



Fakultät Informatik

Praktikumsbericht

im Studiengang Informatik

vorgelegt von

Jan-Eric Gedicke

Matrikelnummer 3653446

Erstgutachter: Prof. Dr. Trommler

Betreuer: Conrad, Maximilian

© 2025

Inhaltsverzeichnis

1	Einleitung	1
2	Vorstellung der Siemens AG	2
3	Vorstellung meiner Fachabteilung	3
4	Vorstellung der SmartFactory@EH-Anlage	4
4.1	Station 1: Abfüllung (AS)	6
4.2	Station 2: Quality Gate (QG)	7
4.3	Station 3: Kommissionierbahnen (KB)	8
4.4	Station 4: Eckstation (ES)	8
4.5	Station 5: Recycling (Recy)	9
4.6	Station 6: Modulares Produktionssystem (MPS)	11
4.7	Netzwerkverbund	11
4.8	Begriffsklärungen	12
5	Projekt SmartFactory@EH-Anlage	14
5.1	Projekt Smart Factory-Anlage: Aufgabenstellung	14
5.2	Projekt Smart Factory-Anlage: Zielsetzung	14
5.3	Projekt SmartFactory-Anlage: Projektorganisation	15
5.4	Projekt SmartFactory-Anlage: Projektverlauf	17
5.4.1	Auftaktwoche:	17
5.4.2	Hands-On SmartFactory-Anlage:	18
5.4.3	Bibliotheksverantwortlicher (erweiternder Aufgabenbereich):	18
5.4.4	Flaschenmanagement in der Unit:	19
5.4.5	Kommunikation:	20
5.4.6	Drehteller:	21
5.4.7	Kugelabfüllung:	23
5.4.8	Kugelrückführung:	23
5.5	Projekt SmartFactory-Anlage: Ergebnis	24

6	Persönliches Fazit zum Praktikum	26
----------	---	-----------

1 Einleitung

Dieser Praktikumsbericht dokumentiert mein Praxissemester bei der Siemens AG in der Fachabteilung DI CS SD EH. Als dualer Student mit vertiefter Praxis bei Siemens bin ich sowohl in meiner Fachabteilung (DI CS SD EH FA 2), als auch der SPE (Siemens Professional Education), also einer Ausbildungsabteilung, tätig. Mein Praxissemester umfasst daher zwei praktische Projekte, eines in meiner Abteilung und eines der Ausbildungsabteilung, welche ich begleiten durfte. In meiner Fachabteilung "Expert House" (DI CS SD EH), war ich Teil der Weiterentwicklung der Smart Factory-Anlage, einer für Studierende aber auch Mitarbeitern konzipierten Trainings- und Automatisierungsanlage zur Simulation einer Flaschenabfüllungsproduktion. Während meiner Zeit in der Ausbildungsabteilung habe ich dem Projekt "BersiBot", die Programmierung eines Roboters mit KI-Bilderkennung und Webseitensteuerung, mitgewirkt. Die spezifischen Aufgabenstellungen, Ziele, das Projektmanagement, der Projektverlauf und die erreichten Ergebnisse des Projekts „SmartFactory- Anlage“ sind Hauptgegenstand dieses Berichts. Auf das Projekt "BersiBot" werde ich in diesem Bericht nicht gesondert eingehen, da es gerade im Hinblick auf das Projektmanagement und die allgemeine Vorgehensweise sehr starke Überschneidungen beider Projekte gibt und die SmartFactory-Anlage das deutlich umfangreichere Projekt ist.

2 Vorstellung der Siemens AG

Die Siemens AG wurde 1847 von Werner Siemens und Georg Halske als „Telegraphen Bau-Anstalt von Siemens und Halske“ in Berlin gegründet.¹ Anfangs konzentrierte sich das Unternehmen auf die Entwicklung und den Vertrieb von Telegrafenapparaten, wodurch Siemens schnell zu einem bedeutenden Akteur in der Elektroindustrie wurde. In den darauffolgenden Jahren expandierte das Unternehmen nach Russland und England² und erschloss neue Geschäftsfelder wie die Elektrifizierung der Eisenbahn.³

In den 1930er Jahren erweiterte Siemens seine Produktpalette um elektrische Haushaltsgeräte, Wärme- und Heizungskonzepte sowie Medizintechnik.⁴ Heute ist Siemens in vielen Branchen vertreten, darunter Industrie, Gebäudetechnik, Energietechnik, Mobilität, Gesundheitstechnik, Business Administration und Human Resource Management. Die Schwerpunkte des Unternehmens liegen auf der Digitalisierung, insbesondere der digitalen Transformation von Fertigungsprozessen, Transportnetzwerken, Gebäuden und Energiesystemen.

Die Siemens AG beschäftigt weltweit rund 320.000 Mitarbeiter und erzielte im Geschäftsjahr 2022/23 einen Umsatz von 74,8 Milliarden Euro sowie einen Nettogewinn von 8,5 Milliarden Euro.⁵ Ein Jahr später, im Jahr 2024, lag der Umsatz bei 75,9 Milliarden Euro, was die anhaltende Stärke des Unternehmens unterstreicht.

¹Vgl. Ernst Klett Verlag: Infoblatt Siemens AG, 2004, <https://www.klett.de/alias/1036900>. Zugriff: 5. Januar 2025

²Vgl. Merkur: Siemens – Geschichte, Aktie und Tätigkeitsfelder, 2022, <https://www.merkur.de/wirtschaft/siemens-geschichte-aktie-und-taetigkeitsfelder-91268844.html>. Zugriff: 19. Januar 2024

³Vgl. Ernst Klett Verlag, 2004

⁴Vgl. Merkur, 2022

⁵Vgl. Siemens, 2023 zitiert nach de.statista.com, <https://de-statista-com.thn.idm.oclc.org/statistik/daten/studie/73827/umfrage/umsatz-von-siemens-seit-2005/>. Zugriff: 5. Januar 2025

3 Vorstellung meiner Fachabteilung

Ich bin in der Abteilung **Digital Industries Customer Service Service Delivery Expert House (DI CS SD EH)** tätig, die Teil des Kundenservices der Sparte Digital Industries der Siemens AG ist. Das Expert House bietet Siemens-Kunden Zugang zu fachlichen Spezialisten, die bei der Umsetzung spezifischer Projekte unterstützen. Innerhalb des Expert House existieren verschiedene Fachbereiche, darunter die Bereiche **Process Automation** und **Factory Automation**, letzterem bin ich zugeordnet.

Die **Factory Automation** umfasst ein breites Spektrum an Tätigkeiten wie die Programmierung und Inbetriebnahme von Automatisierungsanlagen für Kunden sowie die Entwicklung maßgeschneiderter Softwarelösungen. Aufgrund des hohen fachlichen Spezialisierungsgrads ist der Bereich in mehrere nummerierte Gruppen unterteilt. Meine Gruppe, **Factory Automation 2 (FA2)**, fokussiert sich auf Themen wie **Safety**, **Reliability** und **Technology Services**.

Wir stellen sicher, dass die Anlagen nicht nur den höchsten Sicherheitsstandards entsprechen, sondern auch zuverlässig und technologisch auf dem neuesten Stand sind. Dazu gehört die Überprüfung und Optimierung sicherheitsrelevanter Prozesse, die Analyse der Anlagenzuverlässigkeit sowie die Bereitstellung technologischer Beratungsdienste. Diese Expertise ermöglicht es uns, Kunden bestmöglich zu unterstützen oder bei Bedarf individuelle Lösungen mit ihnen zu entwickeln, die höchsten Anforderungen an Sicherheit und Effizienz gerecht werden.

4 Vorstellung der SmartFactory@EH-Anlage

Die **SmartFactory-Anlage** ist eine speziell für Schulungszwecke konzipierte Automatisierungsanlage, die von der Abteilung **Expert House** der Siemens AG entwickelt wurde. Sie dient als Trainingsplattform für Auszubildende, Studierende und Mitarbeitern, um praxisnahe Erfahrungen in der Automatisierungstechnik zu sammeln und mit einer Vielzahl von Produkten aus dem Siemens-Katalog vertraut zu werden. Die Anlage simuliert eine Flaschenabfüllung und wird zur Ausbildung sowie zum Testen von neuen Technologien und Funktionen genutzt.

Die Anlage kann über drei Modulbetriebsarten gesteuert werden: **Einrichtungsbetrieb**, **Handbetrieb** und **Automatikbetrieb**, sowie zwei Anlagenbetriebsarten: **Modularbetrieb** und **Inselbetrieb**. Im **Einrichtungsbetrieb** kann der Bediener alle Anlagenteile uneingeschränkt bewegen, auch wenn dies zu Beschädigungen führen könnte. Der **Handbetrieb** ermöglicht es, einzelne Funktionen auszuführen, wie das Befüllen einer Flasche oder das Entleeren eines Bandes. Im **Automatikbetrieb** läuft die Anlage vollständig automatisiert und erfordert nur im Fehlerfall, zum Anlegen eines Auftrags oder bei einer Bedieneranforderung, einen Bediener. Eine Bedieneranforderung wäre hier zum Beispiel das manuelle Entleeren der Kommissionierbahn oder das Entleeren eines Deckelausschussbands. Während es im **Modularbetrieb** darum geht, dass die Anlage als Ganzes betrachtet wird, also eine Kommunikation mit Nachbarstationen aufgenommen werden kann, wird im **Inselbetrieb** jedes Modul einzeln betrachtet und getestet.

Der modulare Aufbau der Anlage umfasst sieben verschiedene Teilmodule: die **Abfüllstation**, die **Qualitätskontrolle**, die **Kommissionierbahn**, die **Eckstation**, die **Recyclingstation**, den **Delta-Picker** und das **MPS-Modul**. Diese Stationen sind durch ein Kreislaufsystem miteinander verbunden, wobei die Flaschen von der Abfüllstation über die verschiedenen Stationen transportiert werden. Im Kreislaufsystem werden die Flaschen am Ende des Prozesses, nach dem Recycling, wieder zur erneuten Verarbeitung bereitgestellt.

Jedes Modul der Anlage hat einen spezifischen Schwerpunkt und ist mit ähnlicher Hardware ausgestattet, einschließlich einer **Siemens-Simatic S7-1500 Speicherprogrammierbaren Steuerung** (SPS), außer dem Quality-Gate, bei dem ältere Steuerungstechnik mit einer ET200Sp verbaut ist, einem **Siemens HMI** (Human-Machine Interface) für Statusmeldungen und Benutzereingaben sowie Transportbändern. Zudem sind an einigen Stationen pneumatisch betriebene Greifer und RFID-Schreib-Lesegeräte verbaut, die eine präzise Steuerung und Überwachung der Anlage ermöglichen. Die Anlage läuft iterativ, was bedeutet, dass nach Abschluss des Kreislaufs die Flaschen, Deckel und Kugeln für die erneute Abfüllung bereitgestellt werden.

Die Funktionsweise der einzelnen Stationen im **Automatikbetrieb** ist wie folgt: In der **Abfüllstation** werden die Flaschen mit Kugeln befüllt, bevor sie in die nächsten Stationen weitergeleitet werden. In der **Qualitätskontrolle** wird die Qualität der befüllten Flaschen geprüft. Danach erfolgt in der **Kommissionierbahn** und der **Eckstation** die Sortierung der Flaschen sowie eine weitere Qualitätsprüfung. Der Kreislauf endet in der **Recyclingstation**, in der die Flaschen entleert und die Kugeln in die richtigen Behälter sortiert zurückgeführt werden.

Durch diese modulare und vielseitige Struktur bietet die **SmartFactory-Anlage** den Auszubildenden eine umfassende Ausbildung und ein realistisches Trainingsumfeld für die Arbeit mit modernen Automatisierungssystemen.

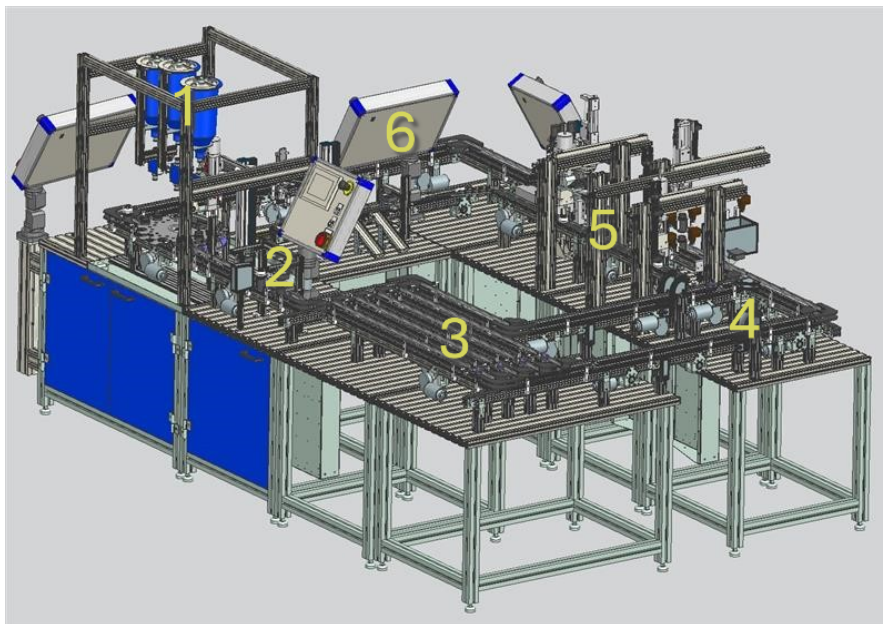


Abbildung 4.1: Stationen der SmartFactory-Anlage[5]

Abbildung 4.1 zeigt ein Modellbild der gesamten Anlage und veranschaulicht die räumliche Anordnung sowie die Verknüpfung der einzelnen Stationen. In den folgenden Abschnitten werden die Funktionsweisen der jeweiligen Module detailliert beschrieben.

4.1 Station 1: Abfüllung (AS)

Die Abfüllstation ist der Start des Kreislaufs der Anlage und hat die Aufgabe, Flaschen mit einer vorgegebenen Anzahl von Kugeln zu befüllen. Diese Anzahl wird durch die Erstellung von Aufträgen am Bedienpanel (HMI) oder Remote über Open Platform Communications Unified Architecture (OPC UA) auf einem Webinterface festgelegt. Hier kann zum Beispiel der Prozentsatz der Kugelfarbe (bei einem Maximum von 250 Kugeln) oder ein Absolutwert für die Anzahl der Kugeln festgelegt werden. Für das Abfüllen, hat die Abfüllung drei Container mit jeweils, roten, blauen und gelben Kugeln. Außerdem kann die Anzahl der abzufüllenden Flaschen angegeben werden. Diese Anzahl wird per “On-Demand”-TCP-Verbindung an das Modulare Produktionssystem gesendet. Die danach von dem modulare Produktionssystem gesendeten Flaschen werden durch einen Drehteller an verschiedenen Verarbeitungsstationen bewegt. Als erstes wird die Flasche, mit der des Auftrages angegebenen Anzahl an Kugeln, an der ersten Verarbeitungsstationen befüllt (4.2[1]). Dann wird in Verarbeitungsstationen 2 (4.2[2]) ein Deckel von dem modulare Produktionssystem angefordert und von diesem auf die Flasche gesetzt. Der Deckel wird danach in Verarbeitungsstationen 3 (4.2[3]) mit einen drehbaren Greifer verschraubt und an der vierten Verarbeitungsstationen (4.2[4]) wird der RFID-Tag im Deckel beschrieben. Der RFID-Tag enthält Datum und Uhrzeit der Abfüllung, sowie welcher Auftrag abgefüllt wurde. Jeder dieser Schritte wird parallel ausgeführt werden, um die Produktionszeit zu minimieren.

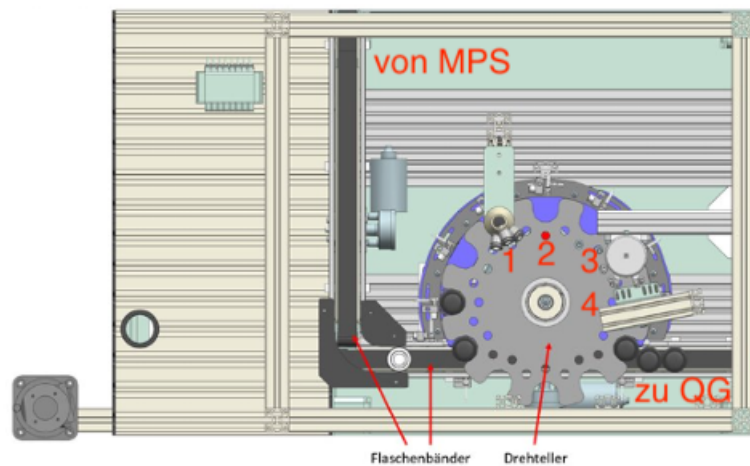


Abbildung 4.2: Abfüllstation[5]

4.2 Station 2: Quality Gate (QG)

Das Quality Gate gleicht die Auftragsdaten auf dem RFID-Tag mit dem Inhalt der Flasche ab. Hierzu wird an der ersten Verarbeitungsstation der RFID-Tag der Flasche eingelese, und an der zweiten Verarbeitungsstation prüft ein Farbsensor die farbliche Zusammensetzung. Stimmt die farbliche Zusammensetzung überein, wird an der dritten Verarbeitungsstation der RFID-Tag der Flasche mit „Gut“ beschrieben. Stimmt sie nicht überein, wird der RFID-Tag mit „Ausschuss“ beschrieben. Jede Verarbeitungsstation hat dafür einen eigenen Vereinzelter, um die richtige Verarbeitung der Flaschen zu gewährleisten. Der Anlagenteil dient zur Qualitätssicherung der abgefüllten Flaschen.

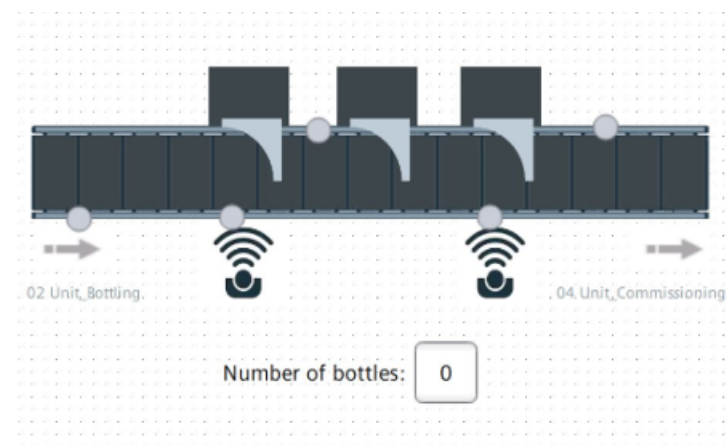


Abbildung 4.3: Quality Gate[5]

4.3 Station 3: Kommissionierbahnen (KB)

Die Kommissionierbahnen sortieren die vom Quality-Gate ankommenden Flaschen nach Auftragsnummer, hierzu wird zu Beginn der RFID-Tag gelesen und die Flasche auf eine von vier Bahnen sortiert. Durch das Quality-Gate als fehlerhaft markierte Flaschen oder Flaschen ohne lesbaren RFID-Chip werden direkt über ein Ausschussband an die nächste Station weitergegeben. Ist die Flasche “In Ordnung”, wird sie auf eine von vier Bändern sortiert. Die sortierten Flaschen bleiben in der Station, bis der entsprechende Auftrag abgeschlossen wurde oder der Bediener ein manuelles Entleeren der Bänder anfordert.

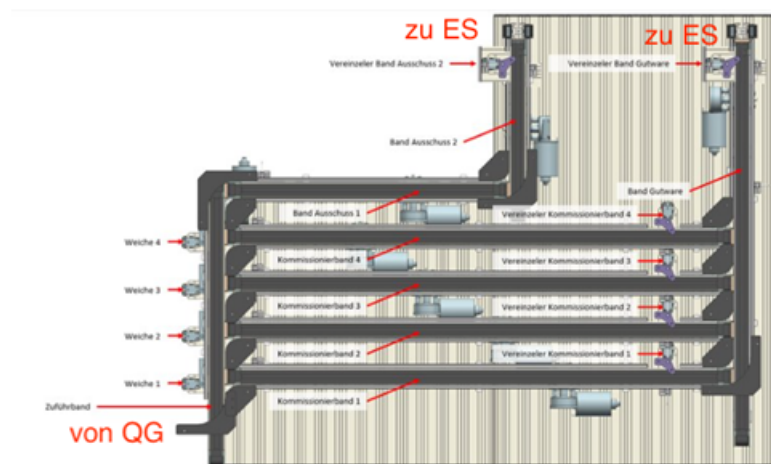


Abbildung 4.4: Kommissionierbahnen[5]

4.4 Station 4: Eckstation (ES)

Die Eckstation dient als Übergangspunkt, an dem die Flaschen von der Kommissionierbahn zur Recyclingstation weitergeleitet werden. Die Besonderheit der Eckstation sind, zwei Laufbänder, welche als hochklappbare Brücken fungieren, eine für die Abhandlung von Aufträgen und eine für aussortierte Flaschen. Das Hochklappen ermöglicht es, einen Schaltschrank in der Mitte der Anlage zu erreichen. Das Hochklappen der Laufbänder wird unterbunden, wenn noch Flaschen auf dem Laufband sind. Dies erfordert das Tracken der Flaschenanzahl und einen Stopper, um den Zulauf der Flaschen bei hochgeklappten Laufbändern zu verhindern und das Hochklappen bei vollem Laufband zu verhindern. Ist das Laufband

voll, wird der Zulauf durch den Stopper verhindert. Des Weiteren ist es möglich, ein entsprechendes Anforderungssignal an die Eckstation, zum Anheben der Brücke, zu senden, auf dessen Quittierung Flaschen warten müssen. Die Station übernimmt keine weiteren Funktionen und dient ausschließlich dazu, die Flaschen zwischen den beiden Stationen zu verfahren und Zugang zum Einschaltschrank zu gewährleisten.

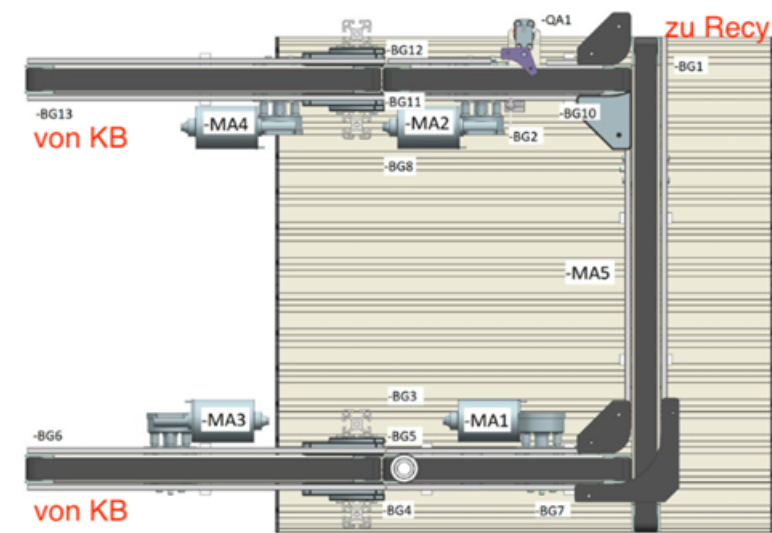


Abbildung 4.5: Eckstation[5]

Abbildung 4.5 zeigt den Aufbau der Eckstation. Die Kontaktpunkte für die versenkbaren Brücken befinden sich an den Sensoren BG4, BG5, BG11 und BG12. Diese Sensoren detektieren, ob die Brücken richtig in die ursprüngliche Position verfahren wurden.

4.5 Station 5: Recycling (Recy)

Die Recyclingstation übernimmt eine entscheidende Rolle im geschlossenen Produktionskreislauf der SmartFactory-Anlage. Sie verarbeitet alle ankommenden Flaschen, unabhängig davon, ob diese aus erfolgreich abgeschlossenen Produktionsaufträgen stammen oder aufgrund von Fehlern aussortiert wurden.

Zu Beginn wird jede Flasche von einem **RFID-Reader** erfasst, der ihre Identität und den Zustand überprüft. Diese Informationen werden an die Steuerung weitergeleitet, um den weiteren Verarbeitungsprozess zu optimieren. Anschließend

gelangt die Flasche zum **Deckelentferner**, wo ein präziser Greifer den Flaschendeckel entfernt. Der Deckel wird auf ein separates Laufband transportiert, das die Deckel für eine spätere Verarbeitung zur Verfügung stellt (4.6[1]).

Nach der Entfernung des Deckels fährt die Flasche weiter zum **Entleerer**. Hier wird sie von einem Greifer angehoben und über einen Trichter positioniert, wo die Kugeln entleert werden (4.6[2]). Der Trichter ist mit einem **Farbsensor** ausgestattet, der die Farbe der Kugeln erkennt. Diese Information wird an das Steuerungssystem weitergegeben, um die Kugeln entsprechend zu sortieren.

Das Rückführungssystem für die Kugeln wird **pneumatisch** betrieben. Je nach erkannter Farbe wird der passende Abfüllschlauch aktiviert, durch den die Kugeln in den entsprechenden Container an der Abfüllstation geschossen werden. Dies gewährleistet eine exakte farbliche Trennung und ermöglicht die Wiederverwendung der Kugeln im weiteren Produktionszyklus.

Während die erste Flasche entleert wird, bereitet das System die nächste Flasche für den Deckelentfernungsprozess vor, um eine hohe Effizienz sicherzustellen. Nach dem Entleeren wird die Flasche wieder auf das Laufband gesetzt und zum **Modularen Produktionssystem (MPS)** weitergeleitet. Die MPS macht die Flasche anschließend für weitere Produktionszyklen verfügbar.

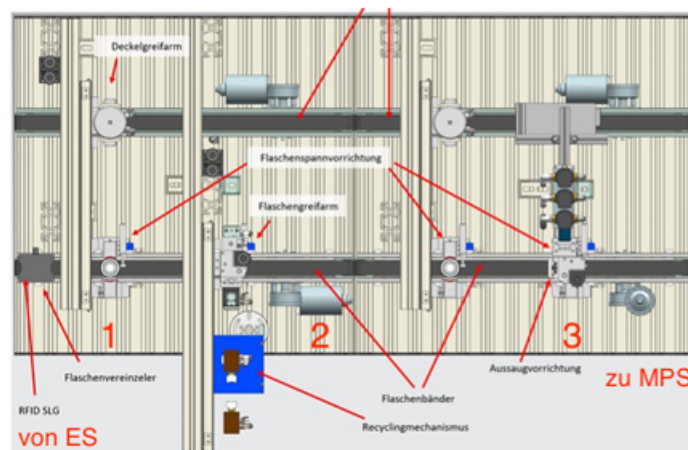


Abbildung 4.6: Recyclingstation[5]

4.6 Station 6: Modulares Produktionssystem (MPS)

Das Modulare Produktionssystem (MPS) ist eine zentrale Komponente der SmartFactory-Anlage und dient der Bereitstellung von Flaschen und Deckeln für die Abfüllstation. Es nimmt Deckel von der Recycling-Anlage entgegen, überprüft diese mithilfe eines RFID-Lesegeräts und eines Farbsensors auf Lesbarkeit und Farbe und sortiert fehlerhafte oder andersfarbige Deckel, wie rote, aus.

Auf Anforderung der Abfüllstation liefert das MPS die benötigte Anzahl an Flaschen und setzt mithilfe eines Greifers die passenden weißen Deckel auf die befüllten Flaschen. Fehlerhafte Flaschen oder Deckel werden aussortiert, um die Produktionsqualität zu sichern. Das System überwacht kontinuierlich den Materialvorrat, um Unterbrechungen zu vermeiden, und stellt relevante Daten über ein HMI bereit.

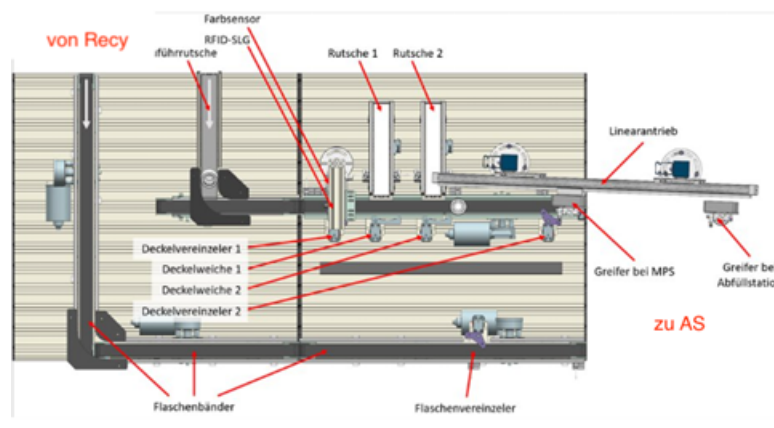


Abbildung 4.7: Modulares Produktionssystem[5]

4.7 Netzwerkverbund

Alle Anlagenteile sind via Ethernet mit der Einspeisung verbunden und bilden damit ein Subnetz. Dies ermöglicht eine effiziente und stabile Kommunikation zwischen den verschiedenen Stationen der Anlage und ermöglicht das Senden und Empfangen von Daten. Die gesamte Kommunikation der Anlage erfolgte "On-Demand" und passiert dadurch nur wenn sie von einer anderen Station angefordert wird.

4.8 Begriffsklärungen

Totally Integrated Automation (TIA):

Im nachfolgenden Abschnitt wird das programmieren von speicherprogrammierbaren Steuerungen (SPS) mit Hilfe von Totally Integrated Automation (TIA) mehrfach thematisiert. Dabei handelt es sich um die Entwicklungsumgebung für alle Automatisierungsprodukte von Siemens [6].

Eine speicherprogrammierbare Steuerung (SPS) ist ein digitales Gerät, das zur Steuerung von Maschinen und Anlagen eingesetzt wird. Die Prozesslogik für speicherprogrammierbare Steuerung (SPS) lässt sich in verschiedenen Programmiersprachen umsetzen, darunter FUP (Funktionsplan), AWL (Anweisungsliste) und SCL (Structured Control Language). Für die SmartFactory-Anlage wurde festgelegt, die Programmierung bedarfsgerecht in FUP und SCL durchzuführen.

Da eine speicherprogrammierbare Steuerung (SPS) als Echtzeitsystem zyklisch arbeitet, sie führt Berechnungen durch und aktualisiert Ausgaben in Intervallen, sind bei der Programmierung spezielle Anforderungen zu beachten. Dafür stehen verschiedene Konzepte zur Verfügung:

- **Funktion:** Ähnlich wie eine Methode in C#, ermöglicht eine Funktion die Verarbeitung von Übergabeparametern und die Berechnung entsprechender Ausgabewerte. Allerdings sind diese Ausgabewerte nicht persistent, da sie in jedem Programmzyklus neu berechnet werden.
- **Funktionsbaustein:** Ein Funktionsbaustein bietet dieselben Möglichkeiten wie eine Funktion, jedoch mit der zusätzlichen Fähigkeit, Daten über mehrere Zyklen hinweg zu speichern.
- **Datenbaustein:** Ein Datenbaustein dient der persistenten Speicherung von für das Programm relevanten Daten. Wird ein Funktionsbaustein erstellt, wird automatisch ein zugehöriger Datenbaustein zur Speicherung seiner Daten angelegt.
- **User Defined Datatype (UDT):** Ein UDT ist ein benutzerdefinierter Datentyp, der individuell vom Programmierer erstellt werden kann. Er ermöglicht die Zusammenfassung mehrerer Variablen in einem einzigen Datentyp und wird vor allem für die Softwarekonfiguration einzelner Module verwendet.

Konzeptionierung nach ISA-88:

Die Software der Anlage wurde gemäß dem ISA-88-Standard konzipiert. Dieser Standard legt fest, dass die Software in wiederverwendbare Module für Bauteile und Baugruppen strukturiert wird. Die Module wurden in die folgenden Typen unterteilt:

- **Control Modul:** Ein Control Modul repräsentiert einzelne Bauteile, die an jeder Station der Anlage vorkommen. Ein typisches Beispiel ist ein Sensor. Obwohl an der Anlage verschiedene Sensortypen verbaut sind, teilen sie alle die grundlegende Funktion, ein Bit umzuschalten. Das Control Modul `CM_BinarySensor` ist so entwickelt worden, dass es universell für alle Sensortypen von Siemens einsetzbar ist, wobei lediglich die Hardwareadresse des jeweiligen Sensors angegeben werden muss.
- **Equipment Modul:** Equipment Module bestehen aus einer Kombination mehrerer, aber mindestens einem Control Module¹ und bilden komplexere Baugruppen ab. Beispielsweise sind an den Förderbändern der SmartFactory-Anlage neben Sensoren auch Pneumatikzylinder verbaut. Da die Anzahl der Control Module pro Förderband variieren kann, wurde das Equipment Modul `EM_Conveyor` so gestaltet, dass es eine beliebige Anzahl an Control Modulen unterstützt und flexibel konfigurierbar ist.
- **Unit:** Die Unit bildet die spezifische Prozesslogik eines Anlagenmoduls ab und implementiert die erforderlichen Programmabläufe.

Die Struktur eines nach ISA-88 entworfenen Systems ist hierarchisch organisiert. In diesem Aufbau kommuniziert eine Unit ausschließlich mit Equipment Modulen, während Equipment Module sowohl mit Units als auch mit Control Modulen interagieren. Control Module hingegen stehen nur in Verbindung mit Equipment Modulen.

¹Vgl. ResearchGate, https://www.researchgate.net/figure/SA-88s-physical-model-and-an-example-production-allocation_fig2_308384237/, zuletzt aufgerufen am 11. Januar 2025.

5 Projekt SmartFactory@EH-Anlage

5.1 Projekt Smart Factory-Anlage: Aufgabenstellung

Die SmartFactory-Anlage ist eine speziell für Schulungszwecke konzipierte Automatisierungsanlage, die von der Abteilung Expert House entwickelt und betrieben wird. Sie beinhaltet verschiedene Produkte aus dem Siemens-Produktkatalog wie zum Beispiel speicherprogrammierbare Steuerungen, Lichtsensoren und Motoren. Meine Aufgabe bestand darin, zusammen mit 18 weiteren Studierenden die Weiterentwicklung für diese Anlage durchzuführen. Dies umfasste sowohl die vollständige Ausbesserung und Behebung bereits bekannter Mängel, welche in einer List of Open Points (LOP) festgehalten wurden, als auch das Testen und Verbessern der Robustheit der Software. Die Software wurde nach dem ISA-88-Standard umgesetzt und alle Erweiterungen werden auch nach diesem hinzugefügt werden. Zur Programmierung wurde die Siemens-Automatisierungssoftware „TIA-Portal“ verwendet. Eine weitere Anforderung war es, die umfassenden Dokumentationsunterlagen in Form von Word-Dokumenten für nachfolgende Studierende auszubessern und gegebenenfalls mit den neuen Funktionen zu erweitern. Die Nutzung des ISA-88-Standard soll gewährleisten, dass die Anlage ohne einen Ansprechpartner aus unserem Projektteam weiterentwickelbar ist.

5.2 Projekt Smart Factory-Anlage: Zielsetzung

Das Ziel am Ende des Praxissemesters war es, die SmartFactory-Anlage weiterzuentwickeln, ihre Robustheit zu erhöhen und uns ein tiefes Verständnis für komplexe Systeme eigenständig zu erarbeiten, um uns eine Grundlage für zukünftige Ausbildungs- und Weiterbildungsmodule zu schaffen. Dafür müssen für jedes Anlagenmodul bekannte Fehler und offene Punkte aus der List of Open Points (LOP)

abgearbeitet werden, sowie bei der Erkennung neuer Fehler, diese zur List of Open Points (LOP) hinzugefügt werden. Des Weiteren müssen die Bedingungen wie eine einheitliche HMI-Schnittstelle und eine leichte Bedienung der Anlage umgesetzt werden.

5.3 Projekt SmartFactory-Anlage: Projektorganisation

Das Projektteam besteht aus 9 Informatikstudenten und 10 Elektro- und Informationstechnikstudenten, welche über die verschiedenen Anlagenteile und übergreifende Aufgaben verteilt wurden, sowie drei Fachbetreuern. Diese haben uns als “Product Owner” im Projekt unterstützt und geholfen.

Das gesamte Projekt wurde auf der agilen Methode **Scrum** basierend durchgeführt. Dabei wurden jedoch Anpassungen vorgenommen, die dem Projektablauf und den spezifischen Umständen gerecht wurden. Wie bereits in Abschnitt 1, „Einleitung“, erläutert, wurde die Arbeitsphase im Expert House sowohl bei uns Informatikern als auch bei den Elektro- und Informationstechnikern durch die **SPE-Phase** unterbrochen, allerdings zu unterschiedlichen Zeitpunkten. Dies führte zu einem geteilten Arbeitszeitplan: In den ersten acht Wochen arbeiteten alle gemeinsam am Projekt, in den darauffolgenden acht Wochen nur die Elektro- und Informationstechniker, und in den abschließenden acht Wochen waren ausschließlich wir Informatiker beteiligt.

Trotz der Abweichungen von den klassischen Scrum-Praktiken war dieser Ansatz sinnvoll. Durch die iterative Arbeitsweise und die klare Struktur von Scrum konnten wir den Überblick über die aktuellen Aufgaben besser behalten. Zur Organisation der Aufgaben und Projektmeilensteine half uns insbesondere ein **Kanban-Board**, das die Aufgaben nach den Kategorien strukturierte. Dieses visuelle Hilfsmittel erleichterte die Nachverfolgbarkeit von Fortschritten erheblich und förderte die Verständlichkeit während der regelmäßigen Meetings.

Die Kombination aus Scrum-Elementen und der Verwendung des Kanban-Boards sorgte für eine verbesserte Teamkommunikation und einen reibungsloseren Workflow. So konnte jeder Beteiligte stets nachvollziehen, welche Aufgaben anstanden,

bearbeitet wurden oder bereits abgeschlossen waren. Besonders in der finalen Phase, als die Teams unabhängig voneinander arbeiteten, erwies sich diese Arbeitsweise als vorteilhaft, da sie die Selbstorganisation förderte und klare Prioritäten setzte.

Dies führte dazu, dass an den Übergabepunkten große und kleine Änderungen klar an das andere Projektteam übergeben werden mussten. Gleichzeitig war es essenziell, die Dokumentation stets auf dem aktuellen Stand zu halten, um eine reibungslose Weiterarbeit zu ermöglichen.

Dadurch stimmten wir uns gemeinsam, Betreuer und Auszubildende, in Daily-Standup-Meetings ab. Wir arbeiteten in Zwei-Wochen-Sprints, die jeweils mit einer Vorstellung der Ergebnisse an die Stakeholder endeten. Diese Präsentation ist ein wichtiges Mittel um unseren Fortschritt an der Anlage zu zeigen und Punkte für die nächste Bearbeitung festzulegen. Eine zentrale Rolle spielte die "List of Open Points" (LOP), die zur Dokumentation von bestehenden Fehlern und Mängeln, aber auch Funktionsanforderungen und Verbesserungen genutzt wird.

Die LOP war in Dringlichkeitsstufen unterteilt:

- **Sehr wichtig:** Prozessbeendende Fehler oder solche, die mechanische Schäden verursachen könnten.
- **Mittel:** Fehler, die den Prozess nicht vollständig stoppen, aber zu Fehlern in der Verarbeitung von Flaschen führen.
- **Niedrig:** Fehler mit geringeren Auswirkungen, z. B. das Herausfallen einer Kugel bei der Initialisierung der Abfüllstation.

Die Stakeholder legten diese Prioritäten fest. Für jeden Sprint suchten sich die Verantwortlichen der jeweiligen Station die wichtigsten Aufgaben mit der höchsten Priorität heraus und arbeiteten diese ab.

Gerade zu bearbeitende Aufgaben und bereits abgeschlossene wurden in ein Kanban-Board eingetragen. Dies half, den Fortschritt zu visualisieren und machte in den Daily-Meetings deutlich, welche Aufgaben eventuell vom Plan abwichen. Dadurch konnten notwendige Anpassungen schnell besprochen und vorgenommen werden, um weiterhin das Projektziel in der gegebenen Zeit zu erreichen.

Das Kanban-Board wurde in folgende Bereiche unterteilt:

- **Aufgaben:** Hier befinden sich alle Aufgaben, die im aktuellen Sprint erledigt werden müssten, mit denen sich aber noch niemand befasst hat.
- **In Bearbeitung:** Hier wurden die aktuell bearbeiteten Aufgaben gesammelt.
- **Internes Review:** Wurde eine Aufgabe abgeschlossen, wurde das Ergebnis zunächst durch einen weiteren Studierenden korrekturgelesen und auf Verständlichkeit überprüft.
- **Ready for Review:** Eine von einem Studierenden als in Ordnung befundene Aufgabe wird noch ein weiteres Mal durch einen Betreuer geprüft.
- **Nacharbeiten:** Wurde eine durchgeführte Aufgabe entweder von einem Betreuer oder einem Studierenden als nicht in Ordnung befunden, oder es traten im Projektverlauf Probleme damit auf, ist sie in diesem Bereich zu finden.
- **Erledigt:** In diesem Bereich befinden sich schließlich komplett abgeschlossene Aufgaben.

Des Weiteren haben wir mit dem Siemens-Multiuser-Server gearbeitet, eine angepasste Lösung für die Zusammenarbeit an Automatisierungsprojekten. Das bedeutet, dass jeder Studierende eine lokale Instanz des Projekts hatte, diese bearbeiten und an der Anlage testen konnte. Große Änderungen konnten dann auf einen GIT-ähnlichen Server hochgeladen werden, von dem alle anderen Instanzen diese Änderungen wieder herunterladen konnten. Außerdem konnte man über ein File-Locking-System bei Dateien, an denen man gerade arbeitete, festlegen, ob ein Konflikt besteht, wenn zwei Studierende gleichzeitig an derselben Datei arbeiten. Dies hat "Merge-Konflikte" und den Verlust von Code verhindert, dass die Zusammenarbeit erleichtert und ermöglicht eine grundlegende Versionskontrolle.

5.4 Projekt SmartFactory-Anlage: Projektverlauf

5.4.1 Auftaktwoche:

Noch vor dem offiziellen Start des Projekts wurde in meiner Abteilung eine Auftaktwoche organisiert, die dazu diente, uns auf die bevorstehenden Aufgaben vor-

zubereiten. Der Schwerpunkt lag darauf, unser Wissen und unseren Umgang mit dem TIA Portal aufzufrischen. Obwohl wir als dual Studierende bereits einen zweiwöchigen Kurs zur Nutzung der Software absolviert hatten, fehlte uns durch ein Semester ohne praktische Anwendung die nötige Routine. In dieser Woche bestand unsere Aufgabe darin, die Steuerung eines Aufzugs mit fünf Stockwerken zu programmieren. Dabei wurden uns auch Programmierkonzepte wie die „State Machine“ vermittelt, die später eine wichtige Rolle bei der Anwendung an der SmartFactory-Anlage spielten.

5.4.2 Hands-On SmartFactory-Anlage:

Nach der Auftaktwoche wurden wir zufällig in Teams von zwei bis drei Personen eingeteilt, um die SmartFactory-Anlage in Betrieb zu nehmen. Ziel war es, am Ende der Woche eine Flasche, mit dem bereits bestehenden Code, vollständig durch die Anlage zu führen. Jedem Team wurde dabei eine spezifische Station zugewiesen. Ich war für die Abfüllstation zuständig. Die Aufgabenstellung war bewusst vage gehalten, um uns die Möglichkeit zu geben, die Anlage eigenständig kennenzulernen. Das Ergebnis bei der Abfüllstation bestand darin, das angefragte Flaschen und Deckel richtig für erstellte Aufträge verwendet wurden und ein betriebsbeendender Fehler der Laufbänder behoben wurden. Allerdings wurden auch einige neue Punkte, wie die fehlerhafte Kalibrierung des Drehtellers oder die fehlende Initialisierungsfahrt, in die LOP (List of Open Points) nachgetragen. Nach Abschluss dieser Aufgabe führten wir ein „Lessons Learned“ durch. Die wichtigste Erkenntnis hierbei war, dass die Anlage ohne einen sorgfältig erarbeiteten Plan und ein durchdachtes Konzept nicht effektiv in Betrieb genommen werden kann.

5.4.3 Bibliotheksverantwortlicher (erweiternder Aufgabenbereich):

Mir wurde eine zusätzliche Aufgabe, in Form des Bibliotheksverantwortlichen, zugewiesen. Die Aufgabe bestand darin, die Projektbibliothek des Multiuser-Servers konsistent zu halten. Dies ist notwendig, da Programmbausteine gemäß dem ISA-88-Standard, wie beispielsweise *Control Modules* oder *Equipment Modules*, von mehreren Instanzen im Projekt genutzt werden können.

Ein Lichtsensor kann beispielsweise Teil eines Förderbands oder eines Drehtellers sein. Um doppelten Code zu vermeiden und die Erweiterbarkeit und Wiederverwendbarkeit so hoch wie möglich zu halten, wird im Code, wenn ein Lichtsensor benötigt wird, der Baustein aus der Projektbibliothek verwendet. Dieser Baustein wird im Projekt als neue Instanz des Bausteins aus der Projektbibliothek eingefügt. Möchte man den genannten Lichtsensor aktualisieren, werden alle Instanzen im Projekt sowie in der Projektbibliothek aktualisiert.

Da die Projektbibliothek versionsabhängig arbeitet, kann es vorkommen, dass jemand am Förderband eine Änderung vornimmt, die nicht mit der neuen Version des Lichtsensors kompatibel ist. Dies würde zu einem Ausfall der Lichtsensoren am Förderband führen. Ebenso könnte es passieren, dass auf dem Förderband noch eine ältere Version des Lichtsensors ohne die aktuellen Änderungen verwendet wird. Dies führt zu Inkonsistenzen, die von mir bereinigt werden mussten.

5.4.4 Flaschenmanagement in der Unit:

Das Flaschenmanagement der Abfüllstation hatte zwei wesentliche Schwachstellen: Erstens wurden nicht immer ausreichend Flaschen geliefert, um einen Auftrag vollständig zu erfüllen. Beispielsweise wurden bei einem Auftrag über zwölf Flaschen nur neun bereitgestellt, wodurch die Abfüllung anhielt und auf die fehlenden Flaschen wartete (siehe Abbildung 4.2). Zweitens wurden die Auftragsdaten nicht remanent gespeichert, was bedeutete, dass nach jedem Neustart oder Kaltstart der Steuerung alle Flaschen aus dem Drehteller entfernt werden mussten, bevor ein neuer Auftrag gestartet werden konnte. Dies führte zu langen Stillstandszeiten und erschwerte die Bedienung der Anlage erheblich. Ein zusätzliches Problem ergab sich aus der Tatsache, dass der Drehteller nach einem Neustart erkannte, dass sich Flaschen in ihm befanden, jedoch keine gültigen Auftragsdaten vorlagen. Dadurch geriet der Drehteller in einen ungültigen Zustand und konnte die Aufgaben an den einzelnen Positionen nicht korrekt ausführen, was die gesamte Anlage blockierte.

Eine der zentralen Herausforderungen bestand darin, das Flaschenmanagement so anzupassen, dass sowohl die Lieferung unvollständiger Flaschenmengen als auch der Verlust von Auftragsdaten behoben werden konnte, ohne dabei die bestehenden Prozesse zu beeinträchtigen. Es war erforderlich, den Arbeitsfluss auch bei

unvollständigen Flaschenlieferungen aufrechtzuerhalten und die Zustandsinformationen des Drehtellers nach einem Neustart korrekt zu initialisieren. Zudem musste eine Lösung gefunden werden, um unbekannte Flaschen im Drehteller zuverlässig zu erkennen und zu entfernen.

Um die Probleme bei der Flaschenlieferung zu beheben, wurde die Steuerung der Abfüllstation angepasst. Wenn innerhalb von sieben Sekunden keine neuen Flaschen geliefert werden, werden die bereits vorhandenen Flaschen im Drehteller dennoch abgefüllt und verschlossen. Dies gewährleistet einen kontinuierlichen Arbeitsfluss und minimiert Stillstandszeiten. Für das Problem der nicht remanenten Auftragsdaten wurde die Steuerung so erweitert, dass die Zustandsinformationen des Drehtellers nach einem Neustart korrekt initialisiert werden. Flaschen ohne gültige Auftragsdaten werden nun automatisch als unbekannt markiert und aus dem Drehteller entfernt. Diese Flaschen werden anschließend an das Quality Gate weitergeleitet, wodurch ein blockierter Zustand der Anlage verhindert wird. Die vorgenommenen Änderungen verbesserten die Effizienz der Abfüllstation erheblich und steigerten sowohl die Benutzerfreundlichkeit als auch die Robustheit der Anlage.

5.4.5 Kommunikation:

Ein weiteres Problem bestand darin, dass die Kommunikation zwischen den einzelnen Stationen der Anlage auf „On-Demand“-Basis ausgelegt war. Das bedeutete, dass jede Station nur zu einem bestimmten Zeitpunkt mit einer anderen Station kommunizierte. Wenn jedoch eine Station nicht bereit war, die Kommunikation zu empfangen, ging diese verloren. Dies führte dazu, dass Anlagenteile nicht mehr synchron arbeiteten, die Anlage in einen fehlerhaften Zustand geriet und die Produktion gestoppt wurde.

Um dieses Problem zu lösen, musste ein neuer Baustein mit einem Watchdog-Timer eingeführt werden, der die Kommunikation zwischen den Stationen überwacht. Der neue Kommunikationsbaustein ersetzte den alten und die Logik wurde zu einer Acknowledge-Logik umgeschrieben. Bis ein Acknowledge von der anderen Station kommt, wird die Kommunikation nicht als abgeschlossen betrachtet. Dies führte dazu, dass die Produktion nach Wiederherstellung eines fehlerfreien

Zustands der Anlage fortgesetzt werden konnte und keine Kommunikation verloren ging. Dadurch konnte auch die Flaschenproduktion nach einem Fehlerzustand automatisch von der Anlage fortgesetzt werden.

Ein weiteres Problem bestand darin, dass es für andere Anlagenteile, vor allem aber für die Kommissionierung, nicht möglich war, Flaschen, welche von der Qualitätsprüfung am Quality-Gate als ungenügend beschrieben wurden, nachzubestellen. Diese Flaschen wurden aussortiert, was dazu führte, dass in der Kommissionierung nicht vollständige Aufträge vorlagen.

Die Herausforderung bestand darin, eine nahtlose Integration der Nachbestellfunktion in das bestehende System zu gewährleisten, ohne die laufenden Prozesse zu stören. Es war notwendig, sicherzustellen, dass die Kommunikation zwischen den verschiedenen Anlagenteilen zuverlässig und effizient abläuft. Zudem musste die Nachbestellfunktion so gestaltet werden, dass sie flexibel auf unterschiedliche Anforderungen reagieren kann, ohne dass umfangreiche Anpassungen erforderlich sind.

Um dieses Problem zu lösen, wurde ein neuer Kommunikationsbaustein für die Kommunikation mit der Kommissionierung und eine neue Nachbestellfunktion implementiert. Der Kommunikationsbaustein war bereits vorhanden und es musste nur eine Kopie mit der IP der Kommissionierung in das Projekt eingefügt werden. Für die Nachbestellfunktion konnte die bestehende Funktion zum Bestellen von Aufträgen verwendet und direkt in die Unit integriert werden. Anstatt jedoch Eingaben vom HMI für die Anzahl der Flaschen und Kugeln zu erhalten, wurden hier die Auftragsdaten der noch nachzubestellenden Flaschen verwendet.

5.4.6 Drehteller:

Ein weiteres Problem bestand darin, dass die Taster auf dem HMI-Bediener-Panel zur Steuerung des Drehtellers nicht korrekt eingestellt waren. Dies führte dazu, dass bei der Kalibrierung des Drehtellers ein Wechsel zwischen „drehe rechts“ und „drehe links“ nicht möglich war und zu einem nicht quittierbaren Fehler an der Abfüllstation führte. Beispielsweise drehte sich der Drehteller bei Betätigung des Knopfes „drehe rechts“ zwar in die gewünschte Richtung, setzte seine Bewegung jedoch auch nach dem Loslassen des Knopfes fort. Dies widersprach den Anforderungen an Sicherheit, Benutzerfreundlichkeit und Robustheit der Anlage.

Eine große Herausforderung bei dieser Aufgabe war es, die Fehlerursache zu identifizieren. Zunächst vermuteten wir, dass die Kommandos zum Anhalten des Drehtellers nicht korrekt gesendet wurden. Allerdings stellte sich heraus, dass die Kommandos nicht nur fehlerhaft gesendet wurden, sondern gar nicht implementiert waren. Stattdessen wurde eine Nebenwirkung des Drehtellers als Absolutwertgeber ausgenutzt, denn der Befehl, einen neuen Absolutwert an der aktuellen Position des Drehtellers zu setzen, wurde mit seiner höheren Priorität verwendet, um das Drehen nach links oder rechts zu stoppen.

Es musste ein neuer Befehl auf der untersten Steuerungsebene des Drehtellers implementiert werden, der das Stoppen des Drehtellers ermöglicht. Hierfür wurde im Control-Modul `CM_PositioningDrive` das UDT (User Defined Type) um ein neues Kommando erweitert. Dies war erforderlich, damit im Equipment-Modul `EM_Drehteller` ein entsprechender neuer Befehl hinzugefügt werden konnte.

Als Technologieobjekt besitzt der Drehteller bereits vorgeschriebene Bibliotheksfunktionen. Durch das Einfügen dieser neuen Funktionalität konnte der Halte-Befehl korrekt in die Steuerung der Taster integriert werden. Mit der korrekten Implementierung des Halte-Befehls war es außerdem möglich, in der Anlagenlogik den zuvor missbräuchlich genutzten Befehl zum Setzen eines neuen Absolutwerts zu ersetzen. Dadurch konnte auch die Kalibrierung des Drehtellers ohne Benutzereingaben beim Kaltstart ermöglicht werden, da der remanent gespeicherte Absolutwert nicht beim Neustart überschrieben wurde und der Drehteller sich dadurch selbstständig korrigieren kann. Beim Anfahren der Drehscheibe sowie bei einem Not-Halt trat ein Fehler im Technologieobjekt auf, der nicht quittiert werden konnte. Infolgedessen verließ die Drehscheibe ihre Ausgangsposition und musste neu kalibriert werden, was dazu führte, dass die Flaschen und die Auftragsposition nicht mehr übereinstimmten. Zudem wurde die Position des Auftrags nicht korrekt übergeben, da im Fehlerzustand keine steigende Flanke des Drehtisches weitergegeben wurde.

Ein weiteres Problem bestand darin, dass während eines Not-Halts die Spannung an der EP-Klemme abgeschaltet wurde, wodurch jegliche Kontrolle über die Drehscheibe verloren ging. Dies stellte eine Schwachstelle dar, da weder `MC_Power` noch `MC_Halt` Einfluss auf den Prozess hatten.

Erarbeitete Lösungsansätze waren die Verzögerung der 24V-Versorgung am Antrieb, um ein kontrolliertes Anhalten zu ermöglichen, sowie die Anpassung der

Verdrahtung der Failsafe-Digitaleingänge am Sinamics-System – entweder durch eine Safety-Schaltung oder durch eine direkte Verdrahtungsänderung. Aus Zeitgründen wurde sich jedoch gegen diese Ansätze entschieden.

Aufgrund des Zeitdrucks wurde sich jedoch gegen diese Ansätze entschieden. Letztendlich wurde das Problem jedoch anders gelöst: Eine automatische Nachbestellung wird nun ausgelöst, wenn sich kein Deckel auf der Flasche befindet. Dies erfolgt durch einen speziellen Baustein für die Nachbestellung und dem RFID-Leser, welcher prüft, ob sich ein RFID-Chip unter sich befindet. Darüber hinaus wurde die statische Variable zur Fehlerspeicherung durch eine temporäre Variable (Temp) ersetzt, da der ursprüngliche Stat-Wert über einen Zyklus hinweg bestehen blieb und nach dem Quittieren den Fehler erneut in den Status schrieb.

5.4.7 Kugelabfüllung:

Eine Kundenanforderung bestand darin, dass die Abfüllung schneller abfüllt, da durch Aufträge mit vielen Kugeln die Anlage ausgebremst wurde und durch die einzelne Abfüllung der drei Farben nacheinander sehr langsam war. Die Herausforderung bestand darin, die Abfüllung zu beschleunigen, ohne die Farben zu mischen. Eine Beschleunigung des Timers führte jedoch zu einem Einklemmen der Kugeln, was zu einem blockierten Zustand der Anlage führte. Dieser Zustand konnte nur durch manuelles Eingreifen behoben werden.

Die Lösung bestand darin, den Vereinzler so anzupassen, dass er automatisch weiterarbeitet, wenn eine Kugel eingeklemmt wird. Dadurch konnte die Abfüllung beschleunigt werden, ohne dass die Kugeln sich gegenseitig blockierten. Die Logik wurde entsprechend angepasst, um Fehler abzufangen und sicherzustellen, dass sich der Vereinzler in die andere Richtung bewegt, wenn eine Kugel eingeklemmt ist. Zudem wurden die Fehler aus der Fehlerliste im Equipment-Modul der Abfüllstation entfernt.

5.4.8 Kugelrückführung:

Eine Kundenanforderung bestand darin, die Lautstärke und die Genauigkeit der Kugelrückführung der Recyclingstation zu verbessern. Ursprünglich wurde die Kugelrückführung durch einen angebrachten Rüttelmotor realisiert, der den Einlauf

der Kugeln in einen Trichter erleichtern sollte. Der Rüttelmotor und die angebrachte Pneumatik zum Zurückschießen der Kugeln führten jedoch zu einer sehr lauten Arbeitsumgebung. Um die Lautstärke zu reduzieren und die Genauigkeit der Kugelrückführung zu erhöhen, wurde ein neues Konzept entwickelt. Dieses Konzept sah vor, dass die Kugeln durch einen Drehteller mit 20 Löchern in den Trichter zurückgeführt werden. Der Drehteller wird durch einen Schrittmotor angetrieben, der die Kugeln in den Trichter schiebt. Der Schrittmotor wird durch einen Binärsensor gesteuert, der erkennt, ob sich eine Kugel im Trichter befindet. Wenn eine Kugel erkannt wird, stoppt der Schrittmotor und der Farbsensor wird aktiviert. Der Farbsensor erkennt die Farbe der Kugel und leitet sie an die entsprechende Position weiter. Sobald die Kugel die Position erreicht hat, wird der Schrittmotor erneut aktiviert und schiebt die Kugel in den Trichter zurück.

Eine Herausforderung bestand darin, dass die Funktion `move_relative` nicht korrekt funktionierte, da der Schrittmotor nur 1,8 Grad drehte und die 12 Löcher des Drehtellers eine Drehung von 30 Grad erforderten, was nicht durch 1,8 teilbar ist. Zudem war der Positionsgeber generell ungenau, was dazu führte, dass sich der Drehteller mehr oder weniger als die erforderlichen 18 Grad drehte.

Die Lösung bestand darin, den eigentlichen Binärsensor des `EM_PelletReturn` in das Input-Mapping korrekt einzufügen und nutzbar zu machen. Zudem wurde die Geschwindigkeit für die `move_jog`-Funktion des Technologieobjekts in der Softwarekonfiguration anstatt als statische Variable im `CM_PositioningDrive` definiert, um eine leichte Änderung der Geschwindigkeit zu ermöglichen. Anstelle von `move_relative` wurde `move_jog` verwendet. Wenn der Sensor aktiviert ist, stoppt der Drehteller, der Farbsensor wird aktiviert und der Drehteller läuft weiter, bis der Sensor wieder deaktiviert ist.

5.5 Projekt SmartFactory-Anlage: Ergebnis

Der finale Funktionsumfang des Projekts lässt sich zum Zeitpunkt der Erstellung dieses Berichts noch nicht vollständig abschätzen, da einige Aspekte der Entwicklung und Implementierung noch in Arbeit sind. Dennoch konnte das Team bereits bedeutende Fortschritte erzielen und wesentliche Meilensteine gemäß den ursprünglichen Vorgaben und dem Projektplan erreichen. Besonders hervorzuheben

ist, dass alle kritischen Fehler, die den Betrieb der Abfüllstation maßgeblich beeinträchtigen oder den Prozess unterbrechen konnten, erfolgreich identifiziert und behoben wurden. Lediglich kleinere Bugs, die keine direkte Auswirkung auf den kontinuierlichen Betrieb haben, verbleiben noch zur Bearbeitung. Damit wurde eines der zentralen Projektziele, die Robustheit der gesamten Anlage erheblich zu steigern, erfolgreich realisiert.

Der Fokus der weiteren Arbeiten liegt nun darauf, die Benutzerfreundlichkeit des Systems weiter zu verbessern und die Anlage so zu gestalten, dass sie zukünftig leicht erweitert und angepasst werden kann. Die Verbesserung der Bedienoberflächen, die Optimierung der Steuerungslogik und die Anpassung der Kommunikationsprozesse zwischen den Stationen sind dabei wichtige Arbeitsbereiche, die bereits angestoßen wurden.

Aus den regelmäßigen Teammeetings sowie dem wertvollen Feedback der Projektmitglieder und betreuenden Lehrkräfte geht klar hervor, dass auch an den anderen Stationen der Anlage signifikante Fortschritte erzielt wurden. Trotz der Herausforderungen, die sich durch die komplexen Abhängigkeiten der Systeme ergeben, konnte das Projektteam durch gezielte Koordination und klare Zielsetzungen einen stabilen Fortschritt sicherstellen. Sollte es dennoch vorkommen, dass der Projektfortschritt bis zum Abschluss des Praxissemesters nicht alle gesetzten Ziele vollständig erreicht, gewährleisten die sorgfältig erstellten Konzept- und Entwurfsdokumente eine solide Grundlage für die Weiterentwicklung des Projekts durch zukünftige Studierende. Diese Dokumente enthalten detaillierte Beschreibungen der bisherigen Arbeiten, der zugrunde liegenden Systemarchitektur sowie Empfehlungen für künftige Implementierungen.

Besonders bemerkenswert ist, dass durch die vorgenommenen Änderungen an der Abfüllstation nicht nur die Effizienz des Produktionsprozesses erheblich gesteigert werden konnte. Auch die Benutzerfreundlichkeit und Robustheit des gesamten Systems wurden signifikant verbessert. Die Reduktion von Störungen, die Erhöhung der Systemstabilität und die klar strukturierte Benutzerführung tragen dazu bei, dass die Anlage zuverlässiger und einfacher zu bedienen ist. Die ergriffenen Maßnahmen bilden somit eine wertvolle Grundlage für den langfristigen Erfolg des Projekts.

6 Persönliches Fazit zum Praktikum

Rückblickend möchte ich auf meine persönlichen Eindrücke des Praxissemesters bei der Siemens AG eingehen. Für mich war die Arbeit in einem großen Industrieunternehmen eine völlig neue und spannende Erfahrung. Besonders die Arbeitsweise in meiner Fachabteilung, dem Expert House, hat mich beeindruckt. Hervorheben möchte ich die täglichen Meetings, die es jederzeit ermöglichten, einen ganzheitlichen Überblick über den Projektfortschritt zu erhalten.

Ich habe gelernt, dass verschiedene Projektmanagement-Modelle, wie z. B. das Wasserfallmodell oder Kanban, nicht zwingend getrennt voneinander eingesetzt werden müssen, sondern dass eine sinnvolle Kombination der Methoden je nach Situation große Vorteile bringen kann. Auch Konzepte wie „Lessons learned“ und „Expert Talks“ empfand ich als äußerst sinnvoll und hilfreich.

Etwas ungünstig war es aus meiner Sicht, dass die SPE-Phase mitten in der Abteilungsphase stattfand und nicht zu Beginn oder am Ende. Dadurch ging meines Erachtens ein Teil des Blicks auf den Projektfortschritt der SmartFactory-Anlage verloren. Neben den organisatorischen Aspekten empfand ich die Programmierung mit dem TIA-Portal als besonders interessant. Diese Art der Programmierung unterscheidet sich deutlich von klassischen Programmiersprachen und erfordert, beispielsweise durch den zyklischen Programmablauf und das Bausteinkonzept, besondere Aufmerksamkeit.

Mein persönliches Highlight war jedoch die Arbeit an einer Industrieanlage. Es ist ein besonderes Erfolgserlebnis, wenn ein gut durchdachtes Konzept in einen funktionierenden Code umgesetzt wird und dadurch ein reibungsloser Prozessablauf an einer Anlage mit unzähligen Sensoren und Aktoren ermöglicht wird. Wer neben der reinen Informatik auch an Mechanik und Elektrotechnik interessiert ist, dem kann ich ein Praxissemester im Expert House bei der Siemens AG sehr empfehlen.

Literaturverzeichnis

- [1] Ernst Klett Verlag (2004): *Infoblatt Siemens AG*. Verfügbar unter: <https://www.klett.de/alias/1036900>. Zugriff: 5. Januar 2025.
- [2] Merkur (2022): *Siemens – Geschichte, Aktie und Tätigkeitsfelder*. Verfügbar unter: <https://www.merkur.de/wirtschaft/siemens-geschichte-aktie-und-taetigkeitsfelder-91268844.html>. Zugriff: 19. Januar 2024.
- [3] Siemens, 2023, zitiert nach de.statista.com: *Umsatz von Siemens SE seit 2005*. Verfügbar unter: <https://de-statista-com.thn.idm.oclc.org/statistik/daten/studie/73827/umfrage/umsatz-von-siemens-seit-2005/>. Zugriff: 5. Januar 2025.
- [4] ResearchGate: Verfügbar unter: https://www.researchgate.net/figure/SA-88s-physical-model-and-an-example-production-allocation_fig2_308384237/. Zugriff: 11. Januar 2025.
- [5] Siemens (2022): *Pflichtenheft-SmartFactory*. Internes Dokument.
- [6] Liam Bee (2022): *PLC and HMI development with Siemens TIA Portal: develop PLC and HMI programs using standard methods and structured approaches with TIA Portal V17*. Packt Publishing, Limited.