TECHNISCHE UNIVERSITÄT WIEN
Vienna University of Technology

ACIN
AUTOMATION & CONTROL INSTITUTE
INSTITUT FÜR AUTOMATISIERUNGS- & REGELUNGSTECHNIK

IEC 61499 basierende Basisautomation für ein verteilt gesteuertes Hochregallager

DIPLOMARBEIT

Ausgeführt zum Zwecke der Erlangung des akademischen Grades eines

Diplom-Ingenieurs (Dipl.-Ing.)

unter der Leitung von

Univ.-Prof. Dipl.-Ing. Dr. sc. techn. Georg Schitter
Dipl.-Ing. Dr. techn. Alois Zoitl

eingereicht an der

Technischen Universität Wien
Fakultät für Elektrotechnik und Informationstechnik
Institut für Automatisierungs- und Regelungstechnik

von

Andreas Neubauer
Nestroygasse 3
3830 Waidhofen an der Thaya
Österreich

Wien, im Oktober 2012

Vorwort

Zuerst möchte ich mich bei allen bedanken, die dazu beigetragen haben, dass diese Diplomarbeit entstanden ist. In diesem Sinne bedanke ich mich bei Professor Georg Schitter für seine Betreuung. Besonderer Dank gebührt meinem betreuenden Assistent Alois Zoitl, der mir mit vielen hilfreichen Besprechungen bei der Umsetzung dieser Arbeit geholfen hat. Ebenfalls möchte ich mich bei Ingo Hegny bedanken für die Unterstützung beim Aufbau des mechanischen Modells.

Der größte Dank gebührt aber meiner Frau Nicole für die Unterstützung während meines Studiums und meinen Eltern Josef und Maria Neubauer, die mir ein sorgenfreies Studieren ermöglichten.

Kurzfassung

Logistiksysteme müssen heutzutage sehr wandlungsfähig sein, um sich schnell an neue Gegebenheiten anpassen zu können. Dies ist jedoch mit herkömmlichen Lösungsansätzen kaum möglich. Durch den Einsatz von verteilten Steuerungskonzepten erhofft man sich diesen Anforderung gerecht zu werden. Die Evaluierung verteilter Steuerungskonzepte erfolgt zu meist mittels Simulation. Damit kann bereits ein Großteil des Funktionsumfangs evaluiert werden, jedoch können sie die Evaluierung auf industriellen Anlagen bzw. Testanlagen nicht vollständig ersetzen. Aus diesem Grund beschäftigt sich diese Arbeit mit der Erweiterung des bestehenden Palettentransportsystems, welches sich im Odo Struger Labor der TU Wien befindet, um ein Hochregallager. Das entwickelte Hochregallager soll in weiterer Folge als Testplattform für intelligente verteilte Steuerungskonzepte dienen. Die Integration des neuen Anlageteils erfolgt dabei so, dass möglichst alle Anforderungen eines modernen Hochregallagers erfüllt werden. Das entwickelte Hochregallager besteht aus drei Lagergassen und besitzt insgesamt 108 Lagerplätze.

Die entwickelte Steuerung des Hochregallagers basiert auf den Standard IEC 61499 und stellt die Basisfunktionen für den Betrieb des Lagers zur Verfügung. Auf diese Funktionen kann eine übergelagerte Steuerung über eine definierte Schnittstelle zugreifen. Um die übergelagerte Steuerung zu entlasten, wurde versucht möglichst viel an Funktionalität in die entwickelte Basissteuerung zu integrieren. Dazu wurde eine Analyse eines verteilten Steuerungskonzepts durchgeführt. Basierend auf dieser Analyse wurde anschließend der Funktionsumfang der Basissteuerung festgelegt. Die Umsetzung der Basissteuerung erfolgte mit Hilfe des ,,Framework for Distributed Industrial Automation and Control" (4DIAC). Auf Grund der verwendeten Komponenten musste ein CAN Interface entwickelt werden, da in 4DIAC noch keine dementsprechenden Funktionsblöcke zur Verfügung standen. Das CAN Interface ermöglicht die Kommunikation zwischen Steuerung und Motorcontroller. Die notwendigen Anforderungen an das zu entwickelnde CAN Interface wurde aus dem CANopen Standard abgeleitet, da dieser als Kommunikationsprotokoll verwendet wurde. Bei der Analyse des verteilten Steuerungskonzepts wurde festgestellt, dass für die Abarbeitung der Bewegungsabläufe für das Ein-, Aus- und Umlagern keine weiteren Information bis auf Quell- und Zieladresse notwendig sind. Diese Bewegungsabläufe konnten in der Basissteuerung umgesetzt und mit Hilfe des entwickelten mechanischen Aufbaus erfolgreich getestet werden. Obwohl das entwickelte Hochregallager noch Potential zur Optimierung besitzt, ist in Verbindung mit dem bestehenden Transportsystem ein komplexes Logistiksystem entstanden, mit dem neu entwickelte verteilte Steuerungskonzepte praxisnahe getestet werden können.

Abstract

Today logistic systems must be very versatile to adapt quickly to new requirements. However, this is hardly possible with conventional approaches. The use of distributed control concepts allow to fulfill these requirements. The evaluation of distributed control concepts is usually done by simulation. Thereby the majority of the functionality could be evaluated, but simulation could not completely replace the evaluation with industrial plants and test plants. For this reason this study deals with the enlargement of the existing palette transport system, which is located in the Odo Struger Labatory of the Vienna Technical University, by an automatic storage and retrieval system. The integration of the new system should be done in a way, that all requirements of a modern automatic storage and retrieval system are fulfilled. The requirements of the automatic storage and retrieval system results on one side from the requirements of the coworkers of the Automation and Control Institute and results on the other side from the requirements by the state of the art. Based on these requirements a CAD model is developed, which can be easily implemented. The developed model has a total number of 108 bins.

The developed automatic storage and retrieval system will serve as a test platform for intelligent distributed control concepts. The developed control of the automatic storage and retrieval system is based on the standard IEC 61499 and provides the basic functions for the operation of an storage system. The functions are available to an higher level intelligent control system, which have access to these functions via a defined interface. The goal is to relieve the higher level control by integrate as much functionality in the base control as possible. Therefor an analysis of a distributed control concept for an automatic storage and retrieval system was made. Based on this analysis, the functions of the base control were defined. The implementation of the base control was performed using the Framework for Distributed Industrial Automation and Control (4DIAC). Because of the used components and the fact that in 4DIAC are no function blocks for CAN communication are available, a CAN interface had to be develop. The developed interface allows the basic control to communicate with the motor control. With the help of the analysis of the distributed control concept there has been established, that for the execution of the movements for import, export and repositioning no other information necessary than source and destination address. These movements has been implemented in the base control and has been successfully tested with the help of the developed aisle. Although the developed automatic storage and retrieval system can be further optimized, in combination with the existing transport system a complex logistics system has been formed.

Inhaltsverzeichnis

Abbildungsverzeichnis

Abkürzungsverzeichnis

4DIAC	Framework for Distributed Industrial Automation and Control
AS/RS	Automatic Storage and Retrieval System
CAD	Computer Aided Design
CAL	CAN Application Layer
CAN	Controller Area Network
CiA	CAN in Automation
CFB	Composite Function Block
ECC	Execution Control Chart
FB	Function Block
FIFO	First In First Out
HLC	High Level Control
HRL	Hochregallager
IEC	International Electrotechnical Commision
ISO	International Organization for Standardization
LAM	Lastaufnahmemittel
LLC	Low Level Control
OSI	Open System Interconnection
PDO	Process Data Object
PPS	Produktionsplanungssystem
PTS	Palettentransportsystem
RBG	Regalbediengerät
RFID	Radio Frequency Identification
SDO	Service Data Object
SIFB	Service Interface Function Block
SPS	Speicherprogrammierbare Steuerung
TCP/IP	Transmission Control Protocol/Internet Protocol
UML	Unified Modeling Language
WWS	Warenwirtssystem

1. Einleitung

Produktions- und Logistiksysteme müssen heutzutage so dynamisch, vernetzt und wandlungsfähig sein, wie die individuellen Anforderungen der Kunden, deren Wünsche diese Anlagen zu befriedigen haben. Dabei ist es aber kaum möglich, diesem Umstand mit herkömmlichen Lösungsansätzen gerecht zu werden – auch wenn dabei modernste Technik eingesetzt wird. [7]

Im Bereich der klassischen Automatisierungstechnik in der Intralogistik besteht derzeit die Gefahr der Deautomatisierung, da die Anlagen als zu unflexibel in Bezug auf Veränderungen der Anforderungen und der Prozesse empfunden werden. Die Inflexibilität rührt daher, dass eine Anpassung der automatisierten Anlagen sehr kostenaufwändig und risikoreich ist. Der Grund hierfür ist unter anderem, dass die Funktionen zur Erbringung einer logistischen Leistung auf verschiedene Ebenen und Systeme verteilt sind. Die Gesamtfunktion wird durch eine ebenfalls auf mehrere Ebenen verteilte Steuerungssoft- und -hardware erbracht. Eine selbsttätige Anpassung der Struktur an neue Aufgaben sehen die derzeitigen Systeme der Intralogistik nicht vor. Bei einer Veränderung der Abläufe müssen heute alle diese Systeme typischerweise durch mehrere unterschiedliche Personen geändert werden. [5]

Bei großen Änderungen muss meist ein tiefer Eingriff in das System erfolgen. Dies kann zu Fehlern führen, die erst bei der Wiederinbetriebnahme des Gesamtsystems auftreten. Solche Fehler können nur schwer im Vorfeld gefunden werden. Deswegen wird durch aktuelle Entwicklungen versucht die Modularisierung von Produktions- und Materialflussanlagen voranzutreiben. Dadurch soll die Projekt übergreifende Wiederverwendbarkeit gesteigert und die Systemkomplexität gesenkt werden. Um solche Modularisierungen entwickeln zu können müssen auch die vormals zentralen Steuerungsfunktionen dezentralisiert werden. Die dezentralen Systeme sind dann in der Lage sich selbstständig zu konfigurieren und können sich an Änderungen der Anforderungen automatisch anpassen. Dazu werden jedoch intelligente verteilte Steuerungskonzepte benötigt, die eine selbstständige Anpassung an neue Anforderungen ermöglichen.

Bei solchen verteilten Steuerungskonzepten kommt der Kommunikation zwischen den einzelnen dezentralen Steuerungen ein hoher Stellenwert zu, da in dezentralen Systemen die Systeminformationen nicht mehr gesammelt in einer, sonder verteilt über mehrere Steuerungen verfügbar sind. Diese müssen nun die Informationen untereinander austauschen um die Anforderungen an das Gesamtsystem erfüllen zu können. Ziel ist es nicht jeder Steuerung alle Informationen des Gesamtsystems zur Verfügung zu stellen, sondern nur jene Informationen, die für eine Entschei-

dungsfindung notwendig sind. Die Größe und Menge der zu übertragenden Daten wirken sich auf die Leistungsfähigkeit des Gesamtsystems aus. Um den Aufwand an zu übertragenden Daten zu verringern, werden intelligente Funktionen für die Entscheidungsfindung benötigt. Diese können auf mehreren Steuerungen ablaufen. Dadurch müssen nur mehr Zwischenergebnisse übertragen werden, was zu einer Reduktion der zu übertragenden Daten führt. Typischerweise werden entwickelte verteilte Steuerungskonzepte durch Simulation überprüft, da industriell genutzte Anlagen meist für Forschungszwecke nicht zur Verfügung stehen. Mit heutigen Simulationen kann bereits ein Großteil des Funktionsumfangs von verteilten Steuerungskonzepten evaluiert werden, jedoch können sie die Evaluierung auf industriellen Anlagen bzw. Testanlagen nicht vollständig ersetzen.

1.1. Aufgabenstellung

Ziel dieser Arbeit ist es, ein Hochregallagermodell zu entwickeln, welches das bestehende Palettentransportsystem (PTS) im Odo Struger Labor der TU Wien (siehe Abbildung 1.1) erweitert. Das entwickelte Hochregallager (HRL) soll die Funktionalität eines herkömmlichen Hochregallagers bieten. Dabei wird der Funktionsumfang aus dem Stand der Technik definiert. Das HRL soll aus mehreren Lagergassen bestehen und eine ausreichende Anzahl an Lagerplätzen besitzen um praxisnahe Forschung zu ermöglichen.

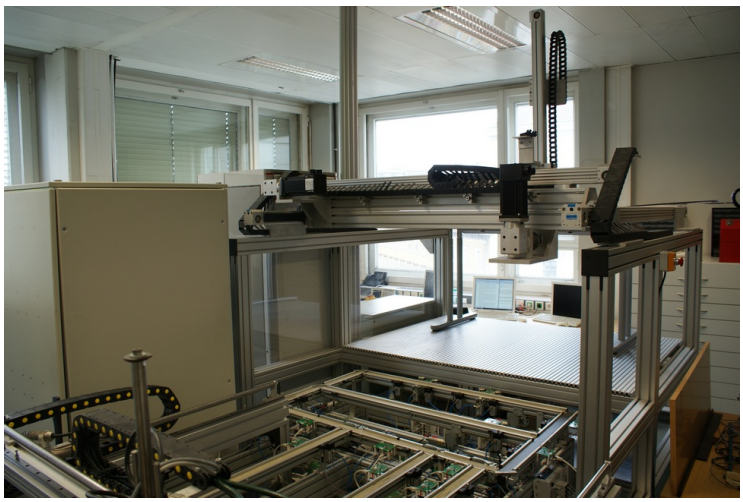


Abbildung 1.1.: Verteiltes Palettentransportsystem im Odo Struger Labor der TU Wien

Dazu soll in einem ersten Schritt der Entwurf des Hochregallagers verifiziert werden, welcher im Zuge eines vorangegangenen Projekts im Rahmen der Lehrveranstaltung „Automation, Vertiefung" erstellt worden ist. Aufbauend auf dem Entwurf soll ein

genaues CAD-Modell des Hochregallagers angefertigt werden. Die für den Aufbau benötigten Teile sollen bestellt bzw. gefertigt werden. Anschließend soll der mechanische Entwurf durch Aufbau einer Lagergasse evaluiert werden. Zusätzlich sollen die notwendigen mechanischen Änderungen für die Anbindung an das PTS erhoben werden.

In einem zweiten Schritt sollen alle Basisfunktionen, die für den Betrieb eines HRLs notwendig sind, identifiziert werden. Dazu soll ein verteiltes Steuerungskonzept analysiert werden. Aufbauend auf den durch die Analyse gewonnenen Daten und dem Stand der Technik sollen die notwendigen Funktionen identifiziert und in weiterer Folge im Standard IEC 61499 implementiert werden. Die so entwickelte Steuerung soll die implementierten Basisfunktionen einer übergelagerten Steuerung über ein geeignetes Interface zur Verfügung stellen. Die Steuerung soll mit Hilfe des mechanischen Aufbaus einer Lagergasse evaluiert werden.

1.2. Leitfaden durch die Arbeit

Die Arbeit gliedert sich im wesentlichen in drei Teile. Im ersten Teil wird der allgemeine Aufbau von HRL behandelt sowie die Strategien, die von modernen HRL-Steuerungen unterstützt werden. Zusätzlich werden verschiedene Technologien vorgestellt, wie zum Beispiel der Kommunikationsstandard CANopen, welche für die Realisierung der Steuerungssoftware eine wichtige Rolle spielen.

Der zweite Teil beschäftigt sich mit dem mechanischen Aufbau des HRL. Dazu wird in einem ersten Schritt eine Bestandsaufnahme durchgeführt und die notwendigen Anforderungen an das HRL erhoben. In einem nächsten Schritt werden die notwendigen Änderungen für das an das HRL angrenzende Transportsystem erhoben. Anschließend wird die mechanische Umsetzung und die einzelnen Anlageteile im Detail vorgestellt. Abschließend wird auf die Sicherheit der Anlage eingegangen und verschiedene Schutzfunktionen für einen sicheren Betrieb erklärt.

Im dritten Teil wird auf die Umsetzung der Basisfunktionen der HRL-Steuerung eingegangen. Für die Festlegung des Funktionsumfangs wird eine Analyse eines verteilten Steuerungskonzept für ein HRL durchgeführt. Aufbauend auf der Analyse wird der Umfang der Basisfunktionen der HRL-Steuerung festgelegt. Die Steuerungssoftware wurde mit Hilfe eines Ebenenmodells realisiert. Im weiteren werden die wichtigsten Funktionsblöcke der jeweiligen Ebenen vorgestellt. Eine wichtige Ebene ist das CAN Interface. Dieses ist notwendig, um mit den verwendeten Motorcontrollern kommunizieren zu können. Da in 4DIAC noch keine dementsprechenden Funktionsblöcke zur Verfügung standen, mussten welche entwickelt werden. Diese werden im weiteren genauer erklärt.

Abschließend werden in einem Ausblick mögliche Forschungsgebiete vorgestellt, welche durch die Ergebnisse dieser Arbeit ermöglicht werden sowie eine Zusammenfassung der Arbeit gegeben.

2. Stand der Technik

Dieses Kapitel behandelt die Grundlagen, die die Basis dieser Arbeit bilden. Zusätzlich wird im Bezug auf das verteilte Steuerungskonzept und der Umsetzung der Basisfunktionen einer HRL-Steuerung der aktuelle Stand der Technik aufgezeigt. Ein wesentlicher Schwerpunkt der Arbeit ist die Erstellung und Ansteuerung eines Hochregallagers für Testzwecke im Odo Struger Labor. Aus diesem Grund wird zu Anfang auf den allgemeinen Aufbau eines Hochregallagers eingegangen und ein Überblick über die verschiedensten Typen von HRL gegeben. Im weiteren werden einige Aspekte zur Planung von HRL vorgestellt und einige Strategien erläutert, welche in aktuellen HRL-Steuerungen umgesetzt sind. Anschließend wird ein Einblick in ein Agenten basierendes Auftragsvergabeproblem gegeben.

Die Ansteuerung des Regalbediengeräts erfolgt über drei Motorcontroller, die mittels CAN-Bus an die Steuerung angeschlossen sind. Für die Kommunikation zwischen Steuerung und Motorcontroller wird CANopen verwendet. Da in IEC 61499 noch keine passenden Funktionsblöcke für CAN bzw. CANopen zur Verfügung standen, mussten welche entwickelt werden. Deshalb wird abschließend ein Überblick über CAN, CANopen und Kommunikationsfunktionsblöcke in IEC 61499 gegeben.

2.1. Hochregallager

Hochregallager (HRL) dienen der automatischen Lagerung großer Stückzahlen von Paletten. Die Kapazität heutiger HRL reicht dabei von wenigen tausend bis zu mehreren hunderttausenden Lagerplätzen. Die Lagerstellen besitzen feste Feldbreiten und bieten Platz für eine oder mehrere Paletten. Die Paletten besitzen alle die gleiche Grundfläche und sind meist standardisiert. Der bekannteste Vertreter ist die Euro-Poolpalette (800 mm $\times$ 1200 mm), wobei größere Ladungseinheiten kein Problem darstellen. Von HRL spricht man in der Regel ab einer Regalhöhe von 12 Metern. Die Maximalhöhe heutiger Hochregale beträgt ca. 50 Meter. Die Regalkonstruktion besteht zumeist aus Stahlprofilen und ist ab einer Höhe von 28 Metern Gebäude tragend, da diese Regalkonstruktion kostenmäßig die bessere Lösung ist. Die Mehrkosten für die stärkere Dimensionierung des Stahlbaus (Dach- und Wandlasten, etc.) werden durch den Wegfall der sonst separat zu erstellenden Gebäudehülle mehr als kompensiert.

2.1.1. Typen von Hochregallager

Hochregallager gibt es in den verschiedensten Ausführungen und mit den unterschiedlichsten Optionen, aus diesem Grund soll in diesem Abschnitt ein kurzer Überblick über die verschiedenen Typen von HRL gegeben werden. Die einfachste Realisierung eines HRLs besteht aus mehreren Lagergassen wobei jede Lagergasse aus zwei Regalen besteht. In der Mitte jeder Lagergasse ist ein RBG installiert, welches die beiden Regale bewirtschaftet. Das RBG besitzt eine Entnahmeachse und kann in einem Zyklus immer nur ein Palette transportieren. Die Regallagerung ist statisch und die Regalkonstruktion besitzt einfach tiefe Lagerplätze. Dadurch kann jede Ladungseinheit direkt vom RBG erreicht werden. Der schematische Aufbau eines solchen HRLs ist in Abbildung 2.1 ersichtlich.

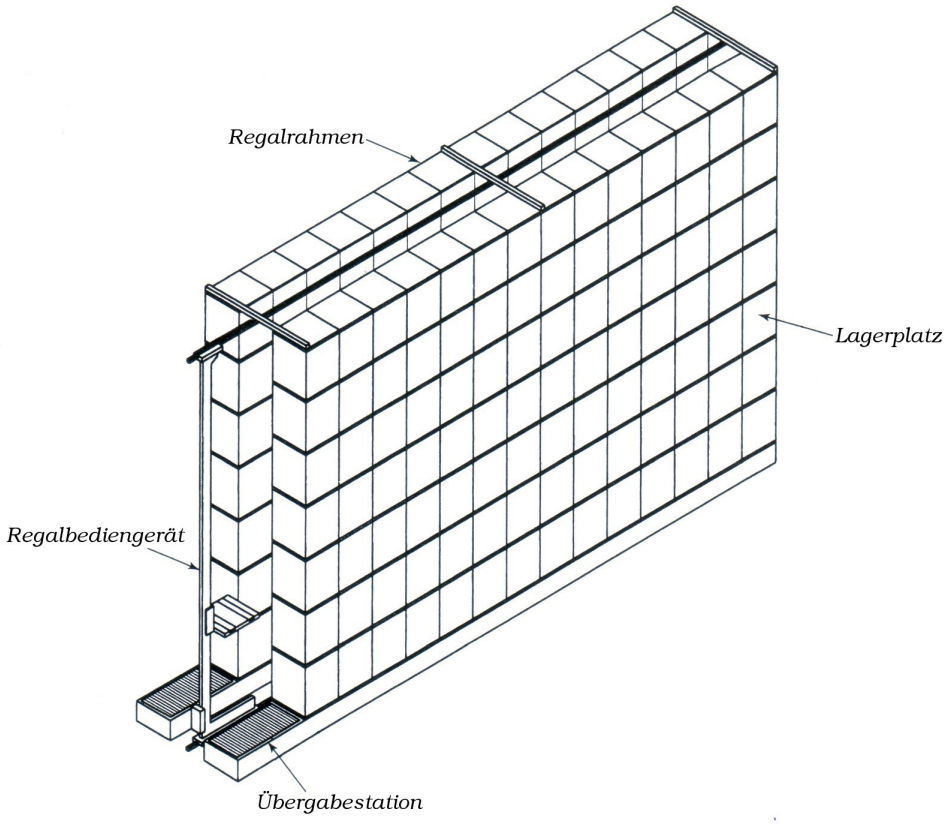


Abbildung 2.1.: Hochregallager mit Regalbediengerät [6]

Grundsätzlich kann man die verschiedenen Ausführungen eines HRLs an Hand dreier Kriterien unterscheiden:

- Regallagerung
- Regalbediengerät
- Handling

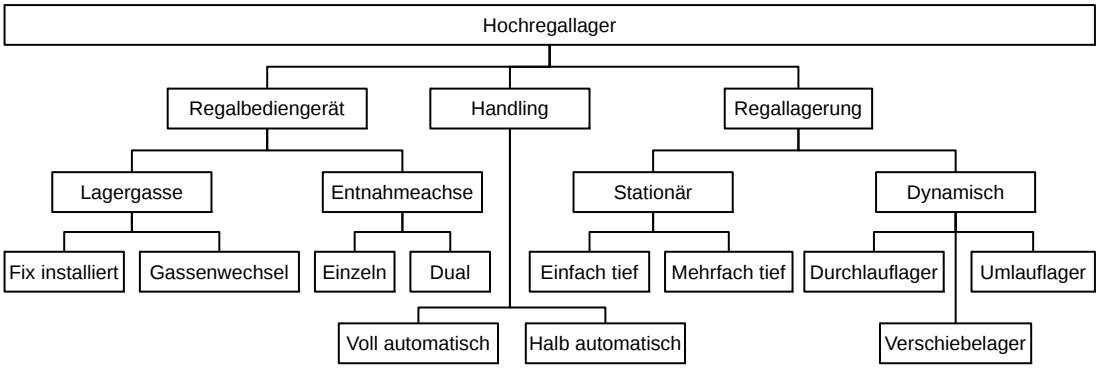


Abbildung 2.2.: Einteilung von Hochregallagern

Einen Überblick über die verschiedenen Ausführungsformen von HRL wird in Abbildung 2.2 gegeben.

Regallagerung

Grundsätzlich unterscheidet man bei der Regallagerung zwischen statischer und dynamischer Regallagerung. Bei der statischen Regallagerung werden die Ladeeinheiten auf ortsfesten Regalen einfach oder mehrfachtief eingelagert und bis zum Auslagervorgang nicht bewegt. Bei der dynamischen Regallagerung hingegen werden die Ladeeinheiten zwischen Ein- und Auslagervorgang bewegt. Dies kann entweder durch Eigenbewegung der Ladeeinheiten auf dem Regal (Durchlauflager) oder durch Verschiebung der gesamten Regalanlage (Verschiebelager, Umlauflager) erfolgen. [1]

Regalbediengerät

Regalbediengeräte sind schienengeführte Fördermittel zur Bedienung von Zeilenregalen. Durch den Einsatz von Regalbediengeräten konnte die Bauhöhe von Palettenregalen gesteigert und damit die Flächennutzung wesentlich verbessert werden. Das Regalbediengerät besteht aus ein bis zwei Masten, einem daran vertikal verfahrbaren Hubwagen, der ein oder mehrere Lastaufnahmemittel trägt, und dem Fahrwerk. Das Regalbediengerät ist boden- und deckenseitig geführt [17]. Die Antriebe werden mit Reibradantrieb oder Zahnriemenantrieb ausgeführt. Der Hubantrieb des RBGs erfolgt mittels umlaufendem Zugmittel wie Zahnriemen, Kette oder Seil. Die obere und untere Führung besteht aus Schienen, an denen sich die Rollen des RBGs abstützen. Die Antriebsdimensionierung richtet sich nach dem benötigten Momenten- und Geschwindigkeitsverlauf des repräsentativen Arbeitsspieles und wird mit rechnergestützten, starrkörperkinetischen, in anspruchsvollen Fällen mit elastomechanischen Auslegungsverfahren durchgeführt [1]. Je nach Umschlagleistung kann das RBG in jeder Lagergasse fix installiert sein oder mehrere Lagergassen werden von

einem RBG bewirtschaftet. Der Gassenwechsel kann entweder durch ein kurvengängiges RBG oder durch eine Umsetzerbrücke erfolgen [4].

Handling

Das Handling der zu lagernden Güter kann entweder voll oder halb automatisch erfolgen. Oft wird nicht der gesamte Inhalt einer Palette benötigt, sondern nur einzelne Artikeln. Um zu vermeiden, dass die gesamte Palette ausgelagert, zur Kommissionierungsstation gebracht und anschließend wieder eingelagert werden muss, gibt es Ausführungen bei denen sich eine Person an Bord des RBGs befindet. Die Kommissionierung kann dann direkt vor Ort erfolgen. Bei voll automatisierten Lösungen werden verschiedene Lastaufnahmemittel (LAM) eingesetzt um die zu lagernden Güter in das Regal einzulagern bzw. aus dem Regal auszulagern. Im Bereich des Palettenhandling wird die Last üblicherweise mittels Teleskopgabel in das Lagerfach gesetzt. Dadurch tritt keine Relativbewegung zwischen Fach und Ladeeinheit auf. Dieser Prozess erfordert allerdings durch sequentielle Bewegungsabläufe (Positionierung vor Lagerfach – Gabel ausfahren – Kurzhub des Hubwagens – Gabel einfahren) und damit verbundene Positionierzeiten relativ lange Lastübergabezeiten. Im Bereich der leichten Stückgüter ergeben sich dagegen vielfältige Gestaltungsformen der Lastübergabe, da unter anderem ein Ziehen der Einheiten unproblematisch möglich ist (entsprechend glatte und ebene Böden der Ladeeinheit und reibarme Regalaufnahmen vorausgesetzt).

2.1.2. Planung eines Hochregallagers

Bei der Planung eines HRLs ist es wichtig möglichst auf alle aktuellen und zukünftigen Anforderungen einzugehen. Bei der Auslegung muss darauf geachtet werden, dass im System sowohl keine Kapazitätsengpässe als auch Überkapazitäten auftreten. Dies ist besonders wichtig, da der Aufbau eines HRLs sehr unflexibel bezüglich mechanischen Veränderungen ist. Bei diesen Betrachtungen darf jedoch nie außer acht gelassen werden, dass das HRL nur eines von mehreren Systemen in einem Logistiksystem ist. Die Leistungsfähigkeit eines HRLs ist daher auch sehr von der Leistungsfähigkeit der anderen Systemen abhängig. Am meisten macht sich dies bemerkbar bei der Anbindung des HRLs an ein Transportsystem. Ist der Durchsatz dieser Anbindung niedrig, so hat dies Auswirkungen auf die Leistungsfähigkeit des HRLs.

Bei der Planung wird als erstes der Typ eines HRLs bestimmt. Dies ergibt sich aus den Anforderungen der einzulagernden Güter. Als zweites wird die Anzahl der Lagergassen bestimmt. Diese lassen sich aus der geforderten Umschlagleistung, den benötigten Lagerplätzen, den verfügbaren Platz, etc. ableiten. Das Produkt aus Anzahl der Lagergassen, Regalhöhe und Regallänge pro Gasse ist dabei konstant. [16]

2.1.3. Steuerung eines Hochregallagers

Genauso wichtig wie der mechanische Aufbau eines HRLs ist die Steuerungssoftware. Bei der Steuerung eines HRLs wird versucht unter Verwendung verschiedener Strategien die an das HRL gestellten Anforderungen hinsichtlich Verfügbarkeit, Umschlagleistung, etc. zu erfüllen. Aufgrund der großen Anzahl an Strategien werden im folgenden nur jene angeführt, welche auf die konkrete Umsetzung des Hochregallagers angewendet werden können. Eine genauere Auflistung der verschiedenen Strategien wird in [17] und [16] gegeben.

Belegungsstrategien

Die Zuweisung eines Lagerplatzes kann unter Berücksichtigung verschiedener Gesichtspunkte wie Verfügbarkeit, Umschlagleistung, Ausnutzung etc. erfolgen. Die wesentlichsten Belegungsstrategien seien hier kurz erklärt. Dabei lassen sich zum Teil mehrere Strategien gemeinsam verwenden.

Freie Lagerplatzvergabe: Bei dieser Strategie ist die Wahrscheinlichkeit für alle Lagerplätze gleich groß, dass ein Artikel eingelagert wird. Dadurch kann eine maximale Ausnutzung der Lagerkapazität erreicht werden.

Zonung: Bei der Zonung werden verschiedene Artikel gemäß ihrer Umschlaghäufigkeit in verschiedenen Zonen gelagert. Dadurch werden die Fahrzeiten minimiert und die Umschlagleistung erhöht.

Nähester freier Lagerplatz: Der am nächstliegendste freie Lagerplatz wird verwendet um den Artikel einzulagern. Dadurch beginnt sich das Lager rund um den Ein-/Auslagerungspunkt zu füllen.

Querverteilung: Durch Querverteilung wird versucht die Verfügbarkeit eines Artikel zu erhöhen. Dazu werden mehrere Stück des gleichen Artikels über mehrere Lagergassen verteilt. Dadurch ist auch bei Ausfall einer Lagergasse der Artikel noch verfügbar.

Bewegungsstrategien

Für die Ein- bzw. Auslagerung von Artikeln stehen zwei verschiedene Bewegungsabfolgen zur Verfügung, das *Einfachspiel* und das *Doppelspiel*. Beim *Einfachspiel* nimmt das RBG das einzulagernde Teil am Einlagerungsplatz auf und bringt es zum vorgegebenen Regalfach. Danach fährt es wieder zum Einlagerungsplatz zurück um ein neues Teil einzulagern. Beim *Doppelspiel* nimmt das RBG das einzulagernde Teil am Einlagerungsplatz auf und bringt es wiederum zum vorgegebenen Regalfach. Anstatt wieder zum Einlagerungsplatz zurück zu fahren, fährt das RBG zu einem anderen Regalfach um das dort eingelagerte Teil zum Auslagerungsplatz zu transportieren.

Auslagerungsstrategien

Die Durchführung der Auslagerung erfordert die Berücksichtigung verschiedenster Zielvorgaben und wird unter Anwendung bestimmter Auslagerungsstrategien durchgeführt. Einige relevante Strategien seien hier kurz erklärt.

FIFO (First-In- First-Out): Um die Überalterung und den Verfall einzelner Ladungseinheiten eines Artikels zu vermeiden, wird die zuerst eingelagerte Ladungseinheit (jene die sich am längsten im Lager befindet) ausgelagert.

Kürzester Fahrweg: Es wird jene Ladungseinheit eines Artikels ausgelagert, welche den kürzesten Anschlussweg besitzt. Dadurch können die Fahrwege der RBG minimiert und die Umschlagleistung erhöht werden.

Mengenanpassung: Je nach Auftragsmenge wird eine volle oder angebrochene Ladungseinheit des entsprechenden Artikels ausgelagert. Dadurch kann unter Umständen die Rückeinlagerung vermindert werden, was wiederum zu einer Erhöhung der Umschlagleistung führt.

Verweilpositionsstrategien

Die Verweilpositionsstrategien werden in der Fachliteratur auch oft unter dem englischen Fachbegriff ,,Dwell-point Rules" gefunden. Sie legen fest, wo das RBG auf den nächsten Auftrag warten soll. Je nach Anforderung kann so die Umschlagleistung des HRLs erhöht werden. Werden z.B. hauptsächlich nur Artikeln eingelagert, so macht es Sinn das RBG immer beim Einlagerungsplatz auf den nächsten Auftrag warten zu lassen. Werden gleichermaßen Artikel ein- wie auch ausgelagert, so kann das RBG in der Mitte des Hochregallagers warten, da so der Abstand zu allen Lagerplätzen gleich ist. Das RBG kann aber auch einfach nach Beendigung eines Arbeitsspiels beim letzten Lagerplatz stehen bleiben. Die drei Dwell-point-Rules sind hier nochmals namentlich aufgelistet:

- Input station
- Midpoint
- Last location

Reihung der Aufträge

Die Abarbeitungsreihenfolge von Ein- bzw. Auslagerungsaufträgen hat eine große Auswirkung auf die Umschlagleistung von HRL. Allein die Abarbeitung von Aufträgen in Form von Doppelspielen führt zu einer bis zu 30%igen Reduktion der Fahrzeit gegenüber Einzelspielen. Durch die intelligente Aneinanderreihung von Ein- und Auslagerungsaufträgen kann der Abstand zwischen dem Lagerplatz des einzulagernden Teils und dem Lagerplatz des auszulagernden Teils bei einem Doppelspiel gering gehalten werden. Dies verkürzt die Fahrzeit und steigert somit den Durchsatz. [18]

Wie bereits erwähnt, ist das HRL nur eines von mehreren Systemen eines Logistiksystems. Um die optimale Umschlagleistung erzielen zu können, müssen zu meist übergeordnete Maßnahmen getroffen werden. Diese sollen anhand eines Beispiels erläutert werden. Ein LKW bringt Güter für ein Warenhaus am Morgen und ein anderer LKW holt Güter vom Warenhaus am Abend ab. Die Ein- und Auslagerung kann jeweils nur mit einem Einzelspiel erfolgen. Überlappen sich jedoch die Zeiten in denen die beiden LKW am Warenhaus sind, so kann die Ein- und Auslagerung in Form von Doppelspielen abgearbeitet werden. Dieses Beispiel verdeutlicht auf einfache Weise, dass das HRL zu jedem Zeitpunkt optimal arbeiten kann, jedoch der Durchsatz des Gesamtsystems unterschiedlich ist. [16]

Eine Lösung für ein solches Auftragsvergabeproblem wurde in [9] mit Hilfe eines Agenten basierenden Ansatzes für ein Warenhaus mit 16 separaten Lagerbereichen gezeigt. Das implementierte Agentenframework besitzt sowohl eine heterarchische wie auch hierarchische Struktur. Dadurch ist es möglich neben den lokalen Zielen eines jeden einzelnen Agenten auch globale Ziele vorzugeben. Die entworfene Struktur besteht aus drei Ebenen. Die oberste Ebene bereitet die Aufträge vor, die in einem Zyklus abgearbeitet werden sollen und weist die Lagerbereiche zu, aus denen die benötigten Artikel ausgelagert werden sollen. Dies geschieht unter Einhaltung der globalen Ziele. Bei der Abarbeitung der Aufträge überprüft die unterste Ebene die Zuweisung der Lagerbereiche. Dazu wird für jeden Auslagerungsantrag eine Anfrage erstellt. Jeder Lagerbereich gibt für die Anfrage ein Gebot ab. Ist das beste Gebot von einem anderen Lagerbereich als von jenem Lagerbereich, der von der obersten Ebene zugewiesen wurde, kann die unterste Ebene eine Änderungsanfrage an die Mittelebene stellen. Diese hat eine bessere Übersicht über das System und kann so die Entscheidung treffen. Die Evaluierung dieser Lösung für ein Auftragsvergabeproblem wurde durch Simulation mit einer kommerziellen Simulationssoftware durchgeführt.

2.2. CAN

Das CAN (Controller Area Network) Protokoll wurde ursprünglich für den Einsatz in Kraftfahrzeugen entwickelt. Heute wird CAN auch in mobilen Systemen, als maschinen- oder anlageninternes Kommunikationssystem, im Feldbereich der Fertigungsautomatisierung, in der Gebäudeleittechnik und in vielen anderen Bereichen eingesetzt. Dies ist nicht zuletzt auf die geringen Kosten zurückzuführen, die die hohen Stückzahlen aus der Automobilindustrie mit sich brachten. Wie praktisch alle Feldbusse setzt auch CAN auf das ISO/OSI-Modell. Dabei besteht CAN nur aus den ersten beiden OSI-Schichten und wurde selbst als ISO 11898-1 international standardisiert. Die ,,fehlende" Anwenderschicht wird durch die auf CAN aufsetzenden Protokolle wie DeviceNet, CAL und die CANopen Profile definiert.

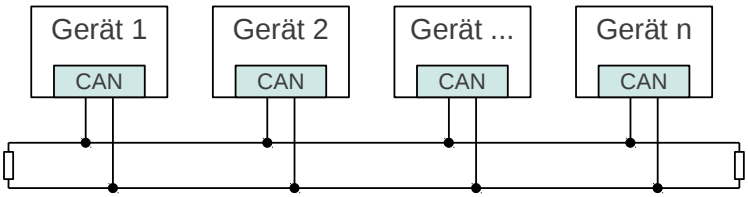


Abbildung 2.3.: Struktur eines CAN-Netzwerks

2.2.1. Topologie

Der CAN-Bus ist ein serielles Bussystem, bei dem alle angeschlossenen Stationen gleichberechtigt sind, d.h. jeder Busteilnehmer (Knoten) kann unabhängig von anderen Busteilnehmern senden und empfangen. Abbildung 2.3 zeigt die Struktur eines CAN-Netzwerks. Durch die Linienstruktur des Netzwerks bleiben bei Ausfall einer Station alle anderen Stationen weiterhin erreichbar. Als Übertragungsmedium wird hauptsächlich eine verdrillte Zweidrahtleitung eingesetzt. Durch die lineare Struktur des Busses können neue Teilnehmer in das System integriert werden bzw. bestehende entfernt werden, ohne dass die Verdrahtung geändert werden muss. Die beiden Leitungen werden in der Regel mit CAN-Low und CAN-High bezeichnet.

2.2.2. Datenübertragung

Ein Hauptmerkmal von CAN ist die objektorientierte Datenübertragung. Hierfür werden keine Busteilnehmer (Knoten) adressiert, sondern der zu übertragende Inhalt (z.B. Druck, Drehzahl, Position, etc.) wird durch einen netzweit festgelegten Identifier gekennzeichnet (entspricht einer Sender-/Quelladresse). Der Identifier eines Standard CAN Telegramms ist 11 Bit lang. Der Identifier bestimmt zusätzlich auch die Priorität einer Nachricht. Je niedriger der Identifier desto höher die Priorität. Eine Prioritätenvergabe ist für das verwendete Buszugriffsverfahren notwendig. Ein Vorteil der inhaltsbezogenen Adressierung ist, dass die übermittelten Daten gleichzeitig von mehreren Knoten empfangen werden können, ohne dass der Sender jeden Empfänger einzeln physikalisch adressieren muss. [10]

Möchte ein beliebiger Teilnehmer eine Nachricht versenden, übergibt dieser die Nachricht und den Identifier an den CAN-Controller. Der CAN-Controller wartet bis der Bus frei ist und beginnt anschließend seine Nachricht am Bus abzusetzen. Anhand des Identifier können nun alle anderen CAN-Controller feststellen, ob die Nachricht für sie bestimmt ist. Um dies durchführen zu können, wird dem CAN-Controller bei der Initialisierung mitgeteilt, welche Nachrichten dem jeweiligen Teilnehmer zugeordnet sind. Ist die empfangene Nachricht nicht für die jeweiligen Teilnehmer bestimmt, verwirft der CAN-Controller diese sofort. Bei der Vergabe der Identifier ist es wichtig, dass jeder Identifier nur einmal im Netzwerk vorkommt.

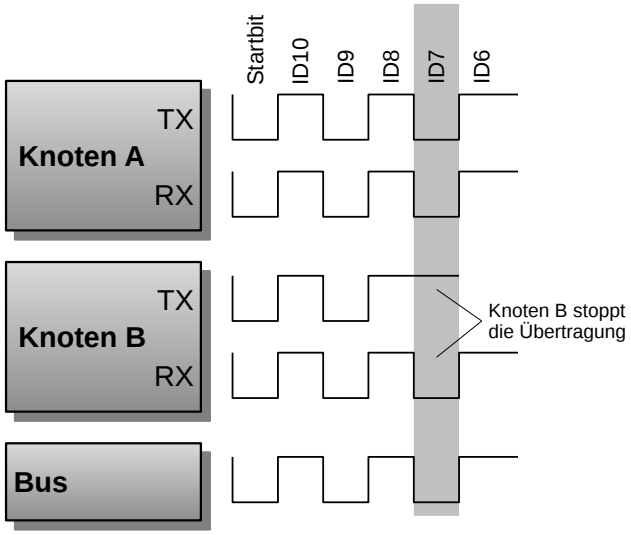


Abbildung 2.4.: CAN-Arbitrierung

2.2.3. Arbitrierung

Der CAN-Bus ist ein Netzwerk mit Multimaster Funktionalität. Alle Teilnehmer sind prinzipiell gleichberechtigt. Es ist von Seiten der Bus-Konzeption kein Frage-/ Antwortverhalten vorgesehen. Vielmehr soll jeder Teilnehmer seine Datenübertragung selbständig einleiten. Die Buszugriffssteuerung bei CAN geschieht mit Hilfe der zerstörungsfreien, bitweisen Arbitrierung. Zerstörungsfrei bedeutet dabei, dass der Gewinner der Arbitrierung - also die höher priore Botschaft - sein Telegramm nicht erneut von vorn beginnen muss [11]. Auf dem Bus werden die Pegel ,,aktiv" entsprechend einer ,,0" im CAN-Telegramm als dominant und ,,passiv" entsprechend einer ,,1" im CAN-Telegramm als rezessiv bezeichnet. Dabei überschreibt ein dominanter Pegel einen rezessiven Buszustand. Dies bedeutet, dass ein CAN-Knoten, der einen rezessiven Pegel auf den Bus senden will, von einem dominant sendenden Knoten überschrieben wird.

Wenn also zwei Stationen gleichzeitig erkennen, dass der Bus frei ist, geben sie nacheinander die Bits ihres Telegramms auf den Bus. Zuerst sendet jede Station ein Startbit, das immer dominant ist. Dann folgt die Adressinformation. Wenn innerhalb der Adressbits, die mit dem höchstwertigen Bit beginnend gesendet werden, eine ,,0" und eine ,,1" aufeinander treffen, setzt sich die dominante ,,0" auf dem Bus durch [15]. Jede Station hört auch gleichzeitig den Bus ab. Dadurch erkennt die Station, die eine rezessive ,,1" auf den Bus schreiben wollte, das der Bus eine dominante ,,0" aufweist. Die Station mit der nieder prioren Nachricht (höherer Identifier) bricht die eigene Übertragung ab und versucht es zu einem späteren Zeitpunkt erneut. Die höher priore Nachricht (Nachricht mit kleinstem Identifier) bleibt dadurch auf dem Bus fehlerfrei erhalten. Dieser Sachverhalt soll in Abbildung 2.4 nochmals verdeut-

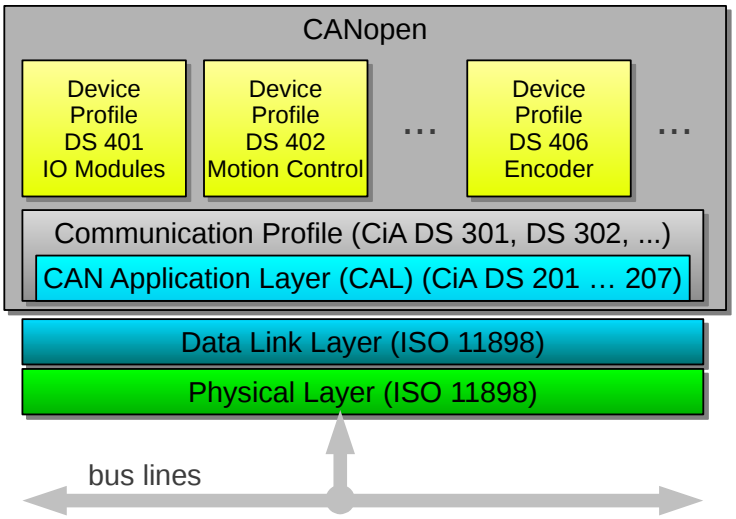


Abbildung 2.5.: Struktur von CANopen

licht werden. Der hier beschriebene Buszugriffsmechanismus wird als CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) bezeichnet.

2.3. CANopen

CANopen ist ein Kommunikationsstandard für CAN-Netzwerke und wurde von der Nutzerorganisation ,,CAN in Automation" (CiA) standardisiert. CANopen basiert auf einer Teilmenge des ,,CAN application layer" (CAL) und enthält zusätzlich einige Erweiterungen speziell für industrielle Echtzeitsysteme. Dabei hat CANopen im Gegensatz zu CAL nicht mehr den Anspruch auf weitgehende Universalität, sondern man hat den Horizont ganz bewusst etwas eingeschränkt um die Funktionen einfach halten zu können. Für die Voraussetzung der Austauschbarkeit von Geräten wurden deswegen bestimmte Funktionselemente und Attribute von Standardkomponenten im Systemmodell definiert. Zusätzlich wurden Mechanismen zur Synchronisation aller Geräte im Netz geschaffen. Damit können deterministische Übertragungszeiten gewährleistet werden, ohne die Fähigkeiten von CAN zur effizienten Übertragung von asynchronen Botschaften wesentlich einzuschränken. [11]

Abbildung 2.5 zeigt den grundsätzlichen Aufbau von CANopen in Anlehnung an das ISO/OSI-Referenzmodell. Die Schichten 3-6 werden dabei, wie auch bei CAN, nicht verwendet. Die Schicht 7 wird durch den vereinfachten CAL gebildet, welcher die verschiedenen Kommunikationsprofile definiert. Darauf aufgesetzt befinden sich die Geräteprofile, welche die Funktionselemente und Attribute in definierte Gruppen zusammenfassen. Dadurch sind alle Geräte, die konform zu den Geräteprofilen sind untereinander austauschbar.

Tabelle 2.1.: Allgemeiner Aufbau des CANopen Objektverzeichnis

Index (hex)	**Objekt**
0000	nicht benutzt
0001 - 001F	Statische Datentypen (nur für Referenzzwecke)
0020 - 003F	Komplexe Datentype (allen Geräten gemeinsam)
0040 - 005F	Herstellerspezifische Datentypen
0060 - 007F	Gerätespezifische statische Datentypen
0080 - 009F	Gerätespezifische komplexe Datentypen
0060 - 0FFF	reserviert
1000 - 1FFF	Kommunikationsprofildaten
2000 - 5FFF	Herstellerspezifische Profildaten
6000 - 9FFF	Standardisierte Profildaten
A000 - FFFF	reserviert

2.3.1. Device Profiles - Geräteprofile

CANopen beschreibt die Eigenschaften von Geräten in sogenannten Geräteprofilen. Dabei stellt ein Geräteprofil eine Sammlung von bestimmten Funktionalitäten und Eigenschaften dar. Ein solches Geräteprofil besteht dabei aus drei Teilen:

- Grundfunktionen: beschreibt alle Funktionen, die unbedingt vorhanden sein müssen, um konform mit dem jeweiligen Profil zu sein
- optionale Funktionen: beschreibt Funktionen die nicht vorhanden sein müssen, aber falls sie implementiert sind, dann so wie beschrieben
- Hersteller spezifische Funktionen: beschreibt alle individuellen Funktionen und Erweiterungen

Die verschiedenen Geräteprofile sind in den Standards DS 40x beschrieben.

2.3.2. Object Dictionary - Objektverzeichnis

Die zuvor angesprochenen Funktionalitäten und Eigenschaften, werden in CANopen als Objekte bezeichnet. Das Objektverzeichnis ist nun die Zusammenstellung aller Funktionalitäten und Eigenschaften eines CANopen Geräts. Je nach Funktionalität bzw. Eigenschaft werden mehrere Objekte in Gruppen zusammengefasst. Der Zugriff auf die einzelnen Objekte erfolgt über einen sogenannten ,,Index" und ,,Sub Index". Tabelle 2.1 zeigt den prinzipiellen Aufbau des Objektverzeichnisses.

2.3.3. Communication Profile - Kommunikationsprofil

Die Kommunikation in CANopen erfolgt in Form von Telegrammen, mit denen die Nutzdaten übertragen werden. Dabei unterscheidet man im wesentlichen zwischen

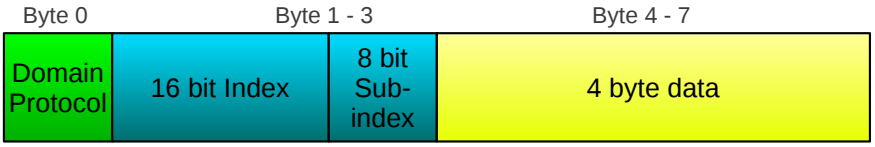


Abbildung 2.6.: Servicedatenobjekt SDO

Servicedatenobjekte und Prozessdatenobjekte. Zusätzlich werden Telegramme für das Netzwerk Management, Gerätesynchronisation und für die Fehlermeldungen definiert. CANopen verzichtet zu Gunsten des Netzwerkmanagements auf die Multimasterfähigkeit des CAN-Busses und führt einen CAN-Master ein, der die Aufgaben des Netzwerkmanagements übernimmt. Alle weiteren CAN-Knoten werden als sogenannte Slave-Baugruppen implementiert. Für das Netzwerkmanagement und Fehlermeldungen gibt es vordefinierte logische Kommunikationskanäle wie

- Kommunikationsobjekte für den Boot-Up (das Hochfahren des Netzes), Starten, Anhalten, Zurücksetzen eines Knotens usw.,
- Kommunikationsobjekte für die dynamische Identifier-Verteilung
- Kommunikationsobjekte für das Nodeguarding und Lifeguarding – damit kann eine Überwachung des Netzwerks durchgeführt werden,
- ein Kommunikationsobjekt für die Synchronisation,
- Kommunikationsobjekte für Notfallmeldungen (Emergency).

Diese sind in CANopen fest definiert und besitzen einen globalen Nachrichtencharakter (Broadcast). [13]

2.3.3.1. Service Data Object (SDO) - Servicedatenobjekt

Mit Hilfe von SDOs kann auf alle Objekte des Objektverzeichnisses zugegriffen werden. In der Praxis werden SDOs meist zur Konfiguration des Gerätes verwendet. Innerhalb eines SDOs kann immer nur auf ein Objekt zugegriffen werden. SDOs werden grundsätzlich immer beantwortet und dürfen nur nacheinander gesendet werden. D.h. das nächste SDO darf erst gesendet werden, wenn das zuvor gesendete SDO beantwortet wurde.

Da die Objekte des Objektverzeichnisses unterschiedliche Datenfelder besitzen und mit Hilfe von SDOs auf die Objekte lesend wie auch schreibend zugegriffen werden kann, benötigt das Servicedaten-Telegramm eine Protokollstruktur. Abbildung 2.6 zeigt den Aufbau eines SDOs. Durch das ,,Domainprotokoll" (8-Bit) wird angegeben, ob ein lesender oder schreibender Zugriff erfolgt. Zusätzlich gibt das ,,Domainprotokoll" die Länge der Nutzdaten im Falle eines schreibenden Zugriffs an.

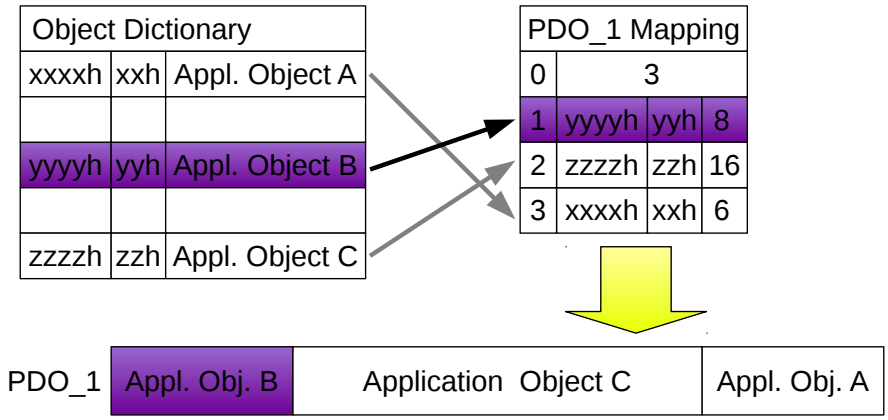


Abbildung 2.7.: PDO Mapping

2.3.3.2. Process Data Object (PDO) - Prozessdatenobjekt

PDOs dienen der Übertragung von aktuellen Prozesszuständen. Dabei können mit PDOs die Nutzdaten mehrerer Objekte auf einmal übertragen werden. Man sagt, die Objekte sind in ein PDO gemappt. Abbildung 2.7 zeigt das Mapping dreier Objekte in ein PDO. Die Objekte, die mit Hilfe eines PDOs übertragen werden sollen, können mittels SDOs eingestellt werden. Die Länge der aneinandergereihten Nutzdaten darf dabei maximal 8 Byte betragen. Die Übertragung des PDO-Telegramms erfolgt ohne zusätzlichen Protokoll-Overhead. D.h. dem Empfänger muss die Länge und die Reihenfolge der einzelnen Objekte bekannt sein. Die Übertragung von PDOs kann zyklisch ober bei Änderung der zu übertragenen Objekte erfolgen.

Bei PDOs übernimmt der Slave die Rolle des Servers und der Master die Rolle des Client. Dadurch wird auch ein Transmit PDO vom Slave gesendet und vom Master empfangen bzw. ein Receive PDO vom Master gesendet und vom Slave empfangen.

2.4. Kommunikationsfunktionsbausteine in IEC 61499

Der Standard IEC 61499 definiert eine offene Architektur für verteilte Automatisierungssysteme. Wesentliche Begriffe zur Beschreibung verteilter Automatisierungssysteme sind System, Gerät, Ressource, Anwendung und Funktionsbaustein (FB). Ein System kann aus einer oder mehreren Anwendungen bestehen. Dabei können sich die Anwendungen über mehrere Ressourcen verteilen. Die Ressourcen selbst können sich wiederum auf verschiedenen Geräten befinden. Wenn eine Anwendung auf mehrere Geräte verteilt ist, werden sogenannte Kommunikationsfunktionsbausteine benötigt um die einzelnen Funktionsbausteine über die Gerätegrenzen hinweg zu verbinden. Mit Hilfe der Kommunikationsfunktionsbausteine können auf die Kommunikationsschnittstellen des jeweiligen Gerätes zugegriffen werden.

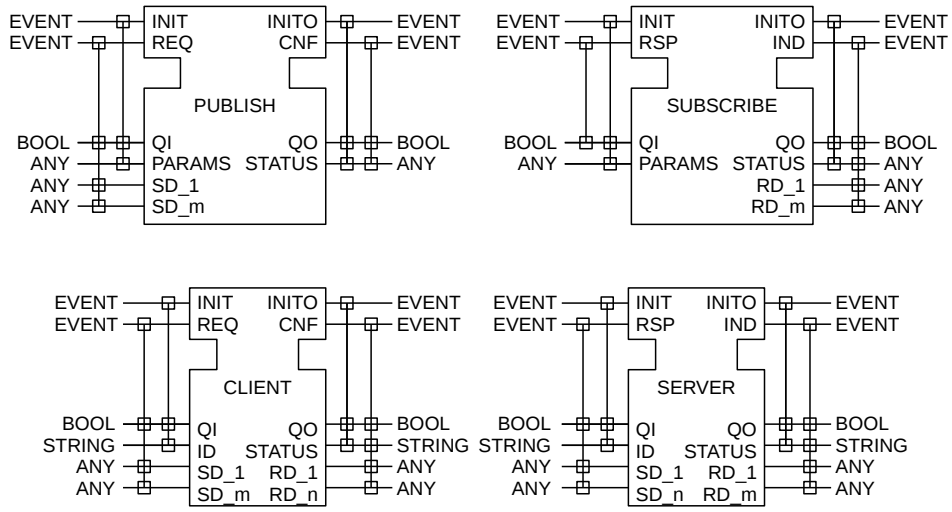


Abbildung 2.8.: Interface der Kommunikationsfunktionsblöcke laut IEC 61499-1

Tabelle 2.2.: Set an Statusmeldungen

Value	**Corresponds to**	**Semantics**
"OK"	INIT+, CNF+, IND+	Valid operation
"INVALID_ID"	INITO-	Invalid ID input
"TERMINATED"	INITO-	Service termination
"INVALID_OBJECT"	CNF-, IND-	Invalid object type received or requested to be sent
"DATA_TYPE_ERROR"	CNF-, IND-	Received object type does not match actual output type
"INHIBITED"	CNF-	Caused by REQ-
"NO_SOCKET"	CNF-	No socket available to serve REQ+
Other values	INITO-, CNF-, IND-	Socket service errors

Kommunikationsfunktionsbausteine gehören zur Klasse der Dienstschnittstellen-Funktionsbausteine. Grundsätzlich unterscheidet man zwei Typen von Kommunikationsfunktionsbausteinen: Client/Server FB für bidirektionale und Publish/Subsriber FB für unidirektionale Datenübertragung. Abbildung 2.8 zeigt das Interface der verschiedenen Kommunikationsfunktionsbausteine. [8]

Im Standard selbst werden keine einheitlichen STATUS-Werte definiert. In [2] ist jedoch ein Set an Statusmeldungen beschrieben, welches bei der Umsetzung der Ethernet Komm.-FB verwendet wurde. Dieses Set an Statusmeldungen ist in Tabelle 2.2 ersichtlich.

In [12] wird die Verwendung von EtherNet/IP (Industrie Protocol) mit IEC 61499 Komm.-FBs beschrieben. Da IEC 61499 Geräte in verschiedenster Art und Weise in industriellen automatisierten Systemen eingesetzt werden können, wurden verschiedene Anwendungsfälle untersucht. Dabei wurden drei Szenarien identifiziert.

Szenario 1: Smart Field Device: In diesem Szenario ersetzt das IEC 61499 Gerät ein I/O-Modul und muss aufbereitete Daten an die Steuerung weitergeben. Die Steuerung übernimmt dabei die Rolle des Masters und das IEC 61499 Gerät die des Slaves.

Szenario 2: Reuse of Legacy I/O-Modules: In diesem Szenario wird eine Steuerung durch ein IEC 61499 Gerät ersetzt und muss mit bestehenden I/O-Modulen kommunizieren. Das IEC 61499 Geräte übernimmt dabei die Rolle des Masters. Die I/O-Module sollen dabei keinen Unterschied zwischen Steuerung und IEC 61499 Gerät bemerken.

Szenario 3: Peer to Peer Communication: In diesem Fall besteht das System nur aus IEC 61499 Geräten. Dabei gibt es keine Master/Slave Struktur, da alle Geräte gleich berechtigt sind.

Die drei Szenarien lassen sich in zwei Übertragungsarten einteilen. In Szenario 1 und 2 wird eine bidirektionale Datenübertragung verwendet. Dabei übernimmt das IEC 61499 Gerät einmal die Rolle des Servers (Slave) und einmal des Client (Master). In Szenario 3 wird eine unidirektionale Datenübertragung verwendet, da alle Geräte gleiche Rechte besitzen. Es gibt also keinen Master/Slave. Aus diesem Grund ist das Publish/Subscriber Modell hier besser geeignet.

2.5. Zusammenfassung

In diesem Kapitel wurde einerseits ein Überblick über CAN, CANopen und Kommunikationsfunktionsblöcke in IEC 61499 gegeben. Auf Basis dieser Grundlagen kann dann in weiterer Folge das notwendige Kommunikationsinterface entwickelt werden, mit dessen Hilfe die verwendeten Motorcontroller angesprochen werden können. Andererseits wurde der allgemeine Aufbau eines HRLs erklärt und eine Übersicht über die verschiedenen Typen gegeben. Im weiteren wurden einige Aspekte vorgestellt, die bei der Planung berücksichtigt werden müssen. Es wurde ein Überblick über die verschiedenen Strategien gegeben, welche von moderne HRL-Steuerungen unterstützt werden. Ein verteiltes Auftragsvergabeproblem für ein konkretes Warenhaus mit 16 separaten Lagerbereichen wurde vorgestellt und ein möglicher Agenten basierender Lösungsansatz erläutert. Die Evaluierung des Lösungsansatz erfolgte, wie bei sehr vielen Arbeiten im Bereich verteilte Systeme, durch Simulation. Aus diesem Grund soll in weiterer Folge ein Hochregallagermodell und die notwendige Steuerungssoftware entwickelt werden, um eben solche Problemstellungen auch auf Testanlagen evaluieren zu können. Die Steuerungssoftware stellt die notwendigen Basisfunktionen einer übergeordneten Steuerungsebene (z.B. einem Agenten) zur Verfügung.

Bei der Entwicklung der Steuerungssoftware stellt sich nun die Frage, welche Aufgaben von der Steuerungssoftware übernommen werden können, um bestmöglich die übergeordneten Steuerungsebenen entlasten zu können. Aus diesem Grund wird in weiter Folge eine objektorientierte Analyse einer HRL-Steuerung durchgeführt um die notwendigen Funktionen identifizieren zu können.

3. Aufbau des Hochregallagers

Dieses Kapitel beschäftigt sich mit der Umsetzung des HRLs. Dazu wird in einem ersten Schritt die bestehende Infrastruktur aufgenommen. Anschließend werden Anforderungen an das mechanische Design festgelegt. In einem nächsten Schritt wird das entwickelte mechanische Design vorgestellt und die einzelnen Komponenten des HRLs erklärt. Abschließend wird auf die Sicherheit der Anlage eingegangen. Dazu werden die verschiedenen Sicherheitsmechanismen vorgestellt, die einen sicheren Betrieb der Anlage ermöglichen.

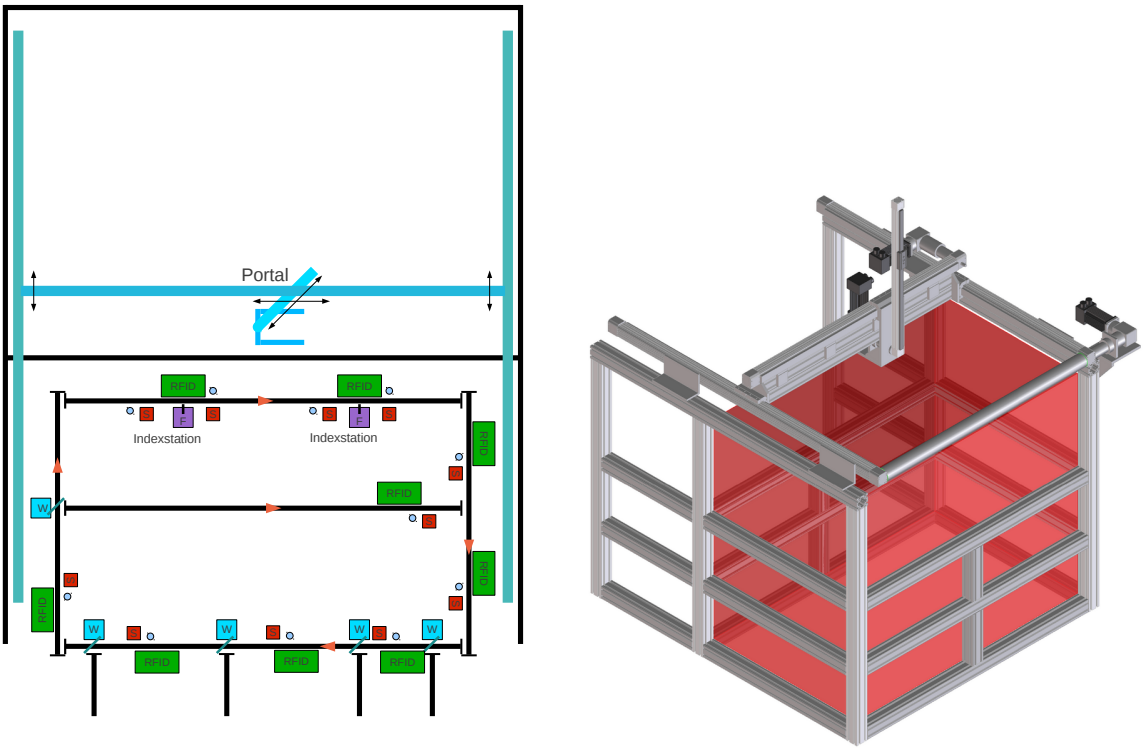
3.1. Bestandsaufnahme

Für die genaue Planung des HRLs wurde vom bestehenden Anlagenteil ein genaues 3D-CAD Modell für die mechanische Konstruktion angefertigt (siehe Abb. 3.1 (b)). Zusätzlich wurde ein Schaubild erstellt, welches die Topologie des Anlagenteils widerspiegelt (siehe Abb. 3.1 (a)). Im 3D-Modell wurde der verfügbare Raum für das HRL in rot eingezeichnet. Die Höhe des Lagers wird dabei vom Portal begrenzt, da dieses noch über den gesamten Lagerbereich fahren können soll.

3.2. Anforderungen an das mechanische Design

Die Anforderungen an das mechanische Design ergeben sich einerseits aus den Anforderungen der Institutsmitarbeiter, welche bereits im Zuge der Arbeit „Projektierung eines Hochregallagers" erhoben wurden. Diese werden hier nochmals der Vollständigkeit halber angeführt. Andererseits muss das mechanische Design die Kriterien erfüllen, die bei der Planung eines HRLs beachtet werden müssen. Diese sind in Abschnitt 2.1.2 beschrieben. Aus diesen beiden Punkten lassen sich nun folgende Anforderungen ableiten:

1. Einzulagernde Teile
    - Teile für Forschungszwecke (z.B.: für das Handling mit dem ABB-Roboter)
    - kleine Pneumatikzylinder
    - Schüttgutteile (Schrauben, Muttern, ...)



(a) (b)

Abbildung 3.1.: Bestehende Topologie

Auf Grund der verschiedenen Teile, die sich sowohl in Größe und Form unterscheiden, wurde versucht Lösungen zu finden, wie die oben genannten Teile am besten eingelagert werden können. Dabei stellte ein eigener Werkstückträger die beste Lösung dar. Je nach Anforderung können auf diesen ein oder mehrere Teile geordnet abgelegt werden. Auch ein Träger für Schüttgut (Behälter) ist so realisierbar. Auf Grund der bestehenden Transportpaletten (siehe Abbildung 3.2), welche eine Aufnahmefläche von 160 x 160 mm haben, würde sich die gleiche Fläche als Grundfläche des Werkstückträgers als optimal erweisen

2. Anzahl und Größe der Lagerplätze

   - 20-100 Lagerplätze
   - Grundfläche von 160x160mm mit unterschiedlichen Höhen

   Im Gespräch stellte sich heraus, dass den meisten Mitarbeitern die Anzahl der Lagerplätze nicht so wichtig ist, da es sich um eine Forschungsanlage handelt. Wichtiger ist es, dass alle Anlagenteile gut zugänglich sind, um im Fehlerfall eine einfache Wiederinstandsetzung durchführen zu können. Um die Komplexität der Anlage zu erhöhen, sollen die Lagerplätze in verschiedene Höhen eingeteilt werden. Dadurch lässt sich der verfügbare Platz optimal ausnutzen.

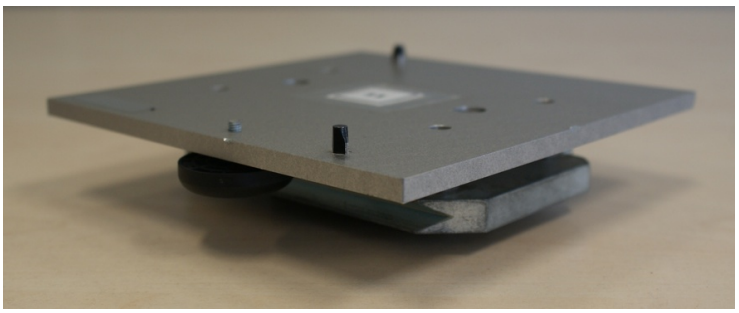


Abbildung 3.2.: Transportpalette mit einer Grundfläche von 160 x 160mm

3. Manuelle Ein- und Auslagerung

   Die manuelle Ein- und Auslagerung soll über ein eigens dafür vorgesehenes Terminal erfolgen. Das Terminal besteht im wesentlichen aus einer bestimmten Anzahl an Lagerplätzen, die sowohl für den Bediener als auch für das Regalsystem zugänglich sind. Um Verletzungen vorzubeugen ist es wichtig, dass eine maschinelle Entnahme bzw. Befüllung nur dann durchgeführt wird, wenn der Bediener momentan kein Teil befüllt bzw. entnimmt. Dies kann durch einen Lichtvorhang oder ähnliche Sicherheitseinrichtungen sichergestellt werden. Werden die Lagerplätze nun mit Teile für die Einlagerung befüllt, muss dem System mitgeteilt werden, wo sich welches Teil befindet. Dies kann auf zwei verschiedene Arten erfolgen:

   a) Die Lagerplätze des Terminals sind durchnummeriert. Der Bediener teilt dem Lagersystem nun mit welches Teil er in welchen Lagerplatz gegeben hat. So kann das Lagersystem eine eindeutige Zuordnung vornehmen.

   b) Jeder Werkstückträger muss mit einer maschinell lesbaren ID ausgestattet werden (RFID, Strichcode, ...). Der Bediener ordnet nun das einzulagernde Teil einem Werkstückträger zu. Aufgrund der Werkstückträger ID kann das System nun eine eindeutige Zuordnung vornehmen.

4. Direkte Anbindung an das PTS.

   Wie in Abschnitt 2.1.2 beschrieben, hängt die Leistungsfähigkeit des HRLs stark von der Leistungsfähigkeit der angrenzenden Systeme und deren Anbindung an das HRL ab. Ursprünglich sollte die Anbindung des HRLs an das PTS mittels Portal realisiert werden. Das Portal hätte in diesem Fall die Übergabe der Werkstückträger vom HRL zum PTS für alle drei Lagergassen übernommen. Dadurch würde jedoch einerseits ein Kapazitätsengpass und andererseits ein ,,Single Point of Failure" entstehen. Aus diesem Grund wird jede Lagergasse direkt an das PTS angebunden. Dies führt zwar dazu das die Konstruktion der Entnahmeachse komplexer wird, jedoch wird dadurch die Entstehung eines Kapazitätsengpasses und ,,Single Point of Failure" verhindert.

5. Weitere Optionen

   - Garagenfunktion

     Als Option wurde die Möglichkeit angesprochen leere Transportpaletten einzulagern. Dadurch kann die Anzahl der im Transportsystem verwendeten Paletten an die momentanen Erfordernisse angepasst werden. Je nachdem ob sich zu viele oder zu wenige Paletten im Transportsystem befinden, können diese ein- bzw. ausgelagert werden. Das Lager fungiert in diesem Fall als Garage.

   - Direkte Ein-/Auslagerung durch das Portal

     Durch geeignetes Design der einzelnen Regalbediengeräte kann das Portal in die einzelnen Lagergassen der jeweiligen Regalbediengeräte einfahren und direkt Teile ein- bzw. auslagern. Dabei ist jedoch darauf zu achten, dass es zu keiner Kollision zwischen Regalbediengerät und Portal kommt.

Die Funktion der manuellen Ein- und Auslagerung kann am besten mit dem Portal umgesetzt werden, da dieses einerseits das Ein-/Auslagerungsterminal, andererseits alle Lagergassen anfahren kann. Da zum Zeitpunkt der Umsetzung des HRLs das Portal nicht zur Verfügung stand, wurde diese Funktion für die weitere Konstruktion nicht berücksichtigt. Dieses kann später leicht integriert werden, wenn das Portal wieder zur Verfügung steht. Ausgehend von diesen Anforderungen konnte nun mit der genauen Konstruktion des Hochregallagers begonnen werden. Dabei stellte die Forderung der direkten Anbindung an das PTS die größte Herausforderung dar.

3.3. Mechanischer Aufbau

Das HRL besteht aus drei baugleichen Lagergassen. In jeder Lagergasse ist ein RBG fix installiert, das jeweils 36 Lagerplätze bedienen kann. Dadurch ergibt sich eine Gesamtanzahl von 108 Lagerplätzen. Die vorderen beiden Lagergassen sind dabei gleich orientiert. Die hintere Lagergasse ist gegenüber den anderen gespiegelt eingebaut. Durch diese Art des Einbaus lässt sich der verfügbare Platz optimal ausnutzen. Abbildung 3.3 zeigt das 3D Modell des gesamten Aufbaus des HRLs. Das bestehende Portal wurde der Übersichtlichkeit wegen weggelassen.

3.3.1. Änderungen am Palettentransportsystem

Um die Anforderung, das jede Lagergasse direkt an das PTS angebunden sein muss, erfüllen zu können, muss auch das bestehende PTS geändert werden. Die Anbindung des HRLs an das PTS erfolgt über eine Indexstation, die vom RBG angefahren werden kann. Da im bestehenden Anlagenteil des PTSs nur zwei Indexstationen vorhanden sind (siehe Abbildung 3.1), muss eine weitere Indexstation hinzugefügt

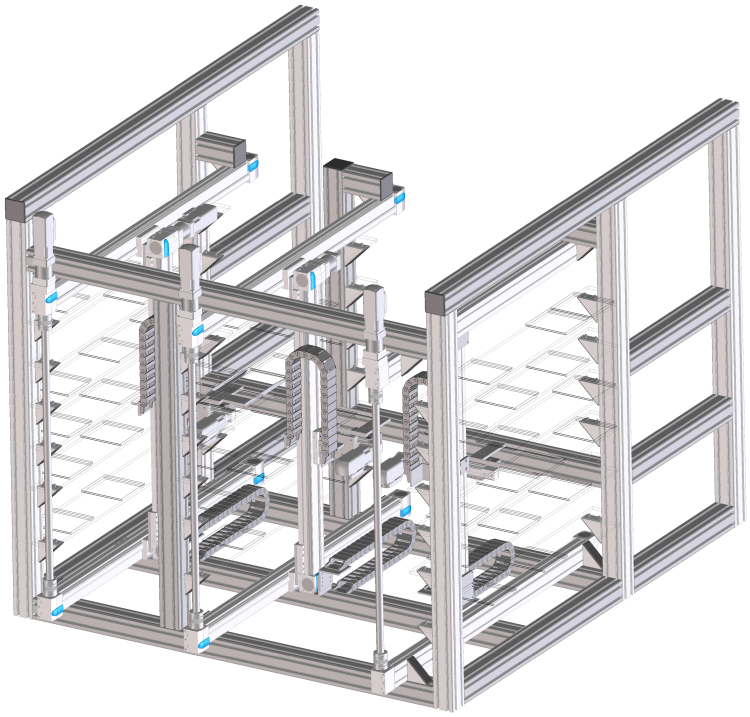


Abbildung 3.3.: Aufbau des Hochregallagers ohne Portal

werden. Die Anordnung der Indexstationen erfolgt so, dass diese vom RBG in der Nullposition der z-Achse erreicht werden können. Der strukturelle Aufbau ist in Abbildung 3.4 ersichtlich.

Vergleicht man die drei Indexstationen in Abbildung 3.4 so stellt man fest, dass die mittlere einen zusätzlichen Stopper besitzt. Dieser wird für die Garagenfunktion, also die Einlagerung der Transportpalette, benötigt. Die Indexstationen sind so aufgebaut, dass sie die Transportpalette fixieren. Dadurch kann gewährleistet werden, dass sich die Transportpalette immer an der gleichen definierten Stelle befindet. Zusätzlich wird verhindert, dass die Palette vom Förderband gehoben werden kann. Dies ist vor allem beim Einlagern von Werkstückträgern sinnvoll. Verklemmt sich z.B. ein Werkstückträger auf der Transportpalette, so würde die Transportpalette mit heraus gehoben werden. Diese könnte dann während der Bewegung des RBGs abstürzen oder beim Einfahren in das Regal dieses oder das RBG beschädigen. Aus diesem Grund wird eine weitere definierte Position benötigt, an der die Transportpalette vom Förderband abgehoben werden kann. Da die Länge des Förderbandes auf dem sich die Indexstationen befinden zu kurz ist, kann die Garagenfunktion nur bei der mittleren Lagergasse umgesetzt werden.

Analysiert man die Fahrwege der einzelnen Transportpaletten der in Abbildung 3.4 gezeigten Struktur des PTS, so stellt man fest, dass durch die Anordnung der Indexstationen sich die einzelnen Transportpaletten behindern. Dies geht sogar soweit,

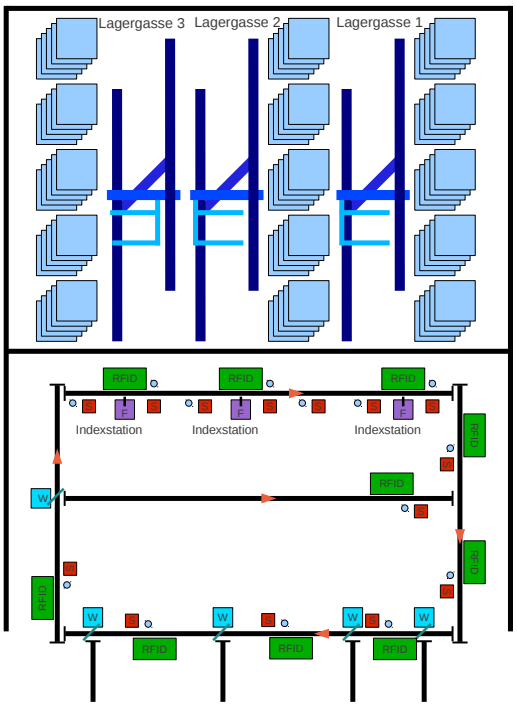
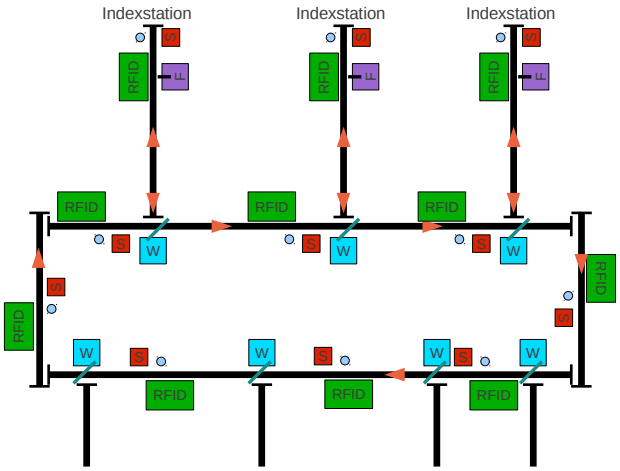


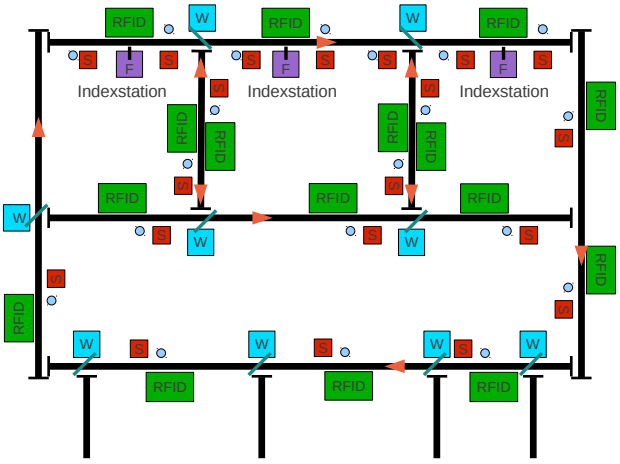
Abbildung 3.4.: Struktureller Aufbau

dass bei Ausfall einer Indexstation, im schlechtesten Fall alle anderen Indexstationen nicht mehr verwendet werden können. Aus diesem Grund muss einen Lösung gefunden werden, um jede Indexstation von der anderen unabhängig zu machen, sodass sich die Ein- und Ausfahrenden Paletten nicht gegenseitig behindern. Nur so kann eine hohe Verfügbarkeit des Systems gewährleistet werden. Der mechanische Aufwand des Umbaus soll dabei so gering wie möglich gehalten werden.

Im folgenden werden nun zwei Lösungsansätze vorgestellt, die die Indexstationen von einander entkoppeln. Der erste Lösungsansatz ist in Abbildung 3.5(a) dargestellt. Dabei befindet sich jede Indexstation am Ende einer Sackgasse. Dadurch kann jede Indexstation separat angefahren werden. Zusätzlich bietet diese Lösung den Vorteil, dass das RBG die jeweilige Indexstation anfahren kann, ohne dafür die Staplergabeln schwenken zu müssen. Das RBG kann in den freien Zwischenraum zwischen den Indexstationen einfahren und den Werkstückträger wie bei einem Lagerplatz aufnehmen. Jedoch besitzt diese Lösung einen wesentlichen Nachteil. Das Ein- und Ausfahren der Transportpalette in die jeweilige Sackgasse führt zu einer Drehung der Transportpalette. Somit sitzt dann der RFID Tag auf der verkehrten Seite. Dadurch können die nachfolgenden Knoten den RFID Tag nicht mehr auslesen. Da auch die Fixiereinrichtung der Transportpalette nicht symmetrisch aufgebaut ist, müsste die Palette wieder gedreht werden bevor diese die nächste Indexstation anfährt. Aus diesen Gründen ist dieser Lösungsansatz nicht umsetzbar.



(a) Indexstation am Ende einer Sackgasse



(b) Indexstation zwischen zwei Förderbänder

Abbildung 3.5.: Lösungsansätze zur Entkopplung der Indexstationen

Der zweite Lösungsansatz ist in Abbildung 3.5(b) dargestellt. Dabei werden zwei Förderbänder zwischen den einzelnen Indexstationen hinzugefügt. Dadurch kann jede Indexstation separat angefahren werden. Die Förderbänder müssen jedoch die Drehrichtung ändern können um den Anforderungen gerecht zu werden. Die Drehrichtungsänderung darf jedoch nur erfolgen, wenn sich keine Transportpalette auf dem jeweiligen Förderband befindet. Ansonst tritt genau das selbe Problem wie bei Lösungsansatz 1 auf. Die zusätzlichen notwendigen Weichen können alle als einfache Weichen ausgeführt werden. Um jedoch das Ein- bzw. Auslagern durch das RBG zu ermöglichen, muss dieses die Staplergabeln schwenken können. Da das RBG diese Anforderung erfüllt, kann mit Hilfe von Lösungsansatz 2 die Entkopplung der Indexstationen durchgeführt werden. Die Topologie des geänderten Gesamtsystems ist in Abbildung 3.6 ersichtlich.

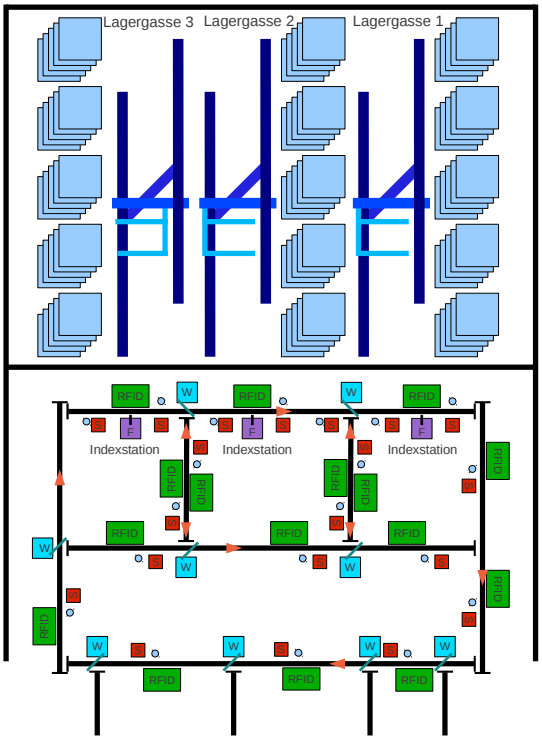


Abbildung 3.6.: Topologie des Gesamtsystems

3.3.2. Wahl der Komponenten

Da die Umsetzung des HRLs ausschließlich mit Komponenten der Firma Festo erfolgte, wurden zwei Festo Mitarbeiter eingeladen um mit ihnen den Aufbau des HRLs zu diskutieren. Dabei diente der Entwurf, der im Zuge der Vertiefung ,,Projektierung eines Hochregallagers" erstellt wurde, als Diskussionsgrundlage. Die Mitarbeiter von

Festo halfen bei der Wahl der richtigen Komponenten. Für die x- und y-Achse wurden Zahnriemenachsen des Typs EGC gewählt. Diese zeichnen sich durch eine hohe Steifigkeit aus. Ein weiterer Vorteil ist, dass die Motormontage an allen vier Seiten der Achse erfolgen kann. Dies kann gerade beim Aufbau eines Prototypen sehr hilfreich sein. Bei der z-Achse wird eine Zahnriemenachsen des Typs DGE eingesetzt. Die DGE-Achse besitzt im Gegensatz zur EGC-Achse ein besseres Verhältnis zwischen Arbeitshub und Gesamtlänge der Achse. Auf Grund der kompakten Abmaße der DGE-Achse ist es möglich drei RBG zu integrieren. Der Antrieb aller Achsen erfolgt mittels Servomotoren. Diese besitzen gegenüber Schrittmotoren eine höhere Dynamik. Zusätzlich kann die erreichbare Genauigkeit erhöht werden. Eine Liste der verwendeten Komponenten ist in Anhang A ersichtlich.

3.3.3. Regalbediengerät

Bei der Konstruktion wurde darauf geachtet das RBG möglichst kompakt zu bauen. Dadurch kann der verfügbare Raum optimal ausgenutzt werden. Bei der Auslegung der Achsen musste deswegen ein Kompromiss zwischen Nutzlast und Kompaktheit eingegangen werden. Der gesamte Aufbau des RBGs ist in Abbildung 3.7 dargestellt. Das RBG besteht aus vier Zahnriemenachsen. Die zwei horizontalen Achsen sind über eine starre Welle gekoppelt. Dadurch kann ein synchrones Fahren der beiden Auflagepunkte der Hubachse garantiert werden und somit ein Verspannen ausgeschlossen werden. Bei der Position des Antriebs wurde darauf geachtet, dass dieser nicht im Bewegungsraum des bestehenden Portals liegt um mögliche Kollisionen auszuschließen. Die Längen der Achsen wurden so gewählt, dass sich der verfügbare Bereich komplett abdecken lasst und ein direktes Anfahren der jeweiligen Indexstation möglich ist. Auf der Hubachse ist die Entnahmeachse montiert. Die Hubachse wird von einem Servomotor mit integrierter Bremse angetrieben. Nur so kann gewährleistet werden, dass die Entnahmeachse im stromlosen Zustand der Anlage nicht abstürzt.

3.3.4. Entnahmeachse

Mit Hilfe der Entnahmeachse können Werkstückträger an den jeweiligen Lagerpositionen bzw. bei der Indexstation aufgenommen bzw. abgelegt werden. Die Entnahmeachse besteht aus einer Zahnriemenachse, einem Drehantrieb und einer Staplergabel. Wie bereits in Abschnitt 3.2 beschrieben, muss jedes RBG direkt Paletten vom Transportsystem ein- bzw. auslagern können. Die Positionen der Lagerplätze sind zu der Position der Indexstation um 90 Grad verdreht. Das direkte Anfahren der Indexstation wird mit Hilfe des Drehantriebs ermöglicht. Da die Staplergabel nur zwischen zwei Positionen geschwenkt werden muss, kann ein pneumatischer Drehantrieb verwendet werden. In Abbildung 3.8 ist die Entnahmeachse des RBGs in der Endposition 2 abgebildet. Endposition 1 ist zum besseren Verständnis

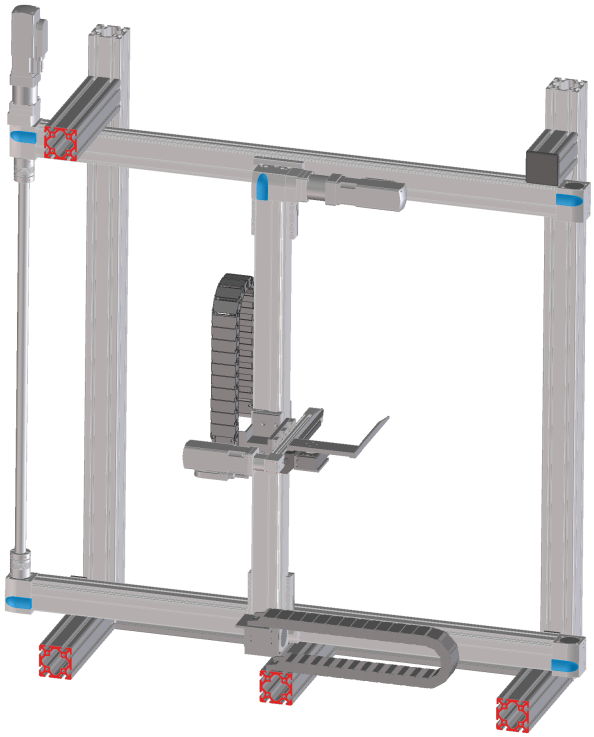


Abbildung 3.7.: Aufbau des Regalbediengeräts

in rot angedeutet. Zusätzlich sind die wichtigsten Komponenten und Merkmale in der Abbildung eingezeichnet. Die roten Pfeile zeigen die Bewegungsrichtung der Achse an. Die Drehung der Staplergabeln darf nur im eingefahrenen Zustand erfolgen. Wie man aus Abbildung 3.8 erkennt, sind die Staplergabeln an der Vorderseite leicht abgeschrägt. Dadurch können sich die Gabeln beim Einfahren in den Werkstückträger zentrieren.

3.3.5. Werkstückträger

Mit Hilfe des Werkstückträgers können je nach Anforderung ein oder mehrere Teile transportiert, ein- bzw. ausgelagert werden. Um die Werkstückträger von der Transportpalette nehmen bzw. Werkstückträger auf der Transportpalette ablegen zu können, muss der Werkstückträger eine Nut an der Unterseite der Auflagefläche besitzen. Dadurch entsteht ein Spalt zwischen Transportpalette und Werkstückträger. Dieser Sachverhalt ist in Abbildung 3.9 dargestellt. Der Spalt ermöglicht es nun mit den Staplergabeln zwischen Werkstückträger und Transportpalette einzufahren und den Träger von der Palette abzuheben bzw. nach dem Ablegen des Trägers auf der Palette herauszufahren. Um während des Einfahrens bzw. Herausfahrens den Träger nicht zu verschieben, wird dieser auf der Palette mit Zentrierbolzen gehalten.

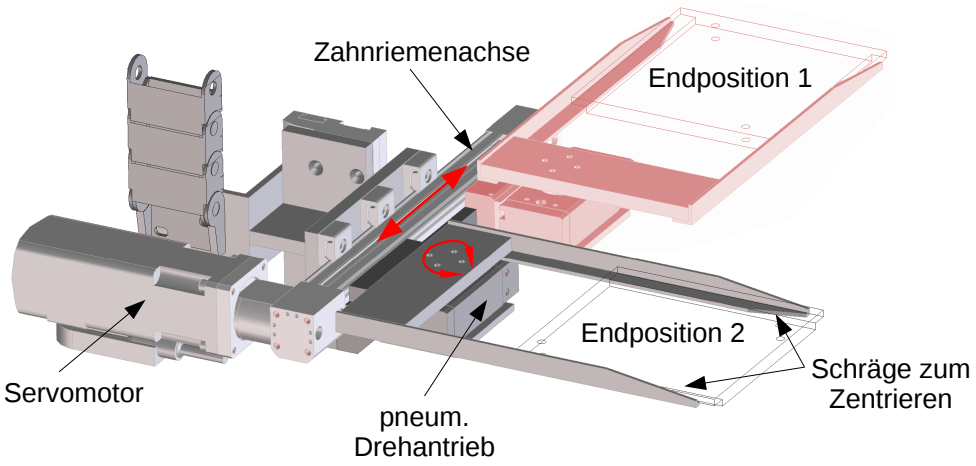


Abbildung 3.8.: Entnahmeachse des RBG

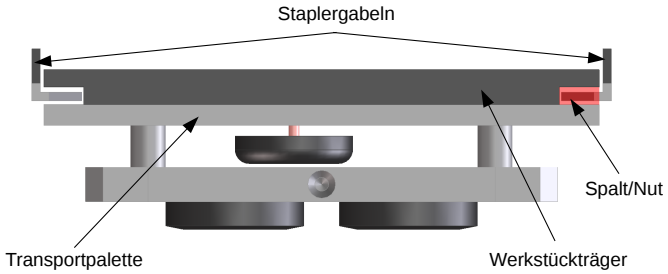


Abbildung 3.9.: Nut zwischen Werkstückträger und Transportpalette

3.3.6. Regal

Die Regalkonstruktion besteht aus sechs Lagerebenen von denen jede Lagerebene Platz für sechs Werkstückträger besitzt. Dadurch ergeben sich eine Gesamtanzahl von 36 Lagerplätze für ein Regal. Eine Lagerebene besteht aus einer Aluminiumplatte, auf der mit Hilfe von Zentrierbolzen die einzelnen Werkstückträger abgelegt werden können. Die Zentrierbolzen sind notwendig um sicherzustellen, dass sich die Träger immer an einer definierten Postion befinden, da das RBG kein Messsystem besitzt mit dem sich die genaue Position des Werkstückträgers bestimmen lässt. Diese Konstruktion ist sehr platzsparend in der Höhe und ermöglicht einen guten Zugang zu den einzelnen Lagerplätzen im Falle von Wartungsarbeiten. Die Positionen der Lagerplätze werden alle im Speicher der Steuerung hinterlegt und müssen dazu einmal eingelernt werden. Um das Einlernen zu ermöglichen muss die Steuerung das manuelle Verfahren des RBGs unterstützen.

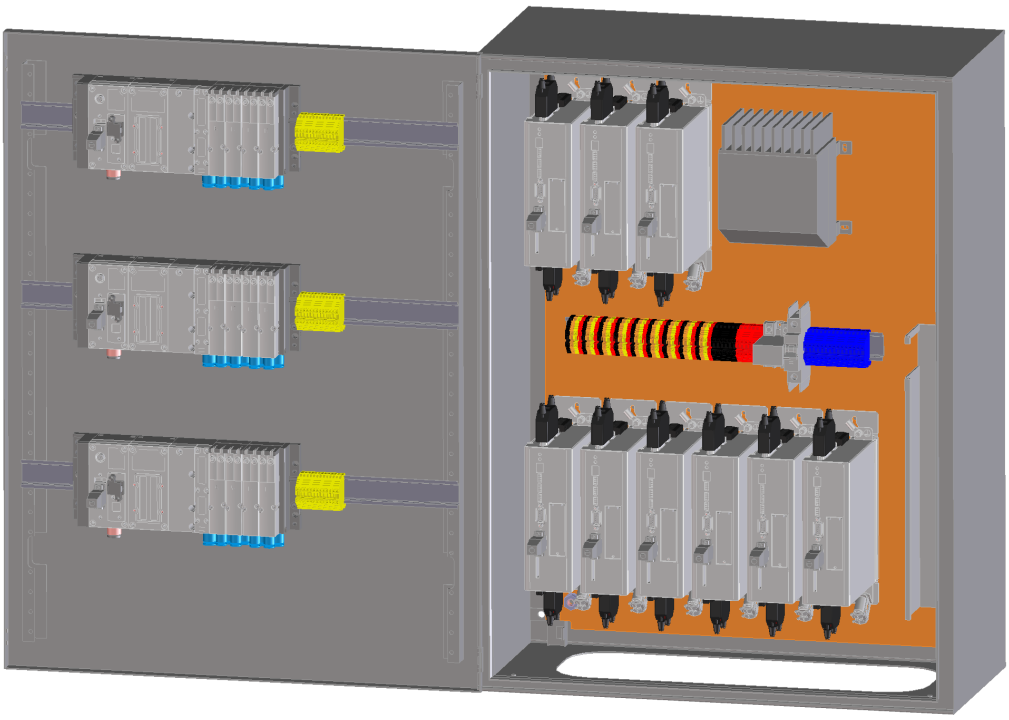


Abbildung 3.10.: Aufbau des Schaltschranks

3.3.7. Schaltschrank

Für den Einbau der elektronischen Komponenten (Motorcontroller, SPS, Netzteil) sollte der vorhandene Schaltschrank verwendet werden. Ziel war es alle notwendigen Komponenten für die Ansteuerung der drei Lagergassen in einen Schaltschrank einzubauen. Da der verfügbare Platz im Bezug auf die zu verbauenden Komponenten eher gering ist, wurde ein 3D Modell des Aufbaus erstellt. Mit Hilfe des Modells wurde versucht unter Berücksichtigung der Mindestabstände zwischen den Motorcontrollern und den Biegeradien der jeweiligen Leitungen eine optimale Aufteilung der Komponenten zu finden. Abbildung 3.10 zeigt den Aufbau des Schaltschranks bei geöffneter Schaltschranktür.

3.4. Sicherheit der Anlage

Dieser Abschnitt beschäftigt sich mit dem sicheren Betrieb des Hochregallagers. Dazu werden im folgenden die verschieden Sicherheitsmechanismen vorgestellt die einerseits den Bediener vor Verletzungen, andererseits die Anlage selbst vor Zerstörung schützen soll.

3.4.1. Schutzabdeckung

Um einen sicheren Betrieb der Anlage gewährleisten zu können, sollte das Hochregallager nach der Fertigstellung der beiden anderen Lagergassen durch Schutzwände gegen das unbefugte Betreten gesichert werden. Zusätzlich sollte die Möglichkeit vorgesehen werden, dass im Falle eines Fehlers oder für Wartungsarbeiten die einzelnen Lagerplätze leicht erreicht werden können. Dies kann z.B. durch zwei Türen an der Vorderseite des HRLs ermöglicht werden. Aufgrund des gespiegelten Aufbaus der dritten Lagergasse kann die zweite und dritte über eine Tür erreicht werden. Es ist jedoch darauf zu achten, das beim Öffnen einer Türe die Anlage abgeschaltet wird.

3.4.2. Notauskreis

Um in Falle eines Fehlers die Anlage abschalten zu können, wurde ein Notauskreis installiert. Dieser wird, nicht wie oft üblich durch Wegnahme der Lastspannung, sondern über den Reglerfreigabe-Eingang der Motorcontroller realisiert. Nach Wegnahme der Reglerfreigabe bremsen die Motorcontroller die Servomotoren gezielt ab. Im Falle der Hubachse des RBGs wird nach Erreichen der Drehzahl 0 die Haltebremse des Motors geschlossen. Anschließend werden alle Motoren stromlos geschaltet.

3.4.3. Überlastabschaltung

Die verwendeten Motorcontroller besitzen alle eine integrierte Überlastabschaltung. Kommt es zu einer Kollision zwischen RBG und anderen Anlagenteilen (z.B. Indexstation, Regalebene, etc.) so schaltet der Motorcontroller selbständig ab. Dadurch wird versucht einen Schaden der Anlageteile zu verhindern.

3.4.4. Endschalter

Alle verwendeten Achsen des RBGs sind mit Endschalter ausgestattet. Die Position der Endschalter wurde so gewählt, dass nur der notwendige Arbeitsbereich des RBGs zur Verfügung steht. Dadurch kann eine Kollision in Folge einer Fehlparametrierung bzw. Fehlfunktion verhindert werden.

3.5. Zusammenfassung

In einem ersten Schritt wurde die bestehende Infrastruktur analysiert. Zu diesem Zweck wurde einerseits ein Schaubild erstellt, welches die Topologie des bestehenden

Anlagenteils widerspiegelt. Andererseits wurde ein CAD Modell angefertigt, welches die Grundlage für die weitere Konstruktion des HRLs war. In einem nächsten Schritt wurden die Anforderungen an das HRL erhoben. Diese ergaben sich aus den Anforderungen der Institutsmitarbeiter und aus dem Stand der Technik. Auf Basis dieser Daten wurden dann die notwendigen Änderungen des bestehenden Systems erhoben und die Konstruktion des HRLs durchgeführt. Zum Schluss dieses Kapitels wurden die Sicherheitsfunktionen vorgestellt, welche einen sicheren Betrieb der Anlage ermöglichen.

4. Analyse des verteilten Steuerungskonzepts

Steuerungen für ein verteiltes Lagersystem sind in den meisten Fällen in zwei Ebenen aufgebaut. Die obere Ebene besitzt die Steuerungsintelligenz und ist oft in Form von Agenten umgesetzt. Die untere Ebene ist für die Ansteuerung der Aktuatoren zuständig. In weiterer Folge wird die obere Steuerungsebene als High Level Control (HLC) und die untere Steuerungsebene als Low Level Control (LLC) bezeichnet. Um die HLC zu entlasten wird nun versucht verschiedene Funktionen in die LLC auszulagern. Um eine robuste LLC entwickeln zu können, muss bereits im Vorfeld der gesamte Funktionsumfang der Steuerung definiert sein. Nur so kann entschieden werden, welche Funktionen in der LLC implementiert werden sollen.

Aus diesem Grund beschäftigt sich dieses Kapitel mit der Analyse eines verteilten Steuerungskonzepts eines HRLs. Die dabei identifizierten Anwendungsfälle dienen als Grundlage für die Entscheidung, welche Funktionen in der LLC umgesetzt werden können. In einem ersten Schritt wird das bestehende System analysiert, in welches das HRL integriert werden soll. Als nächster Schritt wird die Systemidee skizziert. Anschließend werden die Geschäftsprozesse des Lagersystems identifiziert. Um abschließend die Anwendungsfälle identifizieren zu können, wird der Prozessablauf ermittelt. Dieser wird im wesentlichen durch die Interaktion zwischen den am Lagerprozess beteiligten Komponenten beschrieben. Die identifizierten Anwendungsfälle und die daran beteiligten Akteure dienen als Basis für die Festlegung des Funktionsumfangs in der LLC.

4.1. Systemanalyse

Um das HRL bestmöglich in das bestehende System integrieren zu können, muss das Gesamtsystem einerseits auf der Materialflussebene und andererseits auf der Steuerungsebene betrachtet werden. Die Materialflussebene gibt die physikalischen Zusammenhänge wieder, da der Material-/Warenaustausch nur zwischen benachbarten Knoten stattfinden kann. Aus diesem Grund wurde eine Abstraktion des mechanischen Aufbaus des Gesamtsystems (vergleiche Abbildung 3.6) durchgeführt, um diese Zusammenhänge besser ersichtlich zu machen. Dieser Sachverhalt ist in Abbildung 4.1 dargestellt. Hauptaugenmerk liegt auf der Anbindung des HRLs an das PTS. Die blauen Verbindungslinien signalisieren zwischen welchen Knoten

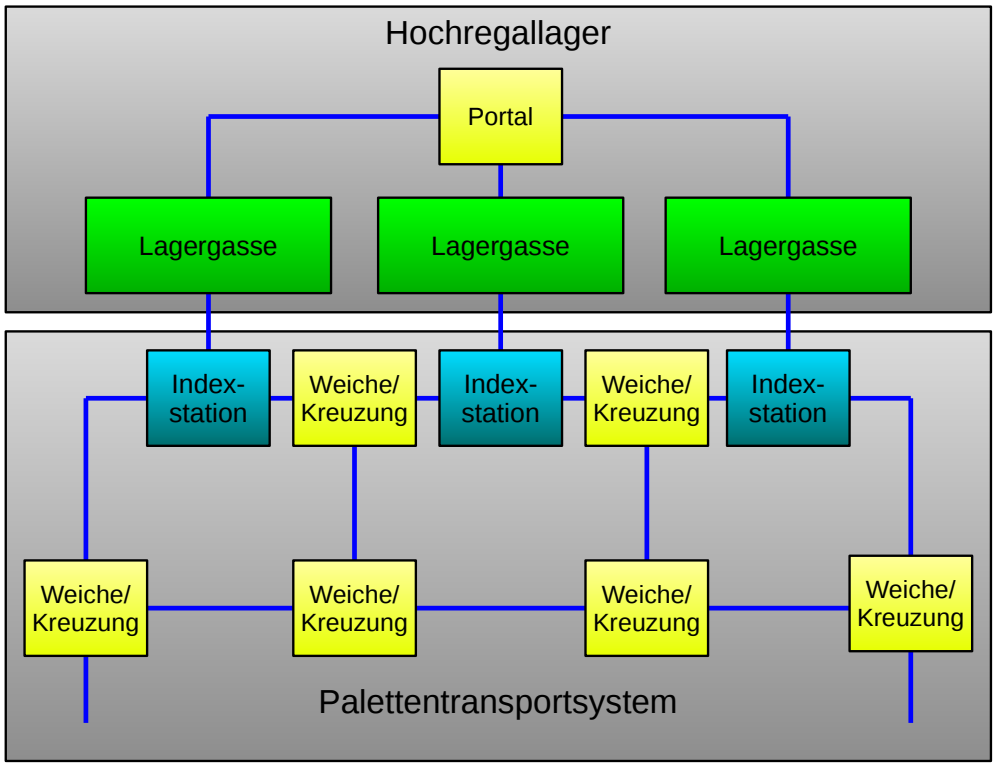


Abbildung 4.1.: Abstraktion des mechanischen Aufbaus

ein Material-/Warenaustausch stattfinden kann. Innerhalb des PTSs erfolgt der Materialfluss durch den Einsatz von Förderbändern. Der Material-/Warenaustausch zwischen PTS und HRL kann durch die spezielle Konstruktion des RBGs gewährleistet werden. Innerhalb des HRLs kann mit Hilfe des Portals ein Material-/Warenaustausch zwischen zwei Lagergassen stattfinden.

Während in der Materialflussebene nur physikalische Komponenten auftreten können, treten in der Steuerungsebene zusätzlich steuerungstechnische Komponenten auf, die nicht direkt einer Hardware zugeordnet werden können. Diese Komponenten sind z.B. für die Verwaltung oder für die Aufgabenvergabe zuständig. Ein weiterer wesentlicher Unterschied ist, dass im Gegensatz zur Materialflussebene auf der Steuerungsebene eine Interaktion auch zwischen zwei nicht benachbarten Knoten stattfinden kann. Unter Berücksichtigung dieser Faktoren kann nun ein Modell erstellt werden, das diesen Sachverhalt widerspiegelt. Dabei geht es jedoch weniger um die Vollständigkeit des Modells, sondern es soll viel mehr der strukturelle Aufbau und die Kommunikation zwischen den Komponenten dargestellt werden. Abbildung 4.2 zeigt dieses Modell für das Gesamtsystem. Die oberste Ebene bildet das Warenwirtschaftssystem bzw. Produktionsplanungssystem. Diese beiden Systeme sind für die Vorgabe der globalen und lokalen Ziele zuständig. Für die Umsetzung der lokalen bzw. globalen Ziele sind die Systeme der unteren Ebenen (HRL, PTS und die Pro-

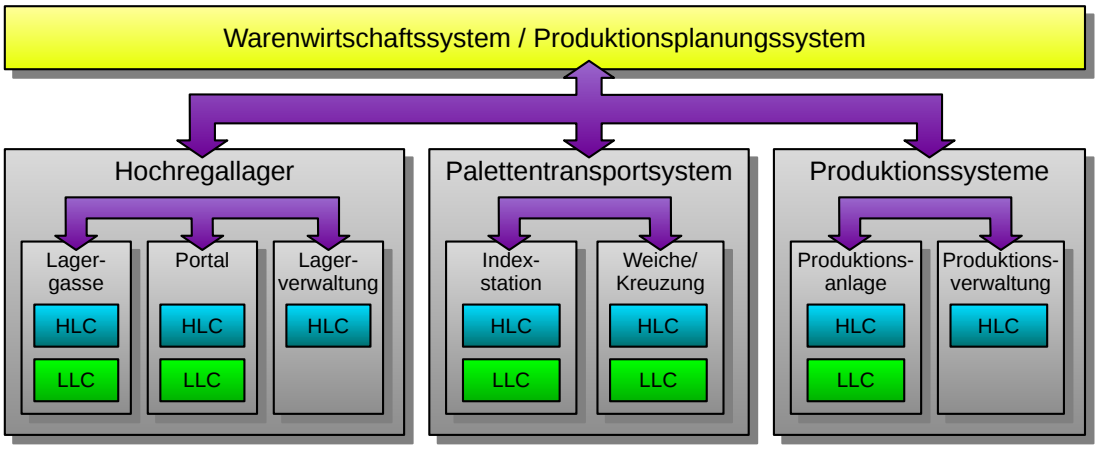


Abbildung 4.2.: Steuerungstechnischer Aufbau des Gesamtsystem

duktionssysteme) zuständig. Um die lokalen bzw. globalen Ziele erreichen zu können, muss einerseits eine Kommunikation innerhalb der jeweiligen Systeme stattfinden, andererseits müssen die Systeme untereinander kommunizieren. Wie man aus Abbildung 4.2 erkennt, sind die jeweiligen Subsysteme wiederum in zwei Ebenen geteilt.

- High Level Control (HLC)

  Die HLC besitzt die Steuerungsintelligenz und ist in Form eines Agenten umgesetzt. Sie kommuniziert mit anderen HLC um die globalen und lokalen Ziele umsetzen zu können. Je nach Aufgabengebiet kann der HLC keine, eine oder mehrere LLC zugeordnet sein.

- Low Level Control (LLC)

  Hauptaufgabe der LLC ist die Ansteuerung der Aktuatoren. Sie stellt verschiedene hardwarenahe Funktionen der HLC über ein Interface zur Verfügung.

Da nun das physikalische und steuerungstechnische Umfeld des HRLs bekannt ist, kann die Systemidee skizziert werden. Diese bildet die Basis für die in [14] beschriebene objektorientierte Analyse des verteilten Steuerungskonzepts.

4.2. Systemidee und Zielsetzung

Mit Hilfe des entwickelten Steuerungskonzepts soll das HRL an das verteilt gesteuerte Transportsystem angebunden werden. Das Steuerungskonzept des HRLs soll so aufgebaut werden, dass die Komponenten möglichst autonom arbeiten können. Das System soll Funktionen für das Ein- und Auslagern von Werkstückträgern und für die Lagerverwaltung zur Verfügung stellen.

Ziel ist es nun durch Analyse des verteilten Systemkonzepts jene Anwendungsfälle zu finden, welche für die Entwicklung einer robusten LLC einer Lagergasse wichtig sind.

Um die Komplexität des Systems zu vermindern wird für die weiteren Betrachtungen das Portal nicht berücksichtigt. Dieses wird für den direkten Lagerprozess nicht benötigt und hat so keine Auswirkungen auf die LLC der Lagergasse.

4.3. Geschäftsprozesse

Grundsätzlich lassen sich die Geschäftsprozesse eines verteilten Lagersystems in zwei Kategorien einteilen: Prozesse die nur das Lagersystem selbst betreffen und Prozesse die sowohl das Lagersystem als auch externe Akteure betreffen. Von größerer Interesse ist hierbei die letztere Kategorie, denn im Gegensatz zu systeminternen Prozessen muss bei externen Prozessen eine Kommunikation zwischen den beteiligten Akteuren stattfinden.

Die drei wesentlichen Geschäftsprozesse eines HRLs sind

- das Einlagern von Teilen,
- das Auslagern von Teilen und
- die Lagerverwaltung

Wie bereits oben beschrieben liegt das Hauptaugenmerk auf das Einlagern und Auslagern von Teilen, da bei diesen Geschäftsprozessen auch externe Akteure beteiligt sind. Für diese Geschäftsprozesse lassen sich die Anwendungsfälle durch die Interaktion zwischen den Akteuren finden. Um diese auf einfache Weise zu veranschaulichen, werden Analogien in vertrauten verteilten Systemen verwendet, die das Verhalten eines verteilten Lagersystems widerspiegeln. Die verwendeten Analogien werden im weiteren auf das Lagersystem übertragen und können so für die Identifikation der Anwendungsfälle und deren Akteure verwendet werden.

4.3.1. Einlagerungsprozess

Für den Einlagerungsprozess lässt sich das Lagersystem in gewisser Weise mit einem Hotel vergleichen. Es besitzt mehrere Zimmer und stellt diese seinen Kunden für die Übernachtung zur Verfügung. Kunden können im Vorfeld Zimmer reservieren um bei der Ankunft im Hotel sicher ein Zimmer zu bekommen. Will ein Kunde ein Zimmer einer gewissen Kategorie haben und kein Zimmer dieser ist mehr verfügbar, kann ihm ein Zimmer einer anderen Kategorie angeboten werden.

Überträgt man dies jetzt auf das Lagersystem, so stellt jede Lagergasse ein Hotel dar. Die einzelnen Zimmer sind die Lagerplätze jeder Lagergasse. Dabei kann die Eigenschaft der unterschiedlich hohen Lagerplätze auf die unterschiedlichen Zimmerkategorien übertragen werden. Der Kunde kann jeder beliebige Knoten im Transportsystem sein, der ein Teil einlagern will. Dazu gibt es die Möglichkeit für das einzulagernde Teil einen Lagerplatz zu reservieren. Da das HRL aus mehreren Lagergassen

besteht, stellt sich die Frage, bei welcher Gasse der Kunde um einen Lagerplatz anfragt. Die Antwort ist denkbar einfach, bei allen. Da wie auch im richtigen Leben jeder Kunde für sich das am besten passende Zimmer haben möchte, fragt man meistens nicht nur bei einem Hotel sondern gleich bei mehreren an. Anhand der gesammelten Daten, wie Größe, Preis etc. trifft man dann eine Entscheidung. Auch dieses Verhalten lässt sich auf das verteilte System übertragen. Ein Knoten fragt also bei allen verfügbaren Lagergassen um einen Lagerplatz für das einzulagernde Teil an. Dabei können bei der Anfrage schon gewisse Forderungen (z.B. die Höhe, Traglast des Lagerplatzes, etc.) gestellt werden. Jede Lagergasse bietet dann einen Lagerplatz zu bestimmten Kosten an. Die Kosten sind in diesen Fall die Kriterien anhand der Knoten (Kunde) entscheiden soll, welchen Lagerplatz er nimmt. Kosten können z.B. der Füllgrad des Lagers, die Entfernung zum Lager, etc. sein. Ist in einer Lagergasse kein Lagerplatz mit der geforderten Höhe mehr verfügbar, jedoch einer mit einer größeren Höhe, so kann die Lagergasse diesen zu höheren Kosten anbieten. Wurde ein Lagerplatz ausgewählt, so soll dieser bis zur Ankunft des Teils im Lager reserviert werden. Diese Interaktion beschreibt den Anwendungsfall *Lagerplatz festlegen*, dessen Ergebnis ein reservierter Lagerplatz ist.

Da zwischen einer Lagerplatzanfrage und einer Lagerplatzreservierung auch Lagerplatzanfragen von anderen Knoten verarbeitet werden können, kann der Fall auftreten, dass mehrere Knoten den selben Lagerplatz reservieren wollen. Bei der Bestätigung der Reservierung gewinnt dann der schnellste Knoten und alle anderen gehen leer aus und müssen eine erneute Anfrage stellen. Soll dies vermieden werden, so kann der Lagerplatz bereits bei der Anfrage reserviert werden. Der reservierte Lagerplatz wird dann keinen anderen Knoten mehr angeboten. Es kann somit sichergestellt werden, dass jeder Knoten eine Lagerplatzanfrage nur einmal stellen muss. Nachteil ist jedoch, dass mit Abschluss der Lagerplatzfestlegung nicht nur der reservierte Lagerplatz bestätigt werden muss, sondern auch die nicht gewählten Lagerplätze wieder frei gegeben werden müssen. Dies führt zu einem erhöhten Datenaufkommen. Zusätzlich stehen die schlussendlich nicht gewählten Lagerplätze während der Lagerplatzfestlegung für andere Knoten nicht zur Verfügung. Welche Art der Umsetzung sinnvoll ist hängt von mehreren Faktoren ab. Zwei wesentliche sind die Häufigkeit der Anfragen und die Kostenfunktion. Wie bereits erwähnt wird auf Basis der Kostenfunktion die Entscheidung getroffen, welcher Lagerplatz gewählt werden soll. Führt diese für verschiedene Knoten meist zum selben Ergebnis (Lagerplatz), so ist eine Reservierung während der Anfrage sinnvoll. Für die weiteren Betrachtungen wird eine Reservierung während der Anfrage nicht vorgesehen.

Es kann der Fall auftreten, dass sich ein Teil gerade auf dem Weg ins Lager befindet. Parallel dazu wird das gleiche Teil im System an einer anderen Stelle benötigt. Anstelle ein gleichwertiges Teil auszulagern, kann das Ziel des sich noch im Transportsystem befindlichen Teils einfach geändert werden. Dadurch kommt das Teil nie im Lager an, da es jetzt an einer anderen Stelle im System verwendet wird. Es muss also die Reservierung des Lagerplatzes storniert werden.

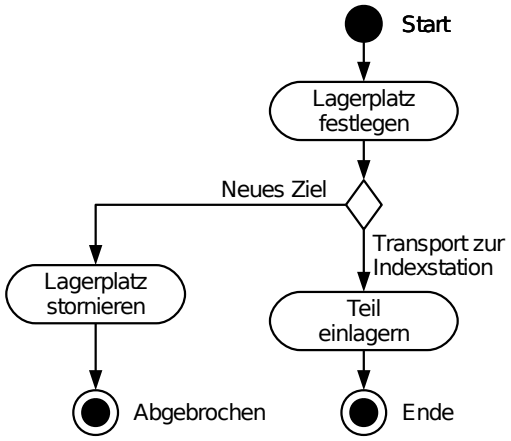


Abbildung 4.3.: UML Aktivitätsdiagramm für den Prozess Einlagern

Der beschriebene Einlagerungsprozess lässt sich nun in Form eines UML-Aktivitätsdiagramms angeben. Dieses ist in Abbildung 4.3 ersichtlich.

4.3.2. Auslagerungsprozess

Für den Auslagerungsprozess lässt sich das Lagersystem mit einem Versandhandel vergleichen. Dieser bietet verschiedene Produkte zu gewissen Kosten an. Es kann die Verfügbarkeit überprüft werden und schlussendlich ein Produkt bestellt werden. Will ein Kunde nun ein Produkt bestellen, so kann er als erstes die Verfügbarkeit überprüfen. Ist das Produkt verfügbar, kann er es zu den angegebenen Kosten kaufen. Die Zustellung der Produkte übernimmt der Versandhandel nicht selbst, sondern beauftragt einen Paketdienst oder einen ähnlichen Dienstleister damit.

Überträgt man dies wieder auf das Lagersystem, so stellt jede Lagergasse einen Versandhandel dar. Jeder Knoten, der ein Teil benötigt, kann als Kunden angesehen werden. Um das am besten passende Teil zu finden, fragt der Knoten die Verfügbarkeit in jeder Lagergassen an. Die Lagergasse übermittelt dann, ob das Teil verfügbar ist und die dazugehörigen Kosten. Kosten können in Fall des Auslagerungsprozess z.B. die Anzahl der verfügbaren Teile im Lager, die Entfernung zum anfragenden Knoten, die Länge des Aufenthalts im Lager des benötigten Teils (um Überalterung zu vermeiden), etc. sein. Auf Basis der Kosten kann der Knoten entscheiden von welcher Lagergasse er das Teil anfordert. Diese Interaktion beschreibt den Anwendungsfall *Teil festlegen*, dessen Ergebnis ein reserviertes Teil ist. Es besteht die Möglichkeit, dass eine Anfrage eines anderen Knoten zwischen einer Teilanfrage und einer Teilreservierung verarbeitet wird. Das bereits beim Anwendungsfall *Lagerplatz festlegen* erläuterte Verhalten kann hier eins zu eins angewandt werden. Jedoch muss dabei berücksichtigt werden, dass die Teilevielfalt meist um ein vielfaches größer ist als die Anzahl der freien Lagerplätze pro Lagergasse. Die Wahrscheinlichkeit, dass zwei

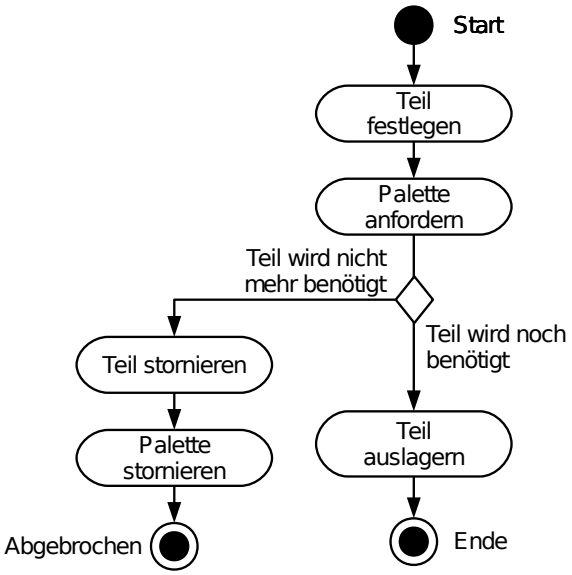


Abbildung 4.4.: UML Aktivitätsdiagramm für den Prozess Auslagern

identische Teile angefragt werden, ist also um ein vielfaches geringer. Für die weiteren Betrachtungen wird eine Reservierung während der Anfrage nicht vorgesehen.

Für die Zustellung des Teils ist das Transportsystem zuständig. Dazu muss die Lagergasse eine Transportpalette anfordern, auf welcher das benötigte Teil ausgelagert werden kann. Ab diesem Zeitpunkt ist das Lagersystem mit der Abarbeitung des Auftrages fertig. Da zwischen der Auftragserteilung (Festlegung des auszulagernden Teils) und dem tatsächlichen Auslagern des Teils auf die Transportpalette ein zeitlicher Abstand besteht, kann in dieser Zeit der Auftrag storniert werden. Ein möglicher Grund dafür wäre z.B. der Ausfall einer Produktionsmaschine. Die zuvor angeforderte Transportpalette wird dann ebenfalls nicht mehr benötigt bzw. kann für eine andere Anfrage verwendet werden.

Der beschriebene Auslagerungsprozess lässt sich nun in Form eines UML-Aktivitätsdiagramms angeben. Dieses ist in Abbildung 4.4 ersichtlich.

4.4. Anwendungsfälle und Akteure

Mit Hilfe der beiden angeführten Beispiele konnten auf einfache Art und Weise alle Anwendungsfälle der beiden Geschäftsprozesse ,,Einlagern von Teilen" und ,,Auslagern von Teilen" identifiziert werden. Wie bereits zu Anfang beschrieben, beziehen sich die identifizierten Anwendungsfälle auf die Interaktion zwischen dem Lagersystem und den externen Akteuren. Im weiteren müssen jetzt nur mehr die Anwendungsfälle für den Geschäftsprozess Lagerverwaltung gefunden werden. Dazu werden als Grundlage die Funktionen einer herkömmlichen Lagerverwaltung herangezogen.

Die identifizierten Anwendungsfälle sind in der nachfolgenden Liste zusammengefasst, werden aber im nächsten Abschnitt noch genauer behandelt.

Anwendungsfälle:

- Lagerplatz festlegen
- Lagerplatz stornieren
- Teil einlagern
- Teil festlegen
- Teil stornieren
- Teil auslagern
- Palette anfordern
- Paletten stornieren
- Teil umlagern
- Zustand anzeigen
- Teile anlegen
- Teile ändern
- Teile löschen

Die Anwendungsfälle *Teile anlegen*, *Teile ändern* und *Teile löschen* sind hier der Vollständigkeit halber angeführt, werden aber der Übersichtlichkeit halber im weiteren im Anwendungsfall *Teile verwalten* zusammengefasst. Diese Anwendungsfälle sind für die Erzeugung/Manipulation eines virtuelle Abbildes eines jeweiligen Teils zuständig. Dieses Abbild beinhaltet alle Daten (z.B. ID, Höhe, Gewicht, etc.), die für den Lagerprozess notwendig sind. Dabei muss eine einheitliche Teileidentifikation über alle Systeme hinweg ermöglicht werden.

Im weiteren müssen nun die beteiligten Akteure identifiziert werden. Diese können anhand der bereits identifizierten Anwendungsfälle leicht gefunden werden. Die beteiligten Akteure sind in der nachfolgenden Liste zusammengefasst und werden dort kurz erklärt.

Beteiligte Akteure:

- *Knoten:* ist ein beliebiger Akteur (Indexstation, Produktionsanlage, etc.), der ein Teil ein- bzw. auslagern möchte.
- *Übergabestation:* ist eine Indexstation, die für den Material-/Warenaustausch zwischen Lagergasse und PTS zuständig ist.
- *Lagergasse:* stellt alle für den Lagerprozess notwendigen Funktionen zur Verfügung. Sie ist somit der Hauptakteur der Steuerung und ist deswegen an allen Anwendungsfällen beteiligt.
- *Palettentransportsystem (PTS)*: stellt jene Funktionen zur Verfügung, die keinem einzelnen Knoten des PTSs zugeordnet werden können, sondern dem gesamten System.
- *Warenwirtschaftssystem/Produktionsplanungssystem (WWS/PPS)*: ist für eine einheitliche Teileidentifikation über alle Systeme hinweg zuständig und kann unter Berücksichtigung der Zustände einzelner Systeme die globalen und lokalen Ziele vorgeben.

Nachdem nun alle notwendigen Anwendungsfälle und die daran beteiligten Akteure identifiziert worden sind, kann mit der genaueren Analyse des jeweiligen Anwendungsfalls begonnen werden.

4.5. Anwendungsfallanalyse

In diesem Abschnitt werden nun alle Anwendungsfälle genauer erklärt. Abbildung 4.5 zeigt nochmals alle Anwendungsfälle und ihre Akteure in einem Anwendungsfalldiagramm.

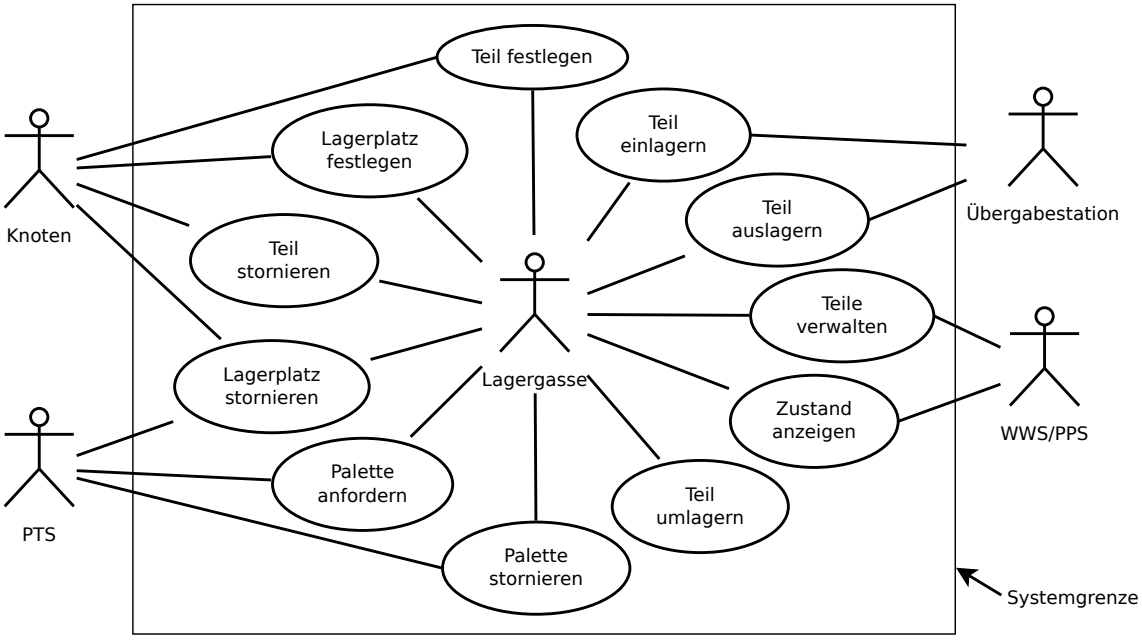


Abbildung 4.5.: Anwendungsfälle und ihre Akteure

Name:	**Lagerplatz festlegen**
Kurzbeschreibung:	Ein Knoten stellt eine Lagerplatzanfrage um für das einzulagernde Teil den besten Lagerort bestimmen zu können.
Akteure:	Knoten, Lagergassen
Auslöser:	Teil soll eingelagert werden
Vorbedingungen:	Knoten muss die Lagergassen kennen, einzulagerndes Teil muss dem Lager bekannt sein
Eingehende Informationen:	Teil-ID, Parameter (spiegelt die lokalen und globalen Ziele des jeweiligen Knotens wieder)
Ergebnisse:	reservierter Lagerplatz (Reservierungsnummer)
Nachbedingungen:	Palette wird zur ermittelten Lagergasse geroutet
Ablauf:	1. Knoten übermittelt Teileidentifikation und Parameter an alle verfügbaren Lagergassen 2. Lagergassen bestimmen unter Berücksichtigung der Teilspezifikation (Höhe, Gewicht, etc.), des Parameters des Knotens und der eigenen lokalen und globalen Ziele einen Lagerplatz. 3. Lagergassen berechnen die Kosten für den Lagerplatz und übermittelt diese dem anfragenden Knoten 4. Knoten bestimmt auf Basis der Kosten den passenden Lagerplatz 5. Knoten reserviert bei der Lagergasse mit den geringsten Kosten einen Lagerplatz 6. Lagergasse bestätigt die Reservierung

Name:	**Lagerplatz stornieren**
Kurzbeschreibung:	Ein Knoten bzw. das PTS storniert den zuvor reservierten Lagerplatz für ein bestimmtes einzulagernde Teil.
Akteure:	Knoten, PTS, Lagergasse
Auslöser:	Teil wird an einer anderen Stelle im System benötigt und muss deswegen nicht mehr eingelagert werden
Vorbedingungen:	Lagerplatzreservierung muss vorliegen
Eingehende Informationen:	Reservierungsnummer
Ergebnisse:	zuvor reservierter Lagerplatz ist wieder frei verfügbar
Nachbedingungen:	Teil wird zu seinem neuen Bestimmungsort geschickt
Ablauf:	1. Knoten übermittelt Reservierungsnummer an Lagergasse 2. Lagergasse storniert Reservierung 3. Lagergasse übermittelt Bestätigung an den anfragenden Knoten

Name:	**Teil einlagern**
Kurzbeschreibung:	Ein Teil wird von der Transportpalette aufgenommen und auf den reservierten Lagerplatz abgelegt.
Akteure:	Übergabestation, Lagergasse
Auslöser:	Übergabestation meldet die Ankunft einer vollen Transportpalette
Vorbedingungen:	volle Transportpalette befindet sich in der Indexstation, Lagerplatz wurde reserviert
Eingehende Informationen:	Teil-ID, Reservierungsnummer
Ergebnisse:	Teil befindet sich am zugewiesenen Lagerplatz, Transportpalette ist leer
Nachbedingungen:	Übergabestation meldet leere Palette
Ablauf:	1. Übergabestation meldet Ankunft einer vollen Transportpalette 2. Zugewiesener Lagerplatz wird ermittelt 3. Teil wird in den reservierten Lagerplatz eingelagert 4. Erfolgreiche Abarbeitung wird bestätigt

Name:	**Teil festlegen**
Kurzbeschreibung:	Ein Knoten benötigt ein bestimmtes Teil. Aus diesem Grund stellt er eine Anfrage an alle verfügbaren Lagergassen um von mehreren Teilen des selben Typs, das am besten passende Teil bestimmen zu können.
Akteure:	Knoten, Lagergassen
Auslöser:	Teil wird im System benötigt
Vorbedingungen:	Knoten muss die Lagergassen kennen, auszulagerndes Teil muss dem Lager bekannt sein
Eingehende Informationen:	Teil-ID, Parameter (spiegelt die lokalen und globalen Ziele des jeweiligen Knotens wieder)
Ergebnisse:	reserviertes Teil
Nachbedingungen:	
Ablauf:	1. Knoten übermittelt Teileidentifikation und Parameter 2. Lagergassen bestimmen die Verfügbarkeit des angefragten Teils 3. Lagergassen berechnen Kosten und übermittelt diese an den anfragenden Knoten 4. Knoten bestimmt auf Basis der Kosten das passenden Teil 5. Knoten reserviert bei der Lagergasse mit den geringsten Kosten das passenden Teil 6. Lagergasse bestätigt die Reservierung

Name:	**Teil stornieren**
Kurzbeschreibung:	Ein Knoten storniert das zuvor reservierte Teil, da diese im System (z.B. durch Ausfall einer Fertigungsanlage) nicht mehr benötigt wird.
Akteure:	Knoten, Lagergasse
Auslöser:	Teil wird nicht mehr benötigt
Vorbedingungen:	Teilreservierung muss vorliegen
Eingehende Informationen:	Reservierungsnummer
Ergebnisse:	zuvor reserviertes Teil ist wieder frei verfügbar
Nachbedingungen:	
Ablauf:	1. Knoten übermittelt Reservierungsnummer an Lagergasse 2. Lagergasse storniert Reservierung 3. Lagergasse übermittelt Bestätigung an den anfragenden Knoten

Name:	**Teil auslagern**
Kurzbeschreibung:	Ein Teil wird von einem zugewiesenen Lagerplatz aufgenommen und auf der Transportpalette abgelegt.
Akteure:	Übergabestation, Lagergasse
Auslöser:	Übergabestation meldet die Ankunft einer leeren Transportpalette
Vorbedingungen:	leere Transportpalette befindet sich in der Indexstation, Teil wurde angefordert
Eingehende Informationen:	Reservierungsnummer
Ergebnisse:	Teil befindet sich auf der Transportpalette, Lagerplatz ist leer
Nachbedingungen:	Volle Transportpalette befindet sich in der Übergabestation
Ablauf:	1. Übergabestation meldet Ankunft einer leeren Transportpalette 2. Lagerplatz des reservierten Teils wird ermittelt 3. Teil wird vom zuvor ermittelten Lagerplatz ausgelagert 4. Erfolgreiche Abarbeitung wird bestätigt

Name:	**Palette anfordern**
Kurzbeschreibung:	Für das Auslagern eines Teils wird eine Transportpalette benötigt. Diese wird vom PTS angefordert.
Akteure:	Lagergasse, PTS
Auslöser:	Teil soll ausgelagert werden.
Vorbedingungen:	Teil für die Auslagerung wurde festgelegt und reserviert
Eingehende Informationen:	Paletten-ID
Ergebnisse:	Leere Palette ist reserviert.
Nachbedingungen:	Leere Transportpalette befindet sich auf dem Weg zur Übergabestation der anfragenden Lagergasse
Ablauf:	1. Lagergasse fordert eine leere Transportpalette vom PTS an 2. PTS ermittelt verfügbare Transportpalette in der Nähe der Übergabestation der anfragenden Lagergasse 3. PTS übermittelt Paletten-ID

Name:	**Palette stornieren**
Kurzbeschreibung:	Ein zuvor reserviertes Teil wir im System (z.B. durch Ausfall einer Fertigungsanlage) nicht mehr benötigt wird. Dadurch wird die angeforderte Transportpalette ebenfalls nicht mehr benötigt. Diese wird von der Lagergasse storniert.
Akteure:	Lagergasse, PTS
Auslöser:	reserviertes Teil wird storniert
Vorbedingungen:	Anforderung einer leeren Palette muss vorliegen
Eingehende Informationen:	Paletten-ID
Ergebnisse:	angeforderte Palette ist wieder verfügbar
Nachbedingungen:	Bestimmungsort der Palette wird geändert
Ablauf:	1. Lagergasse übermittelt Paletten-ID an PTS 2. PTS storniert die angeforderte Transportpalette 3. PTS übermittelt Bestätigung an die anfragende Lagergasse

Name:	**Teil umlagern**
Kurzbeschreibung:	Ein Teil wird aus dem zugewiesenen Lagerplatz aufgenommen und auf einen optimaleren Lagerplatz abgelegt.
Akteure:	Lagergasse
Auslöser:	Leistungsfähigkeit des Lagers steigern
Vorbedingungen:	Lagerplätze mit eingelagerten Teilen können mit freien nicht reservierten Lagerplätzen getauscht werden. Die freien Lagerplätze müssen sich jedoch näher zum Übergabepunkt zwischen Lagergasse und Übergabestation befinden, als der bestehende Lagerplatz.
Eingehende Informationen:	
Ergebnisse:	Ausgehend vom Übergabepunkt befinden sich zuerst alle besetzten Lagerplätze.
Nachbedingungen:	
Ablauf:	1. Lagergasse bestimmt für den bestehenden Lagerplatz einen neuen freien nicht reservierten Lagerplatz der näher beim Übergabepunkt liegt. 2. Lagergasse nimmt Teil am bestehenden Lagerplatz auf und legt es am neuen optimaleren Lagerplatz ab.

Name:	**Zustand anzeigen**
Kurzbeschreibung:	Die Lagergasse übermittelt den aktuellen Zustand des Lagers (Belegung, Parameter, etc.)
Akteure:	WWS/PPS, Lagergasse
Auslöser:	WWS/PPS benötigt den aktuellen Zustand der Lagergasse
Vorbedingungen:	Lagergasse muss in Betrieb sein
Eingehende Informationen:	Zustandsabfrage
Ergebnisse:	belegte und reservierte Lagerplätze mit den zugehörigen Teile-IDs
Nachbedingungen:	
Ablauf:	1. WWS/PPS übermittelt Zustandsabfrage an Lagergasse 2. Lagergasse übermittelt belegte und reservierte Lagerplätze mit den zugehörigen Teile-IDs

Name:	**Teile verwalten**
Kurzbeschreibung:	Das WWS/PPS aktualisiert die Teiledatenbank der Lagergasse
Akteure:	WWS/PPS, Lagergasse
Auslöser:	neues Teil aufgenommen, bestehendes Teil geändert oder gelöscht
Vorbedingungen:	Teil vorhanden bzw. nicht vorhanden
Eingehende Informationen:	aktualisierter Teiledatensatz
Ergebnisse:	Teiledatenbank des Lagers ist auf den aktuellen Stand
Nachbedingungen:	
Ablauf:	1. WWS/PPS übermittelt aktuellen Teiledatensatz 2. Lagergasse bestätigt die Übermittlung

4.6. Zusammenfassung

In einem ersten Schritt wurde eine Systemanalyse eines verteilt gesteuerten Lagersystems durchgeführt. Dazu wurde einerseits das System von der Materialflussebene, andererseits von der Steuerungsebene betrachtet. In der Materialflussebene können nur physikalische Komponenten auftreten. Sie zeigt an zwischen welchen Komponenten ein Material-/Warenaustausch stattfinden kann. In der Steuerungsebene treten zusätzlich steuerungstechnische Komponenten auf, die keiner Hardware direkt zugeordnet werden können. Sie zeigt den steuerungstechnischen Aufbau des Gesamtsystems an. In einem nächsten Schritt wurde die Systemidee hinter dem verteilten Steuerungskonzept skizziert. Aufbauend auf der Systemidee wurden die Geschäftsprozesse eines verteilt gesteuerten Lagersystems ermittelt und in weiterer Folge die notwendigen Anwendungsfälle identifiziert und genauer beschrieben. Basierend auf der objektorientierten Analyse kann nun der Funktionsumfang der LLC festgelegt werden.

5. Low Level Control

Dieses Kapitel beschäftigt sich mit der konkreten Umsetzung der LLC im Standard IEC 61499 des in Kapitel 3 vorgestellten HRLs. In einem ersten Schritt wird der Funktionsumfang der LLC festgelegt. Anschließend wird die Struktur der gesamten Steuerung vorgestellt. In weiterer Folge werden die wesentlichen Funktionsblöcke der LLC genauer erklärt.

5.1. Funktionsumfang der Low Level Control

Um den Funktionsumfang der LLC festlegen zu können, diente einerseits die im Kapitel 4 erstellte Analyse und andererseits der Stand der Technik als Grundlage. Mit Hilfe der Anwendungsfallanalyse können die hardwarenahen Anwendungsfälle identifiziert werden, die die Grundfunktionen der LLC definieren. Aus dem Stand der Technik werden die relevanten Strategien ermittelt, die in der LLC umgesetzt werden sollen.

5.1.1. Hardwarenahe Anwendungsfälle

Die hardwarenahen Anwendungsfälle lassen sich sehr leicht aus den in Abschnitt 4.5 analysierten Anwendungsfällen identifizieren, da mit dessen Hilfe der Materialfluss manipuliert werden kann. Die nachfolgende Liste zeigt die hardwarenahen Anwendungsfälle.

- Teil einlagern
- Teil auslagern
- Teil umlagern

Um nun die Grundfunktionen festlegen zu können, wird eine Betrachtung auf der Materialflussebene durchgeführt. Dazu wird für jeden der oben angeführten Anwendungsfälle jeweils die Anfangs- und Endposition des Materialflusses betrachtet. Beim Einlagern ist die Anfangspostion die Übergabestation und die Endposition ein beliebiger Lagerplatz der Lagergasse. Beim Auslagern kann die Anfangsposition ein beliebiger Lagerplatz sein und die Endposition ist die Übergabestation. Beim Umlagern können Anfangsposition und Endposition ein beliebiger Lagerplatz sein. Da für den Materialfluss selbst keine weiteren Informationen außer Anfangs- und

Endposition notwendig sind, kann der gesamte Bewegungsablauf in der LLC implementiert werden. Die Voraussetzung jedoch dafür ist, dass die Anfangsposition belegt und die Endposition frei ist. Dies muss durch die HLC sichergestellt werden. Die Anfangs- und Endposition werden dabei durch die HLC vorgegeben.

In einem nächsten Schritt kann nun versucht werden, die Wahl der Anfangs- bzw. Endposition durch die LLC vornehmen zulassen. Dazu muss zuvor eruiert werden, wie und wann die Vorgabe durch die HLC stattfindet. Da beim Ein- und Auslagern bereits jeweils eine Position fix vorgegeben ist (Übergabestation), werden als erstes diese beiden Anwendungsfälle betrachtet. Die Festlegung der anderen Position (Lagerplatz) erfolgt bereits bei einer Lagerplatz-/Teilanfrage. Hier wird anhand der eingehenden Parameter und der lokalen und globalen Ziele der passende Lagerplatz/Teil gewählt und die jeweiligen Kosten dafür berechnet. Um diese Funktion in die LLC auslagern zu können, benötigte die LLC den selben Informationsgehalt wie die HLC. Da in der LLC meist zeitkritische Funktionen implementiert sind (Regelungen, Überwachung von Anlagenteilen, ...) könnte dieser Mehraufwand die LLC zu stark belasten und die HLC fast überflüssig machen. Aus diesem Grund macht es keinen Sinn diese Funktion in die LLC auszulagern.

In einem letzten Schritt wird nun untersucht welche Akteure an den verschiedenen hardwarenahen Anwendungsfällen beteiligt sind und wie weit diese während der Abarbeitung der Anwendungsfälle belegt sind. Die beiden relevanten Anwendungsfälle sind *Teil einlagern* und *Teil auslagern* da bei diesen Anwendungsfällen mehr als ein Akteur beteiligt ist. Die beteiligten Akteure sind die Übergabestation und die Lagergasse. Als erstes wird der Anwendungsfall *Teil einlagern* betrachtet. Sobald das Teil von der Transportpalette abgehoben worden ist, wird die Übergabestation nicht mehr benötigt. Diese kann freigegeben werden sobald das RBG sich außerhalb des Übergabebereichs befindet. Dadurch kann auf einfache Weise die Effizienz der Anlage gesteigert werden. Betrachtet man nun den Anwendungsfall *Teil auslagern* so ist der Sachverhalt etwas komplizierter. Die Übergabestation wird hier erst beim Ablegen des Teils auf der Transportpalette benötigt. Der optimale Fall wäre, wenn die Transportpalette in dem Augenblick in die Übergabestation einfahren würde, in dem das RBG das Teil auf die Palette ablegen will. Um dies jedoch realisieren zu können, ist eine komplexe Kontrollstruktur notwendig, welche den Aufenthalt der jeweiligen Transportpaletten kennt. Diese leitet dann den Auslagerungsauftrag zeitgerecht ein. Da eine solche Kontrollstruktur noch nicht existiert, wird der Auslagerungsauftrag mit dem Einfahren der Transportpalette in die Übergabestation gestartet. Dadurch muss sich die Transportpalette während des gesamten Vorgangs in der Station befinden.

5.1.2. Relevante Strategien

In weiterer Folge werden nun die für die LLC relevanten Strategien gesucht. Zur Erinnerung werden die im Stand der Technik vorgestellten Strategien hier nochmals angeführt. Diese sind

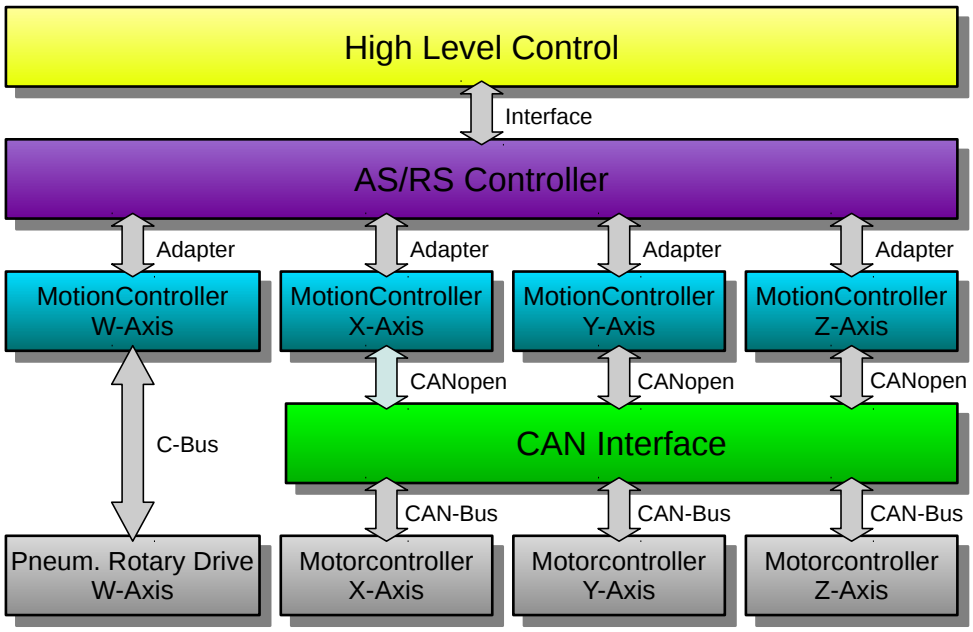


Abbildung 5.1.: Struktureller Aufbau der Hochregallagersteuerung

- Belegungsstrategien,
- Bewegungsstrategien,
- Auslagerungsstrategien und
- Verweilpositionsstrategien.

Um die Belegungs- und Auslagerungsstrategien in der LLC umsetzen zu können, müsste die Wahl des Lagerplatzes von der LLC getroffen werden. Da dies wie bereits oben beschrieben die LLC zu stark belasten könnte, sind diese beiden Strategien für die LLC nicht relevant. Die Bewegungsstrategien beschreiben die Art des Materialflusses. Diese müssen für die Umsetzung der LLC berücksichtigt werden. Die LLC muss die beiden Bewegungsabfolgen für Einfachspiel als auch Doppelspiel unterstützen. Welche der beiden Bewegungsabfolgen jedoch gewählt werden soll, legt die HLC fest. Die verschiedenen Verweilpositionsstrategien beschreiben wo das RBG auf den nächsten Auftrag warten soll. Da für das Anfahren der Verweilposition keine zusätzlichen Informationen notwendig sind, kann dies ebenfalls in der LLC umgesetzt werden. Die Wahl der Verweilpositionsstrategie wird jedoch von der HLC festgelegt.

5.2. Struktur der Steuerung

Abbildung 5.1 zeigt die Struktur der Hochregallagersteuerung. Die unterste Ebene stellt die Hardwareansteuerung dar. Diese stellt Funktionen zur Verfügung mit denen es möglich ist die jeweiligen Aktuatoren zu treiben. Im Falle der Motorcontroller stehen zusätzlich weitere Funktionen (z.B. Positionsregelung, etc.) zu Verfügung, welche über CANopen angesprochen werden können. Die Motorcontroller sind mittels CAN mit der Steuerung verbunden und der pneumatische Drehantrieb wird über Ventile und Sensoren direkt über die I/O-Module an die Steuerung angebunden. Im konkreten Fall wird eine SPS von Festo verwendet, bei der die Kommunikation mit den I/O-Modulen über den firmeneigenen Bus, genannt C-Bus, stattfindet. Für die CAN Kommunikation zwischen Steuerung und Motorcontroller wird das in Abschnitt 5.5 beschriebene CAN Interface verwendet. Dieses stellt im Falle der Motorcontroller das Bindeglied zwischen Hardwareansteuerung und Motion Controller-Ebene dar. Der Motion Controller übernimmt die Parametrierung, Ansteuerung und Überwachung der Motorcontroller bzw. Pneumatikventile und zugehörigen Sensoren. Die Kommunikation zwischen Motion Controller und Motorcontroller erfolgt über CANopen. Dabei steht das ,,Device Profile DS402 Motion Control" zur Verfügung. ,,CANopen für Motorcontroller CMMS/CMMD" [3] gibt Auskunft über die herstellerspezifischen Objekte. Die Koordination der einzelnen Achsen wird durch den AS/RS Controller übernommen. Die Kommunikation mit den Motion Controllern erfolgt über einen einheitlichen Adapter. Die Lagerverwaltung ist in der HLC implementiert. In Kapitel 4 wurden die notwendigen Funktionen der HLC beschrieben. Damit die HLC ihre Aufgaben erfüllen kann, stellt ihr die LLC gewisse Funktionen zur Verfügung.

5.3. AS/RS Controller

Der AS/RS Controller bildet die oberste Ebene der LLC. Er ist einerseits die Schnittstelle zur HLC und stellt die Funktionen der LLC über sein Interface der HLC zur Verfügung. Andererseits übernimmt er die Koordination der einzelnen Achsen.

5.3.1. Interface - Schnittstelle zur HLC

Wie bereits erwähnt bildet das Interface des AS/RS Controller die Schnittstelle zur HLC. Zusätzlich stellt es alle notwendigen Funktionen für den Betrieb einer Lagergasse zur Verfügung. Abbildung 5.2 zeigt das Interface des AS/RS Controller FBs. Die Funktionen die der HLC zur Verfügung stehen sind:

- Initialisierung
  Der AS/RS Controller koordiniert die Initialisierung der Achsen des RBGs und wartet bis diese erfolgreich abgeschlossen ist. Die Initialisierung wird über den `INIT` Eventeingang angestoßen.

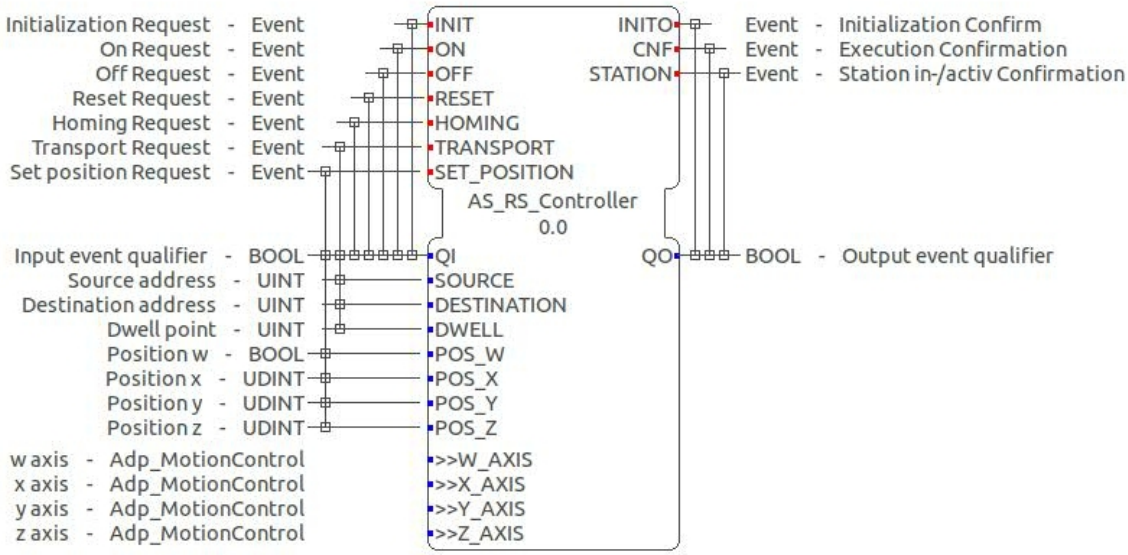


Abbildung 5.2.: Interface des AS/RS Controller Funktionsblocks

- Ein- bzw. Ausschalten der Achsen
  Der AS/RS Controller koordiniert das Ein- bzw. Ausschalten der Achsen des RBGs und wartet bis dieser Vorgang erfolgreich abgeschlossen sind. Das Ein- bzw. Ausschalten der Achsen wird über den `ON` bzw. `OFF` Eventeingang angestoßen.
- Zurücksetzen
  Im Falle eines Fehlers bzw. in Folge einer Notabschaltung kann über den Eventeingang `RESET` der AS/RS Controller wieder in einen definierten Zustand gebracht werden.
- Referenzieren der Achsen
  Der AS/RS Controller koordiniert die Referenzierung aller Achsen des RBGs und wartet bis diese erfolgreich abgeschlossen ist. Die Referenzierung wird mit Hilfe des Homing-Modus durchgeführt und wird über den `HOMING` Eventeingang angestoßen.
- Automatische Abarbeitung eines Ein-, Aus- oder Umlagerungsauftrages
  Über den `TRANSPORT` Eventeingang wird der Automatik-Modus gestartet, mit Hilfe dessen die automatische Abarbeitung eines Ein-, Aus- oder Umlagerungsauftrages durchgeführt werden kann.
- Individuelle Bewegung der Achsen
  Über den `SET_POSITION` Eventeingang wird der Manuelle-Modus gestartet, mit Hilfe dessen sich alle Achse individuell verfahren lassen.

Um die oben beschriebenen Funktionen umsetzen zu können, nutzt der AS/RS Controller wiederum die Funktionen der darunter liegenden Motion Controllern . Die Kommunikation mit den Motion Controllern der jeweiligen Achsen erfolgt über einen

einheitlichen Adapter. Dieser ist in Abbildung 5.3 ersichtlich. Der Motion Control Adapter fasst alle notwendigen Event- und Datenleitungen zusammen, die für die Kommunikation notwendig sind.

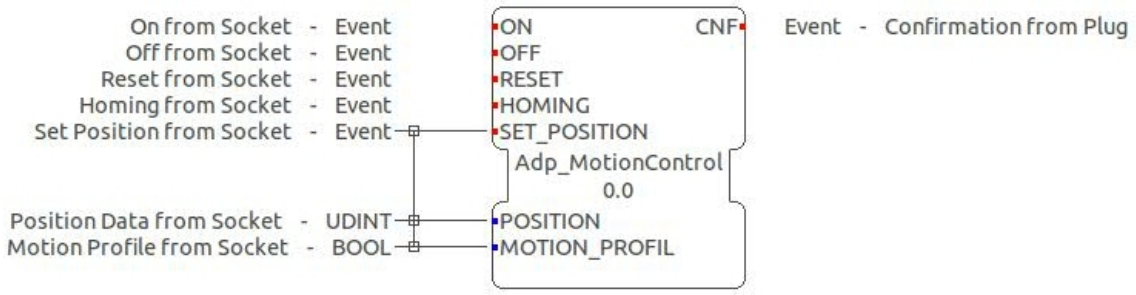


Abbildung 5.3.: Interface des MotionControl Adapters

5.3.2. Betriebszustände

Der AS/RS Controller kennt prinzipiell drei Betriebszustände. Diese sind

- Homing-Modus,
- Automatik-Modus und
- Manuell-Modus.

Die Wahl des Betriebszustandes erfolgt über den jeweiligen Event-Eingang des AS/RS Controllers. Um jedoch in den Automatik- bzw. Manuell-Modus wechseln zu können, muss nach jeder Neuinitialisierung einmalig eine erfolgreiche Referenzierung erfolgen. Diese wird im Homing-Modus durchgeführt.

5.3.2.1. Homing-Modus

Der Homing-Modus wird über den Event-Eingang `HOMING` gestartet. In diesem Modus werden die Referenzpositionen der jeweiligen Achsen angefahren. Dabei ist es wichtig, dass eine gewisse Reihenfolge eingehalten wird. Als erstes wird die Referenzposition der z-Achse angefahren. Dadurch kann sichergestellt werden, dass es bei einem Verfahren der x- und y-Achse zu keiner Kollision mit den Regalebenen kommt. Als nächstes werden die Referenzpositionen der x- und y-Achse angefahren. Dies passiert gleichzeitig. Wenn beide Achsen die Referenzposition erreicht haben, wird die Referenzposition der w-Achse angefahren. Vor diesem Zeitpunkt ist die w-Achse drucklos. Dadurch soll bei einer fehlerhaften Referenzierung der anderen Achse die w-Achse flexibel bleiben und ein Schaden auf Grund einer Kollision minimiert bzw. vermieden werden.

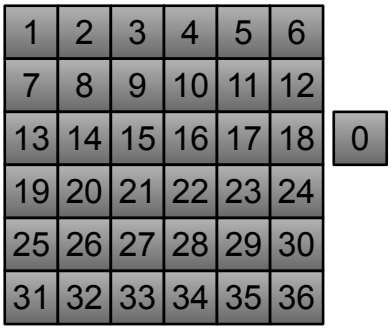


Abbildung 5.4.: Aufteilung der Adressen für eine Lagergasse

5.3.2.2. Automatik-Modus

Der Automatik-Modus wird über den Event-Eingang `TRANSPORT` gestartet. In diesem Modus wird der Bewegungsablauf, der für das Ein-, Aus- bzw. Umlagern notwendig ist, automatisch abgearbeitet. Um welchen der drei Bewegungsabläufe es sich handelt, wird anhand der Quell- und Zieladressen („Source" und „Destination") bestimmt. Die Aufteilung der Adressen für eine Lagergasse ist in Abbildung 5.4 ersichtlich. Die Übergabestation hat die Adresse „0" und besitzt eine Sonderstellung. Diese ist gegenüber den anderen Lagerpositionen um 90 Grad verdreht. Um die Übergabestation anfahren zu können, muss die w-Achse geschwenkt werden. Die anderen Lagerplätze sind ausgehend von der linken oberen Ecke durchnummeriert.

Wenn nun die Quelladresse „0" und die Zieladresse z.B. „12" ist, dann handelt es sich um einen Einlagerungsauftrag. Umgekehrt ist die Quelladresse „12" und die Zieladresse „0", so handelt es sich um einen Auslagerungsauftrag. Ist weder die Quell- noch die Zieladresse „0" so wird ein Umlagerungsauftrag durchgeführt. Die Abarbeitung des jeweiligen Bewegungsablaufes erfolgt mit Hilfe eines „Execution Control Chart" (ECC). Jeder Schritt des Bewegungsablaufes entspricht einem Zustand im ECC. Bei genauerer Betrachtung stellt man fest, dass alle notwendigen Schritte für das Umlagern bereits in den Bewegungsabfolgen für Ein- und Auslagern enthalten sind. Abbildung 5.5 zeigt die vereinfachte Darstellung des ECC für die Bewegungsabläufe von Ein-, Aus- und Umlagern. Das RBG unterstützt zwei Fahrprofile für schnelles und langsames Positionieren. Das verwendete Fahrprofil für jeden Zustand ist über dessen Farbe ersichtlich (grün für schnell und rot für langsam). Wie bereits im Abschnitt 5.1 besprochen kann bei der Abarbeitung eines Einlagerungsauftrags die Übergabestation vorzeitig freigegeben werden. Dies geschieht nach dem Zustand „Aus Übergabestation herausfahren" über den Event-Ausgang `STATION`.

Am Schluss jedes Bewegungsablaufes wird die Verweilposition angefahren, wo das RBG auf den nächsten Auftrag wartet. Es stehen drei verschiedene Verweilpositionen zur Verfügung. Diese können über den `DWELL` Eingang vorgegeben werden. Je nach

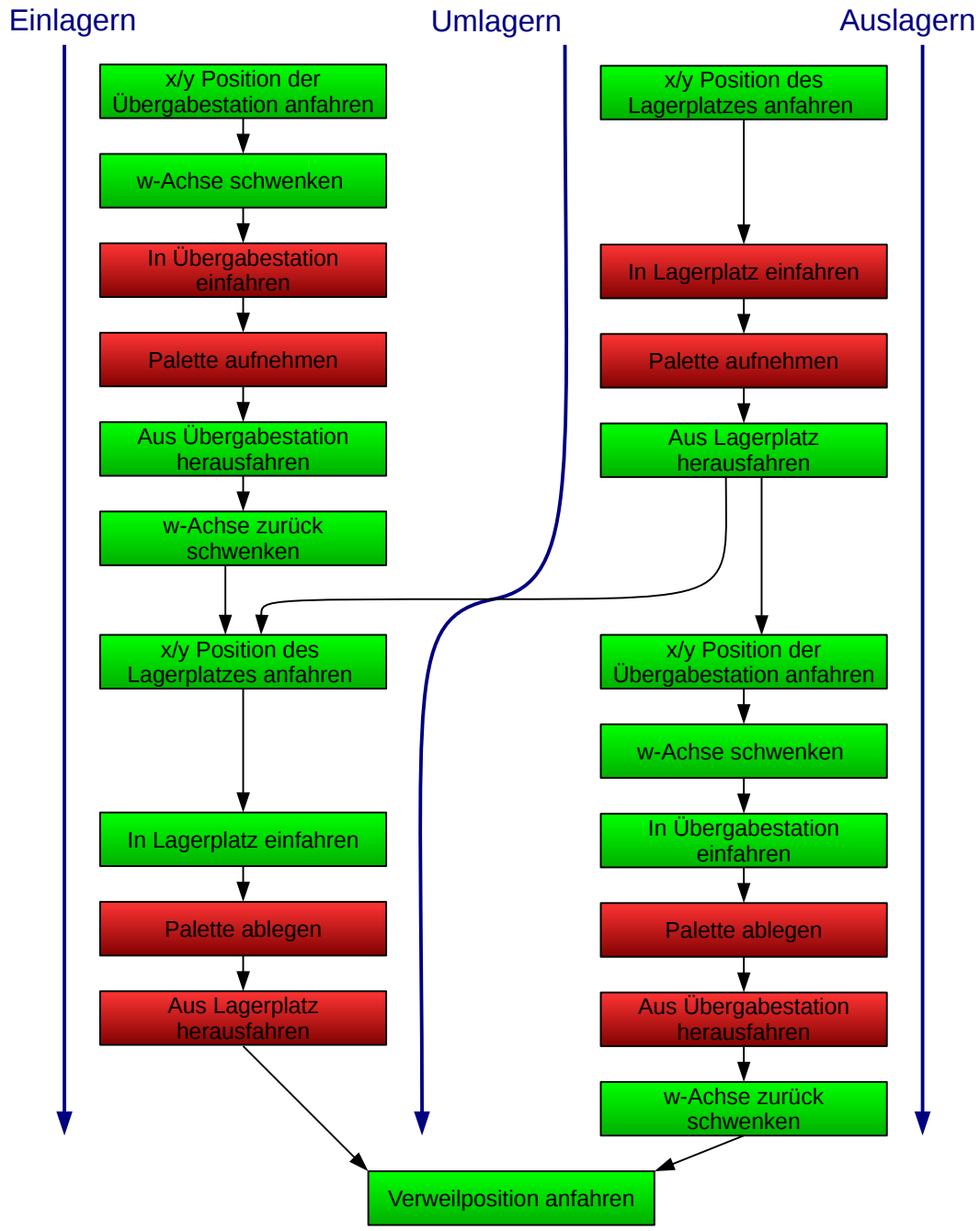


Abbildung 5.5.: Vereinfachte Darstellung des ECC für die Bewegungsabläufe von Ein-, Aus- und Umlagern

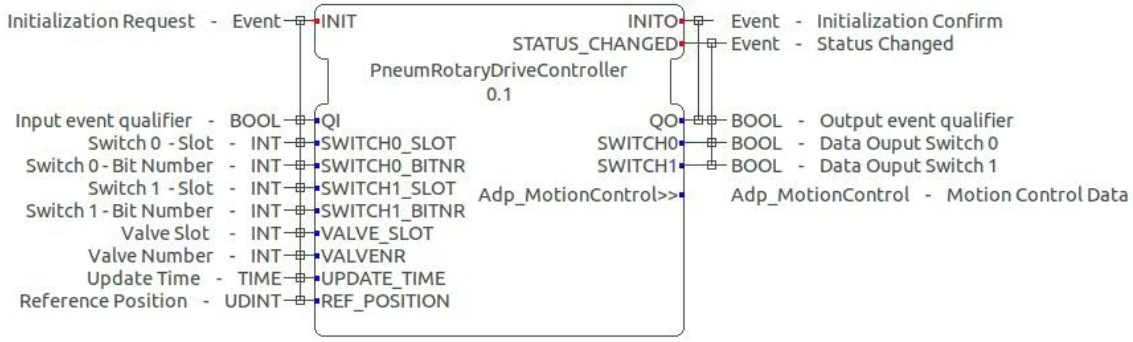


Abbildung 5.6.: Interface des Pneumatic Rotary Drive Funktionsblocks

Eingangswert wartet das RBG bei der Übergabestation (`DWELL` Eingang ist ,,0"), in der Mitte des Lagers (`DWELL` Eingang ist ,,1") oder an der letzten angefahrenen Position (`DWELL` Eingang ist ,,2"). In Verbindung mit den Verweilpositionen lassen sich nun Einzelspiel und Doppelspiel realisieren. Bei einem Einzelspiel muss die Verweilposition die Übergabestation sein, hingegen bei einem Doppelspiel muss die Verweilposition die letzte angefahrene Position sein. Die Verweilposition kann mit jeden neuen Transportauftrag geändert werden.

5.3.2.3. Manuell-Modus

Der Manuell-Modus wird über den Event-Eingang `SET_POSITION` gestartet. In diesem Modus können alle Achsen individuell verfahren werden. Dieser dient hauptsächlich zum einmaligen Einlernen der Lagerpositionen und eventuell für Wartungszwecke. Im Manuell-Modus kann allen Achsen eine Position vorgegeben werden, welche sie anfahren sollen. Die Eingabe der Position erfolgt über die Positionseingänge der jeweiligen Achsen (`POS_W, POS_X, POS_Y, POS_Z`).

5.4. Motion Controller

Die Motion Controller übernehmen die Ansteuerung der einzelnen Achse. Da das verwendete RBG einerseits aus elektrisch angetriebenen Zahnriemenachsen und andererseits aus einer pneumatischen Rotationsachse besteht, werden zwei unterschiedliche Motion Controller eingesetzt.

5.4.1. Pneumatic Rotary Drive Controller

Der Pneumatic Rotory Drive Controller ermöglicht das Ansteuern einer pneumatischen Rotationsachse mit Hilfe der Festo SPS CPX-CEC-C1 die über den C-Bus an eine Ventilinsel angeschlossen ist. Alle notwendigen Parameter können dabei über

die Dateneingänge vorgegeben werden. Mit Hilfe der Parameter kann bestimmt werden, welches Ventil bzw. welche Digitaleingänge der SPS verwendet werden sollen. Die Endlage der pneumatischen Achse wird mit Hilfe zweier Näherungsschalter erkannt. Diese sind an den konfigurierten Eingängen der SPS angeschlossen. Über den Eingang `UPDATE_TIME` wird festgelegt, in welchem Intervall die verwendeten Digitaleingänge eingelesen werden sollen. Mit Hilfe des `REF_POSITION` Eingangs kann festgelegt werden, welche Endlage die pneumatische Achse während der Referenzierung anfahren soll. Die Funktionen des Pneumatic Rotory Drive Controllers stehen über den MotionControl Adapter der übergeordneten Ebene zur Verfügung. Diese sind:

- Einschalten der Achsen
  Beim Einschalten der Achse bleibt die Achse solange drucklos, bis die erste Position vorgegeben wird. Dadurch kann sicher gestellt werden, dass die Achse an der letzten Position stehen bleibt, auch wenn sie sich zwischen den beiden Endlagen befindet. Nur im eingeschalteten Zustand können Positionen vorgegeben werden. Die Achse wird über den `ON` Eventeingang des Adapters eingeschaltet.
- Ausschalten der Achsen
  Beim Ausschalten der Achse wird die Achse drucklos geschaltet und lässt sich frei bewegen. Die Achse wird über den `OFF` Eventeingang des Adapters ausgeschaltet.
- Zurücksetzen
  Im Falle eines Fehlers bzw. in Folge einer Notabschaltung kann über den Eventeingang `RESET` des Adapters der Pneumatic Rotary Drive Controller wieder in einen definierten Zustand gebracht werden.
- Referenzieren der Achse
  Über den `HOMING` Eventeingang des Adapters wird die Referenzierung der Achse gestartet. Dabei fährt die Achse die über den Dateieingang `REF_POSITION` angegebene Endlage an.
- Positionieren
  Über den `SET_POSITION` Eventeingang des Adapters wird die Positionierung der Achse gestartet. Dabei fährt die Achse die über den `POSITION` Dateineingang des Adapters angegebene Endlage an. Da der Pneumatic Rotary Drive Controller keine unterschiedlichen Bewegungsprofile unterstützt, hat der `MOTION_PROFIL` Dateineingang des Adapters keine Funktion.

5.4.2. Linear Motion Controller

Der Linear Motion Controller ermöglicht das Ansteuern einer Zahnriemenachse mit Hilfe des Festo Motorcontrollers CMMS-AS. Der Motorcontroller versorgt den Servomotor, der je nach Achse direkt oder über ein Getriebe mit der Zahnriemenachse verbunden ist. Die Kommunikation mit dem Motorcontroller erfolgt über das CANopen Protokoll. Das dazu notwendige CAN Interface wird im nächsten Abschnitt genauer

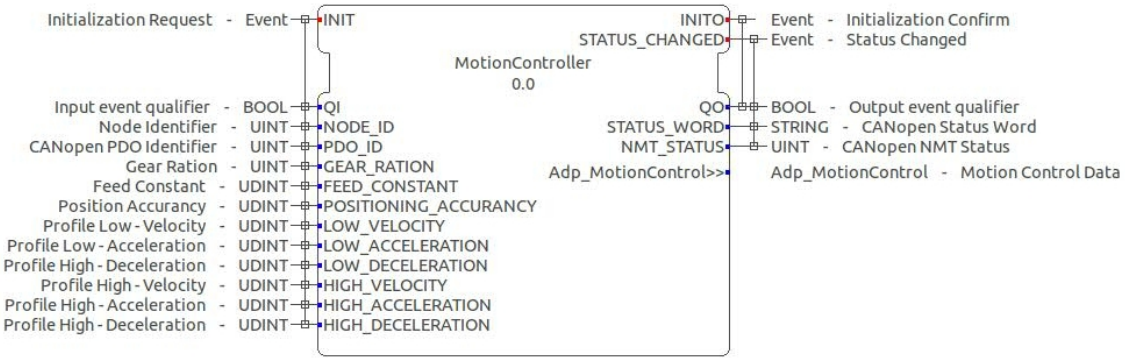


Abbildung 5.7.: Interface des Motion Controller Funktionsblocks

beschrieben. An den Dateneingängen des Linear Motion Controllers können alle Parameter vorgegeben werden, die für den Betrieb notwendig sind. Mit Hilfe der Parameter lassen sich die CANopen Knoten-ID und die CANopen PDO-ID festlegen. Zusätzlich benötigt der Controller die Getriebeübersetzung, die Vorschubkonstante der Zahnriemenachse und die Postioniergenauigkeit. Wie bereits beim AS/RS Controller erwähnt, unterstützt das RBG zwei Fahrprofile, die hier über die Eingänge für ,,LOW" und ,,HIGH" angegeben werden können. Dabei kann für jedes der beiden Profile die Maximalgeschwindigkeit, die Beschleunigung und die Verzögerung definiert werden. Während der Initialisierung wird der Motorcontroller mit den angegebenen Parametern konfiguriert. Der Status des Motorcontrollers steht über die Datenausgänge ,,STATUS_WORD" und ,,NMT_STATUS" zur Verfügung. Wie auch der Pneumatic Rotory Drive Controller stellt der Linear Motion Controller seine Funktionen über den Motion Control Adapter der übergeordneten Ebene zur Verfügung. Diese sind:

- Einschalten der Achsen
  Durch das Einschalten der Achse wird die Positionsregelung der Achse aktiviert. Je nach Ausführung wird zusätzlich die Bremse gelöst (vertikale Achse). Die Achse wird über den `ON` Eventeingang des Adapters eingeschaltet.
- Ausschalten der Achsen
  Beim Ausschalten der Achse wir der Motor stromlos geschaltet. Dadurch wird die Haltebremse der vertikalen Achse aktiviert. Die anderen Achsen lassen sich in diesem Zustand frei bewegen. Die Achse wird über den `OFF` Eventeingang des Adapters ausgeschaltet.
- Zurücksetzen
  Im Falle eines Fehlers bzw. in Folge einer Notabschaltung kann über den Eventeingang `RESET` des Adapters der Linear Motion Controller bzw. der Motorcontroller wieder in einen definierten Zustand gebracht werden.

- Referenzieren der Achse
  Über den `HOMING` Eventeingang des Adapters wird die Referenzierung der Achse gestartet. Dabei fährt die Achse die konfigurierte Endlage an.
- Positionieren
  Über den `SET_POSITION` Eventeingang des Adapters wird die Positionierung der Achse gestartet. Dabei fährt die Achse die über den `POSITION` Dateineingang des Adapters angegeben Position an. Das zu verwendende Fahrprofil (LOW oder HIGH) wird über den `MOTION_PROFIL` Dateineingang des Adapters angegeben.

5.5. CAN Interface

Dieser Abschnitt beschäftigt sich mit dem entwickelten CAN Interface, das für die Kommunikation zwischen dem Linear Motion Controller und dem verwendeten Motorcontroller notwendig ist. In 4DIAC standen noch keine dementsprechenden Funktionsblöcke zur Verfügung, welche die Verwendung der CAN-Schnittstelle ermöglichten. Aus diesem Grund musste ein CAN-Interface entwickelt werden.

5.5.1. Struktur des CAN Interface

Mit Hilfe des entwickelten CAN Interface sollen im wesentlichen nur CANopen Nachrichten übertragen werden. Aus diesem Grund wurde der CANopen Standard verwendet, um die notwendigen Anforderungen an die zu entwickelnden CAN Kommunikationfunktionsblöcke abzuleiten. Wie in Abschnitt 2.3.3 beschrieben, gibt es unterschiedliche CANopen Nachrichtentypen. Diese lassen sich im wesentlichen in drei Klassen einteilen:

- Senden und Empfangen (SDO),
- nur Senden (Receive-PDO) und
- nur Empfangen (Transmit-PDO).

Aufbauend auf diesen drei Klassen wurde versucht eine möglichst einfache universell einsetzbare CAN Kommunikationsschicht zu entwickeln. Beim Design wurde das Publish/Subscriber Prinzip der Kommunikationsblöcke für Ethernet als Vorbild genommen. Im Gegensatz zu Ethernet besitzt CAN jedoch kein System, das dem Socket System von Ethernet bei einer TCP/IP Verbindung zugeordnet werden kann. Wenn also mehrere FB an die selbe Node-ID senden und auf eine Bestätigung warten, kann anhand der empfangen Nachricht nicht festgestellt werden, für welchen FB die Antwort bestimmt ist. Geht z.B. die erste Antwort verloren, so könnte die zweite dem ersten FB zugeordnet werden. Von Seiten CANopen ist eine solche Übertragungsart aber nicht vorgesehen. Es muss immer garantiert werden, dass das nächste SDO erst gesendet wird, wenn das zuvor gesendet SDO verarbeitet und bestätigt wurde.

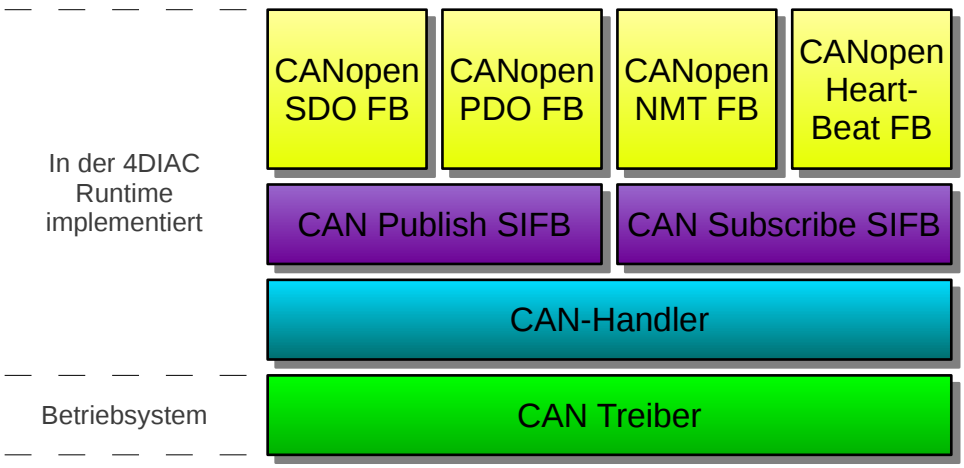


Abbildung 5.8.: Struktur des CAN Interface

Aus diesem Grund kann immer nur ein CAN Subscriber auf eine Node-ID registriert sein und dessen Nachrichten empfangen. In Abbildung 5.8 ist die Struktur das CAN Interface ersichtlich. Die einzelnen Schichten werden nun im weiteren genauer erklärt.

5.5.2. CAN-Handler

Der CAN-Handler bildet die unterste Ebene der CAN Kommunikation und setzt auf auf den CAN Treiber des Betriebssystems auf. Er ist zuständig für die Konfiguration der CAN-Schnittstelle und stellt die notwendigen Funktionen den übergeordneten CAN-SIFB zur Verfügung. Diese lassen sich in drei Gruppen einteilen:

- Funktionen für das Senden von Daten
- Funktionen für das Empfangen von Daten
- Funktionen für das Konvertieren von Daten

Da die Datengröße eines CAN-Telegramms 64-Bit beträgt, wurde als Datentyp String verwendet. Forte unterstützt zwar prinzipiell 64-Bit Integer Datentypen, aber nicht auf allen Steuerungen. String bietet zusätzlich den Vorteil das voran eingegeben Nullen erhalten bleiben. Dies macht das Verarbeiten der Daten in Bezug auf CANopen einfacher.

Für das Senden von Daten wird dem CAN-Handler die ID des Empfängerknotens und die Daten übergeben. Der CAN-Handler versucht nun die Daten am CAN-Bus abzusetzen. Ob der Sendevorgang erfolgreich war, meldet der CAN-Handler dem jeweiligen SIFB mit einem passenden Rückgabewert. Der CAN-Handler hat keinen Mechanismus, welcher den Sendevorgang wiederholt falls die Übertragung scheitert. Ein Sendebuffer ist ebenso nicht implementiert.

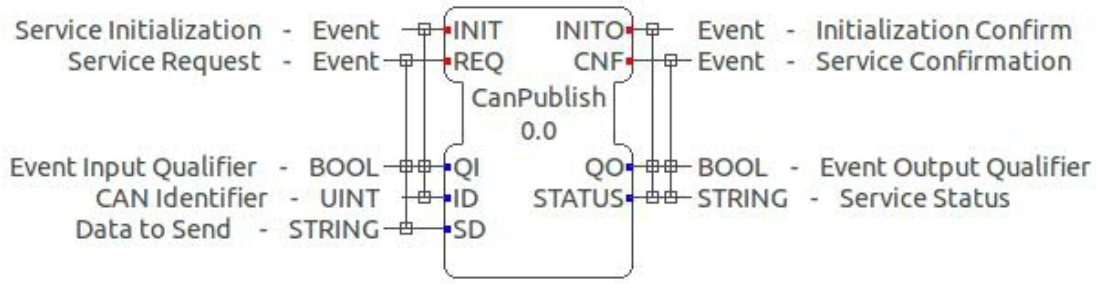


Abbildung 5.9.: Interface des CAN Publish SIFB

Um Daten empfangen zu können, muss als erstes eine Verbindung zum CAN-Handler aufgebaut werden. Dabei registriert sich der jeweilige SIFB beim CAN-Handler mit einem CAN Identifier. Dadurch weiß der CAN-Handler welchen SIFB er aufrufen muss, wenn er eine Nachricht mit der angegebenen CAN ID empfängt. Es kann sich jedoch immer nur ein SIFB auf die gleiche CAN ID registrieren. War die Registrierung erfolgreich, so gibt der CAN-Handler eine Connection ID an den SIFB zurück. Sobald der CAN-Handler eine Nachricht mit der passenden CAN ID empfängt, ruft dieser den jeweils registrierten SIFB auf. Ist für diese CAN ID kein SIFB registriert, so verwirft der CAN-Handler die Nachricht. Der SIFB kann die empfangenen Daten mit Hilfe der bei der Registrierung erhaltenen Connection ID vom CAN-Handler anfordern. Um einen anderen SIFB den Empfang von Daten auf der gleichen CAN ID zu ermöglichen, muss der bereits registrierte SIFB die Verbindung zum CAN-Handler schließen. Erst dann kann sich ein neuer SIFB auf die gleiche CAN ID registrieren.

5.5.3. CAN Service Interface Function Blocks

Die beiden CAN Kommunikationsfunktionsblöcke, CAN Publish FB und CAN Subscribe FB, bilden die vom CAN-Handler zur Verfügung gestellten Funktionen in Form eines SIFB ab. Bei der Umsetzung wurde darauf geachtet möglichst konform zum Standard IEC 61499 zu sein.

5.5.3.1. CAN-Publish SIFB

Mit Hilfe des CAN-Publish SIFB können Telegramme über den CAN Bus gesendet werden. Abbildung 5.9 zeigt das Interface des SIFB. Der Funktionsblock stellt folgende Transaktionen zur Verfügung:

INIT+: Ein INIT+ wird durch ein INIT-Event und durch ein ,,True" am ,,Event Input Qualifier" Eingang (QI) signalisiert. Während des INIT+ Events wird der CAN Identifier (CAN ID) in den FB übernommen. Dabei wird überprüft

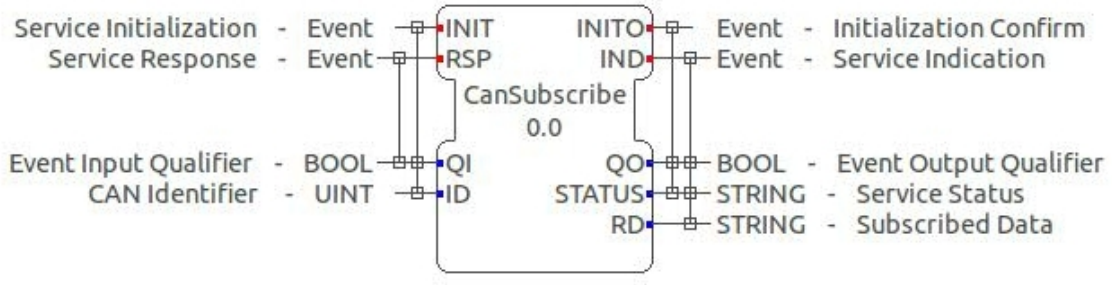


Abbildung 5.10.: CAN Subscribe SIFB

ob die CAN ID im gültigen Wertebereich von 0000h - 7FFFh liegt (11Bit Identifier). Standardmäßig werden nicht gesetzte Eingänge mit dem Wert Null initialisiert. Da Null eine gültige CAN ID darstellt und zusätzlich noch die höchste Priorität besitzt, wurde der Initalwert auf 8000h gesetzt. Dadurch kann sichergestellt werden, dass falls das Setzen des Einganges vergessen wird, der FB nicht initialisiert wird. Zusätzlich zur Überprüfung der CAN ID wird auch eine lose Verbindung zum CAN-Handler aufgebaut. Lose deswegen, da sich der SIFB für das Senden eines Telegramms nicht beim CAN-Handler registrieren muss. Diese Verbindung könnte auch erst kurz vor dem Senden aufgebaut werden. Da jedoch der CAN-Handler erst bei der ersten Verwendung der CAN Schnittstelle initialisiert und konfiguriert wird, ist es sinnvoll dies bereits während der Initialisierung zu machen. Ob die Initialisierung erfolgreich war oder nicht, wird mit einen INITO+ bzw. INITO- gekennzeichnet.

INIT-: Während des INIT- Events wird die CAN ID und die Verbindung zum CAN-Handler zurückgesetzt.

REQ: Bei einem REQ-Event werden die Daten am SD-Eingang an den CAN-Handler übergeben. Dieser konvertiert den Daten-String in ein CAN-Telegramm und setzt diese am CAN-Bus ab.

5.5.3.2. CAN-Subscribe SIFB

Mit Hilfe des CAN-Subscribe SIFB können am CAN Bus abgesetzte Telegramme empfangen werden. Abbildung 5.10 zeigt das Interface des SIFB. Der Funktionsblock stellt folgende Transaktionen zur Verfügung:

INIT+: Während des INIT+ Events wird wie beim CAN-Publish SIFB die CAN ID auf ihre Gültigkeit überprüft. Der Initialwert liegt ebenfalls außerhalb des gültigen Wertebereichs, um feststellen zu können ob das Setzen bzw. das Verbinden des Ausganges vergessen wurde. Zusätzlich zur Überprüfung der CAN ID wird auch eine feste Verbindung zum CAN-Handler aufgebaut. Dazu muss sich der CAN-Subscribe SIFB im Gegensatz zum CAN-Publish

SIFB beim CAN-Handler mit der CAN ID registrieren. Die Registrierung ist notwendig um später CAN Telegramme empfangen zu können (siehe 5.5.2). Eine erfolgreiche Registrierung wird durch Rückgabe einer gültigen Connection ID signalisiert. Ist bereits ein anderer SIFB auf die gleichen CAN ID registriert, so schlägt der Verbindungsaufbau fehl. Ob die Initialisierung erfolgreich war oder nicht, wird mit einen INITO+ bzw. INITO- Event signalisiert.

INIT-: Mit einem INIT- Event wird der CAN-Subscribe SIFB dazu veranlasst die bestehende Verbindung zum CAN-Handler zu schließen. Dazu wird die beim Verbindungsaufbau erstellte Connection ID benötigt. Mit Hilfe dieser ID kann der CAN-Handler die Verbindung identifizieren und schließen. Nach Beendigung der Abarbeitung sendet der FB in jeden Fall ein INITO- Event.

IND: Mit einem IND Event signalisiert der CAN-Subscribe SIFB das ein CAN Telegramm für die bei der Initialisierung angegebene CAN ID empfangen wurde. Das CAN Telegramm wird vom CAN-Handler in einen String konvertiert und steht über den RD-Ausgang des SIFB zur Verfügung.

5.5.4. CANopen Function Blocks

In diesem Abschnitt werden alle für die Umsetzung der Steuerung notwendigen CANopen Funktionsblöcke erklärt. Die CANopen FBs können als eine Art erweiterte SIFB bezeichnet werden. Sie bauen auf den CAN SIFB auf und besitzen zusätzlich weitere integrierte Funktionen. Aus diesem Grund wurden sie in Form von Composite Function Blocks (CFB) realisiert. Im Gegensatz zu den CAN SIFBs besitzen die CANopen FBs keine CAN-ID sondern eine Node-ID, da bei CANopen prinzipiell der Knoten adressiert wird. Aus der Node-ID lassen sich dann die jeweiligen CAN-IDs ableiten. Die wichtigsten FB sind dabei jene für SDOs und PDOs. Die beiden anderen FB für NMT und Heartbeat/Bootup sind der Vollständigkeit halber angegeben und werden daher nur oberflächlich behandelt.

5.5.4.1. CANopen Service Data Object FB

Mit Hilfe des CANopen-SDO FB können SDOs über den CAN Bus geschickt und empfangen werden. Abbildung 5.11 zeigt das Interface des FBs. Wie bereits oben beschrieben wird bei CANopen der Knoten adressiert. Die jeweiligen CAN Adressen werden während der Initialisierung berechnet.

Wie man aus Abbildung 5.12 entnehmen kann, wird in Folge der Initialisierung zwar der CAN-Publish SIFB initialisiert nicht aber der CAN-Subscribe SIFB. Dies ist deswegen, da das implementierte Design es ermöglicht mehrere CANopen SDO FB mit der gleichen Node-ID hintereinander zuschalten. Da intern die Node-ID in eine CAN-ID umgerechnet wird, würde das bedeuten, dass alle FB die gleiche CAN ID besitzen. Würde bereits während der Initialisierung auch der CAN-Subscribe FB

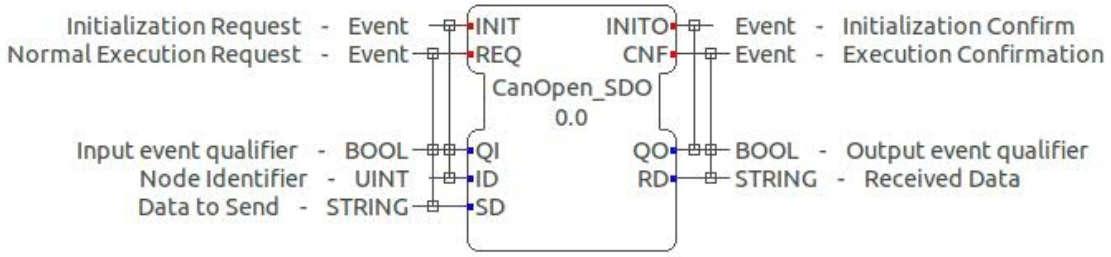


Abbildung 5.11.: Interface des CANopen SDO FB

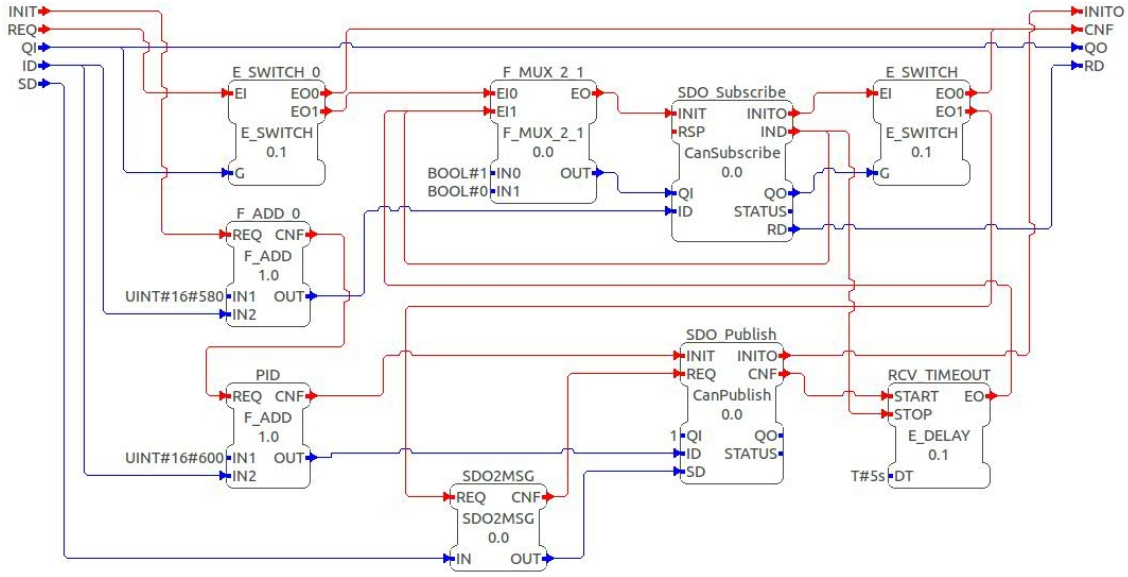


Abbildung 5.12.: Composite Network des CANopen SDO FB

initialisiert, so würde der erste eine feste Verbindung zum CAN-Handler aufbauen und alle nachfolgenden CAN Subscribe FB könnten nicht mehr initialisiert werden. Da der CAN Publish SIFB nur eine lose Verbindung aufbaut können die Publish SIFB initialisiert werden.

Beim Erhalt eines REQ-Events wird als erstes der CAN-Subscribe SIFB initialisiert. Dies erfolgt mit Hilfe des Multiplexer FB F_MUX_2_1, welcher die Funktion eines SR Flip-Flop übernimmt. Im Unterschied zu einem SR Flip-Flop schickt ein Multiplexer in jeden Fall einen Ausgangsevent, auch wenn sich der Zustand am Ausgang des FBs nicht ändert. Würde ein SR Flip-Flop anstelle des Multiplexers verwendet, würde dieses nach einer gescheiterten Initialisierung des CAN Subscribe SIFB die erneute Verwendung des FBs verhindern.

Der CAN-Subscribe SIFB baut im Zuge der Initialisierung eine feste Verbindung zum CAN-Handler auf. War der Verbindungsaufbau erfolgreich, leitet der CAN Publish SIFB die Datenübertragung ein. Die Gegenstelle bestätigt den Erhalt der Daten. Dieses CAN Telegramm wird vom CAN-Handler empfangen und an den CAN Sub-

Tabelle 5.1.: Syntax für die Eingabe von SDOs

Kommando	Index	Subindex	Trennzeichen	Daten	Beschreibung
?	XXXX	SU			Lesezugriff
=	XXXX	SU	:	WW	Schreibzugriff mit 8 Bit Daten
=	XXXX	SU	:	WWWW	Schreibzugriff mit 16 Bit Daten
=	XXXX	SU	:	WWWWWWWW	Schreibzugriff mit 32 Bit Daten

scribe SIFB weitergegeben. Dieser deinitialisiert sich darauf hin mit Hilfe des Multiplexer FBs selbst und baut die Verbindung zum CAN-Handler ab. Für den Fall das die Gegenstelle nicht antwortet, wurde ein Empfangstimeout mit Hilfe eines Timer FBs realisiert. Dieser wird nach dem Absetzen der Daten durch den CAN-Publish SIFB gestartet und nach Erhalt der Antwort durch den CAN-Subscribe gestoppt. Im Falle eines Timeouts wird der CAN-Subscribe FB deinitialisiert.

Da das SDO eine Protokoll-Stuktur besitzt, wurde für die einfachere Eingabe der CANopen Objekte, die von Festo im Kapitel „Simulation von SDO-Zugriffen über RS232" [3] eingeführte Syntax verwendet. Diese hat den Vorteil, dass bei der Eingabe der Daten keine Rücksicht auf das Domain Protocol genommen werden muss. Dieses wird anhand der eingegeben Daten anschließend automatisch erstellt. Die verwendete Syntax beginnt mit einem Kommandozeichen, mit welchem zwischen Lese- und Schreibzugriff unterschieden wird. Anschließend folgt der Index gefolgt vom Subindex. Im Falle eines Schreibzugriffs werden die Parameterdaten durch einen Doppelpunkt vom Subindex getrennt. Tabelle 5.1 zeigt die zulässigen Syntaxen, wobei jeder Buchstabe für eine hexadezimale Zahl steht.

Für die Konvertierung der eingeführten Syntax in ein CAN Telegramm wird der SDO2MSG FB verwendet. Dieser erstellt aus der jeweiligen Eingabe das dazugehörige Domain Protokoll.

5.5.4.2. CANopen Process Data Object FB

CANopen definiert jeweils 4 PDOs für das Senden und Empfangen von Prozessdaten. Mit Hilfe des CANopen TPDO1 FBs lässt sich das erste Transmit PDO konfigurieren und empfangen. Der CANopen TPDO1 FB ist wie auch der CANopen SDO FB als CFB realisiert worden. Das Interface des FBs ist in Abbildung 5.13 ersichtlich.

Auf Grund der Größe des Composite Networks wird auf eine Abbildung verzichtet. Der CANopen TPDO1 übernimmt auch die Konfiguration des Controllers, aus diesem Grund besteht sein Composite Network aus einer aneinander Kettung von CANopen SDO FB mit abschließenden CAN-Subribe SIFB. Mit Hilfe der SDO FB

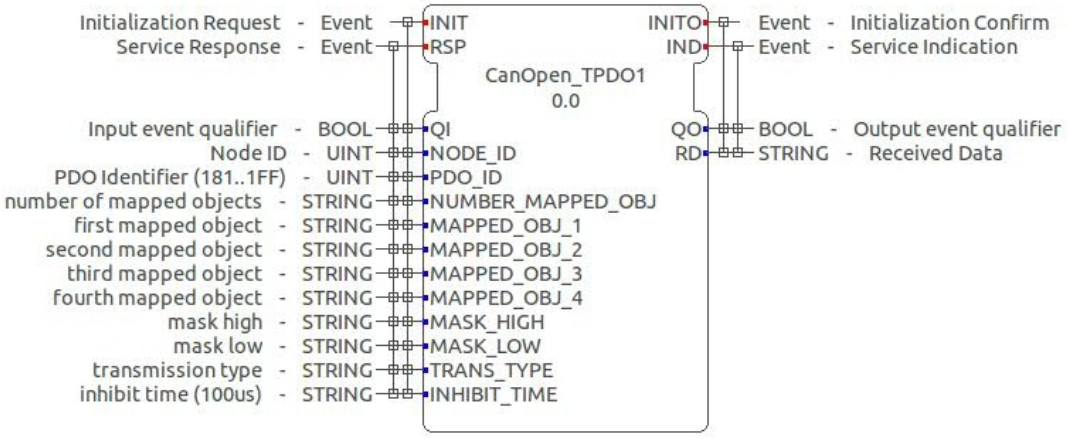


Abbildung 5.13.: Interface des CANopen TPDO1 FB

wird der Controller während der Initialisierung konfiguriert. Der CAN-Subscribe SIFB wird abschließend auf den PDO Identifier eingestellt und ermöglicht so das Empfangen der PDOs.

Für die leichtere Konfiguration muss an den Eingängen des FBs nur mehr der Parameter des jeweiligen Objekts eingegeben werden. Die Voranstellung des Index sowie Subindex wird automatisch durchgeführt. Wird ein Parameter nicht gesetzt, so wird der zugehörige SDO FB einfach übersprungen. Dadurch braucht man nur die benötigten Parameter setzen und kann die anderen frei lassen. Bei der Eingabe der Parameter ist jedoch auf deren Länge zu achten. Es besteht also ein Unterschied zwischen "0001" und "01" als Parameterwert. Obwohl beide vom Wert gleich sind, wird auf Grund der unterschiedlichen Längen ein unterschiedliches Domain Protokoll erzeugt.

5.5.4.3. CANopen Network Management FB

Mit Hilfe des CANopen NMT kann der NMT Status eines Knotens geändert werden. Abbildung 5.14 zeigt das Interface des FBs. Im Zuge der Umsetzung der LLC wurden die beiden Befehle ,,Reset Application" und ,,Start Remote Node" verwendet. Der Befehl ,,Reset Application" ist notwendig um den Knoten in einen definierten Zustand zu bringen bevor er konfiguriert wird. Mit Hilfe des Befehls ,,Start Remote Node" kann das Senden und Empfangen von PDOs aktiviert werden.

5.5.4.4. CANopen Heartbeat/Bootup FB

Mit Hilfe des CANopen Heartbeat/Bootup FBs können die Knoten im CANopen Netzwerk überwacht werden. Diese schicken bei Aktivierung des Error Control Protokolls in regelmäßigen Abständen sogenannte HeartBeat-Nachrichten. Zusätzlich

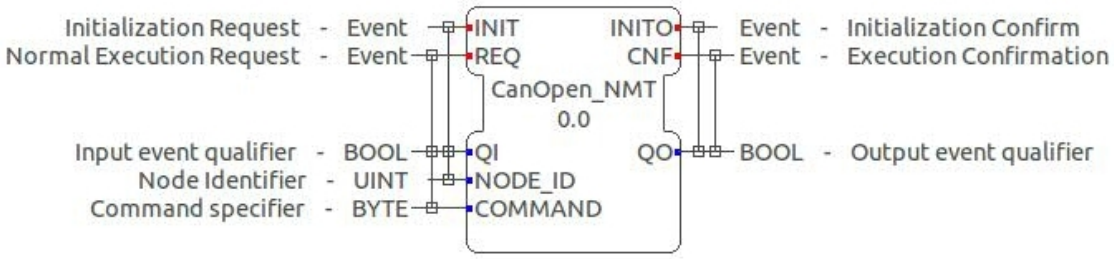


Abbildung 5.14.: Interface des CANopen NMT FB

Abbildung 5.15.: Interface des CANopen Heartbeat/Bootup FB

schicken alle Knoten beim Einschalten bzw. nach einem erfolgreichen Reset eine Bootup Nachricht. Mit dieser wird signalisiert, wann der Knoten für die Konfiguration bereit ist. Abbildung 5.15 zeigt das Interface des Heartbeat/Bootup FBs.

5.6. Testen der Low Level Control

Die entwickelte LLC wurde mit Hilfe der aufgebauten Lagergasse ausführlich getestet. Für den Testbetrieb wurde nur eine Lagerebene eingebaut. Dadurch standen sechs Lagerplätze zur Verfügung. Auf die anderen wurde verzichtet um das Kollisionsrisiko aufgrund von Fehlfunktionen zu minimieren. Um der LLC Befehle vorzugeben wurde ein Testpanel erstellt, welches den Betrieb einer Lagergasse ermöglicht. Dieses Panel ist in Abbildung 5.16 ersichtlich. Im linken Teil des Panels wird der Status der jeweiligen Achse ausgegeben. Im rechten Teil befinden sich die Steuerungselemente mit denen die jeweiligen Befehle an den AS/RS Controller gegeben werden können. Für einen ,,Transport" Befehl müssen in folgender Reihenfolge Quelladresse, Zieladresse und Verweilposition eingegeben werden. Für einen ,,Position" Befehl müssen die Koordinaten der Achsen in angegebener Reihenfolge (w,x,y,z) eingegeben werden. Nach dem erfolgreichen Download des Steuerungsprogrammes werden die Motorcontroller automatisch von der Steuerung konfiguriert. Sind alle Motorcontroller betriebsbereit, kann das RBG mit Hilfe des Panels eingeschaltet werden. Um in weiterer Folge die sechs Lagerplätze und die Übergabestation automatisch anfahren zu können, wurden als erstes die Positionen der Lagerplätze bzw. der Übergabestation eingelernt. Dazu musste das RBG im Homing-Modus ref-

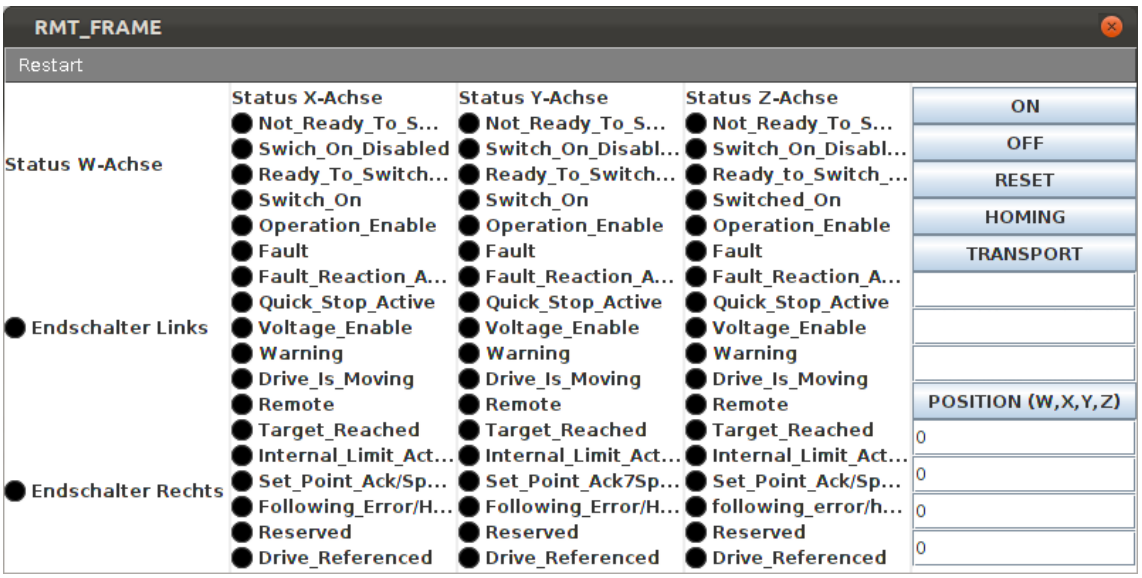


Abbildung 5.16.: Testpanel für den Betrieb einer Lagergasse

erenziert werden. Anschließend konnten im manuellen Modus die sechs Lagerplätze und die Übergabestation angefahren werden. Die Koordinaten wurden in einer Liste im Quellcode des AS/RS Controllers abgelegt. Bei der automatischen Abarbeitung zeigte sich, dass die Koordinaten der Lagerplätze und Übergabestation sehr genau eingelert werden müssen, da sich sonst der Werkstückträger beim Ablegen verkeilen kann. Beim Aufnehmen des Werkstückträgers spielt dies keine so große Rolle, da aufgrund der abgeschrägten Staplergabeln und der Flexibilität des pneumatischen Drehantriebs kleine Ungenauigkeiten ausgeglichen werden können. Das Einlernen der Positionen erfolgt deswegen am besten beim Ablegen eines Werkstückträgers. Durch zusätzliches abschrägen der Zentrierbolzen kann sich der Werkstückträger beim Ablegen leichter zentrieren. Da auch nachträglich kleine Korrekturen an den Koordinaten der Lagerpositionen durchgeführt werden müssen, erwies sich das Ablegen der Koordinaten direkt im Quellcode des AS/RS Controllers als nicht optimal. Bei Änderungen muss jedes mal das gesamte Steuerungsprogramm neu kompiliert und auf die Steuerung übertragen werden.

Nachdem die Koordinaten der Lagerplätze und der Übergabestation im Quellcode des AS/RS Controllers abgelegt sind, kann die Abarbeitung eines Ein-, Aus- bzw. Umlagerungsauftrages automatisch von der LLC durchgeführt werden. Diese konnten erfolgreich getestet werden. Durch die Vorgabe der richtigen Verweilposition steht der Bewegungsablauf für Einzelspiel als auch Doppelspiel zur Verfügung. Die Unterstützung zweier Bewegungsprofile für schnelles und langsames Positionieren ermöglicht es die Abarbeitungszeit für die verschiedenen Aufträge zu vermindern. Da in der momentanen Realisierung der LLC keine Information über den aktuellen Standort der Achsen rückgelesen werden kann, erfolgt das Schwenken der w-Achse immer an einem ,,sicheren" Ort. Würde der aktuelle Standort der Achsen zur Verfü-

gung stehen, so könnte das Schwenken während des Verfahrens der x- und y-Achse durchgeführt werden und der Bewegungsablauf des RBGs weiter optimiert werden.

Aufgrund des fehlenden Fehlermanagements werden momentan nur die Eingangsdaten jedes FBs auf Gültigkeit überprüft, so weit dies möglich ist. Tritt ein solcher Fehler auf, so wird der jeweilige FB nicht abgearbeitet. Eine Fehlerbehandlung bzw. automatische Fehlerbehebung ist momentan nicht implementiert.

5.7. Zusammenfassung

Für die Implementierung der LLC musste als erstes der Funktionsumfang der LLC festgelegt werden. Dazu diente einerseits die im Kapitel 4 erstellte Analyse und andererseits der Stand der Technik als Grundlage. Mit Hilfe der Anwendungsfallanalyse wurden die hardwarenahen Anwendungsfälle identifiziert, die die Grundfunktionen der LLC definieren. Aus dem Stand der Technik wurden die relevanten Strategien ermittelt, die in der LLC umgesetzt werden sollen. Aufbauend auf diesen beiden konnte im weiteren die Struktur der Steuerung festgelegt werden. Im Top-Down Verfahren wurden anschließend alle wichtigen Komponenten der Steuerung vorgestellt. Für die Kommunikation zwischen Steuerung und Motorcontroller wurde unter Berücksichtigung des CANopen Protokolls ein CAN Interface entwickelt. Das CAN Interface besteht im wesentlichen aus drei Ebenen: dem CAN-Handler, den CAN-SIFB und den CANopen-FB. Der CAN-Handler bildet die Schnittstelle zum Betriebssystem. Er ist für die Konfiguration der CAN Schnittstelle zuständig und stellt Funktionen für das Senden und Empfangen von CAN Telegrammen zur Verfügung. Die CAN-SIFB bilden diese beiden Funktionen in Form von SIFB ab. Die CANopen-FB bauen auf den CAN-SIFB auf und ermöglichen das Verwenden des CANopen Protokolls. Für die einfachere Handhabung wurde die von Festo eingeführte Nomenklatur verwendet. Mit Hilfe des mechanischen Aufbaus konnte die LLC erfolgreich getestet werden. Dazu wurde ein Testpanel erstellt, welches einerseits den Status der einzelnen Achsen anzeigt. Andererseits kann mit Hilfe des Panels die verschiedenen Befehle an den AS/RS Controller geschickt werden. Die Abarbeitung eines Ein-, Aus- bzw. Umlagerungsauftrages wird von der LLC durchgeführt und konnte mit Hilfe der aufgebauten Lagergasse demonstriert werden. Aufgrund des fehlenden Fehlermanagement wird bei einem Fehler die Abarbeitung gestoppt, sofern diese erkannt werden können.

6. Ausblick

Da sich diese Arbeit hauptsächlich mit der LLC des HRLs beschäftigt hat, kann aufbauend auf den Erkenntnissen, die im Zuge der LLC Entwicklung gewonnen wurden, ein verteiltes Steuerungskonzept für ein HRL entwickelt werden. Dabei kann einerseits die Interaktion des HRLs mit dem PTS untersucht werden, andererseits bietet die Aufteilung der Ein- bzw. Auslagerungsaufträge auf die jeweiligen Lagergassen ein weiteres interessantes Forschungsgebiet. Gerade die Auftragsvergabe ist in verteilten Systemen ein sehr komplexer Prozess, da die einzelnen Informationen über mehrere Steuerungen verteilt sind und dabei lokale als auch globale Ziele mitberücksichtigt werden sollen.

Im Zuge dieser Arbeit wurde nur eine Lagergasse aufgebaut. In einer weiteren Arbeit können nun die restlichen zwei Lagergassen aufgebaut und in Betrieb genommen werden. Dadurch stehen dann drei Lagergassen zur Verfügung, welche eine anspruchsvolle Testumgebung für verteilt gesteuerte Lagersysteme darstellen. Mit Hilfe dieser Testumgebung kann dann ein entwickeltes verteiltes Steuerungskonzept sehr leicht getestet werden.

Abseits der Lagertechnik stellt das aufgebaute System einen Portalroboter dar, der mit Hilfe des CANopen Protokolls angesprochen werden kann. Da im Zuge dieser Arbeit ein eigenes CAN Interface erstellt wurde, kann aufbauend auf diesen Informationen das CAN bzw. das CANopen Protokoll in den bestehenden Kommunikationslayer des 4DIAC Projektes übertragen werden. Einige Bemühungen hinsichtlich der Implementierung von CAN in 4DIAC wurden bereits von der CiA unternommen. Dabei wurden einige Erkenntnisse, die im Zuge dieser Arbeit gewonnen wurden, berücksichtigt. Da in dem entwickelten CAN Interface nur jene der in CANopen spezifizierten Funktionen implementiert wurden, welche für die Ansteuerung des RBGs notwendig sind, können in weiteren Arbeiten der gesamte Funktionsumfang von CANopen in 4DIAC implementiert werden.

7. Zusammenfassung

Ziel dieser Arbeit, war die Erweiterung des bestehenden PTS, welches sich im Odo Struger Labor der TU Wien befindet, um ein HRL. Dazu mussten die bestehenden Anlagenteile dementsprechend erweitert bzw. geändert werden, um möglichst alle Anforderungen eines modernen HRLs erfüllen zu können. In einem ersten Schritt wurden alle Anforderungen an das HRL erhoben. Diese ergaben sich aus den Anforderungen der Institutsmitarbeiter, durch den Stand der Technik und durch den Aufbau der bestehenden Anlage. Auf dieser Grundlage konnte nun eine genaue Konstruktion des HRLs erstellt werden. Zur Verifikation der mechanischen Konstruktion wurde anschließend eine Lagergasse aufgebaut und an das PTS angebunden.

Im weiteren wurden eine LLC für die Ansteuerung des mechanischen Aufbaus entwickelt. Um die auf die LLC aufbauende HLC zu entlasten, wurde versucht einige Funktionen der HLC in die LLC auszulagern. Dazu wurde eine objektorientierte Analyse eines verteilt gesteuerten Lagersystems durchgeführt. Auf Basis der durch die Analyse gewonnenen Daten konnte anschließend entschieden werden, welche Funktionen in der LLC umgesetzt werden können. Dabei wurde festgestellt, dass die Abarbeitung des Materialflusses für das Ein-, Aus- und Umlagern vollständig durch die LLC übernommen werden kann, da bis auf Quell- und Zieladresse keine weiteren Informationen benötigt werden. Eine Integration weiterer Funktionen könnte jedoch schnell zu einer Überlastung der LLC führen, da diese auch für zeitkritische Funktionen zuständig ist. Die Anforderungen an die LLC ergaben sich nun durch die bei der Analyse des verteilten gesteuerten Lagersystems identifizierten hardwarenahen Anwendungsfälle, durch die relevanten Strategien aus dem Stand der Technik und durch die Wahl der verwendeten Komponenten. Da die verwendeten Motorcontroller, welche für die Ansteuerung der Servomotoren zuständig sind, das CANopen Protokoll unterstützen, musste ein CAN Interface in der LLC implementiert werden. In der verwendeten Entwicklungsumgebung standen jedoch noch keine Funktionen für die Kommunikation mittels CAN bzw. CANopen zur Verfügung. Aus diesem Grund mussten dementsprechende Funktionen entwickelt werden, welche eine Kommunikation zwischen LLC und Motorcontrollern ermöglichen.

Abschließend wurde die entwickelte LLC getestet. Die Abarbeitung eines Ein-, Aus- bzw. Umlagerungsauftrages wird vollständig von der LLC durchgeführt und konnte mit Hilfe der aufgebauten Lagergasse erfolgreich demonstriert werden. Nach dem Start eines Auftrages durch die HLC, werden keine weiteren Informationen von der HLC benötigt. Dies führt zu einer Entlastung der HLC. Die entwickelte LLC kann nun als Ausgangspunkt für aufbauende Arbeiten verwendet werden. Durch

die Erweiterung des PTS um ein HRL steht nun ein komplexes verteilt gesteuertes Logistiksystem zur Verfügung um neue Steuerungskonzepte praxisnahe testen zu können.

A. Liste der verwendeten Komponenten

In den nachfolgenden Tabellen sind alle Komponenten für den Aufbau einer Lagergasse angegeben. Der Übersichtlichkeit halber wurden die Komponenten den jeweiligen Achsen zugeordnet. Komponenten die keiner Achse zugeordnet werden können werden in einer separaten Liste angegeben.

A.1. Komponenten der W-Achse

Stk	Benennung	Hersteller
1	Schwenkantrieb DRQD-16-90-PPVI-A-AR-FW	Festo
2	Näherungsschalter SME-8-S-LED-24	Festo
1	Montageplatte für Staplergabeln	TU Wien
1	Montagewinkel für Drehantrieb	TU Wien
1	Staplergabel für Lastaufnahmeachse (links)	TU Wien
1	Staplergabel für Lastaufnahmeachse (rechts)	TU Wien

A.2. Komponenten der X-Achse

Stk	Benennung	Hersteller
2	Zahnriemenachse EGC-80-1070-TB-KF-25H-GK	Festo
1	Verbindungswelle KSK-80-1058	Festo
4	Winkel 8 80x80	Item
1	Schaltfahne SF-EGC-1-80	Festo
10	Clip SMBK-8	Festo
2	Näherungsschalter SIES-8M-PO-24V-K-7,5-OE	Festo
1	Servomotor EMMS-AS-55-S-TM	Festo
1	Motorcontroller CMMS-AS-C4-3A-G2	Festo
1	Getriebe EMGA-60-P-G5-SAS-55	Festo
1	Axialbausatz EAMM-A-L48-60G	Festo
1	Motorleitung NEBM-T1G7-E-5-N-LE7	Festo
1	Encoderleitung NEBM-T1G8-E-5-N-S1G15	Festo

A.3. Komponenten der Y-Achse

Stk	Benennung	Hersteller
1	Zahnriemenachse EGC-80-860-TB-KF-25H-GK	Festo
4	Profilbefestigung MUE-70/80	Festo
1	Schaltfahne SF-EGC-1-80	Festo
10	Clip SMBK-8	Festo
2	Näherungsschalter SIES-8M-PO-24V-K-7,5-OE	Festo
1	Servomotor EMMS-AS-55-S-TMB	Festo
1	Motorcontroller CMMS-AS-C4-3A-G2	Festo
1	Getriebe EMGA-60-P-G5-SAS-55	Festo
1	Axialbausatz EAMM-A-L48-60G	Festo
1	Motorleitung NEBM-T1G7-E-5-N-LE7	Festo
1	Encoderleitung NEBM-T1G8-E-5-N-S1G15	Festo
1	Energiekette Anschlusselement 114.4.12.PZ	igus
24	Glieder Energiekette 114.4.038	igus
1	Montageplatte für vertikale Achse	TU Wien

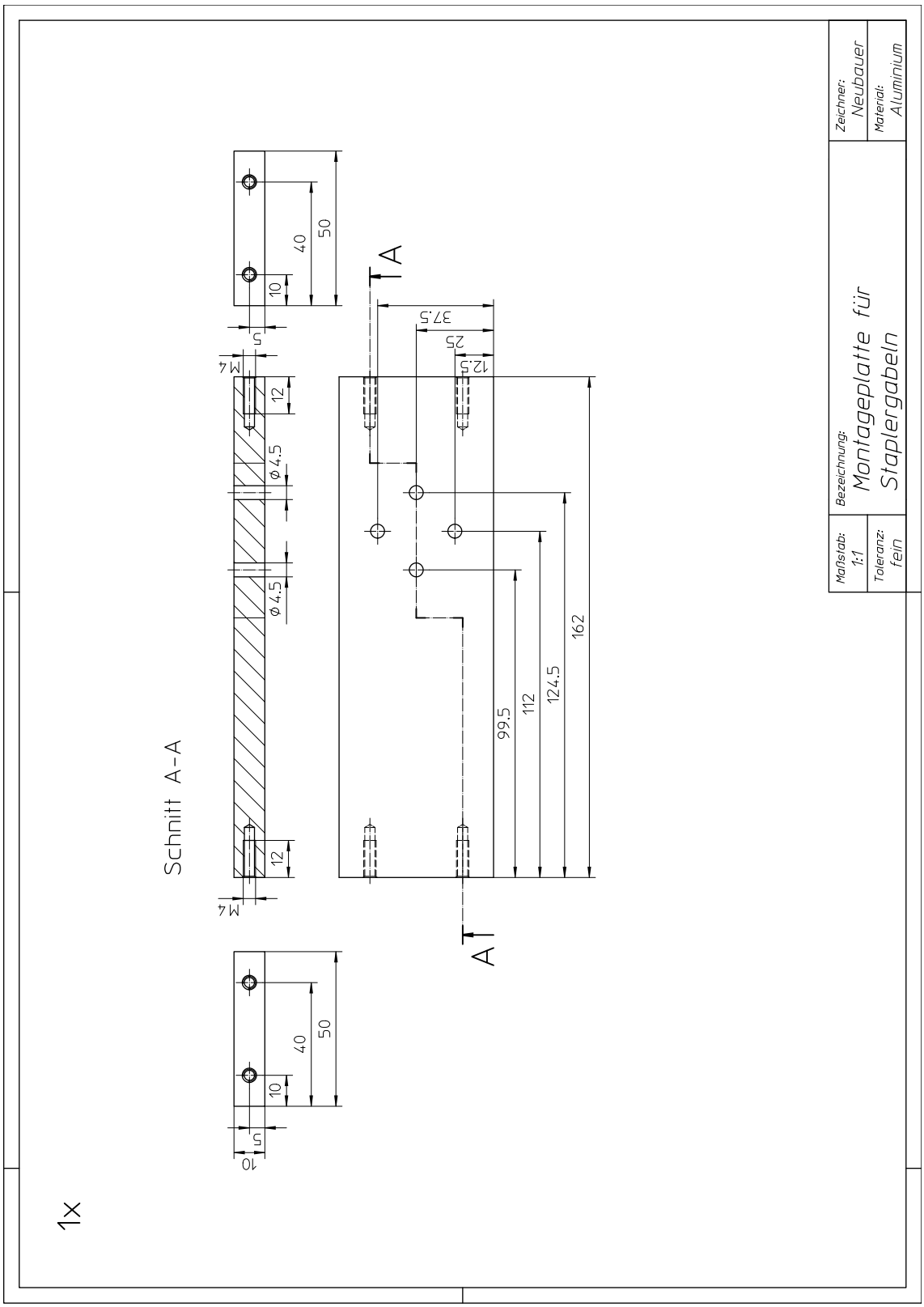
A.4. Komponenten der Z-Achse

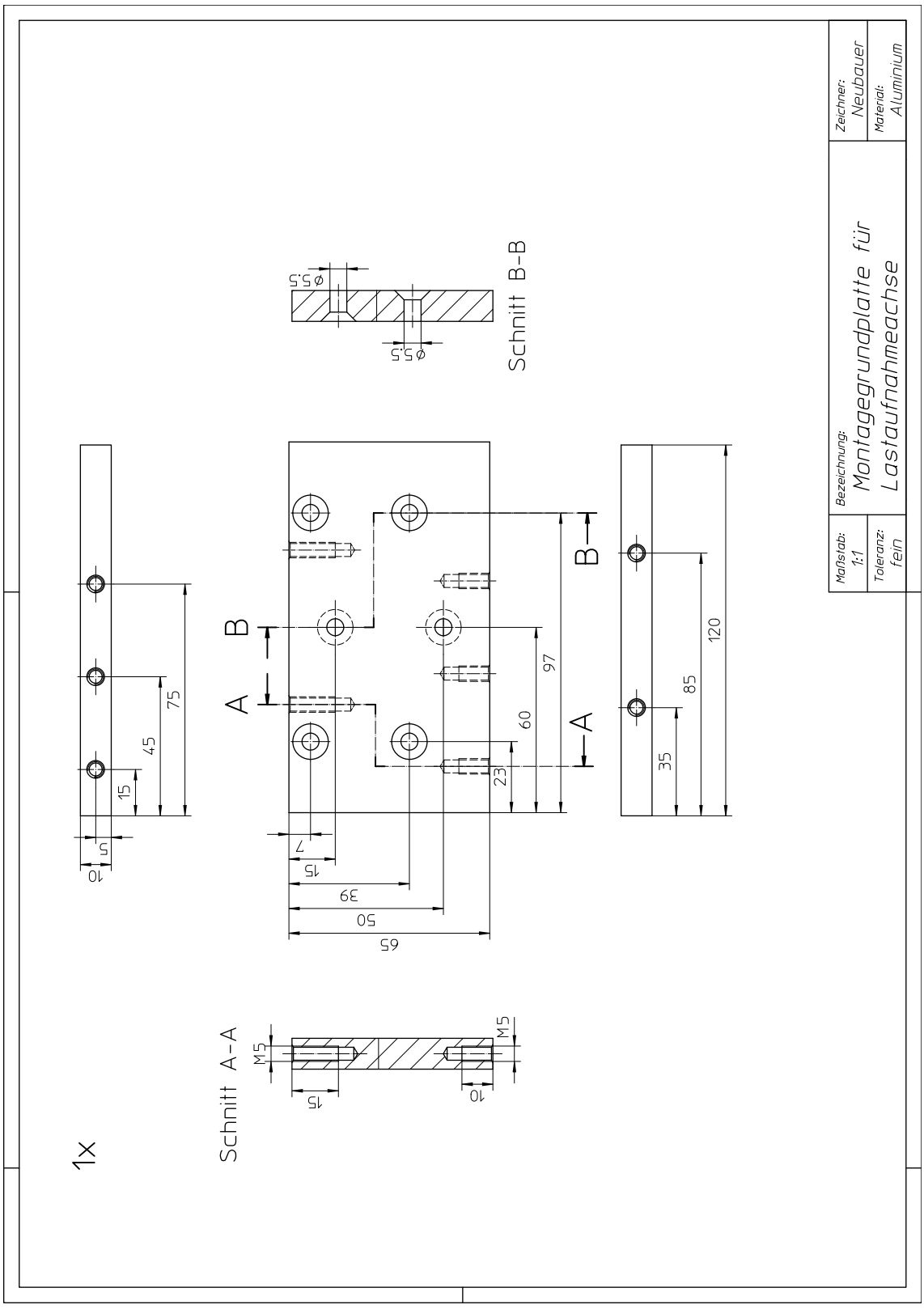
Stk	Benennung	Hersteller
1	Zahnriemenachse DGE-18-80-ZR-LV-RK-KF-GK	Festo
3	Mittenstütze MUP-18/25	Festo
2	Näherungsschalter SME-8-S-LED-24	Festo
1	Servomotor EMMS-AS-55-S-TM	Festo
1	Motorcontroller CMMS-AS-C4-3A-G2	Festo
1	Axialbausatz EAMM-A-G19-55A	Festo
1	Motorleitung NEBM-T1G7-E-10-N-LE7	Festo
1	Encoderleitung NEBM-T1G8-E-10-N-S1G15	Festo
1	Energiekette Anschlusselement 114.3.12.PZ	igus
29	Glieder Energiekette 114.3.028	igus
1	Montagegrundplatte für Lastaufnahmeachse	TU Wien
1	Montageplatte für Energiekette	TU Wien
2	Montageplatte für Lastaufnahmeachse	TU Wien
1	Montagewinkel für Energiekette (LAM)	TU Wien
1	Stützplatte für LAM Montageplatte	TU Wien

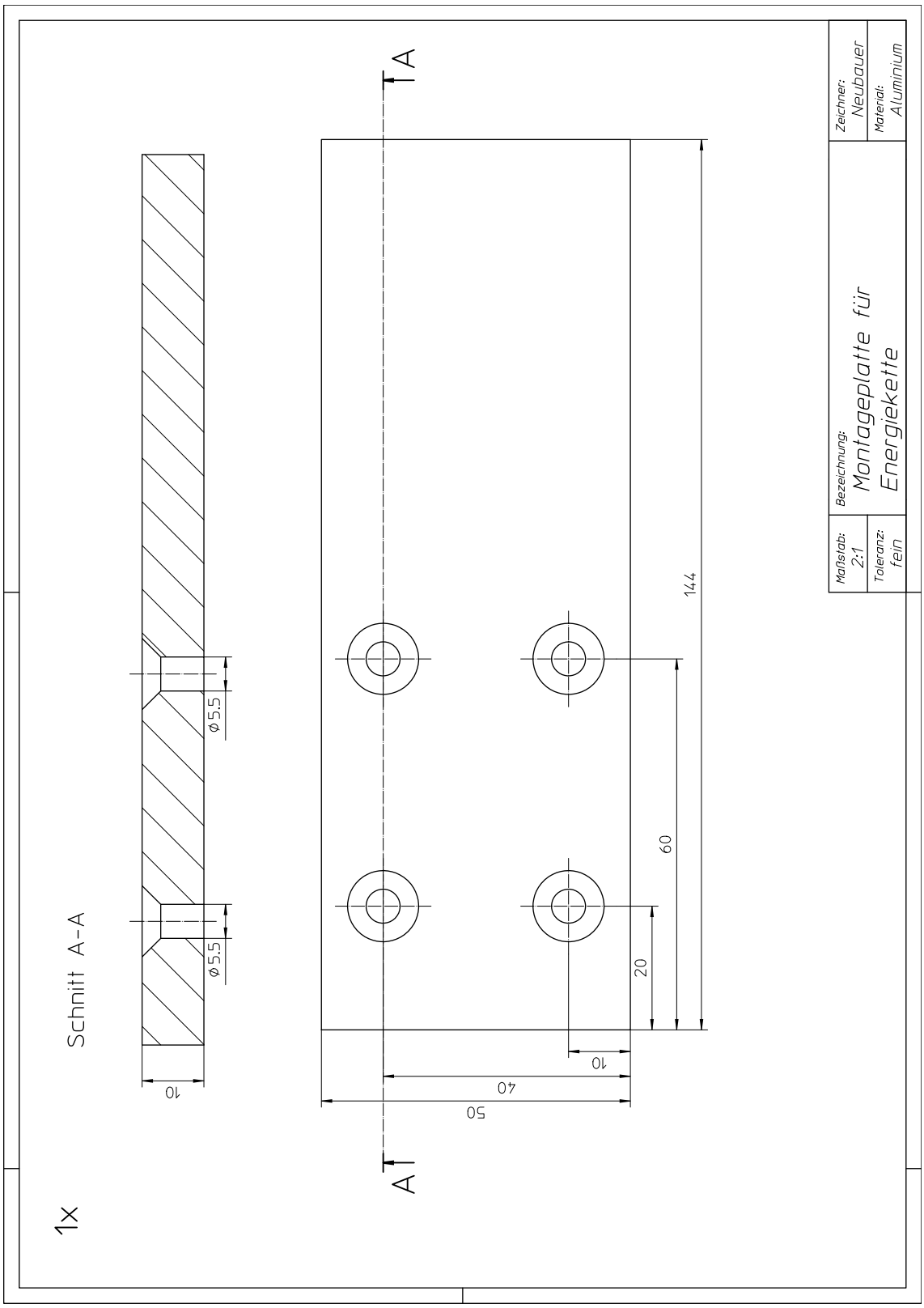
A.5. Diverse Komponenten

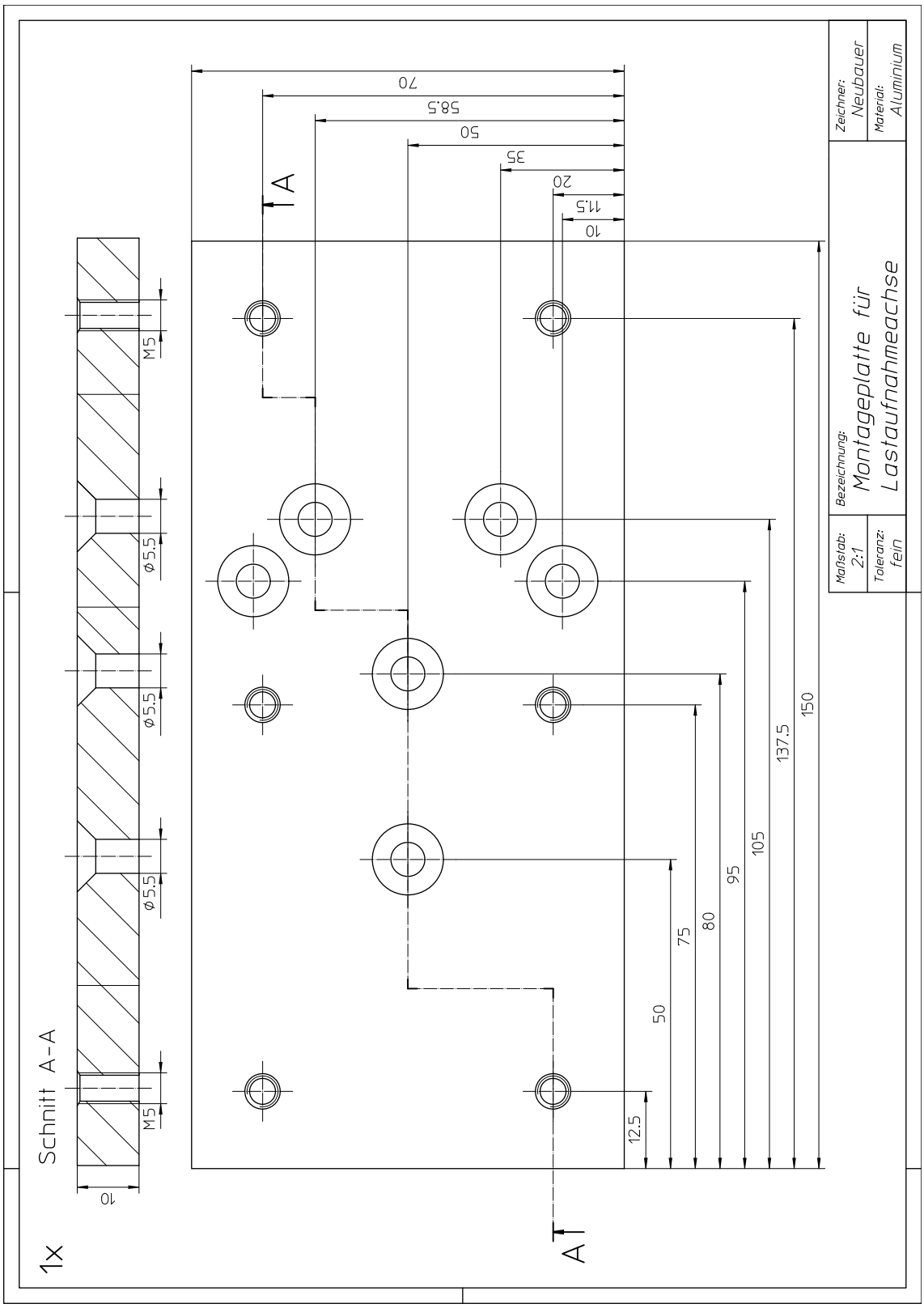
Stk	Benennung	Hersteller
1	Endplatte CPX-EPL-EV	Festo
1	Steuerblock CPX-CEC-C1	Festo
1	Verkettungsblock CPX-GE-EV-S	Festo
1	Anschlussblock CPX-AB-8-KL	Festo
1	Verkettungsblock CPX-GE-EV-S	Festo
1	Eingangsmodul CPX-8DE	Festo
1	Elektronikmodul VMPA1-FB-EMS-8	Festo
1	Anschlussplatte VMPA1-FB-AP-4-1	Festo
1	Endplatte VMPA-FB-EPL-GU	Festo
1	Endplatte VMPA-EPR	Festo
1	Magnetventil VMPA1-M1H-K-PI	Festo
3	Abdeckplatte VMPA1-RP	Festo
6	Regalebene	TU Wien
12	Winkel 8 80x80	Item

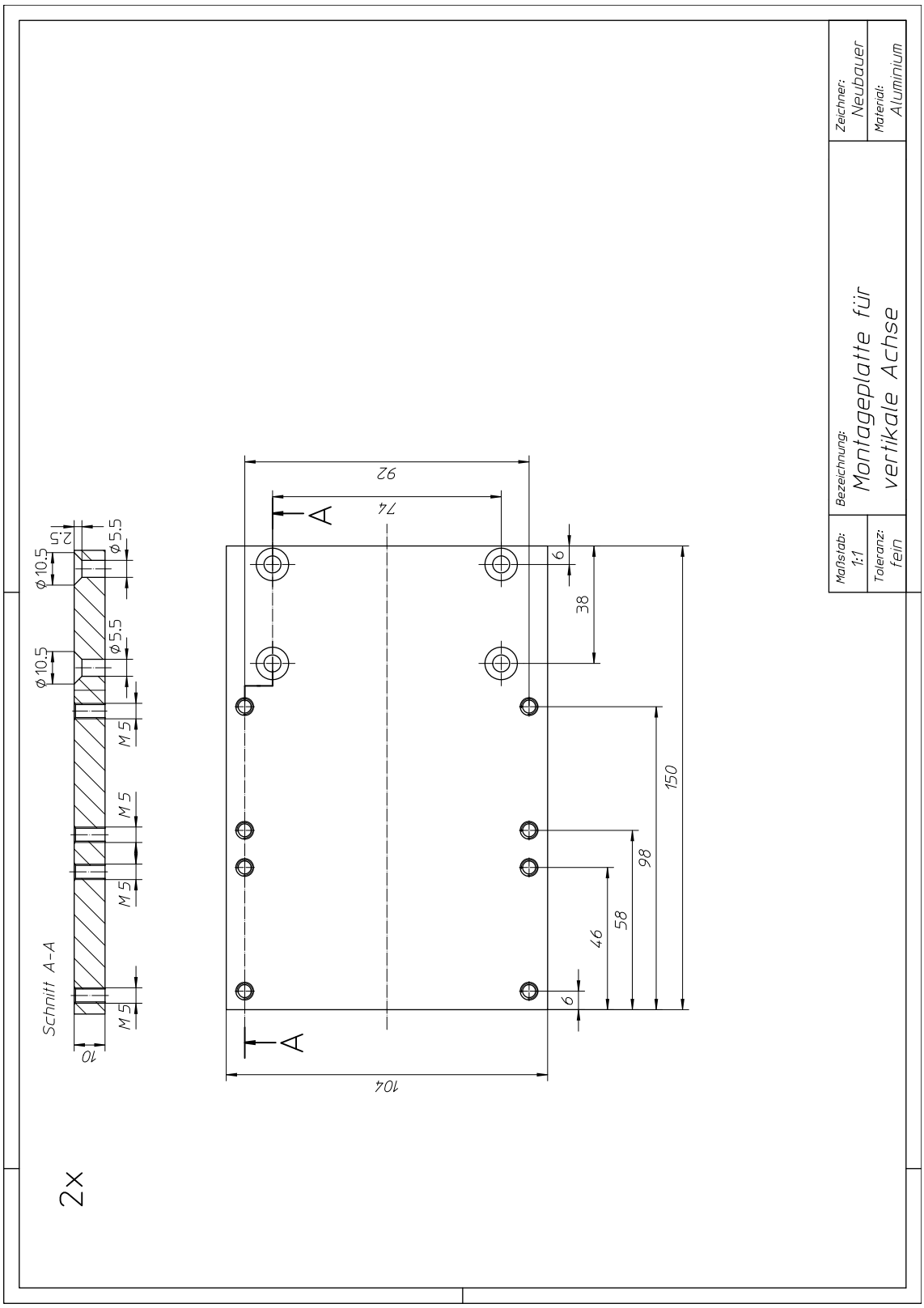
B. Konstruktionszeichnungen

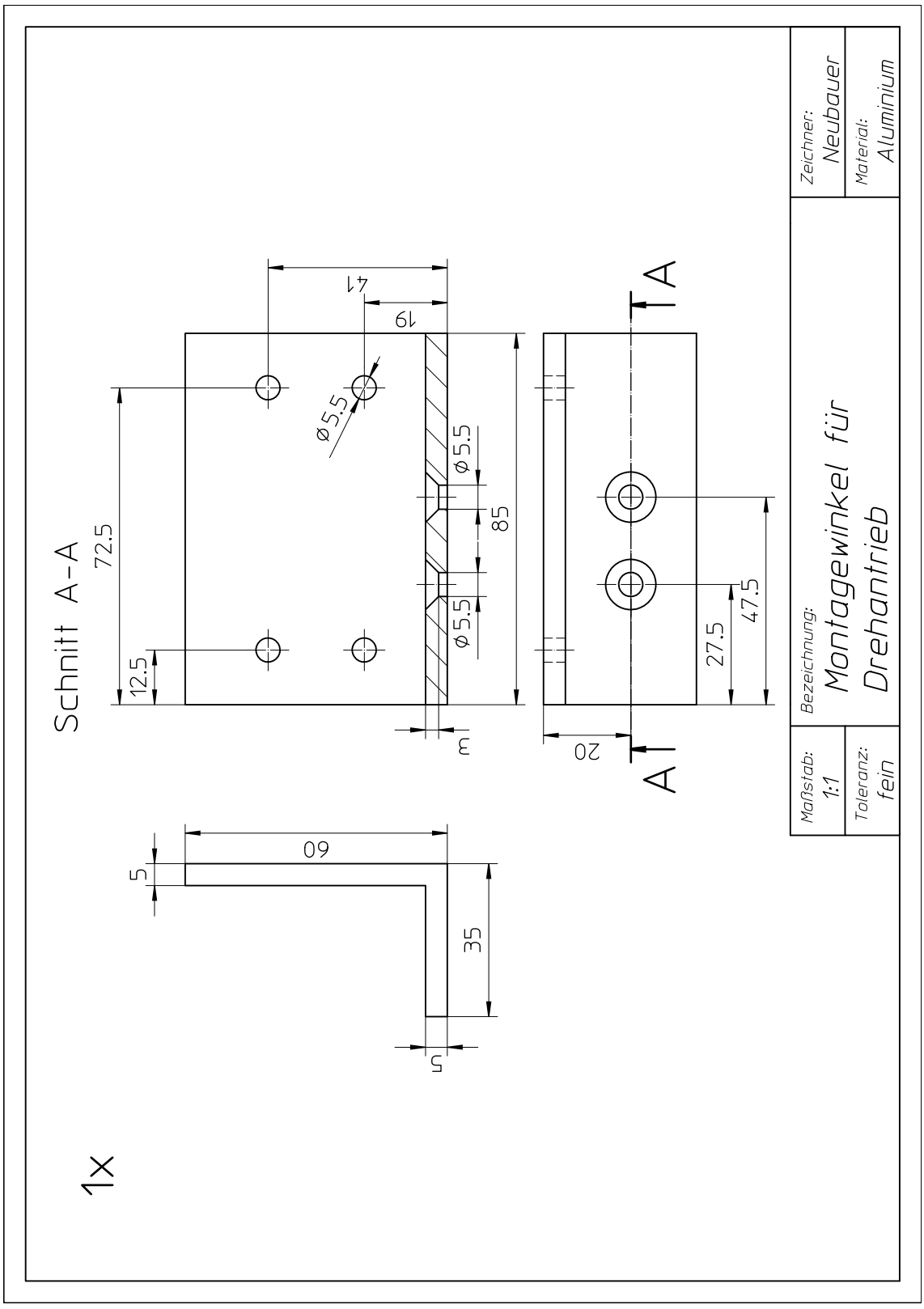


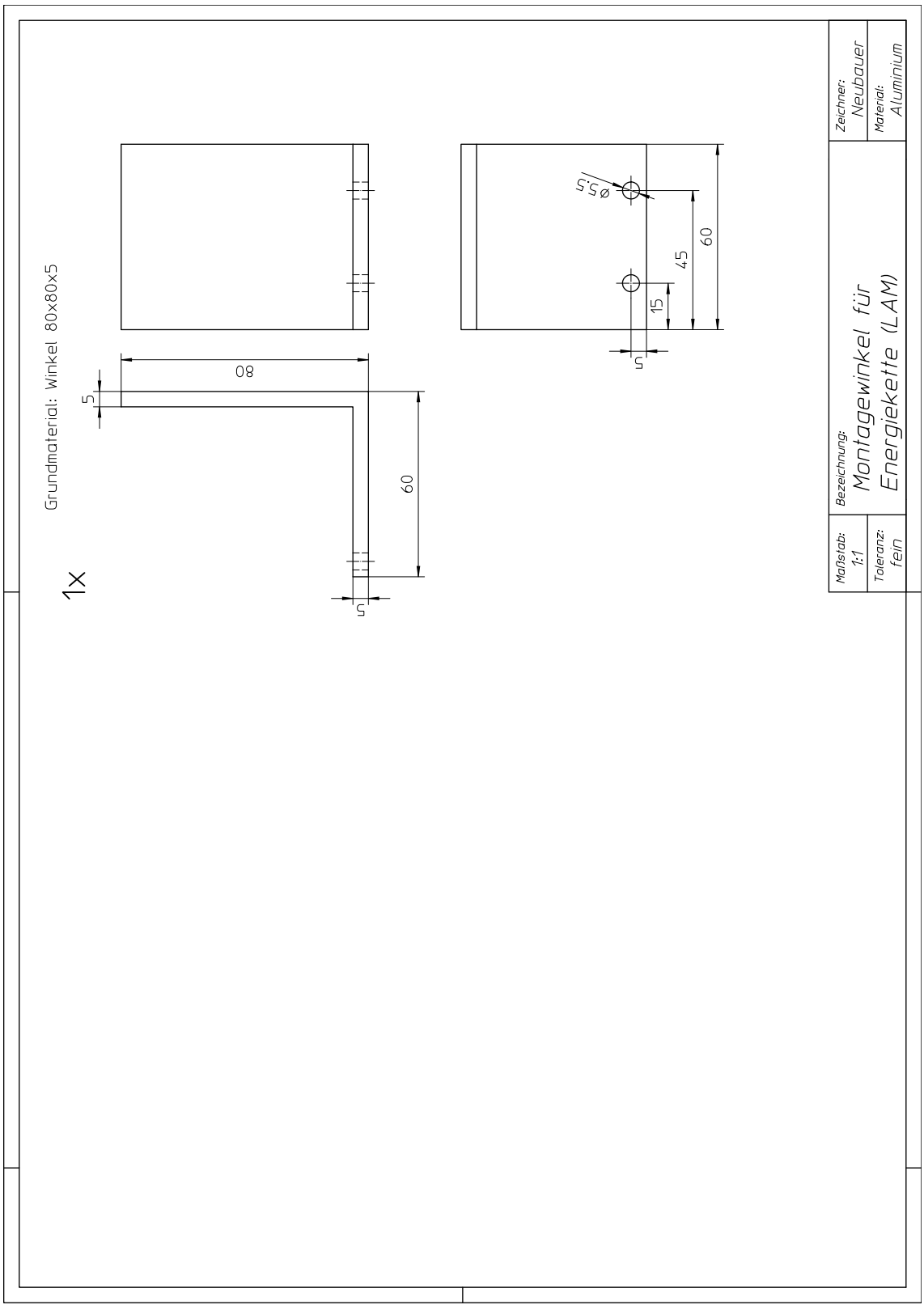

Zeichner: Neubauer
Material: Aluminium
Bezeichnung: Montagegrundplatte für Lastaufnahmeachse
Maßstab: 1:1
Toleranz: fein
Schnitt B-B
Schnitt A-A
A
B
M5
1x
5
10
15
45
75
7
15
39
50
65
23
60
97
35
85
120

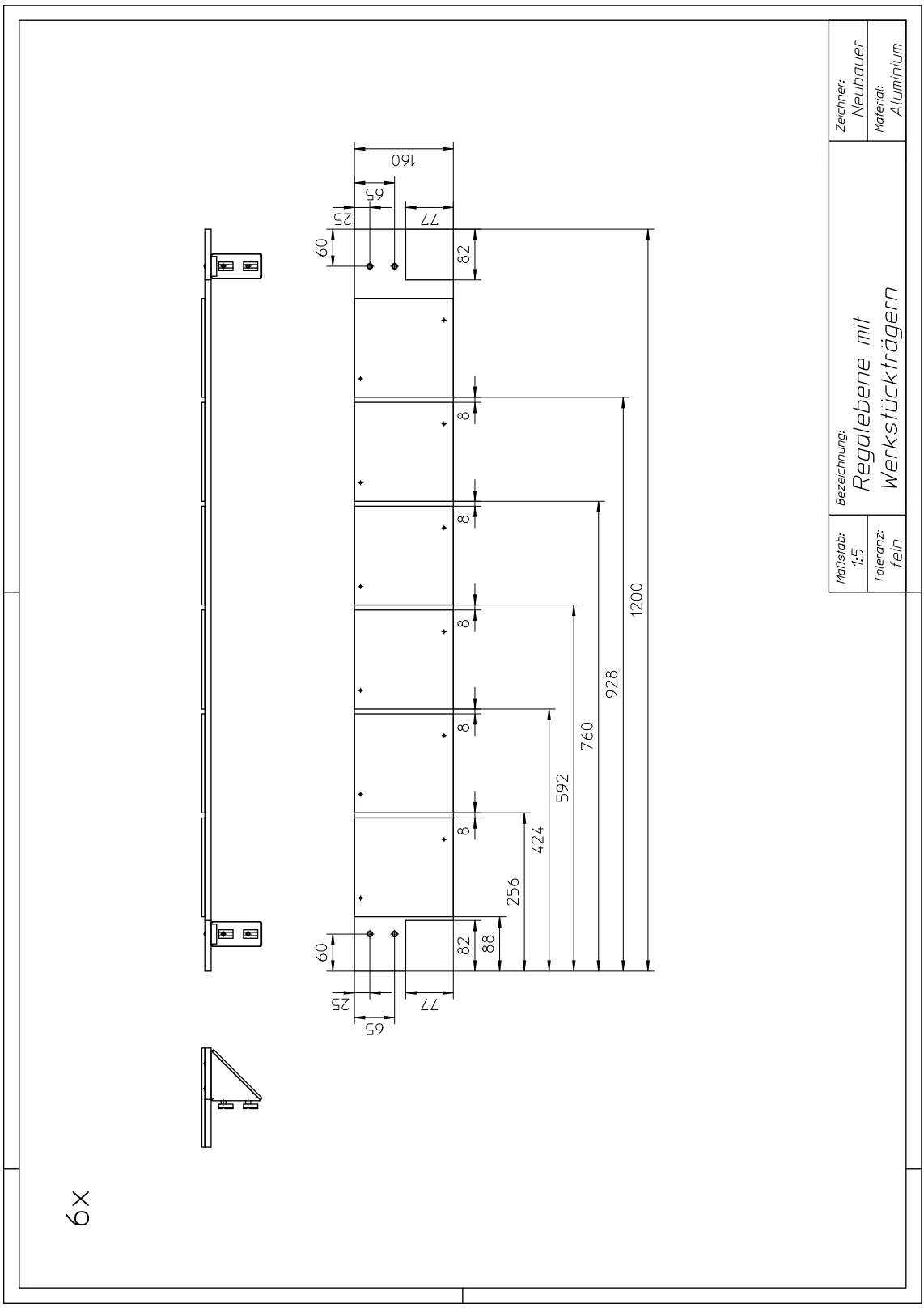

A
A
144
60
20
10
40
50
ø5.5
ø5.5
10
Schnitt A-A
1x
Zeichner: Neubauer
Material: Aluminium
Bezeichnung: Montageplatte für Energiekette
Maßstab: 2:1
Toleranz: fein

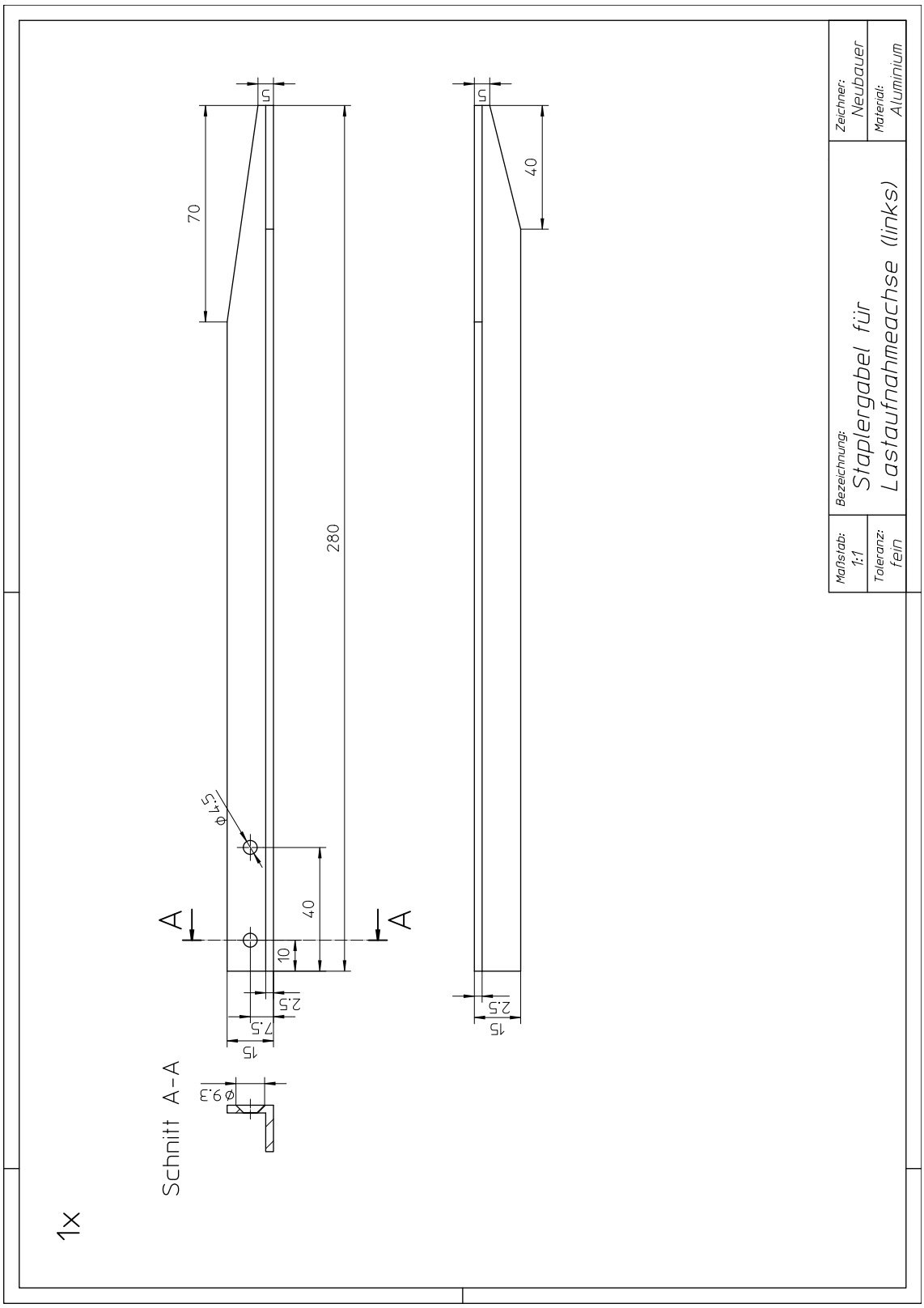

Zeichner: Neubauer
Material: Aluminium
Bezeichnung: Montageplatte für Lastaufnahmeachse
Maßstab: 2:1
Toleranz: fein
70
58.5
50
35
20
11.5
10
A
A
137.5
150
105
95
80
75
50
12.5
M5
ø5.5
ø5.5
ø5.5
M5
10
Schnitt A-A
1x

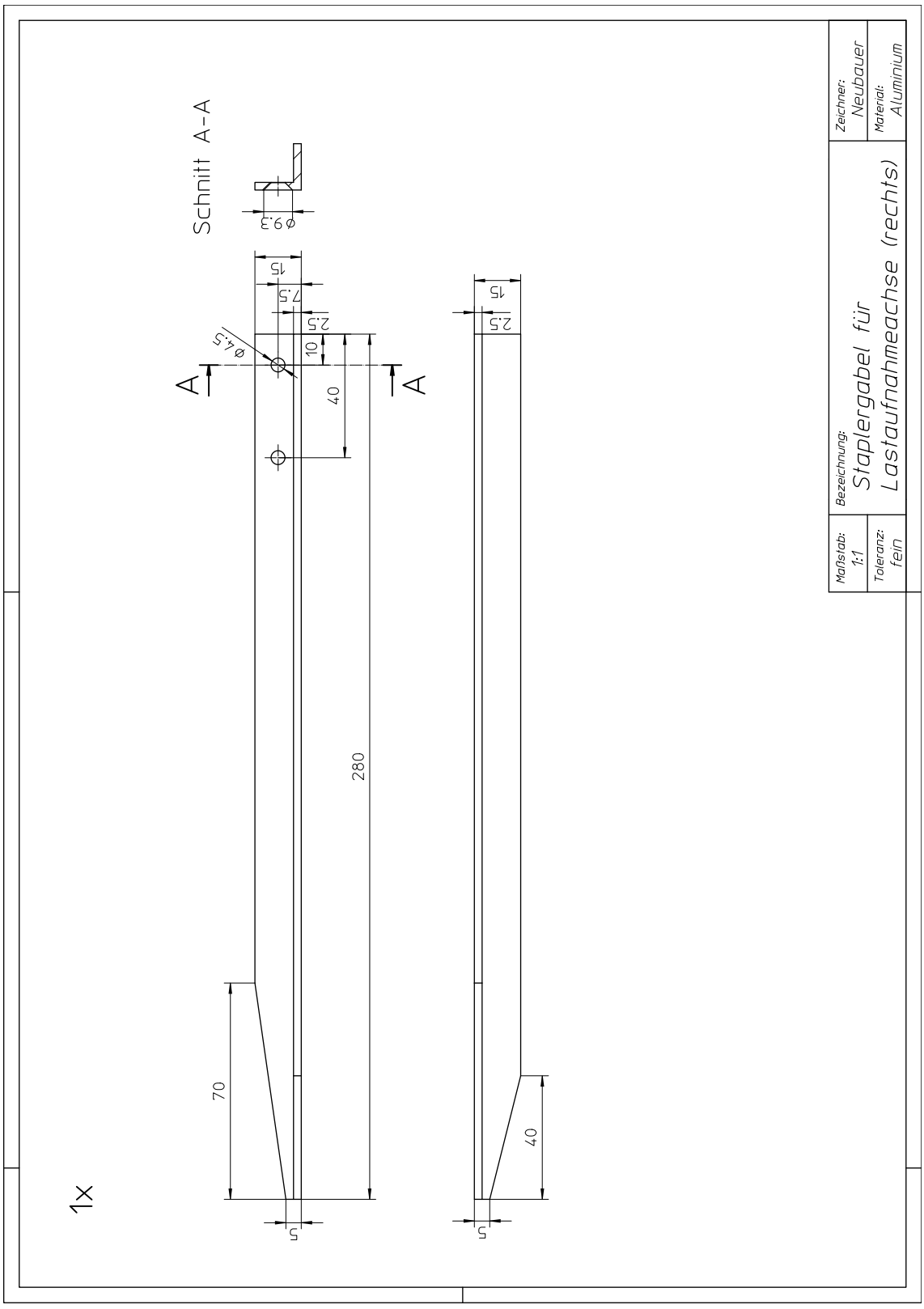

Maßstab: 1:1
Toleranz: fein
Bezeichnung: Montageplatte für vertikale Achse
Zeichner: Neubauer
Material: Aluminium
92
74
A
A
104
6
38
150
98
58
46
6
ϕ10.5
2.5
ϕ5.5
ϕ10.5
ϕ5.5
M5
M5
M5
M5
10
Schnitt A-A
2x

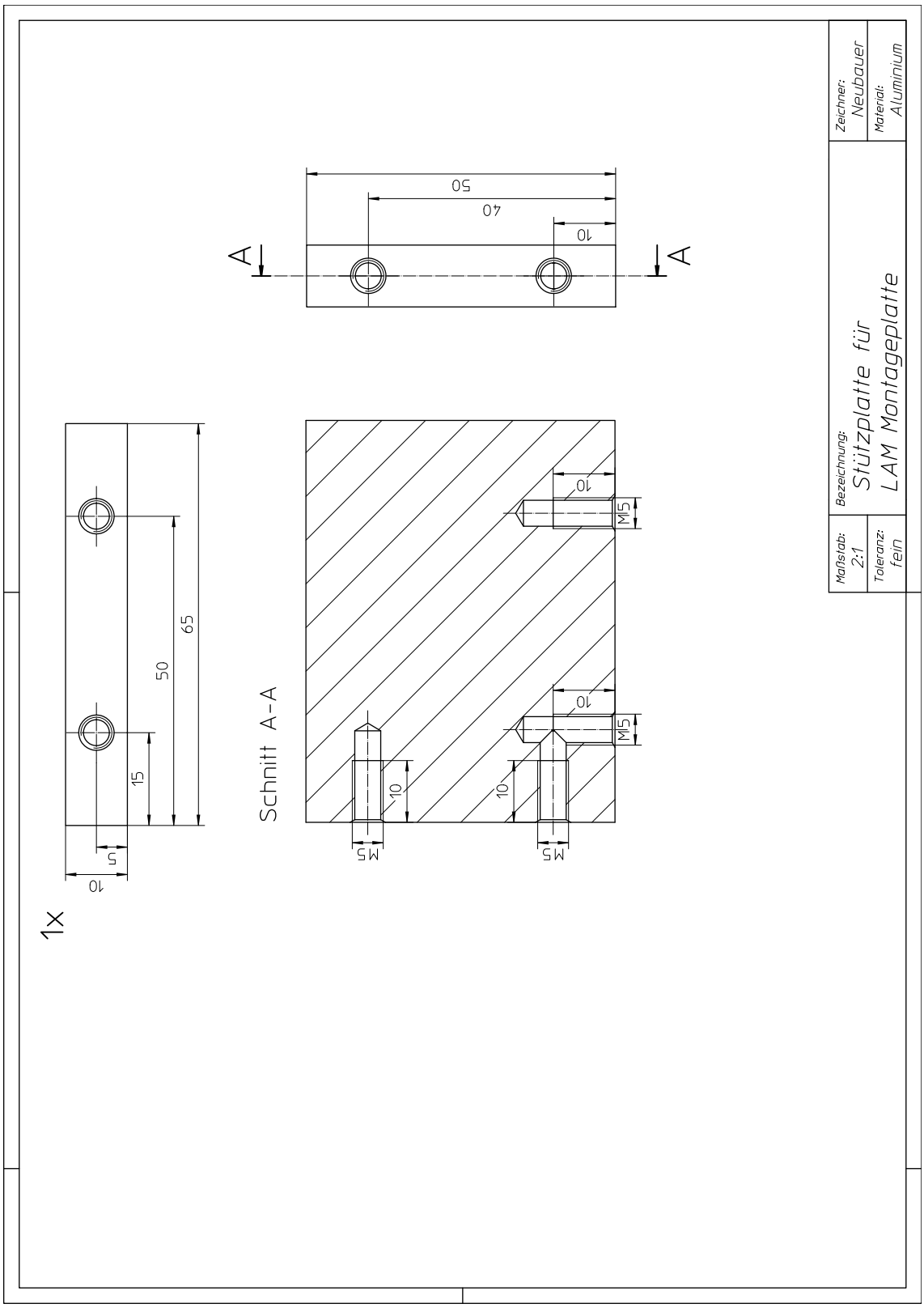

Schnitt A-A
72.5
12.5
41
19
ø5.5
ø5.5
ø5.5
85
5
20
A
A
27.5
47.5
60
5
35
5
1x
Zeichner: Neubauer
Material: Aluminium
Bezeichnung: Montagewinkel für Drehantrieb
Maßstab: 1:1
Toleranz: fein

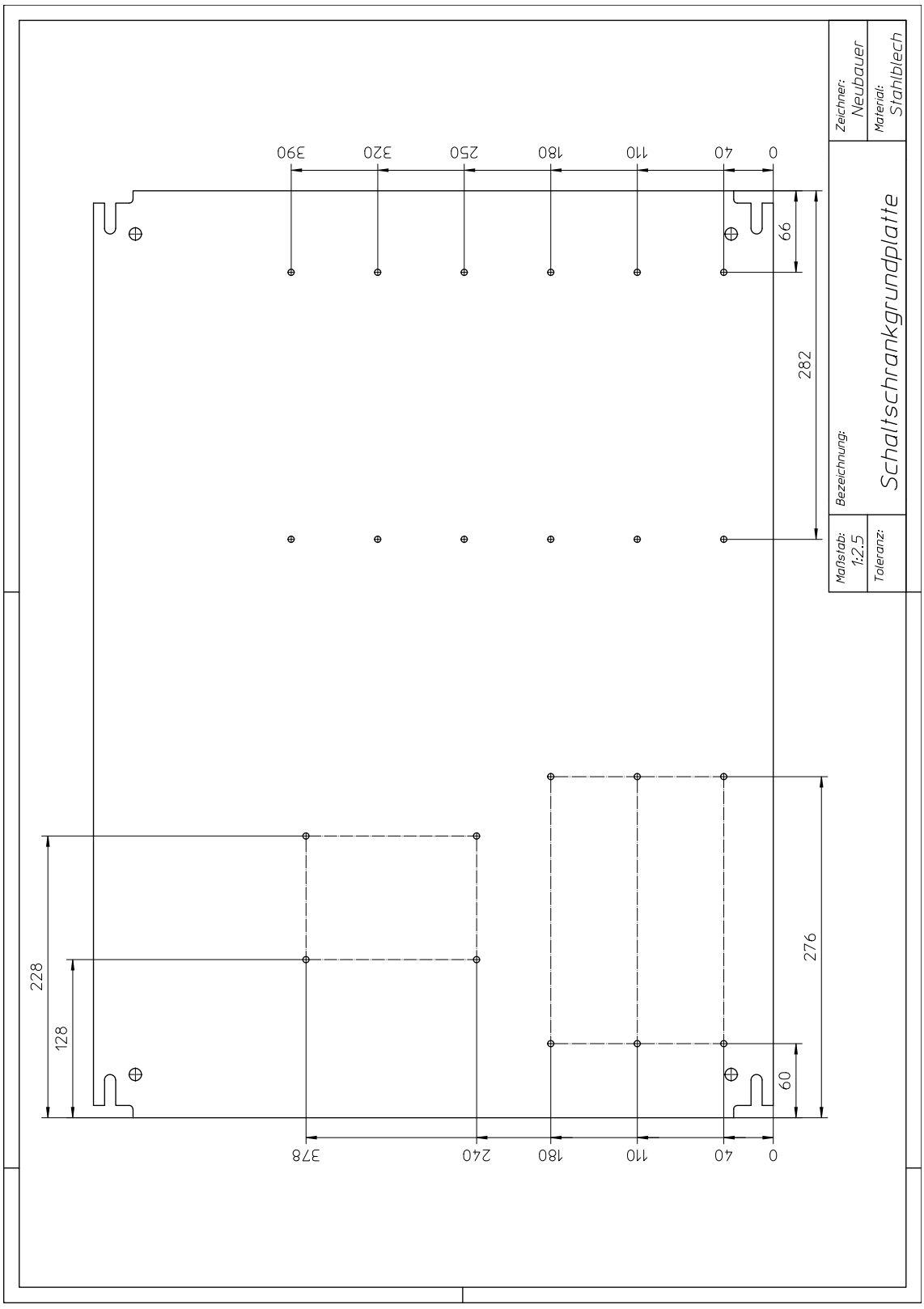

Grundmaterial: Winkel 80x80x5
1X
80
5
60
5
5,5
Ø
60
45
15
5
Zeichner:
Neubauer
Material:
Aluminium
Bezeichnung:
Montagewinkel für
Energiekette (LAM)
Maßstab:
1:1
Toleranz:
fein

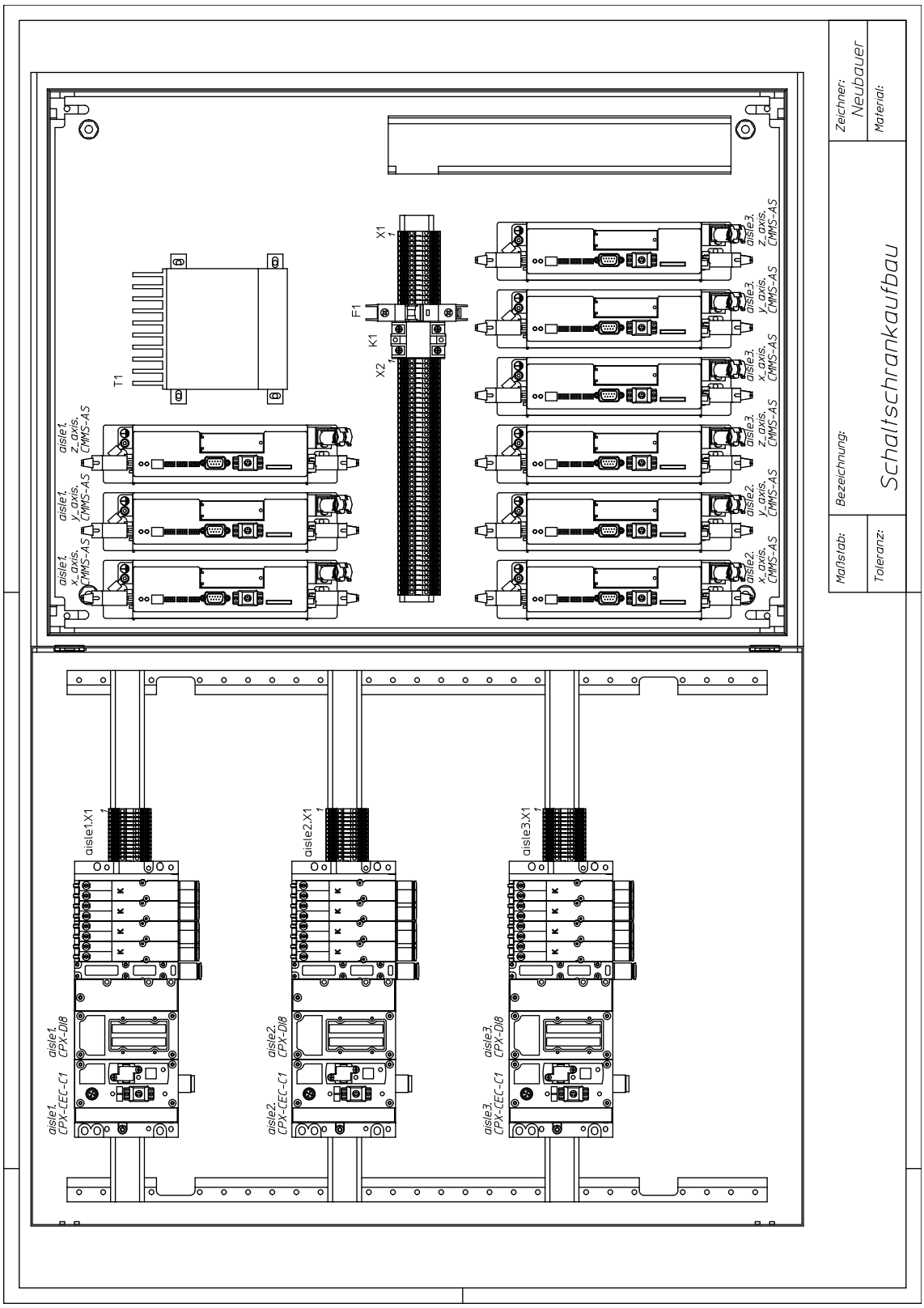

160
65
25
77
60
82
8
8
8
8
8
8
1200
928
760
592
424
256
88
82
60
25
65
77
6x
Zeichner: Neubauer
Material: Aluminium
Bezeichnung: Regalebene mit Werkstückträgern
Maßstab: 1:5
Toleranz: fein

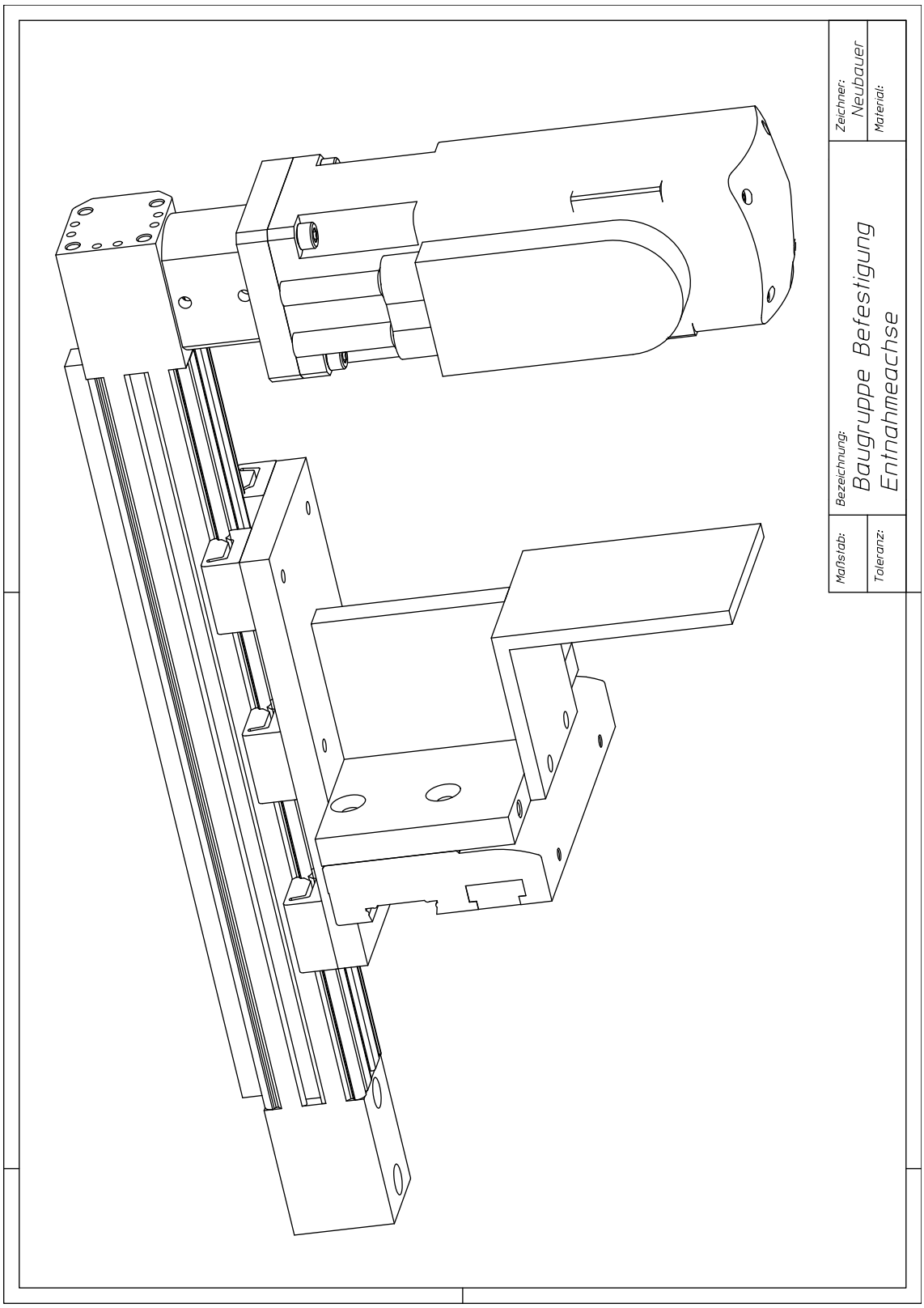

Maßstab: 1:1
Toleranz: fein
Bezeichnung: Staplergabel für Lastaufnahmeachse (links)
Zeichner: Neubauer
Material: Aluminium
1x
Schnitt A-A
A
A
70
280
40
40
5
5
10
2,5
7,5
15
2,5
15
Φ5,4
Φ9,3


Schnitt A-A
Ø9.3
15
7.5
2.5
10
Ø4.5
A
A
40
280
70
5
2.5
15
40
5
1x
Maßstab: 1:1
Toleranz: fein
Bezeichnung: Staplergabel für Lastaufnahmeachse (rechts)
Zeichner: Neubauer
Material: Aluminium


Zeichner: Neubauer
Material: Aluminium
Bezeichnung: Stützplatte für LAM Montageplatte
Maßstab: 2:1
Toleranz: fein
50
40
10
A
A
65
50
15
5
10
1X
Schnitt A-A
M5
10
10
10
10
M5
M5


Zeichner: Neubauer
Material: Stahlblech
Bezeichnung: Schaltschrankgrundplatte
Maßstab: 1:2.5
Toleranz:
0
40
110
180
250
320
390
66
282
0
40
110
180
240
378
60
276
128
228


Zeichner:
Neubauer
Material:
Maßstab:
Bezeichnung:
Schaltschrankaufbau
Toleranz:
X1
F1
K1
X2
T1
aisle1,X1
aisle2,X1
aisle3,X1
aisle1, CPX-DI8
aisle1, CPX-CEC-C1
aisle2, CPX-DI8
aisle2, CPX-CEC-C1
aisle3, CPX-DI8
aisle3, CPX-CEC-C1
aisle1, x_axis, CMMS-AS
aisle1, y_axis, CMMS-AS
aisle1, z_axis, CMMS-AS
aisle2, x_axis, CMMS-AS
aisle2, y_axis, CMMS-AS
aisle3, x_axis, CMMS-AS
aisle3, y_axis, CMMS-AS
aisle3, z_axis, CMMS-AS


Zeichner:
Neubauer
Material:
Bezeichnung:
Baugruppe Befestigung
Entnahmeachse
Maßstab:
Toleranz:

Literaturverzeichnis

[1] D. Arnold, *Handbuch Logistik*. Springer Verlag, 2008.

[2] J. H. Christensen, *IEC 61499 Compliance Profile for Feasibility Demonstrations*, [Online 28.04.2012]. Available: http://www.holobloc.com.

[3] *CANopen für Motorcontroller CMMS/CMMD*, Festo AG & Co KG, 2011.

[4] W. Fischer und L. Dittrich, *Materialfluß und Logistik:*, Reihe VDI-Buch. Springer, 2004.

[5] K. Furmans und L. Overmeyer, "Automatisierung in der logistik-dezentral und geregelt", *at-Automatisierungstechnik*, Vol. 59, Nr. 4, S. 207–209, 2011.

[6] M. Groover, *Automation, production systems, and computer-integrated manufacturing*. Prentice Hall Press, 2007.

[7] W. Günthner und M. Ten Hompel, *Internet der Dinge in der Intralogistik*. Springer Verlag, 2010.

[8] *IEC 61499-1: Function blocks-Part 1: Architecture*, International Electrotechnical Commission Std., 2005.

[9] B. Kim, J. Robert, S. Sunderesh und A. Onge, "Intelligent agent modeling of an industrial warehousing problem", *Institute of Industrial Engineers Transactions*, Vol. 34, Nr. 7, S. 601–612, 2002.

[10] W. Kriesel, T. Heimbold und D. Telschow, *Bustechnologien für die Automation: Vernetzung, Auswahl und Anwendung von Kommunikationssystemen*. Hüthig, 2000.

[11] W. Lawrenz, *CAN Controller Area Network: Grundlagen und Praxis*. Hüthig Verlag, 2011.

[12] W. Leonhardsberger und A. Zoitl, "Using ethernet/ip with iec 61499 communication function blocks", in *Proceedings of the 5th international conference on Industrial applications of holonic and multi-agent systems for manufacturing*, Reihe HoloMAS'11. Berlin, Heidelberg: Springer-Verlag, 2011, S. 39–49.

[13] *CANopen guidline*, Leuze electronic Gmbh + Co. KG.[Online 28.04.2012]. Available: http://www.lenze.de/downloads/los/08/tb_canopen_guideline_de.pdf.

[14] B. Oestereich, *Objektorientierte Softwareentwicklung - Analyse und Design min der Unified Modeling Language*. Oldenbourg Verlag, 2001.

[15] B. Reissenweber, *Feldbussysteme zur industriellen Kommunikation.* Oldenbourg Industrieverl., 2002.

[16] K. J. Roodbergen und I. F. Vis, "A survey of literature on automated storage and retrieval systems", *European Journal of Operational Research*, Vol. 194, Nr. 2, S. 343–362, April 2009.

[17] M. Ten Hompel und T. Schmidt, *Warehouse management: Automatisierung und Organisation von Lager-und Kommissioniersystemen.* Springer, 2005.

[18] J. P. Van den Berg, "A literature survey on planning and control of warehousing systems", *Institute of Industrial Engineers Transactions*, Vol. 31, S. 751–762, 1999.