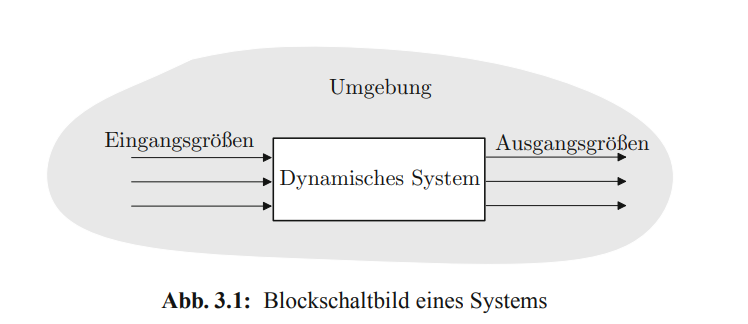
1. **Regelungstechnik - Diskret**

## Was ist ein Blockschaltbild einer Regelung bzw. Steuerung? (1 Punkt)?

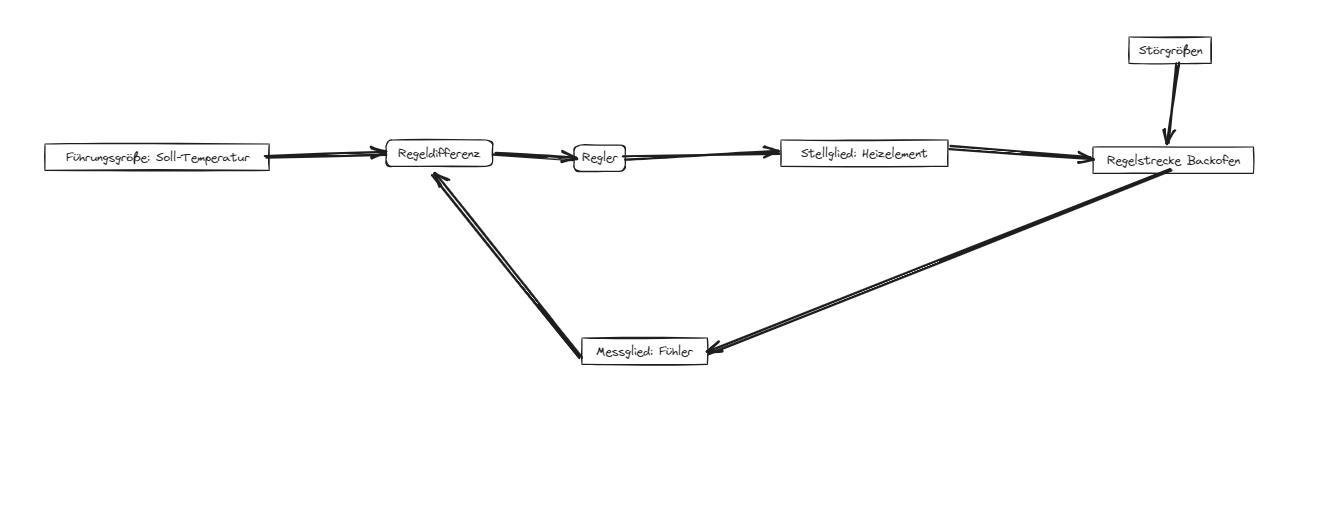
Ein Blockschaltbild einer Regelung / Steuerung ist eine grafische Darstellung eines Regelungssystems. Es zeigt die Wechselwirkungen der einzelnen Komponenten / Sub-Systemen. Systeme werden durch Blöcke dargestellt, die durch Pfeile verbunden sind, welche die Ein- und Ausgangsgrößen darstellen. 

## Zeichen Sie eine Blockschaltbild für folgenden Sachverhalt: Ein Backofen soll seine Temperatur konstant halten. Zeichnen Sie ein Blockschaltbild für die Regelung bzw. Steuerung des Backofens und benennen Sie mindestens indem Sie auch anschreiben durch welche Komponente, diese im konkreten Fall dargestellt wird. (5 Punkt)

* Messglied
* Regelstrecke
* Regler
* Führungsgröße

Je ein Punkt für: Messglied (Temperaturfühler), Regelstrecke (Innenraum des Backofens), Regler (z.B. Microkontroller), Führungsgröße (z.B. Wunschtemperatur)

Ein Punkt für richtige Rückführung der Messgröße vom Messglied. Ein Punkt fürs Einzeichnen der Summationsstelle zum Berechnen der Regeldifferenz.



## Was versteht man unter Störgrößen, wo wirken diese im Blockschaltbild und was wären konkrete Beispiele in diesem System? (1 Punkt)

Wirken auf die Regelstrecke z.B. Öffnen der Backofentür, Thermische Trägheit des Backgutes, Umgebungstemperatur in der Küche

## Handelt es sich hierbei um eine Regelung oder eine Steuerung? Begründen Sie Ihre Antwort. (1 Punkt)

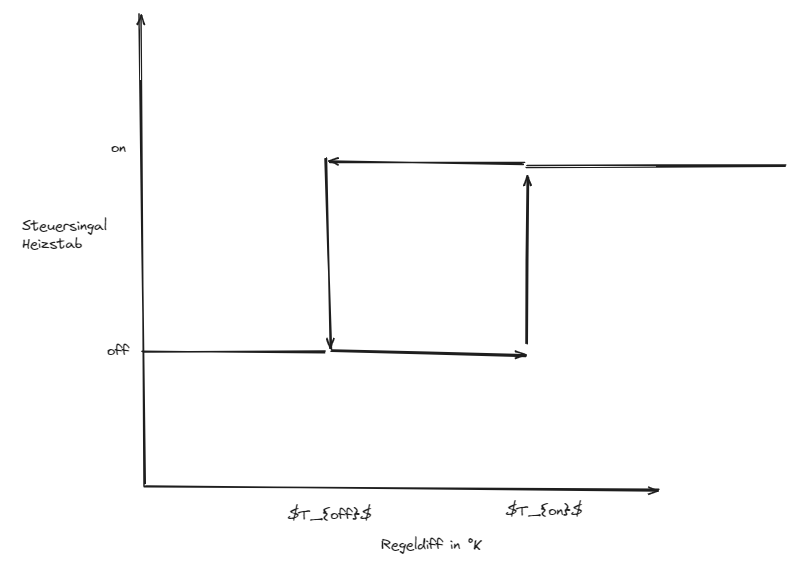
Regelung, da geschlossener Regelkreis.

## Welcher Regler-Typ bietet sich an, wenn der Heizstab nur binär (ein/aus) geschaltet werden kann? (1 Punkt)

Ein Zweipunktregler ggf. mit Hysterese bietet sich an. Bei diesem Regler wird der Heizstab entweder vollständig eingeschaltet oder ausgeschaltet, je nachdem, ob die gemessene Temperatur unter oder über einem bestimmten Schwellenwert liegt. Stetige Regler machen hier keinen Sinn.

## Gäbe es hier Gründe die dafür oder dagegen sprechen eine Hysterese einzubauen? Zeichnen Sie eine Hysteresekurve für den Regler des Backofens (2 Punkt)

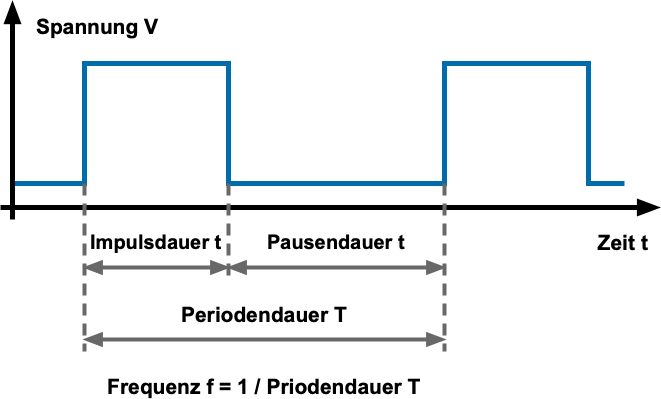
Hysterese nicht nötig, das häufig Schalten dem Heizstab eher nicht schadet. Hingegen würde eine Hysterese zu höheren Temperaturschwankungen in der Regelstrecke führen.



1. **Regelungstechnik - Stetig**

## Angenommen, wir wollen das System nun doch quasi-stetig regeln. Erklären Sie das Prinzip der Pulsweitenmodulation und skizzieren sie dies mit den richtigen Begriffen. (3 Punkte)

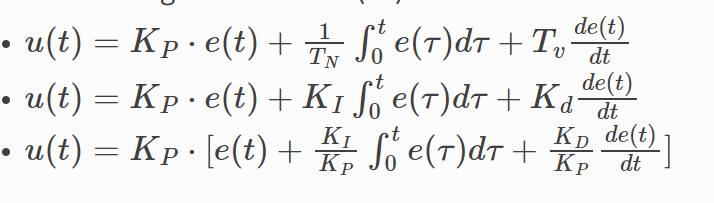
Bei der Pulsweitenmodulation wird eine quasi-stetige Ansteuerung eines Eingangs durch schnelles Ein- und Ausschalten eines digitalen Ausgangs erzielt. Durch Anpassung der Periodendauer / Impulsdauer und des Duty Cycles (Pausendauer) innerhalb einer Periodendauer kann die Ausgangsleistung quasi-stetig angepasst werden.



## Wir nehmen nun an der Heizstab sei stetig regelbar. Welchen universellen Regler könnten wir nun einsetzen? Erklären Sie die Formel, welche den Ein- und Ausgang miteinander in Verbindung bringt (5 Punkte)

Hierzu kann man einen PID-Regler mit Proportional, Integral- und Differential-Teil einsetzen. (1 Punkt)

Der Zusammenhang zwischen Ausgang und Eingang (meinst Regeldifferenz) kann mit einer der folgenden Formeln beschrieben werden: (1 Punkt)



(je 1 Punkt)

Proportional-Teil: Regelabweichung wird mit Faktor K\_i multipliziert

Differential -Teil: Änderungsrate der Regelabweichung über die Zeit wird mit Faktor K\_d oder Vorhaltzeit multipliziert

Integral -Teil: Interal der Regelabweichung über die Zeit wird mit Faktor K\_I oder Nachstellzeit multipliziert

## Angenommen die Regeldifferenz am Eingang folgt e(t)=t^2 und wir haben nun einen PD-Regler mit K\_P = 2 und T\_v =1/2, wie entwickelt sich u(t)? Skizzen Sie. (2 Punkt)

Proportional-Abteil ist 2\*t^2, Differenzialanteil ist (1/2)\* 2 t = t. Beides wird zur Antwort addiert.

1. **Steuerungen und Mikrocontroller**

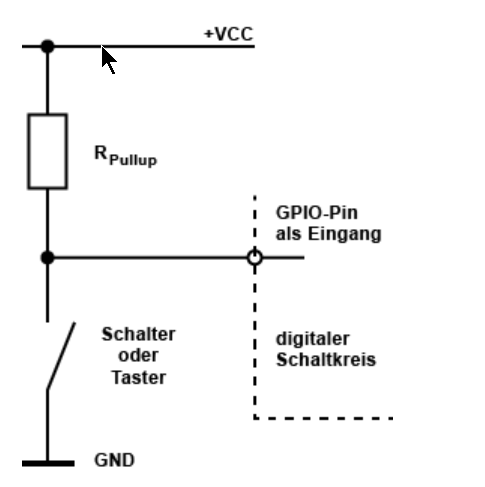
## Frage: Was versteht man unter einem Hauptprogramm eines Microcontrollers oder einer Speicher-Programmierbaren Steuerung? (1 Punkte)

Das Hauptprogramm ist das zentral ausgeführte Programm auf einem Microcontroller oder einer SPS. Es dient als Einstiegspunkt für alle weiteren Programme und wird i.d.R. Zyklisch / wiederholt abgearbeitet, um die Steuerungs- und Regelungsaufgaben zu erfüllen, vergleichbar mit der MAIN-Funktion bei Programmiersprachen wie TwinCat.

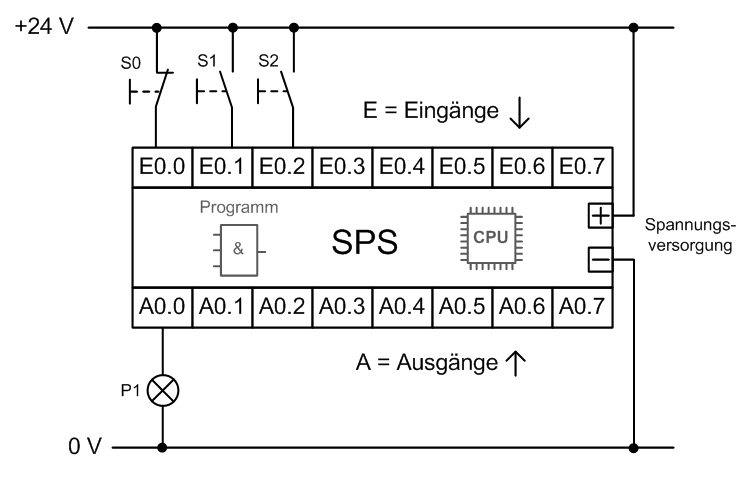
## Frage: Fertigen Sie eine technische Skizze einer Elektronische Schaltung an, wie man einen Taster an einen Digitalen Eingang eines Microcontrollers oder einer SPS anschließt, wenn hierbei ein Pull-Up-Widerstand genutzt wird. Wozu dient dieser? (5 Punkte)

Antwort:

Mit Pull-Up Widerstand (5 Punkte): Ein Taster (richtiges Symbol 1 Punkt) wird an einen digitalen Eingang angeschlossen, wobei ein Pull-Up Widerstand (richtiges Symbol 1 Punkt) verwendet wird, um sicherzustellen, dass der Eingang bei nicht gedrücktem Taster auf ein definiertes Potential (z.B. V+ auf 24 V) gezogen wird (Erklärung 1 Punkt). V- bzw. Ground und V+ bzw. VCC je ein Punkt.



Ohne Pull-Up Widerstand (3 Punkte): Alternativ kann der Taster ohne Pull-Up Widerstand angeschlossen werden, wenn der Microcontroller oder die SPS bereits interne Pull-Up Widerstände besitzt, die aktiviert werden können.



## Was ist die theoretische Definition eines Endlichen Automaten / einer Finite State Machine und wofür kann man diesen bei der Entwicklung von Steuerungen einsetzen? (2 Punkte)

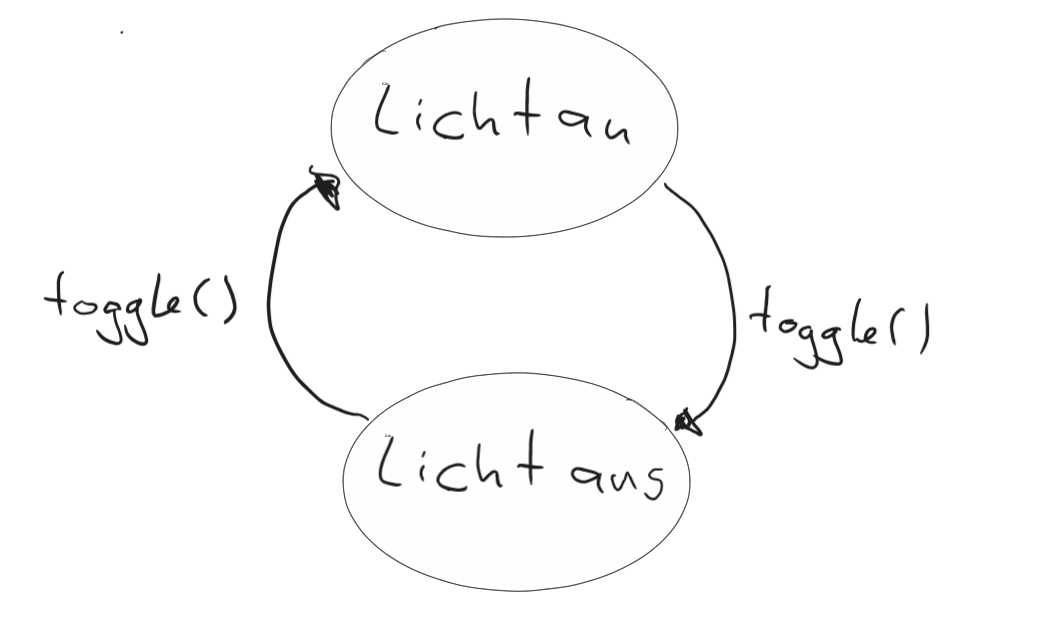
Ein endlicher Automat (Finite State Machine) ist ein Modell eines Systems mit einer endlichen Anzahl von Zuständen und der Übergänge zwischen diesen Zuständen. Es wird durch ein 5-Tupel definiert:

* endliche Zustandsmenge
* Startzustand
* endliches Eingabealphabet
* Endzustandsmenge
* Übergangsfunktion

Anwendungen finden sich in der Steuerung von Aufzügen, Automaten, Protokollen in Kommunikationssystemen und vielen anderen Bereichen, wo klare Zustandsübergänge benötigt werden. Durch die Modellierung mittels Finite State Machine kann die Implementierung erleichtert werden, zudem können klare Systemzustände definiert werden, was z.B. Fehler besser erkennbar macht.

## Frage: Zeichnen Sie eine Finite State Machine für eine Wechselschaltung mit zwei Schaltern A+B (1 Punkte)

Zustände an /aus und Übergänge Schalter A oder Schalter B (zusammengefasst als toggle() oder) oder zwei Übergangspfeile für jeden Schalter einen



1. **Bussysteme**

## Frage: Wonach lassen sich Bussysteme unterscheiden? (3 Punkte)

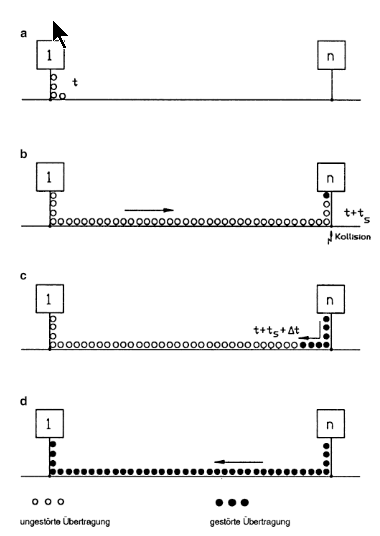
Je ein Punkt

* Telegramminhalt: Datenformate und Struktur der Nachrichten.
* Topologie: Netzwerkstruktur, z.B. Bus, Ring, Stern.
* Teilnehmerhierarchie: Master-Slave, Peer-to-Peer.
* Adressierung: Statische vs. dynamische Adressvergabe.
* Buszugriffsverfahren: deterministisch z.B. Token-Passing, zufällig z.B. CSMA/CD.
* Signalisierung: Elektrische, optische, drahtlose Signalisierung.
* Übertragungsmedium: Kupferkabel, Lichtwellenleiter, Funk.

## Wie funktioniert das Buszugriffsverfahren Carrier Sense / Multiple Access Colission Detection? Zu welcher Familie von Buszugriffsverfahren gehört es? Ist es echtzeitfähig? (3 Punkte)

CSMA ist ein zufälliges Buszugriffsverfahren, bei dem die Master versuchen auf die Leitung zuzugreifen und zu senden und in der Lage sind Buskollisionen zu erkennen. Beim Erkennen einer Kollision wird wer Sendeversuch zu späterem Zeitpunkt wiederholt. Damit ist das System nicht echtzeitfähig. Das bedeutet, dass keine Zeit garantiert werden kann, in der das System seine Nachricht spätestens senden kann.

## Wie werden Störungen vom Sender erkannt. Welche Aussage lässt sich über die physikalische Entfernung zwischen zwei Teilnehmern in einem Bussystem treffen, damit CSMA/CD funktionieren kann? (2 Punkte)

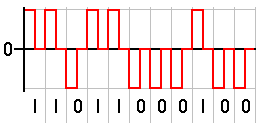
Ein sendender Teilnehmer vergleicht die Spannungen auf der Leitung mit seiner Nachricht. Wird die Sendung durch eine Kollision gestört, so entspricht das Signal (High / Low) auf der Leitung nicht mehr der Erwartungen. Der Abstand zwischen Sender und Empfänger darf deswegen nicht zu hoch werden, damit am Eingang des Senders gleichzeitig das zu sendende und störende Signal anliegen. 

## Berechnen Sie, wie lange das Signal am Sender anliegt, wenn folgende Annahmen gelten (2 Punkte):

* + - Leitungscode ist Return-to-Zero
    - Baudrate ist 16.000 Baud
    - Telegrammlänge ist 2 Byte

**Datenübertragungsrate (Bitrate):**

Da der Leitungscode Return-to-Zero verwendet wird, benötigt jedes Bit zwei Zustandswechsel, also wird die Bitrate halbiert:



Bitrate = Baudrate / 2 (Signale / Bit)

Bitrate = 16.000 Baud / 2 = 8.000 Bit/s

**Telegrammlänge in Übertragungszeit umrechnen:**

Ein Telegramm hat 2 Byte = 16 Bits, also:

Übertragungszeit eines Telegramms = Telegrammlänge / Bitrate

Übertragungszeit eines Telegramms = 16 Bits / 8.000 Bit/s = 0,002 Sekunden = 2 ms

1. **Programmierung (Optional)**

## Erklären Sie das Codebeispiel, wie diese Wechselschaltung objektorientiert Programmiert werden kann indem Sie Methoden und Attribute erklären. (4 Punkte)

import board  
led\_pin = board.GP1

class Wechselschaltung():

def \_\_init\_\_(self, led):

self.licht = false  
self.led = digitalio.DigitalInOut(led)  
self.led.direction = digitalio.Direction.OUTPUT

def toggle():

self.licht = not(self.licht)  
led.value = self.licht

Erklären von Methoden (2 Punkte)

\_\_init\_\_() ist Konstruktor

toggle()invertieren des Zustands

Erklären von Attributen (2 Punkte)

led ist ein Objekt, dass den mit der LED verbundenen Ausgang beschreibt

licht speichert den Zustand in einer Boolschen Variable

## Angenommen dieser Code steht im Hauptprogramm eines Microcontrollers. Was fehlt damit dieses funktioniert? (1 Punkt)

Es fehlt die Instanziierung des Objekts der Klasse Wechselschaltung (1 Punkt)

Ws = Wechselschaltung()

ws.toggle()

Es fehlt die Einbindung der zwei Buttons (1 Punkte)

button\_pin\_1 = board.GP0 # Replace with the GPIO pin connected to your button

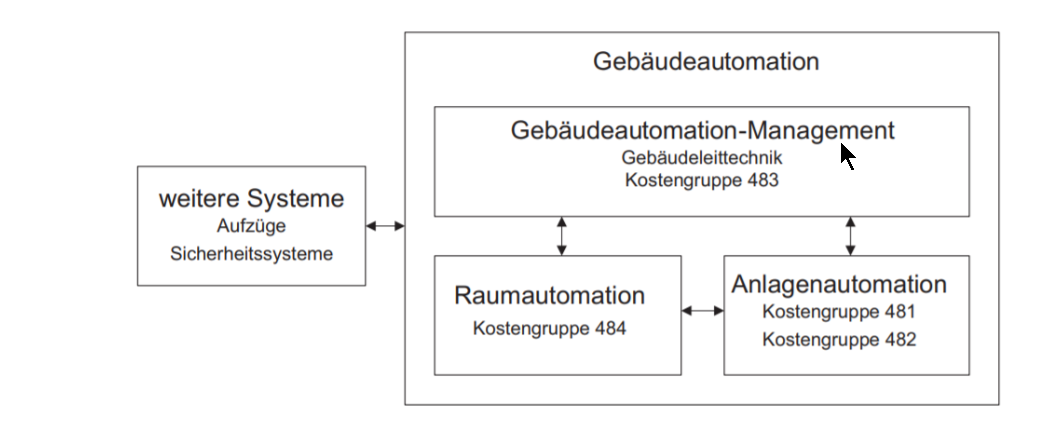
button\_1 = digitalio.DigitalInOut(button\_pin)

button\_1.direction = digitalio.Direction.INPUT

button\_1.pull = digitalio.Pull.UP # Use pull-up resistor; change if using pull-down

1. **Gebäudeautomation**

## In welche Bereiche teilt sich die Gebäudeautomation nach VDI 3814. Was sind die Aufgaben der einzelnen Teile (3 Punkte)



Anlagenautomation (1 Punkt): Hierunter fallen die Regelung, Steuerung, Prozessführung und Überwachung von Technikzentralen im Gebäude. Typische Anlagen sind Heizungsanlagen sowie Lüftungs-, Klima- und Kältetechnik. Diese Ebene kümmert sich um die zentralen technischen Prozesse im Gebäude.

Raumautomation (1 Punkt): Diese Ebene umfasst dezentrale Teilprozesse, die darauf abzielen, ein lokales Raumklima aufrechtzuerhalten. Dazu gehören die Steuerungen und Bedienung von Geräten wie Beleuchtung, Jalousien und Heizung in einzelnen Räumen.

Gebäudeautomation-Management (1 Punkt): Diese Ebene befasst sich mit übergeordneten Managementaufgaben und -funktionen. Hierzu zählen die Überwachung, Steuerung und Optimierung der gesamten Gebäudeautomation, oft durch eine zentrale Leittechnik.

## Frage: Wofür wird ein Raumautomations-Schema (nach VDI 3813) eingesetzt, Skizzieren Sie den Aufbau die Templates und erklären Sie dabei die Begriffe Funktion, Anlage, Raum, Segment, Bereich (4 Punkte)

In der Planung ermöglicht dies eine erste Zuordnung von Sensoren, Aktoren (Anlagen) und Funktionen zu Raum, Segment, Bereich (2 Punkte): Ein Raum (z.B. Treppenhaus) wird in Segmente unterteilt, z.B. Unterbereich eines Raums. Sensoren (Taster) und Aktoren (Leuchten) werden diesen Segmenten oder Raum zugeordnet, um zu bestimmen, wo und in welcher Stückzahl die Hardware installiert werden muss. Zudem wird über die Zuordnung klar, in welcher Granularität Funktionen implementiert werden müssen.



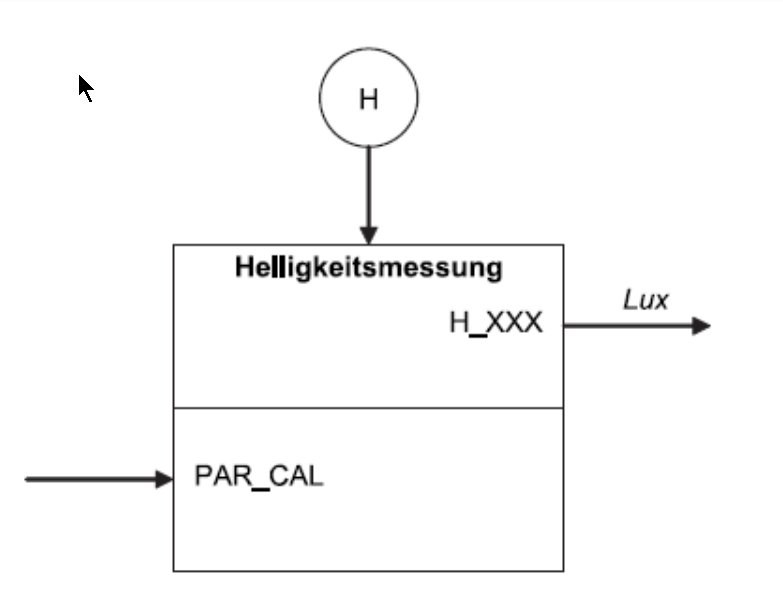
## Frage: Welche fünf Funktionstypen gibt es dabei? Skizzieren Sie für eines ein konkretes Beispiel in das Raumautomationsschema und definieren Sie die Begriffe Eingaben, Ausgaben und Parameter (6 Punkte)

2 Punkte für Nennung aller 5

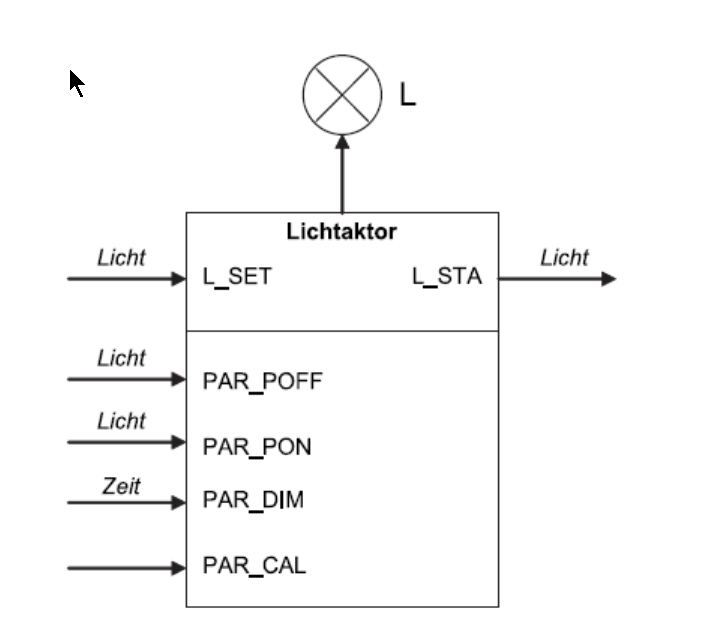
2 Punkte für Erklärung von Eingaben, Ausgaben und Parameter

2 Punkte für Richtige Zeichnung eines Beispiels

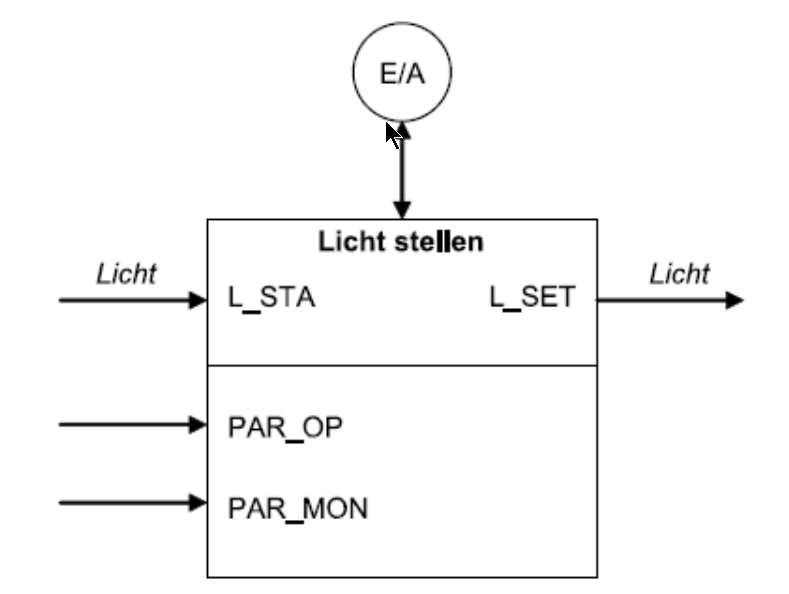
* Sensorfunktionen Ziel: Erfassen von Zuständen Sensorfuktionen != Sensoren / Datenpunkte, welche in die Zeile Anlagen eingetragen werden.



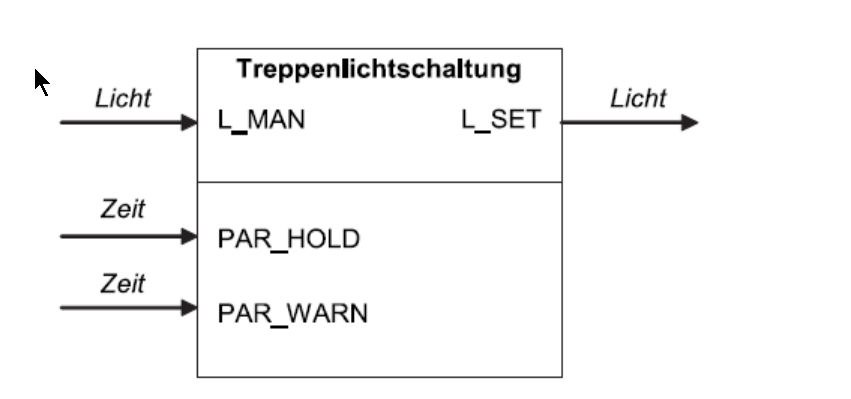
* Aktorfunktionen: Ziel: Steuerung von Komponenten, z.B. binärer Schaltaktor, Lichtaktor,



* Bedien-, Anzeige und Managementfunktionen: Bedien- und Anzeigefunktionen (lokal), Stellantriebsaktor, Sonnenschutzstellen, Antriebstellen, Temperatursollwertstellen, Raumnutzungsart wählen, Präsenzmelden



* Managementfunktionen: Aufzeichnung, Archivierung und statistische Analyse
* Anwendungsfunktionen: sind Programmabläufe die meist mehrere Aktoren und Sensoren miteinbeziehen



* Eingaben: Werte, die in die Funktion eingehen und sich abhängig vom Systemzustand ändern können
* Parameter: unveränderliche Werte, die einmalig konfiguriert werden (z.B. in Abhängigkeit der Raumgröße)
* Ausgaben: Werte, welche von der Funktion abhängig von Parametern und Eingaben zurückgegeben werden

