



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Masterarbeit

Benjamin Ostenberg

Smart Manufacturing: Entwicklung und Inbetriebnahme einer vollautomatisierten Messstation mit digitalem Zwilling und SPS- Steuerung

Benjamin Ostenberg

**Smart Manufacturing:
Entwicklung und Inbetriebnahme einer
vollautomatisierten Messstation mit
digitalem Zwilling und SPS-Steuerung**

Masterarbeit eingereicht im Rahmen der Masterprüfung

im Studiengang Produktionstechnik und -management
am Department Maschinenbau und Produktion
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Erstprüfer: Professor Dr. Alexander Koch
Zweitprüfer: Professor Dr. Christian Müller

Abgabedatum: 03.04.2025

Zusammenfassung

Benjamin Ostenberg

Thema der Masterthesis

Smart Manufacturing: Entwicklung und Inbetriebnahme einer vollautomatisierten Messstation mit digitalem Zwilling und SPS-Steuerung

Stichworte

Industrie 4.0, vernetzte Produktion, Automatisierung, digitaler Zwilling, digitale Simulation, Ciro, SPS-Programmierung, Tia-Portal, OPC-UA, digitale Messmethoden, Inbetriebnahme

Kurzzusammenfassung

Diese Arbeit befasst sich mit der Entwicklung und Implementierung einer automatisierten Vermessungsstation unter Einsatz eines digitalen Zwillings. Ziel ist die Automatisierung des Messprozesses, bei dem ein Bauteil transportiert, auf einem Drehtisch positioniert, mit einem 3D-Scanner vermessen und anschließend weitertransportiert wird. Dazu wird zunächst ein digitaler Zwilling in Ciro erstellt, um die Steuerung und Interaktion der Module zu testen. Neben der Modellierung des digitalen Zwillings wird eine SPS-Steuerung entwickelt, die sowohl die virtuelle als auch die physische Anlage steuert. Die Kommunikation zwischen der SPS und dem Messprogramm Polyworks erfolgt über OPC UA.

Nach der Übertragung auf die physische Anlage zeigen Testläufe Optimierungspotenziale in der Automatisierung des Messprozesses, der Stabilität der Softwarekommunikation und der Sicherheitstechnik. Die Ergebnisse bilden eine Grundlage für künftige Weiterentwicklungen und Anwendungen.

Benjamin Ostenberg

Title of the paper

Smart Manufacturing: Development and Commissioning of a Fully Automated Measurement Station with Digital Twin and PLC Control

Keywords

Industry 4.0, networked production, automation, digital twin, digital simulation, Ciro, plc programming, TIA Portal, OPC UA, digital measurement methods, commissioning

Abstract

This study focuses on the development and implementation of an automated measurement station utilizing a digital twin. The goal is to automate the measurement process, in which a component is transported, positioned on a turntable, scanned with a 3D scanner, and subsequently transported further. To achieve this, a digital twin is first created in Ciro to test the control logic and interaction of the modules. In addition to modeling the digital twin, a PLC control system is developed to manage both the virtual and physical setup. Communication between the PLC and the measurement software Polyworks is facilitated via OPC UA.

After transferring the system to the physical setup, test runs reveal optimization potential in the automation of the measurement process, the stability of software communication, and the implementation of safety measures. The findings serve as a foundation for future developments and applications.

Inhaltsverzeichnis

Zusammenfassung	II
Inhaltsverzeichnis	III
Abkürzungsverzeichnis	V
Tabellenverzeichnis	VI
Abbildungsverzeichnis	VII
1 Einleitung	1
1.1 Problematik und Zielsetzung	3
1.2 Aufbau der Arbeit	4
2 Theoretische Grundlagen	6
2.1 Industrie 4.0 und vernetzte Produktion	6
2.2 Digitaler Zwilling	7
2.2.1 Definition und Einsatzbereiche	7
2.2.2 Vorteile und Herausforderungen.....	8
2.2.3 Ciros von Festo als Simulationssoftware	9
2.3 Automatisierung und SPS-Programmierung	10
2.3.1 Grundlagen der SPS-Programmierung.....	11
2.3.2 Tia Portal von Siemens	19
2.3.3 OPC UA - Schnittstelle für industrielle Kommunikation	23
2.4 Optische Messtechnik und Qualitätssicherung	25
2.4.1 Grundlagen der automatisierten Messtechnik.....	25
2.4.2 Bedeutung der Qualitätssicherung in vernetzten Produktionssystemen.....	26
3 Planung und Konzept der Vermessungsstation	28
3.1 Anforderungen an die Vermessungsstation	28
3.1.1 Anforderungen an den digitalen Zwilling.....	28
3.1.2 Anforderungen an die physische Vermessungsstation	29
3.2 Konzeptentwicklung für die Messstation	30
3.2.1 Mechanische Komponenten	31
3.2.2 Elektronische und pneumatische Komponenten	34
3.2.3 Integration des 3D-Scanners und Messtechnik	36
4 Entwicklung des digitalen Zwillings in Ciros	38
4.1 Erstellung des CAD-Modells der Vermessungsstation.....	38
4.2 Aufbau und Konfiguration des digitalen Zwillings in Ciros.....	46
4.2.1 Integration und Erweiterung der Komponenten in Ciros	47
4.2.2 Definition der Kinematiken, Mechanismen und Steuerungsschnittstellen	52

4.2.3 SPS-Programmierung im TIA-Portal und Integration in CiroS	58
4.3 Test des digitalen Zwillings.....	71
5 Implementierung der realen Vermessungsstation	77
5.1 Aufbau und Montage der Station	77
5.1.1 Montage des Gestells.....	77
5.1.2 Verschaltung der Module.....	78
5.2 Steuerung der physischen Vermessungsstation	81
5.2.1 Ansteuerung des Messprogrammes in Polyworks	81
5.2.2 OPC UA Server und -Client.....	84
5.2.3 SPS-Programm der realen Vermessungsstation	85
5.3 Test der physischen Station	87
6 Ergebnisanalyse und Vergleich der Testdurchläufe	93
6.1 Vergleich der Ergebnisse zwischen digitalem Zwilling und physischen Zwilling	93
6.2 Bewertung der Funktionalität der realen Station	95
6.3 Optimierungspotenziale und Erkenntnisse aus den Tests.....	97
7 Schlussfolgerungen und Ausblick	100
7.1 Zusammenfassung der Arbeit.....	100
7.2 Weiterentwicklung und zukünftige Anwendungen.....	101
Literaturverzeichnis	VIII
Anhang.....	X
A: Stationsbauteile-Liste	X
B: Grafcet - digitaler Zwilling	XI
C: Polyworks-Makro.....	XII
D: OPC UA Client – Python Skript	XIV
E: Grafcet – physische Station.....	XVI
F: Physische Station – Schrittkette.....	XVII
G: Physische Station – Befehlsausgabe	XXIV

Abkürzungsverzeichnis

(SR)-Baustein	<i>Set Reset - Baustein</i>
AS	<i>Ablaufsprache</i>
AWL	<i>Anweisungsliste</i>
CAD	<i>Computer Aided Design</i>
CPS	<i>Cyber-Physische Systeme</i>
CPU	<i>Central Processing Unit</i>
DB	<i>Datenbaustein</i>
DCOM	<i>Distributed Component Object Model</i>
DP	<i>Profibus</i>
EEPROM	<i>Electrically Erasable Programmable Read-Only Memory</i>
FB	<i>Funktionsbaustein</i>
FC	<i>Funktionen</i>
FUP	<i>Funktionsplan</i>
GDB	<i>Globale Datenbausteine</i>
HAW	<i>Hochschule für Angewandte Wissenschaften</i>
HMI	<i>Human Machine Interface</i>
IDB	<i>Instanz Datenbaustein</i>
IoT	<i>Internet of Things</i>
IPT	<i>Institut für Produktionstechnik</i>
<i>Item</i>	<i>Industrietechnik und Maschinenbau</i>
KOP	<i>Kontaktplan</i>
KRL	<i>Kuka Robot Language</i>
MDF	<i>Mitteldichte Faserplatte</i>
MES	<i>Manufacturing Execution System</i>
OB	<i>Organisationsbausteine</i>
PLC	<i>Programmable Logic Controller</i>
PN	<i>Profinet</i>
RFID	<i>Radio-Frequency-Identification</i>
SCL	<i>Structured Control Language</i>
SMT	<i>System Montage Technik</i>
SPS	<i>Speicherprogrammierbare Steuerung</i>
ST	<i>Strukturierter Text</i>
Tia Portal	<i>Totally Integrated Automation Portal</i>

Tabellenverzeichnis

Tabelle 1: Eingänge und Ausgänge der Module in Ciroc	57
Tabelle 2: Tia-Portal - Adressenbeschreibung	63
Tabelle 3: Kontaktbelegungstabelle, Motorcontroller - Modul Band - Stopper	79

Abbildungsverzeichnis

Abbildung 1: mögliche Zustände eines Bits	12
Abbildung 2: Visualisierung eines Bytes	12
Abbildung 3: SPS-Aufbau	14
Abbildung 4: SPS-Programmiersprachen	16
Abbildung 5: KOP-Plan RS Flipflops.....	16
Abbildung 6: Beispiel FUP - Funktionsplan.....	17
Abbildung 7: Beispiel ST- Strukturierter Text	18
Abbildung 8: Beispiel AS - Ablaufsprache.....	19
Abbildung 9: OPC UA vertikale und horizontale Kommunikation	23
Abbildung 10: Modul Band	32
Abbildung 11: Dreh-Hub-Modul	32
Abbildung 12: Drehtisch	33
Abbildung 13: CAD-Grundgerüst - perspektivische Ansicht 1	40
Abbildung 14: CAD-Grundgerüst - Ansicht von unten	41
Abbildung 15: CAD-Grundgerüst - perspektivische Ansicht 2	42
Abbildung 16: CAD-Grundgerüst - perspektivische Ansicht 3	43
Abbildung 17: CAD-Gestell - perspektivische Ansicht 4.....	44
Abbildung 18: CAD-Vermessungsstation - Frontalansicht	45
Abbildung 19: CAD-Vermessungsstation - perspektivische Ansicht	46
Abbildung 20: grundlegendes CAD-Modell für Ciro Import.....	47
Abbildung 21: Ciro- Modell Explorer und Eigenschaftenseite "Lage"	49
Abbildung 22: Positionierung der Festo Module auf der Arbeitsfläche	50
Abbildung 23: Vermessungsstation in Ciro mit Drehtisch und Bauteil	51
Abbildung 24: Achsparameter des Dreh-Hub-Moduls in Ciro	53
Abbildung 25: Mechanismus des Drehtisches in Ciro	54
Abbildung 26: Greiferpriorität eines Greifers in Ciro	56
Abbildung 27: Grafcet-Symboltabelle - digitaler Zwilling	59
Abbildung 28: Auswahl an wichtigen Grafcet-Symbolen.....	60
Abbildung 29: Ausschnitt Grafcet - digitaler Zwilling	61
Abbildung 30: Tia Portal - Hardwarekonfiguration.....	62
Abbildung 31: digitaler Zwilling - Ausschnitt PLC-Variablentabelle.....	63
Abbildung 32: Tia Portal - Ausschnitt Funktionsbaustein "Schrittkette"	65
Abbildung 33: Tia Portal - Funktionsbaustein "Befehlsausgabe".....	66
Abbildung 34: Tia Portal - Organisationsbaustein "OB100".....	67
Abbildung 35: Tia Portal - Organisationsbaustein "Main OB1".....	68
Abbildung 36: PLCSim Advanced - Online Status der Instanz	69
Abbildung 37: Ciro – Eigenschaftenfenster SPS-Konfiguration	70
Abbildung 38: Ciro - SPS-Konfiguration der Ein- & Ausgänge	71
Abbildung 39: Ciro - Simulation, Schritt 1-2	73
Abbildung 40: Ciro - Simulation, Schritt 3-5	74
Abbildung 41: Ciro - Simulation, Schritt 6-8	75
Abbildung 42: Modul Band - Kontaktbelegungstabelle.....	78
Abbildung 43: Dreh-Hub-Modul - Kontaktbelegungstabelle.....	80
Abbildung 44: Polyworks - Ausschnitt Automatisierungs-Makro.....	83
Abbildung 45: Tia Portal - Datenbaustein mit OPC UA Variablen	84
Abbildung 46: Tia Portal - PLC Variablentabelle - reale Station	86

Abbildungsverzeichnis

Abbildung 47: Tia Portal - Befehlsausgabe der OPC UA Variablen in SCL.....	87
Abbildung 48: Test des physischen Zwillings - Bild 1-3.....	89
Abbildung 49: Test des physischen Zwillings - Bild 4-6.....	90
Abbildung 50: Test des physischen Zwillings - Bild 7-9.....	91
Abbildung 51: Test des physischen Zwillings - Bild 10.....	92
Abbildung 52: Beispiel eines Messreports nach Test des physischen Zwillings	96
Abbildung 53: Grafcetbeispiel - Integration eines Reset-Mechanismus	98

1 Einleitung

Die vierte industrielle Revolution, bekannt als Industrie 4.0, markiert einen Wendepunkt in der Geschichte der industriellen Fertigung. Sie verspricht eine beispiellose Transformation der Produktionsprozesse, die weit über die Grenzen einzelner Unternehmen hinausreicht und ganze Wertschöpfungsketten revolutioniert. Diese Entwicklung steht für eine fundamentale Veränderung der Art und Weise, wie wir produzieren, arbeiten und leben. Laut Deckert umfasst Industrie 4.0 wesentliche Aspekte wie Digitalisierung, Vernetzung und den Einsatz von (künstlicher) Intelligenz.¹ Diese drei Säulen ermöglichen es, Produktionsprozesse nicht nur effizienter zu gestalten, sondern auch flexibler auf individuelle Kundenanforderungen einzugehen. Dies wird durch Technologien wie das Internet der Dinge (IoT), cyber-physische Systeme (CPS) und Smart Factories erreicht, die nahtlos miteinander interagieren. Industrie 4.0 ist somit nicht nur ein technologisches Konzept, sondern auch ein Treiber für tiefgreifende Veränderungen in Wirtschaft, Gesellschaft und Bildung, indem es neue Ansätze für Arbeitsmodelle und Qualifikationen erfordert.²

Die Vernetzung von Systemen ist ein zentraler Aspekt der Industrie 4.0 und spielt eine entscheidende Rolle bei der Realisierung intelligenter Fabriken. Vernetzte Systeme ermöglichen die Kommunikation und den Datenaustausch zwischen Maschinen, Sensoren und anderen Komponenten einer Produktionsumgebung. Diese Systeme schaffen die Grundlage für Cyber-Physische Systeme, in denen physikalische Prozesse mit digitalen Systemen verschmelzen und eine dynamische und dezentrale Steuerung der Produktionsabläufe ermöglichen. Die Vorteile dieser Vernetzung liegen in einer gesteigerten Transparenz und Steuerbarkeit von Produktionsprozessen, die zu einer höheren Effizienz und Flexibilität führen.³

Im Kontext von Industrie 4.0 kommt auch der digitalen Messtechnik eine immer größere Bedeutung zu. Optische Messsysteme, die kontaktlose und zerstörungsfreie Prüfmethode ermöglichen, spielen dabei eine Schlüsselrolle. Diese Systeme sind in der Lage, hochpräzise Messungen durchzuführen und bieten den Vorteil, dass sie auch in sensiblen oder schwer zugänglichen Umgebungen eingesetzt werden können. Sie tragen maßgeblich zur Qualitätssicherung und Prozessoptimierung bei, indem sie die Digitalisierung und Automatisierung von Mess- und Kontrollprozessen unterstützen. Besonders in der Industrie 4.0 wird die Fähigkeit optischer Systeme hervorgehoben, Messdaten direkt zu erfassen, zu analysieren und in den digitalen Produktionsfluss zu integrieren.⁴

Ein Schlüsselkonzept der modernen Produktion ist der digitale Zwilling, der als präzises, virtuelles Abbild physischer Objekte oder Systeme dient. „Der digitale Zwilling kann als eine 1-zu-1-Repräsentation eines realen Zwillings verstanden werden, die digitale Artefakte entlang des Lebenszyklus des realen Objekts integriert und so eine fortschreitende Konvergenz zwischen virtueller und realer Welt ermöglicht.“⁵ Durch die Anwendung von Simulationstechniken wird der digitale Zwilling „experimentierbar“, was es erlaubt, verschiedene Szenarien zu analysieren und Optimierungen vorzunehmen. Diese Eigenschaften eröffnen neue Möglichkeiten in der

¹ Vgl. Deckert, R. (2019), S. 1.

² Vgl. Schwab, K. (2016), S. 15–16.

³ vgl. Langmann, R. (2021), S. 18.

⁴ vgl. Schuth, M. / Buerakov, W. (2017), S. 6–7.

⁵ vgl. Schluse, M. (2024), S. 62.

vorausschauenden Wartung, der virtuellen Inbetriebnahme sowie der Optimierung von Produktionsprozessen.⁶

Die Automatisierungstechnik bildet das Rückgrat moderner Produktionssysteme und gilt als treibender Motor für Effizienzsteigerung, Flexibilität und Individualisierung in industriellen Prozessen. Sie ermöglicht es Unternehmen, schnell auf Marktveränderungen zu reagieren, Kosten zu senken und zugleich neue, innovative Produkte zu entwickeln. Ein Schlüsselprinzip der Automatisierung ist die Integration von Technologien und die Schaffung vernetzter Systeme, die physische Prozesse mit digitalen Steuerungen kombinieren, um eine nahtlose Interaktion zwischen Mensch und Maschine zu gewährleisten.⁷

Eine zentrale Rolle innerhalb der Automatisierung nimmt die speicherprogrammierbare Steuerung (SPS) ein. Sie ist das Bindeglied zwischen Sensoren, Aktoren und übergeordneten IT-Systemen und dient der präzisen Steuerung und Überwachung industrieller Prozesse. SPS-Systeme wie die SIMATIC S7 von *Siemens* bieten eine hohe Zuverlässigkeit und Flexibilität, da sie sowohl einfache Steuerungsaufgaben als auch komplexe Automatisierungssysteme unterstützen können. Durch ihre modulare Struktur und die Möglichkeit, sie an spezifische Anforderungen anzupassen, tragen sie entscheidend zur Effizienzsteigerung und Skalierbarkeit von Produktionsprozessen bei.⁸

Die Automatisierungstechnik wird auch in Zukunft unsere Arbeits- und Lebenswelt prägen, indem sie zunehmend intelligente, vernetzte Produkte hervorbringt und eine fortschreitende Digitalisierung der Industrie ermöglicht.⁹

Die Vision von Prof. Dr.-Ing. Alexander Koch an der Hochschule für Angewandte Wissenschaften Hamburg (HAW Hamburg) ist der Aufbau einer vollständig vernetzten, smarten Produktionsumgebung, die exemplarisch für die Prinzipien der Industrie 4.0 steht. Diese Umgebung umfasst autonom fahrende Transportsysteme, die Bauteilträger zwischen den Fertigungsstationen bewegen, unterschiedlichste Fertigungsverfahren zur automatisierten Herstellung von Pneumatikzylindern und hochpräzise Qualitätskontrollsysteme zur Sicherstellung der Produktqualität. Die zentral gesteuerte Anlage soll durch ein Manufacturing Execution System (MES) flexibel, effizient und anpassungsfähig sein, um die Anforderungen moderner Produktionssysteme zu erfüllen.

Ein besonderer Fokus dieser smarten Produktionsumgebung liegt auf ihrer Eignung für die Lehre und Schulung von Studierenden. Sie bietet eine praxisnahe Plattform, auf der die Prinzipien der Automatisierung, Vernetzung und Digitalisierung vermittelt werden können. Zudem ermöglicht sie die Umsetzung studentischer Projekte und wissenschaftlicher Arbeiten, wodurch innovative Ideen direkt in die smarte Fabrik integriert werden können.

Eine zentrale Rolle innerhalb dieser smarten Produktionsumgebung spielt die in dieser Arbeit entwickelte und umgesetzte Scanner Station, die als automatisierte Vermessungsstation zur Qualitätskontrolle konzipiert werden soll. Diese Station wird nicht nur physisch aufgebaut, sondern auch als digitaler Zwilling realisiert, um die Prinzipien der Simulation und Optimierung in der Produktionskette exemplarisch zu demonstrieren. Durch die Integration in das Gesamtkonzept der smarten Produktion soll die Scanner Station mit anderen Stationen und Fördersystemen vernetzt werden und trägt so zur durchgängigen Digitalisierung und Automatisierung der Gesamtanlage bei.

⁶ vgl. Schluse, M. (2024), S. 91–93.

⁷ vgl. Weyrich, M. (2023), S. 18–19.

⁸ vgl. Kaftan, J. (2015), S. 23.

⁹ vgl. Weyrich, M. (2023), S. 18.

Dieses Projekt zur Entwicklung der Vermessungsstation steht exemplarisch für die Möglichkeiten, die Industrie 4.0 bietet. Es kombiniert Automatisierung, digitale Zwillinge und vernetzte Systeme, um die Produktionsqualität zu steigern und flexible, zukunftsfähige Produktionsanlagen zu schaffen. Die Scanner Station ist damit ein wichtiger Meilenstein auf dem Weg zur Realisierung der smarten Produktion in der HAW Hamburg, die nicht nur durch eine höhere Produktivität, verbesserte Qualität und gesteigerte Effizienz überzeugt, sondern auch eine wertvolle Grundlage für die Ausbildung und Weiterentwicklung zukünftiger Ingenieurinnen und Ingenieure bietet.

1.1 Problematik und Zielsetzung

In der heutigen industriellen Landschaft wird der Begriff Industrie 4.0 oft als Synonym für die fortschreitende Digitalisierung und Vernetzung von Produktionssystemen verwendet. Diese Entwicklung birgt großes Potenzial, bringt jedoch auch erhebliche Herausforderungen mit sich. Der Aufbau und die Umsetzung von Konzepten wie einer vollständig vernetzten und automatisierten Produktionsumgebung sind komplexe Aufgaben, die fundiertes Wissen, innovative Ansätze und eine präzise Planung erfordern.

Ein zentraler Aspekt dieser Arbeit ist die Entwicklung einer Scanner-Station, die als Teil einer smarten Produktionsumgebung an der Hochschule für Angewandte Wissenschaften Hamburg realisiert werden soll. Diese Station muss nicht nur physisch aufgebaut, sondern auch als digitaler Zwilling abgebildet werden, um die Simulation, Steuerung und Optimierung innerhalb einer vernetzten Produktionsumgebung zu demonstrieren. Als erste Station dieser Smart Factory stellt sie die Grundlage für eine weiterführende Integration und Vernetzung dar. Dabei ist zu beachten, dass die Höhe der Förderbänder bereits vorgegeben ist, um die Station später an weitere geplante Produktionsstationen anschließen zu können.

Die Hauptaufgabe dieser Station besteht darin, ein zuvor gefertigtes Bauteil – einen Pneumatikzylinder – aus einem Vorprozess über ein Transportband zu übernehmen, es auf einem Drehtisch zu positionieren und mittels eines 3D-Scanners zu vermessen. Nach der Vermessung soll das Bauteil wieder auf das Transportband abgelegt und zu den nachfolgenden Prozessen weitergeleitet werden. Dieser vollautomatisierte Ablauf erfordert eine reibungslose Schnittstellenkommunikation zwischen den einzelnen Modulen, eine präzise Steuerung der Bewegungsabläufe sowie eine zuverlässige Datenverarbeitung zwischen Steuerung und Messsoftware.

Die größte Herausforderung liegt in der Komplexität der Anforderungen, die an die Scanner-Station gestellt werden. Mechanisch muss sie stabil und präzise konstruiert sein, um alle Funktionalitäten zuverlässig auszuführen. Gleichzeitig erfordert die Erstellung des digitalen Zwillings eine detaillierte Abbildung nicht nur der mechanischen Struktur, sondern auch der Bewegungsabläufe und der Steuerungslogik in einer virtuellen Umgebung.

Ein weiterer entscheidender Aspekt ist die Entwicklung der SPS-Steuerung, die sowohl im digitalen Zwilling als auch in der realen Station implementiert werden muss. Die Steuerung muss in der Lage sein, sämtliche Prozessschritte zu koordinieren, Sensoren und Aktoren effizient anzusteuern und flexibel auf Erweiterungen zu reagieren. Zudem ist die Kommunikation zwischen der SPS und der Messsoftware sicherzustellen, wofür ein geeignetes Kommunikationsprotokoll gefunden werden muss.

Die Zielsetzung dieser Arbeit ist es, die Scanner-Station als automatisierte Vermessungsstation erfolgreich zu entwickeln, in Betrieb zu nehmen und als ersten Baustein der Smart Factory der HAW Hamburg zu etablieren. Dabei umfasst das Projekt folgende zentrale Aufgaben:

- Erstellung eines digitalen Zwillings in der Simulationssoftware Ciros, der die mechanischen Bewegungsabläufe und die SPS-Logik abbildet.
- Physischer Aufbau der Station, einschließlich aller notwendigen mechanischen, elektrischen und softwareseitigen Komponenten, unter Berücksichtigung der vorgegebenen Abmessungen und Funktionalitäten.
- Entwicklung und Implementierung der SPS-Steuerung mithilfe von TIA-Portal, um die Steuerung sowohl im digitalen Zwilling als auch in der realen Station zu ermöglichen.
- Automatisierung des Messprozesses, sodass das Bauteil auf einem Drehtisch positioniert, vermessen und anschließend weitertransportiert wird.
- Sicherstellung einer zuverlässigen Kommunikation und Datenverarbeitung, insbesondere durch den Einsatz einer geeigneten Technologie zur Schnittstellenkommunikation zwischen Steuerung und Messsoftware.

Mit dieser Arbeit soll nicht nur eine technische Lösung geschaffen, sondern auch ein Modellprojekt für die smarte Produktion der Zukunft realisiert werden. Die entwickelte Scanner-Station dient als Plattform für Lehre, Forschung und zukünftige studentische Projekte. Sie trägt damit zur praktischen Umsetzung der Industrie 4.0-Vision bei, indem sie Automatisierung, digitale Zwillinge und vernetzte Systeme vereint und praxisnah demonstriert.

1.2 Aufbau der Arbeit

Diese Arbeit ist in sieben Kapitel unterteilt, die eine systematische Herangehensweise an die Entwicklung, Implementierung und Analyse einer vollautomatisierten Vermessungsstation verfolgen. Die Struktur der Arbeit ermöglicht eine schrittweise Einführung in die theoretischen Grundlagen, die praktische Umsetzung sowie die Bewertung der Testergebnisse und Optimierungspotenziale.

Das erste Kapitel bildet die **Einleitung** und dient dazu, das übergeordnete Thema der Arbeit zu erläutern. Dabei wird auf die Problematik und Zielsetzung eingegangen, die den Kontext für die Entwicklung der Vermessungsstation im Rahmen von Industrie 4.0 bildet. Die Bedeutung des digitalen Zwillings und der Automatisierung in modernen Produktionsprozessen wird dabei herausgestellt.

Im zweiten Kapitel werden die **theoretischen Grundlagen** der Arbeit behandelt. Dazu zählen zentrale Konzepte wie die vernetzte Produktion im Sinne von Industrie 4.0, die Automatisierungstechnik sowie der Stellenwert des digitalen Zwillings. Darüber hinaus werden die Grundlagen der SPS-Programmierung, die verwendete Simulationssoftware Ciros und das OPC UA-Kommunikationsprotokoll erläutert. Die Einbindung optischer Messtechnik in automatisierte Systeme wird ebenfalls thematisiert, um die technische Basis für die spätere Implementierung der Vermessungsstation zu schaffen.

Das dritte Kapitel beschreibt die **Planung und Konzeption der Vermessungsstation**. Hier werden die Anforderungen sowohl an den digitalen Zwilling als auch an die physische Station definiert. Anschließend wird das Konzept für den Aufbau, die Steuerung sowie die Integration der verschiedenen Module vorgestellt. Dabei werden mechanische, elektronische und softwareseitige Aspekte betrachtet, um eine möglichst realitätsnahe Abbildung des Messprozesses sicherzustellen.

Im vierten Kapitel erfolgt die **Entwicklung des digitalen Zwillings in Ciro**s. Zunächst wird das CAD-Modell der Vermessungsstation erstellt und für die Simulation in Ciro vorbereitet. Anschließend werden die Komponenten integriert, Mechanismen definiert und die Steuerungsschnittstellen eingerichtet. Die SPS-Programmierung wird dabei mit der Simulationssoftware verknüpft, um den digitalen Zwilling funktionsfähig zu machen. Abschließend wird der digitale Zwilling getestet, um die Interaktion zwischen den Modulen und die Steuerungslogik zu überprüfen.

Das fünfte Kapitel befasst sich mit der **Implementierung der realen Vermessungsstation**. Nach dem Aufbau der physischen Station werden die einzelnen Module verkabelt und die Steuerung angepasst. Dabei werden die Unterschiede zur Simulation analysiert, insbesondere im Hinblick auf die Integration des Messprogramms und die Kommunikation über OPC UA. Zudem wird das überarbeitete SPS-Programm beschrieben, bevor abschließend die Tests an der realen Station durchgeführt werden.

Im sechsten Kapitel folgen die **Ergebnisanalyse und der Vergleich der Testdurchläufe**. Hier wird untersucht, inwieweit sich der digitale Zwilling mit der physischen Station deckt und welche Abweichungen festgestellt werden. Die Funktionalität der realen Station wird bewertet und die gewonnenen Erkenntnisse zusammengefasst. Abschließend werden Optimierungspotenziale identifiziert, um die Station weiterzuentwickeln und ihre Leistungsfähigkeit zu steigern.

Das siebte und letzte Kapitel umfasst die **Schlussfolgerungen und den Ausblick**. Es fasst die wesentlichen Erkenntnisse der Arbeit zusammen und zeigt Perspektiven für zukünftige Entwicklungen auf. Dabei wird sowohl auf mögliche Erweiterungen der Station als auch auf den potenziellen Einsatz in weiteren industriellen Anwendungen eingegangen.

Diese strukturierte Vorgehensweise ermöglicht eine fundierte Analyse der Entwicklung der Vermessungsstation – von der theoretischen Konzeption über die praktische Umsetzung bis hin zur abschließenden Bewertung und Weiterentwicklung.

2 Theoretische Grundlagen

In diesem Kapitel werden die theoretischen Grundlagen vorgestellt, die die Basis für die Entwicklung und Umsetzung der Scanner Station im Rahmen dieser Arbeit bilden. Dabei werden die zentralen Konzepte und Technologien der Industrie 4.0 erläutert, darunter die Vernetzung von Produktionssystemen, der digitale Zwilling, die Automatisierungstechnik sowie die Messtechnik. Diese Grundlagen dienen als Ausgangspunkt für die technische und konzeptionelle Entwicklung der Scanner Station und ermöglichen ein tieferes Verständnis der eingesetzten Methoden und Werkzeuge.

2.1 Industrie 4.0 und vernetzte Produktion

Die vierte industrielle Revolution, bekannt als Industrie 4.0, hat ihren Ursprung im Jahr 2006, als der Begriff erstmals auf dem nationalen IT-Gipfel der deutschen Bundesregierung am Hasso-Plattner-Institut (HPI) in Potsdam erwähnt wurde.¹⁰ Maßgeblich geprägt wurde dieser Begriff von Prof. Dr. Henning Kagermann, dem Präsidenten der Deutschen Akademie der Technikwissenschaften. Das übergeordnete Ziel der Bundesregierung war und ist es, die Position Deutschlands als führenden Standort für Informationstechnologie im globalen Wettbewerb zu festigen und weiter auszubauen. Ein entscheidender Schritt zur Verwirklichung dieser Vision war die Gründung der nationalen Plattform Industrie 4.0 im April 2013 auf der Hannover Messe.

Industrie 4.0 stellt eine neue Ära der Organisation und Steuerung der gesamten Wertschöpfungskette dar, die alle Phasen des Lebenszyklus eines Produkts umfasst. Dieser Lebenszyklus orientiert sich zunehmend an den individuellen Bedürfnissen der Kunden und reicht von der ersten Idee und dem Auftrag über die Entwicklung und Fertigung bis hin zur Auslieferung des Produkts an den Endkunden sowie dem Recycling, einschließlich aller damit verbundenen Dienstleistungen.¹¹

Im Kern beinhaltet Industrie 4.0 mehrere Schlüsselkomponenten. Eine dieser Komponenten ist die Digitalisierung der Produktion, wie sie von der Plattform Industrie 4.0 verstanden wird. Zudem spielt die umfassende Vernetzung sowohl horizontal als auch vertikal entlang der gesamten Wertschöpfungskette eine entscheidende Rolle, da sie eine Verlagerung der Steuerung von zentralen zu dezentralen Einheiten ermöglicht.¹² Eine weitere wesentliche Eigenschaft ist die Verfügbarkeit aller relevanten Informationen in Echtzeit, die durch die Vernetzung aller an der Wertschöpfung beteiligten Instanzen gewährleistet wird. Diese Informationen ermöglichen es, den optimalen Wertschöpfungsfluss abzuleiten. Darüber hinaus entstehen durch Industrie 4.0 dynamische Wertschöpfungsnetzwerke, die Menschen, Objekte und Systeme miteinander verbinden und sich in Echtzeit selbst optimieren. Diese Netzwerke können sich flexibel an verschiedene Kriterien wie Kosten, Verfügbarkeit und Ressourcenverbrauch anpassen.¹³

Ein besonders wichtiger Aspekt von Industrie 4.0 ist die vernetzte Produktion. In solchen Systemen kommunizieren digitalisierte, physische Komponenten über das Internet miteinander und bilden sogenannte Cyber-Physical Systems (CPS). Diese optimierten und vernetzten

¹⁰ vgl. Deckert, R. (2019), S. 21.

¹¹ vgl. Westkämper, E. / Löffler, C. (2016), S. 6.

¹² vgl. Andelfinger, V.P. / Hänisch, T. (2017), S. 44.

¹³ vgl. Westkämper, E. / Löffler, C. (2016), S. 138.

Produktionssysteme und -konzepte sind ebenfalls mit dem Internet verbunden und interagieren sowohl mit intelligenten Materialien als auch untereinander. Die Integration von Radio-Frequency-Identification (RFID)-Chips ermöglicht es, Materialien „intelligent“ zu machen.

Die Vernetzung von Geräten, Maschinen und Anlagen spielt eine zentrale Rolle in der Automatisierung und gewinnt durch Industrie 4.0 zunehmend an Bedeutung. Die Kommunikation zwischen Cyber-Physical Systems bildet die technologische Grundlage für Industrie 4.0-Lösungen und stellt sicher, dass Prozesse effizienter, flexibler und transparenter gestaltet werden können.¹⁴

Die Bedeutung von Industrie 4.0 und einer vernetzten, intelligenten Produktion in der heutigen Zeit ist immens. Einer der wichtigsten Aspekte ist die Stärkung des Wirtschaftsstandorts Deutschland durch die Integration der Produktion mit moderner Informations- und Kommunikationstechnik. Industrie 4.0 überträgt das Konzept des Internets der Dinge und Dienste auf das Fabrikumfeld. Das Hauptziel dieser Entwicklung ist es, den internationalen Wettbewerbsvorteil Deutschlands in der industriellen Produktion zu sichern und durch eine nachhaltige Strategie weiter auszubauen.¹⁵ Eine Studie prognostiziert ein zusätzliches Wertschöpfungspotenzial von 78 Milliarden Euro bis zum Jahr 2025 in verschiedenen Branchen, was einem jährlichen Wachstum von 1,7 Prozent entspricht.

Darüber hinaus ermöglicht Industrie 4.0 wirtschaftlichere und ressourcenschonendere Prozesse und schafft die Grundlage für neue Geschäftsfelder. Unternehmen erhalten dadurch einen erweiterten Gestaltungsspielraum, der es ihnen erlaubt, innovative Produkte und Dienstleistungen zu entwickeln. Die Entwicklungen im Bereich Industrie 4.0 stellen einen revolutionären Wandel dar, der maßgeblich von der Logistik vorangetrieben wird.¹⁶

2.2 Digitaler Zwilling

Der digitale Zwilling ist ein zentrales Konzept der Industrie 4.0 und bezeichnet die virtuelle Abbildung eines physischen Objekts oder Systems. Er ermöglicht die Simulation, Überwachung und Optimierung von Prozessen entlang des gesamten Lebenszyklus eines Produkts. In den folgenden Unterkapiteln wird zunächst eine grundlegende Definition des digitalen Zwillings und seine Einsatzbereiche (Kapitel 2.2.1) gegeben, bevor dessen Vorteile und Herausforderungen (Kapitel 2.2.2) sowie eine spezifische Simulationssoftware, Ciro von Festo, als Beispiel für die Implementierung eines digitalen Zwillings (Kapitel 2.2.3) vorgestellt werden.

2.2.1 Definition und Einsatzbereiche

Ein digitaler Zwilling wird häufig als die virtuelle Repräsentation eines physischen Systems, Objekts oder Prozesses definiert, die mit Echtzeitdaten verknüpft ist und das Verhalten seines realen Gegenstücks über den gesamten Lebenszyklus hinweg simuliert und optimiert. Laut Rückert und Sauer kann ein digitaler Zwilling unterschiedliche technische Aspekte wie zum Beispiel mechanische, thermische oder pneumatische Eigenschaften abbilden. Diese Abbildung

¹⁴ vgl. Reinheimer, S. (2017), S. 49.

¹⁵ vgl. Reinheimer, S. (2017), S. 153–154.

¹⁶ vgl. Andelfinger, V.P. / Hänisch, T. (2017), S. 26.

dient nicht nur der Veranschaulichung, sondern ermöglicht auch präzise Analysen und Optimierungen während der Entwicklungs- und Betriebsphase eines Produkts.

Die Einsatzbereiche des digitalen Zwillings sind vielfältig und reichen von der Produktentwicklung über die Produktionsplanung bis hin zur vorausschauenden Wartung und Optimierung von Produktionsanlagen. Rückert und Sauer heben hervor, dass digitale Zwillinge in der Praxis vor allem dazu genutzt werden, physikalische oder technische Systeme virtuell zu testen, um frühzeitig Fehler zu erkennen und die Entwicklungskosten zu senken. Die Anwendung von digitalen Zwillingen wird auch im Kontext von Industrie 4.0 immer relevanter, da sie zur Integration von cyber-physischen Systemen in vernetzte Produktionsumgebungen beitragen können.

Digitale Zwillinge eignen sich insbesondere für komplexe Systeme, bei denen reale Tests teuer oder schwierig umzusetzen sind. Beispiele hierfür sind die Optimierung von Produktionsabläufen, die Simulation des Verschleißverhaltens von Maschinen oder die Schulung von Mitarbeitern in einer virtuellen Umgebung. Somit bieten digitale Zwillinge einen erheblichen Mehrwert für die Effizienzsteigerung, Fehlervermeidung und die Qualitätssicherung in der modernen Industrie.¹⁷

2.2.2 Vorteile und Herausforderungen

Die Einführung von digitalen Zwillingen und Simulationstechnologien hat neue Möglichkeiten für virtuelle Tests und die Optimierung von Produktionsprozessen geschaffen. Als virtuelle Repräsentation physischer Systeme bieten digitale Zwillinge in Kombination mit fortschrittlichen Simulationen zahlreiche Vorteile, stehen jedoch auch vor einigen Herausforderungen.

Vorteile von Digitalen Zwillingen

1. **Testen und Untersuchen vor dem Einsatz:** Mit einem digitalen Zwilling können technische, physikalische oder biologische Komponenten schon lange vor dem eigentlichen Einsatz getestet werden. Es kann untersucht werden, wie sie funktionieren und sich verhalten.¹⁸
2. **Reduzierung des Testaufwands:** Der in einer Virtuellen Inbetriebnahme eingesetzte Digitale Zwilling bildet die Basis für die Entwicklung einer Effizienzmethode, um den Testaufwand durch Automatisierung zu reduzieren.
3. **Beherrschbarkeit der Testkomplexität:** Durch Testautomatisierung am digitalen Zwilling kann die Testkomplexität beherrschbar gemacht werden.
4. **Frühzeitige Behebung von Softwarefehlern:** Methodische Testplanung und -durchführung am digitalen Zwilling ermöglichen die frühzeitige Behebung etwaiger Softwarefehler.
5. **Qualitätssteigerung der Software:** Der Einsatz von digitalen Zwillingen in der Testautomatisierung kann die Qualität der Software deutlich steigern.¹⁹
6. **Reduzierte Integrationszeit:** Der Einsatz von digitalen Zwillingen kann zu einer Reduzierung der Integrationszeit führen.

¹⁷ vgl. Rückert, F.U. / Sauer, M. (2021), S. 1–3.

¹⁸ vgl. Rückert, F.U. / Sauer, M. (2021), S. 1.

¹⁹ vgl. Verl, A. et al. (2024), S. 192.

7. Verkürzte Lieferzeiten: Digitale Zwillinge können zu verkürzten Lieferzeiten beitragen.
8. Image durch Referenzen: Erfolgreiche Implementierungen von digitalen Zwillingen können das Image eines Unternehmens verbessern.²⁰

Herausforderungen von Digitalen Zwillingen

1. **Komplexität der Integration:** Die Implementierung eines digitalen Zwillings erfordert eine nahtlose Verbindung zwischen physischen Systemen und virtuellen Modellen. Dies setzt technisches Know-how und eine robuste IT-Infrastruktur voraus, was insbesondere für kleine und mittlere Unternehmen eine Herausforderung darstellt.
2. **Datenqualität und -management:** Die Genauigkeit von Simulationen hängt stark von der Qualität der verwendeten Daten ab. Unvollständige oder fehlerhafte Daten können zu falschen Ergebnissen führen und das Vertrauen in die Technologie beeinträchtigen.
3. **Hohe Anfangsinvestitionen:** Die Entwicklung und Implementierung von digitalen Zwillingen erfordert erhebliche Investitionen in Software, Hardware und Fachpersonal. Diese hohen Kosten können insbesondere für kleinere Unternehmen eine Einstiegshürde darstellen, obwohl langfristig Einsparungen und Optimierungen möglich sind.²¹

2.2.3 *Ciros von Festo als Simulationssoftware*

Ciros ist eine leistungsstarke 3D-Simulationssoftware, die speziell für die Automatisierungstechnik entwickelt wurde. Sie dient als universelles Werkzeug zur Erstellung und Anwendung von virtuellen Modellen und ermöglicht sowohl die Simulation komplexer Produktionsprozesse als auch die didaktische Vermittlung von Automatisierungskonzepten in einer virtuellen Umgebung. Ciros kombiniert eine professionelle Simulationsplattform mit zusätzlichen Mechanismen, die es ermöglichen, die Software flexibel in Lern- und Schulungskontexten einzusetzen.

Ein wesentlicher Zweck von Ciros liegt in der Simulation und Optimierung von Produktionsanlagen, einschließlich ihrer Steuerungssysteme. Die Software unterstützt den Import von CAD-Modellen in verschiedenen Formaten (z. B. STEP, STL, Collada) und erlaubt den Aufbau kompletter Produktionslinien basierend auf Modellbibliotheken. Neben den bereits bestehenden Anlagen aus der umfangreichen Bibliothek bietet Ciros jedoch auch die Möglichkeit, vollständig eigene Anlagen und Produktionsprozesse aufzubauen und zu simulieren. Dies erlaubt eine maßgeschneiderte Modellierung, die spezifische Anforderungen und individuelle Szenarien berücksichtigt.

Die umfangreiche Funktionalität von Ciros macht die Software besonders vielseitig:²²

- Sie bietet Zugriff auf über 1100 Roboter- und Anlagenmodelle, die unter anderem mit Steuerungssprachen wie Kuka Robot Language (KRL) oder ABB Rapid programmiert werden können.

²⁰ vgl. Verl, A. et al. (2024), S. 152–153.

²¹ vgl. Verl, A. et al. (2024), S. 318.

²² vgl. o. V.: CIROS 7, Universelles 3D-Simulationssystem | Festo DE.

- Sie ermöglicht die Anbindung an das Manufacturing Execution System (MES4), wodurch digitale Zwillinge für Lernfabriken wie die CP Factory oder das CP-Lab erstellt werden können.
- Mit der Fehlersimulationsfunktion können Szenarien zur Fehlerbehebung simuliert und dokumentiert werden, was besonders für die Schulung und Ausbildung von Vorteil ist.

Ein weiterer Vorteil der Software ist die Integration unterschiedlicher Steuerungssysteme. CiroS ermöglicht die Steuerung der simulierten Anlagen sowohl über integrierte SPS-Steuerungen wie SIMATIC STEP7 als auch über externe Systeme wie MATLAB oder Python, um das Systemverhalten zu erweitern. Die Verbindung von virtuellen Modellen mit physischer Hardware über Hardware-in-the-Loop-Schnittstellen erlaubt zudem realitätsnahe Tests und Optimierungen in einer sicheren Umgebung.

CiroS wird häufig als didaktischer digitaler Zwilling eingesetzt, um Lernende in virtuellen Produktionsumgebungen auszubilden. Dabei können verschiedene Szenarien modelliert und getestet werden, um das Verständnis für Produktionsprozesse, Roboterprogrammierung und Automatisierung zu vertiefen. Die integrierten Schnittstellen wie OPC UA und EtherCAT ermöglichen zudem eine nahtlose Verbindung zwischen der virtuellen und physischen Welt. Die Software bietet somit eine umfassende Plattform für Forschung, Lehre und industrielle Anwendungen. Sie demonstriert praxisnah die Prinzipien der Industrie 4.0 und ist ein leistungsstarkes Werkzeug für die Simulation, Planung und Optimierung moderner Produktionssysteme – sowohl auf Basis bestehender als auch individuell erstellter Anlagen.²³

2.3 Automatisierung und SPS-Programmierung

Die Automatisierungstechnik bildet einen zentralen Bestandteil der modernen industriellen Fertigung und ist eng mit den Prinzipien der Industrie 4.0 verknüpft. Speicherprogrammierbare Steuerungen spielen hierbei eine Schlüsselrolle, da sie die präzise Steuerung und Überwachung von Produktionsprozessen ermöglichen.

Dieses Kapitel gibt zunächst einen Überblick über die Grundlagen der SPS-Programmierung (Kapitel 2.3.1), die die Basis für die Automatisierung bilden. Anschließend wird das TIA Portal von *Siemens* (Kapitel 2.3.2) als zentrale Entwicklungsumgebung für die Programmierung und Simulation von Steuerungssystemen vorgestellt. Schließlich wird die OPC UA-Schnittstelle (Kapitel 2.3.3) erläutert, die als standardisiertes Kommunikationsprotokoll eine reibungslose Vernetzung und Datenübertragung zwischen verschiedenen Komponenten und Systemen gewährleistet.

²³ vgl. o. V.: CIROS 7, Universelles 3D-Simulationssystem | Festo DE.

2.3.1 Grundlagen der SPS-Programmierung

Geschichtliche Entwicklung der Automation

Die Geschichte der Automation reicht bis in die Antike zurück, als Menschen erstmals versuchten, Maschinen für spezifische Aufgaben zu entwickeln. Bereits der griechische Erfinder Heron von Alexandria (ca. 20–62 n. Chr.) beschrieb in seinem Werk „Automata“ innovative Steuerungsmechanismen, wie eine Tür, die sich durch die Wärme eines Feuers öffnet, oder einen Münzweihwasserautomaten, der nach Einwurf einer Münze eine kleine Menge geweihtes Wasser ausgab.

Im Laufe der Jahrhunderte entstanden weitere mechanische Steuerungssysteme, darunter Drehorgeln vor 1600 und Spieldosen im 18. Jahrhundert, die über Lochkarten oder Stiftwalzen gesteuert wurden. Eine bedeutende Weiterentwicklung war der Webstuhl von Joseph-Marie Jacquard, der 1805 mit einem umlaufenden Lochband die Grundlage für automatisierte Steuerungssysteme in der Textilindustrie legte.²⁴

Mit der industriellen Revolution und den Fortschritten in der Elektrotechnik und Elektronik beschleunigte sich die Entwicklung der Steuerungstechnik ab Mitte des 20. Jahrhunderts erheblich. Wesentliche Meilensteine waren:

- **1835:** Entwicklung des Relais durch Joseph Henry, ein grundlegendes Bauelement für elektrische Steuerungen.
- **1941:** Einsatz von Relais in einem frühen Computer von Conrad Zuse.
- **1958:** Einführung der ersten elektronischen Steuerung SIMATIC durch *Siemens*.
- **ab 1970:** Die Entwicklung der speicherprogrammierbaren Steuerungen (SPS), die eine softwarebasierte Programmierung anstelle einer fest verdrahteten Verknüpfungssteuerung erlaubten und damit den Grundstein für die moderne Automatisierungstechnik legten.²⁵

Die Integration von Mikroprozessoren in den 1970er Jahren führte zu einem enormen Fortschritt in der Automatisierung. Dies ermöglichte den Einsatz von Industrierobotern, vollautomatischen Produktionsstraßen und dezentralen Prozessleitsystemen. Die Digitalisierung und die Entstehung des Internets trugen dazu bei, Automatisierungssysteme miteinander zu vernetzen und neue Anwendungsgebiete zu erschließen.

Heute erleben wir durch die Verbreitung künstlicher Intelligenz und fortschrittlicher Technologien wie autonomen Fahrzeugen und intelligenter Verkehrsleitsysteme eine neue Dynamik in der Automatisierungstechnik. Die Steuerungstechnik spielt dabei eine zentrale Rolle, um komplexe Systeme wie urbane Verkehrsinfrastrukturen effizient zu verwalten und die Grundlagen für Industrie 4.0 zu schaffen.²⁶

²⁴ vgl. Tröster, F. (2015), S. 14–15.

²⁵ vgl. Weyrich, M. (2023), S. 24.

²⁶ vgl. Weyrich, M. (2023), S. 26.

Bit und Byte als Grundlage digitaler Informationsverarbeitung

In der digitalen Informationsverarbeitung, welche die Grundlage moderner Automatisierungssysteme bildet, stellen das Bit und das Byte fundamentale Einheiten zur Darstellung und Verarbeitung von Informationen dar. Das Bit (binary digit) ist die elementarste Informationseinheit und kann einen von zwei möglichen Zuständen annehmen. Diese Zustände werden binär als 0 oder 1 repräsentiert und korrespondieren beispielsweise zu den zwei Zuständen eines Schalters (aus/ein) oder zwei unterschiedlichen Spannungspegeln in einer elektronischen Schaltung. Abbildung 1 zeigt anschaulich diese zweiwertige Natur eines Bits.

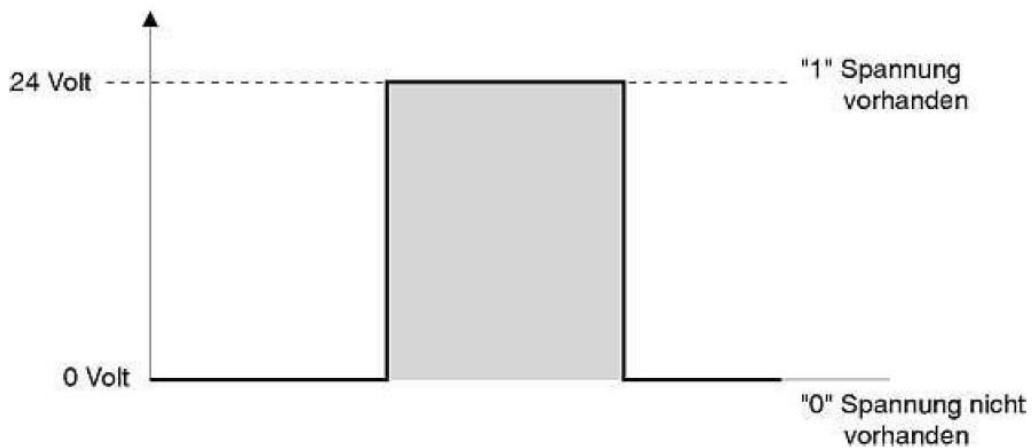


Abbildung 1: mögliche Zustände eines Bits²⁷

Ein Byte ist eine Zusammenfassung von acht einzelnen Bits und dient als gängige Einheit zur Speicherung und Übertragung von Daten. Abbildung 2 visualisiert ein Byte als eine Sequenz von acht Bits, wodurch die strukturelle Beziehung zwischen diesen beiden Einheiten klar wird.

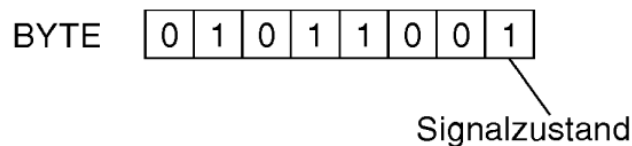


Abbildung 2: Visualisierung eines Bytes²⁸

Durch die Kombination von acht Bits kann ein Byte insgesamt $2^8 = 256$ unterschiedliche Zustände darstellen. Diese Vielfalt an Zuständen ermöglicht die Codierung einer breiten Palette von Informationen, wie beispielsweise numerische Werte, Zeichen oder Steuerungskommandos in digitalen Systemen. Die Konzepte von Bit und Byte bilden somit das Fundament für die digitale Repräsentation von Daten und sind unerlässlich für das Verständnis der Funktionsweise

²⁷ vgl. Kaftan, J. (2015), S. 14.

²⁸ vgl. Kaftan, J. (2015), S. 14.

speicherprogrammierbarer Steuerungen (SPS) und anderer digitaler Systeme in der Automatisierungstechnik.²⁹

Rechnergestützte Automatisierung und der Platz der SPS

Die Entwicklung von SPS ist eng mit der Fortschrittlichkeit der Rechnertechnik verbunden. Im Gegensatz zu analogen Steuerungssystemen, die spezifische Aufgaben schneller ausführen können, bieten rechnergestützte Systeme wie die SPS die Möglichkeit, mehrere Steuerungs- und Regelungsaufgaben gleichzeitig zu bewältigen. Dabei können komplexe Algorithmen umgesetzt werden, die mit traditionellen Steuerungssystemen nicht realisierbar wären.³⁰

Die Hauptmerkmale rechnergestützter Automatisierungsgeräte, zu denen die SPS gehört, sind die Prozesskopplung und der Echtzeitbetrieb. Durch geeignete Ein- und Ausgabemodule wird die SPS direkt mit dem zu steuernden Prozess verbunden. Dies ermöglicht die kontinuierliche Erfassung und Verarbeitung von Prozessdaten sowie die Ausgabe von Steuerbefehlen. Gleichzeitig garantiert ein Echtzeitbetriebssystem, dass alle Aufgaben zeitgerecht erledigt werden.³¹

Hardware-Komponenten einer SPS

Eine SPS besteht aus modular aufgebauten Hardware-Komponenten (*Abbildung 3*), die anwendungsbezogen konfiguriert werden können und umfasst Programmspeicher, einen Zentralprozessor, Merker, Zeit- und Zählglieder sowie Ein- und Ausgabebaugruppen.

²⁹ vgl. Kaftan, J. (2015), S. 14.

³⁰ vgl. Tröster, F. (2015), S. 128.

³¹ vgl. Tröster, F. (2015), S. 129–130.

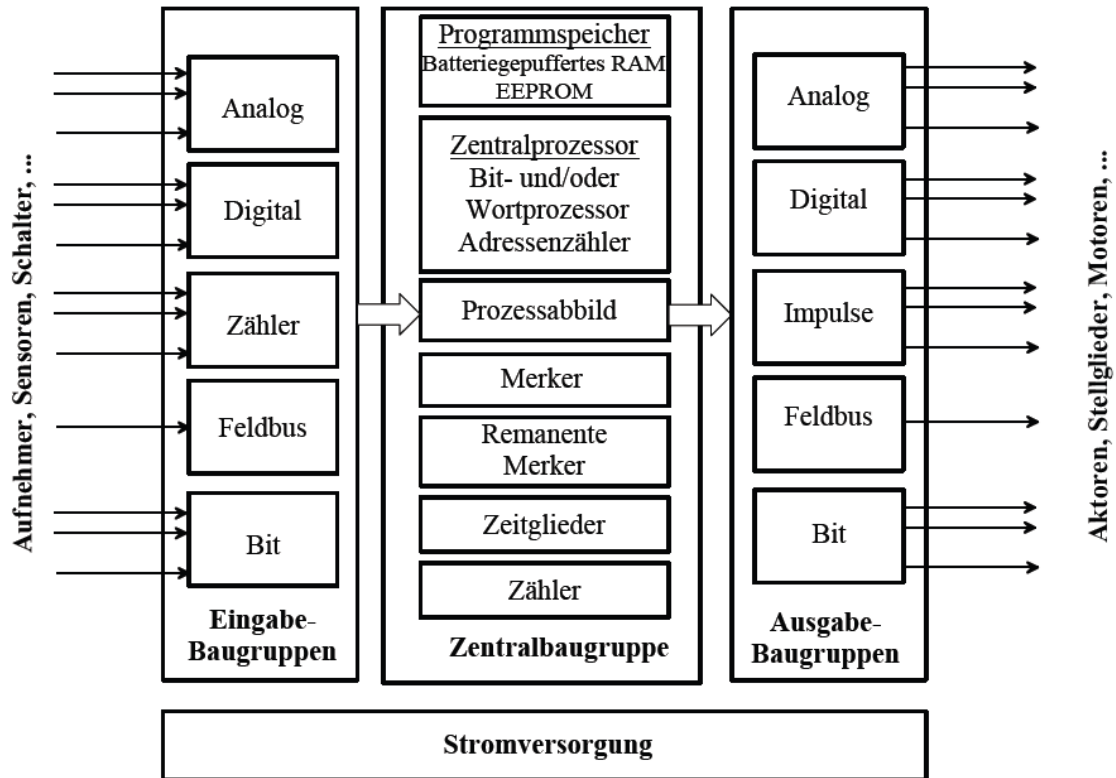


Abbildung 3: SPS-Aufbau³²

Der Programmspeicher dient der Ablage von Steuerungsprogrammen. Typischerweise wird ein EEPROM (Electrically Erasable Programmable Read-Only Memory) genutzt, ergänzt durch batteriegepufferte RAM-Bausteine, die bei Stromausfall die Programmdaten erhalten. Der Zentralprozessor enthält ein Rechen- und Steuerwerk sowie Adressenzähler und ist für die Verarbeitung von Steuerungsanweisungen verantwortlich. Moderne SPS nutzen oft Bit- und Wortprozessoren gleichzeitig, um sowohl binäre Signale als auch komplexere digitale Werte effizient zu verarbeiten.

Das Prozessabbild ist ein Speicherbereich, der die Ein- und Ausgangssignale einer SPS zwischenspeichert. Diese Daten werden während eines Programmzyklus nicht aktualisiert, was die Konsistenz der Prozessinformationen garantiert. Dadurch können die Steuerungsprogramme innerhalb eines Zyklus stets auf dieselben Signaldaten zugreifen.³³

Merker sind Einbit-Speicherplätze, die der Zwischenspeicherung von Signalzuständen oder Berechnungsergebnissen dienen. Sie sind intern und nicht nach außen geführt. Es gibt zwei Arten von Merkern:

Nichtremanente Merker: Verlieren ihren Zustand bei einem Spannungsausfall.

Remanente Merker: Bewahren ihren Zustand auch bei Stromunterbrechungen.

³² vgl. Tröster, F. (2015), S. 134.

³³ vgl. Tröster, F. (2015), S. 134–135.

Zeitglieder und Zähler sind Speicherbereiche, die Zahlenwerte aufnehmen. Diese Werte werden genutzt, um Zeitfunktionen oder Zähloperationen durchzuführen. Zeitglieder kommen etwa bei der Generierung von Verzögerungen zum Einsatz, während Zähler beispielsweise für Stückzählungen verwendet werden.

Die Ein- und Ausgabebaugruppen (E/A-Baugruppen) verbinden die SPS mit Sensoren, Aktoren und weiteren Prozessen. Sie bereiten analoge oder digitale Signale der angeschlossenen Geräte auf und transformieren sie in das Prozessabbild.

Digitale Eingänge arbeiten meist mit einer Spannung von +24 V, die auf den logischen Wert 1 abgebildet wird.

Analoge Eingänge nutzen Spannungen (z. B. ± 10 V) oder Ströme (z. B. 4–20 mA).

Ausgabebaugruppen steuern angeschlossene Aktoren und können analoge oder digitale Signale erzeugen. Die Baugruppen gibt es in Relais- oder Transistorausführungen, um unterschiedliche Anforderungen zu erfüllen.³⁴

Ein Feldbus ist ein Kommunikationssystem, das Sensoren, Aktoren und andere Feldgeräte mit einer SPS verbindet. Alle Geräte sind an einer gemeinsamen Leitung angeschlossen und können individuell adressiert werden, um Datenkonflikte zu vermeiden. Feldbusse erlauben den Aufbau modularer Netzwerke und reduzieren den Verkabelungsaufwand erheblich.

SPS-Systeme können modular aufgebaut werden, was eine anwendungsspezifische Anpassung ermöglicht. Der modulare Aufbau umfasst eine Stromversorgung, eine Zentralbaugruppe, sowie mindestens eine Ein- und Ausgabebaugruppe. Dieser Ansatz erlaubt langfristige Erweiterungen, etwa durch Hinzufügen weiterer Baugruppen für neue Prozessanforderungen.

Für kleinere Steuerungsaufgaben gibt es SPS-Systeme in Kompaktbauweise, die eine kostengünstige Alternative zu modularen Systemen darstellen. Solche Geräte besitzen beispielsweise nur wenige Ein- und Ausgänge und sind vor allem für einfache Steuerungsaufgaben geeignet.

Die Zuverlässigkeit von SPS-Systemen wird durch verschiedene Maßnahmen sichergestellt, wie: Automatischer Wiederanlauf nach Stromausfall, Schutz der gespeicherten Daten und Programme, Trennung zwischen den Potentialen der SPS-Baugruppen und der Prozesshardware.³⁵

Software-Aufbau und Programmierung einer SPS

Die Programmierung einer SPS ist ein zentraler Bestandteil ihrer Nutzung. Sie wird durch spezialisierte Programmiersprachen ermöglicht, die sowohl maschinennahe Befehle als auch grafische Darstellungen umfassen. Diese Sprachen sind normiert (DIN EN 61131-3) und auf die spezifischen Anforderungen der Automatisierungstechnik zugeschnitten.

Nachfolgend werden die wichtigsten SPS-Programmiersprachen ausführlich beschrieben und in Abbildung 4 dargestellt.

³⁴ vgl. Tröster, F. (2015), S. 135–137.

³⁵ vgl. Tröster, F. (2015), S. 137–138.

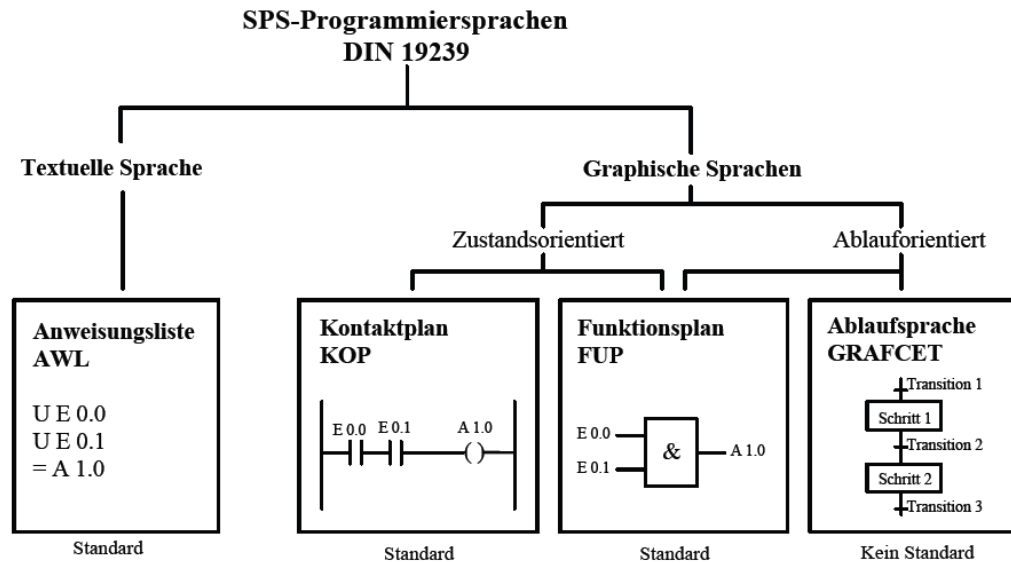


Abbildung 4: SPS-Programmiersprachen³⁶

- **Kontaktplan (KOP):** Der Kontaktplan ist eine grafische Programmiersprache, die auf den Stromlaufplänen der Elektrotechnik basiert. Sie stellt Schalt- und Relaiskontakte dar und eignet sich insbesondere für Anwender ohne tiefgehende Programmiererfahrung. Jedes Netzwerk im KOP wird zeilenweise abgearbeitet, wobei logische Verknüpfungen wie UND, ODER, NOT durch Schalter und Spulen repräsentiert werden.

Ein Beispiel (Abbildung 5) ist die Programmierung eines RS-Flipflops:



Abbildung 5: KOP-Plan RS Flipflops³⁷

Der Ausgang wird durch einen Schalter (Set-Befehl) aktiviert und bleibt aktiv, bis ein anderer Schalter (Reset-Befehl) ihn deaktiviert.

KOP bietet eine intuitive Darstellung und wird daher häufig für die Erstellung einfacher Steuerungsprogramme genutzt. Erweiterungen wie Sprunganweisungen und Flankenerkennung verbessern die Funktionalität.³⁸

- **Anweisungsliste (AWL):** Die Anweisungsliste ist eine vereinheitlichte, textuelle Programmiersprache, die sich besonders für erfahrene Programmierer eignet.

³⁶ vgl. Tröster, F. (2015), S. 145.

³⁷ vgl. Tröster, F. (2015), S. 146.

³⁸ vgl. Tröster, F. (2015), S. 145–146.

Jede Anweisung besteht aus einem Operator (z. B. UND, ODER) und einem Operanden (z. B. eine Speicheradresse). Die Befehle werden zeilenweise ausgeführt.

Beispiel für ein AWL-Programm (RS-Flipflop):

1. L E1.0 – Lade den Wert des Eingangs E1.0.
2. O A1.0 – ODER-Verknüpfung mit dem aktuellen Ausgang A1.0.
3. UN E2.0 – UND NICHT-Verknüpfung mit Eingang E2.0.
4. = A1.0 – Speichere das Ergebnis in A1.0.

Die AWL erlaubt die direkte Kontrolle über die Programmausführung und ist optimal für komplexe Steuerungsaufgaben geeignet, auch wenn sie für Nichtfachleute schwer verständlich sein kann.³⁹

- **Funktionsplan** (FUP): Der Funktionsplan ist eine grafische Sprache, die auf der Verknüpfung von Funktionsblöcken basiert. Eingangs- und Ausgangssignale werden durch Linien miteinander verbunden, und logische Verknüpfungen erfolgen durch standardisierte Symbole (*Abbildung 6*).

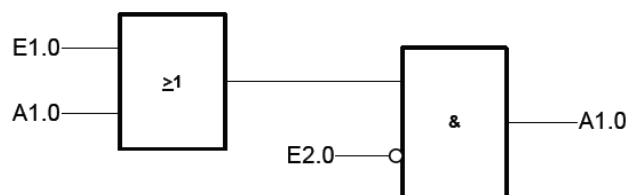


Abbildung 6: Beispiel FUP - Funktionsplan⁴⁰

Funktionen wie UND, ODER, RS-Flipflops und Zähler können direkt eingebunden werden. Erweiterungen erlauben die Nutzung von Sprungbefehlen, um den Funktionsplan modular zu gestalten.

Der FUP eignet sich hervorragend für die Darstellung und Kommunikation komplexer Steuerungslogiken, insbesondere bei größeren Teams aus verschiedenen Fachbereichen.⁴¹

- **Strukturierter Text** (ST): Der Strukturierte Text ist eine Hochsprache, die in ihrer Syntax an Pascal erinnert. Sie erlaubt die Implementierung komplexer Steuerungslogiken mit wenig Code und ist besonders leistungsstark für mathematische Operationen und Schleifen.

³⁹ vgl. Tröster, F. (2015), S. 147–148.

⁴⁰ vgl. Tröster, F. (2015), S. 150.

⁴¹ vgl. Tröster, F. (2015), S. 150–151.

Beispiel einer ST-Anweisung (*Abbildung 7*):

```
IF Schalter_1 AND NOT Schalter_2 THEN
    Licht := TRUE;
ELSE
    Licht := FALSE;
END_IF;
```

Abbildung 7: Beispiel ST- Strukturierter Text⁴²

ST bietet Flexibilität und Effizienz bei der Programmierung und wird bevorzugt von Entwicklern mit Erfahrung in Hochsprachen genutzt. Die Norm EN 61131-3 definiert eine Vielzahl von Operatoren und Kontrollstrukturen wie IF, CASE oder FOR, die ST besonders für komplexe Steuerungsaufgaben attraktiv machen.⁴³

- **Ablaufsprache (AS):** Die Ablaufsprache basiert auf GRAFCET und dient zur Strukturierung und Beschreibung von Ablaufsteuerungen. Sie ist eine der wenigen SPS-Sprachen, die sowohl textuelle als auch grafische Darstellungen unterstützt.
 - **Schritte:** Jeder Schritt ist mit Aktionen verknüpft, die ausgeführt werden, wenn der Schritt aktiv ist.
 - **Transitionen:** Sie legen die Bedingungen für den Übergang von einem Schritt zum nächsten fest.
 - **Verzweigungen:** Alternative oder parallele Abläufe können einfach modelliert werden.

Die *Abbildung 8* zeigt eine grafische Darstellung eines Grafkets.

⁴² Eigene Darstellung

⁴³ vgl. Tröster, F. (2015), S. 164–165.

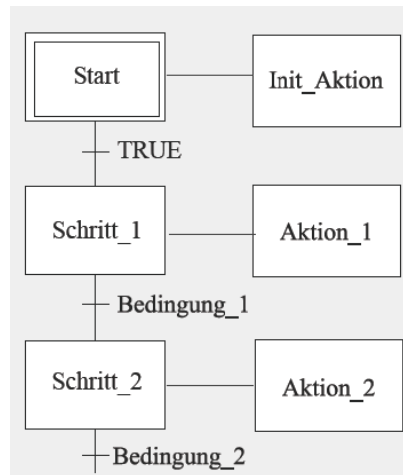


Abbildung 8: Beispiel AS - Ablaufsprache⁴⁴

Die AS ist ideal für die Implementierung von Ablaufsteuerungen mit klaren Zustandsübergängen und wird häufig in der Prozesstechnik eingesetzt.

Echtzeitbetrieb und Zykluszeit

Ein wesentliches Merkmal der SPS ist der Echtzeitbetrieb. Während eines festen Zeitzyklus werden Eingabewerte gelesen, das Steuerungsprogramm abgearbeitet und die Ausgabe aktualisiert. Der Echtzeitbetrieb gewährleistet konsistente Prozessinformationen und ermöglicht eine zuverlässige Steuerung.⁴⁵

Diese Sprachen bieten eine breite Palette von Möglichkeiten, die Steuerungslogik von SPS individuell und bedarfsgerecht zu gestalten. Die Wahl der Sprache hängt von der Art der Anwendung und den Vorkenntnissen der Anwender ab. In den folgenden Kapiteln wird die praktische Umsetzung dieser Programmiersprachen im Kontext des TIA-Portals von *Siemens* näher betrachtet.

2.3.2 Tia Portal von Siemens

Das Totally Integrated Automation (TIA) Portal von *Siemens* ist ein umfassendes Engineering-Framework, das alle Aspekte der Automatisierungstechnik integriert. Es bietet eine einheitliche Umgebung für die Programmierung, Visualisierung und Steuerung von Automatisierungskomponenten wie SPS, HMI und Antrieben. Das TIA Portal ist besonders benutzerfreundlich und ermöglicht eine effiziente Erstellung und Verwaltung von Automatisierungsprojekten.⁴⁶

⁴⁴ vgl. Tröster, F. (2015), S. 167.

⁴⁵ vgl. Tröster, F. (2015), S. 139–140.

⁴⁶ vgl. Siemens AG (2019), S. 9–12.

Aufbau des TIA-Portals

Das TIA Portal ist modular aufgebaut und bietet verschiedene Editoren, die eine intuitive Navigation und Programmierung ermöglichen. Die Projektnavigation erleichtert den schnellen Zugriff auf Programme, Variablen und Bausteine. Insbesondere der FUP-Editor (Funktionsplan-Editor) ist ein zentrales Werkzeug, das die grafische Programmierung durch Drag-and-Drop von Bausteinen unterstützt.⁴⁷

Grundlagen der Programmierung im TIA Portal

Die Programmierung im TIA Portal zeichnet sich durch einen hierarchischen Aufbau von Bausteinen aus, die unterschiedliche Aufgaben und Funktionen übernehmen. Dabei wird eine klare Struktur verwendet, um die Programmierung effizient und modular zu gestalten. Die wichtigsten Bausteinarten sind:

Organisationsbausteine (OB)

Organisationsbausteine bilden den Einstiegspunkt für die Programmausführung in der Steuerung. Sie definieren, wie und wann bestimmte Programme ausgeführt werden sollen. Es gibt verschiedene Arten von OBs:

- **Zyklische OBs:** Diese Bausteine werden in regelmäßigen Abständen ausgeführt und sind für den kontinuierlichen Ablauf von Programmen zuständig (z. B. OB1).
- **Ereignisgesteuerte OBs:** Sie reagieren auf bestimmte Ereignisse, wie Alarmer oder Unterbrechungen, und ermöglichen so eine schnelle Reaktion auf Prozessänderungen (z. B. OB40 für Hardware-Interrupts).
- **Startup-OBs:** Diese Bausteine kommen zum Einsatz, wenn die Steuerung gestartet wird, und initialisieren Variablen oder Zustände (z. B. OB100). Organisationsbausteine arbeiten eng mit anderen Bausteinen zusammen, um die gewünschte Funktionalität zu realisieren.⁴⁸

Funktionsbausteine (FB)

Funktionsbausteine sind modular aufgebaute Programmbestandteile, die sowohl logische Operationen als auch Datenverarbeitung enthalten. Der entscheidende Vorteil eines FBs ist, dass er zustandsbehaftet arbeitet. Das bedeutet, dass Zustandsinformationen zwischen den Zyklen erhalten bleiben. Die Verwendung von Instanz-Datenbausteinen (IDBs) ermöglicht es, dass der FB für verschiedene Instanzen unterschiedliche Werte speichern kann. FBs eignen sich besonders für wiederkehrende Steuerungsaufgaben wie Regelungen oder komplexe Prozesslogik.⁴⁹

⁴⁷ vgl. Siemens AG (2019), S. 12–13.

⁴⁸ vgl. Siemens AG (2019), S. 45–47.

⁴⁹ vgl. Siemens AG (2019), S. 48–51.

Funktionen (FC)

Funktionen sind zustandslose Bausteine, die für Berechnungen oder einfache logische Operationen verwendet werden. Sie haben den Vorteil, dass sie wiederverwendbar und unabhängig von Instanz Daten arbeiten. Funktionen können Parameter übergeben werden, die nur für die Dauer des Funktionsaufrufs gültig sind. Beispiele sind mathematische Berechnungen, wie das Addieren von Werten, oder einfache Signalverarbeitungen. FCs sind besonders geeignet, um häufig benötigte Berechnungen zentralisiert bereitzustellen.

Datenbausteine (DB)

Datenbausteine speichern Variablen und Werte, die entweder von der SPS-Logik verwendet oder zwischen Bausteinen ausgetauscht werden. Es gibt zwei Arten von Datenbausteinen:

- **Instanz-Datenbausteine** (IDBs): Diese sind an Funktionsbausteine gebunden und speichern die individuellen Zustände für jede Instanz des FBs.
- **Globale Datenbausteine** (GDBs): Sie enthalten allgemeine Variablen, die von mehreren Bausteinen genutzt werden können. Datenbausteine unterstützen verschiedene Datentypen wie BOOL, INT, REAL und ARRAY, was ihre Vielseitigkeit erhöht. Eine Besonderheit ist die Möglichkeit, strukturierte Datentypen zu erstellen, die komplexe Datenstrukturen abbilden können.⁵⁰

Zyklische Programmausführung und Priorisierung

Das TIA Portal erlaubt die Organisation der Programmausführung in Tasks, die den OBs zugeordnet sind. Diese Tasks können priorisiert werden, um sicherzustellen, dass kritische Programme mit höherer Priorität verarbeitet werden. Zusätzlich gewährleistet der zyklische Ablauf eine konsistente Verarbeitung der Prozessdaten.⁵¹

Durch diese klar definierte Struktur und die differenzierte Nutzung von Bausteinen wird die Entwicklung komplexer Steuerungsprogramme im TIA Portal übersichtlich und wartungsfreundlich gestaltet.

Datentypen und symbolische Adressierung

Im TIA Portal werden verschiedene elementare Datentypen unterstützt, darunter BOOL, INT, REAL, und WSTRING, die auch Unicode-Zeichen umfassen können. Statt absoluter Adressen wird die Nutzung von symbolischer Adressierung empfohlen, da sie Wartung und Fehlersuche erheblich erleichtert. Die symbolische Adressierung ermöglicht zudem die Nutzung von ARRAYs zur Verarbeitung mehrerer Datenpunkte.⁵²

⁵⁰ vgl. Siemens AG (2019), S. 51–54.

⁵¹ vgl. Siemens AG (2019), S. 47–49.

⁵² vgl. Siemens AG (2019), S. 68–72.

Programmiersprache FUP

Die Programmiersprache FUP (Funktionsplan) ist speziell für Anwender ohne tiefgehende Programmierfahrung konzipiert. Sie bietet eine visuelle Darstellung von logischen Operationen wie UND, ODER und NICHT. Dies macht FUP zu einer intuitiven und leicht verständlichen Programmiersprache, die dennoch leistungsfähig genug ist, um komplexe Automatisierungsaufgaben zu lösen.⁵³

Optimierte Bausteine

Das TIA Portal unterstützt die Verwendung von optimierten Bausteinen, die zahlreiche Vorteile bieten:

- Automatische Sortierung der Variablen für schnelleren Zugriff.
- Symbolische Adressierung reduziert Fehler durch falsche absolute Adressen.
- Speicheroptimierung durch Minimierung von Datenlücken.⁵⁴

Variablen und Schnittstellen

Variablen spielen eine zentrale Rolle in der Programmierung mit dem TIA Portal. Sie werden in verschiedenen Typen unterschieden:

- **PLC-Variablen:** Für die Nutzung innerhalb von Programmen.
- **Aktualparameter:** Für die Datenübergabe bei Bausteinaufrufen.
- **Formalparameter:** Zur Definition von Bausteinschnittstellen (Input, Output, InOut). Ein besonderes Merkmal des TIA-Portals ist die Möglichkeit, Variablen als remanent zu definieren, um ihre Werte auch nach einem Stromausfall zu erhalten.⁵⁵

Funktionen und Bibliotheken

Das TIA Portal ermöglicht die Wiederverwendung von Elementen wie Bausteinen oder Datentypen über Bibliotheken. Diese können gespeichert, versioniert und in anderen Projekten genutzt werden, was die Effizienz und Konsistenz der Programmierung deutlich erhöht.⁵⁶

Fazit

Das TIA Portal von *Siemens* bietet eine moderne, integrierte Umgebung, die eine einfache und effiziente Automatisierungsprogrammierung ermöglicht. Durch die Nutzung symbolischer Adressierung, optimierter Bausteine und der intuitiven FUP-Programmiersprache können auch komplexe Steuerungsaufgaben strukturiert und fehlerfrei umgesetzt werden. Die klare Strukturierung durch OBs, FBs und FCs sowie die umfassende Unterstützung für Datentypen und Variablen machen das TIA Portal zu einem unverzichtbaren Werkzeug in der Automatisierungstechnik.

⁵³ vgl. Siemens AG (2019), S. 12–13.

⁵⁴ vgl. Siemens AG (2019), S. 14–21.

⁵⁵ vgl. Siemens AG (2019), S. 10–11.

⁵⁶ vgl. Siemens AG (2019), S. 79–82.

2.3.3 OPC UA - Schnittstelle für industrielle Kommunikation

OPC UA ist ein plattformunabhängiges Kommunikationsprotokoll, das speziell für die Anforderungen von Industrie 4.0 und IoT entwickelt wurde.

Es ermöglicht den standardisierten Datenaustausch zwischen Maschinen, Geräten und Diensten in unterschiedlichen Industriebranchen. OPC UA unterstützt sowohl horizontale (Maschine-zu-Maschine) als auch vertikale Kommunikation (von der Sensor- bis zur Managementebene, siehe *Abbildung 9*) und ist flexibel skalierbar, wodurch es für lokale und globale Anwendungen geeignet ist.

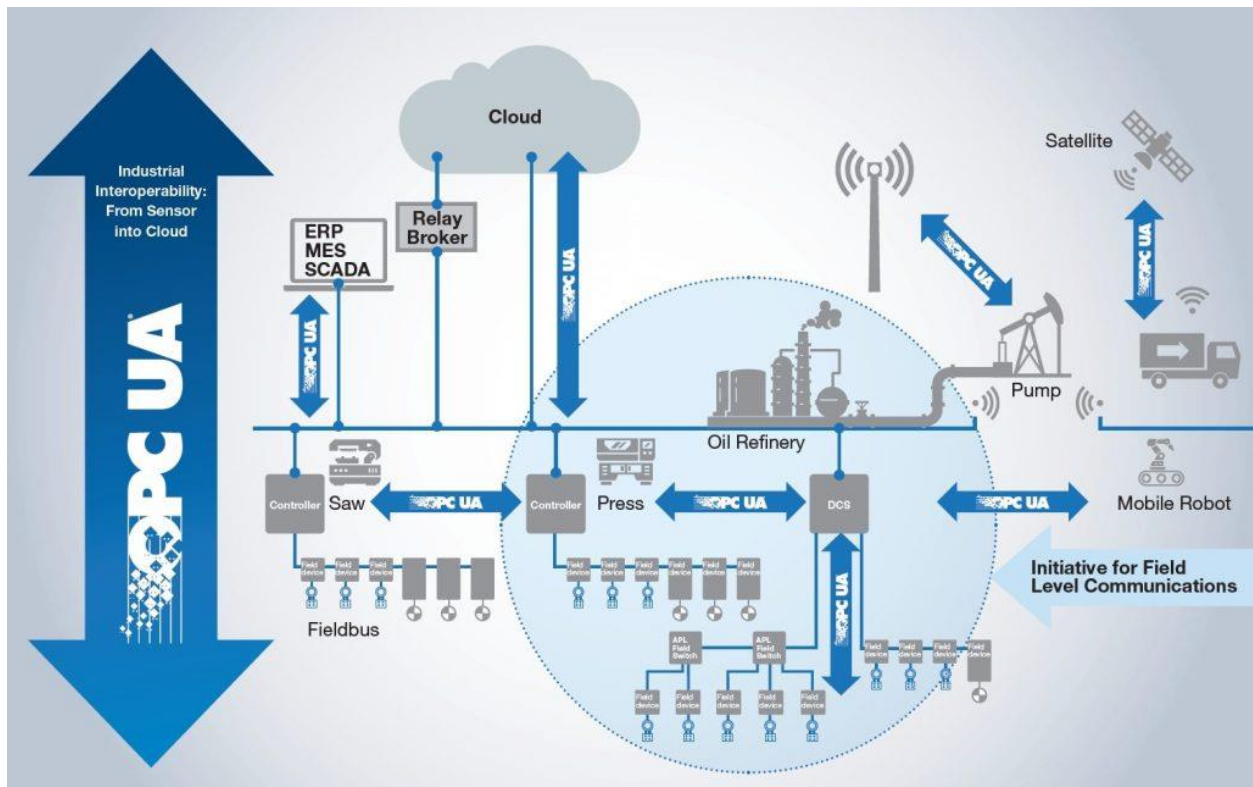


Abbildung 9: OPC UA vertikale und horizontale Kommunikation⁵⁷

Die Entwicklung von OPC UA begann in den frühen 2000er-Jahren unter der Leitung der OPC Foundation, um die Einschränkungen früherer Standards wie DCOM (Distributed Component Object Model) zu überwinden. Der Standard wurde erstmals 2006 veröffentlicht und 2009 überarbeitet. OPC UA adressierte Sicherheitsprobleme und Betriebssystemabhängigkeiten der Vorgängermodelle und wurde zur Norm IEC 62541.

Seit 2010 hat sich OPC UA als internationaler Standard für den industriellen Datenaustausch etabliert und bildet die Grundlage für zahlreiche Anwendungen in Industrie 4.0 und IoT.

OPC UA ist entscheidend für die Umsetzung der Vision von Industrie 4.0, da es eine einheitliche Sprache für den Datenaustausch bietet und die Integration verschiedener Systeme und Hersteller ermöglicht. Die wichtigsten Eigenschaften sind:

- Unterstützung von Echtzeitkommunikation und Cloud-Integration.
- Möglichkeit der sicheren und effizienten Datenübertragung.
- Unabhängigkeit von Betriebssystemen und Hardwareplattformen.

⁵⁷ vgl. o. V.: OPC UA auf der Feldebene - Industrial Communication Journal (2021).

- Einbindung sowohl moderner als auch älterer Systeme. Durch diese Merkmale schließt OPC UA die Lücke zwischen der Automatisierungsebene und der IT-Welt und ermöglicht eine nahtlose Kommunikation entlang der gesamten Automatisierungspyramide.⁵⁸

Die Architektur von OPC UA folgt dem Prinzip eines Client-Server-Modells. Dabei übernimmt der OPC UA-Server die Aufgabe, standardisierte Daten aus der Steuerungsebene – beispielsweise aus einer speicherprogrammierbaren Steuerung (SPS) – bereitzustellen. Der OPC UA-Client hingegen dient dazu, diese Daten abzufragen und in übergeordneten Anwendungen wie SCADA-, MES- oder ERP-Systemen weiterzuverarbeiten. Zudem ist es möglich, dass der Client Informationen an den Server zurücksendet, wodurch eine bidirektionale Kommunikation ermöglicht wird.

Ein wesentlicher Vorteil von OPC UA ist seine Fähigkeit zur horizontalen und vertikalen Kommunikation. Der Standard ermöglicht somit einen durchgängigen Datenaustausch von der Feldebene mit Sensoren und Aktoren bis hin zu betriebswirtschaftlichen Managementsystemen. Zur Gewährleistung von Sicherheit und Integrität der Datenübertragung verwendet OPC UA moderne Sicherheitsprotokolle wie SSL/TLS. Darüber hinaus unterstützt der Standard sowohl binäre Datenformate für eine leistungsstarke Kommunikation als auch XML-basierte Formate für eine höhere Flexibilität in heterogenen Systemumgebungen.

Mit seiner Offenheit, Skalierbarkeit und Interoperabilität ist OPC UA ein zentraler Baustein für die Digitalisierung in der Industrie. Es ermöglicht nicht nur die Vernetzung unterschiedlicher Systeme, sondern schafft auch die Grundlage für die Integration moderner Technologien wie Cloud Computing und IoT/IIoT, wodurch es zu einem unverzichtbaren Werkzeug für die Zukunft der industriellen Automatisierung wird.⁵⁹

⁵⁸ vgl. Babel, W. (2024), S. 470–473.

⁵⁹ vgl. Babel, W. (2024), S. 475–480.

2.4 Optische Messtechnik und Qualitätssicherung

Die Messtechnik spielt eine Schlüsselrolle in der modernen Industrie, insbesondere in der Qualitätssicherung. Mit dem Ziel, Bauteile und Prozesse präzise zu überwachen und zu optimieren, hat sich die Messtechnik über Jahrzehnte hinweg weiterentwickelt.

Optische Technologien, wie der Laser, haben seit ihrer Entdeckung bahnbrechende Fortschritte ermöglicht. Theodore Harold Maiman realisierte 1960 den ersten Laser auf der Grundlage von Einsteins Konzept der stimulierten Emission. Seitdem hat der Laser in der Messtechnik eine Vielzahl von Anwendungen gefunden, darunter holografische Verfahren, digitale Bildverarbeitung und interferometrische Messtechniken. Diese Technologien erlauben die präzise Erfassung von Messdaten und deren Verarbeitung in Echtzeit, was für die automatisierte Qualitätssicherung unerlässlich ist.⁶⁰

In vernetzten Produktionssystemen ist die Qualitätssicherung entscheidend, um die Effizienz und Zuverlässigkeit der Produktionsprozesse zu gewährleisten. Messtechnische Verfahren ermöglichen nicht nur die Erfassung und Analyse von Produktionsdaten, sondern auch deren Integration in industrielle Kommunikations- und Steuerungssysteme. Dadurch können Abweichungen frühzeitig erkannt und korrigiert werden, was sowohl die Produktionsqualität als auch die Wirtschaftlichkeit steigert.⁶¹

2.4.1 Grundlagen der automatisierten Messtechnik

Die automatisierte Messtechnik ist ein integraler Bestandteil moderner Fertigungsprozesse, insbesondere bei der Qualitätssicherung. Eine zentrale Rolle spielt dabei die automatisierte Sichtprüfung, die in den letzten Jahren zunehmend an Bedeutung gewonnen hat. Diese Technik dient der 100%-Prüfung von Bauteilen, um den steigenden Qualitätsanforderungen gerecht zu werden. Während früher vorwiegend Stichprobenprüfungen von Menschen durchgeführt wurden, ermöglicht die Automatisierung eine objektivere, reproduzierbare und wirtschaftlich realisierbare Prüfung.

Bildverarbeitung in der automatisierten Sichtprüfung

Ein wesentliches Element der automatisierten Messtechnik ist die Bildverarbeitung. Sie umfasst mehrere Schritte, die darauf abzielen, den Informationsgehalt von aufgenommenen Bilddaten systematisch zu reduzieren und schließlich eine binäre Entscheidung (Gut oder Schlecht) zu ermöglichen. Der Prozess beginnt mit der Bildakquisition, bei der kontrastreiche Bilder aufgenommen werden, um relevante Merkmale hervorzuheben und Störungen zu minimieren. Die Qualität dieser Aufnahmen hat einen direkten Einfluss auf die Effizienz der nachfolgenden Schritte.⁶²

Die Bildverarbeitung setzt mit der Vorverarbeitung fort, die Filtertechniken einsetzt, um Bildrauschen zu glätten, Kanten hervorzuheben und inhomogene Bereiche zu korrigieren. Diese Schritte bereiten die Daten für die Segmentierung vor, bei der Bildbereiche zu zusammenhängenden Objekten verknüpft werden. Häufig wird die Schwellwertbildung angewandt, bei der Bildpunkte aufgrund ihrer Grauwertintensitäten klassifiziert werden.

⁶⁰ vgl. Keferstein, C.P. / Marxer, M. (2015), S. 195–196.

⁶¹ vgl. Schuth, M. / Buerakov, W. (2017), S. XXXVII–XXXVIII.

⁶² vgl. Marxer, M. et al. (2021), S. 194–195.

Alternativ können Kanten- und Konturanalysen oder Bewegungsanalysen eingesetzt werden, insbesondere bei Bildsequenzen.

Merkmalsberechnung und Klassifizierung

Nach der Segmentierung erfolgt die Berechnung von Parametern wie Länge, Breite, Fläche oder Kontur der erkannten Objekte. Diese Merkmale sind entscheidend für die anschließende Klassifizierung. Hierbei werden die Objekte in definierte Klassen eingeteilt, die bestimmten Fehlertypen entsprechen. Die Klassifizierung erfolgt entweder durch Entscheidungsbäume, statistische Verfahren oder moderne Ansätze wie maschinelles Lernen und neuronale Netze. Dieser Prozess erfordert ein tiefes Verständnis des Werkstücks, des Herstellungsprozesses und der Fehlerentstehung.⁶³

Toleranzbewertung und Systembewertung

Abschließend erfolgt die Toleranzbewertung, bei der festgelegt wird, wie viele erkannte Fehler zur Klassifizierung eines Werkstücks als „Schlechtteil“ führen. Diese Fehlergrenzen beeinflussen direkt die Erkennungsleistung des Systems. Die Qualität der automatisierten Sichtprüfung kann durch Kennzahlen wie Sensitivität, Spezifität und Vorhersagewerte beurteilt werden, die aus einer Konfusionsmatrix abgeleitet werden. Wenn die Ergebnisse unbefriedigend sind, sollte der gesamte Verarbeitungsprozess, einschließlich Bildakquisition, Merkmalsbestimmung und Klassifizierungsverfahren, überprüft werden.

Mit der kontinuierlichen Weiterentwicklung der Bildverarbeitungstechnologien und der Integration datengetriebener Methoden wie maschinellem Lernen wird die automatisierte Messtechnik in Zukunft noch präzisere und effizientere Prüfprozesse ermöglichen.⁶⁴

2.4.2 Bedeutung der Qualitätssicherung in vernetzten Produktionssystemen

Im Zeitalter vernetzter Produktionssysteme, die durch die Prinzipien von Industrie 4.0 und dem Internet der Dinge geprägt sind, gewinnt die Qualitätssicherung eine fundamental neue Bedeutung. Die Fähigkeit, Effizienz, Zuverlässigkeit und Flexibilität in komplexen Wertschöpfungsnetzwerken zu gewährleisten, hängt maßgeblich von einer intelligenten und integrierten Qualitätssicherung ab.⁶⁵

Die traditionelle, isolierte Betrachtung von Produktionsschritten weicht in vernetzten Systemen einer ganzheitlichen Perspektive, in der Daten und Informationen entlang der gesamten Wertschöpfungskette ausgetauscht und genutzt werden. In diesem Kontext ermöglicht eine effektive Qualitätssicherung nicht nur die Erfassung und Analyse von Produktionsdaten in Echtzeit, sondern auch deren nahtlose Integration in industrielle Kommunikations- und Steuerungssysteme.

Moderne Messtechnik, insbesondere der verstärkte Einsatz optischer Technologien wie Laserverfahren und fortschrittlicher Sensorik, spielt hierbei eine Schlüsselrolle. Diese Technologien erlauben eine präzise, berührungslose und oft flächenhafte Erfassung von

⁶³ vgl. Keferstein, C.P. / Marxer, M. (2015), S. 163.

⁶⁴ vgl. Marxer, M. et al. (2021), S. 196–198.

⁶⁵ vgl. Marxer, M. et al. (2021), S. 6–7.

Qualitätsmerkmalen, die weit über die Möglichkeiten konventioneller Methoden hinausgeht. Die Integration von Multisensor-Koordinatenmesstechnik ermöglicht zudem eine umfassende Qualitätskontrolle unterschiedlicher Werkstückmerkmale in einer einzigen Aufspannung.

Ein zentraler Aspekt der Qualitätssicherung in vernetzten Produktionssystemen ist die frühzeitige Erkennung und Korrektur von Abweichungen. Durch die kontinuierliche Überwachung von Prozessparametern und Qualitätsmerkmalen, unterstützt durch statistische Prozessregelung und datengetriebene Analysen, können Fehlerursachen identifiziert und vorbeugende Maßnahmen eingeleitet werden. Dies führt nicht nur zu einer Steigerung der Produktionsqualität, sondern optimiert auch die Wirtschaftlichkeit durch die Reduzierung von Ausschuss und Nacharbeit.⁶⁶

Darüber hinaus ermöglicht die Vernetzung von Produktionssystemen die Implementierung von Konzepten wie dem Digitalen Zwilling. Diese virtuelle Repräsentation physischer Gegenstände erlaubt eine kontinuierliche Überwachung, Simulation und Optimierung von Produktionsprozessen und Produktqualität in Echtzeit. Durch die Verknüpfung von realen Betriebsdaten mit virtuellen Modellen können vorausschauende Instandhaltung (Predictive Maintenance) und Steuerungssysteme realisiert werden, die die Qualität und Zuverlässigkeit der Produktion weiter erhöhen.

Die Bedeutung der Qualitätssicherung in vernetzten Produktionssystemen ergibt sich somit nicht nur in der reinen Produktkontrolle, sondern vielmehr in der aktiven Gestaltung effizienter Produktionsprozesse, die durch den intelligenten Einsatz von Messtechnik und Informationstechnologie kontinuierlich optimiert werden.⁶⁷

⁶⁶ vgl. Keferstein, C.P. / Marxer, M. (2015), S. 63–64.

⁶⁷ vgl. Keferstein, C.P. / Marxer, M. (2015), S. 67.

3 Planung und Konzept der Vermessungsstation

Die Planung und das Konzept der Vermessungsstation stellen eine entscheidende Grundlage für den Erfolg dieser Arbeit dar, da sowohl die virtuelle als auch die physische Umsetzung auf dieser konzeptionellen Basis aufbauen. In diesem Kapitel werden daher die Anforderungen und der daraus folgende strukturelle Entwurf der Station detailliert beschrieben, um sicherzustellen, dass die Messstation den Ansprüchen der Modellfabrik entspricht.

Ein besonderer Fokus liegt auf der Integration der Vermessungsstation in das bestehende Produktionskonzept einer „smarten Produktion“ der HAW Hamburg, das von Herrn Prof. Dr.-Ing. A. Koch im Sinne der Industrie 4.0 erdacht und entwickelt wurde. Die Station muss daher nicht nur die spezifischen Anforderungen einer industriellen Messstation erfüllen, sondern auch in der Lage sein, nahtlos in die vernetzte, automatisierte Produktionsumgebung eingebunden zu werden.

Dies erfordert eine Planung, die Aspekte wie die Zuverlässigkeit, Präzision und Anpassungsfähigkeit an eine laufende Produktion berücksichtigt.

Mit der Entwicklung eines umfassenden Konzepts wird sichergestellt, dass die Vermessungsstation die Anforderungen der Industrie 4.0 erfüllt und langfristig in einer automatisierten Produktionsumgebung funktionsfähig bleibt.

3.1 Anforderungen an die Vermessungsstation

Um die Vermessungsstation sowohl virtuell als digitalen Zwilling als auch physisch aufzubauen, müssen grundlegende Anforderungen definiert werden, die eine funktionsfähige und zuverlässige Implementierung sicherstellen. Diese Anforderungen dienen als Referenzpunkte für die spätere Überprüfung und Evaluierung der Arbeitsergebnisse.

Da der digitale Zwilling und die physische Station dieselben Komponenten und die SPS-Steuerung verwenden sollen, jedoch unterschiedliche Schwerpunkte in ihrer Funktionalität aufweisen, werden die Anforderungen in die spezifischen Anforderungen an den digitalen Zwilling und die physische Station unterteilt.

3.1.1 Anforderungen an den digitalen Zwilling

Der digitale Zwilling der Vermessungsstation soll die Bewegung der mechanischen Komponenten und des zu vermessenden Bauteils präzise simulieren und die SPS-Steuerung in einem virtuellen Umfeld testen. Die Anforderungen konzentrieren sich daher auf die Implementierung und Validierung der Steuerungslogik sowie der Bewegungsabläufe im virtuellen Raum. Die wesentlichen Anforderungen sind:

1. **Simulation der Bewegungsabläufe:** Die Simulation muss alle relevanten Bewegungen der mechanischen Komponenten präzise nachbilden. Dazu gehören Rotationen, Translationen und die Positionierung des Bauteils in der Messstation. Die Bewegungen

sollten in Echtzeit oder in einem simulierbaren zeitlichen Rahmen ablaufen, um die Steuerungslogik und mögliche Bewegungsabläufe umfassend zu testen.

2. **Implementierung der SPS-Steuerung:** Der digitale Zwilling muss die Steuerungslogik abbilden, die im TIA-Portal entwickelt und später auf die physische SPS übertragen wird. Hierbei liegt der Hauptfokus auf der Verifizierung der Steuerungsprogramme und Abläufe, die durch die SPS gesteuert werden. Alle Bewegungssequenzen und Steuerungsvorgänge, die für den Betrieb der Messstation erforderlich sind, müssen digital abgebildet und getestet werden können.
3. **Vorgabe der Abmessungen der Station:** Der digitale Zwilling muss die exakten Abmessungen der Station berücksichtigen, da die Höhen- und Breitenvorgaben der Station vorgegeben sind. Die Station soll an die Höhe der bestehenden Produktionsumgebung angepasst werden, sodass die Arbeitsfläche etwa 780 mm über dem Boden liegt, um eine nahtlose Übergabe der Bauteile von und zu den angrenzenden Förderbändern zu gewährleisten. Die Breite der Station wird durch die 700-mm-lange *Festo*-Komponente, das Förderband der Station, bestimmt und muss diese Dimensionen exakt nachbilden.
4. **Überprüfung der räumlichen Anordnung:** Da der Platz auf der Arbeitsfläche durch die vorgegebenen Abmessungen beschränkt ist, muss die räumliche Anordnung aller Komponenten innerhalb der Station optimiert werden. Der digitale Zwilling soll alle relevanten Komponenten so anordnen, dass ihre Funktionalität gewährleistet ist, und eine effiziente Nutzung der zur Verfügung stehenden Fläche sicherstellen.
5. **Test der Sicherheitsfunktionen:** Da die Vermessungsstation in eine laufende, automatisierte Produktionsumgebung integriert wird, muss der digitale Zwilling auch alle relevanten Sicherheitsfunktionen beinhalten. Hierzu zählen etwa das automatische Anhalten der Station bei unzulässigen Bewegungen oder die Verhinderung von Kollisionen. Diese Sicherheitsfunktionen sollen innerhalb des digitalen Zwillings überprüft und optimiert werden.
6. **Anpassungsfähigkeit der Steuerung:** Der digitale Zwilling muss auch die Anpassungsfähigkeit der SPS-Steuerung an verschiedene Produktionsszenarien ermöglichen. Das System soll flexibel genug sein, um potenzielle Änderungen in den Produktionsabläufen zu berücksichtigen und so angepasst werden können, dass die Steuerung auch in einem variablen Produktionsumfeld funktioniert.

3.1.2 Anforderungen an die physische Vermessungsstation

Die physische Vermessungsstation muss dieselben grundlegenden Funktionen und Steuerungsabläufe wie der digitale Zwilling umsetzen und darüber hinaus die Anforderungen an Stabilität, Zuverlässigkeit und Integration in eine smarte Produktionsumgebung erfüllen. Die wesentlichen Anforderungen an die physische Station sind:

- 1. Mechanische Stabilität, Präzision und Dämpfung:** Die mechanischen Komponenten der Station müssen stabil und präzise montiert sein, um die exakte Positionierung des Bauteils sicherzustellen.
Alle mechanischen Bewegungen müssen mit hoher Wiederholgenauigkeit und ohne Abweichungen ausgeführt werden können, um eine zuverlässige Steuerung zu gewährleisten.
Das Gestell ist zudem so zu konstruieren, dass es die dynamischen Kräfte und Schwingungen, die durch die Bewegungen der mechanischen Komponenten entstehen, aufnehmen und absorbieren kann.
- 2. Funktionalität der SPS-Steuerung:** Die auf dem digitalen Zwilling getestete SPS-Logik muss auf die physische Station übertragbar und anpassbar sein. Dazu gehört die Steuerung aller mechanischen Abläufe sowie die Einbindung des Messprogramms in die Automatisierungslogik. Jede Bewegung und Steuerungsfunktion, die im digitalen Zwilling getestet wurde, soll in der physischen Umgebung verifiziert und stabil ausgeführt werden.
- 3. Integration des Messprogramms:** Die Station muss das Messprogramm in die SPS-Steuerung integrieren und sicherstellen, dass dieses durch die SPS gesteuert und bei Bedarf gestartet oder gestoppt werden kann. Die reibungslose Übergabe an das Messprogramm ist eine zentrale Anforderung, die geprüft und für den praktischen Betrieb optimiert werden muss.
- 4. Sicherheit und Betriebssicherheit:** Um den Betrieb in einer automatisierten Produktionsumgebung zu gewährleisten, müssen alle Sicherheitsmechanismen in der physischen Station implementiert und zuverlässig funktionsfähig sein. Dazu gehören Sensoren und Sicherheitsfunktionen, die eine sichere Arbeitsumgebung schaffen, etwa durch das Verhindern von Kollisionen oder das Anhalten der Station bei unerwarteten Zwischenfällen.
- 5. Netzwerkintegration und Datenkommunikation:** Da die Vermessungsstation in ein „smartes“ Produktionskonzept eingebunden wird, muss sie mit den anderen Maschinen und Steuerungen innerhalb der Produktionsumgebung kommunizieren können. Die Station sollte über eine verlässliche Schnittstelle verfügen, die Daten an das übergeordnete System überträgt und aus der laufenden Produktion empfangen kann.

Diese Anforderungen bieten eine klare Grundlage, um die Funktionalität und Qualität der Vermessungsstation im digitalen Zwilling und in der physischen Umsetzung zu evaluieren. Das Erreichen dieser Ziele wird entscheidend sein, um die Station zuverlässig und funktionsfähig in die smarte Produktionsumgebung der HAW Hamburg zu integrieren und den Anforderungen der Industrie 4.0 gerecht zu werden.

3.2 Konzeptentwicklung für die Messstation

In diesem Kapitel wird das Konzept für die physische Messstation entwickelt, wobei der Fokus auf der strukturellen Gestaltung eines stabilen und funktionalen Gestells liegt, das die vorhandenen Komponenten sinnvoll integriert.

Bereits zu Beginn dieser Arbeit stehen wesentliche Bauteile, insbesondere von *Festo* und dem Institut für Produktionstechnik (IPT), zur Verfügung. Dazu gehören neben den mechanischen

Komponenten auch die Automatisierungssoftware von *Siemens*, die für die SPS-Steuerung eingesetzt wird.

Die Herausforderung besteht darin, ein robustes und kompaktes Gestell zu konzipieren, das die *Festo*-Komponenten, den 3D-Scanner, den Drehteller und den Messrechner sicher aufnimmt, dabei den räumlichen Gegebenheiten gerecht wird sowie vor äußeren Einflüssen wie Staub und Schmutz schützt.

Eine zentrale Anforderung bei der Entwicklung des Konzepts war der begrenzte Platz, der die Anordnung aller Komponenten auf der Arbeitsfläche erschwert. Um die Machbarkeit der geplanten Konstruktion zu überprüfen und die Grundlage für den digitalen Zwilling zu schaffen, wird ein detailliertes Modell der Station im Computer Aided Design - Programm (CAD) erstellt. Dieses Modell ermöglicht es, die räumliche Anordnung zu simulieren und die Platzierung der Komponenten zu optimieren.

3.2.1 Mechanische Komponenten

In diesem Abschnitt werden die mechanischen Hauptkomponenten der Messstation vorgestellt. Diese Komponenten – das Dreh-Hub-Modul, das Transportband und der Drehteller – sind bereits zu Beginn der Arbeit am Institut für Produktionstechnik (IPT) vorhanden und bilden das Herzstück der Messstation. Jede Komponente ist gut auf die Anforderungen der Station abgestimmt und ermöglicht präzise Bewegungen und die sichere Positionierung des Werkstücks für den Messprozess.

Transportband von *Festo*

Das Transportband von *Festo* hat eine Länge von 700 mm und ist für die reibungslose Übergabe des Werkstücks von der vorangehenden Produktionsstation zur Messstation und anschließend zur nächsten Station konzipiert. Es ist mit Reflexlichttastern und Lichtschranken ausgestattet, die eine zuverlässige Detektion und Positionierung der Werkstücke ermöglichen.

Der integrierte DC-Motor (24 V, 1,5 A) sorgt für eine kontrollierte Geschwindigkeit und gewährleistet einen sicheren Materialfluss. Das Band ist in Abbildung 10 dargestellt und zeigt seine Funktionen und seine Rolle in der Automatisierung der Produktionslinie.

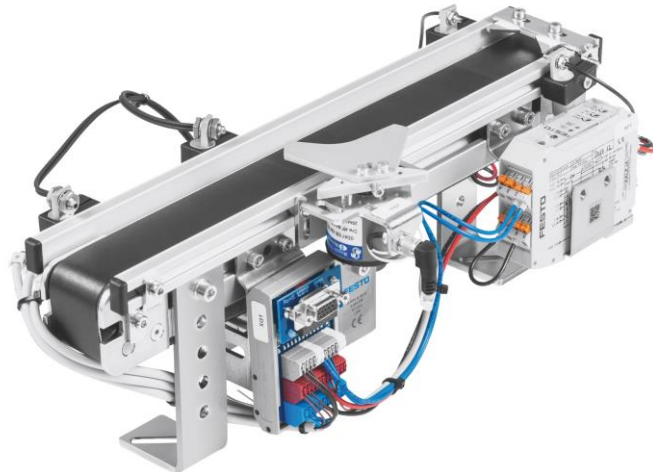


Abbildung 10: Modul Band⁶⁸

Dreh-Hub-Modul von Festo

Das Dreh-Hub-Modul von Festo (siehe Abbildung 11) ist eine kombinierte, pneumatisch angetriebene Einheit, die eine vertikale Hub- und eine horizontale Drehbewegung ausführen kann. Der integrierte Parallelgreifer des Moduls ermöglicht das sichere Greifen und Bewegen von Werkstücken mit einem Durchmesser bis zu 40 mm. Die Hublänge beträgt 20 mm, und der Drehwinkel kann zwischen 0° und 180° frei eingestellt werden. Der pneumatische Antrieb sorgt für schnelle und präzise Bewegungen, die für den Produktionsprozess erforderlich sind. Sensoren und ein Mini-I/O-Terminal erlauben die präzise Steuerung und Positionierung des Moduls. Das Dreh-Hub-Modul ist so positioniert, dass es das Werkstück optimal vom Förderband heben und für den Messprozess ausrichten kann.



Abbildung 11: Dreh-Hub-Modul⁶⁹

⁶⁸ vgl. Festo Didactic InfoPortal: Modul Band.

⁶⁹ vgl. Festo Didactic InfoPortal: Dreh-Hub-Modul.

Drehteller NCTB400C von AICON

Der Drehteller NCTB400C von AICON bietet eine präzise Drehbewegung mit einer hohen Genauigkeit und Wiederholbarkeit, die für die exakte Positionierung des Werkstücks während der Messung notwendig ist.

Er kann Lasten von bis zu 100 kg tragen und wird über eine USB-Schnittstelle gesteuert, die eine einfache Integration in die Steuerungssoftware ermöglicht. Der Drehteller erreicht eine maximale Drehgeschwindigkeit von 2,5 Umdrehungen pro Minute und erlaubt Rotationen mit einer Genauigkeit von bis zu 1/10 Grad. In Abbildung 12 ist der Drehtisch mit seiner Bedieneinheit (ViewMaster) zu sehen.



Abbildung 12: Drehtisch ⁷⁰

Gestell-Komponenten und Struktur

Um den Anforderungen an die mechanische Stabilität, Präzision und Dämpfung gerecht zu werden, wie in Kapitel 3.1.2 „Anforderungen an die physische Vermessungsstation“, Punkt 1 beschrieben, ist es zunächst notwendig, geeignete Komponenten für ein stabiles und flexibel einstellbares Gestell auszuwählen. Aluminiumprofile erweisen sich dabei als besonders geeignet, da sie nicht nur eine hohe Stabilität und Belastbarkeit bieten, sondern auch modular anpassbar sind, sodass Änderungen und Erweiterungen leicht möglich sind.

Die mechanische Basis der Messstation besteht somit aus einem robusten Gestell aus Aluminiumprofilen, die von SMT bezogen werden. Das Gestell dient zur strukturellen Stabilität und zur sicheren Aufnahme der Festo-Komponenten, des Drehtellers sowie weiterer elektronischer Geräte wie des Edu-Trainers und des Messrechners. Die Aluminiumprofile und zusätzliche Befestigungskomponenten werden aufgrund ihrer modularen Anpassungsfähigkeit und hohen Stabilität ausgewählt. Durch den Einsatz von Stellfüßen mit Gummieinsätzen wird eine ausreichende Dämpfung gegen dynamische Kräfte gewährleistet.

⁷⁰ vgl. NCTB400C | OVR Turntable.

Aluminiumprofile und -komponenten

Die Aluminiumprofile bilden das Grundgerüst der Station. Folgende Profile und Komponenten werden verwendet:

- **Profil 6 60x60** (verschiedene Längen) für die tragenden Strukturen, um die Stabilität und Belastbarkeit zu gewährleisten
- **Profil 8 240x40** als Montageplatten, die die Arbeitsfläche bildet und zur Befestigung der mechanischen Komponenten dienen ⁷¹
- **Laschen und Befestigungswinkel:** Diese werden zur Befestigung verwendet und sorgen für die Stabilisierung der Profilen und Verbindungen
- **Abdeckkappen und Polycarbonat-Scheiben:** Diese werden als Schutzverkleidung verwendet und bieten eine zusätzliche Schutzfunktion gegen Staub und Schmutz ⁷²

MDF-Platten für die Arbeitsflächen

Zur sicheren Lagerung des Edu-Trainers und des Messrechners werden zwei MDF-Platten als Arbeitsflächen installiert. MDF (Mitteldichte Faserplatte) ist als Material für die Platten geeignet, da es eine glatte, feste Oberfläche bietet und leicht zu montieren ist. Die Platten sind durch die Profile höhenverstellbar und erlauben eine flexible Anpassung an die räumlichen Anforderungen der Messstation.

Zusammen bilden die Komponenten des Gestells eine stabile und modulare Struktur, die die Messstation optimal stützt und die Sicherheits- und Stabilitätsanforderungen der Produktionsumgebung erfüllt.

3.2.2 Elektronische und pneumatische Komponenten

Die elektronischen und pneumatischen Komponenten der Messstation sind zentral für die Steuerung, Automatisierung und Integration in die digitale Umgebung. Neben der CPU 1516-3 PN/DP im Edu-Trainer, die als Hauptsteuerungseinheit dient, kommen weitere wichtige Module und Sensoren von *Festo* zum Einsatz, die die pneumatischen und automatisierten Bewegungen unterstützen und überwachen. Diese Komponenten werden im Folgenden detailliert beschrieben.

CPU 1516-3 PN/DP im Edu-Trainer

Die CPU 1516-3 PN/DP von *Siemens* ist das Herzstück der Steuerungseinheit und Teil des modularen S7-1500 Systems. Diese CPU, die im Edu-Trainer eingebaut ist, ermöglicht die SPS-Steuerung der mechanischen und elektronischen Komponenten der Messstation. Die Bezeichnung 1516-3 PN/DP enthält wichtige Informationen über die Funktionalität und Schnittstellen der CPU:

⁷¹ vgl. SMT_Angebot_126235.pdf.

⁷² vgl. SMT_Angebot_126237.pdf.

- **15:** Kennzeichnet die Generation der S7-1500 Reihe, die für hohe Leistungsfähigkeit und modulare Erweiterbarkeit steht.
- **16:** Die Modellnummer innerhalb der Reihe, wobei höhere Zahlen für leistungsfähigere CPUs stehen.
- **3:** Die Anzahl der integrierten Kommunikationsschnittstellen der CPU.
- **PN/DP:** Diese Abkürzung weist darauf hin, dass die CPU sowohl PROFINET (PN) als auch PROFIBUS DP (DP) Schnittstellen besitzt, die eine flexible Integration in industrielle Netzwerke ermöglichen.

Die CPU 1516-3 PN/DP bietet eine hohe Verarbeitungsgeschwindigkeit und umfassende Kommunikationsmöglichkeiten. Sie ist für anspruchsvolle Anwendungen in der Automatisierung konzipiert und ermöglicht eine präzise Steuerung der Station. Durch die Unterstützung von PROFINET und PROFIBUS können verschiedene Geräte und Module effizient vernetzt und gesteuert werden. Dies ist insbesondere für die Kommunikation zwischen den SPS-gesteuerten mechanischen Komponenten und der übergeordneten Steuerungssoftware relevant.⁷³

Weitere Komponenten von Festo⁷⁴

Neben der zentralen Steuerungseinheit umfasst die entwickelte Vermessungsstation mehrere zusätzliche Komponenten, die für den reibungslosen Betrieb der automatisierten Prozesse erforderlich sind. Eine wesentliche Rolle übernimmt dabei der **Motorcontroller**, der speziell für die Ansteuerung des Förderbandmotors zuständig ist. Er regelt Parameter wie die Förderbandgeschwindigkeit, die Beschleunigung sowie die Drehrichtung des Bandes (Links- oder Rechtslauf).

Für die pneumatischen Komponenten der Anlage ist eine **Wartungseinheit** integriert, die die Druckluftversorgung sicherstellt. Diese Einheit besteht aus einem Filterregelventil mit Druckmesser zur präzisen Einstellung des Luftdrucks. Zusätzlich ist ein Sicherheitsventil vorhanden, das bei Bedarf die Luftzufuhr automatisch unterbricht und die Anlage entlüftet, um sie in einen sicheren Zustand zu versetzen.

Ein Funktionselement im Materialfluss ist der **pneumatische Stopper Zylinder**, ein einfach wirkender Pneumatik Zylinder mit Ventilsteuerung. Dieser dient dazu, Werkstücke gezielt zu stoppen oder temporär in Position zu halten. Er fungiert somit als Positioniermodul innerhalb der Transportstrecke und ist speziell darauf ausgelegt, die Bauteile am Förderband präzise anzuhalten und auszurichten.

Die **E/A-Kopplungseinheit**, das sogenannte C-Interface, stellt die Verbindung zwischen den Modulen und der SPS her. Sie überträgt sowohl digitale als auch analoge Signale und nutzt dabei 24-polige SysLink- sowie 15-polige D-Sub-Steckverbindungen. Durch integrierte Status-LEDs wird eine einfache Überwachung der Signalzustände ermöglicht, was insbesondere im laufenden Betrieb zur Fehlersuche beiträgt.

⁷³ vgl. Siemens AG (2019).

⁷⁴ vgl. Angebot_Festo_2004059061.pdf.

Ein weiterer wichtiger Bestandteil ist der **Reflexlichttaster**, der zur Objekterkennung auf dem Förderband eingesetzt wird. Mit einer Nennschaltweite von 30 mm erkennt er zuverlässig das Vorhandensein eines Werkstücks und überträgt ein entsprechendes Signal an die SPS. Dies dient als Auslöser für bestimmte Bewegungssequenzen und sorgt so für einen automatisierten und kontrollierten Ablauf innerhalb der Station. Die Kombination dieser Komponenten gewährleistet eine präzise und sicherheitsorientierte Steuerung des Gesamtsystems.

Unterstützende Software: TIA-Portal, WinCC RT und Ciros

Zusätzlich zu den Hardware-Komponenten wird die Steuerung der (digitalen) Messstation und Vermessung der Bauteile durch verschiedene Softwarelösungen unterstützt. Für Details zur Anwendung und Funktionalität von Ciros (Kapitel 2.2.3) sowie TIA-Portal (Kapitel 2.3.2) wird auf die entsprechenden Kapitel verwiesen.

Darüber hinaus kommt die Mess-Software PolyWorks zum Einsatz, die u. a. im nachfolgenden Kapitel 3.2.3 erläutert wird.

Alle beschriebenen elektronischen und pneumatischen Komponenten sowie die unterstützenden Softwaresysteme werden bereits zu Beginn dieses Projekts zur Verfügung gestellt. Die Aufgabe besteht darin, diese Komponenten sinnvoll in die Station zu integrieren und optimal aufeinander abzustimmen, um eine genaue und zuverlässige Steuerung, Automatisierung und Vermessung der Messstation zu gewährleisten.

3.2.3 Integration des 3D-Scanners und Messtechnik

In diesem Abschnitt wird die Integration des 3D-Scanners AICON HE-Scan und der Mess-Software PolyWorks in das Konzept der Messstation beschrieben. Diese Komponenten sind essenziell für die präzise Erfassung und Vermessung der Werkstücke und spielen eine zentrale Rolle in der Qualitätssicherung der Produktionsprozesse. Der 3D-Scanner ermöglicht es, exakte geometrische Daten des Werkstücks zu erfassen, während die Software PolyWorks die Messdaten analysiert und den Drehtisch zur optimalen Ausrichtung steuert.

Strukturlichtscanner AICON HE-Scan

Der AICON HE-Scan, ursprünglich vom deutschen Unternehmen AICON 3D Systems entwickelt und später von Hexagon übernommen, ist ein optischer Streifenlichtscanner, der für hochpräzise 3D-Vermessungen eingesetzt wird. Der Scanner arbeitet nach dem Prinzip der strukturierten Lichtprojektion: Ein Lichtmuster (z. B. Streifen) wird auf die Objekt Oberfläche projiziert und die resultierenden Verzerrungen werden durch Kameras erfasst, um daraus eine hochauflösende dreidimensionale Punktwolke zu berechnen. Diese Punktwolke bildet die Form und Maße des Bauteils mit hoher Genauigkeit ab.

Der HE-Scan ist für seine hohe Messgenauigkeit, Wiederholbarkeit und Geschwindigkeit bekannt und wird insbesondere in industriellen Anwendungen zur Erfassung komplexer Freiformflächen, zur Qualitätssicherung und in Reverse-Engineering-Prozessen eingesetzt.⁷⁵

⁷⁵ vgl. o. V.: SmartScan | Streifenlichtscanner.

In der Messstation ist der Scanner so integriert, dass er Bauteile auf dem Drehtisch erfassen kann, ohne deren Positionierung oder Bewegung zu stören. Eine eigens konstruierte Halterung sorgt für die stabile Befestigung des Scanners und gewährleistet wiederholbare Messergebnisse.

Software PolyWorks

PolyWorks ist eine leistungsstarke Software zur Verarbeitung und Analyse von 3D-Messdaten, die speziell auf die Zusammenarbeit mit 3D-Scannern wie dem AICON HE-Scan ausgelegt ist. PolyWorks ermöglicht die detaillierte Auswertung der erfassten Punktwolken und die Identifikation von Abweichungen oder Toleranzen, die im Produktionsprozess wichtig sind. Neben der Analyse der Messdaten wird PolyWorks auch verwendet, um den Drehtisch der Messstation zu steuern und das Werkstück in die optimale Messposition zu bringen.

Die Software ermöglicht die Steuerung des Drehtisches und sorgt dafür, dass das Werkstück während der Messung in verschiedenen Positionen erfasst werden kann, um ein vollständiges 3D-Modell zu erstellen. Durch die Synchronisierung der Drehtischbewegung mit dem Scanner können alle relevanten Geometrien des Werkstücks erfasst werden, was zur Qualitätssicherung und Dokumentation der Messergebnisse beiträgt. Die Einbindung der PolyWorks-Software stellt sicher, dass die Messprozesse effizient ablaufen und die Ergebnisse direkt in die Qualitätskontrolle der Station integriert werden können.⁷⁶

Einbindung in das Gesamtkonzept

Zusammenfassend bilden die mechanischen Komponenten, die elektronischen und pneumatischen Steuerungselemente sowie die Messtechnik ein aufeinander abgestimmtes Konzept, das die Grundlage für eine automatisierte Messstation darstellt.

Die mechanischen Komponenten sorgen für die nötige Stabilität, Dämpfung und flexible Anpassungsfähigkeit der Station und bieten eine sichere Halterung für alle weiteren Elemente.

Die elektronischen und pneumatischen Komponenten übernehmen die Steuerung und Automatisierung der Bewegungsabläufe, die für die Handhabung und Positionierung des Werkstücks erforderlich sind. Hier spielen die CPU im Edu-Trainer und die Steuerungssoftware TIA-Portal eine zentrale Rolle, indem sie die Synchronisierung der Bewegungen ermöglichen und den Betrieb der Anlage überwachen.

Die Messtechnik, bestehend aus dem AICON HE-Scan und der Software PolyWorks, ergänzt dieses Konzept durch die Erfassung und Analyse der Werkstücke, die den Qualitätsansprüchen der vernetzten Produktion gerecht wird. Durch die Steuerung des Drehtisches und die detaillierte Auswertung der Messdaten werden die Werkstücke optimal für den Messprozess positioniert und ihre Geometrien umfassend dokumentiert.

Dieses Zusammenspiel aller Komponenten schafft eine funktionale Einheit, die alle Anforderungen der Industrie 4.0 an Präzision, Vernetzung und Automatisierung erfüllt. Im folgenden Kapitel wird dieses Konzept in einem digitalen Modell umgesetzt und als digitaler Zwilling der Messstation simuliert, wodurch eine erste Test- und Optimierungsumgebung für den späteren Betrieb geschaffen wird.

⁷⁶ vgl. o. V.: Software für 3D-Maßanalyse und Qualitätskontrolle | PolyWorks.

4 Entwicklung des digitalen Zwillings in Ciro

Die nachfolgenden Ausführungen befassen sich mit der Erstellung und Konfiguration eines digitalen Zwillings der Vermessungsstation unter Verwendung der Simulationssoftware Ciro von Festo. Ziel ist es, ein virtuelles Abbild der Anlage zu erzeugen, das sowohl die mechanischen als auch die elektronischen und pneumatischen Komponenten der Station realitätsnah repräsentiert. Auf dieser Grundlage kann das Zusammenspiel der Module sowie deren Steuerungslogik simuliert und analysiert werden. Die Simulation dient nicht nur der Visualisierung, sondern auch der Erprobung und Optimierung der Prozesse und Steuerbefehle, wodurch potenzielle Fehler frühzeitig erkannt und behoben werden können. Dieses Kapitel beschreibt die schrittweise Umsetzung des digitalen Zwillings – von der Modellierung über die Integration der Steuerung bis hin zur Erprobung im virtuellen Umfeld – und stellt damit eine Grundlage für die spätere Implementierung der physischen Anlage dar.

4.1 Erstellung des CAD-Modells der Vermessungsstation

Um den digitalen Zwilling der Messstation zu erstellen, wird zunächst auf die Definition eines digitalen Zwillings aus Kapitel 2.2 sowie auf die spezifischen Anforderungen aus Kapitel 3.1.1 Bezug genommen. Diese bieten die Grundlage, um zu identifizieren, welche Bauteile und Parameter für die virtuelle Abbildung der Station notwendig sind und wie deren Erstellung gestaltet werden kann.

Zunächst wird ein geeignetes CAD-Programm ausgewählt, um die Station virtuell aufzubauen. Hier fällt die Wahl auf Fusion 360, ein leistungsstarkes CAD-Tool, mit dem bereits Projekte im Studium erfolgreich umgesetzt werden konnten.

Fusion 360 bietet mehrere Vorteile:

- **Intuitive Bedienung:** Die benutzerfreundliche Oberfläche erleichtert die Modellierung und Bearbeitung komplexer Bauteile.
- **Cloud-basierte Datenverwaltung:** Projektdateien können einfach gespeichert und geteilt werden, wodurch die Zusammenarbeit und der Zugriff erleichtert werden.
- **Vielseitige Funktionen:** Es unterstützt verschiedene Dateiformate wie .stl und .step und bietet Werkzeuge für präzises Modellieren, Baugruppenverwaltung und Simulation.

Ausgehend von den Anforderungen an die Station – sowohl virtuell als auch physisch – wird zunächst eine Bauteilliste in Excel erstellt (siehe A: Stationsbauteile-Liste). Diese Liste dient als zentrale Übersicht und enthält die Abmessungen, Stückzahlen, Verwendungszwecke sowie Checkboxes, um zu dokumentieren, ob:

- Die entsprechenden Bauteile bereits als CAD-Dateien vorhanden sind.
- Die Bauteile bereits bestellt oder noch zu beschaffen sind.

Diese strukturierte Übersicht gewährleistet, dass keine benötigten Komponenten vergessen werden. Basierend auf der Liste wird dann ermittelt, welche Bauteile im .stl-Format verfügbar sind oder noch heruntergeladen bzw. erstellt werden müssen.

Die Hauptbauteile wie Aluminiumprofile, Befestigungswinkel, Abdeckungen und Verbindungselemente werden bei *SMT GmbH* bestellt.⁷⁷ Parallel dazu werden die benötigten CAD-Dateien von der Herstellerwebseite von *Item Industrietechnik GmbH*⁷⁸ heruntergeladen, um eine nahtlose Integration der Bauteile in das CAD-Modell sicherzustellen.

Im nächsten Schritt wird die Station vollständig virtuell aufgebaut, um sicherzustellen, dass die Anforderungen aus Kapitel 3.1.1 und die räumlichen Gegebenheiten erfüllt werden. Der CAD-Aufbau dient anschließend als Grundlage für den Import in die Simulationssoftware Ciro, in der der digitale Zwilling weiterentwickelt wird.

Der Aufbau der Vermessungsstation im CAD-Programm Fusion 360 erfolgt in mehreren Schritten, um die räumliche Anordnung und die mechanische Struktur gemäß den Anforderungen aus Kapitel 3.1.1 darzustellen. In den folgenden Abschnitten wird der Konstruktionsprozess beschrieben und durch Abbildungen illustriert.

Grundstruktur des Gestells

In Abbildung 13 sind zunächst die vier vertikalen Aluminiumprofile (60x60 mm) dargestellt, die als tragende Säulen der Station dienen. An den unteren Enden der Profile sind einstellbare Stellfüße montiert, die es ermöglichen, die Station präzise höhenverstellbar zu machen und an die vorgegebenen 781 mm Arbeitshöhe der anderen Förderbänder/Stationen anzupassen. Diese Füße sorgen zudem für Stabilität.

⁷⁷ vgl. GmbH, S.M.T.

⁷⁸ vgl. Profiltechnik – item.

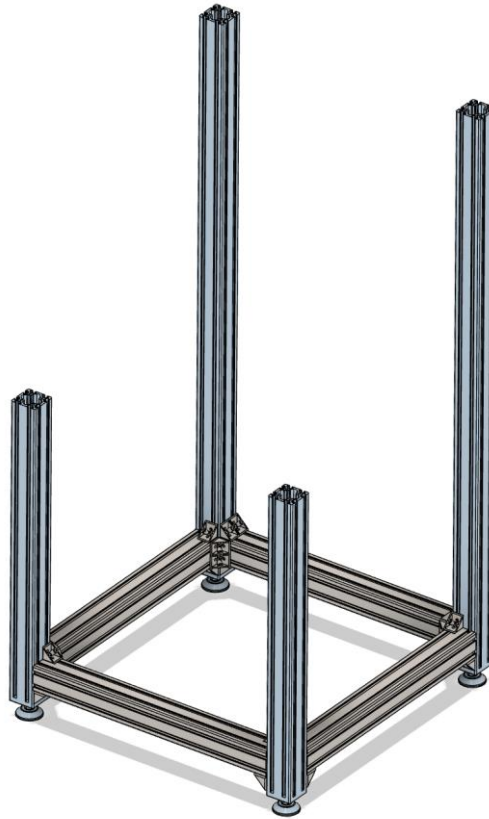


Abbildung 13: CAD-Grundgerüst - perspektivische Ansicht 1 ⁷⁹

Zur Verbindung der vertikalen Profile und zur Herstellung der Steifigkeit des Gestells werden vier horizontale Aluminiumprofile (ebenfalls 60x60mm) verwendet. Diese horizontalen Profile bilden gleichzeitig die Aufnahme­fläche der ersten Ebene. Die Verbindung zwischen den vertikalen und horizontalen Profilen erfolgt durch 90°-Befestigungswinkel, die in Abbildung 14 genauer dargestellt sind. Die Schraubverbindungen zwischen den Profilen und Winkeln sorgen für eine stabile und präzise Struktur.

⁷⁹ Eigene Darstellung

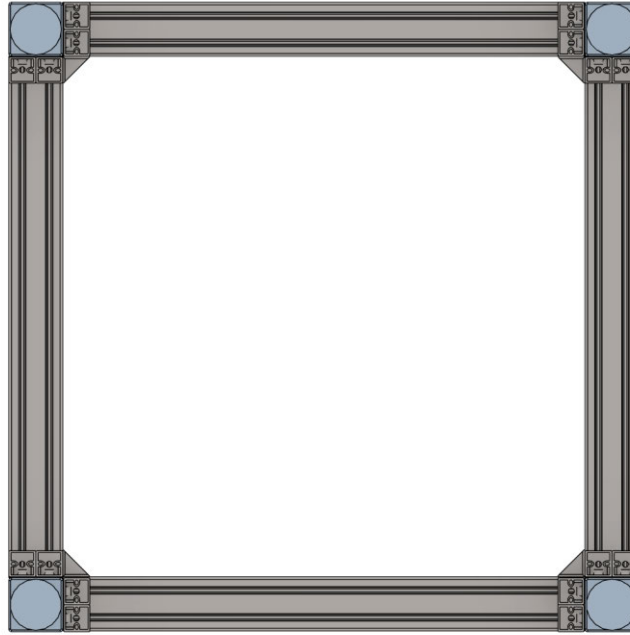


Abbildung 14: CAD-Grundgerüst - Ansicht von unten⁸⁰

Aufbau der weiteren Ebenen

In Abbildung 15 sind zusätzliche Aluminiumprofile mit kleineren Abmaßen (30x30 mm) dargestellt, die die Auflagefläche für die zweite Ablageebene bilden. Zudem sind die beiden MDF-Platten zu sehen, die die Auflageflächen bilden und als Basis für die Platzierung des Edu-Trainers und des Messrechners dienen.

⁸⁰ Eigene Darstellung

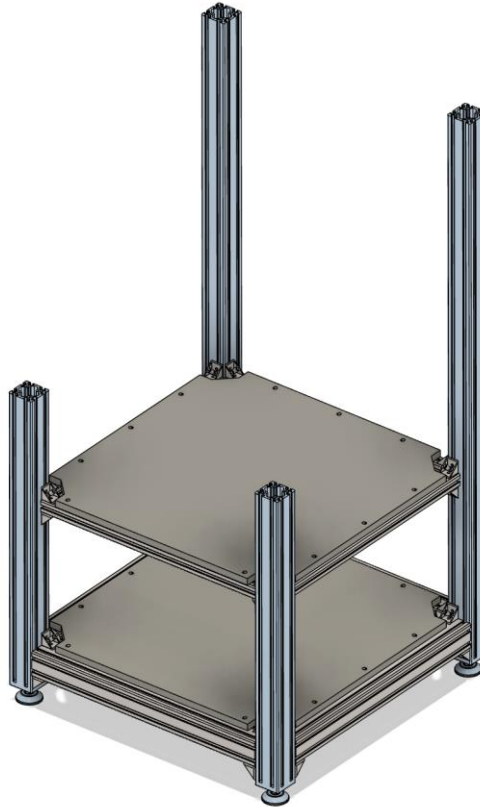


Abbildung 15: CAD-Grundgerüst - perspektivische Ansicht 2 ⁸¹

Die dritte Ebene besteht aus horizontalen Aluminiumprofilen der gleichen Größe wie die Grundstruktur (60x60mm). Diese Ebene bietet Platz für größere und schwerere Komponenten wie den Drehtisch und das Förderband.

In Abbildung 16 ist die obere Ebene mit den Montageplatten dargestellt, die als Befestigungsfläche für die Komponenten der Station dienen. Um den 3D-Scanner später sicher zu positionieren und dessen optimale Arbeitsdistanz zum Bauteil einzustellen, wurde ein zusätzliches horizontales Aluminiumprofil zwischen den längeren, hinteren vertikalen Profilen montiert. Dieser Befestigungspunkt ist der Grund, warum die hinteren vertikalen Aluminiumprofile länger sind als die vorderen. Hier lässt sich gegebenenfalls noch ein Bildschirm montieren.

⁸¹ Eigene Darstellung

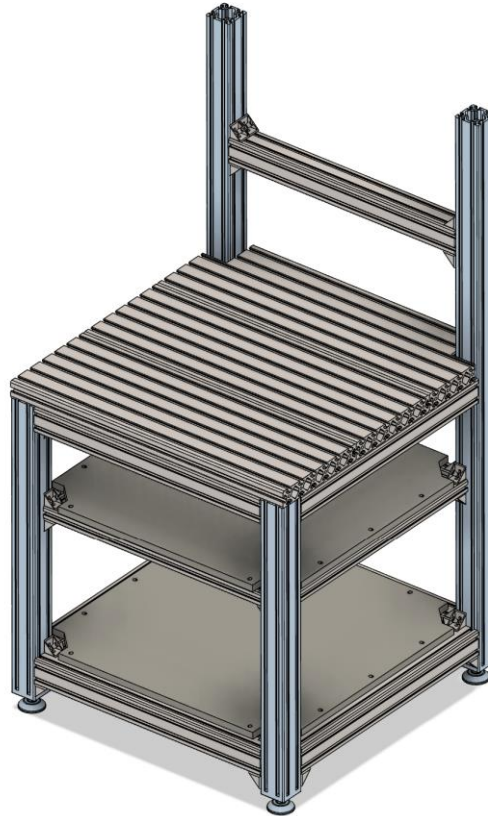


Abbildung 16: CAD-Grundgerüst - perspektivische Ansicht 3⁸²

Integration der Plexiglasscheiben

In Abbildung 17 sind die Plexiglasscheiben dargestellt, die als Schutzverkleidung an der Station montiert werden. Die seitlichen Plexiglasscheiben links und rechts sind in die äußeren Nuten der vertikalen und horizontalen Aluminiumprofile gesteckt, um die vorgeschriebene Breite von 700 mm exakt einzuhalten. Diese Maßvorgabe ermöglicht die nahtlose Anbindung der Vor- und Nachstationen, sodass Bauteile ohne Unterbrechung von einem Förderband zum nächsten übergeben werden können.

Die hintere Plexiglasscheibe hingegen wird bewusst außen an die Aluminiumprofile geschraubt, um bei Bedarf leicht entfernt werden zu können. Dadurch ist ein einfacher Zugriff auf die Geräte im Inneren der Station, wie den Messrechner oder den Edu-Trainer, gewährleistet.

Die vordere Plexiglasscheibe ist mit zwei Scharnieren und einem Schließsystem ausgestattet und fungiert als Tür. Diese Konstruktion ermöglicht es, die Station im Lehrbetrieb sicher zu verschließen und gleichzeitig einen bequemen Zugang zu bieten. Im CAD-Modell sind die Plexiglasscheiben nicht transparent dargestellt, jedoch werden für die physische Station transparente Scheiben bestellt, um die Inhalte der Station sichtbar zu machen. Dies verbessert die Anschaulichkeit und Eignung der Station für den Lehrbetrieb.

⁸² Eigene Darstellung

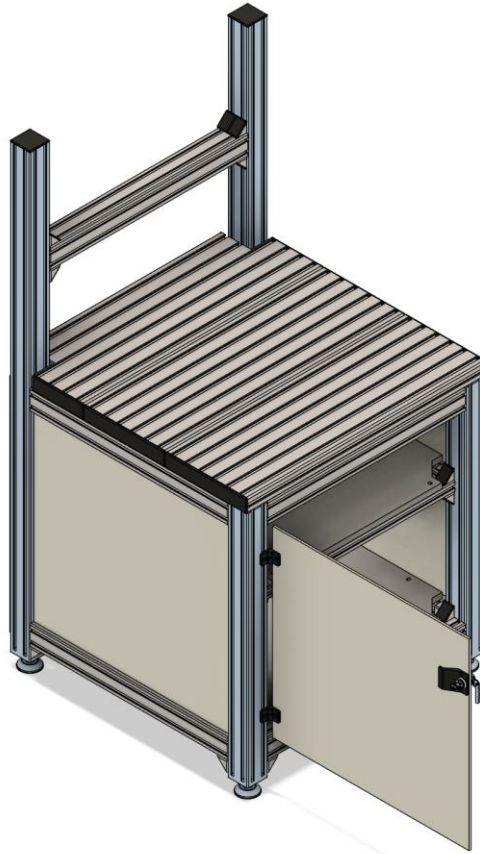


Abbildung 17: CAD-Gestell - perspektivische Ansicht 4

Platzierung der Komponenten

In Abbildung 18 sind die Komponenten in der Station dargestellt:

- Der **Edu-Trainer** in der ersten (unteren) Ebene
- der **Messrechner** in der zweiten Ebene
- der **3D-Scanner** auf dem horizontalen Profil der oberen Ebene sowie
- alle weiteren Stationskomponenten

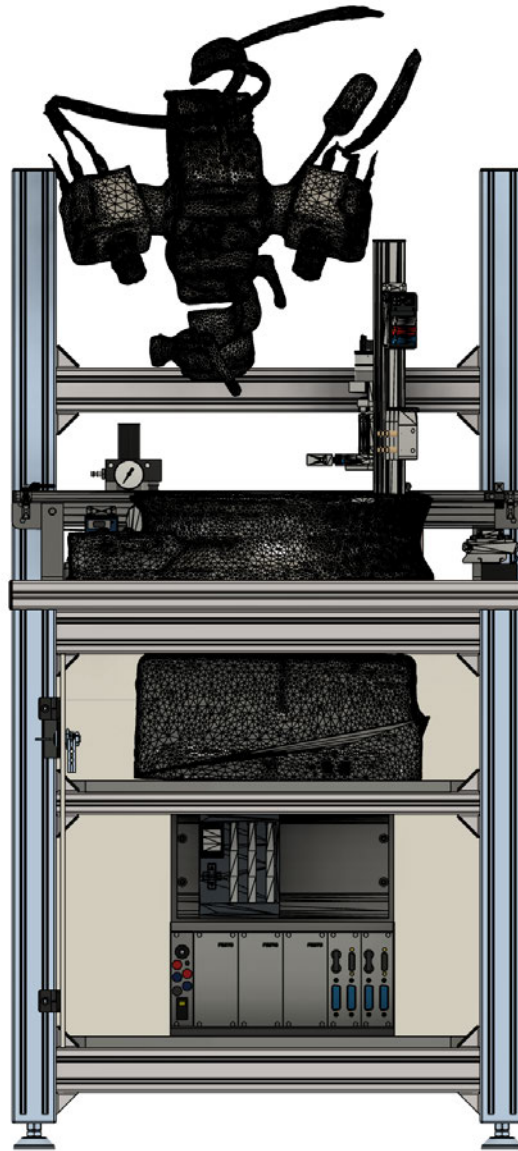


Abbildung 18: CAD-Vermessungsstation - Frontalansicht ⁸³

Da für den Messrechner, den Drehtisch und den 3D-Scanner keine CAD-Dateien von den Herstellern bereitgestellt werden und auch keine Daten im Internet verfügbar sind, werden diese Komponenten mit einer Smartphone-Applikation aufgenommen und anschließend in das CAD-Modell integriert, wie in Abbildung 19 gut zu erkennen ist.

⁸³ Eigene Darstellung

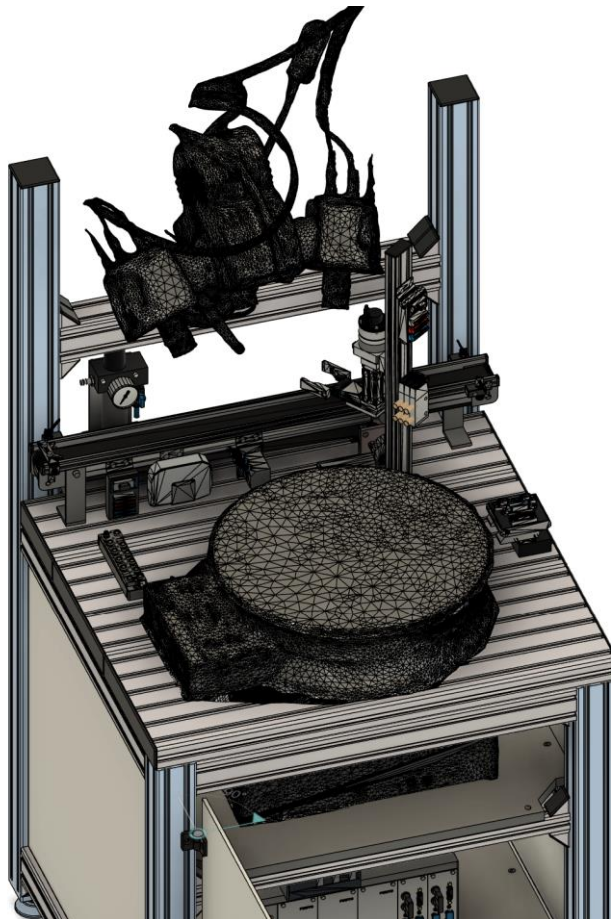


Abbildung 19: CAD-Vermessungsstation - perspektivische Ansicht ⁸⁴

Dieses CAD-Modell ermöglicht es, bereits in der Planungsphase zu prüfen, ob alle Komponenten ausreichend Platz haben und wie sie optimal für einen automatischen Betrieb angeordnet werden können.

Somit ist das CAD-Modell der Messstation erfolgreich erstellt und kann als Grundlage für die Integration in die Simulationssoftware Ciros verwendet werden. Das Modell bildet die Basis für die weitere Entwicklung des digitalen Zwillings.

4.2 Aufbau und Konfiguration des digitalen Zwillings in Ciros

In diesem Kapitel wird der digitale Zwilling der Vermessungsstation in der Simulationssoftware Ciros aufgebaut und konfiguriert. Ziel ist es, das zuvor entwickelte CAD-Modell der Station in die virtuelle Umgebung zu integrieren, die Simulation der Bewegungsabläufe zu ermöglichen und die Steuerungslogik mit einer programmierbaren Steuerung umzusetzen. Neben dem Import und der Ergänzung der Komponenten wird die SPS-Programmierung im TIA-Portal entwickelt und in Ciros integriert, um eine realitätsnahe Simulation der Steuerung und Mechanismen sicherzustellen.

Mit diesen Schritten wird die Grundlage geschaffen, um die Simulation in Kapitel 0 zu testen, die Steuerungslogik zu überprüfen und die Bewegungsabläufe des digitalen Zwillings zu optimieren.

⁸⁴ Eigene Darstellung

4.2.1 Integration und Erweiterung der Komponenten in Ciros

Die Integration und Erweiterung der Komponenten in Ciros bildet einen zentralen Schritt, um eine digitale Simulation der Vermessungsstation zu erstellen, die der realen Station exakt entspricht. Ziel ist es, das Zusammenspiel der einzelnen Komponenten so zu gestalten, dass der Bauteiltransport und die Vermessung reibungslos ablaufen kann.

Erstellung eines neuen Modells und Import der Station

Der Prozess beginnt mit der Erstellung eines neuen Modells in Ciros. Über den Pfad "Datei > Neues Modell" wird eine leere Arbeitsfläche für die Modellierung bereitgestellt. Anschließend wird die step-Datei der Vermessungsstation über "Datei > Importieren" importiert. Die step-Datei enthält ein präzises CAD-Modell, das neben dem Gestell auch den nicht in Ciros verfügbaren, aber wesentlichen Komponente wie den smart-Scan HE umfasst (siehe Abbildung 20). Der Import dieser externen Komponente ist essenziell, um die reale Station digital möglichst genau nachzubilden.

Das Gestell bildet die Grundlage des digitalen Zwillings und dient als Basis für die weitere Integration der *Festo-Module*.

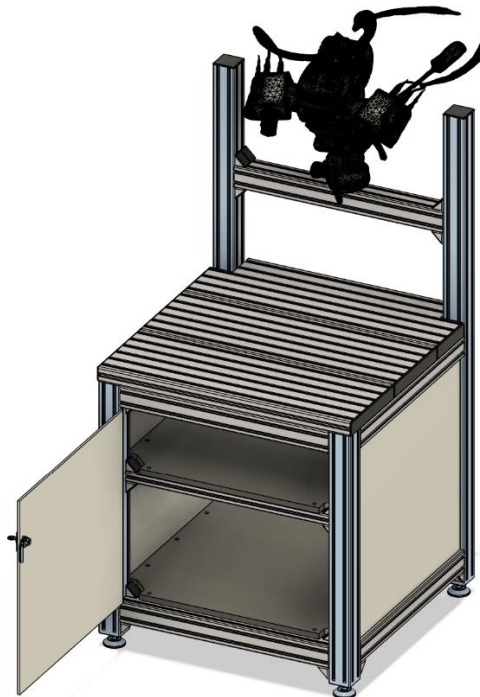


Abbildung 20: grundlegendes CAD-Modell für Ciros Import⁸⁵

Integration der *Festo-Module*

Nach dem Import des Gestells werden die *Festo-Module* integriert, um die Funktionalität der Station abzubilden. Zu den wesentlichen Modulen gehören:

⁸⁵ Eigene Darstellung

- **Transportmodul:** Das Förderbandmodul mit einer Länge von 700 mm ermöglicht den Transport des Bauteils durch die Station.
- **Dreh-Hub-Modul:** Dieses Modul greift das Bauteil und positioniert es auf dem Drehtisch für die Vermessung.
- **C-Interface und Druckventil:** Diese Module gewährleisten die Steuerung und Versorgung der pneumatischen Komponenten.

Die Module können über den Pfad "*Modellierung > Modellbibliothek > MPS-Station*" aus der CiroS-Modellbibliothek hinzugefügt werden. Bestehende MPS-Stationen wie die Station "Messen" dienen als Vorlage, aus der spezifische Komponenten, z. B. das Dreh-Hub-Modul, kopiert und in die eigene Station integriert werden.

Positionierung und Konfiguration der Module

Die präzise Positionierung und Konfiguration der Module sind entscheidend für die korrekte Interaktion zwischen den Komponenten. Über "*Modellierung > Modell-Explorer*" wird der Modellexplorer geöffnet, der alle hinzugefügten Module und Komponenten übersichtlich auflistet (siehe **Error! Reference source not found.**).

Um die Eigenschaften eines Moduls oder Objekts zu bearbeiten, kann über „*Modell-Explorer > Rechtsklick auf Objekt > Eigenschaften*“ das Eigenschaften-Fenster des Moduls geöffnet werden. Im Tab „Lage“ können die Positionen in x, y und z individuell angepasst werden, um die exakte Platzierung im 3D-Raum zu gewährleisten.

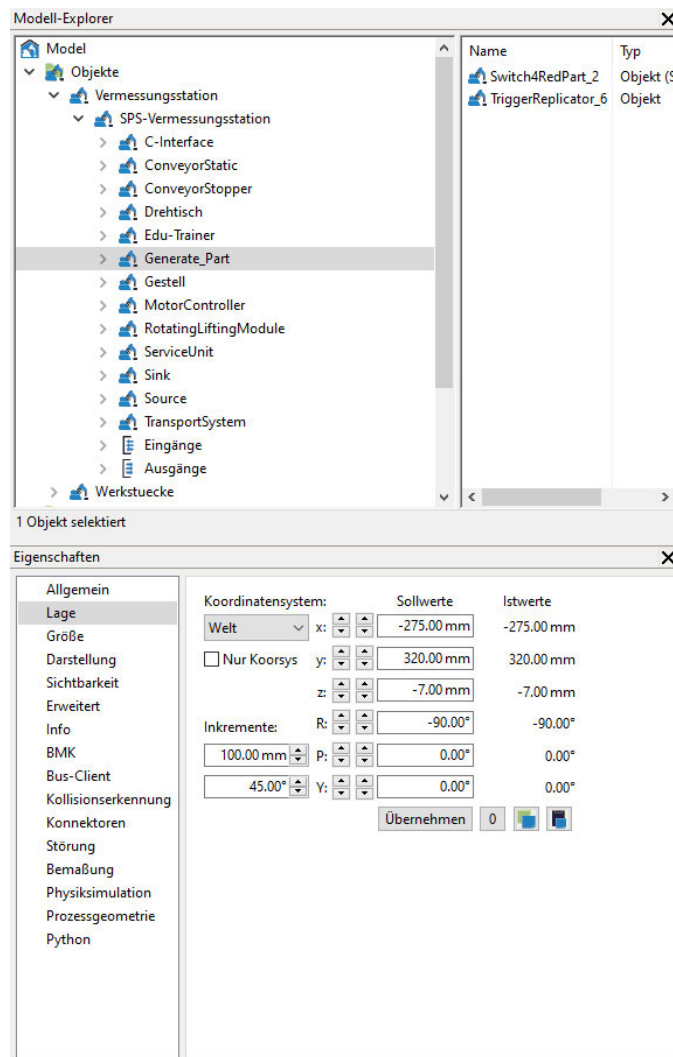


Abbildung 21: Ciros- Modell Explorer und Eigenschaftenseite "Lage"⁸⁶

Um die Stationen optimal miteinander interagieren zu lassen, ist es erforderlich, die spätere physische Station vorab zu berücksichtigen. Eine erste Überlegung betrifft die Position des Transportmoduls. Dieses wird weiter nach hinten verlagert, um im vorderen Bereich genügend Platz für das Dreh-Hub-Modul und den Drehtisch zu schaffen.

Das Dreh-Hub-Modul muss so positioniert werden, dass es das Bauteil problemlos vom Förderband aufnehmen und wieder ablegen kann. Zusätzlich sollte es das Bauteil so weit wie möglich auf den Drehtisch legen, da das Fokusfeld des Scanners nicht den gesamten Drehtischbereich abdeckt. Dabei ist zu beachten, dass das Dreh-Hub-Modul einen maximalen Schwenkradius von 180° besitzt. Innerhalb dieser 180° kann der Schwenkwinkel jedoch flexibel eingestellt werden. Um die Anforderungen zu erfüllen, wird das Dreh-Hub-Modul ca. 35° zum Förderband positioniert, sodass es in einer optimierten Reichweite arbeiten kann.

Das Druckventil wird am hinteren Ende der Station platziert, da sich dort der Druckluftanschluss befindet und ausreichend Platz verfügbar ist. Das C-Interface wird an einer Ecke im vorderen Bereich der Station positioniert, um den Drehtisch nicht zu beeinträchtigen und dennoch leicht zugänglich zu sein.

⁸⁶ Eigene Darstellung

Die Abbildung 22 zeigt die Positionierung der *Festo*-Komponenten und die räumliche Anordnung in der digitalen Station.

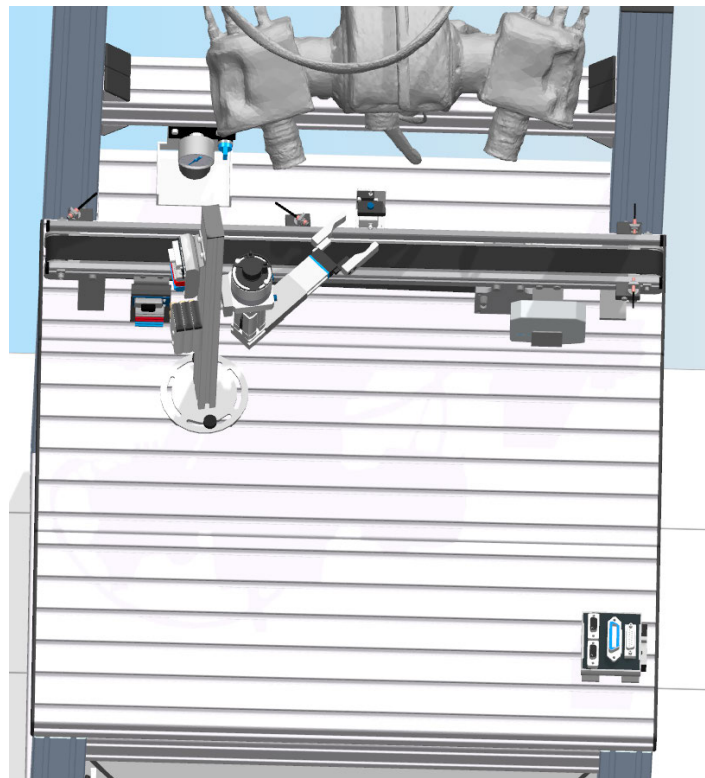


Abbildung 22: Positionierung der Festo Module auf der Arbeitsfläche⁸⁷

Zusätzlich zu den *Festo*-Komponenten werden der Drehtisch und das Bauteil in CiroS erstellt, die den digitalen Zwilling komplementieren und die Grundlage bilden für die Simulation.

Erstellung des Drehtisches und Bauteils

Da der Drehtisch in der virtuellen Simulation bewegt bzw. gedreht werden soll, ist es nicht möglich, eine step-Datei des Drehtisches zu importieren, da solchen Dateien keine Kinematik in CiroS zugeordnet werden kann. Stattdessen wird der Drehtisch mithilfe geometrischer Grundformen direkt in CiroS erstellt und konfiguriert.

Der Drehtisch wird aus mehreren geometrischen Grundkörpern erstellt, die in der CiroS-Modellbibliothek verfügbar sind. Über den Pfad „*Modellierung > Modellbibliothek > Geometrische Grundkörper*“ werden die erforderlichen Elemente hinzugefügt:

1. **Grundplatte:** Ein Zylinder wird als Basis des Drehtisches erstellt, der die physische Plattform repräsentiert.
2. **Drehteller:** Ein weiterer Zylinder wird oberhalb der Grundplatte platziert und stellt den rotierenden Teil des Drehtisches dar.

⁸⁷ Eigene Darstellung

3. **Kontrolleinheit:** Ein Rechteckkörper wird erstellt, das als Anzeige- und Verbindungseinheit dient.

Alle Elemente werden im Modellexplorer miteinander verbunden, um eine konsistente Struktur zu gewährleisten. Die Größe und Position der Grundkörper wird über „Objekt > Eigenschaften > Größe/Lage“ angepasst, sodass die Abmessungen des physischen Drehtisches exakt nachgebildet werden.

Diese Vorgehensweise ermöglicht nicht nur eine realistische Darstellung, sondern bildet auch die Grundlage für die spätere Erstellung von Mechanismen, die den Drehteller rotieren lassen.

Das Bauteil (roter Zylinder), das durch den Prozess transportiert und vermessen wird, wird ebenfalls in Ciro integriert. Um das Bauteil zu erzeugen, wird ein bereits existierendes Objekt aus einer anderen *Festo*-Station kopiert und in die aktuelle Station importiert. Das Bauteil wird zunächst vorne links auf der Vermessungsstation positioniert, um eine klare Sichtbarkeit für die Simulation zu gewährleisten.

Das Bauteil soll so konfiguriert werden, dass es mithilfe eines Mechanismus, der per Mausklick aktiviert wird, am Bandanfang „erzeugt“ wird. Von dort wird es in den Simulationsprozess eingeführt und von den verschiedenen Modulen transportiert. Diese Funktionalität gewährleistet eine kontrollierte Integration des Bauteils in den Prozessablauf und ermöglicht eine präzise Überprüfung der Simulation.

Mit der Erstellung des Drehtisches und des Bauteils sind die grundlegenden physischen Elemente der Vermessungsstation vollständig integriert (siehe Abbildung 23).

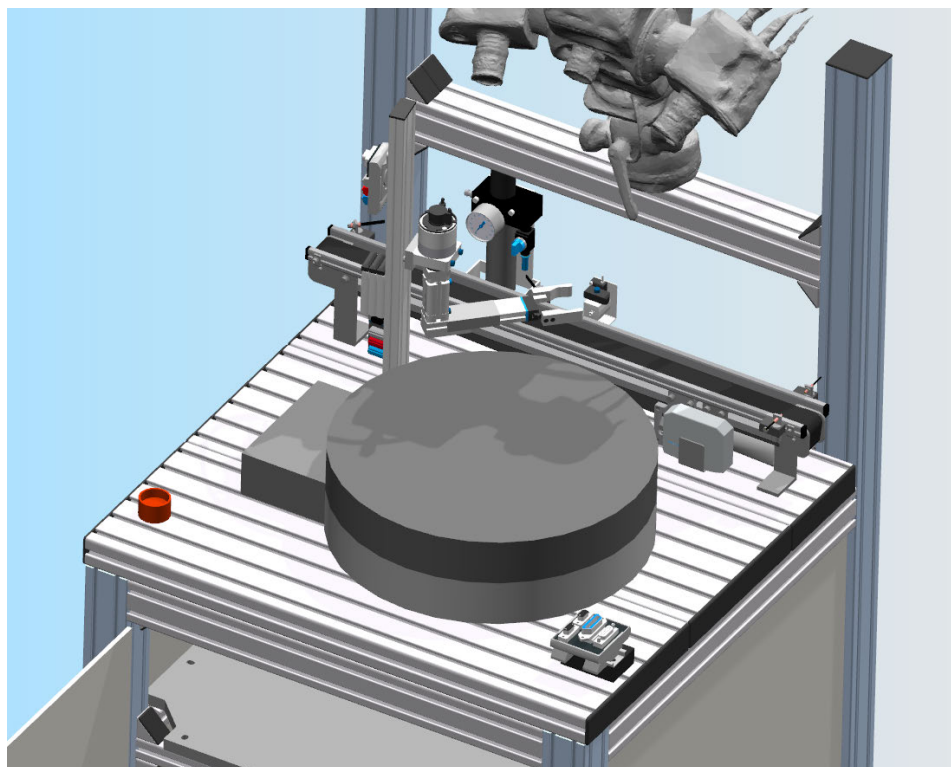


Abbildung 23: Vermessungsstation in Ciro mit Drehtisch und Bauteil⁸⁸

⁸⁸ Eigene Darstellung

Wie die Kinematiken und Mechanismen für die Bewegung des Drehtisches und der Transport des Bauteils über das Förderband und des Drehtisches erstellt werden, wird im folgenden Kapitel erläutert. Diese Abhängigkeiten sind essenziell, um die korrekte Funktion und die realitätsnahe Simulation des gesamten Prozesses sicherzustellen.

4.2.2 Definition der Kinematiken, Mechanismen und Steuerungsschnittstellen

Die Definition von Kinematiken, Mechanismen, Greif- und Greiferpunkten sowie Ein- und Ausgängen bildet die Grundlage für eine funktionale Simulation in Ciro. Diese Elemente ermöglichen die Darstellung der Bewegungsabläufe, die Interaktion zwischen den Modulen sowie die Anbindung an die speicherprogrammierbare Steuerung. Dadurch wird die physische Realität der Vermessungsstation im digitalen Zwillings präzise abgebildet.

- **Kinematiken** beschreiben die Bewegungsmöglichkeiten eines Moduls, beispielsweise Rotations- oder Translationsbewegungen, und legen deren Grenzen fest. Sie sind essenziell, um den realistischen Bewegungsablauf der Module zu simulieren.
- **Mechanismen** definieren die Verhaltensweise eines Objekts oder Moduls. Sie repräsentieren Komponenten wie Zylinder, Schrittmotoren oder Servomotoren, die für die Bewegungsausführung notwendig sind.
- **Greifpunkte** und **Greiferpunkte** werden genutzt, um das Bauteil in der Simulation korrekt zu transportieren und zu positionieren. Sie stellen sicher, dass das Bauteil von einer Station übernommen und an die nächste übergeben wird.
- **Ein- und Ausgänge** verbinden die Module untereinander sowie mit der SPS, um die Steuerungs- und Regelungsfunktionen zu gewährleisten. Diese Schnittstellen dienen zur Signalübertragung und ermöglichen die Simulation der Verdrahtung.

Erstellung und Anpassung von Kinematiken

Für jedes Modul, das Bewegungen ausführen soll, wird eine Kinematik erstellt. Diese wird über den Modell-Explorer durch Rechtsklick auf das „Modul > Neu > Gruppe“ und Auswahl der Gruppe mit „Bearbeiten > Kinematik“ hinzugefügt. Die Kinematik wird anschließend im Eigenschaft-Menü unter dem Reiter „Achsen“ konfiguriert. Dort können die Bewegungsart (z. B. Rotation oder Translation) sowie die Bewegungsgrenzen definiert werden.

Beispiel: Dreh-Hub-Modul

Das Dreh-Hub-Modul besitzt bereits eine vorgefertigte Kinematik, da sie aus einer bestehenden Festo-Station kopiert wird. Die Bewegungsgrenzen der Kinematik muss lediglich für die Vermessungsstation angepasst werden. Der Schwenkantrieb des Greifers ermöglicht Bewegungen zwischen -90° bis $+90^\circ$.

Für die spezifischen Anforderungen werden die Bewegungsgrenzen angepasst auf:

- **Untere Grenze:** +55°
- **Obere Grenze:** +142°

Diese Anpassungen sorgen dafür, dass der Greifer das Bauteil präzise von Förderband und Drehtisch aufnehmen und wieder ablegen kann (Abbildung 24).

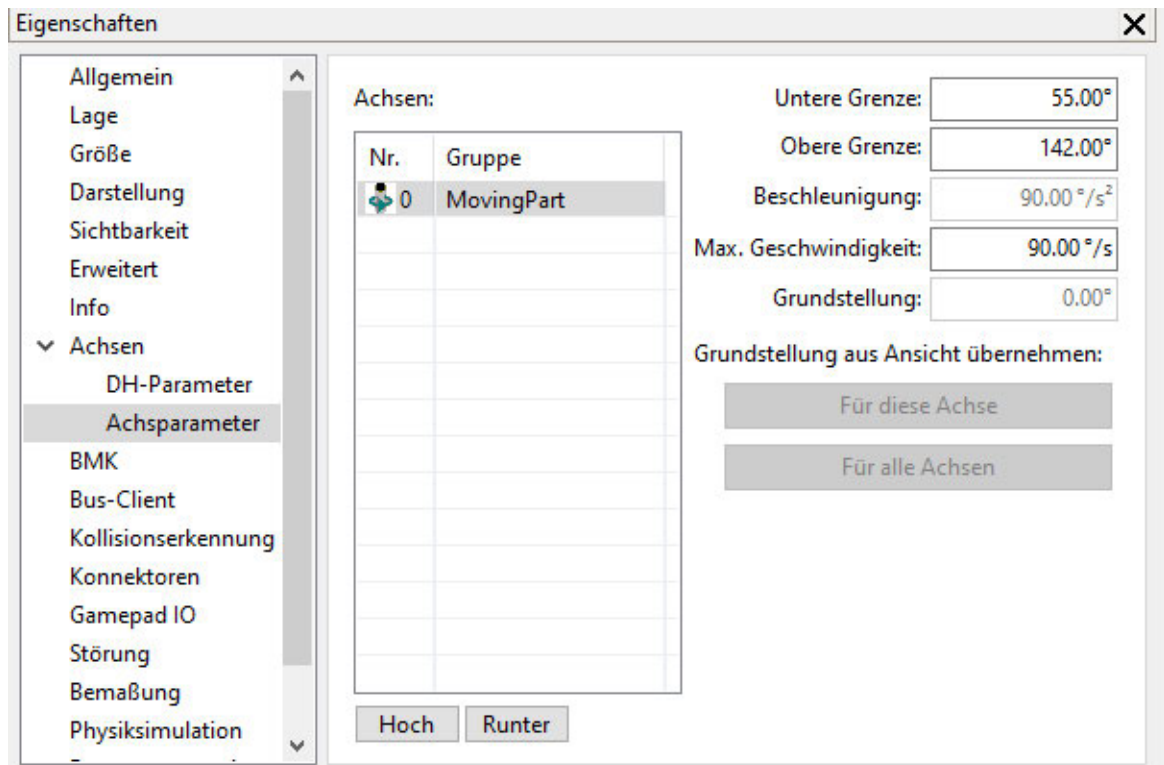


Abbildung 24: Achsparameter des Dreh-Hub-Moduls in Ciro⁸⁹

Erstellung und Anpassung von Mechanismen

Mechanismen in Ciro dienen dazu, die Funktionalität eines Moduls zu simulieren und zu beschreiben. Sie ermöglichen die Interaktion von Komponenten innerhalb der Simulation und sind essenziell, um komplexe Bewegungs- und Ablaufprozesse darzustellen.

Ein Mechanismus wird durch die Umwandlung des Objekttyps eines Objekts definiert. Dies kann über das Eigenschaften-Menü im Reiter „Allgemein“ erfolgen, wo der Objekttyp in einen Mechanismus geändert wird. Dabei werden dem Objekt spezifische Parameter und Funktionen zugewiesen, die seine Bewegung, Verknüpfung oder Interaktion mit anderen Objekten steuern.

Beispiel: Drehtisch

1. Dem Objekt Drehtisch wird eine neue Kinematik hinzugefügt und in „Turn“ umbenannt.
2. Die Kinematik wird in den Eigenschaften bei Allgemein auf den Objekttyp „Mechanismus“ geändert.

⁸⁹ Eigene Darstellung

3. Im Eigenschaften-Menü erscheint der Eintrag „Mechanismen“, indem der Mechanismus auf Servomotor gesetzt wird.
4. Die Ein- und Ausgänge des Drehtisches werden unter „Wert“ dem Servomotor zugeordnet (siehe Abbildung 25), z. B.:
 - Digitaler Eingang „turn_an_digital“ – Startsignal für die Drehbewegung.
 - Analoger Ausgang „turn_IstPosition“ – Rückmeldung der aktuellen Position.

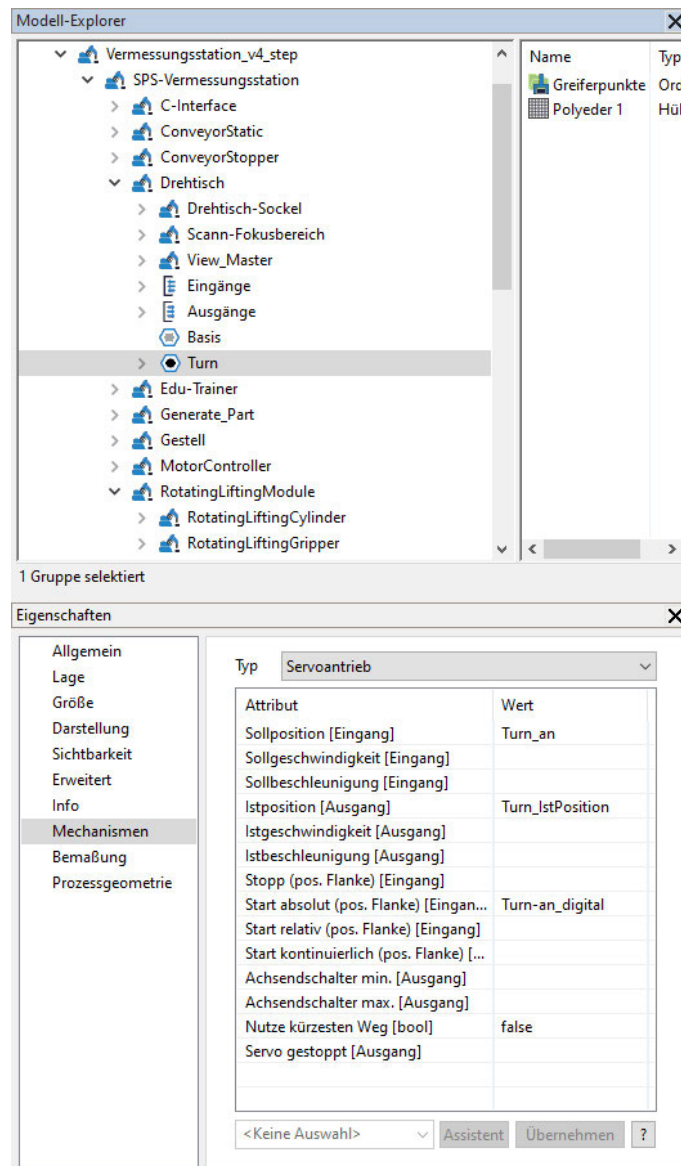


Abbildung 25: Mechanismus des Drehtisches in Ciro⁹⁰

Erstellung von Greif- und Greiferpunkten

⁹⁰ Eigene Darstellung

Greifpunkte und Greiferpunkte definieren die Schnittstellen, über die Objekte in der Simulation korrekt gegriffen, transportiert und positioniert werden. Diese Punkte können einem Objekt hinzugefügt werden, indem man mit einem Rechtsklick auf die Kinematik-Gruppe des Objekts klickt und anschließend „*Neu > Greifpunkte/Greiferpunkte*“ auswählt.

Beispiel: Bauteil

Zwei Greifpunkte werden am Bauteil erstellt:

1. Greifpunkt in der Mitte des Zylinders: Ermöglicht das Greifen des Bauteils durch den Greifer.
2. Greifpunkt am Zylinderboden: Dient dem Transport auf Förderband und Drehtisch.

Für die Stationen wie Förderband, Dreh-Hub-Modul und Drehtisch werden Greiferpunkte hinzugefügt. Diese greifen das Bauteil automatisch, sobald die Greifer- und Greifpunkte aufeinandertreffen.

Die Priorisierung der Greiferpunkte ist entscheidend und kann in der Einstellung des jeweiligen Greiferpunktes unter „Allgemein“ eingestellt werden (siehe Abbildung 26.). Höhere Priorität (z. B. 1.0) ermöglicht, dass ein Greifpunkt das Bauteil von einem niedrigeren Prioritätspunkt (z. B. 0.7) übernimmt. Dies gewährleistet den reibungslosen Übergang zwischen den Modulen.

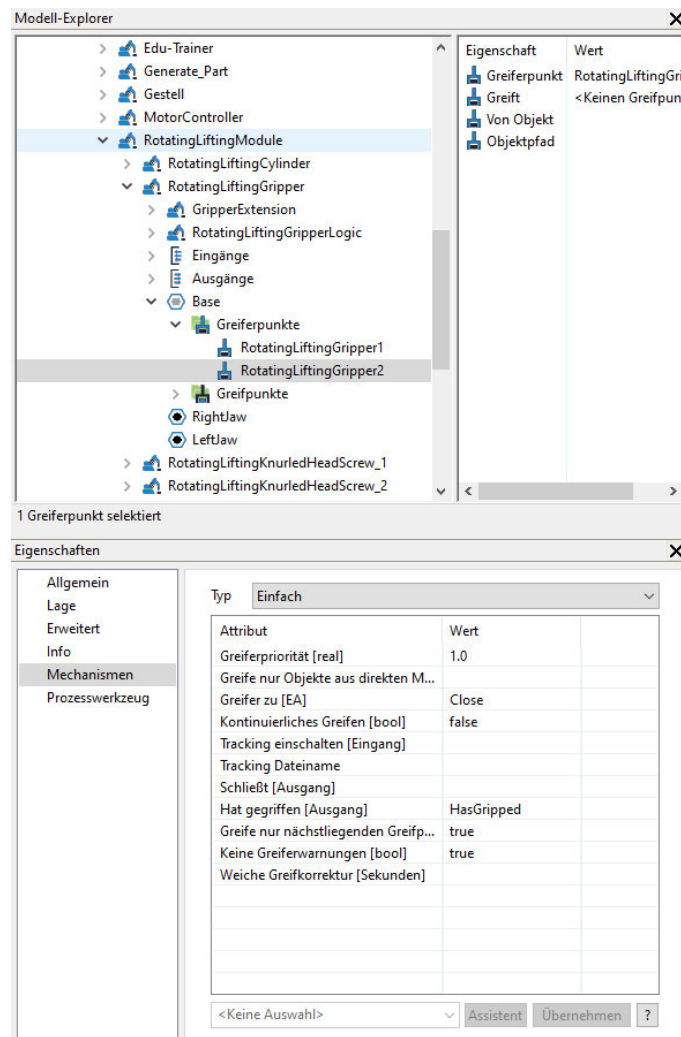


Abbildung 26: Greiferpriorität eines Greifers in Ciro⁹¹

Erstellung und Anpassung der Objektquelle und -senke

Die Objektquelle und die Objektsenke sind wesentliche Bestandteile einer Simulation in Ciro, um den Fluss von Bauteilen innerhalb eines Prozesses zu steuern. Die Objektquelle dient dazu, Bauteile an einem definierten Startpunkt zu erzeugen und in den Prozess einzuspeisen. Die Objektsenke hingegen markiert den Endpunkt eines Bauteils im Prozess und entfernt es aus der Simulation, sobald es den definierten Bereich der Senke erreicht. Diese Funktionalitäten ermöglichen es, den Materialfluss innerhalb der Simulation realitätsgetreu abzubilden.

Beispiel: Bauteil

1. Erstellung der Objektquelle:

Über den Modell-Explorer wird das Bauteil in eine Objektquelle umgewandelt. Dies erfolgt durch einen Rechtsklick auf das Bauteil und Auswahl von „*Bearbeiten > Objektquelle erzeugen*“. Dadurch wird das Bauteil aus den regulären Objekten in die Gruppe der Templates kopiert, und ein neues Objekt namens Objektquelle erscheint im Modell-Explorer.

2. Positionierung der Objektquelle:

Die Position der Quelle wird im Menü „*Eigenschaften > Lage*“ so angepasst, dass das

⁹¹ Eigene Darstellung

Bauteil am Bandanfang erzeugt wird. Diese Positionierung sorgt dafür, dass das Bauteil korrekt in den Materialfluss der Simulation eingespeist wird.

3. Verknüpfung der Signale:

Die Ein- und Ausgänge der Objektquelle werden mit den entsprechenden Signalen des Systems, wie dem Bandanfang, verknüpft. Diese Signale steuern, wann ein neues Bauteil erzeugt wird, und sorgen für die korrekte Synchronisierung mit den anderen Modulen.

4. Erstellung der Objektsenke:

Am Bandende wird eine Objektsenke über den Pfad „Modellierung > Modellbibliothek > Senke“ eingefügt. Dieses Modul ist dafür zuständig, Bauteile aus der Simulation zu entfernen, sobald sie den Bereich der Senke betreten.

5. Anpassung der Größe und Position der Senke:

Die Größe und Lage der Senke werden so eingestellt, dass sie das gesamte Bandende abdeckt. Dadurch wird sichergestellt, dass jedes Bauteil, das den Endbereich des Förderbands erreicht, zuverlässig entfernt wird.

Durch die Einrichtung der Objektquelle und -senke wird ein geschlossener Materialfluss in der Simulation gewährleistet. Dies bildet die Grundlage für die Überprüfung und Optimierung des gesamten Prozesses innerhalb der Simulation.

Erstellung der Ein- und Ausgänge

Die Ein- und Ausgänge eines Moduls werden manuell erstellt, um die Verbindung zur SPS sowie zwischen den Modulen herzustellen. Dies umfasst sowohl digitale Signale (z. B. Startbefehle) als auch analoge Signale (z. B. Positionsrückmeldungen).

Ein- und Ausgänge können hinzugefügt werden, indem im Modell-Explorer ein Rechtsklick auf das jeweilige Objekt gemacht wird und anschließend „Neu > Ein-/Ausgang“ ausgewählt wird. Dadurch lassen sich die benötigten Schnittstellen einfach und strukturiert anlegen.

Die folgender Tabelle 1 zeigt die Ein- und Ausgänge der Module sowie deren Funktion in der Simulation:

Tabelle 1: Eingänge und Ausgänge der Module in Ciro⁹²

E/A	analog/digital	Datentyp	Bezeichnung
Eingang	digital	BOOL	Sensor_FoerderbandAnfang
Eingang	digital	BOOL	Sensor_FoerderbandPos_2
Eingang	digital	BOOL	Sensor_Stopper_ausgefahren
Eingang	digital	BOOL	Sensor_Dreh-Hub-Senken_ingefahren
Eingang	digital	BOOL	Sensor_Dreh-Hub-Senken_ausgefahren
Eingang	digital	BOOL	Sensor_Dreh-Hub-Greifen
Eingang	digital	BOOL	Sensor_Dreh-Hub-Drehen_ingefahren
Eingang	digital	BOOL	Sensor_Dreh-Hub-Drehen_ausgefahren
Eingang	digital	BOOL	Lichtschranke_FoerderbandEnde
Eingang	digital	BOOL	Sensor_Drehtisch

⁹² Eigene Darstellung

Eingang	analog	FLOAT	Sensor_Drehtisch_IstPosition
Ausgang	digital	BOOL	Motor_Foederband
Ausgang	digital	BOOL	Ventil_Stopper_ausfahren
Ausgang	digital	BOOL	Drehrichtung_Drehtisch
Ausgang	digital	BOOL	Ventil_Dreh-Hub-Senken
Ausgang	digital	BOOL	Ventil_Dreh-Hub-Heben
Ausgang	digital	BOOL	Ventil_Dreh-Hub-Greifen
Ausgang	digital	BOOL	Ventil_Dreh-Hub- DrehenRechts
Ausgang	digital	BOOL	Ventil_Dreh-Hub- DrehenLinks
Ausgang	digital	BOOL	Motor_Drehtisch_an
Ausgang	analog	FLOAT	Motor_Drehtisch_SollPosition

Die Definition und Anpassung der Kinematiken, Mechanismen, Greif- und Greiferpunkte sowie der Ein- und Ausgänge bildet die Grundlage für die Simulation der Vermessungsstation. Die Konfiguration ermöglicht eine realitätsnahe Darstellung der Module und deren Interaktion mit der SPS. Dies unterstützt die Validierung des Steuerungskonzepts und sorgt für eine optimale Integration in die physische Anlage.

4.2.3 SPS-Programmierung im TIA-Portal und Integration in Ciro

Bevor die eigentliche Programmierung der SPS im TIA-Portal beginnt, ist es essenziell, die Steuerungslogik detailliert zu durchdenken und grafisch darzustellen. Eine präzise Vorbereitung vereinfacht den Programmierprozess erheblich und minimiert potenzielle Fehlerquellen.

Da es sich bei dem Vermessungsprozess um ein sequenzielles, schrittweises Ablaufschema handelt, ist die Programmiersprache FUP (Funktionsplan) besonders geeignet. Diese Sprache ermöglicht eine intuitive grafische Darstellung und bietet insbesondere Einsteigern im Bereich der SPS-Programmierung einen einfacheren Einstieg.

Die logische Abfolge der Schritte, Bedingungen (Transitionen) und Aktionen kann mittels eines Grafcet-Diagramms effektiv dargestellt werden. Grafcet ist eine Methode zur Modellierung von Ablaufsteuerungen und bietet eine klare und standardisierte Darstellung, die sowohl die Übersichtlichkeit als auch die spätere Implementierung erleichtert.

Erstellung eines Grafcets mit Grafcet-Studio

Zur Erstellung der grafischen Darstellung wird die Software Grafcet-Studio verwendet. Hierbei sind die folgenden Schritte auszuführen:

1. **Anlegen der Symboltabelle** (siehe Abbildung 27):
 - Alle Sensoren und Aktoren werden in einer Symboltabelle definiert.
 - Die Symboltabelle enthält:
 - **Gruppe:** Zuordnung als Sensor oder Aktor.
 - **Datentyp:** Bool (Schalter), Float (Messwerte) oder Int16 (Zahlenwerte).

- **Symbol:** Klarname der Komponente, z. B. „Start_Sensor“.
- **Kommentar:** Beschreibung der Funktion der Komponente.
- Die Daten für die Symboltabelle werden den technischen Datenblättern der Module entnommen.^{93 94}

Symbole									
Filter									
+ Einfügen X Löschen ▼									
	Gruppe:	Typ	Symbol	Kommentar	Adresse	DB-Adresse	Flags	Verwendet	
	Sensor	BOOL	I0	Sensor1 - FörderbandAnfang	I0.0	D0.0		2 (2+0)	
	Sensor	BOOL	I1	Sensor2 - FörderbandPos.2	I0.1	D0.1		7 (7+0)	
	Sensor	BOOL	I2	Stopper Sensor ausgefahren	I0.2	D0.2		2 (2+0)	
	Sensor	BOOL	I3	Dreh-Hub-Sensor Senken: eingefahren (0mm)	I0.3	D0.3		9 (9+0)	
	Sensor	BOOL	I4	Dreh-Hub-Sensor Senken: ausgefahren (+20mm)	I0.4	D0.4		8 (8+0)	
	Sensor	BOOL	I5	Dreh-Hub-Sensor Greifen	I0.5	D0.5		14 (14+0)	
	Sensor	BOOL	I6	Dreh-Hub-Sensor Drehen: eingefahren (55°)	I0.6	D0.6		3 (3+0)	
	Sensor	BOOL	I7	Dreh-Hub-Sensor Drehen: ausgefahren (142°)	I0.7	D0.7		3 (3+0)	
	Sensor	FLOAT	I8	Sensor - Drehtisch analog	I256-I259	D50.0		4 (4+0)	
	Sensor	BOOL	I9	Lichtschranke - FörderbandEnde	I1.0	D1.0		3 (3+0)	
	Sensor	BOOL	I15		I1.1	D1.1		0	
	Aktor	BOOL	Q1	Motor Förderband ein/aus	Q0.0	D25.0		4 (0+4)	
	Aktor	BOOL	Q2	Ventil Stopper ein/aus	Q0.1	D25.1		2 (0+2)	
	Aktor	FLOAT	Q3_Drehen_analog	Motor Drehtisch analog	Q256-Q259	D100.0		2 (0+2)	
	Aktor	BOOL	Q3_Drehen	Motor Drehtisch digital ein/aus	Q0.2	D25.2		3 (0+3)	
	Aktor	BOOL	Q3_Richtung	Drehrichtung-Motor_Drehtisch	Q0.3	D25.3		2 (0+2)	
	Aktor	BOOL	Q4_Senken	Ventil Dreh-Hub-Modul Senken ein	Q0.4	D25.4		4 (0+4)	
	Aktor	BOOL	Q4_Heben	Ventil Dreh-Hub-Modul Heben ein	Q0.5	D25.5		4 (0+4)	
	Aktor	BOOL	Q4_Greifen	Ventil Dreh-Hub-Modul Greifer zu/offen (:=1/0)	Q0.6	D25.6		4 (0+4)	
	Aktor	INT16	Q4_DrehenRechts	Ventil Dreh-Hub-Modul Drehen ein (1=-55°)	Q260-Q261	D104.0		2 (0+2)	
	Aktor	INT16	Q4_DrehenLinks	Ventil Dreh-Hub-Modul Drehen aus (1=90°)	Q262-Q263	D106.0		2 (0+2)	
	Data	UINT32	Counter		-	D175.0		10 (3+7)	

Abbildung 27: Grafcet-Symboltabelle - digitaler Zwilling⁹⁵

2. Grundelemente des Grafkets:

- Ein Grafket besteht aus drei zentralen Elementen:
 - **Schritt:** Ein definierter Zustand (z. B. „Schritt 1 im Ablauf“).
 - **Transition:** Eine Bedingung, die erfüllt sein muss, um von einem Schritt zum nächsten überzugehen. (z. B. „Bauteil erkannt durch Sensor“).
 - **Aktion:** Eine konkrete Steuerungsanweisung, die während eines Schrittes ausgeführt wird (z. B. „Förderband starten“).

In Abbildung 28 sind einige der genormten Symbole (DIN EN 60848) eines Grafkets dargestellt.⁹⁶

⁹³ vgl. o. V.: Modul Band - DataSheet.pdf.

⁹⁴ vgl. o. V.: Dreh-Hub-Modul - DataSheet.pdf.

⁹⁵ Eigene Darstellung

⁹⁶ vgl. o. V.: Grafcet-Studio: Einleitung.



Abbildung 28: Auswahl an wichtigen Grafcet-Symbolen⁹⁷

Das Diagramm wird von oben nach unten aufgebaut und gelesen. Der Initialschritt 0 dient als Startpunkt für den gesamten Prozess. Die logische Abfolge der Schritte wird durch die Transitionen verbunden, die die Bedingungen für den Übergang zwischen den Schritten definieren.

Mit dem Initialschritt 0 beginnt das SPS-Programm der Vermessungsstation. Der Ablauf wird im Grafcet durch aufeinanderfolgende Schritte und Transitionen gesteuert.

Schritt 1 wird aktiviert, sobald die Transition $I0 = \text{True}$ erfüllt ist. Laut der Symboltabelle (Abbildung 27) entspricht $I0$ dem Sensor 1 – „FörderbandAnfang“. Erkennt dieser Sensor ein Bauteil, gibt er ein True-Signal aus, wodurch Schritt 1 aktiviert wird. Dies führt zur Ausführung der Aktionen $Q1:=1$ und $Q2:=1$.

- $Q1$ („Förderband“) = 1 bedeutet, dass das Förderband eingeschaltet wird.
- $Q2$ („Ventil Stopper“) = 1 bewirkt, dass der Stopper Zylinder ausfährt.

Dadurch wird das Bauteil auf dem Förderband transportiert, bis es durch den ausgefahrenen Stopper Zylinder angehalten wird. Die Symbole der Aktionen enthalten Pfeile nach oben, was darauf hinweist, dass diese einmal aktiviert werden und in diesem Zustand verbleiben, bis eine andere Aktion sie auf False ($:= 0$) setzt.

Damit Schritt 2 aktiviert wird, muss die vorhergehende Transition True werden. Dies geschieht, sobald der Sensor 2 – Sensor „FörderbandPos 2“ das Bauteil erkennt. Gleichzeitig wird ein interner Zähler gestartet, der beginnt zu zählen. Der Zähler erhöht sich kontinuierlich, solange der Sensor das Bauteil erfasst.

Sobald der Zähler den Wert 2 Sekunden erreicht, das Bauteil weiterhin am Sensor „FörderbandPos 2“ erkannt wird und der Greifer des Dreh-Hub-Moduls sich in der oberen Endlage befindet ($I3 = \text{True}$), wird Schritt 3 ausgeführt.

⁹⁷ vgl. o. V.: Grafcet-Studio: Grafcet-Norm.

Schritt 3 hat folgende Auswirkungen:

- $Q1:=0$ → Das Förderband wird gestoppt.
- $Q2:=0$ → Der Stopper Zylinder fährt ein.

Damit der Greifarm des Dreh-Hub-Moduls im nächsten Schritt das Bauteil aufnehmen kann, muss sich der Greifarm in der oberen Endlage befinden und der Stopper Zylinder muss sich vollständig eingefahren haben (negiertes Zeichen über I2 bedeutet, dass I2 „False“ sein muss).

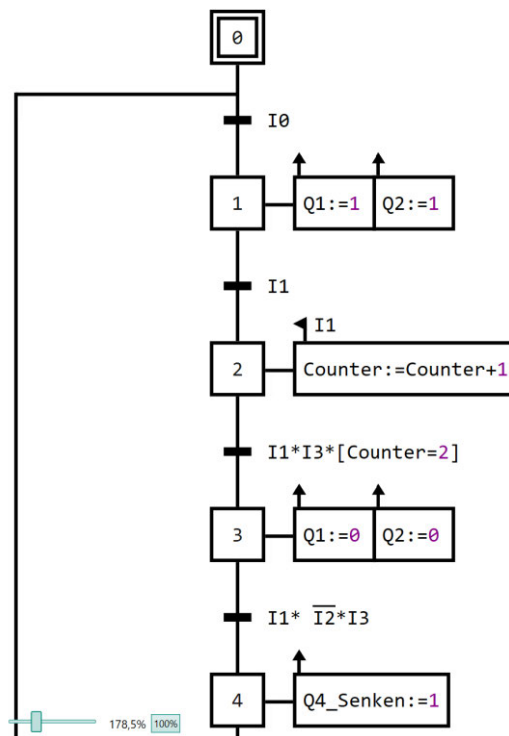


Abbildung 29: Ausschnitt Grafcet - digitaler Zwilling⁹⁸

Diese Erklärung beschreibt die Funktionsweise des Grafcet-Ausschnitts in Abbildung 29. Das vollständige Grafcet für den gesamten Prozessablauf ist in Anhang B: *Grafcet - digitaler Zwilling* dargestellt.

Umsetzung der Steuerung in TIA Portal

Nachdem der Ablauf der Steuerung über den Grafcet beschrieben wurde, wird dieser nun im TIA Portal V17 als Funktionsplan (FUP) umgesetzt. Dafür wird zunächst ein neues Projekt mit dem Namen „Vermessungsstation“ erstellt.

Als erstes wird die Hardwarekonfiguration der speicherprogrammierbaren Steuerung angelegt. Dafür wird im Projektbaum unter „Geräte & Netze > Neues Gerät hinzufügen“ eine neue CPU 1516-3 PN/DP aus der Bibliothek ausgewählt und in das Projekt geladen.

⁹⁸ Eigene Darstellung

Jede CPU verfügt über unterschiedliche Ein- und Ausgangsmodule (E/A-Module), die manuell hinzugefügt werden müssen. In diesem Fall werden digitale und analoge Ein- und Ausgangsbausteine in das Rack eingefügt, sodass die SPS über die benötigte Anzahl an E/A-Kanälen verfügt, um alle Signale aus dem digitalen Zwilling zu verarbeiten. Abbildung 30 zeigt die vollständige Hardwarekonfiguration.

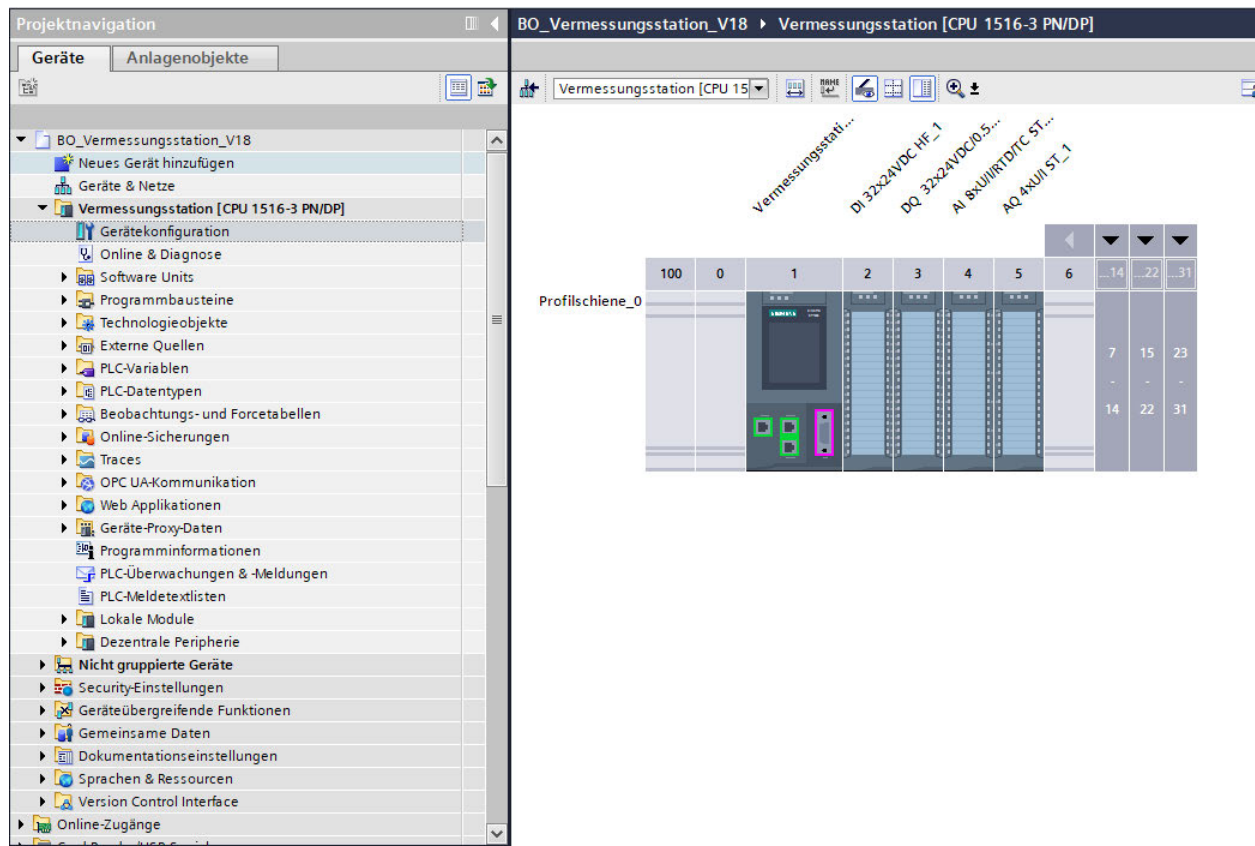


Abbildung 30: Tia Portal - Hardwarekonfiguration⁹⁹

Nach der Hardwarekonfiguration werden die Eingänge, Ausgänge und Variablen der SPS in die PLC-Variablenliste eingetragen. Dafür wird im Projektbaum unter „PLC_1 > PLC-Variablen“ eine neue Symboltabelle erstellt.

Hier werden die Ein- und Ausgangssignale entsprechend der Symboltabelle aus dem Grafnet (Abbildung 27) in der gleichen Reihenfolge angelegt. Dabei ist auf eine eindeutige Bezeichnung, den korrekten Datentyp sowie eine Kommentierung der Variable zu achten, damit die Funktionsweise der Variablen klar ersichtlich ist. Die Adressierung der Variablen wird vom TIA Portal automatisch vergeben, wenn die Einträge in der korrekten Reihenfolge erfolgen. In Abbildung 31 sind die erstellten PLC-Variablen zu sehen.

⁹⁹ Eigene Darstellung

Entwicklung des digitalen Zwillings in Ciros

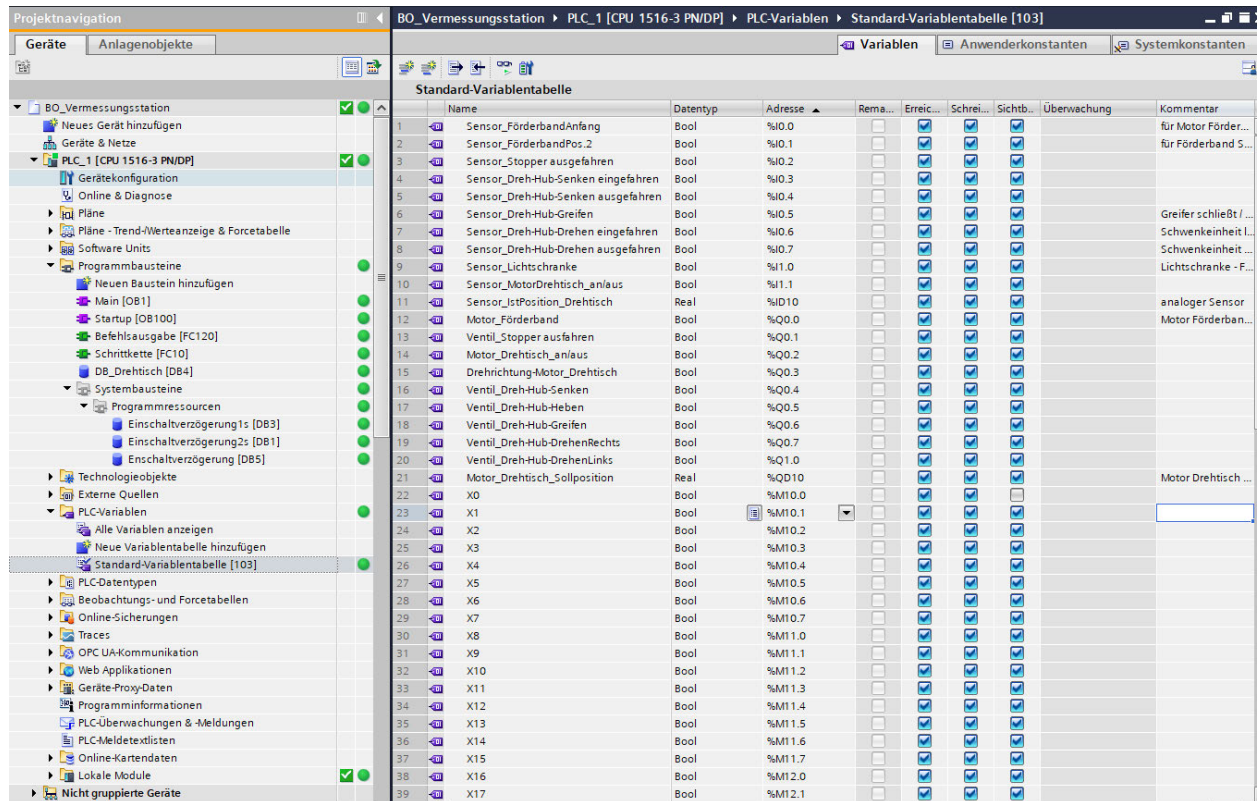


Abbildung 31: digitaler Zwilling - Ausschnitt PLC-Variablen-Tabelle¹⁰⁰

In der folgenden Tabelle 2 sind die unterschiedlichen Adressensymbole erläutert.

Tabelle 2: Tia-Portal - Adressenbeschreibung

Adresse	Datentyp	Bedeutung
%I	Bool	digitaler Eingang: Sensor meldet 1 oder 0 zurück – true oder false
%ID	Real	analoger Eingang: Sensor meldet Zahlenwert zurück wie z. B. Motordrehzahl
%Q	Bool	digitaler Ausgang: Aktor wird auf 1 oder 0 gesetzt – true oder false
%QD	Real	analoger Ausgang: Aktor wird auf Zahlenwert gesetzt wie z. B. Motordrehzahl
%M	Bool	Schritt im Ablauf der SPS-Steuerung

Erstellung des SPS-Programms in TIA Portal

¹⁰⁰ Eigene Darstellung

Die Umsetzung der Steuerung im TIA Portal V17 erfolgt mithilfe einer Ablaufkette, die in Funktionsbausteinen (FCs) programmiert wird. Dabei werden zwei zentrale Funktionsbausteine erstellt:

- **Schritt看ette (FC10):** Dieser Baustein beschreibt die einzelnen Schritte der Ablaufsteuerung sowie die Bedingungen, unter denen diese aktiviert oder deaktiviert werden.
- **Befehlsausgabe (FC120):** Hier werden die Aktoren definiert, also welche Befehle ausgeführt werden, wenn ein Schritt aktiv ist.

Die Umsetzung einer Ablaufsteuerung folgt den „goldenen Regeln“ für Schrittketten:

- „Ein Schritt wird aktiviert, wenn der vorhergehende Schritt aktiv ist und gleichzeitig die vorhergehende Transitionsbedingung erfüllt ist.“
- „Ein Schritt wird deaktiviert, wenn der nachfolgende Schritt aktiviert wird.“¹⁰¹

Diese Regeln stellen sicher, dass die Steuerung deterministisch und nachvollziehbar arbeitet.

Zunächst wird im Projektbaum unter „Programmbausteine“ der Funktionsbaustein Schritt看ette (FC10) erstellt. In diesem werden die einzelnen Schritte der Steuerung als Schrittkettenlogik umgesetzt.

Jeder Schritt wird durch ein Set/Reset (SR)-Baustein realisiert, sodass ein Schritt gesetzt wird, wenn die Transitionsbedingungen erfüllt sind und der vorherige Schritt aktiv ist.

In Abbildung 32 ist ein Ausschnitt der Schritt看ette mit den ersten beiden Netzwerken zu sehen.

¹⁰¹ vgl. o. V.: Programmierung Ablaufsteuerung- YouTube.

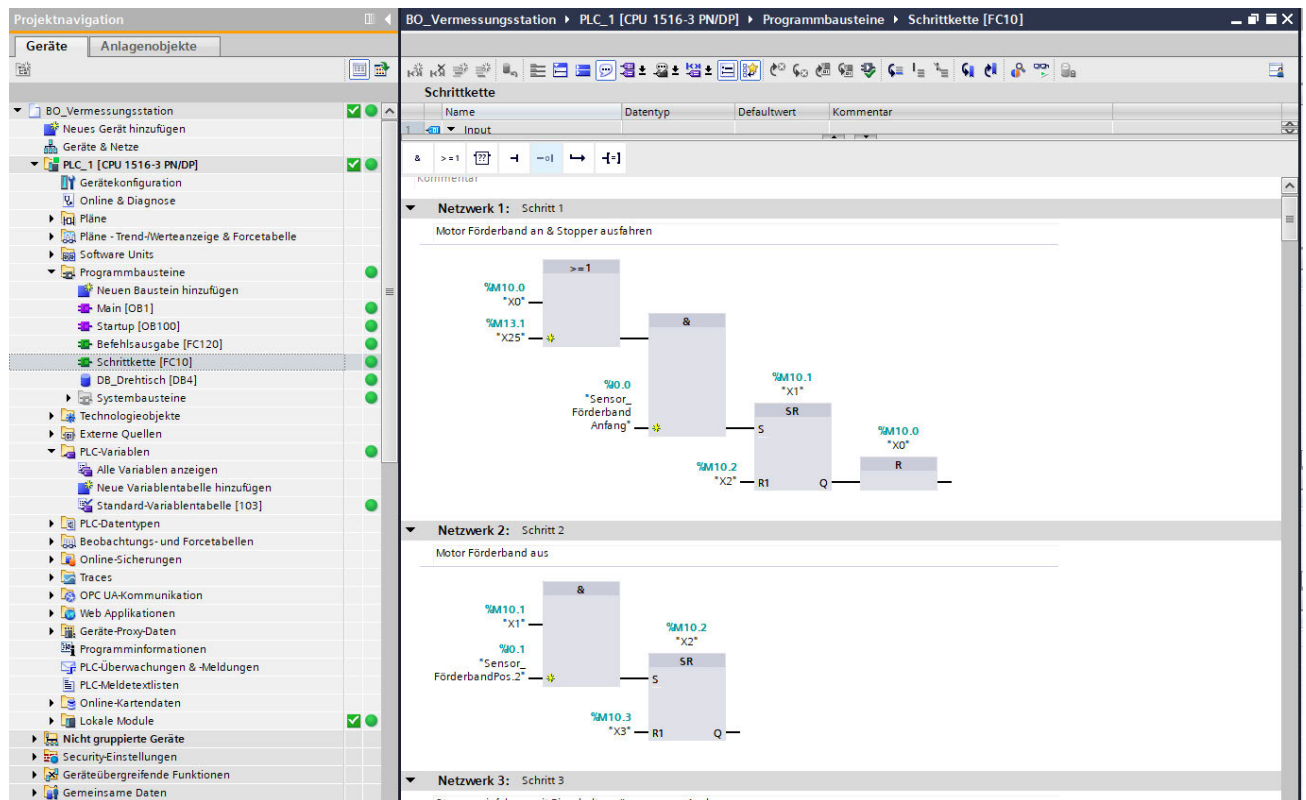


Abbildung 32: Tia Portal - Ausschnitt Funktionsbaustein "Schritt_kette"¹⁰²

Schritt 1 wird aktiviert, wenn entweder X0 (Initialschritt) oder X25 (letzter Schritt der Kette) aktiv ist und gleichzeitig der Sensor „FörderbandAnfang“ = True zurückgibt. Eine Besonderheit ist hier, dass mit dem Schritt 25 zur Aktivierung von X1 sichergestellt wird, dass das Programm nach Ablauf wieder von vorne starten kann. Zudem wird mit dem Setzen von X1 am Ausgang „Q“ X0 zurückgesetzt.

Schritt 2 wird aktiviert, wenn Schritt 1 aktiv ist und die Transitionsbedingung Sensor „FörderbandPos._2“ = True liefert.

Diese Logik setzt sich für die gesamten Schritte der Schritt_kette fort. In einigen Fällen werden Einschaltverzögerungen verwendet, um bestimmte Aktionen zeitlich zu verzögern, bevor der Schritt aktiviert wird, um sicherzustellen, dass der Schritt vollständig ausgeführt wird.

Der Funktionsbaustein „FC120“ (Befehlsausgabe) definiert die Aktoren, also die Ausgangssignale, die durch die Schritte der Ablaufkette gesetzt oder zurückgesetzt werden.

In Abbildung 33 ist das erste und zweite Netzwerk der Befehlsausgabe für das „Förderband“ und das „Ventil Stopper“ zu sehen:

¹⁰² Eigene Darstellung

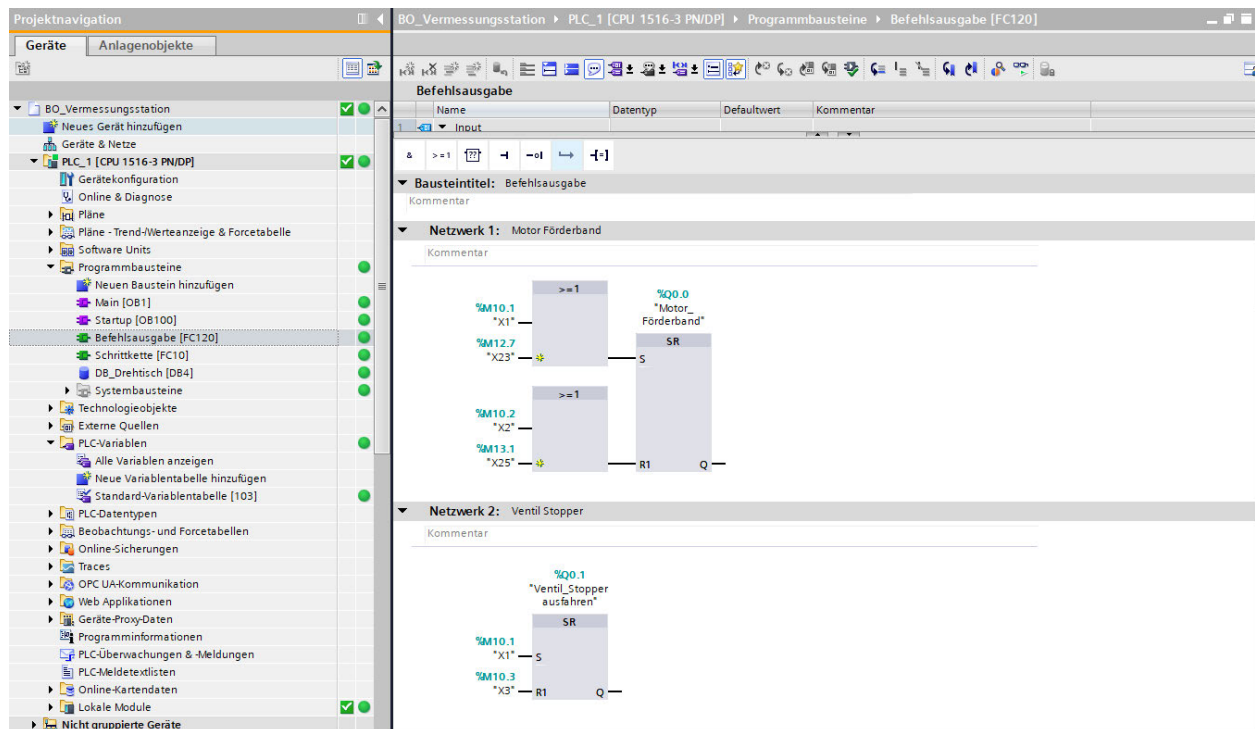


Abbildung 33: Tia Portal - Funktionsbaustein "Befehlsausgabe"¹⁰³

Auch die Aktoren werden in diesem Funktionsbaustein durch SR-Bausteine gesetzt oder zurückgesetzt.

Der Motor des Förderbands (Q0.0) wird zum Beispiel durch Schritt 1 oder Schritt 23 eingeschaltet (auf der Set-Seite). Das Förderband wird gestoppt, sobald Schritt 2 oder Schritt 25 aktiv ist (auf der Reset-Seite). Diese Logik wird mit einem ODER-Baustein realisiert, da mehrere Schritte das Förderband starten oder stoppen können.

Dementsprechend werden alle weiteren Aktoren definiert und durch die jeweiligen Schritte aktiviert oder deaktiviert.

Besondere Logik gilt für den Drehtischmotor, der nicht einfach ein- oder ausgeschaltet wird, sondern über eine analoge Drehzahlsteuerung verfügt.

Die Steuerung des Servomotors erfolgt in Structured Control Language (SCL), da hier eine genaue Drehbewegung (z. B. auf 360° oder zurück auf 0°) programmiert werden muss.

Erstellung der Organisationsbausteine (OBs)

Der Organisationsbaustein OB100 ist für die Initialisierung der Steuerung verantwortlich. In Abbildung 34 ist die Logik zu sehen, mit der Schritt 0 (Initialschritt) gesetzt wird.

¹⁰³ Eigene Darstellung

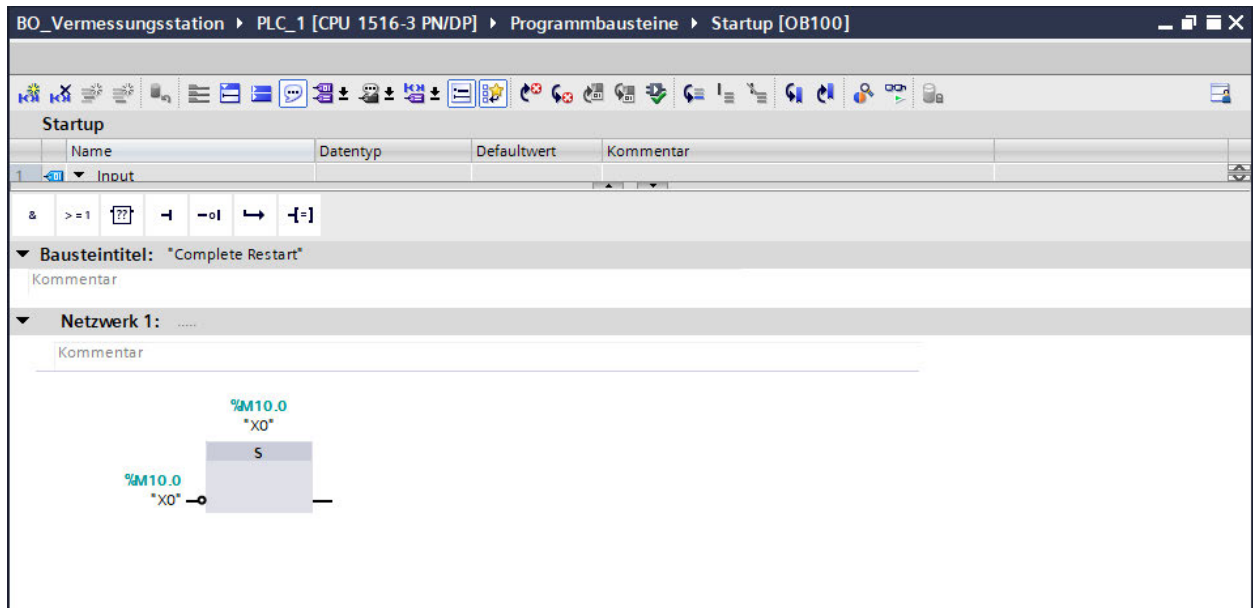


Abbildung 34: Tia Portal - Organisationsbaustein "OB100"¹⁰⁴

Der Initialschritt wird aktiviert, wenn die CPU vom STOP- in den RUN-Modus übergeht. Ein Set-Baustein setzt X0, wobei als Bedingung wiederum X0 negiert ist, da der Initialschritt zu Beginn inaktiv ist. Dieser Baustein stellt sicher, dass die Steuerung nach jedem Neustart der SPS sauber initialisiert wird.

Der Main-OB1 ist der Standard-Organisationsbaustein, der das Hauptprogramm zyklisch ausführt. Er ruft zyklisch die Funktionsbausteine FC10 (Schrittfolge) und FC120 (Befehlsausgabe) auf. Erst nachdem die Startfunktionen in OB100 abgeschlossen sind, wird OB1 ausgeführt. Abbildung 35 zeigt die Konfiguration dieses Bausteins.

¹⁰⁴ Eigene Darstellung

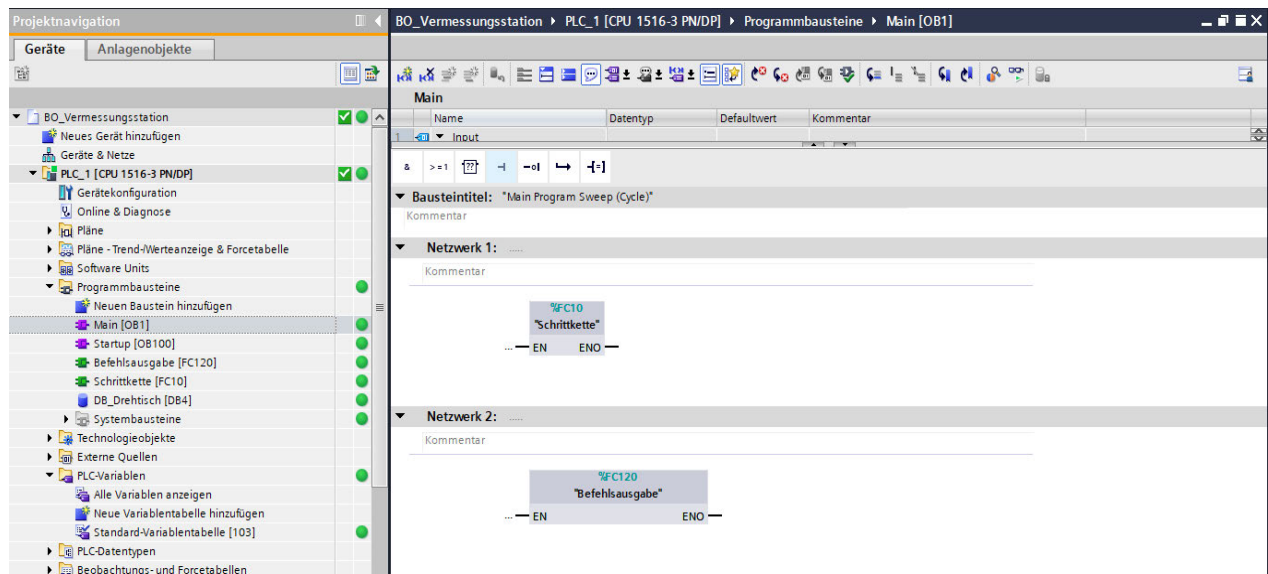


Abbildung 35: Tia Portal - Organisationsbaustein "Main OB1"¹⁰⁵

Implementierung des SPS-Programms in Ciro

Nachdem das SPS-Programm im TIA Portal V17 erstellt wurde, muss es nun in die Simulationsumgebung Ciro integriert werden, um den digitalen Zwilling der Vermessungsstation vollständig zu realisieren. Hierfür wird das SPS-Programm auf eine virtuelle CPU geladen und mit den in Ciro erstellten Ein- und Ausgängen verknüpft.

Zur Simulation der realen Steuerung wird *Siemens* PLCSim Advanced verwendet. Dieses Tool ermöglicht die exakte Nachbildung des Verhaltens der Firmware realer Steuerungen, wodurch das zuvor erstellte SPS-Programm in einer virtuellen Umgebung getestet werden kann.

Die Einrichtung der virtuellen CPU erfolgt in mehreren Schritten:

1. Starten von PLCSim Advanced
 - Nach dem Öffnen von PLCSim Advanced wird eine neue Instanz erstellt.
2. Konfiguration der virtuellen CPU
 - Der Name der Steuerung wird eingegeben („Vermessungsstation“).
 - Die IP-Adresse wird gesetzt („192.168.0.1“), die zuvor in der Hardwarekonfiguration im TIA Portal definiert wurde.
 - Die Instanz der virtuellen CPU wird gestartet.
3. Laden des SPS-Programms in die virtuelle CPU

¹⁰⁵ Eigene Darstellung

- Nach der Übersetzung des SPS-Programms im TIA Portal wird über „Laden in Gerät“ das Programm in die virtuelle CPU geladen.
- Hierfür wird die CPU über „PLCSim“ gesucht und die Instanz mit der passenden IP-Adresse ausgewählt.
- Sobald das Laden abgeschlossen ist, kann sich das TIA Portal mit der virtuellen CPU verbinden („Online verbinden“).

4. Wechsel des CPU-Modus

- Im Online-Modus wird der Status der CPU angezeigt (siehe Abbildung 36).
- Eine gelbe LED signalisiert den STOP-Modus, eine grüne LED den RUN-Modus.
- Der Moduswechsel kann über den Reiter „Testen“ im TIA Portal vorgenommen werden.

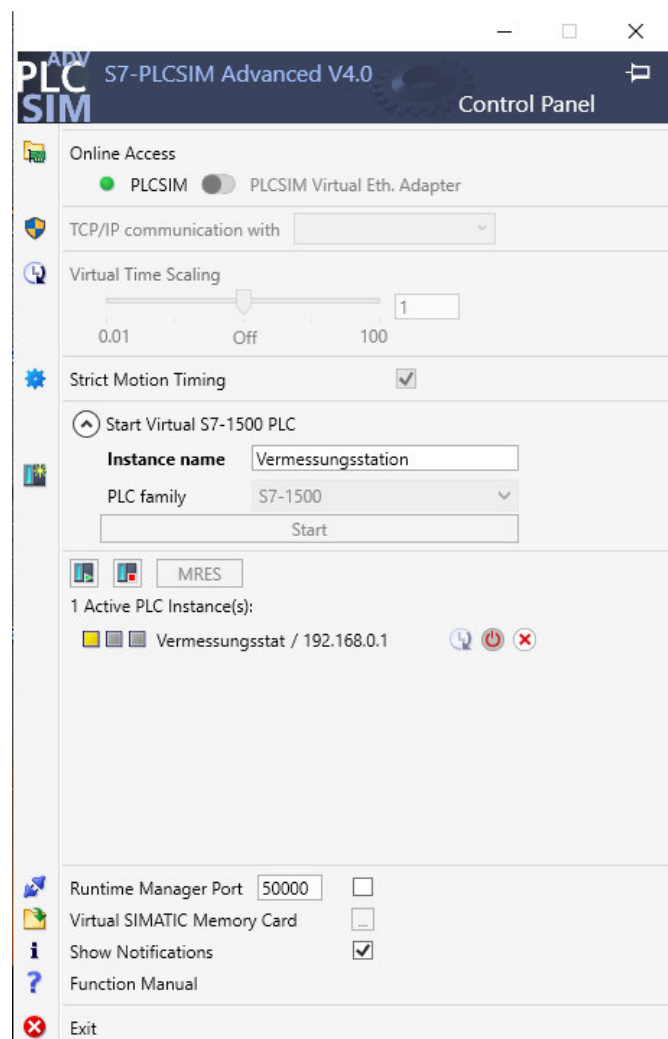


Abbildung 36: PLCSim Advanced - Online Status der Instanz¹⁰⁶

Nun muss die SPS-Steuerung in Ciro eingebunden werden, sodass die erstellten Module und Mechanismen mit den Signalen aus dem SPS-Programm interagieren können. Dies geschieht durch folgende Schritte:

¹⁰⁶ Eigene Darstellung

1. Hinzufügen der SPS-Steuerung in Ciros

- Über die Modellbibliothek wird in Ciros eine SPS-Steuerung hinzugefügt.
- Alle Objekte der Vermessungsstation werden in dieses SPS-Objekt verschoben, sodass sie hierarchisch untergeordnet sind.

2. Verknüpfung mit PLCSim Advanced

- Das neu erstellte SPS-Objekt besitzt in den Eigenschaften den Reiter „S7 PLCSim Advanced“ (siehe Abbildung 37).
- Dort kann die virtuelle CPU über Namen oder IP-Adresse gesucht und ausgewählt werden.

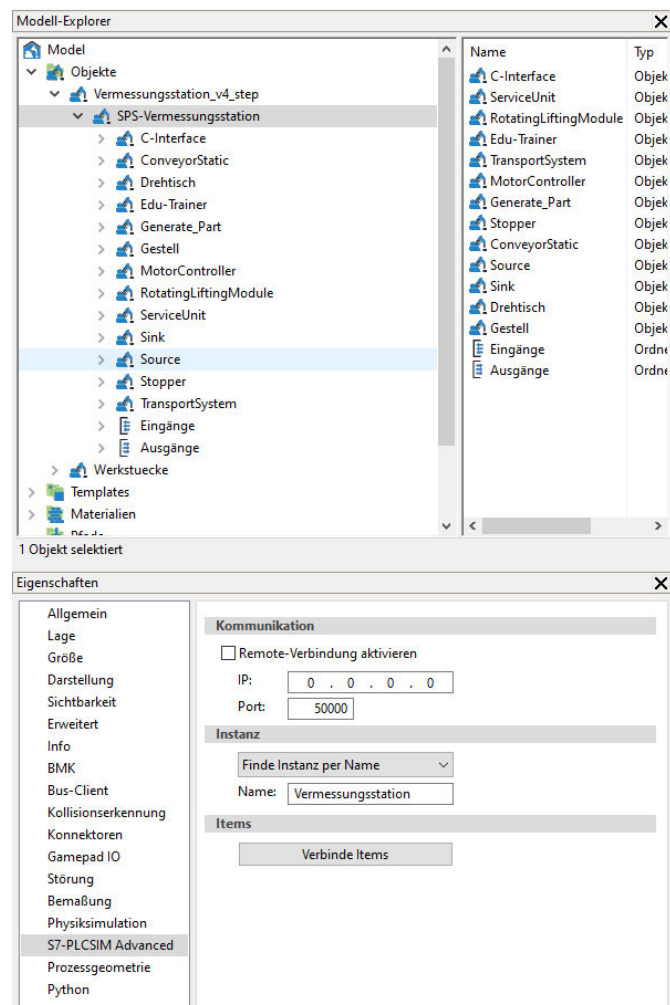


Abbildung 37: Ciros – Eigenschaftenfenster SPS-Konfiguration¹⁰⁷

3. Verbindung der Ein- und Ausgänge mit dem SPS-Programm

- Nach der Auswahl der virtuellen CPU wird der Button „Verbinde Items“ geklickt, woraufhin ein neues Fenster erscheint (siehe Abbildung 38).

¹⁰⁷ Eigene Darstellung

- Über den Button „Verbinden“ wird die direkte Verbindung mit der virtuellen CPU hergestellt.
- Unter dem Reiter „I/O“ werden nun alle Ein- und Ausgänge des SPS-Programms angezeigt.
- Diese werden per Drag & Drop mit den in Ciros erstellten Ein- und Ausgängen verknüpft.
- Durch „Übernehmen“ werden alle Signale gespeichert und synchronisiert.

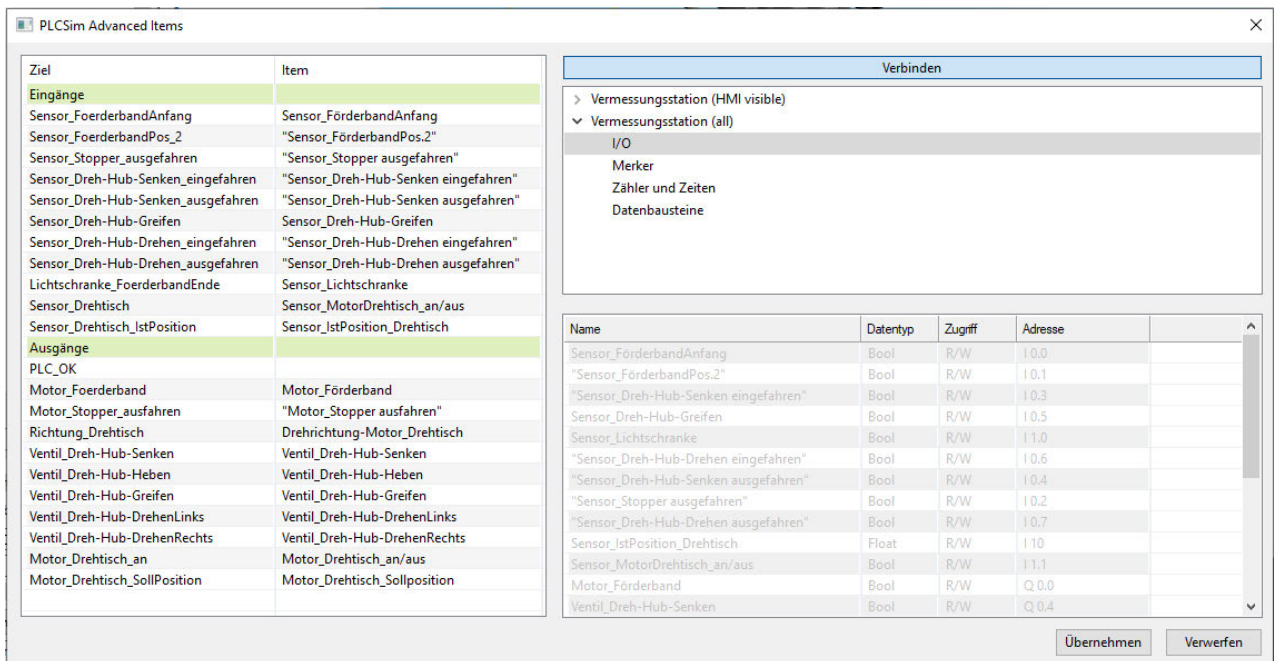


Abbildung 38: Ciros - SPS-Konfiguration der Ein- & Ausgänge¹⁰⁸

Mit der vollständigen Implementierung der SPS-Steuerung in Ciros ist der digitale Zwilling der Vermessungsstation fertiggestellt.

- Die virtuelle CPU steuert nun die digitalen Module in Ciros, genau wie eine reale SPS in der physischen Anlage.
- Alle Bewegungsabläufe, Mechanismen und Sensoren sind mit dem SPS-Programm verknüpft und können getestet werden.
- Dadurch lässt sich die gesamte Steuerungslogik validieren, bevor die reale Anlage in Betrieb genommen wird.

Mit der vollständigen Implementierung der SPS-Steuerung in Ciros ist die digitale Simulation der Vermessungsstation eingerichtet und bereit für die nachfolgende Validierung.

4.3 Test des digitalen Zwillings

¹⁰⁸ Eigene Darstellung

Der Test des digitalen Zwillings stellt einen essenziellen Schritt dar, um die korrekte Funktion der implementierten Steuerungslogik und die Interaktion der virtuellen Komponenten zu überprüfen. Dabei wird die Simulation in Ciros, die virtuelle CPU in PLCSim Advanced sowie das SPS-Programm im TIA Portal gleichzeitig betrieben, um den digitalen Zwilling auf Funktionsfähigkeit zu testen und gegebenenfalls Anpassungen vorzunehmen.

Um eine realitätsnahe Simulation durchführen zu können, müssen vor dem Start des Tests mehrere Programme parallel ausgeführt werden. Dazu gehört Ciros, das für die Simulation der Vermessungsstation verwendet wird, sowie PLCSim Advanced, das die virtuelle CPU bereitstellt. Zusätzlich wird das TIA Portal genutzt, um die SPS-Programmierung zu überwachen und gegebenenfalls Fehler zu beheben. Die Verbindung zwischen Ciros und PLCSim Advanced wurde bereits im vorherigen Kapitel hergestellt, sodass ein Austausch der Ein- und Ausgangssignale gewährleistet ist.

Alle relevanten Dateien für die digitale Simulation, darunter das Ciros-Modell der Vermessungsstation, das SPS-Programm sowie sämtliche CAD-Dateien, sind auf dem Rechner im Raum 105 im IPT unter folgendem Pfad gespeichert:

C:\Desktop\BO Masterarbeit-Ciros & Siemens SPS

Damit während der Simulation gezielt Fehlerbehebungen und Optimierungen vorgenommen werden können, wird das SPS-Programm im TIA Portal geöffnet und mit der virtuellen CPU verbunden. Die Verbindung wird über die Funktion „Online verbinden“ hergestellt. Sobald die Verbindung erfolgreich aufgebaut ist, zeigt das TIA Portal durch grüne Punkte im Projektbaum an, dass das aktuell geladene SPS-Programm mit dem Programm in der CPU übereinstimmt. Falls dies nicht der Fall ist, muss die aktuelle Version aus dem TIA Portal erneut auf die virtuelle CPU geladen werden, um eine einwandfreie Funktion sicherzustellen.

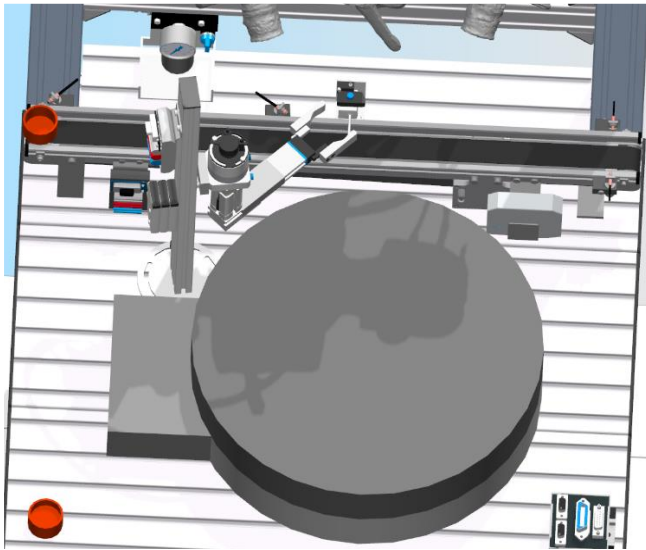
Ein wichtiges Hilfsmittel während der Simulation sind die Beobachtungstabellen im TIA Portal. Diese ermöglichen es, die Ein- und Ausgangssignale der Steuerung in Echtzeit zu überwachen. Dadurch kann analysiert werden, ob die Sensoren die erwarteten Signale liefern und ob die Aktoren korrekt angesteuert werden. Neben den digitalen Signalen, die entweder den Status „True“ oder „False“ annehmen, lassen sich auch analoge Werte der Sensoren überprüfen. Dies ist insbesondere für die präzise Steuerung von Prozessen relevant, bei denen Messwerte kontinuierlich erfasst und verarbeitet werden.

Die Steuerung der virtuellen CPU während des Tests erfolgt über das TIA Portal. Im Reiter „Testen“ kann die CPU entweder in den Run-Modus oder den Stop-Modus versetzt werden. Diese Funktion ist essenziell, um nach Änderungen oder Fehlerbehebungen die Simulation gezielt neu zu starten oder zu pausieren. Dadurch wird sichergestellt, dass Anpassungen unmittelbar überprüft und optimiert werden können, bevor sie final in die physische Steuerung übertragen werden.

Start der Simulation in Ciros

Im nachfolgenden Abschnitt wird der Ablauf des digitalen Zwillings anhand einer Bilderreihe dargestellt. Die Bilder zeigen die einzelnen Prozessschritte der Vermessungsstation in chronologischer Reihenfolge. Neben den Bildern befindet sich jeweils eine kurze Beschreibung, die die wichtigsten Abläufe und Interaktionen der Komponenten erläutert. Diese visuelle

Darstellung ermöglicht eine anschauliche Nachverfolgung der Simulation und hilft dabei, den gesamten Prozessablauf verständlich zu machen.



Mit dem Klick auf den roten Zylinder in der linken unteren Ecke wird ein zweiter Zylinder am Förderbandanfang erzeugt. Der Sensor 1 erkennt das Bauteil - damit fördert das Band das Bauteil und der Stopper fährt aus..

Das Bauteil kommt an dem Stopper zum stehen.

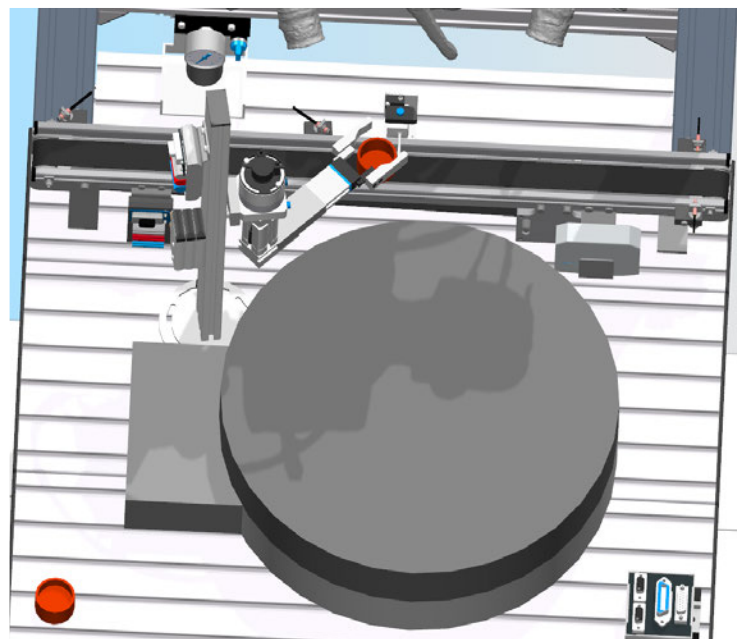
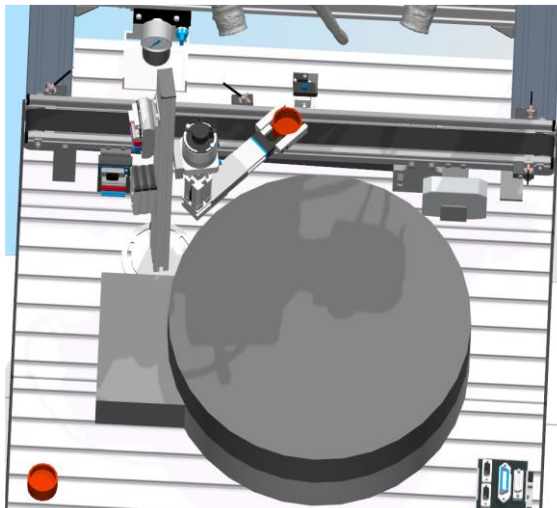


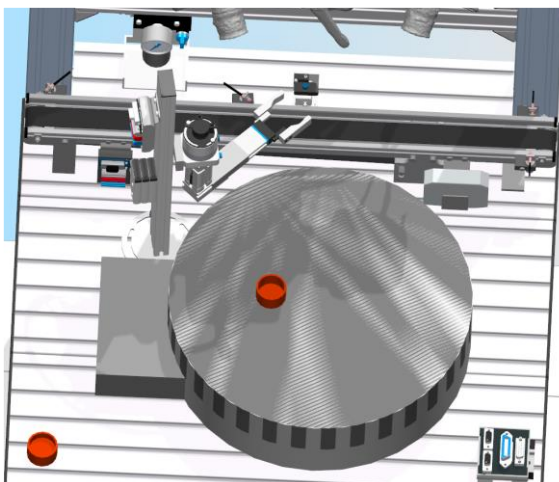
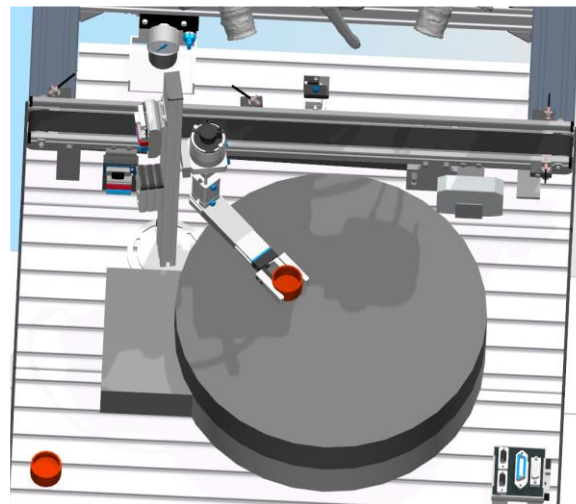
Abbildung 39: Ciro - Simulation, Schritt 1-2 ¹⁰⁹

¹⁰⁹ Eigene Darstellung



Der Schwenkarm mit Greifer des Dreh-Hub-Moduls senkt sich und der Greifer schließt.

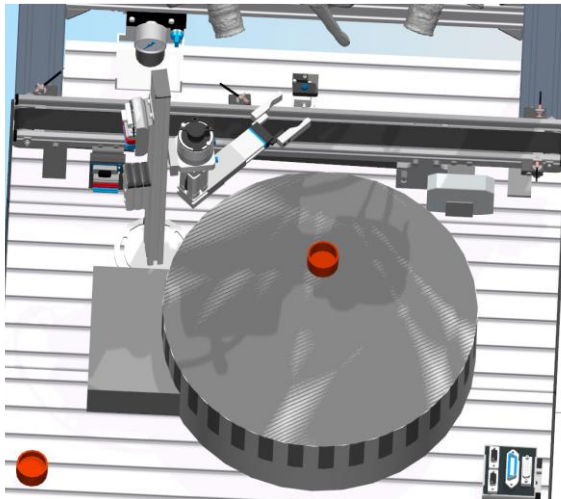
Das Dreh-Hub-Modul hebt das Bauteil an, schwenkt zum Messtisch und senkt es wieder ab auf den Drehtisch. Dort öffnet der Greifer.



Der Schwenkarm hebt sich und schwenkt wieder zum Förderband. Hat er diese Position erreicht, dreht sich der Drehtisch mit dem Bauteil entgegen dem Uhrzeigersinn.

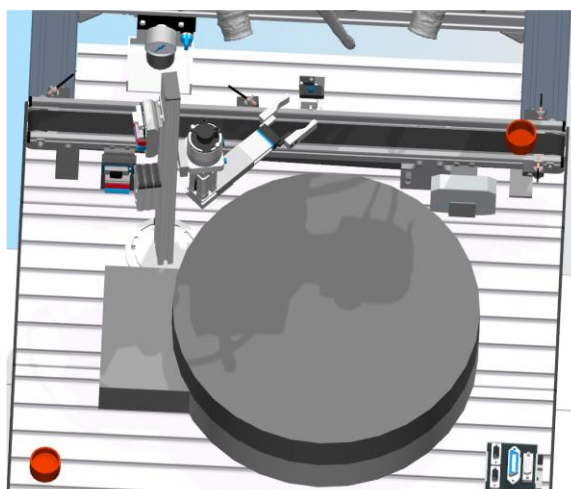
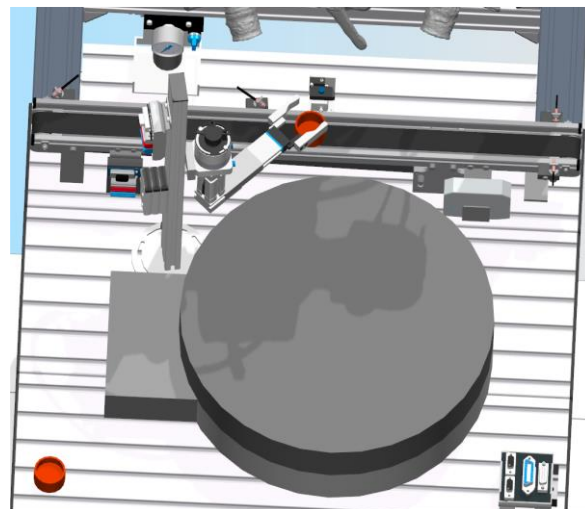
Abbildung 40: Ciros - Simulation, Schritt 3-5 ¹¹⁰

¹¹⁰ Eigene Darstellung



Der Drehtisch dreht das Bauteil um $+360^\circ$, wartet 2 Sekunden und dreht dann wieder zurück auf 0° . Damit wird der Scanprozess nachgestellt.

Dann wird das Bauteil wieder vom Drehtisch gegriffen vom Dreh-Hub-Modul und auf das Förderband gestellt.



Sobald der Schwenkarm wieder die obere Position erreicht hat, startet das Förderband und transportiert das Bauteil ans Bandende, wo in der Senke verschwindet.

Abbildung 41: Ciro - Simulation, Schritt 6-8 ¹¹¹

¹¹¹ Eigene Darstellung

Mit der erfolgreichen Durchführung des Tests kann der digitale Zwilling der Vermessungsstation validiert werden. Der simulierte Prozess läuft vollständig durch, sodass die grundlegende Funktionsfähigkeit der einzelnen Module, ihre Interaktion untereinander sowie die Ansteuerung durch das SPS-Programm überprüft werden.

Die Simulation ermöglicht es, die Steuerlogik der SPS in Verbindung mit den mechanischen Abläufen in Ciro zu testen und erste Optimierungen vorzunehmen. Der Ablauf des digitalen Zwillings konnte realitätsnah nachgestellt werden, wodurch ein Abgleich mit der physischen Anlage ermöglicht wird.

In dem Kapitel 6 wird eine detaillierte Analyse der Testergebnisse vorgenommen. Dabei werden identifizierte Abweichungen und potenzielle Optimierungsansätze betrachtet, um eine möglichst präzise und zuverlässige Simulation zu gewährleisten.

5 Implementierung der realen Vermessungsstation

Der Aufbau der physische Vermessungsstation ist der abschließende Schritt nach der Entwicklung des digitalen Zwillings. Mit der Simulation in CiroS und der Entwicklung der SPS-Steuerung im Tia Portal erfolgte eine detaillierte Planung und Überprüfung der Station, die die Grundlage für die Realisierung der Vermessungsstation bildet.

Die Implementierung gliedert sich in drei unterschiedliche Phasen, dem physischen Aufbau und die elektrische Verkabelung der einzelnen Module, der Steuerungsentwicklung sowie den Test der gesamten Anlage.

5.1 Aufbau und Montage der Station

Die Implementierung der realen Vermessungsstation beginnt mit dem mechanischen Aufbau des Gestells und der anschließenden elektrischen Vernetzung. Basierend auf der digitalen Simulation im CiroS-Modell wird die reale Station gemäß den geplanten Spezifikationen zusammengebaut.

5.1.1 Montage des Gestells

Nachdem die benötigten Komponenten von der Firma SMT GmbH sowie weitere Verbindungselemente wie Schrauben und Befestigungsmaterial aus dem Baumarkt beschafft werden, kann die mechanische Montage erfolgen. Der Aufbau der Station wird schrittweise von unten nach oben durchgeführt.

Zunächst wurden die vier Standfüße der Station mit den dazugehörigen Fußplatten an die vertikalen Aluminiumprofile befestigt. Diese Profile bilden die tragende Struktur der Station. Um eine stabile Basis zu schaffen, werden die vertikalen Profile mit vier horizontalen Aluminiumprofilen verbunden, die die erste Ebene der Station bilden. Zur Befestigung dieser Konstruktion kommen 90°-Winkel und Nutensteine zum Einsatz, die eine ausreichend stabile Verbindung sicherstellen.

Auf diese erste Rahmenkonstruktion wird anschließend eine MDF-Platte montiert, die zuvor mit Nuten an den äußeren Rändern bearbeitet wurde, um eine einfache Fixierung an den Profilen zu ermöglichen. Diese dient hauptsächlich als Aufbewahrungsplatte für den Edu-Trainer in der ersten Ebene.

Im nächsten Schritt werden die vertikalen Profile der zweiten Ebene installiert und erneut mit einer MDF-Platte abgedeckt, die als zusätzliche Fläche für weitere Komponenten, wie den PC dient.

Für den Schutz vor äußeren Einflüssen werden nun die Polycarbonat-Scheiben auf zwei gegenüberliegenden Seiten der Station installiert. Diese werden von oben in die vorgesehenen Nuten der vertikalen Aluminiumprofile eingeschoben, um eine stabile Seitenwand zu bilden. Erst nachdem diese Seitenwände angebracht sind, können die horizontalen Profile der dritten Ebene montiert werden, welche die Arbeitsfläche und Montagefläche der Station bildet. Auf diesen Profilen werden anschließend die Montageprofilplatten für die Befestigung der elektrischen und pneumatischen Komponenten installiert.

An der hinteren Polycarbonat-Scheibe werden nun gezielt Löcher gebohrt, um die Kabel der Module wie dem PC und dem Edu-Trainer nach außen zu führen. Diese Rückwand wird im

Gegensatz zu den Seitenwänden nicht in die Aluminiumprofile eingeschlossen, sondern von hinten angeschraubt. Dadurch bleibt sie abnehmbar, was eine einfache Wartung und Verkabelung der elektrischen Komponenten ermöglicht.

Zum Abschluss wird das Schloss für die vordere Polycarbonat-Tür installiert, sodass die Station bei Bedarf geöffnet oder verriegelt werden kann. Damit ist das Gestell der Vermessungsstation vollständig aufgebaut.

5.1.2 Verschaltung der Module

Die elektrische Verkabelung der Module auf der Montageplatte stellt eine zentrale Aufgabe bei der Implementierung der realen Vermessungsstation dar. Dabei werden die einzelnen Module mit der speicherprogrammierbaren Steuerung verbunden, um eine präzise Steuerung und Datenübertragung sicherzustellen.

Verdrahtung der *Festo*-Module

Die *Festo*-Module, insbesondere das Förderband und das Dreh-Hub-Modul, sind bereits weitestgehend vorverdrahtet und verfügen über ein Mini-I/O-Terminal. Dieses Terminal ist ab Werk mit fast allen Sensoren und Aktoren der jeweiligen Module verbunden und erfordert lediglich den Anschluss an die Steuerungseinheit. Die Verbindung zur SPS erfolgt über ein 15-poliges D-Sub HD-Kabel, das das I/O-Terminal mit dem C-Interface verbindet. Das C-Interface wiederum dient als Schnittstelle zur SPS und ermöglicht zum einen die Steuerung und zum anderen die Energieversorgung der Module.

In Abbildung 42 ist der Kontaktbelegungsplan des Förderbands dargestellt. Die meisten Anschlüsse sind bereits vorverdrahtet, allerdings sind die Ausgänge Q0-Q2 noch nicht verdrahtet.

Funktion	D-Sub HD	Klemmen	Benennung
I0	1	1	Werkstück am Bandanfang
I1	3	2	Werkstück in Bandmitte
I2	5	3	Kein Werkstück am Bandende
I3	7	4	
AI0	9	5	
AI1	10	6	
Q0	2	7	Band vorwärts
Q1	4	8	Band rückwärts
Q2	6	9	Vereinzelner ausfahren
Q3	8	10	
AQ0	11	11/12	
24 V A	12	24 V A	24 V Versorgung der Eingänge
24 V B	13	24 V B	24 V Versorgung der Ausgänge
GND A	15	GND A	0 V Versorgung der Eingänge
GND B	14	GND B	0 V Versorgung der Ausgänge

Abbildung 42: Modul Band - Kontaktbelegungstabelle¹¹²

¹¹² vgl. o. V.: Modul Band - DataSheet.pdf.

Der Motor des Förderbands muss noch zusätzlich über den zugehörigen Motorcontroller mit dem I/O-Terminal verbunden werden, da dieser zuvor nicht angeschlossen ist.

Der Stopper-Zylinder, der das Bauteil an einer definierten Position auf dem Förderband anhält, muss ebenfalls mit dem I/O-Terminal des Förderbands verdrahtet werden.

Die genaue Verdrahtung zwischen dem I/O-Terminal, dem Motorcontroller und dem Stopper-Modul ist der Tabelle 3 zu entnehmen.

Tabelle 3: Kontaktbelegungstabelle, Motorcontroller - Modul Band - Stopper¹¹³

Komponente	Klemme (Motor-controller)	Klemme (I/O-Terminal Modul Band)	Klemme (Motor-controller)	Funktion
Motorcontroller	1	7	-	Links-Lauf (p-schaltend)
	2	8	-	Rechts-Lauf (p-schaltend)
	6	-	8	Analogeingang zu Hilfsspannungsausgang
	9	-	-	Motoranschluss (-)
	10	-	-	Motoranschluss (+)
	11	-	12	Freigabe-Links/Quittierung
	12	-	16	Freigabe-Rechts/Quittierung
	14	24 V B	-	24 V Versorgung
	13	GND B	-	0 V Masse
	Stopper	-	9	-
-		GND B	-	0 V Masse für Stopper (schwarzes Kabel)

Im Gegensatz zum Förderband ist das Dreh-Hub-Modul bereits vollständig verdrahtet. Der Kontaktbelegungsplan dieses Moduls ist in Abbildung 43 dargestellt. Da das Dreh-Hub-Modul pneumatisch betrieben wird, muss es lediglich mit der Druckluftversorgung verbunden werden.

¹¹³ Eigene Darstellung

Funktion	D-Sub HD	Klemmen	Benennung
I0	1	1	Greifer offen
I1	3	2	Greifer oben
I2	5	3	Schwenkantrieb in Position „Band“
I3	7	4	Schwenkantrieb in Position „Messtisch“
AI0	9	5	
AI1	10	6	
Q0	2	7	Greifer öffnen
Q1	4	8	Greifer nach unten
Q2	6	9	Schwenkantrieb zu Position „Messtisch“ drehen
Q3	8	10	
AQ0	11	11/12	
24 V A	12	24 V A	24 V Versorgung der Ausgänge
24 V B	13	24 V B	24 V Versorgung der Eingänge
GND A	15	GND A	0V Versorgung der Ausgänge
GND B	14	GND B	0V Versorgung der Eingänge

Abbildung 43: Dreh-Hub-Modul - Kontaktbelegungstabelle¹¹⁴

Das Modul wird über Pneumatik Schläuche mit dem Druckluft-Modul verbunden, das ebenfalls den Stopper-Zylinder mit Druckluft versorgt.

Alle elektrischen Steuerungssignale werden über das bereits installierte Mini-I/O-Terminal weitergeleitet.

Verbindung der Module mit der SPS

Sowohl das Förderband als auch das Dreh-Hub-Modul werden über 15-polige D-Sub HD-Kabel mit dem C-Interface verbunden. Dabei ist die korrekte Zuordnung der Anschlüsse wichtig für die anschließende Zuweisung der SPS-Variablen:

- Das Förderband-Modul wird auf X1 gesteckt.
- Das Dreh-Hub-Modul wird auf X2 gesteckt.

Diese Reihenfolge legt fest, in welcher Abfolge die Sensoren und Aktoren von der SPS angesprochen werden.

Das C-Interface selbst wird über ein 24-poliges E/A-Datenkabel mit der SPS verbunden. Darüber erfolgt nicht nur die Steuerung der Module, sondern auch deren Spannungsversorgung.

¹¹⁴ vgl. o. V.: Dreh-Hub-Modul - DataSheet.pdf.

Im Gegensatz zu den anderen Modulen wird der Drehtisch nicht über die SPS, sondern direkt über ein PC-Programm gesteuert und mit Strom versorgt.

Der smartScan-HE Scanner wird ebenfalls über ein eigenes Modul mit Spannung versorgt und mit dem PC verbunden, um darüber gesteuert zu werden.

Der Edu-Trainer mit der SPS und der PC werden direkt an die Netzspannung angeschlossen. Damit eine reibungslose Kommunikation zwischen der SPS und dem PC möglich ist, werden beide Geräte über ein Profinet-Kabel miteinander verbunden. Diese Verbindung ermöglicht es später, das SPS-Programm auf die CPU zu laden und sich damit Online zu verbinden, um Fehlerbehebungen vorzunehmen.

Mit dieser vollständigen Verkabelung sind alle Module mit der Steuerung und der Spannungsversorgung verbunden und für die erste Inbetriebnahme vorbereitet.

5.2 Steuerung der physischen Vermessungsstation

Nachdem der digitale Zwilling erfolgreich getestet wurde, kann nun das Steuerungsprogramm auf die reale Vermessungsstation übertragen werden. Dabei wird das im TIA Portal entwickelte SPS-Programm des digitalen Zwillings als Grundlage verwendet. Allerdings sind Anpassungen erforderlich, da es bei der physischen Anlage entscheidende Unterschiede gibt.

Ein zentraler Unterschied ist die Ansteuerung des Drehtisches. Während dieser im digitalen Zwilling als Simulationsmodell mit direkter Motoransteuerung angesteuert wird, muss er nun in der realen Steuerung über das Messprogramm auf dem PC angesteuert werden. Zudem kommt eine neue Komponente hinzu: der HE-Scanner, der in der Simulation nicht integriert werden konnte. Dieser Scanner wird über das Messprogramm Polyworks angesteuert, das wiederum über ein OPC UA-Client Skript mit der SPS kommuniziert.

Im Folgenden werden die notwendigen Anpassungen der Steuerungslogik für die Automatisierung der Anlage erläutert, die den OPC UA-Server und -Client für die Kommunikation zwischen SPS und Messsoftware beinhaltet sowie die Implementierung des Polyworks-Makros zur Aktivierung des Messprozesses erläutert.

5.2.1 Ansteuerung des Messprogrammes in Polyworks

Damit der HE-Scanner von Hexagon automatisiert mit der SPS interagiert, muss das Mess- und Auswerteprogramm Polyworks in den Steuerungsprozess integriert werden. Polyworks bietet hierfür eine Plug-in-Schnittstelle, über die Scanner unterschiedlicher Hersteller eingebunden werden können. Zudem ermöglicht es die Erstellung von benutzerdefinierten Makros, um Abläufe wie den Start eines Scan-Prozesses oder die Erstellung von Prüfberichten zu automatisieren.

Ein solches Polyworks-Makro, das die Steuerung des Scanners übernimmt, wurde bereits in einer studentischen Arbeit an der HAW Hamburg entwickelt und für diese Anwendung bereitgestellt.¹¹⁵ Dieses Makro bildet die Basis für die Kommunikation zwischen der SPS und Polyworks.

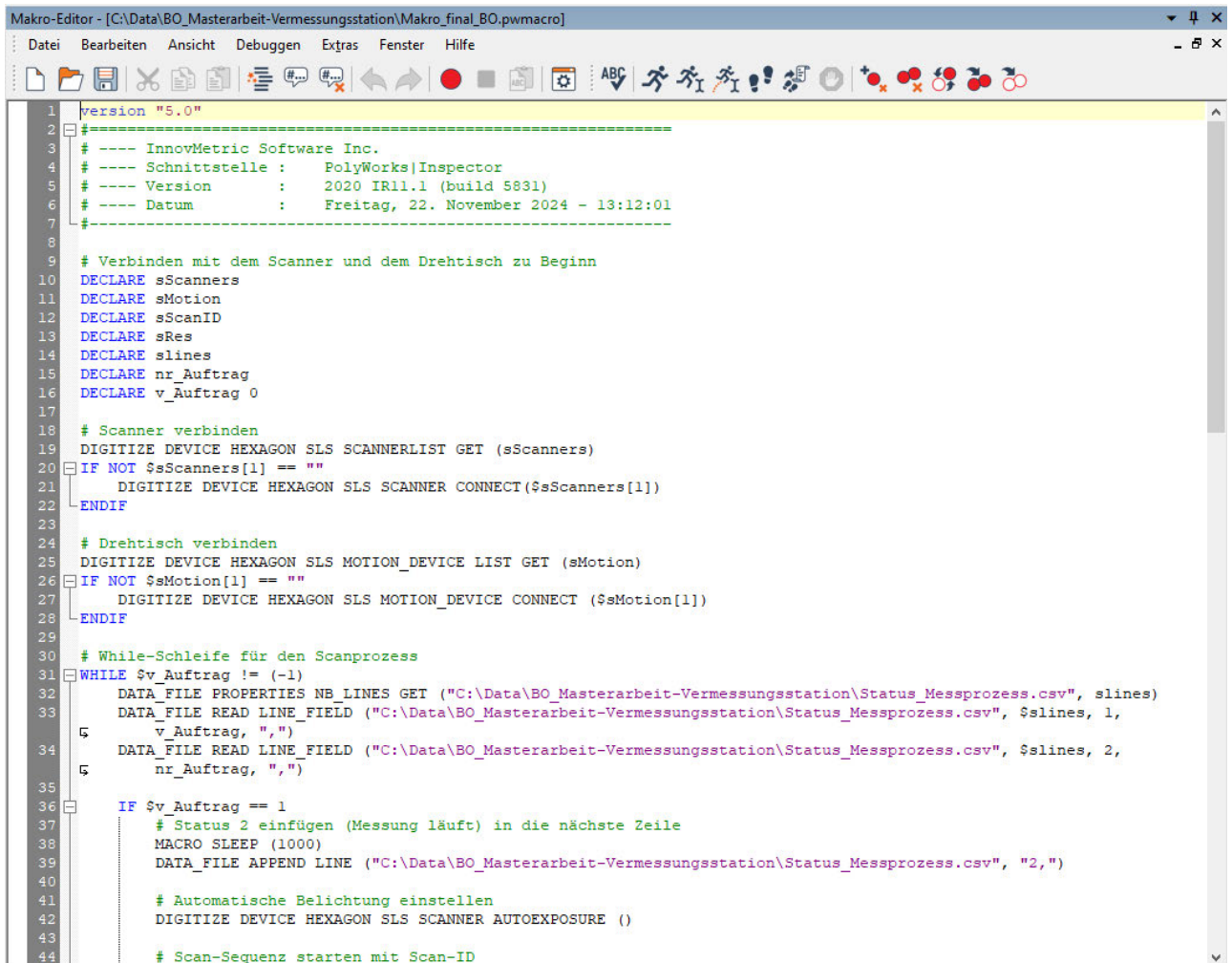
¹¹⁵ vgl. o. V.: VPPA_Hausarbeit_Integration 3D Scanner_Huneke Brouwers Kasten_ver03.pdf.

Funktionsweise des Makros und Interaktion mit der SPS

Das Makro ermöglicht die automatisierte Steuerung des Scanners über eine externe Excel-Datei, die als Schnittstelle zwischen Anwender/SPS und Messsoftware dient. Der Ablauf erfolgt in folgenden Schritten:

1. Anwender sendet Startbefehl:
 - In der ersten Spalte der Excel-Tabelle wird eine "1" eingetragen, die das Makro zum Starten des Scan-Prozesses triggert.
2. Polyworks beginnt mit der Messung:
 - Nach Erkennen des Startbefehls setzt das Makro den Status in der Excel-Tabelle auf "2" ("Scanprozess läuft") und startet einen einfachen Scan.
3. Durchführung des Scan-Vorgangs:
 - Während der Messung verarbeitet Polyworks die erfassten Daten, während der Anwender/ die SPS weiterhin auf die Rückmeldung aus der Tabelle wartet.
4. Scan abgeschlossen – Rückmeldung vom Makro:
 - Sobald die Messung beendet ist, wird der Wert in der Excel-Datei auf "3" gesetzt, was dem Anwender/ der SPS signalisiert, dass der Scan abgeschlossen ist.
5. Erstellung des Messreports:
 - Automatisch generiert Polyworks einen Prüfbericht als PDF, der im Webbrowser des PCs angezeigt wird.

Ein Code-Ausschnitt des Makros ist in Abbildung 44 dargestellt. Hier ist zu erkennen, dass das Programm in einer while-Schleife auf die Bedingung „1“ in der Excel-Tabelle wartet. Sobald diese erkannt wird, beginnt der Scan-Prozess und die Statuswerte werden entsprechend geändert.



```
1 |version "5.0"
2 |#-----
3 |# ---- InnovMetric Software Inc.
4 |# ---- Schnittstelle : PolyWorks|Inspector
5 |# ---- Version : 2020 IR11.1 (build 5831)
6 |# ---- Datum : Freitag, 22. November 2024 - 13:12:01
7 |#-----
8 |
9 |# Verbinden mit dem Scanner und dem Drehtisch zu Beginn
10 |DECLARE sScanners
11 |DECLARE sMotion
12 |DECLARE sScanID
13 |DECLARE sRes
14 |DECLARE slines
15 |DECLARE nr_Auftrag
16 |DECLARE v_Auftrag 0
17 |
18 |# Scanner verbinden
19 |DIGITIZE DEVICE HEXAGON SLS SCANNERLIST GET (sScanners)
20 |IF NOT $sScanners[1] == ""
21 |    DIGITIZE DEVICE HEXAGON SLS SCANNER CONNECT($sScanners[1])
22 |ENDIF
23 |
24 |# Drehtisch verbinden
25 |DIGITIZE DEVICE HEXAGON SLS MOTION_DEVICE LIST GET (sMotion)
26 |IF NOT $sMotion[1] == ""
27 |    DIGITIZE DEVICE HEXAGON SLS MOTION_DEVICE CONNECT ($sMotion[1])
28 |ENDIF
29 |
30 |# While-Schleife für den Scanprozess
31 |WHILE $v_Auftrag != (-1)
32 |    DATA_FILE PROPERTIES NB_LINES GET ("C:\Data\BO_Masterarbeit-Vermessungsstation\Status_Messprozess.csv", $lines)
33 |    DATA_FILE READ LINE_FIELD ("C:\Data\BO_Masterarbeit-Vermessungsstation\Status_Messprozess.csv", $lines, 1,
34 |    v_Auftrag, ",")
35 |    DATA_FILE READ LINE_FIELD ("C:\Data\BO_Masterarbeit-Vermessungsstation\Status_Messprozess.csv", $lines, 2,
36 |    nr_Auftrag, ",")
37 |
38 |    IF $v_Auftrag == 1
39 |        # Status 2 einfügen (Messung läuft) in die nächste Zeile
40 |        MACRO SLEEP (1000)
41 |        DATA_FILE APPEND LINE ("C:\Data\BO_Masterarbeit-Vermessungsstation\Status_Messprozess.csv", "2,")
42 |
43 |        # Automatische Belichtung einstellen
44 |        DIGITIZE DEVICE HEXAGON SLS SCANNER AUTOEXPOSURE ()
45 |
46 |        # Scan-Sequenz starten mit Scan-ID
```

Abbildung 44: Polyworks - Ausschnitt Automatisierungs-Makro¹¹⁶

Anpassungen und Optimierungen des Makros

Während der Implementierung treten Herausforderungen auf, die eine Anpassung des ursprünglichen Makros erforderlich machen. Ein Problem besteht darin, dass das Makro die „3“ für „Scan-Prozess beendet“ zu früh schreibt, während der Scanner noch in der Messphase ist. Dadurch würde eine angeschlossene SPS den Scan fälschlicherweise als abgeschlossen interpretieren und den nächsten Schritt zu früh einleiten.

Die Lösung besteht in einer Verzögerung nach dem Start des Scan-Prozesses. Diese wird in den Makro-Code integriert, sodass der Statuswechsel von „2“ auf „3“ erst dann erfolgt, wenn die Messung tatsächlich beendet ist.

Das finale Polyworks-Makro, das für die Automatisierung der Vermessungsstation verwendet wird, ist im Anhang „C: Polyworks-Makro“ dokumentiert.

Mit diesen Anpassungen ist das Makro nun so vorbereitet, dass es über die Schnittstelle der Excel-Datei ausgelöst werden kann und nach Abschluss des Messvorgangs die entsprechende Rückmeldung liefert.

¹¹⁶ Eigene Darstellung

Die eigentliche Kommunikation zwischen der SPS und dem Messsystem muss jedoch noch hergestellt werden, was im nächsten Kapitel mithilfe eines OPC UA Client-Server-Systems umgesetzt wird.

5.2.2 OPC UA Server und -Client

Eine bewährte und herstellerunabhängige Lösung für die Kommunikation zwischen Steuerungssystemen und externen Applikationen ist die OPC UA. Dieses Client-Server-basierte Kommunikationsprotokoll ermöglicht den sicheren Austausch von Daten zwischen der SPS und dem Messsystem, wodurch Statusmeldungen gesetzt und ausgelesen werden können.

Konfiguration des OPC UA-Servers in der SPS

Zunächst wird die SPS als OPC UA-Server konfiguriert. Dies erfolgt im TIA Portal, indem in den Einstellungen der CPU die OPC UA-Server-Funktionalität durch Aktivieren einer entsprechenden Option freigeschaltet wird. Damit die SPS-Variablen über OPC UA zugänglich sind, müssen die Variablen speziell über eine Häkchen-Setzung im Tia Portal nutzbar gemacht werden.

In Abbildung 45 sind die erstellten OPC UA-Variablen innerhalb eines Datenbausteins dargestellt.

	Name	Datentyp	Startwert	Remanenz	Erreichbar a...	Schrei...	Sichtbar i...	Einstellwert	Überwachung	Kommentar
1	Static			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
2	Start_Messung	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Start des Polyworks-Skripts
3	Status_Messprozess	Int	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Rückmeldung über Status Messprozess (...)

Abbildung 45: Tia Portal - Datenbaustein mit OPC UA Variablen¹¹⁷

Dabei werden zwei Variablen definiert:

- **„Start_Messung“**: Diese Variable dient als Trigger für den Beginn des Messprozesses. Da sie lediglich zwei Zustände (true/false) annehmen muss, ist sie als Datentyp *Bool* definiert.
- **„Status_Messprozess“**: Diese Variable gibt den aktuellen Status des Messprozesses zurück. Da sie mehrere Zustände repräsentieren muss (z. B. 1 = „Start“, 2 = „läuft“, 3 = „abgeschlossen“), wurde sie als integer deklariert.

Die Einbindung dieser Variablen in einen Datenbaustein bietet mehrere Vorteile: Sie ermöglicht eine strukturierte Verwaltung der Steuerungsdaten, erleichtert die Skalierbarkeit des Systems und optimiert die Kommunikation zwischen der SPS und externen Applikationen. Die Implementierung dieser Variablen in das SPS-Programm erfolgt im nächsten Kapitel 5.2.3.

¹¹⁷ Eigene Darstellung

Implementierung des OPC UA-Clients

Während die SPS als OPC UA-Server fungiert, wird für die Kommunikation zusätzlich ein OPC UA-Client benötigt. Dieser übernimmt die Aufgabe, Daten aus der Steuerung auszulesen, Befehle zu übermitteln und den Status der Messung zurückzumelden.

Da das Messprogramm Polyworks keine direkte OPC UA-Schnittstelle besitzt, wird der OPC UA-Client in dieser Arbeit mithilfe eines Python-Skripts realisiert. Dieses Skript übernimmt folgende Funktionen:

1. Herstellung einer Verbindung zum OPC UA-Server (SPS).
2. Überwachung des Startbefehls aus der SPS.
3. Schreiben des Startsignals in eine externe CSV-Datei (Name: „Status_Messprozess“), die als Schnittstelle für das Polyworks-Makro dient.
4. Kontinuierliches Auslesen des Messstatus aus der CSV-Datei und Rückmeldung an die SPS.
5. Fehlerhandling: Falls während der Kommunikation eine Unterbrechung auftritt, wird automatisch eine Wiederverbindung hergestellt.
6. Protokollierung der übermittelten Daten in die CSV-Datei.

Das vollständige Python-Skript für den OPC UA-Client ist im Anhang „

D: OPC UA Client – Python Skript“ dokumentiert.

Mit der erfolgreichen Implementierung des OPC UA-Clients ist die Verbindung zwischen SPS und Messsoftware nun hergestellt. Dies ermöglicht die automatisierte Steuerung des Messprozesses, indem das SPS-Programm Signale an das Messprogramm übermittelt und den Status der Vermessung in Echtzeit zurückmeldet. Die Einbindung dieser Steuerung in das SPS-Programm wird im nächsten Kapitel 5.2.3 erläutert.

5.2.3 SPS-Programm der realen Vermessungsstation

Mit der Anpassung des Messprogramms durch ein Polyworks-Makro und der Implementierung der OPC UA-Kommunikation kann nun das SPS-Programm des digitalen Zwilling auf die physische Vermessungsstation übertragen und entsprechend modifiziert werden.

Da die direkte Steuerung des Drehtisches nicht mehr über die SPS erfolgt, sondern manuell vor Prozessbeginn über Polyworks initialisiert wird, entfallen einige Prozessschritte, die zuvor in der Simulation vorhanden waren. Gleichzeitig müssen neue Befehle für die OPC UA-Kommunikation integriert werden, um den Austausch zwischen SPS und Messsoftware sicherzustellen.

Anpassung der Steuerungslogik

Um die veränderten Abläufe abzubilden, wird der bestehende Grafcet für den digitalen Zwillig überarbeitet und an die physische Station angepasst. Die neue Ablaufsteuerung, die im Anhang „E: Grafcet – physische Station“ dokumentiert ist, zeigt, dass sich die Anzahl der notwendigen Prozessschritte im Vergleich zur Simulation reduziert hat. Diese Anpassung wird auch in der Variablentabelle der realen Vermessungsstation (Abbildung 46) deutlich, in der ersichtlich ist, welche Schritte entfallen sind und welche neuen Signale nun berücksichtigt werden müssen.

Implementierung der realen Vermessungsstation

PLC-Variablen										
	Name	Variablen-tabelle	Datentyp	Adresse	Rema...	Erreic...	Schrei...	Sichtb...	Überwachung	Kommentar
1	X1_Sensor_FörderbandAnfang	X1_Modul Förderb...	Bool	%I0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		für Motor Förderband Start/Stop
2	X1_Sensor_FörderbandPos.2	X1_Modul Förderb...	Bool	%I0.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		für Förderband Start/Stop
3	X1_Sensor_Lichtschränke	X1_Modul Förderb...	Bool	%I0.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		negiert - True =
4	X2_Sensor_Dreh-Hub-Modul - ...	X2_Modul Dreh-Hub	Bool	%I0.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		True = Greifer offen
5	X2_Sensor_Dreh-Hub-Modul - ...	X2_Modul Dreh-Hub	Bool	%I0.5		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		True = Greifer oben
6	X2_Sensor_Dreh-Hub-Modul - ...	X2_Modul Dreh-Hub	Bool	%I0.6		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
7	X2_Sensor_Dreh-Hub-Modul - ...	X2_Modul Dreh-Hub	Bool	%I0.7		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
8	X1_Motor_Förderband	X1_Modul Förderb...	Bool	%Q0.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Motor Förderband ein/aus (:=1/0)
9	X1_Ventil_Stopper ausfahren	X1_Modul Förderb...	Bool	%Q0.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
10	X2_Ventil_Dreh-Hub-Modul - G...	X2_Modul Dreh-Hub	Bool	%Q0.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
11	X2_Ventil_Dreh-Hub-Modul - G...	X2_Modul Dreh-Hub	Bool	%Q0.5		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
12	X2_Ventil_Dreh-Hub-Modul - S...	X2_Modul Dreh-Hub	Bool	%Q0.6		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
13	X0	Standard-Variab...	Bool	%M10.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
14	X1	Standard-Variablen...	Bool	%M10.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
15	X2	Standard-Variablen...	Bool	%M10.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
16	X3	Standard-Variablen...	Bool	%M10.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
17	X4	Standard-Variablen...	Bool	%M10.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
18	X5	Standard-Variablen...	Bool	%M10.5		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
19	X6	Standard-Variablen...	Bool	%M10.6		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
20	X7	Standard-Variablen...	Bool	%M10.7		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
21	X8	Standard-Variablen...	Bool	%M11.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
22	X9	Standard-Variablen...	Bool	%M11.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
23	X10	Standard-Variablen...	Bool	%M11.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
24	X11	Standard-Variablen...	Bool	%M11.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
25	X12	Standard-Variablen...	Bool	%M11.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
26	X13	Standard-Variablen...	Bool	%M11.5		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
27	X14	Standard-Variablen...	Bool	%M11.6		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
28	X15	Standard-Variablen...	Bool	%M11.7		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
29	X16	Standard-Variablen...	Bool	%M12.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
30	X17	Standard-Variablen...	Bool	%M12.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
31	X18	Standard-Variablen...	Bool	%M12.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
32	X19	Standard-Variablen...	Bool	%M12.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
33	X20	Standard-Variablen...	Bool	%M12.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
34	X21	Standard-Variablen...	Bool	%M12.5		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
35	X22	Standard-Variablen...	Bool	%M12.6		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
36	X23	Standard-Variablen...	Bool	%M12.7		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
37	<Hinzufügen>					<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		

Abbildung 46: Tia Portal - PLC Variablen-tabelle - reale Station¹¹⁸

Zusätzlich fällt in der Abbildung auf, dass vor jeder Bezeichnung ein X1 bzw. ein X2 steht. Diese Präfixe kennzeichnen die Zugehörigkeit der jeweiligen Variable zum entsprechenden Stecker im C-Interface und somit zum zugehörigen Modul. Diese Struktur dient der besseren Übersichtlichkeit und ermöglicht dem Anwender eine vereinfachte Zuordnung der Ein- und Ausgänge zu den entsprechenden Modulen. Dadurch wird die Inbetriebnahme, Wartung und Fehlersuche erleichtert, da auf einen Blick erkennbar ist, welche Signale welchem Modul zugeordnet sind.

Integration der OPC UA-Variablen

Ein wesentlicher Bestandteil der Anpassung besteht in der Einbindung der zuvor erstellten OPC UA-Variablen in das Steuerungsprogramm. Diese Variablen übernehmen die Aufgabe, den Messprozess zu starten und den aktuellen Status der Vermessung an die SPS zurückzumelden. Um die Kommunikation mit dem OPC UA-Client reibungslos zu gestalten, werden im Funktionsbaustein Befehlsausgabe zwei neue Netzwerke implementiert und in SCL programmiert. Diese sind in folgender Abbildung 47 zu sehen und werden im Anschluss erläutert.

¹¹⁸ Eigene Darstellung


```
Netzwerk 3: Start Messprozess
Kommentar
1 // Mit Schritt 12 wird der Messprozess gestartet
2 IF "X12" THEN
3
4     "OPC-UA-Variablen".Start_Messung := TRUE; // Start Messprozess über OPC-UA-client
5
6 END_IF;
7

Netzwerk 4: Status Messprozess ändern
Kommentar
1 // Mit Schritt 14 wird der Messprozess Status geändert für Folgeprozess
2 IF "X14" THEN
3
4     "OPC-UA-Variablen".Status_Messprozess := 0; // Status Messprozess zurück auf "Bereit" setzen
5
6 END_IF;
7
```

Abbildung 47: Tia Portal - Befehlsausgabe der OPC UA Variablen in SCL¹¹⁹

Im Netzwerk 3 wird die Variable „Start_Messung“ im Schritt 12 (X12) des SPS-Programms auf TRUE (1) gesetzt. Dies geschieht, nachdem das Bauteil auf den Drehtisch abgelegt und der Greifarm des Dreh-Hub-Moduls in seine Ausgangsposition zurückgefahren ist. Diese Aktion sendet das Startsignal an den OPC UA-Client, der wiederum den Status in die csv-Datei schreibt und damit das Polyworks-Makro zur Messung auslöst.

Im nächsten Netzwerk 4 ist die Befehlsausgabe von Schritt 14 (X14) beschrieben, der im Anschluss an Schritt 13 ausgeführt werden soll. Da in Schritt 13 der Status im Messprogramm Polyworks geändert wird – auf 1 = Messprozess läuft – erfolgt die Statusänderung in TIA Portal erst mit Schritt 14, in dem die Variable Status_Messprozess auf 2 = Messprozess beendet gesetzt wird. Dadurch wird sichergestellt, dass die SPS erst nach erfolgreichem Abschluss der Messung erkennt, dass der Messprozess vollständig durchlaufen wurde und der nächste Zyklus vorbereitet werden kann.

Mit der Implementierung der angepassten Steuerungslogik und der Integration der OPC UA-Kommunikation ist das SPS-Programm nun vollständig auf die physische Vermessungsstation abgestimmt. Die Modifikationen gewährleisten eine reibungslose Interaktion zwischen der SPS und dem Messsystem, sodass die Messung nun automatisiert abläuft und der Status der Vermessung über die Steuerung nachverfolgt werden kann. Die vollständige Schrittkette sowie die finale Befehlsausgabe des SPS-Programms sind in „F: Physische Station – Schrittkette“ und „G: Physische Station – Befehlsausgabe“ dokumentiert. Damit ist das SPS-Programm bereit für den abschließenden Testlauf an der realen Vermessungsstation.

5.3 Test der physischen Station

Vorbereitung des Tests

Bevor die physische Vermessungsstation in Betrieb genommen und getestet werden kann, sind verschiedene Vorbereitungen erforderlich, um einen fehlerfreien Testablauf sicherzustellen. Ziel ist es, dass die gesamte Anlage unter Produktionsbedingungen stabil und zuverlässig arbeitet. Ein wesentlicher Aspekt der Vorbereitung ist die sichere Stromversorgung aller Module. Der Edu-Trainer (CPU), der PC mit dem Polyworks-Messprogramm, das Druckluftsystem sowie alle

¹¹⁹ Eigene Darstellung

angeschlossenen Sensoren und Aktoren müssen mit elektrischer und pneumatischer Energie versorgt werden, um eine vollständige Funktionsprüfung zu ermöglichen.

Neben der Hardware müssen auch die erforderlichen Software-Komponenten aktiv sein. Dazu gehört das Polyworks-Messprogramm, in dem das Makro zur Steuerung des Scanners und des Drehtisches integriert ist. Das Makro muss sich im Laufmodus (Run) befinden, damit es automatisch auf die Befehle aus der SPS reagieren kann. Die Software wird gemäß folgendem Pfad gestartet:

„Polyworks > Workspace1 > Automation_Project > Makro Editor > Einfügen > Makro_final_BO > Start“

Zusätzlich muss das zuvor implementierte OPC UA-Client (Python-)Skript ausgeführt werden. Dieses Skript übernimmt die Kommunikation zwischen SPS und Messprogramm und stellt sicher, dass Statusmeldungen korrekt verarbeitet werden.

Alle erforderlichen Programme und Dateien für den automatisierten Betrieb der physischen Anlage sind auf dem Messrechner unter folgendem Pfad abgelegt:

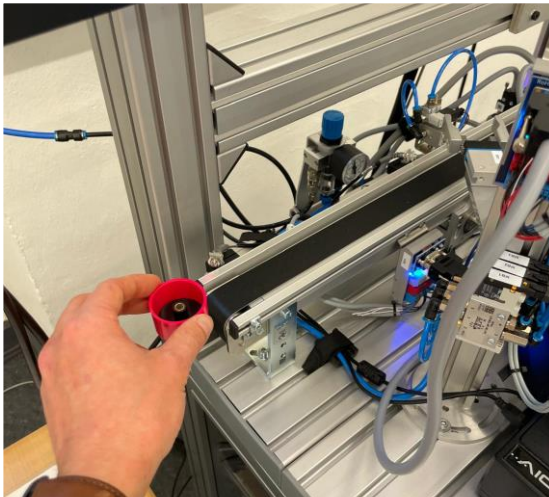
C:\Data\BO_Masterarbeit-Vermessungsstation

Dieses Verzeichnis enthält sämtliche relevanten Dateien, darunter das Polyworks-Makro, das SPS-Programm, das Python-Skript sowie die CSV-Datei zur Statuskommunikation zwischen Messprogramm und SPS.

Während des laufenden Betriebs ist es nicht zwingend erforderlich, dass das TIA Portal geöffnet ist. Es kann jedoch im Online-Modus verbunden werden, um eine Fehlersuche und Analyse in Echtzeit zu ermöglichen. Die CPU selbst muss in jedem Fall aktiv sein, da ohne diese keine Module mit Energie versorgt werden und der Prozess nicht starten kann.

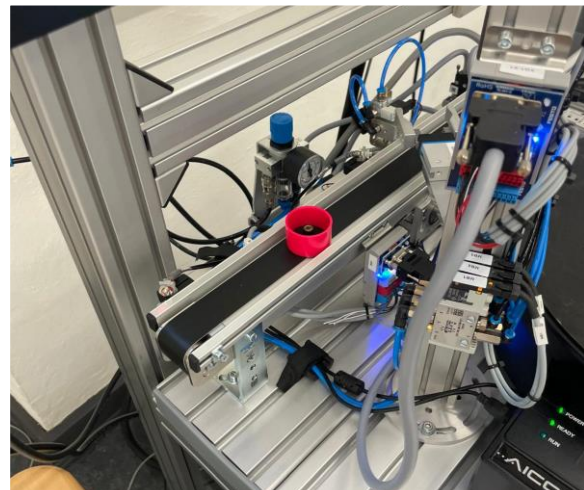
Durchführung des Tests

Mit dieser Vorbereitung kann der erste vollständige Testlauf der physischen Vermessungsstation durchgeführt werden. Der folgende Bildablauf dokumentiert die einzelnen Prozessschritte während der Simulation. Dabei wird jedes Bild mit einer kurzen Prozessbeschreibung versehen, um den Ablauf der Anlage zu veranschaulichen.



Das Bauteil wird am Anfang des Laufbandes positioniert, wo der Sensor "FörderbandAnfang" das Bauteil erkennt.

Mit dem Erkennen des Bauteils durch den Sensor schaltet sich das Förderband ein und befördert das Bauteil..



..bis das Bauteil am Stoppermodul aufgehalten wird. Das Förderband läuft noch 2 Sekunden nach, damit sich das Bauteil am Stopper korrekt positioniert, um korrekt gegriffen werden zu können.

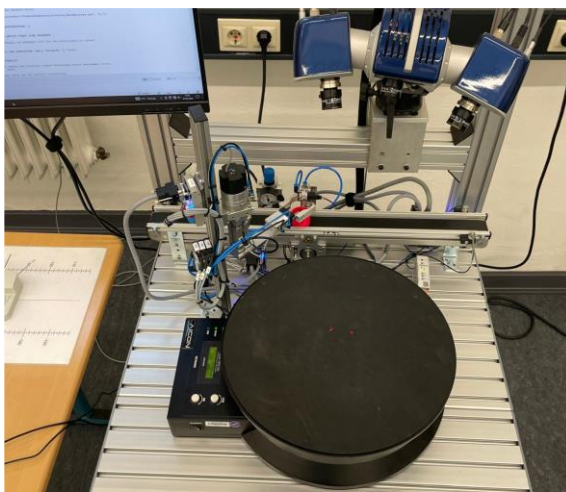
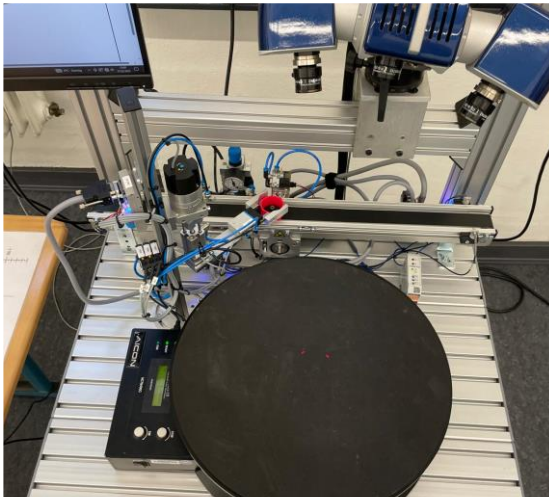
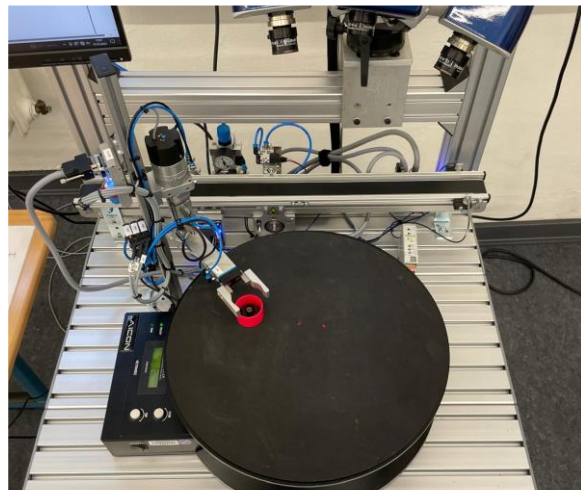


Abbildung 48: Test des physischen Zwillings - Bild 1-3¹²⁰



Nun fährt der pneumatische Greiferarm nach unten, die Greifer schließen und der Greiferarm verfährt wieder nach oben.

Der Greiferarm schwenkt zur Drehtischposition, senkt sich und legt das Bauteil auf den Drehtisch ab, indem es die Greifer öffnet.



Der Schwenkarm des Dreh-Hub-Moduls schwenkt wieder zurück, damit das Bauteil für die Messung nicht verdeckt wird.

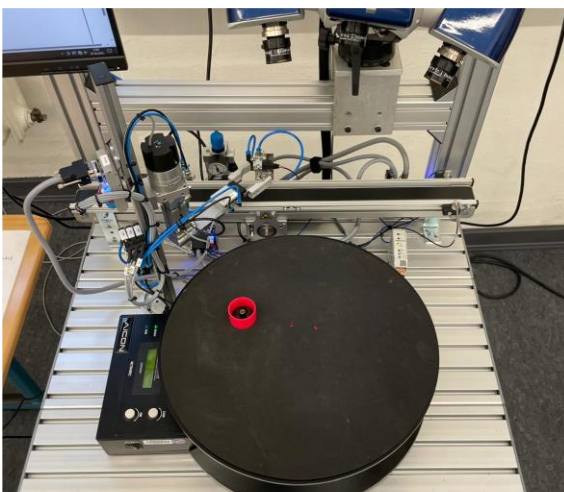
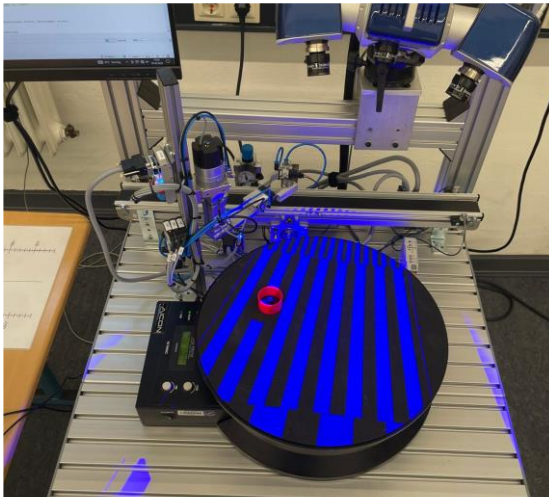
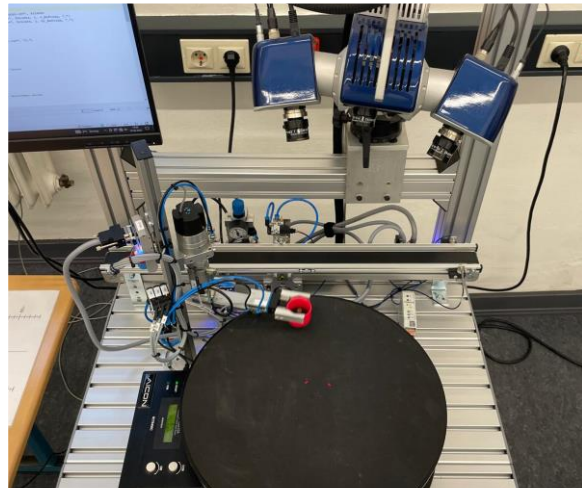


Abbildung 49: Test des physischen Zwillings - Bild 4-6¹²¹



Der Messprozess beginnt. Das Bauteil wird mithilfe von Streifenprojektionen vermessen an der Drehtischposition, auf dem es positioniert wurde.

Sobald der Messprozess beendet ist, verfährt der Greiferarm zum Bauteil, greift es und befördert es wieder zum Förderband, wo es vom Sensor 2 erkannt wird.



Nachdem der Schwenkarm des Greifers seine obere Endposition erreicht hat, startet das Förderband und befördert das Bauteil in Richtung Bandende.

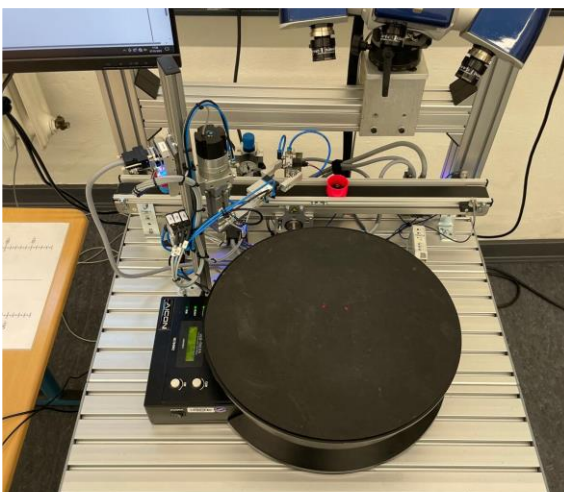
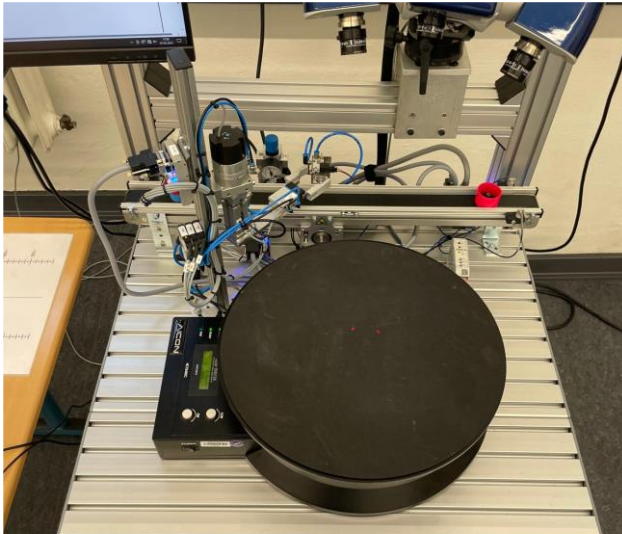


Abbildung 50: Test des physischen Zwillings - Bild 7-9¹²²



Am Förderbandende durchläuft das Bauteil eine Lichtschranke, dass nach durchlaufen einen 3-sekündigen Timer auslöst. In den 3 Sekunden wird das Bauteil noch "zur nächsten Station" transportiert und das Förderband stoppt dann mit Ablauf der 3 Sekunden.

Damit ist der Prozess abgeschlossen.

Abbildung 51: Test des physischen Zwillings - Bild 10¹²³

Nachdem das Bauteil in den Prozess eingegeben wird, kann der vollständige Ablauf der Vermessungsstation erfolgreich durchlaufen werden. Auch bei wiederholter Zugabe von Bauteilen verläuft der Prozess stabil und fehlerfrei, sodass sich die Steuerung, die mechanischen Komponenten sowie die Software-Programme als funktionstüchtig erweisen.

Allerdings zeigen sich in der Vorbereitung und Optimierung der Prozesssteuerung einige Herausforderungen, die weiter analysiert und verbessert werden müssen. Kapitel 6 befasst sich daher u. a. mit dem Ergebnis der Testdurchläufe, den Optimierungsbedarf und den Umsetzungsmöglichkeiten.

¹²³ Eigene Darstellung

6 Ergebnisanalyse und Vergleich der Testdurchläufe

Nach der erfolgreichen Umsetzung und Inbetriebnahme der Vermessungsstation folgt die Analyse der Testergebnisse, um die Funktionalität und Effizienz der entwickelten Lösung zu bewerten. Dabei wird untersucht, inwieweit die Simulation des digitalen Zwillinges mit den realen Abläufen übereinstimmt und welche Abweichungen oder Anpassungen erforderlich sind.

Ein zentraler Aspekt dieser Untersuchung ist die Gegenüberstellung der simulierten und tatsächlichen Prozesse, um die Modellierung und Steuerung zu überprüfen. Darüber hinaus werden die Erweiterung der Steuerung, die Stabilität des Ablaufs sowie die Leistungsfähigkeit der eingesetzten Softwarelösungen für die Vermessung und die Kommunikation mit der Steuerung hinsichtlich ihrer Stabilität und Fehleranfälligkeit geprüft.

Die gewonnenen Erkenntnisse dienen nicht nur der Überprüfung der Systemleistung, sondern auch der Identifikation von Optimierungspotenzialen. Durch eine detaillierte Analyse der durchgeführten Tests können mögliche Verbesserungen abgeleitet werden, um die Prozesssicherheit, Robustheit und Sicherheit der Vermessungsstation weiter zu steigern.

6.1 Vergleich der Ergebnisse zwischen digitalem Zwilling und physischen Zwilling

Die durchgeführten Tests sowohl am digitalen als auch am physischen Zwilling der Vermessungsstation zeigen sowohl Gemeinsamkeiten als auch Unterschiede in der Umsetzung und Funktionalität der Systeme. Der digitale Zwilling wird als präzises, virtuelles Abbild der realen Station entwickelt, um den Steuerungsablauf zu simulieren und zu optimieren. Die Testergebnisse beider Systeme zeigen, dass wesentliche Aspekte des Mess- und Förderprozesses übereinstimmen, aber auch einige Abweichungen bestehen, die durch die begrenzte Möglichkeit der Simulationssoftware als auch die physische Implementierung bedingt sind.

Gemeinsamkeiten zwischen digitalem und physischem Zwilling

Die wesentliche Übereinstimmung zwischen beiden Zwillingen liegt im grundlegenden Aufbau und Ablauf der Steuerung. Die Module und mechanischen Komponenten sind nahezu identisch positioniert, was eine hohe Vergleichbarkeit der Tests ermöglicht. Sowohl im digitalen als auch im physischen Zwilling kann ein vollständiger Prozessdurchlauf realisiert werden, bei dem das Bauteil durch die verschiedenen Stationsmodule geführt wird. Zudem zeigt sich, dass die SPS-Logik weitgehend übertragbar ist, da die grundlegenden Steuerungsfunktionen in beiden Varianten gleich umgesetzt werden.

Ein weiterer zentraler Punkt ist die funktionale Verschaltung der Module. Der Steuerungsablauf hinsichtlich der Greif-, Förder- und Positionierungsvorgänge konnte in beiden Umgebungen erfolgreich abgebildet werden. Dadurch ließ sich die Funktionsfähigkeit der Steuerung sowie die Kommunikation zwischen den Modulen in der Simulation vorab testen und anschließend in der physischen Implementierung nachweisen.

Unterschiede in den Testergebnissen

Trotz der hohen Übereinstimmung gibt es noch viele Abweichungen zwischen den Tests des digitalen und des physischen Zwillings. Eine der auffälligsten Unterschiede ist der Ablauf des Tests. Während der digitale Zwilling die Prozessschritte zwar korrekt simulieren kann, treten bei der Wiederholung von Durchläufen unerwartete Probleme auf. Trotz eines vollständigen und fehlerfreien ersten Prozessablaufs können Folgeprozesse im digitalen Zwilling nicht ordnungsgemäß wiederholt werden. Die genaue Ursache dieses Verhaltens ist unbekannt, jedoch kann die Steuerung als Fehlerquelle ausgeschlossen werden. Während der Simulation lässt sich im TIA Portal beobachten, dass die Steuerungslogik nach einem vollständigen Zyklus wie vorgesehen zurückgesetzt wird und der Prozess erneut gestartet wird. Dies deutet darauf hin, dass die Ursache für die fehlerhafte Wiederholung eher in der Simulation selbst oder in der Interaktion der virtuellen Module in CiroS liegt, die möglicherweise nicht korrekt zurückgesetzt werden oder bestimmte Zustände beibehalten.

Ein weiterer wesentlicher Unterschied besteht in der Ansteuerung des Drehtisches. Während der Drehtisch im digitalen Zwilling als Servomotor-Mechanismus in CiroS implementiert und direkt über die SPS angesteuert wird, erfolgt die Bewegung des physischen Drehtisches ausschließlich manuell über das Messprogramm Polyworks, ohne direkte Verbindung zur SPS. Dies führt dazu, dass der Drehtisch in dem Steuerungsprogramm des digitalen Zwillings als eigenständige Komponente mit entsprechenden Ein- und Ausgangssignalen existiert, während er in der realen Anlage nicht Teil der SPS-Steuerung ist. Folglich sind in der physischen Station keine Befehle zur Drehtischsteuerung im SPS-Programm integriert, was einen strukturellen Unterschied in der Ablaufsteuerung zwischen den beiden Systemen darstellt.

Zusätzlich kann in der Simulation die Einbindung der Messsoftware nicht nachgebildet werden. Dies bedeutet, dass die Kommunikation zwischen Steuerung und Polyworks über OPC UA nur in der realen Umgebung getestet werden kann. Die Integration des OPC UA-Clients und die Kommunikation mit der SPS sind somit Herausforderungen, die erst in der praktischen Umsetzung sichtbar werden und entsprechend angepasst werden müssen.

Ein weiterer Aspekt betrifft mechanische und physikalische Effekte, die in der Simulation nicht vollständig berücksichtigt werden. Beispielsweise treten in der realen Anlage Toleranzen und Ungenauigkeiten auf, die in der virtuellen Umgebung nicht simuliert werden können. Dazu gehört unter anderem die genaue Positionierung des Bauteils nach dem Greifvorgang.

Zusammenfassend zeigt sich, dass der digitale Zwilling eine zuverlässige Grundlage für die Entwicklung der Steuerungslogik und der korrekten Verschaltung der Komponenten mit der SPS und untereinander bietet und es ermöglicht, frühzeitig Fehler in der Ablaufsteuerung zu identifizieren. Dennoch gibt es Aspekte, die erst in der physischen Umsetzung sichtbar werden, insbesondere im Bereich der Sensorik, der mechanischen Interaktion und der Kommunikation mit externen Programmen wie der Messsoftware. Dies verdeutlicht, dass der digitale Zwilling eine wertvolle Ergänzung für die Entwicklung und Testung automatisierter Systeme darstellt, jedoch nicht alle Herausforderungen einer realen Implementierung vollständig erfassen kann.

6.2 Bewertung der Funktionalität der realen Station

Die ursprünglich geplante Funktionsweise der Vermessungsstation sieht eine vollautomatisierte Bearbeitung eines zylindrischen Bauteils vor. Der Prozess beginnt mit dem Transport des Bauteils vom Stationsanfang, gefolgt von einer präzisen Vermessung auf einem Drehtisch aus mehreren Positionen und dem anschließenden Weitertransport des Bauteils zum Stationsende.

Eine nahtlose Interaktion der mechanischen und softwareseitigen Schnittstellen ist dabei entscheidend, um einen reibungslosen Ablauf sicherzustellen. Zudem sind eine hohe Wiederholgenauigkeit und eine stabile Prozessführung unerlässlich, um eine zuverlässige Qualitätssicherung zu gewährleisten. Neben der reinen Funktionalität der einzelnen Module spielt auch die Benutzerfreundlichkeit eine zentrale Rolle. Die Station muss einfach zu bedienen sein und geeignete Schutzmechanismen enthalten, um sowohl den Anwender als auch die technische Ausrüstung vor Fehlbedienung oder Beschädigung zu schützen.

Die ersten durchgeführten Tests haben gezeigt, dass zentrale Funktionalitäten wie der Transport und die Übergabe des Bauteils zwischen den Modulen erfolgreich umgesetzt werden. Das Bauteil wird korrekt erkannt, an definierten Positionen übernommen und sicher in den nächsten Prozessschritt übergeben. Auch das wiederholte Durchlaufen des Prozesses mit neuen Bauteilen konnte zunächst erfolgreich realisiert werden, was die Stabilität der mechanischen Schnittstellen bestätigt. Zudem bietet die physische Konstruktion der Station mit ihrem stabilen Rahmen, transparenten Seitenwänden und einer verschließbaren Tür einen wirksamen Schutz vor äußeren Einflüssen wie Staub oder unerlaubtem Zugriff. Die robuste Bauweise sorgt zudem für ausreichende Dämpfung, sodass ein schwingungsarmer Betrieb gewährleistet ist.

Jedoch haben die durchgeführten Tests auch einige funktionale Einschränkungen und Fehler offenbart. Ein zentrales Problem betrifft den Vermessungsprozess. Während der Scanner das Bauteil zwar erfassen kann, wird aktuell lediglich ein einfacher Scan erstellt, der nicht ausreichend mit der Referenzdatei verglichen und ausgewertet werden kann (siehe Abbildung 52).



Abbildung 52: Beispiel eines Messreports nach Test des physischen Zwilling¹²⁴

Somit ist die geplante vollständige Vermessung derzeit nicht gegeben. Eine wesentliche Ursache hierfür liegt in den limitierten Funktionen des verwendeten Messprogramms Polyworks und der Skriptschnittstelle für den Hexagon-Scanner. Insbesondere der Befehl „WAIT FOR COMPLETION“ innerhalb des Hexagon Plug-In-Codes liefert nicht das erwartete Ergebnis, da der Abschluss des Scans vorzeitig signalisiert wird. Deshalb wurde der Umweg einer Zeitverzögerung gewählt, was jedoch keine robuste und langfristige Lösung darstellt. Außerdem ist das automatische Verfahren des Drehtisches während der Vermessung über das Polyworks-Skript nicht realisierbar. Stattdessen muss der Drehtisch vor der ersten Nutzung durch den Anwender manuell in Polyworks eingestellt werden.

Ein weiterer kritischer Fehler trat während der wiederholten Tests auf: das Messprogramm Polyworks stürzte während des Prozessablaufs unerwartet ab. Dies führte dazu, dass die SPS-Steuerung vergeblich auf eine Rückmeldung des Polyworks-Programms wartet und somit der Prozess an dieser Stelle stoppt. Eine Fortsetzung ist erst möglich, nachdem die CPU manuell neu gestartet wird (durch Drücken von „Stop“ und anschließend „Start“ an der CPU). Dieses Fehlverhalten erhöht die Anfälligkeit und mindert die Zuverlässigkeit der gesamten Anlage erheblich.

Neben diesen softwareseitigen Problemen fehlen derzeit noch essenzielle Sicherheits- und Bedienfunktionen. Sicherheitsrelevante Elemente wie Not-Aus- und Reset-Taster sind aktuell noch nicht implementiert, ebenso fehlt ein Bedienpult für eine komfortable und sichere Bedienung

¹²⁴ Eigene Darstellung

der Anlage. Die Integration solcher Funktionen ist entscheidend für den sicheren Betrieb und den Schutz der Nutzer sowie der Technik.

Zusammenfassend werden zentrale Aspekte der Automatisierung wie die Modulinteraktion und der Bauteiltransport erfolgreich umgesetzt. Dennoch besteht deutlicher Optimierungsbedarf, insbesondere im Bereich der Messsoftware sowie hinsichtlich der Sicherheitsmechanismen und Bedienmöglichkeiten, um die Anlage vollumfänglich funktional und sicher betreiben zu können.

6.3 Optimierungspotenziale und Erkenntnisse aus den Tests

Die durchgeführten Tests haben gezeigt, dass die Vermessungsstation zwar grundlegende Prozessabläufe erfolgreich umsetzen kann, jedoch in mehreren Bereichen Optimierungs- und Weiterentwicklungsbedarf besteht. Damit die Anlage ihre zukünftigen Aufgaben zuverlässig erfüllen kann, müssen insbesondere die Automatisierung des Messprozesses sowie sicherheitsrelevante Einrichtungen weiter verbessert und ergänzt werden. Die identifizierten Schwachstellen betreffen sowohl die Softwarearchitektur als auch die physische Sicherheit der Station.

Ein zentrales Optimierungspotenzial liegt in der Automatisierung und Stabilität des Vermessungsprozesses. Die derzeitige Umsetzung basiert auf einem Polyworks-Makro, das ursprünglich im Rahmen einer studentischen Arbeit entwickelt wurde. Dieses Makro ermöglicht zwar einen automatisierten Scan, die Erstellung eines Reports sowie eine rudimentäre Kommunikation über eine externe CSV-Datei, erweist sich jedoch als unflexibel und fehleranfällig. Die begrenzten Möglichkeiten zur direkten Messwertverarbeitung sowie die Abhängigkeit von einer Datei-basierten Kommunikationsmethode erschweren eine zuverlässige Prozessintegration. Zudem erfordert die derzeitige Lösung, dass der Anwender vor jedem Betrieb manuell alle Programme öffnet und teilweise einmal ausführen muss, was aufwendiger ist und was eine vollständige Automatisierung verhindert.

Ein vielversprechender Lösungsansatz wäre der Einsatz einer alternativen Messsoftware, die sowohl den Scan-Prozess als auch das Verfahren des Drehtisches besser unterstützt und eine direkte Schnittstelle zur SPS bietet. Eine geeignete Software sollte idealerweise als OPC UA-Client fungieren, sodass die Kommunikation direkt über die Steuerung erfolgen kann, ohne den Umweg über externe Dateien. Polyworks könnte weiterhin als Analyse- und Auswertesoftware genutzt werden, während die Steuerung und die Datenübertragung über eine leistungsfähigere Plattform abgewickelt werden. In diesem Zusammenhang sollte auch das aktuell verwendete OPC UA-Client-Skript überdacht werden, da es ursprünglich nur als Übergangslösung konzipiert ist. Eine alternative Messsoftware mit integrierter OPC UA-Schnittstelle könnte dieses Skript vollständig ersetzen und die Systemarchitektur erheblich vereinfachen.

Neben der Optimierung der Software und Kommunikationstechnologie ist die Implementierung zusätzlicher Sicherheitsmaßnahmen ein weiterer wichtiger Punkt. Bisher fehlen obligatorische Sicherheitsvorkehrungen wie Not-Aus- und Reset-Taster, die im Fall von Fehlfunktionen oder Nottfällen eine schnelle und sichere Reaktion ermöglichen. Die Integration eines Bedienpults, das diese Funktionen zusammenfasst, würde nicht nur die Sicherheit erhöhen, sondern auch die Bedienfreundlichkeit der Anlage erheblich verbessern. Die Montageplatte der Station bietet ausreichend Platz für ein Bedienpult oder Betätigungskomponenten, sodass die Implementierung problemlos realisiert werden kann.

Da bereits ein vollständiger Ablaufplan der Steuerung als Grafcet existiert, kann die Einbindung der Sicherheitsfunktionen relativ einfach erfolgen. In Abbildung 53 ist beispielhaft dargestellt, wie ein Reset-Schalter in den Ablauf integriert werden könnte.

Umsetzung Reset-Programm

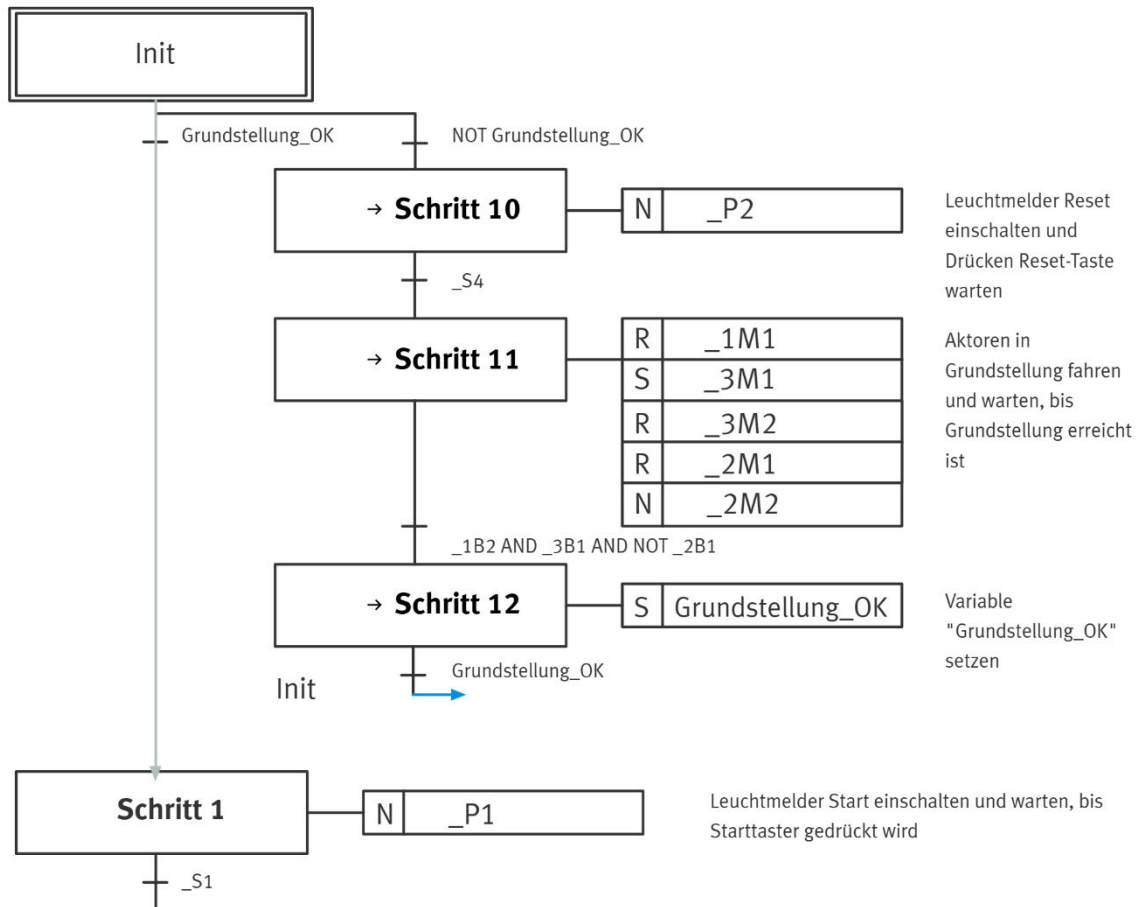


Abbildung 53: Grafcetbeispiel - Integration eines Reset-Mechanismus¹²⁵

Dort ist eine Alternativ-Verzweigung zu sehen, die es ermöglicht, alle Module in eine definierte Grundstellung zu fahren, falls die Anlage sich nicht bereits in diesem Zustand befindet. Durch das Betätigen des Reset-Knopfs (S4) wird sichergestellt, dass die Station wieder in ihre Ausgangsposition zurückkehrt, bevor der Prozess erneut gestartet wird.

Ergänzend dazu könnte der Not-Aus-Taster direkt in der SPS-Programmierung umgesetzt werden. Das TIA-Portal bietet hierfür spezielle Sicherheitsbausteine, wie etwa den „F-Not-Halt“-Baustein, die eine sichere Abschaltung ermöglichen. Um das Sicherheitskonzept abzurunden, wäre zudem die Implementierung eines Quittierungstasters erforderlich, mit dem nach einem Not-Aus das System erst wieder freigegeben werden kann.

Eine weitere sinnvolle Ergänzung wäre der nachträgliche Einbau eines RFID-Trackers über dem Förderband. Dieser könnte genutzt werden, um die Messergebnisse eindeutig mit dem jeweiligen Bauteil zu verknüpfen. Zusätzlich könnte durch die RFID-Technologie eine Separation von Bauteilen realisiert werden, die aufgrund zu hoher Maßabweichungen aussortiert werden müssen. Ein entsprechender Abscheider kann als Verteiler am Förderband implementiert

¹²⁵ vgl. o. V.: Festo Learning Experience.

werden, der solche Bauteile automatisch auf eine separate Schiene oder einen gesonderten Bereich leitet.

Die beschriebenen Maßnahmen stellen wesentliche Optimierungspunkte dar, die vorrangig umgesetzt werden sollten, um die Vermessungsstation für den dauerhaften Einsatz vorzubereiten. Neben diesen zentralen Anpassungen werden sich im laufenden Betrieb weitere Verbesserungsmöglichkeiten ergeben, insbesondere im Hinblick auf die Integration der Anlage in den Lehrbetrieb und die laufende Produktion der Modell Fabrik. Durch kontinuierliche Tests und Anpassungen kann die Funktionalität weiter gesteigert und eine robuste, effiziente und sichere Automatisierungslösung geschaffen werden.

7 Schlussfolgerungen und Ausblick

Die vorliegende Arbeit befasst sich mit der Entwicklung, Implementierung und dem Test einer automatisierten Vermessungsstation, die sowohl als digitaler Zwilling in einer Simulationsumgebung als auch als physische Anlage realisiert wird. Die durchgeführten Tests und Analysen ermöglichen eine umfassende Bewertung der Funktionalität, Prozessstabilität und Steuerungslogik beider Systeme. Dabei werden nicht nur die Gemeinsamkeiten und Unterschiede zwischen dem digitalen und physischen Zwilling herausgearbeitet, sondern auch Optimierungspotenziale identifiziert, die für eine weiterführende Entwicklung der Station von Bedeutung sind.

In diesem abschließenden Kapitel werden die wesentlichen Erkenntnisse der Arbeit zusammengefasst und eingeordnet. Darüber hinaus wird ein Ausblick auf mögliche Weiterentwicklungen und zukünftige Anwendungsbereiche gegeben, um die Anlage noch effizienter, sicherer und flexibler für den lehrbezogenen Einsatz nutzbar zu machen.

7.1 Zusammenfassung der Arbeit

Im Rahmen dieser Arbeit wird die Entwicklung und Implementierung einer vollautomatisierten Messtation für eine smarte Produktionsumgebung untersucht. Dabei wird der digitale Zwilling als ein Konzept der Industrie 4.0 genutzt, um die Anlage zunächst virtuell zu modellieren, zu testen und zu optimieren, bevor sie in der realen Umgebung aufgebaut wird. Die Arbeit umfasst die gesamte Prozesskette von der Konzeption über die digitale Simulation bis hin zur Inbetriebnahme und Testung der physischen Station.

Zu Beginn werden die theoretischen Grundlagen zur digitalen Fertigung, zur Automatisierungstechnik sowie zur Messtechnik dargestellt. Darauf aufbauend wird der digitale Zwilling der Vermessungsstation in der Simulationssoftware Ciroso erstellt. Hierbei werden mechanische Bewegungsabläufe, Steuerungsschnittstellen und Prozessabläufe nachgebildet, um die spätere Implementierung in der realen Umgebung vorzubereiten. Die Simulation ermöglicht die Validierung des Steuerungskonzepts und das frühzeitige Erkennen potenzieller Herausforderungen.

Im Anschluss zur Entwicklung des digitalen Modells wird die physische Station aufgebaut. Die mechanische Montage erfolgt unter Berücksichtigung industrieller Standards, wobei der modulare Aufbau der Anlage eine flexible Anpassung an zukünftige Erweiterungen erlaubt. Die elektrische Verdrahtung wird entsprechend den Anforderungen der speicherprogrammierbaren Steuerung umgesetzt, sodass alle Module zuverlässig miteinander kommunizieren können.

Ein wesentlicher Aspekt der Arbeit ist die Implementierung der Steuerungstechnik. Die SPS-Programmierung wird im TIA Portal realisiert, wobei besondere Anforderungen durch die Integration eines optischen Messsystems berücksichtigt werden müssen. Eine Herausforderung besteht in der Anbindung des Messprogramms Polyworks an die SPS, da die Steuerung des Drehtisches und der Scan-Prozess in der realen Station anders umgesetzt werden müssen als im digitalen Zwilling. Diese Kommunikation ist über die standardisierte OPC UA-Schnittstelle realisiert, wodurch eine effiziente und zuverlässige Datenübertragung zwischen den Systemen gewährleistet wird.

Die durchgeführten Testläufe der physischen Station zeigen, dass der grundlegende Prozessablauf erfolgreich umgesetzt werden konnte. Die Bauteilübergabe zwischen den Modulen funktioniert zuverlässig, und die Prozesssteuerung reagiert wie geplant auf externe Signale. Allerdings werden auch Herausforderungen identifiziert, insbesondere im Bereich der Messsoftwareintegration und der Wiederholbarkeit des Prozesses im digitalen Zwilling. Diese Erkenntnisse führen zu gezielten Anpassungen und Optimierungen, um die Robustheit und Flexibilität der Anlage zu verbessern.

Die abschließende Analyse der Testergebnisse ermöglichte eine detaillierte Bewertung der funktionalen Unterschiede zwischen dem digitalen und dem physischen Zwilling. Während viele Prozesse erfolgreich von der virtuellen in die reale Umgebung übertragen werden können, werden auch Verbesserungspotenziale identifiziert, insbesondere in Bezug auf die Automatisierung der Messdatenerfassung und die Erweiterung der Steuerungslogik.

Insgesamt zeigt die Arbeit, dass durch die Nutzung eines digitalen Zwillings die Entwicklung und Inbetriebnahme einer realen Anlage effizient unterstützt werden kann. Die Kombination aus Simulation, Automatisierung und vernetzter Messtechnik ermöglicht nicht nur eine optimierte Prozesssteuerung, sondern bildet auch die Grundlage für eine weiterführende Integration in smarte Produktionssysteme. Die gewonnenen Erkenntnisse liefern wertvolle Ansätze für zukünftige Forschungs- und Entwicklungsprojekte im Bereich der industriellen Automatisierung und Qualitätssicherung.

7.2 Weiterentwicklung und zukünftige Anwendungen

Die entwickelte Vermessungsstation bietet eine solide Grundlage für zukünftige Erweiterungen und Anwendungen im Bereich der smarten Fertigung. Durch die modulare Bauweise der Anlage und die flexible Steuerungsarchitektur lassen sich weitere Optimierungen und Funktionserweiterungen gezielt umsetzen. Ein wichtigespekt der Weiterentwicklung liegt in der Verbesserung des Messprozesses. Die derzeitige Lösung über das Polyworks-Makro zeigt Einschränkungen in der Flexibilität und Automatisierung. Eine Alternative mit einer direkteren Anbindung des Messsystems an die SPS über eine leistungsfähigere Schnittstelle, beispielsweise durch eine vollständig in OPC UA integrierte Lösung, kann den Prozess effizienter und stabiler gestalten.

Ein weiteres Entwicklungsfeld betrifft die Optimierung der Messgenauigkeit. Da der aktuelle Scan-Prozess des Pneumatikzylinders nicht die gewünschten Ergebnisse liefert, kann eine Lösung darin bestehen, mehrere Einzel-Scans zu einem vollständigen 3D-Bild zu kombinieren und diese Methode in das Makro zu integrieren. Sollte dieser Ansatz nicht erfolgreich sein – erste Tests zeigen, dass dies nicht zuverlässig funktioniert –, kann die Nutzung der Messsoftware Optocat eine vielversprechende Alternative sein. Diese Software bietet erweiterte Möglichkeiten zur präzisen Vermessung und könnte versuchen, in den Automatisierungsprozess integriert zu werden, indem sie über eine OPC UA-Schnittstelle verfügt und mit der SPS verknüpft wird.

Neben der Softwareoptimierung bietet auch die Verkabelung der Steuerungskomponenten noch Verbesserungspotenzial. Derzeit sind die Kabel des Motorcontrollers mit gleichfarbigen Leitungen vernetzt, was die Übersichtlichkeit und Wartung erschwert. Eine Ersetzung durch farblich differenzierte Kabel (z. B. rot für 24V und schwarz für GND) könnte die Verkabelung klarer und wartungsfreundlicher gestalten.

Ein weiteres mechanisches Optimierungsfeld betrifft den Greiferarm des Dreh-Hub-Moduls, der mechanisch durch ein zusätzliches Aluminiumstück verlängert wurde. Während der Tests zeigte sich, dass sich der Greifer nach mehreren Durchläufen leicht lockert. Zur Sicherstellung der langfristigen Stabilität sollte hier eine zusätzliche mechanische Fixierung implementiert werden. Auch der Schutz der Station kann weiter verbessert werden. Die derzeitige Polycarbonat-Tür ist vergleichsweise dünn, wodurch sie möglicherweise nicht ausreichend widerstandsfähig gegen äußere Einwirkungen ist. Eine stärkere Polycarbonat-Tür könnte für einen besseren Schutz der internen Komponenten und eine höhere mechanische Stabilität sorgen.

Zusätzlich könnte die Station durch eine RFID-Integration weiterentwickelt werden. Durch das Einfügen eines RFID-Trackers über dem Förderband können Bauteile individuell identifiziert und die jeweiligen Messdaten eindeutig zugeordnet werden. Dies würde nicht nur eine bessere Dokumentation der Messergebnisse ermöglichen, sondern auch eine gezielte Selektion von Bauteilen mit zu hohen Maßabweichungen. Um diesen Ansatz zu vervollständigen, kann ein Abscheidesystem in das Förderband integriert werden. Ein pneumatischer oder mechanischer Abscheider kann fehlerhafte Bauteile mit unzureichender Maßhaltigkeit automatisch auf eine separate Schiene umleiten und so den Qualitätskontrollprozess optimieren.

Die Automatisierung der Prozessvorbereitung stellt ein weiteres wichtiges Optimierungsfeld dar. Derzeit erfordert die Inbetriebnahme der Station mehrere manuelle Schritte, darunter das Starten der Messsoftware und des OPC UA-Clients. Eine Automatisierung dieser Startsequenzen kann die Bedienung vereinfachen und die Fehleranfälligkeit reduzieren. Hierbei bietet sich eine zentrale Steuerungssoftware an, die den gesamten Prozess aus einer Benutzeroberfläche heraus startet und überwacht.

Die Sicherheit und Bedienfreundlichkeit der Anlage stellen weitere Optimierungspotenziale dar. Die Integration eines physischen Bedienpults mit einem Reset- und Not-Aus-Schalter erhöht nicht nur die Sicherheit, sondern erleichtert auch den Umgang mit der Station im realen Betrieb. Zusätzlich können weitere Sicherheitsmechanismen, wie eine sensorische Überwachung kritischer Bewegungen oder eine Zugangskontrolle über ein Schlüsselsystem, in das Steuerungskonzept integriert werden.

Langfristig kann die Vermessungsstation in größere vernetzte Fertigungssysteme eingebunden werden. Eine Erweiterung der Steuerung um eine Cloud-Anbindung oder eine Anbindung an eine übergeordnete Produktionsleitzentrale würde eine noch tiefere Integration in Industrie 4.0-Umgebungen ermöglichen. Dies kann beispielsweise durch die Implementierung von KI-gestützten Analyseverfahren ergänzt werden, die Messdaten in Echtzeit auswerten und Anpassungen an den Produktionsprozess vorschlagen.

Neben den technischen Weiterentwicklungen bietet die Station auch Potenzial für den Einsatz in Forschung und Lehre. Durch die Kombination von digitalem Zwilling und physischer Automatisierung eignet sich die Anlage hervorragend als Demonstrations- und Schulungsobjekt für Studierende und Fachkräfte im Bereich der Industrieautomation. Dies kann dazu beitragen, zukünftige Ingenieure mit den modernen Methoden der digitalen Fertigung vertraut zu machen und die Nutzung von Simulationstechnologien in der Industrie weiter zu fördern.

Insgesamt stellt die entwickelte Vermessungsstation eine innovative Grundlage für zukünftige Anwendungen dar. Die gewonnenen Erkenntnisse und Optimierungsvorschläge liefern wertvolle Impulse für die Weiterentwicklung der Station, sei es durch technologische Verbesserungen,

erweiterte Automatisierungsfunktionen oder eine stärkere Vernetzung mit anderen Systemen. Die fortlaufende Optimierung und Adaption der Anlage an neue Anforderungen stellt sicher, dass sie auch in Zukunft als leistungsfähige und flexible Lösung für industriell-schulische Messaufgaben genutzt werden kann.

Literaturverzeichnis

Andelfinger, V.P. / Hänisch, T. (2017): Industrie 4.0 - Wie cyber-physische Systeme die Arbeitswelt verändern. Springer Gabler, Pfuhl.

Babel, W. (2024): Systemintegration in Industrie 4.0 und IoT: Vom Ethernet bis hin zum Internet und OPC UA. Springer Vieweg, Weil.

Deckert, R. (2019): Digitalisierung und Industrie 4.0: Technologischer Wandel und individuelle Weiterentwicklung. Springer Gabler, Hamburg.

GmbH, S.M.T.: Katalog Profiltechnik / Onlineshop Aluminiumprofile. SMT GmbH. <https://www.smt-montagetechnik.de/produkte.htm>. 16. November 2024.

Kaftan, J. (2015): SPS-Grundkurs mit SIMATIC S7: Aufbau und Funktion speicherprogrammierbarer Steuerungen, Programmieren mit SIMATIC S7. 8. Auflage, Vogel Business Media, Weißenburg.

Keferstein, C.P. / Marxer, M. (2015): Fertigungsmesstechnik: Praxisorientierte Grundlagen, moderne Messverfahren. 8. Auflage, Springer Vieweg, Werdenberg.

Langmann, R. (2021): Vernetzte Systeme für die Automatisierung 4.0: Bussysteme – Industrial Ethernet – Mobile Kommunikation – Cyber-Physical Systems. Carl Hanser Verlag, Düsseldorf.

Marxer, M. et al. (2021): Fertigungsmesstechnik: Alles zu Messunsicherheit, konventioneller Messtechnik und Multisensorik. 10. Auflage, Springer Vieweg, Buchs.

O. V. (2021): OPC UA auf der Feldebene - Industrial Communication Journal. <https://industrial-communication-journal.net/allgemein/opc-ua-auf-der-feldebene/>. 19. Jänner 2025.

O. V.: Festo Didactic InfoPortal: Modul Band. <https://ip.festo-didactic.com/InfoPortal/MPS/ConveyorModule/DE/index.html>. 9. November 2024.

O. V.: Festo Didactic InfoPortal: Dreh-Hub-Modul. <https://ip.festo-didactic.com/InfoPortal/MPS/RotatingLiftingModule/DE/index.html>. 9. November 2024.

O. V.: NCTB400C | OVR Turntable. nc-turntable. <https://www.ovr.co.kr/nctb400c>. 9. November 2024.

O. V.: Profiltechnik – item. <https://www.item24.com/de-de/profiltechnik>. 16. November 2024.

O. V.: CIROS 7, Universelles 3D-Simulationssystem | Festo DE. https://www.festo.com/de/de/p/ciros-7-universelles-3d-simulationssystem-id_PROD_DID_8140772/?page=0&tab=PRODUCT. 12. Jänner 2025.

O. V.: Grafcet-Studio: Einleitung. <https://www.mhj-wiki.de/de/grafcet-studio/einleitung/>. 26. Jänner 2025.

O. V.: Grafcet-Studio: Grafcet-Norm. <https://www.mhj-wiki.de/de/grafcet-studio/grafcet-din-en-60848/#grafcet-symbole>. 26. Jänner 2025.

O. V.: Programmierung Ablaufsteuerung- YouTube. https://www.google.com/search?q=ablaufplan+tia+portal&sca_esv=9267af3241730e66&biw=1708&bih=942&sxsrf=ADLYWILyakkAdKP1LzudoX8VmLnr4NdH7g:1722598567596&ei=p8SsZs6DJKuPxc8PgYmuqQI&oq=ablaufplan+tia&gs_lp=Eg1nd3Mtd2l6LXZpZGVvlg5hYmxhdWZwbGFuIHRpYSocCAAYBRAhGKABMgUQIRigAUiRiFD9D1iiEnAAeACQAQCYAYwCoAH7BKoBBTAuMy4xuAEBYAEA-

AEBmAlEoAKRBclCBBAjGCfCAgYQABgWGB7CAggQABiiBBiJBclCCBAAGIAEGKIEwgIFEAAyGASYAwCIBgGSBwUwLjMuMaAH5g4&scient=gws-wiz-video&udm=7#fpstate=ive&vld=cid:4e4a77f7,vid:xtMypCDuhkU,st:0. 9. Februar 2025.

O. V.: Festo Learning Experience. <https://lx.festo.com/de/learning-content-viewer?course-id=9316&parent-page=%2Fde%2Fcourses%2F9316&assignment-id=50614>. 23. Februar 2025.

O. V.: SmartScan | Streifenlichtscanner. Hexagon. <https://hexagon.com/de/products/smartsan>. 28. März 2025.

O. V.: Software für 3D-Maßanalyse und Qualitätskontrolle | PolyWorks. <https://www.innovmetric.com/de/products/polyworks-inspector>. 28. März 2025.

O. V.: SMT_Angebot_126235.pdf.

O. V.: SMT_Angebot_126237.pdf.

O. V.: Angebot_Festo_2004059061.pdf.

O. V.: Modul Band - DataSheet.pdf.

O. V.: Dreh-Hub-Modul - DataSheet.pdf.

O. V.: VPPA_Hausarbeit_Integration 3D Scanner_Huneke Brouwers Kasten_ver03.pdf.

Reinheimer, S. (2017): Industrie 4.0 - Herausforderungen, Konzepte und Praxisbeispiele. Springer Vieweg, Nürnberg.

Rückert, F.U. / Sauer, M. (2021): Die Erstellung eines digitalen Zwillings: Eine Einführung in Simcenter Amesim. Springer Vieweg, Saarbrücken.

Schluse, M. (2024): Experimentierbare Digitale Zwillinge: Konvergenz von Simulation und Realität. Springer Vieweg, Aachen.

Schuth, M. / Buerakov, W. (2017): Handbuch Optische Messtechnik: Praktische Anwendungen für Entwicklung, Versuch, Fertigung und Qualitätssicherung. Carl Hanser Verlag, Trier.

Schwab, K. (2016): Industrie 4.0: Die vierte industrielle Revolution. Pantheon.

Siemens AG (Hg) (2019): Programmierleitfaden für S7-1200/1500.

Tröster, F. (2015): Regelungs- und Steuerungstechnik für Ingenieure: Band 2: Steuerungstechnik. 4. Auflage, De Gruyter Oldenbourg, Heilbronn.

Verl, A. et al. (2024): Echtzeitsimulation in der Produktionsautomatisierung: Beiträge zu Virtueller Inbetriebnahme, Digitalem Engineering und Digitalen Zwillingen. Springer Vieweg, Stuttgart.

Westkämper, E. / Löffler, C. (2016): Strategien der Produktion - Technologien, Konzepte und Wege in die Praxis. Springer Vieweg, Stuttgart.

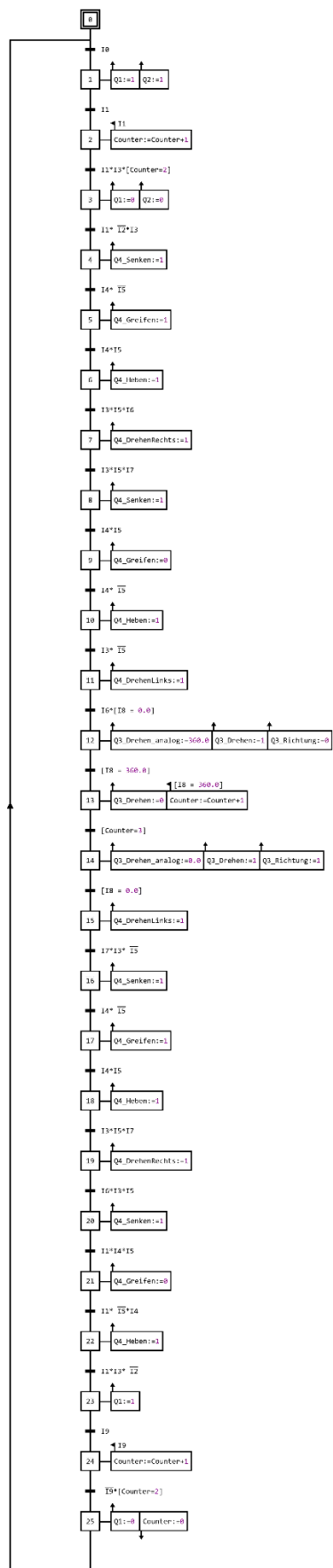
Weyrich, M. (2023): Industrielle Automatisierungs- und Informationstechnik. Springer Vieweg.

Anhang

A: Stationsbauteile-Liste

A	B	C	D	E	F	G	H	I
Position	Bauteilbezeichnung	Baureihe	Abmaße	Anzahl	Kommentar	Check CAD-Daten	Check Angebot	Check Lieferung
1	Alu Profil - Montageplatte oben	8 schwer	40x240x700	3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
2	Lasche Satz	8 Satz	40x160	8	Befestigung für die Montageprofile unter den Platten	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3	Abdeckkappe	8	40x240	6	Abdeckung Stirnseite Montageplatten	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4	Alu-Profil 60x60 vertikal	6 schwer(standard)	1300mm	2	In jedes Profil müssen die Kernbohrungen mit dem Gewinde M6x18 gefertigt werden	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
5	Abdeckkappe	6	60x60	2	für die hinteren vertikalen Profile	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
6	Alu-Profil 60x60 vertikal	6 schwer(standard)	690mm	2	In jedes Profil müssen die Kernbohrungen mit dem Gewinde M6x18 gefertigt werden	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
7	Alu-Profil 60x60 horizontal	6 schwer(standard)	580mm	9	für 2. Ebene	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
8	Alu-Profil 30x30 horizontal	6 schwer(standard)	580mm	4	zur Befestigung der oberen Platte an den Profilen	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
9	Befestigungswinkel Zn	universal	60x60x20	10	Boden für zwei Ebenen	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
10	Rückenwand - Polycarbonat Scheibe		640x640x16	2	Rückenwand	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
11	Wand - Polycarbonat Scheibe		694x654x2	1	gehen jeweils 9mm in die Nut (Nut 9,75mm tief)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
12	Tür - Polycarbonat Scheibe		598x558x6	2	links/rechts 1,5mm Spiel für Schloss, oben/unten 0,5mm	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
13	Scharnier PA 6	6	30x30	2	für die Tür	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
14	Verschlusssystem mit Griff	5/6/8	68x52	1	rechtsschließend, nicht bei smt - bei item bestellen	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
15	Stellfüße	6 D60	M12x75	4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
16	Fußplatte	6 60x60	M12	4	für Stellfüße	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
17	Gummieinsatz	6 D60	∅58	4	für Stellfuß	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
18	Winkel Satz Zn	6-30x30	30x30	76	Befestigung Profile untereinander/ inklusive Schrauben und Nutensteine	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
19	Abdeckprofil	6 Alu	690mm	16	36x Vorderseite und Stirnseiten + 16x Innen + 16x unter für vertikale Profile außen	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
20	Abdeckprofil	6 Alu	580mm	16	Horizontale Profile unten + 8x für Scanner-Prog für horizontale Profile außen	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
21	Kreuz Kabelbinderblock	8-40		15	mit Hammer Nut	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
22	Nutensteine M8	8	19mm	35	Befestigungen auf der Platte	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
23	Nutensteine M6	6		38	22x MDF Platten + 2x Scharniere + 4x Rückenwand + 10x für Befestigungswinkel	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
24						<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
25						<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

B: Grafcet - digitaler Zwilling



C: Polyworks-Makro

```

Makro-Editor - [C:\Data\BO_Masterarbeit-Vermessungsstation\Makro_final_BO.pwmacro]
Datei Bearbeiten Ansicht Debuggen Extras Fenster Hilfe
#-----
1 version "5.0"
2 #-----
3 # ---- InnovMetric Software Inc.
4 # ---- Schnittstelle : PolyWorks|Inspector
5 # ---- Version      : 2020 IR11.1 (build 5831)
6 # ---- Datum       : Freitag, 22. November 2024 - 13:12:01
7 #-----
8
9 # Verbinden mit dem Scanner und dem Drehtisch zu Beginn
10 DECLARE sScanners
11 DECLARE sMotion
12 DECLARE sScanID
13 DECLARE sRes
14 DECLARE slines
15 DECLARE nr_Auftrag
16 DECLARE v_Auftrag 0
17
18 # Scanner verbinden
19 DIGITIZE DEVICE HEXAGON SLS SCANNERLIST GET (sScanners)
20 IF NOT $sScanners[1] == ""
21     DIGITIZE DEVICE HEXAGON SLS SCANNER CONNECT ($sScanners[1])
22 ENDIF
23
24 # Drehtisch verbinden
25 DIGITIZE DEVICE HEXAGON SLS MOTION_DEVICE LIST GET (sMotion)
26 IF NOT $sMotion[1] == ""
27     DIGITIZE DEVICE HEXAGON SLS MOTION_DEVICE CONNECT ($sMotion[1])
28 ENDIF
29
30 # While-Schleife für den Scanprozess
31 WHILE $v_Auftrag != (-1)
32     DATA_FILE PROPERTIES NB_LINES GET ("C:\Data\BO_Masterarbeit-Vermessungsstation\Status_Messprozess.csv", slines)
33     DATA_FILE READ LINE_FIELD ("C:\Data\BO_Masterarbeit-Vermessungsstation\Status_Messprozess.csv", $slines, 1,
34         v_Auftrag, ",")
35     DATA_FILE READ LINE_FIELD ("C:\Data\BO_Masterarbeit-Vermessungsstation\Status_Messprozess.csv", $slines, 2,
36         nr_Auftrag, ",")
37
38     IF $v_Auftrag == 1
39         # Status 2 einfügen (Messung läuft) in die nächste Zeile
40         MACRO SLEEP (1000)
41         DATA_FILE APPEND LINE ("C:\Data\BO_Masterarbeit-Vermessungsstation\Status_Messprozess.csv", "2,")
42
43         # Automatische Belichtung einstellen
44         DIGITIZE DEVICE HEXAGON SLS SCANNER AUTOEXPOSURE ()
45
46         # Scan-Sequenz starten mit Scan-ID

```

```

Makro-Editor - [C:\Data\BO_Masterarbeit-Vermessungsstation\Makro_final_BO.pwmacro]
Datei Bearbeiten Ansicht Debuggen Extras Fenster Hilfe
30 # While-Schleife für den Scanprozess
31 WHILE $v_Auftrag != (-1)
32     DATA_FILE PROPERTIES NB_LINES GET ("C:\Data\BO_Masterarbeit-Vermessungsstation\Status_Messprozess.csv", slines)
33     DATA_FILE READ LINE_FIELD ("C:\Data\BO_Masterarbeit-Vermessungsstation\Status_Messprozess.csv", $slines, 1,
34         v_Auftrag, ",")
35     DATA_FILE READ LINE_FIELD ("C:\Data\BO_Masterarbeit-Vermessungsstation\Status_Messprozess.csv", $slines, 2,
36         nr_Auftrag, ",")
37
38     IF $v_Auftrag == 1
39         # Status 2 einfügen (Messung läuft) in die nächste Zeile
40         MACRO SLEEP (1000)
41         DATA_FILE APPEND LINE ("C:\Data\BO_Masterarbeit-Vermessungsstation\Status_Messprozess.csv", "2,")
42
43         # Automatische Belichtung einstellen
44         DIGITIZE DEVICE HEXAGON SLS SCANNER AUTOEXPOSURE ()
45
46         # Scan-Sequenz starten mit Scan-ID
47         DIGITIZE DEVICE HEXAGON SLS MOTION_DEVICE START SCAN SEQUENCE
48
49         # Verzögerung von 30 Sekunden einfügen, um genügend Zeit für den Scanvorgang zu lassen
50         MACRO SLEEP (30000)
51
52         DIGITIZE DEVICE HEXAGON SLS WAIT_FOR_COMPLETION (sRes, $sScanID, 0, "yes")
53
54         # Daten finalisieren
55         DIGITIZE DEVICE HEXAGON SLS FINALIZE
56
57         # Aktiviert das automatische Update des Projekts, sodass Änderungen sofort übernommen werden
58         PROJECT UPDATE AUTO_UPDATE ("On")
59
60         # Extrahiert alle gemessenen Daten für die weitere Verarbeitung
61         FEATURE EXTRACT MEASURED ALL
62
63         # Richtet die gemessenen Daten an den Referenzdaten aus, um Abweichungen zu ermitteln
64         ALIGN BEST_FIT DATA_TO_REFERENCE OPTIONS CREATE DATA_OBJECTS ("All")
65
66         # Führt eine weitere Ausrichtung durch, verwendet automatische Voreinstellungen für die Referenzoberflächen
67         ALIGN BEST_FIT DATA_TO_REFERENCE CREATE FIT_TO_REFERENCE_OBJECT_SURFACES USING_PRE_ALIGNMENT AUTOMATIC (
68             "Best-Fit Referenz 2")
69
70         TREEVIEW OBJECT SELECT NONE
71         TREEVIEW FEATURE SELECT (1, "On")
72
73         # Extrahiert gemessene Werte aus den zuvor definierten Merkmalen
74         MEASURE EXTRACT MEASURED ()

```

```
Makro-Editor - [C:\Data\BO_Masterarbeit-Vermessungsstation\Makro_fina_BO.pwmacro]
Datei Bearbeiten Ansicht Debuggen Extras Fenster Hilfe
TREEVIEW FEATURE SELECT (1, "On")
# Extrahiert gemessene Werte aus den zuvor definierten Merkmalen
MEASURE EXTRACT MEASURED ()
# Fügt eine Textmarkierung mit der Auftragsnummer in den Bericht ein
TREEVIEW REPORT_ITEM TEXT_FIELD PROPERTIES TEXT ($nr_Auftrag, "Textfeld")
# Exportiert die Messergebnisse in eine CSV-Datei für externe Auswertungen
FILE EXPORT_REPORT CSV_FILE ("C:\Users\student\Desktop\Automation\Python\Messergebnis.csv")
# Erstellt einen formatierten Bericht basierend auf einer vordefinierten Layout-Datei
FORMATTED_REPORT CREATE USING_LAYOUT ("Report 1", "C:\Users\student\innovmetric.20\report\layout\duwe3d.ez1")
TREEVIEW OBJECT SELECT NONE
TREEVIEW FORMATTED_REPORT SELECT (1, "On")
# Erzeugt einen Snapshot des Berichts für die Dokumentation
REPORT_ITEM SNAPSHOT CAPTURE ()
# Exportiert den formatierten Bericht in eine PDF-Datei für die Berichterstellung
FILE EXPORT_FORMATTED_REPORT PDF ("C:\Users\student\Desktop\Automation\Python\Report.pdf", "On")
TREEVIEW OBJECT SELECT NONE
TREEVIEW FORMATTED_REPORT SELECT (1, "On")
EDIT OBJECT DELETE ()
TREEVIEW OBJECT SELECT NONE
TREEVIEW DATA SELECT (1, "On")
EDIT OBJECT DELETE ()
# Status 3 einfügen (Scanprozess abgeschlossen) in die nächste Zeile
MACRO SLEEP (1000)
DATA_FILE APPEND LINE ("C:\Data\BO_Masterarbeit-Vermessungsstation>Status_Messprozess.csv", "3,")
ENDIF
# Wartezeit zwischen Überprüfungen
MACRO SLEEP (3000)
ENDWHILE
# Fehlerstatus setzen, falls Schleife abbricht
DATA_FILE APPEND LINE ("C:\Data\BO_Masterarbeit-Vermessungsstation>Status_Messprozess.csv", "99")
MACRO PAUSE ("Fehler: Makro beendet", "Bitte bestätigen und Makro erneut starten")
```

D: OPC UA Client – Python Skript

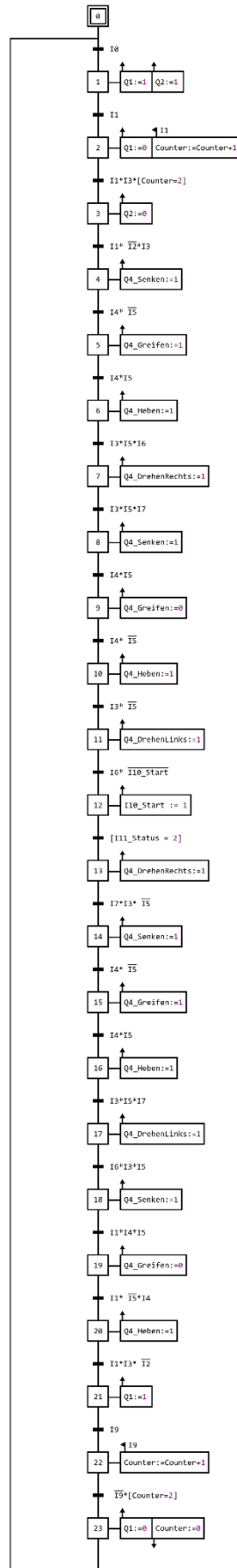
```
OPC-UA-Client.py
1 from opcua import Client, ua
2 import csv
3 import time
4 import traceback
5
6 # Funktion zum Lesen des letzten Status aus der CSV-Datei
7 def read_last_status_from_csv(file_path):
8     try:
9         with open(file_path, 'r', encoding='latin1') as file: # Ändere die Codierung zu 'latin1'
10             lines = file.readlines()
11             if not lines:
12                 return None # Datei ist leer
13             last_line = lines[-1].strip() # Letzte Zeile
14             return last_line.split(',')[0] # Erster Eintrag in der Zeile (Status)
15     except FileNotFoundError:
16         print("CSV-Datei nicht gefunden. Erstelle eine neue.")
17         with open(file_path, 'w', encoding='latin1') as file:
18             pass # Leere Datei erstellen
19     return None
20
21 # Funktion zum Hinzufügen eines Status in die CSV-Datei
22 def append_status_to_csv(file_path, status):
23     try:
24         with open(file_path, 'a', encoding='latin1', newline='') as file:
25             writer = csv.writer(file)
26             writer.writerow([status])
27             print(f"Status {status} in die CSV-Datei geschrieben.")
28     except Exception as e:
29         print(f"Fehler beim Schreiben in die CSV-Datei: {e}")
30
31 # Funktion zur sicheren Verbindung mit dem OPC UA Server
32 def connect_to_opcua(client, retries=3, delay=5):
33     for attempt in range(retries):
34         try:
35             client.connect()
36             print("Connected to OPC UA Server")
37             return True
38         except Exception as e:
39
40 OPC-UA-Client.py
31 # Funktion zur sicheren Verbindung mit dem OPC UA Server
32 def connect_to_opcua(client, retries=3, delay=5):
33     for attempt in range(retries):
34         try:
35             client.connect()
36             print("Connected to OPC UA Server")
37             return True
38         except Exception as e:
39             print(f"Verbindungsversuch fehlgeschlagen (Versuch {attempt + 1}/{retries}): {e}")
40             time.sleep(delay)
41     return False
42
43 # OPC UA Server-Einstellungen
44 opcua_server_url = "opc.tcp://192.168.0.1:4840" # IP-Adresse der SPS
45 csv_file_path = "C:\\Data\\B0_Masterarbeit-Vermessungsstation\\Status_Messprozess.csv"
46
47 client = Client(opcua_server_url)
48
49 try:
50     # Verbindung herstellen
51     if not connect_to_opcua(client):
52         raise ConnectionError("Verbindung zum OPC UA Server fehlgeschlagen")
53
54     # OPC-UA-Variablen definieren
55     start_messung_node = client.get_node('ns=3;s="OPC-UA_Variablen"."Start_Messung"')
56     messprozess_status_node = client.get_node('ns=3;s="OPC-UA_Variablen"."Status_Messprozess"')
57
58     # Initialstatus setzen
59     dv = ua.DataValue(ua.Variant(0, ua.VariantType.Int16))
60     messprozess_status_node.set_value(dv)
61     print("Messprozess-Status auf 0 (Bereit) gesetzt")
62
63     while True:
64         try:
65             # Überprüfen, ob Start_Messung aktiviert wurde
66             start_messung = start_messung_node.get_value()
67
68             if start_messung == 1:
```



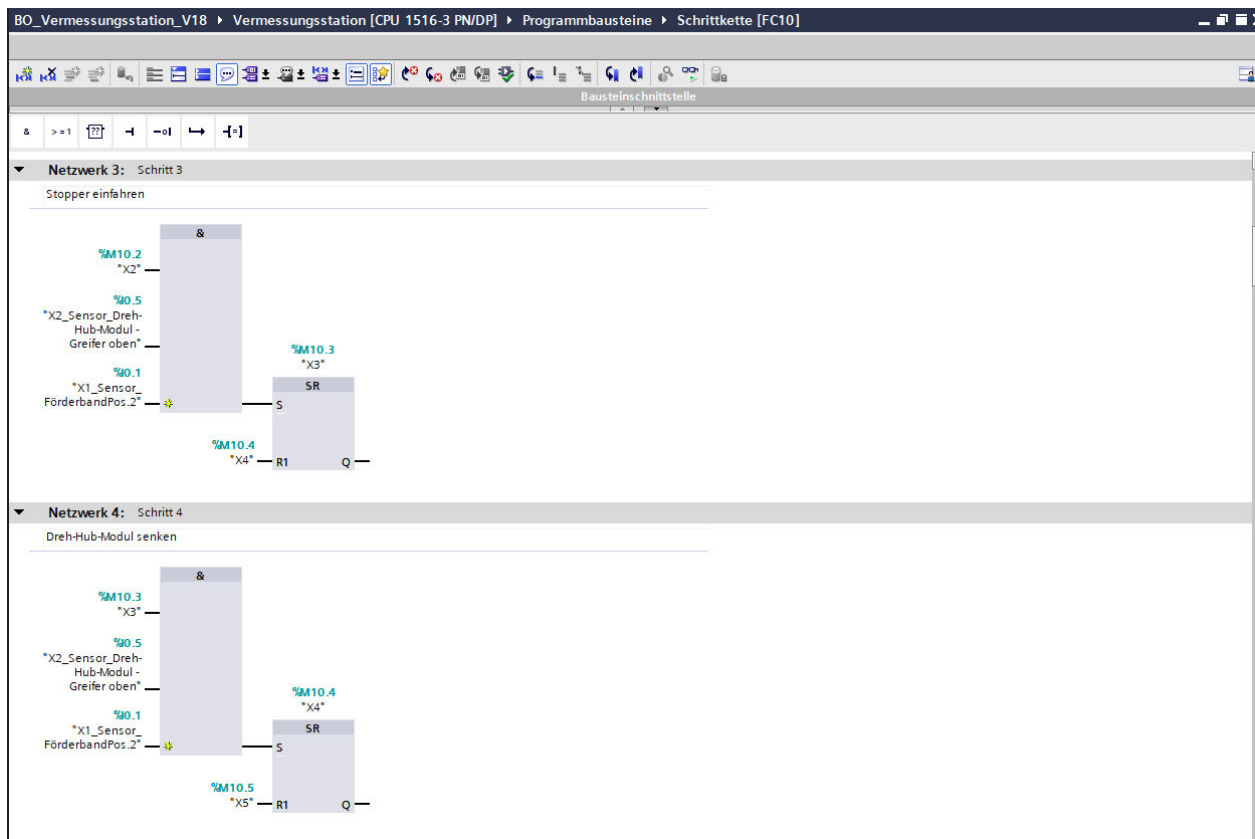
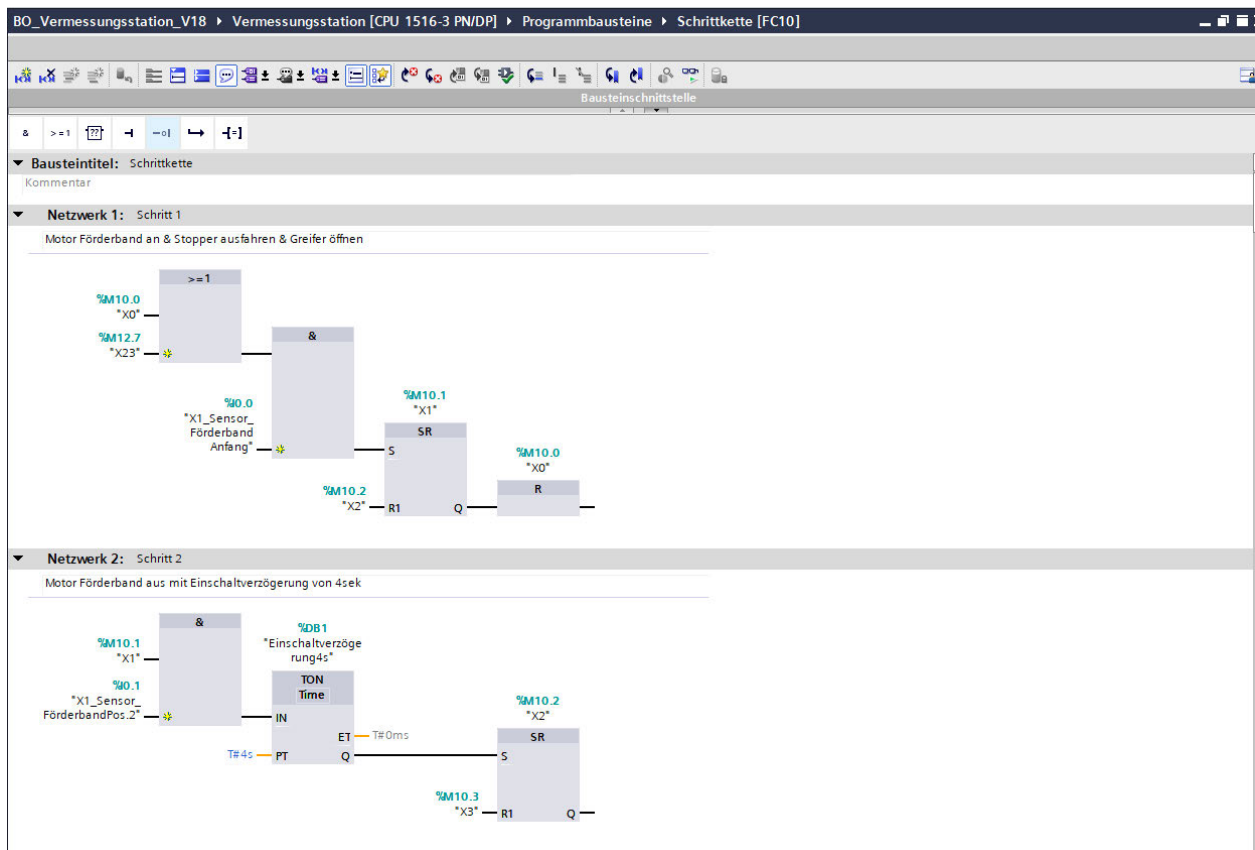
```
OPC-UA_Client.py x
63 while True:
64     try:
65         # Überprüfen, ob Start_Messung aktiviert wurde
66         start_messung = start_messung_node.get_value()
67
68         if start_messung == 1:
69             print("Messprozess gestartet")
70
71             # Status 1 in die CSV-Datei schreiben (Messung läuft)
72             append_status_to_csv(csv_file_path, "1,")
73
74             # Auf den Abschluss der Messung warten
75             while True:
76                 last_status = read_last_status_from_csv(csv_file_path)
77                 if last_status == "3": # Messprozess abgeschlossen
78                     print("Messprozess abgeschlossen. Rückmeldung an SPS.")
79                     dv.Value = ua.Variant(2, ua.VariantType.Int16) # Status "abgeschlossen"
80                     messprozess_status_node.set_value(dv)
81                     break
82                 elif last_status == "99": # Fehlerstatus
83                     print("Fehler im Messprozess. Rückmeldung an SPS.")
84                     dv.Value = ua.Variant(99, ua.VariantType.Int16) # Status "Fehler"
85                     messprozess_status_node.set_value(dv)
86                     break
87
88                 # Kurze Pause vor der nächsten Überprüfung
89                 time.sleep(1)
90
91             # Start_Messung in der SPS zurücksetzen
92             start_messung_node.set_value(ua.DataValue(ua.Variant(0, ua.VariantType.Boolean)))
93             print("Start_Messung zurückgesetzt")
94
95             # Pause zwischen den Hauptschleifen
96             time.sleep(1)
97
98     except Exception as e:
99         print(f"Fehler während der Ausführung: {e}")
100        traceback.print_exc()
```

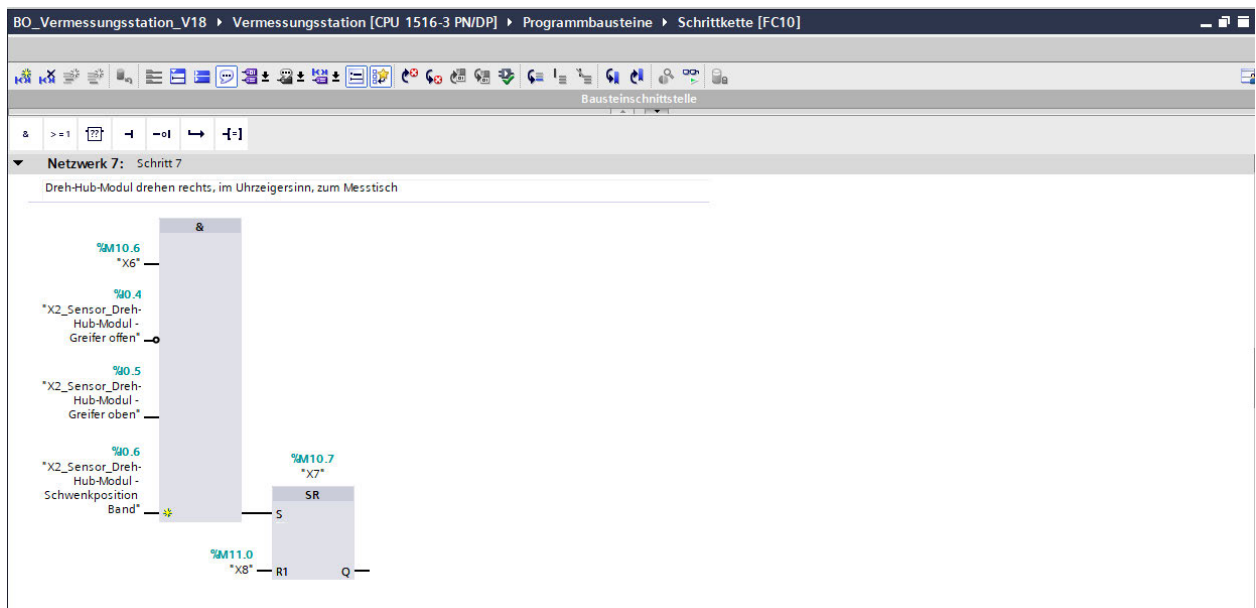
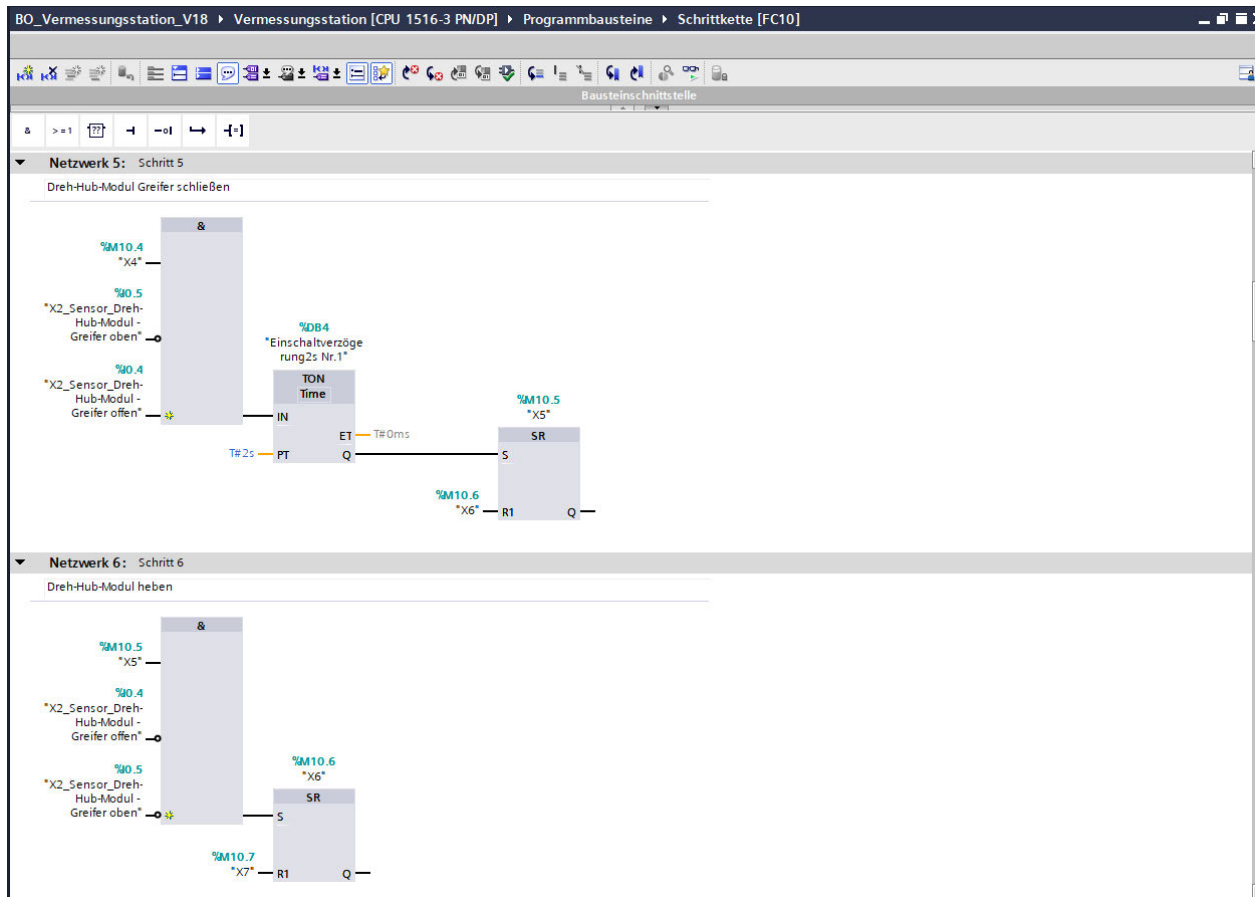
```
OPC-UA_Client.py x
78         print("Messprozess abgeschlossen. Rückmeldung an SPS.")
79         dv.Value = ua.Variant(2, ua.VariantType.Int16) # Status "abgeschlossen"
80         messprozess_status_node.set_value(dv)
81         break
82     elif last_status == "99": # Fehlerstatus
83         print("Fehler im Messprozess. Rückmeldung an SPS.")
84         dv.Value = ua.Variant(99, ua.VariantType.Int16) # Status "Fehler"
85         messprozess_status_node.set_value(dv)
86         break
87
88         # Kurze Pause vor der nächsten Überprüfung
89         time.sleep(1)
90
91         # Start_Messung in der SPS zurücksetzen
92         start_messung_node.set_value(ua.DataValue(ua.Variant(0, ua.VariantType.Boolean)))
93         print("Start_Messung zurückgesetzt")
94
95         # Pause zwischen den Hauptschleifen
96         time.sleep(1)
97
98     except Exception as e:
99         print(f"Fehler während der Ausführung: {e}")
100        traceback.print_exc()
101        print("Versuche, Verbindung wiederherzustellen...")
102        if not connect_to_opcua(client):
103            print("Wiederverbindung fehlgeschlagen. Beende Programm.")
104            break
105
106 finally:
107     # Verbindung trennen
108     try:
109         client.disconnect()
110         print("Disconnected from OPC UA Server")
111     except Exception as e:
112         print(f"Fehler beim Trennen der Verbindung: {e}")
113
```

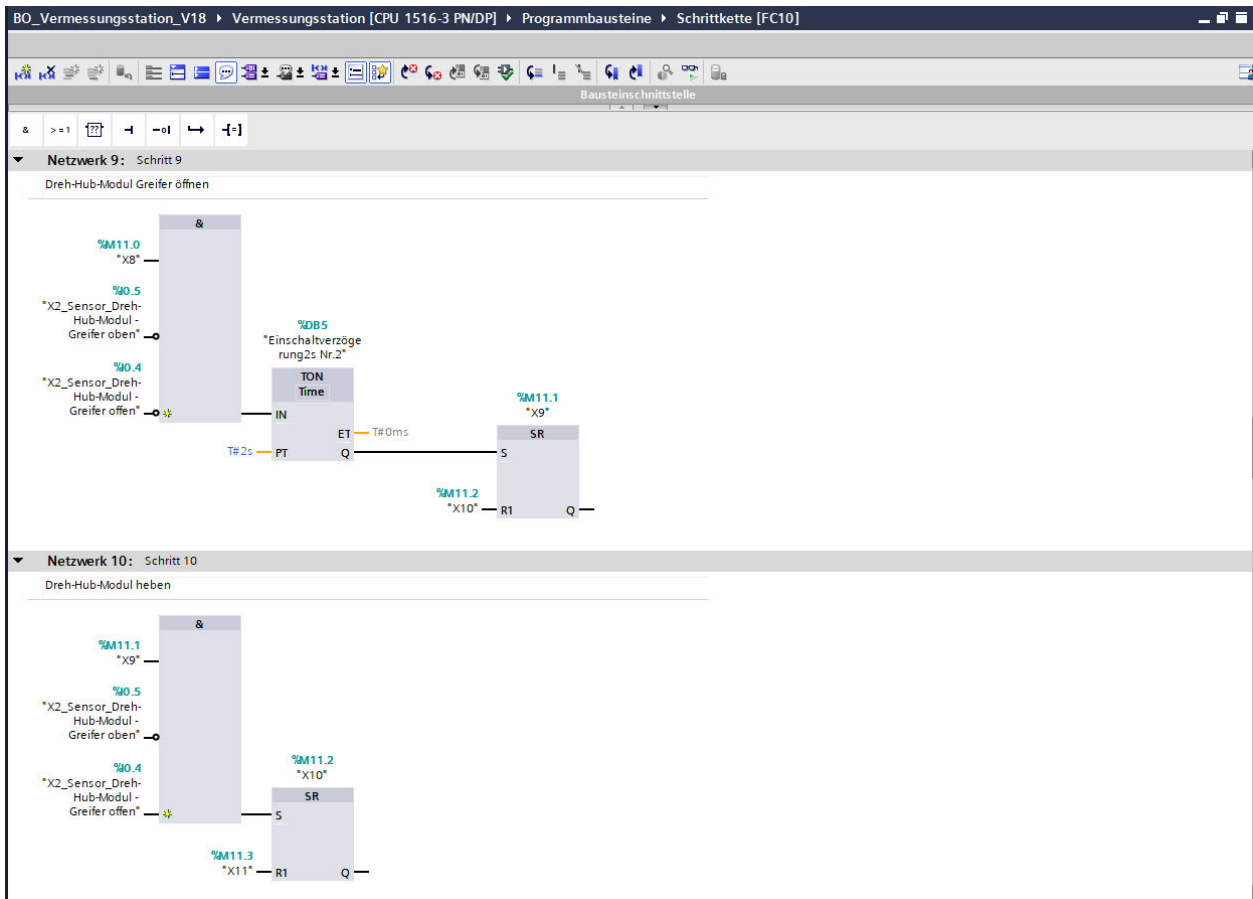
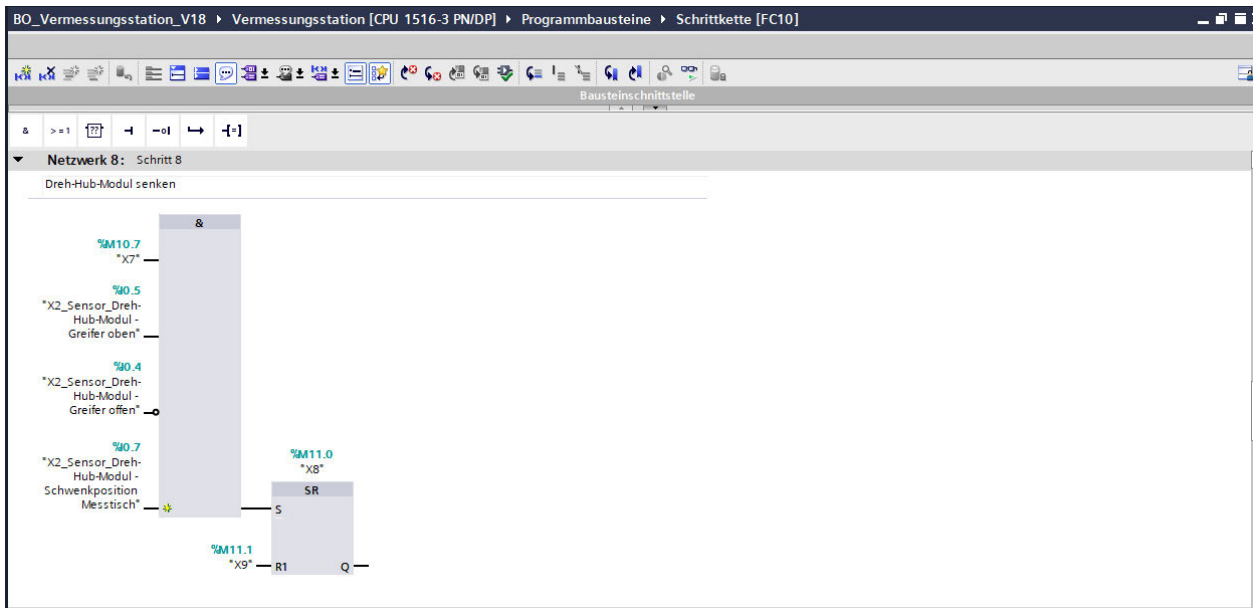
E: Grafset – physische Station

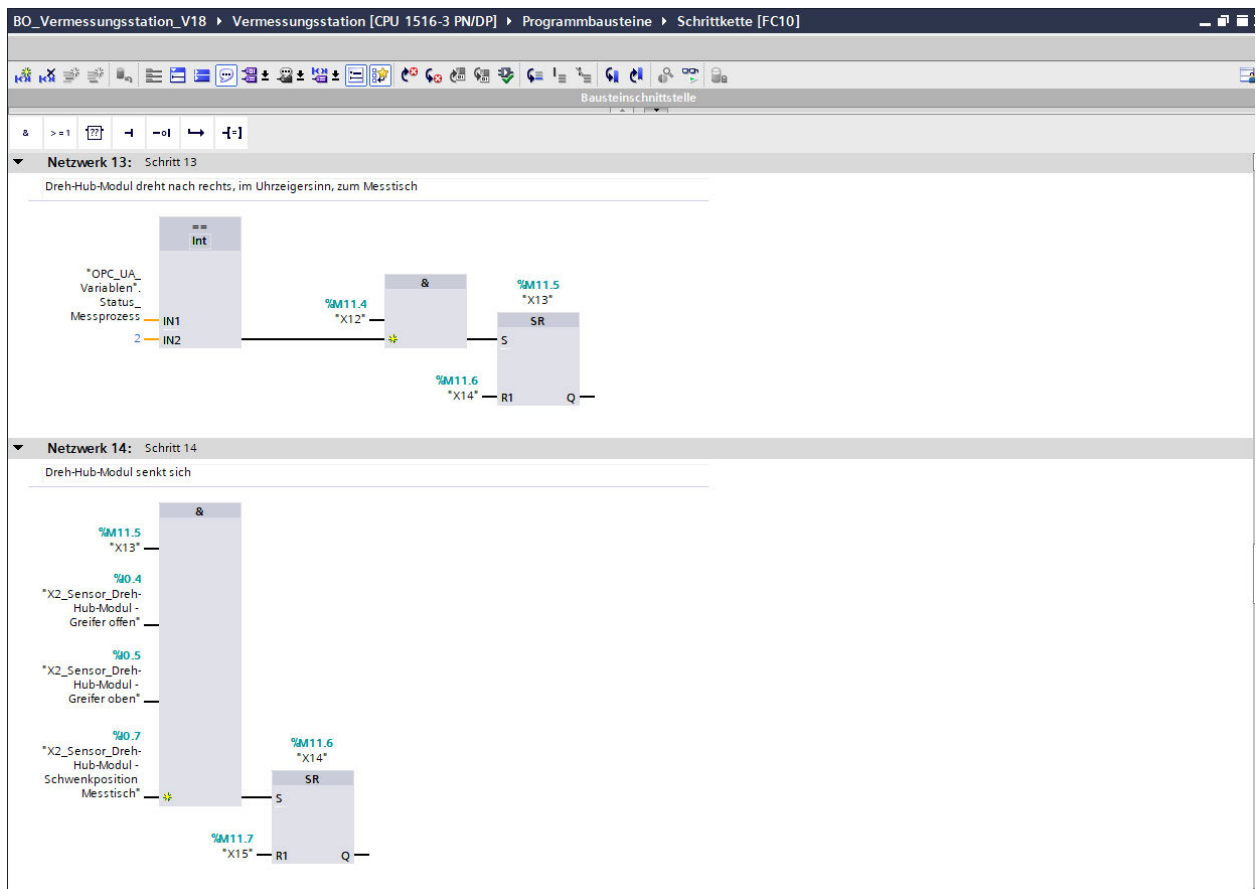
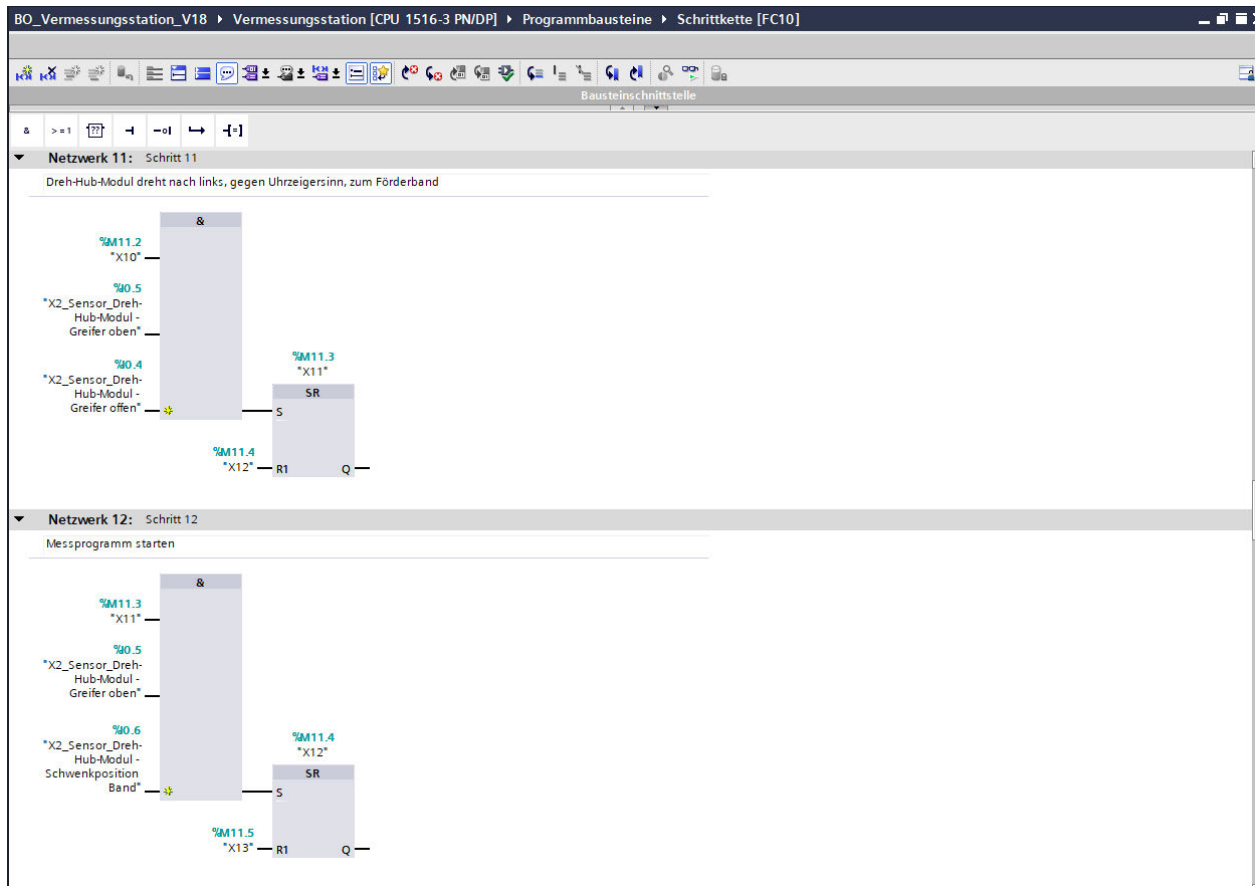


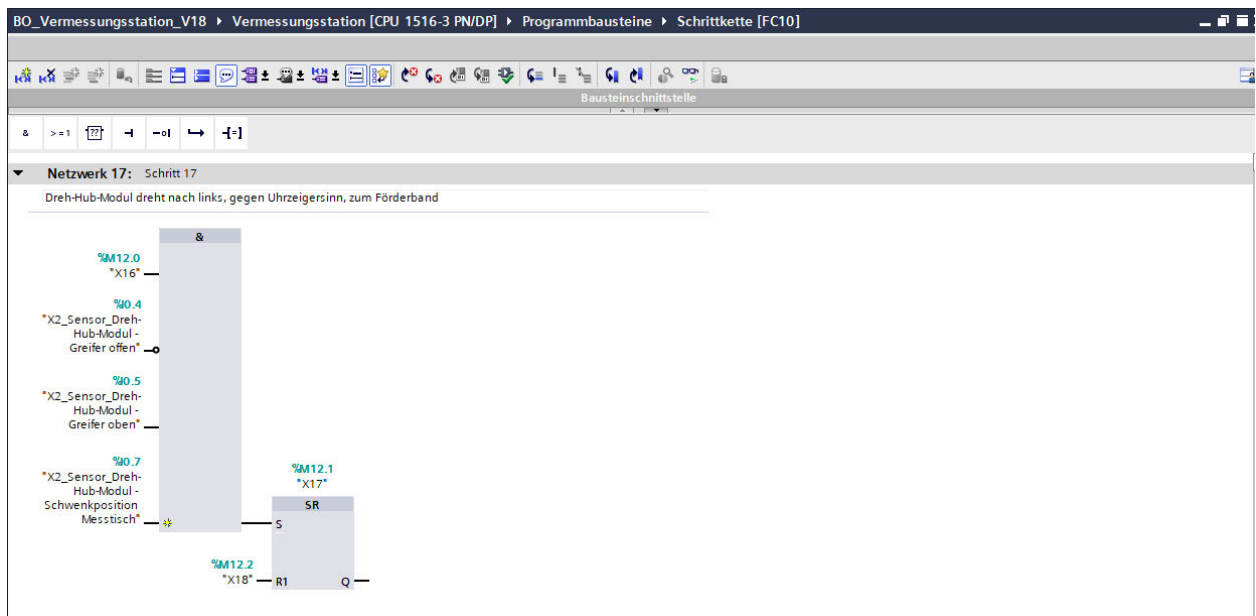
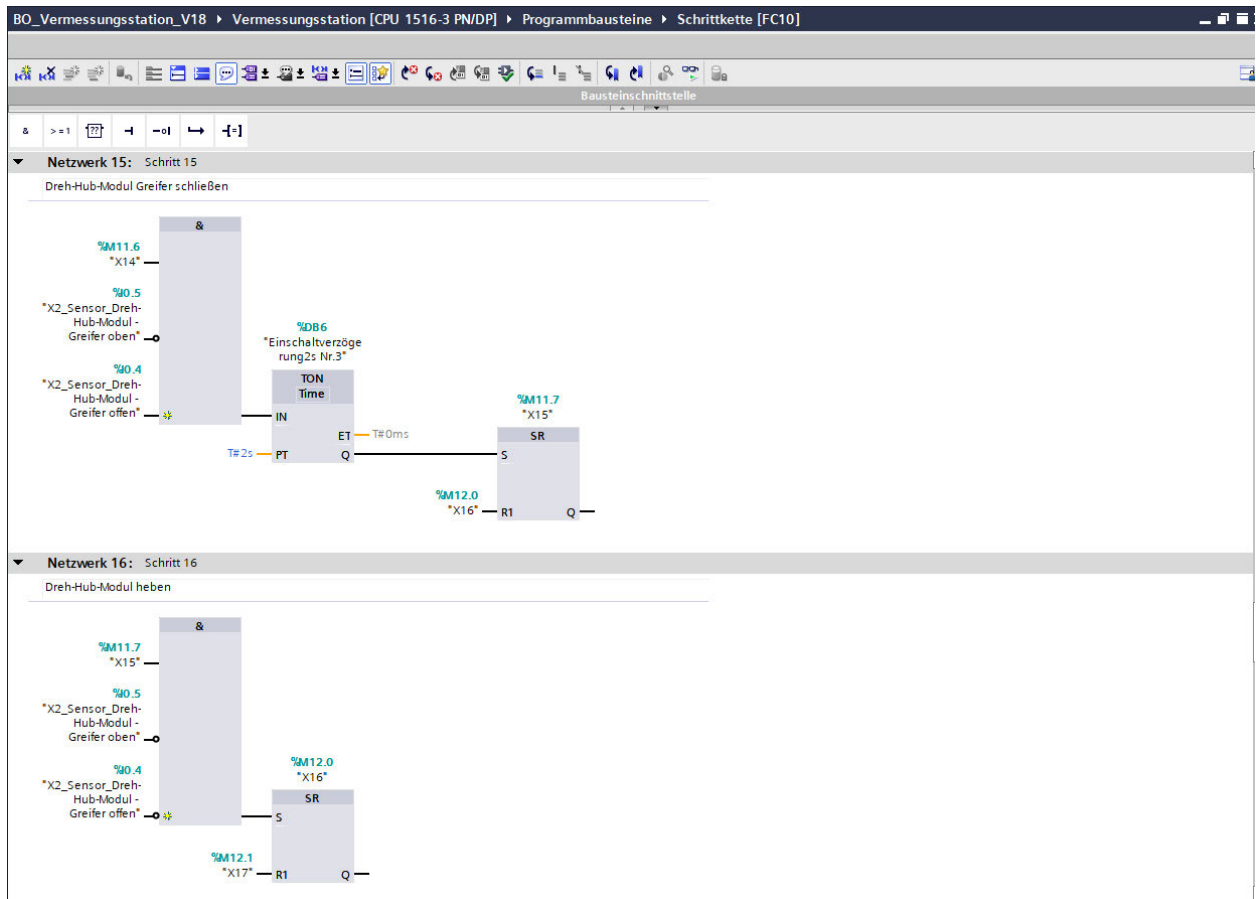
F: Physische Station – Schrittkette

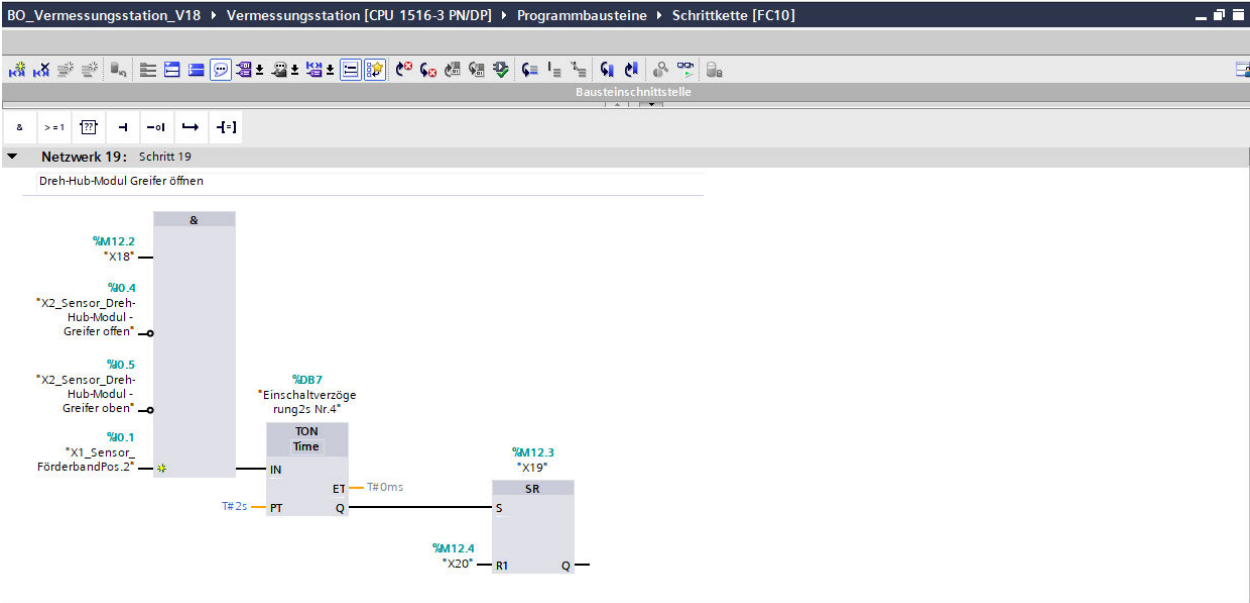
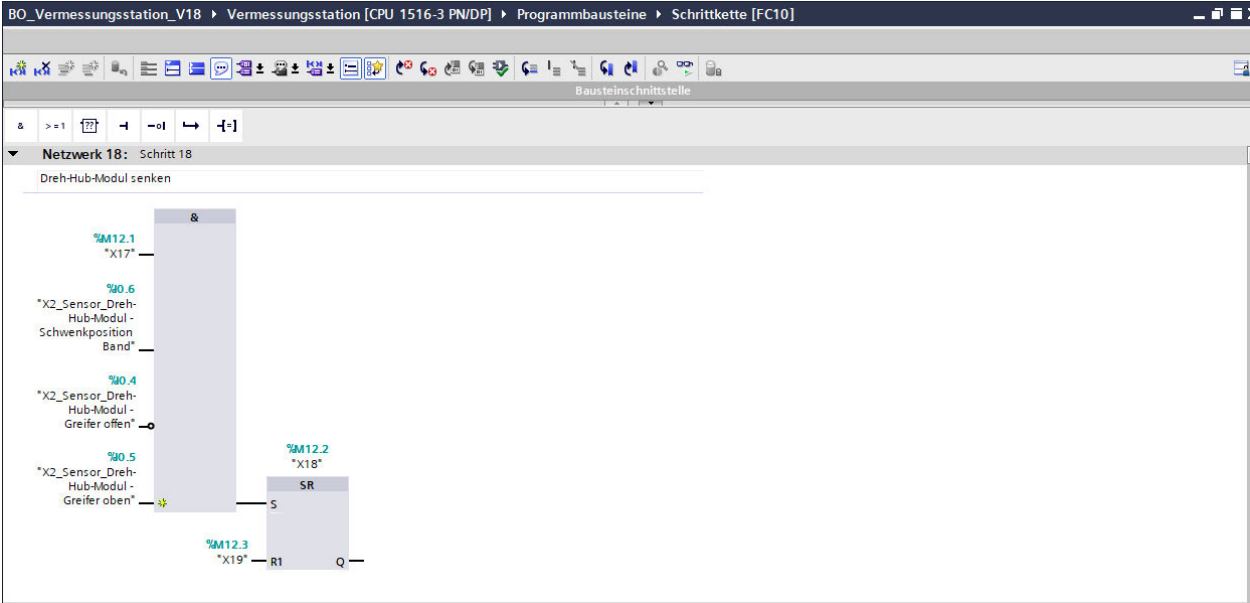


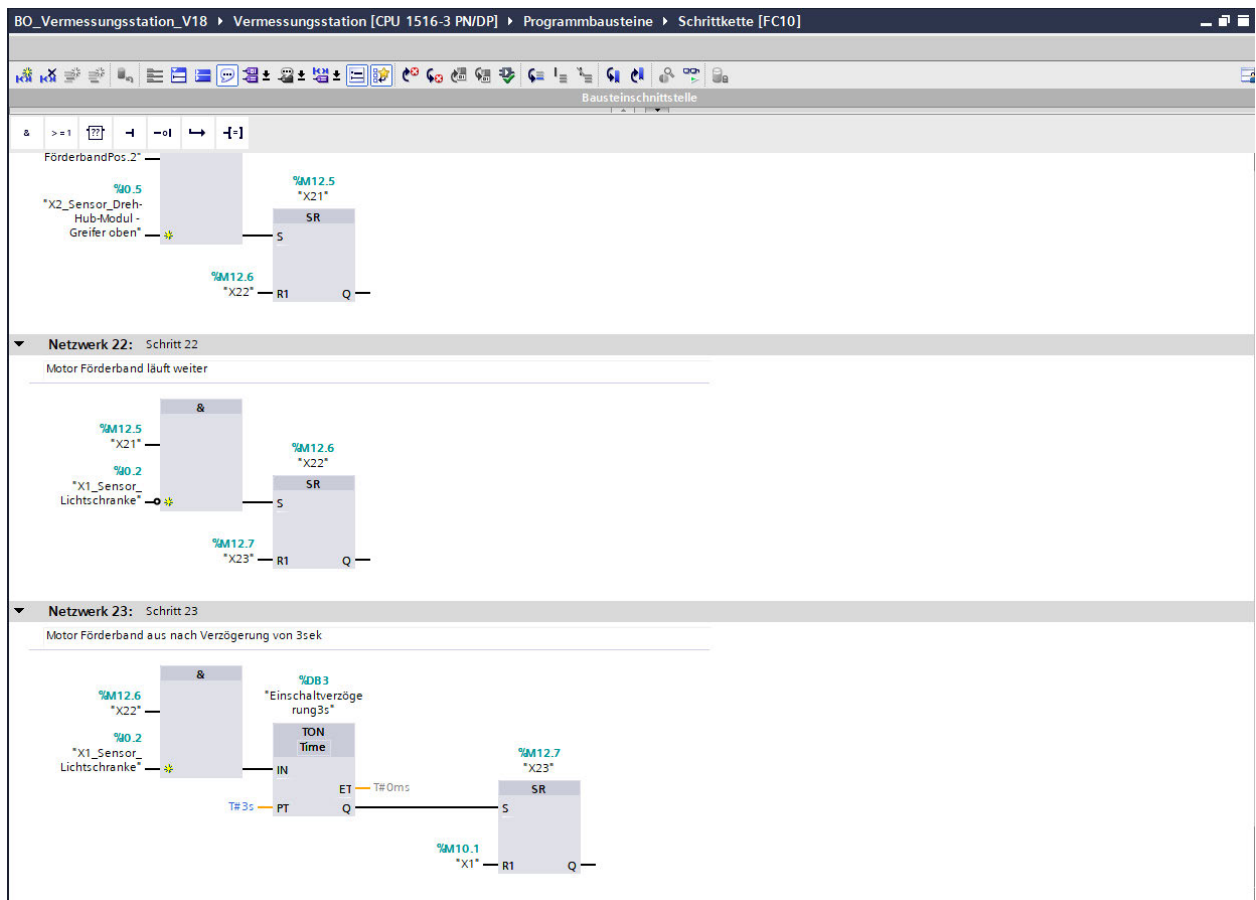
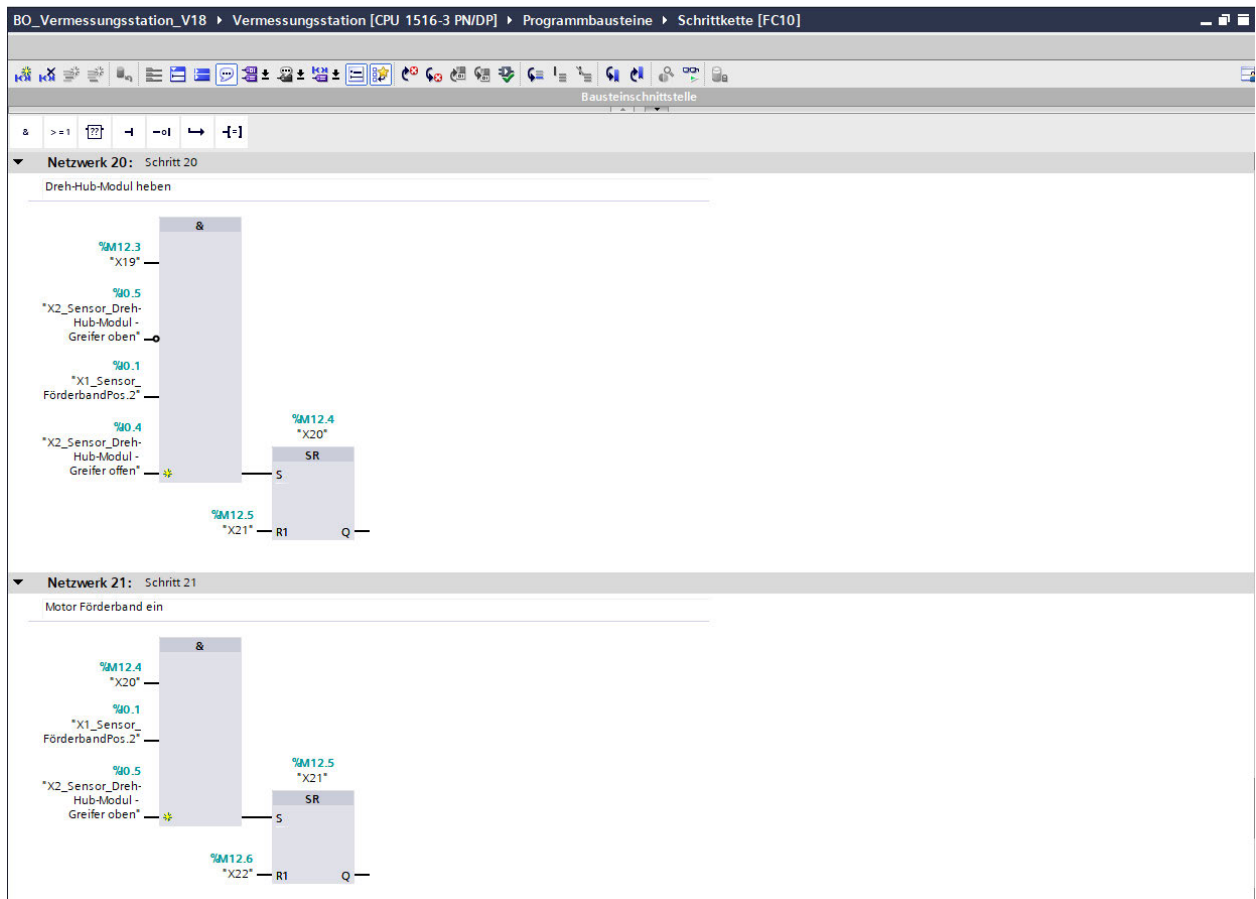












G: Physische Station – Befehlsausgabe

BO_Vermessungsstation_V18 > Vermessungsstation [CPU 1516-3 PN/DP] > Programmbausteine > Befehlsausgabe [FC120]

Bausteinschnittstelle

& >=1 ? ? -o- -o- -o- -o-

▼ Bausteintitel: Befehlsausgabe
Kommentar

▼ Netzwerk 1: Motor Förderband
Kommentar

▼ Netzwerk 2: Ventil Stopper
Kommentar

BO_Vermessungsstation_V18 > Vermessungsstation [CPU 1516-3 PN/DP] > Programmbausteine > Befehlsausgabe [FC120]

Bausteinschnittstelle

& >=1 ? ? -o- -o- -o- -o-

▼ Netzwerk 3: Start Messprozess
Kommentar

```

1 // Mit Schritt 12 wird der Messprozess gestartet
2 IF "X12" THEN
3
4     "OPC-UA-Variablen".Start_Messung := TRUE; // Start Messprozess über OPC-UA-client
5
6 END_IF;
7
    
```

▼ Netzwerk 4: Status Messprozess ändern
Kommentar

```

1 // Mit Schritt 14 wird der Messprozess Status geändert für Folgeprozess
2 IF "X14" THEN
3
4     "OPC-UA-Variablen".Status_Messprozess := 0; // Status Messprozess zurück auf "Bereit" setzen
5
6 END_IF;
7
    
```

