DHBW
Duale Hochschule
Baden-Württemberg
Ravensburg
Campus Friedrichshafen

Produktionsnetzwerk – Optimierung der Steuerung und Vernetzung der Lernfabrik

Studienarbeit T3200

Studiengang Elektrotechnik

Studienrichtung

Duale Hochschule Baden-Württemberg Ravensburg, Campus Friedrichshafen

von

Leo Gulde, Maximilian Langer

Abgabedatum:	14. Juli 2024
Bearbeitungszeitraum:	01.04.2024 - 15.07.2024
Matrikelnummer:	7981437, 9324548
Kurs:	TEA21
Betreuer der Dualen Hochschule:	Prof. Dr. -Ing. Thorsten Kever

Erklärung

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

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

> *Produktionsnetzwerk – Optimierung der Steuerung und Vernetzung der Lernfabrik*

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

Friedrichshafen, den 14. Juli 2024



Leo Gulde, Maximilian Langer

Kurzfassung

Diese Arbeit dokumentiert den zweiten Teil einer Studienleistung des fünften und sechsten Theoriesemesters. Die Gruppe der „Steuerung und Vernetzung“ ist von der Steuerungsseite aus die Schnittstelle der verschiedenen Projektgruppen, um im Optimalfall einen zusammenhängenden Prozess abzubilden.
Das konkrete Hauptziel der Studienarbeit der 6. Theoriephase war die Einführung des Manufacturing Execution System (MES). Weiterhin, wie auch schon in der vorhergehenden Theoriephase, soll die Gruppe den anderen Gruppen bei Steuerungsaufgaben zur Verfügung stehen.
Da der Fokus dieser Arbeit auf der Einführung des MES-System liegen soll, wurden dafür die folgende Schritte umgesetzt.
Zuerst mussten die Grundlagen erarbeitet werden und nach Wahl eines geeigneten Systems namens „Forcam Force Edge“ konnte mit den Vorbereitungen gestartet werden. Dazu gehörte eine Schulung seitens Forcam, sowie die Vorbereitungen für die Installation, welche im Anschluss mit einem Forcam Mitarbeiter getätigt wurde. Nachdem die aufgetretenen Probleme beseitigt wurden, konnte das System erfolgreich installiert werden, wodurch ab jetzt mit der Konfiguration gestartet werden kann.
Da ab einem bestimmten Zeitpunkt die Steuerung der Anlage Priorität hatte, wurde die MES-Einführung pausiert.
In Punkto Steuerung wurde eine Station mit der Entwicklungsumgebung eingerichtet, sodass die Teams die neuen Funktionen der Anlage selber implementieren und testen können. Im Anschluss wird ein Konzept für den Bandumlauf vorgestellt, samt Versionsverwaltung mit QR-Code Scannern. Da das Programm unübersichtlich und schwer nachvollziehbar ist, wurden die einzelnen Stationen erläutert und Lösungsmöglichkeiten vorgestellt.

Abstract

This work documents the second part of a coursework from the fifth and sixth theoretical semesters. From the control side, the group of „control and networking“ is the interface of the various project groups in order to ideally represent a coherent process.
The specific main goal of the coursework in the 6th theoretical phase was the introduction of MES. Furthermore, as in the previous theoretical phase, the group should be available to the other groups for control tasks.
Since the focus of this work is to be on the introduction of the MES system, the following steps were implemented.
First the basics had to be worked out and after choosing a suitable system called „Forcam Force Edge“ the preparations could begin. This included training from Forcam as well as preparations for the installation, which was then carried out with a Forcam employee. After the problems that occurred were eliminated, the system was installed successfully, which means you can now start with the configuration. Since the control of the system had priority at a certain point in time, the MES introduction was paused.
In terms of control, a station with the development environment was set up so that the teams can implement and test the new functions of the system themselves. A concept for belt circulation is then presented, including version management with QR code scanners. Since the program is confusing and difficult to understand, the individual stations were explained and possible solutions were presented.

Inhaltsverzeichnis

1 Einleitung **1**

2 Manufacturing Execution System **2**

2.1 Theoretische Grundlagen 2

2.1.1 Grundlagen der Produktionsanlagenvernetzung 2

2.1.2 Funktionsweise eines MES-Systems 4

2.1.3 Übersicht über Forcam MES 6

2.2 Vorbereitung 9

2.2.1 Schulung „How to connect a machine“ 9

2.2.2 Auswahl und Vorbereitung der Hardware und zu installierenden Software 12

2.3 Installation 14

2.3.1 Ubuntu und andere Add-Ons 14

2.3.2 Installation der Forcam Edge 18

2.4 Probleme und Herausforderungen 21

3 Steuerung der Anlage **23**

3.1 Versionsmanagement 23

3.2 Steuerung der Stationen 25

3.3 Neuaufbau Steuerungsablauf 28

3.3.1 Belt feeding 28

3.3.2 Bin Picking 30

3.3.3 Umleger 1 31

3.3.4 Handarbeitsplatz 31

3.3.5 End of Line 32

3.3.6 Demontage/Umleger 2 32

3.4 Human Machine Interface 33
3.4.1 Human Machine Interface allgemein 33
3.4.2 Profinet Anschluss des Human Machine Interface 33
3.4.3 Stopper Ansteuerung über Human Machine Interface 34

4 Zusammenfassung **35**
4.1 Manufacturing Execution System 35
4.2 Steuerung der Anlage 37

Literaturverzeichnis **38**

Abkürzungsverzeichnis **40**

Abbildungsverzeichnis **41**

1 Einleitung

Im Produktionstechnischen Zentrum (PTZ) der DHBW Ravensburg soll eine Anlage gemäß den Prinzipien der Industrie 4.0 entstehen. Diese Studienarbeit befasst sich mit der Optimierung der Steuerung und Vernetzung dieser Lernfabrik. Das geplante Vorgehen wird in zwei Hauptkapiteln detailliert beschrieben.

Im ersten Hauptkapitel wird die Einführung eines Manufacturing Execution Systems (MES) behandelt. Hierbei liegt der Fokus auf der Auswahl und Implementierung des Forcam Force Edge Systems, welches bereits im Institut für Weiterbildung, Wissens- und Technologietransfer (IWT) im Einsatz ist. Die notwendigen Vorbereitungen umfassten Schulungen, Hardwareauswahl und die Installation der erforderlichen Software, wobei diverse Probleme und Herausforderungen bewältigt werden mussten.

Das zweite Hauptkapitel widmet sich der Steuerung der Anlage. Es umfasst das Versionsmanagement der Steuerungssoftware, sowie die detaillierte Programmierung und Anpassung der Steuerungslogik für die verschiedenen Stationen der Anlage. Hierfür wurden die vorhandenen Funktionalitäten beschrieben und analysiert.

Am Schluss sollen die Hauptarbeitsschritte und Ergebnisse zusammengefasst und ein Ausblick auf zukünftige Verbesserungen und Weiterentwicklungen gegeben werden.

2 Manufacturing Execution System

2.1 Theoretische Grundlagen

2.1.1 Grundlagen der Produktionsanlagenvernetzung

Langfristig gilt es den Automatisierungsgrad immer weiter zu erhöhen. Dieser Grad gibt an, wie stark eine bestimmte Aufgabe oder Arbeitsablauf automatisiert ist. Dazu gehört, dass die manuelle Bedienung der Anlage durch den Bediener immer weiter gesenkt werden soll, beziehungsweise die Prozesse von allein ausgeführt werden. So sollen beispielsweise die schon vorhandenen Produktionsvarianten in Form von eingelesenen Auftragspapieren parallel produziert werden. Damit der Bediener nicht laufend die Varianten am Human Machine Interface (HMI) ändern muss, bedarf es ein übergeordnetes System, welches die Abläufe (welche in der Speicherprogrammierbare Steuerung (SPS) gesteuert werden) koordiniert. Auch in Hinblick darauf, die Produktionsanlage des Produktionstechnischen Zentrums (PTZ) mit der Anlage des IWT über den mobilen Husky Roboter zu koppeln, müssen grundlegende Aspekte wie Material-/ und Bauteilverfolgung berücksichtigt werden.

Klassischerweise findet eine solche Bauteilverwaltung, beziehungsweise Steuerung von Geschäftsprozessen und Datenverarbeitung in einem Enterprise Resource Planning (ERP)-System statt. Bezogen auf unsere Anlage soll hier später die Auftragsabwicklung & Archivierung stattfinden, genauso wie die angesprochene Bauteilverwaltung. Ein sehr verbreiteter Anbieter solch einer ERP-Software ist die SAP SE. Im Produktionstechnisches Zentrum (PTZ) beschäftigt sich eine Gruppe mit der Einführung des SAP. Da das SAP-System aber nicht direkt mit der Anlage/-Steuerung kommunizieren kann, benötigt man eine Schnittstelle zwischen den zwei Komponenten. Hierfür kann man sich nochmal die Automatisierungspyramide in Betracht ziehen:

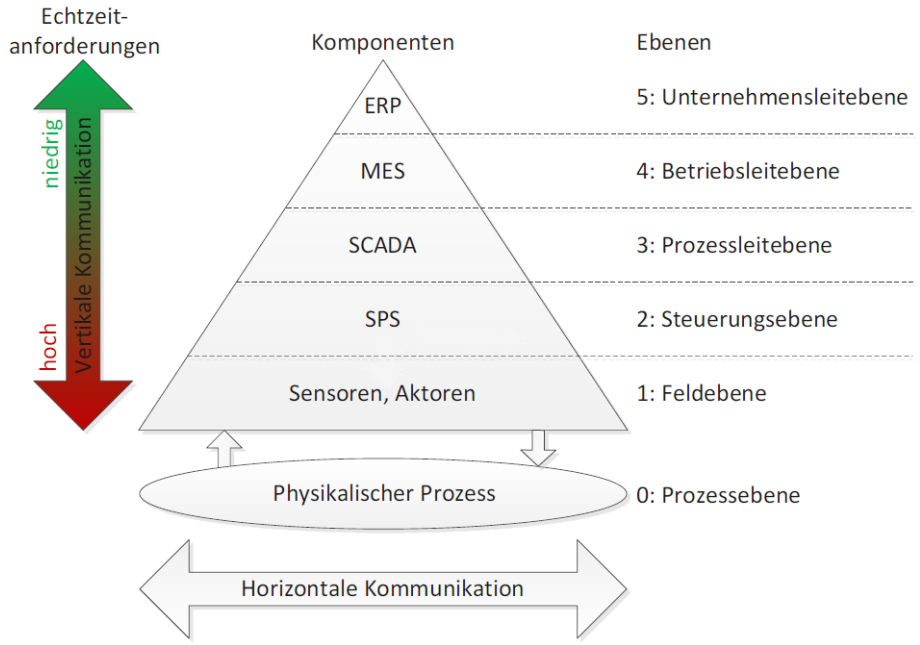


Abbildung 2.1: Automatisierungspyramide (Dürkop, 2016)

Hier ist ersichtlich, das zwischen der Steuerungsebene zwei Ebenen liegen. Zum Einen die Prozessleitebene, welche die Supervisory Control and Data Acquisition (SCADA)-Systeme beinhaltet (bei uns HMI). Zum Anderen (wichtiger für den Datenaustausch in höhere Ebenen) die Betriebsleitebene, welche das MES-System umfässt. Die Steuerungsgruppe soll hierbei ein solches MES-System einführen und somit den Datenaustausch nach „oben“ ermöglichen.[1]

[1] Vgl. Dürkop, 2016, S.8ff.

2.1.2 Funktionsweise eines MES-Systems

Das MES-System ist im Gegensatz zum ERP für die maschinennahe Steuerung zuständig.

Es spielt eine wesentliche Rolle in der Umsetzung von Industrie 4.0. Dabei ermöglicht ein MES-System die Überwachung, die Steuerung sowie die Optimierung von Produktionsprozessen in Echtzeit.

Hauptfunktionalitäten eines MES-Systems:

- **Produktionsplanung und /-steuerung**

  Die Planung und Steuerung der Produktion kann durch ein MES-System unterstützt werden. Das gelingt durch die Erstellung von Produktionsplänen und geordneten Abläufen innerhalb des System. Zusätzlich können Produktionsaufträge aus höheren Ebenen überwacht werden. Durch das System kann schnell auf die Produktion zugegriffen werden und Anpassungen durchgeführt werden.

- **Betriebsdatenerfassung**

  Ein weiterer Punkt, welcher vom MES-System übernommen wird, ist die Erfassung von Daten (auch Echtzeitdaten) der Produktionsanlage. Zu diesen Daten gehören in allgemeinen Fällen Maschinenlaufzeiten, Produktionsmengen, Ausschussraten oder auch Energieverbräuche. Welche dieser Werte für uns relevant sind (für eine Datenerfassung), muss im einzelnen geklärt werden.

- **Qualitätsmanagement**

  Das MES-System kann auch dazu genutzt werden, die Qualität der produzierten Teile zu überwachen und somit eine frühzeitige Erkennung von Qualitätsproblemen zu erreichen. Man kann Produktionschargen zurückverfolgen und durch die Dokumentation Maßnahmen zur Qualitätssicherung treffen. Gerade in unserem Fall kann man durch die in Zukunft an die Anlage angebundene End of Line Prüfung die Qualität hochhalten, in Kombination mit dem MES-System.

- **Materialmanagement**

  Gerade in Bezug auf die Bauteilverfolgung kann von dem MES der Materialfluss verwaltet werden. Dazu gehört die Nachverfolgung sowie Bestandsaufnahme von Bauteilen. In unserem Fall wird dies gerade bei der Kopplung der beiden Anlagen eine Rolle spielen.[2]

[2] Vgl. Kletti und Deisenroth, 2021.

2.1.3 Übersicht über Forcam MES

Im Labor des IWT besteht bereits ein MES-System. Naheliegend ist es also, gerade im Hinblick auf die Kopplung der beiden Produktionsanlagen, im PTZ das gleiche System einzuführen, um die geplante Kopplung der beiden Produktionsstandorte so einfach wie möglich zu halten.

Das System, welches im IWT-Labor zum Einsatz kommt, stammt von der Firma Forcam und nennt sich „FORCE EDGE CONNECT“.

Das Ziel der Lösung „EDGE CONNECT“ ist es, die Digitalisierungspotenziale von Maschinen oder Anlagen auszuschöpfen, ganz unabhängig von Hersteller oder Alter der Maschine.
Auch Maschinen, die bereits in Betrieb sind, können nachträglich mithilfe der „EDGE“ smarter gemacht werden.

Schaut man sich den generellen Aufbau an, erkennt man die folgenden wichtigsten Elemente:

Maschinen Konnektivität & Modell
Dieser Bereich ist Kern der Maschinen Konnektivität. Er beinhaltet zum einen den *MDC Layer*, welcher für die Verbindung der Maschinen mit dem System zuständig ist. Zusätzlich werden hier die passenden Plugins für die Kommunikation mit der Steuerung ausgewählt und Maschinendaten sowie Netzwerkeinstellungen konfiguriert, welche anschließend an den Composition Layer weitergeleitet werden. Zusätzlich ist hier der *DNC Layer* enthalten, welcher die Maschinen mit der Numeric Control (NC)-Versorgung verbindet. Diese Versorgung kümmert sich um die Bereitstellung und Verwaltung von numerischen Steuerprogrammen. Der *Composition Layer* leitet dann die logischen Maschinenzustände ab, nutzt die Skriptsprache, um Ereignisse aus Signalkombinationen abzuleiten und Werte in die Steuerung der Maschine zu schreiben.

Plugins

Sind für Kommunikationsverbindung mit Steuerungen zuständig, wobei Kommunikationsprotokolle wie Open Platform Communication Unified Architecture (OPC UA) oder Message Queuing Telemetry Transport (MQTT) unterstützt werden. Dabei sind die Machine Data Collection (MDC)-Plugins für das Auslesen sowie Schreiben von Maschinensignalen zuständig, während die Distributed Numeric Control (DNC)-Plugins für das Auslesen und Übertragen von NC-Dateien verwendet wird.

Data Lake

Hier werden alle Daten gespeichert, samt Schreiboperationen, Konfigurationen und übertragenen NC-Dateien. Die Daten werden hierbei über die Edge API bereitgestellt.

Edge API

Die Edge API dient als Schnittstelle zu sogenannten 3rd Party Systems, also übergeordneten Systeme. Das können andere MES oder ERP-Lösungen sein, wobei die Verbindung mittels HTTP/Rest (Netzwerkprotokoll für Übertragung über das Web) oder MQTT hergestellt wird.

Edge Komponenten

Weitere Komponenten sind zum einen die *EDGE Konfiguration*, welche die Edge Knoten verwaltet und die Signale mehrerer Maschinen bündelt. Zudem gibt es das *Machine Repository*, welches Vorlagen für die standardisierte Verbindung von Maschinen ermöglicht, was die Konfigurationszeit reduziert.[3]

[3]Vgl. FORCAM GmbH, 2021, S.6ff.

Die Verknüpfung der genannten Elemente ist in der folgenden Abbildung dargestellt.

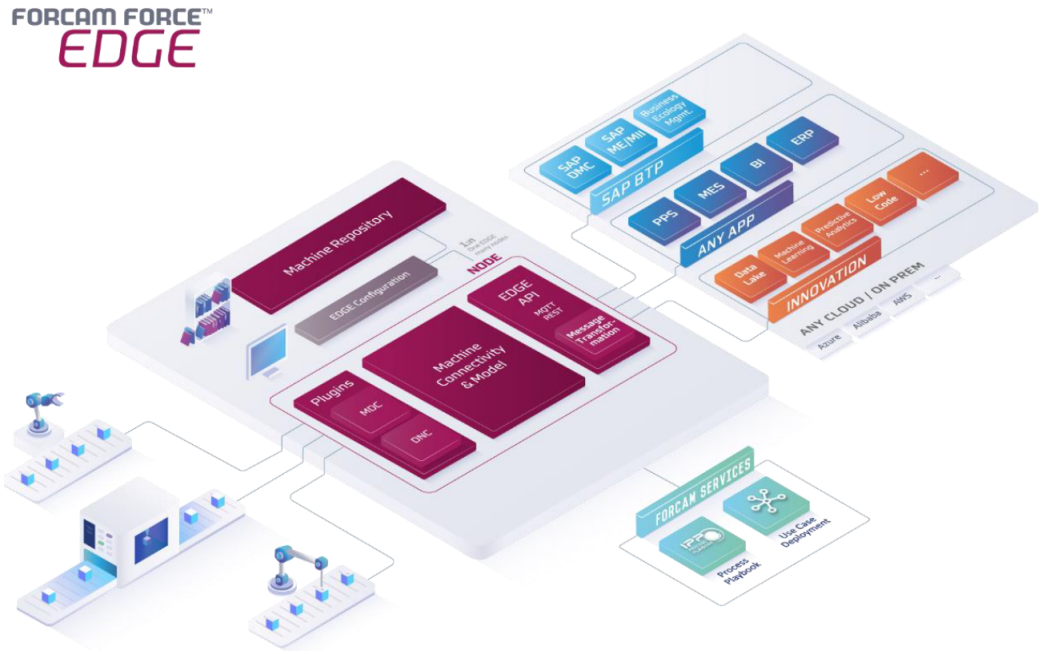


Abbildung 2.2: Edge Aufbau (FORCAM GmbH, 2021)

2.2 Vorbereitung

2.2.1 Schulung „How to connect a machine“

Um mehr über das von Forcam angebotene MES-System „FORCAM FORCE EDGE“ zu erfahren, bot uns die Firma Forcam eine Schulung zum Thema „How to connect a Machine“ an, welche direkt von einem Forcam Mitarbeiter durchgeführt wurde.

Thema der Schulung sollte weniger das MES System an sich sein, also welche Funktionalitäten MES System besitzt oder wie es im Detail aufgebaut ist.
Vielmehr lag der Fokus im Datenaustausch zwischen Maschine und MES-System. Die Schulung war eine Mischung aus Wissensübermittlung des Mitarbeiters bezüglich der Software und Programmierübungen, um mit dem System klar zu kommen.

Zu Beginn wurde auf die grundlegende Architektur des Systems eingegangen. Der Aufbau wurde im vorherigen Abschnitt schon beschrieben, deshalb wird dieser hier nicht mehr erläutert.
Anschließend wurde auf die Verbindung zwischen dem System und der Anlage, genauer gesagt auf die Netzwerkverbindung eingegangen. Für die Verbindung sind Angaben wie Internet-Protokoll (IP)-Adresse, Connection timeout, Bus Typ, Geräte Typ (z.B. in unserem Fall S7-1500), sowie die genutzte Portnummer zu konfigurieren.

Der Signalaustausch findet über ein Skript statt. Die Sprache, welche für das Skript zu verwenden ist, basiert auf Pascal. Pascal ist eine systematische Programmiersprache, welche vor allem gut von Anfänger aufgrund der leicht verständlichen Syntax verstanden werden kann. Vermutlich zielt das System genau darauf ab, den Datenaustausch ohne fortgeschrittene Programmierkenntnisse zu realisieren.

Das Skript besteht im Grunde aus den folgenden Elementen, wobei die Anordnung dieser immer gleich bleiben sollte, um Fehler zu vermeiden oder diese schnell zu finden:

- Header
- Deklaration der Variablen
- Initialisierung
- Zeit Funktionen
- Status Definition
- Counter Definition
- Übermittlung von „Status Events“
- Übermittlung von „Quantity Events“
- Log[4]

Als uns das grundlegende Konzept sowie der Aufbau des Skriptes näher gebracht wurde, sind wir eine beispielhafte Konfiguration des Systems durchgegangen.

[4]Vgl. FORCAM GmbH, 2022.

Um einen Eindruck von der Konfiguration zu bekommen, erhielt jeder Teilnehmer einen Zugang zu einem Schulungssystem. Das Training spiegelt die Konfiguration wieder, jedoch ohne eingesetzte Hardware. Die Aufgaben der Schulung waren wie folgt:

1. Zu Beginn war die Aufgabe gegebene Signale im Skript zu deklarieren mit Name, Adresse und Datentyp.

2. Anschließend sollte das Skript um gegebene Status Informationen erweitert werden. Die Status Definitionen beziehungsweise Status Änderungen werden über einfache Abfragen realisiert. Je nach Abfrage soll ein „state“ (in dem Fall ein Produktion Status) gesetzt werden.

3. Die dritte Aufgabe bestand darin ein sogenannten „quantity reporting“ zu erstellen. Dazu sollen die bereits im Programm vorhandenen Counter („good_parts_counter“ & „bad_parts_counter“) genutzt werden. Diese beiden Variablen sollen den Anzahl der guten und schlechten Teile speichern.

Die Schulung wurde als sehr positiv empfunden, da sie einen recht guten Überblick über die Anbindung eines MES an die Steuerung bietet.
Wie schon erwähnt liegt hier der Fokus gezielt auf der Anbindung an die Steuerung und weniger an dem Aufbau & Nutzen eines MES.

Als die Schulung erfolgreich abgeschlossen war, blieb der Kontakt mit Forcam weiterhin bestehen und es wurden die Vorbereitung für die Installation getätigt.

2.2.2 Auswahl und Vorbereitung der Hardware und zu installierenden Software

Da das MES-System von Forcam spezielle Installationsvoraussetzungen erfordert, mussten zunächst einige Vorbereitungen getroffen werden.
Als Hilfe erhielten wir von Forcam ein Dokument mit den „System Requirements“. Dazu gehören einerseits die Hardwarevoraussetzungen an den Rechner, auf welchem das System installiert wird.

RAM	≥ 16 GB
CPU	– Quad-core processor >2 Ghz
Disc space	– 90 GB free SSD disc space ⓘ This is the bare minimum requirement without additional components. In case Data Lake is used, refer to chapter 6.1 for additional requirements.

Abbildung 2.3: Hardwarevoraussetzungen (FORCAM GmbH, 2024)

Bei der Suche nach einem geeigneten Rechner, welcher die Voraussetzungen erfüllt, kam optimalerweise der PC an der End of Line (EoL)-Prüfung in Frage. Da der PC direkt an der Anlage platziert ist, eignet er sich auch optimal für die Steuerung der Anlage über das MES. Der Rechner für die EoL-Prüfung ist großzügig dimensioniert, somit kann dort (nach Absprache mit zuständigen Team) das MES parallel zur EoL-Prüfung arbeiten.

Nachdem die passende Hardware gefunden wurde, mussten vor der eigentlichen Installation des MES noch einige Installationen vorgenommen werden:

Da „Force Edge Connect“ eine container-basierte Anwendung ist, kann sie nicht auf einem Windows-Betriebssystem installiert werden. Stattdessen kann es in einer Kubernetes Umgebung oder einer Docker Umgebung installiert werden (beides Open-Source Plattformen zur Verwaltung & Ausführung von container-basierten Anwendungen).

Eine Docker Umgebung, welche von Forcam vorgeschlagen wird, ist das Betriebssystem namens „Ubuntu“.
Ubuntu ist eine Linux Distribution, welche das Ziel hat, dem Nutzer ein leicht zu bedienendes Betriebssystem zu bieten.

2.3 Installation

2.3.1 Ubuntu und andere Add-Ons

Da auf dem Rechner Windows 11 installiert ist war klar, dass das Ubuntu System auf einer Virtuelle Maschine (VM) installiert werden muss.

Bei der Suche nach einer Verwaltungsplattform solcher VM's, entschieden wir uns für den Hyper V Manager.
Dieser ist unter Windows 11 Pro standardmäßig vorhanden und muss lediglich aktiviert werden. Das kann einerseits über die Option „Windows-Funktionen aktivieren oder deaktivieren“ oder über die Konsole mit dem Befehl „Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Hyper-V -All“ erledigt werden.[5]

Die Installation von Ubuntu kann direkt im Hyper-V Manager durchgeführt werden mithilfe des „Quick-Create Tools“. Dort kann man die passende Version auswählen und eine virtuelle Maschine erstellen. Wir entschieden uns für die Long Term Support (LTS)-Version von Ubuntu, da diese wie der Name schon verrät länger als andere Versionen mit Updates/Fehlerkorrekturen versorgt wird.[6]

[5] Vgl. Canonical Ltd., 2024.
[6] Vgl. ubuntu Deutschland e.V., 2024.

Innerhalb des Hyper-V Managers gibt es eine Liste der erstellten VM's. Zusätzlich hat man die Möglichkeit Prüfpunkte zu erstellen, um bei Problemen innerhalb des Systems wieder auf den alten Stand zurückkehren kann.

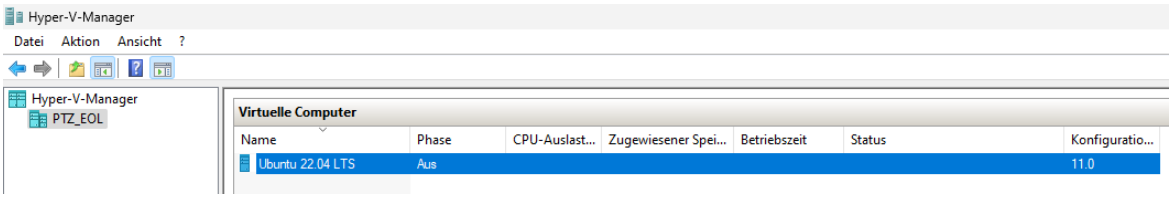


Abbildung 2.4: Virtuelle Maschine Ubuntu in VMware

Wir haben zwei Prüfpunkte erstellt, ein Stand der VM mit den Installationsdateien von für das Forcam MES. Der zweite Stand ist die fertige Installation des Systems.

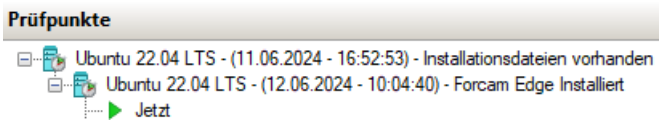


Abbildung 2.5: Prüfpunkte von Virtueller Maschine in VMware

Öffnet man das Ubuntu System erscheint, nach dem Anmeldebildschirm. Der Username lautet „mes“ und das Passwort wie das Passwort des Rechners „qwertz“.

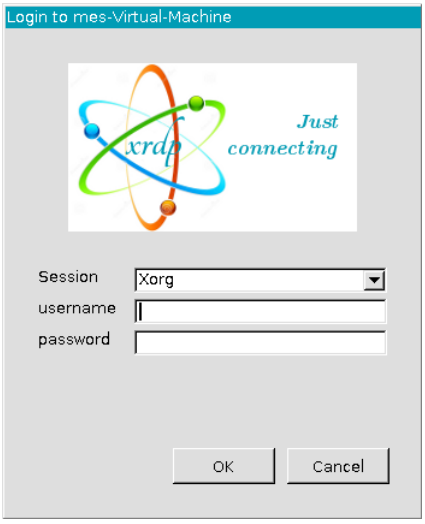


Abbildung 2.6: Anmeldebildschirm bei Ubuntu

Nach der Anmeldung erscheint wie auch von Windows-Betriebssystemen gewohnt eine Oberfläche mit Icons für verschiedene Anwendungen. Von der Bedienung ähnelt es zwar einem Windows Betriebssystem, von den Funktionen/Sicherheit unterscheidet es sich jedoch teilweise nicht unerheblich.

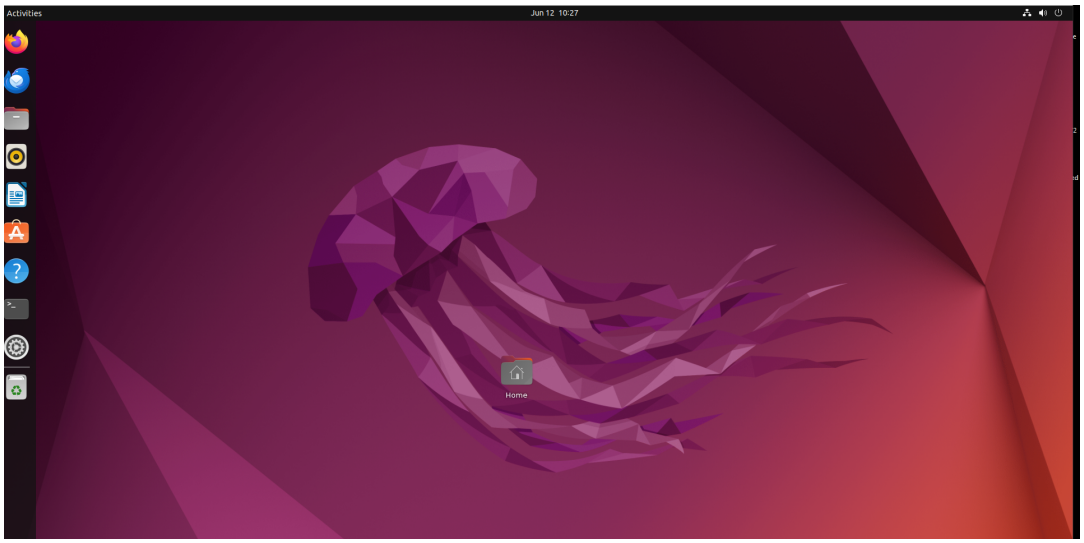


Abbildung 2.7: Ubuntu Oberfläche

Gerade in Hinblick auf die Nachfolger Gruppe der Studienarbeit ist zu erwähnen, dass des Öfteren beim Starten der VM ein hellblauer Screen erscheint.
Dies bedeutet keineswegs ein Fehler oder Systemabsturz sondern, sondern dass im Hintergrund noch eine andere Session offen ist. Beheben lässt sich dieses „Problem“ indem man die erweiterte Sitzung startet (auf Abbildung 2.8 das zweite Element von rechts) und wieder schließt. Danach kehrt man wieder auf den Startbildschirm von Ubuntu zurück.



Abbildung 2.8: Kontrollleiste in der virtuellen Maschine

Über die anderen Kontrollelemente lässt sich der Betriebszustand der VM anpassen.

Nachdem Ubuntu erfolgreich installiert wurde, mussten noch einige Add-Ons installiert werden.
Dazu gehören Installationen, die für das MES notwendig sind (Java JDK), aber auch Anwendungen für die Installation an sich, da diese remote stattfand (Putty & WinSCP, Filezilla, TeamViewer).
Waren die ganzen Tools installiert, kann mit der Installation des eigentlichen MES-System gestartet werden.

2.3.2 Installation der Forcam Edge

Die Installation des MES muss von einem Forcam Mitarbeiter durchgeführt werden, da eine Menge Konfigurationen getätigt werden müssen.

Dazu gehört zum Beispiel die Netzwerkverbindung zwischen Rechner und der VM. Hierbei wurde noch eine Änderung vorgenommen, auf welche später noch eingegangen wird.

Des Weiteren musste der VM noch Arbeitsspeicher, Rechenleistung (Prozessoren) und Zugriff auf Speicher zugewiesen werden. Dies wird in Kapitel 2.4 auch noch ein Thema.

Soweit ist die Installation ohne große Probleme und Verzögerungen abgelaufen. Die Installation lief zwar nicht schnell ab, dass war aber der Tatsache geschuldet, dass die Internetverbindung des PC zu diesem Zeitpunkt über Wireless-LAN (WLAN) war.

Als das System fertig installiert war und der Forcam Mitarbeiter die letzten Konfigurationen getroffen hat, wurde noch thematisiert wie man das System aufruft.

Das gelingt über den Browser, wobei man zwei Möglichkeiten hat. Entweder über die Adresse oder über den „localhost“.

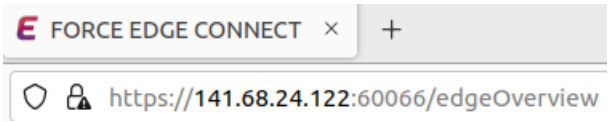


Abbildung 2.9: Aufrufen der Edge über die Adresse

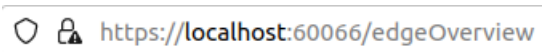


Abbildung 2.10: Aufrufen über localhost

Ist die VM korrekt mit dem Netzwerk des PC verbunden, kann das MES (sobald die VM gestartet ist) auch über den Browser des PC und nicht nur über den Browser innerhalb der VM zugegriffen werden.

Ruft man die Edge auf, gelangt man zu einem Anmeldebildschirm mit folgenden Anmeldedaten:
Username: edgeAdmin
Passwort: Force2020&k

Die Oberfläche ist ziemlich intuitiv gestaltet. In dem Slide-In Menü kann man sich durch das MES navigieren. Dazu gehört die Hauptseite für die Konfiguration, das Nutzer Management, die Lizenzübersicht, Profilübersicht und das Monitoring.
Auf der Startseite kann man dann direkt mit der Konfiguration starten, indem man neben der Übersicht der angelegten Nodes auch neue Nodes hinzufügen kann.

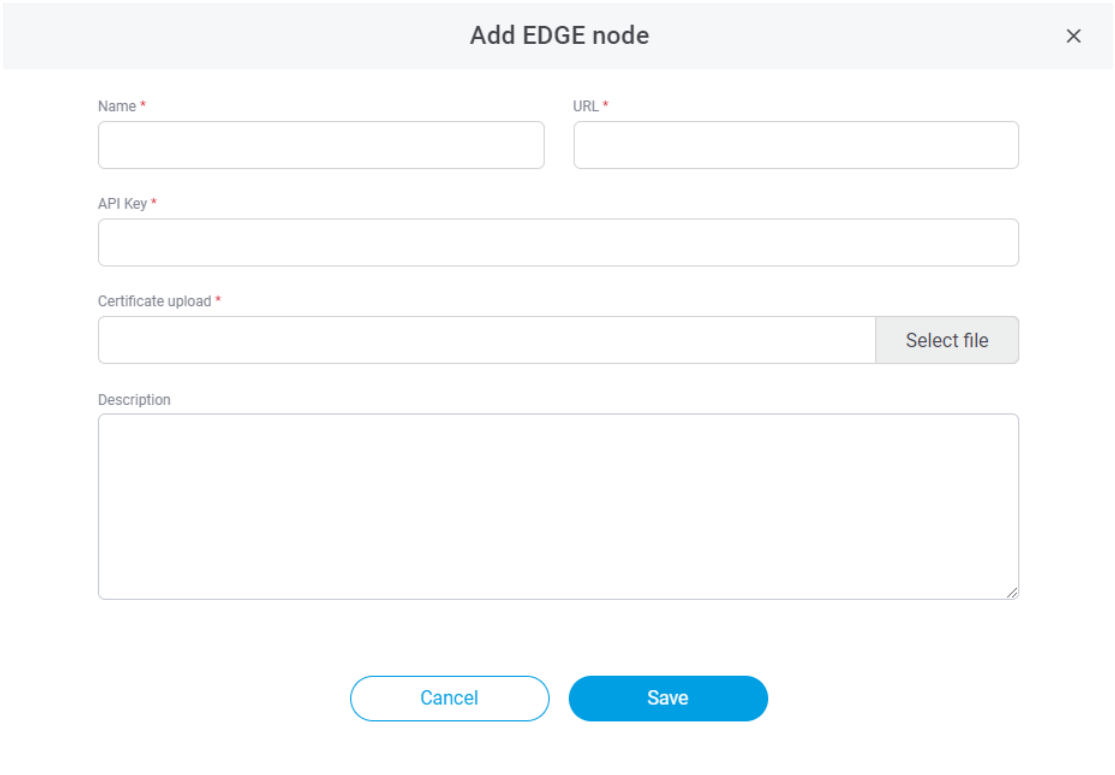


Abbildung 2.11: Anlegen einer Node

Das ist der Stand, indem sich das MES-System aktuell befindet.

Da bei einem Status Meeting des PTZ besprochen wurde, dass die Funktionalität der Anlage Priorität hat (auch in Hinblick auf das Technikforum), wurde vereinbart dass wir die Installation noch durchführen können und unsere Priorität danach auf die Funktion der Anlage legen.

2.4 Probleme und Herausforderungen

Nachdem wir da Thema MES für einige Woche ruhen lassen haben, da der Auftrag war sich auf die Steuerung der Anlage zu konzentrieren, wollten wir das MES erneut starten.

Jedoch konnte mit dem WLAN-Stick keine **Netzwerkverbindung** zur virtuellen Maschine aufgebaut werden. Der erste Gedanke war, dass es an der WLAN Verbindung der DHBW liegt.
Nachdem wir nicht mehr weitergekommen sind, kontaktierten wir die IT-Abteilung der DHBW. Nach langer Suche fiel auf, dass bei der Installation eine statische IP Adresse zugewiesen wurde. Die Vermutung ist, dass die IP einem anderem Gerät vergeben wurde in der Zeit, als das System nicht genutzt wurde.
Nachdem der Fehler identifiziert wurde, wurde eine neue Lösung für die Netzwerkverbindung gesucht, die Entscheidung fiel darauf die Kommunikation über LAN zu realisieren, was am selben Tag noch umgesetzt wurde. Anschließend wurde die Kommunikation des Systems zur VM wieder erfolgreich hergestellt.

Trotz der nun funktionierenden Netzkommunikation konnte das MES nicht gestartet werden, worauf wir den Kontakt mit dem zuständigen Forcam Mitarbeiter wieder herstellten.
Bei der Suche nach dem Fehler stellte der Forcam Mitarbeiter fest, dass die ganzen Installationsdateien nicht mehr vorhanden sind. Der Grund war, dass der VM nicht genügend Speicher zugewiesen wurde. Wir konnten uns das nicht erklären, da das System nach Installation nicht verwendet wurde.
Es blieb daher nur eine Möglichkeit, nämlich das ganze System neu aufzusetzen und installieren. Diesmal haben wir in dem Hyper-V-Manager Prüfpunkte erstellt, wie schon in Kapitel 2.3.1 beschrieben.

Nachdem die Probleme mit der fehlerhaften Netzwerkkommunikation und dem zugewiesenen Speicher geklärt wurden, kann endgültig mit der Konfiguration des MES gestartet werden.
Dazu muss zuerst eine Verbindung zwischen dem MES-System und der SPS hergestellt werden. Dieser Signalaustausch kann beispielsweise mittels OPC UA rea-

lisiert werden. Der Aufbau der Verbindung seitens der SPS wurde bereits in Teil 1 (T3100) dieser Studienarbeit beschrieben.[7]

[7]Vgl. Gulde und Langer, 2024, S.29ff.

3 Steuerung der Anlage

3.1 Versionsmanagement

Da viele verschiedene Gruppen an der Anlage im PTZ mitwirken, wird das Programm der SPS nicht nur von der Steuerungsgruppe der Anlage weiterentwickelt. Einige dieser Gruppen hatten wenig bis keinen Kontakt mit dem TIA-Portal. Weshalb sie sich teilweise erst in diese Thematik einarbeiten müssen, selbst für einfache Programm-Erweiterungen. Dies umfasst auch die Installation der Entwicklungsumgebung auf ihren privaten Rechnern, was für Unerfahrene sehr aufwendig ist. Aus diesem Grund wurde im PTZ ein Rechner mit TIA Portal ausgestattet, damit die anderen Gruppen vor Ort Änderungen vornehmen können.

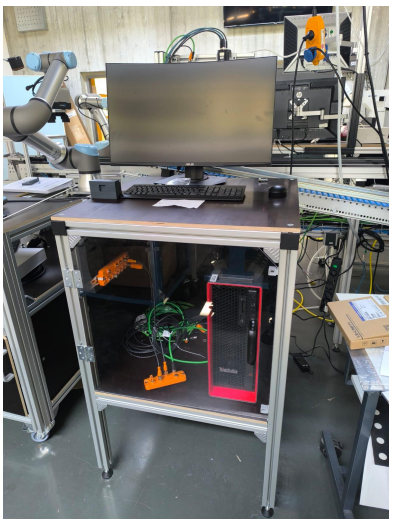


Abbildung 3.1: Arbeitsplatz mit Entwicklungsumgebung

Das Passwort für diesen Arbeitsplatz lautet „qwertz“. Hier können sich die einzelnen Gruppen jederzeit anmelden und ihre Ergänzungen testen. Ihren aktuellen Stand können sie in einer angelegten Ordnerstruktur abspeichern.

Wichtig ist dabei, dass nach den Tests das vorherige Programm wieder auf die SPS geladen wird. Dies ist wichtig, da unfertige Software auf der SPS zu Problemen führen könnte. Nachdem eine Funktion erfolgreich in deren Programmkopie integriert wurde, müssen die Gruppen sich mit der Steuerungsgruppe der Anlage absprechen und gemeinsam den aktuellen Programmstand aktualisieren.

3.2 Steuerung der Stationen

In der T3100 wurde beschrieben, dass jeder Werkstückträger mit einem QR-Code-Scanner ausgestattet ist. Diese Scanner sollten an jeder Station die QR-Codes der Werkstückträger scannen, sodass im MES/SAP System automatisch abgefragt werden kann, welche Manipulation die jeweilige Station durchführen soll. Dieses System ist auf der Abbildung 3.2 dargestellt.

Die QR-Code-Scanner wurden während dieses Projekts erfolgreich montiert. Jedoch wurden sie nicht von uns an die SPS angeschlossen, da im Verlauf des Projekts deutlich wurde, dass dieser Arbeitsaufwand nicht von einem Team aus zwei Personen bewältigt werden kann.

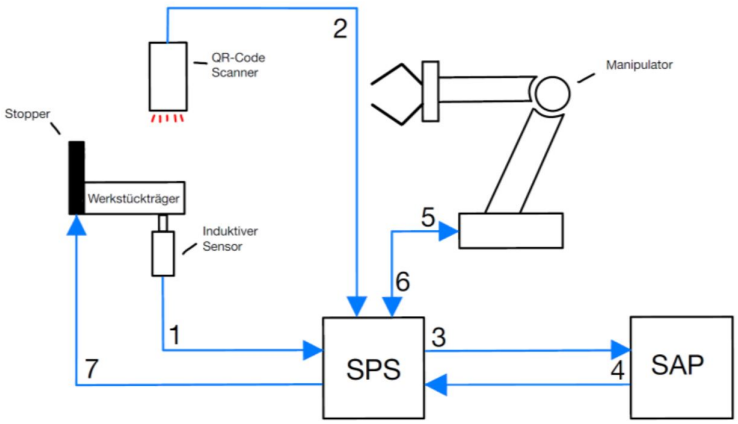


Abbildung 3.2: Schematischer Programmablauf

Somit wurde das abgebildete Schema in einer vereinfachten Version umgesetzt. Die Schritte mit den Ziffern 2-4 wurden während der Umsetzung vorerst nicht betrachtet. Im Fokus stand dabei, dass die Programmteile allgemein gehalten werden.

1. Ein induktiver Sensor erkennt einen Werkstückträger.

2. Die SPS gibt dem Manipulator das Startsignal, damit dieser beginnen kann.

3. Der Manipulator sendet der SPS das Signal, dass die Manipulation abgeschlossen ist.

4. Die SPS öffnet den Stopper, sodass das Umlaufband den Werkstückträger zur nächsten Station transportieren kann.

Die SPS gibt lediglich das Startsignal an die Manipulatoren und erhält von ihnen das Signal, wenn die Manipulation abgeschlossen ist. Informationen über verschiedene Varianten werden in diesem Schritt nicht berücksichtigt.

Um einen Stau von Werkstückträgern an einer Station zu vermeiden, sind alle Stationen nach dem Prinzip der Abbildung 3.3 zu programmieren.

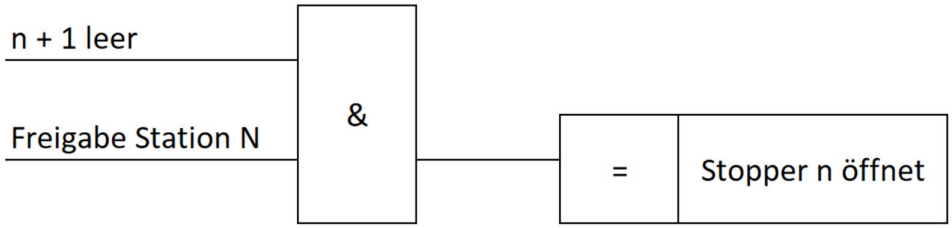


Abbildung 3.3: Stopper Programmierung

Die Abbildung 3.3 beschreibt die Bedingungen für Schritt 7. Die erste Bedingung ist, dass die Station die Freigabe erteilt, damit der Prozess abgeschlossen ist und der Werkstückträger weiterfahren kann. Die zweite Bedingung ist, dass sich an der Station N+1 kein Werkstückträger befindet. Erst dann öffnet die Ventilinsel den Stopper der Station N. Somit können sich im System maximal zehn Werkstückträger befinden.

Für ein reales Produktionsnetzwerk macht dieses System an sich keinen Sinn. Zum Beispiel würde die gesamte Anlage stehen, wenn ein Mitarbeiter am Handarbeitsplatz kurzzeitig seine Arbeit unterbricht. Durch eine andere Ansteuerung der Stopper könnte es passieren, dass sich am Handarbeitsplatz zwei Werkstückträger anstauen, welche der Mitarbeiter dann im Anschluss abarbeiten müsste. Zur Steuerung dieser Situation befinden sich bereits Lasersensoren an der Anlage, wie zum Beispiel auf Abbildung 3.4 dargestellt. Mit diesen Sensoren kann festgestellt werden, ob sich zum Beispiel zwei Werkstückträger am Handarbeitsplatz befinden.

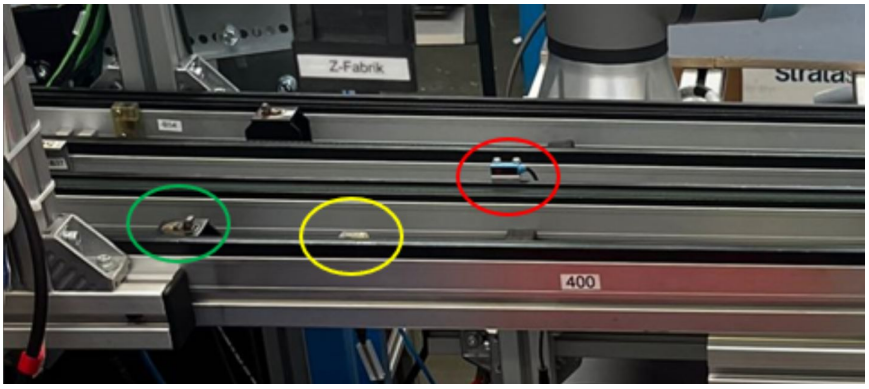


Abbildung 3.4: Förderband mit Stopper (grün), induktiver Sensor (gelb), Lasersensor (rot)

Um dies umzusetzen, müsste ein Konzept erstellt werden, das festlegt, an welcher Station wie viele Werkstückträger aufgestaut werden können.

Die induktiven Sensoren, die Laser und die Stopper wurden während der T3100 und T3200 beschriftet. Diese Beschreibungen wurden im TIA-Projekt nachgetragen, damit bei der Programmierung dieser Sensoren und Aktoren barrierefrei gearbeitet werden kann.

3.3 Neuaufbau Steuerungsablauf

3.3.1 Belt feeding

Das Roboterprogramm der ersten Station besteht aus zwei Teilen, wobei die Kommunikation zwischen Roboter und SPS mithilfe von Profinet realisiert wurde.

1. Auflegen der Werkstückträger:
   Der erste Teil gewährleistet, dass immer ein Werkstückträger am Stopper eins steht. Wenn der induktive Sensor „null“ detektiert, erhält der Roboter die Information, dass er einen Werkstückträger auf das Bandumlaufsystem legen soll. Der Roboter entnimmt die Werkstückträger von den Positionen, wie in Abbildung 3.5 gezeigt. An den Positionen, an denen kein Werkstückträger mehr vorhanden ist, speichert der Roboter die leeren Positionen. Nach dem Greifen des Werkstückträgers positioniert der Roboter diesen genau am Stopper eins.

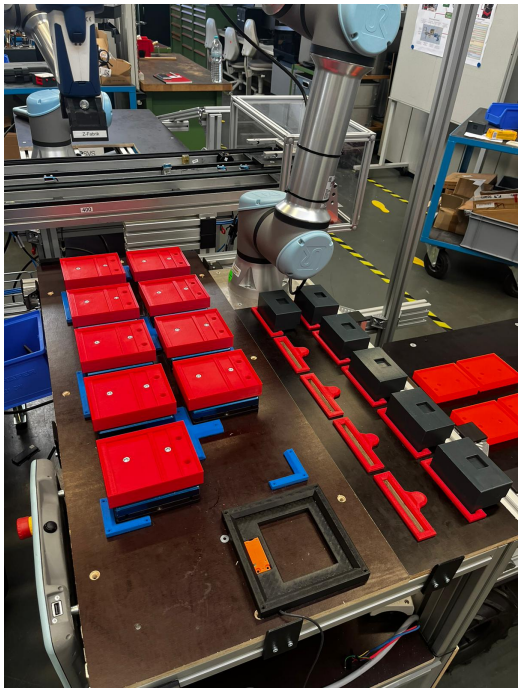


Abbildung 3.5: Werkstückträger an erster Station

2. Auflegen der Werkstücke:
   Mit dem zweiten Teil werden die Werkstückträger von dem Roboter mit den Werkstücken bestückt. Wenn der Werkstückträger wegen eines Fehlers nicht zur nächsten Station fährt, erkennt der induktive Sensor weiterhin einen Werkstückträger und der Roboter möchte somit erneut ein Bauteil auf den Werkstückträger setzen.

3. Umsetzung:
   Damit es nicht zu diesem Problem führt, wurde das SPS-Programm ein wenig abgeändert. Somit wird mithilfe einer Flanke erkannt, wann der Werkstückträger an Station zwei (Bin Picking), zur nächsten Station fährt. Solange diese Flanke nicht erkannt wird, unterbricht die SPS mithilfe eines Bits den Roboter.

   Während der Programmierung kam es jedoch zu erheblichen Problemen. Die Stopper sowie der Sensor waren nicht richtig am Bandumlaufsystem befestigt. Das hatte zur Folge, dass der Werkstückträger beim Anschlag am Stopper sich in einer ungewöhnlichen Art und Weise anschlägt. Zu Beginn wurde von einem Softwarefehler ausgegangen. Da in der Programmierung keine Vorkenntnisse mit Flanken vorhanden waren. Nach genauerer Beobachtung der Hardwarekomponenten ist uns aufgefallen, dass der induktive Sensor beim Anschlag des Werkstückträgers auf „eins“ geht, danach kurzzeitig wieder auf „null“ und dann bleibt das Signal auf „eins“ stehen. Dies wird auch als Glitch bezeichnet.

4. Aussicht:
   Durch die Programmierung des Roboters, dass dieser speichert, an welchen Positionen sich noch Werkstückträger befinden, ist das Programm des Roboters sehr unübersichtlich und schlecht einsehbar. Bei der Anpassung der SPS auf den Roboter war oft unklar, wann der Roboter welche Schritte beginnt.

   Am besten wäre eine weitere Aufteilung des Programms.
   Für jede Werkstückträger-Position sollte ein Unterprogramm erstellt werden.

Für die zehn Positionen der Werkstücke sollte ebenfalls jeweils ein Unterprogramm erstellt werden. Der Roboter würde dann alle Befehle von der SPS erhalten. Das bedeutet konkret, dass die SPS über ein Array mit Booleans speichert, an welchen Positionen sich Werkstückträger bzw. Werkstücke befinden. Durch diese Programmierung wäre die Kommunikation zwischen Roboter und SPS deutlich übersichtlicher.

Der Mensch könnte trotzdem bei dieser Programmierung Probleme verursachen. Wenn ein Mitarbeiter einen Werkstückträger von einer Position entnimmt, bleibt der Boolean dieser Position dennoch auf eins. Im Allgemeinen ist es sehr fehleranfällig, solche Daten zu speichern.

Vergleichbar ist die Produktion verschiedener Varianten zu gleicher Zeit. Hier könnte die SPS so programmiert werden, dass sie speichert, auf welchem Werkstückträger was produziert werden soll. Das Hauptproblem dabei ist, dass die Steuerung nicht bemerkt, wenn ein Werkstückträger entnommen wird, wodurch die Reihenfolge nicht mehr mit der Steuerung übereinstimmt. Aus diesem Grund wurden die QR-Code-Scanner angeschafft, dadurch erkennt die SPS an den einzelnen Stationen, welcher Werkstückträger bearbeitet wird.

Um nun zurück auf das Problem der Werkstückträgerlager zu führen, wäre es auch hier sinnvoll, Sensoren zu ergänzen, damit nicht gespeichert werden muss, an welcher Position sich Werkstückträger oder Werkstücke befinden, sondern dieser Zustand dauerhaft abgefragt wird, ob sich an der jeweiligen Stelle ein Werkstückträger befindet.

3.3.2 Bin Picking

Das Bin Picking funktioniert relativ gut, lediglich bei der Variantenauswahl gab es gelegentlich Probleme. Da es jedoch keine Gruppe gab, die diese Station betreute, war es schwierig, diese zu überarbeiten. Aus diesem Grund wurden am Bin Picking keine Änderungen vorgenommen.

3.3.3 Umleger 1

Auf beiden Seiten des Förderbandes befindet sich jeweils ein Umleger. Diese sind pneumatisch gesteuert, eine zentrale Ventilinsel erhält die Signale von der SPS. Die Programmierung des ersten Umlegers ist voll funktionsfähig. Die Drosseln der Pneumatik-Elemente befinden sich unterhalb des Umlegers, wobei diese gelegentlich neu eingestellt werden müssen.

3.3.4 Handarbeitsplatz

Am Handarbeitsplatz wurde während der T3200 ein induktiver Sensor ergänzt. In Absprache mit dieser Gruppe wurde der passende Sensor ausgewählt, damit er in die Steuerung integriert werden kann. Der Sensor wurde nicht per IO-Link an die SPS angebunden. Die direkte Verdrahtung wurde von der Schaltschrankgruppe durchgeführt. Die Einbindung des Sensors in das TIA-Portal über die PLC-Variablen wurde von der Steuerungsgruppe umgesetzt. Die Programmierung des Sensors wurde wiederum von der Handarbeitsplatzgruppe durchgeführt.

Des Weiteren befindet sich an der Anlage ein IO-Link Master der Firma IFM. An diesem waren ursprünglich vier Sensoren angeschlossen. Zwei davon waren Lichtschrankensensoren am Handarbeitsplatz, die anderen zwei sind optische Sensoren, die auf Höhe der End of Line montiert waren. Keine der anderen Gruppen wusste über diese Sensoren Bescheid, für was diese angebracht wurden.

Während der T3100 ist bereits aufgefallen, dass der IO-Link Master rot blinkt. Das Problem wurde dann in der T3200 angegangen, und mit verschiedenen Ansätzen ist es nicht gelungen, die Verbindung zwischen Master und SPS herzustellen.

Da kurz zuvor im Laborunterricht ein vergleichbares Problem aufgetreten war, war bekannt, dass dieses Problem nicht ohne Weiteres behoben werden kann. Daraufhin haben wir einen anderen IO-Link Master erhalten, den wir erfolgreich mit der SPS verbinden konnten. Es müssen nun lediglich die zwei Lasersensoren am Handarbeitsplatz am IO-Link Master ergänzt werden.

Die optischen Sensoren könnten aus Übersichtsgründen vorerst demontiert werden, da sie für den Ablauf der Anlage irrelevant sind.

3.3.5 End of Line

Zwischen der End of Line und der SPS gibt es keine Kommunikation. Des Weiteren muss beachtet werden, dass sich die Kamera der End of Line nicht über dem dafür vorgesehenen Stopper befindet. Somit kann das Kamerasystem nur benutzt werden, wenn das Förderband steht.

3.3.6 Demontage/Umleger 2

Der Roboter des Demontageprozesses ist ebenfalls nicht mit der SPS verbunden. Für die Demontagestation und den zweiten Umleger ist nur ein Stopper vorgesehen. Unter welchen Bedingungen der Stopper öffnet, war bis zum Schluss des Projekts unklar. Des Weiteren gibt es verschiedene Szenarien, in denen der Roboter der Station eins direkt auf den Werkstückträger greift, welcher an der Demontagestation liegt.

3.4 Human Machine Interface

3.4.1 Human Machine Interface allgemein

Der Grundaufbau des HMI wurde während der T3100 durchgeführt. Eigentlich sollte während der T3200 das HMI weitergeführt werden, so dass verschiedene Prozessdaten dort eingesehen werden können. Da bis jetzt noch nicht feststand wie genau die Anlage betrieben werden soll, konnte die endgültige HMI-Visualisierung noch nicht fertiggestellt werden.

Auch das Variantenmanagement und die Einstellmöglichkeiten, welche Variante produziert werden kann, konnten nicht über das HMI umgesetzt werden. In diesem Fall muss zuerst geklärt werden, ob die Einstellung der Variante im MES oder auf dem HMI getätigt werden soll.

3.4.2 Profinet Anschluss des Human Machine Interface

Zudem wurde in der T3100 beschrieben, dass sich an der Anlage zwei verschiedene Profinet-Netzwerke befinden. Beide gehen von der SPS aus. Im Ersten sind alle Profinet-Teilnehmer über einen Hub an die SPS angebunden. Im zweiten Profinet befinden sich lediglich die SPS und das HMI. Während der Programmierung muss beim Aufspielen der Programme (HMI und SPS) immer die Leitung umgesteckt werden. Bei der Programmierung mit einem externen Gerät ist dies relativ einfach umsetzbar. Jedoch ist das Umstecken bei dem Desktop-PC, auf dem das TIA installiert ist, sehr umständlich und zeitaufwendig.

Daraufhin haben wir versucht, das HMI in das erste Profinet-Netz zu setzen. Dies sollte in der Netzwerktopologie im TIA Portal konfigurierbar sein. Jedoch hat die SPS nach verschiedenen Versuchen immer Fehlermeldungen ausgegeben.

Da sich die Priorität der Studienarbeit etwas geändert hat und noch viele wichtigere Themen anstanden, wurde diese Thematik nicht weiter verfolgt.

3.4.3 Stopper Ansteuerung über Human Machine Interface

In der Zukunft wird es vorkommen, dass einzelne Stationen aufgrund von Umbauarbeiten nicht betrieben werden können. Damit das Bandumlaufsystem dennoch benutzt werden kann, sollten die Stopper über das HMI manuell ansteuerbar sein.

Die Umsetzung der HMI-Seiten verlief ohne Probleme. Jedoch wurden die Ausgänge der Ventilinsel immer in den einzelnen Programm-Bausteinen gesetzt. Normalerweise werden die Ausgänge immer zentral im OB1 gesetzt.

Wenn ein Ausgang aus zwei verschiedenen Bausteinen gesetzt werden soll, führt dies zu Fehlern. Zum Beispiel wird der Ausgang im ersten Baustein gesetzt und im zweiten wieder zurückgesetzt. Deshalb muss ein Ausgang über ein oder-Glied angesteuert werden, indem jeder Baustein, der den Ausgang setzen kann, eingefügt wird.

4 Zusammenfassung

4.1 Manufacturing Execution System

Zusammenfassend lässt sich festhalten, dass das gewünschte MES-System „Forcam Force Edge“ erfolgreich installiert wurde.
Dazu wurden zu Beginn die Grundlagen eines MES verinnerlicht, bevor die Vorbereitung starten konnte.
Da im IWT bereits das System von Forcam besteht, fiel die Auswahl eines geeigneten System aufgrund der zukünftigen Kopplung der beiden Anlagen weg.
Bevor wir die Installation vorbereiten konnten, wurde uns bei der Schulung „How to connect a machine“ die Anbindung einer Anlage an das System mit praktischen Beispielen und kleinen Programmieraufgaben nähergebracht.
Nachdem die passende Hardware bei Berücksichtigung der „System Requirements“ von Forcam ausgewählt wurde, konnten die benötigten Add-Ons sowie die VM mit dem Linux-basierten Betriebssystem Ubuntu installiert werden.
Nachdem die Installation des MES aufgrund von falsch zugewiesenen Speicher und instabiler Netzverbindung gescheitert war, wurde das System neu aufgesetzt und ist jetzt bereit, konfiguriert zu werden. Um die vergangenen Arbeitsschritte sowie die nächsten Schritte nachzuvollziehen wurden in BW-Sync (Cloud des PTZ) alle relevanten Dateien (Manuals, Übungsaufgaben etc.) abgelegt. Zudem ist der Gruppe, welche das MES an die Steuerung anbindet zu empfehlen, an der Schulung „How to connect a machine“ teilzunehmen.

Abschließend lässt sich sagen, dass die Installation des MES nicht reibungsfrei verlief, was unserer Meinung nach unter anderem an dem Austausch mit Forcam lag,

welcher ausschließlich remote stattfand. Fehler wie der unzureichend zugewiesen Speicherplatz oder das Problem mit der Netzwerkverbindung hätten bei genauer Zuteilung der Arbeitsschritte vermieden werden können.
Der abrupte Abbruch der MES-Einführung war im Nachhinein unglücklich, da das gelernte aus der Schulung im Endeffekt nicht umgesetzt werden konnte.

Was in Sachen Anlagensteuerung umgesetzt wurde, wird im nächsten Abschnitt zusammengefasst.

4.2 Steuerung der Anlage

Im Allgemeinen kann gesagt werden, dass die einzelnen Stationen nicht gut auf die SPS abgestimmt sind. Die verschiedenen Roboter speichern Daten, was das gesamte System sehr undurchsichtig macht.

Der vorhandene Code der SPS wurde während der Studienarbeit so gut wie möglich strukturiert. Jedoch sind die einzelnen Programmteile oft sehr unverständlich und schlecht bis gar nicht kommentiert. Der Programmcode der Umleger ist sehr gut und funktionsfähig.

Grundsätzlich wäre es für die Programmierung der gesamten Anlage am besten, wenn die Programmierung bis auf die Umleger komplett neu aufgesetzt wird. Es sollte zu Beginn ein Konzept erstellt werden, wie alle Stationen ineinandergreifen. In diesem Konzept sollte klar definiert werden, auf welcher Ebene der Automatisierungspyramide alle Prozessdaten gespeichert werden sollen. Des Weiteren muss festgelegt werden, welche Daten zwischen den Stationen wie ausgetauscht werden. Auf Basis dieses Konzeptes muss dann definiert werden, in welchen Schritten dieses Konzept umgesetzt werden kann. Dabei sollte der Anspruch sein, den erstellten Code übersichtlich und gut zu kommentieren, damit nachfolgende Studenten besser damit klarkommen.

Dafür wurde bereits von der Gruppe des Energie-Monitorings ein Styleguide erstellt, wie die Programmierung durchgeführt werden soll.

Literaturverzeichnis

Bücher

Dürkop, Lars (2016). *Automatische Konfiguration von Echtzeit-Ethernet*. Bd. Band 5. Springer Vieweg. ISBN: 978-3-662-54124-1.

Kletti, Jürgen und Rainer Deisenroth (2021). *Lehrbuch für digitales Fertigungsmanagement: Manufacturing Execution Systems - MES*. Springer Vieweg. ISBN: 978-3-662-63202-4.

Weitere Quellen

Canonical Ltd. (2024). *How to set up Ubuntu on Hyper-V*. URL: `https://ubuntu.com/server/docs/how-to-set-up-ubuntu-on-hyper-v`. letzte Einsichtnahme: 08.07.2024.

FORCAM GmbH (2021). *Manual - FORCAM FORCE EDGE*. URL: `https://forcam.com/app/uploads/2021/11/Manual-FORCAM-FORCE-EDGE.pdf`. letzte Einsichtnahme: 05.07.2024.

– (2022). *Schulungpräsentation: How to connect a machine - Forcam Academy (Quelle: Christoph Torbohm - Forcam GmbH)*. letzte Einsichtnahme: 06.07.2024.

– (2024). *FORCE EDGE CONNECT System Requirements (Quelle: Christoph Torbohm - Forcam GmbH)*. letzte Einsichtnahme: 03.07.2024.

Gulde, Leo und Maximilian Langer (2024). *Produktionsnetzwerk – Optimierung der Steuerung und Vernetzung der Lernfabrik (Studienarbeit T3100)*.

ubuntu Deutschland e.V. (2024). *Long Term Support*. URL: `https://wiki.ubuntuusers.de/Long_Term_Support/`. letzte Einsichtnahme: 08.07.2024.

Abkürzungsverzeichnis

EoL	End of Line
ERP	Enterprise Resource Planning
HMI	Human Machine Interface
IP	Internet-Protokoll
IWT	Institut für Weiterbildung, Wissens- und Technologietransfer
MDC	Machine Data Collection
MES	Manufacturing Execution System
MQTT	Message Queuing Telemetry Transport
NC	Numeric Control
DNC	Distributed Numeric Control
OPC UA	Open Platform Communication Unified Architecture
PTZ	Produktionstechnisches Zentrum
SCADA	Supervisory Control and Data Acquisition
SPS	Speicherprogrammierbare Steuerung
LTS	Long Term Support
VM	Virtuelle Maschine
WLAN	Wireless-LAN

Abbildungsverzeichnis

2.1 Automatisierungspyramide (Dürkop, 2016) 3
2.2 Edge Aufbau (FORCAM GmbH, 2021) 8
2.3 Hardwarevoraussetzungen (FORCAM GmbH, 2024) 12
2.4 Virtuelle Maschine Ubuntu in VMware 15
2.5 Prüfpunkte von Virtueller Maschine in VMware 15
2.6 Anmeldebildschirm bei Ubuntu 16
2.7 Ubuntu Oberfläche 16
2.8 Kontrollleiste in der virtuellen Maschine 17
2.9 Aufrufen der Edge über die Adresse 18
2.10 Aufrufen über localhost 18
2.11 Anlegen einer Node 19

3.1 Arbeitsplatz mit Entwicklungsumgebung 23
3.2 Schematischer Programmablauf 25
3.3 Stopper Programmierung 26
3.4 Förderband mit Stopper (grün), induktiver Sensor (gelb), Lasersensor (rot) 27
3.5 Werkstückträger an erster Station 28