DHBW
Duale Hochschule
Baden-Württemberg
Ravensburg
Campus Friedrichshafen

Automatisiertes Modell-Hochregallager

T3_3100

Elektrotechnik

Automation

Duale Hochschule Baden-Württemberg Ravensburg, Campus Friedrichshafen

von

Maximilian Zorko und Willi Schaal

Abgabedatum:	
Bearbeitungszeitraum:	16.10.2025 - 08.01.2026
Matrikelnummer:	3960407/5732737
Kurs:	TEA 23
Betreuerin / Betreuer:	Prof. Dr. Ing. Thorsten Kever

Erklärung

gemäß Ziffer 1.1.14 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 24.07.2023.
Wir versichern hiermit, dass wir unsere Studienarbeit mit dem Thema:

Automatisiertes Modell-Hochregallager

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

Friedrichshafen, den 8. Januar 2026

_____

Maximilian Zorko und Willi Schaal

Zusammenfassung

Diese Studienarbeit befasst sich mit der Weiterentwicklung eines automatisierten Modell-Hochregallagers an der Dualen Hochschule Baden-Württemberg. Ziel ist es, bestehende Funktionen um sichere und strukturierte Erweiterungen zu ergänzen und das System stärker an industrielle Anforderungen anzupassen. Der Fokus liegt auf der Integration einer sicherheitsgerichteten Lichtschranke, der Umsetzung eines internen Umlagerungsprozesses sowie der Optimierung der SPS-Programmstruktur durch den Einsatz der Programmiersprache Structured Text (ST). Die Lichtschranke wird über IO-Link an eine Siemens S7-1500 angebunden und mithilfe geeigneter Prozesswerte in die Sicherheitslogik integriert. Durch eine gezielte Auswertung des Prozesswertes „Number of Beams Occupied“ kann der Gefahrenbereich zuverlässig überwacht und bei unzulässigen Eingriffen eine automatische Abschaltung ausgelöst werden. Die Sicherheitsfunktion ist vollständig in die bestehende Steuerungsarchitektur eingebettet und mit dem Störungsmanagement gekoppelt. Ergänzend wird ein Umlagerungsprozess realisiert, der die interne Verlagerung von Lagergütern ermöglicht. Eine eigenständige Schrittkette stellt einen sicheren und flexiblen Ablauf des Prozesses sicher und berücksichtigt sowohl die Bewegungslogik der Achsen als auch das Lagerplatzmanagement. Insgesamt wird das Hochregallagersystem funktional, sicherheitstechnisch und strukturell erweitert und bildet eine belastbare Grundlage für zukünftige, auftragsbasierte Erweiterungen.

Abstract

This study focuses on the further development of an automated model high-bay warehouse at the Baden-Wuerttemberg Cooperative State University. The objective is to enhance existing functionalities by implementing safety-related and structural improvements and to align the system more closely with industrial automation requirements. Key aspects of the work include the integration of a safety light curtain, the implementation of an internal relocation process, and the optimization of the PLC program structure using the programming language Structured Text (ST). The safety light curtain is connected to a Siemens S7-1500 PLC via IO-Link and integrated into the control system using selected process values. By evaluating the “Number of Beams Occupied” parameter, the hazardous area can be reliably monitored and hazardous movements can be automatically stopped in the event of unauthorized access. The safety logic is fully embedded into the existing control architecture and linked to the fault management system. In addition, a relocation process is implemented to enable the internal transfer of load units between different storage locations. A separate sequential control structure ensures a safe and transparent process flow while taking into account axis movements and storage location management. Overall, the high-bay warehouse system is significantly improved in terms of functionality, safety, and software structure, providing a robust foundation for future extensions such as order-based material flow control.

Inhaltsverzeichnis

1 Einleitung

1.1 Motivation

In Zeiten von **Industrie 4.0** hat die Bedeutung des Automatisierungsgrades in den vergangenen Jahren erheblich zugenommen.
Besonders im Hinblick auf Effizienz, Sicherheit und die Effektivität automatisierter Abläufe spielt die Automatisierung eine zentrale Rolle.
Der Begriff **Industrie 4.0** beschreibt den aktuellen Wandel in der industriellen Produktion hin zu intelligent vernetzten Systemen. Dabei spielt die Automatisierung eine entscheidende Rolle. Die Kommunikation der einzelnen Maschinen, Produktionsanlagen und Lager erfolgt hierbei über **industrielle Netzwerke**, was zu einem echtzeitfähigen Verhalten beitragen kann.
Durch die zunehmende Digitalisierung und Vernetzung industrieller Prozesse entstehen stetig neue Anforderungen an die Steuerungs- und Regelungstechnik. Diese Entwicklungen führen zu immer komplexeren, aber auch effizienteren Produktions- und Logistiksystemen.

Gerade im Bereich der Lagerlogistik ist es entscheidend, sämtliche Sicherheitsmaßnahmen einzuhalten, um einen reibungslosen und sicheren Ablauf zu gewährleisten. Gleichzeitig wird eine präzise und fehlerarme Handhabung von Materialien ermöglicht, was wesentlich zur Prozesssicherheit und Produktqualität beiträgt. Lagerprozesse sind ein wesentlicher Bestandteil des Materialflusses innerhalb der Produktion. Fehler in diesem Bereich können zu Stillständen und hohen Kosten führen. Um solche Probleme zu vermeiden, ist ein reibungsloser Ablauf aller Prozessschritte erforderlich.

1.2 Bedeutung der Lagerlogistik

Im Bereich der Automatisierungstechnik existieren zahlreiche Konzepte, um Produkte und Ersatzteile effizient zu lagern und zu verwalten.
Eine klassische Variante stellt die manuelle Lagerung in großen Lagerräumen dar. Dabei sind jedoch umfangreiche Such- und Zuordnungsprozesse erforderlich, um die gewünschten Komponenten zu identifizieren und zu entnehmen.
Eine wesentlich effektivere Lösung bietet die automatisierte Lagertechnik, insbesondere das sogenannte „Hochregallager“.
Ein Hochregallager ermöglicht die automatisierte Ein- und Auslagerung von Materialien und Ersatzteilen. Solche Systeme ermöglichen es durch den Einsatz moderner Sensorik, wie etwa der Materialerkennung über **RFID** oder der automatischen Auftragsausführung mittels QR-Code-Erkennung, Materialien präzise zu lagern und zu verwalten. Des Wei-

teren können die Prozesse über geeignete Softwarelösungen überwacht und gesteuert werden.
Dadurch können Prozesszeiten verkürzt, Fehlerquoten reduziert und die Übersichtlichkeit innerhalb des Lagers deutlich verbessert werden.

Grundsätzlich lässt sich die Handhabung in zwei Varianten unterteilen:

- Automatische Ein- und Auslagerung
- Manuelle Ein- und Auslagerung

Das Hochregallager ist in der Regel in einem separaten Bereich installiert, in dem autorisierte Mitarbeiter mithilfe eines Auftrags- oder Identifikationssystems auf die eingelagerten Komponenten zugreifen können.
Die Automatisierung solcher Systeme ermöglicht eine zuverlässige, reproduzierbare und sichere Abwicklung logistischer Prozesse. Hierdurch werden menschliche Fehler weitestgehend reduziert und somit die Sicherheit erhöht.

1.3 Das DHBW-Hochregallager-Modell

Zur Simulation und Demonstration der grundlegenden Lagerfunktionen wurde an der Dualen Hochschule Baden-Württemberg (DHBW) eine Miniaturversion eines Hochregallagers entwickelt, die die wesentlichen Prozesse der Ein- und Auslagerung abbildet. Das Modell eignet sich hervorragend als Lern- und Übungsplattform für Studierende, um praxisnah Kenntnisse in der industriellen Automatisierung zu erwerben. Ziel des Systems ist es, die Zusammenarbeit der einzelnen Sensoren und Aktoren zu simulieren und dadurch den Studierenden einen praxisnahen Einblick in die industrielle Automatisierung zu bieten.

Das System wurde in den vergangenen Jahren bereits mehrfach für Studien- und Projektarbeiten im Studiengang „Elektrotechnik – Automation“ eingesetzt und kontinuierlich weiterentwickelt. Diese Arbeiten ermöglichten es den Studierenden, ein vertieftes Verständnis für Steuerungs- und Automatisierungssysteme zu erlangen und praxisorientierte Erfahrungen zu sammeln.

1.4 Zielsetzung der Studienarbeit

Trotz der bisherigen Entwicklungen besteht weiterhin Verbesserungspotenzial, insbesondere im Bereich der Sicherheit und der Programmstruktur.
Die vorliegende Studienarbeit befasst sich daher mit der Überarbeitung und Einführung neuer Sicherheitsmaßnahmen. Darüber hinaus soll eine auftragsbasierte Material- ein- und -auslagerung implementiert werden. Weitere Schwerpunkte bilden die vollständige Implementierung der **RFID**-basierten Materialerkennung sowie die Umstellung des bestehenden SPS-Programms auf die Programmiersprache **Structured Text (ST)**.

Aufgrund des hohen Umfangs und der technischen Komplexität des Projekts erfolgt die vollständige Umsetzung in den **Theoriephasen 5 und 6**.
Ziel der aktuellen Bearbeitung (Theoriephase 5) ist die Einführung neuer Sicherheitskonzepte und eines neuen Umlagerung- Prozesses.

Abschließend soll die durchgeführte Arbeit die Grundlage für die vollständige Realisierung eines funktionsfähigen, sicheren und didaktisch wertvollen Hochregallagermodells bilden, das zukünftigen Studierenden als praxisnahes Lehr- und Forschungsobjekt dient.
Die in den folgenden Kapiteln erläuterten Schritte zur Planung und Umsetzung sollen verdeutlichen, wie die Studienarbeit (Teil 1) zur Weiterentwicklung der Funktionalität und der Sicherheit des Systems beiträgt. Nun erfolgt eine Erläuterung der Problemstellung und des aktuellen Standes der Technik.

2 Grundlagen

2.1 Problemstellung

Durch zahlreiche vorangegangene Studienarbeiten wurden bereits viele grundlegende Funktionen des Hochregallagers implementiert.
Hierzu gehören unter anderem:

- automatisiertes Einlagern,
- manuelles Auslagern,
- sowie die **RFID-Materialerkennung**.

Diese Funktionen bleiben im Rahmen dieser Arbeit bestehen und sind nicht Hauptbestandteil der weiteren Bearbeitung.
Der Fokus der vorliegenden Studienarbeit liegt vielmehr auf der Integration zusätzlicher **Sicherheitskomponenten**, der Fertigstellung der **RFID-Materialerkennung** sowie der **Überarbeitung bzw. Umstellung des SPS-Programms** auf die Programmiersprache **Structured Text (ST)**.

Für den sicheren Betrieb eines Hochregallagers sind verschiedene Schutzmechanismen erforderlich, um Gefahren beim Eingriff in den laufenden Prozess zu vermeiden. Einige dieser Maßnahmen wurden bereits in früheren Arbeiten umgesetzt.
Ein wesentliches, bislang fehlendes Sicherheitselement ist jedoch eine Lichtschranke, die sich unmittelbar vor dem Lagerbereich befindet. Diese soll im Rahmen dieser Arbeit in das Steuerungsprogramm integriert werden, um potenzielle Gefährdungen bei manuellen Eingriffen während des Betriebs zu verhindern.

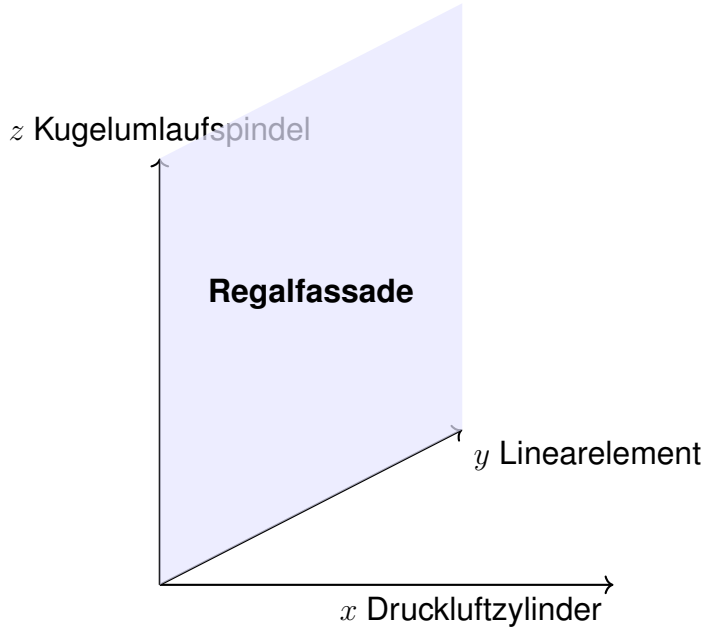
Darüber hinaus ist zur Gewährleistung eines fehlerfreien Ablaufs während der Materialerkennung die Funktionalität der RFID-Prüfung zu analysieren, zu optimieren und gegebenenfalls zu vervollständigen.
Ein weiterer wichtiger Aspekt betrifft die verwendete Programmiersprache des SPS-Systems. In den bisherigen Studienarbeiten wurde überwiegend die grafische Programmiersprache **FBS (Funktionsbausteinsprache)** eingesetzt. Diese bietet den Vorteil einer einfachen, blockorientierten Programmstruktur, stößt jedoch bei komplexeren Anwendungen schnell an ihre Grenzen.
Aus Gründen der Übersichtlichkeit, Wartbarkeit und Fehlersuche soll das bestehende Programm daher in die textbasierte Programmiersprache **ST (Structured Text)** übertragen werden.

Abschließend lässt sich festhalten, dass die wesentlichen Funktionen des bestehenden Hochregallagers bereits implementiert sind. Trotzdem bestehen noch Optimierungspotenziale im Hinblick auf Sicherheit, Struktur und Programmiermethodik. Um die geplanten Maßnahmen gezielt umsetzen zu können, ist es erforderlich, den aktuellen technischen Aufbau und die bestehende Programmstruktur zunächst detailliert zu analysieren. Im folgenden Abschnitt „Stand der Technik“ werden daher die vorhandenen Komponenten und deren Funktion genauer betrachtet, um eine fundierte Grundlage für die anschließende Überarbeitung zu schaffen.

2.2 Stand der Technik

2.2.1 Aufbau des Hochregallagers

Bei der zu optimierenden Anlage handelt es sich um ein **Modell-Hochregallager** im Laborraum H001 der **Dualen Hochschule Baden-Württemberg** am Campus Friedrichshafen. Diese dient dazu, den Studierenden bei vertretbaren Anschaffungskosten ein Modell einer **modernen Hochregalanlage**, wie sie auch in zahlreichen **Industriebetrieben** zu finden ist, zu bieten.
Um den Aufbau der Anlage fachgerecht erklären zu können, ist die Definition eines **Koordinatensystems** für die Bewegung der **Einlagerungsvorrichtung** notwendig:



- x-Achse -> Bewegung des Trägers in das Hochregal
- y-Achse -> Bewegung des Trägers in Horizontale entlang der Regalstruktur
- z-Achse -> Bewegung des Trägers in Vertikale entlang der Regalstruktur

Die **Förder- und Bewegungssysteme** der Modell-Hochregalanlage stellen das zentrale Element zur Realisierung **automatisierter Lagerprozesse** dar. Die Anlage ist so konzipiert, dass sie die **Ein- und Auslagerung** von Lagergütern in einem mehrstöckigen Regalsystem ermöglicht. Die Bewegungsachsen werden in drei Dimensionen unterteilt: **horizontal**, **vertikal** und die eigentliche **Einlagerbewegung**.

Die **horizontale Verfahrbewegung** (y-Achse) des Regalbediengeräts erfolgt über ein **Linearelement**, das entlang einer **Führungsschiene** verläuft. Das Linearsystem basiert auf einem **motorgetriebenen Schlitten**, welcher präzise Positionierungen entlang der Regalfassade ermöglicht. Die Führungsschiene gewährleistet eine stabile und reibungsarme Bewegung, während der Antrieb über einen **Schrittmotor** erfolgt, der über eine **Steuerungseinheit** angesteuert wird. Die Positionserfassung wird gegenwärtig durch **Timer im Steuerungsprogramm** realisiert. Die Positionierung der **Endschalter** erfolgt ausschließlich am Ende der Achsen.

Die **vertikale Bewegung** (z-Achse) wird durch eine **Kugelumlaufspindel** realisiert, die eine hohe Positioniergenauigkeit und mechanische Effizienz bietet. Die Spindel ist mit einem **rotatorischen Antrieb** gekoppelt, der die Drehbewegung in eine **lineare Hubbewegung** umsetzt. Die Verwendung von **Kugelumlaufmuttern** führt zu einer Minimierung der Reibung und einer Erhöhung der Tragfähigkeit, was insbesondere bei mehrstöckigen Regalsystemen vorteilhaft ist. Die vertikale Führung erfolgt über ein stabiles Profil, das die Bewegung des Hubsystems stabilisiert und ein Verkanten verhindert.

Die eigentliche **Einlagerung des Lagerguts** (x-Achse) erfolgt mittels eines **Druckluftzylinders**. Der Zylinder ist am Regalbediengerät montiert und fährt bei Erreichen der Zielposition aus, um das Lagergut in das vorgesehene Fach zu überführen. Die Wahl eines pneumatischen Systems ermöglicht eine schnelle und kraftvolle Bewegung, die unabhängig von der elektrischen Steuerung arbeitet und sich gut für wiederholte, gleichförmige Bewegungsabläufe eignet. Die Steuerung des Zylinders erfolgt durch ein **Magnetventil**, das durch die SPS angesteuert wird.
Die **Sensorik und Steuerungseinheiten** der Modell-Hochregalanlage übernehmen zentrale Aufgaben zur Gewährleistung eines sicheren und präzisen Betriebs. Sie ermöglichen sowohl die **Positionsbestimmung** der beweglichen Komponenten als auch die **Identifikation der Lagergüter** und die **Prozesssteuerung**.

Zur Erfassung der Endstellungen der horizontalen und vertikalen Bewegungsachsen werden mechanische **Endschalter** verwendet. Diese sind jeweils an den Begrenzungspunkten der Lineareinheit sowie der Kugelumlaufspindel angebracht und dienen der sicheren Detektion der maximalen Verfahrwege. Das Erreichen einer Endlage löst

den entsprechenden Schalter aus und führt zur Übertragung eines Signals an die Steuerungseinheit. Diese Vorgehensweise verhindert ein Überfahren der mechanischen Grenzen und gewährleistet den Schutz der Anlage vor potenziellen Schäden.

Die **Materialerfassung** erfolgt mittels eines **RFID-Systems**, welches sich aus einer **RFID-Antenne** und entsprechenden **Transpondern** zusammensetzt. Jedes Element des Lagers ist mit einem **RFID-Tag** ausgestattet, der eine eindeutige Identifikation ermöglicht. Die Positionierung der Antenne erlaubt es, beim Ein- oder Auslagern eines Artikels dessen Transponder auszulesen. Die erfassten Daten werden unmittelbar an die Steuerung übermittelt und dort verarbeitet. Das berührungslose Identifikationsverfahren ermöglicht eine schnelle und fehlerfreie Zuordnung der Lagergüter.

Die zentrale Steuerung der Anlage erfolgt mittels einer **SPS vom Typ Siemens S7-1511**, welche sämtliche Bewegungsabläufe, Sensorrückmeldungen und Aktorsteuerungen koordiniert. Die SPS ist über ein **HMI (Human-Machine Interface)** mit dem Bedienpersonal verbunden, wodurch eine intuitive Bedienung und Visualisierung der Anlagenzustände ermöglicht wird. Über das HMI können Betriebsmodi gewählt, Lagerprozesse gestartet und Diagnosedaten eingesehen werden.

2.2.2 Speicherprogrammierbare Steuerungen

Die **Speicherprogrammierbare Steuerung** (SPS) stellt ein zentrales Element moderner industrieller Automatisierung dar. Sie ermöglicht die flexible, softwarebasierte Steuerung technischer Prozesse und ersetzt zunehmend klassische **verbindungsprogrammierte Steuerungen** wie Relais- und Schütztechnik. Insbesondere die SPS-Systeme der Siemens AG, unter dem Markennamen SIMATIC bekannt, nehmen eine führende Rolle in der industriellen Praxis ein.
Im folgenden eine Gegenüberstellung von VPS und SPS:

Kriterium	**SPS (Speicherprogrammierbare Steuerung)**	**VPS (Verbindungsprogrammierte Steuerung)**
Flexibilität	Hohe Flexibilität durch einfache Softwareänderung	Geringe Flexibilität, Änderungen erfordern Hardwareeingriffe
Programmierung	Softwarebasiert über Programmiersprachen (z. B. KOP, FUP, SCL)	Hardwarebasiert durch Verdrahtung von Relais und Schützen
Fehlersuche	Komfortable Diagnosefunktionen über Softwaretools	Fehlersuche oft manuell und zeitaufwendig
Platzbedarf	Kompakte Bauweise, platzsparend	Großer Platzbedarf durch viele Einzelkomponenten
Kosten bei Änderungen	Geringe Kosten, da keine Hardwareänderung nötig	Hohe Kosten durch Umbau und neue Komponenten
Wartung	Zentralisiert und softwaregestützt	Dezentral und hardwareintensiv
Komplexität der Steuerung	Geeignet für komplexe und vernetzte Systeme	Nur für einfache Steuerungsaufgaben geeignet
Zukunftsfähigkeit	Industrie 4.0-fähig, IoT-Integration möglich	Veraltet, kaum kompatibel mit modernen Systemen

Tabelle 1: Vergleich zwischen SPS und VPS hinsichtlich technischer und wirtschaftlicher Kriterien; Quelle: [21]

Die SIMATIC-Reihe von Siemens ist in zwei Produktlinien gegliedert, welche sich hauptsächlich in ihren Anwendungsbereichen unterscheiden. Die S7-1200 ist für die Realisierung kompakter Automatisierungslösungen konzipiert, wohingegen die S7-1500 erweiterte Funktionen für die Bewältigung komplexer Steuerungsaufgaben in vernetzten

Produktionsumgebungen bietet. Die Funktionsweise beider Systeme ist durch das EVA-Prinzip (Eingabe – Verarbeitung – Ausgabe) determiniert. Die Bereitstellung der Eingangssignale erfolgt durch Sensoren, deren Signale durch die CPU verarbeitet und anschließend durch Aktoren umgesetzt werden.

Abbildung 2.1: EVA-Prinzip; Quelle [20]

Die Programmierung und Konfiguration der Siemens-SPS erfolgt über das **Totally Integrated Automation Portal** (TIA Portal), eine integrierte Entwicklungsumgebung, die verschiedene Automatisierungskomponenten wie SPS, HMI und Antriebstechnik vereint. Das TIA Portal unterstützt die Programmiersprachen gemäß **IEC 61131-3**, darunter **Kontaktplan** (KOP), **Funktionsplan** (FUP), **Anweisungsliste** (AWL) und **Structured Control Language** (SCL). Diese Vielfalt ermöglicht eine anwendungsorientierte und normgerechte Entwicklung von Steuerungsprogrammen.[12]

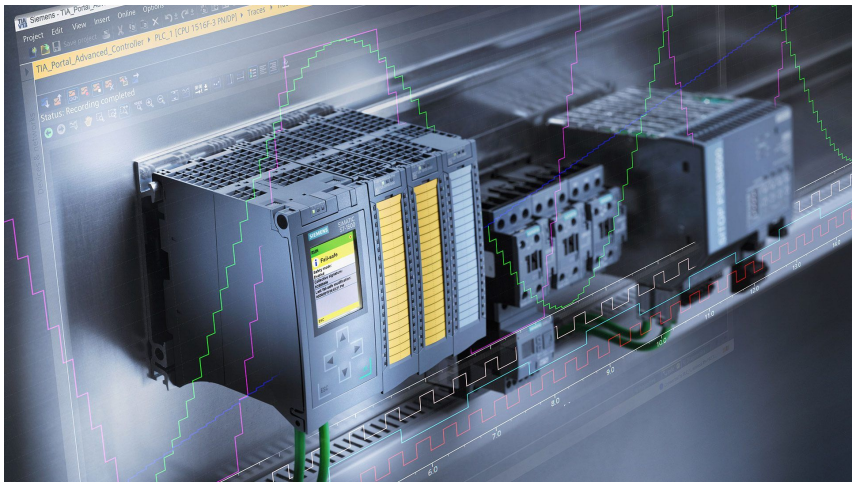


Abbildung 2.2: Beispiel S7-1500; Quelle [19]

Ein wesentliches Merkmal der **Siemens-SPS-Systeme** ist ihre Kompatibilität mit modernen Kommunikationsstandards wie **PROFINE**T und **PROFIBUS**. Dadurch wird eine nahtlose Integration in industrielle Netzwerke gewährleistet. Im Kontext von **Industrie 4.0** kommt den Siemens-SPS eine Schlüsselrolle zu, da sie die Grundlage für intelligente, vernetzte und adaptive Produktionssysteme bilden. Die Fähigkeit zur **Echtzeit-Datenerfassung und -verarbeitung** sowie zur Anbindung an Cloud- und

IoT-Plattformen macht sie zu einem integralen Bestandteil digitalisierter Fertigungsprozesse.[22]

2.2.3 SPS-Programmierung

In der Automatisierungstechnik werden speicherprogrammierbare Steuerungen (SPS) zur Realisierung unterschiedlichster Abläufe eingesetzt. Um die gewünschten Funktionen zur Automatisierung von Prozessen umzusetzen, ist es erforderlich, diese Abläufe in Form eines Programms in die Steuerung zu implementieren.
Ein SPS-Programm folgt dabei in der Regel einem klar strukturierten Aufbau [16]:

- Definition der Ein-, Ausgangs- und Hilfsvariablen in einer Variablentabelle,
- Programmierung des Hauptprogramms mit den zugehörigen Unterprogrammen und Funktionsbausteinen,
- Kompilierung des Programms und Übertragung in die Steuerung.

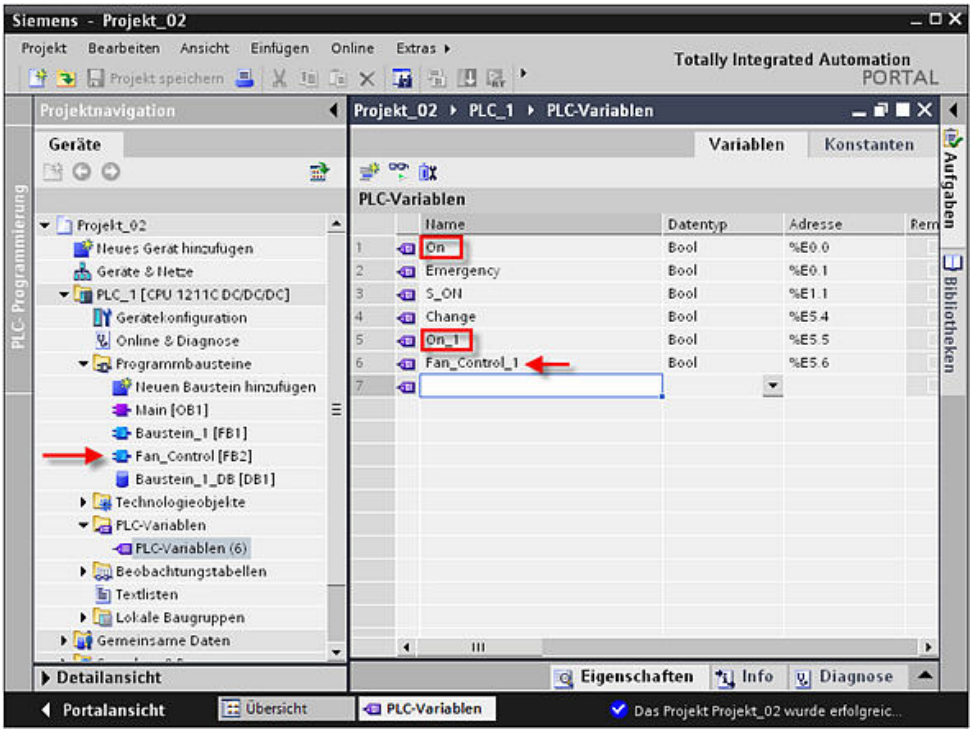


Abbildung 2.3: Übersicht der Variablentabelle in TIA Portal [18]

Das Hauptprogramm besteht aus sogenannten **POUs** (Program Organization Units, deutsch: Programmorganisationseinheiten). Diese werden in drei Arten unterteilt: **Programme**, **Funktionen (FC)** und **Funktionsbausteine (FB)**. Die Realisierung dieser POUs in *Siemens TIA Portal* wurde bereits im vorherigen Kapitel (vgl. **??**) beschrieben.

Nachfolgend wird die Bedeutung und Funktion der einzelnen Organisationseinheiten erläutert.
Das **Programm** dient der Realisierung komplexer Abläufe, beispielsweise von Schrittketten oder Hauptlogiken. Eine **Funktion (FC)** ist mit einer C-ähnlichen Funktion vergleichbar. Sie verarbeitet eine oder mehrere Eingangsvariablen und gibt nach der Abarbeitung ein Ergebnis zurück. Gemäß IEC 61131-3 können Funktionen jedoch nicht rekursiv aufgerufen werden und besitzen kein eigenes Speicherverhalten, d. h. sie können keine Werte dauerhaft speichern [16]. Ein einfaches Beispiel hierfür ist ein logisches **UND-Gatter**.
Ein **Funktionsbaustein (FB)** hingegen besitzt ein internes Gedächtnis und kann Werte über mehrere Zyklen hinweg speichern. Dadurch eignet er sich besonders für Zustandsautomaten oder speichernde Operationen. Ein klassisches Beispiel hierfür ist ein **RS-Flip-Flop**.
Für die Programmierung stehen verschiedene Sprachen nach IEC 61131-3 [1] zur Verfügung:

- KOP (Kontaktplan),
- AWL (Anweisungsliste),
- FUP/FBS (Funktionsplan / Funktionsbausteinsprache),
- ST/SCL (Structured Text / Structured Control Language)[1]

In modernen Industrieanlagen werden überwiegend **FUP** und **ST** eingesetzt. Diese Programmiersprachen zeichnen sich durch eine gute Übersichtlichkeit und eine effiziente Fehleranalyse aus. Trotzdem bestehen zwischen beiden Varianten wesentliche Unterschiede, die in Tabelle 2 dargestellt sind.

ST / SCL (Structured Text / Structured Control Language)	**FUP (Funktionsplan)**
Textbasierte Programmiersprache	Grafische Programmiersprache
Hochsprachenähnlich (z. B. C, Pascal)	Verwendung grafischer Symbole und logischer Verknüpfungen
Klare und kompakte Struktur	Gefahr der Unübersichtlichkeit bei komplexen Programmen
Besonders geeignet für mathematische und logische Operationen	Gut geeignet für einfache logische Abläufe

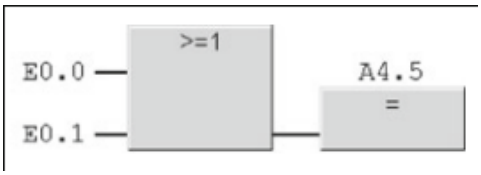
Tabelle 2: Vergleich zwischen ST (Structured Text) und FUP (Funktionsplan) [17]

[1] SCL ist die Siemens-spezifische Implementierung der nach IEC 61131-3 genormten Programmiersprache Structured Text (ST) im TIA Portal [13].

Wie aus der Tabelle hervorgeht, bietet die textbasierte Programmiersprache **ST** insbesondere bei komplexen Strukturen und mathematischen Berechnungen erhebliche Vorteile. **FUP** hingegen ist für Einsteiger leichter verständlich und eignet sich für überschaubare Steuerungslogiken. Mit zunehmender Komplexität der Abläufe stößt FUP jedoch schnell an seine Grenzen, da die grafische Darstellung umfangreicher Strukturen unübersichtlich werden kann [17]. Um die genauen Unterschiede zu erläutern, sollen zunächst grundlegende Elemente der einzelnen Sprachen eingeführt werden. Zunächst erfolgt eine Beschreibung der wichtigsten Grundbausteine in FUP.

Wichtige Grundfunktionen (UND, ODER, NICHT) in FUP

(a) **UND_Gatter** [11]

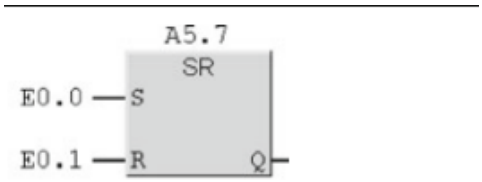


(b) **ODER_Gatter** [11]

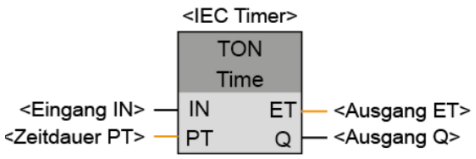
(c) **NOT_Gatter** [11]

Abbildung 2.4: Darstellung der logischen Grundfunktionen in FUP

Wichtige Funktionsbausteine in FUP



(a) **SR_FLIPFLOP** [11]



(b) **Einschaltverzögerung (TON)** [11]

Abbildung 2.5: Darstellung wichtiger Funktionsbausteine in FUP

In den obigen Abbildungen sind die wichtigsten Grundfunktionen und Funktionsbausteine in **FUP** dargestellt. Mit diesen können diverse Abläufe realisiert werden, wie beispielsweise einfache Förderbandsteuerungen. Komplexe, lange Abläufe sind jedoch sehr aufwendig zu realisieren. Für eine SPS ist es daher gebräuchlich, hierfür auf **ST**/**SCL** auszuweichen. In FUP können ebenfalls mathematische Operationen durchgeführt werden. Für Berechnungen existieren spezielle Bausteine. Allerdings ist es

schwierig, längere Rechenvorgänge übersichtlich darzustellen, da die Struktur schnell unübersichtlich wird. Aus diesem Grund werden für sämtliche mathematische Operationen häufig Berechnungen in **ST**/**SCL** durchgeführt. Beim vorliegenden Modell, dem Hochregallager, wurde beispielsweise die Positionsermittlung der Lagerplätze für das automatisierte Einlagern im vergangenen Studienprojekt in **SCL** realisiert.

Grundlegende Anweisungen in SCL

Im vorliegenden Projekt wird mit einer Siemens SPS gearbeitet. Somit ist SCL (Structured Control Language) der Standard im Bereich textbasierte Programmierung.
Hierbei gibt es einige konkrete Anweisungen, welche eine übersichtliche Programmierung gewährleisten.
Zu diesen zählen:

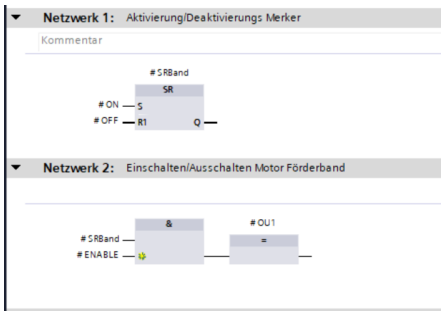
- **Zuweisungen:** `Variable := Wert;` Beispiel: `Motor_Start := TRUE;`
- **Bedingte Anweisungen (IF-Strukturen):** `IF Sensor = TRUE THEN Motor := TRUE; END_IF;`
- **Mehrzweigige Bedingungen (IF-ELSIF-ELSE):** `IF Temp > 50 THEN Alarm := TRUE; ELSIF Temp > 30 THEN Warning := TRUE; ELSE Alarm := FALSE; END_IF;`
- **Schleifenstrukturen:** `FOR i := 1 TO 5 DO Count := Count + 1; END_FOR;`
- **Vergleiche und logische Operatoren:** `AND, OR, NOT, >, <, =, >=, <=`

Diese Anweisungen ermöglichen selbst komplexe Programme zu realisieren. Es können mathematische Berechnungen deutlich einfacher durchgeführt werden als beispielsweise in der Programmiersprache **FUP**.

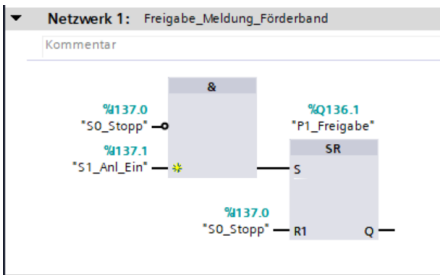
Vergleich SCL - FUP

Die Programmiersprachen **FUP** und **SCL** werden beide für die Programmierung speicherprogrammierbarer Steuerungen verwendet.
Da **SCL** eine Siemens-eigene Sprache ist, konzentriert sich dieser Vergleich ausschließlich auf **STEP 7**.
Wie bereits erwähnt, ermöglicht **FUP** eine einfache blockbasierte Programmierung.
FUP eignet sich besonders für einfache Anwendungen, wie beispielsweise die Ansteuerung von Drehstrommotoren. Auch für einfache Schrittketten ist **FUP** gut geeignet.
Mit zunehmender Programmkomplexität leidet jedoch die Übersichtlichkeit, was zu einer erhöhten Fehleranfälligkeit führen kann.
Um diese Probleme zu vermeiden, wird in großen Produktionsanlagen mit komplexen

Abläufen überwiegend textbasiert, also in **SCL**, programmiert. Die folgenden Darstellungen vergleichen beide Sprachen und verdeutlichen ihre Unterschiede. Dargestellt ist eine einfache Förderbandsteuerung in **FUP** und **SCL** im TIA Portal.



(a) Funktionsbaustein zur Steuerung des Förderbands



(b) Freigabe im Haupt-OB

Abbildung 2.6: Realisierung der Förderbandsteuerung in **FUP**

```
1 IF #OFF THEN
2         #SR_Band := FALSE;
3 ELSIF #ON THEN
4         #SR_Band := TRUE;
5 END_IF;
6
7 #OU1 := #SR_Band AND #ENABLE;
8
```

(a) SCL-Code des Förderband-Bausteins

```
1  IF NOT "S0_Stop" AND "S1_Anl_Ein
       " THEN
2          "Pl_Freigabe" := TRUE;
3  ELSIF "S0_Stop" THEN
4          "Pl_Freigabe" := FALSE;
5  END_IF;
6
7  "Foerderband"(
8  OFF    := "B2_vorne",
9  ON     := "B1_hinten",
10 ENABLE := "Pl_Freigabe",
11 OU1    => "Q1_Band");
12
```

(b) Aufruf des Förderband-Bausteins im Hauptprogramm

Abbildung 2.7: Realisierung des Förderbandprogramms in **SCL**

Anhand des Beispiels (2.6 2.7) wird deutlich, dass **FUP** auf den ersten Blick leichter verständlich wirkt, da es sich um die grafische Verschaltung logischer Funktionen handelt.
Auf den zweiten Blick zeigt sich jedoch, dass der Programmieraufwand deutlich höher ist als in **SCL**. In **SCL** kann dieselbe Funktionalität mit wenigen, klar strukturierten Codezeilen realisiert werden, ohne das Programm mit zusätzlichen Netzwerken zu überladen.
Zusammenfassend lässt sich festhalten, dass sowohl **FUP** als auch **SCL** ihre jeweiligen

Vor- und Nachteile besitzen. Je nach Anwendung und Komplexität des Prozesses fällt die Entscheidung für die eine oder die andere Programmiersprache.

2.2.4 Industrielle Bussysteme

In der Automatisierung sind Kommunikationsprotokolle entscheidend für die Vernetzung einzelner Systeme. Die Funktionalität einer automatisierten Anlage basiert auf der erfolgreichen Kommunikation der einzelnen Komponenten.

Bussysteme in der Automatisierungstechnik

Ein großer Bestandteil im Bereich der Kommunikation zwischen verschiedenen Teilnehmern bilden sogenannte Bussysteme.
Sie kommen zum Einsatz, wenn eine Steuerung mit den Sensoren bzw. Aktoren eines Systems effizient kommunizieren soll. Die Steuerung kann z. B. eine **SPS** oder ein **Mikrocontroller** sein.
Früher wurden Signale hauptsächlich über Hartverdrahtung übertragen. Diese Methode war zwar zuverlässig, jedoch sehr aufwändig zu implementieren und mit hohem Material- und Zeitaufwand verbunden.
Deshalb wurden die bereits erwähnten Bussysteme eingeführt. Mit deren Hilfe ist es möglich, mit geringem Verdrahtungsaufwand mit mehreren Teilnehmern innerhalb eines Systems zu kommunizieren.
Ein Bussystem besteht grundsätzlich aus einer Leitung, über die mehrere Teilnehmer angesprochen werden können. Die Topologie ist hierbei vielfältig. Es gibt sogenannte **Stern**-, **Linien**- oder auch **Ring**-Topologien.

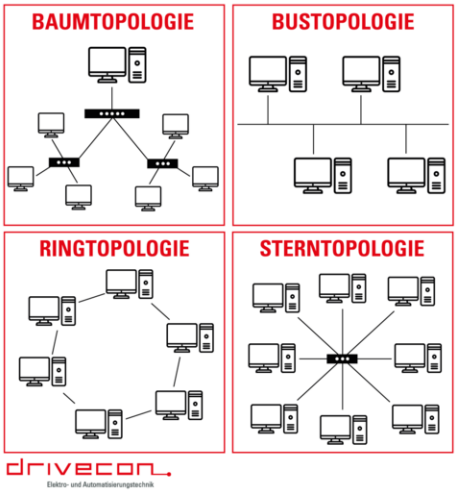


Abbildung 2.8: Gängige Bus- Topologien; Quelle [6]

Die am häufigsten angewandte Methode ist die Kommunikation über eine **Linientopologie**.

Hierbei werden mehrere Teilnehmer über einzelne Abzweige nacheinander angesteuert. Bussysteme können auf den verschiedensten Ebenen der Automatisierungspyramide[5] zum Einsatz kommen. Der Haupteinsatz erfolgt jedoch auf der Feld- und Steuerungsebene.
Weiterhin gibt es verschiedene Arten von Bussystemen, die sich in ihrer Struktur, Geschwindigkeit und Übertragungsart unterscheiden. Die am häufigsten verwendeten sind im Folgenden aufgelistet:

- Feldbus-basierte Kommunikationssysteme (z. B. **Profibus**)
- Ethernet-basierte Kommunikationssysteme (z. B. **PROFINET**)
- Punkt-zu-Punkt-Systeme (z. B. **IO-Link**)

Im Folgenden eine kurze Gegenüberstellung der einzelnen Protokolle:

Kriterium	**IO-Link**	**PROFINET**	**PROFIBUS**
Kommunikationsprinzip	Punkt-zu-Punkt (PTP)	Ethernet-basiert	Feldbus-basiert
Topologie	Baum	Linie, Stern, Ring	Linie, Baum
Kommunikationsrate	4,8 / 38,4 / 230,4 kBaud	bis 100 Mbit/s	bis 12 Mbit/s
Typische Anwendung	Sensorik, Aktorik, Handhabungstechnik	Steuerung, Kommunikation, Anlagenvernetzung	Maschinensteuer., Antriebsregelung

Tabelle 3: Vergleich der Kommunikationssysteme IO-Link, PROFINET und PROFIBUS; Quellen: [5][15][14]

2.2.5 IO-Link

IO-Link ist ein Kommunikationssystem, welches der Anbindung intelligenter Sensoren und Aktoren dient.
Es wurde entwickelt, um die Lücke zwischen Feldbussen und Industrial-Ethernet-Systemen zu schließen. Der Standard ist in der Norm IEC 61131-9 definiert. IO-Link wird häufig in Fabrikautomatisierungsanlagen eingesetzt.

Aufbau und Funktionsweise von IO-Link

IO-Link ist ein Kommunikationssystem, das nach dem **Point-to-Point** (PTP)-Prinzip arbeitet. Das bedeutet, es werden mehrere Teilnehmer Punkt zu Punkt angesteuert.

Der **IO-Link-Master** ist hierbei mit den **IO-Link-Devices** PTP-verbunden. Dieser stellt die Schnittstelle zwischen der Steuerung (**SPS**) und der Sensor-/Aktor-Ebene dar.

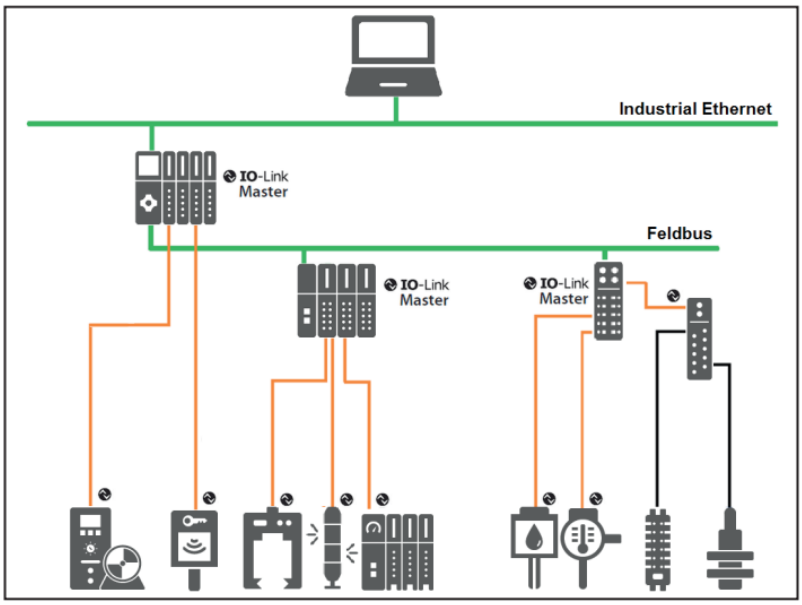


Abbildung 2.9: Architektur IO_Link; Quelle [5]

Das System arbeitet **bidirektional**[5]. Das bedeutet, der IO-Link-Master kann sowohl Prozessdaten empfangen als auch neue Parameter an die Devices senden (z. B. **RFID**). Die Verbindung erfolgt über dreipolige Standardleitungen2.10. Dadurch lässt sich IO-Link problemlos in das Gesamtsystem integrieren. Eine separate Busverkabelung ist nicht notwendig[2].

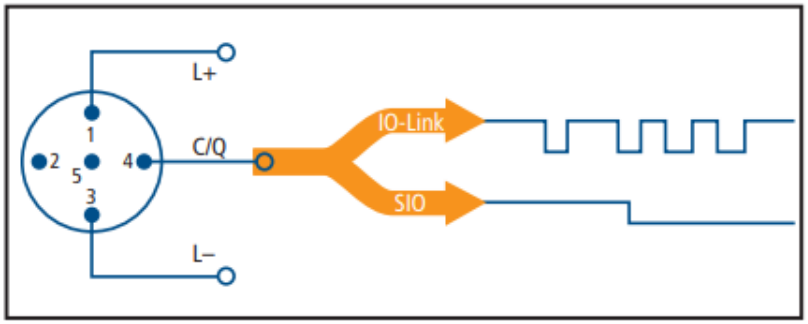


Abbildung 2.10: Anschlussbelegung IO-Link; Quelle [5]

Der IO-Link-Master2.11 kann über ein übergeordnetes Bussystem (z. B. **Profibus**, **PROFINET**) mit der SPS kommunizieren[14].

Abbildung 2.11: Master IO_Link; Quelle [9]

Kommunikation und Datenaustausch

Die Kommunikation über IO-Link erfolgt seriell. Hierbei wird mit festen Baudraten von 4,8 kBaud, 38,4 kBaud oder 230,4 kBaud übertragen.

Die Datenkommunikation gliedert sich in drei Hauptarten[5]:

- **Prozessdaten**:
  Diese werden zyklisch zwischen Master und Device ausgetauscht. Sie enthalten z. B. Messwerte von Sensoren oder Schaltzustände von Aktoren.
- **Servicedaten (Parameterdaten)**:
  Sie werden azyklisch übertragen und dienen zur Parametrierung oder Konfiguration der Geräte. So können beispielsweise Schaltpunkte, Messbereiche oder Filterzeiten direkt über die Steuerung angepasst werden.
- **Ereignisdaten**:
  Hierbei handelt es sich um Status- oder Fehlermeldungen, die das Device bei bestimmten Zuständen (z. B. Übertemperatur, Kabelbruch oder Spannungsfehler) an den Master sendet.

Durch diese Aufteilung ist eine gezielte Überwachung, Diagnose bei Fehlern und Konfiguration der einzelnen Module möglich. Eine Prozessunterbrechung ist hierfür nicht notwendig.
Auf Protokollebene wird zwischen drei Schichten unterschieden[5]:

- **Physikalische Schicht** – definiert die elektrische Verbindung über die dreipolige Leitung.

- **Datenverbindungsschicht** – regelt das Telegrammformat und die Fehlererkennung.
- **Applikationsschicht** – beschreibt den Austausch von Prozess-, Service- und Ereignisdaten.

Komponenten eines IO-Link-Systems

Ein vollständiges IO-Link-System besteht in der Regel aus folgenden Komponenten[5]:

- **IO-Link-Master**:
  Er verwaltet die Kommunikation zu den einzelnen IO-Link-Devices und bildet die Schnittstelle zur übergeordneten Steuerung. Je nach Ausführung kann der Master als Schaltschrankmodul oder als Feldmodul ausgeführt sein. Letzteres wird direkt in der Anlage montiert, wodurch sich Installationsaufwand und Verkabelungslänge reduzieren.
- **IO-Link-Devices**:
  Hierbei handelt es sich um Sensoren, Aktoren oder andere Feldgeräte, die über IO-Link kommunizieren. Typische Beispiele sind Drucksensoren, Wegmesssysteme, Ventilinseln, RFID-Leseeinheiten oder Smart Lights. Jedes Device verfügt über eine IODD-Datei (*IO Device Description*), in der alle Geräteparameter, Kommunikationsdaten und Diagnosemöglichkeiten beschrieben sind. Diese Datei wird im Engineering-Tool der Steuerung verwendet, um das Gerät automatisch zu erkennen und korrekt zu parametrieren.
- **Verkabelung**:
  IO-Link nutzt standardisierte, ungeschirmte Leitungen mit M12- oder M8-Steckverbindern. Die Kabellängen betragen typischerweise bis zu 20 m zwischen Master und Device. Da das Signal digital übertragen wird, ist die Verbindung unempfindlich gegenüber elektromagnetischen Störungen und Spannungsabfällen.

Zusammenfassung

Über IO-Link ist eine Kommunikation bis in die unterste Feldebene möglich. Analoge Signale sind hierbei überflüssig, da die Schnittstelle standardisiert ist. Weiterhin lassen sich Geräte identifizieren und parametrieren. Außerdem stehen umfassende Diagnosemöglichkeiten zur Verfügung.
Damit trägt IO-Link wesentlich zur Transparenz, Flexibilität und Wartungsfreundlichkeit moderner Automatisierungssysteme bei und bildet eine wichtige Grundlage für Konzepte wie **Industrie 4.0**[5].

2.2.6 Vorteile und Nutzen in der Praxis

Der Einsatz von IO-Link in der Industrie bringt zahlreiche Vorteile mit sich. Es werden technische Defizite von klassischen Bus- und Anschlusssystemen eliminiert. Neben der Vereinfachung der Verdrahtung bietet das System eine hohe Flexibilität, verbesserte Diagnosefähigkeiten sowie eine sehr gute digitale Kommunikation bis in die Feldebene hinein.
Durch die genannten Punkte wird nicht nur die Leistungsfähigkeit der Anlage gesteigert, sondern auch die Wirtschaftlichkeit und Nachhaltigkeit in einer Produktion verbessert.

Technische Vorteile

Ein zentraler technischer Vorteil von IO-Link besteht in der vollständig digitalen Signalübertragung zwischen dem IO-Link-Master und den angeschlossenen Devices. Dadurch entfallen typische Nachteile analoger Schnittstellen wie Messfehler durch Spannungsabfall, Rauschen oder Kalibrierungsabweichungen. Jeder über IO-Link verbundene Sensor oder Aktor überträgt exakte Prozessdaten in digitaler Form, wodurch die Messqualität und Zuverlässigkeit deutlich erhöht werden.
Ein weiterer Vorteil ist die sogenannte bidirektionale Kommunikation. Hierbei ist es möglich, dass der IO-Link-Master Daten empfangen kann, sowie in der Lage ist, die Parameter der Sensoren oder Aktoren zu beschreiben.
Darüber hinaus bietet IO-Link standardisierte Schnittstellen und Datenstrukturen, die herstellerübergreifend kompatibel sind. Jedes Gerät wird anhand seiner IODD-Datei (IO Device Description) automatisch erkannt, was die Integration in bestehende Systeme erheblich vereinfacht[5]. Diese Einheitlichkeit führt zu einer hohen Austauschbarkeit der Komponenten, ohne dass Änderungen in der SPS-Software oder in der Verdrahtung notwendig sind.

Weitere Vorteile

Durch den verringerten Verdrahtungsaufwand können erhebliche Kosten eingespart werden. Diese können an anderer Stelle effizienter eingesetzt werden.
Da ein IO-Link-Master in der Lage ist, sowohl klassische als auch digitale Signale zu verarbeiten, entfallen zusätzliche Baugruppen zur Verarbeitung bestimmter Signalarten. Diese Punkte führen zu einer Erhöhung bzw. Verbesserung der Wirtschaftlichkeit. Dadurch kann nachhaltiger und effektiver gearbeitet werden.

Erweiterte Diagnosemöglichkeiten

Eine besondere Eigenschaft von IO-Link liegt in den umfassenden Diagnosemöglichkeiten, die dem Anwender zur Verfügung stehen. Jedes **Device** ist in der Lage, sich selbst

zu diagnostizieren und die entsprechenden Informationen an den **IO-Link-Master** zu übermitteln.
Somit werden Fehler wie Kabelbruch, Kurzschluss oder Unterspannung sofort an die Steuerung gemeldet. Dadurch ist ein schnelles und effektives *Trouble Monitoring* gegeben. Dies führt dazu, dass langfristig Fehlerquellen und Anfälligkeiten in Systemen erkannt und gezielt beseitigt werden können.

Praxisbeispiel

In der industriellen Praxis findet IO-Link zunehmend Anwendung bei intelligenten Sensoren und Aktoren, etwa in Montage- und Prüfprozessen oder bei der Handhabungstechnik.
Ein typisches Beispiel ist die Integration eines IO-Link-Füllstandssensors oder Drucksensors in ein SPS-System. Die Messwerte werden dabei digital und störungsfrei übertragen, während gleichzeitig Geräteparameter über die Steuerung angepasst werden können – etwa der Schaltpunkt oder der Messbereich[5].

Zusammenfassung

IO-Link vereint einfache Installation mit intelligenter Kommunikation. Die Technologie schafft eine durchgängige Datenbasis von der Feldebene bis zur Steuerung und ermöglicht so eine effiziente, flexible und zukunftssichere Automatisierung.
Durch die Kombination aus digitaler Signalübertragung, erweiterter Diagnose und automatischer Parametrierung wird IO-Link zu einem zentralen Bestandteil moderner Industrie-4.0-Architekturen.

2.2.7 Sicherheit in der Automatisierung

Im Kontext der Automatisierung von technischen Anlagen ist die Sicherheit der größte und wichtigste Aspekt [4].
Beim Betreiben eines Systems muss vollständig ausgeschlossen sein, dass keine Gefahr für Menschen, von der Anlage, ausgeht. Das bedeutet, bei Planung der Anlage, steht die Sicherheit im Fokus.
Speziell in der Produktion bzw. in der Logistik sind sicherheitstechnische Aspekte enorm wichtig.
Bei automatisierten Prozessen wie z.B. das Verfahren eines Hochregallagers, können Gefahren durch Implementierung von Sicherheitslogiken ausgeschlossen werden.
Das Ziel ist immer Gefährdungen bzw. Verletzungen von Menschen zu verhindern.

Einordnung in den Kontext der funktionalen Sicherheit

Um die Sicherheitsprinzipien besser zu verstehen, ist es wichtig, die einzelnen Aspekte der Sicherheitsthematik im Gesamtkontext einzuordnen.
In diesem Zusammenhang spielt der Bereich der funktionalen Sicherheit eine entscheidende Rolle.
Hierunter versteht man den Teil einer Maschine, der von den korrekt funktionierenden Steuerungs- und Sicherheitskomponenten abhängt. Sie sorgt dafür, dass bei kritischen Situationen ein sicherer Zustand der Anlage erreicht wird. Ein Beispiel hierfür wäre das direkte Abschalten eines Motors[10].
Damit sicherheitsbezogene Steuerungssysteme zuverlässig arbeiten, müssen sie bestimmte Anforderungen erfüllen, die in internationalen Normen festgelegt sind.
Einige dieser Vorschriften sind im Folgenden aufgelistet[10]:

- **EN ISO 13849-1**: Beschreibt die sicherheitsrelevanten Teile einer Anlage, unter anderem den **PL (Performance Level)**.
- **IEC 61508**: Behandelt die funktionale Sicherheit elektrischer, elektronischer und programmierbarer Steuerungssysteme. Basiert auf dem **SIL (Safety Integrity Level)**.
- **EN 61496**: Sicherheit von Maschinen – Berührungslos wirkende Schutzeinrichtungen (BWS), beispielsweise Lichtschranken oder Lichtvorhänge.

Die genannten Normen zeigen ebenfalls die Bedeutung der Sicherheit im Anlagenbau für den Gesetzgeber. Falls es zu Verstößen kommen sollte, führt dies zu Konsequenzen für den Arbeitgeber.
Das ist der Grund dafür, dass Unternehmen sehr genau auf die Einhaltung dieser Vorschriften achten.
Für die genannten Normen gibt es dementsprechende Geräte, welche beispielsweise im Fehlerfall (Fehlerstrom) abschalten sollen (FI-Schutzschalter).
Die Auswahl der Sicherheitskomponenten erfolgt auf Basis einer sogenannten **Risikobeurteilung**. Diese ermittelt die potenziellen Gefahren und legt das **Performance Level** bzw. den **SIL** fest. In der Praxis werden Sicherheitsfunktionen häufig mit einem **PL d** oder **SIL 2** realisiert, da diese ein gutes Verhältnis zwischen Aufwand und Sicherheit bieten.

Eine zentrale Rolle innerhalb dieser sicherheitsrelevanten Systeme spielt die **Sensorik**. Sie dient als Grundlage zur Erfassung von Zuständen und Bewegungen innerhalb einer Anlage und bildet somit die Basis für jede Sicherheitsfunktion.
Besonders in automatisierten Prozessen ist es wichtig, dass Gefahrenzonen zuverlässig überwacht werden, ohne den Produktionsablauf zu beeinträchtigen.

Hier kommen berührungslos wirkende Schutzeinrichtungen zum Einsatz, wie beispielsweise **Lichtschranken** oder **Lichtvorhänge**. Diese ermöglichen eine sichere und zugleich flexible Absicherung von Arbeitsbereichen, an denen Mensch und Maschine aufeinandertreffen.
Lichtschranken zählen zu den am weitesten verbreiteten Sicherheitseinrichtungen in der Industrie und werden unter anderem bei Robotersystemen, Förderanlagen oder automatisierten Handhabungsstationen verwendet.
Im folgenden Abschnitt wird die Funktionsweise, der Aufbau sowie der Einsatz von Lichtschranken in sicherheitsrelevanten Anwendungen näher erläutert.

2.2.8 Lichtschranken in sicherheitsrelevanten Anlagen

Die Lichtschranke ist eine der wichtigsten Sicherheitseinrichtungen in der Fertigung und Logistik.
Sie verhindert das unbefugte Eingreifen in laufende Prozesse, sodass mögliche Gefahren für Personen zuverlässig ausgeschlossen werden können.
Viele Hersteller bieten ihre Bauteile frei konfigurierbar an. Dadurch ist es möglich, die Betriebsmodi und Parameter individuell an die jeweilige Anwendung anzupassen, beispielsweise bei der Absicherung von Hochregallagern oder automatisierten Förderanlagen.
Lichtschranken kommen in sicherheitsrelevanten Anlagen überall dort zum Einsatz, wo Gefahrenbereiche zuverlässig überwacht werden müssen, ohne dass mechanische Barrieren den Prozessfluss behindern.

Funktionalität einer Lichtschranke

Bei einer Lichtschranke handelt es sich um eine berührungslos wirkende Schutzeinrichtung, die mit einem oder mehreren Lichtstrahlen arbeitet. Diese Lichtstrahlen bilden eine optische Barriere, die bei Unterbrechung ein Signal auslöst.
Eine Lichtschranke besteht grundsätzlich aus zwei Modulen: einem **Sender** und einem **Empfänger**. Der Sender erzeugt ein Lichtsignal, das vom Empfänger erfasst wird. Wird der Lichtstrahl unterbrochen, erkennt die Auswerteelektronik diesen Zustand und leitet eine vordefinierte Sicherheitsreaktion ein – beispielsweise das Abschalten eines Antriebs oder das Anhalten eines Roboters.
Die am häufigsten verwendeten Lichtschrankentypen sind die **Reflexionslichtschranke** und die **Infrarotlichtschranke**[3].
Bei einer **Reflexionslichtschranke** befinden sich Sender und Empfänger im selben Gehäuse. Das ausgesendete Lichtsignal wird über einen Reflektor zurückgeworfen und von der Empfangseinheit detektiert. Wird der Lichtstrahl durch ein Objekt unterbrochen, bleibt das Signal aus und die Sicherheitsfunktion wird aktiviert.

Die im Projekt verwendete Variante ist eine **Infrarotlichtschranke**. Diese arbeitet mit unsichtbarem Infrarotlicht, das unempfindlich gegenüber Umgebungslicht und Staub ist.

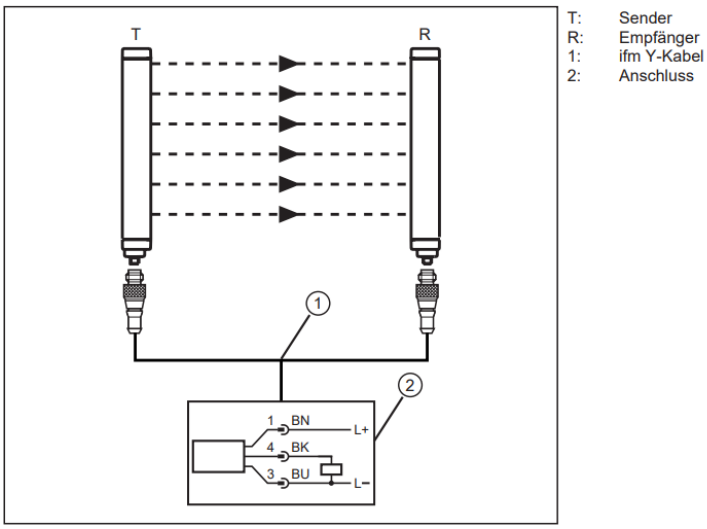


Abbildung 2.12: Schematische Darstellung des IFM Lichtgatters[7]

Hierbei wird ein kontinuierlicher Infrarotstrahl vom Sender ausgesendet, der vom Empfänger registriert wird. Wird der Strahl unterbrochen, erkennt die Steuerung dies sofort als Störung oder Eingriff und löst eine Sicherheitsreaktion aus 2.12.
Durch die hohe Reichweite und Störsicherheit eignet sich die Infrarotlichtschranke besonders für industrielle Umgebungen, in denen eine präzise und zuverlässige Objekterkennung gefordert ist.
Häufig variiert die Bezeichnung solcher Systeme. Viele Hersteller verwenden dafür auch den Begriff **Lichtgitter** oder **Lichtvorhang**.
So auch der Hersteller **ifm**, der das im Projekt verwendete Produkt bereitstellt[8]. Diese Systeme sind modular aufgebaut und lassen sich über verschiedene Schnittstellen – beispielsweise IO-Link – direkt in eine Steuerung integrieren.

3 Implementierung eines Lichtgatters in das Gesamtkonzept

Im aktuellen Ist-Zustand des Hochregallagers fehlt die vollständige Implementierung der Sicherheitslogik.
Lediglich der Not-Aus-Schalter ist hardwareseitig verdrahtet und schaltet bei Betätigung die Zuleitung ab. Es existiert jedoch keine automatische Abschaltung im Gefahrenfall. Die Logik des Ein- und Auslagerns ist somit nicht gefahrenfrei. Sobald ein Eingriff in den Prozess erfolgt, kann dieser im aktuellen Zustand nicht automatisch gestoppt werden. Da das Miniaturmodell eine reale Situation in der Logistik abbilden soll, stellt dieser Punkt ein erhebliches Defizit dar.
Es gibt verschiedene Möglichkeiten, um diesen Nachteil zu beheben. Die naheliegendste Option ist die Implementierung einer **Lichtgatter-Funktion**, um den Gefahrenbereich zuverlässig zu überwachen und den Betrieb im Störfall sicher zu unterbrechen. Durch ein vorheriges Projekt wurde bereits, ein Modell der Firma IFM[8], installiert.

3.1 Zielsetzung der Implementierung

Die Studienarbeiten der vergangenen Jahre haben das Hochregallager in einen Zustand gebracht, der wichtige Funktionen wie Ein- und Auslagern abbildet.
Jedoch wurden hierbei, fälschlicherweise, die notwendigen Sicherheitsaspekte völlig außer Acht gelassen.
In einer realen Umgebung, beispielsweise in Materiallagern mit Hochregalsystemen großer Industrieunternehmen, wäre eine solche Vorgehensweise undenkbar.
Es muss jederzeit gewährleistet sein, dass Gefahren so gut wie möglich unter Kontrolle gehalten werden. Das bedeutet, im Fehlerfall muss jede gefährliche Bewegung sicher abgeschaltet werden.
Dies stellt die zentrale Zielsetzung der Implementierung dar. In jedem Betriebsmodus muss sichergestellt sein, dass bei Erkennung einer Gefahr automatisch abgeschaltet wird.
Dabei ist jedoch zu berücksichtigen, dass der Kolben des Zylinders, welcher die Materialien einlagert, kein Abschaltsignal auslösen darf. Dieser fährt nämlich unmittelbar in den Gatterbereich der Lichtschranke ein.
Somit ist zunächst eine umfassende Analyse des Ist-Zustands sowie der verschiedenen Prozesswerte des Lichtgatters erforderlich.
Erst im Anschluss kann die Implementierung der Sicherheitslogik erfolgen.

3.2 Analyse der Parametrierung des Lichtgatters

Die Konfiguration des Lichtgatters ist entscheidend für die richtige Funktionsweise. Beim Modell von **IFM**[8] gibt es mehrere Möglichkeiten zur Parametrierung.
Das Lichtgatter besitzt mehrere Lichtstrahlen, welche über einzelne Bits definiert sind. Sobald sich ein Objekt im Gatter befindet, werden mehrere dieser Lichtstrahlen gleichzeitig unterbrochen.
Der Sensor des vorliegenden Lichtgatters hat im sogenannten **Detektionsmodus (GTBO)** den Wert 1. Das bedeutet, dass er schaltet, sobald ein Objekt die Strahlen unterbricht.
Über **IO-Link** kann nun festgelegt werden, ab wie vielen unterbrochenen Strahlen der Sensor ein Schaltsignal ausgeben soll.

3.2.1 Prozesswerte zur Steuerung des Arbeitsverhaltens

Zur Steuerung des Verhaltens bei einer Unterbrechung stehen verschiedene Prozesswerte zur Verfügung[7]:

- **FBO (First Beam Occupied)**
- **LBO (Last Beam Occupied)**
- **CBO (Central Beam Occupied)**
- **NBO (Number of Beams Occupied)**
- **NCBO (Number of Consecutive Beams Occupied)**

Im Betriebsmodus **FBO** schaltet der Sensor bereits beim ersten unterbrochenen Lichtstrahl. Das bedeutet, Objekte, die im unteren Bereich des Lichtgatters liegen, werden als erstes erfasst.
Im Modus **LBO** geschieht genau das Gegenteil: Es wird beim letzten Lichtstrahl geschaltet.
Dieser Modus wird beispielsweise bei Endlagenerkennungen oder beim Auslagern verwendet – also immer dann, wenn das Austreten aus einem bestimmten Bereich überwacht werden soll.

Der Modus **CBO** ist etwas komplexer: Hier wird der mittlere Lichtstrahl (bei mehreren zusammenhängenden Strahlen) überwacht. Tritt ein Objekt in das Lichtgatter ein, wird der zentrale Strahl als Referenzpunkt betrachtet. Bei mehreren Objekten wird immer vom größeren Objekt ausgegangen, also von dem, das mehr Strahlen gleichzeitig unterbricht – wie in Abbildung 3.13 zu sehen.

Anwendungsbeispiele sind z. B. Durchgänge oder Förderabschnitte, also überall dort, wo Objekte mittig erfasst werden sollen (Referenzpunkt: Mitte).

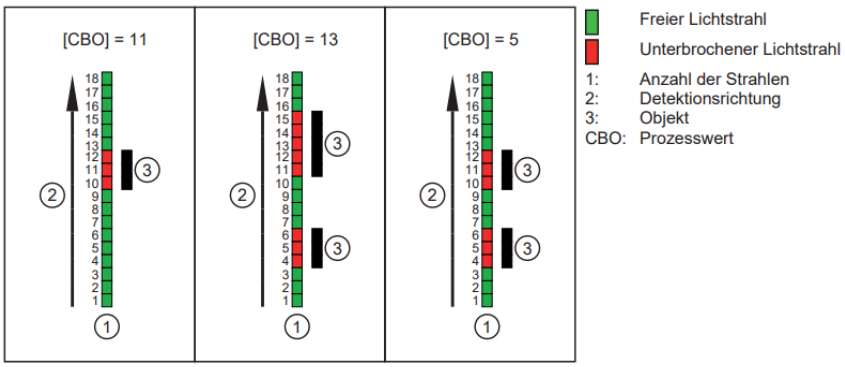


Abbildung 3.13: Modus **CBO**; Quelle: [7]

Die letzten beiden Modi, **NBO** und **NCBO**, beschäftigen sich mit der konkreten Anzahl der Lichtstrahlen, die unterbrochen werden.
Der Unterschied der beiden Modi liegt darin, dass **NBO** die Gesamtanzahl der unterbrochenen Lichtstrahlen angibt, während **NCBO** die Anzahl der Strahlen beschreibt, die *aufeinanderfolgend* unterbrochen werden.
Im Projekt wird der Modus **NBO** verwendet. Eine Abbildung soll die Funktionsweise noch einmal verdeutlichen (siehe Abbildung 3.14).

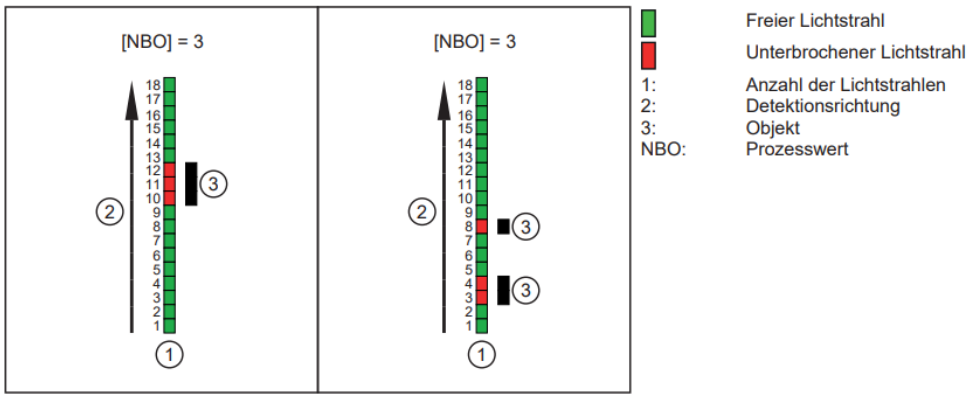


Abbildung 3.14: Funktionsweise des **NBO-Modus**; Quelle: [7]

3.3 Komponentenübersicht und Systemaufbau

Für die Umsetzung der Sicherheitsfunktion wurde ein Systemaufbau realisiert, der die Lichtschranke direkt mit einem IO-Link-Master und der SPS verbindet. Im Folgenden werden die eingesetzten Komponenten sowie deren Zusammenspiel innerhalb des Gesamtsystems beschrieben.

Komponente	**Hersteller / Typ**	**Funktion**	**Schnittstelle**
Lichtgitter / Lichtschranke	IFM OY5100 (Infrarot)	Erfassung von Objekten und Überwachung des Gefahrenbereichs	IO-Link
IO-Link Master	IFM AL1100	Kommunikation zwischen IO-Link-Devices und der SPS	PROFINET / IO-Link
SPS	Siemens S7-1500	Steuerung des gesamten Systems, Verarbeitung der Sensorsignale	PROFINET
Netzteil 24V DC	Siemens SITOP / vergleichbar	Versorgung von IO-Link-Master, Lichtschranke und SPS	24V DC
Aktorik (Zylinder / Motor)	Festo (pneumatisch)	Ausführung der Ein- und Auslagerbewegung	Digitale Ausgänge der SPS

Tabelle 4: Übersicht der verwendeten Komponenten im Systemaufbau

In Tabelle 4 sind die einzelnen Komponenten des Hochregallagers rund um die Lichtschranke dargestellt.
Diese müssen optimal miteinander interagieren, um die gewünschte Funktionalität sicherzustellen.
Die Lichtschranke ist vom Typ **OY5100** und stammt vom Hersteller **IFM**. Es handelt sich hierbei um ein steuerbares Lichtgitter, das über IO-Link mit der SPS kommuniziert.

3.3.1 Systemaufbau

Die folgende Abbildung zeigt den strukturellen Aufbau des Gesamtsystems in vereinfachter Form.

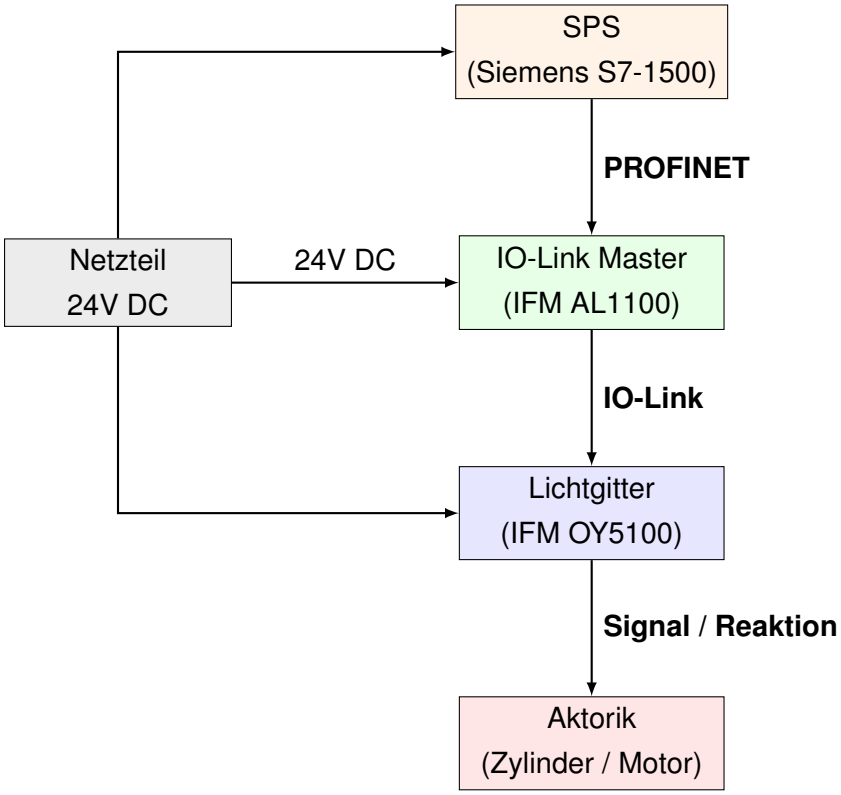


Abbildung 3.15: Vertikaler Systemaufbau des Lichtgatter-Systems

Im Diagramm 3.15 ist der Aufbau der einzelnen Komponenten grafisch dargestellt. Der IO-Link-Master dient hierbei als Schnittstelle zwischen SPS und Lichtgitter und steuert die Kommunikation über das im TIA Portal implementierte Programm.
Im Fehlerfall werden gefährliche Bewegungen, die beispielsweise vom Zylinder oder den Motoren der Linearachsen ausgehen, automatisch gestoppt.
Im folgenden Abschnitt wird die Implementierung dieser Sicherheitslogik in der SPS detailliert erläutert.

3.4 Implementierung der Sicherheitslogik in der SPS

Die Sicherheitslogik wird, wie bereits erwähnt, in eine **Siemens S7-1500** SPS einprogrammiert. Die Programmierung erfolgt mithilfe der Software **TIA Portal**.
Für die Umsetzung der Arbeit wurde das bestehende Programm des Jahrgangs **TEA 22** als Grundlage verwendet. Zunächst wurde eine neue *Program Organization Unit (POU)* mit der Bezeichnung **Lichtschranke** erstellt.
Im Anschluss erfolgte das Einbinden des Moduls **OY5100** im Reiter **Geräte und Netzwerke**. Der Adressbereich *(frei wählbar)* wurde so konfiguriert, dass eine eindeutige Zuordnung innerhalb des Projekts gewährleistet ist.

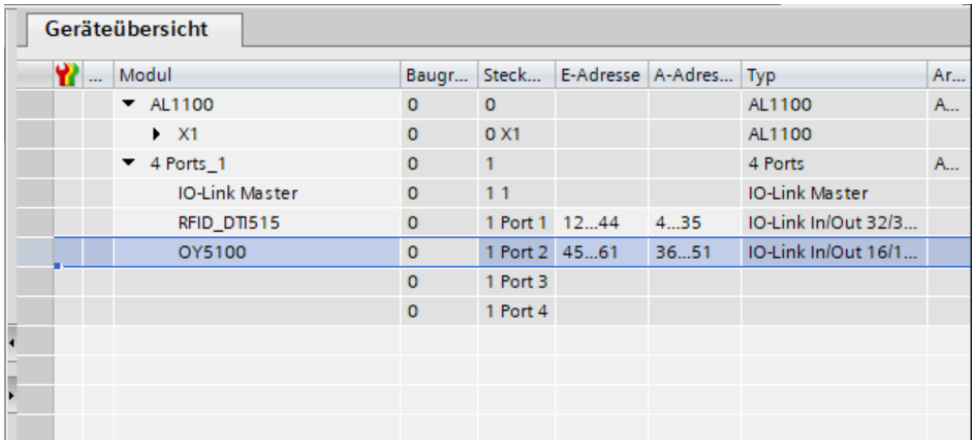


Abbildung 3.16: Einbindung der IFM-Lichtschranke OY5100 im TIA Portal

Die zugewiesenen Adressen werden für die verschiedenen Prozesswerte genutzt (siehe Kapitel 3.2.1).
Als Vorbereitung auf die Programmierung wurde eine sogenannte *Watchtable* erstellt. Diese dient dazu, die einzelnen Prozesswerte des Lichtgatters während der Laufzeit zu überwachen und deren Verhalten zu analysieren.
Die Prozesswerte liegen im Format **WORD** (2 Byte) vor und bilden die Anzahl der unterbrochenen Lichtstrahlen ab.

Nach einer Reihe von Tests wurde festgestellt, dass der Kolben des Zylinders im Arbeitsbereich des Lichtgatters **drei Strahlen** gleichzeitig unterbricht. Hierfür erfolgte ein gezieltes Verfahren der Lineareinheit über alle vier Ebenen des Hochregallagers. So konnte überprüft werden, dass der Kolben in jeder Position konstant drei Strahlen abdeckt.

Im weiteren Verlauf wurde diskutiert, welcher Prozesswert sich für die Sicherheitslogik am besten eignet. Es zeigte sich, dass der Modus **NBO (Number of Beams Occupied)**

die Summe aller unterbrochenen Lichtstrahlen angibt (siehe Kapitel 3.2.1). Daher ist dieser Prozesswert optimal geeignet, um eine zuverlässige Abschaltung über das Lichtgatter zu realisieren.
Nach der Diskussion wurde, für die Realisierung der Abfrage, ein neuer FB (Freigabe_Lichtschranke_FB24) aufgebaut in **SCL**. Im Diagramm 3.17 ist der Zusammenhang zwischen den einzelnen Modi und der **Freigabe** dargestellt.

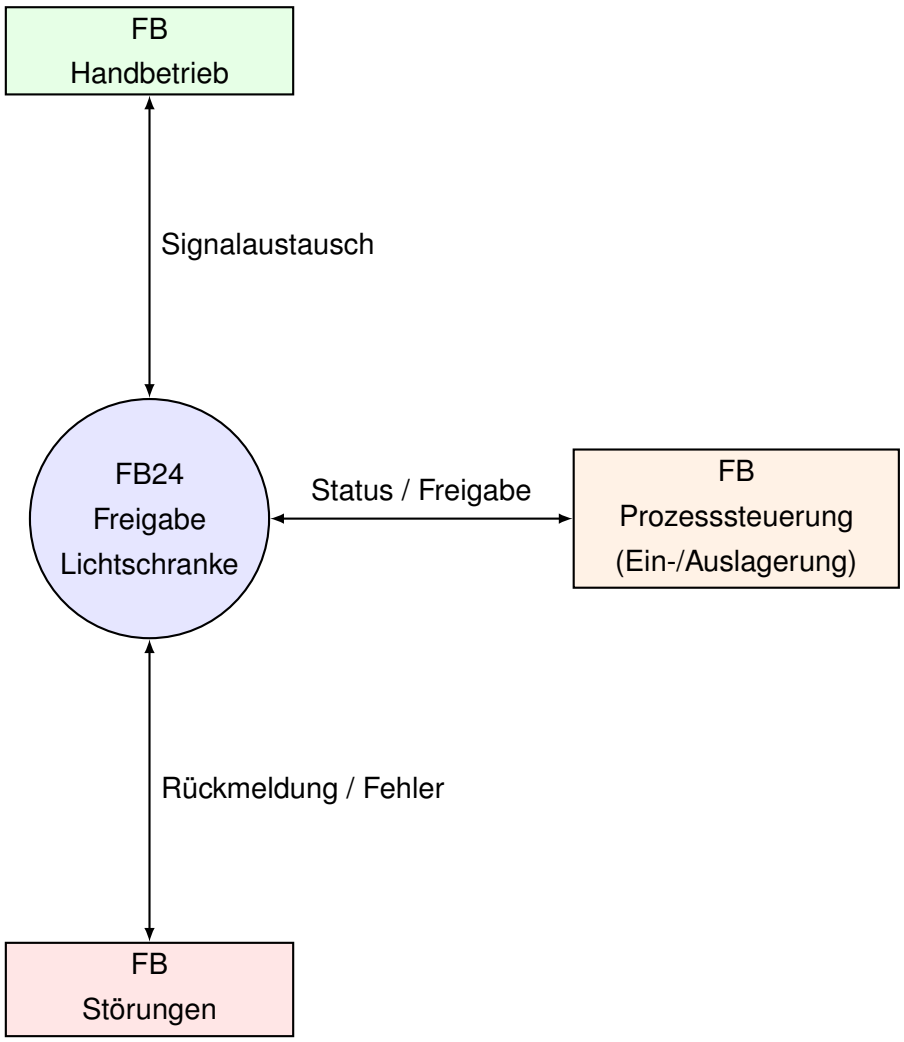


Abbildung 3.17: Einbindung des Bausteins `FB24 - Freigabe_Lichtschranke` in das Gesamtsystem

Der Baustein `FB24 - Freigabe_Lichtschranke` ist ein zentraler Bestandteil der Sicherheitsarchitektur.
Ohne ihn wäre ein sicheres Abschalten im Fehlerfall nicht möglich. Die **Prozesssteuerung** wird durch die Freigabe in jedem Schritt überwacht.
Durch die Einbindung in den `FB11 - Störungen` erfolgt eine Überwachung sämtlicher Ausgänge über die Variable `b_Freigabe`. Diese wird gesetzt, sobald keine Störungen aktiv sind und die Freigabe durch das Lichtgitter gegeben ist.

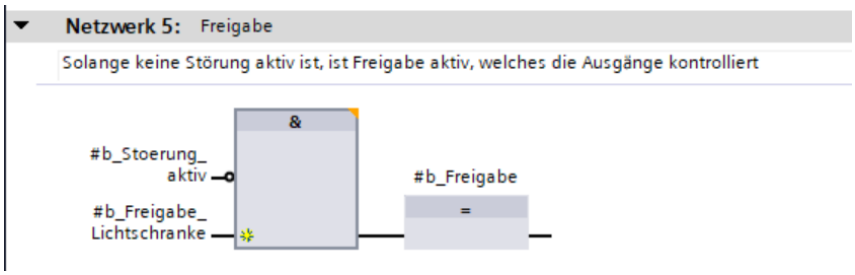


Abbildung 3.18: Setzen der Variable `b_Freigabe` in `FB11 - Störungen`

Jene Variable in Abbildung 3.18 wurde bereits in vorherigen Projekten verwendet. Sie wird im Block `FB_Ausgänge` eingesetzt, als zusätzliche invertierte Rücksetzbedingung. Somit ist ein fehlerfreier Betriebsmodus der einzelnen Aktoren gewährleistet.

3.4.1 Logische Abschaltbedingung der Lichtschranke

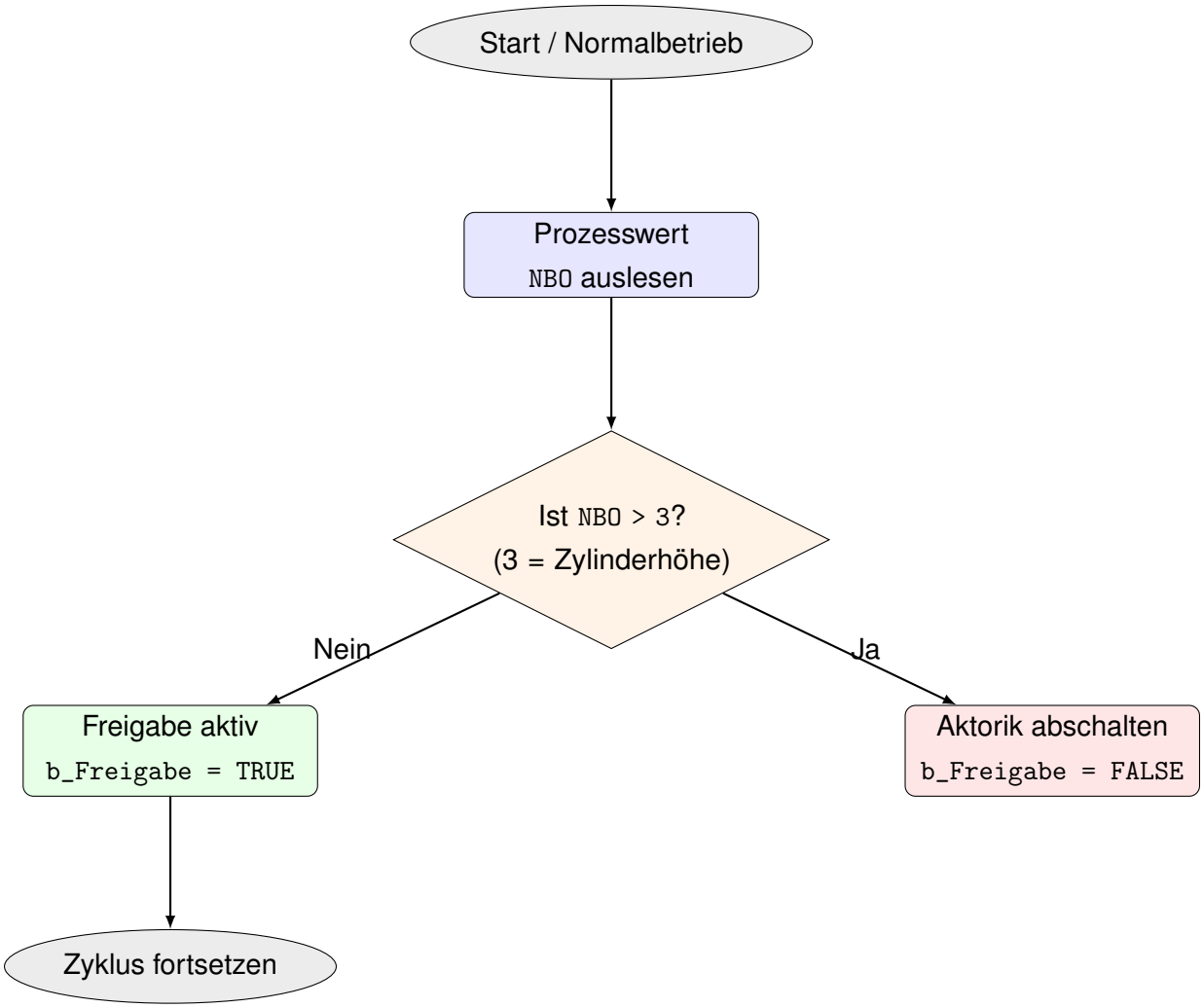


Abbildung 3.19: Abschaltlogik der Lichtschranke basierend auf dem Prozesswert `NBO`

In Abbildung 3.19 ist der Ablauf der Sicherheitsabschaltung grafisch dargestellt. Sobald der Prozesswert **NBO** größer als 3 ist, wird die Aktorik abgeschaltet. Das bedeutet, dass eine sofortige Abschaltung erfolgt, sobald zusätzlich zum Zylinder eine Hand oder ein anderes Objekt detektiert wird – also ein weiterer Lichtstrahl unterbrochen ist.

3.5 Visualisierung auf dem HMI

In diesem Kapitel wurde die Implementierung einer sicherheitsgerichteten Lichtschranke in das bestehende Hochregallager-System beschrieben. Dabei erfolgte zunächst die Analyse der Prozesswerte des IFM-Lichtgatters sowie die Auswahl des geeigneten Betriebsmodus zur sicheren Erkennung von Objekten im Gefahrenbereich. Anschließend wurde der vollständige Systemaufbau erläutert und die Anbindung über den IO-Link-Master an die Siemens-SPS umgesetzt. Die daraus resultierende Sicherheitslogik ermöglicht es, gefährliche Bewegungen im Fehlerfall automatisch zu stoppen und so die funktionale Sicherheit des Systems zu gewährleisten.

Die Visualisierung der Sicherheitszustände erfolgt über den integrierten Störungsbaustein des HMI-Panels, wodurch Unterbrechungen und Fehlermeldungen automatisch angezeigt werden. Eine Erweiterung dieser Funktion um eine gezielte Anzeige des Anlagenzustands (z. B. „Normalbetrieb“, „Sicherer Halt“) bietet zusätzliches Potenzial zur Verbesserung der Bedienerfreundlichkeit und Systemtransparenz.

Damit ist die sicherheitstechnische Basis des Hochregallagers geschaffen, auf der im folgenden Kapitel weitere Funktionen und Erweiterungen aufgebaut werden können.

4 Implementierung eines Umlagerungsprozesses

4.1 Definition „Umlagerung“ im Kontext der Intralogistik

In der Intralogistik bezeichnet eine Umlagerung die **interne Verlagerung von Material oder Ladeeinheiten innerhalb eines Lagersystems**, ohne dass ein Wareneinoder -ausgang erfolgt. Sie dient nicht der Erfüllung eines Kundenauftrags, sondern der **Optimierung der Lagerstruktur und der Betriebsabläufe**. Typische Gründe für Umlagerungen sind:

- **Platzoptimierung**: Zusammenführen von Teilpaletten oder das Freiräumen bestimmter Stellplätze für neue Einlagerungen.
- **Reduktion der Zykluszeiten**: Lagerware, die weit weg von der Entnahmestelle eingelagert wurde, wird in Totzeiten des Systems näher an die Entnahmestelle umgelagert. Dies verkürzt die Zykluszeit bei späterer Anfrage der Ware.
- **Strategische Positionierung**: Umlagerung von Artikeln in Bereiche mit höherer Zugriffshäufigkeit, um Kommissionierzeiten zu reduzieren.
- **Qualitätssicherung**: Verbringen von Waren in Quarantäne- oder Prüfzonen.
- **Technische Anforderungen**: Freihalten von Stellplätzen für Wartungsarbeiten oder zur Vermeidung von Blockaden.
- **Bestandskorrekturen**: Anpassungen nach Inventuren oder bei fehlerhaften Buchungen.

Im Gegensatz zu **Ein- und Auslagerungen**, die externe Materialflüsse abbilden, ist die Umlagerung ein **rein interner Prozess**.

4.2 Prozessablauf der Umlagerung

In diesem Kapitel wird der Funktionsablauf aus einer **systemorientierten Perspektive** betrachtet. Ziel ist es, die **Interaktion der beteiligten Komponenten** sowie die **Abhängigkeiten zwischen mechanischen Bewegungen, Steuerungslogik und Sicherheitsmechanismen** darzustellen. Darüber hinaus werden die wesentlichen Prozessschritte erläutert, die für einen **störungsfreien Betrieb** erforderlich sind. Die Darstellung erfolgt unabhängig von der konkreten Programmierung, um eine klare Trennung zwischen **konzeptioneller Beschreibung** und **technischer Umsetzung** zu gewährleisten.

Entwurf der Schrittkette für Umlagerungen

Die im Rahmen dieser Arbeit implementierte Umlagerungsfunktion basiert konzeptionell auf den bereits vorhandenen Schrittketten für die Einlagerungs- und Auslagerungsprozesse. Beide Abläufe bilden die Grundlage für die Umlagerung, da sie die wesentlichen Bewegungs- und Steuerungslogiken enthalten, die für den sicheren Transport von Ladeeinheiten erforderlich sind. Die Umlagerung kombiniert diese beiden Prozessketten in einer strukturierten Sequenz: Zunächst wird die Ladeeinheit am Quellplatz aufgenommen (analog zur Auslagerung), anschließend erfolgt die Verfahrbewegung zum Zielplatz und die Ablage (analog zur Einlagerung).
Allerdings wurde für die Realisierung der Umlagerungsfunktion eine eigenständige Schrittkette entwickelt. Dies war notwendig, da der Umlagerungsprozess im Vergleich zu den bestehenden Abläufen zusätzliche Anforderungen stellt: Zum einen sind mehr Schritte erforderlich, um die vollständige Sequenz von Aufnahme, Transport und Ablage abzubilden. Zum anderen müssen an zwei Stellen Verzweigungen integriert werden, um unterschiedliche Prozesspfade abhängig von den Betriebsbedingungen zu ermöglichen. Durch die Erstellung einer eigenen Schrittkette wird sichergestellt, dass die Logik klar strukturiert bleibt und die zusätzlichen Schritte sowie Verzweigungen ohne Beeinträchtigung der bestehenden Programme umgesetzt werden können.

Besonderheiten der Umlagerungsschrittkette

Die Umlagerungsschrittkette stellt eine Erweiterung der bestehenden Abläufe für Einlagerung und Auslagerung dar, basiert jedoch nicht ausschließlich auf deren Struktur. Während die Grundlogik der Bewegungen übernommen wurde, erfordert die Umlagerung zusätzliche Schritte und Verzweigungen, um die komplexeren Anforderungen dieses Prozesses abzubilden.

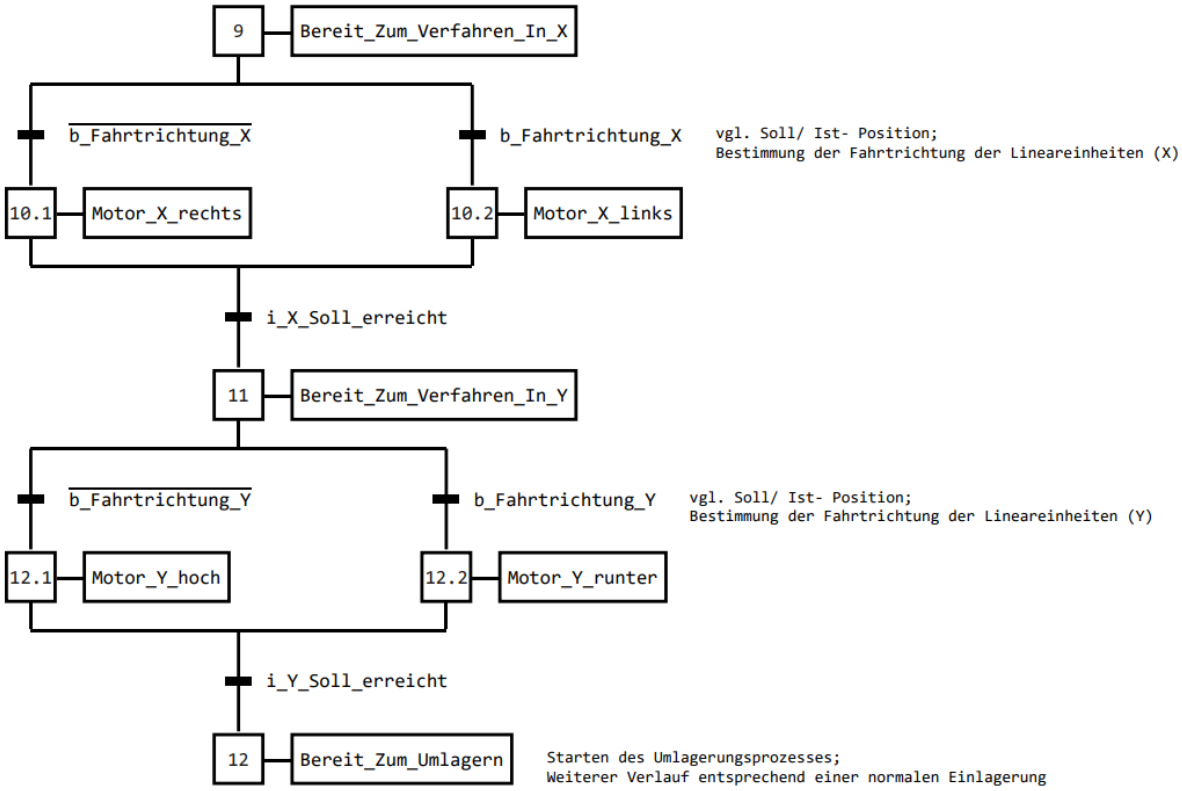


Abbildung 4.20: GRAFCET-Ausschnitt der Umlagerungsschrittkette mit zwei Verzweigungen zur Richtungsbestimmung

Im Gegensatz zu den linearen Schrittketten der Ein- und Auslagerung muss die Umlagerung sowohl die horizontale als auch die vertikale Fahrtrichtung dynamisch bestimmen. Dies wird im dargestellten GRAFCET deutlich: Nach der Initialisierung (*Bereit_Zum_Verfahren_In_X*) erfolgt die erste Verzweigung, bei der abhängig von der Soll-/Ist-Position die Fahrtrichtung der X-Achse gewählt wird (*Motor_X_rechts* oder *Motor_X_links*). Nach Erreichen der Zielposition wird analog die Y-Achse angesteuert, wobei ebenfalls eine Verzweigung zwischen *Motor_Y_hoch* und *Motor_Y_runter* erfolgt. Diese beiden Verzweigungen sind notwendig, um die Umlagerung flexibel zwischen beliebigen Lagerplätzen auszuführen. Aufgrund der zusätzlichen Schritte und der logischen Verzweigungen wurde eine eigenständige Schrittkette entwickelt, anstatt die bestehenden Abläufe lediglich zu kombinieren. Dadurch bleibt die Programmstruktur übersichtlich, und die Steuerungslogik kann die erhöhten Anforderungen hinsichtlich

Prozesssicherheit und Flexibilität erfüllen.

4.3 Umsetzung im TIA Portal

Im Rahmen der Automatisierung der Modellhochregalanlage wurde eine Umlagerfunktion implementiert, die den Transport von Ladeeinheiten zwischen unterschiedlichen Lagerplätzen ermöglicht. Zur Realisierung dieser Funktion wurde eine Schrittkette eingesetzt, da sie eine strukturierte und übersichtliche Abbildung sequenzieller Abläufe erlaubt. Die Schrittkette dient hierbei als Steuerungslogik, um die einzelnen Prozessschritte – vom Anfahren des Quellplatzes über die Aufnahme der Ladeeinheit bis hin zur Ablage am Zielplatz – in einer definierten Reihenfolge auszuführen. Durch diese Vorgehensweise wird eine hohe Prozesssicherheit gewährleistet und die Komplexität der Steuerung reduziert.

4.3.1 Aufruf und Funktionsprinzip der Positionsvergleichsfunktion

```
// Aufruf der Positions- Vergleichsfunktion für den Umlagerungsprozess
IF #i_Schrittketten_Nr_Umlagerung = 3 AND #b_Merker_m THEN
    "Vergleiche_Fahrweg"(i_Lagerplatz_alt_X := #i_Sollwert_X1_m,
                         i_Lagerplatz_neu_X := #i_Sollwert_X2_m,
                         i_Lagerplatz_alt_Y := #i_Sollwert_Y1_m,
                         i_Lagerplatz_neu_Y := #i_Sollwert_Y2_m,
                         b_Fahrtrichtung_X => #b_Fahrtrichtung_X,
                         b_Fahrtrichtung_Y => #b_Fahrtrichtung_Y);
ELSIF #i_Schrittketten_Nr_Umlagerung = 3 AND #b_Merker_ue THEN
    "Vergleiche_Fahrweg"(i_Lagerplatz_alt_X := #i_Sollwert_X1_ue,
                         i_Lagerplatz_neu_X := #i_Sollwert_X2_ue,
                         i_Lagerplatz_alt_Y := #i_Sollwert_Y1_ue,
                         i_Lagerplatz_neu_Y := #i_Sollwert_Y2_ue,
                         b_Fahrtrichtung_X => #b_Fahrtrichtung_X,
                         b_Fahrtrichtung_Y => #b_Fahrtrichtung_Y);
END_IF;
```

Abbildung 4.21: Aufruf des Positionsvergleichs

Zur Bestimmung der für die Umlagerung notwendigen Fahrwege wird eine Positionsvergleichsfunktion (`Vergleiche_Fahrweg`) aufgerufen, welche aus den Sollwerten der Quell- (P_1) und Zielposition (P_2) die Differenzen in den kartesischen Achsen berechnet und daraus die jeweilige Fahrtrichtung ableitet. Der Aufruf erfolgt in Abhängigkeit des aktuellen Umlagerungskontextes (manueller Modus und Übersichts-Modus), wobei die Eingänge als Sollwertpaare für X und Y übergeben werden und die Ausgänge als Richtungsflags bereitgestellt werden.

Formal seien die Sollwerte der Positionen gegeben durch

$$P_1 = (X_1,\, Y_1), \qquad P_2 = (X_2,\, Y_2).$$

Die Funktion berechnet die Restwege (Differenzen)

$$\Delta X = X_2 - X_1, \qquad \Delta Y = Y_2 - Y_1,$$

und leitet hieraus die Richtungsentscheidung für die Verfahrbewegungen ab:

$$b\_\text{Fahrtrichtung\_X} = \begin{cases} \texttt{TRUE} & \text{falls } \Delta X > 0 \quad (\text{Fahrt nach rechts}) \\ \texttt{FALSE} & \text{falls } \Delta X < 0 \quad (\text{Fahrt nach links}) \end{cases}$$

$$b\_\text{Fahrtrichtung\_Y} = \begin{cases} \texttt{TRUE} & \text{falls } \Delta Y > 0 \quad (\text{Fahrt nach oben}) \\ \texttt{FALSE} & \text{falls } \Delta Y < 0 \quad (\text{Fahrt nach unten}) \end{cases}$$

Für den Grenzfall $\Delta X = 0$ bzw. $\Delta Y = 0$ wird keine Bewegungsanforderung in der jeweiligen Achse generiert, womit die Schrittfolge unmittelbar zur nächsten Achsenbewegung bzw. zum nachfolgenden Prozessschritt übergeht.
Der Funktionsaufruf ist in zwei Varianten ausgeführt, die jeweils an den aktiven Modus (`b_Merker_m` bzw. `b_Merker_ue`) gebunden sind. In beiden Fällen werden als Eingangsparameter die Sollwerte von Quell- und Zielplatz übergeben; die Ausgänge setzen die Richtungsflags:
Die Ausgabeparameter sind in beiden Fällen identisch und werden den nachgeschalteten GRAFCET-Verzweigungen zur Richtungswahl zugeführt:

$$b\_\text{Fahrtrichtung\_X} \Rightarrow \begin{cases} \texttt{Motor\_X\_rechts} & \text{bei } \Delta X > 0 \\ \texttt{Motor\_X\_links} & \text{bei } \Delta X < 0 \end{cases}$$

$$b\_\text{Fahrtrichtung\_Y} \Rightarrow \begin{cases} \texttt{Motor\_Y\_hoch} & \text{bei } \Delta Y > 0 \\ \texttt{Motor\_Y\_runter} & \text{bei } \Delta Y < 0 \end{cases}$$

Damit stellt die Positionsvergleichsfunktion die für die Verzweigungen in X- und Y-Richtung notwendigen Entscheidungsgrößen bereit und gewährleistet eine konsistente, zustandsabhängige Ansteuerung der Lineareinheiten. Die klare Trennung von Berechnung ($\Delta X, \Delta Y$) und Aktorik-Auswahl (Richtungsflags) erhöht die Wartbarkeit und unterstützt die modulare Erweiterbarkeit der Schrittkette.

4.3.2 Schritt 10.1 im Kontext der Schrittkette

Die in Abbildung 4.20 dargestellte GRAFCET-Struktur verdeutlicht die logische Verzweigung innerhalb der Umlagerungsschrittkette. Nach dem Schritt *Bereit_Zum_Verfahren_In_X* (Schritt 9) erfolgt die Entscheidung über die Fahrtrichtung der X-Achse. Diese Entscheidung basiert auf dem Ergebnis der zuvor aufgerufenen Positionsvergleichsfunktion (`Vergleiche_Fahrweg`), die die Differenz zwischen Quell- und Zielposition berechnet und die Richtungsflags `b_Fahrtrichtung_X` und `b_Fahrtrichtung_Y` setzt.
Die Verzweigung in Schritt 10 ist wie folgt aufgebaut:

- **Schritt 10.1 – Motor_X_rechts:** Wird das Flag `b_Fahrtrichtung_X` auf `TRUE` gesetzt, erfolgt die Ansteuerung des Motors für die Bewegung in positiver X-Richtung.
- **Schritt 10.2 – Motor_X_links:** Ist das Flag `b_Fahrtrichtung_X` `FALSE`, wird der Motor für die Bewegung in negativer X-Richtung aktiviert.

Die Umsetzung in der SPS-Logik ist in Abbildung 4.22 dargestellt. Hier wird die Schrittnummer über ein `MOVE`-Befehl gesetzt, sobald die Aktivierungsbedingungen erfüllt sind. Die logische Verknüpfung erfolgt über ein Set-Reset-Glied (`SR`), das den Schrittstatus verwaltet. Die Aktivierung von Schritt 10.1 ist somit direkt an die Fahrtrichtungsentscheidung gekoppelt, die aus der Positionsvergleichsfunktion resultiert.

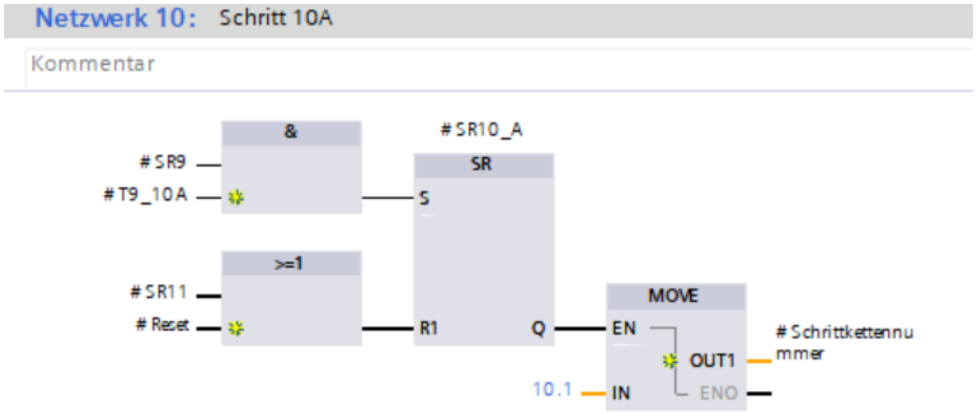


Abbildung 4.22: SPS-Implementierung für Schritt 10.1 (Motor_X_rechts)

Wie in der Abbildung ersichtlich, wird die Aktivierung der Schrittnummer `10.1` durch die Transitionsbedingung `T9_10A` ausgelöst. Diese Bedingung wird erfüllt, sobald die Fahrtrichtung in positiver X-Achse erforderlich ist, wie zuvor durch die Positionsvergleichsfunktion bestimmt. Analog dazu wird bei aktiver Transitionsbedingung `T9_10B` die Schrittnummer `10.2` aktiviert, wodurch die Bewegung in negativer X-Achse erfolgt.
Durch diese logische Struktur entsteht eine klare und funktionale Verzweigung innerhalb der Schrittkette. Sie ermöglicht eine dynamische Auswahl des Prozesspfades

in Abhängigkeit der berechneten Richtungsflags und trägt somit zur Flexibilität und Modularität des Steuerungsprogramms bei. Analog dazu wird mit Schritt 12.1 bzw. 12.2 die Fahrtrichtung der y-Achse festgelegt.

4.3.3 Lagerplatzmanagement im Umlagerungsprozess

Das Lagerplatzmanagement stellt sicher, dass die Umlagerung nur auf freie Zielplätze erfolgt und die Bestandsdaten konsistent aktualisiert werden. Die Umsetzung erfolgt in zwei logischen Schritten, die in den folgenden Abbildungen dargestellt sind.

```
IF #i_Schrittketten_Nr_Umlagerung = 2 THEN
    //aus der Übersicht
    IF #b_Merker_ue = TRUE THEN
        // Platz belegt, je nach Status, Setzen oder Rücksetzen
        IF "DB_Lagerbestand".Lagerbestand.a_Gesamtlager[#i_Sollwert_X2_ue, #i_Sollwert_Y2_ue].b_Status = False THEN
            #b_Platzwahl_abgeschlossen := TRUE;
            #b_Lagerplatz_belegt := FALSE;
        ELSE
            #b_Lagerplatz_belegt := TRUE;
            #b_Platzwahl_abgeschlossen := FALSE;
        END_IF;
```

Abbildung 4.23: Prüfung des Lagerplatzstatus und Setzen der Platzwahl

Die erste Abbildung zeigt die Prüfung des Lagerplatzstatus im Kontext der Schrittkette. Sobald die Schrittnummer für die Umlagerung (`i_Schrittketten_Nr_Umlagerung`) den Wert `2` erreicht, wird die Platzwahl aus der Übersicht aktiviert. Hierbei wird überprüft, ob der Zielplatz im Datenbaustein `DB_Lagerbestand` als belegt oder frei gekennzeichnet ist. Ist der Status `False`, wird die Variable `b_Platzwahl_abgeschlossen` auf `TRUE` gesetzt und der Platz als frei markiert (`b_Lagerplatz_belegt := FALSE`). Andernfalls wird der Platz als belegt gekennzeichnet und die Platzwahl bleibt unvollständig. Diese Logik verhindert fehlerhafte Umlagerungen auf bereits belegte Positionen.

Netzwerk 6: Lagerberstand update Umlagerung Teil 1

Kommentar

```
1  IF (#b_manuelle_Umlagerung_aktiv AND #b_update_Lagerbestand_DB_vor_Umlagerung) THEN
2
3      "DB_Lagerbestand".Lagerbestand.a_Gesamtlager[#i_Soll_Position_X1_m_umlagern, #i_Soll_Position_Y1_m_umlagern].b_Status := FALSE;
4      "DB_Lagerbestand".Lagerbestand.a_Gesamtlager[#i_Soll_Position_X1_m_umlagern, #i_Soll_Position_Y1_m_umlagern].di_UID_Teil1 := #di_leerer_DINT;
5      "DB_Lagerbestand".Lagerbestand.a_Gesamtlager[#i_Soll_Position_X1_m_umlagern, #i_Soll_Position_Y1_m_umlagern].di_UID_Teil2 := #di_leerer_DINT;
6      "DB_Lagerbestand".Lagerbestand.a_Gesamtlager[#i_Soll_Position_X1_m_umlagern, #i_Soll_Position_Y1_m_umlagern].s_Artikelbeschreibung := '--';
7      "DB_Lagerbestand".Lagerbestand.a_Gesamtlager[#i_Soll_Position_X1_m_umlagern, #i_Soll_Position_Y1_m_umlagern].s_Artikelnummer := '--';
8      "DB_Lagerbestand".Lagerbestand.a_Gesamtlager[#i_Soll_Position_X1_m_umlagern, #i_Soll_Position_Y1_m_umlagern].Zeitstempel := #dt_default_Zeitstempel;
9
10 ELSIF (#b_uebersicht_Umlagerung_aktiv AND #b_update_Lagerbestand_DB_vor_Umlagerung) THEN
11     "DB_Lagerbestand".Lagerbestand.a_Gesamtlager[#i_Soll_Position_X1_ue_umlagern, #i_Soll_Position_Y1_ue_umlagern].b_Status := FALSE;
12     "DB_Lagerbestand".Lagerbestand.a_Gesamtlager[#i_Soll_Position_X1_ue_umlagern, #i_Soll_Position_Y1_ue_umlagern].di_UID_Teil1 := #di_leerer_DINT;
13     "DB_Lagerbestand".Lagerbestand.a_Gesamtlager[#i_Soll_Position_X1_ue_umlagern, #i_Soll_Position_Y1_ue_umlagern].di_UID_Teil2 := #di_leerer_DINT;
14     "DB_Lagerbestand".Lagerbestand.a_Gesamtlager[#i_Soll_Position_X1_ue_umlagern, #i_Soll_Position_Y1_ue_umlagern].s_Artikelbeschreibung := '--';
15     "DB_Lagerbestand".Lagerbestand.a_Gesamtlager[#i_Soll_Position_X1_ue_umlagern, #i_Soll_Position_Y1_ue_umlagern].s_Artikelnummer := '--';
16     "DB_Lagerbestand".Lagerbestand.a_Gesamtlager[#i_Soll_Position_X1_ue_umlagern, #i_Soll_Position_Y1_ue_umlagern].Zeitstempel := #dt_default_Zeitstempel;
17
18 END_IF;
```

Abbildung 4.24: Aktualisierung des Lagerbestands nach erfolgreicher Umlagerung

Die zweite Abbildung zeigt die Aktualisierung des Lagerbestands nach erfolgreicher

Umlagerung. Hierbei werden die Daten des Quellplatzes zurückgesetzt und die Zielposition mit den neuen Informationen beschrieben. Dies umfasst die Statusvariable (`b_Status`), die eindeutige Identifikationsnummer (`di_UID_Teil`), die Artikelnummer sowie den Zeitstempel. Die Logik unterscheidet zwischen manueller Umlagerung (`b_manuelle_Umlagerung_aktiv`) und Umlagerung aus der Übersicht (`b_uebersicht_Umlagerung_akt`) wobei in beiden Fällen die Datenbankeinträge konsistent angepasst werden. Durch diese Vorgehensweise wird sichergestellt, dass die Bestandsdaten jederzeit aktuell sind und die Nachverfolgbarkeit der Umlagerungen gewährleistet bleibt.

4.4 Visualisierung auf dem HMI

Die Visualisierung des Umlagerungsprozesses wurde im bestehenden Stil des bisherigen HMI-Layouts umgesetzt, um eine konsistente Benutzerführung und ein einheitliches Erscheinungsbild zu gewährleisten. Sämtliche grafischen Elemente, Farbschemata und Bedienelemente orientieren sich an der zuvor implementierten Struktur für Ein- und Auslagerungsprozesse. Ergänzend wurde das HMI um spezifische Funktionen für die Umlagerung erweitert. Dazu zählen die Anzeige des aktuellen Quell- und Zielplatzes, die Positionsmeldungen während der Verfahrbewegungen sowie die Möglichkeit zur direkten Auswahl der Umlagerungsposition im Menü Übersicht.

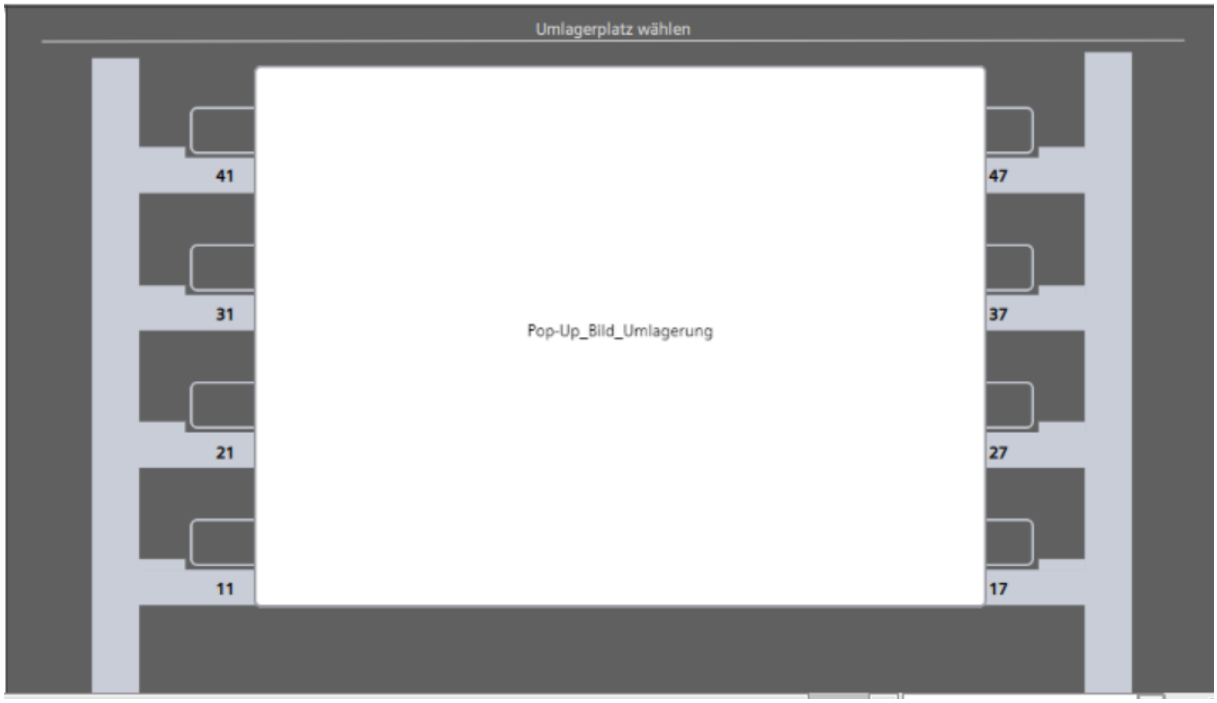


Abbildung 4.25: Übersichtsdarstellung zur Auswahl des Umlagerplatzes im HMI

Oben abgebildet ist die Übersicht des Hochregallagers mit den einzelnen Lagerplätzen, die in einer vertikalen Anordnung dargestellt sind. Die Platznummern (z. B. 11, 21, 31, 41) setzen sich aus den X- und Y-Koordinaten zusammen , um eine schnelle

Orientierung zu gewährleisten. Im Zentrum befindet sich ein Popup-Bereich für die Auswahl des Umlagerplatzes. Das Pop-Up wird geöffnet, sobald der Bediener der Anlage einen Lagerplatz anklickt und ihn dadurch auswählt.

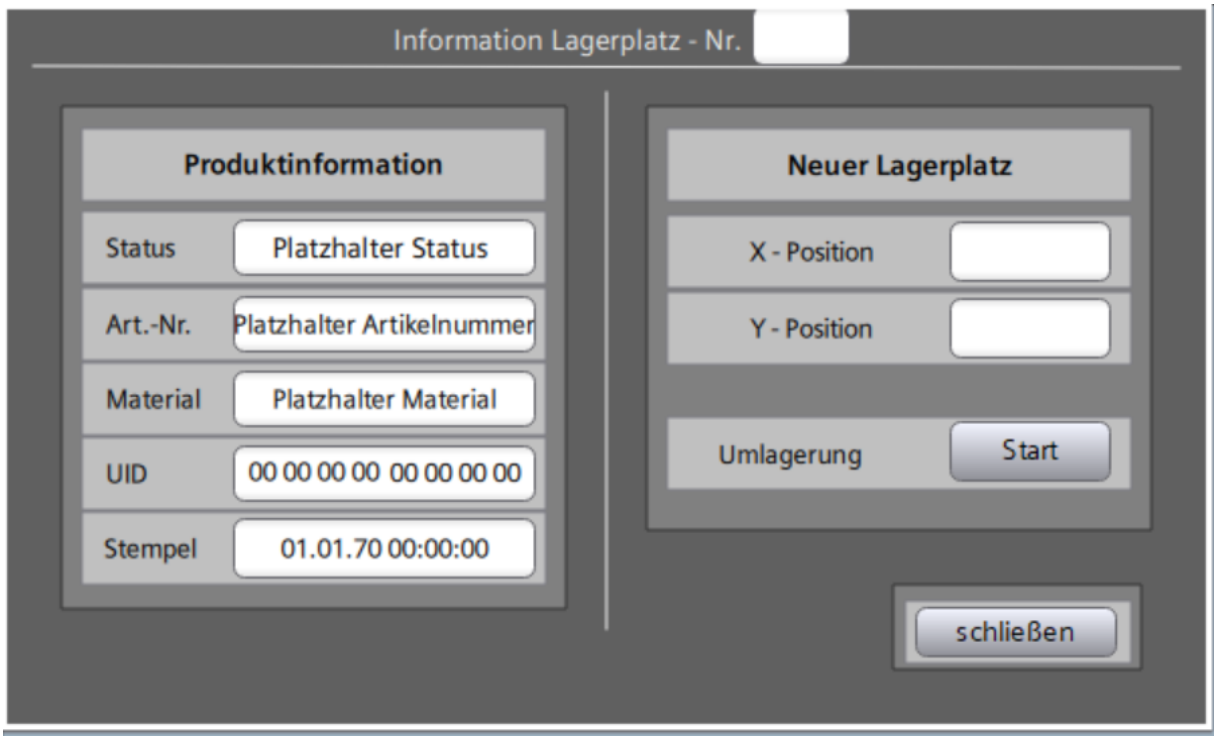


Abbildung 4.26: Popup-Fenster zur Anzeige von Produktinformationen und Eingabe des neuen Lagerplatzes

Das Bild zeigt das Popup-Fenster, das zur Anzeige detaillierter Produktinformationen sowie zum Start des neuen Umlagerprozesses dient. Auf der linken Seite werden Statusinformationen des ausgewählten Lagerplatzes dargestellt, darunter der aktuelle Status, die Artikelnummer, das Material, die eindeutige Identifikationsnummer (UID) sowie ein Zeitstempel. Die rechte Seite des Fensters zeigt den zuvor in der Übersicht ausgewählten Koordinaten für die neue Zielposition (X- und Y-Koordinaten) sowie eine Schaltfläche zur Auslösung des Umlagerungsprozesses. Diese Struktur ermöglicht eine klare Trennung zwischen Informationsanzeige und Steuerungsfunktion.

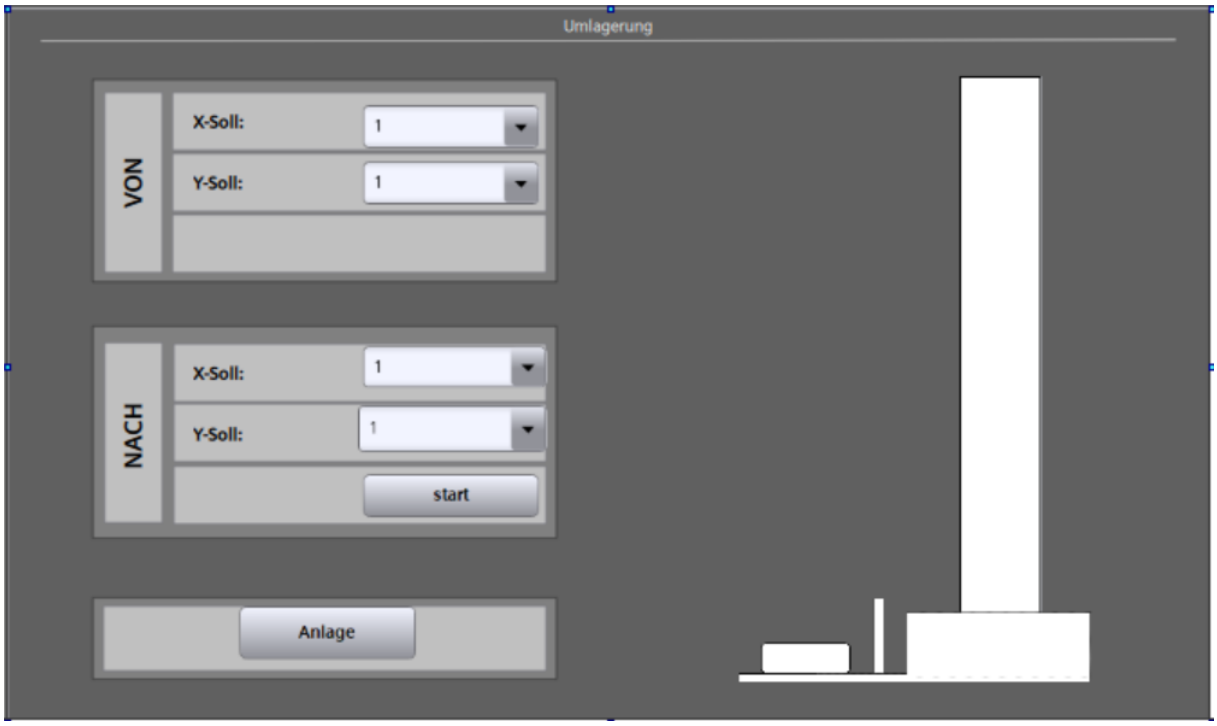


Abbildung 4.27: Manuelles Menü des Umlagerungsprozesses mit Anzeige der aktuellen Position

Abgebildet ist die Hauptansicht für die Durchführung der Umlagerung. Hier werden die Quell- und Zielpositionen in separaten Eingabebereichen dargestellt: Der obere Bereich (*Von*) dient zur Eingabe der aktuellen Position (X-Soll, Y-Soll), während der untere Bereich (*Nach*) die Zielposition aufnimmt. Darunter befindet sich eine Schaltfläche zur Auslösung des Umlagerungsprozesses. Rechts ist eine schematische Darstellung der Anlage integriert, die den räumlichen Bezug zum Hochregallager verdeutlicht. Diese Ansicht ermöglicht eine intuitive Bedienung und stellt alle relevanten Parameter für die Umlagerung kompakt dar.

5 Fazit und Ausblick

5.1 Fazit

Im Rahmen dieser Studienarbeit wurde das automatisierte Modell-Hochregallager der Dualen Hochschule Baden-Württemberg gezielt weiterentwickelt und um wesentliche funktionale sowie sicherheitstechnische Aspekte ergänzt. Ziel war es, bestehende Defizite im Bereich der Sicherheit, der Programmstruktur und der Prozesslogik zu analysieren und durch geeignete technische Maßnahmen zu beheben. Die durchgeführten Arbeiten leisten einen wichtigen Beitrag zur funktionalen Erweiterung und zur praxisnahen Ausgestaltung des Systems.
Ein zentraler Schwerpunkt der Arbeit lag auf der Integration einer berührungslos wirkenden Schutzeinrichtung in Form eines Lichtgatters. Durch die Implementierung der Lichtschranke in Verbindung mit einem IO-Link-fähigen Sensor konnte eine zuverlässige Überwachung des Gefahrenbereichs realisiert werden. Die detaillierte Analyse der verfügbaren Prozesswerte sowie die gezielte Auswahl des Modus „Number of Beams Occupied (NBO)“ ermöglichen eine klare Unterscheidung zwischen zulässigen Prozessbewegungen, wie dem Einfahren des Zylinders, und unzulässigen Eingriffen in den Arbeitsbereich. Dadurch wird eine situationsabhängige und zugleich robuste Sicherheitsabschaltung gewährleistet.
Die sicherheitstechnische Logik wurde vollständig in die bestehende SPS-Architektur integriert und über einen eigenen Funktionsbaustein realisiert. Durch die zentrale Freigabevariable wird sichergestellt, dass sämtliche Aktoren des Systems nur dann angesteuert werden, wenn keine Störung vorliegt und der Gefahrenbereich frei ist. Dieses Vorgehen erhöht nicht nur die Sicherheit des Systems, sondern verbessert auch die Nachvollziehbarkeit und Wartbarkeit der Steuerungssoftware. Die Kopplung der Sicherheitsfunktion mit dem vorhandenen Störungsmanagement stellt sicher, dass sicherheitsrelevante Zustände eindeutig erkannt, verarbeitet und visualisiert werden.
Ein weiterer wesentlicher Bestandteil der Arbeit war die Einführung eines Umlagerungsprozesses. Im Gegensatz zu den bereits vorhandenen Ein- und Auslagerungsfunktionen handelt es sich hierbei um einen rein internen logistischen Prozess, der zur Optimierung der Lagerstruktur dient. Die Entwicklung einer eigenständigen Schrittkette erwies sich als notwendig, um die zusätzlichen Anforderungen hinsichtlich Bewegungslogik, Verzweigungen und Prozesssicherheit abzubilden. Durch die Kombination bestehender Grundlogiken mit neuen Entscheidungsstrukturen konnte ein flexibler und nachvollziehbarer Umlagerungsablauf realisiert werden.
Darüber hinaus zeigt die Arbeit deutlich die Vorteile der textbasierten Programmierung in Structured Text (ST) gegenüber grafischen Programmiersprachen wie FUP. Insbesondere bei komplexeren Abläufen, sicherheitsrelevanten Abfragen und mathematischen

Vergleichen bietet ST eine deutlich höhere Übersichtlichkeit und Skalierbarkeit. Die Umstellung einzelner Funktionsbereiche auf ST trägt langfristig zu einer besseren Lesbarkeit, einfacheren Fehlersuche und verbesserten Erweiterbarkeit des Programms bei.
Zusammenfassend lässt sich festhalten, dass die in dieser Studienarbeit umgesetzten Maßnahmen die funktionale Sicherheit, Prozessstabilität und didaktische Qualität des Modell-Hochregallagers deutlich verbessern. Das System bildet nun nicht nur grundlegende logistische Prozesse ab, sondern berücksichtigt auch sicherheitstechnische Anforderungen, wie sie in realen industriellen Anlagen zwingend erforderlich sind. Damit stellt das Hochregallager eine praxisnahe und zukunftsfähige Lernplattform für Studierende im Bereich der Automatisierungstechnik dar.

5.2 Ausblick

Die im Rahmen dieser Studienarbeit umgesetzten Erweiterungen bilden eine stabile technische und konzeptionelle Grundlage für die weitere Entwicklung des automatisierten Modell-Hochregallagers. Insbesondere durch die Integration der sicherheitsgerichteten Lichtschranke, die Einführung des Umlagerungsprozesses sowie die zunehmende Nutzung der Programmiersprache Structured Text wurde das System sowohl funktional als auch strukturell deutlich verbessert. Aufbauend auf diesen Ergebnissen wird sich die darauffolgende Arbeit auf die Implementierung eines auftragsbasierten Materialflusses konzentrieren.
Der zentrale Fokus der nächsten Entwicklungsstufe liegt auf der Einführung eines Auftragsmanagements mithilfe eines QR-Code-Readers. Ziel ist es, Ein- und Auslagerprozesse nicht mehr ausschließlich manuell oder fest vorgegeben auszulösen, sondern diese auf Basis eindeutig definierter Aufträge automatisch zu steuern. Hierdurch soll das Hochregallager stärker an reale industrielle Anwendungen angelehnt werden, in denen Materialbewegungen in der Regel durch digitale Aufträge aus übergeordneten Systemen angestoßen werden.
Im Rahmen der nächsten Arbeit soll ein QR-Reader als zusätzliche Identifikations- und Eingabeschnittstelle in das bestehende System integriert werden. Über den QR-Code sollen auftragsrelevante Informationen wie Lagerplatz, Material-ID, Prozessart (Einlagern, Auslagern oder Umlagern) sowie gegebenenfalls Prioritäten oder Zielpositionen erfasst werden. Die ausgelesenen Daten werden anschließend von der SPS verarbeitet und in strukturierter Form an die Prozesssteuerung übergeben. Damit entsteht eine direkte Verbindung zwischen physischer Auftragserfassung und automatisierter Anlagensteuerung.
Ein wesentlicher Bestandteil des Auftragsmanagements wird die Definition geeigneter Datenstrukturen innerhalb der SPS sein. Hierzu zählen unter anderem Auftragsobjekte, Warteschlangen sowie Statusinformationen zur Verfolgung des aktuellen Bearbeitungsstands. Durch die Verwendung strukturierter Datentypen in Structured Text können Aufträge klar definiert, verwaltet und sequenziell abgearbeitet werden. Dies ermöglicht eine saubere Trennung zwischen Auftragsebene und Bewegungs- bzw. Sicherheitslogik und erhöht die Modularität des Gesamtsystems.
Darüber hinaus bietet ein auftragsbasiertes System die Möglichkeit, bestehende Prozesse wie Einlagerung, Auslagerung und Umlagerung in einer übergeordneten Logik zusammenzuführen. Anstatt einzelne Funktionen separat auszulösen, kann der Anlagenbetrieb künftig vollständig über Aufträge gesteuert werden. Dies eröffnet Potenzial für eine automatische Priorisierung von Aufträgen, eine flexible Reihenfolge der Abarbeitung sowie eine bessere Auslastung der Anlage. Insbesondere in Kombination mit dem bereits implementierten Umlagerungsprozess lassen sich so optimierte Lagerstrategien

realisieren.
Ein weiterer Aspekt der kommenden Arbeit betrifft die Erweiterung der Bedien- und Visualisierungsebene. Die über den QR-Reader eingelesenen Aufträge sollen auf dem HMI übersichtlich dargestellt werden, inklusive Informationen zum aktuellen Auftragsstatus, zur Restbearbeitungszeit und zu möglichen Störungen. Dadurch wird die Transparenz für den Bediener erhöht und der Anlagenzustand jederzeit nachvollziehbar dargestellt. Ergänzend könnte eine Historie abgeschlossener Aufträge implementiert werden, um Prozessabläufe analysieren und optimieren zu können.
Auch im Hinblick auf die Sicherheit ergeben sich neue Anforderungen. Das Auftragsmanagement muss eng mit den bestehenden Sicherheitsfunktionen verknüpft werden, sodass Aufträge bei aktiven Störungen oder unterbrochenen Schutzfunktionen nicht gestartet oder automatisch pausiert werden. Dadurch wird sichergestellt, dass der auftragsbasierte Betrieb jederzeit den sicherheitstechnischen Anforderungen entspricht und keine gefährlichen Zustände entstehen.
Zusammenfassend stellt die Implementierung eines QR-basierten Auftragsmanagements einen konsequenten nächsten Schritt in der Weiterentwicklung des automatisierten Modell-Hochregallagers dar. Sie ermöglicht eine realitätsnahe Abbildung moderner intralogistischer Prozesse und schafft die Grundlage für eine flexible, skalierbare und industrieorientierte Anlagensteuerung. Die Ergebnisse der vorliegenden Arbeit bilden hierfür eine solide Basis und ermöglichen es, das System in der folgenden Theoriephase gezielt um eine zentrale Funktion moderner Lagerlogistik zu erweitern.

Literatur

[1] Adam, H. J. und Adam, M. *PLC Programming In Instruction List According To IEC 61131-3*. Springer Nature, 2022. ISBN: 978-3-662-65254-1.

[2] Automation, International Society of. „IO-Link Field Device Protocol Architecture“. In: *InTech Magazine* (2016). Zugriff am 25. Oktober 2025. URL: https://www.isa.org/intech-home/2016/may-june/features/io-link-field-device-protocol-architecture.

[3] Baumann, P. *Ausgewählte Sensorschaltungen*. Springer Fachmedien Wiesbaden GmbH, 2022. ISBN: 978-3-658-35707-8.

[4] Bernstein, H. *Sicherheits- und Antriebstechnik*. Springer-Verlag GmbH Deutschland, 2016. ISBN: 978-3-658-12934-7.

[5] e.V., PROFIBUS Nutzerorganisation. *IO-Link Systembeschreibung*. Zugriff am 02. November 2025. IO-Link. 2024. URL: https://www.io-link.com.

[6] GmbH, DriveCon. *FeldBustopologien*. Zugriff am 02. November 2025. 2025. URL: https://www.drivecon.de/de/lexikon/feldbus.

[7] gmbh, ifm electronic. *Betriebsanleitung messendes Lichtgitter OY5100*. Zugriff am 06. November 2025. 2021. URL: https://www.ifm.com/de/de/product/OY5100#documents.

[8] gmbh, ifm electronic. *Datenblatt messendes Lichtgitter OY5100*. Zugriff am 06. November 2025. 2021. URL: https://www.ifm.com/de/de/product/OY5100#documents.

[9] gmbh, ifm electronic. *IO-Link Master mit PROFINET-Schnittstelle AL1100*. Zugriff am 02. November 2025. 2025. URL: https://www.ifm.com/de/de/product/AL1100#documents.

[10] Halang, W. A. *funktionale Sicherheit - Echtzeit 2013*. Springer Viewing, 2013. ISBN: 978-3-642-41309-4.

[11] Hering, E., Endres, J. und Gutekunst, J. *Elektronik für Ingenieure und Naturwissenschaftler*. Springer-Verlag GmbH Deutschland, 2021. ISBN: 978-3-662-62698-6.

[12] *IEC 61131-3:2025 – Programmable controllers – Part 3: Programming languages*. Standard. Zugriff am 04.11.2025. Geneva, Switzerland: International Electrotechnical Commission, 2025. URL: https://webstore.iec.ch/en/publication/68533.

[13] Kanngießer, U. *Programmierung mit Strukturierter Text: Steuerungs-Funktionsbausteine mit ST oder SCL einfach und schnell erstellen. Für Ein- und AWL-Umsteiger - Softcover*. VDE Verlag GmbH, 2017. ISBN: 978-3-800-74409-1.

[14] Langmann, R. *Vernetzte Systeme für die Automatisierung 4.0*. 1. Auflage. Carl Hanser Verlag München, 2021. ISBN: 978-3-446-46984-6.

[15] Schnell, G. und Wiedemann, B. *Bussysteme in der Automatisierungs- und Prozesstechnik*. 9. Auflage. Springer-Verlag GmbH Deutschland, 2019. ISBN: 978-3-658-23688-5.

[16] Seitz, M. *Speicherprogrammierbare Steuerungen in der Industrie 4.0*. 5. Auflage. Carl Hanser Verlag München, 2021. ISBN: 978-3-446-47002-6.

[17] Siemens AG. *PLC Programmieren*. Zugriff am 27. Oktober 2025. 2017. URL: https://support.industry.siemens.com/cs/mdm/109747174?c=98343050763&lc=de-DE.

[18] Siemens AG. *Beispiel einer Variablentabelle im TIA Portal*. Zugriff am 30. Juni 2025. 2024. URL: https://support.industry.siemens.com/cs/document/37855612/warum-wird-in-der-plc-variablentabelle-der-name-einer-variablen-(z-b-on-)-mit-einer-nummer-erweitert-(z-b-on_1-)-?dti=0&lc=de-DE.

[19] Siemens AG. *SIMATIC S7-1500 – Kontaktseite*. Zugriff am 04.11.2025. 2025. URL: https://www.siemens.com/de/de/produkte/automatisierung/systeme/industrie/sps/simatic-s7-1500/kontakt.html.

[20] Studieninstitut, Deutsches eLearning. *E-V-A-Prinzip*. Zugriff am 04. November 2025. 2025. URL: https://www.delst.de/de/lexikon/e-v-a-prinzip/.

[21] Wein, Marcel. *SPS vs. VPS: Was ist der Unterschied? Verständlich erklärt*. Zugriff am 04.11.2025. 2025. URL: https://weinsolutions.de/sps-vs-vps-was-ist-der-unterschied-verstaendlich-erklaert/.

[22] Wellenreuther, Günter und Zastrow, Dieter. *Automatisieren mit SPS – Theorie und Praxis*. 6., korrigierte Auflage. Zugriff am 04.11.2025. Wiesbaden: Springer Vieweg, 2015. ISBN: 978-3-8348-2597-1. URL: https://link.springer.com/book/10.1007/978-3-8348-9018-4.