

## Steuerung

Die vorliegende Unterlage wurde u.a. auf der Basis von H. Heckmann: "Steuerungstechnik" erarbeitet.

| Inhalt:   | Seite: |
|---|--------|
| <b>1 Definition</b>                             | 1      |
| <b>2 Unterscheidungsmerkmale</b>                |        |
| 2.1 Informationsdarstellung                     | 2      |
| 2.2 Signalverarbeitung                          | 3      |
| 2.3 Programmverwirklichung                      | 4      |
| 2.4 Arbeitsweise (parallel / sequenziell)       | 5      |
| 2.5 CPU – Auslastung                            | 7      |
| <b>3 SW- Strukturen (Aufgaben - Gliederung)</b> | 8      |
| <b>4 Methoden</b>                               |        |
| 4.1 Betriebsarten (HAND / AUTO / ...)           | 10     |
| 4.2 Freigabe / Verriegelung / Schutz            | 11     |
| 4.3 Überwachungszeit                            | 11     |
| 4.4 Vorwahl                                     | 11     |
| 4.5 NOT - AUS                                   | 12     |
| 4.6 Spannungswiederkehr bei Antrieben           | 13     |
| 4.7 Wiedereinschaltung nach Spannungsabfall     | 14     |
| 4.8 Absteuerung von Stellantrieben              | 14     |
| <b>5 Signaldefinition</b>                       | 15     |
| <b>6 Verknüpfungssteuerung</b>                  | 16     |
| <b>7 Ablaufsteuerung</b>                        |        |
| 7.1 Prinzip                                     | 17     |
| 7.2 Struktur                                    | 17     |
| 7.3 Anwendungsbeispiel                          | 18     |
| 7.4 Aktions - Steuerung                         | 18     |
| 7.5 Ablaufketten - Struktur                     | 19     |
| <b>8 Realisierung</b>                           |        |
| 8.1 SPS / PLS                                   | 20     |
| 8.2 PC – basierte SPS                           | 21     |

## 1 Definition

(Wiederholung aus "Grundlagen")

Hier nochmals die Definition für "Steuern" und "Regeln" nach DIN IEC 60050-351:

Das **Steuern**, die **Steuerung**, ist der Vorgang in einem System, bei dem eine oder mehrere Größen als Eingangsgrößen andere Größen als Ausgangsgrößen aufgrund der dem System eigentümlichen Gesetzmäßigkeiten beeinflussen. Kennzeichen für das Steuern ist der offene Wirkungsweg oder ein geschlossener Wirkungsweg, bei dem die durch die Eingangsgrößen beeinflussten Ausgangsgrößen nicht fortlaufend und nicht wieder über dieselben Eingangsgrößen auf sich selbst wirken.

Das **Regeln**, die **Regelung**, ist ein Vorgang, bei dem fortlaufend eine Größe, die Regelgröße (die zu regelnde Größe), erfaßt, mit einer anderen Größe, der Führungsgröße, verglichen und im Sinne einer Angleichung an die Führungsgröße beeinflusst wird. Kennzeichen für das Regeln ist der geschlossene Wirkungsablauf, bei dem die Regelgröße im Wirkungsweg des Regelkreises fortlaufend sich selbst beeinflusst.

Bild 1.1 zeigt, dass der wesentliche Unterschied also im Wirkungsablauf bzw. in den verwendeten Signalen liegt.

Im englischen Sprachraum gibt es nur das Wort "Control" für „Leiten“, also Steuerung und Regelung, so dass hier gemäß DIN IEC 60050 durch „Open-Loop Control“ und „Closed-Loop Control“ unterschieden werden muss.

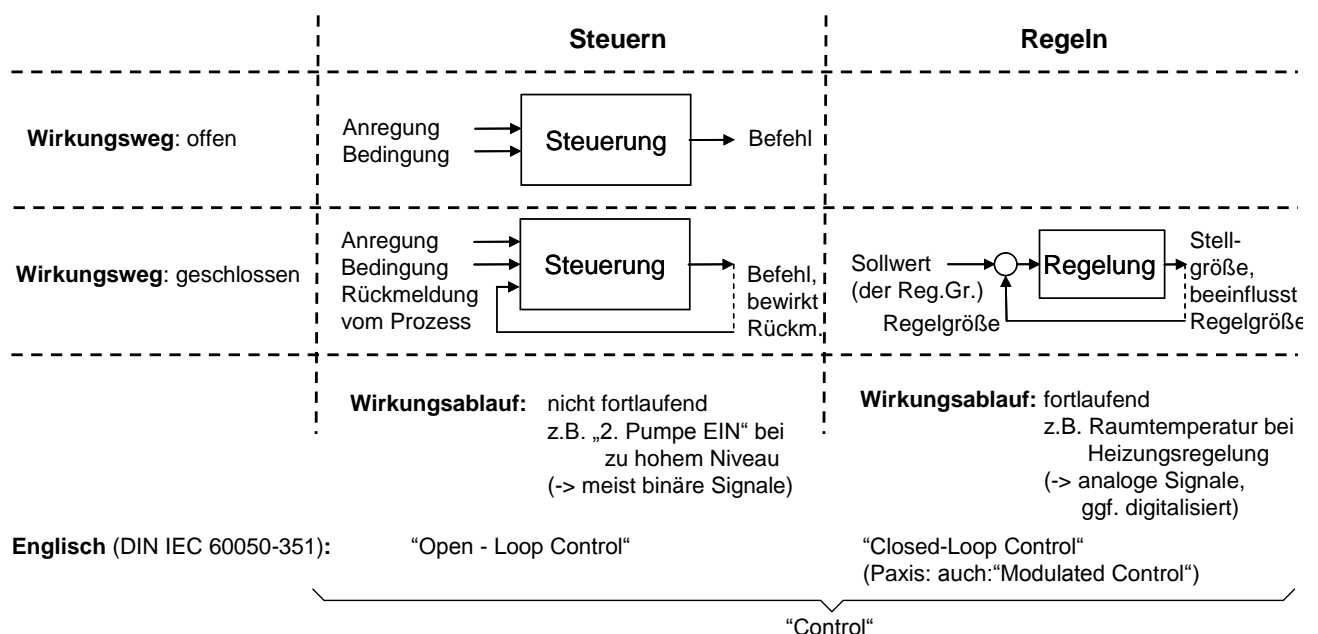


Bild 1.1: Definition für "Steuern" / "Regeln"

## 2 Unterscheidungsmerkmale

### 2.1 Informationsdarstellung

Steuerungen können mit analogen, digitalen und binären Signalen arbeiten und werden dementsprechend unterschieden.

Meist meint man jedoch mit "Steuerung" Einrichtungen, die mit binären Signalen arbeiten.

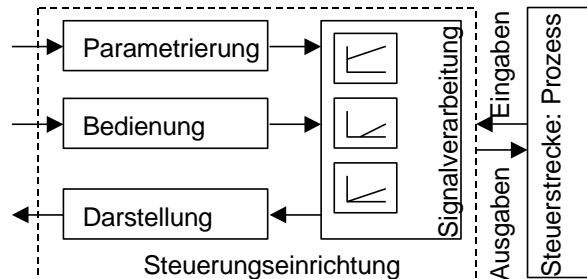
#### Analoge Steuerung

Eine innerhalb der Signalverarbeitung überwiegend mit analogen Signalen arbeitende Steuerung.

(Bei Speicher - programmierbaren Einrichtungen sind die analogen Prozess - Signale in digitaler Form dargestellt)

Anmerkung:

Die Signalverarbeitung erfolgt überwiegend mit stetig wirkenden Funktionsgliedern.

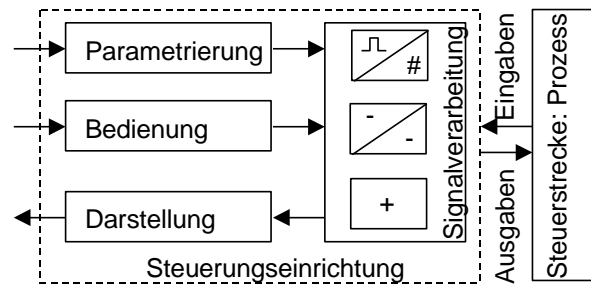


#### Digitale Steuerung

Eine innerhalb der Signalverarbeitung überwiegend mit digitalen Signalen arbeitende Steuerung.

Anmerkung:

Die Signalverarbeitung erfolgt überwiegend mit digitalen Funktionsbausteinen wie Zähler, Register, Speicher, Rechenwerke. Die zu verarbeitenden Informationen sind üblicherweise in einem Binärcode dargestellt.

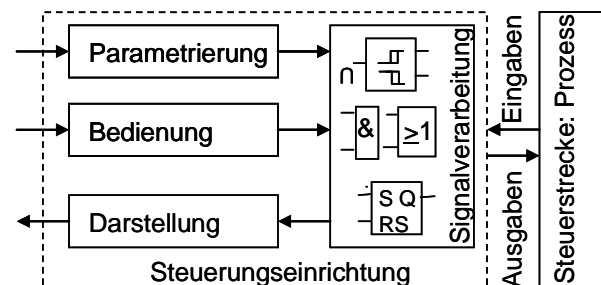


#### Binäre Steuerung

Eine Steuerung, die binäre Eingangssignale, die nicht Bestandteil einer binär codierten, digitalen Informationsdarstellung sind, zu binären Ausgangssignalen verarbeitet.

Anmerkung:

Die Signalverarbeitung erfolgt überwiegend mit Verknüpfungs-, Zeit- und Speichergliedern.



Die Begriffe "digital" und "binär" werden oft irreführend verwendet.

Die Information "T > HOCH" ist z.B. *binär*, da nur 2 Werte möglich sind: wahr und nicht wahr.

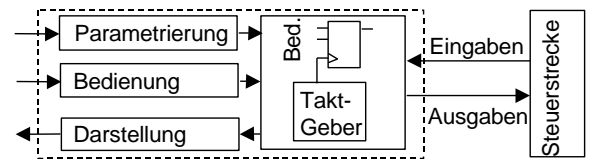
Wird eine analog gemessene Temperatur für eine Speicher - programmierte Einrichtung in eine Gruppe von Bit umgeformt, dann ist das eine *digitale* Darstellung, da die Bit - Zahl (und die Codierung) den kleinsten Unterschied (digit) zwischen zwei Werten festlegt.

## 2.2 Signalverarbeitung

Unterschieden werden nach DIN IEC 60050-351:

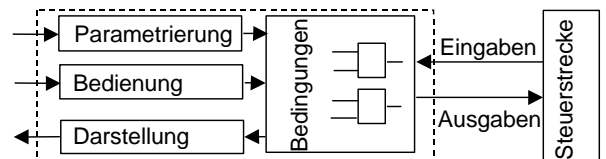
### Synchrone Steuerung

Eine Steuerung, bei der sich die Ausgangsgrößen nur synchron mit einem Taktsignal ändern.  
(Diese Verarbeitungsart wird selten verwendet. Die zyklische Verarbeitung im "Takt" des Prozessors in einer Speicher - programmierten Einrichtung zählt nicht hierzu.)



### Asynchrone Steuerung

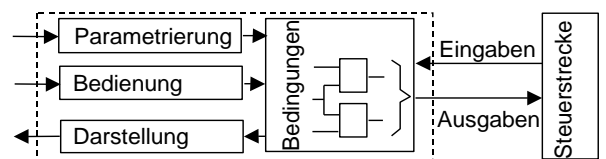
Eine Steuerung, bei der sich die Ausgangsgrößen nur durch Eingangsgrößenänderungen ändern.  
(So arbeitet die "normale" Steuerung.)



Bezogen auf das Steuerungs - Verfahren werden unterschieden:

### (Verknüpfungssteuerung: nicht in der Norm):

Eine Steuerung, bei der den Signalzuständen der Eingangssignale bestimmte Signalzustände der Ausgangssignale im Sinne von Verknüpfungsfunktionen zugeordnet werden



(Normalerweise kein zwangsläufiger Ablauf, kann jedoch durch Bedingungen, die auf früheren Befehlen beruhen, erreicht werden.)

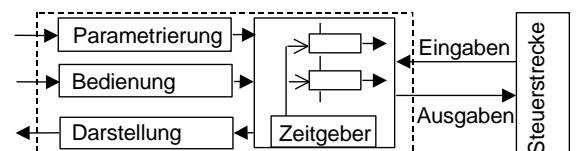
### Ablaufsteuerung

Eine Steuerung mit schrittweisem Ablauf, bei der der Übergang von einem Schritt auf den folgenden programmgemäß entsprechend den vorgegebenen Übergangsbedingungen erfolgt.  
Die Schritte einer Ablaufsteuerung entsprechen den aufeinanderfolgenden diskreten Betriebsbedingungen des technischen Prozesses.

Ausführung je nach Übergangsbedingungen entweder als

#### Zeit- abhängige Ablaufsteuerung

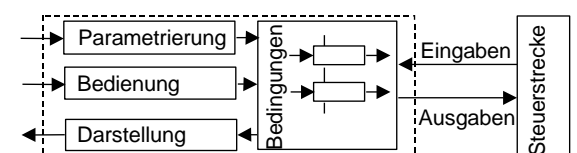
Ablaufsteuerung, bei der die Weiterschaltbedingungen nur von der Zeit abhängig sind, oder als



#### Prozess - abhängige Ablaufsteuerung

Ablaufsteuerung, bei der die Weiterschaltbedingungen nur von Signalen der gesteuerten Anlage abhängig sind.

(Dieses Verfahren wird zur Steuerung von Prozess - Abläufen meist benutzt.)



### 2.3 Programmverwirklichung

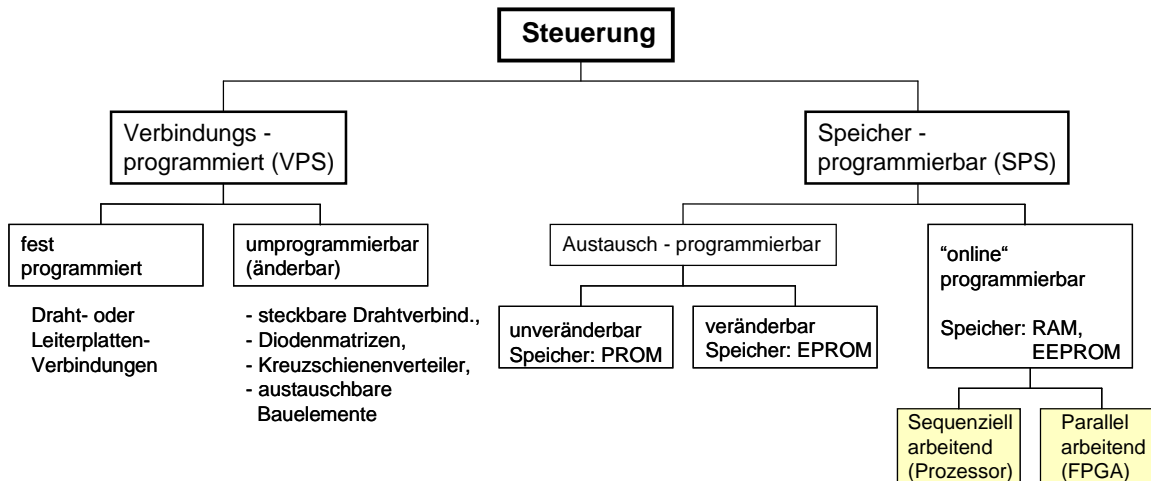


Bild 2.3.1 Arten der Programmverwirklichung

#### Programm

Unter dem "Programm" einer Steuerung soll die Gesamtheit aller Anweisungen und Vereinbarungen für die Signalverarbeitung verstanden werden, durch die eine zu steuernde Anlage (Prozess) Aufgaben - gemäß beeinflusst wird. Das Programm einer binären oder digitalen Steuerung kann prinzipiell entweder durch die Art der Funktionsglieder und deren Verbindungen (VPS: Verbindungs - programmierte Steuerung) oder als eine in Speichern hinterlegte Reihe von Anweisungen (SPS: Speicher - programmierte Steuerung) festgelegt werden.

#### Steuerung, Verbindungs - programmiert

VPS (Verbindungs - programmierte Steuerungen) können entweder

- *fest programmiert*, d.h. unveränderbar, z.B. durch feste Draht- oder Leiterplattenverbindungen, oder
- *umprogrammierbar*, d.h. veränderbar, z.B. durch steckbare Drahtverbindungen, Diodenmatrizen, Lochkarten, änderbare Kreuzschienenverteiler oder austauschbare Bauelemente verwirklicht werden.

Eine VPS arbeitet normalerweise parallel, d.h. es müssen keine Prozessor- oder Bus - Zykluszeiten beachtet werden.

#### Steuerung, Speicher - programmierbar

Bei SPS (Speicher - programmierbare Steuerungen) lassen sich "online" programmierbare Steuerungen von Austausch - programmierbaren Steuerungen unterscheiden.

Bei *online programmierbaren* Steuerungen ist der Programmspeicher ein Schreib - Lese - Speicher (RAM), dessen gesamter Inhalt ohne mechanischen Eingriff in die Steuerungseinrichtung (ziehen einer Karte, wechseln von Speicher - Chips) geändert werden kann. Dabei kann meist auch ein kleiner Teil des Programms geändert werden, weswegen diese auch als "frei" programmierbar bezeichnet werden.

Für den Wiederanlauf nach Spannungs - Abschaltung wird das Programm zusätzlich in einem EEPROM Spannungsausfall - sicher abgelegt.

*Austauschprogrammierbare* Steuerungen sind Steuerungen mit Nur - Lese - Speichern (ROM) als Programmspeicher, deren Inhalt nach erfolgtem Programmieren nur durch mechanischen Eingriff in die Steuerungseinrichtung verändert werden kann. Hierbei lassen sich Steuerungen mit Nur - Lese - Speichern unterscheiden, die nach der Herstellung programmiert und mehrmals verändert werden können (EPROM), sowie solche, die nur einmalig bei oder nach der Herstellung programmiert werden können und dann unveränderbar sind (ROM, PROM). In beiden Fällen wird durch Austausch meist das gesamte Programm gewechselt.

Bisher arbeiteten Speicher - programmierte Steuerungen stets sequenziell (Prozessor).

Eine neue Entwicklung erlaubt auch parallele Verarbeitung (wie bei der VPS) durch Speicherung der "Verbindungen" in einem "FPGA" (Field Programmable Gate Array), die Tools dazu entsprechen denen der sequenziell arbeitenden Einrichtungen. Dadurch entfallen Prozessor - Zykluszeiten sowie Bus - Laufzeiten durch parallele Verbindungen zu den Ein / Ausgängen.

## 2.4 Arbeitsweise

Die einzelnen Funktionen einer Steuerung können **parallel** oder **sequenziell** abgearbeitet werden.

Verdrahtete Steuerungseinrichtungen sowie FPGA - Lösungen arbeiten parallel, Rechner - basierte Lösungen mit Prozessor(en) und Bussen arbeiten sequenziell.

Die Arbeitsweise hat Einfluss auf den Logik - Entwurf, denn:

- bei paralleler Arbeitsweise findet ein "Wettlauf" zwischen parallelen Logik - Strängen statt, der manchmal durch (Laufzeit-) Zeitglieder im Voraus entschieden werden muss um nicht ungewollte Effekte zu erhalten, und
- bei sequenzieller Arbeitsweise ist zu beachten:
  - ein Funktionsergebniswert kann erst wieder im nächsten Zyklus (nach einer Zykluszeit) verändert werden. Er steht daher ab seiner Berechnung allen anderen Funktionen 1 Zyklus lang zur Verfügung.
  - Zu einem Zeitpunkt wird nur eine Funktion bearbeitet. Bei richtiger Reihenfolge der Funktionen sind daher keine Laufzeit - Maßnahmen (Logik, Zeitglieder) nötig.
  - Falsche Reihenfolge der Funktionen einer Logik kann eine Verzögerung um eine Zykluszeit bewirken (Bild 2.4.4)
  - Zeiten unterhalb der Zykluszeit sind nicht realisierbar.

Für eine bestimmte Aufgabe (Bild 2.4.1) sind unten beide Arbeitsweisen im Signal - Zeit - Diagramm dargestellt. Es geht um das Umschalten von Speicher S2 auf Speicher S1 per taste T1.

Bei der parallelen Arbeitsweise (Bild 2.4.2) ist zu beachten, dass jedes Gatter eine Laufzeit hat. Das sind zwar nur  $\mu s$ , reicht aber zur Beeinflussung anderer Eingänge. Hier sind für eine kurze Zeit beide Speicher EIN, was woanders (durch X dargestellt) stören könnte. Zur Vermeidung müsste die Logik geändert werden.

(Die Gatter - Laufzeiten sind im nebenstehenden Signal-Zeit-Diagramm übertrieben lang dargestellt um Abhängigkeiten zu verdeutlichen).

Bei sequenzieller Arbeitsweise (Bild 2.4.3) sind zwar beide Speicher länger gleichzeitig EIN, da aber zu dieser Zeit keine andere Funktion bearbeitet wird, stört das nicht.

Ebenfalls sehr wichtig ist die richtige Abarbeitungs - Reihenfolge der Funktionen / Funktionsbausteine im Programm einer speicherprogrammierten Einrichtung. Die Reihenfolge ergibt sich aus der Reihenfolge der hinzugefügten Funktionen / Funktionsbausteine. „Richtig“ ist sie, wenn sie dem Signalfluss durch eine Logik entspricht. Eine „falsche“ Reihenfolge führt zu einer Verzögerung der Berücksichtigung eines neuen Wertes an einem Eingang. Insbesondere bei Regelungen stört das sehr.

**Beispiel** (Annahme: T1 wird gedrückt)

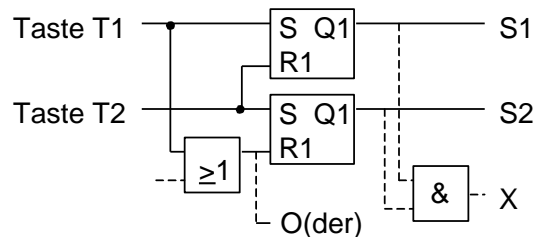


Bild 2.4.1 Aufgabenstellung (Beispiel)

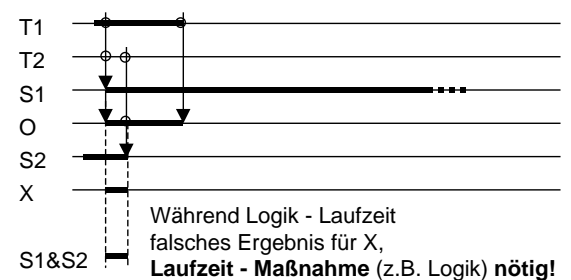


Bild 2.4.2 Signal - Zeit - Diagramm für parallele Arbeitsweise

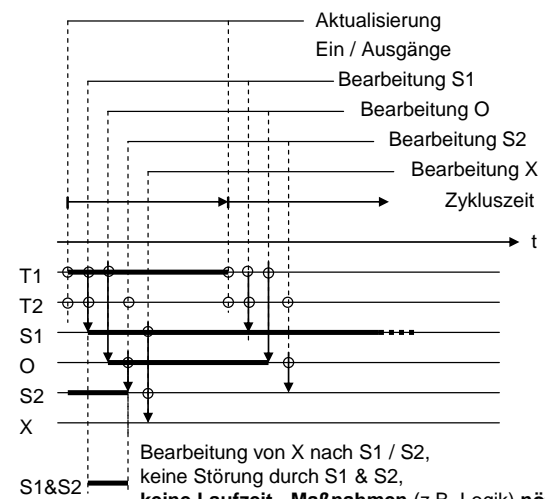


Bild 2.4.3 Signal - Zeit - Diagramm für sequenzielle Arbeitsweise

Die Reihenfolge ist nur in der Darstellung des Programms in einer höheren Programmiersprache oder als Anweisungsliste direkt zu erkennen. In der Funktionsplan - Darstellung (siehe Bild 2.4.4) ergibt sich bei „falscher“ Reihenfolge meist eine nicht „schöne“ Leitungsführung, da der Funktionsplan tatsächlich ein interpretiertes Textfile ist und die Linienführung nicht wie bei einem echten Grafikprogramm beeinflusst werden kann. Zusätzlich wird die Reihenfolge oft im Plan angegeben.

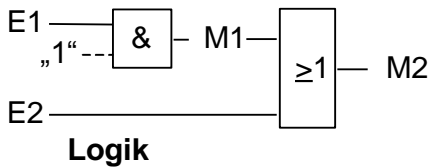


Bild 2.4.4: Beispiel - Aufgabe

Angenommen, die oben im Bild 2.4.4 dargestellte Aufgabe sei zu abuarbeiten.

Bild 2.4.5 zeigt in einem Signal – Zeit – Diagramm was geschieht, wenn bei der ersten Aktualisierung E1 auf „1“ wechselt und die beiden Gatter dem Signalfloss entsprechend im Programm aufgeführt sind (erst M1, dann M2). Das Ergebnis an M2 liegt gleich nach Bearbeitung des ODER – Gatters vor.

In Bild 2.4.6 ist angenommen, dass zuerst M2 und dann M1 bearbeitet wird. Dadurch wird an M2 zunächst der alte Wert „0“ von M1 verwendet und das richtige Ergebnis erscheint erst einen Zyklus später.

Eine weitere Besonderheit der sequenziellen Abarbeitung ist die Behandlung von Zeiten.

Angenommen, es sei ein Einschaltverzögerer abuarbeiten (siehe Bild 2.4.7).

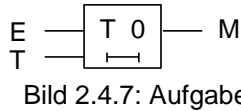


Bild 2.4.8 zeigt als Signal – Zeit – Diagramm wie die eingestellte Zeit T bei einem 0->1 – Wechsel an E gestartet wird. Auch wenn die eingestellte Zeit kürzer (links) oder länger (rechts) als die Zykluszeit ist kann ihr Ende immer nur beim nächsten Durchgang festgestellt werden. Die wirksame Zeit kann also nur die einfache oder mehrfache Zykluszeit lang sein.

In den Signal- Zeit- Diagrammen ist außer der Zykluszeit auch die „Aktualisierung“ dargestellt. Die meisten speicherprogrammierten Automatisierungseinrichtungen lesen vor der Bearbeitung eines Teilprogramms die aktuellen Werte der darin benötigten externen Signale und speichern sie in einem besonderen, diesem Teilprogramm zugeordneten Speicherbereich („Anlagenabbild“). Mit diesen Werten wird dann das Teilprogramm abgearbeitet. Danach erst werden die Ergebnisse weitergegeben. So erhalten alle Funktionen eines Teilprogramms die gleichen Werte.

Das ist gut für die Logik, kann aber eine verzögerte Reaktion bedeuten. In Bild 2.4.9 ändert sich ein Wert kurz nach dem Lesen, kann also erst im nächsten Zyklus gelesen werden. Das Ergebnis steht dann erst nach knapp 2 Zyklen zur Verfügung.

Bild 2.4.10 zeigt eine kontinuierliche Aktualisierung ohne Anlagenabbild. Hier kann eine Wertänderung für ihre Verarbeitung gerade zu spät kommen, wird aber im nächsten Zyklus berücksichtigt und sofort weitergegeben. Die Reaktionszeit ist hier also kürzer.

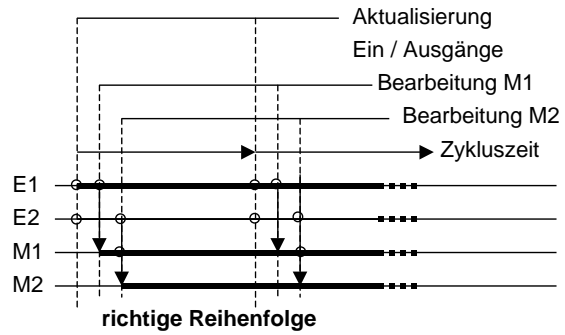


Bild 2.4.5: „Richtige“ Bearbeitungsreihenfolge

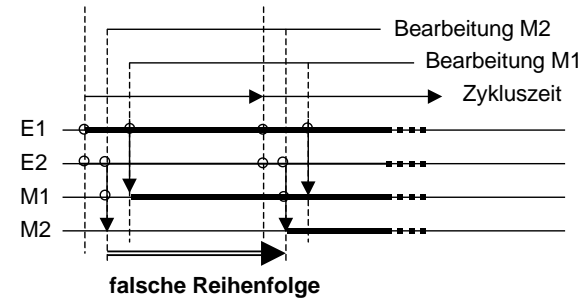


Bild 2.4.6: „Falsche“ Bearbeitungsreihenfolge

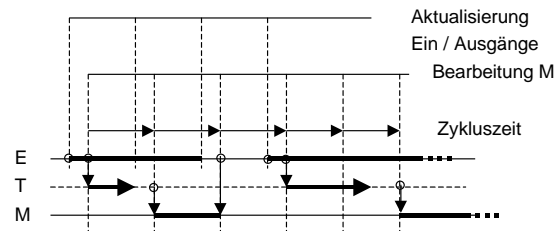


Bild 2.4.8: Behandlung von Zeiten

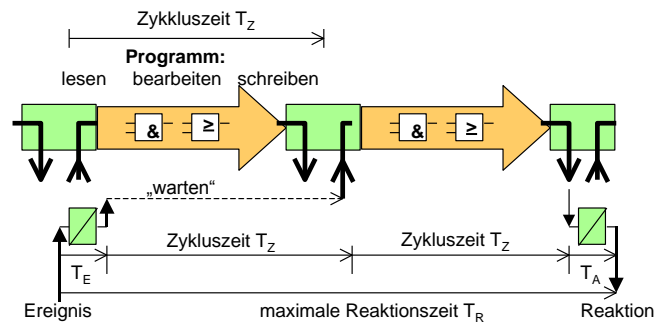


Bild 2.4.9: Aktualisierung mit Anlagenabbild

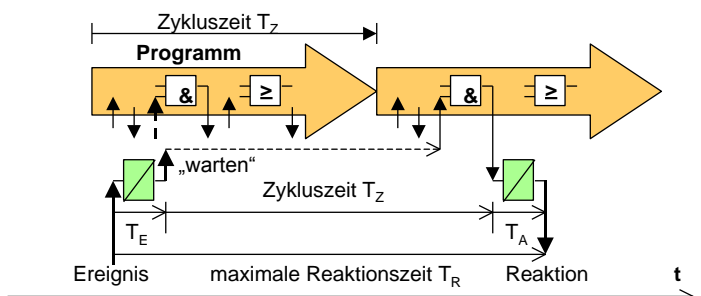


Bild 2.4.10: Kontinuierliche Aktualisierung

## 2.5 CPU - Auslastung

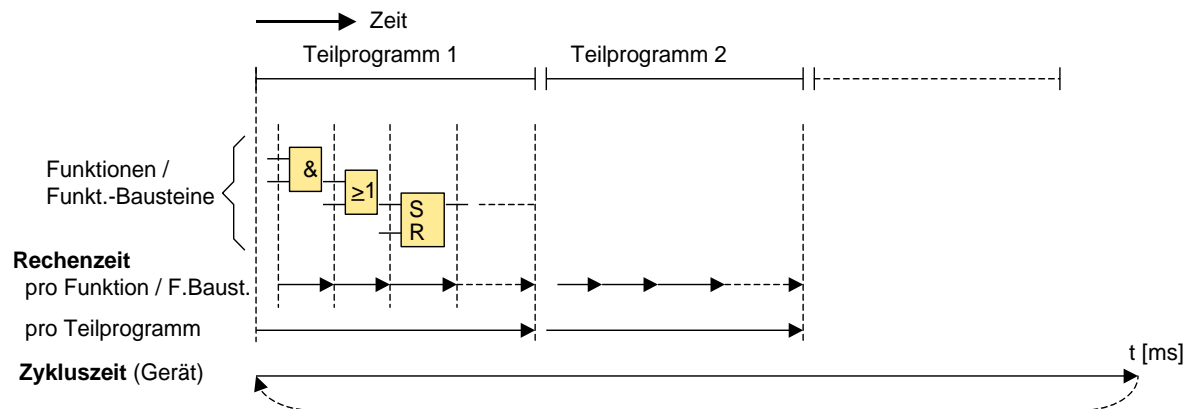


Bild 2.5.1: Geräte - Zykluszeit

Jede Funktion und jeder Funktionsbaustein benötigt eine bestimmte Rechenzeit, abhängig von der Komplexität und der Verarbeitungsgeschwindigkeit der CPU. Diese Rechenzeiten addieren sich zu den Rechenzeiten der Teilprogramme und diese zu der des gesamten Gerätes (der CPU). Erst nach dieser Zeit kann die Abarbeitung wieder neu beginnen. Diese Zeit wird daher die „Zykluszeit“ genannt.

Je mehr Funktionen / Funktionsbausteine von einer CPU bearbeitet werden müssen, desto länger ist die Zykluszeit. Sie kann bis zu mehrere Sekunden lang sein. Welche Länge erlaubt ist ergibt sich aus der vom Prozess benötigten Reaktionszeit, die - wie in Bild 2.4.9 gezeigt - bis zu 2 Zykluszeiten betragen kann. Ist das zu langsam, so muss die Anzahl der Funktionen / Funktionsbausteine reduziert werden.

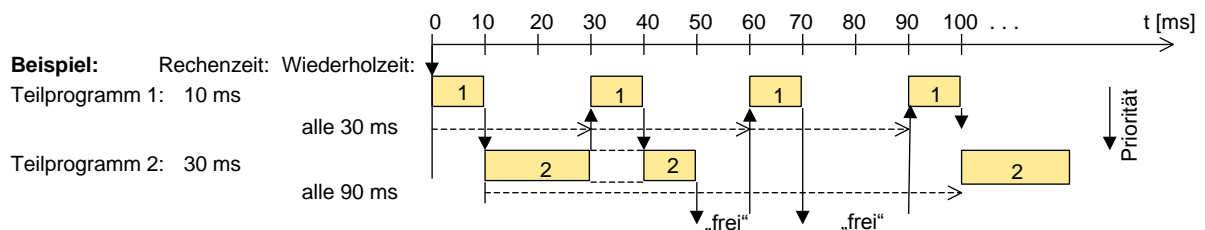


Bild 2.5.2: Teilprogramme mit verschiedenen Wiederholzeiten

Eine andere Maßnahme besteht darin, die Teilprogramme mit verschiedenen Wiederholzeiten auszustatten. Nun können Aufgaben, die eine kurze Reaktionszeit benötigen, in Teilprogrammen mit kurzen Wiederholzeiten realisiert werden (z.B. schnelle Regelungen wie Antriebsregelungen) und zulässige lange Reaktionszeiten in Teilprogrammen mit langen Wiederholzeiten (z.B. Antriebssteuerungen).

In Bild 2.5.2 hat die CPU zwischen den Wiederholungen der Teilprogramme noch „freie“ Phasen. Würden aber andere Zeiten oder mehr Teilprogramme festgelegt, so könnte es nicht mehr möglich sein, alle Teilprogramme mit der gewünschten Wiederholzeit abzuarbeiten. Je nach Fabrikat würden die Zeiten länger oder die Programmabarbeitung würde „abstürzen“.

In Bild 2.5.2 ist ein Beispiel mit zwei Teilprogrammen dargestellt, die unterschiedliche Rechen- und Wiederholzeiten benötigen. Die jeweils zu verwendende Wiederholzeit wird an einem Struktur - Funktionsbaustein pro Teilprogramm festgelegt.

Um das zu vermeiden muss bei der Planung die CPU - Belastung ausgerechnet werden - von Hand oder durch eine Prozedur des Engineeringtools:

$$\text{Last} = \frac{\text{Rechenzeit 1}}{\text{Wiederholzeit 1}} + \frac{\text{Rechenzeit 2}}{\text{Wiederholzeit 2}} \dots$$

Im Bild ist zu sehen, dass die Wiederholzeit für Teilprogramm 1 nur eingehalten werden kann, wenn die Abarbeitung des Teilprogramms 2 unterbrochen wird. Dazu sind den Teilprogrammen Prioritäten zuzuordnen. Meist werden sie aus den Wiederholzeiten abgeleitet, manchmal auch eigens festgelegt.

$$\text{Last} = \frac{10}{30} + \frac{30}{90} = 0,33 + 0,33 = 0,66$$

+ ca. 8% Prozessor-Grundlast = 0,74 (oder 74%)

### 3. Software- Strukturen

#### 3.1 Unstrukturiert

Bei kleineren Steuerungsaufgaben werden meist alle Teilaufgaben in *einer* Steuerungseinrichtung durch *ein* Programm realisiert.

Bild 3.1 zeigt eine solche "**unstrukturierte**" Steuerung für 2 Pumpen, die jeweils einen Absperrschieber (zum Anfahren ohne Last) und eine Durchflussmessung (als Schutz gegen Fehler in Pumpe oder Prozess) haben.

Die Bedienung könnte insgesamt oder für jedes angeschlossene Aggregat (Pumpe bzw. Schieber) getrennt erfolgen.

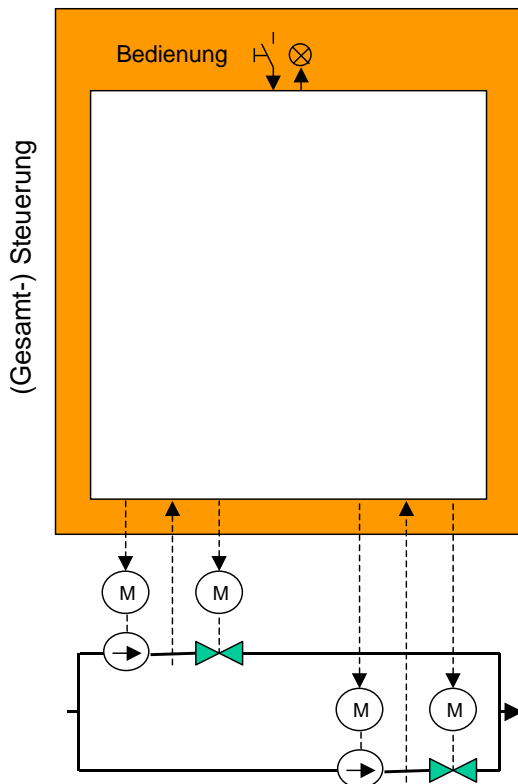


Bild 3.1: unstrukturierte Steuerung

#### 3.2 Hierarchische Struktur

Je umfangreicher ein zu steuernder Prozess ist, desto mehr empfiehlt sich eine klare Gliederung der Teilaufgaben. Dabei wird jede Teilaufgabe als selbständiger Bereich mit definiertem Informationsaustausch zu anderen Bereichen betrachtet.

Das ergibt folgende Vorteile:

- einfachere Planung, Inbetriebnahme und Wartung, da sich ein solcher kleiner Bereich leichter definieren / verstehen lässt,
- Möglichkeit der Verwendung standardisierter Teil - Lösungen wie z.B. Antriebssteuerungen,
- leichtere Änderbarkeit, wenn die Bereiche durch einzeln änderbare Teilprogramme realisiert werden,
- höhere Verfügbarkeit, wenn zumindest redundante (parallel wirkende, alternativ einsetzbare) Teile durch getrennte Steuerungseinrichtungen realisiert werden.

Bild 3.2 zeigt eine **gegliederte** Lösung der gleichen Aufgabe wie in Bild 3.1 durch eine **hierarchische Struktur**, wie sie in der Verfahrenstechnik meist angewandt wird. Dabei werden Ebenen gebildet:

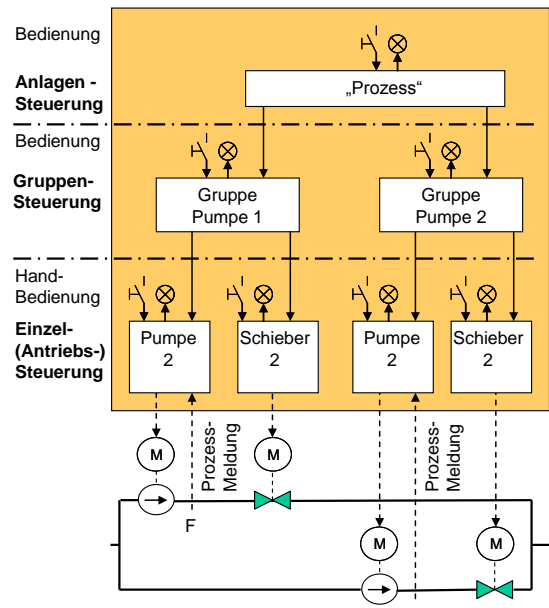


Bild 3.2: Hierarchisch strukturierte Steuerung

- In der **Antriebs-** (Einzelsteuerungs-) **Ebene** erhält jedes Aggregat eine Antriebs- (Einzel-) Steuerung, in der alles enthalten ist, was zum Betrieb und Schutz dieses Aggregates nötig ist. Dies kann sein:
  - Entgegennahme von Befehlen (EIN/AUS, AUF/ZU) höherer Ebenen und ggf. von Hand,
  - Berücksichtigung von Bedingungen zum Ein / Ausschalten bzw. Öffnen / Schließen,
  - Selbsttätiges Ein / Ausschalten bzw. Öffnen / Schließen zum Schutz des Aggregats,
  - Ausgeben von Befehlen an den Leistungsteil und Überwachen der Befehlsausführung und ggf. der Funktionstüchtigkeit durch Rückmeldungen, sowie
  - Meldung des Zustandes und ggf. Störung an die höhere Ebene und ggf. Bedienung.
- In der **Gruppenebene** werden einzelne Aggregate zusammengefasst, die eine gemeinsame Aufgabe erfüllen. Die Gruppensteuerung ist nur für das automatische, ordnungsgemäße Anfahren / in Betrieb halten / Abfahren dieser Aggregate - Gruppe zuständig auf Befehl der darüberliegenden Ebene oder Bedienung. Sie setzt die Aufgaben der unteren Ebene voraus. Dadurch ist sie relativ einfach aufgebaut. Die Gruppenebene kann weiter unterteilt sein.
- In der **Prozess - Ebene** ("Leit - Ebene") werden die Gruppen eines Prozesses zusammengefasst und gesteuert, wobei wieder die Aufgaben der unteren Ebenen vorausgesetzt werden und auch diese Ebene übersichtlich machen.



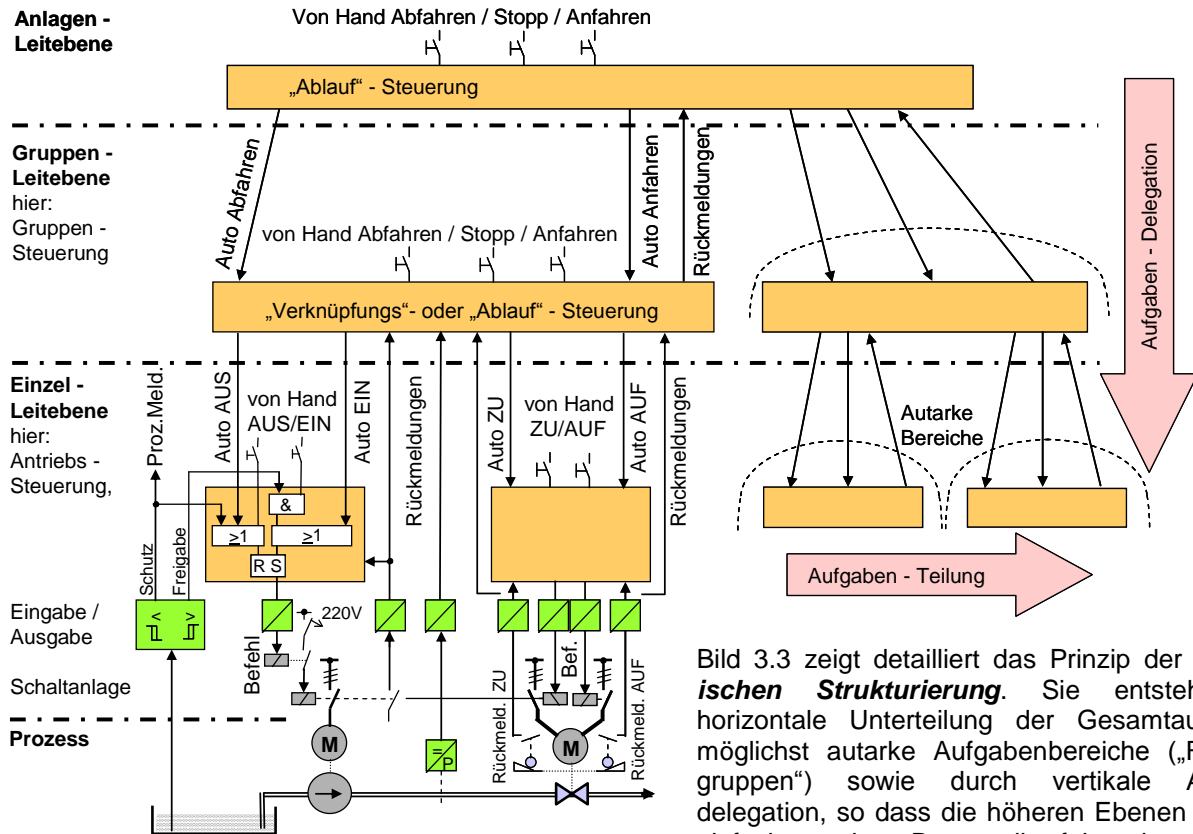


Bild 3.3: Hierarchische Struktur in der Steuerung

In der Verfahrenstechnik ist die hierarchische Struktur sinnvoll, hier geht es meist nur um einen zusammenhängender Prozess mit einem „Ziel“.

Bild 3.3 zeigt detailliert das Prinzip der **hierarchischen Strukturierung**. Sie entsteht durch horizontale Unterteilung der Gesamtaufgabe in möglichst autarke Aufgabenbereiche („Funktionsgruppen“) sowie durch vertikale Aufgaben-delegation, so dass die höheren Ebenen möglichst einfach werden. Das ergibt folgende vorteilhafte Effekte:

- bessere Überschaubarkeit der Teilaufgaben,
- höhere Verfügbarkeit durch die Möglichkeit verteilter Hardware- Realisierung.

### 3.3 Dezentrale Struktur

In der Fertigungsautomatisierung geht es meist um viele kleine Teil- Prozesse, die koordiniert werden müssen. Diese lassen sich durch **dezentrale Struktur** besser beschreiben (Bild 3.4). In einer hierarchischen Struktur würden die oberen Ebenen zu komplex. Die heutigen HW- Möglichkeiten dezentraler, Prozess-naher Verarbeitungseinheiten kommen dem entgegen.

Bild 3.4 zeigt dazu ein vereinfachtes Beispiel aus der Fertigungsautomatisierung mit drei Aufgabenbereichen:

- das Haupt- Band zum Werkstücktransport zu verschiedenen Bearbeitungsschritten (Zellen),
- eine Verzweigung (von mehreren) zu einer Fertigungszelle,
- eine Fertigungszelle mit Greifer und Bearbeitungsmaschine(n).

Jeder Aufgabenbereich (Zelle) hat eine eigene Steuerung bzw. Regelung, die ihre Aufgaben weitgehend selbstständig erfüllt. Sie koordinieren sich durch Kommunikation untereinander (statt Befehlsempfang von „oben“ und Rückmeldung nach „oben“ bei der hierarchischen Struktur).

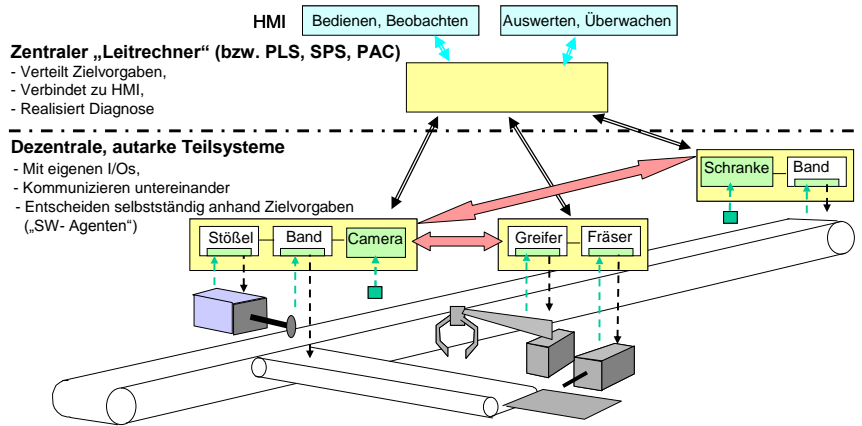


Bild 3.4: Dezentrale Struktur mit „Software- Agenten“

Ein übergeordneter „Leitrechner“ (oder eine SPS oder ein PAC) gibt die Sollwerte vor (Varianten, Stückzahlen) und stellt die Verbindung zur Mensch- Maschine- Kommunikation (englisch: HMI) her.

Die konsequente Weiterführung dieses Ansatzes ist die Entwicklung des Konzepts der **„Software-Agenten“**. Dies sind abgegrenzte SW- Einheiten, die weitgehend autonom agieren um ihnen vorgegebene Ziele zu erreichen. Dabei interagieren sie kontinuierlich mit der Umgebung (Prozess) und anderen Agenten (siehe Kapitel „Engineering“).

## 4. Methoden

Nachfolgend werden in verschiedenen Anwendungen übliche Methoden für bestimmte Teilaufgaben beschrieben.

Definitionen genormter Begriffe sind in Hochkommata gesetzt ' ' und die entsprechende Norm ist im Anschluss daran in [ ] angegeben.

### 4.1 Betriebsarten

#### Betriebsart

„Kennzeichnung von Art und Umfang, wie der Mensch als Bediener in eine Leiteinrichtung eingreift“ (DIN IEC 60050-351)

Die jeweilige Betriebsart kann durch den Menschen (Bediener), durch die Leiteinrichtung oder durch den Prozeß veranlaßt werden. Je nach Betriebsart sind einzelne Befehle, Rückführungen und Rückmeldungen wirksam oder unwirksam.

Der Bedienungseingriff kann sich dabei auf die Einzel- oder Antriebsleitebene, die Gruppenleitebene oder die Anlagenleitebene erstrecken.

#### Betriebsart "Automatik"

„Betriebsart, in der alle Funktionen der Leiteinrichtung ohne Eingriff des Menschen ausgeführt werden“ (DIN IEC 60050-351)

#### Betriebsart "Teilautomatik"

„Betriebsart, in der nur ein Teil der Funktionen der Leiteinrichtung ohne Eingriff des Menschen ausgeführt wird. (DIN IEC 60050-351)

#### Betriebsart Hand

„Betriebsart, in der alle Funktionen der Leiteinrichtung durch Menschen ausgeführt werden. (DIN IEC 60050-351)

Bild 4.1.1 zeigt eine einfache "Automatik" zum AUS- und EIN - Schalten einer Pumpe mit HAND / AUTO - Umschaltung in der Antriebssteuerung. Dadurch ist für die "Automatik" keine eigene Funktion (Logik, ...) notwendig.

Bei aufwändigeren Aufgaben wie in Bild 4.1.2 ist eine übergeordnete Instanz als "Gruppensteuerung" sinnvoll, die dann die HAND / AUTO - Umschaltung enthält. Bei der dargestellten Lösung kann der Bediener auch bei Betriebsart „Automatik“ der Gruppensteuerung von Hand eingreifen. Das ist praktisch für den Bediener, genau genommen aber nicht nach Norm. Daher hat man manchmal eine zusätzliche HAND – AUTO- Umschaltung in der Antriebssteuerung, wie in Bild 4.1.1.

Jede **Automatik** muss **abschaltbar** sein, damit der Bediener in Ausnahmefällen per HAND - Bedienung den Prozeß mit Vorrang im Griff behält.

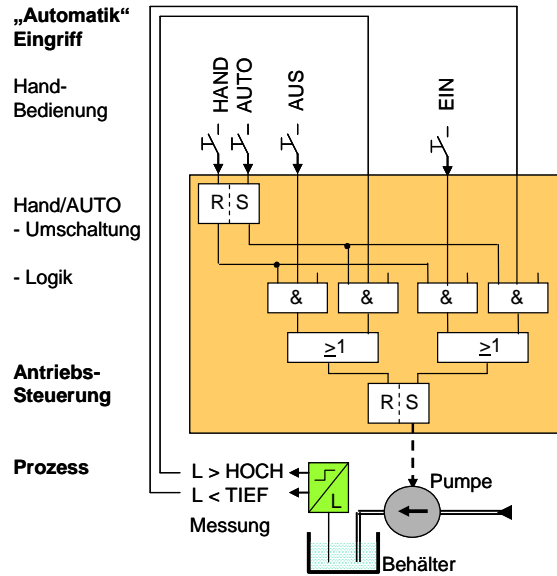


Bild 4.1.1: "Automatik" in der Antriebssteuerung

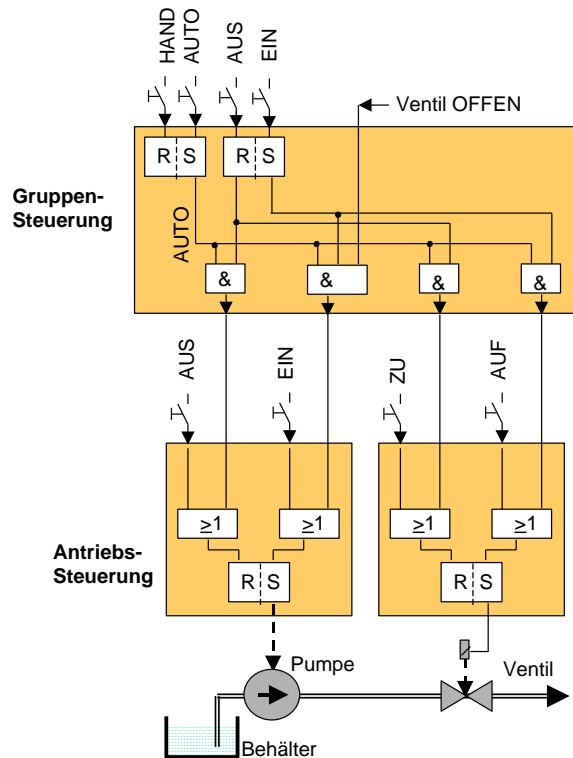


Bild 4.1.2: Gruppensteuerung als Automatik

**Betriebsart Einrichten** (Einstellen einer Maschine)  
Die Betriebsart Einrichten ist eine Betriebsart, in der die Stellgeräte einzeln durch Bedienungseingriff unter Umgehung vorhandener Verriegelungen gesteuert werden. (nicht mehr genormt)

Sicherungsverriegelungen ( / Schutz - Befehle) müssen dabei im Allgemeinen aber wirksam bleiben (siehe 4.2).

Die folgenden Betriebsarten kommen in Ablaufsteuerungen (Schritt - Steuerungen) vor:

**Betriebsart Schrittsetzen**

„Die Betriebsart Schrittsetzen ist eine Betriebsart, in der jeder Schritt in der Ablaufkette einer Ablaufsteuerung direkt gesetzt werden kann“ (DIN IEC 60050-351).

Wahlweise kann Schrittsetzen ohne Übergangsbedingungen oder Schrittsetzen mit Übergangsbedingungen möglich sein.

Ebenfalls kann wahlweise Schrittsetzen ohne Befehlsausgabe oder Schrittsetzen mit Befehlsausgabe möglich sein, wobei Befehlsausgabe hier Eingriff in die nächst tiefere Leitebene bedeutet.

**Betriebsart Tippen (nicht mehr genormt):**

Die Betriebsart Tippen ist eine Betriebsart einer Ablaufsteuerung, in der das Weiterschalten der Ablaufkette auf den nächst folgenden Schritt auch ohne erfüllte Übergabebedingungen durch den Bediener bewirkt werden kann.

**4.2 Freigabe / Verriegelung / Schutz**

(siehe Bild 4.2.1)

**Freigabesignal (Freigabe)**

„ist ein Signal, das die Übertragung eines Signals, das Wirksamwerden eines Elements oder die Ausführung eines Befehls zulässt“. (DIN IEC 60050-351)

**Verriegelung, Verriegelungssignal**

„ist ein Signal, das die Übertragung eines Signals, das Wirksamwerden eines Elements oder die Ausführung eines Befehls blockiert“. (DIN IEC 60050-351)

Zu beachten ist der Gegensatz zwischen Freigabe und Verriegelung.

Freigabe / Verriegelungssignale sind überwiegend Prozess - abhängige Signale.

**Schutz - Befehl (z.B. in der Kraftwerksleittechnik, nicht genormt):**

Prozess - abhängiges Signal, das ein bevorrechtigtes EIN - oder AUS - Schalten zum Schutz eines Aggregats bewirkt (hier: "Trockenlauf-Schutz")

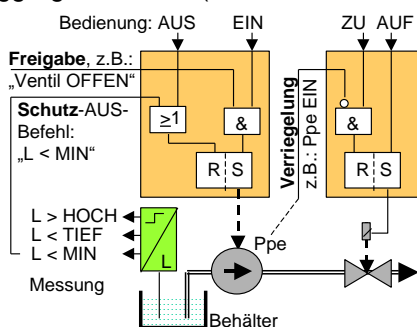


Bild 4.2.1: Freigabe / Verriegelung / Schutz

Ob Freigabe oder Verriegelung verwendet wird liegt an der jeweiligen Sicherheitsphilosophie und dem Vorhandensein der entsprechenden Signale.

Ein Schutzbefehl hat höchste Priorität und verriegelt gegensinnige "normale" Befehle, wenn er aktiv ist (z.B. durch dominierend rücksetzendes FlipFlop).

Schutzbefehle können in der (elektronischen) Steuerung oder zur höheren Sicherheit im Leistungsteil (Schaltanlage) realisiert sein.

**4.3 Überwachungszeit**

„Dauer des Zeitintervalls, nach dem ein Befehl unwirksam werden muss“ (DIN IEC 60050)

Die Ausführung eines Befehls wird oft durch eine Überwachungszeit kontrolliert, die etwas länger eingestellt ist, als zur Ausführung normalerweise nötig. Bild 4.3.1 zeigt dies für die Befehle ZU und AUF eines Stellantriebs in dessen Antriebssteuerung.

Wird ein Befehl länger ausgegeben als die Überwachungszeit, UND ist die jeweilige Endlage noch nicht erreicht (Signalpegel 0) so erfolgt ein Alarm und ggf. Befehlsabsteuerung.

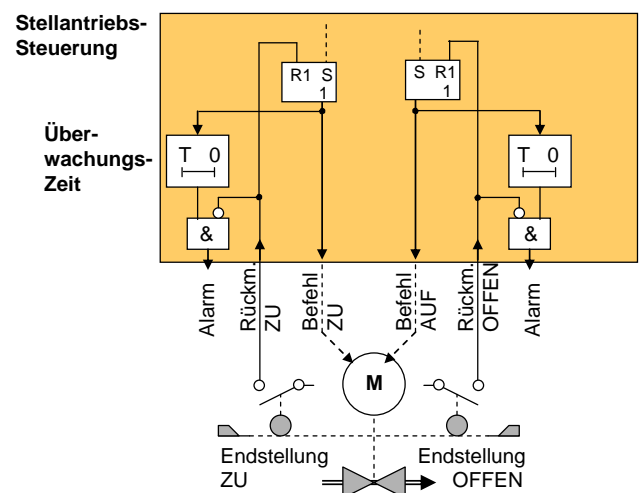


Bild 4.3.1: Überwachungszeit

In Ablaufsteuerungen wird meist in jedem Schritt die Befehlsausführung durch eine Überwachungszeit überwacht.

**4.4 Vorwahl (nicht genormt)**

Insbesondere in der Verfahrenstechnik angewandt: - Auswahl des später autom. einzuschaltenden Aggregats ("Betriebsaggregat") aus zwei od. mehr redundanten Agg.,

- Automatische Umschaltung auf Reserveaggregat bei Störung des Betriebsaggregates.
- Manuelle Tauschmöglichkeit des Betriebsaggregates.

Manchmal sind z.B. 2 von 3 Aggregaten "Betriebsaggregat", wenn 3 Agg. zu 50% vorhanden sind.

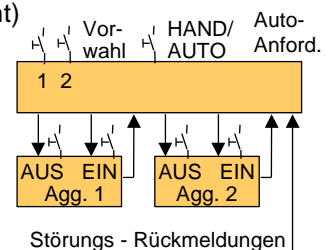


Bild 4.4: Vorwahl (Prinzip)

### 4.5 NOT - AUS

Eine NOT - AUS - Einrichtung soll im Gefahrenfall eine Maschine, einen Anlagenteil oder eine ganze Anlage so stillsetzen, dass Schäden für Personen oder Maschinen vermieden werden. Eine NOT - AUS - Anregung erfolgt meist durch einen vor - Ort angebrachten Schalter oder eine "Reissleine" (Förderbänder).

Es sind nur solche Einrichtungen im Gefahrenfall stillzusetzen, die entweder bereits einen gefährlichen Zustand verursacht haben oder voraussichtlich verursachen werden.

Einrichtungen, durch deren Ausschalten Personen oder andere Einrichtungen gefährdet werden können, dürfen nicht abgeschaltet werden (z.B. Spannvorrichtungen, Stromversorgung von Hilfseinrichtungen, Bremsen, ..). Daher werden Steuerstrom - Kreise und Elektronik - Spannungsversorgung nur in Ausnahmefällen bei NOT - AUS abgeschaltet. Insbesondere die Spannungsversorgung der Elektronik (-Steuerung) ist nötig, um gerade im Falle einer Störung den Zustand der Anlage zu signalisieren.

In **Maschinensteuerungen** (Fertigungsautomatisierung) gilt nach VDE 0113 für NOT - AUS - Einrichtungen in elektronischen Steuerungen:

- NOT - AUS - Schalter sind im Hauptstromkreis der abzuschaltenden Geräte anzuordnen,
- Die Elektronik - Spannungsversorgung bleibt ein,
- Bei NOT - AUS - Betätigung wird die Elektronik in ihre Grundstellung gebracht ("AUS"),
- Erdschlüsse in Hilfsstromkreisen dürfen nicht
  - zum unbeabsichtigten Einschalten einer Maschine führen, oder
  - das Ausschalten verhindern.

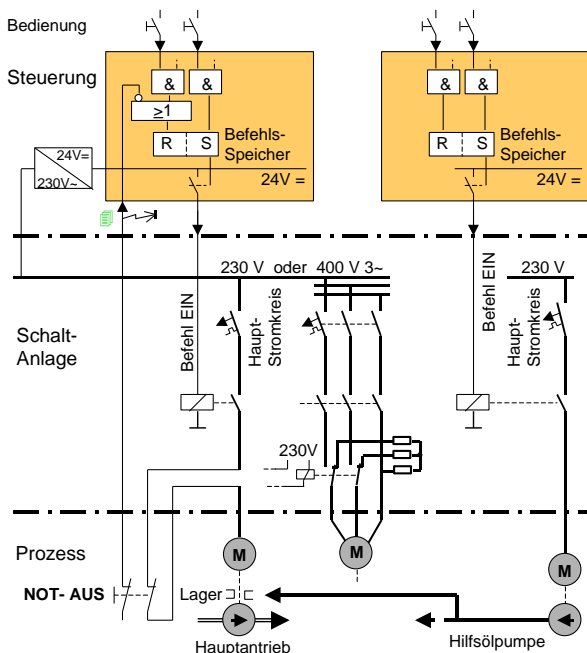


Bild 4.5.1 NOT - AUS in Maschinen - Steuerungen

Bild 4.5.1 zeigt die Steuerung eines Hauptantriebes und einer Hilfsölpumpe.

Der Hauptantrieb wird durch einen NOT - AUS - Schalter (bzw. bei 400V 3~ ein Schütz mit Umschaltern zur Bremsung) im Hauptstromkreis abgeschaltet, wobei auch der Befehlspeicher in der elektronischen Steuerung auf "AUS" zurückgesetzt wird. Dies erfolgt mit "Ruhestrom", und damit sicher gegen einen Erdschluss.

In diesem Steuerstromkreis kann ein Erdschluß weder das Einschalten noch das Ausschalten verhindern.

Für die Hilfsölpumpe ist kein NOT - AUS nötig bzw. erlaubt, da bei fehlender Ölversorgung z.B. das Lager des auslaufenden Hauptantriebes beschädigt würde.

In der **Verfahrenstechnik** werden meist größere Aggregate benötigt, die mit 400 V / 3~ betrieben werden. Hier hilft man sich mit einer NOT - AUS - Schaltung im Steuerkreis des Leistungsschützes (wird auch in der Maschinentechnik angewandt).

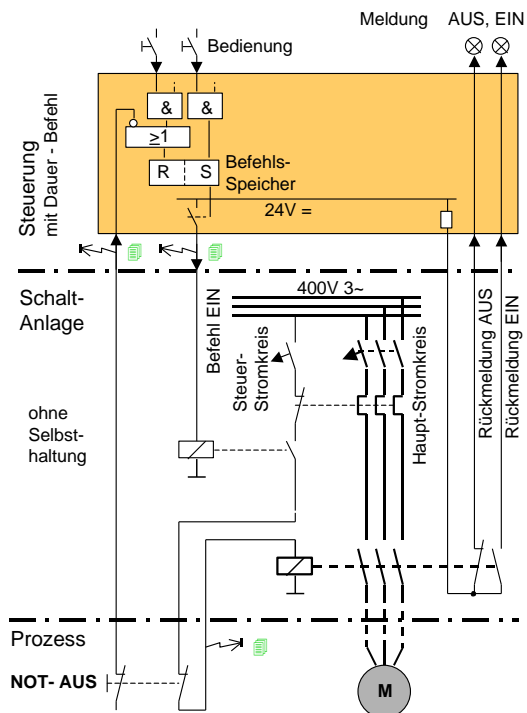


Bild 4.5.2: NOT-AUS bei 400 V, Dauerbefehl

Bild 4.5.2 zeigt ein solches Beispiel, in dem die elektronische Steuerung über ein "Koppelrelais" (meist mit den 24 V der Elektronik) das Leistungsschütz einschaltet. Der NOT - AUS - Schalter wirkt wie der Überstrom - Schutz direkt auf das Leistungsschütz.

In beiden Fällen ist durch mechanische Verklüpfung des Schalters oder Schaltungsmaßnahmen in der Elektronik dafür zu sorgen, dass nach Betätigung des NOT - AUS dieser erst bewusst "quittiert" werden muss, bevor ein neuerlicher EIN - Befehl der Bedienung möglich ist.

Bild 4.5.3 zeigt die Steuerung eines Hauptantriebs mit **Pulsbefehlen** und elektrischer Selbsthaltung in der Schaltanlage, wie insbesondere in der Kraftwerksleittechnik üblich.

Auch hier kann der NOT - AUS - Schalter nur im Stromkreis des Leistungsschalters eingefügt werden. Hier gibt es keinen Befehlsspeicher, der zurück gesetzt werden müsste. Dafür gibt es einen Meldespeicher, mit dessen Hilfe der (vom Bediener) unbeabsichtigte Zustandswechsel als "Differenz" gemeldet wird.

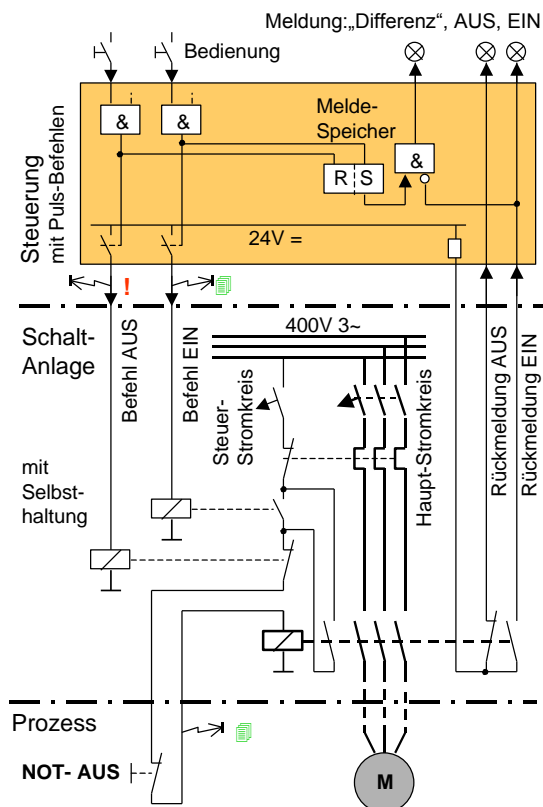


Bild 4.5.3: NOT-AUS bei 400 V, Taster - Befehle

Alle bisher dargestellten Fälle eignen sich nur für Niederspannungs - Antriebe. Für Mittelspannungs - Antriebe (6 - 10 kV) werden mechanisch verlinkte Leistungsschalter eingesetzt, und ein NOT - AUS - Schalter im Hauptstromkreis ist ganz unmöglich. Hier muss ein NOT - AUS - Schalter mit Ruhestrom aktiv in die Schaltanlage eingreifen; die Details sind abhängig von den verwendeten Komponenten.

#### 4.6 Spannungswiederkehr bei Antrieben

In Industrieanlagen muss mit einem Ausfall des elektrischen Versorgungsnetzes bzw. der Schutz - Abschaltung einer einzelnen Sammelschiene gerechnet werden.

Bei Wiederkehr der Spannung nach einem Ausfall dürfen diejenigen Antriebe nicht selbstständig anlaufen, durch deren automatischen Wiederanlauf Personen, Einrichtungen oder Produktionsgüter gefährdet werden können.

Bei Steuerungen nach Bildern 4.5.1 und 4.5.2 ist das nicht immer gewährleistet. Hier muss:

- bei Ausfall der Spannung des Hauptstromkreises auch die Elektronik - Versorgungsspannung mit ausfallen (ist so bei Maschinensteuerungen, Bild 4.4.1), oder
- der Hauptstromkreis über einen Unterspannungsauslöser geführt sein wie in Bild 4.6.1 dargestellt (EIN von Hand, Auslösung bei Unterspannung, Wiedereinschaltung von Hand), oder
- die Steuerung bei Unterspannung abgeschaltet werden wie in Bild 4.6.1.

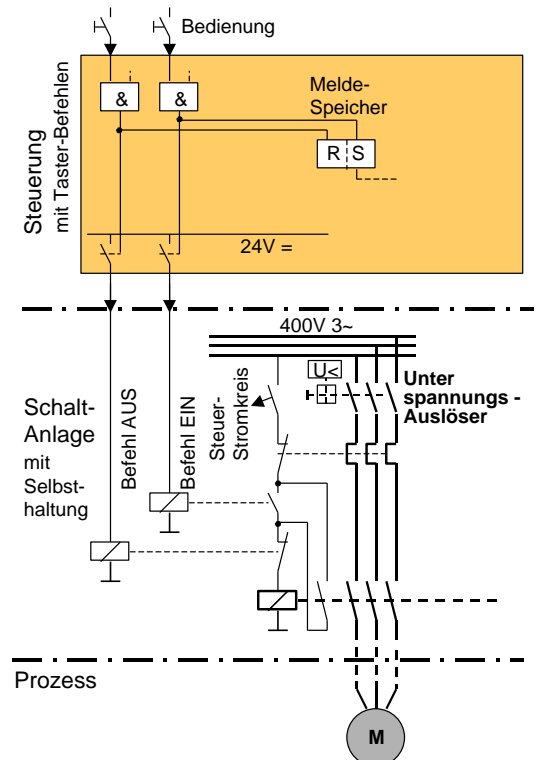


Bild 4.6.1: Abschaltungen bei Unterspannung

Bei Steuerungen nach Bild 4.5.3 kommt es darauf an, woher der Steuerstromkreis seine Spannung erhält.

Ist die Spannungsversorgung von der gleichen Schiene wie der Hauptstromkreis abgeleitet (wie in Bild 4.5.3), so fällt die elektrische Selbsthaltung ab, und da kein Befehlsspeicher verwendet ist schaltet der Antrieb später nicht mehr ein.

Wird eine von der Schiene des Hauptstromkreises unabhängige Spannungsversorgung verwendet, oder ist der Leistungsschalter verlinkt, so muss auch hier Maßnahme b oder c angewandt werden.

### 4.7 Wiedereinschaltung nach Spannungsausfall

In größeren Anlagen werden in der Starkstrom - Anlage meist redundante Einspeisungen und Sammell - Schienen verwendet. Bei Umschaltung einer Schiene auf eine andere Einspeisung (geplant oder bei Störung) fällt die Spannung an der Schiene kurzzeitig aus. Müssten nun alle angeschlossenen Antriebe von Hand wieder eingeschaltet werden (siehe 4.6), so könnte die dazu nötige Zeit den Prozess stören oder stoppen.

Wenn nötig setzt man "Wiedereinschaltgeräte" ein wie in Bild 4.7.1 dargestellt, die durch eine eingestellte zulässige "Umschaltzeit" zwischen Umschaltung und Spannungs - Ausfall unterscheiden.

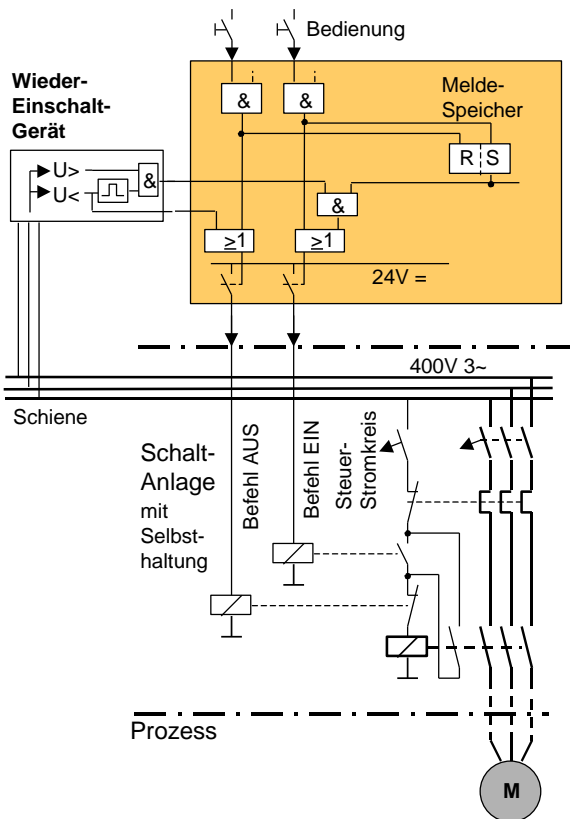


Bild 4.7.1: Wiedereinschaltgerät

Fällt die Schienen - Spannung unter einen einstellbaren " $<$ " - Wert, so wird ein definierter AUS - Befehl gegeben (verhindert "Flattern" des Leistungsschützes) und die "Umschaltzeit" gestartet. Erreicht die Schienen - Spannung innerhalb dieser Zeit einen einstellbaren " $>$ " - Wert, wird ein EIN - Befehl gegeben, der in der Steuerung solche Antriebe wieder einschaltet, die EIN waren (gemäß Meldespeicher). Dabei muss manchmal sukzessiv eingeschaltet werden, um die Sammelschiene nicht zu überlasten.

Mechanisch verlinkte Leistungsschalter bleiben zunächst EIN und erhalten nach der Umschaltzeit einen AUS - Befehl, wenn die Schienenspannung nicht über den " $>$ " - Wert gekommen ist.

### 4.8 Absteuerung von Stellantrieben

AUF / ZU - Befehle an Stellantriebe (z.B. Ventile) müssen in den Endstellungen unterbrochen ("abgesteuert") werden. Bild 4.7.1 zeigt "Weg - abhängige" Endschalter vereinfacht (ohne Leistungsschütze).

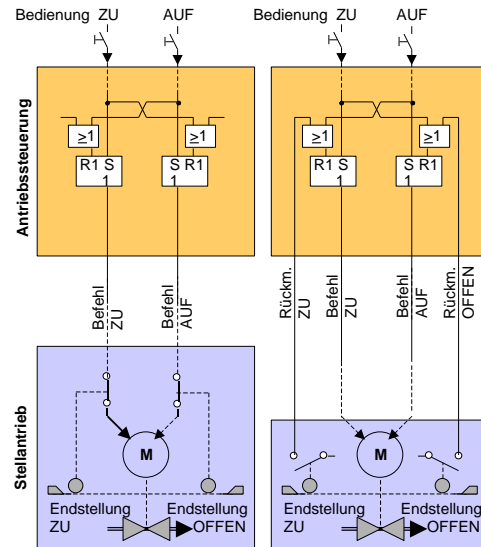


Bild 4.8.1: Weg - abhängige Absteuerung

Bei kleineren Aggregaten sind die Endschalter direkt im Stellantrieb im Hauptstromkreis angeordnet (links im Bild). Bei größeren Aggregaten werden durch die Rückmeldungen die Befehlsspeicher in der Steuerung abgesteuert.

Oft werden Drehmoment - Wächter eingesetzt, um das Aggregat bei Blockierung vor Beschädigung zu schützen. Da diese Wächter nur während eines zu hohen Drehmoments ansprechen erfolgt die Absteuerung über Speicher, die nur durch Gegenbefehle wieder zurückgesetzt werden (Bild 4.8.2). Absteuerung erfolgt oft zur Sicherheit in Ruhestrom.

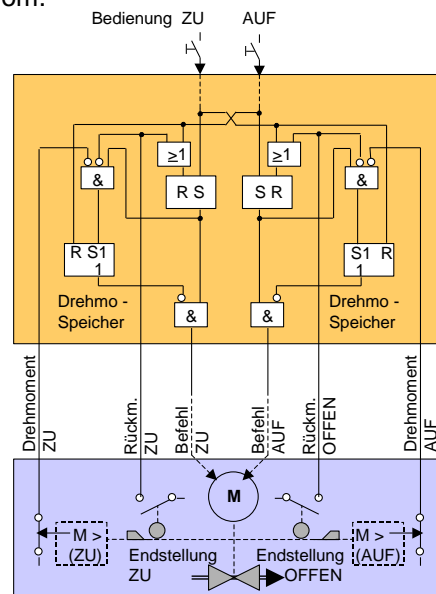


Bild 4.8.2: Drehmoment- (und Weg-) Absteuerung

## 5 Signaldefinition

Die Wirkungsweise einer Steuerung wird durch die Bool'schen (logischen) Signalpegel "1" für aktiv und "0" für inaktiv definiert. Ein UND ist z.B. erfüllt, wenn alle Eingänge "1" haben. Mit welchen physikalischen Abbildern (z.B. Spannung, Strom) die Signale übertragen werden ist eine andere Frage.

Innerhalb von Elektronik - Schaltungen wird der elektrische Signalpegel (H / L) oftmals invertiert. Da die Bauelemente aber relativ störungsarm arbeiten braucht das nicht berücksichtigt zu werden.

Bei der Übermittlung von Signalen zwischen einzeln abgesicherten und getrennt angeordneten Teilen wie Peripherie (Eingabe, Ausgabe) und Steuerung hat das physikalische Abbild beim Auftreten von Störungen aber Einfluss auf Sicherheit und Verfügbarkeit. Bei Verbindungs - programmierten Steuerungen gilt dies auch für Signale zwischen getrennt abgesicherten Steuerungsteilen.

Die häufigsten Störungen sind:

- Ausfall / Abschaltung der Versorgungsspannung einer Komponente, z.B. der Geber - Abfrage,
- Erdschluss (bei geerdeter Elektronik - Spannung gleiche Wirkung wie Spannungsausfall),
- Leitungsunterbrechung (ebenfalls wie Spannungsausfall wirkend).

Wird "Spannung vorhanden" (und fließender Strom) zur Abbildung von "1" verwendet, so bewirken obige Störungen eine "0", also den nicht aktiven Zustand, z.B. "kein Befehl".

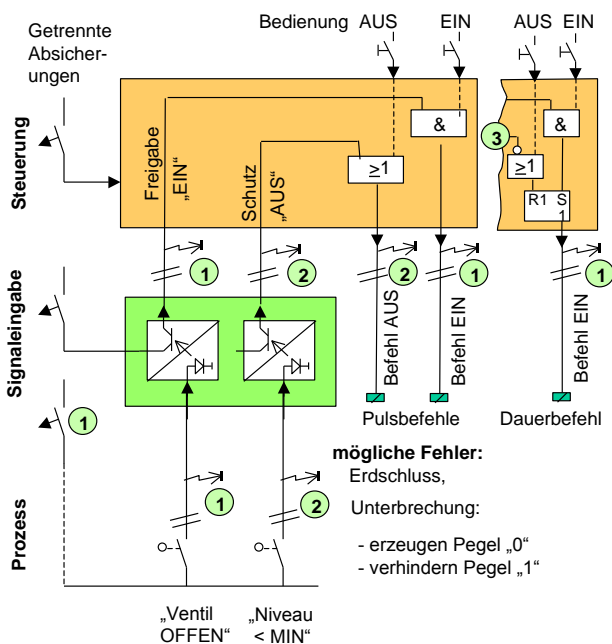


Bild 5.1: Fehler bei der Signalübertragung

Bild 5.1 zeigt die Antriebssteuerung z.B. einer Pumpe mit Freigabe und Schutzbefehl über eine Signaleingabe (getrennte Karten) und Kontakte im Prozess.

Die Pumpe

- darf eingeschaltet werden wenn das Ventil OFFEN ist,
- wird durch Schutz - AUS abgeschaltet wenn das Niveau < MIN ist (Trockenlauf-Schutz).

Links ist eine Steuerung mit Puls - (Taster-) Befehlen und rechts mit Dauerbefehl dargestellt. An den Verbindungen sind die möglichen Fehler dargestellt und nummeriert.

- Nr. 1 verhindert einen EIN - Befehl,
- Nr. 2 verhindert einen Schutz-AUS - Befehl,
- Nr. 3 bewirkt eine Abschaltung wegen
  - invertierendem Eingang und
  - Dauerbefehl

Bild 5.2 zeigt eine Bewertung im Blick auf Sicherheit und Verfügbarkeit.

**Verfügbarkeit:**

- Einschalten nicht verhindern und
- nicht ungewollt Ausschalten

**Sicherheit:**

- Ausschalten nicht verhindern und
- nicht ungewollt Einschalten

**Fehler bewirkt:**

| Nr. |                         | sicher | verfügbar |
|-----|-------------------------|--------|-----------|
| 1   | verhindert EIN - Befehl | ja     | nein      |
| 2   | verhindert Schutz - Aus | nein   | ja        |
| 3   | erzeugt Schutz - Aus    | ja     | nein      |

**Kompromiss:**

Befehle, Freigabe: mit Pegel „1“  
bewirkt keine ungewollten Aktivitäten

Bild 5.2: Verfügbarkeit und Sicherheit

Verfügbarkeit und Sicherheit einer Steuerungseinrichtung verlangen gegensätzliches Verhalten. Beiden Forderungen wird nur eine redundante Anordnung mit zumindest drei Kanälen gerecht, wie für spezielle Schutzsysteme verwendet.

Für "normale" betriebliche Steuerungen ist die Verhinderung von Aktivitäten durch auftretende Fehler ein brauchbarer Kompromiss, also das Abbild von "1" durch Spannung (und Strom), auch "aktiv - 1" genannt.

In Ausnahmefällen wird die umgekehrte Definition verwendet (also Schalten durch "0"), wenn der Sicherheit Vorrang zu geben ist (siehe Fehler Nr. 3)

## 6 Verknüpfungssteuerung

Wesentliches Merkmal einer Verknüpfungssteuerung ist die Zuordnung der Eingangs- und Ausgangssignale der Steuerung im Sinne Boole'scher Verknüpfungen, ohne daß die Ausgangssignale voneinander abhängig sind.

Verknüpfungssteuerungen enthalten - im Gegensatz zu den Ablaufsteuerungen - in der Regel kein definiertes Ein / Aus - Programm als feste Befehlsreihenfolge. (Liefße sich durch Logik nur ziemlich kompliziert und unübersichtlich realisieren).

Verknüpfungssteuerungen sind vorwiegend mit den elementaren Verknüpfungsgliedern UND, ODER, NICHT aufgebaut; teilweise werden auch Zeit- und Speicherglieder benötigt.

Hauptanwendungsbereiche der Verknüpfungssteuerungen sind:

- Realisierung einer Steuerung kleinen Umfangs,
- Aufbau der Antriebssteuerung (mit Freigaben, Verriegelungen, Schutz) als unterste Ebene einer strukturierten Steuerung, insbesondere in umfangreichen Prozeßsteuerungen,
- Kleine Programme, z.B. AUS - Programme in Ablaufsteuerungen, wenn diese im gewählten Produkt durch Ablaufketten nur mit großem Aufwand aufgebaut werden können,
- Verknüpfung der Grundbedingungen im Betriebsarten - Teil von Ablaufsteuerungen.

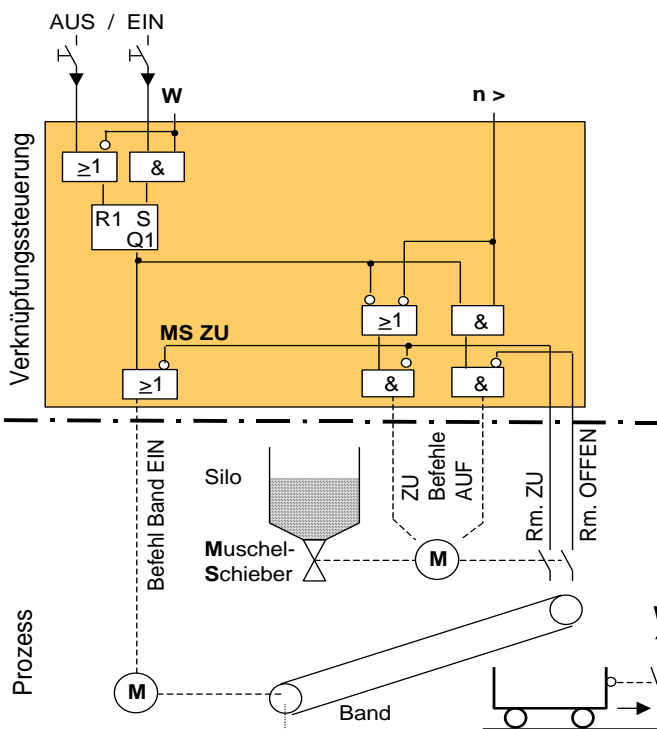


Bild 6.1: Unstrukturierte Verknüpfungssteuerung

Bild 6.1 zeigt eine Verknüpfungssteuerung für eine kleinere Anlage, hier: ein Silo mit Band. Wenn der Transportwagen in Position steht (W) kann eingeschaltet werden. Das bedeutet, dass zuerst das Band gestartet werden muss und dann der "Muschelschieber" (Silo - Material - Austritt) geöffnet wird. Dazu ist eine Zeit nötig (z.B. 10 s).

Die Positionsmeldung W und die Rückmeldungen "Band - Drehzahl n >" sowie "Muschelschieber ZU" dienen als Freigaben bzw. auch als Schutz-Befehle.

Die Steuerung der Antriebe und der automatische Ablauf (Band EIN, dann Schieber AUF) sind hier in einer einzigen Steuerung realisiert, die Funktionen minimiert. Dies führt oft zu Missverständnissen und Planungsfehlern, und Änderungen haben oft ungewollte Auswirkungen.

Strukturiert sähe die Lösung so aus wie in Bild 6.2 dargestellt (vereinfacht):

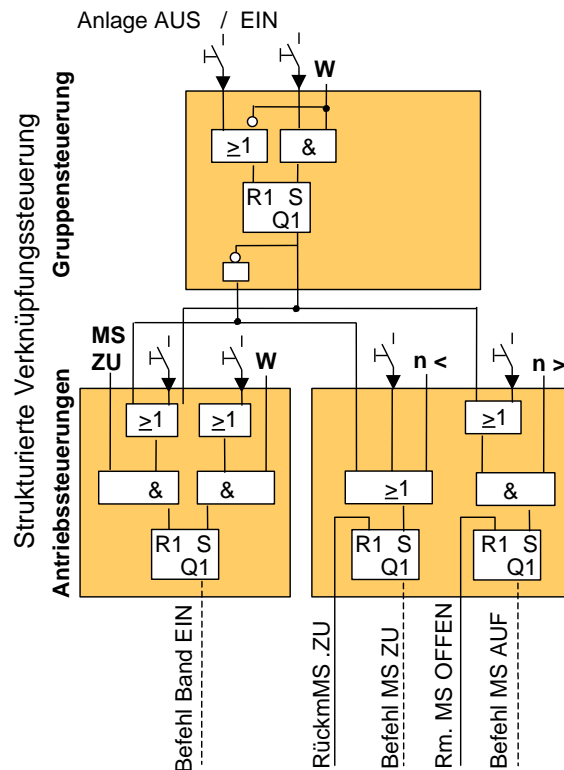


Bild 6.2: Strukturierte Verknüpfungssteuerung

Hier ist das Anfahren der Anlage in einer Gruppensteuerung realisiert. Jeder Antrieb hat eine eigene Antriebssteuerung mit Handbedienung und Berücksichtigung der für ihn geltenden Freigaben und Schutzengriffe. So lässt sich jeder Bereich besser übersehen, was zu weniger Fehlern führt.

Hier fällt wohl auch auf, dass sich in einer strukturierten Lösung die Verwendung von standardisierten Funktionsblöcken (z.B. "Antriebssteuerung") mit Standard - Anschlüssen (Freigabe, Schutz - Befehl) anbietet, insbesondere bei Anlagen mit einer größeren Zahl von Antrieben.



## 7 Ablaufsteuerung

### 7.1 Prinzip

Ablaufsteuerungen geben Befehle in vorgegebener Reihenfolge, abhängig von Übergabebedingungen (Prozess - Signale, Zeit). Sie dienen der Steuerung von diskontinuierlichen Prozessen (Abläufen), z.B. Anfahren einer Turbine).

Eine Ablaufsteuerung besteht aus einer (oder mehreren) *Ablaufkette(n)*. Eine Ablaufkette besteht aus einem oder mehreren Netzwerk(en) von *Schritten*. Ein Schritt bewirkt eine (oder mehrere) *Aktion(en)*. Die *Transition* (Übergang) zum nächsten Schritt wird durch eine (oder mehrere) Übergabebedingung(en) freigegeben. Dies sind meist Rückmeldungen von der Ausführung der Aktion des aktuellen Schrittes. (Bild 7.1.1)

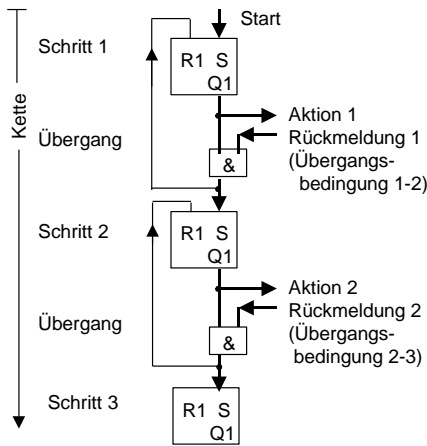


Bild 7.1.1: Ablaufsteuerung, Prinzip

In einer Kette bewirkt eine erfüllte Transition - das Setzen des nächsten Schrittes, und - das Rücksetzen des vorhergehenden Schrittes, daher ist in einer Kette jeweils nur ein Schritt aktiv. Dadurch kann eine Kette gegensinnige Befehle enthalten wie im Beispiel in Bild 7.3.1. Bei Speicher - programmierter Ausführung ohne besondere Maßnahmen ist jedoch zu beachten, dass im Zyklus der Übergabe beide Schritte aktiv sind!

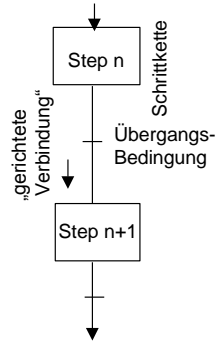


Bild 7.1.2: AS - Sprache

Die links kursiv geschriebenen Begriffe entstammen der Norm DIN EN 61131. Dort ist auch eine spezielle Sprache und Darstellung von Ablaufsteuerungen (AS) festgelegt. Bild 7. zeigt eine Anwendung.

### 7.2 Struktur

Eine Ablaufsteuerung wird meist als übergeordnete Steuerung ("Gruppensteuerung") eingesetzt, und muss daher zumindest zwischen den Betriebsarten HAND und AUTO umschaltbar sein (Bild 7.2.1). Die Gruppensteuerung ist nur in Betriebsart AUTO aktiv. In HAND gibt sie keine Befehle, nun kann nur auf der Ebene der Antriebssteuerung eingegriffen werden. Manchmal werden diese Betriebsarten auch "Aktiv" und "Deaktiv" genannt, damit "HAND" und "AUTO" für Befehle an die Gruppensteuerung von HAND oder von einer übergeordneten Steuerung reserviert bleiben.

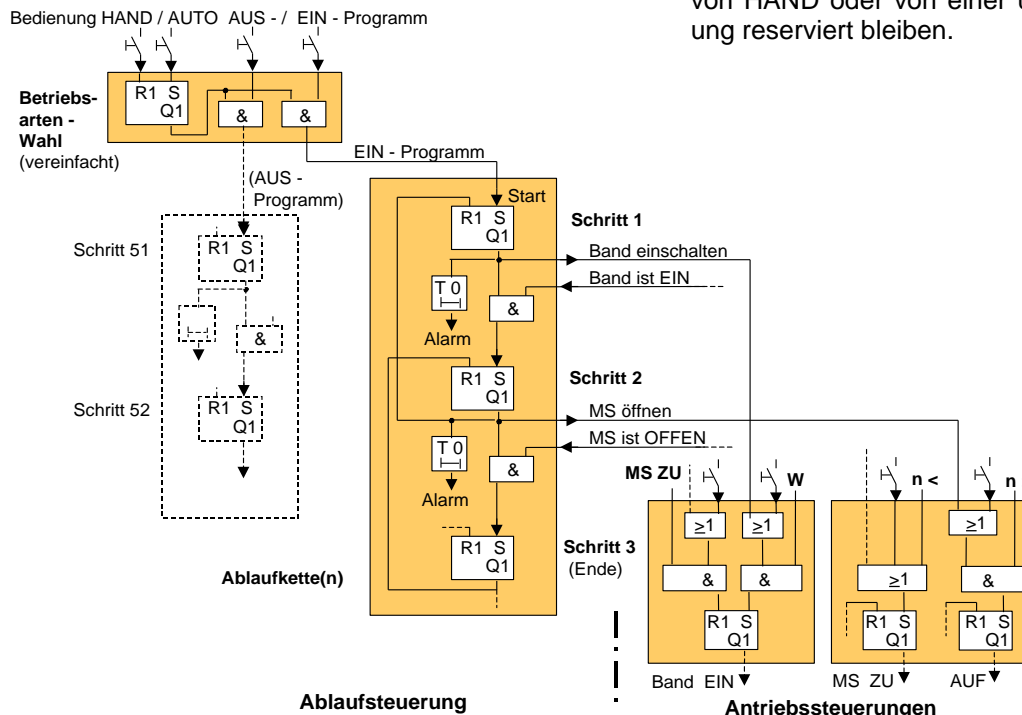


Bild 7.2.1 Ablaufsteuerung, Struktur

### 7.3 Anwendungsbeispiel

Bild 7.3.1 zeigt die Gruppensteuerung einer Pumpe mit ihrem nachfolgenden Absperrschieber, die jeweils eigene Antriebssteuerungen (Verknüpfungssteuerungen) besitzen. Zum Start muss zuerst der Schieber geschlossen werden (Anlauf ohne Last), dann kann die Pumpe eingeschaltet und der Schieber wieder geöffnet werden.

Jedem Schritt ist eine Überwachungszeit zugeordnet, die etwas länger gewählt wird als die normalerweise zur Befehlsausführung nötige Zeit. Wird ein Befehl nicht "rechtzeitig" ausgeführt, so erfolgt ein Alarm.

Der Übergang von Schritt 1 nach 2 ist mit einer "Überlaufbedingung" versehen. Wenn die Pumpe schon läuft und das Programm gestartet wird, springt das Programm gleich in Schritt 2, damit der Schieber nicht unnötigerweise geschlossen wird.

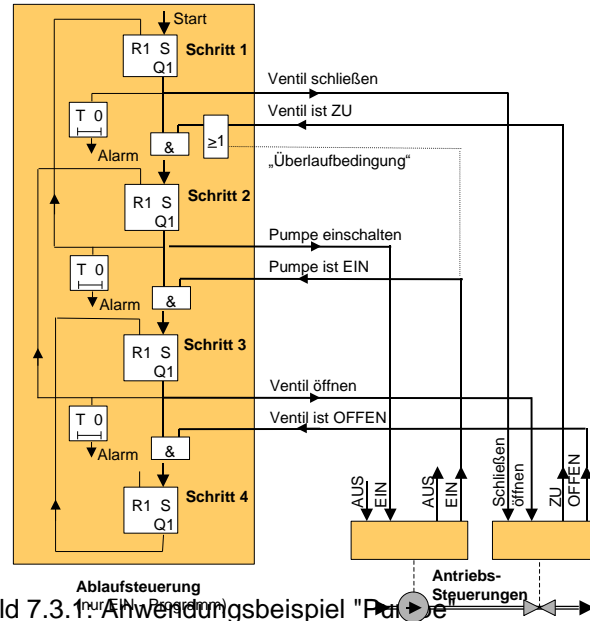


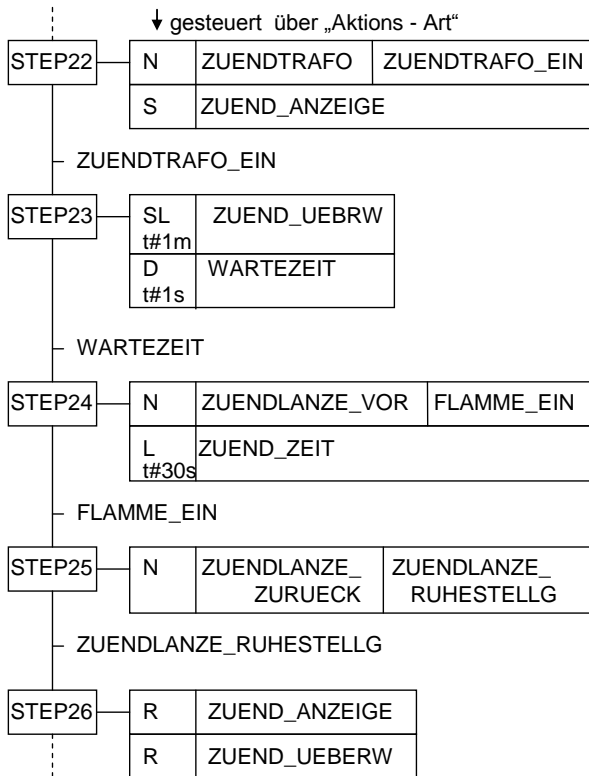
Bild 7.3.1: Anwendungsbeispiel "Pumpe"

### 7.4 Aktions - Steuerung

In DIN EN 61131 sind verschiedene Möglichkeiten zur Steuerung einer "Aktion" (meist Befehlsgabe) eines Schrittes festgelegt. Bild 7.4.1 zeigt diese rechts und dazu links ein Beispiel in "AS" - Sprache. Hierin ist rechts von einem Schritt ein "Aktions - Block" dargestellt, der Aktionsart, Aktion und - optional - die Rückmeldung enthält. Einem Schritt können mehrere Aktionen zugeordnet sein. (Mehr dazu im Kapitel "Engineering")

In dieser Sprache bedeutet .X das "Aktiv" - Signal eines Schrittes.

Ein / Ausgänge der verschiedenen Aktions - Steuerungsmöglichkeiten sind hier auf das Beispiel bezogen bezeichnet. In der Praxis enthält eine Steuerungseinrichtung einen Funktionsbaustein mit diesen Möglichkeiten, aus dem durch die "Aktionsart" im Schritt - Symbol das gewünschte Verhalten aufgerufen wird.



---> N: Nicht speichernd, S: Speichernd, L: zeitlich begrenzt (Limitiert), D: verzögert (Delayed), R: Rücksetzen

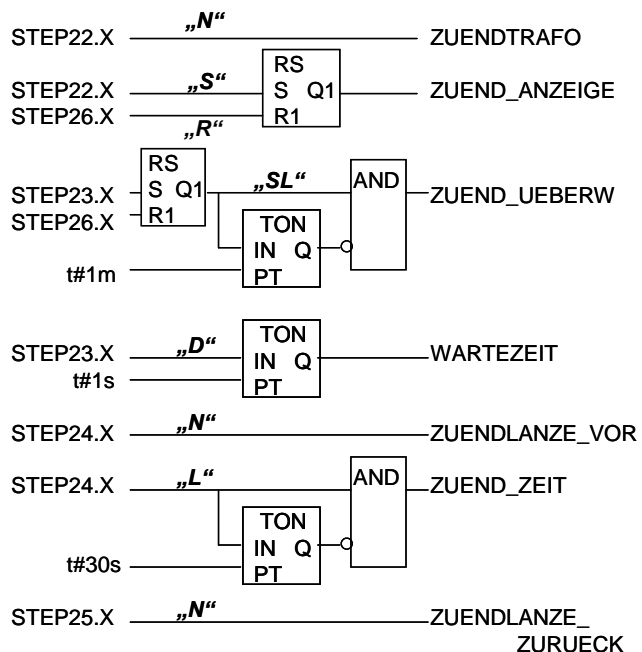


Bild 7.4.1 Aktionssteuerung nach DIN EN 61131

### 7.5 Kettenstrukturen

Ablaufketten können linear sein oder Verzweigungen mit parallelen Ketten oder Sprünge enthalten. In DIN EN 61131 sind verschiedene Strukturen definiert (Bild 7.5.2).

In Bild 7.5.1 ist als Anwendungsbeispiel vereinfacht ein EIN - Programm für einen kombinierten Öl/Gas-Brenner dargestellt, bei dem nach einem gemeinsamen Anfang je nach gewähltem Brennstoff eine andere Schritt - Gruppe zu durchlaufen ist.

Aus Erfahrung lässt sich sagen, dass eine Strukturierung der Gesamtaufgabe in separate Ablaufsteuerungen besser zu überschauen ist als eine raffinierte Ablaufketten- Struktur innerhalb einer Ablaufsteuerung.

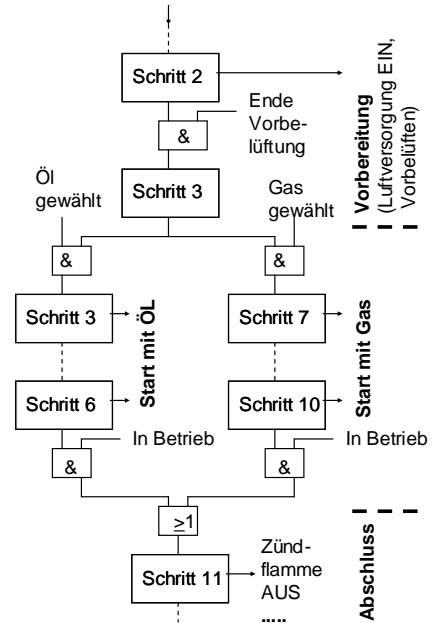
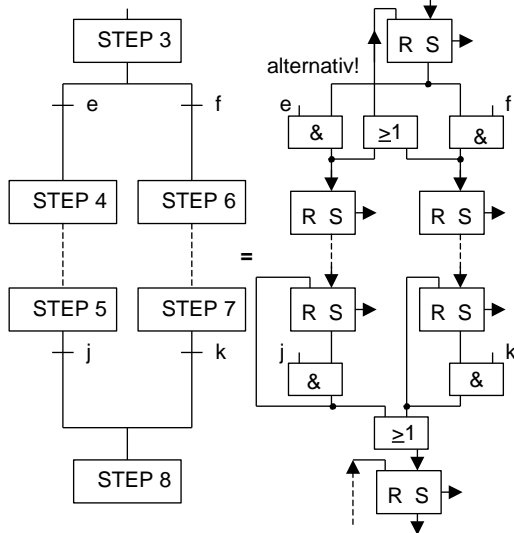
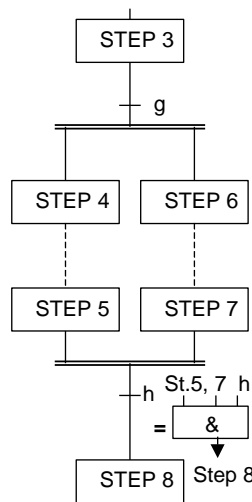


Bild 7.5.1: Kettenverzweigungs - Beispiel

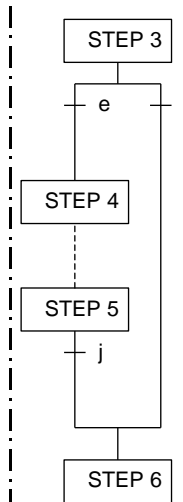
#### Verzweigung / Zusammenführung bei Kettenauswahl



#### Simultankette



#### Ketten - Sprung



#### Ketten - Schleife

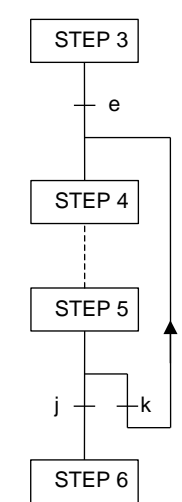


Bild 7.5.2: Ablaufketten - Strukturen  
(Begriffe nach DIN EN 61131)

## 8. Realisierung

### 8.1 SPS / PLS

Steuerungen (und auch Regelungen) in der Prozessautomatisierung sind heute als **SPS** (Speicher – Programmierbare Steuerung) oder **PLS** (Prozess – Leit – System) realisiert.

Bild 8.1.1 zeigt die typischen Strukturen von SPS und PLS. Tabelle 8.1 die typischen unterschiedlichen Eigenschaften.

Eine **SPS** wird allgemein für kleinere Prozessbereiche oder eine Werkzeugmaschine / eine Fertigungszelle eingesetzt. Da die Gesamtkosten solcher Einrichtungen begrenzt sind, ist auch der Anteil der SPS begrenzt und damit ihr Komfort und ihre Ausstattung. SPS sind kompakt in einem Gehäuse oder modular auf Hutschiene aufgebaut. Die Prozesssignale sind über integrierte oder ausgelagerte (RIO) Ein / Ausgabegeräte angeschlossen, Bedienung ist integriert oder als PC mit Standard – SW anschließbar, und heute ist eine Schnittstelle zum Betriebsnetz oder Internet normal. Ein Engineeringtool (auf PC) kann meist direkt angeschlossen werden.

Heutige SPS enthalten auch schon Ein-/Ausgabegeräte mit Steuerungsfunktionen (ABB).

Von **PLS** spricht man bei größeren automatisierten Anlagen, wie sie hauptsächlich in der Verfahrenstechnik vorkommen. Die Signal – Verarbeitung (Steuerung / Regelung / Überwachung) erfolgt in Prozess – Stationen (en: Controller), die verteilt oder sogar dezentral angeordnet und über einen redundanten Prozessbus (Echtzeit!) verbunden sind. Die Prozesssignale sind über Ein / Ausgabegeräte in den Stationen oder in Remote I/Os (RIO) angeschlossen, oder Sensoren / Aktoren sind direkt Feldbus – fähig. Zunehmend werden intelligente Feld – Geräte verwendet, die teilweise Einzel – Steuerungs- / Regelungsaufgaben übernehmen, so dass nur die übergeordneten Aufgaben (Funktionsgruppen, Gesamtanlage) in den Stationen verbleibt.

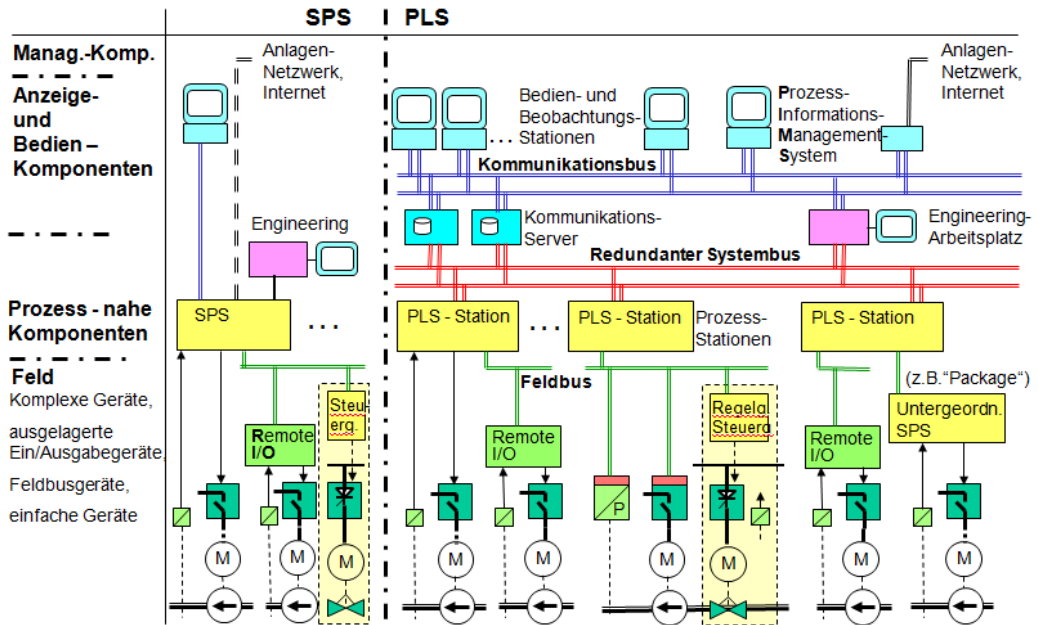


Bild 8.1.1: SPS / PLS - Strukturen

Tabelle 8.1: SPS / PLS - Eigenschaften

| Speicher – Programmierbare Steuerung <b>SPS</b>   | Prozess – Leit - System <b>PLS</b>   |
|---|--|
| <b>Anwendung:</b> - kleinere Prozessbereiche, (z.B. Eine Werkzeugmaschine / Zelle)<br>- Geringer Automatisierungsgrad       | - Große Anlagen<br>(z.B. Walzwerk-Straße, Kraftwerksblock, Chemiefabrik)<br>- Hoher Automatisierungsgrad   |
| <b>Struktur:</b> Zentral (für die entsprechende Einheit)  | Verteilt bis dezentral / Anlagen-nah (bezogen auf die Anlage),<br>Verlagerung von Einzelsteuerung in intelligente Feldgeräte   |
| <b>Engineering:</b> „preisgünstig“, daher eingeschränkt<br><br>- Funktionen / Funktionsbausteine mindestens gemäß DIN 61131 | - Komfortabel, Multi – User – fähig mit zentraler Datenbank<br>- durchgehend grafisch, FB – Deklaration nicht nötig<br>- Mehr Funktionen / Funktionsbausteine als in DIN 61131, Anwendungsspezifische FB erstellbar und vom Hersteller   |
| <b>Funktionalität:</b> - Für Sicherheits – relevante Aufgaben: Fehler - tolerant oder mit integr. Sicherh.                  | - Für Sicherheits – relevante Aufgaben: Schutzsysteme oder PLS - Teile Fehler - tolerant oder mit „integrierte Sicherheit“<br>- projektierbare höhere Verfügbarkeit durch gezielte redundante Eingabe, Übertragung, Verarbeitung, Anzeige<br>- Vorkehrungen zur Qualitätssicherung in allen Phasen (Zugang, Änderungs – Dokumentation)<br>- Integrierte Datenaufbereitung für das Asset Management |
| <b>„Höhere“ Funktionen:</b>   | - Software – Pakete für die Rezeptfahrweise<br>- Selbsteinstellende / adaptive Regler, Gain scheduling<br>- Modell – gestützte Messwerterfassung („Prozessbeobachter“)<br>- für nicht direkt messbare Größen<br>- Modell – gestützte prädiktive Regelung   |

Insbesondere weil manchmal Anlagenteile mit SPS als „Package“ geliefert werden, kommen in PLS auch untergeordnet SPS vor.

Für die Bedienung sind hier viele Bedienplätze notwendig, so dass redundante Server für Prozessdaten und Bilder verwendet werden, die über einen meist redundanten Kommunikationsbus untereinander verbunden sind. Meist auf eigenen PCs läuft ein „PIMS“ (Archivierung, Statistik usw.). Vom Kommunikationsbus aus wird zum Anlagennetz und Internet verbunden.

Sicherheit und höhere Verfügbarkeit können meist projektiert werden, das Engineering ist komfortabler und für gleichzeitige Bearbeitung durch mehrere Bearbeiter ausgelegt, und insbesondere für die Regelung stehen „höhere“ Funktionalitäten zur Verfügung.

## 8.2 PC – basierte SPS

Speicher – programmierbare Steuerungen (SPS) waren als spezielle HW – und SW – Lösungen für den Einsatz in der Fertigungs- und Prozessautomatisierung entwickelt worden.

Als es PCs gab und diese als „Industrie – PCs“ robust genug für die industrielle Umgebung wurden, begann man sie auch als Hardware – Plattform für SPS einzusetzen.

Bild 5.1 zeigt verschiedene Lösungen im Vergleich zu einer „normalen“ SPS.

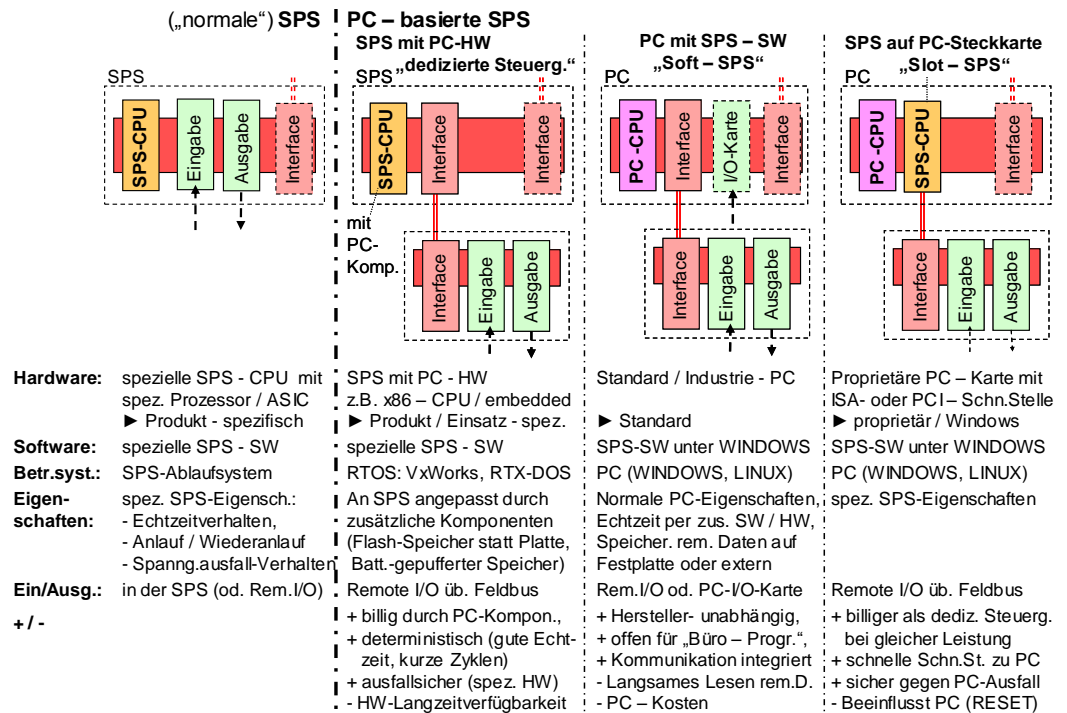


Bild 5.1: PC – basierte SPS - Lösungen

Als „**dedizierte Steuerung**“ (Einsatz – spezifisch) bezeichnet man eine SPS, bei der PC – HW-Komponenten benutzt werden, z.B. ein x86er Prozessor. Als Betriebssystem wird ein Realtime - System eingesetzt, als Speicher anstelle einer Platte Flash-Speicher oder Batterie- gepufferte RAMs (zur Vermeidung Erschütterungs- empfindlicher Teile). Die Verbindung zum Prozess erfolgt über Remote I/O per Feldbus.

Mechanisch steckt eine solche Lösung in einem speziellen SPS – Gehäuse und ist oft „eingebettet“ (embedded) in ein größeres System, z.B. eine Werkzeugmaschine.

Diese Lösung ist durch die Standard-Komponenten billig, diese sind aber wegen der raschen Entwicklung bald überholt und nicht mehr verfügbar.

Hierhin gehören auch „PAC“ (Programmable Automatic Controller, siehe Grundlagen, 6.2).

Als „**Soft – SPS**“ wird eine Steuerung bezeichnet, die als SW – Paket auf einem normalen PC läuft. Echtzeitverhalten und Sicherheit gegen Datenverlust bei Spannungsausfall muss mit zusätzlicher HW/SW

Literatur – Hinweise:

- [1] **Automatisierungstechnik in der Fertigung** (Grundlagen, Komponenten und Systeme) Autorenteam, Bibliothek des Technikers BDT oder
- [2] **"Automatisierungs- Technik"**, Autorenteam, Lektorat: Prof. Dr. Dietmar Schmidt (in der Bibliothek der BA Mannheim mehrfach vorhanden)
- [3] Systembeschreibungen zu Prozessleitsystemen (ABB, SIEMENS)
- [4] „PC – basierte Steuerungen“, Claus Kühnel, Phoenix-Contact, in „Steuerungstechnik aktuell“ (Oldenburg)
- [5] „Prozessleitsysteme“, Uwe Maier, Uni Duisburg-Essen, atp 10/2004

hergestellt werden. Verbindung zum Prozess durch PC – I/O – Karte oder Remote I/O über Feldbus.

Diese Lösung ist HW - Hersteller – unabhängig und von sich aus offen für den Zugriff durch „Büro – Programme“, aber relativ langsam und benötigt – wenn nicht schon vorhanden – einen relativ teuren PC.

Als „**Slot – SPS**“ werden Steuerungen bezeichnet, die auf einer speziellen PC – Einsteckkarte laufen (ISA- oder PCI – Schnittstelle). Diese Hersteller – spezifische Lösung („proprietär“) enthält eine spezielle SPS – SW, die unter Windows oder LINUX läuft und besitzt die speziellen SPS – Eigenschaften (Anlauf / Wiederanlauf – Verhalten, Echtzeit, Datensicherung bei Spannungsausfall). Sie ist bei gleicher Leistung billiger als eine dedizierte Steuerung, hat eine schnelle Schnittstelle zum PC, aber beeinflusst diesen auch (z.B. bei RESET).

Hier muss von Fall zu Fall entschieden werden, welche Lösung die geeignete ist.