



**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 . . . . .	15
1.6.1 KNAPP AG . . . . .	15
1.6.2 Weidmüller GMBH . . . . .	15
1.6.3 Igus . . . . .	15
1.6.4 LAPP Austria GmbH . . . . .	16
1.6.5 Mädler GmbH . . . . .	16
<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 . . . . .	20
2.1.4 X-Achse . . . . .	20
2.1.5 Y-Achse . . . . .	23
2.1.6 Z-Achse . . . . .	26
2.1.7 Lager . . . . .	29
2.1.8 Querförderer . . . . .	29
2.1.9 Fertigung der Einzelteile . . . . .	29
2.1.10 Aufbau . . . . .	32
2.1.11 Fazit . . . . .	32
2.2 Software und Benutzeroberfläche . . . . .	33
2.2.1 Grundlegendes . . . . .	33
2.2.2 Aufbau . . . . .	33

2.2.3 Benutzeroberfläche . . . . .	34
2.2.4 Backend . . . . .	36
2.2.5 APIs . . . . .	37
2.2.6 Lageralgorithmus für SPS . . . . .	38
2.2.7 Datenbanken . . . . .	39
2.2.8 Docker . . . . .	40
2.2.9 Artikelsuche . . . . .	41
2.2.10 Fazit . . . . .	44
<b>3 Integration und Programmierung der Steuerungstechnik (Vincent Sonvilla)</b>	<b>45</b>
3.1 Aufgabenstellung . . . . .	45
3.2 Tia-Portal Grundlagen . . . . .	45
3.2.1 Allgemeines . . . . .	45
3.2.2 Programmbausteine . . . . .	46
3.2.3 Technologieobjekte . . . . .	48
3.2.4 Programmiersprachen . . . . .	49
3.2.5 Bibliotheken . . . . .	50
3.3 Motoransteuerung . . . . .	51
3.3.1 Kontrollsiganle für den Motortreiber . . . . .	51
3.3.2 Motion Control Anweisungen . . . . .	52
3.3.3 Konfiguration Technologieobjekt . . . . .	53
3.3.4 Synchronisation . . . . .	55
3.3.5 Ansteuerung des Förderbands . . . . .	55
3.3.6 Tia-Portal Programm . . . . .	56
3.4 SPS-Server Kommunikation . . . . .	56
3.4.1 Zur Auswahl stehende Kommunikationsprotokolle . . . . .	56
3.4.2 Verbindungsherstellung . . . . .	57
3.4.3 Datenaustausch . . . . .	58
3.4.4 Datenfilterung . . . . .	59
3.4.5 Tia-Portal Programm . . . . .	60
3.5 Herausforderungen . . . . .	61
<b>4 Elektroplanung und Realisierung (Nikolaj Voglauer)</b>	<b>62</b>
4.1 Elektroplanung . . . . .	62
4.1.1 Einleitung - Grundanforderungen . . . . .	62
4.1.2 Elektrik spezifische Anforderungen . . . . .	62
4.1.3 Mechanische Planung . . . . .	64
4.1.4 Elektrische Planung (E-Plan) . . . . .	73
4.2 Die Auslegung von den jeweiligen Komponenten (Tabelle mit allen Komponenten machen?)—Vor die E-Planung setzen um im Folgekapitel nicht konstatieren zu müssen was welches teil macht . . . . .	83
4.3 Realisierung . . . . .	83

---

<b>5 Sensorik und Sicherheitstechnik (Elena Widmann)</b>	<b>84</b>
5.1 Sensorik . . . . .	84
5.1.1 Aufgabenstellung . . . . .	84
5.1.2 Endschalter . . . . .	84
5.1.3 Referenztaster . . . . .	85
5.1.4 Lichttaster . . . . .	89
5.1.5 Barcode-Scanner . . . . .	90
5.2 AS-Interface . . . . .	92
5.2.1 Feldbusssysteme [49] . . . . .	92
5.2.2 Programmierung im TIA-Portal . . . . .	94
5.2.3 Verkabelung . . . . .	94
5.3 Sicherheitstechnik . . . . .	94
5.3.1 Aufgabenstellung . . . . .	94
5.3.2 Personenschutz . . . . .	94
5.3.3 Schutz des Systems . . . . .	95
5.4 Fazit . . . . .	96
<b>6 Anhang</b>	<b>97</b>
6.1 Testergebnisse . . . . .	97
6.2 Abmessungen . . . . .	97
6.3 Datenblattauszüge . . . . .	97
6.4 Projektmanagement . . . . .	97
6.4.1 Aufgabenstellung des Gesamtprojekts . . . . .	97
6.4.2 Scrum-Projektplan . . . . .	97
6.4.3 Terminplanung . . . . .	99
6.4.4 Arbeitspakete . . . . .	100
6.5 Inbetriebnahme . . . . .	101
6.6 Kostenaufstellung . . . . .	102
6.7 Besprechungsprotokolle . . . . .	103
6.8 Arbeitsnachweis . . . . .	107
<b>Literaturverzeichnis</b>	<b>111</b>
<b>Abbildungsverzeichnis</b>	<b>115</b>
<b>Tabellenverzeichnis</b>	<b>117</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

### 1.6.1 KNAPP AG

Die KNAPP AG ist ein vergleichsweise junges Grazer Unternehmen mit dem Schwerpunkt auf Intralogistik und automatisierten Lagerlösungen. Die Firma wurde 1952 von Günter Knapp gegründet und hat sich seitdem als einer der führenden Anbieter für Lagerlogistik behauptet. Mit über 40 Standorten weltweit und mehr als 4.100 MitarbeiterInnen arbeitet KNAPP stets an neuen innovativen Ideen und Lösungen für die komplexen Probleme der Lagerlogistik. Viele der etablierten Marktlösungen wurden von KNAPP entwickelt und umgesetzt. [5]

KNAPP ist seit der ersten Stunde ein Unterstützer des Projektes AFSS. Die Materialspenden deckten die meisten elektrischen Komponenten, sowie die Siemens-SPS, und ein weiterführendes Alleinstellungsmerkmal von KNAPP war, dass sie dem AFSS-Betreuung zur Verfügung gestellt haben. Vor allem im Bereich der Software und der Steuerung der Anlage konnte KNAPP mit dem Wissen aus der Praxis effizient helfen. KNAPP-Mitarbeiter waren bis zum Ende des Projektes stets in Kontakt mit dem AFSS-Team und bemüht zu helfen.

### 1.6.2 Weidmüller GMBH

Weidmüller wurde 1850 von Carl August Weidmüller als Unternehmen in der Textilindustrie gegründet. Heute ist die Firma in mehr als 80 Ländern vertreten und arbeitet in der Branche Elektrotechnik und Automatisierung. Konkret deckt Weidmüller ein breites Spektrum an Produkten ab. Einerseits werden viele Komponenten für den Schaltschrankbau hergestellt, sowie die neuartigen Push-In-Klemmen, und andererseits hat das Unternehmen auch Produkte in Richtung Digitalisierung sowie Komponenten für den PV-Bereich. [6]

Weidmüller stellte dem AFSS viele Komponenten für den Schaltschrank zur Verfügung, wie die genannten Push-In-Klemmen, aber auch DC-Sicherungen, Feldbuskoppler und PWM-/Schrittmotorkarten. Die Firma ermöglichte mit ihren Materialspenden, dass das Projekt so weitergeführt werden konnte.

### 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. [7]

Igus unterstützte das 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.[8]

Um das AFSS verwirklichen zu können wurde es von LAPP durch die Bereitstellung diverser Schleppkettenkabel unterstützt, welche zur Motor- und Geberverkabelung sowie zur Subbusverteilung verwendet wurden. Außerdem wurden 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.[9]

Das Projekt wurde 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 Gewichtsparmannahmen 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.

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



Abbildung 2.1: Item Profil mit Standardverbindungssetz, Quelle: [10]



Abbildung 2.2: V-Slot-Profil mit V-Wheel, Quelle: [11]

## V-Slot

Auch V-Slot-Profile sind Aluminium-Extrusionen. Diese können grundsätzlich auch ähnlich wie Item-Profile mit Nutensteinen 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.

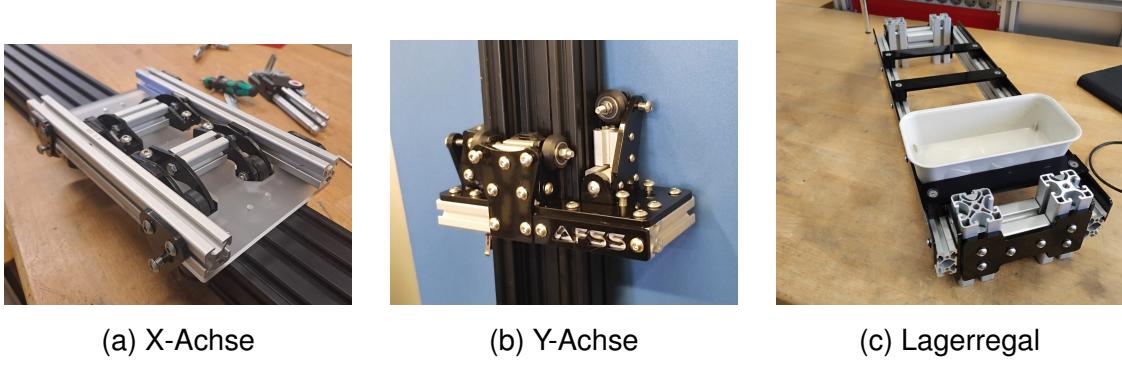


Abbildung 2.3: Prototypen

### 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 Werkstätte 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 BR5 (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 Ordnern. 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 reibungslos 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 verbaut, 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 [12]. 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.

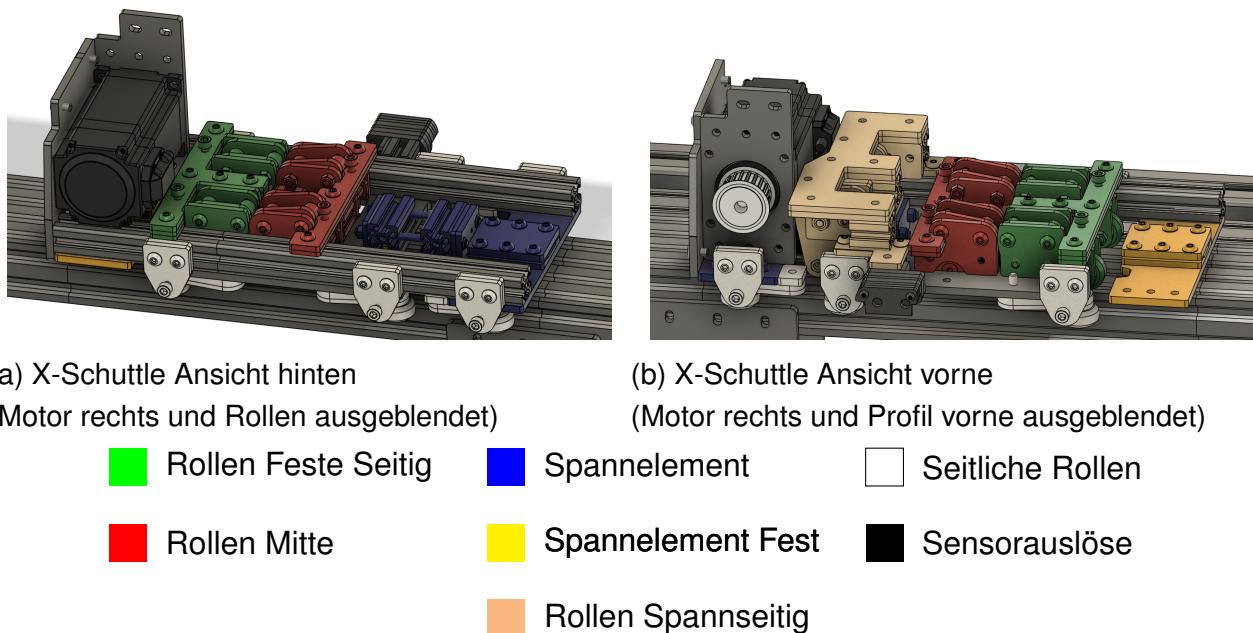


Abbildung 2.4: X-Achse

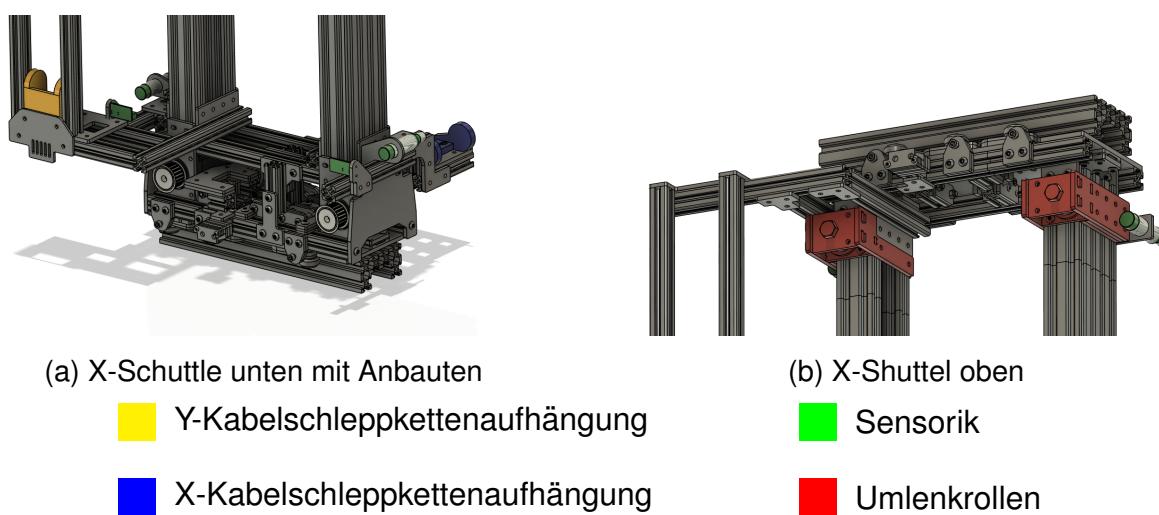


Abbildung 2.5: X-Achse Gesamtansichten

## 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 Antriebe 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 (siehe 2.5b). 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 2.6: Zahnriemenkontraktion grafisch dargestellt

Aus der Abbildung 2.6 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. Dabei muss die Halterung auf der Zahnriemenseite dünner ausgeführt werden, da sonst eine Kollision zwischen Riemen und Halterung zustande kommen würde.

### 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 Anschlusselemente fest um die ersten Kettenglieder extra zu unterstützen. [13]

#### 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 Mindestverfahrweg 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:

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

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 2.7: YZ-Achse



Abbildung 2.8: XY-Shuttle Gesamtansicht

## 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 erhält.

## 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

Die Fertigung der Bauteile erfolgt parallel zur Montage. Hierzu werden viele Teile Gefräßt, Gelasert und Aluminiumprofile zugeschnitten. Manche Teile sind jedoch komplexer und müssen speziell angefertigt werden.

### 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 2.9: Bauteilzeichnung Umlenkrolle

Die Fertigung dieses Teils nach Abb. 2.9 lässt sich in folgende Teilschritte unterteilen:

- Zuerst die Frontfläche plandrehen (Drehzahl:  $900 \text{ U min}^{-1}$ )
- 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. 2.10a 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. 2.10b ). ( $540 \text{ U min}^{-1}$ )
- 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 2.10c) und können auf einem 8-mm-Schaft montiert werden.



Abbildung 2.10: 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  $740 \text{ U min}^{-1}$  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 Meißel 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 Motoraufhängung Platz wegnimmt, ist es erforderlich, die Zahnscheibe unkonventionell mit dem Motor zu verbinden. Es werden also in der Zahnriemen-Kontaktfläche zwei Löcher gebohrt, angesenkt und mit M4-Gewinde versehen, um dort 2 M4-Wurmschrauben einzuschrauben. Bei der Länge der Wurmschrauben muss darauf geachtet werden, dass sie im montierten Zustand vollständig unter der Oberfläche liegen, um den Zahnriemen nicht zu beschädigen.

### 2.1.10 Aufbau

Nach Fertigung der Einzelteile kann das AFSS sukzesive zusammengebaut werden.

## Rahmen

Gestartet wird mit der Montage des Rahmen. Hierfür werden zuerst alle Aluminiumprofile auf Länge zugeschnitten und Gewinde in den Enden geschnitten. Da nicht genug Profile in der Gesamtlänge des AFSS verfügbar sind, müssen teils noch Verbinder 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.

Nach Fertigung der Beuteile erfolgt auch der Zusammenbau des sehr komplexen unteren X-Shuttels relativ reibungslos. Die Stoßverbindung der V-Slot-Profile kann jedoch nicht 100%ig ausgerichtet werden, deshalb werden diese Übergänge noch abgeschliffen.

### 2.1.11 Fazit

In der Mechanik des AFSS sind nach aktuellem stand über 1000 Bauteile verbaut. Die Umsetzung dieses Projekts war nur durch eine sehr umfangreiche theoretische Planung möglich. Zwischen Haupt- und Einzelversionsiterationen wurden mehrere hundert Versionen der Mechanik in Fusion360 durchgeführt. Das dabei erlernte wissen, sowie das daraus resultierende Design, welches viele Aspekte von Fertigung, Wartung und Zusammenbau bedenkt, macht es erst möglich ein solches Vorhaben zu versuchen.

## 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

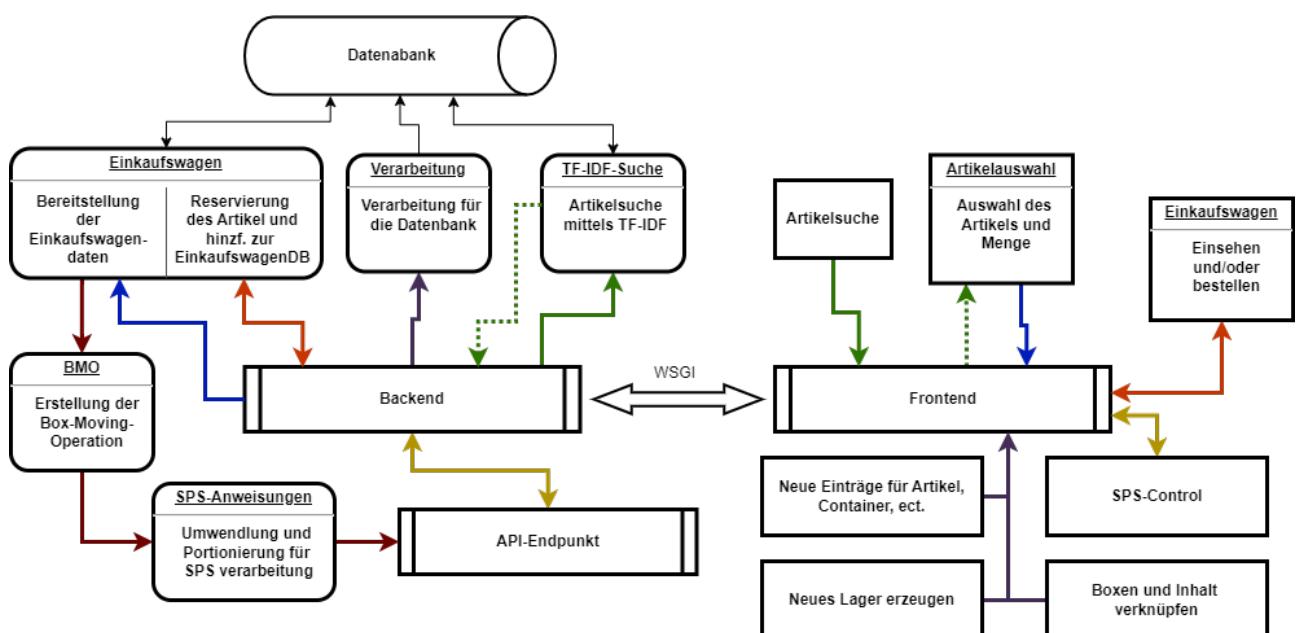


Abbildung 2.11: Gesamtüberblick des Servers

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. [14]

## 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. [14]

## 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. [14]

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.[14]

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.[14]

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 [15]. 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. Dies 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

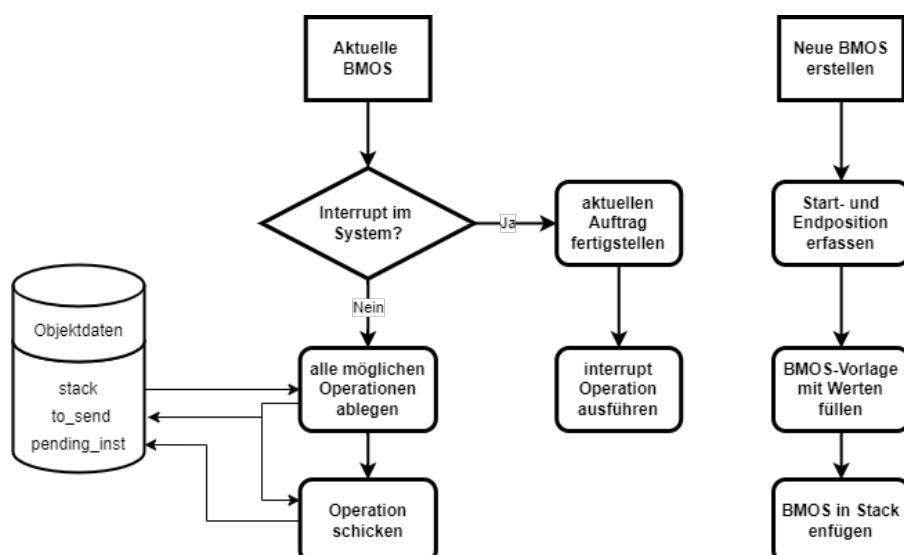


Abbildung 2.12: Diagramm der BMOS bereitstellung

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-Moving-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:

- '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.

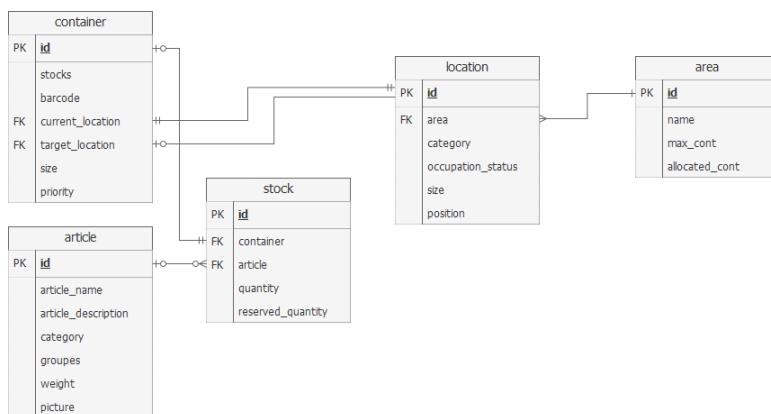


Abbildung 2.13: Datenbankschema des AFSS

Wie in 2.13 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 einer 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 an Box eingelagert wird.

## 2.2.8 Docker

Docker ist eine Umgebung, in der Softwareprojekte isoliert werden können. Da es besonders bei Projekten mit viele Pakete, mit verschiedenen Versionen, zu konflikten kommen kann, 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 den Programmeinstiegspunkt angegeben.

Ein zweier Docker Conteianer wird mit einem MySQL-Image erstellt, dort wird die Datenbank aufgesetzt.

Um diese zwei Container miteinander Kommunizieren zu lassen, ist es ntig 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 aleinstehend 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 Consolenbefehl der auf das docker-compose File zugreift.

```
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.[14]

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  
    // enough  
3     let lower_document = document.to_lowercase();  
4  
5     // Split the document into words  
6     let normalize_document: Vec<&str> = lower_document.split_whitespace()  
        .collect();  
7     // Make sure the searchterm is lowercase  
8     let normalize_term = term.to_lowercase();  
9 }
```

```

10     // Count occurrences of the term in the document
11     let count = normalize_document
12         .iter()
13         .filter(|&&word| word == normalize_term) // Compare each word
14             with the term
15             .count();
16
17     // Calculate the term frequency as occurrences / total number of
18     words
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     let mut num_documents_with_this_term = 0;
4
5     // Iterate over all documents to check if they contain the term
6     for doc in all_documents {
7         // Normalize both term and document by converting them to
8         lowercase
9         let lower_doc = doc.to_lowercase();
10        let normalized_doc: Vec<&str> = lower_doc.split_whitespace().collect();
11
12        // Check if the term exists in the document
13        if normalized_doc.contains(&term.to_lowercase().as_str()) {
14            num_documents_with_this_term += 1;
15        }
16
17        // Calculate IDF
18        if num_documents_with_this_term > 0 {
19            // Apply the IDF formula: 1 + log(total_documents /
20            documents_with_term)
21            1.0 + ((all_documents.len() as f64) / (
22                num_documents_with_this_term as f64)).ln()
23        } else {
```

```
21         // If the term is not found in any document, return 1.0
22         1.0
23     }
24 }
```

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 Einthalten: [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 Durchcklickmenue zu finden.

### 2.2.10 Fazit

Die Serverseitige Software des AFSS, also WMS und Benutzeroberfläche, wurde im Laufe des Projekts sehr komplex. Dadurch war es eine sehr gute Entscheidung den Server in Hochsprachen auszuführen. Um einen zuverlässigen Dauerbetrieb zu gewährleisten, wäre es jedoch noch nötig Unitests für wichtige Funktionen wie APIs oder den Lageralgorithmus zu implementieren. Weiters müsste die Benutzeroberfläche noch Bedienerfreindlicher gestaltet werden, da diese momentan noch eher rudimentär ausgeführt ist. Außerdem wären noch kleinere "quality-of-live"-änderungen durchzuführen. Insgesamt sit die Software aber schon sehr ausgereift und kann die Anforderung, der Lagersteuerung und -verwaltung durchführen.

### 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 Allgemeines

###### Arbeitsweise einer SPS

Eine SPS arbeitet zyklisch. In Abb. 3.1 wird gezeigt wie ein solcher Zyklus aussieht. Bei erstmaligem Starten oder Neustarten der SPS werden zuerst alle Ausgänge, Merker, etc. auf Null gesetzt. Danach startet die zyklische Arbeitsweise. Zuerst wird ein Prozessabbild der Eingänge gemacht. Mit diesen Eingangswerten wird dann das Programm ausgeführt. Anschließend wird ein Prozessabbild der Ausgänge gemacht. Dieses wird dann an die Ausgänge übergeben. Danach beginnt der Zyklus von vorne. [16]

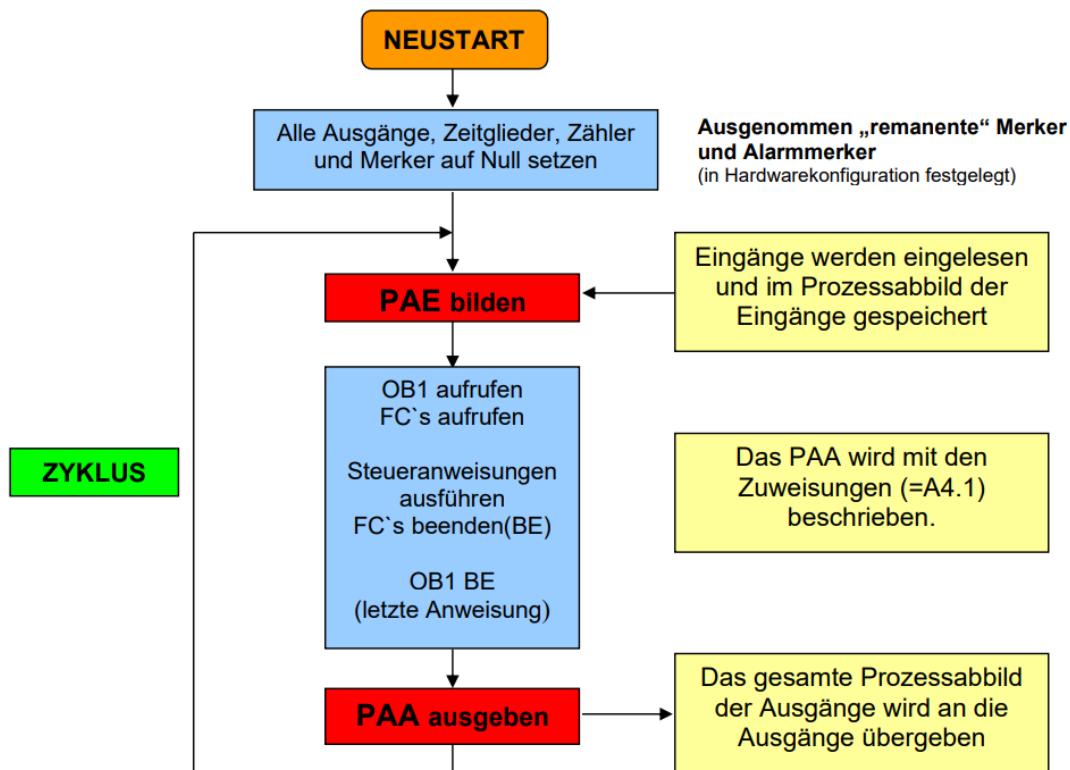


Abbildung 3.1: Arbeitsweise einer SPS [16]

noch Datentypen, ... maybe

### 3.2.2 Programmabausteine [17]

In TIA Portal werden Programmabausteine 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. 3.2 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 3.2: Organisationsbausteine [17]

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.

#### 2. FC(Funktionen)

Funktion 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.

### 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.

### 4. Global-DB(Datenbausteine)

Globale Datenbausteine speichern variable Daten, die dem kompletten Programm zur Verfügung stehen. Wie in Abb.3.3 ersichtlich, bedeutet das ,dass alle Bausteine Zugriff auf den Global-DB haben.

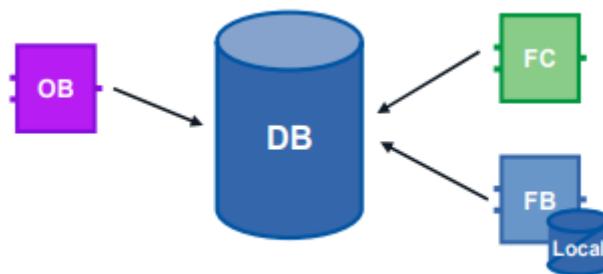


Abbildung 3.3: Zugriff auf Global-DB [17]

In Globalen Datenbausteinen können jegliche Datentypen genutzt werden.

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

### 5. Instanzen

Wird ein Funktionsbaustein aufgerufen, so nennt man das Instanz. Die Daten der Instanz, werden im sogenannten Instanzdatenbaustein gespeichert. Instanz-DBs werden automatisch nach den Vorgaben des Funktionsbausteins erzeugt und können somit nicht direkt geändert werden. Der Instanz-DB hat einen dauerhaften Speicher, welcher die Schnittstellen Input, Output, InOut sowie Static beinhaltet. Zusätzlich besitzt der Instanz-DB einen flüchtigen Datenspeicher in dem temporäre Variablen gespeichert werden. Diese sind dadurch immer nur für einen Zyklus gültig.

### 6. Multiinstanzen

Bei Multiinstanzen speichert der Funktionsbaustein seine Daten in den Instanz-DB

des übergeordneten Funktionsbaustein. Das heißt es wird in einem FB ein anderer FB aufgerufen. Dieser speichert seine Daten dann im Instanz-Db des Funktionsbausteins, welcher ihn aufgerufen hat. Multiinstanzen helfen das Programm übersichtlicher sowie strukturierter zu halten, da man mehrere Instanzen in einer Instanz vereint.

### 3.2.3 Technologieobjekte [18]

Technologieobjekte dienen dazu die Ansteuerung und Handhabung von technischen Funktionen, insbesondere von Motoren, Achsen, etc. zu vereinfachen. Es existieren eine Vielzahl an unterschiedlichen Technologieobjekten. In nachfolgenden Absätzen werden die für das Projekt relevanten Technologieobjekte genauer erklärt.

#### 1. Positionierachse (PositioningAxis)

Dieses Technologieobjekt dient zur genauen Positionierung einer Achse, sowie der Rückmeldung der aktuellen Achsposition. Zusätzlich wird die Zielposition automatisch gehalten.

Für die Positionierachse stehen folgende Motion Control Anweisungen zur Verfügung:

- Home  
Aktives oder passives Referenzieren der Achse.
- MoveAbsolut  
Fahren der Achse auf eine absolute Position.
- MoveRelativ  
Fahren der Achse auf eine Position relativ zur aktuellen Position.
- MoveSuperimposed  
Starten einer überlagerten Bewegung zu einer bereits laufenden Bewegung.
- TorqueLimiting  
Aktivieren einer Momentbegrenzung oder Festanschlagserkennung.
- SetSensor  
Umschalten des Gebers für die Achse.

#### 2. Gleichlaufachse (SynchronousAxis)

Das Technologieobjekt Gleichlaufachse enthält alle Funktionen des Technologieobjekts Positionierachse. Zusätzlich lässt sich die Achse mit einer Leitachse verschalten, sodass diese der Positionsänderung der Leitachse folgt. Dieses Technologieobjekt wird verwendet um synchrone bzw. positionsabhängige Bearbeitungsvorgänge auszuführen.

Der Gleichlaufachse stehen folgende zusätzliche Motion Control Anweisungen zur Verfügung:

- GearIn  
Starten eines relativen Gleichlaufs einer Leit- und Fogeachse.

- GearInPos  
Starten eines absoluten Gleichlaufs einer Leit- und Fogeachse unter Vorgabe einer Synchronposition.
- PhasingAbsolut  
Absolutes Verschieben des Leitwertbeugs während eines aktiven Gleichlaufs.
- PhasingRelativ  
Relatives Verschieben des Leitwertbeugs während eines aktiven Gleichlaufs.
- CamIn  
Start eines absoluten Kurvenscheibengleichlaufs.
- SynchronizedMotionSimulation  
Simulation eines aktiven Gleichlaufs.

### 3.2.4 Programmiersprachen

In Tia Portal stehen unterschiedliche Programmiersprachen zu Verfügung. Je nach Präferenz bzw. Aufgabe ist das Nutzen der richtigen Sprache von Vorteil. Daher folgt hier eine Aufstellung der möglichen Programmiersprachen, welche sich in textbasierte(tb.) oder graphische(gr.) Sprachen unterteilen.

#### 1. Funktionsplan(FUP), gr.

FUP ist eine graphisch augebaute Programmiersprache. Sie besteht aus unterschiedlichen Bausteinen, in Blockdarstellung, welche graphisch durch Linien verknüpft werden. Die Signalverarbeitung bei FUP läuft von links nach rechts. Die Programmierlogik in FUP ist übersichtlich und schnell nachzuvollziehen, weswegen diese Sprache für Anfänger relativ gut geeignet ist. [19]

#### 2. Kontaktplan(KOP), gr.

Der Kontaktplan ähnelt einem Stromlaufplan, der anstatt von oben nach unten von links nach rechts verläuft. Für die Programmierung werden Symbole wie Öffner, Schließer und Ausgänge verwendet. Da nicht für jeden Baustein ein Symbol verfügbar ist, werden solche Bausteine in FUP dargestellt. Der logische Verlauf der Schaltung ist dabei von links nach rechts und von oben nach unten. [19]

#### 3. Anweisungsliste(AWL), tb.

AWL ist eine textbasierte Programmiersprache, welche an Assembler angelehnt ist. Die Programmiersprache AWL wird hauptsächlich zur logischen Verknüpfung von Ein- und Ausgängen verwendet. In AWL werden Anweisungen in der Reihenfolge geschrieben, in der sie ausgeführt werden sollen. Da AWL für die Programmierung von größeren Projekten eher ungeeignet ist, wird es in neueren Programmen immer weniger verwendet. [20]

#### 4. S7-Graph, gr.

S7-Graph wird verwendet um Ablaufsteuerungen übersichtlich und schnell zu programmieren. Die zu ausführenden Aktionen werden in unterschiedliche Einzelschritte aufgeteilt. Zwischen diesen Einzelschritten befinden sich Transitionen. Transitionen sind Weiterschaltbedingungen welche erfüllt werden müssen damit zum nächsten Schritt weitergeschaltet wird. Der Ablauf der Steuerung erfolgt von oben nach unten.

#### 5. Structered Code Language (SCL), tb.

SCL ist eine höhere Programmiersprache, welche sich an Pascal orientiert. SCL ist für mathematische Funktionen, sowie das Programmieren von Schleifen oder if-Bedingungen besser geeignet als andere Programmiersprachen der SPS. In SCL hat man trotzdem Zugriff auf die typischen Elemente einer SPS, wie Eingänge, Ausgänge, Zeiten, Merker, Bausteinaufrufe usw. [21]

### 3.2.5 Bibliotheken

In Tia Portal sind nicht alle Funktionen integriert zur welcher die SPS fähig wäre. Deswegen gibt es Bibliotheken um projektspezifische Funktionen in das Programm zu integrieren. Bibliotheken können zusätzlich dazu genutzt werden um projektspezifische Bausteine oder Funktionen auch für andere Programme zugänglich zu machen. Generell kann man Bibliotheken in zwei unterschiedliche Arten unterteilen: Projektbibliothek (Project library) und Global Bibliothek (Global library). Projektbibliotheken sind im Projekt integriert und werden im Projekt verwaltet. Dies ermöglicht eine Wiederverwendung von Bausteinen, Funktionen, usw. innerhalb des Programms. Globale Bibliotheken hingegen sind projektunabhängig und können deshalb innerhalb von mehreren Projekten verwendet werden. In Bibliotheken gibt es dann wieder zwei unterschiedliche Möglichkeiten der Ablagerung. Man unterscheidet zwischen Kopiervorlagen (Master copies) und Typen (Types). Elemente die in "Kopiervorlagen" gespeichert sind, sind mit dem kopierten Element nicht verbunden. Typen hingegen sind mit ihren Verwendungsstellen im Projekt verbunden. Das heißt wenn Type verändert werden, wird dies im Projekt automatisch aktualisiert. Im Falle, dass ein Typ gelöscht wird, werden alle Verwendungen automatisch mitgelöscht. [17]

#### Wie werden Bibliotheken ins Projekt eingebunden?

Bibliotheken in Tia Portal hinzuzufügen ist intuitiv und simpel. Der erste Schritt ist das Downloaden der richtigen Bibliothek. Diese können Online auf Websites, von Siemens, gefunden werden. Um diese jedoch herunterzuladen braucht man einen dazu berechtigten Siemens-Account. Ist die richtige Bibliothek gefunden und heruntergeladen, kommt man zum zweiten Schritt. Zuerst wird Tia Portal geöffnet. Dann geht man auf die Projektansicht. Daraufhin findet man ganz rechts, als vierten Reiter von oben den Punkt Bibliotheken. Klickt an auf diesen erscheint ein Fenster, welches der Höhe nach in zwei Teile aufgeteilt ist. Die obere Hälfte ist für die Projektbibliothek. In der unteren Hälfte findet man die Globalen Bibliotheken. Um nun eine neue Bibliothek einzubinden klickt man auf das zweite Symbol von links

"Globale Bibliothek öffnen". Das Symbol ist ein Buch mit einem grünen Pfeil rechts oben. Jetzt öffnet sich wieder ein Fenster. Hier ist dann der Ordner mit der zu integrierenden Bibliothek auszuwählen und anschließend die Datei mit dem Datenformat .alxx (xx ... Version Tia-Portal (18)) zu öffnen. Anschließend ist die gewünschte Bibliothek verfügbar. Jegliche Bausteine die diese Bibliothek zur Verfügung stellt, können nun per Drag-and-Drop ins Programm gezogen werden. [22]

### 3.3 Motoransteuerung

Eine weitere Hauptaufgabe der SPS ist es die Ansteuerung der Motoren zu vollziehen. Hierbei stehen die Daten zur Verfügung, welche vom Server geschickt werden. Dies beinhaltet für die Motoransteuerung folgende Aufgaben: Positionsfahren (Ein/Auslagerung), Ansteuerung des Motors des Querförderers sowie die Ansteuerung der Asynchronmaschine des Förderbandes. Bei allen Motoren, bis auf den Motor für das Förderband, handelt es sich um Schrittmotoren. Es werden zwei unterschiedliche Motortypen verwendet: Nema 23 Closed Loop und Nema 17, wobei Nema 23 der stärkere der beiden Motoren ist. Die Nema 23 Motoren werden für das Ansteuern der X- und Y-Achse verwendet. Der Querförderer sowie die Z-Achse über die Nema 17 Motoren. Zu bedenken ist, dass jede Achse von zwei Motoren angetrieben wird und sich daher eine Anzahl von 7 Schrittmotoren plus einen Asynchronmotor ergibt. Für jeden der Schrittmotoren gibt es eine Schrittmotortreiber. Dieser regelt den Motor anhand der Kontrollsiganle, die dieser von der SPS bekommt.

#### 3.3.1 Kontrollsiganle für den Motortreiber [23]

Die zwei Hauptsignale die der Treiber von der SPS bekommt sind ein PTO-Signal sowie ein boolsches Signal welches für die Drehrichtung verwendet wird (Direktion). Zusätzlich dazu gibt es ein zweites boolsches Signal für die Freigabe des Motors (Enable). Ein Signal kommt jedoch vom Motortreiber. Nämlich ein Alarmsignal (Alarm), welches gesendet wird, wenn der Motortreiber folgende Fehler erfährt: Überspannung, Überstrom, Kurzschluss sowie Positionsschleppfehler.

##### 1. Pulse Train Output (PTO)

Das PTO-Signal besteht aus verschiedenen Pulsen mit Ein- zu Auszeit im Verhältnis 50% : 50%. Für jeden dieser Pulse bewegt sich der Motor um einen Schritt. Wie groß dieser Schritt ist hängt davon ab wie viele Schritte es benötigt um eine volle Umdrehung des Motors zu vollziehen. Dies kann man am Schrittmotortreiber einstellen. Bei diesem Projekt ist eine Umdrehung des Motors in 1600 Schritte eingeteilt. Das heißt das PTO Modul muss 1600 solcher Pulse schicken damit sich der Motor einmal dreht. Das heißt die SPS regelt die Frequenz mit welcher die Pulse hinausgeschickt werden und somit die Geschwindigkeit des Motors.

##### 2. Direktion

Dieses Signal ist wie bereits erwähnt ein boolsches Signal. Dies bedeutet, dass das

Signal nur zwei Zustände besitzt. Entweder HIGH oder LOW. Diese Signal bestimmt die Richtung, in welche sich der Motor dreht. Entweder im Uhrzeigersinn (CW) oder gegen den Uhrzeigersinn (CCW). Bei einem HIGH Signal dreht der Motor gegen den Uhrzeigersinn und bei einem LOW Signal im Uhrzeigersinn.

### 3. Enable

Hat wie das Direktion-Signal zwei Zustände: HIGH und LOW. Dieses Signal ist jedoch logisch invertiert. Dies bedeutet das bei einem HIGH Signal der Motor gesperrt ist. Der Motor ist nur bei einem LOW-Signal freigegeben.

### 4. Alarm

Auch ein boolsches Signal mit HIGH und LOW. Sobald der Motortreiber oben genannte Probleme erfährt, wird dieses Signal auf HIGH gesetzt. So sieht man auch in der SPS, dass es zu einem Fehler am Motortreiber gekommen ist.

Um falsche Ausführungen sowie Abweichungen zu vermeiden, müssen die in Abb. 3.4 gezeigten ,minimal und maximal Werte, eingehalten werden. Damit ein Signal als HIGH aufgefasst wird muss es über 3,5 V haben. Soll es jedoch als LOW gesehen werden, so muss die Spannung des Signals unter 0,5 V fallen. Die Periode des PTO Signals muss länger als 5 µs sein, um eine sichere Abarbeitung zu garantieren. Auch darf die Pulsweite, sowie die Zeit die der Puls auf LOW ist, nicht unter 2,5 µs fallen. Zusätzlich muss das Enable-Signal mindestens 5 µs vor dem Direktion Signal und 10 µs vor dem PTO Signal am Motortreiber anstehen. Wie in Abb. 3.4 zu sehen, muss das Direktion-Signal auch mindestens 5 µs vor dem PTO Signal ankommen.

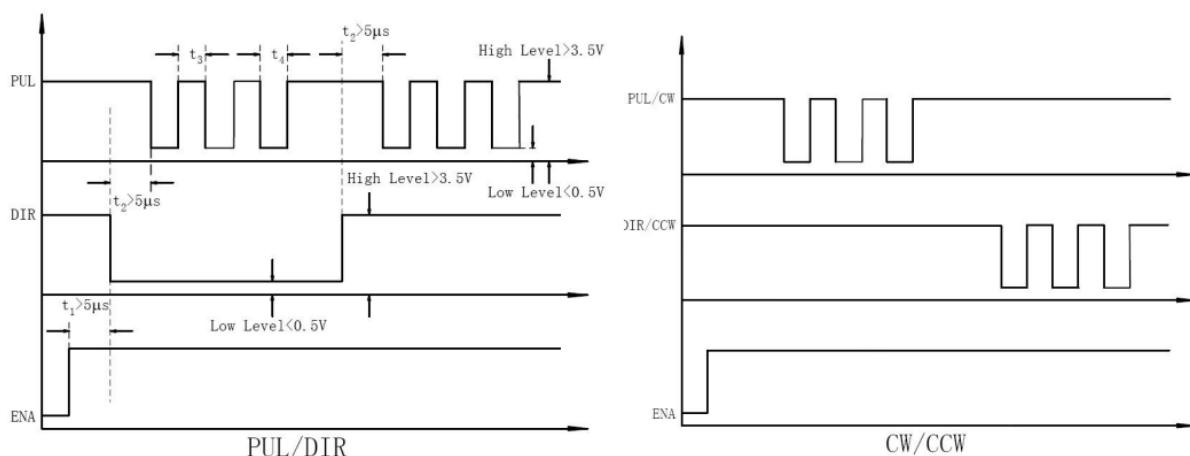


Abbildung 3.4: Sequenzdiagramm der Kontrollsignale [23]

### 3.3.2 Motion Control Anweisungen

Um den Motor in der SPS durch das Programm steuern zu können, wird zuerst das Technologieobjekt für diesen Motor erstellt und mit den richtigen Daten ausgefüllt. Danach ist dieses

Technologieobjekt durch Motion Control Anweisungen steuerbar. In folgenden Abschnitten werden alle für das Projekt relevanten Motion Control Anweisungen erklärt.

#### 1. MC\_Power

Diese Motion Control Anweisung gibt ein Technologieobjekt frei beziehungsweise sperrt es. Wird der Parameter enable auf HIGH gesetzt so wird das Technologieobjekt freigegeben. Das Technologieobjekt ist solange freigegeben, solange enable auf HIGH ist. Am Parameter Status kann man dann erkennen ob das Technologieobjekt freigegeben wurde oder nicht.

#### 2. MC\_Home

Wird verwendet um eine Achse zu referenzieren. Die Dynamikwerte mit welcher dieser Baustein ausgeführt wird, sind jene die in der Konfiguration des Technologieobjekts voreingestellt wurden. Es wird unterschiedlich referenziert, je nach dem welcher Wert (0-13) am Parameter Mode anliegt. Ausgeführt wird diese Motion Control Anweisung sobald eine steigende Flanke am Parameter execute erfasst wird. Referenziert ist die Achse erst wenn der Parameter Done auf HIGH umschaltet.

#### 3. MC\_MoveAbsolut

Mit dieser Anweisung wird eine Achse auf eine absolute Position gefahren. Die Parameter Velocity (Geschwindigkeit), Jerk (Ruck), Acceleration (Beschleunigung) und Deceleration (Verzögerung) können bestimmt werden. Werden diese nicht ausgefüllt oder mit einem Wert kleiner null ausgefüllt so werden die voreingestellten Werte übernommen. Diese Motion Control Anweisung kann nur ausgeführt werden, wenn das Technologieobjekt bereits referenziert ist. Wird ausgeführt sobal am Parameter execute eine steigende Flanke registriert wird. Wenn die Zielposition erreicht wird, schaltet der Parameter Done auf HIGH.

#### 4. MC\_GearIn

Startet einen Getriebegleichlauf zwischen einer Leit- und einer Folgeachse. Dies kann sowohl in Bewegung als auch im Stillstand erfolgen. Als Leitachse können folgende Technologieobjekte ausgewählt werden: Positionierachse, Gleichlaufachse, Externer Geber und Leitachsstellvertreter. Als Folgeachse kann jedoch nur das Technologieobjekt Gleichlaufachse agieren. Der Auftrag wird gestartet sobald am Parameter execute eine steigende Flanke kommt. Die Synchronisation ist fertiggestellt, wenn der Parameter InGear auf HIGH schaltet.

### 3.3.3 Konfiguration Technologieobjekt

Um die Technologieobjekte nutzen zu können müssen die korrekt konfiguriert werden. In nachfolgenden Absätzen wird Schritt für Schritt erklärt wie man ein solches Technologieobjekt für den Nema23 Motor konfiguriert.

## Positionierachse

1. Technologieobjekte
2. Neues Objekt hinzufügen
3. Motion Control
4. TO\_PositioningAxis
5. Konfiguration
  - Antrieb → Local Modules → PTO Modul auszuwählen
  - Mechanik → Spindelsteigung: = 134,9942
  - Dynamik-Voreinstellung:
    - Geschwindigkeit: 300 mm/s
    - Beschleunigung: 6000 mm/s<sup>2</sup>
    - Verzögerung: 6000 mm/s<sup>2</sup>
  - Positionsgrenzen
    - HW-Endschalter aktivieren
    - Variablen wählen
    - Pegelauswahl: oberer Pegel
  - Dynamikgrenzen
    - Geschwindigkeit: 1000 mm/s
    - Beschleunigung: 20 000 mm/s<sup>2</sup>
    - Verzögerung: 20 000 mm/s<sup>2</sup>
  - Referenzieren → aktives Referenzieren
    - Variable wählen
    - Pegelauswahl: oberer Pegel
    - Referenzierrichtung: Positiv
    - Referenziergeschwindigkeit: 25 mm/s

## Gleichlaufachse

Bei der Gleichlaufachse muss bei der Konfiguration ein Schritt mehr gemacht werden. Unter Konfiguration - Leitwertverschaltung wird eine Leitwertverschaltung hinzugefügt in dem man auf "hinzufügen" klickt und anschließend das Technologieobjekt auswählt dem die Gleichlaufachse folgen soll.

### 3.3.4 Synchronisation

Die X,Y,Z -Achsen werden von jeweils zwei Motoren angetrieben. Daher müssen die zwei Motoren ,welche zusammenarbeiten, synchronisiert werden. Dies wird in der Software, durch die Motion Control Anweisung MC\_GearIn, geschafft. Bei dem Slave muss es sich wie bereits gesagt um eine Gleichlaufachse (TO\_SynchronousAxis) handeln. Da die beiden Achsen hardware technisch verbunden sind, muss die Synchronisation im Stillstand erfolgen. Ausgeführt wird die Synchronisation bei einer steigenden Flanke am Parameter Execute. Die Folgeachse bleibt nun solange synchron zur Leitachse, bis es entweder, zu einem MC\_GearOut Auftrag kommt oder die Freigabe der Folgeachse, durch ein LOW Signal am Parameter enable des MC\_Power Bausteins (Folgeachse), nicht mehr vorhanden ist. Ein Sperren der Leitachse über MC\_Power (Leitachse) führt nicht zu einem Synchronisationsverlust. Die voreingestellten Werte bei den Parametern RationNumerator(1) sowie Ratio-Denominator(1) können übernommen werden. Diese werden nur geändert falls zwischen den zu synchronisierenden Motoren, ein Getriebe vorhanden ist, was bei diesem Projekt jedoch nicht der Fall ist. Ansonsten bleibt der Getriebefaktor bei 1.

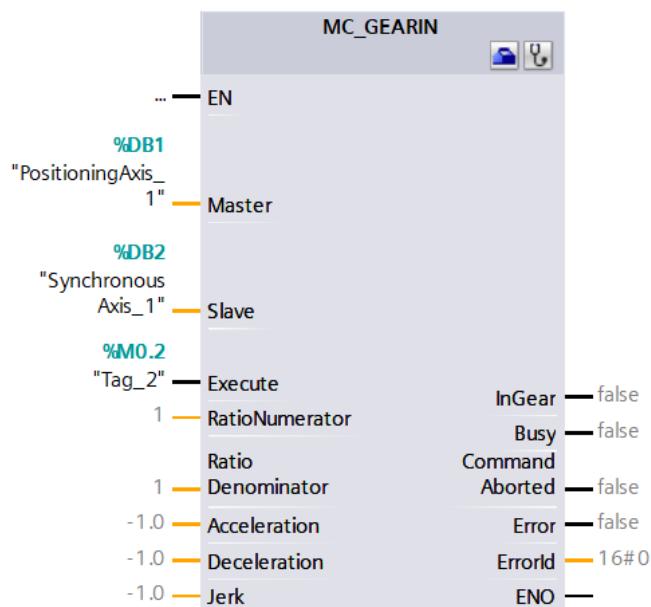


Abbildung 3.5: Baustein MC\_GearIn

### 3.3.5 Ansteuerung des Förderbands

Die Ansteuerung des Förderbands erfolgt im Gegensatz zu den anderen Motoren nicht über Motortreiber. Die Asynchronmaschine des Förderbands wird schlicht und einfach über Relais mit der Spannungsversorgung verbunden oder getrennt. Die Ansteuerung der Relais erfolgt dann über die Digitalausgänge der SPS.

### 3.3.6 Tia-Portal Programm

Durch die SPS-Server Kommunikation stehen die benötigten Daten sowie der Auftrag, welcher ausgeführt werden muss zu Verfügung.

**Ein-/Auslagerung** Bei dieser Aufgabe stehen als Informationen die X,Y,Z -Werte der Lagerposition zu Verfügung und ob es sich um eine Ein- oder Auslagerung handelt. Bei einer Einlagerung muss die Box zuerst beim Querförderer geholt werden. Dann wird die Box auf die Zielposition gelagert.

## 3.4 SPS-Server Kommunikation

Da die Lagerlogik auf einem Server ausgeführt wird und die SPS nur das ausführende Element ist, muss eine Verbindung zwischen SPS und Server aufgebaut werden. Das Ziel dieser Verbindung ist es die nötigen Informationen, sowie die Übertragung der Aufgabenstellung auszuführen.

### 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.: Webbrower) Anfragen an einen Server sendet, welcher anschließend die entsprechenden Daten zurückschickt. Die Anfrage wird als HTTP Request und die Antwort als HTTP Response bezeichnet.[24]

- **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. [25]

## Funktionsweise

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

Eine Kommunikation mit dem HTTP Protokoll findet wie folgt statt. Wie oben bereits erwähnt funktioniert das HTTP- Protokoll nach dem Anforderungs-Antwort-Prinzip. Wie in Abb. 3.6 ersichtlich wird als Erstes eine Anfrage vom Client zum Server gesendet (HTTP-Request). Daraufhin wird diese Anfrage vom Server bearbeitet und dieser sendet dann die geforderten Informationen zurück (HTTP-Response). Danach ist die Verbindung beendet. [26]

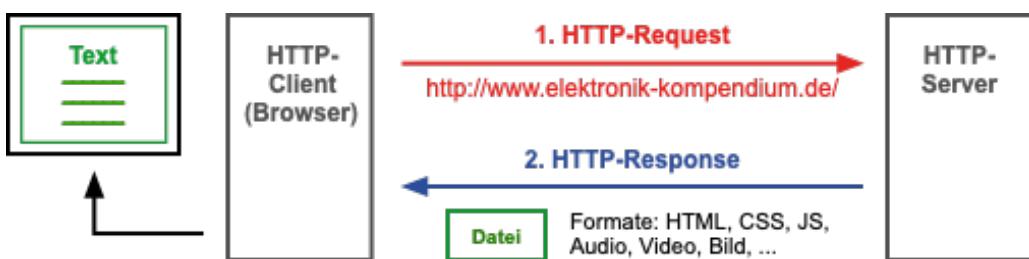


Abbildung 3.6: HTTP-Client-Server Kommunikation [26]

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

Auch bei OPC UA gibt es Server und Clients. Dabei stellt der Server Daten bereit und die Clients können diese Daten abfragen oder Werte überschreiben. Dies ermöglicht einen sichere sowie zuverlässige Datenübertragung. [25]

### Welches Protokoll wurde gewählt?

Bei diesem Projekt wurde das HTTP Protokoll ausgewählt um die Kommunikation zwischen SPS und Server auszuführen. Das HTTP Protokoll wurde ausgewählt, da die SPS direkt mit dem Server kommunizieren kann. Bei OPC-UA hätte ein zusätzlicher Server gehostet werden müssen. Dieser OPC-UA Server hätte dann mit dem eigentlichen Server, welcher die Lagerlogik übernimmt, kommuniziert. Die SPS hätte dann auf den OPC-UA Server zugegriffen und nicht direkt auf den Zielserver. Um diesen zusätzlichen Aufwand sowie weitere Fehlerquellen zu vermeiden, wurde das HTTP Protokoll ausgewählt.

#### 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 für das HTTP Protokoll stehen dann folgende Bausteine zur Verfügung:

- GET
- POST-PUT

Als Baustein zur Verbindungsherstellung wurde der POST-Befehl ausgewählt, da mit diesem unbegrenzte Datenmengen geschickt werden können. Damit der Baustein nun eine Verbindung herstellen kann müssen folgende Parameter angegeben werden: **URL** [string], die zu schickenden Daten **data** [string] und ein Speicherplatz für die empfangenen Daten **response Data** [Array of Char]. Die restlichen Eingabeparameter, welche in Abb. 3.7 ersichtlich sind, müssen nicht unbedingt ausgefüllt werden, sondern werden automatisch vorausgefüllt. Standardmäßig ist die richtige Methode (POST) schon ausgewählt, da die Eingabe **method** automatisch mit **0** ausgefüllt wird.

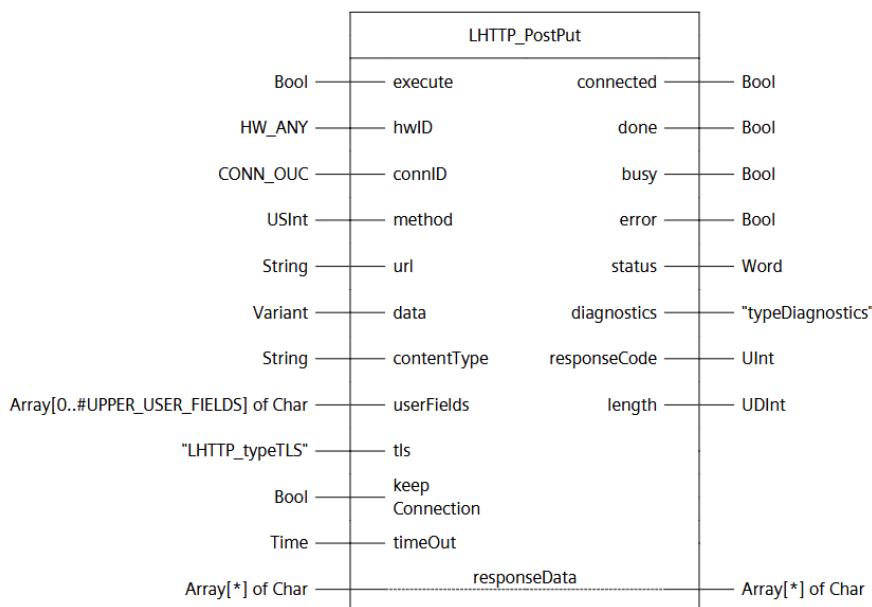


Abbildung 3.7: POST-Put Baustein [27]

Wird nun die Eingabe **execute** HIGH, so wird der Baustein einmalig ausgeführt. Rechnet man mit einem häufigeren Datenaustausch, mit Hilfe dieses Bausteins, so empfiehlt es sich den Parameter **keep Connection** auf **true** zu setzen. Dadurch bleibt die Verbindung mit dem Zielserver erhalten und weitere Kommunikationen werden schneller ausgeführt.

### 3.4.3 Datenaustausch

Die SPS und der Server arbeiten mit verschiedenen Datentypen. Bei dem Datenaustausch erwartet der Server die Daten in einem json-Format. Deswegen müssen die Daten welche gesendet werden zuerst in dieses Format gebracht werden. Um dies in Tia-Portal zu realisieren wird die Bibliothek LStream eingebunden. Mit dieser werden dann Daten aus einem

json-Datenbaum in ein Array of Byte umgewandelt. Dieses Array of Byte wird anschließend in den Datentyp string umgewandelt. Nun kann dieser string an den POST-Datenbaustein weitergegeben werden.

Ist nun der Datenaustausch abgeschlossen, so sind die empfangenen Daten als Array of Char verfügbar. Da man in Tia-Portal nur umständlich mit diesem Datenformat umgehen kann, wird das Array of Char nun in das Datenformat string umgewandelt. Die Daten sind nun noch immer nicht einzeln verfügbar, weswegen man nun diesen string nach den Daten filtern muss.

### 3.4.4 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 welche erforderlich sind um diese Aufgabe auszuführen.

#### Datenformatierung

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

In folgenden Punkten dient x immer als Platzhalter für eine Nummer null bis neun.

- 1.Teil:

IDxxxxAxx

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 zum Ersten. Je nachdem welche Zahl nach A steht, also der Code welche Area angesprochen wird, ist der zweite Teil anders aufgebaut.

- 00: SB

Bei diesem Befehl muss der Barcodescanner angesteuert werden um an der Kommissionierstation einen Barcode einzuscannen.

- 01: P ±xxxx

Gibt an um wie viel sich das Förderband (in mm) bewegen muss. Vor der Zahl

steht jedoch entweder ein plus oder minus um festzulegen in welche Richtung sich das Förderband bewegen muss.

- 10: XXXXXYXXXXZXXXXRx

Nach jeder Achse X,Y,Z steht ein vierstelliger Wert, welcher angibt auf welcher Position sich der gewünschte Lagerplatz befindet. Der Wert nach R ist entweder 0 oder 1 und gibt an ob es sich um eine Einlagerung (1) oder Auslagerung (0) handelt.

- 11: PXXXX

Nach P steht ebenso ein vierstelliger Wert, der angibt auf welche Position der Querförderer gefahren werden muss.

### 3.4.5 Tia-Portal Programm

Im Programm werden zwei Funktionsbausteine aufgerufen. Nämlich FB\_Kommunikation und FB\_Datenfiltern. FB\_Kommunikation übernimmt den Datenaustausch zwischen Server und SPS. In diesem Baustein werden zuerst die Daten die an den Server geschickt werden, wie in Punkt 3.4.3 erklärt ins richtige Datenformat(string) gebracht. Darufhin wird der HTTP POST-Aufruf ausgeführt und die empfangenen Daten wie in 3.4.3 gezeigt wieder ins richtige Datenformat gebracht. Somit ist die Aufgabe dieses Datenbausteins erfüllt. Nun erfolgt der Aufruf des zweiten Funktionsbausteins FB\_Datenfiltern. Dieser Funktionsbaustein ist in unterschiedliche Funktionsbausteine aufgeteilt. Der erste dieser Funktionsbausteine filtert nach ID sowie der Area (Aufgabe). Darufhin wird der Baustein FB\_Kommunikation innerhalb des Bausteins FB\_Datenfiltern aufgerufen, um sofort die ID-Nummer an den Server zurück zu schicken, damit dieser weiß das die SPS den Auftrag erhalten hat. Die restlichen Bausteine die im FB\_Datenfiltern enthalten sind, warten darauf, dass sie von ihrer Area, beispielsweise 01, angesprochen werden. Sollte dies der Fall sein so Filtern sie die benötigten Informationen um den Auftrag auszuführen. Nach dem Aufruf beider Bausteine sind nun die geforderten Informationen verfügbar. Bei beiden Funktionsbausteinen handelt es sich um Multiinstanzen, um das Programm übersichtlicher zu halten.

#### Wie schaut so ein Baustein aus? (am Beispiel “ID + Area”)

Alle Bausteine zur Datenfilterung wurden in SCL programmiert. Als Eingabe ist der Datenstring welcher die Daten beinhaltet. In diesem wird zuerst nach ID gefiltert (Zeile 1). Dies liefert die Position von ID innerhalb ddes Strings. Darufhin werden alle Zeichen links von ID + ID wegelöscht (Zeilen 2 - 5). Nun weiss man das rechts von ID eine vierstellige Nummer (ID-Nummer) steht. Da alles links von dieser Nummer weggeschnitten wurde, weiß man das diese Nummer ganz links im string ist. Deswegen werden die ersten vier Stellen von links genommen und dem Ausgang Q\_Info\_ID zugewiesen (Zeile 6). Dieser Ausgang ist für das gesamte Programm erreichbar. Das selbe Suchen und Löschen der Zeichen links vom Gesuchten wird ebenfalls für A (Area / Auftrag) erledigt (Zeilen 10 - 13). Da man den Wert von Area, welcher den Auftrag definiert, vergleichen will um den richtigen Baustein auszuführen,

werden die zwei Zeichen links von A, von denen wir wissen das es Zahlen sind, in das Datenformat int konvertiert und dem Ein-Ausgang IQ\_Info\_Area zugewiesen (Zeilen 19-23).

```
1      #T_Int := FIND(IN1 := #I_Datenstring, IN2 := 'ID');
2 IF #T_Int > 0 THEN
3   #X_Info_ID := #T_Int + 1;
4   #X_String_bearbeitet := DELETE(IN := #I_Datenstring, L := #X_Info_ID, P
5   := 1); //Alles bis inkl. X wegschneiden
6
7   #Q_Info_ID := LEFT(IN := #X_String_bearbeitet, L := 4);           //
8   4 Zeichen links von X nehmen
9
10 #T_Int := FIND(IN1 := #I_Datenstring, IN2 := 'A');
11 IF #T_Int > 0 THEN
12   #X_Info_Area := #T_Int + 0;
13   #X_String_bearbeitet := DELETE(IN := #I_Datenstring, L := #X_Info_Area,
14   P := 1);
15
16   // Lager herausfinden          10: Lager 1      11: Puscher
17   (Lager 1)        00: Komissionierstation
18   //((zurzeit nur 1 Lager verfuegbar)    20: Lager 2      21: Puscher
19   (Lager 2)        01: Foerderband
20   // -->nur 00/01/10/11 relevant          30: Lager 3      31: Puscher
21   (Lager 3)
22
23   #Area_Info_1_string := LEFT(IN := #X_String_bearbeitet, L := 2);
24   STRG_VAL(IN := #Area_Info_1_string,
25   FORMAT := #Area_convert_Format_auswaehlen,
26   P := 1,
27   OUT => #IQ_Info_Area);
28
29 END_IF;
```

### 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. Die Elektrik des AFSS befindet sich in einem umgebauten Serverschrank, dessen physische Limitierungen bei der Planung ebenfalls zu berücksichtigen waren. Darunter fällt beispielsweise, 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, denn diese überliefern präzise Daten die nicht verzerrt werden sollen.

In der Planung wurde stets bedacht, dass die elektrischen Komponenten so verbaut werden, dass im Falle eines Fehlers sowohl Personen gut geschützt sind und betroffene 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 Dauersummenstrom von über 20A. Die Logik bestehend aus Siemens-SPS, mit Ein und Ausgangskarten sowie PTO-Karten, und einer ET200 mit Asi-Master. Die Logik benötigen ebenfalls 24 V und sollen getrennt versorgt werden, um von potentiellen Fehlern bei den Schrittmotoren geschützt zu sein. 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 (FI), ein Leitungsschutzschalter (LS), ein Motorschutzschalter und für jeden Schrittmotor eine Gleichstromsicherung ausgelegt werden.

Um die Anlage trotz Fehler, die potentiell von den elektrischen Schutzeinheiten nicht erkannt werden, nach wie vor Abschalten zu können soll die Anlage über mehrere Not-Aus-Schalter verfügen. Zwei auf der Anlage selsbt, einen im Serverschrank/Schalschrank und einen am Kommisionierplatz. Diese Positionierung soll es NutzerInnen ermöglichen aus jeder Position an der Anlage, einen Not-Aus-Schalter zu erreichen.

## Bedienelemente

Physische Bedienelemente wären beim AFSS ein Schlüsselschalter, zur Freigabe, und ein dreiphasiger Drehstromschalter, für eine manuelle Freischaltungsoption.

## Schalschrank

Grundsätzlich haben Schalschränke genormte Anforderungen (IEC 60208 und IEC 61439). 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, für den wir uns entschieden haben, schützt gegen Staub und Schutz und kommt mit einer Lüfteranlage, die die Abwärme von mehreren Gleichrichtern gut abführen kann. Zudem sind die Materialien des Schrankes vor Korrosion geschützt. [28]

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

## Kabelauslegung

Beim Auslegen von Kabeln gibt es mehrere Punkte, die beachtet werden müssen. Während Spannungsabfall bei den Längen des AFSS vernachlässigt, werden können muss besonders auf Schleppkettentauglichkeit geachtet werden. Steuer- und Messkabel müssen entsprechend geschirmt werden und abhängig vom Strom muss der Querschnitt gewählt werden. Dabei gehören die Querschnitte aber auch auf die Schutzautomaten im Stromkreis abgestimmt.

## Module

Die Paneele/Module, auf welchen die elektrischen Komponenten montiert werden, müssen ebenfalls alle Erdungserwartungen erfüllen und mechanisch den Belastungen standhalten. Dabei ist das Gewicht die gravierende Belastung. Eine gerechte Verdrahtung 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

#### Schalschrankrahmen

Bevor man sich den Details des Schalschrankes widmet, muss vorerst der Rahmen fest-



Abbildung 4.1: Serverschrank bei Übergabe an das AFSS-Team

gelegt werden. Grundsätzlich würde man einen herkömmlichen Schalschrank verwenden, allerdings sind diese teuer und erfüllen auch nicht unsere Anforderung der Mobilität. Deswegen wurden mehrere alternative Optionen in Betracht gezogen. Anfangs wurde der Ansatz verfolgt die elektrischen Komponenten in den Lagerschrank selbst einzubauen. Dabei hätte man entweder einen eigenen Abteil ins AFSS einplanen können, der auch mittels entsprechendem Material räumlich getrennt wäre, oder man hätte die Elemente fluide unterbringen können. Mit Letzteren ist gemeint, dass in dem ganzen Schrank verteilt die elektrischen

Komponenten montiert wären.

Die fluide Variante benötigt sorgfältige Planung und auch sorgfältige Absprache mit der restlichen mechanischen Planung des AFSS. Dafür hätte man eine kompakte Design aber das Risiko für Verletzungen und Schäden wäre höher, da man die elektrischen Komponenten nur schwer räumlich trennen könnte, so wie es die Wände eines Schaltschranks machen. Aufbauend auf dem fehlenden Sicherheitsaspekt und der Tatsache, dass die benötigte Kommunikation in einer Entwicklungsphase nicht möglich wäre, wurde sich gegen die fluide Option entschieden.

Wesentlich realistischer ist der Ansatz ein eigenes kleines Abteil in den Schrank einzubauen. Der Rahmen wäre, sowie das Lager, aus Aluminiumprofilen gebaut und die jeweiligen Seiten würden mit Kunststoffen verkleidet werden. Um die räumliche Trennung zu gewährleisten. Diese Option wäre Kommunikationstechnisch möglich, da man sich mit der restlichen mechanischen Planung nur auf die Außenmaße und Position dieses Abteils einigen müsste. Allerdings bedeutet ein intigriertes Abteil auch weniger Lagerplätze und einen merkbaren Zusatzaufwand in der Realisierung. Da so viele Lagerplätze wie möglich verwirklicht werden sollen wurde sich gegen diese Variante entschieden.

Auf der weiteren Suche nach einer Alternative zum herkömmlichen Schaltschrank wurde die Möglichkeit, einen alten Serverschrank zu recyceln erkannt. Ein fertig gebauter Schrank, der im Fehlerfall die Umgebung ausreichend schützt und keine Zusatzkosten mit sich bringt war die beste Option. Die im Inneneren bereits vorhandenen Profilschienen bieten viele Möglichkeiten für die Elektrik montiert zu weden. Zudem bietet der Innenraum des Serverschranks viel Platz und auch die Möglichkeit die Elektrik, wenn nötig, zügig zu erweitern, da man die Profilschienen sehr individuell nutzen kann.

Es wurde sich somit entschieden den Serverschrank zu einem Schaltschrank umzubauen. Im Zuge des Umbaus würde der Innenraum umgebaut werden und der Serverschrank müsste auch mobil gemacht werden.

## Modulprinzip

Im Zuge dieses Projektes wurde das Innenleben des Schaltschranks auf mehrere Module getrennt. Die Anforderungen an diese wurden schon beschrieben, doch ursprünglich waren weitere Alternativen für den Innenraum des Serverschranks in Diskussion.

Anstatt von mehreren Modulen, die später genauer beschrieben werden, könnte man eine durchgehende Platte verwenden und diese an die Profilschienen des Serverschranks festschrauben. Der große Vorteil einer durchgehenden Platte ist, man könnte die Elemente so anordnen, dass die Fläche von Leerräumen minimiert wird. Die große Platte entfällt als Möglichkeit allerdings insofern, da diese nicht in der Schule produzierbar gewesen wäre.

Eine andere Option wäre eine plattenlose, dabei würde man die Hutschienen direkt auf die Profilschienen des Serverschranks montieren. Man spaart sich so eine Platte und die Elemente könnten direkt auf die Hutschienen montiert werden. 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, wurde sich für ein Modulprinzip entschlossen. Dieses ermöglicht es allen Elementen, auch für die, die für Hutschienen ungeeignet sind, montiert zu werden und ist dennoch in der Schule produzierbar.

### Platten-Material

Für die Materialwahl gab es drei realistische Möglichkeiten. Die Modulplatten hätten vollständig aus Aluminium oder aus Dibond gefräst werden können. Als dritte Option hätte man die Platten aus einem Kunststoff laser oder fräsen können. 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. Beim einer reinen Kunststoffplatte gibt es keine Leitfähigkeit und zusätzlich bieten die meisten Kunststoffe keine ausreichende mechanische Stabilität.

Aluminium erfüllt alle Anforderungen, ist aber teuer und ein wertvoller Werkstoff. Da ein um-

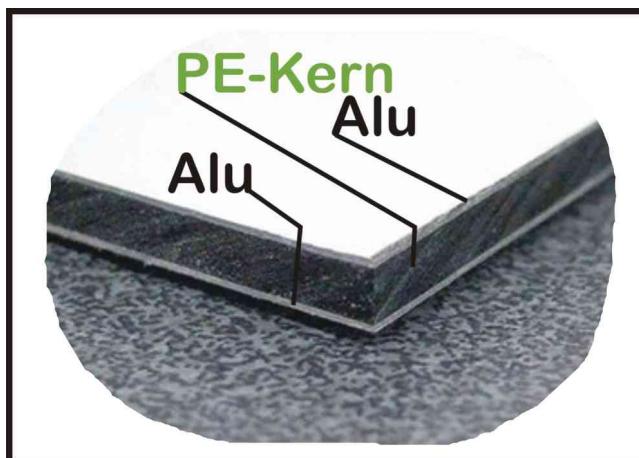


Abbildung 4.2: Dibond-Platte, Quelle: [29]

sichtiger Umgang mit Ressourcen wichtig ist wurden nach einer Alternative gesucht. Dibond wurde daraufhin als Projektstandard für die Module definiert. Dieser Stoff besteht aus zwei dünnen Aluminiumplatten die auf einen Kunststoff aufgepresst werden. Dibond bietet keine elektrische Leitfähigkeit, folglich müssen alle Elemente zusätzlich geerdet werden aber das leichte Gewicht und die hohe mechanische Stabilität machen Dibond zur besten Option.

### 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 ist 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 4.3). Es setzt auch ein breites Feld an Firmen auf dieses Prinzip. Firmen wie

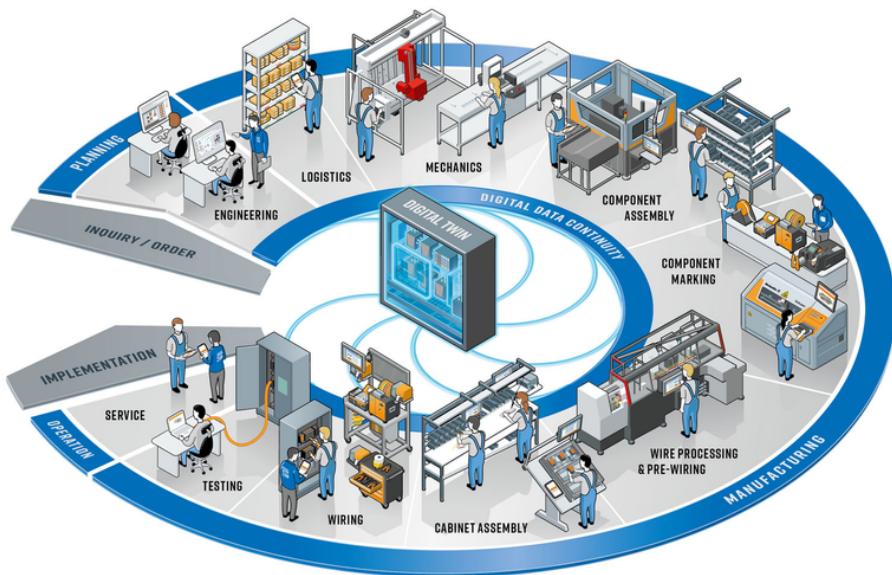


Abbildung 4.3: Digitaler Zwilling, Quelle: [30]

Weidmüller, Komax, Steinhauer und noch viele mehr haben eine Firmenzusammenarbeit, die ohne einen digitalen Zwilling nicht möglich wäre [31]. 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 Schaltschrankes möglichst nahe an die Praktiken aus der Industrie anzugeleichen, wird auch der Schaltschrank des AFSS mithilfe eines digitalen Zwillingen geplant. Dieser wird in Fusion360 gezeichnet und soll den Sollzustand des Schaltschrankes abbilden.

Um die Konstruktion anzufangen, braucht es eine möglichst ausführliche Ausmessung des bereits bestehenden Serverschrankes. 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 - Umsetzung

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



Abbildung 4.4: Sommerprototyp eines Serverschranks

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 Umfanges der Diplomarbeit sowie der begrenzten Zeit wird ein erster Entwurf eines Serverschranks 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.

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 elektrische Komponenten sollen auf dasselbe Modul kommen, und andererseits zusammenhängende Module sollen sich möglichst nahe sein.

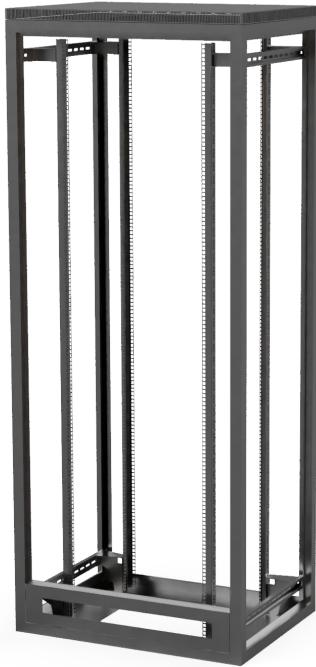


Abbildung 4.5: Serverschrank konstruiert

### 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 Hutschine 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.

### 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 gesammten 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 zur Erdung von Leiterschirmen geplant. Zudem sollen in dieses Modul vier rechteckige Ausfräslungen gemacht werden, in welche die Buchse des RJ45-Stecker hineinpasst.



Abbildung 4.6: Siemens SITOP in Fusion360

## Module - 3D Konstruktion

Die einzelnen Module wurden, nachdem die Konzeptionierung beendet wurde, in Fusion360 gezeichnet. Dafür wurden zuerst die einzelnen Komponenten wie Hutschinen, Verdrahtungskanäle, Gleichrichter und alle anderen Komponenten als unabhängige Konstruktion gezeichnet. Damit wurde eine breite Bibliothek an Komponenten erstellt, mit dieser kann man dann wesentlich einfacher den digitalen Schaltschrank bestücken. Bei den jeweiligen Elementen ist es nicht wichtig kleinere Details miteinzubeziehen, vielmehr die Außenmaße und die Positionen von Montagelöchern oder des Klemmmechanismuses für die Hutschiene sollen akkurat abgebildet sein (siehe 4.6).



Abbildung 4.7: Der Schaltschrank in Fusion gezeichnet

## Fusion 360

Nachdem man die einzelnen elektrischen Komponenten gezeichnet hat, wurde in Fusion die erste Platte gezeichnet. Hierbei muss man nicht direkt die richtige Größe einschätzen, da Fusion späte Änderungen gut zulässt. Nachdem man eine Platte im Serverschrank kon-

struiert hat, kann man die Hutschienen und Verdrahtungskanäle einfügen. Daraufhin kann man die Unterkonstruktion bestücken. Beim Einfügen von Konstruktionen in andere Konstruktionen bleiben die kopierten Elemente mit der originalen Zeichnung verknüpft. Diese Verknüpfung wurde bei den Elementen gelassen, die keine Änderungen erwarten doch bei Komponenten wie eines Verdrahtungskanals sollte man die Verknüpfung trennen um das Element zuschneiden zu können. Eine Verknüpfung bedeutet, dass Veränderungen nur in der ursprünglichen Zeichnung getätigt werden können. So wurde Modul nach Modul ausgeplant.

Beim Zeichnen fiel auf, dass die Module aus dem Serverschrank, beziehungsweise Schaltschrank, rausragten. Es wurde erkannt, dass die Profilschienen nach innen verschoben werden müssen um zu ermöglichen, dass sich die Wände des Schaltschranks auch schließen lassen. Die Bauart des Serverschranks erlaubt eine solche Veränderung.

Es wurden, bis auf das Erdungsmodul, alle Module auf diese Art konstruiert. Man konnte schön erkennen welche Reihenfolgen sinnvoll sind und wie sich welche Komponenten am besten montieren lassen können. Beispielsweise wurde ausgetestet ob bei gewissen elektrischen Komponenten eine vertikale Montage zielführend waren. In der Abbildung 4.7 sieht man die fertige Konstruktion des Schaltschranks.

Zu dieser Version des Schaltschranks ist zu sagen, dass ab der Realisierung des Geplanten, dieser digitale Plan sich fließend transformiert, zum digitalen Zwilling. Dieser Zwilling gehört nach der Realisierung konstant gepflegt. Das Erdungsmodul beispielsweise, welches erst mitten in der Realisierung eingeplant und gemacht wurde, muss in den digitalen Zwilling nachträglich hinzugefügt werden.

## AutoCAD

Nachdem der digitale Plan/Zwilling vollständig ausgeplant wurde, mussten die jeweiligen Platten gefräst werden. Um die Platten auch fräsen zu können müssen diese zuerst als DXF-Datei in Filou-NC eingefügt werden. Dazu mussten die einzelnen Module in AutoCAD nachgezeichnet werden. + Idealerweise hätte man hierfür nur die Oberfläche der Platten in Fusion zur Skizze gemacht und diese dann als DXF exportiert. Bei einem nachträglichen Ausmessen der 3D-Konstruktion wurde allerdings festgestellt, dass sich die Maße der Löcher der Profilschiene nicht mit der Realität decken. Zudem wurden weitere minimale Abweichungen festgestellt. Keiner der Fehler war so gravierend, dass man die digitale Konstruktion des Schaltschranks nicht weiterverwenden könnte aber für eine präzise Fräzung aufbauend auf der Fusion-Konstruktion waren die Abweichungen zu groß. Deswegen wurden alle Module in AutoCAD sorgfältig nachgezeichnet.

Beim Nachzeichnen wurden alle fehlerhaften Mäse ausgebessert. Mit den richtigen Masen wurde festgestellt, dass sich gewisse Module nicht wie geplant ausgingen. Beim Modul 6 gingen sich die vier Schrittmotortreiber nicht vertikal nebeneinander aus. Deswegen wurde in AutoCAD einer der Treiber vertikal angeordnet und weiterführend wurde die Änderung auch in Fusion nachgebessert.

## Räder

In der Transformation vom Serverschrank zum Schaltschrank des AFSS gehört auch, dass der Schrank mobil gemacht wird. Dazu wurden zwei flexible und zwei starre Räder zur Verfügung gestellt. Planungstechnisch wurden diesbezüglich besprochen, dass die Räder direkt an das bestehende Gerüst geschraubt werden. Montiert werden die Räder mit simplen Schrauben und Muttern. Die Räder wurden in AutoCAD nicht gezeichnet, da es keinen sinnerfüllten Zweck gab.

### 4.1.4 Elektrische Planung (E-Plan)

Für das Zeichnen eines E-Plans ist ein großes Produktwissen nötig. Um sich dieses zu beschaffen wurde die mechanische Seite zuerst geplant, da man im Zuge dieser die elektrischen Komponenten sehr gut kennenlernen kann. Als ein Großteil der mechanischen Seite für den Schaltschrank ausgeplant war wurde der E-Plan parallel zur mechanischen Planung angefangen. Als Schüler wurde die E-Plan Education Version verwendet. Diese Version ist kostenlos und bietet alle Funktionen die für die Planung des AFSS nötig sind.

#### E-Plan Allgemein

Die Firma E-Plan bietet verschiedenste Möglichkeiten zur Planung und Dokumentation von elektrischen Anlagen. Das Programm bietet auch die Möglichkeit eines Aufbauplanes, in diesem werden die elektrischen Komponenten der Anlage in einer 2D-Ansicht dargestellt. Diese Art des Aufbauplanes ist weit verbreitet und übersichtlich, doch die Stärke liegt beim herkömmlichen Schaltschrank. Da beim AFSS-Schaltschrank wesentlich mehr zu beachten ist wurde der Aufbauplan nicht in E-Plan sondern in Fusion 360 gezeichnet. Zu dieser Entscheidung kam es einerseits da der Serverschrank als ganzes umgeplant gehörte, zum Schaltschrank, und andererseits da die persönlichen Kenntnisse in Fusion wesentlich besser waren.

E-Plan bietet ebenfalls die Möglichkeit vieler Übersichten. Diese sind vor allem bei sehr großen Anlagen zur Orientierung dringend nötig. Da das AFSS aber eine eher kleinere Anlage ist wurde sich zusammen mit dem Kunden (der Werkstätte der HTL-Mössinger) darauf geeinigt, dass die Übersichtstabellen nicht nötig sind.

Der Fokus liegt daher auf dem reinem Schaltplan des AFSS. Für den grundsätzlichen Schaltplan bietet E-Plan mehrere Tools an. Einerseits gibt es die E-Plan Cloud, in dieser finden sich die meisten Geräte die am Markt erhältlich sind. Unvorteilhafterweise sind mehrere Geräte des AFSS aus einem älterem Jahrgang und damit zu alt für die E-Plan Cloud. Für diesen Fall bietet E-Plan die Möglichkeit ein eigenes Gerät anzulegen. Dazu muss ein Gerätekasten eingefügt werden und in diesen müssen daraufhin die Gräteanschlüsse gelegt werden. Zur besseren Verständnis kann man selbstgezeichnete Geräte noch mit gewissen Schaltzeichen versehen, um eine bessere Verständlichkeit zu garantieren. Weiterführend kann man in E-Plan alle vorstellbaren elektrischen Komponenten finden. Motoren, Geber, Schütze, Relais und noch vieles Mehr findet sich in der lokalen E-Plan Bibliothek. Mit E-Plan lassen sich

auch die benötigten Reihenklemmen herausfinden. Ab einem gewissen Maßstab kann man sich die Reihenklemmen nicht mehr denken und genau da hilft es, dass man in E-Plan die Klemmen genau planen muss. Das ausführliche Beschäftigen mit den Reihenklemmen ist wichtig, da diese im Fehlerfall der Anlage die Fehlersuche erleichtern. Für das Zeichnen eines Schaltplans ist es wichtig im konstanten Austausch mit der Sensorik und der Steuerungstechnik zu stehen, um eine korrekte und realistische Verdrahtung zu planen.

## Komponentenkenntnis

Wie bereits erwähnt braucht es für eine richtige E-Plan Zeichnung ein ausführliches Wissen über die elektrischen Komponenten der Anlage. Dafür ist es auch nötig, dass man weiß wo man sich informieren kann. Deswegen wurde eine Excel Tabelle angelegt, in welcher alle elektrischen Komponenten aufgelistet sind. Diese Liste dient primär zur elektrischen Planung, wurde aber schon angelegt, als die mechanische Seite des Schaltschranks geplant wurde. In dieser Liste wurden Links zu Datenblättern hinterlegt, sowie festgehalten ob das jeweilige Element im Schaltschrank oder am AFSS verbaut wird und auch mechanische Daten wie die Maximalwerte von Breite, Höhe und Tiefe wurden eingefügt. Die Excel-Tabelle beinhaltet weiters auch die Anzahl der jeweiligen Komponente, die Art der Montage (Hutschiene, Siemens-Profilschiene oder andersartige Montage) und die Gerätenummer wurde ebenfalls festgehalten. Diese Liste ist das wichtigste Dokument in der elektrischen Planung gewesen, da man sich konstat an die spezifischen Daten von beispielsweise einem Netzteil erinnern musste. Zu guter Letzt wurde in dieser Excel-Liste ebenfalls dokumentiert wie weit das Element fertiggeplant wurde. Das heißt, es wurde festgehalten ob es im E-Plan fertig war und ob es in der mechanischen Planung konstruiert wurde.

## Schaltplan - Strukturzugang

Es gibt keine allgemeine Orientierung bezüglich einer Schaltplanstruktur. Grundsätzlich gilt aber, das Schaltplan sollte nachvollziehbar gezeichnet werden. Das bedeutet zum Beispiel, dass thematisch zusammenpassende Seiten sollten sich nahe sein. Es wurde sich auch beim Zeichnen für den Schaltplan des AFSS bemüht nachvollziehbar zu bleiben. Zudem war der Zugang zur Struktur des Planes, mit dem Eingang (Starkstromanschluss) zu beginnen und dann dem Strom bis zu den Motoren zu folgen. Das heißt strukturtechnisch beginnt der Plan mit dem Starkstrom zu den Schutzorganen, geht über zu den Netzteilen dann zu den DC Sicherungen, daraufhin zu den SM-Treibern und zu letzt zu den Motoren. Der Sensorkreis mit ASI-Bus wurde auch zusammenhängend daraufhin gezeichnet, der Schaltplan in Detail wird in kürze beleuchtet. Kurz und Knapp war der Anspruch einen intuitiven Plan zu zeichnen der sich nach dem Stromfluss orientiert.

## Schaltplan - Zeichenprozess

E-Plan benötigt immer ein Vorlage-Projekt nach dem es sich orientieren kann, ein Basisprojekt. Für dieses wurde ein Projekt aus der Werkstatt verwendet namens "2022-09-HU Basisprojekt (1).zw9".

Der Schaltplan orientiert sich wie erwähnt am Stromfluss. Damit wurden zuerst der Fehlerstromschutzschalter und der Leitungsschutzschalter sowie der dreiphasige Drehschalter gezeichnet. Diese Elemente sichern die Anlage, bzw dienen zur manuellen freischaltung aller Elemente, und müssen allen anderen Komponenten vorgeschaltet sein.

Auf der zweiten Seite befinden sich alle Netzteile für die normale 24V DC Spannung. Es sind in Summe drei Netzteile auf dieser Seite die jeweils an L1, L2, L3 und PE angeschlossen gehören. Deswegen wurden die vier Leiter zuerst in Klemmen geführt, dort mit Querverbindern so verbunden, dass man ohne potentiell gefährliche Drahtbrücken alle Netzteile versorgen kann. Von diesen querverbundenen Reihenklemmen wurden auch Abgänge für den Asynchronmotor eingezeichnet und ein Abgang für die Versorgung des ASI-Netzteils. Zwei der Netzteile (Deutronic und SITOP) können 20A auf der 24V-Seite schalten. Diese leistungsstarke Versorgung wird verwendet um die Schrittmotoren zu versorgen. Diese beiden Gleichrichter-Netzteile teilen sich die Schrittmotoren auf. Drei auf die Deutronic und vier auf die SITOP. Von der SITOP bekommen wir zwei 24V Ausgänge. Der Eine wird mit einem von zwei Feed-In-Modulen verbunden. Der Zweite wird direkt mit einem ungesicherten und kleineren Schrittmotor verbunden. Die Deutronic bietet einen Ausgang, dieser wird mit dem zweiten Feed-In-Modul verbunden. Damit war auf dieser Seite die Versorgung von den Schrittmotoren vollendet. Überbleibt noch das Netzteil von der Firma Meanwell. Dieses ist mit 10A leistungsschwächer. Doch die Leistung wird nicht benötigt werden, da dieses Netzteil nur zur Versorgung von den Logikkreisen verwendet wird. Das bedeutet, von den beiden 24V DC Ausgängen dieses Netzteils wurden Verbindungen zur SPS, ET200 und internen sowie externen Not-Aus Schaltern gezeichnet. Auf dieser Seite bedeutet dies Klemmen mit Abgängen.

Was jetzt noch übergeblieben ist sind die 24V-DC-Minuspole der Netzteile. Die Reihenklemmen wurden über Querverbinder miteinander zusammenverbunden und im Anschluss auch mit den PE-Klemmen der 230V Seite verbunden. Die Entscheidung den Steuerstromkreis zu Erden beruht auf der Norm EN60204-1. Demnach hilft die Erdung gegen EMV-Probleme und gewährleistet im Fehlerfall die Abschaltung. [32][33]

Die nächste Seite ist ein Nachtrag, denn anfänglich wurde nicht bedacht, dass der ASI-Kreis eine eigene 24V-DC-ASI-Versorgung braucht. Auf die zweite Seite wurde demnach die ASI-Versorgung gezeichnet sowie der Anschluss an den ET200-ASI-Master. Eingehend sind auf dieser Seite der L1, N und ein Set, letzteres ist ein Eingang beim ASI-Master und wird von der SPS aus gesetzt, und ausgehend ist der ASI-BUS vom ASI-Master.

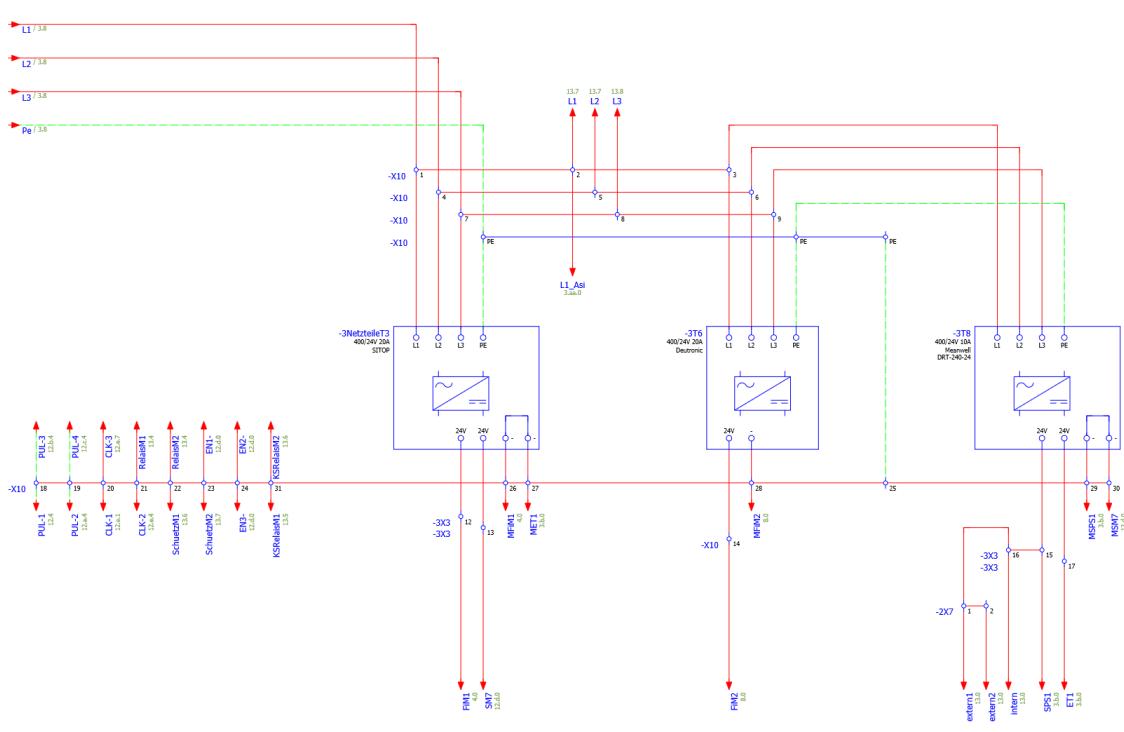


Abbildung 4.8: Schaltplan: Netzteile

Auf der dritten Seite wurde der Versorgungsanschluss von SPS und ET200 gezeichnet. Eingehend sind 24V einmal für die SPS sowie für die ET200 und der jeweilige Minuspol dazu. Auf dieser Seite wurden die Drahtverbindungen gezeichnet, die die SPS, die Ein und Ausgangskarten sowie die beiden PTO-Karten versorgt.

Auf der vierten Seite findet sich das erste Feed-In-Modul von Weidmüller. Hierbei eingehend sind 24V, mit Minuspol, von der SITOP und ausgehend sind die Brückenverbindungen von den DC-Sicherungen. Das wären zwei mal Brücken für 24V, zwei mal Brücken für GND und eine Brücke für den BUS dieser Bugruppe.

Auf den nächsten beiden Seiten finden sich jweils eine 8A-DC-Sicherung. Eingehend sind die bereits genannten Brücken, diese gehen dann auch weiter zur nächsten Sicherung, und ausgehend sind einmal + und - für die stärkeren Schrittmotoren. Wie erwähnt sind das Seite fünf und sechs.

Auf Seite sieben findet sich eine 2A-DC-Sicherung für einen von drei schwächeren Schrittmotor. Diese hat ebenfalls die Brückenverbindung eingehend, allerdings hört die Brückenverbindung mot dieser Sicherung auf. Das bedeutet ein Feed-In-Modul versorgt drei Sicherungen. Abgehend hat diese Sicherung + und - für den passenden Schrittmotor.

Ab der Seite acht wiederholen sich die letzten vier Seiten, mit dem Unterschied dass die gebrückte Versorgungsleitung nicht von der SITOP kommt, sonder von der Deutronic. Auf

der Ersten ist damit wieder das Feed-In-Modul für die zweite Baugruppe. Somit hat man auf den drei folgenden Seiten drei DC-Sicherungen (8A, 8A, 2A), auf jeder Seite jeweils eine und von jeder Sicherung zwei Abgänge für die Schrittmotoren.

Ab Seite 13 sind vier fast identische Schaltpläne. Der Grundaufbau ist immer gleich, man kommt mit der Versorgung von den Sicherungen und geht damit in die Schrittmotorkarte Type CL57C von Nanotec. Von den Treibern wurden dann Verbindungen zu einem Schrittmotor gezogen und zum zugehörigen Geber.

Wenn wir diesen Teil des Schaltplanes von der Motorseite aus betrachten gibt es für den Motor die Anschlüsse A+, A-, B+, B-. Das sind die Anschlüsse für die zwei Pole eines Schrittmotors. Der Geber, welcher mit dem Motor über einen Welle gekoppelt ist, hat die Anschlüsse EB+, EB-, EA+, EA- und für die Versorgung VCC und GND. Der Treiber CL57C hat Ausgänge die genau dieselbe Bezeichnung haben. Die gleichnamigen Anschlüsse wurden miteinander verbunden.

Da der Motor nicht im Schaltschrank sondern am Lagerregal verbaut wird muss vom Treiber zum Motor beziehungsweise Geber ein Kabel verlegt werden. Dies wiederum bedeutet, dass wir im Schaltschrank vom Treiber zuerst auf Reihenklemmen gehen und an diese Reihenklemmen dann die Kabel geklemmt werden. Für den Schrittmotoren wird ein 5adriges Kabel (inklusive PE) verwendet mit der Kabelbezeichnung "ÖLFLEX® CLASSIC FD 810 CY 5G0,75". Der PE Leiter wird an die Erdung des Schaltschranks angeklemmt und dieses Kabel hat, um die elektromagnetische Verträglichkeit zu gewährleisten, eine Schirmung. Die Schirmung muss, um zu funktionieren, an die Schaltschrank-Erdung gelegt werden, weil damit der Schirm eine Senke sein kann für die von den Leitern abgestrahlten elektromagnetischen Wellen. Das Kabel wird auf einer Ankerschiene mit der entsprechenden Zugentlastungsklemme, auf dem Erdungsmodul, nicht nur zugentlastet sondern auch geerdet. Dabei muss dann aber bedacht werden, dass die Klemme Kontakt zum Schirm haben muss, und die Ankerschiene geerdet sein muss. Der Schirm sowie die Erdung dessen wurde im Schaltplan hinzugefügt.

Bezüglich den Motorkabeln wurde ebenfalls die Farbe der Adern bedacht. Während beim "ÖLFLEX® CLASSIC FD 810 CY 5G0,75" die Adern durchgängig nummeriert sind sind es die Adern des Schrittmotors nicht. Diese haben die Farben Schwarz, Grün, Rot und Blau. Da man beim Verdrahten nicht immer wissen kann welche Farbe zu welchem Anschluss führt, wurde auch dies im Schaltplan gezeichnet.

Auch der Geber benötigt ein Kabel. Für diesen wurde auf Patch-Kabel gesetzt. Das achtadrige "FL CAT5 PATCH 10,0" bietet die Möglichkeit RJ45 Stecker zu verwenden. Diese befinden sich jeweils an den Enden und ersparen im Schaltschrank eine Menge Reihenklemmen, da man nur für die Buchse eine Aussparung bedenken muss. Vor allem im Entwicklungsprozess ist das komfortable An und Abstecken ein großer Vorteil aufgrund der Zeitsparnis. Aber auch im späteren Normalbetrieb ist es von Vorteil, bei zum Beispiel einer Umsiedelung des AFSS, die Kabel einfach abstecken zu können. Auch diese Kabel haben einen Schirm der über den RJ45 Stecker geerdet werden kann.

Damit wäre die Kommunikation von Treiber, Motor und Geber gezeichnet. Doch es gehört auch die Kommunikation zur SPS dazu. Der Treiber hat folgende Kontakte: PUL+, PUL-, DIR+, DIR-, ENA+, ENA- und ALM, BRK, COM-. Alle Anschlüsse mit einem - am Ende beschreiben den Minuspol zum jweiligen Gegenstück mit +. Da In unserem Fall alle + Kontakte von der selben SPS kommen und damit alle die selbe Spannung referenzieren können PUL-, DIR-, ENA- und COM- zusammengeschlossen werden und daraufhin zur Sammelschiene der Minuspole geklemmt werden. PUL+ steht für pulse und reguliert die Geschwindigkeit der Drehung, DIR+ steht für direction und reguliert die Drehrichtung, ENA+ steht für enable und Ab Seite 13 sind vier fast identische Schaltpläne. Der Grundaufbau ist immer gleich, man kommt mit der Versorgung von den Sicherungen und geht damit in die Schrittmotorkarte Type CL57C von Nanotec. Von den Treibern wurden dann Verbindungen zu einem Schrittmotor gezogen und zum zugehörigen Geber.

Wenn wir diesen Teil des Schaltplanes von der Motorseite aus betrachten gibt es für den Motor die Anschlüsse A+, A-, B+, B-. Das sind die Anschlüsse für die zwei Pole eines Schrittmotors. Der Geber, welcher mit dem Motor über einen Welle gekoppelt ist, hat die Anschlüsse EB+, EB-, EA+, EA- und für die Versorgung VCC und GND. Der Treiber CL57C hat Ausgänge die genau dieselbe Bezeichnung haben. Die gleichnamigen Anschlüsse wurden miteinander verbunden.

Da der Motor nicht im Schaltschrank sondern am Lagerregal verbaut wird muss vom Treiber zum Motor beziehungsweise Geber ein Kabel verlegt werden. Dies wiederum bedeutet, dass wir im Schaltschrank vom Treiber zuerst auf Reihenklemmen gehen und an diese Reihenklemmen dann die Kabel geklemmt werden. Für den Schrittmotoren wird ein 5adriges Kabel (inklusive PE) verwendet mit der Kabelbezeichnung "ÖLFLEX® CLASSIC FD 810 CY 5G0,75". Der PE Leiter wird an die Erdung des Schaltschranks angeklemmt und dieses Kabel hat, um die elektromagnetische Verträglichkeit zu gewährleisten, eine Schirmung. Die Schirmung muss, um zu funktionieren, an die Schaltschrank-Erdung gelegt werden, weil damit der Schirm eine Senke sein kann für die von den Leitern abgestrahlten elektromagnetischen Wellen. Das Kabel wird auf einer Ankerschiene mit der entsprechenden Zugentlastungsklemme, auf dem Erdungsmodul, nicht nur zugentlastet sondern auch geerdet. Dabei muss dann aber bedacht werden, dass die Klemme Kontakt zum Schirm haben muss, und die Ankerschiene geerdet sein muss. Der Schirm sowie die Erdung dessen wurde im Schaltplan hinzugefügt.

Bezüglich den Motorkabeln wurde ebenfalls die Farbe der Adern bedacht. Während beim "ÖLFLEX® CLASSIC FD 810 CY 5G0,75" die Adern durchgängig nummeriert sind sind es die Adern des Schrittmotors nicht. Diese haben die Farben Schwarz, Grün, Rot und Blau. Da man beim Verdrahten nicht immer wissen kann welche Farbe zu welchem Anschluss führt, wurde auch dies im Schaltplan gezeichnet.[34]

Auch der Geber benötigt ein Kabel. Für diesen wurde auf Patch-Kabel gesetzt. Das achtadrige "FL CAT5 PATCH 10,0" bietet die Möglichkeit RJ45 Stecker zu verwenden. Diese befinden sich jeweils an den Enden und ersparen im Schaltschrank eine Menge Reihenklemmen, da man nur für die Buchse eine Aussparung bedenken muss. Vor allem im Entwicklungspro-

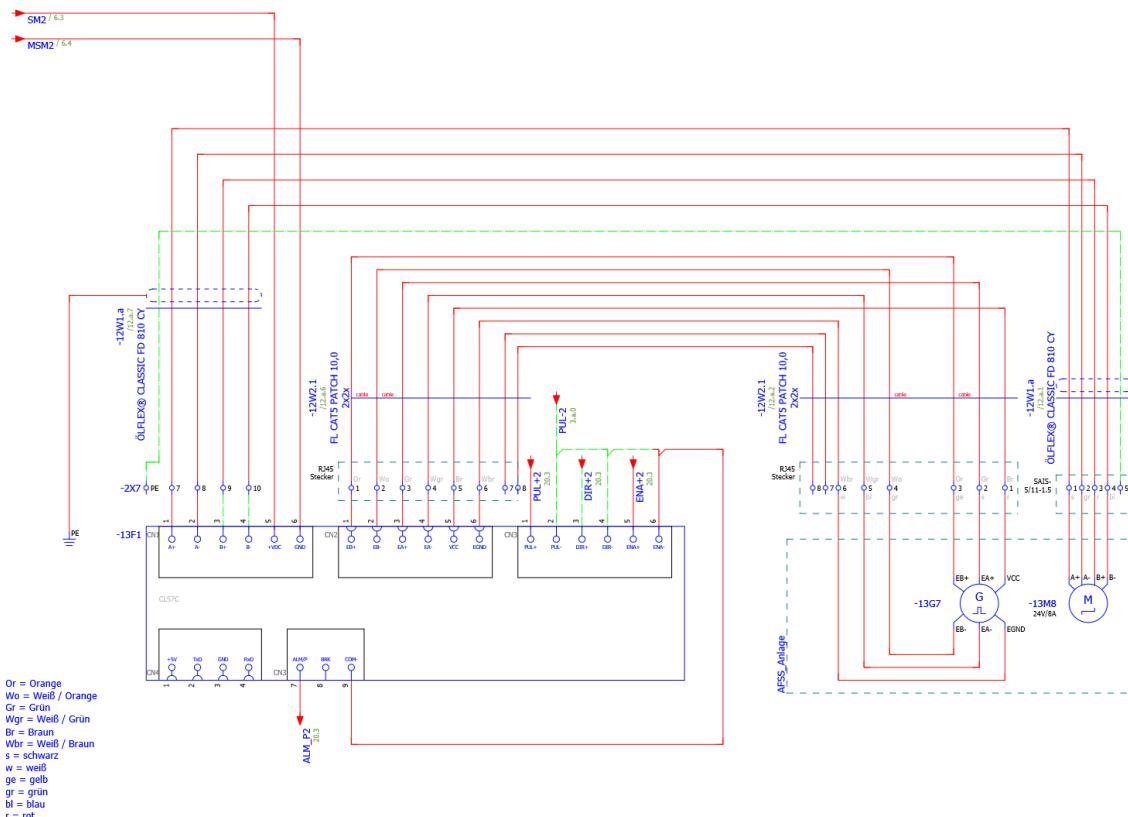


Abbildung 4.9: 13te Seite des Schaltplans, SM ohne Bremse

zess ist das komfortable An und Abstecken ein großer Vorteil aufgrund der Zeitersparnis. Aber auch im späteren Normalbetrieb ist es von Vorteil, bei zum Beispiel einer Umsiedelung des AFSS, die Kabel einfach abstecken zu können. Auch diese Kabel haben einen Schrim der über den RJ45 Stecker geerdet werden kann.

Damit wäre die Kommunikation von Treiber, Motor und Geber gezeichnet. Doch es gehört auch die Kommunikation zur SPS dazu. Der Treiber hat folgende Kontakte: PUL+, PUL-, DIR+, DIR-, ENA+, ENA- und ALM, BRK, COM-. Alle Anschlüsse mit einem - am Ende beschreiben den Minuspol zum jweiligen Gegenstück mit +. Da In unserem Fall alle + Kontakte von der selben SPS kommen und damit alle die selbe Spannung referenzieren können PUL-, DIR-, ENA- und COM- zusammengeschlossen werden und daraufhin zur Sammelschiene der Minuspole geklemmt werden. PUL+ steht für pulse und reguliert die Geschwindigkeit der Drehung, DIR+ steht für direction und reguliert die Drehrichtung, ENA+ steht für enable und entsperrt den Treiber beziehungsweise gibt frei, dass ein Ansteuerung gewünscht ist. BRK steht für break und würde benötigt werden, wenn eine Bremse angedacht wäre, was sie in diesem Fall aber nicht ist. ALM steht für arlarm und ist ein Ausgang der einen erkannten Fehler meldet.

ALM wurde mit der Eingangskarte verbunden und PUL+, DIR+ und ENA+ wurden an die zugehörigen Kontakte bei den PTO Karten der SPS gezeichnet.

Während die ersten beiden SM-Treiber Schrittmotoren ansteuern die keine Bremsfunktion brauchen, da sie den Apparat nur horizontal bewegen und damit bei plötzlichem Ausfall, die

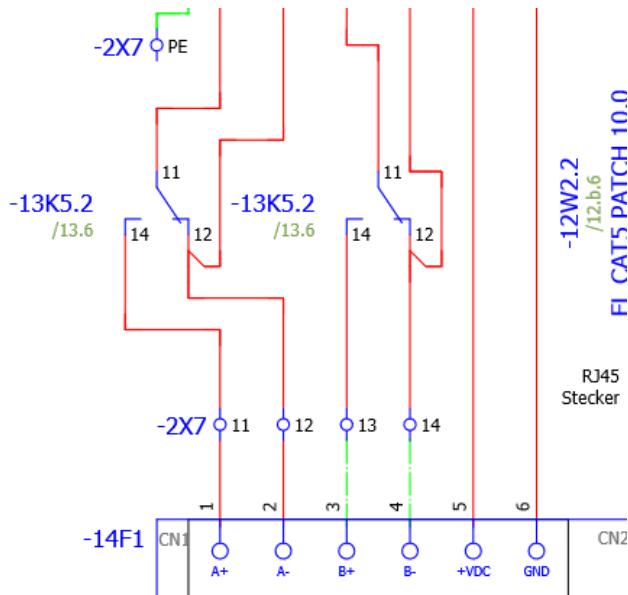


Abbildung 4.10: Zusätzliche Kurzschließung von Spulen bei horizontalen Antrieben

Anlage auf dieser Achse zum Stillstand kommt, brauchen die anderen Zwei eine Bremsfunktion. Die zwei restlichen SM-Treiber steuern Motoren an, die zum Heben des Gabelapparates sind, uns wenn diese auf einmal ausfallen würden, würde die Schwerkraft das Konstrukt zu Boden fallen lassen. Deswegen sollen im Fehlerfall Relaiskontakte die Spulen der Schrittmotoren kurzschließen und damit eine bremsende Wirkung ermöglichen. Dabei funktioniert das Prinzip so, dass bei plötzlicher Spannungsabwesenheit das Relais loslässt und die Spulen kurzgeschlossen sind. Das heißt im spannungsfreien Zustand sind die Spulen kurzgeschlossen siehe Abbildung ??.

Auf den darauf folgenden zwei Seiten befindet sich die Steuerung von den schwächeren Schrittmotoren. Auch diese haben Treiber und Motoren aber keine Geber.

Auch die Geber für die schwächeren Schrittmotoren haben einen enable-Eingang. Dieser ist jedoch dann enabled wenn der Eingang Spannungsfrei ist. Aufgrund dieser Eigenschaft gibt es die Gefahr, dass die Schrittmotoren schneller losfahren als die SPS die Spannung aufgebaut hat. Beispielsweise wenn die Anlage einen Neustart durchführt könnte diese Situation entstehen. Deswegen werden die Eingänge Invertiert. Ein Relaiskontakt, ein Öffner, setzt den EN-Eingang immer auf 24V, erst sobald die SPS das Relais ansteuert kann der Treiber enabled werden. Bei drei Treiber macht dies drei Relais. Auf der 17ten Seite befinden sich somit die Relais und deren Öffnerkontakte. Anfänglich wurden hier fälschlicher Weise Schließer eingeplant, im Verlauf der Arbeit wurde diese Entscheidung hinterfragt und demnach auch ausgebessert. Auf der darauf folgenden Seite finden sich die Treiber und Schrittmotoren.

Zu den Treibern kann man sagen, dass alle Treiber ähnlich aufgebaut sind. Wieder gibt es einen Kontakt zur Geschwindigkeitsregelung (CLK+ und CLK-), einen zur Richtungsbestimmung (CW+ und CW-) und den bereits besprochenen Freigabeeingang (EN+ und EN-).

Weiterführend haben die Treiber noch Anschlüsse für die Versorgungsspannung und A+, A-, B+ und B- für die Schrittmotoren. Die Massekontakte werden wieder alle zusammenge schlossen und weitergeföhrt zur gemeinsamen Masse.

Auch für diese Motoren werden die selben Kabel verwendet wie zuvor. Wieder gilt, dass die Schirmung bei der Zugentlastung geerdet wird und auch die Farben der Adern wurden im Schaltplan eingezeichnet.

Auch hier wurden die Steuerkontakte von den Treibern an die PTO-Karten der SPS ange-

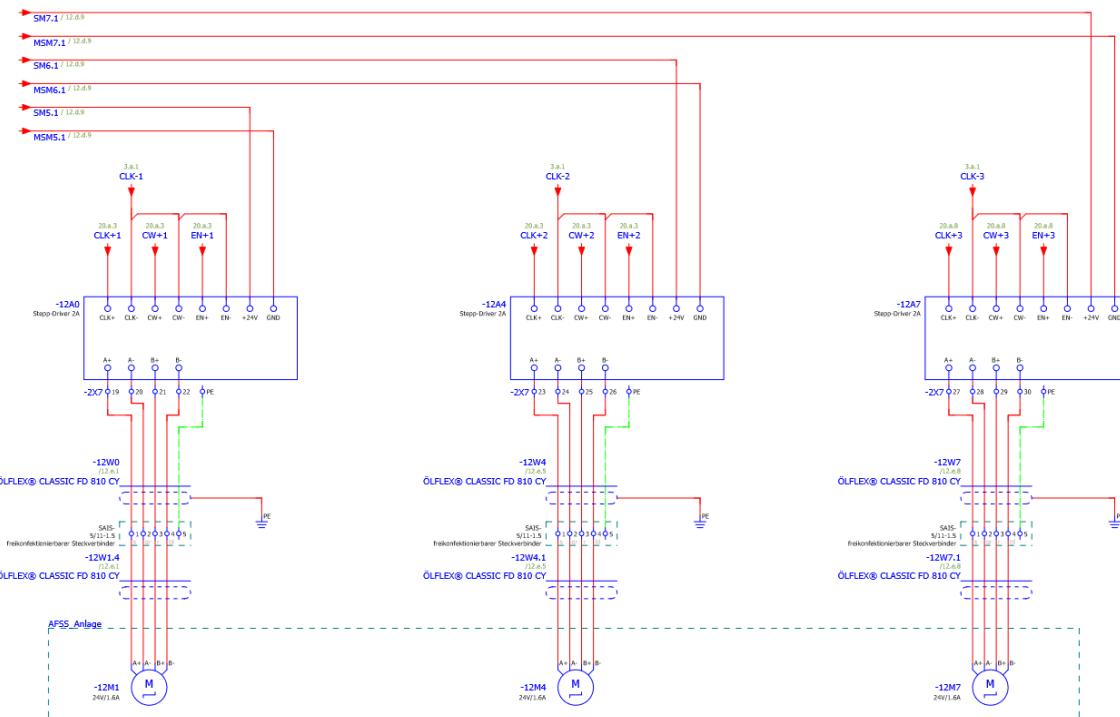


Abbildung 4.11: 40Ncm Schrittmotoren im Schaltplan

schlossen.

Auf der 19ten Seite befinden sich Not-Aus-Schaltungen und die Relais die dem Vertikal antrieb die Bremsfunktion geben. In Summe sind es vier Not-Aus und zusätzlich noch ein Schlüsselschalter zur Freigabe. Für die fünf Sensoren gibt es drei Versorgungsleitungen, je nach räumlicher Position teilen sich die Komponenten eine Leitung, und alle haben jeweils einen Abgang zur SPS-Eingangskarte. Die Relais bekommen ihre Versorgung von der SPS und haben Abgänge zur gemeinsamen Masse. Die Funktion der Relais wurde bereits beschrieben.

Die nächste Seite zeigt die ganze Steuerung des Asynchronmotors, welcher das Förderband antreibt. Hierfür gibt es Schütze die in einer Wendeschützschaltung den Asynchronmotor ansteuern. Es ist keine Drehzahlregelung nötig. Die Schütze blockieren sich gegenseitig und werden angesteuert über zwei Relais. Diese Relais werden direkt von der SPS angesteuert.

Beim Asynchronmotor gibt es noch einen Motorschutzschalter, dessen genauere Auslegung im Verlauf der Diplomarbeit noch genauer beschrieben wird.

Die Seiten 21 und 22 zeigen die PTO-Karten der SPS. PTO steht Pulse-Train-Output und steuert die Treiber an. Ursprünglich wären für diese Aufgaben PWM-Karten von Weidmüller verwendet worden, es ist allerdings bei Testversuchen aufgefallen, dass diese nicht in der Lage sind variable Frequenzen zu erzeugen. Damit musst eine Alternative her. Die Siemens PTO-Karten sind fähig die Frequenzen auch während dem Betrieb zu ändern und damit ermöglichen sie sanftes Anfahren der Schrittmotoren. Eine PTO-Karte kann vier Schrittmotoren steuern.

Diese Karten haben mehrere Ausgänge und Eingänge. Im Fall des AFSS gehen vier mal die PUL+ und DIR+ und ENA+ aus und eingehend sind die vier ALM (Arlarmmeldungen). Das sind die Kontakte von den vier stärkeren Treibern und damit ist die erste PTO-Karte voll ausgenutzt. Mit der Zweiten werden die Drei schwächeren Treiber angesteuert. Diese mal wieder mit den unterschiedlich benannten Kontakten.

Auf der folgenden Seite befindet sich die Eingangskarte der SPS mit den Eingängen. Dabei werden primär die Zustände der Not-Aus-Schalter abgefragt und die des Schlüsselfreigabeschalters.

Auf der Seite 24 befindet sich die Ausgangskarte der Siemens SPS. Mit dieser werden ausschließlich nur Relais angesteuert. Diese sind für die Invertierung der Enable-Eingänge der Treiber zuständig, für das Steuern der Schütze für den Asynchronmotor und zuletzt auch für die Bremsfunktion. Damit ist die Schaltung zur Steuerung aller Schrittmotoren fertig. Auf den folgenden Seiten wurde dann die Schaltung der ASI-Sensoren gezeichnet

Bezüglich dem ASI-Netz ist zu wiederholen, dass der ASI-Master mit den Slaves über einen zweiadrigten ASI-Bus kommuniziert. Für diesen Bus gibt es ein eigenes Kabel, aber da mit dem Material gearbeitet werden muss, was vorhanden ist wird beim AFSS das selbe fünfadrige Ölflex verwendet wie für die Motoren. In diesem ungünstigen Fall werden von den fünf adern am Ende nur zwei verwendet.

Bezüglich den Slaves ist zu sagen, dass ursprünglich ein Slave mit sieben Inputs gedacht. Dieser wäre in der Schule genügend vorhanden gewesen. Allerdings ist bei Testläufen es nicht gelungen eine Kommunikation zwischen diesen Slaves und dem Siemens ASI-Master herzustellen. Deswegen werden anstatt vier Slaves mit sieben Eingängen nun sieben ASI-Slaves von Siemens mit jeweils vier Eingängen verwendet. Der genaue Typ ist 3RK2200-0CE02-0AA2 und es ist möglich mit diesem zu kommunizieren.

Der Bus fährt auf mit Steckverbinder erweiternde Reihenklemmen, von denen dann auf die jeweiligen Slaves. Es sind immer zwei Slaves räumlich beieinander und werden gemeinsam mit Reihenklemmen auf dem AFSS auf Hutschienen montiert. Der siebte Slave ist alleine. Wie die Sensoren mit den ASI-Slaves verbunden sind sieht man auf der Abbildung 4.12 und für eine genaue Erklärung zu den Sensoren siehe Kapitel 5.1.

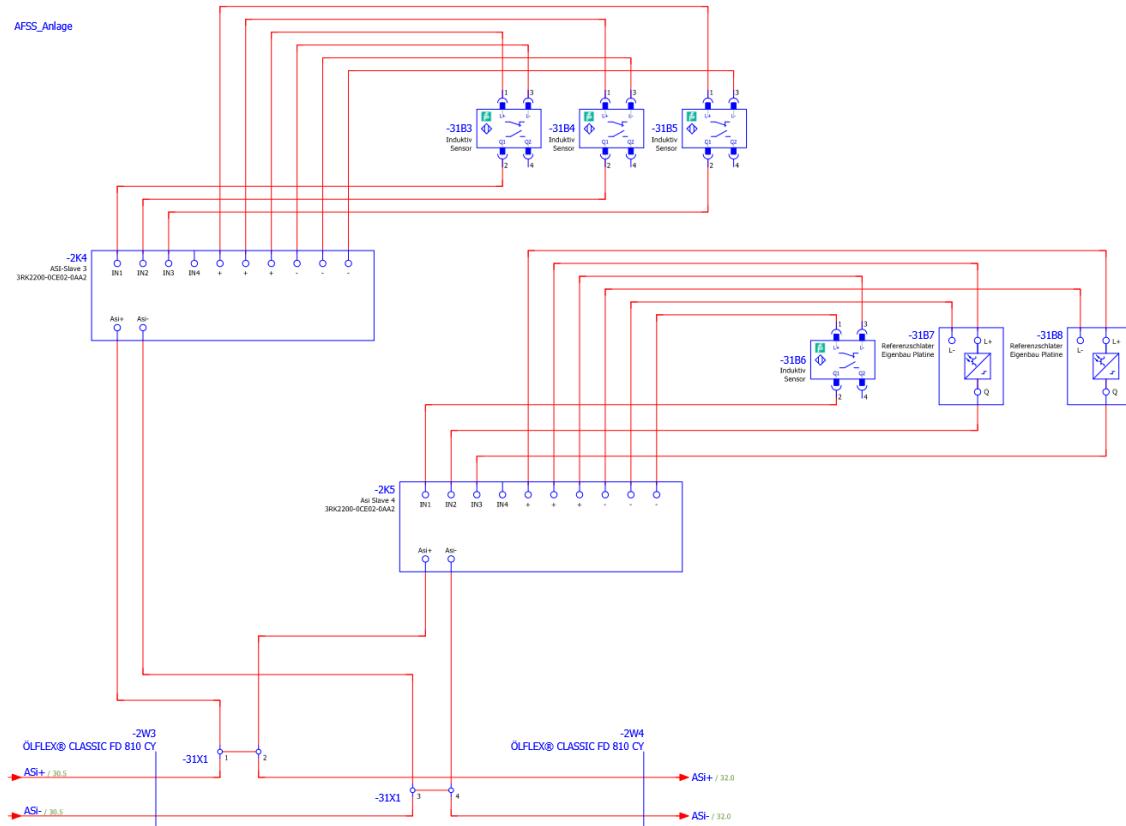


Abbildung 4.12: Verbindung der Sensoren mit den ASI-Slaves

In Summe erstreckt sich der genaue Schaltplan der ASI-Slaves über die letzten vier Seiten.

—Wie wurden Motorschütz ausgelegt— —Wie wurden FI und LS ausgelegt— —ASI-SLaves ursprünglich mit sieben angeschlossen, waren dann nicht kompatibel— —Tabelle—

## 4.2 Die Auslegung von den jeweiligen Komponenten (Tabelle mit allen Komponenten machen?)—Vor die E-Planung setzen um im Folgekapitel nicht konstat erklären zu müssen was welches teil macht

## 4.3 Realisierung

Filou-NC

## 5 Sensorik und Sicherheitstechnik (Elena Widmann)

### 5.1 Sensorik

#### 5.1.1 Aufgabenstellung

#### 5.1.2 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. [35]

Auf jeder der drei Achsen des 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 wurden.

#### 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 einen Öffner- und einen Schließerkontakt [36], wohingegen einer der Endschalter aus zwei Öffnerkontakte besteht [37]. 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 angebracht werden.

### Endtaster

An der yz-Achse werden vier und am Querförderer 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 [38]



(b) Induktiver Sensor [39]



(c) Endtaster

Abbildung 5.1: Endschalter

### 5.1.3 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 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 bzw. dem Gerüst 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.

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

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.[40] Da die Opto Interrupter jedoch über ASi-Bus mit der SPS verbunden werden, welche eine Spannung von 24 V liefert, musste eine eigene Platine entworfen und gelötet werden, um das Bauteil nicht mit einer zu hohen Betriebsspannung zu zerstören. Hierfür wurde die Software Fusion360 verwendet, welche das Designen von Leiterplatten ermöglicht. Hergestellt wurden diese dann durch unsere eigene schulinterne Leiterplattenfertigung. Insgesamt musste die Referenzplatine sieben mal hergestellt werden.

## Schaltungsentwurf

Die Schaltung sollte so konzipiert werden, 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.[41] 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.

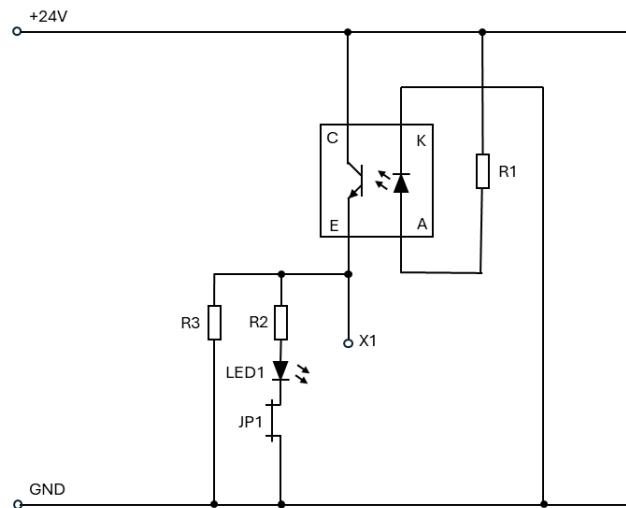


Abbildung 5.2: Schaltplan Referenzplatine

## Platinenentwurf und -herstellung

Um die benötigten Referenzplatten herstellen zu können, muss zuerst ein Leiterplattenplan in Fusion360, ehemals Eagle, erstellt werden. Über den Sharepoint der HTL lässt sich eine Elektronikbibliothek, die alle in der Schule verfügbaren Bauteile beinhaltet, herunterladen. Da der in der Schaltung verwendete Opto Interrupter nicht in der Schule verfügbar ist, sondern extern organisiert werden muss, befindet er sich nicht in dieser Elektronikbibliothek. Daher muss für ihn ein eigenes Symbol sowie ein dazugehöriger Footprint gezeichnet werden.

Zum Entwerfen des Printed Circuit Boards (PCB) muss ein neuer Elektronikentwurf in Fusion erstellt werden. Hier muss zuerst der zugehörige Schaltplan gezeichnet werden. Wichtig ist, dass bei der dreipoligen Schraubklemme das Bauteil 3282837-3 (J1) verwendet wird, da sonst die Abstände zwischen den Lötpads zu klein und diese zu nah bei einander sind. Damit der Jumper (JP2) nicht verloren geht, wenn die Verbindung zwischen Ground und LED aufgehoben werden soll, wird ein dreipoliger Pinheader verwendet, um den Jumper für den gegebenen Zeitraum einfach umstecken zu können.

Nach Fertigstellung des Schaltplans kann in Fusion ein passendes Leiterplattendokument erstellt werden, welches die Bauteile und die zugehörigen Verbindungen direkt übernimmt. Für den fertigen Leiterplattenplan des Referenztasters siehe Abb.5.3. Da beim verwendeten Opto Interrupter Löcher zur Montierung vorhanden sind, müssen auf der Platine selbst keine zusätzlichen Bohrungen eingeplant werden. Beim der Anordnung der Bauteile auf der Platine ist zu beachten, dass sich die Löcher am Opto Interrupter, am schmäleren rand der Platine befinden. Um eine leichtere Verkabelung zu ermöglichen, sollte auch die Schraubklemme am Rand der Platine platziert werden.

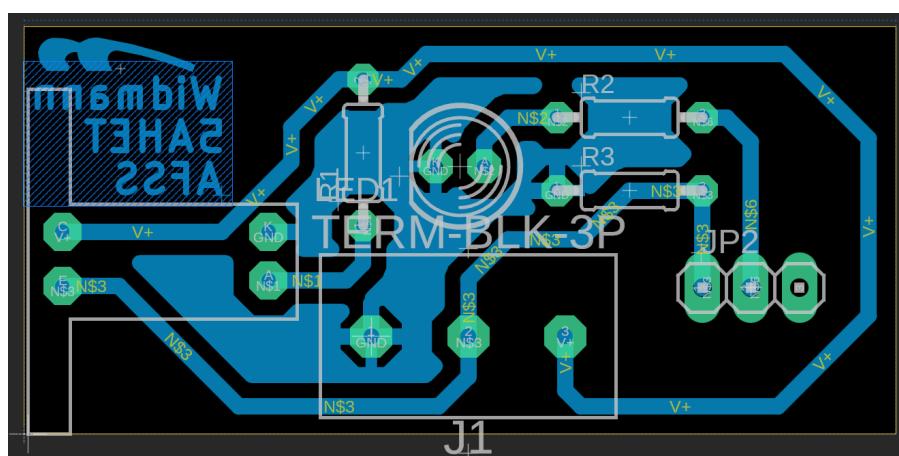


Abbildung 5.3: Leiterplattenplan Referenzplatine

Wenn die Bauteile alle platziert und verbunden worden sind, sowie ein Polygon über die gesamte Platine gezogen worden ist, muss diese noch auf Fehler geprüft werden. Auch hier ist eine bereits fertige Datei, welche die benötigten Design Rules für Fusion360 beinhaltet, auf der Schulwebsite zu finden. Wichtig ist, damit die Platine zur Produktion in der Leiterplattenfertigung der HTL eingereicht werden kann, muss diese den vorgegebenen Anforderungen

entsprechen. Dazu gehört, dass sich das HTL-Logo auf der Platine befindet und die Texteinstellungen Font: Vector, Ratio: 16 % und Size: min 70 mil entsprechen. Auch die Breite der Kupferbahnen darf nicht zu klein sein (hier: 32 mil).

Nach Einreichung des Fertigungsauftrags wird die Platine von Schülerinnen und Schülern der HTL gefertigt. Der Prozess startet mit dem Reinigen des Basismaterials, um es daraufhin mit dem Negativtrockenresist (Trockenfilm) zu laminieren. In den nächsten Schritten werden die Layout-Informationen mit einem Belichter auf das Laminat übertragen und das unbelichtete Laminat mit einer Natrium-Carbonat Lösung von der Platine entfernt. Daraufhin werden durch Ätzen mit einer Eisen-III-Chlorid-Lösung strukturierte Kupferflächen freige stellt. Als Nächstes werden die Löcher in die Kupferpads gebohrt und die Platine auf ihre korrekte Größe zugeschnitten. Durch das Legen der Leiterplatte in eine Entschichtlösung aus 5 %igem Kaliumcarbonat mit Wasser werden Ätzreste von der Platine entfernt. Zu guter Letzt wird die Platine mit einem Versiegelungslack versiegelt, um sie vor Umweltein flüssen und Korrosion zu schützen.

### Platinentestung und Messung

Um die korrekte Funktionsweise der Referenzplatine sicher zu stellen, muss diese nach dem Löten getestet werden. Zum Betrachten der fertigen Platine siehe Abb.**PLATINENENBILD!!!**. An der Klemme X1 wurden Spannungswerte zwischen 19,7 V bis 21,3 V gemessen. Obwohl die Spannungen unter den gewünschten 24 V liegen können die Platinen problemlos verwendet werden, da die verwendeten AS-i-Slaves einen minimalen HIGH-Eingangsschaltpegel von 10 V besitzen.[42]

#### 5.1.4 Lichttaster

Zum Überprüfen, ob während des Ein- oder Auslagerns eine Box auf der richtigen Position angekommen ist, werden Lichttaster verwendet. Insgesamt werden zwei am AFSS montiert, davon einer an der z-Achse und einer am Querförderer. Es handelt sich bei den hierbei verwendeten Reflexionslichttaster um Sensoren der Firma SICK vom Typ W4 WTBxx (siehe Abb.5.4a).

Ursprünglich hätten diese Aufgaben von Lichtschranken übernommen werden sollen, da diese bereits vor Ort verfügbar waren. Lichtschranken funktionieren, indem sie einen Rot lichtstrahl auf einen Reflektor werfen, dessen Reflektion daraufhin wieder vom Sensor erkannt wird. Im Lichtschranken befinden sich Schaltkontakte, die daraufhin geöffnet oder geschlossen werden. Wird nun jedoch der Lichtstrahl unterbrochen, fallen die Schaltkontakte in ihren Anfangszustand zurück. Die Lichtschranken haben jedoch den Nachteil, dass sie immer ein zusätzliches reflektierendes Gegenstück benötigen. Die stattdessen verwendeten Lichttaster mit Hintergrundausblendung funktionieren nach dem selben Prinzip, jedoch bieten sie den Vorteil, dass sie auch die Distanz des Objektes zum Sensor erfassen. Somit können Objekte unabhängig von ihrer Farbe und Oberfläche erkannt werden, soweit sie sich innerhalb des einstellbaren Tastbereiches befinden.[43] Der empfohlene Schaltabstand des

Lichttasters liegt zwischen 40 mm bis 140 mm, wodurch er sich für die Aufgaben im Projekt ideal eignet. Über die Teach-in Funktion des Sensors lässt dieser sich ganz einfach durch drücken der Teach-Taste auf den richtigen Schaltabstand einstellen.[44]

Während das SPS-Programm von der CPU durchlaufen wird, überprüft dieses immer an den jeweiligen Zwischenschritten, ob es ein Signal des zuständigen Lichttasters erhalten hat, bevor das Programm fortgesetzt wird. Somit wird sichergestellt, dass die Boxen im Querförderer richtig positioniert worden sind und von der z-Achse richtig gegriffen werden können. Ansonsten könnte passieren, dass eine von der Gabel angehobene Box hinunterfällt, da diese nicht richtig positioniert war.



(a) Lichttaster W4 WTBxx[45]



(b) Barcodescanner CLV61x-2Port[46]

Abbildung 5.4: SICK Sensoren

### 5.1.5 Barcode-Scanner

Durch anbringen eines sich nicht wiederholenden Barcodes auf jeder Box wird eine gute Möglichkeit geschaffen, in der Software den Behälter und die sich darin befindenden Bauteile einander zuzuordnen. Unter der Voraussetzung, dass die Benutzerinnen und Benutzer vor jeder Wiedereinlagerung ihrer Box zuerst dessen Barcode einscannen, wird die Wahrscheinlichkeit, dass der Lagerplatz einer Box falsch abgespeichert wird, verkleinert. Somit sinkt die Chance, dass bei Bestellen eines Bauteils die falsche Komponente geliefert wird. Das erfassen des Barcodes findet mittels eines Barcodescanners der Firma SICK statt (siehe Abb.5.4b), welcher bei der Kommissionierstation untergebracht ist, um eine einfache Bedienung gewährleisten zu können.

Der uns zur Verfügung gestellte Barcodescanner (CLV61x-2Port) ist in der Lage alle gängigen Codearten einzulesen.[47] Barcodes wurden so entwickelt, dass bereits beim Einlesen erkannt wird, wo dieser anfängt und aufhört, damit beim einscannen nicht auf die richtige Ausrichtung geachtet werden muss.

### Einbindung ins TIA-Portal [48]

Damit der eingelesene Barcode an den Webserver weitergegeben werden kann, muss dieser im Siemens TIA-Portal abgespeichert werden. Mit der CPU verbunden wird der Scanner über PROFINET. Hierbei handelt es sich um einen auf Industrial Ethernet basierenden

Kommunikationsstandard und eine Weiterentwicklung des PROFIBUS Vorgängers. Für eine ausführlichere Erklärung der verschiedenen Feldbussysteme siehe Kapitel 5.2.1. Über ihn lässt sich die gescannte Nummer ganz einfach an die SPS übermitteln.

Im TIA-Portal Projekt muss, um eine Verbindung zum Barcodescanner herstellen zu können, die zugehörige Gerätebeschreibungsdatei (GSD-Datei) installiert werden. Dies funktioniert über das Gerätebeschreibungsdateien verwalten Fenster im TIA-Portal. Nach der Installation ist das Gerät im Hardwarekatalog unter „Weitere Feldgeräte“ zu finden. Die benötigte Datei wird auf der Internetseite des Herstellers als Download zur Verfügung gestellt.

### Auslesen des Barcodes

Wenn eine Verbindung zum Gerät hergestellt wurde, müssen die Parametermodule eingefügt und richtig eingestellt werden. Diese werden über den Hardware-Katalog ausgewählt und eingefügt, für eine Übersicht der eingefügten Module siehe Abb.5.5. Wichtig ist, bei den Baugruppenparametereinstellungen von „47\_Communication Mode\_1“ „No Handshake“ auszuwählen und bei „99\_End Remote Config“ „Don't save parameters perm.“ einzustellen. Um die verwendeten Art von Barcodes einscannen zu können muss darauf geachtet werden, dass er unter „22\_UPC EAN GTIN\_1“ ausgewählt und somit eingeschaltet ist.

Geräteübersicht						
	Modul	Baugr...	Steck...	E-Adresse	A-Adres...	
	CLV61x-2Port	0	0			
	Interface	0	0 X1			
	Ctrl Bits in_1	0	1	126...127		
	Ctrl Bits out_1	0	2		128...129	
	32 Byte Input_1	0	3	128...159		
	32 Byte Output_1	0	4		130...161	
	01_Start Remote Config >>...	0	5			
	11_Reading Config_1	0	6			
	22_UPC EAN GTIN_1	0	7			
	47_Communication Mode_1	0	8			
	99_End Remote Config <<...	0	9			
			0	10		
			0	11		
			0	12		

Abbildung 5.5: Barcodescanner Geräteübersicht im TIA-Portal

Die vom Barcodescanner belegten Ein- und Ausgangsadressen sind in der Geräteübersicht ersichtlich (siehe Abb.5.5). Ein TriggerBit wird verwendet um das Einlesen eines Barcodes zu starten, dabei handelt es sich um das zweite Bit des „Ctrl Bits out\_1“ Moduls (hier: Q129.0). Bei steigender Flanke des ToggleBits wird der Laser eingeschaltet, erst bei der fallenden Flanke wird daraufhin der Code eingelesen.

Die eingelesenen Daten liegen ab dem ersten Input Byte (hier: ab IB128.0). Auf dem ersten Input Byte wird am vierten Bit ein ToggleBit mitgeführt. Am zweiten Byte(IB129.0) befindet sich ein Zähler, der mitzählt, wie viele Codes bereits eingelesen wurden. Das vierte Input

Byte (IB131.0) gibt die Länge des eingelesenen Strings an. Ab dem sechsten Byte (IB133.0) stehen die eigentlichen Daten des Barcodes.

Um die Daten des Barcodes überprüfen zu können eignet sich eine wie in Abb.5.6 abgebildete Beobachtungstabelle. Wichtig ist, dass beim Einfügen der Ein- und Ausgangsadressen das richtige Anzeigeformat ausgewählt wird. Um eine Überprüfung durchzuführen lässt sich das ToggleBit zum Starten es Einlesevorgangs über Eingabe von TRUE und FALSE setzen, und daraufhin ein Code scannen. Die einzelnen Input Bytes werden daraufhin in der Tabelle angezeigt.



i	Name	Adresse	Anzeigeformat	Beobachtungswert	Steuerwert	
1		%Q129.0	BOOL		TRUE	<input checked="" type="checkbox"/> 
2		%IB128	Bin			<input type="checkbox"/>
3		%IB129	DEZ			<input type="checkbox"/>
4		%IB130	DEZ			<input type="checkbox"/>
5		%IB131	DEZ			<input type="checkbox"/>
6		%IB132	DEZ			<input type="checkbox"/>
7		%IB133	Zeichen			<input type="checkbox"/>
8		%IB134	Zeichen			<input type="checkbox"/>
9		%IB135	Zeichen			<input type="checkbox"/>
10		%IB136	Zeichen			<input type="checkbox"/>
11		%IB137	Zeichen			<input type="checkbox"/>
12		%IB138	Zeichen			<input type="checkbox"/>
13		%IB139	Zeichen			<input type="checkbox"/>

Abbildung 5.6: Barcodescanner Beobachtungstabelle im TIA-Portal

## 5.2 AS-Interface

### 5.2.1 Feldbusssysteme [49]

#### Allgemeines

Ein Feldbus ist ein Bussystem, das Feldgeräte (Sensoren) und Stellglieder (Aktoren) mit einem Automatisierungsgerät, beispielsweise einer SPS, verbindet. Über den Feldbus findet die Kommunikation statt. Wenn die Nachrichten mehrerer Kommunikationsteilnehmer über die selbe Leitung gesendet werden, muss festgelegt sein, wer (Kennung), was (Messwert oder Befehl), wann (Initiative) über den Bus sendet. Es existiert bereits eine Vielzahl an unterschiedlichen Bussystemen, zu denen in der Automatisierungstechnik gebräuchlichsten gehören PROFIBUS, PROFINET, Industrial Ethernet, Modbus, AS-Interface, CANopen und EtherCAT. Da die verschiedenen Bussysteme unterschiedliche Datentransferzeiten und Fehlererkennungsmöglichkeiten haben, werden oft in einem Projekt mehrere Feldbusse auf den verschiedenen Automatisierungsebenen verwendet. In Abb.5.7 ist ein solches Projekt mit unterschiedlichen Bussystemen ersichtlich.

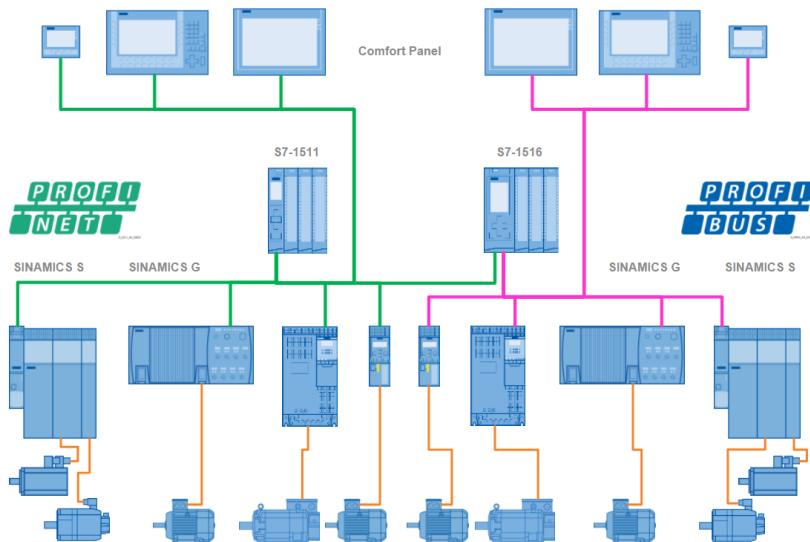


Abbildung 5.7: Topologie verschiedener Feldbussysteme [50]

## PROFIBUS [51]

Bei PROFIBUS (Process Field Bus) handelt es sich um ein der standardisiertes Bussystem das vor allem in der Automatisierungstechnik zu Hause ist. Es ist ein Multi-Master-System, das bedeutet, dass die Kommunikationsabläufe von mehreren Mastern gesteuert werden können. Die am häufigsten verwendete Schnittstelle in einem PROFIBUS-System ist RS485. Zur Datenübertragung wird meist ein 9-poliger Sub-D-Stecker verwendet. Die Anzahl der Teilnehmer ist je PROFIBUS SUB-Netz auf 32 beschränkt, somit ergibt sich eine maximale Anzahl von 128 Teilnehmern. PROFIBUS bietet den Vorteil, dass es sehr einfach in gängige Automatisierungssysteme integriert werden kann. Die Kabelfarbe von PROFIBUS ist violett, was auch beim Verbinden im TIA-Portal ersichtlich ist.

## PROFINET [52]

Wie bereits in Kapitel 5.1.5 erwähnt wurde handelt es sich bei PROFINET (Process Field Network) um eine Weiterentwicklung von PROFIBUS. Es basiert auf Industrial Ethernet und wurde von Siemens und den Mitgliedsfirmen der PROFIBUS-Nutzerorganisation entwickelt. PROFINET-Geräte werden ausschließlich über Switches miteinander verbunden, wobei viele Geräte, sowie eine SPS, bereits einen Switch mit mehreren Anschlüssen integriert haben. Der Vorteil von PROFINET ist, dass es das bekannte 7-schichtige OSI-Modell auf ein 4-schichtiges TCP/IP-Modell reduziert. Der Unterschied zum älteren PROFIBUS liegt vor allem in der Funktion große Datenmengen übertragen zu können und der Echtzeitfähigkeit von PROFINET. Außerdem ist die Anzahl der Teilnehmer in einem PROFINET-System praktisch unbegrenzt, da jeder Teilnehmer eine eigene IP-Adresse besitzt. Bei der Projektierung im TIA-Portal müssen die Kommunikationspartner einem gemeinsamen PROFINET zugewiesen werden und zusätzlich muss jedem Gerät ein PROFINET-Name zugewiesen werden,

nur die IP-Adresse des Geräts genügt nicht. Die Kabelfarbe ist Grün, was sich auch im TIA-Portal widerspiegelt.

## ASi

### Auswahl des Feldbussystems

Da es sich bei PROFINET um ein von Siemens mitentwickeltes Feldbussystem handelt und dieses einfach in Siemens-Steuerungssystemen eingesetzt werden kann, bietet es eine ideale Möglichkeit, die Siemens SPS und den Siemens ASi-Master miteinander zu verbinden. Auch am Barcodescanner befindet sich eine PROFINET-Schnittstelle, womit dieser ganz einfach an die SPS angeschlossen werden kann.

Da die Sensoren jedoch nicht über PROFINET-Schnittstellen verfügen, mussten sie über einen anderen Weg in das PROFINET-System oder ein anderes Feldbussystem integriert werden. Da im Werkstättenunterricht AS-Interface zur Kommunikation zwischen SPS und Sensoren verwendet wird, fiel die erste Wahl auch auf ASi zur Einbindung der Sensorik.

### Die Weidmüller alternative

#### 5.2.2 Programmierung im TIA-Portal

#### 5.2.3 Verkabelung

unterschiedliche Varianten erklären und abwiegen

### 5.3 Sicherheitstechnik

#### 5.3.1 Aufgabenstellung

Die Sicherheitstechnik soll so ausgelegt sein, dass beim Normalbetrieb keine Verletzungsgefahr für die Benutzerinnen und Benutzer besteht. Außerdem soll sie dafür sorgen, dass das System während des Betriebs sich selbst keinen Schaden zufügen kann. Zusätzlich soll sie Fehler, die durch menschliche Bedienung beim Einlagern entstehen, minimieren. Somit kann die korrekte Funktion des AFSS über mehrere Jahre hinweg sicher gestellt werden.

#### 5.3.2 Personenschutz

Der Personenschutz hat die oberste Priorität. Er sorgt dafür, dass bei ordinärem Gebrauch die Benutzerinnen und Benutzer vor Verletzungen geschützt sind. Weiters soll er verhindern, dass Personen in elektrische Stromkreise geraten, oder mit unter Spannung stehenden Betriebsmitteln in Berührung kommen. Bei Wechselstrom mit 50 Hz liegt der in den Vorschriften festgelegte Grenzwert der Spannung, ab dem ein lebensgefährlicher Körperstrom fließen kann, bereits bei 50 V. Bei Gleichstrom ist dieser Grenzwert bei 120 V festgelegt.[53] Durch

den großen Strom wird die Signalleitung der Nerven und Muskeln gestört, und es kann zu Herzklammerflimmern sowie Atemproblemen kommen.[54]

Für den Fall, dass eine Person trotz getroffener Sicherheitsvorkehrungen in den Stromkreis gerät, werden mehrere leicht zugängliche Not-Stop-Schalter am System angebracht, um schnell den Strom abschalten zu können. Als außenstehende Person ist es besonders wichtig, beim Entfernen der bereits in den Stromkreis geratenen Person, sich nicht selbst in Gefahr zu bringen oder auch in den Stromkreis zu kommen.

Als Abgrenzung zwischen Benutzerinnen und Benutzer und dem Schaltschrank ist dieser von außen versiegelt und lässt sich nur mit einem zugehörigen Schlüssel öffnen. Somit wird ein allseitigen Berührungsschutz sichergestellt und zusätzlich wird verhindert, dass Personen ohne Befugnis im Schaltschrank Änderungen durchführen können, und sich dadurch verletzen. Im Schaltschrank befindet sich ein Fehlerstrom-Schutzschalter (FI), welcher bei 30 mA auslöst, und somit zum Personenschutz beiträgt.

### 5.3.3 Schutz des Systems

Damit auch umgekehrt das AFSS vor Fehlern durch Nutzerinnen und Nutzer geschützt ist, müssen auch hierfür eigene Sicherheitsvorkehrungen getroffen werden. Nur so kann für eine langjährige fehlerfreie Funktion des Systems gesorgt werden. Da bereits für den Aufbau ein Raum genutzt wurde, der am Ende des Schuljahres wieder aufgeräumter verlassen werden muss und es durchaus möglich ist, dass das Lager und der Schaltschrank ihren Standort auch in Zukunft wieder wechseln müssen, wurden diese mit Rollen versehen. Durch anbringen der Rollen lässt sich das System problemlos durch das Schulgebäude transportieren ohne, dass es zu einem Schaden kommt. Bereits bei der Planung des Lagers wurde mit-einbezogen, dass das Lager die Höhe und Tiefe des Lifts im Werkstattengebäude nicht überschreiten darf, um den Transport in verschiedene Stockwerke zu ermöglichen. Auch die Höhen und Breiten der Türen wurden schon bei der Konstruktion berücksichtigt.

Das fertige Lagersystem soll planmäßig in der Factory der HTL untergebracht werden. Bei der Raumplanung stellte sich die Frage, ob das Bauteillager oder das Förderband an der Wandseite positioniert wird. Die Entscheidung fiel auf ersteres, um das System weiter Absichern zu können, für den dazugehörigen Raumplan siehe Abb.???. Durch die Positionierung des Bauteillagers an der Wand wird es Schülerinnen und Schüler erschwert, selbst Bauteile auslagern zu können. Beim eigenhändigen Auslagern könnten leicht Boxen vertauscht werden, wodurch es bei Bestellung zu falschen Auslieferungen und Verwechslung von Bauteile kommen kann. Der manuelle Zugriff zum Lager wird durch das Förderband, welches wie eine Absperrung davor steht, verkompliziert. **RAUMPLAN!!** Für den Fall, dass ein händischer Zugriff zu den Bauteilen benötigt wird z.B. bei einem Stromausfall, lässt sich das Lager durch die darauf befestigten Rollen einfach für den benötigten Zeitraum an einen leichter zugänglichen Ort verschieben.

Dem AFSS kann jedoch nicht nur durch den Menschen beschädigt werden, sondern auch sich selbst Schaden zufügen. Ziel der Sicherheitstechnik ist es auch in diesem Falle den verursachten Schaden zu minimieren und wenn möglich, sogar ganz zu verhindern. Einen

weiteren wichtigen Teil zur Sicherheitstechnik trägt die am Lagersystem befestigte Sensorik bei. Damit die x-Schlitten oder das yz-Shuttle nicht in das Gerüst fahren, und dieses dadurch beschädigen, befinden sich Endschalter auf jeder Achse. Wenn diese ausgelöst werden, wird das Technologieobjekt gesperrt und die Motoren werden sofort gestoppt. Außerdem wird durch das Referenzieren der Motoren bei jedem Neustart des Systems verhindert, dass sich die Koordinaten der Boxen verändern. Ansonsten könnte es passieren, dass Boxen, beim Versuch diese auszulagern, nicht richtig von der Gabel angehoben werden und auf den Boden fallen. Zusätzlich überprüfen die Lichttaster, ob die Box während des Ein- und Auslagerungsprozesses an den Zwischenpositionen angekommen ist.

Nicht nur das Lager selbst, sondern auch die Antriebe und die sich im Schaltschrank befindenden Betriebsmittel müssen abgesichert werden. Falls ein zu hoher Strom fließt, zum Beispiel bei einem Kurzschluss, wird der Stromkreis durch einen Leitungsschutzschatz unterbrochen. Speziell für den Asynchronmotor des Fließbands wurde ein eigener Motorschütz eingebaut, um diesen vor Überstrom abzusichern. Weiters befinden sich drei Relais im Schaltschrank, die an die Enable-Kontakte der Motortreiberkarten angeschlossen werden. Es ist besonders wichtig, das AFSS auch für den Fall eines Stromausfalls abzusichern. Gefährdet wäre dadurch in erster Linie das yz-Shuttle. Wenn sich dieses irgendwo zwischen oberer und unterer x-Achse befindet, besteht die Gefahr, dass es durch die Schwerkraft nach unten fällt. Durch die dabei entstehenden Kräfte könnte nicht nur das Shuttle selbst, sondern das ganze Gerüst des Lagers beschädigt werden. Außerdem wird durch das drehen der Motoren in dessen Spulen ein Strom induziert, welcher in weiterer Folge Schäden an anderen Betriebsmitteln anrichten kann. Da die verwendeten Schrittmotoren nicht mit eigenen Bremsen ausgestattet sind, musste ein eigenes Sicherheitskonzept entwickelt werden. Vor den Synchronmotoren der y-Achse werden Relais mit Wechslerkontakte eingebaut. Durch kurzschießen der Spulen des Motors wird verhindert, dass das Shuttle Richtung Boden fällt. Es wird dadurch abgebremst und bewegt sich langsam nach unten. Für eine ausführlichere Erklärung siehe Kapitel 4.1.4.

## 5.4 Fazit

## 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.

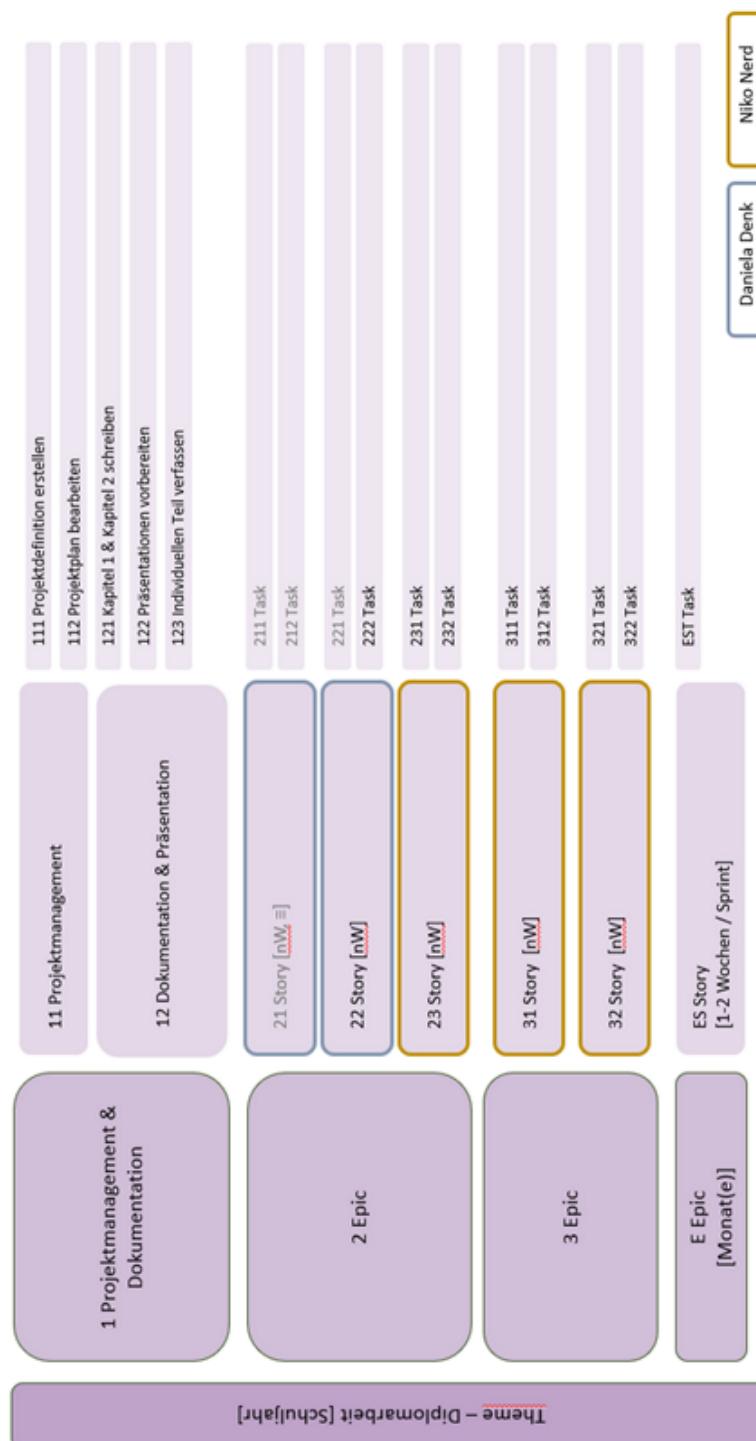


Abbildung 6.1: Scrum Projektplan mit Tasks

### 6.4.3 Terminplanung

#### 6.4.4 Arbeitspakete

##### Maschinenbau (Simbürger)

- Konzeptionierung des Gesamtsystem
- CAD - Planung
- Komponentenfertigung
- Aufbau

##### Softwareentwicklung (Simbürger)

- Benutzeroberfläche
- Warehouse Management System
- Datenbanken

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

**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

Erledigung von, Termin

Datum 18.9.2024      gez. Benedikt Simbürger

Abbildung 6.3: Besprechungsprotokoll 10.12.2024



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

- X-Achsen fortschritt bis Abgabetermin
- SPS Programmierungsstatus
- Projektweiterführungsaspekte
- Status der Verkabelung

**Erledigung von, Termin**

Datum 11.3.2025      gez. Benedikt Simbürger

Abbildung 6.6: Besprechungsprotokoll x.x.xxxx

## 6.8 Arbeitsnachweis

Datum	Tätigkeit	Stunden
4.10.2023	Besprechung des weiteren vorgehens mit WB	0.5
9.10.2023	Gruppenbesprechung für das weitere Vorgehen	1.0
9.11.2023	Reconstruction Siemens Twin Towers	3.5
14.11.2023	Reconstruction Siemens Twin Towers	2.0
23.11.2023	Reconstruction Siemens Twin Towers Lasern	2.0
29.11.2023	RSTT Ausfahrer + Schlitten Fertigstellung	4.0
19.12.2023	CAD Auf/Ab-fahrer	3.0
20.12.2023	CAD RSTT, AutCad vorbereitung	2.0
10.1.2024	Testversuch ET200 SPS Stepdrive und Meeting Knapp	4.0
11.1.2024	Achse mechanisch fertig, Schlitten vertikal	4.0
12.1.2024	CAD Schuttel, Tag der offenen Tür vorbereitung	2.0
16.1.2024	Motoren ansteuern	1.0
20.1.2024	Umlenkungen- und Aufhängungskonstruktion	4.0
29.1.2024	Diagramm Datenaustausch Anfertigung	2.0
1.2.2024	Besprechung WB	4.0
3.2.2024	Website Backend/Frontend Prototyp	9.0
4.2.2024	WS Frontend	3.0
5.2.2024	KWF Antrag schreiben und WS Datenbankmanagement	3.0
7.2.2024	OPC UA Client testung	4.5
15.2.2024	WS Suche usw, Organisation, Maschinenbaubesprechung	5.5
20.2.2024	Python / OPC UA Client testen	1.0
23.2.2024	WS Warenkorb, restructuring	9.0
24.2.2024	WS Warenkorb fertig, OPC anfang und Pflichtenheft Erstversion	6.0
27.2.2024	http-Kommunikation testen	4.0
28.2.2024	Lasten/Pflichtenheft erstellen	3.0
4.3.2024	http-Kommunikation testen	3.5
10.3.2024	CAD X-Achse	7.0
13.3.2024	Absprache mit WB bez. Pflichtenheft	1.0
14.3.2024	http-Kommunikation und CAD	3.0
1.4.2024	Datenbanken und Visualisierung	5.0
2.4.2024	CAD Lagerregal	2.0
17.4.2024	CAD Gabel und Software	2.0
23.4.2024	STT-Fortsetzung / Software einföhrung	2.5
12.4.2024	Datenbanken und Visu	5.0

Datum	Tätigkeit	Stunden
21.4.2024	Datenbanken und Visu	5.0
22.4.2024	TF-IDF Recherche	3.0
23.4.2024	TF-IDF Implementierung	2.0
28.4.2024	Areas und Locations Implementierung	6.0
19.4.2024	Order Algorithmus konzeptionieren	3.0
22.5.2024	Order Algorithmus Implementierung	3.0
2.6.2024	Order Api Programmierung	5.0
3.6.2024	Api Implementierung und Visu	3.0
4.6.2024	STT-Fortsetzung und CAD	2.0
6.6.2024	SPS/Server Communitaion und Z-Prototyp CAD	5.0
8.6.2024	SPS Comm und Simulation implement	4.0
9.6.2024	System Controller	4.0
14.6.2024	Z-Prototyp Bauteile Vorbereitung	1.0
18.6.2024	Return, Cart programmieren	4.0
19.6.2024	Docker (f me)	3.0
21.6.2024	Z-PT, Schaltschrank, SPS-Com	4.5
16.7.2024	Recherche, Referenz-Elektronik	1.0
17.7.2024	Ref-Elektronik	2.0
19.7.2024	Designe/CAD Rollen u. Spannen y	6.0
22.7.2024	Design X-Spannelement	2.0
25.7.2024	Design X-Spannelement und Rollen	1.5
31.7.2024	CAD Z-Achsen zauberei	1.0
1.8.2024	CAD Z-Achse redesign	5.0
9.8.2024	CAD YZ-Achse grobe fertigstellung	5.0
10.8.2024	CAD YZ-Achse feinerschliff	4.0
11.8.2024	CAD YZ-Achse + X-Achse beginn	2.0
12.8.2024	CAD X-Achse	1.0
13.8.2024	CAD X-Achse side roller	4.0
14.8.2024	CAD X-Achse side roller 2. side	2.0
15.8.2024	CAD X-Achse Mid rollers, side Wheels, YZ-Achse Spiegelung	6.0
16.8.2024	CAD YZ-Achse Lichttaster, X-Achse	1.0
17.8.2024	X-Achse Schleppkettengedanken Auslegung	6.0
19.8.2024	Schleppenderketten einplanung	2.0
20.8.2024	CAD vertikale Schleppkette	3.0
20.8.2024	Sponsoren-E-mail beginn	1.3
21.8.2024	Kontaktdaten, Projektzusammenfassung	1.5
22.8.2024	CAD Schlitten Top	1.0
24.8.2024	Stückliste, CAD Schlitten Top	2.0
25.8.2024	CAD X-Top Verbindung, Umlenkung	5.0

<b>Datum</b>	<b>Tätigkeit</b>	<b>Stunden</b>
27.8.2024	CAD Rahmen Aufhängungen	2.0
29.8.2024	CAD Umlenkungen und Motoraufhängungen	5.0
30.8.2024	CAD Endschalter und Rahmen beginn	3.0
31.8.2024	CAD Rahmen, Lagerschrank beginn	6.0
1.9.2024	CAD Lagerschrank und Querfördererausschnitt	2.0
2.9.2024	Verbindungsslider implementieren	1.0
3.9.2024	Bugs beheben, Weidmüller Sortiment Bauteile auswählen	4.0
4.9.2024	Stack Bug behoben und CAD Querförderer	7.0
5.9.2024	CAD Mech. weitestgehende Fertigstellung	4.0
9.9.2024	Latex aufsetzen	2.5
10.9.2024	Meeting Weidmüller	2.0
11.9.2024	Schalschrank konzeptionieren	2.25
15.9.2024	Stückliste anfertigen	2.0
17.9.2024	Autolager Demontieren für Bauteilbeschaffung	2.5
18.9.2024	Verbindungstest, Suchalgorhythmus Rust implementation	6.0
19.9.2024	Suchag. Fertig implementiert, CAD Rollen gezeichnet	3.0
24.9.2024	Projektmanagement	3.0
25.9.2024	Änderungen V-Slot-Aufhängung, Project-Libre	1.0
26.9.2024	Igus, Weidmüller, Motoren ansteuern die 1.	4.0
28.9.2024	Bux im Lageralgorithmus beheben	1.5
1.10.2024	CAD	1.0
3.10.2024	Profile bearbeiten	1.5
8.10.2024	Latex Vorlage	1.0
9.10.2024	Raumeinrichtung	3.0
21.10.2024	Rahmenbau beginn	2.0
22.10.2024	Rahmenbau und Drehstromverlegung, Auftragsvorbereitung X-Aufhängung	5.5
23.10.2024	Rahmenbau	3.0
24.10.2024	Rahmenbau	3.0
25.10.2024	CAD XZ-Redesign	3.0
26.10.2024	CAD XZ-Redesign	7.0
27.10.2024	Auftragsvorbereitung X-Achse	3.0
28.10.2024	Auftragsvorbereitung X-Achse	2.0
5.11.2024	Beginn Umlenkrollen Drehen	0.8
6.11.2024	Weidmüller DP Inbetriebnahme	1.0
7.11.2024	Weidmüller DP ansteuern	3.5
8.11.2024	Weidmüller DP ansteuern	3.5
12.11.2024	Umlenkrollen Drehen	3.5

Datum	Tätigkeit	Stunden
19.11.2024	Umlenkrollen Drehen, Fräsen	2.5
20.11.2024	DAS: Drehen	1.0
21.11.2024	DAS: TFIDF	2.0
22.11.2024	DAS	4.5
23.11.2024	API stack , DAS	3.0
24.11.2024	DAS	2.0
25.11.2024	DAS	1.5
26.11.2024	Drehen	3.5
29.11.2024	DAS	4.0
3.12.2024	Drehen, DAS: Maschinenbau	5.0
6.12.2024	Website, DAS	3.5
10.12.2024	DAS: Software, CAD	5.0
13.12.2024	Drehen	3.5
7.1.2025	CNC-Fräsen, Hülsen Drehen	3.5
8.1.2025	X-Achse Zusammenbauen anfangen	2.0
10.1.2025	X-Achse Zusammenbauen	4.0
14.1.2025	X-Achse Zusammenbauen	4.0
15.1.2025	X-Achse Zusammenbauen	4.5
17.1.2025	Tag der offenen Türe	5.0
21.1.2025	TIA Portal Verbindung	3.5
24.1.2025	Lasern, Förderband, Mechanik	3.5
4.2.2025	Sicherheitstechnik-Besprechung, SPS - Server Kommunikation	3.5
14.2.2025	WMS Location Updateing	1.0
17.2.2025	DAS: Allgemeinteil	1.0
21.2.2025	Ref, Fräsen, Da-Schreiben	5.0
25.2.2025	Zusammenbauen	4.0
28.2.2025	Fräsen, E-Plan,	3.5
2.3.2025	DAS: Aufbau	1.5
3.3.2025	DAS: Besprechungsprotokolle	1.0
5.3.2025	Umlenkung und Motoren einbauen	4.5
6.5.2025	DAS	2.5
9.3.2025	DAS XZ-Achse	1.0
10.3.2025	DAS	3.0
11.3.2025	DAS	3.0
14.3.2025	Verkabelung Schaltschrank	4.0

## 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] knapp. Online. URL: <https://www.knapp.com/> (besucht am 19.03.2025).
- [6] weidmueller. Online. URL: [https://www.weidmueller.at/de/unternehmen/unser-unternehmen/wer\\_wir\\_sind/index.jsp](https://www.weidmueller.at/de/unternehmen/unser-unternehmen/wer_wir_sind/index.jsp) (besucht am 19.03.2025).
- [7] igus. Online. URL: <https://www.igus.at/unternehmen> (besucht am 01.03.2025).
- [8] lapp. Online. URL: <https://www.lapp.com/de/at/unternehmen/ueber-lapp-oesterreich/e/000123> (besucht am 03.03.2025).
- [9] mädler. Online. URL: <https://www.maedler.de/Static/Videos> (besucht am 03.03.2025).
- [10] elektro4000de. URL: <https://www.elektro4000.de/Kabel-Leitungen/Kabeltragsysteme/Stiel-Profilschienen/Verbindungen/Item-Standard-Verbindungssatz-0-0-026-07::1743227.html> (besucht am 19.11.2024).
- [11] globalsources.com. URL: <https://www.globalsources.com/Plastic-coated/plastic-coated-bearing-1197379614p.html> (besucht am 19.11.2024).
- [12] 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).
- [13] 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).
- [14] OpenAI. *ChatGPT: OpenAI's Language Model*. <https://openai.com/chatgpt>. 2024. (Besucht am 21.11.2024).
- [15] 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).
- [16] Berufsschule-Gmunden. *SPS-Grundlagen*. Online. URL: [https://bs-gmunden1.ac.at/fileadmin/user\\_upload/Berufsschule-Gmunden1/Import/Lehrstoffe/SPS\\_S7\\_TIA-Portal/1.\\_S7-1200-1500\\_Grundlagen.pdf](https://bs-gmunden1.ac.at/fileadmin/user_upload/Berufsschule-Gmunden1/Import/Lehrstoffe/SPS_S7_TIA-Portal/1._S7-1200-1500_Grundlagen.pdf) (besucht am 15.03.2025).

- [17] 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).
- [18] Siemens. *Technologieobjekte*. Online. URL: [https://cache.industry.siemens.com/dl/files/134/109743134/att\\_925648/v2/109743134\\_S7-1500T\\_TechnologyObjects\\_DOC\\_v10\\_de.pdf](https://cache.industry.siemens.com/dl/files/134/109743134/att_925648/v2/109743134_S7-1500T_TechnologyObjects_DOC_v10_de.pdf) (besucht am 15.03.2025).
- [19] SPS-Lehrgang. *SPS-Programmierung*. Online. URL: <https://www.sps-lehrgang.de/sps-programmierung/> (besucht am 19.03.2025).
- [20] wikipedia. *Anweisungsliste*. Online. URL: <https://de.wikipedia.org/wiki/Anweisungsliste> (besucht am 19.03.2025).
- [21] Siemens. *Structured Code Language*. Online. URL: <https://www.automation.siemens.com/sce-static/learning-training-documents/tia-portal/advanced-programming-s7-1200/sce-051-201-scl-s7-1200-r1709-de.pdf> (besucht am 21.03.2025).
- [22] Siemens. *Einbindung von Bibliotheken*. Online. URL: <https://support.industry.siemens.com/cs/document/37364723/wie-k%C3%B6nnen-sie-globale-bibliotheken-im-tia-portal-%C3%B6ffnen-bearbeiten-und-hochr%C3%BCsten-?dti=0&lc=de-WW#:~:text=%C3%96ffnen%20Sie%20die%20Task%20Card,Datei-Endung%20%22al%22> (besucht am 21.03.2025).
- [23] unknown. *User Manual CL57T Closed Loop Stepper Driver*. PDF. 22. Juli 2024.
- [24] Xovi Glossar. Online. URL: <https://www.xovi.de/was bedeutet http/#:~:text=HTTP%20ist%20die%20Abk%C3%A4rzung%20f%C3%BCr,Client%20und%20Webserver%20verwendet%20wird.> (besucht am 18.02.2025).
- [25] OPC Router. Online. URL: <https://www.opc-router.de/was-ist-opc-ua/> (besucht am 18.02.2025).
- [26] Elektronik Kompendium. *HTTP Client Server Kommunikation*. Online. URL: <https://www.elektronik-kompendium.de/sites/net/0902231.htm#:~:text=Die%20Kommunikation%20findet%20nach%20dem,Server%20ist%20diese%20Verbindung%20beendet.> (besucht am 21.03.2025).
- [27] Siemens. *Libraries for Communication for SIMATIC Controllers*. Online. 23. März 2025.
- [28] buerklin. Online. URL: <https://www.buerklin.com/de/elektronik-kompetenz/instandhaltung/sicherheit-im-schalschrankbau/> (besucht am 17.03.2025).
- [29] copyservice-shop. Online. URL: <https://www.copyservice-shop.de/Alu-Verbundplatte-Dibond-A4-A3-etc> (besucht am 13.02.2025).
- [30] Komaxgroup. Online. URL: <https://www.komaxgroup.com/en/stories/experience-the-smart-cabinet-building-initiative-up-close-and-personal> (besucht am 13.02.2025).

- [31] 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).
- [32] elektronet. Online. URL: [https://www.elektro.net/media/file/DYh4uat3cf/187f5a2f/DE\\_07\\_10\\_PP08.pdf](https://www.elektro.net/media/file/DYh4uat3cf/187f5a2f/DE_07_10_PP08.pdf) (besucht am 19.03.2025).
- [33] beckhoff. Online. URL: [https://infosys.beckhoff.com/index.php?content=..../content/1031/xts\\_hygienic\\_design\\_ba/7170992011.html&id=8528766451858564804](https://infosys.beckhoff.com/index.php?content=..../content/1031/xts_hygienic_design_ba/7170992011.html&id=8528766451858564804) (besucht am 19.03.2025).
- [34] Stepperonline. *Nema-SM-Steckerbelegung*. Online. URL: <https://tinyurl.com/29e7hdam> (besucht am 20.03.2025).
- [35] Siemens. *S7-1500/S7-1500T Achsfunktionen V5.0 im TIA Portal V16*. Online.
- [36] 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).
- [37] 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).
- [38] 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).
- [39] 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).
- [40] Sivago Semiconductors. Online. URL: <https://www.sivago.com.cn/upload/pdf/2022/TP808.pdf> (besucht am 28.02.2025).
- [41] Kingbright. Online. URL: [https://cdn-reichelt.de/documents/datenblatt/A500/LED5MMSTGE\\_LED5MMSTGN\\_LED5MMSTR~KIN.pdf](https://cdn-reichelt.de/documents/datenblatt/A500/LED5MMSTGE_LED5MMSTGN_LED5MMSTR~KIN.pdf) (besucht am 10.03.2025).
- [42] Siemens. Online. URL: <https://mall.industry.siemens.com/mall/Catalog/DatasheetDownload?downloadUrl=teddatasheet%2F%3Fformat%3DPDF%26caller%3DMall%26mlfbs%3D3RK2200-0CE02-0AA2%26language%3Dde> (besucht am 14.03.2025).
- [43] Baumer. Online. URL: [https://www.baumer.com/de/de/service-support/funktionsweise/funktionsweise-und-technologie-von-lichtschrangen-und-lichttastern/a/Know-how\\_Function\\_Lichtschrangen-Lichttaster](https://www.baumer.com/de/de/service-support/funktionsweise/funktionsweise-und-technologie-von-lichtschrangen-und-lichttastern/a/Know-how_Function_Lichtschrangen-Lichttaster) (besucht am 22.03.2025).
- [44] SICK. Online. URL: [https://cdn.sick.com/media/pdf/8/58/658/dataSheet\\_WTB4FP-22161100A00\\_1124609\\_de.pdf](https://cdn.sick.com/media/pdf/8/58/658/dataSheet_WTB4FP-22161100A00_1124609_de.pdf) (besucht am 22.03.2025).
- [45] SICK. *Produktbild*. Online. URL: <https://www.sick.com/at/de/catalog/produkte/detektionssensoren/lichttaster-und-lichtschrangen/w4/wtb4fp-22161100a00/p/p671658?tab=detail#technical-details> (besucht am 21.03.2025).
- [46] SICK. *Produktbild*. Online. URL: <https://cdn.sick.com/media/pim/5/35/135/IM0058135.png>.

- [47] SICK. Online. URL: [https://cdn.sick.com/media/pdf/4/44/644/dataSheet\\_CLV615-D2410\\_1068608\\_de.pdf](https://cdn.sick.com/media/pdf/4/44/644/dataSheet_CLV615-D2410_1068608_de.pdf) (besucht am 14.03.2025).
- [48] SICK. *SICK Lector6xx CLV6xx Funktionsbaustein: Technische Informationen*. PDF. 18. März 2025.
- [49] Bernhard Wurnitsch. *Einführung Bussysteme der Automatisierungstechnik (Asi, Profibus, Profi-NET)*. WORD Dokument. 16. Okt. 2023.
- [50] Siemens. Online. URL: [https://support.industry.siemens.com/cs/document/73257075/wie-kann-ein-sinamics-oder-ein-antrieb-eines-drittanbieters-%C3%BCber-ger%C3%A4testammdatei-\(gsd\)-in-tia-portal-eingebunden-werden?dti=0&lc=de-BG](https://support.industry.siemens.com/cs/document/73257075/wie-kann-ein-sinamics-oder-ein-antrieb-eines-drittanbieters-%C3%BCber-ger%C3%A4testammdatei-(gsd)-in-tia-portal-eingebunden-werden?dti=0&lc=de-BG) (besucht am 23.03.2025).
- [51] Heinen Elektronik GmbH. Online. URL: <https://heinen-elektronik.de/glossar/profibus/> (besucht am 23.03.2025).
- [52] Bürkert. Online. URL: <https://www.buerkert.de/de/service-support/knowledge-center/glossar/PROFINET-der-sichere-Kommunikationsstandard-in-industriellen-Netzwerken> (besucht am 23.03.2025).
- [53] Sigurd Seyr u. a. *Elektroinstallation Blitzschutz Lichttechnik*. Verlag Jugend und Volk GmbH, 2022.
- [54] LEIFphysik. Online. URL: <https://www.leifiphysik.de/elektrizitaetslehre/ohmsches-gesetz-kennlinien/ausblick/gefahr-durch-strom> (besucht am 18.03.2025).

# Abbildungsverzeichnis

Abb. 2.1	Item Profil mit Standartverbindungssatz, Quelle: [10] . . . . .	18
Abb. 2.2	V-Slot-Profil mit V-Wheel, Quelle: [11] . . . . .	18
Abb. 2.3	Prototypen . . . . .	19
Abb. 2.4	X-Achse . . . . .	22
Abb. 2.5	X-Achse Gesamtansichten . . . . .	22
Abb. 2.6	Zahnriemenkontraktion grafisch dargestellt . . . . .	25
Abb. 2.7	YZ-Achse . . . . .	28
Abb. 2.8	XY-Shuttle Gesamtansicht . . . . .	28
Abb. 2.9	Bauteilzeichnung Umlenkrolle . . . . .	30
Abb. 2.10	Umsetzung der Umlenkrollen . . . . .	31
Abb. 2.11	Gesamtüberblick des Servers . . . . .	33
Abb. 2.12	Diagramm der BMOS bereitstellung . . . . .	38
Abb. 2.13	Datenbankschema des AFSS . . . . .	39
Abb. 3.1	Arbeitsweise einer SPS [16] . . . . .	45
Abb. 3.2	Organisationsbausteine [17] . . . . .	46
Abb. 3.3	Zugriff auf Global-DB [17] . . . . .	47
Abb. 3.4	Sequenzdiagramm der Kontrollsignale [23] . . . . .	52
Abb. 3.5	Baustein MC_GearIn . . . . .	55
Abb. 3.6	HTTP-Client-Server Kommunikation [26] . . . . .	57
Abb. 3.7	POST-Put Baustein [27] . . . . .	58
Abb. 4.1	Serverschrank bei Übergabe an das AFSS-Team . . . . .	64
Abb. 4.2	Dibond-Platte, Quelle: [29] . . . . .	66
Abb. 4.3	Digitaler Zwilling, Quelle: [30] . . . . .	67
Abb. 4.4	Sommerprototyp eines Serverschrankes . . . . .	68
Abb. 4.5	Serverschrank konstruiert . . . . .	69
Abb. 4.6	Siemens SITOP in Fusion360 . . . . .	70
Abb. 4.7	Der Schaltschrank in Fusion gezeichnet . . . . .	71
Abb. 4.8	Schaltplan: Netzteile . . . . .	76
Abb. 4.9	13te Seite des Schaltplans, SM ohne Bremse . . . . .	79
Abb. 4.10	Zusätzliche Kurzschließung von Spulen bei horizontalen Antrieben . . . . .	80
Abb. 4.11	40Ncm Schrittmotoren im Schaltplan . . . . .	81
Abb. 4.12	Verbindung der Sensoren mit den ASI-Slaves . . . . .	83
Abb. 5.1	Endschalter . . . . .	85
Abb. 5.2	Schaltplan Referenzplatine . . . . .	87
Abb. 5.3	Leiterplattenplan Referenzplatine . . . . .	88
Abb. 5.4	SICK Sensoren . . . . .	90
Abb. 5.5	Barcodescanner Geräteübersicht im TIA-Portal . . . . .	91
Abb. 5.6	Barcodescanner Beobachtungstabelle im TIA-Portal . . . . .	92
Abb. 5.7	Topologie verschiedener Feldbusssysteme [50] . . . . .	93
Abb. 6.1	Scrum Projektplan mit Tasks . . . . .	98

Abb. 6.2	Kostenaufstellung . . . . .	102
Abb. 6.3	Besprechungsprotokoll 10.12.2024 . . . . .	103
Abb. 6.4	Besprechungsprotokoll 16.10.2024 . . . . .	104
Abb. 6.5	Besprechungsprotokoll 10.12.2024 . . . . .	105
Abb. 6.6	Besprechungsprotokoll x.x.xxxx . . . . .	106

## **Tabellenverzeichnis**

Tab. 2.1	X-Achse unbeladen und eingefahren . . . . .	24
Tab. 2.2	X-Achse beim Ladevorgang . . . . .	24