Effizienzsteigerung der Abläufe in der Zukunftsfabrik Bodensee

Studienarbeit T4 3200

des Studienganges Elektrotechnik Automation

an der Dualen Hochschule Baden-Württemberg Ravensburg

Campus Friedrichshafen

Von

Fabian Wulf und Nicklas Fischer

12.07.2026

Bearbeitungszeitraum	07.04.2026 – 12.07.2026
Matrikelnummer, Name, Kurs	9556860, Fabian Wulf, TEA23
	1748871, Nicklas Fischer, TEA23
Betreuer der Studienarbeit	Prof. Dr. Thomas Dietmüller

Erklärung zur Eigenleistung

gemäß Ziffer 1.1.13 der Anlage 1 zu §§ 3, 4 und 5 der Studien- und Prüfungsordnung für die Bachelorstudiengänge im Studienbereich Technik der Dualen Hochschule Baden-Württemberg vom 29.09.2017.

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

Effizienzsteigerung der Abläufe in der Zukunftsfabrik Bodensee

selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Ravensburg, den 12.07.2026



Fabian Wulf Nicklas Fischer

Kurzfassung

Die vorliegende Studienarbeit beschäftigt sich mit der Effizienzsteigerung der Abläufe in der Zukunftsfabrik Bodensee, einer Lern- und Forschungsumgebung für automatisierte Fertigungsprozesse. Trotz einer bereits optimierten SPS-Architektur bestanden weiterhin Schwachstellen hinsichtlich der Prozessstabilität, Bedienbarkeit und Wartbarkeit sowie bei der Einbindung zusätzlicher Peripheriegeräte. Ziel der Arbeit war es daher, die bestehenden Abläufe systematisch zu analysieren und durch geeignete technische Maßnahmen robuster, effizienter und besser nachvollziehbar zu gestalten. Hierfür wurden die SPS-Ablaufsteuerung stabilisiert, eine Lichtschranke am Handarbeitsplatz in die Steuerungslogik integriert, zeitabhängige Ablaufparameter optimiert sowie das Human Machine Interface (HMI) überarbeitet. Darüber hinaus wurden Anpassungen an der Auftragsverwaltung vorgenommen, die Integration weiterer Peripheriegeräte untersucht und eine Python-basierte Zeitmessung zur automatisierten Erfassung von Prozesszeiten entwickelt. Ergänzend erfolgte eine umfassende Dokumentation der Programmstruktur im Wissensmanagement der Zukunftsfabrik. Die umgesetzten Maßnahmen führten zu einer höheren Prozesssicherheit, einer verbesserten Bedienbarkeit und einer besseren Wartbarkeit der Anlage. Gleichzeitig wurde eine Grundlage für zukünftige datenbasierte Analysen sowie die weitere Integration von Industrie-4.0-Anwendungen geschaffen. Insgesamt konnte die Zukunftsfabrik Bodensee gezielt weiterentwickelt und für zukünftige Erweiterungen vorbereitet werden.

Abstract

This thesis focuses on improving the efficiency of the production processes in the *Zukunftsfabrik Bodensee*, a learning and research environment for automated manufacturing systems. Although the PLC architecture had already been optimized in previous work, several challenges regarding process stability, usability, maintainability, and integration of additional peripheral devices remained. The objective of this thesis was to systematically analyze the existing processes and enhance their efficiency, robustness, and transparency through appropriate technical improvements. To achieve this, the PLC sequence control was stabilized, a photoelectric sensor was integrated into the manual workstation, time-dependent process parameters were optimized, and the human-machine interface (HMI) was redesigned. Furthermore, the job management system was improved, additional peripheral devices were evaluated for future integration, and a Python-based application for automated process time acquisition was developed. The implemented measures increased process reliability, improved system usability and maintainability, and established a foundation for future data-driven analyses and Industry 4.0 applications. Overall, the work successfully enhanced the *Zukunftsfabrik Bodensee* and prepared it for further technical development.

Inhaltsverzeichnis

ERKLÄRUNG ZUR EIGENLEISTUNG ..... **II**

KURZFASSUNG ..... **III**

ABSTRACT ..... **IV**

INHALTSVERZEICHNIS ..... **V**

ABBILDUNGSVERZEICHNIS ..... **VII**

ABKÜRZUNGSVERZEICHNIS ..... **VIII**

1. EINLEITUNG **1**

1.1 MOTIVATION 1

1.2 ZIELSETZUNG 2

2. THEORETISCHE GRUNDLAGEN **4**

2.1 INDUSTRIE 4.0 – BEGRIFF UND BEDEUTUNG 4

2.2 FUNKTIONSWEISE UND AUFBAU SPEICHERPROGRAMMIERBARER STEUERUNGEN 7

2.2.1 HARDWAREAUFBAU EINER SPS 7

2.2.2 STEUERUNG UND VERARBEITUNG VON DATEN 9

2.2.3 PROGRAMMABLAUF UND ZYKLUSZEIT 9

2.2.4 DAS EVA-PRINZIP AM BEISPIEL SPS 10

2.3 KOMMUNIKATIONSPROTOKOLLE UND SCHNITTSTELLEN IN DER INDUSTRIELLEN AUTOMATISIERUNG 11

3. STAND DER TECHNIK **14**

4. VORGEHENSWEISE **17**

5. UMSETZUNG UND ERGEBNISSE **21**

5.1 STABILISIERUNG DER SPS-ABLAUFSTEUERUNG 21

5.2 ANBINDUNG DER LICHTSCHRANKE AM HANDARBEITSPLATZ 23

5.3 ANPASSUNG ZEITABHÄNGIGER ABLAUFPARAMETER 24

5.4 ÜBERARBEITUNG DES HMI 25

5.5 ANPASSUNGEN AN DER AUFTRAGSVERWALTUNG 27

5.6 BETRACHTUNG UND INTEGRATION WEITERER PERIPHERIEGERÄTE 29

5.7 IFM-MATE AM HANDARBEITSPLATZ 29

5.8 KAMERAANBINDUNG AN DIE SPS 30

5.9 TRANSPORTROBOTER HUSKY 31
5.10 DOKUMENTATION DER PROGRAMMSTRUKTUR IM WISSENSMANAGEMENT 34
5.11 ZEITMESSUNG ÜBER PYTHON 35
6. FAZIT 39
AUSBLICK 41
LITERATURVERZEICHNIS IX
NUTZUNG VON KÜNSTLICHER INTELLIGENZ X

Abbildungsverzeichnis

- Abbildung 1; Industrieller Fortschritt über die Zeit [1] 5
- Abbildung 2; Aufbau einer SPS [2] 8
- Abbildung 3; Lichtschranke am Handarbeitsplatz..... 18
- Abbildung 4; Überarbeitetes HMI 27
- Abbildung 5; Transportroboter HUSKY [4]..... 31

Abkürzungsverzeichnis

Abkürzung	*Bedeutung*
CPU	Central Processing Unit
DB	Datenbaustein
DBD	Datenbaustein-Doppelwort
DINT	Double Integer
ERP	Enterprise Resource Planning
EVA	Eingabe – Verarbeitung – Ausgabe
HMI	Human Machine Interface
I/O	Input/Output (Ein- und Ausgänge)
IT	Informationstechnik
MES	Manufacturing Execution System
OPC UA	Open Platform Communications Unified Architecture
PC	Personal Computer
PROFINET	Process Field Network
PROFIBUS	Process Field Bus
QR	Quick Response
SCADA	Supervisory Control and Data Acquisition
SPS	Speicherprogrammierbare Steuerung
WLAN	Wireless Local Area Network

1. Einleitung

Die Zukunftsfabrik Bodensee am Campus Friedrichshafen der DHBW Ravensburg stellt eine praxisnahe Lern- und Forschungsumgebung für automatisierte Fertigungsprozesse dar. Innerhalb der Anlage werden Werkstücke in mehreren aufeinanderfolgenden Prozessschritten bearbeitet. Dabei laufen einzelne Fertigungsschritte parallel ab, wodurch ein hoher Durchsatz erreicht werden soll. Die Anlage umfasst unter anderem Stationen zur Werkstückzuführung, Quick Response (QR) - Code-Erfassung, roboterbasierten Montage, manuellen Bearbeitung, Qualitätsprüfung und Sortierung. Bereits in der bisherigen Arbeit wurde deutlich, dass die Effizienz der Anlage wesentlich von einer übersichtlichen Steuerungsstruktur, zuverlässigen Schnittstellen und einer nachvollziehbaren Prozesslogik abhängt. Die vorliegende Arbeit mit dem Titel „Effizienzsteigerung der Abläufe in der Zukunftsfabrik Bodensee“ knüpft an diese Vorarbeiten an. Während die bisherige Ausarbeitung vor allem die Verbesserung der SPS-Architektur durch eine klarere Programmstruktur, eine übergeordnete Schrittkette und stationsbezogene Zustandsautomaten betrachtete, erweitert diese Arbeit den Fokus auf die Abläufe der Anlage insgesamt. Im Mittelpunkt steht dabei die Frage, wie bestehende Steuerungs-, Bedien- und Kommunikationsstrukturen weiterentwickelt werden können, um den Betrieb der Anlage stabiler, verständlicher, wartungsfreundlicher und effizienter zu gestalten.

1.1 Motivation

Die Motivation dieser Arbeit ergibt sich aus der praktischen Ausgangssituation der Zukunftsfabrik Bodensee. Die Anlage besitzt ein hohes Maß an Komplexität, da sie viele Komponenten wie SPS, HMI, Roboter, Sensorik, Bildverarbeitung, externe Rechner und übergeordnete IT-Systeme miteinander verbindet. Gleichzeitig führt diese Systemvielfalt zu einer hohen Komplexität.

In der vorherigen Ausarbeitung wurde festgestellt, dass unzureichend dokumentierte Programmabschnitte, uneindeutige Variablennamen und schwer nachvollziehbare Ablauflogiken die Fehlersuche und Weiterentwicklung erheblich erschweren.
Zur Verbesserung wurden unter anderem eine konsistente Benennung, eine klarere Bausteinstruktur und eine ausführlichere Kommentierung umgesetzt. Diese Maßnahmen bilden eine wichtige Grundlage, reichen jedoch allein nicht aus, um alle Abläufe der Anlage dauerhaft effizient und nachvollziehbar zu gestalten. Besonders in einer Lern- und Forschungsfabrik ist eine verständliche Struktur von großer Bedeutung. Die Anlage wird regelmäßig durch Studierende erweitert, analysiert und angepasst. Aufgrund der begrenzten Bearbeitungszeit einzelner Projektgruppen muss eine schnelle Einarbeitung möglich sein. Unübersichtliche Steuerungsstrukturen führen dagegen zu längeren Stillstands Zeiten, erschweren die Fehlersuche und erhöhen das Risiko, dass bestehende Probleme in nachfolgenden Arbeiten erneut auftreten. Darüber hinaus gewinnt die Zukunftsfabrik als digital vernetzte Produktionsumgebung zunehmend an Bedeutung. Neben der reinen Funktion der Anlage rücken daher auch Aspekte wie Bedienbarkeit, Datenverfügbarkeit, Schnittstellenqualität und Erweiterbarkeit stärker in den Vordergrund. Verbesserungen in diesen Bereichen schaffen einen direkten Mehrwert für den Betrieb der Anlage, die Lehre sowie zukünftige Studien- und Projektarbeiten.

1.2 Zielsetzung

Ziel dieser Arbeit ist es, die Abläufe in der Zukunftsfabrik Bodensee systematisch zu analysieren und durch geeignete technische Maßnahmen effizienter, robuster und besser nachvollziehbar zu gestalten. Dabei werden bestehende Ergebnisse aus der vorherigen Steuerungsarbeit aufgegriffen und gezielt weitergeführt. Im Vordergrund steht zunächst die Stabilisierung und Optimierung der SPS-basierten Ablaufsteuerung.

Die bereits eingeführte Kombination aus übergeordneter Schrittkette und stationsbezogenen Zustandsautomaten dient dabei als Grundlage, um einzelne Prozessabschnitte gezielt zu erweitern, Fehlerquellen zu reduzieren und den Anlagenbetrieb transparenter zu gestalten. Darauf aufbauend werden weitere Effizienzpotenziale betrachtet.

Dazu gehören insbesondere die Verbesserung der Bedienbarkeit am HMI, die Reduzierung unnötiger Bedienfehler, die strukturierte Einbindung zusätzlicher Peripheriegeräte sowie eine verbesserte Koordination zwischen Förderband, Stationen, Robotern und Sensorik. Ebenso sollen bestehende Ansätze zur Einbindung von Bildverarbeitung und weiteren Anlagenkomponenten weiter untersucht werden. Zusammenfassend verfolgt die Arbeit das Ziel, eine technische Grundlage zu schaffen, auf der die Zukunftsfabrik Bodensee zuverlässiger betrieben und einfacher weiterentwickelt werden kann. Die angestrebte Effizienzsteigerung bezieht sich daher nicht ausschließlich auf kürzere Prozesszeiten, sondern ebenso auf eine höhere Prozesssicherheit, bessere Wartbarkeit, verständlichere Programmstrukturen sowie eine verbesserte Bedien- und Datenqualität.

2. Theoretische Grundlagen

Um die in dieser Arbeit beschriebenen Maßnahmen nachvollziehen zu können, werden zunächst die relevanten theoretischen Grundlagen erläutert. Die Zukunftsfabrik Bodensee verbindet klassische Automatisierungstechnik mit Elementen der vernetzten Produktion. Daher ist ein grundlegendes Verständnis von Industrie 4.0, speicherprogrammierbaren Steuerungen sowie industriellen Kommunikationsschnittstellen erforderlich.

Die folgenden Unterkapitel ordnen diese Themen fachlich ein und schaffen die Grundlage für die spätere Beschreibung der Vorgehensweise, Umsetzung und Bewertung der durchgeführten Optimierungsmaßnahmen.

2.1 Industrie 4.0 – Begriff und Bedeutung

Die Umsetzung von Industrie-4.0-Konzepten setzt eine durchgängige Kommunikation zwischen der Produktionsebene und den übergeordneten Informationssystemen voraus. Eine zentrale Rolle übernimmt hierbei die SPS. Sie bildet das Bindeglied zwischen der physischen Produktion und den digitalen Informationssystemen eines Unternehmens. Während Sensoren kontinuierlich Prozessdaten wie Positionen, Temperaturen oder Schaltzustände erfassen und Aktoren mechanische Bewegungen ausführen, verarbeitet die SPS diese Informationen in Echtzeit und steuert die jeweiligen Fertigungsprozesse. Aufgrund ihrer deterministischen Arbeitsweise gewährleistet sie einen zuverlässigen und sicheren Betrieb automatisierter Produktionsanlagen.
Mit der zunehmenden Digitalisierung hat sich die Funktion der SPS deutlich erweitert. Neben der eigentlichen Ablaufsteuerung übernimmt sie heute zunehmend Aufgaben der Datenerfassung und Kommunikation.

Moderne Steuerungen verfügen über standardisierte Kommunikationsschnittstellen wie PROFINET oder OPC UA, wodurch Prozessdaten nicht nur innerhalb der

Anlage ausgetauscht, sondern auch an übergeordnete Systeme weitergegeben werden können. Dies ermöglicht eine durchgängige Digitalisierung der Produktion und schafft die Grundlage für datenbasierte Entscheidungen.

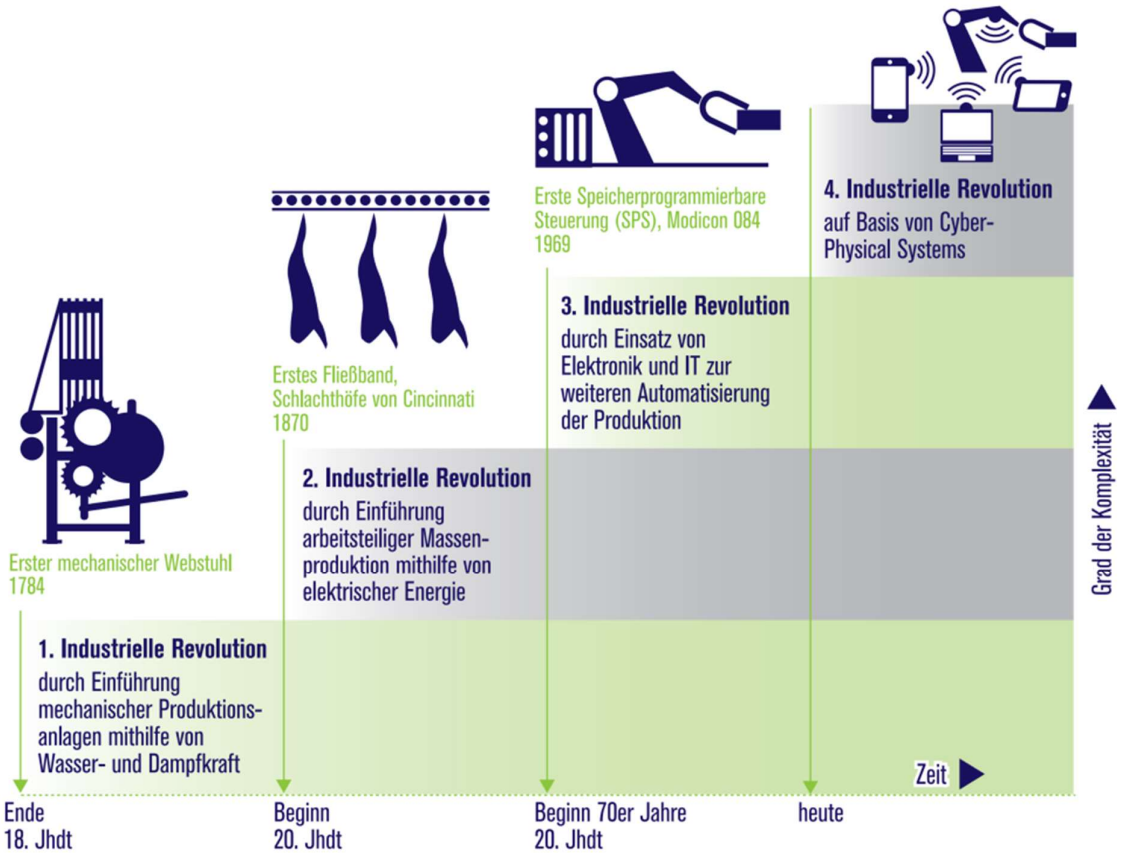


Abbildung 1; Industrieller Fortschritt über die Zeit [1]

Im Rahmen der vertikalen Integration werden die von der SPS bereitgestellten Informationen an Systeme wie Supervisory Control and Data Acquisition (SCADA) oder Manufacturing Execution Systems (MES) übertragen. Dort werden die Daten gespeichert, visualisiert und analysiert, um Kennzahlen zur Produktivität, Qualität oder Anlagenverfügbarkeit bereitzustellen. Gleichzeitig erfolgt eine Rückkopplung von Produktionsaufträgen oder Parametern aus den übergeordneten Systemen an die Steuerung.

Dadurch entsteht ein kontinuierlicher Informationsfluss zwischen der Feldebene und der Unternehmensleitebene, der eine flexible und effiziente Produktionssteuerung ermöglicht.

Neben der vertikalen Vernetzung gewinnt auch die horizontale Integration zunehmend an Bedeutung. Dabei werden Unternehmen, Lieferanten und Kunden digital miteinander verbunden, sodass Informationen entlang der gesamten Wertschöpfungskette ausgetauscht werden können. Dies ermöglicht eine höhere Transparenz sowie eine schnellere Reaktion auf Veränderungen innerhalb der Produktion oder der Lieferkette. Gleichzeitig entstehen neue Möglichkeiten zur Umsetzung individueller Kundenanforderungen bis hin zur wirtschaftlichen Fertigung von Losgröße eins.

Die in dieser Studienarbeit entwickelte Lösung ordnet sich in dieses Industrie-4.0-Konzept ein. Dabei werden Prozessdaten einer Siemens-SPS mithilfe eines Python-Skripts ausgelesen und für ein Manufacturing Execution System bereitgestellt. Durch diese Kopplung wird eine durchgängige Datenübertragung zwischen der Steuerungsebene und der Produktionsebene realisiert.
Die gewonnenen Daten können anschließend zur Analyse, Visualisierung und Optimierung von Fertigungsprozessen genutzt werden. Die entwickelte Schnittstelle trägt somit dazu bei, die Transparenz innerhalb der Produktion zu erhöhen und bildet eine Grundlage für weiterführende Industrie-4.0-Anwendungen. [1], [2]

2.2 Funktionsweise und Aufbau speicherprogrammierbarer Steuerungen

SPS bilden das zentrale Steuerungssystem moderner automatisierter Fertigungsanlagen. Sie übernehmen die Aufgabe, Signale von Sensoren aufzunehmen, diese entsprechend einem programmierten Steuerungsalgorithmus zu verarbeiten und daraus Steuersignale für Aktoren abzuleiten. Im Gegensatz zu klassischen Relaissteuerungen basiert die Funktion einer SPS auf einem frei programmierbaren Speicher, wodurch Änderungen an Steuerungsabläufen ohne aufwendige Verdrahtungsarbeiten vorgenommen werden können. Aufgrund ihrer hohen Zuverlässigkeit, Echtzeitfähigkeit und modularen Erweiterbarkeit werden SPS heute in nahezu allen Bereichen der industriellen Automatisierung eingesetzt. Moderne Steuerungen kommunizieren darüber hinaus über industrielle Bussysteme wie PROFINET oder EtherCAT mit dezentralen Ein- und Ausgabegeräten sowie übergeordneten Leitsystemen und bilden damit eine wesentliche Grundlage von Industrie-4.0-Anwendungen.

2.2.1 Hardwareaufbau einer SPS

Der Aufbau einer speicherprogrammierbaren Steuerung erfolgt modular und setzt sich aus mehreren Hardwarekomponenten zusammen. Das Kernstück bildet die Central Processing Unit (CPU), welche das Anwenderprogramm ausführt und sämtliche Rechen- sowie Kommunikationsaufgaben übernimmt.

Darüber hinaus besitzt die CPU einen Arbeits- und Programmspeicher, in dem sowohl das Steuerungsprogramm als auch Prozessdaten abgelegt werden.

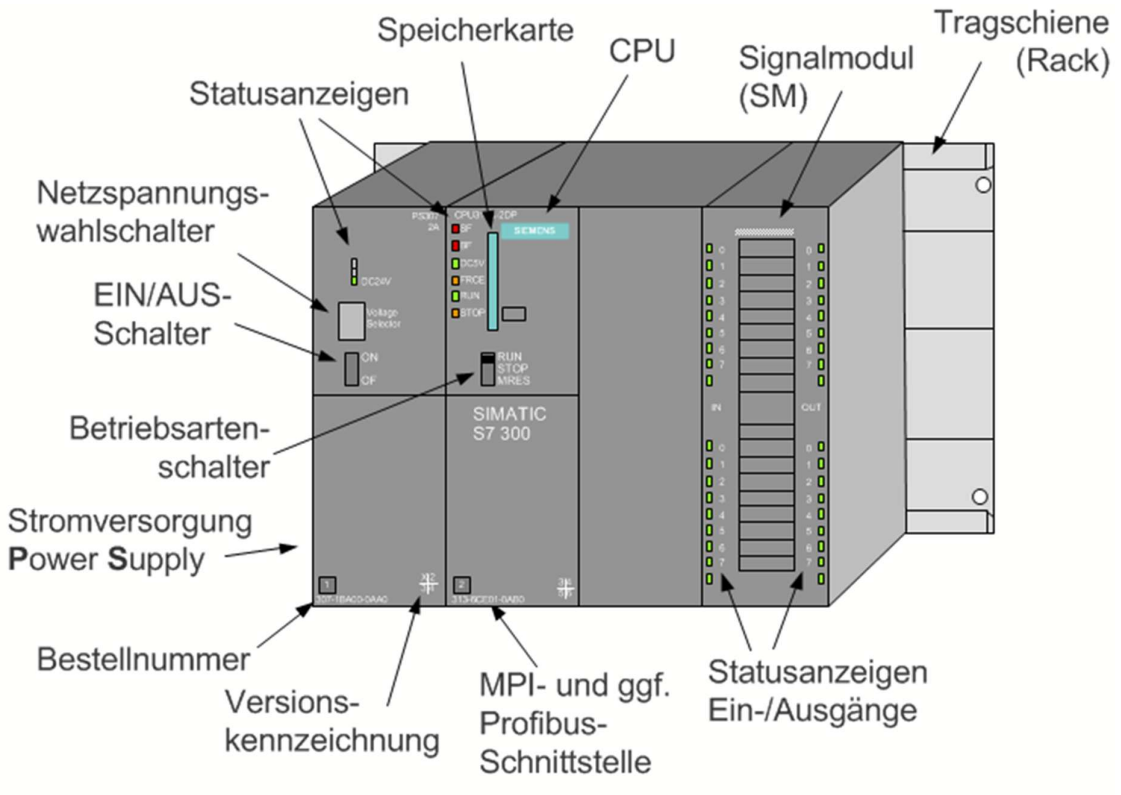


Abbildung 2; Aufbau einer SPS [3]

Zur Anbindung der Peripherie werden Ein- und Ausgangsmodule (I/O-Module) verwendet. Eingabemodule erfassen digitale oder analoge Signale von Sensoren, beispielsweise Lichtschranken, Näherungsschaltern oder Temperatursensoren. Ausgangsmodule übertragen die von der CPU berechneten Schaltsignale an Aktoren wie Motoren, Ventile oder Relais. Ergänzend kommen Kommunikationsmodule zum Einsatz, welche den Datenaustausch über industrielle Netzwerke ermöglichen.

Die Spannungsversorgung stellt sämtliche Baugruppen mit der erforderlichen Betriebsspannung bereit. Je nach Anwendungsfall können zusätzliche Spezialmodule, beispielsweise Sicherheitssteuerungen, Technologiemodule oder dezentrale Peripherie, integriert werden. Durch diesen modularen Aufbau lässt sich eine SPS flexibel an unterschiedliche Automatisierungsaufgaben anpassen und bei Bedarf erweitern. [4]

2.2.2 Steuerung und Verarbeitung von Daten

Die Datenverarbeitung innerhalb einer SPS erfolgt durch das kontinuierliche Einlesen, Verarbeiten und Ausgeben von Prozessdaten. Zunächst werden die Eingangssignale aller angeschlossenen Sensoren in ein sogenanntes Prozessabbild der Eingänge übernommen. Dieses stellt sicher, dass das Anwenderprogramm während eines Programmzyklus mit einem konsistenten Datensatz arbeitet.

Auf Grundlage dieser Eingangsdaten führt die CPU das Anwenderprogramm aus. Hierbei werden logische Verknüpfungen, mathematische Berechnungen, Zeitfunktionen oder Zustandsautomaten verarbeitet. Während der Programmausführung werden interne Variablen, Merker sowie Datenbausteine genutzt, um Zustände oder Parameter dauerhaft zu speichern.

Nach Abschluss der Programmbearbeitung werden die berechneten Ausgangswerte in das Prozessabbild der Ausgänge geschrieben und anschließend an die angeschlossenen Aktoren übertragen. Diese zyklische Verarbeitung gewährleistet ein deterministisches Verhalten der Steuerung, wodurch Prozessabläufe jederzeit reproduzierbar ausgeführt werden können. Insbesondere in industriellen Fertigungsanlagen ist diese Vorhersagbarkeit eine wesentliche Voraussetzung für einen sicheren und stabilen Betrieb. [4], [5]

2.2.3 Programmablauf und Zykluszeit

Der Betrieb einer SPS erfolgt überwiegend zyklisch. Jeder Programmzyklus beginnt mit dem Einlesen sämtlicher Eingangssignale in das Prozessabbild der Eingänge. Anschließend wird das Anwenderprogramm vollständig abgearbeitet. Dabei werden sämtliche Programmbausteine in einer festgelegten Reihenfolge ausgeführt.

Nach Abschluss der Berechnungen werden die ermittelten Ausgangswerte in das Prozessabbild der Ausgänge übertragen und an die angeschlossenen Aktoren ausgegeben.

Die Zeit, welche für diesen vollständigen Ablauf benötigt wird, wird als Zykluszeit bezeichnet. Sie hängt unter anderem von der Leistungsfähigkeit der CPU, der Größe des Steuerungsprogramms sowie von Kommunikations- und Diagnoseaufgaben ab. In industriellen Anwendungen bewegen sich Zykluszeiten häufig im Bereich weniger Millisekunden.

Eine möglichst kurze und konstante Zykluszeit verbessert das Reaktionsverhalten einer Steuerung und erhöht die Prozessqualität. Überschreitet die Zykluszeit einen definierten Grenzwert, können Verzögerungen oder Fehlfunktionen innerhalb der Anlage auftreten. Aus diesem Grund besitzen moderne SPS umfangreiche Diagnosefunktionen zur Überwachung der Zykluszeit und der Programmausführung. [4]

2.2.4 Das EVA-Prinzip am Beispiel SPS

Die Arbeitsweise einer speicherprogrammierbaren Steuerung lässt sich anschaulich durch das EVA-Prinzip (Eingabe – Verarbeitung – Ausgabe) beschreiben. Dieses stellt das grundlegende Funktionsprinzip nahezu aller digitalen Steuerungssysteme dar.

Im ersten Schritt werden Eingaben durch Sensoren erfasst. Dazu zählen beispielsweise Signale von Lichtschranken, Endschaltern oder Drucksensoren. Diese Informationen werden von den Eingabemodulen an die CPU übermittelt.

Anschließend erfolgt die Verarbeitung der Eingangsdaten im Anwenderprogramm. Hier werden logische Entscheidungen getroffen, Grenzwerte überprüft oder Berechnungen durchgeführt. Auf Grundlage dieser Verarbeitung entscheidet die SPS, welche Ausgänge aktiviert oder deaktiviert werden.

Im letzten Schritt werden die berechneten Ausgangssignale an Aktoren übertragen. Dies können beispielsweise Elektromotoren, Magnetventile, Förderbänder oder Signalleuchten sein. Anschließend beginnt der nächste Programmzyklus erneut mit dem Einlesen der Eingänge. Durch diese kontinuierliche Wiederholung entsteht eine nahezu verzögerungsfreie Steuerung des technischen Prozesses.

Das EVA-Prinzip bildet somit die Grundlage für den deterministischen Betrieb einer SPS und stellt sicher, dass auf Veränderungen innerhalb des Prozesses schnell und reproduzierbar reagiert werden kann. [4]

2.3 Kommunikationsprotokolle und Schnittstellen in der industriellen Automatisierung

Die zunehmende Digitalisierung industrieller Fertigungsanlagen führt zu einem steigenden Bedarf an einer zuverlässigen Kommunikation zwischen den einzelnen Automatisierungskomponenten. Sensoren, Aktoren, SPS, Human Machine Interfaces (HMI), Industrie-PCs sowie übergeordnete MES oder Enterprise Resource Planning (ERP)-Systeme müssen Informationen in Echtzeit austauschen, um einen stabilen und effizienten Produktionsprozess sicherzustellen. Die hierfür eingesetzten Kommunikationsprotokolle definieren die Regeln für den Datenaustausch und gewährleisten eine herstellerübergreifende Interoperabilität unterschiedlicher Geräte.

In der industriellen Automatisierung werden je nach Anwendungsfall verschiedene Feldbus- und Industrial-Ethernet-Protokolle eingesetzt. Während klassische Feldbusse wie PROFIBUS oder CANopen vor allem in bestehenden Anlagen anzutreffen sind, dominieren, heute Ethernet-basierte Systeme aufgrund ihrer höheren Übertragungsgeschwindigkeit, größeren Bandbreite und besseren Integrationsmöglichkeiten in moderne IT-Infrastrukturen.

Industrial Ethernet bildet damit eine wesentliche Grundlage für Industrie-4.0-Anwendungen und cyber-physische Produktionssysteme.

Im Umfeld von Siemens-Steuerungen stellt PROFINET den heute am weitesten verbreiteten Kommunikationsstandard dar. Das auf Industrial Ethernet basierende Protokoll ermöglicht sowohl den zyklischen Echtzeitdatenaustausch zwischen SPS und dezentraler Peripherie als auch die azyklische Übertragung von Diagnose- und Parametrierungsdaten. Durch kurze Übertragungszeiten im Millisekunden Bereich eignet sich PROFINET insbesondere für zeitkritische Steuerungsaufgaben in automatisierten Fertigungsanlagen.

Neben der Kommunikation innerhalb der Automatisierungsebene gewinnt die Anbindung von Industrie-PCs zunehmend an Bedeutung. Industrie-PCs übernehmen häufig Aufgaben wie Datenerfassung, Prozessanalyse, Visualisierung oder die Kommunikation mit übergeordneten IT-Systemen. Hierfür kommen standardisierte Schnittstellen und Kommunikationsprotokolle zum Einsatz. Bei Siemens-Steuerungen erfolgt der Datenaustausch häufig über das proprietäre S7-Kommunikationsprotokoll, auf welches Programmbibliotheken wie Snap7 oder python-snap7 zugreifen können. Dadurch können Variablen aus Datenbausteinen einer SPS gelesen und geschrieben werden, ohne Änderungen an der eigentlichen Steuerungslogik vorzunehmen.

Für Anwendungen der Industrie 4.0 gewinnt darüber hinaus das plattformunabhängige Kommunikationsprotokoll OPC UA (Open Platform Communications Unified Architecture) zunehmend an Bedeutung. OPC UA ermöglicht einen sicheren, standardisierten und herstellerunabhängigen Datenaustausch zwischen Maschinen, Steuerungen und IT-Systemen. Im Gegensatz zu klassischen Feldbusprotokollen stellt OPC UA nicht nur Prozessdaten bereit, sondern beschreibt zusätzlich deren Bedeutung durch ein standardisiertes Informationsmodell. Dadurch wird eine semantische Interoperabilität geschaffen, welche insbesondere für digitale Zwillinge, Cloud-Anwendungen und intelligente Produktionssysteme von großer Bedeutung ist.

Die Auswahl eines geeigneten Kommunikationsprotokolls richtet sich nach den Anforderungen der jeweiligen Anwendung. Während PROFINET aufgrund seiner Echtzeitfähigkeit bevorzugt für die direkte Maschinensteuerung eingesetzt wird, eignet sich OPC UA insbesondere für die vertikale Integration zwischen Produktionsebene und Unternehmens-IT. In der vorliegenden Studienarbeit bildet die Kommunikation zwischen einem Industrie-PC und einer Siemens-SPS über das S7-Kommunikationsprotokoll die technische Grundlage für die implementierte Zeitmessung sowie den später beschriebenen Datenaustausch. [1], [2]

3. Stand der Technik

Zu Beginn der zweiten Studienarbeit im zweiten Quartal 2026 lag in der Zukunftsfabrik Bodensee bereits eine weiterentwickelte SPS-Architektur vor. Diese basierte auf den Ergebnissen der vorherigen Studienarbeit, in der die zuvor schwer nachvollziehbare Programmstruktur analysiert und grundlegend überarbeitet wurde. Die Anlage selbst war weiterhin als mehrstufiger Fertigungsprozess für Raspberry-Pi-Gehäuse aufgebaut. Die Werkstücke durchlaufen dabei nacheinander mehrere Stationen, während verschiedene Fertigungsschritte parallel ausgeführt werden, um den Durchsatz der Anlage zu erhöhen. Für jede Bearbeitungsstation wird grundsätzlich zwischen der Anwesenheit des Werkstückträgers und des Werkstücks unterschieden.

Die vorherige Studienarbeit hatte gezeigt, dass das bestehende SPS-Programm ursprünglich nur unzureichend dokumentiert war. Zahlreiche Variablen waren uneindeutig benannt, wodurch das Verständnis des Codes erheblich erschwert wurde. Die Fehlersuche erfolgte deshalb häufig durch wiederholtes Testen an der realen Anlage und durch den Abgleich des beobachteten Verhaltens mit einzelnen Programmstellen. Daraus ergab sich die Notwendigkeit, die Steuerungsstruktur besser zu dokumentieren, Variablen sprechender zu benennen und die Programmbausteine klarer zu gliedern.

Als Ergebnis dieser Vorarbeit bestand zu Beginn des Q2-Projekts bereits eine grundsätzliche Trennung zwischen übergeordneter Ablaufsteuerung und stationsspezifischer Logik. Der Gesamtprozess wurde über eine Schrittkette koordiniert, während die einzelnen Stationen mithilfe von Zustandsautomaten abgebildet wurden. Diese Kombination stellte eine robuste und nachvollziehbare Grundlage für weitere Erweiterungen dar. Gleichzeitig wurde in der vorherigen Arbeit deutlich, dass reale Prozessbedingungen nicht vollständig durch theoretische Zustandsdiagramme abgebildet werden können.

Fehlende Rückmeldesignale, zeitverzögertes mechanisches Verhalten sowie Abhängigkeiten zwischen Robotern, Sensoren und Fördertechnik machten zusätzliche Anpassungen in der praktischen Umsetzung erforderlich.

Der Anlagenzustand war somit nicht als vollständig abgeschlossenes System zu bewerten, sondern als weiterentwickelte, jedoch weiterhin optimierungsbedürftige Steuerungsbasis. Einzelne Stationen und Peripheriegeräte waren bereits funktional vorhanden, aber noch nicht vollständig oder nur eingeschränkt in die Steuerungslogik integriert. Am Handarbeitsplatz wurde beispielsweise ein ifm-mate-System eingesetzt, das den Bediener durch visuelle Anweisungen unterstützt und die Entnahme korrekter Bauteile überprüft. Gleichzeitig wurde die Station noch über eine feste Wartezeit abgeschlossen. Eine Lichtschranke zur Werkstückerkennung war zwar vorhanden, jedoch noch nicht in die Steuerungslogik eingebunden.

Auch im Bereich der Bildverarbeitung bestand weiterer Entwicklungsbedarf. In Station 3 wurde für das Bin Picking bereits ein Kamerasystem verwendet, wobei Bildverarbeitung, Objekterkennung und Berechnung der Greifpose über den Mikado-Rechner erfolgten. Die berechneten Positionsdaten wurden anschließend an den Roboter übergeben. Für die Qualitätsprüfung wurde zusätzlich eine Wenglor-Kamera eingesetzt, welche anhand eines Referenzbildes Kriterien für die Demontage bewertet. Die SPS erhielt dabei qualitätsrelevante Variablen und wertete diese zur Unterscheidung zwischen Gut- und Schlechtteilen aus.

Trotz dieser vorhandenen Bildverarbeitung war die direkte Kameraanbindung an die SPS noch nicht vollständig umgesetzt. Die vorherige Studienarbeit sah deshalb bereits vor, im Rahmen der Fortführung eine Kamera als direktes Peripheriegerät der SPS zu konfigurieren. Ziel dieser Erweiterung war es, Bildverarbeitungsdaten, Statusinformationen und Trigger-Signale deterministischer in der Steuerungslogik verarbeiten zu können und die Abhängigkeit von rein zeitbasierten Annahmen zu reduzieren.

Ein weiterer Ausgangspunkt betraf den Transportroboter HUSKY. Dieser war bereits Bestandteil des Anlagenkonzepts und wurde von Station 1 zur Übergabe fertig montierter Werkstücke verwendet. Die Werkstücke werden dabei vom Roboter der Station 1 aufgenommen und auf dem HUSKY abgelegt. Für eine weitergehende Automatisierung bestand jedoch weiterhin Bedarf an einer zuverlässigeren Übergabe von Statusinformationen, insbesondere im Hinblick auf Zustände wie die Beladung des Transportroboters.

Die Auftragsverwaltung und die HMI-Bedienung stellten ebenfalls relevante Ausgangspunkte dar. Zwar existierten bereits grundlegende Strukturen zur Verwaltung von Jobs und Zählern, insbesondere im Zusammenhang mit Station 6, jedoch waren diese noch nicht vollständig in ein übergeordnetes Zustandsautomaten-System integriert. Zudem wurde in der vorherigen Arbeit bereits festgehalten, dass die Zoom-Funktionalität des HMI deaktiviert werden sollte, um unbeabsichtigte Fehleingaben zu vermeiden und eine konsistente Darstellung der Prozessvisualisierung zu gewährleisten.

Zusammenfassend wurde die Anlage im zweiten Quartal 2026 als grundsätzlich lauffähiges, aber noch nicht vollständig robustes und durchgängig integriertes System vorgefunden. Die vorherige Studienarbeit hatte eine wichtige strukturelle Grundlage geschaffen, insbesondere durch die Einführung von Schrittkette, Zustandsautomaten und verbesserter Programmdokumentation. Darauf aufbauend ergaben sich für die zweite Studienarbeit konkrete Handlungsfelder: systematische Fehlerbehebung, Anbindung weiterer Peripheriegeräte, Durchsatzsteigerung durch Timeranpassungen, Dokumentation der aktuellen Programmstruktur im Wissensmanagement, Python-basierte Zeitmessungen im Kontext der Arbeit von Herrn Bulander sowie die weitere Betrachtung von HUSKY, ifm-mate und Kameraanbindung an die SPS. [6]

4. Vorgehensweise

Dieses Kapitel beschreibt das methodische Vorgehen zur Bearbeitung der Studienarbeit. Aufbauend auf den Ergebnissen der vorherigen Studienarbeit wurden Optimierungspotenziale systematisch abgeleitet und in geeignete Maßnahmen zur Effizienzsteigerung überführt. Im Mittelpunkt stand dabei die Weiterentwicklung der Anlage unter Berücksichtigung der vorhandenen SPS-Struktur, der Bedienoberfläche, der Auftragsverwaltung sowie der angebundenen Peripherie. Die Bearbeitung erfolgte in mehreren aufeinander aufbauenden Schritten. Zunächst wurden die zu bearbeitenden Themenbereiche festgelegt und voneinander abgegrenzt. Dazu gehörten die Einbindung zusätzlicher Sensorik, die Betrachtung einer möglichen Kameraanbindung, die Anpassung von Timerwerten, die Vorbereitung einer vollständigen Zeitmessung über Python, die Dokumentation der Programmstruktur, vereinzelte Fehlerkorrekturen im SPS-Programm sowie die Überarbeitung des HMI und der Auftragsverwaltung.

Ein wesentlicher Schwerpunkt lag auf der systematischen Beseitigung bestehender Programm- und Ablaufprobleme. Dadurch sollte bis zum Technikforum am 27.05.2026 eine möglichst robuste Softwarebasis geschaffen werden, um die Anlage für die abschließende Durchsatzmessung unter stabilen Bedingungen betreiben zu können. Die Fehlerkorrekturen wurden daher nicht isoliert betrachtet, sondern als Bestandteil der Gesamtoptimierung eingeordnet.

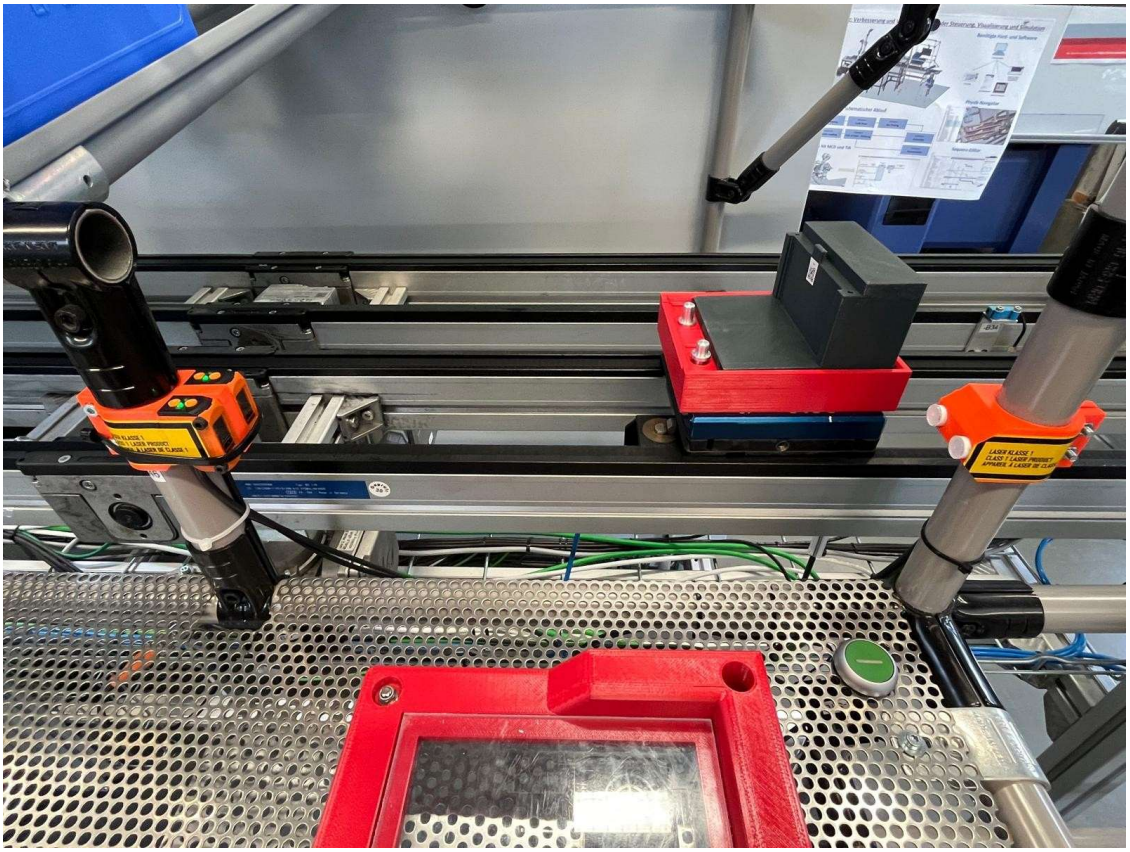


Abbildung 3; Lichtschranke am Handarbeitsplatz

Im nächsten Schritt wurden die geplanten Maßnahmen priorisiert. Technische Anpassungen, die unmittelbar zur Verbesserung der Prozesssicherheit oder zur Verkürzung von Laufzeiten beitragen konnten, wurden vorrangig behandelt. Dazu zählten insbesondere die Anbindung der Lichtschranke (Siehe Abbildung 1), die Anpassung einzelner Timerwerte sowie die Korrektur erkannter Programmfehler. Maßnahmen mit größerem Erweiterungscharakter, wie die theoretische Anbindung der 2D - Kamera oder die weiterführende Entwicklung der Auftragsverwaltung, wurden dagegen zunächst konzeptionell betrachtet und für zukünftige Erweiterungen vorbereitet.

Für die Anpassung der Stationsabläufe wurde festgelegt, dass Änderungen möglichst in die bestehende Struktur integriert werden sollten. Neue oder geänderte Programmteile sollten keine zusätzliche Komplexität erzeugen, sondern die Nachvollziehbarkeit der Steuerung erhalten oder verbessern. Dies betraf insbesondere die Einbindung von Sensorsignalen, die Anpassung zeitabhängiger Übergänge sowie die Erweiterung einzelner Ablaufbedingungen.

Ein weiterer Bestandteil des Vorgehens war die Überarbeitung der Bedienoberfläche. Dabei wurde festgelegt, welche HMI-Bereiche neu strukturiert und welche Bedienfunktionen angepasst oder entfernt werden sollten. Dazu zählten insbesondere die Neustrukturierung der HMI-Seiten, die Anpassung der Auftragsverwaltung sowie die Deaktivierung der Zoom-Möglichkeit. Die Zoom-Funktion wurde über ein JavaScript gezielt unterbunden, um eine unbeabsichtigte Veränderung der Bedienansicht während des Anlagenbetriebs zu vermeiden.

Die vollständige Zeitmessung über Python entstand dabei im Kontext einer Doktorarbeit zur Vereinfachung von Zeitstudien. Herr Florian Bulander trat hierfür mit der Anforderung an die Bearbeitenden heran, eine Lösung zur automatisierten Erfassung relevanter Prozesszeiten zu entwickeln. Aus diesem Grund wurde festgelegt, welche SPS-Variablen für eine aussagekräftige Laufzeitmessung relevant sind und wie diese über den Industrie-PC abgefragt werden können. Die Zeitmessung wurde damit als methodisches Werkzeug eingeordnet, um Prozesszeiten nachvollziehbar zu erfassen und für weiterführende wissenschaftliche Auswertungen nutzbar zu machen.

Ein weiterer Fokus der Bearbeitungsphase war die Dokumentation unserer Arbeit. Da die Zukunftsfabrik Bodensee regelmäßig durch weitere Studien- und Projektarbeiten erweitert wird, sollten die vorgenommenen Änderungen nicht nur funktional umgesetzt, sondern auch nachvollziehbar festgehalten werden.

Aus diesem Grund wurde die Programmstruktur im Wissensmanagement der Zukunftsfabrik Bodensee dokumentiert, um zukünftigen Bearbeitenden die Einarbeitung zu erleichtern und die Wartbarkeit der Anlage zu verbessern.

Abschließend war vorgesehen, die bearbeiteten Maßnahmen im Anlagenbetrieb zu überprüfen. Dabei sollte kontrolliert werden, ob die geänderten Abläufe stabil funktionieren, ob die Bedienung über das HMI verständlicher wird und ob die vorbereiteten Schnittstellen für spätere Erweiterungen geeignet sind. Die konkrete Umsetzung sowie die daraus resultierenden Ergebnisse werden im folgenden Kapitel dargestellt.

5. Umsetzung und Ergebnisse

In diesem Kapitel werden die im Rahmen der Studienarbeit umgesetzten Optimierungsmaßnahmen beschrieben und hinsichtlich ihrer Wirkung auf den Anlagenbetrieb bewertet. Der Fokus liegt auf der praktischen Weiterentwicklung der Zukunftsfabrik Bodensee. Die Maßnahmen bauen auf der bereits vorhandenen SPS-Struktur mit übergeordneter Schrittkette und stationsbezogenen Zustandsautomaten auf. Ziel war es, bestehende Ablaufprobleme zu reduzieren, einzelne Prozessschritte robuster auszuführen, die Bedienbarkeit zu verbessern und die Anlage für weitere Auswertungen und Erweiterungen vorzubereiten.

Die Umsetzung erfolgte schrittweise und wurde jeweils direkt an der Anlage getestet. Dadurch konnten nicht nur Programmänderungen validiert, sondern auch mechanische Verzögerungen, Sensorverhalten und Wechselwirkungen zwischen SPS, HMI, Robotern und Peripheriegeräten berücksichtigt werden. Im Ergebnis wurde die Steuerung nicht grundlegend neu aufgebaut, sondern gezielt erweitert und stabilisiert. Dies war notwendig, da die bestehende Struktur aus Schrittkette und Zustandsautomaten bereits eine geeignete Grundlage darstellte und eine weitere Umstrukturierung ohne konkreten Mehrwert zusätzliche Komplexität erzeugt hätte.

5.1 Stabilisierung der SPS-Ablaufsteuerung

Ein wesentlicher Bestandteil der Umsetzung war die Stabilisierung der vorhandenen SPS-Ablaufsteuerung. Aus der vorherigen Studienarbeit war bereits eine strukturierte Programmarchitektur hervorgegangen, bei der der Gesamtprozess durch eine übergeordnete Schrittkette koordiniert wird. Die einzelnen Stationen werden innerhalb eigener Zustandsautomaten abgebildet. Diese Struktur wurde im Rahmen dieser Arbeit beibehalten und gezielt erweitert.

Zu Beginn der Bearbeitung wurden zunächst die aktuell auftretenden Ablaufprobleme analysiert. Dabei zeigte sich, dass einzelne Fehler nicht durch eine grundsätzlich falsche Steuerungsstruktur verursacht wurden, sondern durch unvollständige Übergangsbedingungen, zeitabhängige Signalverläufe oder nicht vollständig berücksichtigte Rückmeldungen einzelner Komponenten. Besonders bei Förderband, Stoppern und stationsbezogenen Fertigmeldungen entstanden dadurch Situationen, in denen die Anlage nicht eindeutig erkennen konnte, ob ein Prozessschritt bereits abgeschlossen war oder ob noch auf eine mechanische Bewegung bzw. ein externes Signal gewartet werden musste.

Die Fehlerbehebung erfolgte daher nicht durch eine pauschale Ergänzung weiterer Timer, sondern durch eine gezielte Prüfung der jeweiligen Zustandsübergänge. Für jede betroffene Station wurde betrachtet, welche Signale den Wechsel zwischen zwei Zuständen auslösen und ob diese Signale im realen Anlagenbetrieb zuverlässig und zum richtigen Zeitpunkt anliegen. Wo notwendig, wurden zusätzliche Bedingungen ergänzt oder bestehende Übergangslogiken angepasst. Dadurch konnte das Verhalten der Stationen enger an den tatsächlichen Prozessablauf gekoppelt werden.

Ein Schwerpunkt lag auf der Vermeidung uneindeutiger Zwischenzustände. Insbesondere bei Stationen, die mit Stoppern, Werkstückträgern und externen Geräten interagieren, ist eine klare Trennung zwischen „bereit“, „in Bearbeitung“ und „fertig“ erforderlich. Durch die Anpassungen wurde erreicht, dass Fertigmeldungen erst dann gesetzt werden, wenn die für den jeweiligen Prozessschritt notwendigen Bedingungen erfüllt sind. Gleichzeitig wurde darauf geachtet, dass Rücksetzbedingungen eindeutig definiert sind, damit Stationen nach Abschluss eines Zyklus wieder in einen definierten Ausgangszustand übergehen können.

Als Ergebnis steht eine stabilere Ablaufsteuerung zur Verfügung, bei der einzelne Prozessabschnitte besser nachvollzogen und gezielter getestet werden können. Die vorgenommenen Änderungen verbessern nicht nur den unmittelbaren Anlagenbetrieb, sondern erleichtern auch zukünftige Erweiterungen, da neue Funktionen an klar definierten Zuständen und Übergangsbedingungen anknüpfen können.

5.2 Anbindung der Lichtschranke am Handarbeitsplatz

Am Handarbeitsplatz war bereits eine Lichtschranke zur Werkstückerkennung vorhanden, jedoch noch nicht vollständig in die Steuerungslogik eingebunden. Vor der Umsetzung wurde der Abschluss des Handarbeitsplatzes überwiegend zeitbasiert behandelt. Nach einer definierten Wartezeit wurde die Station als abgeschlossen betrachtet. Diese Lösung war zwar einfach umzusetzen, bildete den tatsächlichen Prozess jedoch nur eingeschränkt ab. Besonders bei manuellen Arbeitsschritten kann die Bearbeitungszeit variieren, wodurch eine reine Zeitannahme entweder zu lange Wartezeiten oder zu frühe Freigaben verursachen kann.

Die Lichtschranke wurde deshalb als zusätzliches Sensorsignal in die Stationslogik eingebunden. Ziel war es, den Zustand des Werkstücks am Handarbeitsplatz zuverlässiger zu erfassen und die Freigabe des nächsten Prozessschrittes stärker an den realen Anlagenzustand zu koppeln. Dazu wurde das Signal der Lichtschranke zunächst in der SPS geprüft und anschließend in die bestehende Ablaufstruktur der Station integriert. Die Auswertung erfolgte so, dass das Sensorsignal als zusätzliche Bedingung für den Abschluss bzw. die Freigabe des Handarbeitsplatzes verwendet werden kann.

Durch die Einbindung der Lichtschranke wird der Prozess am Handarbeitsplatz robuster. Die Steuerung kann nun besser erkennen, ob sich ein Werkstück im relevanten Bereich befindet oder ob der Platz frei ist. Dadurch wird die Abhängigkeit von festen Wartezeiten reduziert. Dies ist insbesondere deshalb wichtig, weil der Handarbeitsplatz als manuelle Station stärkeren Schwankungen unterliegt als automatisierte Stationen. Während Roboterbewegungen und Förderbandabläufe weitgehend reproduzierbar ablaufen, hängt die Bearbeitungszeit am Handarbeitsplatz vom Bediener, vom Montagezustand und vom jeweiligen Betriebsmodus ab.

Die Umsetzung verbessert somit sowohl die Prozesssicherheit als auch die Grundlage für spätere Durchsatzoptimierungen. Anstatt den Stationsabschluss ausschließlich über eine vorsichtig gewählte Wartezeit abzusichern, kann die SPS nun auf ein reales Prozesssignal reagieren. Dadurch wird die Steuerung näher an das tatsächliche Anlagenverhalten geführt.

5.3 Anpassung zeitabhängiger Ablaufparameter

Neben der Einbindung zusätzlicher Sensorik wurden bestehende Timerwerte überprüft und angepasst. In der Anlage werden Timer unter anderem verwendet, um mechanische Bewegungen abzusichern, kurze Signalverzögerungen zu überbrücken oder Übergänge zwischen Zuständen zu entprellen. Grundsätzlich sind Timer in SPS-Programmen ein sinnvolles Mittel, wenn reale Verzögerungen berücksichtigt werden müssen. Gleichzeitig können zu großzügig gewählte Timer die Zykluszeit des Gesamtprozesses unnötig verlängern.

Im Rahmen der Umsetzung wurden daher insbesondere solche Timer betrachtet, die direkten Einfluss auf den Durchsatz der Anlage haben. Dazu zählen Wartezeiten an Stationen, Verzögerungen beim Stopperverhalten sowie Zeitfenster nach Sensor- oder Kamerasignalen. Ziel war nicht, sämtliche Timerwerte minimal

einzustellen, sondern eine belastbare Balance zwischen Prozesssicherheit und Durchsatz zu erreichen.
Eine zu starke Reduzierung von Wartezeiten könnte dazu führen, dass mechanische Bewegungen noch nicht abgeschlossen sind oder Signale nicht zuverlässig erkannt werden. Eine zu große Wartezeit hingegen reduziert die Effizienz der Anlage, ohne die Prozesssicherheit weiter zu erhöhen.

Die Anpassungen wurden iterativ vorgenommen. Nach jeder Änderung wurde der betreffende Anlagenabschnitt getestet und das Verhalten beobachtet. Dabei wurde geprüft, ob die Station zuverlässig in den nächsten Zustand wechselt, ob Sensoren und Aktoren ausreichend Zeit für ihre Reaktion haben und ob unerwünschte Nebeneffekte auftreten. Nur Timerwerte, deren Reduzierung im realen Betrieb stabil funktionierte, wurden dauerhaft übernommen.

Das Ergebnis dieser Maßnahme ist eine verbesserte zeitliche Abstimmung einzelner Abläufe. Die Anlage reagiert in mehreren Prozessabschnitten schneller, ohne dass die grundlegende Stabilität der Steuerung beeinträchtigt wird. Gleichzeitig wurde deutlich, dass Timer nicht als Ersatz für fehlende Logik verwendet werden sollten. Wo ein Prozesszustand eindeutig über Sensoren oder Rückmeldesignale erkannt werden kann, ist diese Lösung einer rein zeitbasierten Annahme vorzuziehen.

5.4 Überarbeitung des HMI

Ein weiterer Schwerpunkt der Umsetzung lag in der Überarbeitung der Bedienoberfläche. Das HMI stellt die zentrale Schnittstelle zwischen Bedienperson und Anlage dar. Eine übersichtliche und fehlerarme Bedienung ist deshalb wesentlich für einen stabilen Anlagenbetrieb. Vor der Überarbeitung waren einzelne HMI-Bereiche unübersichtlich oder enthielten Bedienmöglichkeiten, die im laufenden Betrieb zu unbeabsichtigten Eingaben führen konnten.

Zunächst wurde die Struktur der HMI-Seiten überprüft. Ziel war es, die relevanten Anlageninformationen besser zu ordnen und häufig benötigte Bedienfunktionen leichter zugänglich zu machen.
Dabei wurde insbesondere darauf geachtet, dass Zustände der Anlage, Auftragsinformationen und grundlegende Steuerfunktionen klar voneinander getrennt sind. Eine solche Trennung reduziert die Wahrscheinlichkeit von Fehlbedienungen, da die Bedienperson schneller erkennt, in welchem Bereich eine bestimmte Funktion zu finden ist.

Zusätzlich wurde die Möglichkeit geschaffen bzw. verbessert, die Anlage über das HMI gezielt zurückzusetzen. Eine solche Reset-Funktion ist besonders bei Testläufen und Inbetriebnahme Situationen relevant. Da die Zukunftsfabrik regelmäßig im Rahmen von Studienarbeiten erweitert wird, treten während der Entwicklung häufiger Zustände auf, in denen die Anlage nicht vollständig produktiv weiterlaufen soll, sondern in einen definierten Grundzustand zurückgeführt werden muss. Die überarbeitete Reset-Funktion unterstützt diesen Vorgang und erhöht damit die Bedienbarkeit während der Entwicklung und Fehlersuche.

Ein weiterer Punkt war die Deaktivierung der Zoom-Funktion im HMI. Die Zoom-Möglichkeit konnte dazu führen, dass die Bedienansicht unbeabsichtigt verändert wurde. Dies ist im industriellen Bedienkontext ungünstig, da wichtige Schaltflächen oder Anzeigen dadurch verschoben, verkleinert oder aus dem sichtbaren Bereich bewegt werden können. Die Zoom-Funktion wurde deshalb über ein JavaScript gezielt unterbunden. Dadurch bleibt die Bedienoberfläche während des Anlagenbetriebs in einer konstanten Darstellung. Dies verbessert die Bedienqualität und reduziert das Risiko unbeabsichtigter Fehlbedienungen.

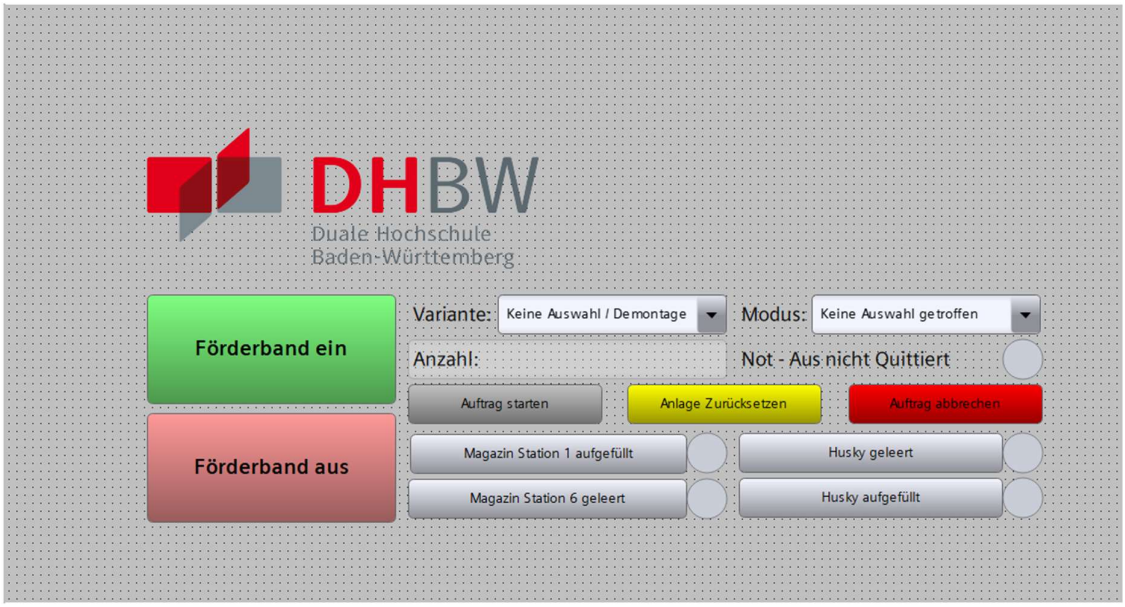


Abbildung 4; Überarbeitetes HMI

Insgesamt wurde das HMI, wie in Abbildung 2 zu sehen, durch die Anpassungen übersichtlicher und robuster gestaltet. Die Bedienperson erhält einen klareren Zugriff auf relevante Funktionen, während potenziell störende Bedienmöglichkeiten eingeschränkt wurden.

5.5 Anpassungen an der Auftragsverwaltung

Die Auftragsverwaltung bildet eine wichtige Grundlage für den automatisierten Betrieb der Zukunftsfabrik. Sie verwaltet auftragsbezogene Informationen wie Stückzahlen, Fertigungszustände und Zähler. Bereits zu Beginn der Arbeit existierten Strukturen zur Verwaltung von Jobs und Zählwerten. Diese waren jedoch noch nicht vollständig in das übergeordnete Zustandsautomaten-System integriert.

Im Rahmen dieser Arbeit wurden erste Anpassungen vorgenommen, um die Auftragsverwaltung stärker mit den vorhandenen Stationszuständen zu verknüpfen. Ziel war es, auftragsbezogene Zähler und Statusinformationen nicht isoliert in einzelnen Stationsabschnitten zu behandeln, sondern sie langfristig zentraler und nachvollziehbarer zu verwalten. Dadurch kann vermieden werden, dass gleiche oder ähnliche Zähllogiken mehrfach im Programm auftreten.

Die Umsetzung erfolgte zunächst vorbereitend. Bestehende Zählerstrukturen wurden analysiert und hinsichtlich ihrer Zuordnung zu Stationen und Aufträgen überprüft. Dabei zeigte sich, dass eine vollständige Zentralisierung der Auftragsverwaltung zwar sinnvoll ist, jedoch eine sorgfältige Anpassung der bestehenden Stationslogik erfordert. Besonders relevant ist dabei die Frage, welcher Zustand einer Station eindeutig als Grundlage für eine Stückzählung verwendet werden kann. Wird beispielsweise zu früh gezählt, kann ein Werkstück erfasst werden, obwohl der Prozessschritt noch nicht abgeschlossen ist. Wird zu spät gezählt, können nachfolgende Prozessschritte bereits gestartet sein, bevor die Auftragsverwaltung aktualisiert wurde.

Aus diesem Grund wurde die Auftragsverwaltung im Rahmen dieser Arbeit nicht vollständig neu aufgebaut, sondern für eine spätere tiefere Integration vorbereitet. Die vorhandenen Erkenntnisse bilden eine Grundlage dafür, Zählerfunktionen künftig gezielt aus den Stationsbausteinen auszulagern und in der Auftragsverwaltung zusammenzuführen. Dies würde die Stationslogik schlanker machen und die Wartbarkeit des Programms erhöhen.

5.6 Betrachtung und Integration weiterer Peripheriegeräte

Neben den direkten Anpassungen an SPS und HMI wurden weitere Peripheriegeräte betrachtet. Die Zukunftsfabrik Bodensee verbindet mehrere Automatisierungskomponenten, die jeweils eigene Schnittstellen, Signalverhalten und Betriebsbedingungen besitzen. Für eine effiziente Gesamtanlage reicht es daher nicht aus, nur die SPS-Logik isoliert zu optimieren. Vielmehr müssen externe Systeme so eingebunden werden, dass deren Zustände für die SPS nachvollziehbar und steuerungstechnisch nutzbar sind.

5.7 ifm-mate am Handarbeitsplatz

Das ifm-mate-System unterstützt die Bedienperson am Handarbeitsplatz durch visuelle Anweisungen und die Überprüfung der korrekten Bauteilentnahme. Im Rahmen dieser Arbeit wurde das System nicht grundsätzlich neu konfiguriert, jedoch im Zusammenhang mit der Stationslogik betrachtet. Dabei lag der Fokus darauf, den Handarbeitsplatz als Kombination aus manueller Tätigkeit, visueller Unterstützung und SPS-basierter Ablaufsteuerung zu verstehen.

Die Einbindung der Lichtschranke verbessert in diesem Zusammenhang die steuerungstechnische Erfassung des realen Werkstückzustands. Während ifm-mate den Montageprozess unterstützt, liefert die Lichtschranke ein physisches Signal zur Anwesenheit bzw. Freigabe des Werkstücks. Beide Systeme ergänzen sich damit funktional. Das ifm-mate-System dient vor allem der Bedienerführung und Fehlervermeidung bei der Montage, während die SPS über Sensorsignale den Anlagenfluss steuert. [6]

Für zukünftige Erweiterungen wäre eine engere Kopplung zwischen ifm-mate und SPS sinnvoll. Denkbar wäre beispielsweise, dass der Abschluss einer Bedienanweisung oder ein erkannter Montagefehler direkt als Signal in die SPS zurückgeführt wird. Dadurch könnte die Station noch stärker ereignisbasiert arbeiten und wäre weniger auf feste Wartezeiten angewiesen.

5.8 Kameraanbindung an die SPS

Ein weiterer betrachteter Punkt war die Kameraanbindung an die SPS. In der Zukunftsfabrik werden Kameras bereits für verschiedene Aufgaben eingesetzt, beispielsweise zur Objekterkennung beim Bin Picking oder zur Qualitätsprüfung. Dabei erfolgt ein Teil der Bildverarbeitung über externe Systeme. Die SPS erhält anschließend ausgewählte Informationen, beispielsweise Qualitätsmerkmale oder Fertigmeldungen.

Im Rahmen dieser Arbeit wurde untersucht, wie eine direktere bzw. besser strukturierte Kameraanbindung in die bestehende Steuerungslogik eingeordnet werden kann. Eine direkte Kameraanbindung an die SPS bietet den Vorteil, dass Trigger-Signale, Statusinformationen und Ergebnisdaten klarer in die Stationszustände integriert werden können. Dadurch können Prozessschritte eindeutiger gestartet, überwacht und abgeschlossen werden.

Besonders bei der Qualitätsprüfung ist eine zuverlässige Signalauswertung entscheidend. Die Kamera liefert qualitätsrelevante Informationen, auf deren Basis entschieden wird, ob ein Bauteil als Gutteil oder Schlechtteil behandelt wird. Für die nachfolgende Sortierung ist es wichtig, dass diese Informationen eindeutig dem richtigen Werkstück zugeordnet werden. Deshalb muss verhindert werden, dass neue Kameradaten einen noch nicht abgeschlossenen Sortierprozess beeinflussen. Eine saubere Zwischenspeicherung der Qualitätsdaten und eine eindeutige Übergabe an die nächste Station bleiben daher zentrale Anforderungen. [6]

Die Kameraanbindung wurde im Rahmen dieser Arbeit vor allem konzeptionell vorbereitet. Die vollständige Umsetzung sollte in einer späteren Arbeit erfolgen, da hierfür neben der SPS-Konfiguration auch die genaue Geräteparametrierung, die Netzwerkanbindung und die Auswertung der kameraseitigen Statussignale detailliert betrachtet werden müssen.

5.9 Transportroboter HUSKY



Abbildung 5; Transportroboter HUSKY [7]

Der Transportroboter HUSKY ist in das Anlagenkonzept eingebunden, indem fertig montierte Werkstücke auf ihm abgelegt werden. Für einen vollständig

automatisierten Ablauf ist jedoch eine zuverlässige Kommunikation zwischen HUSKY, Industrie-PC und SPS erforderlich.
Besonders relevant ist dabei der Beladungszustand des HUSKY. Die Anlage muss erkennen bzw. weitergeben können, ob der Transportroboter bereits vollständig beladen ist oder ob weitere Werkstücke aufgenommen werden können.

Im Rahmen dieser Arbeit wurde deshalb eine softwareseitige Kommunikationslösung zwischen SPS, Industrie-PC und Transportroboter vorbereitet. Ziel der Umsetzung war es, den in der SPS vorhandenen Beladungszustand Husky_voll an den Roboter zu übertragen. Dadurch kann der Roboter bzw. ein angebundenes System erkennen, ob eine weitere Übergabe zulässig ist oder ob keine weiteren Werkstücke auf dem HUSKY abgelegt werden dürfen.

Für diese Kommunikation wurde ein Python-Skript erstellt, das zwei Schnittstellen miteinander verbindet. Auf der einen Seite wird über die Bibliothek snap7 eine Verbindung zur Siemens-SPS aufgebaut. Auf der anderen Seite wird über pymodbus eine Modbus-TCP-Verbindung zum Roboter hergestellt. Das Skript übernimmt somit die Funktion einer Kommunikationsbrücke zwischen SPS und HUSKY.

Die relevante SPS-Variable befindet sich in einem neu angelegten globalen Datenbaustein. Dafür wurde der Datenbaustein DB67 verwendet, welcher zunächst nur die Bool-Variable Husky_voll enthält. Die Variable liegt an der Adresse DB67.DBX0.0. Durch diese separate Datenstruktur bleibt die Schnittstelle übersichtlich und unabhängig von bestehenden Stationsbausteinen. Gleichzeitig kann der Datenbaustein später um weitere HUSKY-relevante Signale erweitert werden, ohne die vorhandene Steuerungsstruktur unnötig zu verändern.

Das Python-Skript liest den Datenbaustein zyklisch mit einer Abtastzeit von 0,2 s aus. Der aktuelle Wert der Variable Husky_voll wird anschließend mit dem zuletzt

übertragenen Wert verglichen. Eine Übertragung an den Roboter erfolgt beim ersten erfolgreichen Lesevorgang sowie immer dann, wenn sich der Wert geändert hat. Dadurch wird verhindert, dass identische Werte unnötig häufig über Modbus gesendet werden. Diese ereignisorientierte Übertragung reduziert die Kommunikationslast und macht das Verhalten des Skripts nachvollziehbarer.

Die Weitergabe an den Roboter erfolgt über Modbus TCP. Dafür wird eine Verbindung zur Roboter-IP-Adresse 192.168.131.1 über den Standardport 502 aufgebaut. Der Bool-Wert wird auf eine Modbus-Coil-Adresse geschrieben. Im Skript wird hierfür die Coil-Adresse 0 verwendet, was in vielen Modbus-Systemen der logischen Coil 00001 entspricht. Damit wird der SPS-Wert in ein für den Roboter auswertbares digitales Signal überführt.

Ein besonderer Fokus der Umsetzung lag auf der Robustheit der Kommunikation. Sowohl die SPS-Verbindung als auch die Modbus-Verbindung zum Roboter werden im laufenden Betrieb überwacht. Falls eine Verbindung unterbrochen wird oder ein Schreibfehler auftritt, wird die betroffene Verbindung geschlossen und nach einer definierten Wartezeit erneut aufgebaut. Zusätzlich ist das Hauptprogramm durch eine äußere Neustartschleife abgesichert. Dadurch kann das Skript auch nach unerwarteten Laufzeitfehlern automatisch neu starten, ohne dass ein manueller Eingriff erforderlich ist.

Da sich die Schnittstelle der Bibliothek pymodbus zwischen verschiedenen Versionen unterscheiden kann, wurde zusätzlich eine Kompatibilitätsfunktion implementiert. Diese versucht zunächst, den Modbus-Schreibbefehl mit dem Parameter device_id auszuführen. Falls diese Schreibweise von der installierten Version nicht unterstützt wird, wird alternativ der Parameter slave verwendet. Dadurch ist das Skript gegenüber unterschiedlichen Bibliotheksversionen robuster und kann leichter auf verschiedenen Industrie-PCs eingesetzt werden.

Die vollständige praktische Anbindung konnte im Rahmen dieser Arbeit jedoch noch nicht abschließend umgesetzt und unter realen Betriebsbedingungen validiert werden. Grund hierfür ist die noch fehlende stabile Netzwerkverbindung zwischen Industrie-PC und HUSKY. Da der HUSKY über einen eigenen Hotspot verfügt, muss der Industrie-PC in dieses Netzwerk eingebunden werden. Hierfür ist zusätzliche Hardware erforderlich, da der vorhandene Industrie-PC nicht WLAN-fähig ist. Erst nach Einrichtung einer geeigneten Netzwerkverbindung kann das Skript dauerhaft im Anlagenbetrieb getestet und die Reaktion des Roboters auf das übertragene Modbus-Signal überprüft werden. [8]

Das Ergebnis dieser Umsetzung ist somit eine funktionsfähige softwareseitige Grundlage zur Übertragung des HUSKY-Beladungszustands aus der SPS an den Roboter. Die Lösung ist bewusst einfach aufgebaut und nutzt mit DB67.DBX0.0 eine klar definierte SPS-Schnittstelle. Dadurch bleibt die Kommunikation transparent und kann durch zukünftige Arbeiten erweitert werden. Denkbar wäre beispielsweise eine bidirektionale Kommunikation, bei der nicht nur Informationen von der SPS an den Roboter übertragen werden, sondern auch Roboterzustände zurück an die SPS gemeldet werden.

5.10 Dokumentation der Programmstruktur im Wissensmanagement

Neben der reinen technischen Umsetzung wurde die Dokumentation der vorgenommenen Änderungen als eigener Arbeitsschwerpunkt betrachtet. Die Zukunftsfabrik Bodensee wird regelmäßig durch neue Studien- und Projektarbeiten erweitert. Dadurch besteht die Gefahr, dass Wissen über Programmstruktur, Schnittstellen und bekannte Problemstellen verloren geht, wenn Änderungen nur im SPS-Projekt selbst erkennbar sind.

Aus diesem Grund wurde die aktuelle Programmstruktur im Wissensmanagement der Zukunftsfabrik dokumentiert. Dabei wurden insbesondere die grundlegende Architektur aus Schrittkette und Zustandsautomaten, relevante Stationslogiken, vorgenommene Anpassungen sowie offene Punkte festgehalten. Ziel war es, zukünftigen Bearbeitenden eine schnellere Einarbeitung zu ermöglichen und die Nachvollziehbarkeit der Steuerung zu verbessern.

Die Dokumentation besitzt damit einen direkten Einfluss auf die langfristige Effizienz der Anlage. Eine gut dokumentierte Steuerung reduziert nicht nur die Einarbeitungszeit, sondern erleichtert auch die Fehlersuche. Zukünftige Änderungen können gezielter vorgenommen werden, wenn bekannt ist, welche Bausteine für welche Funktionen verantwortlich sind und welche Abhängigkeiten zwischen einzelnen Stationen bestehen.

Besonders wichtig ist dabei die Dokumentation von offenen Punkten. Nicht vollständig abgeschlossene Themen wie die weiterführende HUSKY-Kommunikation, die tiefere Integration der Auftragsverwaltung oder die direkte Kameraanbindung an die SPS wurden entsprechend festgehalten. Dadurch können spätere Arbeiten gezielt an diesen Punkten anknüpfen, ohne den Ausgangszustand erneut vollständig analysieren zu müssen.

5.11 Zeitmessung über Python

Zur Unterstützung zukünftiger Untersuchungen an der Zukunftsfabrik Bodensee wurde eine Python-basierte Zeitmessung implementiert. Ziel dieser Umsetzung ist es, Montagezeiten aus der SPS auszulesen und automatisiert in einer Textdatei zu dokumentieren. Dadurch entsteht eine einfache, nachvollziehbare und weiterverarbeitbare Datengrundlage, die beispielsweise für statistische Auswertungen, Prozessvergleiche oder wissenschaftliche Untersuchungen genutzt werden kann.

Die eigentliche Erfassung der Montagezeit erfolgt innerhalb der SPS. Das Python-Skript übernimmt somit nicht die zeitkritische Messung selbst, sondern liest den bereits berechneten Messwert aus einem definierten Datenbaustein aus. Dieses Vorgehen ist sinnvoll, da die SPS zyklisch und echtzeitnäher arbeitet als ein externes Python-Programm auf einem Industrie-PC. Python dient in diesem Fall als Schnittstelle zur Datenerfassung, Speicherung und späteren Auswertung.

Für die Kommunikation zwischen Industrie-PC und SPS wird die Bibliothek `snap7` verwendet. Diese ermöglicht den Zugriff auf Datenbausteine einer Siemens-SPS über eine S7-Kommunikationsverbindung. Im Skript sind hierfür die IP-Adresse der SPS, Rack und Slot sowie der zu lesende Datenbaustein festgelegt. Die Verbindung wird zu Beginn aufgebaut und während der Laufzeit kontinuierlich überwacht. Falls die Verbindung unterbrochen wird oder ein Lesefehler auftritt, trennt das Programm die bestehende Verbindung, wartet eine definierte Zeit und versucht anschließend automatisch, sich erneut mit der SPS zu verbinden. Dadurch ist ein robuster Dauerbetrieb möglich, ohne dass das Skript bei kurzzeitigen Kommunikationsproblemen manuell neu gestartet werden muss.

Der relevante Datenbereich befindet sich im Datenbaustein `DB45`. Aus diesem werden insgesamt fünf Byte gelesen. Die Montagezeit liegt dabei als `DINT` ab Byte 0 vor, also an der Adresse `DBD0`. Zusätzlich wird an Byte 4, Bit 0 ein boolesches Signal ausgewertet. Dieses Signal dient als Trigger für den Schreibvorgang. Sobald das Signal von `False` auf `True` wechselt, erkennt das Python-Skript eine positive Flanke und speichert den zu diesem Zeitpunkt ausgelesenen Montagezeitwert in einer Textdatei. Die Verwendung einer Flankenauswertung ist dabei wesentlich. Würde lediglich der aktuelle Signalzustand ausgewertet, würde das Skript während eines dauerhaft gesetzten Signals bei jedem Polling-Zyklus erneut denselben Messwert speichern. Durch den Vergleich mit dem vorherigen Signalzustand wird sichergestellt, dass pro Messereignis nur ein einzelner Datensatz geschrieben wird. Damit ist die Datenaufzeichnung eindeutig und frei von Mehrfacheinträgen desselben Ereignisses.

Das Skript liest die SPS-Daten zyklisch mit einer Abtastzeit von `0,2 s` aus. Diese Polling-Zeit stellt einen Kompromiss zwischen Reaktionsgeschwindigkeit und Systembelastung dar. Da die eigentliche Zeitmessung bereits in der SPS erfolgt und Python lediglich den fertigen Messwert abholt, beeinflusst die Polling-Zeit die gemessene Montagezeit nicht direkt. Wichtig ist jedoch, dass das Trigger-Signal in der SPS lange genug ansteht, damit es innerhalb eines Polling-Zyklus zuverlässig erkannt werden kann. Bei sehr kurzen Trigger-Impulsen müsste entweder die Polling-Zeit reduziert oder das Signal in der SPS entsprechend verlängert bzw. gespeichert werden.

Die Messwerte werden in der Datei `montagezeiten.txt` abgelegt. Diese Datei wird automatisch im selben Ordner erzeugt, in dem sich auch das Python-Skript befindet. Beim ersten Start prüft das Programm, ob die Datei bereits existiert. Falls nicht, wird sie mit einer Kopfzeile erstellt. Jeder weitere Messwert wird anschließend zeilenweise angehängt. Die Datei besitzt folgenden Aufbau:

Zeitstempel; ms; s

Jeder Datensatz besteht aus einem Zeitstempel, der Montagezeit in Millisekunden und der umgerechneten Montagezeit in Sekunden. Das Semikolon als Trennzeichen ermöglicht eine einfache Weiterverarbeitung, beispielsweise in Excel, Python, MATLAB oder Statistiksoftware. Durch den Zeitstempel können die Messwerte zusätzlich einzelnen Testläufen oder Produktionssituationen zugeordnet werden.

Ein besonderer Fokus der Umsetzung liegt auf der Betriebssicherheit. Sowohl die Initialisierung der Textdatei als auch der Verbindungsaufbau zur SPS und das Schreiben der Messwerte sind mit Fehlerbehandlungen versehen. Tritt ein Fehler auf, wird dieser in der Konsole ausgegeben, das Programm jedoch nicht beendet. Stattdessen wird nach einer kurzen Wartezeit ein erneuter Verbindungs- bzw. Ausführungsversuch gestartet.

Zusätzlich ist das Hauptprogramm selbst durch eine äußere Neustartschleife abgesichert. Dadurch kann das Skript auch bei unerwarteten Laufzeitfehlern automatisch neu starten.

Zusammenfassend stellt die Python-basierte Zeitmessung eine einfache und robuste Möglichkeit dar, Montagezeiten aus der SPS auszulesen und dauerhaft zu protokollieren. Die Lösung trennt dabei klar zwischen der zeitkritischen Messwerterfassung in der SPS und der nicht zeitkritischen Datenspeicherung auf dem Industrie-PC. Dadurch bleibt die eigentliche Steuerung unabhängig von der Auswertung, während gleichzeitig eine strukturierte Datengrundlage für zukünftige Analysen geschaffen wird.
Für spätere Erweiterungen kann das Skript angepasst werden, um zusätzliche Prozesswerte auszulesen, weitere Trigger-Signale zu verarbeiten oder die Messdaten direkt in eine Datenbank bzw. ein MES-System zu übertragen.

6. Fazit

Die umgesetzten Maßnahmen führen insgesamt zu einer stabileren, besser bedienbaren und besser dokumentierten Anlage. Die Effizienzsteigerung zeigt sich dabei nicht ausschließlich in einer Verkürzung einzelner Prozesszeiten, sondern vor allem in einer verbesserten Prozesssicherheit und Wartbarkeit. Dies entspricht der Zielsetzung der Arbeit, da die Zukunftsfabrik nicht nur als Produktionsanlage, sondern auch als Lern- und Forschungsumgebung betrachtet werden muss.

Durch die Stabilisierung der SPS-Ablaufsteuerung wurden bestehende Fehlerquellen reduziert. Die Einbindung der Lichtschranke am Handarbeitsplatz verbessert die Rückmeldung des realen Prozesszustands und reduziert die Abhängigkeit von rein zeitbasierten Annahmen. Die Anpassung einzelner Timerwerte trägt dazu bei, unnötige Wartezeiten zu verringern, ohne die Prozesssicherheit zu gefährden. Die Überarbeitung des HMI erhöht die Bedienbarkeit und reduziert das Risiko unbeabsichtigter Bedienhandlungen. Gleichzeitig schafft die Dokumentation im Wissensmanagement eine bessere Grundlage für zukünftige Erweiterungen.

Nicht alle betrachteten Themen konnten im Rahmen dieser Arbeit vollständig umgesetzt werden. Insbesondere die vollständige Integration der Auftragsverwaltung in das Zustandsautomaten-System, die direkte Kameraanbindung an die SPS und die Kommunikation mit dem HUSKY bleiben relevante Aufgaben für zukünftige Arbeiten. Dennoch wurden diese Themen analysiert und so vorbereitet, dass sie auf der bestehenden Struktur weitergeführt werden können.

Zusammenfassend wurde die Zukunftsfabrik Bodensee durch die umgesetzten Maßnahmen nicht vollständig neu entwickelt, sondern gezielt weiter stabilisiert und erweitert.

Der erreichte Stand bildet eine belastbare Grundlage für weitere Optimierungen, insbesondere für datenbasierte Auswertungen, eine engere MES-Anbindung und eine stärkere Integration externer Systeme.

Ausblick

Die im Rahmen dieser Arbeit umgesetzten Maßnahmen bilden eine weiterentwickelte Grundlage für den Betrieb und die zukünftige Erweiterung der Zukunftsfabrik Bodensee. Durch die Fehlerbehebung, die Anpassung einzelner Stationsabläufe, die verbesserte Bedienbarkeit über das HMI sowie die etablierte Zeitmessung konnten wichtige Voraussetzungen für einen stabileren und besser auswertbaren Anlagenbetrieb geschaffen werden. Dennoch bestehen weiterhin mehrere Ansatzpunkte, die in zukünftigen Studien- oder Projektarbeiten aufgegriffen werden sollten.

Ein wesentlicher nächster Schritt ist die weitere Eingliederung der Auftragsverwaltung in das bestehende Zustandsautomaten-System. Bisher sind Auftragsdaten, Zählerstände und stationsbezogene Zustände noch nicht vollständig in einer einheitlichen Logik zusammengeführt. Perspektivisch sollte die Auftragsverwaltung stärker auf den Zuständen der einzelnen Stationen aufbauen. Dadurch könnten Zählerfunktionen aus einzelnen Stationsbausteinen ausgelagert und zentral in der Auftragsverwaltung verarbeitet werden. Dies hätte den Vorteil, dass die Stationslogik schlanker und übersichtlicher wird, während auftragsbezogene Informationen an einer zentralen Stelle verwaltet werden. Gleichzeitig würde eine solche Struktur die Wartbarkeit erhöhen, da Änderungen an Zählern, Stückzahlen oder Auftragszuständen nicht an mehreren Stellen im Programm vorgenommen werden müssten.

In diesem Zusammenhang sollte auch die Kopplung zwischen Auftragsverwaltung, HMI und einem möglichen MES weiter betrachtet werden. Sobald ein Auftrag durch ein Manufacturing Execution System an die Anlage übergeben wird, sollte die manuelle Auftragseingabe über das HMI gezielt eingeschränkt oder blockiert werden. Dadurch kann verhindert werden, dass Bedienereingaben am HMI mit automatisch übertragenen Auftragsdaten kollidieren.

Für zukünftige Arbeiten ergibt sich daraus die Aufgabe, eindeutige Prioritäten zwischen MES-Aufträgen und lokalen HMI-Eingaben festzulegen. Eine mögliche Lösung wäre ein Betriebsmodus, in dem zwischen manueller Auftragseingabe und MES-geführtem Betrieb unterschieden wird. Im MES-Betrieb könnten relevante Eingabefelder am HMI gesperrt oder nur noch lesend angezeigt werden, während im manuellen Betrieb weiterhin eine lokale Auftragserstellung möglich bleibt.

Die im Rahmen dieser Arbeit etablierte Zeitmessung stellt ebenfalls einen wichtigen Ausgangspunkt für weitere Untersuchungen dar. Die Messstruktur wurde aufgebaut und getestet, sodass zukünftige Testläufe mit vergleichsweise geringem Aufwand durchgeführt werden können. Besonders im Zusammenhang mit der Doktorarbeit von Herrn Florian Bulander bietet die automatisierte Erfassung von Prozesszeiten einen hohen Mehrwert, da Zeitstudien dadurch einfacher, reproduzierbarer und datenbasiert durchgeführt werden können. Zukünftige Arbeiten könnten darauf aufbauen, indem die Messdaten automatisiert aufbereitet, visualisiert und statistisch ausgewertet werden. Denkbar wäre beispielsweise eine Auswertung einzelner Stationszeiten, Gesamtzykluszeiten, Wartezeiten oder Engpässe im Prozess. Dadurch könnten Optimierungsmaßnahmen nicht nur subjektiv anhand des Anlagenverhaltens, sondern anhand belastbarer Messdaten bewertet werden.

Ein weiterer offener Punkt betrifft einen bestehenden Laufzeitfehler im Anlagenablauf, der aktuell dazu führt, dass die Stopper zusätzlich herunterfahren. Die Behebung dieses Fehlers ist grundsätzlich möglich, sollte jedoch nicht durch eine rein zeitbasierte Behelfslösung erfolgen. Eine Anpassung über zusätzliche Timer würde zwar kurzfristig das beobachtete Verhalten beeinflussen, könnte aber die langfristige Nachvollziehbarkeit und Robustheit der Steuerung verschlechtern. Stattdessen sollte die Ursache des Fehlers genauer analysiert werden. Dabei ist zu prüfen, welche Zustandswechsel, Rücksetzbedingungen oder Signalabhängigkeiten das zusätzliche Absenken der Stopper auslösen. Ziel einer zukünftigen Bearbeitung sollte es sein, den Fehler logisch innerhalb der bestehenden Zustandsautomaten- und Schrittkettensystematik zu beheben.

Dadurch bleibt die Programmstruktur konsistent und es wird vermieden, dass neue Abhängigkeiten oder schwer nachvollziehbare Sonderfälle entstehen.

Auch die Kommunikation mit dem Transportroboter HUSKY bietet weiteres Entwicklungspotenzial. Für einen zuverlässigeren Anlagenbetrieb wäre es sinnvoll, den Zustand des HUSKY stärker in die Steuerungslogik einzubinden. Besonders relevant ist dabei die Übergabe einer Variable, die angibt, ob der HUSKY voll bzw. beladen ist. Diese Information könnte genutzt werden, um Übergabeprozesse gezielter zu steuern und Fehlerzustände zu vermeiden. Da der HUSKY über einen eigenen Hotspot verfügt, muss der Industrie-PC für eine direkte Kommunikation in dieses Netzwerk eingebunden werden. Hierfür wären die Beschaffung und Einrichtung eines geeigneten Wireless Local Area Network (WLAN) - Routers sinnvoll, um eine stabile Verbindung zwischen Industrie-PC, HUSKY und gegebenenfalls weiteren Netzwerkteilnehmern herzustellen.

Als technischer Ansatz für die Kommunikation zwischen Industrie-PC und SPS kann die bereits im Rahmen der Zeitmessung verwendete Python-Kommunikation als Orientierung dienen. Insbesondere das Skript bulander.py zeigt, wie Variablen zwischen Industrie-PC und SPS abgefragt bzw. verarbeitet werden können. Für die Kommunikation mit dem HUSKY könnte ein ähnliches Konzept auf Basis von Python und Modbus geprüft werden. In einer zukünftigen Arbeit müsste dafür untersucht werden, welche Schnittstellen der HUSKY bereitstellt, wie die relevanten Statusinformationen abgefragt werden können und wie diese anschließend zuverlässig an die SPS übergeben werden. Ziel wäre eine saubere Integration des Transportroboters in die vorhandene Anlagenlogik, sodass Übergabezustände nicht mehr manuell oder indirekt behandelt werden müssen.

Darüber hinaus sollte die bestehende Dokumentation im Wissensmanagement kontinuierlich gepflegt und erweitert werden. Da die Zukunftsfabrik regelmäßig durch neue Studienarbeiten weiterentwickelt wird, ist eine aktuelle und verständliche Dokumentation ein wesentlicher Faktor für die langfristige Wartbarkeit der Anlage.

Zukünftige Änderungen an Auftragsverwaltung, HMI, HUSKY-Kommunikation oder Stationslogik sollten daher konsequent im Wissensmanagement hinterlegt werden. Besonders hilfreich wären dabei Übersichten über die aktuelle Programmstruktur, Zustandsdiagramme der Stationen, verwendete Variablen, Schnittstellenbeschreibungen sowie Hinweise zu bekannten Fehlern und offenen Arbeitspunkten.

Zusammenfassend bietet die Zukunftsfabrik Bodensee weiterhin ein hohes Potenzial für technische und organisatorische Weiterentwicklungen. Die nächsten sinnvollen Schritte liegen vor allem in der konsequenten Zentralisierung der Auftragsverwaltung, der vertieften Integration externer Systeme wie MES und HUSKY, der sauberen Behebung verbleibender Laufzeitfehler sowie der weiteren Nutzung der Python-basierten Zeitmessung für wissenschaftliche Auswertungen. Werden diese Punkte in zukünftigen Arbeiten umgesetzt, kann die Anlage nicht nur robuster betrieben, sondern auch besser analysiert, erweitert und für Lehr- sowie Forschungszwecke genutzt werden.

Literaturverzeichnis

[1] Promotorengruppe Kommunikation der Forschungsunion Wirtschaft – Wissenschaft, „Abschlussbericht_Industrie4.0_barrierefrei“. April 2013.
[2] *VDI 5600 Blatt 1*.
[3] „Logikfunktionen / SPS: Aufbau | DNA“. Zugegriffen: 11. Juli 2026. [Online]. Verfügbar unter: https://xplore-dna.net/mod/page/view.php?id=294
[4]

Nutzung von Künstlicher Intelligenz

Im Rahmen dieser Arbeit wurden Künstliche Intelligenz (KI) basierte Werkzeuge benutzt. Tabelle 1 gibt eine Übersicht über die verwendeten Werkzeuge und deren jeweiligen Einsatzzweck.

Tabelle 1; Liste der verwendeten KI-basierten Werkzeuge

Werkzeug	**Beschreibung der Nutzung**
ChatGPT	Grundlagenrecherche, Umformulieren einzelner Textabschnitte