Port des Mikrocontrollers. Auch wenn das Gehäuse hier geschlossen ist, kann dieser später dadurch programmiert werden. Auf der Rückseite befinden sich vier Löcher, um eine Halterungsvorrichtung anzuschrauben. Diese besteht aus einer zweiteiligen Monitorhalterung, dessen Komponenten sich ineinanderschieben und einrasten lassen und einfach wieder voneinander gelöst werden können. Am linken und rechten Ende des Gehäuses befinden sich zwei Halterungselemente für Servo-Motoren. Diese verfügen über einen Kanal, um die Kabel

der Servos herauszuführen. Die Servohalterungen haben jeweils einen Deckel, welcher mit drei Schrauben geschlossen werden. Oben an der Aussenseite befinden sich ebenfalls links und rechts Flächen, um zwei separate Servohalterungen anzubringen. Da sich die Halterungen und somit auch die Servos ausserhalb des Gehäuses befinden, können jeweils über einen schmalen Schlitz oder einem engen Kanal die Servokabel ins Gehäuseinnere geführt werden, wo sie angeschlossen werden. Damit genügend Luft im Gehäuse zirkulieren kann und die Elektronik nicht überhitzt wird, befinden sich auf der Unterseite links und rechts Lüftungsschlitze. Die obere Gehäusewand ist verstärkt, beziehungsweise verdickt, da dort ein Schlagpolster befestigt wird und die Stelle daher mechanischem Stress standhalten muss. Um die Installation des Schlagpolsters einfacher zu gestalten, ist dessen Loch für die Schraubfeder zur Hälfte geöffnet und wird mit dem Front Shell Gegenstück geschlossen, welches dort hineingeschoben wird. Die Verbindung zwischen Front Shell und Main Shell erfolgen über je vier Laschen oben und unten an der Front-Shell, welche sich über die Main Shell überlappen und so an den darunterliegenden Gewinden festgeschraubt werden können. Die Front-Shell hat in der Mitte einen leeren Rahmen, der für das Display vorgesehen ist. Das Display ist auf der Innenseite der Front-Shell an vier Bossen festgeschraubt. Auf der Unterseite befindet sich eine Öffnung und zwei Löcher, um einen Gerätestecker Sockel zu installieren.

Die oberen Servohalterungen sitzen jeweils auf einer flachen Platte, welche auf der Main Shell mit drei Schrauben montiert wird. Die Halterungen sind um 15° zur Seite geneigt und um 13° Richtung Zentrum gedreht, sodass sich der Nutzer vom Zentrum wegbewegen muss, um dessen Armen auszuweichen. Im Gegensatz zu den Servohalterungen an der Main Shell werden die Servos dort noch von der Seite gestützt. Es führen ebenfalls wieder Kabelkanäle aus der Halterung und über Schlitze zum Kanal ins Gehäuseinnere.

4.1.3 Arme und Klammern

Die Länge der Schaumstoffarme wird auf 65 cm festgelegt, da das grob der durchschnittlichen Armlänge entspricht [5]. Am Ende der Arme befindet sich ein Adapterstück mit Gewinde, wodurch sich die Arme am Roboter an- und abschrauben lassen. Verbunden ist der Adapter über einen Bolzen in dessen Zentrum, welcher in den Schaumstoff hineingeschoben wird.

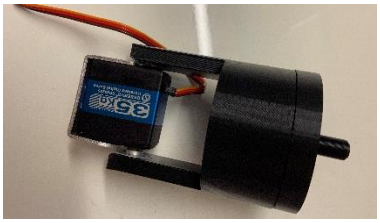


Abbildung 5: Servo mit Klammer und Adapter

Abbildung 6: Ganzer Arm

Die direkte Kraftübertragung der Servos erfolgt über Klammern, an denen die Arme angeschraubt werden. Die Klammern selbst haben zwei Vertiefungen und Schraubenlöcher, in denen die zwei runden Servo-Hörner Platz finden und verschraubt werden kö nnen. Da die Gewinde, an welche die Arme angeschraubt werden, 3D-gedruckt wird, wird das Innengewinde etwas vergrössert, um genug Toleranz zuzulassen.

4.1.4 Metallgerüst und Halterung

Abbildung 7: Metallgerüst

Das Gerüst besteht aus einem Vierkantrohr aus Stahl, welches 35 mm breit und hoch ist. Damit der Roboter einen kleinen Wandabstand zulässt, ist das Gerüst an zwei Stellen angewinkelt. Um den Roboter am Gerüst zu montieren, wird das Gegenstück der Monitorhalterung am oberen Ende des Vierkantrohrs befestigt. Etwas oberhalb der Mitte befinden sich zwei durchgebohrte Löcher, woran das Torsoschlagpolster mit längeren Schrauben befestigt wird. Sechs eingebohrte Löcher machen die Höhe bis zu 30 cm verstellbar.
Das Gerüst lässt sich auf einfache Weise teleskopartig in die Wandhalterung schieben und einrasten. Die Basis der Halterung bildet eine einfache Metallplatte. Die Platte ist ursprünglich eine Fernsehwandhalterung un d wird für dieses Projekt aufgrund ihrer hohen Belastbarkeit von 30 kg eingesetzt. In der Mitte der Platte sitzt ein weiteres Vierkantrohr mit 45 mm Höhe und Breite. Dieses lässt genug Toleranz, um das Gerüst geschmeidig hineinzuschieben, ohne dass es dabei stark wackelt. Ein Loch am oberen Ende des Vierkantrohrs ermöglicht, den Bolzen des Indexkolben hindurchzuschieben. Der Bolzen ist lang genug, um durch beide Rohrwände zu stossen und somit das Gerüst zu halten. An der Wandhalterung befestigt, ist der Indexkolben über eine angeschweisste M16 Mutter.

4.1.5 Schlagpolster

Das eine Schlagpolster repräsentiert den Kopf und das andere den Torsobereich und dienen im Training als Trefferziele. Die Schlagpolster selbst sind aus Schaumstoff und besitzen einen Kunstleder-Überzug. Sie sind mit einer Schraubfeder verbunden, um zusätzlich den Schock von Schlägen zu absorbieren. Das Kopfschlagpolster ist grundsätzliche in seiner Form würfelförmig, aber auf der Vorderseite an den Kanten abgeflacht, was zusätzliche Fläche für Variationen von Schlägen wie zum Beispiel Kinnhaken bietet. Das Schlagpolster hat eine Breite von 16 cm und eine Höhe von 20 cm, wodurch es etwa den Dimensionen eines menschlichen Kopfes entspricht [6]. Das Torsoschlagpolster hingegen hat eine dreieckige Form und bietet auf der Vorderseite eine grosse Schlagfläche. Es ist mit 20 cm in der Breite und 25 cm Höhe etwas grösser als das Kopfschlagpolster. Die Schlagpolster sind über ein Stahlrohr mit der Feder verbunden. Das Rohr sitzt beim Kopfschlagpolster dabei nicht mittig, sondern ist leicht nach hinten versetzt genau wie beim Hals des Menschen. Das Kopfschlagpolster ist direkt im Roboter verbaut, während das Torsoschlagpolster am Metallgerüst befestigt wird. Die Federn haben einen Durchmesser von 56 mm und eine Drahtdicke von 7 mm. Beim Kopfschlagpolster ist die Feder mit einer Platte verbunden, womit es am Gehäuse festgeschraubt wird. Für das Torsoschlagpolster wird die Feder mit einem Vierkantrohr so verbunden, dass sie parallel zum Boden verläuft, wenn es am Gerüst befestigt wird. Die Feder absorbiert dabei vor allem Schläge, welche von der Seite kommen, wie es bei zum Beispiel Haken oder Uppercuts der Fall ist.

4.2 Elektronik

Bei der Elektronik wirken unterschiedliche Komponenten zusammen, um die verschiedenen Anforderungen und Funktionen zu erfüllen. Da das ganze System mit 7.4 V betrieben wird, werden keine Spannungsregler benötigt und es braucht nur eine einzige Stromquelle, welche alle Komponenten mit genügend Strom versorgt. Vor der richtigen Implementation wird es am Gebrauch von Prototypen bedient, um die Funktionalität einzelner Systeme zu verifizieren.

4.2.1 Mikrocontroller

Als Gehirn des Roboters wird für dieses Projekt ein Arduino Mega 2560 R3 verwendet. Dieser kann über einen USB-B-Anschluss in den Programmiersprachen C und C++ programmiert werden [7]. Stromzufuhr erhält der Mikrocontroller über seinen Voltage-Input-Pin (VIN-Pin), welchem 7-12 Volt zugeführt wird [8]. Alle Digital-Pins, sowie einige andere werden vom Touch-Display verwendet. Der Arduino Mega verfügt ausserdem über Pins, welche Pulsweitenmodulations-Signale (PWM-Signale) ausgeben können. Vier dieser Pins werden verwendet, um die Servos zu steuern.

Abbildung 8: Arduino Mega 2560 R3

4.2.2 Servos

Die Servos sind die Aktuatoren des Roboters. Sie bewegen die Arme und führen die Befehle des Mikrocontroller aus. Dieses Projekt verwendet vier der RDS3235 Coreless Digital Servo, welche ein Drehmoment bis zu 35 kg-cm wirken können. Diese können über eine Betriebsspannung von 5-7.4 V laufen, wobei sie bei 7.4 V das grösste Drehmoment und die höchste Drehgeschwindigkeit entwickeln können. Unbelastet beträgt diese Höchstgeschwindigkeit gemäss der Datenblätter 0.11sek/60°. Der Blockierstrom bei maximaler Auslastung beträgt 2.3 A bei 7.4 V.

Abbildung 9: RDS3235 Coreless Digital Servo

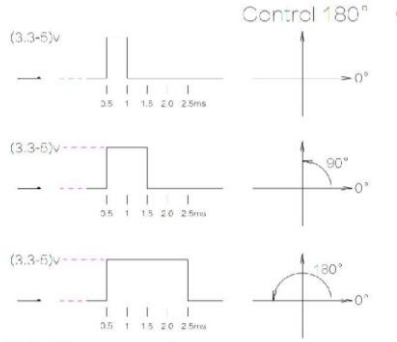


Abbildung 10: PWM Signal

Wie erwähnt, werden die Servos über eine Pulsweitenmodulation angesteuert. Der Winkel, auf den der Servoarm gestellt werden soll, wird über die Breite der Pulse gesteuert. Das Signal springt also immer für eine gewisse Dauer auf High-Pegel und ist für den Rest der Periodenlänge auf Low-Pegel [9]. Im Falle dieser Servos reicht die Breite von 500~2500 Mikrosekunden mit einer Arbeitsfrequenz von 50-330 Hz.

Im Gegensatz zu klassischen Modellbauservos, haben diese Servos keine Befestigungsvorrichtung, mit denen sie festgeschraubt werden könnten, dafür haben sie am Gehäuse nebst der gewöhnlichen Ausgangswelle auf der gegenüberliegenden Seite einen weiteren kurzen Stift, wodurch sich eine Klammer befestigen lässt.

4.2.3 Netzteil und Stromzufuhr

Abbildung 12: RSP-75-7.5

Das RSP-75-7.5 ist ein integriertes Schaltnetzteil, welches bis zu 10 Ampère Strom liefert. Die Spannung kann dabei mit einem kleinen Regler zwischen 7.13~8.25 Volt eingestellt werden. Eingestellt wird sie hier auf 7.4 Volt, da die Servos so die ihre höchste Geschwindigkeit erreichen kann. Ein Gerätestecker-Sockel mit eingebautem Ein-/Ausschalter und einer auswechselbaren Sicherung ist mit dem Netzteil verkabelt. Wenn der Sockel über ein Kaltgerätekabel an der Steckdose angeschlossen ist, kann so Wechselstrom zum Netzteil geführt werden, wo dieser dann für das Projekt in Gleichstrom umgewandelt wird.

Abbildung 11: Gerätestecker-Sockel

4.2.4 Touch-Display und Shield

Abbildung 13: Touch-Display

Das Display ist ein 5 Zoll Thin Film Transistor Liquid Crystal Display (TFT LCD) mit Touch-Funktion. Die Auflösung beträgt 800 x 480 Pixel und es verwendet die Resistive Touch Technologie. Da diese Art von Touchscreens auf Druck reagiert, wäre es auch möglich, den Touchscreen mit Handschuhen zu bedienen [10]. Die Stromzufuhr erfolgt über den Arduino. Da das Display mit 3.3 V läuft, wird ein Shield verwendet, um den Touchscreen mit dem Arduino zu verbinden, welcher nur eine Output-Spannung von 5 V hat. Das Shield senkt die Strom-Spannung und kann direkt auf den Arduino aufgesteckt werden. Das Touch-Display wird dann separat über ein Flachbandkabel an das Shield angeschlossen. Zwei Pins des Shields müssen entfernt werden, da diese gemäss den Datenblättern keine Funktion erfüllen, ansonsten wäre der Zugang zum VIN-Pin und Ground-Pin (GND-Pin) des Arduino blockiert.

4.2.5 Leiterplatte

Um die einzelnen elektronischen Komponenten miteinander zu verbinden, wird eine Leiterplatte dazu entwickelt. Verwendet wird dafür eine Lochrasterplatine, welche auf einer Seite Kupferringe um die Bohrungen aufweist, wodurch Elektronikbauteile angelötet werden können. Die Platine ist 5 x 10 cm gross und passt somit genau in das Gehäuse zwischen dem Netzteil und der Sprungfeder des Schlagpolsters. In den Ecken werden vier Löcher gebohrt, damit die Leiterplatte mit Schrauben im Gehäuse montiert wird.

Abbildung 14: Leiterplatte

Auf der Leiterplatte sitzen Reihenklemmen, Kondensatoren, eine Sicherung und Pin-Headers. Die 3 x 1 Pin-Headers werden verwendet, um die Servos daran anzuschliessen. Die positive (V+) und die negative Versorgungsspannung (V-) des Netzteils sind an Reihenklemmen der Leiterplatte angeschlossen und versorgen so die ganze Elektronik mit Strom. Eine selbstrückstellende Sicherung ist an der positiven Versorgungsspannung verbaut, welche ausgelöst wird, wenn die Stromstärke 2.7 A überschreitet. Diese wird zum Schutz eingebaut, da die Servos ansonsten überhitzt und beschädigt werden können, vor allem wenn sie einen zu hohen Strom aufgrund von beispielsweise Überlastung erfahren [11]. Dadurch, dass die Sicherung bei zu hohem Strom ihren elektrischen Widerstand durch Eigenerwärmung so sehr vergrössert, dass eventuell der Stromkreis unterbrochen wird und erst wieder einsatzfähig ist, sobald sie sich abgekühlt hat, muss sie nicht ausgetauscht werden, wenn sie einmal ausgelöst wird [12].
Der Strom wird parallelgeschaltet, da alle Komponenten mit derselben Spannung laufen und beliebig viele Elemente parallelgeschaltet werden können, ohne dass die Spannung aufgeteilt wird [13]. Da bei Servos der Strom- und Spannungsbedarf in kurzer Zeit stark auslenken und ansteigen kann, führt es tendenziell zu zittrigen oder ungewünschten Bewegungen, wenn die Stromversorgung nicht hinterherkommt, weshalb für eine solchen Fall Kondensatoren eingebaut werden. Für jeden Servo wurde ein Kondensator mit 470 µF Kapazität parallel angelegt, um die Versorgungsspannung zu stabilisieren und Auslenkungen zu puffern [14] [15].

Um den Arduino am Stromkreis anzuschliessen, werden ebenfalls Reihenklemmen auf der Leiterplatte verwendet. Auch die PWM-Signalkabel des Arduino werden über Reihenklemmen mit den Pins für die Servos verbunden.

4.3 Programmierung und Software

Die Software zeigt nach dem Start einen Home-Screen mit sechs Settings- und einem Start-Button. Über die Setting-Buttons gelangt man jeweils auf die respektiven Seiten, in denen die Einstellungen geändert werden können. Es ist möglich, die Anzahl an Runden, die Geschwindigkeit, die Rundendauer, der Schwierigkeitsgrad, die Pausenzeit und die jeweilige Anzahl hintereinander erfolgender Angriffe einzustellen. Beim Drücken des Start-Buttons wird der Benutzer zum Sparring-Modus geführt, in welchem er gegen den Roboter kämpfen kann, nachdem ein kurzer Countdown abgelaufen ist. Der Kampf findet in Runden statt, gefolgt von Pausen und endet, wenn entweder alle Runden durchgelaufen sind oder wenn der Kampf mit einem Knopfdruck auf den Stopp-Button abgebrochen wird.

Die Programmiersprache bei Arduino ist C oder C++. Der Code muss kompiliert und kann dann über ein USB-Kabel auf das Arduino Board geladen werden. Das Programm zur Steuerung des Roboters wird auf drei Dateien aufgeteilt, welche die Dateiendung .ino verwenden. In der Datei Screens.ino werden alle Funktionen definiert, welche für das User Interface relevant sind. Die einzelnen Funktionen in dieser Datei wechseln die Ansicht und sorgen für die Interaktion und Umsetzung der User-Inputs. In einer nächsten Datei wird hingegen in ServoControl.ino um die Steuerung der Servos gekümmert. Basierend auf der user-spezifischen Einstellung werden die Bewegungsmuster der Servos festgelegt. In der Main.ino Datei befindet sich vorwiegend Allgemeines, wie zum Beispiel die verwendeten Bibliotheken, globale Variablen, Klassenkonstruktoren und Standardeinstellungen. Im

Gegensatz zu Standard C oder C++ Programmen wird in Arduino statt einer main() Funktion die Funktionen setup() und loop() verwendet, welche sich ebenfalls in der Main.ino Datei befinden. Die setup() Funktion läuft jeweils einmal am Anfang des Programms und wird benutzt, um zum Beispiel das Display zu initialisieren oder um den Servos je einen Pin auf dem Arduino zuzuordnen. Wie der Name suggeriert, läuft Code, welcher sich in der loopFunktion befindet in einer Endlosschleife. Dadurch kann das Programm Variablen verändern, Daten lesen oder darauf reagieren [16].

Die Programmiersprache bei Arduino ist C oder C++. Der Code muss kompiliert und kann dann über ein USB-Kabel auf das Arduino Board geladen werden. Das Programm zur Steuerung des Roboters wird auf drei Dateien aufgeteilt, welche die Dateiendung .ino verwenden. In der Datei Screens.ino werden alle Funktionen definiert, welche für das User Interface relevant sind. Die einzelnen Funktionen in dieser Datei wechseln die Ansicht und sorgen für die Interaktion und Umsetzung der User-Inputs. In einer nächsten Datei wird hingegen in ServoControl.ino um die Steuerung der Servos gekümmert. Basierend auf der user-spezifischen Einstellung werden die Bewegungsmuster der Servos festgelegt. In der Main.ino Datei befindet sich vorwiegend Allgemeines, wie zum Beispiel die verwendeten Bibliotheken, globale Variablen, Klassenkonstruktoren und Standardeinstellungen. Im Gegensatz zu Standard C oder C++ Programmen wird in Arduino statt einer main() Funktion die Funktionen setup() und loop() verwendet, welche sich ebenfalls in der Main.ino Datei befinden. Die setup() Funktion läuft jeweils einmal am Anfang des Programms und wird benutzt, um zum Beispiel das Display zu initialisieren oder um den Servos je einen Pin auf dem Arduino zuzuordnen. Wie der Name suggeriert, läuft Code, welcher sich in der loop() Funktion befindet in einer Endlosschleife. Dadurch kann das Programm Variablen verändern, Daten lesen oder darauf reagieren [16].

4.3.1 Graphical User Interface

Zur Erstellung des Graphical User Interface (GUI) und zur Steuerung des Touch-Display werden die Bibliotheken UTFT, URTouch und UTFT_Buttons von Henning Karlsen verwendet [17]. Zuerst werden in der Design-Software Figma [18] Mockups vom User Interface erstellt, welche anschliessend in Arduino über die UTFT- und UTFT_Buttons-Bibliotheken implementiert werden. Für die Darstellung werden zusätzlich Fonts verwendet, welche heruntergeladen werden können [19].

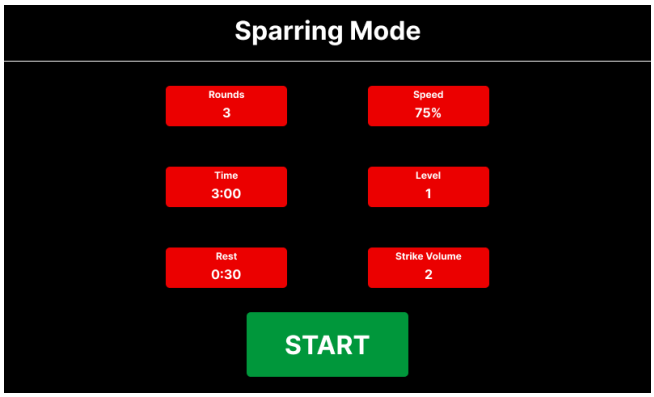


Abbildung 15: Homescreen Figma Mockup

Die Gestaltung und Positionierung der verschiedenen Buttons erfolgen über die UTFT_Buttons Bibliothek. Diese ermöglicht, auf einfache Weise ein Farbschema zu bestimmen und die Position, die Grösse und die Beschriftung eines jeweiligen Buttons zu definieren.

Abbildung 16: Sparring Mode Figma Mockup

Um Texte oder Beschriftungen darzustellen, wird meistens die myGLCD.print()-Funktion aufgerufen. Um Schriftzüge oder sonstige UI-Elemente upzudaten, muss das Element überschrieben werden, indem zum Beispiel die Funktion wieder mit einem neuen Wert aufgerufen wird.

Um zwischen den verschiedenen Screens zu wechseln, wird der ganze Display mit dem myGLCD.clrScr()-Befehl gecleart und ein neuer Screen kann gezeichnet werden.

4.3.2 Servosteuerung

Die allgemeine Steuerung der Servomotoren erfolgt mit der Bibliothek VarSpeedServo [20]. Durch diese kann die Geschwindigkeit eines Servos mit einem Wert zwischen 1-255 festgelegt werden. Da in diesem Programm die Geschwindigkeitsangabe in Prozent wiedergegeben wird, wird jeweils der Maximalwert 255 mit dem prozentualem Geschwindigkeitswert zusammengerechnet, welcher der User ausgewählt hat, um das erwünschte Ergebnis zu erreichen.

Da die Halterungen der Servos auf der linken und rechten Seite symmetrisch zueinander designt sind, schauen zwei Servos wegen deren Kabel sozusagen in die entgegengesetzte Richtung zu den zwei anderen Servos. Mit dem hohen Bewegungsgrad der Servos ist dies kein Problem, doch bei der Steuerung muss beachtet werden, dass die Steuerungswinkel sich von 0 - 90° zu 90 - 180° ändern. Die Angriffsbewegung wird daher für jeden Servo gemäss seiner Ausrichtung in einer Funktion definiert.

```
// functions to move each servo based on the speed settings

void rightStraight() {
  servo3.write(180, (255 * speed / 100), true);
  servo3.write(90, (255 * speed / 100), true);
}

void leftStraight() {
  servo2.write(0, (255 * speed / 100), true);
  servo2.write(90, (255 * speed / 100), true);
}

void rightHook() {
  servo4.write(180, (255 * speed / 100), true);
  servo4.write(90, (255 * speed / 100), true);
}
```

```
void leftHook() {
  servo1.write(0, (255 * speed / 100), true);
  servo1.write(90, (255 * speed / 100), true);
}
```

Grundsätzlich wird im SparringMode() über den random()-Befehl jeweils ein zufälliger Servo ausgewählt und bewegt und je nach User-Einstellungen ändert sich noch über eine For-Schleife die Anzahl hintereinander folgender Angriffe. Der SparringMode() hört auf, wenn die Runde zu Ende geht.

```
void SparringMode() {

  // executes set of strikes based on the strike volume settings
  for (int i = 0; i < volume; i++) {

    // stops servos if a round has ended
    if (countdowntime <= 1 || countdowntime > 4000000000) {
      break;
    }

    // chooses a random value between 1-4
    int servo = random(1,5);

    // moves a servo based on chosen value
    switch (servo) {
    case 1:
      leftHook();
      break;
    case 2:
      leftStraight();
      break;
    case 3:
      rightStraight();
      break;
    case 4:
      rightHook();
      break;
    }
  }
}
```

Die Funktion ServoDuration() kann anhand der eingestellten Geschwindigkeit die Zeitdauer grob abschätzen, welche ein Servo benötigt, um einen Angriff auszuführen. Die Formel dafür lautet wie folgt:

$$y = -410.2 \cdot \ln(0.2 \cdot x - 3) + 1445.5$$

Diese wird experimentell ermittelt. Die millis()-Funktion kann genaustens die vergangene Zeit angeben seit ein Programm auf dem Arduino gestartet ist, misst diese in Millisekunden. Wenn also damit die Differenz berechnet wird, welche sich aus den Zeiten ergibt, wann sich

ein Servo angefangen hat zu bewegen und wann er wieder aufgehört hat, kann die Zeit, die ein Servo für seine Bewegung benötigt hat, sehr genau ermittelt werden. Anhand der 17 Messungen unterschiedlicher Geschwindigkeiten von 20% bis 100% in Geschwindigkeitserhöhungen von 5% kann in Microsoft Excel anhand der Trendlinie eine grobe Formel gefunden werden, welche mit der eingestellten Geschwindigkeit eine Zeitdauer bestimmen kann.

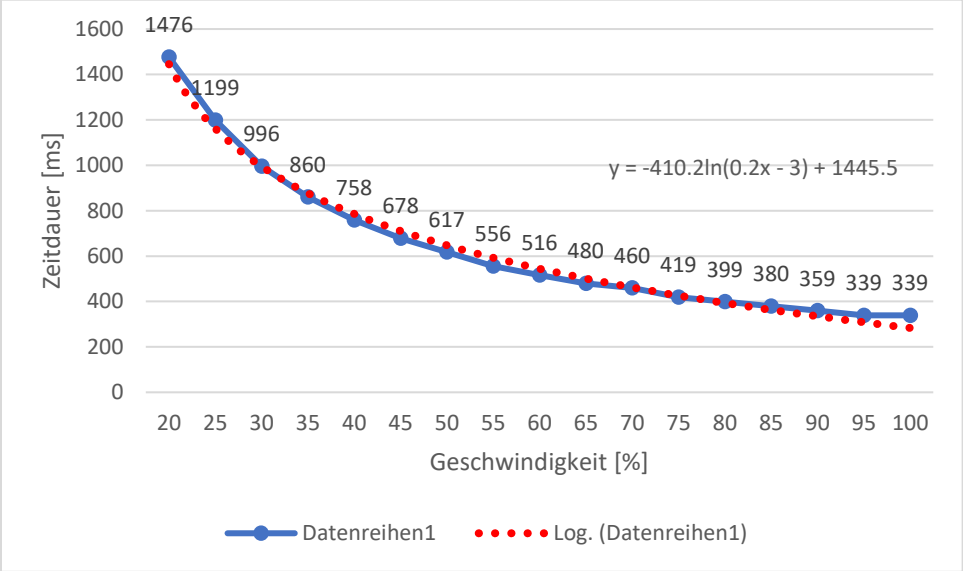


Diagramm 1 Benötigte Zeitdauer eines Servos bei gegebener Geschwindigkeit

Auf den Datenblättern wird in den Spezifikationen der Servos angegeben, dass ihre Maximalgeschwindigkeit bei 110ms/60° beträgt. Wenn das also auf die 180° Grad hochgerechnet wird, welche bei den Messungen jeweils zurückgelegten wurden, entsprechen die Angaben recht genau den tatsächlichen Messungen. Ausserdem kann erstaunlicherweise bei den Messungen mit oder ohne zusätzliche Last der Schaumstoffarme mit der beschriebenen Messmethode keinerlei Abnahme der Geschwindigkeit festgestellt werden.

Die ServoDuration()-Funktion wird benötigt, um die einzelnen Zeitabstände zwischen Angriffsserien zu bestimmen. Dieser Zeitabstand ist abhängig von der Anzahl Schläge in einer Angriffsserie und dem ausgewählten Schwierigkeitsgrad. Beim höchsten Schwierigkeitsgrad, sprich Level 3, ist die Zeitdauer zwischen den Angriffsserien willkürlich, was die Abwehr der Angriffe deutlich erschwert.

```
    // sets the delay between each set of attacks
    // based on level, strike volume and speed settings
    switch (level) {
      case 1:
      // 2500ms delay time
      delaybetween = volume * servoDuration(speed) + 2500;
      break;

      case 2:
      // 1500ms delay time
      delaybetween = volume * servoDuration(speed) + 1500;
      break;
```

```
        case 3:
        // random delay time between 500ms and 4000ms
        delaybetween = volume * servoDuration(speed) + random(500, 4000);
        break;
      }
```

4.3.3 Countdown-Timer

Die Funktionsweise des Countdown-Timers wurde einem Arduino-Projekt auf mechatrofice.com entnommen [21]. Der Timer verwendet ebenfalls die bereits beschriebene millis()-Funktion, um jeweils die Zeit zu ermitteln, die seit Programmstart gelaufen ist. Mit einer For-Loop wird die Anzahl an Runden und Pausen gezählt und es wird dann in einer While-Loop so lange mit der millis()-Funktion von der Rundenzeit abgezogen, bis diese Null beträgt. Wenn eine Runde zu Ende geht, wird der Countdown wieder zurückgesetzt, indem die Zeit wieder zurückaddiert wird, über welche der Timer gelaufen ist. Es wird anschliessend mit einem IF-Statement geprüft, ob als nächstes eine Runde oder eine Pause drankommt und die entsprechende Countdown-Zeit eingestellt. Bevor das Sparring losgeht, läuft der Countdown für 5 Sekunden, damit man Zeit hat, sich vorzubereiten. Da dieser Vorbereitungscountdown bereits zu den Runden in der For-Loop zählt, findet dies nur am Anfang statt, wenn die Bedingung (i == 0) erfüllt ist.

```
// holds the amount of seconds since the board has been started up
  unsigned long initialseconds = millis() / 1000;
  // either holds the value of the round duration or of the resting duration
  unsigned long duration;
  // holds the value to reset the countdown clock
  unsigned long added_time = 0;
  int currentRound = 1;
  float progress = 0.0;
  String TimeString;

  // stop button
  myGLCD.setFont(GroteskBold24x48);
  myButtons.setTextFont(GroteskBold24x48);
  myButtons.setButtonColors(VGA_WHITE, VGA_GRAY, VGA_BLACK, VGA_WHITE,
VGA_RED);
  int StopSpar_btn = myButtons.addButton(300, 380, 200, 80, "QUIT");

  // loops for the total amount of rounds, rest times and the initial
  countdown
  for (int i = 0; i < (2 * rounds); i++) {
    if (i != 0) {

      // prints the current round
      myButtons.drawButtons();
      myGLCD.print("Round " + String(currentRound) + "/" + String(rounds),
CENTER, 50);
      myGLCD.fillRoundRect(100, 315, 700, 300);
```

```
    }

    if (i % 2 == 1) {
      duration = time;
      myGLCD.print("   FIGHT!   ", CENTER, 120);
    }

    else if (i % 2 == 0 && i != 0) {
      currentRound += 1;
      duration = rest;
      myGLCD.print("Resting", CENTER, 120);
    }

      // updates the countdown clock every 100ms
      if (millis() >= 100 + time2 + initialseconds * 1000) {
        time2 += 100;
        if (i == 0) {
          countdowntime = 5 + initialseconds + added_time - (millis() /
1000);  // initial 5 second preparation countdown
          TimeString = countdowntime % 60;

          // displays the preparation countdown
          myGLCD.setFont(SevenSegment96x144Num);
          myGLCD.print(TimeString, CENTER, 150);
          myGLCD.setFont(GroteskBold24x48);
        }

        else {
          // countdown clock during round or rest times
          countdowntime = (duration / 1000) + initialseconds + added_time -
(millis() / 1000);
          // calculates current progress
          progress = 1 - (countdowntime / float(duration / 1000));

          unsigned long countdown_minute = ((countdowntime / 60) % 60);
          unsigned long countdown_sec = countdowntime % 60;

          if (countdown_sec < 10) {
            TimeString = String(countdown_minute) + ":" + "0" +
String(countdown_sec);
          }

          else {
            TimeString = String(countdown_minute) + ":" +
String(countdown_sec);
          }

          // displays countdown clock
          myGLCD.print(TimeString, CENTER, 200);
```

```
          myGLCD.setColor(0, 26, 255);
          myGLCD.fillRoundRect(100, 315, 100 + progress * 600,
300);
          myGLCD.setColor(255, 255, 255);
        }

        // checks if round or rest time has ended
        if (countdowntime == 0 || countdowntime > 4000000000) {
          myGLCD.clrScr();
          break;
        }
      }
    }

    // checks if sparring has finished
    if ((countdowntime == 0 || countdowntime > 4000000000) && (i + 1) == (2 *
rounds)) {
      delay(1000);
      drawHomeScreen();
      break;
    }

    // resets countdown clock
    added_time = (millis() / 1000) - initialseconds;
  }
```

4.4 Bau und Herstellung

4.4.1 3D-Druck und Verarbeitung

Abbildung 17: Creality Ender-3 V2

Der 3D-Druck gehört zu den additiven Fertigungsverfahren, bei dem das Material Schicht für Schicht aufgetragen wird, um ein Werkstück zu erzeugen [22]. Der für dieses Projekt verwendete Drucker ist der Ender-3 V2 von Creality [23], welcher sich konkret dem Fertigungsverfahren Fused Deposition Modeling (FDM) bedient [24]. Der Werkstoff mit dem gedruckt wird, ist ein Filament aus Polylactide (PLA). Nachdem ein 3D-Modell als Datei exportiert wird, muss das Modell noch mit einem Slicer-Programm, in diesem Fall Ultimaker Cura, bearbeitet werden. Danach wird es in G-code umgewandelt, welches vom Drucker eingelesen wird. Im Slicer können Anpassungen zur Druckeinstellungen vorgenommen oder auch die Orientierung des Modells ausgerichtet werden.

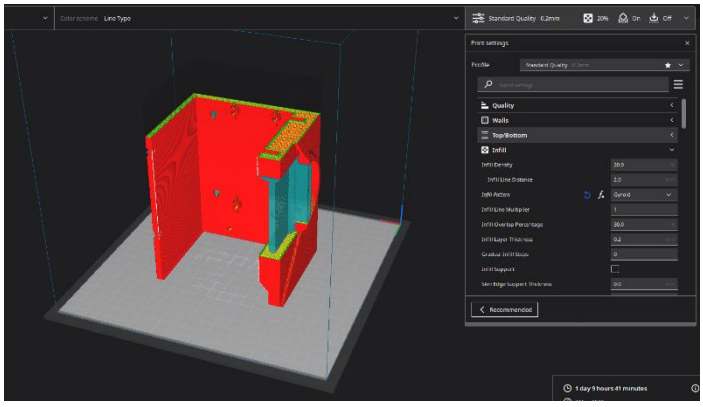


Abbildung 18: Modell im Slicer

Die meisten 3D-gedruckten Objekte sind nicht massiv, sondern verwenden einen sogenannten Infill. Für die meisten Teile wird dabei mit 10% Infill gedruckt. Dafür wird das Infill-Pattern auf Gyroid eingestellt. Aufgrund der speziellen Geometrie dieser Form sind die gedruckten Teile trotz des niedrigen Infill-Anteils sehr stabil, da die Seitenwände von allen Richtungen gestützt werden [25]. Ein weiterer wichtiger Faktor für einen starken 3D-Druck ist die Anzahl an Perimetern beziehungsweise die Dicke der Aussenwände liegt hier meist bei drei. Nur einmal wird für zusätzliche Stabilität mit 4 Perimetern und auch gleichzeitig 20% Infill gedruckt. Da die Druckqualität abnimmt, je steiler der Überhang ist, der gedruckt werden muss, werden zusätzliche Stützstrukturen eingesetzt, welche mitgedruckt und anschliessend entfernt werden. Je nachdem können aber auch komplette Überhänge ohne Supports gedruckt werden, indem mit Bridging gearbeitet wird [26]. Teilweise wird für ein Teil bis zu 41 Stunden lang gedruckt. Um die Gewindeeinsätze zu installieren, werden diese über die vorgedruckten Löcher ausgerichtet und können danach mit einem Lötkolben in wenigen Sekunden erhitzt und in den 3D-Druck eingeschmolzen werden.

Das Gehäuse, welches aufgrund seiner Grösse in jeweils drei Teilen gedruckt wird, wird anschliessend zusammengeschweisst. Nachdem die Werkstücke fixiert sind, können die Verbindungstellen auf der Innenseite mittels Lötkolben so bearbeitet werden, dass sie zusammenschmelzen und so ein robustes Stück bilden. Dabei muss beachtet werden, dass nicht durch das Teil hindurchgeschmolzen wird, aber dass die Teile dennoch genug fest miteinander verschweisst sind. Ein Stück Filament kann als zusätzlicher Werkstoff dienen, um die Schweissnähte wieder mit Material aufzufüllen. Da bei der Verbrennung von PLA giftige Gase entstehen können, ist bei diesem Prozess Vorsicht geboten und er sollte ihn in einer gut durchlüfteten Umgebung stattfinden [27].

4.4.2 Löten

Zunächst werden alle Komponenten an der Leiterplatte ausgerichtet und anschliessend angelötet. Da für die Leiterplatte nicht ein Printed Circuit Board (PCB) verwendet wird, bei welchem die Leiterbahnen schon vorgefertigt sind, müssen diese daraufhin manuell gefertigt werden. Dies wird erreicht, indem Bahnen mit Lötzinn gelegt werden und bei sich überkreuzenden Verbindungen die Enden eines Kabels an die entsprechenden Stellen gelötet werden. Mit einem Multimeter wird sichergestellt, dass alle Bahnen richtig verbunden sind und dass keine ungewollten Verbindungen besteht.

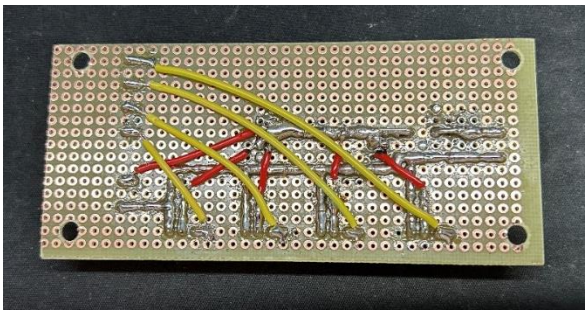


Abbildung 19: Leiterplatte mit Leiterbahnen

4.4.3 Metallverarbeitung

Da ich bisher über relativ wenig Erfahrung in der Metallverarbeitung verfüge, wurde zuerst mit dem Werkstattmeister abgesprochen, inwiefern und auf welche Weise sich meine Konstruktionen umsetzen lassen. Angefertigte technische Zeichnungen werden hierbei und auch in späteren Arbeitsschritten als Referenz herbeigezogen.

Abbildung 20: Vierkantrohre sägen

Zunächst wird das Gerüst gebaut. Dafür werden Vierkantrohre, welche den gestellten Anforderungen entsprechen, herausgesucht und in der Bandsäge in drei Stücke zugeschnitten. Das Vierkantrohr muss zuvor abgemessen und die Schnitte mit einem Stift vormarkiert werden. Um das Rohr abzuwinkeln, werden die jeweiligen Schnitte mit einem Winkel von 67.5° gemacht und später wieder gedreht zusammengeschweisst.

Da die Höhenverstellung nur funktionieren kann, wenn sich der Bolzen genau in die Löcher schieben lässt, müssen diese möglichst genau gebohrt werden. Um die Bohrungen exakt vorzumarkieren, wird ein Höhenanreisser eingesetzt, mit dessen Hilfe sich zum Beispiel die genaue Mitte des Vierkantrohres finden lässt. Bevor das Stück eingespannt und die Löcher anschliessend in der Bohrmaschine gebohrt werden, wird das vorgesehene Bohrloch mit angekörnt, was das Bohren einfacher und präziser macht. Da es schnell gefährlich werden kann, wenn beim Bohren der Gegenstand hochgerissen wird und sich mitdreht, kann bei Metall je nach Lochgrösse und Material langsamer durchgebohrt, kleinere Drehzahlen verwendet und die Bohrstelle mit Schneideöl versehen werden [28]. Für die Höhenverstellung werden sechs M9 Löcher gebohrt und für die Befestigung des Torso-Schlagpolsters im Mittelstück zwei M5 Löcher durchgebohrt. Die Löcher werden danach entgratet.

Abbildung 21: Vierkantrohre bohren

Um die Vierkantrohre für das Schweissen vorzubereiten, wird die Oberfläche rund um die Kanten mit einem Winkelschleifer bearbeitet, um das Werkstück von der Oxidationsschicht zu befreien. Das anschliessend verwendete Schweissverfahren ist das Tungsten-Inert-Gas-Welding (TIG-Welding). Beim TIG-Schweissen brennt ein elektrischer Lichtbogen zwischen dem Werkstück und einer Elektrode aus Wolfram [29]. Die Werkstücke werden mit einer Zwinge fixiert, an den Ecken kurz getackert beziehungsweise zusammengeschweisst, bevor sie in einer sauberen Linie zusammengeschweisst werden. Für das Gerüst wird beim Schweissen kein Zusatzwerkstoff verwendet. Das Stück der Monitorhalterung wird unkompliziert am Gerüst angeschraubt. Die zwei Teile der Halterung können auseinandergeschraubt werden, um beim hinteren Stück das Bohren zweier Schraubenlöcher erleichtert. In das Gerüst werden zwei Löcher mit Gewinde gebohrt, was die einfache Befestigung einer Schraube ohne Hilfe einer Schraubmutter ermöglicht. Die Halterung kann wieder zusammengebaut werden, nachdem die hintere Platte am Gerüst angeschraubt ist.

Abbildung 22: Vierkantrohre schweissen

Für die Wandhalterung wird ein weiteres Vierkantrohr vormarkiert und zugeschnitten. Dieses ist ein wenig grösser als das andere, sodass das Gerüst hindurchgeschoben werden kann. Es wurde davon abgeraten, das Vierkantrohr direkt an die Metall-Platte zu schweissen, da sonst der Lack auf der Platte entfernt werden muss und das Risiko besteht, dass sich die Platte beim Schweissen verbiegen würde. Als alternative Lösung wird das Vierkantrohr zuerst an ein anderes Blech angeschweisst, vier Löcher in das Blech gebohrt und an den bereits bestehenden Löchern der Metall-Platte angeschraubt. Am oberen Ende des Vierkantrohrs wird ein M16 Loch gebohrt, wonach rundherum dann eine M16 Mutter an das Rohr geschweisst wird, um den Indexkolben mühelos anschrauben zu können. Eine zusätzliche Unterlegscheibe wird zwischen den Indexkolben gelegt, da dessen Gewinde ansonsten das Vierkantrohr innendran blockieren würde und man das Gerüst nicht mehr hindurchschieben könnte. Inwiefern sich die Verbindungsstücke der Schlagpolster umsetzen lässt, ist zunächst ein wenig ungewiss, da Federstahl aufgrund des höheren Kohlenstoffgehalts nicht einfach schweissbar ist und auch je nachdem seine Federeigenschaften verlieren kann [30].

Es wird probeweise versucht, an einer Feder zu schweissen und da dies gelungen ist, ist diese Methode, trotz der erwähnten Risiken, hier zur Anwendung gekommen. An der Feder wird das Metall-Aktivgas-Schweissverfahren (MAG-Schweissen) angewendet. Das MAG-Schweissverfahren gehört wie TIG-Schweissen zur Gruppe des Schutzgasschweissens und wird bei Stahl eingesetzt [31].

Beim Kopfschlagpolster wird eine Stahlplatte benötigt, welche die angeforderten Masse von 80 x 120x 5 mm hat. In diese werden vier Löcher in die Ecken gebohrt, die das spätere Anschrauben im Gehäuse gewähren. Anschliessend wird die Oberfläche mit dem Winkelschleifer bearbeitet, um an ihr zu schweissen. Die Feder muss zuerst noch in die richtige Länge zugeschnitten werden. Um die Enden der Federstücke flach zu machen, werden diese angemessen abgeschliffen. Nachdem die Feder an die Platte geschweisst ist, wird die Schweissnaht mit dem Winkelschleifer abgeschliffen, damit alles besser ins Gehäuse passt. Beim Schweissen ist die Platte leicht verbogen, wodurch auch die Platzierung der bereits gebohrten Löcher entsprechend verschoben ist und die Löcher daher nochmals ausgefeilt werden.

Abbildung 23: Fertiges Gerüst

Das Vierkantrohr, welches bereits für das Gerüst verwendet wird, kommt nochmals für das Verbindungstück des Torsoschlagpolsters zum Einsatz und wird wieder mit der Bandsäge zugeschnitten. Zwei kurze Flachstangen werden herausgesucht, welche als Laschen dienen, um das Torsoschlagpolster am Gerüst zu befestigen. In die Laschen wurden jeweils zwei Löcher gebohrt, welche mit dem Höhenanreisser vormarkiert und dann an das Gerüst geschraubt werden, bevor die Laschen an das Vierkantrohr geschweisst werden. Danach kann zusätzlich das andere Federstück direkt an das Vierkantrohr geschweisst werden. Nicht zuletzt lässt sich ein Stahlrohr von 42.5 mm Durchmesser finden, wodurch es perfekt in die Feder passt. Das Rohr bildet jeweils das Verbindungstück zwischen Polster und Feder und wird in der Bandsäge zugeschnitten.

4.4.4 Polsterung

Für die beiden Schlagpolster werden zunächst im Schaumstoffladen zwei Blöcke auf das grob gewünschte Mass zugeschnitten. Die erwünschten Formen werden nachträglich mit einem gezackten Messer herausgearbeitet. Für den Polsterbezug werden vorab Schablonen erstellt, indem die Form des Schaumstoffes auf Papier übertragen wird, mit Berücksichtigung der Naht von ca. 1 cm. Mit den Schablonen können die benötigten Stoffstücke aus Kunstleder herausgeschnitten werden. Ein Bezug besteht jeweils aus drei Stoffstücken. Nachdem bestimmt worden ist, wo die Öffnung des Überzugs sein soll, wird von der gegenüberliegenden Seite her angefangen, die Stoffstücke mit der Nähmaschine zusammenzusetzen. Klettverschlüsse werden angebracht, um die Stoffenden an der Öffnung zu verschliessen.

5 Fazit

In dieser Arbeit ist es mir gelungen, einen Roboter zu bauen, welcher durch den Einsatz von Servomotoren selbstständig Angriffe ausführt und somit die Möglichkeiten des Einzeltrainings im Kampfsport erweitert. Über ein Touch-Display kann das Training mit diesem Gerät individualisiert werden. Der Roboter hat Schlagpolster, welche als Trefferziele dienen und ist in der Höhe verstellbar. Durch verschiedene Mechanismen kann der Roboter innert kurzer Zeit aufgebaut und zum Beispiel für den Transport auf einfache Weise in handhabliche Teile wieder abgebaut werden.

Der Arbeitsprozess hierbei war recht zeit- und auch kostenaufwändig. Der Roboter funktioniert in seinen beschriebenen Funktionen, doch öfters reagiert das Touch-Display nach dem Aufstarten nicht auf Eingaben, obwohl das Display selbst ein Bild anzeigt. Dies scheint allerdings nur der Fall zu sein, wenn das Display beziehungsweise der Mikrocontroller über das Netzteil betrieben wird und gleichzeitig die Servos angeschlossen sind. Woran die Unzuverlässigkeit des Touch-Display genau liegt, ist momentan noch unklar, aber das Problem könnte womöglich umgangen werden, indem der Mikrocontroller und das Display jeweils über eine separate Stromquelle über den USB-Port statt VIN-Pin betrieben werden.

Was die Software angeht, wird sie zum jetzigen Zeitpunkt recht einfach gehalten und kann auf jeden Fall weiter ausgebaut werden. Zum Beispiel indem der Nutzer die Option bekommt, eigene Bewegungskombinationen laufen zu lassen oder die Angriffsmuster spezifischer auf den jeweiligen Kampfsport anzupassen. Da der USB-Port des Mikrocontrollers von aussen erreichbar ist, lässt sich die Software aber sehr einfach updaten und erweitern.

Die Schlagpolster wurden nur bedingt getestet. Da die Federn geschweisst wurden, ist es gut möglich, dass die Federkraft mit der Zeit abnimmt oder dass sich die Feder mit der Zeit brüchig zeigt. Es ist daher nicht vorhersehbar, wie viel Schlagkraft die Polster tatsächlich aushalten. Demnach ist zu empfehlen, sie eher als Trefferziele zu verwenden, als wie bei einem Boxsack üblich wäre.

In dieser Arbeit wird nur der Aspekt, simple, gegnerische Angriffe mit einem Roboter zu simulieren, in Betracht gezogen. Natürlich kann man unterdessen auch die Idee einer mechanischen Trainingshilfe erweitern, indem zum Beispiel Bewegung im Raum miteingebracht oder die Angriffsvielfalt ausgeweitet wird.

Dank dieses Projektes habe ich viele wertvolle Einblicke in die Vielfältigkeit des Ingenieurwesens sowie der Manufaktur gewinnen können. Ich konnte meine Interessen in Sachen Technik und Sport kombinieren und mittels der erlangten interdisziplinären Kompetenzen ein Produkt herstellen, welches mir Mehrwert bietet und ich in meiner Freizeit nutzen kann.

Abbildung 25: Fertiger Sparring-Roboter

Abbildung 24: Fertiger Sparring-Roboter

6 Literaturverzeichnis

[1] STRYK, „STRYKUSA,“ [Online]. Available: https://www.strykusa.com/.

[2] Autodesk, „Fusion 360,“ [Online]. Available: https://www.autodesk.ch/de/products/fusion-360/overview.

[3] Instagram, „STRYK,“ [Online]. Available: https://www.instagram.com/stryk.usa/.

[4] healthline, „What's an Average Shoulder Width,“ [Online]. Available: https://www.healthline.com/health/average-shoulder-width.

[5] Wikipedia, „Boss (engineering),“ [Online]. Available: en.wikipedia.org/wiki/Boss_(engineering).

[6] Alexa Answers, „What Is The Average Human Head Size,“ [Online]. Available: https://alexaanswers.amazon.com/question/4ZL5w6cFqjpKsboBxbgEcr.

[7] Wikipedia, „Arduino,“ [Online]. Available: https://en.wikipedia.org/wiki/Arduino.

[8] Arduino, „Powering Alternatives for Arduino Boards,“ [Online]. Available: https://docs.arduino.cc/learn/electronics/power-pins.

[9] Wikipedia, „Servo,“ [Online]. Available: https://de.wikipedia.org/wiki/Servo.

[10] Wikipedia, „Resistive Touchscreens,“ [Online]. Available: https://de.wikipedia.org/wiki/Touchscreen#Resistive_Touchscreens.

[11] Kontrolmek, „The 7 Most Common Servo Motor Faults,“ [Online]. Available: https://kontrolmek.com/resources/blog/the-7-most-common-servo-motor-faults/.

[12] Wikipedia, „Selbstrückstellende Sicherungen,“ [Online]. Available: https://de.wikipedia.org/wiki/Selbstr%C3%BCckstellende_Sicherung.

[13] Wikipedia, „Parallelschaltung,“ [Online]. Available: https://de.wikipedia.org/wiki/Parallelschaltung.

[14] StackExchange, „Using a capacitor to properly power a servo,“ [Online]. Available: https://electronics.stackexchange.com/questions/175431/using-a-capacitor-to-properly-power-a-servo.

[15] Wikipedia, „Stützkondensator,“ [Online]. Available: https://de.wikipedia.org/wiki/St%C3%BCtzkondensator.

[16] Arduino, „loop(),“ [Online]. Available: https://www.arduino.cc/reference/de/language/structure/sketch/loop/.

[17] H. Karlsen, „Rinky-Dink Electronics,“ [Online]. Available: http://www.rinkydinkelectronics.com/.

[18] Figma, [Online]. Available: https://www.figma.com/de/.

[19] H. Karlsen, „UTFT Fonts,“ [Online]. Available: http://www.rinkydinkelectronics.com/r_fonts.php.

[20] P. v. Allen, „VarSpeedServo Github,“ [Online]. Available: https://github.com/netlabtoolkit/VarSpeedServo.

[21] Mechatrofice, „Arduino countdown LCD display code hour:minute:second format,“ [Online]. Available: https://mechatrofice.com/arduino/arduino-countdown-lcd-display-in-hhmmss-format.

[22] Wikipedia, „3D-Druck,“ [Online]. Available: https://de.wikipedia.org/wiki/3D-Druck.

[23] Creality, „Ender-3 V2,“ [Online]. Available: https://www.creality3dofficial.com/de/products/ender-3-v2-3d-printer.

[24] Wikipedia, „Fused Deposition Modeling,“ [Online]. Available: https://de.wikipedia.org/wiki/Fused_Deposition_Modeling.

[25] Wevolver, „Understanding the Gyroid Infill in 3D Printing,“ [Online]. Available: https://www.wevolver.com/article/understanding-the-gyroid-infill-in-3d-printing.

[26] The 3D Printer Bee, „Bridging im 3D-Druck: Überänge und Winkel Perfekt Drucken,“ [Online]. Available: https://the3dprinterbee.com/de/bridging-3d-drucken-winkel-uberhange/.

[27] 3D Print Scape, „Is PLA Toxic When Burned? Facts and Safety Info,“ [Online]. Available: https://3dprintscape.com/is-pla-toxic-when-burned-facts-and-safety-info/.

[28] BefestigungsFuchs, „Bohren in Metall - worauf ist zu achten,“ [Online]. Available: https://www.befestigungsfuchs.de/blog/tipps-fuer-das-bohren-in-metall/.

[29] Wikipedia, „Wolfram-Inertgas-Schweißen,“ [Online]. Available: https://de.wikipedia.org/wiki/Wolfram-Inertgas-Schwei%C3%9Fen.

[30] Weldingintro, „Can you Weld Spring Steel [An Inforamtive Guide],“ [Online]. Available: https://weldingintro.com/can-you-weld-spring-steel/.

[31] Wikipedia, „Schutzgasschweissen,“ [Online]. Available: https://de.wikipedia.org/wiki/Schutzgasschwei%C3%9Fen.

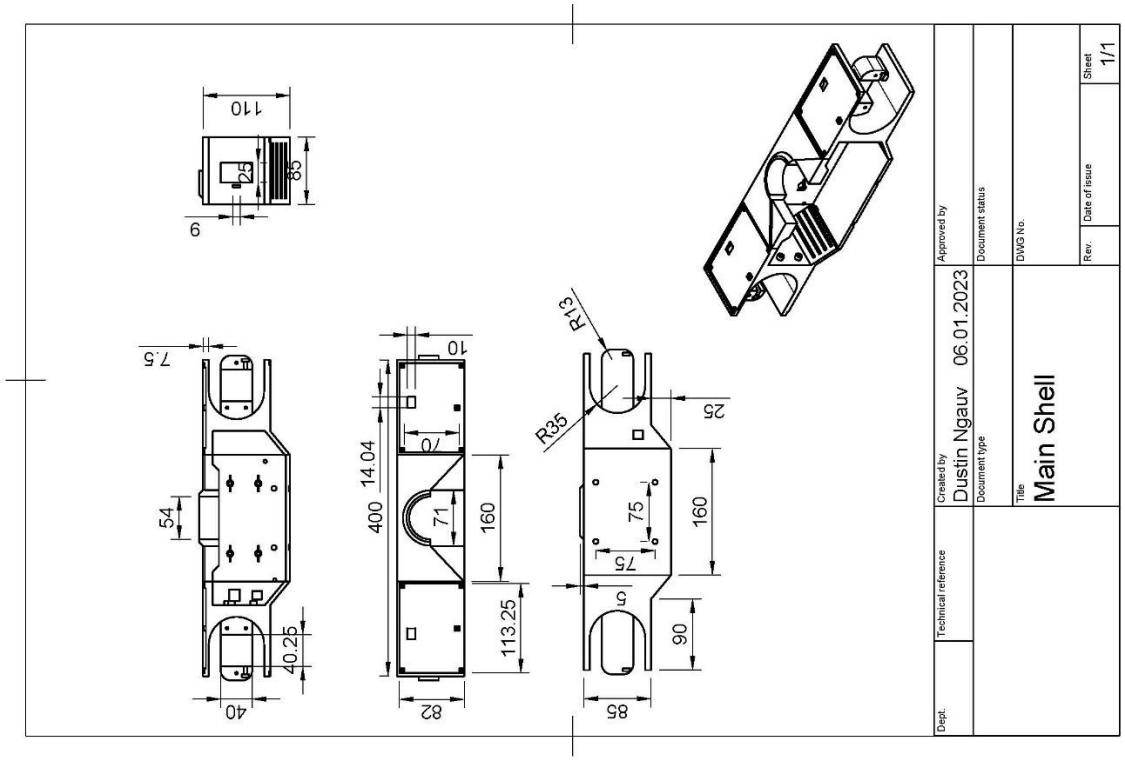
[32] Reference, „durchschnittliche Armlänge,“ [Online]. Available: https://www.reference.com/science-technology/long-average-human-arm-62c7536c5e56f385.

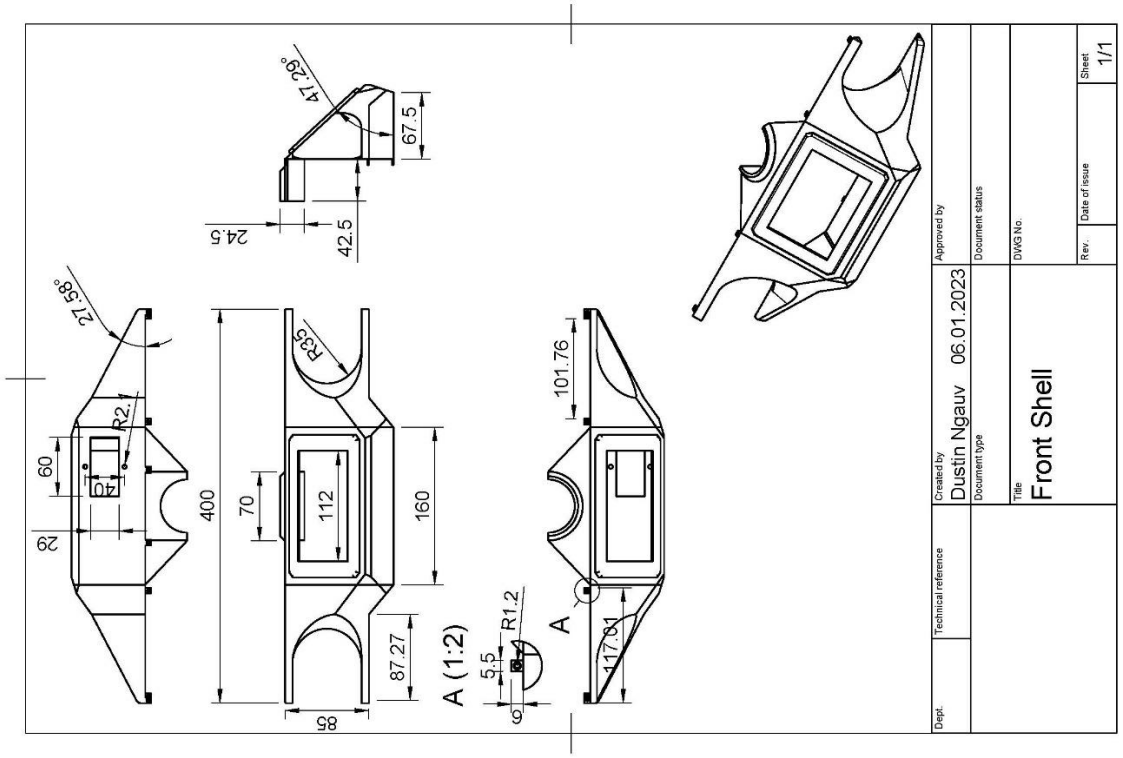
[33] ALL3DDP, „3D Printing Stregh: How to 3D Print Strong Parts,“ [Online]. Available: https://all3dp.com/2/3d-printing-strength-strongest-infill/.

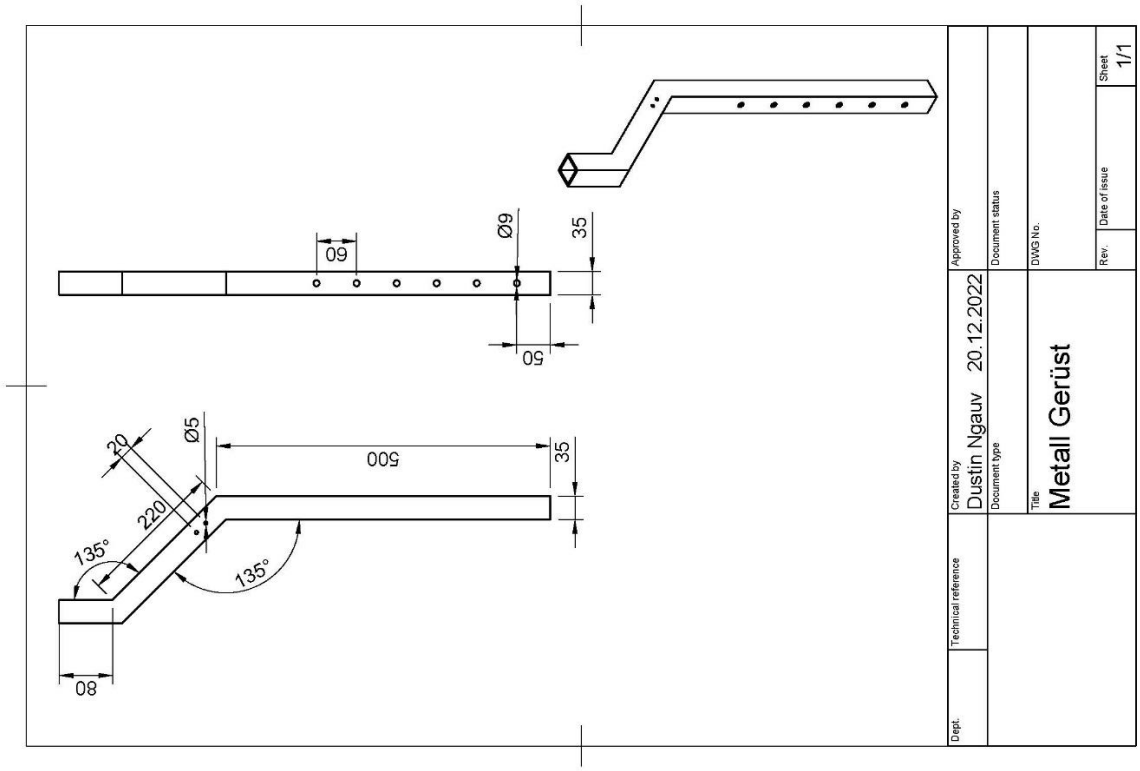
Alle Internetquellen wurden zuletzt am 06.01.2023 besucht

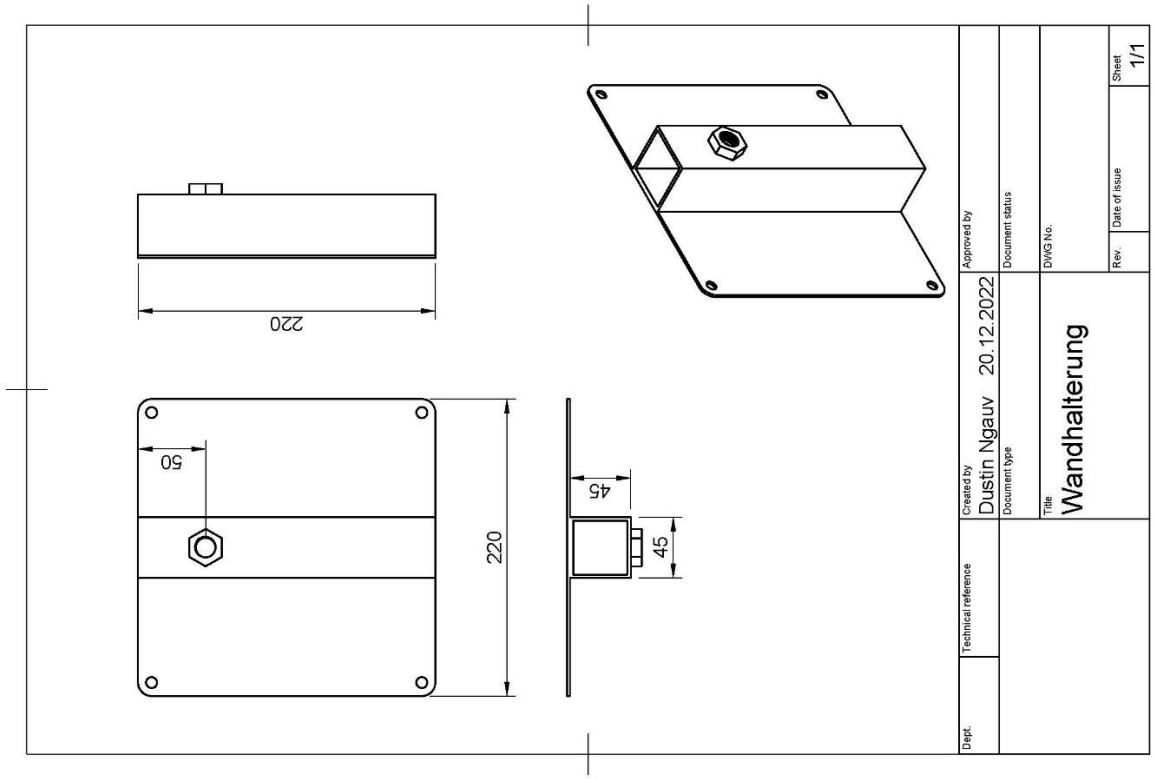
7 Abbildungsverzeichnis

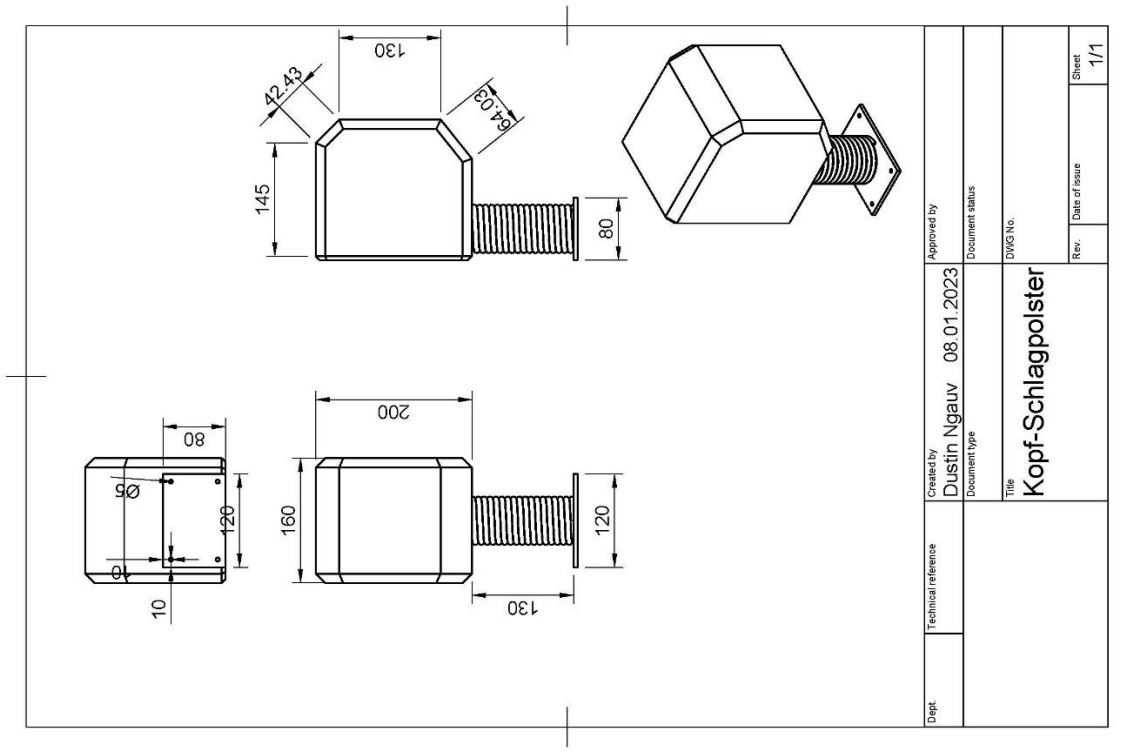
8 Anhang

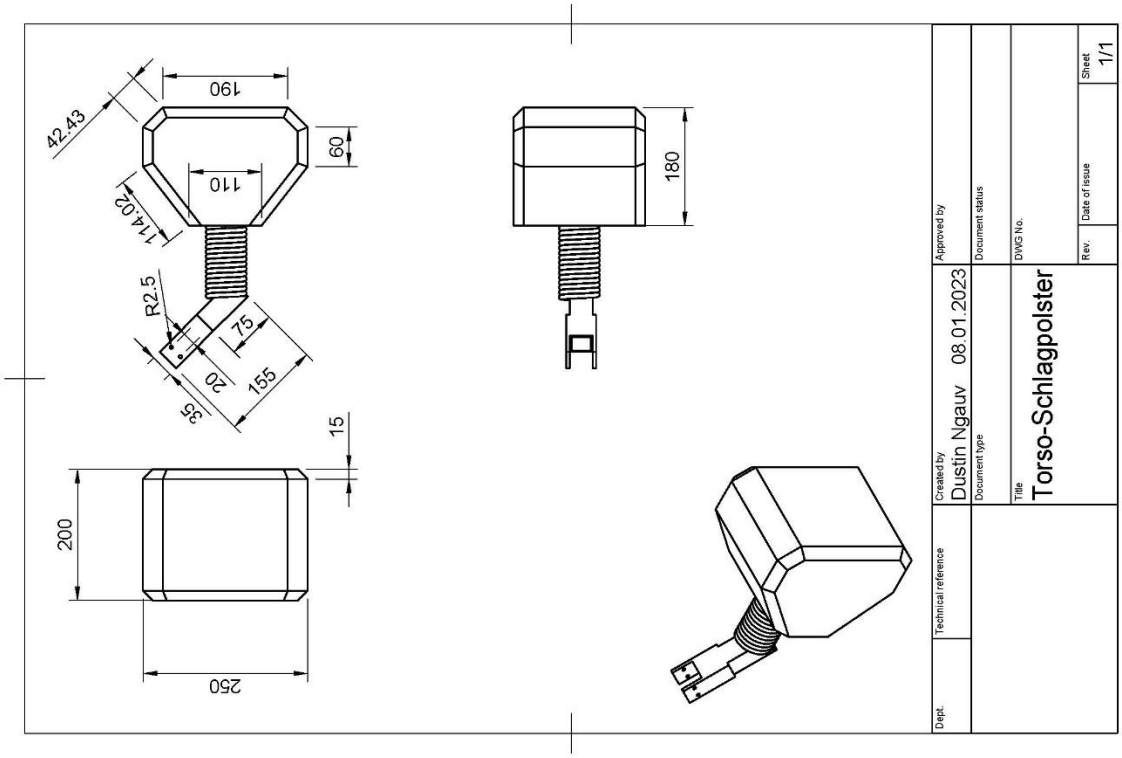

110
85
25
9
7.5
54
40.25
40
14.04
400
10
70
71
160
113.25
82
R13
R35
25
75
75
160
90
85
Dept.
Technical reference
Created by
Dustin Ngauv 06.01.2023
Approved by
Document type
Document status
Title
Main Shell
DWG No.
Rev.
Date of issue
Sheet
1/1

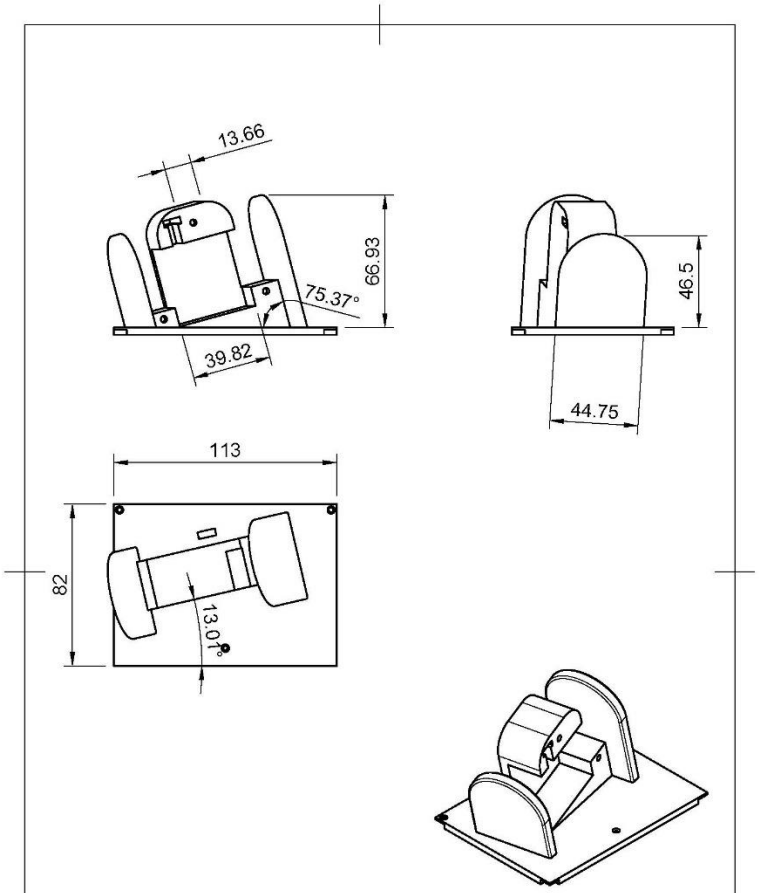

41.29°
67.5
24.5
42.5
27.58°
R2.1
60
29
400
70
112
160
R35
87.27
85
A (1:2)
5.5
R1.2
6
A
101.76
117.01
Dept.
Technical reference
Created by
Dustin Ngauv 06.01.2023
Approved by
Document type
Document status
Title
Front Shell
DWG No.
Rev.
Date of issue
Sheet
1/1


Ø9
35
60
50
Ø5
20
220
500
35
135°
135°
80
Dept.
Technical reference
Created by
Dustin Ngauv 20.12.2022
Document type
Approved by
Document status
Title
Metall Gerüst
DWG No.
Rev.
Date of issue
Sheet
1/1

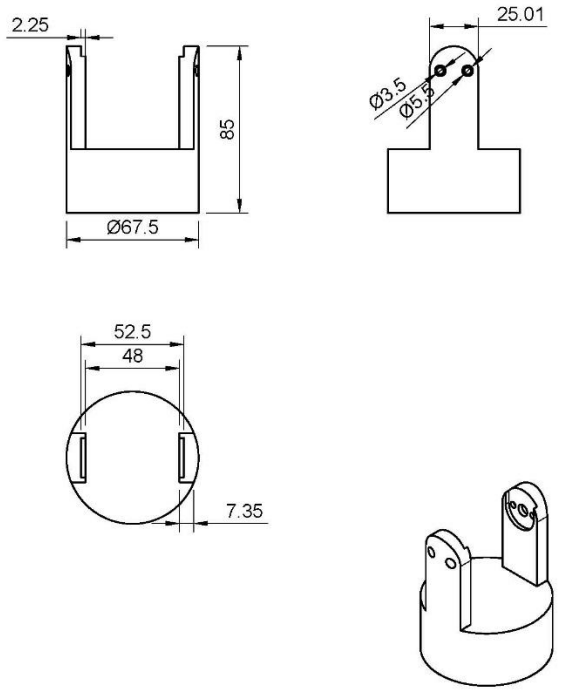

220
50
220
45
45
Dept.
Technical reference
Created by
Dustin Ngauv 20.12.2022
Document type
Approved by
Document status
Title
Wandhalterung
DWG No.
Rev.
Date of issue
Sheet
1/1


42.43
130
64.03
145
80
200
160
120
130
80
Ø5
120
10
10
Dept.
Technical reference
Created by
Dustin Ngauv 08.01.2023
Approved by
Document type
Document status
Title
Kopf-Schlagpolster
DWG No.
Rev.
Date of issue
Sheet
1/1

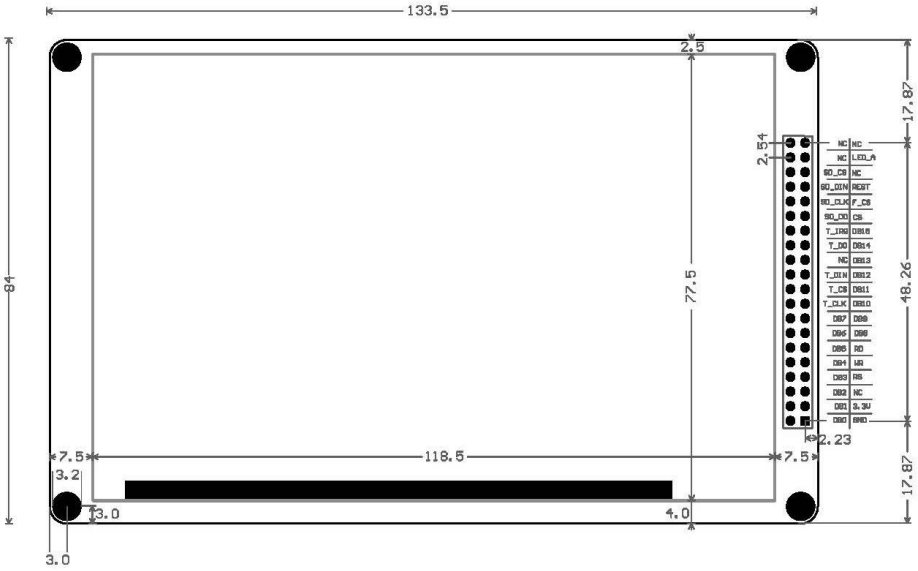

42.43
190
60
110
114.02
R2.5
75
20
155
35
15
200
250
180
Dept.
Technical reference
Created by
Dustin Ngauv 08.01.2023
Approved by
Document type
Document status
Title
Torso-Schlagpolster
DWG No.
Rev.
Date of issue
Sheet
1/1

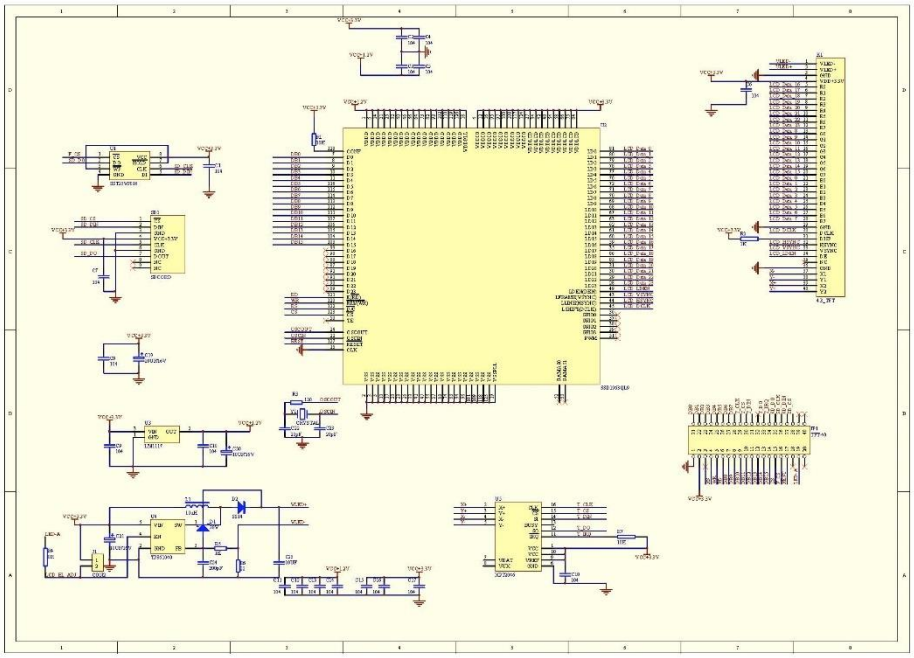


Dept.	Technical reference	Created by Dustin Ngauv 06.01.2023	Approved by
		Document type	Document status
		Title Obere Servo Halterung	DWG No.
			Rev. / Date of issue / Sheet 1/1

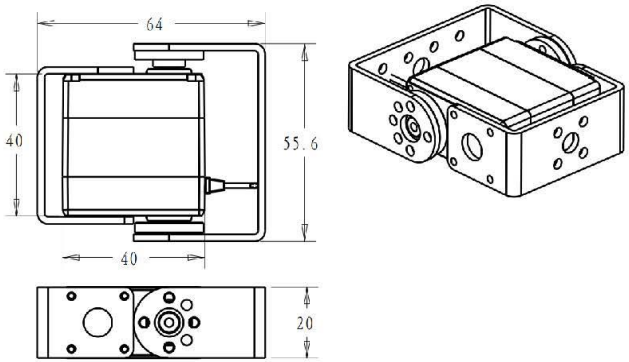


Dept.	Technical reference	Created by Dustin Ngauv 08.01.2023	Approved by
		Document type	Document status
		Title Servo Klammer	DWG No.
			Rev. / Date of issue / Sheet 1/1


133.5
2.5
17.87
2.54
84
77.5
48.26
7.5
118.5
7.5
3.2
2.23
17.87
3.0
4.0
3.0



产品型号 (Product Name): RDS3235
产品描述 (Product Description): 7.4V 35kg Coreless Digital Servo
产 品 图 (Drawing)



1. 使用环境条件 Apply Environmental Condition

No.	Item	Specification
1-1	存储温度 Storage Temperature Range	-30℃～80℃
1-2	运行温度 Operating Temperature Range	-15℃～70℃
1-3	工作电压范围 Operating Voltage Range	5-7.4V

2. 机械特性 Mechanical Specification

No.	Item	Specification
2-1	尺寸 Size	40*20*40mm
2-2	重量 Weight	60g
2-3	齿轮比 Gear ratio	373
2-4	轴承 Bearing	Double bearing
2-5	舵机线 Connector wire	300±5mm
2-6	马达 Motor	Coreless motor
2-7	防水性能 Waterproof performance	No

更多的舵机规格书和 3D 模型，请到网站下载(For more servo datasheet and 3D files, please go to the website to download) www.dsservo.com

东莞市达盛舵机科技有限公司
Dongguan City Dsservo Technology Co.Ltd

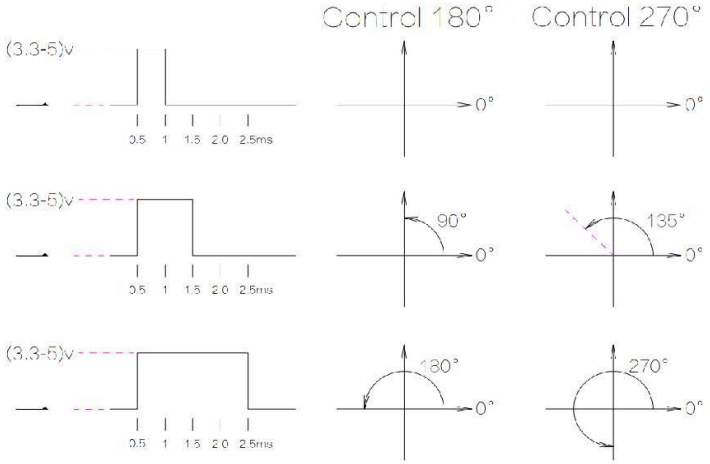
3. 电气特性 Electrical Specification

No.	工作电压 Operating Voltage	5V	6V	7.4V
3-1	待机电流 Idle current(at stopped)	5mA	5mA	5mA
3-2	空载转速 Operating speed (at no load)	0.13 sec/60°	0.12 sec/60°	0.11sec/60°
3-3	堵转扭矩 Stall torque (at locked)	29 kg-cm	32 kg-cm	35 kg-cm
3-4	堵转电流 Stall current (at locked)	1.9A	2.1 A	2.3A

4. 控制特性 Control Specification

No.	Item	Specification
4-1	驱动方式 Control System	PWM(Pulse width modification)
4-2	脉宽范围 Pulse width range	500～2500μsec
4-3	中点位置 Neutral position	1500μsec
4-4	控制角度 Running degree	180° or 270° (when 500～2500 μ sec)
4-5	控制精度 Dead band width	2 μsec
4-6	控制频率 Operating frequency	50-330Hz
4-7	旋转方向 Rotating direction	Counterclockwise (when 500～2500 μsec)

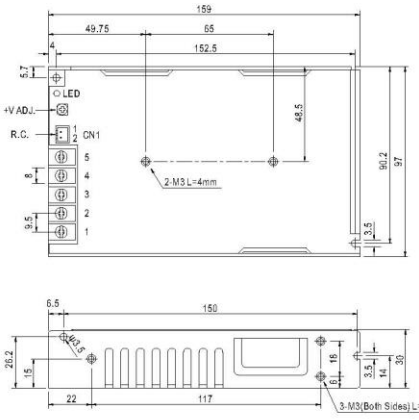
5. 关于 PWM 控制说明 About PWM Control



更多的舵机规格书和 3D 模型，请到网站下载(For more servo datasheet and 3D files, please go to the website to download) www.dsservo.com

■ Mechanical Specification

Case No.227A Unit:mm



Terminal Pin No. Assignment

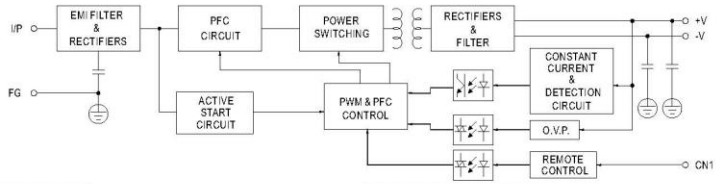
Pin No.	Assignment	Pin No.	Assignment
1	AC/L	4	DC OUTPUT -V
2	AC/N	5	DC OUTPUT +V
3	FG ⏚		

Remote ON/OFF(CN1): JST S2B-XH or equivalent(optional)

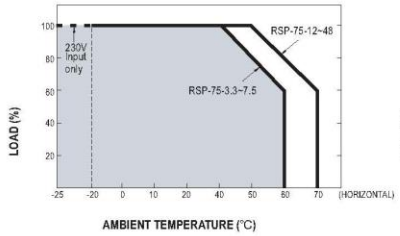
Pin No.	Assignment	Mating Housing	Terminal
1	RC+	JST XHP or equivalent	JST SXH-001T-P0.6 or equivalent
2	RC-		

■ Block Diagram

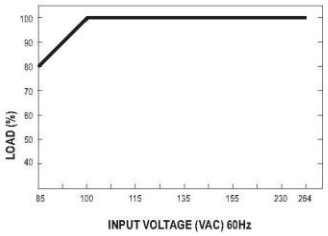
PFC fosc : 65KHz
PWM fosc : 80KHz



■ Derating Curve



■ Output Derating VS Input Voltage



■ Features :
- Universal AC input / Full range
- Built-in active PFC function
- Protections: Short circuit / Overload / Over voltage
- Cooling by free air convection
- Built-in constant current limiting circuit
- 1U low profile 30mm
- Remote ON-OFF control
- LED indicator for power on
- Over voltage category Ⅲ
- 100% full load burn-in test
- 3 years warranty

SPECIFICATION

MODEL		RSP-75-3.3	RSP-75-5	RSP-75-7.5	RSP-75-12	RSP-75-13.5	RSP-75-15	RSP-75-24	RSP-75-27	RSP-75-48
OUTPUT	DC VOLTAGE	3.3V	5V	7.5V	12V	13.5V	15V	24V	27V	48V
	RATED CURRENT	15A	15A	10A	6.3A	5.6A	5A	3.2A	2.8A	1.6A
	CURRENT RANGE	0 ~ 15A	0 ~ 15A	0 ~ 10A	0 ~ 6.3A	0 ~ 5.6A	0 ~ 5A	0 ~ 3.2A	0 ~ 2.8A	0 ~ 1.6A
	RATED POWER	49.5W	75W	75W	75.6W	75.6W	75W	76.8W	75.6W	76.8W
	RIPPLE & NOISE (max.) Note.2	80mVp-p	80mVp-p	80mVp-p	120mVp-p	120mVp-p	120mVp-p	120mVp-p	120mVp-p	200mVp-p
	VOLTAGE ADJ. RANGE	3.14 ~ 3.63V	4.75 ~ 5.5V	7.13 ~ 8.25V	11.4 ~ 13.2V	12.8 ~ 14.9V	14.3 ~ 16.5V	22.8 ~ 26.4V	25.7 ~ 29.7V	45.6 ~ 52.8V
	VOLTAGE TOLERANCE Note.3	±2.0%	±2.0%	±2.0%	±2.0%	±2.0%	±2.0%	±1.0%	±1.0%	±1.0%
	LINE REGULATION	±0.5%	±0.5%	±0.5%	±0.5%	±0.5%	±0.5%	±0.5%	±0.5%	±0.5%
	LOAD REGULATION	±1.0%	±1.0%	±1.0%	±0.5%	±0.5%	±0.5%	±0.5%	±0.5%	±0.5%
	SETUP, RISE TIME	600ms, 30ms at full load								
	HOLD UP TIME (Typ.)	16ms at full load								
INPUT	VOLTAGE RANGE	85 ~ 264VAC 120 ~ 370VDC								
	FREQUENCY RANGE	47 ~ 63Hz								
	POWER FACTOR (Typ.)	PF>0.93/230VAC PF>0.98/115VAC at full load								
	EFFICIENCY (Typ.)	76%	82%	84%	85%	85%	86%	87%	88%	89%
	AC CURRENT (Typ.)	0.9A/115VAC 0.5A/230VAC								
	INRUSH CURRENT (Typ.)	COLD START 35A/230VAC								
	LEAKAGE CURRENT	<2mA / 240VAC								
PROTECTION	OVERLOAD	105 ~ 135% rated output power								
		Protection type : Constant current limiting, recovers automatically after fault condition is removed								
	OVER VOLTAGE	3.63 ~ 4.46V	5.5 ~ 6.75V	8.25 ~ 10.13V	13.2 ~ 16.2V	14.85 ~ 18.23V	16.5 ~ 20.25V	26.4 ~ 32.4V	29.7 ~ 36.45V	52.8 ~ 64.8V
		Protection type : Shut down o/p voltage, re-power on to recover								
	OVER TEMPERATURE	Shut down o/p voltage, recovers automatically after temperature goes down								
FUNCTION	REMOTE CONTROL	CN1: < 0~0.8VDC POWER ON , 4~10VDC POWER OFF								
ENVIRONMENT	WORKING TEMP.	-25 ~ +70°C (Refer to "Derating Curve")								
	WORKING HUMIDITY	20 ~ 90% RH non-condensing								
	STORAGE TEMP., HUMIDITY	-40 ~ +85°C, 10 ~ 95% RH non-condensing								
	TEMP. COEFFICIENT	±0.05%/°C (0 ~ 50°C)								
	VIBRATION	10 ~ 500Hz, 2G 10min./1cycle, 60min. each along X, Y, Z axes								
	OVER VOLTAGE CATEGORY	Ⅲ ; According to EN61558, EN50178, EN60684-1, EN62477-1 ; altitude up to 2000 meters								
SAFETY & EMC (Note 4)	SAFETY STANDARDS	UL60950-1, TUV EN60950-1, EN61558-1,EN61558-2-16, EAC TP TC 004, CCC GB4943.1, BSMI CNS14336-1 approved								
	WITHSTAND VOLTAGE	I/P-O/P:4KVAC I/P-FG:2KVAC O/P-FG:0.5KVAC								
	ISOLATION RESISTANCE	I/P-O/P, I/P-FG, O/P-FG:100M Ohms / 500VDC / 25°C / 70% RH								
	EMC EMISSION	Compliance to EN55032 (CISPR32) Class B, EN61000-3-2,-3, EAC TP TC 020, CNS13438, GB9254 Class B, GB17625.1								
	EMC IMMUNITY	Compliance to EN61000-4-2,3,4,5,6,8,11, EN55024, light industry level, criteria A, EAC TP TC 020								
OTHERS	MTBF	296.7K hrs min. MIL-HDBK-217F (25°C)								
	DIMENSION	159*97*30mm (L*W*H)								
	PACKING	0.44Kg; 30pcs/14.2Kg/0.91CUFT								
NOTE	1. All parameters NOT specially mentioned are measured at 230VAC input, rated load and 25°C of ambient temperature.									
	2. Ripple & noise are measured at 20MHz of bandwidth by using a 12" twisted pair-wire terminated with a 0.1uf & 47uf parallel capacitor.									
	3. Tolerance : includes set up tolerance, line regulation and load regulation.									
	4. The power supply is considered a component which will be installed into a final equipment. The final equipment must be re-confirmed that it still meets EMC directives. For guidance on how to perform these EMC tests, please refer to "EMI testing of component power supplies."									
	5. The ambient temperature derating of 3.5°C/1000m with fanless models and of 5°C/1000m with fan models for operating altitude higher than 2000m(6500ft). (as available on http://www.meanwell.com)									

Der Vollständige Code kann auf GitHub gefunden werden:

https://github.com/DustinNgauv/Sparring-Roboter