

Automatisierungstechnik

1 Einführung

Automatisierung: Reale Aufgaben der Automatisierungstechnik sind im Allgemeinen sehr komplex. Als umfassender Ausdruck für Steuerungs-, Regelungs- und Visualisierungs-Vorgänge hat sich der Begriff der Automatisierung durchgesetzt. Er beinhaltet, dass Automatisierungsgeräte selbsttätig Programme befolgen und dabei Entscheidungen auf Grund vorgegebener Führungsgrößen und rückgeführter Prozessgrößen aus der Anlage sowie erforderlicher Daten aus internen Speichern des Systems treffen, um daraus notwendige Ausgangsgrößen für den Betriebsprozess zu bilden.

Steuerung: Steuern oder Steuerung wird als Ablauf in einem System definiert, bei dem eine oder mehrere Eingangsgrößen andere Größen als Ausgangsgrößen aufgrund der dem System eigentümlichen Gesetzmäßigkeiten beeinflussen. Kennzeichen für das Steuern ist der *offene Wirkungsablauf über der Steuerstrecke*. Eine Steuerung liegt also vor, wenn Eingangsgrößen nach einer festgelegten Gesetzmäßigkeit Ausgangsgrößen beeinflussen. Die Auswirkung einer nicht vorhersehbaren Störgröße wird nicht ausgeglichen.

Regelung: Immer dann, wenn Störgrößenänderungen das System nicht hinnehmbar beeinflussen können, werden Regelungen erforderlich. Die Regelgröße (Aufgabengröße) muss sich messtechnisch erfassen lassen, denn eine Regelung ist ein Vorgang, bei dem die Regelgröße fortlaufend erfasst, mit der Führungsgröße verglichen und abhängig vom Ergebnis dieses Vergleichs im Sinne einer Angleichung an die Führungsgröße beeinflusst wird. Der sich dabei ergebende

Wirkungsablauf findet in einem geschlossenen Kreis, dem *Regelkreis*, statt. Programmierbare Automatisierungsgeräte können die Aufgaben des Steuerns und Regeln ausführen, denn beide Funktionen beruhen auf Programmen unter Verwendung desselben Operationsvorrates.

2 Automatisierungsgeräte

Die derzeit am weitesten verbreitete Hardware-Plattform der Steuerungstechnik ist die *Speicherprogrammierbare Steuerung SPS*, wie sie in Bild 1 abgebildet ist, dort jedoch ohne den heute schon üblichen Anschluss an ein Feldbussystem zur Vernetzung mit anderen Steuerungskomponenten.

Eine Speicherprogrammierbare Steuerung hat die Struktur eines Rechners, deren Funktion als Programm gespeichert ist. Sie besteht im einfachsten Fall aus einer *Stromversorgung PS*, einem *Steuerungsprozessor CPU*, einigen zentralen *digitalen Eingabe- und Ausgabebaugruppen* sowie einem internen Bussystem. Bei Bedarf können auch Baugruppen zur *Analogwertverarbeitung* oder für besondere Funktionen wie *Regler*, schnelle *Zähler* und *Positionierungen* hinzukommen. Die Peripheriebaugruppen und die Programmiersprachen sind auf die Belange der Steuerungstechnik ausgerichtet. Speicherprogrammierbare Steuerungen gibt es als modulare und kompakte Systeme für unterschiedliche Anforderungsniveaus. Ein typisches Merkmal von SPS-Steuerungen ist die zyklische Programmbearbeitung, wie in Bild 2 angedeutet.

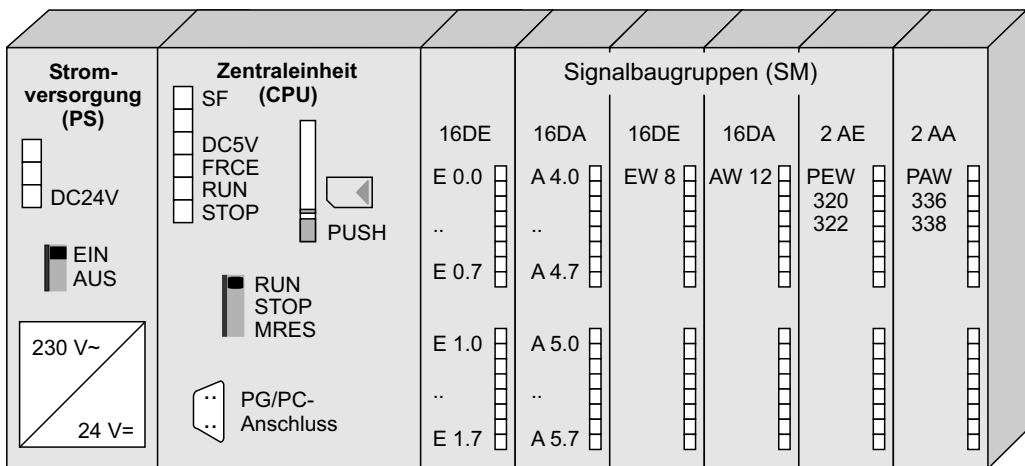


Bild 1 Aufbau einer Speicherprogrammierbaren Steuerung ohne Feldbusanschluss

DE = Digitale Eingänge, DA = Digitale Ausgänge, AE = Analoge Eingänge, AA = Analoge Ausgänge

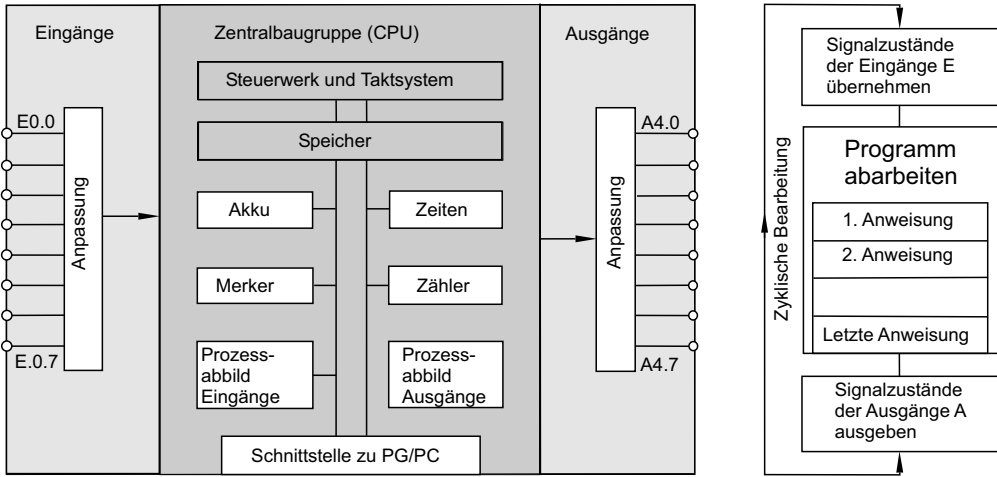


Bild 2 SPS-Struktur und zyklische Programmbearbeitung

In der Praxis werden nicht mehr ausschließlich SPS-basierte Automatisierungssysteme eingesetzt, sondern vernetzte Steuerungssysteme bestehend aus SPSen und PCs mit unterschiedlichen Aufgaben, wobei die SPS mehr prozessnah und der PC mehr übergeordnet und datenverarbeitend genutzt wird. Soll die Steuerungsfunktionalität nicht mehr von einer modernen SPS, sondern von einem PC ausgeführt werden, so gibt es diese Lösung als Industrie-PC mit eingebauter SPS-Karte und zusätzlicher Feldbus-Schnittstelle. In Sonderfällen kommt auch eine sog. Soft-SPS-Lösung in Frage, bei der in einem PC die Funktion einer SPS softwaremäßig nachgebildet und Steuerungsaufgaben „nebenher“ miterledigt werden. Für die Programmerstellung macht es keinen Unterschied, ob die Hardware-Plattform ein PC oder eine SPS ist.

3 Grundzüge der SPS-Norm IEC 61131-3

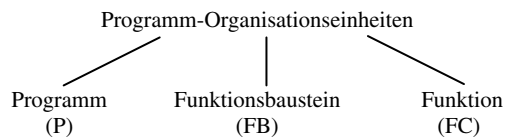
Auch wenn das Programmieren von Steuerungen und Regelungen noch immer ein sehr individueller Prozess ist, muss seit Jahren der Standard der SPS-Programmier-Norm IEC 61131-3 zu Grunde gelegt werden, der eine rationellere Programmerstellung und größere Herstellerunabhängigkeit zum Ziel hat. Den Anwendungsprogrammierern müssen wirksame Mittel an die Hand gegeben werden, um wiederverwendbare und damit kostensparende Programme entwickeln zu können. Die Norm IEC 61131-3 bzw. DIN EN 61131-3 richtet sich daher in erster Linie an SPS-Hersteller. Für Anwendungsprogrammierer ist die SPS-Norm eher ein Dokument im Hintergrund, denn sie sind für ihre Arbeit auf ein reales IEC 61131-3 kompatibles Programmier- und SPS-System angewiesen.

Die Grundzüge dieser SPS-Norm werden in den folgenden Abschnitten dargelegt und bilden die Voraussetzungen dafür, mit einem der SPS-Norm entsprechenden Programmiersystem richtig umgehen zu können.

3.1 Programmorganisationskonzept

Es ist ein hierarchisch gegliedertes System von so genannten Programmorganisationseinheiten (POE) eingeführt worden, bestehend aus einem Hauptprogrammtyp (Schlüsselwort: PROGRAM) mit Zugriffsmöglichkeit auf SPS-Eingänge/-Ausgänge und zwei Unterprogrammtypen davon einen Typ mit Gedächtnisfunktion (Schlüsselwort: FUNCTION_BLOCK) und einen Typ ohne Gedächtnisfunktion (Schlüsselwort: FUNCTION). Durch dieses Programmorganisationskonzept in Verbindung mit dem Datentyp- und Variablenkonzept zur Entwicklung strukturierter Programme wird die Wiederverwendbarkeit von Bausteinen bei späteren Anwendungen ermöglicht. Dadurch ist die neue Anwendungsprogrammierung sehr viel anspruchsvoller als die herkömmliche SPS-Programmierung geworden.

Ein Steuerungsprogramm (Anwenderprogramm) ist eine in Programm-Organisations-Einheiten (POE) gegliederte Einheit. Anstelle der unhandlichen Bezeichnung Programmorganisationseinheit wird vereinfacht auch nur von Bausteinen gesprochen.



Jeder Baustein besteht aus einem

- Deklarationsteil für die Definition aller lokal verwendeten Variablen und einem
- Rumpf für die Anweisungen (Befehle) des ausführbaren Programms.

Funktion: Dieser POE-Typ ist geeignet, wenn ein Funktionsergebnis ausschließlich aus den Eingangsvariablen des Bausteins zu ermitteln ist und unter dem Funktionsnamen des Bausteins zur Verfügung gestellt werden soll. Der Aufruf einer Funktion mit denselben Werten der Eingangsvariablen liefert deshalb immer denselben Ausgangswert zurück. Die SPS-Norm enthält einen Katalog von Standardfunktionen, die in SPS-Systemen zur Verfügung stehen sollten. Falls eine spezielle Funktion benötigt wird, kann diese vom Anwender selbst erzeugt werden, dabei muss jedoch beachtet werden, dass keine internen Zustandsvariablen deklarierbar sind, da der Baustein Typ Funktion dafür keine Speicherfähigkeit (Gedächtnis) besitzt. Eine Funktion stellt das Funktionsergebnis unter dem deklarierten Funktionsnamen zur Verfügung, sodass keine Ausgangsvariable deklariert werden muss. Es ist jedoch zulässig, Funktionen mit mehreren Ausgangsvariablen zu bilden. Funktionen können innerhalb eines Programmzyklus mehrfach aufgerufen werden, um mit unterschiedlichen Werten der Eingangsvariablen entsprechende Funktionsergebnisse zu ermitteln.

Funktionsbaustein: Dieser POE-Typ ist geeignet, wenn aus den Werten von Eingangs- und Ausgangsvariablen sowie bausteininterner Zustandsvariablen neue Ergebnisse für eine oder mehrere Ausgangsvariablen ermittelt werden sollen. Alle Werte der Ausgangs- und Zustandsvariablen bleiben von einer Bearbeitung des Funktionsbausteins bis zur folgenden erhalten. Das bedeutet, dass es bei einer erneuten Bearbeitung des Funktionsbausteins mit den gleichen Werten der Eingangsvariablen zu anderen Ausgangsergebnissen kommen kann. Anschaulich spricht man hier von einem Baustein Typ mit Gedächtnis. Um die Fähigkeiten eines Funktionsbausteins in einem Programm auch mehrfach nutzen zu können, ist die sog. *Instanziierung* der Funktionsbausteine erforderlich, worunter man das Erzeugen einer Kopie (Instanz) des Bausteines versteht. Jede Instanz muss mit einem eigenen Namen versehen werden. Unter dem Instanznamen werden die jeweils letztgültigen Variablenwerte auf entsprechenden Speicherplätzen verwaltet, während das Bausteinprogramm im Original verbleibt und dort den Instanzen bei deren Ausführung zur Verfügung steht. Die SPS-Norm schlägt viele Standardfunktionsbausteine vor, die in SPS-Systemen verfügbar sein sollten. Spezielle Funktionsbausteine können vom Anwender selbst erzeugt werden. Auch sie unterliegen bei ihrer Anwendung der Instanzenbildung.

Programm: Dieser POE-Typ bildet die oberste Hierarchieebene der Programmorganisationseinheiten. Die SPS-Norm sieht vor, dass Programme (P) in Ressourcen (SPS-Systeme) auch instanziiert werden können. Einige SPS-Systeme verwenden den Baustein Typ Programm (P) als alleiniges Hauptprogramm zur Organisation des Anwenderprogramms, das aus Funktionen (FC) und Funktionsbausteinen (FB) besteht. Der Programminhalt eines solchen (Haupt-)Programms besteht dann nur aus Aufrufen der Funktionen und Funktionsbausteine und um deren Eingangs-/Ausgangs-Variablen mit realen SPS-Ein-/Ausgängen zu verbinden.

3.2 Deklaration von FB- und FC-Bausteinen

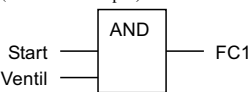


Anwenderprogramme bestehen immer auch aus sog. abgeleiteten Funktionen (FC) und/oder Funktionsbausteinen (FB), die erst durch Deklaration und Programmierung erzeugt werden müssen. Dabei sind Vorschriften zu beachten. Die Deklaration bezieht sich auf die Festlegung des Baustein Typs und auf die Bildung der Außenschnittstelle mit ihren Eingangs- und Ausgangsvariablen (-parametern) sowie der außen nicht erkennbaren bausteininternen Zustandsvariablen bei Funktionsbausteinen. Die Programmierung bezieht sich auf den Bausteinrumpf, der die Steuerungslogik enthalten muss. Deklaration und Programmierung kann in Textform oder in Grafik erfolgen.

• Deklaration einer Funktion mit dem Funktionsnamen FC 1

Bei der Deklaration in Textform sind folgende Elemente zu verwenden:

- das einleitende Schlüsselwort **FUNCTION** gefolgt vom Funktionsnamen, einem Doppelpunkt und dem Datentyp des Funktionswertes,
- das Konstrukt **VAR_INPUT...END VAR**, mit dem die Namen und Datentypen der Eingangsvariablen der Funktion festlegt werden,
- das Konstrukt **VAR_OUT...END VAR** für die Namen und Datentypen von Ausgangsvariablen,
- das Konstrukt **VAR_IN_OUT...END VAR**, das die Namen und Datentypen von Durchgangsvariablen der Funktion festlegt,
- falls erforderlich das Konstrukt **VAR...END VAR**, mit dem die Namen und Datentypen von internen temporären Hilfsvariablen festlegt werden können, deren Daten jedoch bei Beendigung der Funktion verloren gehen,
- einem Funktionsrumpf mit dem auszuführenden Programm,
- das abschließende Schlüsselwort **END_FUNCTION**.

■ **Beispiel 1:** Deklaration einer Funktion FC

Allgemein	Ausführung in Textform	Ausführung in Grafik
FUNCTION FC1 : BOOL ; (*Außenschnittstelle*) VAR_INPUT Bezeichner1 : Datentyp ; Bezeichner2 : Datentyp ; END_VAR (*Funktionsrumpf*) Programm END_FUNCTION	FUNCTION FC1 : BOOL ; (*Außenschnittstelle*) VAR_INPUT Start : BOOL ; Ventil : BOOL ; END_VAR (*Funktionsrumpf*)  END_FUNCTION	FUNCTION FC1 : BOOL ; (*Außenschnittstelle*)  (*Funktionsrumpf*)  END_FUNCTION

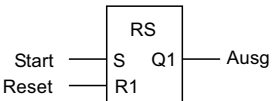

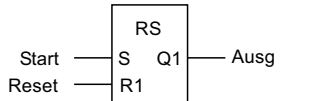
● **Deklaration eines Funktionsbausteins mit dem Namen FB 1:**

Die Deklaration eines Anwender-Funktionsbausteins in Textform erfolgt in ähnlicher Weise wie bei der Anwender-Funktion, dabei sind folgende Elemente zu verwenden:

- das einleitende Schlüsselwort **FUNCTION_BLOCK** gefolgt vom Funktionsbausteinnamen ohne einen Datentyp,
- das Konstrukt **VAR_INPUT...END VAR**, mit dem die Namen und Datentypen der Eingangsvariablen des Funktionsbausteins festlegt werden,
- das Konstrukt **VAR_OUTPUT...END VAR**, mit dem die Namen und Datentypen der Ausgangsvariablen des Funktionsbausteins deklariert werden (mehrere Ausgangsvariablen sind zulässig),

- das Konstrukt **VAR_IN_OUT...END VAR**, das die Namen und Datentypen von Durchgangsvariablen des Funktionsbausteins festlegt, deren Werte innerhalb des Bausteins durch das Programm verändert werden dürfen,
- das Konstrukt **VAR...END VAR**, das die Namen und Datentypen der bausteininternen Zustandsvariablen des Funktionsbausteins festlegt. Dieses Konstrukt wird auch verwendet, um im Funktionsbaustein eine Instanz eines Standardfunktionsbausteins zu erzeugen,
- einen Funktionsbausteinrumpf mit dem auszuführenden Programm,
- das abschließende Schlüsselwort **END_FUNCTION_BLOCK**.

■ **Beispiel 2:** Deklaration eines Funktionsbausteins FB

Allgemein	Ausführung in Textform	Ausführung in Grafik
FUNCTION_BLOCK FB 1 (*Außenschnittstelle*) VAR_INPUT Bezeichner_1 : Datentyp ; Bezeichner_2 : Datentyp ; END_VAR VAR_OUTPUT Bezeichner_3 : Datentyp ; END_VAR VAR Bezeichner_4 : Datentyp ; END_VAR (*Funktionsbausteinrumpf*) Programm END_FUNCTION_BLOCK	FUNCTION_BLOCK FB 1 (*Außenschnittstelle*) VAR Start : BOOL ; Reset : BOOL ; END_VAR VAR_OUTPUT Ausg : BOOL ; END_VAR VAR SRO_1 : RS ; END_VAR (*Funktionsbausteinrumpf*) SRO_1  END_FUNCTION_BLOCK	FUNCTION_BLOCK FB 1 (*Außenschnittstelle*)  (*Funktionsbausteinrumpf*)  END_FUNCTION_BLOCK

3.3 Variablen

In Funktionen (FC) und Funktionsbausteinen (FB) sollte nur mit symbolischen Variablen programmiert werden, um damit bibliotheksfähige Programme zu erhalten, die keine Festlegungen bezüglich der Ver-

wendung realer SPS-Eingänge/Ausgänge oder Merker enthalten. Erst auf der Ebene der Programme (P) sollte die Zuordnung der SPS-Eingänge/Ausgänge/Zähler/Zeitglieder zu den direkten (realen) Eingangs- und Ausgangsvariablen erfolgen.

Eine **Variable** ist ein mit einem Namen (Bezeichner) versehener Speicherplatz, der im Anwenderprogramm als Platzhalter für Daten fungiert, die sich zur Laufzeit des Programms ändern können. Diese Variablen können symbolisch oder direkt adressiert sein.

Unter **Daten** sollen hier Informationen aus technischen Anlagen verstanden werden, wie z.B. Messdaten über Temperaturen, Füllstände, Durchflussmengen, die verarbeitet und gespeichert werden müssen. Die Variablen sind die Mittel, um die Daten zu erfassen. Dabei wird für die Variablen ein bestimmter Datentyp festgelegt. Dieser Datentyp hängt direkt zusammen mit den auf ihn zulässigen Operationen.

Einzelement-Variablen enthalten einzelne Datenelemente mit einfachem Datentyp, die in der Tabelle 1 auszugsweise dargestellt und bei der Deklaration der Variablen zu berücksichtigen sind.

Die Deklaration symbolischer Variablen erfolgt im dafür vorgesehenen Deklarationsteil der Bausteine unter einem Namen (Bezeichner) durch Verwendung von Schlüsselwörtern wie VAR, VAR_INPUT, VAR_OUTPUT, VAR_IN_OUT und Angabe eines Datentyps wie bei der Deklaration der Bausteine FC und FB bereits gezeigt. Der Speicherort wird durch das Betriebssystem automatisch festgelegt. Der Gültigkeitsbereich einer symbolischen Variablen ist lokal auf den Baustein beschränkt, in dem sie deklariert wurde, d.h. die Variable ist nur in diesem Baustein

bekannt. Eine Ausnahme von dieser Regel bilden symbolisch adressierte Globalvariablen, die in allen Bausteinen der SPS bekannt sind aber nur sparsam verwendet werden sollten. Bei der Deklaration von Variablen in Textform sind die in Tabelle 2 angegebenen Schlüsselwörter zu verwenden. Bei grafischer Deklaration führt das Programmiersystem die Anwendung dieser Regeln im Hintergrund aus.

Multi-element-Variablen enthalten mehrere Datenelemente die in Feldern (Arrays) oder Strukturen zusammengefasst sind

- **Feld (Array):**

Ein Feld ist eine Sammlung von Datenelementen des gleichen Datentyps, die durch in eckigen Klammern [] angegebene Feldindizes angesprochen werden. Als Datentyp für Indizes sind z.B. INT, WORD, BYTE, BOOL zulässig. Es gibt eindimensionale und mehrdimensionale Felder.

■ **Beispiel 3:** Eindimensionales Feld

```
(*Deklaration*)
VAR
  Tabelle : ARRAY [0...3] OF BYTE :=
    16#00, 16#0F, 16#80, 16#FF
END_VAR
VAR_OUTPUT
  Wert : BYTE ;
END_VAR
(*Abfrage*)
Wert := Tabelle [2]
```

Tabelle 1 Elementare Datentypen

Schlüsselwort	Datentyp	Größe	Schreibweise und Wertebereiche
Bit-Datentypen			
BOOL	Boolesche Variablen	1 Bit	FALSE, TRUE
BYTE	Bit-Folge und 8 Bit-Hex-Zahlen	8 Bit	16# 0...FF
WORD	Bit-Folge und 16 Bit-Hex-Zahlen	16 Bit	16# 0000...FFFF
DWORD	Bit-Folge und 32 Bit-Hex-Zahlen	32 Bit	16# 0000_0000...FFFF_FFFF
CHAR	ASCII-Zeichen	8Bit	'A'
Arithmetiktypen			
INT	Ganze Zahlen (Festpunktzahlen)	16 Bit	-32768 bis +32767
DINT	Ganze Zahlen (Festpunktzahlen)	32 Bit	L# -2147483648 bis +2147483647
REAL	Reelle Zahlen (Gleitpunktzahlen)	32 Bit	Dezimalzahl mit Punkt: 341.7 oder Exponentialdarstellung: 3.417 E+02
Zeittypen			
TIME	Zeitdauer (IEC-Format)	32 Bit	TIME# -24d20h31m bis +24d20h31m
TIME OF DAY	Uhrzeit (Tageszeit)	32 Bit	TIME_OF_DAY#23:59:59,9
DATE	Datum	16 Bit	DATE#1990-01-01
Hinweis: Es existiert kein besonderer Datentyp für BCD-Zahlen (Binär Codierte Dezimalzahlen), diese sind eine Teilmenge der Hexadezimalzahlen, für die es die Datentypen BYTE, WORD und DWORD gibt.			

Tabelle 2 Schlüsselwörter zur Deklaration von Variablen

Schlüsselwort	Bezeichnung und Gebrauch der Variablen
VAR_INPUT ... END_VAR	Eingangsvariable: Von außerhalb kommend, nicht innerhalb des Bausteins änderbar.
VAR_OUTPUT ... END_VAR	Ausgangsvariable: Nach außen lieferbar.
VAR_IN_OUT ... END_VAR	Durchgangsvariable: Von außen kommend, innerhalb des Bausteins änderbar, nach außen lieferbar.
VAR ... END_VAR	Zustandsvariable: Gebrauch nur innerhalb des Bausteines
VAR_GLOBAL ... END_VAR	Globalvariable: Globaler Geltungsbereich in SPS
VAR RETAIN	Zustandsvariable gepuffert (Remanenzverhalten).
CONSTANT	„Konstante Variable“, nicht veränderbar.
AT	Zuweisung eines direkten Speicherortes.

• **Struktur:**

Eine Struktur ist eine Sammlung von Datenelementen unterschiedlicher Datentypen. Die Datenelemente sind in einer Hierarchie geordnet, z.B. Produkt, Version, Seriennummer, Datum.

Direkte Variablen

In der obersten Hierarchieebene der Programmorganisationseinheiten steht der Bausteintyp Programm (P). In diesem Baustein müssen nicht nur die zum Anwenderprogramm gehörenden Funktionsbausteine (FB) und Funktionen (FC) aufgerufen, sondern auch die Verbindungen zu den SPS-Eingängen/Ausgängen hergestellt werden. Nach den Vorschriften der Norm IEC 61131-3 sind diese Adressen dem Programm (P) jedoch nicht automatisch bekannt, d.h. sie müssen erst durch Deklaration bekannt gemacht werden, dazu dienen die sog. direkten Variablen.

Bei der direkten Adressierung von Variablen wird als Variablenname der physikalische Speicherort des

Datenelements verwendet, also ein SPS-Eingang/ Ausgang oder auch Merker. Zur Unterscheidung der direkten Variablen von ihrem Speicherort wird ein vorgesetztes Prozentzeichen (%) gefolgt von einem Präfix zur Kennzeichnung des Speicherortes und ein Präfix für die Speichergröße verwendet, wie in der Tabelle 3 angegeben.

■ **Beispiel 4:** Deklaration einer direkten Variablen

Allgemein	Ausführung in Textform
VAR AT %Operand : Datentyp ; END_VAR	VAR AT %IX4.7 : BOOL ; END_VAR

In einer zweiten Variante können zur Erzielung einer besseren Programmlesbarkeit Variablennamen eingeführt werden, die jedoch im Unterschied zu den richtigen symbolischen Variablen direkt mit dem physikalischen Speicherort (SPS-Eingang/Ausgang, Merker) verbunden sind (Beispiel 5).

Tabelle 3 Präfix für Speicherort und Größe der Operanden

Präfix	Bedeutung	Beispiele für direkte Variablen
I	Speicherort Eingang	Einzel-Eingänge %IX0.7... %IX0.0
Q	Speicherort Ausgang	Einzel-Ausgänge %QX0.7..%QX0.0
M	Speicherort Merker	Eingangsbyte %IB0 = %IX0.7...%IX0.0 Ausgangsbyte %QB0 = %QX0.7...%QX0.0
X	(Einzel)-Bit-Größe	Eingangswort %IW0 = %IB0+%IB1
B	Byte-(8 Bit) Größe	Ausgangswort %QW0 = %QB0+%QB1
W	Wort-(16 Bit) Größe	Eingangsdoppelwort %ID0 = %IW0+%IW1
D	Doppelwort-(32 Bit)	

■ **Beispiel 5:** Deklaration einer direkten Variablen mit symbolischen Namen

Allgemein	Beispiel
VAR Bezeichner AT %Operand : Datentyp ; END_VAR	VAR Endschalter AT %IX4.7 : BOOL ; END_VAR

Anm.: Direkte Variablen sollten nicht in den Bausteintypen FB und FC verwendet werden, weil diese Programmteile dadurch hardware-spezifische Festlegungen enthalten würden wie beispielsweise, an welchen SPS-Eingang ein bestimmter Sensor angeschlossen wird. Das aber widerspricht der Forderung nach einer bibliotheks-fähigen Gestaltung von FB- und FC-Bausteinen.

Globale Variablen

Globale Variablen sind solche, die durch eine entsprechende Deklaration in allen Bausteintypen innerhalb einer SPS bekannt gemacht wurden, also einen globalen Geltungsbereich haben. Der Deklarationsort ist der Baustein Programm (P).

Globale Variablen werden verwendet, um auf einfache Art einen bausteinübergreifenden Datenaustausch zu erreichen. Das kann jedoch unerwünschte Nebenwirkungen haben und die Verwendbarkeit eines Bausteins einschränken. Eine bessere Lösung besteht in der Deklaration von Übergabevariablen im Baustein Programm (P), um die entsprechenden Eingänge und Ausgänge der dort aufgerufenen Bausteine miteinander zu verbinden.

■ *Beispiel 6:* Deklaration einer Globalvariablen

```
PROGRAM PRG
(*Deklaration*)
VAR_GLOBAL
  Bezeichner : Datentyp ;
END_VAR
(*Programm*)
PROGRAM
END_PROGRAM
```

3.4 Programmiersprachen

Zur Erstellung der Steuerungsprogramme mit Hilfe einer Programmiersoftware stehen gemäß DIN EN 61131-3 fünf Programmiersprachen zur Verfügung stehen: Zwei textuelle Fachsprachen (AWL, ST) und zwei grafische Fachsprachen (KOP, FBS) sowie die übergeordnete Ablaufsprache (AS) mit grafischen und textuellen Elementen.

Der Programmierer kann auswählen, ob er das Bausteinprogramm in einer grafischen Sprache oder in einer Textsprache programmieren will.

- Für die grafischen Sprachen stehen in Standardbibliotheken grafische Symbole für Standardfunktionen (FC) und Standardfunktionsbausteine (FB) zur Verfügung. Diese Bausteinsymbole müssen in das Anwenderprogramm je nach Bedarf eingefügt werden, um die bezeichneten Aktionen dort ausführen zu lassen.
- Die Textsprachen benutzen anstelle von Standardbausteinen sog. Standardoperatoren, die der Programmiersprache implizit bekannt sind. Ein Operator ist ein textliches Symbol, das eine Aktion darstellt, die an einem Operanden, d.h. einer symbolischen oder direkten Variable, ausgeführt wird.

Bei der Textsprache AWL stehen die einzelnen Anweisungen in Listenform untereinander. Jede Anweisung besteht aus einem Operator, der ggf. mit einem Modifizierer ergänzt wird {N für eine Negation, C für eine Bedingung, (für Klammer auf,) für Klammer zu}, gefolgt von einem oder mehreren durch Kommas getrennten Operanden. Eine Anweisung kann auch durch eine Sprungmarke gekennzeichnet sein. Die Verarbeitung der Anweisungen erfolgt in einem Ergebnisregister, dieses enthält das sog. Aktuelle Ergebnis (AE), das auf folgende Weise für das Beispiel in Bild 3 ermittelt wird:

```
LD    E0  Lade den Wert des 1. Operanden (E0) in
        das Ergebnisregister
ANDN E1  Verknüpfe das Aktuelle Ergebnis mit
        dem negierten Wert des 2. Operanden
        (E1)
ST    A4  Speichere das veränderte Aktuelle
        Ergebnis im Zieloperanden (A4)
```

Die Tabelle 4 zeigt die Standardoperatoren der Anweisungsliste (AWL) mit kurzen Erläuterungen, auf die nicht mehr weiter eingegangen wird. Die den Standardoperatoren entsprechenden Standardbausteine werden weiter unten jedoch noch näher dargestellt.

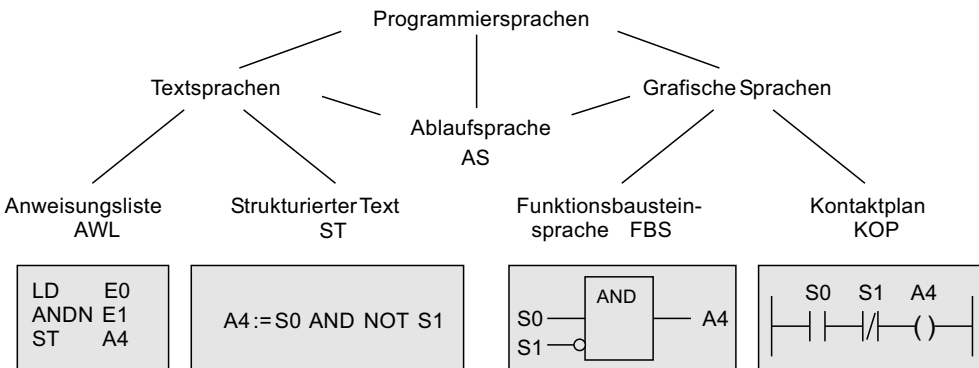


Bild 3 Übersicht zu genormten SPS-Programmiersprachen

Tabelle 4 Operatoren der Anweisungsliste (AWL)

Operator	Modifizierer	Operand/Typ	Bedeutung
LD (Load)	N	Variable/Bool	Setzt das aktuelle Ergebnis (AE) dem Operanden gleich. Speichert das aktuelle Ergebnis (boolesche 1 oder 0) auf die Operandenadresse.
ST (Store)	N		
S (Set) R (Reset)			Setzt booleschen Operator auf 1 Setzt booleschen Operator auf 0 zurück
AND OR XOR)	N, (N, (N, ()		Boolesches UND Boolesches ODER Boolesches Exklusiv-ODER Bearbeitet die eingeklammerte Operation
CAL (Call) JMP	C, N C, N	Instanzname Marke	Aufruf eines Funktionsbausteins Sprung zur Marke
GT GE EQ NE LE LT	((((((Vergleich auf größer als, > Vergleich auf größer gleich, >= Vergleich auf ist gleich, = Vergleich auf ungleich, <> Vergleich auf kleiner gleich, <= Vergleich auf kleiner als, <
ADD SUB MUL DIV	((((Addition Subtraktion Multiplikation Division

4 Programmstrukturen

Unter einer Programmstruktur versteht man den Aufbau eines Anwenderprogramms aus Codebausteinen. Es können drei Strukturen unterschieden werden.

4.1 Lineares Programm

Das gesamte Programm befindet sich in dem zyklisch bearbeiteten Programm (P). Die CPU arbeitet die Anweisungen der Reihe nach ab und beginnt dann wieder von vorne, wie in Bild 4 gezeigt.

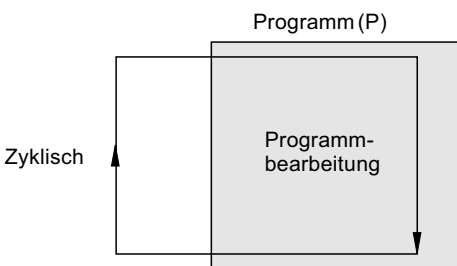


Bild 4 Lineares Programm

4.2 Gegliedertes Programm

Das Programm ist in mehrere Bausteine aufgeteilt, wobei jeder Baustein nur das Programm einer Teil-

aufgabe enthält. Die Programmorganisationseinheit Programm (P) enthält die Aufruf-Anweisungen, nach deren Reihenfolge die einzelnen Bausteine bearbeitet werden. Gegenüber dem linearen Programm besteht der Vorteil in den besser überschaubaren kleinen Einheiten. Bild 5 lässt die Gliederungsstruktur und den Aufruf der beiden Funktionen mit der erforderlichen Parameterübergabe erkennen.

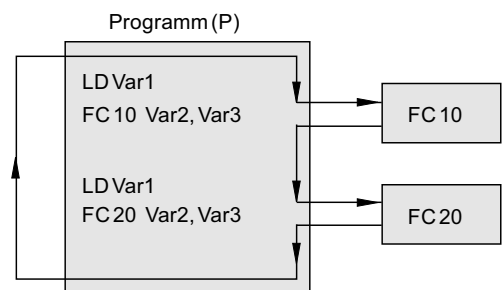


Bild 5 Das Steuerungsprogramm besteht aus zwei unterschiedlichen Programmteilen

4.3 Parametrierbares Programm

Ein Unterprogramm ist parametrierbar, wenn es seine Eingangs- und Ausgangsvariablen dem Hauptprogramm bei jedem Aufruf erneut zur Verfügung stellt und sein Programm für jeden Aufruf erneut ausführt,

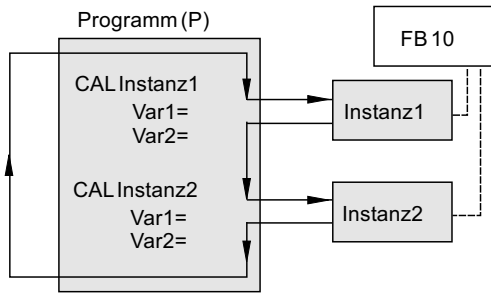


Bild 6 Das Steuerungsprogramm besteht aus zwei Instanzen eines Programmteils

um so aufrufspezifische Ergebnisse zu erzielen. Im Bild 6 ist die Parametrierbarkeit am Beispiel eines Funktionsbausteins (FB) dargestellt. Grundsätzlich ist dies auch bei Funktion (FC) möglich, jedoch mit der Einschränkung der nicht gegebenen Gedächtnisfähigkeit dieses Bausteintyps.

5 Eingabe- und Ausgabesignale

In den technischen Prozessen treten physikalische Größen wie Temperaturen, Drucke, Durchflüsse etc. auf. Automatisierungsgeräte können in der Regel nur elektrische Signale erkennen und ausgeben. Wo erforderlich, muss also eine Signalumwandlung erfolgen. Man unterscheidet verschiedene Signalarten.

5.1 Binäre Signale

Ein binäres Signal ist ein 1 Bit-Signal, das nur einen von zwei möglichen Signalzuständen annehmen kann. Ein typischer Binärsignal-Geber ist ein Schalter.

Ein Signal heißt binär, wenn es nur zweier Werte fähig ist: (TRUE = 1, FALSE = 0). Die SPS-Hersteller haben für ihre Steuerungskomponenten ein Toleranzschema festgelegt, das den Wertebereich konkreter Spannungen den binären Signalzuständen zuordnet, die von den Geräten verarbeitet werden.

Die Automatisierungsgeräte können nicht den Schaltzustand von angeschlossenen Schaltern, sondern nur

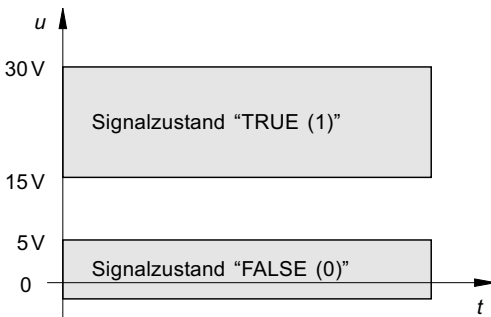


Bild 7 Signalzustände und Spannungspegel

anliegende Signale erkennen, d.h. die unterschiedliche Wirkung von Öffner- und Schließerkontakten in Anlagen muss bei der Programmerstellung bedacht werden.

Offene (unbeschaltete) Steuerungseingänge erzeugen Signalzustand „0“.

5.2 Digitale Signale

Ein digitales Signal ist eine mehrstellige Bitkette, die durch Codierung eine festgelegte Bedeutung erhält, z.B. als Zahlenwert. Ein typischer Digitalsignal-Geber ist ein Zifferneinsteller. Um z.B. die Zahlen 0 bis 9 darstellen zu können, sind vier Binärstellen erforderlich.

Dezimalzahl	Dualzahl				Wert
	8	4	2	1	
0	0	0	0	0	
1	0	0	0	1	
2	0	0	1	0	
3	0	0	1	1	
4	0	1	0	0	
5	0	1	0	1	
6	0	1	1	0	
7	0	1	1	1	
8	1	0	0	0	
9	1	0	0	1	

1 Binärstelle = 1 Bit

1 Byte = 8 Bit

1 Wort = 2 Byte = 16 Bit

1 Doppelwort = 32 Bit

5.3 Analoge Signale

Für ein analoges Signal ist charakteristisch, dass der Signalparameter (z.B. die Spannung) innerhalb bestimmter Grenzen jeden beliebigen Wert annehmen kann. Automatisierungsgeräte können intern keine analogen Signale verarbeiten. So genannte Analogbaugruppen nehmen eine Signalumsetzung vor und wandeln ein analoges Signal in ein digitales Signal um bzw. auch umgekehrt.

6 Eingabe-/Ausgabebaugruppen

Die im Bild 1 angedeuteten Eingabe- und Ausgabebaugruppen der SPS werden üblicherweise als zentrale digitale Eingabe- und Ausgabebaugruppen bezeichnet im Gegensatz zu den dezentralen Baugruppen (Slaves), die über ein Feldbussystem angeschlossen sind. Die digitalen Eingabe- und Ausgabebaugruppen umfassen meistens 1 Byte = 8 Bit; 2 Byte = 16 Bit; 4 Byte = 32 Bit Eingänge bzw. Ausgänge. Im Steuerungsprogramm können Bits, Bytes, Worte oder Doppelworte abgefragt oder angesteuert werden. Bei Analogbaugruppen sind entsprechend die Anzahl der Eingänge bzw. Ausgänge angegeben.

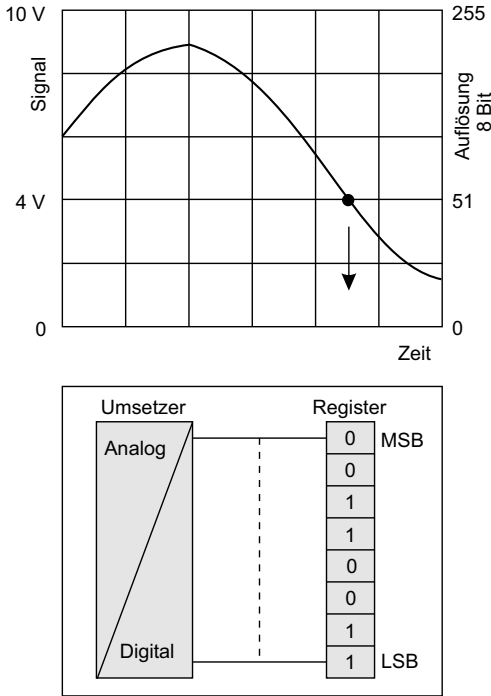


Bild 8 Ein Spannungswert wird in eine Zahl umgesetzt (vereinfacht)

Digitaleingabebaugruppen gibt es für DC 24 V und AC 120/230 V mit Potenzialtrennung über Optokoppler sowie Anzeige des aktuellen Signalzustandes durch Leuchtdioden. Aufgrund von Filtermaßnahmen gegen Störsignale liegt die Frequenzobergrenze für Eingangssignale bei etwa 50 Hz. Die Digitaleingabebaugruppen formen die Pegel der externen digitalen Signale aus dem Prozess in den internen Signalpegel des SPS-Systems um. Die Baugruppen sind z.B. geeignet für den Anschluss von Schaltern und 2-Draht-Näherungsschaltern (BERO).

Digitalausgabebaugruppen gibt es für Lastspannungen DC 24 V oder AC 120/230 V bei spezifizierter Strombelastbarkeit und Potenzialtrennung mittels Optokoppler. Die Schaltfrequenz der Ausgänge wird nach ohmscher Last, induktiver Last und Lampenlast unterschieden und liegt im Bereich bis 100 Hz. Die Digitalausgabebaugruppen formen den internen Signalpegel des SPS-Systems in die externen, für den Prozess benötigten Signalpegel um. Die Baugruppen sind z.B. geeignet für den Anschluss von Magnetventilen, Schützen, Kleinmotoren, Lampen und Motorstartern.

Analogeingabebaugruppen wandeln analoge Signale aus dem Prozess in digitale Signale für die interne Verarbeitung innerhalb der SPS um. Es können Spannungs- und Stromgeber, Thermoelemente, Widerstände und Widerstandsthermometer angeschlossen werden:

Spannung	z.B. ± 10 V
Strom	z.B. 4 bis 20 mA
Widerstand	z.B. 0 ... 300 Ohm
Thermoelement	z.B. Typ E, N, K mit Kennlinien – Linearisierung
Widerstandsthermometer	z.B. Pt 100-Standard mit Kennlinien-Linearisierung

Die Baugruppen verfügen über eine parametrierbare Auflösung von z.B. 9 bis 15 Bit + Vorzeichen, unterschiedliche Messbereiche (einstellbar durch Messbereichsmodule und Software) sowie Alarmfähigkeit (Diagnose und Grenzwertalarne an die CPU).

Analogausgabebaugruppen wandeln digitale Signale aus der SPS in analoge Signale für den Prozess um und sind für den Anschluss analoger Aktoren geeignet. Als Ausgangsbereiche werden angeboten:

Spannungsausgang	z.B. ± 10 V
Stromausgang	z.B. 0 bis 20 mA

Die Baugruppen haben eine Auflösung von 12 bis 15 Bit. Es sind unterschiedliche Messbereiche je Kanal einstellbar.

7 Verknüpfungssteuerungen

SPS-Programme werden mit einem IEC 61131-3 orientierten Projektierungssystem (z.B. CoDeSys) als Projekte angelegt, in Bausteine programmiert und unter einem Dateinamen abgespeichert.

Als Verknüpfungssteuerungen bezeichnet man solche Programme, die Ausgangssignale überwiegend unter Verwendung einfacher Logikbeziehungen, Speicherfunktionen, Zeitglieder, Zähler u.a. in zyklischer Bearbeitungsweise erzeugen. Damit stellen Verknüpfungssteuerungen den größten Teil der SPS-Programme dar. Neben den Verknüpfungssteuerungen gibt es noch die Programmart der Ablaufsteuerungen.

7.1 Logische Grundverknüpfungen in verschiedenen Darstellungen

Nachfolgend wird eine Auswahl wichtiger Programmierfunktionen zur Realisierung von Verknüpfungssteuerungen gezeigt. In Bild 9 werden die logischen Grundverknüpfungen UND, ODER, NEGATION gezeigt. Die in diesen Beispielen vorkommenden Eingangs-/Ausgangsbezeichnungen wie E1, E2, A4 oder auch S1 usw. sind als kurze Namen von deklarer Variablen mit zutreffenden Datentyp in entsprechenden Bausteinen zu betrachten und nicht zu verwechseln mit bekannten SPS-Operanden.

7.2 Zusammengesetzte logische Grundverknüpfungen

Zu den logischen Grundverknüpfungen zählen auch die beiden häufig vorkommenden Strukturen UND-

Funktion	Zeitdiagramm	FBS	KOP	AWL
UND $A4 = E1 \wedge E2$ $A4 = E1 \& E2$ $A4 = E1 E2$				LD E1 AND E2 ST A4
ODER $A4 = E1 \vee E2$				LD E1 OR E2 ST A4
NICHT $A4 = \bar{E1}$				LDN E1 ST A4
Ausgangs-NEGATION $A4 = \bar{E1 \wedge E2}$				LD E1 AND E2 STN A4

Bild 9 Logischen Grundverknüpfungen nach IEC 61131-3

vor-ODER sowie ODER-vor-UND, die in Bild 10 in grafischer Form (FBS) und Textform (AWL) dargestellt sind. Die zugehörigen Schaltfunktionen heißen

Disjunktive Normalform für die UND-vor-ODER-Struktur und Konjunktive Normalform für die ODER-vor-UND-Struktur.

UND-vor-ODER-Verknüpfung 1) Allgemeiner Fall siehe nachfolgend bei DNF $A = E1 E2 \bar{E3} \vee E1 \bar{E2} \vee E3$		LD E1 AND E2 ANDN E3 OR(AND E1 ANDN E2) OR E3 ST A4
2) Spezieller Fall: Antivalenz (Exklusiv-ODER) $A = E1 \leftrightarrow E2$		LD E1 XOR E2 ST A5
ODER-vor-UND-Verknüpfung $A = (E1 \vee E2) \wedge (\bar{E1} \vee \bar{E2}) \wedge E3$		LD E1 OR E2 AND(ORN E1 ORN E2) AND E3 ST A4

Bild 10 Zusammengesetzten logischen Grundverknüpfungen nach IEC 61131-3

7.3 Schließer- und Öffnerkontakte, Drahtbruchsicherheit, Erdschlussgefahr

Befehlsgeber können auf einer Schließer- oder Öffner-Funktion beruhen, d.h. bei Betätigung ein 1-Signal oder ein 0-Signal an den Steuerungseingang liefern. Ein Startbefehl für eine Steuerung ist mit einem 1-Signal, ein Haltbefehl mit einem 0-Signal auszuführen. Bei Gleichzeitigkeit muss der Haltbefehl Vorrang haben.

Das Einschalten einer Steuerung durch einen Schließerkontakt (Arbeitsstromprinzip) und das Ausschalten mit einem Öffnerkontakt (Ruhestromprinzip) macht die Steuerung drahtbruchsicher. Bei Auftreten eines Drahtbruchs erfolgt kein unbeabsichtigtes Einschalten der Steuerung, jedoch wird eine eingeschaltete Steuerung abgeschaltet. Die Erdschlussgefahr erfordert geerdete Steuerkreise oder Isolationsüberwachung.

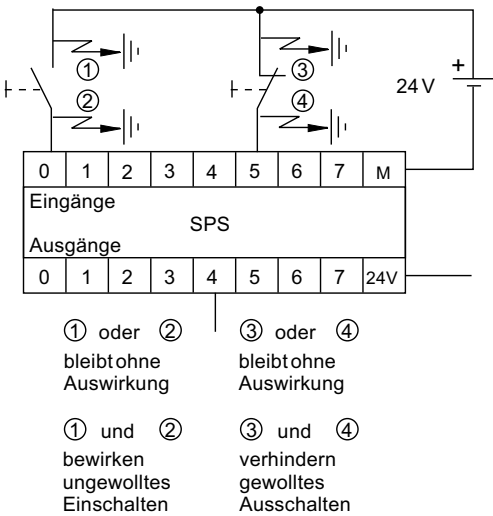


Bild 11 Auswirkungen von Erdschlüssen im nicht geerdeten Steuerkreis

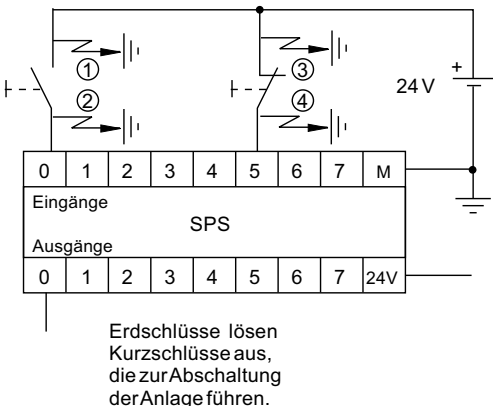
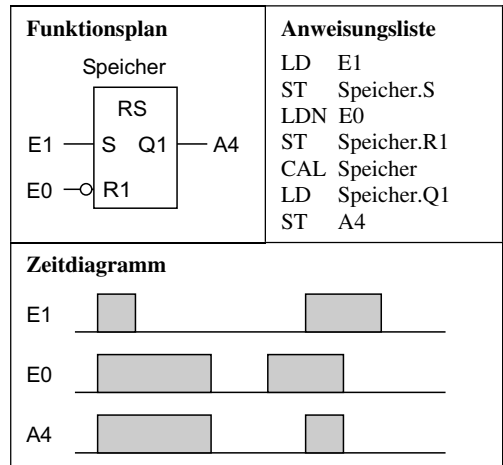


Bild 12 Auswirkung von Erdschlüssen im geerdeten Steuerkreis

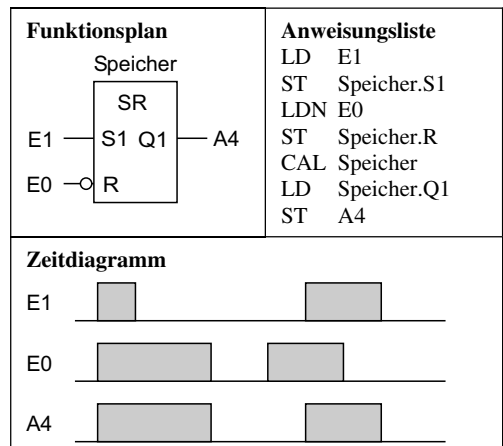
7.4 Speicherfunktionen

Viele Steuerungsaufgabe erfordern die Verwendung von Speicherfunktionen. Eine Speicherfunktion liegt dann vor, wenn ein kurzzeitig auftretender Signalzustand über den Programmzyklus hinaus festgehalten, d.h. gespeichert und erst zu einem späteren Zeitpunkt wieder gelöscht werden muss. Die Ausführung einer Speicherfunktion umfasst das Setzen und Rücksetzen des Speichers. Für die Speicherfunktionen stehen bistabile Standardfunktionsbausteine zur Verfügung, aus denen sich Instanzen zur Verwendung im Anwenderprogramm bilden lassen. Zu unterscheiden sind zwei Speicherbausteintypen, die unterschiedlich auf gleichzeitiges Eintreffen von Setz- und Rücksetzsignalen reagieren.

Speichern mit vorrangigem Rücksetzen



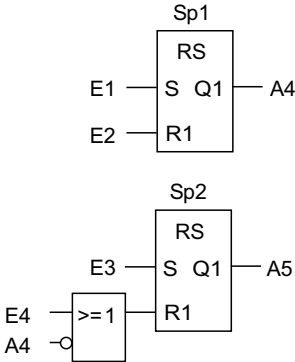
Speichern mit vorrangigem Setzen



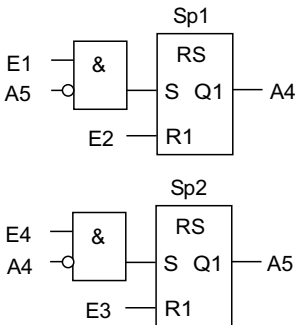
■ **Gegenseitiges Verriegeln**

Zwei Arten von Verriegelungen lassen sich unterscheiden. Bei dem gegenseitigen Verriegeln dürfen die Speicher nicht gleichzeitig gesetzt sein, z.B. für Motor-Rechtslauf und Linkslauf. Für die Ausführung von Speicher-Verriegelungen gibt es zwei Varianten.

Verriegeln über die Rücksetzeingänge



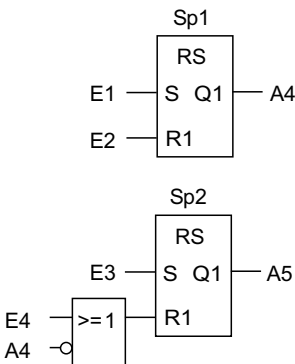
Verriegeln über die Setzeingänge



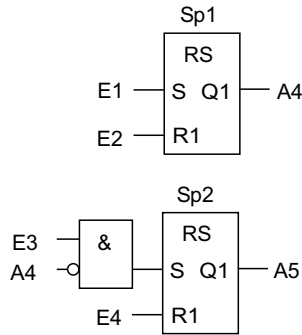
■ **Reihenfolgeverriegelung**

Bei einer Reihenfolgenverriegelung darf ein Speicher nur gesetzt werden, wenn zuvor ein oder mehrere andere Speicher gesetzt sind. Ein Beispiel sind hintereinander geschaltete Förderbänder. Auch bei der Reihenfolgeverriegelung gibt es zwei Ausführungsvarianten.

Verriegeln über den Rücksetzeingang



Verriegeln über den Setzeingang



7.5 Flankenauswertung

Die Flankenauswertung eines Signals ist dann erforderlich, wenn aus einem Signalwechsel von FALSE nach TRUE oder umgekehrt ein kurzer Impuls der Zeitdauer von einer Programmzykluszeit herzuleiten ist, unabhängig von der Betätigungsdauer des Signalgebers. Das ist z.B. wichtig, wenn bei einer Sicherheitsschaltung überprüft werden muss, ob ein Taster wirklich betätigt wurde oder nur dauerhaft durch Manipulation niedergehalten wird.

Im Bild 13 werden Beispiele gezeigt, wie die Impulsgewinnung unter Verwendung von Standardfunktionsbausteinen für Flankenerkennung gelöst werden kann. P-Flanke und N-Flanke sind die Namen der Instanzen der Funktionsbausteine. Der Zykluszeitimpuls steht als Impulsoperand IO zur Verfügung.

7.6 Darstellung und Eigenschaften elektropneumatischer Stellglieder

Elektropneumatische Stellglieder können mit Ausgangssignalen einer SPS angesteuert werden. Ihre Hilfsenergie ist Druckluft, mit der pneumatische Zylinder betätigt werden. Bild 14 zeigt drei häufig verwendete elektropneumatische Stellgliedertypen.

3/2-Wegeventil: Elektromagnetisches Ventil mit Rückstellfeder hat nur einen elektrischen Steuereingang Y1 und kann durch ein Stellsignal aus der Schaltstellung b in die Schaltstellung a geschaltet werden. Nach Beendigung des Stellsignals erfolgt eine federmechanische Rückstellung des Ventils, dessen Vorteil die definierte Schaltstellung im unbetätigten Zustand ist.

5/2-Wegeventil: Elektromagnetische Impulsventile haben zwei elektrische Steuereingänge Y1 und Y2, sie können durch kurze Ansteuerimpulse aus einer Schaltstellung in die andere umgeschaltet werden. Die Ventile übernehmen die RS-Speicherfunktion der Steuerung. Nachteil des Speicherverhaltens ist die nicht definierte Schaltstellung im unbetätigten Zustand, daher erfolgt meistens eine Ansteuerung von Y1 und Y2 mit inversen Signalen.

FUP	AWL	Zeitdiagramm
<p>Positive (steigende) Flanke 0 → 1</p> <p>P_Flanke</p> <p>S1 — CLK — Q — IO</p>	<p>LD S1 ST P_Flanke.CLK CAL P_Flanke LD P_Flanke.Q ST IO</p>	<p>Zyklen 1 2 3 .. n n+1</p>
<p>Negative (fallende) Flanke 1 → 0</p> <p>N_Flanke</p> <p>S1 — CLK — Q — IO</p>	<p>LD S1 ST F_Flanke.CLK CAL F_Flanke LD F_Flanke.Q ST IO</p>	<p>Zyklen 1 2 3 .. n n+1</p>
<p>P_Flanke Speicher</p> <p>E1 — CLK — Q — S E0 — R1 — Q — A4</p>	<p>LD E1 ST P_Flanke.CLK CAL P_Flanke LD P_Flanke.Q ST Speicher.S LDN E0 ST Speicher.R1</p>	<p>Der Impulsoperanden IO kann entfallen, wenn die Flankenauswertung nur an einer Stelle des Programms benötigt wird, im Beispiel zum Setzen des RS-Speichers.</p>

Bild 13 Flankenauswertung nach IEC 61131-3

5/3-Wegeventil: Bei elektromagnetischen Impulsventilen mit Federzentrierung geht das Ventil im unbetätigten Zustand in die Mittelstellung. Daher kann neben der Vorwärts- und Rückwärtsbewegung des Zylinderkolbens auch eine Halteposition veranlasst werden.

Funktionsdiagramme zeigen den Bewegungsverlauf von Zylindern. In der Ordinate wird der zurückgelegte Weg und in der Abszisse werden Schritte oder Zeiten aufgetragen. Zusätzlich können die Zustände von Magnetspulen der Ventile dargestellt werden.

7.7 Regeln für das Umsetzen von Schützschaltungen in SPS-Programme

Eine gegebene Schützsteuerung, die z.B. zur Ansteuerung von Elektromotoren oder Elektropneumatik eingesetzt wurde, kann durch eine SPS-Steuerung

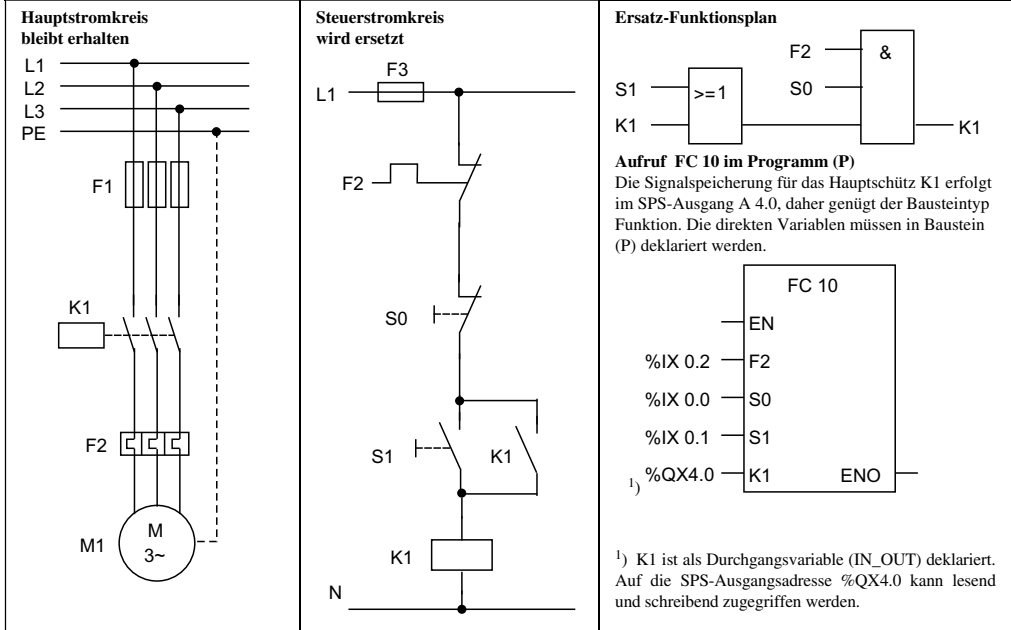
unter sinngemäßer Anwendung der nachfolgenden Umsetzungsregeln ersetzt werden.

1. Der Hauptstromkreis wird unverändert außerhalb der SPS beibehalten.
2. Hauptschütze werden von den SPS-Ausgängen angesteuert und werden als OUT- oder IN-OUT-Variable deklariert. Der Schaltzustand des Hauptschützes wird somit außerhalb des Bausteins in einem SPS-Ausgang gespeichert. Wird ein Kontakt K1 des Hauptschützes K1 am Bausteineingang abgefragt, ist K1 als DurchgangsvARIABLE (IN_OUT) zu deklarieren.
3. Hilfsschütze werden durch temporäre oder statische Variablen innerhalb des Bausteins ersetzt. Wenn Hilfsschütze interne Speicherfunktionen ausüben, muss für jedes Hilfsschütz eine statische SpeichervARIABLE (VAR) deklariert werden.

<p>Einfachwirkender Zylinder mit Ansteuerung durch ein 3/2-Wegeventil</p> <p>1Y1</p>	<p>Doppeltwirkender Zylinder mit Ansteuerung durch ein 5/2-Wegeventil</p> <p>2Y1 2Y2</p>	<p>Doppeltwirkender Zylinder mit Ansteuerung durch ein 5/3-Wegeventil</p> <p>3Y1 3Y2</p>
--	--	--

Bild 14 Beispiele elektropneumatischer Stellglieder

■ **Beispiel 7:** Umsetzung einer Schützschaltung in ein SPS-Programm



4. Parallelschaltungen von Schützkontakten werden durch ODER-Verknüpfungen und Reihenschaltungen durch UND-Verknüpfungen der Variablen ersetzt.
5. Öffner von Schützkontakten werden negiert und Schließer bejaht abgefragt.
6. Öffner- und Schließerkontakte von Signalgebern wie Taster und Schalter werden im Programm bejaht abgefragt und als Eingabevariablen deklariert, wenn derselbe Kontakttyp beibehalten wird.
7. Die bejahte Abfrage von Signalgebern in Regel 6 gilt nicht bei Verwendung der Speicherfunktion anstelle der Selbsthaltung.
8. Die Umsetzungsregeln 1 bis 6 verändern nicht die vorgegebene Steuerungsstruktur, wenn die Schütze keine Zeitverzögerungen oder Wischerkontakte enthalten. Einschalt- und Ausschaltverzögerungen müssen mit Zeitgliedern und Wischerkontakten mit Flankenbewertung nachgebildet werden. Impulse von Wischerkontakten beim Einschalten (Ausschalten) entsprechen steigenden (fallenden) Flanken.

7.8 Zeitfunktionen

Die Zeitbildung ist eine Grundfunktion der Steuerungstechnik. Es können zeitliche Abläufe, wie Warte- und Überwachungszeiten, Zeitmessungen oder Taktimpulse programmiert werden. In der Norm DIN EN 61131-3 sind neben einer Echtzeituhr drei Stan-

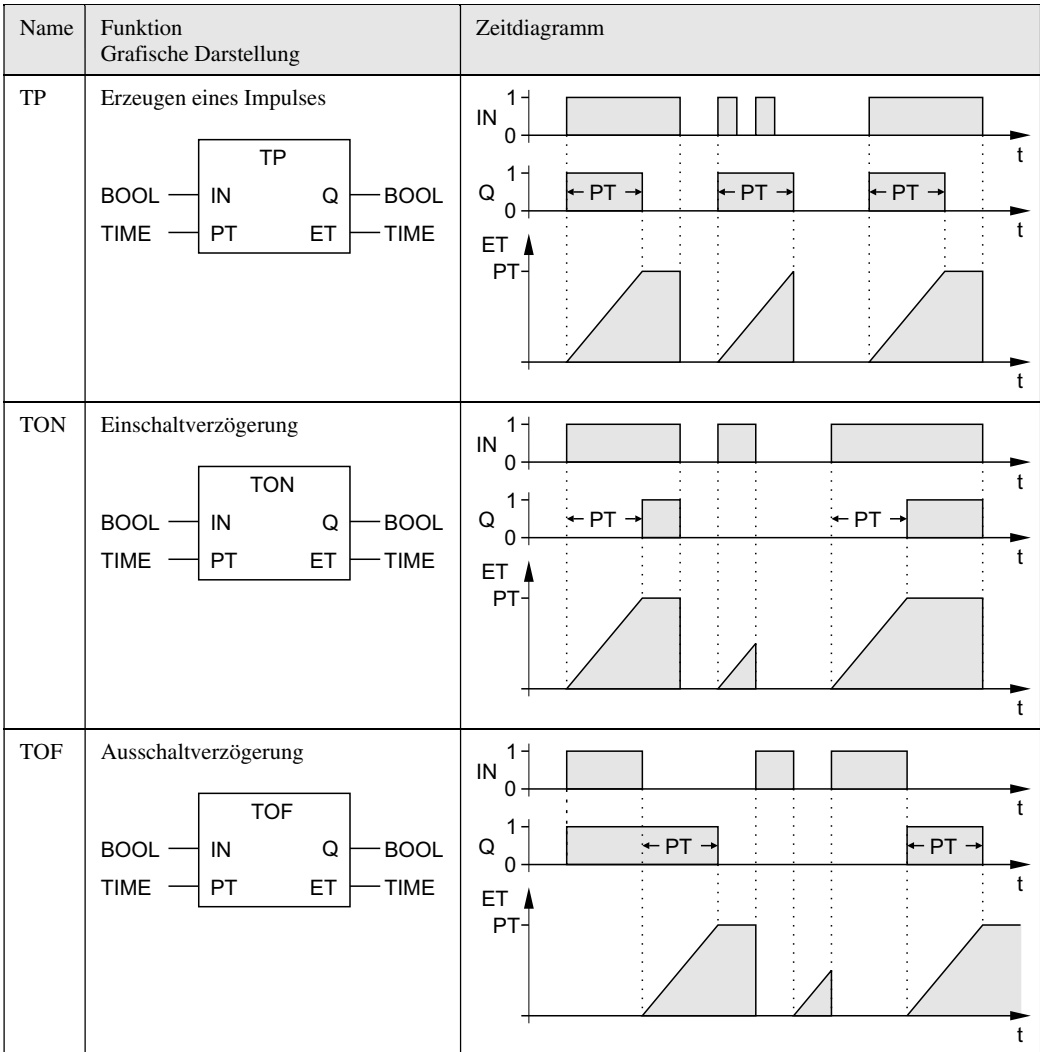
dardfunktionsbausteine für die Zeitbildung aufgeführt, die in Bild 15 gezeigt werden.

7.9 Zählerfunktionen

Zählerfunktionen werden in Steuerungsaufgaben benötigt, um bestimmte Mengen oder Positionen durch Aufzählen von Impulsen zu erfassen, Frequenzen zu ermitteln oder die Funktion von Steuerwerken zu übernehmen.

Grundsätzlich lassen sich Zähler in Automatisierungssystemen auf drei verschiedene Arten realisieren.

- Zähler werden als Funktionsbausteine oder Funktionen im Steuerungsprogramm aufgerufen und parametrierbar. Diese Zähler können je Zykluszeit nur einen Vorwärts- und einen Rückwärtszählimpuls verarbeiten. Die Verarbeitung von externen Zählimpulsen ist von der Zykluszeit und der Schaltfrequenz der Signaleingänge abhängig. Innerhalb dieses Kapitels wird ausschließlich diese Zählerart verwendet.
- Zähler werden durch eine Variable vom Datentyp Integer oder Doppelinteger realisiert. Das Auf- bzw. Abwärtszählen erfolgt bei diesen Zählern mit Additions- bzw. Subtraktionsbefehlen. Diese Zähler können je Zykluszeit mehrere interne Vorwärts- bzw. Rückwärtszählimpulse verarbeiten. Die Verarbeitung von externen Zählimpulsen ist jedoch wieder von der Zykluszeit abhängig.



Operandenbedeutung:

IN = Startbedingung; PT = Zeitvorgabe; Q = Status der Zeit; ET = Aktueller Zeitwert

Bild 15 Zeitglieder nach IEC 61131-3

- Zähler sind auf einer speziellen Baugruppe untergebracht oder sind als „Schnelle Zähler“ mit separaten Signaleingängen Teil des Betriebssystems der Zentralbaugruppe CPU. Mit diesen Zählern ist es möglich, externe Zählimpulse zu erfassen, die schneller als die Zykluszeit sind. Innerhalb des Steuerungsprogramms können Zählerstände mit Übergabevariablen abgerufen werden.

Bei der Realisierung von Zählern unterscheidet man Vorwärts- bzw. Aufwärts-Zähler, Rückwärts- bzw. Abwärts-Zähler und Vor-Rückwärts- bzw. Auf-Abwärts-Zähler. Der Zählerstand ergibt sich aus der Anzahl der positiven Signalflanken an den Eingängen für Vorwärts- und Rückwärtszählen.

In der Norm DIN EN 61131-3 sind für die drei Zähler die grafische Darstellung und die Arbeitsweise der Standardfunktionsbausteine festgelegt wie in Bild 16 abgebildet.

7.10 Vergleichsfunktionen

Mit Vergleichsfunktionen werden die Werte zweier digitaler Operanden des gleichen Datentyps verglichen. Das sind in den meisten Fällen Variablen mit Datentypen für Zahlenwerte wie REAL (Dezimalzahlen mit Komma), INT (Ganzzahlen mit Vorzeichen) und WORD (vorzeichenlose Zahlen).

Zur Durchführung von Vergleichen stehen Standardfunktionen (FC) zur Verfügung, die beliebig oft mit

Name	Grafische Darstellung	Arbeitsweise in ST-Sprache
Aufwärts-Zähler		<pre>IF R THEN CV := 0; ELSIF CU AND (CV < Pymax) Then CV := CV+1; END_IF; Q := (CV >= PV)</pre>
Abwärts-Zähler		<pre>IF LD THEN CV := PV; ELSIF CD AND (CV < Pvmin) Then CV := CV-1; END_IF; Q := (CV <= 0)</pre>
Auf-Abwärts-Zähler		<pre>IF R THEN CV := 0; ELSIF LD THEN CV := PV; ELSIF CU AND (CV < Pymax) THEN CV := CV+1; ELSIF CD AND (CV < Pvmin) THEN CV := CV-1; END_IF; QU := (CV >= PV) QD := (CV <= 0)</pre>

Bedeutung der Eingänge und Ausgänge: CU = Vorwärtszähleingang steigende Flanke; CD = Rückwärtszähleingang steigende Flanke; R = Rücksetzeingang; LD = Ladeeingang; PV = Ladewert; QU = Status Zählerstand (CV größer gleich Ladewert); QD = Status Zählerstand (CV kleiner gleich 0); CV = Zählerstand.

Hinweis: Die numerischen Werte der Grenzwariablen Pymax und Pvmin sind implementierungsabhängig. Mögliche Werte sind Pymax = 32767 und Pvmin = -32768.

Bild 16 Zählfunktionen nach IEC 61131-3

verschiedenen Eingangsparametern aufgerufen werden können. Das Vergleichsergebnis steht als binärer Signalzustand im aktuellen Ergebnis (AE) zur Verfügung, und zwar mit TRUE, wenn der Vergleich zutrifft und mit FALSE, wenn er nicht zutrifft. Die Tabelle 5 zeigt die Vergleichsmöglichkeiten für den einfachsten Fall von zwei Vergleichseingängen IN1 und IN2.

Tabelle 5 Vergleichsfunktionen

Name	Symbol	Beschreibung
GT	>	IN1 größer IN2
GE	>=	IN1 größer gleich IN2
EQ	=	IN1 gleich IN2
LE	<=	IN1 kleiner gleich IN2
LT	<	IN1 kleiner IN2
NE	<>	IN1 ungleich IN2

■ **Beispiel 8:** Vergleich auf „größer“

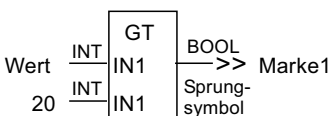


Bild 17

Die Eingangsvariable „Wert“ vom Datentyp Integer wird mit der Integerkonstanten „20“ auf „größer als“ verglichen, siehe Bild 17. Am Ausgang des Vergleichers ist das Sprungsymbol durch eine Signallinie mit Doppelpfeil am Ende angegeben. Der Sprung ist bedingt, weil er vom Ausgang des Vergleichs abhängig ist. Ist die Vergleichsbedingung erfüllt, wird das Programm an der Stelle von Marke1 fortgeführt. Ist die Vergleichsbedingung nicht erfüllt, läuft das Programm linear weiter. Im Vergleichsbaustein ist der Name der Vergleichsfunktion angegeben (GT).

7.11 MOVE-Funktion

Die MOVE-Funktion zählt zu den arithmetischen Standardfunktionen. Sie hat genau einen Eingang und einen Ausgang und bewirkt das „Durchreichen“ von Daten der Eingangsvariablen an die Ausgangsvariable in Abhängigkeit von einem booleschen Freigabesignal EN, dabei darf der Datentyp auf der Ausgangs- und Eingangsseite sogar verschieden sein. Die Nutzung der MOVE-Funktion zur Datentypumwandlung hat enge Grenzen. Sinnvoll ist z.B. die Umwandlung von BYTE in WORD. Mit der Datentypangabe ANY in Bild 18 ist kein neuer Datentyp gemeint. ANY ist der ranghöchste der sog. allgemeinen Datentypen der SPS-Norm, die sich nur an Hersteller wendet und

mitteilt, dass jeder nachgeordnete Datentyp verwendet werden darf. Die Ausgestaltung der MOVE-Funktion ist herstellerabhängig.

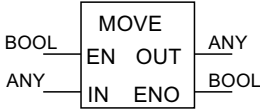


Bild 18
MOVE-Funktion mit Ausführungssteuerung EN/ENO

7.12 EN/ENO-Mechanismus

EN steht für Enable IN (boolescher Freigabe-Eingang) und ENO für Enable OUT (boolescher Freigabeausgang). Die SPS-Norm sieht vor, dass Funktionen (FC) in den Programmiersprachen FBS und KOP über diese Möglichkeit der Ausführungssteuerung verfügen müssen. Der Freigabe-Eingang EN kann mit einer Logikverknüpfung beschaltet sein (z.B. UND-Glied, Vergleichler). Der boolesche Ausgang ENO kann zur Ansteuerung nachfolgender Funktionen benutzt werden, weil er anzeigt, ob die Bausteinfunktion ordnungsgemäß ausgeführt wurde oder nicht.

Die Variablen EN/ENO müssen nicht benutzt werden, wenn die Ausführung der Funktion in jedem Fall erfolgen soll.

Ist bei Funktionsaufruf EN = 0, wird die Funktion nicht ausgeführt und ENO = 0 ausgegeben.

Ist bei Funktionsaufruf EN = 1, wird die Funktion ausgeführt und der Ausgang meldet mit ENO = 1 die Fehlerfreiheit der Ausführung oder mit ENO = 0 einen Ausführungsfehler.

8 Aufruf und Wertübergaben zwischen Bausteinen nach IEC 61131-3

8.1 Aufrufhierarchie der Bausteine P, FB und FC

- Das Anwenderprogramm einer SPS hat einen hierarchischen Aufbau:

An oberster Stelle steht ein Baustein des Typs Programm (P), dessen Deklaration und Gebrauch identisch ist mit denen der bereits beschriebenen Funktionsbausteine, jedoch mit den begrenzenden Schlüsselwörtern **PROGRAM ... END_PROGRAM**. In diesem Baustein können Instanzen von Funktionsbausteine (FB) sowie Funktionen (FC) aufgerufen werden.

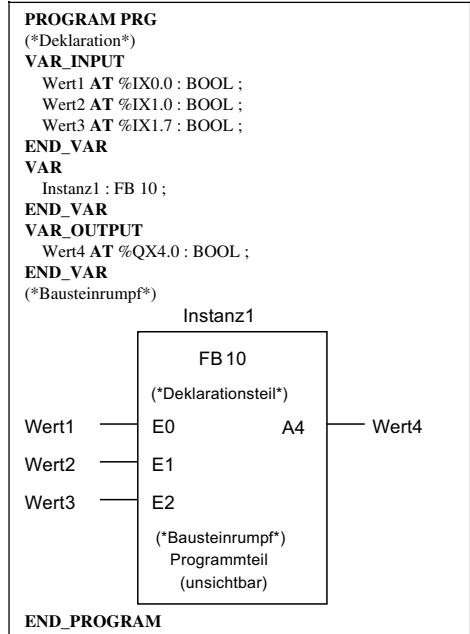
An mittlerer Stelle stehen die Bausteine des Typs Funktionsbaustein (FB). Innerhalb eines Funktionsbausteins können Instanzen anderer Funktionsbausteine oder auch Funktionen (FC) aufgerufen werden. An unterster Stelle stehen die Bausteine vom Typ Funktionen (FC). Innerhalb einer Funktion können andere Funktionen (FC) aufgerufen werden.

8.2 Aufruf von Funktionsbausteinen in FBS

Am einfachsten gestalten sich Bausteinaufrufe und die damit verbundenen Werteübergaben in grafischer Programmiersprache, z.B. FBS. Die Einbindung eines Funktionsbausteins (FB) in ein Programm (P) geschieht in der logischen Reihenfolge: Deklarieren einer Instanz des Funktionsbausteins, Einfügen des Bausteinsymbols aus der Bausteinbibliothek und Zuordnung des Instanznamens, Anbindung der Instanz-Eingänge/Ausgänge an Variablen des aufrufenden Bausteins, z.B. auch an direkte Variablen, welche die Verbindung zu den realen SPS-Eingängen/Ausgängen halten.

Instanzen werden deklariert wie Variablen durch Angabe eines Namens und eines Typs. Als Typ wird der Name des Funktionsbausteins verwendet.

■ *Beispiel 9:* Programm P ruft Funktionsbaustein FB auf



8.3 Aufruf von Funktionsbausteinen in AWL

Gleicher Deklarationsteil wie in der Grafiksprache FBS. Etwas schwieriger gestaltet sich der Bausteinaufruf und die damit verbundenen Werteübergaben in Textsprache, z.B. AWL. Es stehen mehrere Methoden zur Verfügung, deren einfachste ist die mit der formalen Argumentenliste, die auch vom Hersteller fest vorgegeben sein kann. Bei dieser Methode muss ein Funktionsbaustein mit dem Befehl CAL (unbedingt) oder CALC (bedingt) aufgerufen werden, gefolgt vom Instanznamen und der offenen Klammer. Darunter erscheint die auszufüllende Liste mit den Parametern.

tern des aufgerufenen Instanzbausteins, die mit der geschlossenen Klammer beendet wird.

■ **Beispiel 10:** Programm P ruft Funktionsbaustein FB auf

```
PROGRAM PRG
(*Deklaration*)
VAR_INPUT
  Wert1 AT %IX0.0 : BOOL ;
  Wert2 AT %IX1.0 : BOOL ;
  Wert3 AT %IX1.7 : BOOL ;
END_VAR
VAR
  Instanz1 : FB 10 ;
END_VAR
VAR_OUTPUT
  Wert4 AT %QX4.0 : BOOL ;
END_VAR
(*Bausteinrumpf*)
CAL Instanz1 (
  E0 := Wert1,
  E1 := Wert2,
  E2 := Wert3,
  A4 => Wert4 );
END_PROGRAM
```

8.4 Aufruf von Funktionen in AWL

Gegenüber dem Aufruf von Funktionsbausteinen entfällt die Instanzbildung. Der Aufruf in der Grafiksprache FBS ist sonst der gleiche wie bei Funktionsbausteinen. Unterschiede bestehen beim Aufruf in der Textsprache AWL, da nicht der Operator CAL verwendet werden darf. Der Funktionsname ist als Operator in der Anweisung anzugeben. Es sind zwei Aufrufmethoden zu unterscheiden.

Beim „*formalen Aufruf*“ wird als erste Anweisung einfach der Name der aufzurufenden Funktion geschrieben, gefolgt von der offenen Klammer. Danach wird je Anweisungszeile ein Eingangsparameter übergeben, wie bei der obigen formalen Argumentenliste, jedoch nur für die Funktionseingänge. Dann folgt die geschlossene Klammer, mit der die Funktion bearbeitet und das Ergebnis im Ergebnisregister gespeichert wird. Die letzte Anweisung speichert das Aktuelle Ergebnis (AE) des Ergebnisregisters in der Zielvariablen.

■ **Beispiel 11:** Formaler Funktionsaufruf einer Funktion

```
PROGRAM PRG
(*Deklaration*)
VAR_INPUT
  Wert1 AT %IX0.0 : BOOL ;
  Wert2 AT %IX1.0 : BOOL ;
  Wert3 AT %IX1.7 : BOOL ;
END_VAR
VAR_OUTPUT
  Wert4 AT %QX4.0 : BOOL ;
END_VAR
(*Bausteinrumpf*)
FC 10 (
  E0 := Wert1,
  E1 := Wert2,
  E2 := Wert3,
);
ST Wert4
END_PROGRAM
```

Beim „*nichtformalen Aufruf*“ muss zuerst der erste Eingangsparameter in das Ergebnisregister geladen werden. Dann kommt die Anweisung mit dem Namen der Funktion als Operator, gefolgt von den restlichen Eingangsparametern in richtiger Reihenfolge und durch Kommas getrennt im Operandenteil. Mit dem Abschluss dieser Anweisung berechnet die Funktion das Ergebnis und stellt es in das Ergebnisregister. In der letzten Anweisung wird das Aktuelle Ergebnis (AE) des Ergebnisregisters in der Zielvariablen gespeichert.

■ **Beispiel 12:** Nichtformaler Aufruf einer Funktion

```
PROGRAM PRG
(*Deklaration*)
VAR_INPUT
  Wert1 AT %IX0.0 : BOOL ;
  Wert2 AT %IX1.0 : BOOL ;
  Wert3 AT %IX1.7 : BOOL ;
END_VAR
VAR_OUTPUT
  Wert4 AT %QX4.0 : BOOL ;
END_VAR
(*Bausteinrumpf*)
LD Wert1
FC 10 Wert 2, Wert 3
ST Wert4
END_PROGRAM
```

9 Ablaufsteuerung

Steuerungsprogramme, die einen schrittweisen Prozessablauf nach den Vorgaben von *Ablauf-Funktionsplänen* ausführen, werden *Ablaufsteuerungen* genannt.

9.1 Ablauf-Funktionsplan

Der Ablauf-Funktionsplan ist eine eigenständige Planart zur prozessorientierten Darstellung von Steuerungsaufgaben. Eine verbale Aufgabenstellung soll aus Gründen der Klarheit und Vollständigkeit durch eine grafische Darstellung ersetzt werden, die bei der Planung, Inbetriebnahme und Störungssuche hilfreich sein soll. Ein richtig entworfener Ablauf-Funktionsplan muss bereits die Lösung einer entsprechenden Steuerungsaufgabe darstellen.

Geeignete Steuerungsaufgaben sind solche, bei denen unterscheidbare Aktionen in einer ereignis- oder zeitgesteuerten Reihen- oder auch Parallelfolge ablaufen und die auf Wiederholung gerichtet sind. Anschauliche Beispiele dafür sind Produktionsanlagen.

Der Ablauf-Funktionsplan stellt nur die grafischen Elemente für eine Ablaufbeschreibung zur Verfügung. Die Entwurfsmethode besteht darin, dass Steuerungszustände eingeführt und mit Aktionen verknüpft sowie Übergangsstellen zur Berücksichtigung von Steuersignalen vorgesehen werden.

Die Darstellung von Ablauf-Funktionsplänen kann auf zwei verschiedenen Normen beruhen:

- DIN EN 60848 GRAFCET, Spezifikationssprache für Funktionspläne der Ablaufsteuerung. (Nachfolger der DIN 40719-6). Diese Norm definiert eine grafische Entwurfssprache für die funktionale Beschreibung des Verhaltens des Ablaufteils eines Steuerungssystems. Die Entwurfssprache wird „GRAFCET“ genannt.
- DIN EN 61131, Speicherprogrammierbare Steuerungen, Teil 3: Programmiersprachen, hierin Elemente der Ablaufsprache (AS). Der Zweck der Ablaufsprache ist die Darstellung von Ablauffunktionen in SPS-Programm-Organisationseinheiten des Typs Funktionsbaustein oder Programm. Dazu gibt die Norm zwei Darstellungsvarianten für ihre Elemente an, eine ausführlich behandelte grafische Variante und eine textuelle Variante.

DIN EN 60848 und DIN EN 61131-3 haben jeweils ihren eigenen spezifischen Anwendungsbereich. Während die Entwurfssprache GRAFCET für die Beschreibung des Verhaltens unabhängig von einer speziellen Realisierung (elektronisch, elektromechanisch, pneumatisch oder gemischt) ist, legt IEC 61131-3 die Beschreibungsmittel der Ablaufsprache AS zwecks Programmrealisierung fest. Bei der Darstellung der Aktionsblöcke wird auf die Unterschiede eingegangen.

9.2 Grafische Darstellung von Ablaufsteuerungsfunktionen

Zur Darstellung von Ablaufstrukturen werden Ablauf-Funktionspläne verwendet, die auch als SFC (Sequential Function Chart) bezeichnet werden.

Darstellung von Schritten und Transitionen

Jeder mögliche Zustand einer Steuerung wird im Ablauf-Funktionsplan durch einen *Schritt* dargestellt. Ein Schritt ist entweder aktiv oder inaktiv und stellt einen Zustand der Steuerung dar. Ein Schritt muss grafisch durch einen Block dargestellt werden, der einen Schrittnamen in Form eines Bezeichners enthält, siehe Bild 19.

Eine Sonderstellung nimmt der Anfangsschritt ein, der durch eine doppelte Umrahmung gekennzeichnet ist. Der Anfangsschritt muss beim Start des Ablaufs

als einziger Schritt aktiv sein. Die Schritte sind durch *gerichtete Verbindungen* in Form von vertikalen Linien miteinander verbunden. Die Eigenschaft eines Schrittes, aktiv oder inaktiv sein zu können, setzt ein Speicherverhalten im Steuerungsprogramm voraus. Jedem Schritt ist normalerweise eine Aktion zugeordnet, die in einem Aktionsblock angegeben und mit dem Schritt verknüpft werden kann. Die Aktion ist nicht Bestandteil des Schrittes. Die Ablaufkette besteht aus einer Folge von *Schritten* und *Transitionen*, die einfach oder verzweigt sein kann.

Einfache Ablaufkette:

Der Wechsel von Schritt und Transition wird als Folge wiederholt. Es wird eine Kettenschleife gebildet, um wieder zum Anfang zurückzukehren. Die zurückführende Wirkungslinie kann durch eine Pfeildarstellungen mit Angabe von Schrittnummern ersetzt werden.

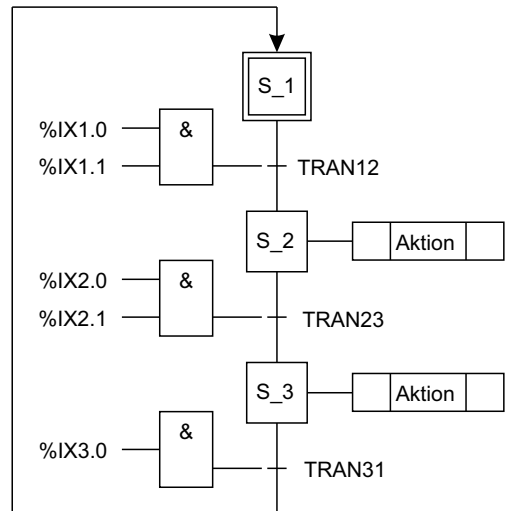


Bild 20 Einfache (lineare) Ablaufkette

Verzweigte Ablaufkette:

Man unterscheidet je nach Art der Verzweigung zwischen der Alternativ-Verzweigung und der Simultan-Verzweigung.

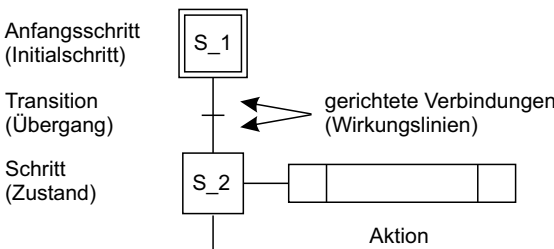


Bild 19 Darstellung von Ablaufschritten

- *Alternativ-Verzweigung* (1- aus n -Verzweigung, ODER-Verzweigung); es erfolgt die Auswahl und Bearbeitung nur eines Kettenstranges aus mehreren Kettensträngen. Am Verzweigungsanfang darf zur gleichen Zeit nur eine Transitionsbedingung wahr sein (Verriegelung) oder es muss eine Priorität vorgegeben werden, indem der Strang mit der niedersten Nummer die höchste Priorität hat. Zusätzlich wird mit einem Stern (✳) angegeben, dass die Transitionen von links nach rechts bearbeitet werden. Jedes Strang-Ende muss eine eigene Transitionsbedingungen zum Verlassen des Kettenstranges aufweisen. Anfang und Ende von Alternativ-Verzweigungen werden durch waagerechte Einfachlinien dargestellt.
- *Simultan-Verzweigung* (Parallelbearbeitung mehrerer Kettenstränge, UND-Verzweigung); es erfolgt die gleichzeitige Aktivierung der Anfangsschritte mehrerer Kettenstränge, die dann aber unabhängig voneinander bearbeitet werden. Alle Kettenstränge unterliegen auf der Anfangsseite nur einer vorgelagerten gemeinsamen Transitionsbedingung. Bei der Zusammenführung der Kettenstränge (Endeseite) darf nur eine gemeinsame Transitionsbedingung vorhanden sein. Anfang und Ende von Simultan-Verzweigungen werden durch waagerechte Doppellinien dargestellt.

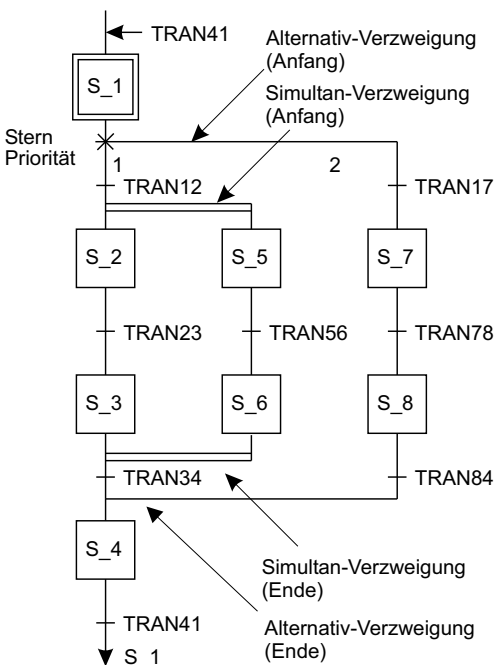


Bild 21 Verzweigte Ablaufkette

Wenn zum Beispiel der Anfangsschritt S_1 aktiv ist und die Transitionsbedingung von TRAN_17 den booleschen Wert TRUE liefert während die Transi-

onsbedingung von TRAN_12 den booleschen Wert FALSE hat, wird der Schritt S_7 aktiviert und Schritt 1 deaktiviert. Damit kann es in dieser Phase des Ablaufs nicht mehr zu einer Aktivierung der Schritte S_2 und S_5 kommen. Der Übergang von Schritt S_8 zu Schritt S_4 erfolgt erst, wenn die Transitionsbedingung von TRAN_84 den booleschen Wert TRUE annimmt. Schritt S_8 wird dann von S_4 zurückgesetzt.

Für den Fall, dass der Anfangsschritt S_1 aktiv ist und die Transitionsbedingungen von TRAN_12 und TRAN_17 den booleschen Wert TRUE haben, wird wegen der festgelegten Priorität die Simultan-Verzweigung bearbeitet. Es werden die Schritte S_2 und S_5 aktiviert und der Vorgängerschritt S_1 deaktiviert. Der Schritt S_4 wird erst erreicht, wenn die Vorgängerschritte S_3 und S_6 aktiv sind und die Transitionsbedingung von TRAN_34 den booleschen Wert TRUE annimmt. Die Schritte S_3 und S_6 werden dann von S_4 zurückgesetzt.

Schleifen und Sprünge

Ein *Sprung* führt unter Steuerung durch eine Transitionsbedingung von einem Schritt zu einem entfernten anderen Schritt, wobei der durch den Sprung gebildete Zweig keine Schritte enthält.

Eine *Schleife* kann die Folge eines Sprunges sein, indem unter Steuerung durch eine Transitionsbedingung auf einen Vorgängerschritt zurückgesprungen wird. Dabei muss verhindert werden, dass sich das Programm in einer Endlosschleife verfängt.

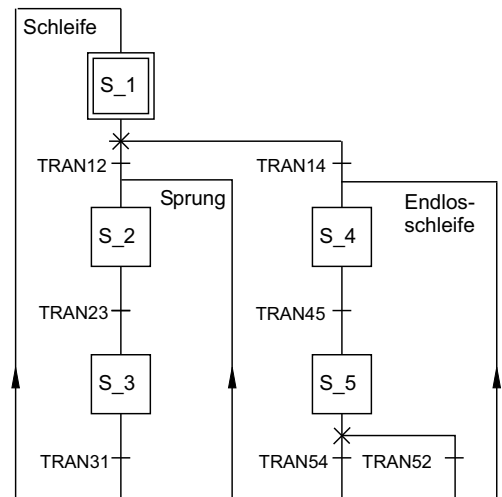


Bild 22 Sprung und Schleife

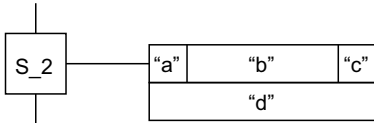
Aktionen, Aktionsblock

Mit einem Schritt ist in der Regel eine Aktion verbunden. Ein Schritt ohne zugehörige Aktion übt eine Warte-Funktion aus bis die nachfolgende Transitionsbedingung erfüllt ist. Da der Ablauf-Funktions-

plan beschreiben soll, was in der gesteuerten Anlage zu geschehen hat, wird in der Norm der Begriff *Aktion* anstelle von Befehl verwendet.

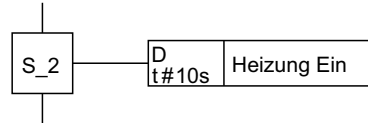
Der *Aktionsblock* ist ein grafisches Element zur Darstellung von Aktionen. Der Aktionsblock ist nicht Teil eines Schrittes und damit auch nicht Teil der Ablaufkette. Der Aktionsblock kann mit einem Schritt verknüpft oder als grafisches Element in einer Kontaktplan- bzw. Funktionsplandarstellung verwendet werden. Nachfolgend wird jedoch die Verknüpfung von Schritt und Aktionsblock bevorzugt.

In vollständiger Darstellung besteht der Aktionsblock aus vier Teilflächen, die nicht alle genutzt werden müssen:



- Feld "a": Bestimmungszeichen
- Feld "b": Aktionsname
- Feld "c": Anzeigvariable
- Feld "d": Beschreibung der Aktion in AWL; ST; KOP, FBS

Im einfachsten Fall wird in Feld "b" der Name einer booleschen Variablen und in Feld "a" ein zutreffendes Bestimmungszeichen eingetragen.



Die Aktion wird ausgeführt, wenn der zugehörige Schritt gesetzt ist und eine Aktionssteuerung die Freigabe erteilt. Die Aktionssteuerung sorgt für die richtige Umsetzung der im Feld "a" eingetragenen Bestimmungszeichen. Im Beispiel lautet die Aktion „Heizung EIN“ und es ist das Bestimmungszeichen D zusammen mit einer Zeitangabe im Feld "a" eingetragen. Ergebnis: Die Heizung wird verzögert eingeschaltet.

Es ist zulässig, mehrere Aktionen mit einem Schritt zu verbinden. Dies wird grafisch dargestellt durch aneinander gereihe Aktionsblöcke.

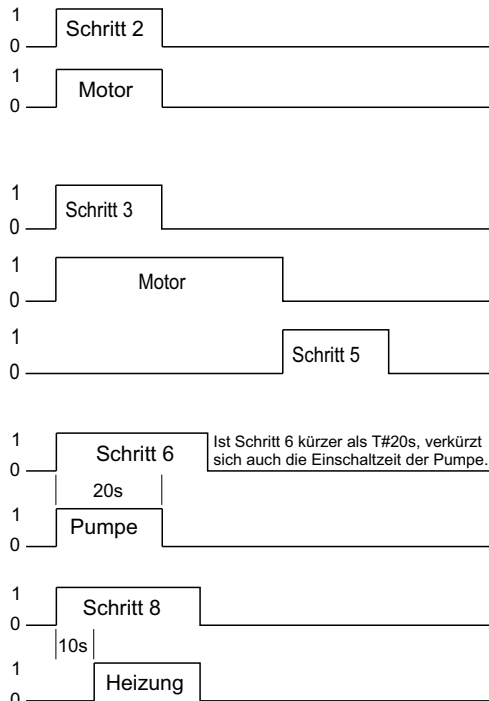
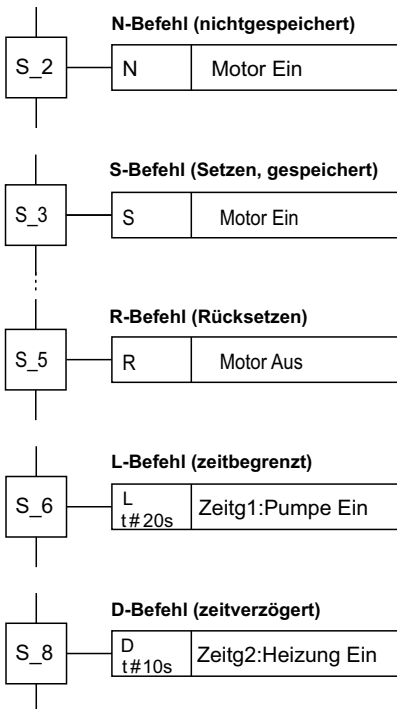
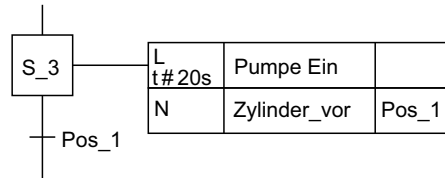
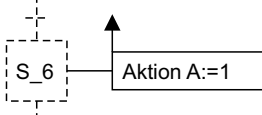
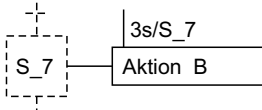
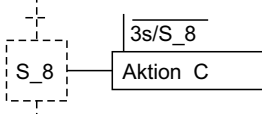
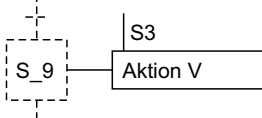


Bild 23 Bestimmungszeichen N, S, R, L und D für Aktionen nach IEC 61131-3

Tabelle 6 Symbole für Aktionen nach GRAFCET

	<p>Speichernde Aktion Aktion wird bei Aktivierung des Schrittes gespeichert. Beispiel: Setze den Wert der booleschen Variablen A auf TRUE</p>
	<p>Zeitverzögerte Aktion Die verzögerte Aktion ist eine kontinuierlich wirkende Aktion, bei der die Zuweisungsbedingung erst nach 3s erfüllt ist. Beispiel: Ausgang B nimmt den Wert TRUE an, nachdem drei Sekunden seit der Aktivierung von Schritt S_7 vergangen sind.</p>
	<p>Zeitbegrenzte Aktion Die begrenzte Aktion ist eine kontinuierlich wirkende Aktion, bei der die Zuweisungsbedingung während der Dauer von 3s erfüllt ist. Beispiel: Ausgang C erhält den Wert TRUE für drei Sekunden nach Aktivierung von Schritt S_8.</p>
	<p>Bedingte Aktion Die Zuweisungsbedingung S3 beeinflusst die kontinuierlich wirkende Aktion. Beispiel: Ausgang V erhält den Wert TRUE, wenn in Schritt 9 Bedingung S3 erfüllt ist. Als boolesche Gleichung: $V = S_9 \ \& \ S3$</p>

Im Aktionsfeld “c” kann eine boolesche AnzeigevARIABLE eingetragen werden, die durch die Aktion gesetzt werden kann, um die Erledigung oder einen Fehlerfall (z.B. Zeitüberschreitung) anzuzeigen oder als Weichschaltbedingung verwendbar ist.

Bestimmungszeichen für Aktionen (IEC 61131-3)

Die Verknüpfung einer Aktion mit einem Schritt erfolgt formal durch ein sog. Bestimmungszeichen. Für jedes Bestimmungszeichen (N, S, R, L, D u.a.) muss in der Ablaufsteuerung eine Programmteil mit einer entsprechenden Ansteuerlogik für die im Feld “b” angegebene boolesche Ausgangsvariable vorhanden sein. Im Bild 23 ist für einzelne Bestimmungszeichen die Art der Ansteuerlogik mit Liniendiagrammen beschrieben.

Die **DIN EN 60848** unterteilt den Aktionsblock nicht. Die Art der Aktionsausgabe wird dort durch Symbole am Aktionsblock festgelegt, siehe Tabelle 6.

Welche der Darstellungen verwendet wird, bestimmen in der Praxis die Vorteile für den jeweiligen Anwendungszweck. Ablaufsteuerungen werden mit speziellen Software-Tools der Automatisierungsfirmen entwickelt, die viel Detailarbeit übernehmen.

9.3 Betriebsartenteil und Bedienfeld

Ablaufsteuerungen in der Praxis bestehen aus Ablaufketten und verfügen über einen übergeordneten Betriebsartenteil für die Inbetriebnahme, Anlagenbetrieb und Störungsbeseitigung. Ablaufsteuerungen bestehen im Prinzip immer aus folgenden Komponenten:

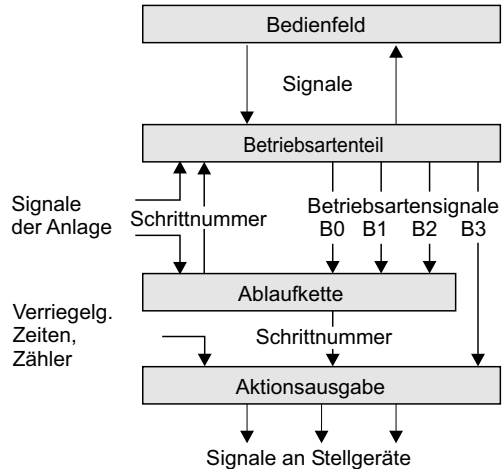


Bild 24 Struktur einer Ablaufsteuerung

Betriebsartensignale

Betriebsartensignale sind die vom Betriebsartenprogramm erzeugten Steuersignale für das Weichschalten und Rücksetzen der Ablaufkette sowie für die Freigabe von Aktionen.

• **Rücksetzen RESET (B0)**

RESET wirkt durch die Übergabevariable B0 auf einen Eingangsparameter RUECKSETZ des Ablaufkettenbausteins und versetzt die Schrittfolge in die Grundstellung oder wirkt über den RESET-Eingang

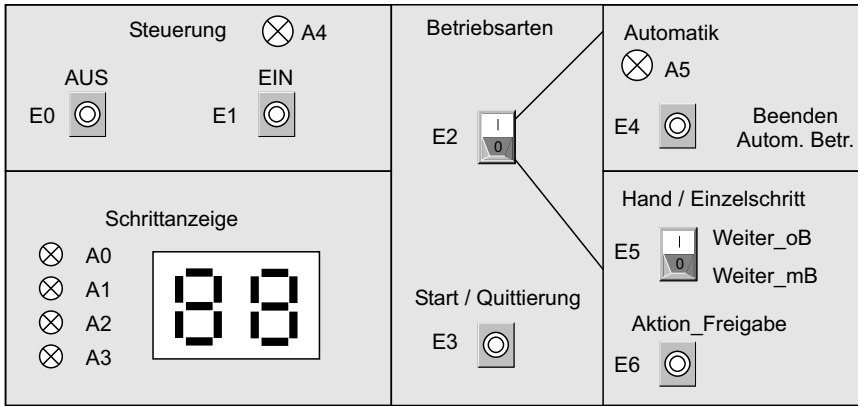


Bild 25 Beispiel eines Bedienfeldes für Ablaufsteuerung

des Aktionsbausteins und setzt dort Speicher, Zeitglieder und Zähler zurück.

- **Freigabe der Kette mit Bedingungen**
FREI_K_mB (B1)

FREI_K_mB wirkt durch die Übergabevariable B1 auf einen Eingangsparameter WEITER_mB im Ablaufkettenbaustein. Nur wenn B1 = 1 ist, wird bei erfüllten Weiterschaltbedingungen der nächste Schritt gesetzt. Das B1-Signal kann ein Impuls bei Hand-Einzelschrittbetrieb oder ein Dauersignal bei Automatikbetrieb sein.

- **Freigabe der Kette ohne Bedingungen**
FREI_K_oB (B2)

FREI_K_oB wirkt durch die Übergabevariable B2 auf einen Eingangsparameter WEITER_oB im Ablaufkettenbaustein. Nur wenn B2 = 1 ist, wird im Hand-Einzelschrittbetrieb der nächste Schrittspeicher gesetzt, ohne dass die Weiterschaltbedingung erfüllt sein muss. Das B2-Signal muss immer ein Impulssignal mit der Länge von einer Zykluszeit sein.

- **Freigabe Aktion FR_AKTION (B3)**

FR_AKTION wirkt durch die Übergabevariable B3 auf einen Eingangsparameter FREIGABE des Aktionsbausteins und kann dort einen Aktionsausgang freischalten bzw. sperren.

Funktionen eines Bedienfeldes

Eingriffe in Steuerungen sollen von einem Bedienfeld aus erfolgen, dessen Bedienoberfläche auf die Erfordernisse von Ablaufsteuerungen mit Betriebsartenteil abgestimmt ist.

Die Kenntnisnahme der Einzelfunktionen der Befehlsgeber und Anzeigen des beispielhaften Bedienfeldes verdeutlicht die Komplexität und Praxisnähe von Ablaufsteuerungen.

Taster E0: **Steuerung: AUS**

Taster E1: **Steuerung: EIN**

Hinweis: Statt der beiden Taster kann auch ein EIN-AUS-Schalter bzw. ein Schlüsselschalter verwendet werden.

Schalter E2: **Betriebsart**

Wahl der Betriebsart: E2 = 1 für Automatik, E2 = 0 für Handbetrieb.

Während eines Bearbeitungsprozesses kann zwischen Automatik und Handbetrieb umgeschaltet werden.

Taster E3: **Start/Einzelschritt**

Bei E2 = 1 (Automatik) werden durch Betätigung von E3 der Automatikbetrieb für das Weiterschalten der Ablaufkette mit Bedingungen sowie die Freigabe der Aktionen eingeschaltet und somit der automatische Ablauf der Kette gestartet. Bei E2 = 0 (Hand-/Einzelschrittbetrieb) erfolgt bei Betätigung von E3 eine Einzelschritt-Weiterschaltung der Schritt-kette.

Bei ausgeschalteter Steuerung kann durch Betätigen der Start/Quittierungstaste ein RESET zum Zurücksetzen der Schritt-kette und der gespeicherten Aktionen ausgelöst werden, wenn sich die Anlage in der Grundstellung befindet.

Taster E4: **Beenden Automatikbetrieb**

Vorwahl zum Beenden der Betriebsart Automatik bei Erreichen der Grundstellung der Ablaufkette (vollständig beendeter Bearbeitungsprozess).

Schalter E5: **Weiterschaltbedingung für Hand-/Einzelschrittbetrieb**

Bei E2 = 0 (Hand-/Einzelschrittbetrieb) muss die Art der Ablaufketten-Weiterschaltung festgelegt werden. Zur Auswahl stehen E5 = 1 (Weiterschalten

ohne Bedingungen) und $E5 = 0$ (Weiterschalten mit Bedingungen).

Taster E6: Aktions-Freigabe

In der Betriebsart Handbetrieb wird durch Taster E6 die Aktion des aktiven Schrittes solange ausgeführt, wie E6 betätigt ist.

10 Kommunikation in Automatisierungssystemen

Die Automatisierungstechnik befindet sich im Umbruch. Durch Anwendung moderner Kommunikationstechnik erhofft man sich entscheidende Verbesserungen bei Effizienz und Flexibilität der Automatisierungsprozesse. Der Begriff der Automatisierungstechnik umfasst heute mehr als nur das Automatisieren im Sinne von Steuern, Regeln und Visualisieren, eingeschlossen ist auch das Kommunizieren.

Zwischen den Anlagenkomponenten eines Automatisierungssystems müssen in der Regel Informationen ausgetauscht werden. Das sind im einfachsten Fall Signale von Sensoren und Aktoren, die zum übergeordneten Automatisierungsgerät gelangen müssen oder von dort herkommen. In anderen Fällen handelt es sich um Messwerte, Statusmeldungen und Diagnoseinformationen, die schon kompliziertere Daten darstellen. Realität ist auch, dass Auftragsdaten der Produktion zwischen Büro und Fertigungsanlage übertragen werden müssen. Es liegen also umfangreiche Kommunikationsbeziehungen in der Automatisierungstechnik vor, zu deren Bewältigung moderne Kommunikationssysteme verwendet werden.

10.1 Bussysteme

Die klassische Informationsübertragung von Prozesssignalen mittels analoger Spannungs- oder Stromwerte passt nicht zur digitalen Datenverarbeitung in den Automatisierungsgeräten. Es ist deshalb naheliegend, auch die Informationsübertragung auf eine digitale Grundlage zu stellen. Um gleichzeitig den Verkabelungsaufwand so gering wie möglich zu halten, überträgt man Daten über Bussysteme in Form serieller Zweidrahtverbindungen, an die alle Teilnehmer angeschlossen sind. Für den Informationsaustausch werden Telegramme mit entsprechenden Send- und Empfangsadressen sowie den Nutzdaten gebildet. Ein solches digitales Kommunikationssystem ist leichter erweiterbar durch Anschluss weiterer Teilnehmerstationen und Änderungen erfordern zum großen Teil nur softwaremäßige Eingriffe. Digitale Kommunikationssysteme vereinfachen nicht nur den sonst erforderlichen Verdrahtungsaufwand radikal, sondern bringen für den Anlagenbetreiber auch noch einen Zusatznutzen, indem außer den eigentlichen Messwerten nützliche Anlageninformationen wie z.B. Fehlermeldungen ohne großen Mehraufwand zusätz-

lich zur Verfügung stehen und Fernparametrierungen intelligenter Sensoren möglich sind.

Je nach Anwendungsbereich werden in der Automatisierungstechnik unterschiedliche Bussysteme eingesetzt, deren wichtigste Merkmale ihre sog. Echtzeitfähigkeit und Störsicherheit sind. Echtzeit bedeutet, dass die neuen Daten immer „rechtzeitig“ eintreffen. Zu spät eintreffende Daten können sonst zu gefährlichen Anlagenzuständen führen. Verwendet werden nur noch *standardisierte, offene Bussysteme*. Standardisiert bedeutet international genormt in der IEC 61158 und *Offenheit* gewährt den Zugang zu Spezifikationen und Technologien, damit sich neue Anbieter mit eigenen Produkten am System beteiligen können. Für die Nutzer ergibt sich so der Vorteil einer größeren Unabhängigkeit von Herstellern bei der Auswahl der zu vernetzenden Anlagenkomponenten.

Im prozessnahen Bereich der Anlage, der Feldebene, findet man überwiegend sogenannte **Feldbussysteme** mit Master-Slave-Kommunikation vor. Die Slaves sind die Buskomponenten, über die alle Eingangs- und Ausgangssignale der Anlage erfasst bzw. ausgegeben werden. Die Master-Station ist ein Kommunikationsprozessor, der für die zyklische Bedienung der zugeordneten Slave-Stationen sorgt, indem er die Daten von Eingängen der Slaves abholt bzw. an Ausgänge von Slaves ausliefert. Bekannte Master-Slave-Systeme für den Feldbereich sind der **AS-i-Bus** (Aktor-Sensor-Interface), der **PROFIBUS-DP** (Process Field BUS-Dezentrale Peripherie), der **INTERBUS-S** und andere Systeme. In der Feldebene zählen Bussysteme seit vielen Jahren zum Stand der Technik. In komplexeren Anlagen mit mehrerer SPSen kann jedoch auch ein Datenaustausch zwischen Steuerungsstationen (SPSen) erforderlich sein. Das erfordert bereits die höherwertige Master-Master-Kommunikation, die aber auch noch dem Produktionsbereich der Fabrik zuzurechnen ist. Muss ein noch weitergehender Datenaustausch unter Einbeziehung von Bürobereichen, z.B. der Fertigungssteuerung, verwirklicht werden, so wird eine zweite Netzinfrastruktur neben dem vorhandenen Feldbussystem erforderlich. Hierfür bietet sich dann das im Bürobereich bereits etablierte **Ethernet-TCP/IP-Netz** an. Schon lange wurde gefordert, das TCP/IP-Netz in die Fertigungsebene zu verlängern bei gleichzeitiger Erfüllung der Echtzeitbedingung. Ein Beispiel hierfür ist **PROFINET** als neues Kommunikationssystem. Bild 26 gibt eine Übersicht zu Bussystemen in der industriellen Kommunikation.

10.2 PROFINET – Offener Industrial Ethernet Standard

PROFINET bietet sich als ein offenes und durchgängiges Konzept für Automatisierungslösungen auf Ethernetbasis an, im Bereich von Einzelmaschinen bis hin zu modular aufgebauten Anlagen mit verteilter Steuerungszintelligenz und gewährleistet dabei

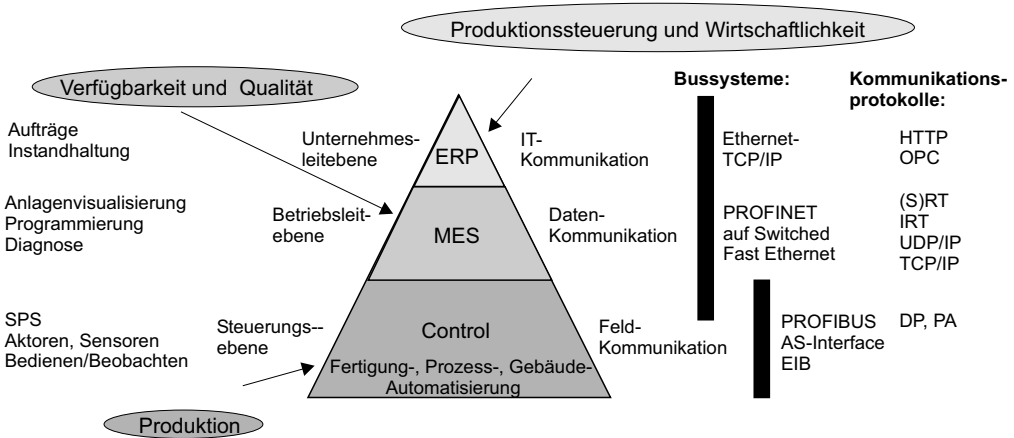


Bild 26 Industrielle Kommunikation
 ERP = Enterprise Resource Planning, MES = Manufacturing Execution Systems

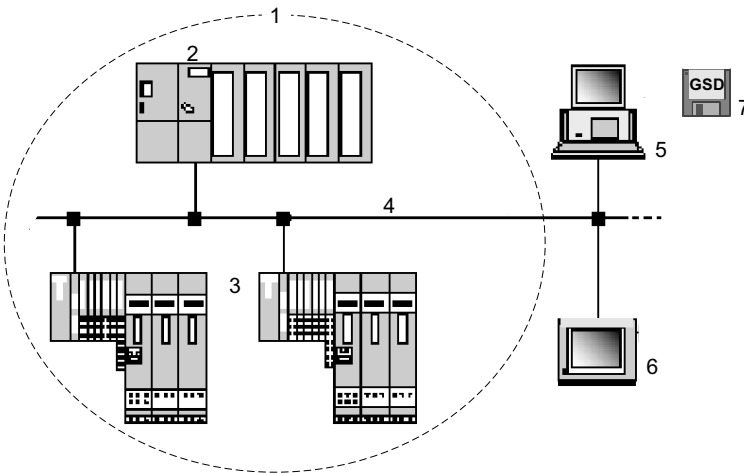


Bild 27 Bussystem PROFINET IO

Ziffer	PROFINET IO	PROFIBUS DP	Bemerkung
1	IO-System	DP-Mastersystem	Alle Geräte (IO-Controller, IO-Devices) und Kommunikationsverbindungen
2	IO-Controller	DP-Master	Gerät, über das angeschlossene Feldgeräte angesprochen werden
3	IO-Device	DP-Slave	Dezentrale Feldgeräte z.T. mit eigener CPU
4	Industrial Ethernet	Profibus	Netzwerkinfrastruktur mit Switches (im Bild 27 nicht dargestellt)
5	IO-Supervisor	PG/PC DP-Master Klasse 2	Programmieren, Inbetriebnahme/Diagnose
6	HMI = Human Machine Interface	HMI	Gerät zum Bedienen und Beobachten mit Zugriff auch auf IO-Devices über Ethernet!
7	GSD (XML-Datei)	GSD (ASCII-Datei)	Gerätebeschreibungsdatei für die IO-Devices und DP-Slaves

durch Einbinden von PROFIBUS DP einen Investitionsschutz.

Das Grundkonzept besteht aus

- PROFINET IO (dezentrale Feldgeräte) und
- PROFINET CBA (verteilte Automatisierung)

und schließt folgenden Leistungsumfang ein:

- Industrial Ethernet-Netzwerke mit aktiven Netzkomponenten (Switches, Router)
- Integration bestehender Feldbusysteme (PROFIBUS DP, INTERBUS,...)
- Kommunikationskanäle für anforderungsabhängige Übertragungsleistung
- Herstellerübergreifendes Engineeringkonzept (Projektierung, Programmierung)
- Einsatz von IT-Technologien (Netzwerkadministration, Webservers, E-Mail, OPC)
- Sicherheitsgerichtete Kommunikation

PROFINET IO

PROFINET IO ist die Kurzbezeichnung für das Steuerungskonzept „Dezentrale Feldgeräte“, dass mit einem Industrial Ethernet Netzwerk ein zentrales Steuerungsgerät (IO-Controller) mit dem dezentralen Feldgerätebereich (IO-Devices) verbindet und somit eine Ähnlichkeit mit dem weitverbreiteten Feldbusystem PROFIBUS-DP besitzt. Die Nutzdaten der Feldgeräte werden auch wieder zyklisch in Echtzeit in das Prozessabbild des IO-Controllers übertragen oder in umgekehrter Richtung an die IO-Devices ausgegeben. Das verwendete Kommunikationsmodell heißt Provider-Consumer-Verfahren und nicht mehr Master-Slave-Verfahren, obwohl der Datenverkehr mit den Feldgeräten in beiden Systemen nach dem gleichen Prinzip abläuft. PROFIBUS regelt den Buszugriff über die Token-Weitergabe, von der die DP-Slaves ausgeschlossen sind. Im Ethernet-System ist das nicht möglich, weil alle Teilnehmer am Netz beim Buszugriff gleichberechtigt sind. Der Provider sendet seine Daten ohne Aufforderung des Kommunikationspartners. Den IO-Devices wird deshalb beim System-Hochlauf mitgeteilt, dass sie mit einem Buszyklus von z.B. 10 ms mit aktuellen Daten versorgt werden. Die Eigenschaften der IO-Devices werden durch deren GSD-Datei (General Station Description auf XML-Basis) beschrieben, wie dieses auch von PROFIBUS DP her bekannt ist.

Die Steuerungsintelligenz in Form eines Anwenderprogramms befindet sich bei PROFINET IO oftmals nur im IO-Controller, kann aber auch teilweise in intelligenten IO-Devices (Feldgeräte mit eigener CPU) untergebracht sein, auch wieder vergleichbar mit dem PROFIBUS DP-System.

Das Bild 27 zeigt zusammen mit der anschließenden Gegenüberstellung der Grundbegriffe die Ähnlichkeit beider Systeme auf.

PROFINET-Kommunikationskanäle

Die PROFINET-Kommunikation findet über Industrial Ethernet statt. Dabei werden die folgenden *Übertragungsarten* unterstützt:

1. Zyklische Übertragung von zeitkritischen Daten (Nutzdaten)
2. Azyklische Übertragung von Engineering-Daten und zeitunkritische Parametrierungs-, Konfigurierungs- und Diagnose-Daten.

Für die genannten Übertragungsarten werden unterschiedliche *Transportprotokolle* verwendet, die man sich vereinfacht als Transportkanäle unterschiedlicher Leistungsstufen vorstellen kann. Bild 28 gibt eine Übersicht:

- **TCP/UDP-IP-Kanal** (Standard-Transportprotokoll der IT-Welt) für die Übertragung zeitunkritischer PROFINET-Daten. Dieser Transportkanal steht auch zur generellen Anbindung der Automatisierungssysteme an die übergeordneten Ethernet-Netze wie den Firmen-eigenen Intranets und dem öffentlichen Internet zur Verfügung.
- **SRT-Kanal** (Soft Real Time) für zeitkritische PROFINET-Daten. Hierbei handelt es sich um ein spezielles Transportprotokoll von PROFINET, um die im Feldbusbereich geforderte Echtzeitkommunikation zu ermöglichen. Werden Aktualisierungszeiten von ca. 10 ms bei zyklischer Datenübertragung gefordert, wird von „weicher“ Echtzeitbedingung gesprochen. Das SRT-Transportprotokoll wird als Software auf Basis vorhandener Controller realisiert.
- **IRT-Kanal** (Isochrone Real Time) für ganz besonders anspruchsvolle Anforderungen an die Übertragung von PROFINET-Daten wie beispielsweise für Antriebssteuerungen. Hier sind „harte“ Echtzeitbedingungen einzuhalten, d.h. Aktualisierungszeiten von ca. 1 ms bei einer garantierten Taktgenauigkeit bis auf 1 µs. Die IRT-Kommunikation ist zeit-schlitzgesteuert und setzt eine entsprechende Konfigurierung mit IRT-fähigen Geräten einschließlich der Switches voraus. IRT-fähige Switches schalten die Verbindungen zeitsynchronisiert (nicht adressgesteuert) bereits vor dem Eintreffen der Ethernet-Telegramme durch. Die Realisierung des IRT-Transportprotokolls erfolgt auf Hardware-Basis durch einen ASIC. PROFINET nutzt auch das Prinzip der „Telegramm-Priorisierung“, um die Übertragung der Daten durch das Ethernet-Netzwerk zu verbessern. Vordringlichere Telegramme sollen die weniger eiligen Telegramme überholen können. Netzwerkkomponenten wie Switches können den Datenfluss priorisierter Telegramme steuern, dazu verwenden sie Zwischenspeicher.

10.3 OPC-Technologie

Die OPC-Technologie (OLE for Process Control) bildet eine Datenbrücke zwischen einer Applikation,

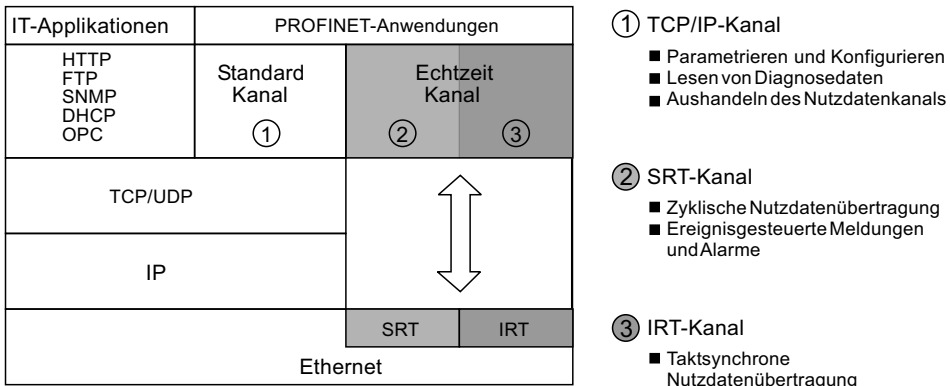


Bild 28 Kommunikationskanäle bei PROFINET

die Prozessdaten übergeordnet zu verarbeiten hat und einer Steuerungs-Hardware, die als gerätespezifischer Datenlieferant angesehen werden kann. OPC ist ein Client-Server-System.

Ein OPC-Server ist eine Software-Komponente, die der Hersteller einer SPS-Hardware für diese zur Verfügung stellt, damit von übergeordneten Anwenderprogrammen (z.B. MS Excel) aus auf die spezifische Hardware zugegriffen werden kann. Für den OPC-Server, der auf einem PC läuft, muss eine unterlagerte Kommunikationsverbindung zur Hardware des Herstellers eingerichtet werden, z.B. PROFIBUS oder Industrial Ethernet-TCP/IP.

Im Anwenderprogramm muss ein OPC-Client angelegt und konfiguriert werden, um auf einen OPC-Server und somit auf die Prozessdaten beispielsweise einer SPS lokal oder entfernt zugreifen zu können. Die Entwicklung eines OPC-Client-Programms kann beispielsweise mit Excel-VBA oder durch Einbindung eines fertigen ActiveX-Elements gelöst werden.

11 Steuerungssicherheit

Jede Maschinensteuerung ist mit einem Fehlerrisiko behaftet, dass sich durch besondere Maßnahmen bei der Entwicklung, Fertigung, Inbetriebnahme und Bedienung verringern aber nicht völlig ausschließen lässt. Fehlerursachen sind technisches oder menschliches Versagen.

Sicherheitsrelevante Maßnahmen sind nicht allein unter dem Aspekt der technischen Funktion (Wie funktioniert die Sicherheitsmaßnahme?) sondern besonders auch unter rechtlichen Gesichtspunkten (Welche Sicherheitsvorschriften gelten?) zu sehen. Hinter allem aber steht eine soziale Verantwortung des Geräteherstellers und Betreibers, der sich bewusst sein sollte, dass der Werker an einer Maschine davon ausgeht, dass diese sicher ist.

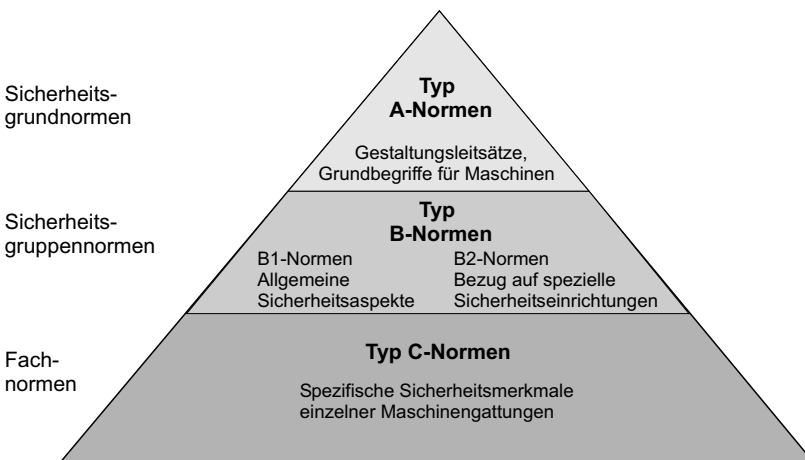


Bild 29: Hierarchie des europäischen Normenwerkes für Sicherheit von Maschinen

11.1 Europäische Richtlinien und Sicherheitsnormen

Drei für die Steuerungstechnik wichtige EG-Richtlinien seien hier besonders erwähnt:

- Niederspannungsrichtlinie 73/23/EWG: Schutz vor Gefahren durch elektrischen Strom bei Niederspannungsgeräten im Spannungsbereich 50 ... 1000 VAC, 75 ... 1500 VDC, CE-Kennzeichnungspflicht seit 1997.
- Maschinenrichtlinie 89/392/EG, letzte Fassung 98/37 EG: Grundlegende Anforderungen an die Sicherheit der Maschinen zum Schutz der Gesundheit des Betreibers. Inzwischen gilt diese Richtlinie auch für Sicherheitsbauteile, CE-Kennzeichnungspflicht seit 1995.
- EMV-Richtlinie 89/336/EWG (Elektromagnetische Verträglichkeit): Zwei grundlegende Anforderungen an die Geräte sind die sehr allgemein gehaltenen Grenzen für Störaussendung und Störfestigkeit bei Einstrahlung, CE-Kennzeichnungspflicht seit 1996.

Die europäischen Normen zur Sicherheit von Maschinen weisen eine dreigeteilte hierarchische Struktur auf:

- Typ A-Normen: Enthalten Gestaltungsleitsätze, die für alle Maschinen gültig sind.
- Typ B-Normen: Sind Sicherheits-Gruppennormen, die auf unterschiedliche Maschinengruppen anwendbar sind, wie z.B. DIN EN 60204-1: Elektrische Ausrüstung von Maschinen. B2-Normen behandeln spezielle Sicherheitseinrichtungen, wie z.B. DIN EN 418: NOT-AUS-Einrichtungen.
- Typ C-Normen: Sind Sicherheits-Fachnormen und beschreiben konkrete Anforderungen an einzelne Maschinenarten.

Für Normen von Typ C gilt das „Vermutungsprinzip“, d.h. bei Einhaltung dieser Normen darf vermutet werden, dass die Anforderungen der betreffenden EG-Richtlinien erfüllt sind.

11.2 Sicherheitsbegriff

Der Begriff *Sicherheit eines Steuerungssystems* ist auf die möglichen Folgen von auftretenden Fehlern bezogen, die Personen und Sachen betreffen. Davon zu unterscheiden ist der Begriff der *Verfügbarkeit* eines technischen Systems, die zwischen 0 und 100 % liegen kann, unabhängig von der Bedeutung der möglichen Folgen eines Ausfalls.

Um funktionale Sicherheit einer Maschine zu erreichen, ist es erforderlich, dass die sicherheitsbezogenen Teile der Schutz- und Steuereinrichtungen korrekt funktionieren und sich im Fehlerfall so verhalten, dass die Maschine in einem sicheren Zustand bleibt oder dorthin gebracht wird.

Das Maß für die erreichte funktionale Sicherheit wird in den Normen mit unterschiedlichen Wertigkeitsstufen ausgedrückt, und zwar mit:

- Safety Integrity Level in IEC 61508: SIL 1 bis SIL 4 (Höchststufe)
- Kategorien in der DIN EN 954-1: KAT 1 bis KAT 4 (Höchststufe)

Startfunktion

- Start-Funktionen müssen durch Erregen des entsprechenden Kreises erfolgen.

Stoppfunktionen

- Stopp-Funktionen müssen durch Entregen des entsprechenden Kreises erfolgen und haben Vorrang vor zugeordneten Start-Funktionen. Das Rücksetzen der Stopp-Funktion darf keinen Gefahr bringenden Zustand einleiten. Bei den Stopp-Funktionen werden die Kategorien 0, 1 und 2 unterschieden.

Anforderungen der Stopp-Kategorien:

1. **Kategorie 0** ist ein Stillsetzen durch sofortiges Abschalten der Energiezufuhr zu den Antrieben, d.h. ein *ungesteuertes Stillsetzen*, z.B. durch Betätigung des Motor Leistungsschalters und aller Bremsen (mechanischer Stillsetzeinrichtungen).

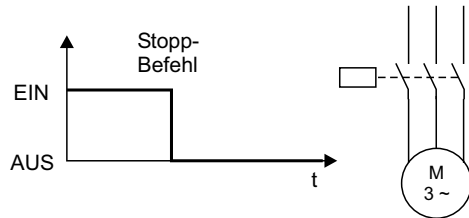


Bild 30 Stopp-Kategorie 0

2. **Kategorie 1** ist ein *gesteuertes Stillsetzen*, bei dem die Energiezufuhr zu den Maschinenantrieben beibehalten wird, um das gesteuerte Stillsetzen ausführen zu können. Die Energiezufuhr wird erst dann unterbrochen, wenn der Stillstand erreicht ist, z.B. Gegenstrombremsung von Drehstrommotoren.

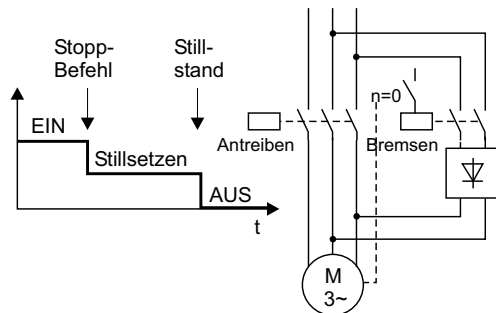


Bild 31 Stopp-Kategorie 1

3. **Kategorie 2** ist ein *gesteuertes Stillsetzen*, bei dem die Energiezufuhr zu den Maschinen-

Tabelle 7 Beschreibung der Anforderungen für Sicherheitskategorien

Kategorie	Kurzfassung der Anforderungen	Systemverhalten	Maßnahmen
B	<ul style="list-style-type: none"> Sicherheitsbezogene Teile von Steuerungen nach dem Stand der Technik. Bauteile müssen den zu erwartenden Einflüssen standhalten. 	Das Auftreten eines Fehlers kann zum Verlust der Sicherheitsfunktion führen.	z.B. 1-kanaliger Sicherheitskreis, Erdung des Steuerstromkreises
1	<ul style="list-style-type: none"> Anforderungen von „B“ müssen erfüllt sein. Einsatz bewährter Bauteile und Sicherheitsprinzipien. 	Das Auftreten eines Fehlers kann zum Verlust der Sicherheitsfunktion führen, aber höhere Zuverlässigkeit als in Kategorie B.	zusätzlich z.B.: zwangsöffnende und zwangsgeführte Kontakte
2	<ul style="list-style-type: none"> Anforderungen von „B“ müssen erfüllt sein. Einsatz bewährter Bauteile und Sicherheitsprinzipien. Testung der Sicherheitsfunktion in angemessenen Zeitabständen durch die Steuerung. 	Das Auftreten eines Fehlers kann zum Verlust der Sicherheitsfunktion zwischen den Prüfungsabständen führen. Der Verlust der Sicherheitsfunktion wird durch die Prüfung erkannt.	zusätzlich z.B.: Funktions-/Anlaufes-tung
3	<ul style="list-style-type: none"> Anforderungen von „B“ sind zu erfüllen, Einsatz bewährter Bauteile und Sicherheitsprinzipien. Einfehler-Sicherheit: Ein einzelner Fehler führt nicht zum Verlust der Sicherheitsfunktion. Der einzelne Fehler wird erkannt mit der Einschränkung „wann immer und in angemessener Weise durchführbar“. 	Wenn der einzelne Fehler auftritt, bleibt die Sicherheitsfunktion immer erhalten. Einige aber nicht alle Fehler werden erkannt. Eine Anhäufung unerkannter Fehler kann zum Verlust der Sicherheitsfunktion führen.	zusätzlich z.B.: 2-kanalige Ausführung von Sicherheitskreisen
4	<ul style="list-style-type: none"> Anforderungen von „B“ sind zu erfüllen, Anwendung bewährter Prinzipien. 1-Fehlerrisikosität ist gewährleistet. Erkennung des einzelnen Fehlers vor oder bei nächster Anforderung an die Sicherheitsfunktion (Selbstüberwachung). Falls die Erkennung des einzelnen Fehlers nicht möglich ist, darf eine Anhäufung von Fehlern nicht zum Verlust der Sicherheitsfunktion führen. 	Wenn Fehler auftreten, bleibt die Sicherheitsfunktion immer erhalten. Die Fehler werden rechtzeitig erkannt, um einen Verlust der Sicherheitsfunktion zu verhindern.	zusätzlich z.B.: Selbstüberwachung der Sicherheitskreise, Querschlusserkennung

antrieben erhalten bleibt, z.B. Anhalten durch Vorgabe von Sollwert „0“.

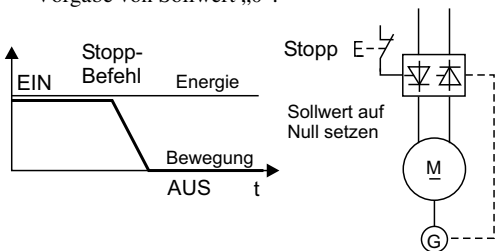


Bild 32 Stopp-Kategorie 2

Handlungen im Notfall (NOT_AUS)

Der neue Begriff *Handlungen im Notfall* ist Teil eines neuen Sicherheitskonzepts und steht für eine differenzierte Betrachtung der Handlungen, die im Notfall auszuführen sind. Im informativen Anhang der Norm DIN EN 60204-1 wird das noch in der Beratung befindliche Konzept erläutert. Eine Handlung im Notfall schließt einzeln oder in Kombination ein:

1. Stillsetzen im Notfall (Risiko durch einen Prozessablauf oder eine Bewegung)
2. Ingangsetzen im Notfall!

3. Ausschalten im Notfall (Risiko durch elektrische Gefährdung)
4. Einschalten im Notfall!

Die Norm macht nur Aussagen über *Stillsetzen im Notfall* sowie *Ausschalten im Notfall* und bestimmt, dass beide durch eine einzige menschliche Handlung auszulösen sind.

Es besteht eine enge Anbindung der Notfallsignale an die Stoppkategorien:

- *Stillsetzen im Notfall* muss entweder als Stopp der Kategorie 0 oder 1 wirken und über die Risikoanalyse bestimmt werden.

Für das Stillsetzen im Notfall der Kategorie 0 dürfen nur festverdrahtete, elektromechanische Betriebsmittel verwendet werden. Zusätzlich darf die Funktion nicht von einer elektronischen Schaltung (Hardware oder Software) oder von der Übertragung von Befehlen über ein Kommunikationsnetzwerk oder eine Datenverbindung abhängen.

Bei der Stopp-Funktion der Kategorie 1 für die Stillsetz-Funktion im Notfall muss die endgültige Abschaltung der Energie der Maschinen-Antriebs Elemente sichergestellt sein und muss durch Verwendung von elektromechanischen Betriebsmitteln erfolgen.

Anmerkung der Norm: Die Norm weist in ihrem deutschen Vorwort daraufhin, dass in Fällen, in denen andere Normen andere technische Lösungen zulassen als in DIN EN 60204-1 festgelegt ist, diese anderen technischen Lösungen zur Anwendung gelangen dürfen! Damit ist auch klargestellt, dass elektronische Betriebsmittel doch für NOT-AUS-Einrichtungen unabhängig von der Stopp-Kategorie eingesetzt werden dürfen, wenn diese unter Anwendung der Normen DIN EN 954-1 (Risikoanalyse) und/oder IEC 61508 die gleiche Sicherheit erfüllen, wie nach DIN EN 60204-1 gefordert.

- *Ausschalten im Notfall* sollte vorgesehen werden, wo ein Schutz gegen direktes Berühren aktiver Stromkreisteile (z.B. Schaltgeräte in elektrischen Betriebsräumen) nur durch Abstand oder Hindernisse erreicht wird oder wo es die Möglichkeit einer anderen Gefährdung durch elektrische Energie gibt.

Ein Ausschalten im Notfall wird durch Abschalten der Maschine von der Versorgung erreicht mit der Folge eines ungesteuerten Stillsetzens (Stopp-0-Kategorie) der Maschine. Ist das nicht zulässig, kann es notwendig sein, einen anderen Schutz z.B. gegen direktes Berühren vorzusehen, sodass ein Ausschalten im Notfall nicht notwendig ist.

12 Regelungstechnische Grundbegriffe der Automatisierungstechnik

Die folgenden Kapitel über regelungstechnische Grundbegriffe, Regelstrecken und Regler sind nur

eine enge Ausschnittsbetrachtung einer umfassenden Wissenschaft, die sich mit der gezielten Beeinflussung dynamischer Prozesse während des Prozessablaufs beschäftigt und dafür mathematische Methoden zur Systembeschreibung und -untersuchung entwickelt hat. Es geht hier nicht um die mathematische Behandlung von Regelkreisen, sondern im Kern nur darum, für eine bestimmte Regelungsaufgabe einen geeigneten Reglertyp auszuwählen und den Einfluss seiner Regelparameter auf das Verhalten im Regelkreis kennenzulernen.

12.1 Unterschied zwischen Steuern und Regeln, regelungstechnische Größen

Kennzeichen einer reinen Steuerung ist der sog. offene Wirkungsablauf, d.h. die Eingangssignale enthalten keine Rückmeldungen über ihre aktuelle Auswirkung auf den Prozess. Dieser Fall ist z.B. bei einer einfachen Drehzahlsteuerung eines Motors gegeben, wie in Bild 33 dargestellt. Mit dem einstellbaren Widerstand kann die Motordrehzahl gesteuert werden. Eine unterschiedlich schwere Last tritt als Störgröße auf und beeinflusst die tatsächliche Motordrehzahl. Eine solche Wirkungsstruktur bezeichnet man als *Steuerung*, im Sinne von Vorwärtssteuerung.

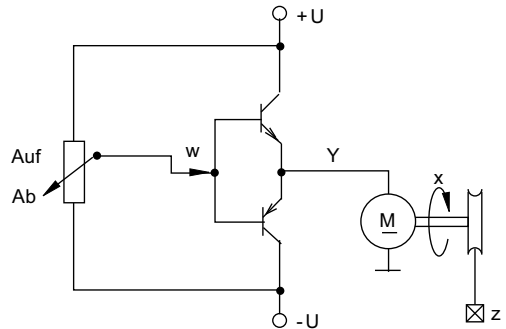


Bild 33 Drehzahl-Steuerung

Um die Darstellung unabhängig von einer bestimmten Anwendung zu machen, führt man allgemeingültige Symbole und steuerungstechnische Größen ein, wie Bild 34 zeigt, die man sich für den jeweiligen Anwendungsfall anschaulich übersetzen muss, z.B.:

- w = Führungsgröße \Rightarrow die Ausgangsspannung des Potenziometers (Sollwertvorgabe)
- y = Stellgröße \Rightarrow die Motorspannung;
- x = Steuergröße \Rightarrow die Drehzahl, bei der man einen gewünschten Sollwert und einen tatsächlichen Istwert zu unterscheiden hat;
- z = Störgröße \Rightarrow die unterschiedlichen Lasten.

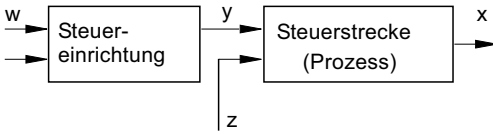


Bild 34 Struktur einer Steuerung

Im Gegensatz zum Typ der Vorwärtssteuerung hat eine Regelung einen Informationsrückfluss, sodass ein geschlossener Wirkungsablauf entsteht, den man als Regelkreis bezeichnet. Beim Motor in Bild 35 wird die tatsächliche Drehzahl zu einem Regler zurückgeführt. Hat sich die Drehzahl durch eine stärkere Belastung verringert, kann dies erkannt und nachgeregelt werden. Steuerungen mit Rückführung werden als *Regelungen* bezeichnet.

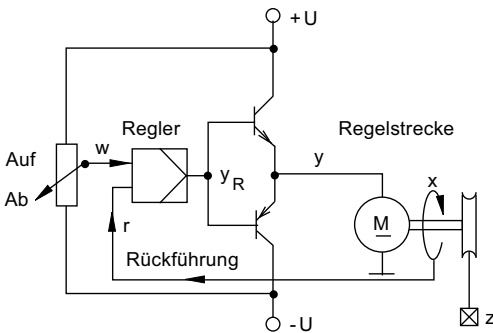


Bild 35 Drehzahlregelung

Regeln ist ein Vorgang, bei dem eine Größe, die man die *Regelgröße* nennt, fortlaufend erfasst und mit einer anderen Größe, die man als *Führungsgröße* bezeichnet, fortlaufend verglichen wird. Abhängig vom Ergebnis des Vergleichs muss das Regelungsprogramm eine Ausgangsgröße bilden, welche die Regelgröße im Sinne einer Angleichung an die Führungsgröße beeinflusst. Der sich dabei ergebende Wirkungsablauf findet in einem geschlossenen Kreis, dem *Regelkreis*, statt.

Das Erfassen der Regelgröße erfolgt durch Messen ihres aktuellen Wertes (Istwert). Die fortlaufende Erfassung bedeutet für eine SPS als Regler soviel wie

eine hinreichend häufige Abtastung des Messwertes. Die Angleichung der Regelgröße an die Führungsgröße wird durch das ablaufende Regelungsprogramm erreicht.

Die SPS als Regler führt dabei einen Vergleich zwischen *Führungsgröße* w und *Regelgröße* x (genauer: *Rückführgröße* r) durch und ermittelt dabei die *Regeldifferenz* e . Aus der Regeldifferenz wird über die Reglerfunktion das *Reglerausgangssignal* y_R ermittelt. Dieses ist eine leistungsarme Signalgröße, die einem Stellgerät zugeführt wird, das den passenden Anschluss an die Regelstrecke herstellt. Das *Stellgerät* besteht oft aus einem *Stellantrieb* und einem *Stellglied*. Der Stellantrieb bildet aus dem Reglerausgangssignal y_R die *Stellgröße* y und betätigt das Stellglied, welches den Massen- oder Energiestrom für die geregelte Anlage dosiert. Die *Regelstrecke* ist derjenige Teil des Regelkreises, in dem die Regelgröße geregelt wird. Sie beginnt am *Stellort*, also dort, wo die Stellgröße y in den Massen- oder Energiestrom eingreift und endet am *Messort*, wo sich der Messfühler zur Erfassung der Regelgröße x befindet. Der Stellantrieb wird zur Regeleinrichtung gezählt, weil er ein eigenes Zeitverhalten besitzt und damit das Gesamtzeitverhalten der Regeleinrichtung beeinflusst. Das Stellglied dagegen wird zur Regelstrecke gerechnet, weil es sich um deren Ventile, Schieber, Klappen oder elektronische Schalter (Transistoren, Thyristoren) handelt. Der Messfühler gehört ebenfalls zur Regelstrecke, denn er arbeitet nicht trägheitslos und beeinflusst damit das Zeitverhalten der Regelstrecke. Bild 36 zeigt die Struktur einer Regelung in ausführlicher Darstellung. Gebräuchlich sind aber auch vereinfachte Darstellungen, wenn sie ihren Zweck erfüllen.

In den Lösungen von Automatisierungsaufgaben kommen Steuerungen und Regelungen zusammen vor. Geregelt werden muss immer dann, wenn Größen auf genaue Werte geführt oder gegen den Einfluss veränderlicher Störgrößen z konstant gehalten werden müssen. So ist z.B. eine Ablaufsteuerung vom Prinzip her eine Steuerung. Wenn aber während der Zeitdauer eines Schrittes z.B. die Temperatur eines Mischgutes bei unterschiedlichen Füllmengen im Behälter konstant gehalten werden muss, kann dies

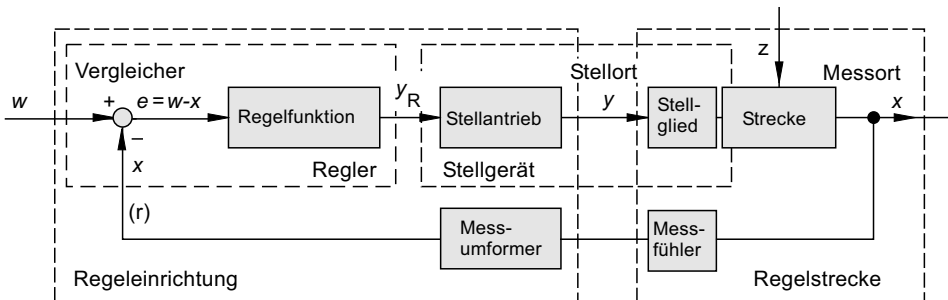


Bild 36 Ausführlich dargestellte Struktur eines Regelkreises

nur mit einer unterlagerten Regelung erreicht werden. Die SPS kann eine Vielzahl zeitgleich ablaufender Vorgänge steuern und regeln.

12.2 Regler-Technologien

In späteren Abschnitten werden Reglertypen nur noch im Sinne von ausführbaren Regelfunktionen, also recht abstrakt, beschrieben. Die technologische Ausführung von Reglern dagegen unterliegt der technischen Entwicklung.

Die ersten Regler waren schaltende Regler, die rein elektromechanische realisiert wurden und deren bekanntester Vertreter der noch heute verwendete *Bimetallkontaktregler* ist, wie er in einfachen Geräte, z.B. Toastern, vorkommt.

Es wurden dann Regler für anspruchsvollere Regelfunktionen benötigt, um sowohl analoge Eingangssignale verarbeiten als auch analoge Ausgangssignale ausgeben zu können. Analoge Signale sind solche, die innerhalb bestimmter Grenzen wie z.B. ± 10 V jeden beliebigen Wert annehmen können. Diese sog. *Analogregler* bestehen technologisch betrachtet aus einem Verstärker und benötigen eine Energiequelle. Verstärken ist dabei ein Vorgang, bei dem mit möglichst kleinem Energieaufwand am Verstärkereingang eine möglichst kräftige Energieabgabe der Energiequelle an den Verstärkerausgang signalgesteuert erfolgt. Als Verstärker für analoge Regler kommen elektronische aber auch pneumatische und hydraulische Elemente in Frage. Bei den elektronischen Operationsverstärkern werden die typischen Reglerfunktionen, die man mit den Kennbuchstaben P für proportional, I für integral und D für differenzial bezeichnet, durch spezielle Beschaltungsmaßnahmen erreicht, die den Anwender jedoch nicht interessieren müssen. Den Schlusspunkt dieser Entwicklung bildet der Universalregler in der Ausführung als analoger PID-Regler, der alle Funktionen ausführen kann, wie in Bild 37 schematisch dargestellt.

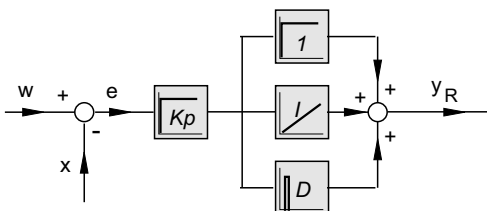


Bild 37 PID-Regelfunktion

Mit der Verfügbarkeit von preiswerten Mikrocontrollern lassen sich heute sog. digitale Regler mit entsprechender Software für den Regelalgorithmus sowie AD- und DA-Umsetzern für die Signalumwandlungen realisieren. Die Signale außerhalb des digitalen Reglers können nach wie vor analog sein, im Innern des Reglers werden nur digitale (zahlenmäßige) Signale verarbeitet. Bild 38 zeigt den prinzi-

piellen Aufbau eines digitalen Reglers, der auch für die SPS zutrifft.

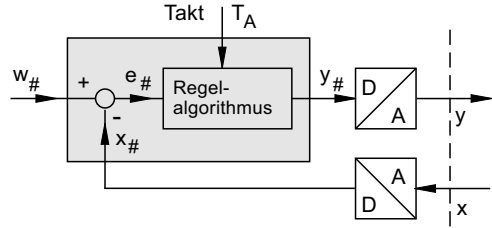


Bild 38 Vereinfacht dargestellter Aufbau eines Digitalreglers. Mit # gekennzeichnete Größen sind die zu Regelungssignalen proportionalen Zahlenwerte

Der digitale Regler führt eine taktgesteuerte Abtastregelung aus, weil er neue Eingangswerte erst entgegennehmen kann, wenn er einen Berechnungszyklus abgeschlossen hat, für den er eine gewisse Rechenzeit benötigt. Das hat zur Folge, dass neue Eingangsgrößen nur zu diskreten Zeitpunkten eingelesen und die Stellgröße nur zur diskreten Zeitpunkten ausgegeben werden können. Im Regelungsprogramm, das in gleichen Zeitabständen bearbeitet werden muss, wird aus dem Momentanwert der digitalisierten Regelgröße $x_{\#}$ und einer Führungsgröße $w_{\#}$ die Regeldifferenz $e_{\#}$ berechnet und daraus nach programmierten Funktionen die Stellgröße $y_{\#}$ gebildet. Die genau einzuhaltenden Zeitabstände werden als Abtastzeit T_A bezeichnet.

Da auch der Analog-Digital-Umsetzer zur Umwandlung eines Momentanwertes der Regelgröße eine gewisse Zeit benötigt, muss dieser Wert für den Wandlungszeitraum mit einem Halteglied konstant gehalten werden, um Umwandlungsfehler zu vermeiden. Bei der Digitalisierung der Regelgröße x im Analog-Digital-Umsetzer kann je nach seiner Stellenzahl nur eine bestimmte Unterscheidungs-fähigkeit für Signalwerte erreicht werden. Bei einer Stellenzahl von 8 Bit stehen nur $2^8 = 256$ mögliche Zahlenwerte zur Verfügung, sodass sich bei einem Analogsignalbereich von 0 bis 10 V nur Spannungsstufen von ca. 40 mV unterscheiden lassen.

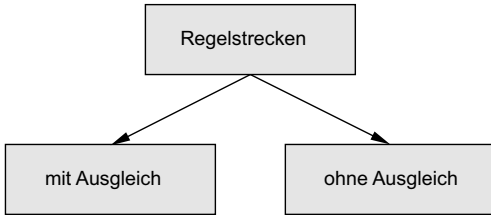
Auf Grund der wert- und zeitdiskreten Signalverarbeitung der digitalen Regler ergeben sich Grenzen hinsichtlich der Verarbeitung höherer Signalfrequenzen. Dafür können digitale Regler aber nicht nur die PID-Regelfunktion nachbilden, sondern ermöglichen auch die Realisierung neuartiger Reglertypen.

13 Regelstrecken

13.1 Beispiele für Regelstrecken

Das nachfolgend beschriebene Ordnungsschema soll helfen, um sich besser in der unübersehbaren Vielfalt möglicher Regelstrecken zurecht zu finden.

Erstes Ordnungskriterium: Regelstrecken mit und ohne Ausgleich



Als Beispiel für eine *Regelstrecke mit Ausgleich* sei die Temperaturregelung einer Heizungsanlage genannt. Bei Veränderung der Mischventilstellung erreicht die Raumtemperatur auf verändertem Niveau wieder einen stabilen Wert.

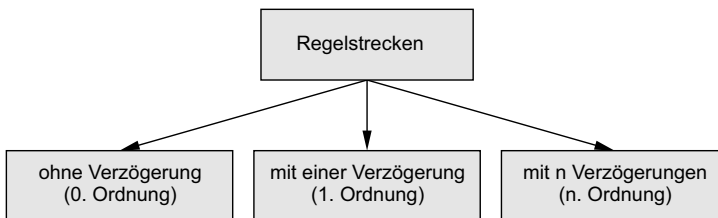
Ausgleich bedeutet, dass die Regelgröße x der Strecke nach einer sprunghaften Änderung der Stellgröße y innerhalb einer abzuwartenden Übergangszeit wieder einen stabilen Zustand, den sog. Beharrungszustand erreicht.

Regelstrecken mit Ausgleich und konstantem Übertragungsbeiwert K_{PS} haben einen proportionalen Charakter und werden deshalb auch *P-Strecken* genannt.

Bei einer *Regelstrecke ohne Ausgleich* würde die Regelgröße x nach einer sprunghaften Änderung der Stellgröße y keinen neuen Beharrungszustand finden. Dies ist z.B. der Fall bei einem Behälter mit dem Füllstand als Regelgröße x , wenn die Ablaufmenge in m^3/h durch eine Pumpe konstant gehalten wird. Jede Änderung der Zulaufmenge in m^3/h führt dann entweder zum Überlaufen oder Leerlaufen des Behälters.

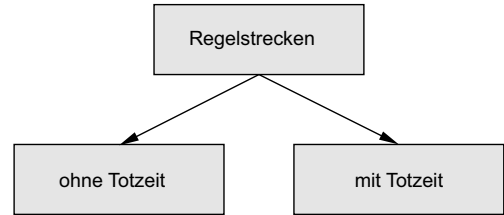
Regelstrecken ohne Ausgleich haben einen integralen Charakter und werden daher als *I-Strecken* bezeichnet.

Zweites Ordnungskriterium: Regelstrecken mit und ohne Verzögerung.



Verzögerung bedeutet, dass die Regelgröße x einer sprunghaften Änderung der Stellgröße y nicht sprunghaft folgen kann, sondern erst nach einer bestimmten Zeit einen neuen stabilen Wert erreicht. Verzögerungen treten bei technischen Prozessen immer auf, wenn Energie zu- oder abgeführt oder Massen beschleunigt oder abgebremst werden müssen.

Drittes Ordnungskriterium: Regelstrecken mit und ohne Totzeit.



Ein bekanntes Beispiel für eine Regelstrecke mit Totzeit ist das Förderband. Durch eine Schieberöffnung (Stellgrößenänderung Δy) gelang mehr Fördergut auf das Band. Die höhere Ausschüttmenge (Regelgrößenänderung Δx) wirkt sich am Ende des Förderbandes jedoch nicht sofort aus, sondern erst nach einer Totzeit, die von der Geschwindigkeit und der Länge des Bandes abhängt.

In der Praxis vorkommende Regelstrecken weisen zumeist Kombinationen von Eigenschaften auf. Die Kriterien Ausgleich, Verzögerung und Totzeit treten dann gemeinsam auf. Die nachfolgende Tabelle 8 zeigt Beispiele für Regelstrecken mit den genannten Ordnungskriterien. Die definierten Streckenparameter und typischen Sprungantworten werden im Anschluss erläutert.

13.2 Beschreibungsmittel zur Darstellung von Regelstreckeneigenschaften

Für die Auswahl eines geeigneten Reglers muss das Verhalten der Regelstrecke bekannt sein. Man versucht das Verhalten der Regelstrecken durch die Angabe von Kennlinien und Zeitfunktionen zu beschreiben, um daraus die Regelstreckenparameter abzuleiten.

Kennlinien von Strecken mit Ausgleich, Übertragungsbeiwert K_{PS}

Die Kennlinie einer Regelstrecke ist die grafische Darstellung des Zusammenhangs zwischen Stellgröße y und Regelgröße x . Eine *Kennlinie* ist somit ein Diagramm, bei dem die Abhängigkeit einer physikalischen Größe von einer anderen physikalischen Größe (nicht der Zeit) dargestellt ist. Auf der senkrechten Achse können z.B. die Raumtemperatur und auf der waagerechten Achse die Stellung des Mischventils aufgetragen sein: $T = f(\alpha)$.

Der *Übertragungsbeiwert* K_S der Regelstrecke ist durch den Quotienten der Änderung der Regelgröße x und der Änderung der Stellgröße y bestimmt. Im Signalfussplan entspricht dies dem Verhältnis der

Ausgangsgröße bezogen auf die Eingangsgröße. Damit kann bei bekanntem Übertragungsbeiwert der Regelstrecke bei einer Änderung der Stellgröße Δy die zu erwartende Änderung der Regelgröße Δx bestimmt werden. Die Zeit für die Änderung spielt dabei insofern keine Rolle, weil die zu jeder Stellgrößenänderung sich ergebende Regelgrößenänderung abgewartet werden muss (Erreichen des Beharrungszustandes). Man spricht deshalb auch von der statischen Kennlinie der Regelstrecke.

Die Kennlinie zeigt den Zusammenhang zwischen Regelgröße x und Stellgröße y innerhalb des Laufbereichs beider Größen bei konstantem Störgrößeneinfluss. Zu jedem Stellgrößenwert y ist der zugehörige Regelgrößenwert x im Beharrungszustand aufgetragen, siehe Bild 39.

Ist diese statische Kennlinie eine Gerade aus dem Achsenursprung, spricht man von einer P-Regelstrecke mit konstantem Übertragungsbeiwert K_{PS} (Index P für proportional, Index S für Strecke). Zur Berechnung von K_{PS} darf man deshalb auch den ganzen Regelbereich einsetzen:

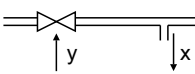
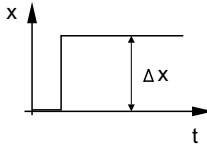
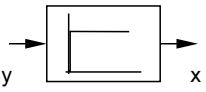
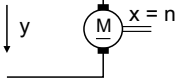
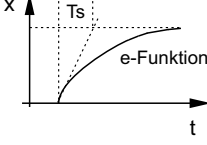
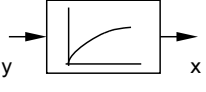
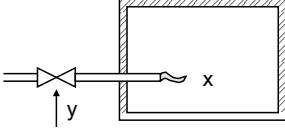
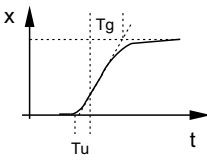
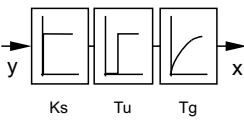
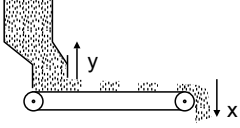
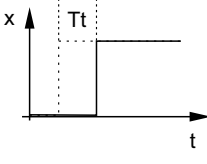
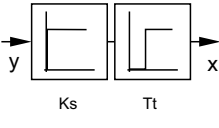
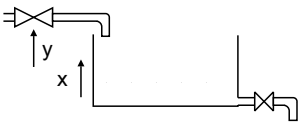
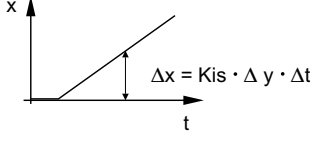
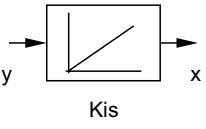
$$K_{PS} = \frac{\Delta X_h}{\Delta Y_h} \quad \text{mit}$$

X_h = Stellwirkung (Regelbereich der Strecke)

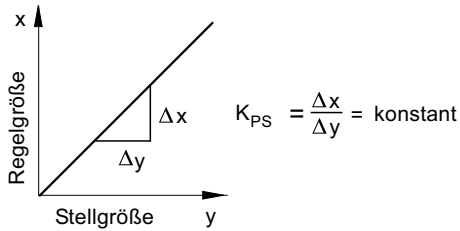
Y_h = Stellbereich des Reglers

Häufig haben Regelstrecken jedoch nichtlinearer Kennlinien, d.h. sie sind gekrümmt, sodass der Übertragungsbeiwert K_{PS} keine Konstante ist. Das möchte man vermeiden, weil solche Strecken viel schwieriger

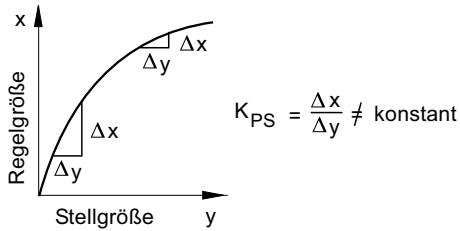
Tabelle 8 Regelstrecken

Art der Strecke Beispiel	Sprungantwort	Streckenparameter	Signalflussplandarstellung
<p>P-Strecke</p>  <p>Druck und Durchfluss in Flüssigkeitsrohrnetzen</p>		K_{PS}	
<p>PT1-Strecke</p>  <p>Drehzahl</p>		K_{PS}, T_S	
<p>PT2-Strecke</p>  <p>Ofentemperatur</p>		K_{PS}, T_u, T_g	
<p>P-Strecke mit Totzeit</p>  <p>Fördermenge</p>		K_{PS}, T_t	
<p>I-Strecke</p>  <p>Füllstand</p>		K_{IS}	

P-Regelstrecke mit linearer Kennlinie



P-Regelstrecke mit nichtlinearer Kennlinie

**Bild 39** Kennlinien von Regelstrecken mit Ausgleich

zu regeln sind. Abhilfe kann durch eine umgekehrt nichtlineare Kennlinie des Stellgliedes geschaffen werden. Dieser mögliche „Trick“ zeigt auch, warum das Stellglied zur Regelstrecke gehört.

Zeitverhalten von Regelstrecken mit Ausgleich

Ausgangspunkt der Betrachtung sei eine Regelstrecke mit Ausgleich und linearer Kennlinie. Damit ist die Regelstrecke noch nicht eindeutig beschrieben, denn es fehlt noch die Kenntnis über das Zeitverhalten der Strecke. Die zusätzliche Fragestellung lautet: Wie reagiert die Regelgröße x der Strecke auf eine plötzliche Änderung der Stellgröße um Δy dargestellt im zeitlichen Verlauf?

1. Regelstrecke mit Ausgleich 0. Ordnung: *P-Strecke*

Kennzeichen einer P-Strecke ist die proportionale Reaktion der Regelgröße x auf die Änderung der Stellgröße y . Wird die Stellgröße sprunghaft verändert, reagiert die Regelgröße ebenfalls sprunghaft. In der Tabelle 8 ist als Beispiel für eine P-Strecke ein Durchfluss in einer Rohrleitung angegeben. Wird durch stärkeres Öffnen eines Ventils am Anfang der Rohrleitung der Durchfluss erhöht, ist dies auch am Ende der Rohrleitung sofort in gleicher Weise der Fall, da die Flüssigkeit nicht komprimierbar ist. Die P-Strecke ist allein durch den Streckenparameter K_{PS} (Übertragungsbeiwert) gekennzeichnet.

2. Regelstrecke mit Ausgleich 1. Ordnung:

PT1-Strecke

Kennzeichen einer PT1-Strecke ist ein anfänglich schneller Anstieg der Regelgröße x , der sich dann

aber immer mehr verzögert und erst nach längerer Zeit den neuen Beharrungszustand erreicht. In der Tabelle 8 ist als Beispiel für eine PT1-Strecke die Drehzahländerung eines Motors angegeben. Der Motor hat auch ohne angeflanschte Arbeitsmaschine eine eigene Schwungmasse, die sich einer sofortigen Drehzahländerung widersetzt. Abstrakt ausgedrückt: Die PT1-Strecke ist eine Strecke mit Ausgleich und einem Energiespeicher, der die Verzögerung bewirkt. Die PT1-Strecke ist durch die Streckenparameter K_{PS} (Übertragungsbeiwert) und T_S (Verzögerungskonstante) gekennzeichnet. Die Verzögerungskonstante lässt sich durch Anlegen einer Tangente an die Sprungantwort ermitteln.

3. Regelstrecke mit Ausgleich 2. Ordnung:

PT2-Strecke

Kennzeichen einer PT2-Strecke ist ein anfänglich langsamer Anstieg der Regelgröße x , der sich dann aber beschleunigt und danach wieder verzögert bis der neue Beharrungszustand erreicht wird. Als Beispiel für eine PT2-Strecke wird in Tabelle 8 eine Heizung angegeben. Es sind zwei Speicher vorhanden. Wird der Heißwasserdurchfluss sprunghaft erhöht, muss sich erst der Heizkörper auf die neue Temperatur einstellen und dann erst wird mit dem Aufheizen der Raumluft begonnen.

Die Streckenparameter aus der Sprungantwort der PT2-Strecke sind K_{PS} (Übertragungsbeiwert) und die durch die eingetragene Wendetangente gebildeten Zeitkonstanten T_u (Verzugszeit) und T_g (Ausgleichszeit).

Das Ordnungszahlschema lässt sich entsprechend der Anzahl der in einer Regelstrecke vorkommenden Speicher noch weiter erhöhen.

4. Regelstrecke mit Ausgleich und Totzeit: PT_t

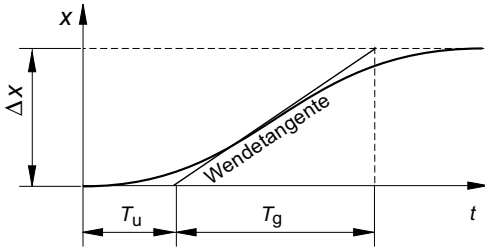
Kennzeichen eines Totzeitverhaltens ist, dass sich die Regelgröße x zwar in gleicher Weise sprunghaft ändert wie die Stellgröße, jedoch erst um die Totzeit T_t zeitverzögert. In der Tabelle 8 ist als Beispiel ein Förderband angegeben, bei dem eine gewisse Zeit vergeht, bis eine veränderte Schüttmenge am Ende des Förderbandes ankommt.

Die Streckenparameter der PT_t -Strecke sind K_{PS} (Übertragungsbeiwert) und T_t (Totzeit).

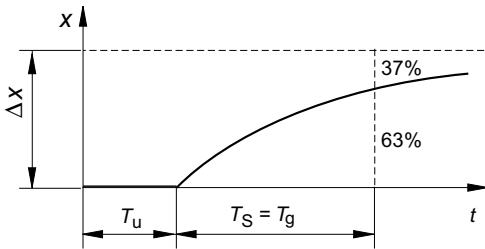
5. Regelstrecke ohne Ausgleich 0. Ordnung: *I-Strecke*

Bei den I-Strecken bildet sich die Regelgröße x durch Aufsummieren einer Stellgrößenänderung Δy über fortlaufende Zeitabschnitte Δt . Kennzeichen einer I-Strecke ist der zeitproportionale Anstieg oder Rückgang der Regelgröße, verbunden mit Überlauf oder Leerlauf, wenn nicht genau Abfluss gleich Zufluss ist. Als Beispiel für eine I-Strecke ist in der Tabelle 8 ein offener Behälter mit freiem Zulauf angegeben.

1. Aufzeichnung der Übertragungsfunktion, Eintrag der Wendetangente, Bestimmung der Verzugszeit T_u , Ausgleichszeit T_g und Änderung der Regelgröße Δx



2. Ersatzfunktion mit T_u als Totzeit und T_g als Verzögerungszeit T_s eines PT1-Gliedes bilden.



3. Die Regelstrecke kann näherungsweise durch ein P-Glied, ein Totzeitglied und ein PT1-Glied mit deren Streckenparametern K_{ps} , T_t , T_u , T_g beschrieben werden.

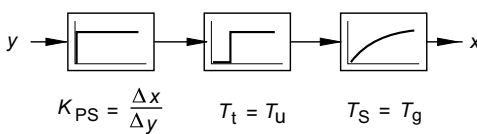


Bild 40 Bestimmung der Regelparameter einer unbekanntes Regelstrecke

Bestimmung der Regelstreckenparameter

Damit ein zur Regelstrecke passender Regler gefunden und günstig dimensioniert werden kann, müssen zuerst einmal die Streckenparameter der Regelstrecke bekannt sein. Dazu muss ein Signalfussplan mit dem zugehörigen Übertragungsbeiwert K_{ps} und den Zeitkonstanten ermittelt werden, der die zunächst noch unbekanntes Regelstrecke möglichst gut beschreibt. Das kann auf experimentelle Art durch Aufnahme der Sprungantwort auf einen Stellgrößen- oder Störgrößenprung geschehen.

Der Ablauf des Experimentes gliedert sich in folgenden Schritte:

1. Das Stellglied wird zum Zeitpunkt t_0 um den Wert Δy verstellt. Da es sich um eine Strecke mit Ausgleich handeln soll, wird die Regelgröße x einen um Δx verschobenen Beharrungszustand einnehmen. Die Antwort der Regelgröße x wird mit einem y - t -Schreiber aufgezeichnet und liegt dann als Liniendiagramm vor. An den Kurvenverlauf der Sprungantwort wird die Wendetangente eingezeichnet und aus den Achsenabschnitte die Verzugszeit T_u und die Ausgleichszeit T_g bestimmt.
2. Daraus kann eine Ersatzfunktion, bestehend aus einem Totzeitglied mit der Totzeit T_t und einem PT1-Glied mit der Verzögerungszeit T_s gebildet werden.
3. Die Übertragungsbeiwerte beider Glieder können als ein Übertragungsbeiwert K_{ps} in einem P-Glied zusammengefasst werden. Da es sich im einfachsten Fall um eine Regelstrecke mit Ausgleich und linearer Kennlinie handeln soll, lässt sich der Übertragungsbeiwert aus

$$K_{ps} = \frac{\Delta x}{\Delta y}$$

berechnen. Das nebenstehende Bild 40 zeigt das Vorgehen anschaulich.

Erfahrungen haben ergeben, dass das Verhältnis von Verzugszeit T_u zu Ausgleichszeit T_g Auskunft über die Regelbarkeit der Strecke ergeben:

gut regelbar	noch regelbar	schlecht regelbar
$\frac{T_u}{T_g} < \frac{1}{10}$	$\frac{T_u}{T_g} = \frac{1}{5}$	$\frac{T_u}{T_g} > \frac{1}{3}$

14 Regler

Aufgabe der Regler ist es, bei Abweichung der Regelgröße x von der Führungsgröße w die Reglerausgangsgröße y_R so zu verändern, dass die Regelgröße x in möglichst kurzer Zeit optimal an die Führungsgröße w angeglichen wird. Ursache für die Abweichung kann eine aufgetretene Störgröße z oder die veränderte Führungsgröße w sein.

In traditioneller Einteilung der Regler unterscheidet man

- stetige und unstetige Regler
- oder
- schaltende Regler (Zweipunkt-, Dreipunktregler),
- analoge Regler und
- digitale Regler.

Aus Gründen der Umfangsbeschränkung soll hier auf die Darstellung der schaltungs- und gerätetechnischen Ausführungen von Reglern verzichtet werden. Es werden nur die typischen Eigenschaften bekannter Reglertypen anhand ihrer Regelfunktion und durch ihre sog. Sprungantworten beschrieben, siehe Übersicht in Bild 42. Danach wird übergeleitet zum PID-Regelalgorithmus für den digitalen Abtastregler, der als Softwareregler in einer SPS realisiert werden kann.

14.1 P-Regler, P-Regelfunktion

Der P-Regler ist ein proportional wirkender Regler, der in seinem Arbeitsbereich, den man als Proportionalbereich bezeichnet, durch die Regelfunktion

$$y_R = K_{PR} \cdot e \quad \text{mit} \quad K_{PR} = \text{Proportionalbeiwert}$$

beschrieben ist. Eine sprunghafte Änderung der Regeldifferenz e am Eingang führt ohne Zeitverzögerung zu einer proportionalen sprunghaften Änderung der Reglerausgangsgröße y_R am Ausgang, solange sich die Regeldifferenz e innerhalb des Proportionalbereichs bewegt.

Setzt man die Regeldifferenz $e = w - x$ in den Regelfunktion ein, so erhält man:

$$y_R = K_{PR} \cdot (w - x)$$

Diese Beziehung lässt erkennen, dass der P-Regler nicht das leisten kann, was von Reglern allgemein erwartet wird, nämlich die genaue Heranführung der Regelgröße x an die Führungsgröße w . Bei $x = w$ wird die Reglerausgangsgröße $y_R = 0$ und damit erhält die Regelstrecke kein Stellsignal y , um die Regelgröße x bilden zu können: Eine Heizung, der kein Wärmeträger zugeführt wird, kann keinen Temperaturwert bilden.

Andererseits kennzeichnet den P-Regler eine direkte Zuordnung von Regelgröße x zu Stellgröße y_R bei fest eingestelltem Proportionalbeiwert K_{PR} und vorgegebener Führungsgröße w , solange sich die Regelgröße x noch innerhalb des Proportionalbereich X_p befindet. Der Proportionalbereich X_p lässt sich aus dem Stellbereich Y_h und dem Proportionalbeiwert K_{PR} berechnen:

$$X_p = \frac{Y_h}{K_{PR}}$$

Die feste Zuordnung von x und y_R kann anschaulich als statische Kennlinie des P-Reglers dargestellt werden, deren Achsenabschnitte sich aus

$$y_R = K_{PR} \cdot (w - x)$$

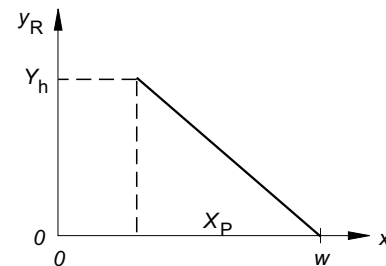
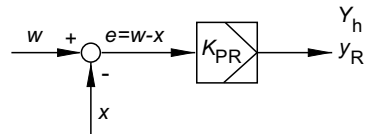
berechnen lassen:

Für $y_R = 0 \quad \Rightarrow x = w$

Für $y_R = Y_h = X_p \cdot K_{PR} \Rightarrow x = w - X_p$

Das Zahlenbeispiel zeigt, dass bei einem P-Regler die Eingangsgröße x und Ausgangsgröße y_R einander starr zugeordnet sind, solange sie sich im Proportionalbereich befinden. Wird in Folge einer Störeinwirkung z auf die Regelstrecke eine Änderung der Reglerausgangsgröße Δy_R erforderlich, dann ist dies nur bei einer gleichzeitigen Änderung der Regelgröße Δx möglich. Der P-Regler kann also eine Störung nicht völlig ausregeln. Der neue Beharrungszustand ist ein anderer Punkt auf der y_R - x -Kennlinie des P-Reglers, in deren fallenden Verlauf die Umkehrung des Wirkungssinns der Regelung zum Ausdruck kommt.

Die Änderung der Regelgröße Δx bei einer erforderlichen gleich großen Änderung der Reglerausgangsgröße Δy_R lässt sich verringern, wenn der Proportionalbereich X_p verkleinert wird, indem man K_p vergrößert. Dieser Verbesserung des P-Reglers sind aber Grenzen gesetzt, weil der Regelkreis instabil wird. Der P-Regler mit einem großen K_p -Wert reagiert überempfindlich auf kleinste Regelgrößenänderungen mit zu starken Ausschlägen bei der Stellgröße, die dann in extremer Weise ständig zwischen $y_R = 0$ und $y_R = Y_h$ umschaltet.



Zahlenbeispiel:

$$w = 3,75V, \quad K_{PR} = 4, \quad Y_h = 10V$$

	x	e	y_R
X_p -Bereich \updownarrow	1,25V	2,5V	10V
	1,75V	2,0V	8V
	2,25V	1,5V	6V
	2,50V	1,25V	5V
	2,75V	1,0V	4V
	3,25V	0,5V	2V
	3,75V	0,0V	0V

$$X_p = \frac{Y_h}{K_{PR}} = \frac{10V}{4} = 2,5V$$

Bild 41 Kennlinie des P-Reglers

Der P-Regler als einfachster kontinuierlicher Regler mit sehr schnellem Reglereingriff kann überall dort gut verwendet werden, wo die Strecke einen kleinen Regler-Proportionalbereich erlaubt. Das ist der Fall, wenn entweder nur geringe Störgrößen z auftreten oder das Verhältnis der Streckenzeitkonstanten T_g/T_u groß ist und geringe bleibende Regeldifferenzen Δx hinnehmbar sind.

14.2 I-Regler, I-Regelfunktion

Der I-Regler ist ein integral wirkender Regler, der die Reglerausgangsgröße y_R in vielen kleinen Stellschritten solange ändert, bis eine aufgetretene Regeldifferenz $e = w - x$ vollständig beseitigt ist. Dabei ist die Änderungsgeschwindigkeit der Stellgröße direkt proportional zur noch vorhandenen Regeldifferenz e und einem Übertragungsbeiwert K_{IR} :

$$\frac{dy_R}{dt} = K_{IR} \cdot e$$

In der Arbeitsweise gleicht der I-Regler einem Boten, dessen Laufgeschwindigkeit von der Dringlichkeit des Auftrags abhängig ist, wenn man die Regeldifferenz e als Dringlichkeitsmaß deutet. Solange noch eine Regeldifferenz besteht, ist der I-Regler damit beschäftigt, sie zu beseitigen.

Summiert man die unendlich vielen kleinen Stellschritte zur erzeugten Reglerausgangsgröße y_R , so erhält man die I-Regelfunktion:

$$y_R = K_{IR} \cdot \int e \cdot dt$$

mit K_{IR} = Integrierbeiwert (Übertragungsbeiwert)

In der Regelfunktion ist K_{IR} als Integrierbeiwert nicht ganz leicht verständlich zu deuten. K_{IR} beinhaltet eine Zeit T_I , die man als Integrierzeit bezeichnet, und die angibt, wie lange es dauert, bis der I-Regler den vollen Stellbereich Y_h durchlaufen hat, wenn als Eingangsgröße die Regelgröße x um den vollen Bereich X_h geändert wird:

$$K_{IR} = \frac{Y_h}{T_I \cdot X_h}$$

Die beiden Bereiche Y_h und X_h sind konstruktiv festliegende Werte, sodass als Einstellparameter des I-Reglers die Integrierzeit T_I in Frage kommt. Je kleiner die Integrierzeit T_I gewählt wird, umso schneller ändert sich die Reglerausgangsgröße y_R in Abhängigkeit von der Regeldifferenz e .

I-Regler arbeiten langsamer als P-Regler, lassen dafür aber keine bleibende Regeldifferenz bestehen. I-Regler sind ungeeignet für Regelstrecken mit großen Zeitkonstanten oder bei Strecken ohne Ausgleich. Für Regelstrecken, die nur durch eine Totzeit charakterisiert sind, ist der I-Regler von allen Reglertypen der am besten geeignete.

14.3 PI-Regler, PI-Regelfunktion

Die Eigenschaften von P-Reglern und I-Reglern ergänzen sich gut, weil P-Regler dynamisch besser sind als I-Regler und I-Regler statisch besser sind als P-Regler. Also ist es naheliegend, die beiden Regelverhalten zum PI-Regler zu kombinieren.

Die Regelfunktion des PI-Reglers ergibt sich aus der Addition der P-Regelfunktion und I-Regelfunktion:

$$y_R = K_{PR} \cdot e + K_{IR} \cdot \int e \cdot dt$$

Die Regelfunktion lässt erkennen, dass die Änderung der Reglerausgangsgröße y_R zum einen proportional zur Regeldifferenz e und zum anderen proportional zum Produkt aus Regeldifferenz und Zeit ist. Der erste Anteil sorgt für den schnellen und stabilen Reglereingriff und der zweite Anteil für die Beseitigung der bleibenden Regeldifferenz.

Eine bekanntere Darstellungsform der PI-Regelfunktion ersetzt den Integrierbeiwert K_{IR} durch einen Quotienten

$$K_{IR} = \frac{K_{PR}}{T_n}$$

mit dem Vorteil, den Proportionalbeiwert K_{PR} ausklammern zu können, sodass die PI-Regelfunktion nun in der Form

$$y_R = K_{PR} \cdot \left(e + \frac{1}{T_n} \cdot \int e \cdot dt \right)$$

entsteht. Die darin neu erscheinende Größe T_n wird *Nachstellzeit* genannt. Ausdrücken möchte man, dass der PI-Regler im Prinzip ein I-Regler ist, dessen Wirkungsbeginn aber um die Nachstellzeit T_n vorverlegt erscheint, also schneller eingreift. Tatsächlich beruht der schnellere Eingriff aber auf dem P-Anteil.

K_{PR} und T_n sind die Regelparameter (Einstellgrößen) mit denen der PI-Regler an die Regelstrecke optimal angepasst werden kann. Beide Parameter können einzeln oder gleichzeitig verstellt werden:

Lässt man zunächst K_{PR} konstant und verändert nur T_n , bleibt der P-Anteil in der Reglerausgangsgröße gleich hoch, es lässt sich aber die Stellgeschwindigkeit beeinflussen. Vergrößert man T_n , verringert sich die Stellgeschwindigkeit bei der Reglerausgangsgröße y_R . Wird die Nachstellzeit T_n soweit vergrößert, dass sie gegen unendlich geht, erhält man das Verhalten eines P-Reglers. Umgekehrt erhöht sich die Stellgeschwindigkeit, wenn man die Nachstellzeit verringert. Dieses darf nicht zu weit getrieben werden, weil es die Schwingneigung und Instabilität des PI-Reglers erhöht.

Was geschieht, wenn die Nachstellzeit T_n fest eingestellt bleibt und die Proportionalverstärkung K_{PR} verändert wird? Durch die Änderung von K_{PR} ändern sich beide Anteile der Reglerausgangsgröße im gleichen Verhältnis. Bei einer Vergrößerung von K_{PR}

Symbol	Signalflussplan	Sprungantworten	Bemerkung
<p>P-Regler</p>		<p>$X_P = \frac{y_h}{K_{PR}}$ P-Bereich</p>	<p>Vorteil: Günstiges Zeitverhalten</p> <p>Nachteil: Lastabhängige bleibende Regeldifferenz</p>
<p>I-Regler</p>		<p>$T_I =$ Integrierzeit</p> <p>$K_{IR} = \frac{y_h}{T_I \cdot X_h}$</p>	<p>Vorteil: Keine bleibende Regeldifferenz</p> <p>Nachteil: Ungünstiges Zeitverhalten</p>
<p>PI-Regler</p>		<p>$K_{PR} =$ Proportionalverstärkung</p> <p>$T_n =$ Nachstellzeit</p>	<p>Vorteile: Günstiges Zeitverhalten, Keine bleibende Regeldifferenz</p>
<p>PID-Regler</p>		<p>(Idealisierte Sprungantwort)</p>	<p>Vorteile: Schnelleres Ausregeln einer Regeldifferenz gegenüber dem PI-Regler</p> <p>Keine bleibende Regeldifferenz</p> <p>Geeignet für schwierige Regelstrecken</p>

Bild 42 Signalflusspläne und Sprungantworten der wichtigsten Reglertypen

erhöht sich der P-Anteil an der Gesamtverstellung und die Stellgeschwindigkeit. Der PI-Regler ist für den Einsatz in allen Streckentypen gut geeignet.

14.4 PID-Regler, PID-Regelfunktion

Der PID-Regler ist ein PI-Regler mit einem zusätzlichen D-Anteil (D = Differenzial). Die Vorstellung von der Wirkung eines D-Anteils in einem Regler entspricht einem kurz geschlagenen Konterschlag gegen eine sehr starke Änderung der Regelgröße x , die sich infolge einer Störgröße z einstellen würde. Der D-Anteil bewirkt eine sehr kräftige aber nur kurz wirkende Veränderung der Reglerausgangsgröße, die sich aus

$$\Delta y_R = K_{DR} \cdot \frac{de}{dt} \quad \text{mit } K_{DR} = \text{Differenzierbeiwert}$$

berechnet und dabei der zeitlichen Änderung der Regeldifferenz e proportional ist. In der Regelungstechnik wird die Eigenschaft eines Reglers, auf eine sich abzeichnende Entwicklung bereits im Voraus zu reagieren, als *Vorhalt* bezeichnet. Im Voraus bedeutet, der Regler beobachtet die Veränderung der Regeldifferenz bereits im aller ersten Ansatz bei ihrer Entstehung.

Die PID-Regelfunktion bildet sich aus der PI-Regelfunktion erweitert mit dem Vorhaltglied:

$$y_R = K_{PR} \cdot \left(e + \frac{1}{T_n} \cdot \int e \cdot dt + T_v \cdot \frac{de}{dt} \right)$$

Die Vorhaltzeit T_v ist vergleichbar der Nachstellzeit T_n eine eingesparte Zeit. Damit ist der PID-Regler im Prinzip ein PI-Regler, dessen Wirkungsbeginn um die Vorhaltzeit T_v vorverlegt erscheint, allerdings kann dies nicht in der Sprungantwort dargestellt werden. Der Vorteil des PID-Reglers gegenüber dem PI-Regler ist die etwas schnellere Beseitigung einer aufgetretenen Regeldifferenz.

Weil der PID-Regler alle bisher besprochenen Regelparameter, nämlich die Proportionalverstärkung K_{PR} für den P-Anteil, die Integrierzeit T_I bzw. die Nachstellzeit T_n für den I-Anteil und die Vorhaltzeit T_v für den D-Anteil enthält, wird er auch als Universalregler bezeichnet. Der PID-Regler ist wegen der Vielzahl der möglichen Einstellkombinationen zur Anpassung an alle Regelstreckentypen geeignet, aber etwas schwierig in der Handhabung.

14.5 Vergleich der verschiedenen Reglertypen

Als Abschluss der Betrachtung verschiedener Reglertypen soll in Bild 43 vergleichend gezeigt werden, wie sich die Regler bei optimaler Einstellung an einer Regelstrecke mit Ausgleich und den Kenngrößen K_{PS} ,

T_u und T_g verhalten, wenn die Auswirkung einer aufgetretenen Störgröße z auf die Regelgröße x ausgeregelt werden soll.

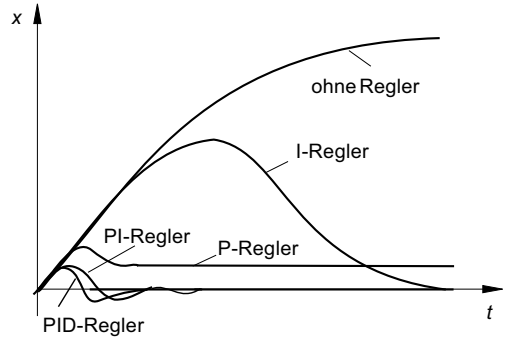


Bild 43 Unterschiedliches Ausregeln einer Störung bei verschiedenen Reglertypen an einer Regelstrecke mit Ausgleich

14.6 PID-Reglerbaustein für digitale Abtastregelung

Eine Faustformel besagt, dass digitale Abtastregler gute Regelergebnisse liefern, wenn die größte Strecken-Zeitkonstante T_s etwa 10 mal größer als die Abtastzeit T_A sein kann. Der Abtastzeitwert bestimmt sich aus der Programmbearbeitungszeit des Rechners einschließlich der Wandlungszeit des A/D-Umsetzers. Ein gut zu erreichender Abtastzeitwert bei einer SPS liegt bei 100 ms. Systembedingt kann sich bei Abtastreglern das Reglerausgangssignal y_R nur wertdiskret, also bestenfalls feinstufig, und zeitdiskret, nämlich im Takt der Abtastzeit T_A , ändern, wie Bild 44 zeigt.

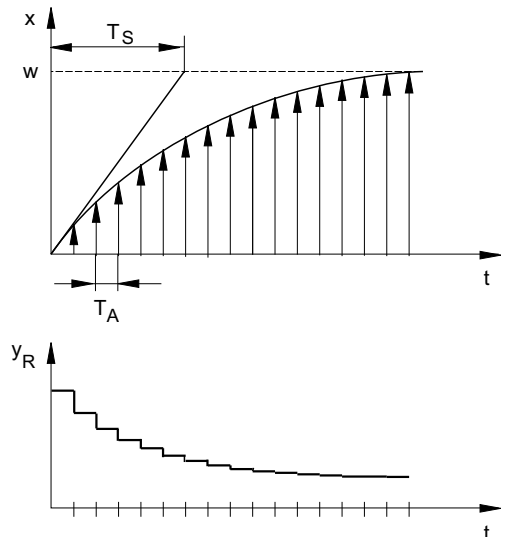


Bild 44 Istwertabtastung und Regelspannungsausgabe

Zusätzlich muss die PID-Regelfunktion an die arithmetischen Möglichkeiten eines digitalen Rechners angepasst werden. Die originale PID-Regelfunktion muss ersetzt werden durch einen wirkungsähnlichen PID-Regelalgorithmus, bei dem das Integral des I-Anteils durch eine Summe und das Differenzial des D-Anteils durch einen Differenzenquotienten ersetzt wird. Die differenziell kleine Zeitspanne dt in der Regelfunktion wird durch die Abtastzeit T_A ersetzt. Eine SPS als Softwareregler verfügt dann mit ihren Additions-, Subtraktions-, Multiplikations- und Divisions-Befehlen über die notwendigen Rechenoperationen zur Durchführung des PID-Regelprogramms, der die Stellgröße y_R für jeden Abtastzeitpunkt k neu berechnet:

$$y_R(k) = K_{PR} \cdot \left(e(k) + \frac{1}{T_n} \cdot \sum_{i=1}^k e(i) \cdot T_A + T_v \cdot \frac{e(k) - e(k-1)}{T_A} \right)$$

Die in der Formel noch vorhandene fortlaufende Summenbildung kann mit einem rekursiven Algorithmus ausgeführt werden, sodass eine Speicherung aller Regeldifferenzen e nicht erforderlich ist. Es ergibt sich eine Formel zur Berechnung der Reglerausgangsgröße $y_R(k)$ zu den bestimmten Zeitpunkten k und bezeichnet sie als *Stellungsalgorithmus*. Die analoge Wertausgabe dieser Reglerausgangsgröße am Abtastreglerausgang der SPS, also nach der Digital-Analog-Umsetzung, muss auf ein P-Stellglied geführt werden, das die richtige Stellgröße y an die Regelstrecke abgibt:

$$y_R(k) = K_{PR} \cdot \left(e(k) + \frac{T_A}{T_n} \cdot e_{SUM(k)} + T_v \cdot \frac{e(k) - e(k-1)}{T_A} \right)$$

PID-Stellungsalgorithmus

Digitale Abtastregler steuern häufig auch sog. I-Stellglieder an. Das sind motorische Stellantriebe, die erst durch ihren Vorschub die Stellgröße y bilden. Da solchen I-Stellgliedern nicht der Stellgrößenwert, sondern nur die Stellgrößenänderung zu geführt werden darf, gibt es auch noch einen zweiten PID-Algorithmus, der nur die Änderung der Reglerausgangsgröße Δy_R pro Abtastzeit T_A berechnet und als *Geschwindigkeitsalgorithmus* bezeichnet wird. Dieses Signal kann zur Bildung von entsprechend langen oder kurzen Schritt-Stellsignalen für die sog. Schrittregler verwendet werden.

$$\Delta y = K_{PR} \cdot \left(\begin{aligned} &e(k) - e(k-1) + \frac{T_A}{T_n} \cdot e(k) \\ &+ \frac{T_v}{T_A} (e(k) - 2e(k-1) + e(k-2)) \end{aligned} \right)$$

PID-Geschwindigkeitsalgorithmus

Nachfolgend wird die Umsetzung des beschriebenen Stellungsalgorithmus in ein universelles PID-Reglerprogramm gezeigt, dessen P-Anteil, I-Anteil bzw. D-Anteil einzeln zu und abschaltbar sind, um quasi-kontinuierliche P-, PI- oder PID-Regelfunktionen ausführen zu lassen. Bild 45 zeigt zunächst das Funktionsschema des zu realisierenden PID-Reglerbausteins.

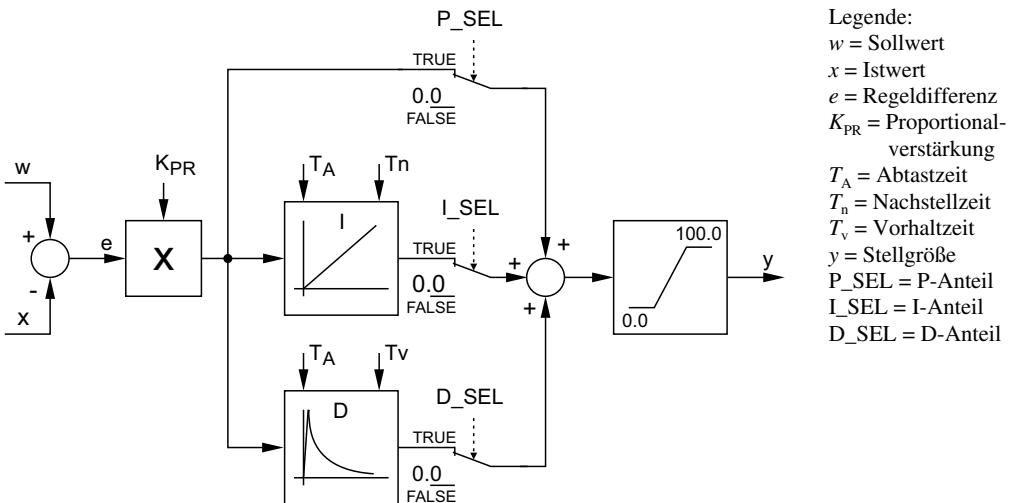


Bild 45 Funktionsschema eines PID-Reglerbausteins für Stellungsalgorithmus

Da die PID-Reglerfunktion über Speichereigenschaften verfügt, wird das Programm mit einem Funktionsbaustein realisiert. Über den Eingangsparameter EIN kann die Regelfunktion ein- oder ausgeschaltet werden. Im ausgeschalteten Zustand werden alle gespeicherten Werte zurückgesetzt. Weitere Eingangsparameter des Funktionsbausteins sind die Führungsgröße w (SW), die Regelgröße x (IW) sowie die Einstellwerte Proportionalbeiwert K_{PR} (KP), Nachstellzeit T_n (TN), Vorhaltzeit T_v (TV) und die Abtastzeit T_A (TA).

Es wird angenommen, dass die Führungsgröße w und die Regelgröße x als Gleitpunktzahlen im Bereich von 0.0 bis 100.0 vorliegen. Die Proportionalverstär-

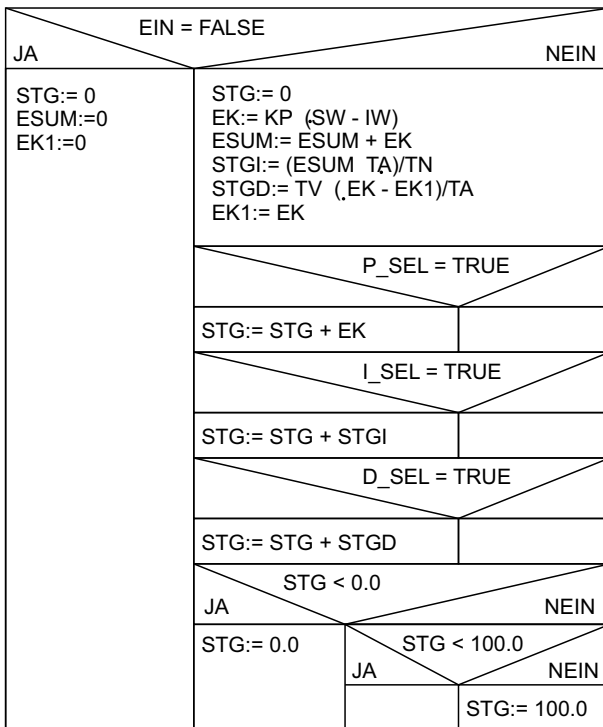
kung KP und die Zeitwerte TN, TV und TA sind ebenfalls als Gleitpunktzahlen anzugeben. Bei den Zeitwerten TN, TV und TA entspricht dabei der Zahlenwert einem Zeitwert in Sekunden.

Mit den binären Eingangsparametern P_SEL, I_SEL und D_SEL können die einzelnen Anteile mit einem „1“-Signal zugeschaltet und mit einem „0“-Signal abgeschaltet werden.

Die Ausgangsgröße STG des PID-Reglerbausteins ist die kontinuierliche Stellgröße y_R in einem Gleitpunktzahlenbereich von 0.0 ... 100.0.

Der PID-Reglerbaustein muss entsprechend der gewählten Abtastzeit T_A in gleichen zeitlichen Abständen zur Bearbeitung aufgerufen werden.

Struktogramm des PID-Reglerbausteins mit Stellungsalgorithmus:



Lokale stationäre Variablen des Funktionsbausteins:

EK1 = Regeldifferenz des vorhergehenden Abtastzeitpunktes mit dem Proportionalbeiwert KP multipliziert.

ESUM = Summe aller Regeldifferenzen, die mit dem Proportionalbeiwert KP multipliziert sind.

Lokale temporäre Variablen des Funktionsbausteins:

EK = Regeldifferenz mit dem Proportionalbeiwert KP multipliziert. Der Wert entspricht dem P-Anteil.

STGI = Berechneter I-Anteil der Stellgröße STG.

STGD = Berechneter D-Anteil der Stellgröße STG.

14.7 SPS als kontinuierlicher PID-Abtastregler

Der in 14.6 beschriebene PID-Reglerbaustein für den Stellungsalgorithmus ist noch kein gebrauchsfertiger digitaler SPS-Regler. Bild 46 zeigt in vereinfachter Form, was noch alles benötigt wird. Die SPS muss neben der CPU-Baugruppe selbstverständlich auch über eine Analogeingabebaugruppe und eine Analog-

ausgabebaugruppe verfügen. Programmäßig ist nicht nur der Regelalgorithmus, sondern auch das Einlesen der analogen Regelgröße x und die Ausgabe der analogen Stellgröße y auszuführen, d.h. die Anbindung der SPS an den Messaufnehmer und das Stellglied der realen oder simulierten Regelstrecke zu realisieren. Praktisch umsetzen lässt sich eine solche SPS-Reglerlösung nur mit fertigen Programmierbausteinen aus der SPS-Bibliothek.

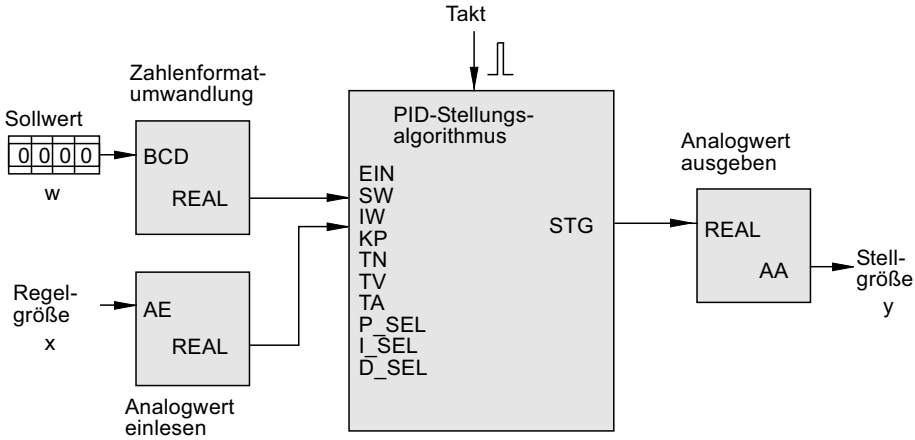


Bild 46 SPS-Regelstruktur (vereinfacht)

14.8 Digitaler Schritztregler mit PI-Verhalten

In der klassischen Analogregeltechnik wurde dieser Reglertyp mit einem Dreipunktregler und verzögerter Rückführung über dem Schaltverstärker realisiert, weshalb er dort auch als Dreipunkt-Schritztregler bezeichnet wurde. Die digitale Nachbildung dieses Reglertyps beruht nicht mehr auf dem Dreipunktregler, sondern auf dem PID-Geschwindigkeitsalgorithmus mit einer nachgesetzten Impulsausgabe zur Ansteuerung eines motorischen Stellantriebs, der einen Schieber oder ein Ventil mit einer Folge von Stellschritten öffnet oder schließt. Die so gebildete Stellgröße ist quasi-kontinuierlich, da jede Stellglied-

position innerhalb des Stellbereichs erreichbar ist, wenn auch in kleinen Schritten. Werden keine Impulse ausgegeben, bleibt die Stellgröße unverändert. Mit den beiden binär wirkenden Impulsausgängen können die erforderlichen drei Schaltzustände „Rechtslauf“, „Linkslauf“ und „Aus“ für den Stellantrieb realisiert werden.

Auf eine sprunghafte Änderung der Regeldifferenz e reagiert der PI-Schritztregler sofort mit einem „langen“ Schritt. Das Stellsignal y_1 gibt dazu einen langen Impuls aus, der durch den P-Anteil des Regelalgorithmus verursacht wird. Die darauffolgenden kürzeren Impulse werden durch den I-Anteil des Regelalgorithmus gebildet, siehe Bild 47. Der PI-Schritztregler wirkt solange, bis keine Regeldifferenz mehr besteht.

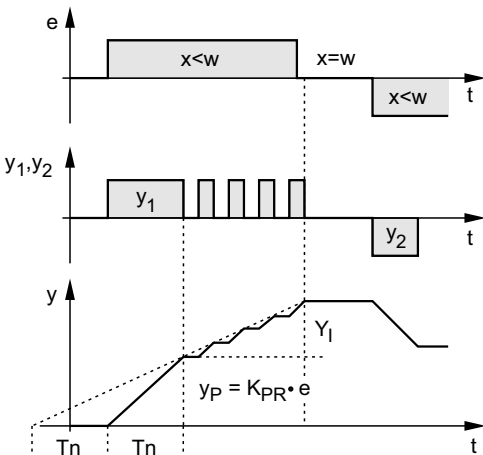


Bild 47 Impulsverhalten eines PI-Schritztreglers

14.9 Zweipunktregler, Zweipunkt-Regelfunktion

Bei der Zweipunkt-Regelfunktion wird ein binäres Stellgrößensignal mit den Zuständen „EIN“ und „AUS“ erzeugt. Bei „EIN“ erfolgt 100 % Leistungszufuhr für die Regelstrecke und bei „AUS“ dagegen 0 %.

Die Kennlinie des Zweipunktreglers zeigt die Position der beiden Schaltpunkte auf der Regelgrößenachse x , deren Mittenlage durch die eingestellte Führungsgröße w festgelegt ist. Die Zuschaltung der Leistungszufuhr erfolgt, wenn die Regelgröße x den unteren Grenzwert unterschreitet. Die Leistungszufuhr wird abgeschaltet, sobald die Regelgröße x den oberen Grenzwert überschreitet.

Es ist charakteristisch für Zweipunktregler, dass es für die Regelgröße x keinen Beharrungszustand gibt, dem sie zustreben kann. Die Stellgröße schaltet

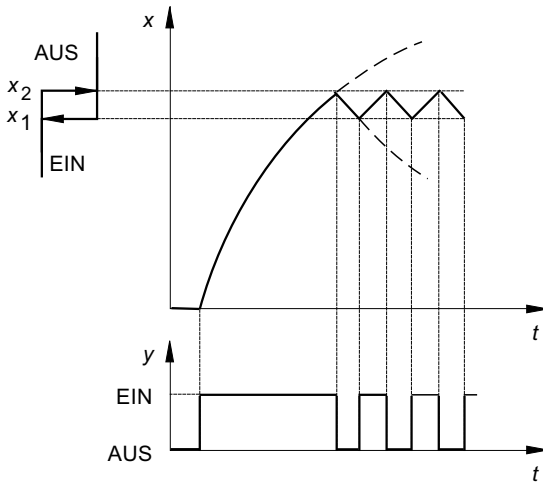


Bild 48
Zweipunkt-Regelfunktion mit Verlauf der Regelgröße

ständig um und die Regelgröße verändert sich fortwährend im Rahmen einer noch zulässigen Schwankungsbreite, die durch die Differenz der beiden Schaltpunkte x_2 und x_1 festgelegt ist. Diese sog. *Schalthysterese* trägt zur Beruhigung des Regelvorgangs bei. Den Einfluss einer Störgröße z korrigiert der Zweipunktregler durch Änderung seines EIN-AUS-Schaltverhältnisses.

Grafisch lässt sich der Zweipunkt-Regelungsvorgang anschaulich darstellen, wie Bild 48 zeigt.

14.10 Regelgüte

Ein Regler hat die Aufgabe, die Regelgröße x möglichst genau auf den durch die Führungsgröße w vorgegebenen Sollwert zu bringen und auch dort gegen den Einfluss einer Störgröße z zu halten. Außerdem wird gefordert, dass der Regelungsvorgang in möglichst kurzer Zeit und nur unter vorübergehenden

noch zulässigem Überschwingen abgeschlossen wird. Diese Forderungen lassen sich nicht alle erfüllen, da einzelne Bedingungen sich auch widersprechen. Dämpft man beispielsweise die Überschwingweite, vergrößert sich die Ausregelzeit. Es ist daher nur möglich für eine vorliegende Regelungsaufgabe die günstigsten Reglereinstellungen im Sinne eines Kompromisses zu finden.

Beurteilungsmaßstäbe für die Regelgüte sind die An- und Ausregelzeit sowie die Überschwingweite, wie in Bild 49 dargestellt.

1. Die Anregelzeit T_{an} ist die Zeit, die bis zum erstmaligen Erreichen des Sollwertes der Regelgröße vergeht.
2. Die Ausregelzeit T_{aus} ist die Zeit, die vergeht bis die Regelgröße x endgültig in den Toleranzbereich einmündet und darin verbleibt.
3. Die Überschwingweite $x_{ü}$ wird in Prozent des Sollwertes der Regelgröße x festgelegt.

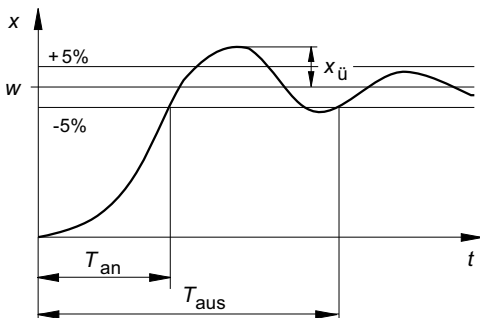


Bild 49 Kriterien zur Regelgüte

Literaturverzeichnis

DIN EN 61131-3, Speicherprogrammierbare Steuerungen, Teil 3: Programmiersprachen, Dezember 2003, Beuth Verlag Berlin

DIN 19226 Teil 4, Leittechnik; Regelungstechnik und Steuerungstechnik; Begriffe für Regelungs- und Steuerungssysteme, Februar 1994, Beuth Verlag Berlin

Lepers, H.: SPS-Programmierung nach IEC 61131-3, Franzis Verlag Poing, 2005

Mann, H., Schiffelgen, H., Froriep, R.: Einführung in die Regelungstechnik, Hanser Verlag München, 8. Auflage 1997

Reuter, M., Zacher, S.: Regelungstechnik für Ingenieure, Vieweg Verlag, 11. Auflage 2004
Seitz, M.: Speicherprogrammierbare Steuerungen, Fachbuchverlag Leipzig, 2003

Wellenreuther, G., Zastrow, D.: Automatisieren mit SPS, Theorie und Praxis, Verlag Vieweg Wiesbaden, 3. Auflage 2005

Wellenreuther, G., Zastrow, D.: Automatisieren mit SPS, Übersichten und Übungsaufgaben, Verlag Vieweg Wiesbaden, 3. Auflage 2007