DHBW
Duale Hochschule
Baden-Württemberg
Ravensburg
Campus Friedrichshafen

# Zukunftsfabrik Bodensee: Predictive Maintenance für das Transfersystem

## Studienarbeit T3_3200

Studiengang Elektrotechnik

Studienrichtung Automation

Duale Hochschule Baden-Württemberg Ravensburg, Campus Friedrichshafen

von

Aaron Hahn, Louis Ranke

| | |
|---|---|
| Abgabedatum: | 1. Juli 2025 |
| Bearbeitungszeitraum: | 07. April 2025 - 26. Juni 2025 |
| Matrikelnummer: | 9080428, 8908532 |
| Kurs: | TEA22 |
| Professorin: | Herr Prof. Dr. Thomas Dietmüller |

# Disclaimer

Das in dieser Arbeit gewählte generische Maskulinum bezieht sich zugleich auf die männliche, weibliche und andere Geschlechteridentitäten. Zur besseren Lesbarkeit wird auf die Verwendung männlicher und weiblicher Sprachformen verzichtet. Alle Geschlechteridentitäten werden ausdrücklich mitgemeint, soweit die Aussagen dies erfordern.

# Erklärung

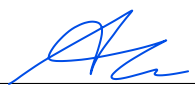
gemäß Ziffer 1.1.14 der Anlage 1 zu §§ 3, 4 und 5 der Studien- und Prüfungsordnung für die Bachelorstudiengänge im Studienbereich Technik der Dualen Hochschule Baden-Württemberg vom 29.09.2017 in der Fassung vom 24.07.2023.

Ich versichere hiermit, dass ich meine Studienarbeit T3_3200 mit dem Thema:

*Zukunftsfabrik Bodensee: Predictive Maintenance für das Transfersystem*

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 1. Juli 2025



Aaron Hahn



Louis Ranke

# Inhaltsverzeichnis

**1 Einleitung** **1**

**2 Theoretische Grundlagen** **2**

2.1 Condition Monitoring . . . . . 2

2.2 MES Forcam . . . . . 4

**3 Datenmonitoring** **7**

3.1 Sensordatenerfassung und SPS-Kommunikation . . . . . 7

3.2 Edge-Computing mit Proxmox und Docker . . . . . 9

3.3 Datenfluss zwischen SPS, Node-RED und TimescaleDB . . . . . 11

3.4 Visualisierung und Statusüberwachung mittels Peakboard . . . . . 14

3.5 Systemübersicht der Datenverarbeitung . . . . . 16

**4 Ausblick: Umsetzung der Datenverarbeitung mit MES FORCAM** **18**

4.1 Sensordatenerfassung und Maschinenanbindung . . . . . 18

4.2 Datenverarbeitung und Speicherung . . . . . 19

4.3 Visualisierung und Überwachung . . . . . 20

4.4 Vorteile der FORCAM-Integration . . . . . 21

**5 Zusammenfassung und Fazit** **22**

**Literaturverzeichnis** **24**

**Abbildungsverzeichnis** **25**

**Abkürzungsverzeichnis** **26**

**A Dokumentation technischer Erfahrungen und Problemlösungen** **27**

A.1 Virtuelle Infrastruktur mit Proxmox . . . . . 27

A.2 Debian-Umgebung und Systemzugriffe . . . . . 29

A.3 Node-RED – Konfiguration und Fehleranalyse . . . . . 31

A.4 TimescaleDB – Zugriff und Tabellenstruktur . . . . . 31

A.5 Zusammenfassung der Lessons Learned . . . . . 32

# Kurzfassung

In dieser Studienarbeit wurde ein Condition Monitoring-System für das Transfersystem der Zukunftsfabrik Bodensee entwickelt und prototypisch umgesetzt. Ziel war es, eine modulare Lösung zur automatisierten Erfassung, Verarbeitung, Speicherung und Visualisierung von Sensordaten wie Temperatur, Schwingung und Drehzahl zu realisieren.

Die erfassten Messwerte werden von Sensoren direkt an den Motoren gewonnen und über eine Siemens S7-1200 SPS an ein Edge-Gerät auf Basis eines Intel NUC übertragen. Dort erfolgt die Verarbeitung mithilfe von Node-RED, die Speicherung in einer TimescaleDB und die Visualisierung über Peakboard. Die Infrastruktur wurde containerbasiert innerhalb einer virtuellen Umgebung mit Proxmox realisiert.

Im Verlauf des Projekts konnten technische Herausforderungen wie Netzwerkkonfigurationen und Datenbankzugriffe erfolgreich gelöst werden. Die entwickelte Systemarchitektur bietet eine solide Basis für zukünftige Erweiterungen, beispielsweise durch Machine-Learning-gestützte Anomalieerkennung oder eine Integration in das FORCAM MES-System.

# Abstract

This thesis presents the development and prototype implementation of a condition monitoring system for the transfer unit at the "Zukunftsfabrik Bodensee". The system is designed to collect, process, store, and visualize sensor data—such as temperature, vibration, and rotational speed—in a modular and automated manner.

Sensor data is gathered directly from motors and transmitted via a Siemens S7-1200 PLC to an edge device (Intel NUC). The data is then processed using Node-RED, stored in a TimescaleDB, and visualized with Peakboard. The entire setup is virtualized and containerized using Proxmox and Docker technologies.

Technical challenges like network integration and SQL communication were identified and successfully resolved during the project. The resulting system provides a scalable and extensible foundation for future developments such as anomaly detection using machine learning or integration with the FORCAM MES platform.

# 1 Einleitung

> *„Condition monitoring is essential for increasing the reliability of machines, reducing maintenance costs, and preventing unexpected failures.“*
> — Mobley, R. K. (2002). *An Introduction to Predictive Maintenance*. Elsevier. [Mob02]

Vor diesem Hintergrund sollen im Rahmen dieser Studienarbeit die Grundsteine für ein sogenanntes *Predictive Maintenance*-System gelegt werden. Dieser Grundstein besteht darin, ein funktionierendes Condition Monitoring-System aufzubauen.

Zu Beginn der Arbeit werden die theoretischen Grundlagen erläutert, auf denen die praktische Umsetzung basiert. Diese umfassen die Grundlagen des Condition Monitorings, den aktuellen Stand der Hardware eines Intel Next Unit of Computing (NUC) sowie eine Einführung in das Zielsystem, ein Manufacturing Execution Systeme (MES) der Firma FORCAM.

Weiterführend wird ein Einblick in das Datenmonitoring inklusive aller relevanten Schnittstellen gegeben. Dabei werden die vielfältigen Verbindungen von einer Siemens SPS über einen Intel NUC mit Datenbanken, Node-RED Workflows und einem Peakboard zur Visualisierung erläutert.

Abschließend folgt ein Ausblick zur Optimierung der Schnittstellen mit dem FORCAM FLEX MES im Hinblick auf Datenverarbeitung und -darstellung. Weitere Optimierungsmöglichkeiten werden aufgezeigt.

# 2 Theoretische Grundlagen

## 2.1 Condition Monitoring

Condition Monitoring bezeichnet die systematische Überwachung technischer Systeme durch die Erfassung und Analyse physikalischer Parameter wie Schwingungen, Temperaturen oder Stromaufnahme. Ziel des Condition Monitoring ist es, den Zustand einer Maschine oder Anlage in Echtzeit oder in regelmäßigen Intervallen zu erfassen. Dadurch sollen potenzielle Störungen frühzeitig erkannt werden. Darauf aufbauend wird die Instandhaltung effizient und vorausschauend gestaltet. Anders als bei rein zeitbasierten oder reaktiven Wartungsansätzen ermöglicht Condition Monitoring eine zustandsbasierte Wartung. Dadurch lassen sich Ausfallzeiten deutlich verringern und Betriebskosten nachhaltig senken. [KB22]

Die Grundlage eines Condition Monitoring-Systems bilde eine strukturierte Zustandsüberwachung, die sich in drei funktionale Phasen gliedern lässt: Zunächst erfolgt die Zustandserfassung durch den Einsatz von Sensorik zur Messung relevanter physikalischer Größen, etwa Schwingungen an Lagern, Temperaturen an Motoren oder die Stromaufnahme von Antrieben. Anschließend wird der aktuelle Zustand mit historischen Daten oder definierten Grenzwerten verglichen. Dies dient dazu, Abweichungen festzustellen. Die dritte Phase umfasst die Diagnose, bei der aus jenen erkannten Abweichungen Rückschlüsse auf mögliche Fehlerursachen gezogen werden. [KB22]

Zunehmende Digitalisierung und Datenverfügbarkeit erweitern den Funktionsumfang moderner Condition Monitoring-Systeme. Aktuelle Entwicklungen fokussieren sich ins-

besondere auf die Integration künstlicher Intelligenz und maschinellen Lernens, um aus den Datenmengen Muster zu extrahieren und Prognosen über die verbleibende Nutzungsdauer (Remaining Useful Life, RUL) technischer Komponenten zu ermöglichen. Dies führt zu einem Paradigmenwechsel von rein reaktiver oder präventiver hin zu prädiktiver Instandhaltung. [BZ22]

Die Einsatzgebiete von Condition Monitoring sind weitläufig und reichen von der klassischen Fertigungsindustrie über die Energie- und Prozessindustrie bis hin zur Transport- und Logistikbranche. In all diesen Bereichen steht die Steigerung der Anlagenverfügbarkeit sowie die Verbesserung der Planungssicherheit im Vordergrund. Besonders deutlich zeigen sich die Potenziale des Condition Monitorings im Bereich von Fördertechniksystemen, etwa in Bandsystemen, wie sie in der automatisierten Produktion zahlreich vorkommen. Hier ermöglicht die kontinuierliche Überwachung zentraler Anlagenkomponenten wie Motoren, Lager oder Fördergurte die rechtzeitige Erkennung von Verschleißerscheinungen. So kann beispielsweise ein zunehmendes Vibrationsniveau auf eine Unwucht oder eine Lagerschädigung hindeuten, während Temperaturanstiege Rückschlüsse auf Überlastung oder Schmierprobleme zulassen. [KB22]

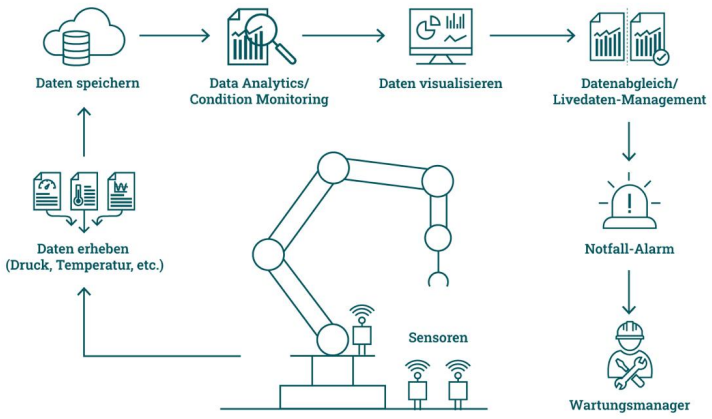



**Abbildung 2.1:** Grafik zum Condition Monitoring. Quelle:[iT 25]

Ein Condition Monitoring-System für ein Bandsystem, wie es auch in dieser Studienarbeit betrachtet wird, basiert typischerweise auf einem Zusammenspiel aus Sensorik, Datenerfassungseinheiten, einem Analysetool und gegebenenfalls einem Visualisierungsmodul. Die erfassten Daten werden entweder lokal oder über industrielle Kommunikati-

onsprotokolle wie OPC UA an eine zentrale Plattform übermittelt. Dort erfolgt die Verarbeitung mittels Algorithmen, die sowohl Grenzwertanalysen als auch modellbasierte oder datengetriebene Verfahren nutzen. Im Idealfall werden auftretende Abweichungen nicht nur gemeldet, sondern es wird automatisch eine Handlungsempfehlung generiert – etwa zur Durchführung einer Wartung innerhalb eines definierten Zeitfensters.

Laut Becker und Zaeh ergeben sich somit nachstehend dargestellte Vorteile von Condition Monitoring-Systemen: Beginnend mit der Reduktion ungeplanter Stillstände und der Vermeidung von Folgeschäden kann die Wartung gezielt auf Basis realer Zustandsinformationen durchgeführt werden. Dies verringere nicht nur die Belastung für das Instandhaltungspersonal, sondern trage auch zur Verlängerung der Lebensdauer von Komponenten bei. Gleichzeitig erlaube die systematische Datenanalyse auch Rückschlüsse auf Prozessstörungen und unterstützt so übergreifend die Optimierung der gesamten Produktionskette. [BZ22]

## 2.2 MES Forcam

Im Zentrum der digitalen Transformation der Fertigungsindustrie im Rahmen der Industrie 4.0 stehen leistungsfähige und flexible Systeme. Diese Systeme werden durch sogenannte MES realisiert. Sie dienen als Verbindung zwischen der operativen Fertigungsebene (Shopfloor) und den übergeordneten Planungssystemen (Topfloor). Eine solche Lösung bietet die FORCAM ENISCO GmbH mit ihrem Produkt FORCAM FORCE MES LITE, das insbesondere auf die Bedürfnisse kleiner und mittelständischer Unternehmen zugeschnitten ist. [FOR23]

### Funktionalitäten von FORCAM FORCE MES LITE

FORCAM FORCE MES LITE ermöglicht eine umfassende Überwachung in Echtzeit oder in bestimmten Zeitintervallen von Produktionsprozessen durch die Erfassung und Analyse relevanter Key Performance Indicators wie Maschinenverfügbarkeit und Ge-

samtanlageneffektivität. Diese Daten bilden die Grundlage für kontinuierliche Verbesserungsprozesse und ermöglichen eine datengetriebene Optimierung der Fertigung. [FOR23]

Die Lösung besteht aus zwei modularen Paketen:

- **Verfügbarkeitspaket**: Es fokussiert sich auf die Erfassung von Maschinenzuständen wie Betriebsbereitschaft, Produktionsstatus und Stillstandszeiten. Durch die Analyse dieser Daten können Ursachen für Produktionsunterbrechungen identifiziert und entsprechende Gegenmaßnahmen eingeleitet werden.

- **OEE-Paket**: Hier werden zusätzlich Produktionsmengen und Qualitätsdaten erfasst, um die Gesamtanlageneffektivität zu berechnen. Dies ermöglicht eine ganzheitliche Bewertung der Produktionsleistung und unterstützt die Identifikation von Optimierungspotenzialen. [FOR23]

Ein zentrales Merkmal von FORCAM FORCE MES LITE ist die nahtlose Integration in bestehende Produktionsumgebungen. Die Anbindung der Maschinen erfolgt standardmäßig über bewährte I/O-Controller, wobei vier digitale Signale (Maschine eingeschaltet, Automatik AN/AUS, Maschine produziert, Maschine steht) erfasst werden. Diese können individuell angepasst und erweitert werden, um spezifische Anforderungen abzudecken. [FOR23]

Darüber hinaus bietet die Lösung umfangreiche Visualisierungen, Dashboards und Reports, die eine transparente Darstellung der Produktionsprozesse ermöglichen. Alle Maschinenzustände und -informationen sind von überall aus und zu jeder Zeit einsehbar, was eine flexible und ortsunabhängige Überwachung der Fertigung erlaubt. [FOR23]

## Einsatz in der Industrie

Die Bedeutung von MES-Lösungen wie FORCAM FORCE MES LITE wird auch durch deren Einsatz in großen Industrieprojekten unterstrichen. So setzt beispielsweise Siemens

Energy bei der digitalen Anbindung von bis zu 3.000 Maschinen an weltweit 80 Fertigungsstandorten auf die Technologie von FORCAM. Ziel ist es, die Produktionsprozesse durch ein standardisiertes IT-System zu optimieren und eine durchgängige Transparenz in der Fertigung zu gewährleisten. [Pre23]

# 3 Datenmonitoring

Die Verarbeitung und Darstellung von Prozessdaten aus einer speicherprogrammierbare Steuerung (SPS) in einem industriellen Umfeld erfolgt in der vorliegenden Projektstruktur über eine durchgängige, modulare Datenpipeline. Ziel ist es, Sensordaten, die an den Motoren des Umlaufbandes erfasst werden, automatisiert und verlustfrei in einer Datenbank zu speichern, grafisch aufzubereiten und letztlich visuell auf einem Anzeigegerät – dem sogenannten Peakboard – darzustellen. Dabei kommen eine Vielzahl etablierter industrieller Kommunikationsprotokolle und Softwarekomponenten zum Einsatz.

## 3.1 Sensordatenerfassung und SPS-Kommunikation

Der Ursprung der erfassten Datenkette liegt in Industriesensoren, die verschiedene physikalische Messgrößen aufnehmen. Zum Einsatz kommen Temperatur-, Schwingungs- sowie Induktionssensoren, welche die Temperatur, die Drehzahl und die Vibration der Motoren erfassen. Wie in Abbildung 3.1 dargestellt, sind die Sensoren direkt an den jeweiligen Motoren angebracht.

Die Anbindung der Sensorik an die SPS erfolgt teils über IO-Link (für die Temperatursensoren) und teils über ein spezielles Diagnosemodul, das VSE150 (Vibration Sensor Electronics), zur Erfassung und Aufbereitung von Schwingungsdaten. IO-Link ist ein standardisiertes Punkt-zu-Punkt-Kommunikationsprotokoll, dass eine bidirektionale Kommunikation zwischen Sensoren, Aktoren und der Steuerungsebene ermöglicht. Es basiert auf der herkömmlichen 3-Leiter-Anschlusstechnik und erlaubt neben der Über-

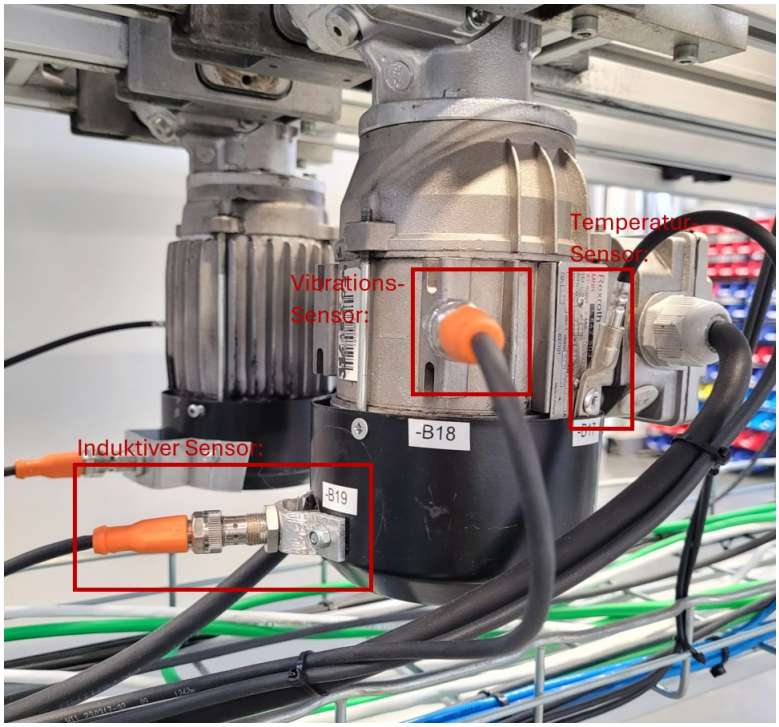



**Abbildung 3.1:** Sensoren an einer der Bandmotoren; eigene Darstellung

tragung von Prozessdaten auch die Parametrierung und Diagnose der angeschlossenen Geräte. [IO-25]

Die im Projekt eingesetzte SPS vom Typ Siemens S7-1200 verarbeitet die eingehenden Sensorsignale zyklisch im Rahmen ihres Steuerungsprogramms. Die verarbeiteten Informationen werden in Datenbausteinen abgelegt und sind dort über fest definierte Adressbereiche abrufbar. Die Verschaltung der Sensoren an der SPS ist schematisch in Abbildung 3.2 dargestellt. Dargestellt sind die vier Motoren, gekennzeichnet mit den Bezeichnungen M1 bis M4, sowie der IO-Link Master in unmittelbarer Nähe zum Rechner und das Überwachungsgerät VSE150, welches im Schaltschrank der SPS integriert ist. Die blauen Verbindungslinien repräsentieren die Anbindung der Temperatursensoren an den IO-Link Master, während die grünen Linien die Verbindung der Schwingungs- sowie Induktionssensoren mit dem VSE150 darstellen.

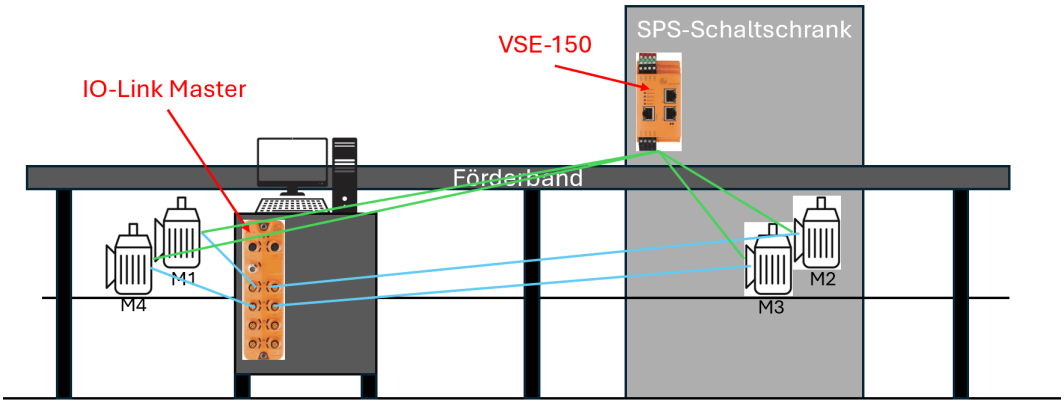



**Abbildung 3.2:** schematische Darstellung des Bandsystems mit der Verbindung der Sensoren an die SPS; eigene Darstellung

## 3.2 Edge-Computing mit Proxmox und Docker

Die Anbindung der außerhalb der SPS befindlichen Datenverarbeitungseinheit, einem Intel NUC, erfolgt über Industrial Ethernet. Der NUC fungiert als Edge-Device und stellt eine containerisierte Softwareumgebung auf Basis von Docker bereit. Innerhalb dieser Umgebung werden mehrere spezialisierte Container betrieben, darunter Node-RED zur visuellen Programmierung und Datenverarbeitung, Portainer zur Containerverwaltung, eine TimescaleDB für die zeitreihenbasierte Datenspeicherung sowie Grafana zur Visualisierung der erfassten Daten. Zur Vorbereitung der Umgebung wurde der Intel NUC zunächst mit dem Open-Source-Virtualisierungssystem Proxmox Virtual Environment ausgestattet. Proxmox ermöglicht die zentrale Verwaltung von virtuellen Maschinen und Containern über eine webbasierte Benutzeroberfläche, die über einen Rechner im selben Netzwerk durch Aufruf der zugewiesenen IP-Adresse im Browser zugänglich ist. Diese Funktionalität erleichtert nicht nur die Verwaltung des Systems selbst, sondern auch den Zugriff auf die innerhalb der Docker-Umgebung betriebenen Dienste. In Tabelle 3.1 sind die den jeweiligen Diensten zugewiesenen IP-Adressen aufgeführt, über die sie innerhalb des Netzwerks erreichbar sind.

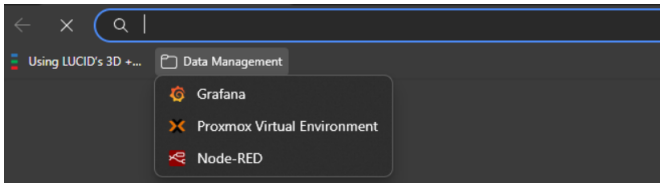
| Dienst | IP-Adresse/Zugriff | Benutzername | Passwort |
|---|---|---|---|
| Proxmox | 192.168.0.90/24 | root | password |
| Proxmox GUI | https://192.168.0.90:8006 | root | password |
| Debian | 192.168.0.91 | root | password |
| Portainer | http://192.168.0.91:9000 | root | password1234 |
| Node-RED | http://192.168.0.91:1880 | | |
| TimescaleDB | Zugriff via Node-RED, Peakboard, etc. | user | password |
| Grafana | http://192.168.0.91:3000 | admin | password |

**Tabelle 3.1:** Zugriffsadressen und Zugangsdaten der verwendeten Dienste

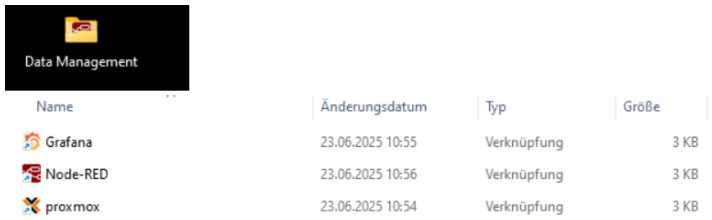
Alternativ besteht die Möglichkeit, den bereits am Bandumlaufsystem platzierten Rechner zu verwenden, da dieser bereits in das SPS-Netzwerk eingebunden ist und somit keine umfangreiche Konfiguration eines privaten Rechners erforderlich ist. Das Passwort zur Anmeldung an diesem Rechner lautet `qwertz`. Der Zugriff auf die relevanten Dienste kann auf unterschiedliche Weise erfolgen: entweder direkt über die entsprechenden Symbole in der Taskleiste (Abbildung 3.3a), mittels der Verknüpfungen im Ordner *Data Management* auf dem Desktop (Abbildung 3.3c) oder über die im Edge-Browser hinterlegten Seiten (Abbildung 3.3b).



**(a)** Zugriff über die Taskleiste




**(b)** Zugriff über den Edge-Browser




**(c)** Zugriff über den Ordner auf dem Desktop

**Abbildung 3.3:** Zugriff auf die Dienste über den Rechner am Transfersystem; eigene Darstellung

## 3.3 Datenfluss zwischen SPS, Node-RED und TimescaleDB

Ein wesentlicher Bestandteil der Datenverarbeitungskette stellt die Applikation Node-RED dar. Diese greift über das Kommunikationsprotokoll S7Comm – ein proprietäres Protokoll von Siemens zur Datenkommunikation über TCP/IP – auf die Datenbausteine der SPS zu. Die Verbindung wird in Node-RED mithilfe des Plugins „node-red-contrib-s7“ realisiert, welches eine parametergestützte Konfiguration (IP-Adresse, Rack, Slot, Verbindungsintervall) zur Herstellung der S7-Kommunikation erlaubt.

Node-RED übernimmt dabei nicht nur das Auslesen der Prozessdaten, sondern auch deren strukturelle Aufbereitung und Weiterleitung. Die aus der SPS extrahierten Werte werden im JSON-Format verarbeitet und anschließend in eine Datenbank überführt. Zu diesem Zweck wird auf dem Intel NUC ein separater Docker-Container mit einer TimescaleDB betrieben – einer auf PostgreSQL basierenden Erweiterung, die speziell für die Speicherung und Analyse von Zeitreihendaten entwickelt wurde.

Die Anbindung zwischen Node-RED und TimescaleDB erfolgt über standardisierte SQL-Befehle. Über den in Node-RED implementierten PostgreSQL-Out-Knoten werden strukturierte INSERT INTO-Anfragen an die Datenbank gesendet. Die erfassten Messwerte werden dabei in vordefinierten Tabellen zusammen mit einem Zeitstempel persistiert. Der Datenbankdienst lauscht auf dem Standardport 5432 und regelt den Zugriff über ein benutzerbasiertes Authentifizierungssystem.

Eine genaue Verarbeitung und Weiterleitung von Sensordaten innerhalb der Node-RED-Umgebung ist in Abbildung 3.4 visualisiert.

Die Messdaten umfassen Temperatur-, Drehzahl- und Schwingungswerte für die Motoren und werden zyklisch aus der SPS ausgelesen. Dies erfolgt über spezifische Zugriffsknoten, die das Siemens-Protokoll *S7Comm* nutzen und über fest definierte Speicheradressen die aktuellen Prozesswerte extrahieren. Die eingehenden Rohdaten werden in anschließenden Funktionsknoten konvertiert, um sie in ein für die Datenbank geeignetes numerisches

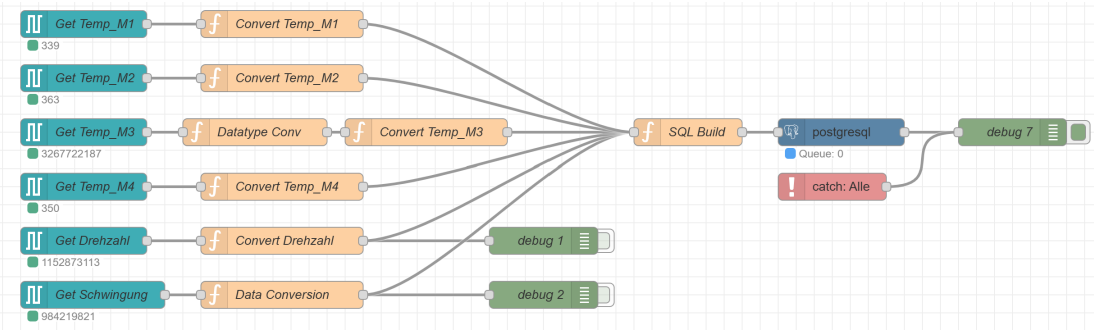



**Abbildung 3.4:** Datenverarbeitung mittels NodeRed; eigene Darstellung

Format zu überführen.

Die so aufbereiteten Messgrößen werden in einem zentralen *SQL Build*-Knoten zusammengeführt, wo sie in strukturierte SQL-Befehle übersetzt werden. Für jeden Motor wird dabei ein vollständiger Datensatz erzeugt, bestehend aus Temperatur, Drehzahl, Schwingung und einem Zeitstempel. Die Einträge werden im Format eines mehrzeiligen `INSERT INTO`-Statements gebündelt, wodurch eine performante Übertragung mehrerer Werte in einem einzigen Datenbankzugriff ermöglicht wird.

Die Weiterleitung erfolgt über den *PostgreSQL-Out*-Knoten, der die Abfrage an die in einem Docker-Container betriebene TimescaleDB übermittelt. Dort werden die Werte in einer Zeitreihentabelle persistiert. Ergänzend sorgen ein *Debug*-Knoten sowie ein *Catch*-Knoten für Fehlerbehandlung dafür, dass Datenintegrität und Übertragung zuverlässig überprüft werden können.

```
root@mqtt-server:~# docker exec -it timescaledb psql -U user -d timeseries
psql (14.17)
Type "help" for help.

timeseries=# Select * from motor_data ORDER BY time DESC LIMIT 15;
           time           | motor_id | temp |   rpm   | vibration
--------------------------+----------+------+---------+-----------
 2025-06-03 06:37:42.932+00 |        2 | 35.9 | 1152.38 |  0.002264
 2025-06-03 06:37:42.932+00 |        1 |   34 | 1152.48 |  0.002083
 2025-06-03 06:37:42.932+00 |        3 | 24.3 | 1152.88 |  0.001894
 2025-06-03 06:37:42.932+00 |        4 | 35.1 | 1155.55 |  0.001878
 2025-06-03 06:37:42.929+00 |        2 | 35.9 | 1152.38 |  0.002264
 2025-06-03 06:37:42.929+00 |        1 |   34 | 1152.48 |  0.002083
 2025-06-03 06:37:42.929+00 |        2 | 35.9 | 1152.38 |  0.002264
 2025-06-03 06:37:42.929+00 |        3 | 24.3 | 1152.88 |  0.001894
 2025-06-03 06:37:42.929+00 |        4 | 35.1 | 1155.55 |  0.001878
 2025-06-03 06:37:42.929+00 |        3 | 24.3 | 1152.88 |  0.001894
 2025-06-03 06:37:42.929+00 |        2 | 35.9 | 1152.38 |  0.002264
 2025-06-03 06:37:42.929+00 |        2 | 35.9 | 1152.38 |  0.002264
 2025-06-03 06:37:42.929+00 |        1 |   34 | 1152.48 |  0.002083
 2025-06-03 06:37:42.929+00 |        1 |   34 | 1152.48 |  0.002083
 2025-06-03 06:37:42.929+00 |        4 | 35.1 | 1155.55 |  0.001878
(15 rows)
```

**Abbildung 3.5:** Direkte Abfrage der erfassten Motordaten aus TimescaleDB über die Debian-Shell von Proxmox; eigene Darstellung

Abbildung 3.5 zeigt den direkten Zugriff auf die TimescaleDB-Datenbank über die Debian-Kommandozeile der virtualisierten Umgebung in Proxmox. Hierzu wird der Befehl `docker exec` genutzt, um in den laufenden TimescaleDB-Container zu wechseln und über die PostgreSQL-Schnittstelle (`psql`) auf die Datenbank *timeseries* zuzugreifen.

Die dargestellte SQL-Abfrage `SELECT * FROM motor_data ORDER BY time DESC LIMIT 15;` ruft die 15 aktuellsten Einträge aus der Tabelle `motor_data` ab. Diese Tabelle speichert die zeitlich erfassten Messwerte der vier Motoren, darunter Temperatur (`temp`), Drehzahl (`rpm`) und Schwingung (`vibration`). Dabei handelt es sich um die Werte, die von NodeRed ausgewertet und verarbeitet wurden. Zur eindeutigen Identifikation der Messwerte wird jedem Datensatz über das Attribut `motor_id` eine numerische Kennung zugewiesen, die den jeweiligen Motor referenziert. Auf diese Weise ist eine klare Zuordnung und differenzierte Auswertung der erfassten Daten im Nachgang gewährleistet.

Die Struktur der Daten belegt eine erfolgreiche Integration der einzelnen Systemkomponenten: Die Messdaten wurden korrekt aus der SPS ausgelesen, im Node-RED-Flow verarbeitet, mit Zeitstempeln versehen und über SQL-Befehle persistent in der Datenbank abgelegt. Durch die Nutzung der TimescaleDB – einer für Zeitreihendaten optimierten Erweiterung von PostgreSQL – lassen sich diese Daten effizient für spätere Analyse- und Visualisierungsaufgaben weiterverarbeiten.

## 3.4 Visualisierung und Statusüberwachung mittels Peakboard

Für die visuelle Darstellung der in der TimescaleDB gespeicherten Prozessdaten des Transfersystems wurde innerhalb dieses Projekts Peakboard eingesetzt. Peakboard ermöglicht die direkte Anbindung an die vorhandene Datenbank, sodass relevante Messwerte wie Temperatur, Schwingungen und Drehzahlen der Motoren übersichtlich und in Echtzeit dargestellt werden können. Ziel dieser Projektphase war es, das bestehende Dashboard (Abbildung 3.6) um zusätzliche Darstellungsseiten zu erweitern. Diese Seiten sollten eine übersichtliche grafische Darstellung der Sensordaten der Motoren mittels Diagrammen ermöglichen. Hierdurch wird eine detailliertere und präzisere Zustandsüberwachung der Motoren im Transfersystem gewährleistet.

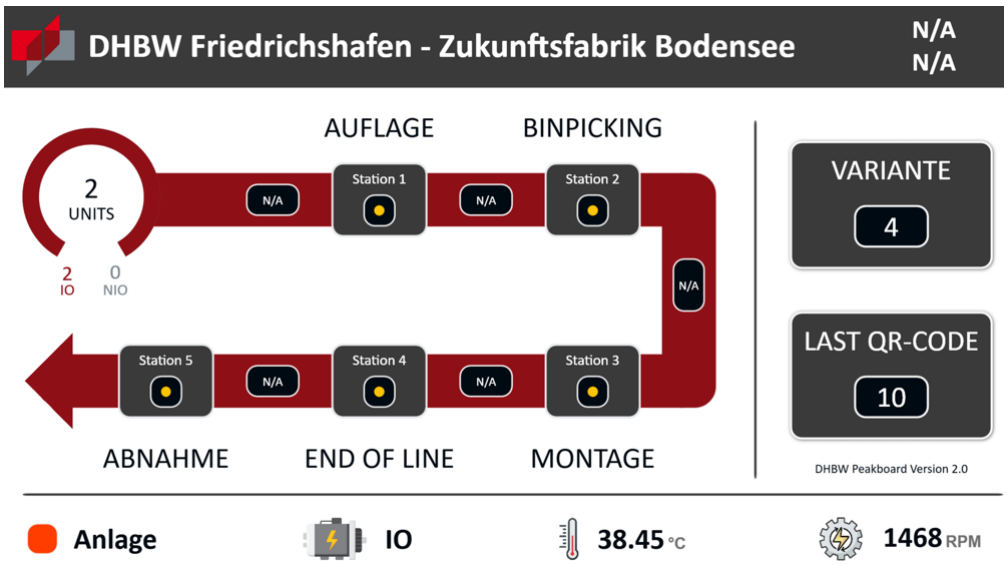



**Abbildung 3.6:** Dashboard des Bandsystems mithilfe von Peakboard auf dem Fernseher im PTZ; eigene Darstellung

Zur Erstellung und Verwaltung entsprechender Dashboards wird die Software Peakboard Designer verwendet. Trotz intensiver Abstimmung mit vorausgegangenen Projektteams, deren betreuenden Personen sowie dem offiziellen Support von Peakboard konnten die notwendigen Zugangsdaten für den Peakboard Designer nicht wiederhergestellt werden. Daher wurde in Abstimmung mit dem Peakboard-Support beschlossen, das Peakboard

mittels eines hierfür vorgesehenen USB-Sticks auf die Werkseinstellungen zurückzusetzen. Der dafür erforderliche USB-Stick wurde beim zuständigen Laboringenieur Herrn Müller hinterlegt und kann dort für die Durchführung des Resets abgeholt werden. Die Zugangsdaten für die Neuanmeldung nach erfolgtem Reset sind im Dokument „DeliverySheet DNDE-25-0116“ hinterlegt beziehungsweise in Abbildung 3.7 dargestellt. Zu beachten ist, dass durch diesen Reset alle zuvor konfigurierten Dashboards unwiderruflich verloren gehen.

| Box | Type – License<br>License Key | Mac Address Ethernet<br>Mac Address WiFi | User Name<br>Initial Password |
|---|---|---|---|
| PB7903 | NFR - Enterprise<br>Ln0ktm2+3jfuZogz7S1SuA== | 00-E0-4C-0A-C5-A6<br>C6-23-60-E0-23-6A | pbadmin<br>pbadmin |

**Abbildung 3.7:** Notwendige Daten nach Reset des Peakboards; eigene Darstellung

Aufgrund des begrenzten zeitlichen Rahmens war es nicht möglich, nach dem Zurücksetzen ein neues Dashboard zur Visualisierung der Transfersystem-Daten zu erstellen. Diese Aufgabe wird daher an nachfolgende Projektgruppen übergeben.

Um das zurückgesetzte Peakboard für die zukünftige Nutzung vorzubereiten, sind folgende initiale Schritte erforderlich:

1. Verbindung zum Peakboard über den Peakboard Designer herstellen, indem die IP-Adresse im Netzwerk identifiziert wird.

2. Zur erstmaligen Anmeldung lautet der Benutzername sowie das Initialisierungspasswort `pbadmin` (Siehe Abbildung 3.7). Im Anschluss daran erfolgt eine Aufforderung zur Festlegung neuer Nutzerdaten.

3. Im Designer grundlegende Einstellungen für Netzwerkzugriff und Authentifizierung konfigurieren.

4. Anschließend eine direkte Verbindung zur TimescaleDB herstellen und dabei IP-Adresse, Port sowie Datenbankzugangsdaten eingeben.

5. Erste Layout-Entwürfe erstellen, die auf die im Projekt relevanten Prozessdaten

(z. B. Temperatur- und Schwingungswerte der Motoren) zugeschnitten sind.

Diese Schritte legen die Grundlage, auf welcher nachfolgende Teams effizient an der konkreten Visualisierung weiterarbeiten können.

## 3.5 Systemübersicht der Datenverarbeitung

/

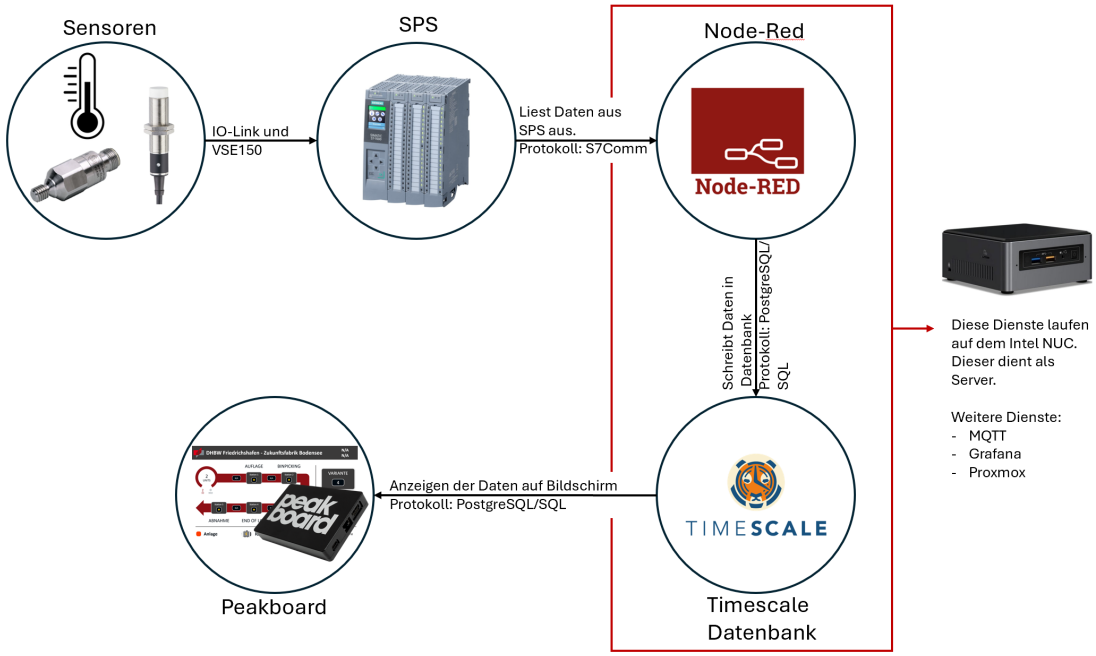



**Abbildung 3.8:** Systemübersicht der Datenverarbeitung und Visualisierung; eigene Darstellung

Die zuvor erläuterte Architektur zur Datenerfassung, -verarbeitung und -visualisierung bildet eine durchgängige Pipeline zur Echtzeitüberwachung industrieller Prozesse. Beginnend mit der physikalischen Erfassung relevanter Messgrößen – wie Temperatur, Schwingung und Drehzahl – durch Sensorik an den Bandmotoren, werden diese Daten zunächst über IO-Link beziehungsweise das Diagnosesystem VSE150 an die SPS übermittelt.

Node-RED übernimmt als zentrales Verarbeitungswerkzeug die Kommunikation mit der SPS, transformiert die Rohdaten in geeignete Datenstrukturen und überträgt sie automatisiert an eine Timescale-Datenbank. Diese fungiert als persistente Speicherlösung für Zeitreihendaten und erlaubt eine performante Weiterverarbeitung der Messwerte. Ergänzt wird die Architektur durch Visualisierungskomponenten wie Grafana und Peakboard, die unterschiedliche Zugriffspfade auf die erfassten Daten bereitstellen und so eine transparente Überwachung des Systemzustands ermöglichen.

Alle eingesetzten Dienste sind auf einem Edge-Device (Intel NUC) in Form containerisierter Anwendungen realisiert, was eine flexible, skalierbare und wartungsarme Systemintegration gewährleistet. Abbildung 3.8 veranschaulicht die vollständige Datenverarbeitungskette im Überblick – von der Sensorik über die SPS und die Verarbeitung in Node-RED bis hin zur Speicherung in TimescaleDB und anschließender Visualisierung.

# 4 Ausblick: Umsetzung der Datenverarbeitung mit MES FORCAM

In Kapitel 3 wurde eine modulare, containerbasierte Lösung zur Erfassung, Verarbeitung und Visualisierung von Sensordaten beschrieben, bestehend aus Komponenten wie Node-RED, TimescaleDB, Grafana und Peakboard. Ziel dieses Abschnitts ist es zu zeigen, wie diese Funktionalitäten – anstelle einer eigenentwickelten Infrastruktur – mit dem zukünftigen Manufacturing Execution System **FORCAM FORCE MES LITE** abgebildet werden können.

## 4.1 Sensordatenerfassung und Maschinenanbindung

Während im Projekt ein Intel NUC mit IO-Link und VSE150 zur Datenerfassung dient, kann die Maschinenanbindung bei FORCAM über standardisierte I/O-Controller realisiert werden. Diese erfassen grundlegende Zustandsinformationen wie:

- Maschine eingeschaltet / ausgeschaltet
- Automatikbetrieb AN / AUS
- Maschine produziert

- Maschine steht

Zur Integration zusätzlicher Signale wie Temperatur, Schwingung oder Drehzahl ist eine Erweiterung der Signalverarbeitung über dedizierte Eingänge bzw. Gateways möglich. FORCAM bietet Schnittstellen zur Einbindung solcher Daten über OPC UA oder Message Queuing Telemetry Transport (MQTT). Die zuvor über SPS und Node-RED erfassten Werte könnten somit direkt in das MES integriert werden, ohne eine Zwischenschicht wie Node-RED zu benötigen.

## 4.2 Datenverarbeitung und Speicherung

Statt auf eine manuell konfigurierte TimescaleDB zuzugreifen, übernimmt das MES die Datenpersistenz automatisch. Produktionsdaten, Maschinenstatus sowie Qualitätskennzahlen werden in FORCAM intern gespeichert und über eine leistungsfähige MES-Datenbank verarbeitet.

Kennzahlen wie:

- Maschinenverfügbarkeit
- Stillstandsgründe
- Overall Equipment Effectiveness (OEE)

werden automatisch ermittelt und stehen zur weiteren Auswertung bereit. Damit entfällt die Notwendigkeit eigener SQL-Abfragen und Datenbankkonfigurationen, wie sie im Docker-Stack implementiert wurden.

## 4.3 Visualisierung und Überwachung

Die mit Grafana und Peakboard umgesetzte Visualisierung wird durch die in FORCAM integrierten Dashboards ergänzt. Diese ermöglichen eine:

- Echtzeitvisualisierung der Maschinenzustände,
- Anzeige von KPIs (z. B. OEE),
- Analyse von Stillstandsmustern,
- Generierung von Berichten und Exporten.

Die Dashboards sind webbasiert zugänglich und bieten Benutzern rollenbasierten Zugriff. Sie können sowohl auf Arbeitsplatzrechnern als auch auf dem Fernseher im PTZ angezeigt werden – etwa über einen Browser im Kioskmodus oder mittels digitalem Signage. Dies ermöglicht eine transparente Darstellung wichtiger Produktionskennzahlen.

Trotz der in FORCAM integrierten Visualisierungsfunktionen bleibt **Peakboard** weiterhin eine Einsatzmöglichkeit. Es kann weiterhin genutzt werden, um individuell gestaltete Dashboards zu erstellen, die spezifisch auf die Anforderungen des Projekts und der Bedieneroberfläche zugeschnitten sind. Diese Dashboards werden wie gehabt auf dem Fernseher ausgegeben und visualisieren beispielsweise aktuelle Zustände, Prozesswerte oder Alarmmeldungen in Echtzeit. Die Datenbereitstellung erfolgt dabei über eine direkte Verbindung zur MES-Datenbank oder über MQTT – wie bereits im bestehenden System implementiert.

Durch die Kombination aus internen MES-Dashboards und Peakboard-Anzeigen entsteht eine flexible und vielseitige Visualisierungslandschaft, die sowohl zentrale Steuerung als auch lokale Übersicht unterstützt.

## 4.4 Vorteile der FORCAM-Integration

Durch den Einsatz von FORCAM ergeben sich folgende Vorteile gegenüber der Open-Source-Architektur:

- **Zentrale Plattform:** Einheitliche Lösung für Datenerfassung, Speicherung, Analyse und Visualisierung.
- **Skalierbarkeit:** Einfache Erweiterbarkeit auf weitere Maschinen oder Standorte.
- **Wartbarkeit:** Reduktion des Wartungsaufwands durch Wegfall individueller Containerlösungen.
- **Zuverlässigkeit:** Industriell erprobte Architektur mit Support und Sicherheitsupdates.

Gleichzeitig ist zu berücksichtigen, dass FORCAM in der Standardkonfiguration eine reduzierte Signaltiefe bietet (vier digitale Signale). Eine detailliertere Zustandsüberwachung erfordert jedoch eine gezielte Erweiterung der Signalverarbeitung.

# 5 Zusammenfassung und Fazit

Im Rahmen dieser Studienarbeit wurde ein Condition Monitoring-System für das Transfersystem der Zukunftsfabrik Bodensee entworfen und prototypisch umgesetzt. Ziel war es, eine automatisierte, skalierbare Lösung zur Erfassung, Verarbeitung und Visualisierung relevanter Sensordaten wie Temperatur, Schwingung und Drehzahl zu entwickeln. Die technische Umsetzung erfolgte mit modernen Open-Source-Werkzeugen wie Node-RED, TimescaleDB und Proxmox in einer containerisierten Infrastruktur auf einem Intel NUC.

Die entwickelte Architektur erfüllt die grundlegenden Anforderungen: Die Datenübertragung zwischen Sensorik, SPS, Edge-Gerät und Datenbank konnte stabil realisiert werden. Die Speicherung in der TimescaleDB erfolgte zuverlässig, und die Anbindung an Visualisierungssysteme wurde vorbereitet. Technische Herausforderungen wie Netzwerkprobleme, SQL-Syntaxfehler oder Containerkommunikation wurden erfolgreich behoben und sind umfassend dokumentiert.

Dennoch zeigen sich bei genauerer Betrachtung auch einige Limitationen: Die gewählte Lösung ist in Teilen komplex in der Wartung, insbesondere durch die Vielzahl eigenständig konfigurierter Container. Zudem erfordert die langfristige Pflege (z. B. Updates, Sicherheit, Monitoring) spezifisches Know-how. Die Visualisierung über Peakboard konnte nicht vollständig realisiert werden, was die Aussagekraft der Darstellung einschränkt. Auch Aspekte wie die Sicherheit der Datenübertragung oder die Belastbarkeit des Systems bei vielen gleichzeitigen Sensoren wurden nur am Rande betrachtet.

Zukünftige Arbeiten könnten daher insbesondere folgende Punkte adressieren:

- vollständige Integration in das FORCAM MES zur Vereinfachung von Verwaltung und Auswertung,
- erweiterte Visualisierung mit benutzerfreundlichen Dashboards,
- Ergänzung weiterer Sensorik zur Erhöhung der Diagnosegenauigkeit,
- Einsatz von Machine-Learning-Algorithmen zur frühzeitigen Anomalieerkennung.

Zusammenfassend wurde mit dieser Arbeit ein robuster Grundstein für ein vorausschauendes Instandhaltungssystem gelegt. Die Architektur ist modular, praxisnah und technisch nachvollziehbar dokumentiert. Sie bildet eine belastbare Grundlage für weitere Entwicklungen im Bereich Smart Maintenance.

# Literaturverzeichnis


[BZ22] M. Benker und M. F. Zaeh. „Condition monitoring of ball screw feed drives using convolutional neural networks“. In: *CIRP Annals - Manufacturing Technology* 71.1 (2022), S. 313–316. DOI: 10.1016/j.cirp.2022.03.017.

[FOR23] FORCAM ENISCO GmbH. *FORCE MES LITE: Die Einstiegslösung in die digitale Produktion*. Zugriff am 03. Juni 2025. 2023. URL: https://forcam-enisco.net/force-mes-lite/ (besucht am 03. 06. 2025).

[IO-25] IO-Link Community. *Was ist IO-Link?* Zugriff am 15. Mai 2025. 2025. URL: https://io-link.com/de/Technologie/Was_ist_IO-Link.php (besucht am 15. 05. 2025).

[iT 25] iT Engineering Software Innovations GmbH. *Anwendungsfall: Condition Monitoring*. Zugriff am 03. Juni 2025. IIoT Building Blocks. 2025. URL: https://iiotbuildingblocks.io/anwendungsfall-condition-monitoring/ (besucht am 03. 06. 2025).

[KB22] J. Kolerus und E. Becker. *Condition Monitoring und Instandhaltungsmanagement*. Renningen: Expert Verlag, 2022.

[Mob02] R. Keith Mobley. *An Introduction to Predictive Maintenance*. 2. Aufl. Amsterdam: Elsevier, 2002. ISBN: 9780750675314.

[Pre23] Presseportal. *FORCAM setzt weltweite Maschinenanbindung in einer SAP-Infrastruktur für Siemens Energy um*. Zugriff am 03. Juni 2025. 2023. URL: https://www.presseportal.de/pm/100391/5550877 (besucht am 03. 06. 2025).

# Abbildungsverzeichnis

2.1 Grafik zum Condition Monitoring. Quelle:[iT 25] . . . . . 3

3.1 Sensoren an einer der Bandmotoren; eigene Darstellung . . . . . 8
3.2 schematische Darstellung des Bandsystems mit der Verbindung der Sensoren an die SPS; eigene Darstellung . . . . . 9
3.3 Zugriff auf die Dienste über den Rechner am Transfersystem; eigene Darstellung . . . . . 10
3.4 Datenverarbeitung mittels NodeRed; eigene Darstellung . . . . . 12
3.5 Direkte Abfrage der erfassten Motordaten aus TimescaleDB über die Debian-Shell von Proxmox; eigene Darstellung . . . . . 13
3.6 Dashboard des Bandsystems mithilfe von Peakboard auf dem Fernseher im PTZ; eigene Darstellung . . . . . 14
3.7 Notwendige Daten nach Reset des Peakboards; eigene Darstellung . . . . 15
3.8 Systemübersicht der Datenverarbeitung und Visualisierung; eigene Darstellung . . . . . 16

A.1 Netzwerkkonfiguration in das SPS-Netzwerk; eigene Darstellung . . . . . 28

# Abkürzungsverzeichnis

**SPS** speicherprogrammierbare Steuerung

**CM** Condition Monitoring

**MES** Manufacturing Execution Systeme

**OEE** Overall Equipment Effectiveness

**MQTT** Message Queuing Telemetry Transport

**NUC** Next Unit of Computing

# A Dokumentation technischer Erfahrungen und Problemlösungen

Im Verlauf der Umsetzung der datenbasierten Zustandsüberwachung mithilfe von Open-Source-Technologien wie Proxmox, Node-RED und TimescaleDB wurden zahlreiche technische Herausforderungen identifiziert und gelöst. Das vorliegende Kapitel dokumentiert zentrale Erkenntnisse, Fehlerquellen sowie bewährte Lösungsstrategien in Form eines strukturierten Erfahrungsarchivs. Ziel ist es, das gewonnene Wissen systematisch zugänglich zu machen und damit künftige Weiterentwicklungen, Fehleranalysen oder Systemanpassungen effizient zu unterstützen.

## A.1 Virtuelle Infrastruktur mit Proxmox

Proxmox Virtual Environment diente als Plattform zur Verwaltung der virtualisierten Systemkomponenten. Eine häufige Fehlerquelle war die Konfiguration der Netzwerkschnittstellen, insbesondere wenn virtuelle Maschinen in unterschiedlichen Subnetzen betrieben wurden. Um Dienste innerhalb einer virtuellen Maschine (z. B. Debian mit TimescaleDB) über das Hauptnetzwerk zugänglich zu machen, musste die Bridge-Schnittstelle korrekt eingebunden und die IP-Adressvergabe explizit gesetzt werden. Der Zugriff auf die Proxmox-Weboberfläche war nur im passenden Subnetz erreichbar.

Zur Herstellung einer Verbindung mit dem industriellen Netzwerk, in dem die SPS sowie der Intel NUC integriert sind, kann ein externer Rechner physisch in das Netzwerk eingebunden werden. Dies erfolgt typischerweise durch den Anschluss eines LAN-Kabels an einen freien Port eines Netzwerkswitches, beispielsweise einer Siemens Scalance XB008, die im Schaltschrank verbaut ist. Das gegenüberliegende Kabelende wird mit dem Rechner verbunden – gegebenenfalls unter Zuhilfenahme eines Adapters (LAN-zu-USB-A bzw. LAN-zu-USB-C), falls das Endgerät über keinen integrierten Ethernetanschluss verfügt.

Im Anschluss ist es erforderlich, die Netzwerkkonfiguration des Rechners manuell anzupassen. In den Eigenschaften der Netzwerkverbindung muss eine statische IP-Adresse vergeben werden, die sich innerhalb desselben Subnetzes wie die SPS und der NUC befindet, um eine direkte Kommunikation zu ermöglichen. Eine beispielhafte Konfiguration ist in Abbildung A.1 dargestellt.

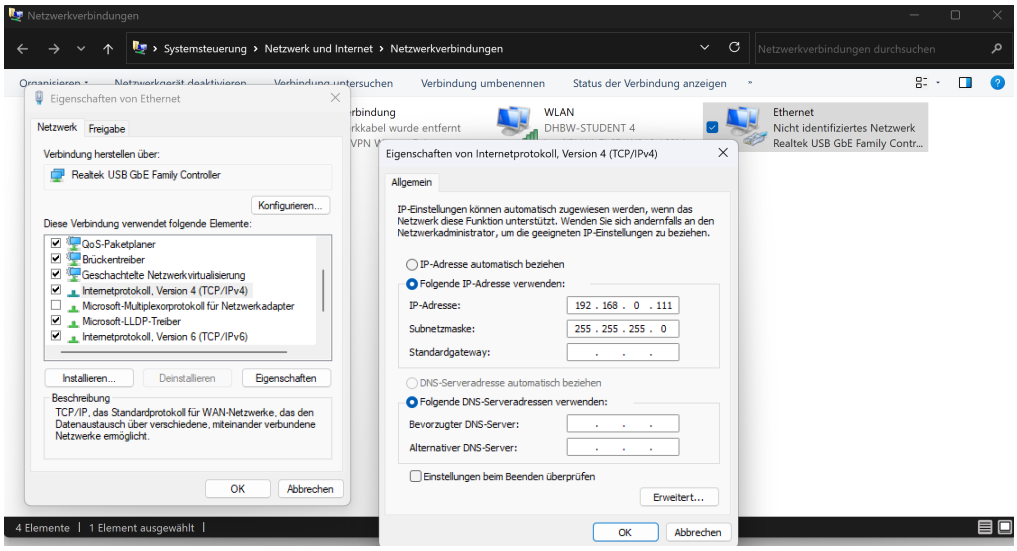



**Abbildung A.1:** Netzwerkkonfiguration in das SPS-Netzwerk; eigene Darstellung

Nach erfolgreicher Integration in das Netzwerk können die verschiedenen Dienste – wie in Tabelle 3.1 aufgeführt – über die jeweilige IP-Adresse im Webbrowser aufgerufen werden.

## A.2 Debian-Umgebung und Systemzugriffe

Innerhalb der Debian-VM traten mehrfach Probleme bei der Installation von Softwarepaketen aufgrund fehlender Internetverbindung auf. Dies war auf das isolierte Subnetz der VM zurückzuführen. Temporäres Umschalten auf das Internet-Subnetz ermöglichte die erfolgreiche Installation fehlender Komponenten (Docker, Node.js, PostgreSQL etc.). Nach Abschluss wurde die IP-Adresse zurück in das produktive Subnetz der SPS gesetzt. Für die Installation internetabhängiger Komponenten ist es erforderlich, sowohl den Hypervisor Proxmox als auch die darauf betriebene Debian-VM in ein Netzwerk mit Internetzugang einzubinden. Zu diesem Zweck wurde ein privater Rechner verwendet, der über WLAN mit dem Internet verbunden war und zusätzlich per LAN-Kabel mit dem Intel NUC verbunden wurde.

**Internetfreigabe über Windows** Unter Windows kann der Internetzugang des WLAN-Adapters über die sogenannte "Internetverbindungsfreigabe" für andere Netzwerkschnittstellen bereitgestellt werden. Dazu wird in den Netzwerkeinstellungen mit einem Rechtsklick auf den WLAN-Adapter das Menü *Eigenschaften* geöffnet. Im Reiter *Freigabe* lässt sich die Verbindung für den entsprechenden LAN-Adapter aktivieren. Dadurch erhält das LAN-Netzwerk eine neue IP-Adresskonfiguration, die typischerweise dem Bereich `192.168.137.0/24` zugeordnet ist.

**Konflikt mit bestehendem SPS-Netzwerk** Dabei ist zu beachten, dass dieses neue Netzwerk in einem anderen Subnetz als das bestehende SPS-Netzwerk liegt. Um Proxmox und die Debian-VM temporär in das internetfähige Subnetz zu verschieben, müssen die Netzwerkeinstellungen entsprechend angepasst werden.

**Netzwerkkonfiguration in Proxmox** Zunächst wird der Intel NUC über HDMI mit einem Monitor verbunden, um direkten Zugriff auf die Proxmox-Konsole zu ermöglichen. Über den Befehl `ip a` kann die aktuelle Netzwerkkonfiguration eingesehen werden. Zur Konfiguration des Netzwerks wird die Datei `/etc/network/interfaces` mit dem Editor `nano` geöffnet:

```
nano /etc/network/interfaces
```

In diesem Konfigurationsfile kann das virtuelle Bridge-Interface `vmbr0` so angepasst werden, dass es in das durch die Internetfreigabe bereitgestellte Subnetz passt. Dafür müssen IP-Adresse, Gateway und Netzmaske entsprechend geändert werden. Nach dem Speichern mit `Strg+O` und dem Verlassen des Editors mit `Strg+X` wird das Netzwerk über den Befehl

```
systemctl restart networking
```

neu gestartet. Die erfolgreiche Verbindung zum Internet kann beispielsweise über einen `ping`-Test auf einen externen Server (z. B. `8.8.8.8`) überprüft werden.

**Netzwerkanpassung in der Debian-VM** Analog dazu muss auch in der Debian-VM die Netzwerkkonfiguration angepasst werden. Dies geschieht über die grafische Benutzeroberfläche von Proxmox, indem im Container die entsprechenden Netzwerke editiert und mit den passenden Adressdaten versehen werd. Nach einem Reboot der VM oder einem Neustart des Netzwerkdienstes steht auch dort Internetzugang zur Verfügung.

**Ergebnis** Nach erfolgreicher Konfiguration sind sowohl Proxmox als auch die Debian-VM über das temporäre Internet-Subnetz erreichbar und können internetbasierte Installationen (z. B. Docker-Images oder Paketquellen) durchführen. Nach Abschluss aller Installationen kann das Netzwerk wieder in das ursprüngliche SPS-Subnetz zurückgeführt werden.

## A.3 Node-RED – Konfiguration und Fehleranalyse

Beim Aufbau der Datenpipeline mit Node-RED traten verschiedene Fehlermeldungen auf. Dazu zählten unter anderem Syntaxfehler in SQL-Statements (“syntax error at or near ’;’“) sowie Netzwerkprobleme beim Zugriff auf Datenbankcontainer (“EAI_AGAIN“). Letzteres ließ sich durch die Verwendung von IP-Adressen anstelle von Container-Namen sowie die Nutzung des Host-Netzwerkmodus in Docker beheben. Zusätzlich war auf die konsistente Übergabe aller Daten (z. B. Temperaturwerte, Drehzahl) an den SQL-Build-Knoten zu achten. Fehlende Werte führten zu nicht ausführbaren SQL-Befehlen.

Da sowohl Node-RED als auch die TimescaleDB nicht über eine gemeinsame `docker-compose`-Konfiguration, sondern als eigenständige Docker-Container betrieben wurden, traten anfänglich Schwierigkeiten bei der korrekten Konfiguration des Hostnamens für die Verbindung von Node-RED zur TimescaleDB auf.

Die Entscheidung, auf eine `docker-compose`-Lösung zu verzichten, basierte auf der Anforderung, dass alle Dienste nach einem Neustart des Intel NUC automatisch wieder gestartet werden sollten. In der Praxis zeigte sich, dass die Container in einer `docker-compose`-Umgebung diesen automatischen Neustart nicht zuverlässig ausführten.

Um dennoch eine stabile Kommunikation zwischen den Containern zu gewährleisten, wurde ein dediziertes Docker-Subnetz erstellt, in das sowohl Node-RED als auch die TimescaleDB eingebunden wurden. Durch die Integration beider Container in dasselbe Netzwerk konnte eine zuverlässige Schnittstellenkommunikation sichergestellt werden.

## A.4 TimescaleDB – Zugriff und Tabellenstruktur

Für die Zeitreihendatenbank TimescaleDB mussten zunächst über `psql`-Befehle Tabellen angelegt und anschließend mittels `create_hypertable` in das Timescale-Format überführt werden. Ein häufiger Fehler bestand darin, beim Hypertable-Aufruf eine nicht existierende Tabelle zu referenzieren. Die containerbasierte Ausführung der TimescaleDB erforderte den gezielten Zugriff via `docker exec` und war über Port 5432 erreichbar. Für Schreiboperationen aus Node-RED wurde SSL deaktiviert.

## A.5 Zusammenfassung der Lessons Learned

- Netzwerkzugriffe zwischen Containern erfordern explizite Docker-Netzwerke oder den Host-Modus.
- Der PostgreSQL-Out-Node in Node-RED akzeptiert keine abschließenden Semikolons im SQL-String.
- IP-Adressen sind stabiler als DNS-Hostnamen in Docker.
- `flow.set`/`get` in Node-RED muss konsistent verwendet werden, sonst fehlen Werte zur SQL-Erzeugung.
- Eine durchdachte Strukturierung von Volumes und Netzwerken erleichtert sowohl Fehlersuche als auch Wartung deutlich.