Konstruktion eines 6-achsigen Roboterarms

Vladimir Morozov  
Gymnasium Bäumlihof  
Begleitperson: Dominik Rohner

13. Oktober 2025

# Vorwort

Lorem Ipsum

**Inhaltsverzeichnis**

[Vorwort 2](#_Toc207804567)

[Zusammenfassung 5](#_Toc207804568)

[Einleitung 6](#_Toc207804569)

[Theoretischer Hintergrund 7](#_Toc207804570)

[Das Arbeitsprinzip eines 6-Achsen Roboterarms 7](#_Toc207804571)

[Verwendete Materialien und deren Wirkungsweisen 8](#_Toc207804572)

[Elektrische Teile 8](#_Toc207804573)

[Meschanische Teile 8](#_Toc207804574)

[Programmierung 8](#_Toc207804575)

[Weitere Werkzeuge 8](#_Toc207804576)

[3D-Drucker 8](#_Toc207804577)

[Fusion 360 und Solidworks 8](#_Toc207804578)

[Visual Studio Code 8](#_Toc207804579)

[PyCharm 8](#_Toc207804580)

[KiCAD 8](#_Toc207804581)

[Praktische Umsetzung 9](#_Toc207804582)

[Resultate 10](#_Toc207804583)

[Quellenverzeichnis 11](#_Toc207804584)

# Vorwort

Ich möchte mich herzlich bei allen Mitwirkenden bedanken, die mir bei diesem Projekt geholfen haben.

# Zusammenfassung

Lorem Ipsum

# Einleitung

Die Idee einen Roboterarm zu konstruieren war die letzte Iteration aus vielen anderen Ideen, die zu mir, während der Gymnasialzeit gekommen ist. Grundsätzlich war nie die Idee etwas darzulegen, oder etwas zu beweisen. Sondern, mich dazu aufzufordern, ein Projekt bis zum Schluss zu führen, welches alle Themenbereiche beinhalten, die mich interessieren oder die ich in der Zukunft erforschen würde. Zu diesen Themenbereichen gehören:

* Informatik (Programmieren, Problemlösung)
* Elektrotechnik (Elektronik, Datenübergabe, Elektrizität)
* Mechanik (grundsätzlich Robotik)
* Konstruktion (CAD, 3D-Druck)

Somit ist die allgemeine Konstruktion eines Roboterarms die bestmögliche Auswahl, um in allen Bereichen mich zu fördern und zu testen.

Die Konstruktion wird also in verschiedene Aspekte gegliedert, die zum Schluss zu einem funktionstüchtigen Roboterarm zusammenkommen:

1. Die technische und mechanische Konstruktion des 6-Achsen Roboterarms (3D-Druck, CAD, etc.)
2. Der Einsatz von elektronischen Teilen, wie Motoren und Sensoren, um dem Roboterarm zu erlauben sich zu bewegen und zu wissen, wo es sich im Raum befinden könnte
3. Programmierung von verschiedenen Skripten, um die Bewegung und Kontrolle des Roboterarms zu verwirklichen.
4. Recherche

Das Ziel dieses Projektes ist somit, einen funktionstüchtigen Roboterarm zu bauen. Mit dem Wort «funktionstüchtig» meint man somit:

* Ein komplett zusammengebauter Roboterarm mit 6 Gliedern (Achsen/Motoren), welche unabhängig voneinander bewegt werden können
* Kontrolle über den Roboterarm per Interface, das positionelle Parameter annehmen kann, welche durch den Roboterarm umgesetzt werden
* Funktionstüchtige Arbeitsweise des «Inverse Kinematics» mit dem Roboterarm

# Theoretischer Hintergrund

Hinter einem 6-Achsen Roboterarm liegen viele verschiedene Themen, die man mit dem ersten Blick nicht leicht erkennen kann. Es versteckt sich jedoch sehr viel Physik und Mathematik dahinter, damit man mit einem Roboterarm z.B. von einem Punkt zum anderen sich bewegen kann.

## Das Arbeitsprinzip eines 6-Achsen Roboterarms

Lorem Ipsum

# Materialien und Methoden

# Elektrische Teile

### Schrittmotoren

Schrittmotoren sind auch Motoren, die jedoch nicht kontinuierlich drehen, sondern Schritt nach Schritt bewegt werden. Solche Schrittmotoren werden vor allem in Maschinen gebraucht, in dem man bestimmte Positionen erreichen kann. Am meisten werden Schrittmotoren in CNC-Maschinen (Computer Numerical Control) sowie auch 3D-Drucker benützt.

A circular object with a red gear

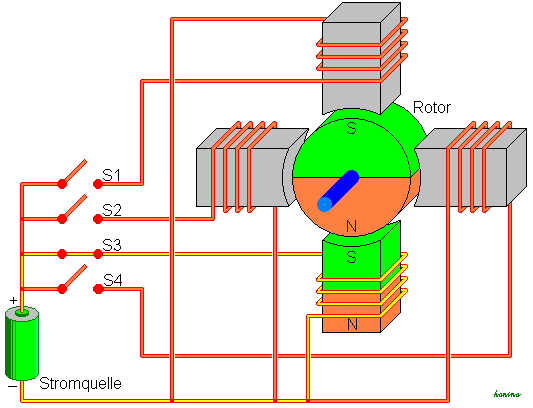
AI-generated content may be incorrect.

Abbildung 1: Arbeitweise eines Schrittmotors ([link](https://upload.wikimedia.org/wikipedia/commons/6/67/StepperMotor.gif))

Abbildung 2: Beschaltung eines Schrittmotoren ([link](https://upload.wikimedia.org/wikipedia/commons/7/74/Schrittmotor.PNG))

Durch das Schrittmotoren Schritt nach Schritt bewegt werden, kann man hingegen zu kontinuierlich drehenden Motoren die relative Position wissen. Abhängig davon welche Grösse und was für ein Drehmoment man braucht hat es viele verschiedene Arten von Schrittmotoren.

A small black and silver machine

AI-generated content may be incorrect.A black and silver electric motor

AI-generated content may be incorrect.Für diese Arbeit wurden NEMA-17 Schrittmotoren benützt, da diese am häufigsten für viele verschiedene Projekte benützt werden. NEMA-17 Schrittmotoren sind gut, da sie ein gutes Drehmoment für deren Grösse haben. Für den Roboterarm wurden 3 verschieden grosse Schrittmotoren verwendet:

A small black and silver device with wires

AI-generated content may be incorrect.

Abbildung 3: NEMA-17 60mm  
([link](https://www.omc-stepperonline.com/image/cache/catalog/stepper-motor/17/17HE24-2104S1-1-500x500.jpg))

Abbildung 4: NEMA-17 48mm  
([link](https://www.omc-stepperonline.com/image/cache/catalog/stepper-motor/17/17HE19-2004S-500x500.jpg))

Abbildung 5: NEMA-17 25mm  
([link](https://www.omc-stepperonline.com/image/cache/catalog/stepper-motor/17/17HS10-0704S-500x500.jpg))

Der Einsatz von verschieden grossen Schrittmotoren war entscheidend, da jedes Glied einen anderen Anspruch an Drehmoment hat. Glieder eher zum Ende der Roboterarms, welche auch weniger Drehmoment brauchen haben kleinere/kürzere Motoren, während Glieder, die fast den ganzen Roboter auf sich tragen und ein grosses Drehmoment benötigen haben grössere/längere Motoren. Für einen grösseren Drehmoment werden dann aber Getriebegehäuse benützt.

Manchmal braucht es aber eine Mischung aus hoher Geschwindigkeit und hohen Drehmoments. Jedoch erhöht sich die maximale Geschwindigkeit nicht zusammen mit der Länge des Schrittmotors und somit auch deren Drehmoment. Jeder Schrittmotor hat eine andere «Speed-Torque» Kurve, welche entscheidet, wie schnell ein Schrittmotor sich drehen kann und unter welchem Drehmoment es bei einer bestimmten Drehgeschwindigkeit ist. Zum Vergleich werden unten zwei «Speed-Torque-Curve» Graphen aufgezeigt.

A graph with a line going up

AI-generated content may be incorrect.

Abbildung 6: "Torque-Curve" von einem 48mm langem NEMA-17 Schritt-motoren ([link](https://www.omc-stepperonline.com/image/catalog/torque-curves/17HE19-2004S_Torque_Curve.svg))

A graph with a line going up

AI-generated content may be incorrect.

Abbildung 7:"Torque-Curve" von einem 60mm langem NEMA-17 Schritt-motoren ([link](https://www.omc-stepperonline.com/image/catalog/torque-curves/17HE24-2104S_Torque_Curve.svg))

Wie man anhand dieses Graphen sieht, ist der Zusammenhang von Drehgeschwindigkeit und Drehmoment entscheidend. Das Drehmoment wird niedriger mit der Drehgeschwindigkeit und die maximale Drehgeschwindigkeit ist auch abhängig von der Länge und des Maximalen Drehmoments des Schrittmotors.

### Mikrokontroller (ESP32)

Unser Körper braucht ein Gehirn, um zu funktionieren. Somit braucht jedes System ihr eigenes Gehirn. Meistens werden dedizierende Computer erschaffen, die dazu dienen, ein System zu kontrollieren. Manchmal braucht man aber nicht immer einen riesigen Computer, um alles zu kontrollieren, sondern etwas Kleineres.

Mikrokontroller sind somit die beste Auswahl für kleinere Projekte, die nicht zu komplexe Operationen durchführen sollte, wie z.B. nach Kommando ein Licht anschalten sollte. Somit kommt ein Mikrokontroller ins Spiel, das solche Operationen leicht durchführen kann. Das Gute and Mikrokontrollen ist, dass diese leicht zu programmieren sind und man diese auch leicht mit weiterer Elektronik, wie Sensoren oder Motoren, verbunden werden kann. Deshalb findet man heutzutage fast wie immer einen Mikrokontroller in DIY-Projekten, da diese sehr einfach zu benützen sind.

Arduinos sind eine beliebte Reihe an Mikrokontrollern und wird heutzutage immer noch weit eingesetzt. Deren Benützerfreundlichkeit für Einsteigende machte sie noch beliebter unter Kinder und Erwachsenen. Selbst wenn Arduinos sehr beliebt sind, würde es für mein Projekt nicht reichen, da man für diesen Roboterarm ein viel mehr I/O-intensiveren (Input/Output) Mikrokontroller bräuchte, der eine grössere Speicherkapazität und einen besseren Prozessor bräuchte.

Die ESP32-Reihe wurde zur Anlaufstelle für ansprechendere DIY-Projekte, mit eingebauten Funktionalitäten wie WiFi oder auch Bluetooth. Deren schnellere Prozessoren und grössere Anzahl von I/O-Ports war entscheidend für diese Arbeit, da die Kontrolle der Motoren viele Outputs bräuchte. Schlussendlich wurde der **ESP32-S3-DevKitM-1** zur besten Auswahl, mit einer genügenden Anzahl von I/O, einer gross genügenden Speicherkapazität und eines schnell genügenden Prozessors.

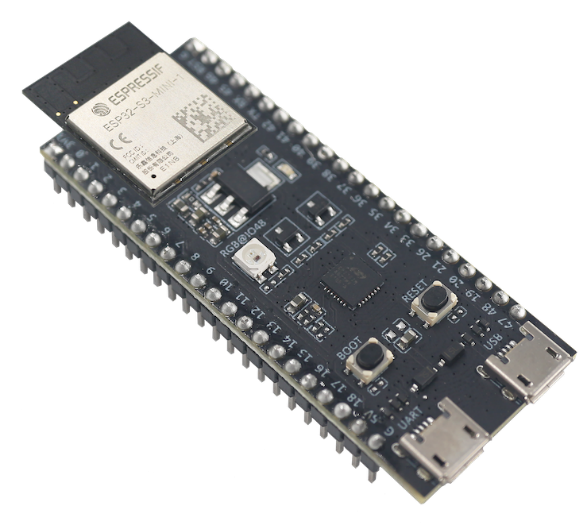


Abbildung 8: ESP32-S3-DevKitM-1 ([link](https://docs.espressif.com/projects/esp-idf/en/v5.2/esp32s3/_images/esp32-s3-devkitm-1-v1-isometric.png))

### Schrittmotor-Treiber (TMC5160)

Da Schrittmotoren aus 4 verschiedenen Motorspulen bestehen und mit Strom versorgt werden müssen, reicht der Strom aus dem Mikrokontroller nicht. Zum Vergleich: Da ein Mikrokontroller meistens durch ein USB-Kabel verbunden wird, werden etwa 500mA bei 5V geliefert. Das sind 2.5W das über ein USB-Kabel geliefert werden. Währenddessen benötigt ein Schrittmotor etwa 800mA (maximum 1500mA) bei 24V, was sich zu 19.2W ergibt (maximal 36W). Es ist also um ein Vielfaches mehr, von was benötigt wird. Deshalb werden Treiber eingesetzt, die diese Leistungsregulierung selbst durchführen können und die Schrittmotoren treiben können.

In diesem Projekt wurden die **TMC5160** Treiber verwendet, welche bis zu 10A Spulenstrom leisten können. Dazu sind die auch sehr praktisch, da sie einen IC (Integrated Circuit) haben, das die ganzen Schrittphasen selber kontrolliert und somit nur durch zwei Inputs kontrolliert werden können (**STEP**: Schritt, **DIR**: Drehrichtung) und einem weiteren Input, mit dem man den Treiber aktivieren oder deaktivieren kann (**EN**: Aktivieren).

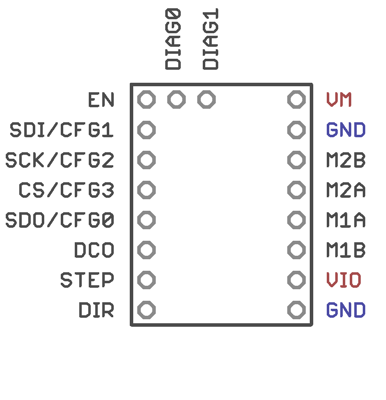


Abbildung 9: TMC5160 SilentStepStick  
([link](https://mm.digikey.com/Volume0/opasdata/d220001/medias/images/695/MFG_TMC5160-SILENTSTEPSTICK.jpg))

Abbildung 10: Pinbelegung des TMC5160 SilentStepStick ([link](https://learn.watterott.com/silentstepstick/pinconfig/tmc5160/tmc5160_v15.png))

Da der TMC5160 über viele verschiedene Parameter verfügt, muss man diese Parameter auf bestimmte Weise einstellen. Doch wenn 6 verschiedene Treiber mit einem Mikrokontroller verbunden wären, bräuchte man für jeden Treiber mindestens ein paar I/O-Pins um diese Parameter einstellen zu können, was einen riesigen Chaos geben würde. Deshalb verfügen über eine SPI-Funktionalität (Serial Peripheral Interface). Diese SPI erlaubt es alle Treiber auf einmal zu verbinden. Damit man aber mit einem Treiber verbinden möchte, wählt man diese aus und kommuniziert damit. Man hat somit insgesamt 4 Pins die zum Treiber verbunden werden. **SDI** (Slave Data In) für die Datenübergabe, **SDO** (Slave Data Out) für die Datenerhaltung, **CS** (Chip Select) für die Auswahl des Treibers und **SCK** (Serial Clock), welches für die Synchronisierung eines Datentransfers benötigt wird. Dieses SPI funktioniert nach der Master-Slave Architektur, der Master die Befehle ergibt und der Slave die Daten verarbeitet und darauf reagiert. undefined([link](https://en.wikipedia.org/wiki/Serial_Peripheral_Interface#/media/File:SPI_single_slave.svg))

Die Parameter, die eingestellt werden können, sind z.B. die maximale Leistung oder auch die Auflösung des Microsteppings. Man kann auch nach spezifischen Daten anfragen, wie z.B., ob der Schrittmotor gehindert wird sich zu drehen.

## Mechanische Teile

### Getriebegehäuse (Planetengetriebe)

Wo das reine Drehmoment eines Schrittmotors nicht genügt, kann ein Getriebe eingesetzt werden. Ein Getriebe dient dazu, den Drehmoment zu erhöhen, durch das es die Drehgeschwindigkeit erniedrigt. Es gibt verschiedene Arten von Getriebegehäuse, doch in dieser Arbeit wurde hauptsächlich das Planetengetriebe (Umlaufgetriebe) eingesetzt.

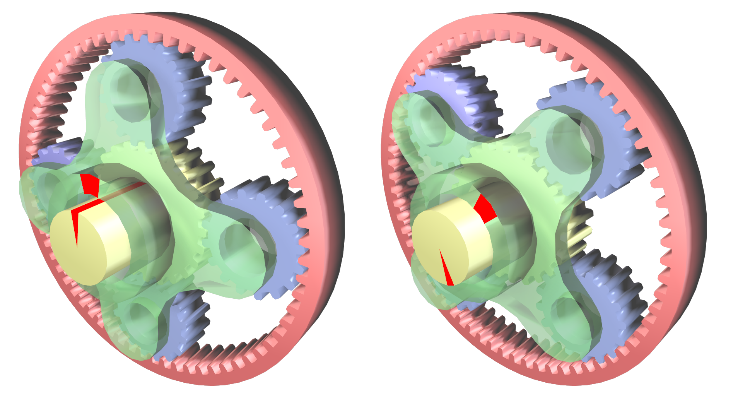


Abbildung 11: Planetengetriebe ([link](https://smcgears.com/wp-content/uploads/2020/10/planetary-reducer.jpg))

Abbildung 12: Rotation eines Planetengetriebes ([link](https://en.wikipedia.org/wiki/Epicyclic_gearing#/media/File:Epicyclic_gear_ratios.png))

A close-up of a mechanical device

AI-generated content may be incorrect.Planetengetriebe sind besser als normale Getriebe, da sie innerhalb eines kleineren Bauraums eine grössere Drehmomentdichte haben und somit leichter einsetzbar und stabiler sind. Planetengetriebe waren somit eine perfekte Auswahl für diesen Roboterarm, womit in einem kompakten Bauraum ein grösseres Drehmoment erreicht werden konnte. Folgend sind ein paar Abbildungen der benützten Planetengetriebe:

A close-up of a mechanical device

AI-generated content may be incorrect.

Abbildung 13: Planetengetriebe mit 10:1 Reduktion ([link](https://www.omc-stepperonline.com/image/cache/catalog/image/catalog/gearbox/EG17-G10-500x500.jpg))

Abbildung 14: Planetengetriebe mit 20:1 Reduktion ([link](https://www.omc-stepperonline.com/image/cache/catalog/image/catalog/gearbox/EG17-G20-500x500.jpg))

Mit Hilfe dieser Getriebegehäuse konnte ich ein höheres Drehmoment mit den Schrittmotoren erreichen, anstatt einen grösseren Schrittmotor zu benützen.

### Kugellager

Damit die ganze Gewichtkraft nicht nur auf die Motore versetzt wird werden Kugellager eingesetzt. Diese erlauben es das Gewicht aller Glieder auf eine grössere Fläche Verteilen zu können und weniger Kraft auf die gedruckten Teile zu versetzen, was diese zum Biegen führen würde. Dazu sind Kugellager auch sehr praktisch vor allem für Rotationsbewegungen, da es erlaubt, zwei verschiedenen Teilen, die zusammen verbunden sind mit weniger Reibung zu drehen.

In diesem Projekt werden Kugellager auch als Riemenspanner benützt, da es sehr nützlich ist, wenn man Teile zusammenbaut und es unmöglich ist, auf traditionelle Weise ein Riemen zu Spannen.

A close-up of a ball bearing

AI-generated content may be incorrect.

Abbildung 15: Miniatur Kugellager 3x9x5mm  
([link](https://www.kugellager-express.de/media/image/product/5022/lg/miniatur-kugellager-603-zz-3x9x5-mm.webp))

Abbildung 16: Rillenkugellager 35x62x14 mm  
([link](https://www.kugellager-express.de/media/image/product/2945/lg/rillenkugellager-6007-2rs-35x62x14-mm.webp))

## Weitere Werkzeuge/Software

### 3D-Drucker

Während dieses Projektes war der Einsatz eines 3D-Druckes obligatorisch. Ohne 3D-Drucker müsste man sich entweder riesige Blöcke von Holz oder auf Teile aus Metall verlassen, was Zeit und Geld braucht, wenn man rasch einen Prototyp braucht, um etwas auszutesten.

3D-Drucker sind somit die beste Wahl um zu Prototypisieren, da es meisten nur Stunden dauert, bis man ein Teil ausgedruckt bekommt. Dadurch ist ein 3D-Drucker ein Muss für jeden, der etwas konstruiert, sei es etwas Kleines (ein Gehäuse für Mikrokontroller) oder etwas Grösseres, wie in diesem Fall einen Roboterarm.

Für meinen Projekt benützte ich einen **Prusa MK4S**, den ich schon seit einer längeren Zeit hab und letztens auf einen **Prusa CORE ONE** mit einem Upgrade durchgeführt habe.

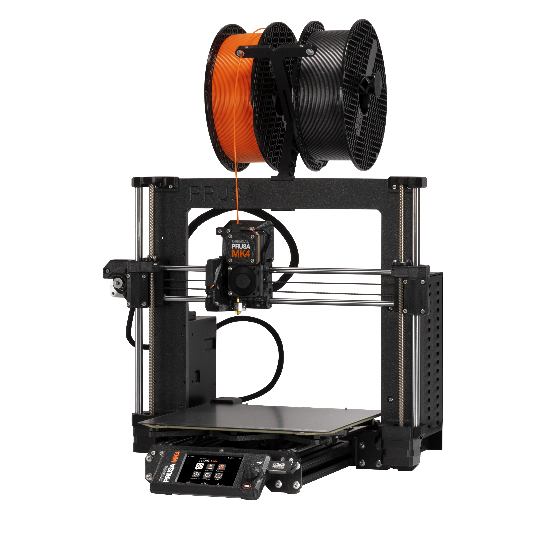
[](https://raps.se/wp-content/uploads/2024/05/MK4_black_version_transparent.png)

Abbildung 2: Prusa CORE One ([link](https://kulturbuero.ch/files-sg/_products/CoreOne-Kopie.png))

Abbildung 1: Prusa MK4S ([link](https://raps.se/wp-content/uploads/2024/05/MK4_black_version_transparent.png))

Durch den Einsatz eines Druckers konnte man schnell und einfach durch verschiedene Prototypen iterieren, was sehr hilfreich war. Ohne Einsatz eines 3D-Druckers müsste man sonst immer online gedruckte oder CNC-Teile bestellen, welche auch Zeit bräuchten. Die Versandzeit wäre dadurch auch ein riesiger Zeitfaktor.

BILD MIT VERSCHIEDENEN ITERATIONEN EINSETZEN

Das Material, das für die Teile benützt worden ist, nennt man PLA (Polyactic Acid) und ist das meistbenutzte Filament auf dem Markt, da es einfach zu drucken und zu verarbeiten ist. Jedoch ist PLA nicht sehr hitzewiderständig, was entscheidend ist, wenn hitzeerzeugende Teile, wie die Motoren, verbunden werden. Deshalb werden manchmal Teile aus PETG (Polyethylene Terephthalate Glycol) oder aus PC (Polycarbonate, reines Polycarbonate kann aber nicht ausgedruckt werden und wird deshalb mit weiteren Substanzen zusammengemischt) ausgedruckt.

Fusion 360 und Solidworks

Um eine digital-visuelle Konstruktion eines Roboterarms zu ermöglichen, benötigte es eine CAD-Software (Computer-aided design). **Fusion 360** und **Solidworks** sind solche Softwares, die dies ermöglichten. Zum Anfang der Arbeit hatte ich Fusion 360 benützt, aufgrund der Familiarität des Programms und weil ich es schon seit einer längeren Zeit benützt habe. Mit Fusion 360 hatte ich meinen ersten «Prototyp» erstellen können. Die Funktionalität, um verschiedene Teile mit Gelenken zu artikulieren war sehr wichtig, da ich die Bewegungen digital schon simulieren konnte.

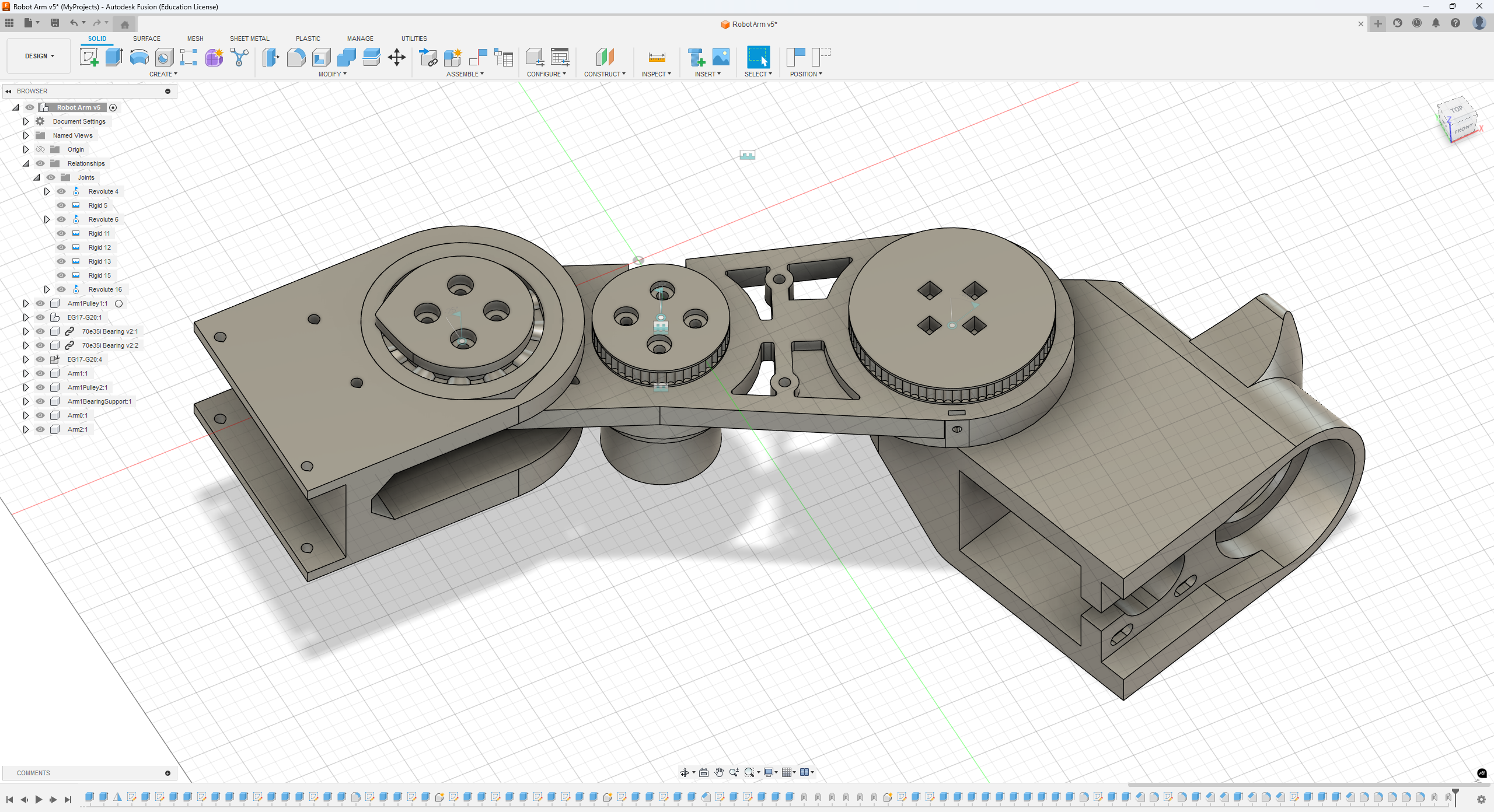


Abbildung 17: Erster "Prototyp" dargestellt auf Fusion 360

Solidworks ist auch eine CAD-Software, die nach meiner Erfahrung viel optimierter und viel benützerfreundlicher ist als Fusion 360, mit vielen Funktionalitäten wie in Fusion 360, aber auch viel mehr. Mit Solidworks konnte ich Teile viel schneller herstellen und viel besser optimieren.



Abbildung 18: Das Endprodukt dargestellt in Solidworks

### Visual Studio Code

A screen shot of a computer

AI-generated content may be incorrect.Visual Studio Code (auch als VSC bezeichnet) ist eine weitbekannte Software, die viele verschiedene Sprachen unterstützt. In dieser Arbeit wurde VSC grundsätzlich für das Programm im Mikrokontroller benützt. Durch die Hilfe eines Plugins **PlatformIO** wird das Hochladen des von mir erstellten Programms auf einen Mikrokontoller leicht gemacht. **PlatformIO** unterstütztviele Mikrokontoller und hat viele verschiedene Funktionalitäten und wird konstant von einer wachsenden Community unterstützt, mit Bibliotheken für Sensoren, Funktionalitäten wie WiFi und weiterem. VSC ist aber nicht nur für das Programmieren von Mikrokontroller nützlich. Durch deren grosse Auswahl an Plugins kann VSC für jede Applikationsart gebräuchlich sein.

Abbildung 19: Ansicht vom Script in VSC

### PyCharm

A computer screen shot of a program

AI-generated content may be incorrect.PyCharm ist von ein von **JetBrains** entwickeltes IDE (Integrated Development Environment), was hauptsächlich für die Programmierung von Python benützt werden kann. PyCharm hat viele Funktionalitäten, die für die Entwicklung von Python-Programmen nützlich sind und vieles erleichtert. Durch einen eingebauten und fortgeschrittenen Debugger kann man Fehler in Programmen viel besser und schneller erkennen.

Abbildung 20: Ansicht vom Code in PyCharm

### KiCAD

KiCAD ist auch eine CAD-Software mit der Hauptfunktionalität, elektrische Schemas zu erstellen und daraus PCBs (Printed Circuit Board) zu entwickeln. Durch dass KiCAD Open Source ist, ist KiCAD komplett kostenlos und wird konstant von einer grossen Community unterstützt. Mit KiCAD kann man eigene elektrische Schemas erstellen, diese auch testen und diese in einen PCB-Design umwandeln kann. Diese PCB-Designs kann man schlussendlich an eine PCB-Manufaktur absenden, welche diese herstellen können.

A computer screen shot of a computer

AI-generated content may be incorrect.

Abbildung 21: Ansicht von der PCB in KiCad

# Praktische Umsetzung

Lorem Ipsum

# Resultate

Lorem Ipsum

# Quellenverzeichnis

Lorem Ipsum