



**HÖHERE TECHNISCHE BUNDESLEHRANSTALT**

**KLAGENFURT, MÖSSINGERSTRASSE**

**ABTEILUNG ELEKTROTECHNIK**

---

**DIPLOMARBEIT**

**Titel der Diplomarbeit Deutsch**

**Automated-Factory-Storage-System**

**JAHRGANG 5AHET**

**eingereicht von**                   **Benedikt Simbürger**

**Vincent Sonvilla**

**Nikolaj Voglauer**

**Elena Widmann**

**Projektbetreuer**                   **Dipl.-Ing. Christian Sallinger**

Diese Diplomarbeit entspricht den Standards gemäß dem Leitfaden zur Umsetzung der Reife- und Diplomprüfung des BMBWF in der letztgültigen Fassung.

Klagenfurt, am 04.04.2025



## EIDESSTATTLICHE ERKLÄRUNG

Ich versichere an Eides statt, dass ich diese Diplomarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe. Alle Gedanken, die im Wortlaut oder in grundlegenden Inhalten aus unveröffentlichten Texten oder aus veröffentlichter Literatur übernommen, oder mit künstlicher Intelligenz generiert wurden, sind ordnungsgemäß gekennzeichnet, zitiert und mit genauer Quellenangabe versehen.

Verfasser/Verfasserin

---

Benedikt Simbürger

---

Vincent Sonvilla

---

Nikolaj Voglauer

---

Elena Widmann

Klagenfurt, am 04.04.2025

## Kurzbeschreibung

Beschreiben Sie das Projekt an dieser Stelle mit maximal zwei aussagekräftigen Sätzen, die dem Leser die Projektidee kompakt vermitteln und eine thematische Zuordnung ermöglichen. Die Kurzbeschreibung, im Umfang von einer A4-Seite, umfasst die wesentlichen Aspekte des Projektes in sozialer und technischer Hinsicht. Die Zielgruppe der Kurzbeschreibung sind auch Nicht-Techniker! Diese Beschreibung wird für Wettbewerbe und PR-Aktivitäten verwendet. Diese Inhalte sind unter dem Menüpunkt „About“ bzw. „Beschreibung“ auf der Projekthomepage darzustellen. Auf die Formulierung der Kurzbeschreibung sollte sehr viel Wert gelegt werden, weil viele Leser oft nur diese Seite lesen und den Rest lediglich durchblättern. Erstellen Sie die finale Kurzbeschreibung erst nach der Fertigstellung der Diplomarbeit.

## Aufgabenstellung

- Warum ist die Themenstellung von Interesse?
- Was ist die vorgegebene Zielsetzung?
- Welche Ergebnisse sind zu erreichen?

## Realisierung

- Von welchem Stand der Technik im Umfeld der Aufgabenstellung wurde ausgegangen?
- Welche Lösungsansätze sind grundsätzlich möglich?
- Warum wurde ein bestimmter Lösungsansatz gewählt?
- Welche experimentelle, konstruktive oder softwaretechnische Methodik wurde angewendet?
- Auf welche fachtheoretischen Grundlagen wurde aufgebaut?
- Welche wirtschaftlichen Überlegungen wurden angestellt?

## Ergebnisse

- Worin besteht der konkrete Beitrag zur Lösung der Aufgabenstellung (Prototyp, Entwurfsplanung, Softwareprodukt, Businessplan etc.)?
- Kann das Ergebnis durch eine typische Grafik, ein Diagramm bzw. ein Foto illustriert werden?
- Kann in die Vollversion der Diplomarbeit Einsicht genommen werden?

**Kurztitel:** Kurztitel der Diplomarbeit (etwa. 30 Zeichen)

**Schlüsselwörter:** Maximal fünf aussagekräftige und durch Kommas getrennte Schlüsselwörter, welche die Inhalte möglichst gut beschreiben.

## Abstract

Die Kurzbeschreibung in englischer Sprache sollte sinngemäß exakt der Kurzbeschreibung in deutscher Sprache entsprechen, siehe Abschnitt „Kurzbeschreibung“.

**Short title:**

**Keywords:**

# Inhaltsverzeichnis

<b>1 Allgemeiner Teil</b>	<b>7</b>
1.1 Ausgangssituation . . . . .	7
1.1.1 Anforderungen . . . . .	7
1.2 Potentielle Lösungen . . . . .	8
1.2.1 Lagermethoden . . . . .	8
1.2.2 Steuerung (bitte keine vergangenheitsformen)4/9 . . . . .	8
1.2.3 Schaltschrank . . . . .	9
1.3 Verfolgter Lösungsansatz 3/4 . . . . .	9
1.3.1 Lagermethoden . . . . .	9
1.3.2 Steuerung 1/4 . . . . .	10
1.3.3 Schaltschrank 1/4 . . . . .	10
1.4 Methodik 5 . . . . .	10
1.4.1 Software 2 . . . . .	10
1.4.2 Hardware 2 . . . . .	11
1.4.3 Fertigung 1 . . . . .	13
1.5 Sicherheitstechnik . . . . .	14
1.6 Kooperationspartner 2X weety, 2x vogi, 1x sonne (otos hinzufügen mach i sonst(BS ),first come first serve . . . . .	15
1.6.1 KNAPP (Vogi) . . . . .	15
1.6.2 Weidmüller (Vogi) . . . . .	15
1.6.3 Igus . . . . .	15
1.6.4 LAPP Austria GmbH . . . . .	15
1.6.5 Mädler GmbH . . . . .	15
<b>2 Hardwareentwicklung, Softwarebackend und Benutzeroberfläche (Benedikt Simbürger)</b>	<b>17</b>
2.1 Hardware . . . . .	17
2.1.1 Planung - Grundanforderungen . . . . .	17
2.1.2 Vorgehensweise . . . . .	18
2.1.3 Rahmen . . . . .	19
2.1.4 X-Achse . . . . .	20
2.1.5 Y-Achse . . . . .	22
2.1.6 Z-Achse . . . . .	25
2.1.7 Lager . . . . .	27
2.1.8 Querförderer . . . . .	28
2.1.9 Fertigung der Einzelteile . . . . .	28
2.1.10 Aufbau . . . . .	30
2.2 Software und Benutzeroberfläche . . . . .	32
2.2.1 Grundlegendes . . . . .	32
2.2.2 Aufbau . . . . .	32

2.2.3 Benutzeroberfläche . . . . .	33
2.2.4 Backend . . . . .	35
2.2.5 APIs . . . . .	36
2.2.6 Lageralgorithmus für SPS . . . . .	37
2.2.7 Datenbanken . . . . .	38
2.2.8 Docker . . . . .	39
2.2.9 Artikelsuche . . . . .	40
<b>3 Integration und Programmierung der Steuerungstechnik (Vincent Sonvilla)</b>	<b>44</b>
3.1 Aufgabenstellung . . . . .	44
3.2 Tia-Portal Grundlagen . . . . .	44
3.2.1 Programm erstellen . . . . .	44
3.2.2 Programmbausteine . . . . .	44
3.2.3 Technologieobjekte . . . . .	46
3.2.4 Programmiersprachen . . . . .	46
3.2.5 Librarys . . . . .	46
3.3 Motorenansteuerung . . . . .	46
3.4 SPS-Server Kommunikation . . . . .	46
3.4.1 Zur Auswahl stehende Kommunikationsprotokolle . . . . .	46
3.4.2 Verbindungsherstellung . . . . .	46
3.4.3 Datenfilterung . . . . .	47
3.5 Herausforderungen . . . . .	48
<b>4 Elektroplanung und Realisierung (Nikolaj Voglauer)</b>	<b>49</b>
4.1 Elektroplanung . . . . .	49
4.1.1 Einleitung - Grundanforderungen . . . . .	49
4.1.2 Elektrik spezifische Anforderungen . . . . .	49
4.1.3 Mechanische Planung . . . . .	51
4.2 Realisierung . . . . .	55
<b>5 Sensorik und Sicherheitstechnik (Elena Widmann)</b>	<b>56</b>
5.1 Aufgabenstellung . . . . .	56
5.2 Sensorik . . . . .	56
5.2.1 Endschalter . . . . .	56
5.2.2 Referenztaster . . . . .	57
5.2.3 Lichttaster . . . . .	60
5.2.4 Barcode-Scanner . . . . .	60
5.3 AS-Interface . . . . .	60
5.3.1 Allgemeines . . . . .	60
5.3.2 Programmierung im TIA-Portal . . . . .	60
5.3.3 Verkabelung . . . . .	60
5.4 Sicherheitstechnik . . . . .	60

---

5.4.1	Grundanforderungen und Planung . . . . .	60
5.4.2	Realisierung . . . . .	60
<b>6</b>	<b>Anhang</b>	<b>61</b>
6.1	Testergebnisse . . . . .	61
6.2	Abmessungen . . . . .	61
6.3	Datenblattauszüge . . . . .	61
6.4	Projektmanagement . . . . .	61
6.4.1	Aufgabenstellung des Gesamtprojekts . . . . .	61
6.4.2	Scrum-Projektplan . . . . .	61
6.4.3	Terminplanung . . . . .	63
6.5	Inbetriebnahme . . . . .	64
6.6	Kostenaufstellung . . . . .	65
6.7	Besprechungsprotokolle . . . . .	66
6.8	Arbeitsnachweis . . . . .	70
<b>Literaturverzeichnis</b>		<b>71</b>
<b>Abbildungsverzeichnis</b>		<b>73</b>
<b>Tabellenverzeichnis</b>		<b>74</b>

# 1 Allgemeiner Teil

## 1.1 Ausgangssituation

Die HTL-Mössingerstraße arbeitet im Zuge der Werkstatt mit einer Factory, diese ist eine Miniatur-Firma bei der Lehrpersonal sowie SchülerInnen Bestellungen abgeben können. Diese Bestellungen werden dann von zugeteilten SchülerInnen abgearbeitet.

In dieser Factory laufen die Bestellprozesse digital über eine schulinterne Website. Verwendet werden von der Factory 3D-Drucker, CNC-Fräsen, Lasercutter und diverse weitere Maschinen, um Angefordertes zu produzieren. Die Factory stellt primär Einzelteile zur Verfügung, dazu gehört auch ein großes Repertoire an Bauteilen, die im Lager der Factory gelagert werden.

Die Handhabung dieses Lagers erfolgt bisher manuell. SchülerInnen schreiben in einer digitalen Applikation mit, welche Teile ein- bzw. ausgelagert werden. SchülerInnen sind im Schnitt nur ein Schuljahr in der Factory. In dieser Zeitspanne ist es kaum möglich eine Routine einzuarbeiten und die Fehlerquote bei der Arbeit im Lager ist relativ hoch. Das führt zu fehlerhaften Lieferungen oder verzögerten Produktionsketten.

Ein weiterer Nachteil des derzeitigen Lagers ist, die Örtlichkeit. Das Lager befindet sich zurzeit im Keller und ist dort fest verbaut. Die schweren Schränke und Regale können im Falle von Hochwasser, wie im Sommer 2023, weder schnell ausgelagert oder verschoben werden.

Im Allgemeinen Schulgeschehen ergeben sich zudem Möglichkeiten der Migration fürs Lager, um Prozesswege der Factory effizienter zu gestalten. Mit dem derzeitigen Lager können keine potenziellen Optionen wahrgenommen werden.

### 1.1.1 Anforderungen

Um die Automatisierung dieses Prozesses zu ermöglichen, soll ein System entwickelt und gebaut werden, das eine lagernde Box automatisch zu einer Kommissionierstation bringt. An dieser Station soll die Möglichkeit bestehen, Inventar anzufordern, Lagerbestände auszugeben oder aufzufüllen sowie Boxen wieder einzulagern.

Weiterhin muss dieses System erweiterbar sein, um zukünftig zusätzlichen Lagerplatz hinzuzufügen und die Integration anderer Systeme, wie beispielsweise einer Bauteilvereinzelung, zu ermöglichen.

Zudem soll das AFSS mobil sein – sowohl mechanisch als auch elektrisch –, da es nicht am endgültigen Standort errichtet wird und einen einfachen Transport bei möglichen Umbauarbeiten ermöglichen soll.

Die Rahmenbedingungen für die Umsetzung dieser Ziele sind stark davon geprägt, dass alle mechanischen Bauteile, die eigens gefertigt werden müssen, so konstruiert werden, dass ihre Herstellung mit den Mitteln der HTL möglich ist.

Das System soll nicht nur produzierbar, sondern auch reproduzierbar sein. Daher muss die

---

Dokumentation der Funktion sowie des Umsetzungsprozesses so erfolgen, dass das Projekt von nachfolgenden Schüler\*innen weitergeführt werden kann.

## 1.2 Potentielle Lösungen

### 1.2.1 Lagermethoden

Die Industrie bietet viele Vorbilder dafür, wie ein boxenbasiertes Lagersystem aufgebaut sein kann.

Eine besonders platzeffiziente Variante ist beispielsweise die PickEngine von KNAPP [1]. Bei dieser Lösung werden Boxen in mehreren Ebenen übereinander gelagert. Auf jeder Ebene gibt es bewegliche Roboter, die die Boxen abholen und zu einem Lift bringen, von dem aus die Box dann zur Kommissionierstation transportiert wird. Dieses System ist sowohl platzsparend als auch sehr ausfallsicher. Allerdings ist es schwierig, ein solches System in Miniatur nachzubauen, da die Hardwarefertigung sehr komplex ist.

Eine weitere Möglichkeit wäre ein rotierendes Magazin, in dem die Boxen auf einem horizontalen Karussell gelagert sind. Wenn ein bestimmtes Produkt benötigt wird, rotiert das Karussell so lange weiter, bis die gewünschte Box zugänglich ist. Das Prinzip dieses Systems ist zwar recht simpel, jedoch stellt die mechanische Umsetzung der Rotation mit den Mitteln der HTL eine Herausforderung dar, insbesondere im Hinblick auf einen zuverlässigen Dauerbetrieb.

Eine dritte Option besteht darin, die Ware vertikal von oben zu lagern. Über der Lagerstätte bewegt sich ein Roboter, der die Boxen ausheben, umschichten sowie ein- und auslagern kann. Dieses System ermöglicht eine hohe Lagerdichte, hat jedoch den Nachteil eines begrenzten Durchsatzes. Wenn eine Box benötigt wird, die nicht an oberster Stelle liegt, müssen erst alle darüberliegenden Boxen umgeschichtet werden. Zudem ist diese Lagervariante erst dann wirklich platzsparend, wenn sie sehr hoch gebaut wird. Bei wenigen Ebenen lohnt sie sich noch nicht, da trotz geringer Höhe bereits viel Fläche verbraucht wird.

### 1.2.2 Steuerung (bitte keine vergangenheitsformen)4/9

Es gibt durchaus verschiedenste Methoden zur Steuerung von Anlagen. Bekannterweise werden hier Speicherprogrammierbare Steuerungen, kurz SPS, verwendet. Diese bieten den Vorteil, komplexe Steuerungen durch Programmierung mit intuitiven Programmiersprachen realisieren zu können. Jedoch erfordern SPS für jeden Zusatz, beispielsweise digitale Ein-/Ausgänge oder das Ansteuern von Motoren, verschiedenste Module.

Eine günstigere Variante zu Speicherprogrammierbare Steuerungen bieten Arduinos. Diese benötigen aber für das Ansteuern von energielastigeren Bauelementen auch zusätzliche Elemente. Zusätzlich steigt der Programmier-Aufwand, sowie die Komplexität des **des is schon a bissi hoch gestochen** Programms, bei großen Anlagen ins Unermessliche, da jedweilige Motoransteuerung oder Verbindung mit Servern ausprogrammiert werden muss. Dieser Aufwand wäre in diesem Projekt zeitlich nicht schaffbar.

---

Ein wenig teurer als Arduino, aber noch unter dem Preis einer SPS, liegen Verbindungsprogrammierbare Steuerungen, welche die Funktion dadurch realisieren, wie sie verdrahtet sind, über Relais, Schalter, etc. Diese sind aber mit einem beträchtlichen Aufwand, aufgrund von Verkabelung sowie Anschaffung benötigter Materialien, verbunden. Zusätzlich ist die Realisierung von komplexeren Projekten durchaus zeitintensiv.

### 1.2.3 Schaltschrank

Für die Elektronik gibt es mehrere Möglichkeiten der Umsetzung. Beleuchtet wurden im Entwicklungsprozess integrierte Bauweisen in das AFSS, neuwertige Schaltschränke sowie Umbaumöglichkeiten.

Die erste Idee, die verfolgt wurde, war ein integrierter Schaltschrank im AFSS. Dieser hätte einen Rahmen aus Aluminiumprofilen, Plexiglasscheiben hätten die Komponenten vom Rest getrennt und die einzelnen Elemente wären auf eine gefrästen Aluminiumplatte montiert worden. Vorteil dieser Option ist eine kompakte Bauweise, man hätte keinen zusätzlichen Schrank. Nachteile sind Zusatzkosten, Lagerplatzverluste da das Lager nicht breiter gemacht werden kann und kein Platz für Erweiterungen.

Ein neuer Schaltschrank wäre grundsätzlich die naheliegendste Lösung. Für das AFSS hätte der Schaltschrank groß sein müssen, um alle Komponenten unterzubringen und um die Erweiterbarkeit zu gewährleisten. Vorteilhaft wäre, verminderter Bauaufwand und eine gute el. Sicherheit. Allerdings ist ein herkömmlicher Schaltschrank nicht mobil und ist somit für eine mobile Anlage ungeeignet, zudem entstehen Zusatzkosten.

Die letzte Option war ein alter Serverschrank, dieser kommt mit vier Profilschienen, einer Lüftungsanlage und war groß genug, um die Anforderung der Erweiterbarkeit auch zu erfüllen. Zusätzlich gemacht gehört sind Module, die auf die Schienen montiert werden, auf diesen wären die el. Komponenten montiert. Die Module können aus kosteneffizienten Platten hergestellt werden. Um den Serverschrank mobil zu machen, müssten kleinere Umbauten durchgeführt werden.

## 1.3 Verfolgter Lösungsansatz 3/4

### 1.3.1 Lagermethoden

Nach Abwägung der Alternativen wurde bei der Auswahl der Lagermethode ein klassisches Palettenlager als Inspiration gewählt. Der Grundgedanke basiert auf einem Portalsystem, das mit einer Gabel Boxen in einem Regal ein- und aushebt. Diese Lagervariante vereint die Balance aus technischer Komplexität und dem Umsetzungsvermögen an der HTL. Zudem gibt es bei dieser Variante ebenfalls Potenzial zur Effizienzsteigerung, da die Möglichkeit besteht, links und rechts des Roboters Bestand zu lagern. Allerdings wird diese Variante nicht forciert, da die Komplexität eines solchen Mechanismus im kleinen Maßstab und mit eingeschränkter Fertigungstechnik recht schwierig umzusetzen ist.

Es wurde also ein Lagersystem gewählt, bei dem zwei Achsen hin- und herfahren und eine

Art Gabel die Boxen ein- und aushebt. Um die Boxen zur Kommissionierstation zu bringen, wird ein Förderband verwendet, das längs zum Lager verläuft. Der Lagerroboter kann die Boxen jedoch nicht selbstständig auf das Förderband laden. Zu diesem Zweck wird ein sogenannter Querförderer eingesetzt, der die Box von einem temporären Lagerplatz auf das Förderband und wieder zurück schiebt. An dieses Förderband können außerdem weitere solcher Lagerschränke angeschlossen werden, um mehr Lagerplatz zu schaffen und auch andere Systeme anzubinden. Darüber hinaus wird ein Lagerschrank als Komplettsystem konzipiert. Durch die Unterbringung aller Systeme in einem einzigen Objekt kann die Transportfähigkeit durch die Montage von Rollen einfach sichergestellt werden.

### 1.3.2 Steuerung 1/4

Als Steuerung wurde eine SPS von Siemens ausgewählt, da das Basic Know-How für die Programmierung solcher, bereits in der Schule gelehrt wird. Zudem sind speicherprogrammierbare Steuerungen ideal für komplexere Anlagen und ermöglichen die Ansteuerung der Motoren sowie die Kommunikation mit dem Server relativ simpel und bleiben dabei auch innerhalb der geforderten Zykluszeit.

### 1.3.3 Schaltschrank 1/4

Der Umbau vom Serverschrank war die beste Option. Es wird Altbestand verwertet, es ist eine kosteneffiziente Lösung die viel Freiheit zur Anpassung an spezielle Komponenten erlaubt. Zudem gibt es viel Platz für Erweiterungen und Räder an den Schrank zu montieren wäre kein großer Zusatzaufwand. Die elektrische Steuerung der Anlage, die Versorgungsgeräte und weiter Elemente finden in diesem Schrank auch platz. Weiters kann mit dieser Option auch erreicht werden, dass der Schaltschrank mittels weniger Handgriffe vollkommen vom Lagerschrank getrennt werden kann. Eine gute Lösung für die Ansprüche.

## 1.4 Methodik 5

### 1.4.1 Software 2

Zur Dokumentation der elektrischen Komponenten und deren Verbindungen wird E-Plan verwendet. E-Plan ist der Industriestandard und bietet unzählige Möglichkeiten für eine ausführliche Dokumentation. E-Plan hilft ein übersichtliches Schaltbild zu erstellen, und die Anlage lesbar festzuhalten. Auch im Planungsprozess ist E-Plan der Schlüssel zu einer vollständigen Planung.

### Fusion360/PCB

Zur mechanischen Planung des AFSS wird vorrangig Fusion360 genutzt. Dies ist ein 3D-CAD (Computer-Aided-Design) Programm, welches eine breite Palette von Funktionen bietet, jedoch noch sehr bedienerfreundlich ist. Weiters ist es sehr nützlich, dass Fusion360

---

eine integrierte Cloud-Speicherung anbietet. So können die Konstruktionen von anderen Personen einfach eingesehen werden.

Die volle Integration von Eagle in Fusion360 im Jahr 2020 macht es möglich, auch in Fusion360 Leiterplatten zu entwerfen und die dazugehörigen Schaltungen zu zeichnen. Der Verkauf des Programms Autodesk Eagle wird mit Juni 2026 eingestellt und ist somit nicht mehr Stand der Technik.[2] Das Entwerfen von PCBs (Printed Circuit Boards) ist durch die Integration in Fusion360 nicht nur einfacher, sondern auch einheitlicher geworden. Außerdem kann durch die Zusammenführung der beiden Programme ermöglicht werden, dass Projekte auf einer zentralen Entwurfsplattform umgesetzt werden.

### **AutoCAD**

Zur mechanischen Planung des AFSS wird auch AutoCAD verwendet. Vor allem für Pläne von Frästeilen oder Laserteilen bietet sich diese Applikation an. AutoCAD wird in der Industrie verwendet und bietet eine benutzerfreundliche Oberfläche sowie eine breite Palette an Funktionen.

### **Lagerverwaltungssoftware**

Um Lagerbestand zu verwalten bedarf es eines ausgeklügelten Systems, welches einerseits eine Benutzerbedienung zulässt, und andererseits die Logik der Auslagerung übernimmt. Eine Möglichkeit dies zu erreichen, wäre eine reine Softwarelösung nur auf SPS-Basis. Dies ist aber sehr sehr umständlich und auch nicht skalierbar. Auch die Verwendung von graphischen Entwicklungswerkzeugen wie Node-Red wäre denkbar, jedoch ist es auch hier schwierig komplexe Operationen durchzuführen, wenngleich die Einbindung zur SPS einfacher ist.

Die Favorisierte Lösungsvariante für diese Anforderungen ist die Entwicklung eines eigenen Servers in einer Programmierhochsprache. Die SPS soll mit diesem kommunizieren können. Außerdem kann dieser Server eine Benutzeroberfläche für die Bedienung zur Verfügung stellen, sowie die Steuerungslogik übernehmen. Dieser soll überdies auf eine Datenbank zugreifen, welche den Lagerbestand abbildet.

### **TIA Portal**

Ist die Software, welche verwendet wird, um speicherprogrammierbare Steuerungen von Siemens anzusteuern. TIA-Portal bietet die Möglichkeit, in verschiedenen Programmiersprachen zu arbeiten (Funktionsplan (FUP), Kontaktplan (KOP), einer Anweisungsliste (AWL), mit Structered Code Language (SCL) oder Graph.) Zudem können externe Anwendungs-spezifische Funktionsbausteine mit Bibliotheken eingebunden werden. [3]

#### **1.4.2 Hardware 2**

## Antriebe

Das elektrische Antriebssystem ist ein Kernbaustein des AFSS. Mit diesem werden alle beweglichen Aktoren betätigt. Hier ist es nötig, einen Kompromiss zwischen Leistung, Kosten und der Ansteuerbarkeit zu finden.

Industrieservomotoren, die typischerweise mit 400 V, Frequenzumrichtern und Gebern betrieben werden, sind zwar sehr leistungsreich und gut steuerbar, jedoch auch sehr teuer. Aus diesem Grund wurde schnell ein Antriebssystem aus Closed-Loop- und Open-Loop-Schrittmotoren angestrebt.

Für alle Anwendungen mit hohen Drehmomenten (X- und Y-Achse) sollen Closed-Loop-Schrittmotoren im Formfaktor Nema23 verwendet werden. Diese können im Falle unerwarteter Laständerungen ohne Schrittverlust weiterbetrieben werden und sind zudem in den benötigten Drehmomentbereichen preiswert verfügbar. Jedoch müssen für diese Motoren Treiber eingesetzt werden. Diese verarbeiten die Signale der SPS und treiben den Motor an. Zudem verarbeiten sie auch die Gebersignale. Falls eine Diskrepanz zwischen Gebersignal und Steuersignal erkannt wird, wird dies automatisch ausgeglichen.

Für die Z-Achse (Gabel) sowie für den Querförderer ist lediglich ein Open-Loop (geberloser) Antrieb nötig. Hier werden Nema17 Motoren eingesetzt. Bei diesen Achsen ist es relativ einfach, bei jedem Verfahren die tatsächliche aktuelle Position mit der vermuteten Position zu vergleichen, da jedes Mal über den Referenzierpunkt gefahren wird.

## Sensoren

Die Sensoren sind wesentlich für eine sichere und korrekte Funktion des AFSS. Sie werden zur Übermittlung der genauen Positionierung der Achsen eingesetzt, aber auch zum Überprüfen, ob die jeweiligen Boxen an den Zielpositionen angekommen sind.

Zur Übertragung des Sensorsignals wird ein AS-i-Bus verwendet. Dieser bietet eine einfache Kommunikation zwischen den Sensoren und der SPS. Die Sensoren werden über den AS-i-Slave mit 24V versorgt, jedoch ist zu beachten, dass einzelne Sensoren mit einer niedrigeren Spannung betrieben werden müssen, um eine lange Lebensdauer und korrekte Funktion garantieren zu können.

Ohne mechanische Endschalter würde die Gefahr bestehen, dass die Motoren nicht gestoppt werden und der Rahmen angefahren wird, was zu erheblichen Schäden am System führen kann. Aus Verfügbarkeitsgründen werden an den verschiedenen Achsen unterschiedliche Arten von Sensoren als mechanische Endschalter eingesetzt. Die Achsen werden mittels eines Photo-Interrupters, der auf einer eigens entworfenen Platine untergebracht ist, referenziert. Zum Überprüfen, ob die jeweilige Box ihre Zielposition erreicht hat, werden Lichttaster eingebaut. Zur Zuordnung der Boxen und den sich darin befindenden Bauteile werden auf den Boxen Barcodes angebracht. Um diese einlesen zu können befindet sich auf der Kommissionierstation ein Barcodescanner, der die jeweilige EAN-Nummer an die SPS weitergibt, die diese wiederum an den Server weiterleitet.

## Elektrik

Die elektrische Seite der Anlage besteht unter anderem aus diesen 7 Schrittmotoren, wobei Drei davon schwächer sind. Wenn alle Motoren gleichzeitig anlaufen können Ströme von gut mehr als 20 A entstehen. deswegen werden die Motoren auf zwei 20 A/24 V Gleichspannungs-Netzteile aufgeteilt. Die Logik an der Anlage besteht aus einer Siemens SPS samt Ein- und Ausgangsmodul sowie zwei PTO-Modulen. Der Logikkreis samt SPS sollen getrennt gespeist werden, mit einem 24V/10A Gleichspannungsnetzteil-Netzteil. Die Sensoren auf der Anlage laufen über einen ASI-Bus, der Master ist eine Karte für eine ET200. Das ASI-System muss über ein ASI-Netzteil gespeist werden. Das Fließband wird von einem Asynchronmotor mit 1.1 A Nennstrom betrieben. dieser wird über eine Wendeschützschaltung angesteuert. Die Schütz werden von Relais angesteuert und mit einen Motorschutz wird der Motor (**vor überlast**) geschützt. Die Anlage wird über einen FI und einen Leitungsschutzschalter abgesichert und hat zudem einen Schlüsselschalter und einen dreiphasigen Drehschalter zur manuellen Freigabe.

Die Motoren haben jeweils einen Encoder dabei. Somit müssen ein Motorkabel und ein Kabel für den Encoder verlegt werden. Beide Kabel müssen geschriftet sein.

### 1.4.3 Fertigung 1

#### Lasern

An der HTL-Mössingerstraße ist es den SchülerInnen möglich einen Lasercutter zur Kunststoffverarbeitung zu verwenden. Um ein gewünschtes Teil fertigen zu können, muss die Kontur dieses als .dwg zur Verfügung stehen. Dieses kann dann unter berücksichtigung der Materialstärke aus verschiedenen Farben geschnitten werden.

#### Fräsen

Um Aluminiumteile zu fertigen steht eine 3-Achsen CNC-Fräse zur Verfügung. In dieser ist es möglich die Teile zu fräsen, die aufgrund ihrer hohen mechanischen Beanspruchung nicht aus Kunststoff gefertigt werden können, aus anderen Materialien herzustellen. Um dies zu bewerkstelligen, muss zuerst der G-Code in Filou-NC16 programmiert werden und kann dann in NC-Easy auf der CNC-Fräse ausgeführt werden. Da der Fräser jedoch einen größeren Durchmesser als der Laser hat, muss, wenn Ecken benötigt werden, eine Aussparung größer dem Durchmesser des Fräzers, eingeplant werden.

Als Aluminiumlegierung wird hier die Legierung EN-AW 5754 (AlMg3) verwendet. Diese Legierung aus Aluminium und Magnesium eignet sich sehr gut zum Fräßen und ist in der Lage, die mechanischen Beanspruchungen auszuhalten.

#### Aluminium-Extrusionen schneiden

Um Aluminiumextrusionen abzulängen wird eine eigens dafür ausgelegte Kreissäge verwendet. In dieser ist es möglich einen Anschlag für eine bestimmte Länge einzustellen und dann

---

abzuschneiden. Dadurch ist es möglich in kurzer Zeit viele verschiedene Längen präzise zu-zuschneiden.

### 3D-Drucken

Für die Herstellung von Kunststoffteilen, die im Lasercutter nicht gefertigt werden können, stehen den Schülern mehrere 3D-Drucker zur Verfügung. Die Drucker können mit verschiedenen Materialien drucken, beispielsweise PLA oder ABS. Komplexe Bauteile die nicht über die räumlichen Begrenzungen der Drucker hinausgehen, können so gefertigt werden. Dazu wird ein 3D-Modell im stl.-Format benötigt, welches dann in einem Slicerprogramm in G-Code oder Bg-Code umgewandelt werden kann[4]. Schulintern wird hierzu primär der PrusaSlicer verwendet.

## 1.5 Sicherheitstechnik

Um eine langjährige korrekte Funktion des AFSS sicherstellen zu können, müssen diverse Sicherheitsvorkehrungen getroffen werden. Diese lassen sich unterteilen in die Sicherheit der Personen, die das Lagersystem bedienen, und in die Sicherheit des Systems selbst, vor Beschädigung durch Fremdeinwirkungen oder im eigenen Fehlerfall. Um die Sicherheitstechnik richtig umsetzen zu können, muss diese bereits im Vorhinein durchdacht und gut geplant werden.

Um die Benutzerinnen und Benutzer gegen Verletzungen zu schützen muss das AFSS so positioniert werden, dass diese bei ordnungsgemäßem Gebrauch nicht in Gefahr geraten können. Der Schaltschrank sollte so ausgelegt und abgesichert werden, dass Personen nicht bei normaler Benutzung in Stromkreise geraten oder in Berührung mit unter Spannung stehenden Betriebsmitteln kommen können. Die Sicherheit von Personen hat oberste Priorität und wird über die Sicherheit des Systems gestellt.

Ebenfalls müssen die mechanischen sowie elektrischen Elemente so ausgelegt werden, dass Nutzer beim Normalgebrauch des AFSS diesem keine Schäden zufügen können. In den meisten Fällen sorgen die Sicherheitsvorkehrungen, die bereits für die Sicherheit der bedienenden Personen sorgen, auch gleichzeitig für Schutz gegen Fremdeinwirkung. Dazu gehört die Abgrenzung der empfindlichen Mechanik des Lagers durch bewusste Positionierung, sowie eine verschließbare Türe am Schaltschrank, die eine klare Trennung zwischen Elektrik und Nutzer gewährleistet.

Damit die Motoren des Lagersystems nicht nur im Normalbetrieb zum Stoppen gebracht werden, sondern im Fehlerfalle auch die Grenzen der jeweiligen Achsen nicht überschreiten und so keine Schäden an der Mechanik anrichten können, werden neben den Software-Endschaltern zusätzliche Hardware-Endschalter eingebaut. Zur Sicherung der Motoren soll ein Überstromschutz eingebaut werden. Der Schaltschrank und die sich darin befindenden Elemente und deren Verkabelung sollen so geplant und verbaut werden, dass es zu keinen Kurzschläüssen, Bränden oder anderen Schäden kommen kann. Sollte es dennoch zu einer

Situation kommen, in der das AFSS sofort zu einem Stopp gebracht werden muss, werden zum Anhalten des gesamten Systems leicht sicht- und erreichbare Not-Aus-Taster verbaut.

## **1.6 Kooperationspartner 2X weety, 2x vogi, 1x sonne (otos hinzufügen mach i sonst(BS) ),first come first serve**

### **1.6.1 KNAPP (Vogi)**

### **1.6.2 Weidmüller (Vogi)**

### **1.6.3 Igus**

Die Firma igus ist Hersteller von Hochleistungspolymeren für Bewegungs-Applikationen. Sie entwickeln und produzieren sogenannte motion plastics, Produkte aus schmierfreien Kunststoffen. Dazu zählen unter anderem Energieketten, Kabel, Gleitlager, Gewindetechnik, Roboter sowie intelligente Sensorik. Der Firmenname leitet sich davon her, dass die meisten Produkte im Spritzgussverfahren hergestellt werden.

**igus = Industriespritzguss.** [5]

Igus unterstützt unser Projekt durch die Bereitstellung der benötigten Kabelschleppketten, Linearführungen sowie Spindeln. Zusätzlich wird Online eine Software von igus gestellt, welche gratis zur Verfügung steht, die zur Konfiguration der benötigten Teile genutzt wurde.

### **1.6.4 LAPP Austria GmbH**

LAPP Austria GmbH wurde 1994 gegründet und ist Mitglied der globalen LAPP Gruppe. Ihr Standort in Österreich befindet sich in Linz. LAPP Gruppe ist weltweit führender Anbieter von Kabel-, Stecker- und Verbindlungslösungen. Sie bietet ein umfassendes Sortiment an hochwertigen Produkten und maßgeschneiderten Lösungen, die in den verschiedensten Industriezweigen zum Einsatz kommen.[6]

Um unser Projekt verwirklichen zu können werden wir von LAPP durch die Bereitstellung diverser Schleppkettenkabel unterstützt, welche zur Motor- und Geberverkabelung sowie zur Subbusverteilung verwendet werden. Außerdem haben wir Kabel zur Verdrahtung des Schaltschranks in verschiedenen Farben von LAPP erhalten.

### **1.6.5 Mädler GmbH**

Die Firma Mädler GmbH wurde 1882 gegründet und ist Hersteller und Großhändler von Antriebs- und Maschinenbauelementen. Ihr Sortiment umfasst eine Vielzahl an Produkten und wird stetig erweitert. Zu ihren angebotenen Diensten gehören das Trennen und Schweißen von Zahnriemen sowie das Trennen und Vernieten von Rollenketten auf die Wunschlänge der Kunden. Mädler geht flexibel auf die Kundenwünsche ein, auch Sonderanfertigungen nach Zeichnung gehören zu ihrem Angebot.[7]

Unser Projekt wird von Mädler durch die Bereitstellung von Zahnriemen sowie Zahnscheiben unterstützt.

## 2 Hardwareentwicklung, Softwarebackend und Benutzeroberfläche (Benedikt Simbürger)

### 2.1 Hardware

#### 2.1.1 Planung - Grundanforderungen

Grundlegende Anforderungen zur Planung der AFSS-Mechanik sind Transportfähigkeit, eine möglichst einfache Realisierung mit HTL-Mitteln und möglichst wenige Kompromisse in der Funktion oder Zuverlässigkeit eingehen zu müssen.

Die Anforderung der Transportfähigkeit limitiert die Größe des Lagers auf 2,3 m Länge, um im Lift transportiert werden zu können, und auf 1,9 m Höhe aufgrund der Türhöhe im Keller. Weiters müssen auch noch Rollen an den Rahmen angebracht werden, um das Lager ohne großen Mehraufwand bewegen zu können. Diese Extrahöhe der Räder (ca. 80 mm) limitiert den Rahmen weiter.

Nun soll dieser rund 2,25 m lange und 1,8 m hohe Raum optimal genutzt werden, um eine möglichst große Lagerdichte sicherstellen zu können.

Um eine möglichst gute Erweiterbarkeit sowie eine Fertigung an der Schule zu ermöglichen, sollen für die mechanische Trägerkonstruktion sogenannte Item-Profile verwendet werden.

#### Item

Das Item-Profilsystem ist ein System, welches Aluminium-Extrusionen in verschiedenen Ausführungen sowie viele Verbindungsmöglichkeiten zu sich selbst sowie anderen mechanischen Elementen bietet. Hierbei gibt es eine breite Auswahl an Profilen, von 20x20 mm bis 40x40 mm Querschnitt. Für alle Komponenten mit hoher mechanischer Beanspruchung werden 40x40-Extrusionen verbaut, da diese eine besonders hohe Biegefestigkeit aufweisen. Für Anwendungen mit geringerer Beanspruchung sowie aus Platz- und Gewichtsparitätsnahmen werden 20x20-Extrusionen verwendet. Zur Verbindung zu anderen Bauelementen gibt es die Möglichkeit, sogenannte Nutensteinen mit verschiedenen Gewinden in die Nut einzulegen und dort Platten o. Ä. anzuschrauben. Um Item-Profile untereinander zu verbinden, können Standardverbindungssätze wie in Abb. 2.1 verwendet werden.



Abbildung 2.1: Item Profil mit Standardverbindungsatz, Quelle: [8]

Die Anwendung im AFSS erfordert außerdem recht lange Verfahrwege. Um dies kostengünstig umsetzen zu können, werden V-Slot-Profile verwendet.

## V-Slot

Auch V-Slot-Profile sind Aluminium-Extrusionen. Diese können grundsätzlich auch ähnlich wie Item-Profile mit Nutenstein etc. verwendet werden, sind aber zusätzlich darauf ausgelegt, dass ein V-Wheel in einer Narbe des Profils rollen kann (siehe Abb. 2.2). Diese Profile sind in einer C-Profil-Form erhältlich. Diese sind für die langen Verfahrwege optimal, da einerseits auch breitere und mehr V-Wheels verwendet werden können, sowie einen sehr großen Widerstandsmoment aufweisen.

### 2.1.2 Vorgehensweise

Aufgrund der besonderen und sehr komplexen Anforderungen dieser Mechanik erfolgt die Entwicklung in mehreren Iterationen. Nach Auslegung der Grundparameter wird eine Grundkonstruktion erstellt, um mögliche Lösungsansätze für die jeweiligen Komponenten zu skizzieren. Durch diese grobe Planung können viele Konzepte mit geringerem Zeitaufwand iteriert werden und auch mögliche Missverständnisse o. Ä. frühzeitig aufgeklärt und überarbeitet werden. Weiters werden bei diesem Prozess wichtige Fähigkeiten in der Bedienung der CAD-Software gewonnen und so die Geschwindigkeit der zukünftigen Designiterationen beschleunigt.

Um bestimmte Elemente der Mechanik einzeln zu testen, werden auch mehrere Prototypen gebaut und die gewonnenen Erkenntnisse in die finale Konstruktion miteingebunden.



Abbildung 2.2: V-Slot-Profil mit V-Wheel, Quelle: [9]



(a) X-Achse



(b) Y-Achse



(c) Lagerregal

Abbildung 2.3: Prototypen

## Rahmenbedingungen für die Fertigung

Damit die Fertigung der Mechanik an der HTL-Mössingerstraße in einem realistischen Zeitrahmen möglich ist, müssen gewisse Rahmenbedingungen bei der Planung beachtet werden.

- Für Verbindungen mit geringer bis mittlerer mechanischer Beanspruchung werden Bauteile so geplant, dass diese im Lasercutter gefertigt werden können, sowie in den Stärken, die der Werkstatt zur Verfügung stehen (3 mm, 4 mm und 5 mm). Dabei muss beachtet werden, dass die Acrylplatten gegossen sind und dadurch recht hohe Toleranzen (bis zu +0,3 mm) aufweisen.
- Verbindungen zwischen Platten und Item- oder V-Slot-Profilen werden möglichst einheitlich gestaltet. Grundsätzlich gilt: Verbindungen mit Item-Baureihe (BR) 5 (20x20) werden grundsätzlich in M5 ausgeführt, außer bei Platzmangel in M3. Verbindungen mit BR 8 (40x40) werden in M8 ausgeführt.
- Bei Bauteilen mit hoher mechanischer Beanspruchung wird Aluminiumblech mit 5 mm Stärke verwendet. Dieses wird zwar CNC-gefräst, doch um bei der Fertigung Zeit zu sparen, wird davon abgesehen, Taschen o. Ä. einzuplanen. Stattdessen wird immer ein flaches Profil verwendet, welches sich mit 2.5D-Fräsverfahren mit Durchgangsfräsen, ähnlich einem Lasercutter, fertigen lässt. Bei Aluminiumteilen wird darauf geachtet, dass diese immer aus 5 mm dickem Aluminiumeloxal gefertigt werden.

## Konstruktionsvorgang

Um ein so umfangreiches Projekt umsetzen zu können, muss auch bei der 3D-Planung größtmögliche Ordnung herrschen. Um effizient zu arbeiten, werden wiederverwendbare Bauteile in einer eigens angelegten Bauteilbibliothek abgelegt. So können beispielsweise Aluminiumprofile in Normlängen oder Sensoren einfach in die aktuelle Konstruktion eingefügt werden, ohne diese jedes Mal neu konstruieren zu müssen.

Auch innerhalb einer Konstruktion ist auf Übersichtlichkeit zu achten. Zu diesem Zweck ist es sinnvoll, zusammenhängende Objekte in Komponenten zusammenzufassen. Diese sind vergleichbar mit Ordner. Wie auch Ordner können Komponenten ineinander verschachtelt werden. Dadurch lässt sich eine saubere Struktur umsetzen, die es beispielsweise erleichtert, einzelne Konstruktionsteile zu betrachten oder gezielt Änderungen daran vorzunehmen.

### 2.1.3 Rahmen

Der Rahmen des AFSS bezeichnet jene Struktur, die sowohl als äußeres Gehäuse als auch zur grundlegenden mechanischen Stabilität dient. Er soll die maximal zulässige Höhe und Länge optimal ausnutzen, während die Breite so gering wie möglich gehalten wird.

Umgesetzt wird dies mit einem Gerüst aus 40x40-Aluminium-Extrusionen. Dieses besitzt zusätzlich oben und unten zwei Längsstreben, die eine möglichst platzsparende Aufhängung der X-Achsen-V-Slot-Profile ermöglichen. Auf einer Seite wird zudem eine Aussparung eingeplant, um ausreichend Freiraum zu schaffen, damit der Querförderer die Boxen reibungsfrei auf das Förderband transportieren kann.

Weiterhin muss berücksichtigt werden, dass Rollen für den Transport angebracht werden.

Da diese die maximale Höhe beeinflussen, muss dies in die Planung der Y-Achse einfließen, um sicherzustellen, dass das System problemlos durch Türen bewegt werden kann.

#### 2.1.4 X-Achse

Die X-Achse des AFSS ist das mechanische Herzstück. Sie ist die Achse, die sich horizontal bewegt und einen langen Verfahrweg sowie hohe mechanische Anforderungen besitzt. Sie muss sowohl ihr Eigengewicht als auch das Gewicht der YZ-Achse tragen.

Das Grundelement der X-Achse besteht aus V-Slot-C-Profilen, die oben und unten aufgehängt werden. Diese Profile sind in Längen von 1,5 m erhältlich, daher wird die gesamte Achse so ausgelegt, dass zwei Stück (also insgesamt 3 m Schiene) verwendet werden. Zusätzlich sind die Profile mit jeweils sechs Nivellierungsschrauben pro V-Profil ausgestattet, um sie möglichst parallel zueinander sowie exakt in Waage auszurichten.

#### Grundanforderungen

Auf der X-Achse bewegt sich der X-Schlitten, dessen Aufgabe es ist, die Y-Achse präzise zu positionieren. Dafür muss er alle erforderlichen Motoren und Sensoren enthalten. Zudem muss er einen Angriffspunkt bieten, um seine Bewegung zu ermöglichen.

Der Schlitten sollte sowohl horizontal als auch vertikal möglichst kompakt konstruiert sein, um einen maximalen Verfahrweg in X- und Y-Richtung zu gewährleisten.

#### Antrieb

Angetrieben wird die X-Achse mit zwei Schrittmotoren, welche jeweils an einer Schiene (oben und unten) angebracht werden. Diese treiben mithilfe eines Zahnriemens die zwei X-Schlitten an. Als Zahnriemen wurde aufgrund der hohen Kraftübertragung sowie der großen Länge ein AT5-Zahnprofil mit 16 mm Riemenbreite der Firma Mähdler gewählt. Dieser wird vom Motor über eine entsprechende Zahnscheibe angetrieben und am Shuttle befestigt. Als Schrittmotor wird aus Einfachheitsgründen dasselbe Modell wie bei der Y-Achse verwendet. Diese erzeugen auch genügend Moment, um die X-Achse anzutreiben.

#### Riemenspannung X-Achse

Da der Riemen für eine zuverlässige Kraftübertragung bei so langen Verfahrwegen eine relativ große Vorspannkraft benötigt, ist das Zahnriemenklemmelement auf der X-Achse auch dementsprechend auszuführen. Um den Zahnriemen zu greifen, werden Teile mit Negativverzahnung 3D-gedruckt, in dieses der Zahnriemen dann auf beiden Seiten des Schlittens geklemmt wird. Die Aufhängung der einen Seite ist einfach mit den Item-Profilen verschraubt, lässt aber noch etwas Platz, um den Riemen mit der Hand vorzuspannen. Auf der anderen Seite ist das Klembrett über zwei Gewindestangen mit dem restlichen Shuttle verbunden. Diese können festgezogen werden, um die nötigen Vorspannkräfte zu erzeugen (siehe 2.4a). Außerdem ist diese Spannvorrichtung in einem Formfaktor ausgeführt, welcher es erlaubt, direkt darüber die Motoren der Y-Achse anzubringen (siehe 2.4b).

## Achsenführung

Die Führung der Achse im V-Slot-Profil wird mit V-Wheels durchgeführt. Sechs V-Wheels werden von oben auf das C-Profil gedrückt. Jedes Rad wird einzeln aufgehängt. Weiters ist überall eine Schraube, mit welcher das Rad weiter in die Führung hineingedrückt werden kann, sowie ein bis zwei Klemmpunkte, um bei Betrieb die Last von der Spannschraube nehmen zu können. Durch den geringen Platz im Schlitten ist es durchaus eine Herausforderung, dass all diese Mechanik neben den Spannelementen und Motoren in einem so kleinen Shuttle Platz finden. Auf der Seite des Schlittens werden noch weitere V-Wheels angebracht, welche die Spurführung übernehmen. Im oberen Schlitten werden nur diese Führungsräder verbaut, da es nicht möglich und nötig ist, vertikale Kräfte zu unterstützen.

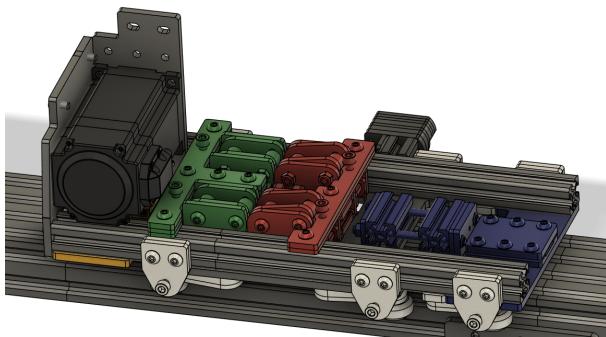
## Kabelführung

Um Sensoren und Aktoren der Y- und Z-Achse zu unterstützen, müssen dementsprechende Leitungen auf das X-Shuttle verlegt werden. Dies wird mithilfe einer Kabelschleppkette der Firma Igus umgesetzt. Diese wird parallel zum unteren C-Profil verlegt und unter Berücksichtigung der Biegeradien am X-Shuttle befestigt. Bei dieser ist darauf zu achten, dass sie alle benötigten Leitungen sowie genügend Freiraum für die Biegung einhält [10]. Weiters müssen Signal- und Aktorstromkreise durch Trennstege voneinander getrennt werden.

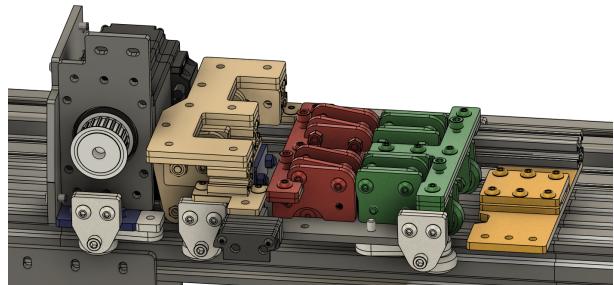
Um die weiteren Geräte am AFSS zu versorgen, müssen zusätzlich noch Verdrahtungskanäle am Rahmen angebracht werden.

## Sensoren

Die Sensorik ist bei jeder Achse ähnlich aufgebaut. Immer zwei Endschalter und ein Referenzierschalter. Bei der X-Achse werden als Endschalter mechanische Rolltaster verwendet. Diese werden neben den V-Slot-Profilen befestigt und von einem, vom Schlitten abstreifenden Arm ausgelöst.



(a) X-Schuttle Ansicht hinten  
(Motor rechts und Rollen ausgeblendet)



(b) X-Schuttle Ansicht vorne  
(Motor rechts und Profil vorne ausgeblendet)

- |  |  |  |
|--|--|--|
| <span style="color: green;">█</span> Rollen Feste Seitig | <span style="color: blue;">█</span> Spannlement          | <span style="border: 1px solid black; padding: 2px;">█</span> Seitliche Rollen |
| <span style="color: red;">█</span> Rollen Mitte          | <span style="color: yellow;">█</span> Spannlement Fest   | <span style="color: black;">█</span> Sensorauslöse                             |
|  | <span style="color: orange;">█</span> Rollen Spannseitig |  |

Abbildung 2.4: X-Achse

### 2.1.5 Y-Achse

Als Y-Achse wird jene Achse bezeichnet, die ihre Bewegung vertikal durchführt. Sie hat die Aufgabe, die Z-Achse bzw. das YZ-Shuttle auf Position zu bringen. Wichtig hierbei ist jedoch, dass die Y-Achse die Aufgabe des Aufhebens der Box übernimmt.

#### Antriebsauslegung

Dadurch, dass die Y-Achse sowohl das YZ-Shuttle als auch die Boxen aufheben muss, muss der Antrieb dementsprechend dimensioniert werden. Als Formfaktor soll ein Nema23-Schrittmotor verwendet werden. Diese sind weit verbreitet und im Vergleich zu Servomotoren relativ kostengünstig. Als Grundformfaktor wird ein 2 Nm Motor gewählt. Nun soll überprüft werden, ob dieser die Last der YZ-Achse auch antreiben kann.

$$F = \frac{M}{\frac{d}{2} \cdot 1000} \cdot n$$

$$F = \frac{2 \text{ Nm}}{\frac{30 \text{ mm}}{2} \cdot 1000} \cdot 2 = 266 \text{ N}$$

$F$ : Antriebskraft der Y-Achse	[N]
$M$ : Drehmoment eines Motors	[Nm]
$d$ : Durchmesser der Zahnscheibe	[mm]
$n$ : Anzahl der Antriebe	

Bei der Konstruktion einer früheren Version des YZ-Shuttles wurde erfasst, dass das Shuttle bis zu 13 N wiegen kann. Dies wird zwar in einer späteren Iteration des Designprozesses noch verbessert, dient jedoch als Richtwert für die Antriebsauslegung. 13 kg erzeugen ohne Berücksichtigung von Reibung knapp 130 N. Die Antriebskraft der Schrittmotoren reicht auf jeden Fall aus. Doch diese Überdimensionierung ist unter dem Aspekt, dass die An-

triebe über keine Bremse verfügen und somit das gesamte Gewicht der Z-Achse sowie der Kabelschleppkette immer unterstützt werden müssen, durchaus sinnvoll.

### Zahnriemen und Umlenkung

Als Zahnriemen wird auch bei der Y-Achse auf ein AT5x16-Profil gesetzt. Doch hier gestaltet sich die Positionierung ebendieses nicht so simpel wie bei der X-Achse. Da der Zahnriemen beim YZ-Shuttle an einem bestimmten Punkt befestigt werden muss, muss er auch dort wieder rückgeführt werden. Die Umlenkung des Zahnriemens gestaltet sich jedoch wesentlich anspruchsvoller als bei der X-Achse, da einerseits ein möglichst kompakter Formfaktor angestrebt werden muss und andererseits eine aufhängungstechnisch sehr unvorteilhafte Positionierung vor dem V-Slot-C-Profil erforderlich ist. Aus diesem Grund wird eine Konstruktion aus Aluminium, welche sich selbst verhakt, konstruiert. Sie muss seitlich an den Profilen verschraubt werden. Im vorderen Überhang werden Aluminiumelemente eingehängt, die die Aufhängung der Achse für die Umlenkung erlauben.

### Zahnriemenaufgängung

Um die optimale Position der Zahnriemenaufhängung für die Y-Achse bestimmen zu können, wird überschlagsmäßig ein Massenschwerpunkt in Z-Richtung berechnet. Um mit der Konstruktion beginnen zu können, werden hierfür ungefähre Werte angenommen.

$$Z_s = \frac{1}{M} \cdot \sum_{i=0}^n z_i \cdot m_i$$

$Z_s$  : Position des Massenschwerpunkts [m]

$M$  : Gesamtmasse [kg]

$z_1$  : Z-Koordinate der Teilmasse [m]

$m_1$  : Masse der Teilmasse [kg]

Gegenstand	Masse in kg	Position in mm
Motor	0.3	21
Z-Schiene	1	150
Z-Gable	0.3	180

Tabelle 2.1: X-Achse unbeladen und eingefahren

Gegenstand	Masse in kg	Position in mm
Motor	0.3	21
Z-Schiene	1	150
Z-Gabel	1.3	380

Tabelle 2.2: X-Achse beim Ladevorgang

So werden zwei Schwerpunkte errechnet: ca. 130 mm im unbeladenen Zustand und 250 mm während dem Ladevorgang. Da die Stabilität des Y-Shuttles während dem Ladevorgang wichtiger ist als während einer Leerfahrt, wird das Y-Shuttle so positioniert, dass die

Aufhängung des Zahnriemens bei rund 200 mm liegt.

Weiters muss auch dieser Zahnriemen wieder gespannt werden. Durch die geringere Zahnriemenlänge wird auch der Spannmechanismus verkleinert. Leider muss dieser aus Platzgründen außerhalb der Flucht zum Zahnriemen angebracht werden. Die Folgen dieser Auslenkung auf die Riemenspannung können durch folgenden Ausdruck, der die Länge, die der Zahnriemen bei einer bestimmten Position braucht, um spannungslos zu sein, approximiert, ermittelt werden. (Hierbei wird von einem linearen Zusammenhang zwischen der Kraft auf die Aufhängung und der Längenkontraktion ausgegangen. In der Realität würde jedoch die Elastizität des Zahnriemens sowie jene der Aufhängung Auswirkungen haben, aber diese verringern die Auswirkungen nur zu unseren Gunsten).

$$K(h) = \sqrt{h^2 + d^2} + (l - h)$$

$$K(h) = \sqrt{h^2 + 28^2} + (1500 - h)$$

$K(h)$  : theoretische Länge des Riemen [mm]  
 $h$  : Position des Schlittens [mm]  
 $l$  : Umlenkungsrollenabstand [mm]  
 $d$  : Abstand der Klemme zur Flucht [mm]

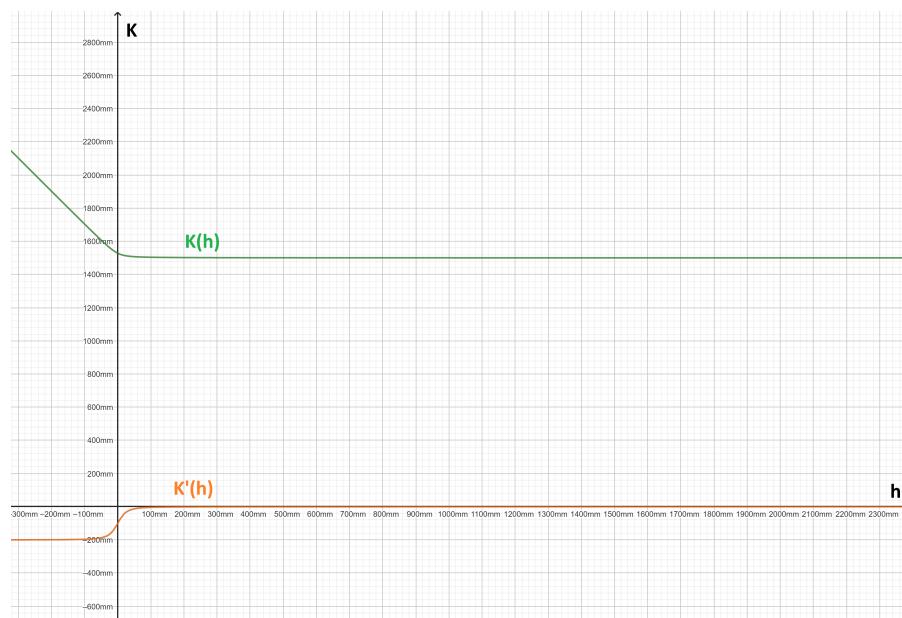


Abbildung 2.5: Zahnriemenkontraktion grafisch dargestellt

Aus der Abbildung 2.5 geht, wie schon intuitiv vermutet, hervor, dass diese Asymmetrie nur dann ein Problem darstellt, wenn der Schlitten weit unten ist. Ein konkretes Maß kann folgendermaßen ermittelt werden:  $K(80) - (750) = 4.24\text{mm}$ . Der Unterschied in der benötigten Riemenlänge zwischen der Mittelstellung (750 mm) und der Ruheposition (80 mm über der Zahnscheibe) beträgt also rund 4 mm. Im Hinblick darauf, dass diese Länge auf zwei Seiten (Hin- und Rücklaufseite) des Zahnriemens aufgeteilt wird (welcher etwas elastisch ist), sowie dass eine leichte Biegung in der Umlenkungsaufhängung auftritt, wird diese Extraspannung toleriert.

## Shuttleführung

Das YZ-Shuttle wird aus Stabilitätsgründen ebenfalls mit einem V-Slot C-Profil geführt. Diese sind auf beiden Seiten jeweils oben und unten befestigt. Die Länge wird so gewählt, dass die Verwendung eines 1,5 m langen Profils perfekt ausreicht. Da das YZ-Shuttle wesentlich weniger Kraft auf die Führung auswirkt, können weniger Führungsräder verwendet werden. Die Hauptführungsräder werden so positioniert, dass sich ein Dreieck ergibt. Dieses dient dazu, dass sowohl beim Aufhebevorgang als auch beim Leerlauf immer eine Klemmung um die Führungsschiene entsteht. Hierbei sind jene V-Wheels, welche beim Aufhebevorgang belastet werden, doppelt ausgeführt. Wie schon bei der X-Achse sind auch hier wieder alle V-Wheel-Abstände mit Schrauben einstellbar. Zusätzlich zur Hauptführung sind auch außen jeweils noch zwei Führungsräder angebracht, um das Shuttle zusätzlich zu stabilisieren.

## Sensoren

Da die Endschalter der Y-Achse kapazitiv ausgeführt sind, muss auf dem Shuttle ein metallisches Gegenstück angebracht sein. Diese stehen oben und unten über und lösen so vor Kollision aus. Um die Sensoren anzubinden, muss auch ein ASi-Client auf dem X-Shuttle angebracht werden.

## Kabelschleppkette

Um Versorgung der Z-Achse herstellen zu können, muss eine Kabelschleppkette vom X-zum Y-Shuttle angebracht werden. Ach diese muss leider Sonderanforderungen erfüllen. Da die Versorgung von unten ausgeht, muss die Kabelschleppkette stehend eingebaut werden. Dies ist grundsätzlich eine äußerst ungünstige Situation, da diese Schleppkette der Beschleunigung des X-Shuttles ausgesetzt ist. Um ein Schwingen möglichst zu verhindern, werden seitlich noch Führungselemente angebracht. Außerdem sind die Anschlusskomponenten fest um die ersten Kettenglieder extra zu unterstützen. [11]

### 2.1.6 Z-Achse

Als Z-Achse oder Gabel, wird jener Teil des AFSS bezeichnet, der die Boxen in das Lager ein- und ausfährt. Dieser ist in das YZ-Shuttle integriert. Es wird davon ausgegangen, dass um Boxen ein- und auszuheben ca. 210 mm Überstand der Gabel benötigt wird. Dies ist also der Mindestverfahrtsweg der Z-Achse

## Linearführung

Geführt wird die Gabel mit zwei Führungsschienen von Igus. Diese bieten optimale Stabilität sowie, in Verbindung mit einem Führungswagen, eine reibungsarme Bewegung. Wichtig ist jedoch, dass der Schwerpunkt der am Führungswagen befestigten Last nicht mehr als die doppelte Wagenlänge über den Wagen hinausgeht. Danach kommt es zu sehr starker Verklemmung, und ein Betrieb ist nur mehr schwer möglich. Da bei einer Auslegeroperation der Schwerpunkt sehr weit übersteht, wird eine TS-01 Führungsschienen-

und -wagenkombination verwendet. Diese wirkt zwar recht überdimensioniert, doch da der Führungswagen so lang ist, kann ein reibungsarmer Betrieb gewährleistet werden.

## Motor

Als Antrieb für die Z-Achse sollen zwei Nema-17 Schrittmotoren verwendet werden. Hier ist es nicht nötig, eine Closed-Loop-Steuerung zu verwenden, da bei jedem Hub auch referenziert werden kann. Weiterhin kann dadurch Kabelschleppkettenplatz gespart werden.

## Spindelauslegung

Die Z-Achse wird mit einer Spindel angetrieben. Dies bietet Vorteile in der Positionsgenauigkeit und der Kraftübertragung. Jedoch ist es wichtig, die richtige Spindelsteigung auszuwählen, um die Balance zwischen Geschwindigkeit und Kraft zu halten. Ziel ist es, für eine Richtung des Hubs maximal 5 Sekunden zu benötigen. Sie wird über eine Zahnscheibe und Zahnriemen mit dem Motor verbunden.

Berechnet werden kann diese Zeit mit folgendem Ausdruck:

$$\begin{aligned}
 M_{mot} &: \text{Drehmoment} & \left[ \frac{N}{m} \right] \\
 v = \frac{k}{n_{welle}} &= \frac{k}{\frac{n_{motor}}{i}} & n_{mot} &: \text{Drehzahl} & \left[ \frac{1}{s} \right] \\
 F = \frac{M_{mot} \cdot i \cdot k}{n_{welle}} &\cdot 2\pi f \cdot \eta & k &: \text{Wellensteigung} & \left[ \frac{mm}{U} \right] \\
 && i &: \text{Übersetzungsverhältniss} \\
 && \eta &: \text{Wirkungsgrad der Gewindeschraube}
 \end{aligned}$$

So wird berechnet, dass eine DS10x12-Spindel mit ihrer 12 mm Steigung, bei einer Motordrehzahl von  $600 \text{ U min}^{-1}$  ( $0,42 \text{ N m}$ ) und einem Übersetzungsverhältnis von 2:1, ca. 4 Sekunden pro Richtung benötigt und mit einer Kraft von ca. 90 N bewegt wird.

Dies entspricht unseren Anforderungen, und somit wird diese Spindel gewählt. Um sie zu lagern, wird vorne und hinten der Durchmesser der Spindel verringert, sodass diese in Kugellagern geführt werden kann.

## Sensorik

Um auch die Endschalter der Z-Achse, sowie weitere Sensoren einzulesen, wird auch hier ein ASi-Slave montiert.

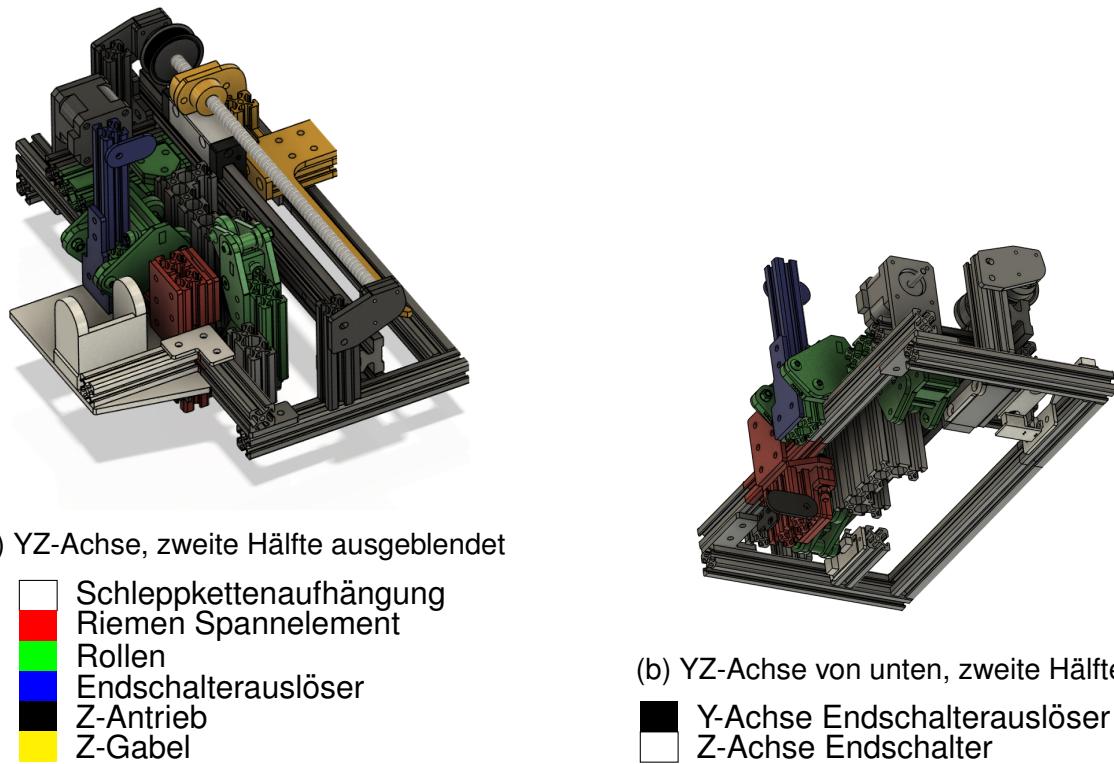


Abbildung 2.6: YZ-Achse

### 2.1.7 Lager

Das Lager soll die Boxen beinhalten und die Möglichkeit zulassen, dass diese von der Gabel ein- und ausgehoben werden. Weiterhin muss die Box in X- und Z-Richtung geführt werden, um die Positionsgenauigkeit sicherzustellen, da sonst die Gabel möglicherweise in die Box fährt. Das Lager soll darauf ausgelegt werden, dass Boxen mit den Maßen 50 mm × 100 mm × 200 mm verwendet werden können. Diese Boxen sind nach unten hin verjüngt und haben oben eine Lippe, an der die Gabel greift. Umgesetzt wird dies mit einem Gerüst aus 40x40-Item-Profilen, welches im Nachhinein in den restlichen Rahmen eingesetzt wird. Auf diese werden 20x40-Profile horizontal aufgeschraubt, auf denen die Boxen stehen. Vorne und hinten wird eine Kunststoffplatte aufgeschraubt, welche leicht übersteht und somit die Box in Z-Richtung positioniert. Zwischen den Boxen wird ein Trennsteg eingebaut, welcher die Boxen in X-Richtung positioniert sowie den richtigen Abstand zwischen den Boxen einhält.

## 2.1.8 Querförderer

Da es dem Portalroboter nicht möglich ist, die Boxen direkt auf das Förderband zu legen, muss hier noch ein System eingebaut werden, welches dies erledigt. Da die Boxen beim Ein- und Auslagern den gleichen Weg zurücklegen, muss dieser Querförderer die Box sowohl auf das Förderband als auch vom Förderband herunterbewegen können.

Zu diesem Zweck wird eine weitere, der Gabel ähnliche Konstruktion montiert, welche auf einem 20x40-V-Slot-Profil verläuft. Die Box wird dann hin- und hergeschoben, um vom Lager auf das Förderband umzuladen. Dadurch ist es noch möglich, dass die Gabel der Z-Achse die Box in der Nullstellung ein- und aushebt.

## 2.1.9 Fertigung der Einzelteile

### Umlenkrollen

Die Umlenkrollen sind jeweils am Ende der X- und Y-Achsen angebracht. Da diese der gesamten Spannkraft ausgesetzt sind, erfordert dies spezielle Anforderungen an die Aufhängung sowie an die Umlenkrolle selbst. Diese soll primär eine 180°-Wende des Zahnriemens ermöglichen und sekundär eine Führung für den Riemen bieten.

Um diese Anforderungen umzusetzen, werden vier Aluminium-Drehteile gefertigt, in welche Kugellager eingepresst werden.

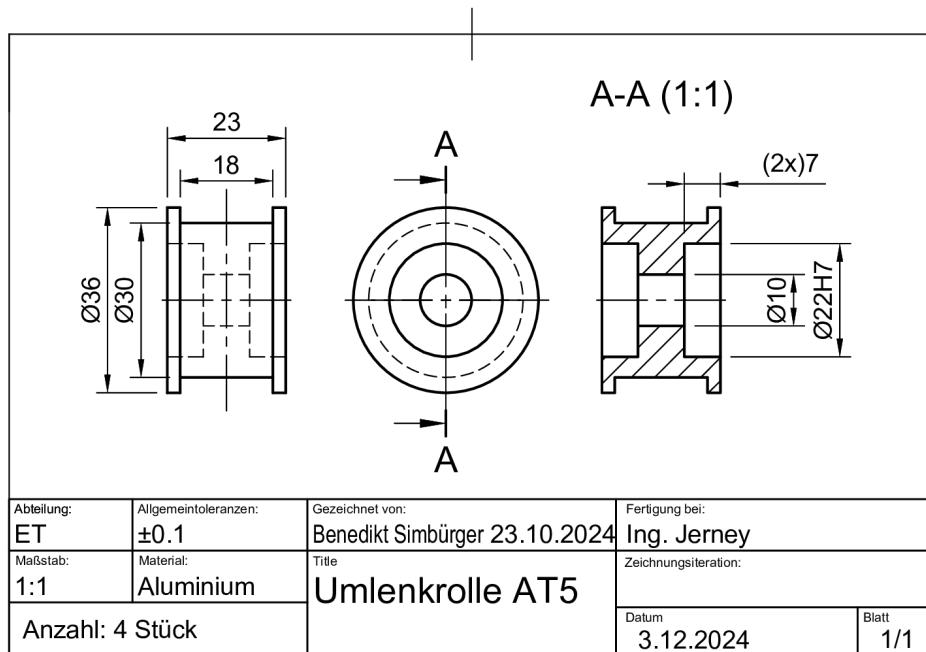


Abbildung 2.7: Bauteilzeichnung Umlenkrolle

Die Fertigung dieses Teils nach Abb. 2.7 lässt sich in folgende Teilschritte unterteilen:

- Zuerst die Frontfläche plandrehen (Drehzahl: 900 U min<sup>-1</sup>)

- Ungefähr 30 mm Länge auf das Aussenmaß von 36 mm längsdrehen
- Mit 9,8 mm Bohrer das mittlere Loch vorbohren ( $540 \text{ U min}^{-1}$ )
- Mit 10 mm Reibeisen und viel Öl das Loch auf eine genaue Passung bringen ( $260 \text{ U min}^{-1}$ )
- Die Position relativ zum Backenfutter markieren, um beim Neu-Einspannen Rundlaufgenauigkeit zu gewährleisten
- Zylinder bei ca. 26 mm abstechen ( $540 \text{ U min}^{-1}$ )
- Umspannen und auf Maß plandrehen ( $900 \text{ U min}^{-1}$ )
- Die Aussparung für die Lager mit Eckdrehmeißel beginnen, jedoch nach innen hin nur 6,8 mm
- Bei ca. 17 mm Lochdurchmesser den tatsächlichen Durchmesser mit der Digitalanzeige vergleichen und gegebenenfalls korrigieren
- Bei 21,5 mm den Oberschlitten die restlichen 0,2 mm zustellen und die gesamte Tiefe plandrehen
- Den Lochdurchmesser auf 21,95 mm erweitern und dann in kleinen Inkrementen zustellen, bis das Lager gerade so nicht passt, um einen Pressitz zu gewährleisten. Dies tritt bei Lagern mit 22 mm Außendurchmesser bei rund 22,045 mm ein.
- Da für die Einsparung der Riemenführungsfläche kein Angriffspunkt verfügbar ist, wurde als Halterung ein Dorn nach Abb. 2.8a gedreht, auf welchen das Drehteil aufgeschraubt wird.
- Mit dem Abstechmeißel wird in 2 mm Inkrementen die Zahnriemenauflagefläche herausgedreht, bis auf 21,95 mm, sowie links und rechts den Rand 1 mm extra dick lassen (siehe Abb. 2.8b). ( $540 \text{ U min}^{-1}$ )
- Am Schluss wird die Rand Tiefe auf Maß gedreht.
- Als letzten Schritt werden links und rechts die zwei Lager eingepresst.

Durch die verhältnismäßig großen Toleranzen bei den Lageraussendurchmessern wird bei 2 der 8 Lagerpassungen zusätzlich Lagerkleber verwendet, um einen zuverlässigen Halt zu gewährleisten, da bei diesen die Toleranzen nicht eingehalten wurden.

Nach diesem Vorgang sind die vier Umlenkrollen fertig (siehe 2.8c) und können auf einem 8-mm-Schaft montiert werden.

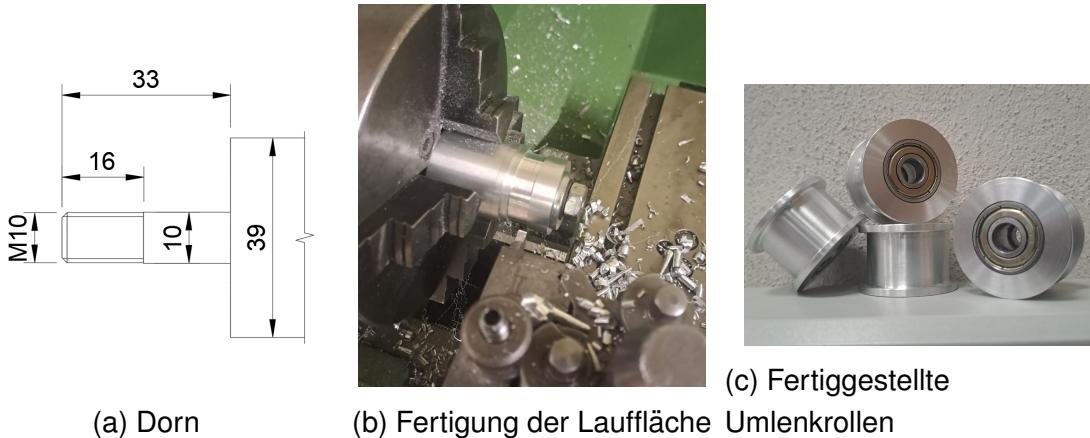


Abbildung 2.8: Umsetzung der Umlenkrollen

## Distanzhülsen

Bei den meisten V-Wheels werden Distanzhülsen zur Klemmung benötigt. Diese müssen ein sehr spezielles Maß (13,4 mm Länge) haben. Deshalb müssen Hülsen mit einer Länge von 20 mm auf diese Länge heruntergedreht werden. Dies wird mit dem rechten Eckdrehmeißel bei einer Drehzahl von 740 U/min ausgeführt. Da relativ viele solcher Teile benötigt werden, wird beim Einspannen der Drehmeißel selbst als Endstopp verwendet, um ein relativ wiederholgenaues Maß zu erhalten sowie eine simple Durchführung zu erlauben. Es wird also der Meiβel auf Position gefahren und das Werkstück eingespannt, sodass es am Drehmeißel ansteht. Nun wird der Drehmeißel zurückgefahren und die Hülse kann einfach durch mehrere Plandrehoperationen, bis der Schlitten ansteht, gekürzt werden. Es muss also weder gemessen noch die Digitalanzeige verändert werden, um schnell mehrere Teile hintereinander zu fertigen.

## Zahnscheiben

Da die Welle des Motors recht kurz ist und die Motoraufhangung Platz wegnimmt, ist es erforderlich, die Zahnscheibe unkonventionell mit dem Motor zu verbinden. Es werden also in der Zahnriemen-Kontaktflache zwei Locher gebohrt, angesenkt und mit M4-Gewinde versehen, um dort 2 M4-Wurmschrauben einzuschrauben. Bei der Lange der Wurmschrauben muss darauf geachtet werden, dass sie im montierten Zustand vollstandig unter der Oberfache liegen, um den Zahnriemen nicht zu beschadigen.

## 2.1.10 Aufbau

Bahmen

Gestartet wird mit der Monate des Rahmen. Hierfür werden zuerst alle Aluminiumprofile auf Länge geschnitten. Da nicht genug Profile in der Gesamtlänge des AFSS verfügbar sind, müssen teils noch Verbinden eingesetzt werden. An den Ecken werden die Profile dann mit Standardverbindungssätzen verbunden. Unten werden dann noch die Rollen montiert.

### X-Achse

Nachdem die Aufhängungen für die V-Slot Profile gefräßt sind, werden diese an den Rahmen angeschraubt. Auf die V-Slot Profile werden das Nivellierungssystem aufgeschraubt, dann werden die Profile an der Aufhängung befestigt.

Danach werden die Umlenkrollen- und die Motoraufhängungen montiert. In diese werden dann Umlenkrollen und Motoren montiert.

## 2.2 Software und Benutzeroberfläche

### 2.2.1 Grundlegendes

Um dem Endnutzer die Möglichkeit zu geben, das AFSS möglichst einfach zu bedienen und die komplexe Logik der Lagersteuerung auszuführen, bedarf es eines Servers (Backend) und einer grafischen Benutzeroberfläche (GUI oder Frontend). Diese müssen eine Vielzahl an verschiedenen Funktionen beinhalten.

### 2.2.2 Aufbau

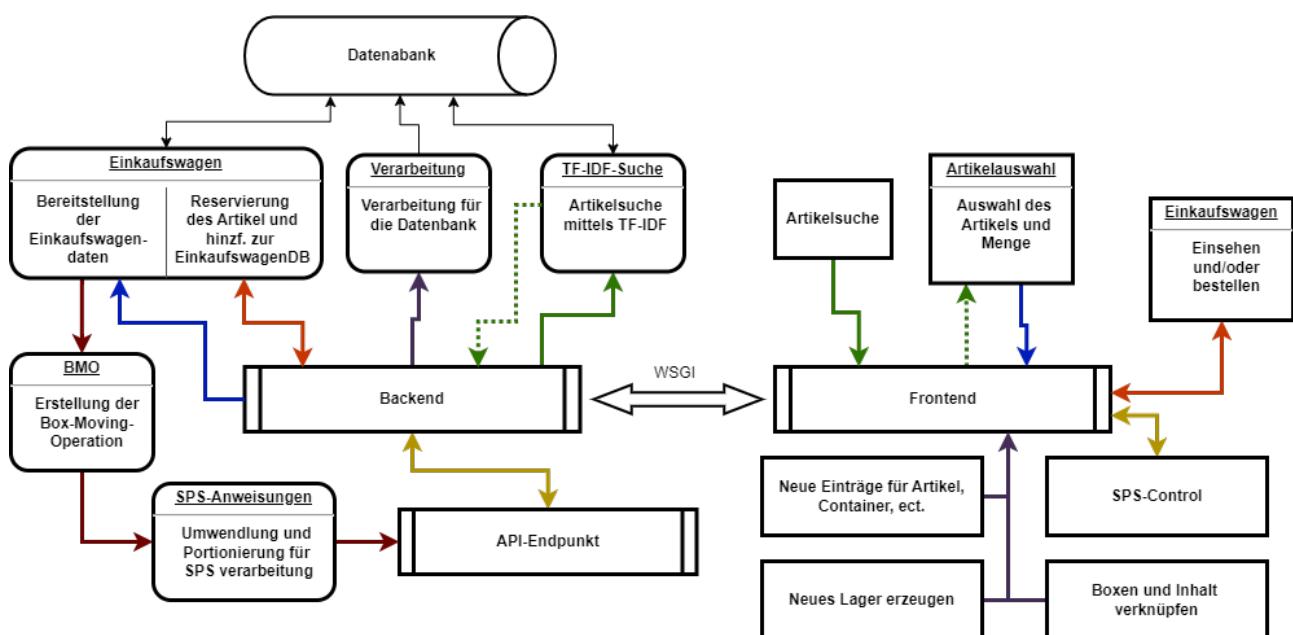


Abbildung 2.9: Gesamtüberblick des Servers

Um diesen Anforderungen gerecht zu werden, muss eine Lösung mit sehr hohem Grad an Freiheit in der Logik sowie der UI-Gestaltung gewählt werden. Weiters muss es möglich sein, dass zukünftige Schülerinnen und Schüler diese instand halten und erweitern. Aus diesen Gründen, sowie den bereits vorhandenen Kenntnissen, wird die Programmiersprache 'Python' als Grundlage des Servers verwendet.

### Python

Python ist eine vielseitige und hochentwickelte Programmiersprache, die für ihre Einfachheit und Lesbarkeit bekannt ist und sich sowohl für Anfänger als auch für Fortgeschrittene eignet. Sie unterstützt mehrere Programmierphilosophien, darunter objektorientierte und funktionale Programmierung, und wird in Bereichen wie Webentwicklung, Datenanalyse, künstliche Intelligenz und wissenschaftlichem Rechnen häufig eingesetzt. [12]

## Flask

Flask ist ein leichtgewichtiges Web-Framework für Python, das durch seine Einfachheit und Flexibilität hervorsteht und sich ideal für kleinere Anwendungen oder Prototypen eignet. Es folgt einem minimalistischen Ansatz, bietet aber Erweiterungsmöglichkeiten, um komplexere Projekte zu realisieren. [12]

## Python - Virtuelle Umgebung

Dieses Projekt enthält sehr viele externe Bibliotheken. In Python sind diese Bibliotheken mit dem Interpreter verknüpft, da bei der Installation diese externen Bibliotheken direkt beim Interpreter gespeichert werden. So kommt es jedoch dazu, dass, wenn das Programm auf einer anderen Maschine ausgeführt wird, diese Bibliotheken nicht vorhanden sind. Um Abhängigkeitskonflikte und Portabilitätsprobleme zu verringern, werden virtuelle Umgebungen verwendet. Diese enthalten den Interpreter sowie die Bibliotheken und könnten einfach auf eine andere Maschine kopiert werden.

Dies ist jedoch bei diesem Projekt nur bei der Entwicklung vonnöten, da es bei Fertigstellung im Docker-Container ausgeführt wird.

### 2.2.3 Benutzeroberfläche

Um die Benutzeroberfläche zu realisieren, muss eine Weboberfläche erstellt werden. Auf dieser werden alle Inhalte angezeigt, die für die Benutzung nötig sind. Sie wird vom Server zur Verfügung gestellt, sobald dieser eine HTTP-Anfrage erhält. Um diese mit eigenen Inhalten und Funktionen zu befüllen, muss dies mit HTML geschehen.

## HTML

HTML (HyperText Markup Language) ist die Standard-Auszeichnungssprache zur Strukturierung und Darstellung von Inhalten im Web. Sie definiert die grundlegende Struktur einer Webseite mit Elementen wie Überschriften, Absätzen, Links, Bildern und Formularen. [12]

Dadurch, dass sich viele Elemente des UI wiederholen, wie z. B. die Navigationsleiste, bietet Flask die Möglichkeit, sogenannte 'Templates' zu verwenden. Diese können einmal definiert und dann an mehreren Teilen der Webseite verwendet werden. Um Elemente wie Formularfelder oder Datenanzeige einfach mit den benötigten Daten anzuzeigen, gibt es die Möglichkeit, Makros zu erstellen, die von Flask mit den bestimmten Daten vorgerendert und in das restliche HTML eingefügt werden. Um HTML, welches grundsätzlich ohne Formatierung auskommt, zu stylen, muss CSS verwendet werden.

## CSS

CSS (Cascading Style Sheets) ist eine Stylesheet-Sprache, die verwendet wird, um das Design und Layout von Webseiten zu gestalten. Sie ermöglicht die Trennung von Inhalt und

Darstellung, indem sie Farben, Schriftarten, Abstände und andere visuelle Aspekte definiert.[12]

Da auch Logik in der Webseite verbaut werden muss, muss zusätzlich auch Javascript verwendet werden, da HTML und CSS alleine, noch nicht gut genug mit dem Server Kommunizieren können.

## JavaScript

JavaScript (JS) ist eine vielseitige Programmiersprache, die hauptsächlich verwendet wird, um interaktive und dynamische Elemente auf Webseiten zu erstellen. Sie läuft direkt im Browser und ermöglicht Funktionen wie Animationen, Formularvalidierungen und die Kommunikation mit Servern in Echtzeit.[12]

Praktisch geschieht diese Server-Kommunikation immer mithilfe dieser Programmblocks:

```
1  function sendData(data, callback) {
2      var xhr = new XMLHttpRequest();
3      var url = "{{url_for('main.add_stock')}}"; //Flask markup, um die
4          richtige url zu erreichen, dies wird vor ausgabe auf der Webseite noch
5          eingesetzt
6
7      xhr.open("POST", url, true);
8      xhr.setRequestHeader("Content-Type", "application/json");
9
10     xhr.onreadystatechange = function () {
11         if (xhr.readyState === 4 && xhr.status === 200) {
12             callback(xhr.responseText) //die funktion wir aufgerufen
13         }
14     };
15     var jsonData = JSON.stringify(data);
16     xhr.send(jsonData);
17 }
```

Dieser ermöglicht die Übergabe von Daten im JSON format, und einer Funktion, die die zurückgeschickten Daten verarbeitet. In der Praxis wird dieser so aufgerufen:

```
1 function add_to_db(){
2     sendData({ "add_stock": { "barcode": barcode, "quantity": quant, "article"
3         : article } },
4         set_gen_stock)
5
6 function set_gen_stock(req){
7     document.getElementById("generated").innerHTML = req
8 }
```

Wie im Quellcode ersichtlich, werden Daten aus der Webseite ausgelesen und in JSON konvertiert. Danach werden diese Daten zusammen mit einer Funktion an 'send\_Data' übergeben.

Wie bereits erwähnt, gibt der Server dann Daten zurück. In diesem Fall werden dann Daten aus der DB vorformatiert. Diese werden dann in der zuvor übergebenen Funktion in die Webseite eingefügt.

## 2.2.4 Backend

Das Backend des Servers ist für die Datenverarbeitung verantwortlich. Es ist, wie bereits erwähnt, in Python geschrieben und stellt mit dem Flask-Framework die Benutzeroberfläche zur Verfügung.

Aufgebaut ist es in mehrere Bereiche: Einerseits die Webanwendung sowie die API (Application Programming Interface, Schnittstelle zwischen Anwendungen), die Anbindung an die Datenbank, die Verarbeitung der SPS-Befehle und auch der Zugriff auf die SPS.

### Serverseite der Benutzeroberfläche

In Flask können 'blueprints' definiert werden. Dies sind Webseitelemente die einen bestimmten URL-Vorsatz haben. So werden anfangs 'blueprints' für Hauptfunktionen ('/', also ohne Vorsatz), Datenbankinteraktionen ('/db\_interactions') usw. definiert. Die Funktionen dafür werden dann in jeweils eigenen Dateien geschrieben. Dies ermöglicht eine weit aus bessere Übersicht bei großen Projekten.

Eine Funktion, die für die Verarbeitung der Anfragen einer bestimmten URL verantwortlich ist, sieht immer ähnlich aus.

```
1  @main.route("/add_stock", methods=["GET", "POST"])
2  def add_stock():
3      if request.method == "POST":
4          if request.data:
5              req = request.get_json()
6
7              if "add_stock" in req.keys():
8                  dt = req["add_stock"]
9                  new = Stock(
10                      container=db.session.query(Container)
11                      .filter_by(barcode=dt["barcode"]))
12                      .first()
13                      .id,
14                      article=dt["article"],
15                      quantity=dt["quantity"],
16                  )
17                  db.session.add(new)
18                  db.session.commit()
19                  return "Success"
20
21      return render_template("add_stock.html")
```

Anfangs wird mit einem Decorator (@main.route(...)) die gewünschte URL, sowie unterstützte http-Requests definiert. Decoratoren verändern oder erweitern die Eigenschaften von Funk-

tionen. In diesem Fall wird in der Funktion (def ...) definiert was geschieht, wenn ein erlaubter Request an der URL 'add\_stock' eintrifft. Dieser Funktionsname kann auch in den Flask-Vorlagen verwendet werden um URLs dynamisch zu vergeben.

Weiters wird sortiert um welche Art von Anfrage es sich handelt. Bei GET-Anfragen wird typischerweise einfach nur das HTML der Webseite zurückgegeben. Bei POST-Anfragen werden zuerst die Daten dieser Anfrage extrahiert und dann entschieden was damit gemacht werden soll. In diesem Fall wird, wenn das richtige Schlüsselwort in der Anfrage enthalten ist, ein Datenbankeintrag, mit den Daten aus dem Request, hinzugefügt. Schlussendlich wird ein Wert zurückgegeben, entweder ein HTML-Statuscode, vorgerendertes HTML oder, wie in diesem Fall, ein Text.

## 2.2.5 APIs

### SPS - Verbindung

Die Daten für die SPS werden in einem Ähnlichen Vefahren zu verfüigung gestellt. Nun schickt nicht die Benutzeroberfläche oder der Browser eine Anfrage an das Backend, sondern die SPS. Der Programmblöck zur Verarbeitung dieser Anfrage sieht folgenderaßen aus:

```
1 @api.route("/afss", methods=["GET", "POST"])
2 def afss():
3     request_data = {...} #aus platzgruenden verkuerzt
4     if request.method == "POST":
5         ... # Abfrage ob der Sender eine Siemens SPS ist
6         if "next_bmos" in req.keys():
7             if req["next_bmos"] == "":
8                 return "400"
9
10        if is_SPS:
11            return convert_instruction_for_PLC(afss_stack.
12                get_current_bmos(int(req["next_bmos"])))
13
14        return jsonify(afss_stack.get_current_bmos(
15            int(req["next_bmos"])))
16
17        if "inst_acknowledge" in req.keys():
18            if is_SPS:
19                ack = afss_stack.inst_acknowledge(
20                    int(req["inst_acknowledge"]))
21                if ack == "204":
22                    return ack
23                return convert_instruction_for_PLC(ack)
```

Die Funktion dieses Codeblocks besteht darin, herauszufiltern, ob eine Anfrage einer Siemens-SPS eintrifft und dann dementsprechend Daten zurückzugeben. Wenn eine Anfrage mit dem Schlüssel 'next\_bmos' (next-box-moveing-operation) empfangen wird, wird diese dem sog. 'stack' weitergegeben, welcher diese dann verarbeitet (dies wird im Anschluss näher behandelt). Sollte diese von einer SPS kommen, werden die Zurückgeschickten Daten vorher noch

so bearbeitet, dass diese von der SPS möglichst gut übersetzt werden können. Sollte die SPS ein 'inst\_acknowledge' sowie die dementsprechenden Daten, wird im 'Stack' vermerkt dass die dementsprechende Operation bei der SPS angekommen ist.

Weiters bietet Siemens auch die Möglichkeit über den Webserver der CPU auf diese zuzugreifen. Dies geschieht über das JSON-RPC Protokoll [13]. Aus Testzwecken ist dies durchaus nützlich, aber direkten Zugriff auf die CPU zu haben, wenngleich es für den Normalbetrieb nicht zwingend benötigt wird. Um eine möglichst einfache Bedienung zu ermöglichen wird ein Objekt angelegt, um JSON-RPC Befehle auszuführen.

- Login: Um die SPS zugreifen zu können muss erst ein Login geschehen. Hierbei werden Benutzername und Passwort benötigt, die zuerst in der SPS festgelegt wurden. Dieser führt bei Erfolg zu einem 'Session-Token', welches bei jeder folgenden Kommunikation zur Authentifizierung mitgeschickt werden muss.
- Daten Lesen: Die Methode 'PlcProgram.Read' gibt den Wert einer gegebenen Variable zurück.
- Daten Schreiben: Mit 'PlcProgram.Write' kann der Wert einer oder mehrerer Variablen geändert werden.

Diese Objekt kann dann einfach in einen Anderen Programmteil importiert werden, und dort dann die Verbindung zu einer SPS herstellen. Theoretisch könnte man auch mehrere dieser Objekte anlegen, um Zugriff mehreren Steuerungen umzusetzen.

## 2.2.6 Lageralgorithmus für SPS

Die Aufträge sind im Einkaufswagen o.ä. im Format 'Location - Location' hinterlegt. Um dies in eine Operation zu zerlegen, die die SPS versteht, wird eine Box-Moveing-Operation erstellt. Diese enthält alle Teilschritte, wie Schlittenposition oder Förderbandstracke, um eine Box von Position A zu Position B zu bringen. Die BMO wird dann in die einzelnen verwendeten Module, Förderband oder Lager, aufgeteilt und dann dem 'stack' zugeführt. Diese Instruktion enthält Daten wie:

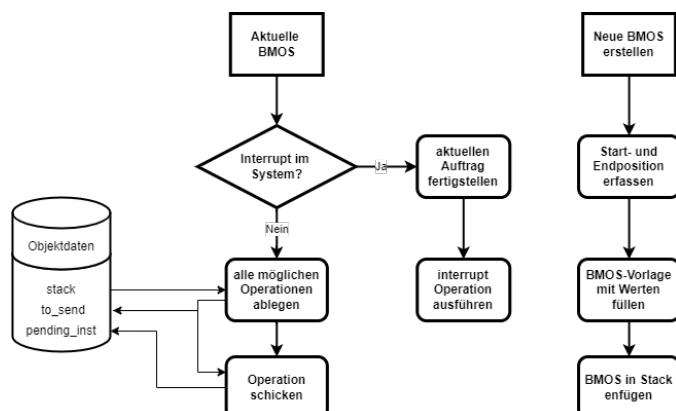


Abbildung 2.10: Diagramm der BMOS bereitstellung

- 'instruction\_id': Eine fortlaufende Nummer um jede Anweisung zu identifizieren

- 'order\_id': Zusammenhängende Anweisungen, bzw. Anweisungen für die gleiche Box, haben eine gleiche Nummer
- 'relation': eine Liste mit 'instruction\_id's', welche vor dieser Anweisung erfüllt sein müssen

Bei Auswahl der zu schickenden Anweisung, wird zuerst die aktuell von der SPS gemachte Anweisung vermerkt, danach wird in allen Modulen des 'stacks' nachgeschaut, welche Anweisungen jetzt ausgeführt werden können. Zu diesem Zweck wird die 'relation' einer Anweisung mit den bereits gemachten Anweisungen verglichen, wenn bereits alle relevanten Anweisungen gemacht wurden, wird der Auftrag in 'to\_send' vermerkt. Schlussendlich wird einer der in 'to\_send' vermerkten Aufträge für die SPS übersetzt und an diese geschickt. Wenn die SPS den Auftrag erhalten hat, schickt sie ein 'acknowledge' mit der erhaltenen ID zurück. Der Auftrag wird dann aus den 'instructions\_to\_send' gelöscht und entweder wird einfach der nächste Auftrag in 'to\_send' geschickt oder der 'BMOS' Algorithmus wird erneut durchlaufen.

## 2.2.7 Datenbanken

Als Datenbanksystem wird aufgrund des guten Supports MySQL gewählt. Dies ist ein relationales Datenbankmanagementsystem welches in einem Docker-Container aufgesetzt wird. In diesem werden alle Daten gespeichert, die zur Auswahl sowie zur Auslieferung von Teilen nötig sind.

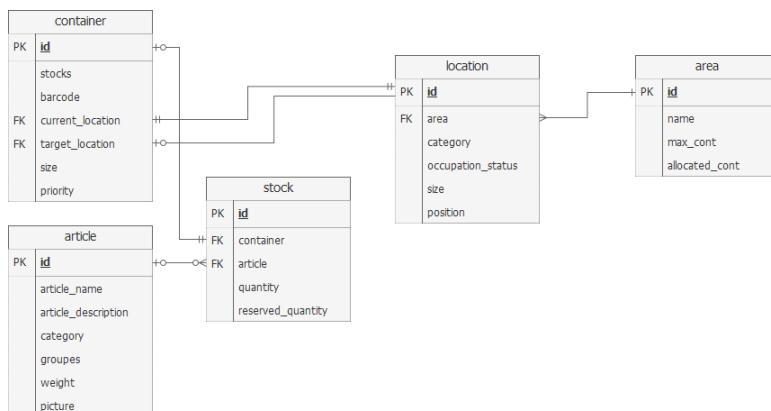


Abbildung 2.11: Datenbankschema des AFSS

Wie in 2.11 ersichtlich, beinhaltet diese Datenbank fünf Tabellen. Diese hohe Komplexität resultiert daraus, dass diese Struktur eine 100%ige Flexibilität in der Ablage von Bauteilen in einem überliegendem System bietet.

Die Erste Tabelle beschreibt ein einziges theoretisches Bauteil. Dieses hat einen Namen, Gewicht, Beschreibung und Kategorien zur Filterung. Unter der Spalte 'picture' wird ein Dateiname gespeichert, der zu einem Bild zeigt, dass das Produkt abbildet. Die zweite beschreibt einen Container. Im Lagersystem entspricht dieser einer Box. Diese kann mehrere 'stocks' beinhalten, sowie durch einen Barcode identifiziert werden. Weiters muss jeder Container immer eine aktuelle Position ('current\_location') besitzen, an der die Box gerade ist.

Im ausgelagerten (und noch nicht eingelagerten) Zustand ist diese 'location' Position 0. Das Ziel der Box wird in 'target.location' gespeichert. Stimmt die aktuelle mit der Zielposition überein, so ist die Box an ihrem Ziel angelangt. Die Kategorie 'size' beschreibt die Größe eines Containers und lässt somit theoretisch zu, dass in Zukunft auch unterschiedlich große Boxen zuverlässig in die richtigen Lagerplätze eingelagert werden. 'priority' wird nicht verwendet.

Container und Artikel werden im sog. 'stock' verheiratet. Dieser kann als bauteilhaufen in einer Box verstanden werden. Es können also auch mehrere 'stocks' mit dem selben Container geben, dies würde mehreren verschiedenen Bauteilen in einem einzigen Container entsprechen. Auch ist es möglich mehrere Container mit den selben 'stocks' abzubilden, welches eine Aufteilung von Abuteilen auf mehrere Container entspräche.

Die Positionen der Container werden in Standorte ('locations') abgebildet. Diese entsprechen den Lagerplätzen. Sie sind einer darüberliegenden 'area' zugeordnet, welche einerseits einen Lagerschrank, aber weiters auch Module wie Vereinzelungsanalgen, abbilden kann. Standorte verfügen weiters über eine Position welche in X, Y und Z-Richtung beschreibt, wo sich ein Standort im Referenzsystem des Lagers befindet. Auch die Größe des Lagerplatzes wird abgebildet, um sicherzustellen, dass auf jeden Fall nur die richtige Größe in Box eingelagert wird.

## 2.2.8 Docker

Docker ist eine Umgebung, in der Softwareprojekte isoliert werden können. Da besonders auch bei Projekten mit großem Python-Anteil, viele Pakete mit verschiedenen Versionen benötigt werden, ist es sehr hilfreich diese zu bündeln.

Umgesetzt wird dies mithilfe von Containern welche einen gesamten Programmteil als alleinstehende Einheit enthält. Diese werden über ein 'Dockerfile' konfiguriert, welches sich im selben Ordner wie die Python-Anwendung befindet. In diesem werden Parameter wie die Python-Version und die benötigten pip-Pakete sowie der Programmeinstiegspunkt angegeben.

Ein zweiter Docker Container wird mit einem MySQL-Image erstellt, dort wird die Datenbank aufgesetzt.

Um diese zwei Container miteinander kommunizieren zu lassen, ist es nötig ein sog. docker-compose.yml File zu erstellen. Dies enthält alle Informationen über verwendete Container, deren Ports, sowie Speicher für Dateien (Volumes). Bei Testbetrieb wird der Datenbankcontainer alleinstehend betrieben und mit einem anderen Port, keine Zugriffsprobleme zu generieren. In Produktion wird dann derselbe Container in den Containerverband übertragen und dort mit einem anderen Port weiterverwendet.

Erstellt wird dieser Containerverband mit den Consolenbefehlen der auf das docker-compose File zugreift.



Abbildung 2.12: Docker-Logo: [14]

```
1 docker build docker-compose.yml
```

Dann werden auch alle Logs in der Kommandozeile ausgegeben sowie

## 2.2.9 Artikelsuche

### TF-IDF und Rust Implementierung

Der TF-IDF (Term Frequency-Inverse Document Frequency) Algorithmus, ist ein Weg um wichtige Wörter aus Dokumenten zu extrahieren. Er wird verwendet um beispielsweise in Suchmaschinen, eine Suchanfrage mit Webpagecontent abzugleichen, und die am besten mit der Suchanfrage übereinstimmenden Dokumente zu sortieren.

Im Fall dieser Anwendung werden die Daten aus der Artikeldatenbank als 'Dokumente' angesehen und die Suchanfrage aus dem Suchfeld wird dafür verwendet um die am besten passenden Artikel zu finden.

Durch den Relativ hohen Rechenaufwand bei dieser Suchoperation wird dieser in der Programmiersprache Rust implementiert. Die Implementierung in Rust ist im Vergleich zu Python schon bei relativ kleinen Datenmenge bis zu 5-mal schneller.

### Rust

Rust ist eine sehr effiziente und schnelle Programmiersprache die in den späten 2000er und frühen 2010ern bei Mozilla und der Open-Source-Community entwickelt. Sie unterstützt unter anderem mehr Typsicherheit und verhindert viele Programmierfehler.[12]

Die Funktion dieses Algorithmus ist in drei Unterteile Unterteilt.

#### 1. Term Frequenz

Die Termfrequenz gibt an, wie oft ein angegebenes Wort in einem Dokument vorhanden ist. Dies wird durch die folgende Funktion kalkuliert.

```
1 fn term_frequency(document: &str, term: &str) -> f64 {
2     // Store the lowercase document as a String to ensure it lives long
3     // enough
4     let lower_document = document.to_lowercase();
5
6     // Split the document into words
7     let normalize_document: Vec<&str> = lower_document.split_whitespace()
8         .collect();
9     // Make sure the searchterm is lowercase
10    let normalize_term = term.to_lowercase();
11
12    // Count occurrences of the term in the document
13    let count = normalize_document
14        .iter()
15        .filter(|&&word| word == normalize_term) // Compare each word
with the term
        .count();
```

```
16     // Calculate the term frequency as occurrences / total number of
17     words
18
19     let total_words = normalize_document.len();
20     if total_words == 0 {
21         0.0 // Avoid division by zero if the document is empty
22     } else {
23         count as f64 / total_words as f64
24     }
25 }
```

Mithilfe dieser wird eine Liste aller Wörter und der Vorkommenshäufigkeit dieser erstellt.

2. Die zweite Komponente ist dann die Inverse Dokument Frequenz. Diese gewichtet, die Anzahl der Dokumente in dem das gesuchte Wort enthalten ist relativ zur Gesamtdokumentanzahl vorkommt. Häufig vorkommende Worte wie z.B. 'und' werden hierbei weniger gewichtet als einzigartige Wörter.

```
1 fn inverse_document_frequency(term: &str, all_documents: &Vec<String>)
2     -> f64 {
3
4     let mut num_documents_with_this_term = 0;
5
6     // Iterate over all documents to check if they contain the term
7     for doc in all_documents {
8         // Normalize both term and document by converting them to
9         lowercase
10        let lower_doc = doc.to_lowercase();
11        let normalized_doc: Vec<&str> = lower_doc.split_whitespace().
12            collect();
13
14        // Check if the term exists in the document
15        if normalized_doc.contains(&term.to_lowercase().as_str()) {
16            num_documents_with_this_term += 1;
17        }
18
19        // Calculate IDF
20        if num_documents_with_this_term > 0 {
21            // Apply the IDF formula: 1 + log(total_documents /
22            documents_with_term)
23            1.0 + ((all_documents.len() as f64) / (
24                num_documents_with_this_term as f64)).ln()
25        } else {
26            // If the term is not found in any document, return 1.0
27            1.0
28        }
29    }
30 }
```

3. Nun liegt Liste davon vor, wie oft ein Wort in den Suchdaten vorkommt, als auch, wie oft ein Suchbegriff in einem bestimmten Dokument ist.

Als nächsten Schritt werden diese beiden Werte für jeden Suchbegriff miteinander multipliziert und ergeben somit einen Vektor der die Suchwörter in Relation zu jedem einzelnen Dokument stellt.

4. Als letzten Schritt wird der zuvor errechnete Dokumentenvektor (der IDF jedes Suchterms in jedem Dokument) mit dem Suchvektor verglichen. Die geschieht mit der sog. Kosinus-Ähnlichkeit.

```
1 fn cos_similarity(query_p: Vec<f64>, document_p: Vec<f64>) -> f64 {  
2     // Ensure that both vectors have the same length  
3     if query_p.len() != document_p.len() {  
4         return -1.0;  
5     }  
6  
7     let mut dot_product = 0.0;  
8     let mut abs_doc_squared = 0.0;  
9     let mut abs_query_squared = 0.0;  
10  
11    // Calculate the dot product and the magnitudes (squared)  
12    for i in 0..query_p.len() {  
13        dot_product += query_p[i] * document_p[i];  
14        abs_doc_squared += document_p[i].powi(2); // document_p[x] ** 2  
15        abs_query_squared += query_p[i].powi(2); // query_p[x] ** 2  
16    }  
17  
18    // Calculate the magnitudes  
19    let abs_doc = abs_doc_squared.sqrt();  
20    let abs_query = abs_query_squared.sqrt();  
21  
22    // Handle division by zero in case of zero vectors  
23    if abs_doc == 0.0 || abs_query == 0.0 {  
24        return 0.0;  
25    }  
26  
27    // Return the cosine similarity  
28    return dot_product / (abs_doc * abs_query);  
29 }
```

Nach der Berechnung dieser für jedes Dokument werden alle Dokumente sortiert und ja nach Anforderung die benötigte Anzahl ausgegeben.

## Artikelsuche nach Kategorien

Um auch einen simpleren Weg der Artikelfindung zur Verfügung zu stellen, wird außerdem die Möglichkeit implementiert, Artikel anhand von Attributen zu suchen. Zu diesem Zweck werden schon bei der Artikelerstellung Attribute für die Einträge 'Gruppen' und 'Kategorien' vergeben. In 'Gruppen' wird eine Art Pfad angelegt, der die Suche eingrenzt. Beispielsweise würde so ein Eintrag folgende Daten enthalten: ["Item", "Verbindungssatz"] enthalten. So kann bei der Artikelsuche zuerst die Überkategorie Item und dann die Unterkategorie

"Verbindungssatz aus mehreren verschiedenen ausgewählt werden. Um Bauteile weiter zu unterscheiden, da es z.B. viele verschiedene Widerstände gibt, werden unter 'Kategorien' Einzelheiten zum Produkt, wie Wert, Farbe o. ä. gespeichert.

Diese Informationen werden auch vom TF-IDF verwendet, dienen aber spezieller dazu, möglichst sicher das gewünschte Bauteil in einem Durchklickmenü zu finden.

### 3 Integration und Programmierung der Steuerungstechnik (Vincent Sonvilla)

#### 3.1 Aufgabenstellung

#### 3.2 Tia-Portal Grundlagen

Tia Portal (Totally Integrated Automation Portal) ist die zentrale Software von Siemens zur Programmierung, Konfiguration und Diagnose von Automatisierungssystemen. Es ermöglicht die Steuerung von SPS (Speicherprogrammierbare Steuerungen), HMI (Bedienpanels) und Antrieben in einer einzigen Umgebung.

##### 3.2.1 Programm erstellen

##### 3.2.2 Programmbausteine

In TIA Portal werden Programmbausteine genutzt, um Steuerungsprogramme modular und strukturiert zu gestalten. Dadurch werden Programme übersichtlicher, wiederverwendbar und effizienter. Es gibt unterschiedliche Arten von Programmierbausteinen:

###### 1. OB(Organisationsbausteine)

Organisationsbausteine werden verwendet um das Anwenderprogramm hierarchisch zu strukturieren. Auch für OBs stehen, wie in Abb. 3.1 gezeigt, unterschiedliche Bausteine zur Verfügung:

Organisationsbaustein Typ	S7-1200	S7-1500	Nutzen
Zyklische und Anlauf OBs	100	100	Modularisierung des Anwenderprogramms
Prozessalarme	50	50	Separater OB je Ereignis möglich
Verzögerungsalarme	4*	20	Modularisierung des Anwenderprogramms
Weckalarme		20	Modularisierung des Anwenderprogramms
Uhrzeitalarme	*	20	Modularisierung des Anwenderprogramms

Abbildung 3.1: Organisationsbausteine [15]

Organisationsbausteine steuern unterschiedliche Vorgänge:

- Anlaufverhalten der Steuerung
- Zyklische Programmbearbeitung

- Alarmgesteuerte Programmbearbeitung
- Behandlung von Fehlern

Werden in einem Programm mehrere OBs aufgerufen, so werden die OBs in aufsteigender Reihenfolge der OB-Nummer abgearbeitet. [15]

## 2. FC(Funktionen)

Funktionen haben keinen zyklischen Datenspeicher, deswegen können Bausteinparameter nicht bis zum nächsten Aufruf gespeichert werden. Darum müssen Funktionen bei jedem Aufruf mit Aktualparametern versorgt werden. Um kein zufälliges Verhalten entstehen zu lassen sind die Werte immer mit einem Defaultwert vorbelegt. Will man die Daten einer Funktion dauerhaft speichern, so muss man einen globalen Datenbaustein verwenden.

Funktionen werden verwendet, um häufig wiederkehrende Anwendungen durchzuführen. [15]

## 3. FB(Funktionsbausteine)

Im Gegensatz zu Funktionen haben Funktionsbausteine einen zyklischen Datenspeicher -Instanz DB-, in welchem Werte dauerhaft gespeichert werden. Dadurch behalten statische Variablen ihren Wert von Zyklus zu Zyklus. Wie bei Funktionen sind die Werte mit einem Defaultwert vorbelegt.

Funktionsbausteine können genutzt werden, um Unterprogramme für unterschiedliche Anwendungen zu erstellen. Außerdem werden sie genutzt um das Anwenderprogramm zu strukturieren. Bei mehrfacher Verwendung von Funktionsbausteinen empfiehlt sich die Verwendung von Multiinstanz-DBs. [15]

## 4. Global-DB(Datenbausteine)

Globale Datenbausteine speichern variable Daten, die dem kompletten Programm zur Verfügung stehen. Wie in Abb.3.2 ersichtlich, bedeutet das ,dass alle Bausteine Zugriff auf den Global-DB haben.

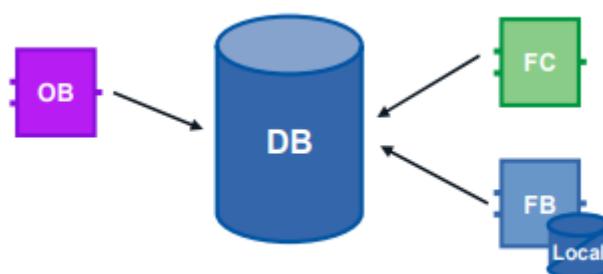


Abbildung 3.2: Zugriff auf Global-DB [15]

In Globalen Datenbausteinen können jegliche Datentypen genutzt werden.

Globale DBs werden verwendet, wenn Daten in verschiedenen Programmteilen bzw. Bausteinen benötigt werden. [15]

### 3.2.3 Technologieobjekte

### 3.2.4 Programmiersprachen

### 3.2.5 Librarys

## 3.3 Motorenansteuerung

## 3.4 SPS-Server Kommunikation

### 3.4.1 Zur Auswahl stehende Kommunikationsprotokolle

Kommunikationsprotokolle ermöglichen den Datenaustausch zwischen unterschiedlichen Systemen, indem sie Standards und Regeln für die Kommunikation definieren. In diesem Projekt wurden zwei Protokolle getestet und miteinander verglichen: OPC-UA sowie das HTTP-Protokoll.

#### Allgemeines

- **HTTP (Hypertext Transfer Protocol):**

HTTP ist eines der bekanntesten Protokolle, welches für die Datenübertragung zwischen Clients und Servern verwendet wird. Es basiert auf einem Anforderungs-Antwort-Prinzip, bei dem ein Client (Bsp.: Webbrowser) Anfragen an einen Server sendet, welcher anschließend die entsprechenden Daten zurückschickt. Die Anfrage wird als HTTP Request und die Antwort als HTTP Response bezeichnet.[16]

- **OPC-UA (Open Platform Communications - Unified Architecture):**

OPC-UA (Open Platform Communications Unified Architecture) ist ein plattformunabhängiges Kommunikationsprotokoll, das speziell für industrielle Anwendungen entwickelt wurde. Es ermöglicht eine herstellerunabhängige Kommunikation zwischen verschiedenen Geräten bzw. Systemen. [17]

#### Funktionsweise

- **HTTP (Hypertext Transfer Protocol):**

- **OPC-UA (Open Platform Communications - Unified Architecture):**

### 3.4.2 Verbindungsherstellung

Aus den in Punkt 3.4.1 genannten Gründen wurde das HTTP Protokoll ausgewählt. Um in TIA-Portal die Verbindung via HTTP aufzubauen, benötigt man bestimmte Libraries die von Siemens zu Verfügung gestellt werden. Diese müssen dann wie im Punkt 3.2.5 gezeigt eingebunden werden, um die Funktionsbausteine der Library nutzen zu können.

## Funktionsbausteine

In der Library stehen dann folgende Bausteine zur Verfügung:

- GET
- POST-PUT

Mit dem POST-PUT Befehl werden Daten zum Server geschickt aber auch Daten erhalten.

### 3.4.3 Datenfilterung

Die vom Server geschickten Daten werden in einem Befehl geschickt. Aus diesem Befehl muss herausgelesen werden um welche Aufgabe es sich handelt, und die Daten die erforderlich sind um diesen Befehl auszuführen.

#### Datenformatierung

Die Daten werden in einem String geschickt, welcher in zwei Teile aufgeteilt wird. Der zweite Teil ist jedoch abhängig vom ersten.

- 1.Teil:

IDXXXXXXX

Aus diesem Teil werden die ID-Nummer sowie der Auftrag herausgefiltert. Die ID-Nummer ist eine 4 stellige Nummer welche nach ID steht. Der Auftrag welcher ausgeführt werden muss steht in den zwei Stellen nach A.

Diese werden nach folgender Codierung ausgelesen:

- 00: Kommissionierstation
- 01: Förderband
- 10: Lager 1 (Aus-/Einlagerung)
- 11: Lager 1 (Querförderer)

- 2.Teil:

Der zweite Teil steht in Abhängigkeit zu ersten. Je nachdem welche Zahl nach A steht, also der Code welche Area angesprochen wird, ist der zweite Teil anders aufgebaut.

- 00:
- 01:

- 10: X-Position, Y-Position, Z-Position, Ein-/Auslagerung

Wobei nach jedem Symbol eine 4 stellige Zahl steht. Also X0000Y0000Z0000R0 bedeutet, dass die X-Position 0, Y-Position 0, Z-Position 0 und es sich um eine Auslagerung handelt.

- 11:

### 3.5 Herausforderungen

## 4 Elektroplanung und Realisierung (Nikolaj Voglauer)

### 4.1 Elektroplanung

#### 4.1.1 Einleitung - Grundanforderungen

Die grundsätzliche Zielsetzung bei der elektrischen Planung, war die Anforderungen so zu erfüllen, dass die Lösung einerseits die Anforderungen von Erweiterbarkeit und Mobilität erfüllen und andererseits in der Schule beziehungsweise in der Werkstatt produzierbar waren. Weiterführend sollte die Umgebung im Serverschrank beachtet werden. Darunter fällt, dass die Module in die Breite von den, nur in die Tiefe verstellbaren, Profilschienen begrenzt werden.

In der Anlage sollten während dem Normalbetrieb alle Komponenten vor elektrischen Störungen geschützt sein. Der Fokus liegt hierbei auf dem Schutz von Messleitungen und Steuerleitungen, an diese gibt es besonders hohe Anforderung bezüglich Präzision.

Weiterführend sollte in der Planung stets bedacht werden, dass die elektrischen Komponenten so verbaut werden, dass im Falle eines Fehlers sowohl Personen gut geschützt sind und dass die Geräte leicht auszuwechseln sind.

#### 4.1.2 Elektrik spezifische Anforderungen

##### Versorgung

Zur Verfügung steht dem AFSS eine 3-phägige Wechselspannung mit 400V Außenleiterspannung. Damit direkt angesteuert werden kann nur der Asynchronmotor für das Fließband. Alle anderen Elemente brauchen eine andere Spannungsebene. Die in Summe sieben Schrittmotoren brauchen 24 V mit einem möglichen Dauerstrom von über 20A. Die Logik bestehend aus Siemens-SPS mit verschiedensten Karten und einer ET200 mit Asi-Master. Diese benötigen ebenfalls 24 V und sollen getrennt versorgt werden, um bei Fehlern geschützte Logikkreise zu haben. Der Asi-Kreis benötigt eine eigene Asi-24V-Versorgung.

##### Ansteuerungen

Angesteuert werden müssen 8 Motoren: 1 Asynchronmotor (250 W), 4 stärkere Schrittmotoren (2 Nm) und 3 schwächeren Schrittmotoren (40 Ncm).

Der Asynchronmotor soll keine Drehzahlregelung haben und über eine Wendeschützschaltung angesteuert werden. Die Schrittmotoren sollen über Schrittmotortreiber angesteuert werden. Diese Treiber werden von den PTO-Karten der SPS angesteuert.

##### Sicherheit

Für die Anlage soll ein Fehlerstromschutzschalter, ein Leitungsschutzschalter, ein Motorschutzschalter und für jeden Motor eine Gleichstromsicherung ausgelegt werden.

Um Fehler zu behandeln die potentiell von den elektrischen Schutzeinheiten nicht unterbrochen werden soll die Anlage über mehrere Not-Aus-Schalter verfügen. Zwei auf der Anlage, einer im Serverschrank/Schalschrank und einer am Kommisionierplatz. Diese Positionierung soll es NutzerInnen ermöglichen aus jeder Position an der Anlage einen Not-Aus-Schalter zu erreichen.

### **Bedienelemente**

An physischen Bedienlementen sollen ein Schlüsselschalter zur Freigabe und ein dreiphasiger Drehstromschalter für eine manuelle Freischaltungsoption eingeplant werden.

### **Schalschrank**

Grundsätzlich haben Schalschränke genormte Anforderungen.

Dazu gehört eine Auslegung von Kabelkanäle, die die Kabel schützen soll und Umbauten nicht zusätzlich erschweren sollen. Freifliegende Kabel sollen unter allen Umständen verhindert werden. Das Gehäuse muss geerdet sein und die inneren Komponenten vor Staub und Schmutz schützen. Bei einem potenziellen Lichtbogen soll der Schalschrank Personen in der Nähe schützen. Zudem muss der Schrank gegen thermische Einflüsse geschützt sein, gegebenenfalls soll der Schalschrank über eine Belüftung verfügen.

Der Serverschrank schützt gegen Staub und Schutz und kommt mit einer Lüfteranlage, die die Abwärme von mehreren Gleichrichtern gut abführen kann. Zudem sind die Materialen des Schrankes vor korrosionsgeschützt.

Bei der Planung muss beachtet werden, dass die Erdung aller leitungsfähigen Elemente eingehalten wird. Außerdem dürfen Umbauten wie die Montage von Rädern keine der angeführten Anforderungen widersprechen.

### **Kabelauslegung**

Bei den Kabeln gibt es mehrere Punkte, die beachtet werden müssen beim Auslegen. Während Spannungsabfall bei den Längen des AFSS vernachlässigt, werden können muss besonders auf Schleppkettentauglichkeit geachtet werden. Steuer- und Messkabel müssen entsprechend geschirmt werden und entsprechend dem Strom muss der Querschnitt gewählt werden. Dabei sind die Querschnitte aber auch stark abhängig von den Schutzeinheiten im Schaltkreis.

### **Module**

Die Paneele/Module, auf welchen die elektrischen Komponenten montiert werden sollen, müssen ebenfalls alle Erdungserwartungen erfüllen und mechanisch den Belastungen standhalten. Dabei ist das Gewicht die beachtlichste Belastung. Eine gerechte Drahtverlegung muss gewährleistet sein und die Modularität der Paneele soll vorteilhaft ausgenutzt werden und sollen nicht das Projekt unnötig verkomplizieren. Kostentechnisch soll dabei ein möglichst billiges, aber standhaftes Material gewählt werden.

### 4.1.3 Mechanische Planung

#### Modulprinzip

Es wurden bereits die Anforderungen an die Module beschrieben. Doch es gäbe noch weitere Alternativen für den Innenraum des Serverschrances, so könnte man eine große Platte verwenden und diese an die Profilschienen festschrauben. Eine weitere Option wäre eine plattenlose, dabei würde man die Hutschienen direkt auf die Profilschienen des Serverschrances montieren.

Die große Platte entfällt als Möglichkeit insofern, da diese nicht in der Schule produzierbar gewesen wäre. Die plattenlose Option wäre eine kosteneffiziente Möglichkeit, allerdings gibt es viele Elemente, die im Schaltschrank nicht auf Hutschienen montiert werden können, diese bräuchten immer eine Montageplatte.

Damit ein einheitliches Design eingehalten werden kann, haben wir uns für das Modulprinzip entschlossen. Dieses ermöglicht es Elementen, die für Hutschienen ungeeignet sind, wie dem 24V-ASI-Gleichrichter, montiert zu werden und ist weiterhin in der Schule produzierbar.

#### Platten-Material

Für die Materialwahl gab es zwei realistische Möglichkeiten. Die Modulplatten hätten vollständig aus Aluminium gefräst werden können oder aus Dibond. Die Aluplatten bieten den Vorteil der Leitfähigkeit und somit müsste man nur die Platte erden und die Elemente auf der Platte wären alle dementsprechend geerdet. Doch Aluminium ist teuer und ein wertvoller Werkstoff, da ein umsichtiger Umgang mit Ressourcen wichtig ist wurden dann doch die Dibond-Platten als Projektstandard definiert. Diese sind zwei dünne Platten aus Aluminium die auf einen Kunststoff aufgepresst werden. Dibond bietet keine elektrische Leitfähigkeit, folglich müssen alle Elemente zusätzlich geerdet werden.

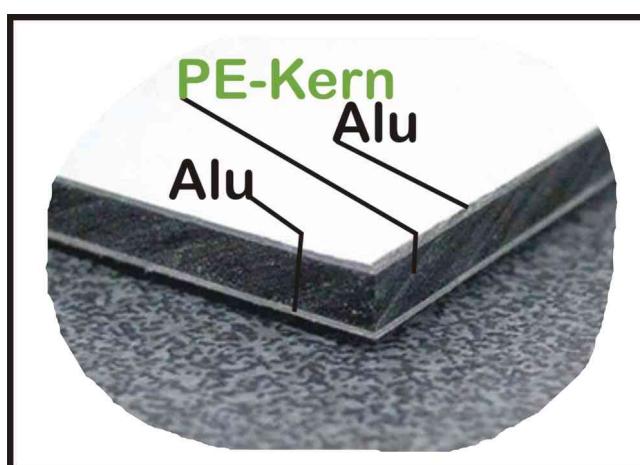


Abbildung 4.1: Dibond-Platte, Quelle: [18]

## Digitaler Zwilling

Moderner Schaltschrankherstellung begegnen im Herstellungsprozess oft große logistische Probleme. Jeder Prozessschritt ist eine Fehlerquelle und wenn Fehler nicht früh erkannt werden, pflanzen sich diese fort. Damit zwischen den Prozessschritten keine Kommunikationsprobleme entstehen setzen viele Hersteller auf das Prinzip des digitalen Zwilling. Dieser im Grunde ein digitaler Schaltschrank, welcher im ersten Prozessschritt, der Planung, ausgeplant wird und im Herstellungsprozess, sei es der Schrankbau oder die Bestückung, wird einerseits immer derselbe digitale Zwilling aktualisiert und aber auch referenziert. Das heißt alle Prozessschritte beziehen sich auf denselben Plan bzw. digitalen Zwilling (siehe 4.2).

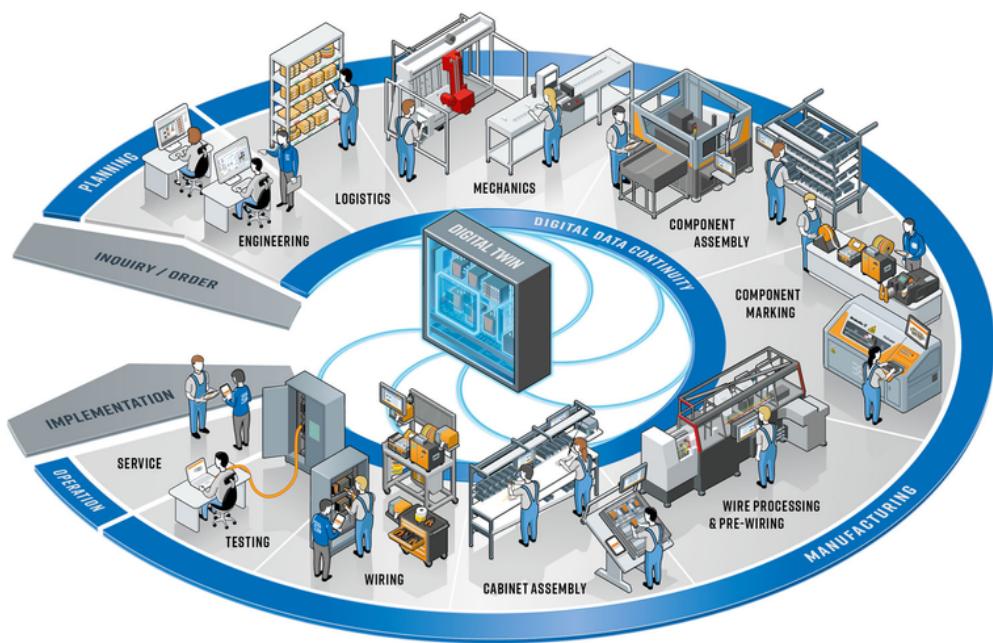


Abbildung 4.2: Digitaler Zwilling, Quelle: [19]

Es setzt auch ein breites Feld an Firmen auf dieses Prinzip. Firmen wie Weidmüller, Komax, Steinhauer und noch viele mehr haben eine Firmenzusammenarbeit, die ohne einen digitalen Zwilling nicht möglich wäre[20]. In diesem Fall werden die jeweiligen Prozessschritte meistens von einer neuen Firma übernommen, in diesem Bündnis ist der digitale Zwilling der Schlüssel zum Erfolg. Man kann dieses Prinzip der Dokumentation bzw. Planung als Industriestandard verstehen.

Um den Prozess der Herstellung des Schaltschranks möglichst nahe an die Praktiken aus der Industrie anzugeleichen, wird auch der Schaltschrank des AFSS mithilfe eines digitalen Zwilling geplant. Dieser wird in Fusion360 gezeichnet und soll den Sollzustand des Schaltschranks abbilden.

Um den die Konstruktion anzufangen, braucht es eine möglichst ausführliche Ausmessung

des bereits bestehenden Serverschrances. Besonders wichtig sind die Elemente die direkt am Umbau beteiligt sind, wie die Profilschienen (Abstände der Löcher, Abstände der Profilschienen zueinander und detaillierte Abmessungen der Profilschienen selbst.), die Türen und die Lüfter.

### Digitaler Zwilling - Schritte

Als Erstes muss der Serverschrank, wie bereits erwähnt ausführlich ausgemessen werden. Äußere Höhe, innere Höhe, Äußere Breite, innere Breite und noch Vieles mehr muss richtig gemessen werden. Bei den Messungen werden Messschieber und bei größeren Abständen Maßbänder verwendet. Um die mechanische Konstruktion zu erleichtern, werden alle Daten digital festgehalten.

Während die Messungen des Serverschranks für den finalen digitalen Zwilling wichtig werden, gibt es aber auch noch andere Punkte, beispielsweise bestand lange die Frage, ob das Modulkonzept so möglich sei. Aufgrund dessen und des Umfangs der Diplomarbeit sowie der begrenzten Zeit wird ein erster Entwurf eines Serverschrances in Fusion360 konstruiert und weiters ein Probemodul gezeichnet. Die Maße dieses digitalen Prototyps sind von einem Standard-Serverschrank aus dem Internet übernommen. Dieser Prototyp hat nicht dieselben Werte wie der richtige Serverschrank, der dem AFSS zur Verfügung steht. Aufgrund der Prototyp-Konstruktion steht fest, dass das Modulkonzept ist so umsetzbar. Weiterführend ist festgestellt, wie man optimierter in Fusion Zeichnen kann. Eine Erkenntnis des Prototyps ist, dass man den Serverschrank nicht als ein großes Element konstruieren sollte, da wenn ein Fehler spät erkannt wird dieser so gut wie nicht mehr zu beheben ist. Wenn die Konstruktion allerdings auf viele verschiedene Elemente aufgeteilt wird, dann ist der Schaden bei einem Fehler begrenzt.

Nachdem das Grundprinzip erfolgreich konstruiert wurde, ist der Serverschrank auf Grundlage der echten Maße zu konstruieren. Die gelernten Erkenntnisse werden dabei bestmöglich miteinbezogen.



Abbildung 4.3: Sommerprototyp eines Serverschrances

Mit dem fertig konstruierten Serverschrank, kann dann die Planung der Module beginnen. Dafür wird vorerst ein Konzept erstellt. Durchgedacht wird welche elektrischen Baugruppen wo im Schaltschrank platziert werden sollen. Das Konzept richtet sich einerseits nach der Vorgabe, zusammengehörige Elemente sollen auf dasselbe Modul kommen, und andererseits zusammenhängende Module sollen sich möglichst nahe sein.

### Modul 1 - Bedienelemente

Es sollen im Schaltschrank ein Notaus, ein Schlüsselschalter und ein Drehstromschalter verbaut werden. Diese Elemente werden gemeinsam auf einem Modul verbaut, da sie alle Bedienelemente sind.

### Modul 2 bis 3 - Schutzorgane und Versorgungen

Direkt unter dem Drehstromschalter sollen die Schutzorgane liegen. Da der Leitungsschutzschalter und der Fehlerstromschutzschalter wenig Platz benötigen, werden auf dem 2. Modul zusätzlich die zwei Gleichrichter verbaut die auf eine Hutschiene montierbar sind. Damit besteht das 2. Modul aus den Schutzorganen und den Gleichrichtern. Auf die Platte muss damit auch eine durchgängige Hutschine und ein Verdrahtungskanal geplant werden. Für die reguläre Versorgung des AFSS werden drei normale Gleichrichter benötigt. Der dritte, der Deutronic Gleichrichter, kann nicht auf eine Hutschine montiert werden und hat zudem einen großen Platzbedarf. Dieser wird auf einem eigenen Modul verplant. In den Leerräumen des 3. Moduls werden Hutschinen mit Reihenklemmen verplant. Diese Reihenklemmen sollen für eine übersichtliche Verdrahtung der Versorgungsleitungen sorgen.

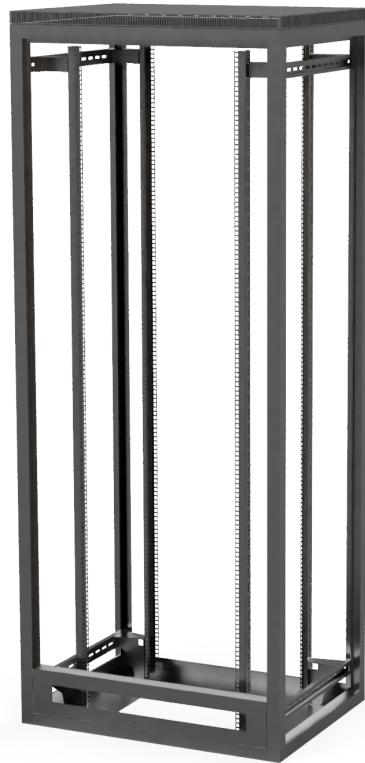


Abbildung 4.4: Serverschrank konstruiert

### Modul 4 - ASI-Elemente

Konzepttechnisch ist das 4. Modul ein Erweiterungsmodul. Nur eine Ein/Ausgangsbaugruppe von Weidmüller ist auf eine kleine Hutschine verplant gewesen und der restliche Platz sollte freigelassen werden für potentielle Erweiterungen.

Im Entwicklungsprozess ist deutlich geworden, dass die PWM-Signalerzeugung, der Ausgangsbaugruppen, nicht fähig sind ein veränderbares PWM-Signal zu erzeugen. Damit werden diese Elemente nicht mehr benötigt. Parallel ist verstanden worden, dass es eine getrennte 24V-Versorgung für den ASI-Kreis geben muss. Deswegen ist im ursprünglichen

Erweiterungsbereich dieses Modules, ein 24V-ASI-Gleichrichter verplant, der keine Hutschiene benötigt, und anstatt der Weidmüllerkomponenten kommt eine ET200, an die ein ASI-Master angeschlossen ist. Zudem ist auch ein Verdrahtungskanal nötig.

### **Modul 5 - SPS und Sicherungen**

Das 5. Modul beinhaltet die Siemens SPS und die gesamten DC-Sicherungen für die Schrittmotoren. Für die SPS muss eine Siemens-Profilschiene auf die Platte und für die Sicherungen eine durchgängige Hutschiene und einen Verdrahtungskanal.

### **Modul 6 - Schrittmotoren**

Für die stärkeren Schrittmotoren werden Treiber verwendet die direkt auf einen Untergund montiert werden müssen. Die vier Treiber werden auf eine eigene Platte montiert. Auch auf diesem Modul ist ein Verdrahtungskanal nötig.

### **Modul 7 - Ausgangsmodul**

Am Ausgangsmodul soll einen Verdrahtungskanal haben und eine durchgängige Hutschiene. Auf diese Hutschine kommen Elemente wie die Motorschütz, Relais, die Treiber für die schwächeren Schrittmotoren und eine ausführliche Menge an Reihenklemmen.

### **Modul 8 - Erdungsmodul**

Etwaige Kabel des AFSS haben einen Schirm der geerdet gehört. Deswegen ist eine Erdungsplatte nötig die aus leitfähigen Aluminium gemacht werden soll. Auf dieser ist eine Ankerschiene geplant. Zudem sollen in dieses Modul vier rechteckige Ausfrässungen gemacht werden, in welche der nichtsteckbare Teil der RJ45-Stecker hineinpasst.

-Module: Ist der Schaltchranc fertig übernommen aus der realität kann man anfangen mit einem Modul. Dabei fällt einem zum Beispiel schon eine sache auf, sowie die Tatsache, dass wenn man die Profilschinen an der selben Position lässt würden sich die Türen nicht schließen lassen. Folge daraus ist, die Profilschinen müssen verschoben werden. Der Aufbau des Serverschranks lässt dies zu.

-Weitere Module: diese werden dann auch in Fusion360 gezeichnet und dann in den digitalen Zwilling eingefügt. Dabei kann man schön erkennen wie die Module Anneinander gereiht gehören, welche Reihenfolge sinnvoll ist und ob sich die Menge an Elektrischen Einheiten auch ausgeht mit den Modulen.

-Dieser soll auch immer die aktuelle version abbilden

## **4.2 Realisierung**

## 5 Sensorik und Sicherheitstechnik (Elena Widmann)

### 5.1 Aufgabenstellung

### 5.2 Sensorik

#### 5.2.1 Endschalter

Beim Verplanen der Endschalter ist zwischen Software- und Hardware-Endschalter zu unterscheiden. Die Software-Endschalter begrenzen den Arbeitsbereich der Achse und sollten innerhalb des Bereichs der Hardware-Endschalter parametriert werden. Ihre Positionen werden direkt im Siemens TIA-Portal eingestellt und können falls notwendig einfach auf die aktuelle Geschwindigkeit angepasst werden. Werden die Software-Endschalter angefahren, wird der Technologiealarm 533 ausgelöst, und die Dynamikwerte werden gestoppt, das Technologieobjekt bleibt hierbei freigegeben. Werden sie jedoch überfahren wird das Technologieobjekt gesperrt.

Die Hardware-Endschalter begrenzen den maximal zulässigen Verfahrensbereich der Achse. Bei ihnen wird nicht unterschieden, ob die Endschalter angefahren oder überfahren werden. Beim Anfahren der Schalter wird der Technologiealarm 531 ausgelöst. Er sperrt das Technologieobjekt und muss, bevor der Auslösebereich der Hardware-Endschalter wieder verlassen werden kann, quittiert werden. [21]

Auf jeder der drei Achsen vom AFSS und auf dem Querförderer müssen Hardware-Endschalter montiert werden. Die Auswahl begrenzte sich hierbei auf die uns zur Verfügung gestellten Sensoren, welche unter Berücksichtigung ihrer Funktion auf den verschiedenen Positionen eingebaut werden.

#### Positionsschalter mit Rollhebel

An der x-Achse werden als Hardware-Endschalter Positionsschalter mit Rollhebel verwendet (siehe Abb. 5.1a). Von den insgesamt vier Stück werden zwei an der unteren und zwei an der oberen x-Achse befestigt. Davon besitzen drei jeweils einem Öffner- und einen Schließerkontakt [22], wohingegen einer der Endschalter aus zwei Öffnerkontakte besteht [23]. Um Einheitlich zu bleiben, und da es sicherheitstechnisch auch von Vorteil ist (Drahtbruchsicherheit), verwenden wir jeweils einen der Öffnerkontakte der Endschalter. Zum Schalten des Rollhebels der Positionsschalter müssen auf dem x-Schlitten der oberen sowie unteren x-Achse Auslöser angebracht werden. Diese befinden sich mittig auf der Seite der Sensoren und gleichen einem vom Schlitten abstehenden Arm, welcher sich aus gestapelten, mit dem Lasercutter gefertigten, Teilen zusammensetzt.

#### Induktive Endschalter

Als Hardware-Endschalter an der y-Achse werden induktive Sensoren verwendet (siehe Abb. 5.1b). Davon werden zwei an der unteren und zwei an der oberen Seite der y-Achse befestigt, also auch hier wieder insgesamt vier Sensoren. Sie funktionieren so, dass durch

eine Spule ein Magnetfeld erzeugt wird, welches dann in einem sich dem Sensor frontseitig nähern elektrisch leitendem Material Wirbelströme erzeugt. Dadurch verändert sich das Magnetfeld und die Kontakte des induktiven Sensors werden über einen Schmitt-Trigger geschaltet. Die Sensoren besitzen jeweils einen Öffner- und einen Schließerkontakt, wir verwenden jedoch ersteres um Drahtbruchsicherheit zu gewährleisten. Damit die induktiven Sensoren korrekt auslösen können, müssen auf dem Shuttle der y-Achse elektrisch leitende Gegenstücke, in unserem Fall aus Metall, angebracht werden.

### Endtaster

An der z-Achse und am Querförderer werden jeweils zwei Stück mechanische Endtaster als Endschalter verwendet. Auf einem Endtaster befindet sich ein Schließerkontakt in Form eines Tasters, welcher durch anfahren geschalten wird (siehe Abb. 5.1c). Zum Betätigen der Taster müssen sich Auslöser auf dem Shuttle und den Seiten der Querfördererstation befinden.



(a) Rollendschalter [24]



(b) Induktiver Sensor [25]



(c) Endtaster

Abbildung 5.1: Endschalter

### 5.2.2 Referenztaster

Um die Motoren auf die richtige Position fahren zu können, müssen diese an allen drei Achsen und am Querförderer referenziert werden. Somit wird verhindert, dass auch bei einem Neustart des Systems sich die Koordinaten der Positionen, auf denen sich die Bauteilboxen befinden, nicht verändern und es zu keiner Kollision zwischen Shuttle und einer Box kommt. Zum Referenzieren müssen Sensoren an den Achsen und am Querförderer angebracht werden, welche den jeweiligen Nullpunkt angeben. Hierfür werden Opto Interrupter verwendet, da diese einfach durch anfahren ausgelöst werden können. In einem Opto Interrupter befindet sich eine LED, dessen Lichtstrahl auf einen Photo Transistor trifft. Dieser schaltet daraufhin durch und es liegt eine Spannung am Emitter an. Wird jetzt jedoch der

Lichtstrahl der LED unterbrochen, sperrt der Transistor und es fließt kein Strom. Bei der SPS-Programmierung ist daher zu beachten, dass sich der Ausgang des Sensors im nicht geschalteten Zustand auf HIGH befindet. Wird der Lichtstrahl jedoch unterbrochen, liegt am Sensorausgang keine Spannung an und der Eingang der SPS erhält ein LOW Signal. In unserem Fall werden TP808 zum Referenzieren verwendet. Hierbei ist zu beachten, dass die sich darin befindende Diode nur mit einer maximalen Flussspannung von 1,35 V betrieben werden darf.[26] Da die Opto Interrupter jedoch über ASi-Bus mit der SPS verbunden werden, welche eine Spannung von 24 V liefert, muss eine eigene Platine entworfen und gelötet werden, um das Bauteil nicht mit einer zu hohen Betriebsspannung zu zerstören. Hierfür wird die Software Fusion360 verwendet, welche das Designen von Leiterplatten ermöglicht. Hergestellt werden diese dann durch unsere eigene schulinterne Leiterplattenfertigung.

### Schaltungsentwurf

Die Schaltung sollte so konzipiert sein, dass keines der involvierten Bauteile über längeren Normalbetrieb oder durch kurzzeitige hohe Ströme bzw. Spannungen, beschädigt wird. Das Ziel ist, die korrekte Funktion des Opto Interrupters auch zukünftig noch sicher stellen zu können. Dafür ist besonders wichtig auf dessen elektrische Eigenschaften zu achten, welche im Datenblatt zu finden sind. Für den fertigen Entwurf der Schaltung siehe Abb.5.2

Im Opto Interrupter befindet sich eine LED mit einer maximalen Durchlassspannung von 1,35 V. und einer typischen Durchlassspannung  $V_F$  von 1,2 V. Um diese nicht mit den vollen 24 V der Betriebsspannung  $V_B$  zu überlasten muss ein Vorwiderstand  $R_1$  eingebaut werden. Um die LED zum leuchten zu bringen, soll ein minimaler  $I_F$  von 10 mA fließen. Dann lässt sich daraus der Vorwiderstand aus dem ohmschen Gesetz und mit Hilfe der Maschenregel berechnen:

$$R_1 = \frac{V_B - V_F}{I_F} = \frac{24 \text{ V} - 1,2 \text{ V}}{10 \text{ mA}} = 2,28 \text{ k}\Omega$$

In der HTL wir den Schülerinnen und Schülern die Widerstandsreihe E12 zur Verfügung gestellt. Daher wird in der Schaltung der nächstgrößere Widerstand mit dem Wert 2,7 kΩ verwendet.

Der Phototransistor, welcher als Gegenstück zur LED dient, darf mit einer maximalen Collector-Emitter-Spannung von 30 V betrieben werden, wodurch er gut geeignet ist für das Ziel der Schaltung. Um jedoch im besten Fall die gesamten 24 V für den Eingang der SPS an der Klemme X1 abgreifen zu können, wird ein Spannungsteiler verwendet, bei dem nach dem Transistor ein Widerstand  $R_3$  parallel zur Klemme X1 eingebaut wird. Da nur ein niedriger Strom benötigt wird, kann  $I_{R3}$  relativ klein sein, hier 1 mA. Daraus lässt sich dann der Widerstandswert wie folgt berechnen:

$$R_3 = \frac{V_B}{I_{R3}} = \frac{24 \text{ V}}{1 \text{ mA}} = 24 \text{ k}\Omega$$

Auch hier wird wieder der nächstgrößere Widerstandswert, der zur Verfügung gestellt wird, verwendet.  $R_3$  entspricht somit dem Wert 27 k $\Omega$ .

Für Funktionstests und die Inbetriebnahme ist es wichtig, dass eine Möglichkeit gegeben ist, den Zustand des Ausgangs der Schaltung anzusehen. Hierfür wird eine grüne 5mm LED verwendet, welche parallel zur Klemme X1 eingebaut wird. Da der Transistor einen maximalen Collector Strom von 20 mA besitzt, wurde die Entscheidung getroffen, die grüne LED nur mit 10 mA zu versorgen, da diese auch bei geringerem Strom genug Leuchtkraft für den benötigten Zweck besitzt. Bei einem Strom  $I_{R2}$  von 10 mA besitzt die LED einen Spannungsabfall  $V_{LED}$  von 2,1 V.[27] Daraus lässt sich dann der Widerstand  $R_2$  berechnen:

$$R_2 = \frac{V_B - V_{LED}}{I_{R2}} = \frac{24 \text{ V} - 2,1 \text{ V}}{10 \text{ mA}} = 2,19 \text{ k}\Omega$$

Der nächsthöhere Widerstand der E12 Reihe entspricht 2,2 k $\Omega$ , um auf Nummer sicher zu gehen wird jedoch der um eine Stufe größere Widerstand mit einem Wert von 2,7 k $\Omega$  verwendet. Wenn der Lichtstrahl im Opto Interrupter nicht unterbrochen wird und der Transistor somit durchschaltet, leuchtet die grüne LED. Wird jetzt der Lichtstrahl unterbrochen erlischt die LED. Damit diese nicht während des Normalbetriebs dauerhaft leuchtet wird ein Jumper eingebaut, mit welchen die LED ganz einfach aus der Schaltung ausgeschlossen werden kann.

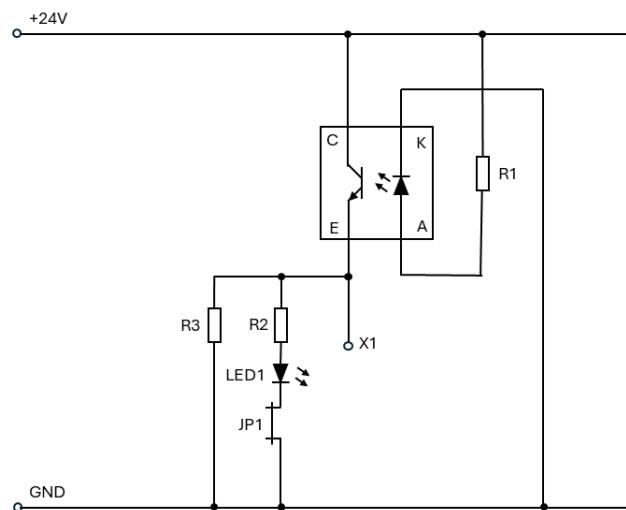


Abbildung 5.2: Schaltplan Referenzplatine

## Platinenentwurf und -herstellung

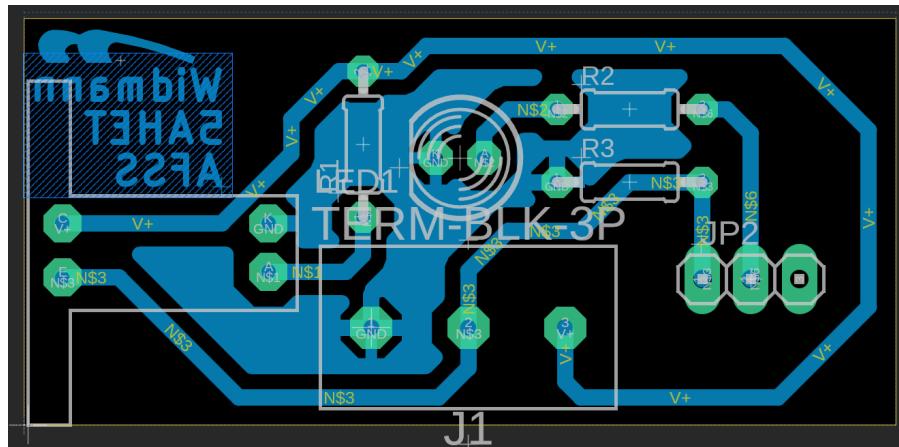


Abbildung 5.3: Leiterplattenplan Referenzplatine

Damit während eines Referenzvorgangs der Lichtstrahl der LED unterbrochen wird und der Referenztaster auslöst, müssen auch hier wieder Auslösevorrichtungen an den x-Schlitten, am Shuttle und auf den Seiten der Querfördererstation angebracht werden.

### 5.2.3 Lichttaster

### 5.2.4 Barcode-Scanner

## 5.3 AS-Interface

### 5.3.1 Allgemeines

### 5.3.2 Programmierung im TIA-Portal

### 5.3.3 Verkabelung

## 5.4 Sicherheitstechnik

### 5.4.1 Grundanforderungen und Planung

### 5.4.2 Realisierung

## 6 Anhang

### 6.1 Testergebnisse

### 6.2 Abmessungen

### 6.3 Datenblattauszüge

Im Anhang befinden sich weitere Detailinformationen des Projekts wie

- Datenblattauszüge, Fertigungsunterlagen (PCB-Layouts, Gehäusezeichnungen, 3D-Druckunterlagen, Montageanleitungen,...) etc.
- sämtliche geforderten Projektmanagementdokumente
- ein Businessplan (optional)

### 6.4 Projektmanagement

In diesem Kapitel soll auf das Projektmanagement des Projektes eingegangen werden. Zu Beginn empfiehlt es sich, die einzelnen Bereiche des Projektmanagements zu erklären und anschließend in einzelnen Kapiteln zu behandeln.

#### 6.4.1 Aufgabenstellung des Gesamtprojekts

Fügen Sie an dieser Stelle den Text der genehmigten Aufgabenstellung ein, der in die Diplomarbeitsdatenbank eingegeben wurde.

#### 6.4.2 Scrum-Projektplan

Fügen Sie hier den vollständigen Scrum-Projektplan, wobei die Nummern der Tasks mit der Arbeitszeitaufzeichnung übereinstimmen müssen. Der Scrum-Projektplan kann auf mehrere Seiten aufgeteilt werden.

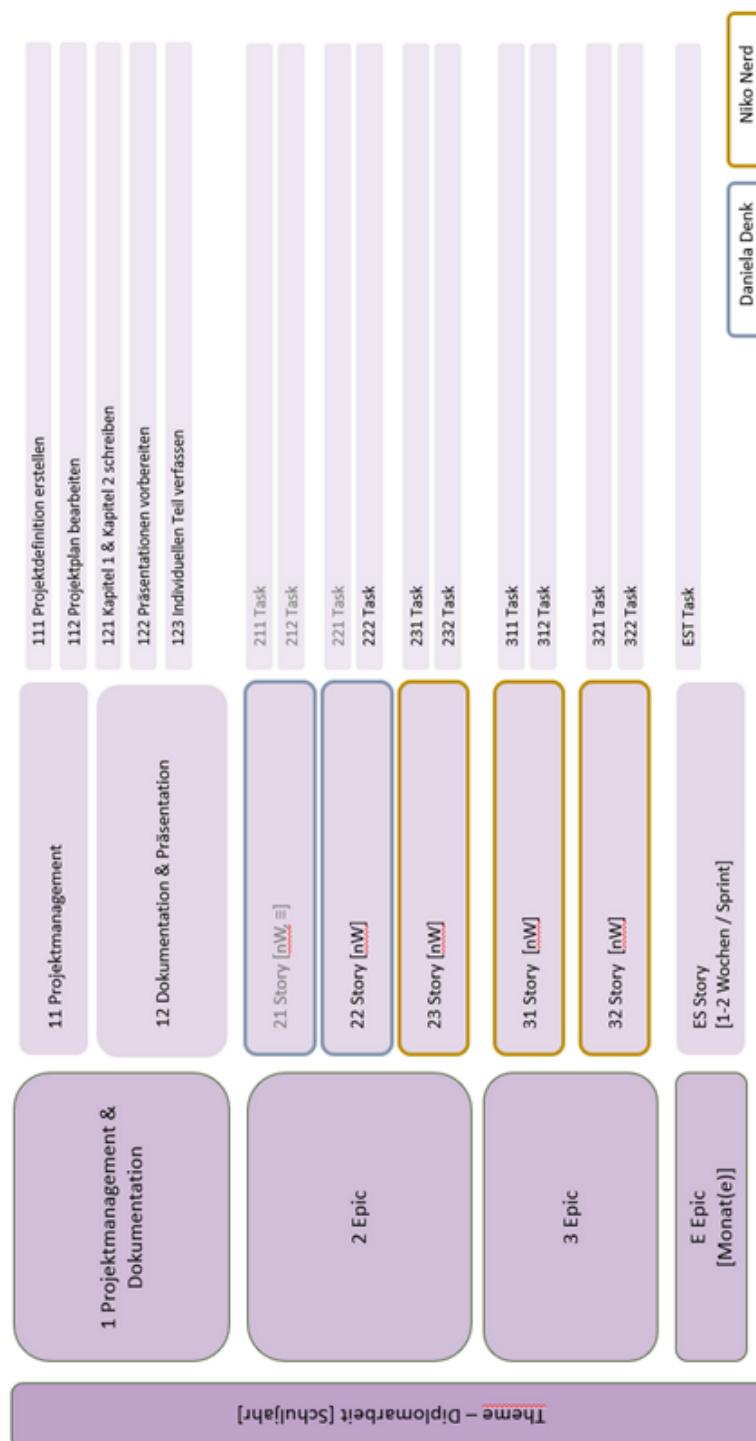


Abbildung 6.1: Scrum Projektplan mit Tasks

### 6.4.3 Terminplanung

## 6.5 Inbetriebnahme

Nachdem typische Projekte aus mehreren Komponenten bestehen, ist es oft nicht trivial die einzelnen Komponenten korrekt zu konfigurieren und das Gesamtsystem in Betrieb zu nehmen. In diesem Kapitel soll eine vollständige, präzise und trotzdem möglichst kompakte Anleitung zur Inbetriebnahme des Systems dargelegt werden. Die Schritte sollen in dem Detailgrad beschrieben werden, dass ein durchschnittlicher Schüler des vierten Jahrganges das Projekt in Betrieb nehmen kann. Exemplarisch sollten Punkte wie die folgenden behandelt werden – die Aufzählung ist nicht vollständig):

- Treiberinstallationen und Systemkonfigurationen
- Zu empfehlen wäre bei Server-Installationen ein Setup-Script, welches auf einem vordefinierten Docker-container aufbaut.
- Welche Schritte sind notwendig, um das Projekt mit dem vorhandenen Code / Schaltplänen (auf GIT, CD, Netzlaufwerk, etc.) in Betrieb zu nehmen.
- Bei Schaltungen mit mehreren Platinen muss beschrieben werden, wie diese miteinander verbunden werden müssen.

## 6.6 Kostenaufstellung

Für die Kalkulation im Gesamtprojekt sind folgende Kosten zu erfassen:

- Kosten für Material (Hard- und Software)
- externe Kosten (z.B.: Zukauf von Sensoren, Funkmodule, spezielle Entwicklungsumgebungen, etc.)

Diplomarbeit: Titel der Diplomarbeit (DE), 03.05.2024		
Kostenart	Stückpreis / Einheiten	Kosten in €
Elektronische Bauteile		
Sonstige Materialien		
Software Lizenzen		
PCB Fertigung, Gehäuse		
Baugruppen, Module		
usw.		
Gesamtkosten		

Abbildung 6.2: Kostenaufstellung

## 6.7 Besprechungsprotokolle

AN	ERGEBNISPROTOKOLL ZUR BESPRECHUNG		Eingangs- vermerke
	Veranstalter <b>Automated Factory Storage System</b>	Protokollführer <b>Benedikt Simbürger</b>	
18.9.2024 am Tag, Datum	Projektbetreuer <b>Dipl.-Ing. Christian Sallinger</b>		Erstellungsdatum: <b>2024-09-18</b> Ort/Raum <b>HTL-Mössingerstraße</b>

**Thema:**
**Projektstartgespräch**
**Projektname:** Automated Factory Storage System

Teilnehmer (alphabetisch)	Unterrichtete (alphabetisch)
Name	Klasse, Standort
Hr. Dipl.-Ing. Christian Sallinger	Klagenfurt
Hr. Benedikt Simbürger	5AHET, Klagenfurt
Hr. Vincent Sonvilla	5AHET, Klagenfurt
Fr. Elena Widmann	5AHET, Klagenfurt
Hr. Nikolaj Voglauer	5AHET, Klagenfurt

- Spezifizierung (Funktionen, Realisierungsansätze)
- Blockschaltbild
- Projektstrukturplan mit Aufgabenverteilung
- Terminplanung

**Erlledigung  
von, Termin**

Datum 18.9.2024      gez. Benedikt Simbürger

Abbildung 6.3: Besprechungsprotokoll 10.12.2024



<b>AN</b>	<b>ERGEBNISPROTOKOLL ZUR BESPRECHUNG</b>	Eingangsvermerke
Teilnehmer und Unterrichtete	Veranstalter <b>Automated Factory Storage System</b>	
	Protokollführer <b>Benedikt Simbürger</b>	
16.10.2024 am Tag, Datum	Projektbetreuer <b>Dipl.-Ing. Christian Sallinger</b>	Erstellungsdatum: <b>2024-10-16</b> Ort/Raum <b>HTL-Mössingerstraße</b>

**Thema:**
**Projektstatusgespräch**

Projektname: Automated Factory Storage System

Teilnehmer (alphabetisch)	Unterrichtete (alphabetisch)
Name	Klasse, Standort
Hr. Dipl.-Ing. Christian Sallinger	Klagenfurt
Hr. Benedikt Simbürger	5AHET, Klagenfurt
Hr. Vincent Sonvilla	5AHET, Klagenfurt
Fr. Elena Widmann	5AHET, Klagenfurt
Hr. Nikolaj Voglauer	5AHET, Klagenfurt

<ul style="list-style-type: none"> <li>• Realisierung der Rahmenmontage</li> <li>• Sponsorenstatus:           <ul style="list-style-type: none"> <li>○ Igus: Komponenten unterwegs</li> <li>○ Weidmüller: Material Zugesagt</li> <li>○ Lapp: unterwegs</li> </ul> </li> <li>• Umsetzung der E-Planung</li> </ul>	<b>Erledigung von, Termin</b>
--	-------------------------------

Datum 16.10.2024 gez. Benedikt Simbürger  
 HTL Mössingerstraße  
 Projektbesprechung\_1.docx

Seite 1 von 1  
 V01

Abbildung 6.4: Besprechungsprotokoll 16.10.2024



<b>AN</b>	<b>ERGEBNISPROTOKOLL ZUR BESPRECHUNG</b>	Eingangs-vermerke
Teilnehmer und Unterrichtete	Veranstalter <b>Automated Factory Storage System</b>	
	Protokollführer <b>Benedikt Simbürger</b>	
10.12.2024 am Tag, Datum	Projektbetreuer <b>Dipl.-Ing. Christian Sallinger</b>	Erstellungsdatum: <b>2024-12-10</b> Ort/Raum <b>HTL-Mössingerstraße</b>

**Thema:**  
**Projektstatusgespräch**

Projektname: Automated Factory Storage System

Teilnehmer (alphabetisch)	Unterrichtete (alphabetisch)
Name	Klasse, Standort
Hr. Dipl.-Ing. Christian Sallinger	Klagenfurt
Hr. Benedikt Simbürger	5AHET, Klagenfurt
Hr. Vincent Sonvilla	5AHET, Klagenfurt
Fr. Elena Widmann	5AHET, Klagenfurt
Hr. Nikolaj Voglauer	5AHET, Klagenfurt

- Motoransteuerungsfortschritte
- X-Achsen Shuttle Konstruktion
- ASi-Verbindungsfortschritte
- Platinenfortschritt

**Erledigung von, Termin**

Datum 10.12.2024 gez. Benedikt Simbürger

Abbildung 6.5: Besprechungsprotokoll 10.12.2024



<b>AN</b>	<b>ERGEBNISPROTOKOLL ZUR BESPRECHUNG</b>	Eingangs-vermerke
Teilnehmer und Unterrichtete	Veranstalter <b>Automated Factory Storage System</b>	
	Protokollführer <b>Benedikt Simbürger</b>	
11.3.2025 am Tag, Datum	Projektbetreuer <b>Dipl.-Ing. Christian Sallinger</b>	Erstellungsdatum: <b>2025-3-11</b> Ort/Raum <b>HTL-Mössingerstraße</b>

**Thema:**
**Projektstatusgespräch**

Projektname: Automated Factory Storage System

Teilnehmer (alphabetisch)	Unterrichtete (alphabetisch)
Name	Klasse, Standort
Hr. Dipl.-Ing. Christian Sallinger	Klagenfurt
Hr. Benedikt Simbürger	5AHET, Klagenfurt
Hr. Vincent Sonvilla	5AHET, Klagenfurt
Fr. Elena Widmann	5AHET, Klagenfurt
Hr. Nikolaj Voglauer	5AHET, Klagenfurt

<ul style="list-style-type: none"> <li>• X-Achsen fortschritt bis Abgabetermin</li> <li>• SPS Programmierungsstatus</li> <li>• Projektweiterführungsaspekte</li> <li>• Status der Verkabelung</li> </ul>	<b>Erledigung von, Termin</b>
--	-------------------------------

Datum 11.3.2025      gez. Benedikt Simbürger

Abbildung 6.6: Besprechungsprotokoll x.x.xxxx

## 6.8 Arbeitsnachweis

Jedes Teammitglied (Schüler/In) hat einen vollständigen Arbeitszeitnachweis, der außerhalb des Unterrichts verrichteten Tätigkeiten, in tabellarischer Form zu erbringen.

Eine entsprechende Vorlage wird auf den Schulrechner in Form einer Excel-Vorlage bereitgestellt.

## Literaturverzeichnis

- [1] Pickengine. *Pickengine*. Online. URL: <https://www.pickengine.com/>.
- [2] Autodesk. Online. URL: <https://www.autodesk.com/de/support/technical/article/caas/sfdcarticles/sfdcarticles/DEU/Autodesk-EAGLE-now-included-with-Fusion-360.html> (besucht am 21.02.2025).
- [3] Siemens. Online. URL: <https://www.automation.siemens.com/sce-static/learning-training-documents/tia-portal/hw-config-s7-1200/programming-guideline-de.pdf> (besucht am 21.02.2025).
- [4] Prusa Research by Josef Prusa. Online. URL: [https://help.prusa3d.com/de/article/erster-druck-mit-prusaslicer\\_1753](https://help.prusa3d.com/de/article/erster-druck-mit-prusaslicer_1753) (besucht am 25.02.2025).
- [5] igus. Online. URL: <https://www.igus.at/unternehmen> (besucht am 01.03.2025).
- [6] lapp. Online. URL: <https://www.lapp.com/de/at/unternehmen/ueber-lapp-oesterreich/e/000123> (besucht am 03.03.2025).
- [7] mädler. Online. URL: <https://www.maedler.de/Static/Videos> (besucht am 03.03.2025).
- [8] elektro4000de. URL: <https://www.elektro4000.de/Kabel-Leitungen/Kabeltragsysteme/Stiel-Profilschienen/Verbinder/Item-Standard-Verbindungssatz-0-0-026-07::1743227.html> (besucht am 19.11.2024).
- [9] globalsources.com. URL: <https://www.globalsources.com/Plastic-coated/plastic-coated-bearing-1197379614p.html> (besucht am 19.11.2024).
- [10] igus. *10 Tipps für eine langlebige „freitragende“ Energieführung*. Online. 2024. URL: <https://www.igus.at/service/kontakt/energieketten-tipps-fuer-freitragende-bewegung> (besucht am 10.12.2024).
- [11] igus. *9 Tipps für eine langlebige „stehende“ Energieführung*. Online. 2024. URL: <https://www.igus.at/service/kontakt/energieketten-tipps-stehende-anwendungen> (besucht am 10.12.2024).
- [12] OpenAI. *ChatGPT: OpenAI's Language Model*. <https://openai.com/chatgpt>. 2024. (Besucht am 21.11.2024).
- [13] Siemens. *Bereitstellung eines Frameworks zum Webseiten-Upload*. Online. 2024. URL: [https://cache.industry.siemens.com/dl/files/059/109814059/att\\_1117153/v1/109814059\\_WebserverFramework\\_DOC\\_v10\\_de.pdf](https://cache.industry.siemens.com/dl/files/059/109814059/att_1117153/v1/109814059_WebserverFramework_DOC_v10_de.pdf) (besucht am 03.10.2025).
- [14] cloudflight. 2024. URL: <https://www.cloudflight.io/de/blog/docker-container-die-zukunft-moderner-applikationen-und-multi-cloud-deployments/> (besucht am 15.11.2024).
- [15] Siemens. Online. URL: <https://www.automation.siemens.com/sce-static/learning-training-documents/tia-portal/hw-config-s7-1200/programming-guideline-de.pdf> (besucht am 07.03.2025).

- [16] Xovi Glossar. Online. URL: <https://www.xovi.de/was bedeutet http/#:~:text=HTTP%20ist%20die%20Abk%C3%BCrzung%20f%C3%BCr,Client%20und%20Webserver%20verwendet%20wird>. (besucht am 18.02.2025).
- [17] OPC Router. Online. URL: <https://www.opc-router.de/was-ist-opc-ua/> (besucht am 18.02.2025).
- [18] copyservice-shop. Online. URL: <https://www.copyservice-shop.de/Alu-Verbundplatte-Dibond-A4-A3-etc> (besucht am 13.02.2025).
- [19] Komaxgroup. Online. URL: <https://www.komaxgroup.com/en/stories/experience-the-smart-cabinet-building-initiative-up-close-and-personal> (besucht am 13.02.2025).
- [20] Smart cabinet building. Online. URL: [https://www.smart-cabinet-building.com/de/loesungsangebot/digitales\\_engineering/loesungsangebot\\_zukene3/index.jsp](https://www.smart-cabinet-building.com/de/loesungsangebot/digitales_engineering/loesungsangebot_zukene3/index.jsp) (besucht am 13.02.2025).
- [21] Siemens. *S7-1500/S7-1500T Achsfunktionen V5.0 im TIA Portal V16*. Online.
- [22] Schmersal. *Datenblatt Z4VH 335-11Z-M20-RMS*. Online. URL: [https://products.schmersal.com/de\\_AT/export/product/846/pdf](https://products.schmersal.com/de_AT/export/product/846/pdf).
- [23] Schmersal. *Datenblatt Z4VH 335-02Z-M20*. Online. URL: [https://products.schmersal.com/de\\_AT/export/product/6992/pdf](https://products.schmersal.com/de_AT/export/product/6992/pdf).
- [24] Schmersal. *Produktbild*. Online. URL: [https://products.schmersal.com/de\\_AT/z4vh-335-02z-m20-101167776.html#mz-expanded-view-708407000705](https://products.schmersal.com/de_AT/z4vh-335-02z-m20-101167776.html#mz-expanded-view-708407000705).
- [25] Pepperl+Fuchs. *Datenblatt NBN8-18GM60-A2-V1*. Online. URL: [https://files.pepperl-fuchs.com/webcat/navi/productInfo/pds/326162-0005\\_ger.pdf?v=20240130135820](https://files.pepperl-fuchs.com/webcat/navi/productInfo/pds/326162-0005_ger.pdf?v=20240130135820).
- [26] Sivago Semiconductors. Online. URL: <https://www.sivago.com.cn/upload/pdf/2022/TP808.pdf> (besucht am 28.02.2025).
- [27] Kingbright. Online. URL: [https://cdn-reichelt.de/documents/datenblatt/A500/LED5MMSTGE\\_LED5MMSTGN\\_LED5MMSTRT~KIN.pdf](https://cdn-reichelt.de/documents/datenblatt/A500/LED5MMSTGE_LED5MMSTGN_LED5MMSTRT~KIN.pdf) (besucht am 10.03.2025).

# Abbildungsverzeichnis

Abb. 2.1	Item Profil mit Standartverbindungssatz, Quelle: [8] . . . . .	17
Abb. 2.2	V-Slot-Profil mit V-Wheel, Quelle: [9] . . . . .	18
Abb. 2.3	Prototypen . . . . .	18
Abb. 2.4	X-Achse . . . . .	22
Abb. 2.5	Zahnriemenkontraktion grafisch dargestellt . . . . .	24
Abb. 2.6	YZ-Achse . . . . .	27
Abb. 2.7	Bauteilzeichnung Umlenkrolle . . . . .	28
Abb. 2.8	Umsetzung der Umlenkrollen . . . . .	30
Abb. 2.9	Gesamtüberblick des Servers . . . . .	32
Abb. 2.10	Diagramm der BMOS bereitstellung . . . . .	37
Abb. 2.11	Datenbankschema des AFSS . . . . .	38
Abb. 2.12	Docker-Logo: [14] . . . . .	39
Abb. 3.1	Organisationsbausteine [15] . . . . .	44
Abb. 3.2	Zugriff auf Global-DB [15] . . . . .	45
Abb. 4.1	Dibond-Platte, Quelle: [18] . . . . .	51
Abb. 4.2	Digitaler Zwilling, Quelle: [19] . . . . .	52
Abb. 4.3	Sommerprototyp eines Serverschrances . . . . .	53
Abb. 4.4	Serverschrank konstruiert . . . . .	54
Abb. 5.1	Endschalter . . . . .	57
Abb. 5.2	Schaltplan Referenzplatine . . . . .	59
Abb. 5.3	Leiterplattenplan Referenzplatine . . . . .	60
Abb. 6.1	Scrum Projektplan mit Tasks . . . . .	62
Abb. 6.2	Kostenaufstellung . . . . .	65
Abb. 6.3	Besprechungsprotokoll 10.12.2024 . . . . .	66
Abb. 6.4	Besprechungsprotokoll 16.10.2024 . . . . .	67
Abb. 6.5	Besprechungsprotokoll 10.12.2024 . . . . .	68
Abb. 6.6	Besprechungsprotokoll x.x.xxxx . . . . .	69

## **Tabellenverzeichnis**

Tab. 2.1	X-Achse unbeladen und eingefahren . . . . .	23
Tab. 2.2	X-Achse beim Ladevorgang . . . . .	23