

2.1 Arten von elektrischen Prozess-Signalen

- analoge Prozess-Signale
 - * amplitudenanalog
 - * frequenzanalog
 - * phasenanalog

- binäre Prozess-Signale

Schalterstellung

- digitale Prozess-Signale

n-Bit-Wort

- Prozess-Signale in Form von Impulsen

Drehzahlgeber

- Prozess-Signale in Form von Impulsflanken

Zustandsübergang einer Prozessgröße

Arten von Aufgaben bezüglich der Prozess-Signale

- Ein- und Ausgaben analoger Prozess-Signale
 - * Umwandlung analoge Information in Dualzahl mit Analog-Digital-Umsetzer (ADU)
 - * Umwandlung Dualzahl in analoge Information mit Digital-Analogo-Umsetzer (DAU)

- Ein- und Ausgabe digitaler Prozess-Signale (inklusive binärer Prozess-Signale)

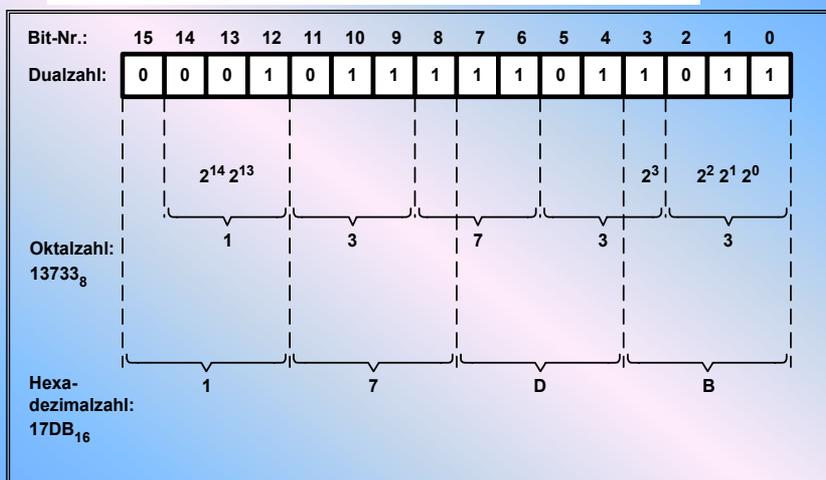
- Ein- und Ausgabe impulsförmiger Prozess-Signale

Umsetzung analoger Prozessgrößen

Darstellungsfehler (% vom Max.Wert)	Wertverschlüsselung des Analog-Digital-Umsetzers		Erford. Wortlänge (Bytes)
	Bits	Dezimalzahlen	
0,5%	8	0 ... 255	1
0,1%	10	0 ... 1 023	2
0,025%	12	0 ... 4 095	2
0,006%	14	0 ... 16 383	2

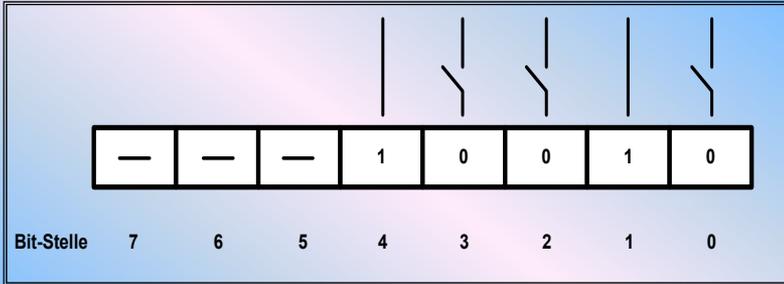
wichtig: zur Darstellung analoger Prozessgrößen reicht eine Wortlänge von 16 Bit aus

Darstellung einer analogen Prozessgröße als Festpunkt-Dualzahl von 16 Bit



wichtig: Für die Darstellung digitaler Prozessgrößen reicht eine Wortlänge von 16 Bit in aller Regel aus

Beispiel für die Darstellung von 5 Kontaktstellungen in einem 8-Bit-Wort



wichtig: Für die Darstellung binärer Prozessgrößen und impulsförmiger Prozessgrößen reicht eine Wortlänge von 16 Bit im allgemeinen völlig aus

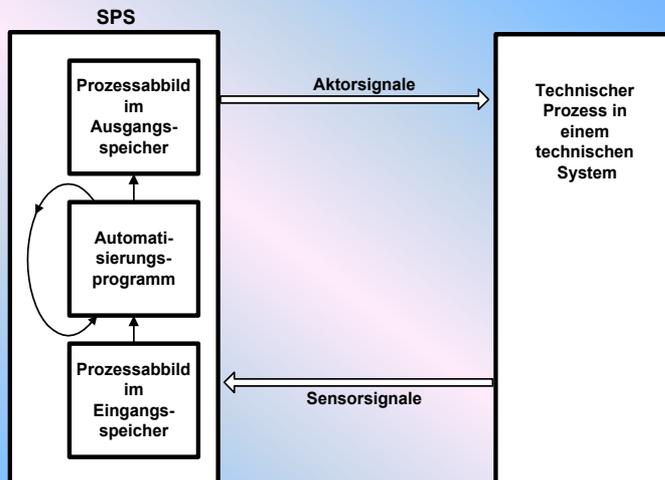
Frage: Wozu 32-Bit-Rechner für Automatisierungsaufgaben?

- ⇒ größerer Adressraum
- ⇒ größere Rechengeschwindigkeit

2.2 Automatisierungs-Computer

Speicherprogrammierbare Steuerungen (SPS)

Zyklischer Betrieb beim Einsatz einer speicherprogrammierbaren Steuerung als Automatisierungs-Computer



Eigenschaften von SPS

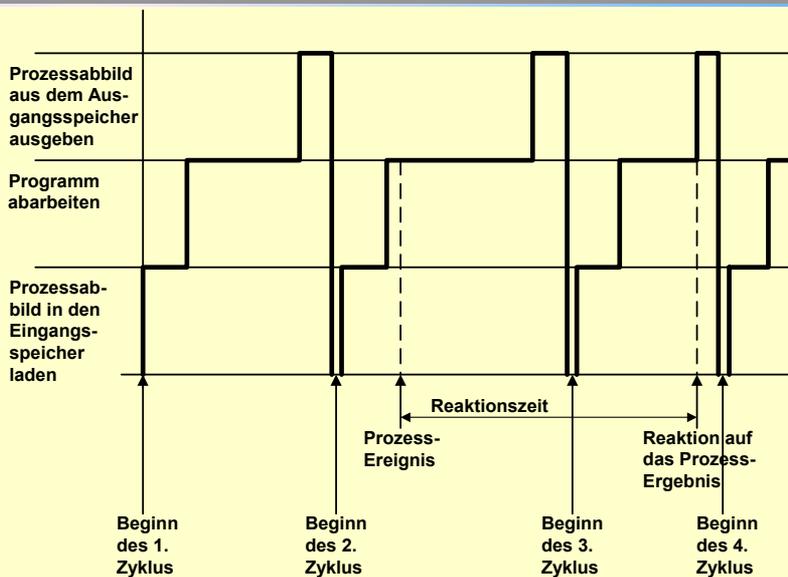
Vorteil: einfache Programmierung durch zyklische Betriebsweise

Nachteil: Reaktionszeit auf Ereignisse im technischen Prozess
maximal zwei Programm-Zyklen

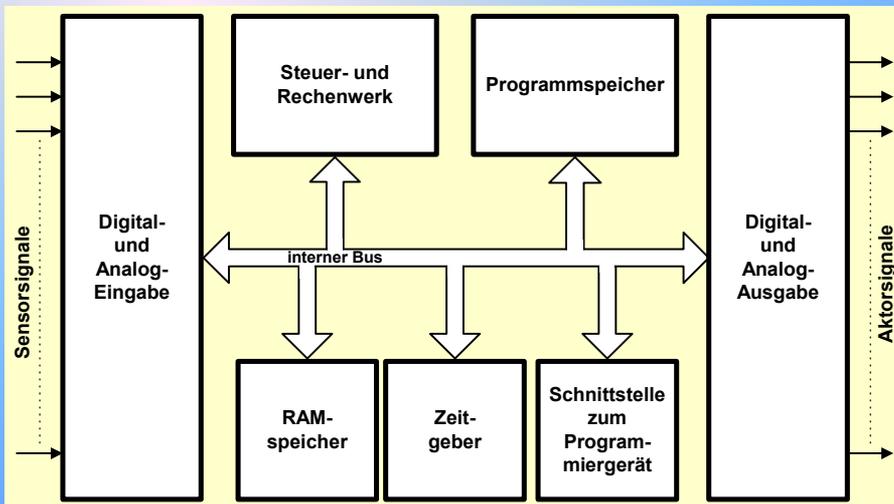
Programmabarbeitungszeit:

- ↪ Zykluszeit nicht konstant
- ↪ 1 ms pro 1000 Anweisungen

Ablauf des zyklischen Programmbetriebes bei einer SPS



Blockschaltbild der Hardwarestruktur einer SPS



Mikrocontroller

Einsatzgebiete als Serien- oder Massenprodukt

Produktautomatisierung

- ⇒ Haushaltsgeräte
- ⇒ Unterhaltungselektronik
- ⇒ Kfz-Elektronik

Eigenschaften von Mikrocontrollern

- kurze Wortlänge
- Integration von Prozessperipherie auf dem Chip
- extrem niedriger Preis 1 - 10 DM
- hohe Anforderungen bezüglich Umgebungsbedingungen Temperatur, Feuchtigkeit
- hohe Zuverlässigkeit und Lebensdauer



Unterscheidung

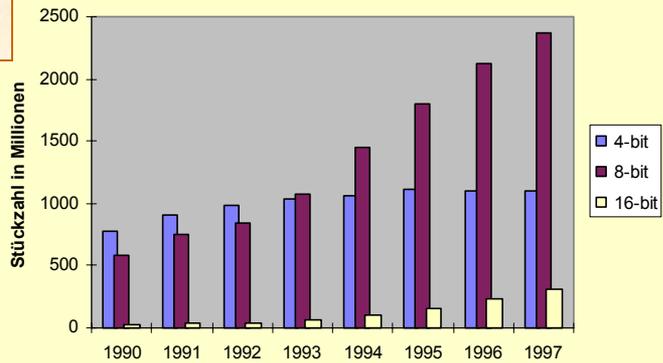
- Mikroprozessor**
 - * Prozessor auf einem Mikroelektronik-Chip
- Mikrocomputer**
 - * Alle Komponenten auf einem Mikroelektronik-Chip, d.h. Prozessor, Speicher, Schnittstellen zur Peripherie
- Mikrocontroller**
 - * Automatisierungs-Computer bzw. ein Automatisierungs-Computersystem auf einem Chip



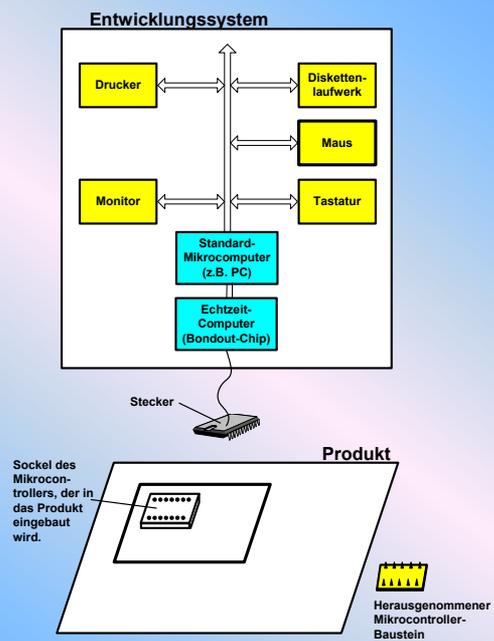
Weltweiter Mikrocontroller-Markt

Herstellerfirmen:

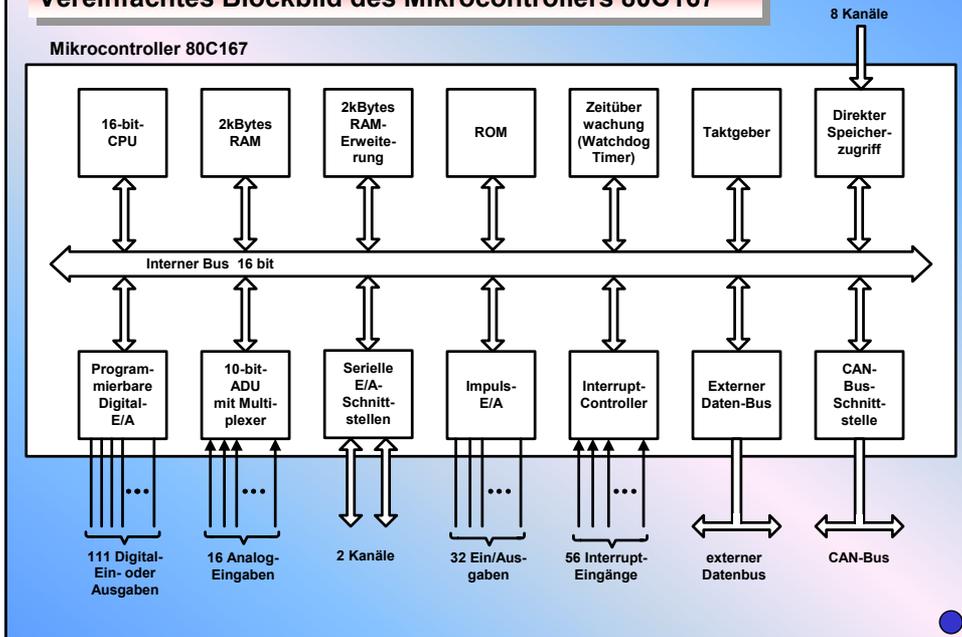
- Intel
- Motorola
- National Semiconductor
- Toshiba



Anwendung einer Entwicklungsumgebung zum Test von Programmen für einen Mikrocontroller (In-Circuit-Emulation)



Vereinfachtes Blockbild des Mikrocontrollers 80C167



Begriffe

CPU = Central Processing Unit

Mikroprozessor

RAM = Random Access Memory

Arbeitsspeicher

EPROM/PROM/ROM = Erasable Programmable Read Only Memory

Festwertspeicher

I/O = parallele bzw. serielle Ein/Ausgabe-Bausteine

Prozess- und Datenperipherie

Counter/Timer = Taktgeber

Interrupt Controller = Unterbrechungswerk

Industrie-PC

- **Rauhe Umgebungsbedingungen**
 - * Temperaturschwankungen
 - * Stöße und Erschütterungen
 - * Staub und Feuchtigkeit
 - * elektrische oder elektromagnetische Störungen

- **Schutzvorrichtungen von Industrie-PCs (IPC)**
 - * schwingungsgedämpfte Laufwerke
 - * hohe Güte der integrierten Bausteine
 - * spezielles Schutzgehäuse

Schutzart von Industrie-PCs mit IP-Index (Ingress Protection)

1. Ziffer	Schutz gegen Festkörper	2. Ziffer	Schutz gegen Wassereinwirkung
0	kein Schutz	0	kein Schutz
1	Handkontakt unmöglich (50mm Objekte)	1	Schutz gegen vertikal fallende Tropfen/ Kondensation
2	Fingerkontakt unmöglich (12mm Objekte)	2	Schutz gegen Tropfen mit einem Fallwinkel von 15°
3	Drahtkontakt unmöglich (2.5mm Objekte)	3	Schutz gegen Regenfall bis zu 60°
4	Feiner Drahtkontakt unmöglich (1.0mm Objekte)	4	Schutz gegen Sprühwasser von allen Seiten
5	Schutz gegen schädlichen Staub	5	Schutz gegen Wasserstrahlen von allen Seiten
6	Komplett staubgeschützt	6	Schutz gegen Wasserfluten (bei schwerem Seegang)
		7	Schutz gegen Wassereinwirkung bei 1m Tiefe
		8	Schutz gegen lange Wassereinwirkung bei > 1m Tiefe

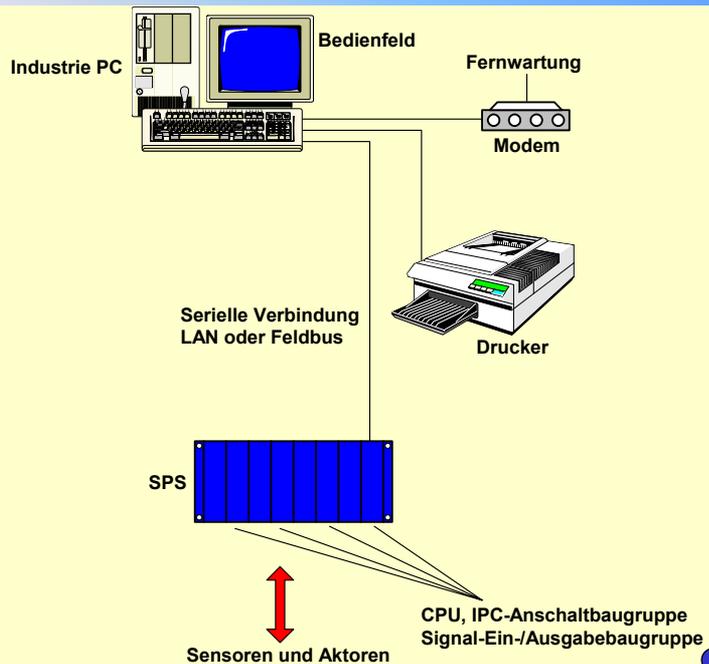
□ **Einsatzgebiete von Industrie-PCs**

- * **Prozess-Visualisierung**
- * **Prozessauswertung und -überwachung**
- * **übergeordnete Steuerungsaufgaben (Leitandaufgaben)**

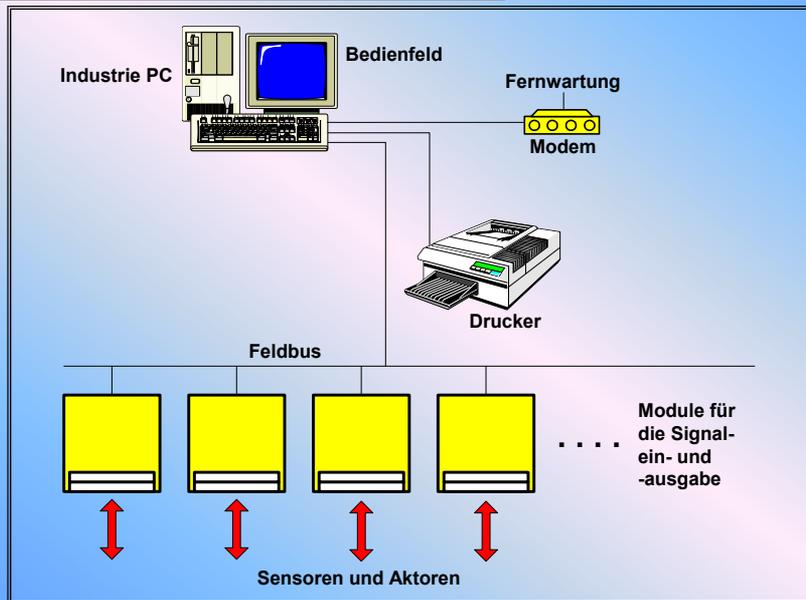
□ **Kombination von Industrie-PCs und SPS-Systemen**

- * **Industrie-PC Schnittstelle zum Bediener**
- * **SPS zum Einlesen/ Ausgeben der Prozess-Signale und Abarbeitung der Steuerungsalgorithmen**

Anordnung eines SPS-IPC-Systems



Anordnung eines reinen IPC-Systems



☐ Aufgaben bei einer reinen IPC-Lösung

- * Erfassung von Prozessgrößen
- * Steuerungsfunktion unter Echtzeit
- * Schnittstelle zum Bediener

☐ Vorteile einer reinen IPC-Lösung

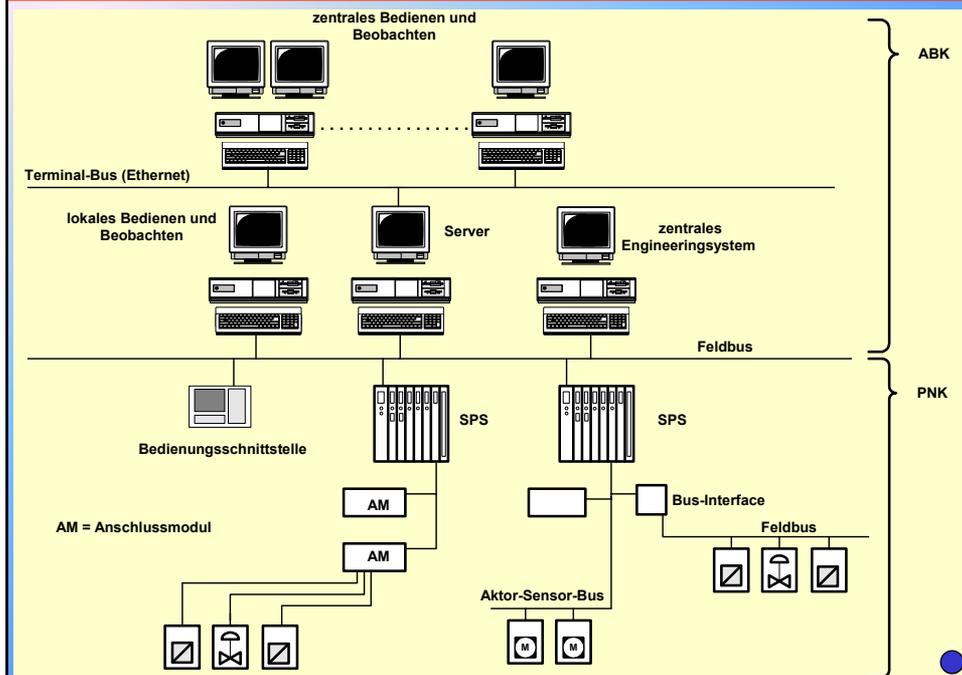
- * bessere Skalierbarkeit der Hardware
- * grosses Angebot an Betriebssystemen
- * grosses Spektrum an Programmiersprachen
- * offenes System zur Integration von fertigen Teillösungen

Prozessleitsysteme

- **Komplettlösungen von einem Hersteller**
 - * keine Kompatibilitätsprobleme
 - * einheitliche Bedienung und Beobachtung des Prozesses
 - hohe Verfügbarkeit
 - * definierte Verantwortlichkeit
 - * lange Lebenszeit

- **Anwendungsgebiete von Prozessleitsystemen**
 - * Kraftwerkstechnik
 - * Verfahrenstechnik
 - * Gebäudetechnik
 - * Fertigungstechnik

- **Bestandteile eines PLS**
 - * Anzeige- und Bedienkomponente (ABK)
 - * Prozessnahe Komponenten (PNK)
 - * Systemkommunikation
 - * Engineering-Tool



Anzeige- und Bedienkomponente

□ Funktionen

- * Rezepte/ Batch Abläufe erstellen/ modifizieren
- * aktuelle Werte ändern
- * Kommunikation mit dem Prozess
- * Alarmer/ Bedienanforderungen bearbeiten
- * Prozessvisualisierung
- * Schnittstelle zu Datenbanksystemen zur Prozessprotokollierung

□ Realisierung

- * PC, IPC, Workstation
- * Windows 95, Windows NT, Unix

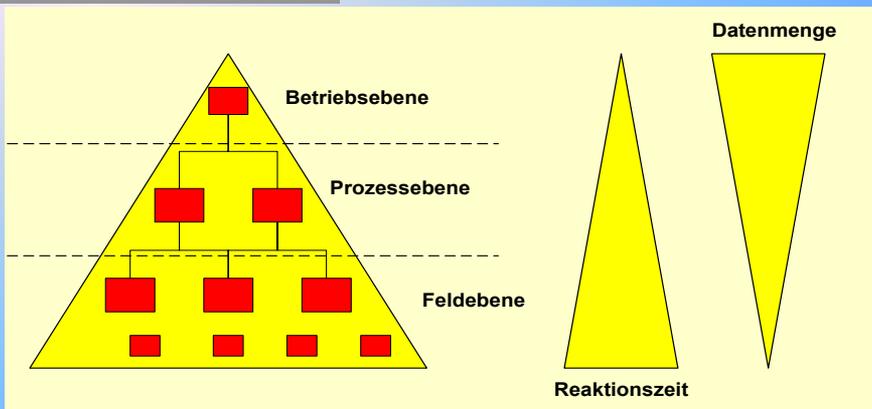
Prozessnahe Komponente

□ Eigenentwicklungen der Leitsystem-Hersteller

□ Aufbau

- * Automatisierungscomputer: SPS, IPC
- * dezentrale Peripherie
- * Feldgeräte: Sensoren, Aktoren

Systemkommunikation



Bus-Systeme

- ABK: standardisierter Ethernet-Bus
- PNK: Feldbus, z.B. Profibus, H1-Bus, Modbus, Interbus-S
- unterste Feldebene: schnelle Aktor-Sensor-Busse

Engineering

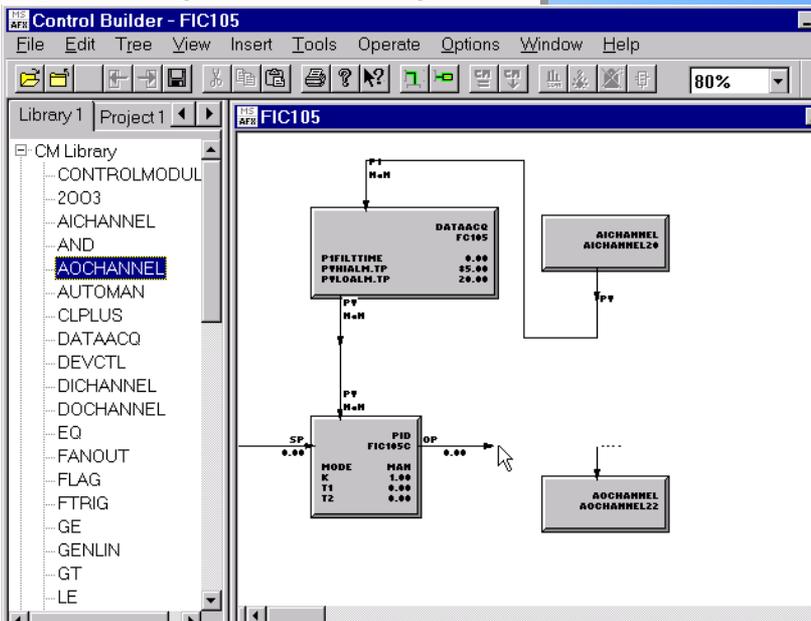
Aufgaben

- * Konfiguration
- * Programmierung
- * Wartung/Pflege

Werkzeuge

- * graphische Werkzeuge (IEC 1131)
- * standardisierte Bibliotheken mit Komponenten
- * mächtige Editoren

Beispiel einer grafischen Konfiguration



Leitsystem-hersteller und deren Produkte

Hersteller	Produktname	Bemerkung
ABB	AdvantOCS	Firmenspezifischer Feldbus
	AdvaSoft	für kleine Anlagen
	Procontrol P	Kraftwerkstechnik
EB Hartmann & Braun	Symphony	Unterstützt Anbindung an Betriebsebene, Remote I/O System
	Contronic E	Kraftwerkstechnik, für große Anlagen
	Contronic P	Verfahrenstechnik, Ausdehnung bis zu 12km
Foxboro-Eckardt	I/A Serie-System	Verfahrenstechnik, PNK mit PCMCIA-Technologie
Siemens	SIMATIC PCS 7	Verfahrenstechnik, ABK auf Basis von Windows 95 und Windows NT. Umfangreiches Hardwareangebot. Feldbus: Profibus. Anbindung an Betriebsebene möglich.
	Teleperm M	Verfahrenstechnik, weitverbreitet. Altes Bussystem (CS 275). Migration von Teleperm M nach SIMATIC PCS 7 möglich.
	Teleperm XP	Kraftwerkstechnik, offene Kommunikation, umfangreiches Hardwareangebot.
Honeywell	PlantScape	Verfahrenstechnik, offenes System, ABK auf Basis von Windows NT, unterstützt Remote I/O. Anbindung an Betriebsebene möglich.
	TDC 3000	Verfahrenstechnik, besitzt mehrere Prozessbusse mit unterschiedlicher Datenübertragung. MODBUS wird unterstützt.

2.3 Zentrale und dezentrale Automatisierungsstrukturen

Arten von Automatisierungsstrukturen

Struktur des technischen Prozesses

- * technischer Prozess als Einheit
Bsp.: Bohrvorgang bei einer Bohrmaschine
- * technischer Prozess, der aus Teilprozessen besteht
Bsp.: Fertigung eines Getriebes

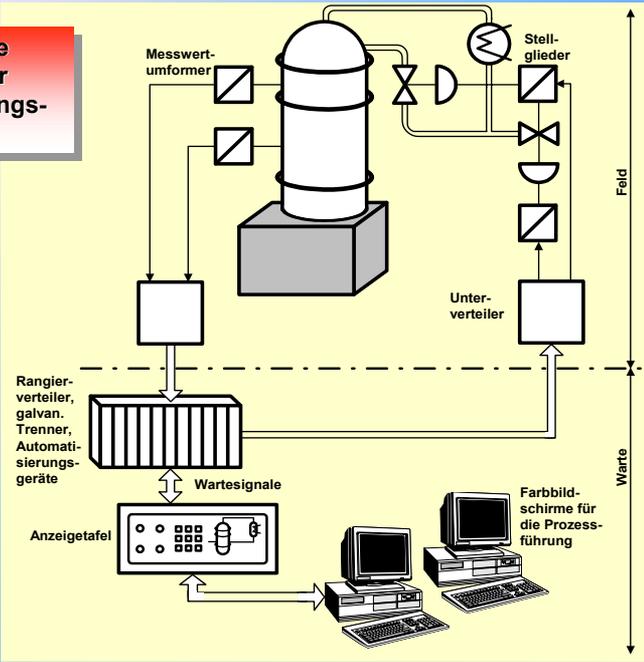
Struktur der Automatisierungsgeräte

- * örtlich zentrale Anordnung
- * örtlich dezentrale Anordnung

Wirkungsmäßige Struktur des Automatisierungssystems (funktionelle Struktur)

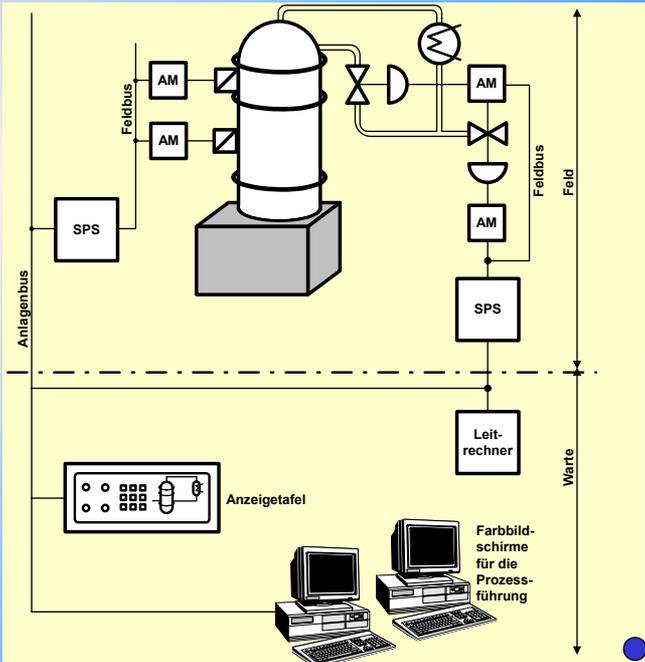
- = Aufteilung der Automatisierungsfunktionen auf die Automatisierungsgeräte
- * wirkungsmäßig zentrale Grundstruktur
- * wirkungsmäßig dezentrale Grundstruktur

Örtlich zentrale Anordnung der Automatisierungsgeräte

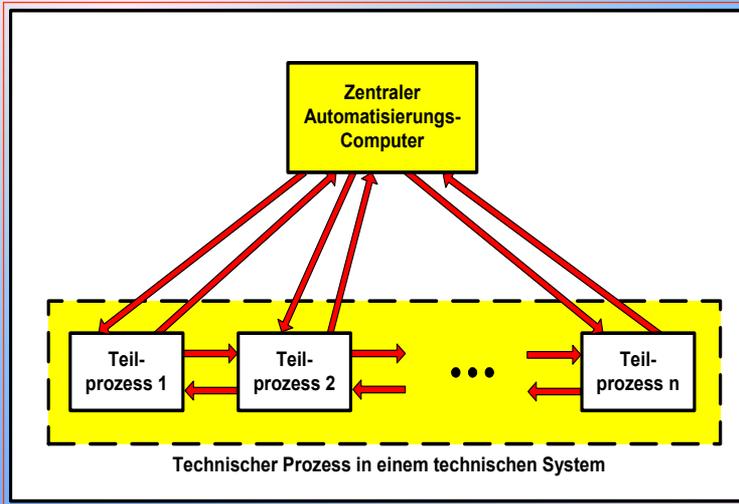


Örtlich dezentrale Anordnung der Automatisierungsgeräte

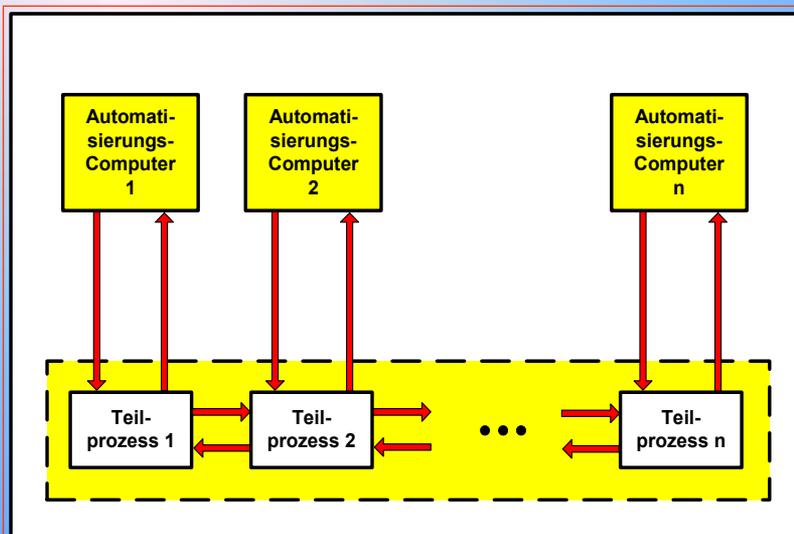
- AM: Anschluss-Modul für Sensoren bzw. Aktoren
- FBK: Feldbus-Koppler
- FEP: Front-End-Prozessrechner
- PBK: Prozessbus-Koppler



Wirkungsmäßig zentrale Automatisierungsstruktur



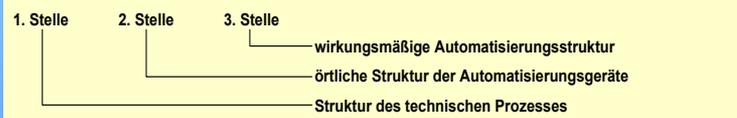
Wirkungsmäßig dezentrale Automatisierungsstruktur



Unterschiedliche Kombination der Automatisierungsstrukturen

		Technischer Prozess als eine Einheit betrachtet (zentrale Prozess-Struktur)		Technischer Prozess in Teilprozesse gegliedert (dezentrale Prozess-Struktur)	
		örtlich zentrale Anordnung der Automatisierungsgeräte	örtlich dezentrale Anordnung der Automatisierungsgeräte	örtlich zentrale Anordnung der Automatisierungsgeräte	örtlich dezentrale Anordnung der Automatisierungsgeräte
wirkungsmäßige Struktur	wirkungsmäßig zentrale Automatisierungsstruktur	ZZZ	ZDZ	DZZ	DDZ
	wirkungsmäßig dezentrale Automatisierungsstruktur	ZZD	ZDD	DZD	DDD

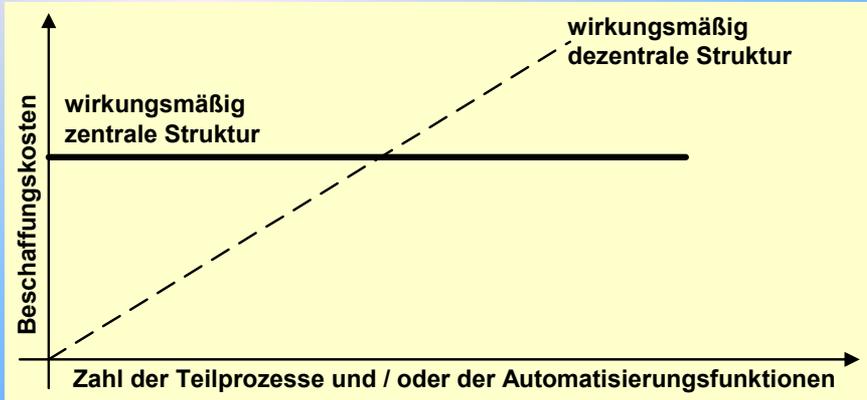
Kennzeichnung der verschiedenen Strukturen: Z = zentral / D = dezentral



Kriterien für einen Vergleich der Eigenschaften von Automatisierungsstrukturen

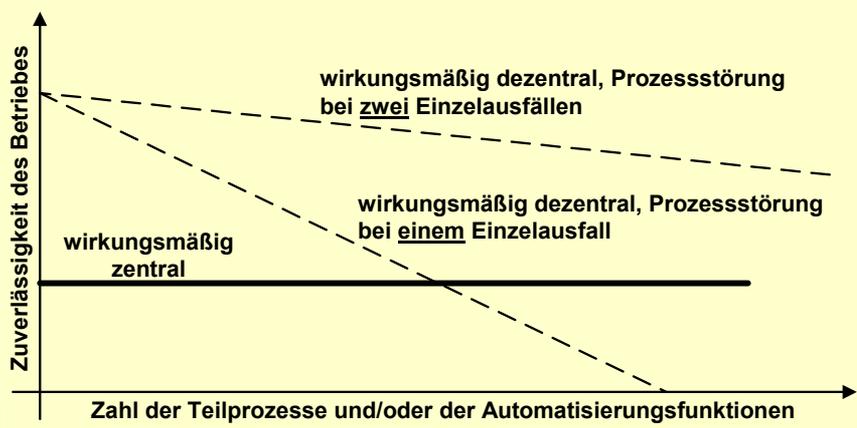
- * die Kosten für die Geräte, die Verkabelung, die Software, die Pflege und die Wartung
- * die Teilverfügbarkeit bei Hardware-Ausfällen oder bei Software-Fehlern
- * die Flexibilität bei Änderungen
- * die Koordinierung der Teilprozesse und die Optimierung des Gesamtprozesses
- * die Bedienbarkeit

Vergleich der Beschaffungskosten



kaum zusätzliche Kosten bei Erweiterung

Vergleich der Zuverlässigkeit des Betriebes des technischen Prozesses



ein Einzelausfall führt zu Prozessstörung, Zuverlässigkeit konstant

zwei Einzelausfälle führen zur Prozessstörung

Im Normalfall Verkopplung schwach

- ↓ kein Totalausfall bei Ausfall von zwei oder mehr Automatisierungseinheiten
- ↓ Zuverlässigkeit des Betriebes bei einer wirkungsmäßig dezentralen Struktur höher als beim Einsatz eines zentralen Prozessrechners

Dezentrale Struktur

- + Flexibilität bei Änderungen
- + Koordination der Teilprozesse
- + Optimierung des Gesamtprozesses
- zusätzlicher Aufwand zur Kommunikation der einzelnen Automatisierungseinheiten
- 0 Bedienbarkeit und Benutzerfreundlichkeit
- + Störfall-Lokalisierung
- + höhere Transparenz

Bewertung von Automatisierungsstrukturen hinsichtlich der Kriterien

Z = zentrale Struktur
D = dezentrale Struktur

ZZZ	Typisch für Automatisierung kleiner Geräte
ZDZ	Geringere Verkabelungskosten als bei ZZZ
DZZ	Ungünstig bzgl. Verfügbarkeit, Wartbarkeit, Verkabelungskosten
DDZ	Ungünstig bzgl. Verfügbarkeit und Flexibilität
ZZD	Günstig bzgl. Wartbarkeit und Flexibilität Ungünstig: Verkabelung
ZDD	Günstig: Flexibilität, Verfügbarkeit, Verkabelung, Transparenz
DZD	Günstig: Verfügbarkeit, Wartbarkeit Ungünstig: Verkabelungskosten
DDD	Günstig: Flexibilität, Verfügbarkeit, Verkabelung, Transparenz

Produkt-
automatisierung

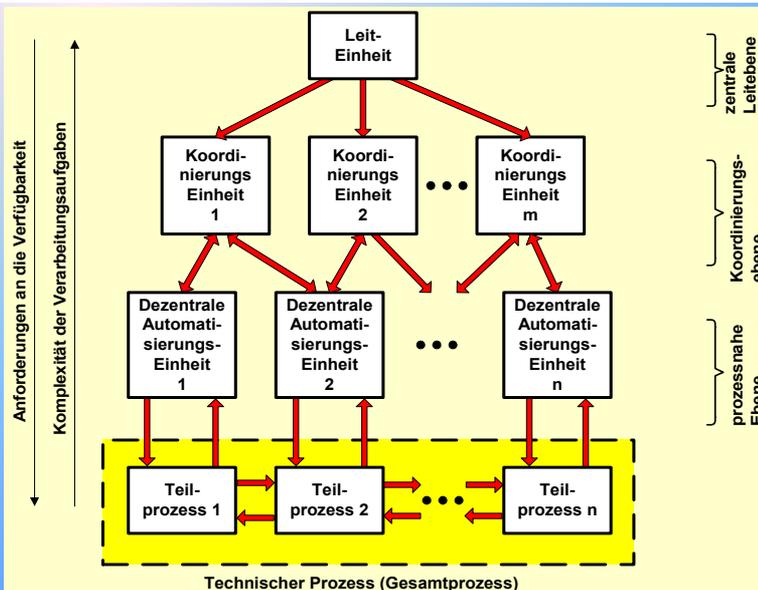
Anlagen-
automatisierung

Kfz-
Elektronik

wirkungsmäßige Struktur
örtliche Struktur der Automatisierungsgeräte
Struktur des technischen Prozesses

⇒ so dezentral wie möglich, so zentral wie nötig

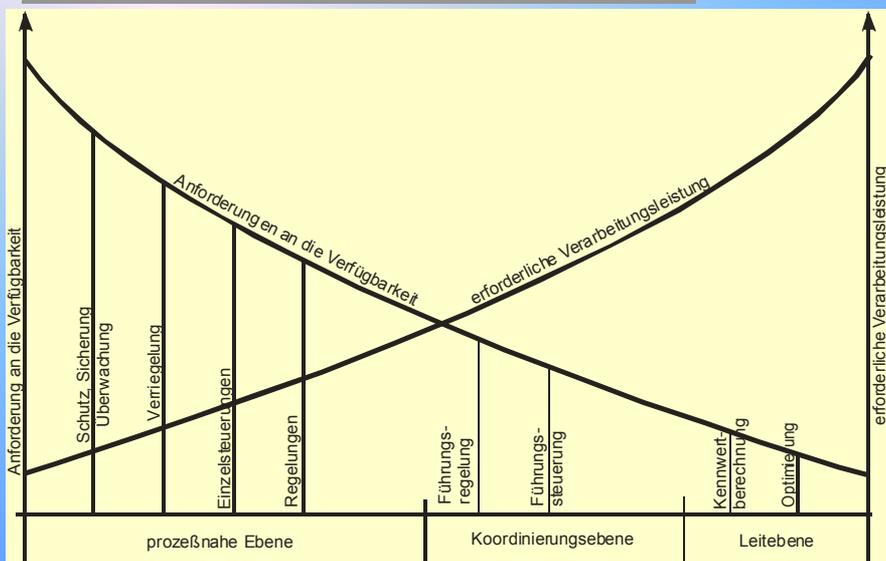
Kombination von zentraler und dezentraler Struktur durch Einführung einer Hierarchie von Automatisierungseinheiten



Zuordnung von Automatisierungsfunktionen zu den Prozessführungsebenen

- * Dezentrale Automatisierungseinheiten erfüllen die Aufgaben der Prozessführungsebene 4, hohe Anforderungen an die Verfügbarkeit
- * Koordinierungs-Einheiten realisieren die Automatisierungsfunktionen der Prozessführungsebene 3 wie Koordinierung der Teilprozesse, Optimierung, Prozessüberwachung und -sicherung
- * Leiteinheit realisiert die Aufgaben der beiden obersten Prozessführungsebenen

Anforderungen an die Hierarchie-Ebenen bezüglich Verfügbarkeit und Verarbeitungsleistung

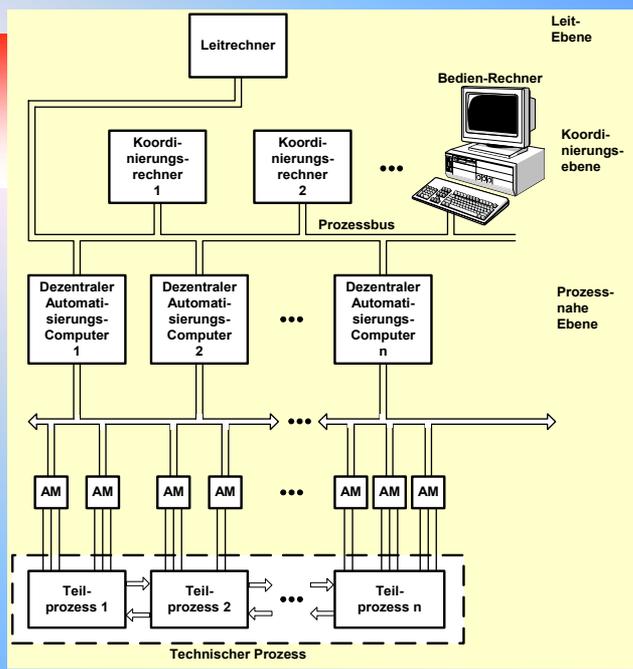


Voraussetzung für die Realisierung einer Automatisierungshierarchie

- * Gliederung des technischen Prozesses in Teilprozesse
- * Einsatz von intelligenten Automatisierungseinheiten
- * Kommunikationssystem zwischen Automatisierungseinheiten

Je nach Größe des Unternehmens und Umfang des technischen Prozesses können Zwischenebenen eingeführt werden bzw. Ebenen zusammengezogen werden.

Realisierung einer Automatisierungshierarchie mit einem busorientierten verteilten Prozessrechnensystem



2.5 Verteilte Automatisierungssysteme (distributed systems)

- Verknüpfung von dezentralen Automatisierungseinheiten mit übergeordnetem Rechner über ein Kommunikationssystem

keine Hierarchie

Unterschied zu Automatisierungs-Hierarchie

- Einheiten kommunizieren dort nur mit nächst höherer Ebene
- Nur prozessnahe Informationsaufgaben werden dort dezentral wahrgenommen

Zielsetzung bei verteilten Automatisierungssystemen

- Hohe Zuverlässigkeit durch Fehlertoleranz
 - * Ausfall eines dezentralen Computers
 - * führt nicht zu Gesamtausfall
 - * Eingrenzung des Fehlers durch Rekonfiguration
- Erhöhung der Verfügbarkeit durch schnelle Wartung und Instandsetzung
 - * gegenseitige Überwachung mit Fehlerdiagnose
- Gegenseitige Aushilfe bei hoher Belastung
 - * selbsttätige Anpassung der Aufgabenverteilung
 - * Reduzierung der Reserven der einzelnen Einheiten
- Einfache Erweiterungsfähigkeit

Grundstrukturen der Kommunikation bei verteilten Automatisierungssystemen

Wahl eines Kommunikationssystems

- niedrige Verkabelungskosten
- standardisierte Schnittstellen bezüglich
 - * Stecker, Leitungen (mechanisch)
 - * Spannungspegel (elektrisch)
 - * Übertragungsprotokoll (logisch)
- Flexibilität bei Änderungen
- geringe Anforderungen an die Kommunikationspartner
 - * Speicherplatzbedarf
 - * Rechenzeitaufwand
- hohe Verfügbarkeit und Zuverlässigkeit
- sichere Übertragung der Informationen
 - * Verwendung von Prüfbits
 - * Bestätigung des korrekten Empfangs

Wesentliche Kosten: mehr Kabel bedeutet mehr Störungen auf den Leitungen

- Realisierung hoher Datenübertragungsraten
- kurze Reaktionszeiten auf Übertragungsanforderungen
- Kopplung unterschiedlicher Kommunikationspartner

Festlegung der Prioritäten bei der Realisierung der sich zum Teil widersprechenden Einzelkriterien

Bsp.: hohe Verfügbarkeit und Zuverlässigkeit durch redundantes Bus-System bedeutet hohe Verkabelungskosten

Grundstrukturen der Kommunikation

- * **Stern-Struktur**

Ausfall der Zentraleinheit bedeutet Ausfall der Kommunikation

- * **Ring-Struktur**

Jede Einheit kann nur an direkte Nachbarn übertragen werden

- * **Netz-Struktur**

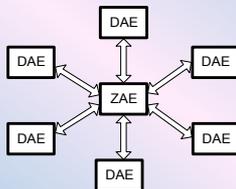
Parallele Informationsübertragung, kurze Reaktionszeit, viele Schnittstellen, hohe Verkabelungskosten

- * **Bus-Struktur**

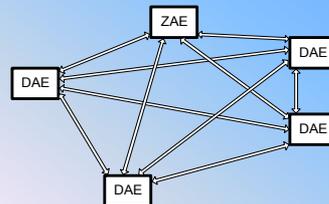
nur jeweils ein Teilnehmer kann senden, gleichzeitige Informationsaufnahme von allen Teilnehmern

Grundstrukturen der Kommunikation bei verteilten Automatisierungssystemen

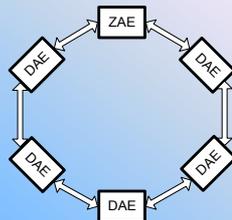
a) Stern-Struktur



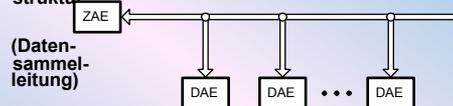
c) Netz-Struktur



b) Ring-Struktur



d) Bus-Struktur



ZAE = zentrale Automatisierungseinheit
 DAE = dezentrale Automatisierungseinheit

Bus-Systeme

Parallele Busse

- Adressen, Daten und Steuersignale werden parallel übertragen
- Leitungsbündel

Distanz: bis 20 m

Serielle Busse

- Bits einer Nachricht zeitlich nacheinander
- Übertragungszeit länger als bei Parallelbus
- niedrigere Leitungskosten
- Erhöhung der Zuverlässigkeit
- Flexibilität bezüglich Leitungsprotokollen

Distanz: 20 m bis 15 km

Arten von Kommunikationssystemen

- Offenes Kommunikationssystem**
OSI = Open-System-Interconnection
- Herstellerspezifisches Kommunikationssystem**
CSI = Closed-System-Interconnection

ISO/ OSI - Basisreferenzmodell

ISO IS 7498



Aufgaben der ISO/OSI-Schichten

Schicht 1:	Physikalische Ebene (Bitübertragungsschicht, Physical Layer)	<ul style="list-style-type: none"> - Übertragungsmedium - Codierungsart - Schnittstelle - Topologie
Schicht 2:	Verbindungsebene (Sicherungsschicht, Data Link Layer)	<ul style="list-style-type: none"> - Fehlerbehandlung/-erkennung - Zugriffsverfahren - Synchronisation
Schicht 3:	Netzebene (Vermittlungsschicht, Network Layer)	<ul style="list-style-type: none"> - Netzprotokolle - Datenadressierung - Datenvermittlung - Wegwahl
Schicht 4:	Transportebene (Transport Layer)	<ul style="list-style-type: none"> - Wegparallelisierung - Paketwiederholung - Paketsortierung
Schicht 5:	Sitzungsebene (Kommunikationssteuer-ebene, Session Layer)	<ul style="list-style-type: none"> - Eröffnung - Überwachung - Beendigung der Kommunikation
Schicht 6:	Darstellungsschicht (Presentation Layer)	<ul style="list-style-type: none"> - Sprachanpassung (z.B. zwischen ASCII & EBCDIC) - Datenverschlüsselung - Datenentschlüsselung
Schicht 7:	Anwendungsschicht (Verarbeitungsschicht, Application Layer)	<ul style="list-style-type: none"> - Grundvorrat an Diensten - Standardanwendungen - anwendungsspezifisch

2.6 Automatisierungsstrukturen mit Redundanz

Formen der Redundanz

Prozessbedienpersonal überwacht parallel zu einem Prozessrechner

- ☐ Hardware-Redundanz
 - * redundante Hardware
- ☐ Software-Redundanz
 - * redundante Software
- ☐ Messwert-Redundanz
 - * redundante Messgrößen
 - * abhängige Messgrößen
- ☐ Zeit-Redundanz
 - * mehrfache Abfrage des gleichen Messwertes in bestimmten Zeitabständen

Hardware- und Software-Redundanz bedeutet höherer Aufwand

- * erhöhte Verfügbarkeit
- * sicherheitsrelevante Systeme

Fehlertolerante Strukturen

Ziel:

Systeme so zu konstruieren, dass sie nach dem Auftreten von Fehlern in einzelnen Komponenten als Ganzes funktionsfähig sind.

Stufen der Fehlertoleranz

⇒ volle Fehlertoleranz

fail operational

⇒ verringerte Leistungsfähigkeit

fail soft, graceful degradation

⇒ Übergang in einen sicheren Zustand

fail safe

Prinzip der Fehlertoleranz

Aufbau eines Systems aus redundanten Modulen (Hardware- und Software), um bei Auftreten eines Fehlers die Funktionsfähigkeit des Systems zu erhalten.

Arten von Redundanz

- ⇒ Statische Redundanz
 - * alle redundanten Module ständig im Einsatz
- ⇒ Dynamische Redundanz
 - * redundante Module werden erst nach einem Ausfall eingesetzt
 - * blinde Redundanz
 - redundante Module sind im fehlerfreien Fall nicht tätig
 - * funktionsbeteiligte Redundanz
 - redundante Module führen im fehlerfreien Fall Stand-by-Funktionen durch

Hardware-Redundanz

Ziel: *Erkennung von Ausfällen der Hardware*

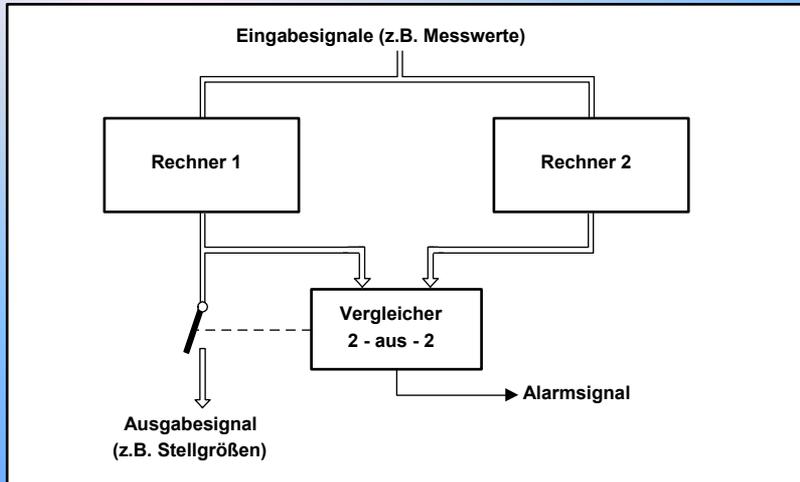
Einsatzprinzip:

- ⇒ m-von-n-Redundanz
 - * Mehrheitsentscheid
 - * Fehler erst bei Mehrfachdefekten

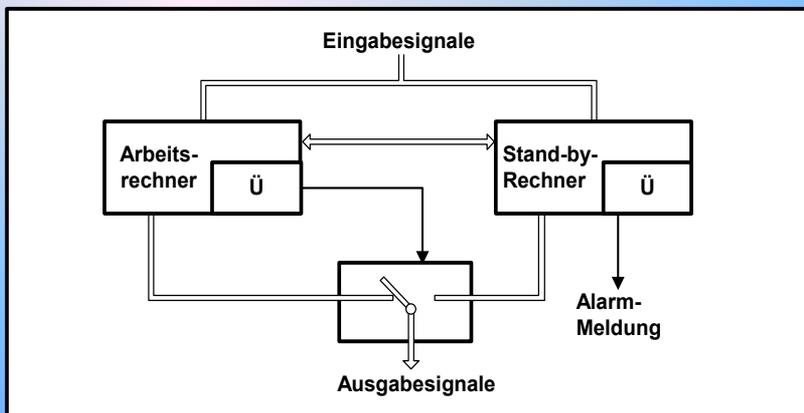
Realisierung der Redundanz

- ⇒ Doppel-Rechner-Strukturen
- ⇒ Drei-Rechner-Strukturen

Doppelrechnerstruktur mit statischer Redundanz

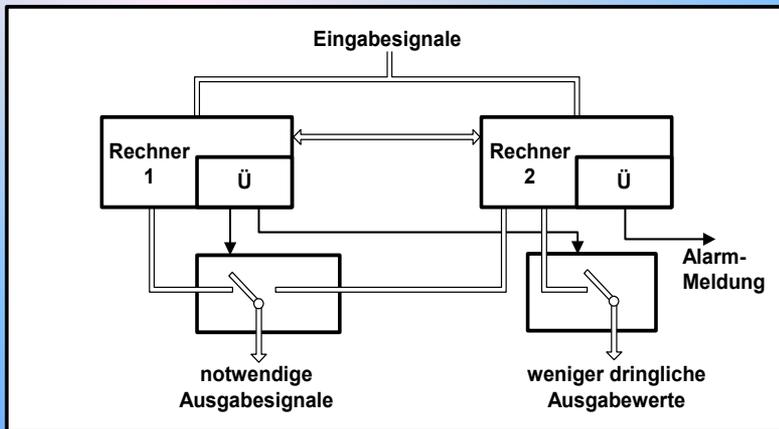


Doppelrechner-Struktur mit dynamisch blinder Redundanz



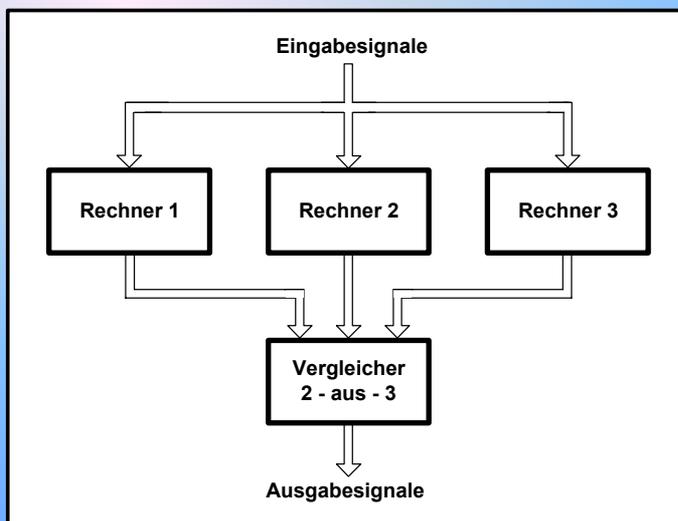
Ü = Überwachungsprogramm

Doppelrechnerstruktur mit dynamisch funktionsbeteiligter Redundanz



Ü = Überwachungsprogramm

Drei-Rechner-Struktur mit statistischer Redundanz



Software-Redundanz

Ziel: *Erkennung von Fehlern in der Software*

Ausgangspunkt:

Software ist fehlerhaft

Redundanzmaßnahmen bei Software

- * verschiedenartiger Aufbau von Programmteilen
- * gleiche Eingangsdaten müssen gleiche Ergebnisse liefern

Verschiedenartigkeit von Software bei gleicher Funktion (Diversität)

- * Unabhängige Entwicklerteams lösen dieselbe Aufgabe
- * Gezielte Entwicklung verschiedener Strategien, Algorithmen und Software-Strukturen

Einsatz bzw. Ausführung diversitärer Software-Teile

- * Redundante Software-Alternativen werden nacheinander ausgeführt und über Entscheider (Voter) verglichen, nicht für Realzeitsysteme mit hohen Zeitanforderungen
- * Parallele Ausführung redundanter Software-Teile auf redundantem Mehrrechnersystem
- * Zyklische Abwechslung der diversitären Teile

Vergleich schwierig

- Zwei Algorithmen mit unterschiedlichen Verarbeitungszeiten
- Beide Ergebnisse können korrekt sein, obwohl Werte unterschiedlich sind