

# Konzeption, Projektierung und Inbetriebnahme eines mehrachsigen Positioniersystems

---

Bachelorarbeit

Name des Studiengangs

Elektrotechnik

Fachbereich 1

vorgelegt von

Aaron Zielstorff

Datum:  
Berlin, 21.09.2021

Erstgutachter\_in: Herr Prof. Dr. Stephan Schäfer  
Zweitgutachter\_in: Herr Dipl.-Ing. Dirk Schöttke

# Inhaltsverzeichnis

<b>Aufgabenstellung</b>	<b>IV</b>
<b>Abbildungsverzeichnis</b>	<b>V</b>
<b>Tabellenverzeichnis</b>	<b>VI</b>
<b>Abkürzungsverzeichnis</b>	<b>VIII</b>
<b>1 Einleitung</b>	<b>1</b>
<b>2 Grundlagen</b>	<b>2</b>
2.1 Requierements Engineering . . . . .	3
2.2 Anlagenprojektierung . . . . .	4
<b>3 Konzeption</b>	<b>5</b>
3.1 Anforderungsanalyse . . . . .	5
3.1.1 Funktionale Anforderungen . . . . .	7
3.1.2 Nicht-funktionale Anforderungen . . . . .	15
3.2 Identifikation der Stakeholder . . . . .	31
3.3 Vorstellung der Laboranlage . . . . .	33
3.3.1 Aufbau des Positioniersystems . . . . .	33
3.3.2 Betriebsumgebung . . . . .	34
3.3.3 Betriebsmodi . . . . .	34
<b>4 Projektierung</b>	<b>39</b>
4.1 Kontextanalyse . . . . .	39
4.2 Anwendungsfallspezifikation . . . . .	44
4.2.1 Entwicklung der Systemprozesse . . . . .	44
4.2.2 Präzisierung der Systemprozesse . . . . .	45
4.3 Verhaltensspezifikation . . . . .	53
4.4 Partitionierung . . . . .	58
4.4.1 Erster Partitionierungsschritt . . . . .	58
4.4.2 Zweiter Partitionierungsschritt . . . . .	60
4.4.3 Dritter Partitionierungsschritt . . . . .	61
4.5 Testspezifikation . . . . .	66
4.6 Stromlaufplan . . . . .	71
4.7 Datenmodell . . . . .	72
4.8 Bedienkonzept . . . . .	77

<b>5 Inbetriebnahme</b>	<b>78</b>
5.1 Implementationsphase . . . . .	78
5.1.1 Hardware-Implementation . . . . .	79
5.1.2 Software-Implementation . . . . .	83
5.2 Verifizierung der Testspezifikation . . . . .	98
5.2.1 Sichtprüfung . . . . .	98
5.2.2 Elektrische Prüfung . . . . .	98
5.2.3 Prüfung der Geräteparametrierung . . . . .	98
5.2.4 Prüfung der Programmfunctionen . . . . .	98
5.3 Korrekturen und Verbesserungen . . . . .	99
5.4 Bedienungsanleitung . . . . .	100
<b>6 Zusammenfassung und Fazit</b>	<b>101</b>
<b>7 Ausblick</b>	<b>102</b>
<b>Literaturverzeichnis</b>	<b>103</b>
Bücher . . . . .	103
Artikel . . . . .	103
<b>Anhang</b>	<b>104</b>
<b>Eidesstattliche Erklärung</b>	<b>105</b>

## Aufgabenstellung

# Abbildungsverzeichnis

1	Logische Kontextabgrenzung . . . . .	41
2	Physikalische Kontextabgrenzung . . . . .	42
3	Anwendungsfalldiagramm . . . . .	45
4	Zustandsdiagramm Automatikmodus . . . . .	54
5	Zustandsdiagramm Handmodus . . . . .	55
6	Zustandsdiagramm Programmänderungen . . . . .	56
7	Zustandsdiagramm Prozessdaten . . . . .	57
8	Erster Partitionierungsschritt . . . . .	59
9	Zweiter Partitionierungsschritt . . . . .	60
10	Dritter Partitionierungsschritt . . . . .	61
11	Anlagengerüst . . . . .	79
12	Steuerungsinstallation . . . . .	80
13	Aktuator- und Sensorinstallation . . . . .	81
14	Schalschrank . . . . .	82
15	Geräteaddressierung . . . . .	84
16	Globale Variablenliste . . . . .	85
17	Zuweisung der digitalen Eingänge des TM5SDI16D Moduls . . . . .	86
18	Zuweisung der digitalen Ausgänge des TM5SDO16T Moduls . . . . .	86
19	Zuweisung der sicheren digitalen Eingänge des TM5SDI4DFS Moduls . . . . .	87
20	Zuweisung der sicheren digitalen Ausgänge des TM5SDO4TFS Moduls . . . . .	87
21	Zuweisung der analogen Eingänge des TM5SAI4L Moduls . . . . .	88
22	Zuweisung der analogen Ausgänge des TM5SAO4L Moduls . . . . .	88
23	Zuweisung der Variablen für den Datentransfer vom SLC zum LMC . . . . .	89
24	Zuweisung der Variablen für den Datentransfer vom LMC zum SLC . . . . .	89
25	Einstellen der Bewegungsparameter für langsame manuelle Testfahrt . . . . .	90
26	Parametrierung des Motors für die x-Achse . . . . .	90
27	Parametrierung der Mechanik für die x-Achse . . . . .	90
28	Einbinden der Endlagesensoren . . . . .	91
29	Konfiguration des Netzteils . . . . .	92
30	Deaktivierung der Netzüberwachung . . . . .	92
31	I/O-Konfiguration des SLC . . . . .	93
32	Hardwareprüfung der sicheren E/A-Module . . . . .	94
33	Sicherer SF_Antivalent Funktionbaustein für die Verarbeitung der Not-Halt-Auslösung . . . . .	95
34	Verarbeitung des Not-Halt-Signals und der Lichtvorhangssignale . . . . .	95
35	Zuweisung der Not-Halt-Variable im LMC . . . . .	97

# Tabellenverzeichnis

1	FA - EIN-Schalter . . . . .	8
2	FA - Wahlschalter Betriebsmodus . . . . .	9
3	FA - Positionieren auf zwei Achsen . . . . .	10
4	FA - Bremsen der Achsbewegungen . . . . .	11
5	FA - Regulierung der Fahrgeschwindigkeit . . . . .	12
6	FA - Greifen von Transportobjekten . . . . .	13
7	FA - Tastersteuerung im Handmodus . . . . .	14
8	FA - Anpassung des Fahrverhaltens . . . . .	15
9	NFA - Sicherheit für Leib und Leben . . . . .	17
10	NFA - Freigabe nach Fehlersituation . . . . .	18
11	NFA - Signalisierung Gefahrensituation . . . . .	18
12	NFA - Indikatorleuchten Betriebsmodi . . . . .	19
13	NFA - Abbremsen der Achsen . . . . .	19
14	NFA - Gehäuse . . . . .	20
15	NFA - Ausfallhäufigkeit . . . . .	21
16	NFA - Verhinderung von Schäden an der Anlage . . . . .	22
17	NFA - Kabelmitführung . . . . .	23
18	NFA - Endlagesensorik . . . . .	24
19	NFA - Geschwindigkeitsreduktion in Endlagennähe . . . . .	25
20	NFA - Programmierschnittstelle . . . . .	26
21	NFA - OPC UA Schnittstelle . . . . .	27
22	NFA - Betriebsstrommessung . . . . .	28
23	NFA - räumliche Vorgabe . . . . .	29
24	NFA - Hardwarevorgaben . . . . .	30
25	Stakeholder . . . . .	32
26	Systemprozess - Objekttransport . . . . .	48
27	Systemprozess - manuelle Funktionsausführung . . . . .	50
28	Systemprozess - Programmänderungen vornehmen . . . . .	51
29	Systemprozess - Prozessdaten bereitstellen . . . . .	52
30	Knotenpunktbeschreibung - Anlagenteil . . . . .	63
31	Knotenpunktbeschreibung - Hardware . . . . .	64
32	Knotenpunktbeschreibung - Software . . . . .	65
33	Testkriterium - Betriebsmodus . . . . .	67
34	Testkriterium - Positionieren . . . . .	68
35	Testkriterium - Programmieren . . . . .	69
36	Testkriterium - Prozessdaten . . . . .	70
37	Datenmodell digitale Eingänge . . . . .	73
38	Datenmodell digitale Ausgänge . . . . .	74
39	Datenmodell Analoge Eingänge . . . . .	74

40	Datenmodell Analoge Ausgänge . . . . .	74
41	Datenmodell sichere digitale Eingänge . . . . .	75
42	Datenmodell sichere digitale Ausgänge . . . . .	75
43	Datenmodell interne Variablen SLC2LMC . . . . .	76

# **Abkürzungsverzeichnis**

**FA** Funktionale Anforderung

**NFA** Nicht-Funktionale Anforderung

**AR** Augmented Reality

**VR** Virtual Reality

**HMI** Human Machine Interface

**MMS** Mensch-Maschine-Schnittstelle

**LMC** Logic Motion Controller

**LXM** Lexium

**OPC** Open Plattform Communications

**UA** Unified Architecture

**ID** Identifikationsnummer

**SPS** speicherprogrammierbare Steuerung

**HTW** Hochschule für Technik und Wirtschaft

**UML** Unified Modeling Language

**RE** Requirements Engineering

**BMK** Betriebsmittelkennung

**E/A** Eingang/Ausgang

**TA** Taster

**Rtg.** Richtungsgeber

**NA** nicht angeschlossen

**Poti** Potentiometer

**SLC** Safety Logic Controller

---

# 1 Einleitung

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

---

## 2 Grundlagen

In diesem ersten Kapitel des Hauptteils der Arbeit sollen zunächst die theoretischen Aspekte als Grundlage für die folgenden Kapitel beleuchtet und erläutert werden. Ohne an dieser Stelle tiefer in Details einzudringen, ist das grundsätzliche Ziel dieser Arbeit, ein mehrachsiges Positioniersystem zu entwickeln und realisieren, so dass es im Laborbetrieb von Studierenden genutzt werden kann. Der Prozess der Umsetzung des Systems wird in dieser Arbeit in drei wesentliche Schritte unterteilt, den **Konzeptionsteil**, den **Projektierungsteil** und die **Inbetriebnahme**. Für die erfolgreiche und normkonforme Dokumentation und Nutzung der drei Kapitel bis hin zur Inbetriebnahme des positionier- systems als Laboranlage an der Hochschule für Technik und Wirtschaft (HTW) Berlin, muss zunächst ein Grundgerüst gesetzt werden. Dieses besteht vor allem aus den beiden essenziellen Themen **Requierements Engineering** (zu deutsch Anforderungsentwick- lung oder auch Anforderungsmanagement) und Anlagenprojektierung mit dem Fokus auf Automatisierungsanlagen.

Die Konzeptionsphase ist besonders geprägt durch das Requierements Engineering. Insbesondere der Abschnitt der Anforderungsanalyse unter Einbeziehung der Stakeholder und der Testspezifikation(en) zu den Anforderungen, begründen sich auf diesem. Die nachfol- gende Systemanalyse im Konzeptionsteil der Arbeit basiert auf den Anforderungen, die mit Hilfe der Dokumentationsempfehlungen des Requierements Engineering entstanden sind.

Vor allem der zweite Teil dieser Arbeit, die Projektierung bedarf der Behandlung von theoretischen Grundlagen zur Projektierung von Automatisierungsanlagen. Bereits die Analysephase in der Systemkonzeption kann der Projektierung zugeordnet werden. In dieser Arbeit liegt der Schwerpunkt im Kapitel der Projektierung jedoch insbesondere auf der Entwicklung der Automatisierungssoftware, bei der die Modelle (vor allem UML Diagramme) aus dem Konzeptionsteil implementiert werden. Für die Anlagenprogrammierung bedarf es der Einhaltung der gültigen Norm DIN/EN 61131, die auf der IEC Norm 61131 basiert und sich mit speicherprogrammierbaren Steuerungen befasst. Insbesondere Teil 3 der Norm beschäftigt sich mit den Programmiersprachen, die auch in der auf Codesys basierenden Entwicklungsumgebung, welche in dieser Arbeit genutzt wird, zur Verwendung kommen.

Der letzte Teil der Arbeit, die Inbetriebnahme, stellt die Verifizierung der Anforderungen aus dem Konzeptionsteil dar. Es handelt sich also um das logische Endstück für das Vorgehen nach den Handlungsempfehlungen des Requierements Engineering. Endstück meint jedoch nicht, dass ein System als fertig umgesetzt gilt, wenn es in der Phase der Inbetriebnahme angelangt ist. Bei der Anforderungsanalyse handelt es sich um ein ite- ratives Verfahren, bei dem aus den Ergebnissen der vorherigen Iteration mögliche neue Anforderungen entstehen könnten.

In dem sich anschließenden Abschnitt wird zunächst das Requierements Engineering als theoretische Grundlage erläutert. Daran schließt sich nachfolgend die Erklärung zur

Vorgehensweise in der Anlagenprojektierung für Automatisierungsanlagen an.

### 2.1 Requierements Engineering

Das Requierements Engineering (RE) ist der Zweig der Ingenieurwissenschaften, der sich mit den realen Zielen, Funktionen und Beschränkungen von Systemen befasst. Weiterhin untersucht es die Beziehung zwischen den genannten Faktoren, um die Spezifikationen des Systemverhaltens zu präzisieren, sowie deren Entwicklung im Laufe der Zeit und über Familien von verwandten Systemen hinweg darzustellen [Lap14, Seiten 2–3]. Ziel der Anwendung des Requierements Engineerings ist zum einen das Verhindern von unnötigen Aufwänden im Verlauf des Lebenszyklus eines Systems. Außerdem kann durch die Einhaltung von Handlungsempfehlungen aus dem RE die Kosteneffektivität optimiert werden.

## 2.2 Anlagenprojektierung

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

---

## 3 Konzeption

Dieses Kapitel unterteilt sich in drei Abschnitte. Die Konzeptionierung der Laboranlage erfolgt nach dem Requirements-Engineering, welches als Grundlage im vorhergegangenen Kapitel bereits behandelt wurde.

Im ersten Unterkapitel wird die Anforderungsanalyse durchgeführt. Schwerpunkt der Arbeit liegt dabei auf dem Software- und Systementwicklungsprozess, somit sind die Anforderungen an die Hardware nur grundlegend aufgelistet.

Der Entwicklungsprozess unter dem Gesichtspunkt der Konzeption des Systems umfasst dabei zwei Kernabschnitte, welche sich in die Anforderungsphase des Entwicklungsprozesses eingliedern. Bei den Kernabschnitten handelt es sich um folgende Analyseschwerpunkte.

- **Anforderungsanalyse:** Es wird unterschieden zwischen funktionalen und nicht-funktionalen Anforderungen an das System.
- **Identifikation der Stakeholder:** Ermittlung aller an der Systementwicklung und Systemnutzung beteiligten Personen zur Feststellung von Randbedingungen an die Anforderungen.

Der dritte Abschnitt dient zum Grobentwurf des Hardwareaufbaus des Positioniersystems. Dazu wird eine Konzeptgrafik entwickelt, die als Konfigurator zu verstehen ist. Weiterhin werden die geplanten Betriebsmodi beschrieben, die in der Projektierung dieser Arbeit zu modellieren sind.

### 3.1 Anforderungsanalyse

In der Analysephase der Systementwicklung werden die Kundenanforderungen zusammengetragen und untersucht. Dabei stellt die Anforderungsanalysephase den ersten Schritt zum Aufstellen der initialen Dokumente für den Prozess dar. In weiteren Iterationen liegen der Anforderungsanalyse zusätzlich zu der ursprünglichen Aufgabenstellung noch die Ergebnisse der Tests und die erkannten Analysefehler ebenfalls als Quelle vor.

Die ermittelten Anforderungen werden untergliedert in funktionale und nicht-funktionale Anforderungen (kurz **FAs** und **NFAs**). Diese Unterteilung findet in der Arbeit in separaten Unterabschnitten statt, die sich nachfolgend anschließen. Die Identifikation der Stakeholder ist grundsätzlich der Anforderungsanalyse zugehörig, wird jedoch in einem gesonderten, sich der Anforderungsanalyse anschließenden, Unterkapitel behandelt, da es sich im Kontext des Konzeptionsteils dieser Arbeit um einen Kernabschnitt handelt.

Zur übersichtlichen Einordnung des jeweiligen Analyseschrittes wird die Grafik Analysephase eingeführt, an der sich die folgenden Kapitel entlangbewegen. Die Anforderungsanalyse kann auf der linken Seite der Grafik identifiziert werden und untergliedert sich in die bereits erwähnten drei Unterpunkte.

Die folgenden Abschnitte betrachten die Erstellung einer konkreten Anforderungsspezifikation, die zum Startbeginn des Entwicklungsprozesses vorliegen muss. In den Unterabschnitten zu den Funktionale Anforderungen (FAs) und Nicht-Funktionale Anforderungen (NFAs) werden die notwendigen Anforderungen für die Entwicklung der Laboranlage vorgestellt. Dabei sind die Hardwareanforderungen nur Beispiellohaft aufgelistet, da die Systemhardware nur eine Untergeordnete Relevanz in dieser Arbeit hat. Alle nicht aufgeföhrten Anforderungen wurden ergänzend im Anhang beigeftigt.

Aus der theoretischen Grundlagen bereits erkenntlich, bestehen Anforderungen aus Zielen, die im Rahmen der Entwicklung erreicht werden sollen. Dabei handelt es sich um einfachen Text, der nach Absprachen mit dem Kunden Dokumentiert wird. Konkret geht es im Fall dieser Arbeit um die definierten Aufgaben und Ziele, welche durch Professoren/innen und Laboringenieure/innen bzw. Mitarbeiter/innen des Fachbereiches augestellt wurden. Auch selbstaufgerlegte Aufgaben (Anforderungen des Systementwicklers) werden mit aufgeführt. Im ersten Schritt ist es notwendig die Menge aller Aufgaben zu konkretisieren, um überflüssige und irrelevante Lösungen diese betreffend zu vermeiden.

Ausgangspunkt für die Entwicklung des mehrachsigen Positioniersystems sind folgende Kernanforderungen bzw. Ziele. Es wird gefordert, eine Laboranlage zu entwickeln, die simple Transportgüter sicher von einer Aufnahmeposition zu einem Ablageort transportieren kann. Dies soll über zunächst zwei Achsen geschehen, die es ermöglichen Bewegungen in horizontale Richtung (X-Achse) und vertikale Richtung (Z-Achse) durchzuführen. Dabei ist es relevant, dass verschiedene Trajektorien von der Anlage gefahren werden können, welche durch den Nutzer programmatisch vorgegeben werden. Die Bewegung der Achsen erfolgt über zwei getrennt ansteuerbare Servomotoren, die über einen Servoregler mit einer Industriesteuerung verbunden sind. Die Steuerungskomponenten sind bereits vorhanden und müssen verwendet werden. Konkret handelt es sich um den LMC101 (Logic Motion Controller) von Schneider Electric, das LXM 62 P Netzgerät (engl. Powersupply, ebenfalls von Schneider Electric) und den LXM 62 D Doppelantrieb (engl. Double Drive). Zusätzlich soll eine PFC200 Steuerung von Wago zum Einsatz kommen, mit der Betriebsströme gemessen und für die Weiterverarbeitung bereit gestellt werden können. Weiterhin sollen auch ausgewählte Prozessdaten aus dem Systemablauf für die externe Verarbeitung zur Verfügung gestellt werden. Es ist vorgegeben, dass diese Daten per Open Plattform Communications (OPC) Unified Architecture (UA) Schnittstelle ausgelesen werden können. Kernziel bei der Entwicklung des Laborsystems ist es die Möglichkeit bereitzustellen, dass die Positioniereinheit von jedem Laborplatz programmiert und als Testsystem für den Lehrzweck eingesetzt werden kann. Für den Betrieb der Anlage sind zwei Betriebsmodi vorgesehen. Ersterer, der Automatikbetrieb soll einen Vollautomatischen Prozessablauf ermöglichen, bei welchem eine konkrete Positionieraufgabe zyklisch durchgeführt wird. Zweiterer, der Handbetrieb, nimmt manuelle Steuerbefehle vom Nutzer entgegen, bei welchen über Tastereingaben an der Laboranlage, Fahrbewegungen entlang der beiden Achsen durchgeführt werden können. Die Auswahl bzw. ein Wechsel zwischen den Betriebsmodi,

ist über einen Wahltafel zu implementieren. Außerdem ist ein Schutz für die Anlage und deren Nutzer, sowie sich um das Positioniersystem befindende Personen zu implementieren. Der Schutz ist manuell auslösbar über Not-Halt Taster an der Laboranlage und durch einen Lichtvorhang vor dem Fahrbereich der beiden Achsen. Abschließend wird gefordert, dass es zu einem späteren Zeitpunkt noch möglich ist, das System um weitere Achsen und Peripheriegeräte wie bspw. Förderbänder zu erweitern.

#### 3.1.1 Funktionale Anforderungen

Der erste Unterabschnitt der Anforderungsanalyse behandelt die Modellierung der funktionalen Anforderungen des Prozesses. Im Requirements Engineering beschreiben funktionale Anforderungen gewünschte Funktionalitäten des Systems. Konkret steht im Mittelpunkt der Analyse, welche Fähigkeiten das System besitzen soll bzw. was es umgangssprachlich formuliert tun kann. Die Auflistung der Anforderungen ist eine Sammlung von systemspezifischen Daten, sowie eine grundlegende Beschreibung des Systemverhaltens.

Die Dokumentation der funktionalen Anforderungen erfolgt typischerweise in Tabellenform. Bereits in den Anforderungen wird ein Abnahmekriterium für diese formuliert, um bei der Inbetriebnahme des Systems die Erfüllung der Anforderung bestätigen oder widerlegen zu können.

Die Nachfolgenden Tabellen zu den funktionalen Anforderungen sind wie folgt strukturiert. Im ersten Eintrag, der **Beschreibung**, wird zunächst in kurzer Textform die Anforderung an das System formuliert. Im nächsten Punkt, dem **Abnahmekriterium** findet eine Erklärung zur Überprüfung der Umsetzung behandelter Anforderung statt. Die Tabellenzeile **Quelle** verweist auf einen oder mehrere Einträge in der Stakeholdertabelle, welche im Unterabschnitt 3.2 vorgestellt wird. Bei Nachfragen zu der jeweils behandelten funktionalen Anforderung ist die Tabelle zur Klärung durch den Prozessentwickler heranzuziehen. Der Eintrag **Begründung** enthält Informationen zur Relevanz der Anforderung, die in der Tabelle beschrieben wird. Dem Punkt **Abhängigkeit** unterliegt eine besondere Wichtigkeit, da hier alle Anforderungen aufgelistet sind, die auf der in der Tabelle beschriebenen Anforderung basieren bzw. in direkter Abhängigkeit zu dieser stehen. Der letzte Eintrag, die **Identifikationsnummer** (kurz ID) dient zur späteren Referenzierung und leichterem Nachschlagen einer Anforderung. Sie ist hilfreich, um Mehrdeutigkeiten zu vermeiden und eine eindeutige Identifizierung sicherzustellen.

Die Nachfolgenden Tabellen folgen dem beschriebenen Muster und beinhalten alle funktionalen Anforderungen des mehrachsigen Positioniersystems. Die Anforderungen an die Hardware des Systems sind nur Beispielhaft am Ende des Unterabschnittes erwähnt. Alle nicht erwähnten Anforderungen können im Anhang nachgeschlagen werden.

<b>Beschreibung</b>	Das Positioniersystem soll über einen dedizierten Einschalter unter Spannung gesetzt werden können.
<b>Abnahmekriterium</b>	Test des gekennzeichneten Einschalters unter Prüfung der Systemspannung nach Betätigung des Schalters.
<b>Quelle</b>	Laborpersonal siehe Stakeholderliste
<b>Begründung</b>	Es wird verlangt, bei Nichtnutzung des Systems dieses zu deaktivieren um das Gefahrenrisiko zu minimieren.
<b>Abhängigkeit</b>	<ul style="list-style-type: none"> <li>• Die Positioniereinheit kann erst nach dem Einschalten genutzt werden (Auswahl des Betriebsmodus).</li> <li>• Es sind Schutzmaßnahmen für Anwender und Anlage umzusetzen, um das Gefahrenrisiko zu minimieren.</li> </ul>
<b>Identifikationsnummer</b>	1.1.1

Tabelle 1: Funktionale Anforderung - Ein-Schalter

<b>Beschreibung</b>	Über einen Wahlschalter soll der Betriebsmodus des mehrachsigen Positioniersystems vorgegeben werden können.
<b>Abnahmekriterium</b>	Auswahl des Betriebsmodus wird über die jeweilige Indikatorenleuchte bestätigt. Der ausgewählte Betriebsmodus kann genutzt werden.
<b>Quelle</b>	Prozessentwickler siehe Stakeholderliste
<b>Begründung</b>	Es ist hilfreich die Auswahl zwischen dem Normalbetrieb (Automatikbetrieb) und dem Handbetrieb zu haben, um das System besser Testen und kalibrieren zu können.
<b>Abhängigkeit</b>	<ul style="list-style-type: none"> <li>• Abarbeitung der Schritte des Automatikbetriebs</li> <li>• Laboranlage befindet sich in Bereitschaft für Eingaben im Handmodus</li> <li>• Indikatorleuchten Betriebsmodi</li> </ul>
<b>Identifikationsnummer</b>	1.1.2

Tabelle 2: Funktionale Anforderung - Wahlschalter Betriebsmodus

<b>Beschreibung</b>	Das Positioniersystem soll zwei bewegbare Achsen besitzen, die sich getrennt steuerbar horizontal und vertikal auf ihrem jeweiligen Profil bewegen können.
<b>Abnahmekriterium</b>	Die beiden Achsen bewegen sich bei Tastereingaben im Handmodus und vollautomatisch im Automatikmodus.
<b>Quelle</b>	Lehrpersonal siehe Stakeholderliste
<b>Begründung</b>	Um Positionieraufgaben durchführen zu können, müssen Achsen zum Einsatz kommen, auf denen bzw. durch welche Bewegungen durchgeführt werden können.
<b>Abhängigkeit</b>	<ul style="list-style-type: none"> <li>• Fahren von Trajektorievorgaben</li> <li>• Joggen der beiden Achsen durch Nutzereingaben</li> </ul>
<b>Identifikationsnummer</b>	1.1.3

Tabelle 3: Funktionale Anforderung - Positionieren auf zwei Achsen

<b>Beschreibung</b>	Bewegungen auf den zwei Achsen sollen gebremst werden können.
<b>Abnahmekriterium</b>	Sowohl ein Erreichen von Endlagepositionen, sowie Start- und Zielpositionen, die Nichtbetätigung von Bewegungstastern im Handmodus und das Auslösen des Not-Halts führen zu einem Bremsen und abschließendem Halten der Achsbewegungen.
<b>Quelle</b>	Lehrpersonal siehe Stakeholderliste
<b>Begründung</b>	Bewegungen entlang der Achsen müssen auch wieder gestoppt werden können, um Beschädigungen der Anlage oder Verletzungen von Menschen zu verhindern.
<b>Abhängigkeit</b>	<ul style="list-style-type: none"> <li>• Verhindern des Runterfallens des beweglichen Schlittens auf der vertikalen Achse (Z-Achse)</li> <li>• Einhalten der Sicherheit für Leib und Leben</li> </ul>
<b>Identifikationsnummer</b>	1.1.4

Tabelle 4: Funktionale Anforderung - Bremsen der Achsbewegungen

<b>Beschreibung</b>	Die Geschwindigkeit, mit der die Positioniereinheit Bewegungen durchführt, soll reguliert werden können.
<b>Abnahmekriterium</b>	Das Einstellen von Geschwindigkeiten über ein Potentiometer an der Schaltschrankfront führt zur Änderung der Fahrgeschwindigkeit der Achsen.
<b>Quelle</b>	Prozessentwickler siehe Stakeholderliste
<b>Begründung</b>	Das Regulieren der Fahrgeschwindigkeit erleichtert auf der einen Seite die Identifikation von Fehlern (langsameres Fahren), auf der anderen Seite kann die Dauer von Positionieraufgaben verringert werden (schnelleres Fahren).
<b>Abhängigkeit</b>	<ul style="list-style-type: none"> <li>• Positionieren auf Zwei Achsen</li> <li>• Verringerung der Beschleunigung und Fahrgeschwindigkeit in Endlagennähe</li> </ul>
<b>Identifikationsnummer</b>	1.1.5

Tabelle 5: Funktionale Anforderung - Regulierung der Fahrgeschwindigkeit

<b>Beschreibung</b>	Durch einen schwenkbaren Greifarm soll es möglich sein zu transportierende Objekte aufzunehmen und wieder abzulegen.
<b>Abnahmekriterium</b>	Transportobjekt befindet sich in Obhut des Systems und kann bewegt werden.
<b>Quelle</b>	Prozessentwickler siehe Stakeholderliste
<b>Begründung</b>	Das Ausführen von Positionieraufgaben wird erst dann ein praxisnahe Beispiel, wenn auch typische Anwendungen aus der Praxis durchgeführt werden (z. B. Transportaufgaben in Hochregallagern).
<b>Abhängigkeit</b>	Bestückung und Abtransport von Auf- und Ablagepositionen mit Transportobjekten (Erweiterung - z. B. Förderbänder)
<b>Identifikationsnummer</b>	1.1.6

Tabelle 6: Funktionale Anforderung - Greifen von Transportobjekten

<b>Beschreibung</b>	Über Tastereingaben soll es möglich sein die beiden Achsen im Handmodus zu bewegen (joggen) und die Greifaktionen manuell auszulösen (triggern).
<b>Abnahmekriterium</b>	Tasteingaben auf dem Vierwegeschalter führen im Handbetrieb zu Achsbewegungen. Durch die Betätigung der vorgesehenen Taster schwenkt der Greifarm um 180° werden und der Greifer wird geöffnet bzw. geschlossen.
<b>Quelle</b>	Lehrpersonal siehe Stakeholderliste
<b>Begründung</b>	Es sind Taster an unter anderem der Schaltschrankfront erforderlich, um das mehrachsige Positioniersystem im Handmodus nutzen zu können.
<b>Abhängigkeit</b>	---
<b>Identifikationsnummer</b>	1.1.7

Tabelle 7: Funktionale Anforderung - Tastersteuerung im Handmodus

<b>Beschreibung</b>	Durch programmatisches Eingreifen soll das Fahrverhalten (z. B. Trajektorien) vorgegeben bzw. angepasst werden können.
<b>Abnahmekriterium</b>	Der Fahrweg zwischen zwei konstanten Punkten wurde erfolgreich angepasst und das System führt abweichende Bewegungsmuster aus.
<b>Quelle</b>	Lehrpersonal siehe Stakeholderliste
<b>Begründung</b>	Durch die Möglichkeit Fahrparameter und somit auch Trajektorien anpassen zu können, wird es möglich verschiedene Parameter zu Testen und diese miteinander zu vergleichen. Dadurch ergibt sich ein Optimierungspotential der Laboranlage.
<b>Abhängigkeit</b>	Programmierschnittstelle zu den Laborcomputern
<b>Identifikationsnummer</b>	1.1.8

Tabelle 8: Funktionale Anforderung - Anpassung des Fahrverhaltens

### 3.1.2 Nicht-funktionale Anforderungen

Dieses Unterkapitel behandelt die Modellierung der nicht-funktionalen Anforderungen in der Anforderungsanalyse. Nicht-funktionale Anforderungen sind Forderungen an die Qualität in welcher Funktionalitäten zu erbringen sind. Auch Randbedingungen für das System bzw. den Prozess werden mit bei den nicht-funktionalen Anforderungen berücksichtigt.

Die **Qualitätsanforderungen** gliedern sich in Zeitanforderungen, Sicherheit für Leib und Leben und Zuverlässigkeit, sowie Verfügbarkeit. Bei *Zeitanforderungen* handelt es sich meist um Reaktionszeiten eines Systems. Dabei wird unterschieden zwischen harten und weichen Zeitanforderungen. Der Verstoß gegen harte Zeitanforderungen kann mitunter sehr gravierend sein, wohingegen das Nichteinhalten von weichen Zeitanforderungen meist nur als Störfaktor gesehen werden kann. Zeitanforderungen finden sich im Entwicklungsprozess überwiegend in der Beschreibung von Systemprozessen oder in Aktivitäten des Zustandsdiagrammes wieder.

Anforderungen bezüglich *Zuverlässigkeit und Verfügbarkeit* treten in der Modellierung in den Knoten des Verteilungsdiagrammes oder fließen in die Systembeschreibung ein.

In die Klasse der Anforderungen bezüglich *Sicherheit für Leib und Leben* fällt die Risiko-vermeidung von Schäden an Menschen, Produkten und die Umwelt.

Abschließend werden die **Randbedingungen** das System betreffend als Sonderklasse der

nicht-funktionalen Anforderungen betrachtet. Man unterteilt diese in zwei Kategorien. Es wird unterschieden zwischen Bedingungen, die sich auf das System und Bedingungen, die sich auf den Entwicklungsprozess auswirken.

Erstere sind Technologievorgaben, physikalische Anforderungen, Umweltanforderungen und Vorgaben für die Einbettung und Verteilung des Systems. Sowohl Technologievorgaben, als auch Vorgaben an die Einbettung und Verteilung fließen direkt in die Modellierung ein. So werden bspw. Nachbarsysteme im Kontextdiagramm und Forderungen nach bestimmter Hardware im Verteilungsdiagramm aufgeführt. Zu den physikalischen Anforderungen zählen z. B. Aussagen über das Gehäuse bzw. die Räumlichkeit, in die das Produkt am Ende der Entwicklung passen muss. Unter Umweltanforderungen versteht man bspw. klimatische Bedingungen, unter denen das System arbeiten muss.

Randbedingungen für den (Entwicklungs-) Prozess basieren auf Vorschriften und Traditionen. Dabei meinen Traditionen Vorschriften, die sich aus bereits früheren Entwicklungen einer Firma ergeben haben.

Zuletzt soll an dieser Stelle noch eine entscheidende Problematik, die durch die Modellierung nicht-funktionaler Anforderungen auftritt, erwähnung finden. Es besteht die Möglichkeit, dass nicht-funktionale Anforderungen entgegensätzliche Dinge verlangen. Um diese Problematik zu beseitigen oder zumindest zu minimieren, hat sich in der Praxis die Vergabe von Prioritäten bewährt. So kann in Tabellenform eine Prioritätsreihenfolge erstellt werden. Diese hilft dem Entwickler zu entscheiden, wie er sich beim Auftreten eines Konfliktes verhält.

Da nun auch die theoretische Grundlage zu den nicht-funktionalen Anforderungen ausreichend beleuchtet ist, folgt die tabellarische Auflistung aller nicht-funktionalen Anforderungen des mehrachsigen Positioniersystems. Dazu wird die selbe Form wie auch schon bei den funktionalen Anforderungen genutzt. Wie auch schon im vorherigen Unterkapitel angewendet, werden die Anforderungen den Hardwareprozess betreffend nur Beispielhaft erwähnt.

<b>Beschreibung</b>	Die Gefahr, dass ein Anwender oder eine sich in Anlagennähe befindende Person durch die Bewegung der Positioniereinheit verletzt wird, soll bestmöglich minimiert werden. Dazu sind Not-Halt Taster vorgesehen, die durch den Anwender betätigt werden können. Zusätzlich ist ein Lichtvorhang verbaut, der die Anlage stoppen soll, falls eine Person durch diesen in den Gefahrenbereich eindringt.
<b>Abnahmekriterium</b>	Durch die Simulation einer Notsituation in Form des Auslösens eines Not-Halt Tasters oder eines Lichtvorhangs muss die Laboranlage unverzüglich Bremsen und in einen haltenden Zustand übergehen, bis die Gefahrensituation behoben ist.
<b>Quelle</b>	Lehrpersonal siehe Stakeholderliste
<b>Begründung</b>	Sicherheit für den Anwender und sich in der Nähe der Anlage befindende Personen.
<b>Abhängigkeit</b>	<ul style="list-style-type: none"> <li>• Freigabe nach Fehlersituation</li> <li>• Signalisierung Gefahrensituation</li> </ul>
<b>Identifikationsnummer</b>	2.1.1

Tabelle 9: Qualitätsanforderung zu Sicherheit für Leib und Leben

<b>Beschreibung</b>	Durch einen Freigabetaster an der Schaltschrankfront kann nach einer Not-Halt-Situation die Anlage wieder in einen betriebsbereiten Zustand zurückgeführt werden.
<b>Abnahmekriterium</b>	Das Positioniersystem führt seinen normalen Prozessablauf nach Freigabebestätigung im Not-Halt-Modus fort.
<b>Quelle</b>	Lehrpersonal siehe Stakeholderliste
<b>Begründung</b>	Das mehrachsige Positioniersystem sollte nach Beendigung der Fehlersituation seinen Betriebsablauf wieder fortsetzen können.
<b>Abhängigkeit</b>	Sicherheit für Leib und Leben
<b>Identifikationsnummer</b>	2.1.2

Tabelle 10: Nicht funktionale Anforderung - Freigabe nach Fehlersituation

<b>Beschreibung</b>	Es soll eine Signalampel verbaut werden, die eine erhöhte Risikosituation signalisiert, wenn das Positioniersystem in Bewegung ist.
<b>Abnahmekriterium</b>	Die Signalampel leuchtet entsprechend des aktuellen Systemzustandes.
<b>Quelle</b>	Prozessentwickler siehe Stakeholderliste
<b>Begründung</b>	Eine den sich um die Laboranlage befindenden Personen ersichtliche Signalisierung einer Gefahrensituation minimiert das Risiko der Verletzung durch die Anlage.
<b>Abhängigkeit</b>	Sicherheit für Leib und Leben
<b>Identifikationsnummer</b>	2.1.3

Tabelle 11: Nicht funktionale Anforderung - Signalisierung Gefahrensituation

<b>Beschreibung</b>	Der aktive Betriebsmodus soll per Indikatorenleuchte signalisiert werden.
<b>Abnahmekriterium</b>	Die Indikator-LED leuchtet entsprechend des aktuellen Betriebsmodus.
<b>Quelle</b>	Prozessentwickler siehe Stakeholderliste
<b>Begründung</b>	Der Aanlgennutzer benötigt eine visuelle Rückmeldung zum ausgewählten Betriebsmodus, um die Laboranlage entsprechend bedienen zu können.
<b>Abhängigkeit</b>	Wahl des Betriebsmodus
<b>Identifikationsnummer</b>	2.1.4

Tabelle 12: Nicht funktionale Anforderung - Indikatorleuchten Betriebsmodi

<b>Beschreibung</b>	Der Abbremsvorgang nach der Auslösung des Not-Halts soll so schnell wie möglich sein.
<b>Abnahmekriterium</b>	Nach Auslösen des Not-Halts bremsen beide Achsen direkt und das Positioniersystem befindet sich nach spätestens 0,5s im Stillstand.
<b>Quelle</b>	Lehrpersonal siehe Stakeholderliste
<b>Begründung</b>	Um Verletzungen zu verhindern muss die Laboranlage so schnell wie möglich abbremsen in einer Not-Halt-Situation.
<b>Abhängigkeit</b>	<ul style="list-style-type: none"> <li>• Sicherheit für Leib und Leben</li> <li>• Bremsen der Achsbewegungen</li> </ul>
<b>Identifikationsnummer</b>	2.1.5

Tabelle 13: Zeitanforderung - Abbremsen der Achsen

<b>Beschreibung</b>	Die rechte und linke Seite der Laboranlage sollen mit Plexiglasscheiben bestückt werden. Die elektrische Verteilung ist in einem Schaltschrank unterzubringen.
<b>Abnahmekriterium</b>	Das Positioniersystem ist von den Seiten mit Plexiglas abgeschirmt und die elektrischen Betriebsmittel bzw. deren Anschlüsse sind in einem Schaltschrank untergebracht.
<b>Quelle</b>	Mitarbeiter der Labore siehe Stakeholderliste
<b>Begründung</b>	Weder der Anlagenbereich, in dem Achsbewegungen stattfinden, noch der Anlagenbereich, in dem das Risiko zur Verletzung durch einen elektrischen Schock bestehen dürfen frei zugänglich sein, um Verletzungen zu vermeiden.
<b>Abhängigkeit</b>	Sicherheit für Leib und Leben
<b>Identifikationsnummer</b>	2.1.6

Tabelle 14: Physikalische Anforderung - Gehäuse

<b>Beschreibung</b>	Das System sollte möglichst nicht während des Betriebs ausfallen.
<b>Abnahmekriterium</b>	Es findet maximal ein Systemausfall alle 10.000h Betriebsstunden statt.
<b>Quelle</b>	Mitarbeiter der Labore siehe Stakeholderliste
<b>Begründung</b>	Das mehrachsige Positioniersystem muss zuverlässig laufen, um sowohl für den Lehrzweck als auch für die Gewinnung von Systemdaten zur Weiterverarbeitung eingesetzt werden zu können.
<b>Abhängigkeit</b>	---
<b>Identifikationsnummer</b>	2.1.7

Tabelle 15: Qualitätsanforderung - Ausfallhäufigkeit

<b>Beschreibung</b>	Die Beschädigung der Laboranlage durch die Bewegung der Achsen soll bestmöglich verhindert werden.
<b>Abnahmekriterium</b>	Es wurden alle üblichen Betriebmittel zu Verhinderung von Schäden durch bewegliche Systemkomponenten verbaut. Weiterhin existiert eine Not-Halt Funktionalität.
<b>Quelle</b>	Mitarbeiter der Labore siehe Stakeholderliste
<b>Begründung</b>	Die Beschädigung der Positioniereinheit würde sowohl zu Kosten als auch potentiellen Personenschäden führen, was dringlichst vermieden werden muss.
<b>Abhängigkeit</b>	<ul style="list-style-type: none"> <li>• Sicherheit für Leib und Leben</li> <li>• Kabelmitführung</li> </ul>
<b>Identifikationsnummer</b>	2.1.8

Tabelle 16: Qualitätsanforderung - Verhinderung von Schäden an der Anlage

---

<b>Beschreibung</b>	Kabel an den Beweglichen Bauteilen der Positionier-einheit müssen sicher mitgeführt werden.
<b>Abnahmekriterium</b>	Die Laboranlage besitzt für jede bewegliche Achse eine E-Kette in welchen sich die Kabel der Betriebsmittel der jeweiligen Achse befinden.
<b>Quelle</b>	Mitarbeiter der Labore siehe Stakeholderliste
<b>Begründung</b>	Sämtliche Kabel zu beweglichen Systemkomponen-ten müssen beweglich gelagert werden, da diese sonst mit umliegenden Betriebsmitteln interferieren könnten. Weiterhin wird dadurch verhindert, dass Kabel reißen.
<b>Abhängigkeit</b>	Verhindern von Schäden an der Anlage
<b>Identifikationsnummer</b>	2.1.9

Tabelle 17: Qualitätsanforderung - Kabelmitführung

<b>Beschreibung</b>	Fahrbewegungen über den Fahrbereich hinaus müssen verhindert werden.
<b>Abnahmekriterium</b>	Es wurden sowohl Induktive Näherungsschalter als auch Bremspuffer an allen Enden der Achsen verbaut. Sämtliche Bewegungen stoppen schon an den induktiven Enlagesensoren. Tritt ein Fehler auf, fährt die jeweilige Achse auf den Bremspuffer auf und stoppt dort.
<b>Quelle</b>	Mitarbeiter der Labore siehe Stakeholderliste
<b>Begründung</b>	Die beweglichen Schlitten auf den Achsen dürfen nicht über die Achsenden hinausfahren, da diese sonst runterfallen würden, was unhinderlich zu Schäden an der Anlage führt.
<b>Abhängigkeit</b>	Verhindern von Schäden an der Anlage
<b>Identifikationsnummer</b>	2.2.0

Tabelle 18: Qualitätsanforderung - Endlagesensorik

---

<b>Beschreibung</b>	Sowohl Beschleunigung als auch Fahrgeschwindigkeit soll in Endlagennähe verringert sein.
<b>Abnahmekriterium</b>	Bewegungen in den Endbereichen der beiden Achsen finden merklich langsamer statt.
<b>Quelle</b>	Mitarbeiter der Labore siehe Stakeholderliste
<b>Begründung</b>	Zu hohe Beschleunigungen und Geschwindigkeiten nah der Enden der Achsen führen zu möglichen längeren Bremswegen. Dies bedeutet, dass beim Auslösen der Endlagesensoren nicht mehr rechtzeitig angehalten werden kann und die beweglichen Schlitten auf den Achsen mit den Bremspuffern kollidieren könnten.
<b>Abhängigkeit</b>	Verhindern von Schäden an der Anlage
<b>Identifikationsnummer</b>	2.2.1

Tabelle 19: Qualitätsanforderung - Geschwindigkeitsreduktion in Endlagennähe

<b>Beschreibung</b>	Es soll die Möglichkeit bereitgestellt werden von jedem Laborcomputer die Anlage mit Programmcode zu bespielen.
<b>Abnahmekriterium</b>	Die SPS kann in der MachineExpert Programmierumgebung gefunden und mit Programmen bespielt werden.
<b>Quelle</b>	Lehrpersonal siehe Stakeholderliste
<b>Begründung</b>	Für den Einsatz der Laboranlage im Lehrbetrieb muss die Anlage im Labornetzwerk gefunden werden können, um von den Studierenden als Anwendungsbeispiel genutzt werden zu können.
<b>Abhängigkeit</b>	---
<b>Identifikationsnummer</b>	2.2.2

Tabelle 20: Randbedingung - Programmierschnittstelle

<b>Beschreibung</b>	Die SPS soll zusätzlich als OPC UA Server fungieren, über welchen Prozessdaten aus einer globalen Variablenliste ausgegeben werden. Dazu gehören Positionsdaten der beiden Achsen, sowie deren aktuelle Geschwindigkeit und Beschleunigung. Es kann der Status der Bremsen ausgelesen werden und der Betriebsstrom der gesamten Anlage.
<b>Abnahmekriterium</b>	Über einen OPC UA Client können alle Daten ausgelesen werden.
<b>Quelle</b>	Mitarbeiter der Labore siehe Stakeholderliste
<b>Begründung</b>	Die Laboranlage soll über einen AR Server betrachtet werden können, um schneller und anschaulicher Informationen über die Anlage zu erhalten. Außerdem wird das Positioniersystem als Anwendungsbeispiel im OpenBASYS Projekt der Hochschule eingesetzt, bei dem die Anlage über Verwaltungsschalen aufgesetzt wird. Dazu ist es nötig interne Variablen nach außen über OPC UA Schnittstelle zugänglich zu machen.
<b>Abhängigkeit</b>	---
<b>Identifikationsnummer</b>	2.2.3

Tabelle 21: Randbedingung - OPC UA Schnittstelle

<b>Beschreibung</b>	Es soll eine zweite SPS verbaut werden, die über eine Energieklemme verfügt, über welche der Betriebsstrom des Systems gemessen werden kann.
<b>Abnahmekriterium</b>	Über einen OPC UA Client kann auch der Betriebssstrom ausgelesen werden.
<b>Quelle</b>	Mitarbeiter der Labore siehe Stakeholderliste
<b>Begründung</b>	Über die Bereitstellung des aktuellen Stromverbrauches der Anlage können Rückschlüsse über die Effektivität der Anlage geschlossen werden. Da der Stromverbrauch des Positioniersystems somit bekannt ist, besteht die Möglichkeit Optimierungen an der Laboranlage durchzuführen. Es könnten Beispielsweise Trajektorien verbessert werden, so dass weniger Strom benötigt wird für die Bewegungen.
<b>Abhängigkeit</b>	OPC UA Schnittstelle
<b>Identifikationsnummer</b>	2.2.4

Tabelle 22: Randbedingung - Betriebsstrommessung

<b>Beschreibung</b>	Das mehrachsige Positioniersystem soll im Laborraum G 422 der HTW Berlin an der Rückwand des Raumes aufgebaut und in Betrieb genommen werden.
<b>Abnahmekriterium</b>	System ist sichtbar an der Rückwand des Raumes eingebaut.
<b>Quelle</b>	Lehrpersonal siehe Stakeholderliste
<b>Begründung</b>	Da das System den Fachbereich 1 und dem Themengebiet Automatisierungstechnik sowie Industrielle Kommunikation zugeordnet werden kann, bietet es sich an diesesm in einem der entsprechenden Laborräume aufzubauen. Aus platztechnischen Gründen und dem Vorhandensein von kompatibler Hardware (zur Simulation des Systems) ist es erforderlich dieses im Raum G 422 zu installieren.
<b>Abhängigkeit</b>	---
<b>Identifikationsnummer</b>	2.2.5

Tabelle 23: Randbedingung - räumliche Vorgabe

<b>Beschreibung</b>	Die Steuerungskomponenten der Laboranlage sind bereits vorhanden und müssen verbaut werden. Die Hauptsteuerung (SPS des Positioniersystems) ist der LMC101 von Schneider Electric. Zusammen mit dem LXM62 P und LXM62 D Modul gehört dieser zum PacDrive System und ist für Motion Systeme (Bewegte Systeme) entwickelt worden. Sowohl das erwähnte Netzteil (LXM62 P) als auch der Servoregler (LXM62 D) sind bereits vorhanden und sollen mit verbaut werden. Aus diesen Technologievorgaben ergibt sich auch die Wahl der Entwicklungsumgebung zur Programmierung des Systems. Um die Laboranlage in Betrieb zu nehmen ist es nötig diese mit einem Automatisierungsprogramm zu bespielen, welches im MachineExpert Logic Builder entwickelt wurde. Die beiden Motoren des mehrachsigen Positioniersystems stammen auch von Schneider Electric. Es handelt sich um die SH3 multturn Servos mit Bremse. Auch die SPS für die Strommessung per Energieklemme ist vorgegeben. Es soll eine Wago SPS der PFC200 Serie verbaut werden, welche sich bereits im Inventar des Fachbereiches befindet.
<b>Abnahmekriterium</b>	Vorgegebene Hardware wurde verbaut.
<b>Quelle</b>	Lehrpersonal siehe Stakeholderliste
<b>Begründung</b>	Die Hardware wurde von besagten Industrieunternehmen gestellt und musste nicht käuflich erworben werden. Da diese somit vorhanden ist und für vergleichbare Systemaufbauten entwickelt wurde, wird diese verwendet.
<b>Abhängigkeit</b>	---
<b>Identifikationsnummer</b>	2.2.6

Tabelle 24: Randbedingung - Hardwarevorgaben

## 3.2 Identifikation der Stakeholder

Wie in Grafik 1 zu erkennen ist, gehört die Ermittlung der für das Projekt wichtigen Personen (nachfolgen als Stakeholder bezeichnet) zur Anforderungsanalyse. Als Stakeholder gelten Personen, die an der Systementwicklung beteiligt sind aber auch zukünftige Anwender bzw. Personen, die vom Einsatz des Systems betroffen sind. Die in den vorangegangenen Unterkapiteln aufgenommenen Anforderungen und Randbedingungen haben ihre Grundlage auf den von den Stakeholdern bereitgestellten Informationen. Dabei vertreten die Stakeholder verschiedene Interessen das zu entwickelnde System betreffend.

Wie auch schon bei den nicht-funktionalen Anforderungen festgehalten wurde, können unterschiedliche und sogar konträre Bedürfnisse und Ansprüche von den Stakeholdern aufgestellt werden. Um den Entwicklungsprozess durch widersprüchliche Anforderungen von Stakeholdern nicht behindern zu lassen, wird eine tabellarische Auflistung aller für das System relevanten Personen erstellt. Es findet eine Klassifizierung der Stakeholder statt, aus der ersichtlich wird, welche Person bzw. welcher Personenkreis für eine bestimmte Thematik als Ansprechpartner gilt. Anforderungen aus einem bestimmten Themenfeld werden priorisiert, wenn diese von Personen des selben Themenfeldes gestellt wurden. Tabelle 3 zeigt die Auflistung aller Stakeholder des mehrachsigen Positioniersystems. Besonders wichtige Spalten der Tabelle sind zum einen die Rolle des oder der Stakeholder(S), der bzw. die Vertreter und deren Wissengebiet.

Rolle der Stakeholder	Beschreibung	Konkrete Vertreter	Wissengebiet	Begründung
Lehrpersonal der Hochschule	Auftraggeber für den eigentlichen Einsatz des Systems	Herr Prof. Dr. Schäfer  Tel.: 5019-3466 E-Mail: Stephan.Schaefer@HTW-Berlin.de	Lehre und Forschung im Gebiet der Automatisierungs-technik	Auftraggeber und Verantwortlicher
Mitarbeiter der Labore für Automatisierung	Geben zusätzliche Anforderungen für die Verwendung vor	Herr Dipl. Ing. Dirk Schöttke  Tel.: 5019-3564 E-Mail: Dirk.Schoettke@HTW-Berlin.de	Ingenieur mit Fachkenntnissen in der Automatisierungstechnik sowie Anlagenprojektierung	Sorgt für die Eingliederung des Systems in Übergeordnete Projekte

Studenten der Hochschule	Sind die eigentlichen benutzer des Systems	keine Vertreter	Arbeiten mit den Laboranlagen des Fachbereiches	Müssen das System im Lehrbetrieb der Hochschule benutzen
Prozessentwickler	Person(en) die für die Entwicklung des Systems verantwortlich ist/sind	Herr Aaron Zielstorff  tel.: +49177/2847470 E-Mail: Aaron.Zielstorff@HTW-Berlin.de	Entwickler des Positioniersystems	Ist verantwortlich für die Realisierung des Systems nach gegebenen Anforderungen

Tabelle 25: Stakeholder des mehrachsigen Positioniersystems

Die Auflistung der Stakeholder ergibt, dass grundsätzlich zwei Interessengebiete und somit auch zwei Interessengemeinschaften existieren, was die Anforderungen und Interessen an das Positioniersystem betrifft. Auf der einen Seite soll die Positioniereinheit im Lehrbetrieb im Labor eingesetzt werden, um vorlesungsbegleitend Studierenden die Möglichkeit zu bieten, praxisnah Industriesteuerungen für Bewegungsaufgaben (Motion Controlling) zu programmieren. Dazu zählt auch die Anforderung von jedem Laborplatz aus Automatisierungssoftware zu entwickeln, die nach Fertiggstellung auf die Steuerung des mehrachsigen Positioniersystems übertragen werden kann, um Positionieraufgaben an einem realen System durchzuführen. Ziel soll es sein dadurch nicht nur rein simulativ die Abläufe bei der Inbetriebnahme eines solchen Systems zu erproben, sondern zusätzlich auch reale physikalische Einflüsse mit zu berücksichtigen, die eventuelle Abweichungen zu simulierten Automatisierungsprogrammen aufweisen. Außerdem müssen bei der Nutzung realer Hardware auch die Sicherheit von Mensch und Maschine mit berücksichtigt werden, da Gefahren durch den Betrieb des Systems auftreten können.

Auf der anderen Seite soll das Positioniersystem in Drittprojekten mit eingebettet werden. Dazu ist es von Relevanz, dass wichtige Systemdaten über einen OPC UA Server (diese Rolle wird von der Steuerung übernommen) bereitgestellt werden. Über diese Schnittstelle können Daten aus dem Prozess abgegriffen werden, die in einem externen System bzw. in einem Peripheriesystem weiterverarbeitet oder genutzt werden können.

Zusammenfassend kann festgehalten werden, dass die Positioniereinheit Wertschöpfung als Lehrmittel für Studierende und als Quelle für relevante Systemdaten zu diversen Zwecken generieren soll. Dazu ist es unabdinglich, dass sie über ethernetbasierte Schnittstellen (Netzwerkschnittstellen) mit externen Geräten kommunizieren kann.

### 3.3 Vorstellung der Laboranlage

In diesem Unterkapitel wird zunächst die Laboranlage vorgestellt, die im Verlauf der Arbeit unter den Gesichtspunkten des Requierements-Engineerings und der Anlagenprojektierung konzipiert, projektiert und in Betrieb genommen werden soll. Im ersten Abschnitt wird das bereits elektrisch fertiggestellte Positioniersystem dargestellt. Im Mittelpunkt steht hierbei die Erläuterung des Aufbaus und die Beschreibung der Funktionalität der Anlage. Der zweite Abschnitt behandelt die Eingliederung des Systems in seine Arbeitsumgebung. Dabei soll ein erster Überblick zum Einsatz der Positioniereinheit gegeben werden. Zuletzt werden die Betriebsmodi der Laboranlage vorgestellt, wobei genauer auf den Workflow im jeweiligen Modus eingegangen werden soll.

#### 3.3.1 Aufbau des Positioniersystems

Wie bereits aus dem Thema der Bachelorthesis erkenntlich ist, handelt es sich bei der behandelten Laboranlage um ein mehrachsiges Positioniersystem. Dieses besitzt zum Zeitpunkt der ersten Inbetriebnahme zwei Achsen (siehe Bild).

Die horizontale Achse des Systems ist fest an der Wand montiert und hat eine Länge von rund 1600 mm (effektiver Fahrtweg). Vertikal montiert auf dieser befindet sich die beweglich gelagerte zweite Achse der Positioniereinheit. Diese besitzt die Möglichkeit, lineare Bewegungen zwischen den Endlagesensoren der Horizontalachse durchzuführen. Bei der Befestigung an der waagerechten Achse handelt es sich um ein doppeltes Schlitzenstsystem auf Rollen. Die Bewegung der Achse erfolgt über ein Gummiringen, der fest an der Vertikalachse befestigt ist, und über Umlenkrollen und einen Servomotor an der Horizontalachse bewegt werden kann. Auf der senkrechten Achse befindet sich ein weiterer Schlitten, der ebenso beweglich gelagert ist und sich auf einem Fahrtweg von rund 2000mm zwischen zwei Endlagen bewegen kann. Auf diesem ist ein simples Greifsystem angebracht, welches horizontale 180 Grad Schwenkbewegungen durchführen kann und in der Lage ist, grundlegende Greifoperationen durchzuführen.

Für die Zuleitungen zu den auf den bewegten Anlagenteilen montierten Aktoren und Sensoren wurden Energieketten verbaut, sodass Kabel prozesssicher mitgeführt und eine dauerhafte Strom- sowie Datenversorgung aller Systemkomponenten gewährleistet werden kann. An den beiden äußersten Profilen (sowohl auf der linken als auch auf der rechten Seite der Anlage) sind Ablagepositionen vorgesehen, von bzw. auf welche simple Transportgüter aufgenommen und abgelegt werden können.

Auf der rechten Seite direkt neben der Positioniereinheit sind der Schaltschrank sowie die speicherprogrammierbare Steuerung (im Folgenden als SPS bezeichnet) an der Wand montiert. Die Kabel der Aktoren und Sensoren des Systems münden an der Unterseite des Schrankes sowie die Stromzuleitung und sämtliche Aus- und Eingangsverbindungen zu bzw. von der SPS und dem sich neben dieser befindenden Servoantrieb. Auf der Vorderseite an der Tür des Schaltschrankes sind Bedienelemente aufgeschraubt, die für die grundlegende

Steuerung der Anlage benutzt werden können.

Zur Gewährleistung der Sicherheit von Mensch und Anlage sind am Eingang des Positioniersystems sowohl ein Lichtvorhang als auch Not-Halt Bedienelemente montiert. Stromfrei kann die Anlage über den Hauptschalter an der rechten Seite des Schaltschrankes geschaltet werden.

#### 3.3.2 Betriebsumgebung

Nachdem im vorhergehenden Abschnitt bereits die grundlegenden Funktionen und der Aufbau der Positioniereinheit dargestellt wurden, beschäftigt sich dieses Unterkapitel mit der Darstellung der Eingliederung des Systems in dessen Arbeitsumgebung.

Aufgebaut befindet sich das mehrachsige Positioniersystem im Laborraum G422 der HTW Berlin am Campus Wilhelminenhofstraße. Dort wurde die Laboranlage im Rahmen meines Praktikums errichtet. Nachfolgen ist es Ziel der Bachelorthesis, diese Anlage für den Lehrzweck in Betrieb zu nehmen. Konkret soll die Positioniereinheit für zwei Anwendungen eingesetzt werden.

Erstere gliedert sich direkt in die Unterrichtseinheiten des Laborbetriebs im späten Bachelor- und das gesamte Masterstudium im Themenfeld Automatisierungstechnik ein. Jeder studentische Laborplatz besitzt die Möglichkeit, sich mit dem System zu verbinden, um es mit Automatisierungssoftware, die in den Lehreinheiten entwickelt wird, zu bespielen und diese an der Anlage zu testen. Es soll die Möglichkeit bestehen, Trajektorien zu fahren, bei denen virtuelle Hindernisse umgangen werden und Objekte von einem Ausgangspunkt zu einem Zielpunkt transportiert werden können. Die Nutzung der realen Anlage dient dabei als Prüfmöglichkeit der vorher von den Studierenden nur simulativ getesteten Automatisierungssoftware. Ziel ist es, den Laboranten eine Laboranlage zur Verfügung zu stellen, die in der Industrie in ähnlicher Weise aufzufinden ist, um bereits im Studium spätere Arbeitsabläufe aufzuzeigen.

Die zweite Anwendung des mehrachsigen Positioniersystems ist Teil eines laborübergreifenden Projektes, welches nicht in dieser Arbeit behandelt wird. Aus dessen Zielen ergeben sich weitere Anforderungen an die Laboranlage. Es sollen Daten aus dem Prozessablauf bereitgestellt werden, aus denen eine Wertschöpfung für das Projekt generiert werden kann. Die gewonnenen Daten sollen extern weiterverarbeitet werden. Dazu müssen weitere Schnittstellen im System bereitgestellt werden, um generierte Daten mit Peripheriegeräten austauschen zu können.

#### 3.3.3 Betriebsmodi

Die Nutzung der Laboranlage erfolgt in zwei verschiedenen Betriebsmodi. Um den Produktivbetrieb des Positioniersystems einzuleiten, muss der Nutzer zwischen dem Automatikmodus und dem Handmodus auswählen, die im Folgenden detailliert beschrieben werden.

**Automatikbetrieb:** Bei dem Automatikmodus handelt es sich um den üblicherweise

genutzten Betriebsmodus der Laboranlage. Dieser kann vollautomatisch im Dauerbetrieb eingesetzt werden und erfordert nicht die Anwesenheit vom Nutzer. Der Prozessablauf ist programmatisch vorgeschrieben und wird zyklisch durchgeführt. Zur erstmaligen Inbetriebnahme sollen einfache Transportaufgaben durchgeführt werden. So könnte beispielsweise von einer Ablageposition A ein Transportobjekt gegriffen und um Hindernisse herum transportiert werden, sodass besagtes Objekt an einer Zielposition B wieder abgesetzt wird. Danach fährt die Anlage wieder zu Position A, um erneut ein Objekt für den Transport aufzunehmen.

Konkret muss das Positioniersystem im ersten Schritt unter Spannung gesetzt, in dem der Hauptschalter (400 V Ebene) betätigt wird. Dieser befindet sich, wie bereits im vorherigen Unterkapitel erwähnt, auf der rechten Seite des Schaltschranks. Darauffolgend muss im zweiten Schritt die Steuerung (LMC Pro von Schneider Electric) eingeschalten sowie alle Betriebsmittel auf der 24 V Ebene mit Strom versorgt werden. Dies geschieht über den Ein-Taster, welcher sich auf der Front des Schaltschranks befindet. Der eingeschaltete Zustand wird über eine Lampe auf der Schaltschränkfront signalisiert. Als Netzteil dient das LXM62 P Powersupply von Schneider Electric, welches 3-phäsig an der Drehstromsteckdose des Laborraumes angeschlossen ist. Dieses versorgt den LXM62 D double Drive aus der gleichen Produktreihe wie das Netzteil. Die 24 V Steuerungsebene wird von einem separatem Netzteil im Schaltschrank versorgt. Mit einem Wahlschalter kann nun der Automatikmodus des Systems angewählt werden. Bestätigt wird dieser über einen weiteren Taster an der Schaltschränktür. Die erfolgreiche Auswahl des Automatikmodus wird über eine Signalleuchte, welche mit „Auto“ betitelt ist, indiziert. Die Anlage wechselt aus dem Leerlauf in den vollautomatischen Betrieb.

Nach der Wahl des Automatikmodus bewegt das Positioniersystem die auf den beiden Achsen montierte Greifeinrichtung aus der Ausgangsposition des Leerlaufes (auch als Home bezeichnet) zur Ablageposition A. Dazu werden zunächst die Bremsen der beiden Motoren gelöst, welche für die Bewegung der jeweiligen Achse verantwortlich sind. Ist die Position vor der Ablageposition A erreicht, wird im nächsten Schritt ein Schwenkarm mit Greifer so zur Ablageposition A rotiert, dass ein sich darauf befindliches Objekt gegriffen werden kann. Es folgt besagter Greifprozess, um das auf Ablageposition A befindliche Objekt aufzunehmen.

Das Positioniersystem muss nun einen Fahrtweg bewältigen, der mit virtuellen Hindernissen bestückt ist, um Trajektorien zum Transport von Gütern in mit Objekten blockierten Umgebungen zu erproben. Es ist nicht möglich, eine geradlinige Bewegung von Startposition A zur Zielposition, dem Ablageort B, zu fahren. Weiterhin kann auch nicht erst der komplette Fahrtweg in vertikaler Richtung (Z-Richtung) bewältigt werden und dann die Bewegung in horizontaler Richtung (X-Richtung), noch eine geradlinige Bewegung, sodass die Z- und X-Koordinate des Ziels gleichzeitig erreicht werden. Die Hindernisse werden programmatisch vorgegeben und sind somit der Laboranlage bzw. der Automatisierungssoftware bekannt.

im nächsten Schritt werden dem Positioniersystem Koordinaten übergeben, die, wenn diese durchfahren werden, den Weg von Startposition A zu Zielposition B ergeben. Dabei soll berücksichtigt werden, dass nur an der Start- und Zielposition umfangreichere Beschleunigungen stattfinden sollen, welche die Achsen aus der Ruhe beschleunigen bzw. diese wieder abbremsen, die einzelnen Punkte auf dem Weg werden nur durchfahren. Zur Minimierung von starken Trägheitsmomenten ist es weiterhin notwendig, dass die beiden Achsen zusammen keine gradlinigen Fahrtwege zwischen den Wegpunkten nutzen, sondern in abgerundeten (gedämpften) Trajektorien die einzelnen Koordinatenpunkte abfahren. Die konkrete Parametrierung der Fahrwegabschnitte und der sich daraus ergebenden Trajektoriemuster soll Teil der Testszenarien des mehrachsigen Positioniersystems sein. An der Zielposition angekommen, schwenkt der sich auf der Z-Achse befindende Arm um, und das Objekt wird über der Ablageposition vom Greifer losgelassen, sodass es auf der Zielposition verweilt. Die Anlage fährt nun den Weg zur Startposition zurück, um ein weiteres Objekt aufzunehmen und dieses wie bereits beschrieben zu transportieren.

Mögliche spätere Erweiterungen könnten sein, dass der Rückweg anders gewählt wird, da kein Objekt transportiert wird und somit auftretende Trägheitsmomente und Schwingungen keine wichtige Rolle spielen. Alternativ könnte auch auf dem Rückweg ein anderes Objekt von Ablageposition B zu Ablageposition A transportiert werden, welches andere Eigenschaften aufweist, was den Fahrtweg beeinflussen könnte.

Für die vollständige Automatisierung des Prozesses ist eine spätere Erweiterung nötig, bei der auch die Ablageposition(en) automatisch mit neuen Transportobjekten bestückt werden. Es würde sich eine Aufrüstung mit Förderbändern von und zu den Ablagepositionen der Anlage lohnen, sodass steig neue Objekte dem Positioniersystem bereitgestellt und von diesem auch wieder entnommen werden können.

Im letzten Schritt kann die Anlage wieder deaktiviert werden, was über die Abwahl des aktuellen Betriebsmodus geschieht. Es muss der gleiche Taster wie bei der Auswahl des Modus betätigt werden. Dies ist in jedem Moment während der Laufzeit des Automatikmodus möglich. Die letzte Transportaufgabe wird noch vollständig zu Ende durchgeführt. Danach findet das Homing statt, bei dem der sichere Ausgangszustand der Anlage wieder angefahren und die Bremsen der Motoren wieder aktiviert werden. Die Bremsen dienen beim Erreichen des Leerlaufes nicht nur zum Abbremsen der Achsen, sondern sind nötig, damit der Schlitten auf der Vertikalachse nicht bis nach unten fällt. Nach erfolgreicher Abwahl des Betriebsmodus erlischt die Indikatorlampe für den Automatikbetrieb wieder. Nur wenn kein Modus ausgewählt ist, kann die 24 V Ebene wieder spannungsfrei geschalten und die Laboranlage wieder deaktiviert werden. Dies geschieht über den Aus-Taster auf der Front des Schaltschrances. Nach Betätigung des Tasters erlischt die Lampe, welche die Betriebsbereitschaft des Positioniersystems signalisiert.

**Handbetrieb:** Bei dem Handmodus handelt es sich um die zweite Betriebsart der Positioniereinheit. Anders als im Automatikbetrieb dient der Handmodus nicht als Abarbei-

tungsmodus für Positionieraufgaben, sondern soll als manuelle Bedienmöglichkeit genutzt werden können. Das heißt konkret, dass erst durch das Betätigen von Tastern Bewegungen und Aktionen durchgeführt werden.

Wie auch schon im Automatikmodus wird die Anlage zunächst unter Spannung gesetzt durch Betätigung des Hauptschalters. Anschließend wird über den Ein-Tasters die 24V Ebene aktiviert, wodurch auch alle verbundenen Anlagenkomponenten (SPS, Sensoren und Aktoren) eingeschalten werden. Zur Auswahl des Handbetriebes muss nur der Betriebsmodusschalter auf „HAND“ eingestellt und nachfolgend per Taster bestätigt werden. Die erfolgreiche Auswahl wird durch das Aufleuchten der zugehörigen Signalleuchte auf der Front des Schaltschrankes symbolisiert.

Nach der Wahl des Handmodus verbleibt die Anlage zunächst im Ruhezustand. Die beiden Achsen befinden sich an der Ausgangsposition, die im Leerlauf hergestellt ist. Um die Positioniereinheit in Bewegung zu setzen, ist nun eine Nutzereingabe nötig.

An der Frontseite des Schaltschrankes befindet sich ein Vierfachtaster mit Pfeilen in X- und Z-Richtung. Mittels der Taster kann per Druck die jeweilige Achse bewegt (gejoggt) werden. Dies geschieht so lange, bis der Taster wieder losgelassen wird oder eine der Endlagen erreicht ist. Bei Betätigung eines Tasters fahren die Achsen jedoch nicht mit voller Geschwindigkeit an, sondern beschleunigen erst langsam. Auch die Beschleunigung beim Loslassen bzw. Abbremsen einer Achse ist verringert gegenüber dem Automatikmodus. Über ein Potenziometer rechts neben den vier Bewegungstastern kann die Fahrtgeschwindigkeit reguliert werden.

Nach manuellem Navigieren zu den Ablagepositionen besteht an diesen die Möglichkeit, den Greifer einzusetzen. Nun muss jedoch jeder einzelne Schritt, also Umschwenken zur Ablage, Greifen und wieder Loslassen eines Transportobjektes per Druckknopf getriggert werden.

Weiterhin ist als Randbedingung im Handbetrieb vorgesehen, dass in den äußeren Bereichen des Positioniersystems zum einen nur geringere Geschwindigkeiten gefahren werden können als auch, dass die Beschleunigung der Achsen in diesen Bereichen gedämpft ist, um zu verhindern, dass die Schlitten auf den jeweiligen Achsen über die Endlagen hinaus Abbremsen und mit den harten Stoppelementen am äußersten Ende der Achsen kollidieren. Im Handmodus sind keine virtuellen Hindernisse vorgesehen auf dem Fahrtweg des Greifers, da kein Mehrwert aus dem manuellen Umfahren gewonnen wird und maximal die Koordination des Nutzers trainiert werden kann. Programmatisch wäre an dieser Stelle kein Mehrwert zu erreichen, falls der Nutzer per Tastendruck Kollisionen mit Hindernissen verhindern sollte.

Nach Wiederabwahl des Handmodus bewegt sich die Anlage zurück in ihre Ausgangsposition (es findet wie auch schon im Automatikmodus ein Homing statt).

**Sicherheitsbezogene Randbedingungen:** Als letzten Unterpunkt in diesem Teilkapitel soll noch ein Überblick zu den Sicherheitsmaßnahmen der Anlage gegeben werden. Für die detaillierte Darstellung und Projektierung des Sicherheitskonzeptes wird an dieser Stelle

auf das Kapitel Sicherheitskonzept im dritten Teil Bachelorthesis verwiesen.

Allgemein wird durch jegliche Sicherheitsmaßnahmen an und um die Laboranlage herum sichergestellt, dass weder Mensch noch Maschine Schaden nehmen kann. Grundlegend muss gewährleistet sein, dass das Positioniersystem nicht außerhalb seiner vorgesehenen Aufgaben und Abläufe agieren kann. Dazu sind kurz vor jedem Ende der zwei Achsen des Systems induktive Endlagesensoren verbaut. Diese lösen aus, wenn ein Schlitten auf einer Achse das Ende eines Fahrbereiches einer Achse erreicht hat. Ist dies der Fall, wird die betreffende Achse umgehend abgebremst. Diese Sicherheitsmaßnahme ist zum einen im Handbetrieb, aber auch im möglichen Fehlerfall von höchster Relevanz. Dem Anlagennutzer darf zum einen nicht eine Achse im manuellen Betrieb auf einen der Puffer am Ende des befahrbaren Weges auffahren lassen, zum anderen muss die Anlage in egal welcher Situation (was auch den Fehlerfall einschließt) unweigerlich an den Endlagesensoren zum Stillstand abbremsen.

Es können weiterhin aber auch im normalen Betriebsablauf Fehler oder Notfälle entstehen, die dem System nicht durch das Erreichen von einem oder mehreren Endlagepositionen bekannt werden. So muss verhindert werden, dass eine sich im Bereich der Positioniereinheit befindliche Person nicht in den Prozess physisch eingreifen kann. Dazu ist, wie bereits zum Eingang des Unterkapitels erwähnt, ein Lichtvorhang vor dem Positionier- bzw. Fahrbereich der Laboranlage installiert. Wird der Vorhang durchbrochen, löst dies ein Signal aus, welches dazu führt, dass die Anlage schnellstmöglich abbremst und zum Stillstand kommt. Es handelt sich folglich um eine Not-Halt-Funktionalität. Selbige kann auch von einer Person manuell ausgelöst werden, auch ohne dass der Lichtvorhang ein Eindringen in den Positionierprozess detektiert hat. Sowohl auf der Linken als auch auf der rechten Seite des Systems ist ein einrastender Not-Halt Taster montiert. Falls Fehler oder Notfall vorliegt, kann dieser betätigt werden.

Damit das mehrachsige Positioniersystem nach einem Fehler wieder seinen Betrieb aufnehmen kann, muss der Fehler zunächst beseitigt werden und anschließend kann über zweifaches Drücken eines dafür deklarierten Tasters am Schaltschrank die Anlage wieder freigegeben werden. Nach dieser Handlung setzt die Anlage entsprechend ihres aktuell ausgewählten Betriebsmodus ihren ursprünglichen Ablauf fort.

Auch durch visuelle Signale soll die Sicherheit von Menschen, die sich in der Nähe oder an der Maschine befinden, verbessert werden. Dazu wird eine Signalampel genutzt, die bei Bewegung von Achsen blinkt und im Eingeschalteten Zustand des Positioniersystems immer mindestens in einer Farbe leuchtet. Konkrete Umsetzungen werden auch hierzu im Kapitel zum Sicherheitskonzept beschrieben.

---

## 4 Projektierung

Nachdem im letzten Kapitel die Anforderungsphase des mehrachsigen Positioniersystems behandelt wurde, schließt sich nun die Design- bzw. Modellierungsphase in diesem Kapitel an.

Der Entwicklungsprozess unter den Gesichtspunkten der Projektierung umfasst folgende fünf Kernabschnitte, die es zu untersuchen gilt:

- **Kontextanalyse:** Finden der Systemgrenzen und Ermittlung von Nachbarsystemen.
- **Anwendungsfallspezifikation:** Identifizierung der Systemprozesse und anschließende Präzisierung.
- **Verhaltensspezifikation:** Modellierung des Systemverhaltens.
- **Partitionierung:** Untergliederung des Systems in logische Sinnesabschnitte zur Verringerung der Komplexität.
- **Testspezifikationen:** Festlegung von Prüfkriterien zur Bestätigung der Anforderungsumsetzung.

Die drei letzten Abschnitte des Kapitels zur Projektierung dienen als direkte Vorbereitung für die Implementationsphase des Positioniersystem. Es wird zum einen kurz die Entwicklung des Stromlaufplanes dargestellt, welcher die Grundlage bildet für die elektrotechnische Inbetriebnahme der Laboranlage. Mit beschrieben in diesem Unterabschnitt wird auch die Netzwerkstruktur zwischen sowohl den Steuerungskomponenten unter sich als auch (übergeordneten) Nachbarsystemen.

Nachfolgend soll in tabellarischer Form das Datenmodell aufgestellt werden, welches zusammen mit den UML Diagrammen aus dem Entwicklungsprozess die Basis für die Softwareimplementation darstellt.

An letzter Stelle in Projektierungsteil der Arbeit soll das Bedienkonzept entwickelt werden. Dabei wird sowohl auf die Bedienung über Eingabetaster an der Anlage selbst eingegangen, als auch eine dezentralisierte Bedienung über Netzwerkfähige Endgeräte, wie unter anderem die Laborcomputer, die sich im selben Raum wie das Positioniersystem befinden.

### 4.1 Kontextanalyse

Ziel der Kontextanalyse ist die Abgrenzung des Kontexts bzw. das Finden von Systemgrenzen. Bei dem mehrachsigen Positioniersystem handelt es sich um ein sogenanntes eingebettetes System (engl. Embedded System). Diese kommunizieren meist stark mit ihrer Umwelt bzw. sind meist stark mit dieser verankert. So auch hat das Positioniersystem Schnittstellen, über die eine Kommunikation mit Nachbarsystemen stattfindet. In der Systemkonzeption muss folglich geklärt werden, wo genau die Systemgrenzen liegen.

Weiterhin findet in der Kontextabgrenzung auch die Identifizierung von Nachbarsystemen statt.

Zuerst muss geklärt werden, ob die Systemumgebung dynamischer Natur ist, das heißt, dass Nachbarsysteme wechseln bzw. das System nicht umgebungstreu ist. Handelt es sich im Gegensatz dazu um ein System mit stabiler Umgebung, ist die Darstellung von Nachbarsystemen simpel und kann nachfolgend im entsprechenden Diagramm dokumentiert werden. Da das mehrachsige Positioniersystem fest in den Laborraum integriert ist, und alle Nachbarsysteme bereits bekannt sind, wird in der Analyse von einer statischen Umgebung ausgegangen. Die sich anschließende Liste zeigt alle derzeitigen Nachbarsysteme, in die das mehrachsige Positioniersystem eingebettet ist.

- Laborcomputer
- Ablageschale/Aufnahmeschale (Ablagepositionen)
- Externe Industriesteuerungen
- Augmented Reality Server
- Verwaltungsschalen (Digitaler Zwilling)
- mögliche spätere Erweiterung: Förderbänder
- mögliche spätere Erweiterung: Vorratslager (statt Aufnahmeschale)
- mögliche spätere Erweiterung: Lagermagazin(e) (statt Ablageschale)

Ist die Identifizierung der Nachbarsysteme abgeschlossen, kann mit der Kontextanalyse begonnen werden. Der Kontext unterteilt sich in den logischen Kontext und den physikalischen Kontext. Der **logische Kontext** betrachtet die Kommunikation mit den Nachbarsystemen, wohingegen der **physikalische Kontext** auf die Kommunikationshardware fokussiert ist.

Für die Dokumentation der logischen Kontextabgrenzung bietet sich das Anwendungsfalldiagramm der UML (Unified Modeling Language) an. Dabei handelt es sich um die allgemein gängige Form für diese Aufgabe. Das Anwendungsfalldiagramm ist geeignet an dieser Stelle für die Modellierung, da für die Darstellung noch keine detaillierten Entscheidungen über die Schnittstellen getroffen werden müssen. Es besitzt die Fähigkeit das zu modellierende System und seine Nachbarsysteme in Beziehung darzustellen und deren Kommunikation grundlegend anzudeuten. Abbildung 1 zeigt die logische Kontextabgrenzung des mehrachsigen Positioniersystems zu den bereits aufgezählten Nachbarsystemen mittels des Anwendungsfalldiagramms.

Noch nicht erwähnt waren bisher die Akteure des Positioniersystems. Akteure eines eingebetteten Systems sind Sensoren, E/A-Geräte, Nachbarsysteme und die Zeit. Sie

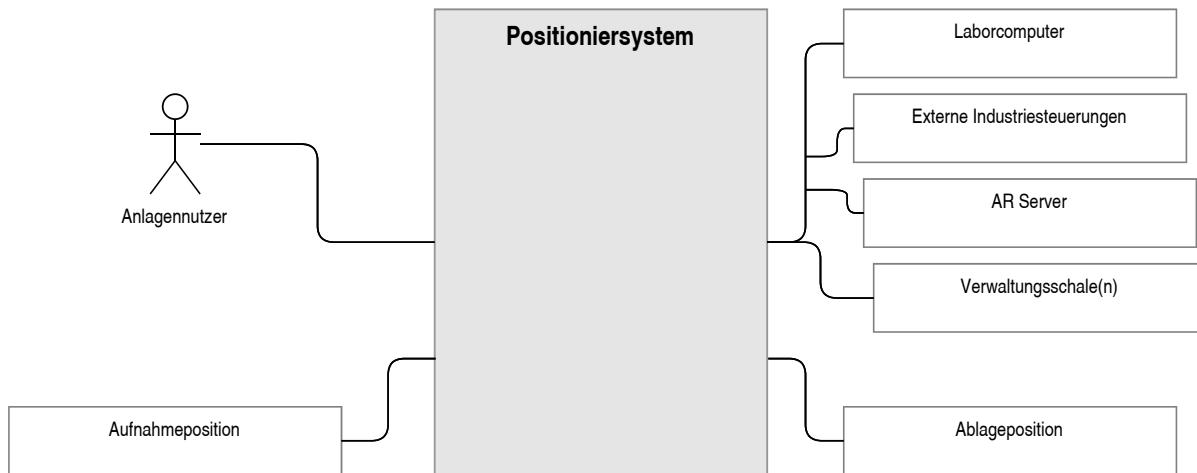


Abbildung 1: Logische Kontextabgrenzung des mehrachsigen Positioniersystems als Anwendungsfalldiagramm

befinden sich grundsätzlich außerhalb des Systems.

Die zu modellierende Laboranlage besitzt folglich mehrere Nachbarsysteme, die als Akteure bezeichnet werden können. Weiterhin sind Menschen, die in Kontakt mit dem System stehen, relevant. Diese gelten auch als Akteure und werden als Strichfigur im Anwendungsfalldiagramm aufgenommen. Der Anlagennutzer des Positioniersystems ist als Akteur auf der linken Seite der Abbildung 1 aufgeführt. Die Nachbarsysteme auf der rechten oberen Seite besitzen wiederum Akteure, die an dieser Stelle jedoch nicht dargestellt sind, da diese mit dem mehrachsigen Positioniersystem nur indirekt über die Nachbarsysteme kommunizieren.

Es kann abschließend festgehalten werden, dass der logische Kontext beantwortet, welche Akteure für das System existieren. Es besteht die Notwendigkeit nach diesen zu suchen, und sie in Form des Anwendungsfalldiagrammes im Bezug zum Positioniersystem darzustellen. Nach der Aufstellung des logischen Kontexts der Laboranlage wird nun darauf aufbauend fortgesetzt mit der physikalischen Kontextabgrenzung. Im Unterschied zum logischen Kontext wird die Fragestellung erweitert um die konkreten Einflüsse der Akteure auf das System. Es gilt zu untersuchen, wie die Kommunikation zwischen den Akteuren und dem mehrachsigen Positioniersystem aufgebaut ist. Dazu bietet es sich an das Verteilungsdiagramm der UML zu nutzen. Wie auch schon bei der logischen Kontextabgrenzung wird das System Positioniereinheit im Zentrum zwischen den Akteuren als zentraler Knoten dargestellt. Die Nachbarsysteme werden ringsherum als eigenständige Knoten aufgeführt. In der folgenden Abbildung der physikalischen Kontextabgrenzung wird auch der Anlagennutzer als Nachbarsystem betrachtet, um mehr Freiräume in der Darstellung der Schnittstelle zwischen diesem und dem Positioniersystem zu ermöglichen. Abbildung 2 zeigt das Verteildiagramm der Positioniereinheit und seiner Nachbarsysteme zur Beantwortung

der Frage nach dem physikalischen Kontext.

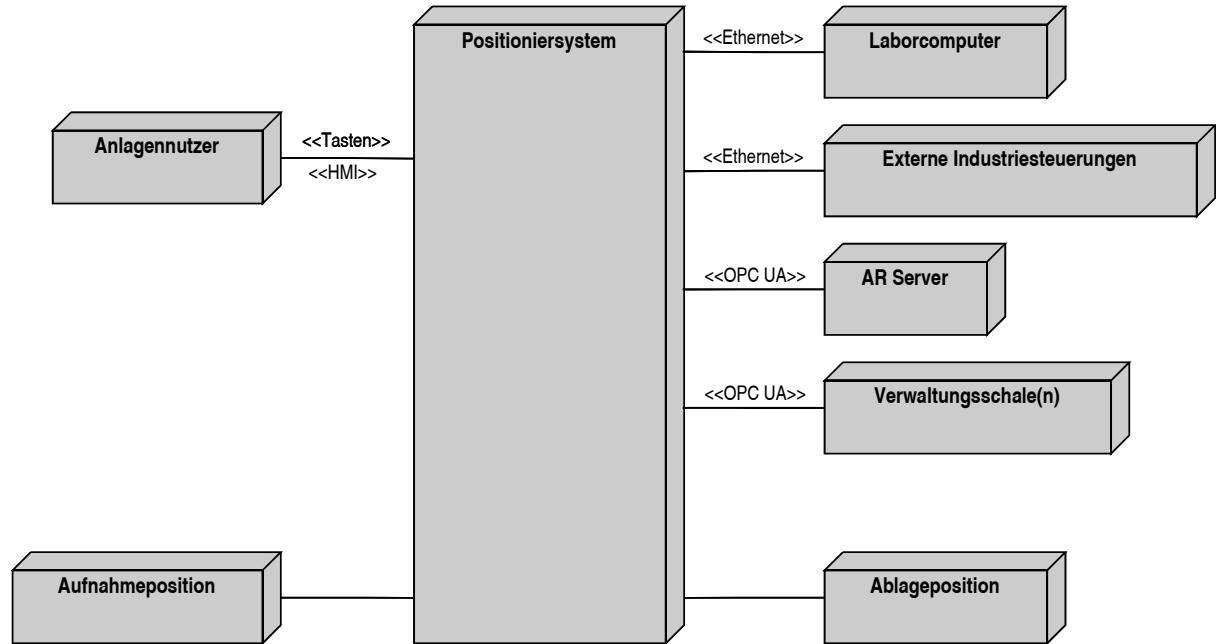


Abbildung 2: Physikalische Kontextabgrenzung des mehrachsigen Positioniersystems als Verteilungsdiagramm

Im Unterschied zu Abbildung 1 werden nun an den Verbindungen zwischen den Systemen Stereotypen mit aufgeführt, falls die Hardware und Kommunikation der untersuchten Systeme bereits bekannt ist. Mit Hilfe des Verteilungsdiagrammes wird die Frage beantwortet, wie die Akteure auf das Positioniersystem Einfluss nehmen. Es kann hier bereits aus den Anforderungen entnommen werden, wie der Anlagennutzer mit dem System interagiert und wie Datenaustausch zwischen der internen Steuerung und externen Industriesteuerungen stattfindet. Auch die Programmierschnittstelle ist bereits vorgegeben. Aus den Anforderungen der Labormitarbeiter geht weiterhin hervor, dass für Verwaltungsschalen aber auch den Augmented Reality (AR) Server Prozessdaten via OPC UA Schnittstelle bereitgestellt werden sollen.

Die Stereotypen für z. B. den Anlagennutzer als Akteur sind somit „Tasten“ und „HMI“, da dieser auf diesem Weg mit der Laboranlage kommuniziert bzw. interagiert. Human Machine Interface (HMI) ist zu deutsch eine Mensch-Maschine-Schnittstelle (MMS). Für die Interaktion mit dem Positioniersystem ist zum einen die grundlegende Steuerung über Taster an der Front des Schaltschranks geplant, weiterhin soll diese erweitert werden um eine Kommunikationsschnittstelle, die auf touch-basierten Displays beruht. Dabei handelt es sich zum einen um ein fest angebundenes Monitor. Aus den Anforderungen geht

zusätzlich hervor, dass die Steuerung auch per Smartphone oder Tablet erfolgen sollte.

In diesem Unterkapitel ist die Kontextabgrenzung des mehrachsigen Positioniersystems analysiert worden. Dabei wurde unterschieden zwischen der logischen- und der physikalischen Kontextabgrenzung. Dazu mussten zunächst die Nachbarsysteme ermittelt werden, die auch als Akteure bezeichnet werden. Der wesentliche Unterschied zwischen dem logischen und dem physikalischen Kontext besteht im Detailgrad der Analyse, welcher sich auch in der Darstellung wiederfindet. Für die logische Kontextabgrenzung empfiehlt sich das Anwendungsfalldiagramm der UML. Die physikalische Kontextabgrenzung erfolgte über das Verteilungsdiagramm. Weiteres betrachtet dabei als Erweiterung auch die Hardware und Art der Kommunikation über die eingezeichneten Schnittstellen. Dies wird als Stereotyp bezeichnet, welcher zwischen den Systemen, die auch als Knotenpunkte bezeichnet werden, dargestellt ist.

## 4.2 Anwendungsfallspezifikation

Nach dem Abschließen der Festlegung des Kontextes des mehrachsigen Positioniersystems folgt nun die Identifizierung von Systemprozessen. Der Findungsprozess erfolgt über die Anwendungsfallanalyse. Dabei wird ein System als Black-Box betrachtet, um möglichst gute Systemprozesse zu finden, ohne sich von internen Gegebenheiten des Systems beeinflussen zu lassen.

Die Anwendungsfallspezifikation wird in dieser Arbeit in zwei Unterkapitel eingeteilt. Ersteres beschäftigt sich mit dem Finden und Entwickeln von Systemprozessen. Das zweite Unterkapitel hat zum Ziel die Systemprozesse zu präzisieren und diese dann übersichtlich darzustellen.

### 4.2.1 Entwicklung der Systemprozesse

Die Anwendungsfallanalyse baut auf dem Anwendungsfalldiagramm aus Abbildung 1 auf. Dabei findet auch an dieser Stelle eine Unterteilung in zwei Abschnitte statt. Im ersten Schritt werden die Akteure aus den Diagrammen des Unterabschnitt 4.1 geprüft und um eventuelle Akteure ergänzt, die bis zu diesem Zeitpunkt nicht erkannt wurden. Diese werden zunächst in die Kontextabgrenzung mit aufgenommen, bevor im folgenden Abschnitt die Anwendungsfallanalyse beginnt.

Im zweiten Schritt werden die Erwartungen der Akteure an das System untersucht. Aus dieser Analyse erfolgt die Ableitung von möglichen Systemprozessen. Dieser Abschnitt hat es folglich als Ziel, die Frage nach den durch die Akteure geforderten Voraussetzungen zu beantworten.

Für die Entwicklung der Systemprozesse wird folglich auf den logischen Kontext zurückgegriffen, da dieser die Akteure des Systems bereits im Anwendungsfalldiagramm (siehe Abbildung 1) zeigt. Da es nur um die Frage nach den Akteuren und ihren Erwartungen geht und dabei die Hardware und die Ausprägung der Kommunikation des Positioniersystems nicht relevant ist, spielen der physikalische Kontext und dessen Ergebnisse keine Rolle.

Die Abbildung 3 zeigt die Systemprozesse, die aus der Anlagenbeschreibung modelliert werden. Es ist ersichtlich, dass gezeigtes Anwendungsfalldiagramm eine Erweiterung der Abbildung 1 aus dem vorhergegangenen Unterabschnitt ist.

Neu daszugekommen sind die Anwendungsfälle, die durch den Anlagennutzer ausgelöst werden können. Konkret handelt es sich also um die zwei auswählbaren Betriebsmodi und den Not-Halt. Weiterhin ist der Transport von Gegenständen im Anwendungsfalldiagramm aufgeführt. Es handelt sich dabei um den grundsätzlichen Nutzen des Positioniersystems. Es ist wichtig zu berücksichtigen, dass dieser Anwendungsfall von den vorher genannten drei Anwendungsfällen/Betriebszuständen abhängig ist. Zuletzt findet sich noch die Bereitstellung von Prozessdaten im Diagramm wieder.

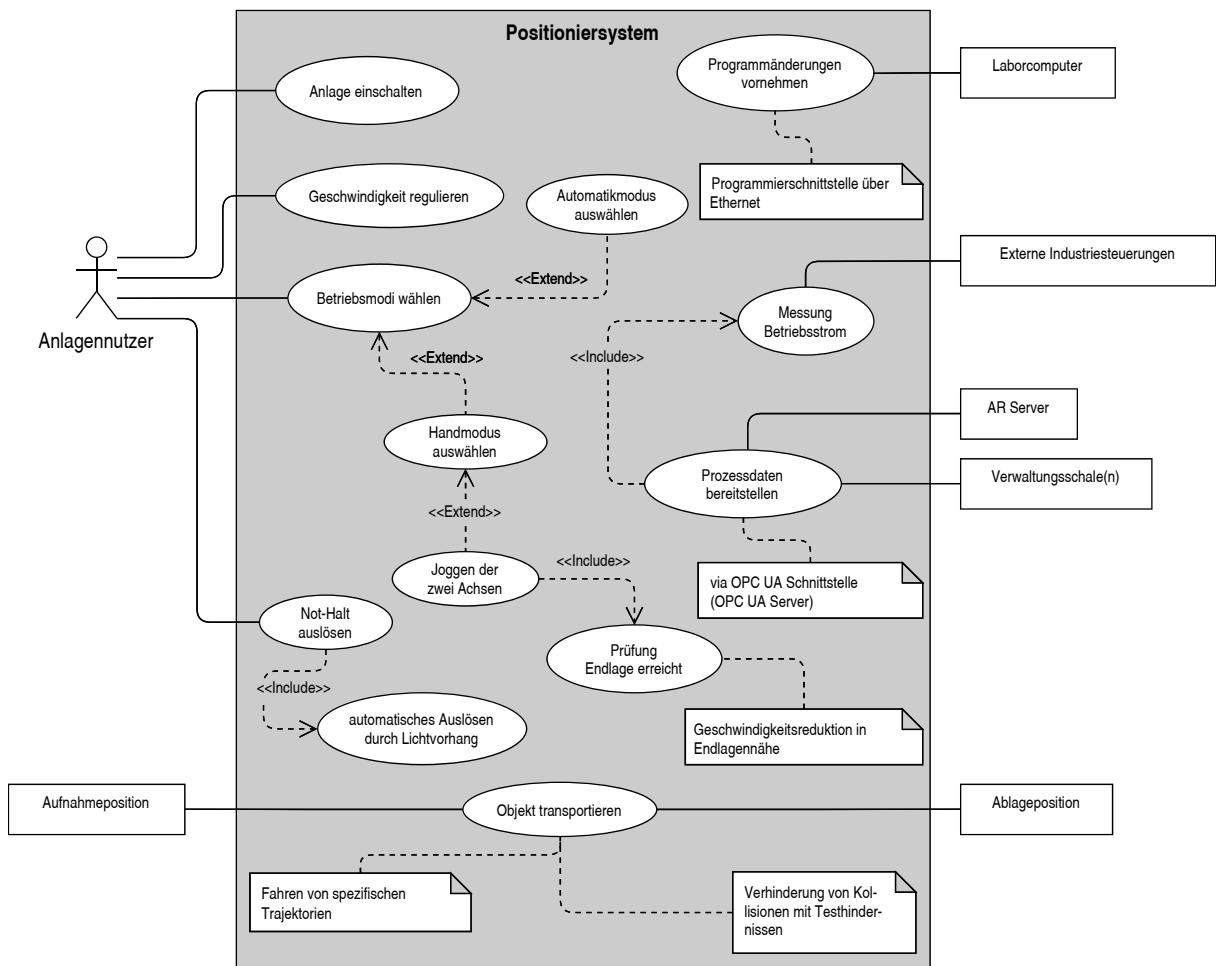


Abbildung 3: Anwendungsfalldiagramm des Positioniersystems

#### 4.2.2 Präzisierung der Systemprozesse

Dieser Unterabschnitt greift die Systemprozesse aus der Anwendungfallanalyse des vorhergegangenen Unterabschnittes noch einmal auf und verfeinert diese. Im Folgenden wird zunächst die genutzte Methodik zur Spezifizierung der Systemprozesse vorgestellt.

Für die Spezifikation von Systemprozessen empfiehlt es sich die Anwendungfallbeschreibung als Mittel zur Dokumentation zu nutzen. Diese sollte in Form von Tabellen erfolgen. Dabei wird jeder einzelne Akteur in seiner eigenen Tabelle dargestellt. Bei den relevanten Tabelleneinträgen handelt es sich um die Zeilen Name, Akteur, auslösendes Ereignis, Kurzbeschreibung, Vorbedingungen, essenzielle Schritte, Ausnahmefälle, Nachbedingungen, Zeitverhalten, Verfügbarkeit und Kommentare/Fragen.

Sowohl der **Name** als auch der **Akteur** wird dabei aus dem Anwendungsfalldiagramm aus Abbildung 3 übernommen. Es sind am Ende alle Akteure aus dem Anwendungsfal-

diagramm tabellarisch aufgenommen. Das Feld **auslösendes Ereignis** beschreibt den Initiator des Anwendungsfalls. Der nächste Eintrag, die **Kurzbeschreibung** ist eine in zwei bis vier Sätzen dokumentierte wörtliche Beschreibung des Prozesses und dient zur Darstellung seines Kerns. Das Feld **Vorbedingungen** enthält zusammengefasst alle Voraussetzungen, die für die Ausführung des Anwendungsfalls nötig sind. Der nächste Eintrag stellt den wichtigsten Schritt in der Dokumentation des Anwendungsfalls dar. Dieser wird unterteilt in zwei weitere Felder, die im direkten Bezug zueinander stehen. Es werden Auf der einen Seite Ereignisse aufgenommen, die während der Standardausführung des Prozesses auftreten bzw.. auftreten können und auf der anderen Seite die Reaktionen des Systems auf diese Ereignisse. Das Feld **Ausnahmefälle** betrachtet alle Fehler und Ausnahmesituationen, die Abweichend von der Standardausführung auftreten können. Die **Nachbedingungen** sind analog zu den Vorbedingungen zu dokumentieren und beschreiben den Endzustand des Prozesses nach einer Standardausführung. In den Punkten **Zeitverhalten** und **Verfügbarkeit** können NFAs des Anwendungsfalls festgehalten werden. Zuletzt, im Feld **Kommentare/Fragen**, können Anmerkungen und Probleme aufgenommen werden, falls diese existieren. Es gilt diese bis zur Fertigstellung des Systems zu eliminieren, so dass dieses Feld leer bleiben kann. Es handelt sich folglich um ein temporäres Hilsmittel.

Es bietet sich im Normalfall an zwei Abstraktionsebenen in der Darstellung der Systemprozesse zu nutzen. Dazu gehört eine detaillierte Dokumentation für die Prozessentwickler und ein abstrakter Überblick für Manager und weniger stark involvierte Personen. Auf dieses Überblick wird jedoch an dieser Stelle verzichtet, da alle für das System relevanten Personen und Stakeholder ausreichend mit der Positioniereinheit und der Umsetzung eines solchen Systems vertraut sind. Im Anhang kann jedoch trotzdem zu jedem Akteur auch ein Überblick gefunden werden.

Es folgen nun die tabellarischen Darstellungen zu den Anwendungsfällen nach beschriebinem Muster.

<b>Name</b>	Objekt transportieren
<b>Akteur</b>	Aufnahmeposition
<b>Auslösendes Ereignis</b>	Ein neues Transportobjekt liegt auf Aufnahmeposition bereit
<b>Kurzbeschreibung</b>	Das Objekt wird mit einem Greifer von der Aufnahmeposition hochgenommen. Anschließend fährt das Positioniersystem eine Hindernissausweichende Trajektorie zur Ablageposition. Dort wird das Objekt wieder losgelassen.

<b>Vorbedingungen</b>	<ul style="list-style-type: none"> <li>• der Automatikmodus wurde ausgewählt</li> <li>• Aufnahmeschale ist mit Transportobjekt bestückt</li> </ul>	
<b>Essenzielle Schritte</b>	<b>Intention der Systemumgebung</b>	<b>Reaktion des Systems</b>
	Anlagennutzer will das System einschalten	Systemkomponenten werden mit Spannung versorgt und sind betriebsbereit
	Anlagennutzer will, dass die Positioniereinheit vollautomatisch Transportgüter von der Aufnahmeposition zur Ablageposition befördert	Laboranlage beginnt Objekte von der Aufnahmeposition zu greifen und zu transportieren
	Anlagennutzer will das System auf Grund einer Gefahrensituation anhalten	Die Laboranlage bremst bis zum Stillstand ab und erwartet eine Bestätigung, dass die Gefahren- bzw. Fehlersituation beseitigt ist
	Anlagennutzer will die Fahrgeschwindigkeit regulieren	Die Achsen des Systems bewegen sich entsprechend der analogen Nutzereingabe schneller bzw. langsamer
<b>Ausnahmefälle</b>	Anlagennutzer will die Laboranlage stoppen	Der Automatikbetrieb beendet seinen aktiven Zyklus und wird dann abgewählt, woraufhin die Anlage stoppt
	<ul style="list-style-type: none"> <li>• Defektbedingte Abschaltung der Anlage</li> </ul>	
<b>Nachbedingungen</b>	Der Automatikmodus ist beendet und die Anlage kann abgeschaltet werden.	

<b>Zeitverhalten</b>	schnell und effizient
<b>Verfügbarkeit</b>	maximal ein Systemausfall in 10.000h
<b>Kommentare/Fragen</b>	---

Tabelle 26: Anwendungsfallbeschreibung - Systemprozess: Objekttransport

<b>Name</b>	manuelle Funktionsausführung	
<b>Akteur</b>	Anlagennutzer	
<b>Auslösendes Ereignis</b>	Der Vierwegetaster oder die Greifertaster an der Schaltschrankfront werden betätigt	
<b>Kurzbeschreibung</b>	Die dem betätigten Taster zugehörige Achse bewegt sich entsprechend der angezeigten Richtung auf diesem Taster. Wird einer der dem Greifer zugehörigen Taster gedrückt, schwenkt der Greifarm um bzw. der Greifer öffnet/schließt	
<b>Vorbedingungen</b>	<ul style="list-style-type: none"> <li>der Handmodus wurde ausgewählt</li> <li>mindestens einer der vier Richtungstaster auf dem Vierwegetaster wird gedrückt oder einer der beiden dem Greifer zugehörigen Taster</li> </ul>	
<b>Essenzielle Schritte</b>	<b>Intention der Systemumgebung</b> Anlagennutzer will das System einschalten Anlagennutzer will auf der horizontalen Achse positionieren	<b>Reaktion des Systems</b> Systemkomponenten werden mit Spannung versorgt und sind betriebsbereit Das Positioniersystem erwartet Tastereingabe, die zum Joggen der X-Achse führt

	Anlagennutzer will auf der vertikalen Achse positionieren	Das Positioniersystem erwartet Tastereingabe, die zum Joggen der Z-Achse führt
	Anlagennutzer will den Greifarm umschwenken	Das Positioniersystem erwartet Tastereingabe, die zum Schwenken des Greifarms führt
	Anlagennutzer will ein Objekt greifen loslassen	Das Positioniersystem erwartet Tastereingabe, die zum Öffnen bzw. Schließen des Greifers führt
	Anlagennutzer will das System auf Grund einer Gefahrensituation anhalten	Die Laboranlage bremst bis zum Stillstand ab und erwartet eine Bestätigung, dass die Gefahren- bzw. Fehlersituation beseitigt ist
	Anlagennutzer will die Fahrgeschwindigkeit regulieren	Die Achsen des Systems bewegen sich entsprechend der analogen Nutzereingabe schneller bzw. langsamer
	Anlagennutzer will die Laboranlage stoppen	Die Anlage ist gestoppt und der Handmodus wird abgewählt
<b>Ausnahmefälle</b>	<ul style="list-style-type: none"> <li>Defektbedingte Abschaltung der Anlage</li> </ul>	
<b>Nachbedingungen</b>	Anlage ist abgeschaltet	
<b>Zeitverhalten</b>	Keine Ansprüche an das Zeitverhalten (Handmodus wird nur zu Testzwecken genutzt)	
<b>Verfügbarkeit</b>	maximal ein Systemausfall in 10.000h	
<b>Kommentare/Fragen</b>	---	

Tabelle 27: Anwendungsfallbeschreibung - Systemprozess: manuelle Funktionsausführung

<b>Name</b>	Programmänderungen Vornehmen	
<b>Akteur</b>	Laborcomputer	
<b>Auslösendes Ereignis</b>	System wird mit neuem Programmcode bespielt	
<b>Kurzbeschreibung</b>	Über eine Ethernetschnittstelle ist das Positioniersystem mit dem Labornetzwerk verbunden. Von Geräten aus dem selben Netzwerk kann auf das System zugegriffen werden bzw. Änderungen an diesem Programmcode vorgenommen werden.	
<b>Vorbedingungen</b>	<ul style="list-style-type: none"> <li>Steuerungskomponenten des Systems sind eingeschalten und im Labornetzwerk findbar</li> <li>Computer des Programmentwicklers befindet sich im selben Netzwerk wie das Positioniersystem</li> </ul>	
<b>Essenzielle Schritte</b>	<b>Intention der Systemumgebung</b>	<b>Reaktion des Systems</b>
	Anlagennutzer will das System einschalten	Systemkomponenten werden mit Spannung versorgt und sind betriebsbereit
	Anlagennutzer will die Steuerung des Systems mit neuem Programmcode bespielen	In Maschinencode übersetzter Programmcode wird über die Ethernetschnittstelle zur Steuerung übertragen
Anlagennutzer das neue Programm aktivieren/ausführen	System startet neu und ist betriebsbereit	

<b>Ausnahmefälle</b>	<ul style="list-style-type: none"> <li>• Labornetzwerk ist ausgefallen oder verhindert die Kommunikation</li> <li>• Defektbedingte Abschaltung der Anlage</li> </ul>
<b>Nachbedingungen</b>	Anlage ist erneut betriebsbereit
<b>Zeitverhalten</b>	---
<b>Verfügbarkeit</b>	Das System sollte jederzeit von jedem Computer im selben Netzwerk erreichbar sein
<b>Kommentare/Fragen</b>	---

Tabelle 28: Anwendungsfallbeschreibung - Systemprozess: Programmänderungen vornehmen

<b>Name</b>	Prozessdaten bereitstellen	
<b>Akteur</b>	OPC UA Server	
<b>Auslösendes Ereignis</b>	System ist aktiv	
<b>Kurzbeschreibung</b>	Die Steuerung des Positioniersystems übernimmt zusätzlich die Aufgabe als OPC UA Server, über welchen per OPC Schnittstelle (ethernetbasiert) Daten aus dem Systemprozess bereitgestellt werden.	
<b>Vorbedingungen</b>	Steuerungskomponenten des Systems sind eingeschalten	
<b>Essenzielle Schritte</b>	<b>Intention der Systemumgebung</b> Anlagennutzer will das System einschalten	<b>Reaktion des Systems</b> Systemkomponenten werden mit Spannung versorgt und sind betriebsbereit

	Anlagennutzer will Prozessdaten der Laboranlage erhalten	Die Steuerung stellt Daten via OPC UA Schnittstelle bereit (ein OPC Client kann diese entgegnehmen)
	Anlagennutzer will Prozessdaten extern weiterverwenden	System kommuniziert mit z. B. dem AR Server oder einer Verwaltungsschale, welche als OPC Client Daten entgegnehmen
<b>Ausnahmefälle</b>	<ul style="list-style-type: none"> <li>• Labornetzwerk ist ausgefallen oder verhindert die Kommunikation</li> <li>• Defektbedingte Abschaltung der Anlage</li> </ul>	
<b>Nachbedingungen</b>	System ist weiterhin aktiv	
<b>Zeitverhalten</b>	Prozessdaten sollen in echtzeit abgefragt werden können	
<b>Verfügbarkeit</b>	Prozessdaten sollten zu jeder Zeit abgefragt werden können	
<b>Kommentare/Fragen</b>	---	

Tabelle 29: Anwendungfallbeschreibung - Systemprozess: Prozessdaten bereitstellen

### 4.3 Verhaltensspezifikation

Dieses Unterkapitel beschäftigt sich mit der Modellierung des Systemverhaltens. Anschließend an die Systemanalyse, ist die nun folgende Modellierung Teil der detaillierten Systemanalyse. Die Verhaltensspezifikation beinhaltet sämtliche Informationen zum Verhalten des gesamten Systems und dessen Systemprozesse. Als Basis dienen die in Unterabschnitt 4.2.2 dargestellten Anwendungsfallbeschreibungen. Ziel dieses Abschnittes ist es ein bzw. mehrere Zustandsdiagramme aus den Informationen der Anwendungsfallbeschreibungen zu entwickeln. Auf Grundlage der Tabellen aus dem vorhergegangenen Kapitel entsteht eine Verhaltensbeschreibung der zugrundeliegenden Systemprozesse. Zunächst erfolgt eine methodische Erläuterung zur Konstruktion eines solchen Zustandsdiagramms. Die Konstruktion des Zustandsdiagrammes kann in folgende sieben Schritte untergliedert werden:

- Zunächst müssen sämtliche Ereignisse bzw. wesentliche Schritte des Prozesses auf Unterbrechbarkeit geprüft werden. Unterbrechbare Elemente werden anschließend als **Aktivitäten** des Zustandsdiagramms modelliert. Ununterbrechbare Elemente sind als **Aktionen** des Zustandsdiagramms zu definieren.
- **Aktivitäten** werden in den Zuständen des Diagrammes abgebildet. Dazu wird eine solche Aktivität hinter dem Schlüsselwort „do“ aufgeschrieben. Es ist hilfreich einen prägnanten Namen zu wählen. (Übergänge, die in den Zustand führen, sind aus dem Ereignis des jeweiligen Anwendungsfalldiagramms zu entnehmen.)
- **Aktionen** werden als Übergänge eingezeichnet und mit einem Ereignis beschriftet. (Am Ende eines Überganges wird die entsprechende Aktion eingezeichnet.)
- Verbleibende freie Enden bzw. Anfänge werden auf potentielle Start- oder Endzustände untersucht. Bei der Ermittlung eines solchen Zustands muss dieser entsprechend der Symbolik des Zustandsdiagramms mit dargestellt werden.
- Falls dennoch frei Übergangsenden verbleiben, müssen Zustände gefunden werden, auf welche diese verweisen. Zunächst sollten existierende Zustände geprüft werden. (Ein Ereignis kann auch an mehreren Zuständen hängen.) Wird kein Zustand gefunden, muss ein neuer Zustand erfunden werden.
- Es besteht die Möglichkeit Regionen oder auch zusammengesetzte Zustände zu definieren, um die Lesbarkeit zu erhöhen.
- Den letzten Schritt der Konstruktion stellt die Anreicherung der Übergänge mit NFAs dar. Dazu gehört unter anderem auch das Zeitverhalten aus den Anwendungsfallbeschreibungen.

Die nun Folgenden Grafiken zeigen die Zustandsdiagramme zu den ermittelten Systemprozessen. Die Modellierung dieser folgt der zuvor beschriebenen Methodik. Auf Grund der starken Abweichungen des Systemverhaltens im Automatikmodus und im Handmodus, werden beide Betriebsmodi in getrennten Diagrammen dargestellt, auch wenn der zugrundeliegende Prozess gleich ist.

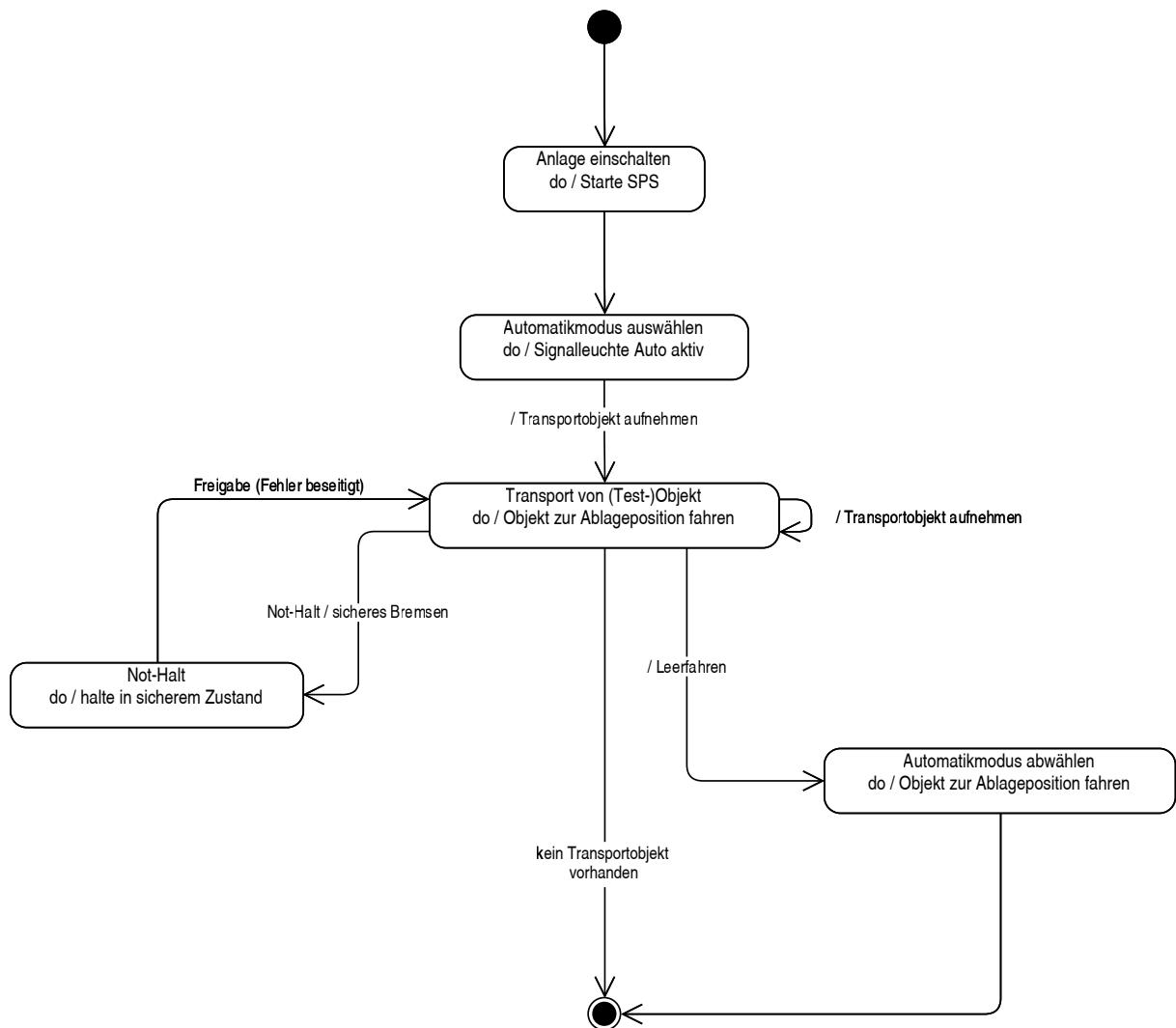


Abbildung 4: Zustandsdiagramm - Systemprozess: Objekttransport im Automatikmodus

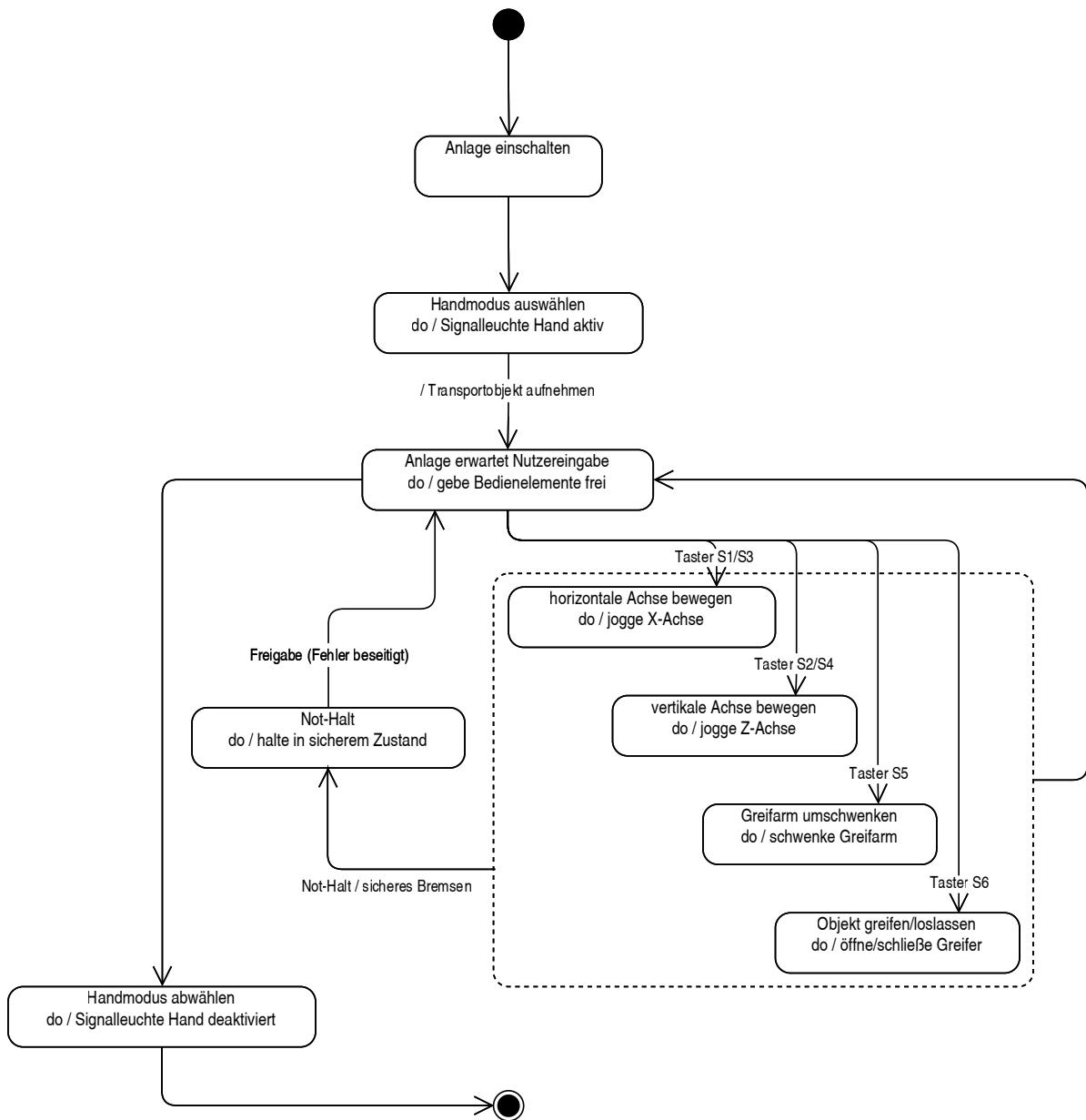


Abbildung 5: Zustandsdiagramm - Verhalten im Handmodus

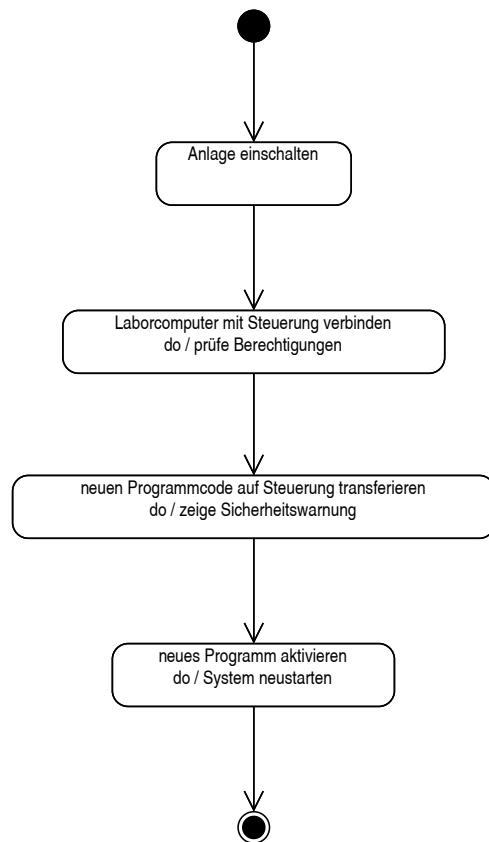


Abbildung 6: Zustandsdiagramm - Programmänderungen vornehmen

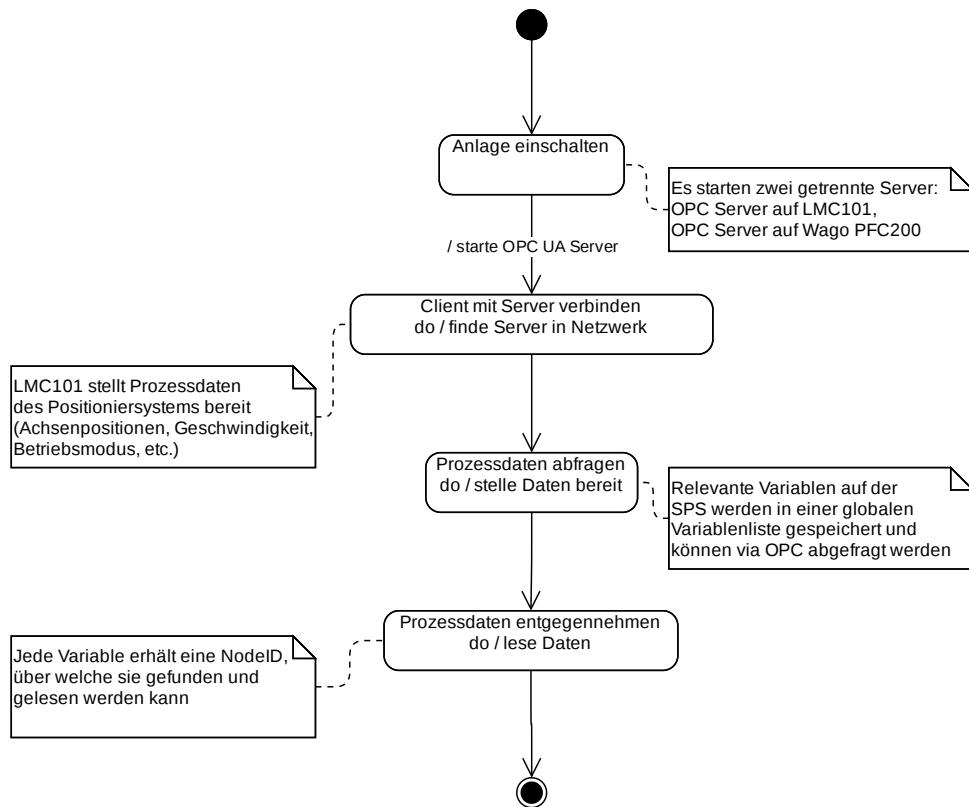


Abbildung 7: Zustandsdiagramm - Prozessdaten bereitstellen

## 4.4 Partitionierung

Die Partitionierung ist der letzte Schritt der detaillierten Systemanalyse und stellt somit auch das Ende der Analysephase dar. Es schließt sich dennoch ein weiteres Unterkapitel nachfolgend an, welches die Testspezifikationen, die im Laufe der Analysephase entstanden sind, zusammenfassend dokumentiert.

Ziel der Partitionierung ist die Unterteilung des Systems in logische Sinnesabschnitte, um die Komplexität der Darstellung und Entwicklung zu verringern. Logische Sinnesabschnitte meint an dieser Stelle jedoch nicht die logische Kontextabgrenzung aus Unterabschnitt 4.1, sondern ganz im Gegenteil, die physikalische Kontextabgrenzung, wie sie in Abbildung 2 als Verteilungsdiagramm dargestellt ist. Das Verteilungsdiagramm dient als Ausgangspunkt für die Partitionierung. Nachfolgend wird die Partitionierung in drei Sinnesabschnitte unterteilt, welche in dieser Arbeit ihren eigenen Unterabschnitt erhalten.

### 4.4.1 Erster Partitionierungsschritt

In der bisherigen Betrachtung wurde das zu entwickelnde System als Black Box dargestellt. Zu erkennen sind bereits die Nachbarsysteme und die Kommunikationspfade zu diesen. In Abbildung 2 sind schon einige Stereotypen an den Kommunikationspfaden zu erkennen. Dabei handelt es sich um konkrete Umsetzungen bzw. Realisierungen der Kommunikation. Aus den Realisierungen der Kommunikationspfade werden in der Partitionierung nun Schnittstellenknoten des Systems definiert. Diese sind im Folgenden in Abbildung 8 zu erkennen. Im bisherigen Stand der Entwicklung sind nur die Stereotypen der Kommunikationspfade spezifiziert und beschreiben die Schnittstelle zu diesem.

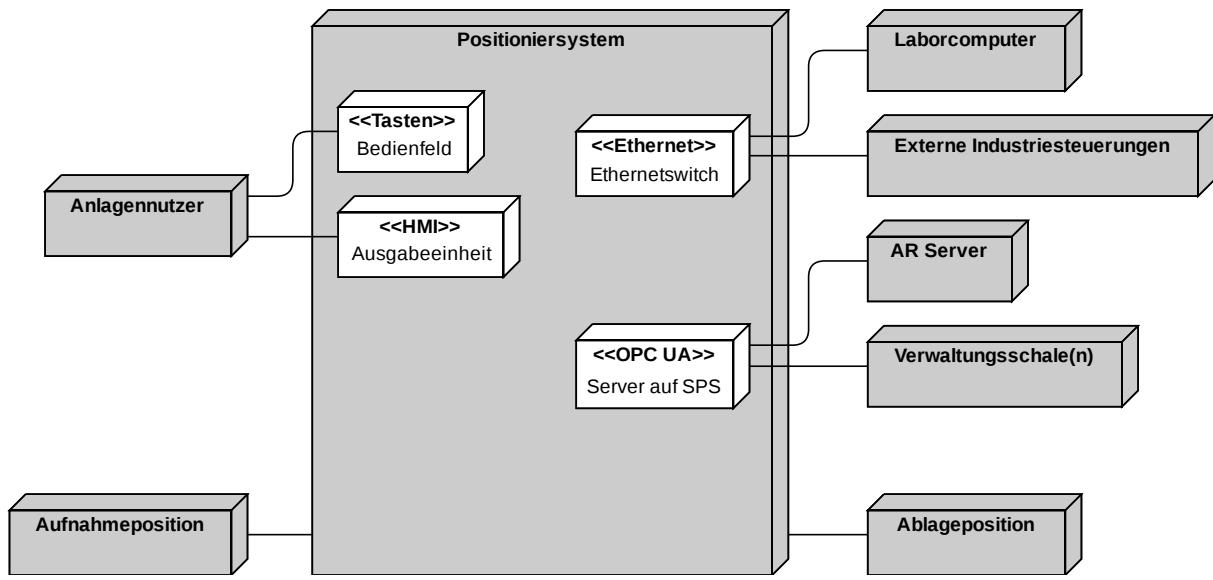


Abbildung 8: Erster Partitionierungsschritt

Abbildung 8 ergänzt zu den bereits bekannten Stereotypen, welche nun in ihren eigenen Knoten aufgeführt werden, die konkreten Knoten als Komponenten des Systems. Wichtig ist hierbei die jeweiligen Anforderungen aus Unterabschnitt 3.1 zu den Knoten zu beachten. Die Interaktion des Anlagennutzers mit dem mehrachsigen Positionersystem erfolgt grundsätzlich über Taster an der Schaltschrankfront der Anlage. Die Menge aller Taster ist im obigen Diagramm zusammengefasst unter dem Begriff Bedienfeld. Auf diesem befinden sich zusätzlich zu den Tastern auch Statusleuchten (Anzeige des ausgewählten Betriebsmodus). Weiterhin ist auch eine Signalampel am äußeren Profil der Laboranlage montiert, wie aus der Beschreibung des Aufbaus der Anlage im Unterunterabschnitt 3.3.1 hervorgeht. Sowohl die Leuchten als auch die Ampel fallen nicht unter den Stereotyp Tasten, sondern werden dem allgemeineren Begriff Human Machine Interface (HMI) zugeordnet. Es handelt sich bei ihnen um anzeigenende Elemente. Dementsprechend ist auch der Knotenbegriff Ausgabeeinheit gewählt.

Das Wort *Interface* suggeriert jedoch einen Datenaustausch in zwei Richtungen. Das HMI im Sinne einer Eingabeeinheit ist in der Industrie meist ein touchfähiges Display, das sowohl Daten anzeigen kann, als auch Befehle entgegennehmen. Im Fall des Positionersystems ist solch ein HMI in Form eines Tablets oder Smartphones implementiert, welches vom Anlagennutzer entweder selbst mitgebracht wird oder an der Anlage in einer entsprechenden Halterung befestigt ist.

Der Knoten Ethernetswitch beschreibt die Schnittstelle zu Nachbarsystemen über das Laborinterne Netzwerk. Dieser wird in der Laboranlage verbaut, um externe Computer und SPSEN mit der Steuerung der Positioniereinheit zu verbinden. Ziel ist es eine Schnittstelle zur Verfügung zu stellen, über die von den Laborcomputern das Automatisierungsprogramm

auf die Steuerung des Systems gespielt werden kann.

Der letzte zu erkennende Knoten, betitelt mit Server auf der speicherprogrammierbare Steuerung (SPS), meint den OPC UA Server, über welchen Prozessdaten von der Steuerung des Systems (LMC101) bereitgestellt werden. Diese können dann von einem OPC Client, wie z. B. dem im Diagramm zu erkennenden AR Server entgegengenommen werden, um von diesem anschließend verarbeitet bzw. genutzt zu werden.

#### 4.4.2 Zweiter Partitionierungsschritt

Im zweiten Schritt wird die genaue Realisierung der Knoten und der Aufbau des Systems geklärt. Dazu werden die Systemprozessbeschreibungen aus Unterabschnitt 4.2 benötigt. Ziel ist es das System unter funktionalen Gesichtspunkten in Komponenten bzw. Einheiten zu zerlegen. Dabei wird noch nicht festgelegt, wie die Realisierung der Einheiten mit konkreter Hardware und Software umgesetzt wird. Es findet lediglich eine Aufteilung in funktionale Komponenten statt, welche wiederum in Form von Knoten Symbolisiert werden.

Abbildung 9 zeigt das entstandene Diagramm nach Anwendung der Systemprozessbeschreibungen auf die im vorherigen Unterabschnitt entwickelte Grafik.

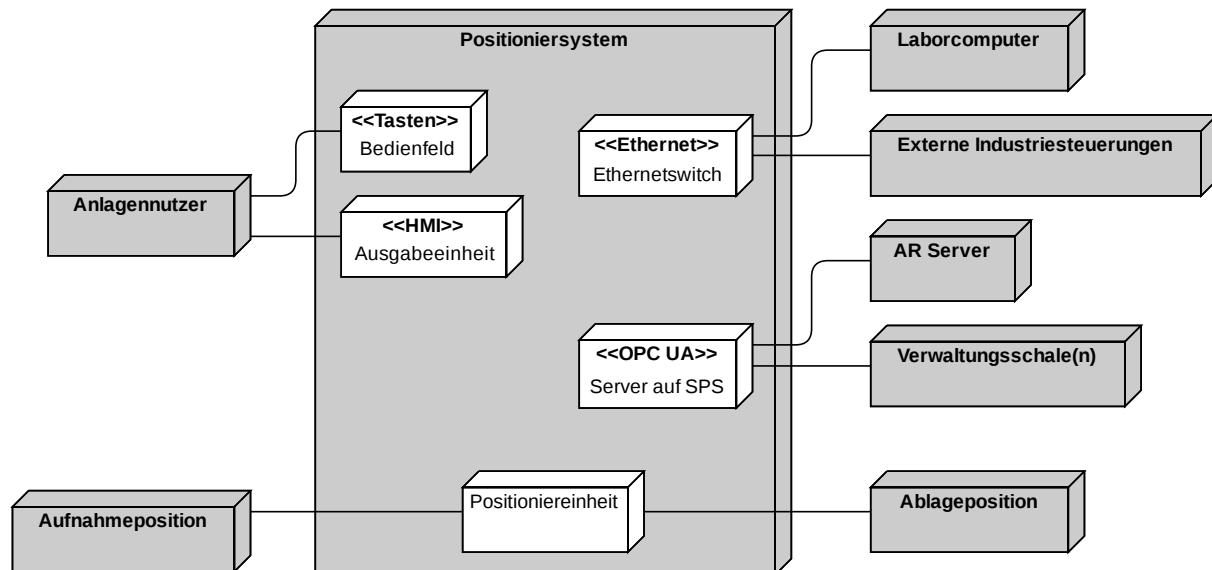


Abbildung 9: Zweiter Partitionierungsschritt

Für diesen Partitionierungsschritt sind die *essenziellen Schritte* und die *Kurzbeschreibung* aus unter anderem Tabelle 26 relevant. Für die Erstellung der kompletten Grafik müssen alle Anwendungsfallbeschreibungen berücksichtigt werden. Es fällt auf, dass im Vergleich zu Grundlegenden Anwendungsfallbeschreibung im Anwendungsfalldiagramm (Abbildung 3), die Betriebsmodi nicht mit aufgenommen wurden. Grund dafür ist, dass

bei der Partitionierung nur die normale Arbeitsweise im Vordergrund steht. Erst bei der Realisierung der in diesem Unterabschnitt gefundenen funktionalen Knoten werden diese wieder betrachtet, da sie eigenschaften dieser Knoten beschreiben.

Da zwischen der Aufnahme eines Transportobjektes von der Aufnahmeposition und der Ablage selbigen Objektes auf der Ablageposition nur der Aufnahmeprozess über einen Greifer und der Transportprozess des Positioniersystems selbst stehen, ist im Diagramm nur ein neuer Knoten wiederzufinden. Dieser ist mit dem Begriff *Positioniereinheit* betitelt. Die Positioniereinheit meint an dieser Stelle nicht das gesamte System, sondern ausschließlich die beweglichen Komponenten des Systems (die beiden Achsen mit dem darauf zu montierendem Greifarm inklusive dem Greifer selbst).

#### 4.4.3 Dritter Partitionierungsschritt

Im dritten Schritt der Partitionierung findet eine Aufteilung der Komponenten/Einheiten in Software-, Hardware und Anlagenteil statt. Dabei ist die Aufteilung der Einheit unabhängig von der aktuell betrachteten Einheit. Das heißt, soweit es möglich ist, wird die Unterteilung für jede Komponente bzw. Einheit vorgenommen. Existiert eine der drei Unterteilungen nicht für die betrachtete Einheit, entfällt diese.

Die Abbildung 10 zeigt die prinzipielle Aufteilung der einzelnen Einheiten des Systems.

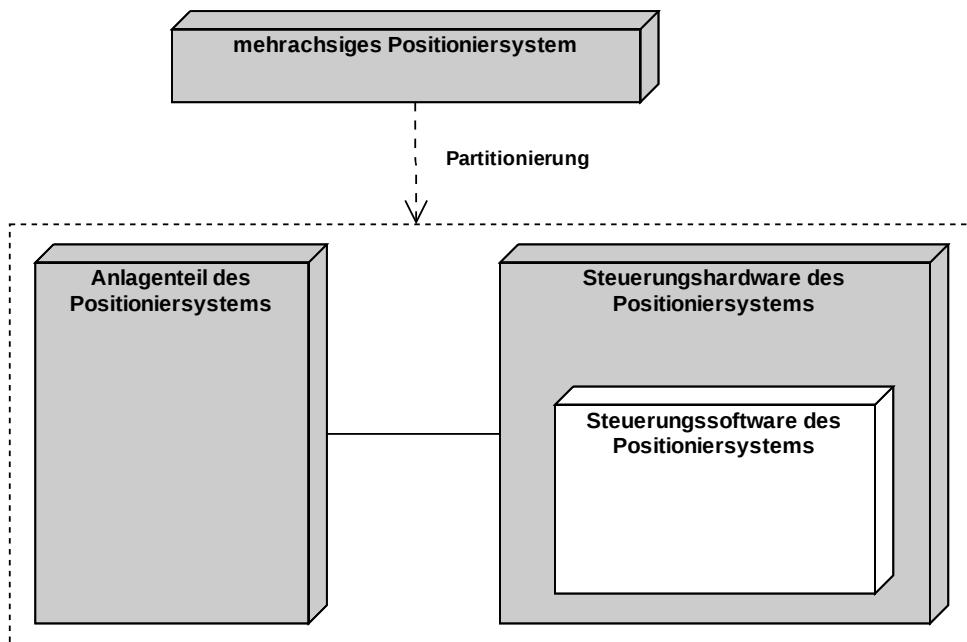


Abbildung 10: Dritter Partitionierungsschritt - Aufteilungsprinzip

Es ist zu erkennen, dass die Grafik aus drei Knoten besteht, die genau die drei Teile Anlage, Hardware und Software darstellen. Gemeinsam decken sie die Funktionalität einer

Einheit ab. Ausgehend von dieser Darstellung werden nachfolgend die Schnittstellen und prinzipiellen Eigenschaften dieser Knoten entwickelt und in einer Knotenbeschreibung dokumentiert.

In diesem Schritt treten die nicht funktionalen Anforderungen an das System wieder in den Vordergrund. Aufgabe ist es nun die Anforderungen nach Informationen an das Design des Systems zu durchleuchten. Ziel ist es zu jeder der drei erwähnten Unterteilungen spezifische Umsetzungen zu finden bzw. zu entwickeln, falls aus den NFAs keine Designentscheidungen entnehmbar sind.

Konkret das mehrachsige Positioniersystem betreffend ist aus den Anforderungen zu erkennen, dass für die Steuerungshardware der Positioniereinheit ein Logic Motion Controller (LMC) von Schneider Electric vorgesehen ist, und für die Energiemessung eine Wago SPS der PFC200 Serie. Zur Steuerungshardware zugehörig sind weiterhin sowohl das Netzteil und der Servoregler. Beide werden in den Anforderungen bereits erwähnt. Es steht fest, um welche Modelle es sich handelt (LXM62 Serie).

Auch zu den Aktuatoren und Sensoren der Steuerungshardware werden Aussagen in der Anforderungsanalyse getroffen. Für die Endlagendetektierung werden induktive Näherungsschalter verwendet. Der Lichtvorhang zum Schutz für Leib und Leben wird auch als Sensor kategorisiert und ist bereits ausgewählt (XUS Serie von Schneider Electric). Bei den Aktuatoren des Systems handelt es sich um multiturn Servos aus der Produktserie SH3 (ebenfalls von Schneider Electric).

Der generelle Aufbau des Systems bestehend aus Gehäuse und Schaltschrank wird nicht direkt in den Anforderungen vorgegeben. Somit müssen erst Anforderungen für den Anlagenteil des Systems getroffen werden. Für die Entscheidung zur Wahl der konkreten Profile, aus denen das Gehäuse aufgebaut wird, sowie die beweglichen Schlitten auf den beiden Achsen und der Schaltschrank, sollten Fachleute herangezogen werden. Im Falle der Laboranlage wurden Designentscheidungen aus den nicht funktionalen Anforderungen des Laboringenieurs Dipl. Ing. Dirk Schöttke (wiederzufinden in der Stakeholdertabelle) getroffen. Es handelt sich dabei hauptsächlich aus Erfahrungen mit bereits umgesetzten Systemen. Resultat ist die Wahl von Profilen und beweglichen Schlitten der Firma Mini-Tec. Auch die Entscheidung, welcher Schaltschrank ausgewählt wird, wird Anhand der definierten NFAs von Herr Schöttke getroffen. Zu beachten ist, dass der Schaltschrank genügend Platz für alle benötigten Komponenten bereitstellt.

Noch zu untersuchen sind die Anforderungen nach Designinformationen zur Steuerungssoftware des Positioniersystems. Aus den nicht funktionalen Anforderungen an die Steuerungshardware erübriggt sich die Wahl der Steuerungssoftware des Positioniersystems. Durch die Nutzung des Logic Motion Controllers von Schneider Electric ist die Programmierung dieses eingeschränkt auf die Entwicklungsumgebung, die ebenfalls durch Schneider Electric bereitgestellt wird. Die später auf der Laboranlage ausgeführte Automatisierungssoftware muss in der SoMachine bzw. MachineExpert Entwicklungsumgebung generiert werden, um diese auf der Steuerung des Systems nutzen zu können. Es handelt sich folglich um ein

abgeschlossenes Ökosystem, welches vom Hersteller etabliert wurde. Resultat ist somit auch eine vorgeschrriebene Vorgehensweise zur Umsetzung von Motion Software für das Positioniersystem. Die Steuerungssoftware der PFC200 SPS soll an dieser Stelle nicht unerwähnt bleiben. Die Konfiguration der Steuerung bedingt die Nutzung der Software e!Cockpit von Wago zur Programmierung der Steuerungssoftware für die Energiemessungskomponente des Systems.

Nachdem alle Partitionierungsentscheidungen bezüglich der Anlage getroffen wurden, werden diese in Form einer Tabelle dokumentiert. Dabei handelt es sich um die Beschreibung der Knoten aus dem Verteilungsdiagramm in Abbildung 10.

Die erste Tabelle (Tabelle 30) behandelt den Anlagenteil der Positioniereinheit. Die Felder der Tabelle enthalten alle wichtigen Informationen über den Knoten des Anlagenteils der Positioniereinheit. Zu dokumentieren sind die Informationen **Name**, **Typ**, **Beschreibung**, **FAs** und **NFAs**. Im Eintrag *Typ* wird die Knotenart definiert, also ob es sich um Software, Hardware oder die Anlage handelt. Die funktionalen und nicht funktionalen Anforderungen, die sich auf den Knoten beziehen, werden ebenfalls in der Tabelle aufgenommen.

<b>Name</b>	Anlagenteil des Positioniereinheit
<b>Typ</b>	Anlage
<b>Beschreibung</b>	Stellt die Infrastruktur bereit, an die die Aktuatoren und Sensoren der Steuerungshardware angeschlossen sind.
<b>Funktionale Anforderungen</b>	Die Anlage muss sich entlang der horizontalen und vertikalen Achse bewegen können und wieder abbremsen. Weiterhin muss sie über einen Greifarm/Greifer Kombination Transportgüter aufnehmen und wieder ablegen können.
<b>Nicht funktionale Anforderungen</b>	Die Anlage muss in den Laborraum G422 integriert werden. Zur Sicherheit sollen Plexiglasabdeckungen an ungeschützten Bereichen montiert werden. Kabel sollen in E-Ketten an den beweglichen Achsen mitgeführt werden.

Tabelle 30: Knotenpunktbeschreibung der Positioniereinheit - Anlagenteil

Analog zur ersten Tabelle wird anschließend die Knotenbeschreibung zur Hardware und Software der Positioniereinheit vorgenommen.

<b>Name</b>	Steuerungshardware des Positioniereinheit
<b>Typ</b>	Hardware
<b>Beschreibung</b>	Die Hardware besteht aus den Controllern (LMC und PFC200), Aktuatoren und Sensoren. Diese sind zur Erfüllung der Aufgaben der Positioniereinheit nötig.
<b>Funktionale Anforderungen</b>	Die Servomotoren der Achsen sollen über den LMC gesteuert Positionieraufgaben ausführen. Dafür wird eine Powersupply/Servoregler Kombination benötigt, um die Motoren zu betreiben. Über Endlagesensoren soll verhindert werden, dass Bewegungen über die Enden der beiden Achsen hinaus durchgeführt werden können. Mit Hilfe eines zweiten Controllers soll die Energiezufuhr zum gesamten System gemessen werden. Nutzereingaben sollen über ein Bedienfeld an der Schaltschrankfront und/oder über ein Display erfolgen. Zur Signalisierung von Gefahrensituationen ist eine Signalampel zu verbauen.
<b>Nicht funktionale Anforderungen</b>	Die Steuerungshardware der Positioniereinheit stammt aus der PacDrive3 Reihe von Schneider Electric. Erweitert wird diese um zwei Motoren und einen Lichtvorhang, die ebenfalls von Schneider Electric gestellt werden. Um weitere digitale und Anlago (sichere) Ausgänge bereitzustellen, sollen Module aus der TM5 Serie von Schneider Electric im Schaltschrank verbaut werden. Die SPS für die Energiemessung ist ein PFC200 Controller von Wago.

Tabelle 31: Knotenpunktbeschreibung der Positioniereinheit - Hardware

Durch die Vorlage der Steuerungssoftware stehen die Schnittstellen zwischen den Steuerkomponenten und den Sensoren sowie Aktuatoren bereits fest. Der LMC400 kommuniziert mit den TM5 Erweiterungsmodulen, dem LXM62 P (Powersupply) und LXM62 D (Servoregler) via SercosIII (in Ringkonfiguration). Zu den Endlagesensoren führen dreiaadrige Sensorleitungen. Der Lichtvorhang (bestehend aus Emitter und Receiver) ist ebenfalls über 8 und 11 adrige Sensorleitungen am Controller (über Klemmen im Schaltschrank) angeschlossen. Zu den beiden Motoren führt sowohl ein Stromkabel (3-phasic) und eine Encoderkabel (für die Bremsfunktion und das Auslesen von z. B. Temperaturwer-

ten). Die beiden Controller sind untereinander und auch mit allen weiteren Computern im Labornetzwerk per Ethernet über einen im Schaltschrank verbauten Switch verbunden.

<b>Name</b>	Steuerungssoftware der Positioniereinheit
<b>Typ</b>	Software
<b>Beschreibung</b>	Die Steuerungssoftware wertet Sensordaten aus, berechnet Trajektorien und Geschwindigkeiten für die Achsenpositionierung und stellt Prozessdaten aus globalen Variablenlisten bereit.
<b>Funktionale Anforderungen</b>	Berechnet Fahrtwege, Geschwindigkeiten und Beschleunigungen aus Nutzereingaben bzw. Vorgaben. Weiterhin wird für das sicherheitsgerechte Bremsen bei Erreichen von Endlagen oder der Not-Halt-Auslösung gesorgt.
<b>Nicht funktionale Anforderungen</b>	Die Automatisierungssoftware muss in der LogicBuilder Entwicklungsumgebung erstellt werden. Diese Umgebung wird bereits mit einigen Entwicklungsvorgehensweisen bzw. Routinen ausgeliefert.

Tabelle 32: Knotenpunktbeschreibung der Positioniereinheit - Software

Da das Positioniersystem nur aus einer Einheit, der Positioniereinheit besteht, kann der dritte Partitionierungsschritt mit drei Tabellen (Anlage, Software, Hardware) modelliert werden. Die weniger hohe Komplexität (bedingt durch das Nichtvorhandensein von mehreren Untereinheiten) führt zu der Partitionierung in die Grundlegenden Einheiten *Anlagenteil*, *Steuerungshardware* und *Steuerungssoftware* der Positioniereinheit.

Auf Basis der Partitionierung kann in der Anlagenprojektierung nun die Entwicklung der Systemsoftware durchgeführt werden, nachdem die detaillierte Systemanalyse mit diesem Unterabschnitt beendet ist. Wie bereits eingangs erwähnt, folgt zunächst die Dokumentation der Testspezifikation, welche nach der Projektierung für die Inbetriebnahme der in der Projektierung entwickelten Software benötigt wird.

## 4.5 Testspezifikation

Dieses letzte Unterkapitel der Konzeptionsphase der Arbeit behandelt die Entstehung und den Aufbau der Testspezifikation. Die Testspezifikation ist an sich kein eigener Schritt in der Analysephase, sondern entwickelt sich über die verschiedenen Schritte der Analyse hinweg.

Testkriterien werden bereits in den Anforderungen aufgestellt und dienen als Abnahmekriterium für diese. Neben den funktionalen und nicht funktionalen Anforderungen, entsteht die Testspezifikation aus sowohl der Anwendungsfallspezifikation, der Verhaltensspezifikation und den Partitionierungsinformationen.

Die Dokumentation der Testinformationen erfolgt in Tabellenform. Die Tabelle unterteilt sich in die Einträge **Name**, **Typ**, **Beschreibung**, **Kriterium**, **Spezialfälle** und **Stakeholder**. Im Feld *Typ* wird der Geltungsbereich des Tests festgehalten. Mögliche Werte sind hier *System*, *Anlage*, *Hardware*, *Software*. Auch das Tabellenfeld *Spezialfälle* bedarf einer gesonderten Erklärung. Es handelt sich um besonders kritische Testfälle eine Anforderung betreffend. Das könnten z. B. Testfälle an der Toleranzgrenze sein.

Die Testspezifikation schließt an die Testkriterien aus den bereits erwähnten Unterkapiteln der Analysephase an und hat zum Ziel diese zu konkretisieren und gesammelt darzustellen. Nachfolgend finden sich die essenziellen Testkriterien des mehrachsigen Positioniersystems. Für eine vollständige Auflistung wird auch an dieser Stelle auf den Anhang verwiesen.

<b>Name</b>	Test der Betriebsmodusauswahl
<b>Typ</b>	System
<b>Beschreibung</b>	Über die Signalisierung mittels Hardware und Softwarevisualisierungen, sowie der erfolgreichen Nutzung von Betriebsmodi spezifischen Funktionen kann die Bereitstellung der Auswahlmöglichkeit zwischen den beiden Betriebsmodi geprüft werden.
<b>Kriterium</b>	Ist ein Betriebsmodus ausgewählt worden, leuchtet die entsprechende Lampe mit der Aufschrift „Hand“ bzw. „Auto“ an der Schaltschrankfront auf (Betriebsmodus wird später auch auf einem Display als Text angezeigt). Die Betriebsmoduspezifischen Funktionen können im Anschluss genutzt werden.
<b>Spezialfälle</b>	Befindet sich die Anlage im Not-Halt, so kann keiner der Betriebsmodi genutzt werden.
<b>Stakeholder</b>	Prozessentwickler siehe Stakeholderliste.

Tabelle 33: Testkriterium - Auswahl des Betriebsmodus

<b>Name</b>	Test der Positionierfähigkeit(en)
<b>Typ</b>	Anlage
<b>Beschreibung</b>	Sowohl Programmatisch als auch über Tastereingaben kann verifiziert werden, dass die beiden Achsen der Positioniereinheit bewegungen durchführen können.
<b>Kriterium</b>	Der Prozessentwickler kann über das Automatisierungsprogramm Trajektorien den Fahrweg betreffend angeben. Diese sollten bei der Ausführung der Positionieraufgabe sichtbar sein und fehlerfrei durchgeführt werden. Auch die Nutzerinteraktion mit dem Vierwhegeschalter führt zu Bewegungen der beiden Achsen.
<b>Spezialfälle</b>	Befindet sich die Anlage im Not-Halt, muss diese zunächst freigegeben werden, sodass die Laboranlage wieder nutzbar ist.
<b>Stakeholder</b>	Prozessentwickler siehe Stakeholderliste.

Tabelle 34: Testkriterium - Positionieren der Achsen

<b>Name</b>	Test der Programmierschnittstelle
<b>Typ</b>	Hardware
<b>Beschreibung</b>	Durch Prüfung der Verbindung von Laborcomputern zum Systemcontroller soll die Programmierschnittstelle und die Verbindung aus dem Netzwerk zum Controller verifiziert werden.
<b>Kriterium</b>	Der LMC400 kann im LogicBuilder über einen PC im selben Netzwerk (z. B. ein Laborcomputer) gefunden und ausgewählt werden. Es besteht anschließend die Möglichkeit Programme auf die Steuerung zu transferieren und diese zu testen.
<b>Spezialfälle</b>	Zugriff von einem unbekannten Drittorechner.
<b>Stakeholder</b>	Lehrpersonal siehe Stakeholderliste.

Tabelle 35: Testkriterium - Programmierschnittstelle des Systems

<b>Name</b>	Test der Prozessdatenbereitstellung
<b>Typ</b>	Software
<b>Beschreibung</b>	Über einen OPC Client können Prozessdaten von den beiden Controllern des Systems empfangen werden.
<b>Kriterium</b>	Über einen Laborcomputer im selben Netzwerk kann eine Verbindung via OPC UA hergestellt werden. Dies wird verifiziert über das Programm „OPC Watch“. Dort kann die Adresse des/der OPC UA Server(s) eingegeben und dessen/deren Daten ausgelesen werden.
<b>Spezialfälle</b>	Verbindung von OPC UA Clients zur Weiterverarbeitung des Datensatzes (AR Server, Verwaltungsschale(n))
<b>Stakeholder</b>	Prozessentwickler siehe Stakeholderliste.

Tabelle 36: Testkriterium - Bereitstellung von Prozessdaten via OPC

Die während der Analyse erstellte Testspezifikation stellt die Grundlage für spätere Tests in der System Integrations- und Testphase dar, welche sich im letzten Kapitel zur Inbetriebnahme des Systems wiederfindet.

## 4.6 Stromlaufplan

## 4.7 Datenmodell

Nr.	Betriebsmittel	BMK	Allgemeine Beschreibung	Datentyp	SPS-Adresse	Logischer Zustand
1	Endlage Oben (Öffner)	g_ixIN_00	Erfassung oberes Ende der z-Achse erreicht	BOOL	%IX230.0	betätigt=0
2	Endlage Unten (Öffner)	g_ixIN_01	Erfassung unteres Ende der z-Achse erreicht	BOOL	%IX230.1	betätigt=0
3	Endlage Links (Öffner)	g_ixIN_02	Erfassung linkes Ende der x-Achse erreicht	BOOL	%IX230.2	betätigt=0
4	Endlage Rechts (Öffner)	g_ixIN_03	Erfassung rechtes Ende der x-Achse erreicht	BOOL	%IX230.3	betätigt=0
5	TA Rtg. Oben (Schließer)	g_ixIN_04	Eingabe Jog+ z-Achse	BOOL	%IX230.4	betätigt=1
6	TA Rtg. Unten (Schließer)	g_ixIN_05	Eingabe Jog- z-Achse	BOOL	%IX230.5	betätigt=1
7	TA Rtg. Links (Schließer)	g_ixIN_06	Eingabe Jog+ x-Achse	BOOL	%IX230.6	betätigt=1
8	TA Rtg. Rechts (Schließer)	g_ixIN_07	Eingabe Jog- x-Achse	BOOL	%IX230.7	betätigt=1
9	TA Grün (Schließer)	g_ixIN_08	EIN / Quittieren eines Fehlers und Freigabe	BOOL	%IX231.0	betätigt=1
10	TA Rot (Öffner)	g_ixIN_09	AUS	BOOL	%IX231.1	betätigt=0
11	TA Weiß Oben (Schließer)	g_ixIN_10	Frei / vom Anwender wählbar	BOOL	%IX231.2	betätigt=1

12	TA Weiß Unten (Schließer)	g_ixIN_12	Frei / vom Anwender wählbar	BOOL	%IX231.3	betätigt=1
13	Schalter Rtg. Links (Schließer)	g_ixIN_13	Auswahl Handmodus	BOOL	%IX231.4	betätigt=1
14	Schalter Rtg. Rechts (Schließer)	g_ixIN_14	Auswahl Automatikmodus	BOOL	%IX231.5	betätigt=1
15	Lichtvorhang OSSD1 (Öffner)	g_ixIN_15	Erfassung Eindringen in Arbeitsbereich	BOOL	%IX231.6	betätigt=0
16	Lichtvorhang OSSD2 (Öffner)	g_ixIN_16	Erfassung Eindringen in Arbeitsbereich	BOOL	%IX231.7	betätigt=0

Tabelle 37: Datenmodell - Digitale Eingänge Modicon TM5 SDI16D E/A Modul

Nr.	Betriebsmittel	BMK	Allgemeine Beschreibung	Datentyp	SPS-Adresse	Logischer Zustand
1	Signalsäule Rot	g_qxOUT_00	Dauerleuchten: Anlage befindet sich im Fehlerzustand	BOOL	%QX224.0	betätigt=1
2	Signalsäule Grün	g_qxOUT_01	Dauerleuchten: Anlage ist betriebsbereit (Leerlauf)	BOOL	%QX224.1	betätigt=1
3	TA LED Grün	g_qxOUT_02	Anlage ist eingeschaltet	BOOL	%QX224.2	betätigt=1
4	TA LED Rot	g_qxOUT_03	Anlage ist Ausgeschaltet (Hauptschalter jedoch EIN)	BOOL	%QX224.3	betätigt=1
5	TA LED Weiß Oben	g_qxOUT_04	Frei / vom Anwender definierbar	BOOL	%QX224.4	betätigt=1

6	TA LED Weiß Unten	g_qxOUT_05	Frei / vom Anwender definierbar	BOOL	%QX224.5	betätigt=1
7-16	NA	g_qxOUT_06-15	—	BOOL	%QX224.6-225.7	—

Tabelle 38: Datenmodell - Digitale Ausgänge Modicon TM5 SDO16T E/A Modul

Nr.	Betriebsmittel	BMK	Allgemeine Beschreibung	Datentyp	SPS-Adresse	Wertebereich
1	Poti Oben	g_irIN_00	Vorgabe Maximalgeschwindigkeit z-Achse	REAL	%IX237.0	0.0 - 1.0
2	Poti Unten	g_irIN_01	Vorgabe Maximalgeschwindigkeit x-Achse	REAL	%IX237.1	0.0 - 1.0
3	NA	g_irIN_02	—	—	%IX237.2	—
4	NA	g_irIN_03	—	—	%IX237.3	—

Tabelle 39: Datenmodell - Analoge Eingänge Modicon TM5 SAI4L E/A Modul

Nr.	Betriebsmittel	BMK	Allgemeine Beschreibung	Datentyp	SPS-Adresse	Wertebereich
1-4	NA	g_qrOUT_00-04	—	—	%QW113-116	—

Tabelle 40: Datenmodell - Analoge Ausgänge Modicon TM5 SAO4L E/A Modul

Nr.	Betriebsmittel	BMK	Allgemeine Beschreibung	Daten-typ	SPS-Adresse	Logischer Zustand
1	NOT-Halt (Öffner)	g_ixSAFE_IN_00	Auslösen NOT-Halt der Achsbewegungen	BOOL	%IX250.0	betätigt=0

2	NOT-Halt (Schließer)	g_ixSAFE_IN_01	Auslösen NOT-Halt der Achsbewegungen	BOOL	%IX250.1	betätigt=1
3	Lichtvorhang OSSD1 (Schließer)	g_ixSAFE_IN_02	Auslösen NOT-Halt der Achsbewegungen	BOOL	%IX250.2	betätigt=1
4	Lichtvorhang OSSD2 (Schließer)	g_ixSAFE_IN_03	Auslösen NOT-Halt der Achsbewegungen	BOOL	%IX250.4	betätigt=1

Tabelle 41: Datenmodell - Sichere digitale Eingänge Modicon TM5 SDI4DFS E/A Modul

Nr.	Betriebsmittel	BMK	Allgemeine Beschreibung	Datentyp	SPS-Adresse	Logischer Zustand
1	Inverter Enable	g_qxSAFE_OUT_00	Signalabbruch führt zum Abschalten des Servoreglers (LXM62 D)	BOOL	%QX234.0	betätigt=1
2	Inverter Enable	g_qxSAFE_OUT_01	Signalabbruch führt zum Abschalten des Servoreglers (LXM62 D)	BOOL	%QX234.1	betätigt=1
3	NA	g_qxSAFE_OUT_02	—	BOOL	%QX234.2	—
4	NA	g_qxSAFE_OUT_03	—	BOOL	%QX234.3	—

Tabelle 42: Datenmodell - Sichere digitale Ausgänge Modicon TM5 SDO4TFS E/A Modul

Nr.	Funktion	BMK	Allgemeine Beschreibung	Datentyp	SPS-Adresse	Anfangs-/Standardwert
1	Error Output	bError_Out	Error Meldung SF_Antivalent SoSafe Programmable	BOOL	%IX0.0	FALSE

2	Hardware Status	bHardwareOk	Hardwarestatus Modicon TM5 SLC	BOOL	%IX0.1	FALSE
3	NOT-Halt (intern)	bNotHalt	Anbindung NOT-Halt an Programmtemplate	BOOL	%IX0.2	FALSE
4	Lichtvorhang OSSD1 (intern)	bNotHalt	Anbindung Lichtvorhangsstatus an Programmtemplate	BOOL	%IX0.3	FALSE
5	Lichtvorhang OSSD2 (intern)	bNotHalt	Anbindung Lichtvorhangsstatus an Programmtemplate	BOOL	%IX0.4	FALSE

Tabelle 43: Datenmodell - Businterne Variablen: geschrieben durch SLC, gesendet an LMC (SLC2LMC in Modicon TM5 CSLC Safety Steuerung)

## 4.8 Bedienkonzept

---

## 5 Inbetriebnahme

Dieses Kapitel unterteilt sich in drei Abschnitte. Das erste Unterkapitel behandelt die Implementation bzw. die Implementationsphase der Realisierung des mehrachsigen Positioniersystems. Dabei geht es auf der einen Seite um die Beschreibung des Hardware- bzw. Geräteentwurfs aus den zugehörigen Inhalten im Kapitel zur Projektierung der Anlage. Konkret meint dies den konstruktionellen Aufbau der Laboranlage aus der Konfigurator Grafik und der Konstruktionszeichnung an erster Stelle und nachfolgend die Verdrahtung der Komponenten nach dem entwickelten Stromlaufplan. Auf der anderen Seite beinhaltet das sich anschließende Kapitel die Implementation der Software aus den in der Modellierung erstellten UML Diagrammen wie unter anderem das Zustandsdiagramm.

Nach der Implementation kann und muss das System auf seine Funktionalität geprüft werden, sowohl was seine Hardware als auch seine Software anbelangt. Dazu werden die Testkriterien aus der Anforderungsanalyse herangezogen und in Testfällen zur Überprüfung aller Anforderungen genutzt. Dabei wird eine Unterteilung in mehrere Ebenen vorgenommen, die aufbauend nacheinander durchgangen werden. Ziel ist es zunächst grundlegende Testfälle zu behandeln, die Voraussetzung für speziellere, untergeordnete Validierungen von Testspezifikationen sind.

Wird ein Test nicht bestanden, muss in einer weiteren Iteration der Entwicklungsphase der bzw. die Mängel beseitigt werden. Diese Korrekturen werden im letzten Unterkapitel der Inbetriebnahme behandelt.

### 5.1 Implementationsphase

In diesem Unterkapitel findet sich die Dokumentation der Implementation sämtlicher Modelle aus der Projektierungsphase wieder. Zunächst wird im Unterunterabschnitt 5.1.1 die Montage der Laboranlage kurz dargestellt. Dazu wird aus der Konfiguratorskizze unter Berücksichtigung der Maße des Laborraumes und der entsprechenden Anforderungen an den Aufbau des Positioniersystems die Anlage aufgebaut. Nachdem alle Gehäuseelemente an der Wand und dem Boden verankert sind können die Steuerungskomponenten, die Aktuatoren und die Sensoren an diesen befestigt werden. Weiterhin umfasst der nachfolgende Abschnitt die Verdrahtung der elektrischen Komponenten nach entwickeltem Stromlaufplan und der Netzwerkdarstellung aus dem Unterabschnitt 4.6.

Im zweiten Unterabschnitt (Unterunterabschnitt 5.1.2) wird die Implementation der modellierten Diagramme zur Steuerungssoftware vorgenommen. Da durch die Wahl der Steuerung (Logic Motion Controller von Schneider Electric), wie bereits in der Anforderungsanalyse festgestellt, die Umgebung zur Programmierung der Systemsoftware festgelegt ist, besteht die Möglichkeit Templates aus dieser zu nutzen, welche den Softwareentwicklungsprozess vereinfachen. Der Unterabschnitt zur Software-Implementation behandelt somit die schrittweise Darstellung der Umsetzung der Automatisierungssoftware aus dem von Schneider Electric bereitgestellten *Motion Template Full*.

### 5.1.1 Hardware-Implementation

Für die Umsetzung der Hardware wird die Kofiguratorgrafik aus der Konzeptphase wird aufgegriffen und gilt als Grundlage für die reale Umsetzung der Hardwarebereiche des Systems. Da im Konfigurator bereits die Kernanforderungen an die Systemhardware berücksichtigt wurden, müssen beim Bau und der Montage des Gehäuses bzw. des Anlagengerüsts nur noch die nicht-funktionalen Anforderungen an dieses und den umliegenden Raum berücksichtigt werden.

Durch die Form und die Ausmaße des Laborraumes ergibt sich eine maximale Höhe des Gehäuses von 2230mm. Die horizontale Ausdehnung der Anlage wird nicht durch den Raum begrenzt. Die Entscheidung wurde auf Grund von Subjektiven Anschaulichkeitskriterien getroffen. Resultat ist eine horizontale Ausdehnung der x-Achse von 2000mm, was ungefähr der Gangbreite im Laborraum entspricht. Folglich ist der bewegliche Teil des Positioniersystems mittig zum Durchgang im Raum ausgerichtet. Sowohl rechts als auch links neben der Positioniereinheit ist ein Bereich von jeweils 600mm reserviert, in dem Ablagepositionen an der Wand befestigt werden können. Die rechte Seite der Anlagenkonstruktion besitzt zusätzlich noch ein weiteres Aluminium Profil, welches später benötigt wird, um den Schaltschrank und die Steuerungshardware am System zu fixieren. Nachfolgende Grafik zeigt das an der Laborraumwand montierte Anlagengerüst, an welches im nächsten Schritt die Steuerungshardware befestigt wird.



Abbildung 11: Anlagengerüst des Gehäuses vom mehrachsigen Positioniersystem an der hinteren Laborwand im Raum G 422

Im nächsten Schritt der Hardware-Implementation werden die Steuerung (LMC400) das Netzgerät (LXM 62P) und der Servoregler (LXM 62D) an Querverstrebungen der beiden rechten Profile verschraubt. Es gilt die Montageanleitung zu beachten.

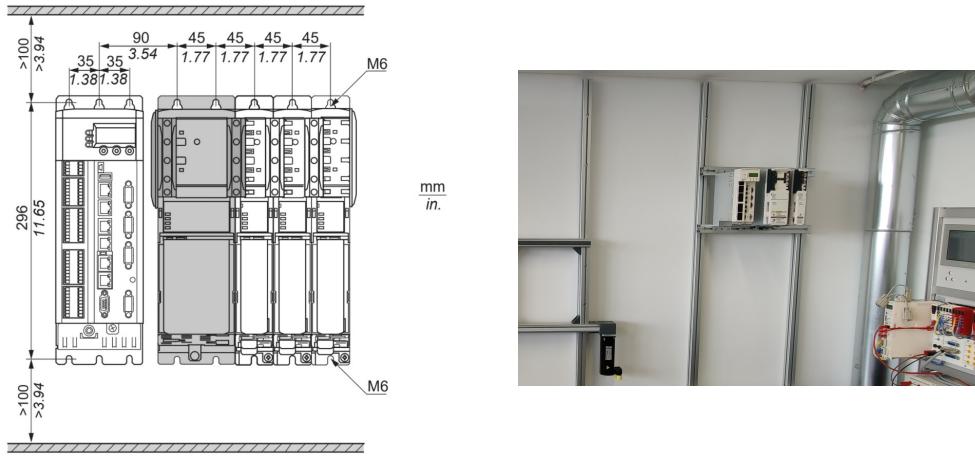


Abbildung 12: Installation der Steuerungshardware am Gehäuseaufbau des Positioniersystems

Bevor die in der Konfiguratorskizze aufgeführten Sensoren und Aktuatoren verbaut werden können, sind noch weitere Profile notwendig, um den Gehäusebau zu komplettieren. 500mm von der Wand entfernt auf Höhe der äußeren Profile des Positionierbereiches werden zwei senkrechte Aluminiumstempel an Decke und Boden befestigt. Diese sind das vordere Ende des mehrachsigen Positioniersystems. Das Gehäusegerüst ist nun vollständig und umschließt den Bereich, in dem Positionieraufgaben durchgeführt werden können mit der Laboranlage.

Zwischen den beiden linken Profilen und den beiden rechten Profilen werden Plexiglasscheiben angebracht, die das Hineingreifen in den Fahrbereich der Anlage von den Seiten verhindern sollen. An der Front wird ein Lichtvorhang bestehend aus Emitter und Receiver montiert. Dieser befindet sich auf der Innenseite an den zuletzt angebauten senkrechten Stempeln. Der Lichtvorhang dient ebenso wie die Plexiglasscheiben zum Schutz von Leib und Leben. Im Gegensatz zu den Scheiben erlaubt der Lichtvorhang jedoch im unbewegten Zustand des Systems das Eindringen von Personen in den Arbeitsbereich.

Folgende Grafik zeigt zu den soeben genannten Komponenten zusätzlich noch die in den Anforderungsanalyse ermittelten vier Endlagesensoren, die Servomotoren für x- und z-Achse, sowie E-Ketten zu den beweglichen Achsen.



Abbildung 13: Installation von Sensoren und Aktuatoren des mehrachsigen Positioniersystems

Der Schaltschrankbau und die Verdrahtung des Systems stellt den letzten Schritt in der Hardware-Implementierung dar. Als Schaltschrank wurde ein Modell der Firma Rittal gewählt. Die Breite des Schrankes ist durch den gewählten Aufbau bereits festgelegt und beträgt 600mm. Aufgrund der Anzahl der Klemmen und dem Wunsch E/A-Module sowie weitere Steuerungskomponenten physisch von normalen Klemmen zu trennen, wurde entschieden eine Schaltschrankkonfiguration zu wählen, die in der Höhe zwei Hutschienen unterbringt. Die gewählte Schrankhöhe liegt deswegen bei 380mm. Zuletzt muss die Tiefe des Schaltschranks ausreichend sein, um die tiefste Komponente, die im Schaltschrank verbaut werden soll, unterbringen zu können. Da bereits die kleinste verfügbare Variante des gewählten Schrankes diese Anforderung erfüllt, hat der zu verbauende Schaltschrank eine Tiefe von 210mm.

Wie bereits angedeutet ist in der Grafik unten zu erkennen, dass die untere der beiden Hutschienen sämtliche Klemmen beherbergt inklusive der Absicherungen für die jeweilige Spannungsebene, so wie im Stromlaufplan geplant. An der Oberen Hutschiene befinden sich die Modicon TM5 E/A-Module, die als Erweiterung für die Ein- und Ausgänge des LMC dienen. Die im Bild als rot gefärbte Komponente zu erkennende Steuerung, ist der Safety Logic Controller, der für die Berechnung der Sicherheitsfunktionen des Systems verantwortlich ist. Dazu besitzt dieser jeweils vier digitale Ein- und Ausgänge.

Rechts daneben auf der selben Schiene ist die Wago PFC 200 Steuerung angebracht, welche mit Hilfe ihrer Energieklemme die Leistungsaufnahme des Systems messen soll.

Ganz Rechts auf der Hutschiene ist ein Ethernetswitch montiert, an welchen beide Steuerungen (LMC400 und Wago PFC 200) per Ethernetkabel angeschlossen sind. Durch den Einsatz des Switches führt nach Vertigstellung der Verdrahtung nur ein Kabel zur Programmierung der beiden Steuerungen aus dem Schaltschrank heraus.

Weiterhin ist auf der rechten Schrankwand der Hauptschalter platziert. Über dieser aktiviert oder deaktiviert die Stromversorgung des Schaltschranks und somit auch aller Systemkomponenten.

Die Verdrahtung erfolgt nach Stromlaufplan. Dieser beinhaltet auch die Kopplung der einzelnen Module aus der PacDrive3 Serie, die verbaut wurden (LMC400, LXM 62P, LXM 62D, Modicon TM5 Module, Modicon TM5 SLC100 Sicherheitsmodule).

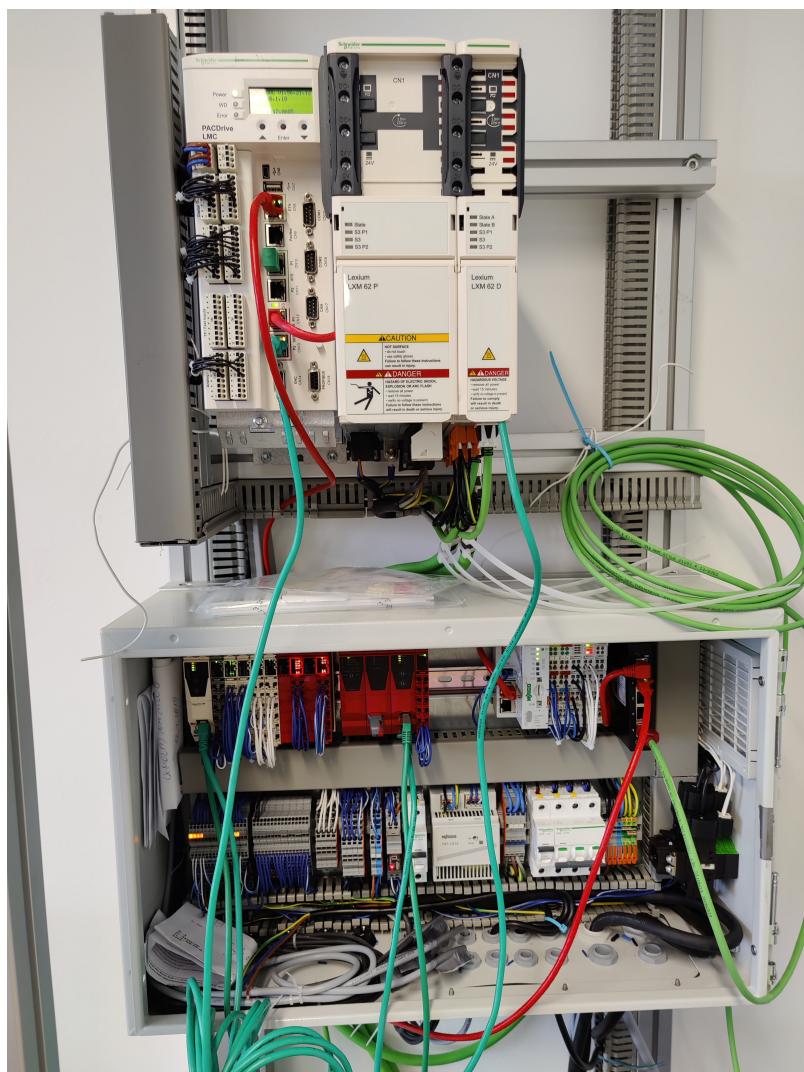


Abbildung 14: Einbau und Verdrahtung des Schaltschranks

### 5.1.2 Software-Implementation

In diesem Unterabschnitt wird die Realisierung der Automatisierungssoftware dokumentiert. Grundlegend soll aus der Modellierung der Software im vorhergegangenen Kapitel (Projektierung) das Programm für das mehrachsige Positioniersystem geschrieben werden. Wie bereits in der Einleitung zu diesem Unterkapitel erwähnt, resultiert die Wahl der Steuerungskomponenten, also konkret die Wahl Komponenten aus der PacDrive3 Serie von Schneider Electric auszuwählen, in dem Zwang mit der auf Codesys 3.5 basierten Entwicklungsumgebung Machine Expert zu arbeiten. Diese besitzt bereits mehrere Templateprojekte für verschiedene Anwendungsfälle. Die Firma Schneider Electric empfiehlt die Nutzung des jeweiligen Programmtemplates für den gewünschten Anwendungsfall. Grund dafür ist die Minimierung vom Programmieraufwand. Ziel soll es sein lediglich Konfigurationen an einem Modularen Template vorzunehmen, so dass dieses die eigenen Anforderungen erfüllt.

Begründet durch diese Aussagen wird zunächst davon ausgegangen, dass durch die Nutzung des *Motion Template Full* mit überschaubarem Programmier- und Parametriereraufwand die Sammlung der eigenen Anforderungen an das Automatisierungsprogramm und dessen Funktionen erfüllt werden. Dies gilt im anschließenden Unterkapitel in den jeweiligen Testfällen zu bestätigen.

Zunächst findet eine schrittweise Auflistung zur Umsetzung der Steuerungssoftware aus dem bereitgestelltem Template statt. Diese unterteilt sich in folgende Vorgehensschritte:

- Anlegen des Projektes
- Zuweisen der Ein- und Ausgänge der Steuerungen und deren Erweiterungsmodule (engl. Mapping)
- Parametrierung und Inbetriebnahme des Servoreglers für die x-Achse
- Parametrierung und Inbetriebnahme des Servoreglers für die z-Achse
- Parametrierung und Inbetriebnahme des Netzteils
- Implementation der funktionalen Sicherheit

Für das **Anlegen des Projektes** wird die Software Machine Expert Logic Builder auf den Rechnern des Labores benötigt. Alternativ kann auch die veraltete Software SoMachine Logic Builder genutzt werden. Soll von jedem PC aus das System programmiert und zunächst auch gesteuert werden, so muss die Programmiersoftware vorhanden sein. Ist dies der Fall, so muss anschließend wie folgt vorgegangen werden:

1. *LogiBuilder* Programm am Laborcomputer starten

2. Neues Projekt aus Projektvorlage *MotionTemplateFull* für LMC300/400/402/600/800 anlegen
3. Im Projektbaum oben links Doppelklick auf *LMC\_PacDrive*
4. Anschließend im Reiter *Steuerungsauswahl* den LMC400c auswählen (vorher sichergehen, dass der Controller eingeschaltet ist, sonst kann dieser nicht gefunden werden)
5. Im Projektbaum unter *Sercos\_Master (Sercos Master)* alle untergeordneten Geräte markieren und anschließen entfernen
6. Etwas tiefer im Projektbaum Doppelklick auf *Geräteaddressierung*
7. Im neu geöffneten Fenster oben rechts den Button *Sercos Scan starten* drücken
8. Es erscheinen neue tabellarisch angeordnete Einträge in der Mitte des Fensters (Einträge sind rot eingefärbt)
9. Unten rechts im offenen Fenster auf *Geräteparameter übernehmen* klicken (Alle vorhandenen Einträge sollten die Farbe zu grün wechseln)
10. Den *IEC Bezeichner* des Powersupplies ändern zu *PSM\_PowerSupply*
11. Den *IEC Bezeichner* des von Drive A und B ändern zu *DRV\_Slave1* bzw. *DRV\_Slave2*
12. Oben links im offenen Fenster über Kombobox drei neue LXM62DxS hinzufügen
13. *IEC Bezeichner* des ersten neuen Eintrages auf *DRV\_Master* ändern (Wichtig: Gerät sollte auf virtuell eingestellt sein)
14. *IEC Bezeichner* der beiden anderen Einträge ändern zu *DRV\_Slave3* bzw. *DRV\_Slave4* (Wichtig: Geräte sollte auf virtuell eingestellt sein)

Geräte der Steuerungsconfiguration								Gescannte Geräte	
Topolog. Adresse	IEC-Bezeichner	Typ	Geräte-Seriennr.	Motor-Seriennumm.	Anwendungstyp	SERCOS-A	Identifikationsmodus	Betriebsart	Gescannte, 0 zugeordnete Geräte
1	PSM_PowerSupply	LXM62DxS	2910053203		PowerSupply	1	Topologische A... ✓	Real ✓	...
2	DRV_Slave1	LXM62DxS	2910018244-A	2800573624	Drive A	2	Topologische A... ✓	Real ✓	...
3	DRV_Slave2	LXM62DxS	2910018244-B	2800573623	Drive B	3	Topologische A... ✓	Real ✓	...
4	SLC_TMS5C10x	TM5CLC10...	B3560170926		Modular Safety D...	4	Topologische A... ✓	Real ✓	...
5	BC_TMNS31	TM5H31	B37C0196882		Modular IO Device	5	Topologische A... ✓	Real ✓	...
6	DRV_Slave3	LXM62DxS			DRV_LexumS2	100	Topologische A... ✓	Virtuell ✓	...
7	DRV_Slave4	LXM62DxS			DRV_LexumS2	100	Topologische A... ✓	Virtuell ✓	...
8	DRV_Master	LXM62DxS			DRV_LexumS2	100	Topologische A... ✓	Virtuell ✓	...

Farbenlegende:	>Login fehlerfrei möglich	Kein fehlerfreier Login möglich	Kein gescanntes Gerät zugeordnet
	>Login fehlerfrei möglich: Unterschiede in irrelevanten Werten werden jedoch hervorgehoben		
	Login fehlerfrei möglich: gescanntes Gerät wird jedoch nicht zugeordnet		
	Login möglich: eingerichteter Identifikationsmodus wird jedoch nicht unterstützt		

Werte aller zugeordneten Geräte übernehmen

Abbildung 15: Geräteaddressierung der SERCOS III Busteilnehmer

Um die anliegenden Sensor- und Eingabegerätesignale an den SPS Eingängen, sowie die Aktuatoren und Indikatoren an den SPS Ausgängen nutzen zu können, müssen den Hardwareadressen im Programm Variablen zugewiesen werden (engl. **Mapping**). Damit alle E/A-Variablen an einem Ort als auch global im gesamten Programm verfügbar sind, sollte eine Globale Variablenliste angelegt werden, in der alle Variablen eingetragen werden können. Folgendes Bild zeigt die Globale Variablenliste für die Ein- und Ausgänge des mehrachsigen Positioniersystems.

```

2 | VAR_GLOBAL
3 | // Modicon TMS digital Inputs
4 | q_ixIN_00: BOOL; // Endl. Oben
5 | q_ixIN_01: BOOL; // Endl. Unten
6 | q_ixIN_02: BOOL; // Endl. Links
7 | q_ixIN_03: BOOL; // Endl. Rechts
8 | q_ixIN_04: BOOL; // Tst. Reg. Oben
9 | q_ixIN_05: BOOL; // Tst. Reg. Unten
10 | q_ixIN_06: BOOL; // Tst. Reg. Links
11 | q_ixIN_07: BOOL; // Tst. Reg. Rechts
12 | q_ixIN_08: BOOL; // Tst. Grün
13 | q_ixIN_09: BOOL; // Tst. Rot
14 | q_ixIN_10: BOOL; // Tst. Weiß Oben
15 | q_ixIN_11: BOOL; // Tst. Weiß Unten
16 | q_ixIN_12: BOOL; // Schlit. Links (Betr. Mod.)
17 | q_ixIN_13: BOOL; // Schlit. Rechts (Betr. Mod.)
18 | q_ixIN_14: BOOL; // LV. OSSDI
19 | q_ixIN_15: BOOL; // LV. OSSD2
20 |
21 | // Modicon TMS digital Outputs
22 | q_qxOUT_00: BOOL; // Sign. A. Rot
23 | q_qxOUT_01: BOOL; // Sign. A. Grün
24 | q_qxOUT_02: BOOL; // Tst. LED Grün
25 | q_qxOUT_03: BOOL; // Tst. LED Rot
26 | q_qxOUT_04: BOOL; // Tst. LED Weiß Oben
27 | q_qxOUT_05: BOOL; // Tst. LED Weiß Unten
28 | q_qxOUT_06: BOOL; // ***
29 | q_qxOUT_07: BOOL; // ***
30 | q_qxOUT_08: BOOL; // ***
31 | q_qxOUT_09: BOOL; // ***
32 | q_qxOUT_10: BOOL; // ***
33 | q_qxOUT_11: BOOL; // ***
34 | q_qxOUT_12: BOOL; // ***
35 | q_qxOUT_13: BOOL; // ***
36 | q_qxOUT_14: BOOL; // ***
37 | q_qxOUT_15: BOOL; // ***
38 |
39 | // Modicon TMS analog Inputs
40 | q_ixIN_00: REAL; // Pot. Oben (z-Achse)
41 | q_ixIN_01: REAL; // Pot. Unten (x-Achse)
42 | q_ixIN_02: REAL; // ***
43 | q_ixIN_03: REAL; // ***
44 |
45 | // Modicon TMS analog Outputs
46 | q_qxOUT_00: REAL; // ***
47 | q_qxOUT_01: REAL; // ***
48 | q_qxOUT_02: REAL; // ***
49 | q_qxOUT_03: REAL; // ***
50 |
51 | // Modicon TMS SLC100 safe digital Inputs
52 | q_ixSAFE_IN_00: BOOL; // Not-Halt Öffner
53 | q_ixSAFE_IN_01: BOOL; // Not-Halt Schlieder
54 | q_ixSAFE_IN_02: BOOL; // LV. OSSDI
55 | q_ixSAFE_IN_03: BOOL; // LV. OSSD2
56 |
57 | // Modicon TMS SLC100 safe digital Outputs
58 | q_qxSAFE_OUT_00: BOOL; // ***
59 | q_qxSAFE_OUT_01: BOOL; // ***
60 | q_qxSAFE_OUT_02: BOOL; // ***
61 | q_qxSAFE_OUT_03: BOOL; // ***
62 |
63 | // Release of the safety Outputs (required by the safety Controller)
64 | q_qxRelease_00: BOOL;
65 | q_qxRelease_01: BOOL;
66 | q_qxRelease_02: BOOL;
67 | q_qxRelease_03: BOOL;
68 |
69 | // Safety Variables
70 | bHardwareOK : BOOL; // Check if Safety Modules are working
71 | bError_Out : BOOL; // Error Status from Safety Blocks
72 | bEmergency : BOOL; // Emergency Stop Status
73 | bOSDI : BOOL; // Lightstatus
74 | bOSD2 : BOOL; // Lightstatus
75 |
76 | END_VAR

```

Abbildung 16: Globale Variablenliste der E/A-Variablen

Nachfolgend wird das Mapping schrittweise für die Ein- und Ausgangsmodule der mit dem LMC verbundenen Modicon TM5 Geräte vorgenommen. Bei den zuzuordnenden Variablen handelt es sich um die im Datenmodell aufgelisteten Variablen. Dieses dient somit als Grundlage für die Zuweisung der Ein- und Ausgänge.

- Zuweisung der digitalen Eingänge des Modicon *TM5SDI16D* Modul:
1. Im Gerätebaum öffnen der Geräteeinstellungen des Moduls unter *Sercos\_Master* (*Sercos Master*) ->*BC\_TM5NS31* (*BC\_TM5NS31*) -> *TM5SDI16D* (*TM5SDI16D*)
  2. Öffnen des Reiters *SERCOS III Module E/A-Abbild*

## 5.1 Implementationsphase

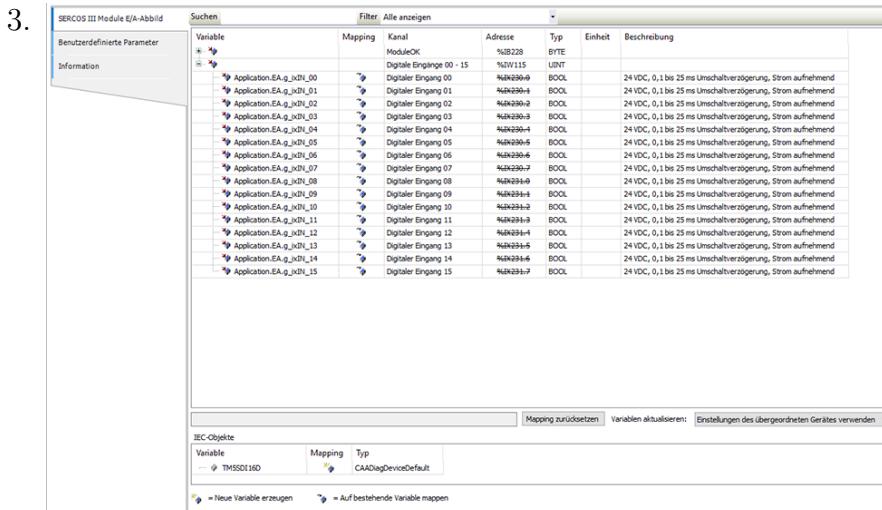


Abbildung 17: Zuweisung der digitalen Eingänge des TM5SDI16D Moduls

- Zuweisung der digitalen Ausgänge des Modicon *TM5SDO16T* Modul:

- Im Gerätebaum öffnen der Geräteeinstellungen des Moduls unter *Sercos\_Master* (*Sercos Master*) -> *BC\_TM5NS31* (*BC\_TM5NS31*) -> *TM5SDO16T* (*TM5SDO16T*)
- Öffnen des Reiters *SERCOS III Module E/A-Abbild*

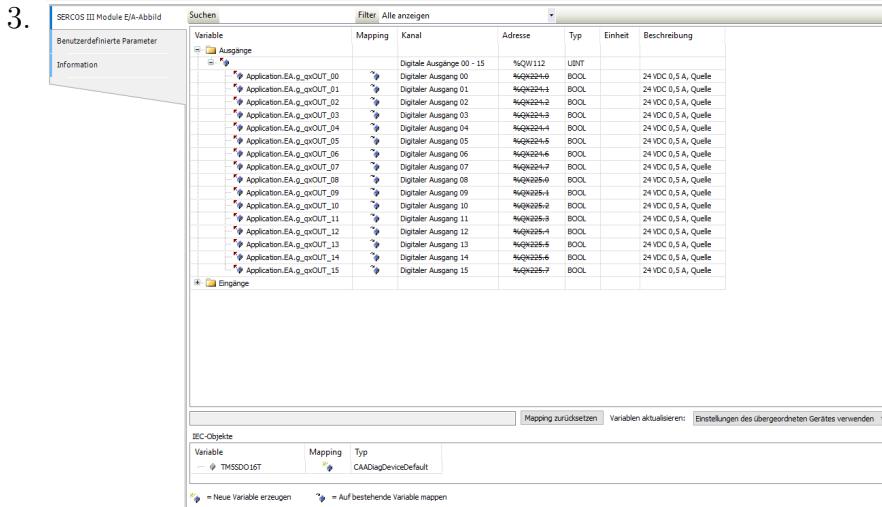


Abbildung 18: Zuweisung der digitalen Ausgänge des TM5SDO16T Moduls

- Zuweisung der sicheren digitalen Eingänge des Modicon *TM5SDI4DFS* Modul:

- Im Gerätebaum öffnen der Geräteeinstellungen des Moduls unter *Sercos\_Master* (*Sercos Master*) -> *BC\_TM5NS31* (*BC\_TM5NS31*) -> *TM5SDI4DFS*
- Öffnen des Reiters *TM5 Modul E/A-Abbild*

3.

TMS Modul E/A-Abbild						
	Suchen	Filter	Alle anzeigen			
TMS Modul Parameter	Variable	Mapping	Kanal	Adresse	Typ	Einheit
Benutzerdefinierte Parameter	= Application.EA_g_qsSAFE_IN_00	"	SafeDigitalInput01	%I0:248	BYTE	
	= Application.EA_g_qsSAFE_IN_01	"	SafeDigitalInput02	%I0:249	BOOL	Sicherer digitaler Eingang 01 (24VDC)
	= Application.EA_g_qsSAFE_IN_02	"	SafeDigitalInput03	%I0:250	BOOL	Sicherer digitaler Eingang 02 (24VDC)
	= Application.EA_g_qsSAFE_IN_03	"	SafeDigitalInput04	%I0:251	BOOL	Sicherer digitaler Eingang 03 (24VDC)
			SafeDigitalInput05	%I0:252	BOOL	Sicherer digitaler Eingang 04 (24VDC)
			SafeDigitalInput06	%I0:253	BOOL	Zwei Kanalauflösung äquivalent sicherer digitaler Eingang 01/02
			SafeDigitalInput07	%I0:254	BOOL	Zwei Kanalauflösung äquivalent sicherer digitaler Eingang 03/04
			SafeDigitalInput08	%I0:255	BOOL	Zwei Kanalauflösung äquivalent sicherer digitaler Eingang 03/04
			SafeDigitalInput09	%I0:256	BOOL	Zwei Kanalauflösung äquivalent sicherer digitaler Eingang 03/04
			SafeDigitalInput10	%I0:257	BOOL	Zwei Kanalauflösung äquivalent sicherer digitaler Eingang 03/04
			SafeDigitalInput11	%I0:258	BOOL	
			SafeDigitalInput12	%I0:259	BOOL	Status Kanal sicherer digitaler Eingang 02 (1=0Q)
			SafeDigitalInput13	%I0:260	BOOL	Status Kanal sicherer digitaler Eingang 03 (1=0Q)
			SafeDigitalInput14	%I0:261	BOOL	Status Kanal sicherer digitaler Eingang 04 (1=0Q)
			SafeDigitalInput15	%I0:262	BOOL	Status zwei Kanalauflösung äquivalent sicherer digitaler Eingang 01/02
			SafeDigitalInput16	%I0:263	BOOL	Status zwei Kanalauflösung äquivalent sicherer digitaler Eingang 03/04
			SafeDigitalInput17	%I0:264	BOOL	Status zwei Kanalauflösung äquivalent sicherer digitaler Eingang 03/04

IEC-Objekte

Variable	Mapping	Typ
M5SDO4DFS	CADaylightDeviceDefault	

= Neue Variable erzeugen    = Auf bestehende Variable mappen

Abbildung 19: Zuweisung der sicheren digitalen Eingänge des TM5SDI4DFS Moduls

- Zuweisung der sicheren digitalen Ausgänge des Modicon *TM5SDO4TFS* Modul:

- Im Gerätebaum öffnen der Geräteeinstellungen des Moduls unter *Sercos\_Master* (*Sercos Master*) -> *BC\_TM5NS31* (*BC\_TM5NS31*) -> *TM5SDO4TFS*
- Öffnen des Reiters *TM5 Modul E/A-Abbild*

3.

TMS Modul E/A-Abbild						
	Suchen	Filter	Alle anzeigen			
TMS Modul Parameter	Variable	Mapping	Kanal	Adresse	Typ	Einheit
Benutzerdefinierte Parameter	= Application.EA_g_qsSAFE_OUT_00	"	SafeDigitalOutput1-4	%Q0:234	BYTE	Sicherer digitaler Ausgang 01 (Zustimmignal, 24VDC / 0.5A)
	= Application.EA_g_qsSAFE_OUT_01	"	SafeDigitalOutput01	%Q0:235	BOOL	Sicherer digitaler Ausgang 02 (Zustimmignal, 24VDC / 0.5A)
	= Application.EA_g_qsSAFE_OUT_02	"	SafeDigitalOutput02	%Q0:236	BOOL	Sicherer digitaler Ausgang 03 (Zustimmignal, 24VDC / 0.5A)
	= Application.EA_g_qsSAFE_OUT_03	"	SafeDigitalOutput03	%Q0:237	BOOL	Sicherer digitaler Ausgang 04 (Zustimmignal, 24VDC / 0.5A)
			Status Module	%I0:252	BOOL	

IEC-Objekte

Variable	Mapping	Typ
TM5SDO4TFS	CADaylightDeviceDefault	

= Neue Variable erzeugen    = Auf bestehende Variable mappen

Abbildung 20: Zuweisung der sicheren digitalen Ausgänge des TM5SDO4TFS Moduls

- Zuweisung der analogen Eingänge des Modicon *TM5SAI4L* Modul:

- Im Gerätebaum öffnen der Geräteeinstellungen des Moduls unter *Sercos\_Master* (*Sercos Master*) -> *BC\_TM5NS31* (*BC\_TM5NS31*) -> *TM5SAI4L*
- Öffnen des Reiters *TM5 Module E/A-Abbild*

3.

The screenshot shows the 'TMS Module E/A-Abbildung' configuration window. The main table lists analog inputs (AnalogInput 0-3) mapped to Modicon variables (%W119). The table columns are: Variable, Mapping, Kanal, Adresse, Typ, Einheit, and Beschreibung. The IEC-Objekte section shows the target variable TM5SAO4L mapped to CAADagDeviceDefault.

Variable	Mapping	Kanal	Adresse	Typ	Einheit	Beschreibung
Application.EA_g_IN_00			%B236	BYTE		
		Status Eingang 00				
Application.EA_g_IN_01			%B237	BYTE		
Application.EA_g_IN_02			%B238	BOOL		
Application.EA_g_IN_03			%B239	BOOL		
			%B240	BOOL		
			%B241	BOOL		
			%B242	BOOL		
			%B243	BOOL		
			%B244	BOOL		
			%B245	BOOL		
			%B246	BOOL		
		AnalogInput 0-3	%W119			

IEC-Objekte

Variable	Mapping	Typ
TM5SAI4L		CAADagDeviceDefault

= Neue Variable erzeugen    = Auf bestehende Variable mappen

Abbildung 21: Zuweisung der analogen Eingänge des TM5SAI4L Moduls

- Zuweisung der analogen Ausgänge des Modicon *TM5SAO4L* Modul:

1. Im Gerätebaum öffnen der Geräteeinstellungen des Moduls unter *Sercos\_Master* (*Sercos Master*) -> *BC\_TM5NS31* (*BC\_TM5NS31*) -> *TM5SAO4L*

2. Öffnen des Reiters *TM5 Module E/A-Abbildung*

3.

The screenshot shows the 'TMS Module E/A-Abbildung' configuration window. The main table lists analog outputs (AnalogOutput 0-3) mapped to Modicon variables (%QW113). The table columns are: Variable, Mapping, Ausgang, Adresse, Typ, Einheit, and Beschreibung. The IEC-Objekte section shows the target variable TM5SAO4L mapped to CAADagDeviceDefault.

Variable	Mapping	Ausgang	Adresse	Typ	Einheit	Beschreibung
Application.EA_g_OUT_00		Analoger Ausgang 00	%QW113	INT		±10 V / 0 bis 20 mA, Auflösung 12 Bits
Application.EA_g_OUT_01		Analoger Ausgang 01	%QW114	INT		±10 V / 0 bis 20 mA, Auflösung 12 Bits
Application.EA_g_OUT_02		Analoger Ausgang 02	%QW115	INT		±10 V / 0 bis 20 mA, Auflösung 12 Bits
Application.EA_g_OUT_03		Analoger Ausgang 03	%QW116	INT		±10 V / 0 bis 20 mA, Auflösung 12 Bits
		ModuleOK	%B246	BYTE		

IEC-Objekte

Variable	Mapping	Typ
TM5SAO4L		CAADagDeviceDefault

= Neue Variable erzeugen    = Auf bestehende Variable mappen

Abbildung 22: Zuweisung der analogen Ausgänge des TM5SAO4L Moduls

- Zuweisung der Ein- bzw. Ausgänge der *SLC\_TM5CSLC* Sicherheitssteuerung für den Datenaustausch mit dem LMC:

1. Im Gerätebaum öffnen der Geräteeinstellungen der Sicherheitssteuerung unter *Sercos\_Master* (*Sercos Master*) -> *SLC\_TM5CSLCx00FS* (*SLC\_TM5CSLCx00FS*)

2. Öffnen des Reiters *E/A-Abbildung*

3.

Parameter	Suchen	Filter	Alle anzeigen				
E/A-Abbildung	Variable	Mapping	Kanal	Adresse	Typ	Einheit	Beschreibung
Information	SLC2MC_BOOL			SLC2MC_BOOL0-7	%B0	BYTE	
	Application.EA.bError_Out		Boo1	%B0#0-9	BOOL		
	Application.EA.bHardwareOK		Boo2	%B0#0-8	BOOL		
	Application.EA.bIOSD1		Boo3	%B0#0-2	BOOL		
	Application.EA.bIOSD2		Boo4	%B0#0-4	BOOL		
	Application.EA.bIOSSD1		Boo5	%B0#0-5	BOOL		
	Application.EA.bIOSSD2		Boo6	%B0#0-6	BOOL		
	Application.EA.bIOSSD3		Boo7	%B0#0-7	BOOL		
	SLC2MC_BOOLExt			SLC2MC_BOOL0-15	%B0	BYTE	
	SLC2MC_BOOLExt			SLC2MC_BOOL16-23	%B2	BYTE	
	SLC2MC_INT			SLC2MC_BOOL24-31	%B3	BYTE	
	SLC2MC_INT			SLC2MC_BOOL32-39	%B4	BYTE	
	SLC2MC_INT			SLC2MC_BOOL40-47	%B5	BYTE	
	SLC2MC_INT			SLC2MC_BOOL48-55	%B6	BYTE	
	SLC2MC_INT			SLC2MC_BOOL56-63	%B7	BYTE	
	SLC2MC_INT			SLC2MC_BOOL64-71	%B8	BYTE	
	SLC2MC_INT			SLC2MC_BOOL72-79	%B9	BYTE	
	SLC2MC_INT			SLC2MC_BOOL80-87	%B10	BYTE	
	SLC2MC_INT			SLC2MC_BOOL88-95	%B11	BYTE	
	SLC2MC_BOOLExt						
	SLC2MC_INT						
	SLC2MC_LINT						
	SLC2MC_LINT						
	UMC2SLC_BOOL						
	UMC2SLC_BOOLExt						
	UMC2SLC_INT						
	UMC2SLC_INT						

IEC-Objekte

Variable	Mapping	Typ
SLC_TM5CSLCx00FS	SLC_PLC	

= Neue Variable erzeugen    = Auf bestehende Variable mappen

Abbildung 23: Zuweisung der Variablen für den Datentransfer vom SLC zum LMC

4.

Parameter	Suchen	Filter	Alle anzeigen				
E/A-Abbildung	Variable	Mapping	Kanal	Adresse	Typ	Einheit	Beschreibung
Information	SLC2MC_BOOL			LMC2SLC_BOOL0-7	%Q00	BYTE	
	SLC2MC_BOOLExt		Boo1	%Q0#0-9	BOOL		
	SLC2MC_INT		Boo2	%Q0#0-8	BOOL		
	SLC2MC_INT		Boo3	%Q0#0-2	BOOL		
	SLC2MC_INT		Boo4	%Q0#0-4	BOOL		
	SLC2MC_INT		Boo5	%Q0#0-5	BOOL		
	SLC2MC_INT		Boo6	%Q0#0-6	BOOL		
	SLC2MC_INT		Boo7	%Q0#0-7	BOOL		
	SLC2MC_BOOLExt			SLC2MC_BOOL0-15	%Q1	BYTE	
	SLC2MC_BOOLExt			SLC2MC_BOOL16-23	%Q2	BYTE	
	SLC2MC_INT			SLC2MC_BOOL24-31	%Q3	BYTE	
	SLC2MC_INT			SLC2MC_BOOL32-39	%Q4	BYTE	
	SLC2MC_INT			SLC2MC_BOOL40-47	%Q5	BYTE	
	SLC2MC_INT			SLC2MC_BOOL48-55	%Q6	BYTE	
	SLC2MC_INT			SLC2MC_BOOL56-63	%Q7	BYTE	
	SLC2MC_INT			SLC2MC_BOOL64-71	%Q8	BYTE	
	SLC2MC_INT			SLC2MC_BOOL72-79	%Q9	BYTE	
	SLC2MC_INT			SLC2MC_BOOL80-87	%Q10	BYTE	
	SLC2MC_INT			SLC2MC_BOOL88-95	%Q11	BYTE	
	SLC2MC_BOOLExt						
	SLC2MC_INT						
	SLC2MC_LINT						
	SLC2MC_LINT						

IEC-Objekte

Variable	Mapping	Typ
SLC_TM5CSLCx00FS	SLC_PLC	

= Neue Variable erzeugen    = Auf bestehende Variable mappen

Abbildung 24: Zuweisung der Variablen für den Datentransfer vom LMC zum SLC

Für die Nutzung der beiden Achsen des Systems muss die **Parametrierung und die Inbetriebnahme des Servoreglers** vorgenommen werden. Physisch gesehen handelt es sich zwar um ein Gerät, dass zwei Servoantriebe betreiben kann, in der Konfiguration im Programm werden die x-Achse und die z-Achse jedoch separat in Betrieb genommen. Die Parametrierung und die Inbetriebnahme erfolgt für beide Achsen Analog, da es sich bei beiden um die selben Motoren handelt, die in der selben Konfiguration genutzt werden. Zunächst müssen einige physikalische Daten der beiden Achsen und deren zugehörigen Hardwarekomponenten in der Geräteparametrierung aufgenommen werden:

1. Im Projektbaum öffnen der Datei *Application -> TemplateFullProgrammingFramework -> EquipmentModules -> SR\_BravoModule (PRG) -> Init\_Slave1*
2. Bis zum Kommentar **\*\*\*Manual\*\*\*** scrollen (Zeile 171)

3.

```
/* *** Manual ***

stSlaveInterface.stManual.i_lrVel           := 50.0;          /* Velocity in units/s */
stSlaveInterface.stManual.i_lrAcc           := 10.0;          /* Acceleration in units/s^2 */
stSlaveInterface.stManual.i_lrDec           := 1000.0;        /* Deceleration in units/s^2 */
stSlaveInterface.stManual.i_lrJerk          := 1000.0;        /* Jerk in units/s^3 */
stSlaveInterface.stManual.i_lrMaxDistance   := 120.0;         /* Max pathlengths for one step in units */
stSlaveInterface.stManual.i_xEndless        := FALSE;         /* TRUE: jogging endless: Position between the periods */
stSlaveInterface.stManual.i_lrPeriod         := 360.0;         /* Period of the axis */
```

Abbildung 25: Einstellen der Bewegungsparameter für langsame manuelle Testfahrt

rVel gibt die Geschwindigkeit in mm/s an, rAcc die Beschleunigung in mm/s<sup>2</sup> und rDec die negative Beschleunigung mm/s<sup>2</sup>.

4. Im Projektbaum öffnen der Gerätedatei *Sercos\_Master (Sercos Master) -> DRV\_Slave1*

5.

MotorType	STRING(20)	=	=	Mot
MotorTypePlateVersion	STRING(20)	=	=	Vers
PartNumberMotor	STRING(20)	=	=	Serv
SerialNumberMotor	STRING(20)	=	=	Serv
NominalSpeed	LREAL			UPM
MaxSpeed	REAL			UPM
TorqueConstant	LREAL			Nm/A
MotorInertia	LREAL			kg*cm^2
MotorClass	Enumeration of DINT	Synchronmotor / 0	Synchronmotor / 0	Mot
BrakeIntegrated	Enumeration of BOOL			Brem
BrakePowerOffBehaviour	Enumeration of BOOL			Verh
BrakeState	Enumeration of BOOL			Brem
BrakeMode	Enumeration of DINT	Bremse beim Austrudeln koppeln / 1	Bremse beim Austrudeln koppeln / 1	Betr
BrakeDisconnectionTime	DINT(0..1000)	0	0 ms	Verk
BrakeCouplingTime	DINT(0..1000)	0	0 ms	Klem
MotorConnectionTest	Enumeration of BOOL	ein / 1	ein / 1	Prüf
MotorCommutationMode	Enumeration of DINT	minimale Bewegung / 1	minimale Bewegung / 1	Mod
MotorCommutationControl	Enumeration of DINT	aus / 0	aus / 0	Aktiv
MotorCommutationState	Enumeration of INT	Motor ist nicht kommutiert / 0	Motor ist nicht kommutiert / 0	Zust
MotorTemperatureMonitoring	Enumeration of DINT	Thermisches Modell / 1	Sensor / 2	Betr

Abbildung 26: Parametrierung des Motors für die x-Achse

Der Wert *MotorTemperatureMonitoring* wurde zur Temperaturüberwachung des Servomotors auf *Thermisches Modell* gesetzt.

6.

GearIn	DINT(1..999999)	1	1	Getr
GearOut	DINT(1..999999)	1	1	Getr
Direction	Enumeration of BOOL	rechts / 1	rechts / 1	Dreh
FeedConstant	LREAL(0.01..999999.999)	188.4	360.0 Einheiten/Umdrehung	Vors
J_Load	REAL(0..1000000000)	0.8588328916751	0.0 kg*cm^2	Last
J_Gear	REAL(0..1000)	0.0	0.0 kg*cm^2	Träg
StatFriction	REAL(0..1000)	0.0	0.0 Nm	Stat
ViscousFriction	REAL(0..1000000)	0.0	0.0 Nm / (1000 Einheiten/s)	Visk

Abbildung 27: Parametrierung der Mechanik für die x-Achse

Der Wert *Direction* gibt den Drehsinn des Servomotors an. Die Einstellung *rechts / 1* bedeutet, dass bei der Bedienung des Motors über die Steuerungsvisualisierung

die Eingabe *Jog+* zu einer Bewegungs der Achse nach links führt. Der Wert *Feeder-Constant* muss selbst berechnet werden aus dem Motordurchmesser ( $\pi * M60\text{mm}$ ). Der *JLoad* Wert gibt das Lastenträgheitsmoment in  $\text{kg} * \text{cm}^2$  an. Der einzusetzende Wert kann über die Software *Machine Expert MotionBuilder* bestimmt werden.

Die gleichen Schritte müssen für die z-Achse analog zu der obigen Beschreibung durchgeführt werden.

Für den sicheren Betrieb der Achsen ist es erforderlich im *MotionTemplateFull* Programmänderungen vorzunehmen, damit die Achsen bei den Endlagesensoren des Positioniersystems stoppen und nicht darüber hinaus bewegt werden können. Folgende Schritte sind dazu nötig:

1. Im Projektbaum öffnen der Datei *Application -> TemplateFullProgrammingFramework -> EquipmentModules -> SR\_BravoModule (PRG) -> SubModules\_Action*
2. Bis zum Abschnitt 3 Scrollen und diesen auskommentieren oder löschen
3. Abschnitt 4 (beinhaltet den Funktionsbaustein und Variablenzuweisungen für den *DRV\_Slave2* bzw. die z-Achse) kopieren und wieder einfügen (duplizieren)
4. Die Kopie editieren, so dass an jeder Stelle, wo *DRV\_Slave2* aufgeführt ist nun *DRV\_Slave1* steht

5.



Abbildung 28: Einbinden der Endlagesensoren

Sowohl für den *DRV\_Slave1* als auch den *DRV\_Slave2* die globalen Variablen für die Endlagesensoren an den Stellen *i\_xHwLimitPos* bzw. *i\_xHwLimitNeg* einsetzen.

Im nächsten Schritt wird die **Parametrierung und Inbetriebnahme des Netzteils** durchgeführt. Dafür muss zunächst wieder eine Hardwareparametrierung nach folgenden Schritten vorgenommen werden:

1. Im Projektbaum öffnen der Gerätedatei *Sercos\_Master (Sercos Master) -> PSM\_PowerSupply*

2.		Allgemeines und Status	DINT				
		DiagClass	DINT				Diagnoseklasse (AD)[0x00CD]
		DiagCode	DINT				Diagnosenummer (AD)[0x00001]
		DiagSource					Diagnosequelle (AD)[0x00CE]
		udType	UDINT	0	0		
		udInstance	UDINT	0	0		
		udParameterId	UDINT	0	0		
		DiagTxt	STRING(38)	"	"		Diagnosetext (AD)[0x0003]
		DiagExtMsg	STRING(14)	"	"		Erweiterte Diagnosemeldung (AD)
		PowerSupplyCheckSet	Enumeration of BOOL	aus / 0	aus / 0		Leistungsversorgung-Überwachung
		PowerSupplyCheck	Enumeration of BOOL				Status der Leistungsversorgung
		PhaseCheckMode	Enumeration of DINT	Dreiphasige Überwachung / 2	Dreiphasige Überwachung / 2		Phasenüberwachungsmodus (EF)
		MainsVoltageMode	Enumeration of DINT	400V / 2	400V / 2		Netzspannung (ES)[0x0000]
		BrakingResistorMode	Enumeration of DINT	400V Modus / 1	400V Modus / 1		Bremsewiderstand Modus (EF)[0x0000]
		Ready	Enumeration of BOOL				Betriebsbereit (AD)[0x0008]
		GroupState	Enumeration of DINT	Initialisierung / 0	Initialisierung / 0		Betriebszustand (Netzteil und Anl)
		GroupReady	Enumeration of BOOL				Bereit fuer Auftragbearbeitung
		InternalDeviceState	Enumeration of DINT	Initialisierung / 0	Initialisierung / 0		Gerätezustand (AS)[0x0010]
		PowerSupplyOverload	INT		%		Überlast des Netzteils (AS)[0x0011]
		AutoDischarge	Enumeration of BOOL	Aktiviere automatische Entladung / 1	Aktiviere automatische Entladung / 1		Automatische Entladung des Zws
		ControlBoardTemp	INT(-10..150)	0	0 °C		Temperatur der Steuerplatine (A)
		RectifierTemp	INT(-10..150)	0	0 °C		Temperatur der Gleichrichters (A)

Abbildung 29: Konfiguration des Netzteils

Der *PhaseCheckerMode* muss auf *Dreiphasige Überwachung / 2* eingestellt werden, da das Netzteil dreiphasig angeschlossen und genutzt wird. Der *MainsVoltageMode* wird gesetzt auf *400V / 2* und der *BrakingResistorMode* wird ebenfalls gesetzt auf *400V Modus / 1*. Diese Werte ergeben sich ebenfalls aus dem dreiphasigen Anschluss des Netzteils.

Je nachdem, ob bei der Verdrahtung ein Netzschütz verbaut oder diese weggelassen wurde, müssen nun weitere Programmänderungen am *MotionTemplateFull* vorgenommen werden. An erster Stelle wird erklärt, wie ohne ein Netzschütz vorgegangen werden muss:

1. Im Projektbaum öffnen der Datei *Application -> TemplateFullProgrammingFramework -> TaskCalls -> SR\_MainMachine (PRG) -> Input\_Action*

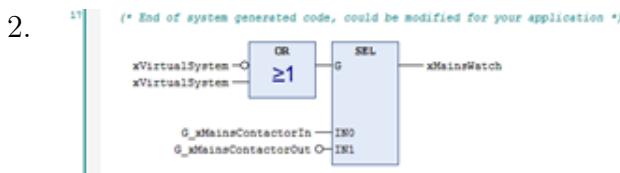


Abbildung 30: Deaktivierung der Netzüberwachung

Das Netzwerk in Abschnitt 17 muss erweitert werden um einen *OR-Funktionsbaustein*, zu dem die Eingangsvariable *xVirtualSystem* einmal negiert und einmal nicht-negiert zugewiesen wird. Diese Änderung deaktiviert die Netzüberwachung. Grundsätzlich sollte dieser Weg nicht gewählt und ein Netzschatz verbaut werden.

Bei der Verwendung eines Netzschatzes sollte wie folgt vorgegangen werden:

1. Mappen der Eingangsvariablen am LMC für die Netzschatzrückmeldung
2. Im Projektbaum öffnen der Datei *Application -> TemplateFullProgrammingFramework -> TaskCalls -> SR\_MainMachine (PRG) -> Input\_Action*
3. Zuweisen der Globalen Variablen für *G\_xMainsContactorIn* und *G\_xMainsContactorOut*

Im letzten Schritt der Software-Implementation findet die **Implementation der funktionalen Sicherheit** statt. Dazu wird zunächst der Safety Logic Controller (SLC) konfiguriert:

1. Im Projektbaum öffnen der Gerätedatei *Sercos\_Master (Sercos Master) -> SLC\_TM5CSLCx00FS (TM5CSLCx00FS)*

2.

I/O-Konfiguration				
SLC2LMC_NumberOfBOOLs	Enumeration of UDINT	8 Bool / 1 0 ...	Anzahl an BOOL-Werten die von der SafeLogic zum LMC übertragen werden	
SLC2LMC_NumberOfBOOLsExt	Enumeration of UDINT	0 Bool / 0 0 ...	Anzahl an BOOL-Ext-Werten die von der SafeLogic zum LMC übertragen werden	
SLC2LMC_NumberOfINTs	UINT(0..30)	0 0	Anzahl an INT-Werten die von der SafeLogic zum LMC übertragen werden	
SLC2LMC_NumberOfUINTs	UINT(0..30)	0 0	Anzahl an UINT-Werten die von der SafeLogic zum LMC übertragen werden	
SLC2LMC_NumberOfUDINTs	UINT(0..15)	0 0	Anzahl an UDINT-Werten die von der SafeLogic zum LMC übertragen werden	
LMC2SLC_NumberOfBOOLs	Enumeration of UDINT	8 Bool / 1 0 ...	Anzahl an BOOL-Werten die vom LMC zu SafeLogic übertragen werden (EF)	
LMC2SLC_NumberOfBOOLsExt	Enumeration of UDINT	0 Bool / 0 0 ...	Anzahl an BOOL-Ext-Werten die vom LMC zu SafeLogic übertragen werden	
LMC2SLC_NumberOfINTs	UINT(0..30)	0 0	Anzahl an INT-Werten die vom LMC zu SafeLogic übertragen werden (EF)	
LMC2SLC_NumberOfUINTs	UINT(0..30)	0 0	Anzahl an UDINT-Werten die vom LMC zu SafeLogic übertragen werden (EF)	
LMC2SLC_NumberOfUDINTs	UINT(0..15)	0 0	Anzahl an UDINT-Werten die vom LMC zu SafeLogic übertragen werden (EF)	

Abbildung 31: I/O-Konfiguration des SLC

Der Wert *SLC2LMC\_NumberOfBOOLs* muss auf *8 Bool / 1* gesetzt werden, sodass bis zu 8 Bool Werte vom Safety Logic Controller an den LMC übertragen werden können. Analog muss der Wert *LMC2SLC\_NumberOfBOOLs* auf *8 Bool / 1* gesetzt werden, dass bis zu 8 Bool Werte vom **LMC!** an den **SLC!** gesendet werden können.

3. Mapping der neu verfügbaren E/A-Variablen von bzw. zu dem SLC (siehe Abbildung 23 und Abbildung 24)

Anschließend kann das Programm für die Software-Implementation der funktionalen Sicherheit geschrieben werden. Dazu wird wie folgt vorgegangen:

1. Im Projektbaum REchtsklick auf *Sercos\_Master (Sercos Master) -> SLC\_TM5CSLCx00FS (TM5CSLCx00FS)*

2. Im sich geöffneten Menü den Punkt *SoSafe -> SoSafe Programmable starten* auswählen
3. Zunächst muss sowohl ein Passwort für die Entwicklung und die Inbetriebnahme des Sicherheitsprogramms festgelegt werden. Im der Arbeit beigefügten Programm ist das Passwort admin1 für sowohl die Entwicklung als auch die Inbetriebnahme.
4. Es müssen im sich nun geöffneten Dialogfenster alle Module (drei Module) per Checkbox ausgewählt werden
5. Nun kann mit der Programmierung der Sicherheitsfunktionalitäten begonnen werden in der Funktionsbausteinsprache (FUP) nach IEC 61131-3

6.

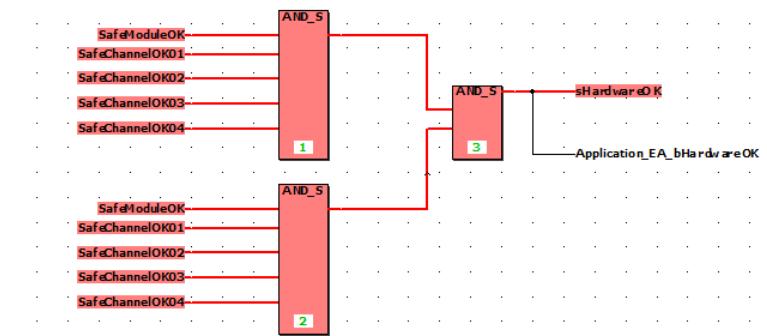


Abbildung 32: Hardwareprüfung der sicheren E/A-Module

Die sichere Variable *sHardwareOK* wird auf den Wert **TRUE** gesetzt, wenn sowohl die beiden E/A-Module keine Hardwarefehler aufweisen und die Verdrahtung der Eingänge und Ausgänge korrekt ist. Weiterhin wird der Wert über die Variable *Application\_EA\_bHardwareOK* auch an den LMC weitergeleitet, um im Hauptprogramm genutzt zu werden (z. B. für die Ausgabe über OPC UA).

7.

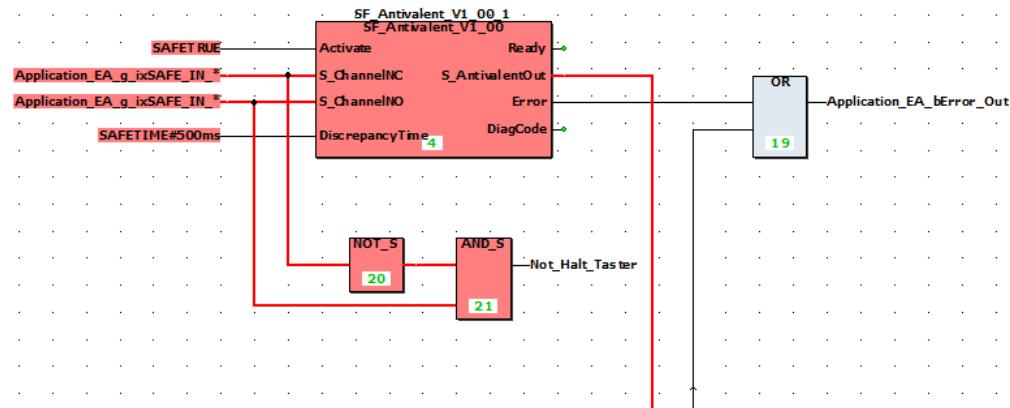


Abbildung 33: Sicherer SF\_Antivalent Funktionbaustein für die Verarbeitung der Not-Halt-Auslösung

Bei dem sicheren Funktionsbaustein **SF\_Antivalent\_V1\_00\_1** handelt es sich um einen Baustein, der die beiden Antivalenten Eingänge für die Not-Halt-Taster Auslösung verarbeitet. Ist die Variable **Application\_EA\_g\_ixSafe\_IN\_01 FALSE** und die Variable **Application\_EA\_g\_ixSafe\_IN\_00 TRUE**, so schaltet der Ausgang **S\_AntivalentOut** auf **TRUE**. Der Wert des Ausgangs wird im nächsten Bild weiterverarbeitet. Für die Weitergabe des Gleichen Wertes an den LMC wird der Baustein nicht benötigt und die beiden Eingänge können per **AND\_S**-Baustein verundet werden (unter Berücksichtigung, dass der Eingang **Application\_EA\_g\_ixSafe\_IN\_01** negiert werden muss). Der **Error** Ausgang kann auch verbunden und die Information an den LMC weitergegeben werden in der Variable **Application\_EA\_bError\_Out**.

8.

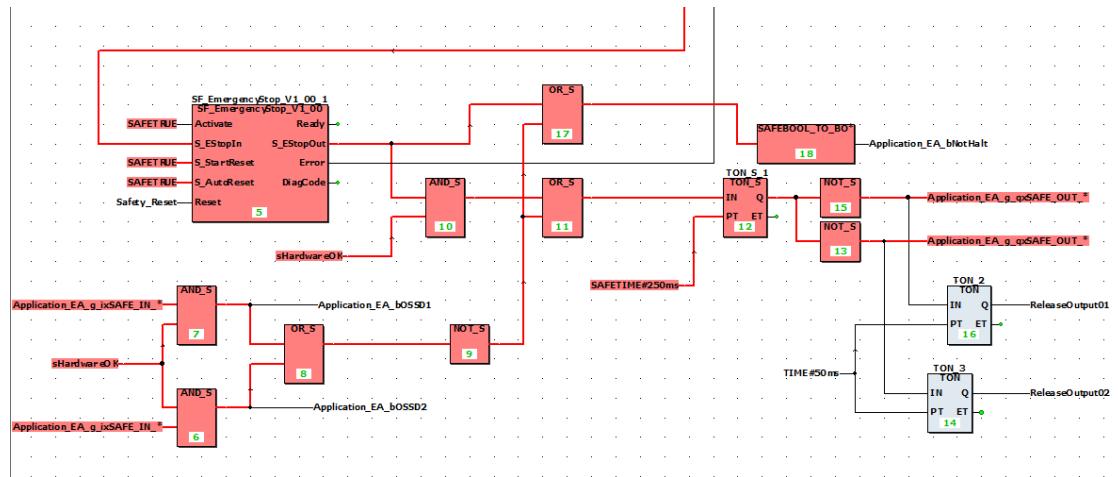


Abbildung 34: Verarbeitung des Not-Halt-Signals und der Lichtvorhangssignale

9. Zunächst wird der sichere Funktionsbaustein *SF\_EmergencyStop\_V1\_00\_1* hinzugefügt. In den Eingang *S\_EStopIn* wird das Ausgangssignal des Antivalent-Funktionsbausteins gelegt. Wird der Not-Halt aktiviert, schaltet der Ausgang *S\_EStopOut* des Bausteins auf **TRUE**. Über den *Reset*-Eingang muss der Baustein wieder Freigegeben werden, so dass der Ausgang wieder auf **FALSE** umschaltet. Auf den Reset-Eingang wird die Variable *>Safety\_Reset* gelegt, die vom LMC an den SLC weitergeleitet wird. Somit kann durch das Auslösen des Hardware Reset-Tasters am LMC der Funktionsbaustein wieder freigegeben bzw. zurückgesetzt werden. Die Eingänge *S\_StartReset* und *S\_AutoReset* sollten auf **SAFEFALSE** gesetzt werden. Im Bild ist jedoch vorerst zu erkennen, dass die Werte auf **SAFETRUE** gesetzt sind, da zum Testen des Sicherheitsprogramms ein automatisches zurücksetzen das Drücken des Reset-Tasters hinfällig macht und somit Zeit spart.
10. Anschließend werden die beiden sicheren Eingänge *Application\_EA\_g\_ixSafe\_IN\_02* und *Application\_EA\_g\_ixSafe\_IN\_03* verarbeitet. Dabei handelt es sich um den Lichtvorhang (OSSD1 und OSSD2). Die beiden Eingänge müssen jeweils verundet werden mit der Variable *bHardwareOK*. Ist eines der beiden Signale des Lichtvorhangs **FALSE**, so muss der Ausgang, dessen Signal später mit dem Not-Halt-Signal verodert wird **TRUE** sein (umgesetzt durch ein *S\_ODER*- und ein *S\_Not*-Baustein).
11. Wie bereits im vorherigen Schritt angedeutet wird nun das Ausgangssignal der LichtvorhangSchaltung mit dem Ausgang der Not-Halt-Schaltung verodert. Wichtig ist, dass das Not-Halt-Signal auch mit dem Wert der Variablen *sHardwareOK* verundet wird, um Hardwarefehler auszuschließen.
12. Im nächsten Schritt wird das Not-Halt-Signal zu einem nicht-sicheren Bool-Wert umgewandelt, welcher über die Variable *Application\_EA\_bNotHalt* an den LMC übergeben wird.
13. Der letzte Schritt in der Programmierung des Sicherheitsprogrammes ist das setzen der Ausgänge des SLC. Wichtig ist zu beachten, dass die Ausgänge mit dem *InverterEnable*-Eingang des Servoreglers (LXM 62D) verbunden sind. Nehmen die Ausgänge den Wert **FALSE** an, schaltet der Servoregler ab. Das soll dann passieren, wenn der Not-Halt ausgelöst bzw. der Lichtvorhang durchbrochen wurde. Jedoch ist es notwendig, dass die Motoren zunächst angehalten haben, bevor der Regler abgeschaltet wird, da sonst bei zu frühem Abschalten des Reglers der Bremsvorgang nicht fortgesetzt wird und die beiden Achsen des Positioniersystems austrudeln würden. Dies könnte dafür sorgen, dass die Schlitten auf den Achsen über die Endlagen hinausrutschen und Beschädigungen an der Anlage verursachen. Deshalb wird über ein *S\_TON*-Baustein die Abschaltung der Ausgänge um 250ms verzögert. In Abbildung 34 ist zu erkennen, dass zwei weitere nicht-sichere TONs genutzt wurden. Diese sind für das Lösen der Ausgänge nach einem Signalwechsel zuständig. Es handelt sich

dabei um eine Sicherheitsmaßnahme, so das Ausgänge nach Wertänderung immer gesteuert wieder freigegeben werden.

Nachdem nun das Sicherheitsprogramm komplettiert ist, kann die SoSafe Programmable Umgebung wieder verlassen werden. Abschließend muss eine letzte Änderung am *MotionTemplateFull* vorgenommen werden, um den Not-Halt auch auf LMC-Seite zu implementieren. Dazu sind folgende Schritte nötig:

1. Im Projektbaum öffnen der Datei *Application -> TemplateFullProgrammingFramework -> TaskCalls -> SR\_HWCopyIO*



Abbildung 35: Zuweisung der Not-Halt-Variable im LMC

Der internen Variable *G\_xEmergencyIn* wird die Not-Halt-Ausgangsvariable aus dem SLC zugewiesen. Nun bekommt auch das *MotionTemplateFull* das reale Not-Halt-Signal zur Verfügung gestellt und kann dieses verarbeiten (z. B. für den Software Not-Halt-Reset aus der Steuerungsvisualisierung).

Mit Fertigstellung der Software-Implementation der funktionalen sicherheit ist die Implementierung sämtlicher Modelle aus der Projektierungs- bzw. Modellierungsphase abgeschlossen. Anschließend gilt es die Anforderungen an das System mit Hilfe der Testkriterien zu diesen zu überprüfen, um sicherzustellen, dass die nun implementierten Funktionalitäten die Anforderungen erfüllen.

## 5.2 Verifizierung der Testspezifikation

darstellung tabellarisch für Testkriteriumsprüfung, Ablauf der Prüfungsschritte in Diagrammform

### 5.2.1 Sichtprüfung

Vorhandensein Endlagen, Not-Halt-Taster, Lichtvorhang, Bedienelemente, Signalsäule

### 5.2.2 Elektrische Prüfung

Können die Betriebsmittel eingeschalten werden, ist alles richtig verdrahtet?

### 5.2.3 Prüfung der Geräteparametrierung

läuft der sercos, können daten zwischen den PLCs versendet werden, ist der Servoregler eingerichtet, ist das Netzteil eingerichtet, funktionieren die sicheren Ein-/Ausgänge, könne Daten per OPC ausgelesen werden?

### 5.2.4 Prüfung der Programmfunctionen

Führt die Programmierung des Systems zum geplanten Verhalten?

### 5.3 Korrekturen und Verbesserungen

Getriebe x-Achse

## 5.4 Bedienungsanleitung

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

---

## **6 Zusammenfassung und Fazit**

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

---

## 7 Ausblick

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

---

# Literaturverzeichnis

## Bücher

- [Kar09] Michael Tiegelkamp Karl Heinz John. *SPS-Programmierung mit IEC 61131-3*. Springer-Verlag GmbH, Juni 2009. 402 Seiten. ISBN: 978-3-642-00269-4. URL: [https://www.ebook.de/de/product/12469251/karl\\_heinz\\_john\\_michael\\_tiegelkamp\\_sps\\_programmierung\\_mit\\_iec\\_61131\\_3.html](https://www.ebook.de/de/product/12469251/karl_heinz_john_michael_tiegelkamp_sps_programmierung_mit_iec_61131_3.html).
- [Kel19] Hubert B. Keller. *Entwicklung von Echtzeitsystemen*. Springer-Verlag GmbH, Dez. 2019. 287 Seiten. ISBN: 978-3-658-26641-7. URL: [https://www.ebook.de/de/product/38401910/hubert\\_b\\_keller\\_entwicklung\\_von\\_echtzeitsystemen.html](https://www.ebook.de/de/product/38401910/hubert_b_keller_entwicklung_von_echtzeitsystemen.html).
- [Kle13] Stephan Kleuker. *Grundkurs Software-Engineering mit UML*. Springer-Verlag GmbH, Juli 2013. 402 Seiten. ISBN: 978-3-658-00642-6. URL: [https://www.ebook.de/de/product/25604282/stephan\\_kleuker\\_grundkurs\\_software\\_engineering\\_mit.uml.html](https://www.ebook.de/de/product/25604282/stephan_kleuker_grundkurs_software_engineering_mit.uml.html).
- [Lap14] Phillip Laplante. *Requirements engineering for software and systems, second edition*. Boca Raton, FL: CRC Press, 2014. ISBN: 978-1-4665-6082-6 (siehe Seite 3).
- [Tho17] Dieter Hofmann Thomas Bindel. *Projektierung von Automatisierungsanlagen*. Springer-Verlag GmbH, 2. Aug. 2017. 273 Seiten. ISBN: 978-3-658-16416-4. URL: [https://www.ebook.de/de/product/33168410/thomas\\_bindel\\_dieter\\_hofmann\\_projektierung\\_von\\_automatisierungsanlagen.html](https://www.ebook.de/de/product/33168410/thomas_bindel_dieter_hofmann_projektierung_von_automatisierungsanlagen.html).

## Artikel

- [Gmb12] t2informatik GmbH. „Requirements Engineering“. In: (Okt. 2012). URL: <https://t2informatik.de/wissen-kompakt/requirements-engineering/> (besucht am 23. 09. 2021).

---

## Anhang

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

---

## Eidesstattliche Erklärung

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.