

IIoT-Projekt

im Studiengang Elektrische Systeme

Edge-Gateway zur Anbindung industrieller Prozessdaten an eine IoT-Plattform

Vorgelegt von

Chin-I Feng

Matrikel-Nummer: 300545

Durchgeführt an der

Hochschule Konstanz

Technik, Wirtschaft und Gestaltung

Betreuung:

Prof. Dr. Alexander Krupp

Prof. Dr. Boris Böck

Konstanz, 10. Juli 2025

Kurzfassung

In diesem Projekt wurde ein flexibles Industrial Internet of Things (IIoT)-System zur automatisierten Erfassung und Übertragung von Prozessdaten einer Taktstraße im Automatisierungslabor der HTWG realisiert. Kernkomponenten des Systems sind Node-RED und Mosquitto auf dem Edge-Gateway, über die die Prozessdaten gesammelt, vorverarbeitet und mithilfe des MQTT-Protokolls an die zentrale IoT-Plattform ThingsBoard gesendet werden. Auf ThingsBoard werden die Daten anschließend gespeichert, visualisiert und für weiterführende Analysen bereitgestellt.

Inhaltsverzeichnis

Kurzfassung	II
Abbildungsverzeichnis.....	IV
Tabellenverzeichnis.....	IV
Abkürzungsverzeichnis	V
1 Einleitung	1
2 Design	3
2.1 Systemarchitektur	3
2.2 Softwaremodule	3
2.2.1 Edge-Gateway	5
2.2.2 IoT-Plattform	5
2.3 Datenfluss	6
3 Testergebnisse.....	10
4 Entwicklungsaufwand.....	11
5 Fazit und Ausblick	13
Literaturverzeichnis	X
Eidesstattliche Erklärung.....	XI
Anhang.....	XII

Abbildungsverzeichnis

Abb. 1: Architektur der Datenübertragung von der Taktstraße zur IoT-Plattform	1
Abb. 2: Systemkontextdiagramm der IoT-Datenpipeline	3
Abb. 3: Deployment-Architektur der containerisierten IoT-Datenpipeline	4
Abb. 4: Beispiel für ein vom Gateway geliefertes Prozessdatenpaket im JSON- Format	6
Abb. 5: JavaScript-Code in Node-RED zur Erstellung des flachen JSON- Formats.....	7
Abb. 6: Beispiel einer flachen JSON-Struktur der Prozessdaten nach dem Data Mapping im Edge-Gateway.....	8
Abb. 7: Visualisierung der Prozessdaten und Durchlaufzeiten auf der IoT- Plattform ThingsBoard	8
Abb. 8: Visualisierung der Systemstatusdaten des Edge-Gateways auf der IoT- Plattform ThingsBoard	9

Tabellenverzeichnis

Tab. 1: Arbeitsaufwand im Projektverlauf.....	11
---	----

Abkürzungsverzeichnis

IoT	Internet of Things
IIoT	Industrial Internet of Things
JSON	JavaScript Object Notation
MQTT	Message Queuing Telemetry Transport
OPC UA	Open Platform Communications Unified Architecture
RPC	Remote Procedure Call
SPS	Speicherprogrammierbare Steuerung

1 Einleitung

In der heutigen industriellen Fertigung werden an Taktstraßen zahlreiche wichtige Prozessdaten erfasst, die für die Überwachung und Optimierung der Produktionsabläufe unerlässlich sind [1]. Im Automatisierungslabor der Hochschule Konstanz können diese Daten jedoch aktuell nur lokal im Labor gesichert und ausgewertet werden. Ein Zugriff von außerhalb, etwa zur Fernüberwachung oder zentralen Analyse, ist nicht möglich.

Das Hauptziel dieses Projekts ist es daher, die Prozessdaten einer Taktstraße im Automatisierungslabor aus der lokalen Umgebung heraus verfügbar zu machen und für weiterführende Analysen zu nutzen. Durch die Entwicklung und Implementierung eines Edge-Gateways sollen die Prozessdaten der Taktstraße an eine zentrale Internet of Things (IoT)-Plattform übertragen werden (siehe Abb. 1).

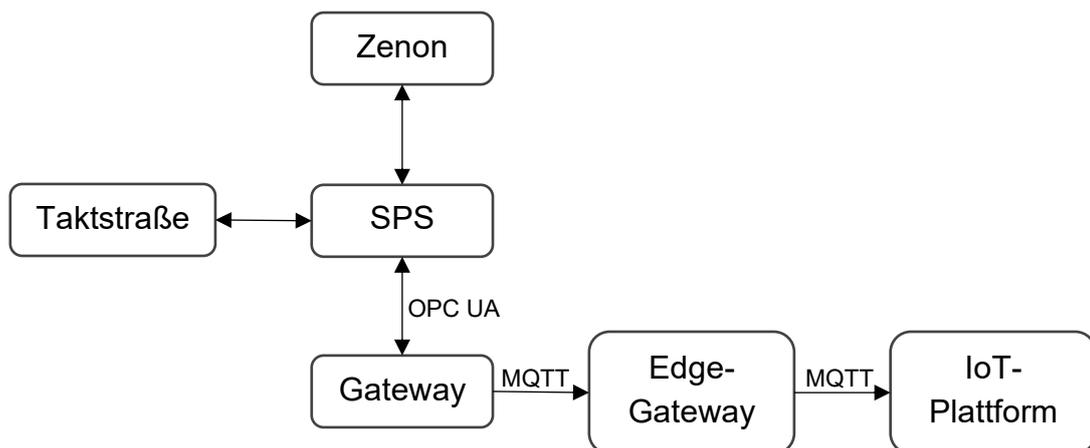


Abb. 1: Architektur der Datenübertragung von der Taktstraße zur IoT-Plattform

Die Prozessdaten werden zunächst direkt an der Taktstraße erfasst und durch die Speicherprogrammierbare Steuerung (SPS) aggregiert. Für die Weiterleitung der Prozessdaten an die höheren Ebenen wird ein Gateway verwendet, das die Kommunikation über das industrielle Open Platform Communications Unified Architecture (OPC UA)-Protokoll mit der SPS ermöglicht. Das Gateway überträgt die gesammelten Daten anschließend mittels Message Queuing Telemetry Transport (MQTT)-Protokoll im JavaScript Object Notation (JSON)-Format an das Edge-Gateway. Das Edge-Gateway dient als zentrale Schnittstelle, um die empfangenen Prozessdaten vorzuerarbeiten, zu filtern und über das MQTT-Protokoll an die IoT-Plattform weiterzuleiten. Die IoT-Plattform übernimmt die zentrale Speicherung,

Visualisierung und Auswertung der Prozessdaten und ermöglicht den standortunabhängigen Zugriff auf aktuelle und historische Produktionsdaten.

Im folgenden Verlauf dieser Projektdokumentation wird die Entwicklung des Edge-Gateways sowie die Anbindung an die IoT-Plattform und die Übertragung von Prozessdaten einer Taktstraße am Beispiel des Automatisierungslabors der Hochschule Konstanz beschrieben. **Kapitel 2** befasst sich mit dem Software-Design des Projekts. Hier werden die Architektur, die eingesetzten Technologien und die wichtigsten Software-Komponenten detailliert vorgestellt.

In **Kapitel 3** werden die Testergebnisse dokumentiert und ausgewertet. Dabei wird insbesondere darauf eingegangen, inwiefern die Anbindung an die IoT-Plattform erfolgreich realisiert werden konnte und welche Herausforderungen sich im Praxistest ergeben haben. **Kapitel 4** bietet eine Übersicht über den tatsächlichen Entwicklungsaufwand. Der Projektfortschritt wird hier aufgeschlüsselt nach den Kategorien des Projektplans dargestellt. Ein wöchentlicher Stundenreport pro Teammitglied bildet die Grundlage für diese Auswertung. Abschließend werden in **Kapitel 5** mögliche Verbesserungen des Systems diskutiert und das Fazit des Projekts gezogen.

2 Design

In diesem Kapitel wird das Software-Design des Projekts beschrieben. Zunächst wird die Gesamtarchitektur mit den wichtigsten Komponenten wie dem Edge-Gateway und der IoT-Plattform vorgestellt. Anschließend werden die zentralen Softwaremodule, deren Aufgaben sowie die Umsetzung der Kommunikation und die verwendeten Protokolle erläutert.

2.1 Systemarchitektur

Die Systemarchitektur des Projekts ist so ausgelegt, dass das Gateway die Prozessdaten der Taktstraße über das OPC UA-Protokoll abonniert und als autonome Datenquelle im standardisierten JSON-Format über das MQTT-Protokoll publiziert. Das Edge-Gateway empfängt diese Daten vom MQTT-Broker, übernimmt unmittelbar die Vorverarbeitung und bereinigt die empfangenen Nachrichten. Die verarbeiteten Daten werden anschließend erneut als JSON-Payload via MQTT an die IoT-Plattform weitergeleitet. Dort erfolgt die zentrale Speicherung der Datensätze in einer Datenbank, und die Daten werden in Live-Dashboards für die weitere Analyse und Auswertung bereitgestellt (siehe Abb. 2).



Abb. 2: Systemkontextdiagramm der IoT-Datenpipeline

Die Architektur ist modular aufgebaut. Die verwendeten Software-Module sowie die Gründe für die jeweilige Technologieauswahl werden im folgenden Unterkapitel vorgestellt.

2.2 Softwaremodule

Die zentrale Rolle im System übernehmen mehrere Softwaremodule, die jeweils in eigenen Containern innerhalb einer Docker-Umgebung ausgeführt werden. Abbildung 3 zeigt den modularen Aufbau der Software-Komponenten auf dem Edge-Gateway sowie auf dem zentralen PC.

Dabei stellt Docker die Isolation der jeweiligen Docker-Container zu den darunterliegenden Hosts bzw. virtuellen Maschinen sicher. Durch den Einsatz von Docker wird gewährleistet, dass Laufzeitabhängigkeiten und Konfigurationsparameter

strikt gekapselt sind und sich dadurch reproduzierbare Betriebsumgebungen realisieren lassen.

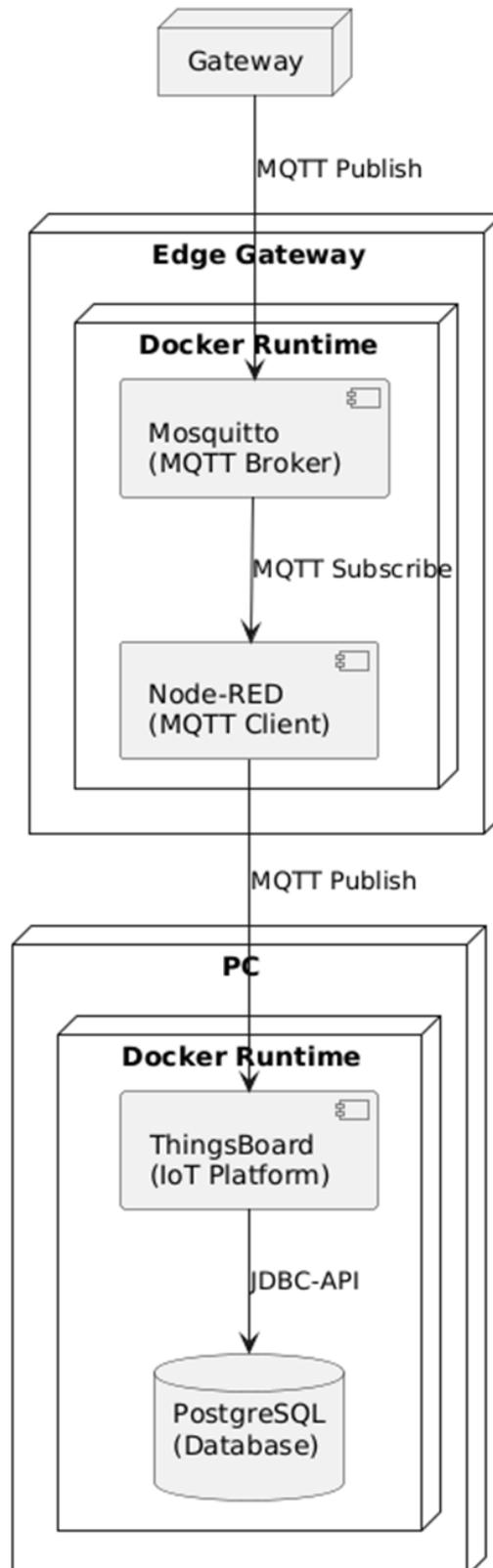


Abb. 3: Deployment-Architektur der containerisierten IoT-Datenpipeline

Auf dem Edge-Gateway werden zwei spezialisierte Container betrieben: Einerseits ein Mosquitto-Broker als lokal verfügbarer MQTT-Server, andererseits eine Node-RED-Instanz, die als flow-basierte Integrations- und Datenverarbeitungsumgebung fungiert. Die IoT-Plattform-Instanz, bestehend aus ThingsBoard und einer PostgreSQL-Datenbank, ist auf einem separaten Host in eigenständigen Containern untergebracht. Im Folgenden werden die Softwaremodule des Edge-Gateways und der IoT-Plattform jeweils im Detail beschrieben.

2.2.1 Edge-Gateway

Für das Edge-Gateway wurden gezielt die Komponenten Mosquitto und Node-RED ausgewählt, da sie in Bezug auf Erweiterbarkeit und Integrationsfähigkeit besonders geeignet sind.

Als leichtgewichtiger, ressourcenschonender MQTT-Broker ist Mosquitto seit Langem der Standard in IoT-Umgebungen. Er unterstützt alle MQTT-QoS-Stufen und lässt sich ohne großen Konfigurationsaufwand betreiben [3]. Aufgrund der minimalen Betriebsvoraussetzungen eignet sich Mosquitto insbesondere für ressourcenlimitierte Edge-Geräte und garantiert eine wartungsarme Bereitstellung.

Node-RED implementiert ein visuelles Flow-Paradigma, das in der Forschung als Beispiel für modellgetriebene Entwicklung in Systemen herangezogen wird. Die deklarative Flusslogik erlaubt eine beschleunigte Prototypenentwicklung, während eine umfangreiche Community-Bibliothek von Contrib-Nodes eine modulare, wiederverwendbare Architektur sicherstellt [4].

Durch die Kombination dieser beiden Softwarelösungen kann das Edge-Gateway flexibel und wartungsarm an die Anforderungen der Prozessdatenerfassung und Datenübertragung angepasst werden.

2.2.2 IoT-Plattform

Für die zentrale IoT-Plattform wurde ThingsBoard ausgewählt. ThingsBoard ist eine kostenlose, Open-Source-IoT-Plattform, die für die Verwaltung, Speicherung, Visualisierung und Analyse von IoT-Daten entwickelt wurde. Sie bietet viele wichtige Funktionen wie die Unterstützung des MQTT-Protokolls, individuell gestaltbare Dashboards und eine integrierte PostgreSQL-Datenbank zur langfristigen Datenspeicherung [5].

Im Gegensatz zu kommerziellen Lösungen wie Microsoft Azure, Google Cloud oder AWS punktet ThingsBoard vor allem durch seine Kosteneffizienz und einfache lokale Installation, insbesondere mithilfe von Docker-Containern. Damit ist die Plattform gut für Hochschulprojekte mit begrenztem Budget geeignet.

2.3 Datenfluss

Die vom Gateway bereitgestellten Prozessdaten werden in strukturierter Form im JSON-Format übertragen. Jeder Datenpunkt enthält neben dem aktuellen Wert auch einen Zeitstempel, der angibt, wann die Information am Gateway erfasst wurde. Ein Beispiel für den Aufbau der gesendeten Daten ist in Abbildung 4 dargestellt.

```
{
  "Subscriptions": {
    "get_sensor_conveyor_input_in": {
      "value": true,
      "gateway_time": "2025-07-03 17:40:46"
    },
    "get_sensor_conveyor_output": {
      "value": true,
      "gateway_time": "2025-07-03 17:41:01"
    },
    "get_drilling": {
      "value": false,
      "gateway_time": "2025-07-03 17:40:51"
    },
    "get_milling": {
      "value": false,
      "gateway_time": "2025-07-03 17:40:57"
    }
  }
}
```

Abb. 4: Beispiel für ein vom Gateway geliefertes Prozessdatenpaket im JSON-Format

Das JSON-Objekt umfasst verschiedene Schlüssel, die jeweils den Zustand unterschiedlicher Sensoren und Maschinenkomponenten repräsentieren. Jeder dieser Einträge besteht aus einem „value“-Feld, das den aktuellen Status (z.B. true/false) angibt, sowie einem „gateway_time“-Feld, das den genauen Erfassungszeitpunkt dokumentiert.

Diese strukturierte Datenübertragung ermöglicht eine eindeutige und zeitgenaue Nachverfolgung der Zustände entlang der gesamten Taktstraße und bildet die Grundlage für die weitere Verarbeitung im Edge-Gateway.

Im Edge-Gateway übernimmt Node-RED die Aufgabe, die verschachtelten JSON-Strukturen der eingehenden Prozessdaten in ein flaches, übersichtliches Format zu

überführen. Da die IoT-Plattform ThingsBoard mit flachen JSON-Strukturen arbeitet, können die einzelnen Prozesswerte so direkt und übersichtlich dargestellt und ausgewertet werden. Die Umsetzung des Datenmappings im Edge-Gateway erfolgt in Node-RED mithilfe einer eigenen JavaScript-Funktion (siehe Abb. 5).

```
const subs = msg.payload.Subscriptions || {};  
  
const inputTime = subs.get_sensor_conveyor_input_in?.gateway_time;  
const outputTime = subs.get_sensor_conveyor_output?.gateway_time;  
  
let deltaSeconds = null;  
if (inputTime && outputTime) {  
    const inputDate = new Date(inputTime.replace(' ', 'T') + 'Z');  
    const outputDate = new Date(outputTime.replace(' ', 'T') + 'Z');  
  
    // Differenz berechnen  
    deltaSeconds = (outputDate.getTime() - inputDate.getTime()) / 1000;  
    // Wenn negativ, auf null setzen  
    if (deltaSeconds < 0) {  
        deltaSeconds = null;  
    }  
}  
  
// Flaches JSON bauen  
const flat = {};  
for (const key in subs) {  
    if (subs.hasOwnProperty(key)) {  
        const obj = subs[key];  
        flat[`_${key}_value`] = obj.value;  
        flat[`_${key}_gateway_time`] = obj.gateway_time;  
    }  
}  
flat['conveyor_delta_time'] = deltaSeconds; // Differenz in Sekunden  
  
msg.payload = flat;  
return msg;
```

Abb. 5: JavaScript-Code in Node-RED zur Erstellung des flachen JSON-Formats

Dabei werden die verschachtelten JSON-Strukturen eingelesen, alle Prozesswerte und Zeitstempel extrahiert und in ein flaches JSON-Format umgewandelt. Zusätzlich berechnet die Funktion automatisch die Differenz zwischen Ein- und Ausgangszeitpunkt, um die Prozessdauer zu bestimmen.

Nach dem Data Mapping im Edge-Gateway durch Node-RED liegen die Prozessdaten in einer flachen JSON-Struktur vor. Dadurch können die einzelnen Prozesswerte und zugehörigen Zeitstempel direkt und übersichtlich in der IoT-Plattform angezeigt und ausgewertet werden. Zusätzlich enthält die Struktur das Feld „conveyor_delta_time“, das die berechnete Durchlaufzeit des Werkstücks angibt (siehe Abb. 6).

```

{
  "get_sensor_conveyor_input_in_value": true,
  "get_sensor_conveyor_input_in_gateway_time": "2025-07-03 17:40:46",
  "get_sensor_conveyor_output_value": true,
  "get_sensor_conveyor_output_gateway_time": "2025-07-03 17:41:01",
  "get_drilling_value": false,
  "get_drilling_gateway_time": "2025-07-03 17:40:51",
  "get_milling_value": false,
  "get_milling_gateway_time": "2025-07-03 17:40:57",
  "conveyor_delta_time": 15
}

```

Abb. 6: Beispiel einer flachen JSON-Struktur der Prozessdaten nach dem Data Mapping im Edge-Gateway

Nach der erfolgreichen Übertragung und Verarbeitung der Daten werden die Prozesswerte auf der IoT-Plattform ThingsBoard übersichtlich visualisiert. In der abgebildeten Zeitserientabelle sind die einzelnen Zustände und Zeitstempel für Werkstück-Eingang, Werkstück-Ausgang sowie die Bearbeitungsstationen (Bohren, Fräsbearbeitung) aufgeführt. Besonders anschaulich ist die direkte Anzeige der berechneten Durchlaufzeit in Sekunden, wie im gezeigten Beispiel mit einem Wert von 15 Sekunden für den betrachteten Prozess (siehe Abb. 7).

Werkstück_Eingang	Werkstück_Eingangszeit	Werkstück_Ausgang	Werkstück_Ausgangszeit	Bohrung	Fräsbearbeitung	Durchlaufzeit
true	2025-07-03 17:40:46	false	2025-07-03 17:41:01	false	false	15
true	2025-07-03 17:40:46	true	2025-07-03 17:40:10	false	false	
true	2025-07-03 17:40:46	true	2025-07-03 17:40:10	true	false	
true	2025-07-03 17:40:46	true	2025-07-03 17:40:10	false	false	
true	2025-07-03 17:40:46	true	2025-07-03 17:40:10	false	true	
true	2025-07-03 17:40:46	true	2025-07-03 17:40:10	false	false	
false	2025-07-03 17:40:46	true	2025-07-03 17:40:10	false	false	

Abb. 7: Visualisierung der Prozessdaten und Durchlaufzeiten auf der IoT-Plattform ThingsBoard

Neben der Auswertung der eigentlichen Prozessdaten besteht auch die Möglichkeit, Systeminformationen des Edge-Gateways zu überwachen. Dazu kann in Node-RED eine zusätzliche JavaScript-Funktion implementiert werden, die regelmäßig Systemstatusdaten wie CPU-Temperatur und RAM-Auslastung ausliest und diese Werte ebenfalls über MQTT an ThingsBoard überträgt.

Auf der IoT-Plattform lassen sich diese Kennzahlen dann übersichtlich in Dashboards visualisieren, sodass der Gesundheitszustand des Edge-Gateways in Echtzeit überwacht werden kann (siehe Abb. 8).

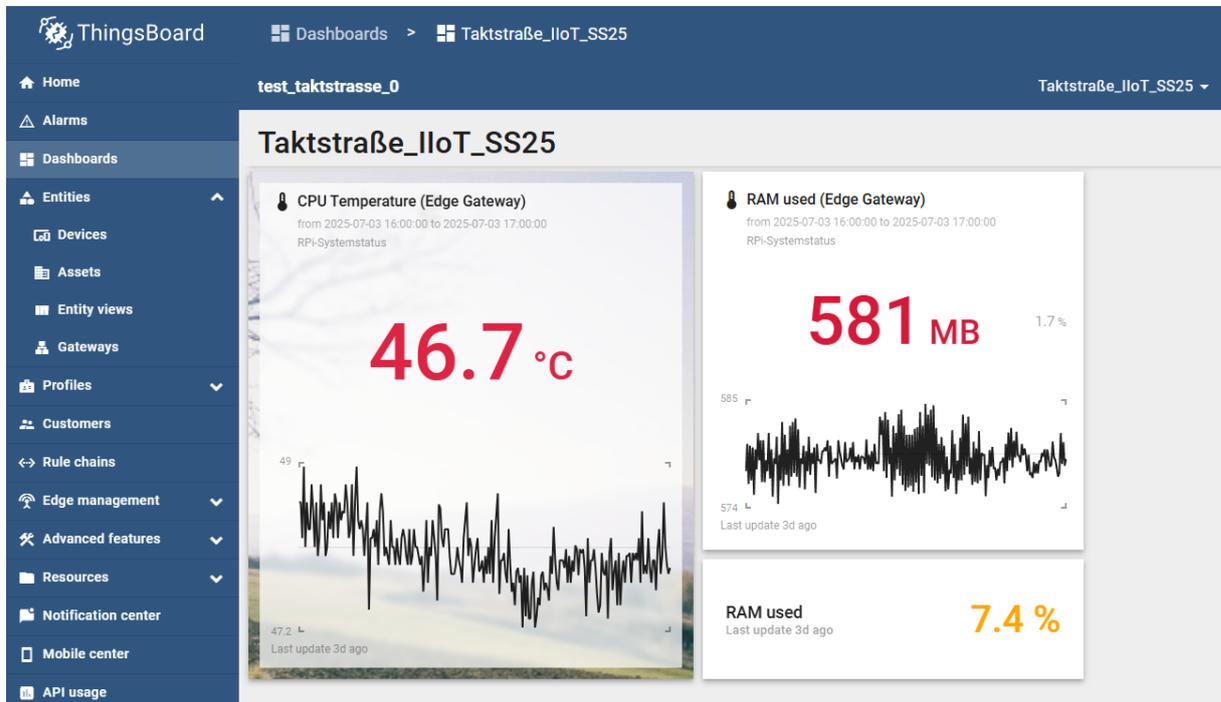


Abb. 8: Visualisierung der Systemstatusdaten des Edge-Gateways auf der IoT-Plattform ThingsBoard

Dadurch ist es möglich, neben der Produktionsüberwachung auch präventiv auf drohende Systemprobleme zu reagieren und die Betriebssicherheit zu erhöhen.

3 Testergebnisse

Im Rahmen der Tests wurde das Gesamtsystem im Automatisierungslabor der HTWG unter realen Bedingungen in Betrieb genommen. Zunächst zeigte sich, dass das Gateway die Prozessdaten zuverlässig im JSON-Format per MQTT publizierte und vom Edge-Gateway problemlos empfangen werden konnte. Die Weiterverarbeitung der Daten in Node-RED funktionierte fehlerfrei. Sämtliche Sensor- und Maschinenzustände wurden in das geforderte flache JSON-Format transformiert und mit den zugehörigen Zeitstempeln versehen. Besonders positiv fiel auf, dass auch die berechnete Durchlaufzeit der Werkstücke korrekt bestimmt und zusammen mit den übrigen Werten an die IoT-Plattform ThingsBoard übermittelt wurde.

Die Visualisierung der Prozessdaten in ThingsBoard verlief wie erwartet. Die Dashboards stellten alle Prozesswerte, Zeitstempel und die ermittelte Durchlaufzeit übersichtlich und in Echtzeit dar, wodurch eine transparente Analyse der Produktionsabläufe möglich war. Auch nach längeren Testläufen mit kontinuierlicher Datenübertragung zeigten sich keine Datenverluste oder Verbindungsabbrüche. Das System arbeitete stabil und zuverlässig.

Darüber hinaus wurde die Übertragung von Systemstatusdaten des Edge-Gateways, wie CPU-Temperatur und RAM-Auslastung, erfolgreich getestet. Mithilfe einer in Node-RED implementierten JavaScript-Funktion konnten diese Werte ebenfalls per MQTT an ThingsBoard übertragen und dort in separaten Dashboards visualisiert werden. So ließ sich der Gesundheitszustand des Edge-Gateways jederzeit überwachen.

Zusammenfassend belegen die Testergebnisse, dass die komplette Datenpipeline vom Gateway über das Edge-Gateway bis zur IoT-Plattform zuverlässig und effizient arbeitet. Die Anforderungen an eine sichere Datenerfassung, stabile Übertragung und transparente Visualisierung der Produktions- und Systemdaten wurden vollumfänglich erfüllt.

4 Entwicklungsaufwand

Die Übersicht in Tabelle 1 zeigt den tatsächlichen Arbeitsaufwand, der im Verlauf des Projekts für die einzelnen Arbeitspakete angefallen ist. Für jede Projektphase wurden die jeweils geleisteten Stunden dokumentiert und den relevanten Aufgabenbereichen zugeordnet. Der Zeitaufwand wurde dabei in Kalenderwochen (KW) und geschätzten Arbeitsstunden pro Aufgabe festgehalten.

Tab. 1: Arbeitsaufwand im Projektverlauf

Person	Arbeitspaket	Beschreibung	Zeitaufwand
C.Feng	Projektplanung	Erstellung des Projektplans, Aufteilung der Arbeitspakete, Zeitplanung	KW 18 (~6 h)
C.Feng	Softwarearchitektur	Anforderungsanalyse, Festlegen der Systemarchitektur und Kommunikationswege	KW 19/20 (~12 h)
C.Feng	Entwicklung Edge-Gateway	Implementierung und Test von Node-RED und Mosquitto, Einrichtung der Docker-Umgebung	KW 20/21 (~15 h)
C.Feng	Anbindung IoT-Plattform	Installation und Konfiguration von ThingsBoard, MQTT-Anbindung und Datenübertragung getestet	KW 22/23 (~15 h)
C.Feng	Systemtests und Validierung	Funktionstests, Auswertung der Datenübertragung	KW 24 (~6 h)
C.Feng	Datentransformation und Mapping	Umsetzung des Datenmappings in Node-RED (JSON-Transformation, Zeitberechnung)	KW 25 (~6 h)
C.Feng	Systemtests und Validierung	Funktionstests, Auswertung der Datenübertragung und Visualisierung	KW 26 (~9 h)

C.Feng	Dokumentation	Bericht schreiben, Diagramme erstellen, Ergebnisse zusammenfassen	KW 27/28 (~18 h)
Summe			~87 h

Wie die Tabelle zeigt, entfiel der größte Teil des Aufwands auf die Entwicklung und Implementierung der Softwaremodule für das Edge-Gateway und die IoT-Plattform. Auch die Planung, die Einrichtung der Systemarchitektur, die Durchführung von Systemtests sowie die abschließende Dokumentation machten einen wesentlichen Teil des Projektumfangs aus. Insgesamt wurden für das Projekt rund 87 Arbeitsstunden aufgewendet.

5 Fazit und Ausblick

Das Projekt hat erfolgreich gezeigt, wie Prozessdaten einer Taktstraße im Automatisierungslabor der Hochschule Konstanz mithilfe eines Edge-Gateways zuverlässig erfasst, vorverarbeitet und an eine IoT-Plattform übertragen werden können. Die entwickelte Systemarchitektur, basierend auf Docker-Containern, Node-RED, Mosquitto, ThingsBoard und PostgreSQL, ermöglichte eine flexible, erweiterbare und transparente Datenpipeline. Durch die Visualisierung der Prozess- und Systemdaten in ThingsBoard war eine übersichtliche und praxisnahe Analyse möglich. Alle definierten Kernanforderungen an Datenübertragung, Datensicherheit, Vorverarbeitung und Visualisierung wurden erfüllt. Die Systemtests bestätigten eine hohe Zuverlässigkeit und Stabilität des gesamten Aufbaus.

Allerdings sind Verbesserungen und Erweiterungen für das implementierte System möglich und sinnvoll. Zum einen könnte die in Kapitel 2.2 eingesetzte Datenbank zur Speicherung der Prozessdaten der Taktstraße weiter optimiert werden. Die aktuell genutzte PostgreSQL-Datenbank von ThingsBoard eignet sich gut für relationale Daten, ist jedoch für die effiziente Verarbeitung und Speicherung großer Mengen zeitbasierter Prozessdaten weniger optimal. Der Einsatz einer spezialisierten Zeitreihendatenbank wie InfluxDB wäre hier vorteilhaft, da sie für das Management und die Analyse von Zeitreihendaten entwickelt wurde und eine schnellere Abfrage sowie bessere Skalierbarkeit bietet. Dadurch könnten insbesondere umfangreiche historische Auswertungen und Analysen der Prozessdaten deutlich effizienter durchgeführt werden.

Des Weiteren könnten auch die Remote Procedure Call (RPC)-Funktionen von ThingsBoard für zukünftige Erweiterungen genutzt werden. Damit wäre es möglich, von der IoT-Plattform aus gezielt Steuerungsbefehle an die Taktstraße zu senden, etwa um bestimmte Prozesse zu starten, zurückzusetzen oder im Fehlerfall automatisch zu stoppen. Dies würde nicht nur den Bedienkomfort erhöhen, sondern auch die Integration fortschrittlicher Automatisierungs- und Fernwartungsfunktionen ermöglichen. Insgesamt ließe sich so die Flexibilität und Zukunftsfähigkeit des gesamten Systems weiter steigern.

Literaturverzeichnis

- [1] N. Boysen, P. Schulze, und A. Scholl, „Assembly line balancing: What happened in the last fifteen years?“, *European Journal of Operational Research*, Bd. 301, Nr. 3, S. 797–814, 2022.
- [2] B. B. Rad, H. J. Bhatti, und M. Ahmadi, „An introduction to docker and analysis of its performance“, *International Journal of Computer Science and Network Security (IJCSNS)*, Bd. 17, Nr. 3, S. 228–235, 2017.
- [3] Eclipse Mosquitto. [Online]. Verfügbar: <https://mosquitto.org> (Zugriff am 06. Juli 2025).
- [4] K. Ferencz und J. Domokos, „Using Node-RED platform in an industrial environment“, in *XXXV. Jubileumi Kandó Konferencia*, Budapest, 2019, S. 52–63.
- [5] A. A. Ismail, H. S. Hamza, und A. M. Kotb, „Performance evaluation of open source IoT platforms“, in *2018 IEEE Global Conference on Internet of Things (GCIoT)*, Alexandria, 2018, S.1-5.

Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich dieses Projekt selbstständig und nur unter Benutzung der angegebenen Literatur und Hilfsmittel angefertigt habe und alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, als solche gekennzeichnet sind. Die Arbeit wurde noch nicht bei einer Prüfungsbehörde eingereicht oder veröffentlicht. Zusätzlich erkläre ich, dass ich Generative KI-Technologien ausschließlich zur Verbesserung des Codes und der Rechtschreibung eingesetzt habe. Dabei wurden keine sensiblen Daten eingegeben oder ganze Texte generiert. Der Einsatz dieser Technologien beschränkte sich strikt auf technische Unterstützung ohne Einfluss auf die inhaltliche Eigenleistung.

Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben wird.

10. Juli 2025

Datum

Chin-I Feng

Unterschrift

Anhang

Die folgenden fünf Anleitungen befinden sich im Abgabeordner im Unterverzeichnis „**Anleitungen**“ als separate PDF-Dateien. Alle Anleitungen sind auch im [GitLab](#) verfügbar (siehe jeweiliger Link):

- **Zugriff_auf_TB_Datenbank.pdf:**
Diese Anleitung beschreibt, wie der Zugriff auf die ThingsBoard-Datenbank eingerichtet und genutzt werden kann.
[Zur Anleitung im GitLab-Wiki](#)
- **Neuinstallation_Betriebssystem.pdf:**
Schritt-für-Schritt-Anleitung zur Neuinstallation des Betriebssystems auf dem Edge-Gateway.
[Zur Anleitung im GitLab-Wiki](#)
- **Verbindung_EdgeGateway_TB.pdf:**
Beschreibung der Schritte, um das Edge-Gateway mit ThingsBoard zu verbinden.
[Zur Anleitung im GitLab-Wiki](#)
- **IP_Adresse_konfigurieren_A231.pdf:**
Anleitung zur Konfiguration der IP-Adresse für das Edge-Gateway im Labor A231.
[Zur Anleitung im GitLab-Wiki](#)
- **EdgeGateway_Setup_Anleitung.pdf:**
Umfassende Anleitung zur grundlegenden Einrichtung des Edge-Gateways.
[Zur Anleitung im GitLab-Wiki](#)

Zusätzlich enthält der Ordner „Anleitungen“ die Datei **EdgeGateway_Login.pdf** mit den Zugangsdaten (IP-Adresse, Benutzername, Passwort) für das Edge-Gateway im Labor A231.

Der Zugriff auf das GitLab-Wiki ist nur im HTWG-Campusnetz oder mit aktiviertem HTWG-VPN gewährleistet.

Letzter Zugriff auf alle verlinkten Seiten: 07.07.2025.