Bestandsaufnahme der Anlagensteuerung der Fertigungsstraße im PTZ

Studienarbeit T3100

des Studienganges Elektrotechnik - Automation

an der Dualen Hochschule Baden-Württemberg Ravensburg Campus Friedrichshafen

von

David Weberbauer, Robin Wörner, Aaron Gräf

13.01.2025

Bearbeitungszeitraum	09.10.2024 - 13.01.2025
Matrikelnummer	8113187, 3062326, 8254936
Kurs	TEA22
Gutachter der Dualen Hochschule	Edvin Hamzic

Sperrvermerk

Gemäß Ziffer 1.1.13 der Anlage 1 zu §§ 3,4 und 5 der Studien- und Prüfungsordnung für die Bachelorstudiengänge im Studienbereich Technik der Dualen Hochschule Baden-Württemberg vom 29.09.2017 in der Fassung vom 25.07.2018:

„Der Inhalt dieser Arbeit darf weder als Ganzes noch in Auszügen Personen außerhalb des Prüfungsprozesses und des Evaluationsverfahrens zugänglich gemacht werden, sofern keine anders lautende Genehmigung vom Dualen Partner vorliegt."

Friedrichshafen, den 13.01.2025

Erklärung

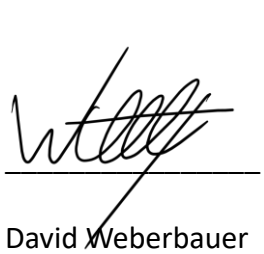
Gemäß Ziffer 1.1.13 der Anlage 1 zu §§ 3,4 und 5 der Studien- und Prüfungsordnung für die Bachelorstudiengänge im Studienbereich Technik der Dualen Hochschule Baden-Württemberg vom 29.09.2017 in der Fassung vom 25.07.2018:

Ich versichere hiermit, dass ich meine Projektarbeit mit dem Thema:

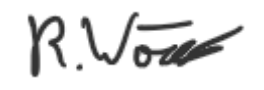
Bestandsaufnahme der Anlagensteuerung der Fertigungsstraße im PTZ

selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Friedrichshafen, 13.01.2025
Ort, Datum



David Weberbauer



Robin Wörner



Aaron Gräf

Zusammenfassung

Diese Studienarbeit befasst sich mit der Optimierung einer Produktionslinie im Produktionstechnischen Zentrum (PTZ) durch den Einsatz von speicherprogrammierbaren Steuerungen (SPS), Robotik und einem Manufacturing Execution System (MES). Ziel des Projekts war es, bestehende Schwächen in der Netzwerkarchitektur, der Programmierung und der Schnittstellenintegration zu analysieren und gezielte Lösungen zu entwickeln.

Im Fokus stand die Verbesserung der Kommunikation zwischen verschiedenen Systemkomponenten, einschließlich Sensoren, Robotern und Steuerungseinheiten, um eine effiziente und flexible Fertigung zu gewährleisten. Die Maßnahmen umfassten die Neustrukturierung des SPS-Programmcodes, die Integration von Bildverarbeitungsdaten sowie die Entwicklung einer klaren und erweiterbaren Architektur. Besondere Herausforderungen ergaben sich aus der mangelnden Dokumentation, der wechselnden Teamzusammensetzung und der Komplexität der vorhandenen Systeme.

Die Einbindung moderner Technologien zur Datenerfassung und -analyse ermöglichte jedoch erste Erfolge, darunter die Stabilisierung der Systemleistung und die Automatisierung bislang manuell durchgeführter Prozesse. Die Ergebnisse der Arbeit zeigen, dass datenbasierte Steuerungstechniken nicht nur die Effizienz, sondern auch die Skalierbarkeit und Flexibilität moderner Produktionslinien erheblich steigern können. Dies legt eine solide Grundlage für zukünftige Entwicklungen im Rahmen von Industrie-4.0-Initiativen.

Abstract

This study explores the optimization of a production line at the Production Technology Center (PTZ) using programmable logic controllers (PLC), robotics, and a Manufacturing Execution System (MES). The project aimed to analyze and address existing weaknesses in network architecture, programming, and interface integration. A key focus was improving communication among various system components, including sensors, robots, and control units, to ensure efficient and flexible manufacturing.

Measures included restructuring PLC code, integrating image processing data, and developing a clear, expandable system architecture. Challenges arose due to insufficient documentation, frequent team changes, and the complexity of existing systems.

The integration of modern data collection and analysis technologies achieved initial successes, such as stabilizing system performance and automating previous manual processes. The findings demonstrate that data-driven control techniques can significantly enhance the efficiency, scalability, and flexibility of modern production lines. This establishes a solid foundation for future developments within Industry 4.0 initiatives.

Inhaltsverzeichnis

1. Einleitung 1

2. Die Zielsetzung 3

3. Bearbeitung offener Punkte 2024 5

3.1 Vernetzung der SPS 5

3.2 Integration verschiedener Komponenten im Schaltschrank 7

4. Weiterentwicklung SPS-Architektur - Grundlagen 10

4.1 Der Organisationsbaustein 11

4.2 Der Datenbaustein 12

4.3 Die Funktion 12

4.4 Der Funktionsbaustein 13

4.5 Die Hierarchie 13

4.6 Unterschiedliche Programmiersprachen 15

5. Weiterentwicklung SPS-Architektur – Ergebnisse 17

5.1 Ausgangszustand 17

5.2 Die Variablen 20

5.3 Die Roboter 21

5.4 Das Human Maschine Interface (HMI) 22

5.5 Die Problematik 23

5.6 Konzept: “end of line” 25

5.7 Ausstehende Aufgaben 27

5.8 Bisherige Programmierergebnisse 28

6. Schnittstellen 29

6.1 Eingebundene Schnittstellen: 30

6.1.1 AL1100 30

6.1.2 AL1102 32

6.1.3 Roboter „Station 1“ 33

6.1.4 Roboter „Station 2“ 37
6.1.5 VSE150 39
6.1.6 Switch 40
6.1.7 wenglor Kamera 41
6.1.8 Lichtschranken/Endlagenschalter 42
6.1.9 Ventilinsel 42
6.1.10 QR-Scanner 42
6.2 Zukünftige Schnittstellen: 42
6.2.1 ifm Kamera 42
6.2.2 Roboter „Station 5“ 43
6.2.3 MES 43
7. Fazit 45
7. Literaturverzeichnis 46
9. Abbildungsverzeichnis 47

1. Einleitung

Die vorliegende Studienarbeit beschäftigt sich mit der Optimierung einer Produktionslinie im PTZ (Produktionstechnisches Zentrum), die mithilfe mehrerer Roboterarme, verschiedenen Sensoren und Aktoren, sowie einer SPS (Speicherprogrammierbaren Steuerung) realisiert wird. Ziel dieser Arbeit ist es, die Automatisierung von Fertigungsprozessen durch den kombinierten Einsatz von Robotik und SPS-Technologie zu untersuchen und zu verbessern. Die SPS dient hierbei als zentrale Steuereinheit, um die Synchronisation und Ansteuerung der Roboterarme sowie weiterer Maschinen (z.B. Umleger) entlang der Produktionslinie zu gewährleisten. In Abbildung 1 („MES und Data Connection“) wird die Rolle der SPS als integraler Bestandteil der Maschinensteuerung im Kontext einer vernetzten Produktionsumgebung visualisiert.

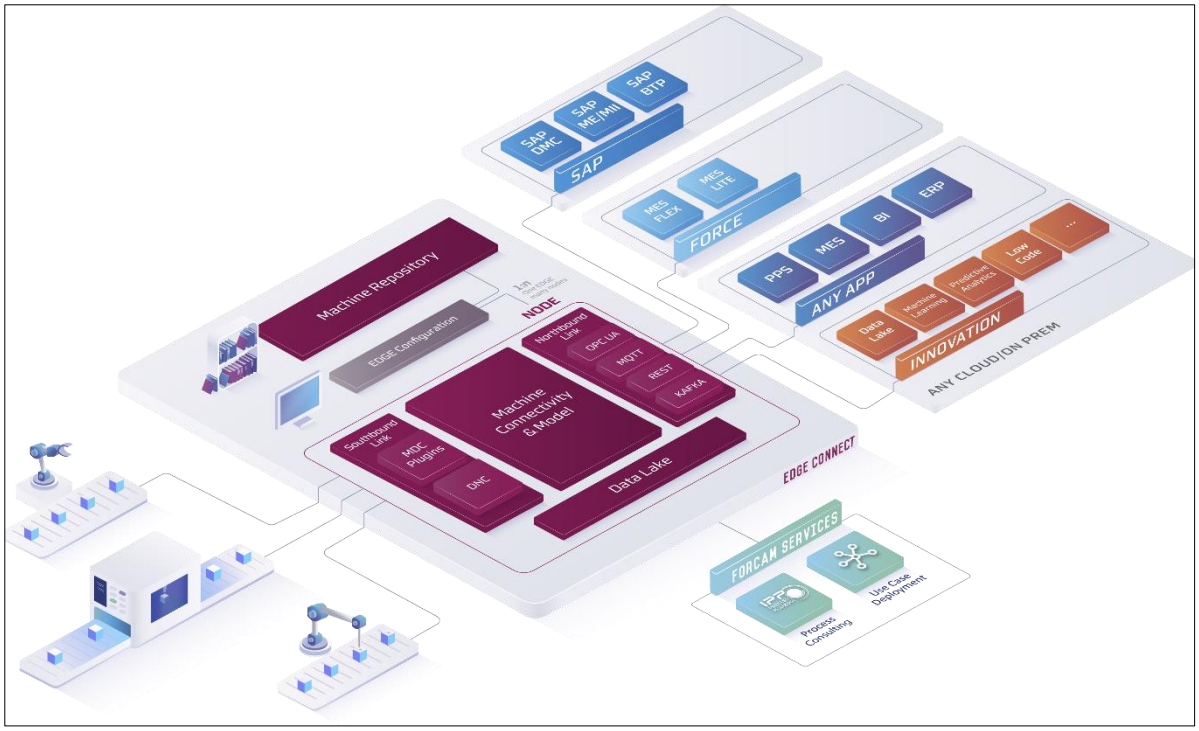


Abbildung 1:SPS, MES und Data Connection [1]

Im Kontext von Industrie 4.0 und der fortschreitenden Digitalisierung sind datenbasierte, vernetzte und sich selbst steuernde Wertschöpfungsketten von entscheidender Bedeutung. Diese Konzepte zielen darauf ab, Produktionsprozesse effizienter, flexibler und anpassungsfähiger zu gestalten. Abbildung 1 [1] zeigt, wie verschiedene Systeme wie ERP, MES und andere Plattformen miteinander verbunden sind, um einen kontinuierlichen Datenfluss und die Automatisierung in der Produktion zu ermöglichen. Durch die Implementierung einer SPS-gesteuerten Produktionslinie wird eine Grundlage geschaffen, um diese Prinzipien praktisch zu erproben und weiterzuentwickeln.

Ein weiterer Aspekt dieser Arbeit ist die Weiterentwicklung eines Manufacturing Execution Systems (MES), das als Schnittstelle zwischen der Produktionssteuerung und der Managementebene fungiert. In Abbildung 1 wird dies durch die Verbindung zwischen der Maschinensteuerung und den höheren IT-Systemen deutlich. Das MES ermöglicht die Überwachung und Analyse der Produktionsprozesse in Echtzeit, was zu einer verbesserten Transparenz und Entscheidungsfindung führt. Quelle? Die Data Connection, also die Vernetzung von SPS, Robotern und MES, bildet die Grundlage für die Erhebung und Auswertung von Mess- und Prozessdaten. Dies schafft die Voraussetzungen für datengetriebene Optimierungen und fördert die Integration von Automatisierungslösungen in ein modernes, vernetztes Fertigungsumfeld. Quelle?

Die Studienarbeit legt besonderen Fokus auf die Verknüpfung von Sensorik, Robotik und Datenverarbeitung, um präzise und automatisierte Fertigungsabläufe zu ermöglichen. Die Analyse umfasst sowohl die technische Implementierung als auch die potenziellen Verbesserungen im Hinblick auf Effizienz und Prozesssicherheit. In Abbildung 1 wird die Rolle der Datenablage durch Elemente wie „Machine Repository“ und „Data Lake“ hervorgehoben, die zur Speicherung und Verarbeitung von Produktionsdaten dienen und so eine umfassende Datenanalyse unterstützen. Die Darstellung der Edge-Connect-Technologie in der Abbildung zeigt, wie die lokale Datenverarbeitung in der Nähe der Produktionseinheiten erfolgt, um eine schnelle Reaktionszeit und effiziente Datennutzung zu gewährleisten.

2. Die Zielsetzung

Ziel der Projektarbeit T3100 ist die systematische Abarbeitung und Optimierung offener Punkte in der Programmierung der SPS S7-1500 sowie eine grundlegende Verbesserung der aktuellen, unstrukturierten Programmarchitektur. Der vorhandene Code ist unkommentiert und schwer nachvollziehbar, wodurch Wartung und Weiterentwicklung erschwert werden. Diese strukturellen Schwächen sollen im Rahmen des Projekts durch gezielte Anpassungen und Optimierungen behoben werden, um eine stabilere und wartungsfreundlichere Basis für den weiteren Betrieb und zukünftige Erweiterungen zu schaffen.

Ein weiterer zentraler Aspekt des Projekts ist die Integration der Bildverarbeitungsdaten zweier Kamerasysteme – einer wenglor- und einer ifm-Kamera –, die aus parallelen Projekten stammen. Damit die Kameradaten im Produktionsprozess nutzbar werden, wird ein neues Programmmodul entwickelt, das die beiden Systeme effizient einbindet und eine fehlerfreie Datenübertragung und -verarbeitung sicherstellt. So wird gewährleistet, dass die Kameradaten den Produktionsprozess sinnvoll unterstützen und die Effizienz der Abläufe steigern.

Durch die Neustrukturierung und Verbesserung des gesamten Programmcodes soll eine robuste, gut dokumentierte Architektur geschaffen werden, die künftige Anpassungen und Erweiterungen erleichtert und die Systemstabilität im Betrieb deutlich erhöht. Eine enge Zusammenarbeit und klare Aufgabenverteilung im Team stellt sicher, dass alle Projektergebnisse sorgfältig dokumentiert und für nachfolgende Teams leicht nachvollziehbar sind.

Die nachfolgende Abbildung Verweis? zeigt den Ablaufplan für das Projekt „T3100“, der sich von September 2024 bis Januar 2025 erstreckt. Der Plan beginnt mit der Einarbeitung in das SPS-Programm, die Ende September startet und bis zum Meilenstein „Programmbausteine sowie Datenbausteine für Kameras erstellt“ läuft. Anfang Oktober findet der Projekt-Kickoff statt, gefolgt von einer kurzen Phase zur Bearbeitung offener Punkte für das Jahr 2024. Im Anschluss beginnt die Weiterentwicklung der SPS-Architektur, die fast bis zum Ende des Jahres andauert.

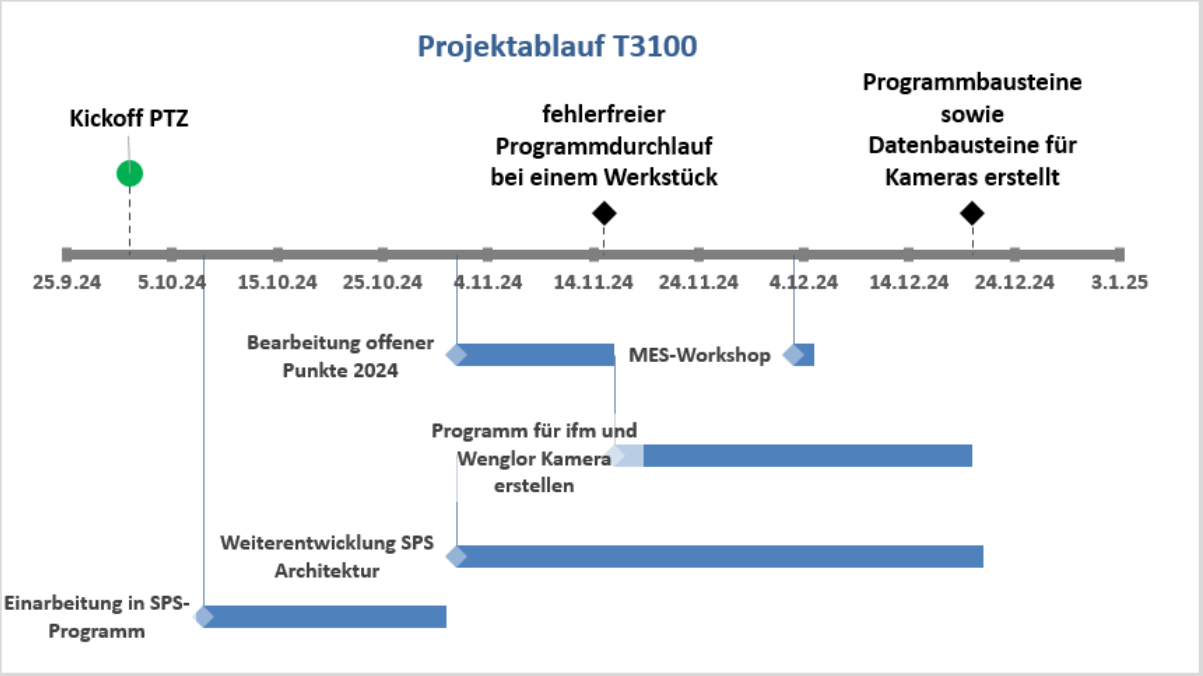


Abbildung 2: Projektablauf T3100

Parallel dazu wird im November ein Programm für die ifm- und wenglor-Kameras erstellt, das bis Anfang Dezember abgeschlossen sein soll. Am 14. November ist der erste wichtige Meilenstein erreicht: ein fehlerfreier Programmdurchlauf bei einem Werkstück.

Anfang Dezember wird zusätzlich ein MES-Workshop durchgeführt, in dem das MES-System von Forcam vorgestellt und dessen Integration in die Produktionsabläufe erarbeitet wird.

Zum Jahresende, am 20. Dezember, steht die Fertigstellung der Programm- und Datenbausteine für die Kameras an, womit ein weiterer wichtiger Meilenstein erreicht wird. Der Plan verdeutlicht, wie die einzelnen Aufgaben zeitlich aufeinander abgestimmt sind und wann wichtige Fortschritte im Projekt erzielt werden sollen.

3. Bearbeitung offener Punkte 2024

3.1 Vernetzung der SPS

Eine zuverlässige Kommunikation zwischen den Geräten ist in der Automatisierungstechnik unerlässlich, um eine effiziente und störungsfreie Produktion sicherzustellen. Dieses Kapitel untersucht die Netzwerk-Topologie der Produktionslinie und beleuchtet die bestehenden Mängel, die zu Projektbeginn vorhanden waren. Zu diesen Mängeln gehört zum einen die IP-Adressvergabe und die Port-Zuordnung an Switches, wie sie in der Abbildung zu sehen sind.

Auf welche Abbildung wird referenziert?

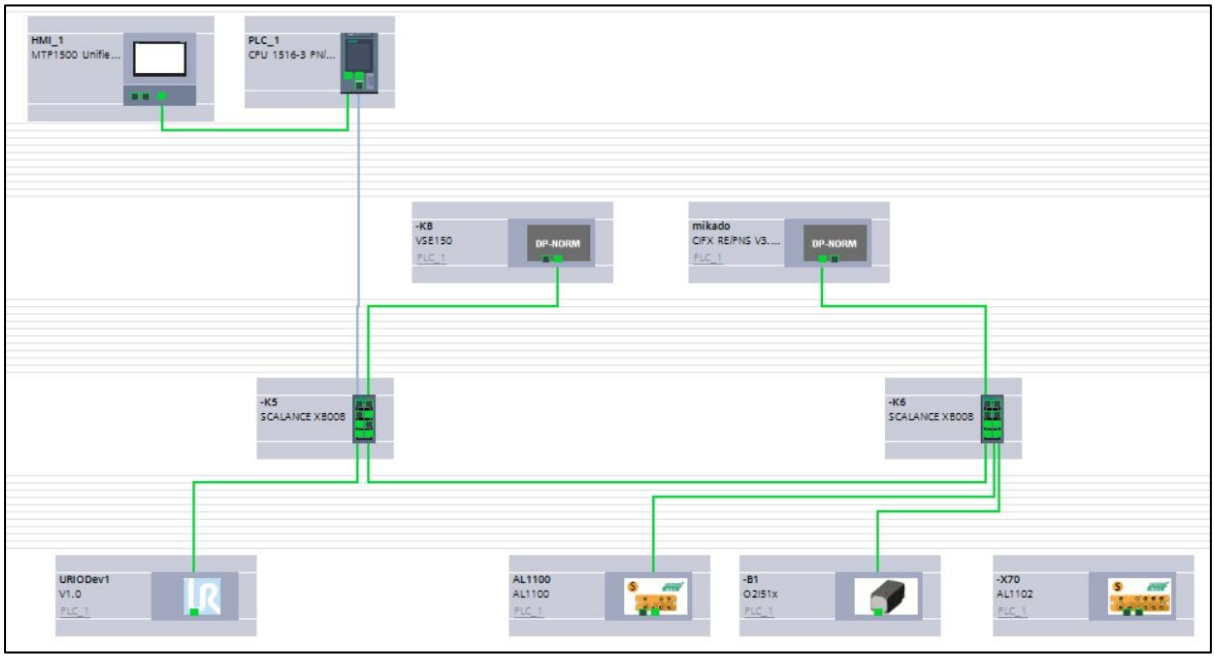


Abbildung 3: Topologieansicht SPS

Verweis auf welche Abbildung

In der dargestellten Topologie (siehe Abbildung) sind verschiedene Komponenten über ein industrielles Netzwerk verbunden. Dazu gehören ein HMI-Bediengerät, eine CPU (SPS), Sensoren, Aktoren sowie Switches, die als zentrale Verbindungspunkte fungieren. Die Kommunikation erfolgt über das PROFINET-Protokoll, das für industrielle Anwendungen entwickelt wurde und schnelle Datenübertragungen ermöglicht. Eine der anfänglichen Herausforderungen bestand darin, dass eine IP-Adresse versehentlich mehrfach vergeben wurde.

Quelle?

Dies führte zu Kommunikationsproblemen, da in einem Netzwerk jede Komponente eine eindeutige IP-Adresse benötigt. Hierbei wurde dem HMI sowie dem Switch „AL1102“ die IP-Adresse 192.168.0.2 vergeben. Mehrfachvergebene IP-Adressen verursachen Konflikte, die die zuverlässige Datenübertragung beeinträchtigen. Eine systematische Vergabe und eine feste Adressierungstabelle könnten hier helfen, um solche Fehler künftig zu vermeiden. [2]

Neben der IP-Adressvergabe ist auch die korrekte Port-Zuordnung an den Switches, wie in der Abbildung dargestellt, entscheidend. Die Switches im Netzwerk dienen als Knotenpunkte, an denen verschiedene Geräte angeschlossen sind. Bei falscher Portbelegung oder fehlerhafter Konfiguration kann die Verbindung zu bestimmten Geräten unterbrochen werden. Eine klare Dokumentation der Portzuordnung sowie die Zuordnung der Geräte zu den entsprechenden Ports sind daher für eine reibungslose Kommunikation unerlässlich.

ja, ist wichtig. Wurde es von euch umgesetzt?

Das Problem mit dem HMI bestand darin, dass die Programmierung nur über einen separaten ProfiNet-Port erfolgen konnte, was dazu führte, dass das Profinet-Kabel für den Anlagenbetrieb regelmäßig zwischen den Ports P1 und P2 umgesteckt werden musste. Dieses Vorgehen war nicht nur umständlich, sondern störte auch den Arbeitsablauf und erhöhte das Risiko von Verbindungsproblemen.

Die Lösung bestand darin, die IP-Adresse des HMIs anzupassen und die Bus-Topologie im TIA Portal so zu konfigurieren, dass das HMI korrekt integriert wurde. Dabei wurde das HMI in denselben Bus wie die übrigen Peripheriegeräte eingebunden. Dies eliminierte die Notwendigkeit des ständigen Umsteckens und gewährleistete eine stabile, durchgängige Verbindung. Dadurch wurde nicht nur der Arbeitsablauf erheblich vereinfacht, sondern auch die Kommunikation zwischen den Geräten nachhaltig verbessert.

Die Analyse zeigt, dass eine sorgfältige Planung und Konfiguration der Netzwerk-Topologie in der Produktionsumgebung notwendig sind. IP-Adresskonflikte, fehlerhafte Portzuweisungen und eine fehlerhafte Bus-Integration können zu erheblichen Störungen führen, die die Produktionsqualität beeinträchtigen. Durch eine strukturierte Adressvergabe, eine exakte Portzuordnung und die korrekte Integration aller Geräte in die Netzwerk-Topologie kann die Netzwerkstabilität gewährleistet und eine sichere Kommunikationsbasis geschaffen werden.

3.2 Integration verschiedener Komponenten im Schaltschrank

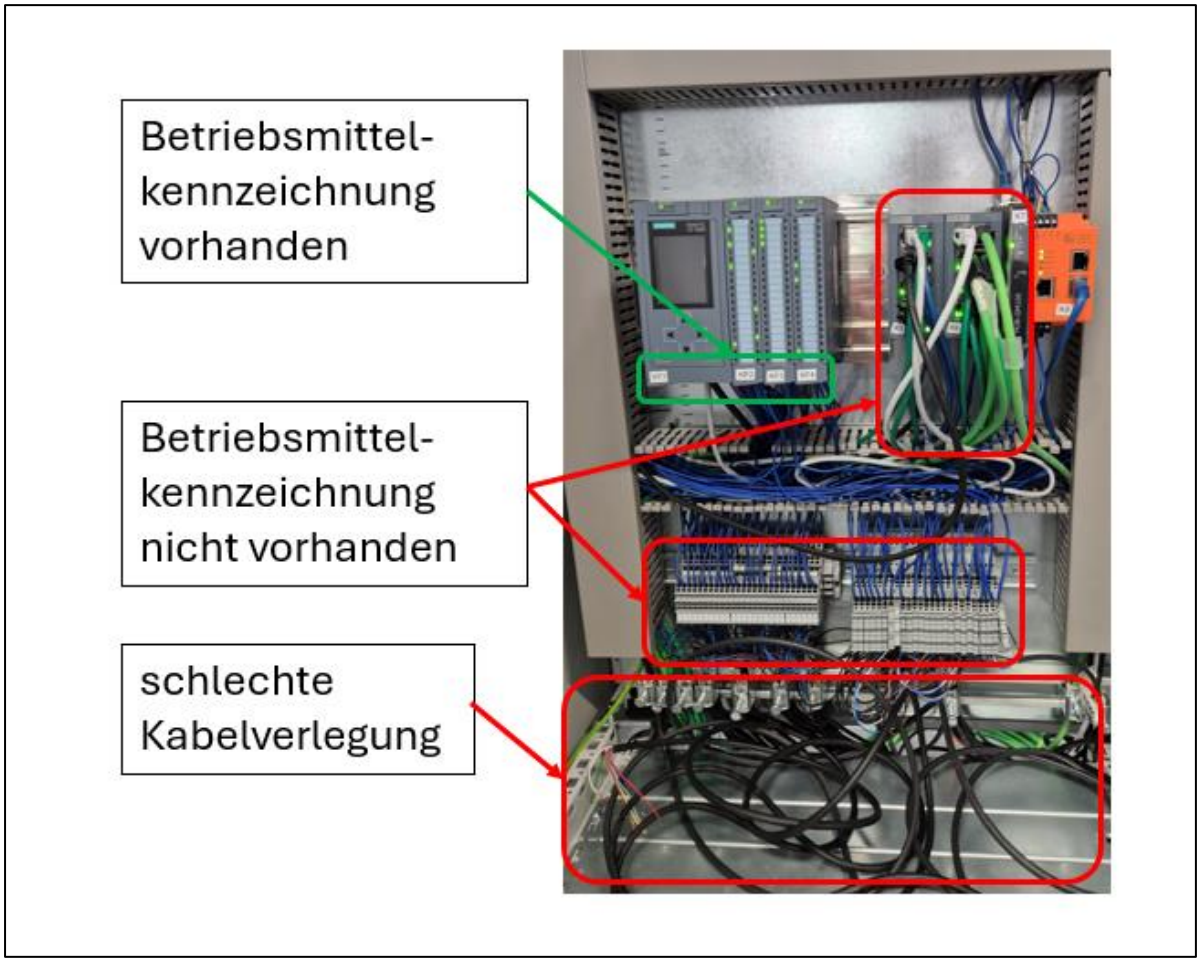


Abbildung 4: Topologieansicht SPS ????

Im Schaltschrank des PTZ wurden mehrere Mängel festgestellt, die sowohl die Übersichtlichkeit als auch die Wartungsfreundlichkeit der Installation erheblich beeinträchtigen. Wie in Abbildung 4: Ansicht SPS und Klemmreihen zu erkennen ist, sind die Betriebsmittelkennzeichnungen an den SPS-Komponenten im oberen Bereich des Schaltschranks (grün markiert) vorhanden. Ebenso wurden die beiden Switches, die sich rechts neben der SPS befinden, korrekt gekennzeichnet. Allerdings besteht weiterhin das Problem, dass die Kabel, die zu den Switches führen, keine Beschriftung aufweisen. Dies erschwert die Zuordnung der einzelnen Verbindungen und erhöht den Aufwand bei der Fehlersuche oder bei Wartungsarbeiten erheblich.

Ein weiteres Problem, das in Abbildung 4 deutlich wird, betrifft die beiden Klemmenreihen, die sich unterhalb der SPS befinden. Beide Klemmenreihen weisen keinerlei

Betriebsmittelkennzeichnungen auf, wodurch die Zuordnung der angeschlossenen Sensoren und Aktoren außerordentlich erschwert wird. Insbesondere bei Instandhaltungsarbeiten führt dies zu unnötigen Verzögerungen, da ohne eine klare Kennzeichnung fehlerhafte Verbindungen oder Bauteile nur schwer und zeitaufwendig identifiziert werden können.

Zusätzlich ist in der unteren Sektion des Schaltschranks eine unzureichende Kabelverlegung zu erkennen. Die Kabel sind nicht ordentlich gebündelt, sondern verlaufen lose und unsortiert, was sowohl die Übersichtlichkeit beeinträchtigt als auch das Risiko von Beschädigungen erhöht. Eine solche Verlegung behindert zudem den Zugang zu anderen Bauteilen, was die Wartung weiter erschwert. Zudem liegt weiterhin kein aktualisierter Schaltplan vor, der die Verdrahtung und Anschlüsse dokumentiert. Ohne diese Dokumentation sind Änderungen an der bestehenden Installation nicht möglich, was die Problemlösung und zukünftige Erweiterungen stark einschränkt.

Um diese Defizite zu beheben, wurden verschiedene Maßnahmen eingeleitet. Zunächst wurde eine Bestellung für Klemmenkennzeichnungen aufgegeben, um die beiden Klemmenreihen nachträglich zu beschriften und ausreichend Anschlussports für die Wenglor- und ifm-Kameras bereitzustellen. Dies wird die Zuordnung der Sensoren und Aktoren erleichtern und die Wartungsarbeiten deutlich effizienter gestalten.

Die unzureichende Kabelverlegung soll durch den Einsatz von Kabelkanälen und Kabelbindern verbessert werden, um die Kabel sauber zu bündeln und die Übersichtlichkeit im Schaltschrank zu erhöhen. Eine strukturierte Kabelführung reduziert das Risiko von Beschädigungen und sorgt für besseren Zugang zu den einzelnen Komponenten.

Ein weiterer wichtiger Lösungsansatz ist die Verwendung eines Y-Kabels von ifm. Wie in der Abbildung Verweis auf welche Abbildung zu erkennen ist, könnte das Kabel optimal in die bestehende Struktur integriert werden. Es wurde bestellt, um den ifm-Switch X70 mit dem ifm-Switch AL1100 zu verbinden, ohne zusätzliche Anschlussports im Schaltschrank zu belegen. Diese Lösung ist besonders praktisch, da das Y-Kabel direkt an das bereits verlegte Kabel angeschlossen werden kann. Dadurch wird nicht nur die Netzwerkintegration vereinfacht, sondern auch der Platzbedarf im Schaltschrank reduziert. [3]

Trotz dieser Verbesserungen bleibt das Fehlen eines aktualisierten Schaltplans ein bedeutendes Hindernis. Dieser ist erforderlich, um die Verdrahtung vollständig zu

dokumentieren und Änderungen oder Erweiterungen in Zukunft nachvollziehbar zu gestalten. Die Umsetzung der genannten Maßnahmen trägt jedoch bereits jetzt dazu bei, die Übersichtlichkeit, Funktionalität und Wartungsfreundlichkeit des Schaltschranks deutlich zu erhöhen.

Die Kosten für die gesamte Bestellung beliefen sich auf etwa 150 €. Die Umsetzung dieser Maßnahmen wird die Übersichtlichkeit und Funktionalität im Schaltschrank verbessern und eine effizientere Wartung sowie Erweiterung der Installation ermöglichen.

4. Weiterentwicklung SPS-Architektur- Grundlagen

Die Weiterentwicklung der SPS-Architektur im PTZ ist ein kontinuierlicher Prozess, der auf die Optimierung von Funktionalität, Übersichtlichkeit und Effizienz abzielt. Dazu gehört unter anderem das konsequente Auskommentieren von Programmabschnitten, um deren Zweck nachvollziehbar zu machen, sowie das Umbenennen von Variablen, um die Lesbarkeit zu erhöhen. Diese Maßnahmen erleichtern sowohl die Zusammenarbeit als auch die spätere Wartung der Steuerung.

Zu Projektbeginn zeigte sich, dass das Programm kaum auskommentiert war und viele Variablen irreführend benannt waren. Das Verständnis des Codes gestaltete sich äußerst schwierig und erforderte viel Zeit. Es musste oft durch Trial-and-Error-Methoden sowie durch Abgleich mit der Produktionslinie nachvollzogen werden, wobei die Linie so lange betrieben wurde, bis ein Fehler auftrat. Diese Probleme verdeutlichten die Notwendigkeit, die Dokumentation und Struktur des Programms zu verbessern.

Die Weiterentwicklung umfasst auch die fehlerfreie Neuprogrammierung von Funktionen sowie die Optimierung der Datenstruktur, um eine effizientere Verarbeitung zu gewährleisten. Ziel ist es, eine robuste und flexible Architektur zu schaffen, die den Anforderungen des PTZ gerecht wird.

Ein weiterer Aspekt ist der interdisziplinäre Ansatz, der über die reine SPS-Programmierung hinausgeht. So müssen auch die programmierten Abläufe der Roboter analysiert werden, um festzulegen, wann diese Statusbits zurückgeben oder wann die SPS bestimmte Register überschreiben darf. Diese ganzheitliche Betrachtung stellt sicher, dass alle Systemkomponenten reibungslos zusammenarbeiten und eine stabile sowie effiziente Steuerung ermöglichen.

Im Folgenden wird erläutert, was das TIA Portal ist, wie Hierarchien aufgebaut werden, welche Bausteine existieren und welche Programmiersprachen für das TIA Portal existieren.

Das TIA Portal (Totally Integrated Automation Portal) von Siemens ist eine zentrale Softwareplattform für die Automatisierungstechnik. Es ermöglicht die Programmierung, Konfiguration und Überwachung von Steuerungen, HMI-Systemen (Human Machine

Interface) und Antrieben in einer integrierten Umgebung. Durch die benutzerfreundliche Oberfläche und die nahtlose Integration verschiedener Automatisierungskomponenten optimiert das TIA Portal die Effizienz und Flexibilität von Engineering-Prozessen. Die Plattform unterstützt zahlreiche Siemens-Produkte und ist besonders in der industriellen Automatisierung weit verbreitet. Des Weiteren bietet es die Möglichkeit Siemensfremde Produkte zu integrieren, wodurch ein Zusammenspiel von Aktoren und Sensoren unterschiedlicher Hersteller ermöglicht wird. Quelle?

Um die Hierarchien im TIA Portal zu verstehen, müssen zuerst die unterschiedlichen Bausteine erläutert werden. Es existieren Organisationsbausteine (OB), Funktionsbausteine (FB), Funktionen(FC) und (Instanz-) Datenbausteine(DB). Quelle?

4.1 Der Organisationsbaustein

Der Organisationsbaustein ist für die Gliederung unterschiedlicher Aufgaben der CPU zuständig. Jeder OB wird mit FBs, DBs und FCs gefüllt und zu einer definierten Zeit aufgerufen, der sogenannten Zykluszeit. Sie gibt an, in welchem Zeitintervall der OB aufgerufen wird. Dies kann von wenigen Millisekunden bis zu mehreren Sekunden reichen. Ebenfalls kann ein OB zu einem gewissen Zeitpunkt aufgerufen werden, oder nur ein einziges Mal. Die Zyklischen OBs beinhalten meist die Programmierung des Ablaufprogramms (Zustands- und Logistiksteuerung, Regelung, Überwachung und Fehlerhandling, Berechnungen und Datenverarbeitung). OBs die nur zu bestimmten Zeiten aufgerufen werden sind meist für die Datenübertragung an angeschlossene, aber eigenständige, Netzwerke zuständig. Gesammelte Daten und Trends könnten an einen Backupserver übertragen werden, oder Informationen über Ressourcenverbrauch kann an den Einkauf gesendet werden. OBs, die nur einmal aufgerufen werden, sind Initialisierungsbausteine. Beispielsweise nach Neustart der CPU wird ein bestimmtes Prozessbild geladen, um Fehler bei Flip-Flops oder anderen Zuständen zu vermeiden. Quelle?

4.2 Der Datenbaustein

Es existieren drei unterschiedliche Arten von Datenbausteinen: Globaler, Instanz- und Multiinstanz-Datenbaustein. Quelle? Abb. 1 zeigt das Zusammenspiel der einzelnen Bausteine miteinander.

Abbildung 5

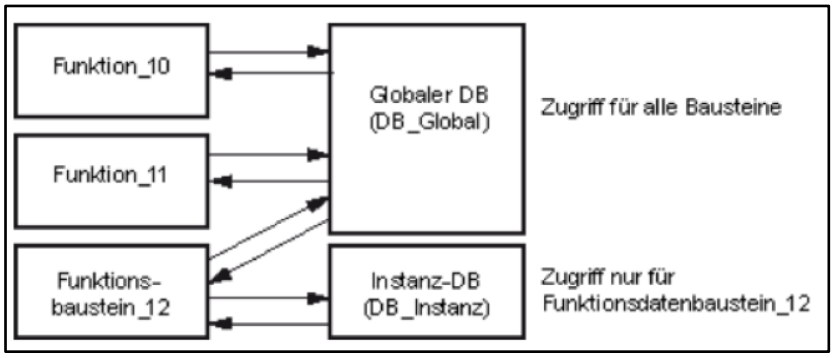


Abbildung 5: Kommunikationswege von Datenbausteinen [4]

Zu jedem Funktionsbaustein gehört ein eindeutiger Instanz-DB, es können sich auch mehrere FBs einen einzigen Multiinstanz-DB teilen. Funktionen können lediglich in Globale DBs schreiben und aus ihnen lesen. Datenbausteine sind spezielle Speicherbereiche, die zur Ablage von Daten in einem Steuerungssystem genutzt werden. Sie ermöglichen die strukturierte Verwaltung und den Zugriff auf Variablen, die für Programme oder Funktionen benötigt werden. Datenbausteine helfen, Daten konsistent zu organisieren und den Speicherzugriff in der SPS-Programmierung effizient zu gestalten. Quelle?

4.3 Die Funktion

Funktionen bieten die Möglichkeit, Parameter im Anwenderprogramm zu übergeben. Sie eignen sich deshalb besonders gut zur Programmierung von häufig wiederkehrenden komplexen Konstrukten, wie z.B. Berechnungen. Eine Funktion enthält ein Programm, welches immer dann ausgeführt wird, wenn die Funktion von einem anderen Baustein aufgerufen wird. Einfache Funktionen wären z.B. AND, OR, NAND, CMP (compare) Funktionen. Sie benötigen keine Parameter oder Variablen und damit auch keine Datenbausteine. Quelle?

4.4 Der Funktionsbaustein

Der Funktionsbaustein spielt in SPS-Programmierung durch seine Modularität und Wiederverwendbarkeit eine zentrale Rolle. Ein FB fasst eine bestimmte Logik in einer modularen Einheit zusammen, die an unterschiedlichen Stellen im Ablaufprogramm immer wieder verwendet werden kann. Den Aufruf eines Funktionsbausteins wird Instanz genannt. Der FB benötigt bei jedem Aufruf einen eignen Instanz-DB, um Variablen und Zustände des Bausteins zu speichern. Quelle?

4.5 Die Hierarchie

Im TIA Portal existieren zwei Arten von Hierarchien. Zum einen die Hardware-Hierarchie und zum anderen die Programm Hierarchie. Die Hardwarehierarchie ergibt bei sich bei der Anlagenplanung und wird in der Topologie-Ansicht bzw. in der Netz-Ansicht erstellt. Hier werden alle benutzen Geräte (I/O-Link, E/A-Module, Motoren, Switches, HMI-Panels, etc.) mittels Drag und Drop aus dem Siemenskatalog platziert und nach realem Vorbild Verbunden. Siehe hierzu Abb. 2. Abbildung 6

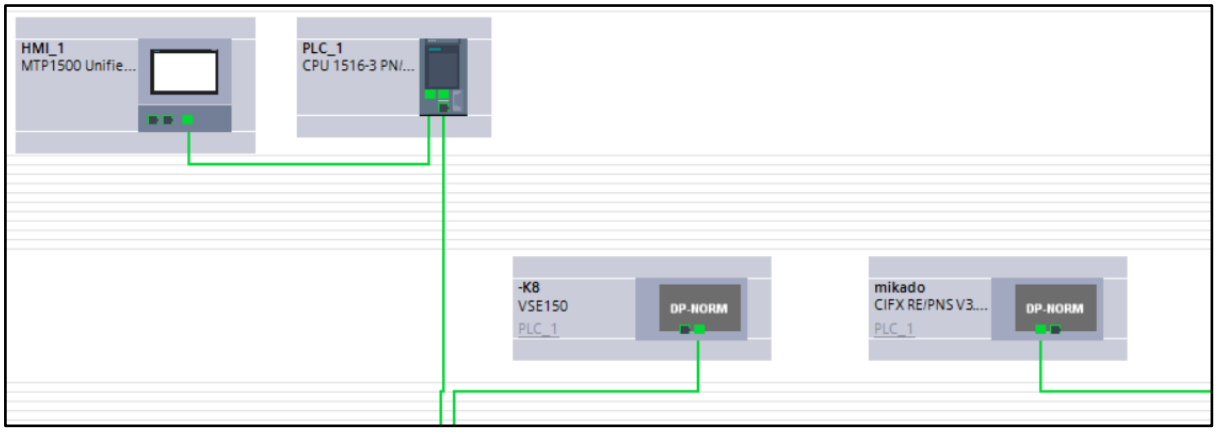


Abbildung 6: Topologie-Ansicht aus dem TIA Portal von Siemens [5]

In Abb. 2 ist der PLC_1 zu sehen, was die SPS darstellt. Sie muss mit allen anderen Hardware-Teilnehmern verbunden sein, um einen Datenaustausch zu gewährleisten. Für das Verbindungsprotokoll wird bei Siemens-SPSen meist PROFINET verwendet. Wenn Siemensfremde Hardware zum Einsatz kommt, dann existieren von dem jeweiligen Hersteller

GSD-Dateien (Generic Station Description), die im jeweiligen Projekt installiert werden muss, damit die Hardware implementiert werden kann. Des Weiteren müssen IP-Adressen vergeben werden. Zum einem im TIA Portal und zum anderen an der Hardware selbst. Eine eindeutige IP-Adresse und ein eindeutiger Name sind wichtig, damit die Adressierung über den Bus funktioniert.

Die zweite Hierarchieart ist die Strukturierung des Ablaufprogramms. In Abbildung 3 ist der Ausgangszustand der SPS in der Lernfabrik in Friedrichshafen.

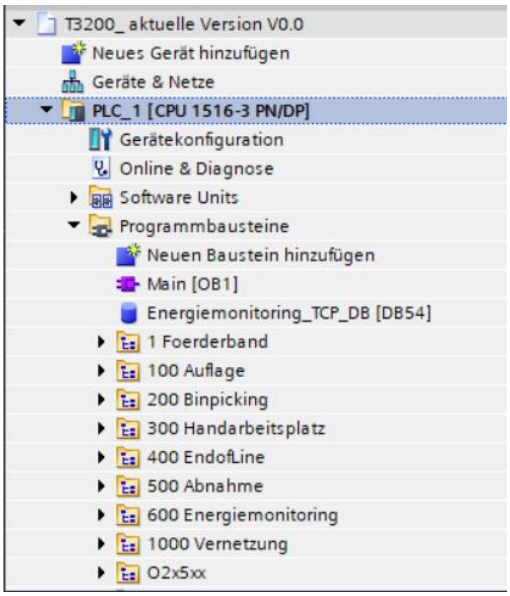


Abbildung 7: Programmhierarchie [5]

Unter der CPU/SPS existiert der Reiter Programmbausteine. Unterhalb dieses Reiters ist der Main [OB1] zu sehen. Im OB1 werden, in dieser Anlage, alle anderen Funktionsbaustein und Funktionen aufgerufen, die in den anderen Reitern programmiert wurden. Für das Erstellen von Hierarchien sollten einige Regeln beachtet werden, um eine Übersichtlichkeit zu gewährleisten: Quelle?

1. **Modularisierung**: Zerlege das Projekt in sinnvolle, unabhängige Module, die spezifische Funktionen abdecken (z. B. Antriebe, Sensorik, HMI). So wird die Wartung und Wiederverwendbarkeit erleichtert.

2. **Trennung von Daten und Logik**: Verwende separate Datenbausteine für Variablen und Funktionsbausteine für die Logik. So bleiben Daten und Steuerungslogik sauber getrennt, was die Übersichtlichkeit verbessert.

3. **Konsistente Benennung**: Verwende eine einheitliche Benennungsstruktur für Bausteine, Variablen und Module. Klare, sprechende Namen helfen dabei, die Funktion jedes Elements leicht zu verstehen und Verwechslungen zu vermeiden.

4. **Hierarchie nach Funktionalität und Priorität**: Ordne Bausteine nach ihrer Funktionalität und Priorität in der Steuerung. Kritische Prozesse sollten oben in der Hierarchie stehen, während untergeordnete Prozesse in den unteren Ebenen organisiert werden.

5. **Vermeidung redundanter Bausteine**: Teile häufig genutzte Funktionen in wiederverwendbare Funktionsbausteine auf, um Redundanz zu reduzieren und den Speicherbedarf zu optimieren.

6. **Dokumentation und Kommentierung**: Dokumentiere die Hierarchie und kommentiere wichtige Module und Bausteine. Eine gute Dokumentation erleichtert das Verständnis und die spätere Wartung des Projekts.

4.6 Unterschiedliche Programmiersprachen

Das TIA Portal bietet die Möglichkeit Funktionen und Funktionsbausteine in drei unterschiedlichen Programmiersprachen zu programmieren. Quelle

1. **FUP (Funktionsplan)**: Eine grafische Programmiersprache, in der Logikverknüpfungen und Funktionen in Form von Blöcken dargestellt werden. FUP ist besonders intuitiv, da es visuell an Schaltpläne erinnert.

2. **KOP (Kontaktplan)**: Eine grafische Darstellung, die an klassische Relais-Schaltungen erinnert und auf Schaltplänen basiert. KOP wird häufig in der Automatisierung verwendet und ist besonders gut geeignet für Personen, die an elektromechanische Schaltungen gewöhnt sind.

3. **SCL (Structured Control Language)**: Eine textbasierte Programmiersprache, die sich an die Hochsprache Pascal anlehnt. SCL ermöglicht eine effiziente, strukturierte Programmierung komplexer Steuerungslogiken und ist besonders nützlich für mathematische Berechnungen und Schleifen.

Dies hat den Vorteil, dass jede Person entsprechend ihren Stärken in der von ihr bevorzugten Sprache programmieren kann. Dieser Vorteil kann auch zu dem Nachteil führen, dass die Verständlichkeit leidet. Personen, die in FUP oder KOP gut sind, können mit SCL-Probleme bekommen. Eine Kenntnis in allen Sprachen ist daher sinnvoll.

5. Weiterentwicklung SPS-Architektur – Ergebnisse

Im Folgenden soll nun der Ausgangszustand beschrieben werden, anschließend wird darauf eingegangen, wo die Probleme liegen und welche Problematik mit wechselnden Teams einhergeht. Daraufhin wird erläutert welche Veränderungen an dem bestehenden Programm vorzunehmen sind, was die Ziele sind und wie dies umgesetzt werden können.

5.1 Ausgangszustand

Der Zustand, in dem unser Team das vorliegende Programm übernommen hat, sieht wie folgt aus. Es existiert ein Mainbaustein, in dem 14 programmierte Netzwerke vorliegen. In dem Netzwerk 1 werden die Sicherungen des Förderbandes überwacht und ggf. eine Verriegelung ausgeführt. Diese Verriegelung existiert exakt so noch einmal in dem Baustein „001FB_active“. In diesem wird auch der Start und Stopp des Förderbands realisiert.

In **Netzwerk 2** wird der Baustein „001FB_active“ aufgerufen und falls die Bedingungen passen, damit das Förderband laufen soll, werden die jeweiligen Schütze dafür geschaltet.

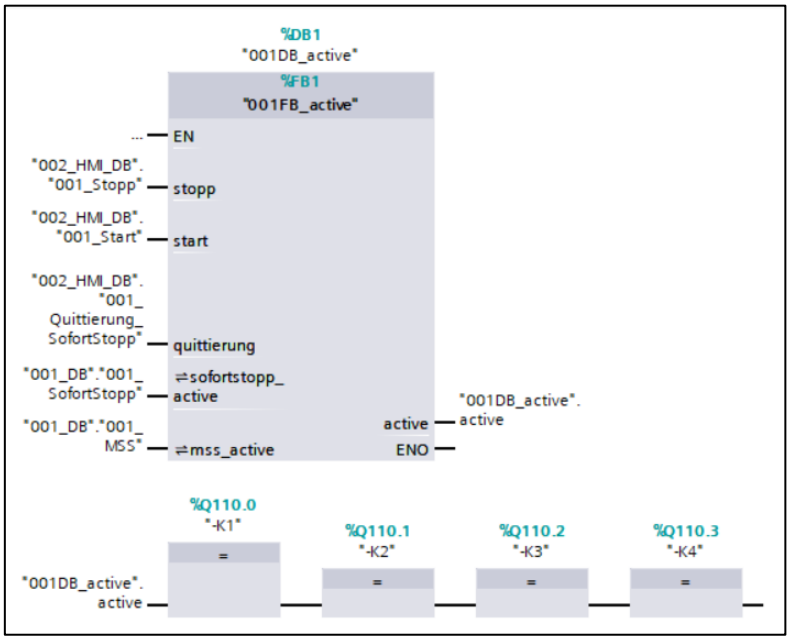


Abbildung 8: Netzwerk 2 aus dem Ausgangsprogramm [5]

Wo ist der Verweis auf die Abbildung 8????

Netzwerk 3 & 4 steuern die Umleger. Hierbei handelt es sich um Funktionen, die in SCL geschrieben wurden und eine Schrittkette nutzen, um das Umlegen zu realisieren. Diese funktionieren einwandfrei und beschreiben einen DB, in dem steht, ob der Umleger gerade aktiv ist.

Netzwerk 5 regelt die die Stopper 0 und 5, sowie die Befehle an Roboter 1. Diese sind die Stopper an der Station „QR-Scan“ und „Rob-Leading“. Die vorgefundene Funktion besagt, dass der Roboter 1 immer Carrier auf das Förderband setzt, solange kein Carrier an seiner Station ist. Weiterhin soll der Roboter 1 von der Station „Rob-Leading“ das Bauteil auf den Husky setzten, wenn dieses von dem „End of Line“ als I.O. befunden wurde. Weiterhin wird verhindert, dass weitere Bauteile aufgelegt werden, bis das „Bin-Picking“ abgeschlossen ist. Des Weiteren befindet sich in **Netzwerk 5** eine QR-Code-Verarbeitung, welche die QR-Codes der einzelnen Werkstücke in der Datenbank „ExternalClients“ speichert. Diese Datenbank speichert ebenfalls, die Temperatur, Schwingung und Drehzahl der einzelnen Motoren, die Umlegerpassierung und die Variante.

Netzwerk 6 beinhaltet den FC „001_stopper_call“. Dieser ist dafür zuständig die Stopper zu schalten. Es existieren 6 Stopper. An den Stationen „Belt-Leading“, „Bin-Picking“, “Assembly“, „EoL“, „Rob-Leading“ und vor Umleger 2, damit keine Kollision an dem Umleger selbst stattfindet. Die ersten beiden Stopper der Funktion wurden über Funktionen realisiert, welche theoretisch für alle genutzt werden könnte wie in Abb. 5 zu sehen ist.

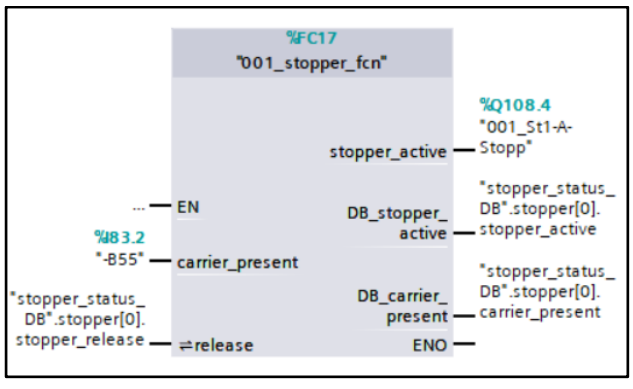


Abbildung 9: Wiederverwendbare Stopperfunktion [5]

Alle anderen Stopper wurden händisch über Logik umgesetzt.

Netzwerk 7 erzeugt Fertigungsdaten der einzelnen Bauteile und speichert diese in der „Pseudodatenbank“ ab. Diese Datenbank wurde etabliert, aber nie in Betrieb genommen. Das bedeutet, dass sie zwar einsatzbereit ist, aber nicht angeschlossen ist. Sie eignet sich, um spätere Fertigungsdaten zwischenzuspeichern oder als Vorbereitung um Daten wie Seriennummer o.ä. auf eine externe Datenbank auszulagern.

Netzwerk 8 enthält einen FB namens „008_EoL_Schnittstellen“, welcher je nach Auswertung der Qualitätsprüfung die unterschiedlichen Roboter ansteuern soll. Dieser FB wurde allerdings nie in Betrieb genommen.

Netzwerk 9 schreibt die Daten aus dem FC „ExternalClients“, der bereits in **Netzwerk 5** erwähnt wurde, in die Datenbank. Diese Daten sollten mit jedem neuen Werkstück überschrieben werden, dadurch handelt es sich um eine temporäre Speicherung.

Netzwerk 10 sendet Steuerbytes an die Mikado-Steuerung. Je nachdem welche Variante am HMI ausgewählt wird, wird eine andere Zahl übertragen. In der Robotersteuerung sind die unterschiedlichen Varianten hinterlegt und je nachdem wird die jeweilige Variante umgesetzt. Wenn der Roboter mit dem Einsetzen des Pins fertig ist, sendet er ein Bit zurück an die SPS, wodurch mit der Produktion fortgefahren werden kann.

Netzwerk 11 verhindert das weitere Aufsetzen von Werkstücken von Roboter 1. Erst wenn der Roboter am „Bin-Picking“ fertig ist, wird ein neues Werkstück auf den Träger gesetzt. Hierdurch wird eine Kollision beim „Bin-Picking“ vermieden. Falls ein Carrier weiterfahren sollte, kann der Mikado-Roboter den Pin nicht richtig einsetzen und fährt eine Kollision.

Netzwerk 12 regelt die Fehlermeldungen, wobei es sich hierbei lediglich um zwei verschiedene Kommunikationsfehler handelt, die mittels einer Taste auf dem HMI quittiert werden können. Die genaue Definition der Fehlerquelle kann aus Gründen mangelnder Kommentierung nicht angegeben werden.

Netzwerk 13 beinhaltet einen Zeitmesser. Dieser zählt einen Input bis 7 und wird anschließend zurückgesetzt. Dadurch dass der Eingang mehrfach verwendet wird, keinerlei Kommentierung vorliegt, kann auch hier der Gedanke nicht nachvollzogen werden, was mit diesem Timer gemacht werden sollte.

Netzwerk 14 zeichnet den Energieverbrauch auf. Hierfür wurde ein vorgefertigter Funktionsbaustein verwendet, der die Daten in einen eigenen DB schreibt. Wie dieser Baustein funktioniert, wird nicht erläutert. Scheinbar verbindet er sich mit einer IP-Adresse, und empfängt von einem Gerät die jeweiligen Daten. Diese IP-Adresse ist jedoch nicht vergeben. Hierbei handelt es sich um 10 Messgeräte, von denen die Protokollierung fehlt. Dieser ist in Abb. 6 zu sehen.

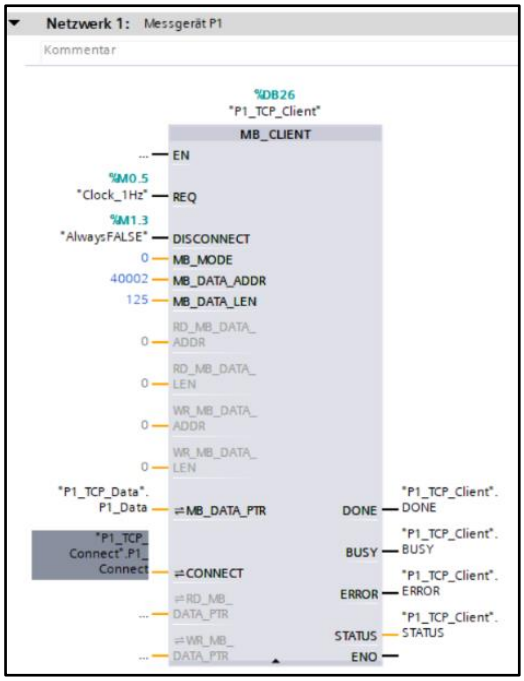


Abbildung 10: Messbaustein 1 von 10 [5]

5.2 Die Variablen

Woher kommt der Prozentwert? genauere Herleitung..

Die PLC-Variablen sind zu 80% unkommentiert. Es existieren unterschiedliche Stile des Benennens und ca. 10% der Variablen wurden nicht benannt und stehen noch mit „Tag…“ in der Tabelle. Hinzu kommt, dass Adressen mehrfach vergeben wurden, wodurch es zu Übersetzungsfehlern kommen kann, oder schlimmer - zu Sicherheitsrisiken. Die doppelte Adresse wird in dem Programm mehrfach umgesetzt, wodurch zu unvorhergesehenen Ereignissen kommt. Des Weiteren existieren doppelte Variablen oder Abkürzungen, die unverständlich sind und nicht ersichtlich wird was damit gemeint ist. Abb. 7 zeigt eine mangelnde Benennung und Kommentierung.

80	-K1	001_DQ16	Bool	%Q110.0					
81	-K2	001_DQ16	Bool	%Q110.1					
82	-K3	001_DQ16	Bool	%Q110.2					
83	-K4	001_DQ16	Bool	%Q110.3					
84	Merker Pnoz	Standard-Variablen...	Bool	%Q110.4					
85	Tag_7	Standard-Variablen...	Bool	%Q110.5					
86	Tag_8	Standard-Variablen...	Bool	%Q110.6					
87	Tag_9	Standard-Variablen...	Bool	%Q110.7					
88	Mikado_out	Standard-Variablen...	Word	%QW112					
89	Tag_16	Standard-Variablen...	"UDT_mikadoControl"	%Q112.0					
90	Mikado_out_profi	Standard-Variablen...	"UDT_mikadoParameter"	%Q116.0					

Abbildung 11: Auszug aus Variablentabelle [5]

5.3 Die Roboter

Zu unterscheiden sind drei Roboter in der Anlage. Roboter 1 ist für das „Belt-Leading“ und das „Rob-Leading“ verantwortlich. Roboter 2 ist für das „Bin-Picking“ verantwortlich und ist wie alle Roboter ein UR (Universal Robot), nur, dass beim „Bin-Picking“ eine Mikado-Software und -Hardware zum Einsatz kommt. Damit ist es möglich Gegenstände aus einer Kiste heraus zu greifen und zu vermessen. Roboter 3 kommt zum Einsatz, wenn der EoL-Test schlecht ausfällt und das Werkstück aussortiert werden muss. Da alle Roboter von UR stammen ist auch ihre Programmierung ähnlich durchführbar. In Abb. 8 ist der Teil einer Programmierung zu sehen.

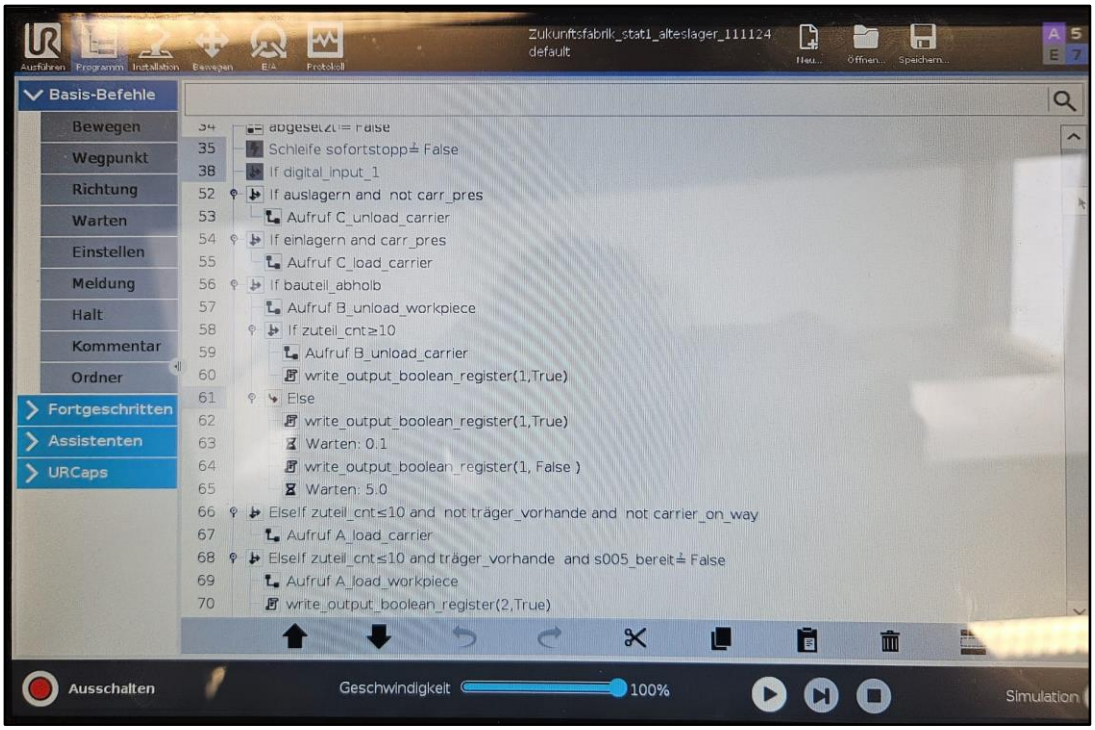


Abbildung 12: Auszug Roboterprogrammierung von Roboter 1 [6]

Wie erkennt man eine Programmierspache C???

Zu erkennen ist, dass die Programmiersprache stark an C erinnert. Es können Verzweigungen und Entscheidungen programmiert werden, wodurch der Roboter vielseitig einsetzbar wird.

Bei **Roboter 1** ist eine Programmierung vorhanden, diese ist jedoch nur für Demonstrationszwecke geeignet. Es können Carrier und Werkstücke auf das Förderband gesetzt werden und auch fertige Bauteile auf das Transportsystem geladen werden. Dennoch ist die Programmierung unvollständig, da zum einen maximal 10 Teile hintereinander gefertigt werden können und die Programmierung des Roboters so ungünstig realisiert wurde, dass tiefgreifende Eingriffe notwendig sind, um das Ziel von 50 Werkstücken zu erreichen. Hauptproblem ist, die sperrige und unkomfortable Art den Roboter zu programmieren, da diese nur über das Touchpad getätigt werden kann. Des Weiteren sind die einzelnen If-Anweisungen nicht voneinander unabhängig und eine deutliche Fallunterscheidung ist auch nicht zu erkennen.

Roboter 2 ist zu 100% programmiert. Er funktioniert in der Regel ohne Fehler und verfügt über alle nötigen Fallunterscheidungen zu den Varianten. Durch das angeschlossene Mikado-System ist die Programmierung des Roboters deutlich angenehmer, da diese über eine Tastatur und den Bildschirm umgesetzt werden kann.

Der **Roboter 3** ist überhaupt nicht programmiert. Es ist in den vorliegenden Programmen keines dabei, welches an die Anlage angepasst wurde. Hinzu kommt, dass der Roboter nicht an das Profi-Net angeschlossen ist.

5.4 Das Human Maschine Interface (HMI)

Das HMI ist direkt in die Schaltschranktür integriert. Es verfügt über mehrere angelegten Seiten. Auf der einen kann das Förderband ein- und ausgeschaltet werden wie in Abb. 9 zu sehen ist. Auf einer anderen kann die gewünschte Variante ausgewählt werden, sie Abb. 10. In der Vorliegenden Parametrierung sind 5 Varianten hinterlegt. Es existieren noch zahlreiche weitere Bilder, diese wurden jedoch nicht bearbeitet oder mit der Programmierung verbunden.



Abbildung 13: Auszug HMI-Bilder [5]

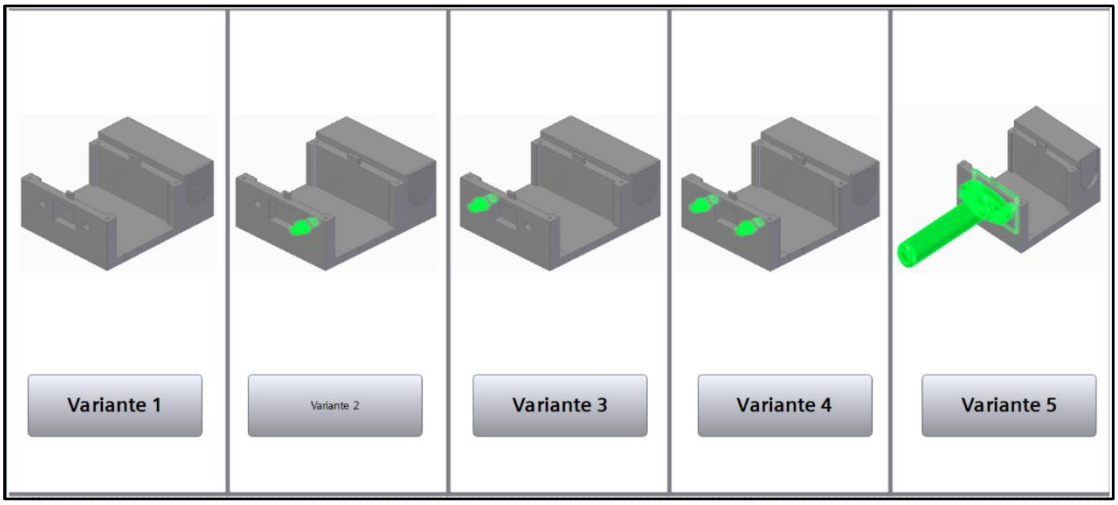


Abbildung 14: Auszug HMI-Bilder; Variantenauswahl [5]

5.5 Die Problematik

Ein zentrales Problem der Zukunftsfabrik sind ihre wechselnden Teams. Dadurch, dass die jeweiligen Teams nur 6 Monate an der Anlage arbeiten können und keine Einarbeitung in das bestehende System erfolgt, treten jährlich die gleichen Probleme auf. Es wird viel Zeit investiert, um sich in die Anlage hineinzuarbeiten, herauszufinden, wo die Probleme liegen, sich ein Konzept überlegen, etwas Umsetzen und die benötigten Informationen

zusammensuchen. Diese Punkte spiegeln sich in der Programmierung wider. Es wurden unterschiedlichste Programmierstile eingebracht, sowie verschiedene Methoden der Namensvergabe. Hinzu kommt, dass offensichtlich auch Personen mit wenig, bis keine Programmiererfahrung Funktionen geschrieben haben. Dies wird vor allem bei der fehlenden Kommentierung und der fragwürdigen Programmierung ersichtlich wie in Abb. 11 zu sehen ist.

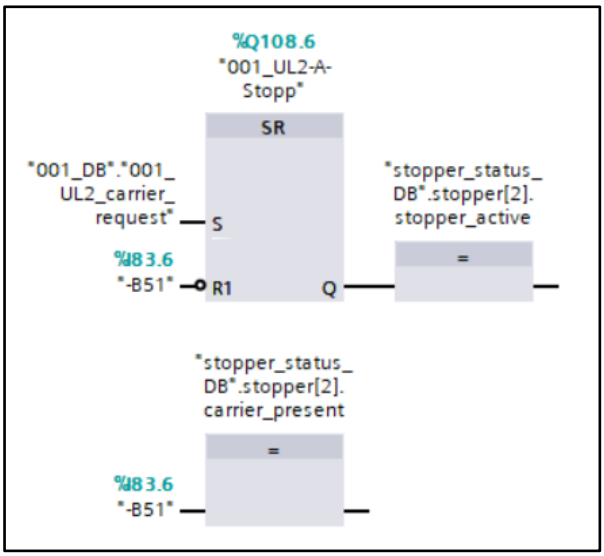


Abbildung 15: Programmierfehler [5]

Einen Ausgang über ein Flipflop zu legen kann zu ernsthaften Problemen führen. Diese und weitere Fehler gilt es ausfindig zu machen und zu beheben, was ein gewisses Pensum der zur Verfügung stehenden Zeit in Anspruch nimmt.

Ein weiteres Problem ist, dass kein klares Konzept existierte. Jeder Jahrgang hatte neue Ideen, die meist nur teilweise umgesetzt werden konnten. Ohne klare Vorstellungen der Anlage kann jedoch keine Programmierung vorgenommen werden. Da eine Programmierung immer klare Definitionen für die Anlage schafft, müssen diese Definitionen geklärt sein. Häufige Änderungen oder unterschiedliche Vorstellungen skalieren die Fehler und den Programmieraufwand.

5.6 Konzept: “end of line”

Der Prozess der automatisierten Demontage in der Anlage des PTZ zielt darauf ab, Werkstücke am Ende der Produktionslinie zu prüfen und je nach Zustand entweder für die weitere Verarbeitung freizugeben (Übergabe des Werkstücks an „Husky“) oder auszusortieren.

Verweis auf Abbildung? "in nachfolgender Abbildung 16 als ….."

Der Prozess ist im nachfolgenden Flussdiagramm dargestellt.

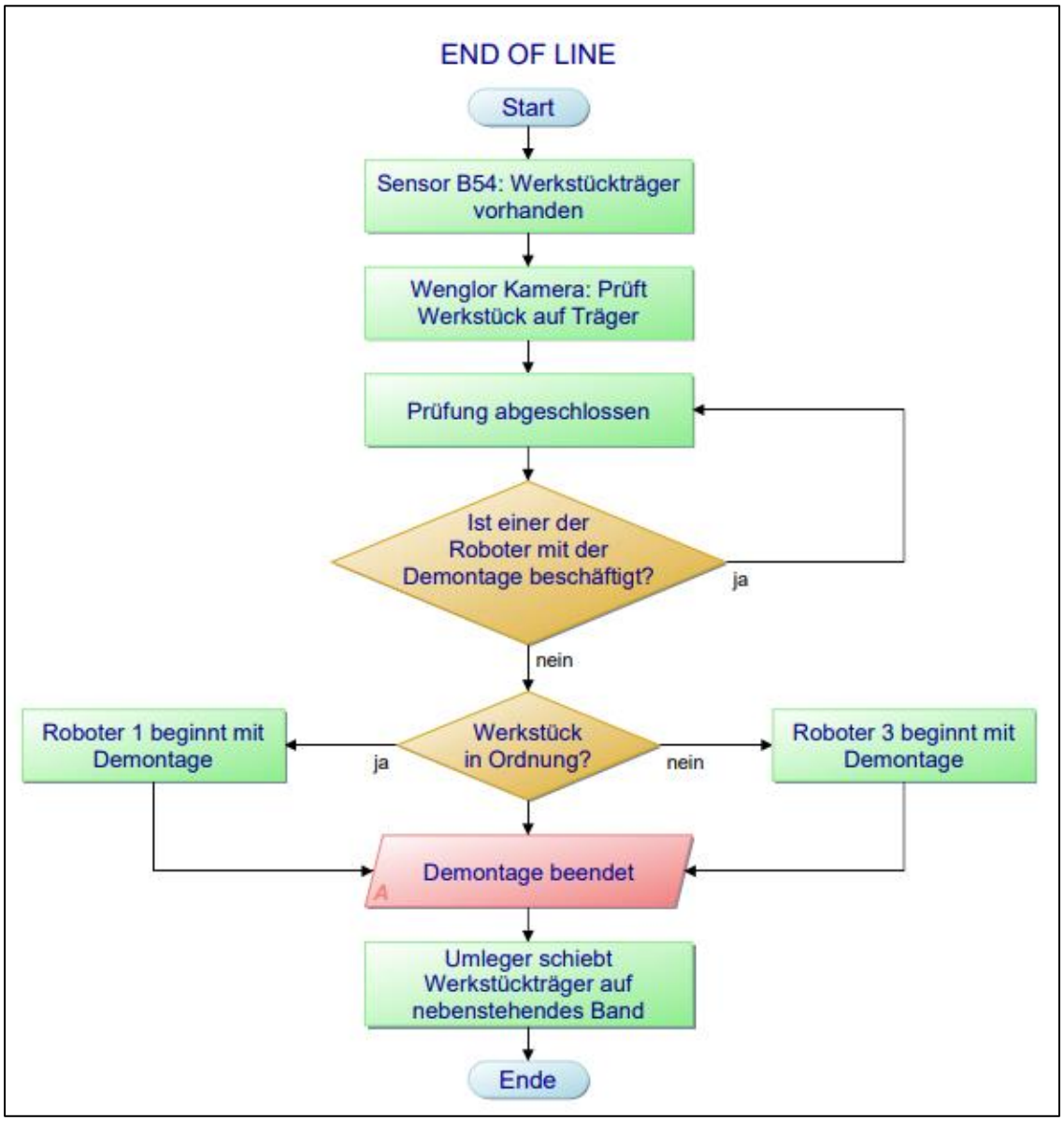


Abbildung 16: Konzept "end of line"

Zunächst überprüft der Sensor B54, ob ein Werkstückträger vorhanden ist. Diese Abfrage stellt sicher, dass der Prozess nur fortgesetzt wird, wenn tatsächlich ein Träger mit einem Werkstück

in der Station vorhanden ist. Anschließend führt die wenglor-Kamera eine Qualitätsprüfung durch, bei der überprüft wird, ob das Werkstück den Anforderungen entspricht.

Nach Abschluss der Prüfung entscheidet die Steuerung, wie weiter verfahren wird. Wenn das Werkstück in Ordnung ist und keiner der Roboter mit einer Demontage beschäftigt ist, beginnt Roboter 1 mit der Demontage. Sollte Roboter 1 bereits im Einsatz sein (z.B. Auflegen neuer Werkstückträger), dann wird gewartet, bis dieser den aktuellen Prozess abgeschlossen hat und mit dem nächsten Prozess fortfahren kann.

Nach Abschluss der Demontage, unabhängig davon, ob das Werkstück in Ordnung ist oder nicht, wird der Werkstückträger zunächst durch einen Stopper zurückgehalten. Sobald die Demontage vollständig abgeschlossen ist, wird der Stopper nach unten gefahren, sodass der Werkstückträger über diesen hinweg, auf den Umleger befördert werden kann. Der Umleger schiebt anschließend den leeren Werkstückträger auf ein benachbartes Förderband, um es wieder für die nachfolgenden Prozessschritte bereitzustellen. Dieser Ablauf sorgt dafür, dass der Demontageprozess effizient abgeschlossen wird und die Produktionslinie kontinuierlich arbeitet.

Vor der Einführung dieses Prozesses existierte kein spezifisches Programm für den End-of-Line-Bereich, wodurch alle Werkstückträger an der letzten Station hängen blieben und der Produktionsfluss erheblich beeinträchtigt wurde. Die Implementierung dieses Konzepts stellt einen großen Fortschritt in Richtung einer automatisierten und kontinuierlichen Produktion dar. Durch die klare Strukturierung und Optimierung der Abläufe am Ende der Linie wird eine effiziente Prüfung, Ausschleusung und Demontage der Werkstücke ermöglicht. Die Steuerung durch die SPS gewährleistet dabei eine präzise Koordination und trägt maßgeblich zur Steigerung der Produktivität der Anlage bei.

5.7 Ausstehende Aufgaben

Anfang Dezember im Jahr 2024 wurde ein MES-Workshop realisiert, indem die genaue Funktionsweise der Anlage definiert wurde. Für das Team Vernetzung und Steuerung ergibt sich daraus ein nicht unerheblicher Workload.

Anpassungen der Programmierung müssen vorgenommen werden in den folgenden Bereichen:

- Roboter 1 muss komplett überarbeitet werden.
- Die QR-Scanner müssen implementiert werden und deren Daten an eine externe Datenbank gesendet werden.
- Die Weiterfahrbedingung des Handarbeitsplatz soll geändert werden (Der grüne Knopf soll verschwinden). Ziel: Wenn zwei QR-Codes erfasst werden, dann soll der Carrier mit dem Werkstück weiterfahren.
- Die EoL-Kontrolle muss komplett programmiert werden. Das bedeutet handeln, von leeren Carriern, die richtigen Roboter ansteuern, je nach Ergebnis der Qualitätsprüfung.
- Ausarbeiten und etablieren einer Logik, wie viele Carrier und wie die Carrier sich auf dem Förderband bewegen sollen.
- Roboter 3 muss an das Netz gebracht werden und komplett programmiert werden.
- Die SPS-Architektur muss größtenteils neu geschrieben werden, da die bisherige Programmierung nicht zu den erarbeiteten Vorstellungen passt.
- Die Daten der Kamera, die den Husky erfasst müssen ausgewertet, verarbeitet und an den Roboter 1 übermittelt werden.
- Es muss überlegt werden, wie das Datenhandling aussehen soll, um mit der Edge zu kommunizieren und Daten in eine externe Datenbank abzulegen.

Die Einschätzung der Machbarkeit aller Punkte fällt eher negativ aus, da neben der Programmierung der benannten Punkte, noch die Edge aufgesetzt werden muss, der Schaltschrank überarbeitet werden muss, sowie die Schnittstellen realisiert werden sollen.

5.8 Bisherige Programmierergebnisse

Aufgrund der bereits erörterten Problematik über fehlende Konzepte, fehlende Berater und Einarbeitung und ein sehr spätes stattfinden des Workshops konnten bisher nur kleine Teile in der Programmierung umgesetzt werden.

- Die Programmierung des Roboters 1 wurde dahingehend angepasst, dass ein Betrieb überhaupt möglich ist. Bisher kam der Carrier nur, bis Station 5 und eine Weiterfahrt war nicht möglich. Da die Qualitätsprüfung noch nicht fertig ist keine Entscheidungen, ob das Werkstück I.O. ist oder nicht, wurden auf dem HMI-Variablen erstellt, um diese Entscheidungen zu simulieren. Danach wurde ein Konzept für die EoL entwickelt, programmiert und getestet. Durch die Anpassung von Roboter 1, war es möglich 10 Teile am Stück zu produzieren. Dies konnte reproduziert werden, war aber mit einem Neustart der Anlage verbunden. Um das Ziel von 50 Teilen zu erreichen, muss Roboter 1 und 3 angepasst werden und ein neue Transportlogik, sowie EoL programmiert werden.
- Die Programmierung des Roboters 2 wurde ebenfalls angepasst. Es waren nicht alle Varianten in der Programmierung enthalten und ein Rücksprung an den Anfang des Programms bei einer bestimmten Variante war nicht vorhanden. Dies wurde dahingehend verbessert, dass die Station des „Bin-Pickings“ nun als fertig und getestet betrachtet werden kann.
- Die Variablen wurden gründlich studiert es konnte herausgefunden werden, wofür die meisten Variablen zuständig sind. Weiterhin wurden die doppelten Adressen entfernt und den richtigen Ein- und Ausgängen zugeordnet.

Da vor allem falsche angeschlossene Geräte, sowie falsch konfigurierte Schnittstellen für Probleme sorgten wurde das Hauptaugenmerk darauf gerichtet, alle Teilnehmer störungsfrei in das Netz zu implementieren.

6. Schnittstellen

Die SPS verfügt über eine Vielzahl verschiedener Sensoren, Kameras und Aktoren. Aufgrund der zahlreichen Hardwarekomponenten gibt es unterschiedliche Schnittstellen innerhalb der SPS, um auf die Eingangs- und Ausgangsdaten zuzugreifen.

Aufgrund der letzten zwei Jahre wurden kontinuierlich neue Hardwarekomponenten eingebunden, ohne dass das Vorgehen transparent dokumentiert wurde. Deshalb wird im folgenden Kapitel auf die aktuellen Schnittstellen eingegangen und deren Funktion sowie die Umsetzung näher betrachtet. Danach erfolgt eine Aufstellung der zukünftigen Schnittstellen. Unter den aktuellen Schnittstellen befinden sich auch bereits neue Komponenten, die in dieser Studienarbeit eingebunden wurden. Teilweise mussten vorherige Schnittstellen überarbeitet werden aufgrund fehlerhafter Implementierung.

Vorab werden die Begriffe PROFINET und PROFIBUS kurz erläutert.

Endlich eine Begriffsdefinition, jedoch viel zu spät!

PROFINET ist ein industrielles Ethernet Protokoll. Dieses Kommunikationsprotokoll ermöglicht es zwischen Geräten und Steuerungen eine echtzeitfähige Verbindung herzustellen. Ebenso ist eine Koexistenz von PROFINET und anderen Protokollen z.B. OPC UA im selben Netzwerk gegeben, um einen Übergang in die IT-Infrastruktur zu ermöglichen. [7]

PROFIBUS ist ein serieller Feldbus. Dieses basiert auf dem Master-Slave Prinzip mit begrenzter Teilnehmerzahl. Ebenfalls ist es für die Vernetzung von Geräten und Steuerungen verantwortlich. Im Vergleich zu PROFINET ist er schlechter bezüglich Schnelligkeit und Menge der zu übertragenden Daten. [8]

6.1 Eingebundene Schnittstellen:

6.1.1 AL1100

Der AL1100[1] ist ein IO-Link Master der Firma „ifm“. Er besitzt vier Ports für die Anbindung von Sensoren mithilfe eines 4-adrigen M12-Kabels. Der Master erfordert eine Spannungsversorgung sowie eine Verbindung über PROFINET zur SPS.

Das TIA-Projekt benötigt die „GSDML-Datei“, um eine Einbindung zu ermöglichen. In der Geräte- und Netzansicht kann der Master im Hardwarekatalog per „Drag and Drop“ hinzugefügt werden. Verweis auf Abbildung 17???

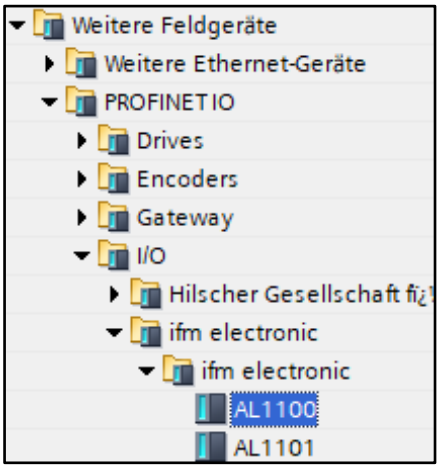


Abbildung 17: Hardwarekatalog TIA-Portal Warum jetzt ausgeschrieben und nicht als Quelle referenziert???

Die Verbindung des Masters erfolgt in der Topologieansicht durch eine intuitive Verbindung der Ports. Zu beachten ist jedoch, dass der Adressbereich aller Teilnehmer und der zu verbindenden Ports korrekt zueinander passen.

Um die Eingangsdaten der Sensoren zu erhalten, müssen die einzelnen Ports des Masters entsprechend konfiguriert werden. In der Geräte- und Netzansicht kann der Master durch einen Doppelklick aufgerufen werden. Das sich öffnende Fenster enthält die leeren Ports. Jeder Sensor besitzt eine sogenannte Schnittstellenbeschreibung, auffindbar auf der

[1] https://www.ifm.com/de/de/product/AL1100#documents

Homepage des Herstellers[2]. An dem Master AL1100 werden die Sensoren „OPD101“[3] angebunden. Hierbei handelt es sich um Profilkameras, die jeweils 160-Bit Eingang- und 8-Bit Ausgangsdaten besitzen. Deshalb wird ein 64-Byte Eingangs- und Ausgangsmodul für die Ports konfiguriert. Verweis auf Abbildung 18???

AL1100	0	0			AL1100	AL1100		
X1	0	0 X1			AL1100			
4 Ports_1	0	1			4 Ports	AL1100		
IO-Link Master	0	1 1			IO-Link Master			
IO-Link In/Out 32/32 Byt...	0	1 Port 1	453...485	240...271	IO-Link In/Out 32/3...			OPD101 -B44
IO-Link In/Out 32/32 Byt...	0	1 Port 2	486...518	272...303	IO-Link In/Out 32/3...			OPD101 -B45
	0	1 Port 3						

Abbildung 18: Portkonfiguration im TIA-Portal

Zur Auswahl stehen nur gleichgroße Eingangs- und Ausgangsmodule. Für die Datenverarbeitung ist es notwendig, die Schnittstellenbeschreibung und den konfigurierten Adressbereich zu verwenden. Bezüglich der Schnittstellenbeschreibung ist zu beachten, dass jedes Word aus zwei separaten Bytes besteht. Das linke Byte mit der niedrigeren Adresse enthält die höchstwertigen Bits und das rechte Byte mit der höheren Adresse enthält die niederwertigen Bits (Big Endian). Innerhalb des Bytes steht das höchstwertige Bit auf der höchsten Adresse. Um die Daten in einem Word darzustellen, müssen beide Bytes in der richtigen Reihenfolge zusammengefügt werden.

Verweis auf Abbildung 19???

z.B. Entfernung 16 Bit Adresse %IB1.0 und %IB2.0

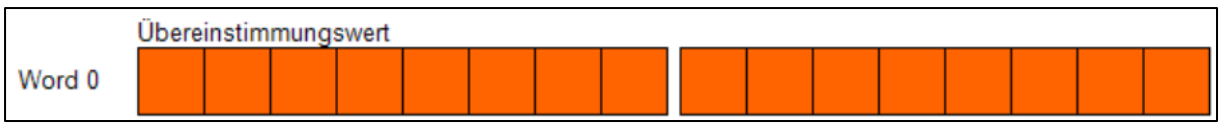


Abbildung 19: OPD101 Schnittstellenbeschreibung

Word 0 = SHL (IN:=%IB1.0 N:8) + %IB2.0

[2] https://www.ifm.com/de/product/O8T202
[3] https://www.ifm.com/de/opd101

Aktuell werden die Profilkameras nicht verwendet. Sie befinden sich an der „End of Line“ (EoL). Denkbare Einsatzmöglichkeiten wären, um zu unterscheiden, ob sich auf dem aktuellen Werkstückträger ein Werkstück befindet oder nicht. Damit könnte das Durchreichen der Werkstückträger umgesetzt werden. Ebenfalls besitzt der Master noch zwei freie Ports für weitere IO-Link Geräte.

6.1.2 AL1102

Der AL1102[4] ist ebenfalls ein IO-Link Master der Firma „ifm“. Dieser wird wie der vorherige AL1100 eingebunden. Für ein übersichtliches Kabelmanagement wird das PROFINET-Kabel durchgeschleift.

Verweis auf Abbildung 20???

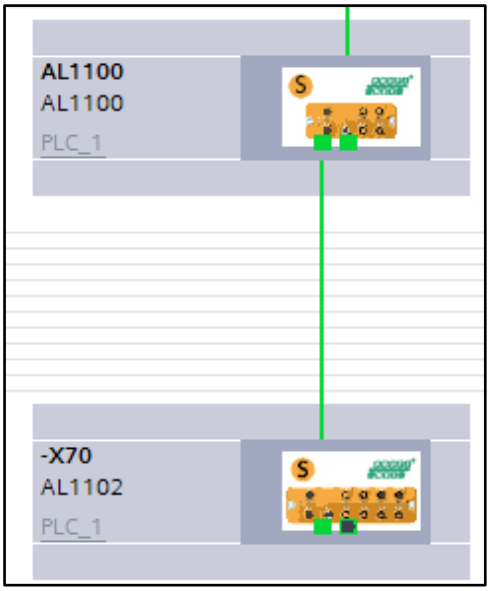


Abbildung 20: Master durchschleifen

An diesem Master werden die Auswerteelektroniken (TP3231)5 für die Temperatursensoren der vier Motoren und die zwei Lichtschranken am Handarbeitsplatz (O6P700)6 angeschlossen. Die Lichtschranke besitzt jeweils nur ein Eingangsbyte und die Auswerteelektronik jeweils zwei Eingangsbytes. Aufgrund dessen müssen die Ports konfiguriert werden.

[4] https://www.ifm.com/de/product/AL1102
[5] https://www.ifm.com/download/files/_ifm-00008C
[6] https://www.ifm.com/download/files/ifm-O6P700

-X70	0	0			AL1102	A...		
X1	0	0 X1			AL1102			
8 Ports_1	0	1			8 Ports	A...		
IO-Link Master	0	1 1			IO-Link Master			
IO-Link In 2 B...	0	1 Port 1	88....		IO-Link In 2 Byte + ...			Temperatur Motor 1
	0	1 Port 2						
	0	1 Port 3						
IO-Link In 2 B...	0	1 Port 4	91....		IO-Link In 2 Byte + ...			Temperatur Motor 4
	0	1 Port 5						
	0	1 Port 6						
IO-Link In 1 B...	0	1 Port 7	94....		IO-Link In 1 Byte + ...			Lichschranke Handarbeitsplatz
IO-Link In 1 B...	0	1 Port 8	96....		IO-Link In 1 Byte + ...			Lichschranke Handarbeitsplatz

Abbildung 10: Port Belegung AL1102

Wieso jetzt Abbildung 10????? und wo ist der Verweis auf diese Abbildung???

Die Lichtschranke wird im aktuellen SPS-Programm nicht verwendet. Sie könnte jedoch detektieren, ob eine Person am Handarbeitsplatz ein Werkstück vom Werkstückträger herunternimmt oder daraufsetzt. Sie ist voll funktionsfähig und kann ohne Probleme verwendet werden.

6.1.3 Roboter „Station 1“

Der Roboter an Station 1 ist von der Firma „Universal Robot“ und trägt die Bezeichnung „UR3“. Roboter 1 ist zuständig für das Auflegen der Werkstückträger und der Werkstücke von einem feststehenden, zweidimensionalen Magazin. Die vorherigen Studienarbeiten haben sich mit einem Magazin beschäftigt, das die Werkstücke zweidimensional vertikal lagert. Dieses befindet sich noch in der Entwicklung. Zusätzlich ermöglicht er bereits fertige Werkstücke auf den „Husky“ abzusetzen, die sich an Stopper 6 (EoL) befinden.

Der Roboter 1 ist über PROFINET[7] mit der Steuerung verbunden. Er besitzt sogenannte Register, über die Daten zwischen Roboter und SPS ausgetauscht werden. Dieses Projekt greift bisher nur auf die Input- und Output-Bit-Register zu.
Beide Register besitzen ihren eigenen Speicher. Das bedeutet, dass das Output-Register des

[7] https://www.universal-robots.com/interface-communication/profinet-how-to-guide

Roboters, das Input-Register der Steuerung ist und das Output-Register der Steuerung, das Input-Register des Roboters ist.

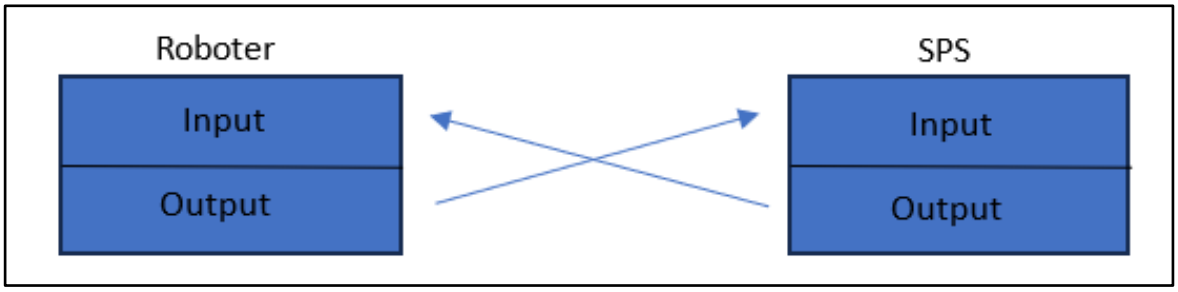


Abbildung 21: Kommunikation Roboter und SPS

Verweis auf Abbildungen???

004_output_bool_reg	"UR_9_O2T_Re...	%Q0.0	☐	☑	☑	☑
Bits	UR__O2T_bits	%Q0.0		☑	☑	☑
Register	Array[0..31] of ...	%Q0.0		☑	☑	☑
Register[0]	Bool	%Q0.0		☑	☑	☑
Register[1]	Bool	%Q0.1		☑	☑	☑
Register[2]	Bool	%Q0.2		☑	☑	☑
Register[3]	Bool	%Q0.3		☑	☑	☑
Register[4]	Bool	%Q0.4		☑	☑	☑
Register[5]	Bool	%Q0.5		☑	☑	☑
Register[6]	Bool	%Q0.6		☑	☑	☑

Abbildung 22: Output Register SPS

Das Setzen der Register löst folgende Funktionen im Roboterprogramm aus:

Register [0]	%Q0.0	True	Signalisiert dem Roboter das ein Werkstückträger vorhanden ist Roboterprogramm setzt so lange ein Werkstückträger auf bis das Register [0] gesetzt wird
Register [1]	%Q0.1	True	EoL Werkstückträger vorhanden Fertiges Werkstück auf Husky setzen
Register [2]	%Q0.2	True	Unbenutzt, Qualitätskontrolle, wenn Kamera Bauteil I.O signalisiert
Register [3]	%Q0.3	True	Unbenutzt im Roboterprogramm, sollte benutzt werden, um den Roboter zu pausieren
Register [4]	%Q0.4	True	Verrieglung, dass neues Werkstück auf den Träger gesetzt wird. Wird rückgesetzt, wenn Bin Picking fertig ist. Im Roboterprogramm wird auf False detektiert.
Register [5]	%Q0.5	True	Merker, dass ein Werkstückträger auf dem Weg zur Auflage ist, damit der Roboter keinen neuen Träger aufsetzt. Wird rückgesetzt durch Stopper 1 release und Umleger im ersten Schritt.

Tabelle 1: Output Register der SPS

Roboter sendet folgende Signale zur SPS:

Register [0]	%I0.0	True Kurzer Impuls	Roboterprogramm (Werkstück auf Träger) fertig abgelaufen. Stopper wird heruntergefahren
Register [2]	%I0.2	True Kurzer Impuls	Roboterprogramm (Werkstück auf Husky) Fertig abgelaufen. Stopper wird heruntergefahren mit Umleger verknüpft

Tabelle 2: Input Register der SPS

Register 5 wurde nachträglich hinzugefügt und im Roboterprogramm hinterlegt. Um eine aufwendige Verriegelung in der SPS zu umgehen, wurde der Informationsaustausch mit dem Roboter erweitert und die Verriegelung im Roboterprogramm hinterlegt.

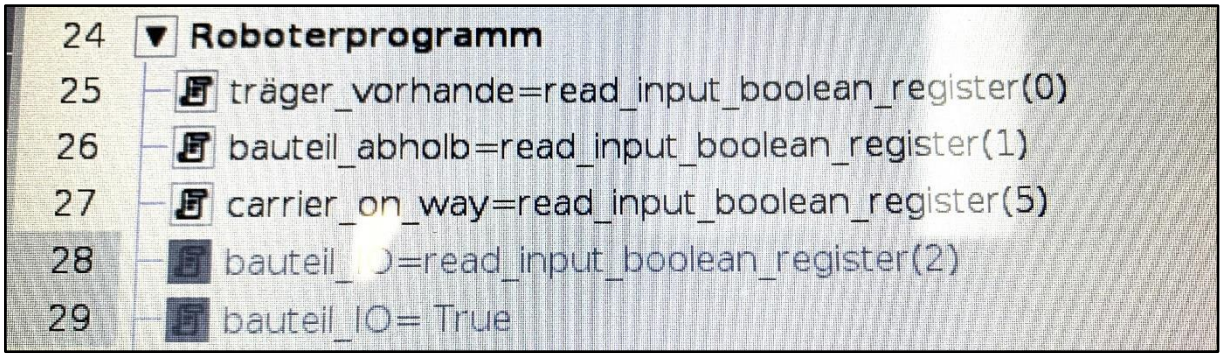


Abbildung 23: Auflage Register lesen

Verweise auf Abbildungen fehlen!!!!!

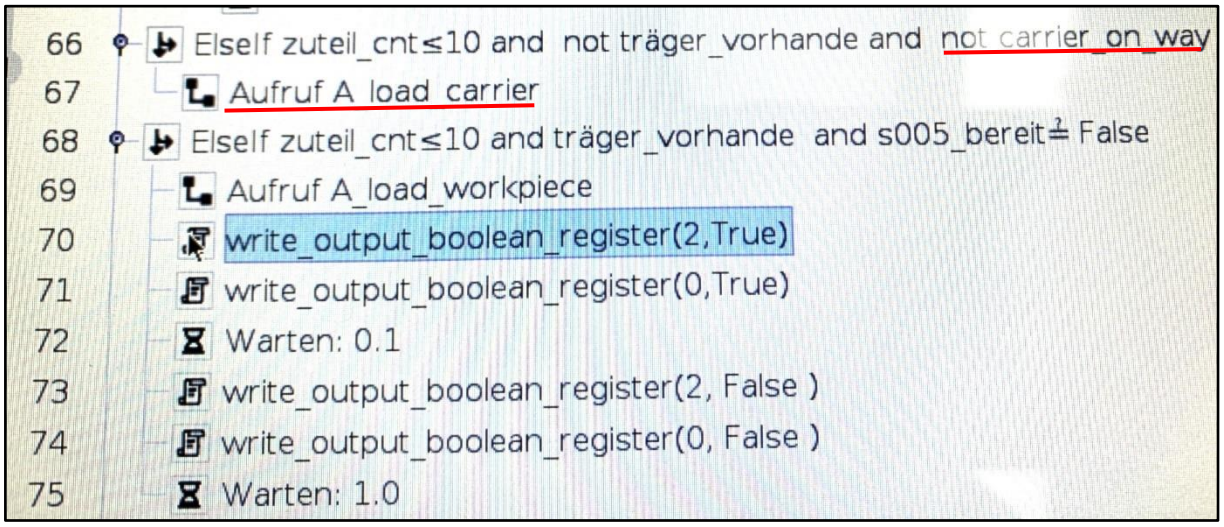


Abbildung 24: Verrieglung Auflage von neuen Trägern

Die gesamte Einbindung und Kommunikation zwischen Roboter und SPS ist der Fußnote 7 zu entnehmen. Dort wird am Ende ein Beispiel für die Verwendung in der SPS und im Roboter gezeigt.

Geeigneter wäre es hier in diesem Abschnitt darauf zu referenzieren

Ein weiteres anschauliches Tutorial befindet sich unter:

https://www.youtube.com/watch?v=F6rHbVb6Y10

https://www.youtube.com/watch?v=MhUDFoAdZNc

Wäre für den Anhang sinnhafter

Wie der Roboter zu programmieren ist, ist der folgenden Anleitung zu entnehmen:

https://Scriptmanual_5.13

6.1.4 Roboter „Station 2“

Roboter 2 ist für das „Bin Picking“ zuständig. Mikado liefert hierfür eine vorgefertigte Applikation. Diese besteht aus einer Software und einer Kamera, die die Lage der aufzunehmenden Pins detektiert.

Der Roboter kommuniziert ebenfalls über PROFINET und mehrere Register mit der SPS. Jedoch sind nur drei Adressen relevant für die Steuerung. Die SPS sendet, welche Variante gefertigt werden soll, unter der Adresse „Mikado_out_profil.productID“ und einen Startbefehl unter „Mikado_out_profil.cmd“. Die SPS erhält unter der Adresse %I424, ob der Roboter seinen Auftrag erfüllt hat.

Anführungszeichen ""

Die vorherigen Gruppen haben die fertigen Strukturen für „mikado_result“ und „mikado_status“ nicht richtig verwendet. Diese wurden mit Ausgangsadressen beschalten, jedoch sind diese ein Input-Register. Deshalb erhält die SPS unter einer einzelnen Inputadresse die Information, dass der Roboter fertig ist.

Damit in der Zukunft auf alle verfügbaren Daten des Roboters zugegriffen werden kann, werden die Register mit den korrekten Input-Adressen versehen.

Mikado_in_results	"UDT_mikadoR...	%Q174.0
Mikado_in_status	"UDT_mikadoSt...	%Q206.0

Abbildung 25: Falsche Adressierung

Verweis auf Abbildungen???

Mikado_in_results	"UDT_mikadoR...	%I394.0
Mikado_in_status	"UDT_mikadoSt...	%I389.0

Abbildung 26:Richtige Adressierung

Das erstellte Bin Picking Programm enthält einen Fehler. Das HMI bietet fünf Varianten zur Auswahl. Im Programm des „Bin Picking“ sind nur vier Varianten hinterlegt. Wenn die fünfte Variante ausgewählt wurde, geriet das Programm in eine Schleife und ein Ändern der Variante hatte keinen Einfluss mehr. Deshalb wurde eine fünfte Variante erstellt.

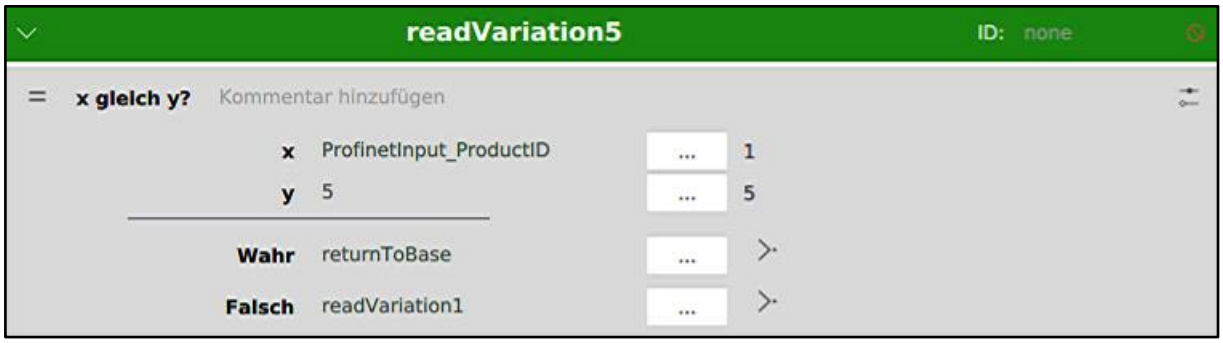


Abbildung 27: Variante 5

Mikado_out_profi	Sta...	"UDT_mikadoP...	%Q116.0		☑	☑	☑	
poseX		Real	%QD116		☑	☑	☑	X-Position im Bezugs-KOS [mm]
poseY		Real	%QD120		☑	☑	☑	Y-Position im Bezugs-KOS [mm]
poseZ		Real	%QD124		☑	☑	☑	Z-Position im Bezugs-KOS [mm]
poseR1		Real	%QD128		☑	☑	☑	Rotation R1 im Bezugs-KOS[°] (je nach Konvention)
poseR2		Real	%QD132		☑	☑	☑	Rotation R2 im Bezugs-KOS[°] (je nach Konvention)
poseR3		Real	%QD136		☑	☑	☑	Rotation R3 im Bezugs-KOS[°] (je nach Konvention)
boxID		Int	%QW140		☑	☑	☑	Gibt Index der aktiven Kiste an
cameraID		Int	%QW142		☑	☑	☑	ID der Kamera, welche verwendet werden soll
sequenceID		Int	%QW144		☑	☑	☑	ID der Sequenz, die gestartet werden soll
cmd		Int	%QW146		☑	☑	☑	Kommando-Nummer
listIndex		Int	%QW148		☑	☑	☑	Gibt Index der ausgewählten Pose an
productID		Int	%QW150		☑	☑	☑	Produkt-ID des Produkts, das verwendet werden soll

Abbildung 28: Mikado Register Output

6.1.5 VSE150

Das VSE150[8] ist eine Diagnoseelektronik für Schwingungssensoren der Firma „ifm“. Das Gerät besitzt vier analoge Anschlüsse für Sensoren. Bei Projektübernahme sind zwei Sensoren angeschlossen. Der erste Sensor ist ein Beschleunigungssensor (VSA001)[9] und der zweite ein induktiver Sensor (IFC201)[10] zur Messung der Drehzahl. Die Diagnoseelektronik verfügt über ein eigenes Programm (VSE004)[11].

Das Programm verarbeitet die Analogwerte der Sensoren nach gewünschter Parametrierung. Ebenfalls wird hier die PROFINET-Schnittstelle konfiguriert.

Der Anwender kann selbstständig auswählen, welche Ergebnisse der Auswertung über PROFINET versendet werden sollen. Über die PROFINET-Schnittstelle werden aktuell drei Variablen beschrieben. Jedoch sind die Werte, die an den Variablen ankommen, nicht sinnvoll (siehe Abbildung 29). Gegen Ende des ersten Teils der Studienarbeit wurde von der zuständigen Gruppe das VSE150 neu parametriert und alle Sensoren eingebunden. Jedoch gibt es aktuell noch Probleme mit der Konfiguration der PROFINET-Schnittstelle im VSE150. Die Problembehebung wird auf den zweiten Teil der Studienarbeit gelegt.

003_M03_Temperatur	Real	%ID329	☐	☑	☐	☑	-98.74178
003_M03_Drehzahl	Real	%ID333	☐	☑	☐	☑	0.0
003_M03_Schwingung	Real	%ID337	☐	☑	☐	☑	1.154103E-05

Abbildung 29: VSE150 falsche Werte

Ein anschauliches Tutorial für die Konfiguration der PROFINET-Schnittstelle, sowie die Einbindung in TIA befindet sich unter:

PROFINET-Setup: https://www.youtube.com/watch?v=BLULasY2KOQ

Part 1: https://www.youtube.com/watch?v=v-tg6978T9o

Part 2: https://www.youtube.com/watch?v=hV11Q3WtA2c

[8] https://www.ifm.com/de/de/product/VSE150
[9] https://www.ifm.com/de/de/product/VSA001
[10] https://www.ifm.com/de/de/product/IFC201
[11] https://www.ifm.com/de/de/product/VES004

6.1.6 Switch

Aufgrund der Vielzahl von neuen PROFINET-Geräten musste ein dritter SPS-Switch in Betrieb genommen werden. Dieser benötigt eine eigene 24V Spannungsversorgung und muss an einen bereits eingebunden Switch mit einem LAN-Kabel verbunden werden. Die aktuelle Belegung der Switches ist den nachfolgenden Tabellen zu entnehmen:[3,4,5]

Switch: -K5	Port	Gerät
	P1	PAC3220
	P2	Condition Monitoring
	P3	Roboter 1
	P4	SPS
	P5	EoL Station
	P6	VSE150
	P7	K7 HUB-GM100
	P8	Switch: -K6

Tabelle 3: Belegung K5

Switch: -K6	Port	Gerät
	P1	Peak Board
	P2	B4
	P3	Mikado
	P4	K5
	P5	K9
	P6	B6
	P7	AL1100
	P8	B1 (QR)

Tabelle 4: Belegung K6

Switch: -K9	Port:	Gerät:
	P1	
	P2	
	P3	K6
	P4	wenglor Kamera
	P5	
	P6	
	P7	
	P8	ifm Kamera

Tabelle 5: Belegung K9

6.1.7 wenglor Kamera

Die Kamera (B60P203)[12] soll bei der „EoL“ im Demontagezyklus eine Aussage treffen, ob die demontierten Teile weiterhin verwendet werden können und ein Bestandteil der Kreislaufwirtschaft bleiben. Die Einbindung der Kamera erfolgt über PROFINET, sowie der „GSDML-Datei“. Gegen Ende des ersten Teils der Studienarbeit wurde zum ersten Mal der gesamte Prozess näher beleuchtet und ausgelegt. Die Kamera wird mit hoher Wahrscheinlichkeit, bei der Überprüfung der montierten Werkstücke ebenfalls eingesetzt.

Im zweiten Teil der Studienarbeit müssen vermutlich mehrere „Jobs“ angelegt werden, sowie aufgrund der Eingangsdaten der erste und dritte Roboter angesteuert werden. „Jobs“ bedeutet in diesem Kontext, dass die Kamera mehrere Prüfprogramme besitzt und die SPS der Kamera mitteilt, welches Programm für das vorhandene Werkstück eingesetzt werden muss.

Die Einbindung der Kamera lief ohne weitere Probleme und ist voll funktionsfähig.

[12] https://www.wenglor.com/de/B60P203

6.1.8 Lichtschranken/Endlagenschalter

Ebenfalls besitzt die Anlage mehrere Einweglichtschranken der Firma „sick“ (WT2S-P211)[13]. Diese sind alle an einer Eingangskarte der SPS verbunden und geben somit ihre Bit Signale an die SPS weiter.

Die Endlagenschalter befinden sich an den pneumatischen Zylindern der Umleger. Diese sind auch an einer Eingangskarte der SPS verbunden.

6.1.9 Ventilinsel

Die Ventilinsel (CPV10-VI)[14] ist zuständig für die Betätigung der Stopper und Zylinder. Die Verbindung zwischen SPS und Ventilinsel erfolgt mit PROFIBUS.

6.1.10 QR-Scanner

An jeder Station befindet sich ein QR-Scanner (O2I5xx) der Firma „ifm“. Aktuell ist nur einer angebunden, aber funktionsfähig. Bei der Anbindung der anderen kann sich dort orientiert werden. Die Komponenten des Werkstückes sollen in Zukunft eindeutig identifiziert werden, um zum Beispiel herauszufinden welche Bauteile bei der Qualitätskontrolle entsorgt werden.

6.2 Zukünftige Schnittstellen:

6.2.1 ifm Kamera

Eine Kamera (O3D304)[15] der Firma „ifm“ soll die Lagererkennung des „Huskys“ und der Werkstücke übernehmen. Die Koordinaten werden von der SPS aufbereitet und an den ersten Roboter weitergeleitet. Durch einen flexiblen Referenzpunkt kann das Roboterprogramm die Lagetoleranz des „Huskys“ beim Anfahren der Lade- und Entladezone ausgleichen.

[13] https://www.sick.com/de/de/catalog/archiv/wt2s-p211/p/p207166
[14] https://www.festo.com/de/de/a/18200/
[15] https://www.ifm.com/de/de/product/O3D304

Die Kamera wird mit PROFINET angebunden. Aktuell besteht noch keine Verbindung zur SPS und es findet kein Datenaustausch statt. Dies wird im zweiten Teil der Studienarbeit behandelt.

6.2.2 Roboter „Station 5“

Wie bereits auf der vorherigen Seite erwähnt, wurde der Prozess das erste Mal definiert. Der gesamte Prozess benötigt an der Qualitätskontrolle einen weiteren Roboter. Dieser soll für mehrere Dinge zuständig sein. Einerseits muss dieser demontierte Werkstückteile aufgrund der Qualität sortieren und nicht korrekt montierte Werkstücke an den Handarbeitsplatz zurückliefern.

Es befindet sich bereits ein ungenutzter Roboter desselben Typs wie an „Station 1“ an „Station 5“. Bei der Einbindung an die SPS kann sich an „Station 1“ orientiert werden. Für die Umsetzung des Prozesses müssen beide Roboter und die Qualitätskontrolle sehr eng zusammenarbeiten. Das bedeutet, die SPS muss notwendige Informationen zwischen beiden Teilprozessen austauschen, um ein fehlerfreies Zusammenspiel zu gewährleisten.

Wichtig für den zweiten Teil der Studienarbeit ist es, dass die Roboter mit der SPS einen hohen Informationsaustausch haben, um komplizierte bzw. aufwendige Programmierungen zu vermeiden.

6.2.3 MES

MES oder auch „**M**anufacturing **E**xecution **S**ystem“ genannt bzw. wird auch als Fertigungssteuerungssystem oder Produktionsleitsystem betitelt. Seine Funktion besteht darin einen Prozess zu überwachen, verfolgen und zu dokumentieren. Quelle?

Es ermöglicht den verantwortlichen Personen aufzeigen, an welchen Stellen der Prozess optimiert werden sollte. [9]

Dieses System von „FORCAM“ soll mit der Anlage im PTZ verknüpft werden, um alle Informationen der Anlage zu bündeln und ein Auftragsmanagement zu etablieren. Hierfür muss die SPS mit dem System verbunden werden und alle relevanten Daten übertragen werden. Die Erstellung der Schnittstelle zum MES erfordert eine eindeutige und klare

Strukturierung der Anlage, welche Daten bereits zur Verfügung stehen bzw. welche erforderlich sind. Durch die Schulung von „FORCAM“ wurde der gesamte Prozess analysiert und ausgelegt. Die Umsetzung erfolgt im zweiten Teil der Studienarbeit.

7. Fazit

Die Studienarbeit hat gezeigt, dass die Optimierung der SPS-Architektur und die Integration moderner Technologien entscheidend für die Effizienz und Flexibilität einer Produktionslinie sind. Durch die gezielte Analyse und Bearbeitung von Schwachstellen, wie der unstrukturierten Programmierung, fehlenden Schnittstellen und mangelnder Dokumentation, konnten erste Erfolge erzielt werden. Die Neustrukturierung des Programmcodes und die Einbindung von Bildverarbeitungsdaten legten eine Basis für die Automatisierung und Erweiterbarkeit der Anlage. Insbesondere die Entwicklung eines funktionierenden Konzepts für den End-of-Line-Prozess sowie die Integration von Robotik in die Steuerungsarchitektur verdeutlichen die Fortschritte.

Dennoch bestehen weiterhin Herausforderungen, darunter die vollständige Programmierung von Roboter 1 und 3, die Verbesserung der Kommunikationslogik zwischen Komponenten und die Integration des MES. Diese Punkte bedürfen weiterer Aufmerksamkeit, um das volle Potenzial der Produktionslinie auszuschöpfen. Die wechselnde Teamzusammensetzung und unzureichende Einarbeitungszeit haben gezeigt, wie wichtig eine klare Dokumentation und strukturierte Übergabe sind, um eine nachhaltige Weiterentwicklung zu gewährleisten.

Die Ergebnisse unterstreichen, dass datengetriebene Automatisierungslösungen, kombiniert mit einer klaren Systemarchitektur, nicht nur die Produktivität, sondern auch die Skalierbarkeit und Anpassungsfähigkeit moderner Produktionssysteme erheblich steigern können. Die Arbeit hat eine Grundlage geschaffen, auf der zukünftige Teams aufbauen können, und zeigt das Potenzial von Industrie-4.0-Technologien zur nachhaltigen Transformation industrieller Prozesse.

7. Literaturverzeichnis

[1] n.a., „PowerPoint Präsentation SPS, MES und Data Connection," n.a., Friedrichshafen, n.a..

[2] Siemens AG, „SiePortal," 27. 02. 2024. [Online]. Available: https://support.industry.siemens.com/cs/document/109744319/change-ip-address-and-profinet-device-name-during-operation?dti=0&lc=en-DE.

[3] ifm electronic gmbh, „ifm Produkte," [Online]. Available: https://www.ifm.com/de/de/product/EVC431. [Zugriff am 28. 12. 2024].

[4] A. Hanser, „SPS," Friedrichshafen, 2024.

[5] D. Weberbauer, *Auszug aus dem TIA Portal von Siemens,* Friedrichshafen, 2024.

[6] D. Weberbauer, *Auszug aus Programmierung eines UR.*

[7] e.V., PROFIBUS Nutzerorganisation, „PROFINET - Technology Descriptiom," [Online]. Available: https://www.profinet.com/profinet-explained/technology-description. [Zugriff am 12. 1. 2025].

[8] DriveCon GmbH, „DriveCon Lexikon - Profibus," [Online]. Available: https://www.drivecon.de/de/lexikon/profibus. [Zugriff am 12. 1. 2025].

[9] SAP, „SAP - Was ist ein MES?," [Online]. Available: https://www.sap.com/germany/products/scm/execution-mes/what-is-mes.html#:~:text=Deutschland-,Was%20ist%20ein%20MES%20(Manufacturing%20Execution%20System)%3F,Rohmaterial%20bis%20zum%20fertigen%20Produkt. [Zugriff am 12. 1. 2025].

Für die sprachliche und stilistische Überprüfung der Arbeit wurde das KI-Tool ChatGPT (GPT-4, OpenAI, 2024) verwendet.

9. Abbildungsverzeichnis

Abbildung 1:SPS, MES und Data Connection [1] 1
Abbildung 2: Projektablauf T3100 4
Abbildung 3: Topologieansicht SPS 5
Abbildung 4: Topologieansicht SPS 7
Abbildung 5: Kommunikationswege von Datenbausteinen [4] 12
Abbildung 6: Topologie-Ansicht aus dem TIA Portal von Siemens [5] 13
Abbildung 7: Programmhierarchie [5] 14
Abbildung 8: Netzwerk 2 aus dem Ausgangsprogramm [5] 17
Abbildung 9: Wiederverwendbare Stopperfunktion [5] 18
Abbildung 10: Messbaustein 1 von 10 [5] 20
Abbildung 11: Auszug aus Variablentabelle [5] 21
Abbildung 12: Auszug Roboterprogrammierung von Roboter 1 [6] 21
Abbildung 13: Auszug HMI-Bilder [5] 23
Abbildung 14: Auszug HMI-Bilder; Variantenauswahl [5] 23
Abbildung 15: Programmierfehler [5] 24
Abbildung 17: Konzept "end of line" 25
Abbildung 18: Hardwarekatalog TIA-Portal 30
Abbildung 19: Portkonfiguration im TIA-Portal 31
Abbildung 20: OPD101 Schnittstellenbeschreibung 31
Abbildung 21: Master durchschleifen 32
Abbildung 22: Kommunikation Roboter und SPS 34
Abbildung 23: Output Register SPS 34
Abbildung 24: Auflage Register lesen 36
Abbildung 25: Verrieglung Auflage von neuen Trägern 36
Abbildung 26: Falsche Adressierung 37
Abbildung 27:Richtige Adressierung 37
Abbildung 28: Variante 5 38
Abbildung 29: Mikado Register Output 38
Abbildung 30: VSE150 falsche Werte 39