

# Implementierung eines Tracking Systems für Produkte auf einer Fertigungskette

## Studienarbeit T3\_3101

Studiengang Embedded Systems

Duale Hochschule Baden-Württemberg Ravensburg, Campus Friedrichshafen

von

Kilian Röper

Abgabedatum:	18. Mai 2025
Bearbeitungszeitraum:	01.10.2024 - 15.07.2025
Matrikelnummer:	5928177
Kurs:	TSA22
Betreuerin / Betreuer:	Markus Wengle

# 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\_3101 mit dem Thema:

*Implementierung eines Tracking Systems für Produkte auf einer Fertigungskette*

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 18. Mai 2025

---

Kilian Röper



# Kurzfassung

Die Lernfabrik produziert teilautomatisiert über eine Fertigungskette verschiedene Produkte. Zur Verbesserung der Steuerung, Nachverfolgbarkeit und Dokumentation soll ein System entwickelt werden, das die Position der Produkte mittels NFC-Technologie erfasst und die entsprechenden Daten an ein Manufacturing Execution System (MES) übermittelt. Während geeignete NFC-Tags bereits identifiziert wurden, muss die restliche Systemarchitektur standortspezifisch konzipiert, implementiert und getestet werden.

Ziel dieser Arbeit ist die Entwicklung eines industrietauglichen Systems zur positionsbasierten Produktverfolgung innerhalb der Fertigung. Die erfassten Informationen sollen in Echtzeit an ein vorhandenes System übertragen und dort visualisiert werden.

Zur Entwicklung der Systemarchitektur kamen klassische ingenieurwissenschaftliche Methoden zum Einsatz. Der Fokus lag auf einem kosteneffizienten, störungstoleranten und modular erweiterbaren Aufbau der Hardware- und Softwarekomponenten.

Die Arbeit mündete in einem vollständigen Systementwurf zur produktionsbegleitenden Datenerfassung über NFC. Realisiert wurde ein Prototyp bestehend aus dezentralen Erfassungseinheiten und einer zentralen Steuereinheit, die eine Schnittstelle zum MES bereitstellt. Neben der Hardware wurden die erforderlichen Softwarekomponenten zur Datenübertragung und -visualisierung sowie eine technische Dokumentation für zukünftige Erweiterungen erstellt.

# Abstract

The learning factory produces various products semi-automatically via a production chain. To improve control, traceability, and documentation, a system is to be developed that records the position of the products using NFC technology and transmits the corresponding data to a Manufacturing Execution System (MES). While suitable NFC tags have already been identified, the remaining system architecture must be designed, implemented, and tested site-specifically.

The goal of this work is to develop an industrial-grade system for position-based product tracking within production. The recorded information is to be transmitted in real time to an existing system and visualized there.

Classical engineering methods were used to develop the system architecture. The focus was on a cost-effective, fault-tolerant, and modularly expandable design of the hardware and software components.

The work culminated in a complete system design for production-related data acquisition via NFC. A prototype was realized consisting of decentralized acquisition units and a central control unit that provides an interface to the MES. In addition to the hardware, the necessary software components for data transmission and visualization as well as technical documentation for future extensions were created.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Grundlagen</b>	<b>3</b>
2.1	RFID und NFC-Technologie . . . . .	3
2.2	Das MQTT-Protokoll . . . . .	5
2.3	Node-Red - Eine Steuerungs- und Visualisierungssoftware . . . . .	8
2.4	Speicherprogrammierbare Steuerungen (SPS) . . . . .	9
2.4.1	Aufbau einer SPS . . . . .	9
2.4.2	Funktionsweise: Der SPS-Zyklus . . . . .	10
2.4.3	Programmiersprachen nach IEC 61131-3 . . . . .	11
2.4.4	Einsatz im Kontext dieser Arbeit . . . . .	11
<b>3</b>	<b>Konzeptentwurf</b>	<b>13</b>
3.1	Anforderungsanalyse mittels Requirements Engineering . . . . .	14
3.2	Hardwarebewertung mithilfe einer Nutzwertanalyse . . . . .	14
3.3	Systementwurf . . . . .	14
<b>4</b>	<b>Ergebnisse</b>	<b>15</b>
4.1	Umsetzung des Softwaredesigns auf den Mikrocontrollern . . . . .	15
4.2	Visualisierung der Prozessdaten in Node-Red . . . . .	15
<b>5</b>	<b>Zusammenfassung</b>	<b>16</b>
	<b>Literaturverzeichnis</b>	<b>17</b>
	<b>Verzeichnis verwendeter Formelzeichen und Abkürzungen</b>	<b>18</b>

<b>Abbildungsverzeichnis</b>	<b>19</b>
<b>Tabellenverzeichnis</b>	<b>20</b>
<b>A Nutzung von Künstliche Intelligenz basierten Werkzeugen</b>	<b>21</b>
<b>B Ergänzungen</b>	<b>22</b>
B.1 Details zu bestimmten theoretischen Grundlagen . . . . .	22
B.2 Weitere Details, welche im Hauptteil den Lesefluss behindern . . . . .	22
B.3 Ausführung von Quellcode . . . . .	22
<b>C Details zu Laboraufbauten und Messergebnissen</b>	<b>23</b>
C.1 Versuchsanordnung . . . . .	23
C.2 Liste der verwendeten Messgeräte . . . . .	23
C.3 Übersicht der Messergebnisse . . . . .	23
C.4 Schaltplan und Bild der Prototypenplatine . . . . .	23
<b>D Zusatzinformationen zu verwendeter Software</b>	<b>24</b>
D.1 Struktogramm des Programmentwurfs . . . . .	24
D.2 Wichtige Teile des Quellcodes . . . . .	24
<b>Sachwortverzeichnis</b>	<b>25</b>

# 1 Einleitung

Mit der zunehmenden Automatisierung industrieller Produktionsprozesse wächst der Bedarf an intelligenten Systemen, die eine lückenlose Nachverfolgbarkeit und effiziente Steuerung ermöglichen. Besonders in vernetzten Fertigungsumgebungen, wie sie im Kontext von Industrie-4.0 realisiert werden, spielt die automatische Identifikation von Produkten entlang der Fertigungskette eine zentrale Rolle. Zum Einsatz kommen dafür verschiedene Technologien - von optischer Erkennung bis hin zu RFID und NFC (Near Field Communication).

Die Lernfabrik der DHBW bietet eine reale, teilautomatisierte Fertigungsumgebung, in der solche Konzepte unter praxisnahen Bedingungen erprobt werden können. Aktuell fehlt jedoch ein System zur durchgängigen Identifikation und Nachverfolgung von Produkten innerhalb der Fertigungslinie. Dies erschwert nicht nur die Prozessüberwachung, sondern limitiert auch Möglichkeiten zur Datenanalyse, Optimierung und Dokumentation.

Ziel dieser Arbeit ist die Entwicklung eines Systems, das die Position und Identität von Produkten während des Fertigungsprozesses mit Hilfe von NFC-Sensoren automatisch erfasst. Zusätzlich soll es möglich sein Informationen automatisch auf die NFC-Tag zu schreiben. Die erfassten Daten werden an ein bestehendes Manufacturing Execution System (MES) übermittelt und dort zur Visualisierung bereitgestellt. Darüber hinaus sollen die von den NFC-Sensoren erfassten Informationen genutzt werden, um den logischen Ablauf des Bandumlaufsystems zu steuern. Hierfür ist ein Kommunikationsmechanismus zwischen den einzelnen Bandumlaufstationen und den NFC-Sensoren zu entwerfen. Im Rahmen dieser Arbeit wird eine geeignete Systemarchitektur für die Hard- und Softwarekomponenten konzipiert, implementiert und evaluiert.



Die zentrale Fragestellung der Arbeit lautet:

*Wie kann ein robustes, kosteneffizientes und erweiterbares Trackingsystem für Produkte in einer bestehenden Fertigungsumgebung mithilfe von NFC-Technologie gestaltet und in ein MES integriert werden?*

Zur Beantwortung dieser Frage werden klassische ingenieurwissenschaftliche Methoden wie Anforderungsanalyse, Systementwurf, prototypische Umsetzung und Validierung angewendet. Neben der technischen Umsetzung liegt ein besonderes Augenmerk auf der industriellen Tauglichkeit sowie der zukünftigen Erweiterbarkeit des Systems.

Die Arbeit gliedert sich wie folgt: Kapitel 2 beschreibt die theoretischen und technischen Grundlagen der verwendeten Technologien. Kapitel 3 analysiert die Anforderungen an das System und erläutert das methodische Vorgehen. Kapitel 4 enthält den Entwurf der Systemarchitektur sowie die Darstellung und Auswertung der Ergebnisse. Abschließend fasst Kapitel 5 die wesentlichen Erkenntnisse zusammen und gibt einen Ausblick auf mögliche Weiterentwicklungen.

## 2 Grundlagen

Dieses Kapitel stellt die theoretischen und technischen Grundlagen bereit, die für das Verständnis und die Umsetzung der im weiteren Verlauf beschriebenen Konzepte und Lösungen erforderlich sind. Dazu gehören sowohl technologische Aspekte – wie die Funktionsweise von RFID/NFC-Systemen oder des MQTT-Protokolls – als auch methodische Grundlagen im Bereich der Kommunikation und Steuerung verteilter Systeme.

Ziel ist es, ein solides Fundament zu schaffen, auf dem die spätere Systemkonzeption und Implementierung aufbauen kann. Die hier dargestellten Inhalte stehen daher in engem Zusammenhang mit den folgenden Kapiteln und werden dort zur Anwendung gebracht – beispielsweise bei der Auswahl geeigneter Kommunikationsprotokolle, Hardwarekomponenten oder der Integration in die Gesamtarchitektur des Systems.

### 2.1 RFID und NFC-Technologie

Near Field Communication (NFC) ist eine auf der RFID-Technologie (radio-frequency identification) basierende drahtlose Kommunikationstechnologie, die speziell für die Kommunikation über kurze Distanzen von wenigen Zentimetern entwickelt wurde. Sie ermöglicht den kontaktlosen Datenaustausch zwischen zwei NFC-fähigen Geräten, wobei ein Gerät aktiv (z.B. ein Smartphone) und das andere passiv (z.B. ein NFC-Tag) sein kann. Wesentliche Unterschiede zwischen RFID und NFC sind in Tabelle 2.1 festgehalten.

**Tabelle 2.1:** Vergleich zwischen RFID und NFC

Eigenschaft	RFID	NFC
Frequenzbereich	Verschiedene Bereiche: LF (125–134 kHz), HF (13,56 MHz), UHF (860–960 MHz)	HF (13,56 MHz)
Kommunikations-Richtung	In der Regel unidirektional (Reader zu Tag)	Bidirektional (Peer-to-Peer möglich)
Reichweite	Bis zu mehreren Metern (je nach Frequenz)	Typisch 0–10 cm
Datenrate	Abhängig vom Standard, z.B. bis zu 640 kbit/s (UHF)	Bis zu 424 kbit/s
Typische Anwendungen	Lagerlogistik, Zutrittskontrolle, Tierkennzeichnung, Supply Chain	Mobile Payment, Geräteverbindung, Zugangskontrolle im Konsumerbereich
Standardisierung	ISO/IEC 18000, ISO/IEC 14443, 15693 usw.	ISO/IEC 14443, ISO/IEC 18092, NFC Forum Spezifikationen
Interoperabilität	Eingeschränkt (herstellerspezifisch, viele Standards)	Hoch (interoperabel durch NFC Forum Vorgaben)

NFC arbeitet in einem Frequenzbereich von 13,56MHz und ermöglicht Datenraten von bis zu 424kbit/s. Der wichtigste Unterschied zu klassischen RFID-Systemen liegt in der bidirektionalen Kommunikation: Während RFID meist eine unidirektionale Leser-Tag-Kommunikation darstellt, ermöglicht NFC einen wechselseitigen Datenaustausch. Die typische maximale Reichweite beträgt etwa 10cm, wodurch eine gezielte, sichere Interaktion gewährleistet wird [Fin23, Seite 47 Tabelle 3.6].

Die Technologie kommt u.a. bei mobiler Bezahlung (z.B. Apple Pay), elektronischen Zugangssystemen, öffentlichen Verkehrsmitteln sowie bei der schnellen Geräteverbindung (z.B. Bluetooth-Kopplung) zum Einsatz. In dieser Arbeit wird die Technologie zum Austausch von Informationen zwischen NFC-Readern und Tags an einer Fertigungskette verwendet. In Kapitel 3 wird genauer darauf eingegangen wie hierfür die Integration in die Gesamtarchitektur erfolgt. Zur Implementierung von NFC-basierten

Funktionen stehen für Embedded-Systeme verschiedene Softwarebibliotheken zur Verfügung (vgl. 3.3). Diese ermöglichen etwa das Auslesen und Schreiben von NFC-Tags oder das Initiieren von Peer-to-Peer-Verbindungen.

## 2.2 Das MQTT-Protokoll

MQTT (Message Queuing Telemetry Transport) ist ein Client-Server-Publish/Subscribe-Nachrichtentransportprotokoll. Es ist leichtgewichtig, offen, einfach und leicht zu implementieren. Diese Eigenschaften prädestinieren es für den Einsatz in vielen Situationen, auch in eingeschränkten Umgebungen, wie z. B. für die Kommunikation im Machine-to-Machine- (M2M) und Internet of Things-Kontext (IoT), wo ein geringer Code-Footprint erforderlich ist und/oder die Netzwerkbandbreite knapp ist [OAS19, Abstract].

Das Protokoll läuft über TCP/IP oder andere Netzwerkprotokolle, die geordnete, verlustfreie und bidirektionale Verbindungen ermöglichen. Zu seinen Funktionen gehören:

- Verwendung des Publish/Subscribe-Nachrichtenmusters, das eine One-to-Many-Nachrichtenverteilung und die Entkopplung von Anwendungen ermöglicht.
- Drei Dienstqualitäten für die Nachrichtenübermittlung
- Ein Nachrichtentransport, der vom Inhalt der Nutzlast unabhängig ist.
- Geringer Transportaufwand und minimierter Protokollaustausch reduzieren den Netzwerkverkehr.
- Ein Mechanismus zur Benachrichtigung interessierter Parteien bei einer ungewöhnlichen Verbindungsunterbrechung.

Das Protokoll wurde mit dem Ziel entwickelt, eine möglichst leichtgewichtige und ressourcenschonende Kommunikation zu ermöglichen, insbesondere über Netzwerke mit hoher Latenz oder geringer Bandbreite. Im Vergleich zu traditionellen Client-Server-Modellen verwendet MQTT das Publish-/Subscribe-Prinzip, das eine entkoppelte Kommunikation erlaubt. In dieser Arbeit wird MQTT als Kommunikationsschicht zwischen Sensoren und einem Raspberry Pi eingesetzt [3.3].

Im Folgenden werden die zentralen Konzepte des Protokolls näher erläutert: das Publish-/Subscribe-Prinzip, die Rollen von Client und Broker sowie die verschiedenen Quality-of-Service-Stufen.

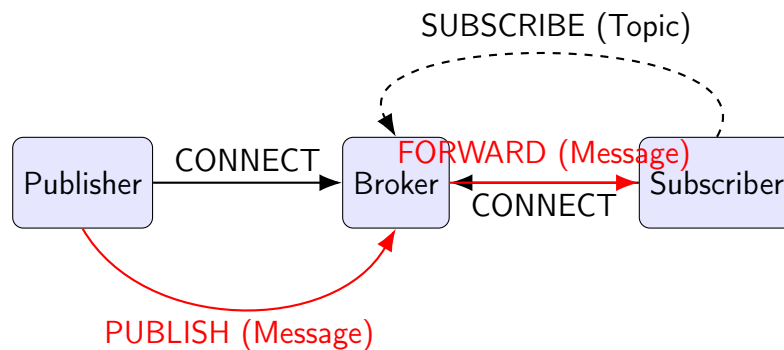
### **Publish-/Subscribe-Prinzip**

MQTT basiert auf einem Publish-/Subscribe-Modell. Ein *Publisher* sendet Nachrichten an ein bestimmtes *Topic*, ohne dabei explizit anzugeben, wer diese Nachrichten empfangen soll. Topics dienen in MQTT als hierarchisches Adressierungsschema zur Organisation von Nachrichten. Sie sind UTF-8-kodierte Strings, die durch Schrägstriche ("/") strukturiert sind, z.B. `sensor/temperatur/wohnzimmer`. Ein *Subscriber* hingegen abonniert ein oder mehrere Topics und erhält automatisch alle Nachrichten, die zu diesen Topics veröffentlicht werden [Abbildung 2.1].

### **Client und Broker**

Die zentrale Vermittlungsinstanz in diesem Modell ist der sogenannte *Broker*. Er empfängt alle vom Publisher gesendeten Nachrichten und leitet sie an die entsprechenden Subscriber weiter. Jeder Teilnehmer in einem MQTT-Netzwerk ist ein *Client*, unabhängig davon, ob er Nachrichten veröffentlicht oder abonniert. Der Broker übernimmt die vollständige Verwaltung der Kommunikation: Er sorgt dafür, dass Nachrichten zuverlässig und effizient verteilt werden, ohne dass sich Publisher und Subscriber gegenseitig kennen oder direkt kommunizieren müssen. Diese Architektur entkoppelt

Sender und Empfänger sowohl räumlich als auch zeitlich [OAS19, Abschnitt 4.7] [Abbildung 2.1].



**Abbildung 2.1:** Ablauf des MQTT Publish-/Subscribe-Verfahrens mit Verbindungsaufbau

### Quality of Service (QoS)

MQTT bietet drei verschiedene Stufen der Nachrichten-Zustellungssicherheit (Quality of Service), die je nach Anwendungsfall gewählt werden können:

- **QoS 0** – "höchstens einmal": Nachrichten werden nach bestem Wissen und Gewissen der Betriebsumgebung zugestellt. Nachrichtenverlust kann auftreten. Diese Stufe kann beispielsweise bei Umgebungssensordaten eingesetzt werden, bei denen der Verlust eines einzelnen Messwerts unerheblich ist, da der nächste kurz darauf veröffentlicht wird.
- **QoS 1** – "mindestens einmal": Nachrichten werden garantiert ankommen, es können aber Duplikate auftreten.
- **QoS 2** – "genau einmal": Nachrichten werden garantiert genau einmal ankommen. Diese Stufe kann beispielsweise bei Abrechnungssystemen eingesetzt werden, bei denen doppelte oder verlorene Nachrichten zu falschen Gebühren führen können. [OAS19, Abschnitt 4.3].

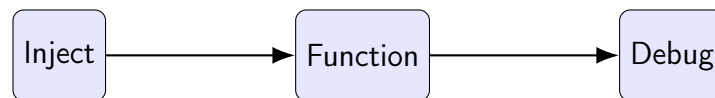
## 2.3 Node-Red - Eine Steuerungs- und Visualisierungssoftware

Node-RED ist ein von IBM entwickeltes Open-Source-Tool zur visuellen Programmierung, das auf Node.js basiert. Es ermöglicht die Erstellung von Anwendungen durch das Verbinden sogenannter Nodes in einem webbasierten Editor. Ursprünglich für das Internet der Dinge (IoT) konzipiert, findet Node-RED heute Anwendung in Bereichen wie Heimautomation, industrieller Steuerung und Datenintegration [Nod24].

Die zentrale Einheit in Node-RED ist der Flow, eine Abfolge von miteinander verbundenen Nodes, die Daten verarbeiten und weiterleiten. Jede Node erfüllt dabei eine spezifische Funktion, z.B. das Empfangen von Daten, deren Verarbeitung oder das Senden von Ergebnissen. Die Kommunikation zwischen den Nodes erfolgt über standardisierte Nachrichtenobjekte im JSON-Format, die typischerweise folgende Eigenschaften enthalten:

- msgid: Eine eindeutige Kennung der Nachricht
- payload: Die eigentlichen Nutzdaten
- topic: Ein optionaler Themenbezeichner zur Kategorisierung

Diese Struktur bildet die Grundlage für die Datenverarbeitung in Node-RED. Ein beispielhafter Flow ist in Abbildung 2.2 abgebildet.



**Abbildung 2.2:** Ein einfacher Node-RED-Flow: Inject → Function → Debug

Ein *Inject*-Node erzeugt eine Nachricht mit einem definierten *Payload*. Ein *Function*-Node verarbeitet die Nachricht, beispielsweise durch Anwendung einer JavaScript-Funktion. Der *Debug*-Node gibt das Ergebnis im Debug-Fenster des Editors aus. Sol-

che Flows ermöglichen eine schnelle und intuitive Entwicklung von Anwendungen, ohne tiefgreifende Programmierkenntnisse.

Darüber hinaus bietet Node-RED die Möglichkeit, grafische Benutzeroberflächen mithilfe des sogenannten *Node-RED Dashboards* zu erstellen. Diese Oberflächen können in einem separaten Browser-Fenster aufgerufen werden und eignen sich insbesondere zur Echtzeitvisualisierung von Prozessdaten. In Kapitel 4 wird diese Funktionalität exemplarisch genutzt, um eine interaktive Darstellung des Bandumlaufsystems zu realisieren und Benutzereingaben zur Steuerung zu ermöglichen.

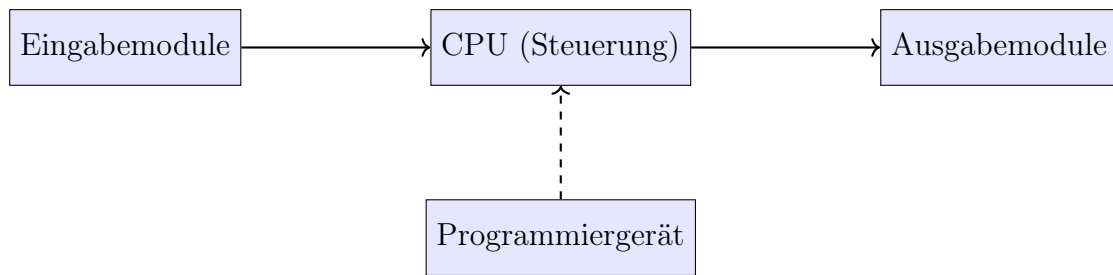
## 2.4 Speicherprogrammierbare Steuerungen (SPS)

Speicherprogrammierbare Steuerungen (SPS) sind spezialisierte, industrielle Recheneinheiten zur Automatisierung technischer Prozesse. Sie ersetzen konventionelle, fest verdrahtete Steuerungssysteme und bieten eine flexible, softwarebasierte Möglichkeit zur Realisierung komplexer Ablaufsteuerungen. SPS-Systeme sind in nahezu allen Bereichen der Industrieautomation zu finden – von einfachen Förderbändern bis hin zu hochkomplexen Produktionslinien [Sie21].

### 2.4.1 Aufbau einer SPS

Eine SPS besteht aus mehreren funktionalen Einheiten: einer zentralen Recheneinheit (CPU), digitalen oder analogen Ein- und Ausgabemodulen (I/O), einer Spannungsversorgung sowie optionalen Kommunikationsschnittstellen und Programmierschnittstellen. Das folgende Diagramm zeigt den grundsätzlichen Aufbau:

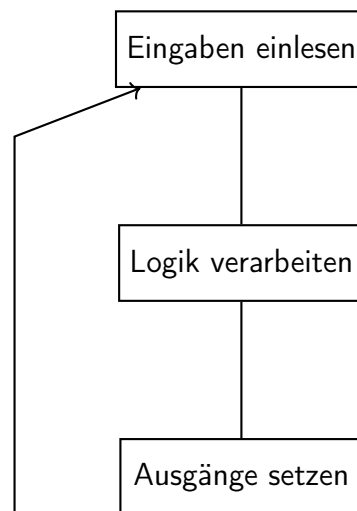




**Abbildung 2.3:** Schematischer Aufbau einer SPS

### 2.4.2 Funktionsweise: Der SPS-Zyklus

Die Arbeitsweise einer SPS basiert auf dem EVA-Prinzip (Eingabe – Verarbeitung – Ausgabe). Dabei werden zyklisch alle Eingänge gelesen, der Steuerungsalgorithmus (z.B. in Form eines SPS-Programms) verarbeitet und anschließend die entsprechenden Ausgänge gesetzt. Dieser Ablauf, auch *SPS-Zyklus* genannt, wiederholt sich kontinuierlich.



**Abbildung 2.4:** Zyklus einer SPS-Steuerung

### 2.4.3 Programmiersprachen nach IEC 61131-3

Die Norm IEC 61131-3 definiert fünf standardisierte Programmiersprachen für SPS:

- **KOP (Kontaktplan)** – grafische Darstellung mit Relaislogik
- **FUP (Funktionsplan)** – logische Gatterstruktur (ähnlich wie digitale Schaltungen)
- **AWL (Anweisungsliste)** – Assembler-ähnlich
- **ST (Structured Text)** – textbasierte Hochsprache (ähnlich Pascal)
- **SFC (Sequential Function Chart)** – Ablaufsteuerungen mit Schritten und Transitionen

Ein Beispiel in *Structured Text (ST)* zeigt eine einfache UND-Verknüpfung zweier Eingänge:

**Listing 2.1:** Beispielprogramm in Structured Text

```
1  IF Eingang1 = TRUE AND Eingang2 = FALSE THEN
2    Ausgang := TRUE;
3  ELSE
4    Ausgang := FALSE;
5  END_IF;
```

### 2.4.4 Einsatz im Kontext dieser Arbeit

In dieser Arbeit wird eine SPS zur Steuerung einzelner Aktoren innerhalb eines Bandumlaufsystems verwendet. Die Steuerung wird über digitale Ein- und Ausgänge realisiert, während Status- und Diagnosedaten über ein MQTT-Gateway von einem zentralen Raspberry Pi übertragen werden sollen. Die genaue Anbindung sowie die Realisie-

rung der Schnittstelle zwischen Raspberry Pi und SPS werden in Kapitel 4 detailliert beschrieben.

# 3 Konzeptentwurf

Dieses Kapitel beschreibt die methodische Herleitung eines technischen Konzepts zur Umsetzung des in Kapitel 2 vorgestellten Systems. Basierend auf den identifizierten technologischen Grundlagen und Rahmenbedingungen erfolgt zunächst eine strukturierte Analyse der funktionalen und nicht-funktionalen Anforderungen mithilfe etablierter Methoden des Requirements Engineerings.

Darauf aufbauend werden verschiedene Lösungsoptionen erarbeitet, hinsichtlich ihrer technischen und wirtschaftlichen Eignung bewertet und schließlich zu einem konsistenten Gesamtkonzept zusammengeführt. Eine zentrale Rolle spielt dabei der Vergleich möglicher Hardwarekomponenten unter Berücksichtigung definierter Bewertungskriterien, welcher durch eine Nutzwertanalyse unterstützt wird.

Ein wesentlicher Bestandteil des Konzepts ist der Systementwurf. In diesem wird die Integration der verwendeten Technologien – wie z. B. NFC-Komponenten, die Anbindung der SPS über MQTT sowie die Einbettung in das Gesamtsystem – detailliert beschrieben. Dabei werden auch Aspekte wie die Strukturierung von Speicherbereichen auf NFC-Tags, die Auswahl geeigneter Softwarebibliotheken und die Kommunikationsarchitektur berücksichtigt.

Ziel ist es, eine nachvollziehbare und belastbare Begründung für die gewählte Vorgehensweise zu liefern, die die Anforderungen der Anwendung erfüllt und eine verlässliche Grundlage für die anschließende Umsetzung bietet.

### 3.1 Anforderungsanalyse mittels Requirements Engineering

### 3.2 Hardwarebewertung mithilfe einer Nutzwertanalyse

### 3.3 Systementwurf

## 4 Ergebnisse

### 4.1 Umsetzung des Softwaredesigns auf den Mikrocontrollern

### 4.2 Visualisierung der Prozessdaten in Node-Red

Je nach Art der Arbeit kann diese Kapitelüberschrift auch „Ergebnisse“ lauten, z. B. bei rein messtechnischen Aufgaben.

Beschreibung der Umsetzung des zuvor gewählten Vorgehens (theoretische Untersuchung, Erhebungen, Durchführung von Experimenten, Prototypenaufbau, Implementierung eines Prozesses, etc.).

Verifikation anhand der zuvor erarbeiteten Anforderungen und Validierung in Bezug auf das zuvor gestellte Ziel. Diskussion der Ergebnisse. Spätestens hier auch auf die Zuverlässigkeit der gewonnenen Erkenntnisse eingehen (z. B. anhand der Genauigkeit von Messergebnissen).

# 5 Zusammenfassung

Auf zwei bis drei Seiten soll auf folgende Punkte eingegangen werden:

- Welches Ziel sollte erreicht werden
- Welches Vorgehen wurde gewählt
- Was wurde erreicht, zentrale Ergebnisse nennen, am besten quantitative Angaben machen
- Konnten die Ergebnisse nach kritischer Bewertung zum Erreichen des Ziels oder zur Problemlösung beitragen
- Ausblick

In der Zusammenfassung sind unbedingt klare Aussagen zum Ergebnis der Arbeit zu nennen. Üblicherweise können Ergebnisse nicht nur qualitativ, sondern auch quantitativ benannt werden, z. B. „...konnte eine Effizienzsteigerung von 12 % erreicht werden.“ oder „...konnte die Prüfdauer um 2 h verkürzt werden“.

Die Ergebnisse in der Zusammenfassung sollten selbstverständlich einen Bezug zu den in der Einleitung aufgeführten Fragestellungen und Zielen haben.

# Literaturverzeichnis

- [Fin23] Klaus Finkenzeller. *RFID Handbuch: Grundlagen und praktische Anwendungen kontaktloser Identifikation*. 7. Aufl. Zugriff am 18.05.2025. Hanser Verlag, 2023. ISBN: 978-3-446-43943-6. URL: [http://rfid-handbook.de/downloads/G6E\\_20120413\\_212413155-24\\_978-3-446-42992-5\\_Leseprobe.pdf](http://rfid-handbook.de/downloads/G6E_20120413_212413155-24_978-3-446-42992-5_Leseprobe.pdf).
- [Nod24] Node-RED Community. *Node-RED – Flow-based programming for the Internet of Things*. Zugriff am 18.05.2025. 2024. URL: <https://nodered.org>.
- [OAS19] OASIS. *MQTT Version 5.0*. <https://docs.oasis-open.org/mqtt/mqtt/v5.0/os/mqtt-v5.0-os.html>. Zugegriffen am 14.05.2025. 2019.
- [Sie21] Siemens AG. *SCE Lernpfad: Übersicht der verfügbaren SCE-Lern-/Lehrunterlagen*. Online verfügbar unter <https://www.automation.siemens.com/sce-static/learning-training-documents/classic/guide/00-guide-de.pdf>, zuletzt abgerufen am 18.05.2025. Siemens AG, Digital Industries. Nürnberg, Deutschland, Juni 2021.



# Verzeichnis verwendeter Formelzeichen und Abkürzungen

$a$ .....	Beschleunigung
ABS .....	Antiblockiersystem
EMF .....	Enhanced Metafile
ESC .....	Electronic Stability Control, Fahrdynamikregelung
etc. ....	et cetera
$F$ .....	Kraft
f. ....	folgende Seite
ff. ....	fortfolgende Seiten
JPG .....	Joint Photographic Experts Group
KI .....	Künstliche Intelligenz
$m$ .....	Masse
$P$ .....	Leistung
PDF .....	Portable Document Format
PNG .....	Portable Network Graphics
$R$ .....	Widerstand
$U$ .....	Spannung
vgl. ....	vergleiche
z. B. ....	zum Beispiel

# Abbildungsverzeichnis

2.1	Ablauf des MQTT Publish-/Subscribe-Verfahrens mit Verbindungsaufbau . . . . .	7
2.2	Ein einfacher Node-RED-Flow: Inject → Function → Debug . . . . .	8
2.3	Schematischer Aufbau einer SPS . . . . .	10
2.4	Zyklus einer SPS-Steuerung . . . . .	10

# Tabellenverzeichnis

2.1	Vergleich zwischen RFID und NFC . . . . .	4
A.1	Liste der verwendeten Künstliche Intelligenz basierten Werkzeuge . . .	21

# A Nutzung von Künstliche Intelligenz basierten Werkzeugen

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

**Tabelle A.1:** Liste der verwendeten KI basierten Werkzeuge

<b>Werkzeug</b>	<b>Beschreibung der Nutzung</b>
ChatGPT	<ul style="list-style-type: none"><li>• Grundlagenrecherche zu bekannten Prinzipien bei der Nachrichtenübertragung und Kommunikationsprotokollen (Abschnitt 2.2)</li><li>• Generierung von Beispielcode bei der Softwareimplementierung auf den Mikrocontrollern (Abschnitt 3.3)</li><li>• Grammatikalische Überarbeitung von Textpassagen</li><li>• Rechtschreibprüfung</li></ul>

## B Ergänzungen

B.1 Details zu bestimmten theoretischen Grundlagen

B.2 Weitere Details, welche im Hauptteil den Lesefluss behindern

B.3 Ausführung von Quellcode

## C Details zu Laboraufbauten und Messergebnissen

C.1 Versuchsanordnung

C.2 Liste der verwendeten Messgeräte

C.3 Übersicht der Messergebnisse

C.4 Schaltplan und Bild der Prototypenplatine

## D Zusatzinformationen zu verwendeter Software

D.1 Struktogramm des Programmentwurfs

D.2 Wichtige Teile des Quellcodes