

Professur für Prozessleittechnik und Arbeitsgruppe Systemverfahrenstechnik

# Einzelsteuerfunktion und Templates als Basis für Masseneengineering

VL Prozessleittechnik 1  
Sommersemester 2020

# Lernziele



Nach dem Durcharbeiten der Präsentation kennen Sie die wesentlichen Prozessschritte und Methoden für den Entwurf und die Implementierung von **Einzelsteuerfunktionen** als Basis für ein effizientes Masseningeering.

Sie können im Besonderen:

- ... die notwendigen Schritte zur systematischen Strukturierung einer Anlage aus Automatisierungssicht und deren Ergebnisse wiedergeben
- ... typische Betriebsarten von Funktionsbausteinen verstehen und voneinander abgrenzen,
- ... die Funktionsweise und die Parameter der häufigsten Einzelsteuerfunktionen skizzieren.

# Agenda

## Funktionsbausteine

- Einzelsteuerfunktionen
- Analogwertverarbeitung
- Regler

## Massenengineering

- Templates
- Parameterlisten

# Motivation

## Herausforderung im Massen-Engineering

Anlagen der chemischen Prozessindustrie bestehen aus automatisierungstechnischer Sicht aus einer Vielzahl von funktionalen Elementen:

Individuelle Parameter:

- Viele I/O Signale
- Verriegelungen
- Regelfunktionen

Wiederkehrende funktionale Objekte:

- Messgeräte (z.B. Temperaturmessung)
- Ventile (z.B. Zuflussventile)
- Motoren ( z.B. Rührer, Pumpen)

**Muss jedes wiederkehrende Objekt und dessen Instanzen jedes mal neu angelegt werden?**



Abbildung : Laboranlage der TU Dresden als Übungsfeld für moderne Prozessleittechnik

# Massenengineering

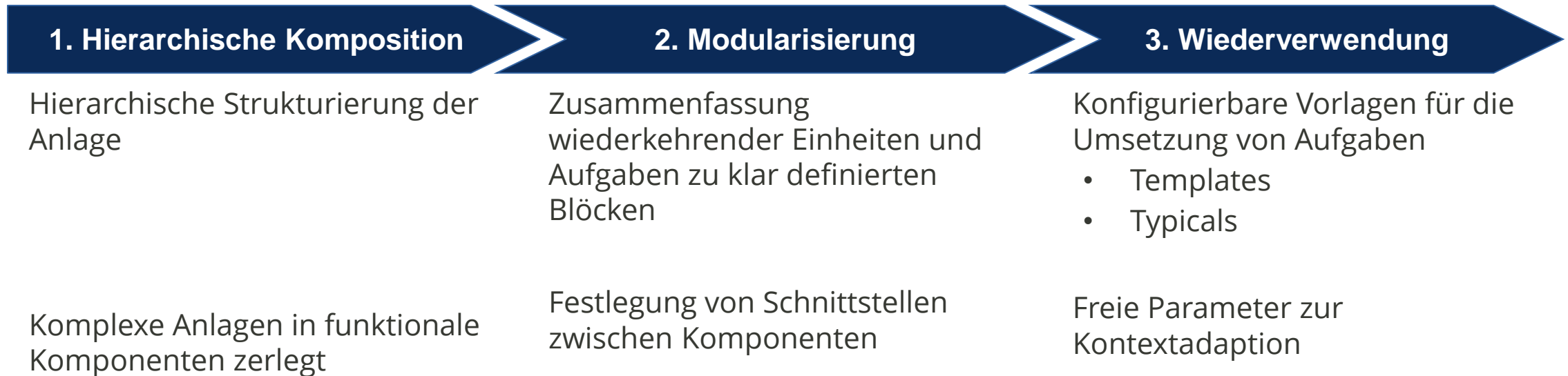
Komponenten der Steuerungsprogramme sollen wiederverwendbar sein

- Bereitstellung von Steuerfunktionen (FBS) durch Hersteller für komplexe der anspruchsvolle Baugruppen und Systeme
- Verwertung firmeneigener Entwicklungen/IP in späteren Projekten

Projektplanung sollen durchgehend und wiederverwendbar sein

- Gliederung des Projektes in Ebenen:
  - **Signalebene** ( Signale A1T1L001.LSA+, A1T2T001.PV, A1T1S001, ...)
  - **Logikebene** (Überlaufsicherung der Tanks, Trockenlaufsicherung der Pumpen)
  - **Prozesseinheitsebene** (identische Reaktoren, identische Tanks)
- Unterstützung durch CAE-Tools

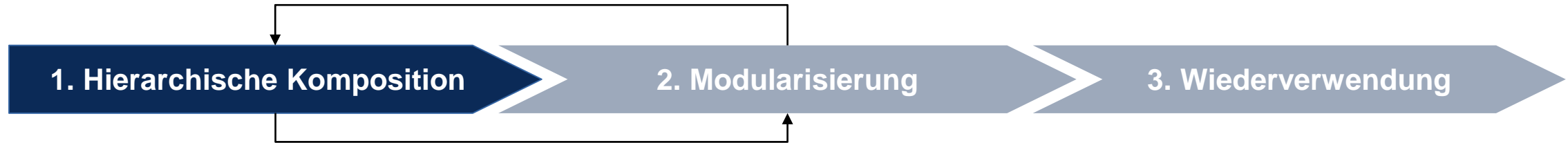
# Strukturierungsprinzipien Übersicht



Kapitel 7.1, S.93: Allgemeine Prinzipien der Strukturierung

# Strukturierungsprinzipien

## Hierarchische Komposition



- Funktionale Ebenen-Kategorisierung einer Aufgabe oder eines Elementes
- Merkmale guter hierarchischer Kompositionen:
  - **Zweck:** Der Zweck der Elemente ist deutlich erkennbar und verstehbar
  - **Autonomie:** Die Elemente können möglichst weitgehend autonom geprüft (design-time) bzw. überwacht (run-time) werden
  - **Beziehungen:** Die Menge der Beziehungen zu und Abhängigkeiten von anderen Elementen ist überschaubar
  - **Schnittstellen:** Die Schnittstellen zu anderen Elementen sind sauber definiert
- Unmittelbare Wechselwirkung mit Modularisierung:
  - Atome der hierarchischen Komposition sind Module

# Strukturierungsprinzipien

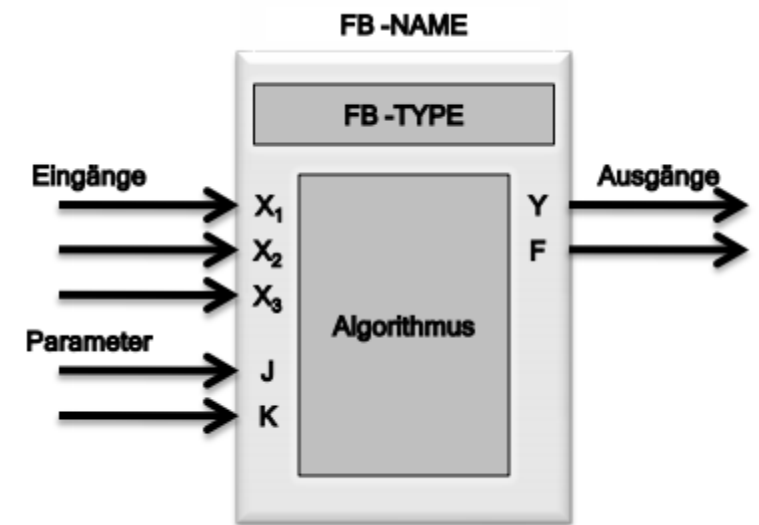
## Modularisierung

1. Hierarchische Komposition

2. Modularisierung

3. Wiederverwendung

- Unterteilung einer Anlage und deren Software in weitgehend entkoppelte Sachverhalte oder Steuerbereiche,
  - Maximierung des Wiederverwendungswertes der Software
- Kapselung der Funktionalität in modulare Software  
Beispiele:
  - Motoransteuerungs-FUB
  - Analogwertverarbeitung
  - Reglerstrukturen (PI, PID...)



# Strukturierungsprinzipien

## Wiederverwendung und Templating

### 1. Hierarchische Komposition

### 2. Modularisierung

### 3. Wiederverwendung

- Erhöhung des Wiederverwendungspotenzials modularer Softwarekomponenten
- Parametrisierung
  - Bausteine können über Parameter auf den Sachverhalt angepasst werden
  - Entwicklung möglichst generischer Bausteine
  - Nachteil: Parameter oft schwer zu verstehen, viele Ports
- Templating
  - Software wird aus einer Vorlage hergeleitet (*Design-Time*)
  - Generiert eine spezialisierte Instanz, bsp. durch Code-Generatoren
  - Einfache (oft visuell unterstützte) Parametrierung, wenig Ports, aber Aufwand für Templatesystem



Kapitel 7.4, S.99: Massenbearbeitung

# Einzelsteuerfunktionen



Kapitel 8, S.110: Einzelsteuerfunktionen

# Einzelsteuerfunktionen

## Grundsätzlicher Aufbau

- Hierarchische Komposition
  - Pro Feldgeräte-Typ ein Funktionsbaustein-Typ
  - Baustein beinhaltet Signalebene, Modulebene
- Wiederverwendung:
  - Parameter-Eingänge für kontextualisierung der Funktionalität
  - Mehrere, ähnliche Funktionsbausteine mit unterschiedlichen Nutzungsszenarien
  - Hinterlegung mehrerer Funktionsbausteine in einer Bibliothek zur freien Auswahl durch den Nutzer
    - Inklusive Dokumentation
- Anwenderprogramm nutzt die Einzelsteuerfunktion zum Realisieren der gewünschten Steuerung für einen Prozess

# Bereitstellung von Einzelsteuerfunktionen

## Beispiel Bibliotheksinhalte Siemens APL

Einzelsteuerfunktion	Anwendungsbereich	Objektname
<b>MotL</b>	Ansteuerung von Motoren mit einem Steuersignal (ein/aus) und einer Laufrückmeldung	FB 1850
<b>MotRevL</b>	Ansteuerung von Motoren mit zwei Drehrichtungen (Linkslauf/Rechtslauf) und maximal zwei Rückmeldungen	FB 1851
<b>MotSpdL</b>	Ansteuerung von Motoren mit zwei Geschwindigkeiten (langsam/schnell) und maximal zwei Rückmeldungen	FB 1856
<b>VlvL</b>	Ansteuerung von Steuerventilen mit einem Steuersignal (öffnen/schließen) und Stellungsrückmeldung (offen/geschlossen)	FB 1899
<b>VlvMotL</b>	Ansteuerung von Motorventilen mit zwei Steuersignalen und Stellungsrückmeldung (offen/geschlossen).	FB 1900

# Betriebsarten für Einzelsteuerfunktionen

- Übliche Betriebsarten für Einzelsteuerfunktionen
  - Außer Betrieb:           Wartung, Gerät deinstalliert oder nicht funktionstüchtig
  - Handbetrieb:            Einzelne Parameter des Funktionsbausteins werden manuell festgelegt (über HMI, Leitwarte)
  - Vor-Ort-Betrieb:        Wie Handbetrieb, manuelle Steuerung über Diagnosegerät
  - Automatikbetrieb:    Einbindung in Steuerfunktionen der Anlage
- Immer nur eine Betriebsart zulässig
- Betriebsart wird dem Operator angezeigt & erfordert dessen Freigabe



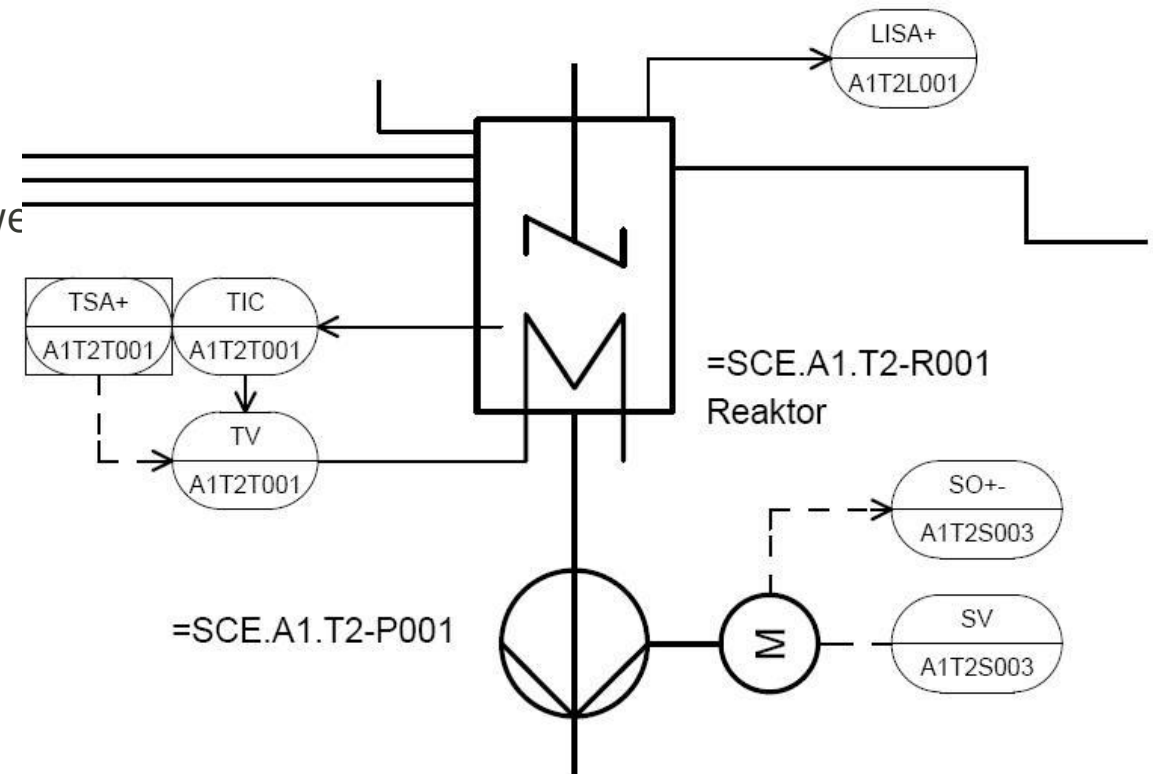
# Exemplarische Einzelsteuerfunktionen

## Motorantrieb

- Antriebskontrolle für einen Motor (der Pumpe P001)

Aufgaben des Funktionsbausteines:

- Bedienen, Beobachten, Melden
  - Bedienen und Beobachten von Prozess- und Sollwert
- Steuerung von Signalen
  - Steuersignalausgabe (statisch, gepulst)
  - Wertvorgabe
- Überwachung
  - Warnungen/Alarmer bei Grenzwertverletzungen



# Exemplarische Einzelsteuerfunktionen Motorantrieb (cont.)

- **Verriegelungsfunktionen**

- Einschaltfreigabe, Verriegelung mit/ohne Rücksetzen
- Motorschutzfunktion
- Schnellstopp (höchste Priorität)

- **Betriebszustandsauswahl**

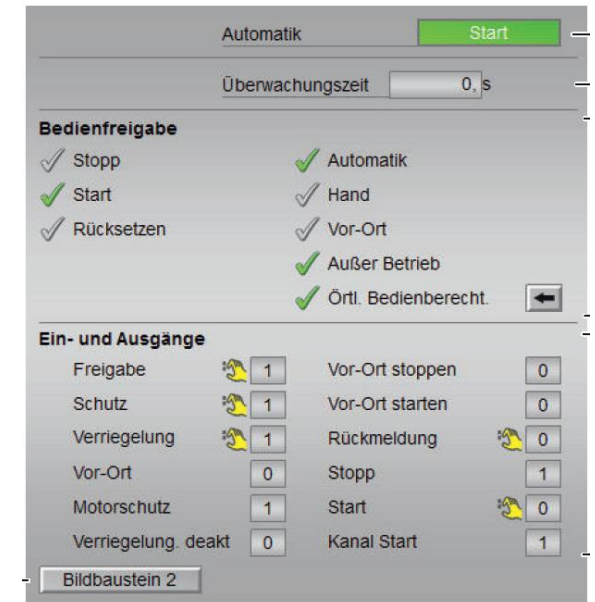
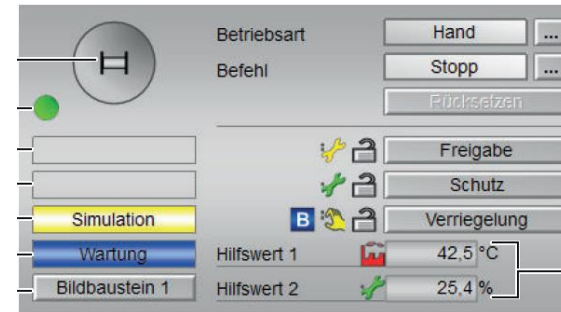
- Genannte Betriebszustände in allen Einzelsteuerfunktionen von PCS 7 verfügbar
- Zusätzliche Force-Funktion

Bausteinbezeichner	→	Pumpe_A1T2S003	
Bausteintyp	→	MotL	OB35
		Motor -	1/1
		0 StartAut	MS_Relea
		0 StopAut	MonDynEr
Laufrückmeldung	→	0 ModLiOp	MonStaEr
		0 AutModLi	R_StpAct
		0 ManModLi	LockAct
		0 LocalLi	GrpErr
		0 OosLi	RdyToSta
		0 StartLoc	RdyToRes
		0 StopLoca	WarnAct
		0 FbkRun	Start
		1 Monitor	P_Start
		3.0 MonTiSta	P_Stop
		3.0 MonTiDyn	LocalAct
		0 RapidStp	AutAct
		0 RstLi	ManAct
		1 Trip	OosAct
		1 Permit	FbkRunOu
		1 Intlock	AV_Out
		1 Protect	AV_OpSca
		0 StartForc	ErrorNum
		0 StopForc	
		0 CSF	
		AV	
		16#0 EventTsI	

← Steuerausgang

# Exemplarische Einzelsteuerfunktionen Motorantrieb (cont.)

- HMI Elemente für verschiedene Betriebsarten
  - Oft Bestandteil der Bibliotheken

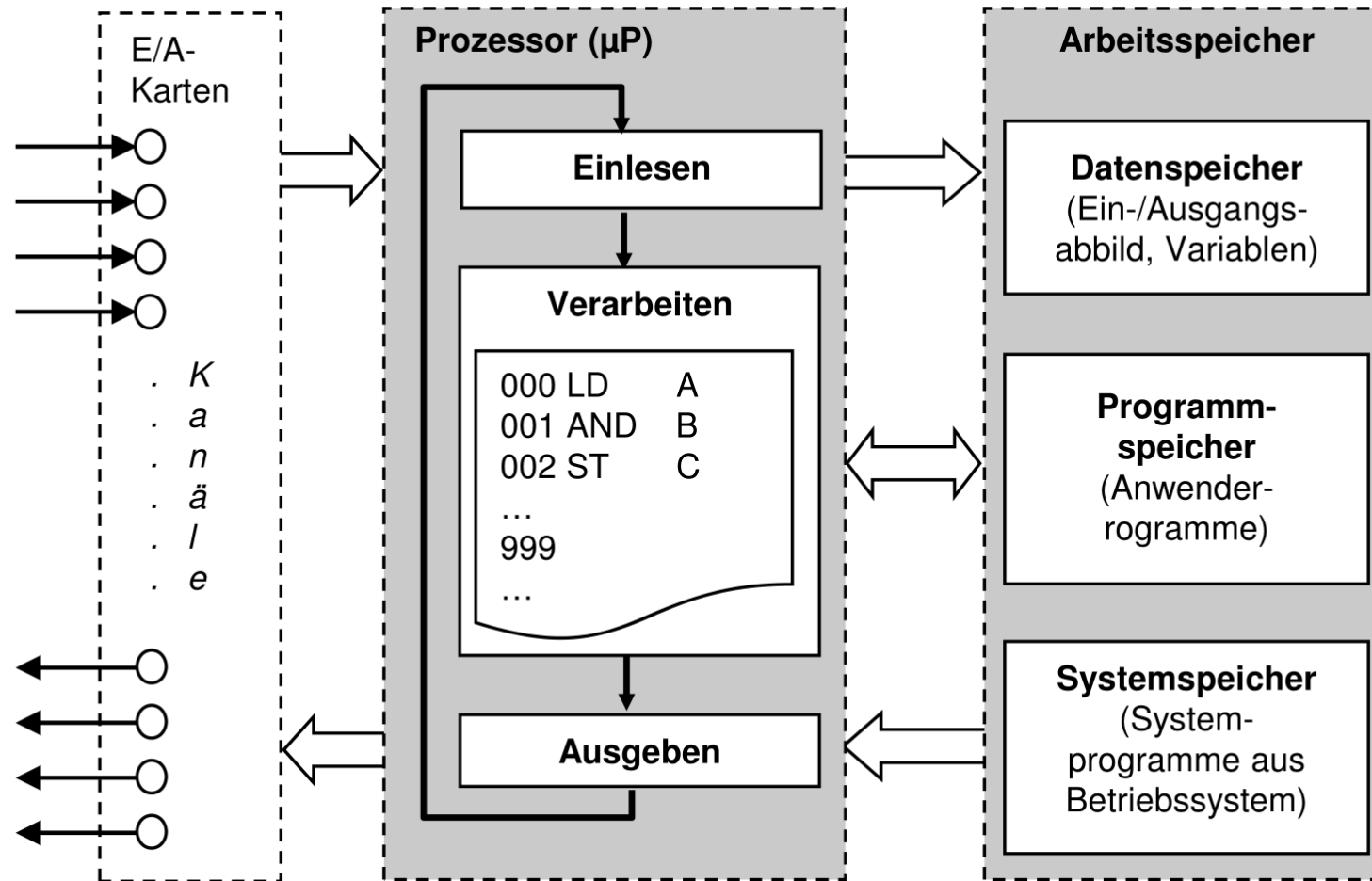


# Analogwertverarbeitung

# Analoge Steuersignale

- Physikalische (kontinuierliche) Größen, repräsentiert durch Strom- oder Spannungswert
  - Füllstand
  - Temperatur
  - Geschwindigkeit
  - Drehzahl
- Häufig unter Einsatz eines Messumformers (Skalierung, Offset des Messsignals)
- Erfassung der Messgröße in SPS benötigt **Analogeingabebaustein**

# Signalverarbeitung SPS





# Analog-Digital-Wandung

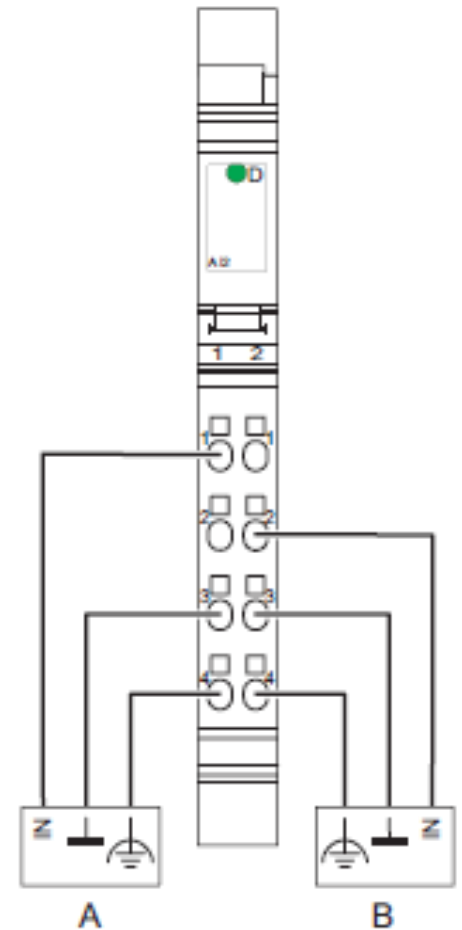
## Eigenschaften ADC

- Messung bezieht sich auf Spannungspegel
  - Strommessung als Spannungsabfall über präzisiertem Widerstand
    - evtl. Messverstärker-Schaltung vor ADC für geringe Spannungen
  - Herstellerabhängig: Strom/Spannungs/Temperatur/Widerstand-Messmodi
- Digitales Ausgabewort des ADCs
  - N bit Wort → Messbereich  $0..2^N-1$
  - Minimum „0“ →  $V_{\text{mess}} == V_{\text{ADC-}}$
  - Maximum „ $2^N-1$ “ →  $V_{\text{mess}} == V_{\text{ADC+}}$

Referenzspannungen müssen gut reguliert und gegen Störungseinflüsse gesichert sein

- Allgemeine Rechenvorschrift:

$$ADC_{\text{Value}} = \frac{(2^N - 1) \cdot V_{\text{Mess}}}{V_{\text{ADC+}} - V_{\text{ADC-}}}$$



# Messsignale in industriellen Systemen

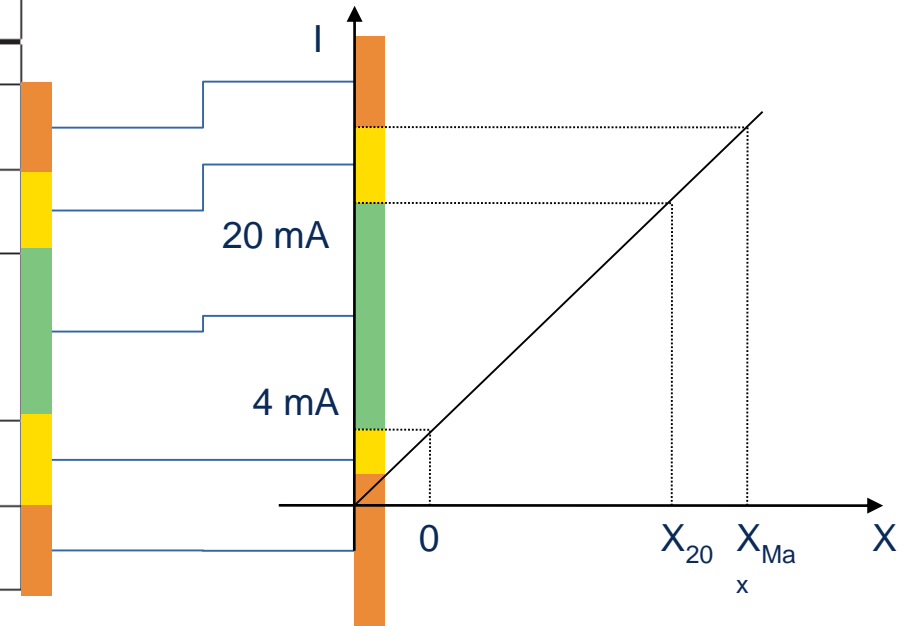
## 4-20mA

- Stark verbreitete, Strombasierte Mess- und Steuergröße für industrielle Systeme
  - Erlaubt Bus-Powered Sensorik (2-Draht)
  - Zusätzliche Leitungsmodulation (Bell202) für digitale Datenübertragung → HART
- Wird sowohl als Steuersignal wie auch als Sensorsignal verwendet
- Extremwerte<sup>1</sup>:
  - $< 3.6 \text{ mA}$ : Interpretation als „Drahtbruch“
  - $> 20.5 \text{ mA}$ : Interpretation als „Kurzschluss“
- 2-wire oder 4-wire Schnittstellen
  - 4 Kabel: Bestimmung der Verlustleistung durch lange Kupferleitungen

<sup>1</sup>) gemäß NAMUR NE43

# Analog-Digital-Wandung Wertekonvertierung

System		Strommessbereich		
dez.	hex.	0 bis 20 mA	4 bis 20 mA	
32767	7FFF	23,70 mA	22,96 mA	Überlauf
32512	7F00			
32511	7EFF	23,52 mA	22,81 mA	Übersteuerungsbereich
27649	6C01			
27648	6C00	20 mA	20 mA	Nennbereich
20736	5100	15 mA	16 mA	
1	1	723,4 nA	4 mA + 578,7 nA	
0	0	0 mA	4 mA	
-1	FFFF			Untersteuerungsbereich
-4864	ED00	-3,52 mA	1,185 mA	
-4865	ECFF			Unterlauf
-32768	8000			



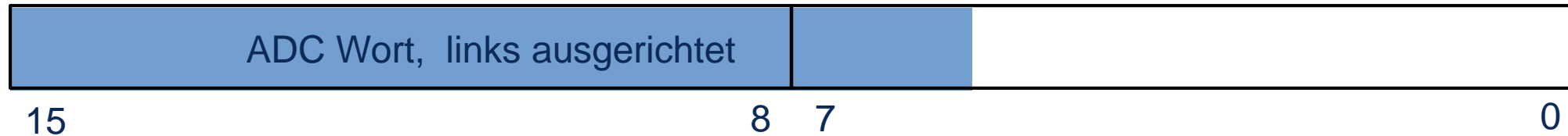
Siemens AG, "Automatisierungssystem S7-300 Baugruppendaten: Gerätehandbuch, 02/2013, A5E00105504-08"  
S. 272

$$X_W = \frac{I_E - 4mA}{16mA} * \boxed{27648}$$

# Analog-Digital-Wandung

## Eigenschaften gewandelter Datenworte

- Wortausrichtung: Meistens 16-Bit Register (INT, WORD), ADC-Wert mit geringerer Breite



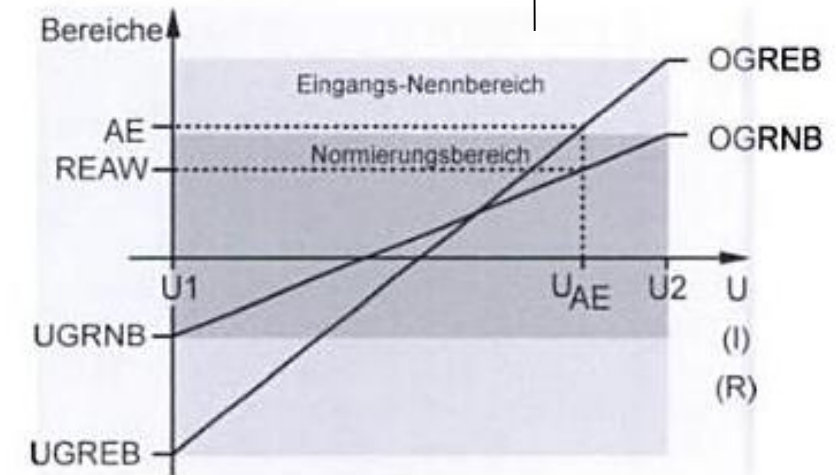
- Parametrisierung:
  - Auflösung, Wortausrichtung
  - Sample-and-Hold-Zeit
    - Höhere Zeiten → Störunterdrückung, niedrige zeitl. Auflösung
  - Zuschaltbare (Hoch-, Tiefpass-) Filter

# Normierung von ADC-Werten

## Normierungsverfahren

- Ursprüngliche Messgröße geht durch
  - 4-20 mA Transmissionsverfahren
  - ADC-Wertdarstellung
- verloren. Normierung des dig. Rohsignals notwendig.
  - AE: Digitalisierter Eingangswert
  - REAW: Normierter Analogeingabewert
  - OGREB, UGREB: Grenzen Eingangsnennbereich
  - **OGRNB, UGRNB: Grenzen Normierungsbereich**
  - U1,U2: Spannungsmessbereich
  - UAE: Anliegende Spannung

Physikalischer Messwert [m³/h]	Ausgang des Sensors [mA]	Digitaler Eingangsnennwert AE_Nenn	Normierter Wert AE_Norm
0	4	0	0
500	20	27648	500

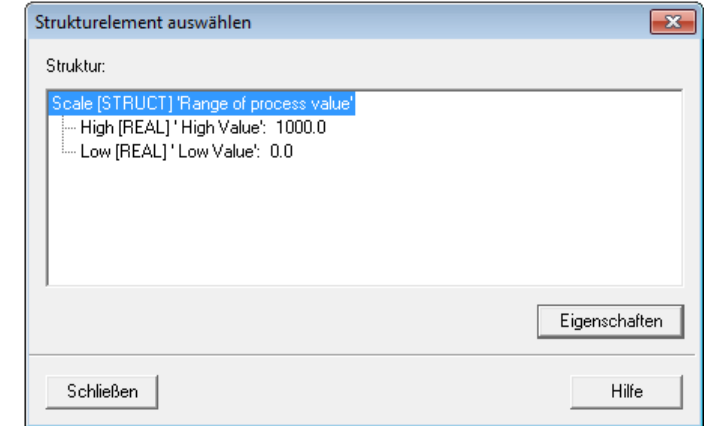
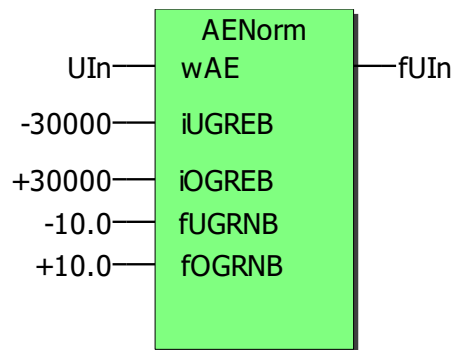


$$REAW = UGRNB + (AE - UGREB) \frac{\Delta GRNB}{\Delta GREB}$$

# Normierung von ADC-Werten

## Einzelsteuerfunktion in PCS7

- Normierung implementierbar als (speicherlose) Funktion
  - Wiederverwendbare Einzelsteuerfunktion
- Schnelle Parametrierung des Bausteins über graphischen Assistenzdialog
  - Konfiguration einer Template



Füllstand	
Pcs7AnIn	
Analog i	OB35
	1/1
16#0	PV_In
1000.0	Scale
1040	PV_InUni
21.5	HighLimi
3.3	LowLimit
0	SimOn
0.0	SimPV_In
0.0	SubsPV_I
0	MS_Relea
16#0	MS
16#0	Mode
16#0	DataXchg

# Normierung von ADC-Werten

## Einzelsteuerfunktion in AWL

- Funktion (ohne Outputvariablen)
  - 3 lokale Variablen

### Benötigter Sprachumfang für diese AE

```
LD op (* vke := op *)
ST op (* op := vke *)
ADD op (* vke := vke + op *)
SUB op (* vke := vke - op *)
MUL op (* vke := vke * op *)
DIV op (* vke := vke / op *)
INT_TO_REAL (* vke := float(vke) *)
WORD_TO_REAL (*vke := float(vke) *)
```

AENorm

wAE

iUGREB

iOGREB

fUGRNB

fOGRNB

fUIn

-30000

+30000

-10.0

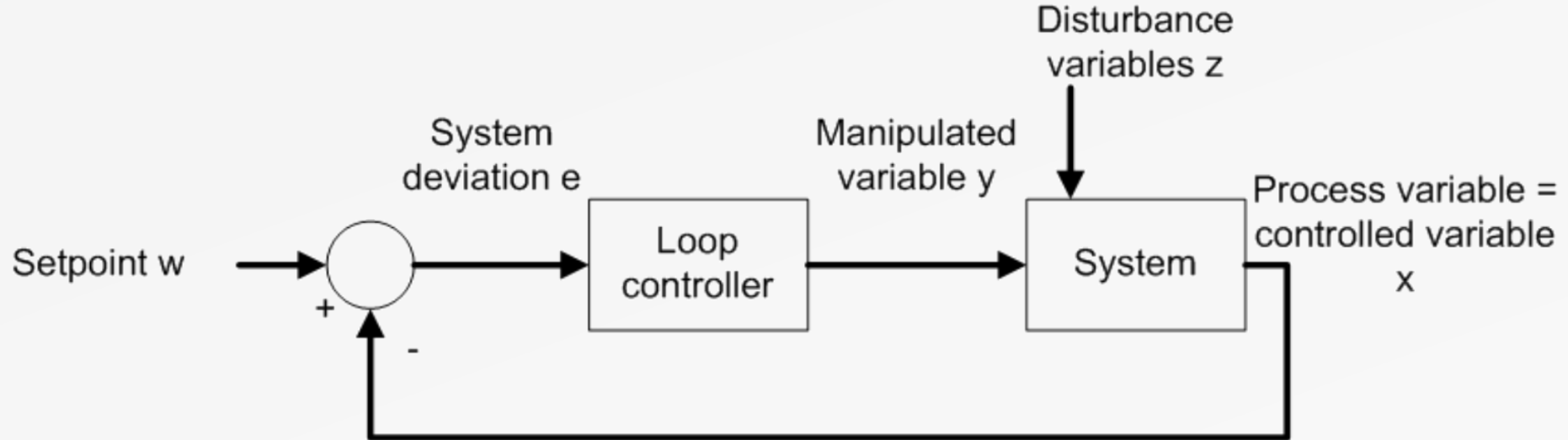
+10.0

Name	Typ	Verwendung	Beschreibung
<b>Default</b>			
fUGRNB	REAL	VAR_INPUT	Untergrenze Normierungsbereich
fOGRNB	REAL	VAR_INPUT	Obergrenze Normierungsbereich
iOGREB	INT	VAR_INPUT	Obergrenze Eingangsnennbereich
iUGREB	INT	VAR_INPUT	Untergrenze Eingangsnennbereich
wAE	WORD	VAR_INPUT	Rohrer digitalisierter Analogwert
fDGRNB	REAL	VAR	Range Normierungsbereich
fDGREB	REAL	VAR	Range Eingangs-Nennbereich
fUGREB	REAL	VAR	Untergrenze Eingangs-Nennbereich

```

AENorm: AENorm - STD_CNF.STD_RES.FUNCTIONS.AENorm.AENorm
1  LD iUGREB          (* Konvertiere iUGREB *)
2  INT_TO_REAL
3  ST fUGREB
4
5  LD iOGREB          (* Konvertiere iOGREB *)
6  INT_TO_REAL
7  SUB fUGREB         (* fDGREB <- (f)OGREB - fUGREB *)
8  ST fDGREB
9
10 LD fOGRNB          (* fDGRNB <- fGRNBR - fGRNBR *)
11 SUB fUGRNB
12 ST fDGRNB
13
14 (* fUGRBNR + ( (f)AE - fUGREB ) * fDGRNB / fDGREB *)
15 LD wAE             (* Konvertiere AE *)
16 WORD_TO_REAL
17 SUB fUGREB
18 MUL fDGRNB
19 DIV fDGREB
20 ADD fUGRNB
21 ST AENorm         (* Speichere Ergebnis (entspr. Return) *)
    
```



# Regler



Kapitel 9, S.122: Regelungen

# Kontinuierliche Regler

## PI, PD, PID Regler



### Zeitkontinuierliche Regler

- Jede digitale Implementierung arbeitet zeitlich diskret !!
- Diskretisierung hat kritischen Einfluss auf die Reglerstabilität !

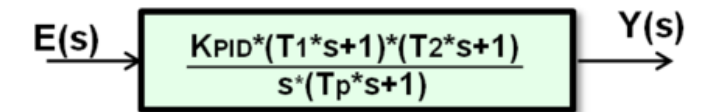
### Parametrierung (PID)

- Verstärkung (Gain): **KP**
- Nachstellzeit: **TN**
- Vorhaltezeit: **TV** (+ TP für realisierbare Implementierung)

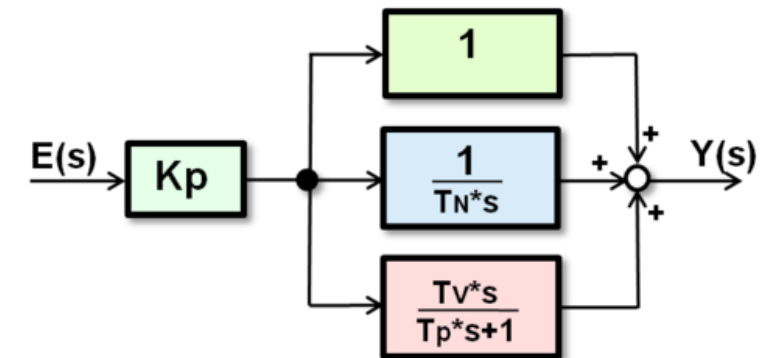
### Weitere Parameter zur Prozessanbindung

- Modus (AUTO/MANUELL)
- Interner/Externer Sollwert
- Begrenzungen
- Anti-Windup

2	CTRL_PID	
	PID Cont	0B35
		2/2
0.0	DEADB_W	QSPEXTON
1.0	GAIN	QLMN_HLM
10.0	TN	QLMN_LLM
0.0	TV	QMAN_AUT
1.0	TM_LAG	QCAS_CUT
100.0	NM_LMHR	LMN
0.0	NM_LMHLR	QC_LMN
0.0	PV_IN	ER
16#80	QC_PV_IN	SP
100.0	NM_PVHR	
0.0	NM_PVLR	
0.0	DISV	
0	CSF	
no Track	SP_TRK_0	
0.0	SP_EXT	
100.0	SPEXTHLM	
0.0	SPEXTLLM	
0	LIOP_INT	
0	SPEXON_L	
100.0	LMN_HLM	
0.0	LMN_LLM	
0.0	LMN_TRK	
0	LMN_SEL	
0	LIOP_MAN	
0	AUT_L	
0.0	LMNR_IN	
16#80	QC_LMNR	
16#80	QC_LMN_I	



Realer PID-Regler in Reihenstruktur



Realer PID-Regler in Parallelstruktur

Die realen PID-Regler in Reihen- und Parallelstruktur verhalten sich nahezu identisch, wenn  $T_p < 5\%$  von der kleineren der beiden Zeitkonstanten  $T_1$  oder  $T_2$  gewählt wird.

The original Uploader, "PID-Regler in Reihen- und Parallelstruktur", Wikimedia Commons

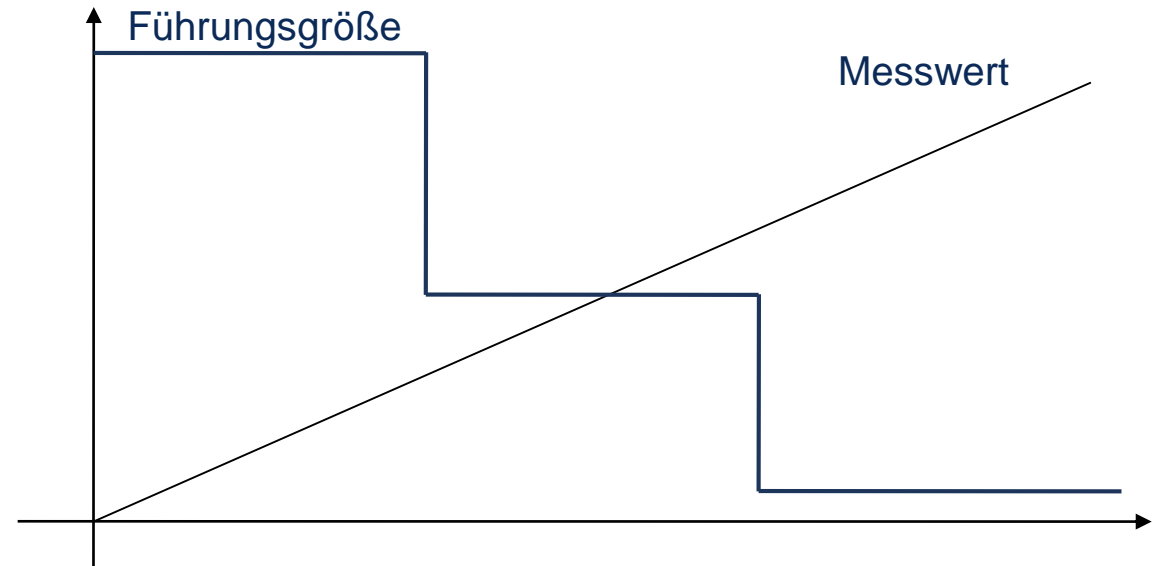
# Wertdiskrete Regler: Zwei- und Dreipunktregler

Regelsystem mit 2 (3) Zuständen

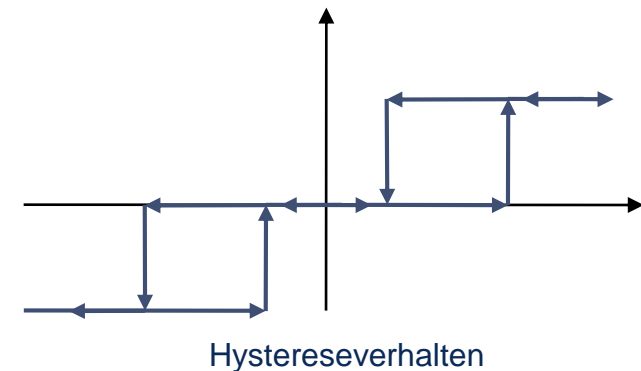
- AN
- (IDLE)
- AUS

Parametrisierung

- Führungswerte für AN, AUS, IDLE
- Hysterese-Verhalten
  - Aufschaltpunkte
  - Hysteresebreite / Deadband



3	SPLITR_P	0B35
	Split ra	2/3
100.0	UHRANGE	QV1
0.0	ULRANGE	QV2
0.0	U	V1
50.0	NEUT_POS	V2
10.0	DEADB_W	
100.0	V1HRANGE	
0.0	V1LRANGE	
100.0	V2HRANGE	
0.0	V2LRANGE	



# Zusammenfassung

Massenengineering erfordert wiederverwendbare POEs

- Umsetzung als Funktionsbausteine, Funktionen

i.d.R. ein Baustein pro Feldgerätetyp

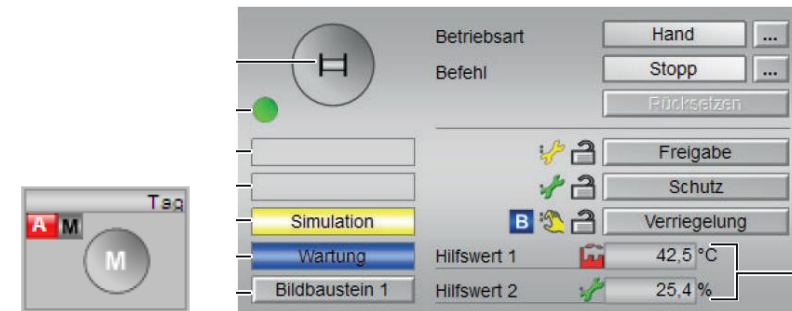
- Modularisiert: Mehrere, ähnliche Baublöcken für bestimmte Aufgabenklassen
- Wiederverwendbar: Parametrierung der Bausteine

Parameter

- Typische Fahrweisen
- Interlock und Simulationsanbindung möglich

Einzelsteuerfunktionen sind typischer Bestandteil von Herstellerbibliotheken

- Beinhalten Dokumentation des Bausteins
- Beinhalten häufig auch grafische Elemente zur Darstellung in HMIs
  - Symbole für die Darstellung in Fließbilder (Mimics)
  - Faceblades zur (Hand-)Bedienung der Parameter



# Lernzielabfrage



1. Welche drei Schritte sind notwendig, um eine Anlage für das Massenengineering vorzubereiten?
2. Welche Betriebsarten haben sich für Einzelsteuerfunktionen etabliert?
3. Welche drei (bis vier) Parameter konfigurieren das zeitliche Verhalten eines kontinuierliche PID-Regelbausteins?
4. Welche grundlegenden Parameter sind für elektrische Antriebe wichtig?
5. Wie funktioniert das Massenengineering mit Einzelsteuerfunktionen im allgemeinen und im besonderen in PCS7 (Buch Kapitel 7.4, Seiten 99-109)

# Vielen Dank für Ihre Aufmerksamkeit

Ich stehe Ihnen für Rückfragen zur Verfügung:  
[leon.urbas@tu-dresden.de](mailto:leon.urbas@tu-dresden.de)

[LV PLT1 auf OPAL](#)

