



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Mike Perkovic

Entwicklung einer Demoanwendung mit Visualisierungen auf Basis von ausgewählten und bewerteten Industrial-IoT-Systemen

Mike Perkovic

**Entwicklung einer Demoanwendung
mit Visualisierungen auf Basis von
ausgewählten und bewerteten
Industrial-IoT-Systemen**

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang Mechatronik
am Department Fahrzeugtechnik und Flugzeugbau
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

in Zusammenarbeit mit:
Stadler + Schaaf Mess- und Regeltechnik GmbH
Niederlassung Hamburg
Großmoorbogen 9
D-21079 Hamburg

Erstprüfer: Prof. Dr.-Ing. Hans-Joachim Beyer
Zweitprüfer: Dipl.-Ing. (FH) Henry Fischer

Abgabedatum: 31.01.2020

Zusammenfassung

Mike Perkovic

Thema der Bachelorthesis

Entwicklung einer Demoanwendung mit Visualisierungen auf Basis von ausgewählten und bewerteten Industrial-IoT-Systemen

Stichworte

Industrial Internet of Things, IoT, Visualisierung, Industrie 4.0, System, Cloud, Server, Steuerung, Anbindung, MQTT, Protokolle, Testen, Daten, Bewertung

Kurzzusammenfassung

Diese Bachelorarbeit umfasst die Gegenüberstellung und Bewertung von industriellen IoT-Systemen verschiedener Anbieter auf dem Markt. Es werden IoT-Systeme ausgewählt, analysiert und getestet. Zu jedem IoT-System sollen Datenquellen angebunden und Daten auf eine Cloud übertragen werden. Auf der Cloud sollen die übertragenden Daten in einer Visualisierung dargestellt werden. Zum Abschluss werden die optimalen Lösungen für den Einsatz in der Lebensmittelindustrie erläutert.

Mike Perkovic

Title of the paper

Development of a demo application with visualizations based on selected and evaluated industrial IoT systems

Keywords

Industrial internet of things, IoT, Visualization, Industry 4.0, System, Cloud, Server, Control, Connection, MQTT, Protocols, Testing, Data, Valuation

Abstract

This bachelor thesis involves the comparison and evaluation of industrial IoT systems from different providers on the market. IoT systems are selected, analyzed and tested. For every IoT system, data sources should be connected and data transferred to a cloud. The transferred data should be displayed in a visualization on the cloud. Finally, the optimal solutions for use in the food industry are explained.

Danksagung

An dieser Stelle möchte ich mich bei all denjenigen bedanken, die mich während der Anfertigung dieser Bachelorarbeit motiviert und unterstützt haben.

In erster Linie gebührt mein Dank Herrn Prof. Dr.-Ing. Hans-Joachim Beyer, der meine Bachelorarbeit betreut und begutachtet hat. Für die hilfreichen Anregungen und die konstruktive Kritik bei der Erstellung dieser Arbeit möchte ich mich herzlich bedanken.

Ein besonderer Dank gilt meinem Zweitprüfer Herrn Dipl.-Ing. (FH) Henry Fischer von der Firma *Stadler + Schaaf Mess- und Regeltechnik GmbH*, der mir Informationen zur Verfügung gestellt und meine Bachelorarbeit betreut hat.

Ich danke meinen Arbeitskollegen in Hamburg und Bremen, insbesondere Herrn Adrian Lühr und Herrn Jörg Bussen, die mir während meiner Forschung zur Seite gestanden und eine angenehme Zusammenarbeit mit der Firma ermöglicht haben.

Meiner Familie und meiner Freundin Joan Calma danke ich besonders für den starken emotionalen Rückhalt über die Dauer meines gesamten Studiums.

Außerdem möchte ich Joan Calma, Vivien Stoffers, Zeolani Tuazon, Christiane Seemann und Sascha Seemann für das Korrekturlesen meiner Bachelorarbeit danken.

Bachelorthesis

Herr Mike Perkovic



Entwicklung einer Demoanwendung mit Visualisierungen auf Basis von ausgewählten und bewerteten Industrial-IoT-Systemen

Die Firma „Stadler + Schaaf Mess- und Regeltechnik GmbH“ liefert elektro-, mess- und regeltechnische Ausrüstungen sowie Automatisierungssysteme für den sicheren, zuverlässigen und wirtschaftlichen Betrieb der weltweiten Kunden in der Prozessindustrie. Aktuell hat ein Kunde der Lebensmittelindustrie Bedarf an einer Lösung, Fertigungsmaschinen „smarter“ zu gestalten. Dies soll mit Hilfe einer „IIoT-Box“ realisiert werden. Hierdurch sollen neue Technologien der Industrie 4.0 wie Cloud-Anbindungen, OPC UA, MQTT oder HTML5-Visualisierung angewandt werden.

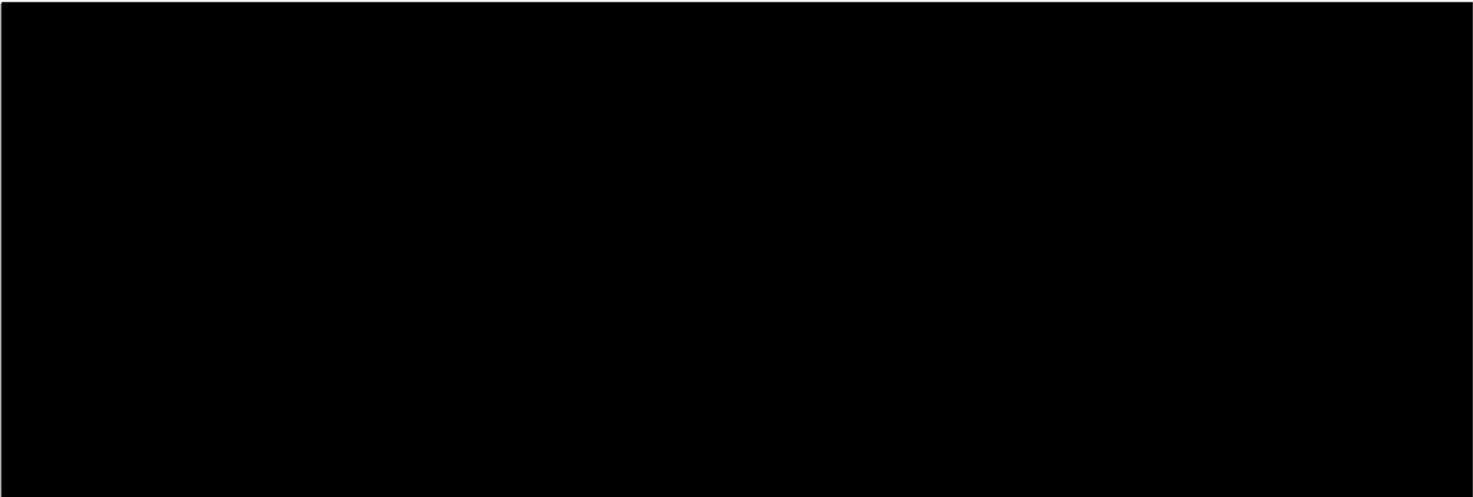
In der Bachelorthesis soll Herr Perkovic verfügbare IIoT-Systeme auf dem Markt suchen, bewerten, testen, eine Demoanwendung entwickeln und diese an Hardware anbinden. Ziel ist, dass der Kunde sehen soll, wie die Praxislösungen aussehen können, um hierdurch Einsatzmöglichkeiten und deren Potenzial praxisnah darzustellen. Dies dient der Kundengewinnung im Bereich Betriebsdatenerfassung. Zudem sollen Vor- und Nachteile der Systeme für spätere Anwendungsmöglichkeiten erforscht werden.

Aufgabenstellung

Die Bachelorthesis umfasst folgende Teilaufgaben:

1. Ermittlung des Standes der Technik.
2. Definition von Bewertungskriterien für IIoT-Anwendungen.
3. Bewertung verschiedener IIoT-Lösungen nach den definierten Bewertungskriterien.
4. Entwicklung einer Demoanwendung mit Visualisierungen eines ausgewählten Systems.
5. Dokumentation als Handlungsempfehlung für zukünftige Projekte.

Falls die verbleibende Zeit es zulässt, sollen weitere Systeme getestet werden.



Inhaltsverzeichnis

Inhaltsverzeichnis	VI
Abkürzungsverzeichnis und Glossar	VIII
1 Einführung	1
1.1 Hintergrund und Motivation.....	1
1.2 Zielsetzung und allgemeine Vorgehensweise	2
2 Vorstellung des Unternehmens	3
3 Marktübersicht und Stand der Technik.....	4
3.1 Grundlagen	5
3.2 Anwendungsfall	7
3.3 Anforderungen	8
4 Festlegen der Bewertungskriterien	9
4.1 Funktionalität	9
4.2 Hardwareseitige Konnektivität	10
4.3 Softwareumfang.....	10
4.4 Kosten.....	11
5 Test der IoT-Systeme	12
5.1 <i>GreenBox ECO</i> von <i>Schneider Electric SE</i>	12
5.1.1 Aufbau und Funktionsweise der Hardware	12
5.1.2 Vorkonfiguration	14
5.1.3 Datenerzeugung.....	15
5.1.3.1 Flow 1 - Auslesen der SPS-Daten.....	15
5.1.3.2 Flow 2 - Generieren von Zufallszahlen	16
5.1.3.3 Flow 3 - Manuelle Steuerung.....	18
5.1.3.4 Flow 4 - Manipulation von Amplitude und Offset	19
5.1.4 Cloud-Anbindung	20
5.1.4.1 Flow 1 - Bekanntgabe des Tokens	20
5.1.4.2 Flow 2 - Übertragung der generierten Daten	21
5.1.4.3 Flow 3 - Übertragung der <i>GreenBox</i> -Daten.....	23
5.1.5 Visualisierung.....	24
5.1.5.1 Simulierte Signale.....	24
5.1.5.2 OEE-Überwachung.....	26
5.2 <i>IoT-Box</i> von <i>WAGO Kontakttechnik GmbH & Co. KG</i>	27
5.2.1 Aufbau und Funktionsweise der Hardware	27
5.2.2 Vorkonfiguration	29
5.2.3 Festlegen der Übertragungsdaten	30
5.2.3.1 Variante 1	30
5.2.3.2 Variante 2	31
5.2.4 Cloud-Anbindung	33
5.2.5 Visualisierung.....	34
5.3 <i>MindSphere</i> von <i>Siemens AG</i>	36
5.3.1 Aufbau und Funktionsweise des Systems	36
5.3.2 Programmierung	38

5.3.3	Fazit	39
6	Bewertung der IoT-Systeme und Auswahl	40
6.1	Bewertungsmethode	40
6.2	Bewertung	41
6.3	Auswahl	43
7	Zusammenfassung und Schlussfolgerung	44
8	Ausblick	45
	Literaturverzeichnis	46

Abkürzungsverzeichnis und Glossar

Add-on	Optionale Softwarekomponente, die eine bestehende Software erweitert
Amplitude	Längster Ausschlag einer Schwingung von der Mittellinie bis zum Maximum
Android	Betriebssystem für mobile Geräte des Unternehmens <i>Google LLC</i>
Array	Z. Dt. Feld – Bezeichnet in der Informatik eine Datenstruktur, deren Inhalte mit Hilfe von Indizes zugegriffen wird
Browser	Software zur Darstellung von Webseiten
Cloud	Zusammenschluss von nicht-lokalen Rechnern über ein Netzwerk
Client	Ein Endgerät oder Computerprogramm
CPU	C entral P rocessing U nit (z. Dt. Zentrale Prozessoreinheit) – Ein Prozessor
Dashboard	Eine grafische Benutzeroberfläche
Debugging	Diagnostizieren und Auffinden von Computerfehlern
DHCP	D ynamic H ost C onfiguration P rotocol – Kommunikationsprotokoll
Flow	Zusammensetzung aus mehreren Funktionsbausteinen
HTML5	H ypertext M arkup L anguage 5 – Informatiksprache zur Darstellung von Webinhalten wie Texte und Bilder
HTTP	H ypertext T ransfer P rotocol – Protokoll für den Austausch von Webdaten
HTTPS	H ypertext T ransfer P rotocol S ecure – Verschlüsseltes HTTP-Protokoll
IIoT	Industrial IoT – Industrielle Ausprägung des IoT
iOS	Betriebssystem für mobile Geräte des Unternehmens <i>Apple Inc.</i>
IoT	Internet of Things (z. Dt. Internet der Dinge)
IP-Adresse	Internet p rotokoll-Adresse – Identifiziert ein Gerät innerhalb eines Netzwerkes
JSON	J ava S cript O bject N otation – Datenformat für den Datenaustausch

MAC-Adresse	Media-Access-Control-Adresse – Hardware-Adresse eines Netzwerkadapters
Modbus	Nachrichtenprotokoll zur Datenübertragung
MQTT	Message Queuing Telemetry Transport – Nachrichtenprotokoll für den Austausch von Maschinendaten
OEE	Overall Equipment Effectiveness (z. Dt. Gesamtanlageneffektivität) – Eine Kennzahl zur Festlegung der Anlagenproduktivität
Offset	Kennzahl zur Behebung von Fehlern
Onboarding	Bezeichnet in der Informatik die Vorkonfiguration von Einstellungen
OPC UA	Open Platform Communications Unified Architecture – Standard für den Datenaustausch von Automatisierungssystemen
Payload	Bezeichnet die Nutzdaten eines Datenpakets
Port	Bereich einer IP-Adresse für Zuweisungen von Verbindungen
SchuKo	Schutzkontakt
Server	Ein Zentralrechner
SPS	Speicherprogrammierbare Steuerung – Programmierbares Gerät zur Steuerung und Regelung einer Anlage
Subnetzmaske	Begrenzt ein bestehendes Netzwerk in ein Teilnetzwerk
Tag	Z. Dt. Etikett – Bezeichnet in der Informatik eine Auszeichnung von Daten mit Zusatzinformationen
Token	Dient in der Informatik zur Identifizierung und Authentifizierung von Systemen, Benutzern oder Diensten
URL	Uniform Resource Locator – Lokalisiert eine Webseite über HTTP/HTTPS
VPN	Virtual Private Network – Eine verschlüsselte Verbindung zwischen Client und Server über das Internet
WLAN	Wireless Local Area Network – Ein lokales Funknetz

1 Einführung

Diese Bachelorarbeit wird im Rahmen der Abschlussprüfung des Studienganges Mechatronik an der *Hochschule für Angewandte Wissenschaften Hamburg* in Kooperation mit dem Unternehmen *Stadler + Schaaf Mess- und Regeltechnik GmbH* in Hamburg durchgeführt.

Das Industrial Internet of Things (IIoT) bezeichnet eine globale Infrastruktur für den Informations- und Kommunikationsaustausch im Industriebereich, worüber physische und virtuelle Gegenstände miteinander verbunden werden [vgl. 1]. Das Thema dieser Bachelorarbeit befasst sich mit der Gegenüberstellung und Bewertung von industriellen IoT-Systemen unterschiedlicher Anbieter auf dem Markt. Es werden IoT-Systeme ausgewählt, analysiert und getestet. In jedem IoT-System sollen Datenquellen angebunden und Daten auf eine Cloud übertragen werden. Auf der Cloud sollen die übermittelten Daten in einer Visualisierung dargestellt werden. Zum Abschluss werden die optimalen Lösungen für den Einsatz in der Lebensmittelindustrie erläutert.

1.1 Hintergrund und Motivation

Der Start der industriellen Entwicklung, beginnend mit der industriellen Revolution, erfolgte Mitte des 18. Jahrhunderts. Menschliche Arbeitskraft konnte durch Wasser- und Dampfkraft betriebene Maschinen ersetzt werden. In der zweiten Hälfte des 19. Jahrhunderts begann die zweite industrielle Revolution. Diese ist geprägt durch den Einsatz von elektrischer Energie und Fließbänder für die Massenproduktion. Durch die digitale Revolution Ende des 20. Jahrhunderts erfolgte erstmals die Automatisierung der Produktion mit Hilfe von Elektronik und der Informatik, z. B. mittels Computer oder speicherprogrammierbaren Steuerungen (SPS) [vgl. 2]. Der aktuelle Abschnitt (Stand: 2019) wird als Industrie 4.0 bezeichnet. Der Ausdruck Industrie 4.0 wurde von der deutschen Bundesregierung im November 2011 ins Leben gerufen. Hauptaufgabe soll die Vernetzung von Menschen, Maschinen und Geräten über Sensoren jeder Art sein. Die Kommunikation erfolgt über eine globale Infrastruktur mit Hilfe von Technologien für den Informationsaustausch. Die Industrie 4.0 kann, im Gegensatz zu einer vierten Revolution, als eine nächste Stufe der digitalen Revolution betrachtet werden. Die IoT wird daher immer bedeutender und ist nicht wegzudenken.

Die Motivation dieser Arbeit liegt in der näheren Untersuchung von IoT-Systemen und deren Vorteile anhand einer Visualisierung aufzuzeigen. Mit Hilfe der Digitalisierung können Dinge miteinander verbunden und Daten an einen zentralen Server gesammelt werden. Von dort aus werden relevante Daten analysiert, ausgewertet und überwacht. Ein IoT-System kann die Wirtschaftlichkeit im Produktionsbereich durch Reduktion von Anlagenstillstandzeiten um 30-50 % steigern [vgl. 3, S. 3]. Eine Steigerung der Arbeitsproduktivität um 45-55 % und der Produktivität um 3-5 % kann ebenfalls erreicht werden [vgl. 3, S. 3]. Als Beispiel dient die Kennzahl der Gesamtanlageneffektivität (OEE), die sich aus dem Produkt der Kenngrößen Qualität, Verfügbarkeit und Leistung ergibt. OEE-Kennzahlen von Anlagen können mit Hilfe von IoT-Systemen automatisiert und zeiteffizient berechnet werden. Über mobile Clients lassen sich Verbindungen zu einer Datenbank aufbauen, um auf Daten zuzugreifen, die in einer Visualisierung angezeigt werden. Die automatisierte Erfassung der Daten spart Zeit, Kosten und optimiert die bedarfsorientierte Produktion.

1.2 Zielsetzung und allgemeine Vorgehensweise

Das Ziel dieser Arbeit liegt in der näheren Untersuchung von vorhandenen IoT-Systemen auf dem Markt und das Finden einer optimalen Lösung für die Lebensmittelindustrie. Auch im Interesse des Unternehmens soll aufgezeigt werden, wie IoT-Systeme in der Praxis funktionieren und welche Vorteile dadurch sichtbar werden. Dazu soll eine Kommunikation innerhalb eines IoT-Systems erfolgen und anhand einer Visualisierung auf einem Cloud-Server veranschaulicht werden.

Zur allgemeinen Vorgehensweise der Arbeit werden Lösungen für industrielle IoT-Systeme von Anbietern auf dem Markt ermittelt. Anschließend werden Bewertungskriterien für eine spätere Auswahl festgelegt. Die ermittelten IoT-Systeme werden analysiert, getestet und programmiert. Die Hardware soll mit einer Cloud-Plattform verbunden werden und beide sollen miteinander kommunizieren. Datenquellen werden mit der IoT-Hardware verbunden und Daten werden auf die Cloud übertragen. In einer Visualisierung sollen die Daten auf der Cloud veranschaulicht werden. Zum Abschluss werden IoT-Systeme gegenübergestellt, bewertet und die optimale Lösung erläutert.

2 Vorstellung des Unternehmens

Das Unternehmen *Stadler + Schaaf Mess- und Regeltechnik GmbH* bietet elektro-, mess- und regeltechnische Ausrüstungen sowie Automatisierungssysteme für den zuverlässigen Betrieb von weltweiten Kunden in der Prozessindustrie an. Die Firma beschäftigt aktuell (Stand 2019) 1100 Mitarbeiter an 26 Standorten in Deutschland und Österreich. Der Hauptsitz liegt in Offenbach an der Queich. [vgl. 4]

Die Bachelorarbeit wird am Standort Hamburg-Harburg durchgeführt. Dort werden Ingenieure für die Bereiche Softwareprogrammierung, Hardwareplanung, Konstruktion und Verfahrenstechnik beschäftigt. Die Arbeiten erfolgen in unterschiedlichen Projekten im Bereich Automatisierung in Niederlassungen sowie beim Kunden zur Inbetriebnahme und Installation. Die Aufgaben der Ingenieure reichen von der Überarbeitung von Stromlaufplänen bis hin zur Erstellung von Visualisierungen für Industrieanlagen.

3 Marktübersicht und Stand der Technik

Auf dem Markt befindet sich eine Vielzahl von IoT-Systemen unterschiedlicher Hersteller für den Einsatz in der Industrie. Der Elektrotechnik-Konzern *Schneider Electric SE* bietet die hardwaretechnische IoT-Produktreihe *GreenBox* mit folgenden drei Varianten an: GATEWAY, ECO und PRO. Sie unterscheiden sich in der Hardwareleistung. Für diese Arbeit wird die *GreenBox ECO* verwendet. Das deutsche Unternehmen *WAGO Kontakttechnik GmbH & Co. KG* bietet die IoT-Lösung *WAGO IoT-Box* an. Einer der bekanntesten Anbieter ist *Siemens AG*. Der Konzern betreibt eine cloudbasierte IoT-Plattform namens *MindSphere*. Maschinen und Anlagen mit unterschiedlichen Steuerungen können mit der Cloud verbunden werden.

Tabelle 1: Übersicht der ausgewählten IoT-Systeme

IoT-System	Produkt	Hersteller	Kategorie
 <p>[5]</p>	<p><i>GreenBox ECO</i></p>	<p><i>Schneider Electric SE</i></p>	<p>Hardware</p>
 <p>[6]</p>	<p><i>WAGO IoT- Box Basic 9466</i></p>	<p><i>WAGO Kontakttechnik GmbH & Co. KG</i></p>	<p>Hardware</p>
 <p>[7]</p>	<p><i>MindSphere</i></p>	<p><i>Siemens AG</i></p>	<p>Cloudbasierte IoT-Plattform</p>

Alle drei Anbieter werben für ihre IoT-Systeme mit unterschiedlichen Merkmalen. Die Firma *Stadler + Schaaf* ist ein Kunde von diesen Herstellern. Um einen näheren Eindruck über die in *Tabelle 1* genannten IoT-Systeme zu gewinnen, wurden die Firmen zu *Stadler + Schaaf* eingeladen, um ihre Lösungen zu präsentieren.

3.1 Grundlagen

Eine Infrastruktur des IoT ist als Beispiel in *Abbildung 1* dargestellt. Die IoT-Hardware ist ein eigenständiges System und sammelt Daten der Produktionsanlagen. Verbindungen erfolgen über Ein-/Ausgangsmodule an der IoT-Hardware, die mit Anlagen verknüpft sind. Die Daten werden über speicherprogrammierbare Steuerungen (SPS) von Anlagen bezogen und, falls keine Steuerungen vorhanden sind, über Sensoren. Daten werden in der IoT-Hardware im Zwischenspeicher gespeichert und von dort aus mit einer Datenbank auf der Cloud synchronisiert. In der Datenbank werden Daten gespeichert und gesammelt. Die Übertragung geschieht mit Hilfe von standardisierten Protokollen. Ein berechtigter Benutzer, beispielsweise ein Produktionsleiter, kann mit einem Gerät auf die Cloud zugreifen und die Anlagenzustände überwachen sowie Daten analysieren.

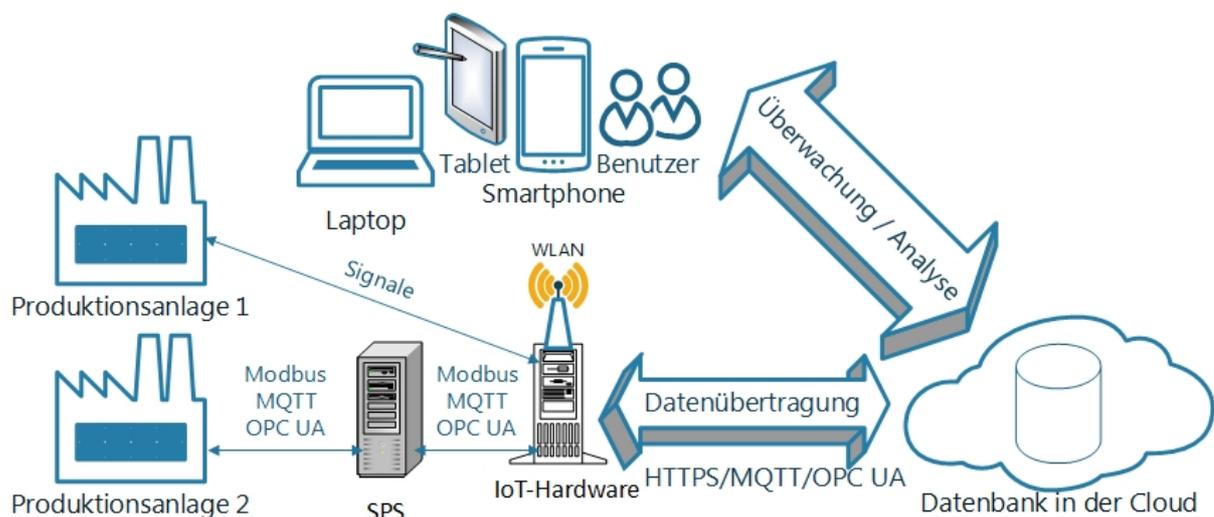


Abbildung 1: Beispiel für eine IoT-Infrastruktur

Für eine erfolgreiche Kommunikation über ein Netzwerk braucht jedes netzwerkfähige Gerät eine individuelle MAC-Adresse (Physikalische Adresse). Die Adresse ist 12-stellig und besteht aus folgender Syntax:

xx:xx:xx:xx:xx:xx

Das x stellt einen Platzhalter für eine Hexadezimalzahl von 0 bis 9 und von A bis F dar. Ein Beispiel für eine MAC-Adresse lautet wie folgt: 12:3A:B4:5C:0F:EE. Bereits während des Herstellens von Netzwerkadaptoren wird jedem eine individuelle Adresse zur Identifikation zugeteilt. Sobald ein Gerät mit einem Netzwerk verbunden ist, bekommt dieses eine IP-Adresse (Internetprotokoll-Adresse) zugewiesen. Eine IPv4-Adresse besteht aus einer vierstelligen Zahl mit jeweils einem ganzen Wert von 0 bis 255, z. B. 192.168.0.12.

Optional kann ein Gerät durch Zuweisung eines Gerätenamens identifiziert werden, der jederzeit änderbar ist. Dieser besteht aus Klein-, Großbuchstaben, Zahlen und Unterstrichen.

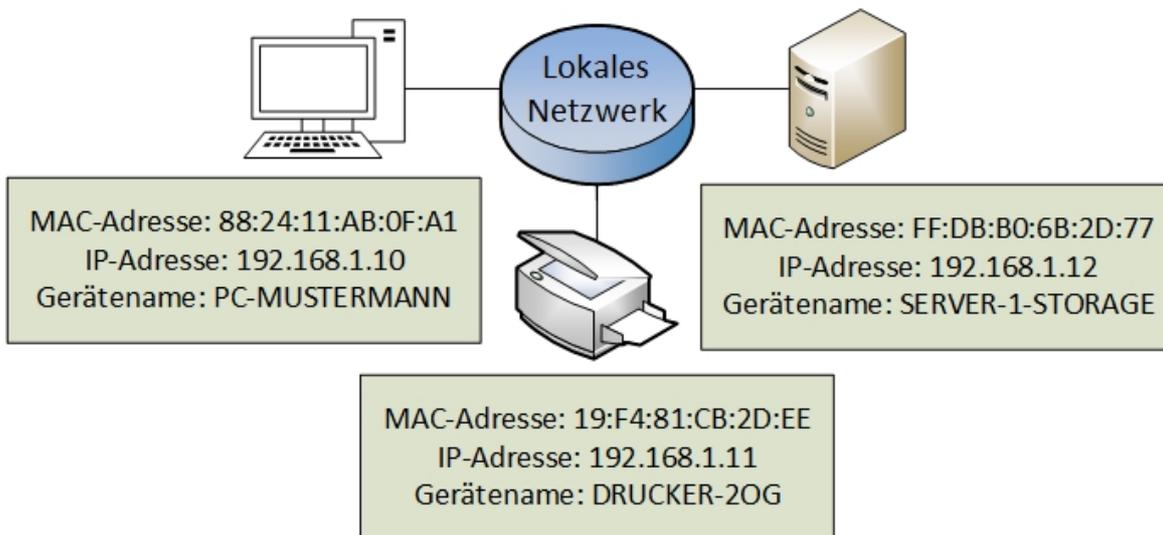


Abbildung 2: Lokales Netzwerk

Maschinendaten werden über die Übertragungsprotokolle MQTT (Message Queuing Telemetry Transport) oder Modbus sowie über die Architektur OPC UA (OPC Unified Architecture) übertragen. Diese Standards dienen zur Übertragung von Telemetriedaten. Eine mit Hilfe des MQTT-Protokolls versendete Nachricht ist hierarchisch aufgebaut, z. B.:

Fabrik/Anlage/1/Maschine/3/Band/Geschwindigkeit

Die Verwaltung der Hierarchien und die Speicherung der Daten erfolgen auf einem Server. Daten können mit Hilfe des Sicherheitsprotokolls HTTPS verschlüsselt werden, um die Sicherheit während der Übermittlung zu gewährleisten. Die Übertragung erfolgt beispielsweise mit Hilfe eines Ethernet-Kabels. Die Standards müssen für eine erfolgreiche Kommunikation in der Maschine und Steuerung implementiert sein. Signale müssen über digitale oder analoge Ein-/Ausgangskarten am IoT-System mit Kabel verbunden sein, falls keine Steuerungen vorhanden sind.

3.2 Anwendungsfall

In *Abbildung 3* ist als Beispiel eine Panademaschine für die Lebensmittelindustrie dargestellt. Mit Hilfe eines motorisierten Förderbandes wird tiefgekühlter Fisch transportiert. Über eine Rohrleitung, die mit einem Elektroventil blockiert ist, wird Panade auf den Fischen aufgetragen. Das Band wird gestoppt, sobald der Näherungssensor Fisch erkennt. Anschließend wird der Durchfluss der Panade über das Ventil freigegeben. Sobald eine festgelegte Menge auf dem Fisch aufgespritzt ist, wird das Ventil blockiert und das Band läuft weiter. Die Anlage wird über eine lokale SPS verwaltet. Das IoT-System ist mit der Steuerung der Anlage gekoppelt und erfasst die Anlagenzustände über das MQTT-Protokoll. Über ein Netzwerkmodul synchronisiert das System die Daten mit einer zentralen Datenbank auf der Cloud.

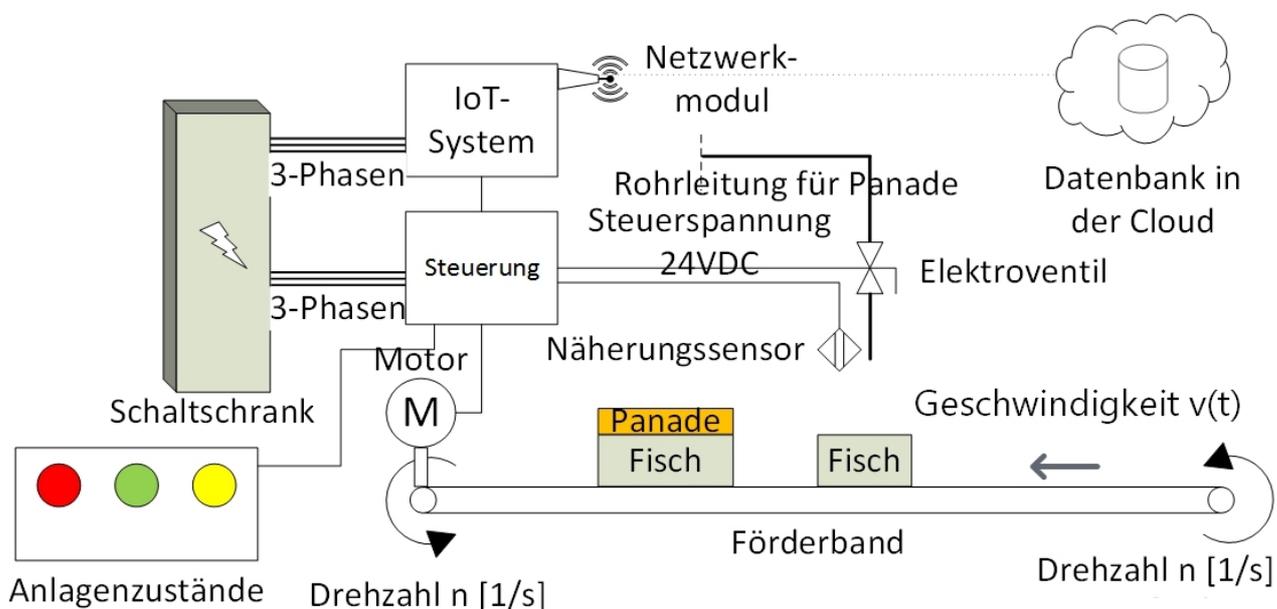


Abbildung 3: Beispiel einer Panadeanlage

3.3 Anforderungen

Für die Auswahl eines IoT-Systems werden Anforderungen in *Tabelle 2* festgelegt. Die Software soll benutzerfreundlich aufgebaut sein. Datenquellen sollen über Ein- und Ausgangsmodule mit dem IoT-System verbunden werden. Eine Erweiterbarkeit der Software und der Hardware soll umfangreich realisierbar sein. Fehlende Funktionen sollen dadurch nachimplementiert und Hardwarekomponenten aufgerüstet werden können. Die Gestaltung des Dashboards auf der Cloud soll übersichtlich aufgebaut sein sowie umfangreiche Visualisierungsoptionen enthalten. Die IoT-Hardware muss aufgrund von Hygienevorschriften [8] der Lebensmittelindustrie in ein von Flüssigkeit und Staub geschütztes Gehäuse platziert sein.

Tabelle 2: Anforderungsliste

ID	<u>W</u> unsch / <u>F</u> orderung	Anforderung	Datum	Ersteller
1	F	Anwenderfreundlichkeit	01.11.2019	M. Perkovic
2	F	Erweiterbarkeit der Hardware	01.11.2019	M. Perkovic
3	F	Erweiterbarkeit der Software	01.11.2019	M. Perkovic
4	F	Dashboard	01.11.2019	M. Perkovic
5	F	Umfangreiche Visualisierungsoptionen	01.11.2019	M. Perkovic
6	F	Wasser- und Staubgeschütztes Aluminiumgehäuse	01.11.2019	M. Perkovic
7	W	Kosten	01.11.2019	M. Perkovic

4 Festlegen der Bewertungskriterien

In diesem Kapitel werden Bewertungskriterien festgelegt, die für eine Bewertung und Auswahl von IoT-Systemen relevant sind. Zusätzlich werden Erfahrungen, die während des Testens in *Kapitel 5* entstehen, für die optimale Auswahl in *Kapitel 6* mitberücksichtigt.

Das Thema Sicherheit und Datenschutz wird nicht mitberücksichtigt, da das Gebiet breit gefächert ist und eine Vertiefung würde über den Rahmen dieser Arbeit hinausgehen. Daten, die auf gemieteten Servern eines Drittanbieters gespeichert werden, könnten jederzeit von Unbefugten eingesehen werden. Ein Speicherort für Daten auf einem lokalen Server, der im eigenen Besitz ist, erhöht die Sicherheit. Der Schutz vor einem Fremdzugriff steigt, wenn der lokale Server vom öffentlichen Internet strikt getrennt ist oder ein Zugriff von außen über eine VPN-Verbindung erfolgt.

4.1 Funktionalität

Als Funktionalität werden softwareseitige Funktionen bezeichnet, die der Hersteller bereits in ein IoT-Betriebssystem oder auf der Cloud-Plattform implementiert hat.

Tabelle 3: Kriterien zur Funktionalität

Kriterium	Beschreibung
HTML5	Texte und elektronische Inhalte können in einem Webbrowser optimal angezeigt werden.
Dashboard	Eine konfigurierbare und grafische Benutzeroberfläche, die nach einem erfolgreichen Login angezeigt wird.
Datennormalisierung	Daten werden strukturiert und übersichtlich gespeichert, z. B. im Tabellenformat.
Visualisierung	Daten können als Visualisierung in unterschiedlichen Diagrammen und Tabellen angezeigt werden.
Analytik	Daten können mit Hilfe von mathematischen Operationen und Tools untersucht und ausgewertet werden.
Benutzermanagement	Ein Administrator kann Berechtigungen und Rechte erteilen.

4.2 Hardwareseitige Konnektivität

Mit der Konnektivität ist die Anbindungsvielfalt hardwaretechnischer Komponenten an das vorhandene IoT-System gemeint. Ein WLAN-Modul soll beispielsweise an eine Ethernet-Schnittstelle angebunden werden. Auch digitale und analoge Ein- und Ausgänge sollen erweitert werden. Komponenten müssen dabei mit dem System kompatibel sein.

4.3 Softwareumfang

Im Folgenden werden die bedeutsamsten Eigenschaften genannt, die ein IoT-System in Bezug auf die Software der Cloud sowie der lokalen SPS beinhalten sollte.

Tabelle 4: Kriterien zur Software

Kriterium	Beschreibung
Verschlüsselte Kommunikation	Der Datenaustausch soll verschlüsselt übertragen werden. Dies schützt vor Angriffen durch Abfangen des Netzwerkverkehrs, z. B. HTTPS.
Protokolle	Unterstützung von Kommunikationsprotokollen für die Datenübertragung, z. B. MQTT, OPC UA.
Erweiterbarkeit	Bestehende Software kann um fehlende Funktionen erweitert werden, z. B. mit Add-ons. Dies kann seitens des Herstellers oder durch eigene Entwicklung erfolgen.
Benutzerfreundlichkeit	Eine Software soll übersichtlich gestaltet sein. Vorlagen und Anleitungen seitens des Herstellers erleichtern die Arbeit des Anwenders.
Fernwartung	Über das Internet kann eine Fernzugriffverbindung hergestellt werden, um beispielsweise aus der Ferne manuelle Softwareupdates durchzuführen.
Herstellersupport	Im industriellen Bereich ist es bedeutsam, dass der Support gewährleistet ist. Damit sollen optimale Fehlerbehebungen erfolgen.

4.4 Kosten

Kosten können während des Beschaffens, für die Inbetriebnahme sowie für Wartungsarbeiten an der Hardware entstehen. Wird weitere Software benötigt, besteht die Alternative kostenpflichtige oder frei verfügbare Lizenzen zu erhalten. Cloud-Plattformen von Drittanbietern wie *Microsoft Azure* oder *Amazon Web Services (AWS)* verlangen Kosten pro Zeiteinheit. Dabei fallen zusätzliche Kosten für den Netzwerkverkehr an. Die Höhe der Kosten ist abhängig von der Anzahl der Datenquellen, Größe der Datenpakete und das Zeitintervall für die Synchronisation zwischen Client und Server.

Für eine Bewertung werden folgende Kosten berücksichtigt:

- Cloud-Mietung
- Datenübertragung von der Hardware zur Cloud
- Softwarelizenzen
- Hardware

Nähere Angaben zu den Preisen einzelner IoT-Systeme der genannten Hersteller werden nicht explizit genannt aufgrund von Datenschutz und der stark variierenden Preise, die abhängig von Menge, Aufwand und Performance sind.

5 Test der IoT-Systeme

In diesem Kapitel werden die in *Tabelle 1* genannten IoT-Systeme getestet. Dazu werden diese analysiert und konfiguriert. Anschließend werden Datenquellen eingebunden. Die IoT-Hardware wird mit einer Cloud verbunden und Daten zur Cloud übertragen. Visualisierungen werden auf der Cloud erstellt, worüber Daten dargestellt werden.

5.1 *GreenBox ECO* von *Schneider Electric SE*

Die *GreenBox ECO* und die dazugehörigen Komponenten werden vom Hersteller kostenlos zum Testen für diese Arbeit zur Verfügung gestellt.

5.1.1 Aufbau und Funktionsweise der Hardware

Der Testaufbau der *GreenBox ECO* in *Abbildung 4* enthält folgende Komponenten:

1. *GreenBox ECO*
 - 2 WLAN-Antennen
2. *Harmony HubWireless* (SE, ZBRN1)
3. Leistungsschutzschalter
 - Eine Variante A (SE, A9F03110 iC60N B 10 A)
 - Eine Variante B (SE, A9F03116 iC60N B 16 A)
4. Schaltnetzteil (SE, ABL8REM24030)
5. Ethernet TCP/IP Switch (SE, TCSESU053FN0)
 - 5 Ports
6. Analoges Eingangsmodul (SE, BMXAMI0410)
 - 4 Eingänge
7. Digitales Ein-/Ausgangsmodul (SE, BMXDDM16025)
 - 8 Eingänge
 - 8 Ausgänge
8. Steuerung M580 (SE, BMEH584040C)
9. Modul zur Spannungsversorgung (SE, BMXCPS4002)
 - 100-240 V AC
10. Verteilerklemmen
11. Energiesensor *PowerTag* A9 F63 (SE, A9MEM1560)
 - max. Strom 63 A

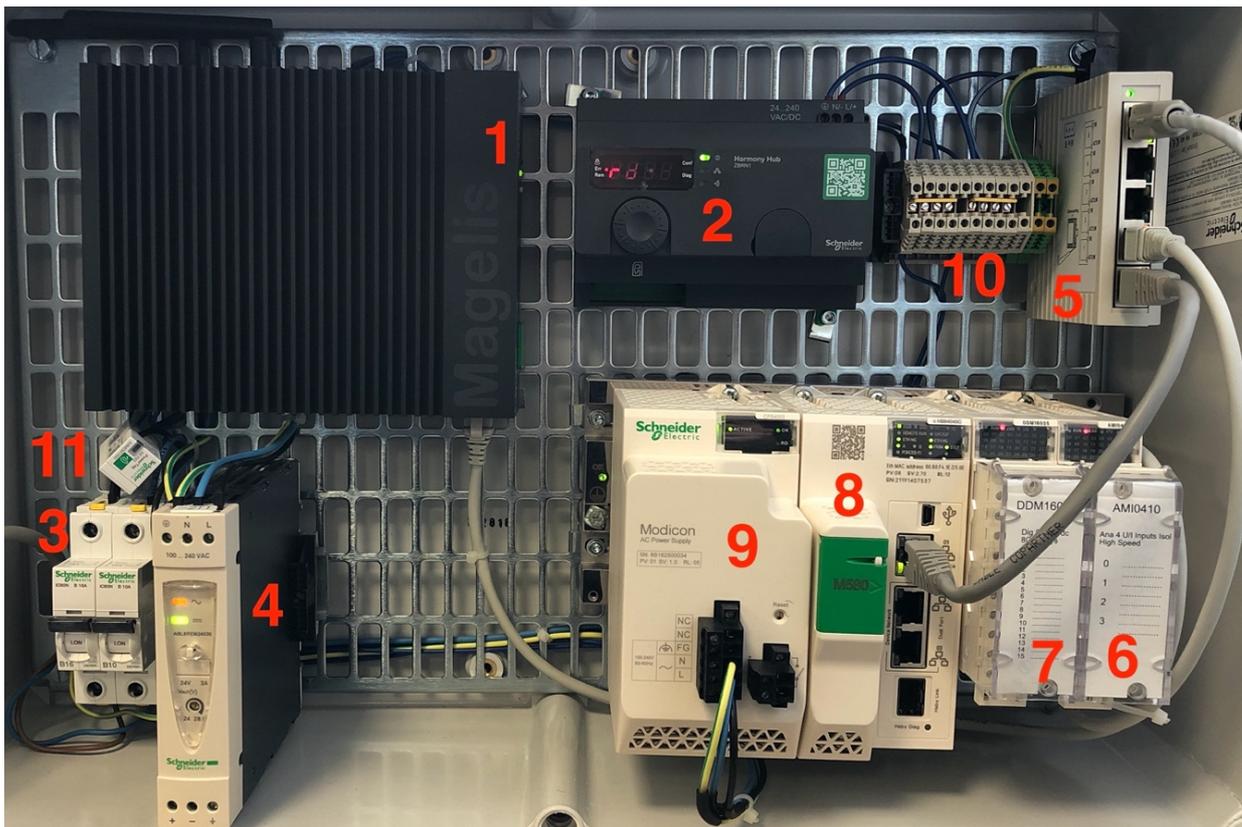


Abbildung 4: Testaufbau der *GreenBox ECO*

Die *GreenBox* (1) enthält folgende Anschlüsse:

- DC-Netzstecker 24 V
- COM1 Port RS232 (USB 2.0)
- COM2 Port RS-422/485
- 2x USB 2.0-Eingänge
- 2x USB 3.0-Eingänge
- 2x Ethernet-Eingänge (10/100/1000 Mbit/s)
- 1x Erdungsanschluss
- 2x DisplayPort-Eingang

Die *GreenBox* (1) ist über ein Ethernet-Kabel mit dem Switch (5) verbunden. Der Switch (5) dient als Verbindungsknoten, der neben der *GreenBox* (1) mit der Steuerung (8) und einem externen Computer verbunden ist. Die Steuerung (8) ist mit einem digitalen Ein-/Ausgangsmodul (7) und einem analogen Eingangsmodul (6) verbunden. Der *Harmony Hub* (2) kann mit Hilfe von IoT-Geräten des Herstellers *Schneider Electric* kabellos verbunden werden. Der Hub (2) ist aufgrund eines fehlenden Bauteils nicht mit der SPS (8) und der *GreenBox* (1) verbunden. Die Stromversorgung erfolgt von einer

Steckdose mit 230 V und drei Phasen über zwei Leistungsschutzschalter (3), einem Schaltnetzteil (4) und einem Spannungsversorgungsmodul (9). Der Leistungsschutzschalter sichert alle Module im Falle einer Überspannung ab. Das Schaltnetzteil (4) regelt die Frequenzen und das Spannungsversorgungsmodul (9) versorgt die Module. Der Energiesensor (11) wird mit dem Leistungsschutzschalter gekoppelt und dient zur Erfassung der Leistungsaufnahme. Klemmen (10) teilen die Stromadern auf und leiten diese an Module weiter, die eine Stromversorgung benötigen.

Die Kommunikation zwischen der *GreenBox* (1) und der SPS (8) erfolgt über das serielle Übertragungsprotokoll *Modbus/TCP*. Das Protokoll verwendet das Master-Slave-Prinzip. Dabei fragt die *GreenBox*, sprich der Master, durch periodische Abfragen den Status der Slaves, z. B. die SPS, über eine gemeinsame Ethernet-Leitung ab.

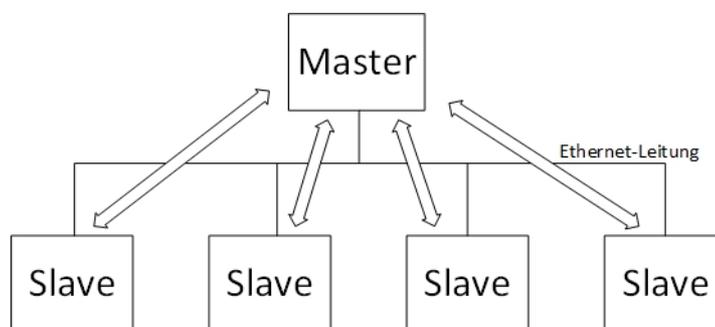


Abbildung 5: Master-Slave-Prinzip

5.1.2 Vorkonfiguration

Auf einer Festplatte der *GreenBox* mit 64 GB Speicherkapazität ist das Betriebssystem *Windows 10 IoT Enterprise LTSB-2016 (64-Bit)* vorinstalliert. Um auf das System zuzugreifen, werden eine Tastatur und eine Maus jeweils mit einem USB-Port der *GreenBox* sowie ein Monitor über DisplayPort verbunden. Nach dem Einschalten der *GreenBox* erscheint eine Anmeldemaske. Die benötigten Anmeldedaten werden vom Hersteller bereitgestellt:

- **Nutzername:** *GreenBoxDemo*
- **Passwort:** *Schneider1*

Um per Remotedesktopverbindung auf die *GreenBox* zuzugreifen, wird auf dem System die statische IP-Adresse 192.168.10.155 und die Subnetzmaske 255.255.255.0 eingestellt. Der Internetzugriff des Gerätes erfolgt über das WLAN mit Hilfe der zwei installierten Antennen und einem externen WLAN-Router.

5.1.3 Datenerzeugung

Für eine Datenübertragung von der *GreenBox* zur Cloud werden Testdaten benötigt. Anhand eines Beispiels werden drei periodische Signale simuliert. Dazu werden zufällige Werte mit unterschiedlichen Amplituden in einem festgelegten Zeitintervall generiert.

Zum Speichern der Testdaten muss Speicherplatz freigegeben werden. Das Speichern erfolgt innerhalb eines Datenregisters in der SPS. Ein Register wird erstellt und mit dem Namen *SIM_Rec* definiert. Dieser besteht aus dem Datentyp *ARRAY[0..9] OF INT*. Darin können jeweils 10 ganze Zahlen im Wertebereich von -2^{31} bis $2^{31}-1$ in den Adressen 20 bis 29 gespeichert werden. Die Startadresse 20 wird für *SIM_Rec[0]* zugewiesen.

Für eine Programmierung wird die auf der *GreenBox* vorinstallierten *Node-RED* verwendet. „Node-RED ist ein von IBM entwickeltes grafisches Entwicklungswerkzeug.“ [9] Die freie Software setzt Anwendungsfälle im Bereich des Internets der Dinge mit einem Baukastenprinzip um [vgl. 9]. Eine Dokumentation für die Programmierung befindet sich auf der offiziellen Homepage [vgl. 10]. *Node-RED* läuft als Hintergrundprozess im Betriebssystem und ist unter der lokalen URL *http://127.0.0.1:1880* erreichbar. Im Dashboard von *Node-RED* werden Flows für die Datengenerierung erstellt. Flows beinhalten grafische Funktionsbausteine und Verbindungen (*Abbildung 6*). Zusammengesetzt resultiert daraus eine Gesamtfunktion.

5.1.3.1 Flow 1 - Auslesen der SPS-Daten

Der erste Baustein in *Abbildung 6* liest die gespeicherten Nutzdaten, auch Payload genannt, des Registers *SIM_Rec* von der SPS über das Protokoll Modbus in einem festgelegten Zeitintervall ab. Dabei werden die Adressen des Registers abgefragt. Anschließend kommt eine Nachricht vom Datentyp *Struktur* an, die den Payload enthält:

msg.payload.data

Die einzelnen Werte werden im zweiten Funktionsbaustein extrahiert und in dem lokalen Array *SIM_REC[9]* gespeichert. Das Array wird abschließend in den Payload der Nachricht gespeichert und weitergeleitet. Der Baustein *limit 1 msg/5s* legt das Intervall zum Ablesen der Daten auf 5 s fest. Die Nachricht wird weitergeleitet und in *Abschnitt 5.1.4.1* um Daten erweitert.



Abbildung 6: Intervallgesteuerte Abfrage der SPS-Daten

Tabelle 5: Erklärung des erstens Flows

Baustein-gruppe & -typ	Baustein-name	Baustein-funktion	Programmierung
modbus/ SEModbus Read	SEModbus Read	Abfrage der Daten von der SPS	Address: 10 Server: modbus@192.168.10.1:Port:502
function/function	extract modbus msg values	Extrahiert Nutzdaten von der SPS in das Register <i>SIM_REC</i>	<pre>var SIM_REC = {}; SIM_REC[„SIM01“] = msg.payload.data.SIM01.value; SIM_REC[„SIM02“] = msg.payload.data.SIM01.value; SIM_REC[„SIM03“] = msg.payload.data.SIM03.value; msg.payload = SIM_REC; return msg;</pre>
function/delay	limit 1 msg/5s	Sendet alle 5 s eine Nachricht	Rate: 1 message each 5 seconds
output/link	-	Weiterleitung der Nachricht	-

5.1.3.2 Flow 2 - Generieren von Zufallszahlen

Im zweiten Flow in *Abbildung 7* werden Zufallswerte für eine Simulation gebildet und in den Adressen des Registers *SIM_Rec* gespeichert. Der Baustein *timestamp* erzeugt in einem Intervall von 2 s ein Zeitstempel, damit die Daten der Zeit zugeordnet werden können. Die beiden darauffolgenden Funktionsbausteine generieren Zufallszahlen mit Hilfe der mathematischen Funktion *Math.Floor()*. Dabei werden ganze und rationale Zahlen in einem Wertebereich gebildet. Anschließend werden die Zufallswerte als Payload in das Register *FC 6: Preset Single Register* in der SPS gespeichert. Der Baustein *msg.payload* gibt die Zufallswerte zum Testen im Debug-Modus aus.

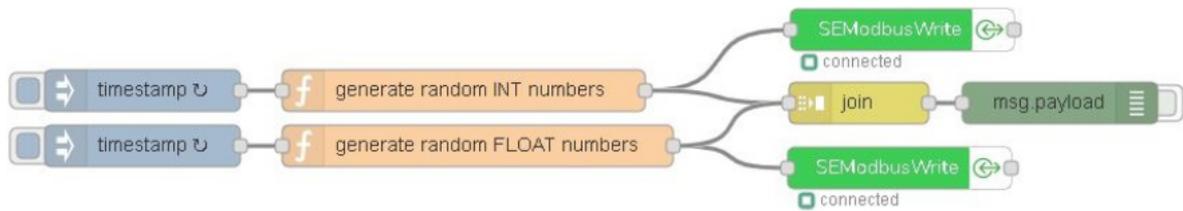


Abbildung 7: Zufallszahlengenerator

Tabelle 6: Erklärung des zweiten Flows

Baustein- gruppe & -typ	Baustein- name	Baustein- funktion	Programmierung
input/inject	timestamp	Generiert alle 2 sec einen Zeitstempel	Payload: timestamp Repeat: interval every 2 sec
function/function	generate random [INT/FLOAT] numbers	Generiert Zufallszahlen im Bereich von 30 bis 70	<code>rnd = Math.floor(Math.random() * (70-30)+30); msg.payload=rnd; return msg;</code>
modbus/ SEModbus- Write	SEModbus - Write	Schreibt den Payload in <i>SIM_Rec</i> der SPS rein	Address: 21 Address: 22
function/join	join	Fasst Werte in einer Struktur zusammen	Mode: manual Combine each msg.payload to create an array
output/debug	msg.payload	Debug- Modus	Output msg.payload

5.1.3.3 Flow 3 - Manuelle Steuerung

Im dritten Flow in *Abbildung 8* werden Simulationen ein- und ausgeschaltet, um die Signale manuell zu beeinflussen. Dabei werden Werte in das Register FC 6: Preset Single Register gespeichert, die angeben, ob eine Simulation aus- oder eingeschaltet ist.

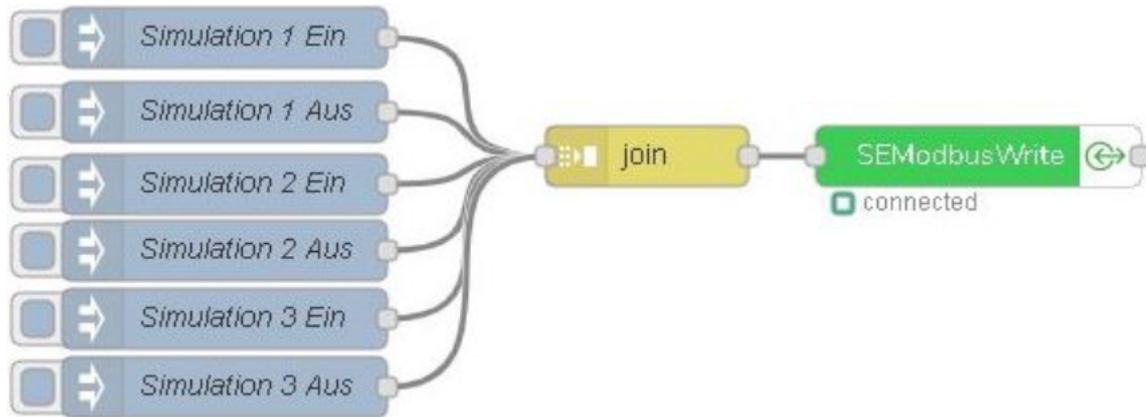


Abbildung 8: Manuelle Zustandsänderung der Simulationen

Tabelle 7: Erklärung des dritten Flows

Baustein- gruppe & -typ	Baustein- name	Baustein- funktion	Programmierung
input/inject	Gen[1/2/3]- [Ein/Aus]	Generiert Zustandszahlen	Payload: [1...6]
function/join	join	Fasst Werte in einer Struktur zusammen	Mode: manual Combine each msg.payload to create an array
modbus/SEModbus Write	SEModbus- Write	Schreibt den Payload in ein Register in der SPS	Type: FC 6: Preset Single Register Address: 2

5.1.3.4 Flow 4 - Manipulation von Amplitude und Offset

Im vierten Flow in *Abbildung 9* können Amplituden und Offsets der Signale manuell verändert werden. Es werden Werte erzeugt und in ein Register in der SPS gespeichert.

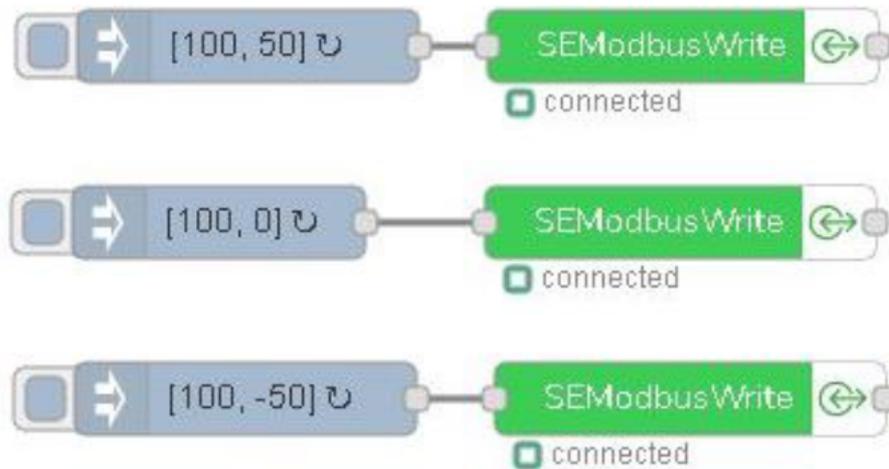


Abbildung 9: Manuelle Beeinflussung der Signale

Tabelle 8: Erklärung des vierten Flows

Baustein-gruppe & -typ	Baustein-name	Baustein-funktion	Programmierung
input/inject	[100, 50] [100, 0] [100, -50]	Generiert Zufallswerte im JSON-Format	Payload: {[100, 50]}
modbus/SEModbus Write	SEModbus Write	Schreibt Werte in ein Register in der SPS	Type: FC 16: Preset Multiple Registers Address: 24,26,28 Quantity: 2

5.1.4 Cloud-Anbindung

Als Cloud-Plattform wird *Wonderware AVEVA Insight* von der Firma *AVEVA Group plc* verwendet, einem Teilunternehmen von *Schneider Electric SE*. Die Plattform ist unter der URL <https://online.wonderware.com> erreichbar. Nach Erstellung eines Accounts sind 45 Tage Testlaufzeit verfügbar.

Um die in *Kapitel 5.1.3* erzeugten Daten an die Cloud zu senden, muss eine virtuelle Datenquelle in *AVEVA Insight* erstellt werden. Dies geschieht unter folgendem Pfad:

Insight Suite > Administration > Data Sources > Hinzufügen > CSV/JSON

Nach Anlegen wird ein individueller Sicherheits-Token generiert. Dieser muss für die *GreenBox* global bekannt sein, um eine verschlüsselte Verbindung zur Cloud herzustellen. Der Syntax des Tokens lautet wie folgt:

Bearer X

Das *X* ist ein Platzhalter für eine kryptische Zeichenkette. In diesem Fall enthält der Token 787 Zeichen, bestehend aus zufälligen Zahlen, Groß- und Kleinbuchstaben. Für eine Anbindung und Datenübertragung von der *GreenBox* zur Cloud werden drei Flows in *Node-RED* erstellt.

5.1.4.1 Flow 1 - Bekanntgabe des Tokens

Der erste Flow in *Abbildung 10* ist für die Bekanntgabe des Tokens im System zuständig. Dabei wird der Token im ersten Baustein bekanntgegeben und im zweiten Baustein in die globale Variable *Token* gespeichert.



Abbildung 10: Bekanntgabe des Login-Tokens im System

Tabelle 9: Erklärung des ersten Flows

Baustein- gruppe & -typ	Baustein- name	Baustein- funktion	Programmierung
input/inject	Token	Token wird bekannt- gegeben	Payload: Bearer eyJ0eXAiOiVB5...
function/function	Store Token to variable	Speichert den Token in eine globale Variable	<pre>var token = msg.payload; flow.set("globaltoken", token); msg.payload = token; return msg;</pre>
output/debug	msg.payload	Gibt die Nachricht für den Debug- Modus frei	Output: msg.output

5.1.4.2 Flow 2 - Übertragung der generierten Daten

Nach Bekanntgabe des Tokens im *Abschnitt 5.1.4.1* werden in einem zweiten Flow die generierten Daten der SPS an die Cloud übertragen. Dieser Flow ist eine Fortsetzung des erstellten Flows von *Abschnitt 5.1.3.1*, der die Register ausliest. Zu Beginn wird das Register *SIM_Rec* als Datenquelle eingebunden. Anschließend wird ein Objekt erstellt, worin die Nutzdaten der SPS gespeichert werden. Das Objekt wird im JSON-Format umgewandelt und um das festgelegte Sicherheits-Token erweitert, das anschließend an die Cloud über eine HTTP-Anfrage gesendet wird.

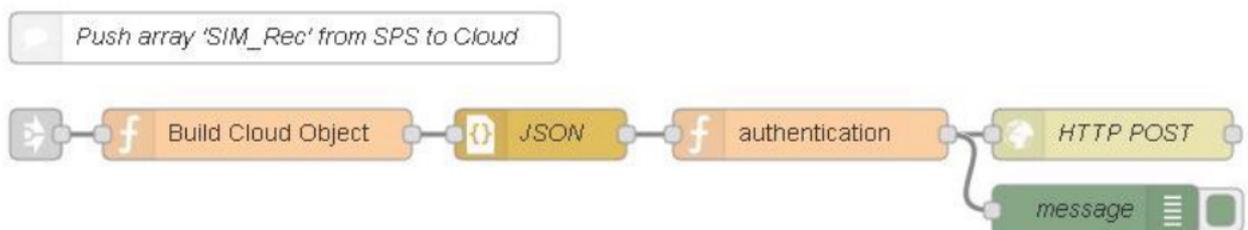


Abbildung 11: Datenübertragung an die Cloud

Tabelle 10: Erklärung des zweiten Flows

Baustein- gruppe & -typ	Baustein- name	Baustein- funktion	Programmierung
input/link	-	Empfängt SPS-Daten	siehe <i>Abbildung 5.1.3.1</i>
function/function	Build Cloud Object	Speichert SPS-Daten in ein Cloud- Objekt	<pre>var CloudData = {}; CloudData["data"] = msg.payload; msg.payload. = CloudData; return msg;</pre>
function/json	JSON	Umwandlung im JSON- Format	-
function/function	authentication	Erweiterung um ein Token- Header	<pre>var token=flow.get("gtoken"); msg.headers = { "Authorization":token, "Content-type" : "application/json" } return msg;</pre>
function/http request	HTTP POST	Sendet die Nachricht an die Cloud	URL: <i>https://online.wondershare.eu/apis/upload/datasource</i>
output/debug	message	Schließt das Objekt ab	Output: <i>complete msg object</i>

5.1.4.3 Flow 3 - Übertragung der *GreenBox*-Daten

Im dritten Flow wird die Temperatur des Prozessors (CPU) und Spannung der *GreenBox* an die Cloud gesendet. Die Daten werden in einem Intervall von 10 s abgefragt und im Baustein *hwmonitor info* angezeigt sowie zusammengefasst. Das Objekt wird im JSON-Format umgewandelt und um das Token erweitert. Per HTTP-Anfrage wird die Nachricht über eine URL an die Cloud gesendet. Dort wird die Nachricht mit Hilfe des Sicherheitstokens verifiziert.

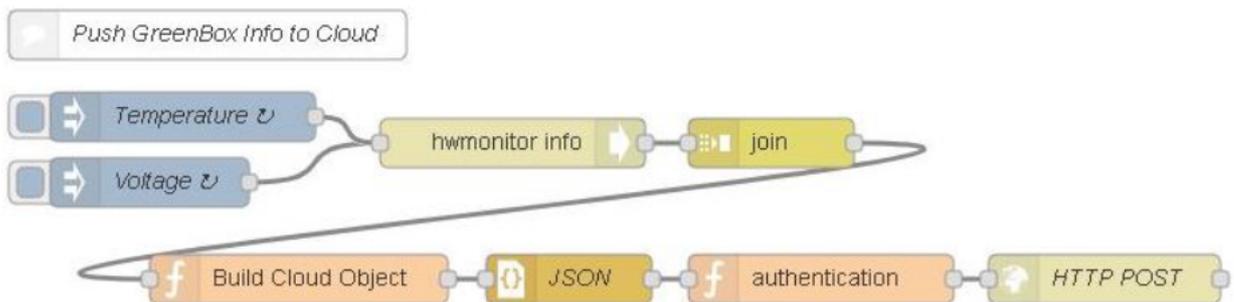


Abbildung 12: Versendet Daten der *GreenBox* an die Cloud

Tabelle 11: Erklärung des dritten Flows

Baustein- gruppe & -typ	Baustein- name	Baustein- funktion	Programmierung
input/inject	Temperature/ Voltage	Auswahl von Daten der <i>GreenBox</i>	Payload: {"Temperature": true} Payload: {"Voltage": true}
Schneider_IIoT/ hwmonitor info	hwmonitor info	Zeigt die Werte an	From: <i>msg payload</i>
function/join	join	Fasst Werte in einer Struktur zusammen	Mode: <i>manual</i> Combine each <i>msg.payload</i> to create a <i>merged Object</i>

function/function	Build Cloud Object	Extrahiert die Daten in ein Cloud-Objekt	<pre>var BoxData = {}; BoxData["CPUTemp"] = msg.payload.Temperature.CPU; BoxData["Vcore"] = msg.payload.Voltage.Vcore; BoxData["DC"] = msg.payload.Voltage.DC; var CloudAvevaData = {}; CloudAvevaData["data"] = GreenBoxData; msg.payload = CloudAvevaData; return msg;</pre>
function/json	JSON	Umwandlung im JSON-Format	-
function/function	authentication	Erweiterung um ein Token-Header	<pre>var token=flow.get("gtoken"); msg.headers = { "Authorization":token, "Content-type" : "application/json" } return msg;</pre>
function/http request	HTTP POST	Sendet die Nachricht an die Cloud	<p>URL:</p> <p>https://online.wondershare.eu/apis/upload/datasource</p>

Die *GreenBox* ist nun mit der Cloud verbunden und Daten werden übertragen.

5.1.5 Visualisierung

In diesem Kapitel werden Visualisierungen auf der *AVEVA Insight Cloud* erstellt. Dabei werden die freigegebenen Daten der *GreenBox* und die erstellten Testdaten verwendet.

5.1.5.1 Simulierte Signale

Die generierten Daten im *Abschnitt 5.1.3* werden als Tags im Speicher der Cloud gespeichert. Die Tags lauten *Simulation 1*, *Simulation 2* und *Simulation 3*. Durch Auswählen der Tags auf der Cloud werden die dazugehörigen Daten in einer Visualisierung angezeigt. Arten der Visualisierung sind Statusboard, Liniendiagramm, Säulendiagramm, Karte, Grafik, Detailraster oder XY-Graph. Über eine Zeitachse kann der Zeitbereich der gemessenen Daten benutzerdefiniert eingestellt werden.

Die im *Abschnitt 5.1.3* erzeugten Signale werden in einem Liniendiagramm in *Abbildung 13* grafisch dargestellt. Auf der X-Achse wird die Zeit und auf der Y-Achse die erzeugten Werte angezeigt. Zum Zeitpunkt 18:40 Uhr verringert sich die Amplitude der *Simulation 1*, da mit Hilfe des Flows in *Abschnitt 5.1.3.4* die Werte manipuliert wurden.

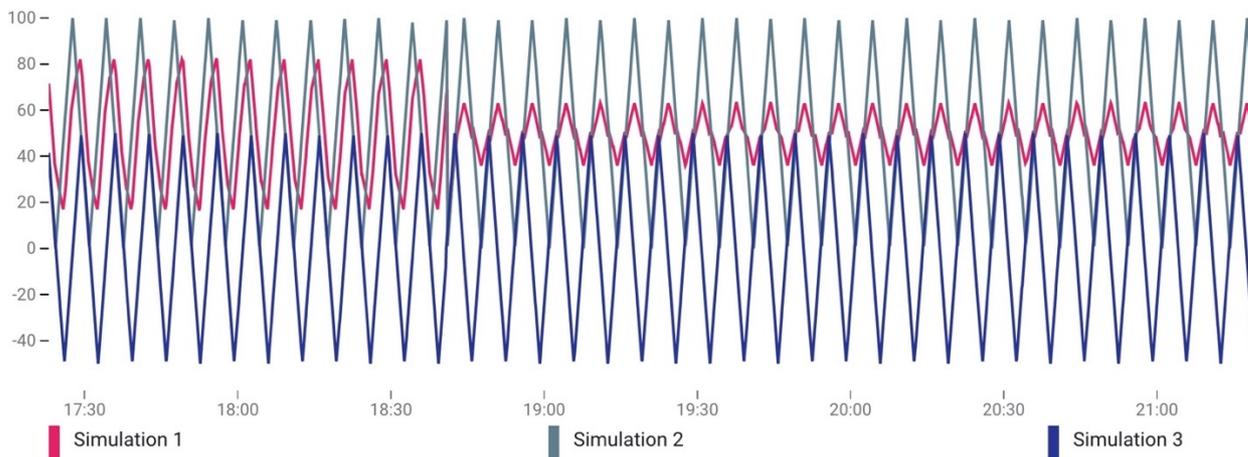


Abbildung 13: Visualisierung der simulierten Signale

In *Abbildung 14* ist als Visualisierungsart der Statusboard dargestellt. Es werden die Werte der Spannung in Volt und der CPU-Temperatur in Grad Celsius der letzten drei Tage angezeigt. Die grünen Trend-Symbole bedeuten, dass die Durchschnittswerte gestiegen sind. Dies lässt sich dadurch erklären, dass das Betriebssystem der *GreenBox* den Ruhezustand verlassen hat und in den Betriebszustand gewechselt ist.

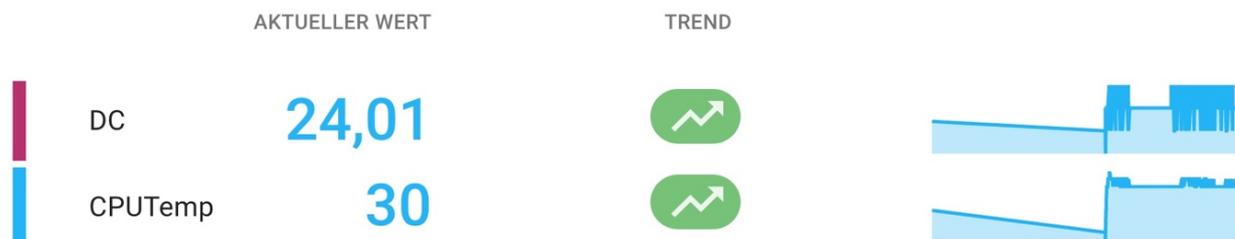


Abbildung 14: Visualisierung von CPU-Temperatur und Spannung der *GreenBox*

5.1.5.2 OEE-Überwachung

In *AVEVA Insight* ist eine Überwachung der OEE-Kennzahlen von Anlagen mit Hilfe einer Funktion [11] realisierbar. Dazu wird das in *Kapitel 3.2* beschriebene Anwendungsbeispiel einer Panadeanlage angewendet und simuliert (*Abbildung 15*). Die benötigten Daten werden mit Hilfe von Dateien im Tabellenformat unter *Menu > Administration > Efficiency* definiert und bekanntgegeben.

Der untere Balken in *Abbildung 15* stellt die Anlagenzustände innerhalb einer festgelegten Zeitspanne farblich dar. Die Bedeutung der Farben lautet wie folgt:

- Rot: Ungeplante Ausfallzeit der Anlage, z. B. durch Verursachung von Störungen.
- Gelb: Geplante Ausfallzeit der Anlage, z. B. durch Schichtwechsel.
- Grün: Aktive Betriebs- und Produktionszeit der Anlage.

Der grüne Knopf unter *Betriebsmittelstatus* bestimmt den Anlagenzustand. Der Schalter unter *Produktionslauf* legt fest, ob die Anlage gestartet oder gestoppt ist. Im Reiter *Produktionsergebnisse* werden Eigenschaften des Produktionslaufes festgelegt, wie z. B. die geplante Dauer, Menge oder die ideale Produktionsrate. Gut- und Ausschussmengen werden manuell erfasst, um den Qualitätsfaktor zu beeinflussen. Die Kennzahlen OEE, Verfügbarkeit, Leistung und Qualität werden anhand der Zustände über die Zeit berechnet und prozentual angezeigt.

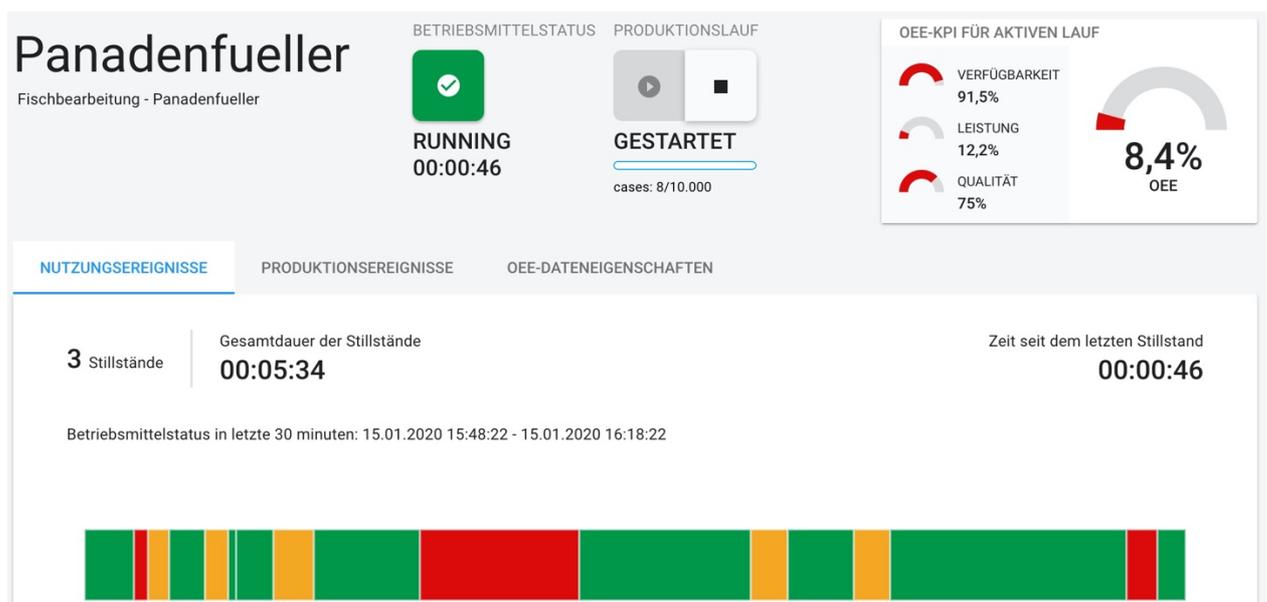


Abbildung 15: Simulation einer Panademaschine auf der *AVEVA Insight Cloud*

5.2 *IoT-Box* von *WAGO Kontakttechnik GmbH & Co. KG*

Die *WAGO IoT-Box* und die verwendeten Komponenten werden vom Hersteller kostenlos zum Testen für diese Arbeit zur Verfügung gestellt.

5.2.1 Aufbau und Funktionsweise der Hardware

Der Testaufbau des IoT-Systems enthält folgende Komponenten:

1. Controller PFC100 2ETH (*WAGO*, 750-8101)
2. 8-Kanal-Digitaleingang/-ausgang (*WAGO*, 750-1506)
3. 2/4-Kanal-Analogeingang (*WAGO*, 750-464)
4. 3-Phasen-Leistungsmessung (*WAGO*, 750-494)
5. Endmodul bzw. -klemme (*WAGO*, 750-600)
6. Verbindungsklemme
7. Zwei Ethernet-Anschlüsse (*X1* für oben, *X2* für unten)
8. 3 Aufsteck-Stromwandler (*WAGO*, 855-1700/032-000)
 - Primärer Bemessungsstrom $I_{\text{prim}} = 32 \text{ A}$
 - Sekundärer Bemessungsstrom $I_{\text{sek}} = 320 \text{ mA}$
9. Bauelementklemmen
10. Nano Router (tp-link, TL-WR802N)
11. 3 Schuko-Steckdosen *L1*, *L2*, *L3*
12. Stromversorgungsmodul (787-1602)
 - $U_{\text{out}} = 24 \text{ V, DC}$
 - $I_{\text{out}} = 1 \text{ A}$

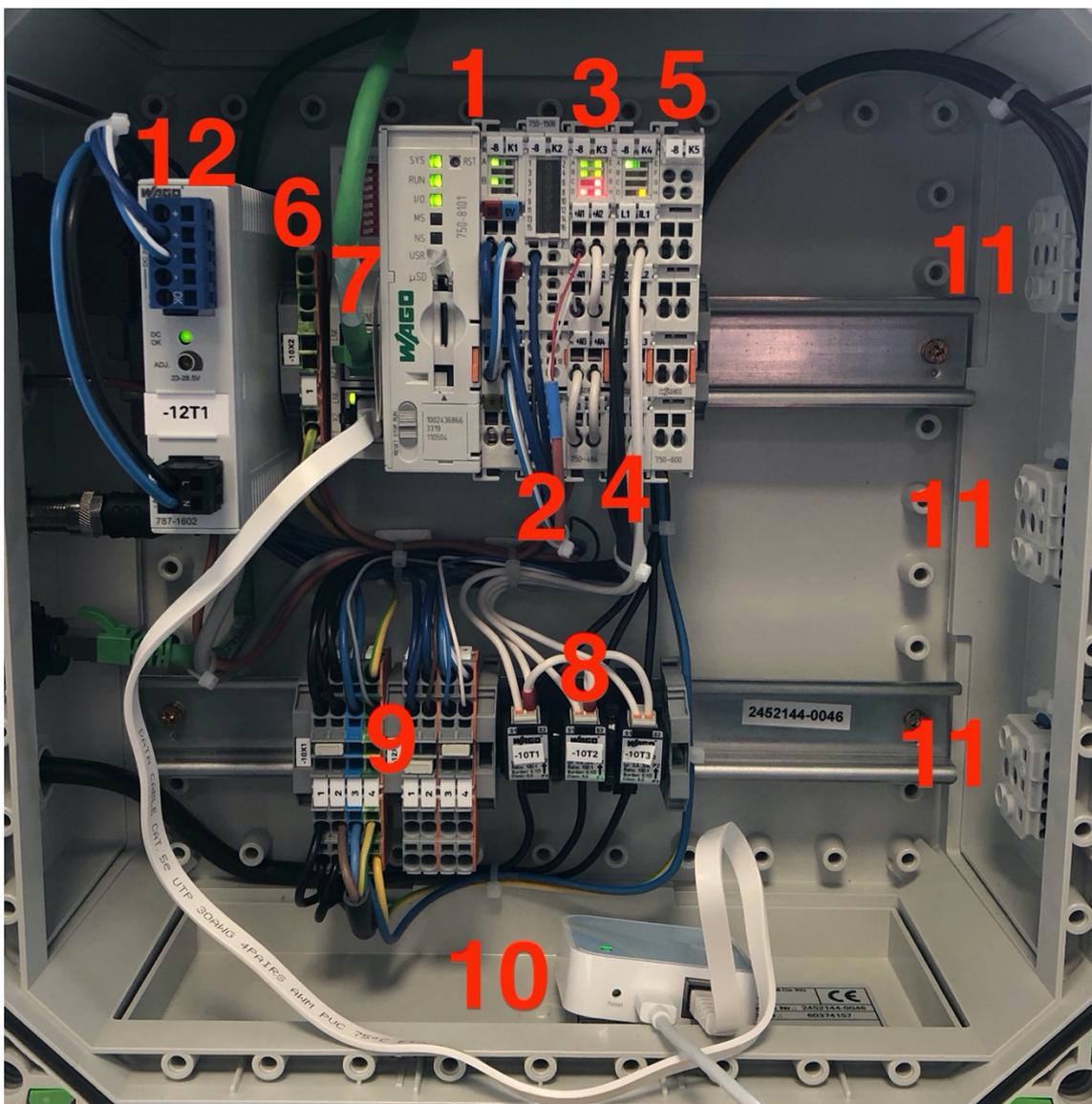


Abbildung 16: Testaufbau der WAGO IoT-Box Basic 9466

Das Gehäuse besteht aus Plastik und elektronische Komponenten werden auf zwei Schienen befestigt. Auf dem Gehäuse wird eine durchsichtige Plastikklappe mit Hilfe von Drehschrauben montiert. Von der Außenseite ist ein Ethernet-Anschluss erreichbar, das mit einer Gummidichtung geschützt ist. Das geschlossene Gehäuse erreicht die Schutzart IP44. Dadurch ist sie gegenüber festen Fremdkörpern mit einem Durchmesser von $\geq 1,0$ mm und Zugängen mit einem Draht sowie gegen allseitiges Spritzwasser geschützt [vgl. 12]. Nähere Informationen über die Bedeutung der Kennziffern sind in den Normen ISO 20653 und DIN EN 60529 festgelegt.

Der Controller (1) besitzt zwei Ethernet-Anschlüsse, ein SD-Karten-Slot und einen Zustandsschalter. Ein-/Ausgangsmodule (2) (3) (4) werden direkt mit dem Controller (1) verbunden und abschließend mit einem Endmodul (5) bestückt. Der obere Ethernet-

Anschluss X1 (7) wird für eine Verbindung mit einem externen Laptop bereitgestellt und der zweite Anschluss X2 wird mit einem WLAN-Modul (10) verknüpft. Das WLAN-Modul (10) greift auf ein Smartphone zu, das als WLAN-Hotspot agiert, worüber eine Verbindung zum Internet hergestellt wird. Die Stromversorgung verläuft über eine Haushaltssteckdose mit 230 V zu den Klemmen (9). Von den Klemmen aus werden Phasen aufgeteilt und alle Komponenten mit Spannung versorgt. Die Stromwandler (8) dienen als Sensoren für die Leistungsmessung. Diese sind mit den Schuko-Steckdosen (11) und der direkten Stromversorgung über drei Phasen verbunden.

5.2.2 Vorkonfiguration

Um auf das vorinstallierte System des Controllers (1) über den Ethernet-Anschluss X1 zuzugreifen, muss die IP-Adresse bekannt sein. Dazu ist der Zustandsschalter des Controllers auf STOP zu setzen. Anschließend wird der Reset-Button RST acht Sekunden gedrückt gehalten. Sobald alle Blinker grün aufleuchten, ist das System unter der Standard-IP-Adresse 192.168.1.17 und der Subnetzmaske 255.255.255.0 erreichbar. Über das HTML-basierte Portal, erreichbar unter der URL `http://192.168.1.17/wbm/index.php`, können allgemeine Einstellungen im *Web-based Management* (siehe *Abbildung 17*) vorgenommen werden.

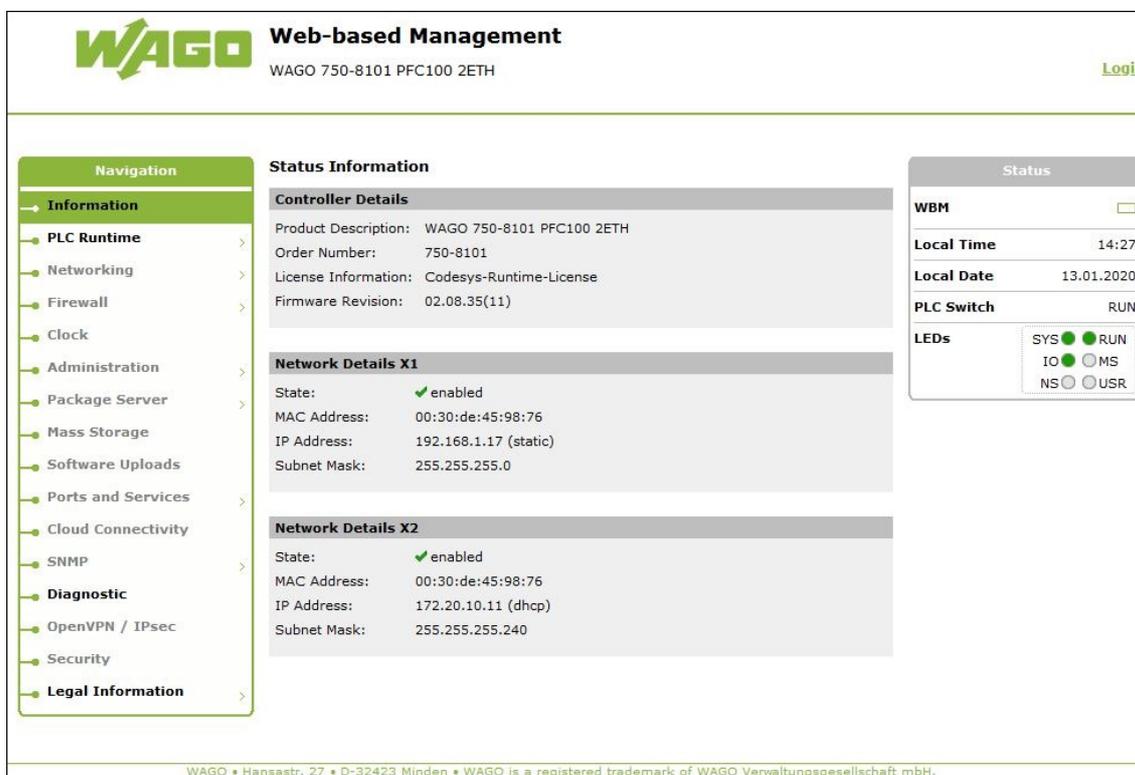


Abbildung 17: Software *Web-based Management* der Firma WAGO

Die Standardzugangsdaten lauten *admin* für den Nutzer und *wago* für das zugehörige Passwort. Unter *Navigation > Networking > TCP/IP* wird der Ethernet-Anschluss *X1* als eine statische und der Anschluss *X2* als eine DHCP-basierte IP-Adresse festgelegt. Damit ist das System über die feste IP-Adresse 192.168.1.17 erreichbar. Über den Anschluss *X2* wird das Internet des Hotspots mit Hilfe von DHCP bezogen.

Daten werden mit Hilfe des MQTT-Protokolls über den offiziellen Port 8883 übertragen. Der Port 8883 muss dabei im Netzwerk für eine erfolgreiche Übertragung freigegeben sein.

5.2.3 Festlegen der Übertragungsdaten

Es gibt zwei Varianten zur Festlegung der Übertragungsdaten für die Cloud. Nachfolgend werden diese näher beschrieben.

5.2.3.1 Variante 1

Datenquellen können mit Hilfe der *WAGO e!COCKPIT Automation Software* festgelegt werden. Für eine Programmierung stehen folgende Programmiersprachen zur Auswahl:

- Ablaufsprache (AS)
- Anweisungsliste (AWL)
- Continuous Function Chart (CFC)
- Funktionsbausteinsprache (FUP)
- Kontaktplan (KOP)
- Strukturierter Text (ST)

Zur Verwendung der Software wird eine Lizenz benötigt, die von *WAGO* kostenlos zur Verfügung gestellt wird. In einer vorhandenen Bibliothek werden die verwendeten Hardware-Module vom IoT-System ausgewählt. Anschließend wird eine visuelle Übersicht (*Abbildung 17*) angezeigt.

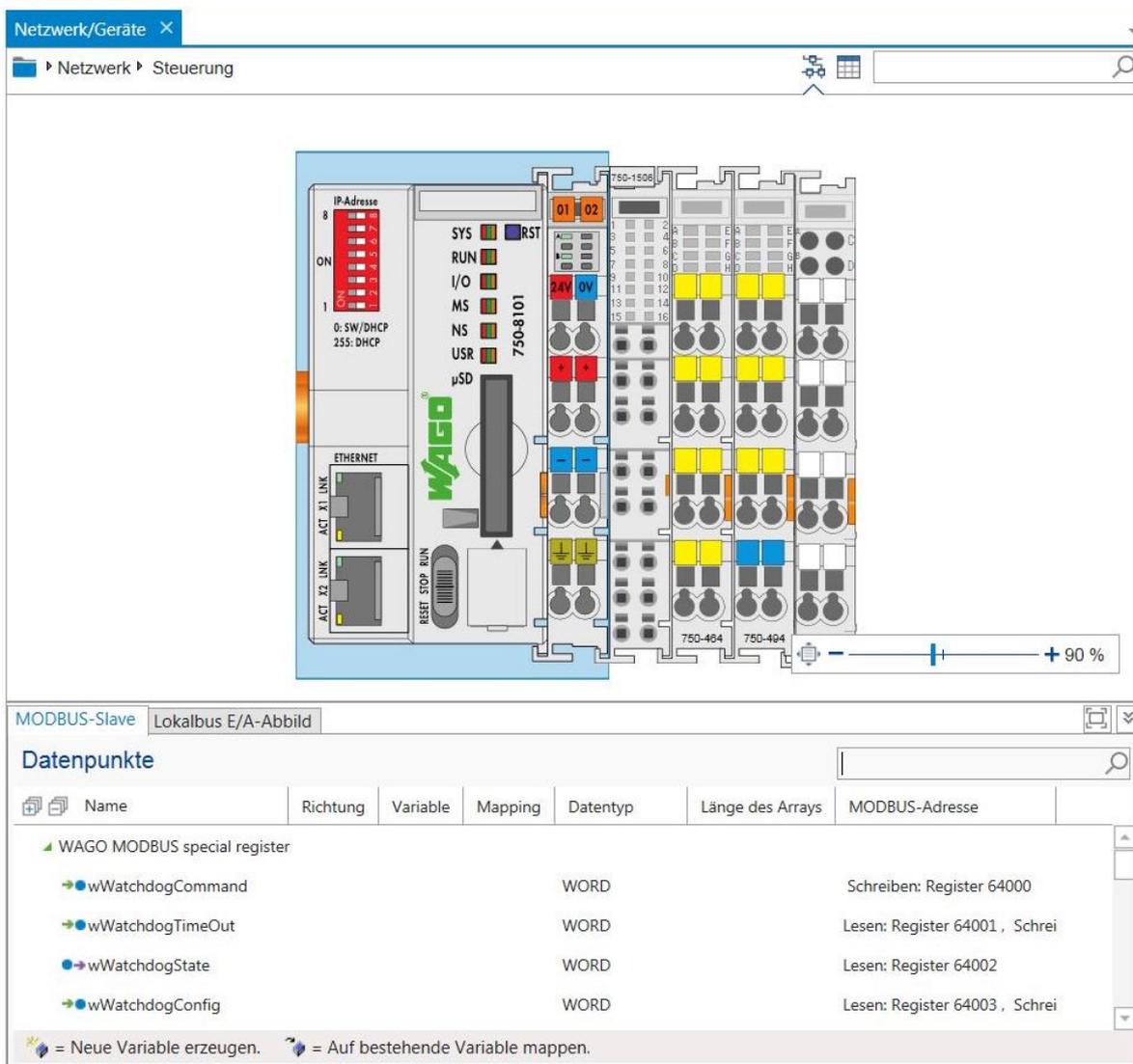


Abbildung 18: Ausschnitt der Softwareoberfläche e!COCKPIT von WAGO

5.2.3.2 Variante 2

Die Auswahl von Datenquellen erfolgt über das webbasierte Add-on *Cloud IO Connector v1.0.1.2*, das von WAGO zur Verfügung gestellt wird. Nach Download der Datei *AutoUpdate_lot8101.zip* [13] wird das Add-on auf eine SD-Karte mit 2 GB Speicherkapazität entpackt und in den SD-Slot des Controllers eingesetzt. Der SPS-Schalter ist auf den Zustand *STOP* gestellt und der Controller wird neugestartet. Die Installation beginnt und wird beendet, sobald alle Leuchten des Controllers dauerhaft grün leuchten. Das *Cloud IO Connector*-Dashboard (siehe *Abbildung 19*) ist anschließend über der lokalen IP-Adresse unter der URL <http://192.168.1.17/webvisu/webvisu.html> erreichbar.

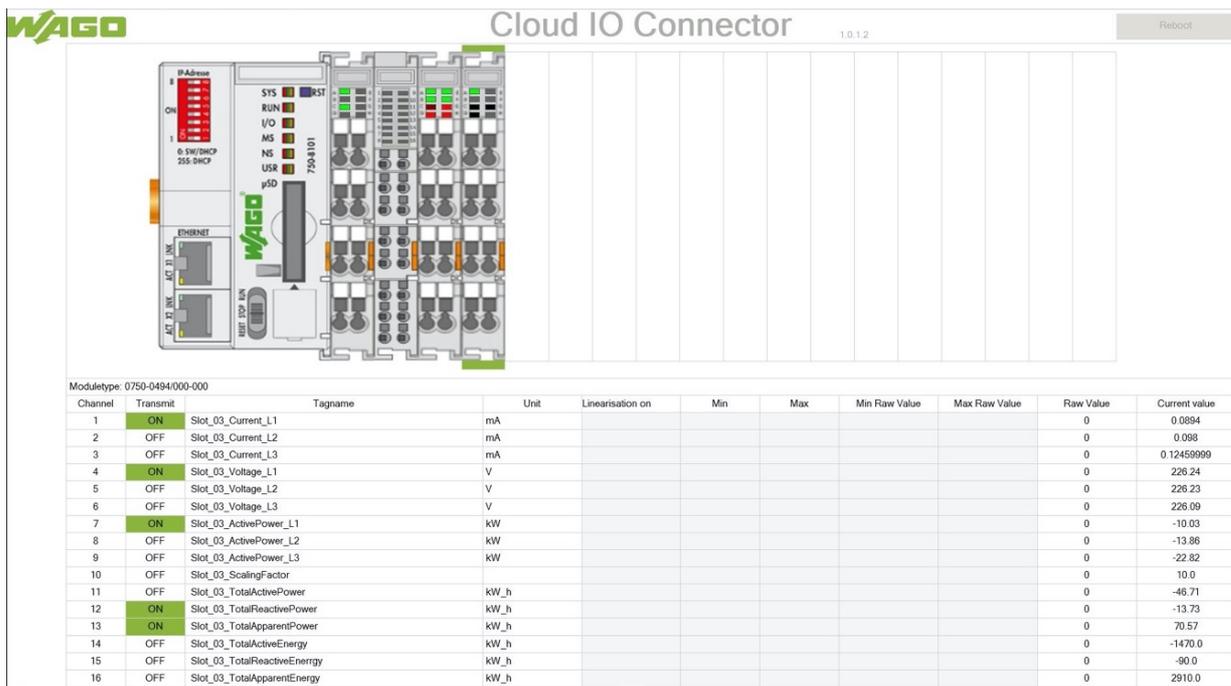


Abbildung 19: Dashboard des Add-ons *Cloud IO Connector* der Firma WAGO

In der dargestellten Tabelle in *Abbildung 19* werden Datenquellen angezeigt, die mit Ein-/Ausgangsmodulen an der Steuerung verbunden sind. Die Spalte *Channel* stellt den Pin eines angeschlossenen Moduls dar. Im oberen Bereich der Abbildung befindet sich eine Visualisierung der Hardware, worüber die angeschlossenen Karten ausgewählt und bearbeitet werden können. In der Spalte *Transmit* wird festgelegt, ob die Datenquelle zur Cloud übertragen werden soll. Übertragene Datenquellen sind grün markiert. In der Spalte *Unit* wird die Einheit des empfangenen Wertes definiert, z. B. V für Volt. Unter *Current value* wird der zuletzt gemessene Wert angezeigt, der sich in Echtzeit aktualisiert.

Für eine Visualisierung werden Daten der Schuko-Steckdose *L1* für eine Übertragung an die Cloud freigegeben. Darunter zählen die Spannung, der Strom und die Wirkleistung. Mit Hilfe eines internen Temperatursensors der Steuerung wird die Temperatur im Gehäuse gemessen. Dieser Wert wird für eine Freigabe markiert. Zusätzlich werden die Summen der Wirk-, Blind- und Scheinleistung des IoT-Systems für eine grafische Darstellung markiert.

5.2.4 Cloud-Anbindung

Als Cloud-Plattform wird die *WAGO Cloud* verwendet. Die Plattform basiert auf der *Microsoft Azure Cloud* und ist unter der URL <https://cloud.wago.com> erreichbar. Nach Erstellung eines Accounts sind 30 Tage Testlaufzeit kostenlos verfügbar.

Für eine Anbindung der *WAGO IoT-Box* zur Cloud muss über das System des Controllers der Host, die Geräte-ID und der Aktivierungsschlüssel bekanntgegeben werden. Diese Daten werden auf der Cloud erzeugt. Sie werden für eine verschlüsselte Verbindung und Bekanntgabe der *WAGO IoT-Box* benötigt.

Zuerst wird auf der Cloudsoftware unter *Arbeitsbereiche* ein Arbeitsbereich festgelegt. Dieser bekommt den Namen *Stadler + Schaaf Mess- und Regeltechnik* zugewiesen. Unter *Geräteverwaltung > Konfiguration > Neu > Gerät* wird ein virtuelles Gerät mit dem Namen *WAGO IoT-Box* erstellt. Parameter für eine Anbindung zur Cloud werden generiert. Darunter zählen ein individueller Aktivierungsschlüssel und die Geräte-ID.

In den Einstellungen des *WAGO-Controllers* unter *Navigation > Cloud Connectivity* werden folgende Daten für eine Cloud-Anbindung eingegeben:

- Cloud platform: *Wago Cloud*
- Hostname: *wagocloud.azure-devices.net*
- Device ID: *765d0e3c-8372-45c9-8504-a9fd5c3f8d29*
- Activation Key: *8iVJLxIHbgJtUYsudyApsf3g0jOPE8UEENQhKfAHmgg=*

Anschließend wird das System neugestartet. Nach erfolgreichem Neustart werden die Daten erfolgreich auf die Cloud übertragen. Der Status kann unter *Geräteverwaltung > Statusübersicht* auf der *WAGO-Cloud* überprüft werden.

① Statusübersicht

Struktur		Übersicht	
<input type="text" value=""/> <input type="button" value="+"/> <input type="button" value="-"/>		Verbindungsstatus 1 Verbunden, 0 Getrennt, 0 Telemetried...	
Stadler + Schaaf Mess- und Regeltechnik		Datenübertragungsstatus 1 Aktiviert, 0 Deaktiviert	
WAGO IoT-Box		Alarme 0 Alarm(e)	

Abbildung 20: Statusübersicht der *WAGO IoT-Box* auf der *WAGO Cloud*

Das Übertragungsintervall der Daten kann in der *WAGO-Cloud* eingestellt werden. Dazu werden sogenannte Lizenzpunkte benötigt, die gekauft werden können. Eine

Funktion für die Berechnung der Cloud-Kosten wird von WAGO über die URL <https://wagocloudcostcalculator.azurewebsites.net/de> bereitgestellt.

5.2.5 Visualisierung

In diesem Kapitel werden die im *Abschnitt 5.2.3* freigegebenen Daten in Grafiken und Tabellen veranschaulicht.

Mit Hilfe der Funktion *Visualisierungen* auf der *WAGO Cloud* werden Visualisierungen erstellt. Dazu wird innerhalb des festgelegten Arbeitsbereiches *Stadler + Schaaf Mess- und Regeltechnik* ein Dashboard mit dem Namen *WAGO IoT-Box* definiert. Unter *Neu* werden Texte, Tabellen, Karten, Tachos, Werte und Befehlsschaltflächen eingerichtet.

Als Beispiel wird in *Abbildung 21* ein Kartenelement erstellt. Dazu wird der Breiten- und Längengrad vom Standort des IoT-Systems bekanntgegeben. Auf dem Dashboard ist der Standort der *WAGO IoT-Box* in einem Kartenabschnitt dargestellt. In einem Tacho-Element wird die Datenquelle *Temperatur* zugewiesen. Dieser zeigt den letzten übertragenen Wert der *WAGO IoT-Box* an. Für den Tacho müssen die Skalierung, der Start- und der Endwert definiert sein. Um normale und kritische Bereiche der Temperatur visuell zu unterscheiden, sind diese mit den Farben blau, grün und rot markiert. In einem Textelement werden Werte in Texte umgewandelt. Werte können auch alleinstehend angezeigt werden.

Dashboard > WAGO IoT-Box

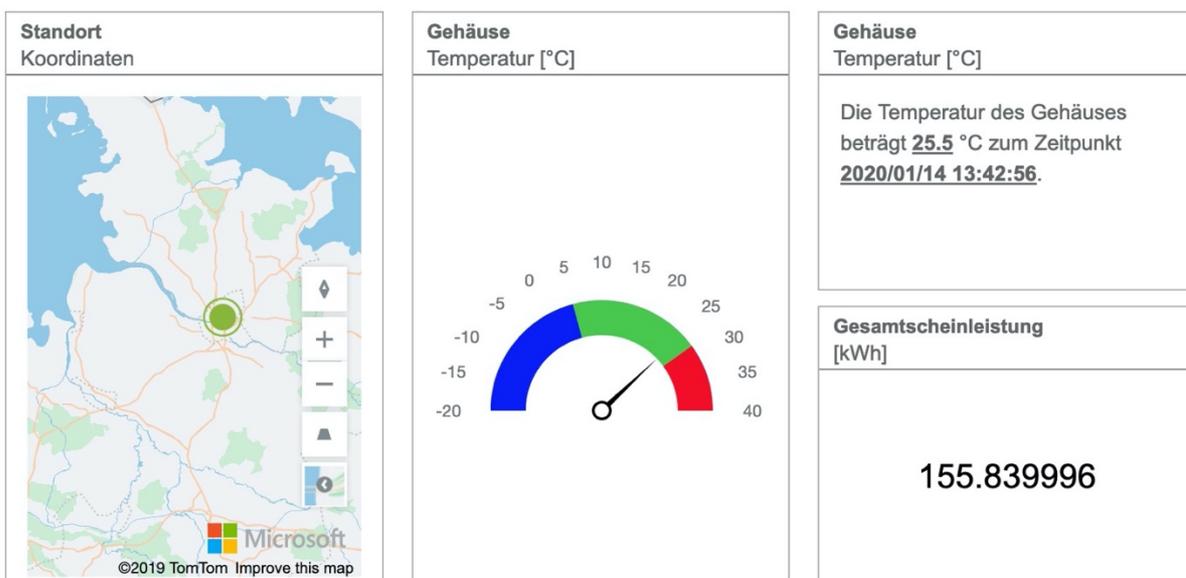


Abbildung 21: Dashboard mit Visualisierungselementen

In *Abbildung 22* sind die Strom-, Spannungs- und Wirkleistungswerte der Schuko-Steckdose *L1* dargestellt. Ein negativer Wert des Stroms und der Wirkleistung weist darauf hin, dass kein Verbraucher angeschlossen ist. Zum Zeitpunkt 13:17 Uhr wird die Steckdose *L1* mit einem Kabel zum Aufladen eines Laptops verbunden. Daraufhin steigen der Stromverbrauch und die Wirkleistung jeweils vom negativen in den positiven Bereich. Die Spannungsversorgung an der Steckdose bleibt konstant zwischen 227 V und 229 V.

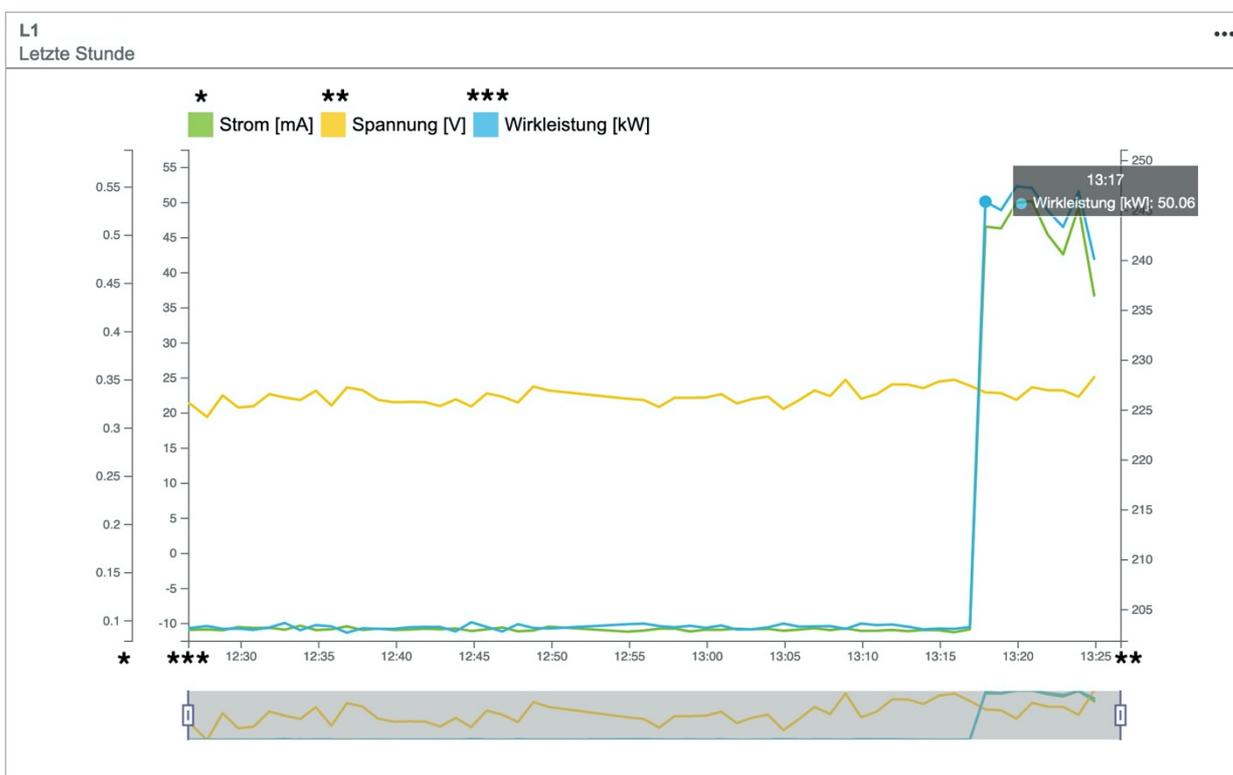


Abbildung 22: Übersicht von Daten der Schuko-Steckdose *L1*

5.3 *MindSphere* von Siemens AG

Nachfolgend wird die *MindSphere* näher beschrieben und eine Kommunikation zwischen einer SPS und die *MindSphere* aufgebaut.

5.3.1 Aufbau und Funktionsweise des Systems

Die *MindSphere* ist eine offene und cloudbasierte IoT-Plattform. Gehostet wird sie auf Servern von *Siemens*. Für das Testen steht ein Testzugang kostenlos zur Verfügung. Über einen erstellten Account wird auf die Plattform über die URL <https://sus2019.mindsphere.io> zugegriffen.

Die *MindSphere* bietet auf der HTML5-Oberfläche vorgefertigte Applikationen an. Dazu ist ein Portal vorhanden, worüber kostenlose und -pflichtige Applikationen installiert werden können. Über die Applikation *Asset Manager* werden Verbindungen zwischen Steuerungen und der *MindSphere* konfiguriert. Für eine Konfiguration im *Asset Manager* müssen dabei sogenannte *Assets*, *Types*, *Aspects* und *Variablen* erstellt werden. Beispielsweise ist ein *Asset* als eine Produktionslinie für die Fischbearbeitung definiert. Ein Teil dieser Anlage ist für den Aufguss der Panade zuständig, dieser wird als eine *Type* definiert. Eine Teilkomponente der Maschine ist der Elektromotor, der das Förderband antreibt. Dieser ist als ein *Aspect* anzusehen. Die Motordrehzahl n eines Elektromotors ist als eine *Variable* definiert.

Tabelle 12: Beispiel einer Typenzuweisung

Typenbezeichnung	Definition
Asset	Produktionsanlage für die Fischbearbeitung
Type	Maschine für den Aufguss
Aspect	Elektromotor
Variable	Motordrehzahl n

Als Verbindungselement zwischen den Datenquellen der Anlage und den *Variablen* in der *MindSphere* dient die in *Abbildung 23* dargestellten Steuerung *Siemens SIMATIC S7-1500*. Die Spannungsversorgung an der CPU beträgt DC 24 V.



Abbildung 23: Steuerung Siemens S7-1500 6ES7511-1AK02-0AB0 [14]

Die Steuerung wird an der Anlage über Bussysteme verbunden. Der Datenaustausch erfolgt über die Protokolle S7 SPS oder OPC UA. Die S7-1500-SPS weist mit Hilfe vom Onboarding, sprich die Vorkonfiguration von Einstellungen, die Datenpunkte den *Variablen* auf der Cloud automatisch zu und synchronisiert diese.

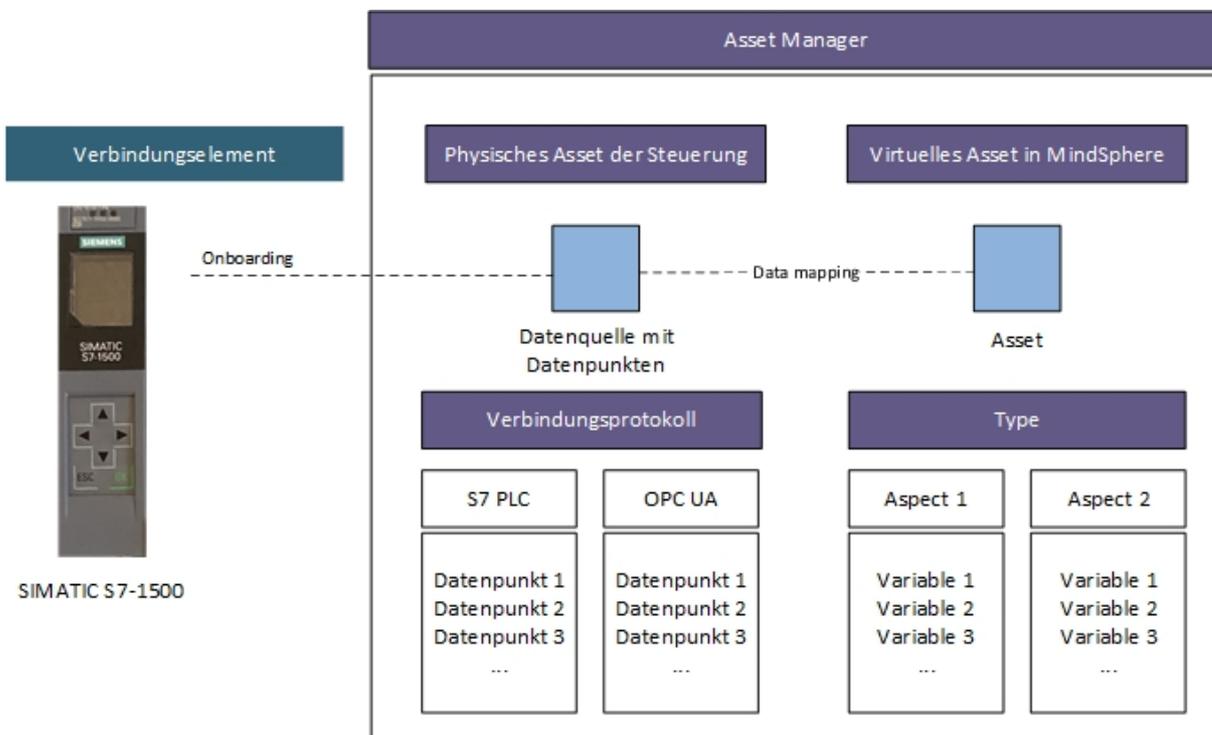


Abbildung 24: Datenmodell der *MindSphere*, vgl. [15, S. 11]

5.3.2 Programmierung

Die Programmierung erfolgt über eine Software namens *Siemens TIA Portal V15.1*. *Siemens* stellt dem Nutzer Bibliotheken mit Funktionsbausteinen zur Verfügung. Diese können auf der Homepage von *Siemens* [16] heruntergeladen werden. Mit Hilfe dieser Funktionsbausteine kann eine Verbindung zwischen der SPS und *MindSphere* hergestellt werden. Die Bibliothek *MindConnectFB* wird für eine Verbindung zwischen der SPS in *Abbildung 23* und der *MindSphere* verwendet. Sie beinhaltet die in *Tabelle 13* genannten Funktionsbausteine.

Tabelle 13: Funktion der benötigten Bausteine

Funktionsbaustein (FB)	Funktion (FC)	Beschreibung
MCFB_Communication		Kommunikationsablauf zwischen der SPS <i>S7-1500</i> und <i>MindSphere</i>
	MCFB_CollectDataSource	Definiert die Datenquelle für die <i>MindSphere</i>
	MCFB_CollectDataValue	Erfasst Werte von Variablen in einem Array
	MCFB_DemoBuffer	Dient als Datenpuffer zum Zwischenspeichern von Übertragungsdaten
	MCFB_DataSource	Gibt die Art der Datenquelle an

In *Abbildung 25* ist der Kommunikationsablauf zwischen den Bausteinen, Funktionen und der *MindSphere* dargestellt. Der Zustandsverwalter steuert den Ablauf und die Kommunikation im Netzwerk. Dieser ist mit *MCFB_Communication*, *MCFB_CollectDataSource*, *MCFB_CollectDataValue* und einem Organisationsbaustein, in diesem Fall das Hauptprogramm, verbunden. In der Funktion *MCFB_CollectDataSource* werden Datenquellen bekanntgegeben und der Datenfluss wird weitergeleitet. In der Funktion *MCFB_CollectDataValue* werden Daten gesammelt und gespeichert. Der *MCFB_DemoBuffer* dient als Zwischenspeicher von Daten. Der Baustein

MCFB_Communication überträgt die Daten mit Hilfe eines Sicherheitstokens zur *MindSphere*.

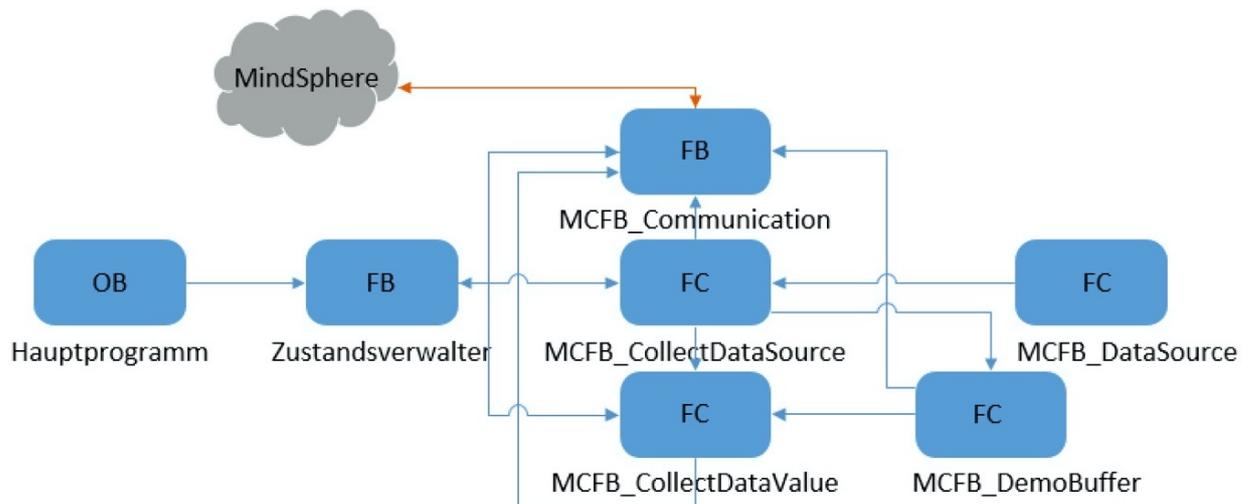


Abbildung 25: Kommunikationsablauf von der SPS zur *MindSphere*

5.3.3 Fazit

Die Cloud-Anbindung zwischen der *MindSphere* und der *S7-1500-SPS* konnte im Zeitrahmen dieser Bachelorarbeit nicht realisiert werden. Gründe hierfür waren die verzögerten Antwortzeiten des technischen Supports. Dies führte zum Ablauf des *MindSphere*-Testzuganges, ohne die Möglichkeit einer erfolgreichen Verbindung.

6 Bewertung der IoT-Systeme und Auswahl

In diesem Kapitel werden zwei der drei IoT-Systeme mittels der festgelegten Bewertungskriterien in *Kapitel 4* bewertet: Die *WAGO IoT-Box* und die *GreenBox ECO*. Die *MindSphere* wurde begrenzt getestet, sodass diese für eine vollständige Bewertung nicht mitberücksichtigt wird.

6.1 Bewertungsmethode

Zur Entscheidungsfindung wird die Nutzwertanalyse angewendet. Diese Methode eignet sich für eine Bewertung der Effektivität eines Gesamtsystems hinsichtlich der in *Kapitel 4* festgelegten Kriterien.

Innerhalb einer Bewertungsmatrix werden spaltenweise IoT-Systeme und zeilenweise Bewertungskriterien aufgezählt. Alle Bewertungskriterien werden für die Übersichtlichkeit in Bewertungsbereiche unterteilt. Der Gesamtgewichtungsfaktor G gibt die Wichtigkeit der jeweiligen Bewertungsbereiche und -kriterien an. Die Summe aller Gewichtungs faktoren G_i für die jeweiligen Bewertungskriterien ergibt den Gesamtgewichtungs faktor G 100:

$$G = \sum G_i = 100 \quad (1)$$

Der Wert W_i bewertet eine IoT-Lösung in Bezug auf das jeweilige Bewertungskriterium. Dieser enthält einen natürlichen Wert zwischen 0 und 10, wobei die 0 als negative und die 10 als positive Bewertung anzusehen ist:

$$W_i = \{x \in \mathbb{N}_0 \mid 0 \leq x \leq 10\} \quad (2)$$

Der gewichtete Wert GW stellt den Zusammenhang zwischen den bewerteten Kriterien für ein System und der Wichtigkeit der Kriterien her. GW ist das Ergebnis aus der Multiplikation des Gesamtwertes W und dem Gesamtgewichtsfaktor G :

$$W = \sum W_i \quad (3)$$

$$GW = G * W \quad (4)$$

Das Maximum des gewichteten Wertes GW_{max} beträgt 1000 und stellt die maximale Bewertungszahl für eine Lösung dar:

$$GW_{max} = 1000 \quad (5)$$

6.2 Bewertung

Die vollständige Bewertungsmatrix befindet sich in *Tabelle 14*.

Die *WAGO IoT-Box* und die *GreenBox ECO* besitzen gemeinsame Eigenschaften. Sie unterstützen HTML5 auf der Cloud sowie auf beiden Betriebssystemen der lokalen IoT-Steuerung. Ein Dashboard kann auf beiden Systemen erstellt und individuell gestaltet werden. Beide Systeme sind übersichtlich gestaltet. Systemrechte für Benutzer können auf beiden Geräten gleichermaßen eingestellt werden. Zur Übertragung von Maschinendaten bieten beide Systeme die Protokolle Modbus und MQTT an und Daten werden verschlüsselt übertragen. Beide Hersteller lieferten einen einwandfreien Support und deren Systeme ermöglichen einen Fernzugriff. Die für diese Arbeit berücksichtigten Kosten der Systeme und Cloud-Plattformen weichen minimal voneinander ab. Die *WAGO Cloud* bietet dabei eine Funktion, die die voraussichtlichen Kosten in Bezug auf Datenübertragungen berechnet. Beide Unternehmen bieten mobile Applikationen für die Betriebssysteme *iOS* und *Android* an, worüber Visualisierungen eingebunden und angezeigt werden können.

Unterschiede zwischen den beiden IoT-Systemen finden sich zum einen in den Visualisierungen der Oberflächenelemente und Anordnungen auf den Clouds. Die *AVEVA Insight Cloud* wird dabei am fortschrittlichsten dargestellt. Mathematische Verknüpfungen von Datenpunkte können mit Hilfe von *Node-RED* auf der *GreenBox* problemlos verknüpft werden. Für die *WAGO IoT-Box* wird dies über die Software *e!COCKPIT* aufwendiger programmiert. Visualisierungen lassen sich auf der *WAGO Cloud* beschränkter einstellen als auf der *AVEVA Insight Cloud*. Die *AVEVA Cloud* bietet umfangreichere Graphen und Tabellen als Optionen für eine Visualisierung an. Zum anderen können zusätzliche *WAGO*-Module an der *WAGO*-Steuerung mit minimalem Aufwand angebunden werden. Die Hardware der *WAGO IoT-Box* ist raumsparender als die *GreenBox*. Eine lokale Softwareerweiterung lässt sich auf der *GreenBox* optimaler realisieren. Fehlende Funktionen können seitens des Herstellers erweitert werden.

Aus der Nutzwertanalyse erreicht die *WAGO IoT-Box* 814 Punkte und die *GreenBox ECO* 811 Punkte von insgesamt 1000 Bewertungspunkten.

Tabelle 14: Bewertungsmatrix

Legende		WAGO Kontakttechnik GmbH & Co. KG IoT-Box			Schneider Electric SE GreenBox ECO		
- Wert W [0-3=ungenügend/4=ausreichend/5-6=befriedigend/7-8=gut/9-10=sehr gut]		WAGO Kontakttechnik GmbH & Co. KG IoT-Box		Schneider Electric SE GreenBox ECO			
- Gewichtung G; gewichteter Wert GW = G*W		WAGO Kontakttechnik GmbH & Co. KG IoT-Box		Schneider Electric SE GreenBox ECO			
Bewertungsbereich	Bewertungskriterium	Gewichtung Gi [%]	Faktor Gewichtung Gi	Wert Wi	Gewichteter Wert Gi*Wi	Wert Wi	gewichteter Wert Gi*Wi
Funktionalität	HTML5	100	30	7	10,5	10	15
	Dashboard	5	1,5	7	52,5	8	60
	Datennormalisierung	25	7,5	6	45	8	60
	Visualisierung	10	3	5	15	9	27
	Analytik	25	7,5	6	45	9	67,5
	Benutzermanagement	10	3	10	30	9	27
Konnektivität		100	25				
	Hardware	100	25	10	250	7	175
Software		100	30				
	Verschlüsselte Kommunikation	15	4,5	10	45	10	45
	Protokolle	20	6	10	60	10	60
	Erweiterbarkeit	20	6	5	30	8	48
	Benutzerfreundlichkeit	20	6	8	48	8	48
	Fernwartung	10	3	9	27	9	27
	Herstellersupport	15	4,5	10	45	10	45
Kosten		100	15				
	Cloud	30	4,5	8	36	7	31,5
	Hardware	20	3	7	21	7	21
	Softwarelizenzen	10	1,5	8	12	8	12
	Datenpunkte	40	6	7	42	7	42
Gewichtete Wertigkeit (GW)					814		811
Gewichtete Wertigkeit (GW_max)					1000		1000

6.3 Auswahl

Die Auswahl der IoT-Hardware fällt auf die *WAGO IoT-Box*.

Die Komponenten des *WAGO*-Systems sind größentechnisch kompakter als die Hardware von der *GreenBox*. Für den Einsatz in der Lebensmittelindustrie sollte das Gehäuse jedoch aus Aluminium bestehen und eine höhere Schutzklasse als IP44 besitzen. Eine Hardwareerweiterung erfolgt seitens der *WAGO*-Steuerung komfortabel. Zusatzmodule von *WAGO* lassen sich zeiteffizient an die Steuerung anschließen und Datenquellen im Rahmen des Add-ons *Cloud IO Connector* übersichtlich darstellen.

Die Auswahl der Cloud fällt auf die *Wonderware AVEVA Insight*.

Visualisierungen auf der *WAGO Cloud* sind beschränkt ausgeprägt. Die Cloud-Plattform *Wonderware AVEVA Insight* von *Schneider Electric SE* bietet ein breiteres Spektrum an Funktionalität. Dadurch lassen sich Daten optimaler visualisieren und analysieren. Arten von Visualisierungen wie Diagramme, Tabellen und Graphen sind umfangreicher gestaltet. Der visuelle Aufbau der *AVEVA Insight Cloud* ist deshalb fortschrittlicher.

Eine Kombination aus der *WAGO IoT-Box* und der Cloud *Wonderware AVEVA Insight* ist die optimale Lösung.

7 Zusammenfassung und Schlussfolgerung

Es wurden vorhandene Industrial-IoT-Systeme auf dem Markt und der Stand der Technik ermittelt. Die Anbieter *Schneider Electric SE*, *WAGO Kontakttechnik GmbH & Co. KG* und *Siemens AG* wurden in die Firma *Stadler + Schaaf* eingeladen und haben ihre Lösungen präsentiert. Alle IoT-Systeme wurden kostenlos zum Testen zur Verfügung gestellt. Dazu gehörten die *GreenBox ECO*, die *WAGO IoT-Box Basic 9466* und die *MindSphere*. Anforderungen und Bewertungskriterien wurden zur Bestimmung für eine optimale Lösung festgelegt. Der Aufbau und die Funktionsweise der IoT-Systeme wurden analysiert. Dabei wurden die IoT-Systeme aufgesetzt und mit dem lokalen Netzwerk verbunden. Testdaten wurden mit Hilfe der Programmierung in *RED-Node* generiert und Signale simuliert. Als Datenquellen dienten Testdaten und über Leistungssensoren gewonnene Daten einer Schuko-Steckdose. Beide IoT-Systeme konnten sich mit ihrer hauseigenen Cloud erfolgreich verbinden, die *WAGO IoT-Box* mit der *WAGO Cloud* sowie die *GreenBox ECO* mit der *AVEVA Insight Cloud*. Die *S7-1500*-Steuerung und die *MindSphere* von *Siemens* können nicht mit in die Bewertung einfließen. Entstandene Fehler während des Programmierens und die verzögerten Antwortzeiten des technischen Supports führten dazu, dass die Testlaufzeit für die *MindSphere* erlosch. Dadurch konnte keine erfolgreiche Verbindung zwischen der Hardware und Cloud hergestellt werden. Die freigegebenen Daten der *GreenBox* und der *WAGO IoT-Box* wurden erfolgreich auf die Cloud-Plattformen übertragen. Visualisierungen konnten auf beide Clouds aufgesetzt werden, worüber Daten angezeigt und analysiert wurden. Auf der *WAGO-Cloud* konnten Messwerte des Controllers und einer Schuko-Steckdose angezeigt werden. Auf der *AVEVA Insight Cloud* wurden eine OEE-Überwachung anhand einer Panademaschine, *GreenBox*-Daten und simulierte Signale veranschaulicht werden. Nach dem Testen wurden beide Systeme mit Hilfe der Nutzwertanalyse in einer Matrix bewertet. In der Bewertung flossen die Bewertungskriterien sowie die während des Testens gewonnenen Erfahrungen ein. Die optimale Lösung ergibt sich aus einer Kombination der *WAGO IoT-Box* und der *AVEVA Insight Cloud*. Die *WAGO IoT-Box* konnte bezüglich der Hardware überzeugen und die *AVEVA Insight Cloud* bezüglich ihrer Funktionalität. Eine Kombination aus der *WAGO-Hardware* sowie der *AVEVA Insight Cloud* ist realisierbar. Die Vorteile eines IoT-Systems lassen sich anhand der Visualisierung von OEE-Kennzahlen optimal veranschaulichen.

Diese Arbeit hat dazu beigetragen, dem Unternehmen *Stadler + Schaaf* die Suche nach einem optimalen IoT-System einzugrenzen.

8 Ausblick

Die Definition der Aufgabenstellung erwies sich schwieriger als gedacht. Das Thema IoT ist breit gefächert und musste eingegrenzt werden. Die vorzeitige Gliederung aller Kapitel und Erstellung eines Zeitplans halfen, den roten Faden zu behalten. Zwei IoT-Systeme wurden zum Jahreswechsel (Stand 2020) verzögert geliefert, wodurch die Bearbeitungszeit stark eingeschränkt wurde. Erhöhter Zeitdruck war die Folge. Die *MindSphere* konnte im Rahmen dieser Arbeit nicht vollständig getestet werden. Als Alternative zu einer fertigen IoT-Lösung wäre ein eigenentwickeltes IoT-System. Ein Fokus auf die Entwicklung könnte in einer zukünftigen Arbeit thematisiert werden. Die Teilbereiche des Themas IoT können ebenfalls tiefgründiger erforscht werden. IoT-Systeme für den industriellen Einsatz werden zunehmend komplexer und stetig weiterentwickelt. Dazu bietet die eingesetzte Software viel Spielraum, erweitert und optimiert zu werden. Im Zeitalter der Digitalisierung spielt die Sicherheit für die Industrie eine erhebliche Rolle. Firmengeheimnisse wie Produktionsabläufe von der in dieser Arbeit als Beispiel genannten Panademaschine liefern dem IoT-System sensible Informationen, die nicht in fremde Hände gelangen dürfen. Durch Cyberangriffe in der Vergangenheit wurden Unternehmen erfolgreich attackiert. Systeme wurden kompromittiert und Produktionsanlagen lahmgelegt. In Folge dessen entstanden massive Verluste. Die vorgestellten IoT-Systeme gewähren bei ständiger Wartung, in Form von Softwareupdates, Sicherheit für Firmen und ihr geistiges Eigentum. Ebenso eine bedeutsame Rolle für die Industrie ist ein effizienter Workflow. Das garantieren IoT-Systeme mit Hilfe von Analyse- und Fehlererkennungsverfahren auch an riesigen Datenmengen. Letztlich ermöglichen IoT-Systeme über ein Endgerät ein schnelles Eingreifen und eine Korrektur an der Produktionsanlage kann erfolgen. Eine ökonomische sowie ökologische Innovation.

Literaturverzeichnis

- [1] URL: https://de.wikipedia.org/wiki/Internet_der_Dinge [15.01.2020]
- [2] URL: <https://industrie-wegweiser.de/von-industrie-1-0-bis-4-0-industrie-im-wandel-der-zeit>
[Stand: 20.01.2020]
- [3] McKinsey & Company: Developing the future of manufacturing (31.03.2017), Online im WWW unter URL: https://www.mckinsey.de/~ /media/McKinsey/Locations/Europe%20and%20Middle%20East/Deutschland/News/Presse/2017/2017-03-31/dcc_brochure_may_2017.ashx [Stand: 03.01.2020]
- [4] URL: <https://www.stadler-schaaf.de> [Stand: 15.11.2019]
- [5] URL: <https://www.se.com/de/de/work/campaign/green-box.jsp> [Stand: 13.01.2020]
- [6] URL: https://www.wago.com/de/download/public/AU-NA-DE-DE-FP-180827_001%2BloT-Box%2BBrochure_web.pdf/AU-NA-DE-DE-FP-180827_001%2BloT-Box%2BBrochure_web.pdf [Stand: 13.01.2020]
- [7] URL: <https://assets.new.siemens.com/siemens/assets/api/uuid:557d6f2fa7641f3a67d4ecf963bf6cc6d296ffd6/width:3840/quality:high/version:1545310899/asiemens-mindsphere.jpg> [Stand: 11.01.2020]
- [8] URL: https://www.bmel.de/DE/Ernaehrung/SichereLebensmittel/Hygiene/_Texte/LebensmittelhygienelmHandel.html [Stand: 13.01.2020]
- [9] URL: <https://de.wikipedia.org/wiki/Node-RED> [Stand: 13.01.2020]
- [10] URL: <https://nodered.org> [Stand: 13.01.2020]
- [11] URL: <https://online.wonderware.eu/mes/oe> [Stand: 21.01.2020]
- [12] URL: <https://de.wikipedia.org/wiki/Schutzart> [Stand: 22.01.2020]
- [13] URL: <https://www.wago.com> [Stand: 21.01.2020]
- [14] URL: <https://www.kobel.info/files/kobel/bilder/produkte/sps/s7-1500.jpg> [Stand: 22.01.2020]
- [15] URL: <https://documentation.mindsphere.io/resources/pdf/asset-manager-en.pdf>
[Stand: 22.01.2020]

- [16] URL: <https://support.industry.siemens.com/cs/document/109756878/-mindconnectfb-library-for-simatic-s7-1500-and-mindsphere-v3-0> [Stand: 20.11.2019]

Eidesstattliche Erklärung



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Erklärung zur selbstständigen Bearbeitung einer Abschlussarbeit

Gemäß der Allgemeinen Prüfungs- und Studienordnung ist zusammen mit der Abschlussarbeit eine schriftliche Erklärung abzugeben, in der der Studierende bestätigt, dass die Abschlussarbeit „- bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit [(§ 18 Abs. 1 APSO-TI-BM bzw. § 21 Abs.1 APSO-INGI)] – ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt wurden. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich zu machen.“

Quelle: § 16 Abs. 5 APSO-TI-BM bzw. § 15 Abs. 6 APSO-INGI

Dieses Blatt, mit der folgenden Erklärung, ist nach Fertigstellung der Abschlussarbeit durch den Studierenden auszufüllen und jeweils mit der Originalunterschrift als letztes Blatt in das Prüfungsexemplar der Abschlussarbeit einzubinden.

Eine unrichtig abgegebene Erklärung kann –auch nachträglich– zur Ungültigkeit des Studienabschlusses führen.

Erklärung zur selbstständigen Bearbeitung der Arbeit

Hiermit versichere ich,

Name: Perkovic

Vorname: Mike

dass ich die vorliegende Bachelorarbeit bzw. bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit – mit dem Thema:

*Entwicklung einer Demoanwendung mit Visualisierungen auf Basis
von ausgewählten und bewerteten Industrial-IoT-Systemen*

ohne fremde Hilfe selbstständig und nur die angegebenen Quellen und Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

Hamburg

Ort

31.01.2020

Datum

Unterschrift im Original