



Fakultät Informatik

**Bewertung bekannter
Ad-hoc-Routing-Protokolle zur
prototypischen Realisierung einer
mobilen Messenger-Applikation**

Wissenschaftliches Schreiben im Studiengang Informatik

vorgelegt von

Jan-Eric Gedicke

Matrikelnummer 3653446

Erstgutachter: Marcus Fiebig

Zweitgutachter:

© 2025

Inhaltsverzeichnis

1	Einleitung (evtl noch etwas umschreiben)	1
2	Vorstellung der Siemens AG (noch umschreiben)	2
3	Vorstellung meiner Fachabteilung (noch umschreiben)	3
4	Projekt Smart Factory-Anlage	4
4.1	Projekt Smart Factory-Anlage: Aufgabenstellung (evtl noch umschreiben)	4
4.2	Projekt Smart Factory-Anlage: Zielsetzung	5
4.2.1	Station 1: Modulares Produktionssystem (umschreiben)	6
4.2.2	Station 2: Abfüllung (geschrieben)	6
4.2.3	Station 3: Quality Gate (geschrieben)	7
4.2.4	Station 4: Kommissionierbahnen (geschrieben)	7
4.2.5	Station 5: Eckstation (geschrieben)	7
4.2.6	Station 6: Recycling (geschrieben)	8
4.2.7	Netzwerkverbund (geschrieben)	8
4.3	Projekt SmartFactory-Anlage: Projektorganisation (geschrieben)	8
4.4	Projekt SmartFactory-Anlage: Projektverlauf (umschreiben)	10
4.5	Projekt SmartFactory-Anlage: Ergebnis (weiter schreiben)	16
4.6	Persönliches Fazit zum Praktikum (umschreiben)	17

1 Einleitung (evtl noch etwas umschreiben)

Dieser Praktikumsbericht dokumentiert mein Praxissemester bei der Siemens AG. Als dualer Student bei Siemens bin ich sowohl meiner Fachabteilung (DI CS SD EH FA 2), als auch der SPE (Siemens Professional Education), also einer Ausbildungsabteilung, zugewiesen. 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 FA 2), war ich Teil der Weiterentwicklung der Smart Factory-Anlage, einer für Studierende konzipierten Trainings- und Automatisierungsanlage zur Simulation einer Flaschenabfüllungsproduktion. Während meiner Zeit in der Ausbildungsabteilung habe ich das Projekt "BersiBot", die Programmierung eines Roboters mit KI-Bilderkennung und Webseitensteuerung, begleitet. 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.

textitAd Hoc Network based Android Messaging Apps [12]

2 Vorstellung der Siemens AG (noch umschreiben)

Siemens wurde 1847 von Werner Siemens und Georg Halske als „Telegraphen Bau-Anstalt von Siemens und Halske“ in Berlin gegründet. In den darauffolgenden Jahren expandierte die Firma in weitere Standorte in Russland und England und erschloss neue Geschäftsfelder, wie z.B. der Elektrifizierung der Eisenbahn. In den 1930er Jahren kamen unter anderen Sparten elektrische Haushaltsgeräte, Wärme und Heizungskonzepte und Medizintechnik hinzu. Heute sind die Bereiche Digital Industries, Smart Infrastructure und Healthineers, die nach Anzahl der Mitarbeiter größten Segmente des Unternehmens. Die Siemens AG mit ihren aktuell 320.000 Mitarbeitern konnte im Geschäftsjahr 2022/23 einen Umsatz von rund 77,8 Millionen Euro erzielen und dabei einen Nettogewinn von 8,5 Millionen Euro erwirtschaften.

3 Vorstellung meiner Fachabteilung (noch umschreiben)

Ich bin in der Abteilung Expert House eingestellt. Diese gehört zum Kundenservice der Digital Industries – Sparte des Siemens AG. Das Expert House dient dazu, Siemens – Kunden fachliche Spezialisten für ihre eigenen Projekte zur Verfügung zu stellen. Innerhalb des Expert House gibt es unterschiedliche Fachbereiche, darunter den Bereich Process Automation, sowie Factory Automation, welchem ich anhöre. Typische Tätigkeitsbereiche der Factory Automation sind zum Beispiel die Programmierung und Inbetriebnahme einer Automatisierungsanlage für den Kunden oder die Entwicklung von kundenspezifischer Software. Der Factory Automation Bereich wird aufgrund es fachlichen Umfangs noch ein weiteres Mal in nummerierte Gruppen unterteilt. Meine Gruppe, 2, beschäftigt sich vor Allem mit der Entwicklung von Cloud Services und Edge Anwendungen. Allerdings werden Mitarbeiter auch über die eigentlichen Aufgabenbereiche ihrer Gruppe hinaus eingesetzt.

4 Projekt Smart Factory-Anlage

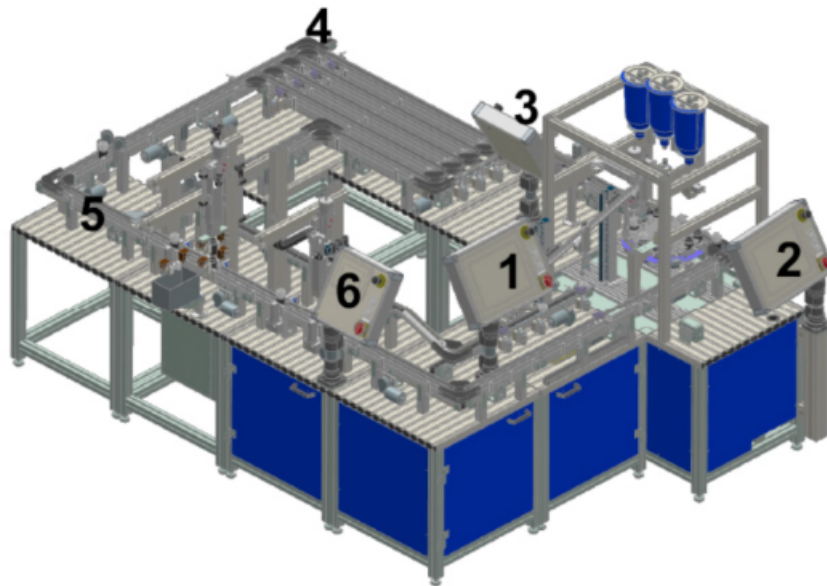


Abbildung 1: Stationen der SmartFactory-Anlage⁸

Abbildung 4.1: Stationen der SmartFactory-Anlage[?]

4.1 Projekt Smart Factory-Anlage: Aufgabenstellung (evtl noch umschreiben)

Die SmartFactory-Anlage ist eine speziell für Schulungszwecke konzipierte Automatisierungsanlage, die von der Abteilung Expert House entwickelt wurde. 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 von Bugs, als auch das Testen der Robustheit der Software. Die

Software wurde nach dem ISA-88-Standard umgesetzt und alle Erweiterungen sollen auch nach diesem hinzugefügt werden. Zur Programmierung war die Siemens-Automatisierungssoftware „TIA-Portal“ zur Verwendung vorgegeben. 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. (evtl auf aufteilung unit,em,cm eingehen?)

4.2 Projekt Smart Factory-Anlage: Zielsetzung

Das Ziel am Ende des Praxissemesters war es, die funktionsfähige SmartFactory-Anlage Weiterzuentwickeln, ihre Robustheit zu erhöhen und dies zu präsentieren. Dafür muss für jedes Anlagenmodul bekannte Fehler und offene Punkte aus der LOP abgearbeitet werden, sowie bei der Erkennung neuer Fehler, diese zur LOP hinzugefügt werden. Desweiteren müssen die Bedingungen des Projektowners, eine einheitliche HMI-Schnittstelle und eine leichte Bedienung der Anlage umgesetzt werden.

(Maybe auf zyklische Kommunikation und Bugfixes eingehen? Teil der Zielsetzung?: Bei der SmartFactory-Anlage handelt es sich um eine Übungs- und Testanlage zu Ausbildungszwecken und dem Testen von Prototypen. Sie simuliert eine Flaschenabfüllungsproduktion in einem kleinen Format und soll Auszubildenden mit vielen Produkten aus dem Siemens-Katalog vertraut machen. Die Anlage ist daher in sechs Anlagenmodule unterteilt, von der Materialzuführung bis hin zum abschließenden Recycling, jedes mit verschiedenen Schwerpunkten aber ähnlicher Hardware. Hardwaretechnisch sind alle Anlagen mit einer Siemens- Simatic S7-1500 Speicherprogrammierbaren Steuerung, sowie einem Siemens- HMI, einem Touchscreen für Statusmeldungen und Benutzereingaben und Transportbändern ausgestattet. Desweiteren sind an einigen Stationen pneumatisch betreibende Greifern und RFID-Schreib-Lesegeräte verbaut. Die Anlage läuft iterativ. Im Falle der Anlage bedeutet dies, dass am Ende des Kreislaufes, nach dem Recycling, die Flaschen, Deckel und Kugeln wieder für die erneute Verarbeitung bereitgestellt werden, damit sie erneut abgefüllt werden können. Die Anlage kann über drei Betriebsmodi, Einrichtungs-, Hand- und Automatikbetrieb und zwei Betriebsarten, Modular- und Inselbetrieb über das stationszugehörigen HMI gesteuert werden.

Im Einrichtungsbetrieb kann der Bediener uneingeschränkt alle Anlagenteile bewegen, auch wenn dies zu mechanischen Fehlern führen könnte. Der Handbetrieb ermöglicht es, einzelne Funktionen, die sich aus der Ansteuerung mehrerer Aktoren zusammensetzen auszuführen, beispielsweise das Befüllen einer Flasche oder das Entleeren eines Bandes. Im Automatikbetrieb soll die Anlage vollständig automatisiert laufen und nur im Fehlerfall oder zum Anlegen eines Auftrags einen Bediener erfordern. Im Folgenden wird der Ablauf der verschiedenen Stationen der Anlage im Automatikbetrieb erläutert.)

noch teil der Zielsetzung?

Maybe als kurzbeschreibung der Anlage?

4.2.1 Station 1: Modulares Produktionssystem (umschreiben)

Das Modulare Produktionssystem nimmt Flaschen und Deckel von der Recycling-Anlage entgegen und stellt diese auf Anforderung der Abfüllstation zur Verfügung. Sie leitet nur weiße Deckel weiter, während deckel mit einer anderen farbe aussortiert werden. Deckel mit unlesbaren RFID-Tag sollen grundsätzlich aussortiert werden. Sollte eine Anforderung der Abfüllstation erfolgen, wird die angegebene Anzahl der Flaschen gesendet und bei Anforderung ein Deckel auf die befüllte Flasche gesetzt.

4.2.2 Station 2: Abfüllung (geschrieben)

Die Abfüllstation hat die Aufgabe, Flaschen mit einer vorgegebenen Anzahl von Kugeln zu befüllen. Diese Anzahl wird durch die Erstellung von Aufträgen am HMI 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 Ön-DemandTCP-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 des Auftrages

angegebenen Anzahl an Kugeln befüllt. Dann wird ein Deckel von dem modularen Produktionssystem angefordert. Der Deckel wird danach mit einem drehbaren Greifer verschraubt und an der nächsten Station wird der Deckel mit einem RFID-Tag beschrieben. Der RFID-Tag enthält Datum und Uhrzeit der Abfüllung, sowie welcher Auftrag abgefüllt wurde.

4.2.3 Station 3: Quality Gate (geschrieben)

Das Quality Gate soll wie der Name sagt, die Auftragsdaten auf dem RFID-Tag mit dem Inhalt der Flasche abgleichen. Hierzu soll ein Farbsensor dienen, der die farbliche Zusammensetzung prüft. Sollte die farbliche Zusammensetzung übereinstimmen, wird der RFID-Tag der Flasche mit "Gut" beschrieben, sollte sie aber nicht übereinstimmen wird der RFID-Tag mit "„Ausschuss" beschrieben.

4.2.4 Station 4: Kommissionierbahnen (geschrieben)

Die Kommissionierbahnen sollen die vom Quality-Gate ankommenden Flaschen nach Auftragsnummer sortieren, hierzu wird zu Beginn der RFID-Tag gelesen und die Flasche auf eine von vier Bahnen sortiert. Fehlerhafte Flaschen werden direkt über ein Ausschussband an die nächste Station weitergegeben. Sollte die Flasche "In Ordnung" sein, wird sie auf eine von vier Bändern sortiert. (ein Auftrag pro Band). Die sortierten Flaschen sollen so lange in der Station gehalten werden, bis der entsprechende Auftrag abgeschlossen wurde oder der Bediener ein manuelles Entleeren der Bänder anfordert.

4.2.5 Station 5: Eckstation (geschrieben)

Das Feature der Eckstation ist, zwei Laufbänder, die wie hochklappbare Brücken fungieren. Das Hochklappen ermöglicht es, einen Schaltschrank in der Mitte der Anlage zu erreichen. Das Hochklappen der Laufbänder soll unterbunden werden, 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.

4.2.6 Station 6: Recycling (geschrieben)

Die Recyclingstation recycelt alle ankommenden Flaschen, also alle aus erfolgreich erfüllten Aufträgen und alle aussortierten Flaschen. Der Flaschendeckel wird mit einem Greifer entfernt und auf ein anderes Laufband gesetzt. Danach wird die Flasche angehoben und die Kugeln über einem Trichter entfernt. Die Kugeln werden daraufhin mit einem Farbsortierer und Luftdruck in die farblich übereinstimmenden Container an der Abfüllung geschossen, und die Flasche wird auf das Laufband zum modularen Produktionssystem gesetzt.

4.2.7 Netzwerkverbund (geschrieben)

Alle Anlagenteile sind mit Ethernet mit dem Einschaltschrank 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 erfolgt "On-Demand" und passiert dadurch nur wenn sie von einer anderen Station angefordert wird. Dies minimiert unnötigen datenverkehr und trägt dadurch zur Reduzierung der verbrauchten Ressourcen bei.

4.3 Projekt SmartFactory-Anlage: Projektorganisation (geschrieben)

Das Projektteam besteht aus ... Informatikstudenten und ... Elektro- und Informationstechnikstudenten sowie drei Fachbetreuern. Diese haben uns als "Product Owner" im Projekt unterstützt und geleitet.

Das gesamte Projekt wurde auf Scrum basierend durchgeführt. Allerdings wurde, wie in Abschnitt 1, "Einleitung", erläutert, sowohl bei uns, den Informatikern, als auch bei den Elektro- und Informationstechnikern, die Phase im Expert House durch die SPE-Phase unterbrochen, dies allerdings zu unterschiedlichen Zeitpunkten. Dadurch wurde in den ersten acht Wochen von allen gemeinsam am Projekt gearbeitet, in den folgenden acht Wochen nur von den Elektro- und Informationstechnikern, und in den letzten acht Wochen ausschließlich von uns Informatikern.

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

Dies passte gut, da wir unser Team mit Scrum organisiert haben. Dadurch stimmten wir uns gemeinsam, Betreuer und Auszubildende, in Daily-Meetings ab. Wir arbeiteten in Zwei-Wochen-Sprints, die jeweils mit einer Vorstellung der Ergebnisse an die Stakeholder endeten. Eine zentrale Rolle spielte die "List of Open Points"(LOP), die zur Dokumentation von bestehenden Fehlern und Mängeln genutzt wurde.

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.

Obwohl wir uns nicht strikt an Scrum hielten, ergab dieser Ansatz aus unserer Sicht Sinn. Der Überblick über die aktuellen Aufgaben war so deutlich besser, und die Verständlichkeit in den Meetings wurde durch die Visualisierung wesentlich erleichtert.

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 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 speziell 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, die Zusammenarbeit erleichtert und ermöglicht eine grundlegende Versionskontrolle.

4.4 Projekt SmartFactory-Anlage: Projektverlauf (umschreiben)

"Begriffsklärung „TIA Portal“:

Im nachfolgenden Abschnitt wird das TIA-Portal mehrfach thematisiert. Dabei handelt es sich um die Entwicklungsumgebung für alle Automatisierungsprodukte von Siemens [?].

Die Prozesslogik für SPS (Speicherprogrammierbare Steuerungen) 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 PLC (Programmable Logic Controller) als Echtzeitsystem zyklisch arbeitet, sie führt Berechnungen durch und aktualisiert Ausgaben in festen Intervallen, sind bei der Programmierung spezielle Anforderungen zu beachten. Um diesen Anforderungen gerecht zu werden, 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.

Auftaktwoche:

Noch vor dem offiziellen Start des Projekts wurde in meiner Abteilung eine Auftaktwoche organisiert, die dazu diente, uns auf die bevorstehenden Aufgaben vorzubereiten. 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 Entwicklung der SmartFactory-Anlage spielten.

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 des ersten Laufbandes 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.

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` soll so entwickelt werden, dass es universell für alle Sensortypen einsetzbar ist, wobei lediglich die Hardwareadresse des jeweiligen Sensors angegeben werden muss.
- **Equipment Modul:** Equipment Module bestehen aus einer Kombination mehrerer Control Module 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.

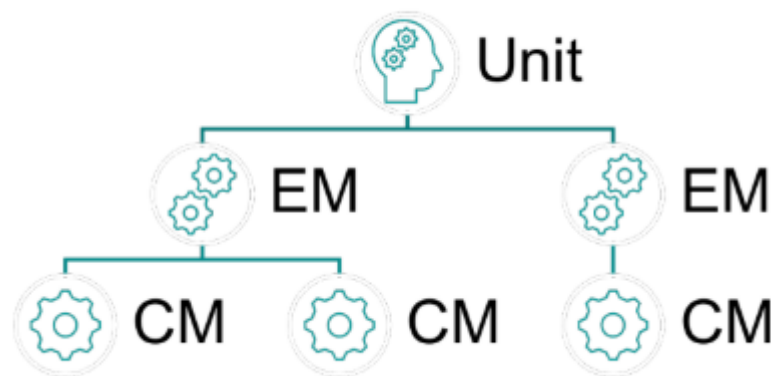


Abbildung 4.2: ISA-88 Modulhierarchie[?]

Bibliotheksverantwortlicher (erweiternder Aufgabenbereich):

Mir wurde ein erweiterter Aufgabenbereich in Form des Bibliotheksverantwortlichen zugewiesen. Meine 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 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.

Flaschenmanagement in der unit: wie es war -> wie es werden soll (z.b. auftrag abarbeiten wenn keine weitere flasche kommt oder ersten drei flaschen für neuen auftrag verwendet) leertfahren in der unit (bei unbekannten Flaschen) current order abbrechen

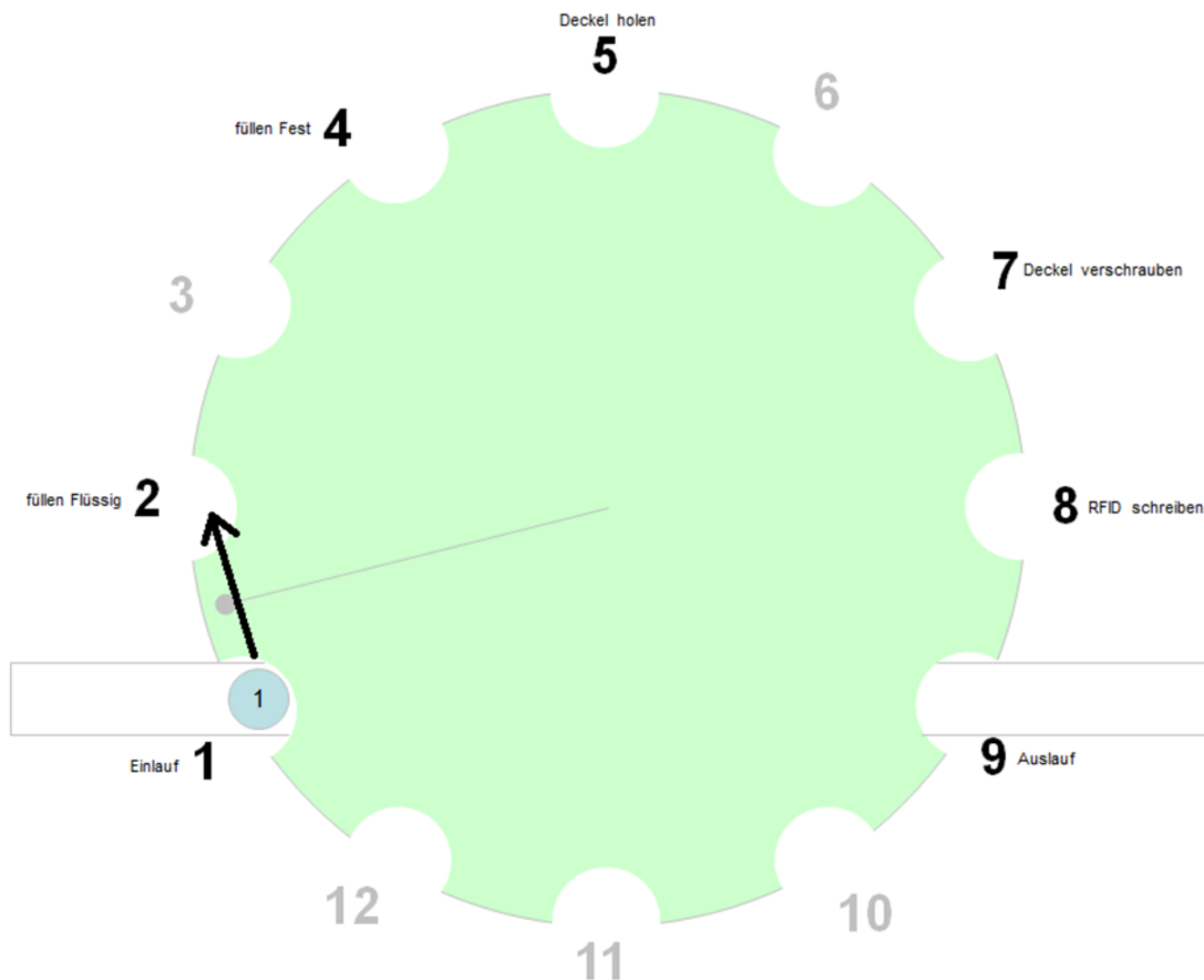


Abbildung 4.3: Drehteller der Abfüllanlage[?]

Das Flaschenmanagement der Abfüllstation wurde so umgesetzt, dass für jeden angeforderten Auftrag die benötigte Anzahl an Flaschen vom modularen Produk-

tionssystem bereitgestellt wird. Diese Flaschen werden anschließend befüllt und mit Deckeln verschlossen. Ein Problem tritt jedoch auf, wenn nicht genügend Flaschen nachgeliefert werden. Beispielsweise können von 12 angeforderten Flaschen nur 9 ankommen, wodurch die Abfüllung anhält und auf die Ankunft einer neuen Flasche an Position 1 des Drehtellers wartet (Verweis auf Abbildung?).

Um den Arbeitsfluss kontinuierlich aufrechtzuerhalten, habe ich in Absprache mit den Betreuern eine Anpassung in der Unit der Abfüllstation vorgenommen. Diese Änderung sorgt dafür, dass, falls innerhalb von sieben Sekunden keine neue Flasche geliefert wird, die verbliebenen Flaschen, die sich bereits im Drehteller befinden und einem Auftrag zugewiesen wurden, dennoch abgefüllt und verschlossen werden. Dies gewährleistet eine höhere Effizienz und verhindert lange Stillstandszeiten der Abfüllstation.

Ein weiteres Problem betraf die Speicherung der Auftragsdaten. Diese wurden ursprünglich nicht remanent gespeichert, also nicht über einen PLC-Stop hinaus erhalten. Dies führte dazu, dass nach jedem Neustart oder Kaltstart der Steuerung alle Flaschen aus dem Drehteller entfernt werden mussten, bevor ein neuer Auftrag angelegt werden konnte. Dieses Verhalten widersprach dem Ziel, die Anlage auch für unerfahrene Bediener einfach und benutzerfreundlich zu gestalten.

Der Drehteller erkannte nach einem Neustart weiterhin, dass sich Flaschen an den einzelnen Positionen (1–9) befanden. Dies führte dazu, dass der Drehteller in einen ungültigen Zustand geriet, da er für jede Position, an der eine Aufgabe auszuführen war, fehlerhafte Zustandsinformationen erhielt. Beispielsweise müssen an Position 4 die Flaschen befüllt, an Position 5 die Deckel angefordert, an Position 7 die Deckel verschraubt und an Position 8 die RFID-Tags beschrieben werden. Ohne korrekte Zustandsinformationen konnte der Drehteller diese Aufgaben nicht ausführen, was die gesamte Anlage blockierte. Nun nach meiner Änderung, wird der Drehteller darauf hingewiesen, dass sich unbekannte Flaschen, also Flaschen ohne Auftrag, im Drehteller befinden. Diese Flaschen werden jetzt automatisch aus dem Drehteller entfernt und an das Quality Gate weitergeleitet.

Drehteller: fehlerfindung der knöpe -> Initialisierungsfahrt

Trusted Tracebility: rfid überschreibung abgepasst + anzeige auf dem bildschirm

Kommunikation: zyklisch und unitlogik zwischen mps und Abfüllung zwischen abfüllung und quality gate

4.5 Projekt SmartFactory-Anlage: Ergebnis (weiter schreiben)

Das finale Ergebnis des Projekts ist zum Zeitpunkt der Erstellung dieses Berichts noch nicht vollständig absehbar. Dennoch wurden bereits wesentliche Meilensteine gemäß den Vorgaben erreicht. Mit Ausnahme einiger kleinerer Bugs konnten alle kritischen, den Prozess unterbrechenden Fehler an der Abfüllstation behoben werden. Dadurch wurde das Hauptziel, die Robustheit der Anlage zu steigern, erfolgreich erreicht. Die weiteren Arbeiten konzentrieren sich nun auf die Verbesserung der Benutzerfreundlichkeit und der Erweiterbarkeit der Anlage.

Aus den täglichen Meetings und dem Feedback der Projektmitglieder geht hervor, dass auch an den anderen Stationen der Anlage deutliche Fortschritte erzielt wurden. Falls der Projektfortschritt bis zum Ende des Praxissemesters nicht vollständig den gesetzten Zielen entspricht, bieten die sorgfältig erstellten Konzeptdokumente eine solide Grundlage für eine erfolgreiche Weiterführung durch zukünftige Studierende.

Durch die vorgenommenen Änderungen wurde nicht nur die Effizienz der Abfüllstation gesteigert, sondern auch die Benutzerfreundlichkeit und Robustheit des Systems deutlich verbessert.

4.6 Persönliches Fazit zum Praktikum (umschreiben)

Rückblickend möchte ich noch auf meine persönlichen Eindrücke des Praxissemesters bei der Siemens AG eingehen. Da ich dualer Student mit Berufserfahrung in einer anderen Firma bin, war für mich die Arbeit in einem großen Industrieunternehmen keine gänzlich neue Erfahrung. Allerdings unterscheidet sich die Arbeitsweise in meiner Fachabteilung, dem Expert House deutlich davon, wie ich es bereits kannte. Besonders hervorheben möchte ich die täglichen Meetings, durch welche es jederzeit möglich war, einen ganzheitlichen Überblick über den Projektfortschritt zu haben. Ich gelernt, dass verschiedene Projektmanagement-Modelle, wie z.B. das Wasserfallmodell oder Kanban, nicht zwingend getrennt voneinander stattfinden können, sondern das sinnvolle Methoden mehrerer dieser Modelle auch kombiniert eingesetzt werden können. Auch Konzepte wie „Lessions 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 aus meiner Sicht etwas der Blick auf den Projektfortschritt der SmartFactory-Anlage verloren. Neben den organisatorischen Aspekten empfand ich die Programmierung mit dem TIA-Portal besonders interessant, da es sich dabei um eine völlig andere Art der Programmierung im Vergleich zu den mir bekannten Programmiersprachen handelt und viele Besonderheiten zu beachten sind, wie zum Beispiel die Berücksichtigung des zyklischen Programmablaufs und dem Bausteinkonzept. Mein persönliches Highlight war allerdings das Arbeiten an einer Industrieanlage. Mir wurde klar, dass es ein besonderes Erfolgserlebnis ist, wenn ein gut ausgearbeitetes Konzept, zu einem funktionsfähigen Code führt und dadurch ein reibungsloser Prozessablauf an einer mit unzähligen Sensoren und Aktoren ausgestatten Anlage möglich wird. Ist man neben der reinen Informatik auch an Mechanik und Elektrotechnik interessiert wird, ist ein Praxissemester im Expert House bei der Siemens AG sehr empfehlenswert.