
Mohieddine Jelali

Prozessautomatisierungstechnik

2. Komponenten und Strukturen von Automatisierungssystemen

Inhalt der Veranstaltung – Aktualisierung 1

	Thema
1	Einführung in die Prozessautomatisierung 1.1 Grundbegriffe 1.2 Fachliche Teilgebiete der Automatisierungstechnik 1.3 Aufbau von Automatisierungssystemen 1.4 Automatisierungsgrad und Rechnereinsatzarten 1.5 Ziele der Automatisierung 1.6 Historische Entwicklung 1.7 Einsatzgebiete der Automatisierung 1.8 Ebenen-Modelle der Automatisierung 1.9 Funktionen und Aufgaben der Automatisierung 1.10 Graphische Darstellung in der Verfahrenstechnik 1.11 Beispiele für Prozessautomatisierungssysteme 1.12 Lösungsweg für Automatisierungsaufgaben 1.13 Auswirkungen der Automatisierung auf Menschen)
2	Komponenten und Strukturen von Automatisierungssystemen 2.1 Automatisierungsrechner 2.2 Sensoren 2.3 Aktoren 2.4 Ein- und Ausgabengeräte 2.5 Automatisierungsstrukturen 2.6 Redundanz und Fehlertolerante Strukturen

Inhalt der Veranstaltung – Aktualisierung 1

	Thema
3	Kommunikationsnetzwerke 3.1 Netztopologien 3.3 Übertragungsmedien 3.3 Feldbussysteme 3.4 Buszugriffsverfahren 3.5 Wichtige Feldbussysteme
4	Echtzeitsysteme und Echtzeitprogrammierung 4.1 Echtzeitsysteme 4.2 Aufgaben von Echtzeitsystemen 4.3 Echtzeitsysteme – Beispiele 4.4 Anforderungen an Echtzeitsysteme 4.5 Echtzeit-Programmierverfahren 4.6 Synchronisierung von Tasks 4.7 Synchronisierungsverfahren 4.8 Scheduling-Verfahren
...	...

Inhaltsangaben zu Kapitel 2

Komponenten und Strukturen von Automatisierungssystemen

- 2.1 Automatisierungsrechner
- 2.2 Sensoren
- 2.3 Aktoren
- 2.4 Ein- und Ausgabengeräte
- 2.5 Automatisierungsstrukturen
- 2.6 Redundanz und Fehlertolerante Strukturen

2.1 Automatisierungsrechner

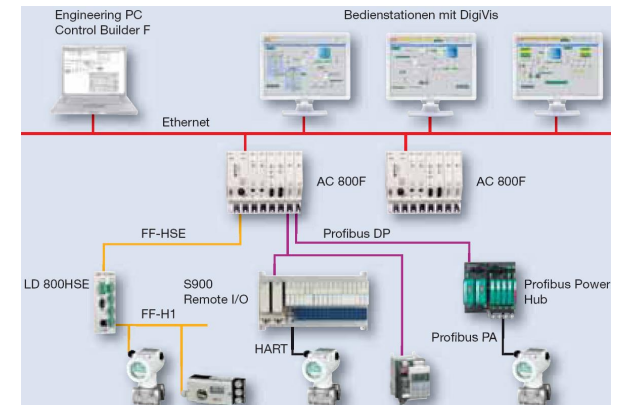
Arten von Automatisierungsrechnern

- Speicherprogrammierbare Steuerungen (SPS)

- Microcontroller (μC)

- Personal Computer (PC) bzw.
Industrial Personal Computer (IPC)

- Prozessleitsysteme (PLS)



2.1 Automatisierungsrechner

■ Speicherprogrammierbare Steuerungen: SPS

■ Komponenten von SPS

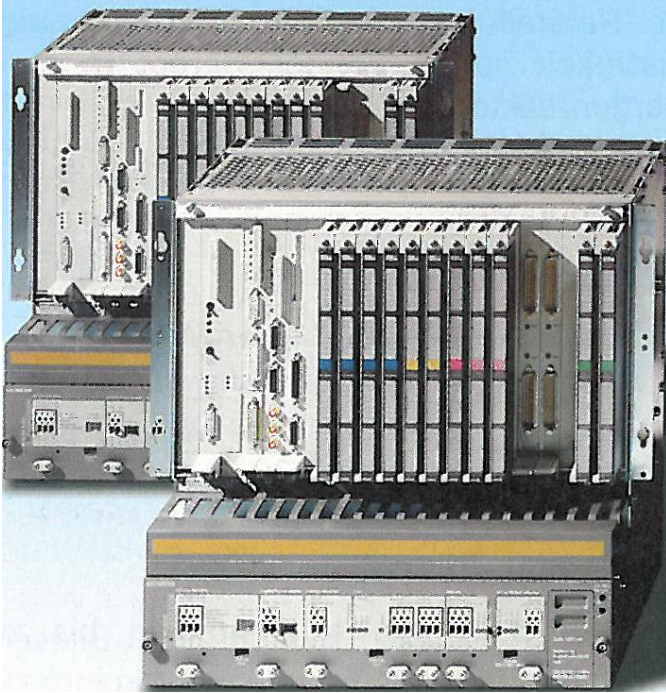
- Hardware: Elektronische Steuerung mit einer festen internen Verdrahtung unabhängig von der Steuerungsaufgabe
- Software: Programm zur Realisierung des gewünschte Ablaufes je nach Aufgabe einer zu steuernde Maschine oder Anlage; Programm frei programmierbar in einem Programmspeicher
- E/A-Geräte: Feste Verdrahtung

■ Entwicklungsgeschichte der SPS

- Richard Morley (Modicon) und Odo J. Struger (Allen Bradley): Väter der SPS
- 1974: Erste SPS in Deutschland auf den Markt (Firma Klaschka)
- 1979: andere deutsche Hersteller mit ihren Modellen auf den Markt
- Heute: mehr als 300 Hersteller auf dem europäischen Markt
- Prominente Vertreter: Simatic-Familie der Fa. Siemens S5/S7

2.1 Automatisierungsrechner

■ Typische Varianten von SPS



Legendäre Variante (1995): **SIMATIC S5**

Moderne Variante (2005): **SIMATIC S7**



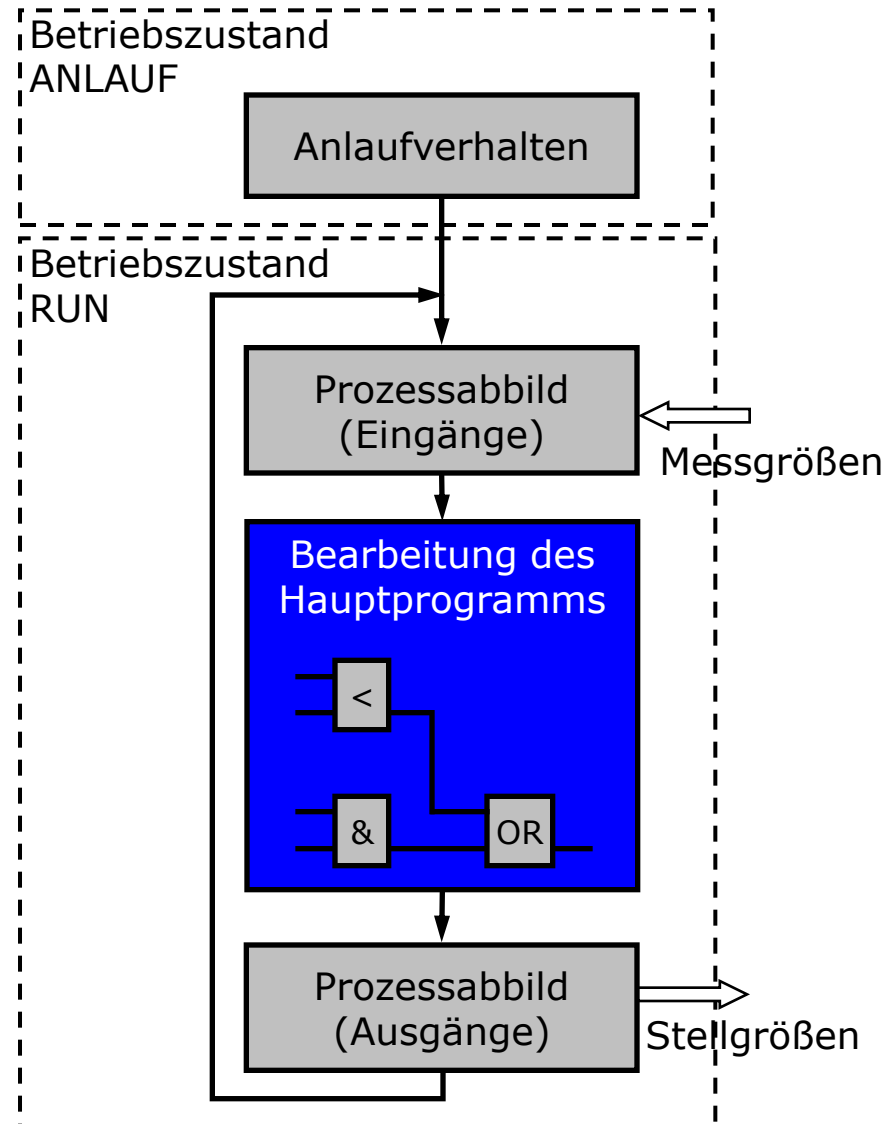
2.1 Automatisierungsrechner

Zyklische SPS-Programmbearbeitung

- Vor jedem Zyklus: Laden der Ein- und Ausgangssignale in Speicher (sog. Prozessabbild)
- Während eines Zyklus: Einlesen der gespeicherten Werte des Prozessabbildes + Ausgabe des Prozessabbildes an die Aktoren

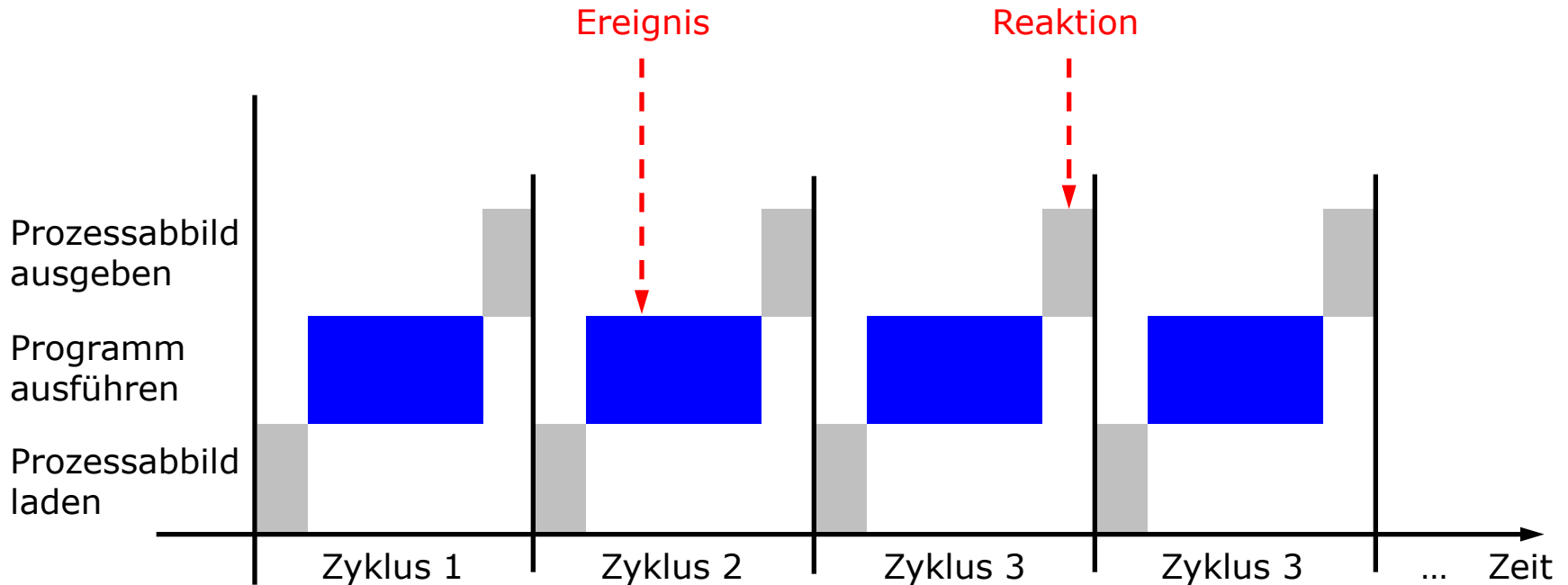
Zusatzfunktionen für SPS

- Überwachung
- Fehlerbehandlung
- Diagnose



2.1 Automatisierungsrechner

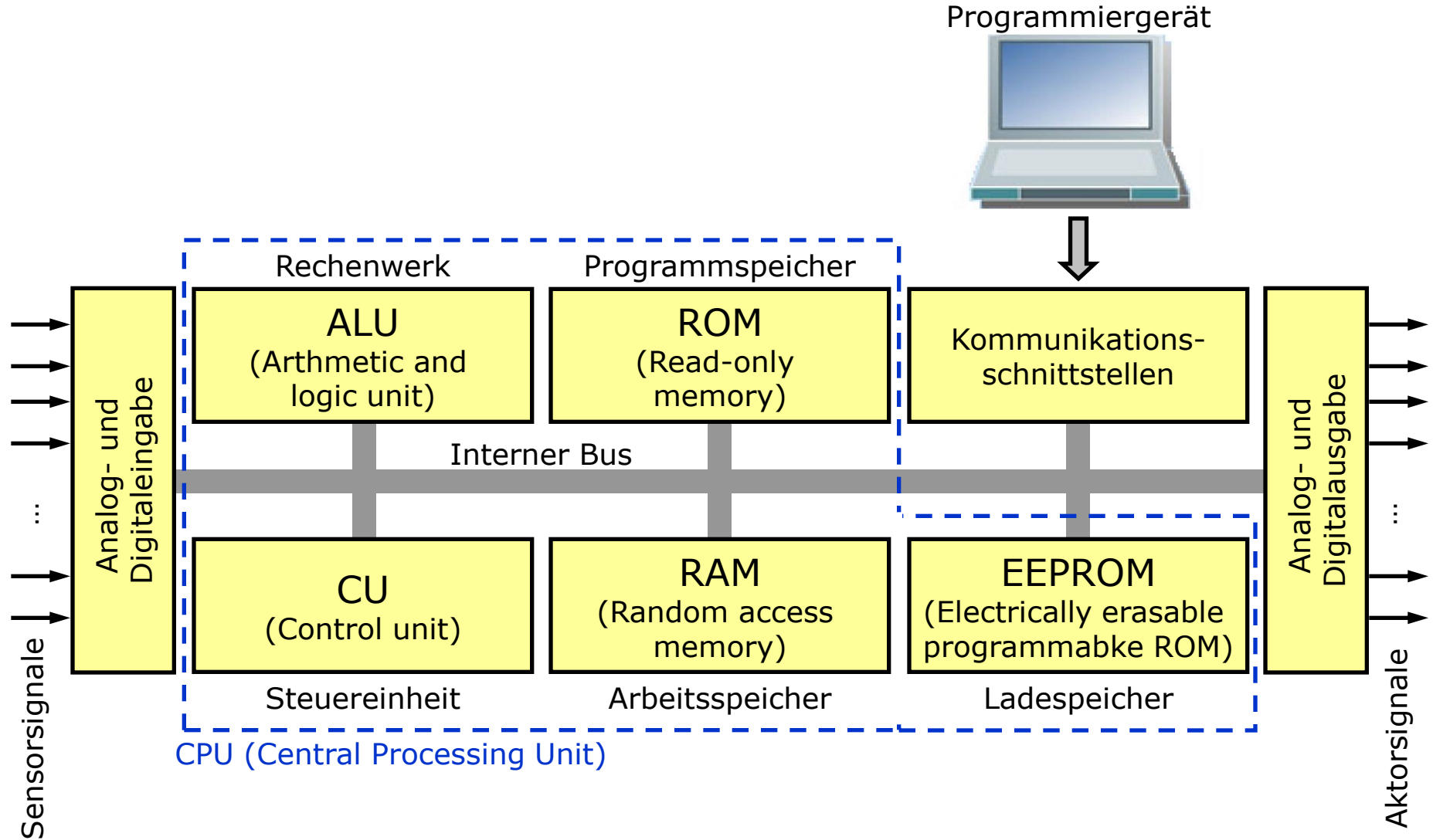
Zyklische SPS-Programmbearbeitung



- Vorteile: Vermeidung von Inkonsistenzen bei sich ändernden Prozesszuständen; einfache Programmierung
- Nachteil: Späte Reaktion auf Ereignisse nach bis zu mehr als zwei Programmzyklen (worst case) = $2 \cdot \text{Zykluszeit} + \text{Verzögerungszeit des Eingangs} + \text{Reaktionszeit des Ausgangs}$

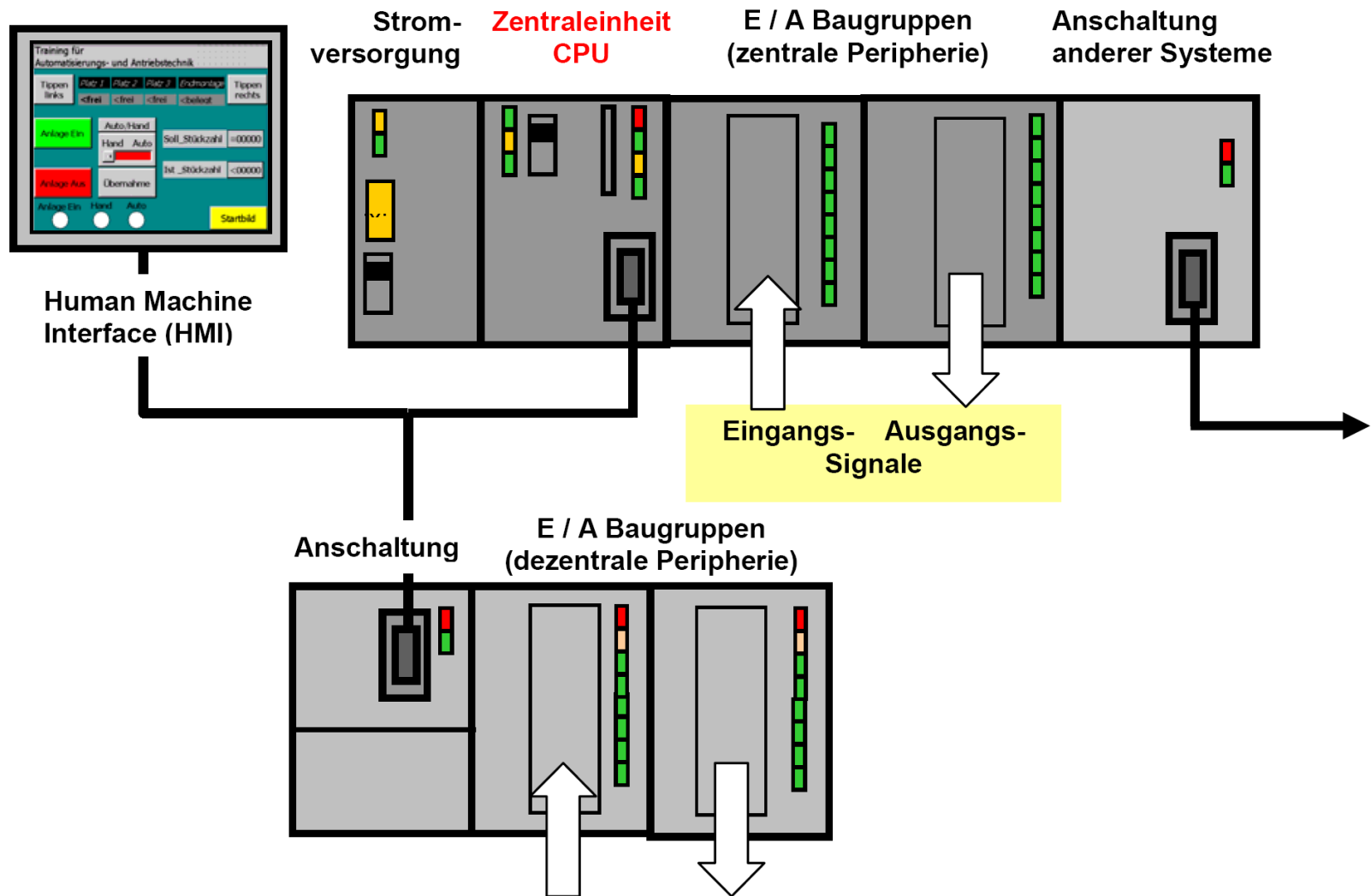
2.1 Automatisierungsrechner

■ Prinzipieller Aufbau einer SPS



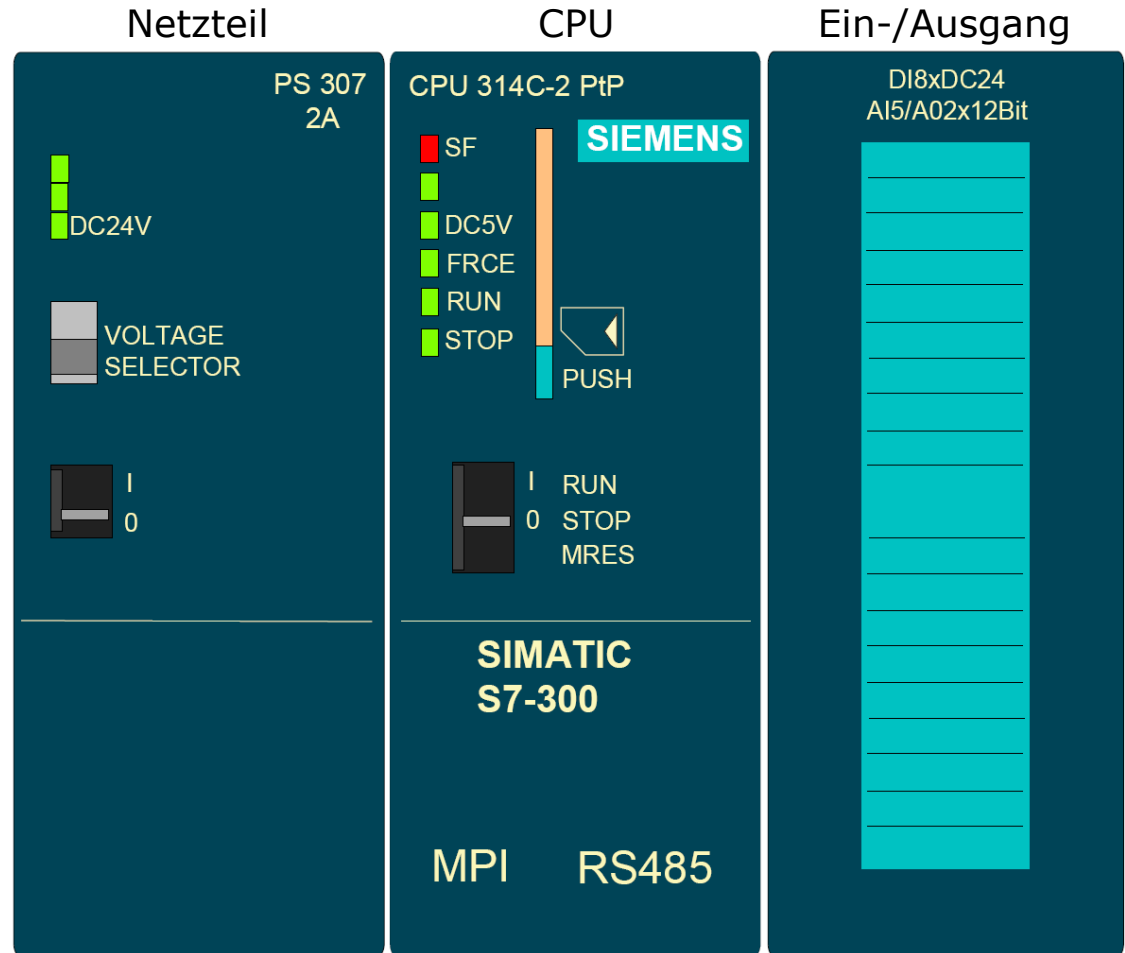
2.1 Automatisierungsrechner

■ Grundsätzliche Struktur der Automatisierungstechnik mit SPS



2.1 Automatisierungsrechner

Aufbau einer SIMATIC S7



MPI-Adapter

2.1 Automatisierungsrechner

■ Wesentliche Vorteile von SPS

- Verschleißfreiheit
- Kleine Baumaße
- Steuerfunktionen sind schnell und einfach änderbar
- Vereinfachte Fehlerdiagnose
- Zuverlässigkeit
- Große Leistungsfähigkeit
- Langfristige kostengünstig

■ Nachteile von SPS

- Zusätzliche Infrastruktur notwendig (Programmiergeräte bzw. Software, Datensicherung usw.)
- Qualifiziertes Personal erforderlich

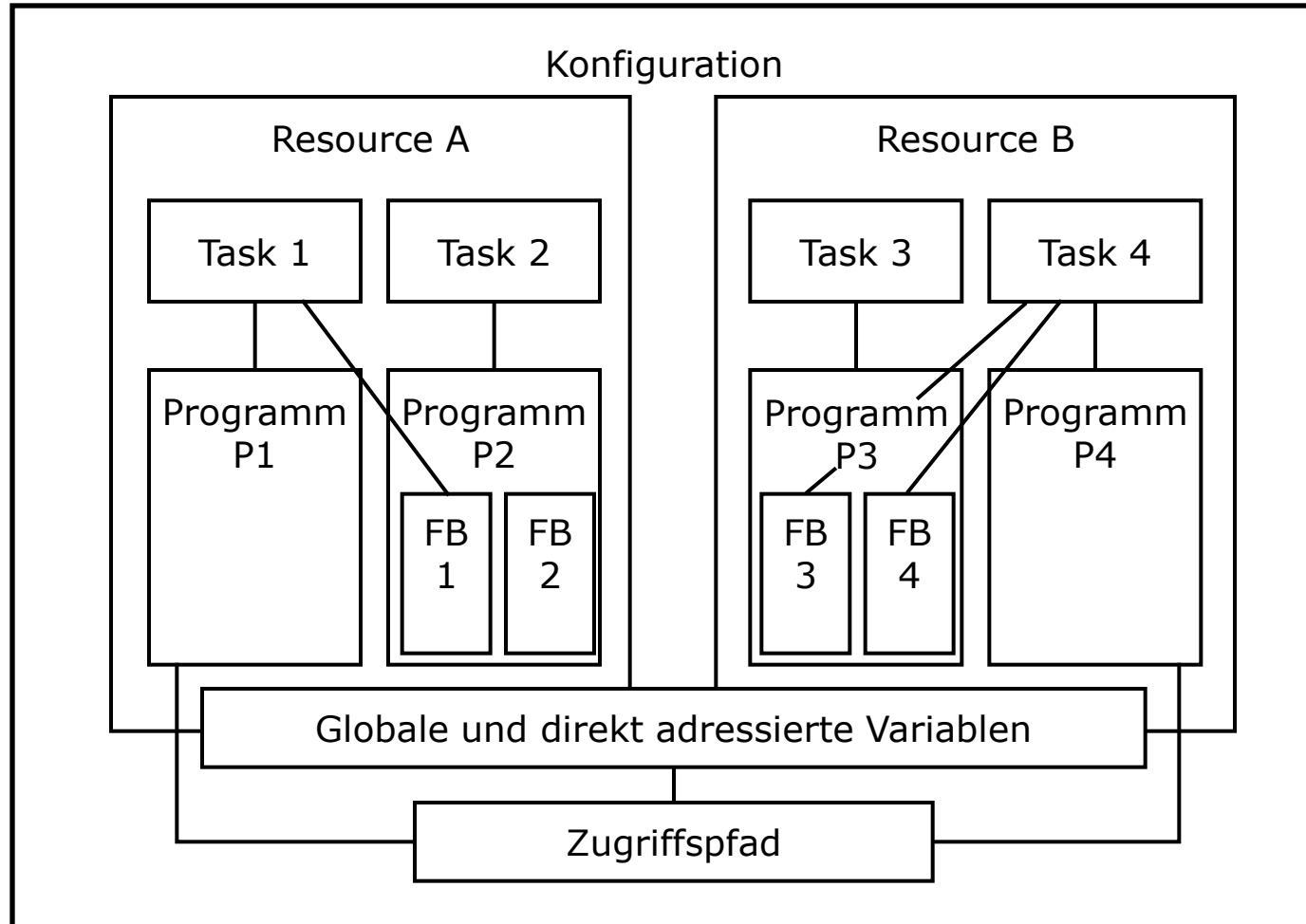
■ Programmierung von SPS

Strukturparadigma der IEC 61131

(Konfigurationselemente, Programmorganisationseinheiten, Vorschriften zur Definition von Datentypen und Variablen, Programmiersprachen)

2.1 Automatisierungsrechner

■ Konfigurationselemente von SPS-Systemen nach IEC 61131



2.1 Automatisierungsrechner

■ SPS-Programmierung / Anweisungsliste (AWL) - Instruction List (IL)

Zeilen- und textorientierte Sprache; assemblerähnlich

0001	CAL	Tastendruck_lang(IN:=T_Ein,PT:=T#1s)	
0002	LD	Tastendruck_lang.Q	
0003	JMPC	M1	(*Bei langem Tastendruck Nullen der Listeneinträge bei Label M1*)
0004			
0005	CAL	posFI_INILochl(CLK:=INI_Lochleiste)	
0006	CAL	negFI_INILochl(CLK:=INI_Lochleiste)	
0007			
0008	LD	Antr_rechts	
0009	AND	(
0010	LD	posFI_INILochl.Q	
0011	OR	negFI_IniLochl.Q	
0012)		
0013	JMPC	M2	(* Bei Flankenimpuls Eintrag der analogen Positionen bei Label M2 *)
0014	RET		
0015	M2:		
0016	LD	Wegeinheiten	(*Eintrag von Positionen*)
0017	ST	Liste[Index]	
0018			

2.1 Automatisierungsrechner

■ SPS-Programmierung / Strukturierter Text (ST) - Structured Text (ST)

Rein textorientierte Sprache

```
(*Steuersignal Antrieb Rechtslauf bei Funktion 2*)
Antrieb_re(SET:=T_Start AND Hand_Auto AND Fkt2,RESET1:=INI_rechts OR NOT T_Aus);
Signal_Antr_re:= Antrieb_re.Q1;

(*Pause rechts*)
Pause_re(IN:=INI_rechts,PT:=T#2s);

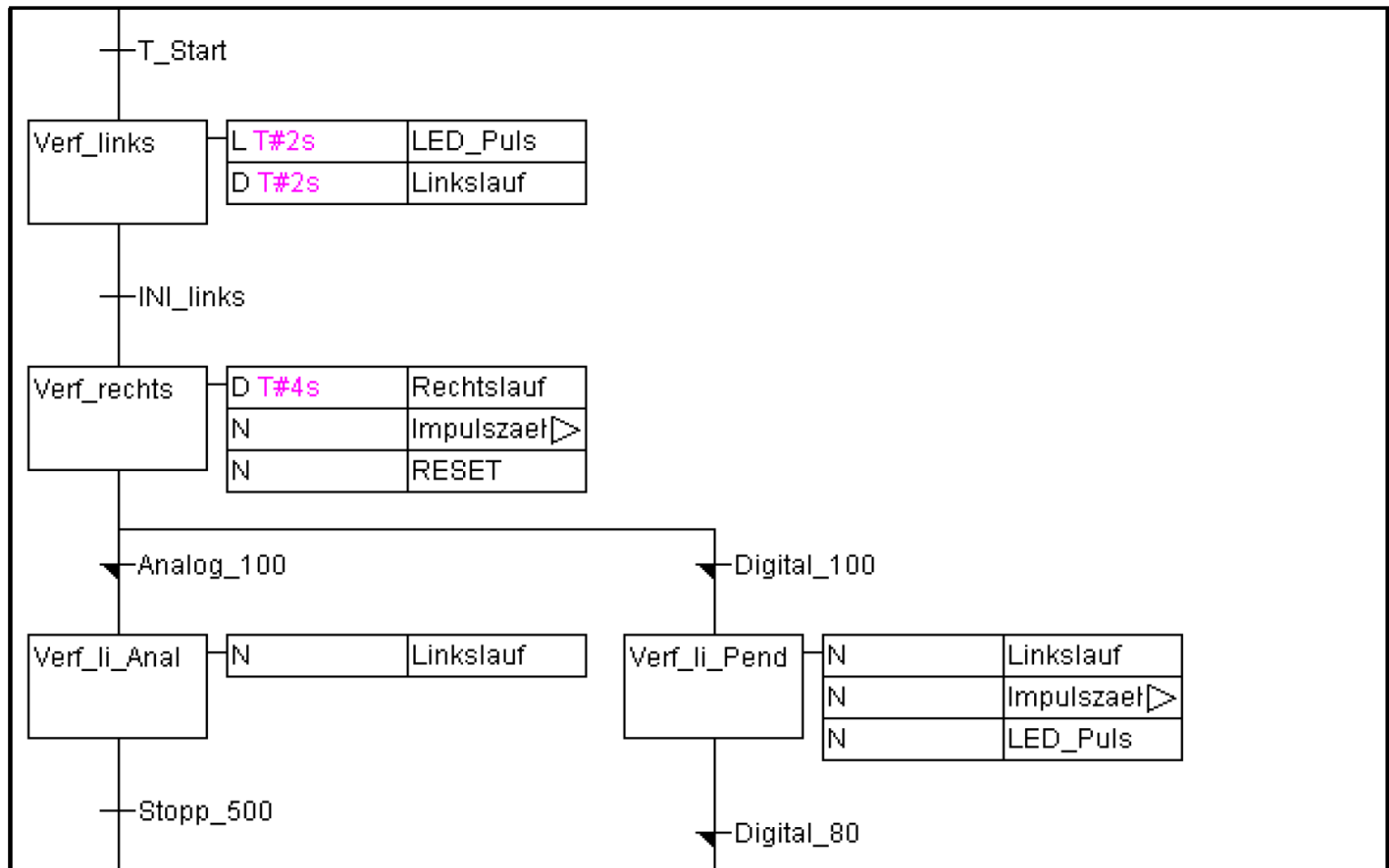
(*Steuersignal Antrieb Linkslauf bei Funktion 2*)
Antrieb_li(SET:=Pause_re.Q,RESET1:=Flankenzaehler.Q OR NOT T_Aus);
Signal_Antr_li:= Antrieb_li.Q1;

(*Flankenzähler zur Auswertung der Lochleiste*)
posFI_1(CLK:=INI_Lochleiste AND Signal_Antr_li);
Flankenzaehler(CU:=posFI_1.Q,RESET:=T_Start,PV:=3);
```

2.1 Automatisierungsrechner

■ SPS-Programmierung / Ablaufsprache (AS) - Sequential Function Chart (SFC)

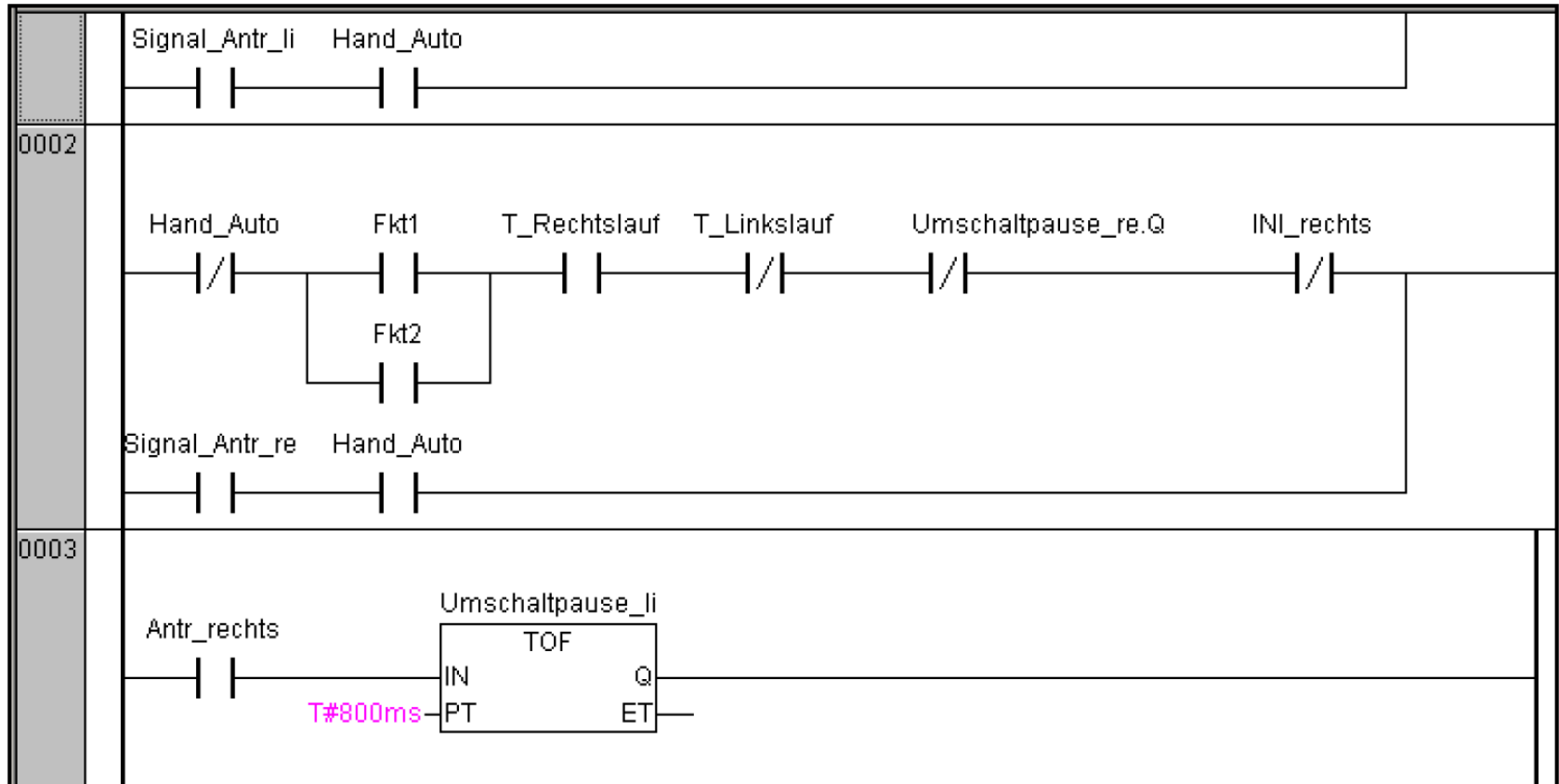
Zwei Varianten: textbasiert oder rein graphisch



2.1 Automatisierungsrechner

■ SPS-Programmierung / Kontaktplan - Ladder Diagram (LD)

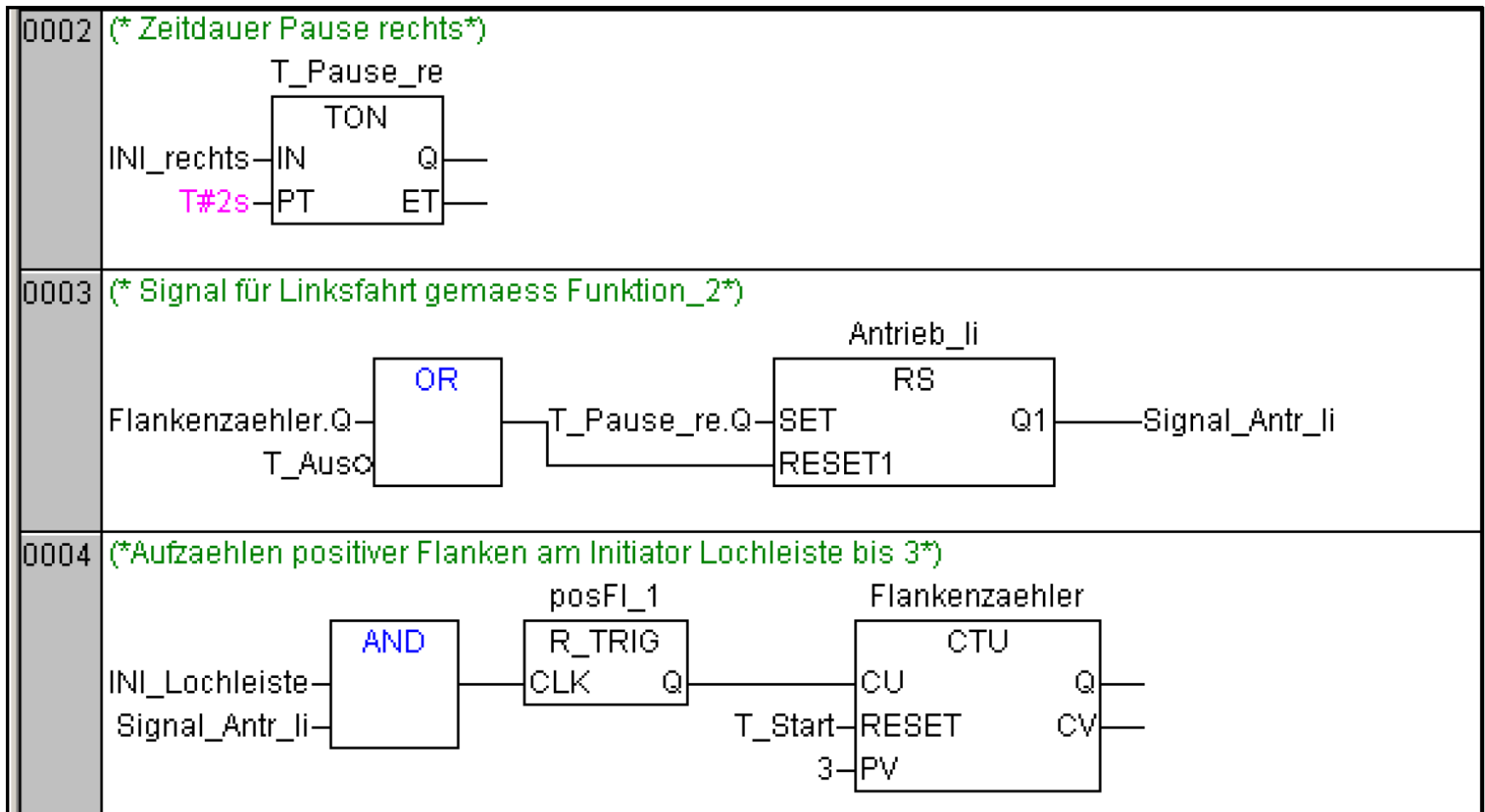
Graphische Sprache in Anlehnung am Stromlaufplan kontaktbehafteter Steuerungstechnik („Leiter-Diagramm“)



2.1 Automatisierungsrechner

■ SPS-Programmierung / Funktionsplan - Function Block Diagram (FBD)

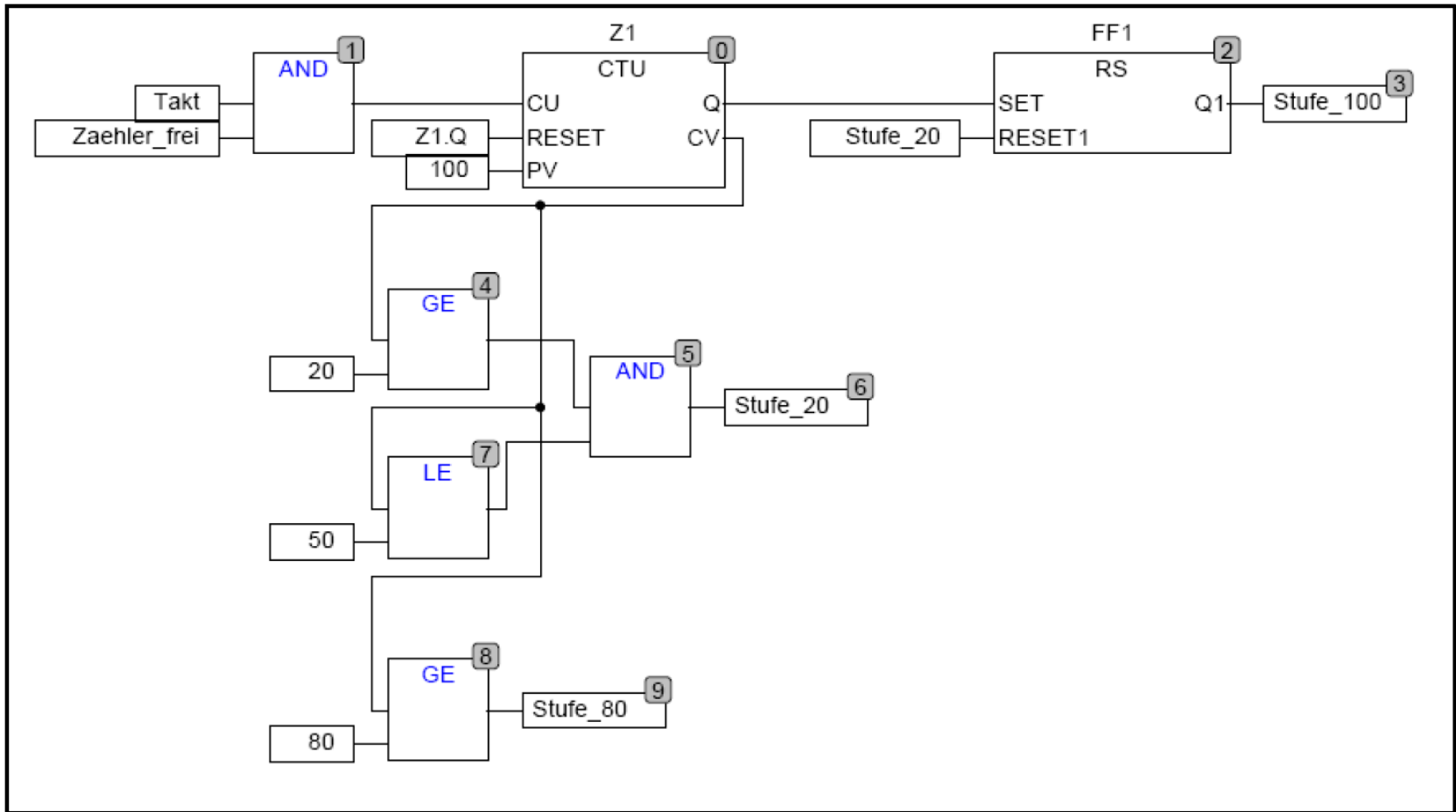
Graphische Sprache: bekannte Bausteine und Symbole von Logikplänen



2.1 Automatisierungsrechner

■ SPS-Programmierung / Freigraphischer Funktionsplan - Continuous Function Chart (CFC)

Graphische Sprache mit beliebig platzierbaren Bausteinen und Wirkungslinien:



2.1 Automatisierungsrechner

■ Mikrocontroller (μC)

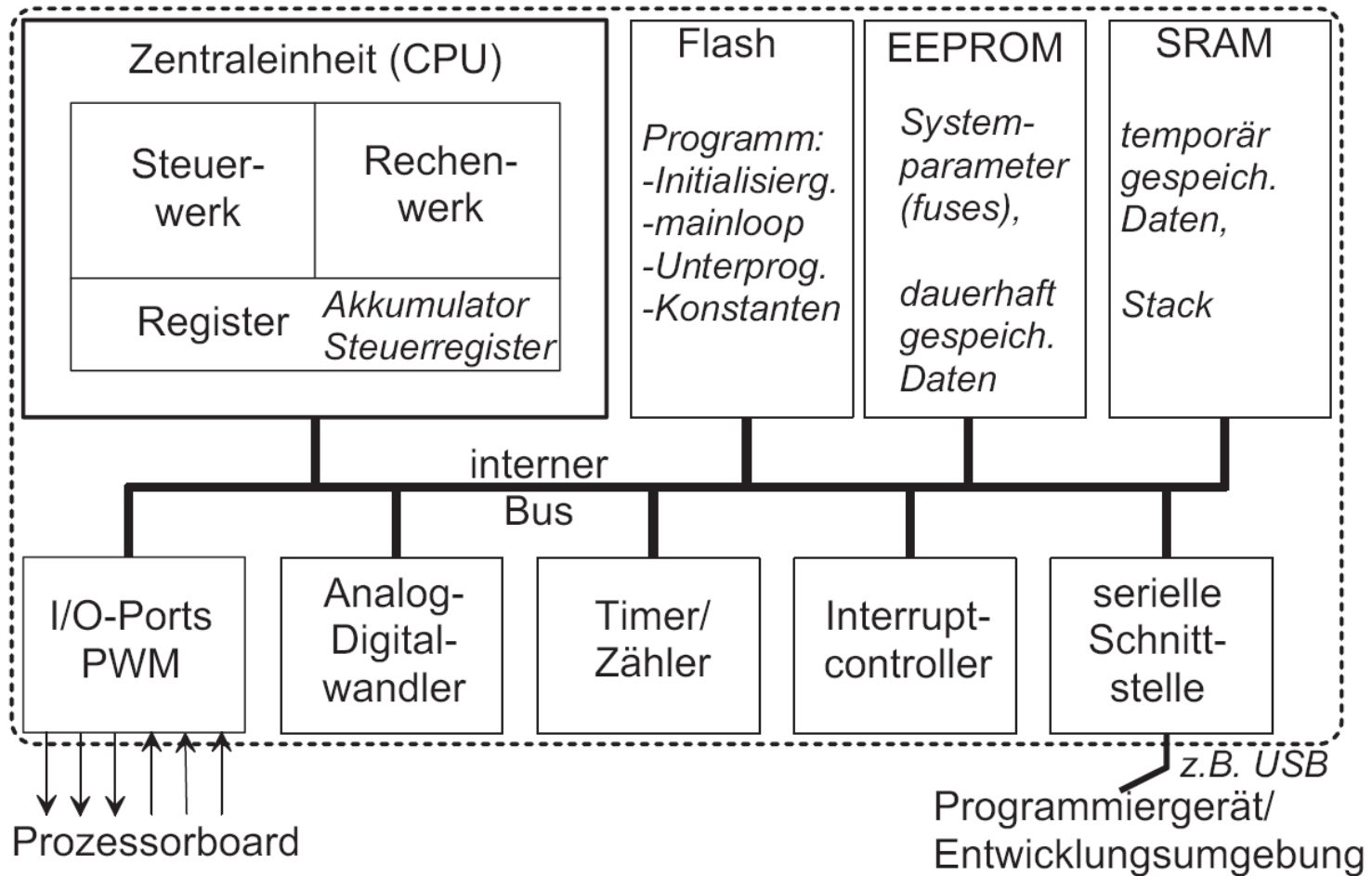
Automatisierungsrechner bzw. komplettes Automatisierungsrechnersystem auf einem Chip zum Einbau in ein Produkt (embedded control computer)

■ Merkmale von Mikrocontrollern

- kurze Wortlänge
- Integration von Peripheriegeräten, z.B. ADU, auf dem Chip
- extrem niedriger Preis (1–10€) / Hoher Stückzahl
- hohe Anforderungen bzgl. Temperaturbereich, Feuchte, aggressive Umgebung usw.
- hohe Zuverlässigkeit und Lebensdauer
- Programmierung noch heute vielfach in Assembler (4-bit und 8-bit Mikrocontroller)
- Bei steigender Komplexität der Programme: Programmierung zunehmend in Hochsprachen, z.B. C (16-bit Mikrocontroller)

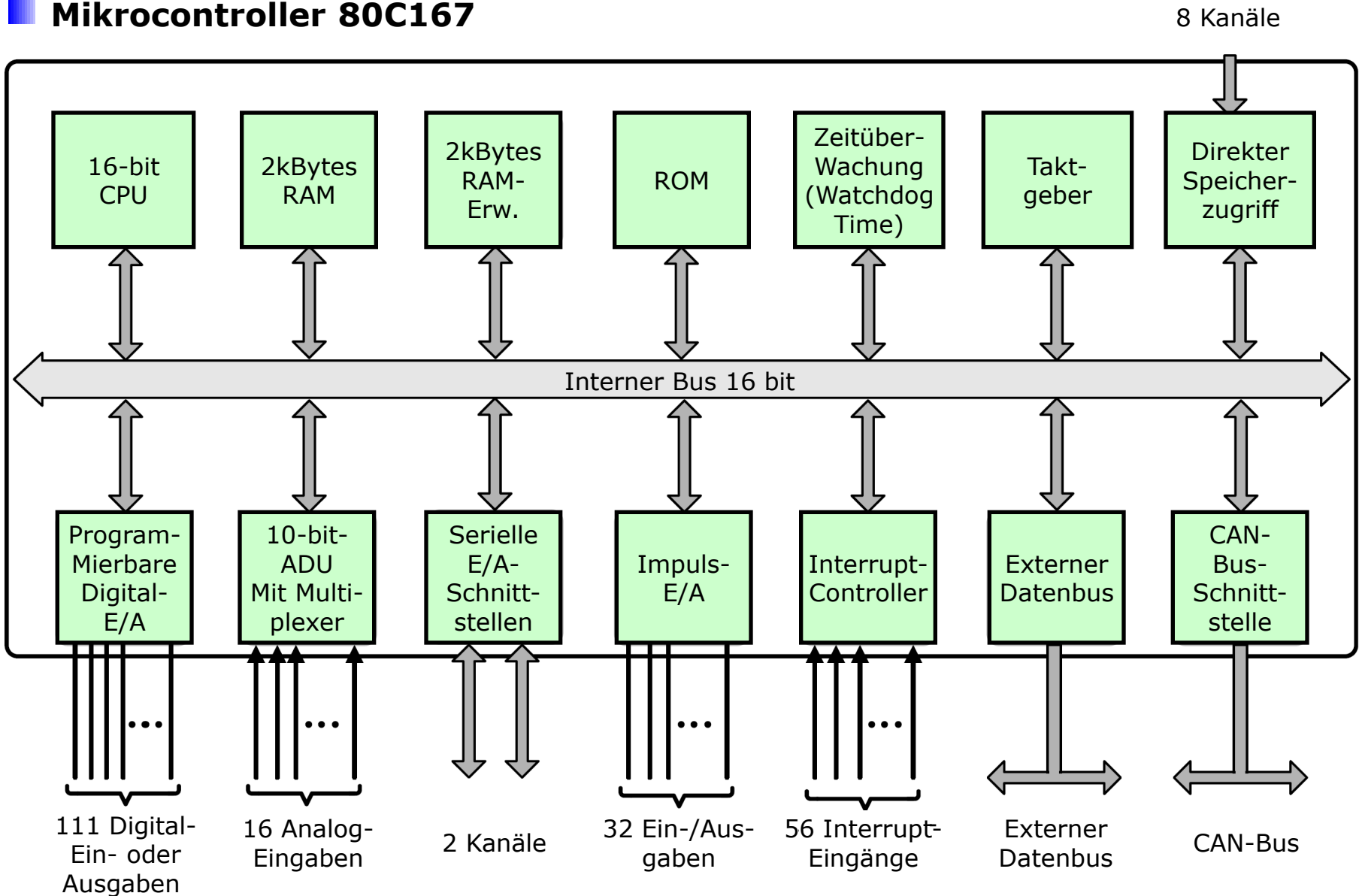
2.1 Automatisierungsrechner

Komponenten eines RISC-Mikrocontrollers



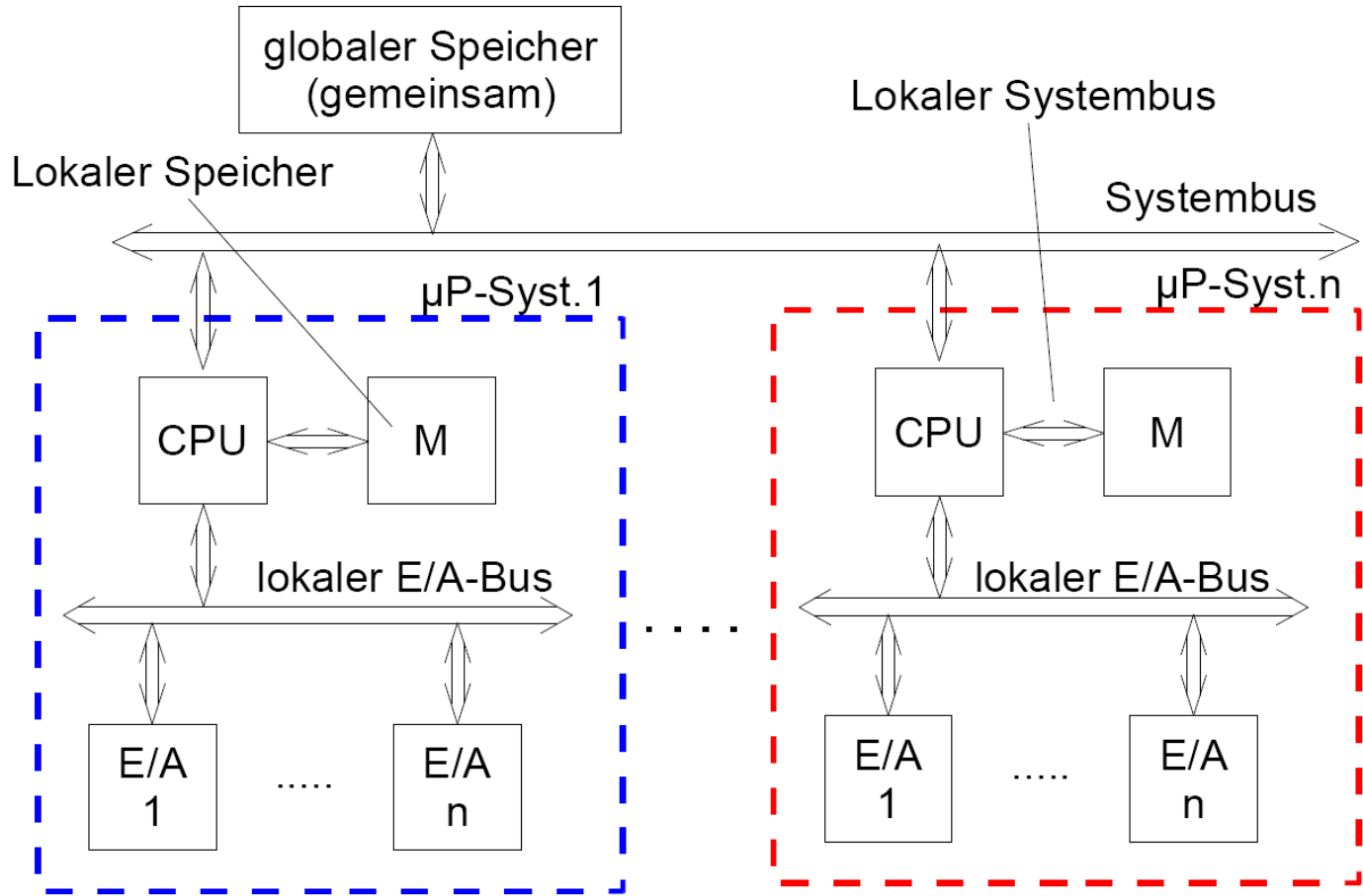
2.1 Automatisierungsrechner

■ Mikrocontroller 80C167



2.1 Automatisierungsrechner

■ Multiprozessorsystem mit Dreifach-Bus

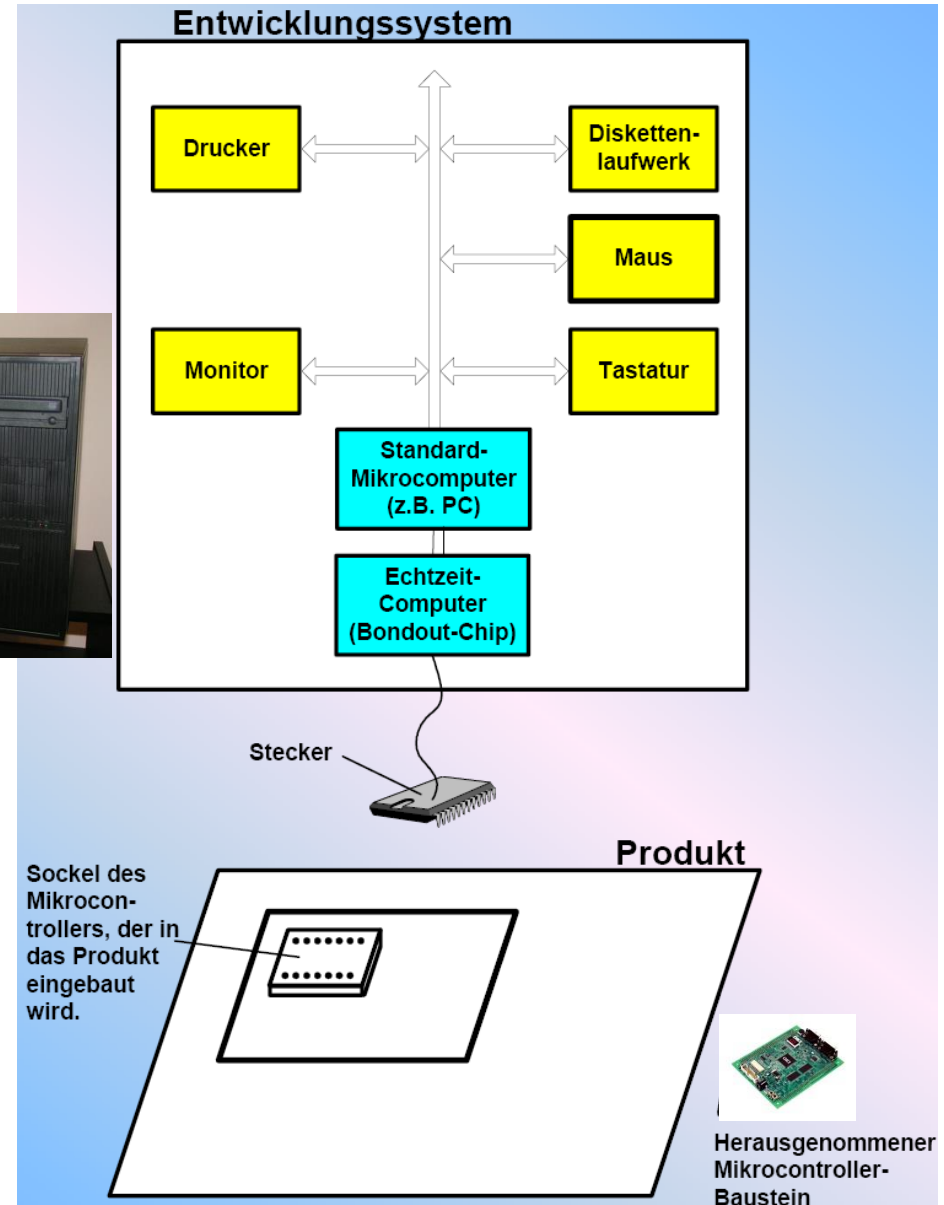


M: lokaler Speicher

2.1 Automatisierungsrechner

■ Mikrocontroller-Entwicklungsumgebung

Rechnersystem mit Standardperipherie und Systemprogrammen



2.1 Automatisierungsrechner

■ Industrie-PC (IPC)

- Einsteckbare Leiterplatten zum Anschluss von elektrischen, optischen Prozesssignalen oder Bussystemen
- Programmierung in Hochsprachen
- Einsatz von Echtzeit-Betriebssystemen



■ Merkmale von IPCs

- Eignung für **raue Umgebungsbedingungen**: Temperaturschwankungen, Stöße und Erschütterungen, Staub und Feuchtigkeit, elektrische oder elektromagnetische Störungen
- Schutzvorrichtungen von IPCs: schwingungsgedämpfte Laufwerke, hohe Güte der integrierten Bausteine, spezielles Schutzgehäuse

2.1 Automatisierungsrechner

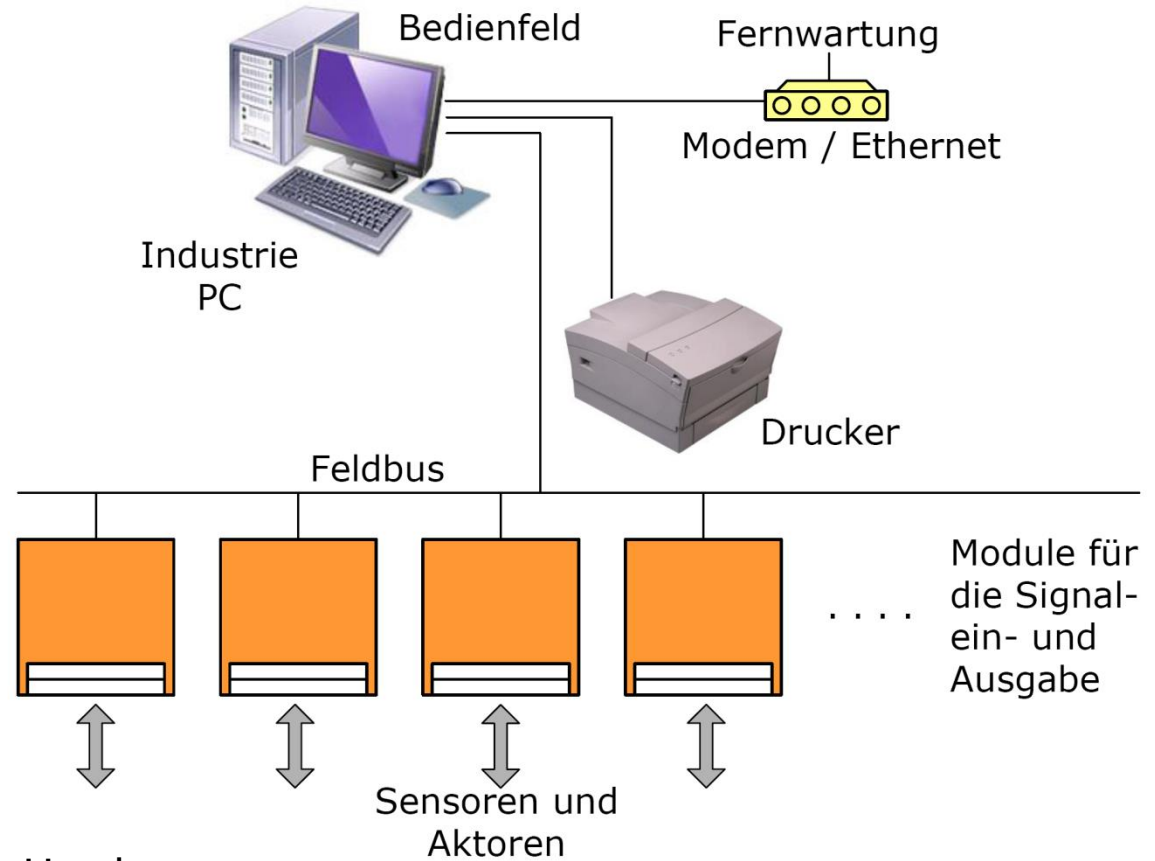
■ Schutzart von IPCs

IP(Ingres Protection)-Index:

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

2.1 Automatisierungsrechner

■ Reines IPC-System

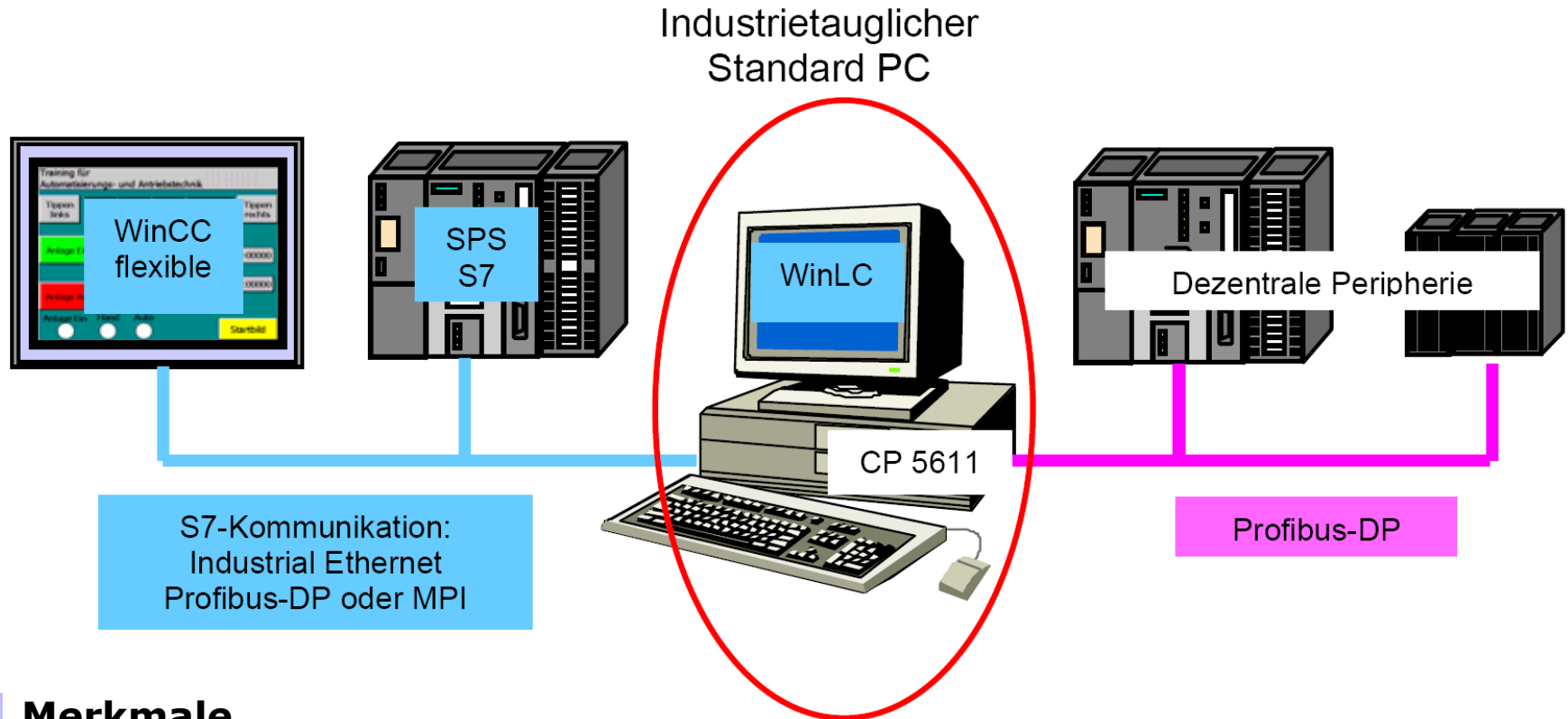


■ Vorteile

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

2.1 Automatisierungsrechner

■ SPS-IPC-System („Soft-SPS“)



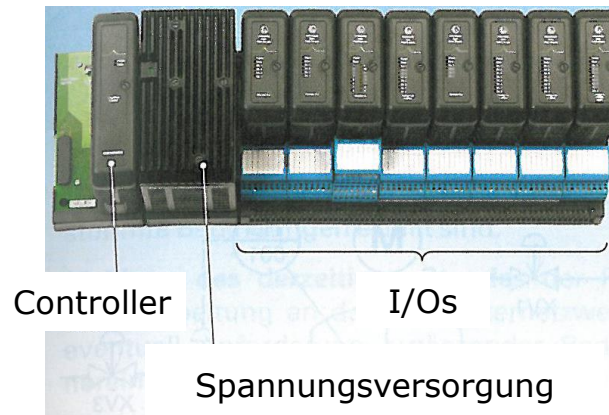
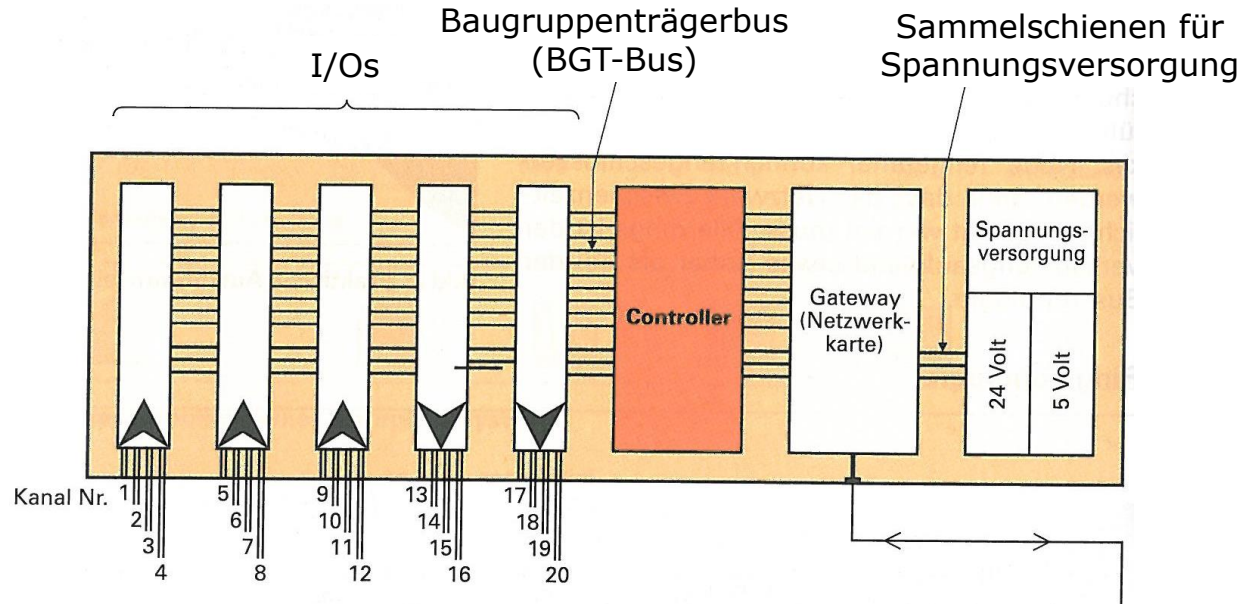
■ Merkmale

- IPC dient als Schnittstelle zum Bediener.
- SPS übernimmt Einlesen/Ausgeben der Prozesssignale und Abarbeitung der Steuerungsalgorithmen
- Integrierte Entwicklungsumgebung zur Modifikation der Programme

2.1 Automatisierungsrechner

■ Controller-Baugruppen/Computer-Module

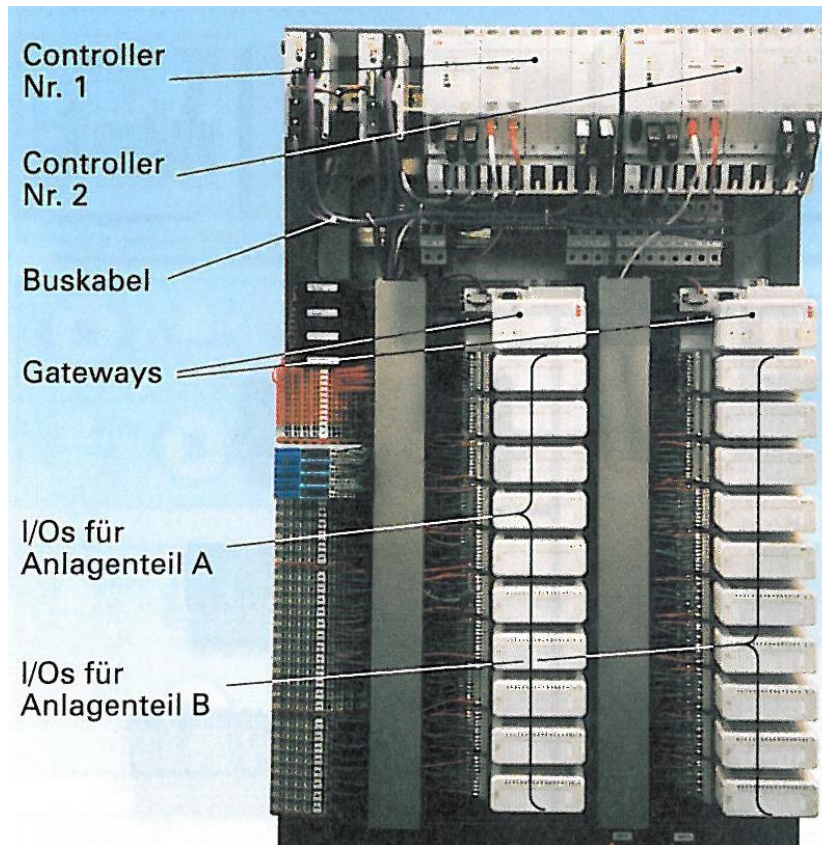
Ausführung ausgelagerter Funktionen auf sog. Prozessstationen



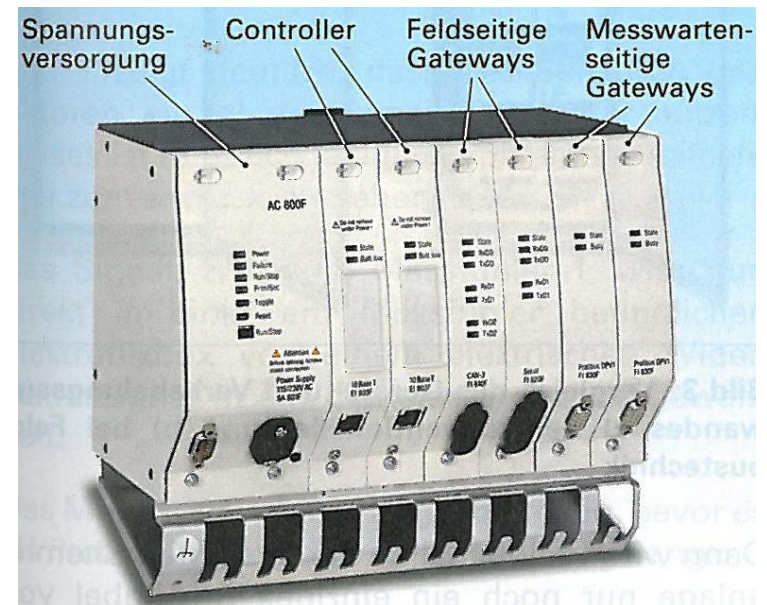
2.1 Automatisierungsrechner

■ Controller-Baugruppen/Computer-Module

Zwei Controller mit ihren zwei zugehörigen Reihen von I/Os



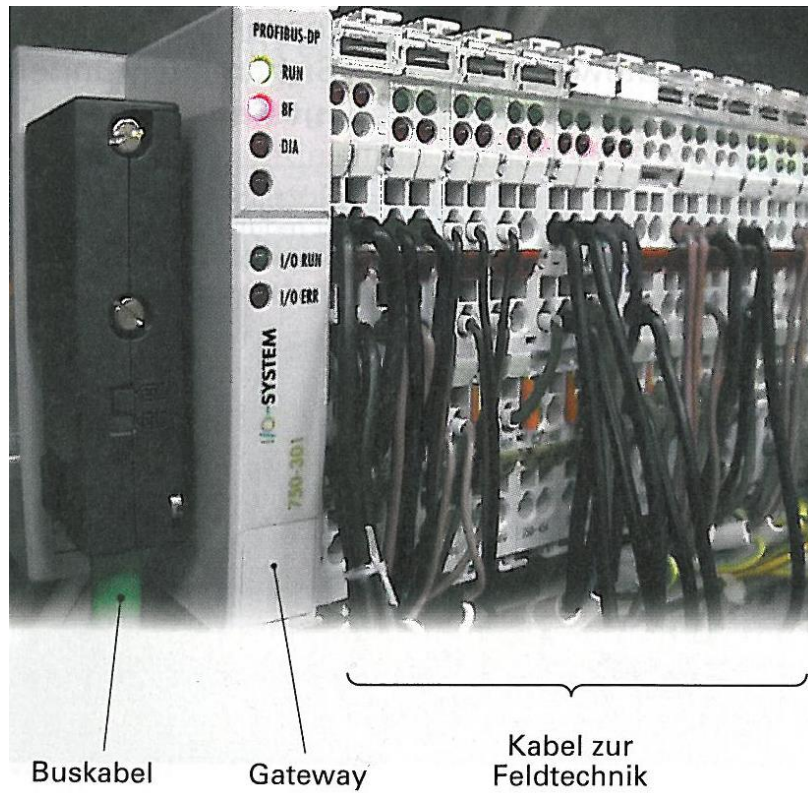
Controllerbaugruppe mit angebauten und jeweils doppelt vorhandenen Gateway-Modulen



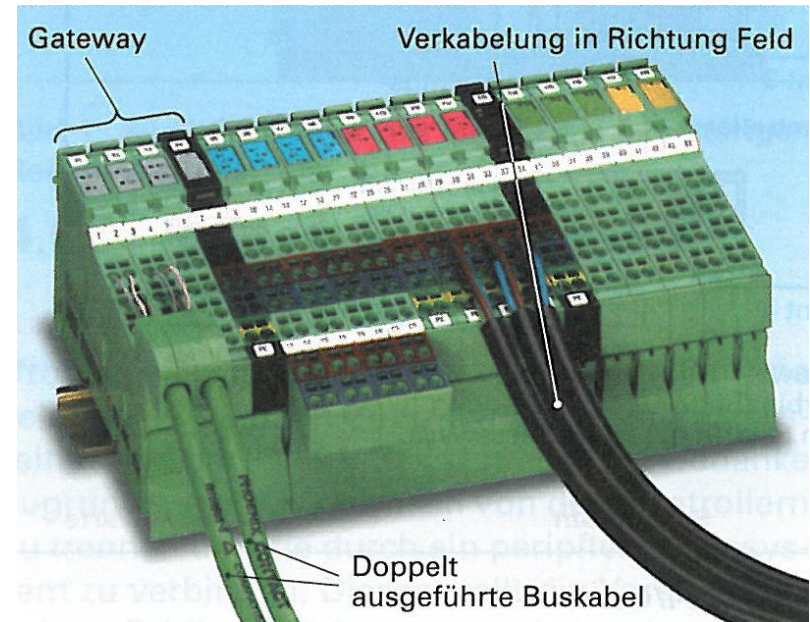
2.1 Automatisierungsrechner

Remote I/Os (Dezentrale Peripherie)

Remote I/Os mit Gateway
und Buskabel



Redundante Buskabelanschlüsse
am Gateway von Remote I/Os



2.1 Automatisierungsrechner

■ Prozessleitsysteme (PLS)

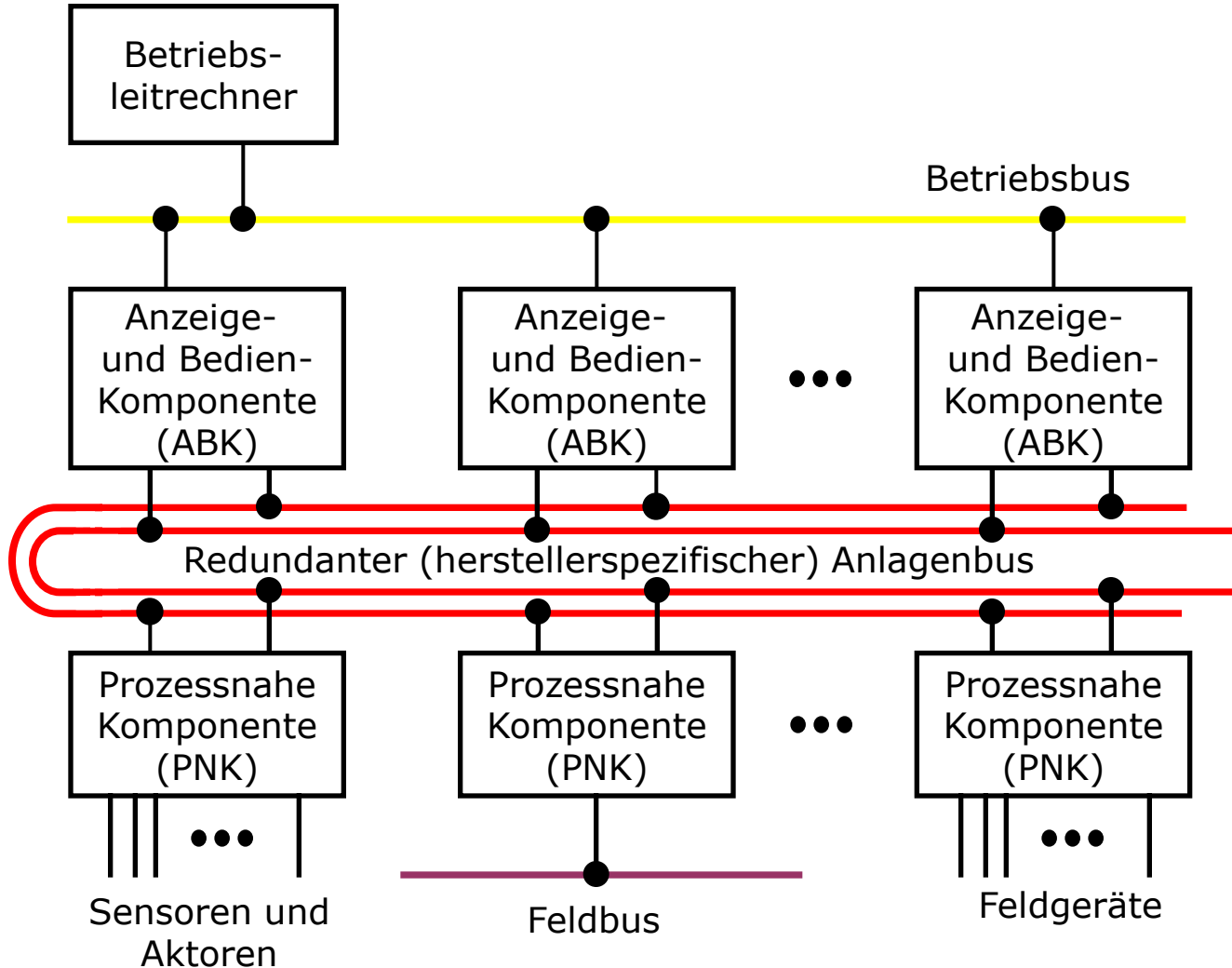
- Verteilte, über Bus-Systeme verbundene Rechnersysteme
- Kopplung mit SPS-Rechnern
- Einsatz von vorkonfektionierten, vom Hersteller des PLS entwickelten Programmbausteinen
- Konfigurierung durch Anwender

■ Vorteile einer PLS-Komplettlösung von einem Hersteller

- keine Kompatibilitätsprobleme
- einheitliche Bedienung und Beobachtung des Prozesses
- hohe Verfügbarkeit
- definierte Verantwortlichkeit
- lange Lebenszeit

2.1 Automatisierungsrechner

■ Schematischer Aufbau von PLS



Leitstand

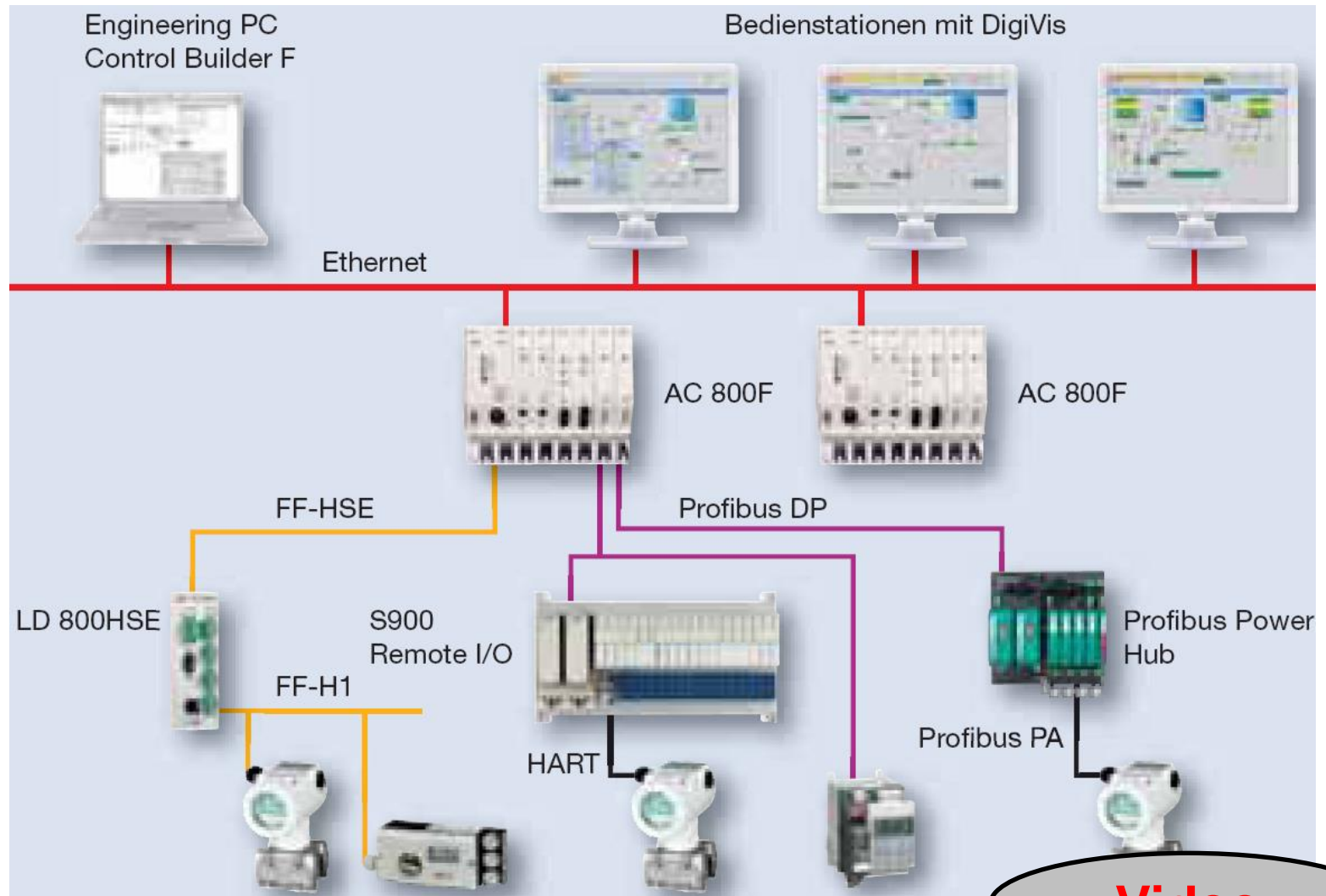


Schaltschrank



2.1 Automatisierungsrechner

■ PLS Freelance 800F (ABB)



Video

2.1 Automatisierungsrechner

■ PLS / Anzeige- und Bedienkomponente (ABK)

- Funktionen: Rezepte/Batch-Abläufe erstellen, modifizieren; aktuelle Sollwerte ändern; Kommunikation mit dem Prozess; Alarme/ Bedienanforderungen bearbeiten; Prozessvisualisierung; Schnittstelle zu Datenbanksystemen zur Prozessprotokollierung
- Realisierung: PC, IPC, Workstation; Windows 2000 / XP / Vista / Windows 7; Unix

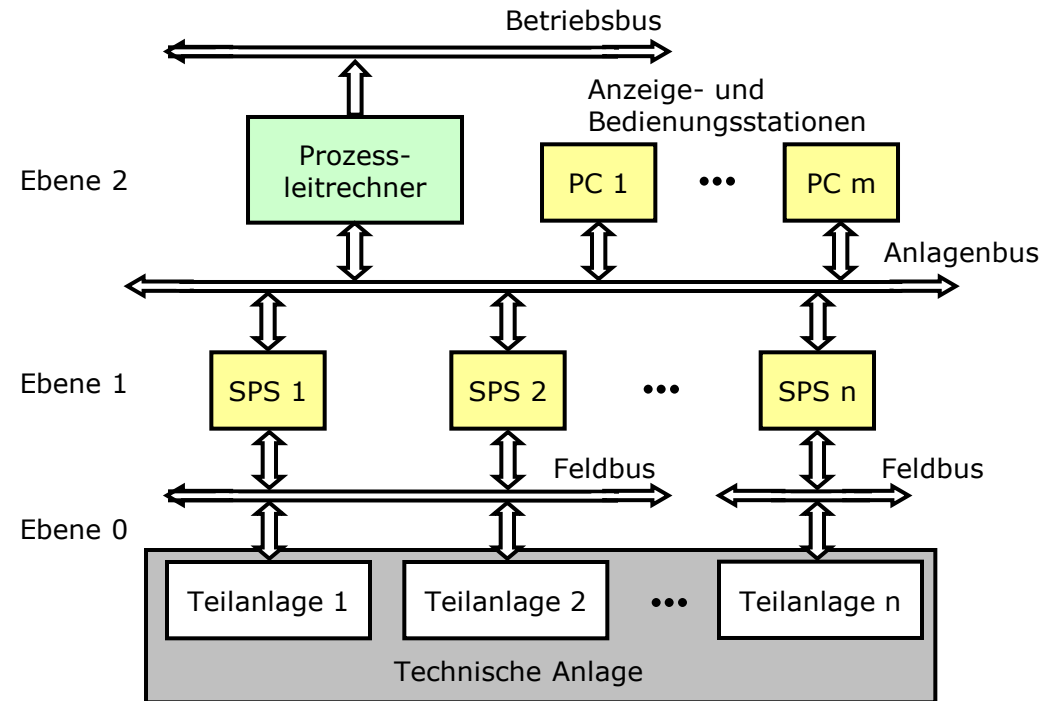
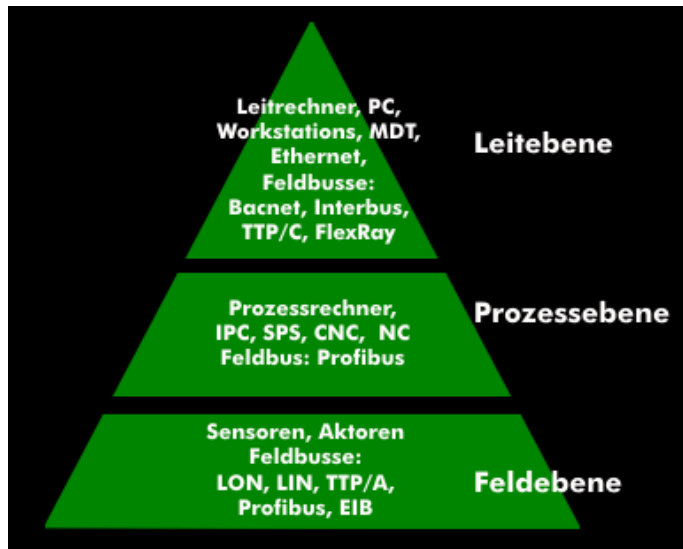
■ PLS/ Prozessnahe Komponente (PNK)

- Funktionen: Regelungs- und Steuerfunktionen mit kleiner Zykluszeit; Grenzwertprüfung; Zwischenpuffern von Prozesssignalen
- Realisierung: Eigenentwicklungen der Leitsystem-Hersteller: Automatisierungsrechner (SPS, IPC); dezentrale Peripherie; Feldgeräte (Sensoren, Aktoren)

2.1 Automatisierungsrechner

■ PLS / Systemkommunikation/Bussysteme

- Betriebsleitebene: standardisierter Ethernet-Bus
- Prozessleitebene: Feldbus, z.B. Profibus, H1-Bus, Modbus, Interbus-S
- Unterste Feldebene: schnelle Aktor-Sensor-Busse



2.1 Automatisierungsrechner

■ PLS / Engineering-Tool

- Aufgaben: Konfiguration; Programmierung; Wartung/Pflege
- Werkzeuge: graphische Werkzeuge (IEC 61131);
standardisierte Bibliotheken mit Komponentenmächtigen Editoren



Video

2.1 Automatisierungsrechner

■ Leitsystemhersteller und deren Produkte

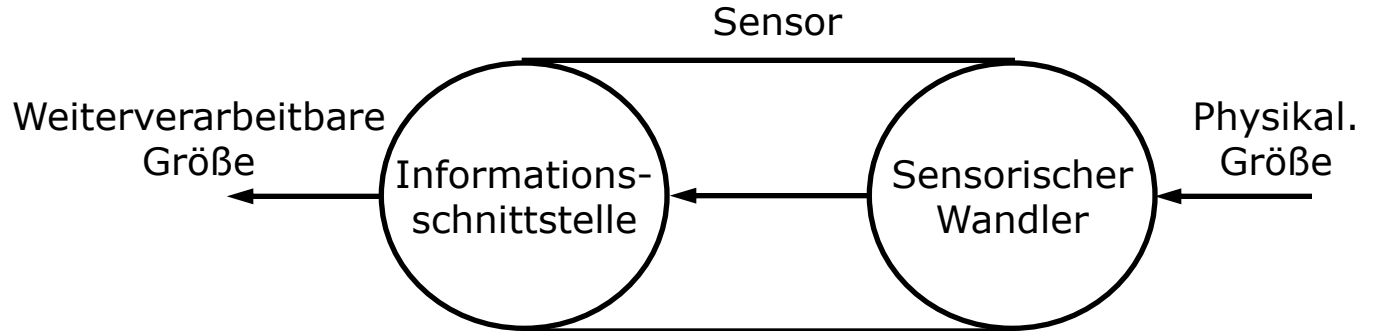
Hersteller	Produktname	Bemerkung
ABB	AdvantOCS	Firmenspezifischer Feldbus
	AdvaSoft	Für kleine Anlagen
	Procontrol P	Kraftwerktechnik
	Symphony	Unterstützt Anbindung an Betriebsebene; RemoteI/O System
	Contronic P	Verfahrenstechnik; Ausdehnung bis zu 12km; ehemals Hartmann & Braun
Foxboro-Echardt	I/A Series System	Verfahrenstechnik; PNK mit PCMCIA-Technologie
Siemens	SIMATIC PCS 7	Verfahrenstechnik, ABK auf Basis von Windows 2000, XP; Umfangreiches Hardwareangebot; Feldbus: Profibus; Anbindung an Betriebsebene möglich
	Teleperm M	Verfahrenstechnik; weit verbreitet; kein Einsatz in Neuanlagen; Migration von TelepermM nach SIMATIC PCS 7 möglich
	Teleperm XP	Kraftwerkstechnik; offene Kommunikation; umfangreiches Hardwareangebot.
Honeywell	PlantScape	Verfahrenstechnik; offenes System, ABK auf Basis von Windows NT; unterstützt RemoteI/O; Anbindung an Betriebsebene möglich
	TDC 3000	Verfahrenstechnik; besitzt mehrere Prozessbusse mit unterschiedlicher Datenübertragung; MODBUS wird unterstützt.
Yokogawa	CENTUM CS 3000	Verfahrenstechnik; Kraftwerktechnik

2.2 Sensoren

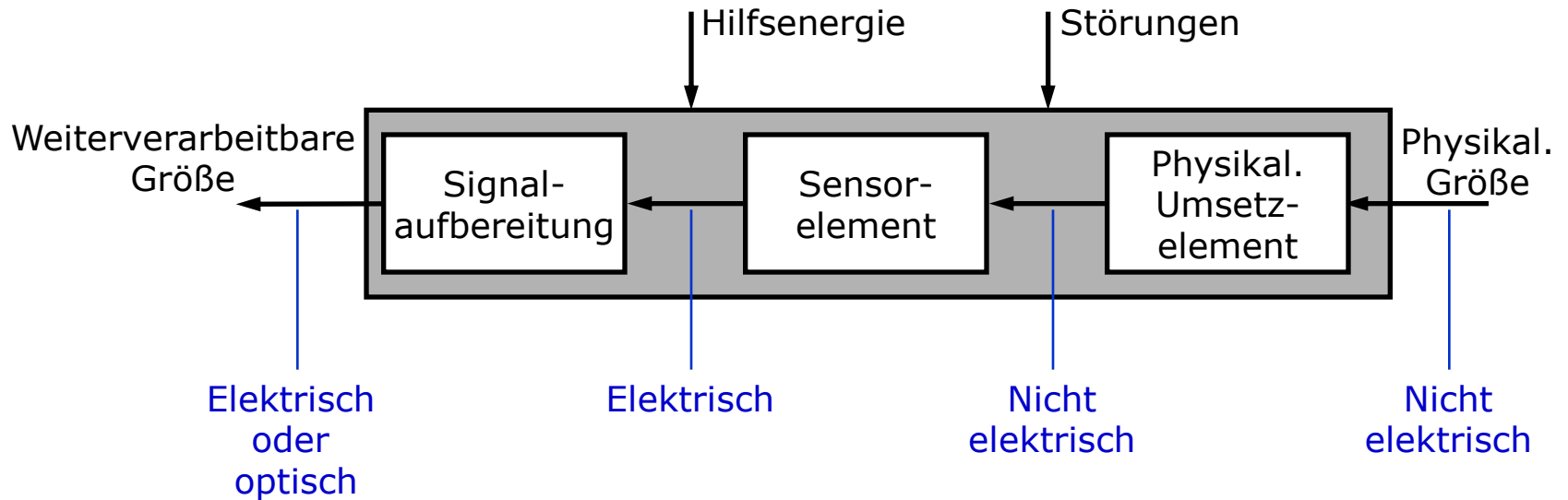
■ Sensoren (Messsysteme) – Sinne des Automatisierungssysteme

■ Bestandteile von Sensoren

- Prinzip



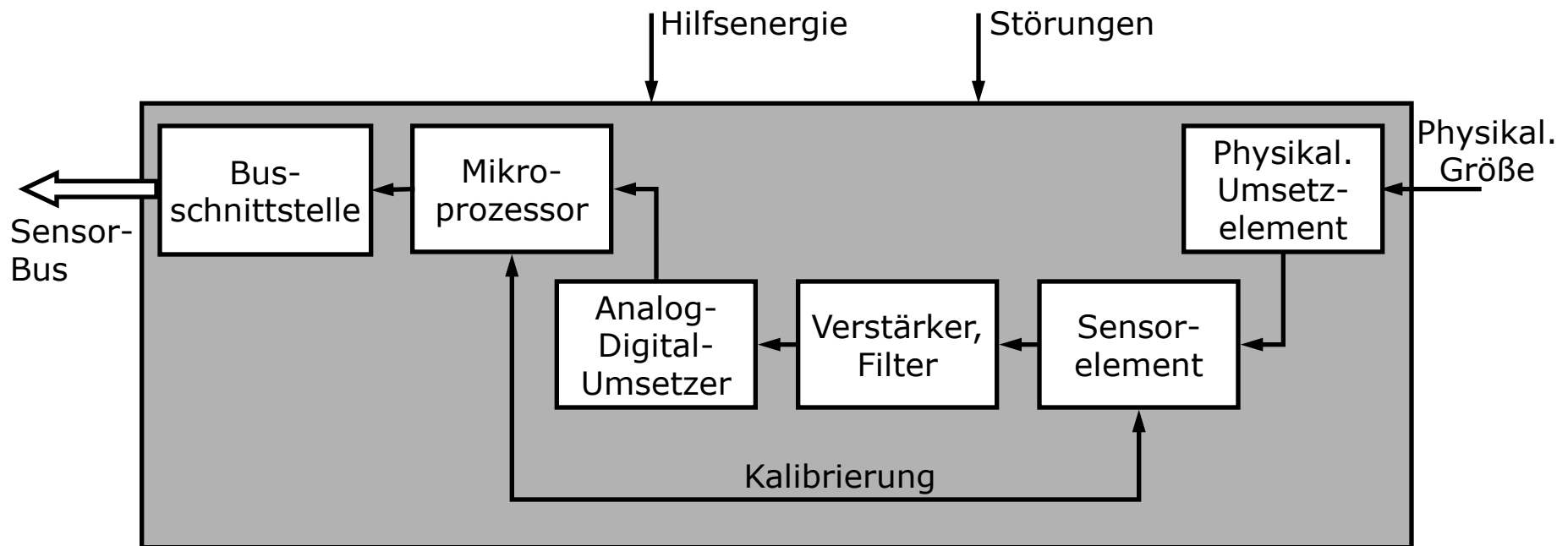
- Detailauflösung



2.2 Sensoren

■ Intelligenter Sensor

- Integration eines Sensors in ein Mikrorechner: Vorverarbeitung des Messsignals (z.B. Linearisierung, Temperaturkompensation, Skalierung), autonome Kalibrierung, ggf. Durchführung von komplexen Auswertefunktionen (z.B. Leistungsspektren, FFT)
- Dezentrale Informationsverarbeitung möglich



2.2 Sensoren

■ Klassifizierung nach der Art des Sensorausgangssignals

- binäre Sensoren
- digitale Sensoren
- amplitudenanaloge Sensoren
- frequenzanaloge Sensoren

2.2 Sensoren

■ Binäre Sensoren

- Ja/Nein-Information
- Grenzwertgeber/Schwellwertgeber
- Beispiele: Endschalter, berührungslos arbeitende Gabellichtschranke

■ Digitale Sensoren

- Umwandlung der nichtelektrischen Messgröße in ein digitales Ausgangssignal
- Umsetzarten
 - Direkte Umsetzung: Umwandlung des nichtelektrischen Signals mittels Codescheibe (Drehbewegung) oder Codelineal (Linearbewegung) in ein digitales Signal
 - Indirekte Umsetzung: Umwandlung des nichtelektrischen Signals in amplitudenanaloges Signal, dann Analog-Digital-Wandlung

2.2 Sensoren

■ Amplitudenanaloge Sensoren

- Umformung einer nichtelektrischen Messgröße in ein amplitudenanaloges Ausgangssignal, d.h. Amplitude des elektrischen Ausgangssignals ist proportional der zu messenden physikalischen Größe
- Oft mit Umformerstufe im selben Gehäuse
 - Verstärkung
 - Normierung
 - Linearisierung
 - Temperaturkompensation
- Für alle wichtigen Messgrößen **verfügbar**

2.2 Sensoren

■ Frequenzanaloge Sensoren

- Liefern ein Wechselspannungssignal, dessen Frequenz der zu messenden physikalischen Größe proportional ist.
- Vorteile:
 - Unempfindlichkeit gegen Störbeeinflussung auf den Übertragungsleitungen
 - einfache Frequenz-Digitalumsetzung durch Zähltechnik
 - kein Genauigkeitsverlust bei Verstärkung und Übertragung
 - einfache Potenzialtrennung mit Übertragern
- **Selten** im Vergleich zu amplitudenanalogen Sensoren

2.2 Sensoren

■ Arten

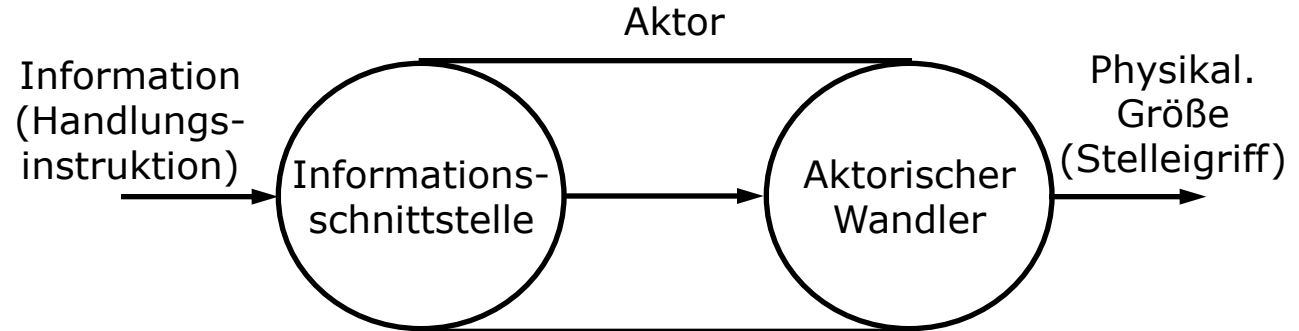
Physikalische Prozessgröße	Sensorelement	Ausgangsgröße
Temperatur	Thermoelement Metallische Widerstände Halbleiter-Widerstände (Heißeiter) Keramik-Widerstände (Kaltleiter)	mV Widerstandsänderung Widerstandsänderung Widerstandsänderung
Druck	Druckdose mit Membran und Dehnungsmessstreifen Druckdose mit Silizium-Membran (piezoresistiver Effekt)	Widerstandsänderung Widerstandsänderung
Kraft	Dehnungsmessstreifen Induktiver Kraftmessfühler Piezoelektrischer Fühler	Widerstandsänderung Induktivitätsänderung Ladung
Drehzahl	Tachogenerator Impulszählung	V Impulsfolge
Beschleunigung	Silizium-Piezowiderstand Silizium-Kondensator	Widerstandsänderung Kapazitätsänderung
Durchfluss	Ringkolbenzähler Induktive Durchflussmesser	Impulsfolgem V
Annäherung	Hallelement aus Silizium	mV
Winkel	Winkelkodierer Impulsgeber	Digitalwert Impulsfolge
Feuchte	Lithiumchlorid-Feuchtefühler	mV
Lichtintensität	Photodiode Photowiderstand	µA Widerstandsänderung

2.3 Aktoren

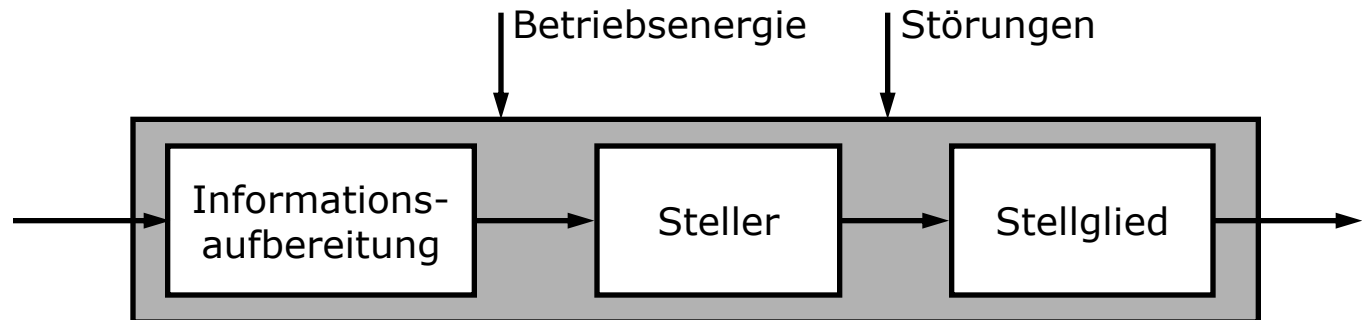
■ Aktoren (Stellsysteme) – Muskeln des Automatisierungssystems

■ Bestandteile von Aktoren

- Prinzip



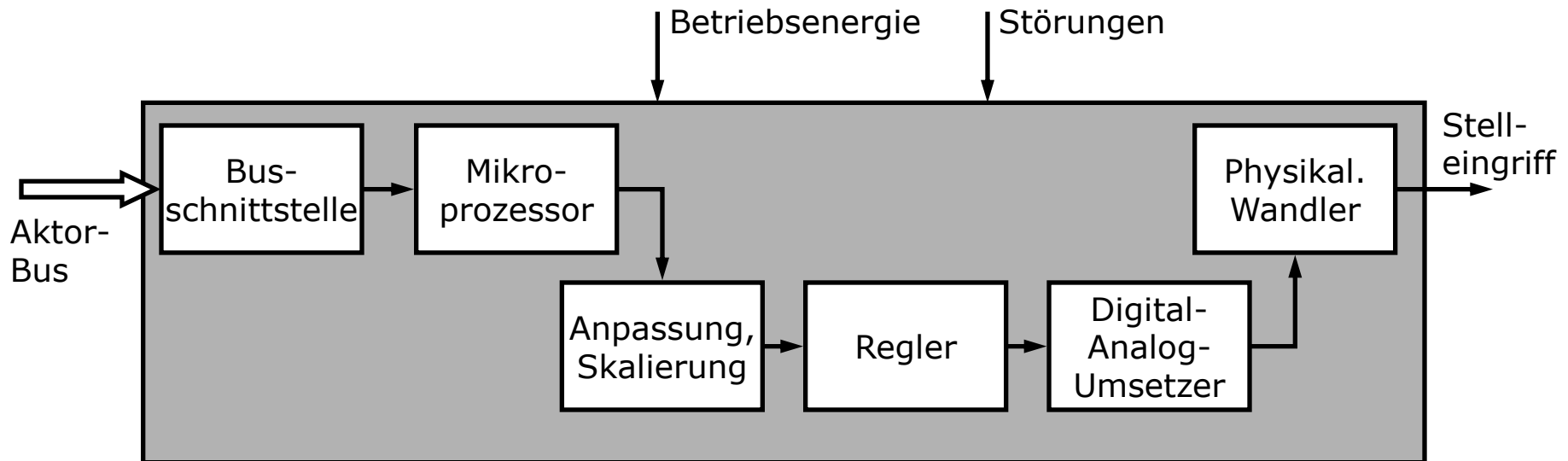
- Detailauflösung



2.3 Aktoren

■ Intelligenter Aktor

- Integration eines Aktors in ein Mikrorechner: lokale Datenaufbereitung, Erfüllung lokaler regelungstechnischer Aufgaben, Koordinierung der Kommunikation mit einem Feldbussystem, Realisierung von Sicherheits- und/oder Diagnosefunktionen
- Komponente eines verteilten Steuerungssystems mit eigener Steuerungsfunktionalitäten



2.3 Aktoren

■ Aktoren mit mechanischer Ausgangsgröße

Aktorprinzip	Aktoren
Elektromechanische Bewegung	Elektromotor, Schrittmotor, Elektromagnet, Linearmotor
Hydraulischer Stellzylinder	Hydraulische Aktoren
Pneumatischer Stellzylinder	Pneumatische Aktoren
Piezoelektrischer Effekt	Piezoelektrische Aktoren, Elektrostriktive Aktoren
Magnetostriktiver Effekt	Magnetostriktive Aktoren
Elektrorheologischer Effekt	Elektrorheologische Aktoren
Magnetorheologischer Effekt	Magnetorheologischer Aktoren
Elektrisch erzeugter Gasdruck	Chemische Aktoren
Bimetall-Effekt	Thermo-Bimetall-Aktoren
Memory-Metalle	Memory-Metalle-Aktoren
Silizium-Mikrotechnik	Mikromechanische Aktoren

2.3 Aktoren

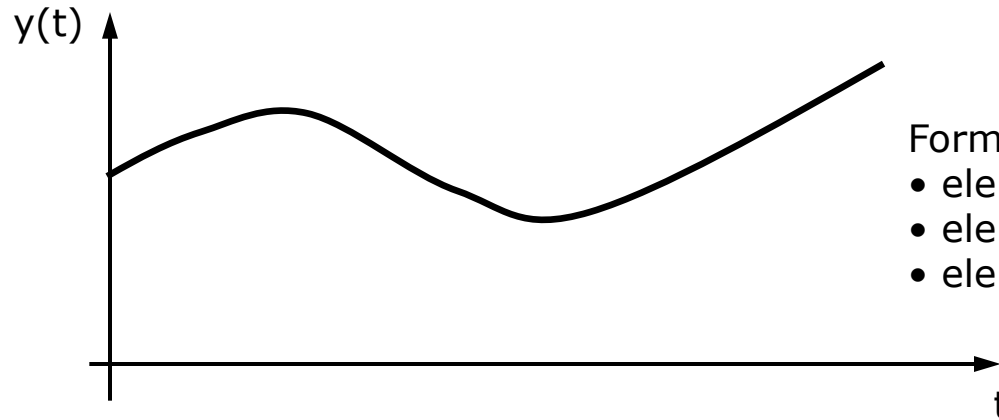
■ Erklärungen zu Aktorprinzipien

- Piezoelektrischer und elektrorestriktiver Effekt: Anlegen einer elektrischen Spannung an einen piezoelektrischen Kristall oder Keramikkörper führt zu einer Änderung der Länge des Körpers.
- Magnetorestriktiver Effekt: Bestimmte Eisenlegierungen mit Elementen aus der Gruppe der seltenen Erden dehnen sich aus oder ziehen sich zusammen unter dem Einfluss eines magnetischen Feldes.
- Elektrorheologischer oder magnetorheologischer Effekt: Flüssigkeiten ändern ihren Fließwiderstand (ihre Viskosität) unter dem Einfluss eines äußeren elektrischen bzw. magnetischen Feldes innerhalb weniger Millisekunden um mehrere Größenordnungen.
- Chemischer Aktorprinzip: beruht auf einer Gasentwicklung, die beim Anlegen einer kleinen Gleichspannung einsetzt und die in einem geschlossenen Behälter einen Druck aufbaut; beim Umpolen oder bei Kurzschluss baut sich der Druck wieder ab.
- Memory-Metalle (Gedächtnislegierungen): ändern ihre Form bei einer bestimmten Temperatur in genau vorhersagbarer Weise.
- Mikromechanischer Effekt: Ätzen von Silizium

2.4 Ein- und Ausgabegeräte

■ Arten von Signalen

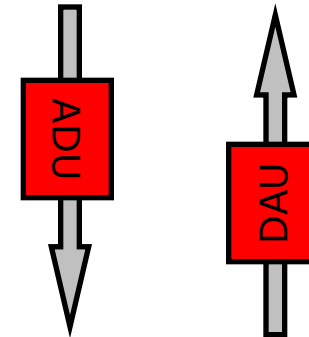
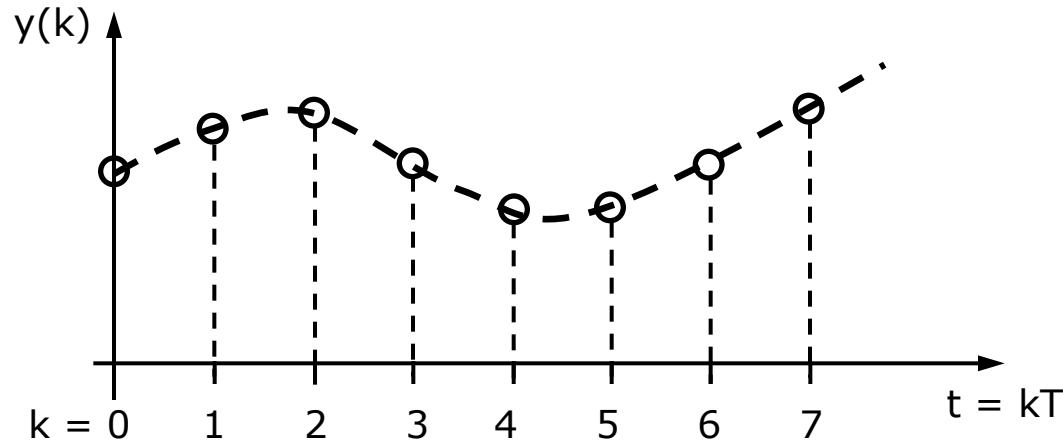
- Analoge (kontinuierliche) Signale



Formen:

- elektrische Spannungssignale
- elektrische Stromsignale
- elektrische analoge Widerstandswerte

- Digitale (diskrete) Signale

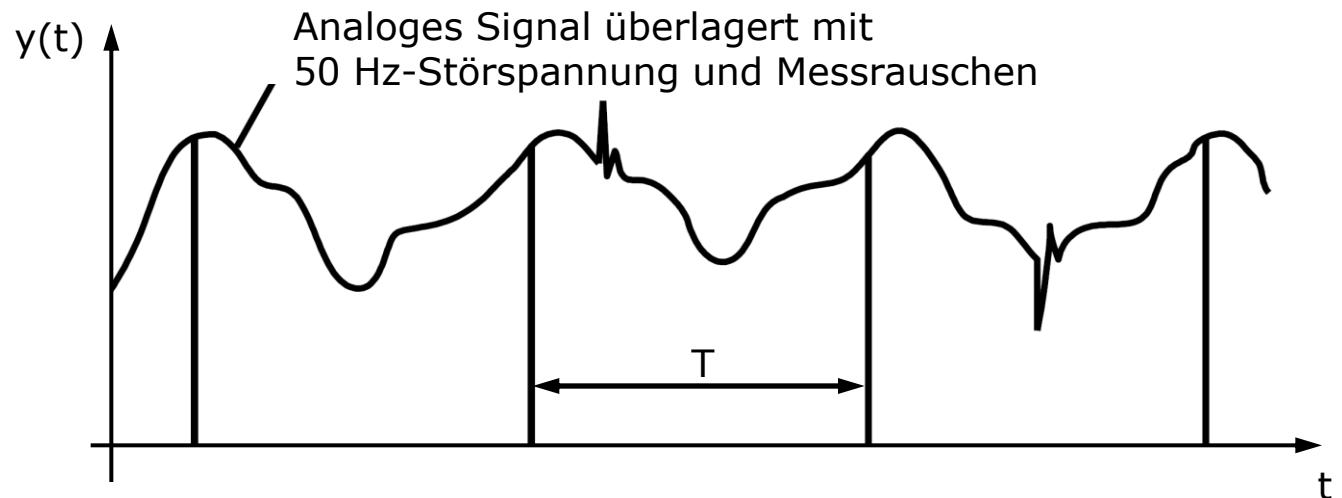


2.4 Ein- und Ausgabegeräte

■ Analog-Digital-Umsetzer (ADU)

- **Momentanwert**umsetzer:

Einzelne Werte eines analogen Signals werden abgetastet und in digitale Werte umgesetzt.



- **Mittelwert**umsetzer (Integrierender Umsetzer)

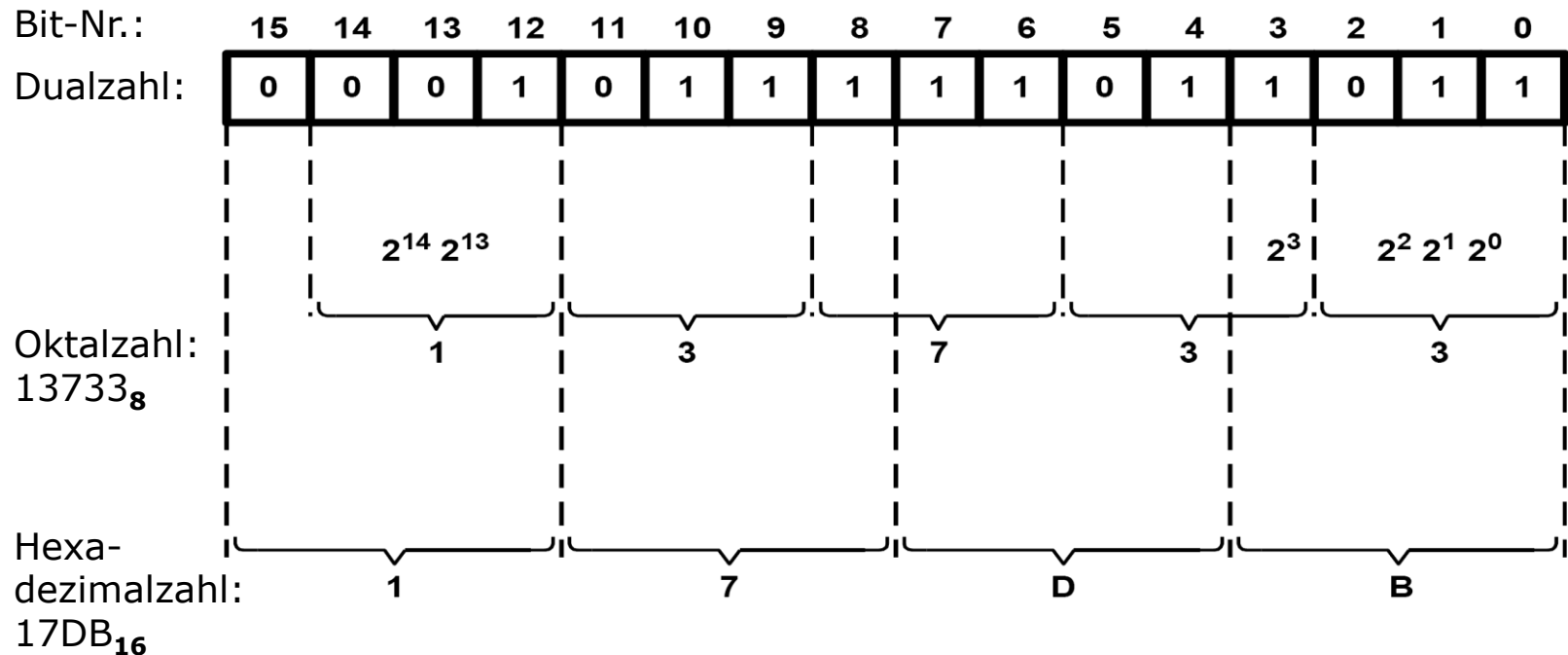
- Analoges Signal wird über eine Periode T der Netzfrequenz integriert und der Mittelwert gebildet.
- Mittelwert wird in digitalen Wert umgesetzt:

$$y(kT) = \int_{(k-1)T}^{kT} y(t) dt$$

2.4 Ein- und Ausgabegeräte

■ Darstellung digitaler Prozessgrößen

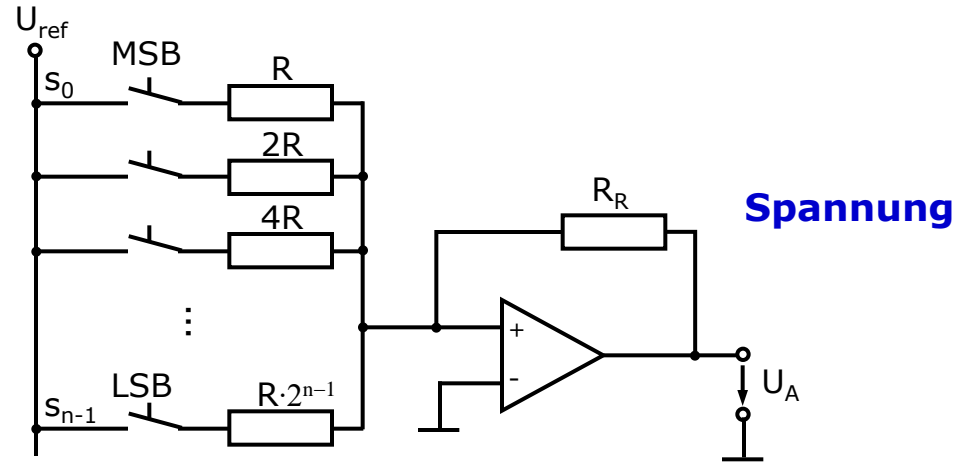
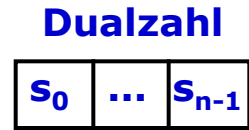
Dual-, Oktal- oder Hexadezimalzahlen



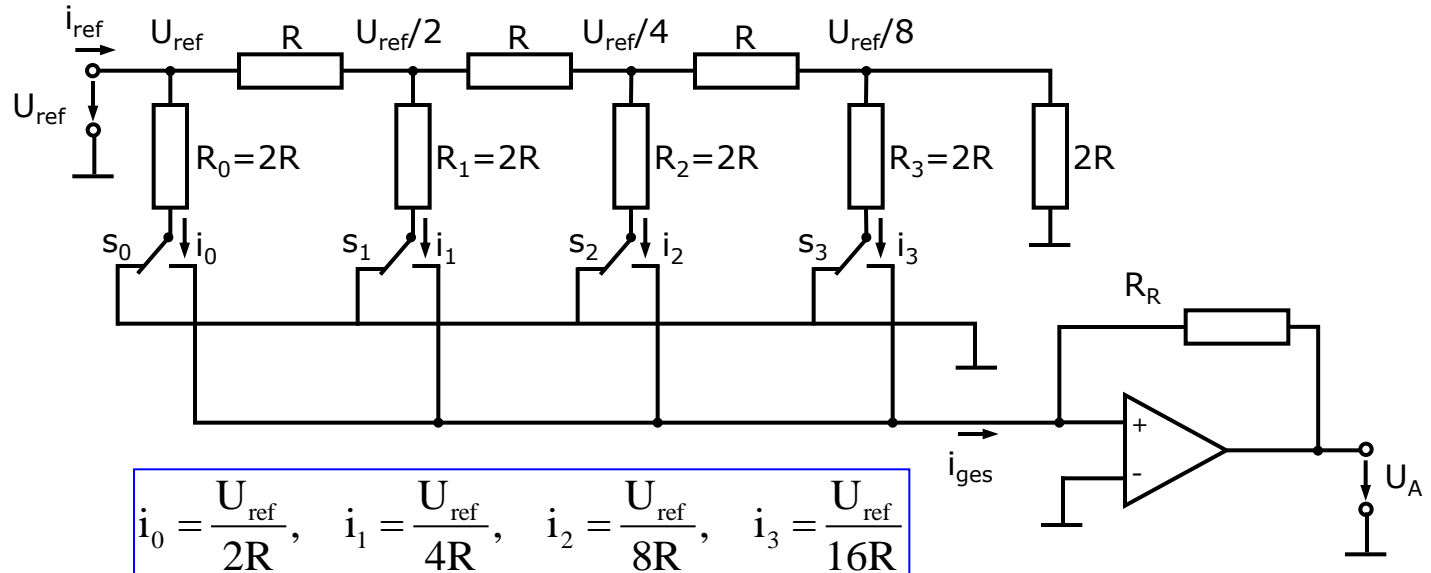
2.4 Ein- und Ausgabegeräte

Digital-Analog-Umsetzer (DAU)

- Summation gewichteter Ströme:



- Leiternetzwerk:



2.4 Ein- und Ausgabegeräte

■ Schnittstellen

Messgrößenerfassung

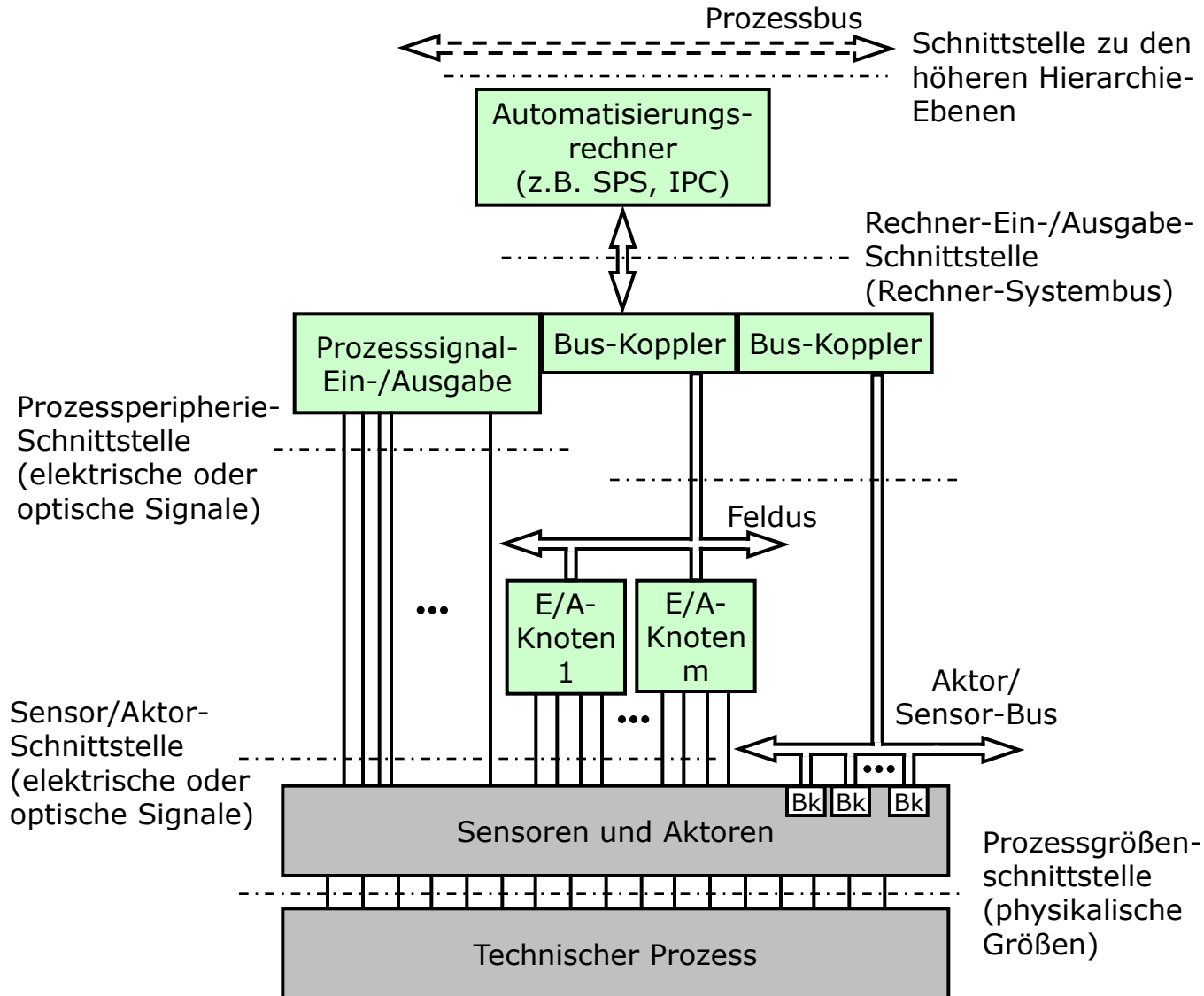
Ansteuerung von Stellgliedern



- Möglichkeiten zur Ankopplung von Sensoren und Aktoren
 - Direkter Anschluss (Leitungsbündel)
 - Anschluss über Feldbussysteme
 - Anschluss über Sensor/Aktor-Bussysteme (bei intelligenten Geräten)

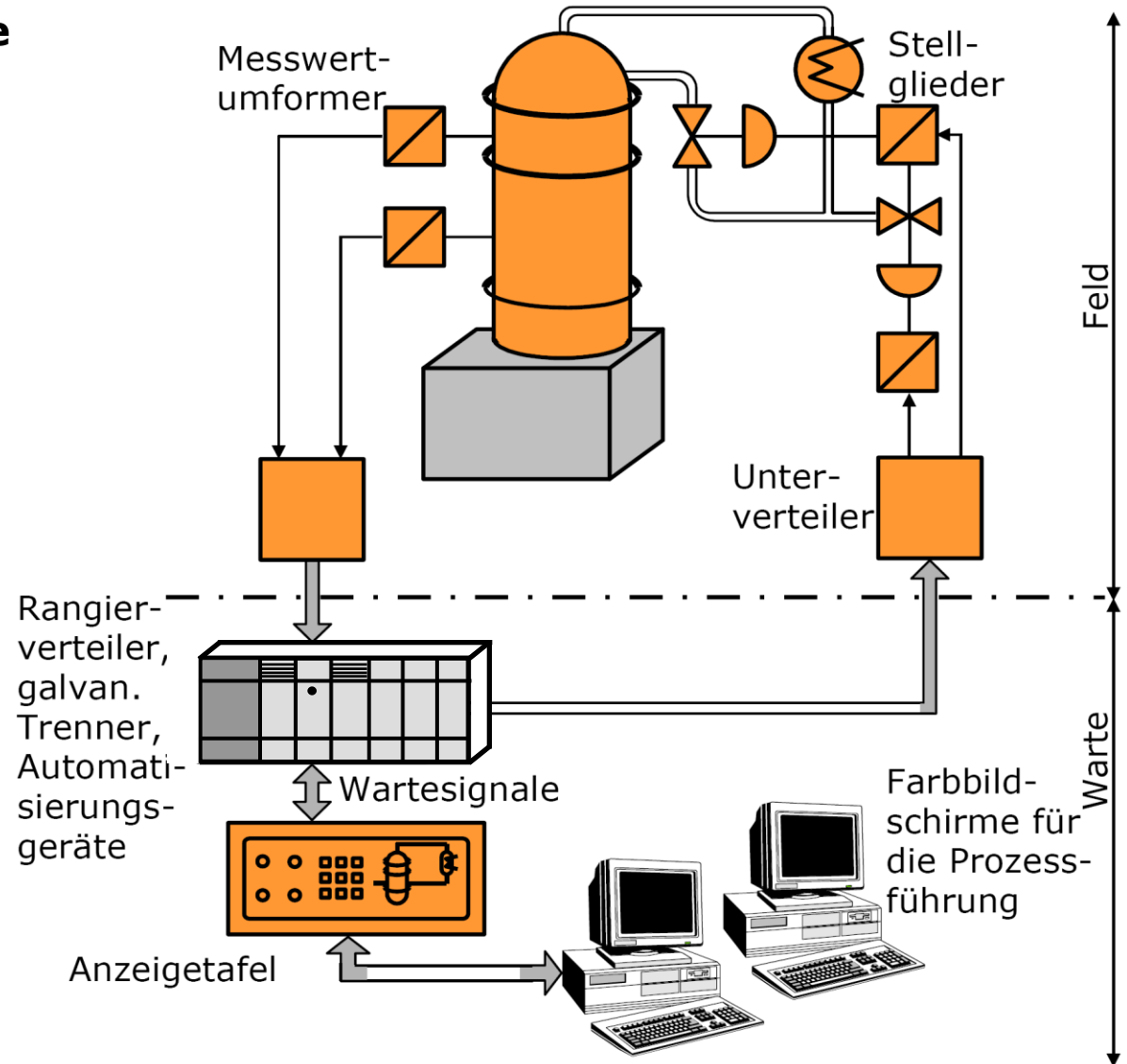
2.4 Ein- und Ausgabegeräte

Realisierung der Schnittstellen



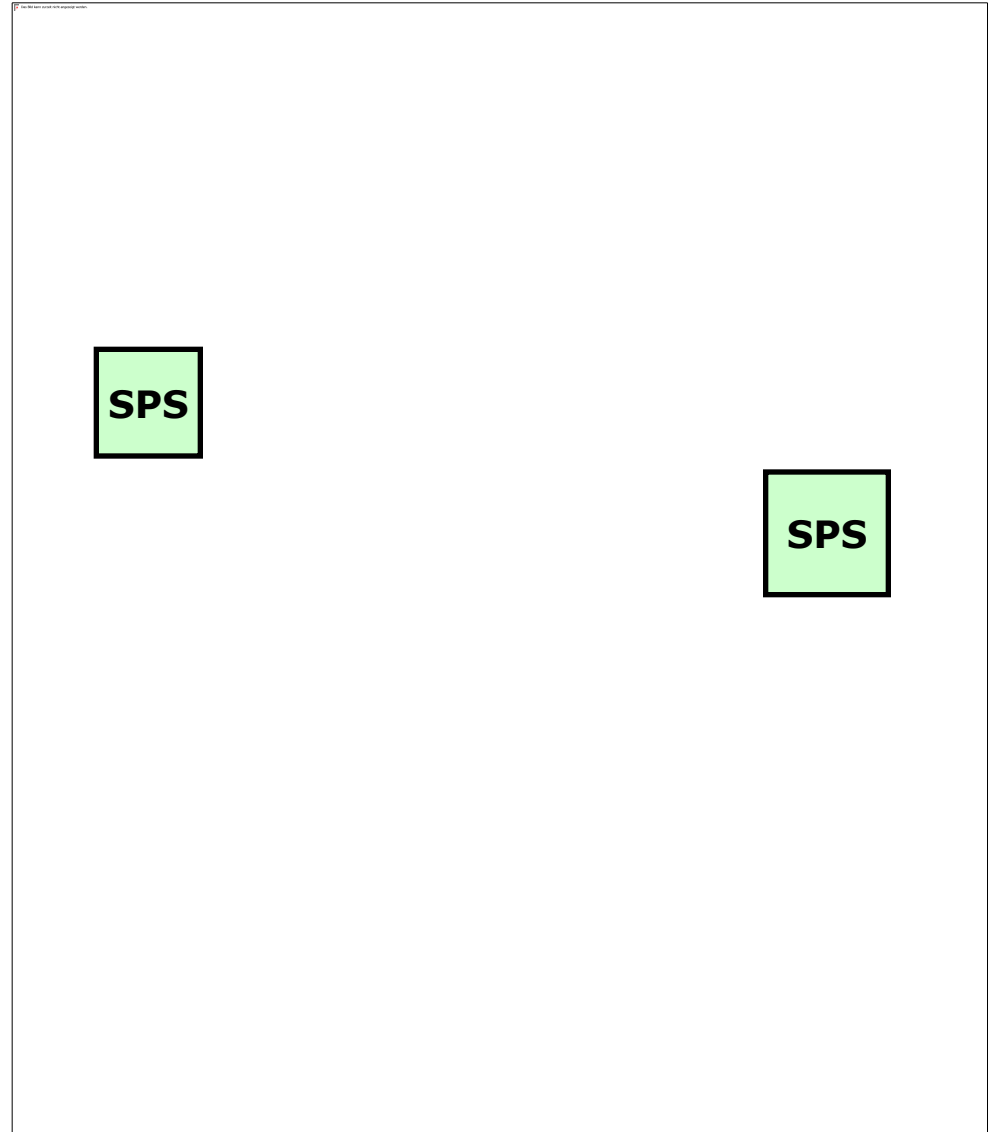
2.5 Automatisierungsstrukturen

■ Örtlich zentrale Anordnung der Automatisierungsgeräte



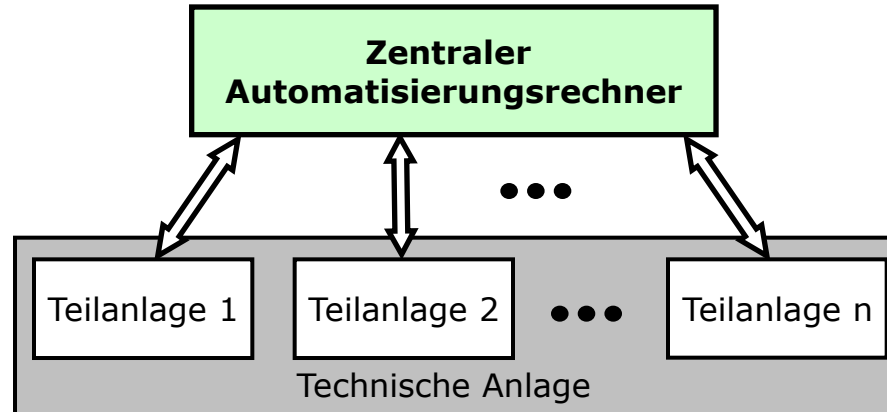
2.5 Automatisierungsstrukturen

- **Örtlich dezentrale Anordnung der Automatisierungsgeräte**

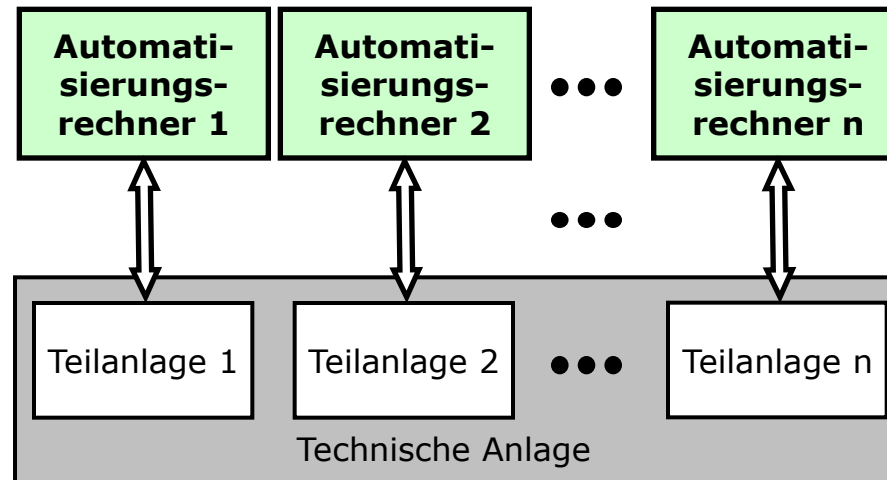


2.5 Automatisierungsstrukturen

■ Wirkungsmäßig zentrale Automatisierungsstruktur



■ Wirkungsmäßig dezentrale Automatisierungsstruktur

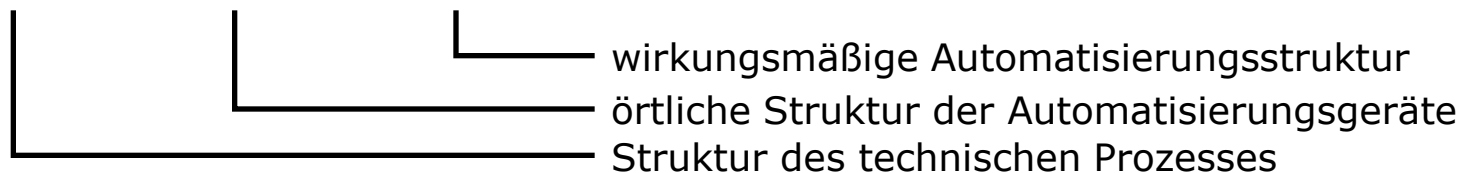


2.5 Automatisierungsstrukturen

■ Kombination von Automatisierungsstrukturen

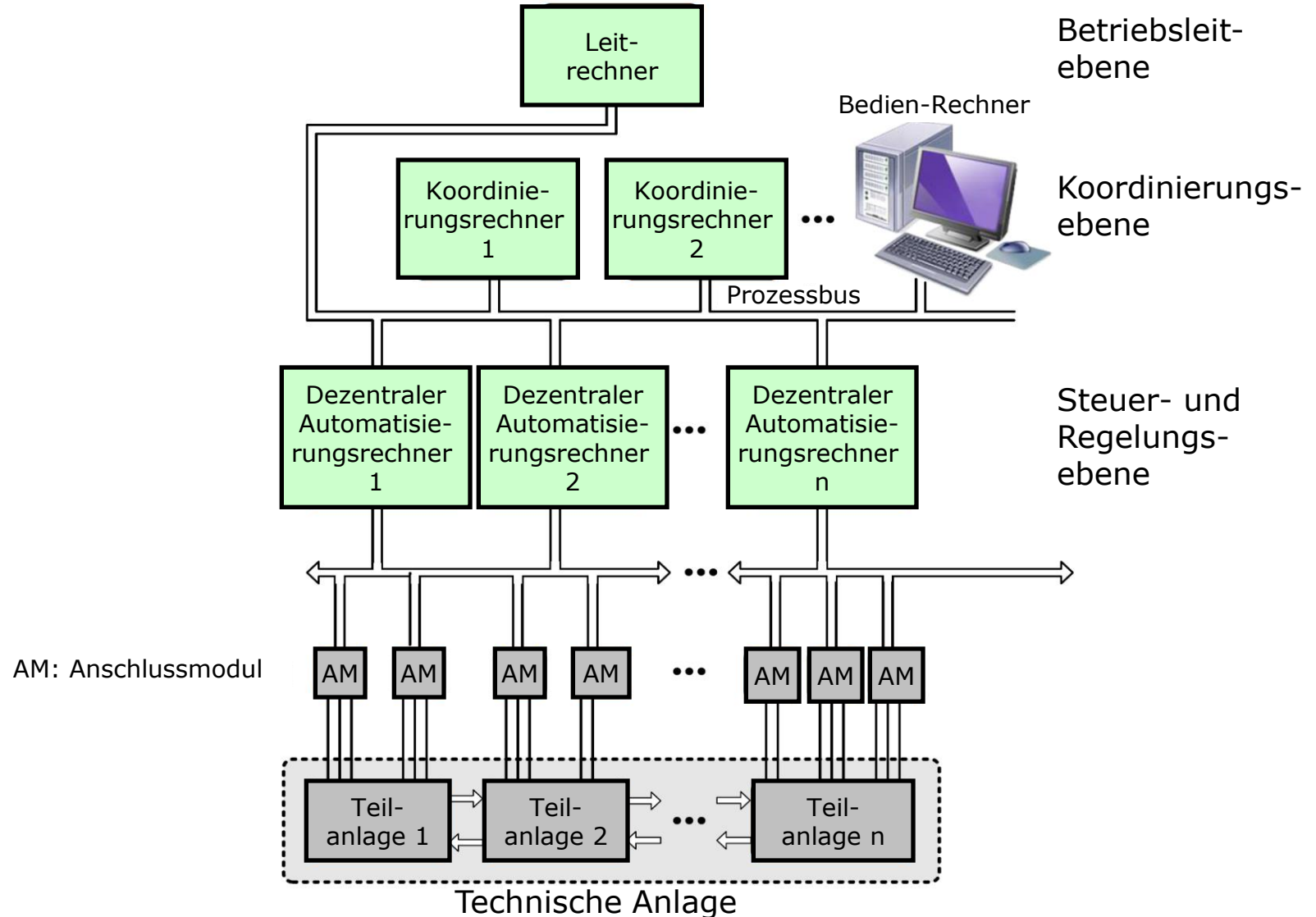
	Technischer Prozess als Einheit betrachtet (zentrale Prozess-Struktur)		Technischer Prozess in Teilprozesse gegliedert (dezentrale Prozess-Struktur)	
	Örtlich zentrale Anordnung der Automatisierungsgeräte	Örtlich dezentrale Anordnung der Automatisierungsgeräte	Örtlich zentrale Anordnung der Automatisierungsgeräte	Örtlich dezentrale Anordnung der Automatisierungsgeräte
Wirkungsmäßig zentrale Automatisierungsstruktur	Z Z Z	Z D Z	D Z Z	D D Z
Wirkungsmäßig dezentrale Automatisierungsstruktur	Z Z D	Z D D	D Z D	D D D

1. Stelle 2. Stelle 3. Stelle



2.5 Automatisierungsstrukturen

■ Einführung einer Hierarchie von Automatisierungseinheiten

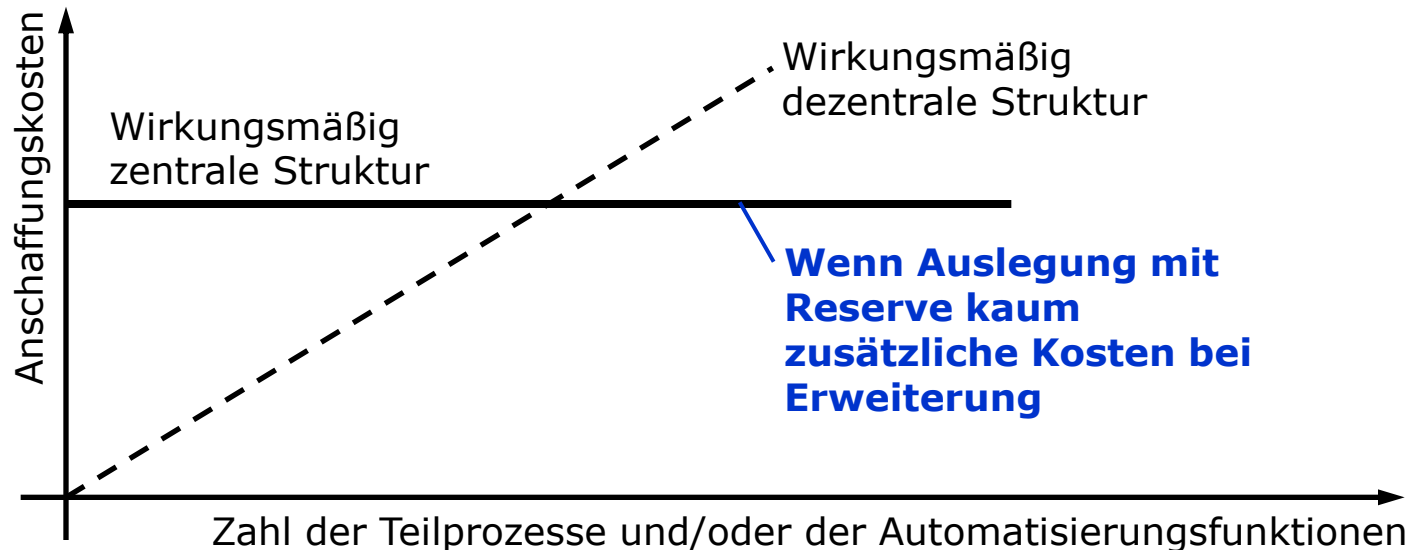


2.5 Automatisierungsstrukturen

■ Kriterien zur Bewertung/Vergleich von Automatisierungsstrukturen

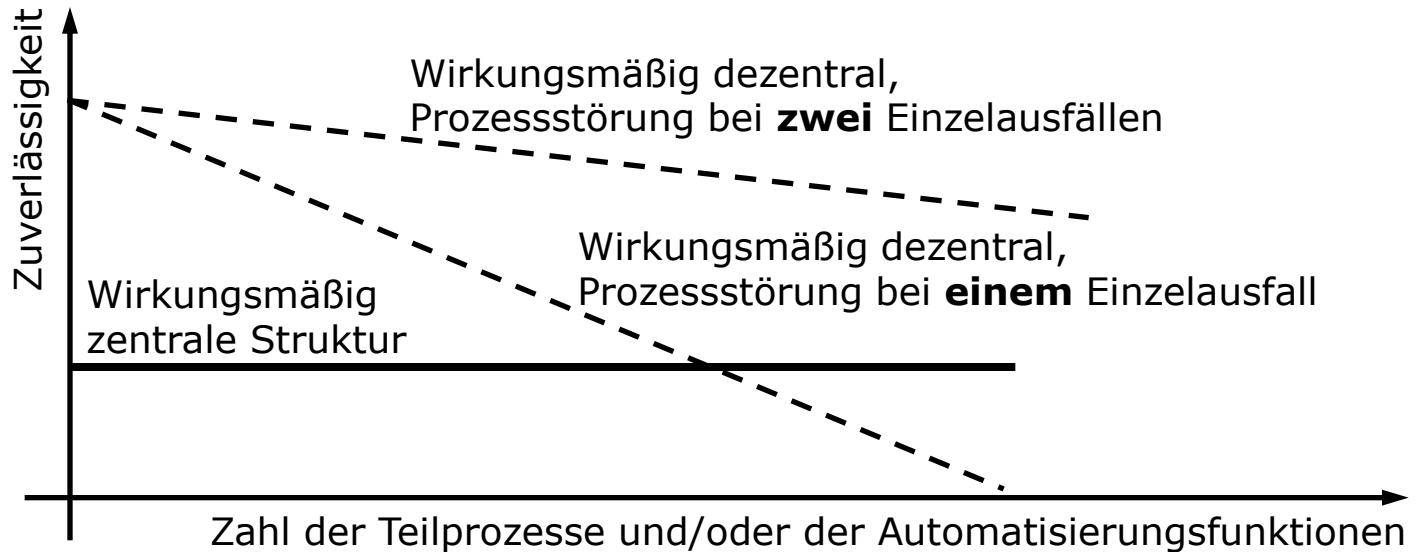
- **Kosten** für Geräte, Verkabelung, Software, Pflege und Wartung
- **Teilverfügbarkeit** bei Hardware-Ausfällen oder bei Software-Fehlern
- **Flexibilität** bei Änderungen
- **Koordinierung** der Teilprozesse und **Optimierung** des Gesamtprozesses
- **Bedienbarkeit**

■ Vergleich bzgl. Anschaffungskosten



2.5 Automatisierungsstrukturen

■ Vergleich bzgl. Zuverlässigkeit des Betriebes



Normalfall: schwache Verkopplung



kein Totalausfall bei Ausfall von zwei oder mehr Automatisierungseinheiten



Zuverlässigkeit des Betriebes bei einer wirkungsmäßig dezentralen Struktur höher als beim Einsatz eines zentralen Prozessrechners

2.5 Automatisierungsstrukturen

■ Vorteile einer dezentralen Automatisierungsstruktur

- Flexibilität bei Änderungen
- Koordination der Teilprozesse
- Optimierung des Gesamtprozesses
- Störfall-Lokalisierung
- höhere Transparenz

■ Nachteil

- zusätzlicher Aufwand zur Kommunikation der einzelnen Automatisierungseinheiten

2.5 Automatisierungsstrukturen

■ Bewertung von Automatisierungsstrukturen

Z: zentrale Struktur
D: dezentrale Struktur

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

**Produkt-
automatisierung**

**Anlagen-
automatisierung**

**Kfz-
Elektronik**

- └─ wirkungsmäßige Automatisierungsstruktur
- └─ örtliche Struktur der Automatisierungsgeräte
- └─ Struktur des technischen Prozesses



so dezentral wie möglich, so zentral wie nötig!

2.6 Redundanz und Fehlertolerante Strukturen

■ Prinzip der Fehlertoleranz

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

■ Formen der Redundanz

- Hardware-Redundanz
- Software-Redundanz
- Messwert-Redundanz: redundante oder abhängige Messgrößen
- Zeit-Redundanz: mehrfache Abfrage des gleichen Messwertes in bestimmten Zeitabständen
- Hardware -und Softwareredundanz: bedeutet höherer Aufwand, führt aber zu erhöhter Verfügbarkeit und Sicherheit.

2.6 Redundanz und Fehlertolerante Strukturen

■ Stufen der Fehlertoleranz

- volle Fehlertoleranz (fail operational)
- verringerte Leistungsfähigkeit (fail soft, graceful degradation)
- Übergang in einen sicheren Zustand (fail safe)

■ Arten der 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.

2.6 Redundanz und Fehlertolerante Strukturen

■ Zielsetzung der Hardware-Redundanz

Erkennung von Ausfällen der Hardware

■ Einsatzprinzip

m-von-n-Redundanz

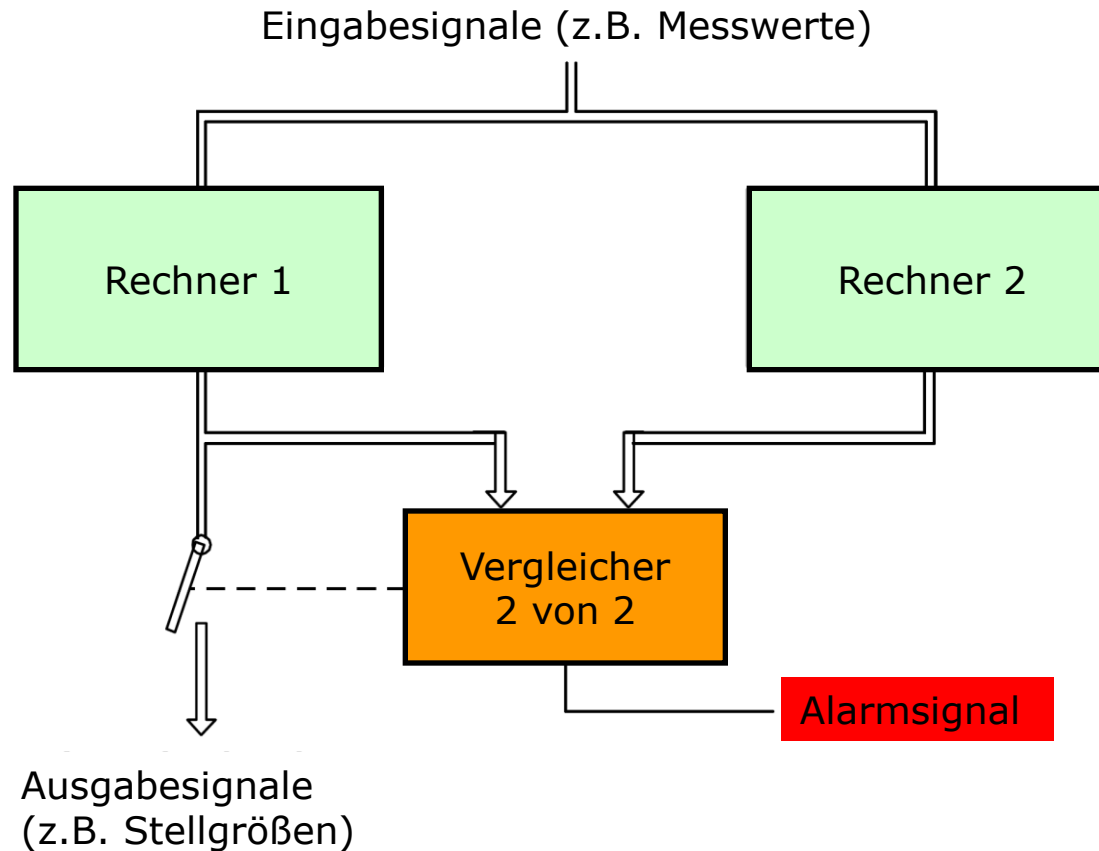
- Mehrheitsentscheid
- Fehler erst bei Mehrfachdefekten

■ Realisierung der Redundanz

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

2.6 Redundanz und Fehlertolerante Strukturen

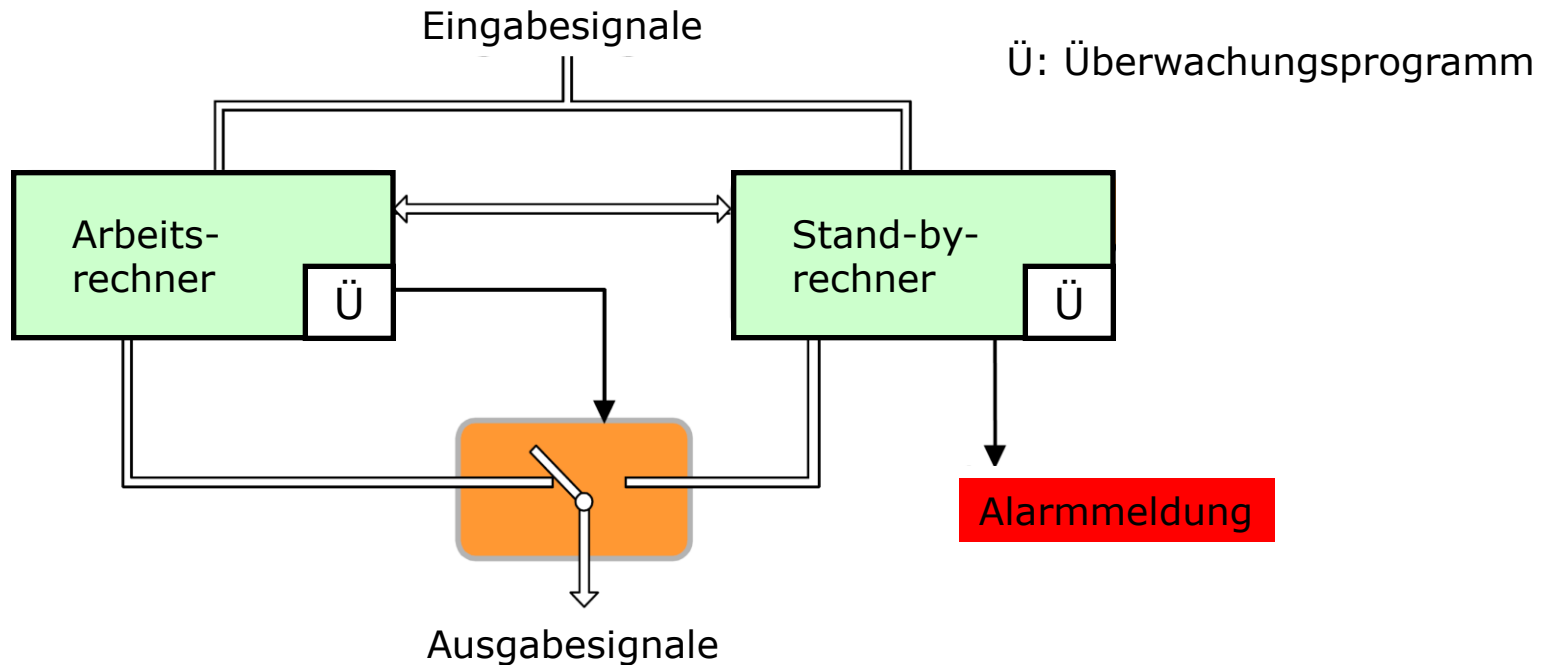
■ Doppelrechner-Struktur mit statischer Redundanz



Keine Fehlertoleranz, Betrieb wird beim Auftreten eines Fehlers abgebrochen!

2.6 Redundanz und Fehlertolerante Strukturen

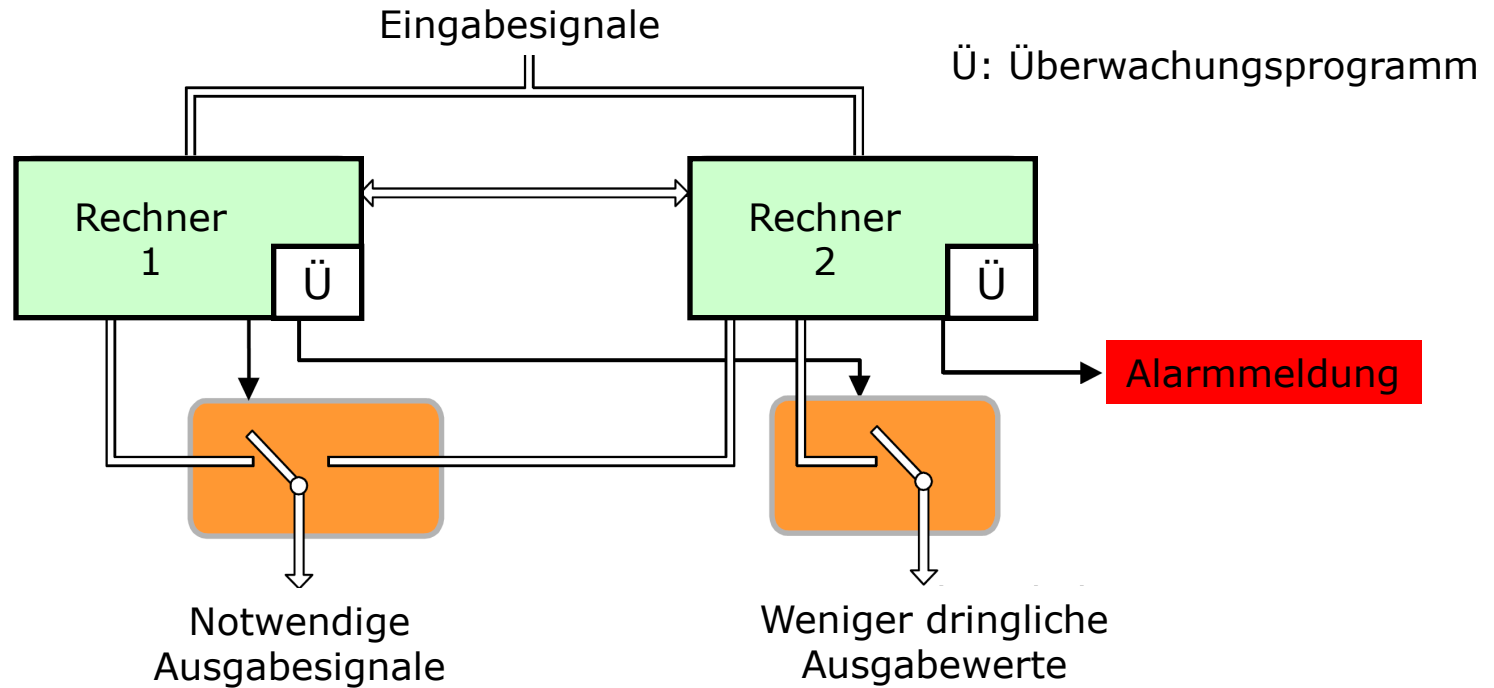
■ Doppelrechner-Struktur mit dynamisch blinder Redundanz



Bei Erkennung eines Fehlers wird auf den Stand-by-Rechner umgeschaltet.

2.6 Redundanz und Fehlertolerante Strukturen

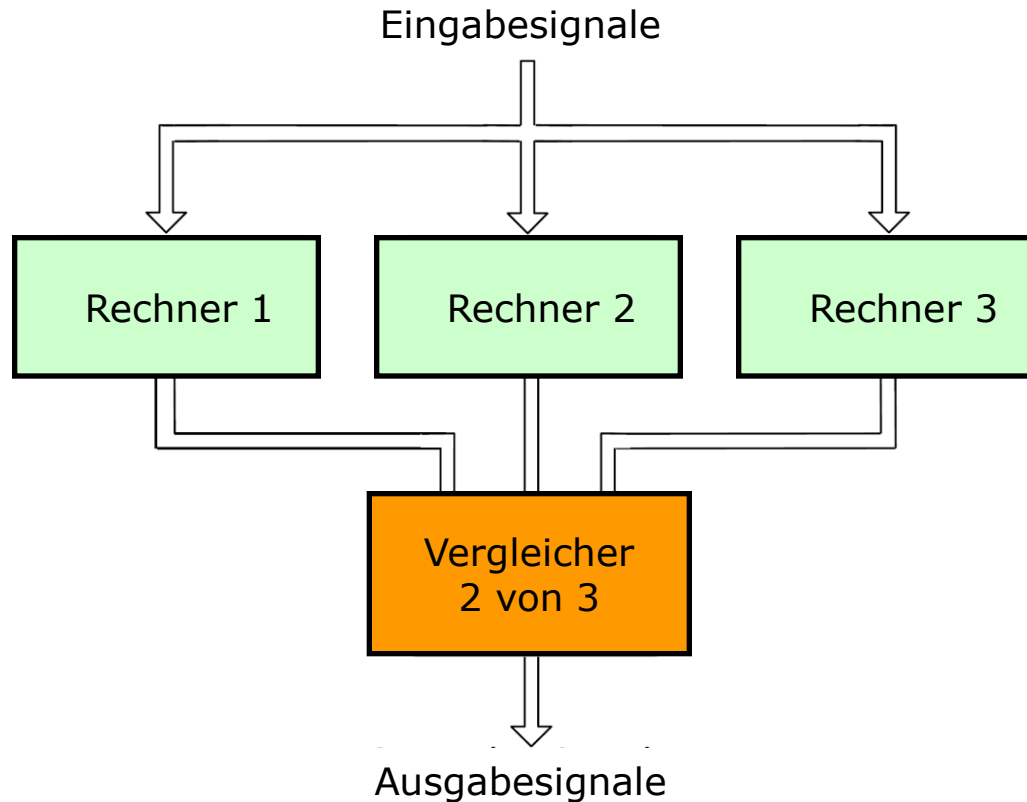
■ Doppelrechner-Struktur mit dynamisch funktionsbeteiligter Redundanz



Bei Erkennung eines Fehlers übernimmt Rechner 2 voll die Aufgaben von Rechner 1, der dann seine Aufgaben abwirft!

2.6 Redundanz und Fehlertolerante Strukturen

■ Drei-Rechner-Struktur mit statischer Redundanz



Prozesssignal wird zum technischen Prozess durchgeschaltet, wenn mindestens zwei von drei Ausgangssignalen übereinstimmen.

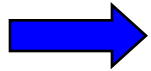
2.6 Redundanz und Fehlertolerante Strukturen

■ Zielsetzung der Software-Redundanz

Erkennung von Fehlern in der Software

■ Redundanzmaßnahmen bei Software

- verschiedenartiger Aufbau von Programmteilen
- Gleiche Eingangsdaten müssen gleiche Ergebnisse liefern.



Mehrfacher Einsatz der gleichen Software unsinnig, da mit mehrfachen fehlerhaften Programmteilen nichts gewonnen wäre!

2.6 Redundanz und Fehlertolerante Strukturen

■ Diversitäre Software

Diversität = Verschiedenartigkeit von Software bei gleicher Funktion

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

■ Ausführung diversitärer Software-Teile

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