Einführung Gebäudetechnik: IT & Bussysteme

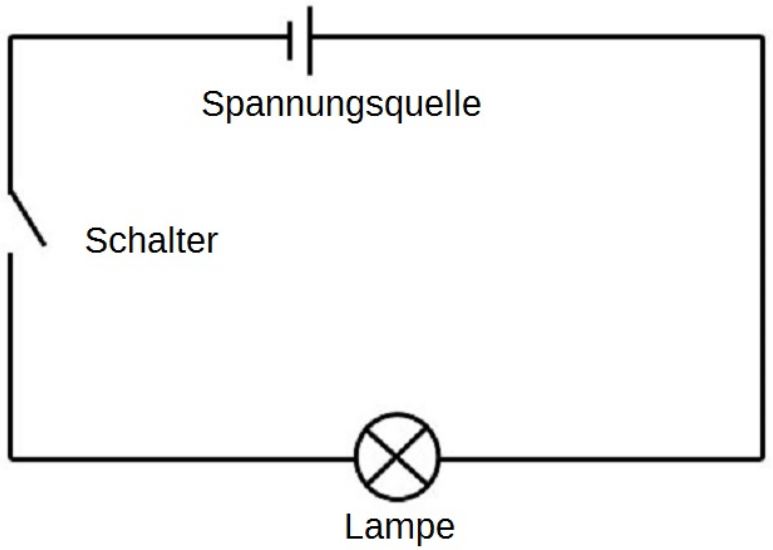
Dr. Julian Huber

Management Center Innsbruck

Beispiel Glühbirne

Klassische Glühbirne

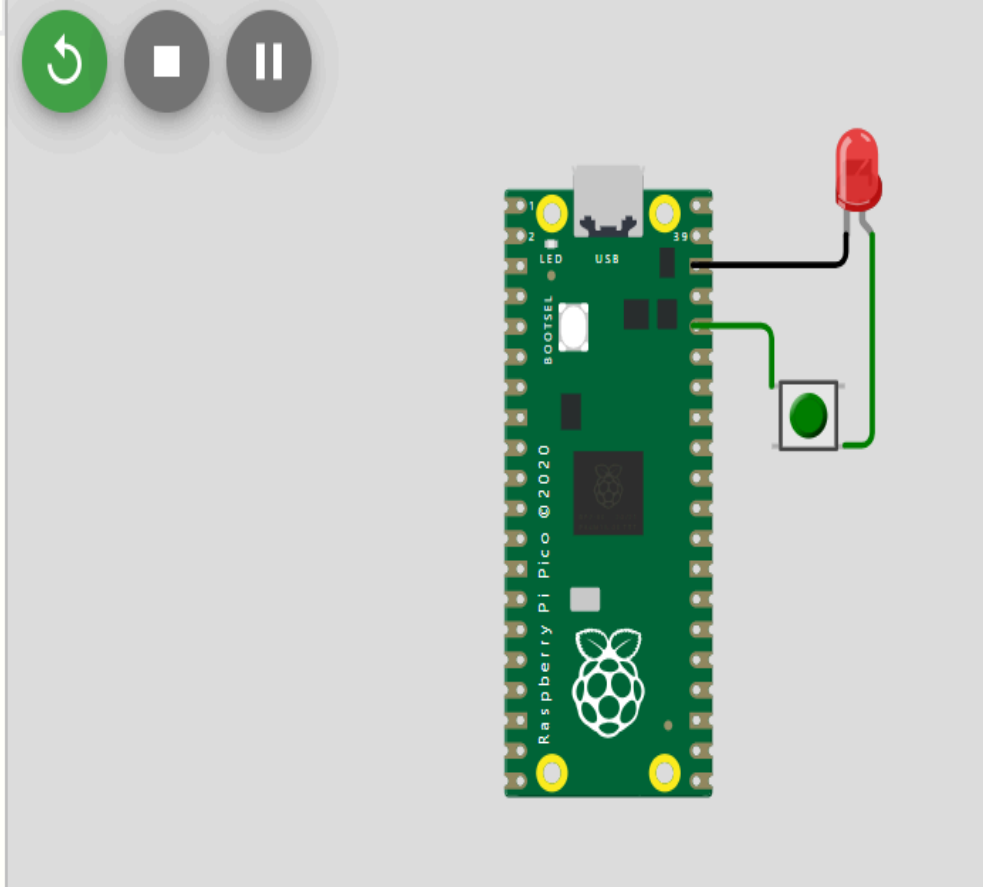
- **Energieeffizienz**: 5%
- **Lebensdauer**: 1.000 Stunden
- **Farbtemperatur**: fix z.B. 2.700 K
- **dimmbare Variante**: z.B. durch Unterspannung
- **Preis**: 1 €



code.py diagram.json

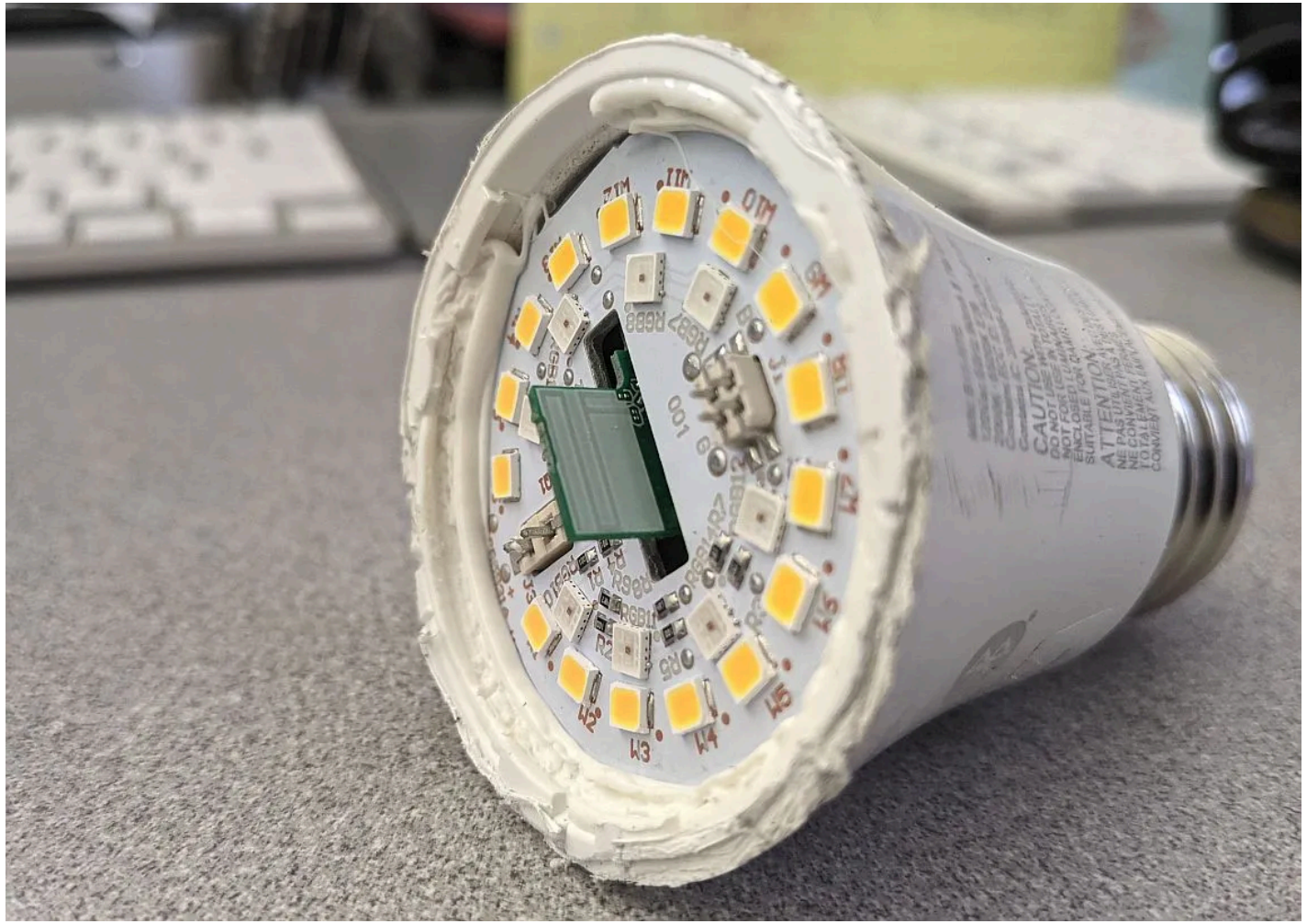
```
1  # No code needed
```

Simulation



Moderne LED-Glühbirne

Moderne LED-Glühbirne

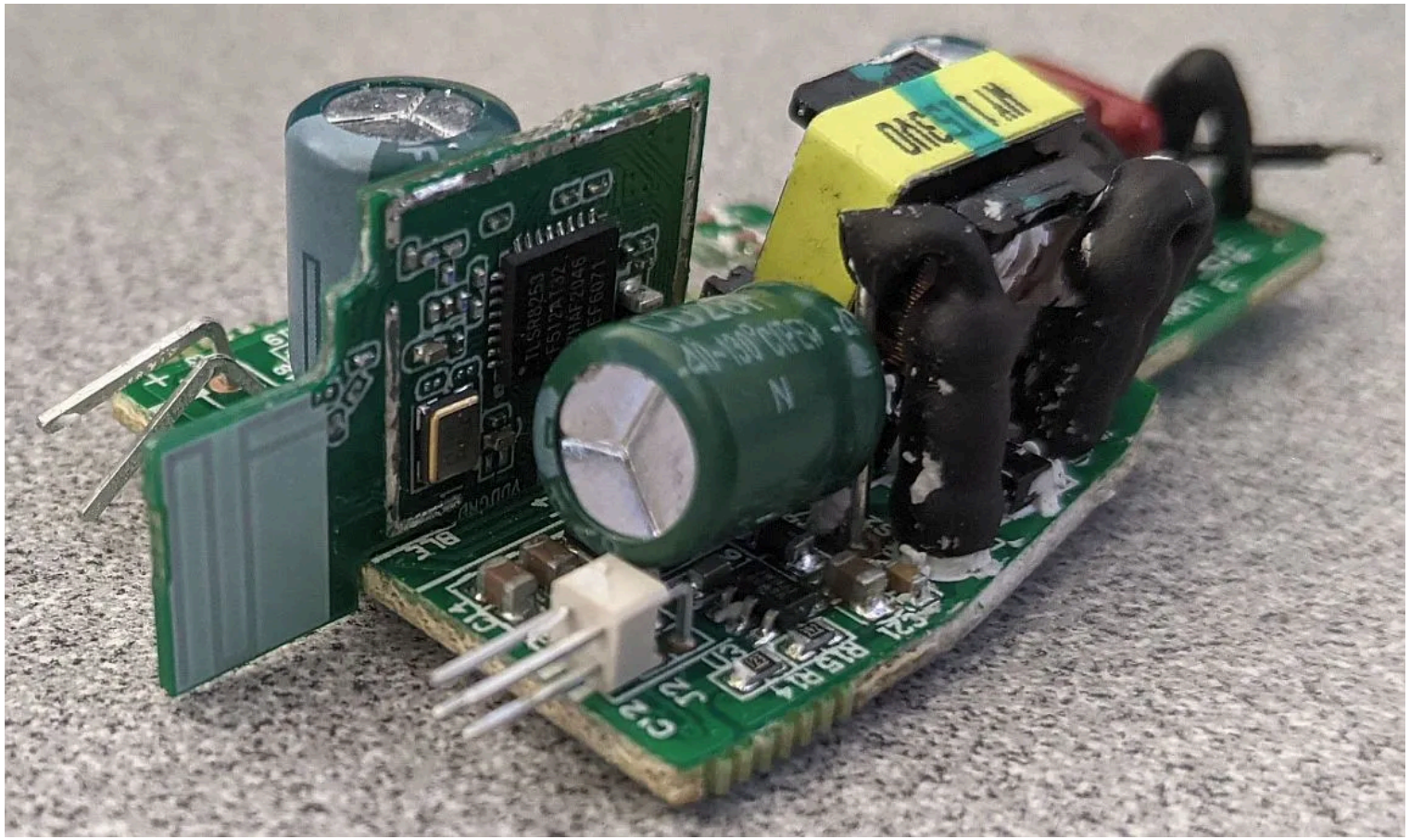


Moderne LED-Glühbirne

- **Energieeffizienz**: 80%
- **Lebensdauer**: 25.000 Stunden
- **Farbe/temperatur**: variabel
- **dimmbare Variante**: z.B. durch Pulsweitenmodulation
- **Preis**: 10 €

Eingebettetes System

- jedes Gerät hat einen eigenen Mikrocontroller
- **Software-Code** steuert die Farbe, Helligkeit, ...



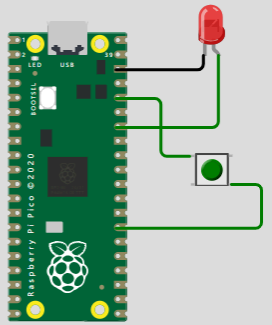
- Trennung von Ein- und Ausgängen
- diese werden erst durch Software verknüpft

WOKWI SAVE SHARE 2_LED_Taster

code.py diagram.json

```python
import board
import digitalio
import time

# Define the LED
led = digitalio.DigitalInOut(board.GP28)
led.direction = digitalio.Direction.OUTPUT

# Define the button
button = digitalio.DigitalInOut(board.GP21)  # Assuming GP21 is the button pin
button.direction = digitalio.Direction.INPUT
button.pull = digitalio.Pull.DOWN

# State variable to track button presses
button_last_state = False

# LED state (False = Off, True = On)
led_state = False

while True:
    # Read the current state of the button
    button_state = button.value

    # Check if the button was pressed
    if button_state and not button_last_state:
        # If the LED is off, turn it on immediately
        if not led_state:
            led_state = True
            led.value = led_state
        # If the LED is on, delay switching it off by 2 seconds
        else:
            time.sleep(2)
            led_state = False
            led.value = led_state

    # Update last button state
    button_last_state = button_state

    # Small delay to debounce the button
    time.sleep(0.05)

```

Simulation



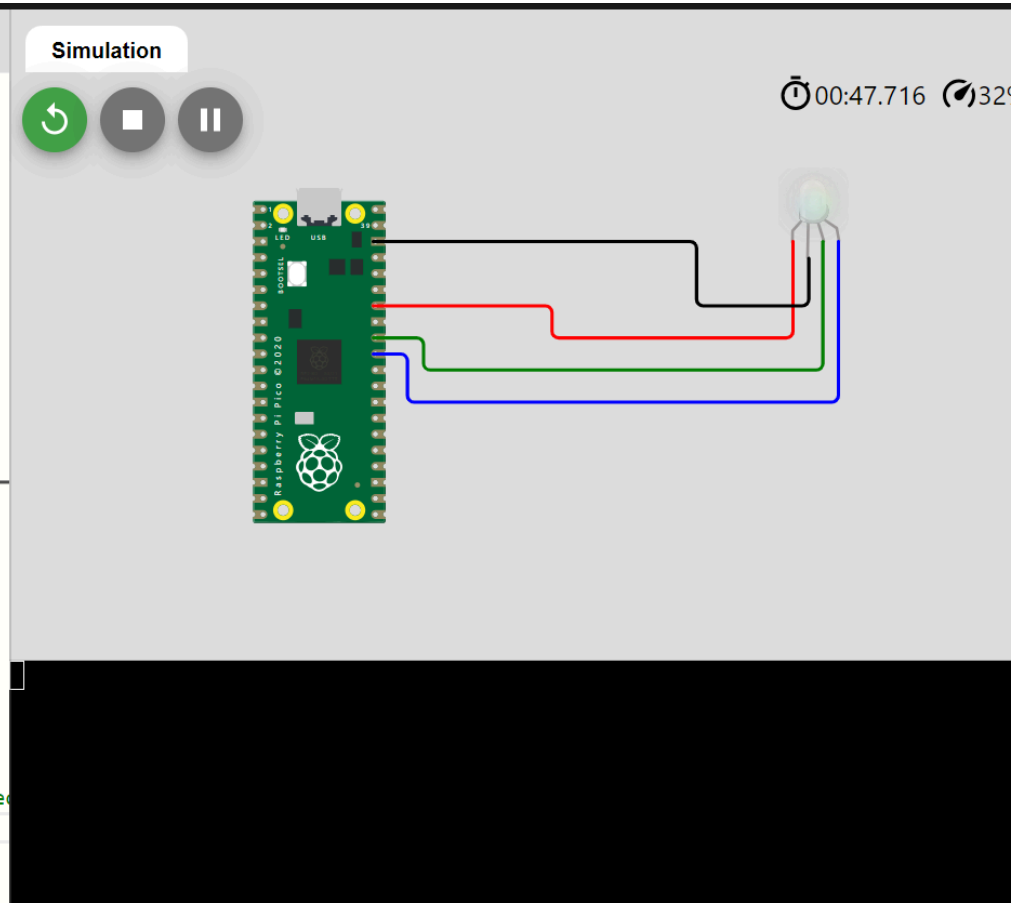
🎯 Lernziele

- Konzeption von Schaltregeln als **Endliche Automaten** oder **Wahrheitstabellen**
- Anschluss von Sensoren und Aktoren an **Mikrocontroller** und **Speicherprogrammierbaren Steuerungen**
- Umsetzung in **Software-Programmen** mittels Kontrollstrukturen

3-Kanal-LED-Controller mit Pulsweitenmodulation

code.py | diagram.json

```python
import board
import pwmio

# Initialize PWM outputs for each color channel
red_pwm = pwmio.PWMOut(board.GP28, frequency=5000, duty_cycle=0)
green_pwm = pwmio.PWMOut(board.GP27, frequency=5000, duty_cycle=0)
blue_pwm = pwmio.PWMOut(board.GP26, frequency=5000, duty_cycle=0)

# Function to set the color of the RGB LED
def set_rgb_color(red_value, green_value, blue_value):
    # Convert 0-255 range to 0-65535 for PWM
    red_pwm.duty_cycle = int((red_value / 255) * 65535)
    green_pwm.duty_cycle = int((green_value / 255) * 65535)
    blue_pwm.duty_cycle = int((blue_value / 255) * 65535)

# Example usage: Set initial color to purple (Red + Blue)
red_value = 255  # Example value for red channel (0-255)
green_value = 255  # Example value for green channel (0-255)
blue_value = 255  # Example value for blue channel (0-255)

set_rgb_color(red_value, green_value, blue_value)

while True:
    # The main loop can be used to update the color dynamically if neede
    set_rgb_color(red_value, green_value, blue_value)
```



🎯 Lernziele

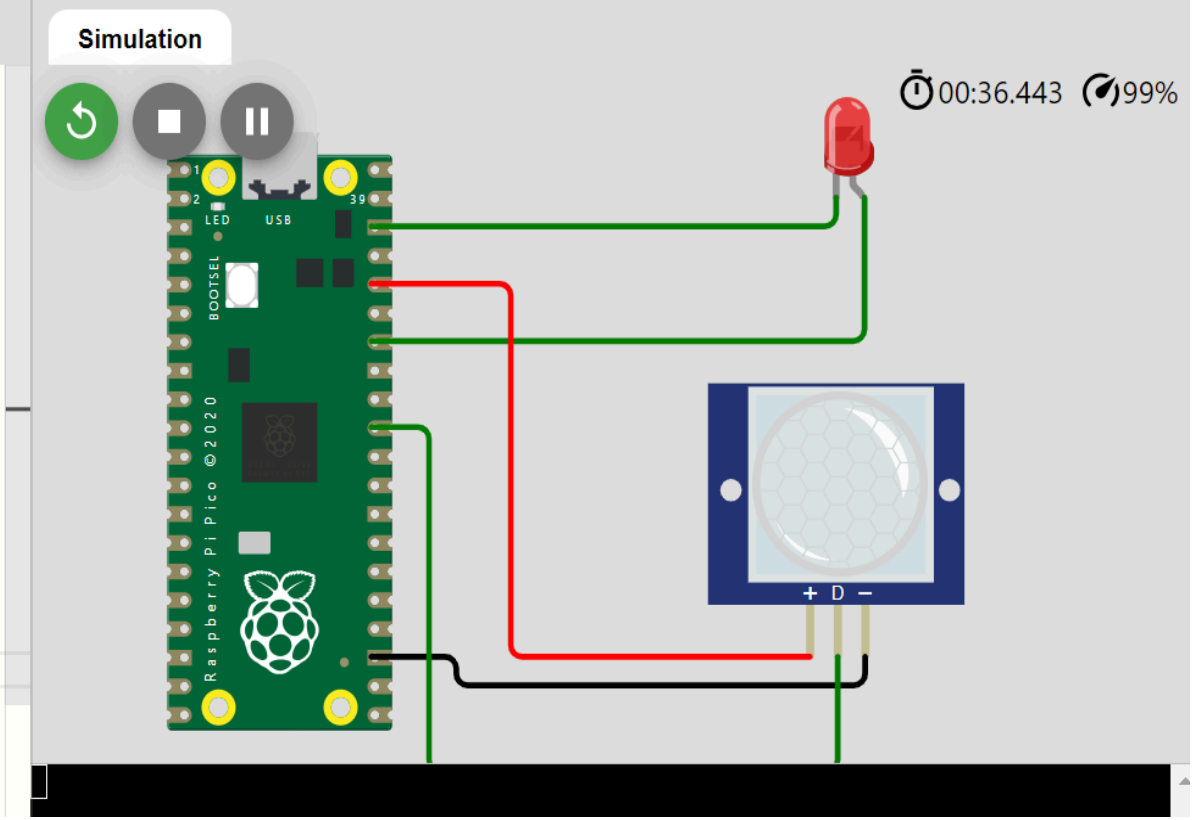
- Einsatz von **Zahlensysteme** und **Bit-Operationen**
- Unterschiede zwischen **Analogen** und **Digitalen Signalen**
- Auswahl von **Ein- und Ausgabegeräten**

Verbindung von Sensoren und Aktoren

WOKWI SAVE SHARE 6_LED_Bewegungsmelder Docs

code.py diagram.json

```python
import board
import digitalio
import time

# Set up the PIR motion sensor
pir_sensor = digitalio.DigitalInOut(board.GP26)
pir_sensor.direction = digitalio.Direction.INPUT

# Set up the LED
led = digitalio.DigitalInOut(board.GP28)
led.direction = digitalio.Direction.OUTPUT

while True:
    if pir_sensor.value:
        led.value = True  # Turn on LED when motion is detected
        time.sleep(2)
        led.value = False
    else:
        led.value = False  # Turn off LED when no motion is detected

    time.sleep(0.1)  # Small delay to debounce the sensor
```



🎯 Lernziele

- **Informationsübertragung mittels Bussystemen** und verschiedenen Randbedingungen z.B. Echtzeitfähigkeit, Teilnehmerzahl, Störsicherheit
- Grundkonzepte der **Steuerungs- und Regelungstechnik** z.B. PID-Regler für Konstantlichtregelung

Vernetzung in Gebäuden

SOFTWARE-AUSFALL

Seit 1,5 Jahren brennt das Licht an einer Schule

Seit Sommer 2021 leuchten Tag und Nacht 7.000 Lampen in einer US-Schule. Auslöser dafür ist ein Software-Ausfall.

in Pocket speichern | merken

22. Januar 2023, 10:35 Uhr, Ingo Pakalski

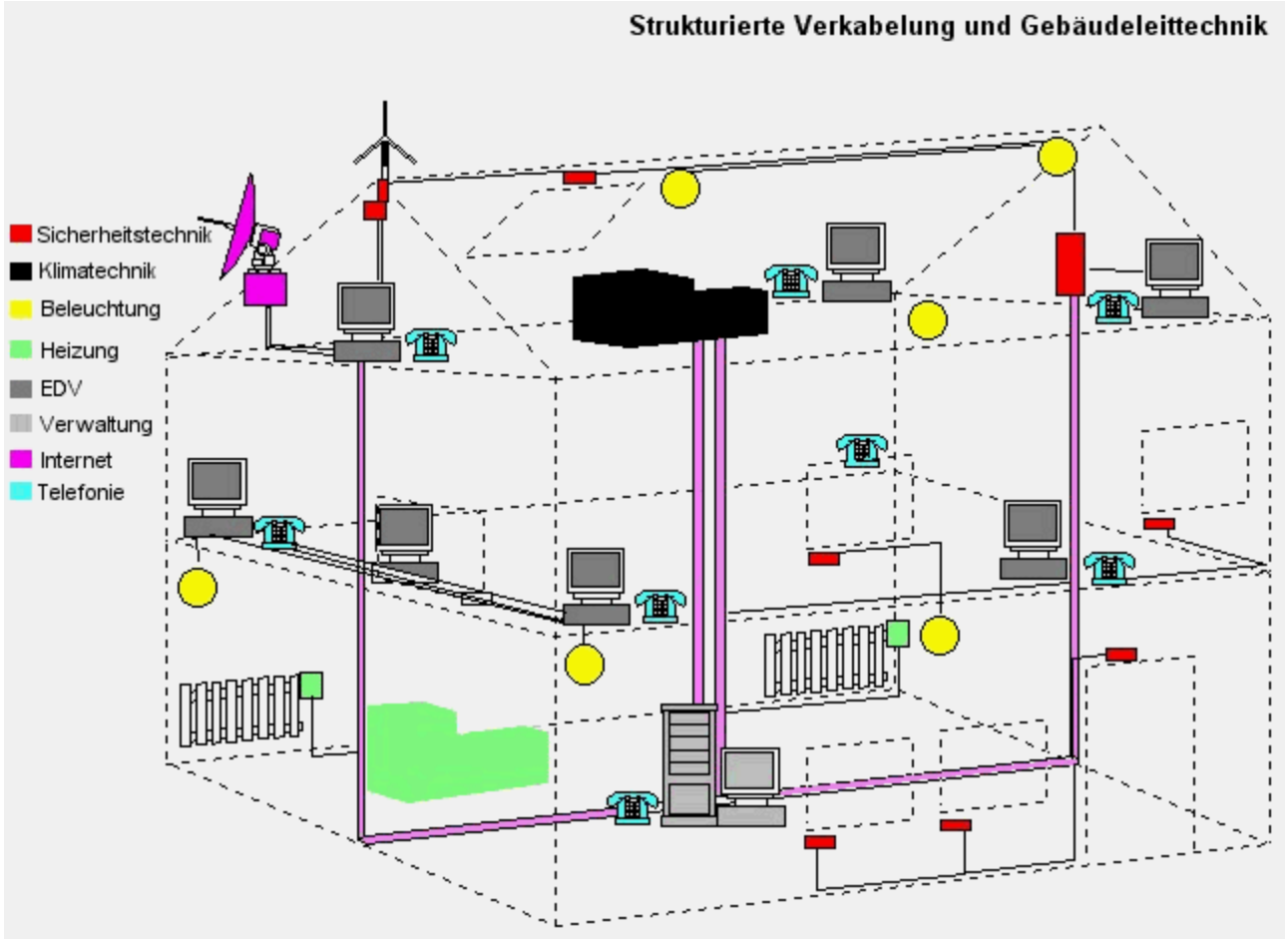
(Bild: Minnechaug Regional High School)

6. Instandhaltung und Betrieb

🎯 Lernziele

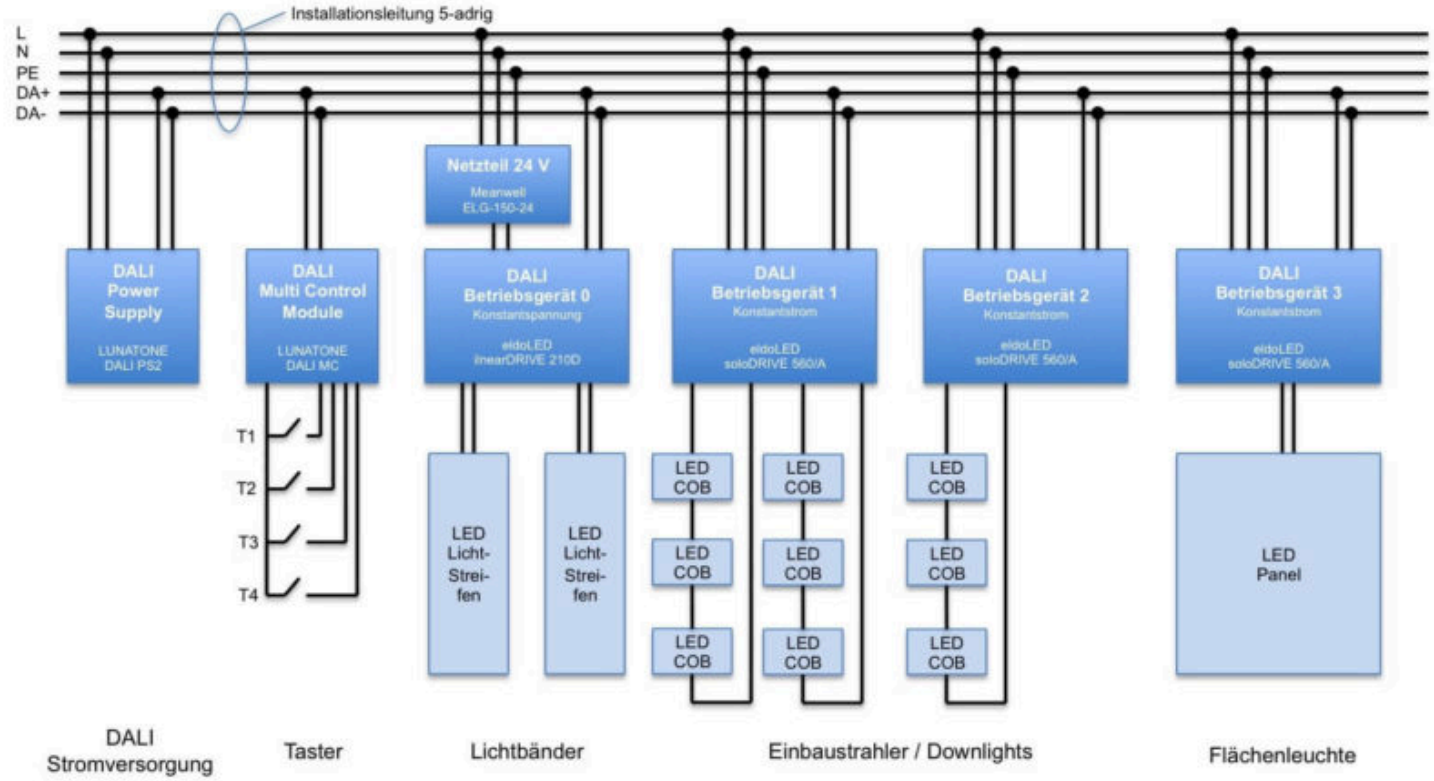
- **Netzwerkkonfiguration** für TCP/IP-Netzwerke
- **Sicherheitsaspekte** z.B. Firewall, VPN, Verschlüsselung
- **Datenspeicherung** und **Datensicherung** im Betrieb und bei Störungen

Bussysteme als Zentrales Nervensystem

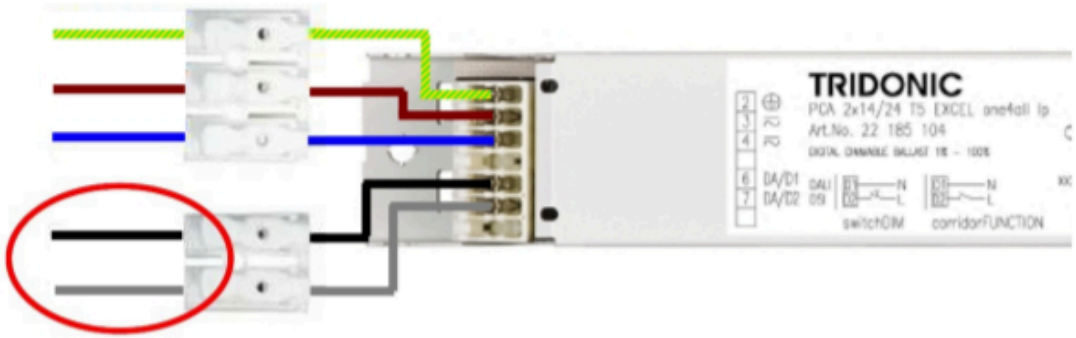


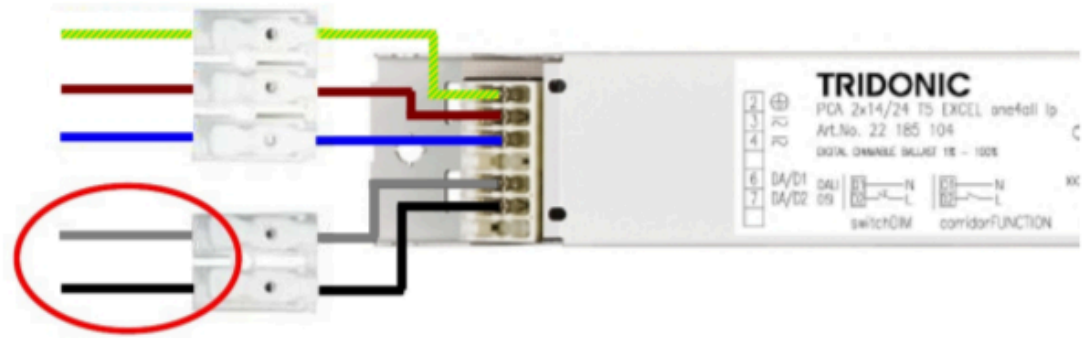
Bussysteme für verschiedene Gewerke

🎯 DALI - Intelligente Lichtsteuerung

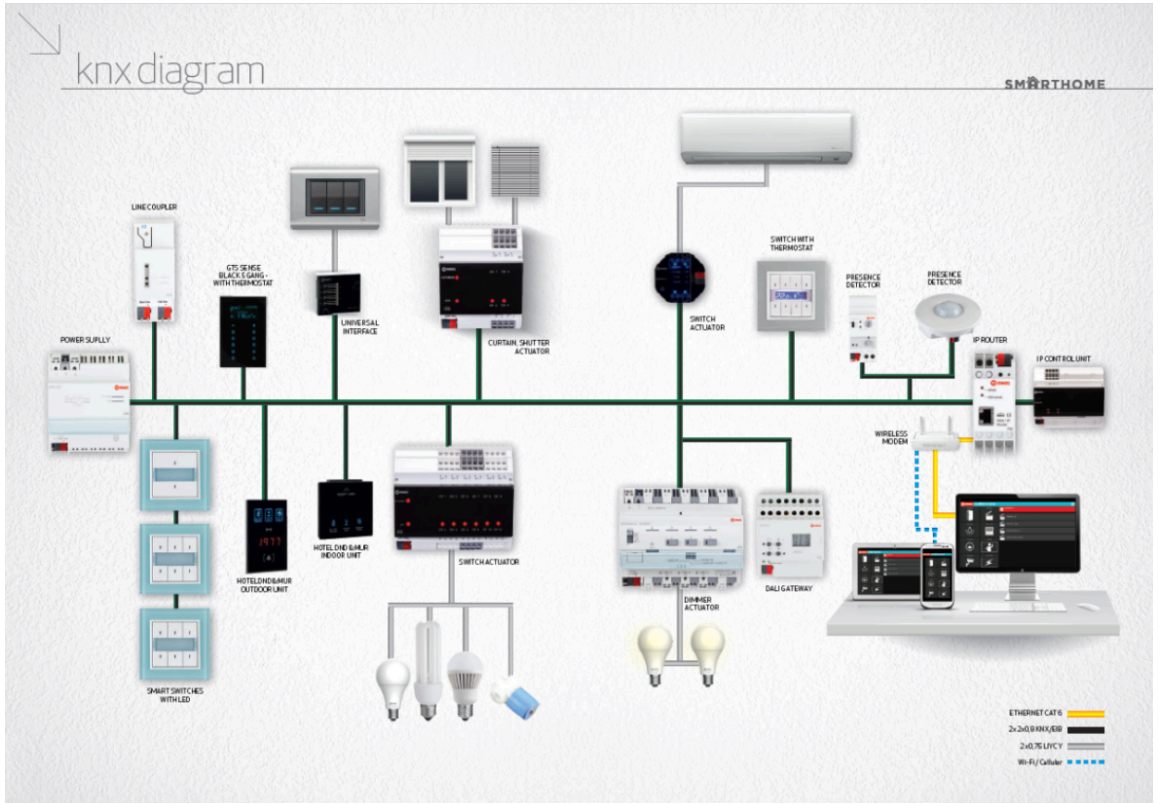


- **Verkabelung** und **Addressierung** von bis zu 64 Leuchten und Zusammenfassen in **Gruppen**
- Konfiguration von **Szenen** für verschiedene Anforderungen


TRIDONIC
PCA 2x14/24 T5 EXCEL one4all lp
Art.No. 22 185 104
DIGITAL DIMMABLE BALLAST 1% - 100%
DA/D1
DA/D2
DALI
switchDIM
corridorFUNCTION
N
L


TRIDONIC
PCA 2x14/24 T5 EXCEL one4all lp
Art.No. 22 185 104
DIGITAL DIMMABLE BALLAST 1% - 100%
DA/D1
DA/D2
DALI
switchDIM
corridorFUNCTION
N
L

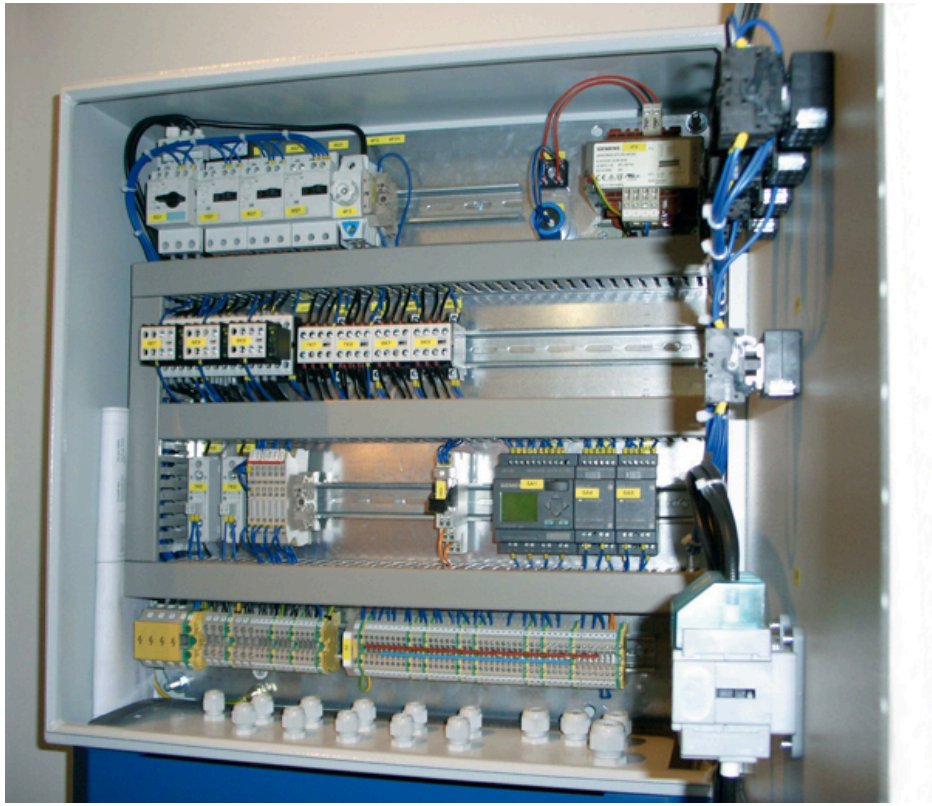
🎯 KNX & BACnet - Übergreifende Gebäudeautomation



- **Einsatzgebiete** und **Topologien** verschiedener Bus-Systeme
- Planung von **Raumautomations-Funktionen** mittels **Schemata**

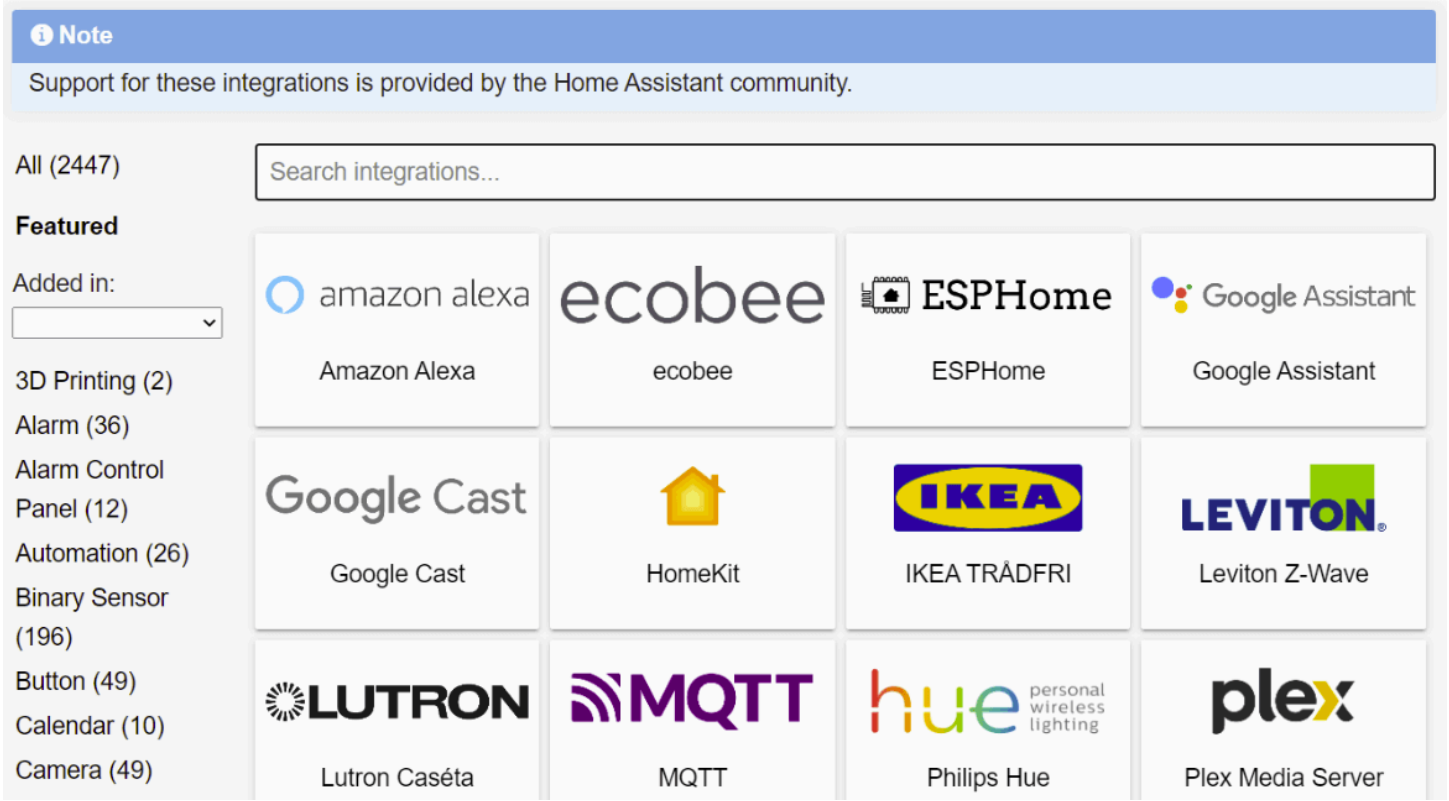
Anlagenautomation

Klassische Automatisierungstechnik

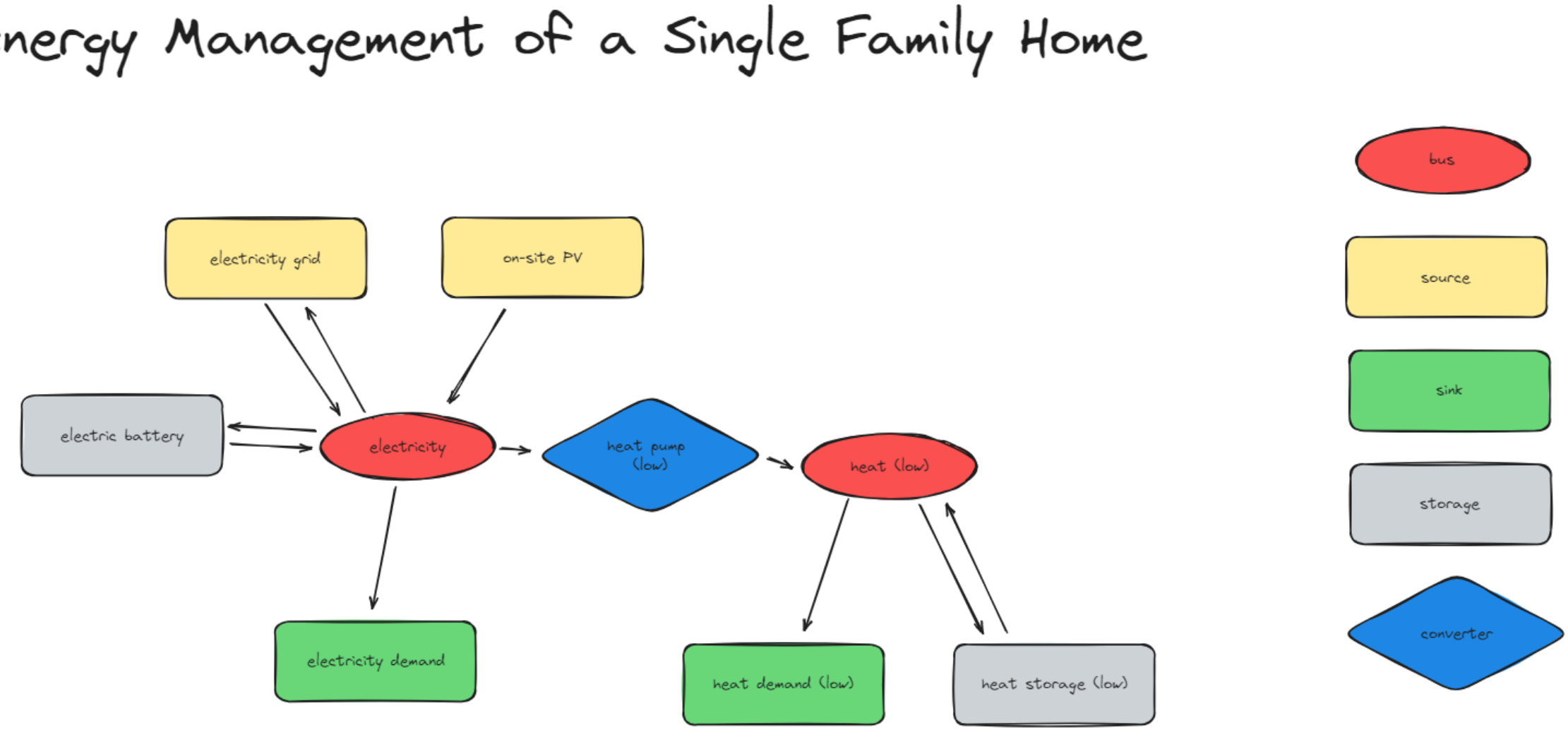


- Verdrahtung von **Aufbau von SPS-Sytemen**
- Programmierung mittels **Funktionsplänen** oder **Strukturiertem Text**

Ethernet/IP



- Grundtechnologien des **Internets**, z.B. **TCP/IP**, **JavaScript**, **HTML**
- Datenübertragung mittels **HTTP**, **MQTT**



1. Sicherstellung von Versorgung und Entsorgung

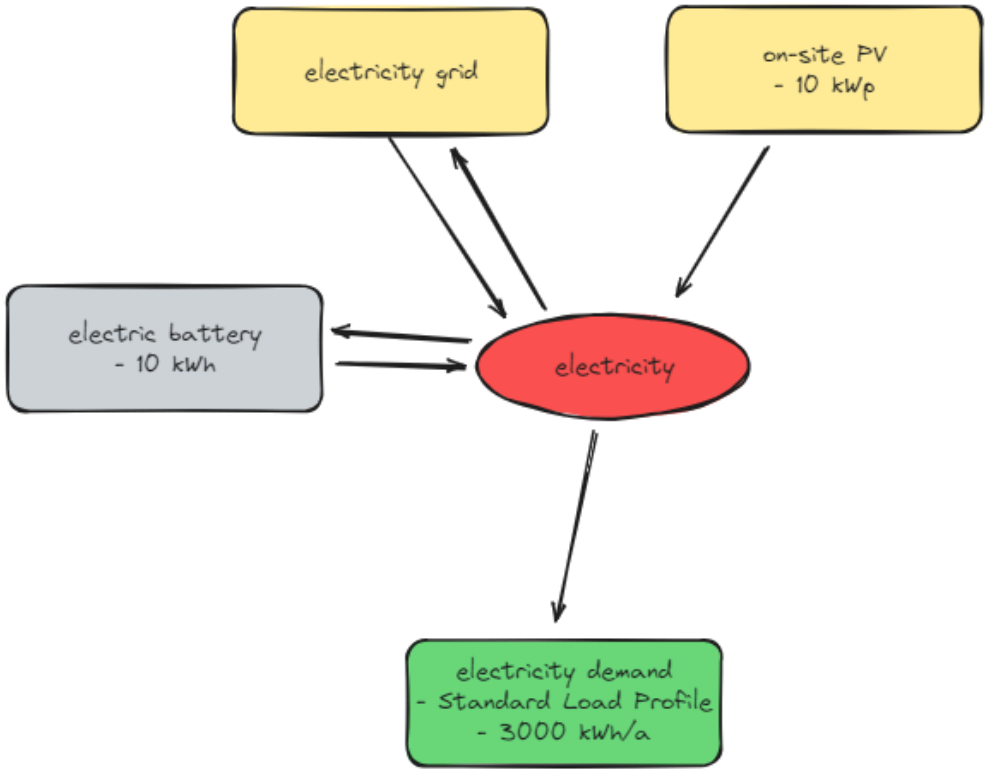
Energiesysteme

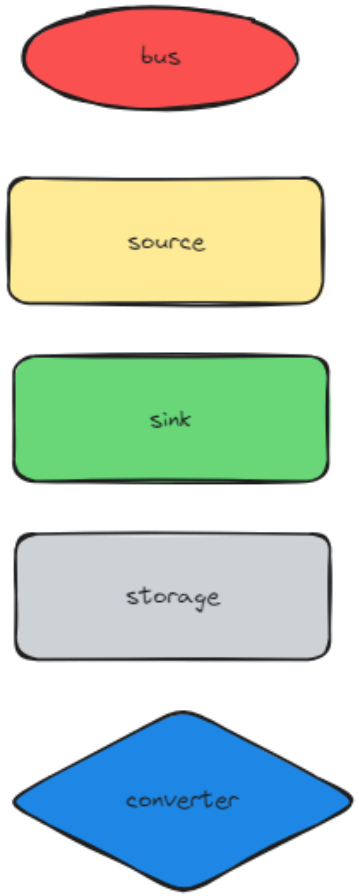
- **Versorgung**, Erzeugung, Speicherung, Verteilung, Verbrauch
- **Skalen**: Gebäude, Stadt, Region, Land, Welt
- **Sektoren**: Strom, Wärme, Mobilität
- **Technologien**: Erneuerbare, Speicher, Wärme, Elektrolyse, ...
- **Zielkonflikte**: Klimaneutralität, Versorgungssicherheit, Wirtschaftlichkeit

Energiemanagement

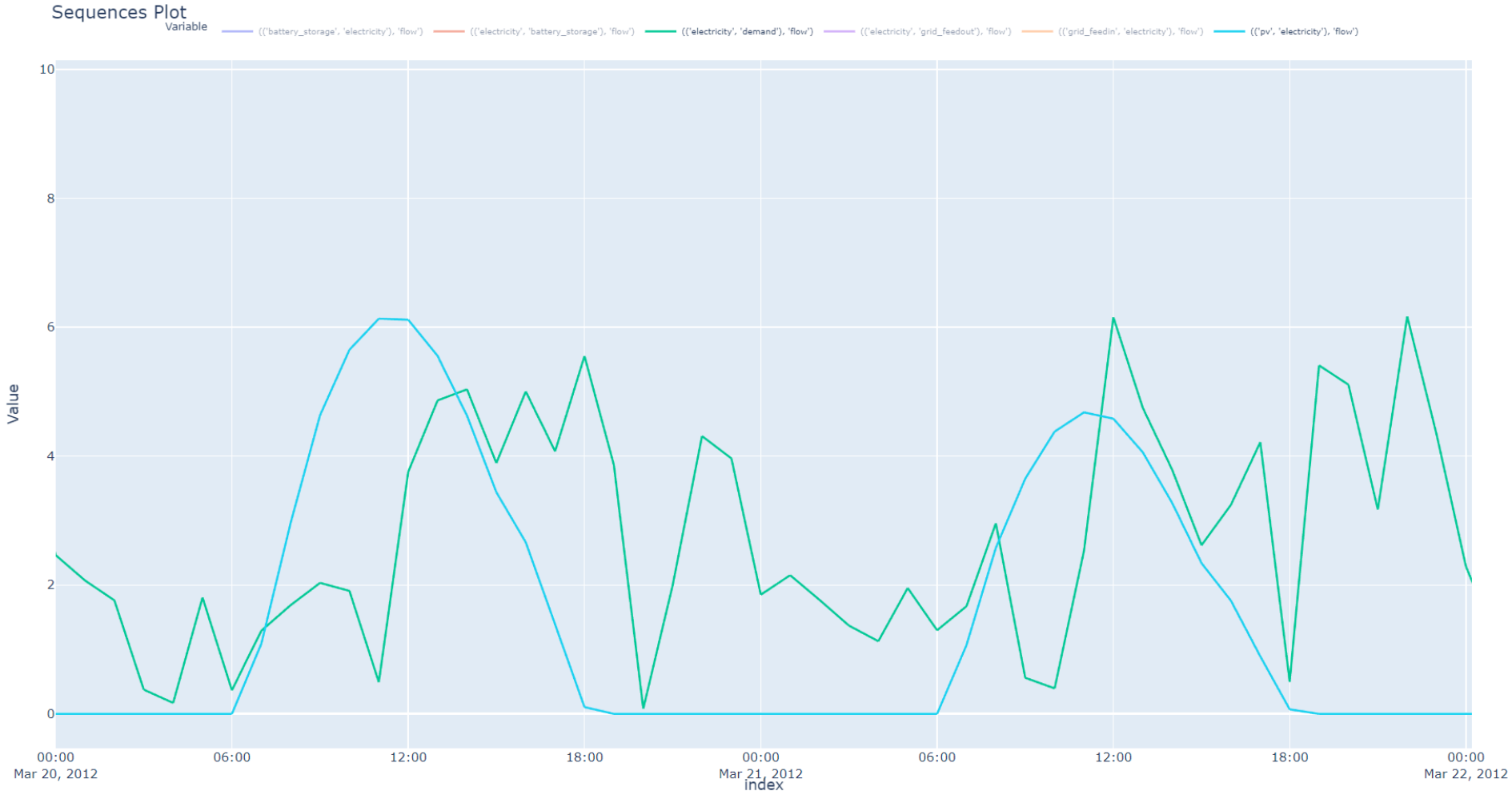
- **Planung**: Welche Technologien, Standorte, Größen
- **Betrieb**: Welche Steuerung, Regelung, Optimierung
- **Optimierung**: Potenziale aufdecken, Kosten minimieren

Beispiel: Optimierung eines EFH mit PV und Speicher

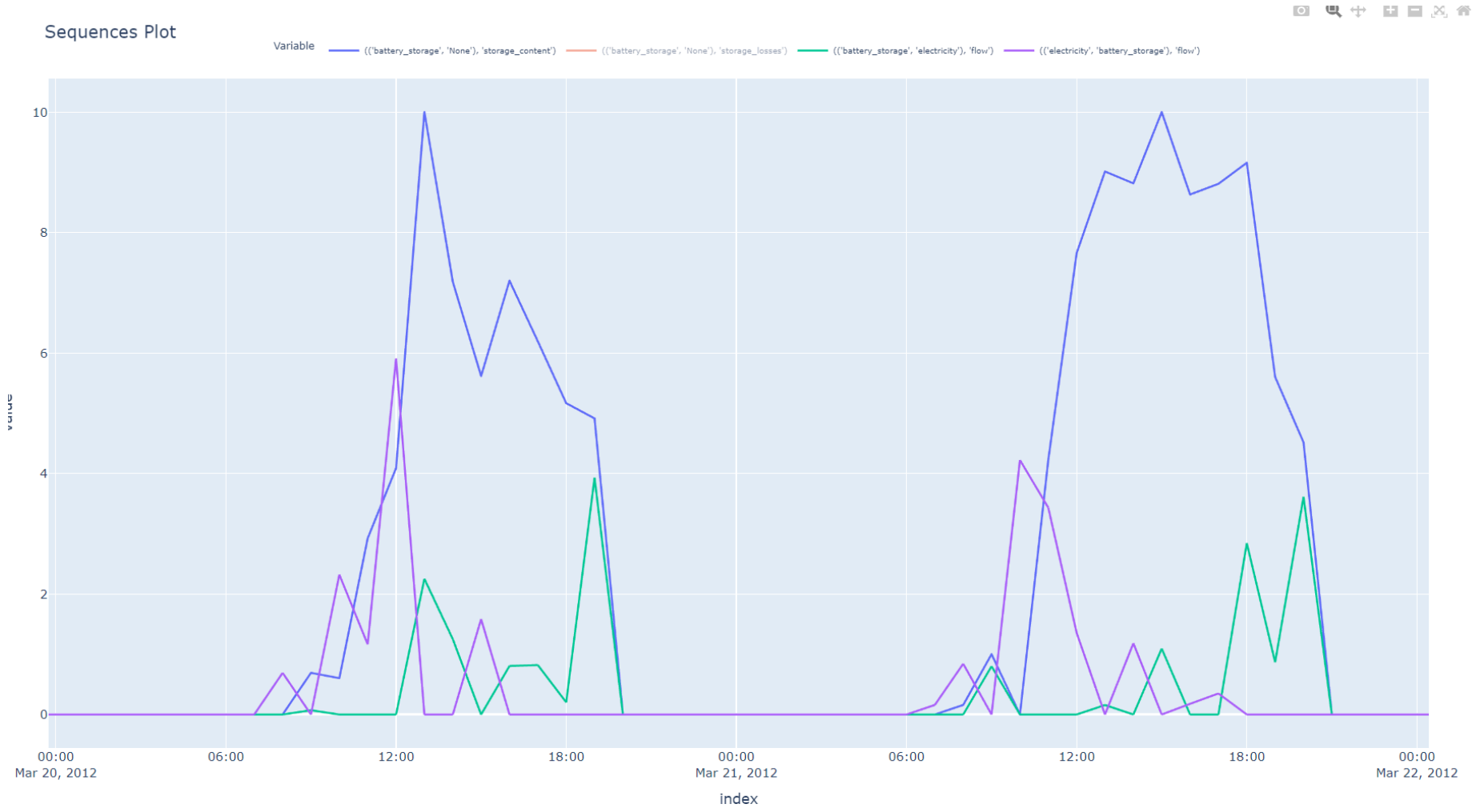




Lasten und Ertäge



Speicherbewirtschaftung



🎯 Dafür benötigt

- Zentrales System zur **Steuerung** und **Optimierung**
- **Smart Metering** für genaue Verbrauchsdaten und Abrechnung
- Schnittstelle oder Bus-Systeme für **Speicher-Management** und
- externe **Datenquellen** für Wetterdaten, Strompreise, ...

Semester 1: Grundlagen Informationstechnologie & Datensicherheit

1. Rechnersysteme
2. Informationsdarstellung
3. Compiler und Algorithmen
4. Programmieren
5. Datenspeicherung
6. Kommunikation
7. IT-Sicherheit und Datenschutz

Semester 2: Bussysteme

1. Gebäudeautomation und Planung
2. Messkette und Computer-Systeme
3. Steuerung- und Regelungstechnik
4. Speicherprogrammierbare Steuerung
5. Bussysteme
6. Smart Metering

Semester 2: Labor Bussysteme