

# **Vorstellung der Abschlussarbeit**

Johannes Müller  
2022

# Wer bin ich?

- Johannes Müller
- 24 Jahre alt
- Student an der Hochschule Anhalt
- Studiengang: Elektro- und Informationstechnik



**HOCHSCHULE  
ANHALT** University  
of Applied Sciences

**FB6 - EMW**

Fachbereich Elektrotechnik, Maschinenbau  
und Wirtschaftsingenieurwesen  
Campus Köthen

# Worum geht es?

- Abschlussarbeit für das Masterstudium
- Thema: Energiemonitoring zur Prozessanalyse in industriellen Anlagen am Beispiel von Kernschießmaschinen
- Abgabe: 07.12.2022

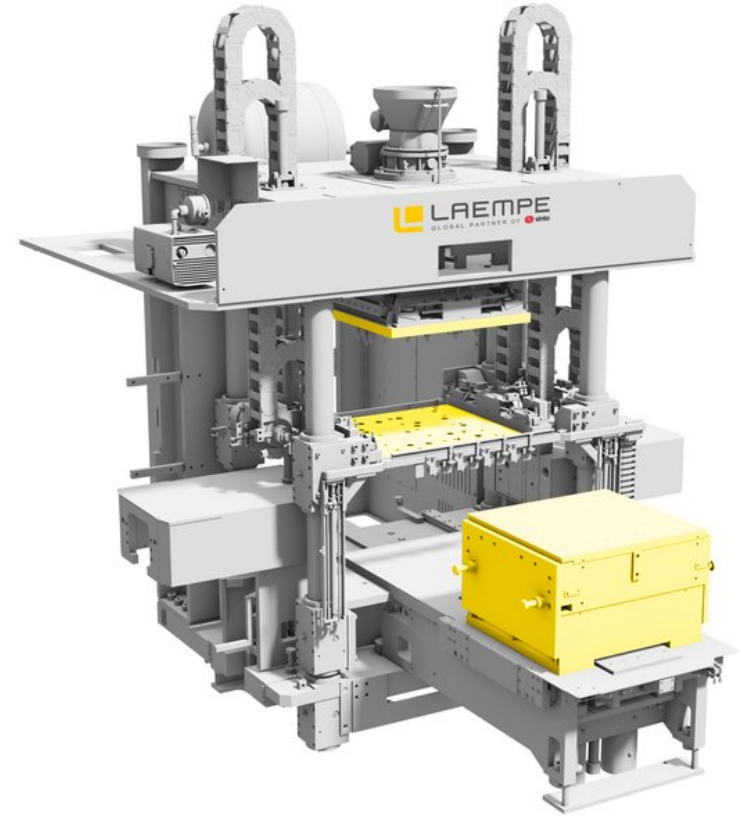
# Woher kommt das Thema?

- Werkstudent bei Laempe Mössner Sinto GmbH
- seit 2021
- Abteilung: Elektroplanung
- Betreuer: Thomas Mengewein



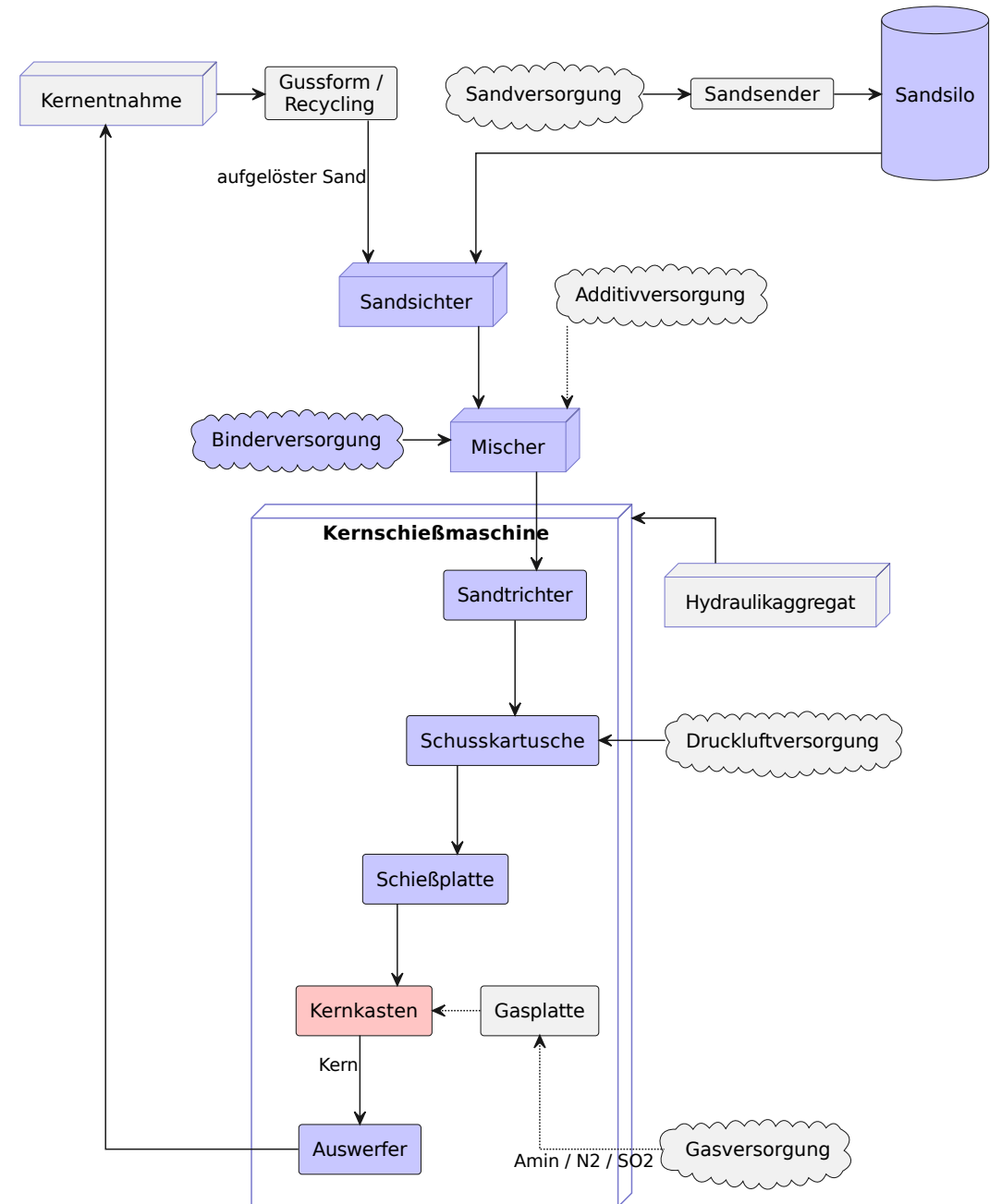


# Was sind Kernschießmaschinen?



# Prozessablauf

- Sandversorgung
- Sandsichter
- Binderversorgung
- Mischer
- Kernschießmaschine



# Warum Energieüberwachung?

- Energieverbrauch ist ein wichtiger Kostenfaktor
- Vorzeitige Fehlererkennung
- Prozessoptimierung

# Bestandteile der Energieüberwachung

- Erfassung der Einspeisung
- Erfassung der Steuerströme
- Datenverarbeitung
- Datenspeicherung
- Visualisierung

# Bestandteile der Prozessanalyse

- Erfassung des Anlagenzustandes
- Erfassung der Prozessparameter
- Korellation der Zustände mit dem Energieverbrauch

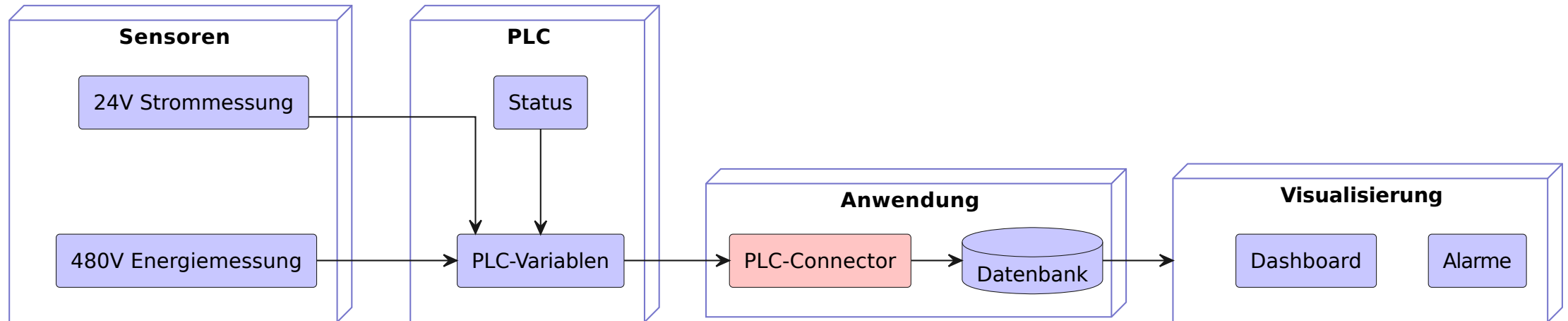
# Besonderheiten

- Unterstützung von mehreren Steuerungstypen und Kommunikationsbussen

Hersteller	Steuerungstyp	Dezentrale Peripherie	Feldbus
Siemens	SIMATIC S7-1500	SIMATIC ET 200SP	Profinet
Allen-Bradley	Compact Logix	Point IO	EtherNet/IP
Mitsubishi	MELSEC iQ-R		CC-Link IE Field EtherNet/IP

# Implementierung

## Datenfluss





# 24V Strommessung

- IFM DF2102
- modulare Elektronische Sicherungen
- IO-Link
- bis zu 16 Kanäle
- 1-10A pro Kanal (max. 40A)



# 480V Strommessung

- Siemens AI-Energy-Meter
- 3-phasig
- 3x 150/5A (50/5A) Stromwandler



# Software

- Pythonprojekt: PLC-Connector
- Lesen von Daten aus der Steuerung
- Lesen von Messwerten aus der Peripherie
- Synchronisation der Datenquellen
- Aufarbeitung durch Filterung, Aggregation und Selektion

# Einrichtung des PLC-Connectors

- Installation
- Konfiguration der Datenquellen (Inputmodule)
- Konfiguration der Datenverarbeitung (Middleware)
- Konfiguration der Datenspeicherung (Outputmodule)

# Installation mit Docker

Voraussetzungen:

- Linux-Umgebung
- Git
- Docker
- docker-compose

# Installation mit Docker

```
git clone https://gogs.justprojects.de/Master/smart-energy-monitor.git  
cd smart-energy-monitor/box-pc  
sudo docker-compose build  
sudo docker-compose up -d
```

# Installation ohne Docker

Voraussetzungen:

- Git
- Python 3.x
- Python-Paketmanager `pip`
- optional Admin-Rechte (für den `snmp7`-Server auf Port 102)
- `influxdb`-Server

# Installation ohne Docker

```
git clone https://gogs.justprojects.de/Master/smart-energy-monitor.git  
cd smart-energy-monitor/box-pc/plc-connector  
python -m pip install -r requirements.txt  
python main.py -c config.yml
```



# Konfiguration der Datenquellen

Inputs:

```
# Definition des ersten Moduls
#   ClassName: Name der Python-Klasse des Moduls
#   path.to.module: relativer Importpfad des Input-Moduls
- ClassName: path.to.module
  # Ob das Modul geladen werden soll
  # Default: True
  enabled: True
  # Parameter, die an den Konstruktor des Moduls übergeben werden
  param1: "value 1"
  param2: "value 2"

# Definition weiterer Module
- ClassName2: path.to.module2
  enabled: True
```

# Konfiguration der Datenquellen

- Beispiel an einer Siemens S7-1500 CPU
- als aktiver Partner

Inputs:

```
- SiemensCPU: siemens.snap7_connect  
  enabled: True  
  host: "192.168.0.10"
```

# Konfiguration der Datenquellen

- Beispiel an einer Siemens ET200 SP CPU
- als passiver Partner

Inputs:

```
- SiemensServer: siemens.snap7_server  
  enabled: True  
  port: 102
```

# Konfiguration der Datenverarbeitung

## Middlewares:

- # Definition des ersten Moduls*
- # ClassName: Name der Python-Klasse des Moduls*
- # path.to.module: relativer Importpfad des Middleware-Moduls*
- ClassName: `path.to.module`  
enabled: False
- TimeCorrelation: `time_correlation`  
*# Zwischenmodule können geschachtelt werden, so dass*  
*# sie die Ergebnisse des überliegenden Moduls weiterverarbeiten*  
submodules:
  - PrintStats: `print_stats`  
*# Standardmäßig werden die Ergebnisse von Middleware-Modulen*  
*# ohne Untermodulen für die Ausgabe gesammelt und dedupliziert*

# Konfiguration der Datenverarbeitung

- Beispiel für die Filterung

Middleware:

- TimeCorrelation: `time_correlation`  
submodules:
  - MatchAny: `filters`  
series: `plant`  
enable\_output: `False`  
table\_move\_up: `1`
  - ComplexFilter: `filters`  
predicate: `"measurement_480v and measurement_24v and avg(measurement_24v.current) > 0.6"`  
submodules:
    - ComplexSelector: `selectors`  
selector: `"avg(measurement_480v.current)"`  
enable\_output: `True`

- Beispiel für die Aggregation

Middleware:

- TimeCorrelation: `time_correlation`
- Aggregate: `aggregators`
  - series: `24v`
  - timespan: `0.1`
  - avg:
    - `voltage`
  - min:
    - `status`
    - `overload`
- submodules:
  - PrintStats: `debug`
    - enable\_output: `True`

# Konfiguration der Datenspeicherung

## Outputs:

```
# Ausgabemodule zum Schreiben von Ergebnisse in eine beliebige Anzahl von Datenbanken
# Definiton des ersten Moduls
#   ClassName: Name der Python-Klasse des Moduls
#   path.to.module: relativer Importpfad des Output-Moduls
- CSVStorage: csv_file
  path: logs

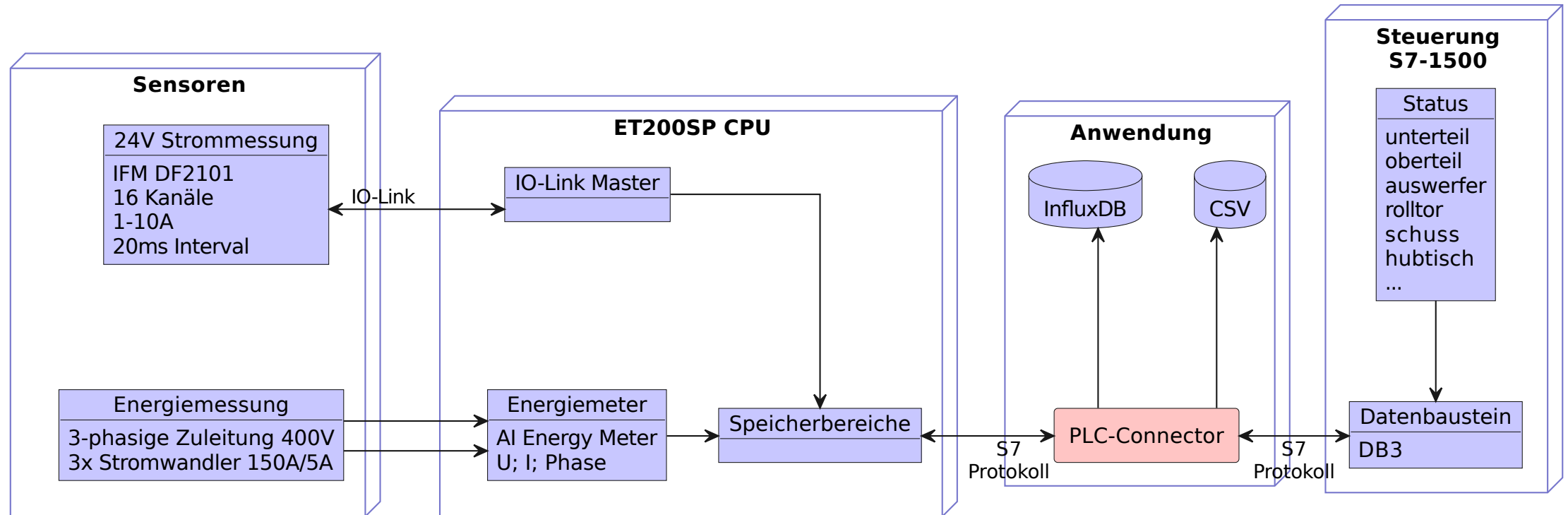
- InfluxDB: influxdb
  url: "http://localhost:8086"
  token: "<token>"
  org: "laempe"
  bucket: "energy-monitor"
```

# Praktische Versuche

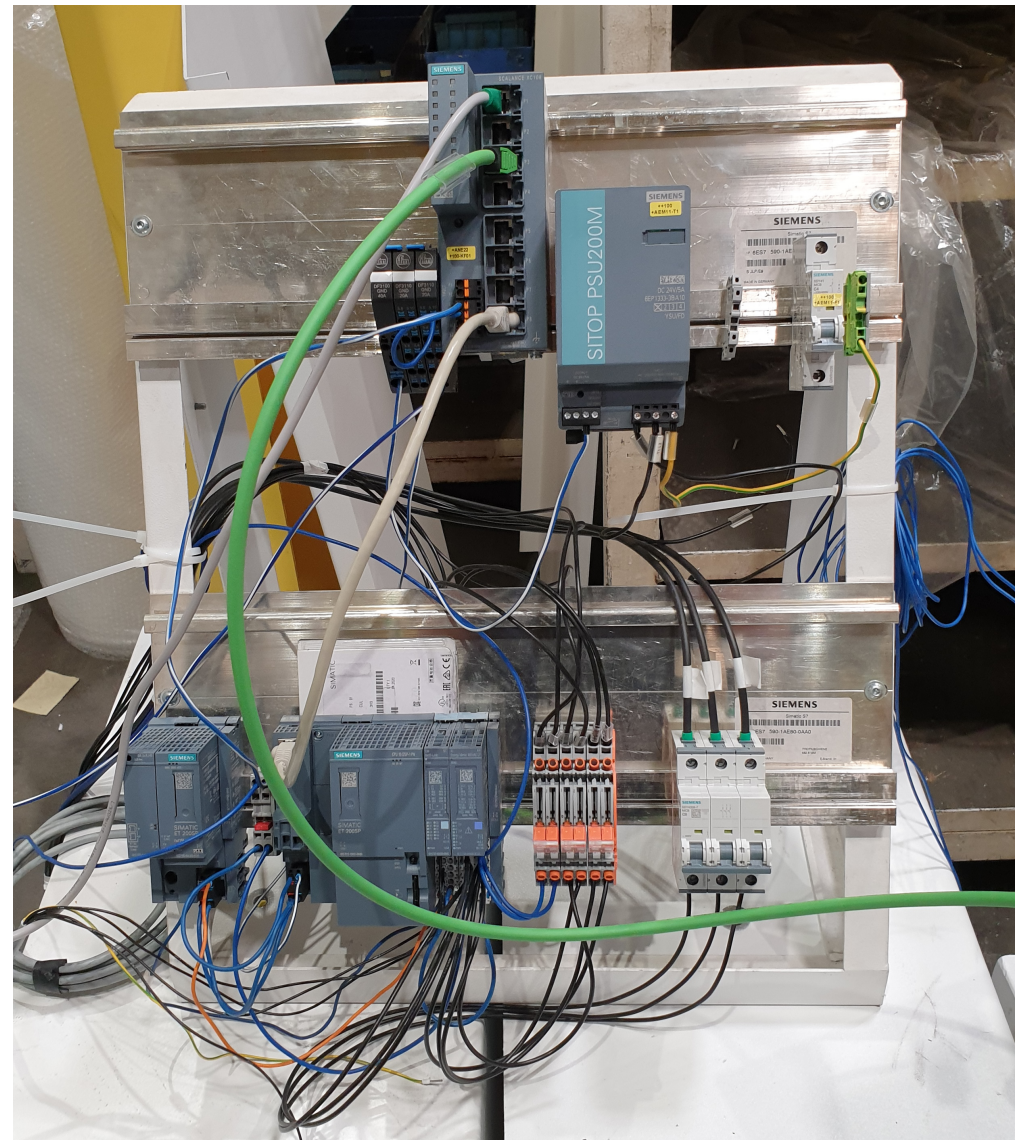
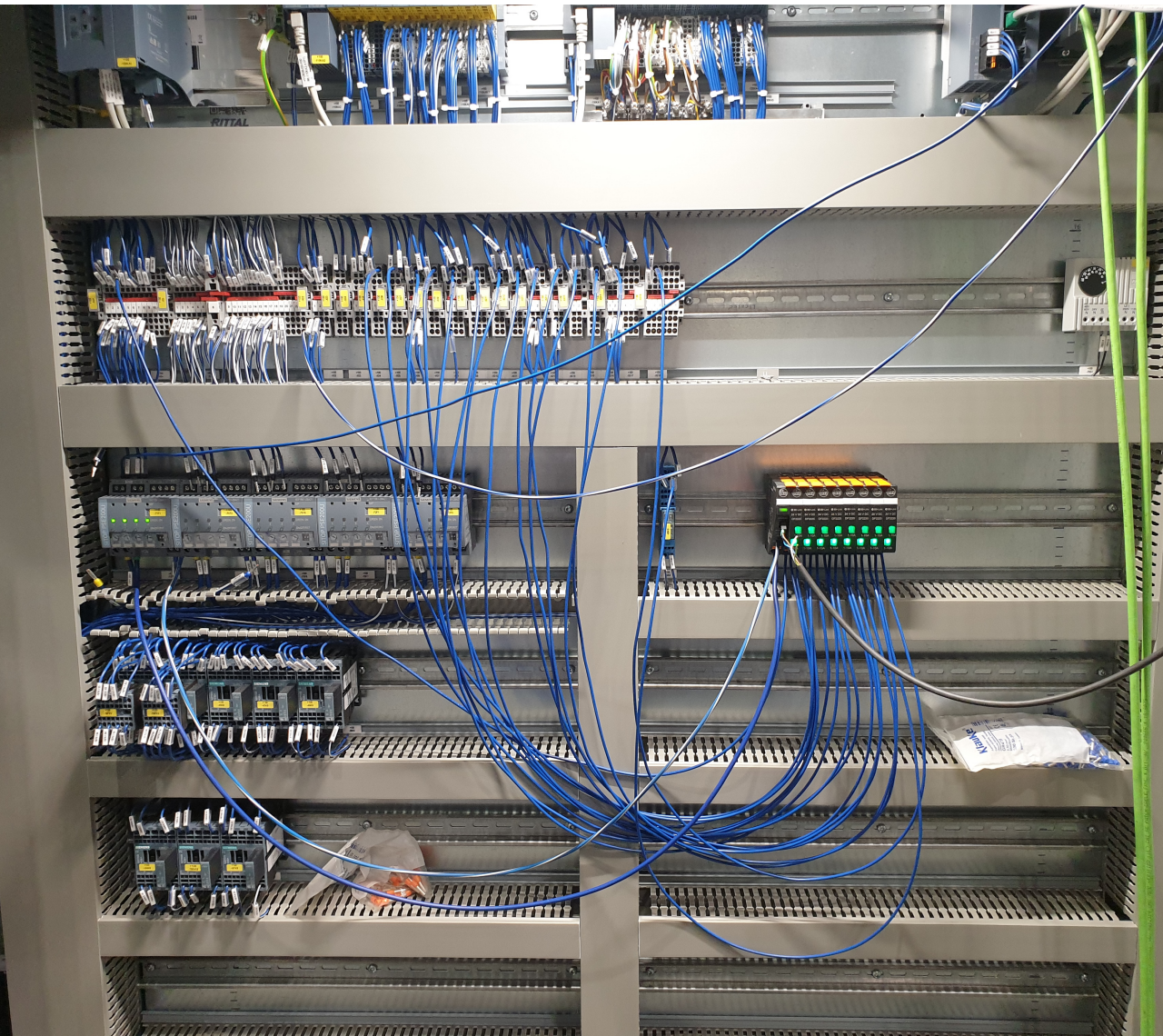
- Testaufbau
- 1. Versuch an einer LFB65 Kernschießmaschine
- 2. Versuch an einer LL25 Kernschießmaschine
- Messergebnisse



# Messaufbau LFB65



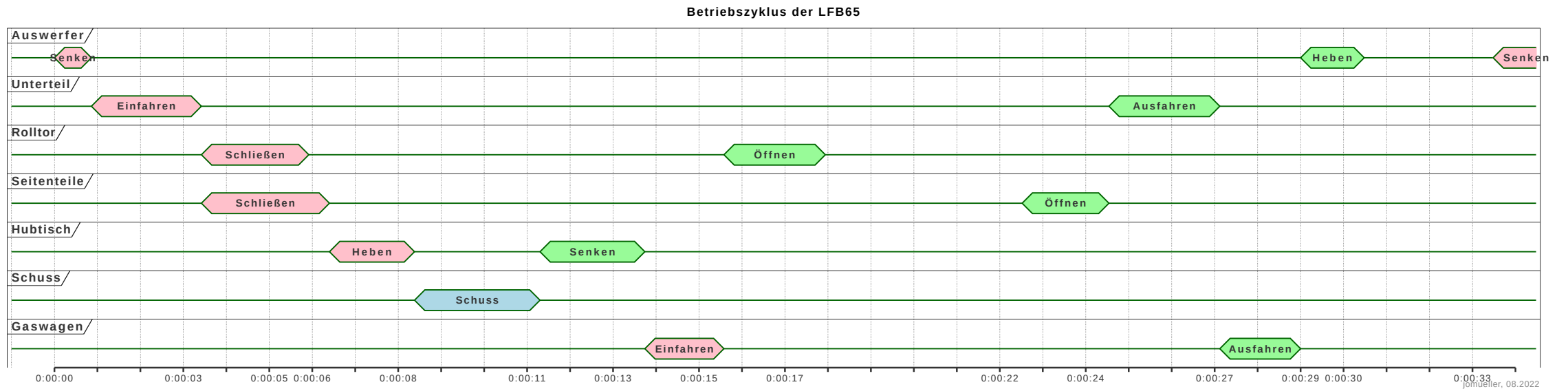






# Ergebnisse

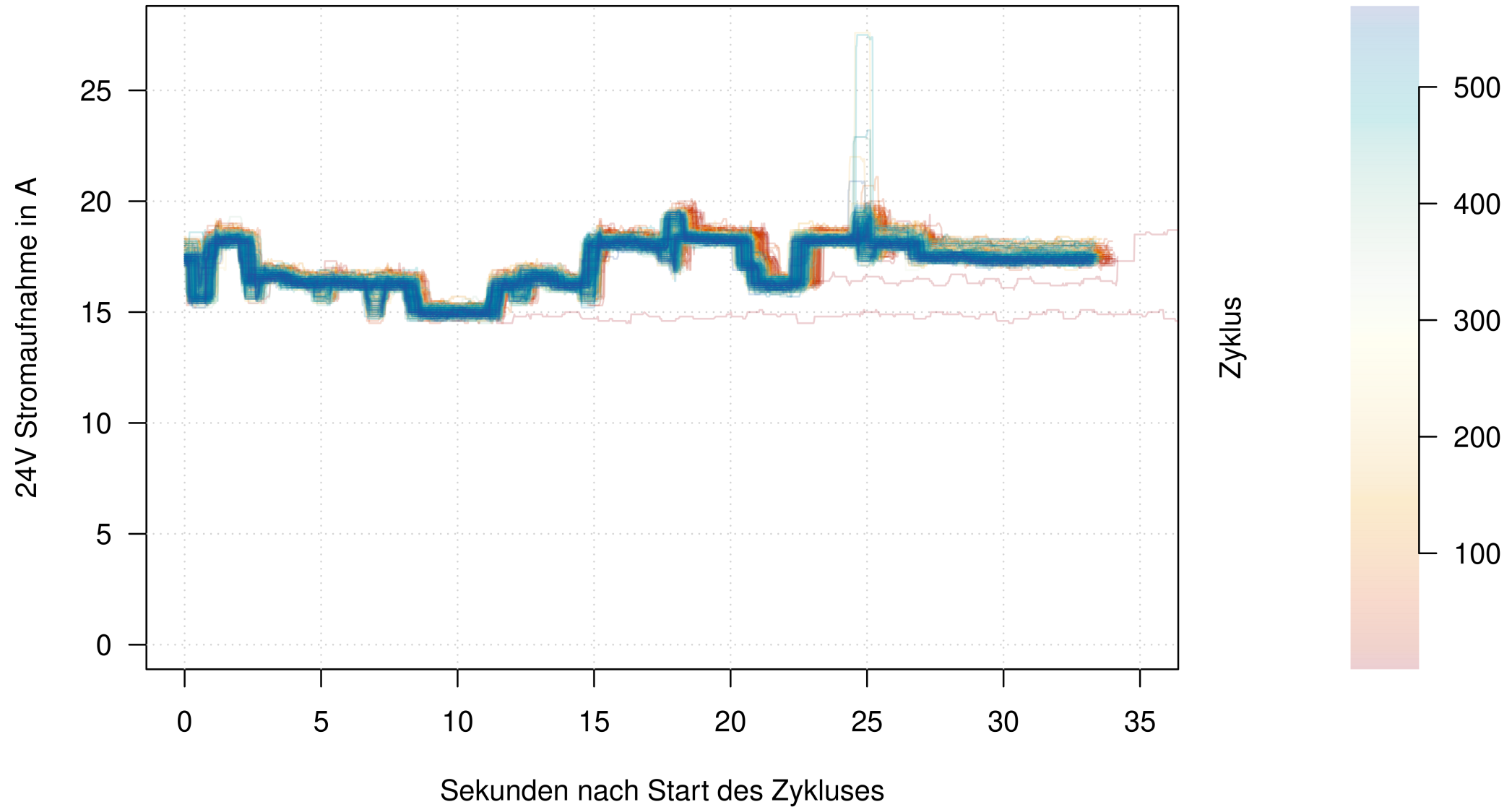
- Einsicht in den Prozessablauf



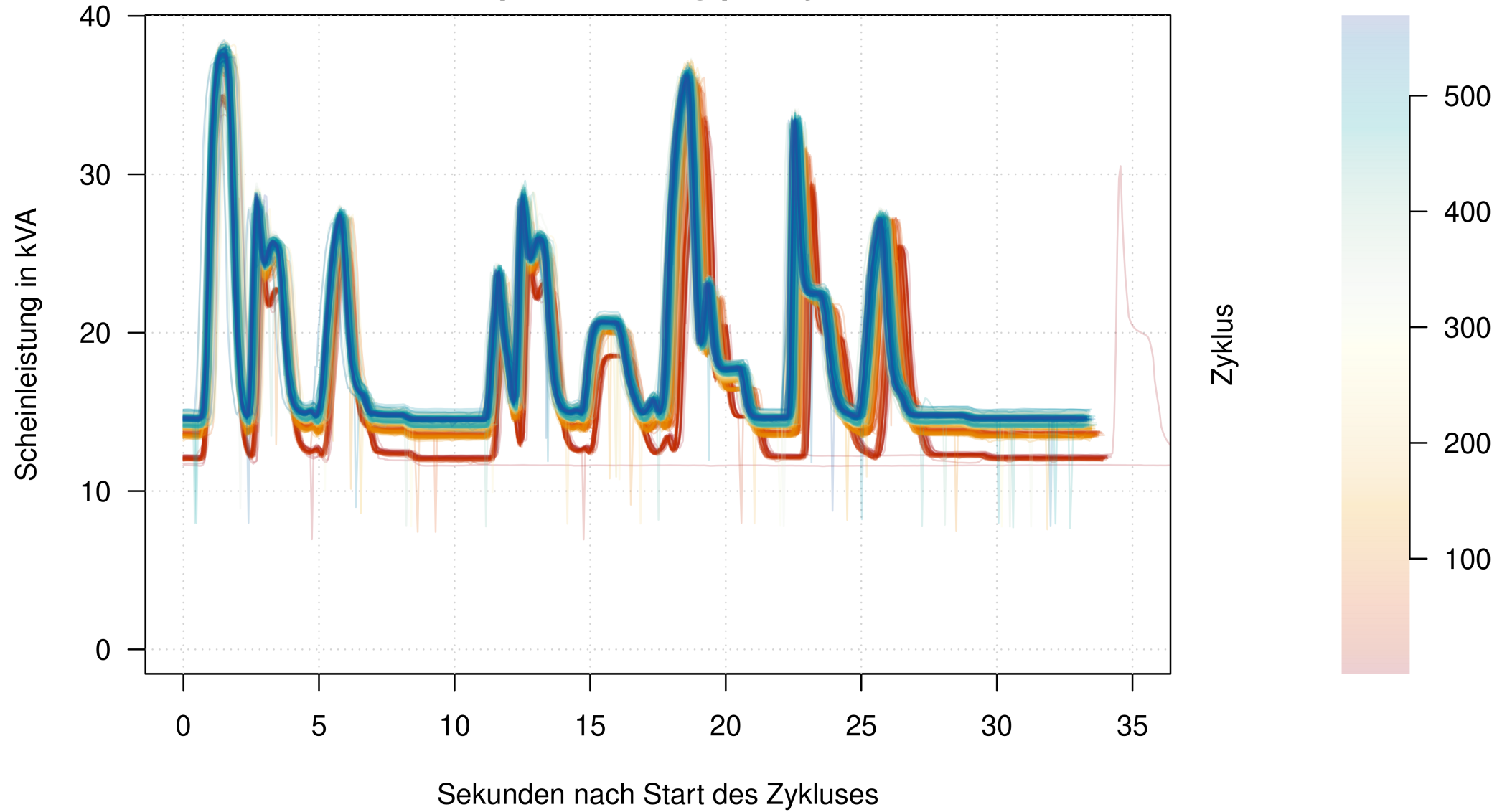
# Ergebnisse

- Korrelation von Anlagenstatus mit dem Energieverbrauch
- Auswertung für den Versuch an der LFB65
- Identifikation von möglichen Problemen

**Strom am 24V Netzteil pro Zyklus**



## Einspeiseleistung pro Zyklus



# Demo PLC-Connector

- Mit dem Replay-Inputmodul