

Automatisierung: Chemische Reaktoranlage

MSE-Master-Projekt 2

Studiengang	Master of Science in Engineering
Autor	Spatz Yannick
Dozent	Prof. Borer Melchior
Projekt Partner	Berner Fachhochschule

Version 1.0 vom 2. August 2024

Abstract

Das Projekt konzentriert sich auf die Automatisierung einer chemischen Pharmaanlage, die aus sieben Glasreaktoren mit verschiedenen Komponenten wie Ventilen, Heiz-/Kühlsystemen und Förderpumpen besteht. Das Projekt zielt darauf ab, auf den bisherigen theoretischen und simulationsbasierten Arbeiten von Bachelorstudenten aufzubauen und diese praktisch umzusetzen. Die Anlage soll in Zukunft den Bachelorstudenten zur Verfügung stehen, um ihre entwickelte Software zu testen.

Die Hauptaufgaben des Projektes teilen sich in drei Bereiche ein:

Auf Basis der beschafften Reaktoreinheiten wurde ein neuer Aufbau geplant und umgesetzt. Dieser wurde spezifisch auf die Anforderungen der BFH ausgelegt, damit dieser für die Vertiefung der Bachelorstudenten verwendet werden kann. Innerhalb des Projektzeitraumes wurde eine neue Reaktoreinheit gebaut.

Anhand der definierten Komponenten wurde ein Schaltschrank umgesetzt. Die Feldgeräte werden hierbei mit Beckhoff-Klemmen ausgewertet und angesteuert. Jeder Reaktor ist identisch aufgebaut und besitzt die selben Komponenten.

Für den Betrieb der Anlage wurde eine Software mit TwinCat umgesetzt. Diese vereint verschiedene Aspekte wie den objektorientierten Aufbau mit einer Rezeptsteuerung. Innerhalb der Anlage gibt es eine Gesamtsystemebene und eine Reaktoreinheitsebene, welche über eine OPC-UA-Kommunikation miteinander interagieren können.

Inhaltsverzeichnis

Abstract	iii
1 Rahmen der Arbeit	1
1.1 Einleitung ins Projekt	1
1.2 Auftragsinterpretation	2
1.3 Zeitplan des Projektes	2
2 Einarbeitung in Thematik	3
2.1 Situationsanalyse	3
2.1.1 Allgemein	3
2.1.2 Mechanischer Aufbau und Komponenten	5
2.1.3 Software und Struktur	9
2.1.4 Prozess	15
2.2 System- und Kontextgrenzen des Projektes	16
2.3 Projektanforderungen	17
3 Umsetzung der Systemhardware	19
3.1 Konstruktiver Aufbau der Anlage	19
3.1.1 Schematischer Aufbau	19
3.1.2 Konstruktive Umsetzung	20
3.2 Aufbau des Schaltschranks	22
3.2.1 Planung des Schaltschranks	22
3.2.2 Umsetzung des Schaltschranks	25
4 Planung der Software	27
4.1 Entwurf einer Rezeptsteuerung	27
4.1.1 Definieren der Grundbausteine	27
4.1.2 Analyse des Prozesses	30
4.1.3 Komposition des Prozesses	32
4.1.4 Management der Leitung	36
4.2 Kommunikation über OPC-UA	39
4.2.1 Topologie der Kommunikation	40
4.3 Grundstruktur des Programms	41
4.3.1 Aufbau des Programms	41
4.3.2 Wichtige Definitionen für Funktionsbausteine	44
5 Umsetzung der Software	47
5.1 Information über aktuellen Stand	47
5.2 Arbeitsweise der Software	48
5.3 Technische Umsetzung	50
5.3.1 Status- und Control-Word	50
5.3.2 Interaktion zwischen zwei Reaktoren	52
5.3.3 Gesamtsystem-Applikation	53
5.3.4 Reaktoreinheit-Applikation	58

6 Auswertung	63
6.1 Aktueller Stand	63
6.1.1 Hardware	63
6.1.2 Software	64
6.2 Vergleich mit Anforderungen	65
6.3 Weiterführende Arbeiten	66
6.3.1 Hardware	66
6.3.2 Software	66
6.3.3 Allgemein	67
7 Schlussfolgerung / Fazit	69
8 Anhang	71
Literaturverzeichnis	75
Abbildungsverzeichnis	77
Tabellenverzeichnis	79
Glossar	81

1 Rahmen der Arbeit

1.1 Einleitung ins Projekt

Innerhalb der Vertiefungsrichtung «Industrielle Automatisierungstechnik» automatisierten Bachelorstudenten des Studiengangs «Maschinenbau» eine Pharmaanlage. Dieser Auftrag wird bis jetzt rein theoretisch umgesetzt und mittels einer Simulation getestet. Diese Pharmaanlage wurde nun beschafft. Das Ziel des Masterprojektes ist die vollständige Automatisierung und Inbetriebnahme dieser Anlage. Die Erarbeitung wird unabhängig von den Bachelorprojekten durchgeführt und soll schlussendlich auch dafür genutzt werden können. Die Pharmaanlage umfasst dabei 7 Glasreaktoren, welche je nach Konfiguration mit diversen Elementen versehen sind wie Ventilen, HK-System und/oder Förderpumpen. Für die Automatisierung wird für jeden Reaktor ein IPC von Beckhoff eingesetzt.

Die Automatisierung der Pharmaanlage umfasst folgende Arbeiten:

- ▶ Aufbau der Anlage (mechanisch)
- ▶ Verkabelung der Anlage (inkl. E-Schema / Schaltschrank etc.)
- ▶ Aufbau einer objektorientierten Programmstruktur
- ▶ Aufbau einer Rezeptursteuerung der Reaktoren
- ▶ Aufbau einer Bedien- und Beobachtebene pro Reaktor

Das Vorgehen innerhalb des Projektes unterscheidet sich vom klassischen Entwicklungsprozess (Analyse / Konzipieren / Ausarbeitung / Auswertung). Zu Beginn wird eine ausführliche Analyse der Thematik durchgeführt um alle lösungsrelevanten Fragen zu beantworten. Anhand dieser Analyse werden die diversen Kernthemen der Arbeit definiert, welche jeweils innerhalb einer Konzept- und Ausarbeitungsphase umgesetzt werden. Am Ende wird die Arbeit ausgewertet.

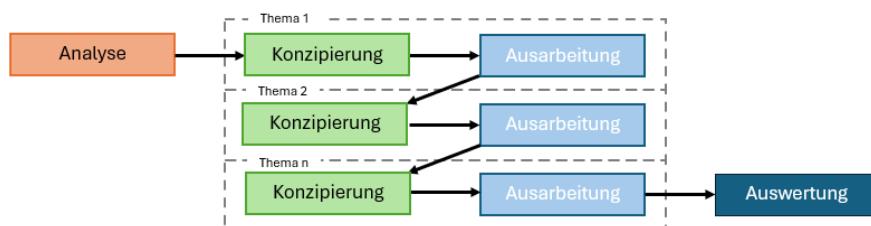


Abbildung 1.1: Grobprojektplanung

Mit der Lieferung der Anlage in der Woche vom 26.02.2024 wird das Master-Projekt 2 offiziell gestartet. Das Projekt umfasst ca. 450 Arbeitsstunden. Das Projektende wurde auf den 02.08.2024 gelegt.

1.2 Auftragsinterpretation

Projektname:	Automatisierung einer chemischen Reaktoranlage														Datum:														
Projektart:	Master-Projekt 2														28.02.2024														
Problembeschreibung:																													
<p>Innerhalb der Vertiefungsrichtung "Industrielle Automatisierungstechnik" automatisieren Bachelorstudenten des Studiengangs "Maschinentechnik" eine Pharmaanlage. Dieser Auftrag wird bis jetzt rein theoretisch umgesetzt und mittels einer Simulation getestet. Diese Pharmaanlage wurde nun beschafft.</p> <p>Das Ziel des Masterprojektes ist die vollständige Automatisierung und Inbetriebnahme dieser Anlage. Die Erarbeitung wird unabhängig von den Bachelorprojekten durchgeführt, soll schlussendlich aber auch dafür genutzt werden können.</p>																													
Projektziel:																													
<ul style="list-style-type: none"> - Montage der Anlage (inkl. Konstruktion eines passenden Gerüstes) - Entwicklung der Ablauflogik (inkl. Rezeptursteuerung) - Aufbau der Netzwerkstruktur mittels OPC UA - Aufbau einer geeigneten Bedien- und Beobachtungsebene für jeden Reaktor 																													
Projektbegrenzung, Richtlinien, Rahmenbedingungen																													
<ul style="list-style-type: none"> • Die Anlage soll den Bachelorstudenten für zukünftige Arbeiten dienen • Alle Versuche werden mit Wasser durchgeführt 																													
Projektorganisation:	<u>Name:</u>	<u>Status:</u>	<u>E-Mail:</u>																										
Betreuer:	Borer Melchior	Dozent BFH	melchior.borer@bfh.ch																										
Co-Betreuer:	/	/	/																										
Erarbeitet durch:	Spatz Yannick	Student	yannick.spatz@bfh.ch																										
Meilensteine:	<u>Soll:</u>															<u>Ist:</u>													
Start:	26.02.2024															26.02.2024													
Ende:	02.08.2024															02.08.2024													
Relevante Termine:	Siehe Zeitplan																												

Tabelle 1.1: Auftragsinterpretation

1.3 Zeitplan des Projektes

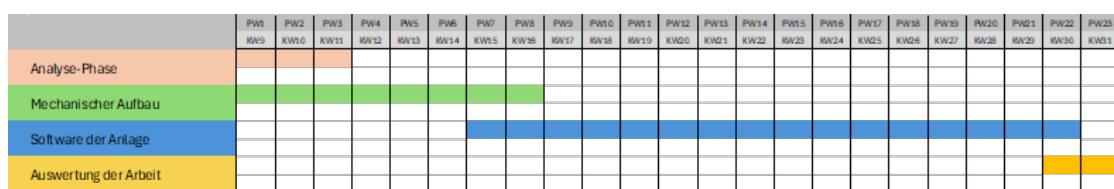


Abbildung 1.2: Grober Zeitplan



Ein detaillierter Zeitplan wird im Anhang beigelegt.

2 Einarbeitung in Thematik

2.1 Situationsanalyse

Die Situationsanalyse spielt bei diesem Projekt eine zentrale Rolle und ist massgeblich für die weitere Planung des Projektes relevant. Die Umsetzung der Anlage weicht vom klassischen Entwicklungsprozess stark ab und setzt auf eine dynamischere Projektstruktur. Viele Aspekte der Erarbeitung sind zum jetzigen Stand nicht definiert. Um den Rahmen der Arbeit trotzdem einschätzen zu können, müssen diverse theoretische Grundlagen innerhalb der Situationsanalyse erarbeitet werden. Aus diesem Grund wird der detaillierte Zeitplanung erst nach der Situationsanalyse erstellt.

2.1.1 Allgemein

Frage 1.1: Wie sieht das momentane Layout der Anlage aus?

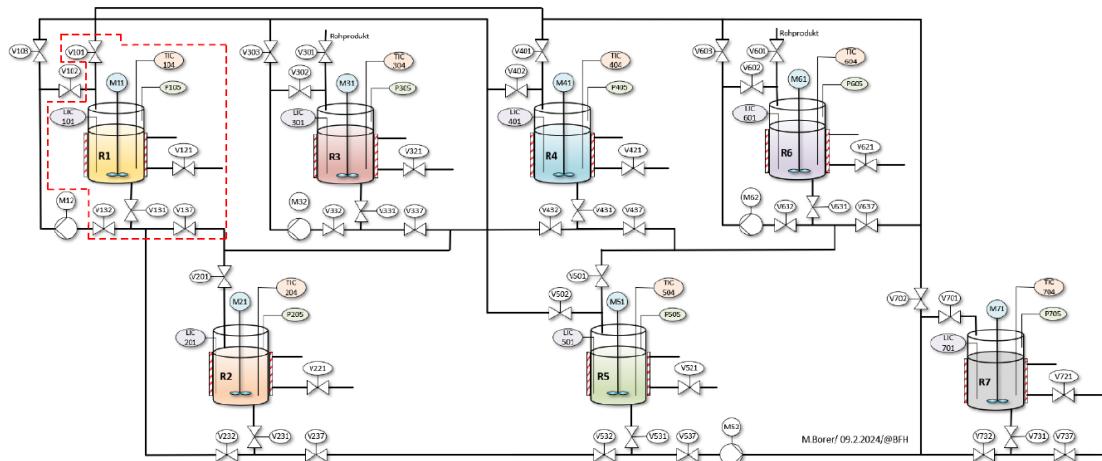


Abbildung 2.1: Startsystemübersicht

Ursprünglich wurde die Idee einer automatisierten Pharmaanlage für ein Vertiefungsprojekt der Bachelorstudenten definiert. Dabei wurde von den Studenten die Software (Ablauflogik) für ein solches System entwickelt. Jeder Reaktor hatte dabei spezifische Aufgaben, welche umgesetzt werden sollten. Diese Idee wird nun weiterverfolgt und weiterentwickelt. Das Layout der Anlage basiert jedoch auf der ursprünglichen Aufgabenstellung für das Bachelorprojekt. Im Laufe des nun durchgeführten Projektes, kann dieses Layout noch verändert und optimiert werden. Das Layout besteht aus 7 Reaktoren. Jeder Reaktor hat einen Grundaufbau, welcher identisch ist (rote Markierung). Dieser besteht aus den Reaktorbehälter, einem Rührsystem, einer Heizung, 4 Ventilen (1 Einlass / 1 Auslass und 2 Verteiler), einer Füllstandmessung, einem Drucksensor und einer Temperaturmessung. Je nach Konfiguration des Reaktors gibt es zusätzliche Ventile oder Förderpumpen. Auf die Verbindungen zwischen den Reaktoren wird unter Kapitel 2.1.4 eingegangen.

Frage 1.2: Wo wird die Anlage betrieben?

Die Pharmaanlage wird in der Maschinenhalle der BFH in Burgdorf betrieben. Die Halle bietet genügend Platz und stellt alle nötigen Anschlussmöglichkeiten für den Betrieb zur Verfügung.

Frage 1.3: Wer soll die Anlage bedienen können?

Die Anlage wird durch Mitarbeiter und Studenten der BFH betrieben. Es gibt keine kritischen oder gefährlichen Aspekte, welche die Regulierung der Bedienung der Anlage zur Folge hätte. Die Bachelorstudenten erarbeiten innerhalb eines Projektes eine Software, welche mit der Anlage getestet werden soll. Dafür müssen die Studenten in der Lage sein, mit dem System zu interagieren.

Frage 1.4: Welche Schnittstellen soll die Anlage besitzen

Zum Stand der Analyse besitzt die Pharmaanlage zwei Input-Schnittstellen für das Rohprodukt. Damit können Reaktor 3 und 6 befüllt werden. Das System besitzt ausserdem einen Auslass für das prozessierte Produkt. Als Kommunikationsschnittstelle wird eine Ethernetverbindung verwendet. Wie genau die Topologie für die Kommunikation aussieht, wird noch definiert. Das System benötigt auch eine Schnittstelle für die Leistungsversorgung, auch dies bezüglich wird die konkrete Umsetzung zu einem späteren Zeitpunkt definieren.

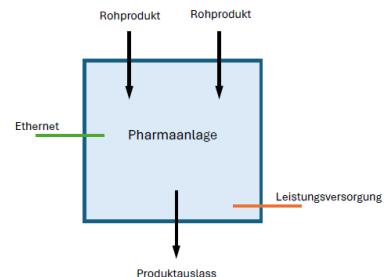


Abbildung 2.2: Übersicht über Schnittstellen

Frage 1.5: Gibt es sicherheitsrelevante Aspekte

Bei der Anlage handelt es sich um ein Laborgerät mit Prototypenstatus, entsprechend können industrielle Sicherheitsaspekte vernachlässigt werden. Folgende Punkte müssen trotzdem berücksichtigt werden:

- ▶ Das Risiko eines Stromschlags bei 230 V muss bei den Schaltschränken minimiert werden
- ▶ Verbrennungen durch das HK-System müssen verhindert werden (Bezug zu Fähigkeit von HK)
- ▶ Die Gefahr durch Quetschungen, Einklemmen oder Kollisionen durch das Rührsystem oder die Förderpumpe muss minimiert werden (beide Systeme besitzen einen Motor).

Da für den Betrieb der Anlage nur Wasser verwendet wird, werden keine zusätzlichen Sicherheitsvorkehrungen bezüglich des Mediums benötigt.

2.1.2 Mechanischer Aufbau und Komponenten

Frage 2.1: Welche Komponenten wurden bereits beschafft?

Als Basis für die komplette Anlage wurden 7 Reaktorsysteme beschafft. Diese Systeme beinhalten die Hauptkomponenten, welche für die Anlage benötigt werden. Die gelieferten Reaktorsystemkomponenten werden für die Konstruktion der neu entwickelten Anlage verwendet. Die wichtigsten Komponenten sind:

Bezeichnung	Typ	Eigenschaften	Anzahl
Glass-Reaktor	S212-5L	Volumen: 5L	1x7 Stück
Drucksensor	/	Max Druck: 1 bar	1x7 Stück
Motoreinheit für Rührer	5IK120GU-SFT		1x7 Stück
Magnetventile	2W-15 (DC24V)		54 Stück
Temperatursensor	PT100 (3 Leiter)		1x7 Stück
Füllstandsensor	UHZZP-10		1x7 Stück
HK-System	GDX-5/30		3 Stück
Schlauchquetschpumpe	BT600J-1A		4 Stück

Tabelle 2.1: Gelieferte Grundkomponenten

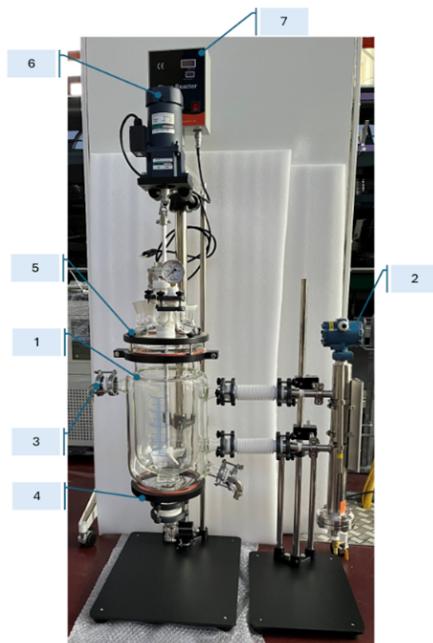


Abbildung 2.3: Basis der Reaktoreinheit

In einem ersten Schritt wird eine der sieben gelieferten Reaktoreinheiten in der ursprünglichen Konfiguration aufgebaut. Alle sieben Systeme beinhalten die gleichen Komponenten und werden entsprechend gleich aufgebaut. Das Herzstück der Reaktoreinheit ist der Reaktorbehälter (Punkt 1). Der doppelwandige Glasbehälter kann ein Fassungsvolumen von 5 l aufnehmen. Der Behälterkörper hat 4 seitliche Anschlussmöglichkeiten. Zwei Anschlüsse sind für die Füllstandmessung vorgesehen (Punkt 2). Hierbei wird mechanisch über einen Schwimmer der Füllstand angegeben. Außerdem wird der Füllstand auch induktiv über einen Sensor ermittelt. Der Stand wird mit einem 4 – 20 mA Signal angegeben. Das Messsystem wird mechanisch über flexible Adapterstücke mit dem Reaktorkörper verbunden. Die Möglichkeit muss abgeklärt werden, ob der Sensor auch direkt im Glasbehälter positioniert werden kann. Die zwei weiteren Anschlüsse sind für das HK-System vorgesehen (Punkt 3). Via Adaptergewindeverbindungen kann das System angeschlossen werden. Das HK-System bietet die Möglichkeit Öl auf ei-

ne bestimme Temperatur zu erhitzen oder abzukühlen. Drei dieser Systeme stehen für die Anlage zur Verfügung. An der Unterseite des Reaktors wird ein Magnetventil angeschlossen (Punkt 4). Mit dem Ventil kann der Inhalt des Reaktor ausgelassen werden. Auf der Oberseite wird über ein Befestigungsschnittstelle ein Deckel montiert (Punkt 5).

Der Deckel bildet mit sechs Öffnungen die Schnittstelle zu diversen Elementen. Dazu gehören im Moment die Druckmessung, das Magnetventil für den Einlass und der Rührer. Vier Öffnungen sind nicht besetzt und könnten für eine alternative Füllstandsmessung oder zusätzliche Magnetventile genutzt werden. Die Schnittstelle zwischen Reaktor und Deckel wurde ohne Dichtung geliefert. In einem ersten Schritt wird an dieser Stelle auch keine Dichtung vorgesehen. Nach Bedarf kann eine Flachdichtung ergänzt werden. Im oberen Bereich des Reaktorsystems befindet sich der Motor [6] und der FU [7]. Beim Motor handelt es sich um einen AC-Wechselstrommotor mit Getriebe, welcher mit 230 V betrieben wird. Über den FU mit integriertem Potentiometer kann der Motor bewegt werden. Der FU besitzt jedoch keine 0 – 10 V ansteuerbare Schnittstelle für die Integration mit Beckhoff. Hier muss eine Alternative gefunden werden. Der FU bietet auch die Möglichkeit einen PT100-Temperatursensor anzuschliessen. Geliefert wurden ausserdem vier Schlauchquetschpumpen. Diese sind für die Beförderung des Mediums verantwortlich. Die Pumpe wird mit einem Kunststoffschlauch betrieben. Bedient wird diese entweder über die vorhandene Bedienoberfläche oder über eine «Standard External Control (4-20 mA)» Schnittstelle, welche die Integration in TwinCat ermöglicht. Die Möglichkeiten dieser Schnittstelle müssen noch abgeklärt werden.

Für den Aufbau der Steuerung wurden im Vorfeld diverse Komponenten definiert und bestellt:

Bezeichnung	Model	Anzahl
Beckhoff Ultra-Kompakt-Industrie-PC (SPS)	C6015-0020	7 Stück
Beckhoff Buskoppel (8 Eingänge / 4 Ausgänge)	EK1818	7 Stück
Beckhoff 8-Kanal-Ausgangskarte (Digital)	EL2008	7 Stück
Beckhoff 2-Kanal-Ausgangskarte (Analog)	EL4012	7 Stück
Beckhoff 4-Kanal-Eingangskarte (Analog)	EL3014	7 Stück
Ethernet Switch mit 8 RJ45-Ports	SW-508	7 Stück

Tabelle 2.2: Definierte Steuerungskomponenten

Da die Reaktoreinheiten geliefert wurden und die Elemente nun bekannt sind, kann es bei den Komponenten der Steuerung zu Anpassungen oder Ergänzungen kommen. Dies wird jedoch nicht innerhalb der Analyse definiert, sondern bei der Planung des Schaltschrankes. Auf die genauen Spezifikationen aller Komponenten wird nicht eingegangen.



Entsprechende Datenblätter werden im Anhang beigelegt.

Frage 2.2: Wie ist das eingekaufte Motorsteuerungssystem aufgebaut?

Betrachtet man die verschiedenen In- und Outputs der Motorenanschlussbox, so gibt es diverse Möglichkeiten. Auf der Unterseite befindet sich eine Schnittstelle für 230 V und eine 3-Leiter-Schnittstelle für den Temperatursensor (PT100). Als zusätzlicher Input befin-

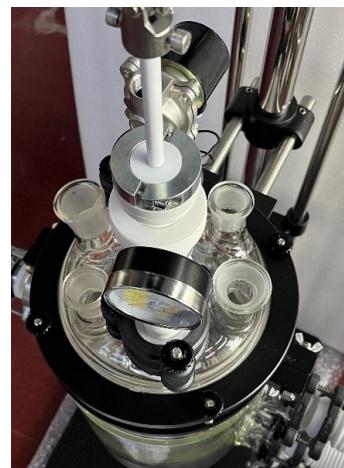


Abbildung 2.4: Reaktordeckel der Basis

det sich auf der Vorderseite ein Potentiometer. Für den Motoroutput gibt es eine 3-Leiter-Schnittstelle. Außerdem werden die Drehgeschwindigkeit (in RPM) und die Temperatur (in °C) über Anzeigen ausgegeben. Im Innern der Box befindet sich der FU und ein Temperaturkontroller. Beides wird über 230V gespiesen. Der Temperaturkontroller nimmt die drei Anschlüsse des PT100's auf und besitzt eine Schnittstelle zur entsprechenden Anzeige auf der Aussenseite der Box. Der FU besitzt zwei Ebenen. Die obere Ebene stellt die Schnittstelle zum Potentiometer und zur Anzeige dar. An der unteren Ebene wird der Motor und die Speisung angeschlossen. Im Moment wird nicht weiter auf die verschiedenen Komponenten und deren Spezifikationen eingegangen.



Abbildung 2.5: FU-Kontrollboard

Frage 2.3: Wie können bestehenden Komponenten in die Steuerung eingebunden werden?

Temperatursensor:

Da es sich mein Temperatursensor um einen Standard-PT100-Sensor mit 3 Leitern handelt, kann dieser über eine entsprechende Beckhoff-Karte (EL3202) eingelesen werden.

Schlauchquetschpumpe:

Die Schlauchquetschpumpe kann über eine «Standard External Control» Schnittstelle angesprochen werden.

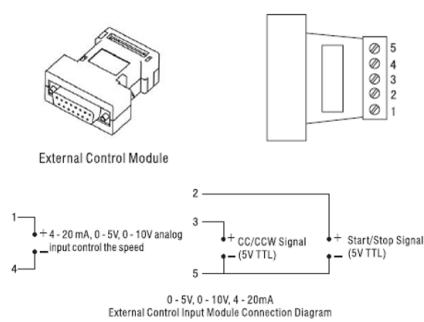


Abbildung 2.6: Standard External Controll Schnittstelle

Der Stecker hat fünf Anschlüsse. Anschluss 1 und 4 sind für das 4 - 20 mA Signal. Damit kann die Geschwindigkeit der Pumpe vorgegeben werden. Dafür wird die Beckhoff-Karte EL4012 verwendet. Mit Anschluss 2 kann die Pumpe gestoppt/gestartet werden und mit Anschluss 2 wird die Drehrichtung definiert. Diese zwei Anschlüsse benötigen eine Spannung zwischen 5 und 10 V.

Antriebsmotor:

Die Integration des Antriebsmotors kann über das bestehende FU-Element realisiert werden. Im Moment wird der Motor über ein Potentiometer gesteuert, welches eine Spannung von 0 - 10 V ausgibt. Entsprechend kann der Motor über eine Beckhoff-Karte (z.B. EL4102) gesteuert werden.

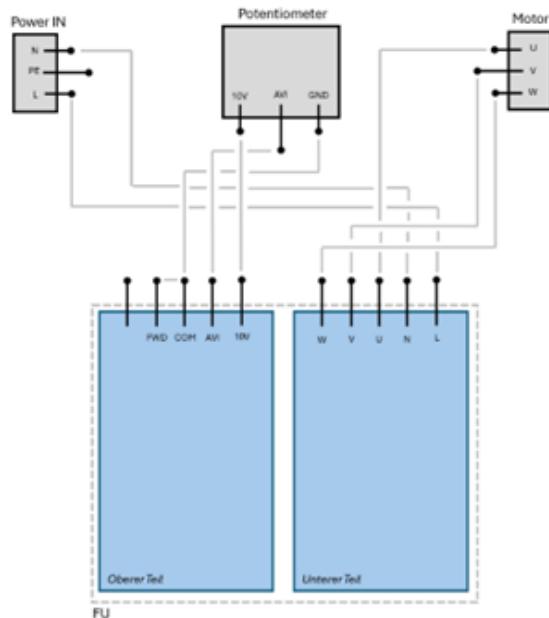


Abbildung 2.7: Anschluss des Antriebsmotors



Bei Versuchen hat sich gezeigt, dass die externe Geschwindigkeitsanzeige (LCD-Anzeige) mit dem FU-Element verbunden sein muss, damit der Motor betrieben werden kann.

HK-System:

Das HK-System besitzt keine direkte Schnittstelle für die Interaktion mit einer Beckhoff-Klemme.

Hinter dem Frontpanel befindet sich eine Leiterplatine mit diversen Anschlüssen. Diese steuert die Prozesse innerhalb des Systems und wird durch die Buttons auf dem Frontpanel bedient. Die Anschlussklemmen auf der Oberseite der Platine sind wahrscheinlich mit den diversen Komponenten des Systems verbunden und sind direkt für den Betrieb dieser zuständig. Aus der Dokumentation des HK-Systems kann die Funktionsweise der Anschlüsse nicht definiert werden.

Das System muss in Betrieb genommen werden, um die Signale an der Klemmen auswerten zu können. Damit sollte es möglich sein, eine Aussage über die Interaktionsmöglichkeiten mit dem System zu machen. Der Aufwand einer solchen Analyse ist im Moment schwer einzuschätzen.

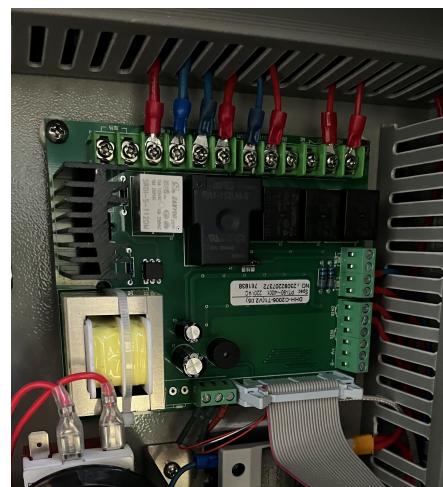


Abbildung 2.8: Anschlüsse des HK-Systems

2.1.3 Software und Struktur

Frage 3.1: Was ist die Struktur eines automatisierten Systems?

Ein komplett automatisiertes System kann in unterschiedliche Levels eingeteilt werden. Man spricht hierbei auch von der Automationspyramide [1].

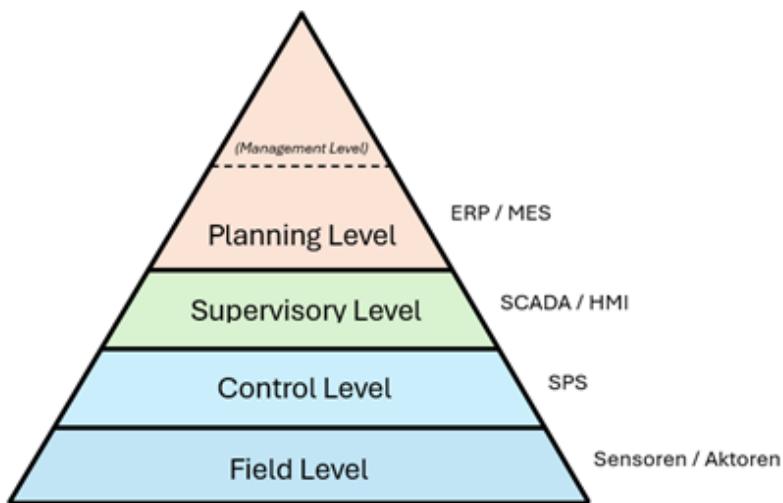


Abbildung 2.9: AutomatisierungsPyramide

Field Level:

Bildet die unterste Ebene der Pyramide. Diese Ebene umfasst die verschiedenen Sensoren und Aktoren (Feldgeräte), welche für den Betrieb der Anlage eingesetzt werden. Zusätzlich werden diverse Prozessdaten erfasst.

Control Level:

Auf dieser Ebene befindet sich die SPS, welche für die Steuerung des jeweiligen Prozesses zuständig ist. Hierbei wird direkt mit den Sensoren und Aktoren interagiert.

Supervisory Level:

Das Ziel dieser Ebene ist die Verwaltung und Steuerung der verschiedenen Elemente der Anlage. Man spricht dabei vom SCADA-System (Supervisory Control And Data Acquisition). Das SCADA-System überwacht und steuert die Produktionsabläufe der Anlage. Dabei kann von einem Standpunkt, kann PC oder IPC sein, auf Daten und Elemente im «Control Level» zugegriffen werden. Das HMI (Human-Machine-Interface) ist ein wichtiger Bestandteil des SCADA-Systems. Es ermöglicht die Interaktion mit dem System durch einen Bediener und die Darstellung von Informationen.

Planning Level:

Die oberste Ebene ist für den Gesamtüberblick des Prozesse und ermöglicht Entscheidungen zur Optimierung des Prozesses zu treffen. Ein wichtiges System ist das MES (Manufacturing Execution System). Das MES überwacht alle Prozesse innerhalb einer Anlage oder einer Fabrik. Dies erlaubt dem Management einen Überblick über den aktuellen Stand zu erhalten und entsprechend Entscheidungen über zukünftige Prozesse zu treffen. Die Spitze der Pyramide bezeichnet man als Management Level. Auf diese Ebene wird jedoch nicht im Detail eingegangen. Hier werden strategische Entscheidungen auf Unternehmungsführungsebene getroffen. Ein wichtiges Tool dabei ist das ERP (Enterprise Resource Planning).

Frage 3.2: Wie sieht die Struktur der Anlage aus (bezogen auf die Pyramide)?

Grundsätzlich bewegt sich die chemische Pharmaanlage in den ersten 3 Ebenen.

Auf dem «Field Level» befinden sich die sieben Reaktoren mit den entsprechenden Aktoren und Sensoren. Das «Control Level» umfasst die jeweilige SPS zum Reaktor. Im «Supervisory Level» befindet sich eine SPS oder ein PC, welcher zur Bedienung, Datenerfassung und Datenanzeige genutzt werden kann. Dazu gehört eine entsprechende HMI-Umsetzung.

Das Ziel ist, dass die Anlage mit einer Rezeptursteuerung betrieben wird. Eine Rezeptur stellt eine detaillierte Anweisung zur Herstellung eines Produktes dar. Die Planung der Rezepte wird auf dem «Planning Level» durchgeführt, während die eigentliche Ausführung und Koordination im «Supervisory Level» stattfindet.

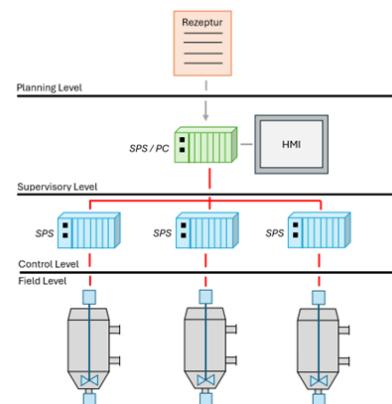


Abbildung 2.10: Ebenen der Anlage

Frage 3.3: Was ist eine Rezeptursteuerung?

Ein Rezept beschreibt den Herstellungsweg einer Produktcharge mit der Minimalmenge an dafür benötigten Informationen. Beim Rezept geht es nicht um eine Dispositionsplanning, der Fokus liegt auf prozessbezogene Informationen. Dadurch wird der Prozess der Anlage nicht durch eine Ablauflogik definiert, sondern durch die im Rezept definierten Schritte und Parameter. Dies ermöglicht ein flexibles Einsetzen der Anlage und deren Ressourcen. Sofern es die Infrastruktur der Anlage zulässt, kann damit jedes Rezept gefahren werden. Dies reduziert Stillstandszeiten der Anlage und macht diese effizienter. Damit eine Rezeptursteuerung angewendet werden kann, braucht die Software eine vorgegebene Struktur [2].

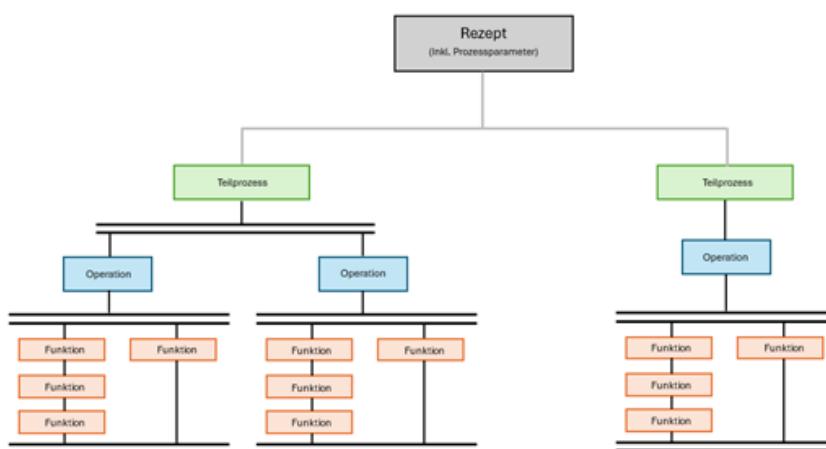


Abbildung 2.11: Struktur einer Rezeptursteuerung

Ein Rezept setzt sich oft aus mehreren Schritten zusammen, die verschiedene Teilprozesse umfassen können. Ein Beispiel hierfür ist die Herstellung von Lösungen A und B, die in einem nachfolgenden Schritt gemischt und weiterverarbeitet werden. Jeder dieser Teilprozesse besteht aus einem oder mehreren Grundoperationen. Bei der Nutzung eines Reaktors könnte eine solche Grundoperation das Befüllen sein. Innerhalb dieser Operation müssen verschiedene Funktionen entweder parallel oder sequenziell ablaufen. Beim Befüllen beinhaltet dies das Öffnen der Ventile, das Erfassen des Füllstands und das anschliessen-

de Schliessen der Ventile. Die unterste Ebene der Struktur repräsentiert die Implementierung der physischen Komponenten der Anlage als Funktionsbausteine in der Software. Für die Implementierung der Rezeptursteuerung für die Pharmaanlage wird in einem ersten Schritt die Teilprozessebene vernachlässigt. Wie genau die Rezeptursteuerung umgesetzt wird, wird nicht innerhalb der Analyse erarbeitet.



Innerhalb der Dokumentation wird von Rezept und Rezeptur gesprochen. Damit ist aber immer ein Rezept gemeint. Bei einer richtigen Rezeptur würde das System automatisch den Reaktor definieren, welcher einen Prozess durchführt. Dies wird in einem ersten Schritt nicht umgesetzt. Im Moment wird der Reaktor durch das Rezept vorgegeben.

Die Norm ISA-88 (EN 61512) beschreibt in 4 Teilen die chargenorientierte Prozessierung und gibt dafür verschiedene Design-Philosophien vor. Innerhalb dieser Analyse wird nicht weiter auf diese Normen eingegangen. Bei der Erarbeitung der konkreten Rezeptursteuerung wird sich gegebenenfalls auf die Norm bezogen.



Die zur Verfügung stehenden Normteile werden im Anhang beigelegt.

Frage 3.4: Welche möglichen Topologien können in Frage kommen?

Die Topologie beschreibt das Anordnen und Verbinden der vorhandenen Systemkomponenten über ein Ethernet-Netzwerk. Für die chemische Reaktoranlage wird mit einer Beckhoffsteuerung gearbeitet. Die Kommunikation findet dadurch über EtherCAT statt. Innerhalb dieser Arbeit wird nicht weiter auf die Funktionsweise von EtherCAT eingegangen. Für dieses System sind Anforderungen wie die «Update Time» und die «Line Depth» nicht relevant, da kein zeitkritischer Prozess durchgeführt wird. Der wichtiges Aspekt der Topologie für das System ist der modulare Aufbau. Die Netzwerkstruktur soll so ausgelegt werden, dass Reaktorsysteme entfernt oder hinzugefügt werden können. Dafür eignet sich eine Stern-Topologie gut. Dabei gibt es ein zentrales Gerät, mit welchem alle weiteren Elemente verbunden sind. Die genaue Topologie-Struktur wird erst zu einem späteren Zeitpunkt erarbeitet. Als interessante Funktion bietet Beckhoff eine Hot-Connect-Funktionalität an [3]. Dabei ist es möglich, während dem Betrieb definierte Abschnitte aus dem System zu entnehmen oder hinzuzufügen. Dies kommt zum Beispiel bei Fertigungsanlagen zum Einsatz, bei welchen der Werkzeughalter (mit integriertem EtherCAT-IO) gewechselt werden kann. Es stellt sich jedoch die Frage, welchen Nutzen eine solche Funktionalität für die chemische Reaktoranlage bringen würde. Voraussichtlich werden einzelne Reaktorsysteme nicht während dem Betrieb vom Gesamtsystem getrennt oder hinzugefügt. Bei Bedarf wird die Hot-Connect-Thematik jedoch weiter vertieft.

Frage 3.5: Welche Möglichkeiten für die Umsetzung der HMI gibt es?

Das HMI bildet die Schnittstelle nach aussen und entsprechend zum Bediener. Die Hauptaufgabe ist Darstellung von system- und prozessrelevanten Informationen und Daten, wie auch die Kommunikation mit der Anlage durch den Bediener. Das HMI ist meistens Bestandteil des SCADA-Systems. Das SCADA-System ist für die Erfassung der Daten und Steuerung der Anlage verantwortlich. Das HMI ist dabei die Interaktionsschnittstelle mit dem SCADA-System. Die HMI kann sehr einfach über Buttons und spezifische Anzeigen für Daten umgesetzt werden oder über ein detailliertes Benutzerinterface auf einem Bildschirm mit Touch-Funktionen. Idealerweise kann von überall auf das HMI zugegriffen werden. Für die chemische Reaktoranlage gibt es diverse Möglichkeiten ein HMI aufzubauen.

Verwendung von Drittanbieter-Software:

Auf dem Markt werden verschiedenen HMI-Systeme angeboten. Als interessante Möglichkeit bietet sich die «Zenon» Softwareplattform von Copadata an [4]. Das Software-Paket bietet SCADA/HMI-Funktionen und ermöglicht auch die Umsetzung einer Rezeptursteuerung.

Umsetzung mit einfacher TwinCat-Visualisierung:

TwinCat bietet die Möglichkeit einfache Benutzeroberflächen zu erstellen. Diese fallen jedoch sehr rudimentär aus und je nach geplanten Funktionen der HMI müssen Kompromisse eingegangen werden. Auch die Möglichkeit über Remote auf das HMI zuzugreifen ist mit dieser Umsetzung nicht gewährleistet.

Verwenden von entsprechendem Beckhoff-HMI-Tool:

Beckhoff bietet mit dem TE2000-Paket eine umfassende und direkt in TwinCat integrierte Lösung für die Umsetzung eines HMI's. TE2000 bietet dabei die Möglichkeit, die HMI webbasiert aufzubauen. Auf die HMI kann somit von einem Browser, mit der entsprechenden Adresse, zugegriffen werden [5]. Das TE2000-Paket steht der BFH zur Verfügung und könnte für die Anlage eingesetzt werden.

Programmieren von eigenem HMI:

Grundsätzlich wäre es auch möglich via C# und XML eine eigene HMI-Applikation aufzubauen. Dies könnte auch das SCADA-System miteinbeziehen. Das SCADA/HMI-Tool könnte dadurch spezifisch auf die Anwendung ausgelegt werden. Dies umfasst z.B. eine Erfassung der Daten in einer SQL-Datenbank. Der Aufwand eines solchen Tools würde wahrscheinlich den Rahmen des Projektes sprengen.

Im Verlauf des Projektes werden die Anforderungen an das SCADA/HMI-System definiert. Anhand dieser Anforderungen kann die Umsetzung bestimmt werden.

Frage 3.6: Was ist eine objektorientierte Struktur?

Bei einem objektorientiertem Programmaufbau werden Sensoren und Aktoren, so nahe an der Realität abgebildet wie möglich. Ein objektorientierter Programmaufbau ermöglicht das einfache und übersichtliche Umsetzen von Software für komplexe Anlagen. Die verschiedenen Typen von Elementen, wie z.B. ein Motor, werden als Klasse angelegt. Im TwinCat geschieht das über Funktionsbausteine. Diese Klassen werden für alle Elemente der Anlage verwendet, um das entsprechende Objekt zu instanziiieren. Durch diesen Aufbau wird der Programmieraufwand minimal gehalten, da die Logik eines Elementes nur einmal erstellt werden muss (in der Klasse) und anschließend beliebig oft instanziiert werden kann [6].

Die Eigenschaften der Klassen ermöglichen Variablen von aussen zu beschreiben oder zu lesen. Eine Klasse kann aber trotzdem Input- und Output-Variablen besitzen, welche nicht als Eigenschaft definiert wurden, z.B. für die Verriegelung eines Motors. Falls mehrere Klassen dieselben Eigenschaften und

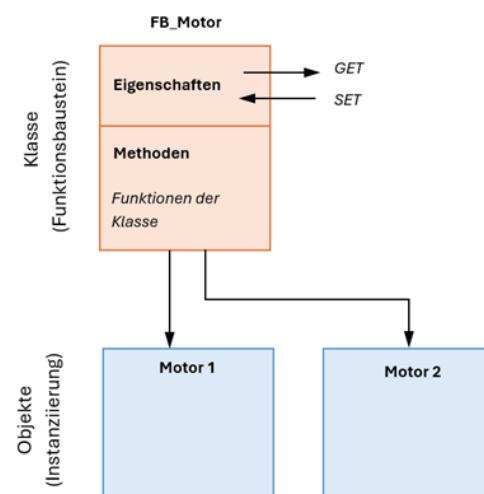


Abbildung 2.12: Objektorientierte Struktur

Methoden benötigen, kann dies über ein Interface gewährleistet werden. Ein Interface ist eine Vorgabe, welche Methoden oder Eigenschaften eine Klasse haben muss. Als konkretes Beispiel wird die mögliche Umsetzung einer Motorenklasse beschrieben:

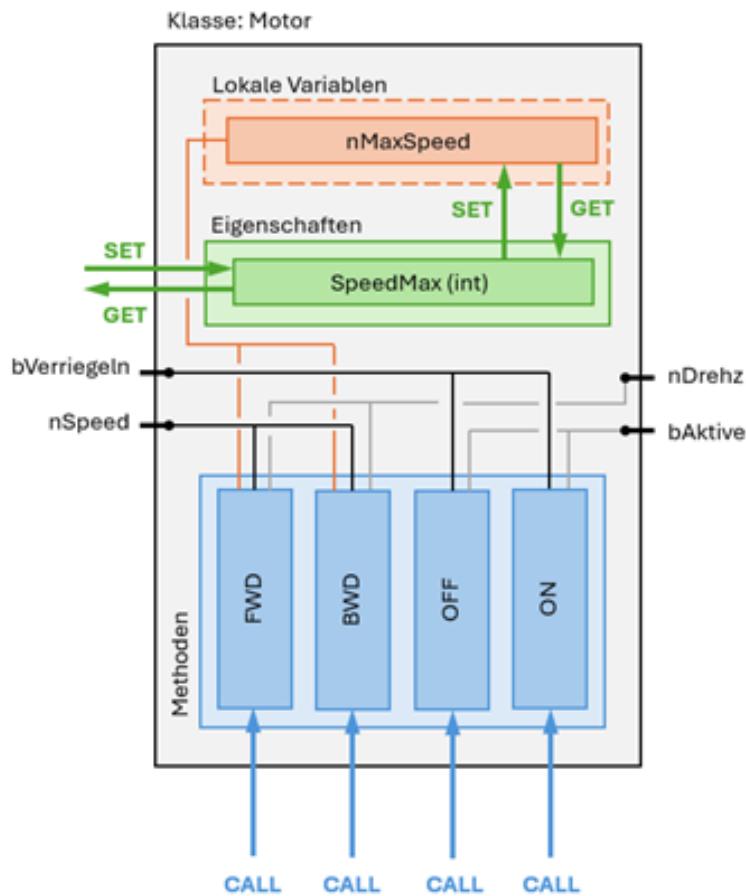


Abbildung 2.13: Beispiel einer Motorklasse

Die Klasse besitzt zwei Inputs und Outputs. Die Inputs definieren, ob der Motor verriegelt ist und wie hoch die Geschwindigkeit sein soll. Die Outputs geben die Drehzahl und den Zustand an. Es gibt vier Methoden, mit welchen die Funktionen des Motors definiert sind. Der Motor kann ein- und ausgeschalten werden. Außerdem kann der Motor vorwärts oder rückwärts drehen. Die Methoden greifen auf die jeweiligen Inputs zu, um die entsprechenden Outputs zu setzen. Über eine Eigenschaft kann die maximale Geschwindigkeit definiert werden. Der Wert kann über «SET» geschrieben und über «GET» ausgelesen werden. Innerhalb der Eigenschaft wird eine lokale Klassenvariable beschrieben oder gelesen. Die Methoden arbeiten mit dieser lokalen Variable. Der Unterschied zwischen Input-Variablen und Eigenschaften liegt darin, dass Input-Variablen für den Betrieb des Motors benötigt werden. Der Motor kann ohne eine Geschwindigkeitsangabe nicht bewegt werden. Das Gleiche gilt für die Ausgänge. Die definierten Ausgänge werden verwendet, um den Motor zu betreiben und sind dadurch direkt mit der Hardware verbunden. Die Eigenschaft ist für spezifische Parameter des Motors. Unterschiedliche Motortypen, welche mit der Klasse instanziert werden, haben z.B. unterschiedliche maximalen Geschwindigkeiten.

Frage 3.7: Was ist OPC UA?

OPC UA ist eine der im Moment wichtigsten Kommunikationsprotokolle für Industrieanwendungen. Basierend auf TCP/IP-Technologie ermöglicht OPC UA eine vertikale und horizontale Kommunikation. Die Kommunikation baut dabei auf 4 Grundprinzipien auf [7]:

- ▶ Bereitstellung einfacher Schnittstellen
- ▶ Bereitstellung eines einheitlichen Nachrichtenformats
- ▶ Bereitstellung flexibler Erweiterungsmöglichkeiten
- ▶ Implementierung hoher Sicherheitsstandards und verschiedener Sicherheitslevel

Die Kommunikation kann mit einem Server-Client-Model oder einem Publisher-Subscriber-Model umgesetzt werden. Für die chemische Reaktoranlage wird auf eine Server-Client-Kommunikation gesetzt. Der Server bildet die Basis für die Kommunikation und stellt die OPC-Schnittstellen gegen aussen bereit. Die entsprechenden Daten werden durch den Server gesammelt und bereitgestellt. Der Client bildet das Gegenstück zum Server, indem er sich mit dem Server verbinden und die Daten auslesen kann. Dabei kann jeder Client auf jeden beliebigen OPC-Server zugreifen. [8].

Der grosse Vorteil von OPC UA ist, dass durch die einheitliche Schnittstellenstruktur unterschiedliche Geräte miteinander kommunizieren können.

Frage 3.8: Wie kann OPC UA mit TwinCat realisiert werden?

Beckhoff bietet das Software-Paket TF6100 für TwinCat an. Damit lassen sich OPC UA Server, Clients und Gateways aufbauen. Eine detaillierte Beschreibung bezüglich Aufbau und Verwendung kann im entsprechenden Handbuch gefunden werden. Innerhalb der Analyse wird TF6100 nicht genauer angeschaut. Zum entsprechenden Zeitpunkt in der Erarbeitung werden die relevanten Aspekte detaillierter erläutert.

2.1.4 Prozess

Frage 4.1: Wie werden die jeweiligen Reaktoren gefüllt (Stand: Aufgabenstellung)?

Bezogen auf das Reaktor-Layout aus dem Auftrag gibt es zwei Möglichkeiten, wie Reaktoren gefüllt werden können. Ein Reaktor kann durch das Gefälle zwischen zwei Reaktoren gefüllt werden oder durch eine Pumpe. In der folgenden Matrix wird dargestellt, wie die jeweiligen Reaktoren gefüllt werden können.

R	1	2	3	4	5	6	7
1	2	1	2	2	2	/	/
2	2	2	/	2	2	2	2
3	2	1	2	2	1	/	/
4	2	1	2	2	1	/	/
5	2	2	2	2	2	2	2
6	2	/	/	2	1	2	1
7	/	/	/	/	/	/	/

1: Füllung durch Gefälle

2: Füllung durch Pumpe

Tabelle 2.3: Reaktorkonfiguration

Welchen Nutzen diese Matrix für die Umsetzung der Software haben wird, kann im Moment nicht gesagt werden.

Frage 4.2: Aufteilung des Reaktorlayouts (Stand: Aufgabenstellung)?

Es ist sinnvoll das Layout in Bereiche einzuteilen. Konkret bedeutet dies, dass die Leitungen in Bereiche unterteilt werden, welche durch Ventile abgetrennt sind. Die entsprechenden Ventile sind diesen Bereichen zugeordnet. Wichtig ist zu wissen, welche Leitungen verwendet werden, um von einem Reaktor zum anderen zu kommen. Daraus lässt sich dann definieren, welche Ventile verriegelt werden müssen.

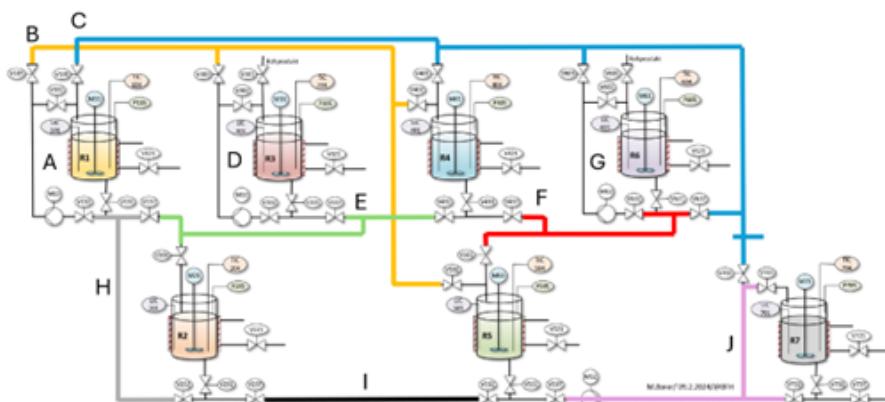


Abbildung 2.14: Reaktorlayout aufgeteilt

2.2 System- und Kontextgrenzen des Projektes

Über die System- und Kontextgrenzen wird der Rahmen für das Projekt gesetzt. Man unterscheidet zwischen System (*kann beeinflusst werden*), Systemkontext (*ist relevant für das Projekt, jedoch können solche Aspekte nicht beeinflusst werden*) und irrelevanter Umgebung (*spielt zum jetzigen Zeitpunkt keine Rolle*).

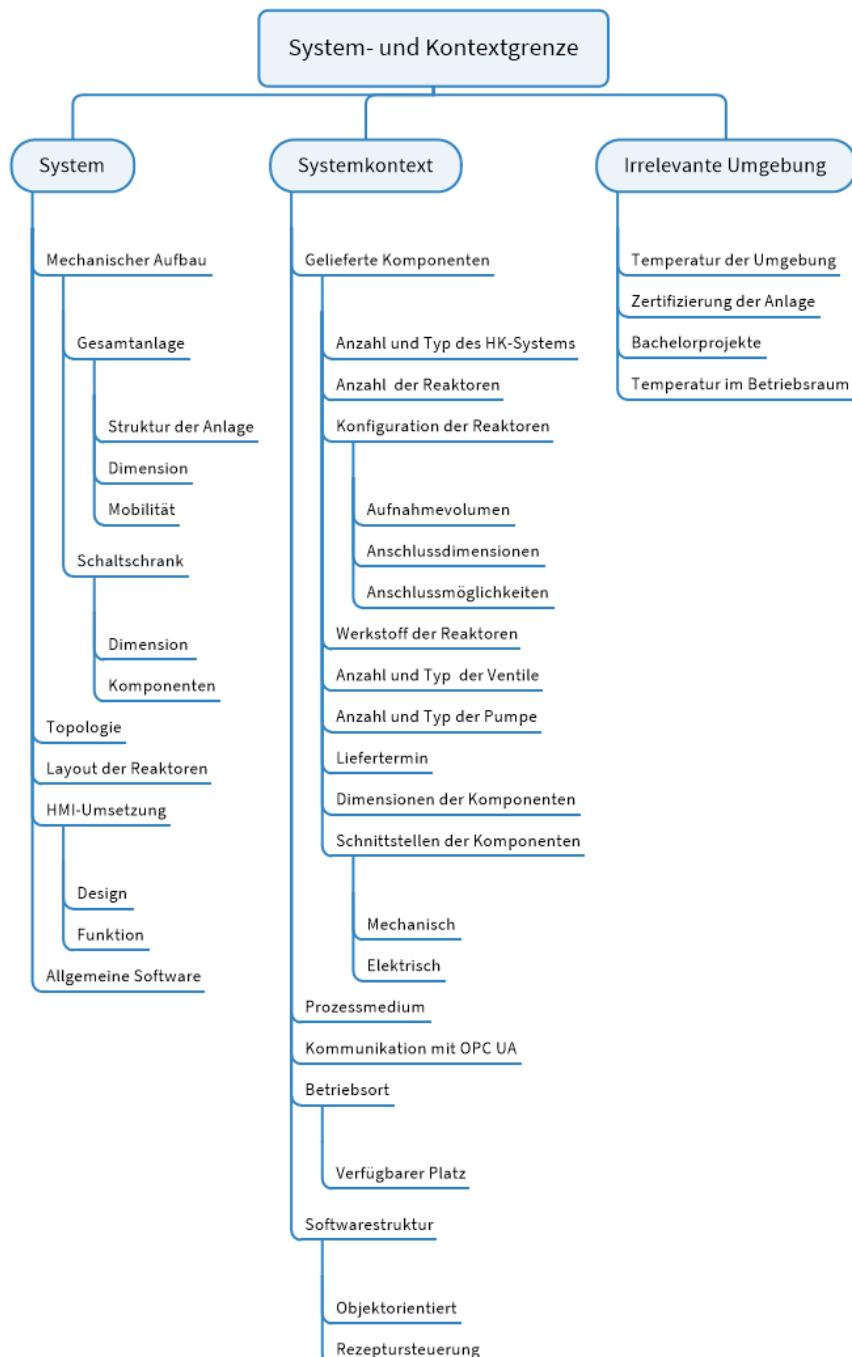


Abbildung 2.15: System- und Kontextgrenzen

2.3 Projektanforderungen

Mit der Definition und Abnahme der Projektanforderungen wird auch die Analyse-Phase abgeschlossen. Die Projektanforderungen bilden die rote Linie durch das Projekt und müssen entsprechend sinnvoll definiert werden. Man unterscheidet dabei zwischen Forderung (F), Mindestanforderung (M) und Wunsch (W). Die Anforderungen stellen ein wichtiges Element des Projektes dar, entsprechend ist es wichtig, dass diese von allen relevanten Personen abgenommen werden.

	Nr	Beschreibung	F / M / W	Anpassung
Mechanischer Aufbau	A1.1	Die chemische Reaktoranlage soll aus 7 Reaktoreinheiten bestehen	M	
	A1.2	Reaktoren können aus dem Gesamtsystem entfernt oder hinzugefügt werden	W	
	A1.3	Die chemische Reaktoranlage ist mobil	W	
	A1.4	Die Position der einzelnen Reaktoren kann in der Höhe eingestellt werden	F	
	A1.5	Die chemische Reaktoranlage wird in der Maschinenhalle der BFH betrieben	F	
	A1.6	Jede Reaktoreinheit enthält einen eigenen Schaltschrank (inkl. SPS)	F	
	A1.7	Schnittstelle zu externen Komponenten / Einheiten (HK / Pumpen etc.)	F	
Software-Struktur	A2.1	Die Software wird mit TwinCat umgesetzt und setzt auf Beckhoff-Komponenten	W	
	A2.2	Die Software wird mit einer objektorientierten Struktur aufgebaut	F	
	A2.3	Die Feldinstrumente (Aktoren / Sensoren) werden als Klassen abgebildet	F	
	A2.4	Der Produktionsprozess wird über eine Rezeptursteuerung definiert	F	
	A2.5	Die Rezeptursteuerung wird nach der Norm ISA-88/95 definiert	F	
	A2.6	Die chemische Reaktoranlage kann im Automatik- oder Manuell-Modus betrieben werden	F	
Kommunikation	A3.1	Die Kommunikation wird über OPC UA realisiert	F	
	A3.2	Die OPC UA Kommunikation verwendet ein Server-Client-Modell	F	
	A3.3	Die OPC UA Kommunikation soll mit der objektorientierten Struktur der Softwareumsetzung arbeiten	F	
	A3.4	Eine Reaktoreinheit kann durch entfernen / verbinden einer Schnittstelle, aus der Topologie entfernt oder integriert werden	W	
	A3.5	Aufbau einer Netzwerktopologie	F	
	A3.6	Von jeder Reaktoreinheit kann man auf die Gesamtanlage zugreifen (wenn integriert)	W	
HMI	A4.1	Die HMI wird mit objektorientierter Software realisiert	W	
	A4.2	Es gibt ein HMI auf der Ebene der Gesamtleitebene	F	
	A4.3	Jede Reaktoreinheit besitzt eine eigene Bedienebene	F	
	A4.4	Eine Rezeptur kann über die HMI der Gesamtleitebene ausgewählt und gestartet werden	F	
	A4.5	Die HMI der Reaktoreinheit ermöglicht die Direktbedienung des Reaktors	F	
	A4.6	Berücksichtigung von Alarmierungen und Trenddarstellung	F	
	A4.7	Die HMI der Reaktoreinheit kann webbasiert erfolgen	W	
	A4.8	Die chemische Reaktoranlage kann im Automatik- oder Manuell-Modus betrieben werden	F	
Allgemein	A5.1	Ein Elektroschema soll erstellt werden	F	
	A5.2	Der Zustand der Anlage wird über eine Signallampe dargestellt	W	
	A5.3	Durchführen eines FAT (Factory Acceptance Test)	W	
	A5.4	Erstellen einer Inbetriebnahmeanleitung	W	
	A5.5	Projektende ist der 02.08.2024	F	

Tabelle 2.4: Projektanforderungen

3 Umsetzung der Systemhardware

3.1 Konstruktiver Aufbau der Anlage

Für die Anlage wird eine neue Konstruktion entworfen, welche die gelieferten Reaktoreinheiten als Basis verwendet. Viele der Komponenten können unverändert oder in leicht bearbeiteter Form eingesetzt werden. Der gesamte Aufbau verfolgt einen modularen Aufbau, mechanisch wie auch bei der Umsetzung der Software. Jeder Reaktor soll ein eigenständiges Teilsystem darstellen, welches alleine oder zusammen mit anderen Reaktoren betrieben werden kann. Die Entwicklung der Konstruktion wird nicht in allen Teilschritten dokumentiert, da der Fokus dieser Arbeit nicht auf der mechanischen Umsetzung liegt. Es wird hauptsächlich die Endlösung beschrieben und welche Überlegungen für einzelne Aspekte gemacht wurden.

3.1.1 Schematischer Aufbau

Jede Reaktoreinheit soll identisch aufgebaut werden. Das Ziel ist, dass das vorgegebene Layout umgesetzt werden kann, jeder Reaktor aber jede Position einnehmen kann. Dafür wurde in einem ersten Schritt ein Schema entworfen, wie die Reaktoreinheit aufgebaut wird. Es gibt 2 Konfigurationen, mit und ohne Pumpe.

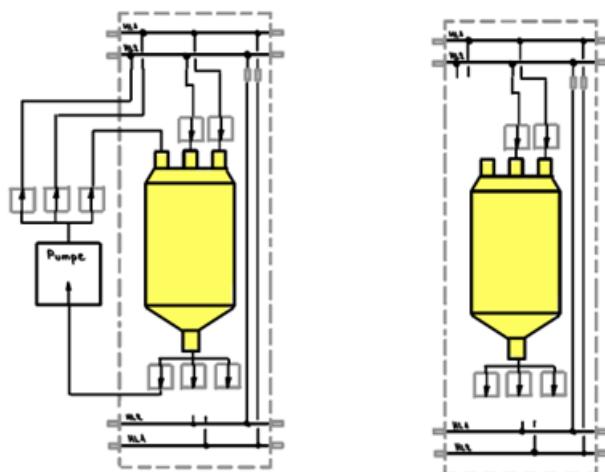


Abbildung 3.1: Mechanisches Schema des Aufbaus

Die Reaktoreinheit besitzt zwei Magnetventile für den Einlass und 3 Ventile für den Auslass. Ein zentraler Aspekt der Einheit sind die zwei Hauptleitungen (HL1 / HL2), welche sich ober- und unterhalb des Reaktors befinden. Diese Hauptlinien sind vertikal verbunden. Alle Schnittstellen aus der Reaktoreinheit können mit einem Handventil geschlossen werden. Die oberen Hauptleitungen führen über die zwei Einlassventile in den Reaktor. Die unteren Leitungen können mit den Auslassventilen verbunden werden. Wenn die Reaktoreinheit eine Pumpe besitzt, kann diese mit einer der Auslassventile verbunden werden. Über Ventile kann die Pumpe in eine der zwei Hauptleitungen oder in den Reaktor

pumpen. Dieser Grundaufbau ermöglicht eine grosse Flexibilität, welche diverse Konfigurationen erlaubt. So auch das vorgegebene Layout der Anlage.

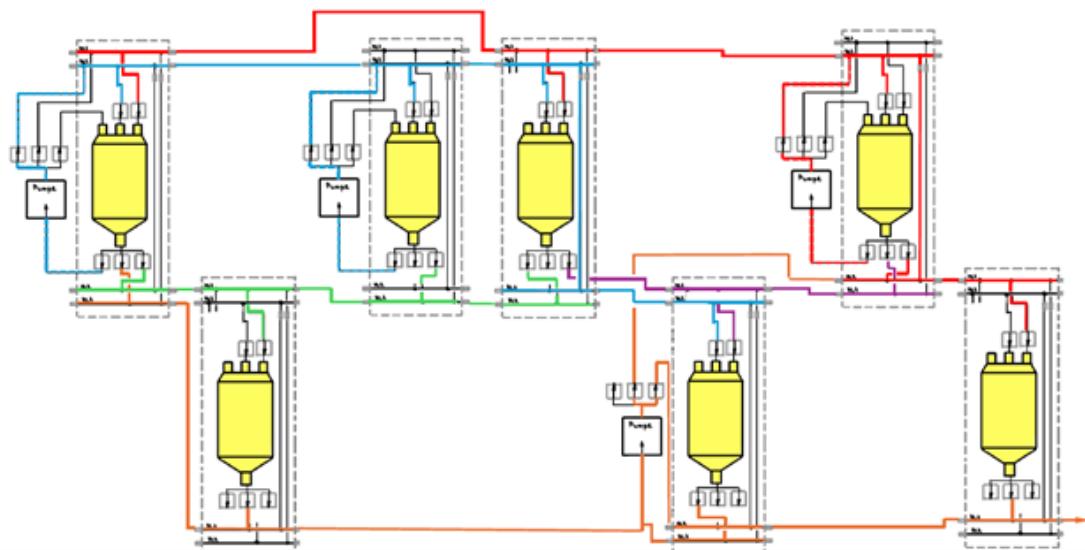


Abbildung 3.2: Umsetzung des Layouts

Die unterschiedlichen Farben repräsentieren die Unterteilung der Abschnitte. Die Schnittstellen zwischen den Reaktoreinheiten fallen dabei einfach aus, da nur die Hauptleitungen miteinander verbunden werden müssen.

3.1.2 Konstruktive Umsetzung

Die Entwicklung des mechanischen Aufbaus hat diverse Iterationen durchlaufen. Der Aufbau wurde innerhalb des zeitlichen Rahmens des Projektes umgesetzt. Der Schwerpunkt lag dabei auf der Funktionalität.

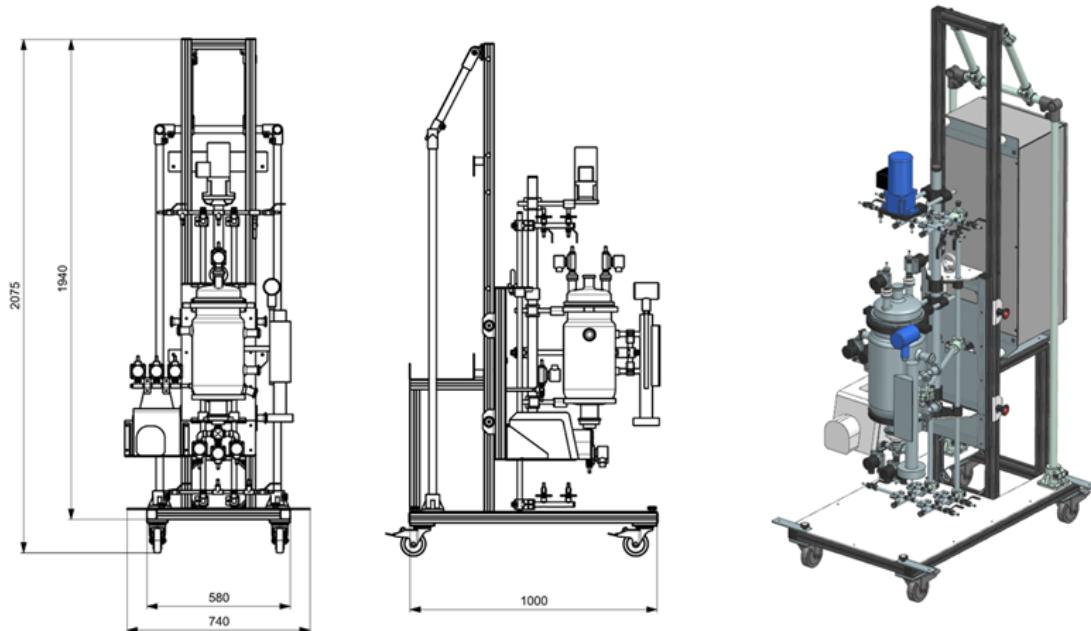


Abbildung 3.3: Überblick über Gesamtsystem

Die Basis bildet ein Grundgestell aus Alu-Profilen. Die hohe Flexibilität und die Möglichkeit, nachträglich Ergänzungen oder Anpassungen vorzunehmen, waren die Hauptargumente für diesen Entscheid. Die schlechte Zugänglichkeit für die Reinigung wurde vernachlässigt. Bei einer späteren Optimierung, können die Alu-Profile durch geschlossene Profile ersetzt werden.

Das Grundgestell nimmt die Reaktoreinheit wie auch den Schaltschrank auf. Letzterer wird auch zur Stabilisierung verwendet. Der gesamte Aufbau ist deutlich grösser als die ursprüngliche Reaktoreinheit. Die Position des Reaktors in Z-Richtung kann mit definierten Stufen eingestellt werden. Grundsätzlich gibt es zwei Positionen, oben und unten. Die unteren Hauptleitungen eines oberen Reaktors sind leicht höher als die oberen Hauptleitungen des unteren Reaktors. Das leichte Gefälle zwischen den Leitungen soll das Medium zum Fliessen bringen. Um die Stabilität auch auf der höchsten Stufe sicherzustellen, braucht der Grundriss des Grundgestells eine gewisse Grösse. Der Schaltschrank wird auf der Rückseite befestigt und ist ein fixer Teil des Systems.

Die Haltevorrichtung für den Reaktor ist eine separate Baugruppe, welche über Befestigungsvorrichtungen mit dem Grundgestell verbunden wird. Diese dienen auch als Führung für die Höheneinstellung des Reaktors.

Über eine Ringschraube kann die Höhe, mittels eines Werkstattkran versteilt werden. Die Höhenverstellung wurde einfach gehalten, da die Höhe nicht oft geändert werden muss. Maximal für jedes neue Semester, um die Anlage in eine neue Konfiguration zu bringen. Die Haltevorrichtung nimmt alle prozessrelevanten Komponenten auf. Viele der Befestigungskomponenten der gelieferten Reaktoreinheit werden unverändert oder leicht angepasst eingesetzt. Der Reaktor und der Rührantrieb werden durch die originalen Befestigungen gehalten. Die Position des Füllstandssensors wurde so angepasst, dass die Anzeige nach vorne zeigt. Für die Schlauchquetschpumpe wurde ein Abstellplatz vorgesehen. Falls die Reaktoreinheit eine Pumpe besitzen soll, kann diese auf der entsprechenden Ablage positioniert werden. Die Haltevorrichtung beinhaltet auch die zwei oberen und unteren Hauptleitungen.

Der gesamte Aufbau für einen Reaktor ist für jeden Reaktor gleich. Die Reaktoreinheiten können mechanisch miteinander verbunden werden. Dafür sind Halbleche auf dem Grundgestell vorgesehen. Die Reaktoreinheiten können dadurch schnell miteinander verbunden werden, aber auch gelöst werden. Die Anlage bietet somit eine grosse Flexibilität, falls eine Reaktoreinheit alleine betrieben werden soll.

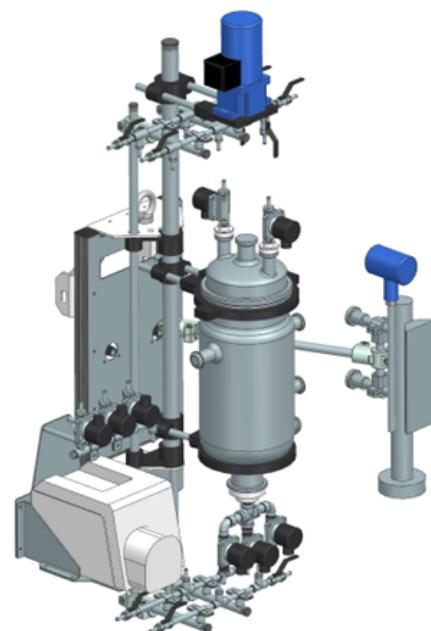


Abbildung 3.4: Überblick über Reaktoraufbau



Alle Fertigungszeichnungen werden im Anhang beigelegt

3.2 Aufbau des Schaltschranks

3.2.1 Planung des Schaltschranks

Damit der Schaltschrank gebaut werden kann, müssen die benötigten Komponenten definiert werden. Da die Feldgeräte bereits durch die Anlage definiert sind, geht es nur noch um die elektrischen Komponenten, wie Netzteile oder Beckhoff-Karten. Für letztere wurden bereits einige Karten für das Projekt bestellt (Kapitel 2.1.2). Somit muss die Frage geklärt werden, ob noch weitere Karten benötigt werden. Dafür wird ein schematischer Aufbau der elektrischen Situation definiert. Der Aufbau zeigt die Schnittstellen und Verbindungen, die für den Betrieb einer Reaktoreinheit benötigt werden. Jede Reaktoreinheit wird dabei identisch aufgebaut. Das Schema ist nur eine einfache Repräsentation der Schnittstellen und stellt nicht ein Elektroschema dar.

IPC und E/A-Koppler:

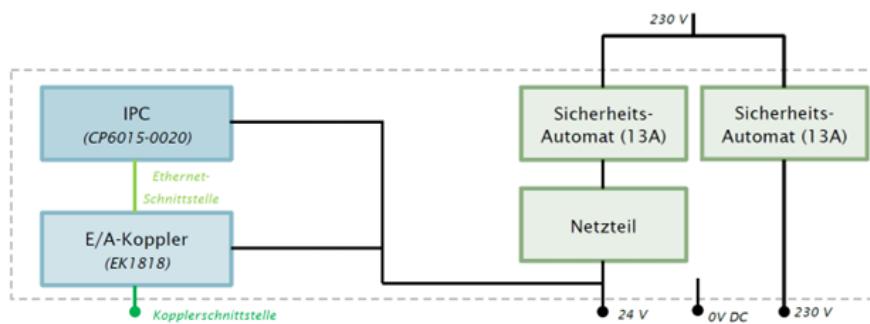


Abbildung 3.5: Schnittstellen für IPC und E/A-Koppler

Die Reaktoreinheit wird über den IPC (CP6015-0020) von Beckhoff gesteuert. Der IPC ist mit dem E/A-Koppler (EK1818) über eine Ethernet-Schnittstelle verbunden. Beide Komponenten werden durch ein Netzteil mit 24 V gespiesen. Diese drei Komponenten wurden bereits in Vorbereitung für das Projekt beschafft. Innerhalb der Reaktoreinheit werden 24 V und 230 V Schnittstellen benötigt. Für die 230 V Leitungen werden 2 Sicherheitsautomaten vorgesehen.

Anschluss der Ventile:

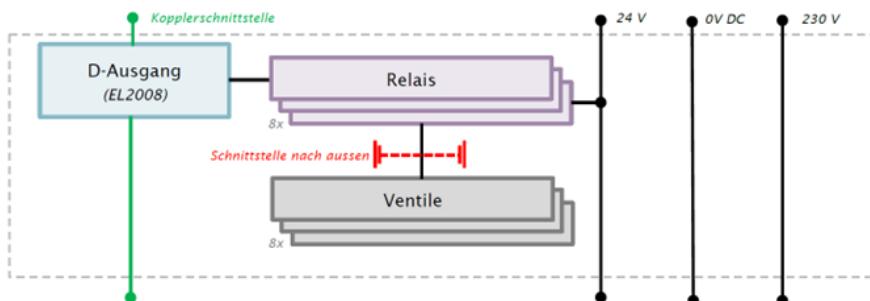


Abbildung 3.6: Schnittstellen Ventilanschluss

Für die Ventile wird die 8-Kanal-Digital-Ausgangskarte (EL2008) eingesetzt. Von dieser Karte wurden acht Stück für das Projekt bestellt. Die Karte wird verwendet um acht Relais anzusteuern. Diese sind mit den Ventilen verbunden und versorgen diese beim Ansteuern mit 24 V. Zwischen Relais und Ventile gibt es eine Schnittstelle nach aussen, welche über Anschlussklemmen realisiert wird.

Auslesen des Füllstandssensors:

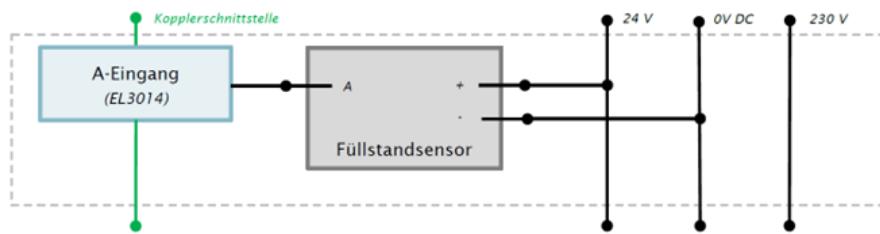


Abbildung 3.7: Schnittstellen Füllstandauswertung

Das 4-20 mA Signal des Füllstandssensors wird über die 4-Kanal-Analog-Eingangskarte (EL3014) ausgelesen. Von dieser Karte wurden acht Stück für das Projekt bestellt.

Ansteuern der Schlauchquetschpumpe:

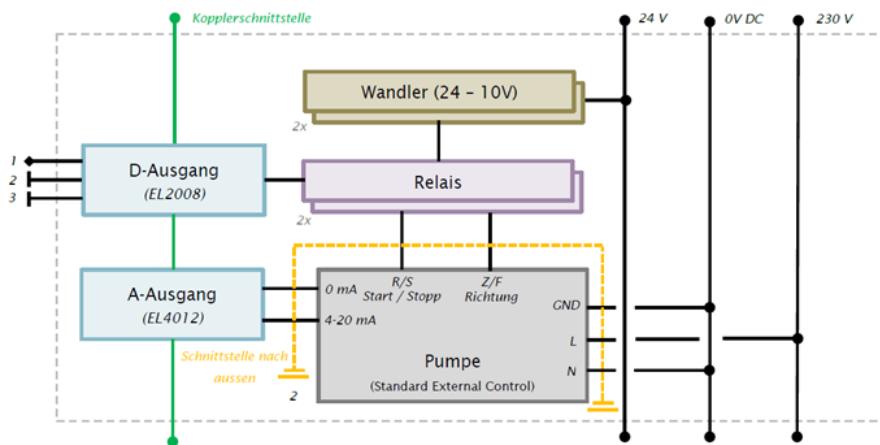


Abbildung 3.8: Schnittstellen Schlauchquetschpumpe

Die Geschwindigkeit der Schlauchquetschpumpe kann über ein analoges 4-20 mA Signal definiert werden. Dafür wird die 2-Kanal-Analog-Ausgangskarte (EL4012) verwendet. Von dieser Karte wurden acht Stück für das Projekt bestellt. Über zwei 10 V Signale kann die Pumpe gestartet / gestoppt und die Drehrichtung bestimmt werden. Für das Schalten der 10 V werden 2 Relais verwendet, welche wieder von einer EL2008 gesteuert werden. Diese acht EL2008-Karten können aus BFH-Beständen entnommen werden und müssen nicht bestellt werden. Das 10 V Signal kommt von einem Wandler. Die Pumpe selber muss mit 230 V gespiesen werden. Alle Anschlüsse der Pumpe sind Schnittstellen aus dem Schaltschrank heraus und werden über Anschlussklemmen umgesetzt. Die EL2008-Karte wird auch verwendet um zu prüfen, ob eine Pumpe oder ein HK-System angeschlossen ist. Eine angeschlossene Pumpe schliesst den Kontakt und gibt ein Signal weiter an eine digitale Eingangskarte.

Ansteuern des HK-Systems:

Zum jetzigen Zeitpunkt konnte das HK-System noch nicht ausführlich getestet werden. Wie genau das System angeschlossen werden kann, ist somit nicht bekannt. Die im Moment definierten Karten bieten jedoch viele Möglichkeiten das HK-System anzusteuern. Es besteht die Möglichkeit digitale Ausgangssignale zu verwenden oder ein analoges 4-20 mA Signal.

Ansteuern des Antriebs für den Rührer:

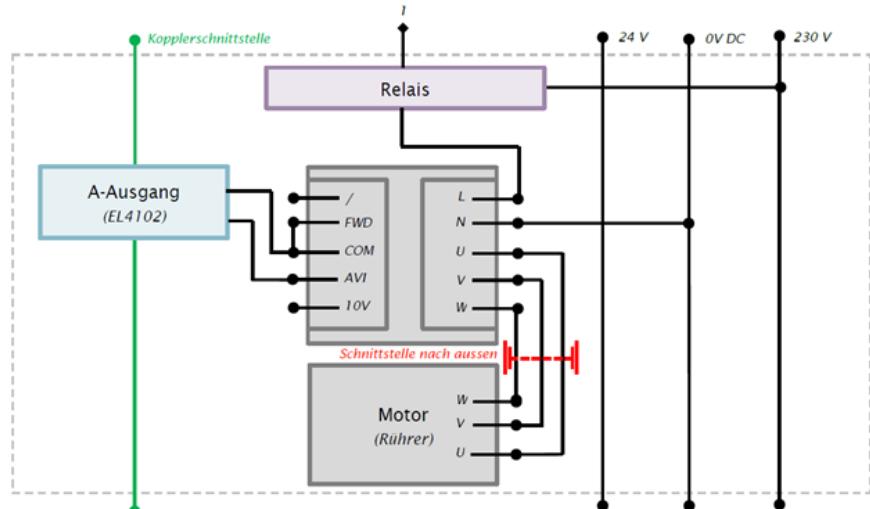


Abbildung 3.9: Schnittstellen Rührantrieb

Der Antrieb für den Rührer wird über ein FU-Element betrieben. In der ursprünglichen Konfiguration wurde die Geschwindigkeit über ein Potentiometer eingestellt, welches einen Wert von 0 – 10 V ausgab. Das Potentiometer wird durch die 2-Kanal-Analog-Ausgangskarte (EL4102) ersetzt. Die Karte kann ein analoges 0 – 10 V Signal ausgeben. Die Karte muss nachträglich bestellt werden. Das FU-Element wird mit 230 V gespiesen. Damit der Motor ein- und ausgeschalten werden kann, wird die Spannungsversorgung des FU-Elements über ein Relais realisiert. Dieses wird von der EL2008-Karte geschaltet.

Auslesen der Temperatur:

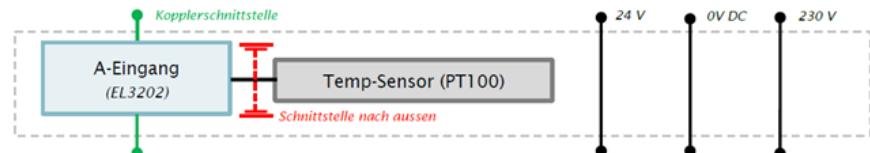


Abbildung 3.10: Schnittstellen Temperatursensor

Der PT100 wird über eine spezielle 2-Kanal-Analog-Eingangskarte (EL3202) ausgelesen, welche spezifisch für PT100-Sensoren geeignet ist. Die Karte muss nachträglich bestellt werden.

Einlesen des Prüfsignals für Komponenten:



Abbildung 3.11: Schnittstellen Prüfsignalauswertung

Damit geprüft werden kann, ob eine Schlauchquetschpumpe oder ein HK-System angeschlossen ist, wird eine 8-Kanal-Digital-Eingangskarte verbaut. Die Karte muss nachträglich bestellt werden.

3.2.2 Umsetzung des Schaltschranks

Auf Basis des Schemas wurde ein Elektroschema erstellt (erstellt von Melchior Borer). Damit wurden die Schaltschränke für die Reaktoren gebaut. Die Planung, Umsetzung und Probleme während des Schaltschrankbaus werden in dieser Dokumentation nicht aufgegriffen.



Das Elektroschema und eine Liste der Komponenten wird im Anhang beigelegt

4 Planung der Software

4.1 Entwurf einer Rezeptsteuerung

4.1.1 Definieren der Grundbausteine

Für den objektorientierten Aufbau der Software sind die Grundbausteine ein wichtiger Bestandteil. Die Grundbausteine spiegeln unterschiedliche Typen von Aktoren und Sensoren dar und deren Funktion im System. Die jeweiligen Feldgeräte werden durch diese Grundbausteine instanziert. Gut geplante und umgesetzte Grundbausteine vereinfachen die Realisierung des Ablaufs erheblich. Jede Reaktoreinheit kann dadurch identisch aufgebaut werden und enthält dieselben Grundbausteine. Grundbausteine werden auch als Klassen bezeichnet. Eine Klasse besteht aus Methoden, Eigenschaften und den verschiedenen Variablen. Für die Beschreibung der Klassen werden nur die Input- und Output-Variablen angegeben. Die lokalen Variablen und die konkrete Programmierung der Klasse wird nicht weiter vertieft. Definiert werden sechs Grundbausteine:

Bezeichnung:	A_Motor1R	
Methoden:	- M_A_AN	<i>Motor wird eingeschalten im Automatik-Modus</i>
	- M_A_AUS	<i>Motor wird eingeschalten im Automatik-Modus</i>
	- M_A_START	<i>Motor beginnt zu drehen im Automatik-Modus</i>
	- M_A_STOP	<i>Motor stoppt im Automatik-Modus</i>
	- M_M_AN	<i>Motor wird eingeschalten im Manuell-Modus</i>
	- M_M_AUS	<i>Motor wird eingeschalten im Manuell-Modus</i>
	- M_M_START	<i>Motor beginnt zu drehen im Manuell-Modus</i>
	- M_M_STOP	<i>Motor stoppt im Manuell-Modus</i>
Eigenschaften:	- P_MaxDrz (SET)	<i>Angeben der maximalen Motordrehzahl</i>
	- P_Laufzeit (GET)	<i>Rückgabe der aktuellen Laufzeit des Motors</i>
	- P_Zustand (GET)	<i>Rückgabe des aktuellen Zustandes des Antriebs</i>
Input-Variablen:	- I_iModus	<i>Zustand des Systems (z.B. 0 = AUS / 1 = MAN / 2 = AUTO)</i>
	- I_bVerriegeln	<i>Verriegelte Elemente sind nicht ansteuerbar</i>
	- I_iGeschw	<i>Angabe über Soll-Geschwindigkeit</i>
Output-Variablen:	- O_iDrz	<i>Ausgabe der Drehzahl für den Motor</i>
	- O_iStörung	<i>Anzeigen von Störung über eine INT-Variable</i>

Tabelle 4.1: Planung von Rührer-Grundbaustein

Bezeichnung:	A_Motor2R	
Methoden:	- M_A_FWD	Drehrichtung wird auf vorwärts gesetzt im Automatik-Modus
	- M_A_BWD	Drehrichtung wird auf rückwärts gesetzt im Automatik-Modus
	- M_A_START	Motor beginnt zu drehen im Automatik-Modus
	- M_A_STOP	Motor stoppt im Automatik-Modus
	- M_M_FWD	Drehrichtung wird auf vorwärts gesetzt im Manuell-Modus
	- M_M_BWD	Drehrichtung wird auf rückwärts gesetzt im Manuell-Modus
	- M_M_START	Motor beginnt zu drehen im Manuell-Modus
	- M_M_STOP	Motor stoppt im Manuell-Modus
Eigenschaften:	- P_MaxDrz (SET)	Angeben der maximalen Motordrehzahl
	- P_Laufzeit (GET)	Rückgabe der aktuellen Laufzeit des Motors
	- P_Zustand (GET)	Rückgabe des aktuellen Zustandes des Antriebs
Input-Variablen:	- I_bVerbunden	Angabe ob Pumpe angeschlossen ist
	- I_iModus	Zustand des Systems (z.B. 0 = AUS / 1 = MAN / 2 = AUTO)
	- I_bVerrigeln	Verriegelte Elemente sind nicht ansteuerbar
	- I_iGeschw	Angabe über Soll-Geschwindigkeit
Output-Variablen:	- O_iDrz	Ausgabe der Drehzahl für den Motor
	- O_bRichtung	Ausgabe der Drehrichtung
	- O_iStörung	Anzeigen von Störung über eine INT-Variable

Tabelle 4.2: Planung von Schlauchquetschpumpe-Grundbaustein

Bezeichnung:	A_HKSystem	
Methoden:	- M_A_HEIZEN	HK-System heizt in Automatik-Modus
	- M_A_KÜHLEN	HK-System kühlt in Automatik-Modus
	- M_M_HEIZEN	HK-System heizt in Manuell-Modus
	- M_M_KÜHLEN	V HK-System kühlt in Manuell-Modus
Eigenschaften:	- P_MaxTemp (SET)	Angeben der maximal Temperatur des Reaktormediums
	- P_MinTemp (SET)	Angeben der minimal Temperatur des Reaktormediums
	- P_Zustand (GET)	Rückgabe des aktuellen Zustandes des HK-Systems
Input-Variablen:	- I_iModus	Zustand des Systems (z.B. 0 = AUS / 1 = MAN / 2 = AUTO)
	- I_bVerriegeln	Verriegelte Elemente sind nicht ansteuerbar
	- I_iTemp	Angabe der gewünschten Temperatur
Output-Variablen:	- O_bKühlen	HK-System heizt das Medium auf
	- O_bHeizen	HK-System kühlt das Medium ab
	- O_iStörung	Anzeigen von Störung über eine INT-Variable

Tabelle 4.3: Planung von HK-System-Grundbaustein

Bezeichnung:	A_Magnetventil	
Methoden:	- M_A_ÖFFNEN	Ventil öffnen in Automatik-Modus
	- M_A_SCHLIESSEN	Ventil schliessen in Automatik-Modus
	- M_M_ÖFFNEN	Ventil öffnen in Manuell-Modus
	- M_M_SCHLIESSEN	Ventil schliessen in Manuell-Modus
Eigenschaften:	/	
Input-Variablen:	- I_iModus	Zustand des Systems (z.B. 0 = AUS / 1 = MAN / 2 = AUTO)
	- I_bVerriegeln	Verriegelte Elemente sind nicht ansteuerbar
Output-Variablen:	- O_bAktivieren	Ventil wird geöffnet / geschlossen
	- O_iStörung	Anzeigen von Störung über eine INT-Variable

Tabelle 4.4: Planung von Magnetventil-Grundbaustein

Bezeichnung:	A_Füllstandssensor	
Methoden:	- M_START	Messprozess wird gestartet
	- M_STOP	Messprozess wird gestoppt
Eigenschaften:	/	
Input-Variablen:	- I_iSignal	Messsignal des Sensors
Output-Variablen:	- O_iFüllstand	Ausgabe des momentanen Füllstandes
	- O_iStörung	Anzeigen von Störung über eine INT-Variable

Tabelle 4.5: Planung von Füllstandssensor-Grundbaustein

Bezeichnung:	A_Temperatursensor	
Methoden:	- M_START	Messprozess wird gestartet
	- M_STOP	Messprozess wird gestoppt
Eigenschaften:	/	
Input-Variablen:	- I_iSignal	Messsignal des Sensors
Output-Variablen:	- O_iTemp	Ausgabe der momentanen Temperatur
	- O_iStörung	Anzeigen von Störung über eine INT-Variable

Tabelle 4.6: Planung von Temperatursensor-Grundbaustein

Die definierten Grundbausteine dienen als Referenz für die Umsetzung. Während der konkreten Erarbeitung der TwinCat-Software können diese von der Planung abweichen. Sobald die Grundbausteine final definiert wurden, werden diese nochmals beschrieben.

4.1.2 Analyse des Prozesses

Ein wichtiger Bestandteil der Entwicklung einer Rezeptsteuerung ist die Prozessanalyse. Hierbei wird der Prozess in Teiltätigkeiten zerlegt, welche die Grundoperationen (GOP) bilden. Diese Operationen werden aus den Rezeptparametern und Grundfunktionen (GF) zusammengesetzt. Die Grundfunktionen stellen geräteneutrale Tätigkeiten in Form von Schrittabläufen dar. Innerhalb der Schrittabläufe werden die jeweiligen Methoden und Eigenschaften der Schnittstellen verwendet. Im Unterschied zu Klassen, definieren Schnittstellen nur welche Methoden und Eigenschaften eine Klasse haben muss.

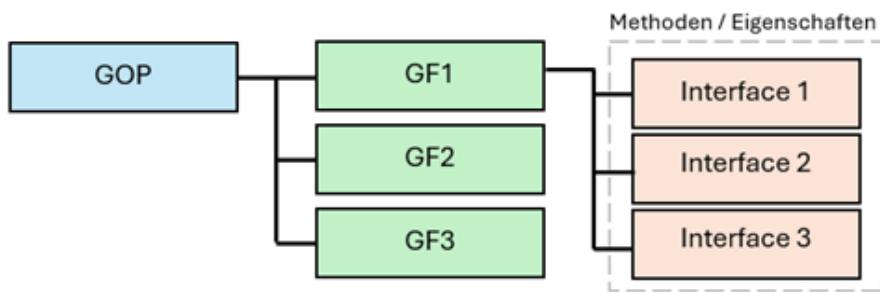


Abbildung 4.1: Struktur eines Rezeptprozesses

Das Prinzip der Grundfunktionen mit Schnittstellen (Interface) ist das gleiche wie bei den Klassen. So wie die Klassen verwendet werden können um die anlagenspezifischen Feldgeräte zu instanzieren, so dienen die Grundfunktionen als Basis für die jeweiligen Reaktoreinheiten. Die Funktion der Schnittstelle ist dabei, die Methoden und Eigenschaften der entsprechend instanzierten Objekte verwendbar zu machen.

Für die Umsetzung der chemischen Reaktoranlage kann für die Rezeptur direkt mit den Grundfunktionen gearbeitet werden. Der Prozess lässt sich nicht in Teilprozesse und somit Grundoperationen aufteilen.

Für die chemische Reaktoranlage werden folgende Grundfunktionen definiert:

Bezeichnung	Aktoren	Sensoren	Ablaufspezifische Parameter
GF_Heizen	HK-System	Temperatursensor	Temperatur
GF_Kühlen	HK-System	Temperatursensor	Temperatur
GF_Halten	HK-System	Temperatursensor	Haltestdauer
GF_Entleeren	Ventil / Pumpe	Füllstandsensor	Wohin (Ventil), Menge
GF_Röhren	Antriebsmotor	/	Geschwindigkeit, Dauer
GF_Pumpen	Ventil / Pumpe	/	Dauer
GF_Warten	/	/	Dauer

Tabelle 4.7: Definierte Grundfunktionen

Damit die Komplexität der Grundfunktionen eingeschätzt werden kann, werden die jeweiligen Schrittabläufe schematisch definiert. Diese stellen nicht die finalen Schrittabläufe dar, sondern dienen als Brainstorming für relevante Punkte, welche für den Betrieb beachtet werden müssen. Alle Schrittabläufe beziehen sich auf eine Reaktoreinheit.

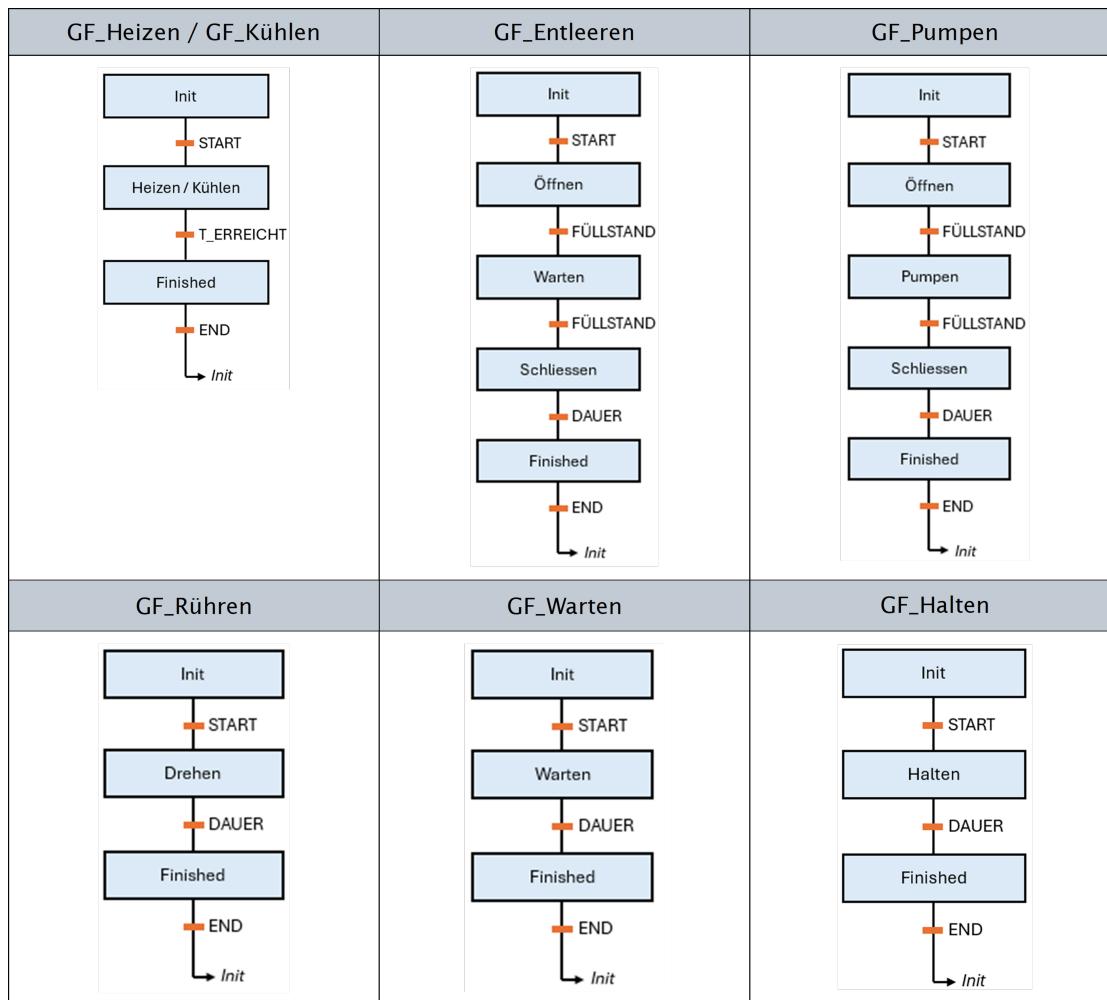


Tabelle 4.8: Grobschema der Grundfunktionen

Im Moment machen die Schrittabläufe keine Aussage darüber, wie die Interaktion zwischen den Reaktoren aussieht. Dies ist jedoch ein zentraler Bestandteil der Funktionalität des kompletten Systems. Das System benötigt einen Mechanismus, welcher die Interaktion zwischen Reaktoren regelt. Dies kann der Status der Leitungen sein oder ob eine Pumpe benötigt wird, um das Medium zum gewünschten Reaktor zu befördern. Sehr vereinfacht dargestellt, muss folgende Interaktion stattfinden, unabhängig ob dies direkt zwischen den Reaktoren geschieht oder über eine übergeordnete Ebene. Beim Beispiel soll das Medium von Reaktor 1 zu Reaktor 2 übergeben werden.

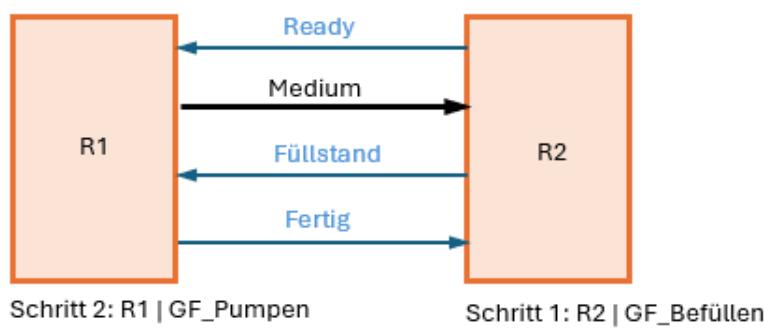


Abbildung 4.2: Grobschema der Reaktorinteraktion

In einem ersten Schritt muss der Reaktor 2 die Grundfunktion «Befüllen» starten. Dabei werden die entsprechenden Ventile geöffnet. Innerhalb dieses Ablaufs wird dem Reaktor 1 das Signal «Ready» gegeben und dieser startet mit der Grundfunktion «Pumpen». Der Füllstand von Reaktor 2 dient als Bedingung für den Reaktor 1, wie lange dieser pumpen muss. Beim Erreichen des entsprechenden Füllstandes wird die Grundfunktion «Pumpen» beendet. Mit dem Rückmeldungssignal wird dann bei Reaktor 1 die Grundfunktion «Befüllen» beendet.

4.1.3 Komposition des Prozesses

Das Rezept definiert den Prozessablauf und wird entsprechend auf der Gesamtsystemebene ausgeführt. Innerhalb der Rezeptschritte werden über «Aktionen» die entsprechenden Grundfunktionen gestartet. Diese rufen über eine «Aktion» die instanzierten Funktionsblöcke mit ihren Methoden und Eigenschaften auf. Die Grundfunktionen werden auf Stufe der Reaktoreinheiten ausgeführt. Das Rezept muss entsprechend wissen, bei welcher Reaktoreinheit die Grundfunktion ausgelöst werden muss.

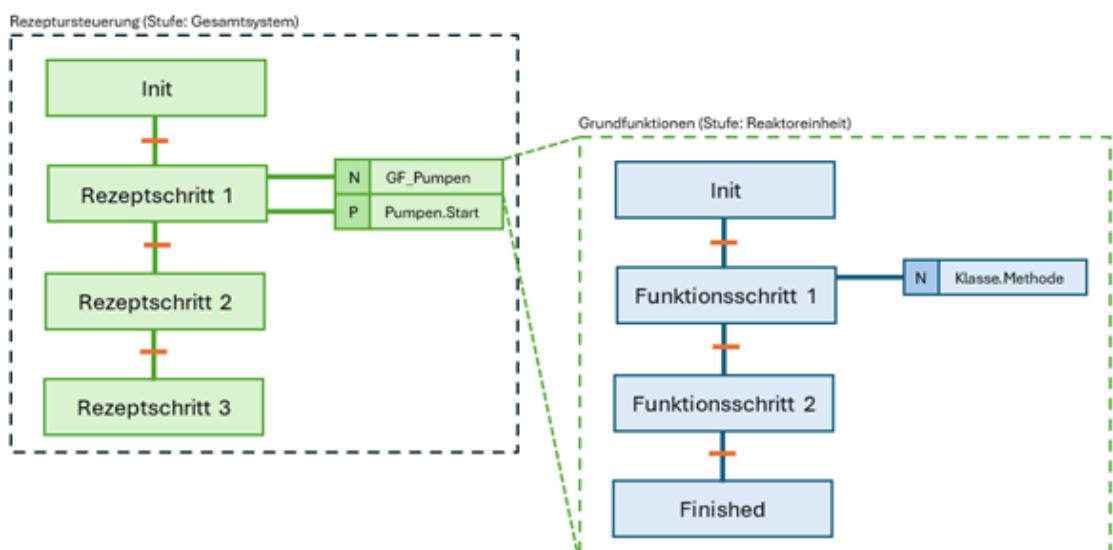


Abbildung 4.3: Arbeitsweise einer Rezeptsteuerung

Nun stellt sich die Frage, wie die Grundfunktionen ausgelöst werden sollen, wenn ein spezifischer Reaktor angesprochen wird. In der Literatur «Speicherprogrammierbare Steuerungen für die Fabrik- und Prozessautomation» von Matthias Seitz wird nicht mit unterschiedlichen Untersystemen gearbeitet. Entsprechend kann eine Grundfunktion über einen Schrittablauf aufgerufen und ausgeführt werden. Auch die Berücksichtigung einer Kommunikationsschnittstelle via OPC-UA wird nicht behandelt. Das Buch fokussiert sich auf eine lokale Rezeptsteuerung, auf einer SPS.

Die chemische Reaktoranlage ist durch das Kombinieren von Reaktorsystemen mittels OPC-UA-Schnittstelle deutlich komplexer und erfordert eine eigene Struktur. Das Ziel ist jedoch immer noch, dass die Grundfunktionen innerhalb eines Reaktorsystems über das Rezept ausgelöst werden.

Die Kommunikationsstruktur könnte wie folgt aussehen:

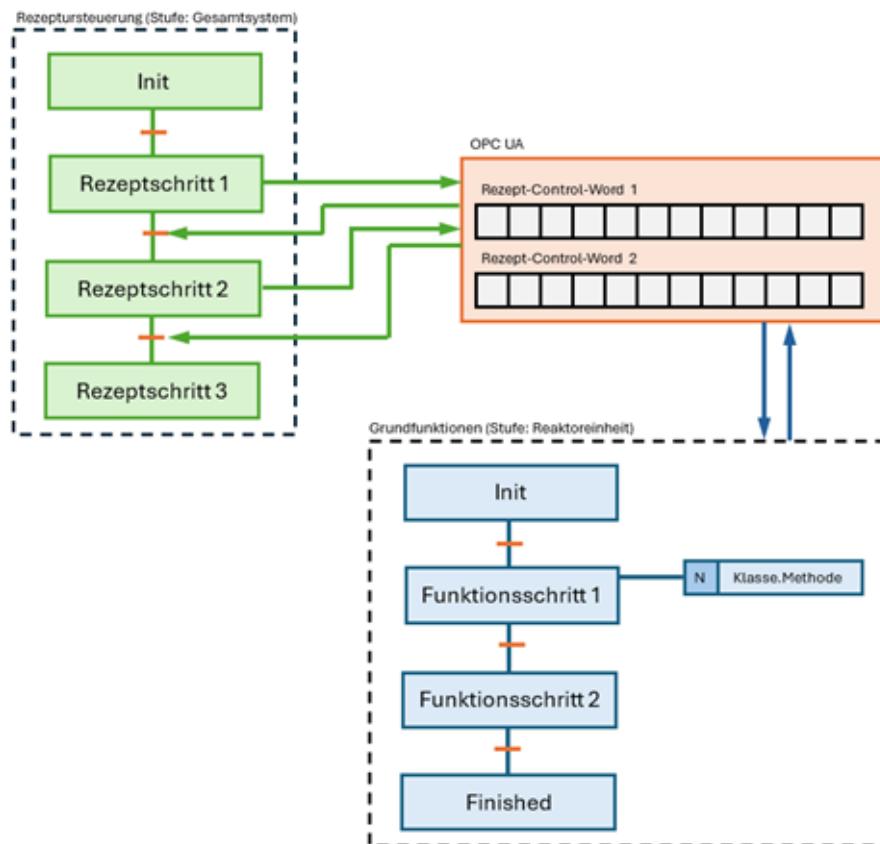


Abbildung 4.4: Erster Entwurf einer möglichen Reaktorkommunikation

Die Übermittlung von Rezept zu Reaktoreinheit wie auch die Interaktion zwischen zwei Reaktoreinheiten wird über ein Control-Word gehandelt. Das Control-Word ist ein Kommunikationsframe, wobei jede Stelle eine definierte Funktion hat. Eine Position kann jeden Datentyp annehmen. Das Control-Word hat folgende Aufgaben:

- ▶ Definieren des Reaktors
- ▶ Definieren der auszuführenden Grundfunktion
- ▶ Definieren der Prozessparameter
- ▶ Darstellen von Prozess- und Systemzustand

Das Rezept gibt anhand des Rezeptschrittes ein Control-Word vor mit den notwendigen Informationen. Alle Reaktoreinheiten im System können über die OPC-UA-Schnittstelle auf das Control-Word zugreifen. Dadurch weiss der angesprochene Reaktor, welche Grundfunktion mit welchen Parametern ausgeführt werden muss. Während des Prozesses kann das Control-Word durch die Reaktoreinheiten verändert werden. Ein Reaktorsystem kann somit Informationen über seinen Zustand oder den Prozess an die Gesamtsystemebene weitergeben. Dadurch werden Interaktionen zwischen Reaktoren möglich. Jede Grundfunktion wird über ein separates Control-Word angesprochen. Somit kann ein Rezeptschritt mehrere Control-Words bereitstellen und Grundfunktionen können parallel ausgeführt werden.

Das Control-Word muss folgende Informationen abbilden:

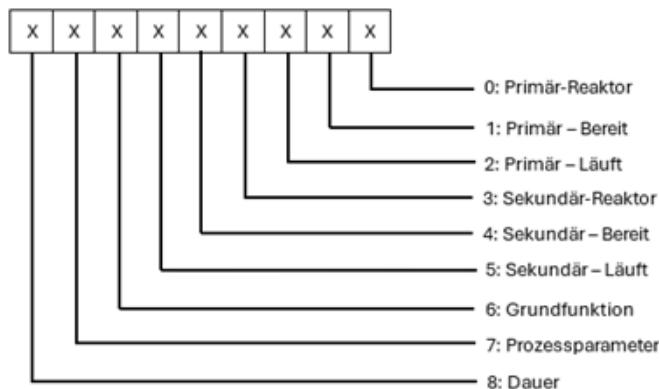


Abbildung 4.5: Erster Entwurf des Control-Words

- | | |
|-------------------|--|
| Primär-Reaktor: | Der Primärreaktor führt die definierte Grundfunktion durch und stellt bei der Interaktion zwischen zwei Reaktoren das Medium bereit. |
| Primär–Bereit: | Definiert, ob Primärreaktor bereit ist für den Prozess. |
| Primär–Läuft: | Definiert, ob der Prozess beim Primärreaktor läuft. |
| Sekundär–Reaktor: | Der Sekundärreaktor ist bei der Interaktion zwischen zwei Reaktoren der Empfänger. |
| Sekundär–Bereit: | Definiert, ob der Sekundärreaktor für den Prozess bereit ist. |
| Sekundär–Läuft: | Definiert, ob der Prozess beim Sekundärreaktor läuft. |
| Grundfunktion: | Definiert, welche Grundfunktion durchgeführt wird. |
| Prozessparameter: | Die Prozessparameter definieren je nach Grundfunktion die relevanten Informationen, wie Temperatur oder Geschwindigkeit. |
| Dauer: | Definiert die Warte- oder Prozessdauer |

Um die Funktionalität des definierten Rezept-Control-Words zu zeigen und prüfen, wird ein Beispiel anhand einer vereinfachten Situation durchgespielt. Das vereinfachte System besteht aus 2 Reaktoren. Reaktor 1 soll 100 ml des Mediums in Reaktor 2 pumpen. Dieser Vorgang wird durch den «Rezeptschritt 1» ausgelöst.

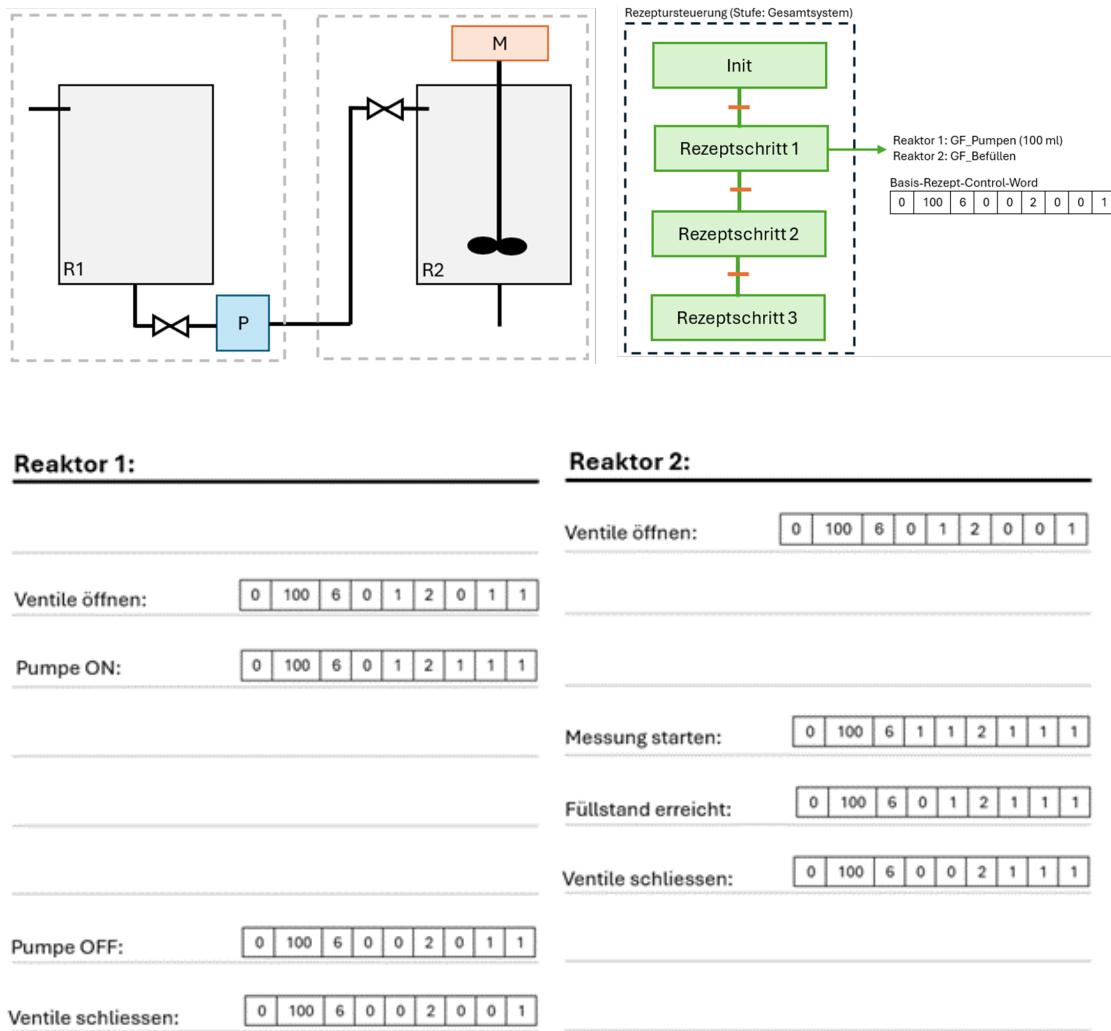


Abbildung 4.6: Erster Entwurf von Interaktion zwischen zwei Reaktoren

Der dargestellte Ablauf bezieht sich nur auf die Interaktion zwischen zwei Reaktoren. Im finalen System sind 7 Reaktoren vorgesehen. Mit der momentanen Struktur des Rezept-Control-Words kann pro Control-Word nur eine Grundfunktion definiert werden. Falls zwei Funktionen parallel ausgeführt werden möchten, müssen zwei Control-Words verwendet werden. Da es dadurch mehrere Rezept-Control-Words geben muss, muss auch jede Reaktoreinheit kontrollieren, ob diese angesprochen wird. Dies kann zu einer sehr unübersichtlichen Verwaltung führen. Als Vereinfachungsmassnahme wird das Rezept-Control-Word neu definiert. Das Ziel ist, dass alle Grundfunktionen parallel abgebildet werden können. Dies ermöglicht, dass jede Reaktoreinheit ein fix zugewiesenes Rezept-Control-Word erhält. Als zusätzliche Massnahme wird die Rezept-Interaktion in ein Control- und Status-Word aufgeteilt. Diese sind wie folgt definiert:

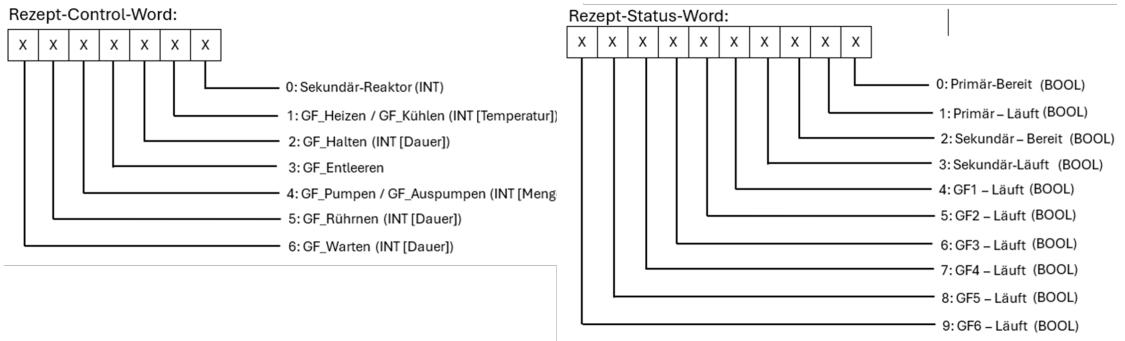


Abbildung 4.7: Nächste Iteration von Control/Status-Word

Das Rezept-Control-Word kommuniziert von der Gesamtsystemebene an die Reaktoreinheitsebene. Hierbei werden prozessrelevanten Informationen an die Reaktoreinheiten weitergegeben. Der aktuelle Zustand des Prozesses wird im Status-Word abgebildet. Dieses dient als Rückmeldung zwischen den zwei Ebenen. Der Reaktor hat ein eigenes Rezept-Control/Status-Word um mit der Gesamtsystemebene zu kommunizieren. Die Reaktoren interagieren über das Rezept-Status-Word miteinander.

4.1.4 Management der Leitung

Wie bereits erwähnt, wird das Management der Leitungen auf der Gesamtsystemebene ausgeführt. Der Vorteil ist, dass die Reaktoreinheit nicht Teil des Entscheidungsmanagements sind. Somit können auch keine Probleme bezüglich Berechtigung entstehen. Ansonsten müsste klar definiert werden, welcher Reaktor, zu welchem Zeitpunkt eine Leitung besetzen oder freigeben darf.

Wie bereits in der Situationsanalyse grob definiert wurde (Kapitel 2.1.4), wird das Layout in Bereiche eingeteilt, welche durch die Ventile voneinander getrennt werden. Jedem Bereich werden entsprechende Ventile zugeordnet. Das Ziel des Leitungsmanagements ist, dass Ventile, welche von einem Leitungsbereich «besetzt» wurden, nicht durch eine angrenzende Leitung verwendet werden können. Bezogen auf die Rezeptsteuerung bedeutet dies, dass ein Rezeptschritt erst ausgelöst werden darf, wenn die notwendigen Leitungen frei sind. Um dies sicherzustellen wird für jeden Rezeptschritt ein definierter Ablauf durchgeführt. Dabei wird in einem ersten Schritt der Leitungszustand geprüft. Falls die entsprechende Leitung frei ist, wird diese besetzt und der Ablauf kann zum nächsten Schritt übergehen. Dabei wird das entsprechende Rezept-Control-Word beschrieben. Der Ablauf wartet anschliessend auf die Rückmeldung des Rezept-Status-Words, dass der Rezeptschritt abgeschlossen werden kann. Das Rezept-Control-Word wird dann wieder zurückgesetzt und steht einem nächsten Rezeptschritt zur Verfügung.

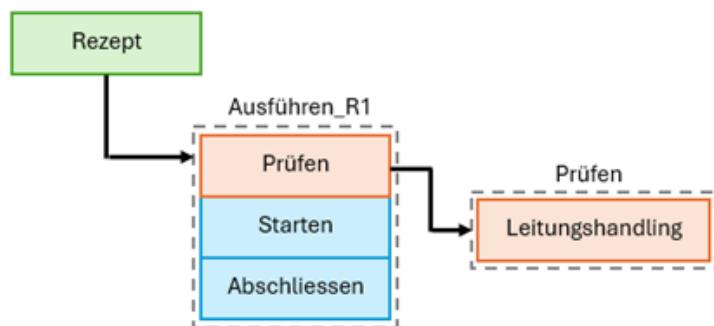


Abbildung 4.8: Ablauf einer Leitungsprüfung

Alle Leitungsbereiche werden durch ein Leitungs-Control-Word überwacht:

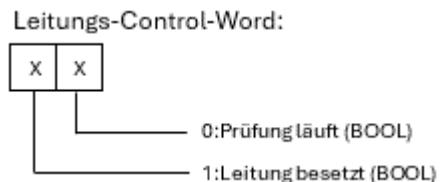


Abbildung 4.9: Entwurf des Leitungs-Control-Words

Die erste Stelle gibt an, ob sich die Leitungsbereiche in Prüfung befinden. Diese Position ist für das korrekte Management wichtig und hilft Doppelbelegungen zu vermeiden. Wenn ein Rezeptschritt die Leitungen prüft, wird diese Position auf 1 gesetzt. Die Leitungen können nun nicht durch einen zweiten Rezeptschritt parallel geprüft/besetzt werden. Die zweite Position zeigt an, ob die Leitung besetzt wurde oder frei ist. Jeder Leitungsbereich besitzt eine eigene Stelle.

Das Leitungsmanagement wird anhand eines einfachen Beispiels erklärt. Das System besteht dabei aus 4 Reaktoren, welche mittels zwei Leitungsbereichen (A und B) miteinander verbunden sind.

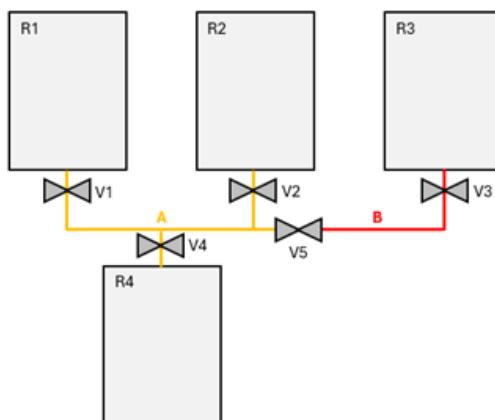


Abbildung 4.10: Beispielsituation für Leitungsmanagement

Im ersten Schritt wird der Rezeptschritt gestartet. Hierbei wird lokal (nur auf der Gesamtsystemebene) das Rezept-Control-Word definiert. Im Beispiel wird das Rezept-Control-Word des Reaktor 1 beschrieben. Als Sekundärreaktor wurde Reaktor 4 definiert. Es sollen 100 ml entleert werden.

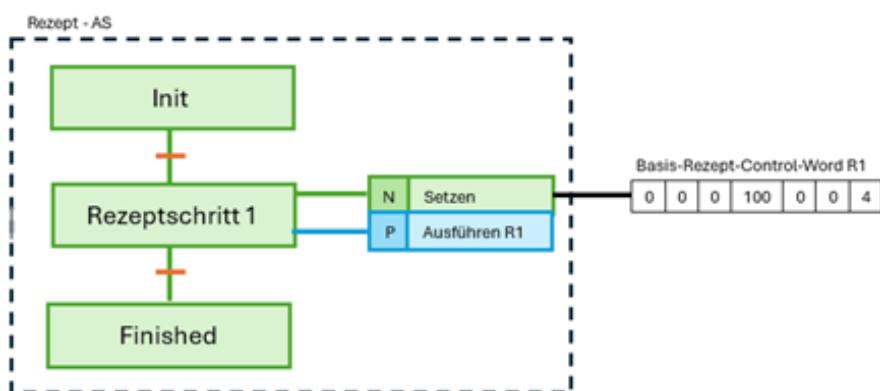


Abbildung 4.11: Leitungsmanagement auf Rezeptstufe

Im nächsten Schritt wird der Ablauf «Ausführen R1» aufgerufen. Dieser Ablauf ist für die Vorbereitung, Ausführung und Abschliessung des Rezeptschrittes verantwortlich. Neben dem Prüfen der Leitungen wird hier auch die Kommunikationsschnittstelle mit OPC-UA aufgebaut. Nach erfolgreicher Prüfung und Besetzung der entsprechenden Leitung wird das definiert Rezept-Control-Word über OPC-UA an die Reaktorebene weitergegeben. Der eigentliche Bearbeitungsprozess wird nun gestartet. Der Ablauf wartet anschliessend auf das Ende des Prozesses. Der Rezeptschritt wird dann abgeschlossen, indem das Rezept-Control-Word zurückgesetzt wird.

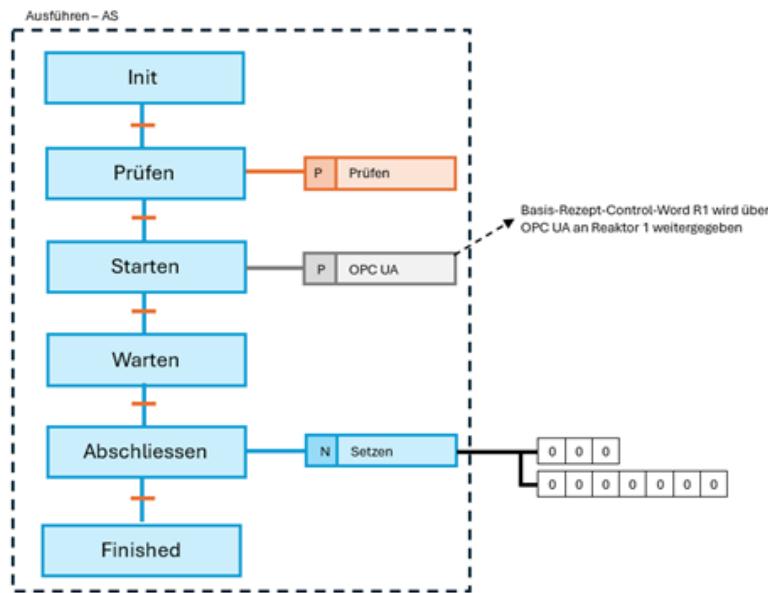


Abbildung 4.12: Schematischer Ausführungsprozess des Rezeptes

Der Prüfablauf gibt in einem ersten Schritt das Signal an das System, dass die Leitungen geprüft werden. Dafür wird die erste Position des Leitungs-Control-Word auf 1 gesetzt. Der Ablauf prüft, welche Leitungen benötigt werden und vergleicht diese mit den aktuell besetzten Leitungen. Wie dieser Prozess genau aussieht, wird im Moment nicht weiter erläutert. Bei der Umsetzung des Programms wird dieser Schritt näher beschrieben. Der Ablauf besetzt nun die entsprechenden Leitungen oder wartet darauf, dass diese frei werden. Beim Besetzen wird das Leitungs-Control-Word an der jeweiligen Position mit einer 1 versehen. Nach der Prüfung wird die erste Position auf 0 zurückgesetzt.

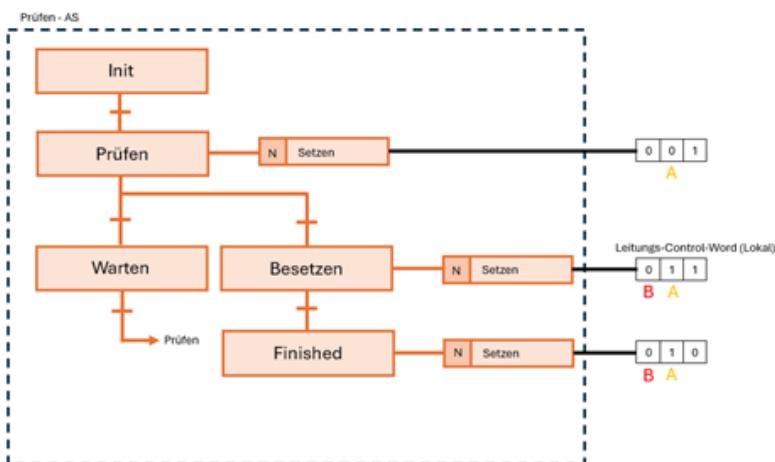


Abbildung 4.13: Schematischer Prüfprozess

4.2 Kommunikation über OPC-UA

Die Kommunikation der chemischen Reaktoranlage wird via OPC-UA realisiert. Es wird eine vertikale Kommunikation zwischen Gesamtsystemebene und Reaktoreinheitsebene aufgebaut. Dafür wird das TwinCat 3 OPC UA Paket (TF6100) von Beckhoff eingesetzt. Innerhalb des Paketes gibt es verschiedene Komponenten, welche für eine OPC-UA-Kommunikation eingesetzt werden können:

OPC-UA-Server:

Der Server stellt die standardisierte Kommunikationsschnittstelle bereit. Das bedeutet, dass der Server die Daten und Methoden zur Verfügung stellt, auf welche der Client zugreifen kann. In der Regel wird eine Client-To-Server-Kommunikation verwendet. Grundsätzlich ist auch eine Server-To-Server-Kommunikation möglich, diese müsste über ein Gateway realisiert werden. Es wurden keine Referenzen gefunden, dass eine direkte Server-To-Server-Kommunikation mit TF6100 realisierbar ist. Ein OPC-UA-Server bietet folgende Funktionen, welche für das Projekt relevant sein könnten [9]:

Data Access (DA):	Eine Funktion, welche die Verwendung und Darstellung von Variablenwerten beschreibt. Dies ermöglicht den Zugang auf aktuellen Prozessdaten. Die definierten Variablen können vom Client gelesen wie auch beschrieben werden.
Historical Access (HA):	Mit dieser Funktion können Variablenwerte in einem Speicher (Datei oder Datenbank) abgelegt werden. Der Client hat die Möglichkeit auf diese Daten zuzugreifen und diese auszulesen.
Alarms and Conditions (AC):	Alarm and Conditions beschreibt ein Modell zur Überwachung von Symbolwerten und das Auslösen von Alarmen und Ereignissen, falls definierte Schwellen überschritten werden.
Methodenaufrufe:	Der Client kann über den Server auf definierte SPS-Methoden zugreifen. Dafür stehen zwei Konzepte zur Verfügung, die RPC- und Job-Methode. In der Regel wird mit der Job-Methode gearbeitet. Hierbei wird die OPC-UA-Methode über einen SPS-Funktionsbaustein repräsentiert und wird über mehrere SPS-Zyklen ausgeführt.

Welche dieser Funktionen sinnvoll für die Applikation ist, wird sich bei der Umsetzung der Kommunikation in TwinCat zeigen. Eine vollständige Auflistung der Funktionen kann aus dem entsprechendem Beckhoff-Handbuch entnommen werden.

OPC-UA-Client:

Der Client ist das Gegenstück des Servers. Wie bereits erwähnt, kann dieser auf die vom Server zur Verfügung gestellten Daten und Methoden zugreifen. Daten können durch den Client gelesen oder beschrieben werden. Eine detailliertere Übersicht über die Interaktion zwischen Server und Client wird bei der Kommunikationstopologie dargestellt. Ein OPC-UA-Client bietet grundsätzlich zwei Möglichkeiten mit einem OPC-UA-Server zu kommunizieren. Der Client kann als ein I/O-Gerät aufgebaut werden. Hierbei wird die Kommunikationsschnittstelle wie eine Hardwareschnittstelle definiert, um Variablen und Methoden aufrufen zu können. Die Verbindung zum Server kann auch über PLCopen genormte Funktionsbausteine aufgebaut werden. Hierbei kann die Verbindung direkt aus der SPS-Logik realisiert werden. Der Fokus dieser Verbindung liegt jedoch auf Daten und Metho-

den. Funktionen wie das Auslesen von historischen Daten oder das Alarmhandling können über eine Kommunikation mittels I/O-Gerät besser umgesetzt werden. Da die PLCopen-Funktionsbausteine direkt in die SPS-Logik integriert werden, muss die Integration dieser Bausteine gut geplant werden, so dass diese in die Gesamtfunktion des Systems passt [10].

OPC-UA-Gateway:

Ein Gateway wird hauptsächlich dafür eingesetzt verschiedene Protokolle und Systeme miteinander zu kombinieren, so dass diese trotzdem Daten und Befehle austauschen können. Daten von mehreren Quellen können auf dem Gateway auch kombiniert werden, um einen einheitlichen Zugriff zu gewährleisten. Die Konfiguration und korrekte Einrichtung eines OPC-UA-Gateway kann komplex und zeitintensiv sein. In einem ersten Schritt wird die Kommunikation mit einer reinen Client-Server-Kommunikation aufgebaut [11].

OPC-UA-Configurator:

Ist eine Benutzeroberfläche für die Konfiguration des OPC-UA-Servers. Der Konfigurator kann über die Visual-Studio-Umgebung ausgeführt werden oder über eine Standalone-Variante [12].



Die Beckhoff-Handbücher zu TF6100 werden in Anhang beigelegt

4.2.1 Topologie der Kommunikation

Die Topologie der Kommunikation kann in zwei Bereiche aufgeteilt werden. Auf der Gesamtsystemebene wird ein OPC-UA-Client definiert. Der Client ist in die TwinCat-Applikation integriert und kann über die ADS-Schnittstelle (interne Kommunikationsschnittstelle) mit dem Programm interagieren. Innerhalb der Reaktorebene wird für jede Reaktoreinheit ein OPC-UA-Server definiert. Dafür kann direkt im TwinCat-Projekt ein entsprechendes Server-Projekt angelegt werden. Auch hier findet ein Austausch mit der TwinCat-Applikation statt. Über die OPC-UA-Kommunikationsschnittstelle gibt der Server vor, welche Variablen und Methoden dem Client zur Verfügung stehen. Der Client kann nun über die Schnittstelle auf die definierten Daten zugreifen, um diese zu lesen oder zu beschreiben. Jeder Server definiert eine eigene OPC-UA-Kommunikationsschnittstelle für den Client.

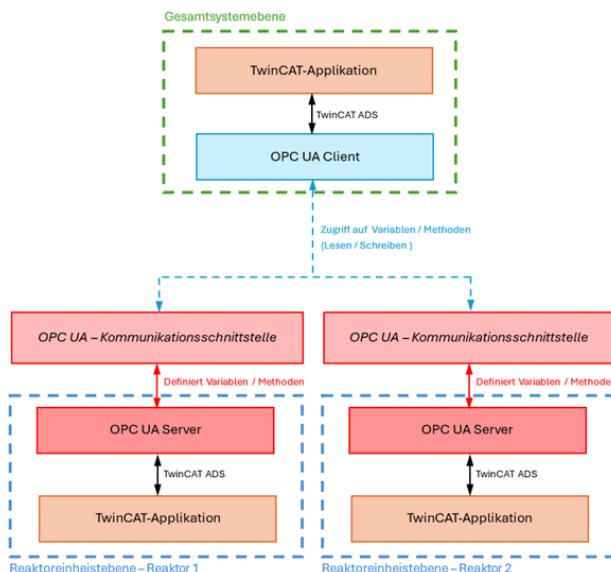


Abbildung 4.14: OPC-UA-Topologie

4.3 Grundstruktur des Programms

Als Literaturreferenz wird das Buch «Speicherprogrammierbare Steuerungen in der Industrie 4.0» von Matthias Seitz verwendet. Die objektorientierte Programmierung wird hierbei detailliert und ausführlich beschrieben mit konkreten Beispielen für Codesys. Die im Buch behandelten Systeme lassen sich jedoch nicht direkt mit der chemischen Reaktoranlage vergleichen. Der Hauptunterschied ist das Verwenden von mehreren IPC's, welche eigenständige Systeme bilden. Diese kommunizieren über eine OPC-UA-Schnittstelle miteinander. Im Buch werden die Programme auf einem geschlossenen System mit einem IPC umgesetzt [6].

4.3.1 Aufbau des Programms

Ein wichtiges Element der Software ist der objektorientierte Aufbau. Dies bringt bei größeren und komplexeren Softwareprojekten verschiedene Vorteile mit sich. Die Programmierung gestaltet sich effizienter, übersichtlicher und flexibler. Um diese Vorteile auszunutzen zu können, müssen nicht nur die Objektklassen objektorientiert aufgebaut werden. Alle Aspekte der Software müssen darauf ausgelegt werden. Damit die Software der chemischen Reaktoranlage objektorientiert aufgebaut werden kann, wird in einem ersten Schritt die angewandte Struktur im Buch analysiert. Sobald diese verstanden wurde, können die entsprechenden Anpassungen für die Reaktoranlage definiert werden. Im Buch wurde folgender Aufbau umgesetzt.

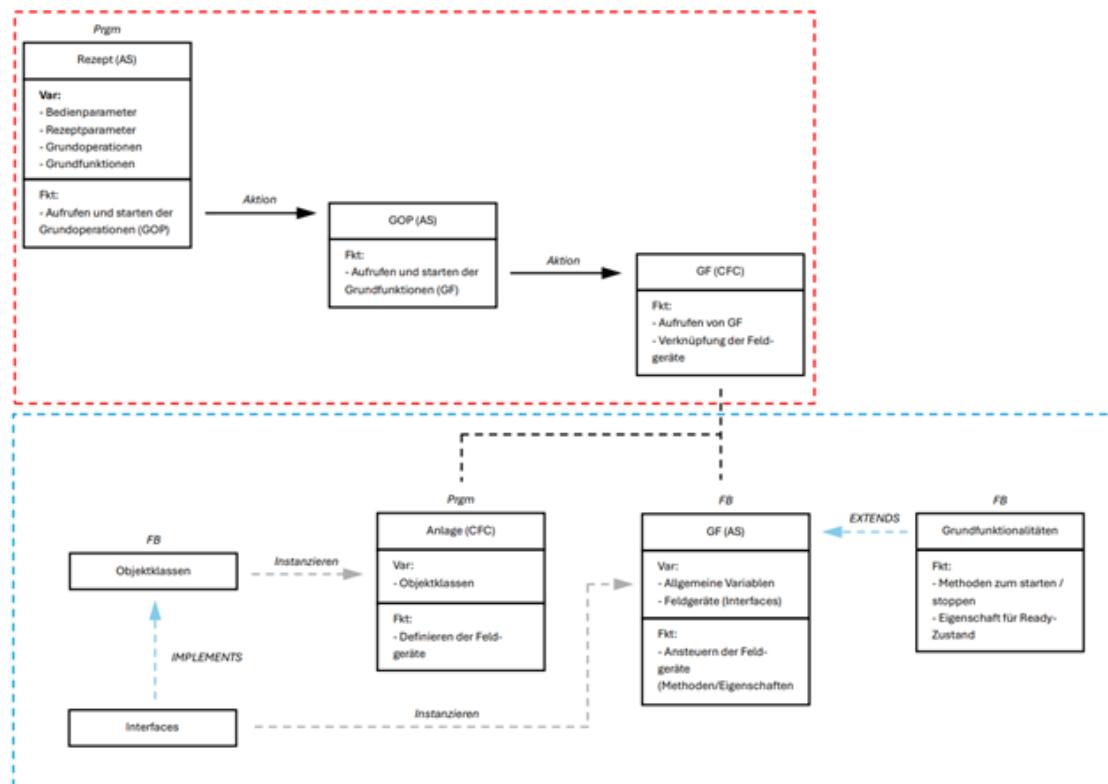


Abbildung 4.15: Programmstruktur aus Literatur

Der Aufbau lässt sich in zwei Bereiche aufteilen. Der rote Bereich setzt die Ablauflogik des Systems um. Das erste Element ist das Rezept, welches in Ablaufsprache programmiert wird und ein Programm darstellt. Es werden verschiedene Variablen und Funktionsbausteine instanziert.

Bedienparameter:	Die Bedienparameter werden verwendet, um Schritte im Rezept auszulösen.
Rezeptparameter:	Rezeptparameter sind prozessrelevante Parameter. Diese werden innerhalb der Grundfunktionen verwendet, um einen definierten Prozess auszuführen.
Grundoperationen/ Grundfunktionen:	Die Grundoperationen und Grundfunktionen bilden die Bestandteile des Rezeptes. Diese werden innerhalb der Schritte aufgerufen und gestartet.

Das Rezept ruft eine definierte Grundoperation mittels einer Aktion auf und startet diese. Die Grundoperation wird auch in Ablausprache umgesetzt. In den jeweiligen Schritten werden die Grundfunktionen mit einer Aktion aufgerufen und gestartet. Die Grundfunktionen wurden als Funktionsbaustein angelegt. Innerhalb der Aktion werden die jeweiligen Feldgeräte oder die Prozessparameter mit den Eingängen der Grundfunktionsbausteine verknüpft.

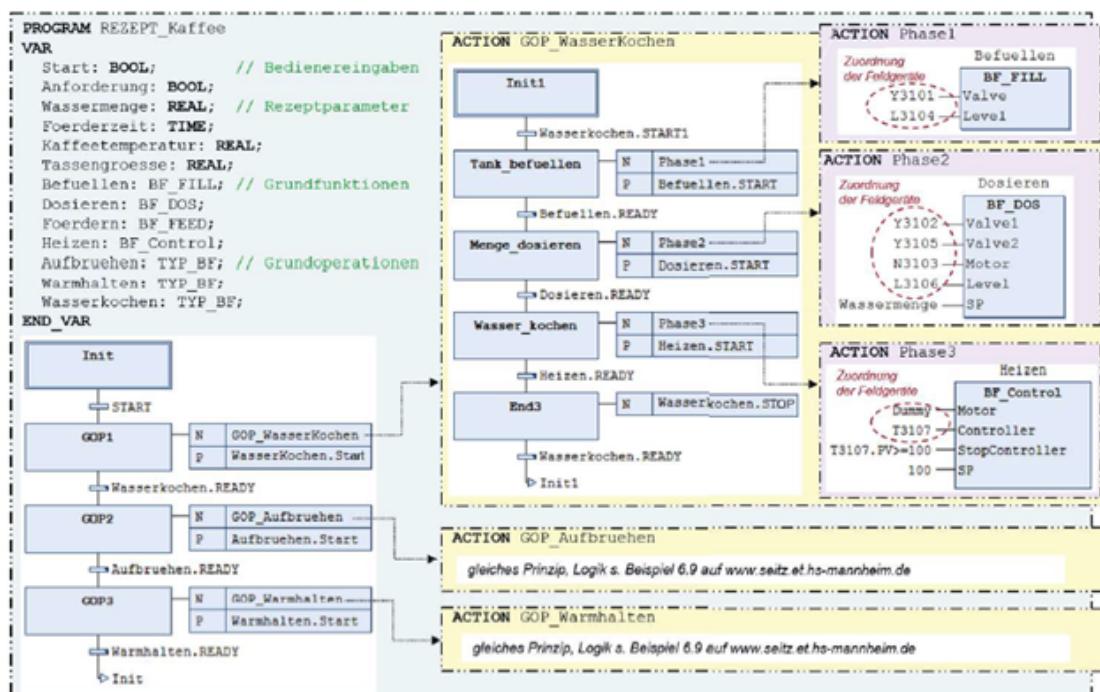


Abbildung 4.16: Beispiel einer Programmstruktur

Im blauen Bereich werden die Grundfunktionen als Abläufe (programmiert in Ablaufsprache) abgebildet. Diese werden wie bereits erwähnt als Funktionsblöcke definiert. Als Eingangsvariablen werden die benötigten Feldgeräte instanziert. Dafür müssen die Schnittstellen (Interfaces) verwendet werden. Die Schnittstelle definiert, welche Methoden und Eigenschaften eine Objektklasse eines Feldgerättyps haben muss. Die so instanziierten Feldgeräte können für den Schrittablauf verwendet werden. Die Grundfunktionen werden dadurch anlagenunabhängig. Über eine Vererbung werden Basisfunktionalitäten an die Grundfunktionen weitergegeben. Dabei handelt es sich um Methoden zum Starten / Stoppen und der Eigenschaft für die Zustandsabfrage. Die Vererbung ist ein einfaches Mittel, um Funktionalitäten in verschiedenen Grundfunktionen zu implementieren. Wenn der definiert Funktionsbaustein mit der Definition «EXTENDS» vererbt wird, kann auf dessen Funktionalität zugegriffen werden. Innerhalb des blauen Bereiches wird auch die Anlage definiert. Die Anlage ist das CFC-Programm, in welche die vorhandenen Feldgeräte im System, mit den Objektklassen instanziert werden. Wichtig ist, dass die Objektklassen

die Schnittstellen (Interfaces) implementieren. Mit der Implementierung weiss das Programm, welche Methoden und Eigenschaften die Objektklasse haben muss.

Der Vorteil dieser Struktur ist die Anlagenunabhängigkeit. Das Rezept kann ohne Definition der Feldgeräte aufgebaut werden. Das Rezept kann für unterschiedliche Anlagen verwendet werden. Dies ist ein relevanter Aspekt für die chemische Reaktoranlage. Das Rezept auf der Gesamtsystemebene kann für alle Reaktoren, ohne Anpassungen, verwendet werden. Die Zuweisung der Feldgeräte wird auf der Stufe der Grundfunktionen umgesetzt und befinden sich auf der Reaktoreinheitsebene. Der im Buch verwendete Aufbau muss dafür angepasst werden, da eine Kommunikationsschnittstelle über OPC UA vorhanden ist.

Der angepasste Aufbau gliedert sich in den Bereich der Gesamtsystemebene (rot) und der Reaktoreinheitsebene (blau). Dazwischen liegt die OPC UA – Kommunikationsschnittstelle.

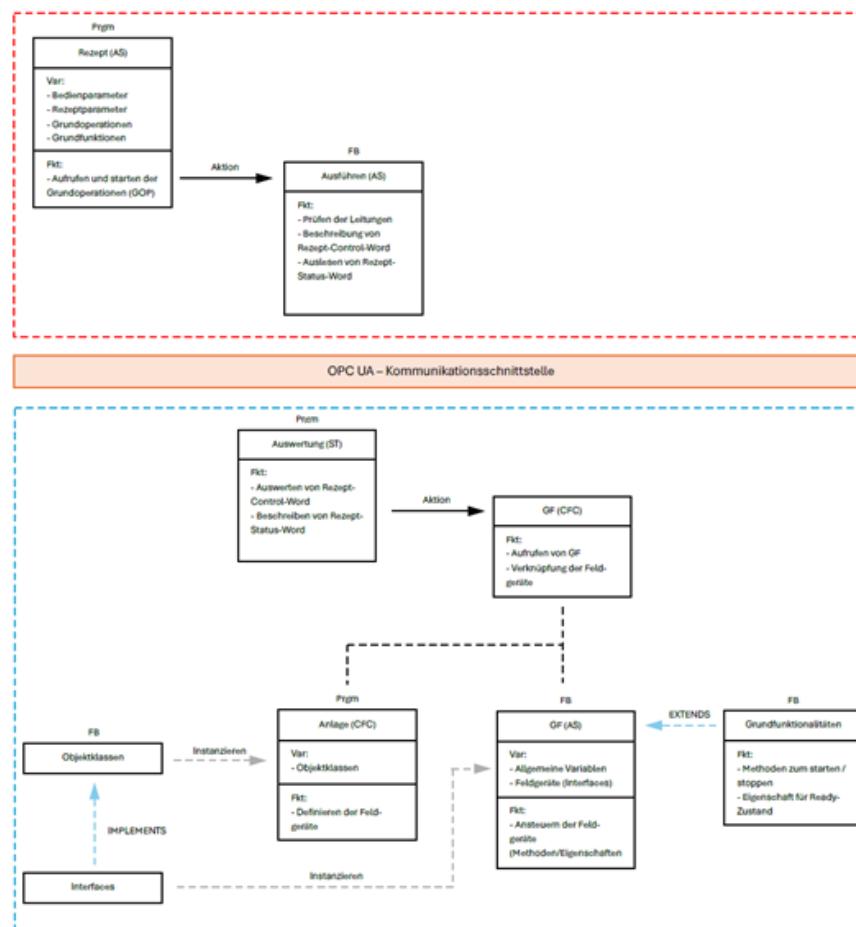


Abbildung 4.17: Angepasste Grundstruktur

Die Gesamtsystemebene ist für die Koordinierung der Abläufe zuständig. Wie bei der Referenzstruktur aus der Literatur gibt es ein Rezept. Dieses ruft jedoch nicht direkt eine Grundoperation oder Grundfunktion auf. Stattdessen wird ein Funktionsblock aufgerufen, welcher für die Ausführung des Rezeptschrittes zuständig ist (wie in Kapitel 4.1.4 beschrieben). Hier findet die Interaktion mit der OPC-UA-Schnittstelle statt. Die Gesamtsystemebene hat keine Information über die instanzierten Feldgeräte oder die Grundfunktionen. Diese sind Teil der Reaktoreinheitsebene. Die Struktur innerhalb dieser Ebene ähnelt sehr stark der Referenzstruktur. Als Ergänzung kommt ein Programm dazu, welches die OPC-UA-Schnittstelle auswertet und entsprechend über Aktionen die Grundfunktionen auslöst.

4.3.2 Wichtige Definitionen für Funktionsbausteine

Handling von Störungen

Ein Kernaspekt einer solchen Anlage ist das Handling von Störungen und das Zurücksetzen dieser. Wie reagiert das System, wenn bei einem Feldgerät eine Störung auftritt.

Jeder Funktionsbaustein der Feldgeräte kann mit einer BOOL-Variable angeben, ob sich dieser in einem Störungszustand befindet. Über eine INT-Variable kann die Art der Störung definiert werden. Eine Störung wirkt sich nur auf das betroffene Element aus und nicht auf die gesamte Anlage oder die Reaktoreinheit.

Es gibt zwei Kategorien von Störungen. Die leichte Störung (Soft) führt zu einem Warnhinweis. Eine harte Störung (Hard) führt zum Stop des Bauteils. Sobald eine Störung auftritt, muss diese zurückgesetzt werden. Im Moment ist nicht abschliessend definiert, ob dies auf der Reaktoreinheit- oder Gesamtsystemebene stattfindet. Sicher ist jedoch, dass beim Auftreten einer Störung nicht jedes Bauteil einzeln zurückgesetzt werden muss.

Durch die INT-Variable wird nicht nur die Art der Störung angegeben, sondern auch um welches Element und welchen Reaktor es sich handelt. Die Störung-ID wird durch das System ausgewertet und die entsprechende Meldung oder Reaktion wird ausgeführt. Wie dieses Handling der Störungs-ID aussieht, wird zu einem späteren Zeitpunkt definiert. Wie unter Kapitel 4.2 aufgelistet wurde, besitzt der OPC-UA-Server eine «Alarms and Conditions (A/C)» Funktion. Diese könnte möglicherweise für die Auswertung der Störungs-ID verwendet werden.

Die Störungs-ID wird wie folgt definiert und es gibt folgende Störungen:

Kat.	Reaktor	Element	ID	Beschreibung
S	X	XX	01	Überschreitung eines Schwellenwertes
S	X	XX	02	Bauteil nicht eingeschalten
S	X	XX	03	Bauteil nicht angeschlossen
S	X	XX	04	Verbindung zu Bauteil wurde getrennt
S	X	XX	05	Heiz-Kühlprozess ist noch aktive

Tabelle 4.9: Definierte Störungen für Funktionsbausteine

Die Liste zeigt nur die im Moment identifizierten Störungen an und stellt nicht eine vollständige Aufführung dieser dar. Im Verlauf der Entwicklung wird diese Liste angepasst und ergänzt.

Handling von Objektmodi

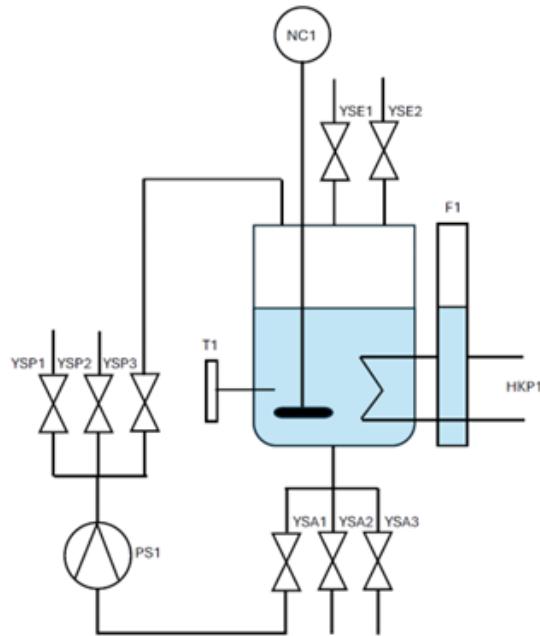
Ein weiterer übergreifender Aspekt ist das Definieren des Modus, in welchem sich das System und der Funktionsbaustein gerade befinden. Dafür wird für jeden Funktionsbaustein eine INT-Input-Variable definiert. Die Zahl steht dabei für den aktuellen Zustand.

INT-Variable	Bedeutung
0	Befindet sich im ausgeschalteten Zustand
1	Befindet sich im Manuel-Modus
2	Befindet sich im Automatik-Modus

Tabelle 4.10: Definierte Modi für Funktionsbausteine

Definierung der Feldgerätbezeichnung

Die Grundfunktionen greifen auf verschiedene Feldgeräte zu, sei es bei der Instanziierung der Schnittstellen oder bei der Zuweisung. Um Fehler zu verhindern, werden die Bezeichnungen der Feldgeräte fix definiert. Für das Programm werden folgende Bezeichnungen verwendet:



Feldgerätetyp	Bezeichnung für Schnittstellen	Bezeichnung für Objektklassen
Magnetventil	Ventil	YSE1, YSE2, YSA1, YSA2, YSA3, YSP1, YSP2, YSP3
Rührer	Motor	NC1
Pumpe	Pumpe	PS1
HK-System	HKSystem	HKP1
Temperatursensor	Temp	T1
Füllstandsensor	Level	F1

Abbildung 4.18: Definierung der Feldgerätbezeichnung

5 Umsetzung der Software

5.1 Information über aktuellen Stand

Die Umsetzung der Software war ein zeitaufwändiger Prozess mit vielen Teilschritten und Meilensteinen. Die Dokumentation fokussiert sich auf den momentanen Stand der Software, wie diese funktioniert und wie der Aufbau definiert wurde. Das Ziel ist das Verständnis der Software, um weiterführende Arbeiten zu erleichtern. Das Vorgehen für die Erarbeitung wurde wie folgt definiert:

- Schritt 1: Erstellen der Funktionsbausteine der Feldgeräte
- Schritt 2: Testen der definierten Funktionsbausteine
- Schritt 3: Erstellen der Grundfunktionen
- Schritt 4: Testen der Grundfunktionen
- Schritt 5: Aufbau der Gesamtsystemebene
- Schritt 6: Testen des Ablaufs auf Gesamtsystemebenen
- Schritt 7: Einarbeiten in OPC-UA-Paket von Beckhoff (TF6100)
- Schritt 8: Umsetzen der OPC-UA-Schnittstelle (Client/Server)
- Schritt 9: Programmoptimierung

Während der Erarbeitung dieser Teilschritte sind viele Probleme und Hürden aufgetreten. Jeder Teilaspekt, wie die objektorientierte Struktur oder die OPC-UA-Kommunikation, kam mit eigenen Herausforderungen. Eine umfassende Beschreibung dieser würde den Rahmen dieser Dokumentation sprengen. Aus diesem Grund ist die Erarbeitung nicht Teil der Dokumentation.

Die Software ist im Moment soweit, dass über ein HMI eine OPC-UA-Verbindung mit den Servern aufgebaut werden kann. Der Status dieser Verbindung kann überprüft werden. Die Verbindung kann nach Belieben auch getrennt werden. Ein vordefiniertes Rezept wird über einen Button gestartet. Innerhalb der Gesamtsystemebene wird das Rezept-Control-Word beschrieben und über einen PLCopen-Funktionsbaustein an die OPC-UA-Schnittstelle übertragen. Der Server (Reaktoreinheitsebene) erkennt die Veränderung des Rezept-Control-Words und startet die entsprechende Grundfunktion. Während dieser Zeit beschreibt der Server das Rezept-Status-Word mit der Information über den aktuellen Status. Der Client (Gesamtsystemebene) liest dieses über den entsprechenden PLCopen-Funktionsbaustein aus und kann auf die verschiedenen Situationen reagieren.

Die Software erfüllt im Moment die grundlegende Funktion, welche für den Betrieb des Systems nötig ist. Auf diesem Gerüst kann in zukünftigen Arbeiten aufgebaut werden. Viele geplante Funktionen konnten leider aus zeitlichen Einschränkungen nicht umgesetzt werden. Die Priorität war in einem ersten Schritt eine funktionierende Grundlage zu schaffen. Die offenen Punkte und weiterführenden Arbeiten werden im Kapitel 6.3 beschrieben.

5.2 Arbeitsweise der Software

Der wichtigste Aspekt der Software ist die Interaktion zwischen Server (Reaktoreinheit) und Client (Gesamtsystemebene). Dafür wird das bereits erwähnte Rezept-Status/ Control-Word verwendet. Alle rezeptrelevanten Informationen werden über das Rezept-Control-Word an den definierten Reaktor weitergegeben. Der aktuelle Status des Reaktors wird über das Rezept-Status-Word an die Gesamtsystemebene weitergegeben. Diese Interaktion findet nur zwischen dem definierten Reaktor und der Gesamtsystemebene statt. Grundfunktionen wie das Entleeren und das Pumpen benötigen eine Interaktion zwischen zwei Reaktoren.

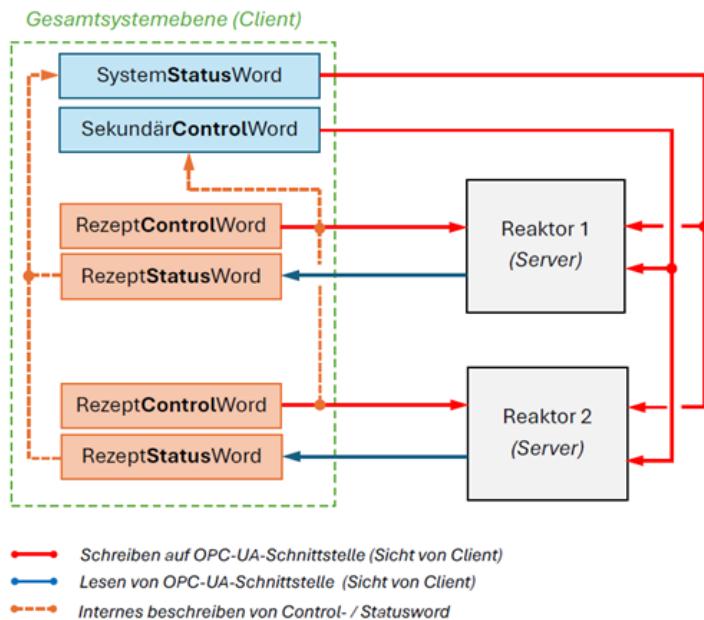


Abbildung 5.1: Interaktionsschnittstellen der Software

Diese Interaktion wird auch über den Client gehandelt. Dafür wurden zwei neue Kommunikationsvariablen definiert. Diese Variablen stehen allen Reaktoreinheiten zur Verfügung. Das Sekundär-Control-Word gibt an, welcher Reaktor als Sekundärreaktor (Empfänger) für einen Entleer- oder Pumpprozess definiert wurde. Diese Information wird anhand des Rezept-Control-Words definiert.

Das System-Status-Word zeigt in welchem Zustand (Ready oder Activ) sich jeder Reaktor befindet. Der Primärreaktor erkennt dadurch, dass der Sekundärreaktor bereit für die Übergabe des Mediums ist. Das System-Status-Word übernimmt die Informationen aus den jeweiligen Rezept-Status-Words. Die genaue Definierung aller Kommunikationsvariablen wird innerhalb der Beschreibung der technischen Umsetzung gezeigt.

Die Gesamtsystemebene und Reaktoreinheitsebenen besitzen zwei unterschiedliche Applikationen. Diese unterscheiden sich nicht nur in ihrer Funktionen, sondern auch in Ihrer Struktur. Die einzige Schnittstelle zwischen den Applikationen sind die oben genannten Kommunikationsvariablen, welche als globale Variablen definiert wurden. Für die folgenden Erklärungen wird von Programmen und Funktionsblöcken gesprochen. Dabei ist der Programmtyp innerhalb der Programmorganisationseinheit (POU) gemeint. Programme werden als blaue Felder dargestellt, Funktionsblöcke als grüne.

Die folgenden Schemen zeigen die grobe Arbeitsweise der jeweiligen Applikation. Es werden nur die Elemente abgebildet, welche für die Grundfunktion benötigt werden. Die einzelnen Elemente werden bei der Beschreibung der technischen Umsetzung ausführlich beschrieben.

Die Interaktion mit der OPC-UA-Schnittstelle bei der Gesamtsystem-Applikation wird über die instanzierten Funktionsblöcke «Reaktoreinheit» und «Auswertung» durchgeführt. Die Funktionsblöcke besitzen Methoden, mit welchen der Schreib- oder Lese-Prozess gestartet werden kann. Die Interaktion wird mit PLCopen-Funktionsbausteinen umgesetzt, welche innerhalb von TF6100 zur Verfügung stehen. Die «Reaktoreinheit» ist verantwortlich für das Handling des Rezept-Control/Status-Words, die «Auswertung» für das System-Status-Word und Sekundär-Control-Word.

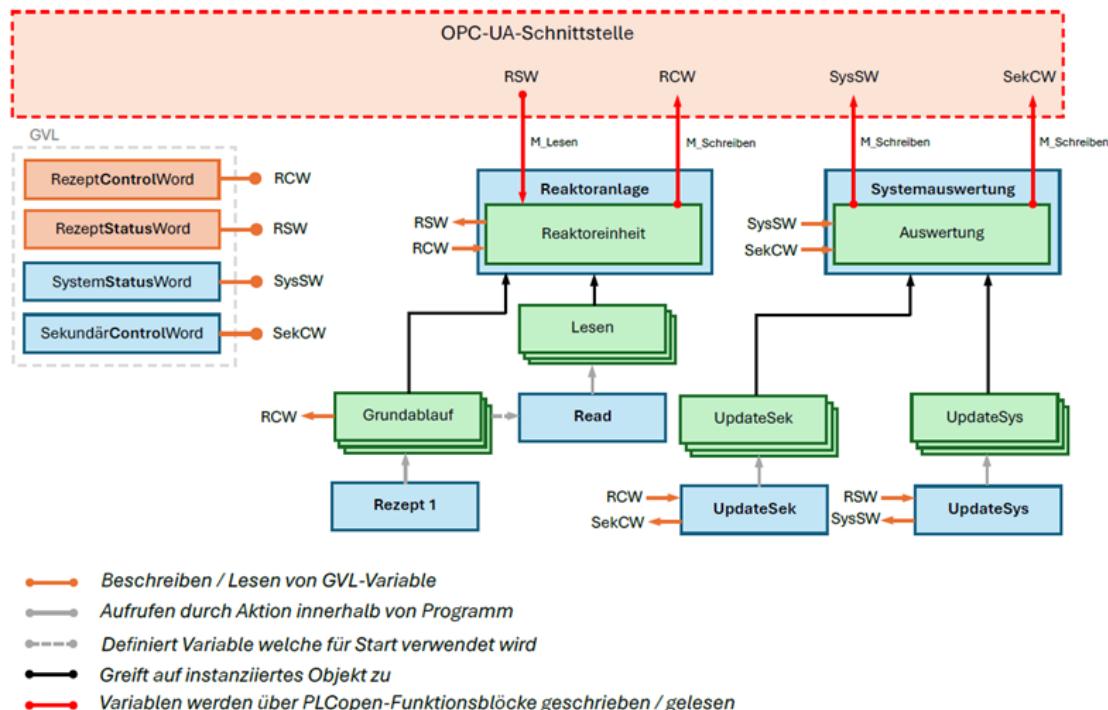


Abbildung 5.2: Arbeitsweise der Gesamtsystem-Applikation

Der Grundablauf ist ein Funktionsblock, welcher den Prozessablauf steuert. Innerhalb des Ablaufs wird das Rezept-Control-Word beschrieben. Der Ablauf wird durch das Rezept-Programm gestartet. Der Funktionsblock «Lesen» liest innerhalb des Ablaufs das Rezept-Status-Word. Das Programm «Read» startet den Funktionsblock «Lesen» für jede definierte Reaktoreinheit. Parallel laufen die Programme «UpdateSek» und «UpdateSys». Diese starten die jeweiligen Funktionsblöcke zur Beschreibung des System-Status-Words und Sekundär-Control-Word. Das Rezept-Status-Word und Rezept-Control-Word werden dabei ausgewertet und starten bei bestimmten Angaben den Ablauf.

Bei der Reaktoreinheit-Applikation müssen die Kommunikationsvariablen nicht über PLCopen-Funktionsbausteine gelesen oder beschrieben werden. Über Attribute können die globalen Variablen für die OPC-UA-Schnittstelle freigegeben werden. Diese können in Echtzeit durch die Applikation beschrieben und gelesen werden. Die Auswertung findet in der «Systemauswertung» statt. Im Funktionsblock «Auswertung» wird das Rezept-Control-Word, System-Status-Word und Sekundär-Control-Word ausgewertet, um die entsprechende Grundfunktion zu starten. Die Grundfunktionen sind als Funktionsblöcke definiert. Die Grundfunktionen greifen anschliessend auf die instanzierten Feldgeräte im Programm «Reaktoranlage» zu. Innerhalb der Grundfunktion wird das Rezept-Status-Word beschrieben.

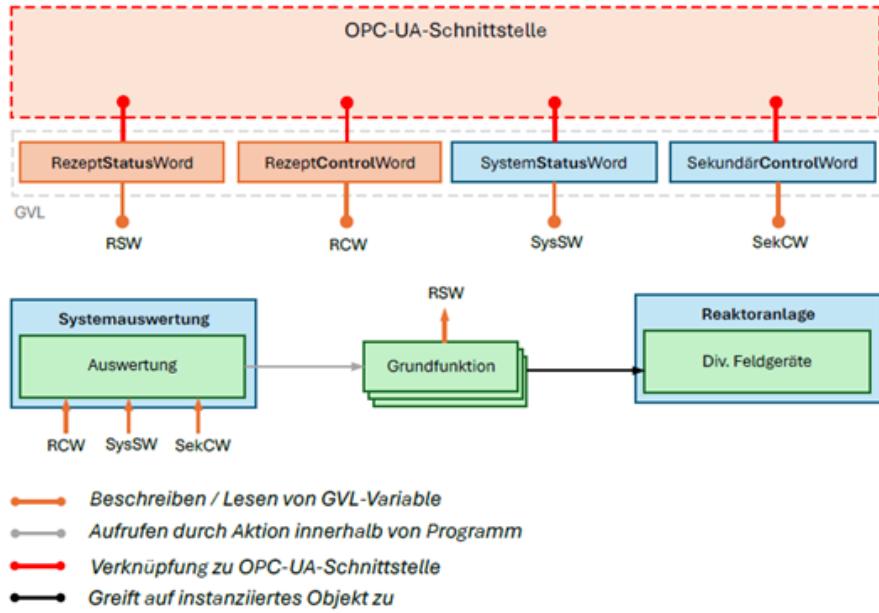


Abbildung 5.3: Arbeitsweise der Reaktoreinheit-Applikation

Der gesamte Prozess wird im Moment über ein einfaches HMI in der Gesamtsystem-Applikation gestartet. Die Reaktoreinheit-Applikation wertet die Kommunikationsvariablen nach dem Start aus und befindet sich im Automatik-Modus. Es gibt im Moment kein HMI, welches zur Steuerung genutzt werden kann. Die Applikation wurde aber darauf ausgelegt, dass Funktionalitäten wie ein Manuel-Modus umgesetzt werden können. Alle Objektklassen der Feldgeräte können bei entsprechender Einstellung auch im Manuel-Modus betrieben werden.

5.3 Technische Umsetzung

5.3.1 Status- und Control-Word

Während der Erarbeitung der Software wurden viele Erfahrungen gesammelt, anhand welcher das Rezept-Status/Control-Word angepasst werden musste. Das Rezept-Control-Word wurde mit einer Position für das Einschalten der Reaktoreinheit ergänzt wie auch mit Startpositionen für Prozesse (z.B. Zirkulieren). Zusätzlich mussten die TIME-Datentypen mit INT ersetzt werden, da es bei der Kommunikation über OPC-UA zu Problemen kam und die Verbindung nicht aufgebaut werden konnte. Die INT-Variablen dient als ein Multiplikator für eine TIME-Angabe in der Reaktoreinheit-Applikation.

Das Rezept-Status-Word gibt nun keine Angabe mehr über den Sekundärreaktor an. Dafür werden Informationen zur Reaktoreinheit angegeben (SysReady und SysActiv). Jede Reaktoreinheit besitzt aber immer noch ein eigenes Rezept-Status/Control-Word.

Rezept-Control-Word			
Pos	Bezeichnung	Datatype	Beschreibung
0	PowerOn	BOOL	Einschalten der Reaktoreinheit (Feldgeräte)
1	SekReakt	INT	Definieren des Sekundärreaktors (Empfänger)
2	Leitung	INT	Definieren welche Leitung verwendet werden soll (Von Applikation definiert)
3	Temp	REAL	Definieren der Temperatur für den Heiz- / Kühlprozess
4	Entleermenge	REAL	Definieren der zu entleerenden Menge
5	Pumpmenge	REAL	Definieren der zu pumpenden Menge
6	Halitedauer	INT	Definieren der Halitedauer der Temperatur in Sekunden
7	Rührdauer	INT	Definieren der Rührdauer in Sekunden
8	Wartedauer	INT	Definieren der Wartedauer in Sekunden
9	Zirkulierdauer	INT	Definieren der Zirkulierdauer in Sekunden

Tabelle 5.1: Definierung von Rezept-Control-Word

Rezept-Status-Word			
Pos	Bezeichnung	Datatype	Beschreibung
0	SysReady	BOOL	Applikation läuft und ist bereit für Kommunikation
1	SysActiv	BOOL	System ist eingeschalten (Feldgeräte)
2	ReactReady	BOOL	Reaktor ist bereit für Prozess
3	ReactActiv	BOOL	Reaktor führt Prozess durch
4	GF1Activ	BOOL	Heiz- und Kühlprozess ist aktiv
5	GF2Activ	BOOL	Entleerprozess ist aktiv
6	GF3Activ	BOOL	Pumpprozess ist aktiv
7	GF4Activ	BOOL	Halteprozess ist aktiv
8	GF5Activ	BOOL	Rührprozess ist aktiv
9	GF6Activ	BOOL	Warteprozess ist aktiv
10	GF7Activ	BOOL	Befüllprozess ist aktiv
11	GF8Activ	BOOL	Zirkulierprozess ist aktiv

Tabelle 5.2: Definierung von Rezept-Status-Word

Damit eine indirekte Interaktion zwischen den Reaktoren stattfinden kann, wurden zwei neue Status- und Control-Words definiert. Alle Reaktoren können auf diese zwei Kommunikationsvariablen zugreifen und diese auslesen.

Sekundär-Control-Word			
Pos	Bezeichnung	Datatype	Beschreibung
0	R1	BOOL	Reaktor 1 wurde als Sekundärreaktor (Empfänger) definiert
1	R2	BOOL	Reaktor 2 wurde als Sekundärreaktor (Empfänger) definiert
2	R3	BOOL	Reaktor 3 wurde als Sekundärreaktor (Empfänger) definiert
3	R4	BOOL	Reaktor 4 wurde als Sekundärreaktor (Empfänger) definiert
4	R5	BOOL	Reaktor 5 wurde als Sekundärreaktor (Empfänger) definiert
5	R6	BOOL	Reaktor 6 wurde als Sekundärreaktor (Empfänger) definiert
6	R7	BOOL	Reaktor 7 wurde als Sekundärreaktor (Empfänger) definiert

Tabelle 5.3: Definierung von Sekundär-Control-Word

System-Status-Word			
Pos	Bezeichnung	Datentype	Beschreibung
0	R1_Ready	BOOL	Reaktor 1 ist bereit für Prozess
1	R1_Activ	BOOL	Reaktor 1 führt Prozess durch
2	R2_Ready	BOOL	Reaktor 2 ist bereit für Prozess
3	R2_Activ	BOOL	Reaktor 2 führt Prozess durch
4	R3_Ready	BOOL	Reaktor 3 ist bereit für Prozess
5	R3_Activ	BOOL	Reaktor 3 führt Prozess durch
6	R4_Ready	BOOL	Reaktor 4 ist bereit für Prozess
7	R4_Activ	BOOL	Reaktor 4 führt Prozess durch
8	R5_Ready	BOOL	Reaktor 5 ist bereit für Prozess
9	R5_Activ	BOOL	Reaktor 5 führt Prozess durch
10	R6_Ready	BOOL	Reaktor 6 ist bereit für Prozess
11	R6_Activ	BOOL	Reaktor 6 führt Prozess durch
12	R7_Ready	BOOL	Reaktor 7 ist bereit für Prozess
13	R7_Activ	BOOL	Reaktor 7 führt Prozess durch

Tabelle 5.4: Definierung von System-Status-Word

5.3.2 Interaktion zwischen zwei Reaktoren

Um die Arbeitsweise der Software zu verstehen, ist es wichtig den Kommunikationsablauf zu erklären. Hierbei geht es darum, wie die einzelnen Ebenen über die Control- und Status-Words miteinander interagieren. Dafür wird die Grundfunktion «Entleeren» und «Befüllen» angeschaut. Reaktor 1 entleert seinen Inhalt in Reaktor 2. Die Interaktion wird bis zur Aktivierung beider Reaktoren dargestellt.



Abbildung 5.4: Aktuelle Reaktorinteraktion

Bevor ein Prozess gestartet werden kann, müssen die Applikationen auf der Reaktoreinheitsebene gestartet werden. Wenn die Applikation läuft, wird dies über das Rezept-Status-Word an die Gesamtsystemebene mit der Position «SysReady» weitergegeben. Nun kann ein Rezeptschritt über das Rezept-Control-Word definiert werden. Die Position «PowerOn» und «Leitung» müssen nicht vom Benutzer für das Rezept definiert werden. Dies wird innerhalb des entsprechenden Ablaufs durch die Applikation gemacht. Wenn der Prozess gestartet wird, wird die Position «PowerOn» aktiviert und an den Primärreaktor gesendet.

Der Reaktor schaltet das System ein (Feldgeräte) und meldet dies über das Rezept-Status-Word zurück. Die Gesamtsystemebene kann nun den eigentlichen Prozess starten und beschreibt dafür die notwendigen Positionen im Rezept-Control-Word. Durch die Angabe des Sekundärreaktors im Control-Word wird auch das Sekundär-Control-Word beschrieben und an den Sekundärreaktor geschickt. Beide Reaktoren bereiten nun den Prozess vor (z.B. durch das Öffnen von Ventilen auf der Sekundärseite). Der Zustand der jeweiligen Reaktoren wird auf der Gesamtsystemebene in das Reaktorübergreifende System-Status-Word geschrieben. Dadurch weiss der Primärreaktor, wann der Sekundärreaktor bereit für die Übergabe ist. Der Prozess kann nun gestartet werden. Die Reaktoren geben nun die Position «ReactActiv» und die entsprechende Position für die Grundfunktion an.

5.3.3 Gesamtsystem-Applikation

Das folgende Schema zeigt die wichtigsten Elemente der Applikation und wie diese miteinander verbunden sind. Der Informationsfluss ist nicht im Schema integriert. Die Applikation besteht aus vier Segmenten: Funktionsabbildung, Anlagenabbildung, Kommunikationsabläufe und der Rezeptumsetzung

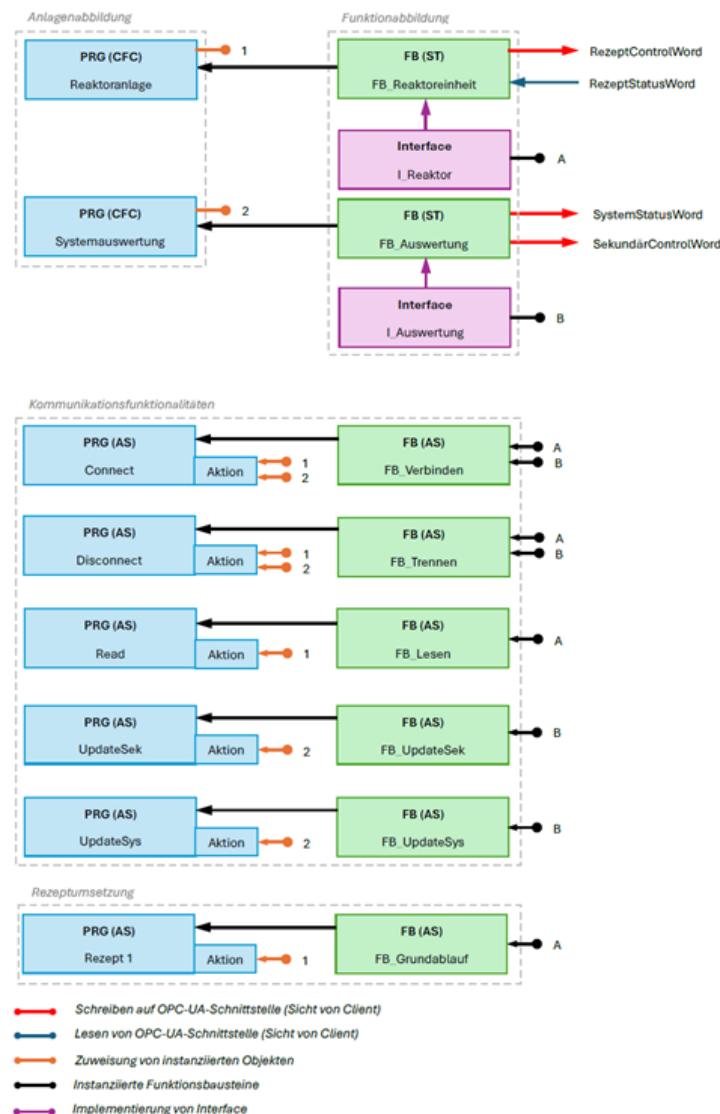


Abbildung 5.5: Gesamtsystem-Applikation

Funktionsabbildung:

Innerhalb des Segments «Funktionsabbildung» werden die Funktionen der Reaktoreinheit und der Auswertung definiert. Dabei geht es um die Kommunikation mit der OPC-UA-Schnittstelle. Die Reaktoreinheit ist für das Handling des Rezept-Contro/Status-Word zuständig, die Auswertung für das System-Status-Word und Sekundär-Control-Word. Für die Kommunikation stellt das TwinCat-Paket «TF6100» PLCopen genormte Funktionsbausteine zur Verfügung. Damit kann direkt aus dem Programm oder Funktionsblock heraus auf die OPC-UA-Schnittstelle zugegriffen werden.



Abbildung 5.6: PLCopen-Funktionsbaustein

Im Gegensatz zur OPC-UA-Schnittstelle via TwinCat-I/O-Gerät hat man mit den PLCopen-Funktionsbausteinen mehr Kontrolle über die Kommunikation. Die Applikation gibt vor, wann mit der OPC-UA-Schnittstelle interagiert werden soll. Damit eine Verbindung aufgebaut werden kann, werden Verbindungsparameter benötigt. Folgende Parameter müssen für eine Verbindung zwingend definiert werden.

ServerUrl:

Hierbei handelt es sich um die IP-Adresse des zu verbindenden Servers. Für die verschiedenen Reaktoren wurden folgende IP-Adressen definiert:

Reaktor 1:	'opc.tcp://169.254.240.1:4840'
Reaktor 2:	'opc.tcp://169.254.240.2:4840'
Reaktor 3:	'opc.tcp://169.254.240.3:4840'
Reaktor 4:	'opc.tcp://169.254.240.4:4840'
Reaktor 5:	'opc.tcp://169.254.240.5:4840'
Reaktor 6:	'opc.tcp://169.254.240.6:4840'
Reaktor 7:	'opc.tcp://169.254.240.7:4840'

SessionConnectInfo:

Es werden spezifische Verbindungsinformationen definiert. Dafür wird die TF6100-Struktur ST_UASessionConnectInfo verwendet. Alle Reaktoren benötigen dieselben Angaben:

eSecurityMode:	3 = Sign_Encrypted
eSecurityPolicyUri:	4 = Basic256
eTransportProfileUri:	1 = UATcp
stUserIdentTokenType:	1 = Username
stUserIdentTokenType.sTokenParam1:	'Administrator'
stUserIdentTokenType.sTokenParam2:	'1'

NamespacelURI:

Die Namespace-URI gibt an, in welchem Namespace die OPC-UA-Schnittstelle nach Variablen suchen soll. In der Regel ist dies die erste SPS-Applikation auf der Server-Seite. Alle Reaktoren haben die selbe Namespace-URI: 'urn:BeckhoffAutomation:ua:PLC1'

NodeID:

Definiert die eindeutige Adressierung der UA Node, welche der OPC-UA-Schnittstelle angibt, nach welchen Variablen gesucht werden soll. Dafür wird die TF6100-Struktur ST_UANodeID verwendet. Alle Reaktoren benötigen dieselben Angaben. Dabei wird je nach Kommunikationsvariable die entsprechende Bezeichnung angegeben.

eIdentifierType:

sIdentifier:

Rezept-Status-Word:

Rezept-Control-Word:

'eUAIdentifierType_String'

'GVL_OPc_UA.SW'

'GVL_OPc_UA.CW.PowerOn'

'GVL_OPc_UA.CW.SekReakt'

'GVL_OPc_UA.CW.Leitung'

'GVL_OPc_UA.CW.Temp'

'GVL_OPc_UA.CW.Enlteermenge'

'GVL_OPc_UA.CW.Pumpmenge'

'GVL_OPc_UA.CW.Haltedauer'

'GVL_OPc_UA.CW.Ruehrdauer'

'GVL_OPc_UA.CW.Wartedauer'

'GVL_OPc_UA.CW.Zirkulierdauer'

'GVL_OPc_UA.SysSW'

'GVL_OPc_UA.SekCW'

System-Status-Word:

Sekundär-Control-Word

Beim Rezept-Control-Word muss jeder Parameter der Struktur einzeln gesendet werden. Das Control-Word besteht aus verschiedenen Datentypen. Beim Senden der kompletten Struktur kommt es dadurch zu Problemen. Dies wurde bei Versuchen festgestellt. Die übertragenen Informationen konnten auf der Server-Seite nicht korrekt gelesen werden.

Die Aufsetzung und Funktionsweise der OPC-UA-Kommunikation wird in dieser Dokumentation nicht weiter behandelt. Weitere Informationen können aus den Dokumentationen von Beckhoff bezüglich TF6100 entnommen werden. Der Funktionsblock FB_Reaktoreinheit und FB_Auswertung sind objektorientiert aufgebaut. Beide besitzen Methoden und Eigenschaften. Innerhalb der Funktionsblöcke sind entsprechende TwinCat-Interfaces implementiert. Die Interfaces ermöglichen die anlagenunabhängige Verwendung der Funktionsblöcke innerhalb von Abläufen. Die Funktionen der Funktionsblöcke werden durch eine Case-Anweisung ausgelöst. Die Methode setzt die Case-Variable auf den jeweiligen Wert und startet somit die gewünschte Funktion. Folgende Methoden und Eigenschaften wurden definiert:

FB_Reaktoreinheit		
	Bezeichnung	Beschreibung
Methode:	M_LESEN	Das Rezept-Status-Word wird von der OPC-UA-Schnittstelle gelesen
	M_SCHREIBEN	Das Rezept-Control-Word wird auf die OPC-UA-Schnittstelle geschrieben
	M_TRENNEN	Die Reaktoreinheit trennt sich von der definierten OPC-UA-Schnittstelle
	M_VERBINDE	Die Reaktoreinheit verbindet sich mit der definierten OPC-UA-Schnittstelle
	M_ZUSTAND	Der aktuelle Zustand der Verbindung und des Servers wird abgefragt
Eigenschaft:	P_Status	Information über aktuellen Stand der Reaktoreinheit <i>Status = 0: System nicht verbunden</i> <i>Status = 1: System verbunden</i> <i>Status = 2: System bereit für Kommunikation</i> <i>Status = 3: System eingeschalten</i> <i>Status = 4: Reaktor ready</i> <i>Status = 5: Reaktor läuft</i> <i>Status = 6: Rezept-Status-Word wurde gelesen</i> <i>Status = 9: Error</i>

Tabelle 5.5: Definierung von Reaktoreinheit-Funktionsblock

FB_Auswertung		
	Bezeichnung	Beschreibung
Methode:	M_STATUS_SYSTEM	Das Rezept-Status-Word wird von der OPC-UA-Schnittstelle gelesen
	M_STATUS_SEK	Das Rezept-Control-Word wird auf die OPC-UA-Schnittstelle geschrieben
	M_TRENNEN	Die Reaktoreinheit trennt sich von der definierten OPC-UA-Schnittstelle
	M_VERBINDEN	Die Reaktoreinheit verbindet sich mit der definierten OPC-UA-Schnittstelle
Eigenschaft:	P_Status	Information über den aktuellen Stand der Reaktoreinheit Status = 0: System nicht verbunden Status = 1: System verbunden Status = 2: System wurde beschrieben Status = 9: System eingeschalten

Tabelle 5.6: Definierung von Auswertung-Funktionsblock

Die detaillierte Beschreibung der Funktionsblöcke würde den Rahmen der Dokumentation sprengen und wäre nicht zielführend. Somit wird auf Aspekte wie die Funktionsweise oder die definierten Variablen (Input, Output und Intern) nicht eingegangen. Bei Bedarf ist es am Effizientesten die Funktionsblöcke direkt in TwinCat anzuschauen, um die Funktionsweise zu verstehen.

Anlagenabbildung:

Die Anlagenabbildung stellt das Abbild der möglichst realen Anlage dar. Auf der Reaktoreinheitsebene kann dies sehr nachvollziehbar umgesetzt werden. Auf der Gesamtsystemebene kann dieses Prinzip nicht ganz realisiert werden. Die CFC-Programme bilden die Kommunikationsschnittstellen der einzelnen Reaktoren mit der OPC-UA-Schnittstelle ab. Für jeden Reaktor werden die entsprechenden Funktionsblöcke instanziert und mit In-/Outputs versehen.

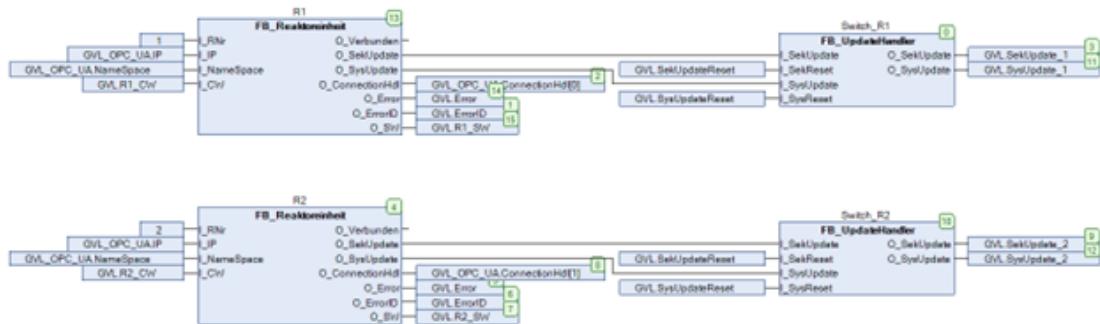


Abbildung 5.7: Abbildung der Anlage

Kommunikationsfunktionalitäten:

Die verschiedenen Kommunikationsfunktionen bestehe aus zwei Elementen. Das erste Element ist das Programm. In diesem Fall ist es das Programm Connect. Das Programm ist dafür da, den Verbindungsprozess für alle definierten Reaktoren durchzuführen. Innerhalb des Schrittes wird eine Aktion aufgerufen. Die Aktion verbindet den instanzierten Funktionsblock FB_Verbinden mit den entsprechenden instanzierten Objekten aus der Anlagenabbildung. Der Funktionsblock stellt das zweite relevante Element für die Kommunikationsfunktionen dar. Im Schritt wird ausserdem die Startmethode, für den im Funktionsblock definierten Ablauf, ausgeführt.

Der Aufbau dieser Struktur wurde im Kapitel 4.3.1 beschrieben. Alle Kommunikationsfunktionalitäten sind nach dieser Struktur definiert. Es gibt Unterschiede wie die Abläufe der Programme Connect, Disconnect und Read werden

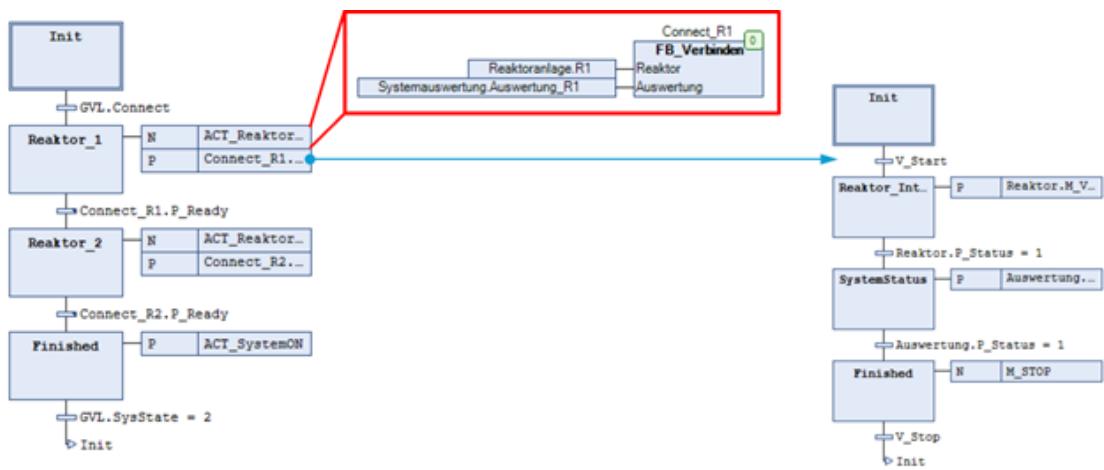


Abbildung 5.8: Programminteraktion für Kommunikation

über eine globale Variable gestartet. Die Programme UpdateSek und UpdateSys überprüfen, ob die instanzierte Reaktoreinheit (Aus der Anlagenabbildung) ein Update-Signal auf «True» setzt. Ist dies der Fall, so wird der Ablauf gestartet. Die Reaktoreinheit wertet somit während dem Betrieb das Rezept-Status/Control-Word aus und gibt vor, wann das System-Status-Word und Sekundär-Control-Word in die OPC-UA-Schnittstelle geschrieben werden soll.

Rezeptumsetzung:

Die Struktur der Rezeptumsetzung ist identisch zur Struktur der Kommunikationsfunktionalitäten. Das Rezept weist über eine Aktion dem instanziierten Funktionsblock FB_Grundablauf die instanzierte Reaktoreinheit und die Rezeptparameter zu. Der Rezeptschritt startet anschliessend über die entsprechende Methode den Ablauf.

Der Grundablauf hat verschiedene Aufgaben. In einem ersten Schritt sollen die Leitungen geprüft werden. Durch diese Prüfung wird definiert, welche Leitung für den Prozess eingesetzt werden kann. Dieses Feature ist im Moment noch nicht implementiert. Der zweite Schritt ist das Einschalten der Reaktoreinheit. Hierbei wird im Rezept-Control-Word das entsprechende Bit gesetzt und in die OPC-UA-Schnittstelle geschrieben. Dadurch werden auf der Reaktoreinheitsebene alle Feldgeräte eingeschalten. Die Rückmeldung über das Rezept-Status-Word wird im dritten Schritt ausgelesen. Nachdem das System eingeschaltet wurde, wird im vierten Schritt die eigentliche Grundfunktion ausgelöst. Auch hier wird das Rezept-Control-Word mit den definierten Rezeptparametern auf die OPC-UA-Schnittstelle geschrieben. Sobald dieser Prozess abgeschlossen ist, wird das Rezept-Control-Word zurückgesetzt und der Grundablauf wird beendet.

5.3.4 Reaktoreinheit-Applikation

Das folgende Schema zeigt die wichtigsten Elemente der Applikation und wie diese miteinander verbunden sind. Auch hier ist der Informationsfluss nicht im Schema integriert. Die Applikation besteht aus 3 Segmenten: Funktionsabbildung, Anlagenabbildung und System- und Kommunikationsauswertung.

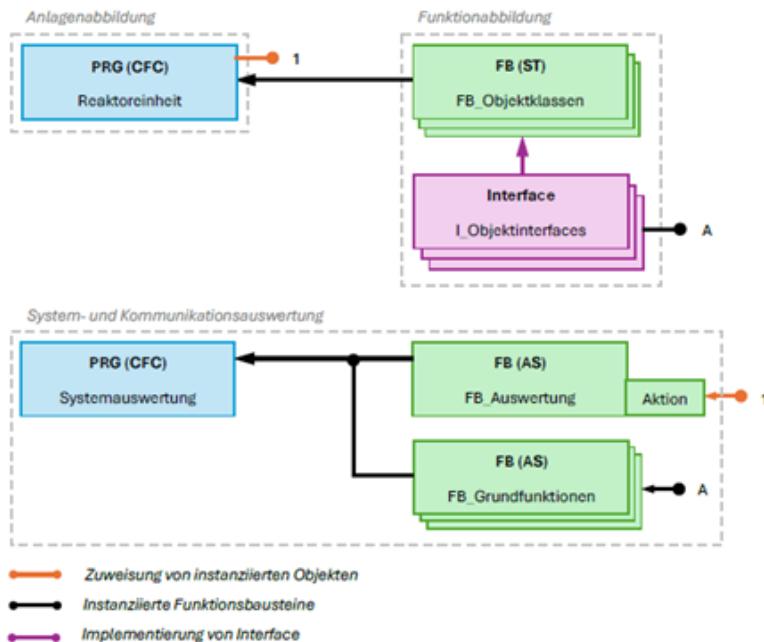


Abbildung 5.9: Reaktoreinheit-Applikation

Funktionsabbildung:

Innerhalb dieses Segments werden die verschiedenen Grundbausteine der Feldgeräte definiert. Die im Kapitel 4.1.1 beschriebenen Grundbausteine wurden als Funktionsblöcke umgesetzt. Während der Entwicklung kam es jedoch zu diversen Anpassungen der ursprünglichen Definierung (Bezogen auf Methoden, Eigenschaften und Variablen). Die ursprüngliche Definierung war jedoch eine gute Grundlage, um darauf aufzubauen und erleichterte den Entwicklungsprozess. Folgende Funktionsblöcke wurden erstellt:

Funktionsblock für Magnetventil:

Der Funktionsblock für das Magnetventil ist sehr einfach gehalten. Die Ventil besitzen leider keine Stellungsrückmeldung, dadurch kann sich das Ventil nicht selber kontrollieren.

FB_A_Magnetventil		
	Bezeichnung	Beschreibung
Methode:	M_A_OEFFNEN	Magnetventil wird im Automatik-Modus geöffnet
	M_A_SCHLIESSEN	Magnetventil wird im Automatik-Modus geschlossen
	M_M_OEFFNEN	Magnetventil wird im Manuell-Modus geöffnet
	M_M_SCHLIESSEN	Magnetventil wird im Manuell-Modus geschlossen
Eigenschaft:	P_Status	Information über den aktuellen Stand des Feldgerätes <i>Status = 0: Ventil geschlossen</i> <i>Status = 1: Ventil offen</i> <i>Status = 10: Ventil verriegelt</i>

Tabelle 5.7: Definierung von Magnetventil-Funktionsblock

Funktionsblock für Rührmotor:

Der Rührmotor muss über die Relais eingeschalten werden, erst nachdem diese reingeschalten wurde, kann gestartet werden. Für den Betrieb braucht der Funktionsblock eine Sollvorgabe für die Geschwindigkeit. Diese darf die definierte Maximalgeschwindigkeit (Eigenschaft) nicht überschreiten. Der Motor wird mit einem 0 – 10 V Signal betrieben, welches die Geschwindigkeit vorgibt. Die Faktorisierung der angegebenen Geschwindigkeit zum Ausgangssignal muss während den ersten Versuchen mit der Anlage definiert werden.

FB_A_Motor1R		
	Bezeichnung	Beschreibung
Methode:	M_A_AN	Motor wird im Automatik-Modus eingeschalten
	M_A_AUS	Motor wird im Automatik-Modus ausgeschalten
	M_A_START	Motor wird im Automatik-Modus gestartet
	M_A_STOP	Motor wird im Automatik-Modus gestoppt
	M_M_AN	Motor wird im Manuell-Modus eingeschalten
	M_M_AUS	Motor wird im Manuell-Modus ausgeschalten
	M_M_START	Motor wird im Manuell-Modus gestartet
	M_M_STOP	Motor wird im Manuell-Modus gestoppt
Eigenschaft:	P_Status	Information über den aktuellen Stand des Feldgerätes <i>Status = 0: Motor nicht eingeschalten</i> <i>Status = 1: Motor eingeschalten</i> <i>Status = 2: Motor läuft</i> <i>Status = 10: Motor verriegelt</i>
	P_Laufzeit	Information über die aktuelle Laufzeit des Motors
	P_MaxDrz	Für den Motor wird die maximale Drehzahl angegeben

Tabelle 5.8: Definierung von Rührer-Funktionsblock

Funktionsblock für Schlauchquetschpumpe:

Nicht jede Reaktoreinheit besitzt eine angeschlossene Schlauchquetschpumpe. Der Funktionsblock prüft über ein digitales Signal, ob eine Pumpe verbunden ist. Erst bei angeschlossener Pumpe kann der Funktionsblock die entsprechenden Outputs für den Betrieb setzen. Über ein 5 V Signal kann die Richtung definiert oder gestartet / gestoppt werden. Die 5 V Spannung kommt von einem Relais, welches über digitale Ausgänge angesteuert wird. Dies wird über die verschiedenen Methoden des Funktionsblocks gemacht.

FB_A_Motor2R		
	Bezeichnung	Beschreibung
Methode:	M_A_BWD	Motor wird im Automatik-Modus auf Rückwärtsbewegung eingestellt
	M_A_FWD	Motor wird im Automatik-Modus auf Vorwärtsbewegung eingestellt
	M_A_STAR	Motor wird im Automatik-Modus gestartet
	M_A_STOP	Motor wird im Automatik-Modus gestoppt
	M_M_BWD	Motor wird im Manuell-Modus auf Rückwärtsbewegung eingestellt
	M_M_FWD	Motor wird im Manuell-Modus auf Vorwärtsbewegung eingestellt
	M_M_STAR	Motor wird im Manuell-Modus gestartet
	M_M_STOP	Motor wird im Manuell-Modus gestoppt
Eigenschaft:	P_Status	Information über den aktuellen Stand des Feldgerätes <i>Status = 0: Motor nicht angeschlossen</i> <i>Status = 1: Motor angeschlossen</i> <i>Status = 2: Motor läuft</i> <i>Status = 10: Motor verriegelt</i>
	P_Laufzeit	Information über die aktuelle Laufzeit des Motors
	P_MaxDrz	Für den Motor wird die maximale Drehzahl angegeben

Tabelle 5.9: Definierung von Schlauchquetschpumpen-Funktionsblock

Funktionsblock für HK-System:

Während der Erarbeitung des Projektes konnte das HK-System nicht ausführlich getestet werden. Bei ersten Versuchen konnte sichergestellt werden, dass das System funktioniert. Die genaue Untersuchung des Signalflusses stellte sich jedoch als aufwändiger dar als gedacht. Das Ziel wäre, zu definieren, welche Klemmanschlüsse für welche System-Funktion verantwortlich sind und welche Signale diese benötigen. Die beigelegte Dokumentation gibt nur wage Informationen zu Schnittstellen an. Aus diesem Grund wurde die Analyse des HK-System aus diesem Projekt ausgeklammert. Es wurde ein provisorischer Funktionsblock für das HK-System definiert. Ohne genau zu wissen wie das System angesteuert werden kann, ist dieser im Moment nicht anwendbar. Dies hat auch einen Einfluss auf die Regelung. Diese kann nicht definiert werden, sobald das System nicht ausführlicher analysiert wurde. Der Funktionsblock für das HK-System wird somit in dieser Dokumentation nicht weiter vertieft.



Erhaltene Dokumentation von HK-System wird im Anhang beigelegt

Funktionsblock für Sensor:

Der Funktionsblock benötigt keine Methoden. Dieser misst das Signal kontinuierlich aus und gibt einen Temperatur- Füllstandwert aus. Die Skalierung ist dabei über eine lineare Funktion definiert. Der maximale Signalwert ist 27648. Das sind rund 85% des zu Verfügung stehenden 16 bit Signals. Dieser Wert wurde so aus der Literatur übernommen. Der Funktionsblock kann für den Temperatur- und Füllstandssensor eingesetzt werden.

FB_S_Sensor		
	Bezeichnung	Beschreibung
Eigenschaft:	P_Messwert	Der Signalwert wird in einen Temperaturwert umgerechnet
	P_GrenzeOben	Definiert eine obere Grenze, welche nicht überschritten werden darf
	P_GrenzeUnten	Definiert eine untere Grenze, welche nicht unterschritten werden darf

Tabelle 5.10: Definierung von Sensor-Funktionsblock

Die jeweiligen Funktionsblöcke konnten noch nicht an der realen Anlage getestet werden. Während den ersten Versuchen wird sich zeigen, welche Aspekte der Funktionsblöcke angepasst oder optimiert werden müssen. Im Moment stellt dies eine Basis dar, mit welcher gearbeitet und darauf aufgebaut werden kann.

Anlagenabbildung:

Alle real vorhandenen Feldgeräte werden im Programm Reaktoreinheit instanziert und abgebildet. Die entsprechenden In- und Outputs der instanziierten Funktionsblöcke werden mit den angeschlossene Klemmen verbunden und bilden die Schnittstellen zu den realen Geräten an der Anlage. Der Funktionsblock Mode ist ein Hilfsblock für das Handling des Komponenten-Modes.

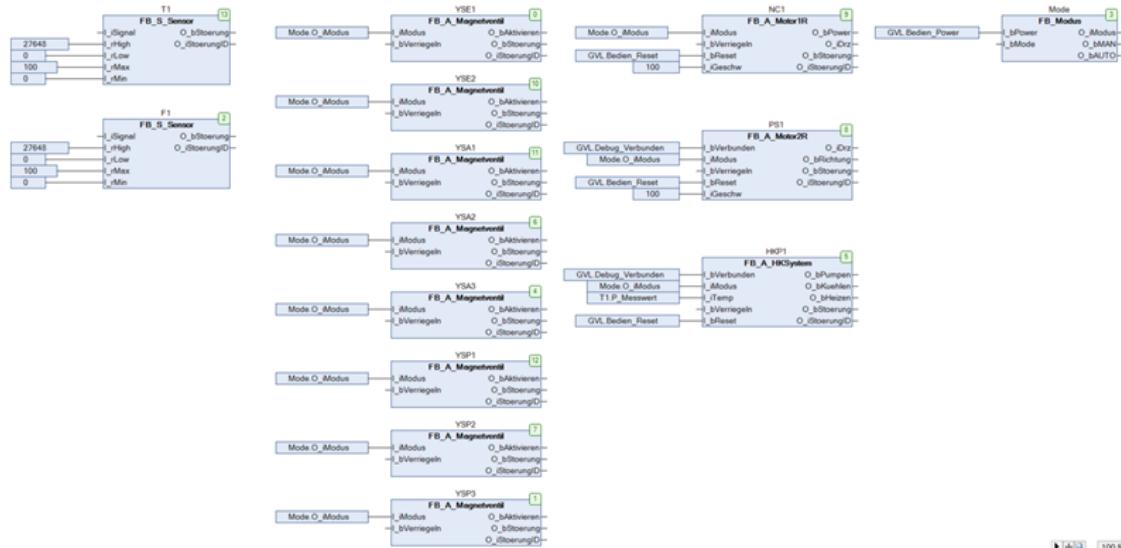


Abbildung 5.10: Instanziierung der Feldgeräte

System- und Kommunikationsauswertung:

Das CFC-Programm Systemauswertung instanziert den Funktionsblock für die Auswertung der Kommunikationsvariablen und die verschiedenen Grundfunktionen. Im Programm wird aber nur der Funktionsblock des Typs FB_Auswertung erstellt.



Abbildung 5.11: Instanziierung von Auswertung-Funktionsblock

Der Funktionsblock Auswertung überwacht alle Kommunikationsvariablen und hat folgende Aufgaben:

- ▶ Einschalten des Feldgeräte
- ▶ Starten der Grundfunktionen in der Rolle als Primärreaktor
- ▶ Starten der Grundfunktion "Befüllen" in der Rolle als Sekundärreaktor (Empfänger)
- ▶ Auswahl der entsprechenden Ventile, je nach Leitungsdefinition

Grundsätzlich hat der Funktionsblock auch einen Manuell-Modus. Hierbei würde die Applikation die Kommunikationsvariablen nicht auswerten und somit nicht auf diese reagieren. Dies wäre der Fall, wenn man die Reaktoreinheit manuell über eine HMI bedienen möchte. Der Funktionsblock handelt auch das Zurücksetzen des Systems auf den Startzustand.

Dies ist ein wichtiges Feature für Versuche. Um das System zurückzusetzen, muss beim Rezept-Control-Word eine Wartedauer von 99 Sekunden angegeben werden. Entsprechend kann das System auch von der Gesamtsystemebene zurückgesetzt werden.

Wird von der Auswertung erkannt, dass eine Grundfunktion gestartet werden soll, so wird eine definierte Aktion gestartet und die Startmethode wird aufgerufen. Die Struktur ist gleich aufgebaut wie bei den Kommunikationsfunktionalitäten in der Gesamtsystemebene. Die Grundfunktionen sind als Funktionsblöcke definiert. Die innerhalb des Prozesses verwendeten Feldgeräte werden als Inputvariablen instanziert. Dafür werden die jeweiligen Eigenschaften verwendet. Zur Veranschaulichung der Arbeitsweise eines Grundfunktion wird diese anhand der Grundfunktion «Befüllen» erklärt.

Der Ablauf wird über die Variable P_Status gestartet. Diese wird durch die Start-Methode gesetzt. Im Öffnen-Schritt wird das System aktiviert. Die globale Variable SysState wird auf 1 gesetzt. Nun weiss die Applikation, dass eine Grundfunktion durchgeführt wird. Nun wird das entsprechende Ventil für die Befüllung (Einlassventil des Reaktors) geöffnet. Schlussendlich meldet die Applikation der Gesamtsystemebene, dass der Reaktor ready für die Übergabe des Mediums ist. Der Ablauf wechselt nun in den Befüllen-Schritt. Der Reaktor wird nun auf «aktiv» gesetzt und überwacht die Füllstandmenge. Erreicht diese die vorgegebene Menge, so wird die Aktiv-Angabe zurückgesetzt und das Ventil wird geschlossen. Im letzten Schritt wird auch die Ready-Angabe des Reaktors zurückgesetzt und die globale Variable SysState wird auf 0 gesetzt. Die Grundfunktion wurde abgeschlossen und das System ist bereit für die Ausführung der nächsten Grundfunktion.

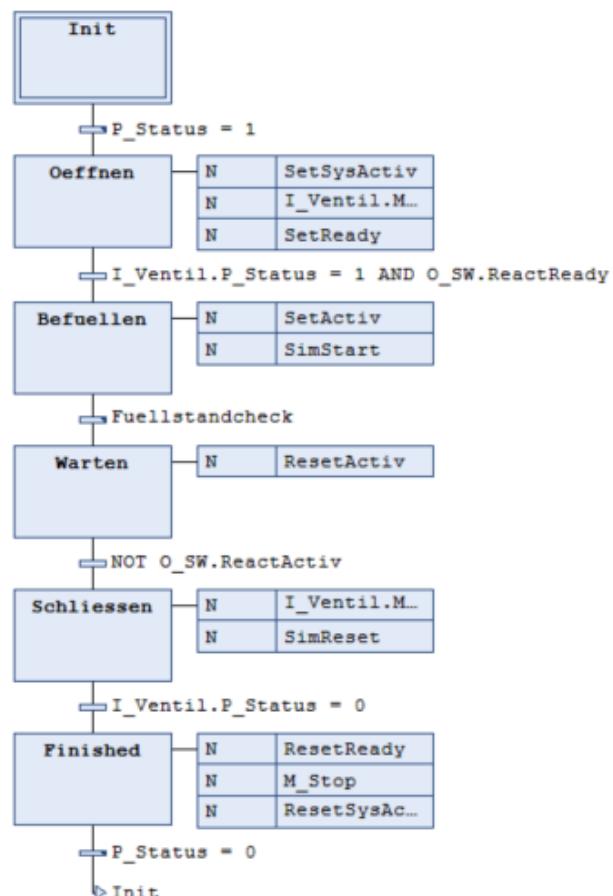


Abbildung 5.12: Beispiel einer definierten Grundfunktion



Aktueller Stand der Software wird in Anhang beigelegt

6 Auswertung

6.1 Aktueller Stand

6.1.1 Hardware

Im Moment wird die erste Reaktoreinheit montiert. Grundsätzlich fehlen nur noch die oberen und unteren Hauptleitungen. Alle Komponenten für die komplette Montage sind vorhanden. Bis jetzt gab es keine grösseren mechanischen Probleme bei der Montage. Kleine Optimierungsmöglichkeiten wurden bereits erkannt und werden für die nächsten Reaktoreinheiten umgesetzt. Sobald die erste Reaktoreinheit gebaut und getestet wurde, werden die 6 weiteren gebaut.

Die Montage hat sich deutlich weiter nach hinten verschoben als ursprünglich geplant. Verantwortlich dafür war die Auslastung der Werkstatt durch die Bachelorthesen während des Semesters. Diverse Fertigungsteile konnten jedoch selber umgesetzt werden, was den Prozess wieder verkürzt hat. Auch der Schaltschrank befindet sich in den letzten Zügen. Alle Schaltschränke werden auf einmal gebaut. Die Verdrahtung der Komponenten wird im Moment durch einen Sommerjobstudenten durchgeführt. Auch die eigentlichen Schaltschränke sind soweit vorbereitet. Da diese jedoch noch nicht ganz abgeschlossen sind, konnten auch noch keine Feldgeräte mit dem Schaltschrank verbunden werden.

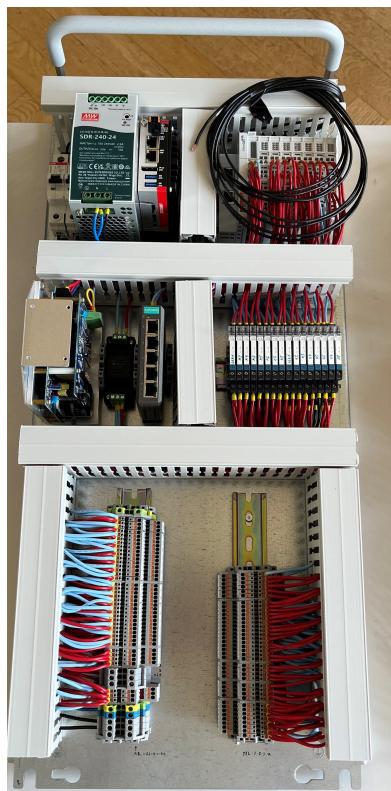


Abbildung 6.1: Verdrahtung für Schaltschrank

6.1.2 Software

Die Software ist soweit, dass die Gesamtsystemebene mit der Reaktoreinheitsebene kommunizieren kann. Ein Rezeptschritt kann gestartet und durchgeführt werden. Das Grundgerüst der Funktionalität des Systems ist somit vorhanden.

Die Software wurde direkt auf der SPS getestet. Dabei gab es zwei SPS, welche als Server konfiguriert waren. Ein Laptop diente als Client. Über ein einfaches HMI kann der Prozess auf der Gesamtsystemebene gestartet werden. Die Software definiert anschliessend anhand der angegebenen Prozessparameter das Rezept-Control-Word. Die Reaktoreinheit kann auf dieses reagieren und entsprechend Feedback geben. Im Moment wird aber noch kein Alarm- und Event-handling über die Kommunikationsschnittstelle gemacht. Auch die Leitungswahl auf der Gesamtsystemebene ist noch nicht implementiert. Die Priorität lag im ersten Schritt auf der allgemeinen Funktionalität. In einem zweiten Schritt können nun die entsprechenden Features hinzugefügt und optimiert werden.

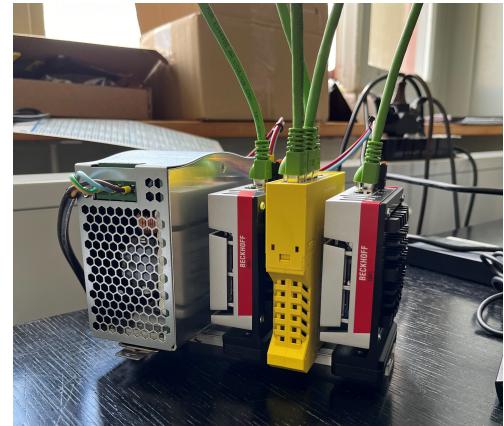


Abbildung 6.2: Testaufbau für Software

6.2 Vergleich mit Anforderungen

Alle Anforderungen werden mit der erarbeiteten Lösung verglichen. Die Anforderungen werden wie folgt bewertet:

	Nr	Beschreibung	F / M / W	Erfüllt
Mechanischer Aufbau	A1.1	Die chemische Reaktoranlage soll aus 7 Reaktoreinheiten bestehen	M	Ja
	A1.2	Reaktoren können aus dem Gesamtsystem entfernt oder hinzugefügt werden	W	Ja
	A1.3	Die chemische Reaktoranlage ist mobil	W	Ja
	A1.4	Die Position der einzelnen Reaktoren kann in der Höhe eingestellt werden	F	Ja
	A1.5	Die chemische Reaktoranlage wird in der Maschinenhalle der BFH betrieben	F	Ja
	A1.6	Jede Reaktoreinheit enthält einen eigenen Schaltschrank (inkl. SPS)	F	Ja
	A1.7	Schnittstelle zu externen Komponenten / Einheiten (HK / Pumpen etc.)	F	Ja
Software Struktur	A2.1	Die Software wird mit TwinCat umgesetzt und setzt auf Beckhoff-Komponenten	W	Ja
	A2.2	Die Software wird mit einer objektorientierten Struktur aufgebaut	F	Ja
	A2.3	Die Feldinstrumente (Aktoren / Sensoren) werden als Klassen abgebildet	F	Ja
	A2.4	Der Produktionsprozess wird über eine Rezeptursteuerung definiert	F	Nein
	A2.5	Die Rezeptursteuerung wird nach der Norm ISA-88/95 definiert	F	(ja)
	A2.6	Die chemische Reaktoranlage kann im Automatik- oder Manuell-Modus betrieben werden	F	(ja)
Kommunikation	A3.1	Die Kommunikation wird über OPC UA realisiert	F	Ja
	A3.2	Die OPC UA Kommunikation verwendet ein Server-Client-Model	F	Ja
	A3.3	Die OPC UA Kommunikation soll mit der objektorientierten Struktur der Softwareumsetzung arbeiten	F	Ja
	A3.4	Eine Reaktoreinheit kann durch entfernen / verbinden einer Schnittstelle, aus der Topologie entfernt oder integriert werden	W	Ja
	A3.5	Aufbau einer Netzwerktopologie	F	Ja
	A3.6	Von jeder Reaktoreinheit kann man auf die Gesamtanlage zugreifen (wenn integriert)	W	Nein
HMI	A4.1	Die HMI wird mit objektorientierter Software realisiert	W	Nein
	A4.2	Es gibt ein HMI auf der Ebene der Gesamtleitebene	F	(ja)
	A4.3	Jede Reaktoreinheit besitzt eine eigene Bedienebene	F	Nein
	A4.4	Eine Rezeptur kann über die HMI der Gesamtleitebene ausgewählt und gestartet werden	F	(ja)
	A4.5	Die HMI der Reaktoreinheit ermöglicht die Direktbedienung des Reaktors	F	Nein
	A4.6	Berücksichtigung von Alarmierungen und Trenddarstellung	F	Nein
	A4.7	Die HMI der Reaktoreinheit kann webbasiert erfolgen	W	Nein
	A4.8	Die chemische Reaktoranlage kann im Automatik- oder Manuell-Modus betrieben werden	F	Nein
Allgemein	A5.1	Ein Elektroschema soll erstellt werden	F	Ja
	A5.2	Der Zustand der Anlage wird über eine Signallampe dargestellt	W	Nein
	A5.3	Durchführen eines FAT (Factory Acceptance Test)	W	Nein
	A5.4	Erstellen einer Inbetriebnahmeanleitung	W	(ja)
	A5.5	Projektende ist der 02.08.2024	F	Ja

Tabelle 6.1: Auswertung der Anforderungen

Viele der Anforderungen wurden mit der erarbeiteten Lösung erfüllt. Dennoch gibt es einige Anforderungen, die nicht oder nur teilweise erfüllt wurden:

- A2.4: In einem ersten Schritt wurde eine Rezeptursteuerung umgesetzt. Sobald alle Features der Anlage umgesetzt wurden und diese läuft, kann eine Rezeptursteuerung erarbeitet werden.
- A2.5: Es wurde keine Rezeptursteuerung umgesetzt. Jedoch konnte diese Struktur auch für die Rezeptursteuerung eingesetzt werden.
- A2.6: Die Software ist grundsätzlich darauf ausgelegt und wurde auch auf diese Funktion getestet. Das provisorische Test-HMI bietet aber im Moment keine Möglichkeit dies umzuschalten.

- A3.6: Im Moment ist es nicht möglich, von der Reaktoreinheitsebene auf die Gesamt-systemebene aktiv zuzugreifen. Zwischen den Ebenen findet die Interaktion nur zwischen den definierten Status- und Control-Word statt.
- A4.1: Das HMI ist noch nicht soweit, dass dieses objektorientierte Elemente beinhalten könnte.
- A4.2: Es gibt ein HMI, jedoch ist dieses nur für Test- und Erarbeitungszwecke.
- A4.3: Die Reaktoreinheit hat zum jetzigen Zeitpunkt kein HMI.
- A4.4: Ein Rezept kann gestartet werden, die Parameter sind aber fix hinterlegt. Es kann noch nicht zwischen mehreren Rezepten ausgewählt werden.
- A4.5: Die Reaktoreinheit hat zum jetzigen Zeitpunkt kein HMI.
- A4.6: Wurde zum jetzigen Zeitpunkt noch nicht implementiert.
- A4.7: Wurde zum jetzigen Zeitpunkt noch nicht implementiert.
- A4.8: Die Software ist grundsätzlich darauf ausgelegt und wurde auch auf diese Funktion getestet. Das provisorische Test-HMI bietet aber im Moment keine Möglichkeit dies umzuschalten.
- A5.2: Im Moment wurde noch keine Signallampe verbaut. Die Anschlüsse lassen aber das Ansteuern einer Signallampe zu.
- A5.3: Die Anlage ist noch nicht soweit, dass ein FAT durchgeführt werden könnte.
- A5.4: Die Anlage ist noch nicht soweit, dass eine Inbetriebnahmeanleitung geschrieben werden müsste. Es wurde jedoch gut beschrieben, wie die Software aufgebaut und strukturiert ist.

6.3 Weiterführende Arbeiten

6.3.1 Hardware

- ▶ Fertigmontage der ersten Reaktoreinheit (rein mechanische Komponenten)
- ▶ Fertigstellen aller Verdrahtungen
- ▶ Einbauen des kompletten Schaltschrankes in die erste Reaktoreinheit
- ▶ Fertigen / Bestellen von Komponenten für weitere Reaktoreinheiten
- ▶ Optimieren von diversen mechanischen Punkten, welche bei der Montage aufgefallen sind
- ▶ Vorsehen von Signallampe
- ▶ Analyse von Anschlussmöglichkeiten für HK-System

6.3.2 Software

- ▶ Optimieren von Stabilität des Grundablaufes
- ▶ Optimierung der Funktionsblöcke der Feldgeräte
- ▶ Ausarbeiten von Funktionsblock für HK-System
- ▶ Einbindung von Signallampe
- ▶ Implementierung der Leitungswahl
- ▶ Implementierung von Alarm- und Eventhandling
- ▶ Implementierung der Rezeptwahl

- ▶ Erstellen eines HMI für Reaktoreinheitsebene
- ▶ Erstellen eines HMI für Gesamtsystemebene (nicht nur für Testzwecke)
- ▶ Testen der Software an realem System

6.3.3 Allgemein

- ▶ Konfigurieren von Reaktor-SPS 3 bis 7
- ▶ Durchführen eines FAZ (Factory-Acceptance-Test)
- ▶ Erstellen einer Anleitung für die Inbetriebnahme
- ▶ Informieren über TE2000 als Möglichkeit für HMI
- ▶ Informieren über MTP als mögliche Softwaregrundlage für Anlage

7 Schlussfolgerung / Fazit

Das Projekt hatte sehr viele interessante, aber auch sehr frustrierende Aspekte. Auf einer technischen Ebene konnte ich extrem viel dazulernen. Vor allem im Bereich der Software konnte ich viel vom Projekt profitieren. Die verschiedenen Aspekte wie OOP, die Kommunikation und eine Rezeptsteuerung zu kombinieren war sehr spannend. Da dies eine Anwendung war, welche man nicht einfach in der Literatur oder in Beispielen findet, musste ich viel selber erarbeiten. Dadurch musste ich mich intensiv mit den verschiedenen Aspekten auseinandersetzen. Für die objektorientierte Struktur habe ich mich intensiv mit der entsprechenden Literatur von Matthias Seitz auseinander gesetzt. Somit konnte ich sehr tief in das Thema von objektorientierter Programmierung eintauchen. Auch bezüglich der Kommunikation via OPC-UA musste ich mich sehr in die Thematik hineinarbeiten. Hierbei ging es aber mehr um die technische Umsetzung und Anwendung. Es war sehr befriedigend, sich intensiv mit den Dokumentationen von Beckhoff auseinanderzusetzen, um schlussendlich eine robuste und gute Kommunikation aufzubauen zu können.

Zusätzlich sind all diese Aspekte sehr industrienahe und aktuell. Mich damit zu beschäftigen, ist auch ein Mehrwert für die Arbeit nach dem Studium.

Vom Rahmen war das Projekt deutlich grösser als zu Beginn erwartete. Der mechanische Aufbau benötigte sehr viel Zeit für die Planung wie auch für die Fertigung und Montage. Auch die Software war sehr zeitintensiv. Vor allem die Umsetzung der OPC-UA-Kommunikation via PLCopen-Funktionsbausteine benötigte viel Zeit und Arbeit.

Für ein nächstes Mal würde ich mehr darauf pushen, den mechanischen Aufbau in einem ersten Schritt simpel zu halten. Meiner Meinung nach hätte die Anlage zu erst im kleinen Massstab aufgebaut werden sollen (im gleichen Rahmen wie das frühere «SITAS Tanksystems» der BFH). Die Elektronik und die Software wäre im Endeffekt gleich geblieben. Jedoch hätte man schneller einen Prototypen aufbauen und Erfahrungen sammeln können. Fertigungs- und Bestellteile wären bei einem kleineren System einfacher und schneller bereit gewesen. Mit den gewonnenen Erfahrungen hätte man dann die «grosse», industrielle Anlage planen können. Mit dem nun vorhandenen System kommt man sicher auch ans Ziel. Anpassungen und Optimierungen sind einfach mit mehr Aufwand und Kosten verbunden.

Als finales Fazit kann ich sagen, dass ich wirklich viel gelernt habe und mich intensiv mit relevanten Themen der Industrie beschäftigen konnte. Wie bei vielen Projekten war das Einschätzen des Projektrahmen sehr schwierig und es hat sich gezeigt, dass deutlich mehr Zeit benötigt worden wäre, um die Anlage zum gewünschten Stand zu bringen. Ich bin aber überzeugt, dass dieser Stand in naher Zukunft erreicht und den Studenten damit eine gute Lernplattform geboten werden kann. Ich bin sehr gespannt darauf, die Anlage das erste Mal in Betrieb zu sehen.

8 Anhang

1. Auftrag
2. Brainstorming
- 3: Zeitplan
- 4: Datenblätter
- 5: Fertigungsdaten
- 6: Bestellliste
- 7: CAD
- 8: Komponentenlisten
- 9: Elektroschema
- 10: Handbücher
- 11: Software
- 12: Normen
- 13: Fotos

Eigenständigkeitserklärung

Hiermit erkläre ich, dass ich die hier vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Sämtliche Ausführungen, die anderen Schriften wörtlich oder sinngemäß entnommen wurden, habe ich als solche kenntlich gemacht.

Hiermit stimme ich zu, dass die vorliegende Arbeit in elektronischer Form mit entsprechender Software überprüft wird.

2. August 2024



Y. Spatz

Literaturverzeichnis

- [1] Kevin Cope. What is the automation pyramid.
- [2] Technisches Komitee TK65 <Industrielle Prozessleit und Automatisierungstechnik>. *Chargenorientierte Fahrweise (Teil 1/3/4)*. Electrosuisse, Luppenstarsse 1, CH-8320 Fehraltorf, 1995 / 2008 / 2010.
- [3] Beckhoff Information System. Hotconnect.
- [4] COPADATA. Zenon software platform.
- [5] Beckhoff Information System. Te2000 | twincat 3 hmi engineering.
- [6] Matthias Seitz. *Speicherprogrammierbare Steuerungen in der Industrie 4.0 - Kapitel 7*. Hanser Fachbuch, 2021.
- [7] Stefan Luber / Nico Litzel. Was ist opc ua.
- [8] OPC-Router. Was ist opc ua? die wichtigsten begriffe im Überblick.
- [9] Beckhoff Information System. Handbuch - tf6100 - twincat 3 | opc ua server, 2024.
- [10] Beckhoff Information System. Handbuch - tf6100 - twincat 3 | opc ua client, 2024.
- [11] Beckhoff Information System. Handbuch - tf6100 - twincat 3 | opc ua gateway, 2024.
- [12] Beckhoff Information System. Handbuch - tf6100 - twincat 3 | opc ua configurator, 2024.

Abbildungsverzeichnis

1.1	Grobprojektplanung	1
1.2	Grober Zeitplan	2
2.1	Startsystemübersicht	3
2.2	Übersicht über Schnittstellen	4
2.3	Basis der Reaktoreinheit	5
2.4	Reaktordeckel der Basis	6
2.5	FU-Kontrollboard	7
2.6	Standard External Controll Schnittstelle	7
2.7	Anschluss des Antriebsmotors	8
2.8	Anschlüsse des HK-Systems	8
2.9	Automatisierungspyramide	9
2.10	Ebenen der Anlage	10
2.11	Struktur einer Rezeptursteuerung	10
2.12	Objektorientierte Struktur	12
2.13	Beispiel einer Motorklasse	13
2.14	Reaktorlayout aufgeteilt	15
2.15	System- und Kontextgrenzen	16
3.1	Mechanisches Schema des Aufbaus	19
3.2	Umsetzung des Layouts	20
3.3	Überblick über Gesamtsystem	20
3.4	Überblick über Reaktoraufbau	21
3.5	Schnittstellen für IPC und E/A-Koppler	22
3.6	Schnittstellen Ventilanschluss	22
3.7	Schnittstellen Füllstandauswertung	23
3.8	Schnittstellen Schlauchquetschpumpe	23
3.9	Schnittstellen Rührantrieb	24
3.10	Schnittstellen Temperatursensor	24
3.11	Schnittstellen Prüfsignalauswertung	24
4.1	Struktur eines Rezeptprozesses	30
4.2	Grobschema der Reaktorinteraktion	31
4.3	Arbeitsweise einer Rezeptursteuerung	32
4.4	Erster Entwurf einer möglichen Reaktorkommunikation	33
4.5	Erster Entwurf des Control-Words	34
4.6	Erster Entwurf von Interaktion zwischen zwei Reaktoren	35
4.7	Nächste Iteration von Control/Status-Word	36
4.8	Ablauf einer Leitungsprüfung	36
4.9	Entwurf des Leitungs-Control-Words	37
4.10	Beispielsituation für Leitungsmanagement	37
4.11	Leitungsmanagement auf Rezeptstufe	37
4.12	Schematischer Ausführungsprozess des Rezeptes	38
4.13	Schematischer Prüfprozess	38

4.14 OPC-UA-Topologie	40
4.15 Programmstruktur aus Literatur	41
4.16 Beispiel einer Programmstruktur	42
4.17 Angepasste Grundstruktur	43
4.18 Definierung der Feldgerätbezeichnung	45
5.1 Interaktionsschnittstellen der Software	48
5.2 Arbeitsweise der Gesamtsystem-Applikation	49
5.3 Arbeitsweise der Reaktoreinheit-Applikation	50
5.4 Aktuelle Reaktorinteraktion	52
5.5 Gesamtsystem-Applikation	53
5.6 PLCopen-Funktionsbaustein	54
5.7 Abbildung der Anlage	56
5.8 Programminteraktion für Kommunikation	57
5.9 Reaktoreinheit-Applikation	58
5.10 Instanziierung der Feldgeräte	61
5.11 Instanziierung von Auswertung-Funktionsblock	61
5.12 Beispiel einer definierten Grundfunktion	62
6.1 Verdrahtung für Schaltschrank	63
6.2 Testaufbau für Software	64

Tabellenverzeichnis

1.1 Auftragsinterpretation	2
2.1 Gelieferte Grundkomponenten	5
2.2 Definierte Steuerungskomponenten	6
2.3 Reaktorkonfiguration	15
2.4 Projektanforderungen	17
4.1 Planung von Rührer-Grundbaustein	27
4.2 Planung von Schlauchquetschpumpe-Grundbaustein	28
4.3 Planung von HK-System-Grundbaustein	28
4.4 Planung von Magnetventil-Grundbaustein	29
4.5 Planung von Füllstandssensor-Grundbaustein	29
4.6 Planung von Temperatursensor-Grundbaustein	29
4.7 Definierte Grundfunktionen	30
4.8 Grobschema der Grundfunktionen	31
4.9 Definierte Störungen für Funktionsbausteine	44
4.10 Definierte Modi für Funktionsbausteine	44
5.1 Definierung von Rezept-Control-Word	51
5.2 Definierung von Rezept-Status-Word	51
5.3 Definierung von Sekundär-Control-Word	51
5.4 Definierung von System-Status-Word	52
5.5 Definierung von Reaktoreinheit-Funktionsblock	55
5.6 Definierung von Auswertung-Funktionsblock	56
5.7 Definierung von Magnetventil-Funktionsblock	58
5.8 Definierung von Rührer-Funktionsblock	59
5.9 Definierung von Schlauchquetschpumpen-Funktionsblock	59
5.10 Definierung von Sensor-Funktionsblock	60
6.1 Auswertung der Anforderungen	65

Glossar

HK-System Heiz- und Kühlssystem (HK-System) Damit wird ein Gerät beschrieben, welches eine Flüssigkeit präzise auf einer definierten Temperatur halten kann

IPC Industrie-PC (IPC) Ein alternativer Begriff für eine SPS

SPS Steicherprogrammierbare Steuerung (SPS) Gerät zur Steuerung und Regelung von Maschinen und Anlagen